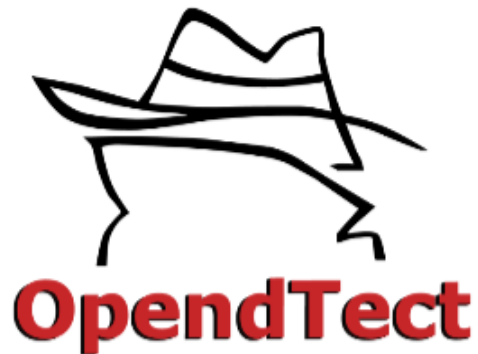


OpenTect Pro & dGB Plugins

Documentation - 7.0



Created by



Copyright © 2002-2024 by dGB Beheer B.V.

All rights reserved. No part of this publication may be reproduced and/or published by print, photo print, microfilm or any other means without the written consent of dGB Beheer B.V.

Under the terms and conditions of any of the following three license types, license holders are permitted to make hard copies for internal use:

- GNU GPL
- OpendTect Pro License
- Academic License

Table of Contents

OpendTect Pro & dGB Plugins Documentation - 7.0	1
Table of Contents	3
1 OpendTect Pro	21
1.1 PetrelDirect	22
1.1.1 Survey Setup From Petrel*	25
1.1.2 Import From Petrel*	29
1.1.2.1 Import Fault From Petrel*	31
1.1.2.2 Import FaultStickSet From Petrel*	33
1.1.2.3 Import Horizon From Petrel*	34
1.1.2.3.1 Import 2D Horizon From Petrel*	35
1.1.2.3.2 Import 3D Horizon From Petrel*	37
1.1.2.4 Import Seismic From Petrel*	39
1.1.2.4.1 Import 2D Seismic From Petrel*	40
1.1.2.4.2 Import 3D Seismic From Petrel*	42
1.1.2.4.3 Import 3D prestack Seismic From Petrel*	44
1.1.2.5 Import Wavelet From Petrel*	46

1.1.2.6 Import Well From Petrel*	47
1.1.2.7 Import Pointsets and Polygons From Petrel*	50
1.1.2.8 Import All Data From Petrel* at Survey Creation	52
1.1.3 Export To Petrel*	57
1.1.3.1 Export Fault To Petrel*	58
1.1.3.2 Export FaultStickSet To Petrel*	59
1.1.3.3 Export Horizon To Petrel*	61
1.1.3.3.1 Export 3D Horizon To Petrel*	62
1.1.3.4 Export Seismic* To Petrel	67
1.1.3.4.1 Export 2D Seismic To Petrel*	68
1.1.3.4.2 Export 3D Seismic To Petrel*	69
1.1.3.5 Export Wavelet To Petrel*	70
1.1.3.6 Export Well to Petrel*	71
1.1.3.7 Export Pointsets and Polygons To Petrel*	72
1.1.4 Link OpendTect and Petrel* projects	73
1.1.5 Settings Window	75
1.2 Basemap	77
1.3 PDF3D	101

PDF3D	102
glTF	108
1.4 Thalweg Tracker	110
1.5 Advanced Ray Tracer	117
1.6 Horizon from Well Markers	118
1.7 Well Table	120
1.8 Create Body	121
1.9 Horizon Mathematics (2D and 3D)	123
2 Faults and Fractures	126
2.1 Attributes	126
2.1.1 Fault Likelihood (FL)	127
2.1.2 Thinned Fault Likelihood (TFL)	130
2.1.3 Create TFL from FL	136
2.1.4 Basement Fracture Likelihood (BFL)	137
2.1.5 Thinned Basement Fault Likelihood (TBFL)	140
2.1.6 Convolve (Edge Enhancement)	143
2.1.7 Curvature	144
2.1.8 Dip - Azimuth Attributes	145

2.1.9 Fault Dip - Azimuth	146
2.1.10 Fault Painter	149
2.1.11 Finger Vein Attribute	153
2.1.12 Fracture Density - Proximity	160
2.1.13 Gradient - Azimuth Of Curvature	165
2.1.14 Semblance	170
2.1.15 Similarity	171
2.2 Filters	172
2.2.1 Dip-Steered Median Filter	173
2.2.2 Fault Enhancement Filter	174
2.2.3 Ridge Enhancement Filter	175
2.2.4 Edge-Preserving Smoother	176
2.2.5 Skeletonization	180
2.3 Planes	183
2.3.1 Extract Planes	184
2.3.2 Voxel Connectivity	189
2.4 Tools	190
2.4.1 Blend Attributes & Faults	191

2.4.2 Fault Dip Calculation	195
2.4.3 Show Rose Diagram	197
2.4.4 Trim Horizons Against Faults	201
3 Machine Learning	202
3.1 Introduction and Terminology	202
Terminology	206
Resources	211
3.2 Python Settings, Data Flow and Data Management	214
3.3 Wells	223
3.3.1 Log-Log Prediction	223
Extract Data	223
Training	228
Apply	236
3.3.2 Lithology Classification	239
3.4 Seismic	240
3.4.1 Horizon-based Seismic Facies (2D and 3D)	243
3.4.2 Seismic Classification (Supervised 2D and 3D)	243
Extract Data	245

Input Data	246
Labels	247
Training	252
Apply	263
3.4.3 Seismic Image Segmentation (Supervised 2D and 3D)	266
Extract Data	269
Training	273
Apply	283
3.4.4 Seismic Image Regression (Supervised 2D and 3D)	286
Extract Data	289
Training	293
Apply	302
3.5 Seismic + Wells models	307
Extract Data	309
Input Data	310
Log Tab	313
Training	316
Apply	327

3.6 Pre-trained Models	329
Seismic - UNet 3D Fault Predictor	333
Seismic - UNet VGG19 2D Fault Predictor	338
Seismic - UNet Fill Missing Traces	341
Seismic - UNet Supersample Bin Size	343
Seismic - Lundin GeoLab AJAX	344
Seismic - Lundin GeoLab DeSmile	346
Seismic - Lundin GeoLab SimpleDenoise	348
Seismic - Lundin GeoLab SimpleHmult	349
Wells – NPHI from RHOB, GR, DT, and MD	351
3.7 Neural Networks	353
3.7.1 Introduction	353
3.7.1.1 Supervised Neural Networks	354
3.7.1.2 Unsupervised Neural Networks	357
3.7.2 Manage Neural Networks	358
3.7.2.1 Data Manager	358
3.7.2.2 Neural Network Management	359
3.7.2.3 Neural Network Information	362

3.7.2.3.1 Supervised Neural Network Information	363
3.7.2.3.2 Unsupervised Neural Network Information	364
3.7.2.4 Import GDI Networks Window	366
3.7.3 Pattern Recognition (using Picksets)	368
3.7.4 Property Prediction (using Well Logs)	375
3.7.4.1 Balance Data	378
3.7.4.2 NN Lithology Codes	380
3.7.5 Training and Application	382
3.7.5.1 Unsupervised Training	383
3.7.5.1.1 Quick UVQ	384
3.7.5.1.2 Quality-based UVQ Stacking	387
3.7.5.2 Supervised Training from Picksets	389
3.7.5.3 Supervised Training from Well Data	392
3.7.5.4 Application	393
4 Dip-Steering	396
4.1 Background	396
4.2 Create Steering Data	401
4.2.1 Import Steering Data	404

4.2.2 Create SteeringCube	406
4.3 Filter	412
4.4 Display SteeringCube	413
4.5 Attributes with Steering	414
4.5.1 Curvature Analysis	414
4.5.1.1 Volumetric Curvature	417
4.5.1.2 Horizon-Based Curvature	418
4.5.2 Dip	421
4.5.3 Dip Angle	423
4.5.4 Position	424
4.5.5 Reference Shift	426
4.5.6 Similarity	428
4.5.7 Volume Statistics	431
4.5.8 Perpendicular Dip Extractor	433
4.6 Benchmark SteeringCube Creation	434
4.6.1 Speed vs Algorithm and Calculation Cube Size	435
4.6.2 Visual Quality Check	436
4.6.3 Crossline Dip Attribute	438

4.6.4 Filtering of the SteeringCubes	441
4.6.5 Steered Similarity Attribute	445
4.6.6 Choosing a Steering Algorithm	446
4.7 Track Using Inversion	447
4.8 Track Using Inversion+	456
5 HorizonCube	457
5.1 Introduction	457
5.2 HorizonCube Control Center	460
5.2.1 Processing	461
5.2.1.1 Create A SteeringCube	462
5.2.1.2 Filter A SteeringCube	463
5.2.1.3 Create HorizonCube	464
5.2.1.3.1 Model Driven Settings	469
5.2.1.3.2 Data Driven Settings	470
5.2.1.3.2.1 Advanced Options	474
5.2.1.3.2.2 Continuous Events	477
5.2.1.3.2.3 Truncated Events	479
5.2.1.4 2D HorizonCube Processing	481

5.2.1.4.1 Preparing the Framework	483
5.2.1.4.2 Processing 2D HorizonCube	488
5.2.1.4.3 Correlate 2D HorizonCube Regions	494
5.2.1.5 Create HorizonCube - Multiple Lines	496
5.2.1.6 Convert To SteeringCube	497
5.2.1.7 Create 2D Seismic Lattice	498
5.2.1.8 Create Horizons From A SteeringCube	501
5.2.1.9 Edit Horizons From SteeringCube	509
5.2.1.10 Interpolate Logs Along A HorizonCube	510
5.2.2 Analysis	512
5.2.2.1 3D Slider - HorizonCube Visualization	513
5.2.2.2 Extract Attributes Along HorizonCube	518
5.2.2.3 Make Crossplots Of HorizonCube Data	521
5.2.3 Tools	523
5.2.3.1 Add More Iterations	524
5.2.3.2 Check Crossings Between Horizons	526
5.2.3.3 Extract Horizons	528
5.2.3.4 Grid A Horizon	531

5.2.3.5 Filter A HorizonCube	534
5.2.3.6 Get Continuous HorizonCube	536
5.2.3.7 Grid HorizonCubes	537
5.2.3.8 Merge HorizonCubes	538
5.2.3.9 Modify / Recalculate HorizonCube	539
5.2.3.10 Trim Against Faults	542
5.2.3.11 Truncate HorizonCube	546
5.3 HorizonCube Attributes	547
5.4 Import HorizonCube Surface Data	555
5.5 Export HorizonCube	556
5.6 Manage HorizonCube	558
5.7 HorizonCube Display Options	561
6 Sequence Stratigraphic Interpretation System	566
6.1 Introduction	566
6.1.1 SSIS Toolbar	570
6.2 Interpretation Window	571
6.2.1 Overview	572
6.2.2 Select, Define a Depositional Model	577

6.2.3 Interpretation Workflow	581
6.2.4 2D Interpretation Window	587
6.2.5 Display System Tracts	590
6.2.6 SSIS Display Settings	592
6.2.7 Create Statistical Curves	594
6.3 HorizonCube Slider	596
6.4 Wheeler Transform, Wheeler Scene	598
6.4.1 Wheeler Scene	599
6.4.2 Create Wheeler Output (2D, 3D)	602
6.5 Flatten Horizon, Seismics	604
6.5.1 Flatten	605
6.5.2 Unflatten HorizonCube	609
6.6 System Tracts Attributes	611
7 Well Correlation Panel	613
7.1 Introduction	613
7.2 WCP Main Window	615
7.3 Correlation Displays and Settings	622
7.4 Pick Markers and Correlate	630

8 SynthRock	631
8.1 Introduction	631
8.2 Volume Synthetics	634
8.3 Stochastic Pseudowell Modeling	638
8.3.1 Add New Modeling Node	640
8.3.2 Model Definition	648
8.3.2.1 Random Distribution	652
8.3.2.2 PDF Distribution	653
8.3.2.3 Math-based Layer Properties	654
8.3.3 Analysis of the Existing Wells	658
8.3.4 Well-Log - based Generation	668
8.3.5 Manage layer Properties	672
8.3.6 Export Synthetic Dataset	677
8.4 Profile Modeling	682
8.4.1 Simple Wedge Model/Property Variations Using one Well	685
8.4.2 Model an existing line through well(s)	694
8.4.3 Put Together a Structural Situation from Scratch	707
8.4.4 Add, Edit Existing Well	731

8.4.4.1 Define Property Changes	734
8.4.5 Add, Edit Control Profile	738
8.4.6 Use Horizon	748
8.4.7 Set Process Parameters	753
8.4.8 Dump Model to File	757
9.5 Fluid Replacement	759
8.5.1 Define Fluid Contents	760
8.5.2 Fluid Replacement Parameters	763
8.6 HitCube Stochastic Inversion	770
8.6.1 Input Data and Inversion Type	774
8.6.2 HitCube Analysis	775
8.6.2.1 HitCube Parameters QC Evaluation	778
8.6.2.2 Define Output Layer Properties	779
8.6.3 Output Data - Inversion Batch Processing	780
9 Velocity Model Building	781
9.1 Introduction	781
9.2 Vertical Velocity Analysis	783
9.3 Horizon-based Velocity Update	798

9.4 Input-Output	804
9.4.1 Prestack Event Import	805
9.4.2 Prestack Events Export	807
9.5 Velocity Display	809
9.6 Velocity Correction	811
9.7 VMB Specific Gridding Step - Gridding of Velocity Picks	813
9.8 VMB Specific Gridding Step - Surface-limited Filler	815
10 Fluid Contact Finder (formerly CCB)	817
10.1 Introduction	817
10.2 FCF Main Window	819
10.3 FCF Analysis	821
10.4 Local FCF Attribute	826
11 Applications	827
11.1 How to Make the ChimneyCube	827
11.1.1 Workflow	827
11.1.2 Picking Example Locations	829
11.1.3 Neural Network Training	831
11.1.4 Evaluation and Application of the Trained Neural Network	833

11.2 Dip-Steered Median Filter	834
11.2.1 Example Results	837
11.2.2 Create a Dip-Steered Median Filter	839
11.2.3 Note	840
12 Default Attribute Sets	841
12.1 Evaluate Attributes	842
12.2 dGB Evaluate Attributes	843
12.3 NN Chimney Cube	844
12.4 NN Fault Cube	849
12.5 NN Fault Cube Advanced	853
12.6 NN Salt Cube	854
12.7 NN Slump Cube	857
12.8 Unsupervised Waveform Segmentation	860
12.9 Ridge-enhancement Filter	863
12.10 Dip-Steered Median Filter	868
12.11 Dip-Steered Diffusion Filter	869
12.12 Fault Enhancement Filter	870
12.13 Fault Enhancement Attributes	872

12.14 Fault Enhancement Filter (Expert)	873
13 Licenses	874
13.1 Install or Edit Licenses	876
13.2 Clear License Installation	878
13.3 System Information	880
14 References	883
Appendix A - Synthetic Data Generation	885
Glossary	896

1 *OpendTect Pro*





[OpendTect_Pro](#) is an extended version of OpendTect for professional users. OpendTect Pro users benefit from the following extra functionality: a direct link to Petrel*; a basemap utility; a PDF3D tool to capture 3D images in pdf format; a Thalweg tracker for auto-tracking seismic facies; and an accurate ray-tracer for computing angle stacks and AVA attributes.

OpendTect Pro can be extended further with commercial plugins developed by dGB and its partners. This document describes the functionality in the Pro layer (that is available to all commercial users) as well as the plugins developed by [dGB_Earth_Sciences](#). The user documentation of plugins developed by other vendors is given in separate documents.

*is a mark of Schlumberger.

1.1 PetrelDirect

PetrelDirect is an OpendTect Pro feature for direct data transfer between OpendTect and Petrel* projects. [Reading data from a Petrel* project](#) is available in various OpendTect workflows via insert icon  , which allows to either access data directly from a Petrel* data store via links or physically copy data to an OpendTect project. Either method gives full potential for manipulation, interpretation and processing. [Writing data to a Petrel* project](#) is also available in various OpendTect workflows by choosing  *PetrelDirect* output format.



Survey > Import and Survey > Export menus have PetrelDirect option for supported objects.

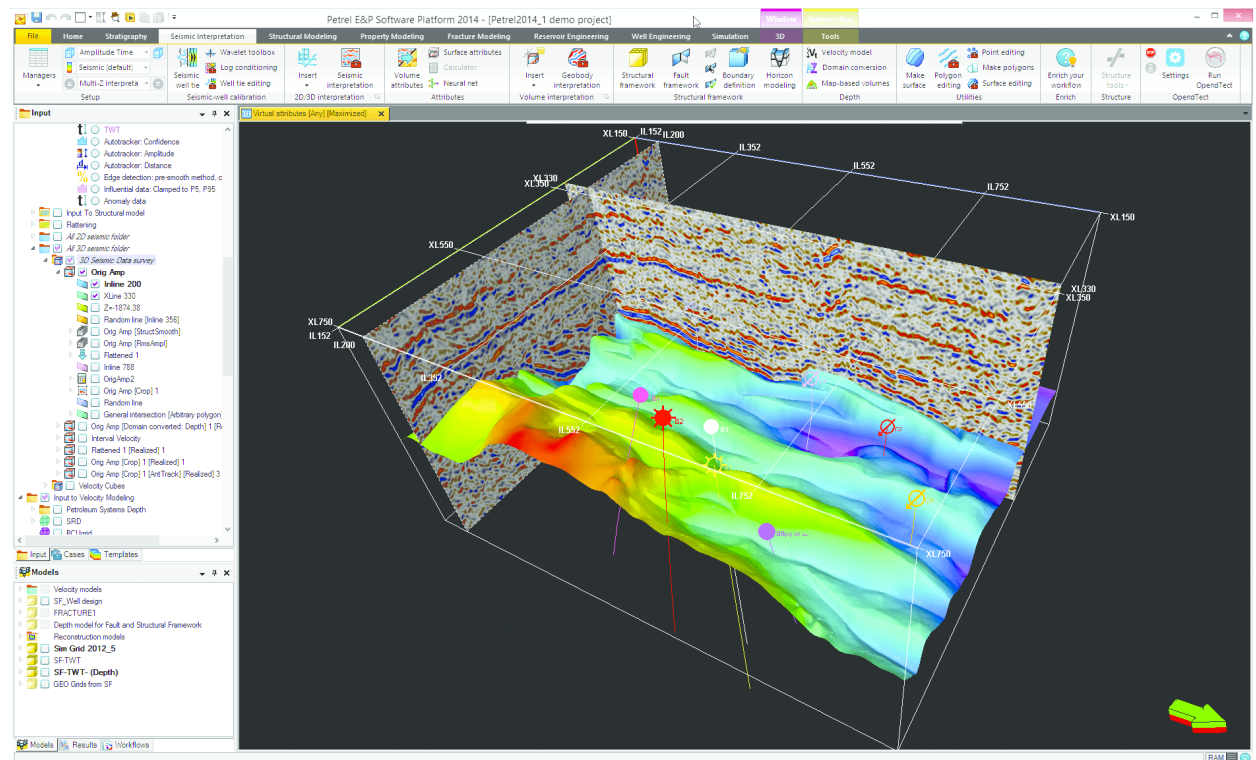
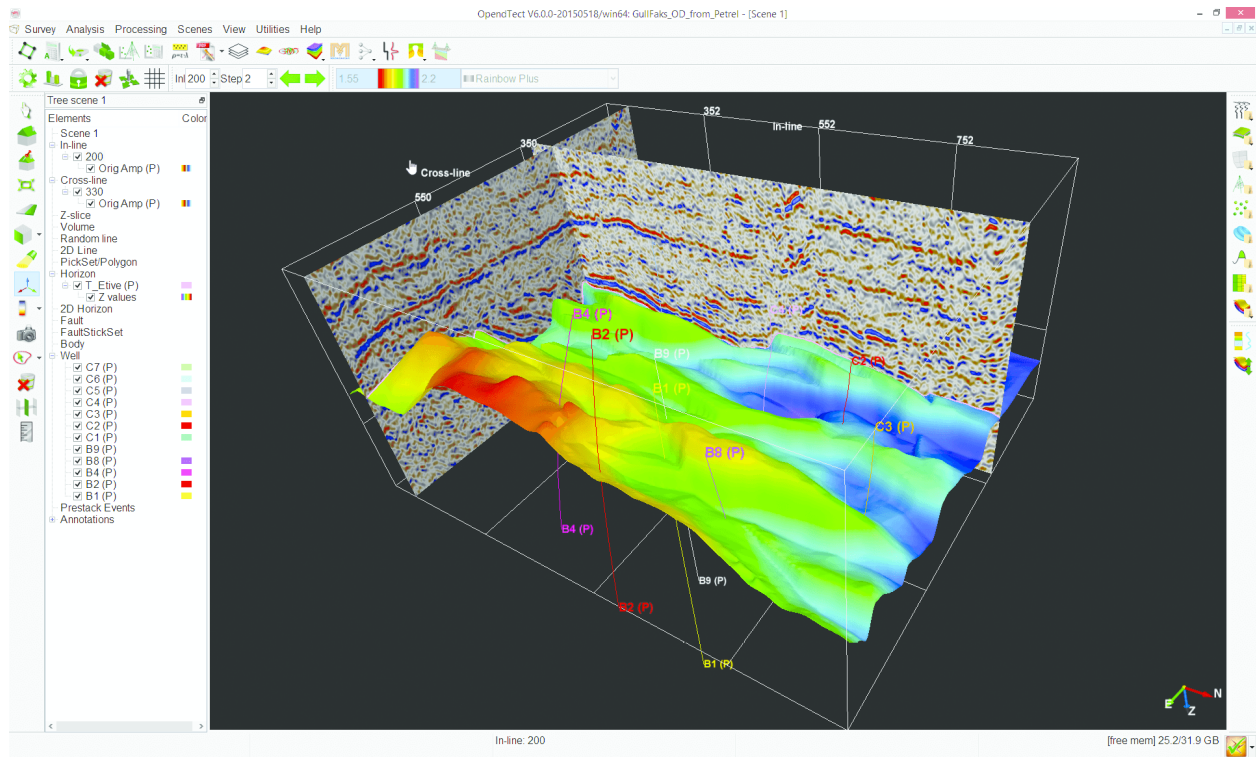
Full two-way access (Petrel* <> OpendTect):

- Faults (since 6.0.4 release) and FaultStickSets
- 3D horizons
- 2D and 3D seismic
- Wavelets
- Wells (tracks, time-depth models and logs)


One way access (Petrel* > OpendTect):




- 2D horizons
- 3D prestack seismic
- Wells (markers)

The images below show an example of an OpendTect project with key data being read directly from the Petrel* data store.





OpenDTect (l) and Petrel (r) showing a linked demo project*

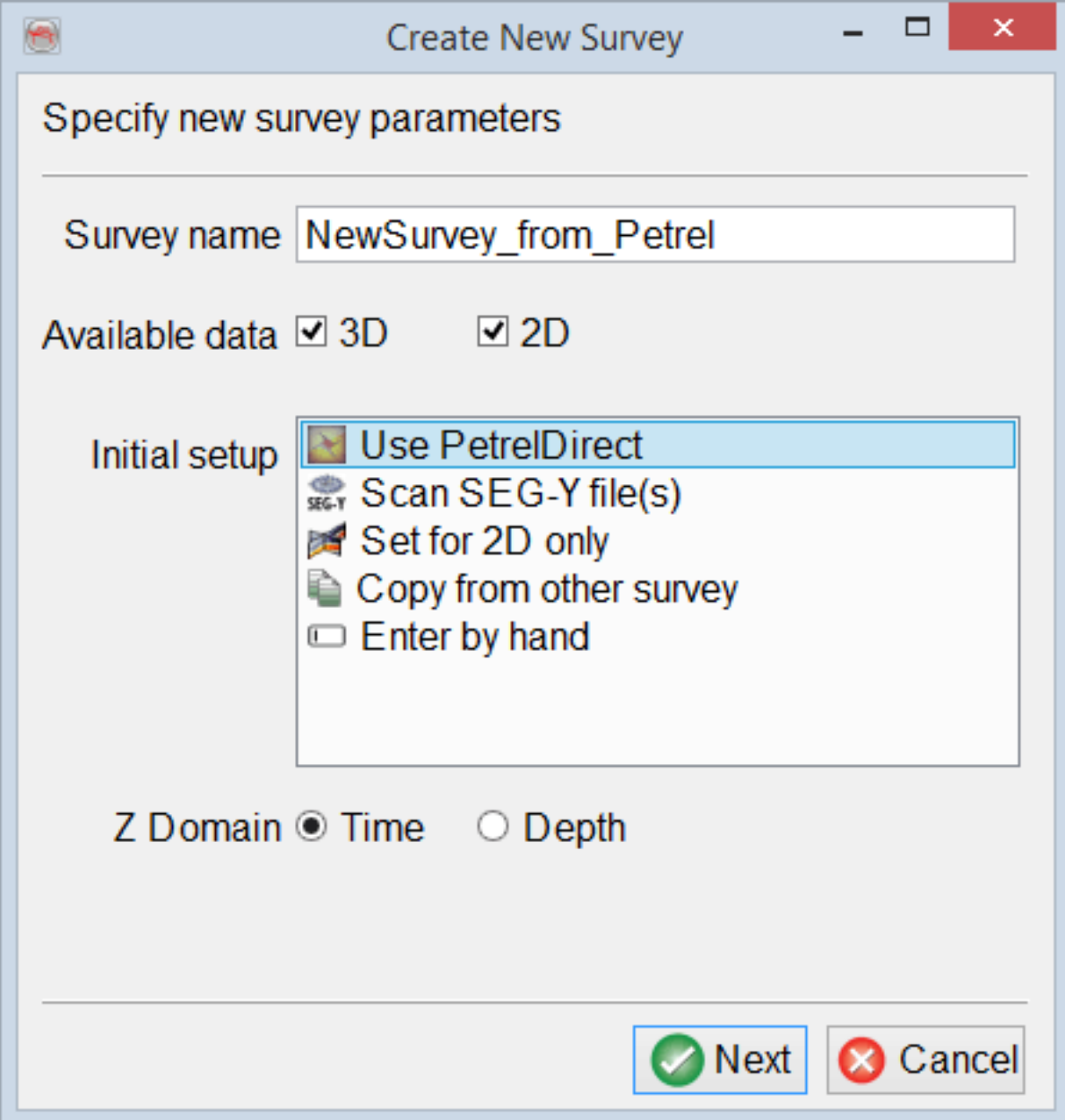
PetrelDirect status button  (in the lower right corner of the main OpenDTect window) helps to monitor the status of the connection to Petrel*. Pushing the button will lead to the [Settings Window](#). The drop-down menu next to it allows to directly control the status of the connection:

-  - disabled (drop-down menu next to it allows to *Enable* connection)
-  - uninitialized (connection is enabled, i.e. will be activated once PetrelDirect is used; drop-down menu next to it allows to either *Disable* connection or *Initialize now*)
-  - active (drop-down menu next to it allows to *Disable* connection)

*is a mark of Schlumberger.

1.1.1 Survey Setup From Petrel*

An OpendTect survey can be quickly set up from an existing Petrel* project. Go to *Survey > Select/Setup* or click on  icon in the main OpendTect window and then click on  icon to *Create New Survey*.



The screenshot shows a dialog box titled "Create New Survey" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area is titled "Specify new survey parameters".

Under "Specify new survey parameters", there is a text input field for "Survey name" containing the text "NewSurvey_from_Petrel".

Below the survey name, there are two checked checkboxes under the heading "Available data": 3D and 2D.

Under the heading "Initial setup", there is a list of options:

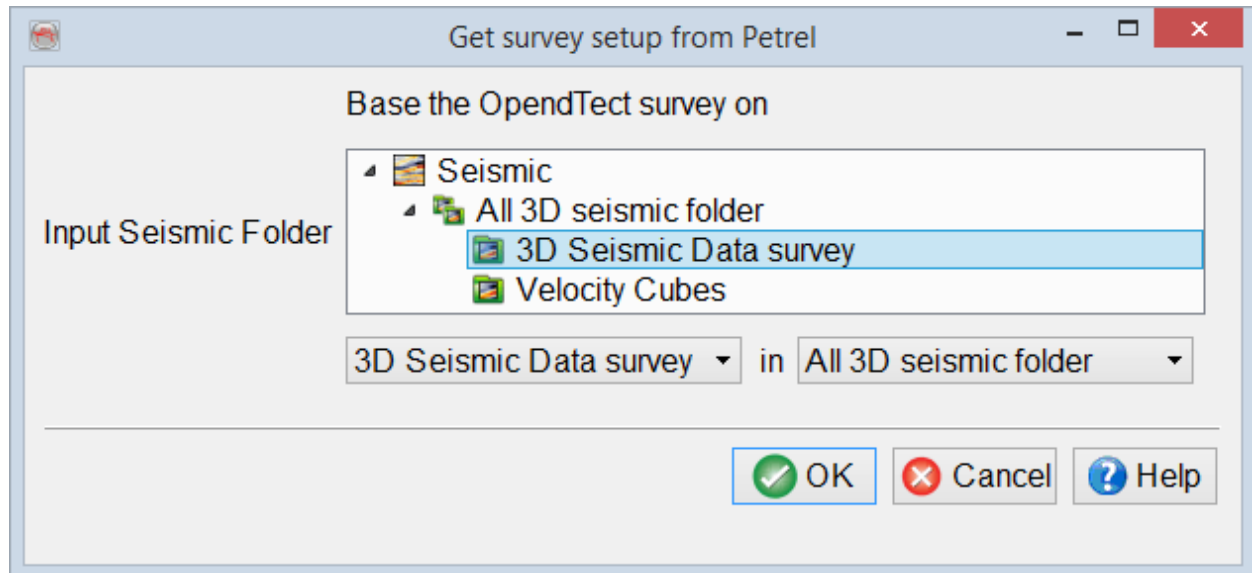
- Use PetrelDirect (highlighted)
- Scan SEG-Y file(s) (with a SEG-Y icon)
- Set for 2D only (with a 2D icon)
- Copy from other survey (with a document icon)
- Enter by hand (with a hand icon)

At the bottom of the dialog, there are two radio buttons for "Z Domain": Time and Depth.

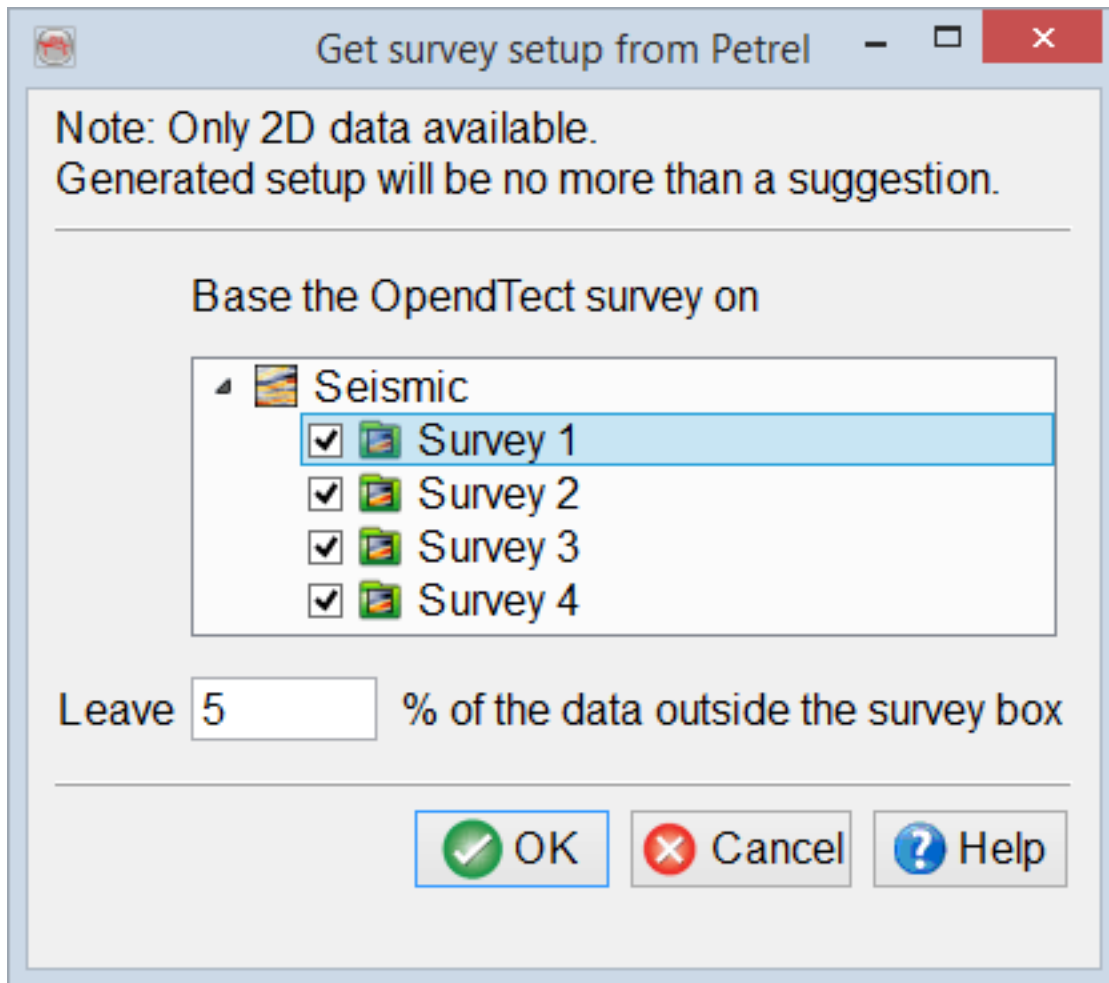
At the bottom right, there are two buttons: "Next" (with a green checkmark icon) and "Cancel" (with a red X icon).

Use *PetrelDirect* option of *Initial setup* forwards to *Get survey setup from Petrel** window which looks different depending on the availability of 2D and 3D seismic survey folders in Petrel* project.

3D: choose one of the 3D seismic folders either in the tree view or using drop-down menus.



2D: choose one or more 2D seismic surveys which will be used to generate a suggested survey setup and set the percentage of data to be left outside the survey box.



The **Edit Survey Parameters** windows opens with all survey with pre-filled parameters from the selected Petrel* project set-up.

Edit Survey Parameters

Survey name:

Location on disk:

Survey type:

Ranges/coordinate settings:

Survey ranges Coordinate settings I/C to X/Y transformation Coordinate System

In-line range: Step Nr. In-lines: 651

Cross-line range: Step Nr. Cross-lines: 951

Z range: Step

Display depths in Meter Feet


Seismic Reference Datum (m)

Free space on disk: 25.40 GB

The survey parameters need to be checked, and edited if required.

*is a mark of Schlumberger.

1.1.2 Import From Petrel*

Reading data from a Petrel* project is available in various OpendTect workflows via insert icon , which allows to either access data directly from a Petrel* data store via links or physically copy data to an OpendTect project. Either method gives full potential for manipulation, interpretation and processing.

 *Survey > Import menus have PetrelDirect option for supported objects.*

There are two ways to use data from a Petrel* project in OpendTect:

- **Link to Petrel*:**
 - no data duplication;
 - data is available only when the Petrel* project is running and PetrelDirect connection is active (i.e. Petrel* license is tied).
- **OpendTect copy:**
 - physical copy in OpendTect format;
 - no restrictions on data access (i.e. Petrel* license is not tied and data is accessible to all users).

The following data can be read from Petrel* to OpendTect:

- Faults (since 6.0.4) and FaultStickSets
- 2D and 3D horizons
- 2D and 3D seismic
- 3D prestack seismic
- Wavelets
- Wells (tracks, time-depth models, logs and markers)



Use Ctrl+A and Ctrl+Shift+A shortcuts to select/de-select all items in the list.

It is now possible to import/export multiple objects.


Batch processing for importing objects is now available for Faults, FaultSticks, Horizons and Seismic Cubes. A check box to use the original name is provided which should be checked in case the same needs to remain identical to the object in the Petrel project. Any name entered into the field adjacent to this will be used as the 'base name' for the objects.

*is a mark of Schlumberger.

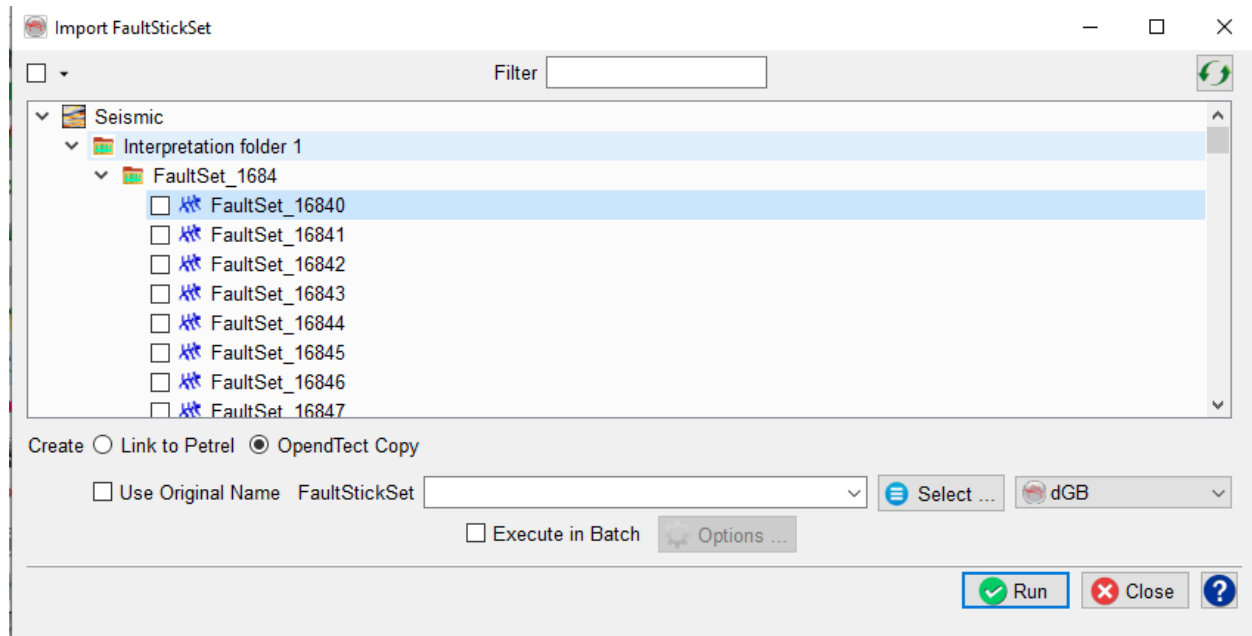
1.1.2.1 Import Fault From Petrel*

Petrel* fault interpretation can be accessed in OpendTect as Faults by clicking on


PetrelDirect insert icon  in the following places:

- *Manage Fault* window (*Survey > Manage > Faults...* or click on  icon in the *Manage Data* toolbar)
- *Select Input Fault(s)* window when adding Fault(s) to the 3D scene, 2D viewer or Basemap by right click on *Fault* in the tree > *Add*)

Select Fault Interpretation(s) window allows to select faults from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.





Fault Import User Interface

OpendTect does not support crossing fault sticks (a fault plane cannot cross itself). If faults were picked on inlines, crosslines and horizontal slices,  only the largest subset of the three will be used to import the faults. Manual editing (removing unwanted sticks) is possible after import and might be required in some cases.

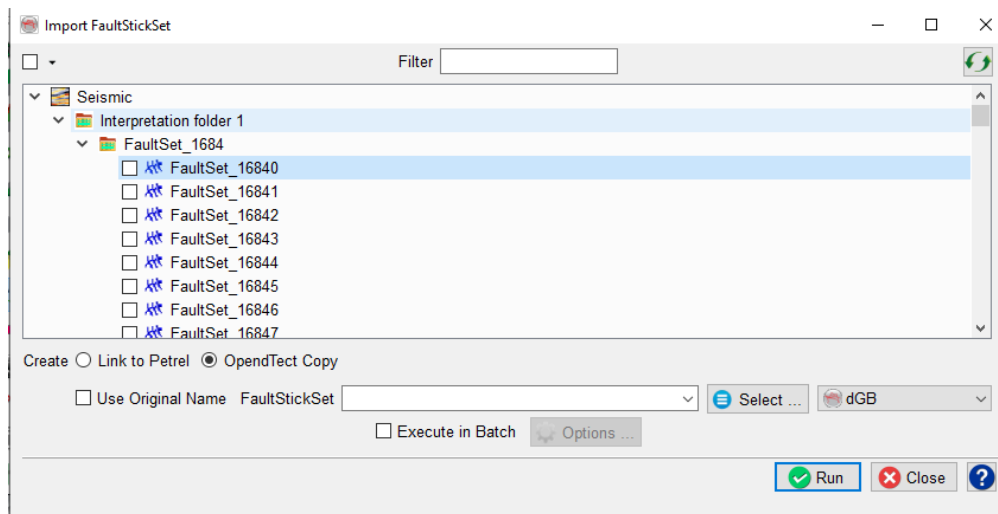
*is a mark of Schlumberger.

1.1.2.2 Import FaultStickSet From Petrel*

Petrel* fault interpretation can be accessed in OpendTect as FaultStickSets by clicking on PetrelDirect insert icon  in the following places:

- *Manage FaultStickSet* window (*Survey > Manage > FaultStickSets...* or click on  icon in the *Manage Data* toolbar)
- *Select Input FaultStickSet(s)* window when adding FaultStickSet(s) to the 3D scene, 2D viewer or Basemap by right click on *FaultStickSet* in the tree > *Add*)

Select Fault Interpretation(s) window allows to select faults from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.




FaultStickSet Import User Interface



FaultStickSet in OpendTect is a set of fault sticks which can be grouped into faults.

*is a mark of Schlumberger.


1.1.2.3 Import Horizon From Petrel*


Petrel 2D and 3D horizons along with horizon attributes can be accessed in OpendTect by clicking on PetrelDirect insert icon  in various workflows.

*is a mark of Schlumberger.

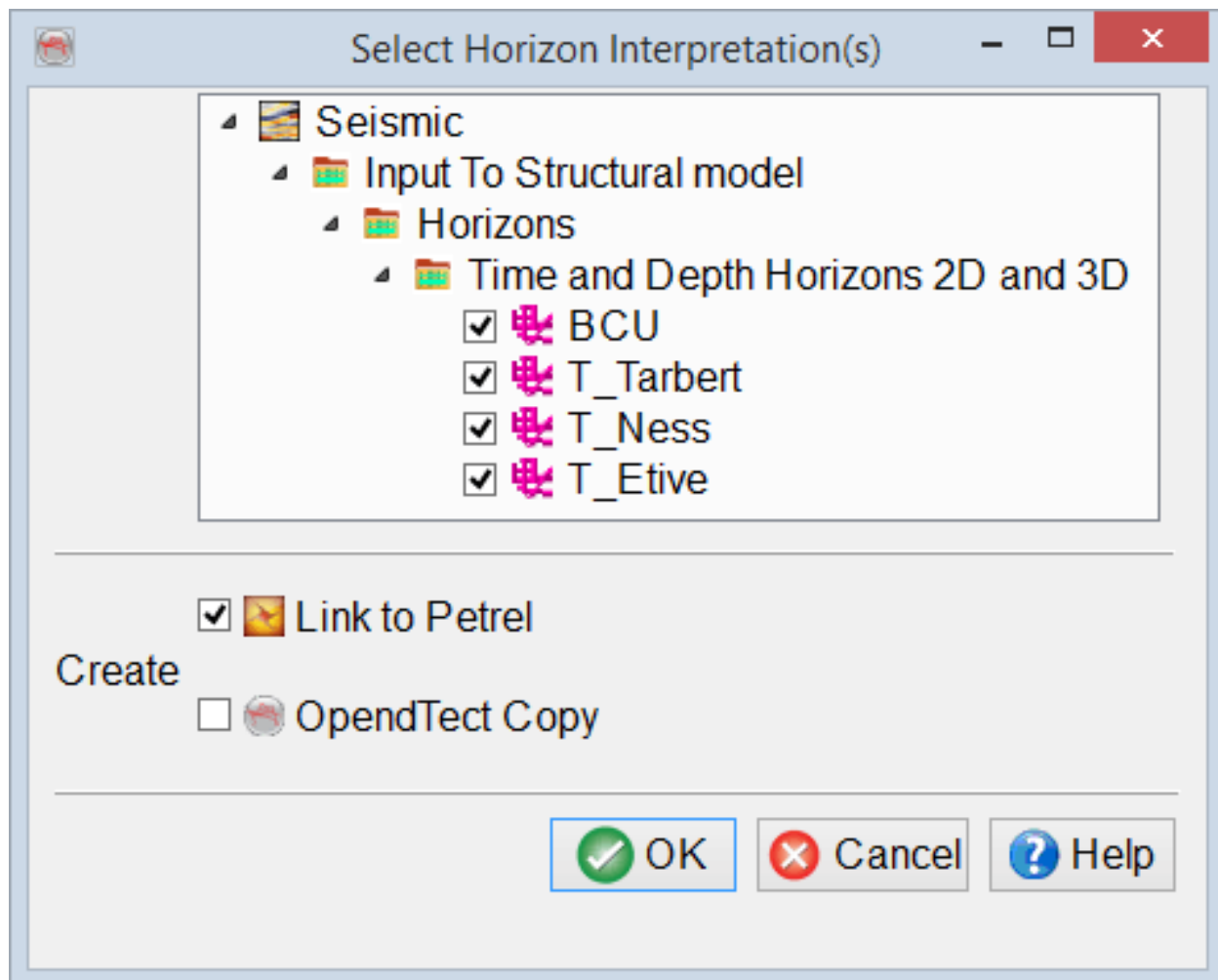
1.1.2.3.1 Import 2D Horizon From Petrel*


Petrel* 2D horizons can be accessed in OpendTect by clicking on PetrelDirect insert

icon  in the following places:

- *Manage 2D Horizons* window (*Survey > Manage > Horizons > 2D...* or click on  icon in the *Manage Data* toolbar)
- *Select Input 2D Horizon(s)* window when adding 2D Horizon(s) to the 3D scene, 2D viewer or Basemap by right click on *2D Horizon* in the tree > *Add*)
- other workflows, including creating flattened scene, etc.


Select Horizon Interpretation(s) window allows to select 2D horizons from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.




 In order to ensure smooth import of a 2D horizon, OpendTect project must contain 2D line geometries with names matching the Petrel* 2D seismic survey associated with that 2D horizon.

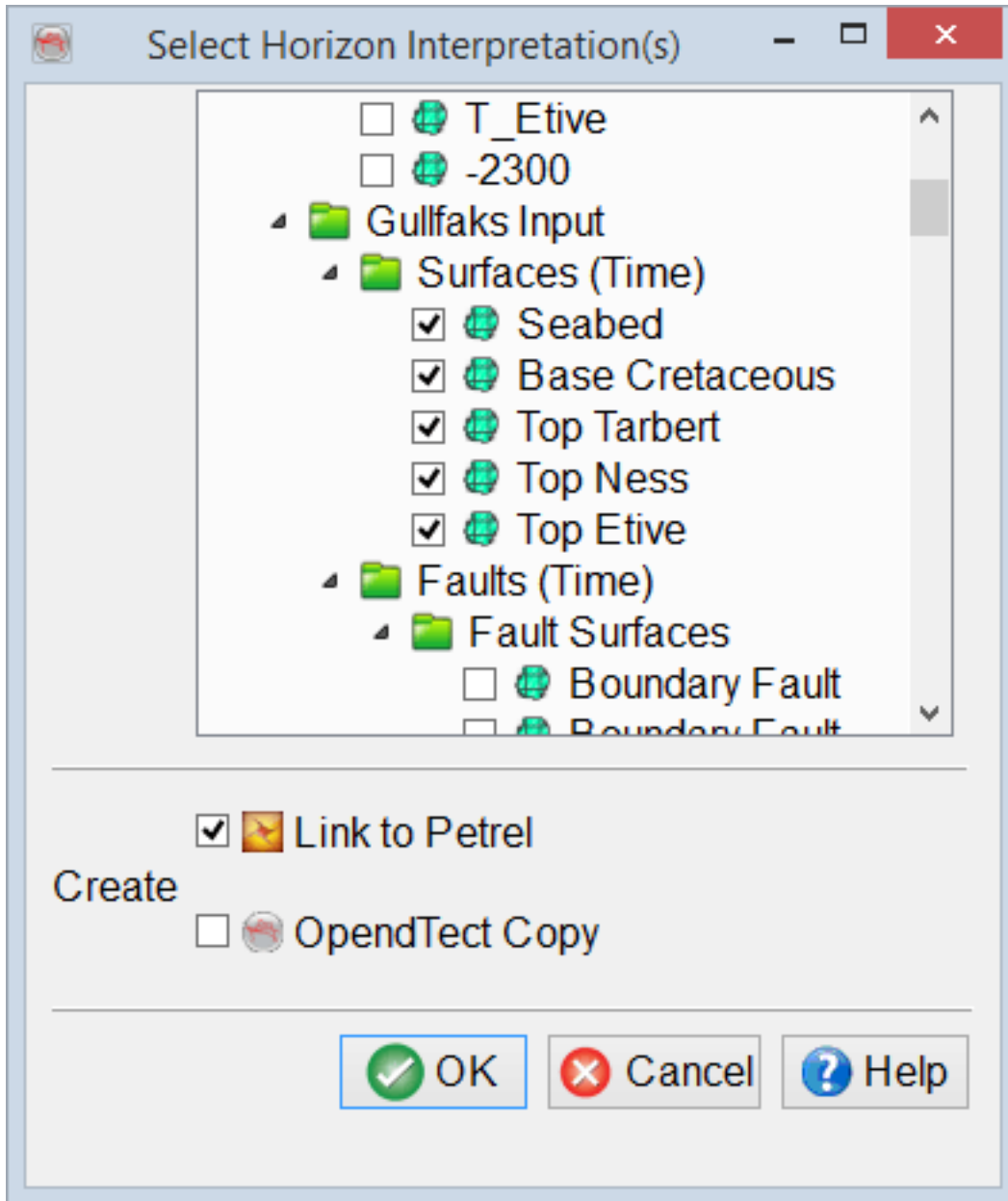
*is a mark of Schlumberger.

1.1.2.3.2 Import 3D Horizon From Petrel*

Petrel 3D horizons and surfaces can be accessed in OpendTect as 3D Horizons by clicking on PetrelDirect insert icon  in the following places:

- *Manage 3D Horizons* window (*Survey > Manage > Horizons > 3D...* or click on  icon in the *Manage Data* toolbar)
- *Select Input Horizon(s)* window when adding 3D Horizon(s) to the 3D scene, 2D viewer or Basemap by right click on *3D Horizon* in the tree > *Add*)
- other workflows, including 3D Horizon gridding and filtering, creating flattened scene, etc.

Select Horizon Interpretation(s) window allows to select 3D horizons and surfaces from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.



OpendTect project is based on a particular 3D survey set up. Petrel 3D horizons associated with 3D seismic surveys which geometries are different and Petrel* surfaces based on grids which are different are snapped to an OpendTect grid during import.*

*is a mark of Schlumberger.


1.1.2.4 Import Seismic From Petrel*


Seismic data from Petrel* project (2D poststack; 3D post- and prestack) can be accessed in OpendTect by clicking on PetrelDirect insert icon  in various workflows.

*is a mark of Schlumberger.

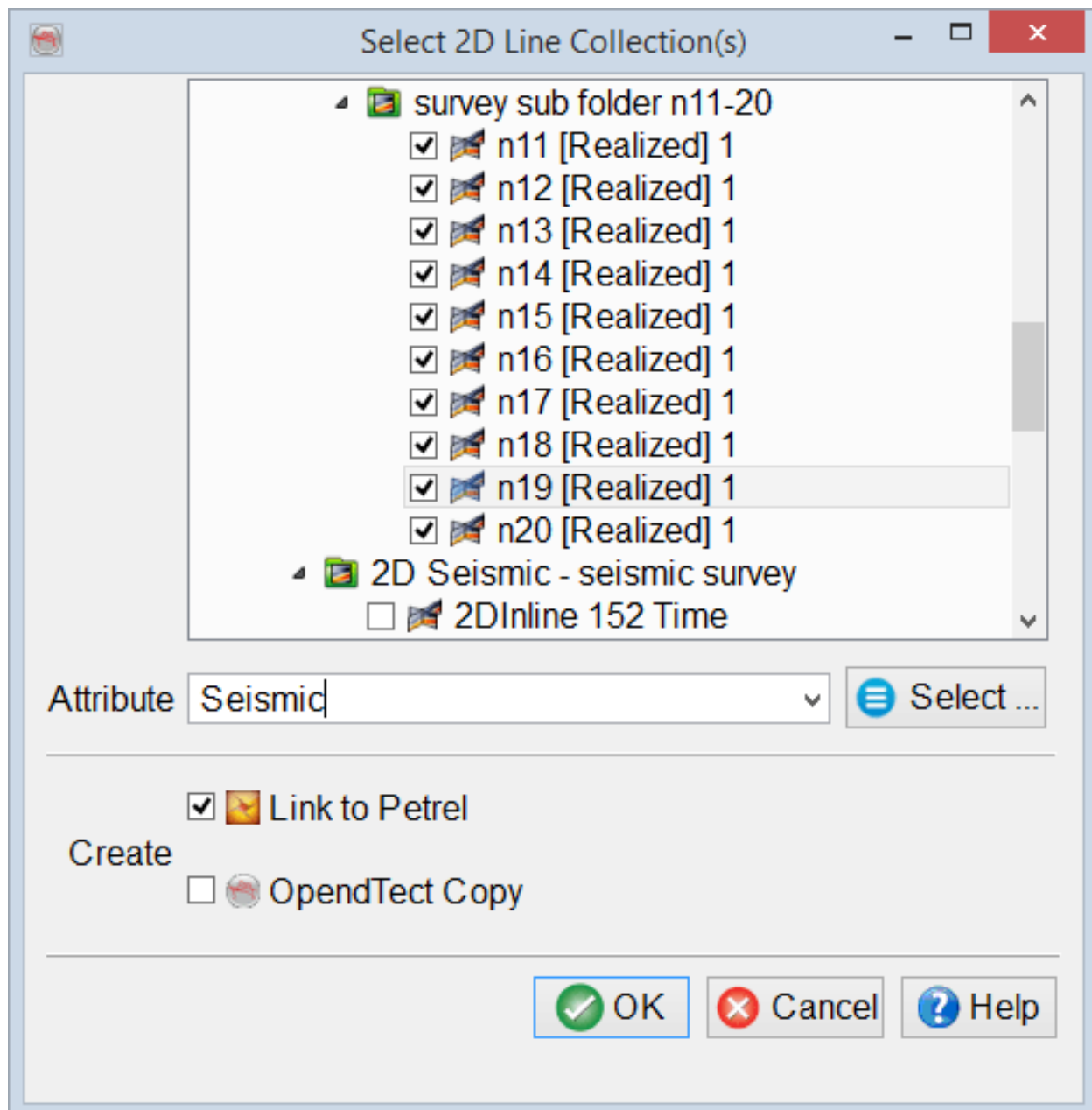
1.1.2.4.1 Import 2D Seismic From Petrel*


Petrel* 2D seismic can be accessed in OpendTect by clicking on PetrelDirect insert

icon  in the following places:

- *Manage 2D Seismics* window (*Survey > Manage > Seismics > 2D...* or click on  icon in the *Manage Data* toolbar)
- *Select 2D Line(s)* window when adding 2D Line(s) to the 3D scene by right click on *2D Line* in the tree > *Add*)
- other workflows, including attribute definition and processing, etc.

Select 2D Line Collection(s) window allows to select 2D seismic data from Petrel* project, provide a dataset name in the *Attribute* field and create either *Link to Petrel** or *OpendTect Copy*.





 *2D seismic data in OpendTect is organized in datasets and 2D line geometries. A dataset contains unique data for each 2D line, i.e. seismic data characterized by common processing history or a specific seismic attribute.*

*is a mark of Schlumberger.

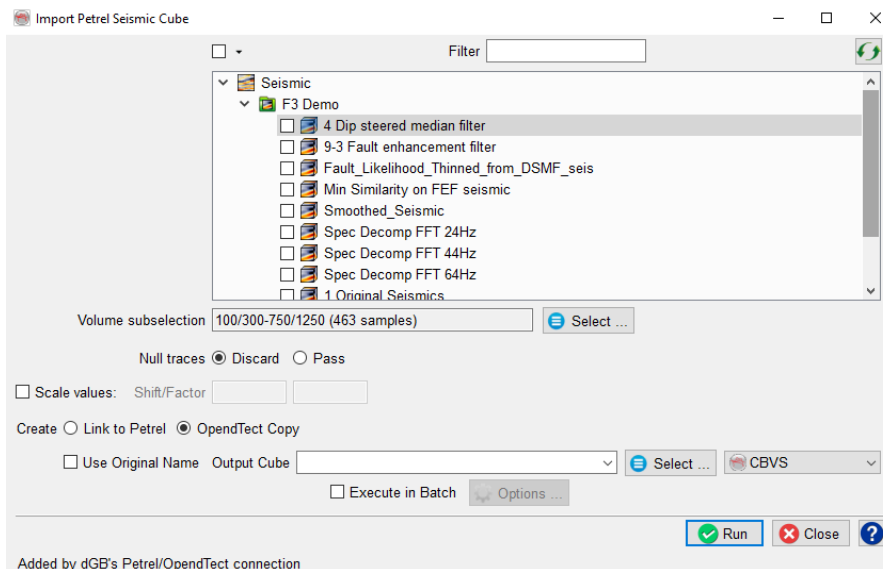
1.1.2.4.2 Import 3D Seismic From Petrel*

Petrel* 3D seismic cube can be accessed in OpendTect by clicking on PetrelDirect

insert icon  in the following places:

- *Manage 3D Seismics* window (*Survey > Manage > Seismics > 3D...* or click on  icon in the *Manage Data* toolbar)
- *Select* window when adding a seismic attribute display at In-line/Cross-line/Z-slice/3D Horizon in the 3D scene
- other workflows, including attribute definition and processing, etc.

Select Seismic Cube window allows to select 3D seismic data from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*. The tree view and the drop-down menus offer alternative ways to select data from a Petrel* project.



Seismic Cube Import User Interface




Petrel 3D seismic cube must have the same geometry as OpendTect survey set up. The easiest way to achieve this is to [Get survey set up from Petrel*](#) when setting up a survey and choose an appropriate Petrel* seismic survey folder as the geometry source.*

*is a mark of Schlumberger.

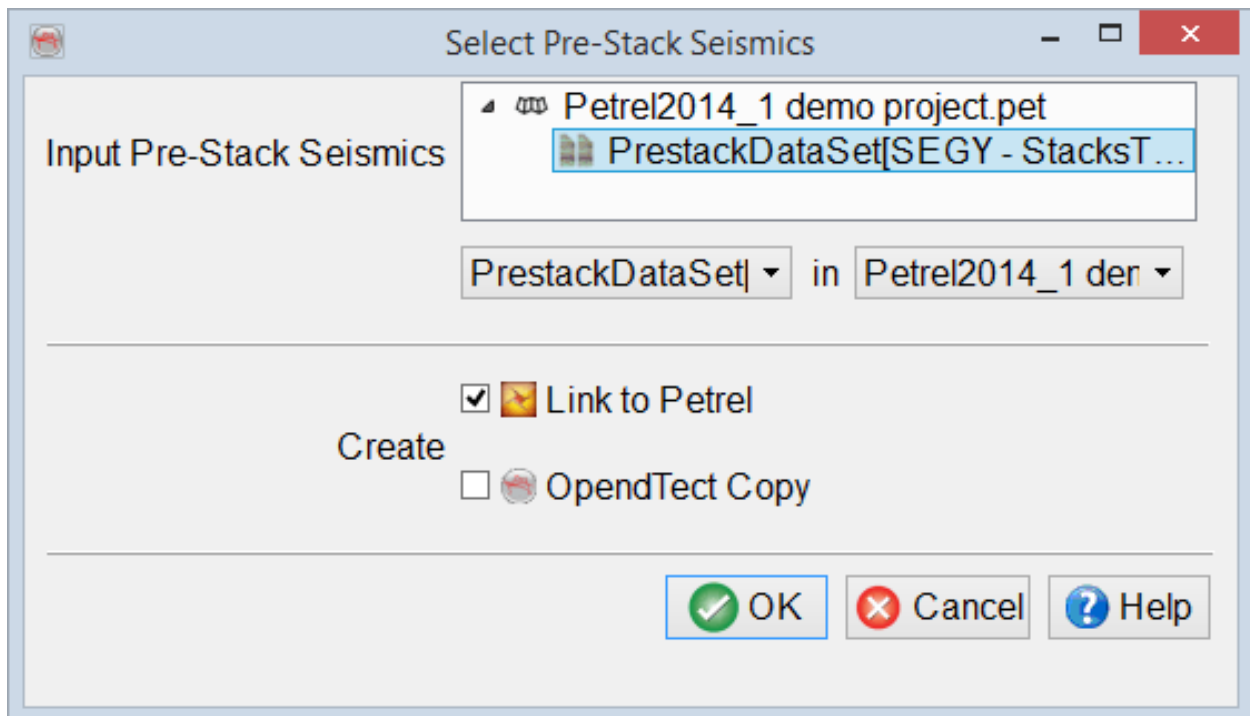
1.1.2.4.3 Import 3D prestack Seismic From Petrel*


Petrel 3D Prestack seismic dataset can be accessed in OpendTect by clicking on

PetrelDirect insert icon  in the following places:

- *Manage Prestack Data* window (*Survey > Manage > Seismics Prestack > 3D...* or click on  icon in the *Manage Data* toolbar)
- other workflows, including Prestack attribute definition and processing, etc.

Select Prestack Seismics window allows to select 3D prestack seismic data from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*. The tree view and the drop-down menus offer alternative ways to select data from a Petrel* project.



 *Petrel* 3D prestack data must have compatible geometry with OpendTect survey set up. The easiest way to achieve this is to [Get survey set up from](#)*



Petrel* when setting up a survey and choose an appropriate Petrel* seismic survey folder as the geometry source.

*is a mark of Schlumberger.

1.1.2.5 Import Wavelet From Petrel*

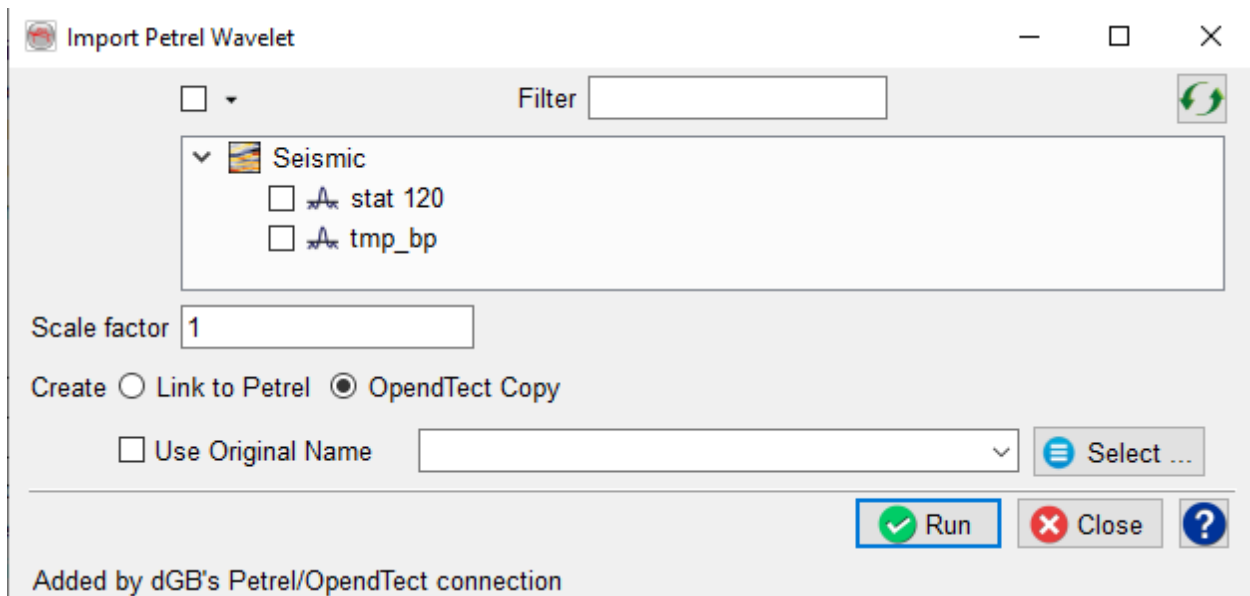
Petrel* wavelets can be accessed in OpendTect by clicking on PetrelDirect insert icon



in the following places:

- *Manage Wavelets* window (*Survey > Manage > Wavelets* or click on  icon in the *Manage Data* toolbar)


Select Wavelet(s) window allows to select wavelets from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.




Import Wavelet User Interface

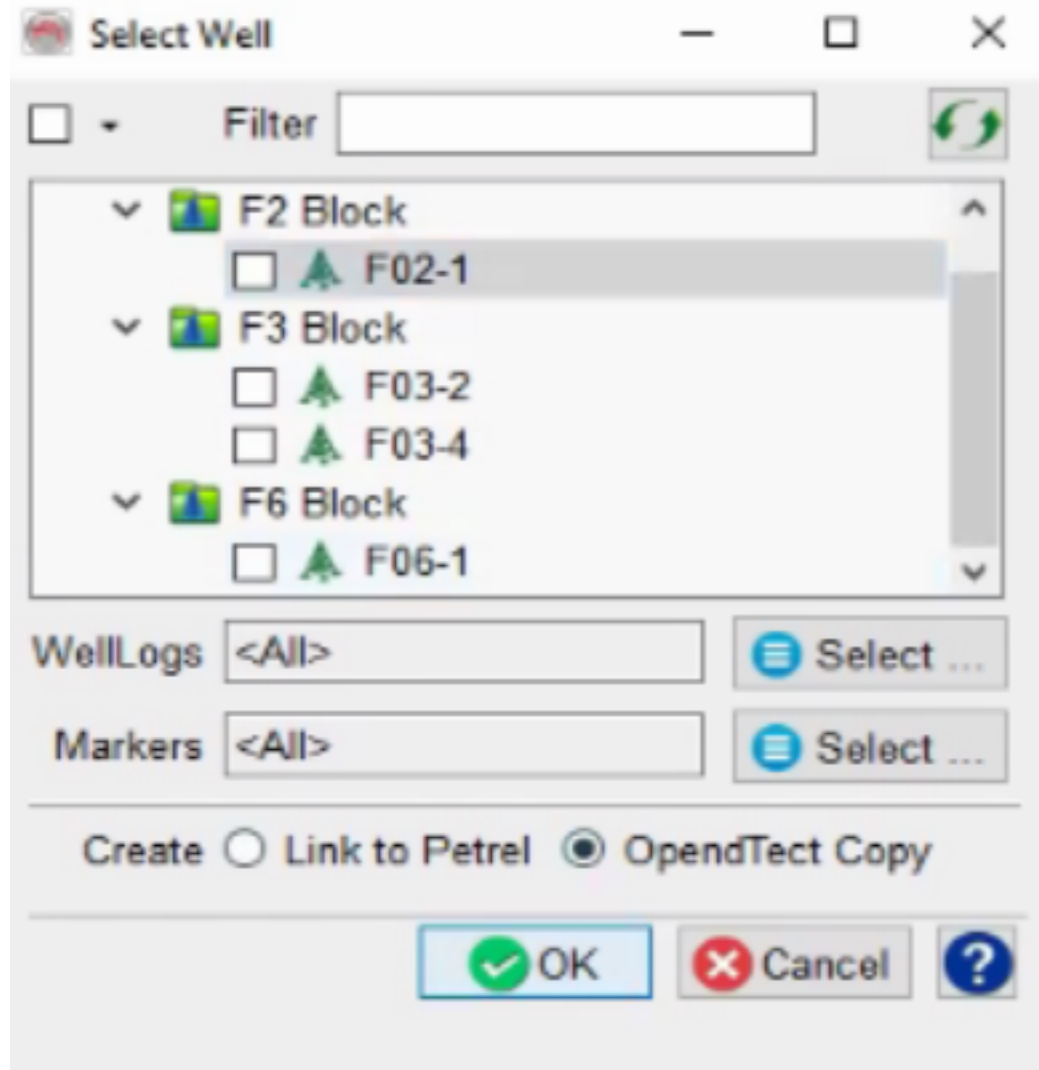
*is a mark of Schlumberger.

1.1.2.6 Import Well From Petrel*

Petrel well data (deviation survey, time-depth model, logs and well tops) can be accessed in OpendTect by clicking on PetrelDirect insert icon  in the following places:

- *Manage Well* window (*Survey > Manage > Wells...* or click on  icon in the *Manage Data* toolbar)
- *Load Well(s)* window when adding well(s) to the 3D scene or Basemap by right click on *Well* in the tree > *Add*)
- other workflows, including well-to-seismic tie, Log attribute definition and processing, etc.

Select Well(s) window allows to select wells from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*. Active time-depth model in a Petrel* project is always used.



Import Well User Interface

Selection of well logs and markers.

- By default all well logs and markers are selected.
- To change the default within the current import session: highlight one well in the tree and click *Select* buttons to make a selection of well logs and markers which is automatically applied to all the wells.


Use Ctrl+A and Ctrl+Shift+A shortcuts to select/de-select all items in the list.

*is a mark of Schlumberger.

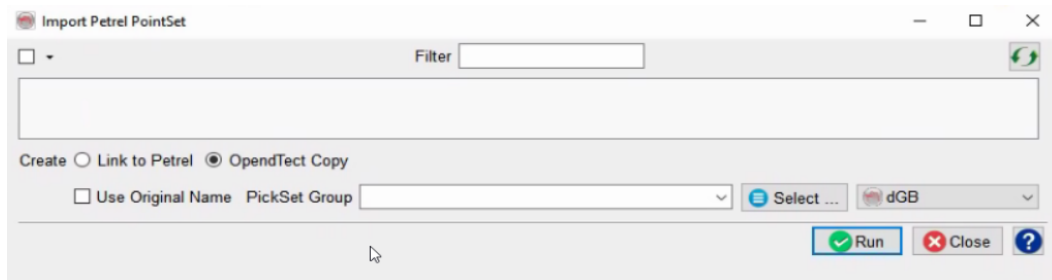
1.1.2.7 Import Pointsets and Polygons From Petrel*

Petrel pointsets and polygons can be accessed in OpendTect by clicking on PetrelDir-

ect insert icon  in the following places:

- *Manage Pointset/Polygons Windows (Survey > Manage > Pointsets/Polygons...* or click on the  icon in the *Manage Data* toolbar). When adding from PointSet/Polygon manager, user will have a selection box to choose which kind of object they want to import from Petrel*
- *Load Pointsets (Polygons) window* when adding pointsets (polygons) to the 3D scene or Basemap by right clicking on *Pointsets (Polygons)* in the tree > *Add*). The relevant dialog box pops up.

Import Pointsets (Polygons) window allows to select Pointsets (Polygons) from Petrel* project and create either *Link to Petrel** or *OpendTect Copy*.



Selection of well pointsets (polygons)

By default all pointsets (polygons) are selected.

To change the default within the current import session: highlight one polygon (point-set) in the tree and click Select buttons.

Use Ctrl+A and Ctrl+Shift+A shortcuts to select/de-select all items in the list.

*is a mark of Schlumberger.

1.1.2.8 Import All Data From Petrel* at Survey Creation

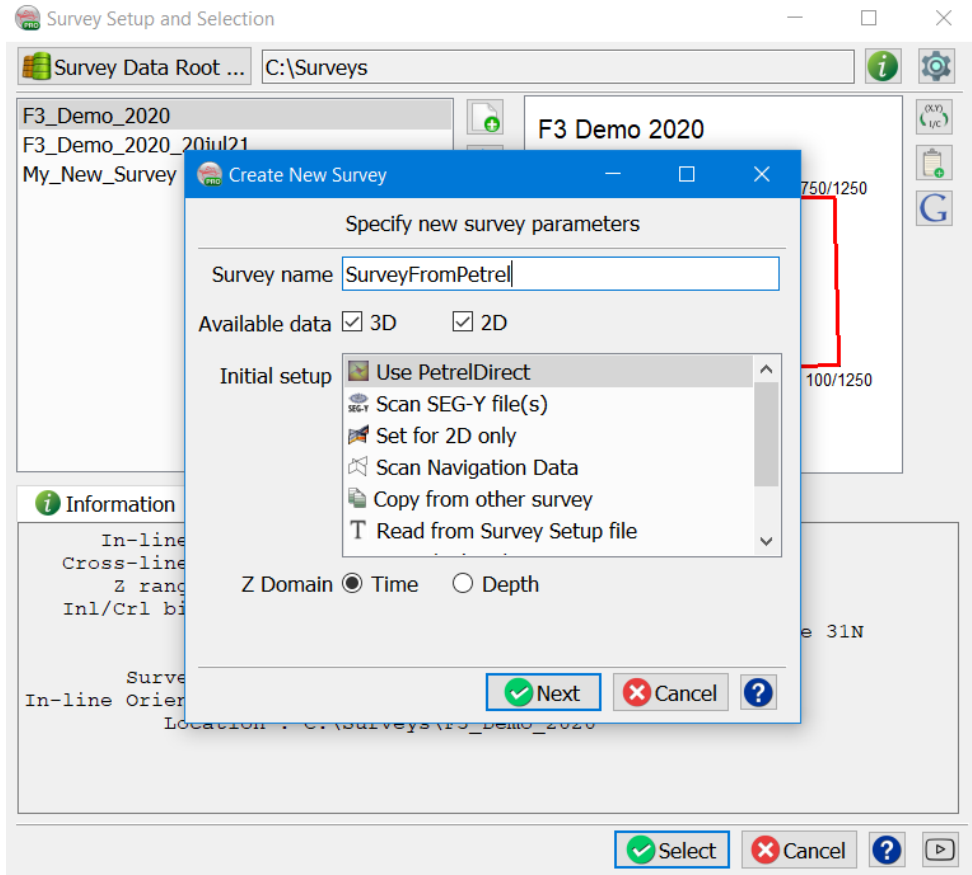
This OpendTect Pro feature facilitates the import of an entire project from Petrel* to OpendTect during the survey creation phase.

All data objects are supported (including but not limited to: *survey parameters, seismic 2D/3D, pre-stack seismic, horizon 2D/3D, fault, pointset, polygon, wavelet and wells*) and can be copied in a single click of a button.

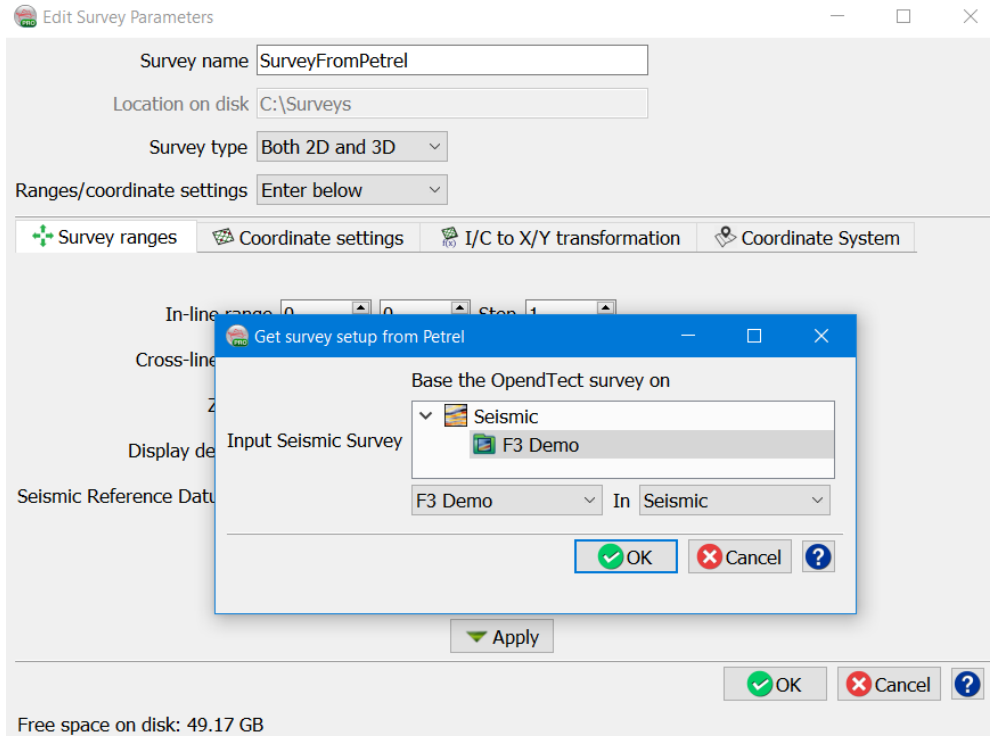
This enables the user to do his or her seismic interpretation work in OpendTect Pro and push these interpretations back into Petrel* for subsequent modeling work.

The initial steps in this workflow are similar to the workflow described in [section 1.1.1 Survey Setup From Petrel*](#).

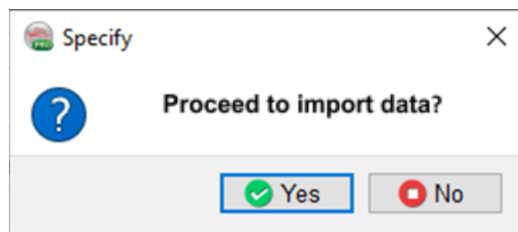
Firstly, specify the creation of a new survey using the *PetrelDirect* option in the initial setup:



Then select the Petrel* seismic survey to be used for creating the new OpendText project:



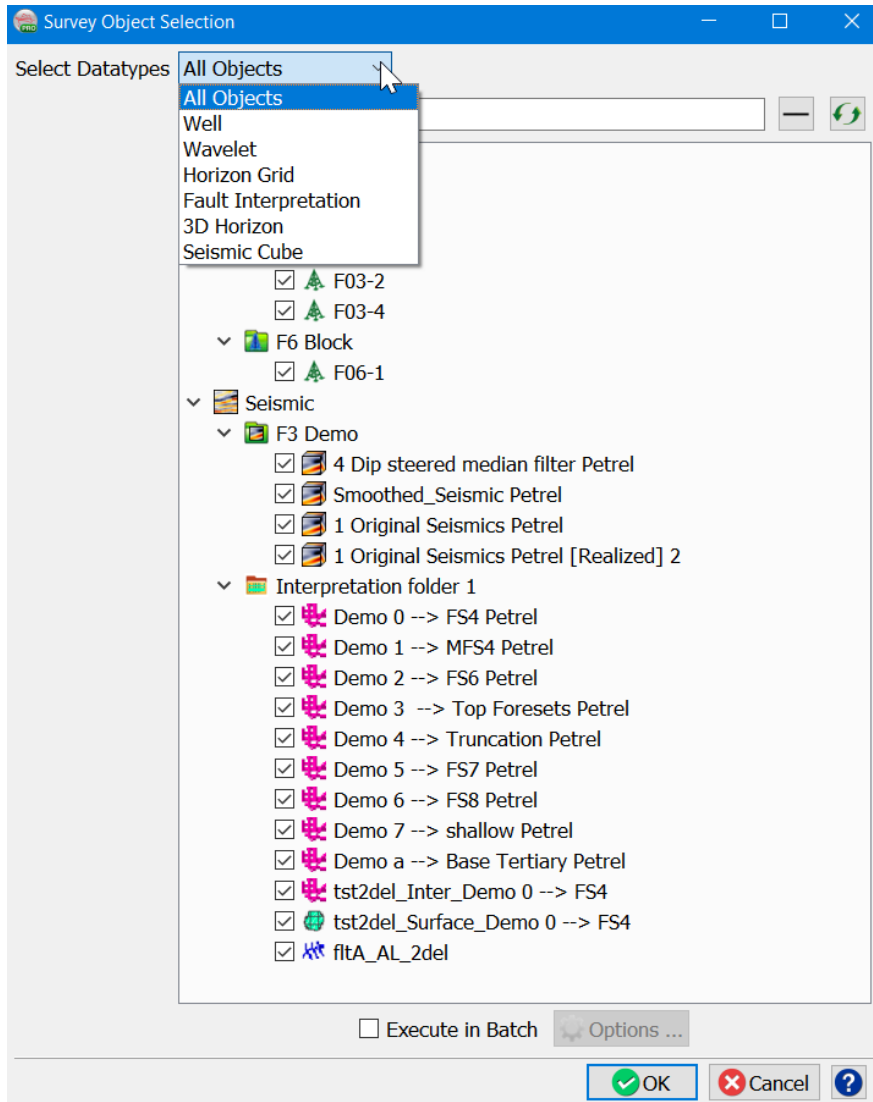
Upon clicking **OK** in the *Edit Survey Parameters* dialog box, a message asks if we would like to import data or not:



If the user wants to proceed with importing data at this time (pressing **Yes**), then the *Object Selection* dialog box is shown.

The dialog box depicts the tree as seen on Petrel* side with identical icons.

The tree can be filtered in two ways, either on the basis of name of the object or on the basis of object type, or in combination:



The *Datatypes* list is populated with the data types available in the selected Petrel* project that are supported by OpendTect.


Note: All OpendTect data types are now supported - Seismic 2D/3D, Pre-stack Seismic, Horizon 2D/3D, Faults, FaultStickSets, PointSets, Polygons, Wavelets and Wells

Upon selecting the required objects, they will be imported and saved in the newly-created OpendTect project.

If a large amount of data needs to be transferred to OpendTect, it is recommended to use the *Batch* option to speed up the processing.

*is a mark of Schlumberger.

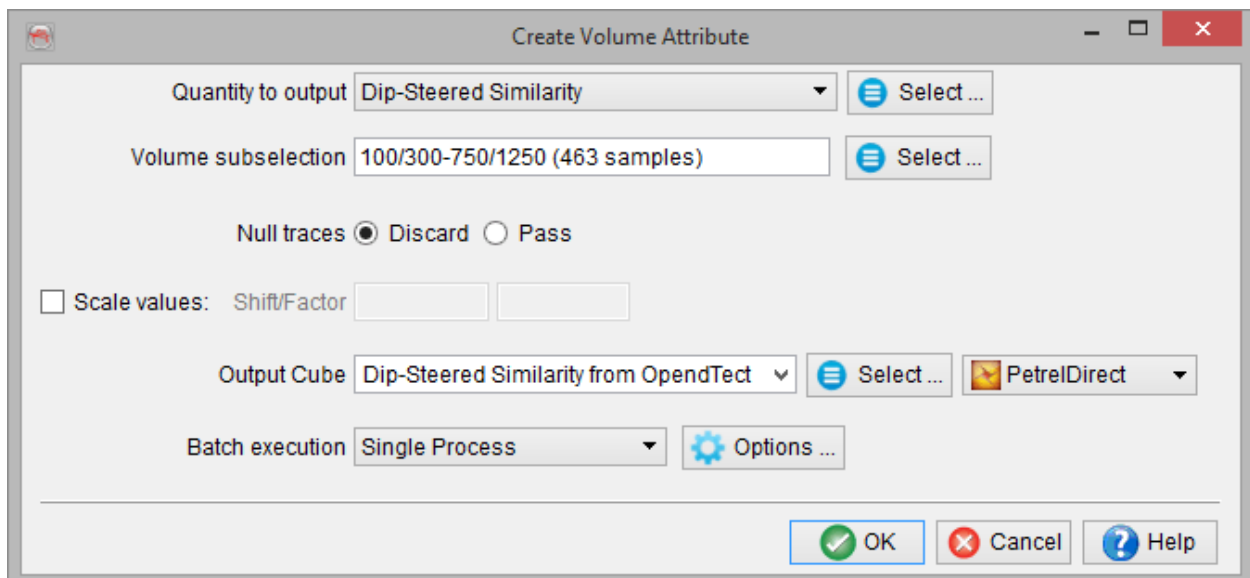
1.1.3 Export To Petrel*

Writing directly to Petrel* data store is available in various OpendTect workflows by choosing  *PetrelDirect* output format.

The following data can be written directly to Petrel* data store:

- FaultStickSets
- 3D horizons and horizon attributes
- 2D and 3D seismic data
- Wavelets

The example below shows processing the dip-steered similarity attribute to be output directly to Petrel* project.



Click on Select button to choose an output Petrel folder, otherwise the data is written to the first appropriate folder.*


*is a mark of Schlumberger.

1.1.3.1 Export Fault To Petrel*

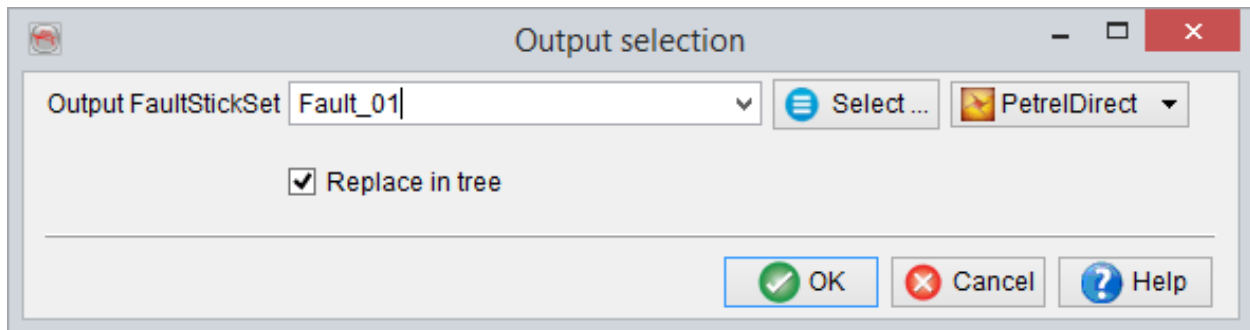
Currently only [FaultStickSets](#) can be written to Petrel* data store as Fault Interpretation object.


*is a mark of Schlumberger.

1.1.3.2 Export FaultStickSet To Petrel*

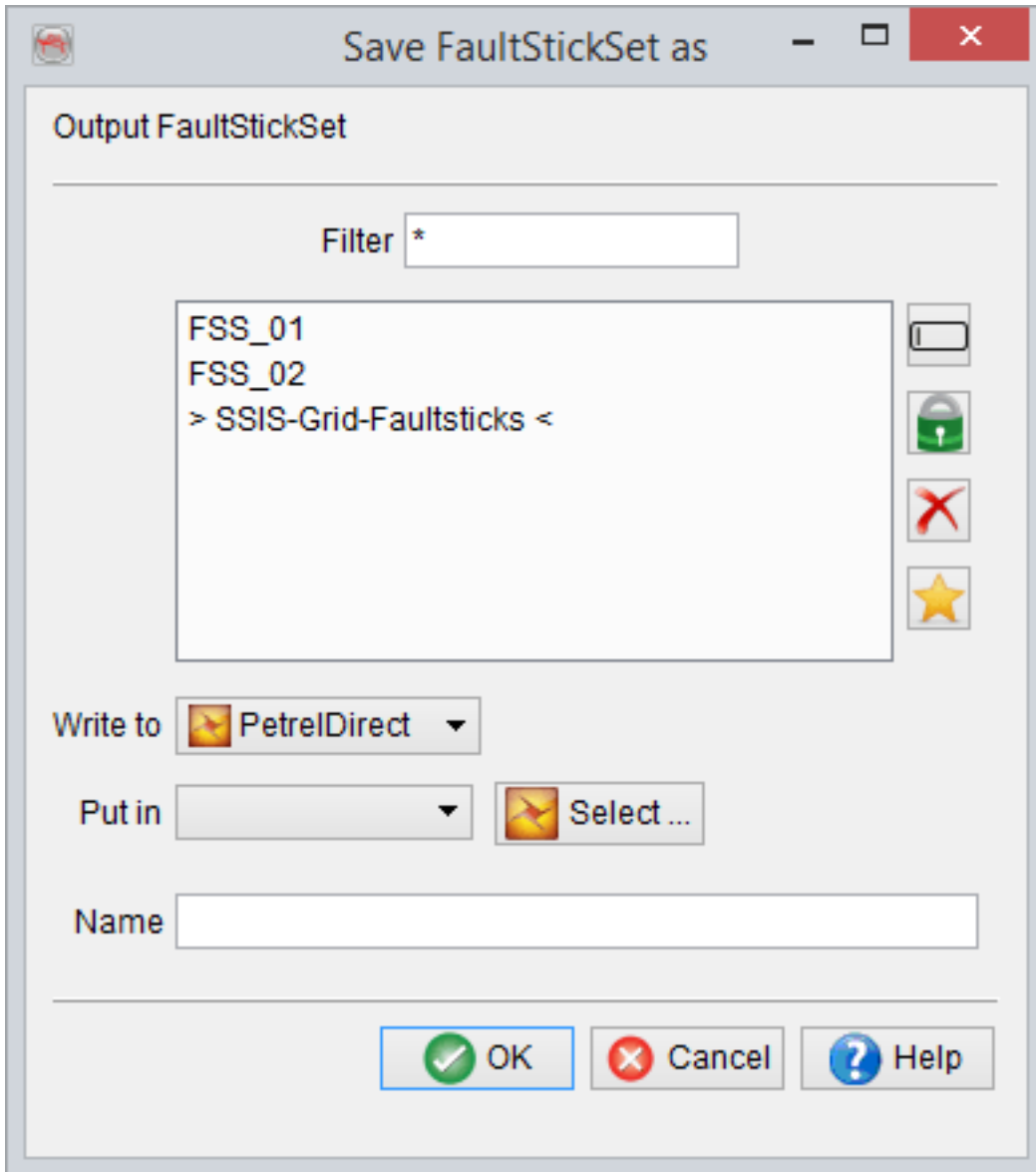
OpenTect FaultStickSets can be written directly to Petrel* data store as Fault Interpretation object by choosing  *PetrelDirect* output format in the following places:

- Right-click on a fault stick set in the Tree of 3D scene or 2D viewer and choose *Save as*:




-  *Click on Select button to choose an output Petrel* folder, otherwise the data is written to the first appropriate folder.*

- *Save FaultStickSet as window:*




*is a mark of Schlumberger.

1.1.3.3 Export Horizon To Petrel*

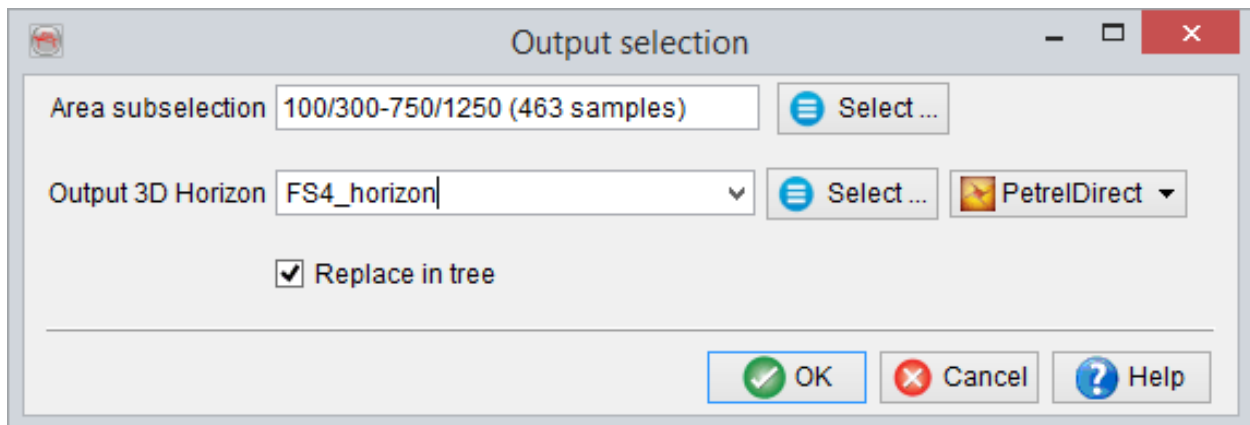
Writing seismic horizons directly to Petrel* data store is available in various OpendTect windows by choosing  *PetrelDirect* output format. Currently only [3D horizons](#) are supported.

*is a mark of Schlumberger.

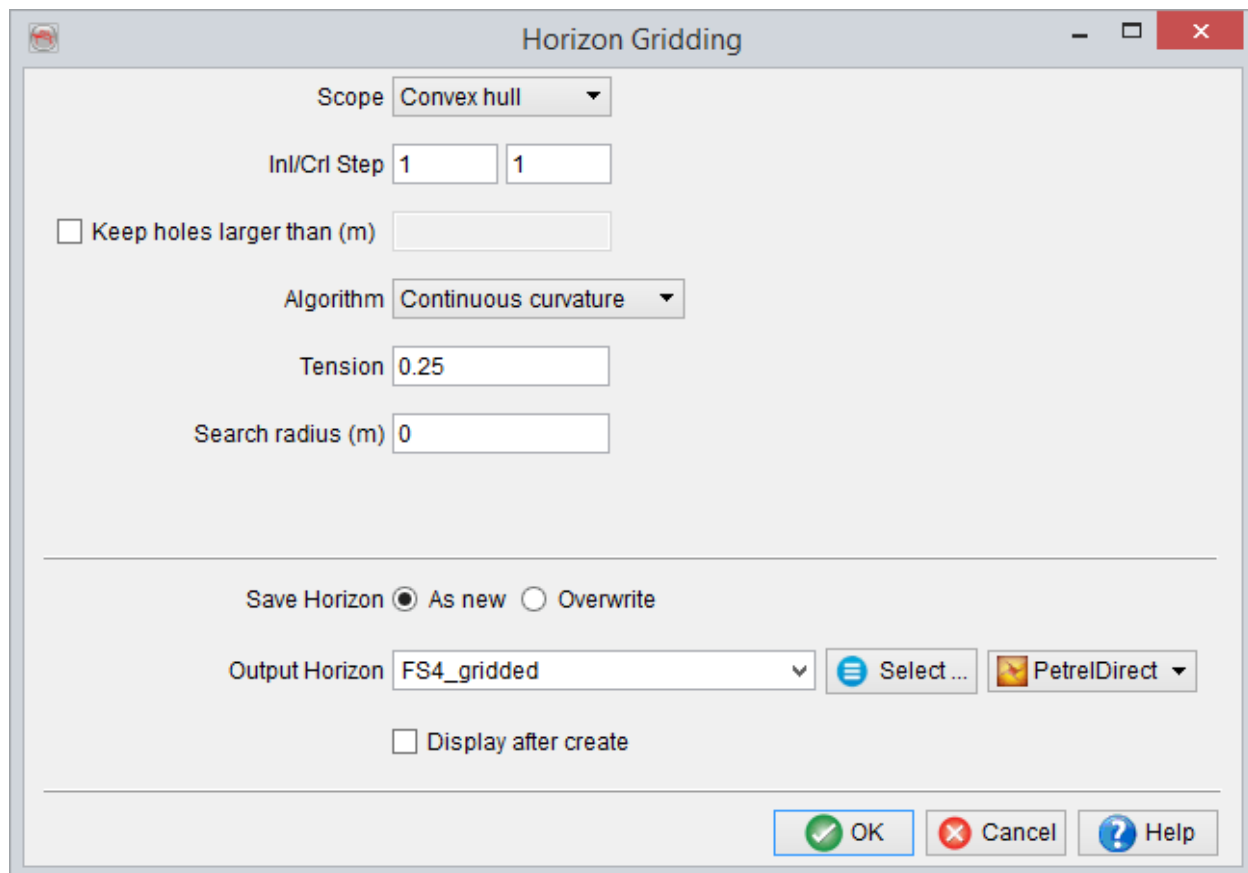
1.1.3.3.1 Export 3D Horizon To Petrel*


Writing 3D horizons directly to Petrel* data store is available in various OpendTect windows by choosing  *PetrelDirect* output format:

- Right-click on a 3D horizon in the Tree of 3D scene or 2D viewer and choose *Save as*:

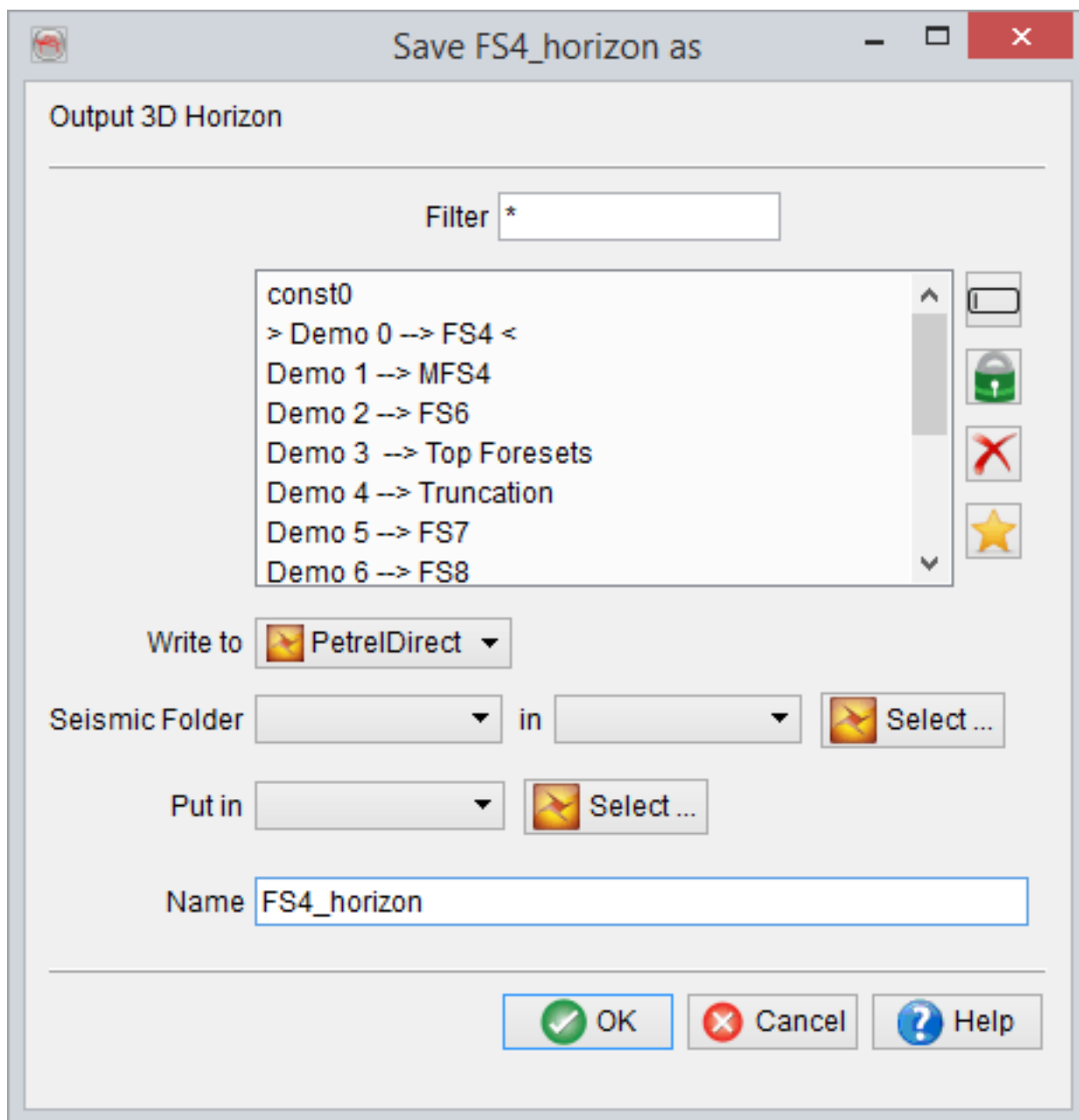


- In various workflows, such as gridding, filtering, etc.:

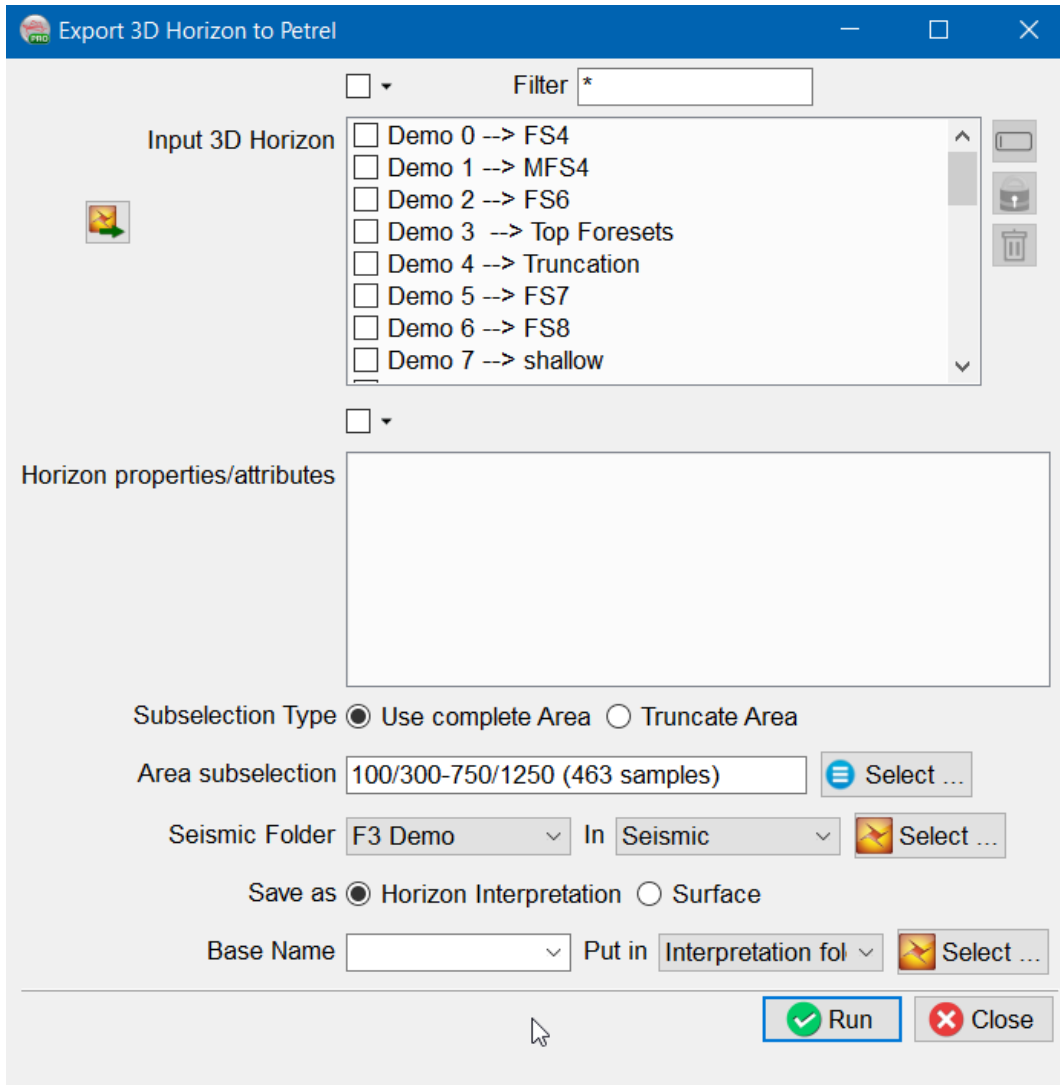


 *Click on Select button to choose an output Petrel* folder, otherwise the data is written to the first appropriate folder.*

- *Save 3D horizon as window:*



- Under Survey, Export Horizons with PetrelDirect:



On the Petrel side, multiple seismic surveys are supported.

Save as Horizon Interpretation or Surface: It is a new option in Version 6.6.0 that allows to export a surface as Horizon Interpretation or as Surface. A Surface is not bounded by seismic survey dimensions, it can exist across multiple seismic surveys. Whereas, Horizon Interpretation can only exist within a seismic survey.

OpendTect is created on the basis of single survey dimensions, hence no distinction between Surface and Horizon Interpretation. When a horizon is imported to OpendTect, it is imported as surface (i.e. even if data is outside survey ranges it will be displayed). But when exporting to Petrel, the user needs to make a clear distinction between Surface and Horizon Interpretation.


Use Complete Area: This option as per its name does not truncate the area. In case the surface is greater than the seismic cube, a warning message will pop up stating these facts. If a user still wants to transfer such surfaces as Horizon Interpretation, it will be truncated according to survey dimension.

Truncate Area: This allows subselection of area.

Base Name: This allows a common string to be added before the name of the object being exported. If kept blank original name of object will be used to create an object on Petrel* side. This option is available in all export Petrel* dialog boxes.


*is a mark of Schlumberger.

1.1.3.4 Export Seismic* To Petrel

Writing 2D and 3D seismic data directly to Petrel* data store is available in various OpenTect windows by choosing  *PetrelDirect* output format.

*is a mark of Schlumberger.

1.1.3.4.1 Export 2D Seismic To Petrel*

Writing 2D seismic data directly to Petrel* data store is available in various OpendTect workflows by choosing  *PetrelDirect* output format:


- [processing 2D seismic attributes](#);
- [copying 2D seismic data](#);
- [creating 2D grid](#) and [extracting 2D data](#) from 3D seismic




Click on Select button to choose an output Petrel folder, otherwise the data is written to the first appropriate folder.*

*is a mark of Schlumberger.

1.1.3.4.2 Export 3D Seismic To Petrel*


Writing 3D seismic data directly to Petrel* data store is available in various OpendTect windows by choosing  *PetrelDirect* output format:

- processing [3D seismic attributes](#), [Volume builder output](#), [Bayesian Classification](#), etc.;
- operations on 3D seismic volumes such as [copying](#) and [merging](#).

 *Click on Select button to choose an output Petrel* folder, otherwise the data is written to the first appropriate folder.*

*is a mark of Schlumberger.

1.1.3.5 Export Wavelet To Petrel*


Writing wavelets directly to Petrel* data store is available in various OpendTect windows by choosing  *PetrelDirect* output format.



Click on Select button to choose an output Petrel folder, otherwise the data is written to the first appropriate folder.*

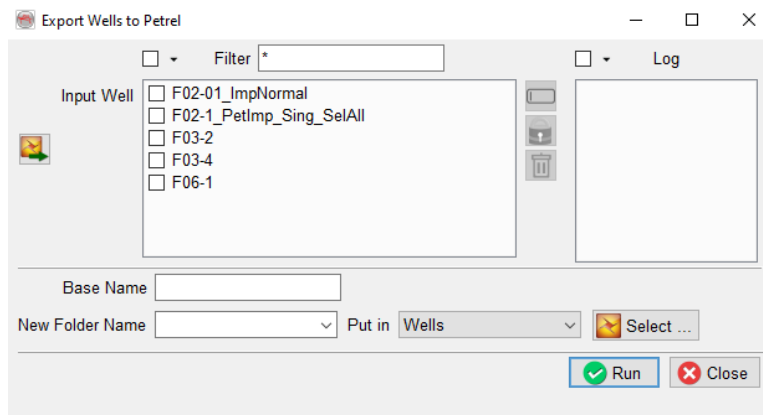
*is a mark of Schlumberger.

1.1.3.6 Export Well to Petrel*

Petrel well data (deviation survey, time-depth model and logs) can be accessed in OpendTect by clicking on PetrelDirect insert icon  in the following place:

- *Export Well* window (*Survey > Export > Wells > PetrelDirect**)

The *Select Well(s)* window allows to select wells from OpendTect project and create a copy in Petrel*.



Well Export User Interface


Multiple exports are supported. Different selection of wells and logs can be made.

Use Ctrl+A and Ctrl+Shift+A shortcuts to select/de-select all items in the list.

*is a mark of Schlumberger.

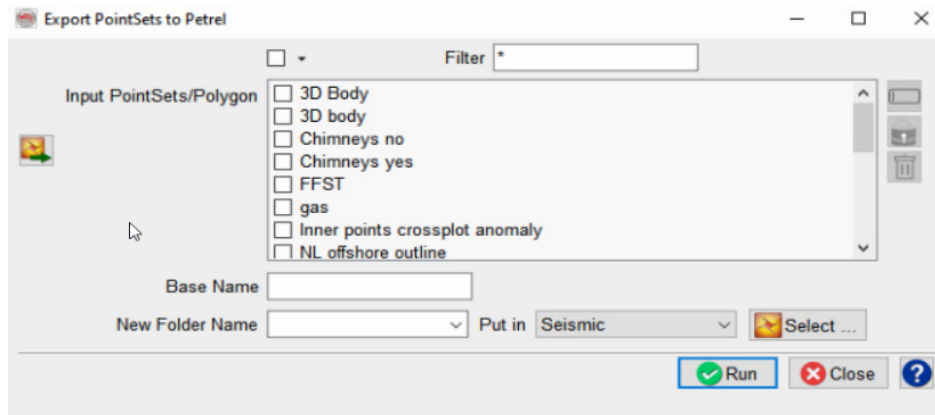
1.1.3.7 Export Pointsets and Polygons To Petrel*

Petrel pointsets (polygons) can be accessed in OpendTect by clicking on PetrelDirect

insert icon  in the following place:

- *Export Pointsets (Polygons) window (Survey > Export > Pointsets/Polygons> PetrelDirect)*

Export Pointsets (Polygons) window allows to select pointsets (polygons) from OpendTect project and create a copy in Petrel*.



Pointsets Export User Interface

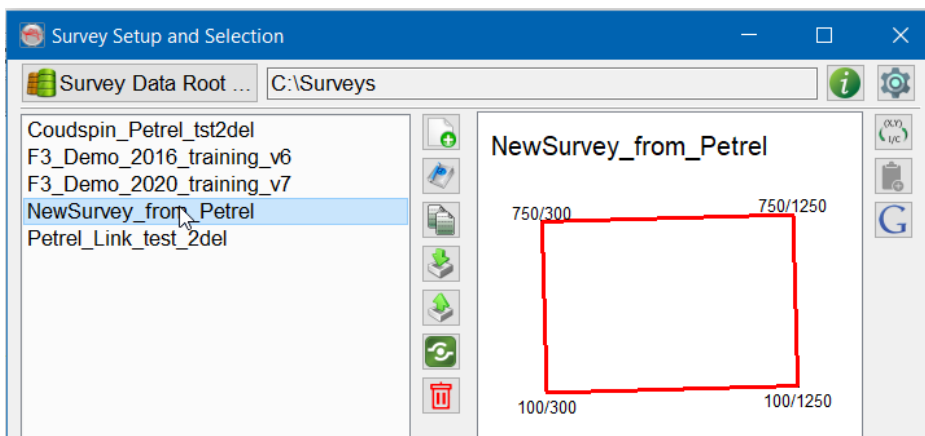
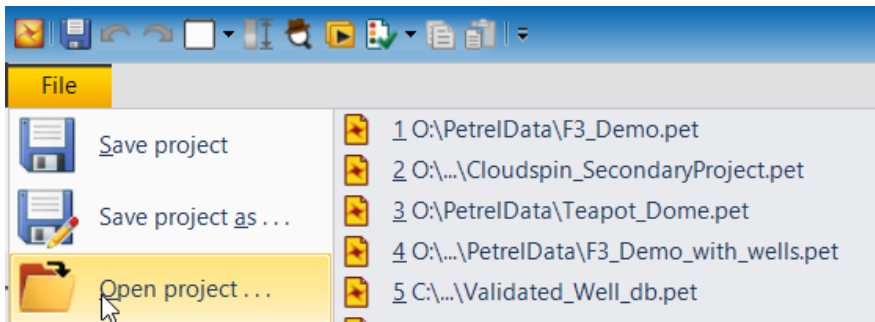
Multiple exports are supported. Different selection of pointsets (polygons) can be made.


Use Ctrl+A and Ctrl+Shift+A shortcuts to select/de-select all items in the list.

*is a mark of Schlumberger.

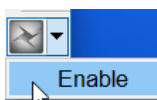
1.1.4 Link OpendTect and Petrel* projects

1. Open both Petrel and OpendTect applications, *Select* the project/survey you want to transfer data to/from:



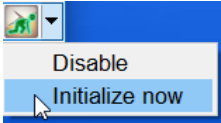
2. At the start, the Petrel connection icon  is disabled (in the lower right corner of the main OpendTect window).


3. Enable the Petrel connection (drop-down menu next to the PetrelDirect icon):



When the connection is enabled, the PetrelDirect status button becomes active: 

4. *Initialize* Petrel connection (drop-down menu next to the PetrelDirect icon):



When the connection is initialized, the PetrelDirect status button indicates that the connection has been established between Petrel and OpendTect projects, thus: 

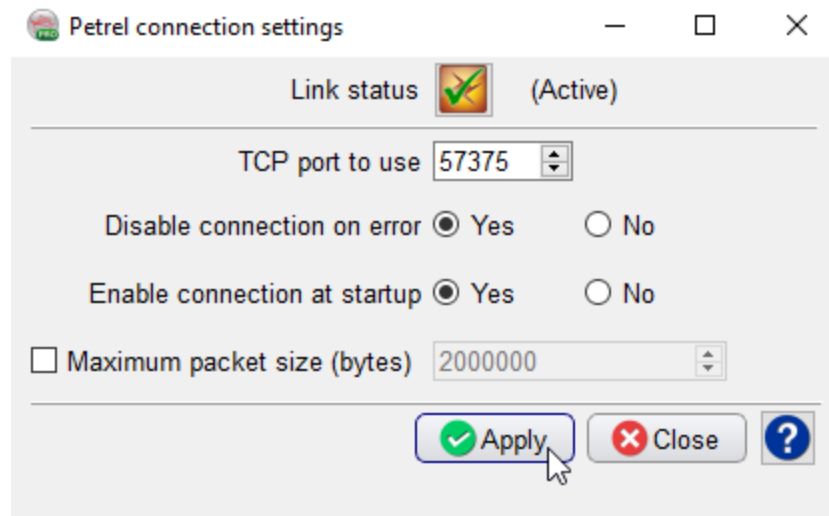
Note: If the connection is enabled and left uninitialized, the PetrelDirect status will become active once Petrel is used.

*is a mark of Schlumberger.

1.1.5 Settings Window


*Petrel** connection settings can be set/modified by clicking on PetrelDirect status button


 (located in the lower right corner of the main OpenText window).



Link status: the icon shows the status of connection to *Petrel** and can be clicked on to change it:

 - disabled (click on it to *Enable* connection)

 - uninitialized (connection is enabled, i.e. will be activated once *PetrelDirect* is used; click on it to *Initialize now* or use drop-down menu next to it to *Disable* connection)

 - active (click on it to *Disable* connection)


TCP port to use: TCP port number must be the same as the port number specified on the Petrel* side. The default value of 57375 should work in most cases.



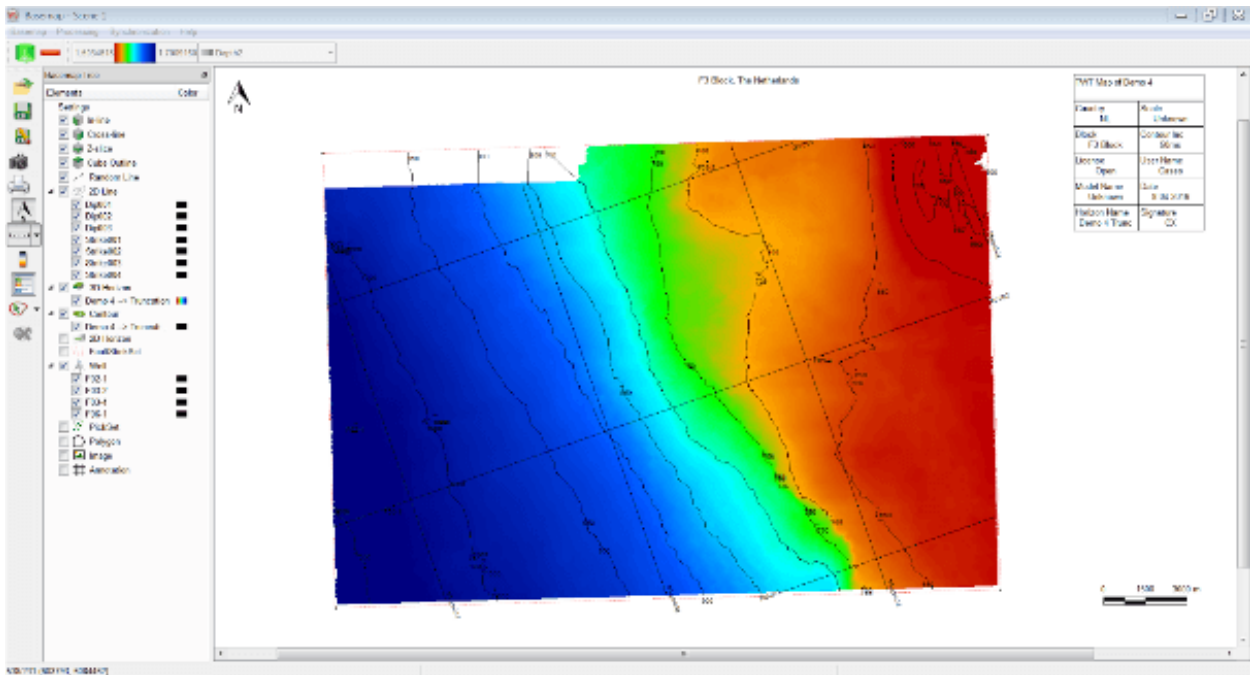
The default value of TCP port can be changed using DTECT_PETREL_PORT environment variable. In doubt consult your system administrator.

*is a mark of Schlumberger.

1.2 Basemap

OpenTect's basemap is accessible from OpenTect's main menu (View > Basemap) or this shortcut icon (). A separate detached window is launched, which is by default synchronized with the 3D scene. We recommend turning such synchronization off if the data size (e.g. area or number of 2D seismic line) is huge.

Basemap comes with its own tree allowing an independent control of data displays. An overview of the basemap window is given below:

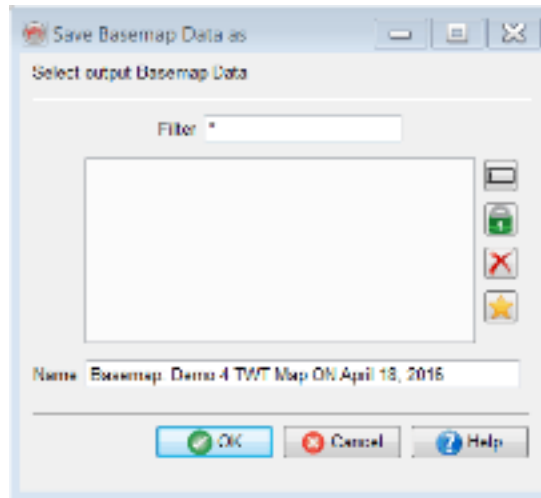


Main Menu

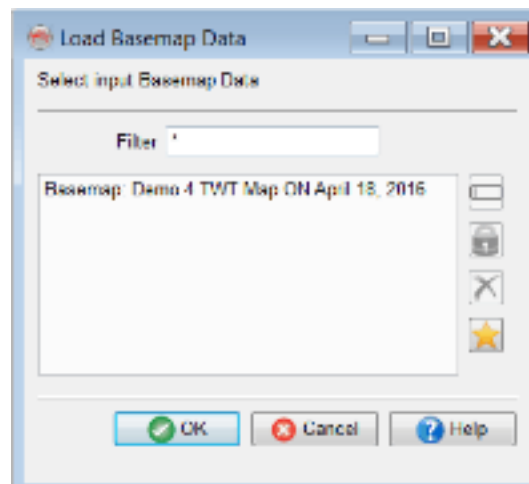
Basemap

This includes:

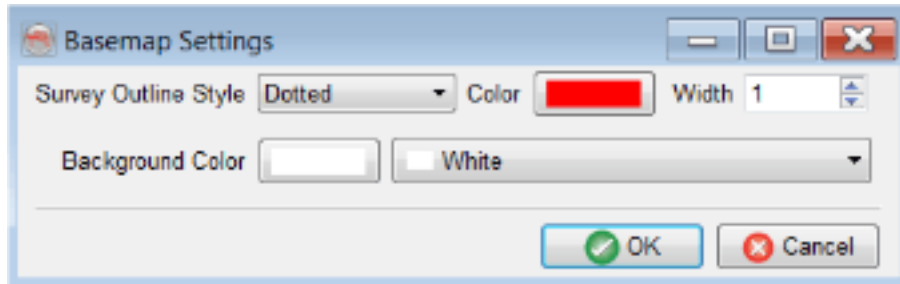
Save (📁) or Save As (✎) are used to save the current displays as a map for future use. We highly recommend saving a basemap on a routine bases. Multiple copies of the map are also allowed.



A previously stored basemap can be retrieved using the Open (📁) icon. You can set one basemap as a favourite /default item (★). If set, the selected basemap will be opened every time a new OpendTect is started.

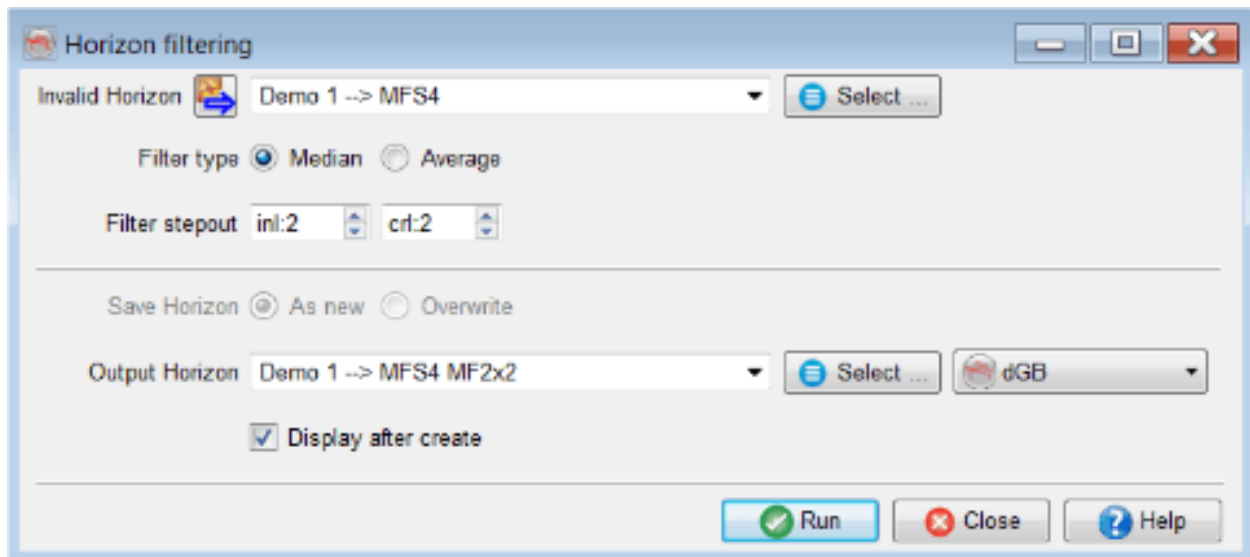


The settings sub-menu is used to change the outline style (line pattern, thickness, and color) of the current basemap. The default background color is white, which can also be changed, if needed be.



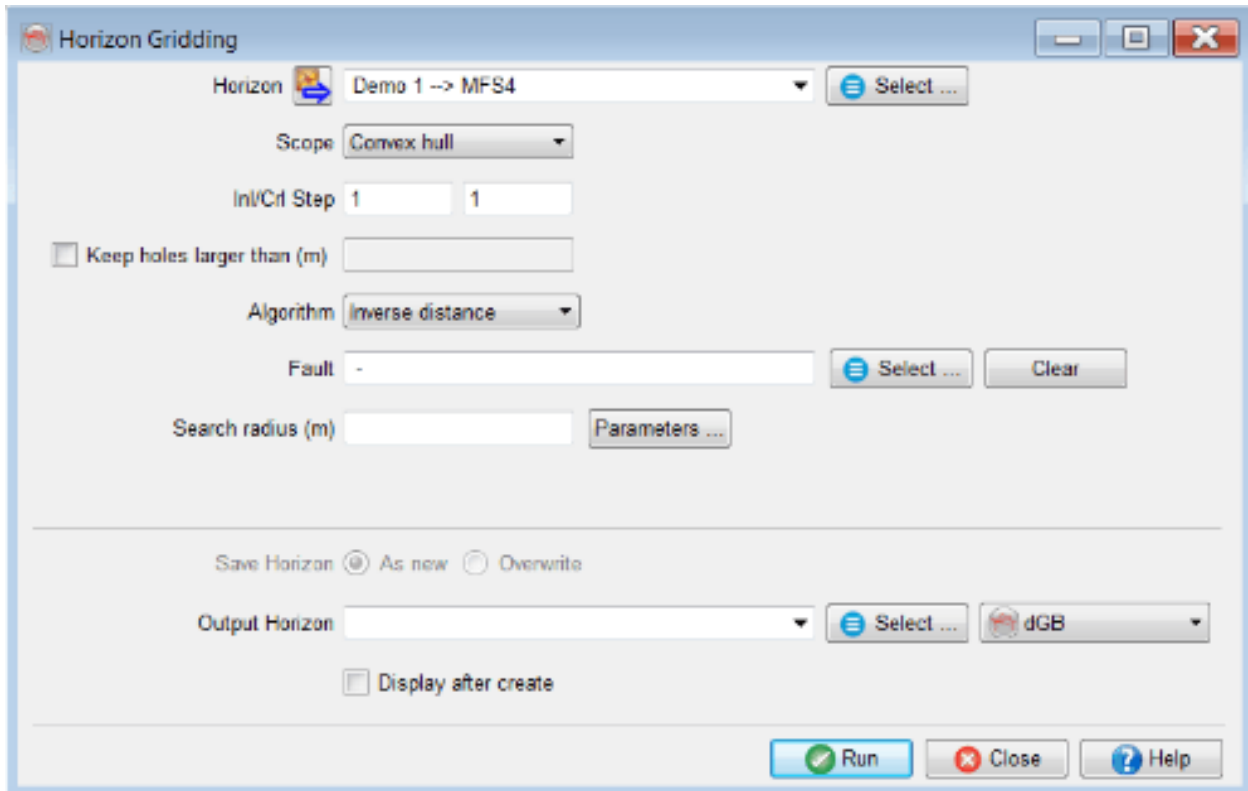
Processing

Filtering: A selected horizon grid can be filtered within the basemap. Median filtered is preferred if you intend to de-spike a horizon grid. Average filter is considered as a general smoothing filter. If you intend to display the resultant grid then check the *Display after create* option.



Gridding: OpendTect supports several gridding algorithms (For details refer to [3.7.3.1 Grid](#) in OpendTect User Documentation). The supported parameters are summarized below:

- **Scope:** This is used to control the exterior geometry of the output horizon. Convex hull is mostly used, which automatically creates polygon (hull) using the exterior data points of a given horizon and creates a horizon within this polygon.
- **Inl/Crl Steps:** This option defines the output grid spacing. Often the seismic surveys are defined with an average spacing of 12.5m or 25m as a step. For larger surveys, we recommend under-sampling the output grid to 2 by 2 or even 5 by 5.
- **Keep holes larger than (m):** This option is used to keep larger holes in the survey. The gridding algorithm will not interpolate the areas larger than the set size.
- **Algorithm:** Several gridding algorithms are supported (For details refer to [3.7.3.1 Grid](#) in OpendTect User Documentation). Some of them use the fault e.g. dip steered gridding algorithm (requires 3D seismic).
- **Search radius (m):** It is mostly a best practice to include a part of data points within the search radius. For instance, if 2D lines are evenly spaced at 2000m and most of the lines have been interpreted, then the search radius should approximately be 2000m or larger.




Synchronization

Data displays can be handled from the basemap. This is done if the synchronization is turned off/on for specific items. At present, the synchronization is set globally between the basemap, 2D Viewers and 3D Scene.

If it is turned ON, then the data will be displayed in 3D as well as in 2D Viewers - if launched.

Contrarily, if the sync is turned OFF, you can display on an element from the basemap to a 3D scene / a 2D Viewer. This is done via the right-click menu.

 For larger 2D surveys, it is recommended to turn off the synchronization between the 3D Scene and the basemap. This will avoid memory consumption issues and instabilities in large data displays.

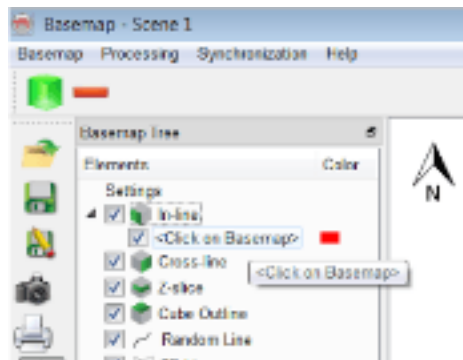
Help

It is a standard help button pointing to this documentation.

Trees and Elements

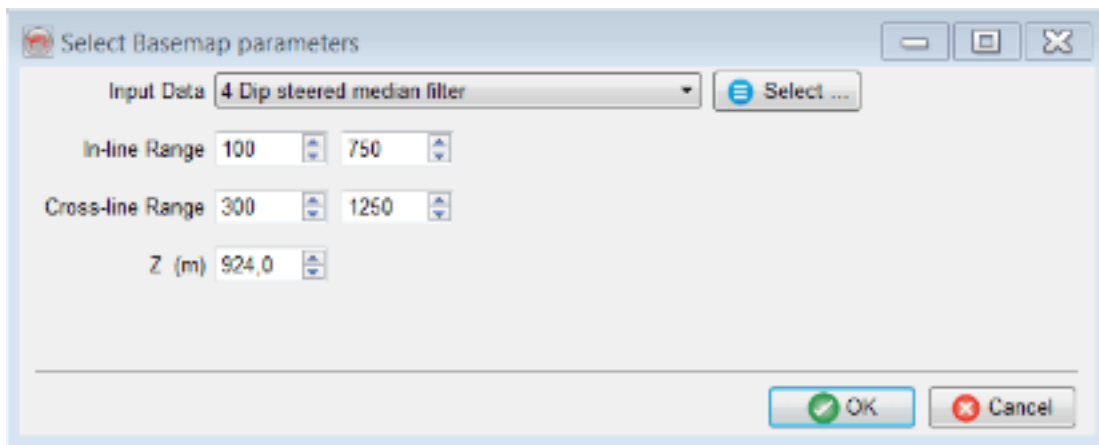
Inline, Crossline and Z-slices

These elements are added in the tree through a right-click menu. The position of the element (Inline and Crossline) is selected by clicking on the basemap. If the synchronization (via the Main Menu) between the 3D scene and basemap is turned ON, the inserted inline will be visible in the 3D scene. Dragging of the elements will not only change the position of the lines but also interactively update the 3D scene if the data is preloaded.



This is an example screenshot of adding an Inline element in the basemap. Note the label along with the tool tip for guide. The same implies for the Crossline.

If a Z-slice is added, it will show the following dialog to sub-select the area of the z-slice. By default, the values will be the survey range.



It adds z-slice in the basemap with a given range.

Z-slice also has a right-click menu (tree > z-slice > right-click) containing several options. The properties option will show the positioning dialog to reposition the slice. One can also show the z-slice in a 2D Viewer or in a 3D scene via the right-click menu.

If you intend to add a z-slice in the basemap, we will recommend preloading the data as the seismic data visualization along these slices could be slower.

Alternatively, you can also add inline at a mouse position. This is done via the right-click menu:



Cube Outline

This element displays an outline polygon for a given cube.

Random Line

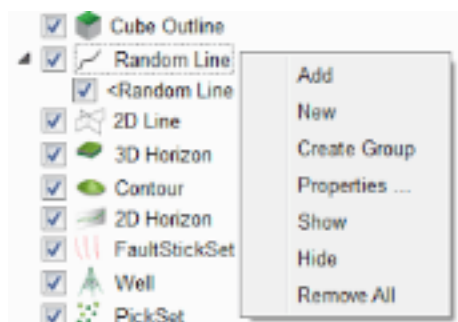
New Lines

In the basemap, random lines can be created as a group or individual sub-elements. Once an individual random line is added in the tree, a user has to click on the basemap to draw its geometry. Once finished, double click to confirm the geometry. The geometry can be manipulated by dragging the line nodes on the basemap. If the data is pre-loaded, the random line will be interactively displayed in the 3D scene.

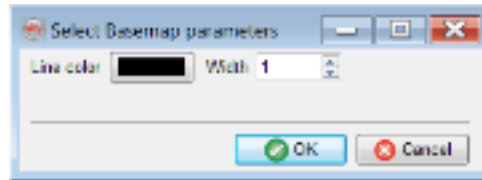
Furthermore, one can drag/rotate the line in the 3D scene. Its geometry will be updated in the basemap and vice versa.

Random lines can also be launched in a 2D Viewer to perform interpretation (within the basemap, right-click on the line and choose 2D Viewer).

Like other elements, random line also has its own right-click menu giving a flexibility to manage the element.



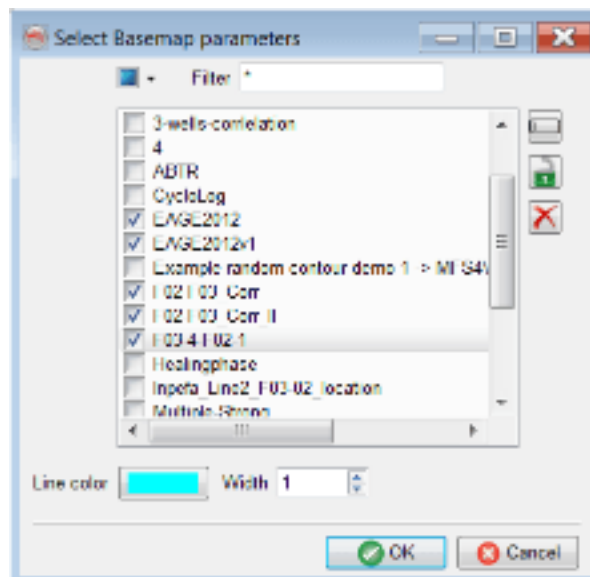
Random line right-click menu



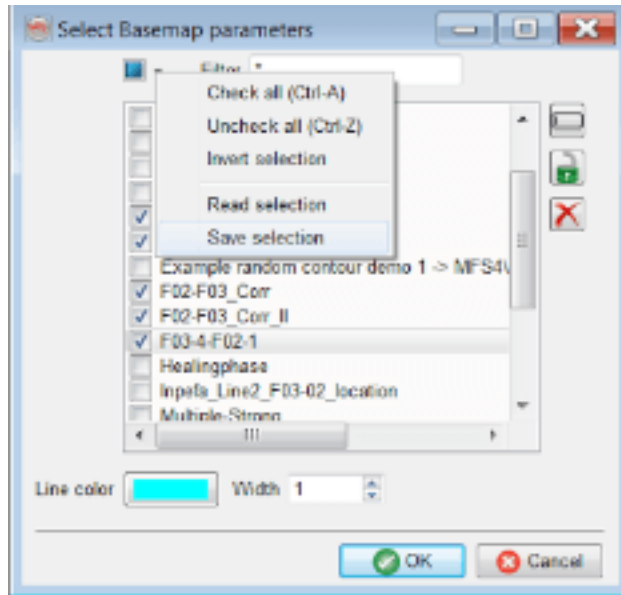
Random line properties

Add Existing Lines

Pre-existing random lines are added using the Random Line > Add right-click menu. When multiple lines are selected a common property (such as line color and line width) is set to all selected lines.



Add selected random lines in the basemap.

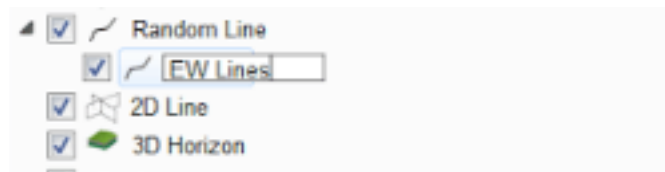


This image shows how to Save selection for future use. Using the same method, one can retrieve the selection by selecting the Read selection option.

Grouping

Random lines can be grouped (Basemap Tree > Random Line > Create Group). In each group, a new random line can be created or an existing random line can be inserted. Sub-groups can also be created using the right-click menu.

To rename a group, double click on the newly added group name.



For the time being, basemap does not support saving of the groups. If you have grouped the random lines in the basemap, we recommend saving the



basemap so that you can retrieve the current layout and settings else the set grouping will be lost.



2D Line

2D lines can be added on individual basis or as a group. The latter is preferred if you are dealing with multiple seismic vintages.

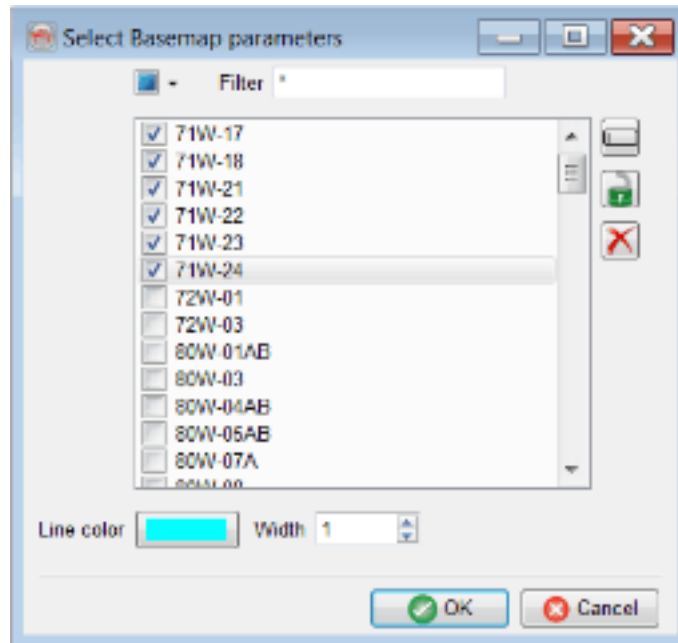


If you intend to add many 2D lines in the basemap, we recommend turning the synchronization off between the basemap and 3D Scene. This will speed-up the displays and will give you a freedom to choose what to display in a 3D Scene.

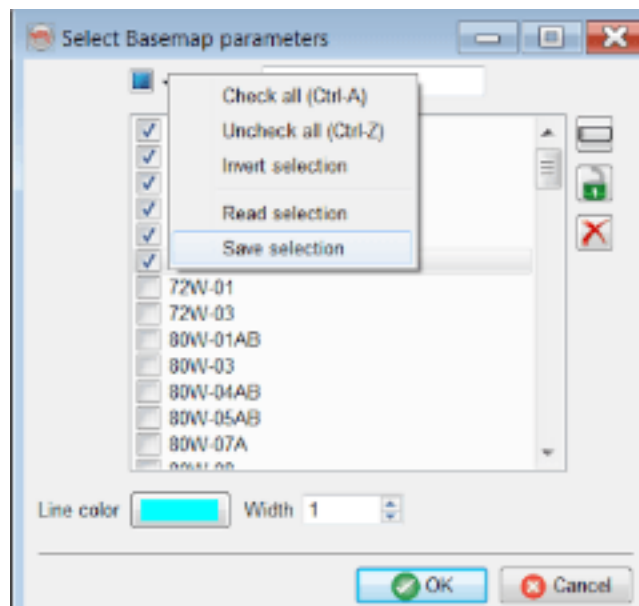
Add Selected Lines

Selected 2D lines can be added in the basemap via the right-click menu (Basemap Tree > 2D Lines > Add). For future use, you may save the selection using the menu available at the top level (a drop-down arrow). This is identical to saving a selection list for the random lines (as mentioned above).

Common line properties (line color and line width) are set to a given selection.



Multiple selection of a list of 2D lines.

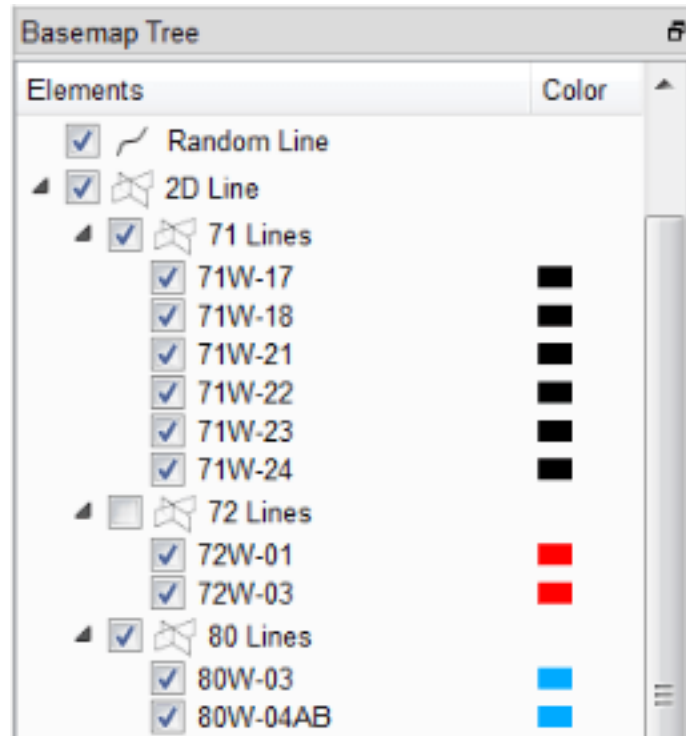


Save selection for future usage.

Grouping

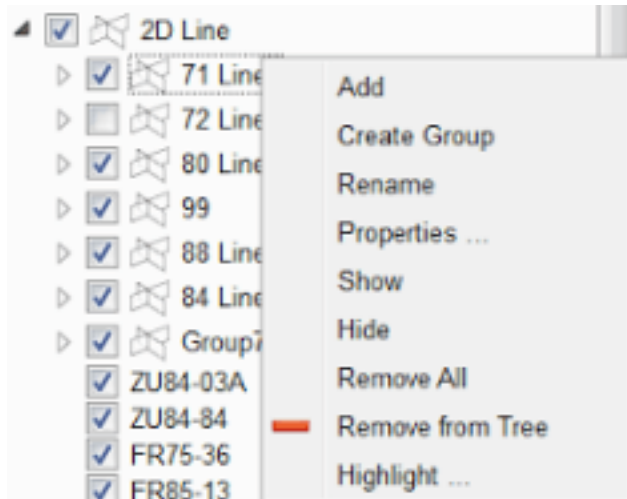
2D lines can be grouped per vintage or as you like. A group contains a list of selection that can be extended at later stages.

If grouping is done in the basemap, we recommend saving the basemap for future use.



An example structure of grouping a set of 2D lines.

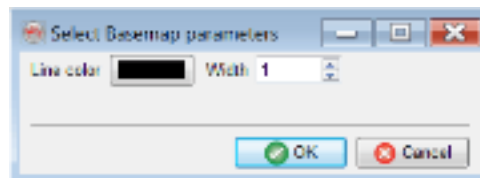
A 2D line group has its own right-click menu to manipulate the displays. Most of the options are self-explanatory. Highlight is the only option that could be used for QC purposes, which highlights the 2D lines on the basemap that have dataset.



2D Line grouping and a right-click menu.

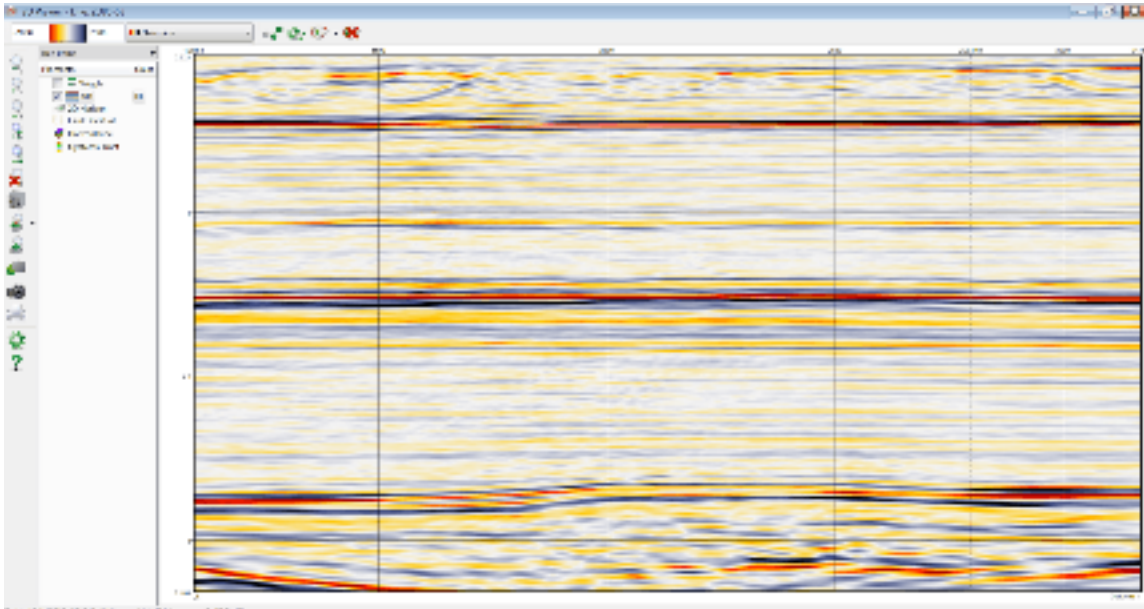
Properties

Line properties can be changed at a group level or at an individual level. Only line color and line width are currently supported.



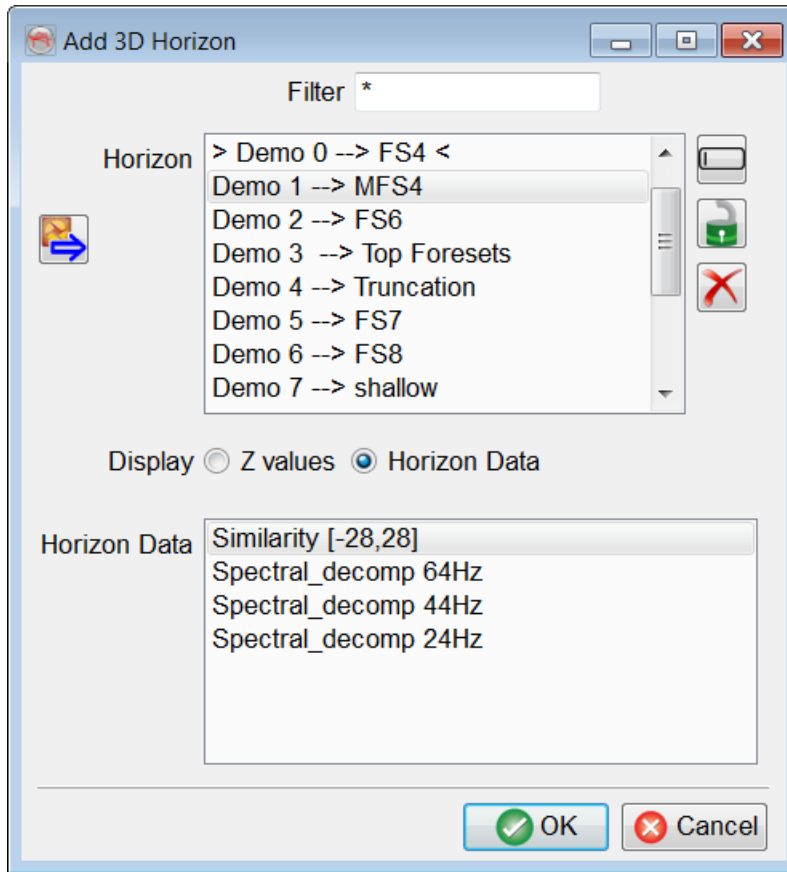
Displaying a 2D Line

A 2D line can be displayed in a 2D viewer or in a 3D Scene. Traditionally, the 2D lines are displayed in a flat viewer such as 2D Viewer in OpendTect. This is done via the right-click menu on the basemap and choosing Show in 2D (as shown below). In such a viewer, one can perform seismic interpretation and continue working with the basemap.

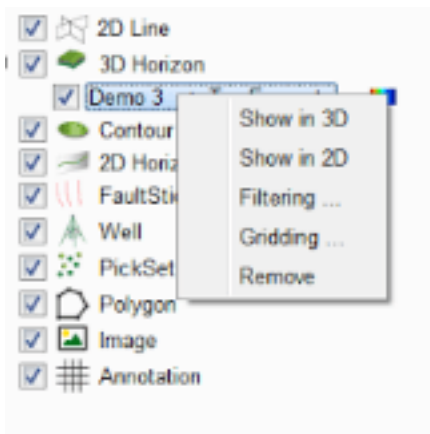


3D Horizon

A color coded horizon grid is displayed in the basemap via this tree item. Attributes computed on a horizon can be displayed as horizon data. To add one or more horizons, use the right-click menu from the basemap tree (3D horizon > Add). The color table of the horizon grid can be changed from the color bar shown in the top toolbar (by default).



If you intend to add large number of horizons in the basemap, it is recommended to turn of the Synchronization with 3D (via the Main Menu > Synchronization).



Show in 3D/2D: Shows the selected horizon in a 3D scene or a 2D Viewer.

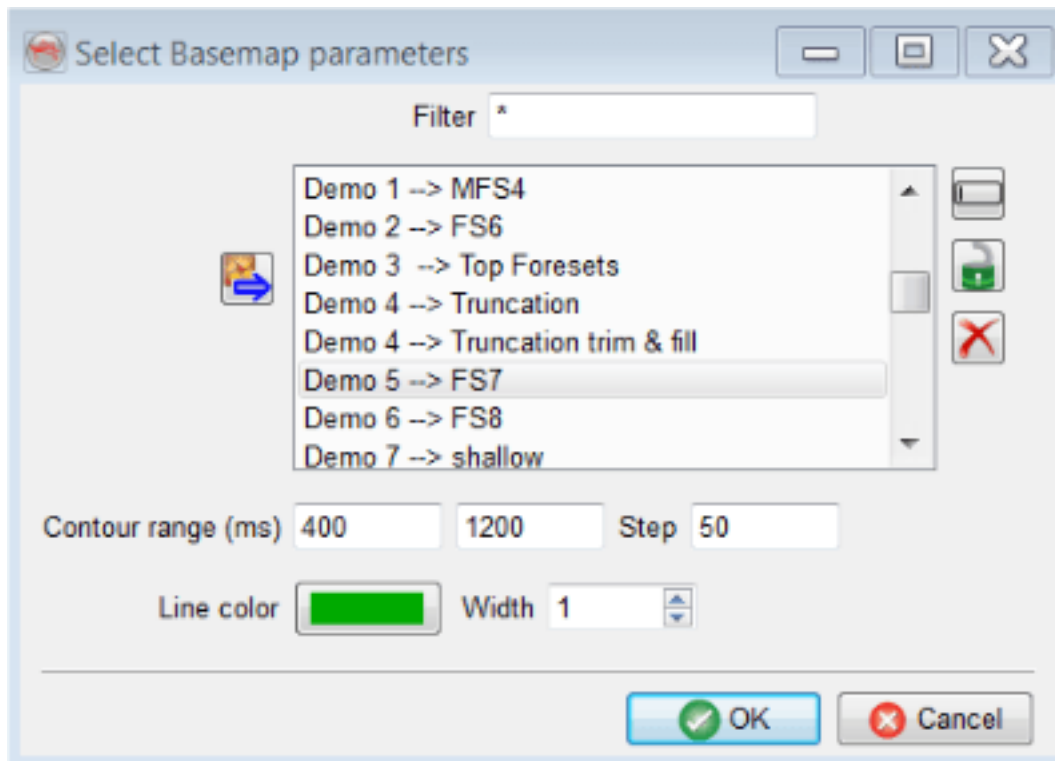
Filtering / Gridding: Each horizon can be individually filtered or gridded by following the right-click menu or from the main menu of the basemap (see under Processing).

Remove: Only removes the horizon display from the basemap.

Note that the order/overlay of the horizon display in the basemap is according to their Z-values (m or ft).

Contour

Contoured display of a horizon grid can be added using this tree item. The following window shows the contour range, contour step, and line properties for a selected window. By default, contours are plotted on top of the 3D horizon.



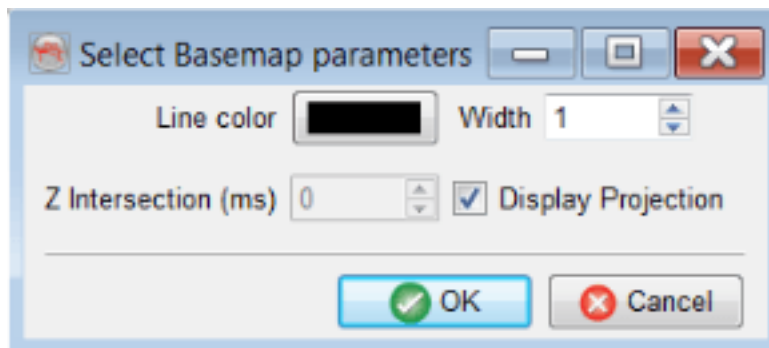
After the display, you can change the contour range/step by right-clicking on the contour item in the tree and choosing the properties option.

2D Horizon

Since version 6.2 2D horizons are displayed as color-coded lines. Grouping feature is also supported for 2D Horizons.

Fault Stick Set

Fault sticks are projected on the basemap and hence may show the spatial location and extent of the fault.



Fault stick properties

FaultSet

The FaultSet is supported in version 6.6.0 and can be displayed in the basemap.

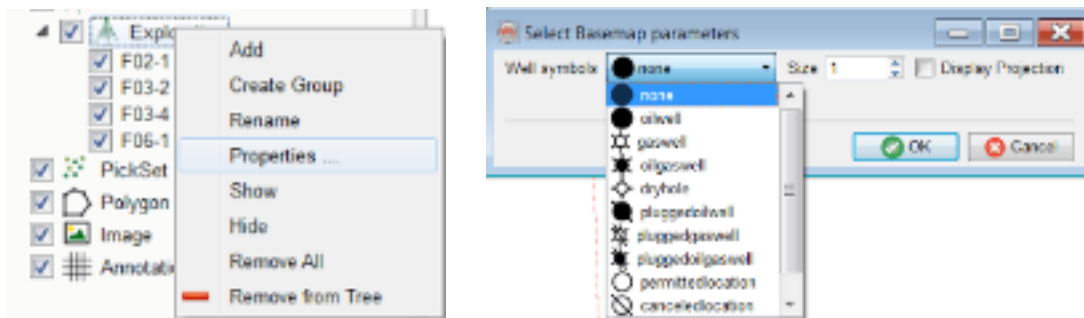
The following shortcut can be used for drawing a polygon in basemap: CTRL+Z CTRL+Y.

Properties can be set for both pointset and polygon.

Well

To add one or more wells in the basemap use the right-click in the basemap tree. Just like other elements (e.g. Random lines / 2D lines), wells can be grouped by creating a group and adding a set of wells in it. If you intend to create such a thing, we recommend saving the basemap.

Well properties allow you to control the well symbol, size, and projection.



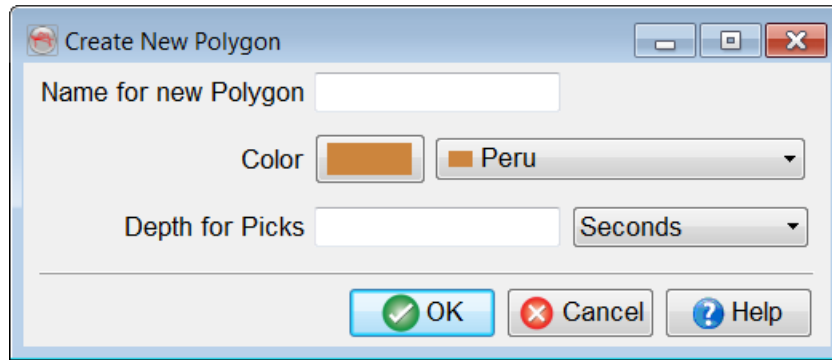
Well properties – Display projection to see the well trajectory.

Pickset

This element is reserved for displaying a point (or pickset) data in the basemap.

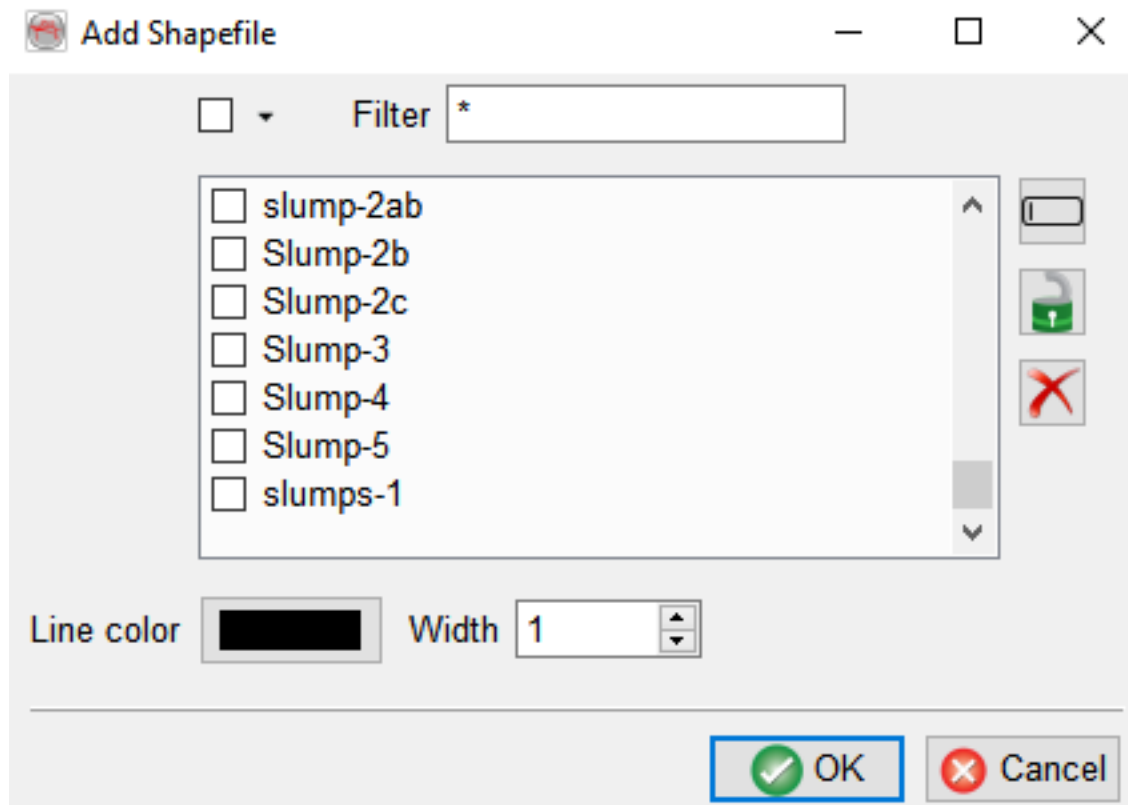
Polygon

This element is reserved for displaying existing polygons or creating new polygons in the basemap.



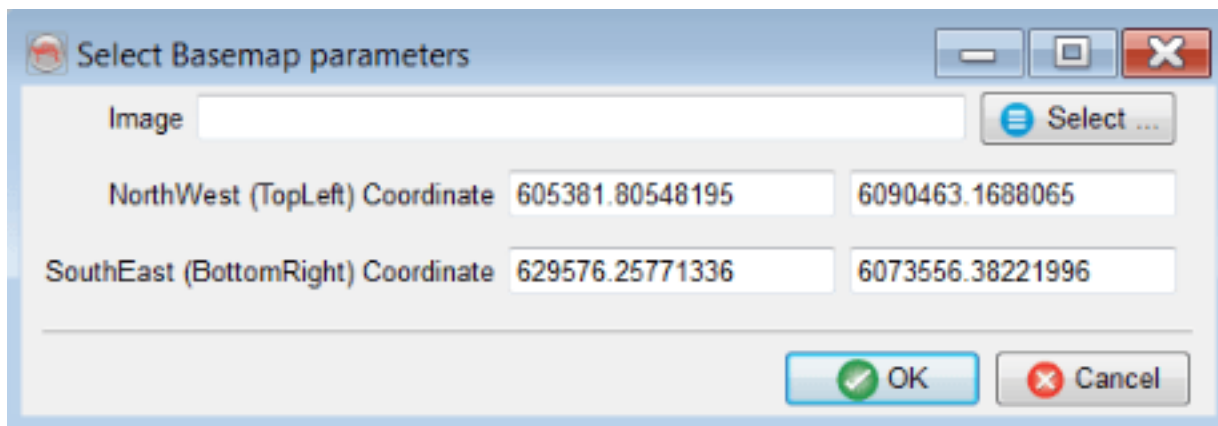
Shapefile

Allows adding shapefiles to the basemap.



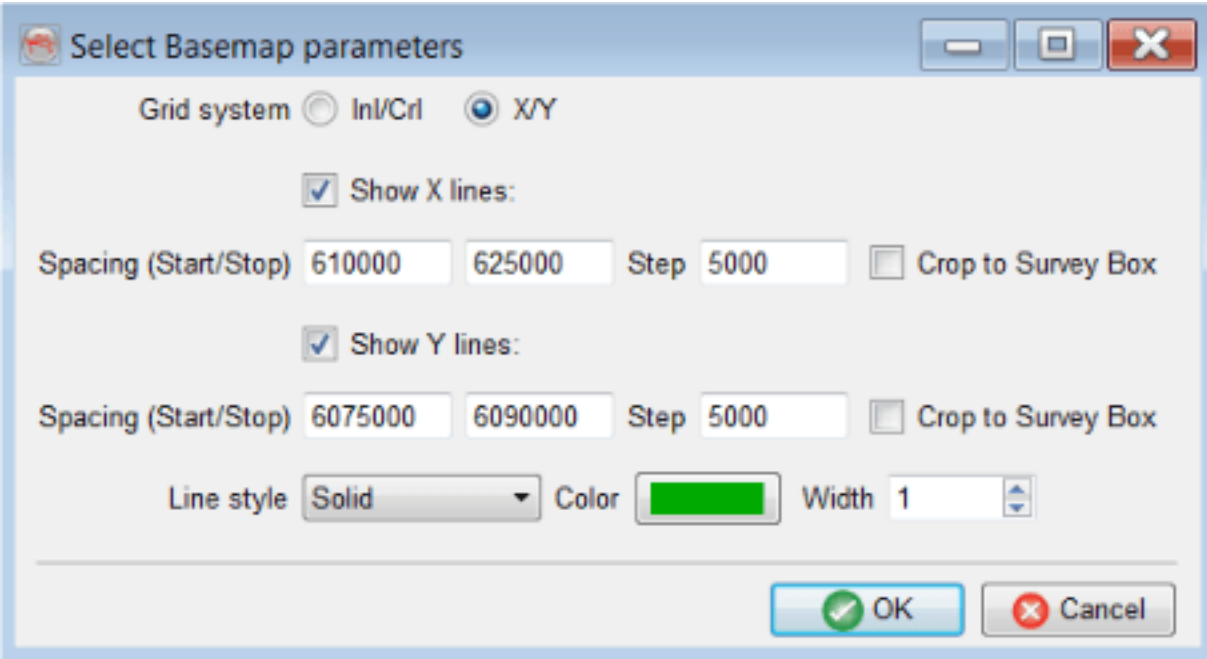
Image

An image (PNG, JPEG, XPM, or BMP) can be added in the basemap. Its coordinates have to be properly added to position it at correct level. An image can have borders, which may not position the image correctly. You can use geo-reference (e.g. [QGIS](#)) tools to measure the actual coordinates of the image.



Annotation

To add grid lines on the basemap, this element is used. Grid lines and their annotation can be based on inline / crossline steps or XY spacing.



Toolbars

Main (Top) Toolbar

This toolbar appears (by default) under the main menu of the basemap. It contains the following icon:



Shows the selected item in a 3D scene.



Removes selected items from the basemap (and/or 3D scene if sync is ON).

General Toolbar



Open a stored basemap setup.



Save the basemap.



Save the basemap with a new name.



Take a high resolutions snapshot of the basemap.



Prints the basemap on a larger scale (e.g. PDF of A0 size).



Show or hide the North Arrow in the basemap.



Show or hide the scale bar in the basemap. Scale bar also has settings next to it (small arrow) to change the width of the bar.



Show or hide the color bar in the basemap.



Show/hide the legends in the basemap. Note that this item is editable (click on the text to change it). It can also be freely moved and placed at a desired location on the basemap.



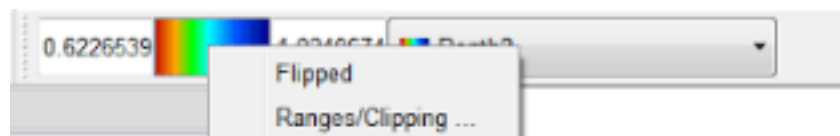
Selects data within the polygon e.g. horizon.



Removes/deletes the selected data.

Color Tables

Color tables are linked to 3D horizon grids displayed in the basemap. You can flip the color bar values, clip the range, or manually fill in the numbers.



Title and Printing

Basemap title appears within the basemap. This field is editable (single click on the text and start editing). We support two ways of printing the basemap. The first one is via the Screenshot method, where you can take a high resolution image of the current map. The second one is via the print icon, which allows you to change the orientation (e.g. landscape), and output printer (e.g. PDF if installed).

1.3 PDF3D



The PDF3D functionality has been expanded to include the option to export the 3D scene to a 3D viewing format called glTF.

Scroll down to view the possibilities for this option.

PDF3D

The plugin allows you to grab a 3D scene in OpendTect and store this in pdf format. The file can be opened in the free Acrobat Reader, or embedded into a Powerpoint presentation. In Acrobat Reader (v8 and higher) you can view, rotate, zoom, toggle elements on and off, and otherwise manipulate 3D graphics.

Currently supported objects include:


- In-line/cross-line/z-slice
- 3D, 2D horizons
- 2D line
- Well data
- Fault
- FaultStickSet
- Body
- Random lines
- Top/Bottom Image
- HorizonCube and SSIS models*

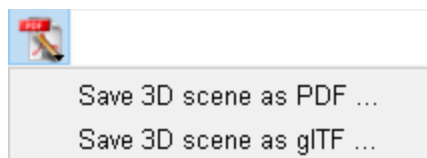
- PointSet/Polygon. Multiple pointset shapes such as cylinder, cone and cube. If a shape is other than these just listed, it will be converted to sphere. These shapes' size is dependent on the size of the pointset displayed in 3D Scene.

- ColorBar

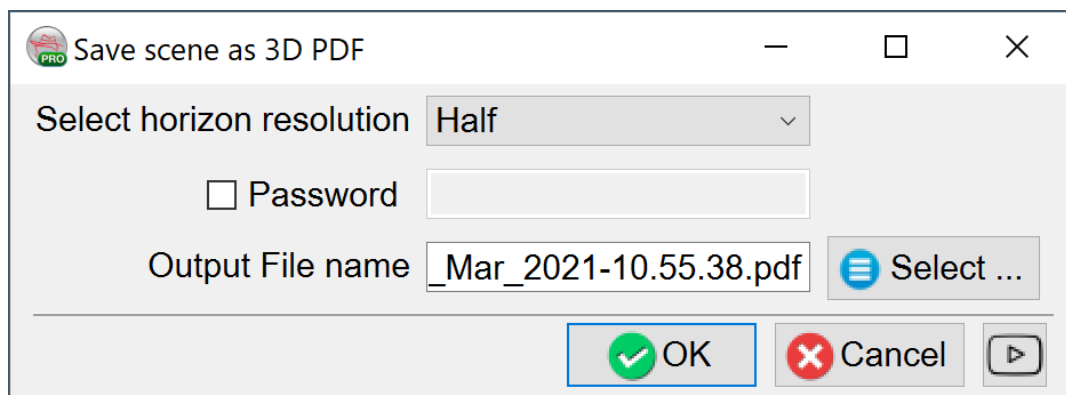
- 2D Seismic Line is now shown.

** requires HorizonCube and SSIS commercial plugins*

To create the file click on PDF3D icon  . This will open a pdf file containing all the elements displayed in the active scene and will automatically save it to the Misc folder of the project.



Save scene as option will first bring a window where you can choose the desired path for the pdf file and the name, select horizon resolution and set a password if necessary.




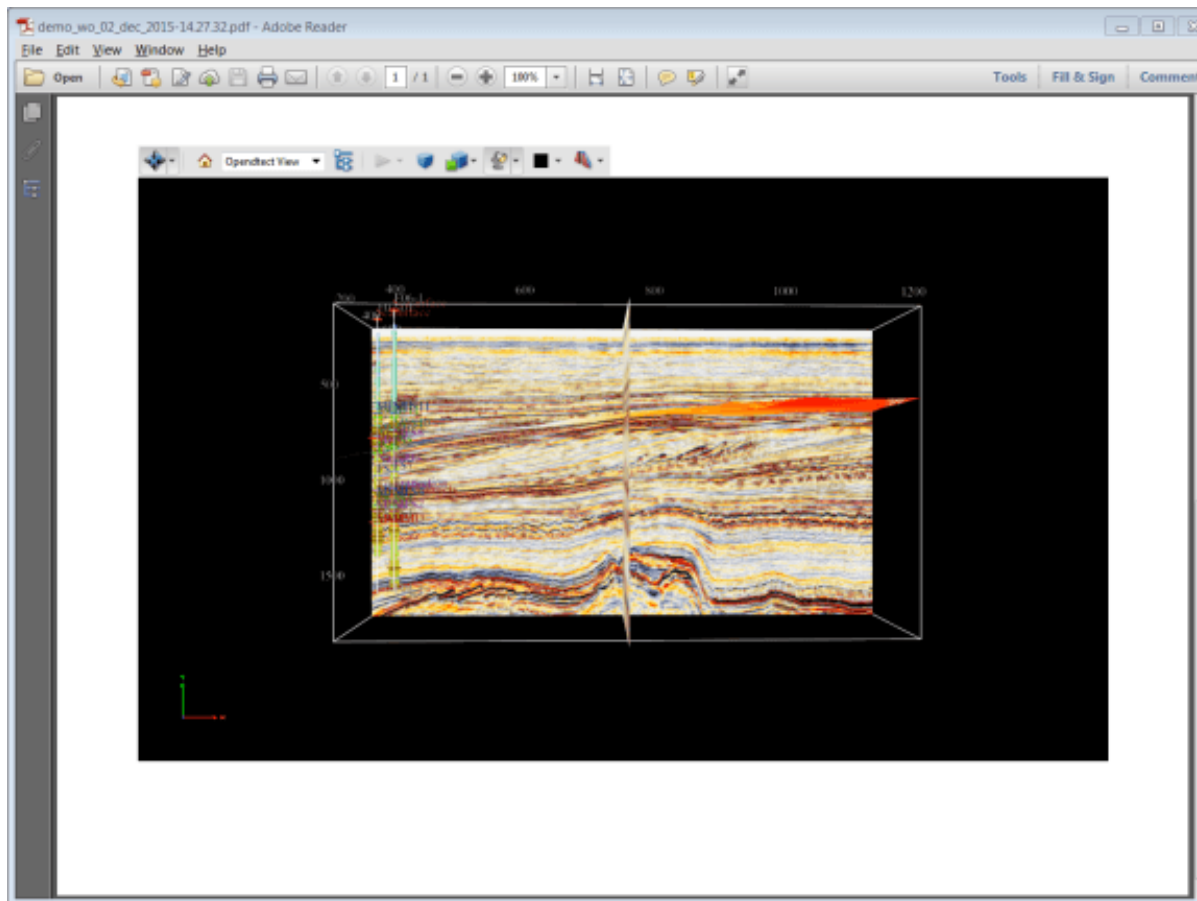
The 3D toolbar contains several icons to manipulate the scene. You will find more information on the [Adobe website](#).

Please note: It may be necessary to give Adobe permission to display the 3D content, as in the following image. Use the Options drop-down to trust the document either once or always:

! Multimedia and 3D content has been disabled. Enable this feature if you trust this document.

Options ▾

Toggle on the Model Tree icon  to display show a list where you can see the names of the scene elements and toggle them on and off. Decreasing of the Framerate Threshold to 4FPS will help to avoid blinking of the elements while rotating or repositioning. To do so, right-click on the scene > 3D Preferences > 3D and Multimedia Category.

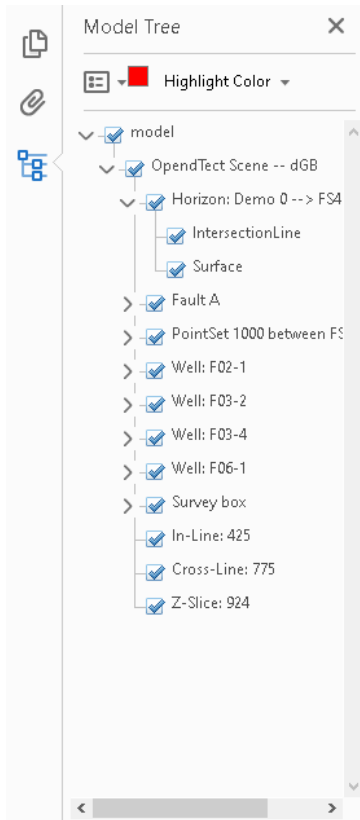



For optimal viewing in Adobe Reader (v8 and higher) we recommend the following:

- Run Adobe Reader (not embedded within a browser such as Internet Explorer)
- Click on the menu *Edit->Preferences...* to pop up Preferences dialog
- Select under Categories: (box on left) *3D and Multimedia*
- Turn on *Enable double-sided rendering* in *Renderer Options* box
- If the scene is not too large then you might wish to change the *Optimization Scheme for Low Framerate* from "Bounding Box" to "None" in the *Auto-Degrade Options* box. Alternatively you might wish to continue using *Bounding Box* but reduce the *Framerate Threshold* from 15FPS to 1FPS.

Furthermore, we recommend viewing directly in Adobe Reader and not as a plugin to a browser. Toggling elements on and off is easiest if you use the tree.

It is now possible to toggle between horizon intersection lines with Seismic slices and full display of horizon (called Surface in tree item):



In case the model tree is not visible, it can be toggled on from the  button in the sidebar.

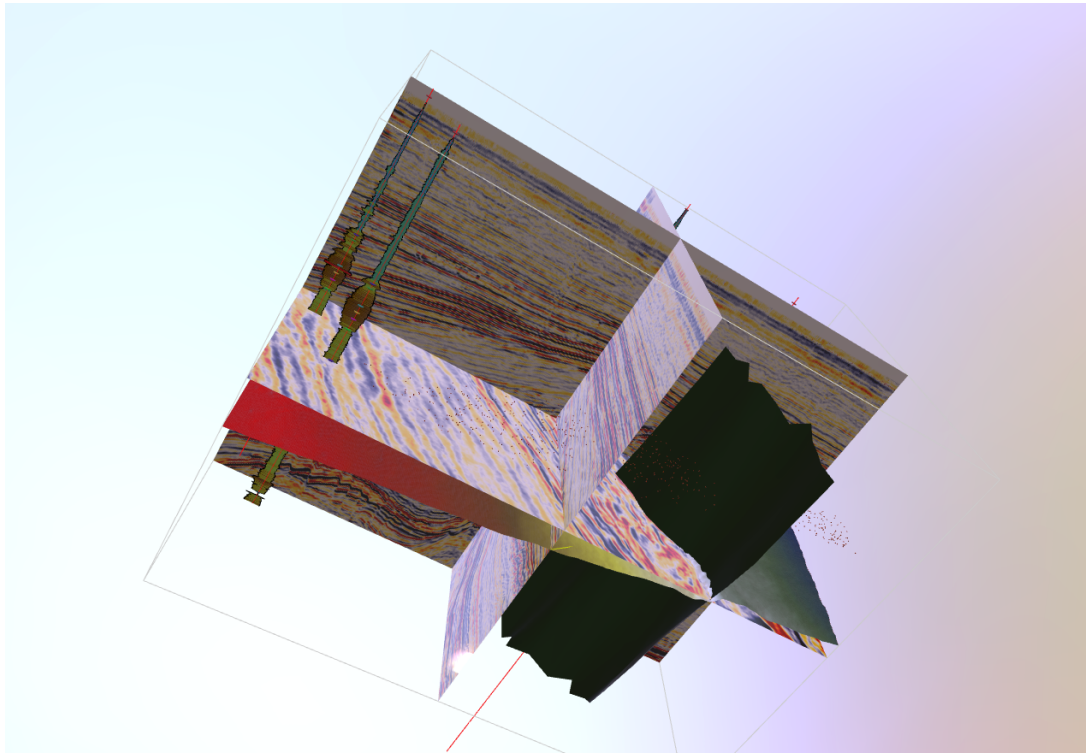
Note: Currently supported items in PDF3D should not include the ColorBar as it can not be toggled off from within the Adobe reader.

System requirement (macOS)

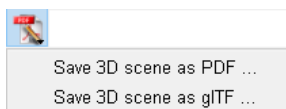
macOS needs to have the XQuartz application installed for PDF3D to load. XQuartz can be downloaded for free from their [web page](#). XQuartz is a 3rd party application, and dGB Earth Sciences should not be held responsible for the performance of XQuartz.

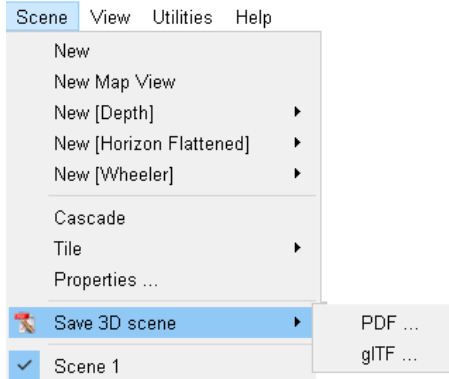
gITF

The files produced in this format can be viewed on the native Windows 3D viewer, available in Windows 10 & 11. These files can also be viewed online.

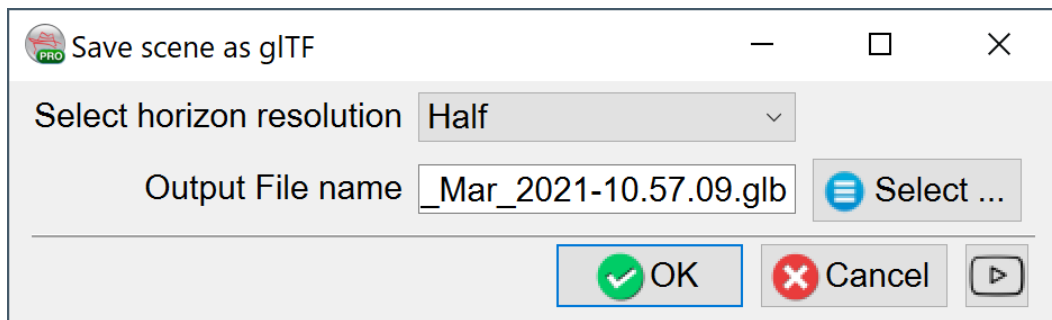


This option can be accessed via the *PDF3D Tool Button* (or from *Scene -> Save 3D Scene -> gITF...*):



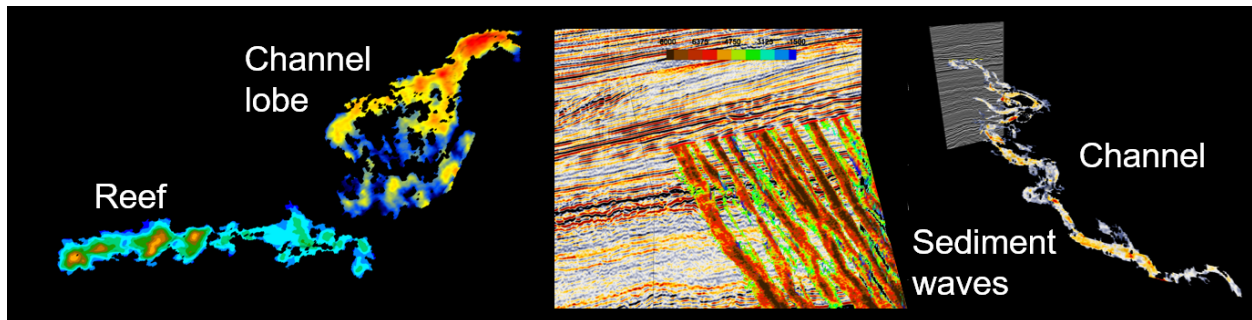


On selecting to save as glTF, the following dialogue box appears, allowing the user to specify the resolution and to set the output file location.



The default location is in the Survey\Misc folder.

1.4 Thalweg Tracker



Introduction

This plugin tracks seismic facies such as channels and reefs. It outputs either horizons with a set of tracking attributes, or 3D objects, or Pick sets.

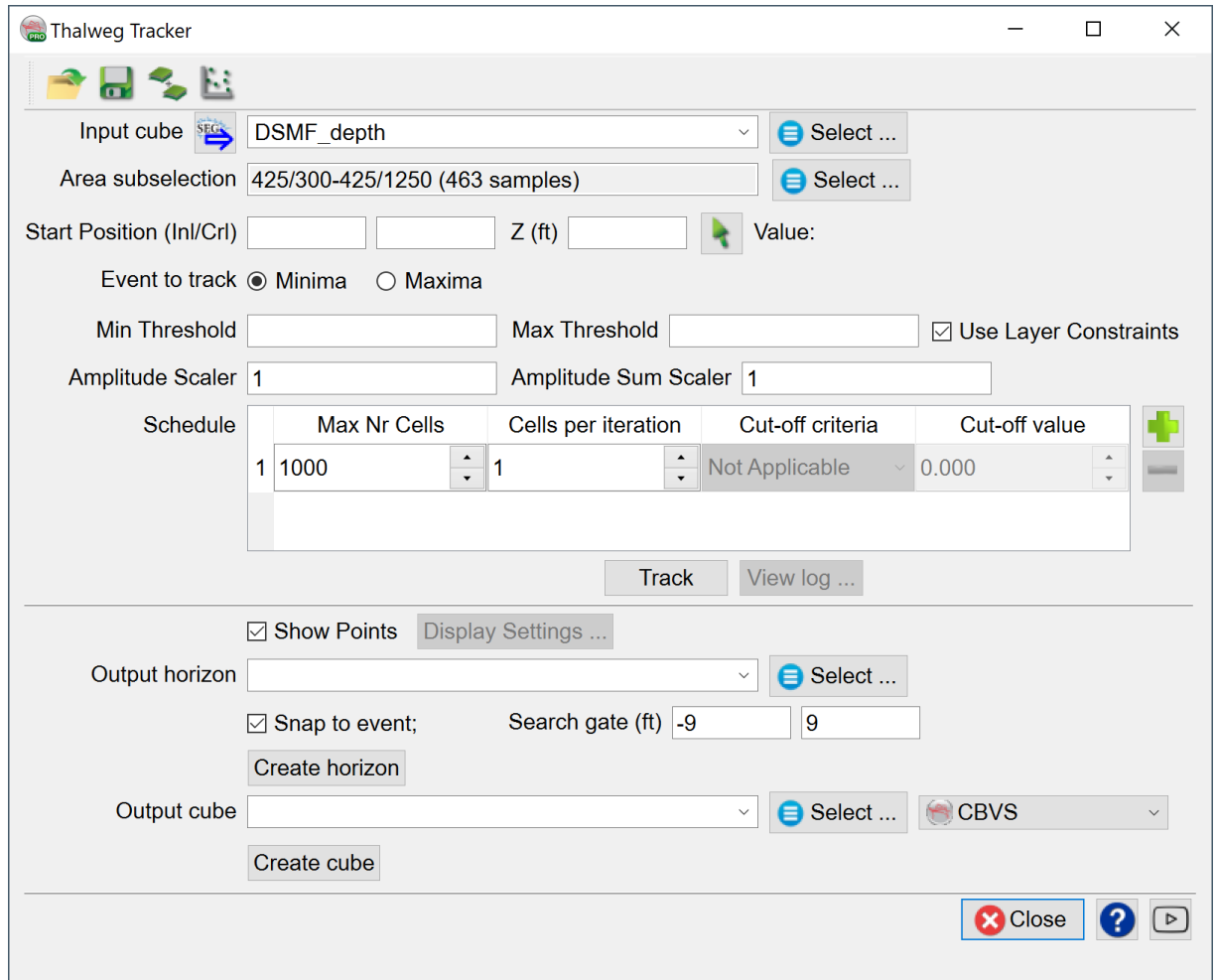
A Thalweg is a geologic term to describe the path of a river as it flows through a valley (Thalweg is a German word; Thal means valley and Weg is path). A Thalweg tracker operates in a similar way: the tracker follows the path of least resistance.

The Thalweg tracker is a special form of a voxel connectivity filter. Based on certain user-specified constraints, It tracks neighboring samples in an input seismic cube. Initially the user picks a single seed position. The seed is considered to be a cube of unit size and the next sample to be tracked is chosen from all available samples along the 'faces' of the seeds. In the first iteration, all six neighboring samples along the six faces of the initial seed act as candidates for tracking. Only the best matching position (i.e. the highest, or lowest amplitude) is added. In the next iteration, all samples neighboring the two currently accepted positions now act as candidates and again only the best matching position is added. This process continues until it is no longer possible to add candidates that meet the tracking constraints.

A Thalweg tracker adds only one position per iteration. If you choose to accept more than one position per iteration, the Thalweg tracker becomes a margin tracker. Thalweg and margin trackers are typically run sequentially.

The output of a Thalweg tracker is a Point Set that can be saved as "snapped" horizon (parameter window), or it can be saved as a 3D body. or Pick set (3D scene, right-click on the object). When it is saved as a horizon all tracking attributes will be saved as Horizon data with the horizon. The tracking attributes can be used for further analysis in the Crossplot tool.

Menu



Input cube: This is the input seismic (or attribute, impedance, ...) cube.

Area subselection: The tracking area (or volume) can be constrained using one of the many available criteria, viz. Range, Polygon, Horizon etc.

Start Position (Seed): This can be picked interactively with the green arrow. The picked amplitude is displayed immediately after picking.

Event to track: Samples can be tracked either on maximum, or minimum amplitude.

Thresholds and Layer Constraints: The automatic values for min and max thresholds are estimated from the picked position. These values can be overruled manually. When the Layer Constraints toggle is on the software will not allow a body to grow on traces where the body already exists. This (default option) prevents tracking multiple forks of an event.

Amplitude Scalar: This optional scalar is applied to the Amplitude attribute output and can be used to improve visualization.

Amplitude Sum Scalar and Amplitude Sum Step: The optional scalar is applied to the Amplitude Sum attribute output and can be used to improve visualization. The Amplitude Sum Step is the re-sampling rate that is used to compute the Amplitude Sum attribute (see Display options).

Schedule: This is the control center of the tracking process. The idea is that objects are tracked in one, or more sequential processing steps. Steps can be added with the green plus and red minus icons, respectively. Each step has its own parameters. Typically, the first step is run in Thalweg tracking mode, which means that only one cell per iteration is added. Only the best matching cell (the lowest, or highest amplitude of all neighbors) is added. In following steps the constraints can be lowered. More cells are added per iteration. This is the margin tracking mode.

A typical work flow is the following:

- First the Thalweg tracking mode is used to track a channel.
- Start by tracking a large number of positions. Let's say Max. Number of Cells is set to 20,000.
- Examine the result and observe (from the iterations attribute) where the body left the first Thalweg and spilled over into the next Thalweg.
- Re-track the object from the same seed position but now use a smaller number of points that will track the object until it reaches the spill-point. You have now found the first channel patch.

- To track the channel margins add another layer of let's say another 20,000 points but now accept more cells per iteration and accept more points than the best matching only.
- Save the tracked patch as a horizon.
- Repeat the process from another seed position and merge the patches (merge icon at the top of the window).

Max. Nr cells: This is the maximum number of cells that the body can grow with when **Track** is pressed.

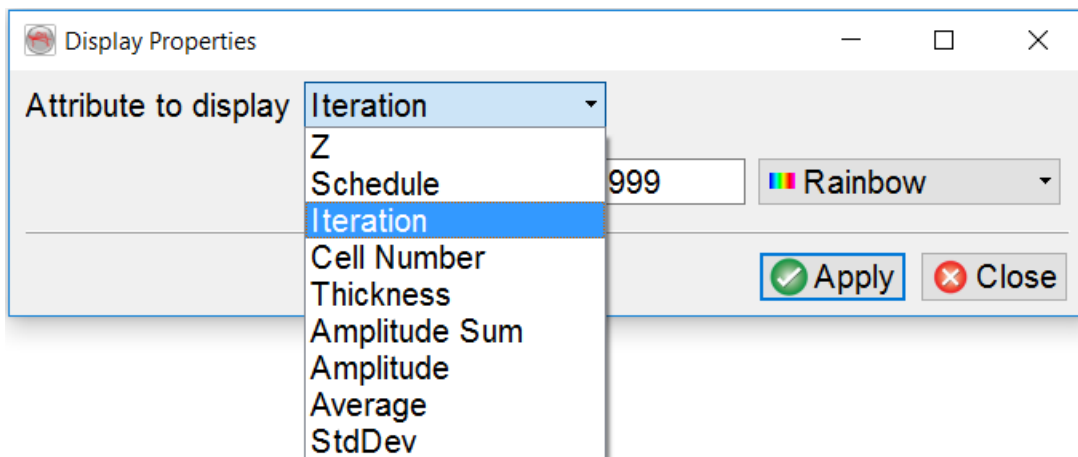
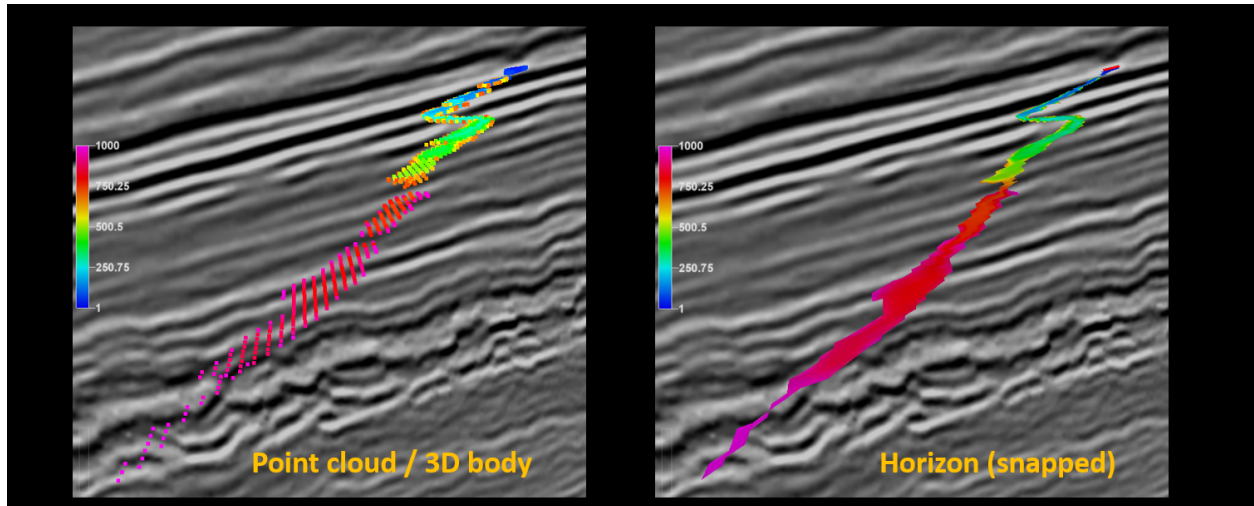
Cells per iterations: This is the maximum number of cells that will be added per iteration, i.e. each time the neighbors are evaluated and the body grows. If set to one the tracker runs in Thalweg tracking mode. A value larger than one means margin tracking mode.

Cut-off criteria and Cut-off value: These constraints control which cells are acceptable when tracking in margin mode. A percentage cut-off of 5 means that the highest, or lowest 5% of all possible neighboring values are eligible. Out of this group the best matching cell is added first, followed by the next best and so on. This continues until either the number of cells per iteration is reached, or there are no more cells in the pool of eligible cells to add. The standard score criterion operates in the same way. The eligible neighboring values are the amplitudes above, or below the mean value of the body plus, or minus the number of standard deviations given in the standard score cut-off.

Track and View Log ...: Press track to actually run the tracker. Only the last step is processed. The output is shown in the 3D scene as a point cloud (left image below). **Note** that the point cloud is not saved! This is an interactive process. Play with the parameters until you are satisfied. Only then save the results: either as a horizon (right

image below), or as a point set, or 3D object (in the 3D scene). Press View log to see the processing log file that prints statistics after each iteration.

Display and Output options



Attribute to display: Two sets of attributes are computed by the Thalweg tracker: Process attributes and Body attributes. The Process attributes tell us how the body and its properties grew from the seed position. Attributes in this group are: Schedule, Iteration, Cell Number, Average and StdDev. Schedule corresponds to the processing step in

which the cell was added to the body. Iteration returns the iteration step that was reached when the cell was added. Remember that in margin mode more than one cell is added per iteration. The cell number is the sequential number assigned to the cell as the body grows. Average and StdDev are the running average amplitude and the running standard deviation of the amplitudes. These statistical properties vary as the body grows.

The Body attributes are: Z, Thickness, Amplitude Sum, and Amplitude. Z is the depth of the cell in 2WT, or depth, as the case may be. Thickness is the vertical thickness of the body. Amplitude sum is the vertical sum of the amplitudes computed from interpolated amplitudes with the user-defined sampling rate given in the Amplitude Sum Step field. Amplitude is the amplitude of the cell.

Output Horizon: The tracking results can be saved in the form of a Horizon. Typically, the horizon will be snapped to a maximum, or minimum event that is found within the user-defined search gate. All attributes will be saved as Horizon data with the new horizon. If the horizon was snapped, the amplitude attribute will return the amplitude of the snapped event.

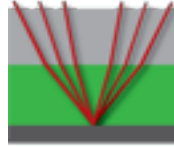
Output 3D Objects, or Pick Sets: To save the results as Opendtect Pick set, or as 3D object, right-click on the point cloud in the 3D scene and use the corresponding menu option.

References:

Pelissier, M., Changhua Yu, Singh, R., Qayyum, F., de Groot, P. and Romanova, V. 2016. Thalweg Tracker: A Voxel-Based Auto-Tracker to Map Channels and Associated Margins. EAGE 78th EAGE Conference & Exhibition, Vienna, 30 May - 2 June 2016.

See also the [pdf](#), and/or [video](#) recording of the oral presentation.

1.5 Advanced Ray Tracer



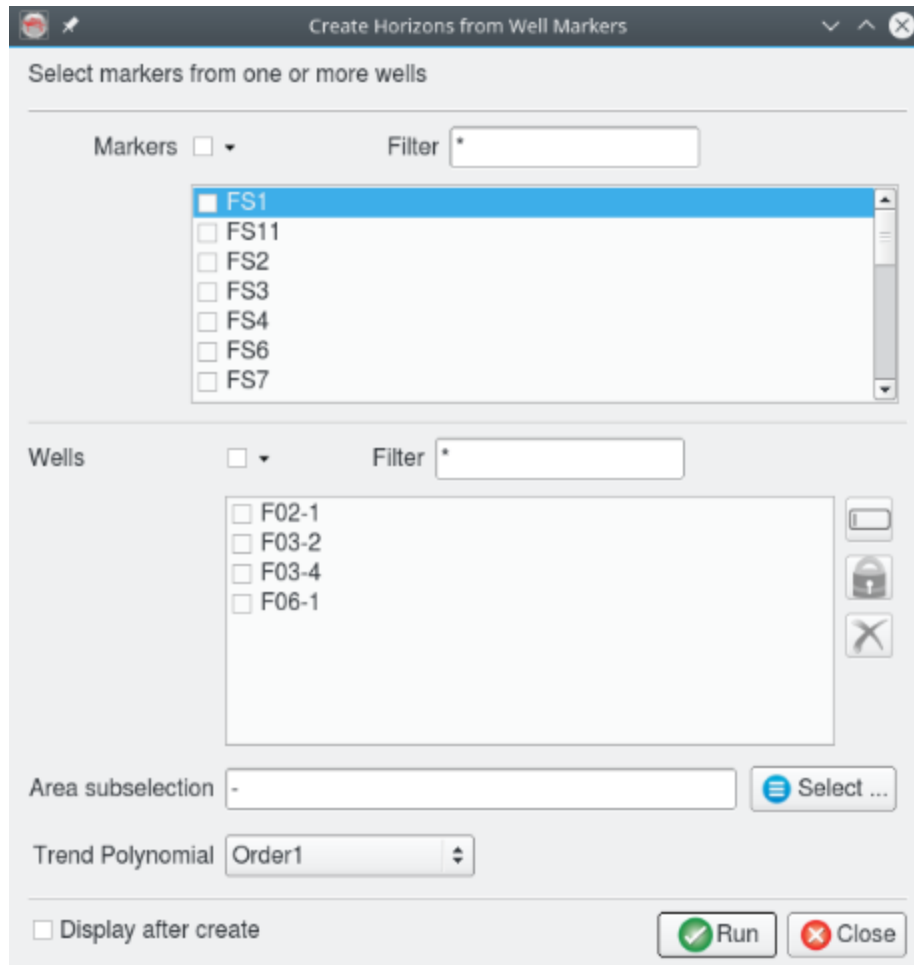
The full article can be found in the [user documentation](#) under *5 Analysis > 5.5 Layer Modeling > 5.5.1 Basic > 5.5.1.2 Synthetic- and Property-Log Generation > 5.5.1.2.3 Ray Tracing*.

1.6 Horizon from Well Markers

The tool is utilized for creating 3D horizons from well markers using [Radial Basis Function](#) gridding.



If need be, 2D horizons can be later derived from the 3D ones using [Create from 3D](#) option.



Selecting more than one markers will result in creating several horizons at a time. The output horizon will be saved to the database under the name Horizon From *selected marker*.

Trend Polynomial: Trend processing must be applied during gridding using Radial Basis Functions. The order of the trend removal may influence more or less the resulting 3D horizon:

- **None:** Disables trend removal processing.
- **Order0:** RBF gridding with average removal
- **Order1:** RBF gridding with linear trend removal (default)
- **Order2:** RBF gridding with parabolic trend removal

The resulting output horizon depends largely on the lateral distribution of the control points (the wells). Try and choose the best parameter for the Trend Polynomial based on the most realistic result, as it is not possible to state which order will return the most reliable horizon.

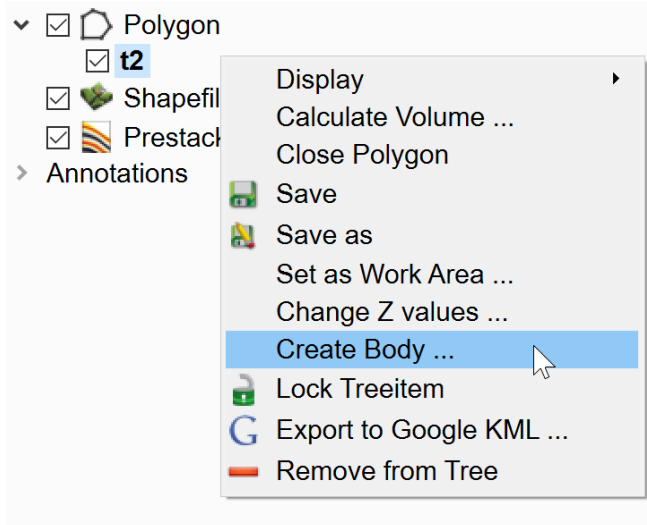
1.7 Well Table

Well name	UWI	X (m)	Y (m)	TD (m)	KB (m)	GL (m)	Density	Sonic	Gamma Ray
F02-1	F02-1	606554.00	6080126.00	1695.00	30.00		kg/m3	us/m	API
F03-2	F03-2	619101.00	6089491.00	2140.00	30.00		kg/m3	us/m	API
F03-4	F03-4	623255.98	6082586.87	2048.00	34.10		kg/m3	us/m	API
F06-1	F06-1	607903.00	6077213.00	1701.00	28.64		kg/m3	us/m	API

New tool for well data management, introduced in the 6.6 release of OpendTect. Color-coded cells and rows show which data in which interval is present.

Log plots and crossplots that use bokeh plotting are available via the corresponding icons.

1.8 Create Body



A body can be extracted from a drawn polygon. A polygon can outline a geobody e.g. channel, valley, other geomorphological feature, or any arbitrary shape.

The generated volume (body) is limited by the polygon surface area and the horizons chosen for top and bottom.

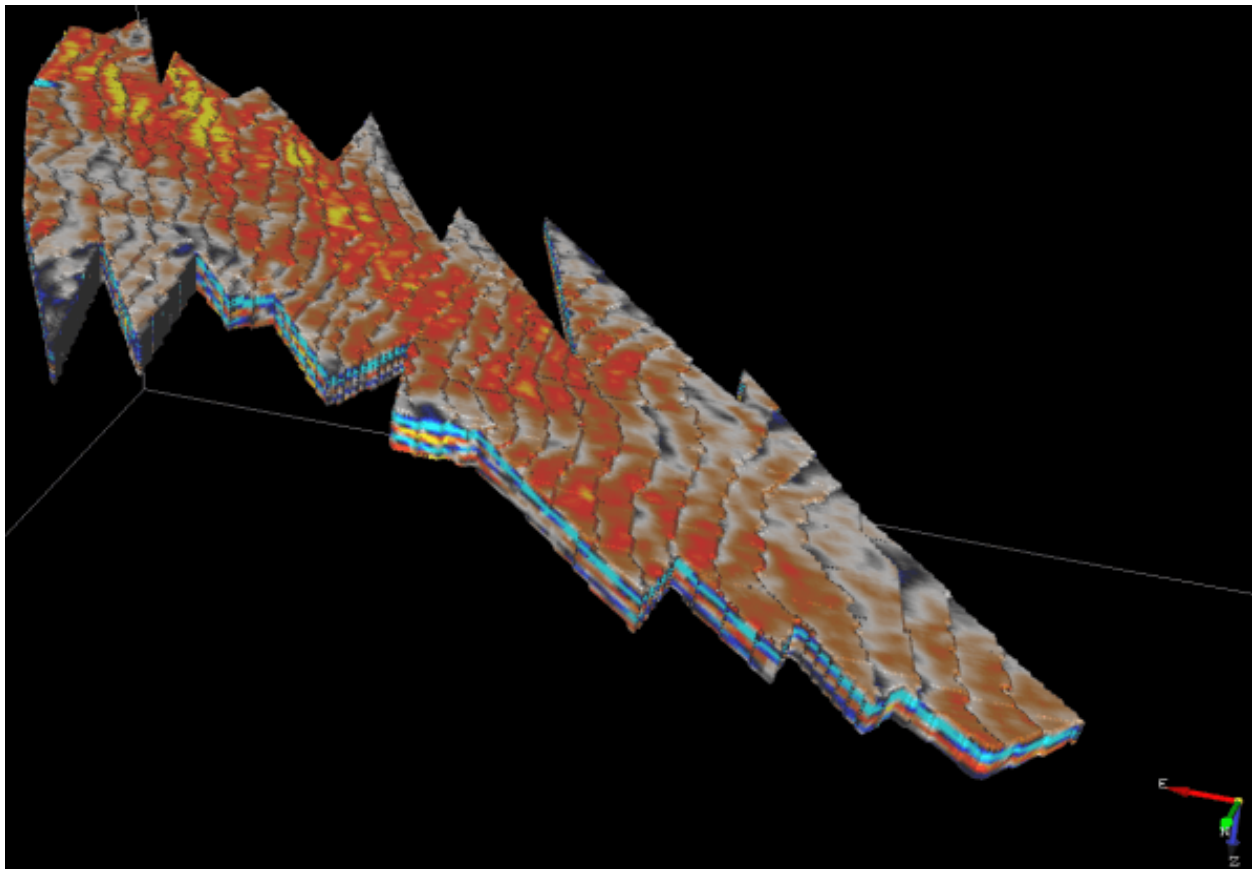
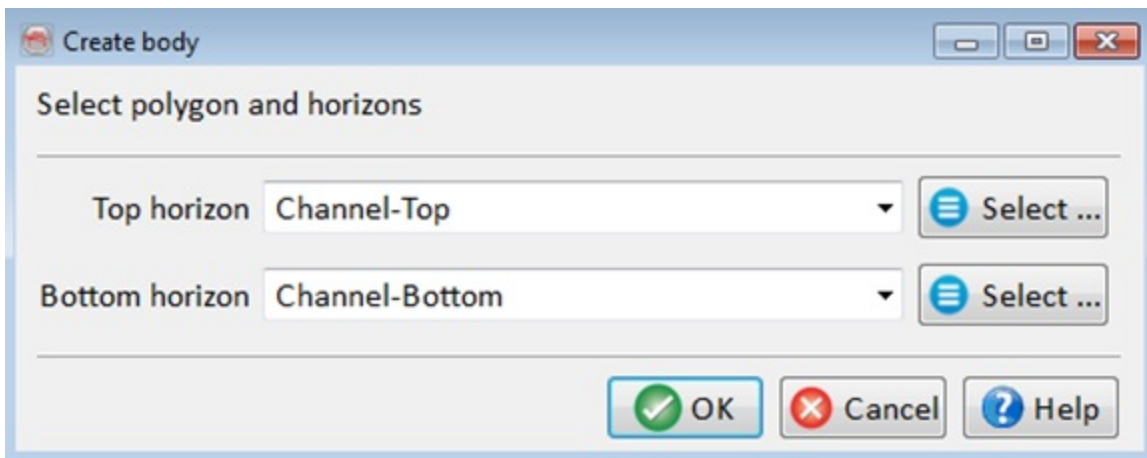
Different attributes can be displayed on the body on-the-fly, which in turn can be interpreted further.

The workflow is as follows:

1. Create a new polygon that defines a geological shape.
2. Right-click on the new polygon name and select *Create body*
3. In the pop-up window select the top and bottom horizons to define a vertical limit.

Press *OK* to continue

The setup will create and display the output body. An example of a body of sand-waves from offshore seismic data is shown below.



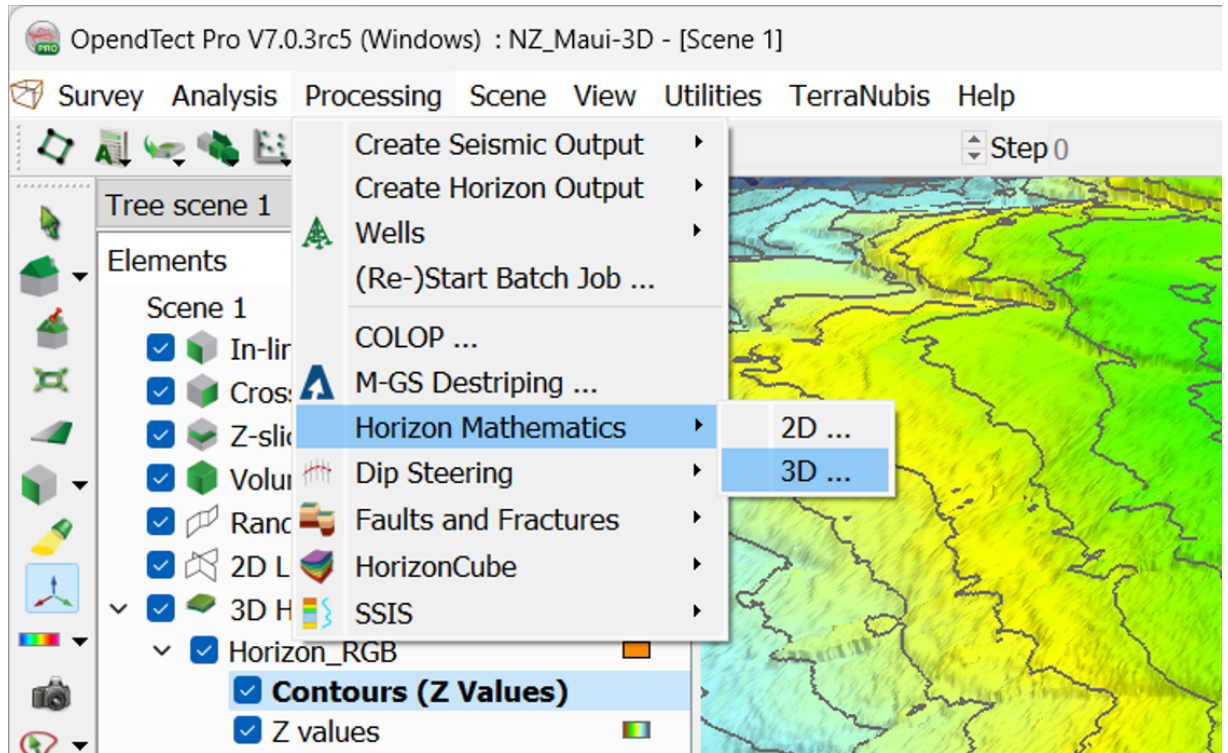
1.9 Horizon Mathematics (2D and 3D)

The horizon math (both 2D and 3D) tool allows for utilization of the following statements and statistics:

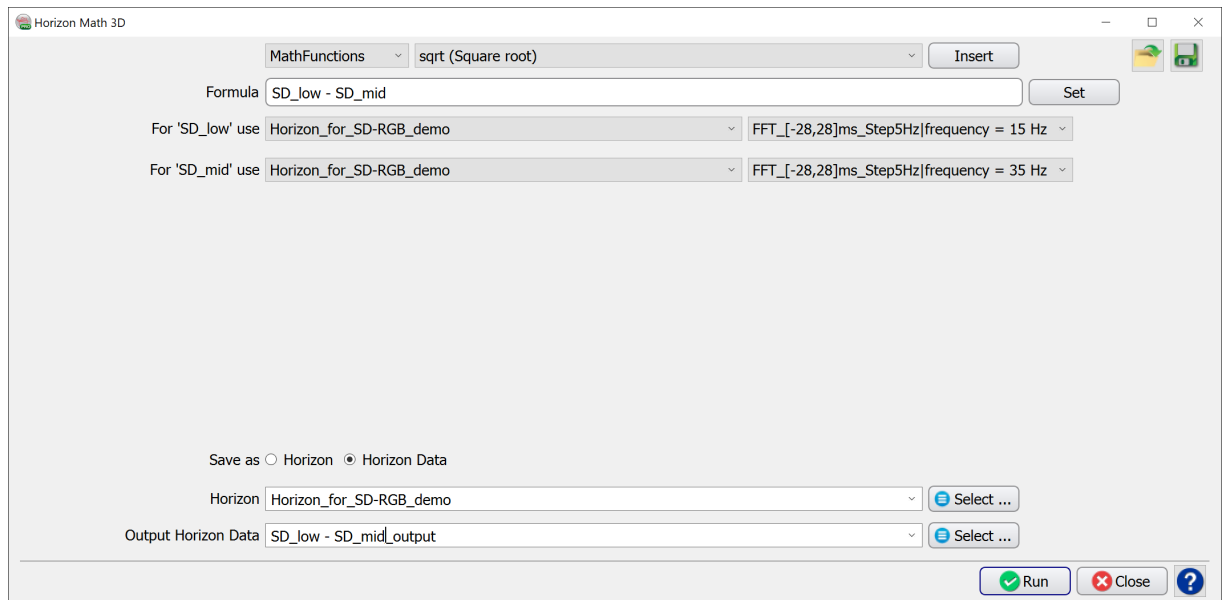
- Basic math (+, -, *, /, ^)
- Functions (exp, ln, sin, ...)
- Logical (<, ==, &&, ...)
- Statistical (rand, min, avg, ...)
- Constants (pi, e, undef)
- Other (inl, crl, X, Y)

You can type in any formula and apply any values and mathematical operators and formulas, e.g.: for converting grids from time to depth.

The tool can be accessed from the Processing menu → Horizon Mathematics. Here, select if you want to apply the horizon math tool on your 2D line data or 3D horizons.



Depending on your selection, the UI for either the 2D or 3D Horizon Maths will appear:



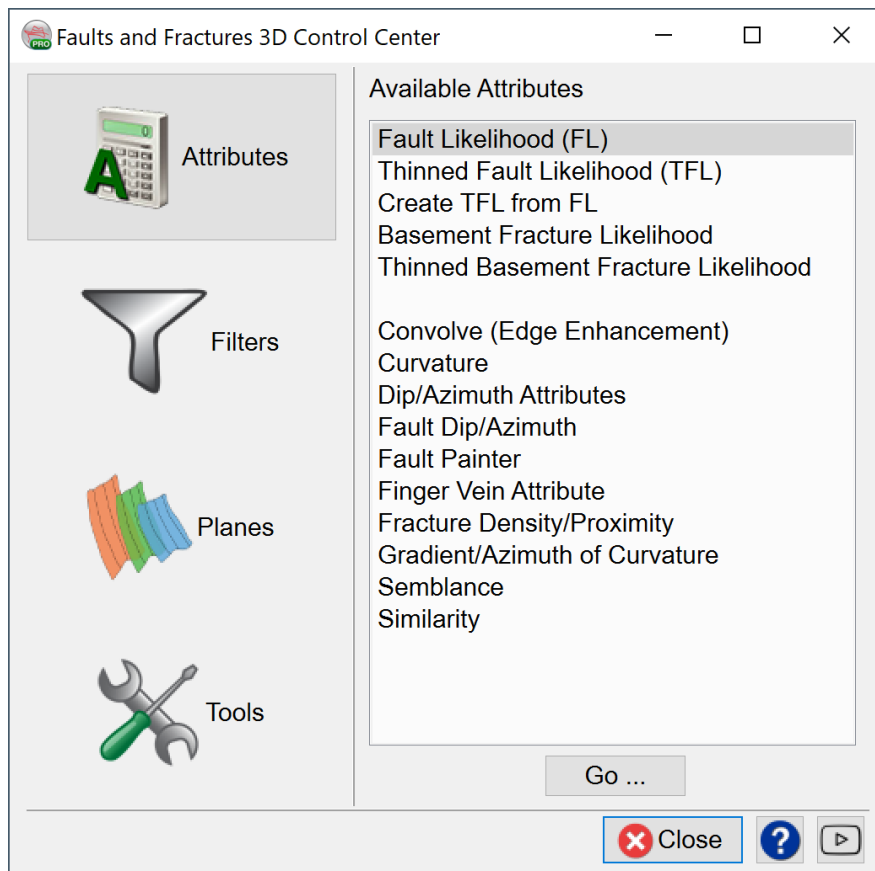
Above is an example formula. On clicking *Set*, you can then select which element to use for names typed in the formula bar. Constants can be designated the letter 'c' with an optional number as the suffix, e.g. c1, c2 etc.

For more detailed information on the Input Parameters, operators, other options and expressions of examples, please take a look at the [OpendTect User Documentation entry for the Mathematics Attribute](#).

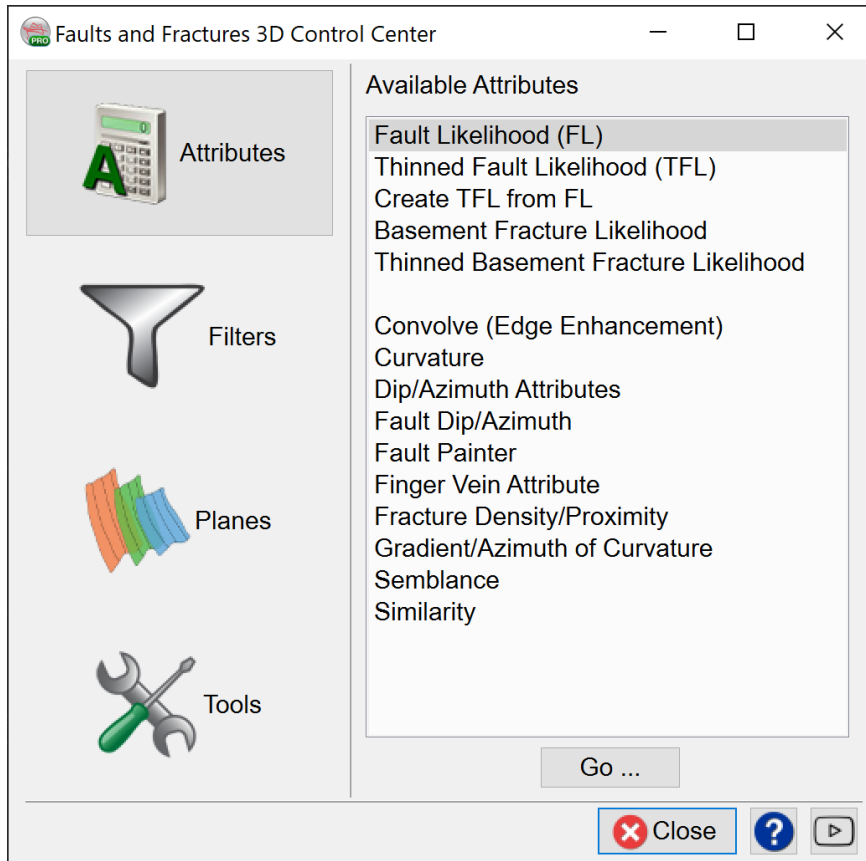
2 Faults and Fractures



2.1 Attributes




2.1.1 Fault Likelihood (FL)

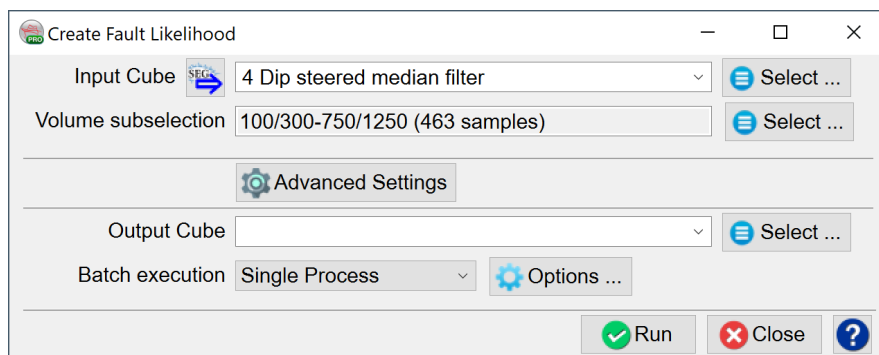


Fault Likelihood is the unthinned version of our Thinned Fault Likelihood volume, which is used as an input to the [Fault Extractor](#) and [Thinned Fault Likelihood](#).

Workflow

To process the Fault Likelihood volume, click on this icon  and select Fault Likelihood under Attributes.

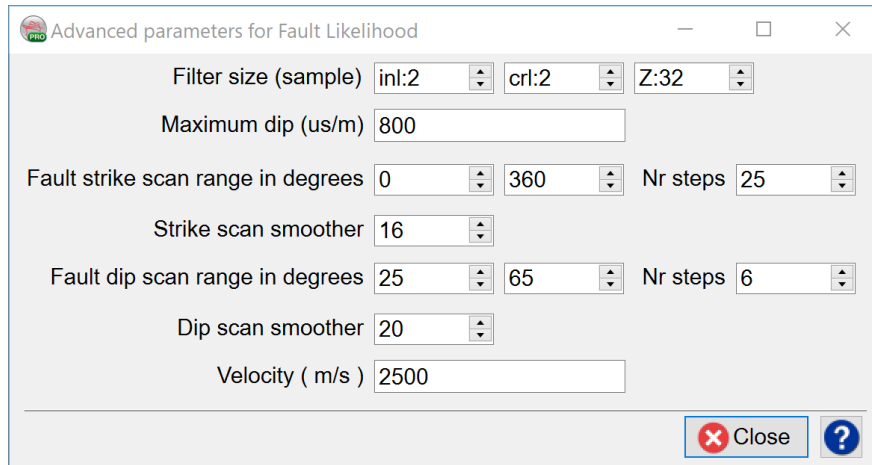
Parameters



Input Volume: Normally the standard poststack seismic volume is a good default input for computing TFL. In case the seismic is extremely noisy, we suggest to first filter the raw seismic using our *structurally oriented median filter*, and then use the filtered seismic as input for TFL. Furthermore, if you have a *color inverted seismic* or *spectrally blued seismic* volume, then using that as input is recommended over the standard seismic volume.

Volume subselection: this is useful if you the user want to speed the process, a subselection area can be set.

Advanced settings



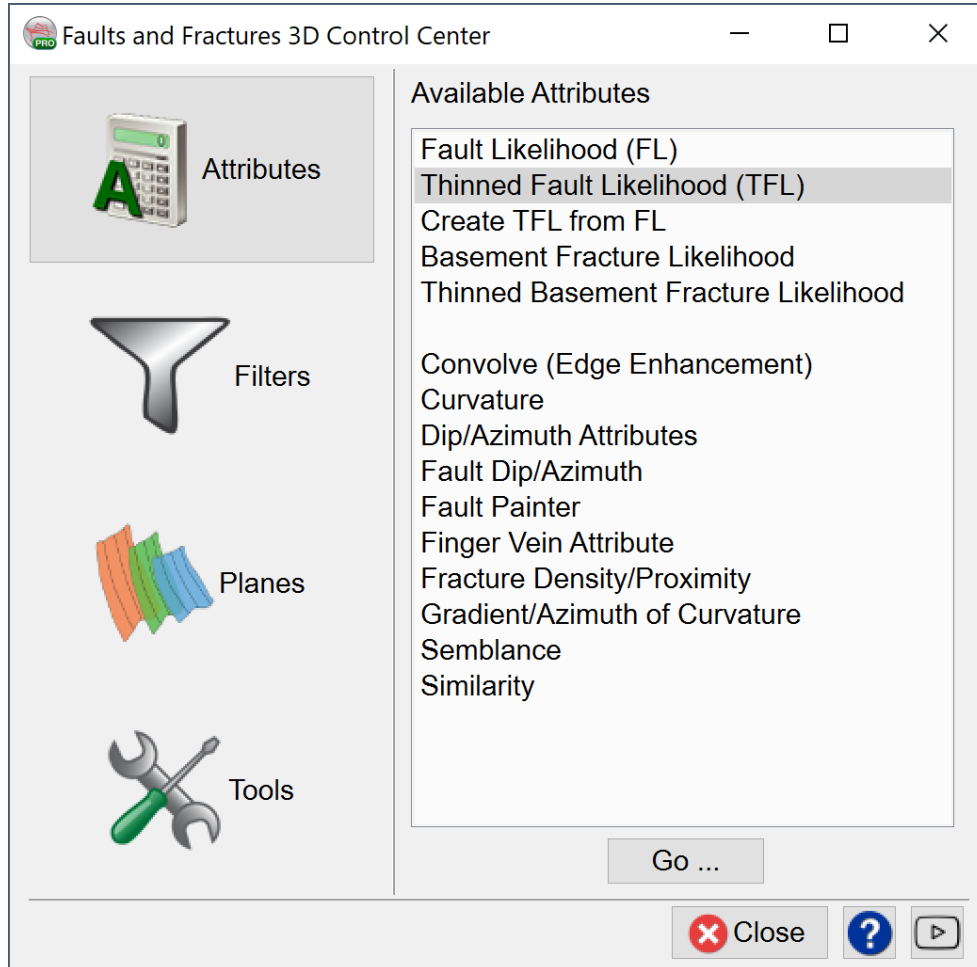
Filter Size (inl/crl/sample): Smaller step-outs used in FL dip computation, e.g. 1-1-16 or 2-2-32, capture small scale features. However, bigger FL dip computation step-outs e.g. 4-4-100, significantly improve the characterization of the bigger fault lineaments by improving their connectivity.

Fault likelihood scan setting: The algorithm will scan for the fault dips and strikes within a given range of the dips to maximize the fault likelihood. The input parameters here depend on how flat/steep the seismic data is. If you are dealing with a layer cake data set, small angle range would be reasonable. Nr Steps is how many scans should be done. Initially the full strike (0 - 360 degrees) and a large dip range (25 - 65 degrees) should be tested. Afterwards, better (more realistic) results can be achieved by constraining the strike/dip range. As a rule of thumb, it is always recommended to use wider dips and strikes to extract more information from the data.

Velocity (m/s) Velocity needs to be used in time surveys to convert angles. Use an average value for your data set, it will not affect the results much.

Apply thinning: The user should select yes in order to output the Thinned Fault Likelihood.

2.1.2 Thinned Fault Likelihood (TFL)



Thinned Fault likelihood or TFL is a revolutionary new attribute that produces accurate and sharp faults. The thinned Fault likelihood attribute which is defined as a power of semblance, has a range of value between 0 and 1. This attribute is developed aiming to capture and delineate faults and fractures in area of interest, such that the algorithm scans the range of fault dips to identify maximum likelihood. The Thinned fault likelihood attribute provides razor-sharp fault images on horizontal slices as well as on vertical sections (*Figure 1*). Fault likelihood attribute is used as input to directly compute

density of fractures in a given area and the proximity of fractures with each other (*Figure 2*). This new plugin also comes with a new, edge-preserving smoothing filter. This smoothing algorithm uses Thinned Fault Likelihood attribute as input. It generates seismic volumes with razor-sharp edges that are very well suited for structural interpretation (*Figure 3*). For further quantitative work milder filters such as the dip-steered median filter are needed.

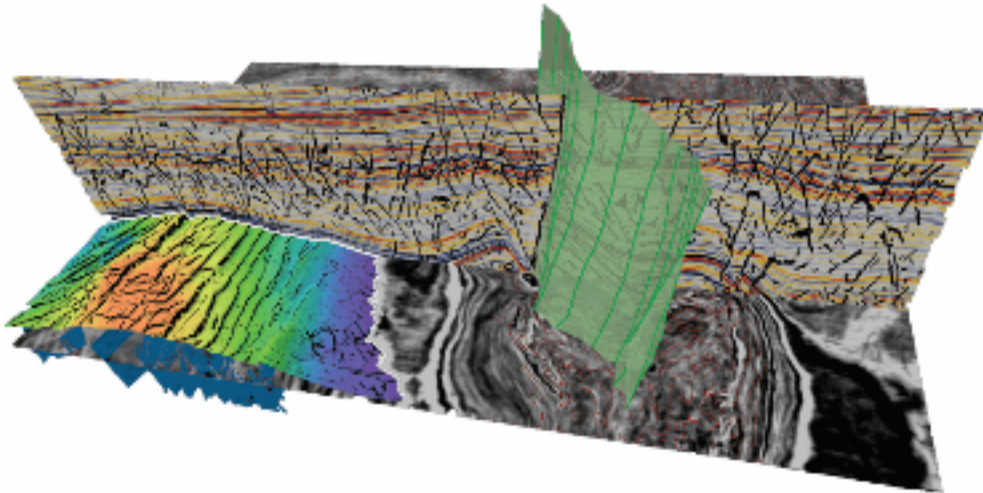


Figure 1. Smoothed seismic with Thinned Fault Likelihood attribute overlay.

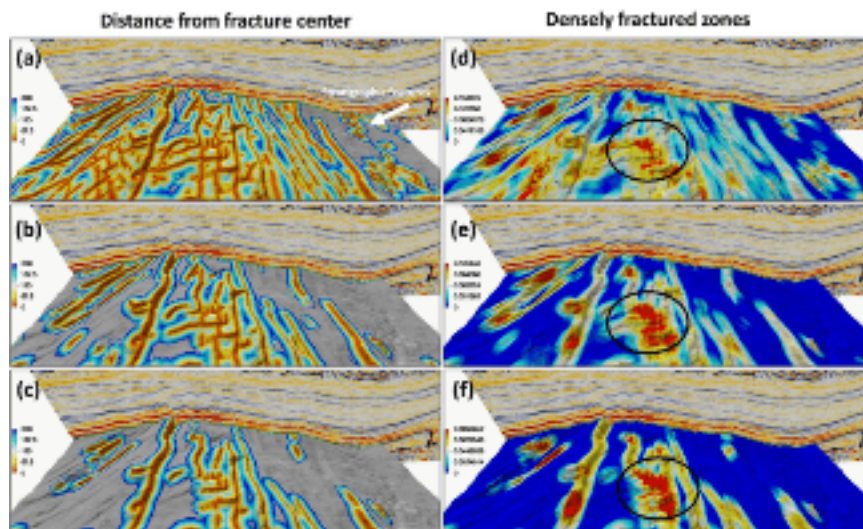


Figure 2. Fracture Density (right) and Fracture Proximity (left) attributes computed from Thinned Fault Likelihood attribute.

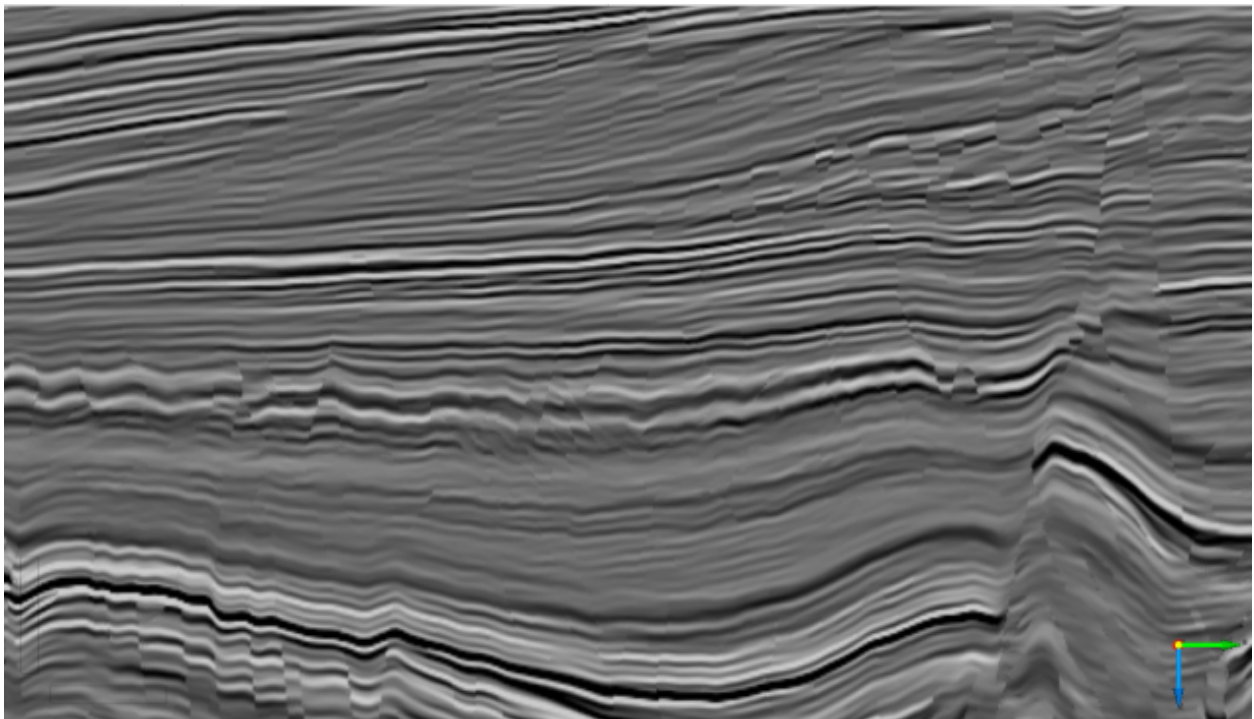
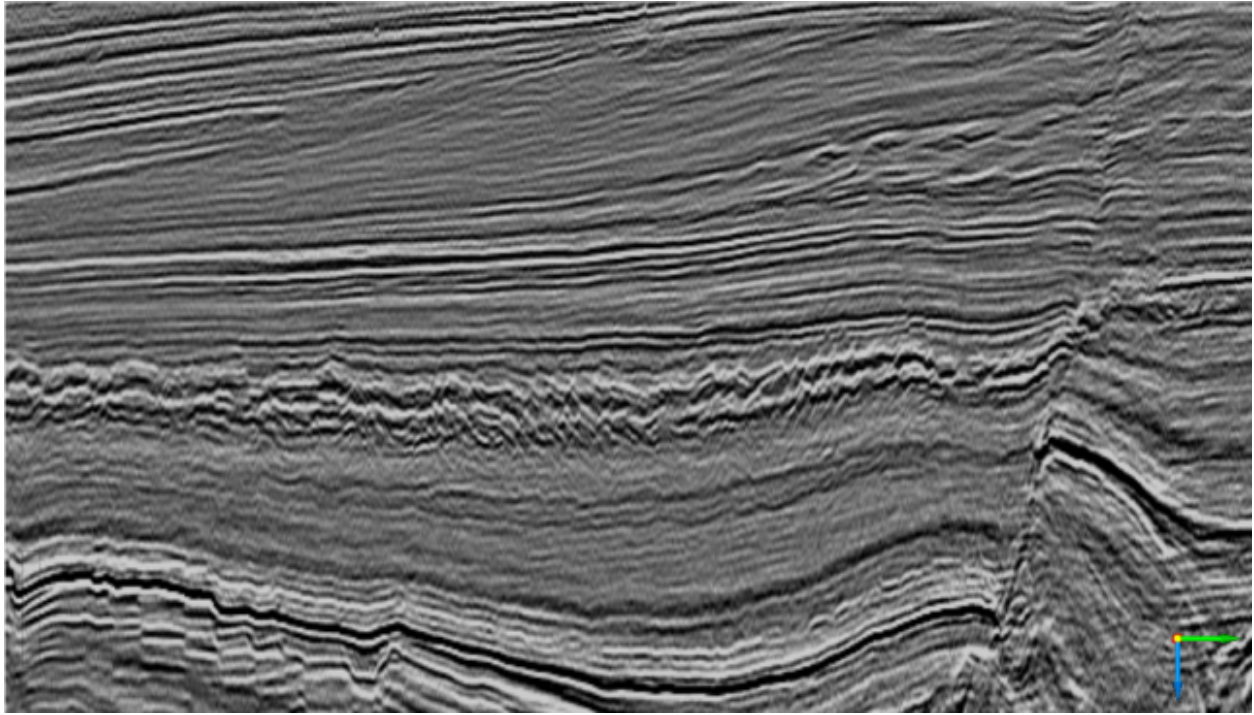

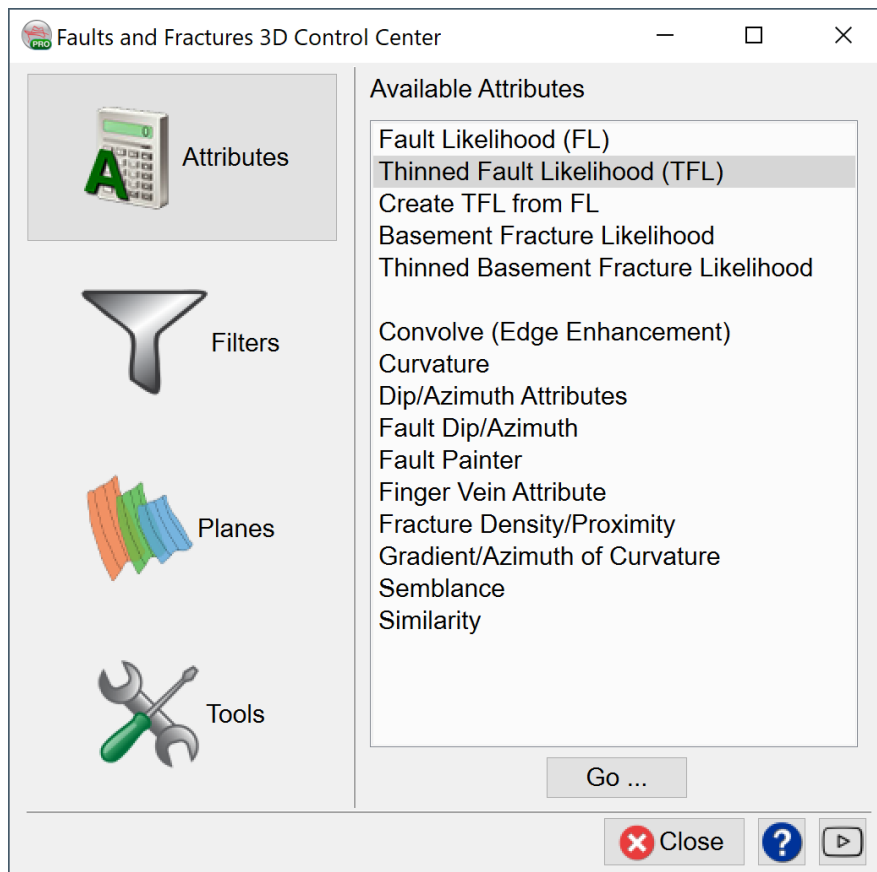


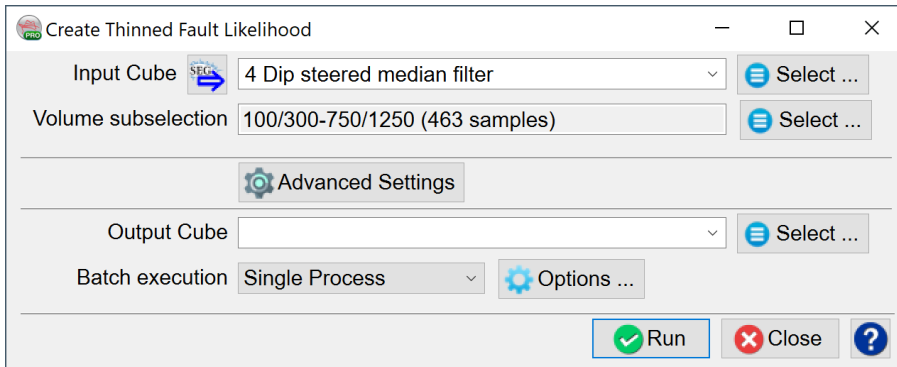
Figure 3. Result of the new smoothing filter (first grey image above) on seismic data (second grey image above). North Sea example

Workflow

To access the fault likelihood plugin, click on this icon  .



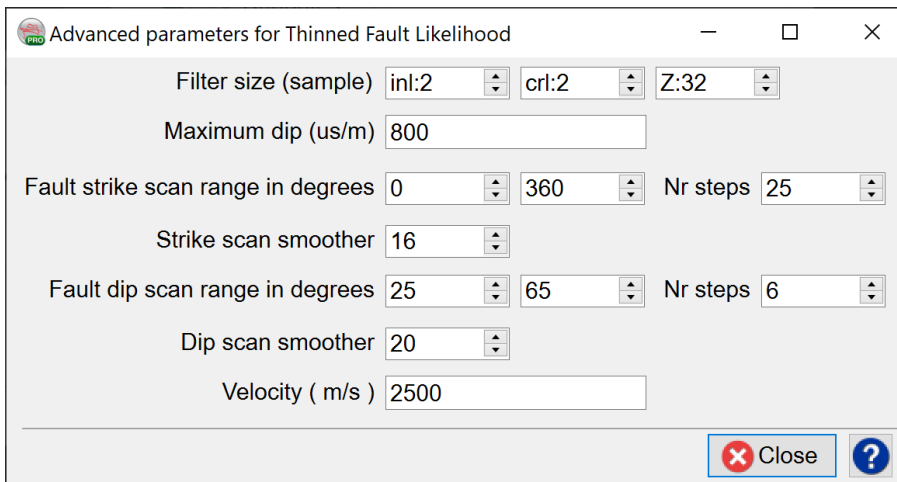
Parameters



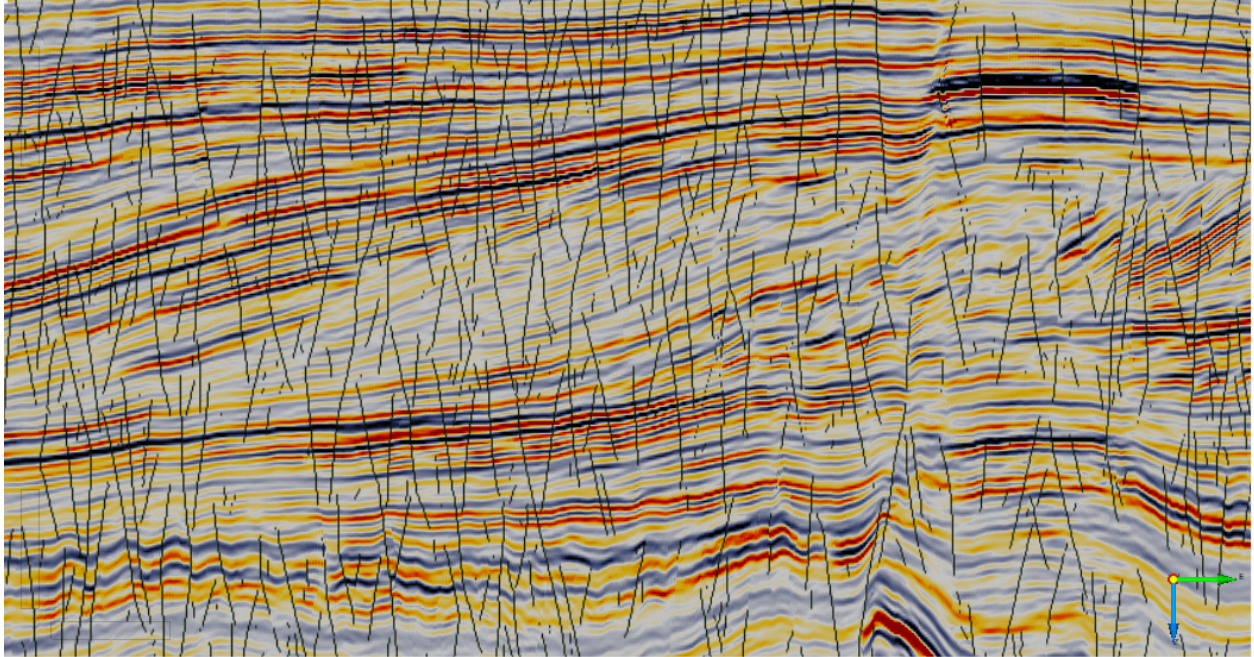
Input Fault Likelihood Cube: Select or Create a [Fault Likelihood Volume](#).

Volume subselection: this is useful if you the user want to speed the process, a subselection area can be set.

Advanced Settings

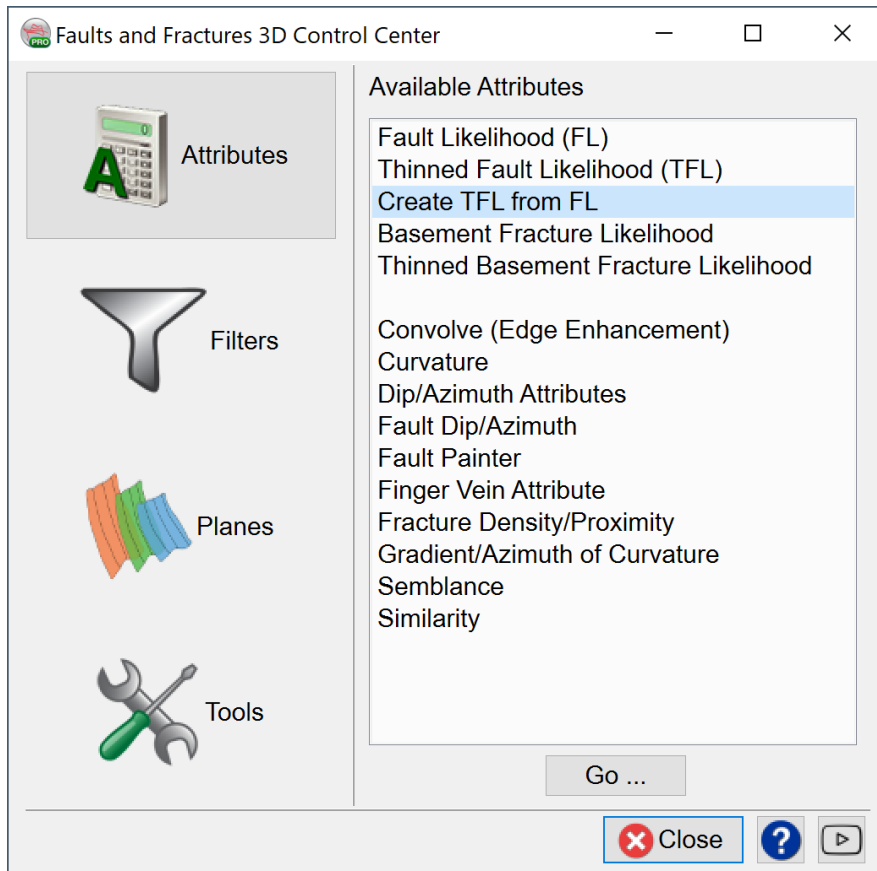


Final results can look like the image below.

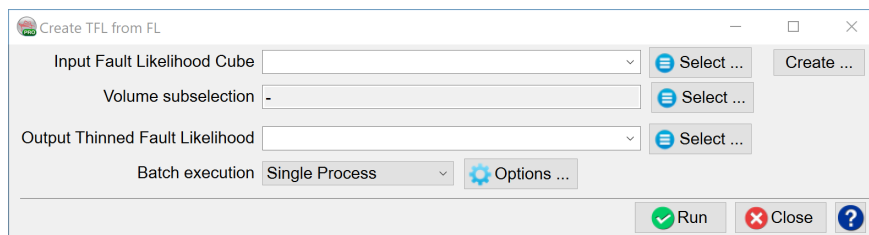


Fault likelihood results on IL 250, F3 dataset, North Sea.

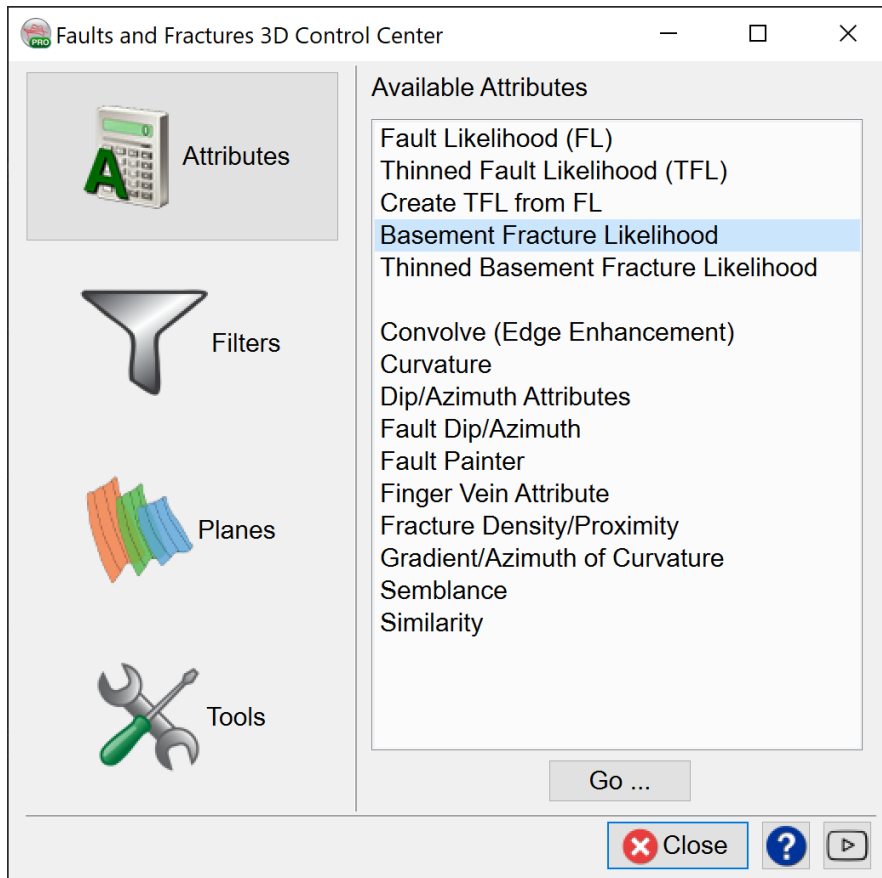
2.1.3 Create TFL from FL



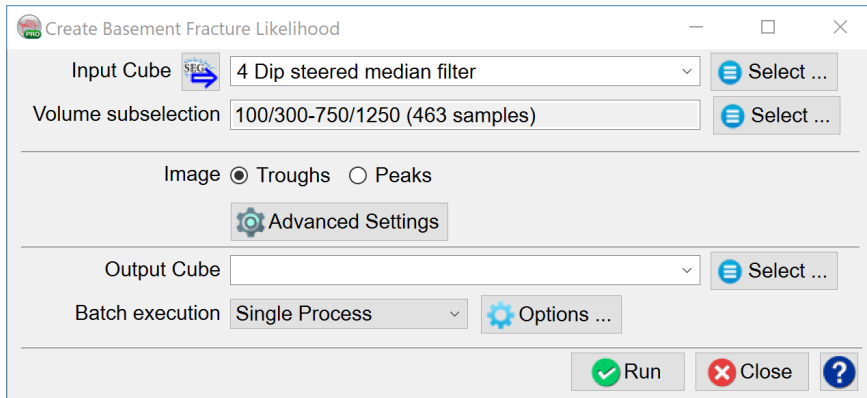
This option allows for easy creation of a TFL volume from an existing FL volume.



2.1.4 Basement Fracture Likelihood (BFL)



The Basement Fracture attribute attempts to track steeply dipping reflections (peaks/troughs) originating from typically soft fracture anomalies in an otherwise tight metamorphic/igneous basement rock.

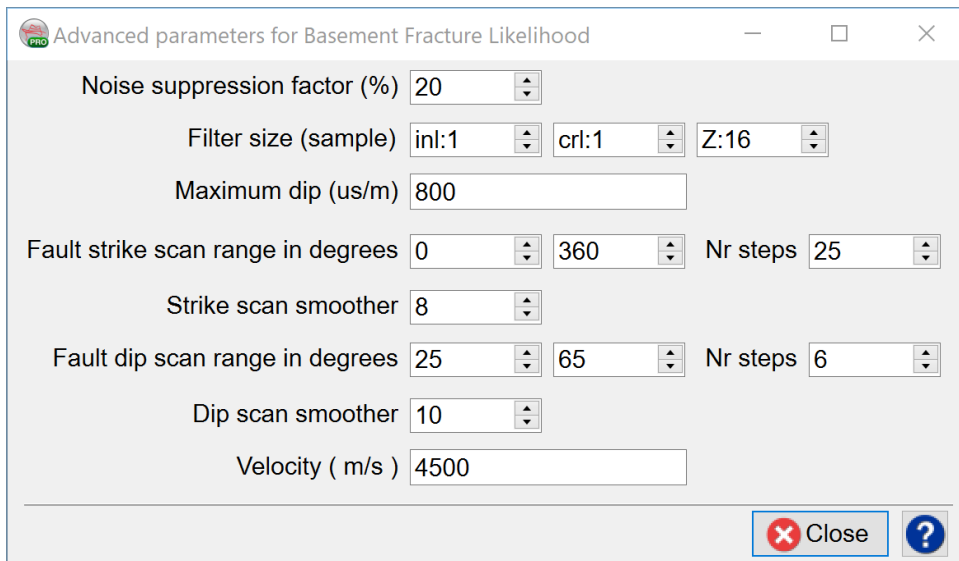


Parameterization

Image

Specify either troughs or peaks to be imaged/tracked seismic. For example, if the seismic data is with European polarity (increase in impedance = seismic trough), the low AI fracture in high AI basement rock would produce a seismic peak.

Advanced settings



Noise Suppression Factor

The higher the value is the less noise the attribute image will contain.

Dip Calculation Stepout

Usually, the default values of 1-1-16 works well for Dip Calculation Stepout for a typical 25 m x 25 m x 4 ms survey sampling. For smaller bin sizes it is recommended to increase the stepouts.

Fault Strike Scan range

The 0 deg reference line is along the Cross-line of the survey (not true north coordinate) from smaller In-line values to larger In-line values

Fault Dip Scan Range

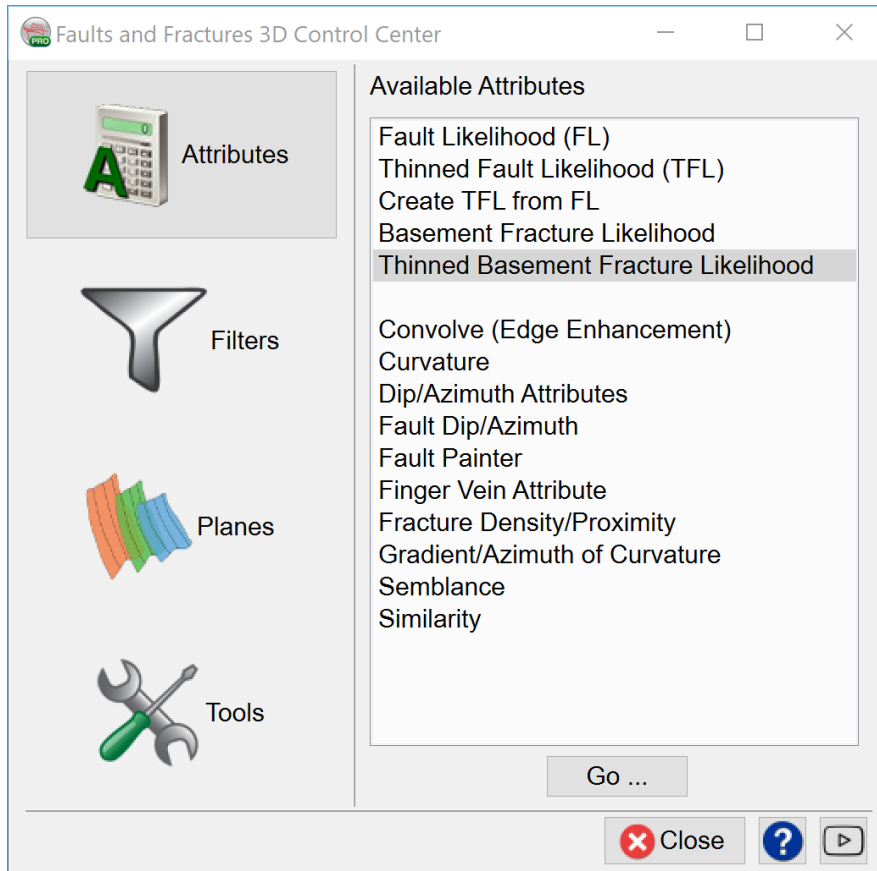
The 0 deg reference line is along the In-line of the survey from smaller Cross-line values to larger Cross-line values.

Velocity

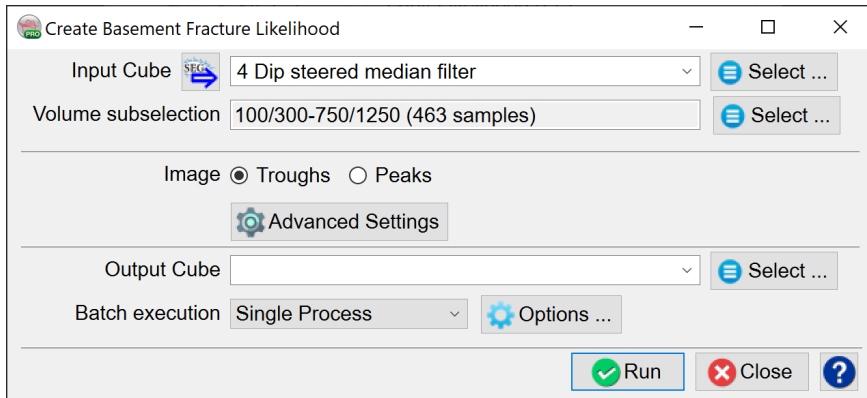
Needs to be used in time surveys to convert angles.

Two filtering algorithms are available: the Deriche filter is applied for standard filter sizes (below 32), and the Van Vliet filter for bigger sizes ($\sigma = 32$ and higher). The runtime, for both filters, is independent of σ : Running it with $\sigma = 1$ takes as much time as with $\sigma = 20$. For example a processing done with filter size [2, 6, 40] will first apply a Van Vliet filter of $\sigma = 40$ in the vertical direction. Then a Deriche filter of $\sigma = 6$ is applied to the previous output in the crossline direction. Finally a Deriche filter of $\sigma = 2$ is applied in the inline direction to the output.

2.1.5 Thinned Basement Fault Likelihood (TBFL)

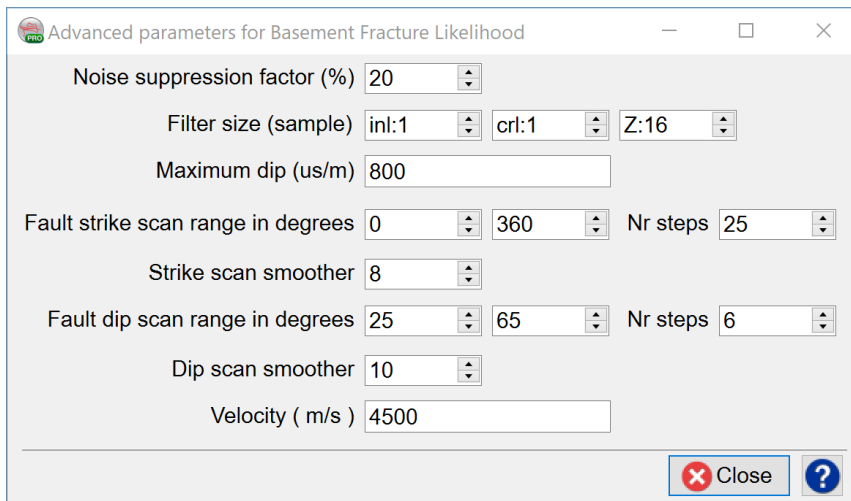


Parameters



Specify either troughs or peaks to be imaged/tracked seismic. For example, if the seismic data is with European polarity (increase in impedance = seismic trough), the low AI fracture in high AI basement rock would produce a seismic peak.

Advanced Settings



Noise Suppression Factor

The higher the value is the less noise the attribute image will contain.

Dip Calculation Stepout

Usually, the default values of 1-1-16 works well for Dip Calculation Stepout for a typical 25 m x 25 m x 4 ms survey sampling. For smaller bin sizes it is recommended to increase the stepouts.

Fault Strike Scan range

The 0 deg reference line is along the Cross-line of the survey (not true north coordinate) from smaller In-line values to larger In-line values

Fault Dip Scan Range

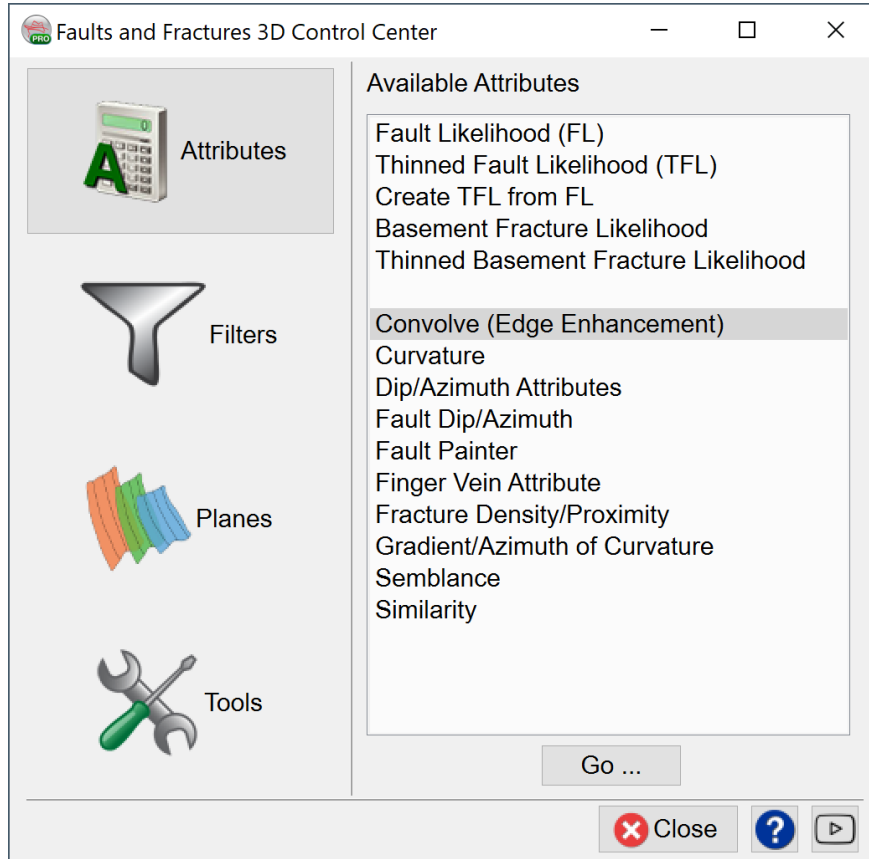
The 0 deg reference line is along the In-line of the survey from smaller Cross-line values to larger Cross-line values.

Velocity

Velocity needs to be used in time surveys to convert angles.

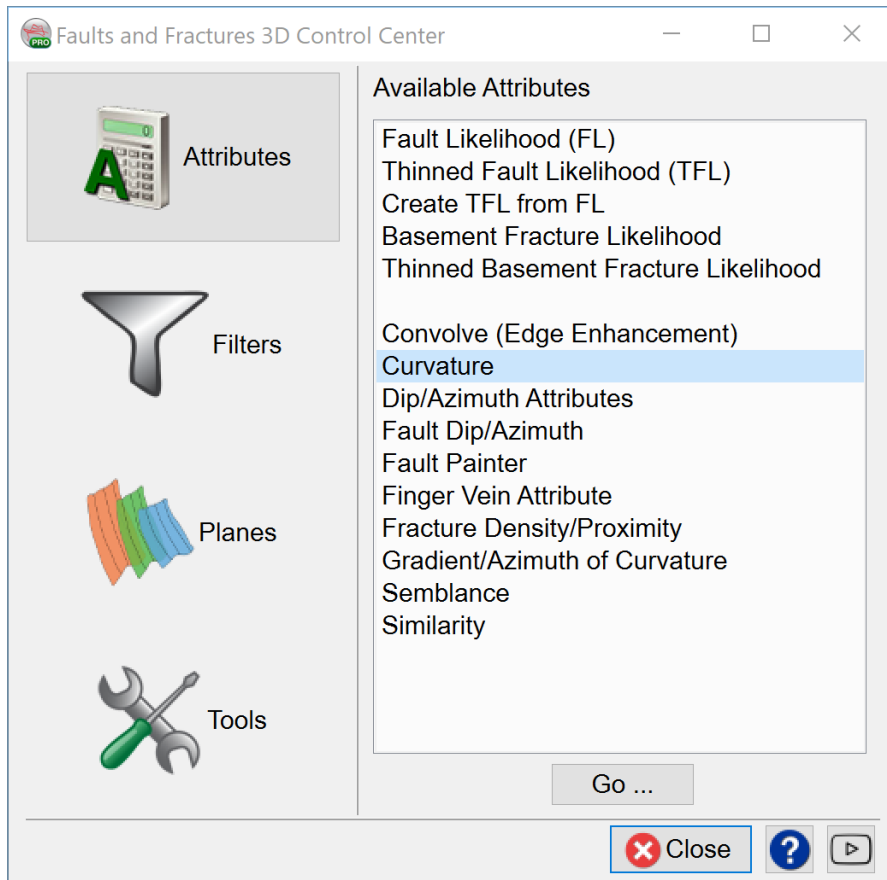
Two filtering algorithms are available: the Deriche filter is applied for standard filter sizes (below 32), and the Van Vliet filter for bigger sizes ($\sigma = 32$ and higher). The runtime, for both filters, is independent of σ : Running it with $\sigma = 1$ takes as much time as with $\sigma = 20$. For example a processing done with filter size [2, 6, 40] will first apply a Van Vliet filter of $\sigma = 40$ in the vertical direction. Then a Deriche filter of $\sigma = 6$ is applied to the previous output in the crossline direction. Finally a Deriche filter of $\sigma = 2$ is applied in the inline direction to the output.

2.1.6 Convolve (Edge Enhancement)



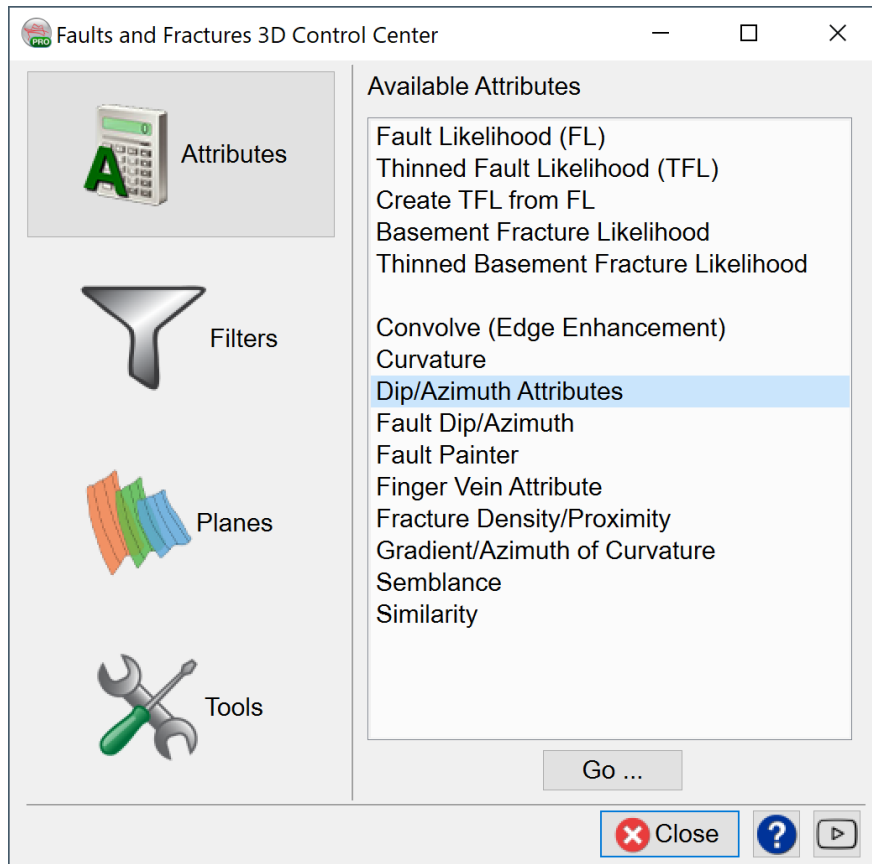
The full article can be found in the user documentation under *11 Appendix A - Attributes and Filters > 11.2 Convolve*.

2.1.7 Curvature



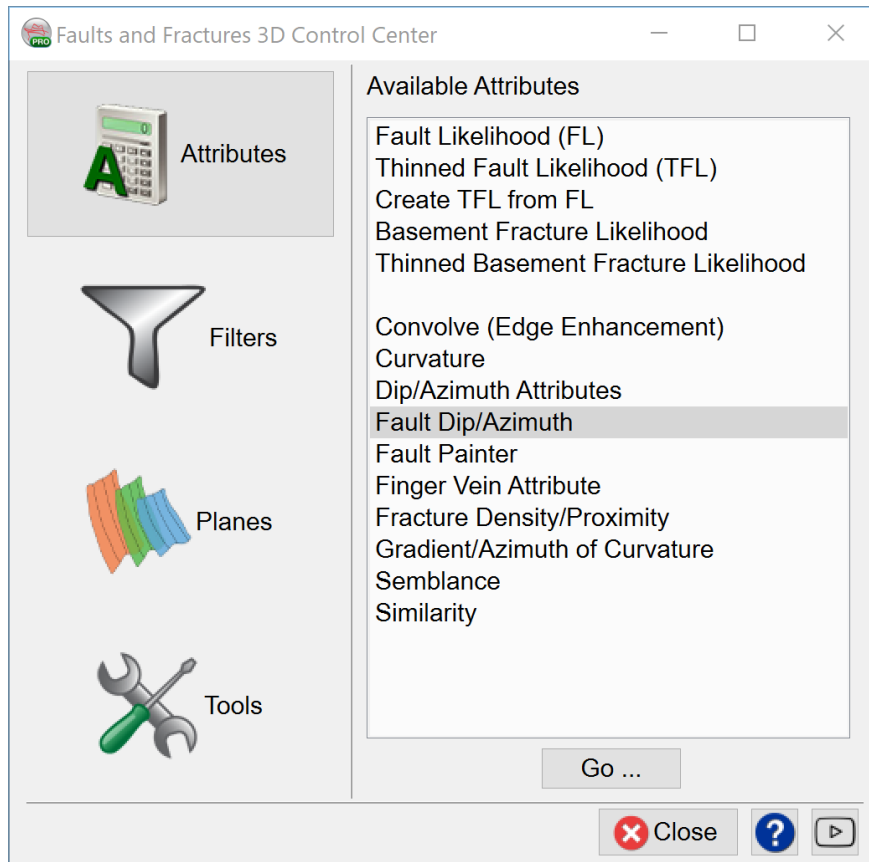
The full article can be found here under *4 Dip-Steering > 4.5 Attributes with Steering > 4.5.1 Curvature Analysis > 4.5.1.1 Volumetric Curvature*.

2.1.8 Dip - Azimuth Attributes



The full article can be found [here](#) under *4 Dip-Steering > 4.5 Attributes with Steering > 4.5.2 Dip*.

2.1.9 Fault Dip - Azimuth

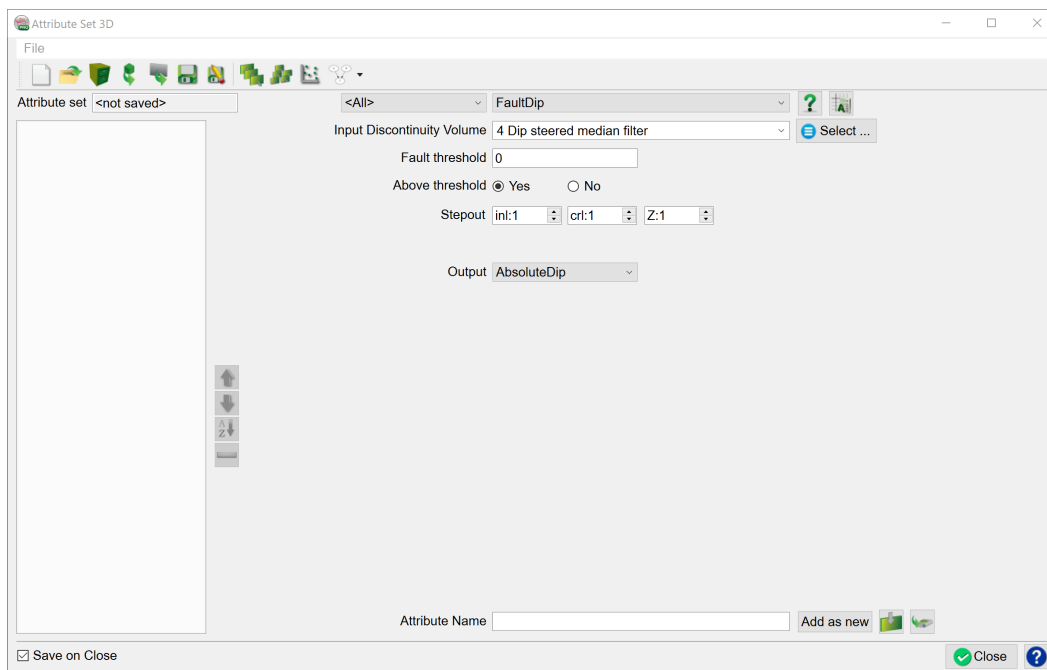


For the fault dip attribute, the input has to be one of the discontinuity volumes, such as Similarity, Thinned fault likelihood, etc. Binary score volume, which a user can output by applying Finger vein algorithm, can be also used an input for the Fault Dip attribute.

Fault threshold: you can put a number between 0, 1.

Step out: if you use 1,1,1 the output will be similar to BG algorithm used when calculating steeringcube. However, if you the user want to optimize the fault, a high Z value is needed. Step out of 1,1,16 is ideal parameters.

Output: Absolute Dip. The Inline Dip, Crl Dip, and the Azimuth can be output in the same way.



Absolute dip: Fault surface intersects with the horizontal angle

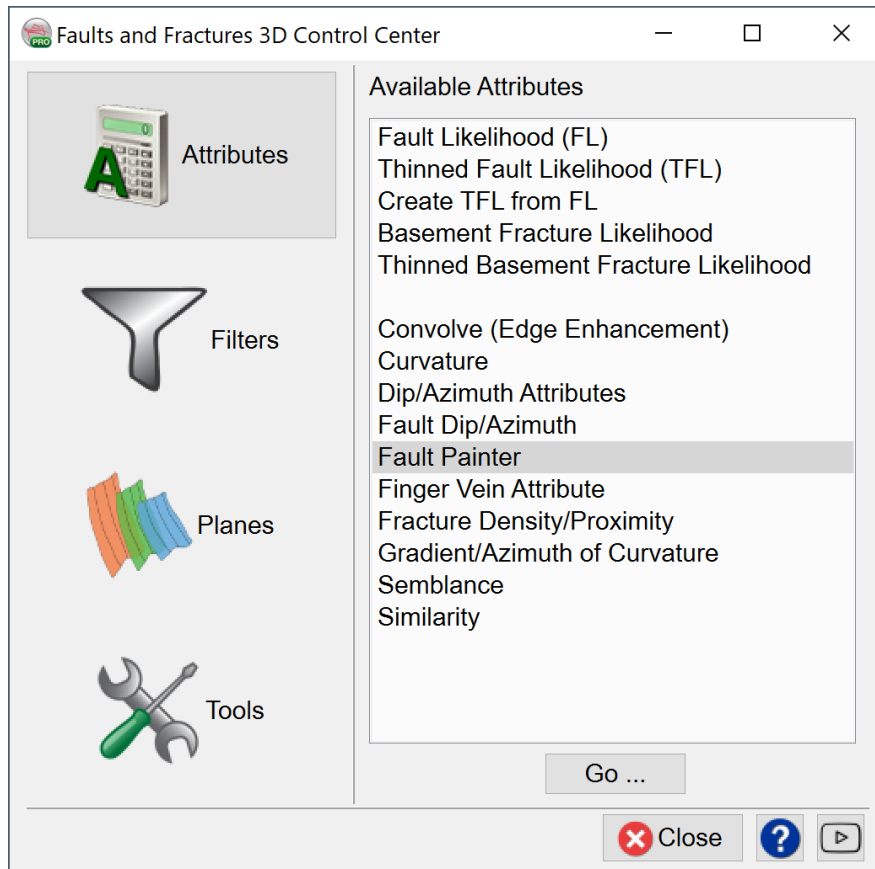
Inline dip: attribute returns the dip along the inline direction as extracted by the steering algorithm. It is the first stored component of the steering cube, in $\mu\text{s}/\text{m}$ or mm/m .

Crossline dip: attribute returns the dip along the crossline direction as extracted by the steering algorithm. It is the second stored component of the steering cube, in $\mu\text{s}/\text{m}$ or mm/m .

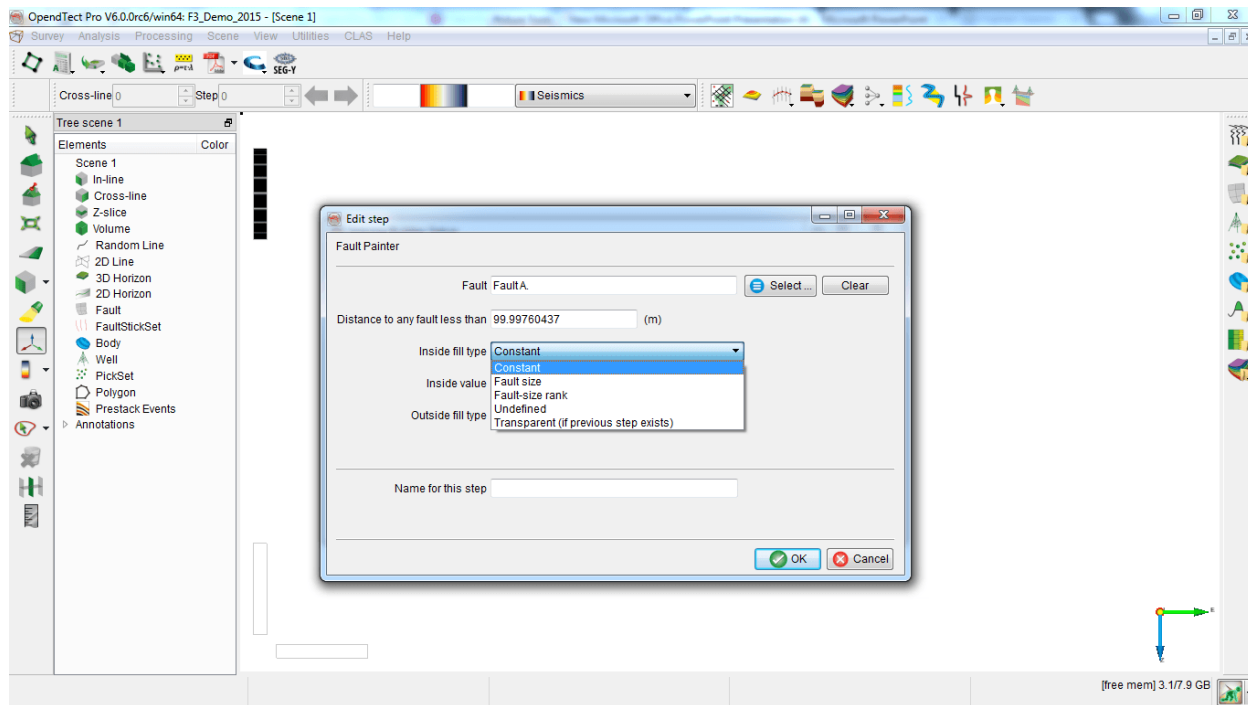
The Azimuth attribute: returns the Azimuth of the dip direction in degrees ranging from -180 to +180. Positive azimuth is defined from the inline in the direction of increasing crossline numbers. Azimuth = 0 indicates that the dip is dipping in the direction of

increasing cross-line numbers. Azimuth = 90 indicates that the dip is dipping in the direction of increasing in-line numbers.

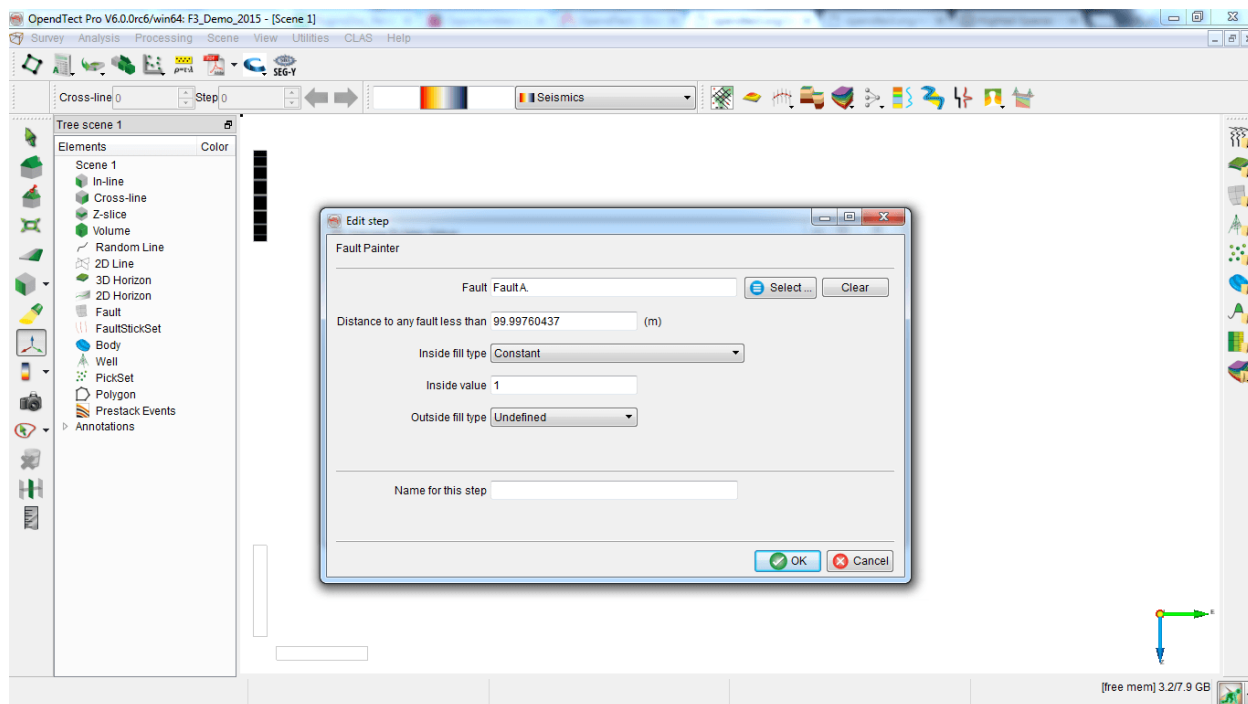
2.1.10 Fault Painter



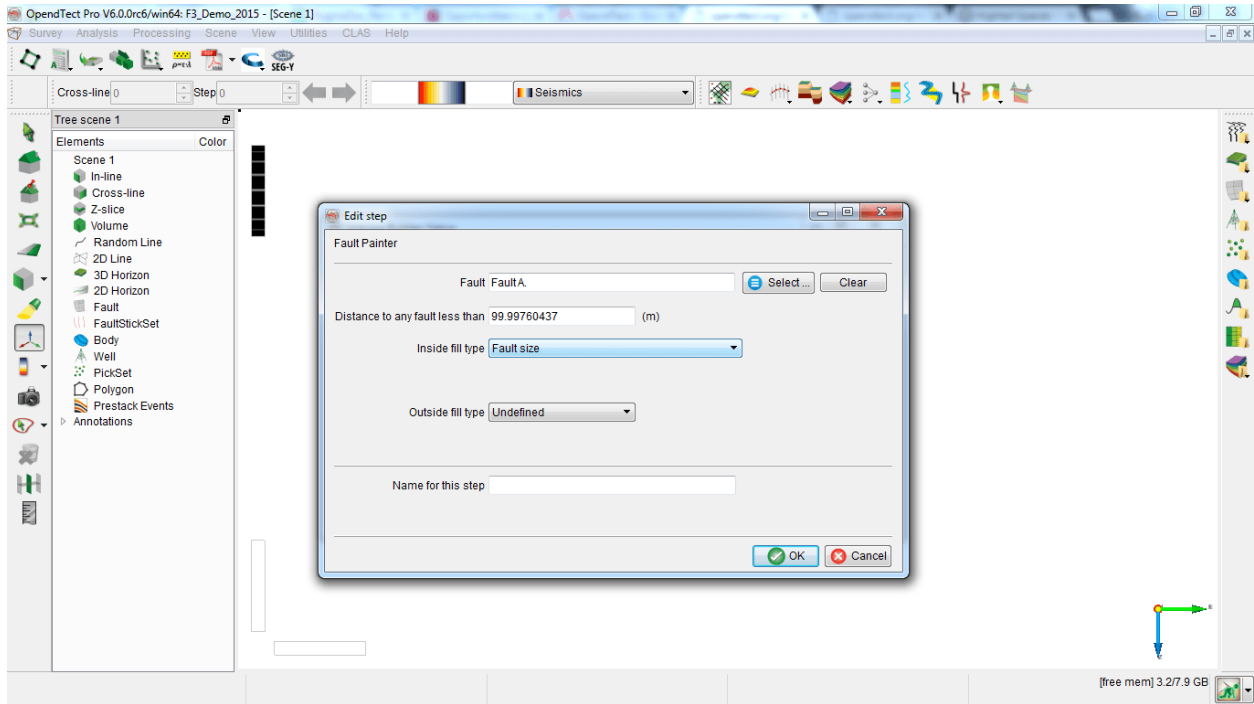
The fault shape painter is used to define the inside and outside values for an OpenTect fault. The options for 'Inside Fill Type' are shown in the image below. The same options are available for 'Outside Fill Type':



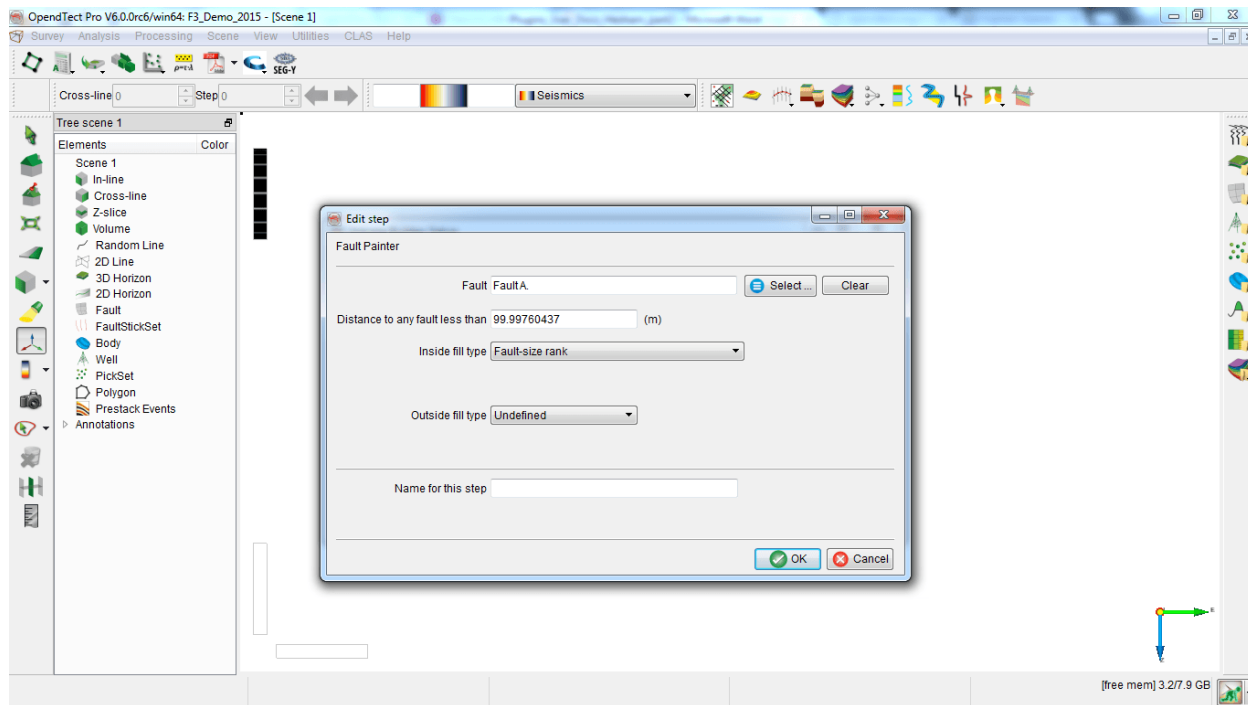
Fill Type I: Constant Value: Single, user-defined value:



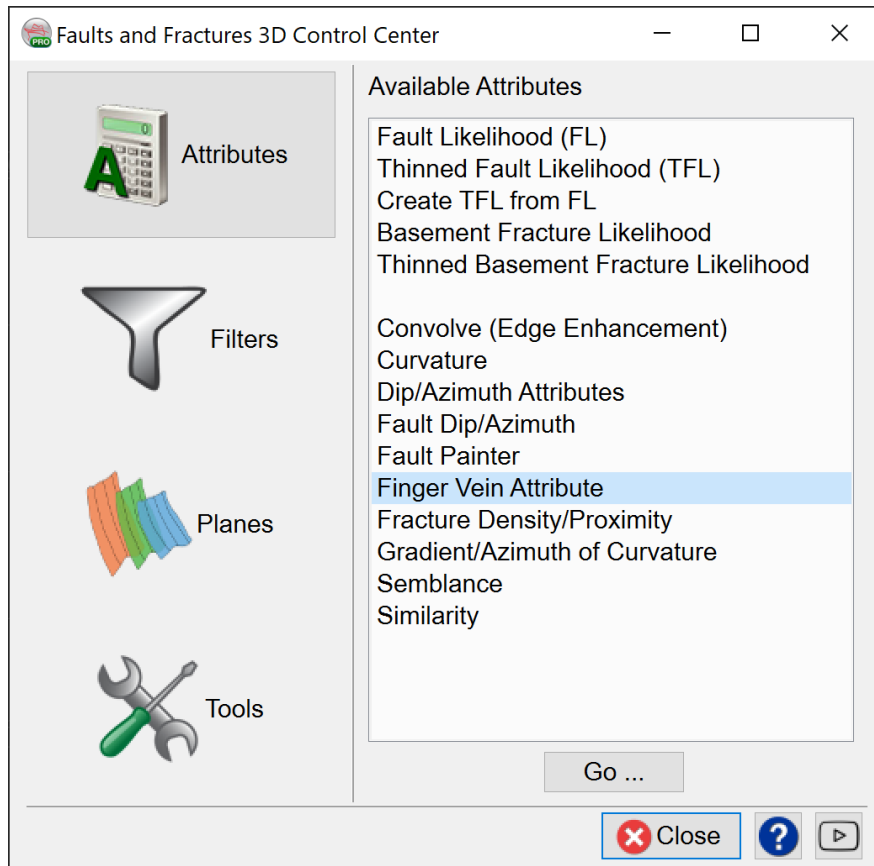
Fill type II: Fault size. This will count the number of position the fault(s) have



Fill type III: Fault size-rank. If you have more than one fault, this option will rank the fault position based on the fault size.

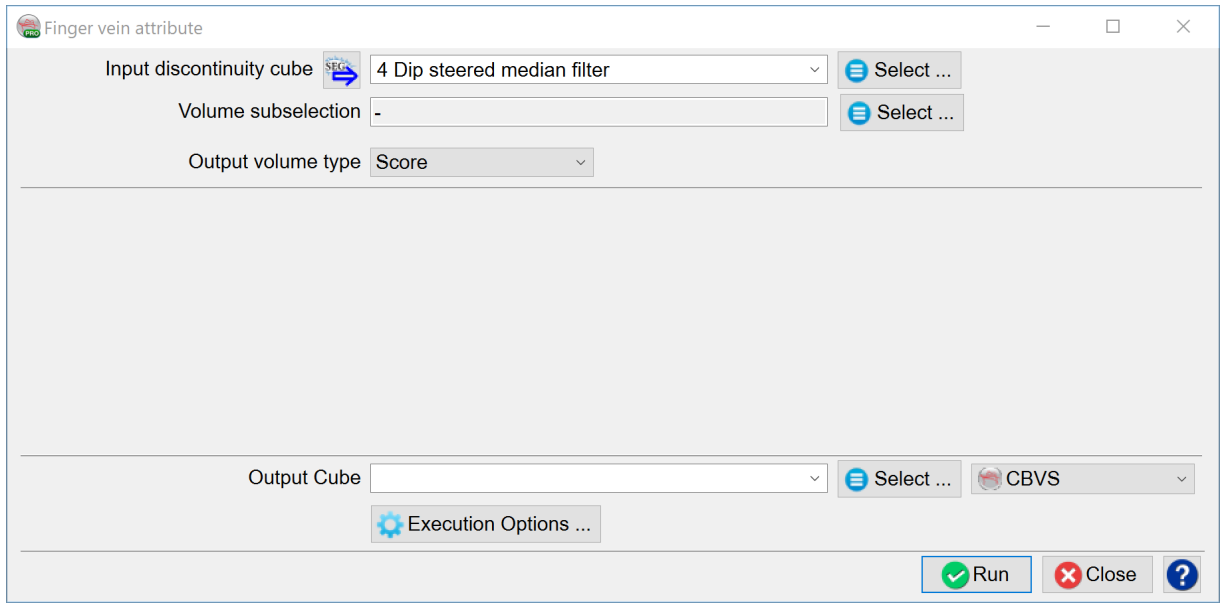


2.1.11 Finger Vein Attribute



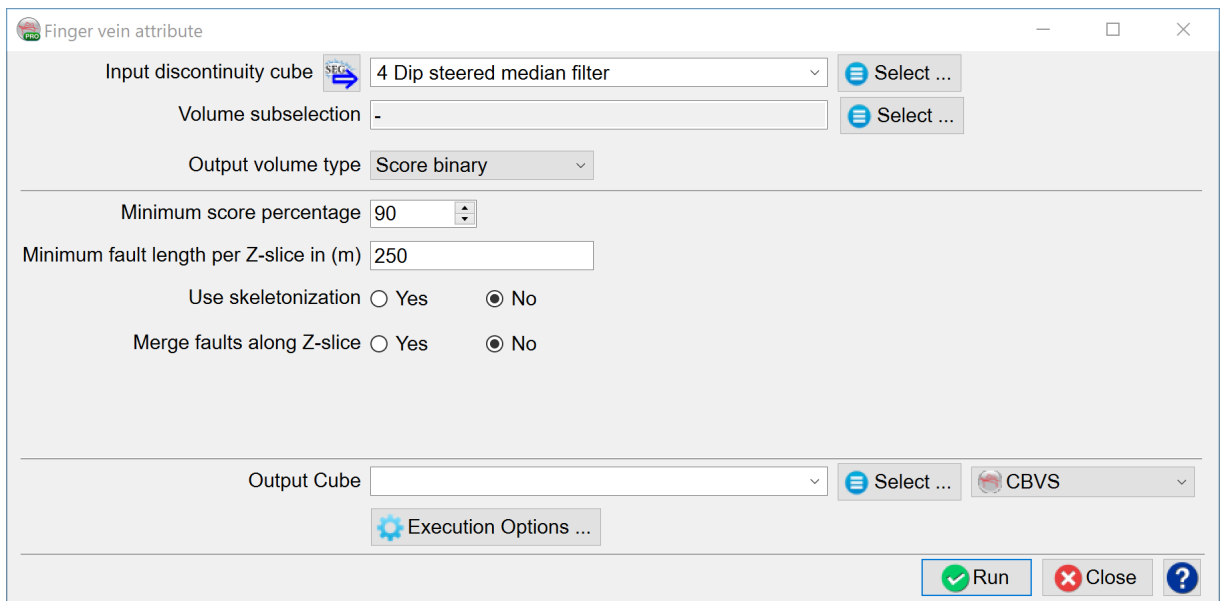
This technique is based on algorithm that automatically tracks faults.

Input: A discontinuity volume with an enhanced fault pattern (e.g. Semblance, Curvature or Similarity).



Output: Score

Score binary:

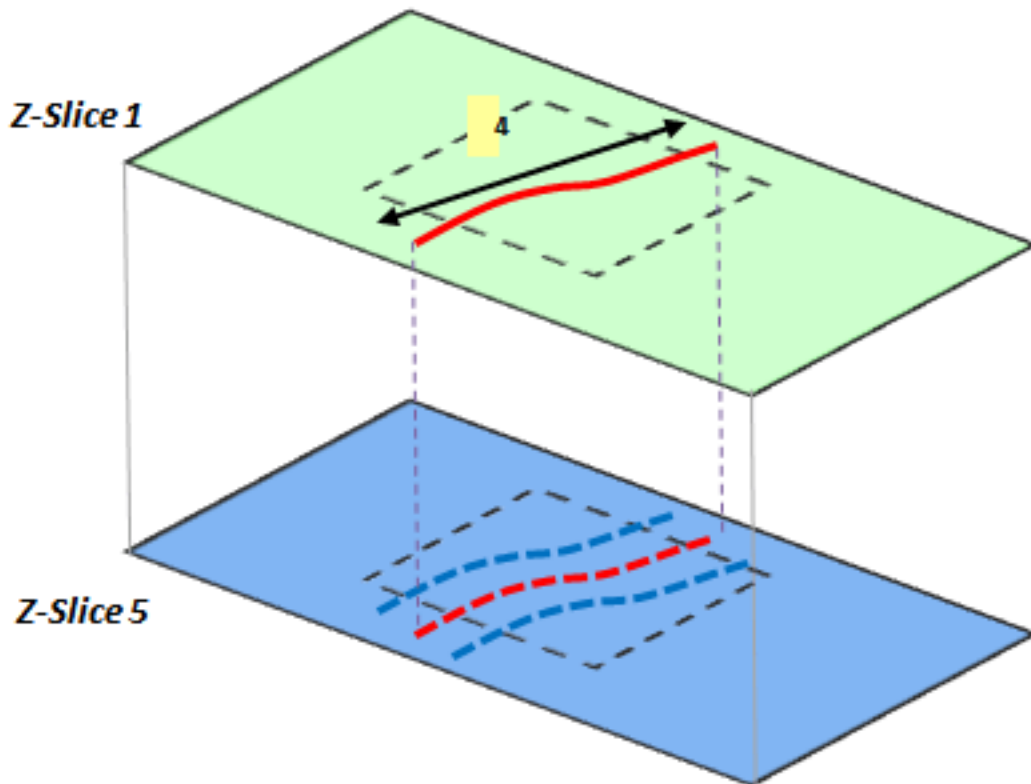


Input: One of the discontinuity volumes (Similarity or TFL)

Output volume type: Score binary. Values will be 0, or 1.

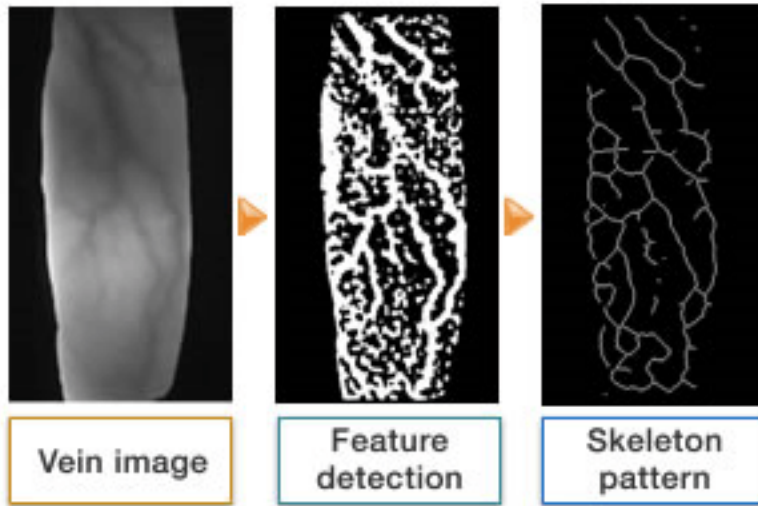
Minimum score percentage: 90

Minimum fault length per Z-slice in (m): is the minimum horizontal length of the fault on any Z-slice. In this example, use 250.



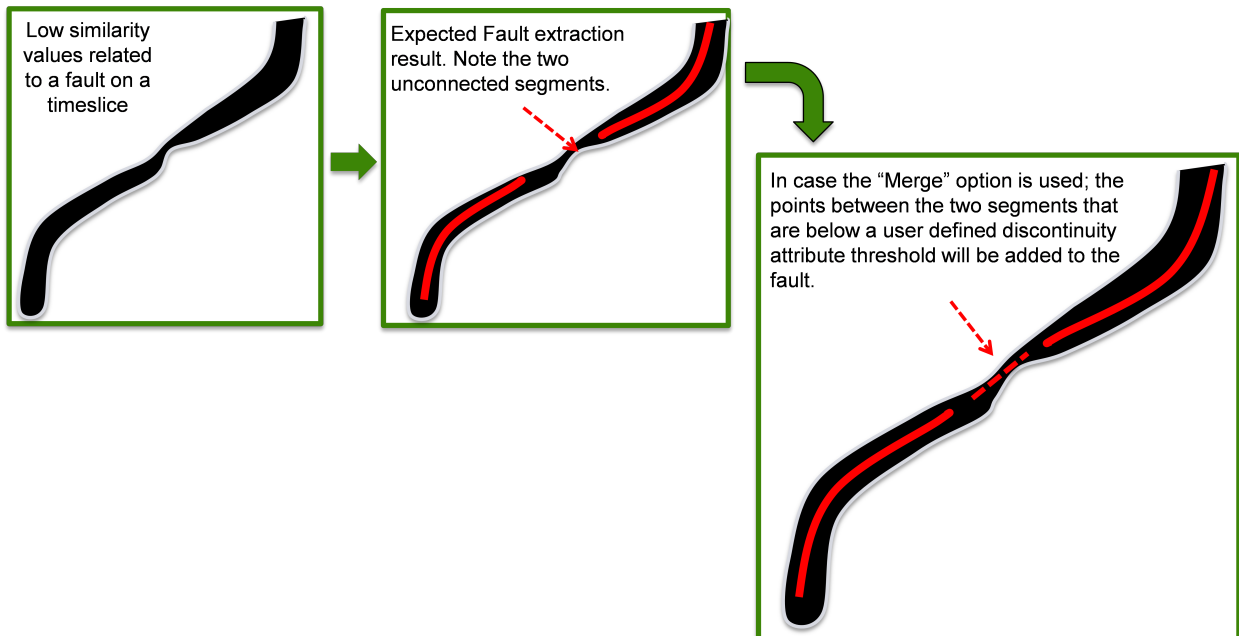
Use Skeletonization: This option will apply thinning.

Finger Vein Pattern Processing Algorithms

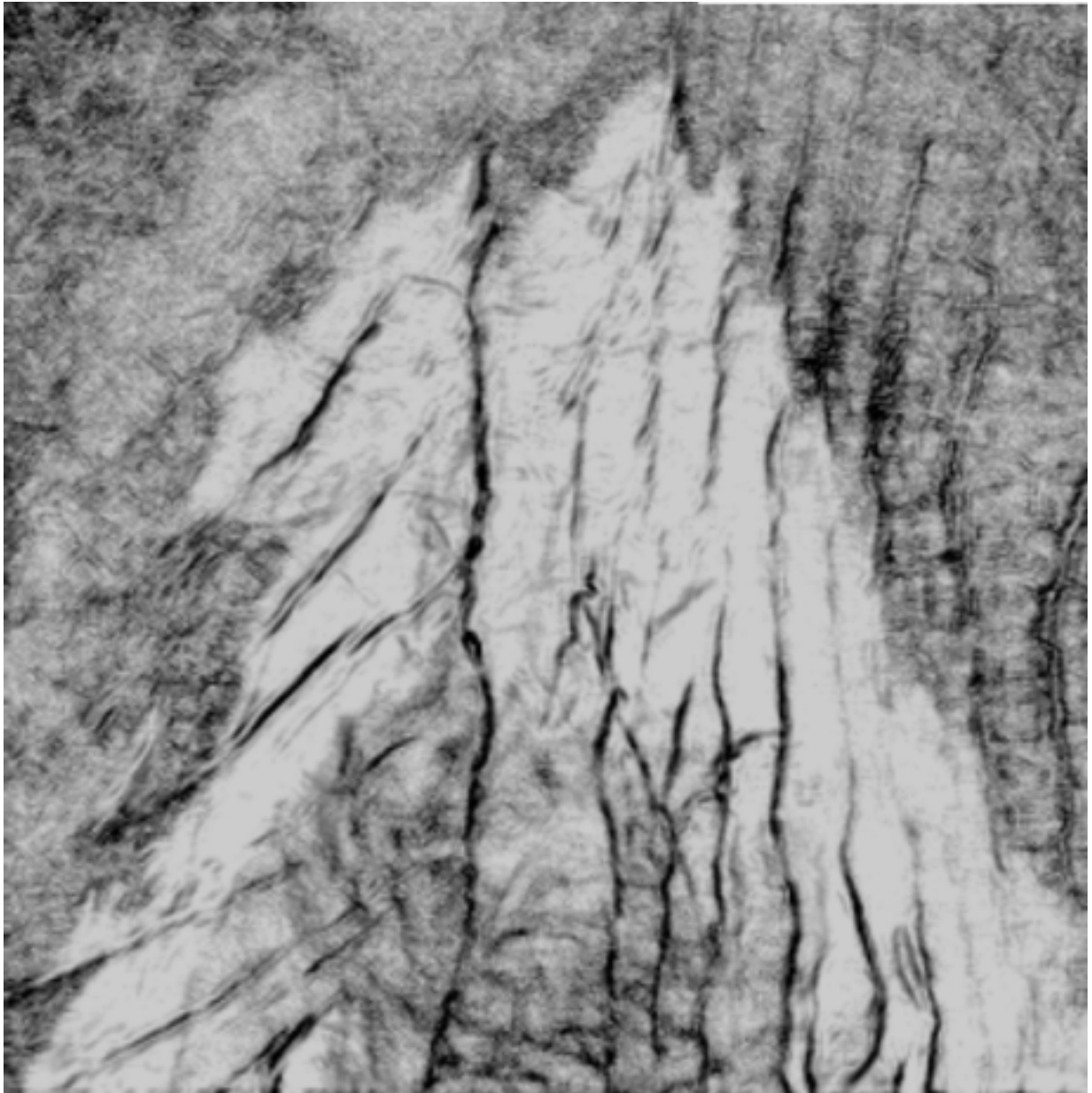


Merge faults along Z –slice: Below threshold

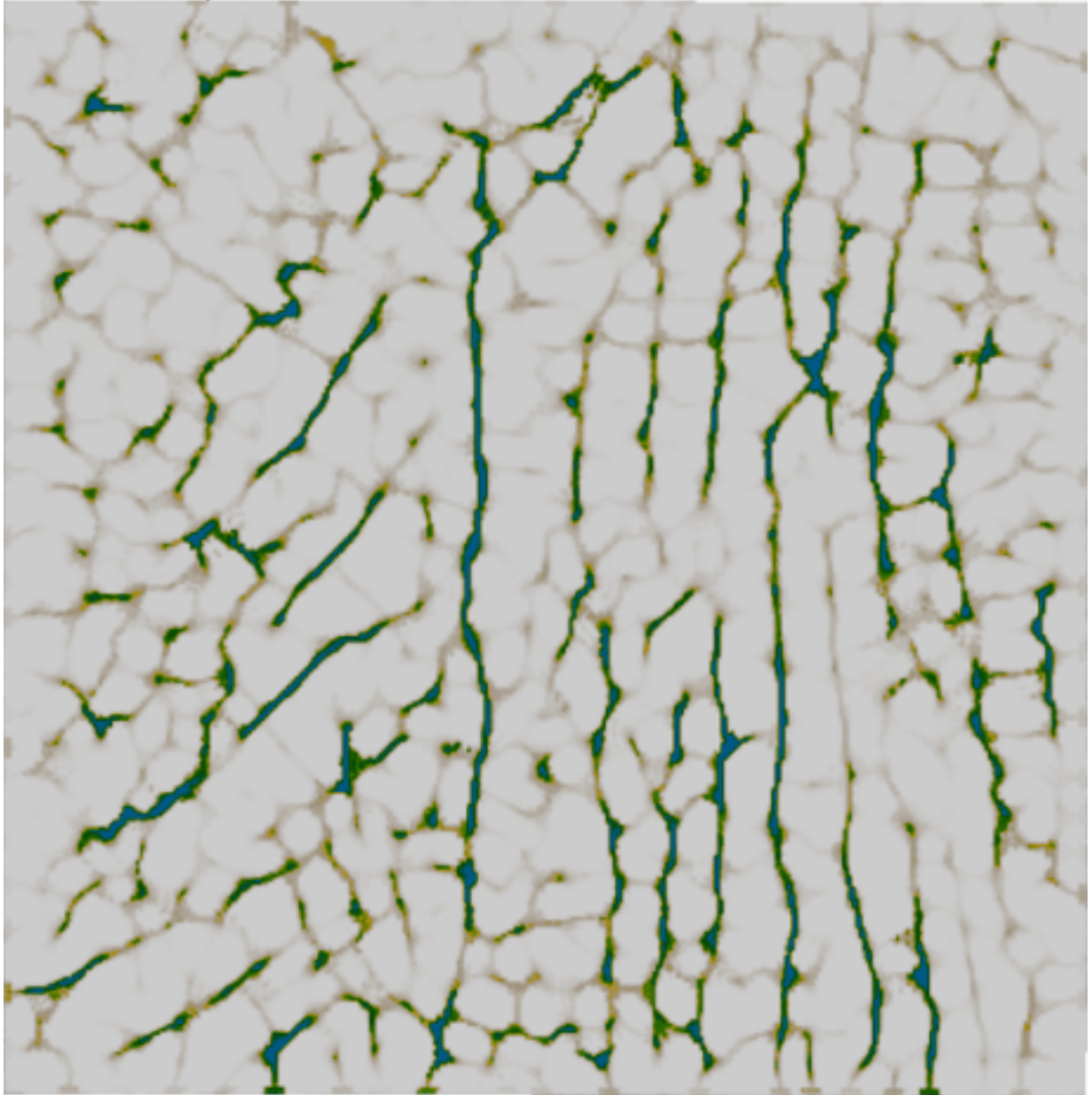
Threshold value: 0.8



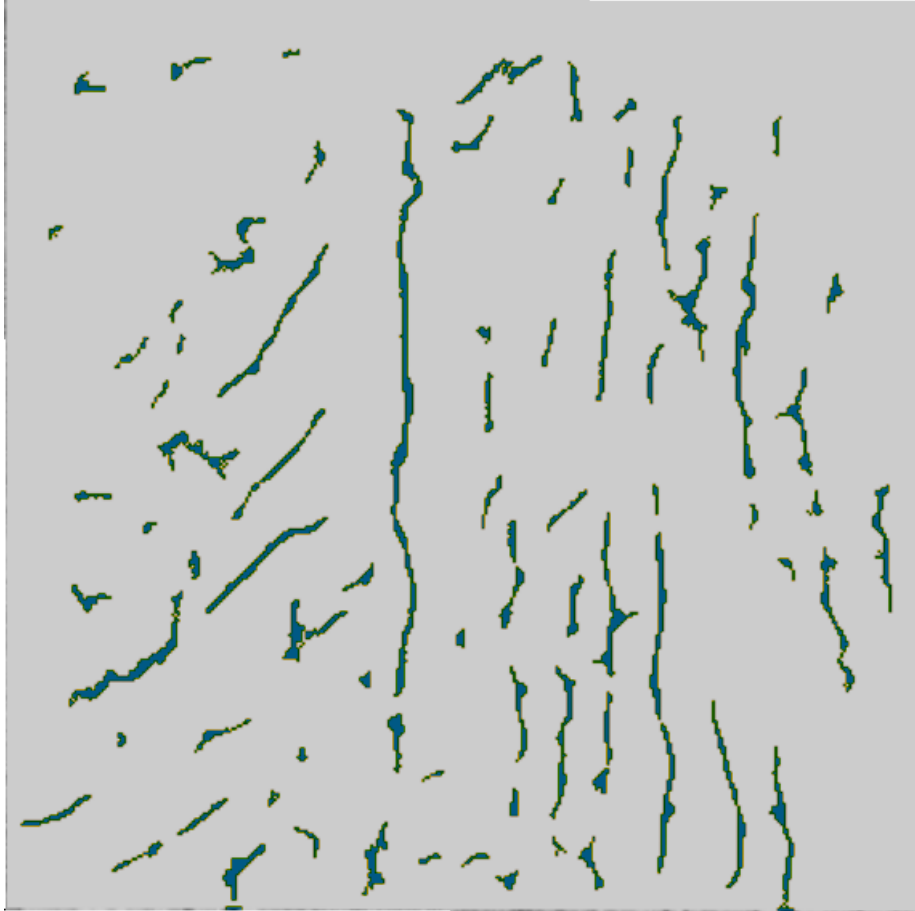
Results should look like this for score and score binary



a) Similarity attribute

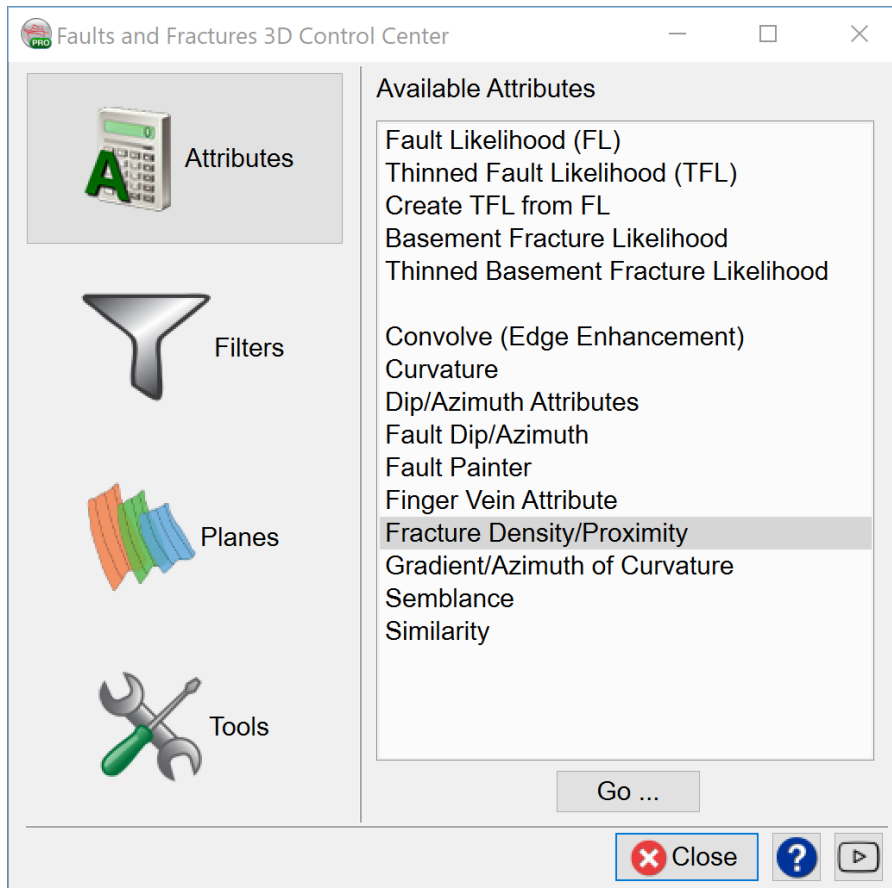


b) Score



c) Score binary

2.1.12 Fracture Density - Proximity



Fracture density attribute is used to highlight the areas of high dense fractures.

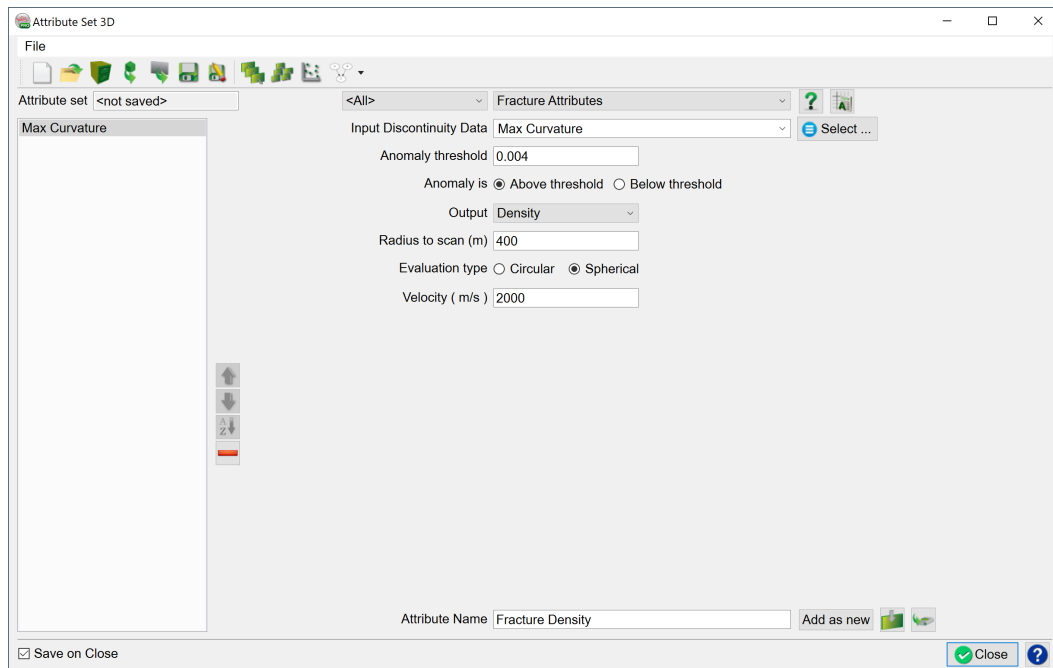
- Improves visualization of potential fracture anomalies.
- It computes the ratio of “number of traces classified as being fractures” to the “total number of traces present”, in a circle of given radius along Z-slices.
- Whether a particular trace can be defined as being part of a fracture, is determined by a user-specified threshold, on various curvature/coherence related attributes such as Max Curvature.

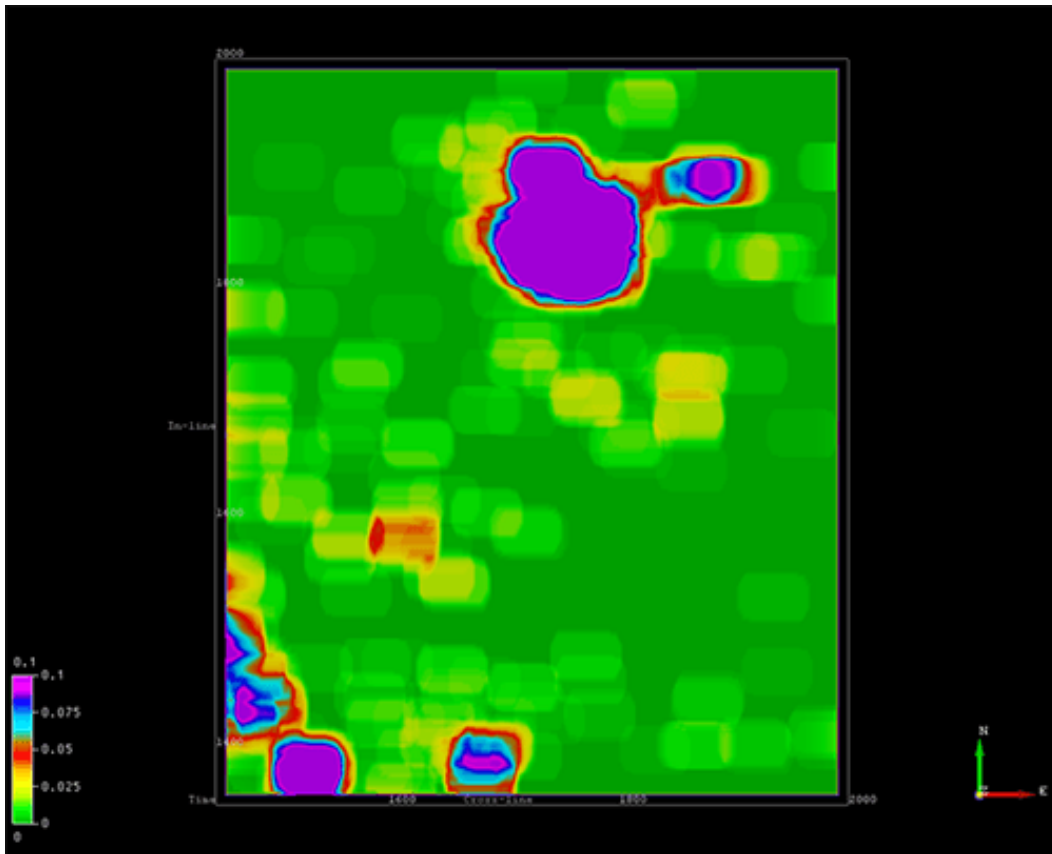
Input Data: Max Curvature

Determine anomaly threshold: 0.004

Output: Density

Radius to scan: A radius for scanning and computing the density of fracture anomalies is also required. Use 400m in this example.





Fracture Proximity

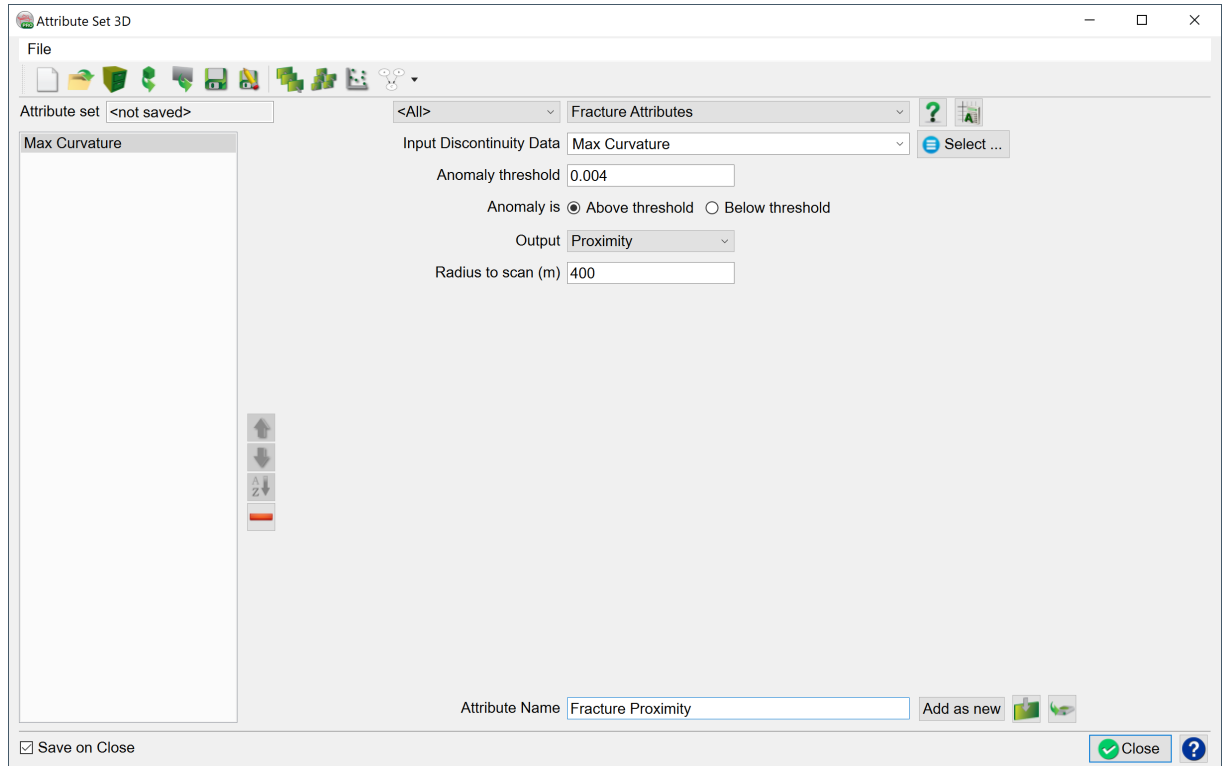
It is an attribute is useful in pin-pointing locations with maximum fracture activity, within a user-defined radius.

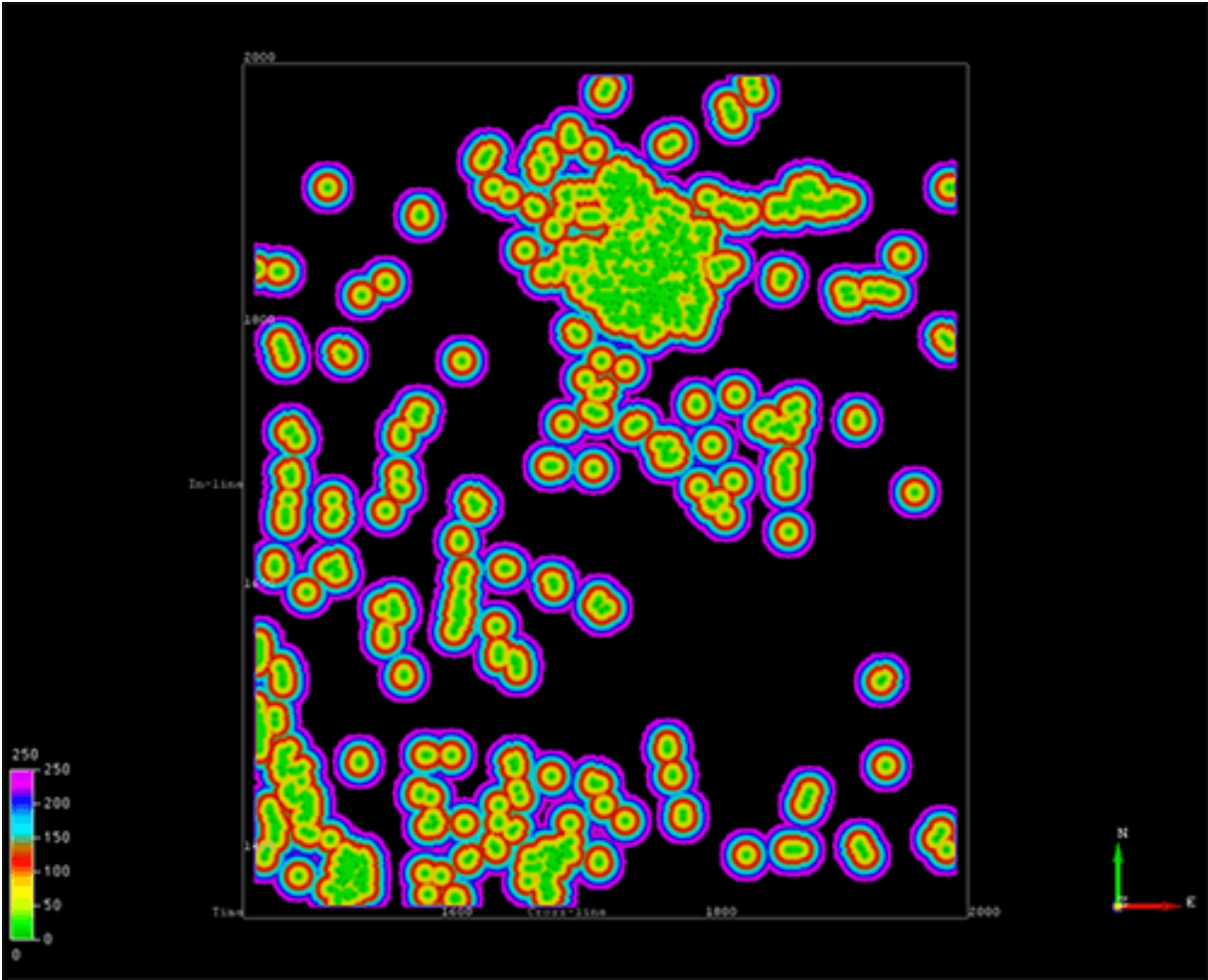
- This “radius” can for example be linked to tracking radius for drilling.
- It computes the lateral distance (i.e. along Z-slice) from a trace location classified as a fracture.

Input Data: Max Curvature (Input data should be a discontinuity attribute, e.g. Similarity and Curvature)

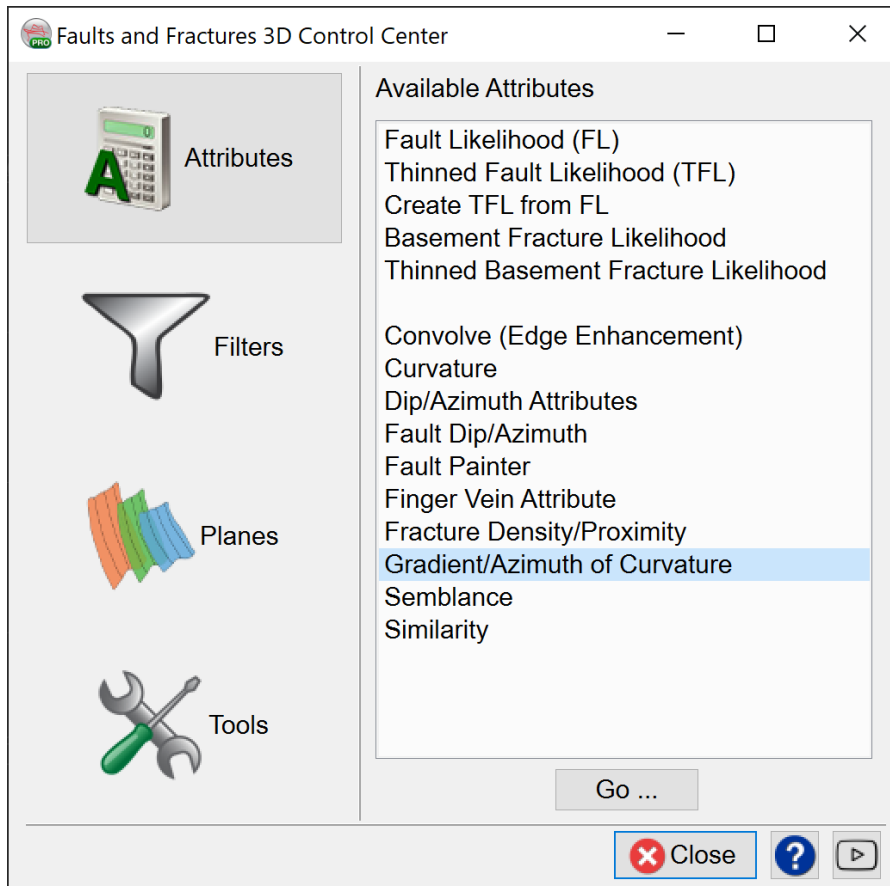
Determine anomaly threshold: Threshold value of input discontinuity attribute above which a fracture anomaly is expected. In this example use 0.004

Output: Proximity





2.1.13 Gradient - Azimuth Of Curvature



Using maximum curvature attribute as an input, the gradient curvature attribute will enhance the curvature following the center of faults and fractures. You can also output the azimuth curvature which is supplementary products that can be used to know the faults orientation, thus it can help assessing the stress regime of the area of study.

For the gradient curvature use the following parameters:

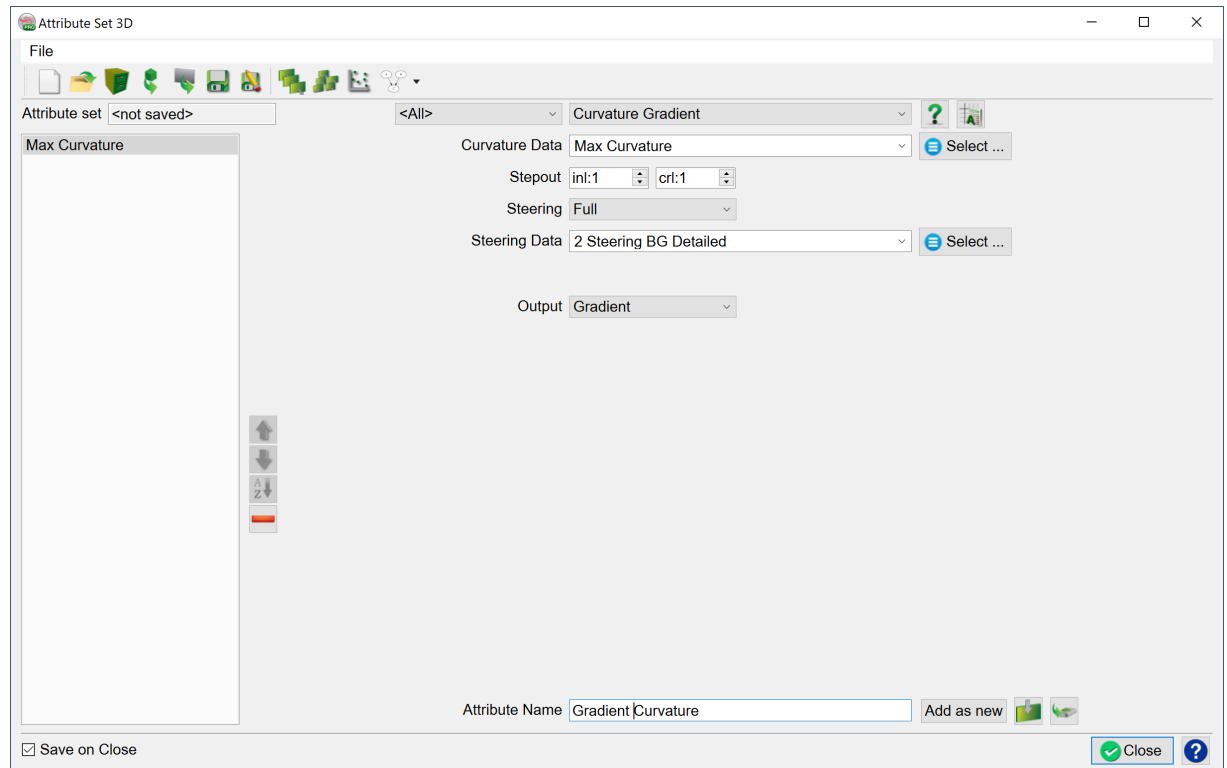
Curvature Data: Max.Curvature

Stepout: 1, 1

Steering: Use Full

Steering Data: Use detailed steering cube

Output: Gradient



For the azimuth curvature use the following parameters:

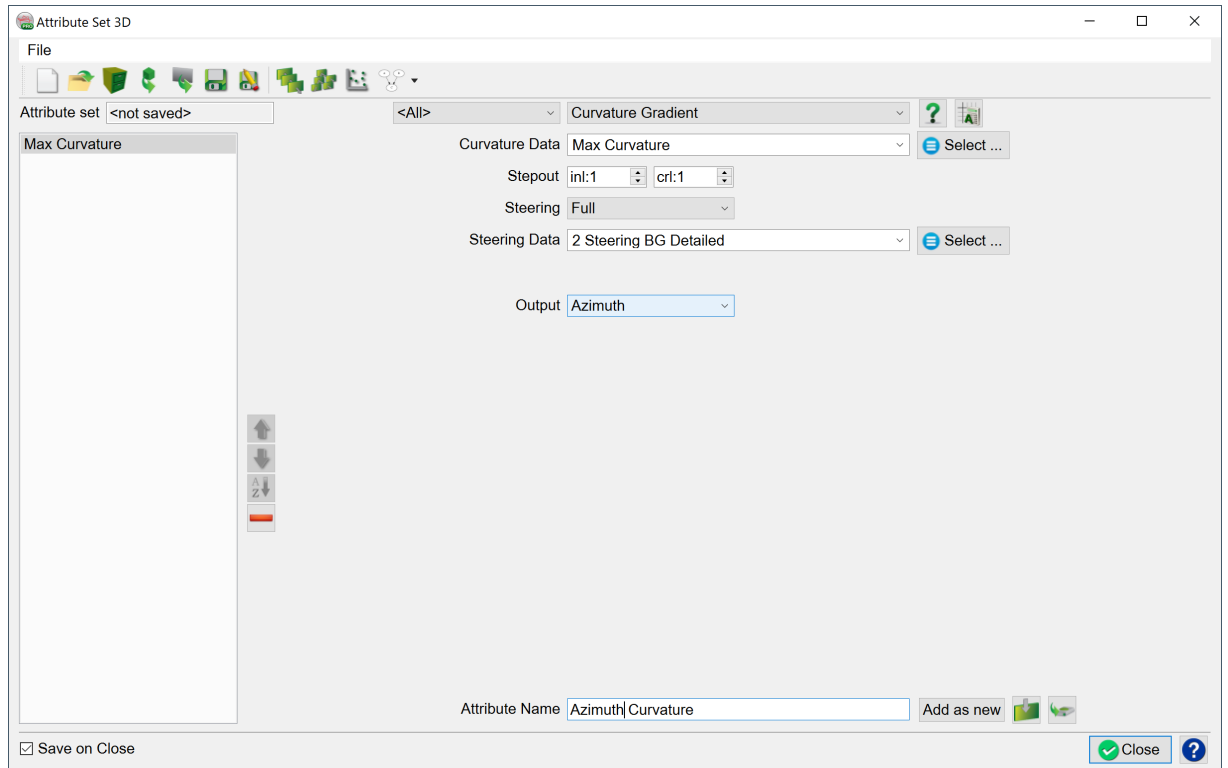
Curvature Data: Maximum Curvature

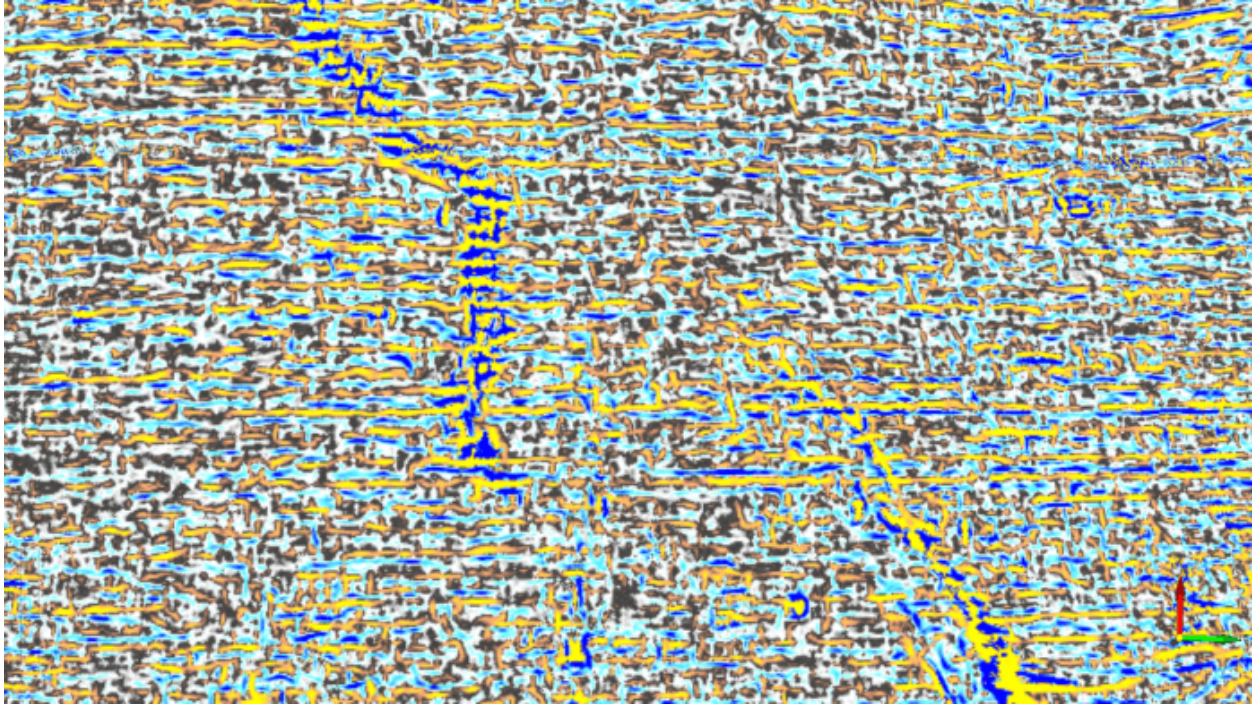
Stepout: 1,1

Steering: Full

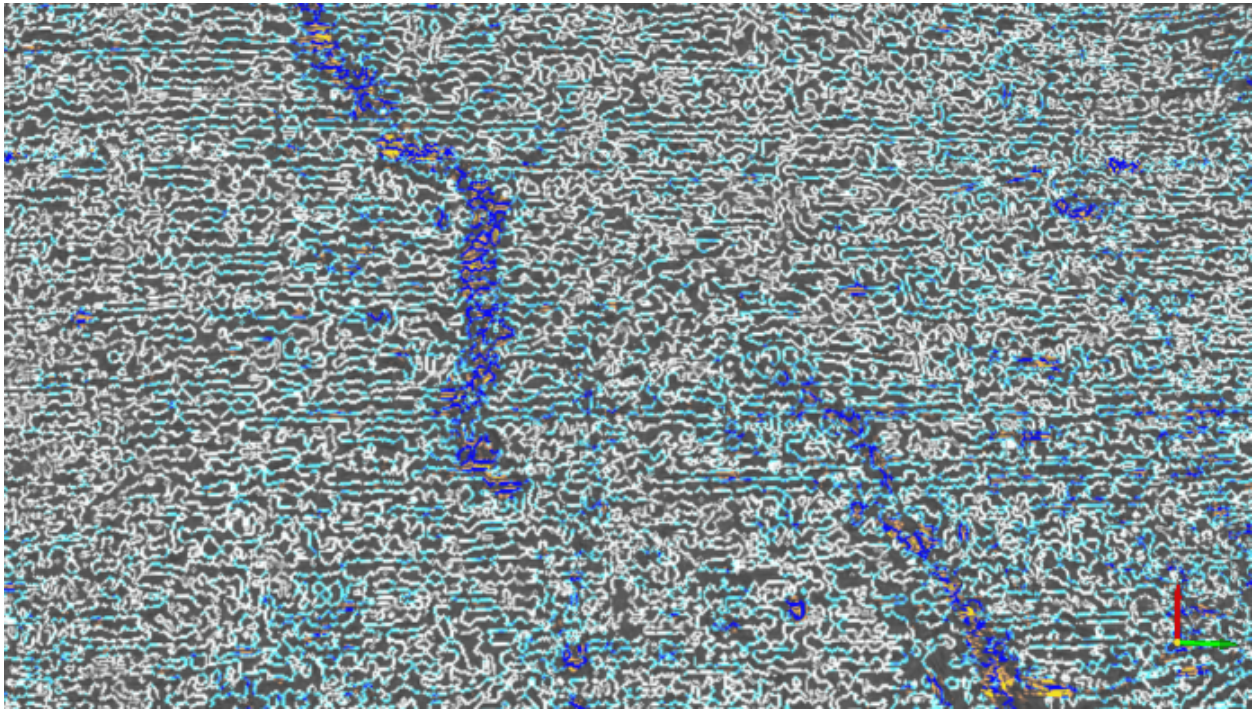
Steering Data: Detailed Steering Cube

Output: Azimuth





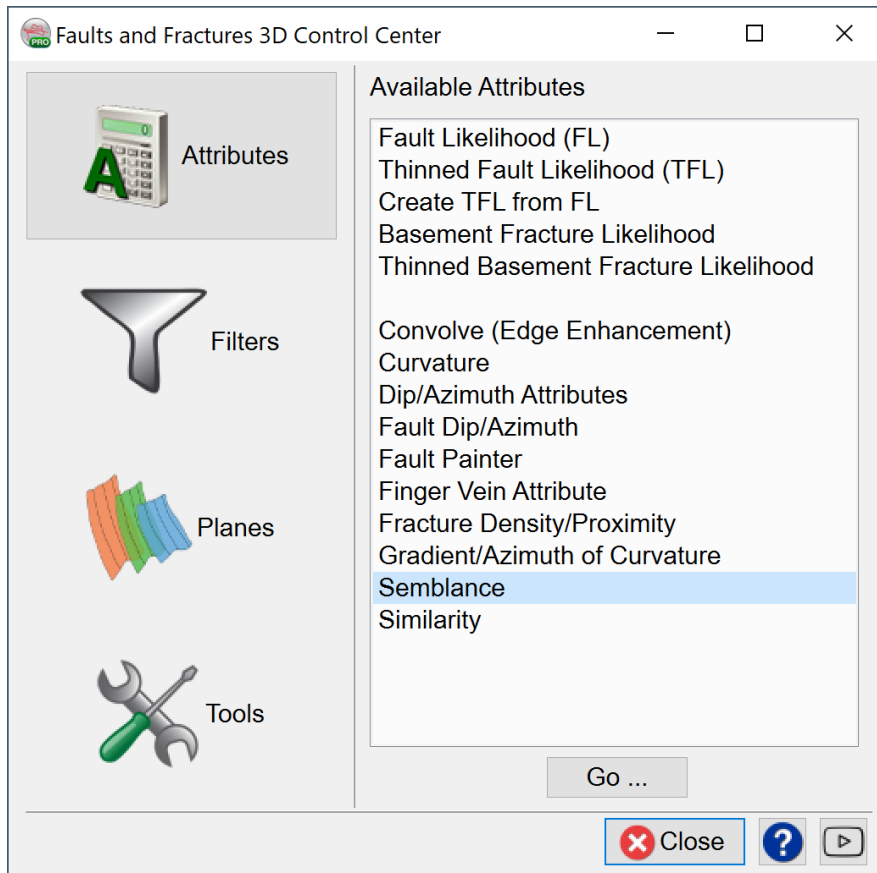
Curvature A



Curvature B

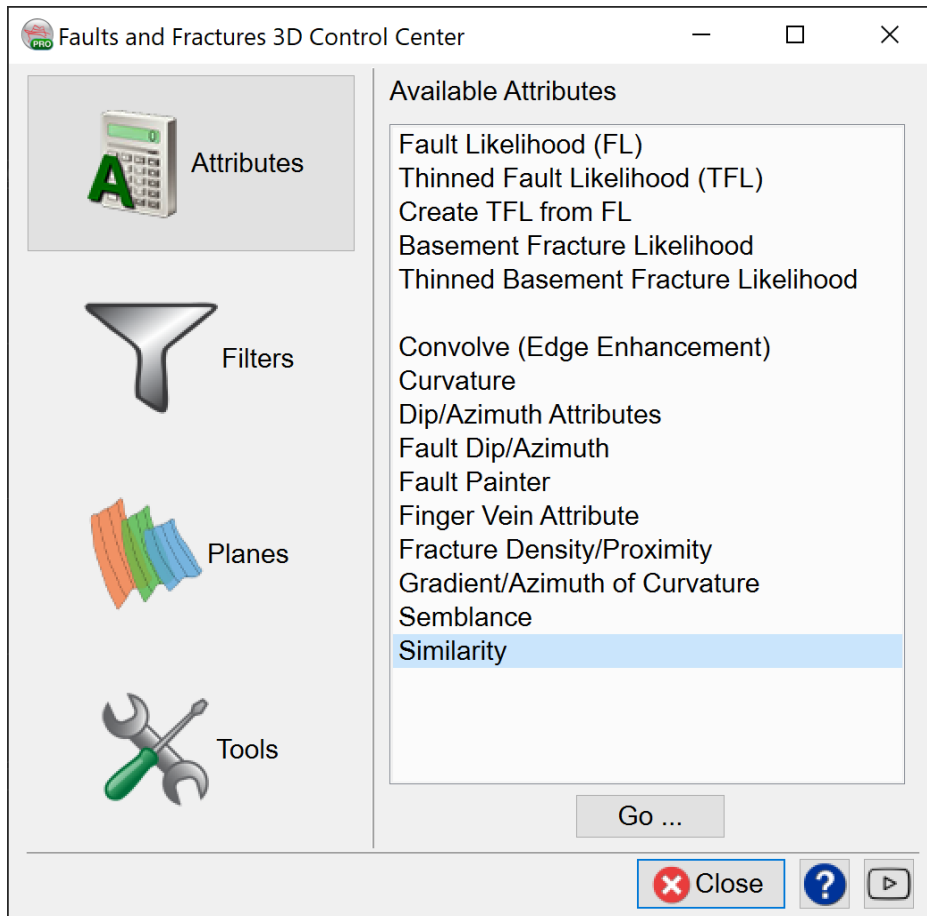
Results of Gradient curvature. A: Maximum Curvature. B: Gradient Curvature.

2.1.14 Semblance



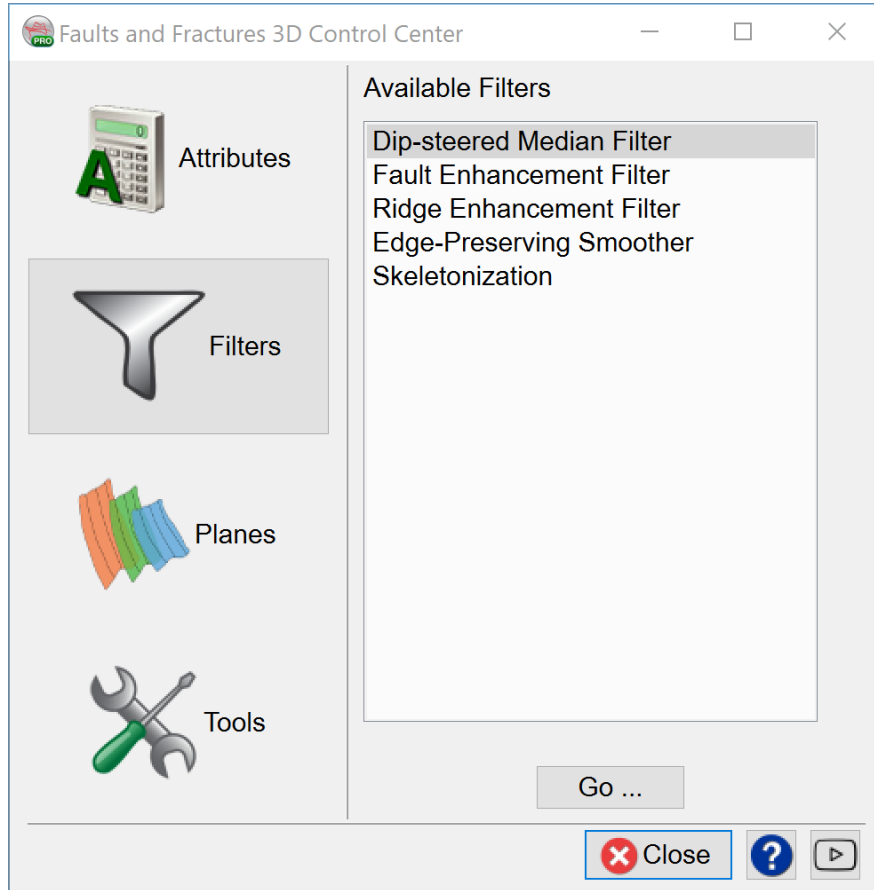
The full article can be found in the [user documentation](#) under *11 Appendix A - Attributes and Filters > 11.24 Semblance*.

2.1.15 Similarity

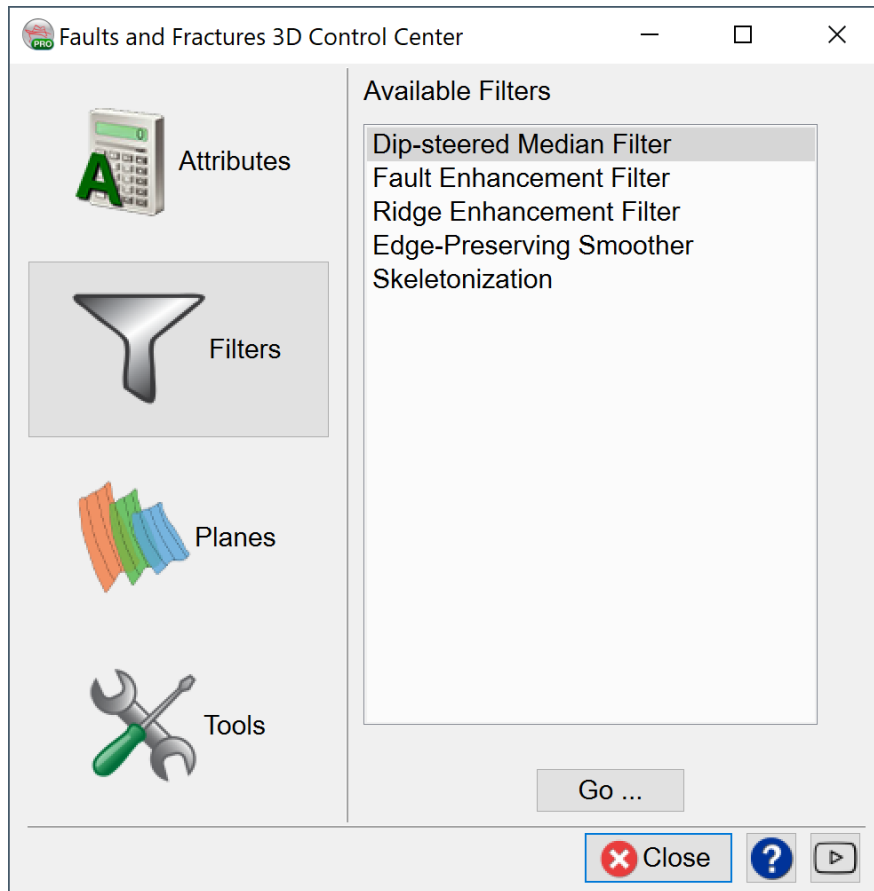


The full article can be found in the [user documentation](#) under *11 Appendix A - Attributes and Filters > 11.25 Similarity*.

2.2 Filters

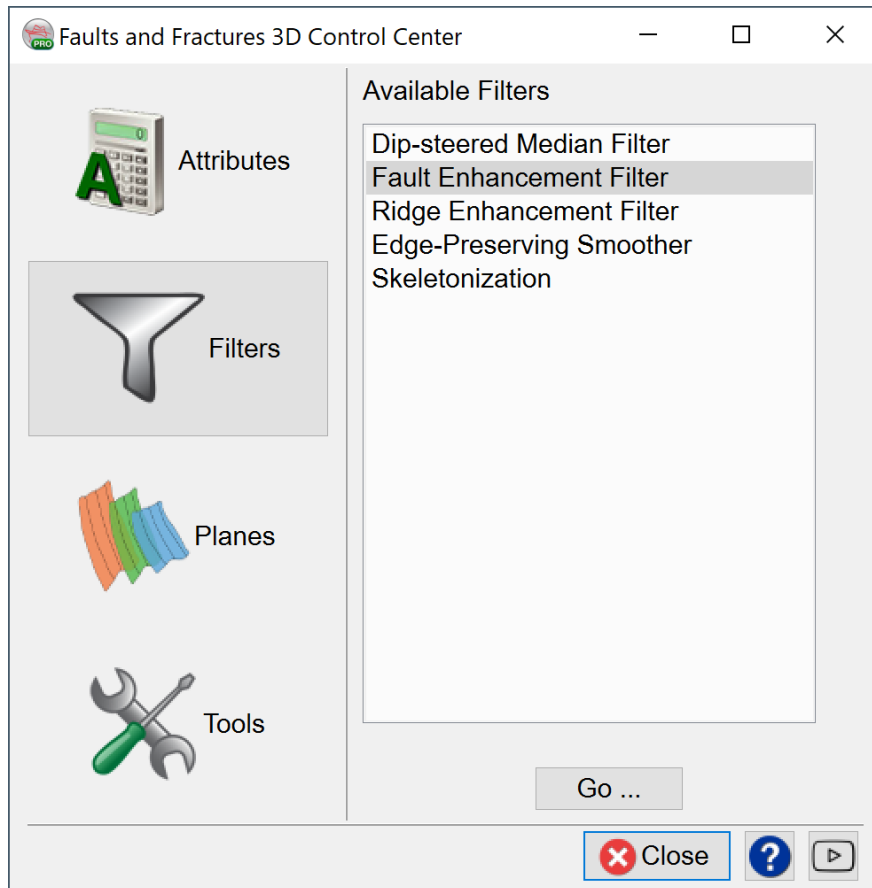


2.2.1 Dip-Steered Median Filter



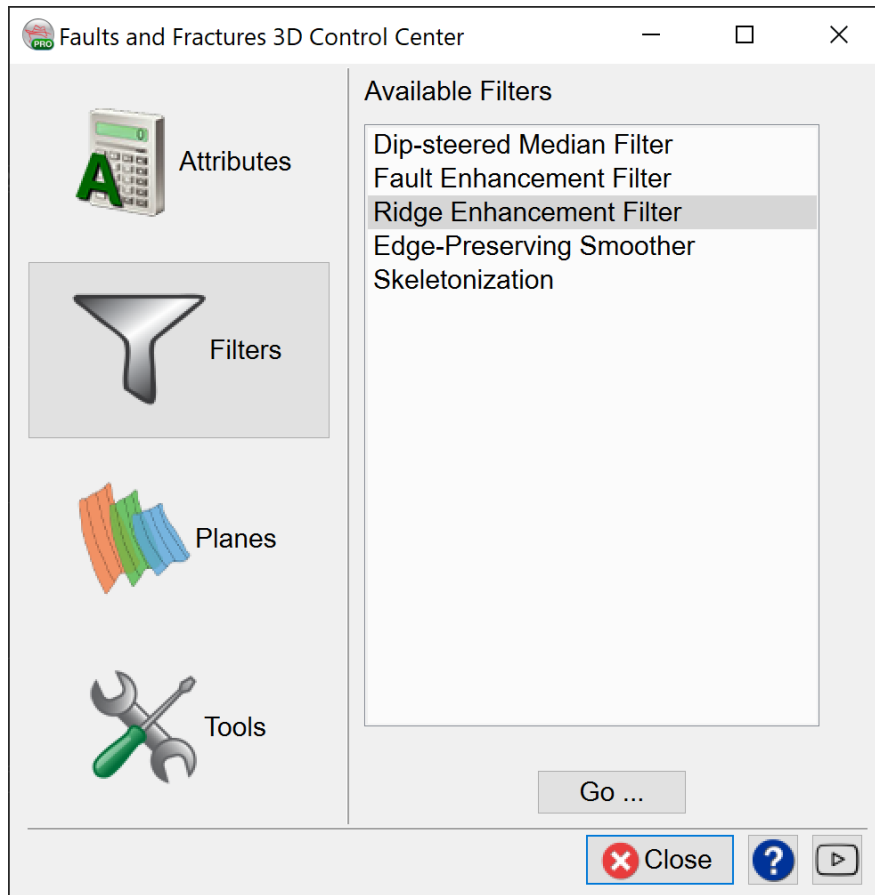
The full article can be found in the [user documentation](#) under *12 Default Attribute Sets > 12.10 Dip-Steered Median Filter*.

2.2.2 Fault Enhancement Filter



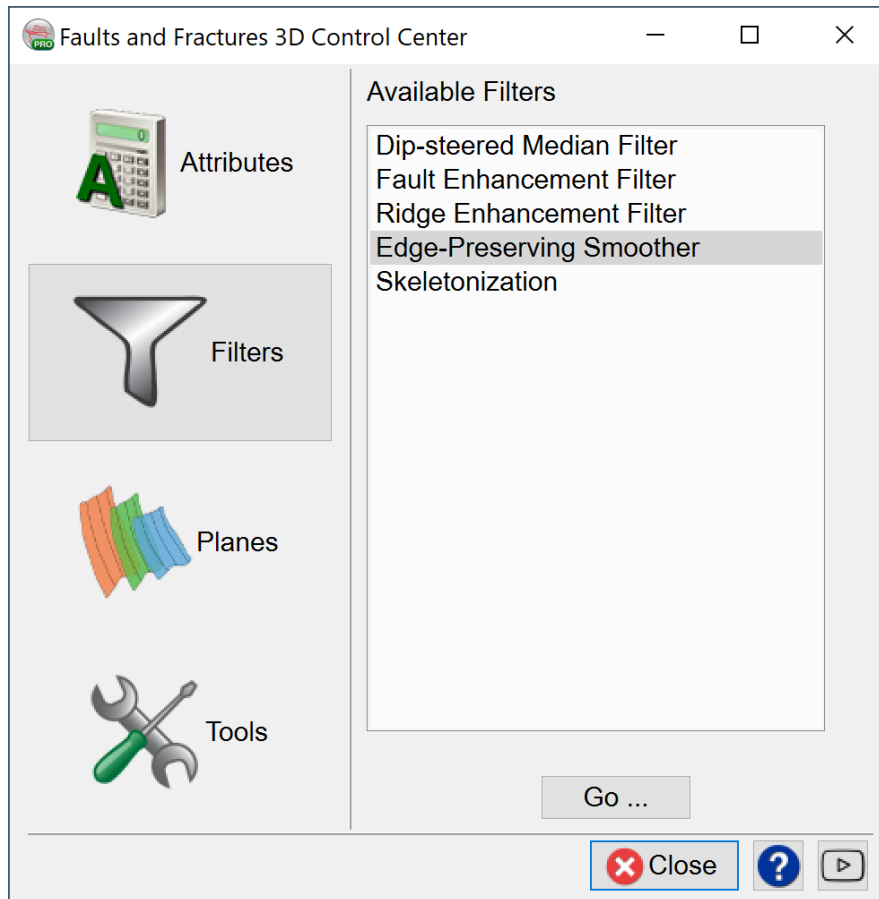
The full article can be found in the [user documentation](#) under *12 Default Attribute Sets* > *12.12 Fault Enhancement Filter*.

2.2.3 Ridge Enhancement Filter

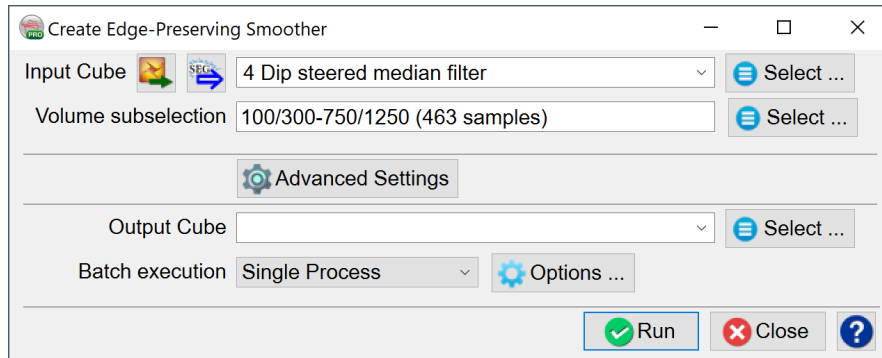


The full article can be found in the [user documentation](#) under *12 Default Attribute Sets* > *12.9 Ridge-Enhancement Filter*.

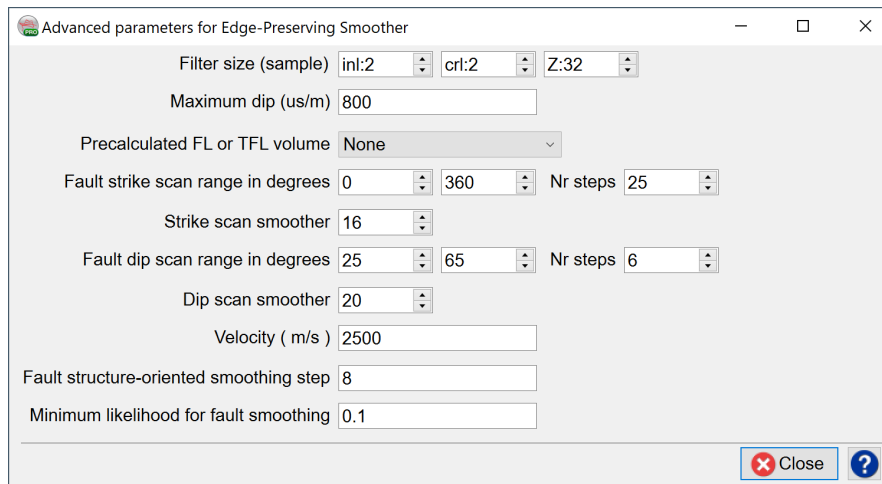
2.2.4 Edge-Preserving Smoother



Fault interpretation is a crucial step in seismic structural interpretation. However, interpreting faults is not always an easy task especially when you are dealing with inconspicuous data. A supplementary application of the TFL attribute, the Edge-Preserving Smoother filter, is designed to make it easy for interpreters to visualize, pick and map the faults. This filter can enhance and improve seismic data visualization for structural interpretation purposes. It generates seismic volume with a razor-sharp edges that is suitable for quickly mapping fault planes.



Advanced settings

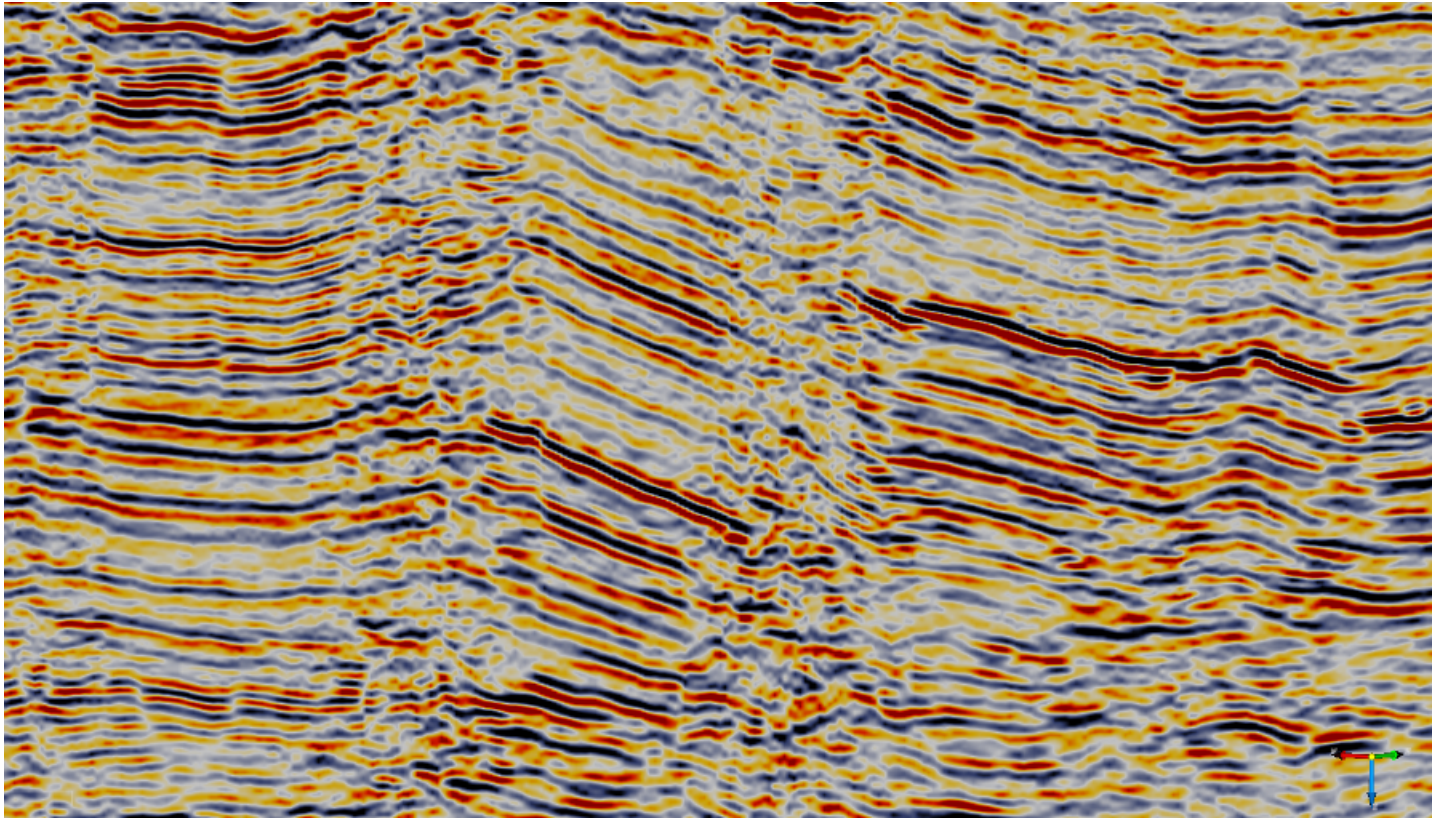


Filter Size (inl/crl/sample): Smaller step-outs, e.g. 1-1-16 or 2-2-32, capture smaller scale features. Setting bigger TFL dip computation step-outs, e.g. 4-4-100, allows to better delineate the bigger fault lineaments.

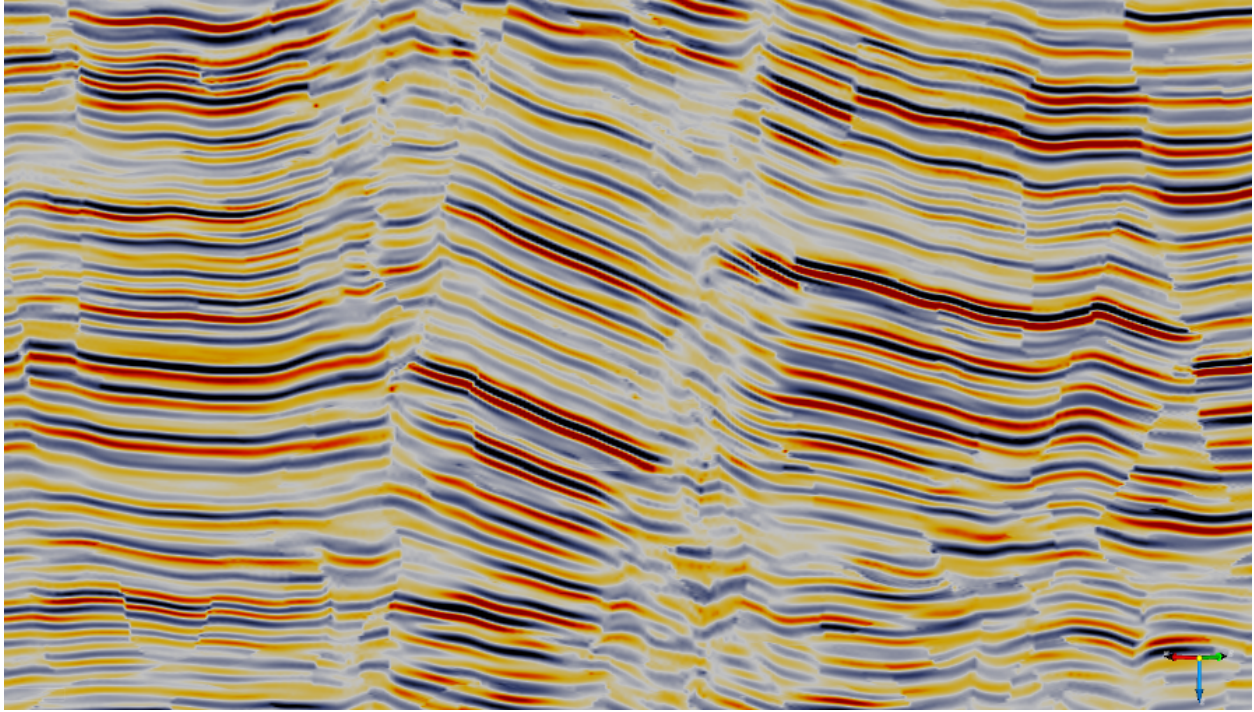
Use thinned fault likelihood volume: A user could either provide the processing parameters for a TFL volume, or to use an existing one.

Fault structure-oriented smoothing step: Radius of the dip-steered median filter.

Minimum likelihood for fault smoothing: The process uses this threshold and the thinned fault likelihood attribute to generate a binary fault location mask array. This array will provide the 3D fault distribution for the dip-steered filter.

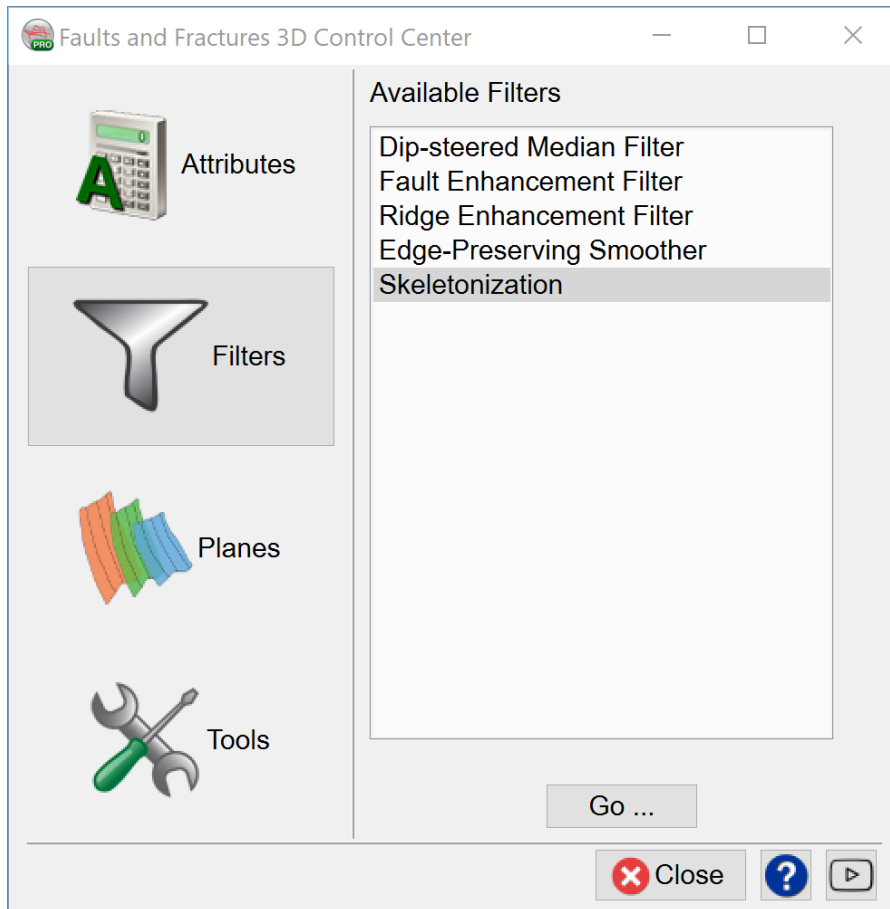


Seismic data before filtering



Seismic data after applying Edge-Preserving Smoother.

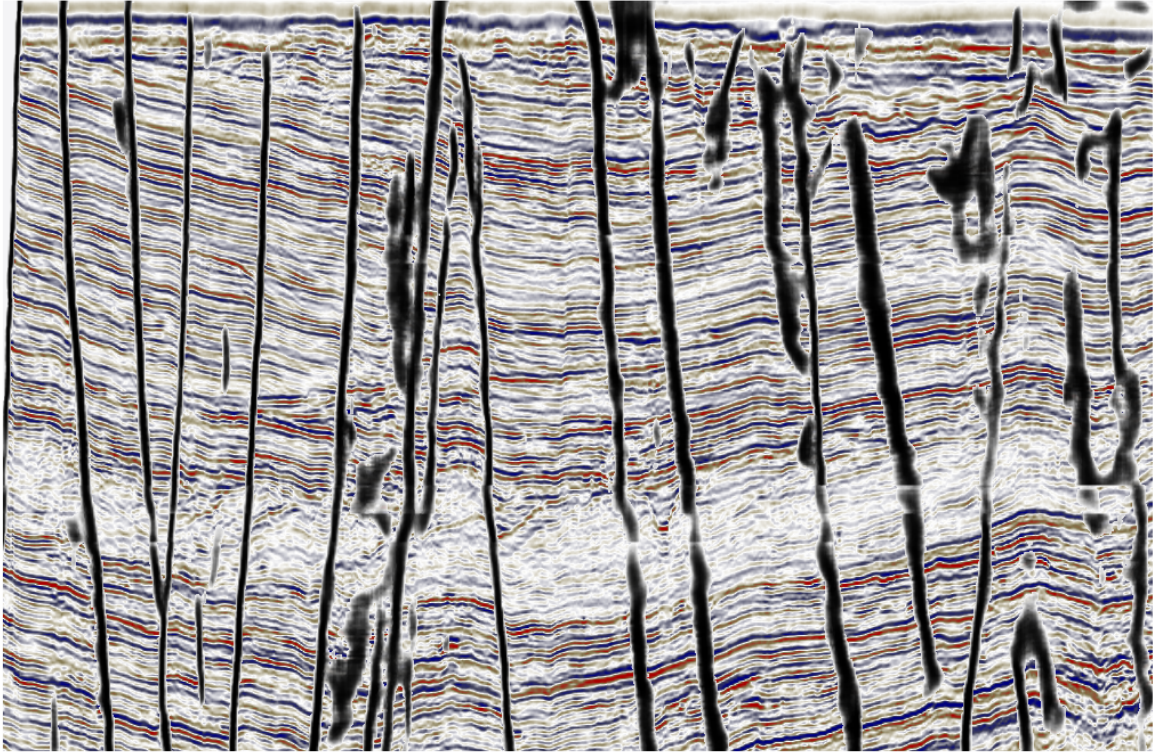
2.2.5 Skeletonization



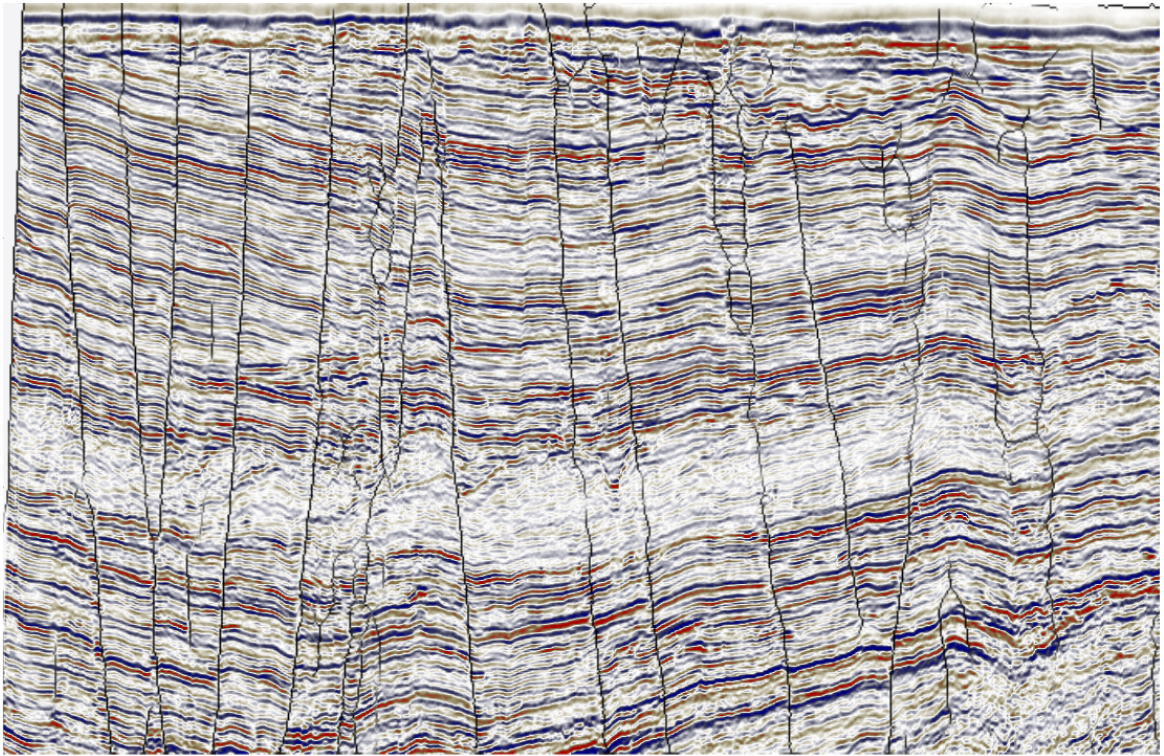
The aim of this filter is to fine tune the Thinned Fault Likelihood volume or the Unet Fault Probability volume.

In the Volume Skeletonization window, the user can set a certain threshold, and choose to keep the value above or below the threshold and run the output.

Results



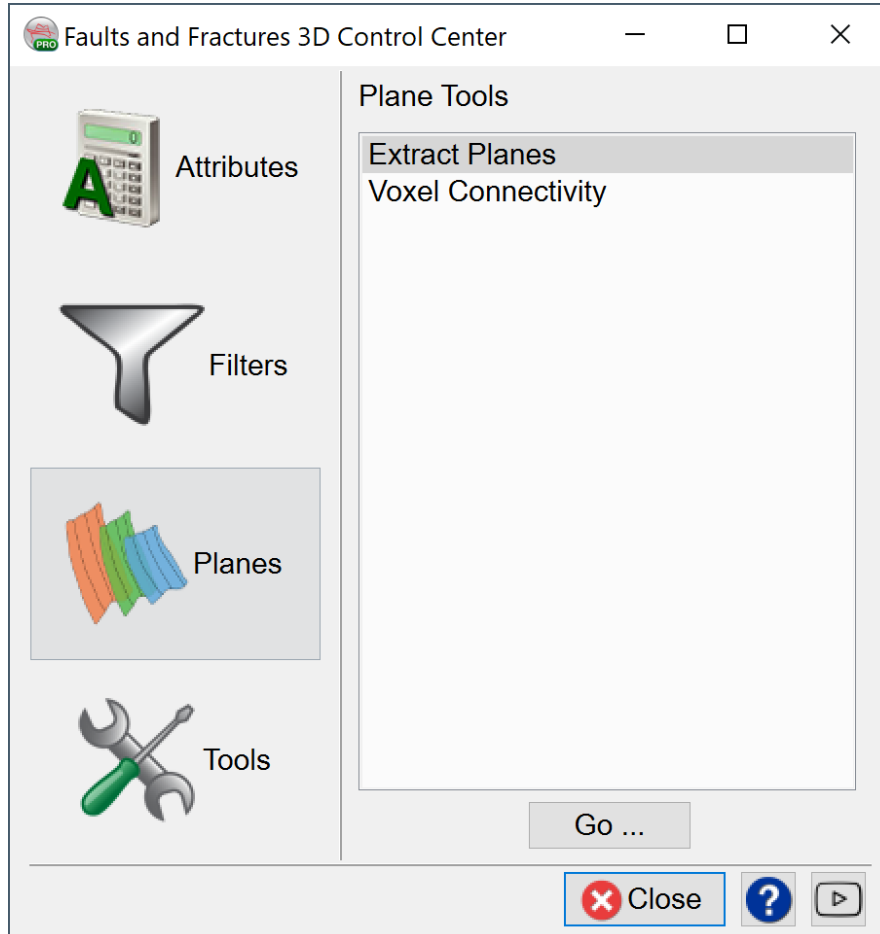
Unet Fault Predictor before thinning



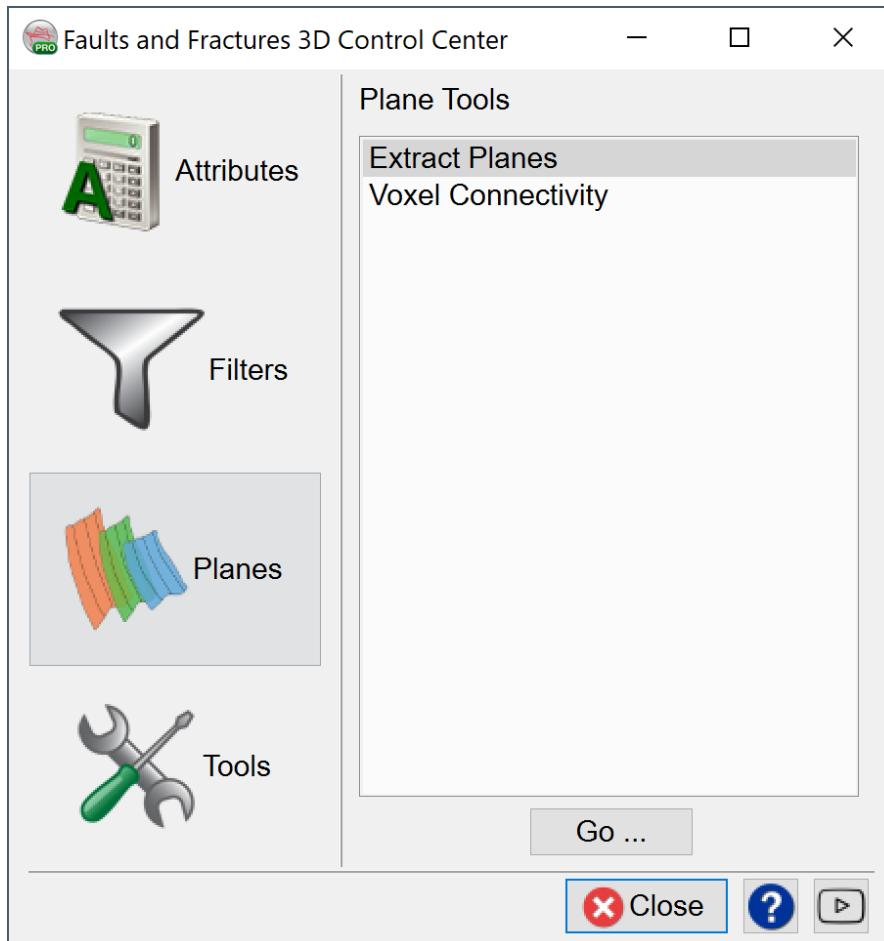
Unet Fault Predictor after thinning

* Images: Kerry 3D survey, Taranaki Basin. Data source: New Zealand government.

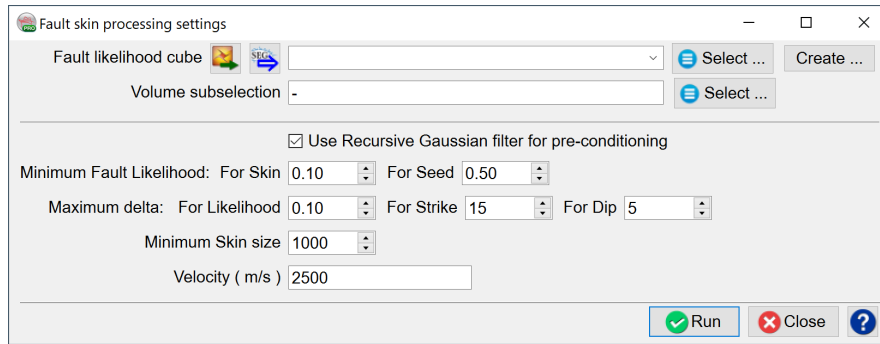
2.3 Planes



2.3.1 Extract Planes



The tool extracts fault planes from Fault Likelihood (NOT Thinned Fault Likelihood). It first produces and displays the best possible fault skins, which can later be converted to actual fault planes.



Fault Likelihood Cube: select a pre-processed Fault Likelihood volume. To process it, go to Attribute > Thinned Fault Likelihood and in the Advanced Settings set Apply Thinning parameter to No.

Volume subselection: can be restricted to a faulted interval of interest.

Use Recursive Gaussian filter for pre-conditioning: switched on by default.

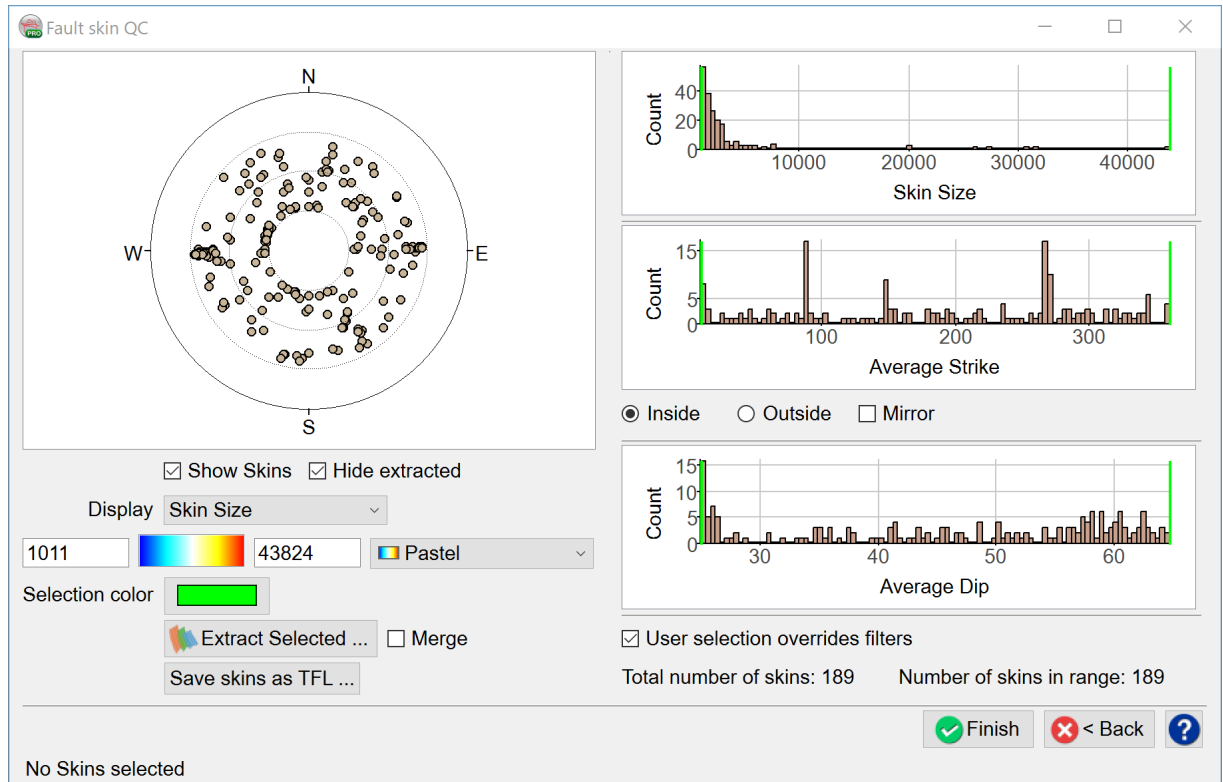
Minimum Fault Likelihood: choose minimal value of fault likelihood for a skin and a seed, where the skin is a set of connected XYZ points which belong to one fault plane, and the seed is the point with the largest fault likelihood value.

Maximum delta: allowed difference between the neighboring points of the fault skin. Choose the maximum value for the likelihood, strike and dip.

Minimum skin size: increase the value to dismiss smaller fault skins.

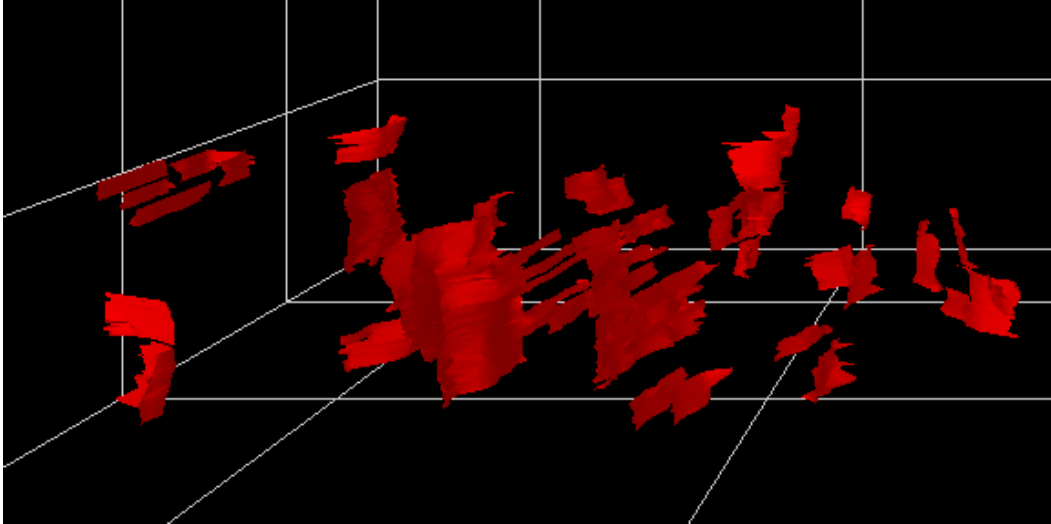
Velocity (m/s) Velocity needs to be used in time surveys to convert angles. Use an average value for the data set.

On pressing Run, fault skins will be displayed in the scene along with the Rose diagram Fault skin QC window.

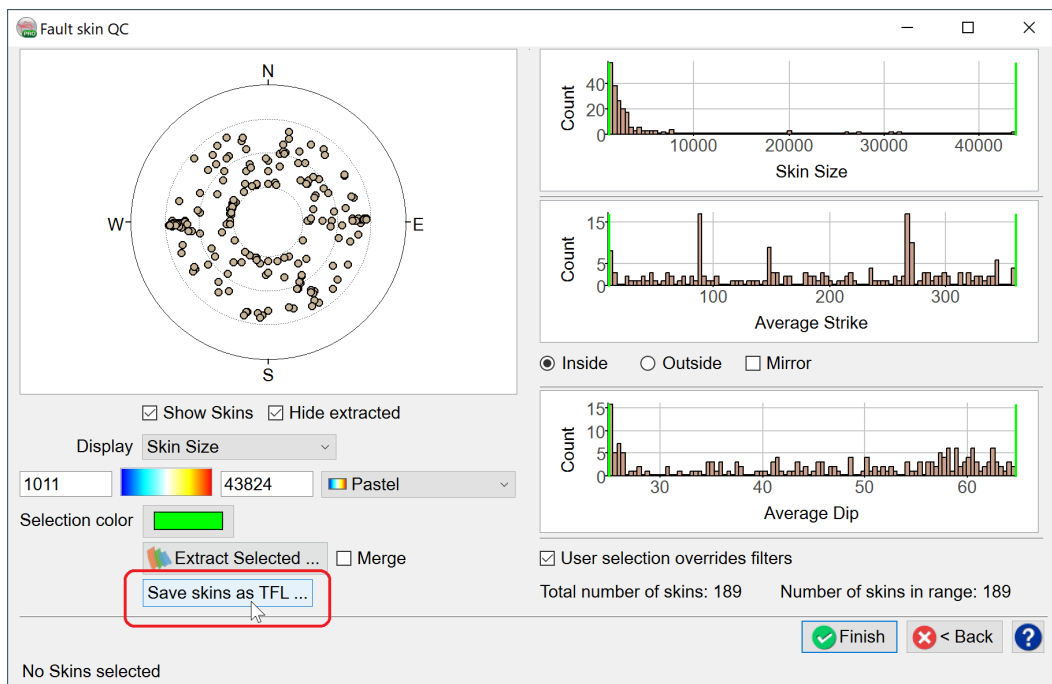


The Fault skin QC window contains the Rose Diagram, the information box and the distribution histogram. The green vertical sliders in the distribution histogram can be dragged to select, for example, only the largest faults.

Each skin can be selected with a left-mouse click, Ctrl+A selects all the skins in the scene. Click the *Extract Selected* button to extract the faults. Skins may also be merged and saved as a single fault plane. To select multiple skins press Left-Click to select the first skin and use press SHIFT-Left-Click to select additional skins.

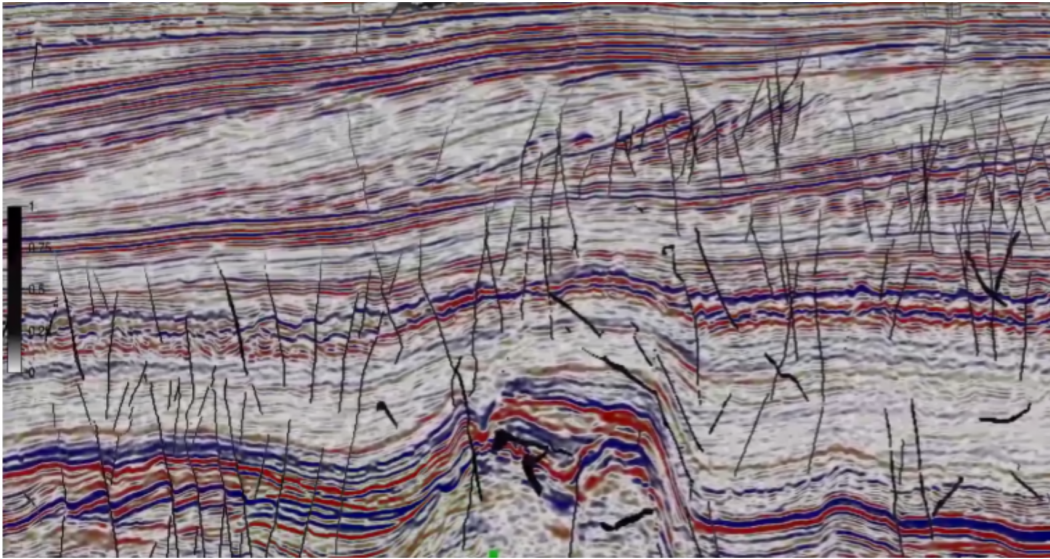


Users can compute clean TFL from skins using the *Save skins as TFL* option:



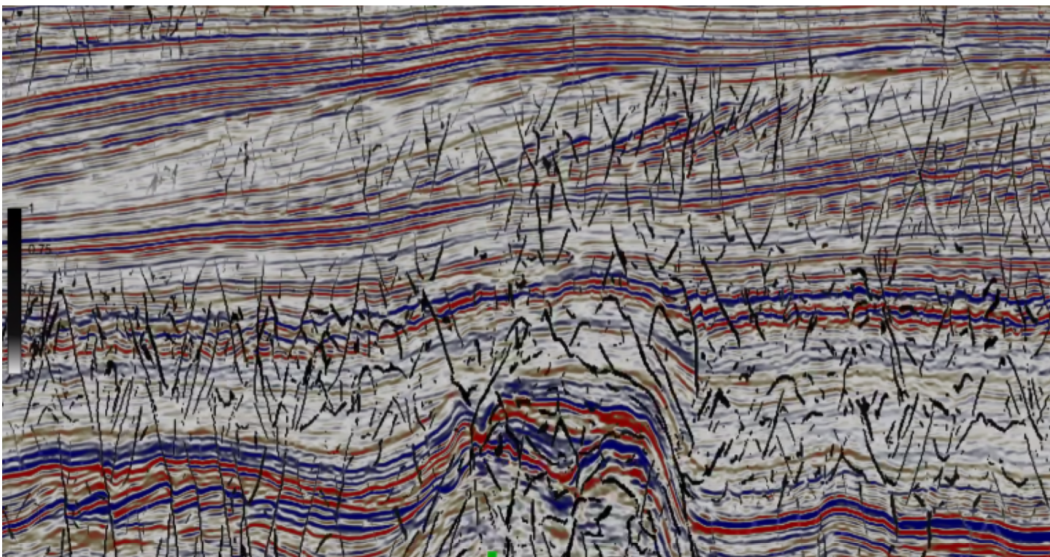
Mind that this process completes very quickly.

TFL from skins:

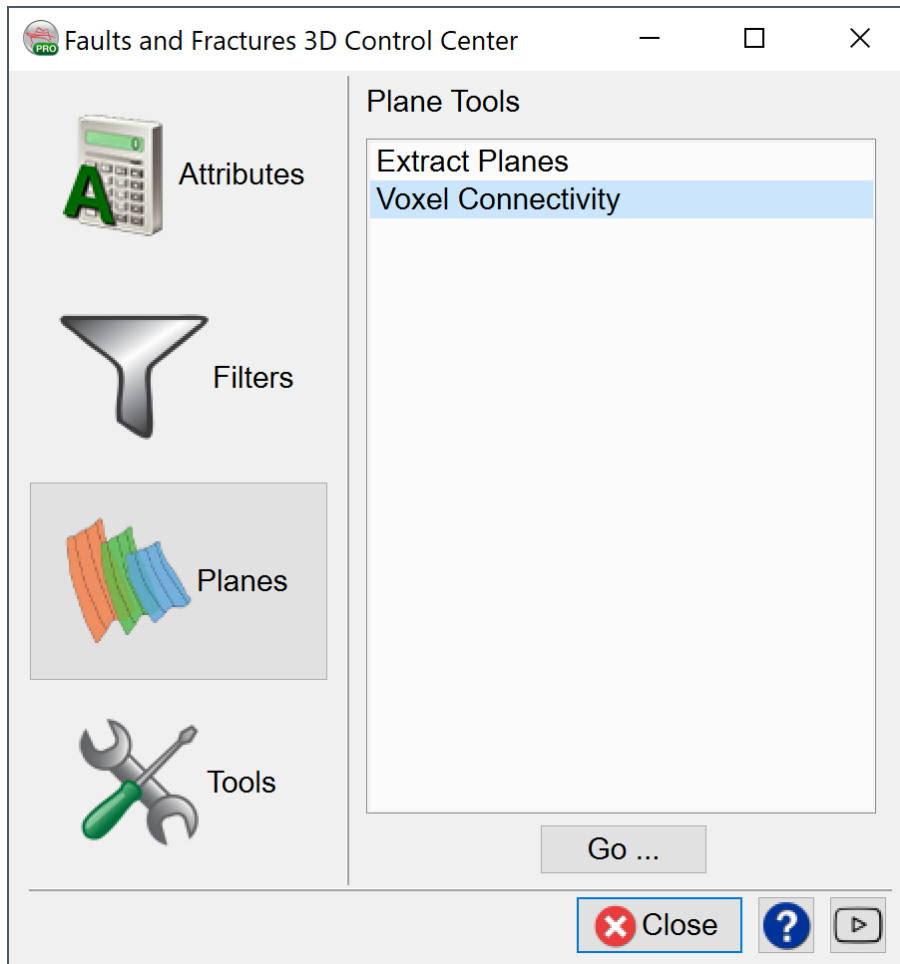


Note the clean, continuous faults compared to TFL in next slide

TFL:

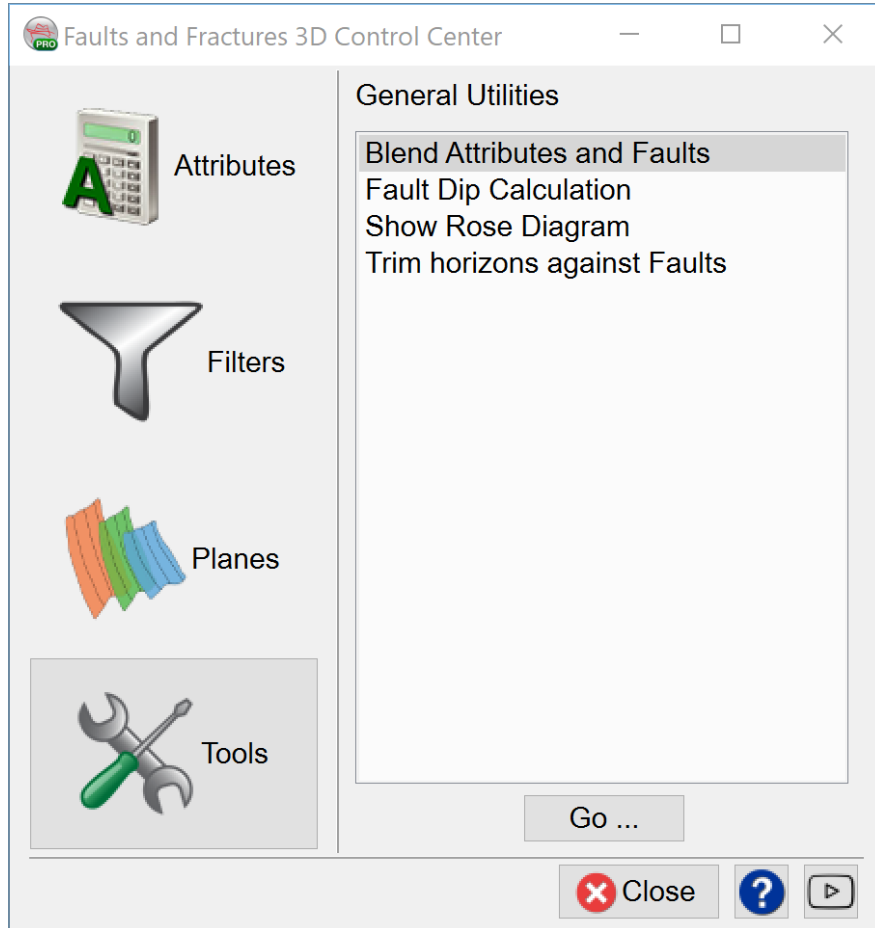


2.3.2 Voxel Connectivity

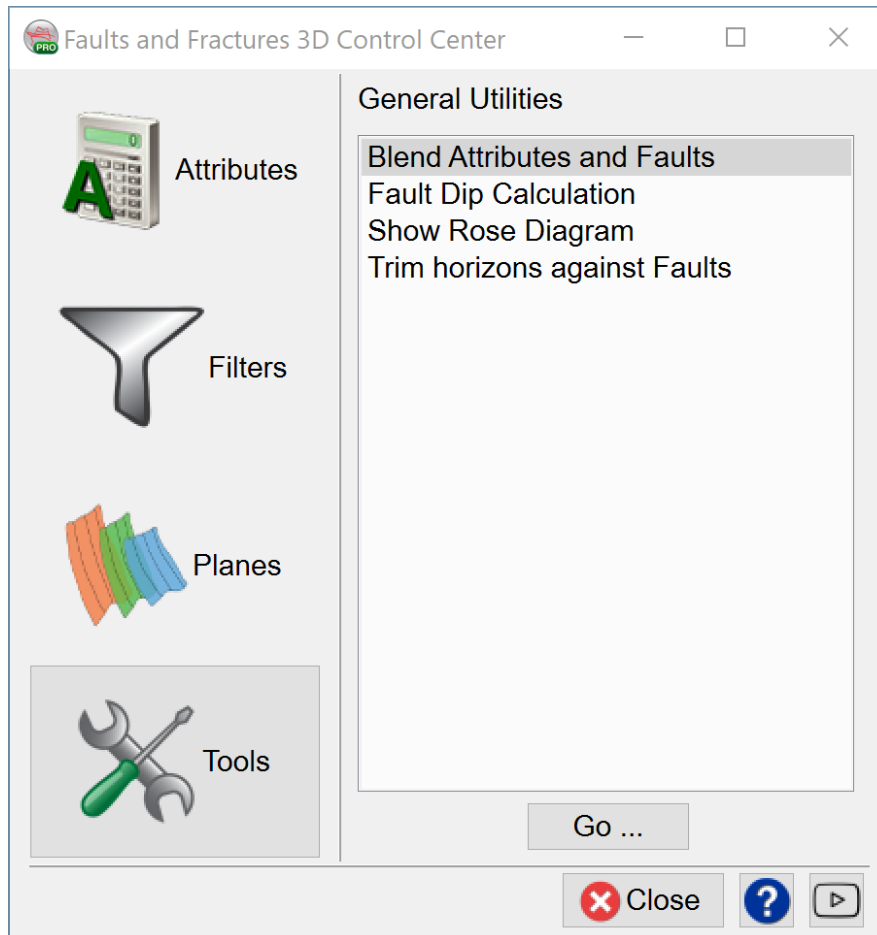


The full article can be found in the [user documentation](#) under *5 Analysis > 5.2 Volume Builder Setup > 5.2.7 Voxel Connectivity Filter*.

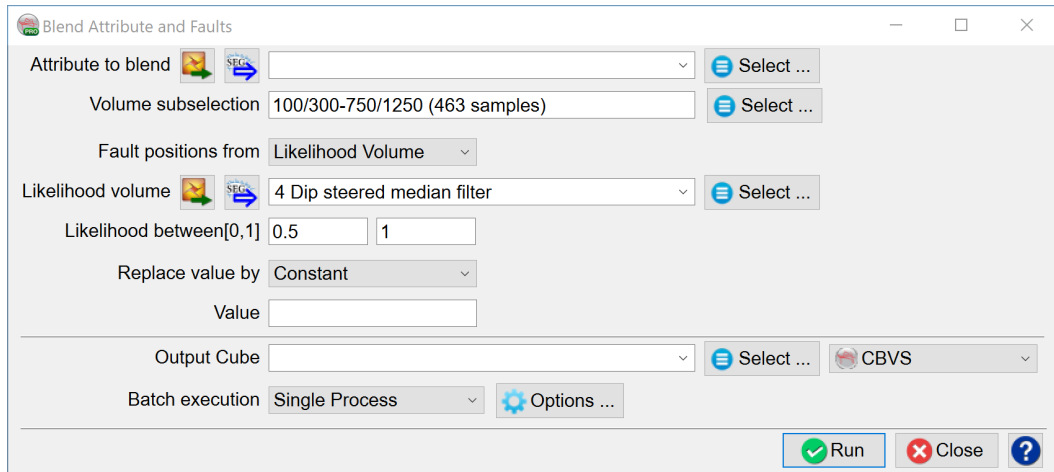
2.4 Tools



2.4.1 Blend Attributes & Faults



Blend Attributes and Faults is a powerful technique to visualize faults and attributes as one volume. The tool can be started from the Tools menu of the Faults and Fractures Control Center.



Attribute to blend: Select an attribute to be blended.

Volume sub-selection: To control what area to use, also useful for speeding up the process.

Fault positions from: Two options are available - choose between a Fault Likelihood volume or manually interpreted faults.

Likelihood between [0,1]: If a Fault Likelihood volume is selected, the user needs to define what range of the volume will be blended and replaced by a constant value, from a volume, or by undefined value.



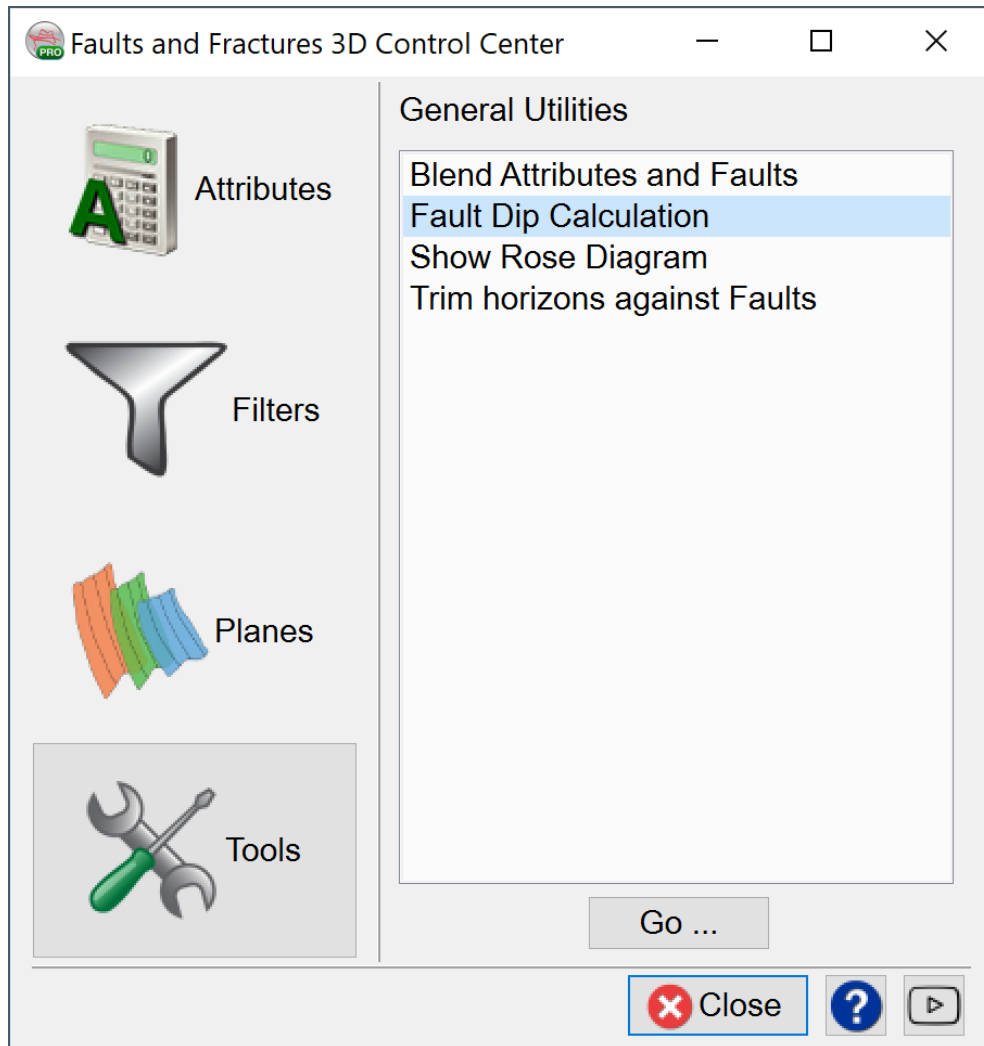
Similarity attribute



Similarity blended with a TFL volume. Note that only TFL values between 0.5-1 are used here and replaced by the value of 1.

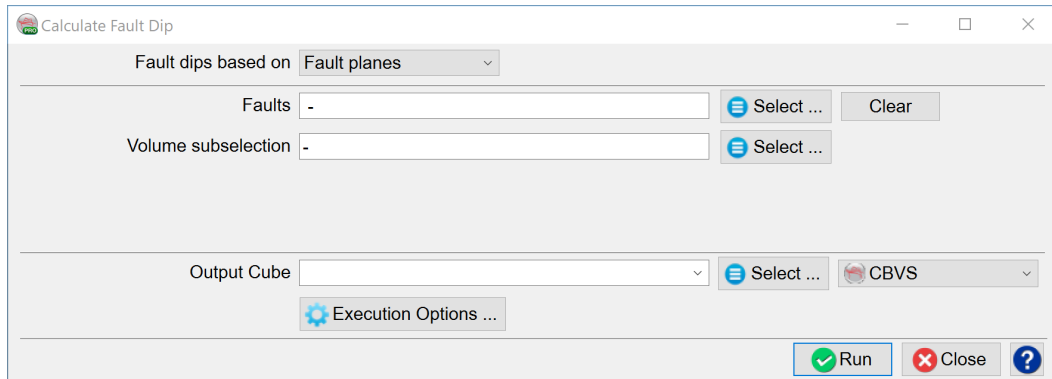
Tip: If you display the original TFL volume over the new blended attribute, and restrict the color bar scale to 0.5-1, the TFL results will be identical to the blended results.

2.4.2 Fault Dip Calculation



The Fault Dip, is calculated in the Faults and Fractures 3D Control Center, under Tools. It generates a ratio cube of the vertical displacement per horizontal distance. The cube has two components - Inline Dip and Crossline Dip. In a time survey, the values are in microseconds/meter and in a depth survey (with meter as unit), the values are in mil-

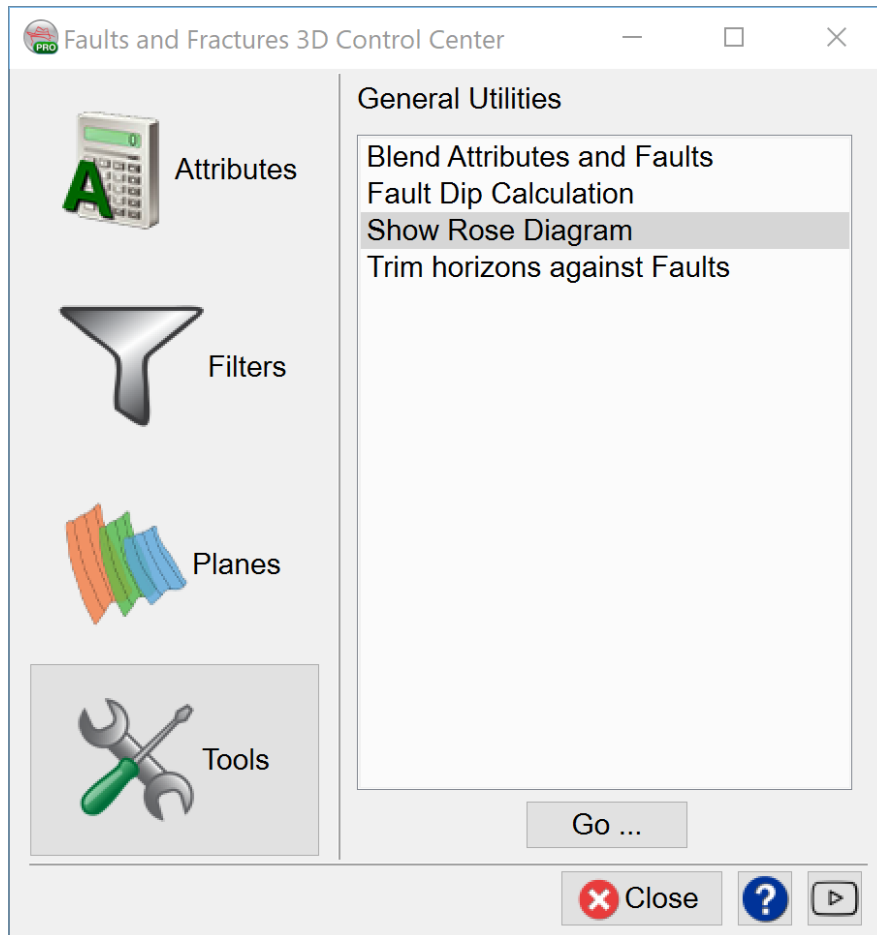
limeter/meter. The Fault Dip unit is the same as the unit used in Steering cubes, created using the Dip Steering plugin.



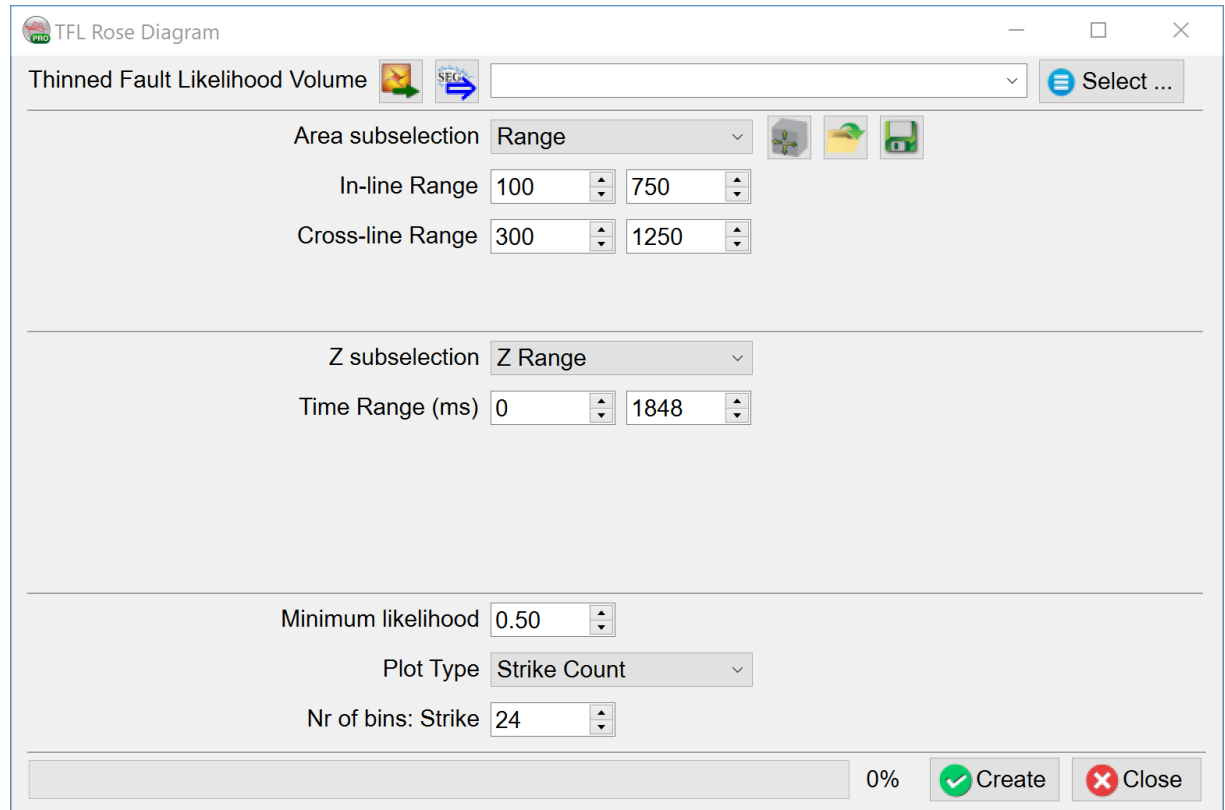
Once the dip cube has been calculated (either using the Fault Dip Calculation or Dip Steering), the user can read the values from the Manage Seismic window using the [Browse](#) tool.

In OpendTect 6.6, the user can manually measure the dip of the line joining any points in either degrees or microseconds/meter, with the Measure tool in the Graphics Toolbar. The user can use this as a QC tool. The dip angle is measured from the horizontal and is usually lower than the angle 'visible' on screen as the Z-axis is usually stretched in comparison to the horizontal. Therefore it may be that what appears to be steeply dipping on screen has only a moderate dip. If you increase the velocity, it will increase the calculated vertical distance and the dip, and vice-versa.

2.4.3 Show Rose Diagram

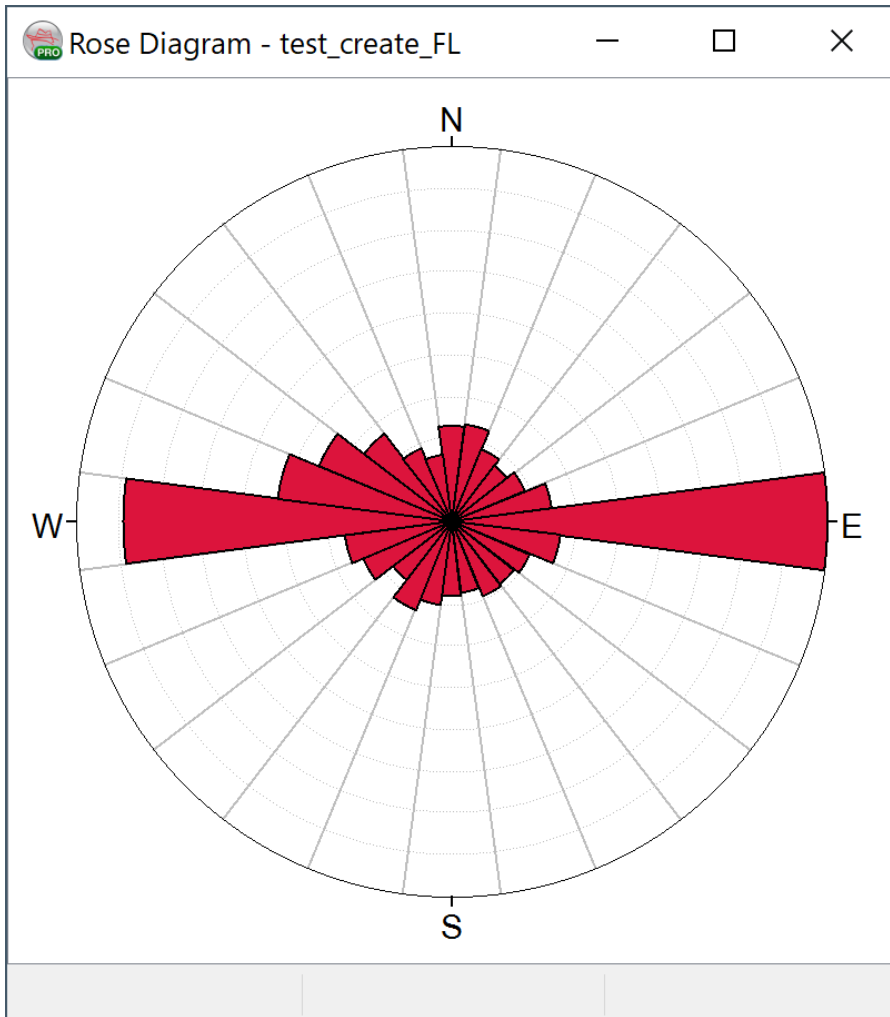


The rose diagram works similar to a histogram. It counts the number of samples which have a TFL value above the given threshold. The diagram is divided into 36 sectors, representing a 10 degree strike angle range.

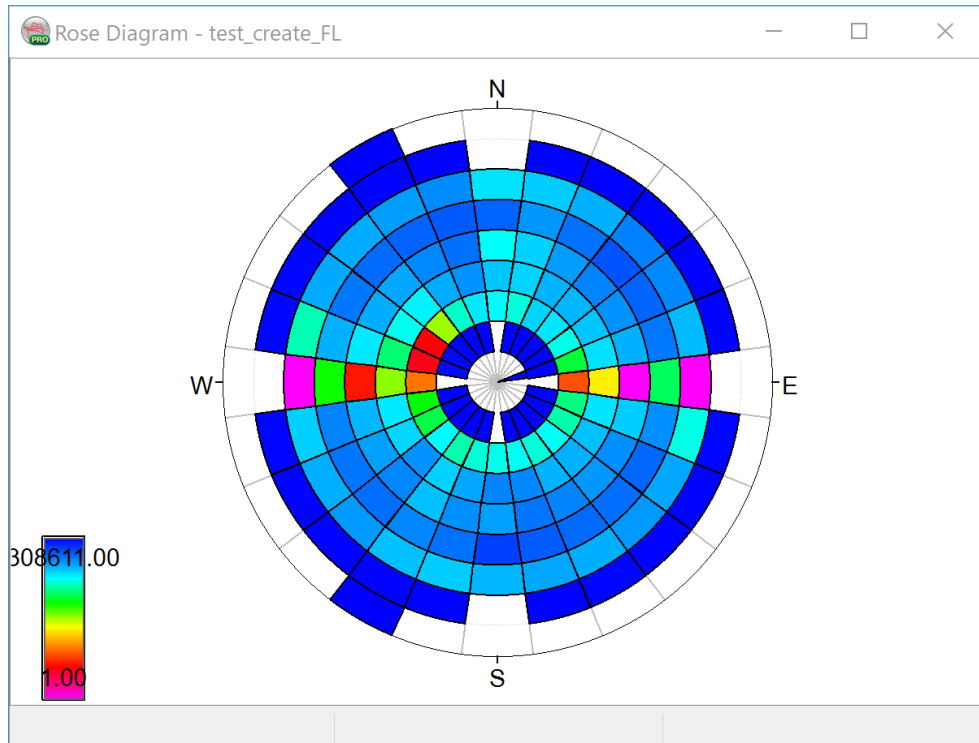


Two types of rose diagram are supported:

1. The Strike Rose Diagram. This diagram shows the count (or number of samples) in each strike sector. The more samples in a certain strike angle range, the longer the bar. Therefore, the radius of the rose diagram represents the count.
2. The Strike/Dip Rose Diagram. This diagram shows the count based on strike and dip direction. Each strike sector is divided into 9 dip angle parts, representing a 10 degree dip range. The radius of this rose diagram therefore represents the dip angle from 0 in the center to 90 on the edge. The color tells how many samples fall into a certain strike/dip sector-part.

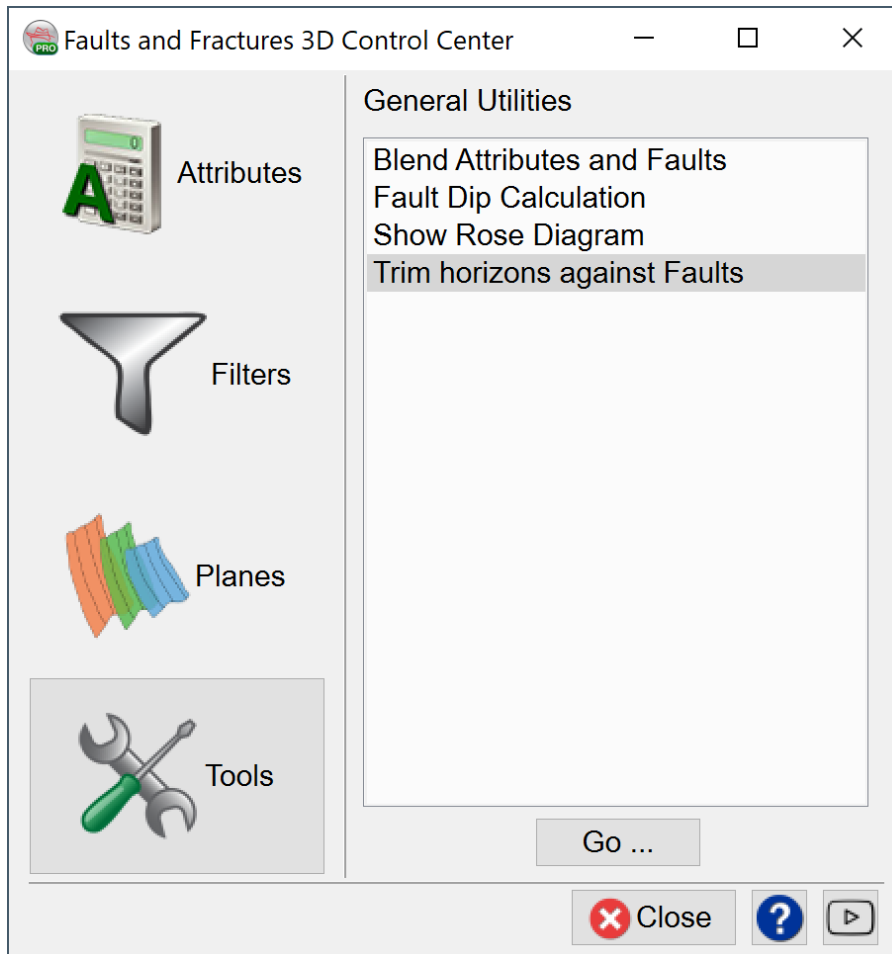


An example of a Strike Count Rose Diagram.



An example of Strike/Dip Rose Diagram. It shows that the most of the faults/fractures have a North-South direction and have a dip angle between 60-80 degrees.

2.4.4 Trim Horizons Against Faults



The full article can be found in the [user documentation](#) under *5 HorizonCube > 5.2 HorizonCube Control Center > 5.2.3 Tools > 5.2.3.3 Trim Against Faults*.

3 Machine Learning



3.1 Introduction and Terminology

Machine Learning (ML) is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed. ML is subfield of **Artificial Intelligence (AI)** which is defined as a branch of computer science dealing with the simulation of intelligent behavior in computers.

The Machine learning plugin links the OpendTect Pro environment to the [Keras \(TensorFlow\)](#), [PyTorch](#) and [Scikit Learn](#) platforms. With the Machine learning plugin, OpendTect users can do Clustering, Classification and Regression of the following objects:

1. **Wells** e.g. for predicting missing logs, markers, stratigraphic trends, litho-facies, clustering logs etc.
2. **Seismic** e.g. for clustering 3D seismic, imaging faults, chimneys, salt bodies, to predict missing traces, picking velocities, picking mutes, increasing spatial and temporal sampling rates, transforming seismic data to broadband etc.

3. **Seismic & wells** e.g. for predicting reservoir properties, explaining patterns, tying wells, etc.
4. **Pre-trained Models:** models trained outside of OpendTect can be imported and stored in the predefined default pre-trained model disk location and applied to unseen data.
5. **Neural Network** e.g. for predicting pattern and properties using the conventional Neural Network.

The Machine Learning plugin can be accessed as pointed in the image below:



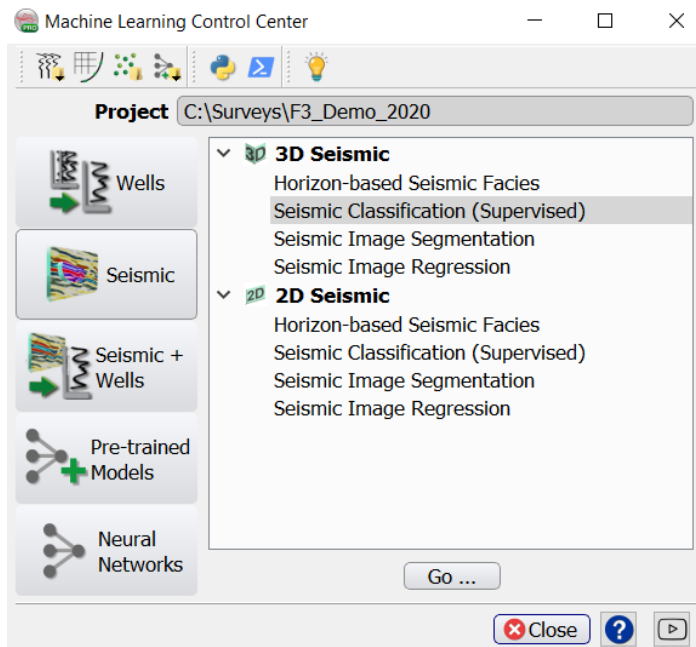
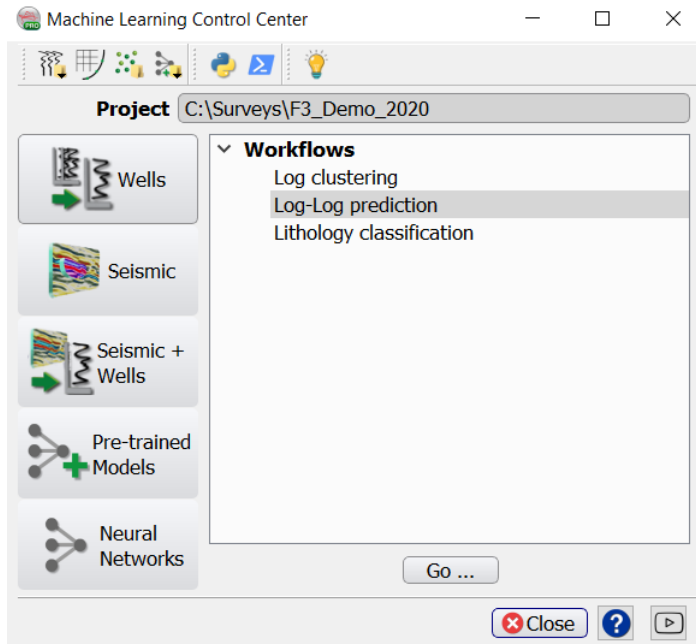
The Machine Learning Control Center is developed with the same look-and-feel as other control center plugins of OpendTect.

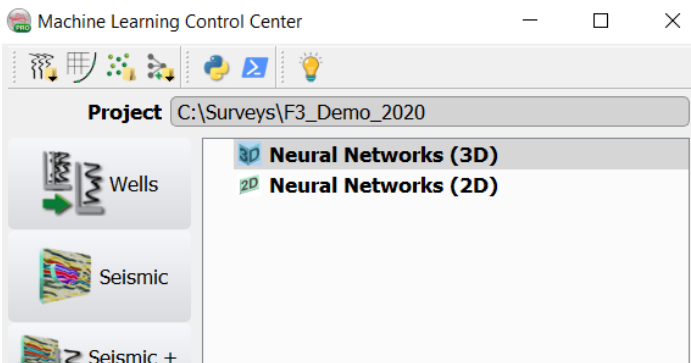
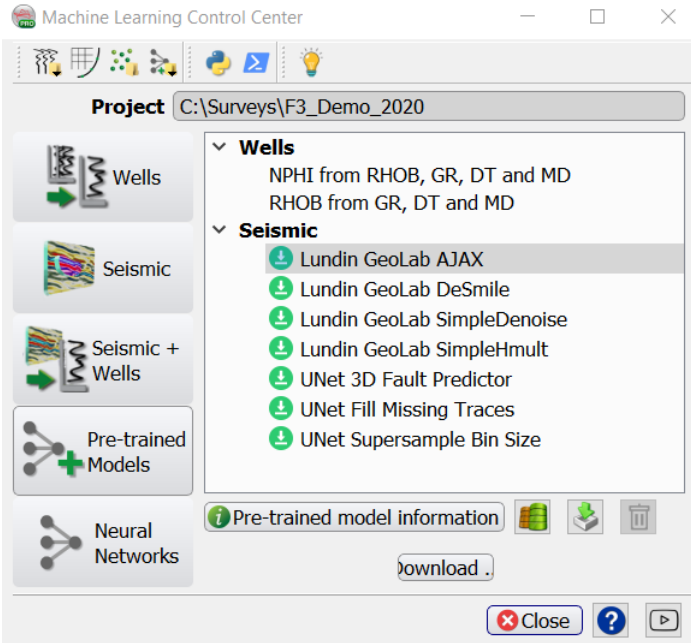
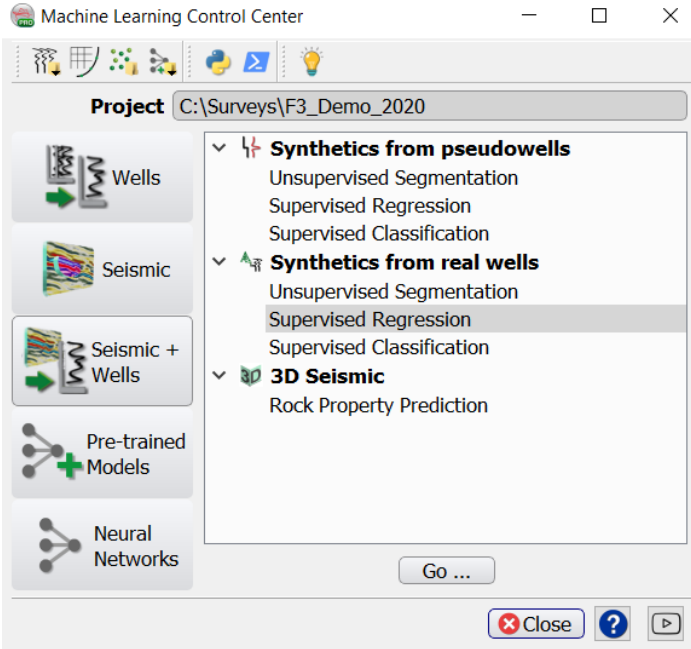
First we select the type of data we are working with: Wells, Seismic, Seismic + Wells, or Neural Networks. The latter refers to the pre-existing Neural Networks plugin that is now started from inside the ML control center.

Next, we select the workflow we want the execute from the available options (see images below). The UI then guides us through the selected workflow.

Typically, each workflow consists of three steps, which are captured in three tabs: **Select input data > Train Model > Apply Trained Model.**

ML workflows for: Wells, Seismic, Seismic + Wells, Pre-trained Models and the predecessor Neural Networks plugin:





Terminology

In this Chapter we will use the following terms:

1. Accuracy and Overfitting

- a. A Model is accurate if it can be said to predict as well in the validation data as it did in the training data.
- b. An overly complex Model may be trained such that it is very accurate for predictions on the Training Set but does not generalize well.
- c. Overfitting results when the Model no longer describes the relationship between Features and Target, but instead is trained to output only the Training data.

2. Algorithm

- a. A process, set of rules or mathematical relationships used to calculate a solution (or solve) a problem.
- b. Typical Machine Learning Algorithms are Regression, Random Forest, Boosting, Neural Networks, Self-Organizing Maps (SOMs).

3. Classification

- a. A prediction method that assigns a data point (with some feature coordinates) to a predefined category or bin.
- b. Classification is strictly speaking a Supervised learning approach (target labels are assigned by experts).
- c. Classification is nowadays also used in the context of Unsupervised learning approaches (target labels are generated by the model).
- d. In this document we use Classification for both Supervised and Unsupervised learning approaches.

4. Clustering

- a. An Unsupervised learning approach aimed at Organizing data in groups with similar structure.

5. Convolutional Neural Network (CNN)

- a. A specific type of neural network in which a (1D, 2D, 3D) image is convolved with a set of filters and the filtered output is passed on to the next layer in the network.
- b. The filter coefficients are found in a supervised learning run.
- c. The [LeNet](#) and [Unet](#) models implemented in OpendTect Machine Learning plugin are [CNNs](#).
- d. Webinar: [Machine Learning Part 2](#).

6. Feature

- a. An independent variable such an observable quantity that has been previously recorded.
- b. Features can also be engineered by combining multiple features or applying transformations.
- c. Features are the inputs of Models.

7. LeNet Model

- a. Convolutional Neural Network introduced by Yann LeCunn
- b. Used in OpendTect for Seismic Bodies detection workflows (e.g. creating a Chimney Cube with a deep learning model instead of with a shallow MLP type of neural network).
- c. Webinar: [Machine Learning Part 2](#); Publications: <http://yann.lecun.com/exdb/lenet/>

8. Model

- a. A mathematical representation of some real-world process.
- b. A predictive model attempts to produce an outcome based on past experience.

9. Multi-Layer-Perceptron (MLP)

- a. A fully-connected neural network with an input layer, one or more hidden layers and an output layer.

- b. The non-linear transformation happens in the hidden layer(s) when the sum of the weights multiplied by the input features passes through a sigmoidal shape function.
- c. [MLP](#) networks are supervised networks for classification and regression problems.
- d. MLPs are nowadays referred to as “shallow” networks as a distinction between conventional neural networks and newer deep learning models.
- e. In OpenTect MLPs are implemented in the pre-existing Neural Networks plugin and in various workflows with Scikit Learn models.
- f. Webinar: [Machine Learning Part 1](#).

10. **Segmentation**

- a. Strictly speaking Segmentation is the same as Clustering, i.e. an Unsupervised learning approach.
- b. Segmentation is nowadays also used for Supervised learning methods.
- c. We primarily use Segmentation in the context of image-to-image predictions whereby the Target is an image with labeled information (aka mask).

11. **Target**

- a. Defined in Statistics as the dependent variable.
- b. It is the output (or prediction) of the Model or variable.

12. **Test Set**

- a. A data set that is (automatically) split off by random selection from the Training Set. The Test Set is used during training to check for signs of Overfitting.
- b. Examples from the Test Set are passed through the Model to compute the error. In contrast to examples from the training set the errors of the Test Set are not used to update the weight function (gradients) during training.
- c. Training should ideally be stopped when the error on the Test Set has reached its minimum. Prolonged training may further decrease the error on the Training Set while the error on the test set increases (overfitting).

- d. The Test Set differs from the Validation set in that the Validation set is not used during training whereas the Test Set is.
- e. TensorBoard does not make a distinction between Test Set and Validation.

13. Training (“Learning”)

- a. The process of creating a Model from the Training data using some algorithm.
- b. The data is fed into the algorithm, which learns a representation of the relationship between Feature(s) and Target(s).
- c. Supervised Learning is an approach to learning under supervision from an intelligent source. For example, learn by advice, or learn by example.
- d. Unsupervised Learning is a competitive learning approach aimed at finding structure in the data, thereby extracting relevant properties or features.

14. Training Set

- a. A (cleaned up, pre-processed) data set that is used by the learning algorithm to find potentially predictive relationships.
- b. Typically, the Training data set is a subset of the entire dataset, with some portion being held aside for Validation.
- c. For seismic applications a Training Set can also be synthesized. This is especially useful in cases where we do not have sufficient reliable examples to train on. In OpendTect we synthesize data (1D, 2D and 3D) in SynthRock. Also, the Unet Fault predictor was trained on synthetic data.
- d. The key to successful machine learning applications is creating a training set that is truly representative of the problem we try to solve.

15. Unet Model

- a. An auto-encoder - decoder type of Convolutional Neural Network used in Segmentation and Regression problems.
- b. Webinar: [Machine _ Learning _ Part _ 2](#); Publications: https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28

16. Unsupervised Vector Quantizer (UVQ)

- a. Network for segmenting (clustering) data into a user-defined number of segments (UVQ is similar to K-means clustering).
- b. Cluster centers are found by unsupervised learning; Random prototype vectors are updated until they fall in cluster centers; Post-training cluster centers are sorted to ensure that similar clusters are displayed with neighboring colors in the segmentation map (UVQ + sorting approximates a Kohonen map, which sorts during training as it updates not only the winner but also the nearest neighbor).
- c. In the application phase the trained UVQ network compares input vectors to 'cluster center' vectors; Output is the index of 'winning' cluster and 'match' (confidence measure between 1 and 0).
- d. UVQ is used in seismic facies interpretation workflows. Typically, it is used to cluster seismic waveforms around mapped horizons but it can also be used to cluster 3D bodies.
- e. Quick UVQ is a workflow for waveform segmentation, i.e. clustering trace segments extracted around a mapped horizon.
- f. In OpendTect UVQ is implemented in the pre-existing Neural Networks plugin.
- g. Webinar: [Machine Learning Part 1](#).

17. Validation Set

- a. A data set separate from the Training Set, yet having the same predictive features and targets as the training data set, e.g. a well with existing logs that is set aside as blind test well. Please note that TensorBoard does not make a distinction between Test Set (see above) and Validation Set.
- b. Comparison of the predictions versus known results in the training and validation data speak to the accuracy and generality of the model.

Resources

You can find the complete [dGB Machine Learning Webinar series](#) here:

- [Machine Learning Webinars: Part 1: Theory: Introduction](#)
- [Machine Learning Webinars: Part 2: Theory: Deep Neural Networks and Other New Algorithms](#)
- [Machine Learning Webinars: Part 3: Applications: Seismic, Logs, and, Seismic-to-log](#)
- [Machine Learning Webinars: Part 4: Synthesizing Training Data with SynthRock](#)
- [Machine Learning Webinars: Part 5: Creating and Adding New Models](#)
- [Machine Learning Webinar Q&A - Demo Image to Image workflow](#)
- [Machine Learning Webinar Q&A - Demo Log-Log Prediction workflow](#)
- [Machine Learning Webinar Q&A - Demo Seismic Object Detection workflow](#)
- [OpendTect Webinar: Machine Learning workflows for seismic data interpolation](#)
- [OpendTect Webinar: Machine Learning Applications for Seismic Interpretation](#)
- [OpendTect Webinar: Seismic Classification: a Thalweg Tracker / Machine Learning Approach](#)
- [OpendTect Webinar: Develop your own Machine Learning tools and workflows with OpendTect](#)
- [OpendTect Webinar: How to prepare well logs to get optimal Machine Learning results](#)
- [OpendTect Webinar: OpendTect's Hybrid Machine Learning Solution](#)
- [Image '21 Master Class Webinar: Log-log Prediction Using Machine Learning, Seismic Classification a Thalweg Tracker & Machine Learning Approach](#)
- [Image '21 Master Class Webinar: How to extract data from OpendTect into a Python environment by coding in Jupyter Notebook](#)
- [OpendTect Machine Learning Developers Q&A Webinar: how can I add my own trained model and other Q+A](#)

- [OpendTect Machine Learning Developers Q&A Webinar: how to use my own Keras model in the ML UI?](#)
- [OpendTect Webinar: Porting Machine Learning horizon tracking Notebook to OpendTect](#)
- [OpendTect Webinar: Accelerating the Time from Machine Learning R&D to Deployment](#)
- [OpendTect Webinar: Thalweg tracker](#)
- [OpendTect Webinar: Cleaning up your data - Dip Steering and other filters](#)
- [OpendTect Webinar: Machine Learning with SynthRock Link](#)
- [OpendTect Webinar: Data Conditioning](#)
- [OpendTect Webinar: Building and Training Your 2D CNN Model with OpendTect](#)
- [OpendTect Webinar: Applying and Finetuning your Trained Model: From U-Net Architecture to Seismic Data Interpretation](#)

Other dGB resources include:

- [Machine Learning Knowledge Base](#)
- [Develop your own Models Documentation](#)
- [odpy Documentation](#)
- [Machine Learning FAQ](#)

And a selection of relevant dGB Machine Learning workflows (in .pdf):

- [Machine Learning Workflow: Wells Log-Log Prediction \(Density\)](#)
- [Machine Learning Workflow: Wells Lithology Classification](#)
- [Machine Learning Workflow: Seismic bodies \(Supervised 3D\)](#)
- [Machine Learning Workflow: Seismic Unet 3D Fault Predictor](#)
- [Machine Learning Workflow: 3D Seismic + Wells Rock Property Prediction](#)
- [Machine Learning Workflow: Seismic Image to Image Faults Prediction](#)
- [Machine Learning Workflow: Seismic Image Regression \(Unet\) - Fill Seismic Traces](#)
- [Machine Learning Workflows: Seismic Lundin Smoother](#)

Machine Learning Workflow Blogs:

Publish Date	Title	Resources
02 March 2023	Machine Learning Workflows – Quick UVQ Waveform Segmentation	Video
23 March 2023	Machine Learning Workflows - Using AI for Salt Detection	Video
30 March 2023	Machine Learning Workflows - Ready to go AI workflows - Apply Pre-trained Model	Video
06 April 2023	Machine Learning Workflows - Fast and Simple Seismic Facies Analysis - 3D UVQ Waveform Segmentation	Video
20 April 2023	Machine Learning Workflows - De-risking charge and seal issues with AI - Neural Network Chimney Cube	Video
04 May 2023	Machine Learning Workflows - Seismic Inversion using AI - Machine Driven Seismic Inversion Workflow	Video
25 May 2023	Machine Learning Workflows - Supervised AI Seismic Facies	Video

GitHub repositories:

- [dgbpy](#) is a framework for research and deployment of machine learning models from seismic and well data
- [odpy](#) is a framework for research and deployment that allows for basic interactions with the OpendTect software and database
- [OpendTect-ML-Dev](#) contains examples on how to develop your own Machine Learning tools and workflows. Also it includes examples used in webinars.

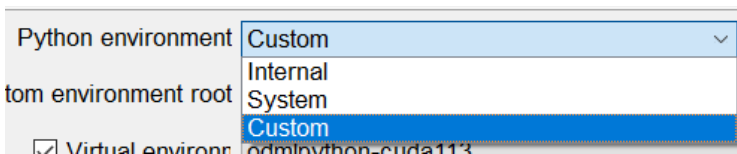
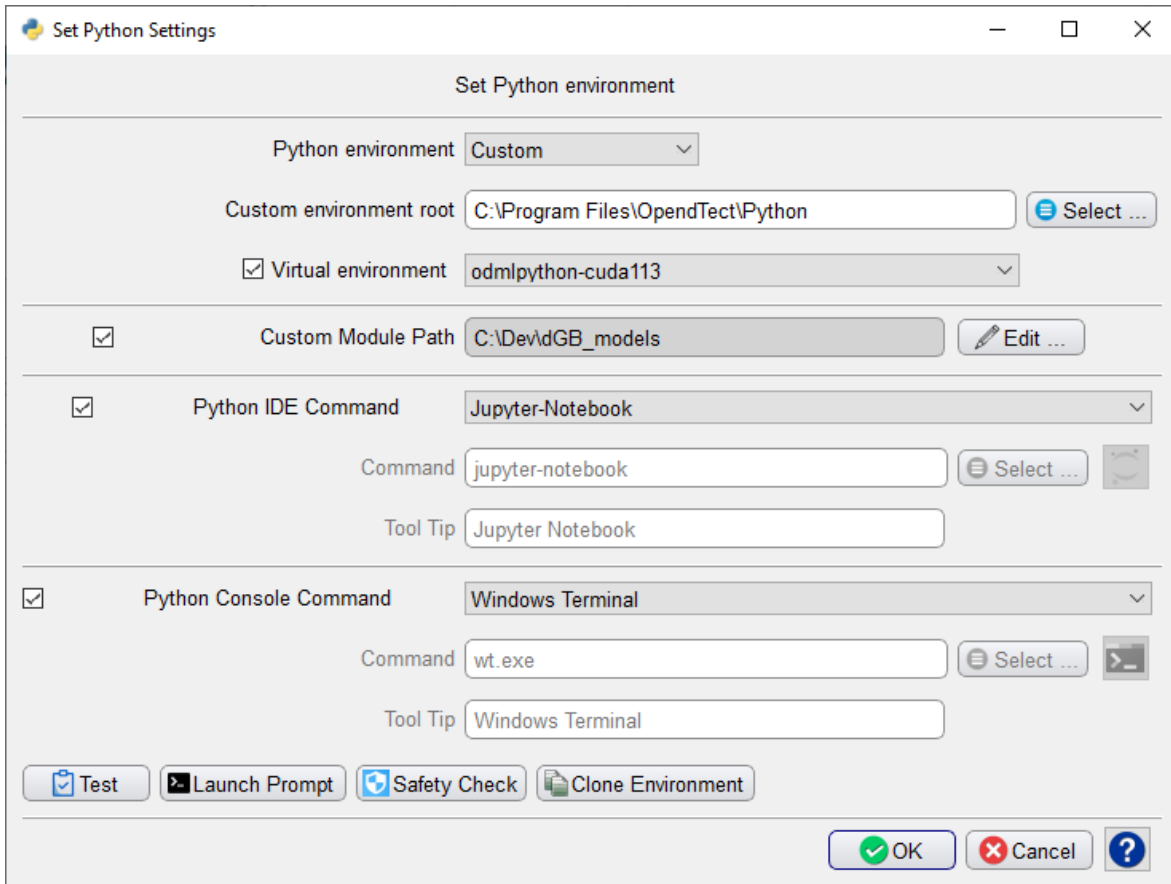
For a general (non dGB) beginner's tutorial course in TensorFlow 2.0, please watch [this YouTube video](#).

You can join the [OpendTect Machine Learning Developers' Community on Discord](#)

For more information on how to become a member and be part of the Community please read the [FAQ](#).

3.2 Python Settings, Data Flow and Data Management

The Python Settings Window is started from Utilities > Installation > Python Settings.



Users can change the **Python environment** here. OpendTect Machine Learning comes with its own Miniconda Python environment. If it has been installed then make sure that the Python environment is set to **Internal**. Users who prefer to work in a private Python environment will have to change the Python environment to **Custom**. When Custom environment is selected you need to specify the root of the environment. The remaining option **System** can be used if you want to use a system Python installation that is not a miniconda or anaconda installation. If you want more information about installing the Python environments please take a look in the [Administrator's Manual](#).

The **Virtual environment**: odm1python-cuda113 runs models on the GPU; odm1python-cpu-mkl runs on the CPU. Computations on a GPU are many factors faster than computations on a CPU. If the GPU in your environment is too small for certain jobs to run you can switch virtual environments to CPU usage.

Note: *You may notice that the GPU is not using all its available resources for certain processes. Within the Machine Learning plugin, it is the Python application that is running either the training or the prediction, not OpendTect itself. Within Python, the performance and behavior of each process (training/prediction) depends entirely on the python module being used: It will be very different between Scikit Learn (CPU only, very small memory footprint), and Tensorflow (GPU or CPU, large memory utilization). We keep monitoring for any available updates for these Python packages and will implement these newer, improved versions immediately as they become available.*

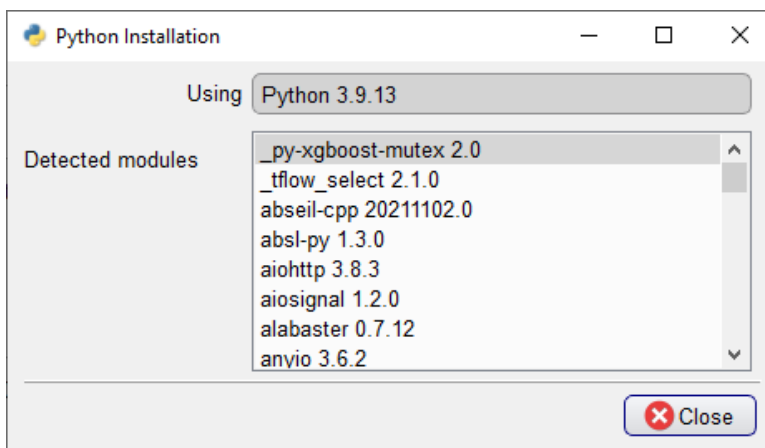
The **Custom Module Path** is the path for Python developers in which they develop private Machine Learning Models in this environment. All models in the given directory, which are named: **'mlmodel....'** will appear in the UI of the Machine Learning Control Center and can thus be applied to other data sets.

The **Python IDE Command** is the Python editor you wish to use. You can launch the IDE using the specified **Command**, Optionally with some **Arguments** from a **Python Command Window**. Alternatively, you can add an **icon** to your plugin toolbar by pressing the corresponding icon in this window. In this case the **Jupyter-Notebook icon** with Jupyter Notebook **Tool Tip** is added to the icon toolbar.


Pressing this icon from the toolbar will launch Jupyter Notebook with the correct settings.

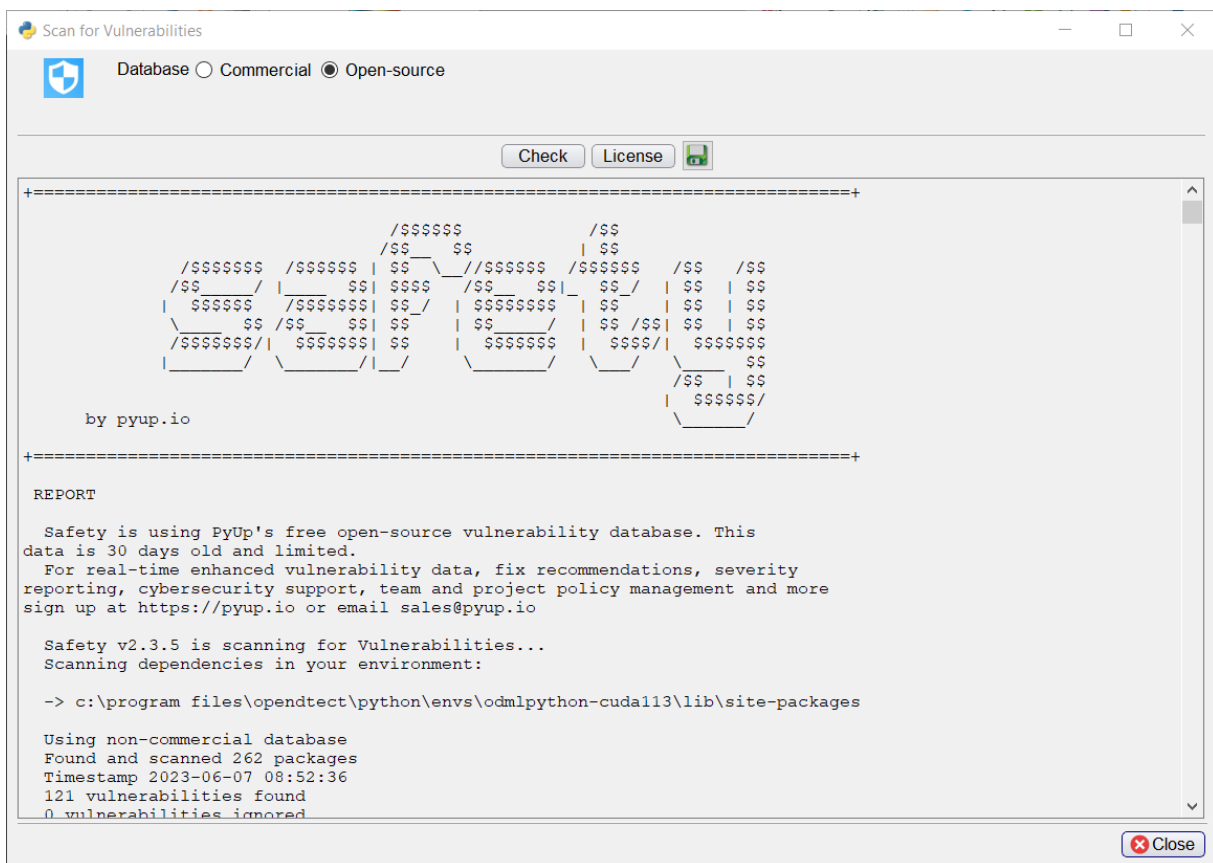
The **Python Console Command** is the console you wish to use. You can launch the console using the specified **Command**. Alternatively, you can add an **icon** to your plugin toolbar by pressing the corresponding icon in this window. In this case the **Windows Terminal icon** with Windows Terminal **Tool Tip** is added to the icon toolbar.

The Python environment can be tested with the **Test button**. If the test is successful it will show the Python version and a list of installed modules.



The Python Console can be launched with the **Launch Prompt** button.

The Python modules in the Python environment can be scanned for vulnerabilities and checked for licenses with the **Safety Check** button. You can either use PyUp's **Commercial** or **Open-source** database. The free open-source database is 30 days old and limited. The Commercial database provides real-time enhanced vulnerability data and fix recommendations. In order to use the Commercial database or the scan for licenses feature you need a valid **API key**, which can be filled in the API key field. For more information about the API key please send an email to info@dgbes.com. With the  icon the safety report can be saved.



```

+-----+
                    /$$$$$$          /$$
                   /$$   $$         | $$
  /$$$$$$$/  /$$$$$$ | $$ \___/  /$$$$$$$/  /$$$$$$  /$$  /$$
 /$$   $$/  |_____ $$ |  $$$$/  /$$   $$/  | $$  | $$
 |  $$$$$$ /$$$$$$$$ | $$ \___/  |  $$$$$$/  | $$  | $$
 \_____ $$ /$$   $$ | $$   |  $$$$/  | $$ /$$ | $$  | $$
 /$$$$$$$/ |  $$$$$$/ | $$   |  $$$$$$/  |  $$$$$$/  |  $$$$$$/
 |_____ /  \_____ / |___/  \_____ /  \_____ /  \_____ /
                                             /$$   |  $$
                                             |  $$$$$$/

by pyup.io
+-----+

REPORT

Safety v2.3.5 is scanning for Licenses...
Scanning licenses in your environment:

-> c:\program files\opendtect\python\envs\odmlpython-cuda113\lib\site-packages

Using an API KEY and the PyUp Commercial database
Found and scanned 262 packages
Timestamp 2023-06-07 08:58:35
0 vulnerabilities found
0 vulnerabilities ignored
0 remediations recommended
+-----+
LICENSES
+-----+


```

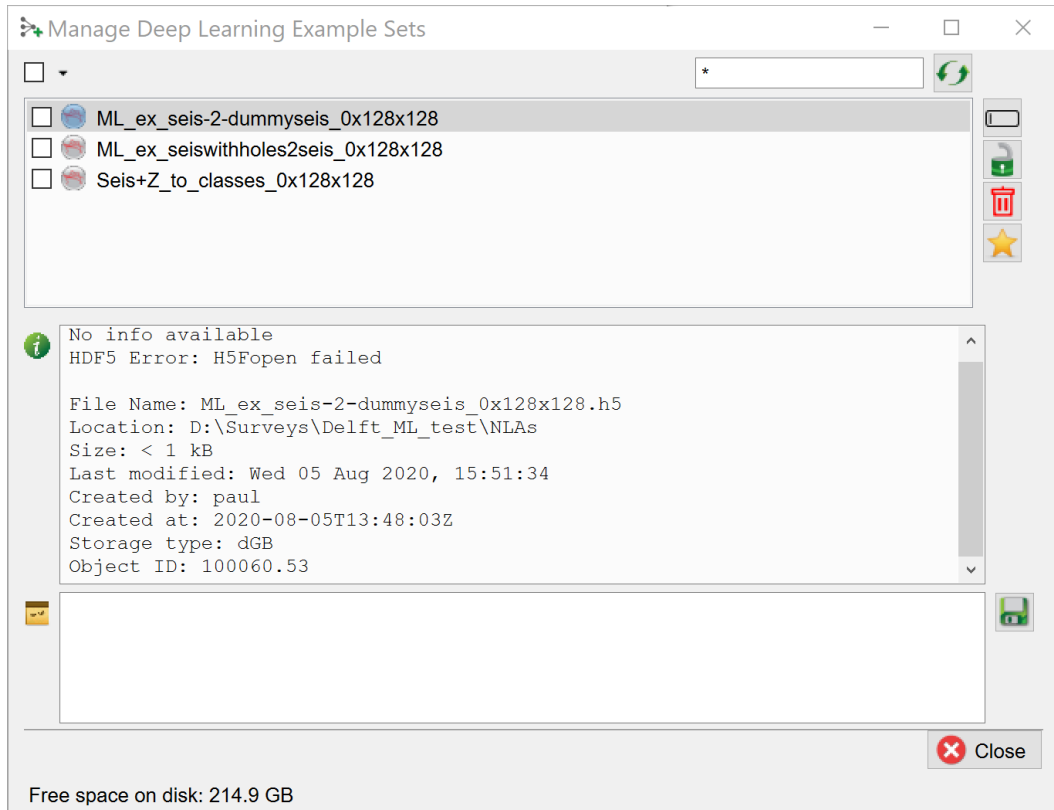
The Python environment can be cloned by clicking the **Clone Environment** button.

The general flow of a Machine Learning workflow is as follows:


1. Based on the data you have and the problem you want to solve: **Select a workflow** from the control center
2. **Create a Training Set.** Construction starts with a selection of the Target (output) feature. Next you select the input features and the dimensions of the input features. Training Sets can be constructed from real data over multiple surveys or from synthetic data (SynthRock). Training Sets selection files can be saved, restored and edited. Training Sets themselves are stored in [hdf5 format](#). These files can be managed from the **Manage Machine Learning icon**.

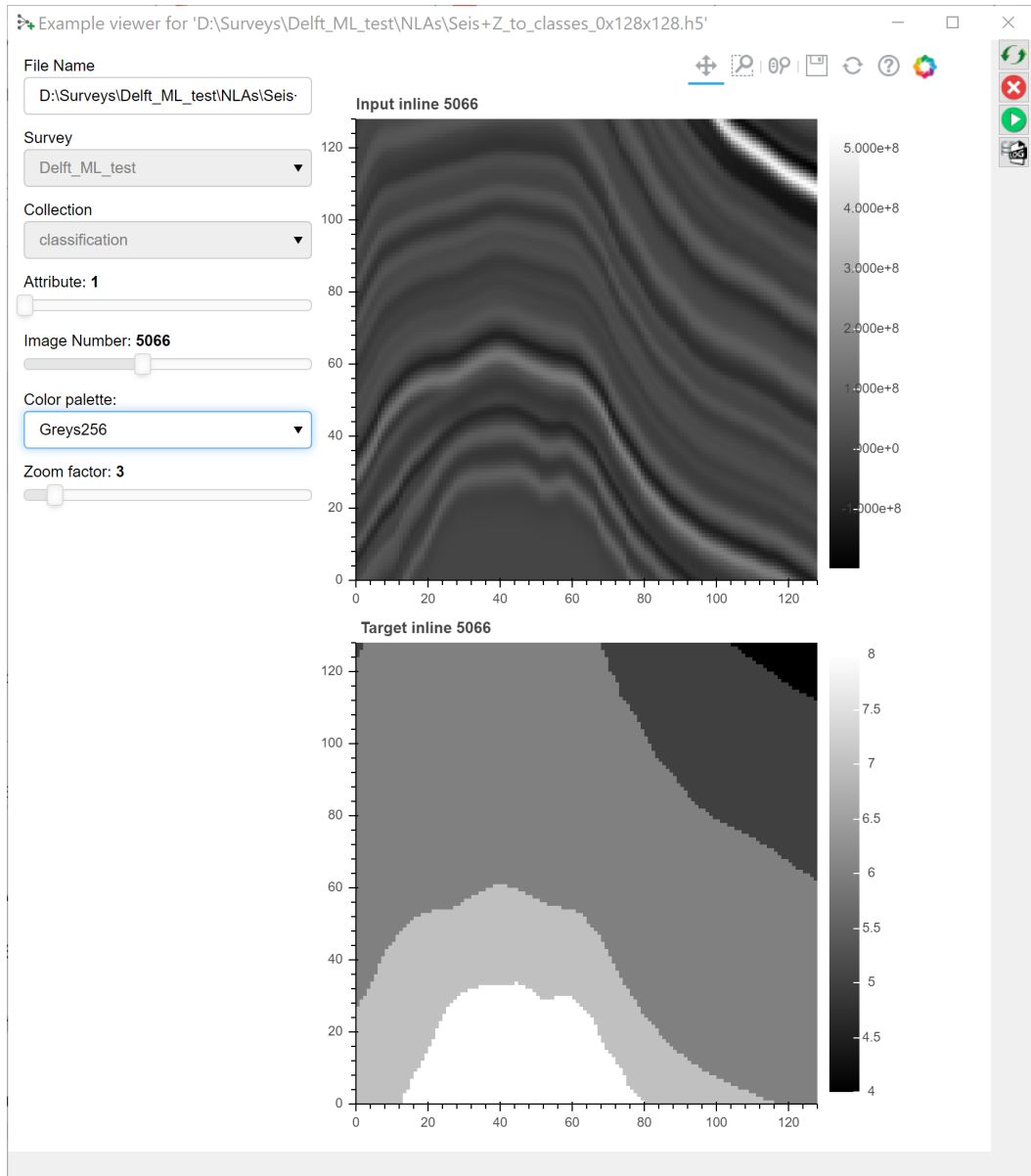
3. **Select a Model.** Depending on the workflow the plugin supports Machine Learning Models from **Scikit Learn** , **PyTorch** and/or from **Keras (TensorFlow)**. Set the **Training Parameters** and train the model. A Test Set to monitor Overfitting is automatically split off from the Training Set by the software. Models are also stored in hdf5 format and can be managed from the **Manage Machine Learning icon**.
4. Monitor training. This is done in a **log file** that outputs information about the loss and accuracy of the model for both Training and Test Sets. **Keras** models are also monitored graphically in **TensorBoard**, which is started in your default browser when training starts. Models can be trained from scratch (**New**), continued from a stored model (**Restore**) and continued from a trained model with new data (**Transfer**). In Transfer training the convolutional parts of a deep learning model are not updated to save time. Only the weights of the last layer are updated with the new training examples.
5. **Apply** the trained model. Select the input data set(s) on which the trained models will be applied.

Training Sets can be **managed** in the “Machine Learning Deep Example Sets” window that is launched from all windows with a “Manage Example Sets” icon  next to a Select button. The “Machine Learning Deep Example Sets” window pops up.



In this window you can use the corresponding icons in the ribbon on the right to rename, lock, remove and set defaults. The info box in the middle gives detailed information on the file. Personal textual information can be added in the bottom field. Press the save button to save this information with the file.

Example data sets can be viewed with an **hdf5** viewer that is launched from all windows with the “View Example Sets” icon  next to the “Manage Example Sets” icon. The “View Example Sets” viewer pops up.



The hdf5 viewer is a **2D viewer** that allows you to inspect 2D and 3D input - and target images. Use the sliders to select the input **attribute** (typically 1, but more are possible) and the **input** image to view. In the case of 3D images (3D cubelets) you can slide through the selected cubelet in the inline, crossline and Z directions with 3 additional sliders.

The display can be changed by changing the **color palette** and the **zoom factor**.

Note, the hdf5 viewer is developed in Bokeh, a Python library for interactive displays. The icons above the image are standard **Bokeh plot icons** to control zoom, pan, reset etc.

The **Bokeh server icons** in the ribbon on the right-hand side of the viewer can be used to start, stop and restart the Bokeh server in case something goes wrong. The Bokeh server is a separate process that is started by OpendTect. Information about this process is given in the log file that can be inspected by pressing the corresponding icon.

3.3 Wells

The well log prediction and well log classification workflows in this section are Supervised Machine Learning workflows and require only well logs as inputs. The target logs are used for generating/predicting missing logs, and for classifying lithological logs.

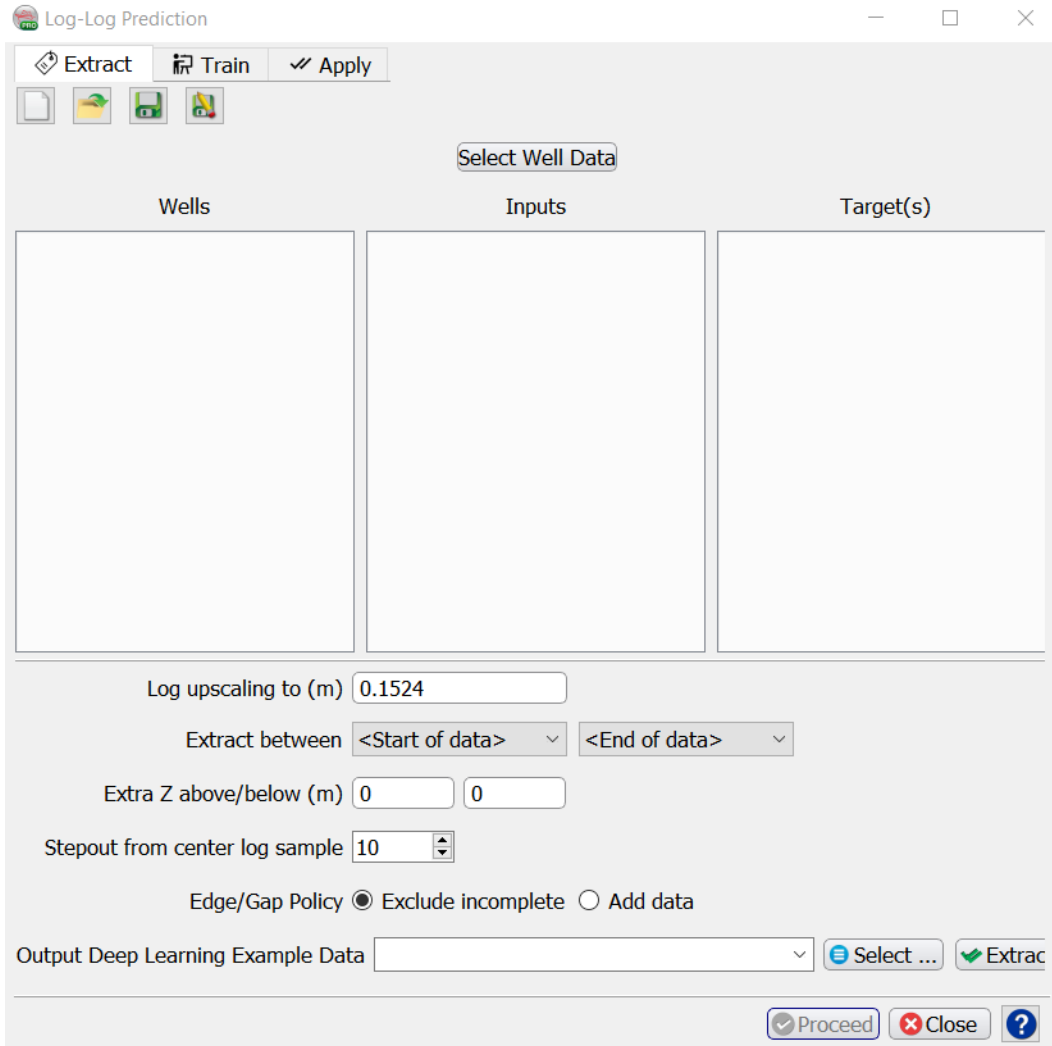
Log Prediction produces continuous logs and **Log Classification** produces discrete logs.

3.3.1 Log-Log Prediction

This workflow is used to predict a missing log from existing logs. The input and target logs must be present in all training wells. For the application of the trained model, only the input logs must be present. The Data is extracted from one or more wells. Typically, the target log is predicted from multiple input logs.

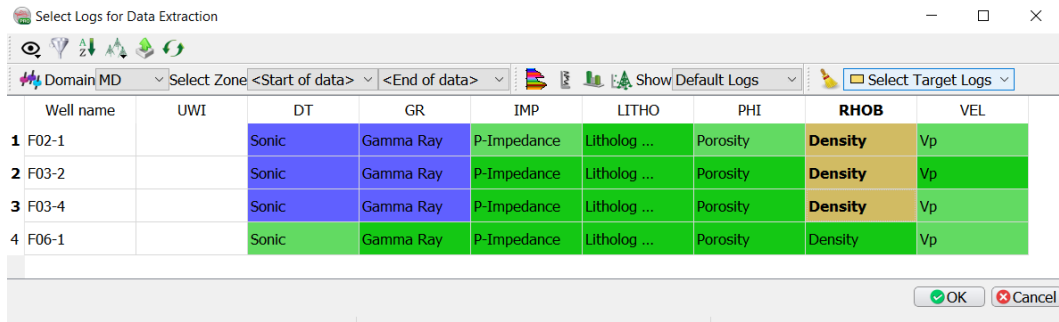
Extract Data

Data extraction is the first step in all Machine Learning workflows, and as such, we will work through extracting the well curve data that we will use later in the training and predictions steps. We will divide the data into training and validation sets, where the training data will be used in the learning stage and generation of a trained model, and the validation data will be used to test our model with data that has not yet been seen by the trained model.



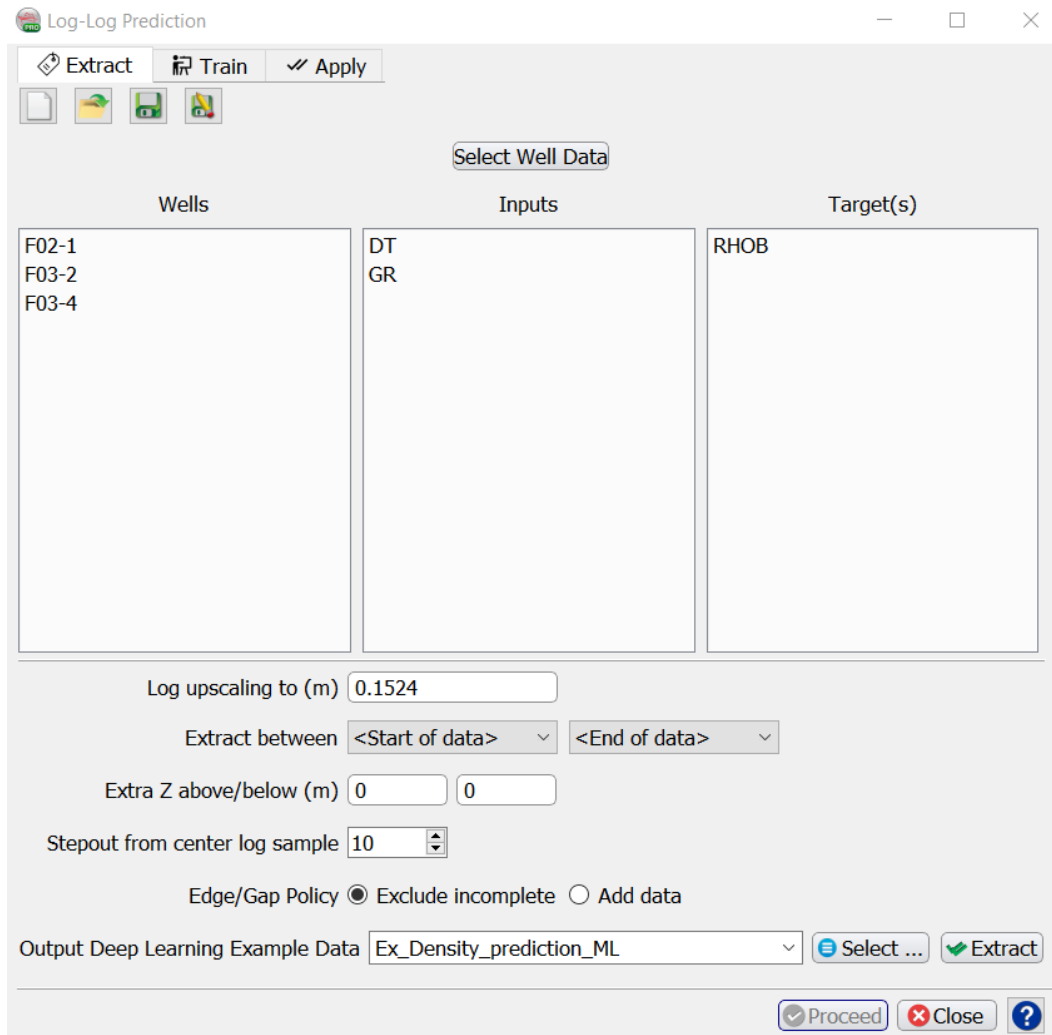
Log-Log Prediction Window

Select the Input and Target Data:



In this example, we select three wells for our training data, the Density logs as our **Target log**. The fourth well (F06-01) is left out, because we want to use it as a blind test to validate prediction of the trained model at a later stage. It may be that we also leave out wells because they do not have the required combination of input logs and target log.

Note: use the “**Well Table**” (launched from the ‘Select Well Data’) to see which logs are present in which wells.



The selected input wells, logs and target log names are listed.

In this example, the **Sonic** and **Gamma** logs are chosen as input logs and the **Density** as target log.

As a reminder: These selected input logs must be present in each of the Input wells and in the target well in which we want to predict **Density** Log.

The **Stepout** parameter helps the model to take small trends into account, and the specified value determines the input log samples to extract around each evaluation point (in a +/- range around the specified value).

To extract only in a zone of interest, we can select the corresponding markers to **Extract between** a pair of specified start and end points. If left to defaults, the entire log will be used.

The **Edge/Gap Policy** controls how we handle examples with missing input features. With a **Stepout** of 10 we need 10 samples above and 10 samples below each prediction point. The default **Exclude incomplete** will not return a value meaning we lose 10 samples above and 10 samples below each gap. If the toggle is set to **Add data**, the value at the edge of an input feature is copied Stepout times to ensure the model can make a prediction at the edge. In other words the predicted log covers the same interval as the input logs (gaps do not increase).

Hitting the **Proceed** button completes the Data extraction step. The extracted data is stored and the training tab pops up.

Training

When we already have stored the extracted data for training, we can move on to the **Training tab** with **Select** training data (Input Deep Learning Example data). After the training data is selected the UI shows which models are available.

Under the Select training data there are three toggles controlling the **Training Type**:

- **New** starts from a randomized initial state.
- **Resume** starts from a saved (partly trained) network. This is used to continue training if the network has not fully converged yet.
- **Transfer** starts from a trained network that is offered new training data. Weights attached to Convolutional Layers are not updated in transfer training. Only weights attached to the last layer (typically a Dense layer) are updated.

These available models are divided over three platforms: **Scikit Learn**, **Keras (TensorFlow)** and **PyTorch**.

Check the **Parameters** tab to see which models are supported and which parameters can be changed. From **Scikit Learn** we currently support:

1. Linear Regression
 - a. Ordinary Least Squares
2. Ensemble Methods
 - a. Random Forests
 - b. Gradient Boosting

- c. Adaboost
- d. XGBoost
- 3. Neural Networks
 - a. Multi-Layer Perceptrons
- 4. Support Vector Machines
 - a. Linear
 - b. Polynomial
 - c. Radial Basis Functions
 - d. Sigmoid

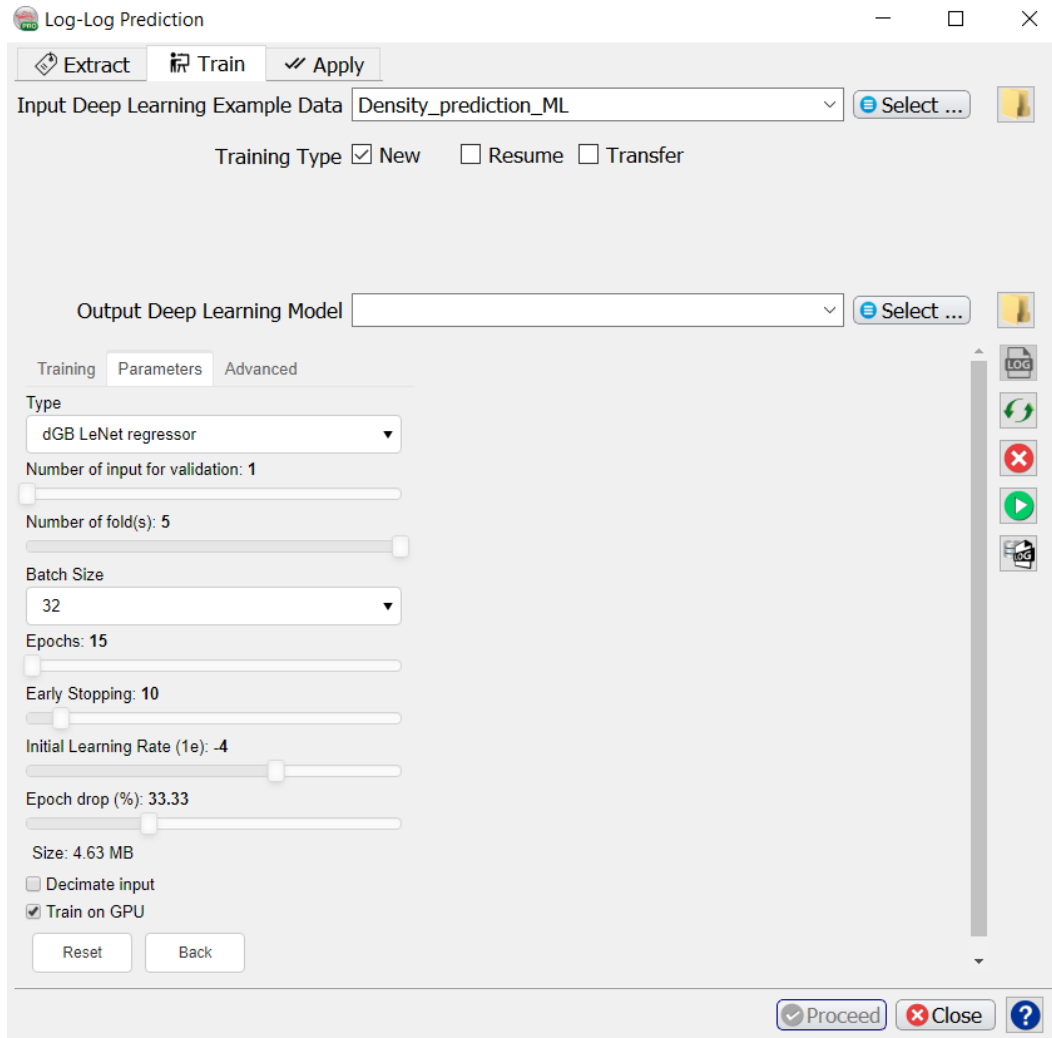
Our best results have been generated with Random Forests and XGBoost (eXtreme Gradient Boosting of Random Forest Models). Random Forests build a bunch of trees at once and average the results out at the end, while gradient boosting generates and adds one tree at a time. Because of this gradient boosting is more difficult for parameter tuning, but since it goes more in depth, it could result in higher accuracy for complicated or imbalanced datasets.

From a statistics point of view, a Random Forest is a 'bagging' algorithm, meaning that it combines several high variance, low-bias individual models to improve overall performance. Gradient boosting is a 'boosting' algorithm, meaning that it combines several high-bias, low variance individual models to improve overall performance.

For parameter details we refer to [the Scikit Learn User Guide](#).

The dGB LeNet regressor is a fairly standard Convolutional Neural Network that is based on the well-known LeNet architecture (and available in Keras-TensorFlow) It should be noted that satisfactory log-log prediction results have not been generated with this model. It is included here to give

researchers a starting point for log-log prediction with Keras / TensorFlow models. We encourage workers to use this model as a starting point, and perhaps find a better one (that we will gladly add to the list of supported models.) Tip: other workers have reported good results with LSTM type of models.



The above image shows the following training parameters that can be set.

Batch Size: this is the number of examples that are passed through the network after which the model weights are updated. This value should be set as high as possible to increase the representativeness of the samples on which the gradient is computed, but low enough to have all these samples fit within the memory of the training device (much smaller for the GPU than the CPU). If we run out of memory (raises a Python OutOfMemory exception), lower the batch size!

Note that if the model upscales the samples by a factor 1000 for instance on any layer of the model, the memory requirements will be upscaled too. Hence a typical 3D Unet model of size 128-128-128 will consume up to 8GB of (CPU or GPU) RAM.

Epochs: this is the number of update cycles through the entire training set. The number of epochs to use depends on the complexity of the problem. Relatively simple CNN networks may converge in 3 epochs. More complex networks may need 30 epochs, or even hundreds of epochs. Note, that training can be done in steps. Saved networks can be trained further when you toggle **Resume**.

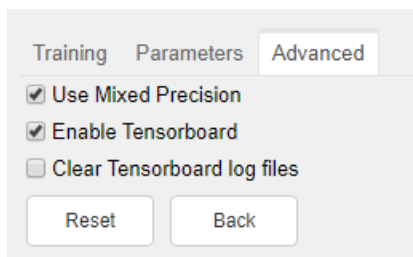
Early Stopping: this parameter controls early stopping when the model does not change anymore. Increase the value to avoid early stopping.

Initial Learning Rate: this parameter controls how fast the weights are updated. Too low means the network may not train; too high means the network may overshoot and not find the global minimum.

Epoch drop: controls how the learning rate decays over time.

Decimate Input: This parameter is useful when we run into memory problems. If we decimate the input the program will divide the training examples in chunks using random selection. The training is then run over chunks per epoch meaning the model will eventually have seen all samples once within one epoch, but only no more than one chunk of samples will be loaded in RAM while the training is performed.

Advanced Parameters



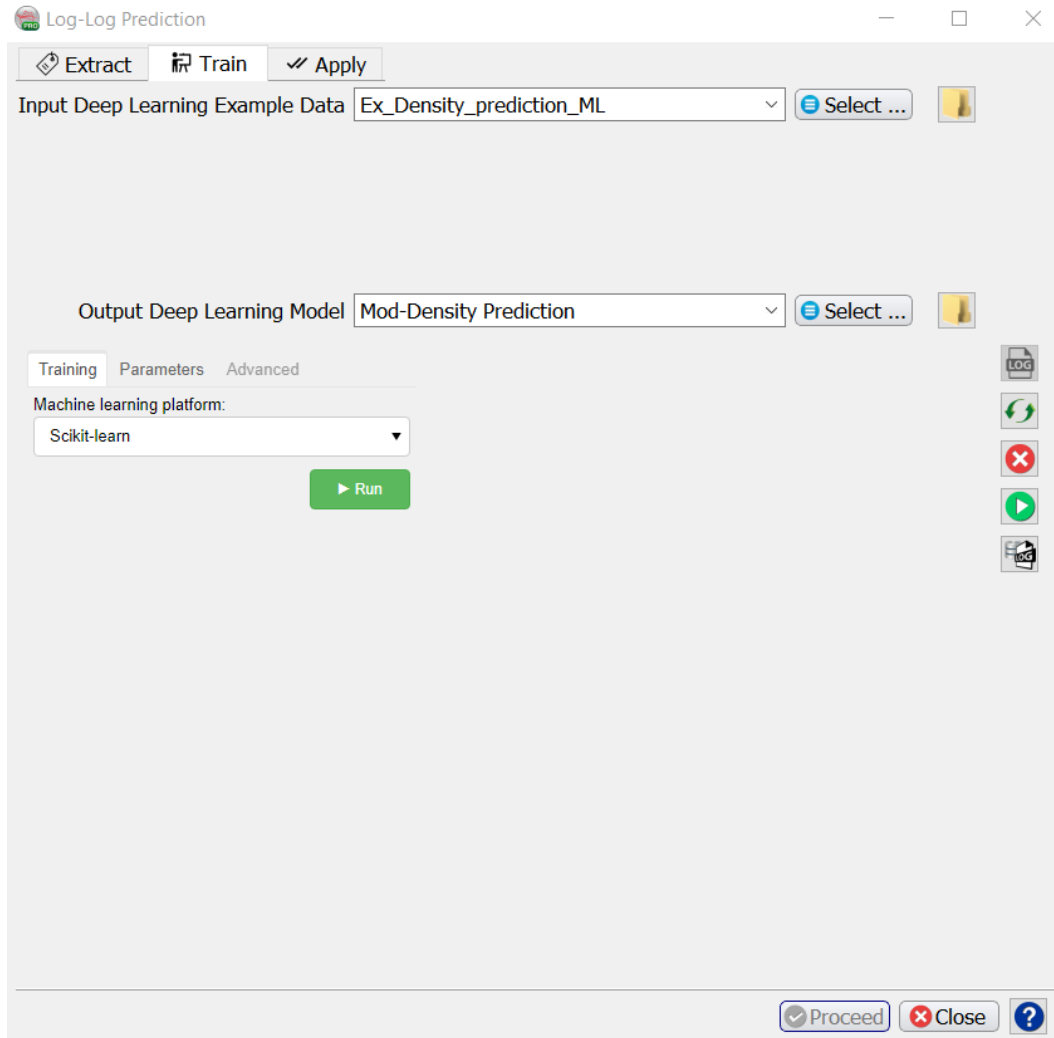
The image shows a screenshot of a web-based training interface. At the top, there are three tabs: 'Training', 'Parameters', and 'Advanced'. The 'Advanced' tab is selected. Below the tabs, there are three checkboxes: 'Use Mixed Precision' (checked), 'Enable Tensorboard' (checked), and 'Clear Tensorboard log files' (unchecked). At the bottom of the panel, there are two buttons: 'Reset' and 'Back'.

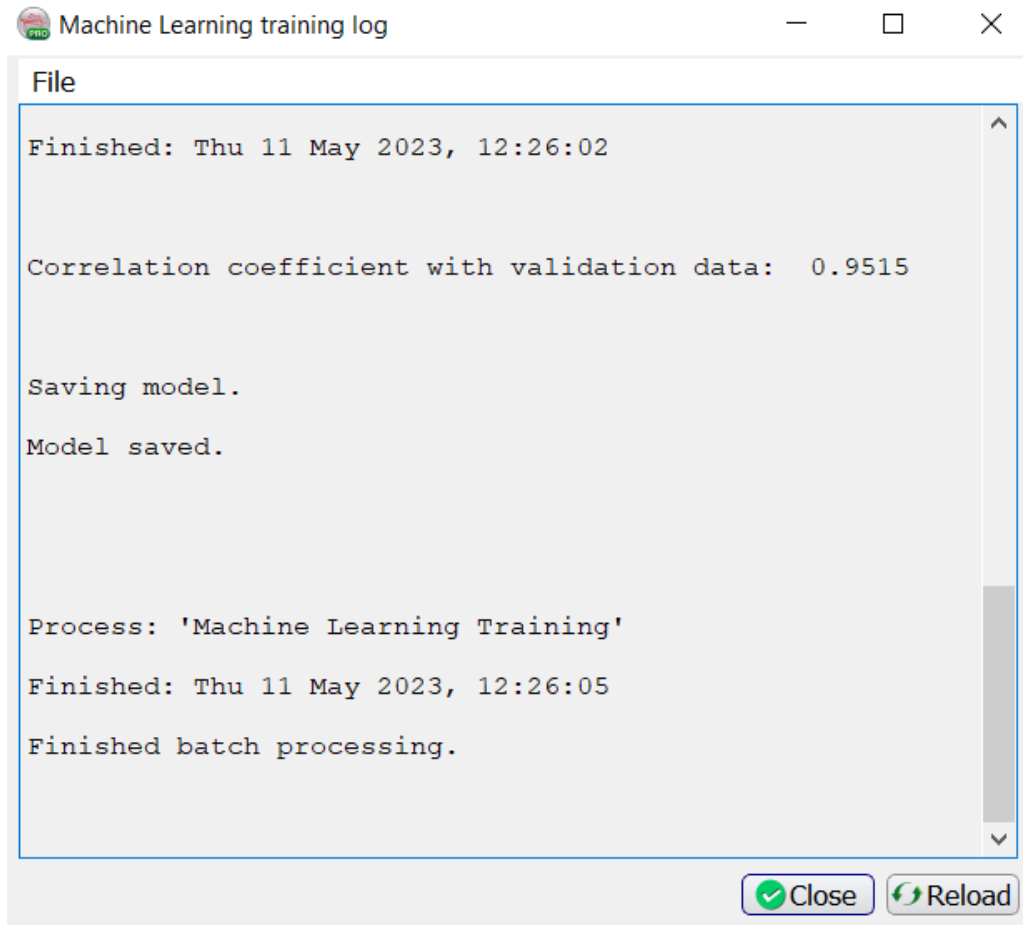
Mixed precision: Mixed precision option has been added to dgbpy for both PyTorch and Keras platform. This allows to train models using a combination of 16-bit and 32-bit precision formats which will optimize GPU memory usage and efficiency. The GPU memory usage can be reduced by 50% and 75% in some cases. It also increases training speed 2 - 3 times on GPUs with Volta or Turing architectures. This doesn't also affect training accuracy or performance.

For more details about what is 'Mixed precision training' please refer to:

- [Keras documentation: Mixed precision](#)
- [NVIDIA documentation: Train with Mixed Precision](#)

Return to the **Training tab**, specify the **Output Deep Learning model** and press the green **Run** button.

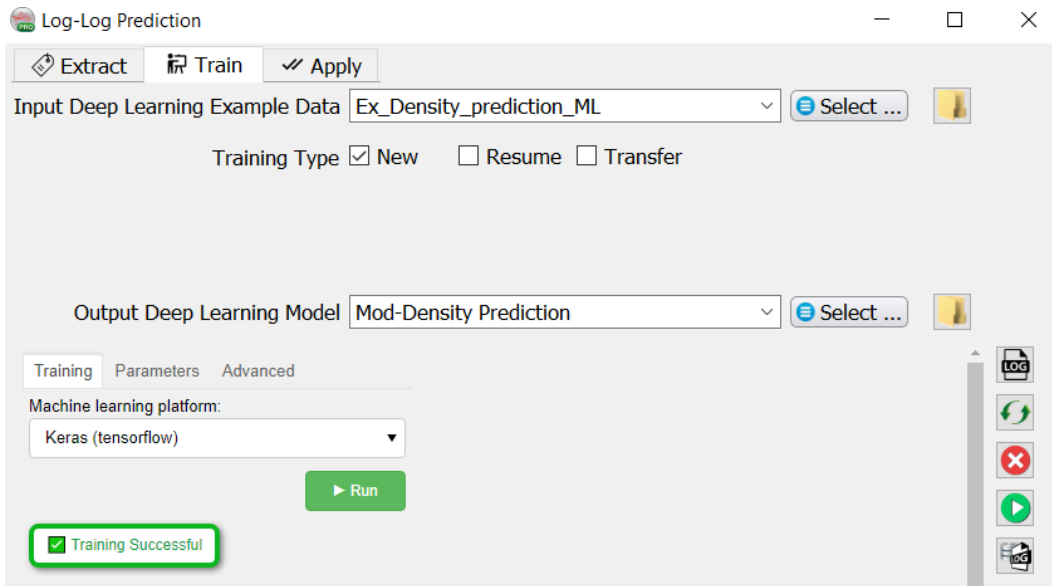




The progress can be followed in a log file. If this log file does not start automatically, please press the **log file icon** in the toolbar on the right-hand side of the window.

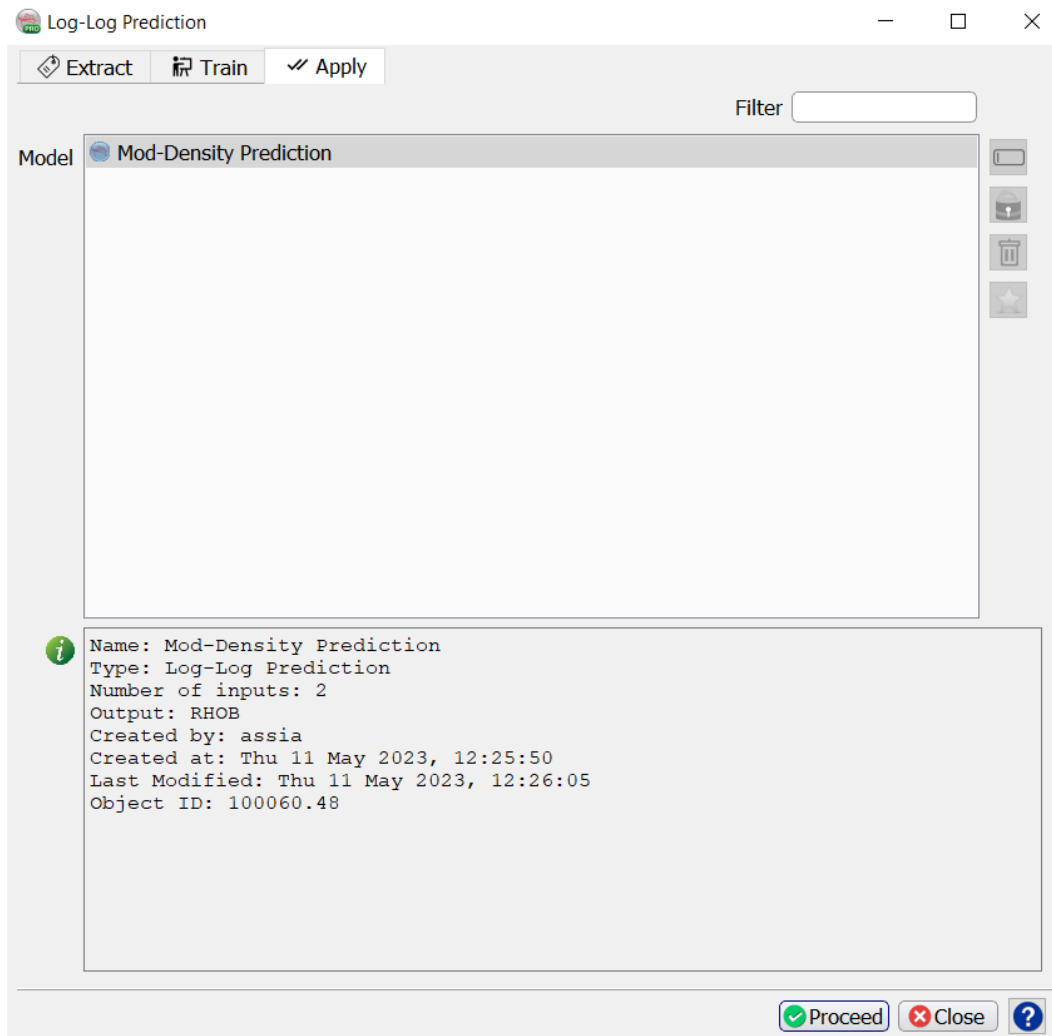
Below the log file icon there is a **Reset** button to reload the window. Below this there are three additional icons that control the **Bokeh server**, which controls the communication with the Python side of the Machine Learning plugin. The server should start automatically. In case of problems it can be controlled manually via the **Start** and **Stop** icons. The current status of the Bokeh server can be checked by viewing the **Bokeh server log file**.

When the processing log file shows “Finished batch processing” and the Training tab shows “Training Successful” we can move to the **Apply** tab.

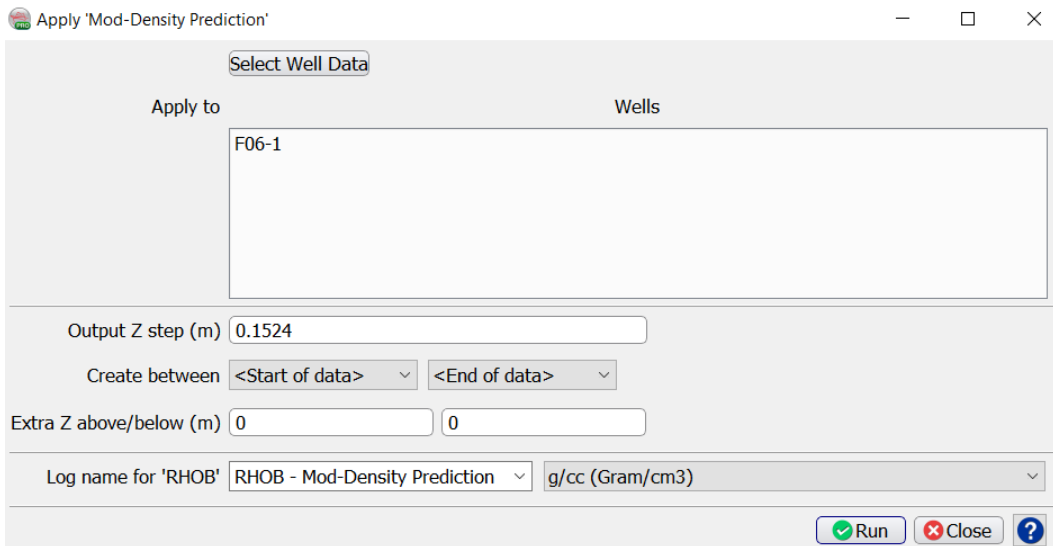
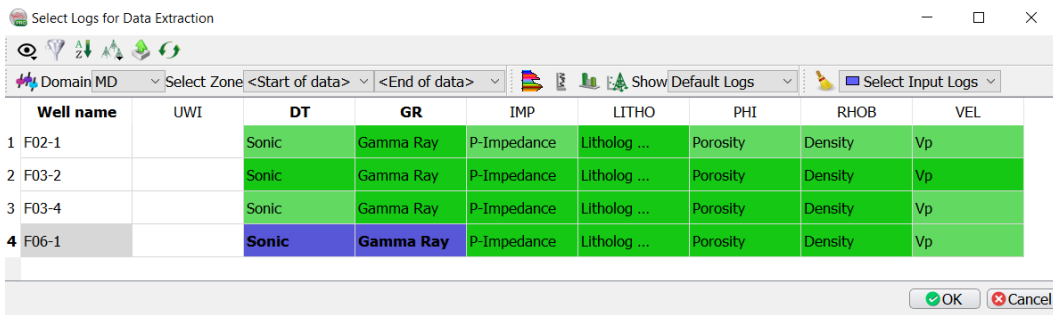


Apply

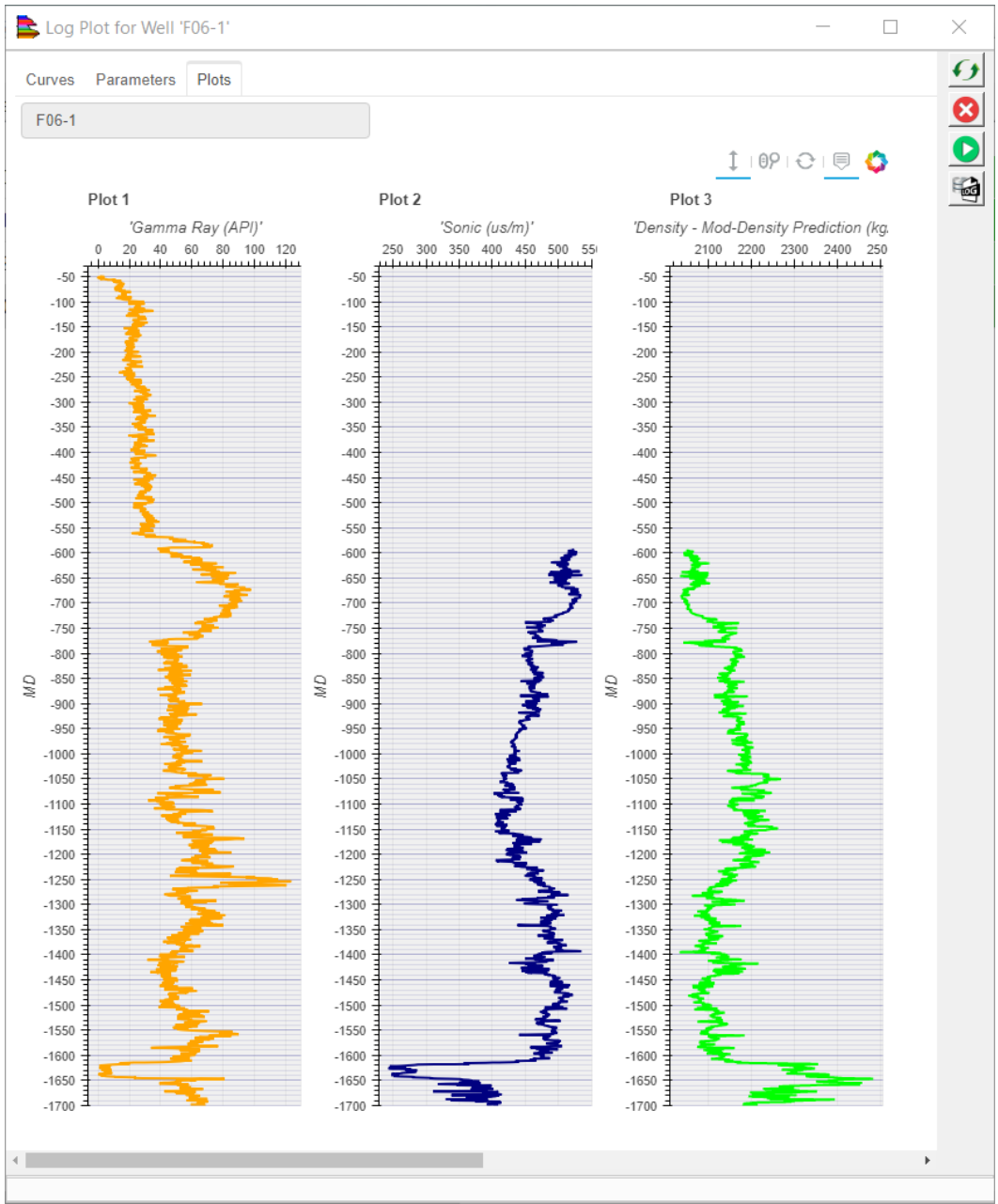
Once the Training is done, the trained model can be applied. Select the trained model and press **Proceed**.



Select the input logs and wells in which to predict the missing target log.



Press **Run** to predict the missing log. In this case a Density is predicted for the blind test well (F06-1).

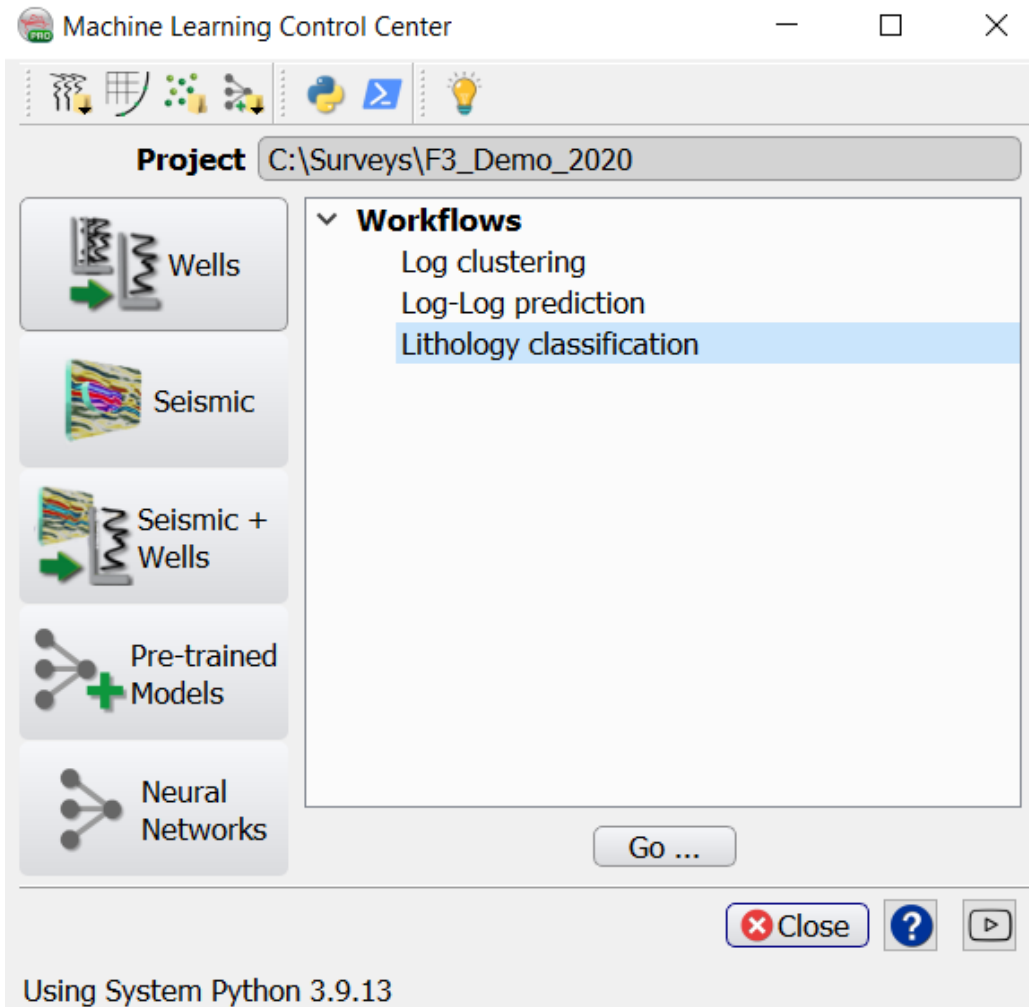


Input: Gamma and Sonic logs, **Predicted:** Density log

3.3.2 Lithology Classification

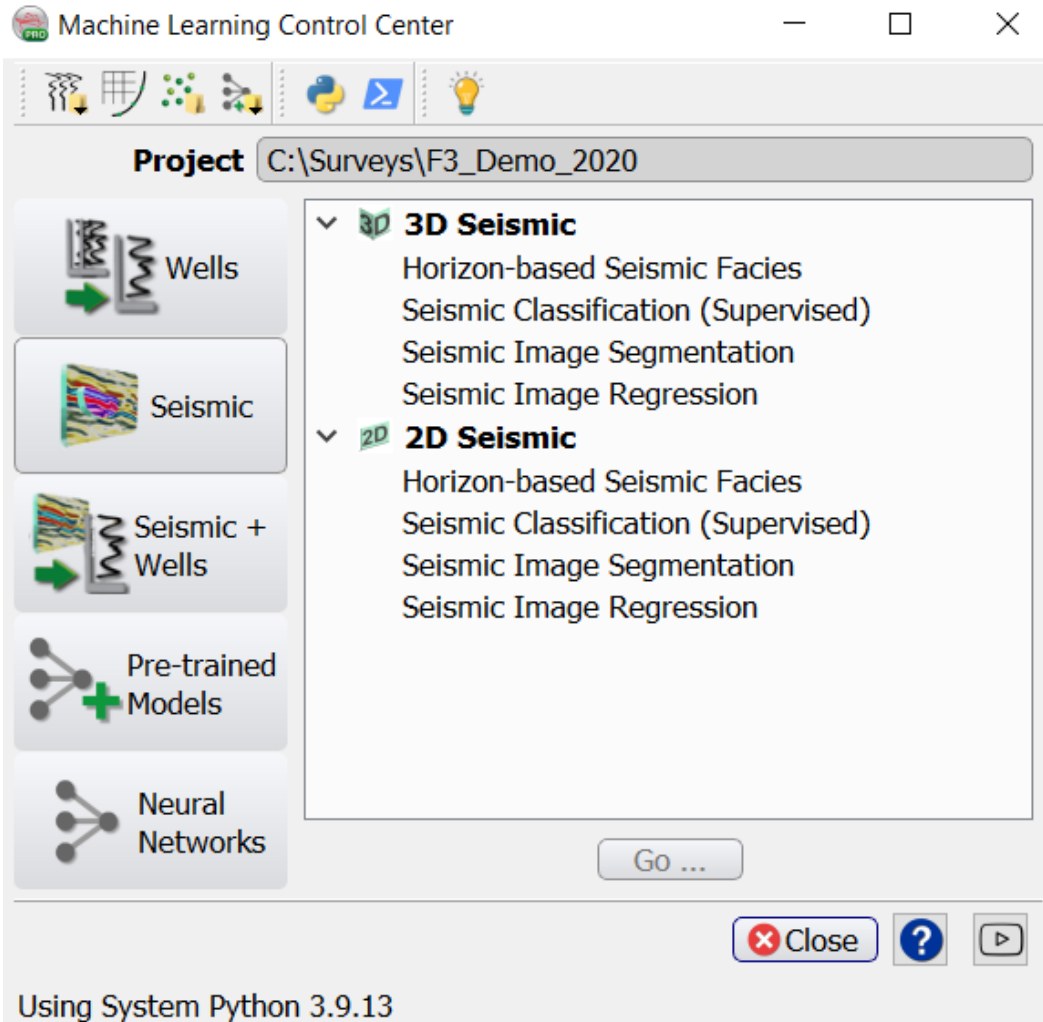
This workflow is very similar to [log-log prediction](#). The only difference is that the target log is not a continuous value (e.g. density) but a discrete number (e.g. 1, 2, 3, 4 whereby each number represents a lithology class). The values must be integers, but do not necessarily need to start at 0 or 1, and do not need either to be following each other, i.e. there can be 'gaps' in the numbering. For details, please follow the log-log prediction workflow.

Please note that lithology logs are relatively rare. If no lithology logs are available, we may consider synthesizing pseudo-wells with SynthRock.



3.4 Seismic

Under the **Seismic** button you will find workflows for 3D seismic and 2D seismic.



The 2D and 3D workflows are grouped into:

- **Horizon-based seismic facies:** this is the original “Quick UVQ” waveform segmentation workflow that can also be started from a horizon in the tree using the Workflows-> Quick UVQ menu. Waveforms are extracted along a mapped horizon and clustered into a user-defined number of clusters. The output is a waveform segmentation map and a match map (0 to 1) that shows the confidence in the clustering.
- **Seismic Classification (Supervised):** These are the deep learning equivalents of the Neural Networks seismic body workflows (Chimney Cube, Salt Cube, ...).

The difference is the use of a Convolutional Network instead of a shallow MLP network. The input is a seismic image (1D, 2D, 3D) that is mapped to a single point representing a class (e.g. Chimney-Yes or Chimney-No). The targets are given by pointsets.

- **Seismic Image Segmentation:** These workflows map (1D, 2D, 3D) images to (1D, 2D, 3D) masks of equivalent size. For example: you have a 3D seismic volume and a 3D mask volume with values 1 where there is a fault and value 0 elsewhere. You then create a training set by extracting cubelets of say 128x128x128 samples from both volumes and you train a U-net to map seismic to faults. Application of the trained model (to an unseen data set) delivers the desired output: a Fault likelihood volume.
- **Seismic Image Regression:** These workflows map (1D, 2D, 3D) images to (1D, 2D, 3D) images of equivalent size. For example: you have a 3D Acoustic Impedance volume and a 3D target volume of let's say Pore Volume. You then create a training set by extracting cubelets of say 128x128x128 samples from both volumes and you train a U-net to map Acoustic Impedance to Pore Volume. Application of the trained model (to an unseen data set) delivers the desired output: a Pore volume cube.

3.4.1 Horizon-based Seismic Facies (2D and 3D)

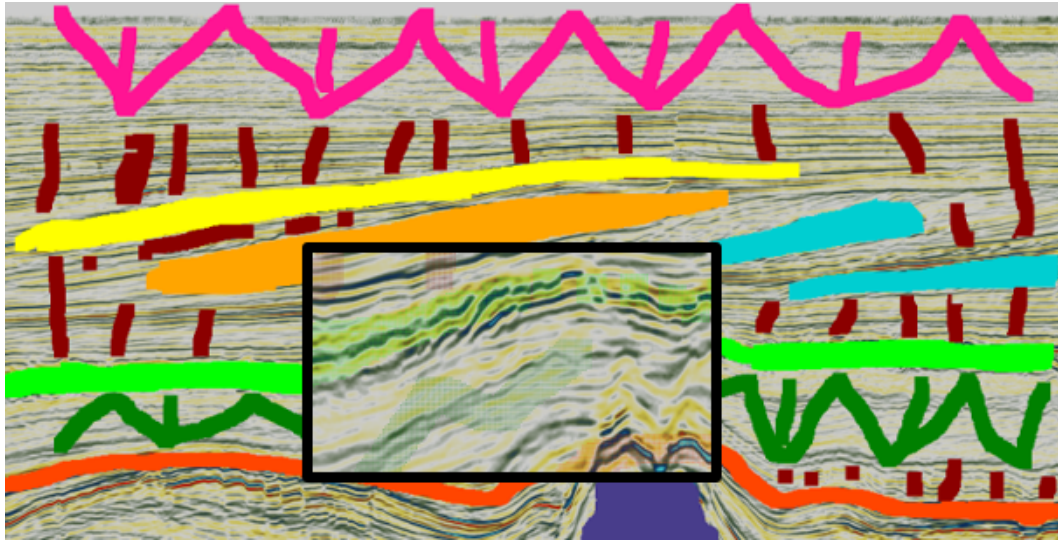
This workflow clusters seismic waveforms along a mapped (2D or 3D) horizon. The clustering algorithm is an Unsupervised Vector Quantizer. The workflow is described in the OpendTect dGB Plugins Doc [Chapter 3.6.5.1.1 Quick UVQ](#).

3.4.2 Seismic Classification (Supervised 2D and 3D)

This workflow maps a seismic image (1D, 2D, or 3D) to a single point that represents a certain class. This is a supervised method in which the user provides the desired output in the form of a set of pointsets. Each pointset represents a different class. Application of the trained model delivers a classification cube (or 2D line set).

We will discuss the workflow and UI on the basis of a 3D seismic classification into 9 different seismic facies classes. Each class represents a seismic class interpreted on one line. The pointsets for each case can be picked manually, or they can be created automatically using the tools in OpendTect (e.g. pointsets can be generated by sampling between mapped horizons, or by sampling inside interpreted 3D bodies).

In this case the point sets were copied from the MalenoV example by: Rutherford Ildstad, C., and Bormann, P., 2017. MalenoV: Tool for training and classifying SEG-Y seismic facies using deep neural networks: <https://github.com/bolgebrygg/MalenoV>.



Extract Data

Seismic Classification - 3D

Extract Train Apply

Are you selecting Seismic and PointSets Data from this survey only? Yes No

Input Data Labels

Filter

Seismic Data

- 1 Original Seismics
- 2 Steering BG Detailed
- 3 Steering BG Background
- 3a Steering PCA111 MF225
- 3b Steering FFT225 MF113
- 4 Dip steered median filter**
- 5 ChimneyCube
- 6 Wheeler-stratal-slicing
- 7a AI Cube Std
- 7b AI Cube from HorizonCube
- 8a PorosityCube from HC-NN
- 8b Gamma from HorizonCube
- 9-1 Similarity on FEF seismic
- 9-2 Dip-steered diffusion filter
- 9-3 Fault enhancement filter
- BLI_RhoB
- BLI_seismic_18deg
- BLI_seismic_28deg

Image dimensions inl:17 crl:17 z:33

Edge/Gap Policy Exclude incomplete Add data

Output Deep Learning Example Data Select ... Extract

Proceed Close ?

Input Data

Select the seismic data in the **Input Data** tab.

Image dimensions are the number of samples of the extracted cubelets.

Examples:

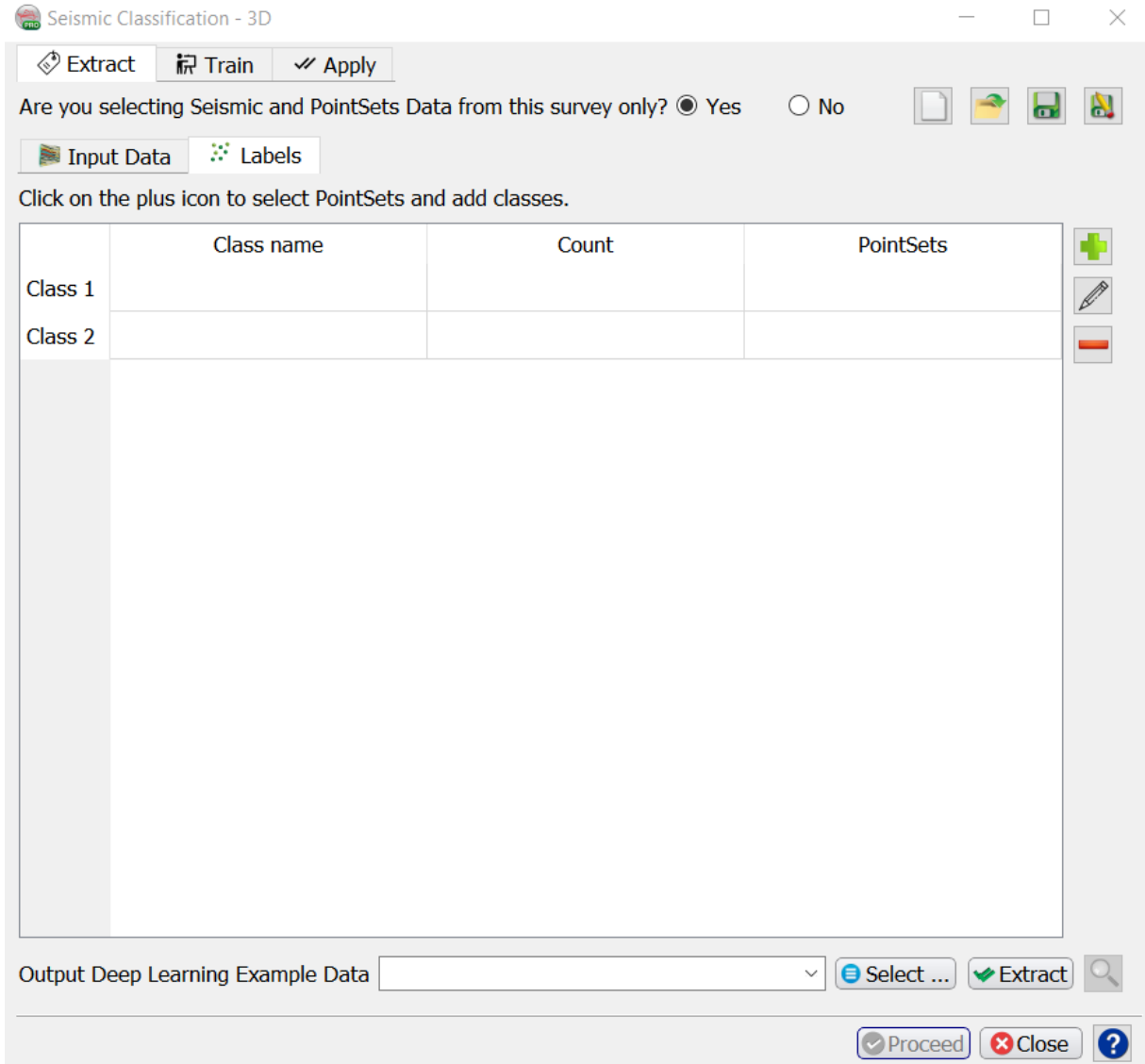
- Stepouts Inl: 8; Crl: 8, Z: 16 (number of samples on either side of the evaluation point) extracts cubelets of 17x17x33 samples.
- Stepouts Inl: 0; Crl: 16, Z: 8 extracts 2D images along inlines with dimensions 33x17 samples.

Edge/Gap policy determines how to treat incomplete examples. The default is to exclude incomplete data. **Add data** copies neighboring samples to complete the example.

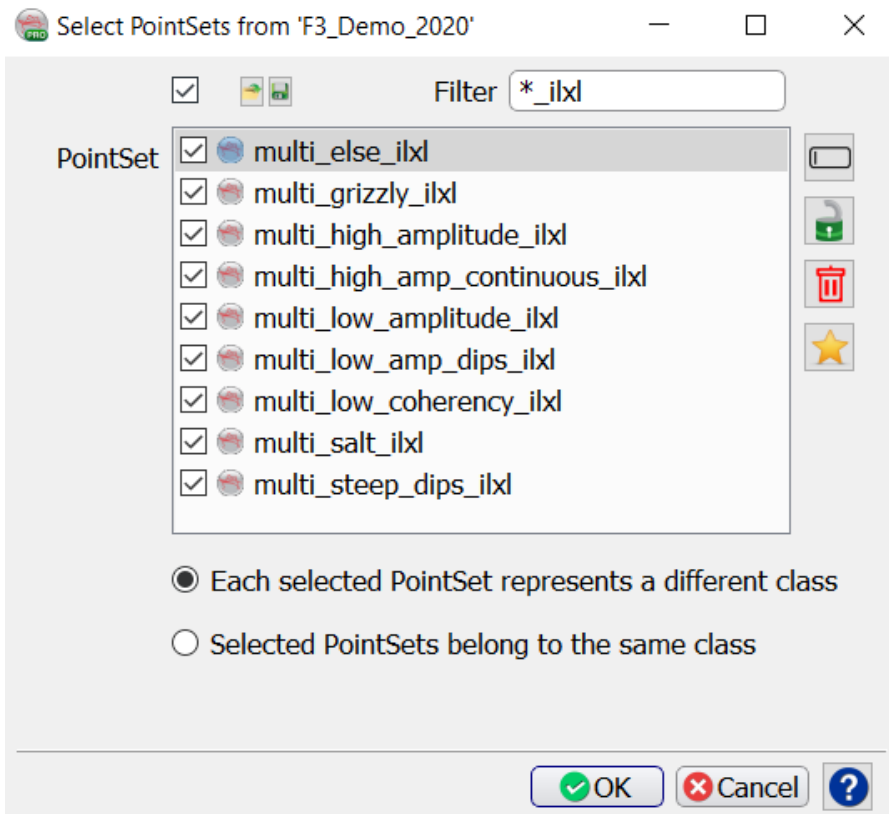
Labels

Select the PointSets and add classes in the **Labels** tab.

Press the **Select** button to start the extraction process for the input data.



Press the **+** icon. This pops up the “Select PointSets” window.



In this example select the nine PointSets 'multi..._ilxl'. The model will be trained to classify the response into the numbers 1, 2, ..., 8, 9.

When you have selected all point sets for one class press **OK** to return to the "Deep Learning Class Definition" window.

Seismic Classification - 3D

Extract Train Apply

Are you selecting Seismic and PointSets Data from this survey only? Yes No

Input Data Labels

Click on the plus icon to select PointSets and add classes.

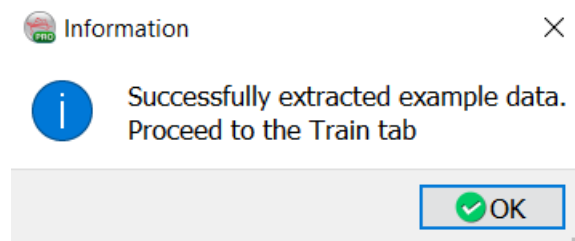
	Class name	Count	PointSets
Class 1	multi_else_ixl	20070	'multi_else_ixl'
Class 2	multi_grizzly_ixl	18257	'multi_grizzly_ixl'
Class 3	multi_high_amplitude_ixl	13283	'multi_high_amplitude_ixl'
Class 4	multi_high_amp_continuous_ixl	13366	'multi_high_amp_continuous_i...
Class 5	multi_low_amplitude_ixl	19010	'multi_low_amplitude_ixl'
Class 6	multi_low_amp_dips_ixl	17991	'multi_low_amp_dips_ixl'
Class 7	multi_low_coherency_ixl	21433	'multi_low_coherency_ixl'
Class 8	multi_salt_ixl	7499	'multi_salt_ixl'
Class 9	multi_steep_dips_ixl	10202	'multi_steep_dips_ixl'

Output Deep Learning Example Data

Repeat this process if you want to add more examples from other point sets. These point sets can be located in different surveys. For example, if you have created Chimney Cubes in the past and you saved the Chimney-Yes and Chimney-No point sets, you can re-use these to train a deep learning model on all these examples.

Repeat the process by pressing + until all examples for classes are defined.

Specify the name of the **Output Deep Learning Example Data** and press **Extract** to start the extraction process.



When this process is finished you are back in the “Seismic Classification” start window. The **Proceed** button has turned green. Press it to continue to the Training tab.

Seismic Classification - 3D

Extract Train Apply

Are you selecting Seismic and PointSets Data from this survey only? Yes No

Input Data Labels

Click on the plus icon to select PointSets and add classes.

	Class name	Count	PointSets
Class 1	multi_else_ixl	20070	'multi_else_ixl'
Class 2	multi_grizzly_ixl	18257	'multi_grizzly_ixl'
Class 3	multi_high_amplitude_ixl	13283	'multi_high_amplitude_ixl'
Class 4	multi_high_amp_continuous_ixl	13366	'multi_high_amp_continuous_i...
Class 5	multi_low_amplitude_ixl	19010	'multi_low_amplitude_ixl'
Class 6	multi_low_amp_dips_ixl	17991	'multi_low_amp_dips_ixl'
Class 7	multi_low_coherency_ixl	21433	'multi_low_coherency_ixl'
Class 8	multi_salt_ixl	7499	'multi_salt_ixl'
Class 9	multi_steep_dips_ixl	10202	'multi_steep_dips_ixl'

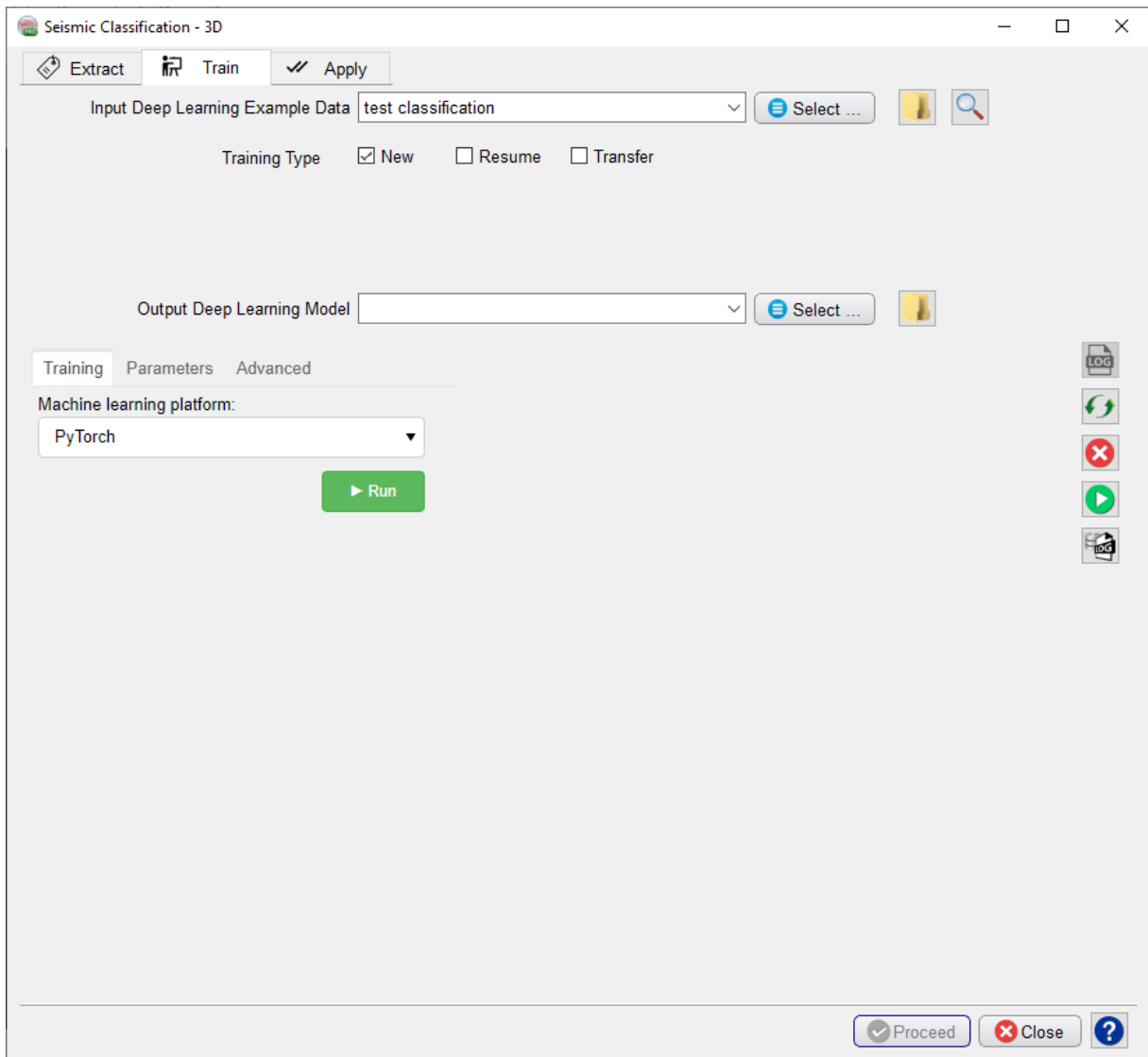
Output Deep Learning Example Data multi_Class_DL_Ex Select ... Extract

Proceed Close ?

Both the Class Examples and the Class Definition selections can be **Saved**, retrieved (**Open**) and modified (**Edit**) using the corresponding icons below the **+** icon. **Remove** deletes the respective selection file from the database.

Training

When you already have stored data, you can start in the **Training tab** with Select training data (Input Deep Learning Example data).



Under the Select training data there are three toggles controlling the **Training Type**:

- **New** starts from a randomized initial state.
- **Resume** starts from a saved (partly trained) network. This is used to continue training if the network has not fully converged yet.
- **Transfer** starts from a trained network that is offered new training data. Weights attached to Convolutional Layers are not updated in transfer training. Only weights attached to the last layer (typically a Dense layer) are updated.

After the training data is selected the UI shows which models are available. These models are divided over three platforms: **Scikit Learn**, **PyTorch** and **Keras (TensorFlow)**.

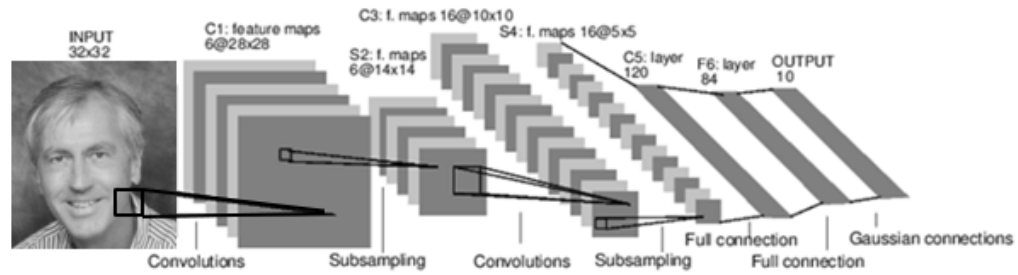
Check the **Parameters** tab to see which models are supported and which parameters can be changed. From **Scikit Learn** we currently support:

1. Logistic
 - a. Logistic Regression Classifier
2. Ensemble Methods
 - a. Random Forests
 - b. Gradient Boosting
 - c. Adaboost
 - d. XGBoost
3. Neural Networks
 - a. Multi-Layer-Perceptrons
4. Support Vector Machines
 - a. Linear
 - b. Polynomial
 - c. Radial Basis Function
 - d. Sigmoid

For parameter details we refer to: [the Scikit Learn User Guide](#).

From **Keras (TensorFlow)** we currently support the classic LeNet Convolutional Neural Network for log-log prediction.

The dGB LeNet regressor is a fairly standard Convolutional Neural Network that is based on the well-known LeNet architecture (below).



A Full Convolutional Neural Network (LeNet)

Seismic Classification - 3D

Extract Train Apply

Input Deep Learning Example Data multi_Class_DL_Ex Select ...

Training Type New Resume Transfer

Output Deep Learning Model DL_Model_al Select ...

Training Parameters Advanced

Type
dGB LeNet classifier

Validation Percentage Split: 0.20

Batch Size
32

Epochs: 15

Early Stopping: 10

Initial Learning Rate (1e): -4

Epoch drop (%): 33.33

Size: 5.37 GB

Decimate input

Train on GPU

Reset Back

LOG

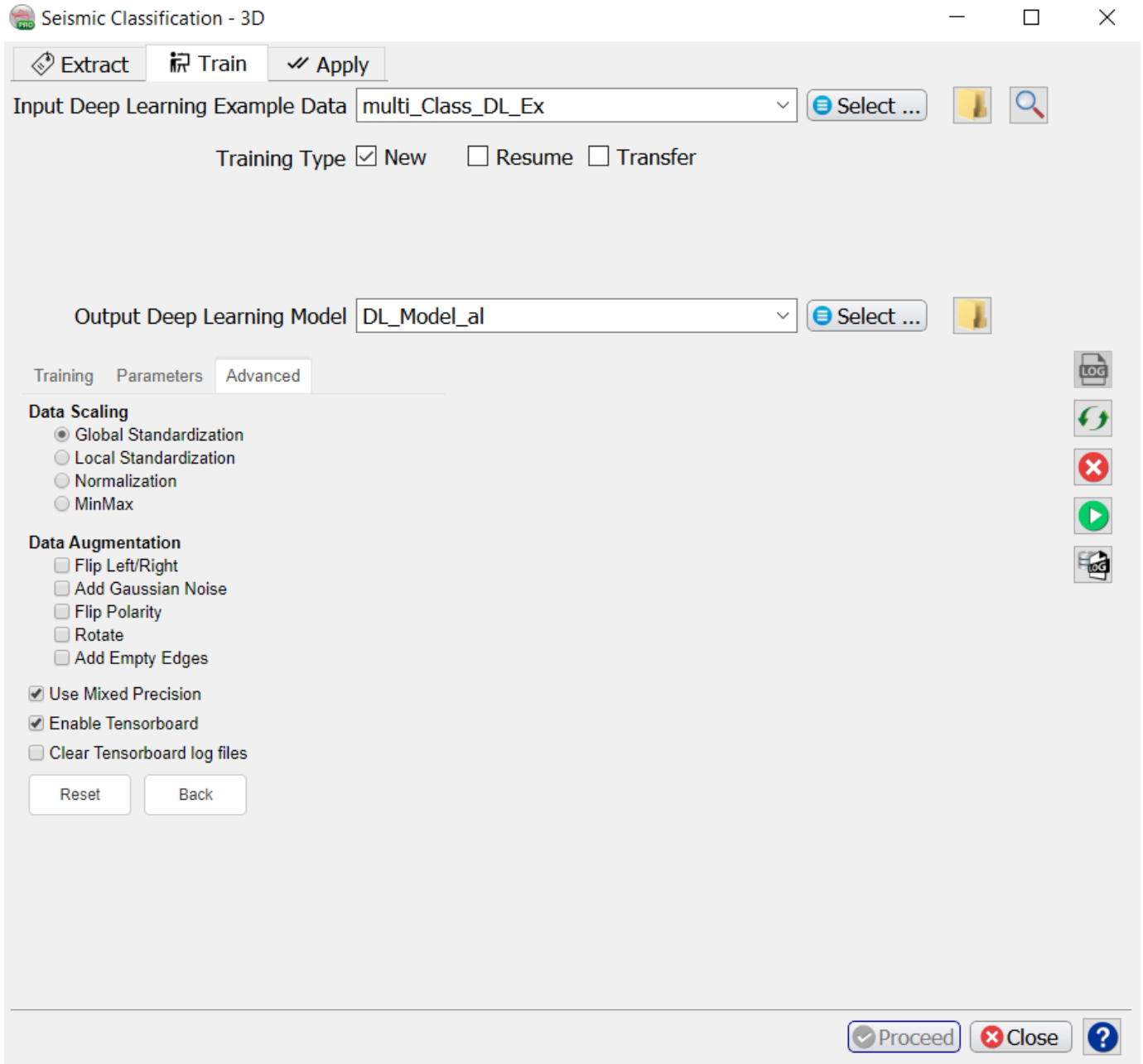
Refresh

Cancel

Play

Help

Proceed Close ?



The following training parameters can be set in the **Parameters** tab:

Batch Size: this is the number of examples that are passed through the network after which the model weights are updated. This value should be set as high as possible to increase the representativeness of the samples on which the gradient is computed, but

low enough to have all these samples fit within the memory of the training device (much smaller for the GPU than the CPU). If we run out of memory (raises a Python OutOfMemory exception), lower the batch size!

Note that if the model upscales the samples by a factor 1000 for instance on any layer of the model, the memory requirements will be upscaled too. Hence a typical 3D Unet model of size 128-128-128 will consume up to 8GB of (CPU or GPU) RAM.

Epochs: this is the number of update cycles through the entire training set. The number of epochs to use depends on the complexity of the problem. Relatively simple CNN networks may converge in 3 epochs. More complex networks may need 30 epochs, or even hundreds of epochs. Note, that training can be done in steps. Saved networks can be trained further when you toggle **Resume**.

Early Stopping: this parameter controls early stopping when the model does not change anymore. Increase the value to avoid early stopping.

Initial Learning rate: this parameter controls how fast the weights are updated. Too low means the network may not train; too high means the network may overshoot and not find the global minimum.

Epoch drop: controls how the learning rate decays over time.

Decimate Input: This parameter is useful when we run into memory problems. If we decimate the input the program will divide the training examples in chunks using random selection. The training is then run over chunks per epoch meaning the model will eventually have seen all samples once within one epoch, but only no more than one chunk of samples will be loaded in RAM while the training is performed.

Mixed precision: Mixed precision option has been added to dgbpy for both PyTorch and Keras platform. This allows to train models using a combination of 16-bit and 32-bit

precision formats which will optimize GPU memory usage and efficiency. The GPU memory usage can be reduced by 50% and 75% in some cases. It also increases training speed 2 - 3 times on GPUs with Volta or Turing architectures. This doesn't also affect training accuracy or performance.

For more details about what is 'Mixed precision training' please refer to:

- [Keras documentation: Mixed precision](#)
- [NVIDIA documentation: Train with Mixed Precision](#)

After model selection, return to the **Training tab**, specify the **Output Deep Learning model** and press the green **Run** button. This starts the model training. The **Run** button is replaced in the UI by **Pause** and **Abort** buttons.

The progress of Keras / TensorFlow runs can be monitored in **TensorBoard**, which automatically starts up in your default browser. Please note that it may take a few minutes before TensorBoard has gathered the information that is gathered during training. As all TensorBoard information on your machine is retrieved for all current and historic runs it is recommended to clear the TensorBoard log files from time to time. You can do this with the **Clear TensorBoard log files** toggle in the **Advanced** parameters before any Run.

TensorBoard SCALARS GRAPHS

Show data download links
 Ignore outliers in chart scaling
 Tooltip sorting method: default

Smoothing 0,6

Horizontal Axis
 RELATIVE WALL

Runs
 Write a regex to filter runs

- F3_Demo_2019_run1_2020-08-30T13m02 \train
- F3_Demo_2019_run1_2020-08-30T13m02 \validation

TOGGLE ALL RUNS

D:\Surveys\MachineLearning\malenov_input_8x8x16_Aug_2020

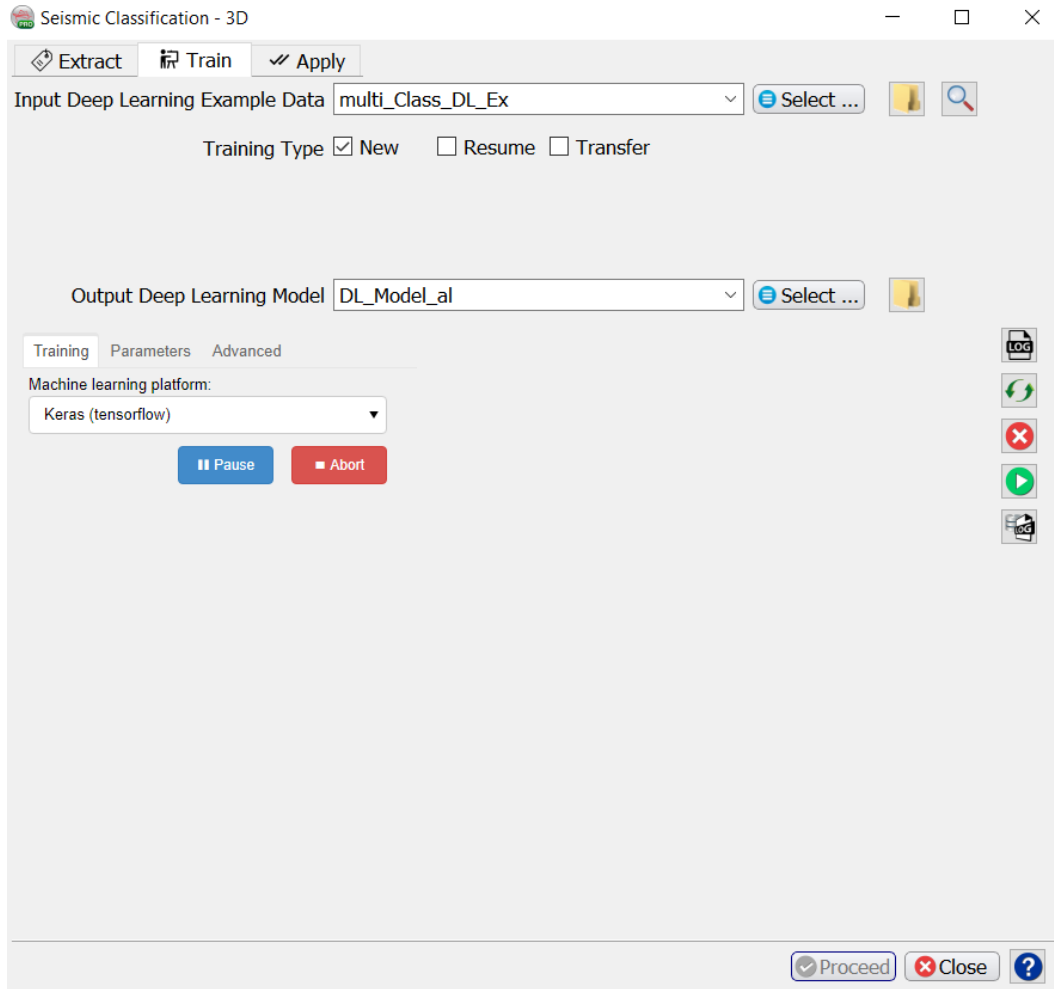
Filter tags (regular expressions supported)

epoch_accuracy

epoch_loss

epoch_lr

The progress can be also followed in a text log file. If this log file does not start automatically, please press the **log file icon** in the toolbar on the right-hand side of the window:



Below the log file icon there is a **Reset** button to reload the window. Below this there are three additional icons that control the Bokeh server, which controls the communication with the Python side of the Machine Learning plugin. The server should start automatically. In case of problems, it can be controlled manually via the **Start** and **Stop** icons. The current status of the Bokeh server can be checked by viewing the Bokeh server log file.

When the processing log file shows “Finished batch processing”, and the Training window shows “Training Successful”, you can move to the **Apply** tab.

File

```
loss: nan
accuracy: 0.14243550598621368
val_loss: nan
val_accuracy: 0.14244557917118073
Time: 01:13
Saving model.
Model saved.

Process: 'Machine Learning Training'
Finished: Tue 16 May 2023, 00:18:16
Finished batch processing.
```

Close Reload

Seismic Classification - 3D

Extract Train Apply

Input Deep Learning Example Data multi_Class_DL_Ex Select ...

Training Type New Resume Transfer

Output Deep Learning Model DL_Model_al Select ...

Training Parameters Advanced

Machine learning platform:
Keras (tensorflow)

Run

Training Successful

LOG

LOG

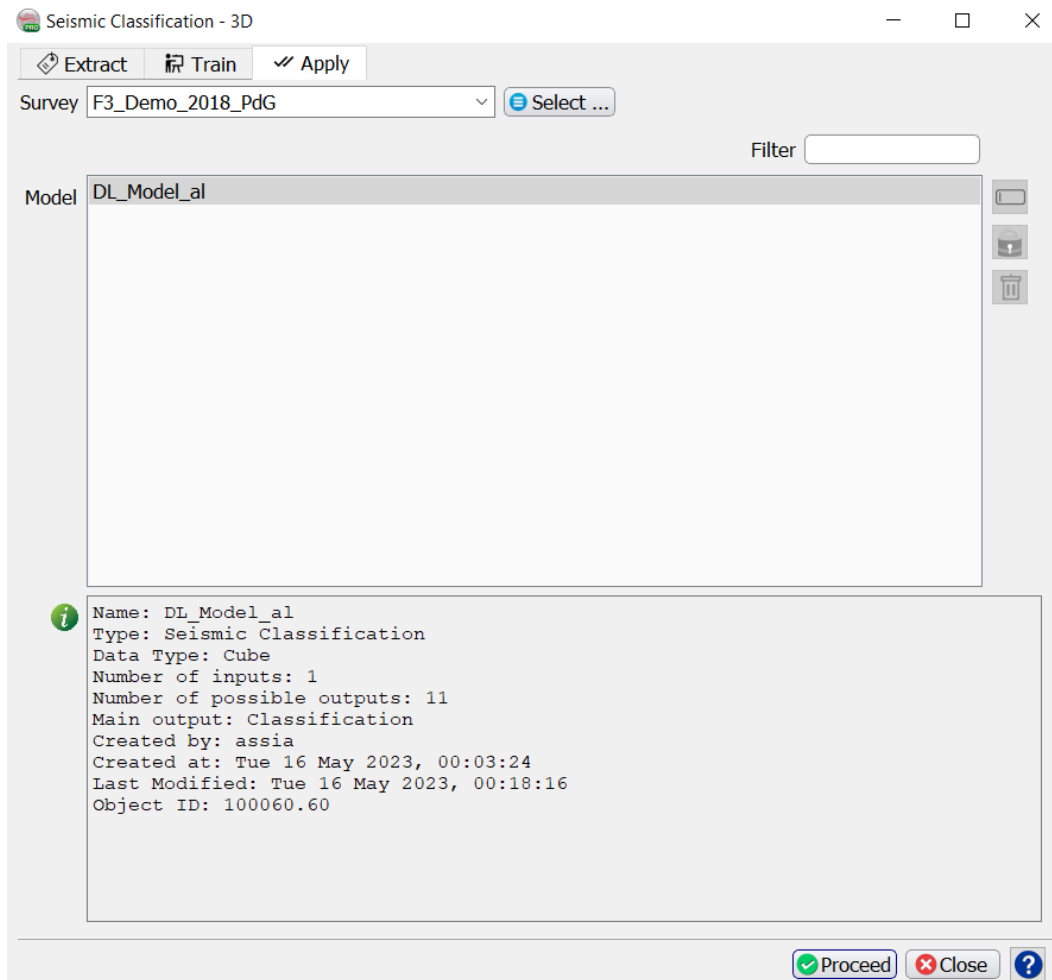
LOG

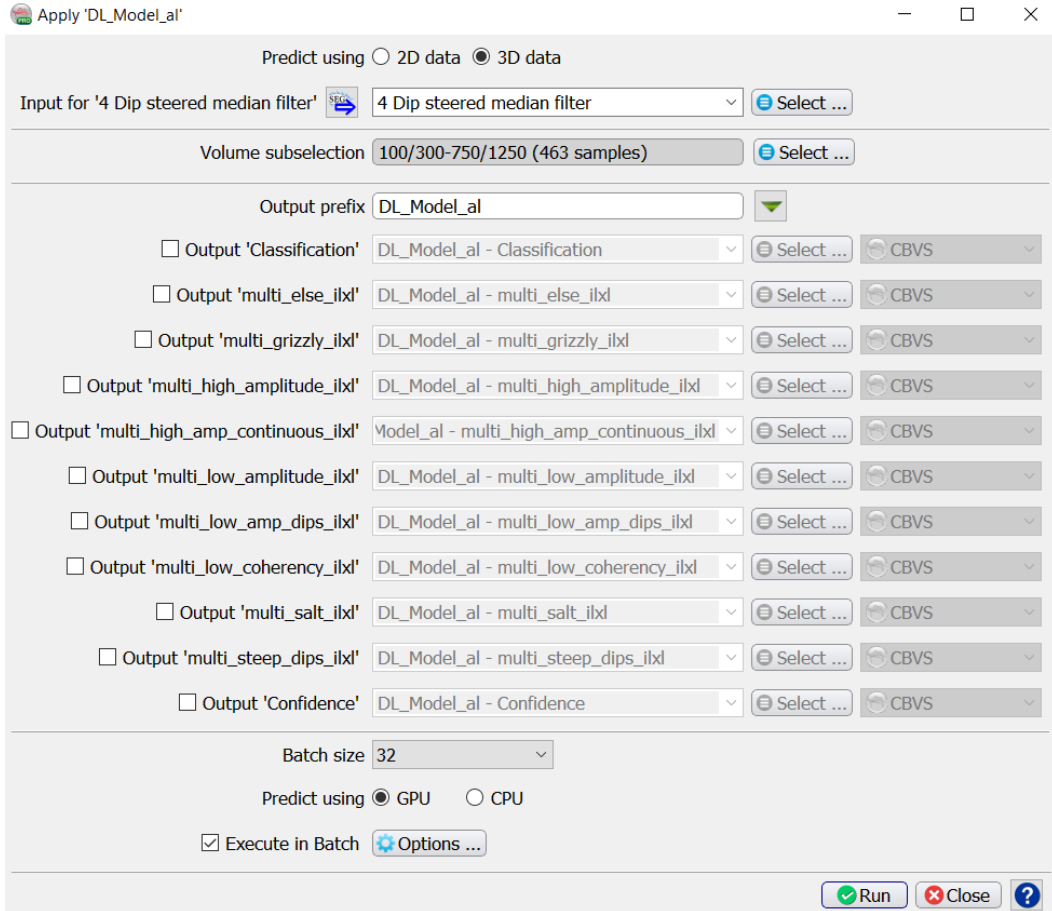
LOG

Proceed Close ?

Apply

Once the Training is done, the trained model can be applied. Select the trained model and press **Proceed**.



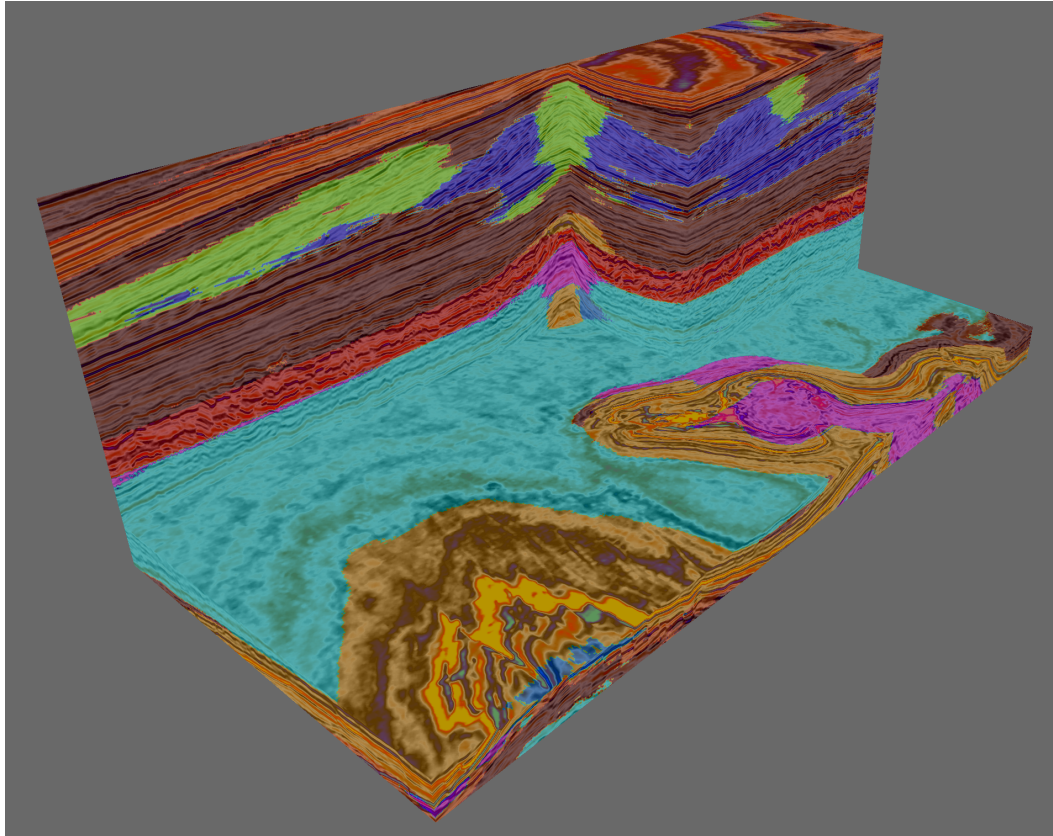


The Apply window pops up. Here you optionally apply to a **Volume subselection**. You can output a **Classification** Volume (values 1 to 9 in this case), a **Confidence** volume and **Probability** volumes for each of the output classes. Each output is stored in a separate cube in **CBVS** format, or **SEGY** format. The names of all outputs are given the same **Output prefix**. To change the prefix for all selected output cubes, press the **funnel icon** next to the prefix name.

Probability is simply the output of the prediction node which ranges between 0 and 1.

Confidence is computed as the value of the winning node minus the value of the second best node.

The **Batch size** is only important for the speed. If the batch size is set too high you may run out of memory (GPU or CPU depending on the **Predict using GPU** toggle). Running the application on a GPU is many times faster than running it on a CPU. Press Run to create the desired outputs. The image below is taken from a seismic facies Classification cube generated in this way.



Output of the trained model: seismic facies cube.

3.4.3 Seismic Image Segmentation (Supervised 2D and 3D)

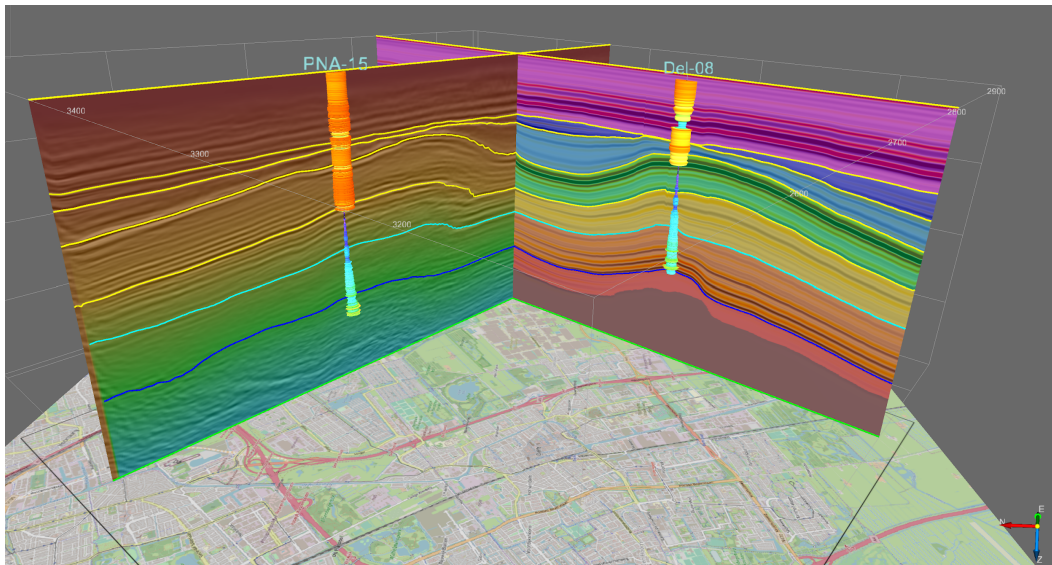
A Segmentation model maps a seismic image (1D, 2D, or 3D) to a labelled image (mask) of similar size where the output has discrete values (0 or 1, or 1, 2, ..., N).

These workflows are categorized as supervised methods in that the input data set(s) and the target data are provided to facilitate training. For example, if you have a 3D seismic volume and a target 3D volume with values 1 at fault planes and 0 elsewhere, you can extract (2D images or 3D cubelets) from both volumes to create a model for fault prediction. Application of the trained model delivers a segmentation volume (3D seismic) or a line set (2D seismic). We will discuss the workflow and user interface components on the basis of a 3D example.

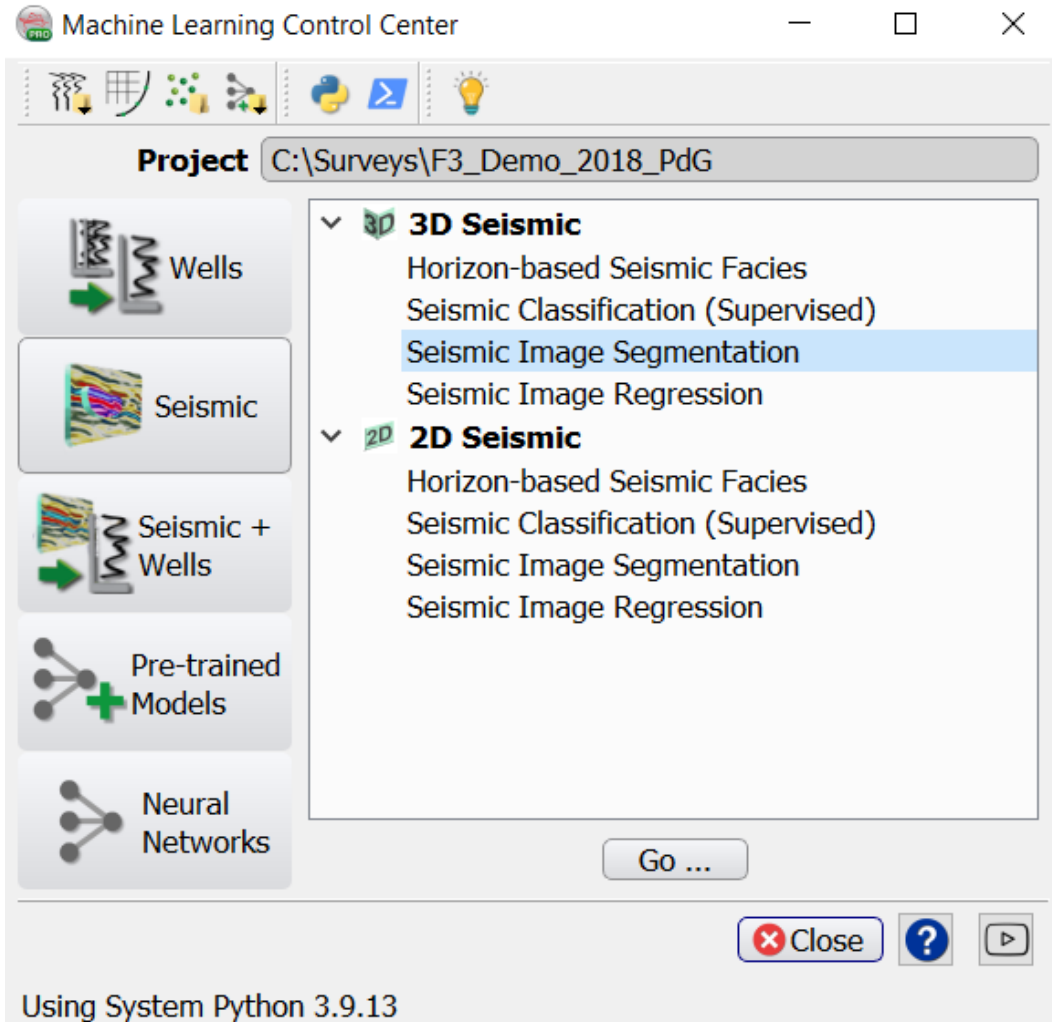
The objective is to predict geologically meaningful layers from seismic input. We will train a Unet on 2 input features:

1. 2D synthetic seismic images (128 x 128 samples)
2. 2D images of Two-Way-Time (TWT also of 128 x 128 samples)

The desired output are 2D images (128 x 128 samples) containing labels (1 to 8). The reason for using TWT in addition to the seismic input feature is to help the model find the logical order in the segments (layers are stacked: 1 is on top; 8 is the lowest). The trained Unet is applied to real 3D data. The output is a 3D segmented volume. If successful, this workflow can be applied to quickly segment unseen seismic volumes with similar geology. The Figure below shows the synthetic data.

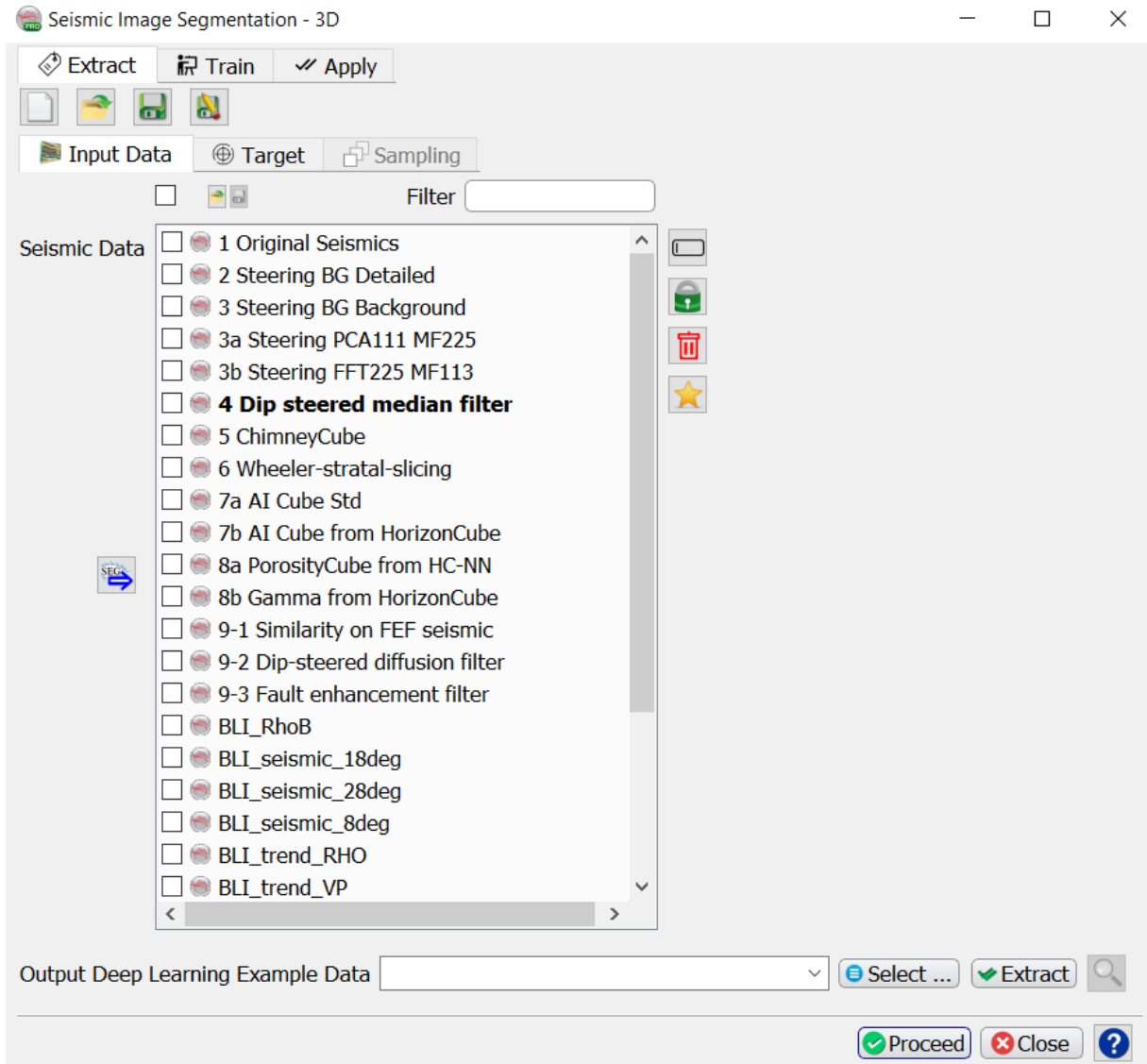


Synthetic data over the city of Delft. The synthetic data set was created in OpendTect from a set of interpreted 3D horizons and 2 wells with Acoustic Impedance logs. The logs were interpolated in the Volume Builder. The synthetic seismic volume, interval labels volume (mask) and Two-Way-Time volume were created in the attribute engine. Seismic inline through DEL-08 shows synthetic seismic co-rendered with target labels. Seismic crossline through PNA-15 shows synthetic seismic co-rendered with Two-way time.

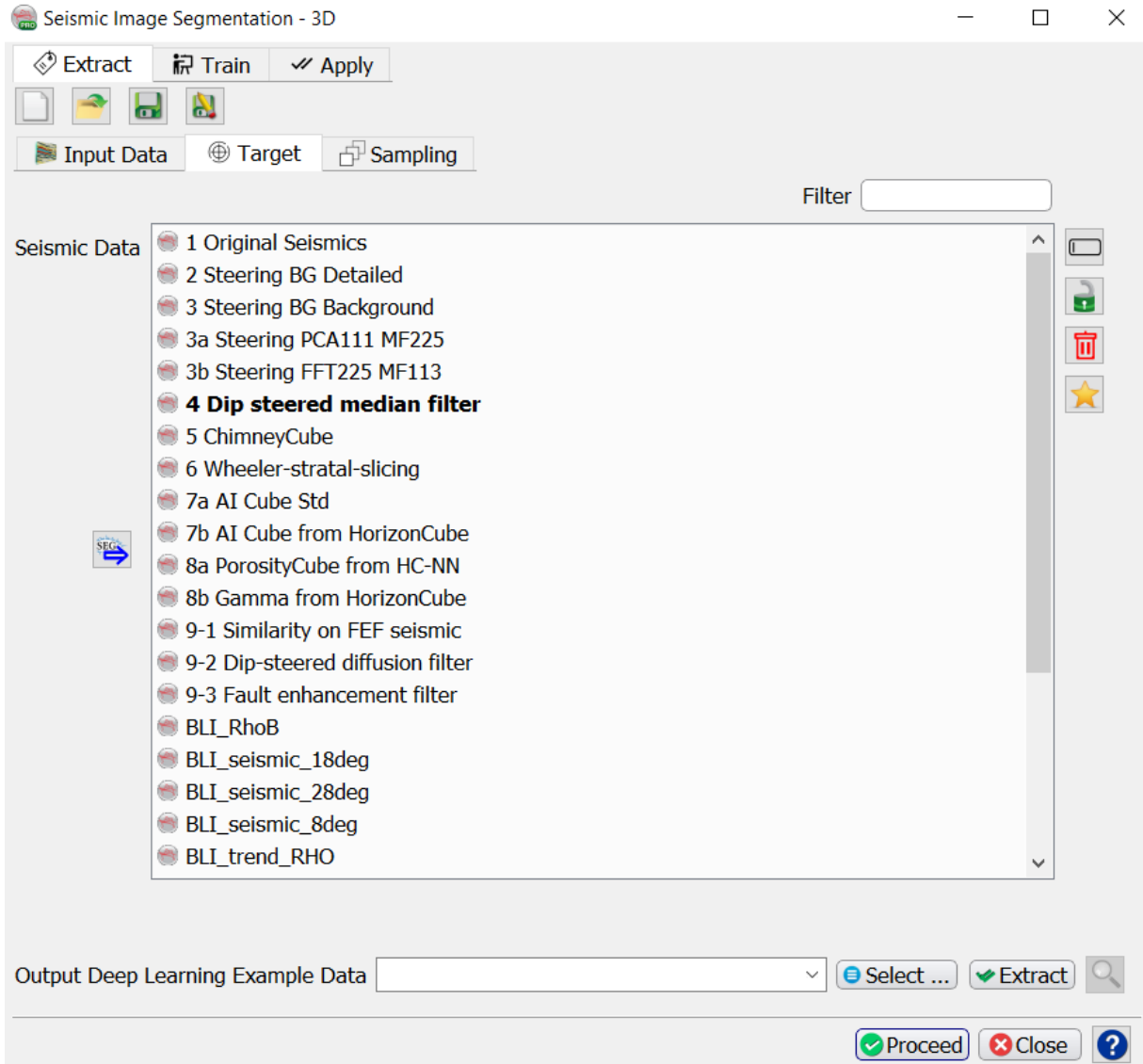


Select the Seismic Image Segmentation workflow and press **Go**.

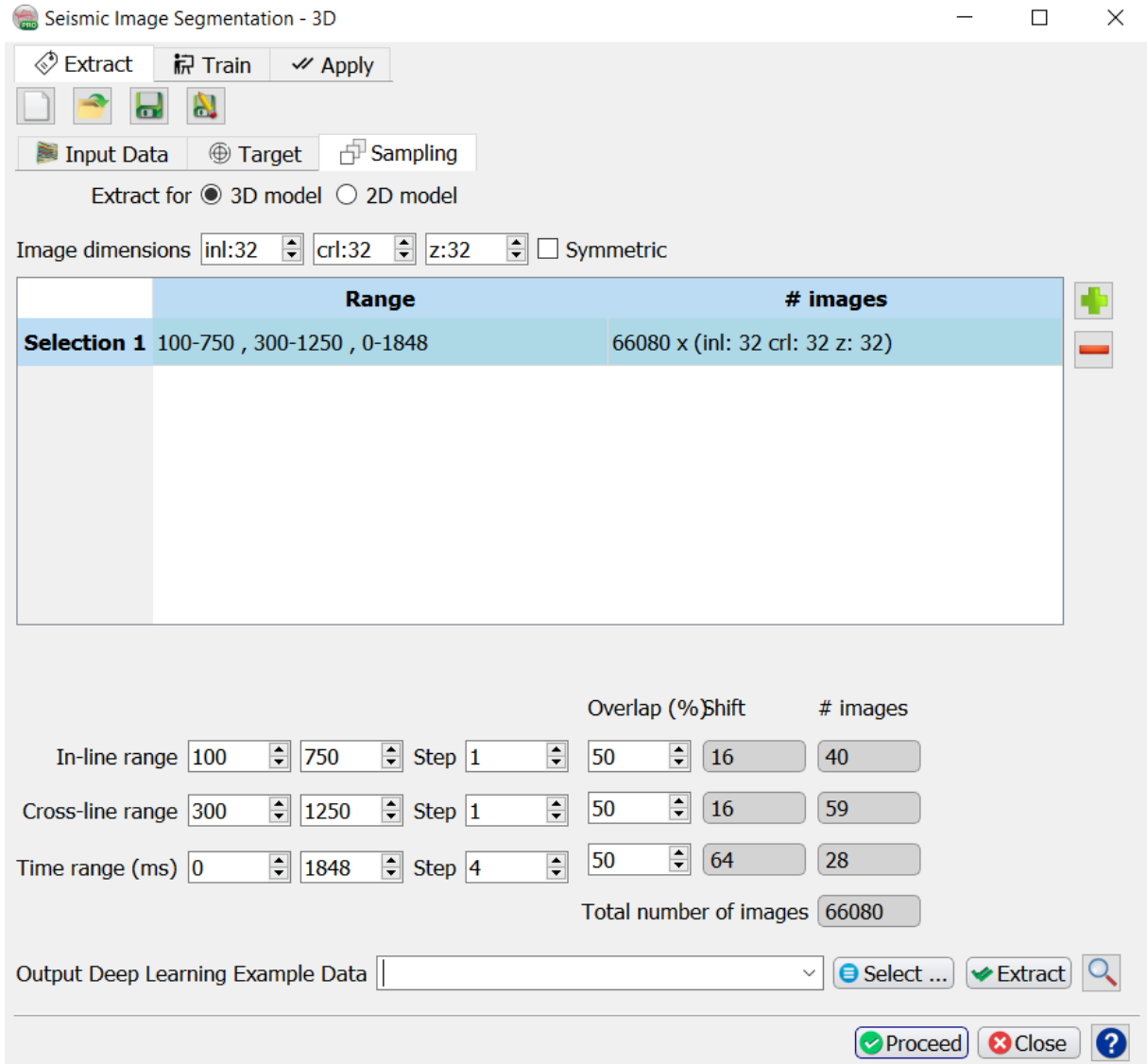
Extract Data



Input Data tab, select the input seismic data for prediction.



Target tab, select the target seismic volume containing the labels. In the example this volume is called “classification”.



Sampling tab, select the image dimensions of the input features.

Image Dimensions are the number of samples in inline, crossline and Z directions.

Examples:

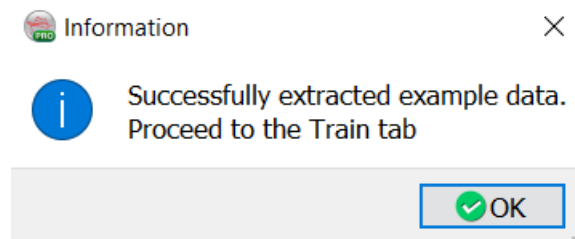
- Inl: 0; Crl: 128, Z: 128 - extracts 2D images along inlines with dimensions 128 x 128 samples
- Inl: 64, Crl: 64, Z: 64 - extracts cubelets of 64x64x64 samples.

The examples (images) are extracted from the specified **Inline**, **Crossline** and **Time Ranges**. **Step** is the sampling rate in the bin (time) directions. The **Overlap** determines how much overlap there is between neighboring images. The total number of images is given on the right-hand side of the window. You can control this number by playing with the ranges and overlap parameters.

Different seismic selections can be used by pressing the **+** icon.

The Seismic Image Segmentation Parameters can be **Saved**, retrieved (**Open**) and **Deleted** using the corresponding icons.

Specify the name of the **Output Deep Learning Example Data** and press **Extract** to start the extraction process.



When this process is finished the **Proceed** button has turned green. Press it to continue to the Training tab.

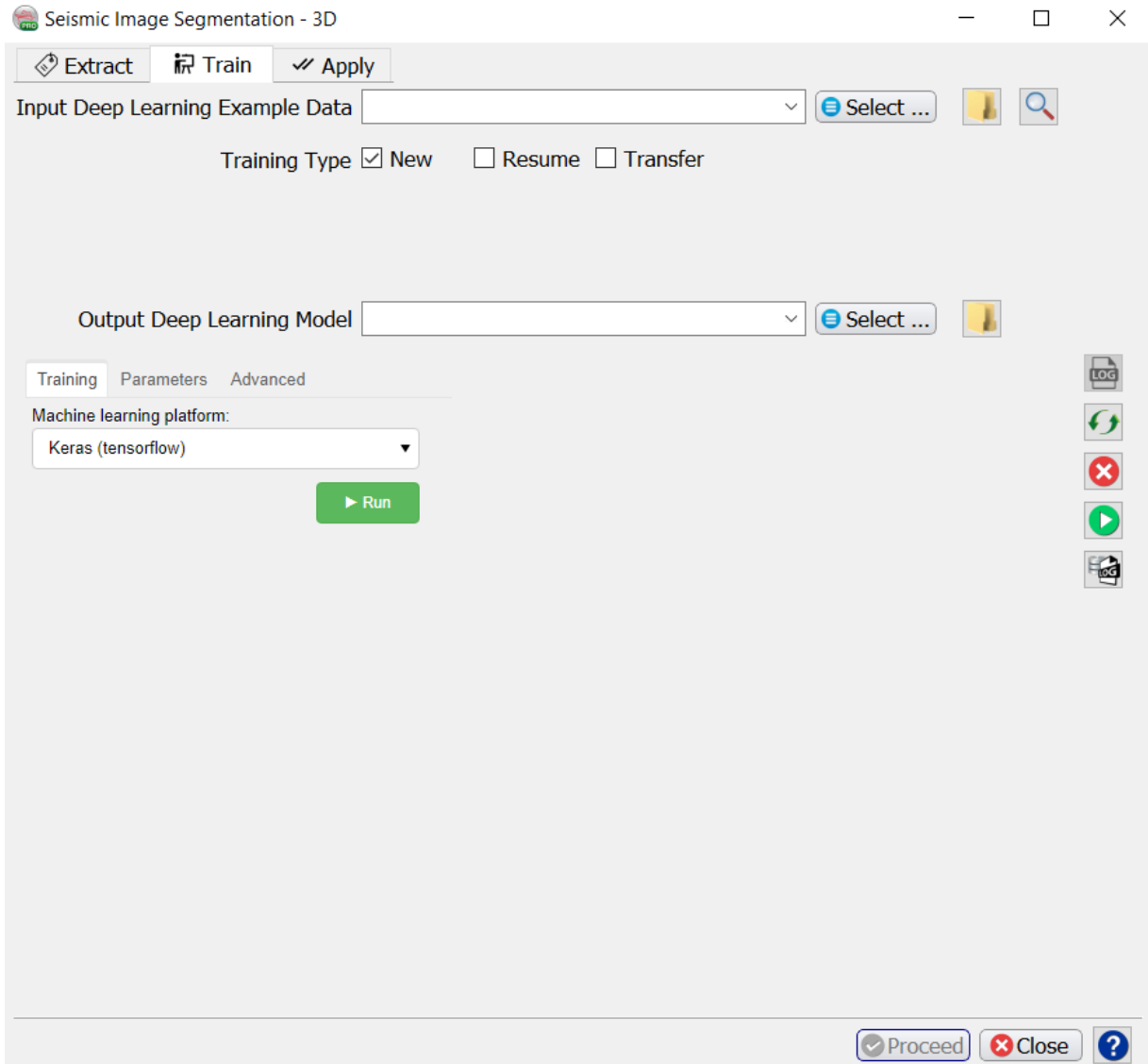
Training

When you already have stored data, you can start in the **Training tab** with **Select** training data (Input Deep Learning Example data).

Under the Select training data there are three toggles controlling the **Training Type**:

- **New** starts from a randomized initial state.
- **Resume** starts from a saved (partly trained) network. This is used to continue training if the network has not fully converged yet.
- **Transfer** starts from a trained network that is offered new training data. Weights attached to Convolutional Layers are not updated in transfer training. Only weights attached to the last layer (typically a Dense layer) are updated.

After the training data is selected the UI shows which models are available. For seismic image workflows we use **Keras (TensorFlow)**.



Check the **Parameters** tab to see which models are supported and which parameters can be changed.

Seismic Image Segmentation - 3D

Extract Train Apply

Input Deep Learning Example Data Select ...

Training Type New Resume Transfer

Output Deep Learning Model Select ...

Training Parameters Advanced

Type
dGB UNet Segmentation

Validation Percentage Split: 0.20

Batch Size
4

Epochs: 15

Early Stopping: 10

Initial Learning Rate (1e): -4

Epoch drop (%): 33.33

Size: 25.98 GB

Decimate input

Train on GPU

Reset Back

LOG

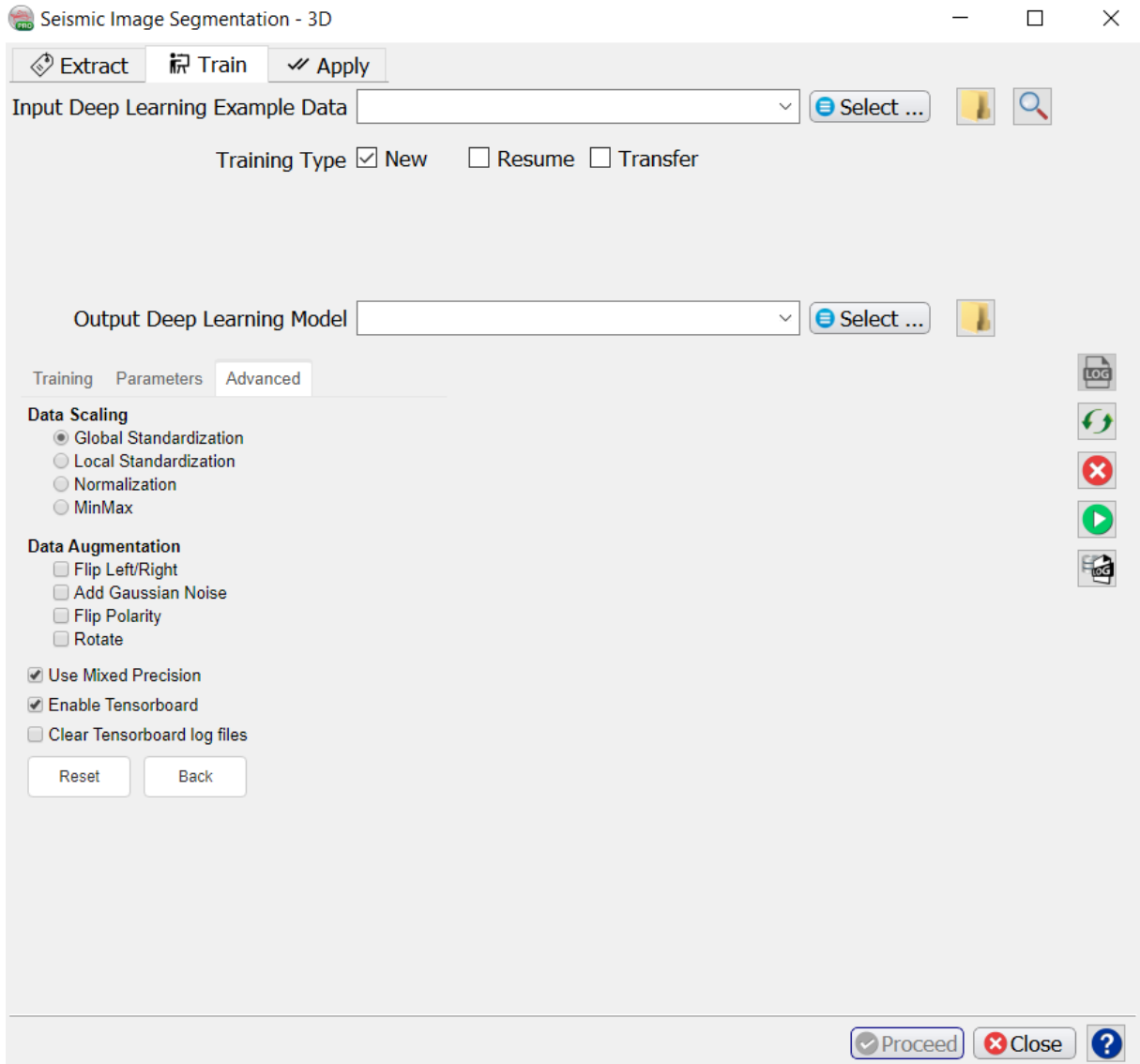
Refresh

Close

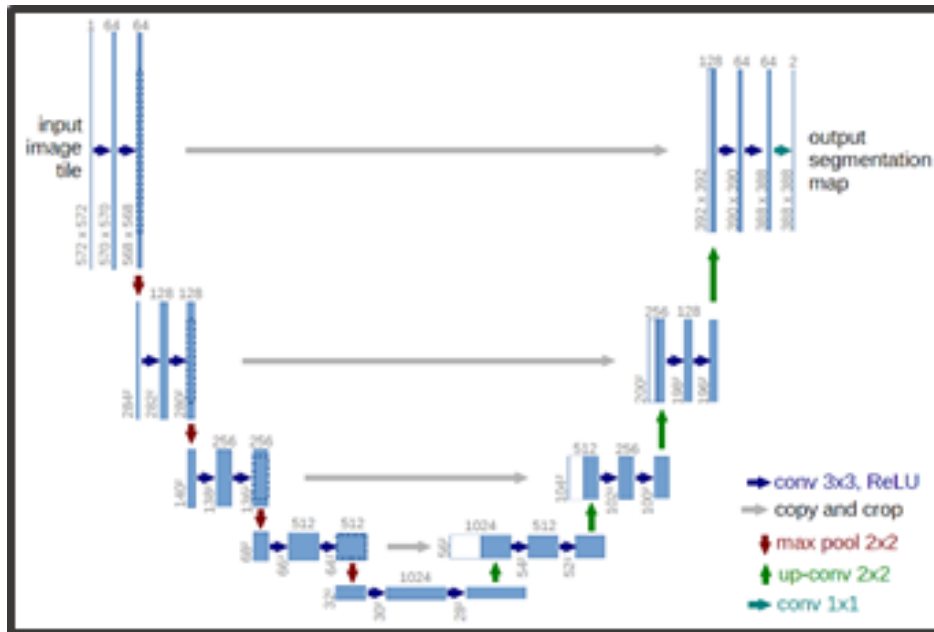
Play

Help

Proceed Close ?



The dGB Unet Segmentation is an auto-encoder - decoder type of Convolutional Neural Network. The architecture is shown schematically below.



The following training parameters can be set in the **Parameters** tab:

Batch Size: this is the number of examples that are passed through the network after which the model weights are updated. This value should be set as high as possible to increase the representativeness of the samples on which the gradient is computed, but low enough to have all these samples fit within the memory of the training device (much smaller for the GPU than the CPU). If we run out of memory (raises a Python OutOfMemory exception), lower the batch size!

Note that if the model upscales the samples by a factor 1000 for instance on any layer of the model, the memory requirements will be upscaled too. Hence a typical 3D Unet model of size 128-128-128 will consume up to 8GB of (CPU or GPU) RAM.

Epochs: this is the number of update cycles through the entire training set. The number of epochs to use depends on the complexity of the problem. Relatively simple CNN networks may converge in 3 epochs. More complex networks may need 30 epochs, or

even hundreds of epochs. Note, that training can be done in steps. Saved networks can be trained further when you toggle **Resume**.

Early Stopping: this parameter controls early stopping when the model does not change anymore. Increase the value to avoid early stopping.

Initial Learning rate: this parameter controls how fast the weights are updated. Too low means the network may not train; too high means the network may overshoot and not find the global minimum.

Epoch drop: controls how the learning rate decays over time.

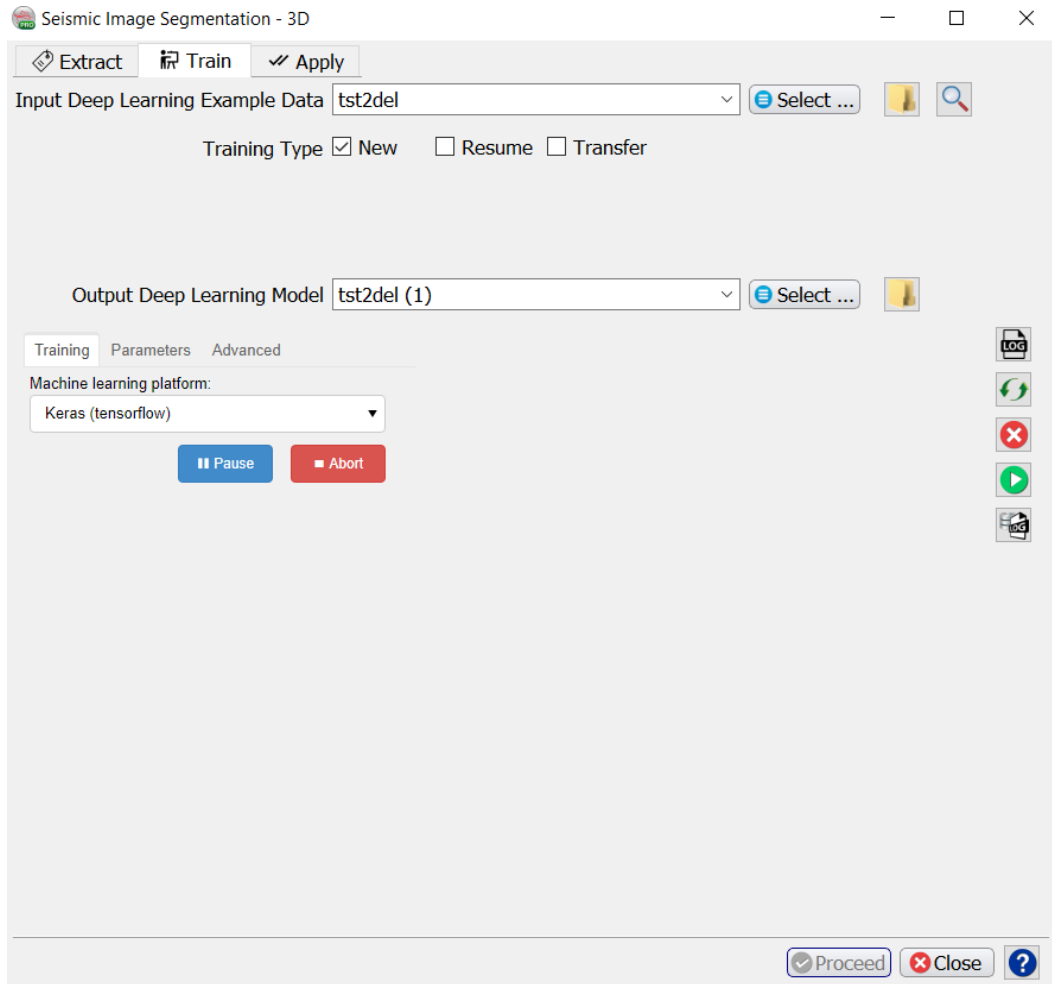
Decimate Input: this parameter is useful when we run into memory problems. If we decimate the input the program will divide the training examples in chunks using random selection. The training is then run over chunks per epoch meaning the model will eventually have seen all samples once within one epoch, but only no more than one chunk of samples will be loaded in RAM while the training is performed.

Mixed precision: Mixed precision option has been added to dgbpy for both PyTorch and Keras platform. This allows to train models using a combination of 16-bit and 32-bit precision formats which will optimize GPU memory usage and efficiency. The GPU memory usage can be reduced by 50% and 75% in some cases. It also increases training speed 2 - 3 times on GPUs with Volta or Turing architectures. This doesn't also affect training accuracy or performance.

For more details about what is 'Mixed precision training' please refer to:

- [Keras documentation: Mixed precision](#)
- [NVIDIA documentation: Train with Mixed Precision](#)

After model selection, specify the **Output Deep Learning model** and press the green **Run** button. This starts the model training. The Run button is replaced in the UI by **Pause** and **Abort** buttons.



The progress of Keras / TensorFlow runs can be monitored in **TensorBoard**, which automatically starts up in your default browser. Please note that it may take a few minutes before TensorBoard has gathered the information that is gathered during training. As all TensorBoard information on your machine is retrieved for all current and historic runs it is recommended to clear the TensorBoard log files from time to time. You

can do this with the **Clear TensorBoard log files** toggle before any Run in the **Advanced** tab.

The screenshot shows the TensorBoard interface with the **SCALARS** tab selected. The interface includes a search bar for filtering tags, a list of runs with checkboxes for selection, and two line charts showing training metrics over 14 epochs.

TensorBoard SCALARS GRAPHS

Filter tags (regular expressions supported)

Show data download links
 Ignore outliers in chart scaling

Tooltip sorting method: **default**

Smoothing: 0,6

Horizontal Axis: **STEP** RELATIVE WALL

Runs

Write a regex to filter runs

- Delft_ML_test_run1_2020-08-26T13m34\train
- Delft_ML_test_run1_2020-08-26T13m34\validation
- Delft_ML_test_run2_2020-08-26T13m55\train
- Delft_ML_test_run3_2020-08-26T13m58\train
- Delft_ML_test_run3_2020-08-26T13m58\validation
- Delft_ML_test_run4_2020-08-27T12m43\train
- Delft_ML_test_run4_2020-08-27T12m43\validation

TOGGLE ALL RUNS

D:\Surveys\MachineLearning\Seis+Z_to_classes_0x128x128

epoch_accuracy

epoch_loss

epoch_loss

Epoch	Loss
0	0.35
1	0.25
2	0.18
3	0.15
4	0.12
5	0.10
6	0.08
7	0.07
8	0.06
9	0.05
10	0.04
11	0.035
12	0.03
13	0.025
14	0.02

epoch_lr

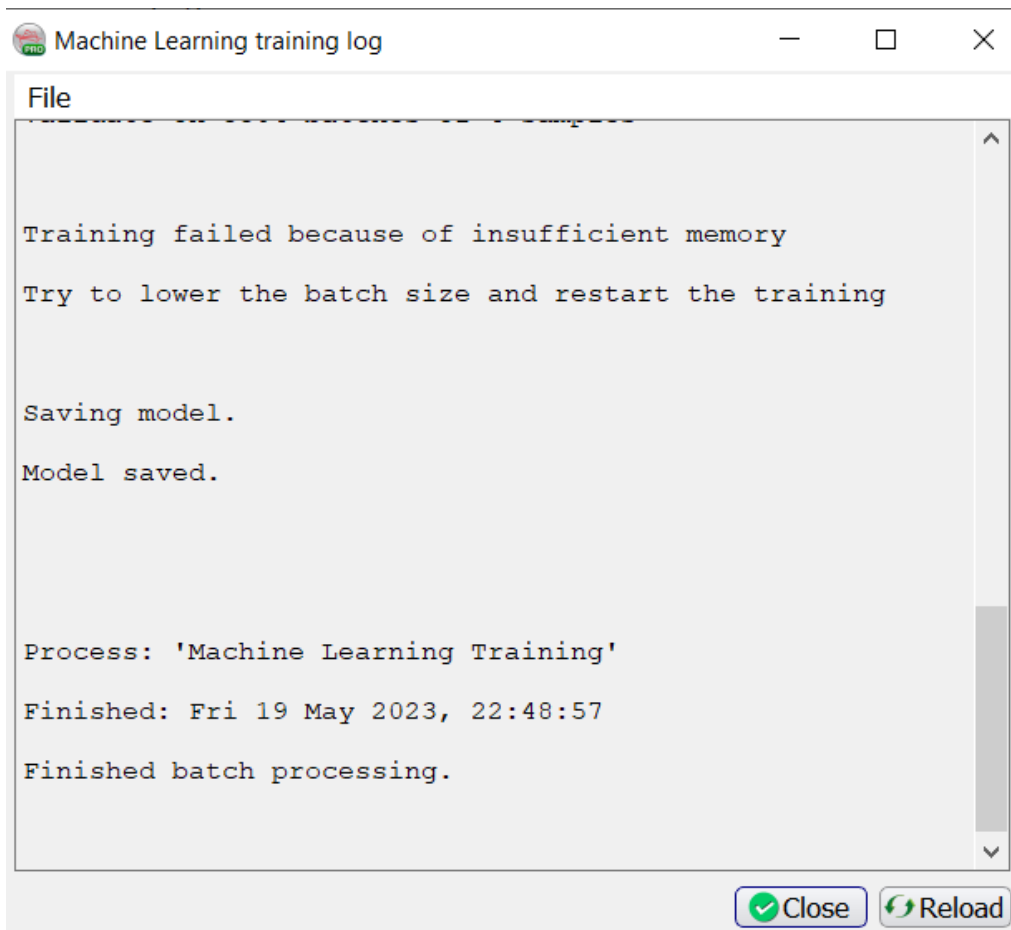
epoch_mae

epoch_mae

Epoch	MAE
0	0.35
1	0.25
2	0.18
3	0.15
4	0.12
5	0.10
6	0.08
7	0.07
8	0.06
9	0.05
10	0.04
11	0.035
12	0.03
13	0.025
14	0.02

The progress can be also followed in a text log file. If this log file does not start automatically, please press the **log file icon** in the toolbar on the right-hand side of the window.

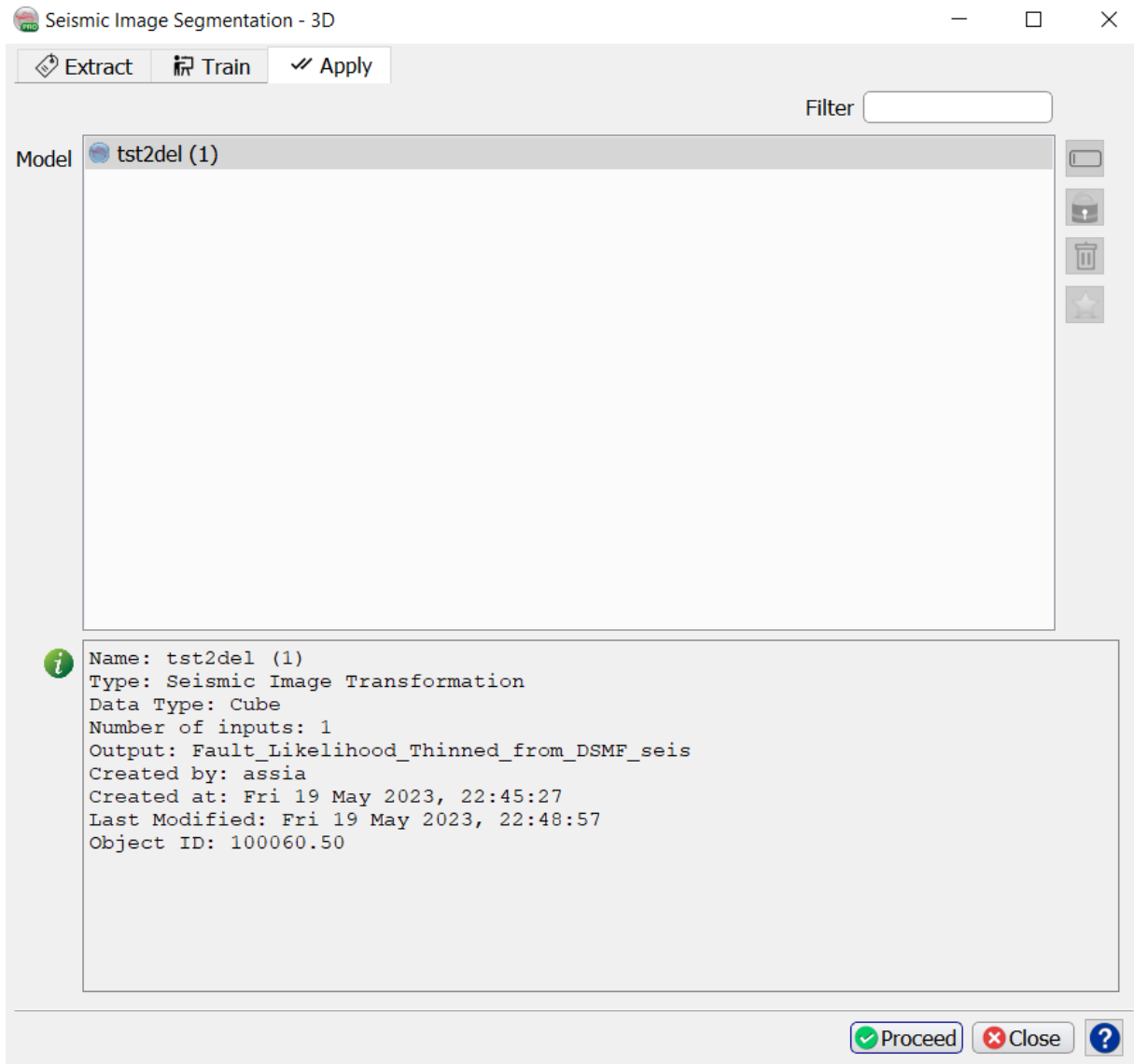
Below the log file icon there is a **Reset** button to reload the window. Below this there are three additional icons that control the Bokeh server, which controls the communication with the Python side of the Machine Learning plugin. The server should start automatically. In case of problems, it can be controlled manually via the **Start** and **Stop** icons. The current status of the Bokeh server can be checked by viewing the **Bokeh server log file**.

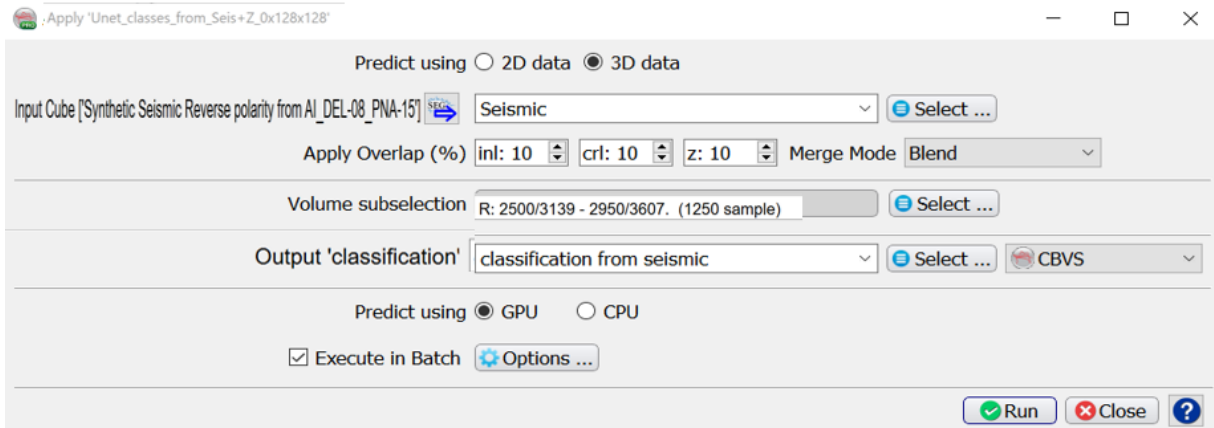


When the processing log file shows “Finished batch processing”, you can move to the **Apply** tab.

Apply

Once the Training is done, the trained model can be applied. Select the trained model and press **Proceed**.



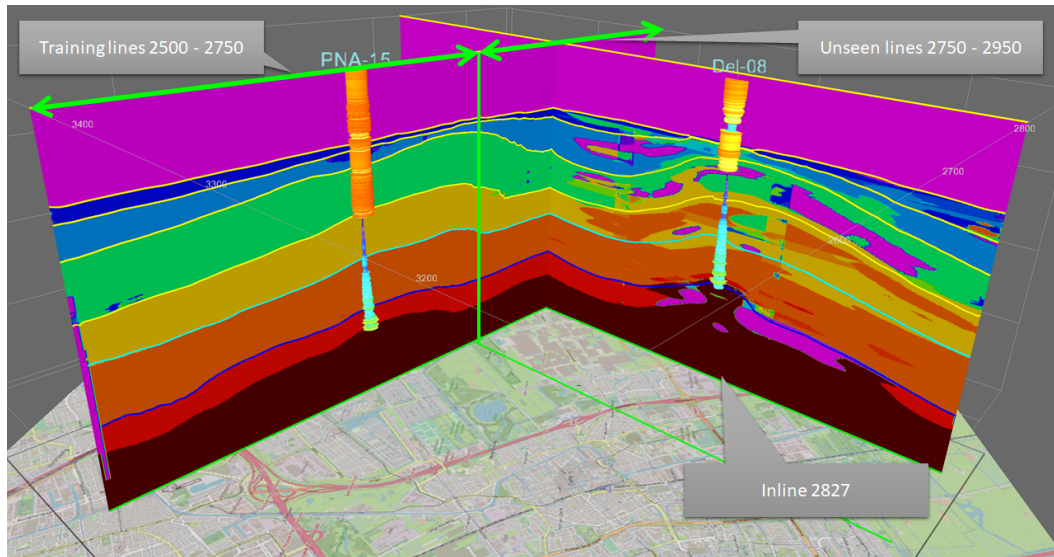


The Apply window pops up. Here you optionally apply to a **Volume subselection**. You can output a **Classification** Volume (values 1 to 8 in this case).

Probability is simply the output of the prediction node which ranges between 0 and 1.

Confidence is computed as the value of the winning node minus the value of the second best node.

(You can run on GPU or CPU depending on the **Predict using** toggle). Running the application on a GPU is many times faster than running it on a CPU. Press Run to create the desired outputs. The image below is taken from a seismic facies Classification cube generated in this way.



Output of the trained segmentation model: seismic segmentation cube. Input features are extracted from synthetic seismic and Two-Way-Time volumes. Images are 128x128 samples along the inline direction. The model was trained on examples extracted from inlines 2500 - 2750. Prediction is from real seismic and TWT. Model is probably overfitted. Real seismic is more variable than the synthetic data it was trained on. Possible improvements to consider: if possible, create a synthetic model from more real wells, apply data augmentation (i.e. randomly changing input examples, e.g. by cropping, padding, and flipping left-right). Data augmentation requires programming in the Python environment. For more info on programming in OpendTect Machine Learning environment, please visit [this webpage](#).

3.4.4 Seismic Image Regression (Supervised 2D and 3D)

A Seismic Image Regression model maps a seismic image (1D, 2D, 3D) to an image with continuous values of similar size. A mapping of seismic inputs (e.g. Acoustic Impedance) to a rock property, such as Porosity, is an example of a Regression model.

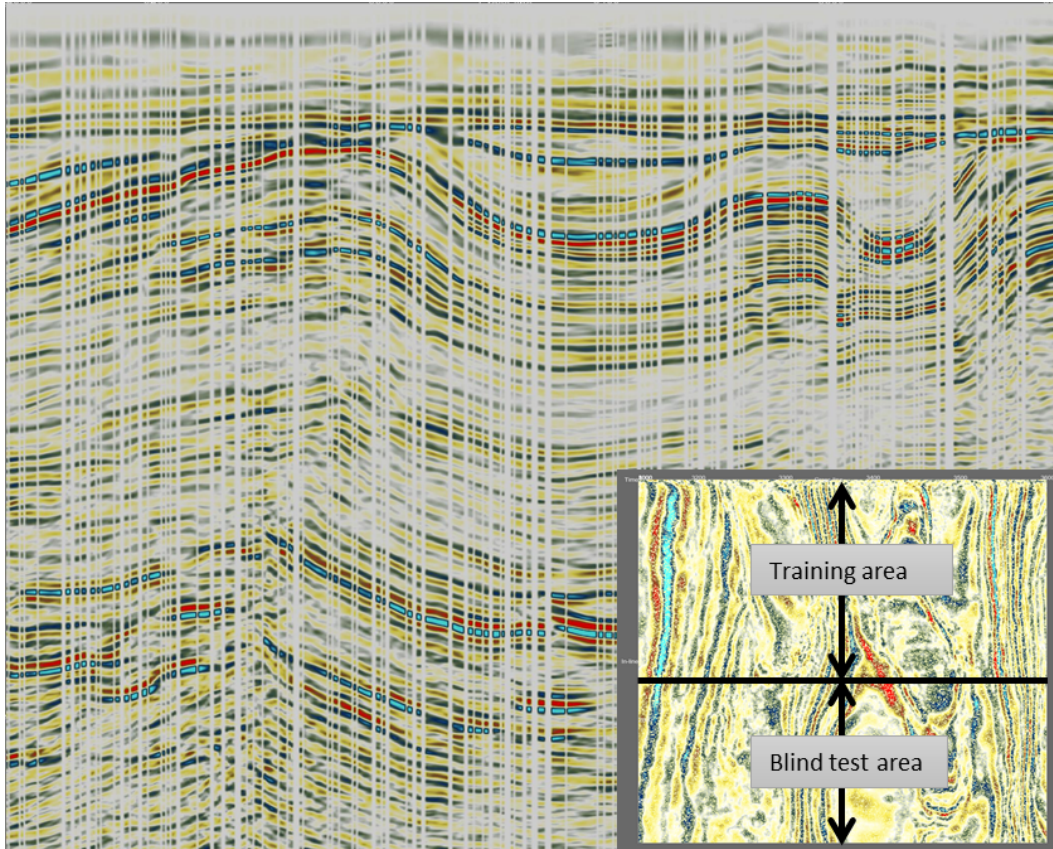
These workflows are categorized as supervised methods in that the input data set(s) and the target data are provided to facilitate training. Application of the trained model delivers a new volume (3D seismic) or a new attribute to a line set (2D seismic). We will discuss the workflow and user interface components on the basis of a 3D example.

The objective of this workflow is to interpolate seismic traces where those traces are missing. The model will have to learn how to recreate an image from example images containing blank traces. Therefore, we need an input data set in which we have deliberately blanked some of the traces. In our workflow we do this by randomly blanking approx. 33% of all traces (see caption of the image below for details on how this can be done in OpendTect).

In this example workflow we train a 2D Unet but you can equally well train a 3D Unet. The differences between 2D and 3D Unets are as follows:

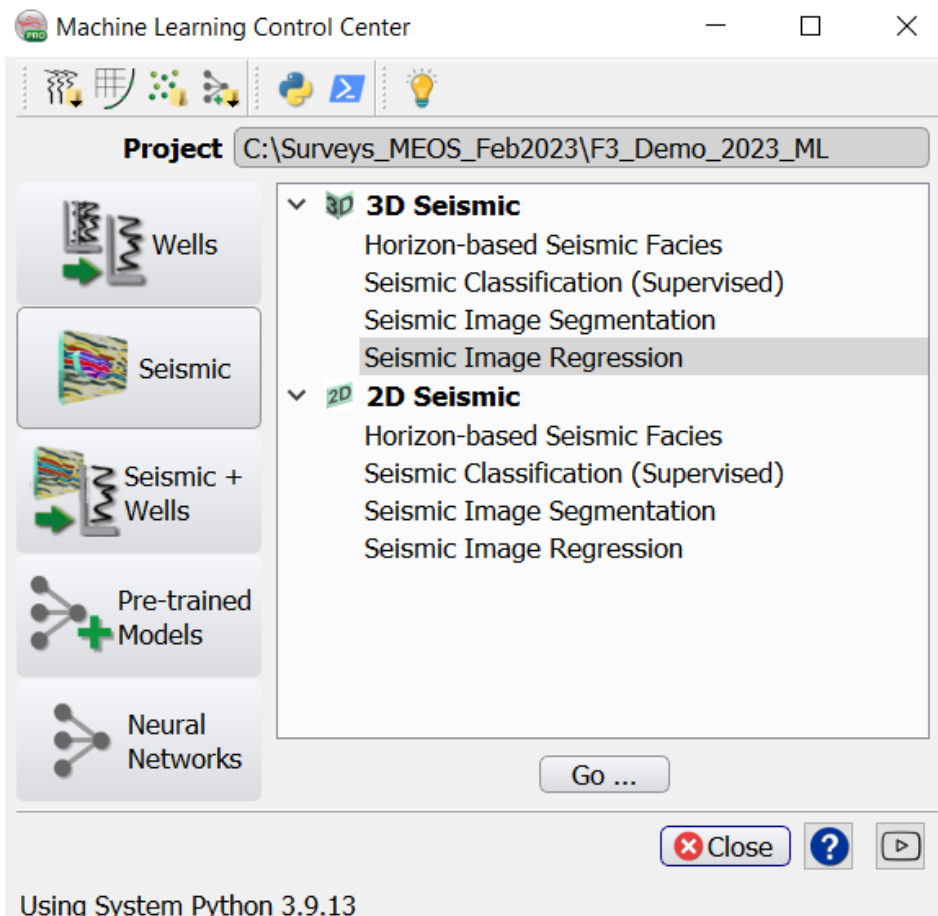
1. A 2D model trains much faster (hours vs days)
2. 2D models can be trained on workstations with less GPU / CPU capacity
3. Interpolation results are comparable although 2D interpolation may introduce some striping (like a footprint)

4. Application of a trained 3D model is much faster than a trained 2D model (minutes vs hours)



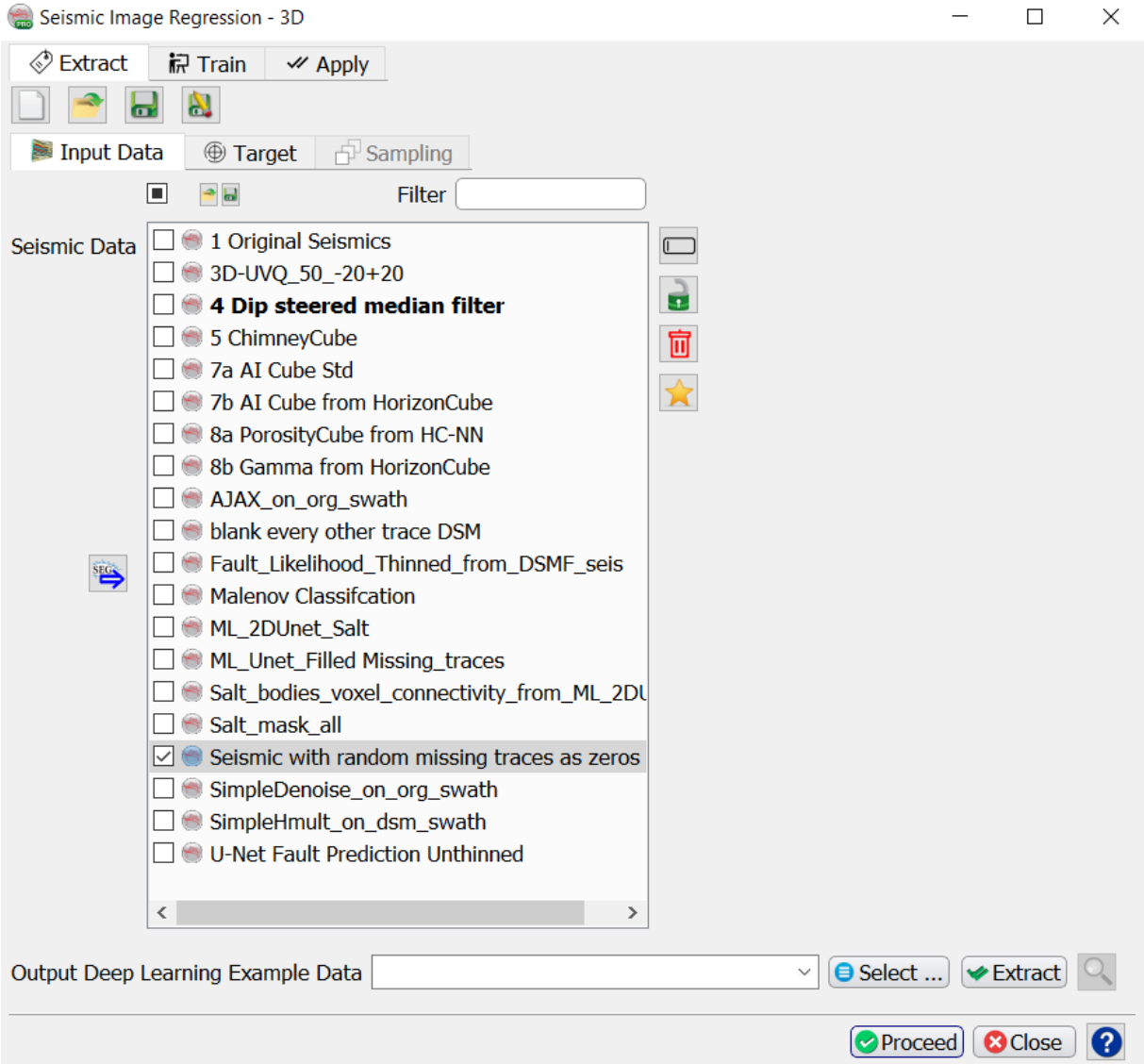
Randomly blanked traces. To train our 2D Unet regression model we create a data set with 33% randomly blanked traces. From this cube we extract examples for training in a restricted area (insert time-slice). The trained model is applied to the entire volume, whereby the area from which no examples are extracted acts as blind test area. The real value is of course when we apply the trained model to an area with real missing traces (which we don't have in this case). Random blanking (replacing the values with hard zeros) is done in OpendTect's Attribute engine and can be done in different ways. In this case: 1) Math attribute with formula: "randg(1)". This generates random values

with a Gaussian distribution and 1 standard deviation; 2) Apply this attribute to a horizon and save as horizon data; 3) Horizon attribute that retrieves the random values from the saved horizon data. A Horizon attribute replaces a value at an inline, crossline position with the value extracted from the given horizon; 4) Math attribute with formula: “abs(value)> 1 ? 0: seis”. We assign the retrieved horizon data to the variable “value” and the seismic data to “seis”. This attribute assigns values larger than the absolute value of 1 standard deviation to zero while all other values are given the value of the seismic data.

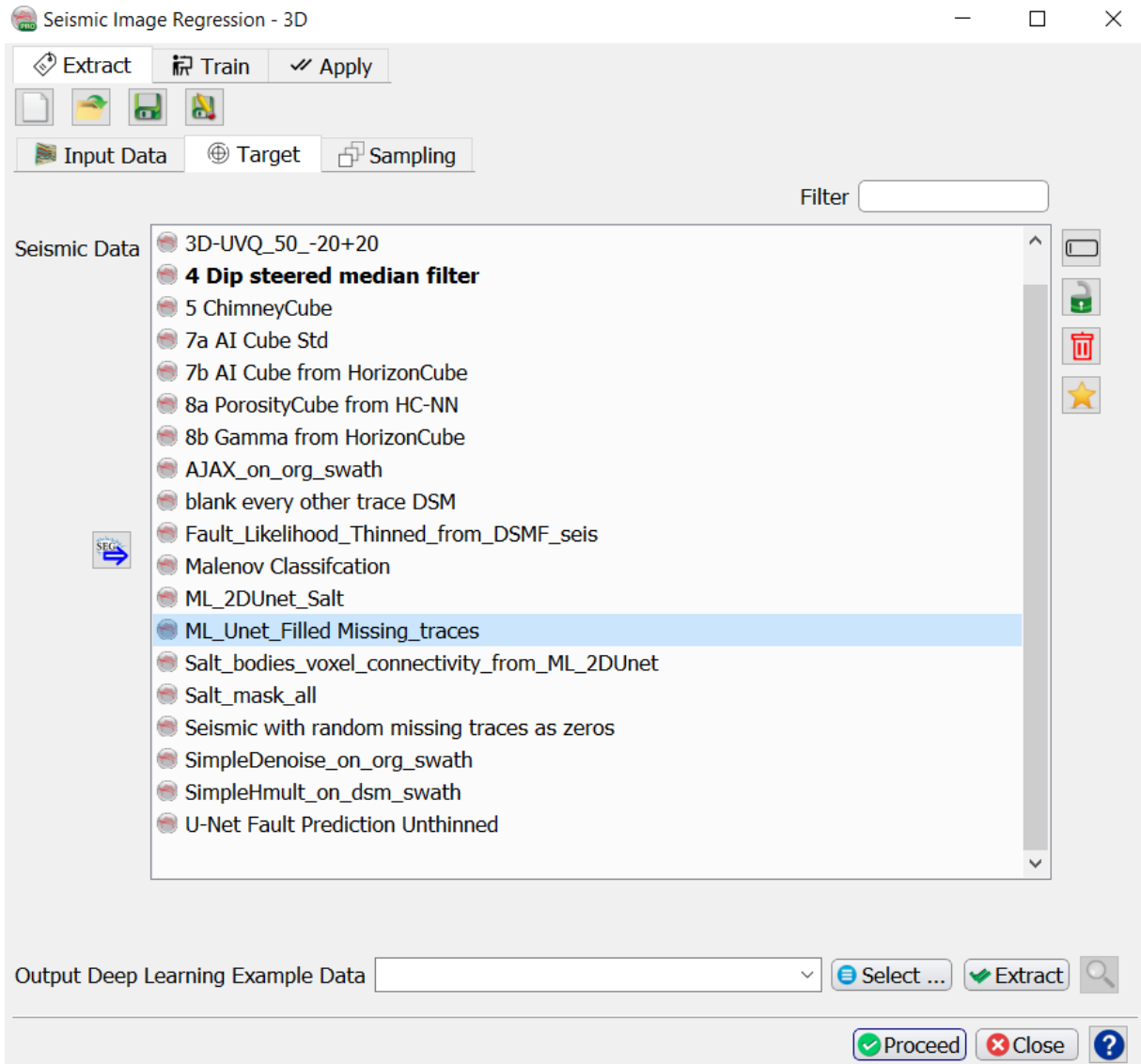


Select the Seismic Image Regression workflow and press **Go**.

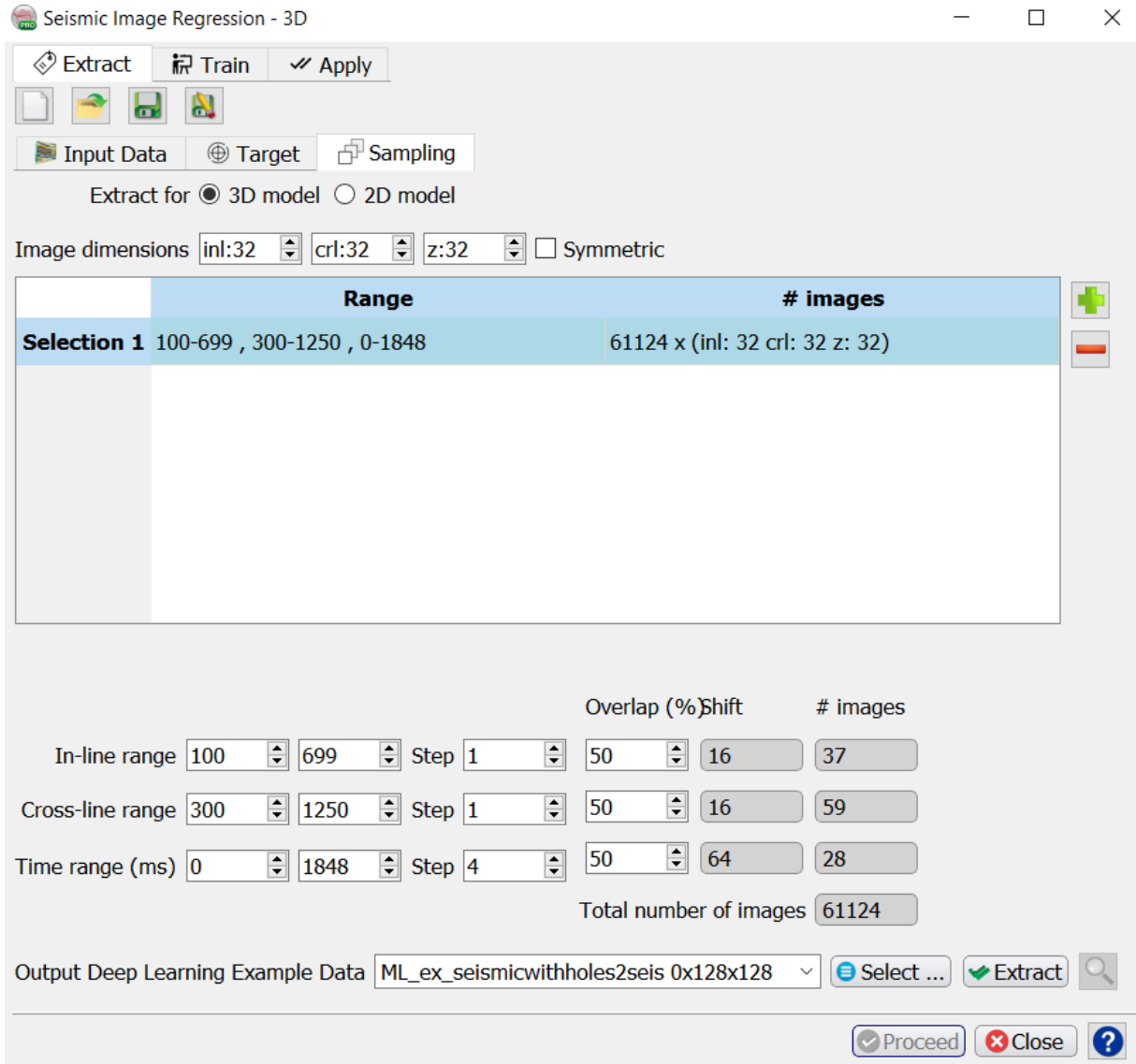
Extract Data



Input Data tab, Select the input seismic data.



Target tab, Select the target seismic volume containing the labels.



Sampling tab, specify the **Image Dimension**, the number of samples in inline, cross-line and Z directions of the input features.

Examples:

- Inl: 0; Crl: 128, Z: 128 - extracts 2D images along inlines with dimensions 128 x 128 samples
- Inl: 64, Crl: 64, Z: 64 - extracts cubelets of 64x64x64 samples.

The examples (images) are extracted from the specified **Inline**, **Crossline** and **Time Ranges**. **Step** is the sampling rate in the bin (time) directions. The **Overlap** determines how much overlap there is between neighboring images. The total number of images is given on the right-hand side of the window. You can control this number by playing with the ranges and overlap parameters.

It is possible to create a Training Set from different selections by pressing the **+** icon.

The selection can be **Saved**, retrieved (**Open**) and **Deleted** using the corresponding icons.

Specify the name of the **Output Deep Learning Example Data** and press **Extract** to start the extraction process.

When this process is finished the **Proceed** button has turns green. Press it to continue to the Training tab.

Training

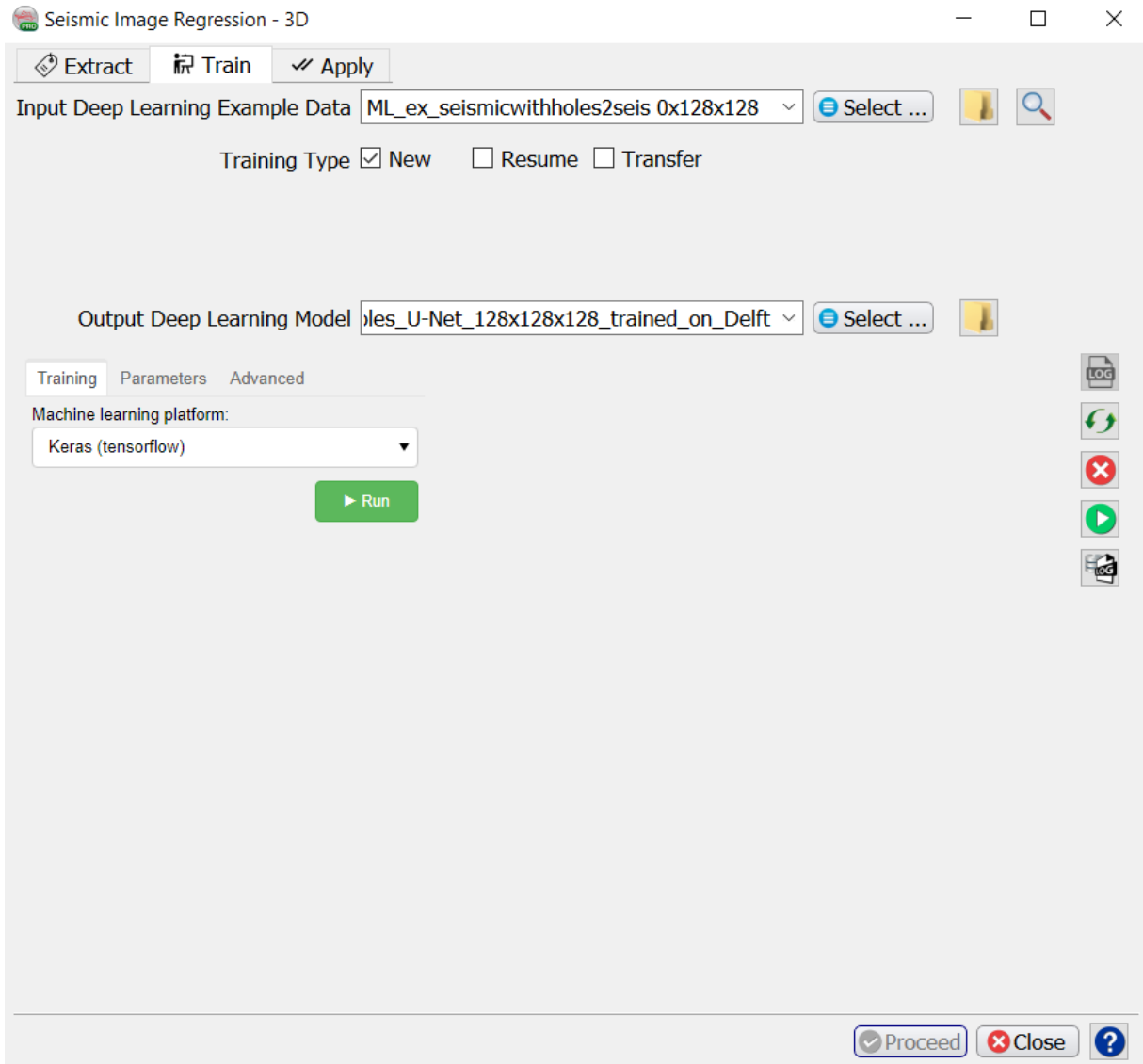
When you already have stored data, you can start in the **Training tab** with **Select** training data (Input Deep Learning Example data).

Under the Select training data there are three toggles controlling the **Training Type**:

- **New** starts from a randomized initial state.
- **Resume** starts from a saved (partly trained) network. This is used to continue training if the network has not fully converged yet.
- **Transfer** starts from a trained network that is offered new training data. Weights attached to Convolutional Layers are not updated in transfer training. Only weights attached to the last layer (typically a Dense layer) are updated.



After the training data is selected the UI shows which models are available. For seismic image workflows we use **Keras (TensorFlow)**.

Check the **Parameters** tab to see which models are supported and which parameters can be changed.




The dGB Unet is an auto-encoder - decoder type of Convolutional Neural Network. The architecture is shown schematically below.

Extract **Train** **Apply**

Input Deep Learning Example Data  

Training Type New Resume Transfer

Output Deep Learning Model 

Training Parameters Advanced

Type

Validation Percentage Split: 0.20

Batch Size

Epochs: 15

Early Stopping: 10

Initial Learning Rate (1e): -4

Epoch drop (%): 33.33

Size: 1 byte

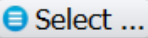
Decimate input
 Train on GPU

 Extract  Train  Apply

Input Deep Learning Example Data 



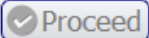
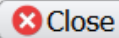
Training Type New Resume Transfer

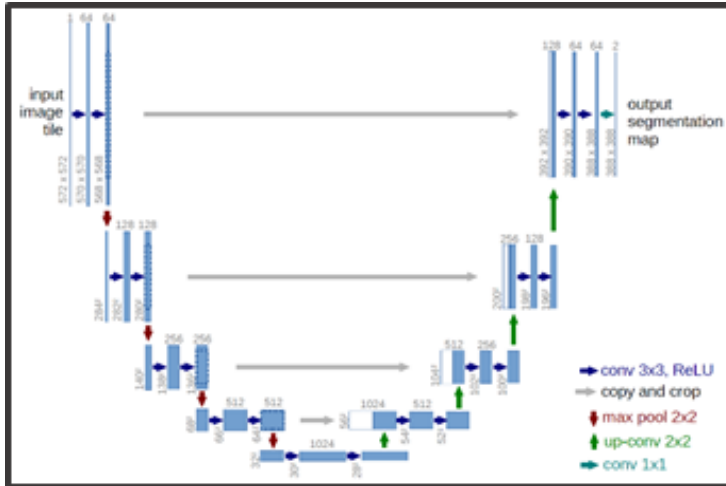
Output Deep Learning Model 



Training Parameters **Advanced**

- Use Mixed Precision
- Enable Tensorboard
- Clear Tensorboard log files



The following training parameters can be set in the **Parameters** and **Advanced** tabs:

Batch Size: this is the number of examples that are passed through the network after which the model weights are updated. This value should be set as high as possible to increase the representativeness of the samples on which the gradient is computed, but low enough to have all these samples fit within the memory of the training device (much smaller for the GPU than the CPU). If we run out of memory (raises a Python OutOfMemory exception), lower the batch size!

Note that if the model upscales the samples by a factor 1000 for instance on any layer of the model, the memory requirements will be upscaled too. Hence a typical 3D U-net model of size 128-128-128 will consume up to 8GB of (CPU or GPU) RAM.

Epochs: this is the number of update cycles through the entire training set. The number of epochs to use depends on the complexity of the problem. Relatively simple CNN networks may converge in 3 epochs. More complex networks may need 30 epochs, or even hundreds of epochs. Note, that training can be done in steps. Saved networks can be trained further when you toggle **Resume**.

Early Stopping: this parameter controls early stopping when the model does not change anymore. Increase the value to avoid early stopping.

Initial Learning rate: this parameter controls how fast the weights are updated. Too low means the network may not train; too high means the network may overshoot and not find the global minimum.

Epoch drop: controls how the learning rate decays over time.

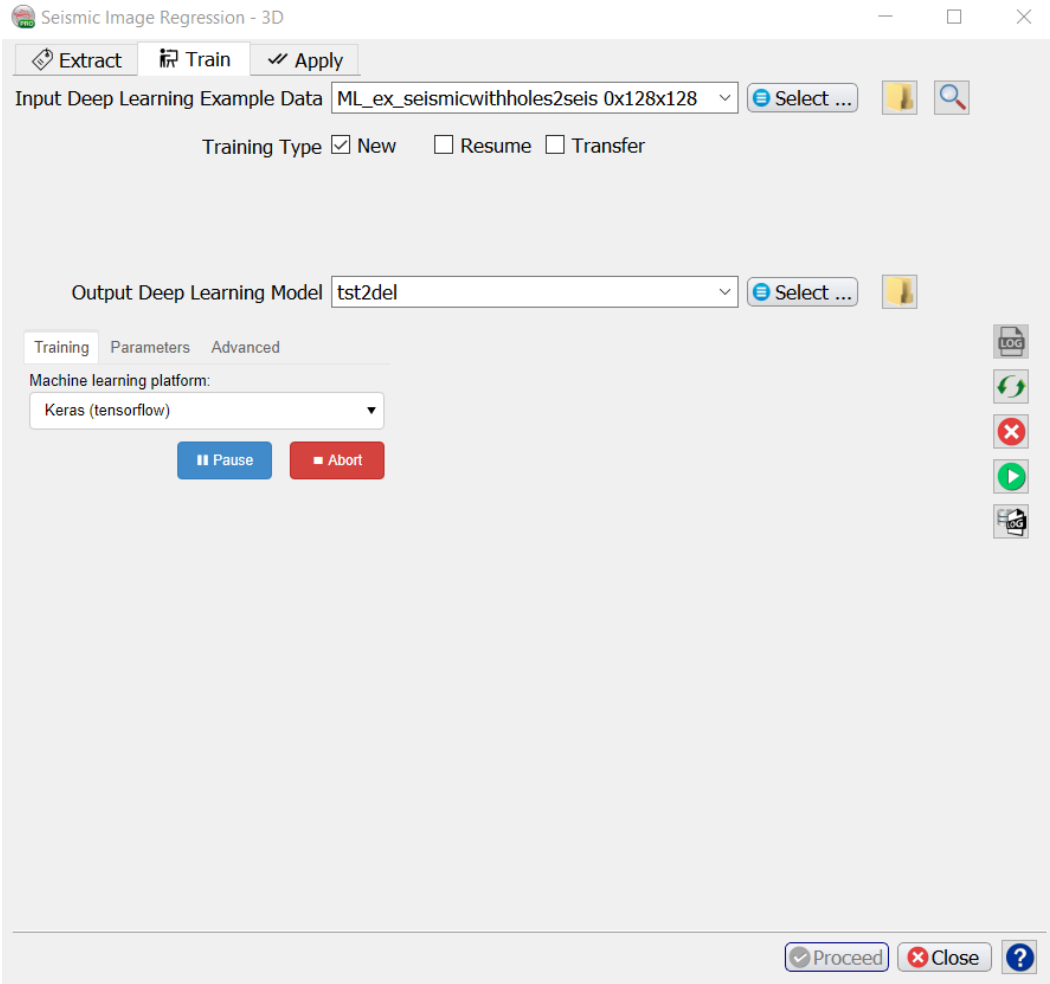
Decimate Input: this parameter is useful when we run into memory problems. If we decimate the input the program will divide the training examples in chunks using random selection. The training is then run over chunks per epoch meaning the model will eventually have seen all samples once within one epoch, but only no more than one chunk of samples will be loaded in RAM while the training is performed.

Mixed precision: Mixed precision option has been added to dgbpy for both PyTorch and Keras platform. This allows to train models using a combination of 16-bit and 32-bit precision formats which will optimize GPU memory usage and efficiency. The GPU memory usage can be reduced by 50% and 75% in some cases. It also increases training speed 2 - 3 times on GPUs with Volta or Turing architectures. This doesn't also affect training accuracy or performance.

For more details about what is 'Mixed precision training' please refer to:

- [Keras documentation: Mixed precision](#)
- [NVIDIA documentation: Train with Mixed Precision](#)

After model selection, return to the **Training tab**, specify the **Output Deep Learning model** and press the green **Run** button. This starts the model training. The Run button is replaced in the UI by **Pause** and **Abort** buttons.



TensorBoard SCALARS GRAPHS

Filter tags (regular expressions supported)

Show data download links
 Ignore outliers in chart scaling
 Tooltip sorting method: default

Smoothing

0,6

Horizontal Axis

STEP RELATIVE WALL

Runs

Write a regex to filter runs

- Delft_ML_test_run1_2020-08-26T13m34\train
- Delft_ML_test_run1_2020-08-26T13m34\validation
- Delft_ML_test_run2_2020-08-26T13m55\train
- Delft_ML_test_run3_2020-08-26T13m58\train
- Delft_ML_test_run3_2020-08-26T13m58\validation
- Delft_ML_test_run4_2020-08-27T12m43\train
- Delft_ML_test_run4_2020-08-27T12m43\validation

TOGGLE ALL RUNS

D:\Surveys\MachineLearning\Seis+Z_to_classes_0x128x128

epoch_accuracy

epoch_loss

epoch_lr

epoch_mae

The progress can be also followed in a text log file. If this log file does not start automatically, please press the **log file icon** in the toolbar on the right-hand side of the window.

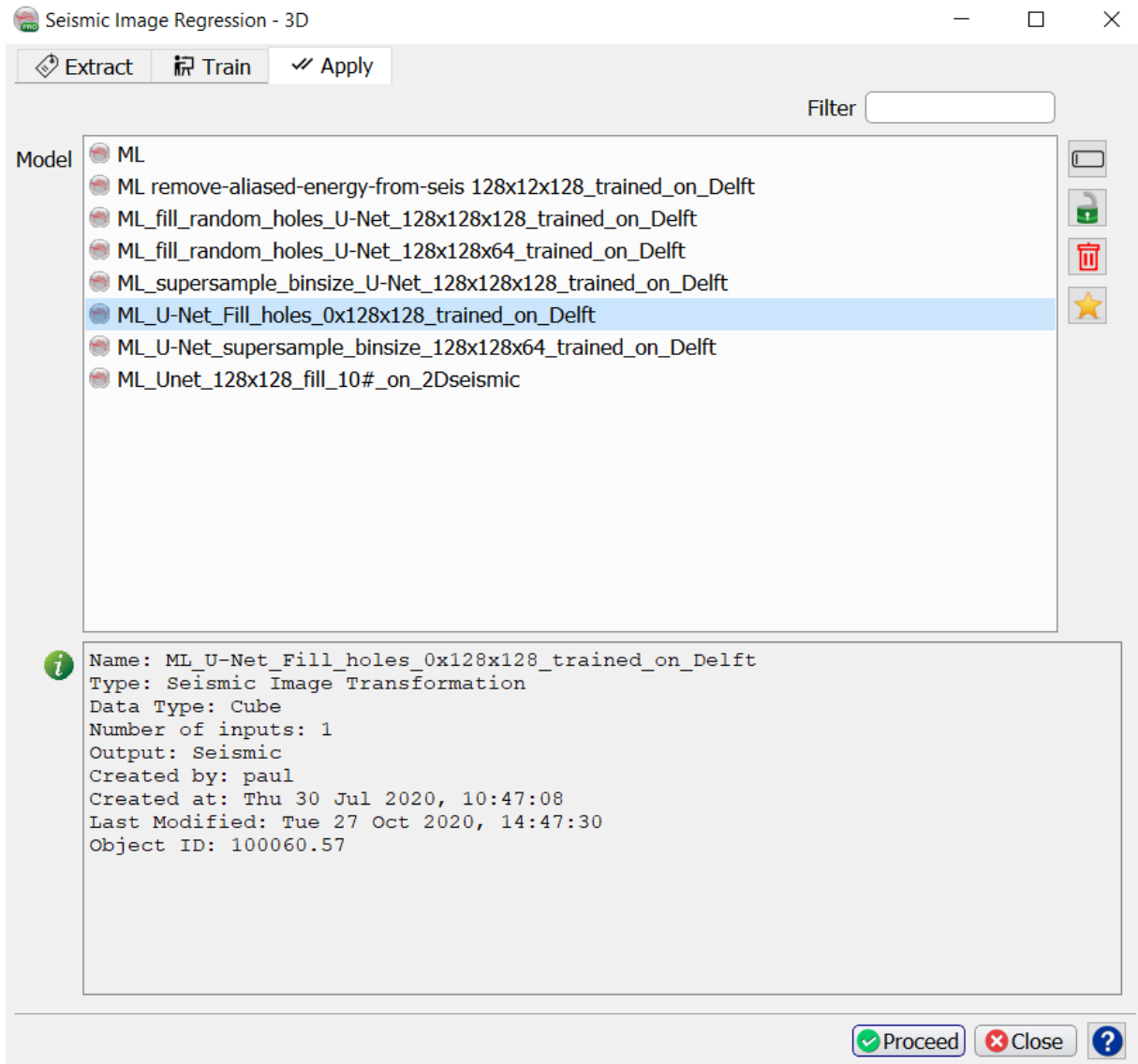
Below the log file icon there is a **Reset** button to reload the window. Below this there are three additional icons that control the Bokeh server, which controls the communication with the Python side of the Machine Learning plugin. The server should

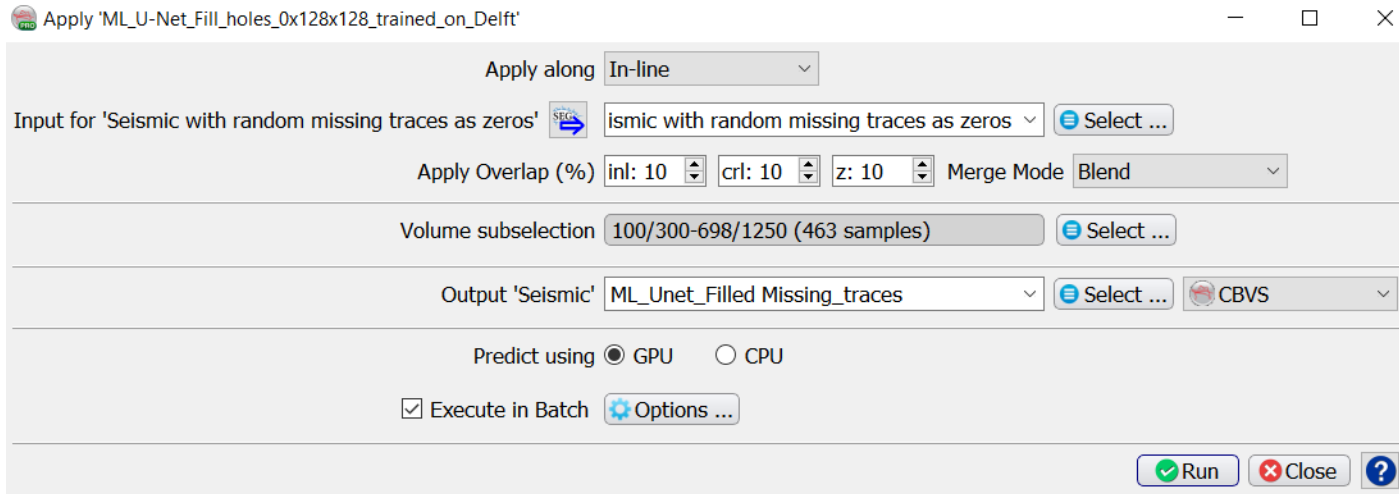
start automatically. In case of problems it can be controlled manually via the **Start** and **Stop** icons. The current status of the Bokeh server can be checked by viewing the **Bokeh server log file**.

When the processing log file shows “Finished batch processing”, you can move to the **Apply** tab.

Apply

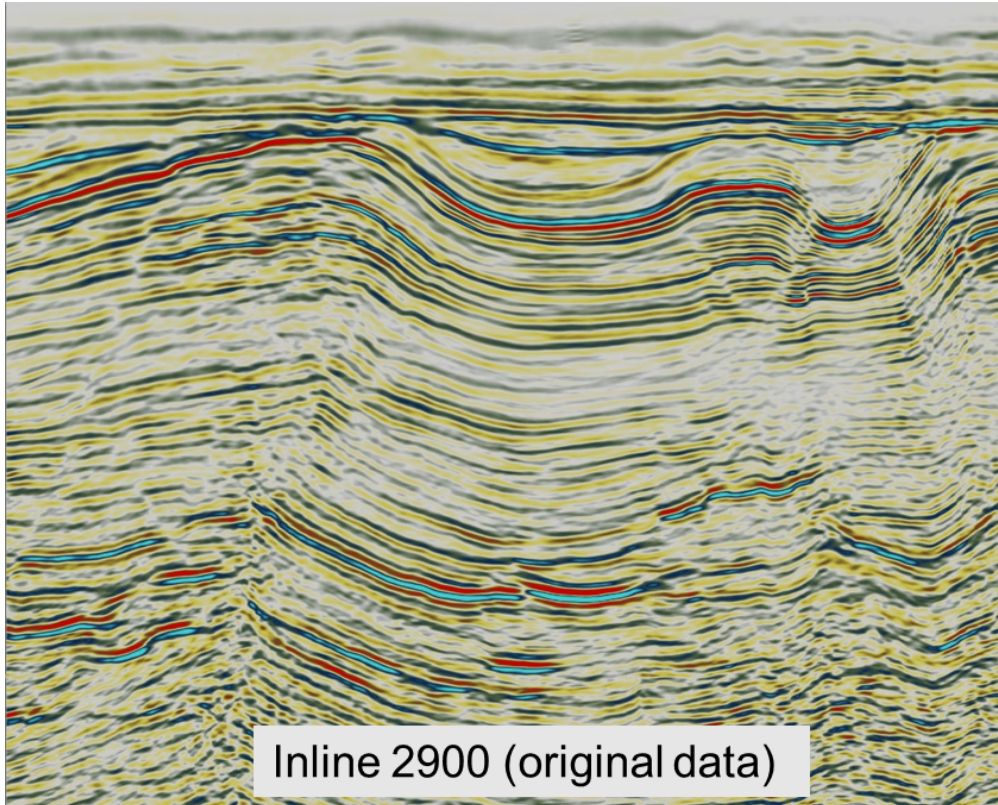
Once the Training is done, the trained model can be applied. Select the trained model and press **Proceed**.

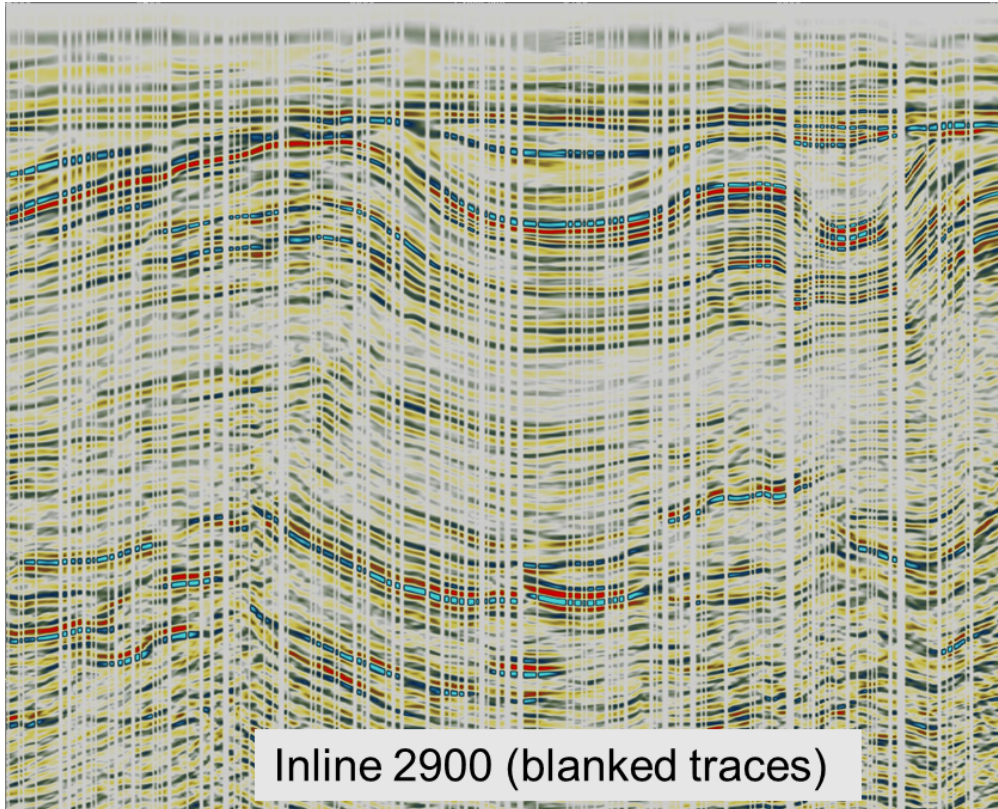


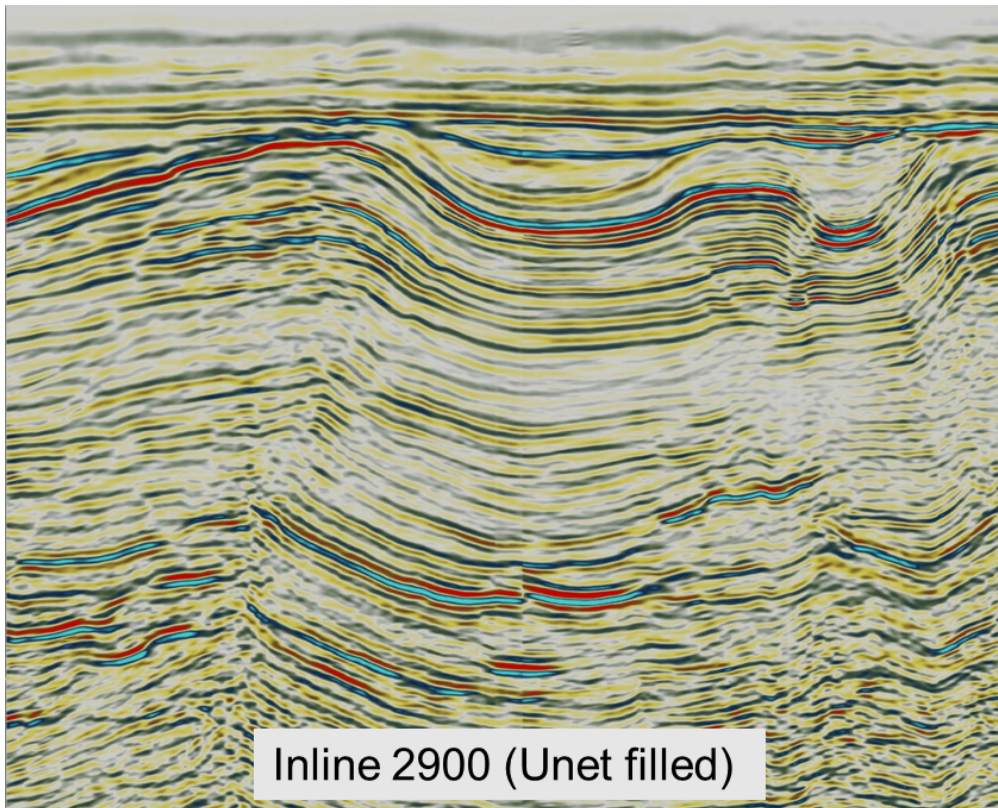


The **Apply** window pops up. Here you optionally apply to a **Volume subselection**.

You can run on GPU or CPU depending on the **Predict using** toggle. Running the application on a GPU is many times faster than running it on a CPU. Press **Run** to create the desired outputs. The images below compare the original data with the Unet reconstructed result from input with blanked traces. The line is extracted from the blind test area.



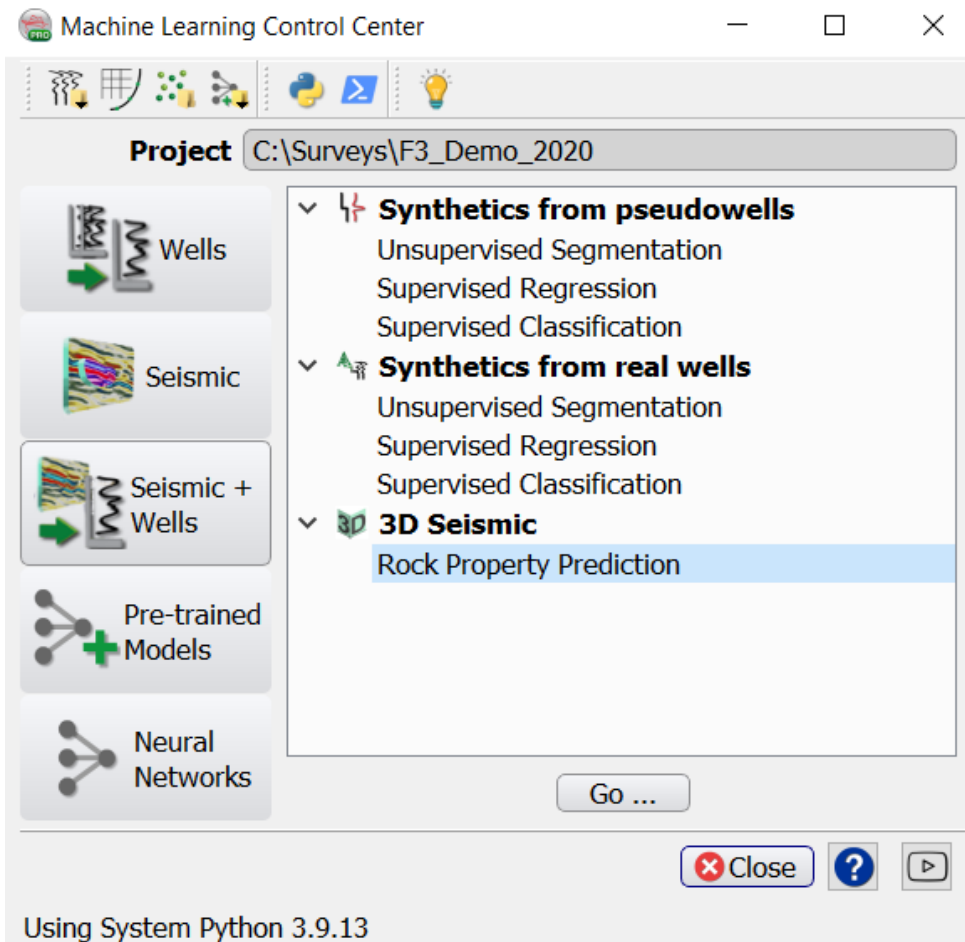




Inline 2900 (Unet filled)

3.5 Seismic + Wells models

The aim of seismic + wells workflows is to predict a well log property from seismic input features. Input and target can either come from real wells and real seismic features (e.g. elastic impedance data) or from synthetic data created by SynthRock.



We will explain the user interface on the basis of a simple example in which we create a Porosity volume from an inverted Acoustic Impedance volume.

Please note that the current implementation of this workflow only accepts input features from the seismic side (most often in Two-Way-Time) while target features are extracted from well logs (in depth). This is not desirable because to extract matching inputs and targets the well ties must be perfect. This is never the case. As we only need to find a relationship between Acoustic Impedance and Porosity it makes sense to extract both inputs and targets from logs. In this case: Acoustic Impedance logs and corresponding Porosity logs. This ensures alignment over the entire well track. Application of this relationship (the trained model) will transform the AI volume to a Porosity volume with the same mis-ties inherent in the AI volume.

The workflow described above will be implemented in the near future. Training data can then be extracted from real wells and from pseudo-wells created with SynthRock. The workflow can already be emulated in OpendTect albeit a bit cumbersome. It requires a few extra data preparation steps: use the **LogCube** option to convert the input AI logs to pseudo-seismic cubes at 4ms sampling (you can even use and create extra attribute cubes from them like log of AI, sqrt of AI, derivative of AI with depth etc. to have more input features). Before using these cubes to predict porosity in the Seismic+Wells UI, you might want to apply a low pass filter to have them at the same frequency as the inverted AI.

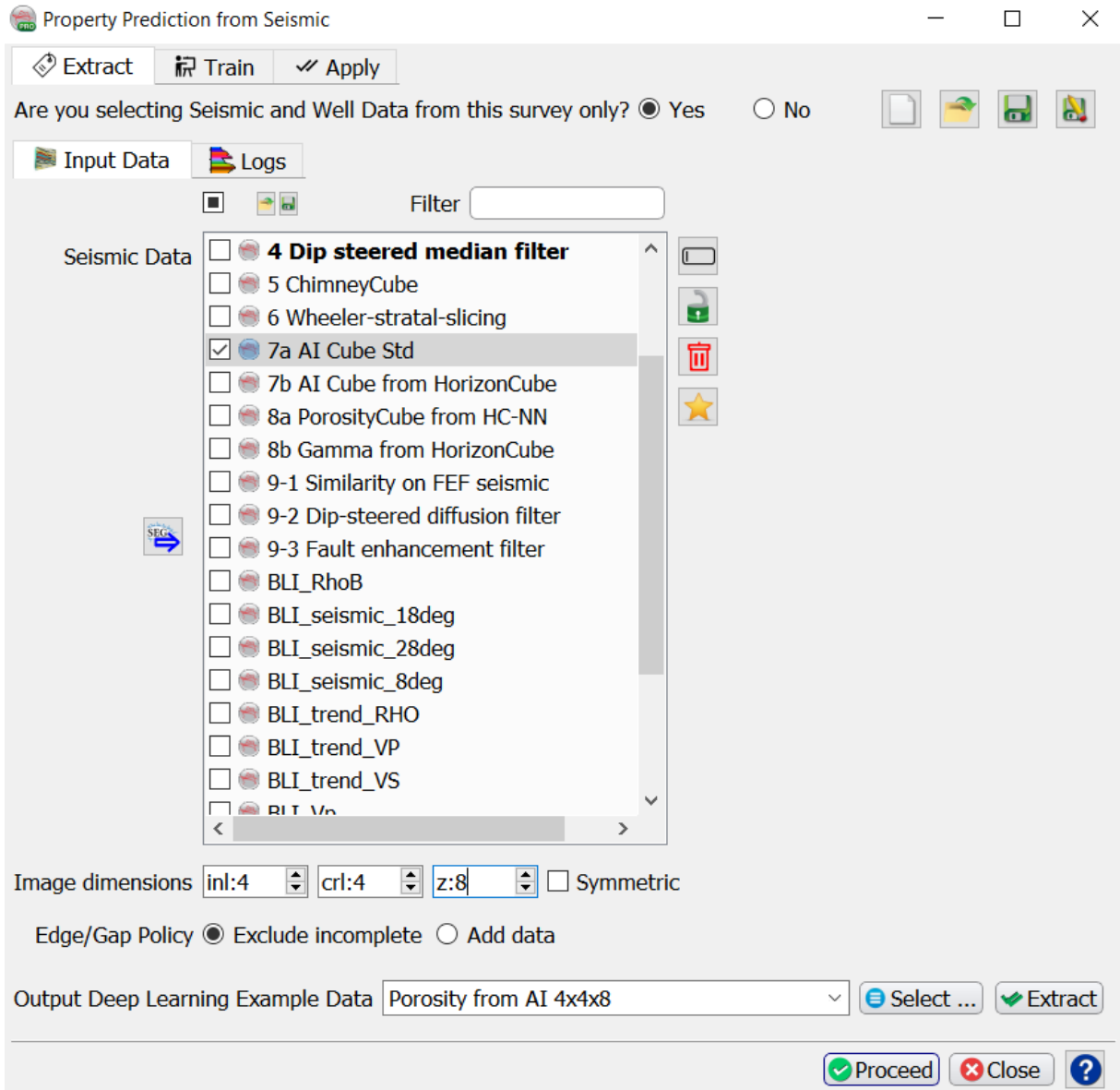
Let's follow the workflow as implemented in the current release without such pre-processing steps. We have 4 wells with Porosity logs. Select the workflow and press **Go**.

Extract Data

Input Data

In the **Input Data** tab of the 'Property Prediction' Window, **Select** the input feature(s). In this case only one volume is selected: the Acoustic Impedance Cube with name "7a AI Cube std".

Select the **Target Log**. In this demo exercise we select all four wells for training. In a real study we recommend reserving at least one well as blind test well (Validation Set).

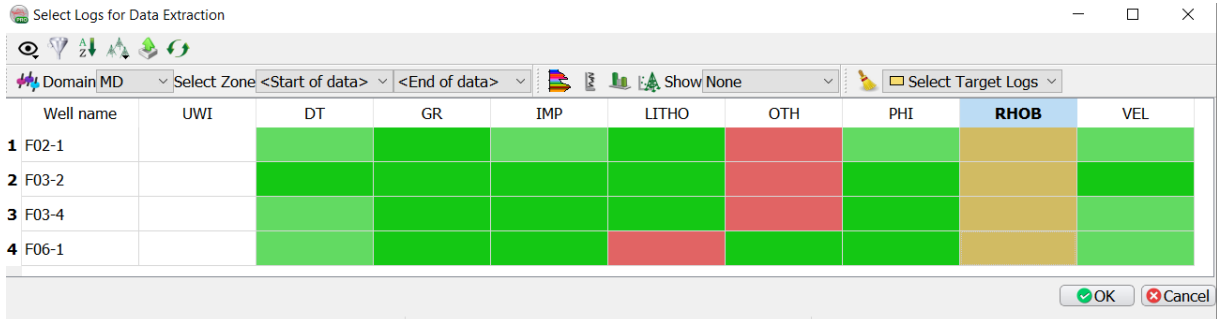


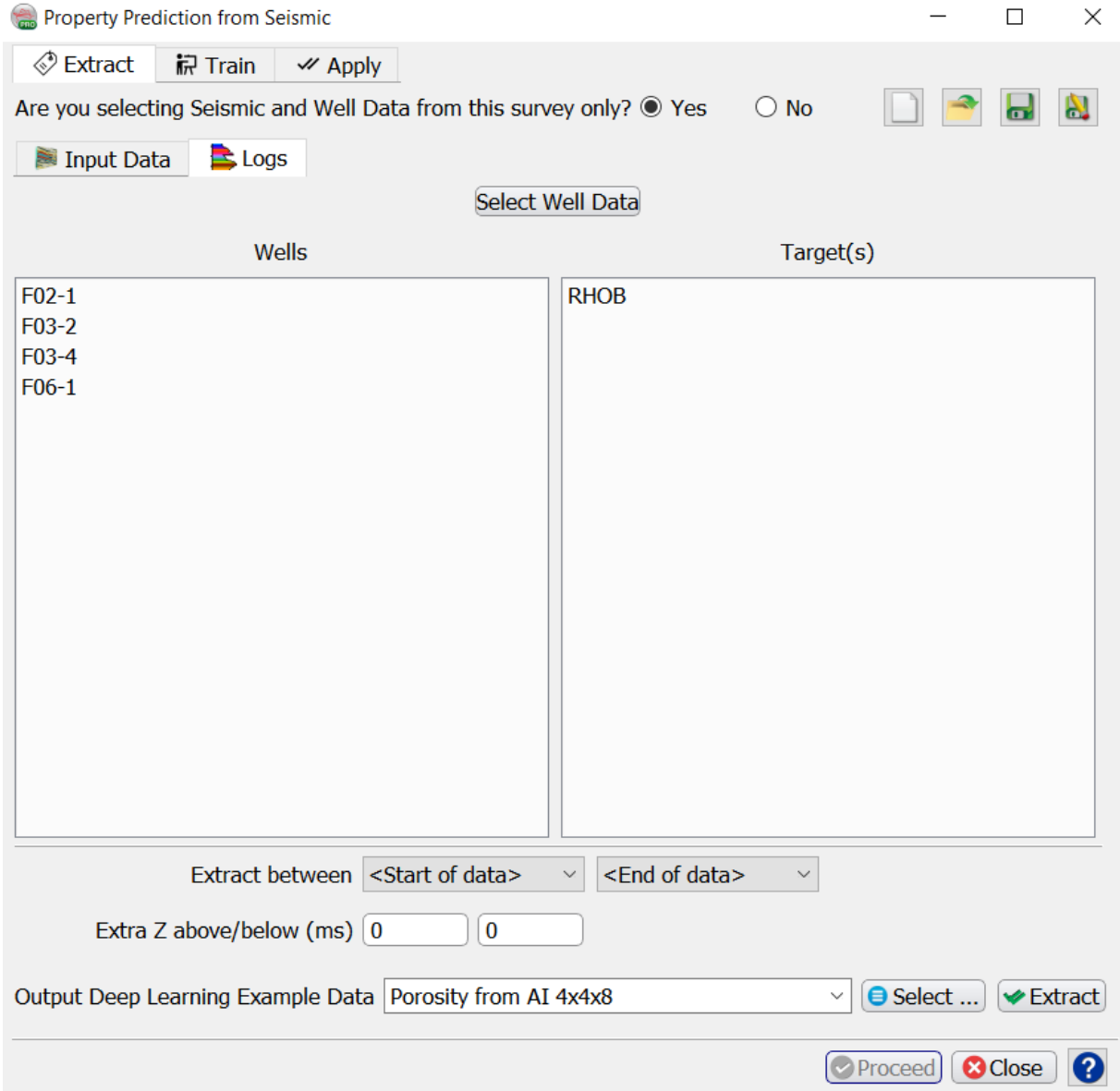
The **Image dimensions** specify how many traces in inline and crossline steps around the well track will be extracted. The Image dimensions in inline and crossline directions multiply our examples. Each example has slightly different seismic characteristics but the target porosity value at each position along the track is the same.

The **Edge/Gap Policy** controls how we handle examples with missing input features. With a **Stepout** of 10 we need 10 samples above and 10 samples below each prediction point. The default **Exclude incomplete** will not return a value meaning we lose 10 samples above and 10 samples below each gap. If the toggle is set to **Add data**, the value at the edge of an input feature is copied Stepout times to ensure the model can make a prediction at the edge. In other words the predicted log covers the same interval as the input logs (gaps do not increase).

Log Tab

In the **Log** tab, select the **Target Logs**. In this exercise select all four wells for training. In a real study we recommend reserving at least one well as blind test well (Validation Set). Then press **OK**.





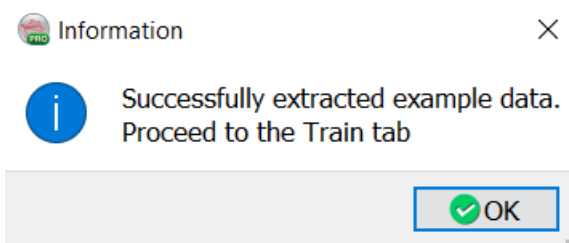
To extract only in a zone of interest, we can select the corresponding markers to **Extract between** a pair of specified start and end points. If left to defaults, the entire log will be used.

The Extra Z above/below (ms) depends on the velocity. We have seismic in TWT and logs in depth. We want to have roughly the same sampling rates for both (we can

ensure this, as well as perfect alignment if we use the LogCube preprocessing steps explained above).

The selection can be **Saved**, retrieved (**Open**) and **Removed** using the corresponding icons.

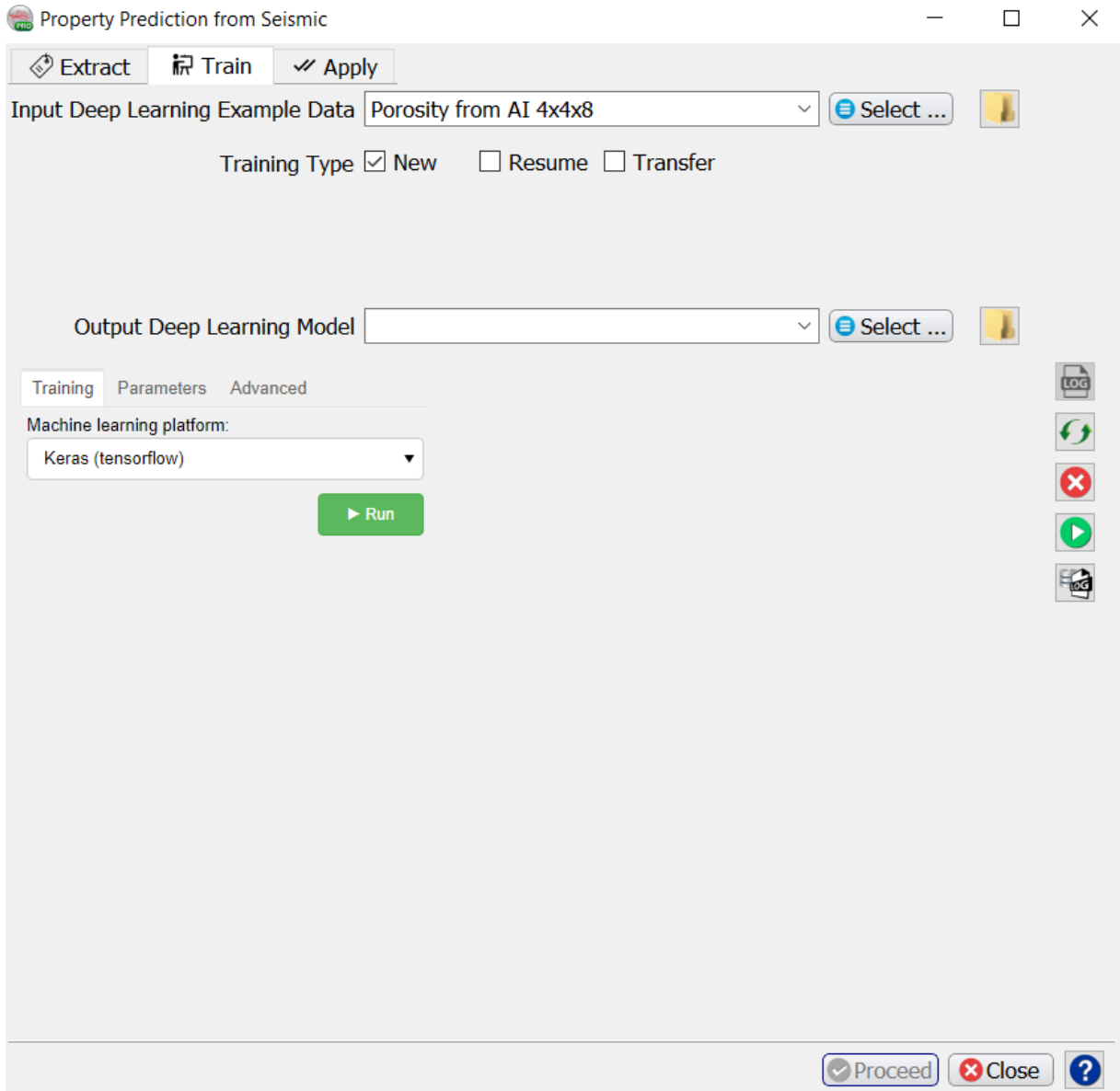
Specify an output name for the **Deep Learning Example Data** (the Training Set) and press **Extract**.



Press **Proceed** to continue to the Training tab.

Training

When we already have stored the extracted data for training, we can move on to the **Training tab** with **Select** training data (Input Deep Learning Example data). After the training data is selected the UI shows which models are available.



Under the Select training data there are three toggles controlling the **Training Type**:

- **New** starts from a randomized initial state.
- **Resume** starts from a saved (partly trained) network. This is used to continue training if the network has not fully converged yet.
- **Transfer** starts from a trained network that is offered new training data. Weights attached to Convolutional Layers are not updated in transfer training. Only weights attached to the last layer (typically a Dense layer) are updated.

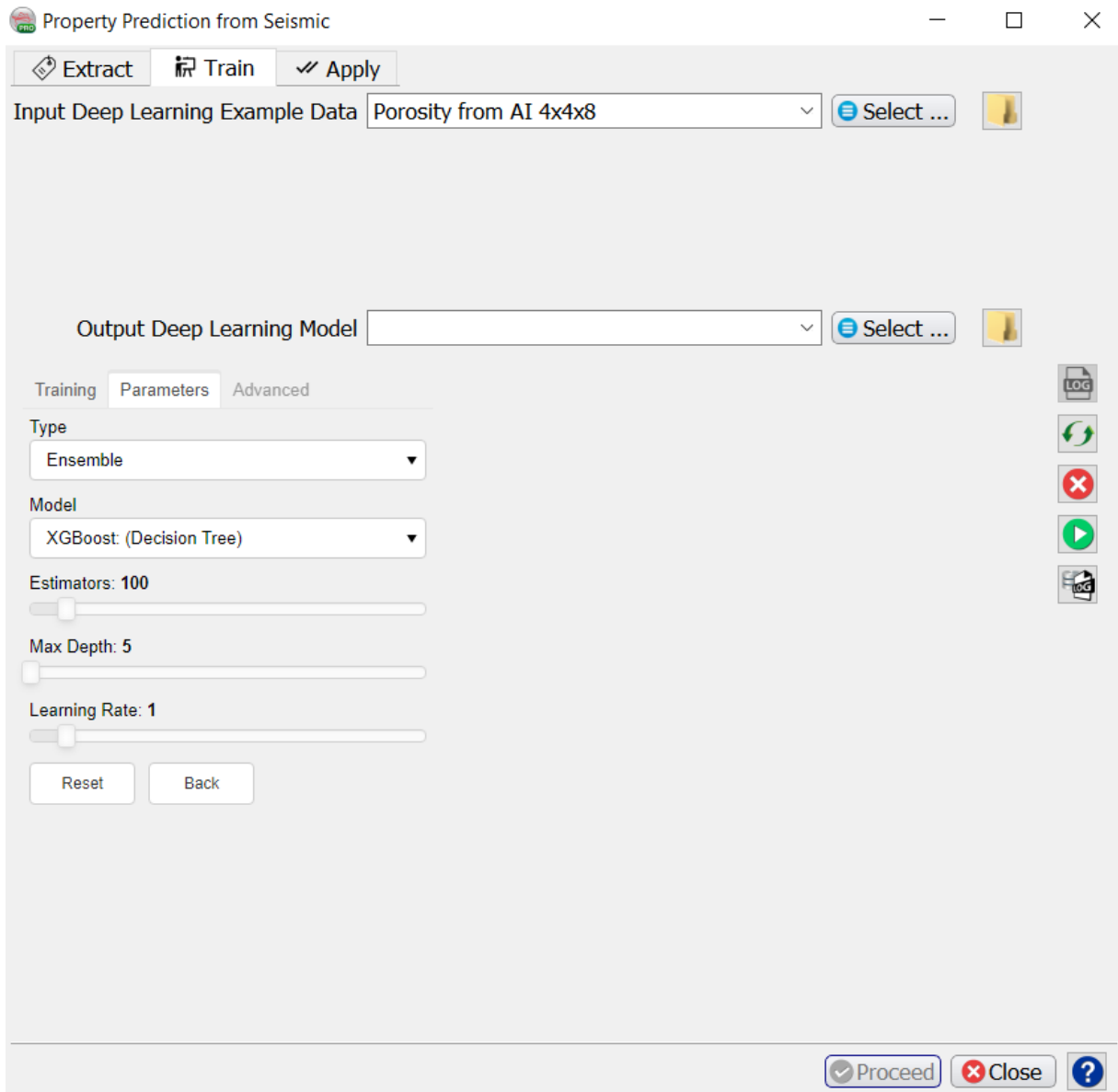
The supported models are divided over three platforms: **Scikit Learn**, **PyTorch** and **Keras (TensorFlow)**.

Check the **Parameters** tab to see which models are supported and which parameters can be changed. From **Scikit Learn** we currently support:

1. Linear Regression
 - a. Ordinary Least Squares
2. Ensemble Methods
 - a. Random Forests
 - b. Gradient Boosting
 - c. Adaboost
 - d. XGBoost
3. Neural Networks
 - a. Multi-Layer-Perceptrons
4. Support Vector Machines
 - a. Linear
 - b. Polynomial
 - c. Radial Basis Function
 - d. Sigmoid

Our best results have been generated with Random Forests and XGBoost (eXtreme Gradient Boosting of Random Forest Models). Random Forests build a bunch of trees at once and average the results out at the end, while gradient boosting generates and adds one tree at a time. Because of this gradient boosting is more difficult for parameter tuning, but since it goes more in depth, it could result in higher accuracy for complicated or imbalanced datasets.

From a statistics point of view, a Random Forest is a 'bagging' algorithm, meaning that it combines several high variance, low-bias individual models to improve overall performance. Gradient boosting is a 'boosting' algorithm, meaning that it combines several high-bias, low variance individual models to improve overall performance. For parameter details we refer to [the Scikit Learn User Guide](#).



After model selection, return to the **Training tab**, specify the **Output Deep Learning model** and press the green **Run** button. This starts the model training. In this tab, the **Run** button is now replaced by **Pause** and **Abort** buttons.

The progress can be followed in a text log file. If this log file does not start automatically, please press the **log file icon** in the toolbar on the right-hand side of the window.

Below the log file icon there is a **Reset** button to reload the window. Below this there are three additional icons that control the Bokeh server, which controls the communication with the Python side of the Machine Learning plugin. The server should start automatically. In case of problems it can be controlled manually via the **Start** and **Stop** icons in the side ribbon. The current status of the Bokeh server can be checked by viewing the **Bokeh server log file**.

When the processing log file shows “Finished batch processing”, you can move to the Apply tab.

The following section describing training of a Keras model is given for completion's sake.

Property Prediction from Seismic

Extract Train Apply

Input Deep Learning Example Data Porosity from AI 4x4x8 Select ...

Training Type New Resume Transfer

Output Deep Learning Model Select ...

Training Parameters Advanced

Type
dGB LeNet regressor

Validation Percentage Split: 0.20

Batch Size
32

Epochs: 15

Early Stopping: 10

Initial Learning Rate (1e): -4

Epoch drop (%): 33.33

Size: 96.77 KB

Decimate input

Train on GPU

Reset Back

LOG
↻
✖
▶
LOG

✓ Proceed ✖ Close ?

Training Parameters Advanced

Use Mixed Precision

Enable Tensorboard

Clear Tensorboard log files

Reset Back

LOG
↻
✖
▶
LOG

The dGB LeNet regressor is a fairly standard Convolutional Neural Network that is based on the well-known LeNet architecture (and available in Keras-TensorFlow) It should be noted that satisfactory seismic-log prediction results have not been generated with this model. It is included here to give researchers a starting point for seismic-log prediction with Keras / TensorFlow models. We encourage workers to use this model as a starting point, and perhaps find a better one (that we will gladly add to the list of supported models).

The following training parameters can be set:

Batch Size: this is the number of examples that are passed through the network after which the model weights are updated. This value should be set as high as possible to increase the representativeness of the samples on which the gradient is computed, but low enough to have all these samples fit within the memory of the training device (much smaller for the GPU than the CPU). If we run out of memory (raises a Python OutOfMemory exception), lower the batch size!

Note that if the model upscales the samples by a factor 1000 for instance on any layer of the model, the memory requirements will be upscaled too. Hence a typical 3D Unet model of size 128x128x128 will consume up to 8GB of (CPU or GPU) RAM.

Epochs: this is the number of update cycles through the entire training set. The number of epochs to use depends on the complexity of the problem. Relatively simple CNN networks may converge in 3 epochs. More complex networks may need 30 epochs, or even hundreds of epochs. Note, that training can be done in steps. Saved networks can be trained further when you toggle **Resume**.

Early Stopping: this parameter controls early stopping when the model does not change anymore. Increase the value to avoid early stopping.

Initial Learning rate: this parameter controls how fast the weights are updated. Too low means the network may not train; too high means the network may overshoot and not find the global minimum.

Epoch drop: controls how the learning rate decays over time.

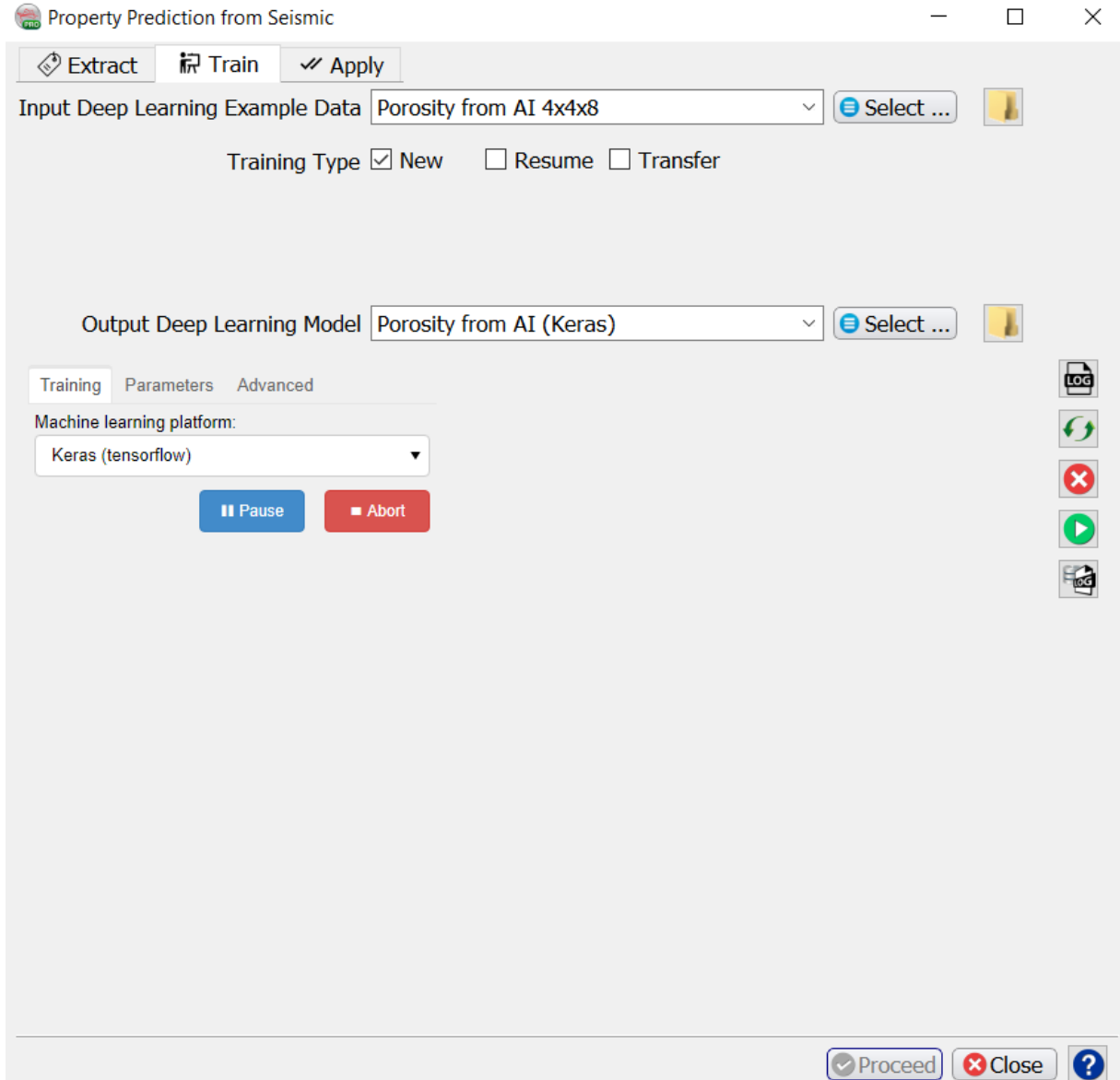
Decimate Input: This parameter is useful when we run into memory problems. If we decimate the input the program will divide the training examples in chunks using random selection. The training is then run over chunks per epoch meaning the model will eventually have seen all samples once within one epoch, but only no more than one chunk of samples will be loaded in RAM while the training is performed.

Mixed precision: Mixed precision option has been added to dgbpy for both PyTorch and Keras platform. This allows to train models using a combination of 16-bit and 32-bit precision formats which will optimize GPU memory usage and efficiency. The GPU memory usage can be reduced by 50% and 75% in some cases. It also increases training speed 2 - 3 times on GPUs with Volta or Turing architectures. This doesn't also affect training accuracy or performance.

For more details about what is 'Mixed precision training' please refer to:

- [Keras documentation: Mixed precision](#)
- [NVIDIA documentation: Train with Mixed Precision](#)

After model selection, return to the **Training tab**, specify the Output Deep Learning model and press the green **Run** button. This starts the model training. In this tab, the **Run** button is now replaced by **Pause** and **Abort** buttons.



The progress of Keras / TensorFlow runs can be monitored using **TensorBoard**, which automatically starts up in your default browser. Please note that it may take a few minutes before TensorBoard has gathered enough information to present training progress charts. It is recommended to clear the TensorBoard log files from time to time using **Clear TensorBoard log files** as all TensorBoard information is retrieved for all current and historic runs.

Property Prediction from Seismic

Extract Train Apply

Input Deep Learning Example Data Porosity from AI 4x4x8 Select ...

Training Type New Resume Transfer

Output Deep Learning Model Porosity from AI (Keras) Select ...

Training Parameters Advanced

Machine learning platform:
Keras (tensorflow)

Run

Training Successful

LOG

LOG

LOG

LOG

Proceed Close ?

File

```
loss: inf
mae: 2160.607177734375
val_loss: inf
val_mae: 2159.146728515625
Time: 00:00
Saving model.
Model saved.
```

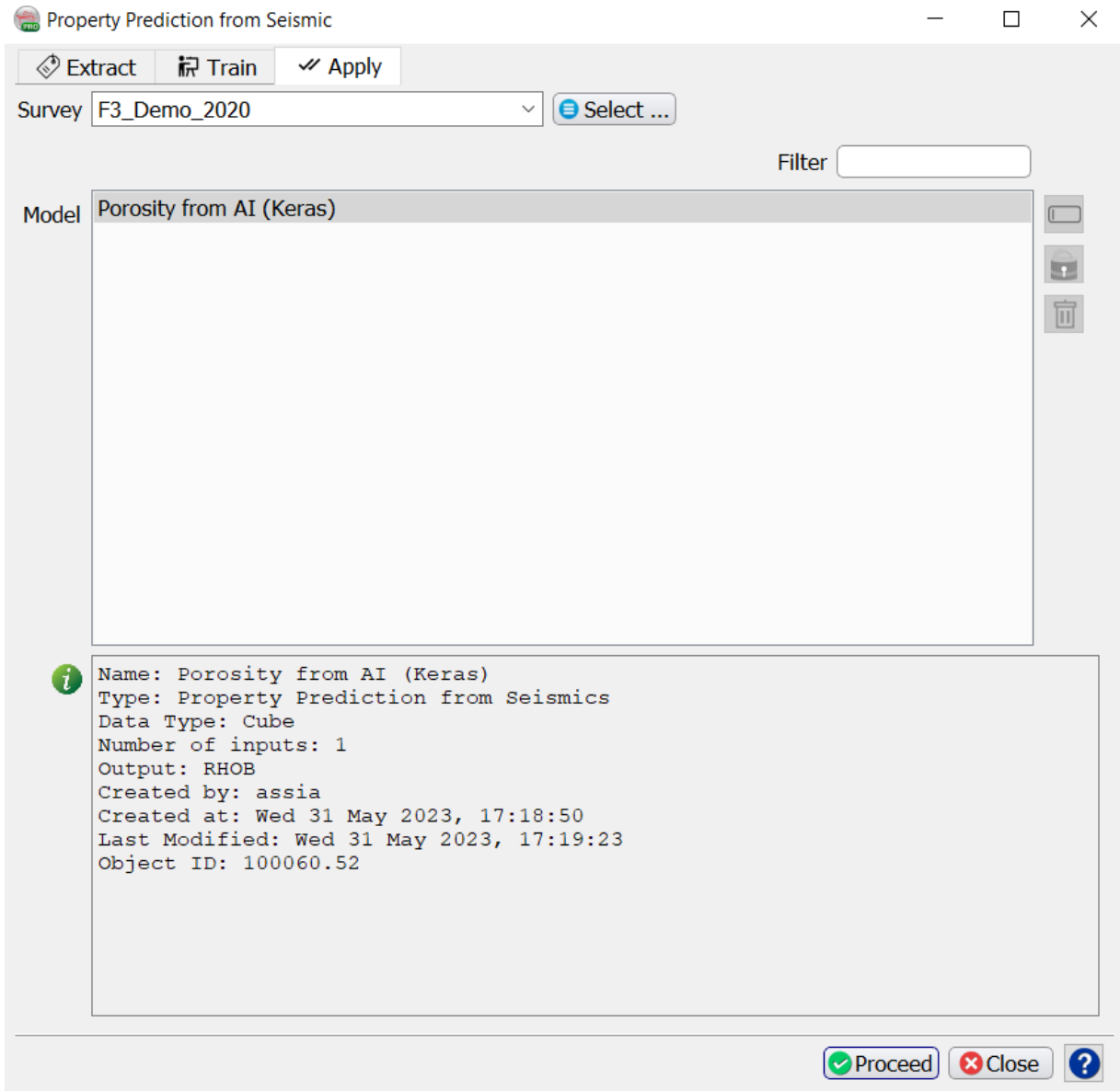
```
Process: 'Machine Learning Training'
Finished: Wed 31 May 2023, 17:19:23
Finished batch processing.
```

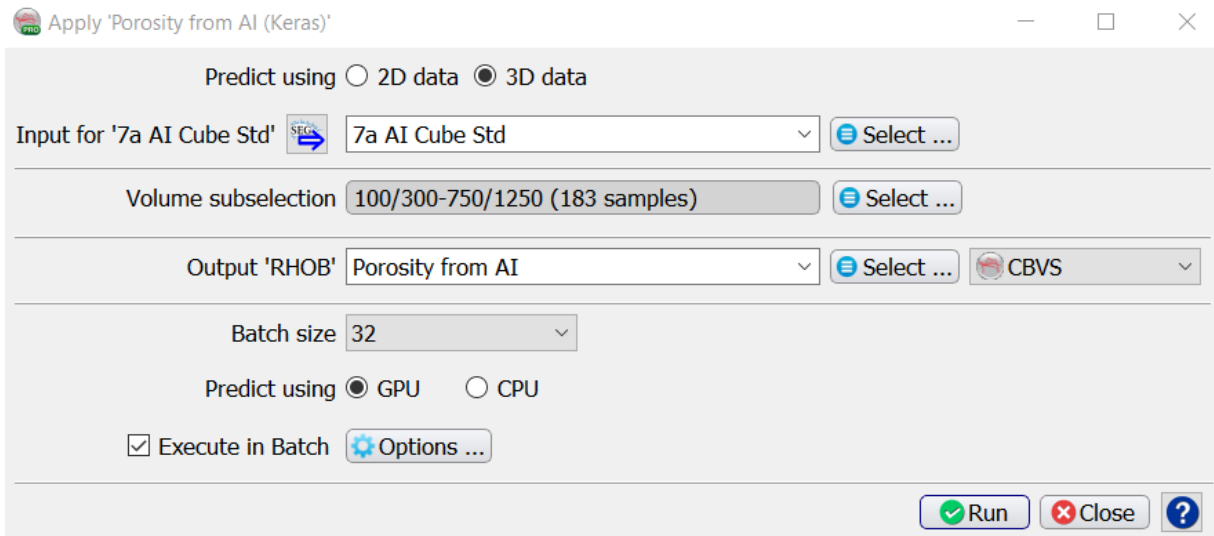
✓ Close

↻ Reload

Apply

In the “Property Prediction from Seismic” window, select the trained model and press **Proceed**.





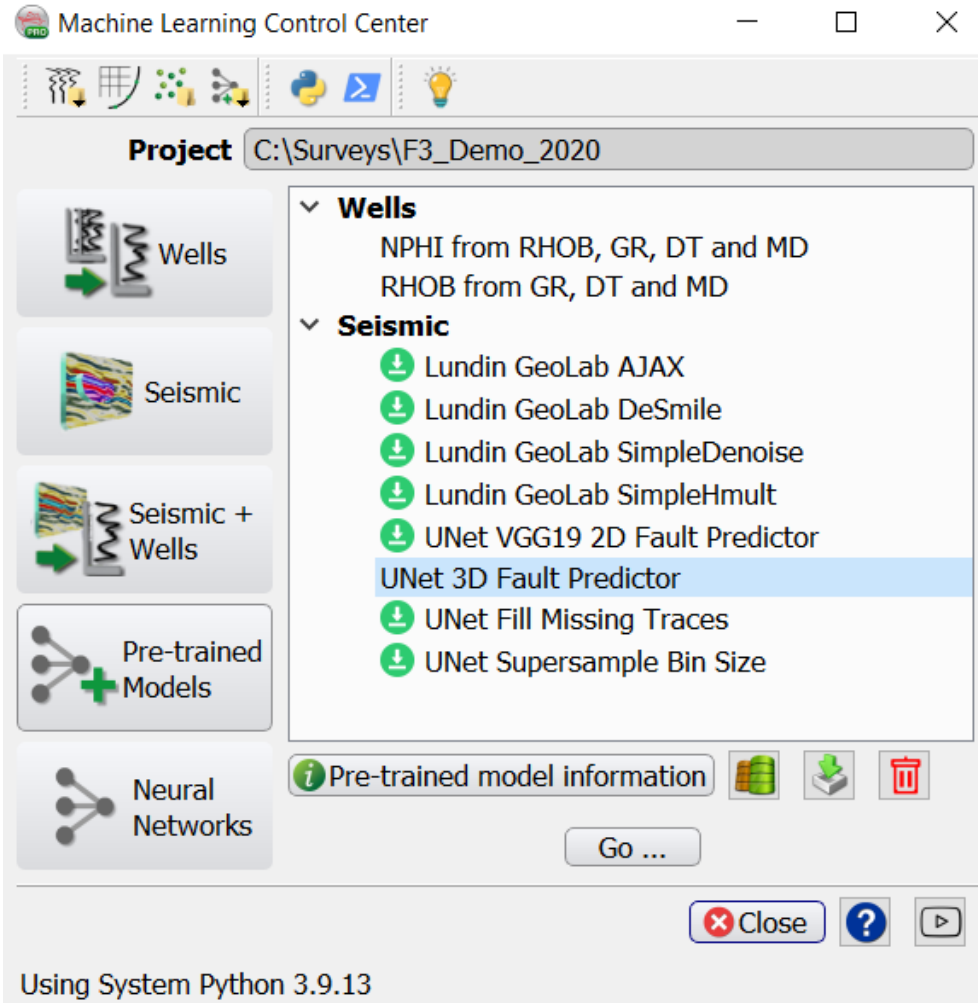
Select the **Input Cube**, optionally specify a **Volume subselection** and give an output name for the seismic **Output** volume. Press **Run**. This will start a batch job. Progress can be monitored in the log file that is automatically launched.

3.6 Pre-trained Models






A geophysics library of trained Machine Learning models is released to the users of OpendTect Machine Learning plugin. These pre-trained models can be reused AS IS to solve a wide range of generic problems on unseen data.

Applying such models is easy with no need to train the data, and valuable results can be obtained much faster than through alternative workflows involving reprocessing or expert knowledge.

We will show which type of models are available. As well as examples of seismic and well log models that are applied to blind test data sets.

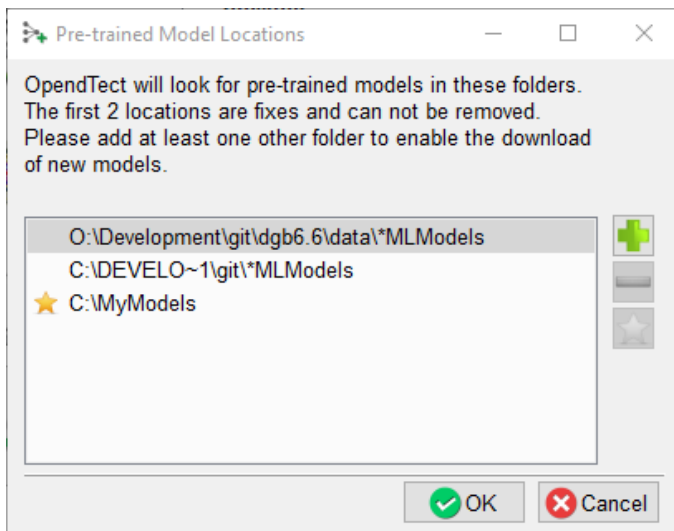


Tools have been developed (in Keras / TensorFlow, PyTorch and Scikit Learn) for importing models into OpendTect's Machine Learning solution.

-  Download status*
-  Model information
-  Pre-trained model disk locations
-  Import models
-  Delete models

Researchers & Experimental geoscientists can use these tools to create a proprietary library of trained models that can be shared with operational geoscientists inside their own company.

Because of the size of Machine Learning models, the trained models are stored in our library in the cloud.



★ Default download location (can be an on-premise shared network location)

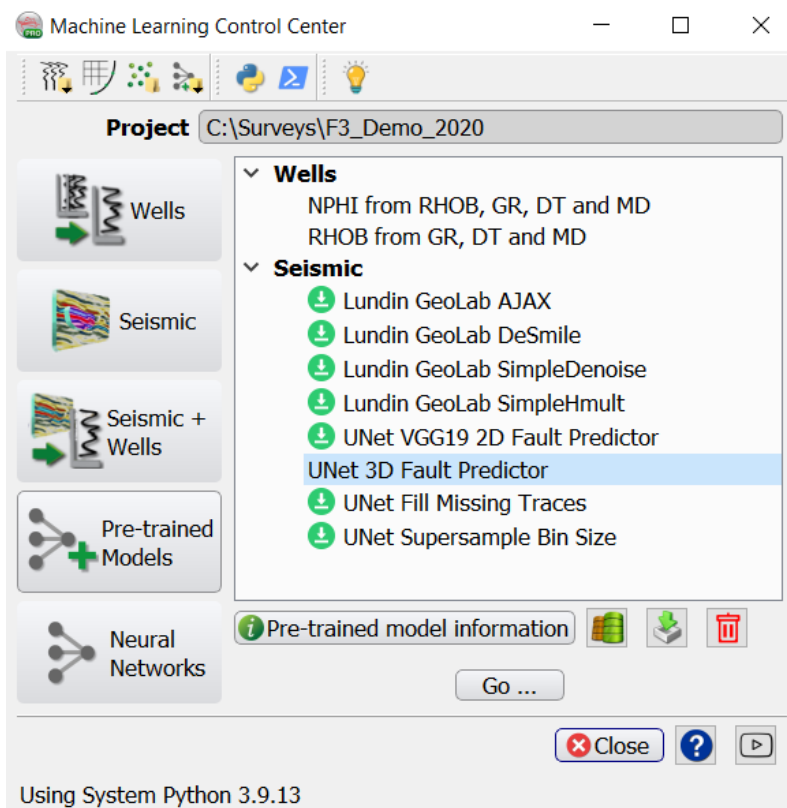
Users have access to meta information such as: model objective, required inputs, generated outputs, model shape, and what examples the model was trained on. Based on this information, users can download models of interest and apply these to their own data. Trained models in the current library originate from dGB and from researchers of Lundin GeoLab (nowadays AkerBP) who granted permission to distribute their seismic models to our users.

Note: All models are 3D Unets trained on synthetic data with shape 128x128x128.

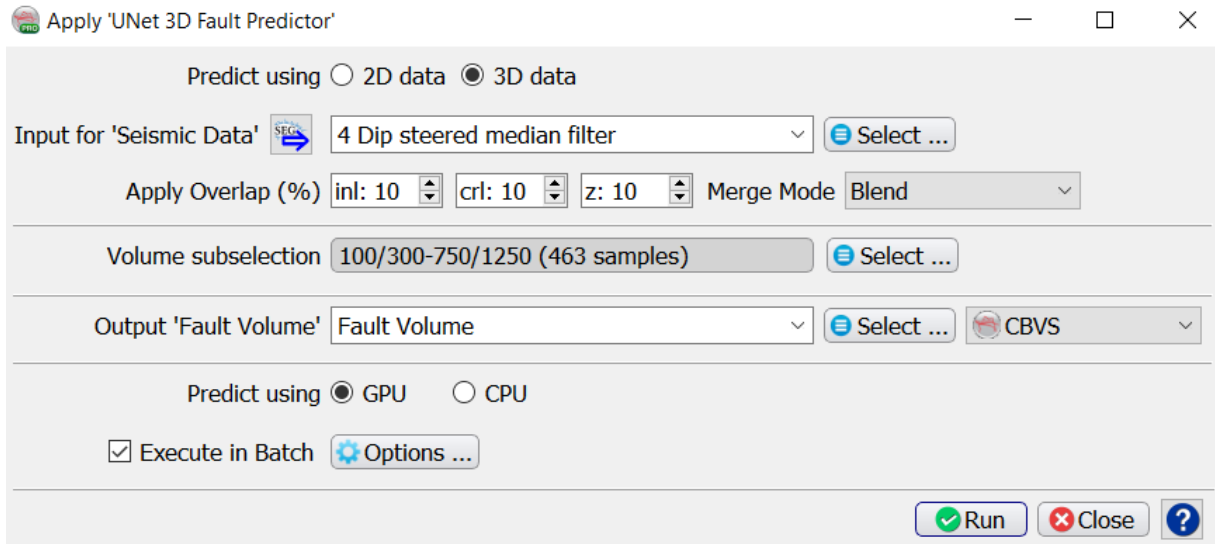
Seismic - UNet 3D Fault Predictor

Predict Fault Likelihood from 3D seismic.

The Unet 3D Fault Predictor, is a powerful and super fast tool to predict faults and fractures in seismic data. This model was trained on synthetic seismic cubelets of 128x128x128 samples.



To use this pre-trained model, select the **Unet 3D Fault Predictor** and press **Go**.



The Apply window pops up. Select the **Input (Seismic Data)** volume. The **Overlap %** is the percentage of overlap in Inline, Crossline and Z ranges of the images that are passed through the model. Optionally you can apply the trained model to a **Volume subselection**. Specify the output name of the **Fault “Probability” Volume**.

Predict using GPU is the default because this is much faster than predicting using CPU. Switch the toggle off if you do not have sufficient GPU memory to use this mode.

We do recommend enhancing the seismic before applying the Unet. Users can choose to remove any undesired noise from their seismic data by either using our dip steering median filter or the edge preserving filter which is part of the fault and fracture plugin.

The input cube is the seismic amplitude volume with 128x128x128 dimension in the inline/crossline/Z directions and the output will be a fault probability volume. Results of the Unet can be thinned in Opendtect by applying the Skeletonization (thinning algorithm).

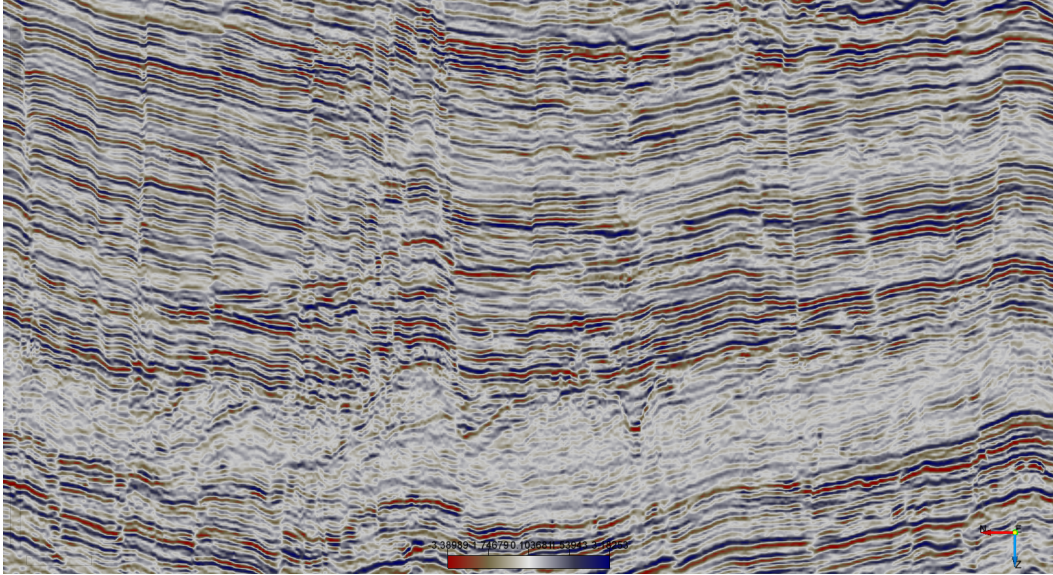


Figure 1. Original seismic (above).

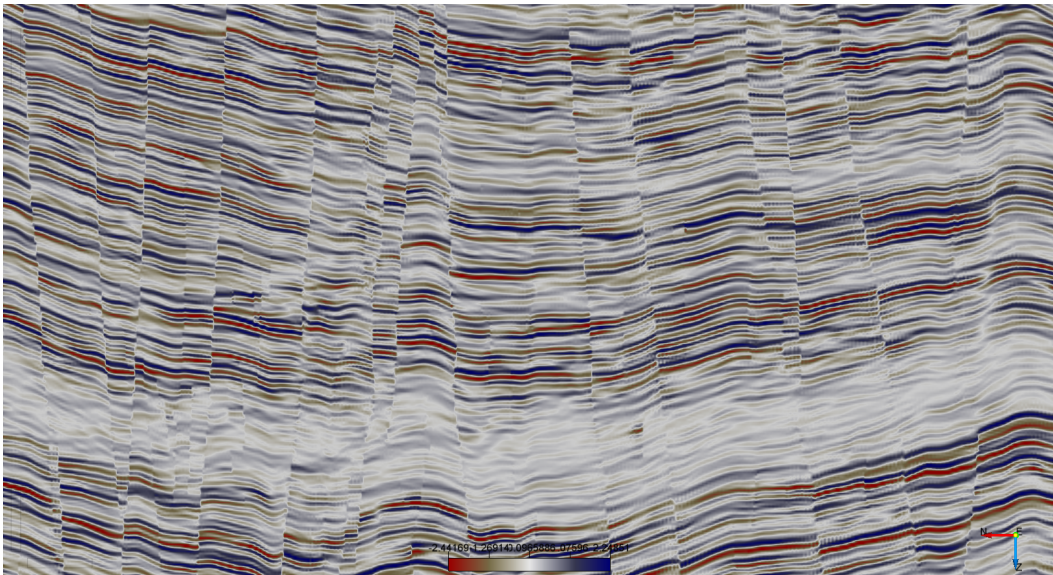


Figure 2. Seismic section after applying Edge Preserving Smoother (EPS).

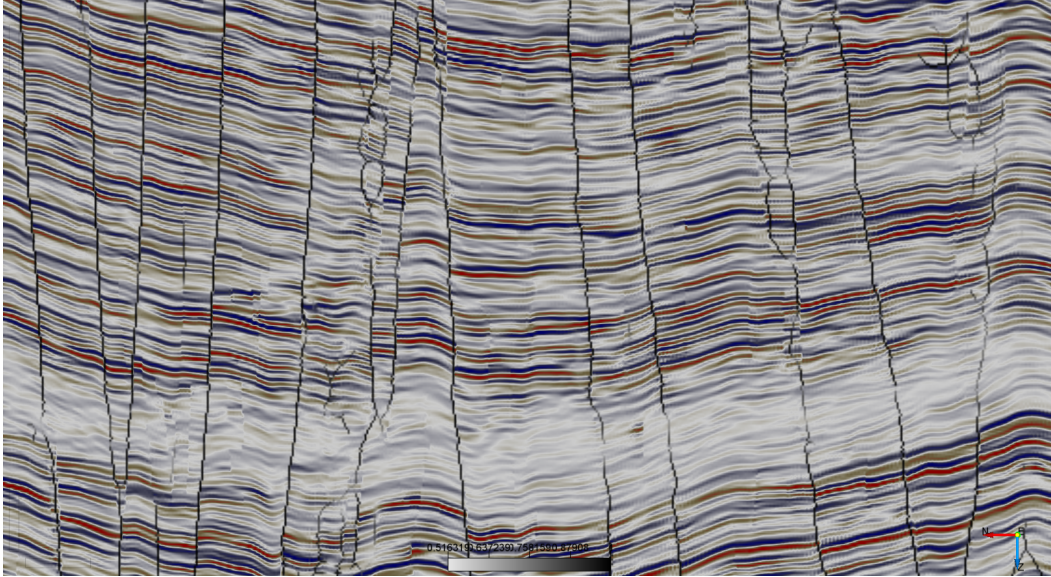


Figure 3. Results of the Unet Fault Predictor after applying the thinning. Faults are clearly imaged on both section and time slice.

For a different angle on the results, the image below is taken from an unthinned Fault Probability cube generated in this way.

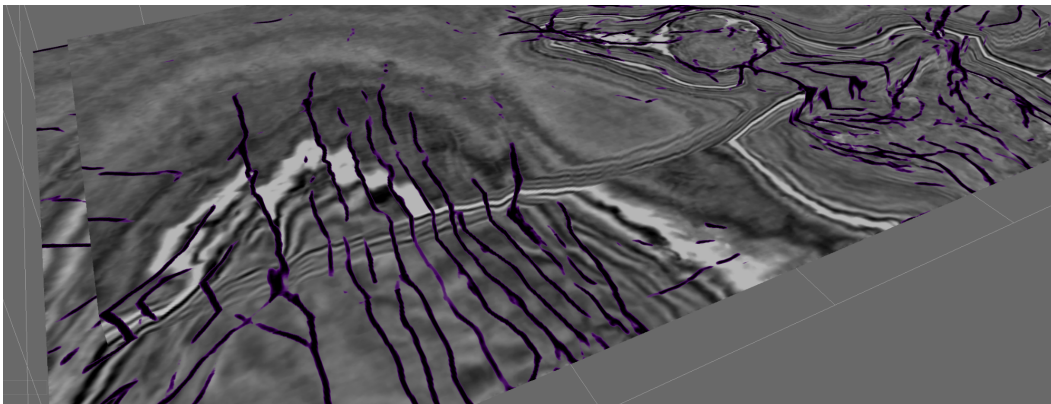


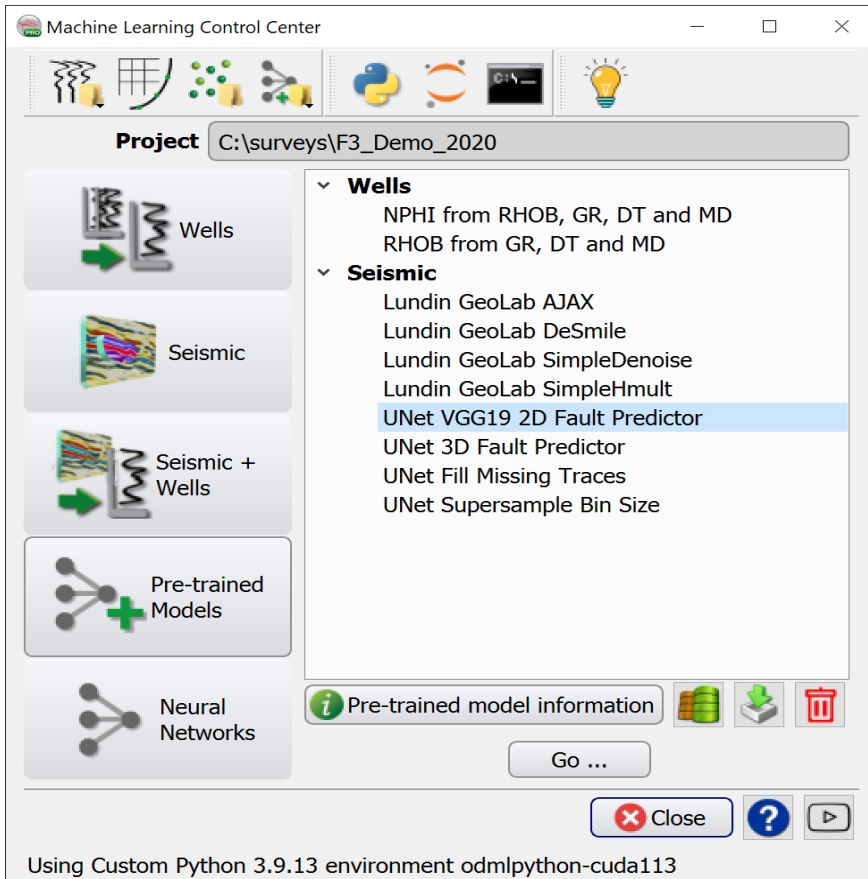
Figure 4: Output of the Unet 3D Fault Predictor on F3 Demo.

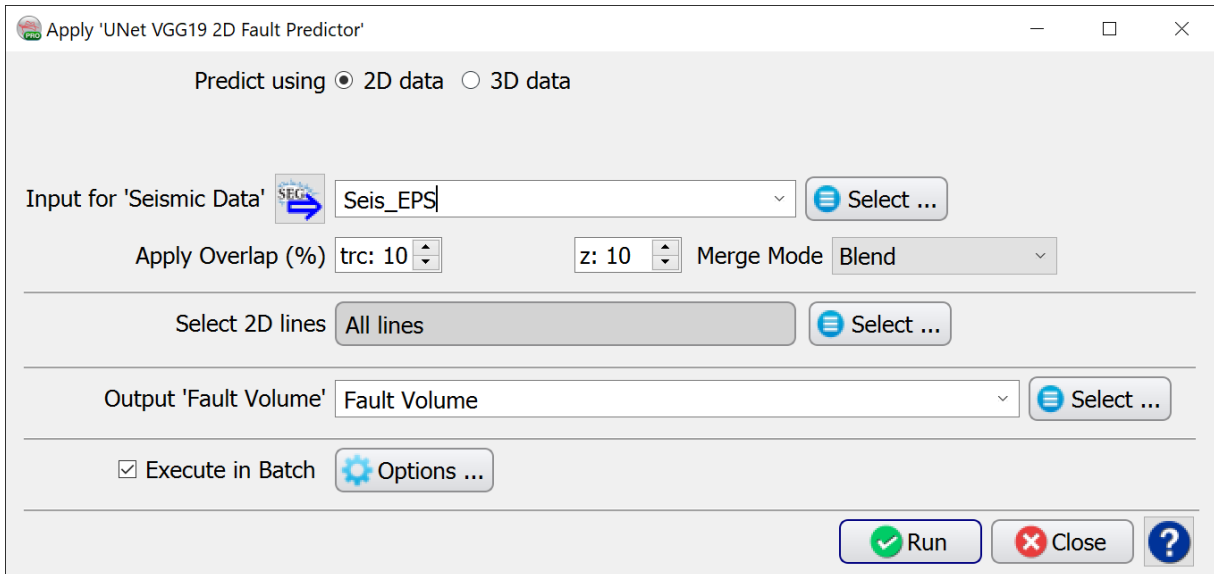
Reference: Wu, X, Liang, L., Shi, Y. and Fomel, S. [2019] FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation. *GEOPHYSICS*, Vol. 84, No. 3 (May-June 2019); P. IM35–IM45, 11 FIGS. 10.1190/GEO2018-0646.1

Seismic - UNet VGG19 2D Fault Predictor

Predict Fault Likelihood from 2D seismic

2D U-Net VGG19 is an auto-encoder model that shares similarities with dGB's 2D U-Net. However, compared to the standard 2D U-Net, 2D U-Net VGG19 offers several advantages. By incorporating the VGG19 backbone, it can capture more complex and abstract features from the input data. This makes it better suited for transfer-training (tuning to a particular data set) than the standard 2D U-Net. Fault Likelihood range 0 to 1. It works best on Edge Preserving Smoothed data.





This model was trained on thousands of non-seismic images. In this model, we utilized the VGG19 model as our transfer learning model. The VGG19 model was the winning model in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Simonyan and Zisserman, 2015]. The VGG19 model have been trained using a large dataset of images called ImageNet. ImageNet is an open dataset that contains a wide range of natural images. It consists of 1000 image categories that are unrelated to seismic faults, such as candles, oysters, airliners, cats, cars, and more [see Russakovsky et al., 2015].

Next, this model was transfer-trained by dGB on synthetic seismic / fault images. Within the implementation of OpendTect's 2D VGG19, we extracted the pre-trained weights from the VGG19 model. These weights, representing the knowledge and patterns learned from ImageNet, were then transferred and applied to our fault predictor model. By leveraging the pre-existing knowledge encoded in the VGG19 weights, we accelerated the training process and enhanced the fault detection capabilities of our model. To facilitate the training process specifically for seismic fault prediction, we utilized Wu's synthetic fault seismic dataset [Wu et al., 2019].

This model can also be adjusted to work with real data by further training it (transfer-training) on interpreted sections. This means that we can fine-tune the model using actual seismic data, which improves its ability to detect faults in real seismic data [OpendTect Webinars: [Building and Training Your 2D CNN Model with OpendTect, Applying and Fine-tuning your Trained Model: From U-Net Architecture to Real Seismic Data Interpretation](#)].

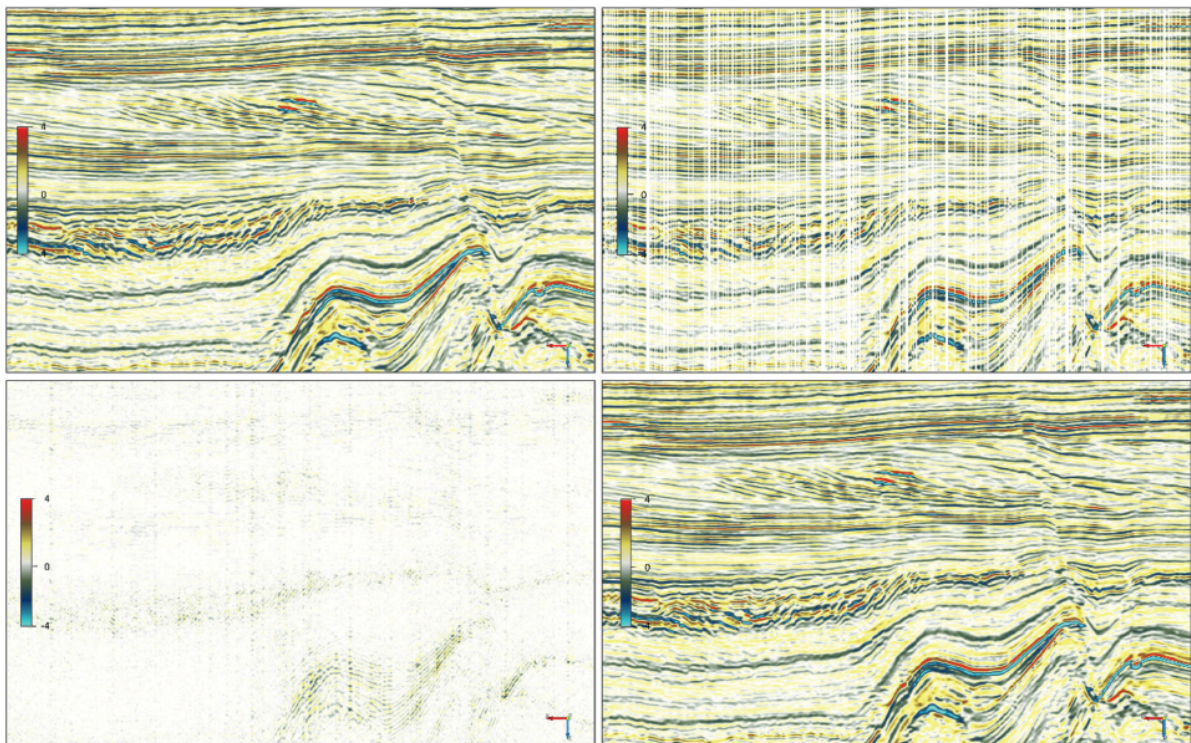
Reference: A. Nawaz, U. Akram, A. A. Salam, A. R. Ali, A. Ur Rehman and J. Zeb, "VGG-UNET for Brain Tumor Segmentation and Ensemble Model for Survival Prediction," 2021 International Conference on Robotics and Automation in Industry (ICRAI), Rawalpindi, Pakistan, 2021, pp. 1-6, doi: 10.1109/ICRAI54018.2021.9651367.

Seismic - UNet Fill Missing Traces

Fill in missing traces in seismic.

Gaps of up to 10 traces can be filled. Output may need to be scaled post-application. Model has proved to work on an offshore data set with different geology, acquisition and processing located hundreds of kms away. It also worked to infill V-shape acquisition gaps in the shallow section of a nearby dataset.

The workflow to create this model is available as a Machine Learning Exercise from the [Machine Learning Knowledge Base](#).



Example of Seismic crossline from F3 offshore The Netherlands. Clockwise from top-left: Input seismic; seismic with 33% randomly blanked traces; Machine learning

reconstructed section; Difference between ground truth (upper-left) and machine learning reconstructed data (lower-right). All data is displayed after whole trace RMS scaling.

Reference: De Groot, P. and van Hout, M. [2021]. Filling gaps, replacing bad data zones and super-sampling of 3D seismic volumes through Machine Learning. EAGE Annual Conference & Exhibition, Amsterdam, 18-21 Oct. 2021.

Seismic - UNet Supersample Bin Size

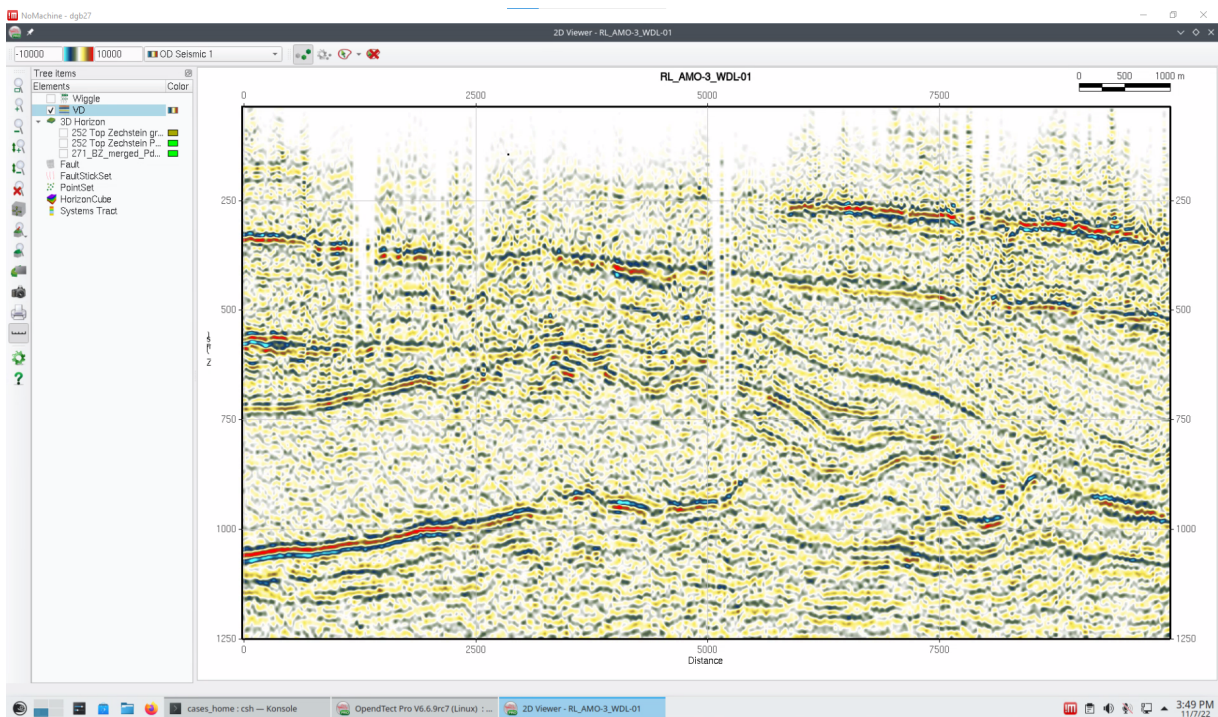
Create a seismic data set with two times the number of traces in the inline and crossline directions.

To apply this model, you first need to create a new survey with an inline, crossline grid that has twice the number of traces as the original survey. You import the seismic data into this survey and use the tools in OpendTect (attribute engine and volume builder) to create a seismic volume with alternating live traces and traces with hard zeros in both inline and crossline direction. The output may need to be scaled post-application. Model has proved to work on an offshore data set with different geology, acquisition and processing located hundreds of kms away.

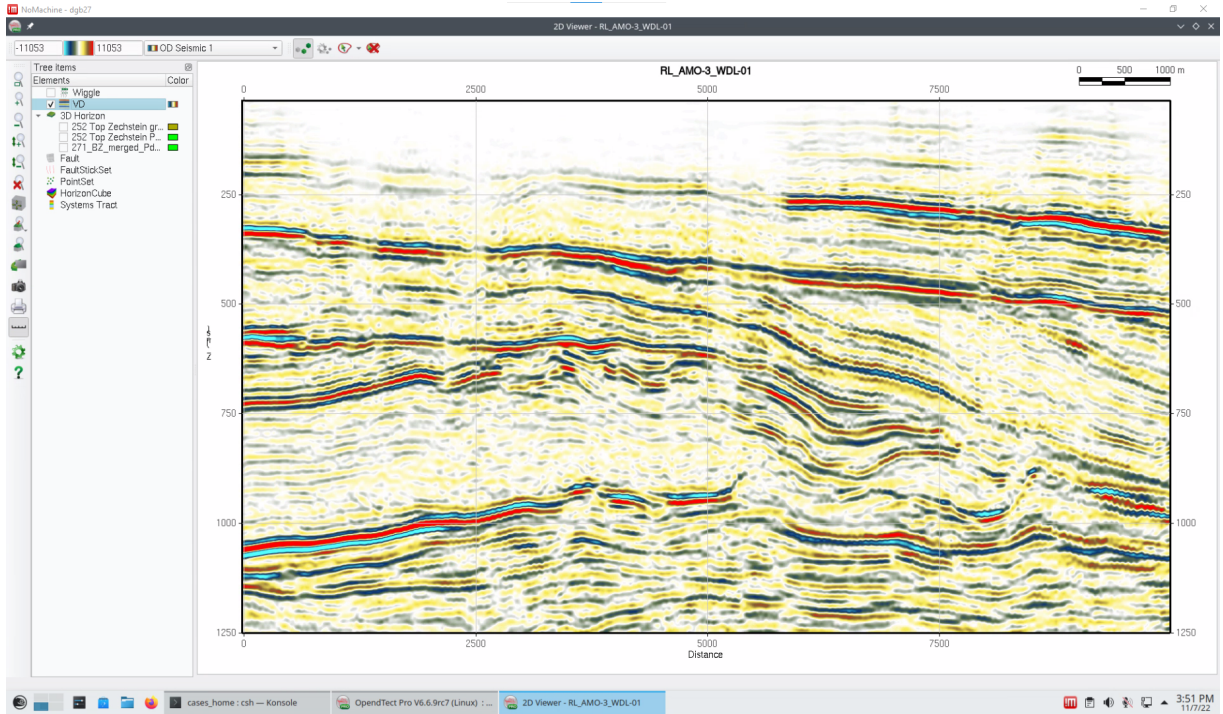
Seismic - Lundin GeoLab AJAX

Attenuate frequency dependent structurally consistent noise, enhance S/N and bandwidth for better interpretability.

AJAX is a model trained to do a series of postprocessing steps in order to enhance the visual quality and interpretability of a seismic volume. This includes (1) Frequency dependent structurally consistent noise attenuation and (2) Bandwidth enhancement. The model does not preserve amplitudes and frequency content of input volume.



Before – AJAX Frequency dependent structurally consistent noise attenuation and bandwidth enhancement



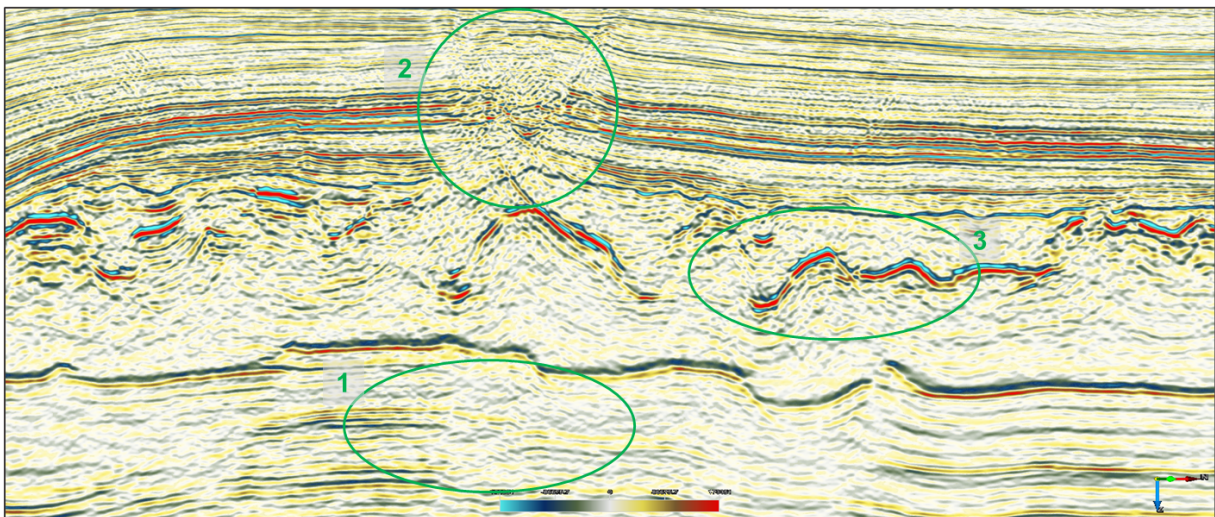
After - AJAX Frequency dependent structurally consistent noise attenuation and band-width enhancement

Seismic - Lundin GeoLab DeSmile

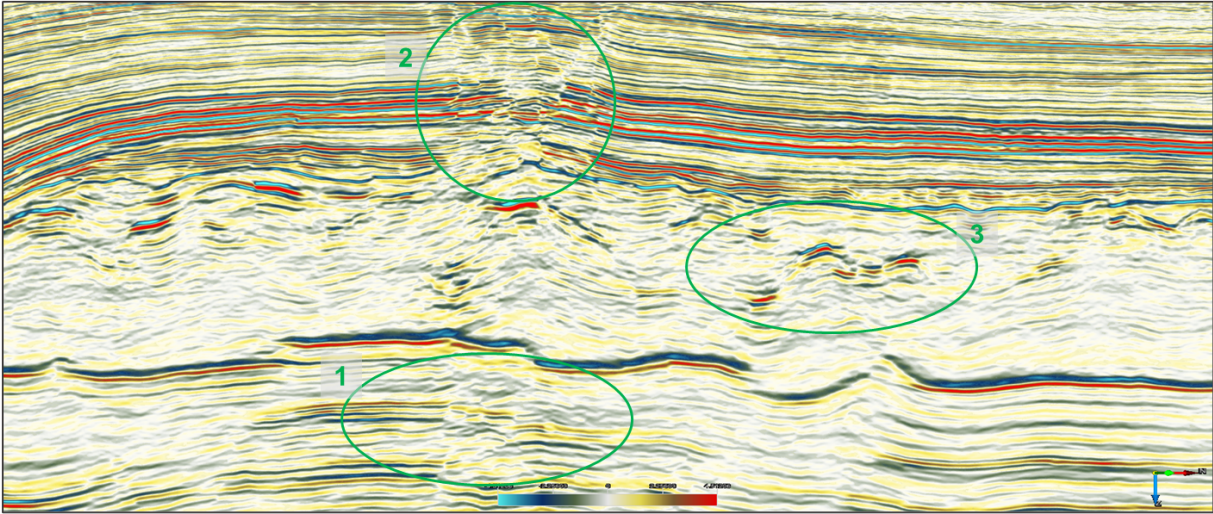
Attenuates migration smiles/artifacts and dip noise.

The migration of seismic data often causes moveout artifacts (smiles) that occur on seismic crosslines in final production volumes due to the sparse sampling in the cross-line direction compared to the inline sampling. This model is trained to remove these artifacts and other dipping noise.

Suggestion: apply only in focus area where artifacts and dip noise is present to avoid affecting dipping primaries.



Before - DeSmile Attenuates migration smiles



After – DeSmile Attenuates migration smiles

Seismic - Lundin GeoLab SimpleDenoise

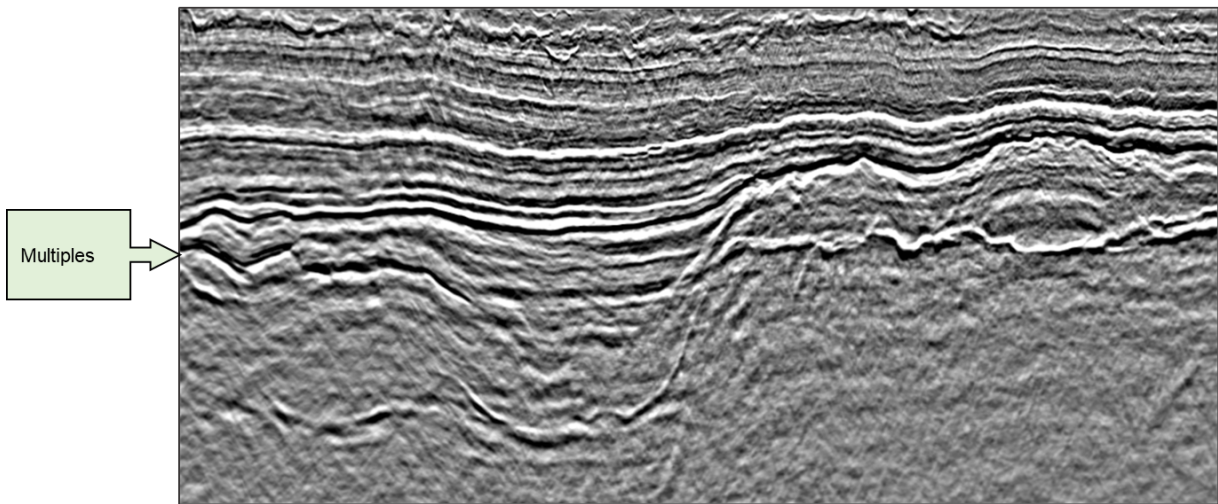
Attenuates random noise and jitter.

This model strictly attenuates noise (“salt & pepper”/jitter). Hence, both amplitudes and frequency of the primaries should be preserved.

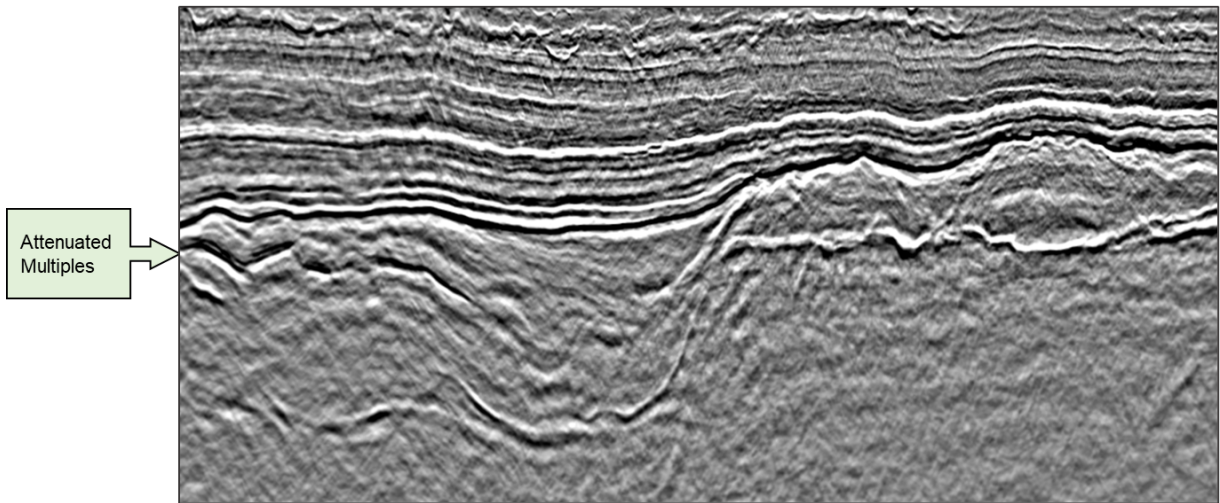
Seismic - Lundin GeoLab SimpleHmult

Attenuates horizon-parallel multiples.

This model removes horizon-parallel multiples (flat multiples). It is to be used to remove either seabed multiples, or other targeted multiples after flattening on target horizon. Should be applied only in window of interest to avoid removing too much primary energy elsewhere. Additional use: a tool to play with in order to understand the originator of the multiples.



Before – SimpleHmult Attenuates horizon-parallel multiples



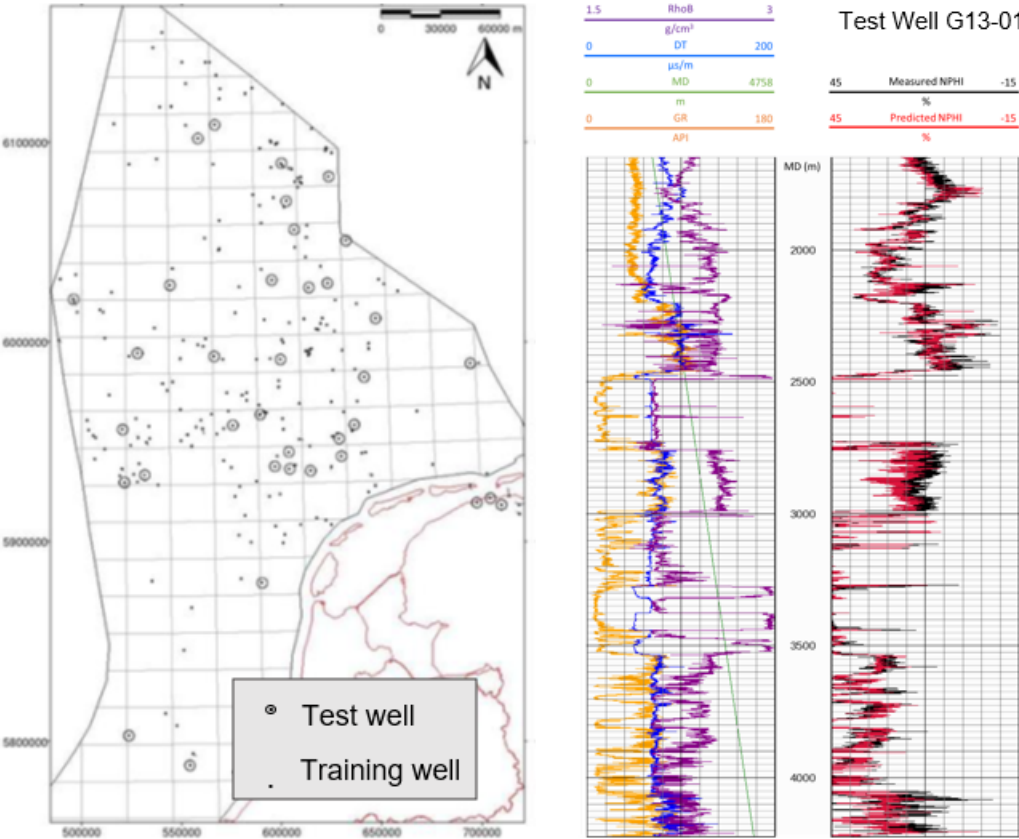
After - SimpleHmult Attenuates horizon-parallel multiples

Wells – NPHI from RHOB, GR, DT, and MD

Predict missing Neutron Porosity log from other input logs.

This example is a log-log prediction. The XGBoost model in this example learned to predict NPHI logs from DT, RhoB, GR and MD curves. The training set consists of 219 wells from the Dutch North Sea.

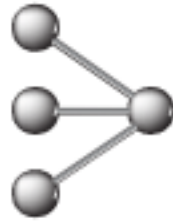
Works best when all input logs are present in the interval of application.



Results of the trained model on well G13-01, a well from the validation set.

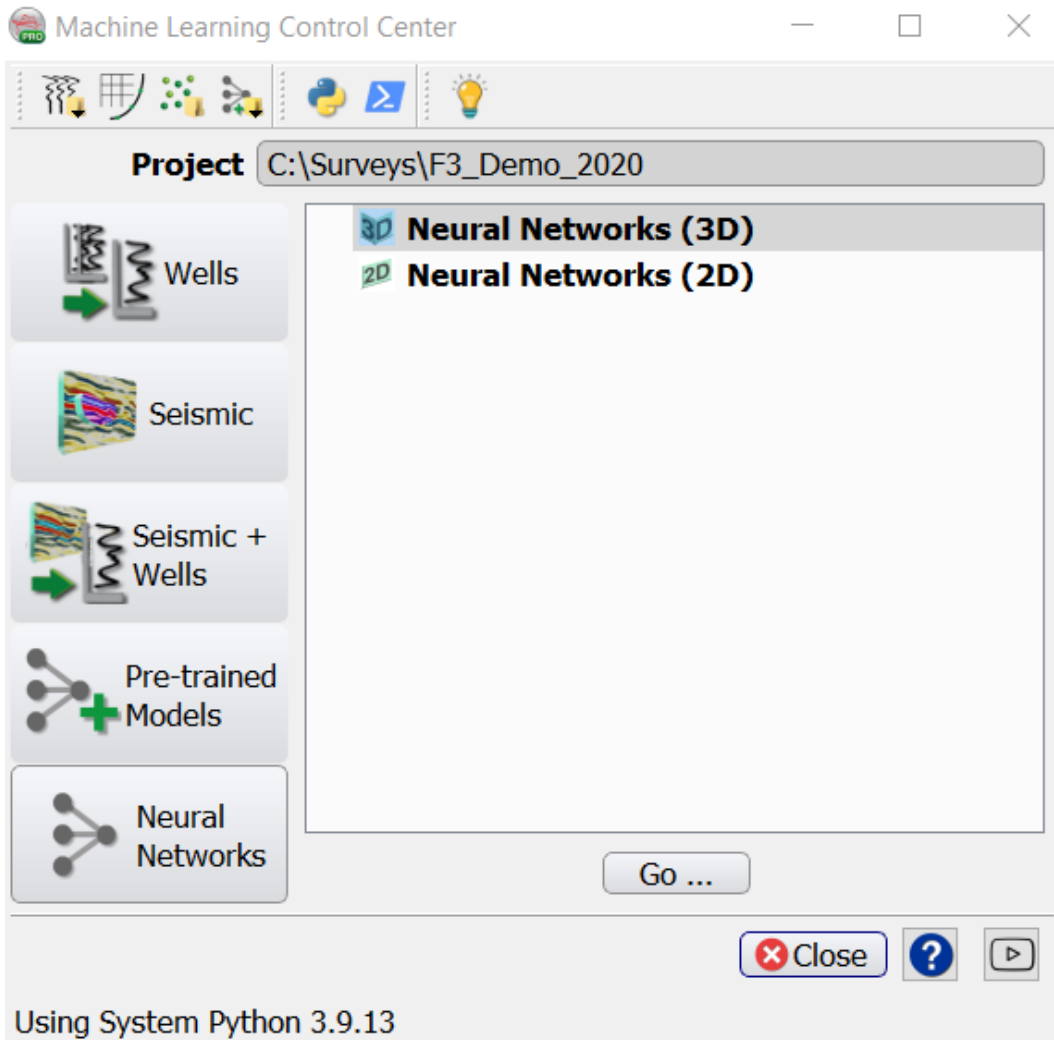
XGBoost prediction of NPHI from GR, DT, RhoB and MD in G13-01, a blind test well. The correlation coefficient between measured NPHI (black) and predicted NPHI (red) is 0.93 for the entire validation set of 36 wells and 0.96 for this well.

3.7 Neural Networks



3.7.1 Introduction

This module in the Machine Learning Control Center launches the Neural Networks plugin, the predecessor of Machine Learning. All workflows of the Neural Networks plugin remain valid and can be found in the following topics in this section.



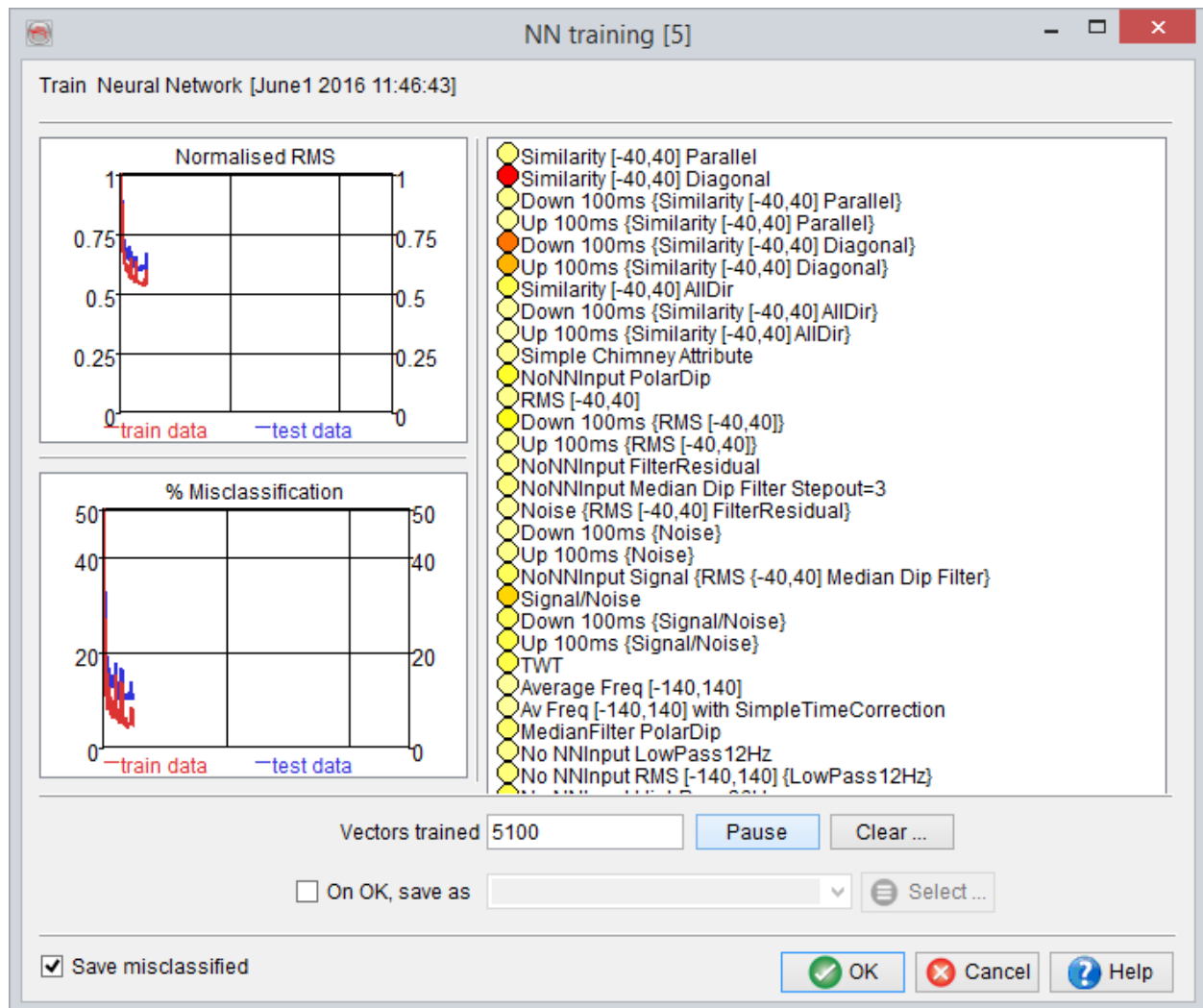
3.7.1.1 Supervised Neural Networks

By supervised training, the user is teaching the network to distinguish between two or more pick sets. At each pick location, a number of attribute values is collected.

As an example, fault detection requires two picksets. The user needs to pick locations where a fault is present (pickset A), and locations where there aren't any faults (pickset B). This whole operation can serve two purposes:

- Efficiency/Automatization. You don't have the time to go through the entire set and pick each and every little fault.
- Find "more than meets the eye". Subtle faults can be present without noticing it but the network picks them up anyway.

It will be clear that the interpreter needs to define attributes that are sensitive to fault characteristics. Therefore, you need to make picksets of positions that contain different objects ("yes, there is a fault here") and counter-examples ('no, there aren't any faults here'). Other examples include *Channel vs. Non-channel* or maybe three choices, e.g. *Channel deposits, Overbank deposits and Shale*. All in all, the very first requirement is that you can pick examples from the data set.



Neural Network Training for the ChimneyCube

During application, the software extracts the same attributes at volumes or horizons and uses the same neural network to predict whether we have a type A or type B situation there. When displaying, you can choose between two network outputs:

- The probability that a certain position is a "type A" location
- The type number (A=1, B=2 etc.).

At some locations, the network is more 'certain' than at other locations. You can get this 'confidence' by looking at the *probability* or at the *Confidence*.

During training of the neural network, you can monitor whether the network can figure out, given the attributes you have defined, whether a location is more like the ones in set A than the ones in set B. If the network would predict that a position would be type A, but in fact it actually is picked by you in pickset B, then that is a *misclassification* of the network. The lower the number of misclassification, the better.

3.7.1.2 Unsupervised Neural Networks

In the *unsupervised approach*, you want the network to come up with a "*natural*" division of the seismic data. This approach is very useful when you want to perform, for example, horizon-based or volume-based segmentation. After training the network and application of the neural network output to an element, the results should be interpreted.

The (single) pickset holds the example positions at which the software calculates the chosen attributes. Therefore, each position in the pickset will yield a vector of values. The result of the extraction of the attributes at each picked location is the *training set*.

The neural network tries to cluster this set of vectors. Similar vectors go into the same *Segment*. This operation can be seen as subdividing the hyperspace of the attribute vectors in compartments. Each compartment has a centre: *the cluster center*.

After the training, the network can be applied to a horizon, time-slice, or volume. That means that the vectors are extracted in a volume or along a horizon. The network can then classify all those vectors into a *Segment*. A vector can be close to the cluster centre or further away from it, which is indicated by the *Match*. The closer to the cluster center, the higher the match.

3.7.2 Manage Neural Networks

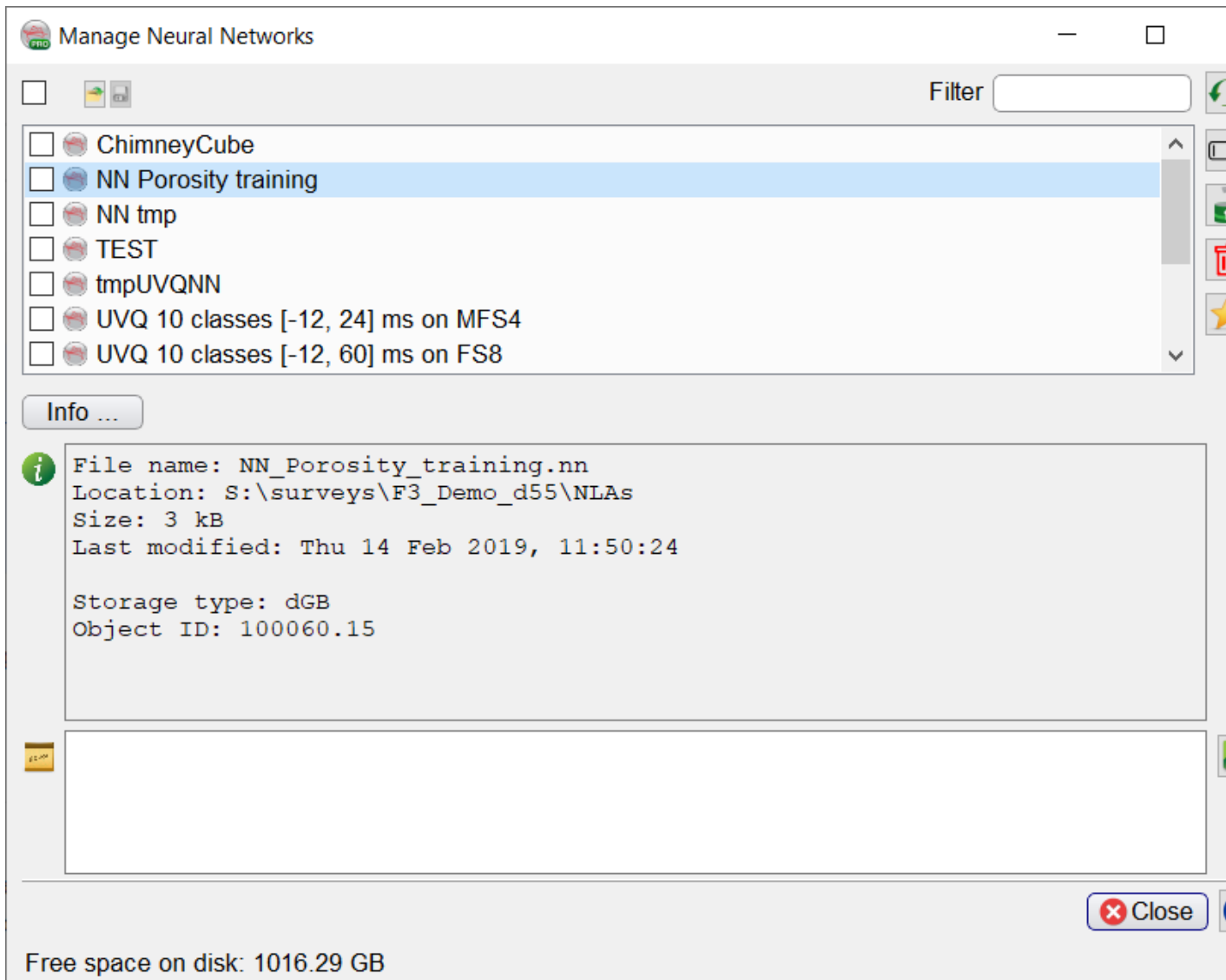
Neural networks can be managed from two locations:

- From [Survey > Manage > Machine Learning > Neural Networks](#): Delete, rename, set the default neural network.
- From [Analysis > Machine Learning > Neural Networks](#): Create a new neural network, load an existing before applying it on tree elements.

3.7.2.1 Data Manager


The neural network management allows to manage them, i.e.: Change disk location, rename, remove, copy, etc.

This window will contain the list of stored neural networks as show below:



For each neural network, you can press the [Info](#) button to get information on how it was trained.

3.7.2.2 Neural Network Management

When clicking the  icon, the *Machine Learning Control Center* window appears. Under the Neural Network tab, the Neural Network management allows to work with

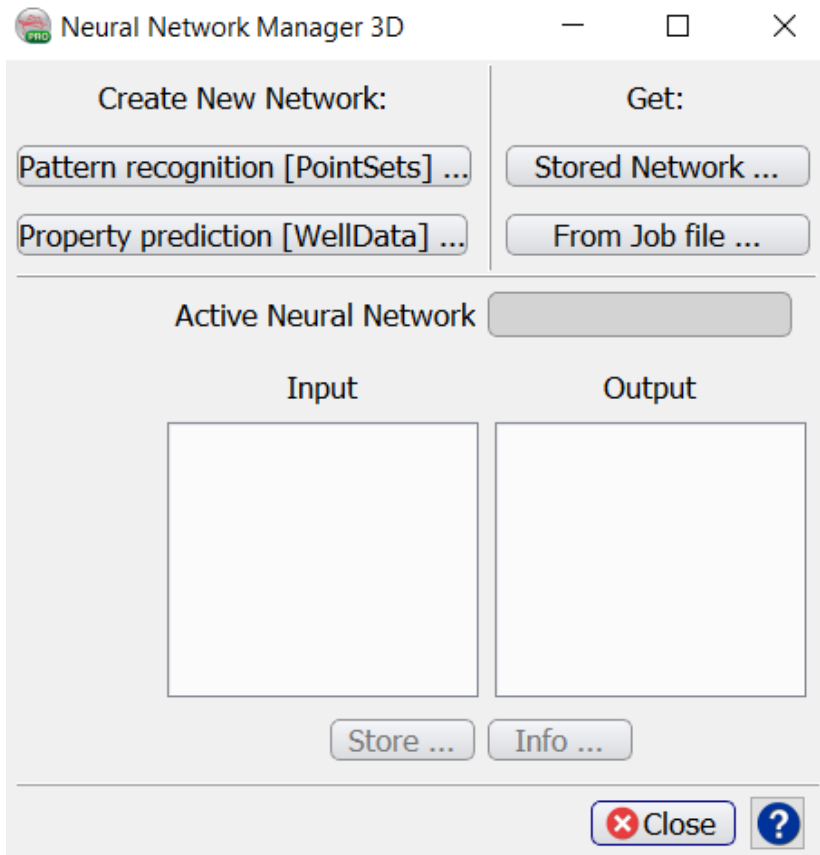
neural networks in the current session: Train a new network, save the current network, load a stored network before applying it to a tree item.

To create a new network, one must first choose the appropriate mode:

- Pattern recognition: To cluster your dataset(s) in several groups (classes), whose distribution is either unknown (unsupervised mode: 1 pickset), or already interpreted (supervised mode, at least 2 picksets).
- Property prediction: To generate a volume representing the prediction of a given well log, extracted by relating it to seismic attributes.

When a neural network is loaded (clicking Stored Network), the input and output nodes of this network are shown in the lists.

When the training of a neural network is finished, the user can store the trained neural network or look at the relevant statistics with the [Info](#) button.



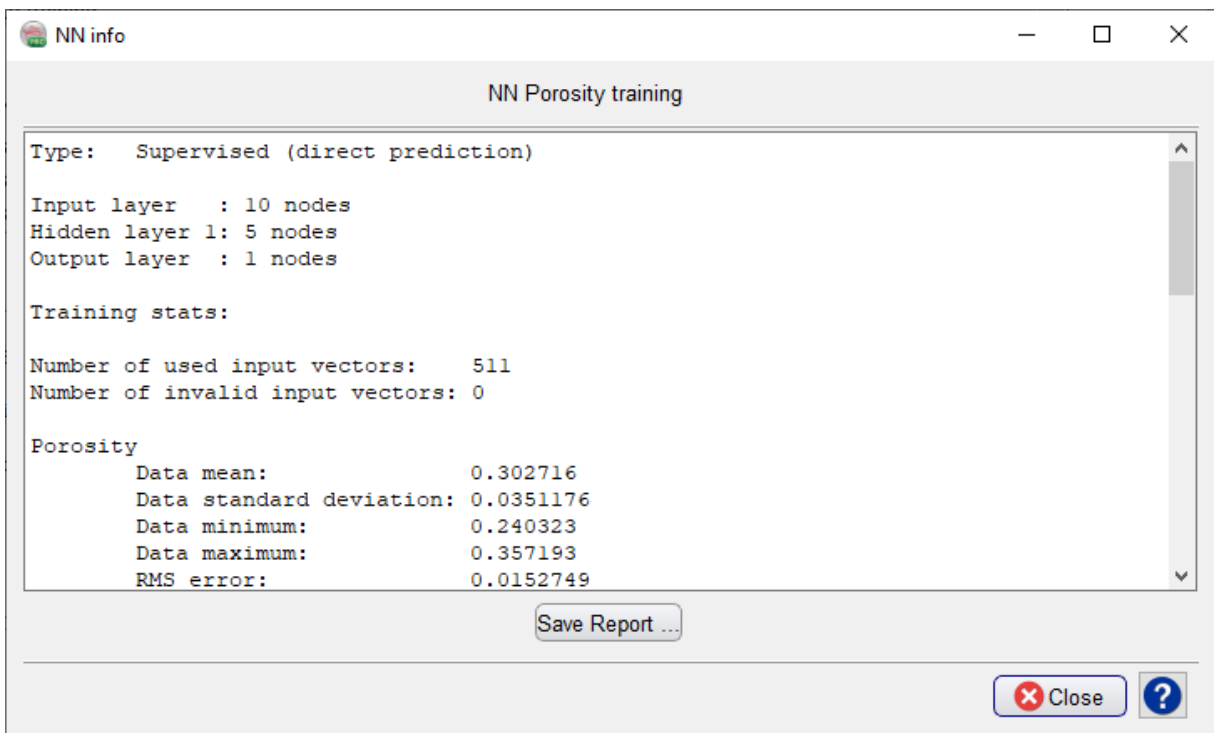
3.7.2.3 Neural Network Information

After training of a neural network, the information window shows some information on the network topology, the attributes that were used, and the relative importance (weight)--on a scale from 0 to 100--of the individual attributes. Be careful to judge these weights. This information tells a lot about individual attributes, but only a little about relationships between attributes. This information can be stored into a report (a simple text file) by pressing the *Save Report* button. Press *Dismiss* to go back to the Neural Network Manager.

3.7.2.3.1 Supervised Neural Network Information

The information window shows neural network information for the training set and the test set. For each set the numbers of usable and invalid vectors are shown.

[Balancing](#) is automatically performed between the data extraction and the training. This ensures that there is an equal representation of all inputs picksets. Misclassification information is displayed based on the membership to the input pickset.



3.7.2.3.2 Unsupervised Neural Network Information

The information window for an unsupervised neural network will show the number of vectors assigned to each class and their average match to the corresponding class center.

The last table will display the attribute values of each class center:

The screenshot shows a window titled "NN info" with the subtitle "Volume-based segmentation between FS7 and FS6". The window displays the following information:

```

Type:  Unsupervised (segmentation/clustering)

Input layer   : 4 nodes
Hidden layer 1: 6 nodes
Output layer  : 2 nodes

Vectors classified: 993
Class 1 vectors: 77.   Average match: 0.91276
Class 2 vectors: 98.   Average match: 0.920992
Class 3 vectors: 222.  Average match: 0.928109
Class 4 vectors: 116.  Average match: 0.908803
Class 5 vectors: 197.  Average match: 0.910349
Class 6 vectors: 283.  Average match: 0.935054
    
```

Below this text is a table showing the attribute values for each of the six class centers:

	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6
Dip-steered Inline dip	1.60994	2.44379	1.3387032	0.80299	-13.719986	-4.671539
Dip-steered Crossline dip	-9.8074188	-5.576252	-8.30827	-10.9425964	-27.3888073	-18.0190868
Dip-steered inst. amp.	4723.621...	2970.94751	2255.556...	1298.405...	1140.298...	1010.932...
Dip-steered inst. env. weight. freq.	212.217804	164.94931	222.6360...	120.135521	187.2240...	240.910553

At the bottom of the window, there is a "Save Report ..." button, a "Close" button with a red X icon, and a help button with a question mark icon.

When the segmentation was performed on a waveform (i.e. all input attributes contain the string "Sample") an extra option *display* will enable the visualization of the data of this table in a 2D viewer in order to see the class centres:

UVQ 10 classes [-12, 24] ms on MFS4

Type: Unsupervised (segmentation/clustering)

Input layer : 10 nodes
 Hidden layer 1: 10 nodes
 Output layer : 2 nodes

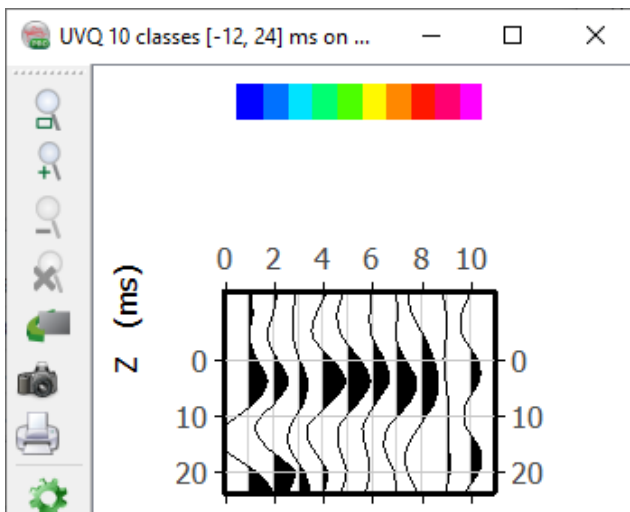
Vectors classified: 1000

Class 1 vectors: 58. Average match: 0.889825
 Class 2 vectors: 66. Average match: 0.906868
 Class 3 vectors: 110. Average match: 0.916101
 Class 4 vectors: 93. Average match: 0.921663
 Class 5 vectors: 153. Average match: 0.9242
 Class 6 vectors: 99. Average match: 0.920937
 Class 7 vectors: 146. Average match: 0.921739
 Class 8 vectors: 71. Average match: 0.907396
 Class 9 vectors: 106. Average match: 0.901485
 Class 10 vectors: 98. Average match: 0.900962

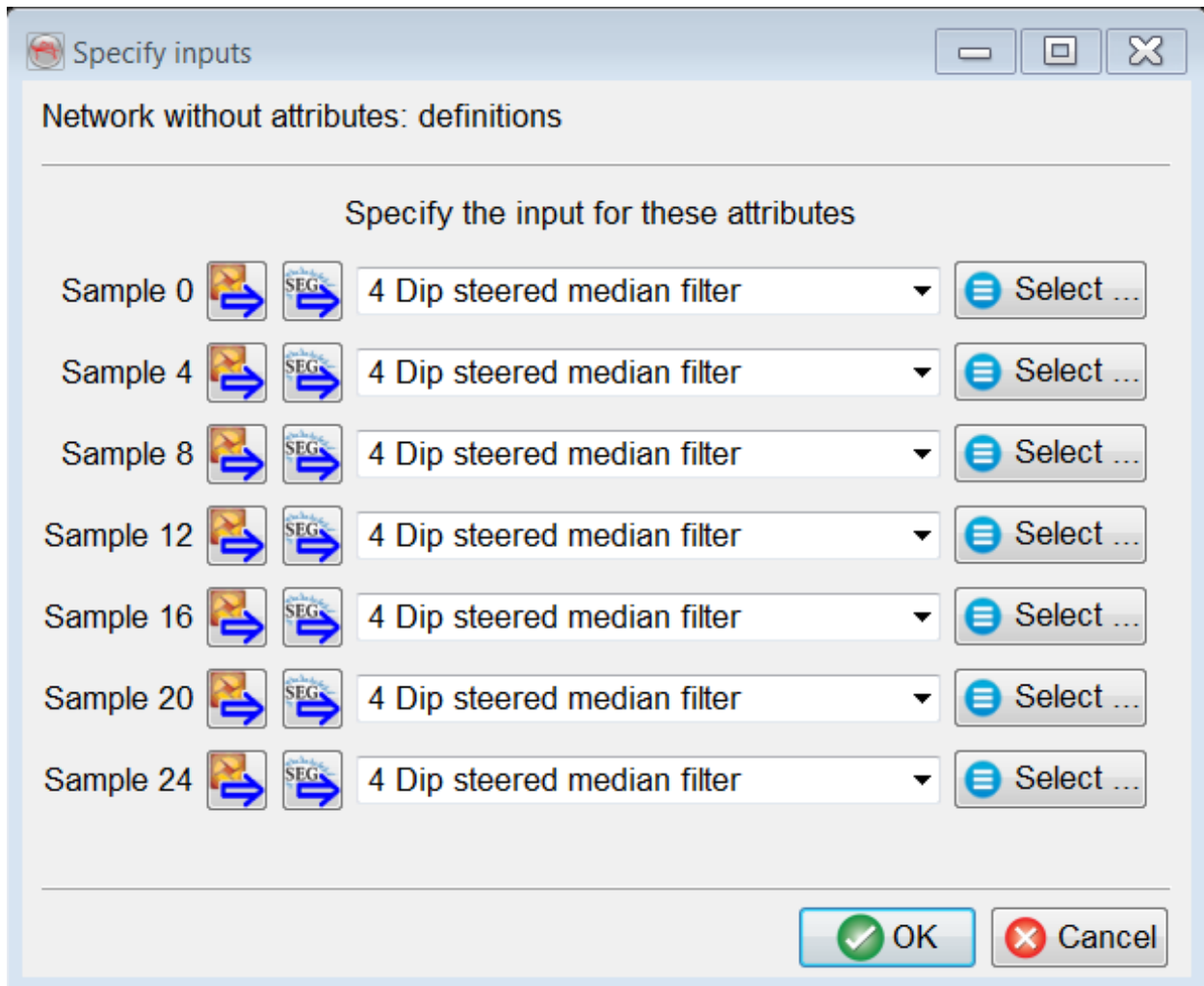
	Center 1	Center 2	Center 3	Center 4	Center 5	Center 6	Center 7	Center 8	Center 9
Sample -12	605.918...	1060.41...	-608.594...	1125.15...	-797.277...	93.2113...	-20.4211...	-3192.9...	-728.1...
Sample -8	1075.77...	-23.0787...	-1097.01...	-1532.30...	-2414.40...	-418.102...	438.559...	-2190.8...	198.52...
Sample -4	259.591...	-1468.43...	-1149.06...	-2392.68...	-1193.72...	91.067627	-126.59...	840.207...	164.94...
Sample 0	2067.33...	1083.21...	487.108...	2304.56...	3210.04...	3084.97...	2103.36...	3137.68...	-392.3...

Save Report ... Display ...

Close ?



3.7.2.4 Import GDI Networks Window



When a GDI neural network is used in OpenTect, the input to the network will be examined. For each input, there are two options:

1. OpenTect recognizes the attribute and has an equivalent available. In that case, you'll need to specify from which cube OpenTect should extract this attribute. This is usually (but not necessarily) the cube from which the training set is extracted. This has to be specified in the top section of the window.

2. OpendTect has no equivalent attribute available. You'll have to extract the attribute into a "*Seismic Cube*" in GDI and provide this cube in the bottom section of the window. Note that "*Frequency*" is one of these unmatched attributes (GDI has event-based, OpendTect has FFT-based frequency).

Depending on the availability of the attributes in OpendTect, the applicable part of the window shown here appears.

3.7.3 Pattern Recognition (using Picksets)

The Neural Network plugin supports two types of neural networks based on picks: fully connected Multi-Layer-Perceptrons (MLPs) and Unsupervised Vector Quantizers (UVQs). MLPs are used in supervised (*with a priori, learn by example*) mode, while UVQs are used in unsupervised experiments (*segmentation = clustering*).

Analysis method. The *Supervised* method allows the choice of one or more output nodes. The groups of nodes (*PickSets*) indicate how the neural network should separate the character found in the input attributes. *Unsupervised* separates the nodes in the (single!) pickset based on a clustering in a user defined number of classes. A saved pickset can be used but a random pickset can also be created for this purpose.

Input Training data set. Specify whether the input data set must be extracted on the fly, or retrieved from a stored input set. An input training set consists of a range of attributes (names and values) at given example locations. To create a training set the user must specify which attributes to use and at which locations these attributes must be calculated.

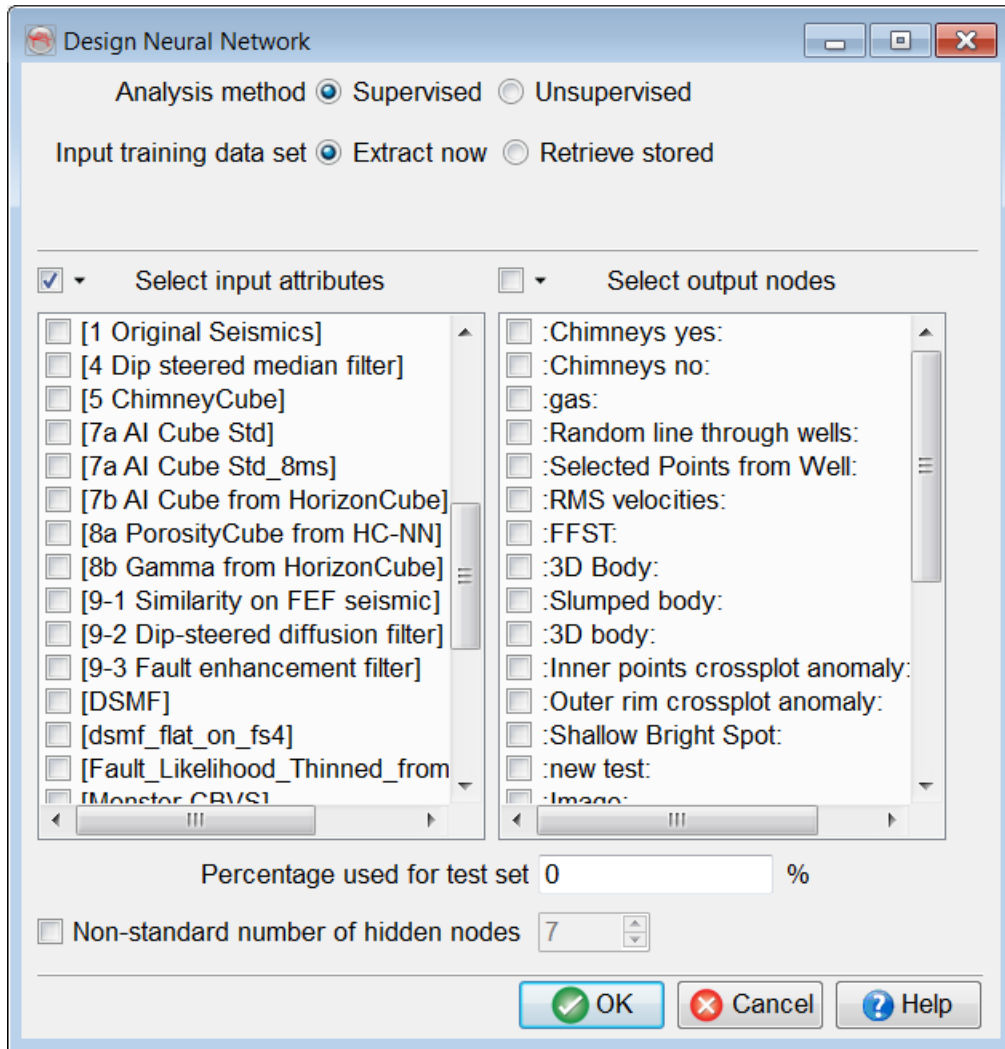
Select input/output attributes. The *Select input attributes* lists all attributes defined in the current attribute set as well as all data that is stored on disk. Select any or all of these to serve as input to the neural network. The *Select output nodes* on the right contains all available pickset groups. Select the pickset group containing the locations at which attributes must be extracted to create a training set. Note that for an object probability cube such as TheChimneyCube the user needs two pickset groups: chimneys and non-chimneys.

Percentage used for test set. In the supervised mode, it is recommended to create a subset for testing the neural network's performance during training, specify a *Percentage* of the pickset to use for the test set. The test set is created by randomly

drawing example locations from the selected picksets. Test set examples will not be used to update the neural network weight set during training, they are merely passed through the network to compare the network's classification with the actual classification.

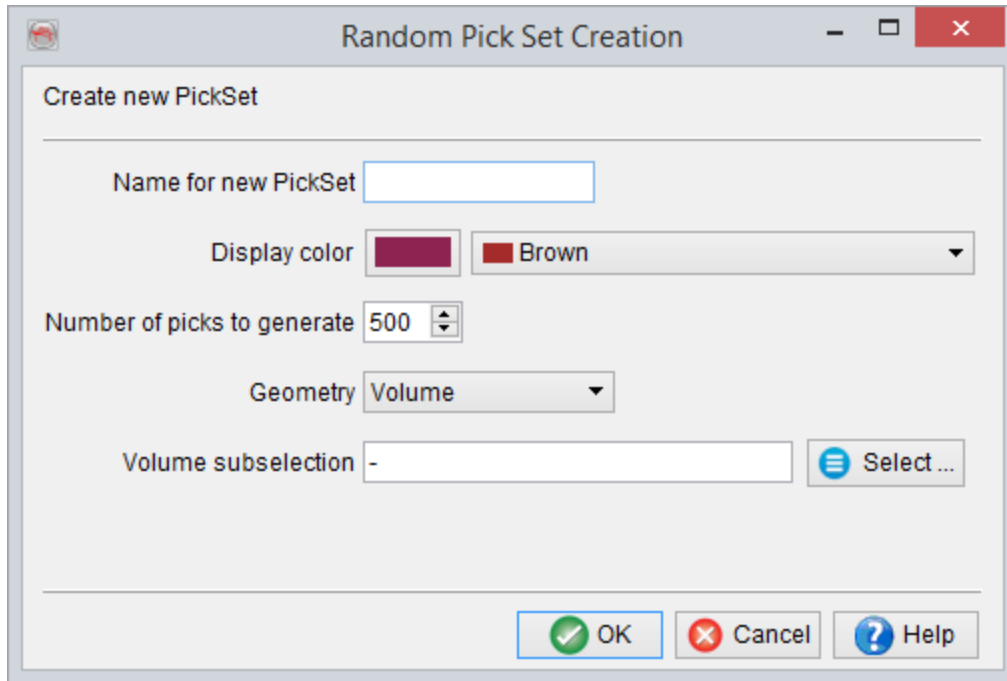
Non-standard number of hidden nodes. If the option is activated, a user can define a non-standard number of the hidden nodes.


Number of Classes. In unsupervised mode, attributes at locations in the specified pickset are clustered (segmented) into the specified *Number of classes*. During the training phase the UVQ network learns to find the cluster centers. At each iteration, when an vector of values has been assigned to a cluster, the cluster center is moved to minimize the (euclidean) distance with the different vectors of attributes values. In the application phase the input attributes are compared to each cluster center. The input is assigned to the winning segment, which is a number from 1 to N, where N is the number of clusters. In addition, the network calculates how close the input is to the cluster center of the winning cluster. This measure of confidence is called a *match*, which can range between 1 (perfect match, i.e. input and cluster center are the same) and 0 (input and cluster center are completely different).

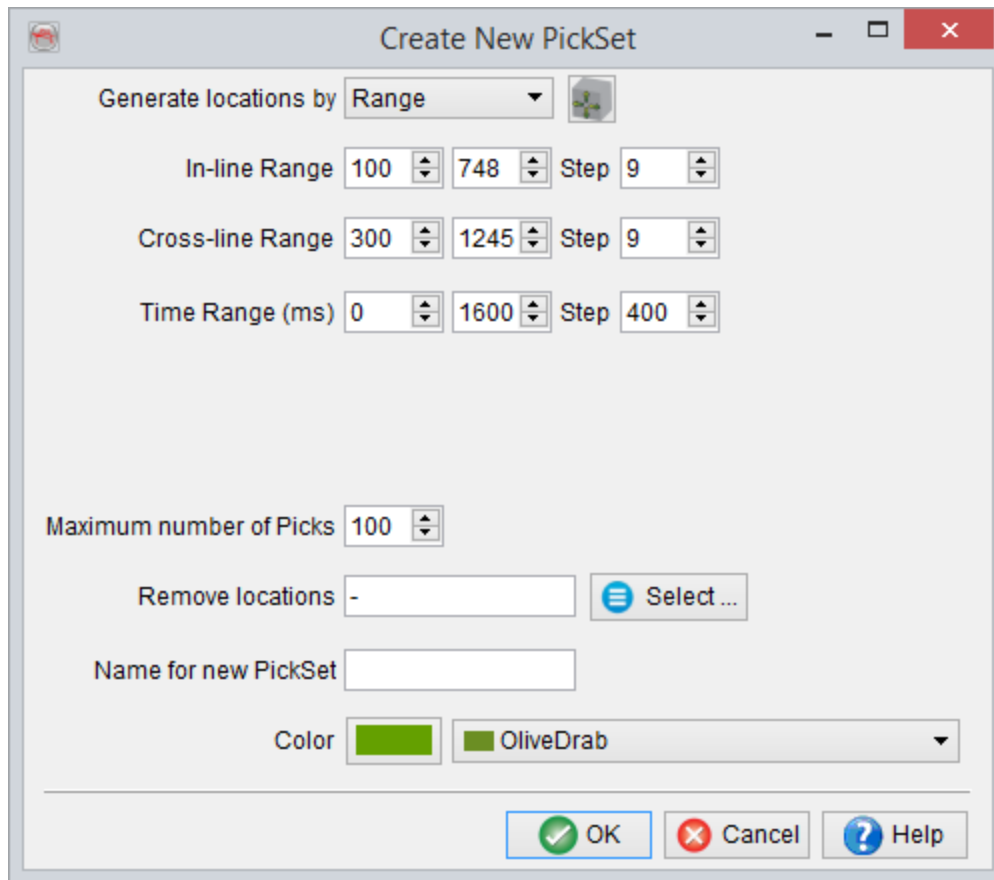


Fast pickset creation

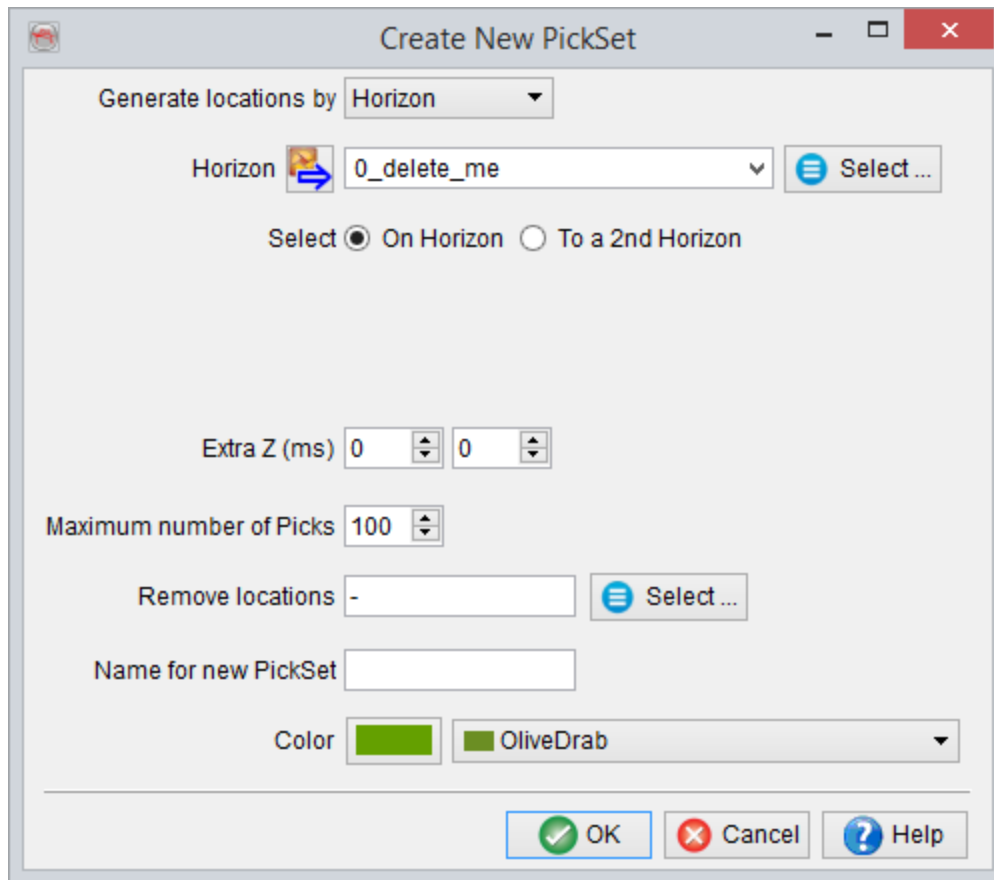
The neural network extraction module can quickly create in unsupervised mode the required input set of training locations by pressing the 'Create' button:



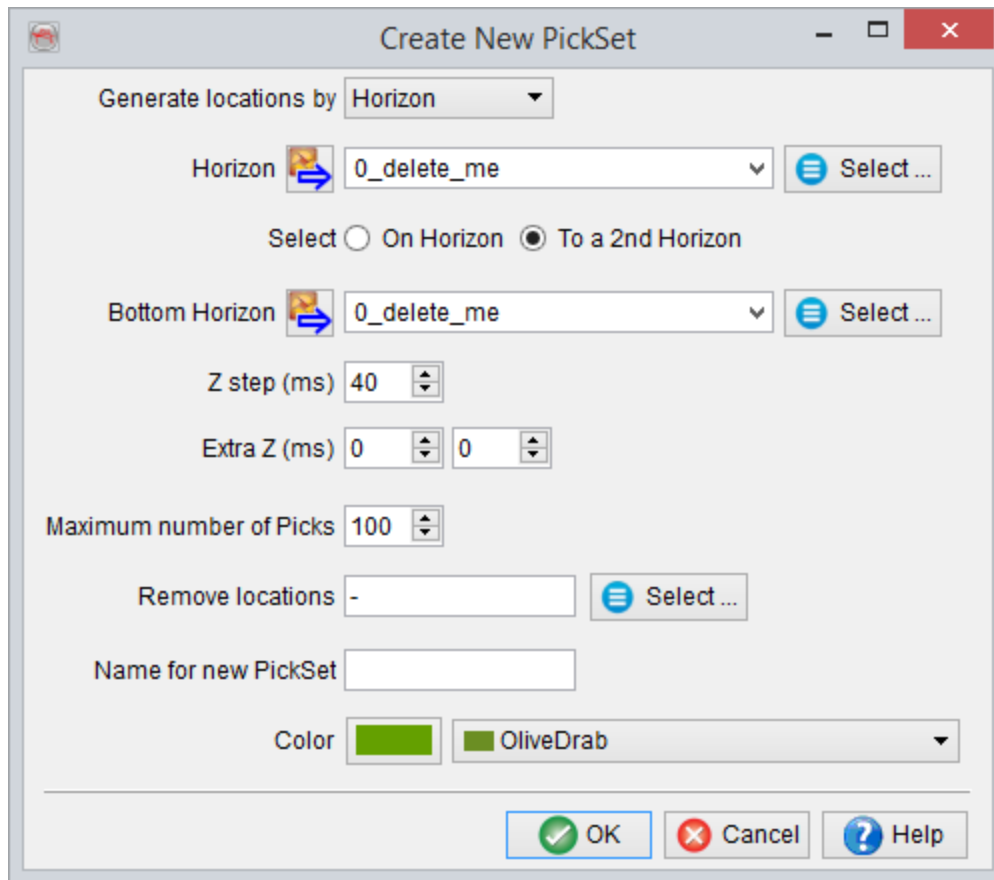
 *More sophisticated 3D pickset creation tools are available from the tree by right-clicking on the element 'Pickset/Polygon'.*



Extraction setting in volume for 3D UVQ




Extraction settings along single horizon for 3D UVQ



Extraction setting between two horizons for 3D UVQ

3.7.4 Property Prediction (using Well Logs)

With this module, it is possible to relate any well log to seismic data. Select an input variable (from cubes and attributes) and, an output log variable (e.g. porosity or Vshale). The neural network will try to look for a relation between the two sets. Provided a working relationship is established, the user can then apply the trained neural network to a larger volume.

 *Neural networks are good interpolators, but not good in extrapolating. If you train a neural network on a certain formation or interval, it is not recommended to apply it outside that formation or interval.*

Since log data is being related directly to seismic data, it is essential that the well logs have a very good tie and are well aligned with the seismic data in the interval of interest. If this is not the case, results may easily become disappointing.

Input training data set: The training data set is the collection of input and target values that the neural network is trained on. Usually, the user will leave this option at *Extract now*. In case you stored an input training data set before, tick then *Retrieve stored* and select the input training data set with a standard file browser.

Select input/target attributes: The input attributes from the active attribute set or any of the stored cubes can be selected in the left screen. The stored cubes appear in square brackets [] at the bottom of the list. On the right, select the output log variable. All available logs from all wells are shown here. If the log of interest has different names for different wells, rename first these logs so that they all have exactly the same name. The log(s) can be renamed in *Survey > Manage > Wells*.

Target contains: Specify if the target contains ordinary well logs value or if a lithology log is available and can be used.

Wells to use: Select the wells on the right. From these wells, the *selected target* log is retrieved, if available.

Extract between: Select the markers to specify the interval the neural network should be trained on. Just as with the target logs, all available markers from the available wells are displayed. You can use the same marker twice, in combination with non-zero distance above/below. It is up to the user to make sure a marker with exactly the same name actually exists in the selected wells. To edit the markers and their names, see *Survey > Manage > Wells*.

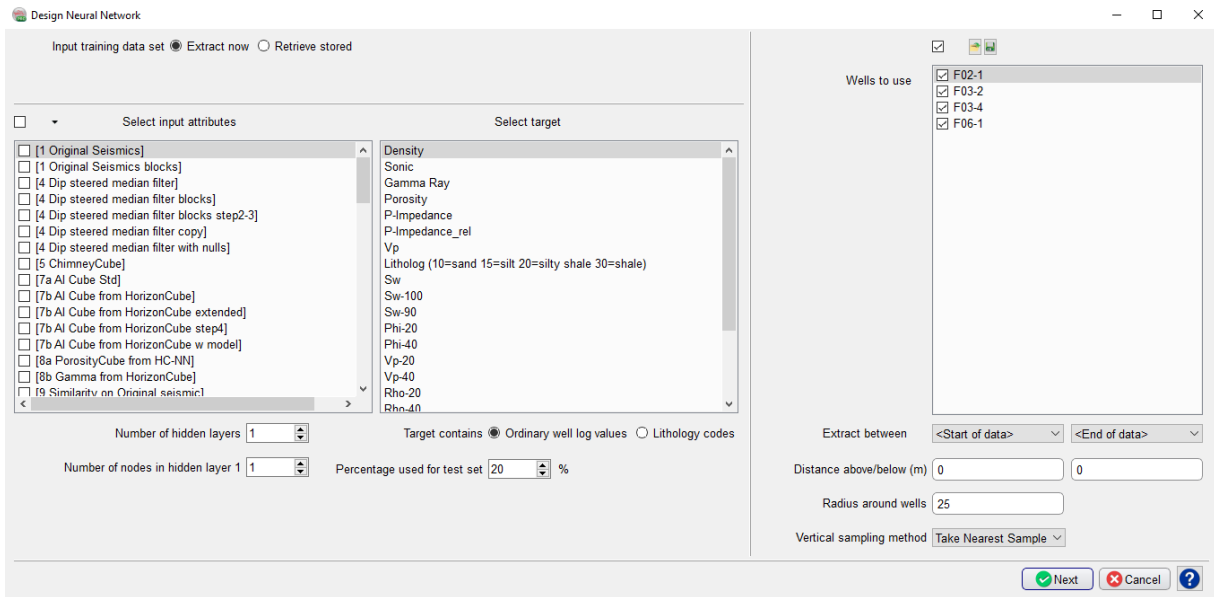
Distance above/below: Indicate the extra distance above and below the start and stop marker respectively that should be taken into account in the training. Negative values are possible; negative above the top marker means start below the top marker. Negative below the bottom marker means stop before the bottom marker.

Radius around wells: Indicates the radius around wells where the selected input attributes are calculated for each depth of extraction. All traces within the radius are selected. For example for nearest trace leave blank or put zero.

Vertical sampling method: Well data have a much higher (vertical) resolution than seismic cubes. This means several log values correspond to a single seismic sample. Averaging can prevent aliasing problems, although this is not necessarily a problem. Therefore, one can use the median or average of the well log values corresponding to the seismic sample location.

When predicting a binary variable (like sand/shale), or a lithology code, the most frequent filter will be necessary. *Nearest sample* selects only the nearest log value. This will be the best option in pre-filtered curves.

On pressing *OK*, the software starts collecting all necessary data. When all data is collected, the training starts and this can be monitored in the [NN training window](#).



3.7.4.1 Balance Data

Balancing is a mandatory pre-processing step for neural network training. The distribution of input vectors is modified in order to get a flat distribution for the output quantity (target log), in the training data set. Balancing is done automatically when training from picksets or on a target log that is containing lithology codes. In both cases, the output values are discrete (integers), thus *Balancing* can be automated safely. An ordinary log will show continuous values between a minimum and maximum.

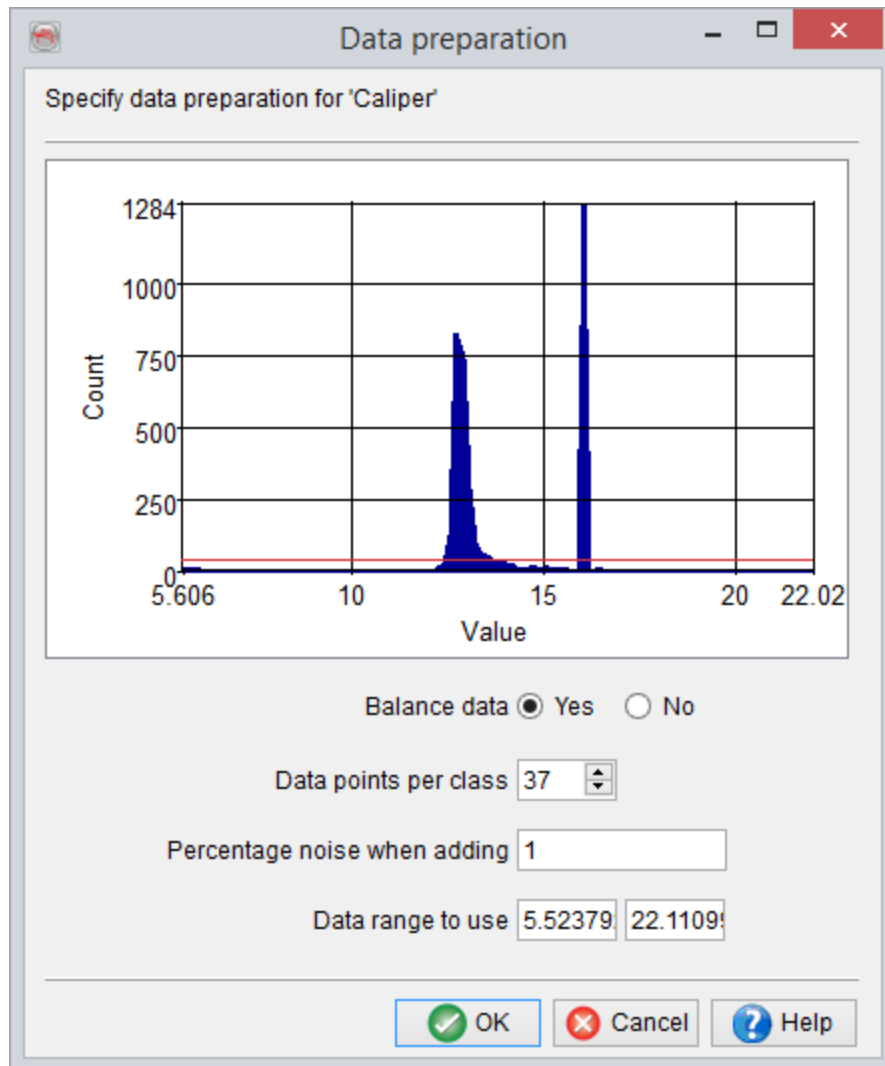
Based on the histogram display the user may:

- Adjust the output level for the flat distribution using the *Data points per class* parameter.
- Train for a range of values smaller than the extracted minimum and maximum values by specifying another *Data range to use*.

The over-represented classes will be decimated to the *data point per class* parameter.

The under-represented classes will be duplicated to the *data point per class* parameter, with a small change of the target value for each duplicated vector. This change is controlled by the parameter *Percentage noise when adding*, but the default value will be appropriate for most situations.

The binning is automatically performed to compute the number of classes based on minimum, maximum and number of points. At least 10 classes will be used, with a maximum of 100 classes. 20 and 50 classes may be used to reach an optimal of 25 vectors per class. This optimal is considered as the best compromise between statistical representation and training speed.



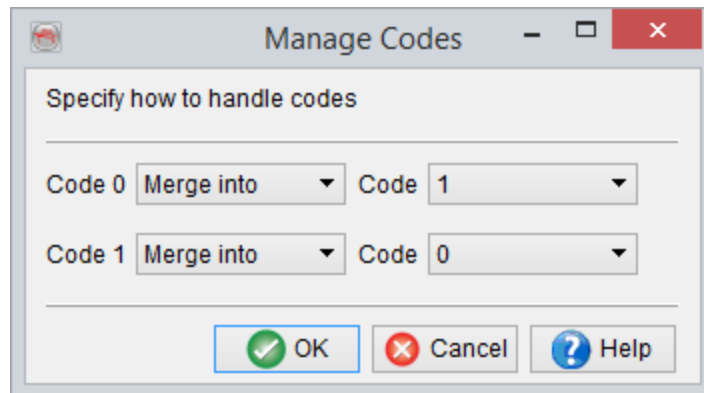
3.7.4.2 NN Lithology Codes

The neural network may be used to train for lithologies. For such a prediction the target log must contain lithology codes in form of integers. Multiple outputs will be available, on the contrary to training from ordinary log values:

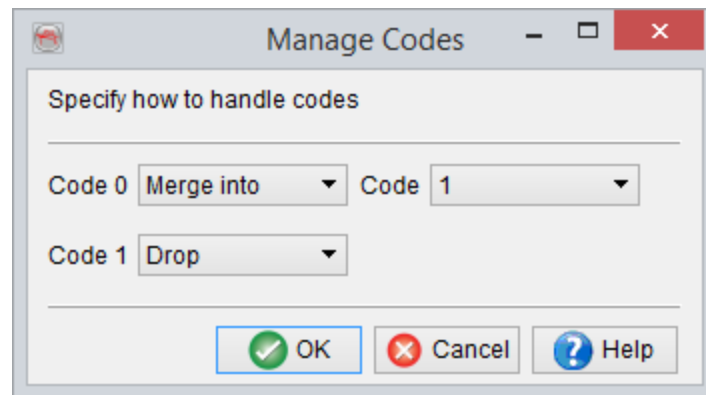
- A *classification* output providing the integer corresponding to the most likely lithology at each sample.
- For each lithology code, the *probability* for a sample to belong to this lithology.
- A *confidence* output that is the difference between the probability of the most likely lithology and the probability of the second most likely probability.

A maximum of 20 lithologies may be present in the input lithology logs. The upscaling (vertical sampling) method will be set to *Most Frequent* when using this option "Target contains: Lithology codes" such that the integers will remain integers after the up scaling. If non integer values are found in the input logs, they will be round-up to the nearest integer.

All lithology codes present in the logs may not be used for training. For instance the input log may contain four lithology codes for respectively sand, cemented sand, shale and salt. The first step is to assign a name to each code:



The second step may be to group (*Merge into*) several similar lithologies and/or to discard lithologies that will not be used to training (*Drop*):



Training begins when pressing "OK" on this window. The training window is similar to the training window for pattern recognition ([Picksets](#))

3.7.5 Training and Application

This window pops up as soon as OpendTect has created or retrieved the train and test sets. In order to create a training set, the software must compute all selected attributes at all picked locations. This may take some time. The user will be notified when the data is collected. Acknowledging the notification will automatically start the training phase.

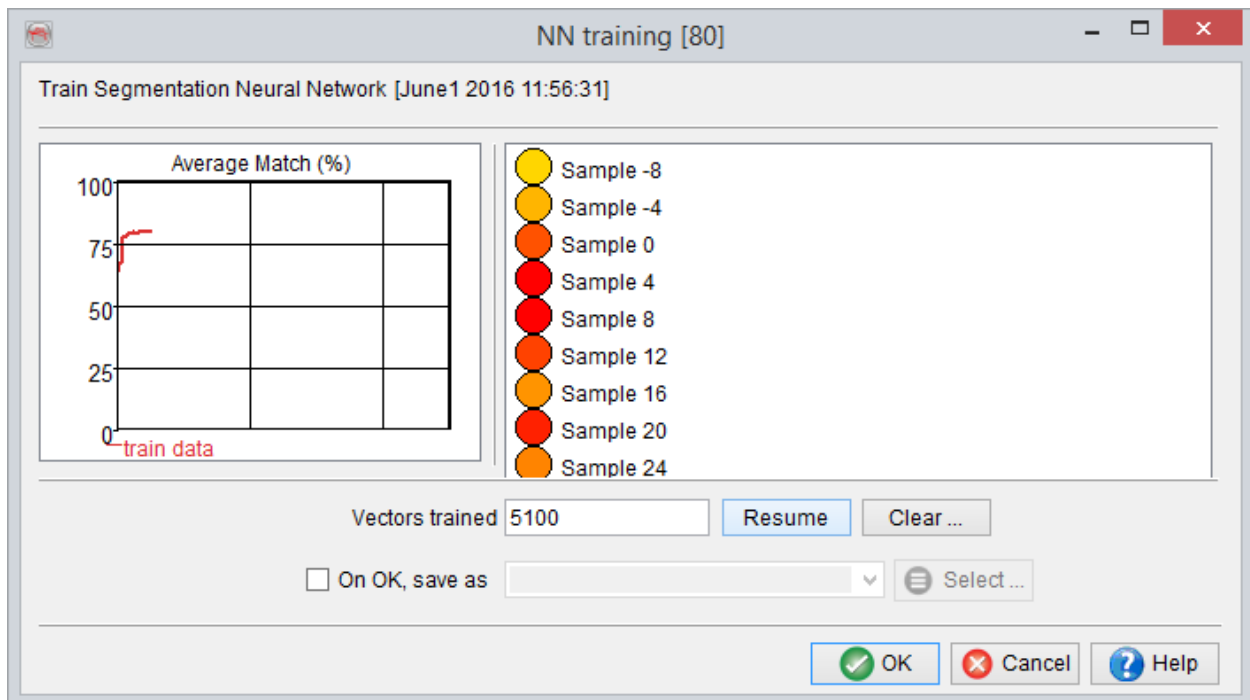
Training can be stopped and restarted with the *Pause (Resume)* button. *Clear* randomizes the weights of the current network. This implies that all training results are lost. The *Clear* option can be used when over fitting (see below) has occurred and the user wishes to restart training from scratch. The randomized network is then re-trained and the network is stopped before over fitting occurs. This can be done manually (pressing *Pause*) or by specifying a number in the *number of training vectors* field.

3.7.5.1 Unsupervised Training

In *Unsupervised training*, the network performance is tracked in a graph that shows the average match (confidence) of clustered input. Typically, the average match increases in a step-function. Each step indicates that the network has found a new cluster. Training can be stopped as soon as the average match has reached a stable situation. Usually this will be around 90%.

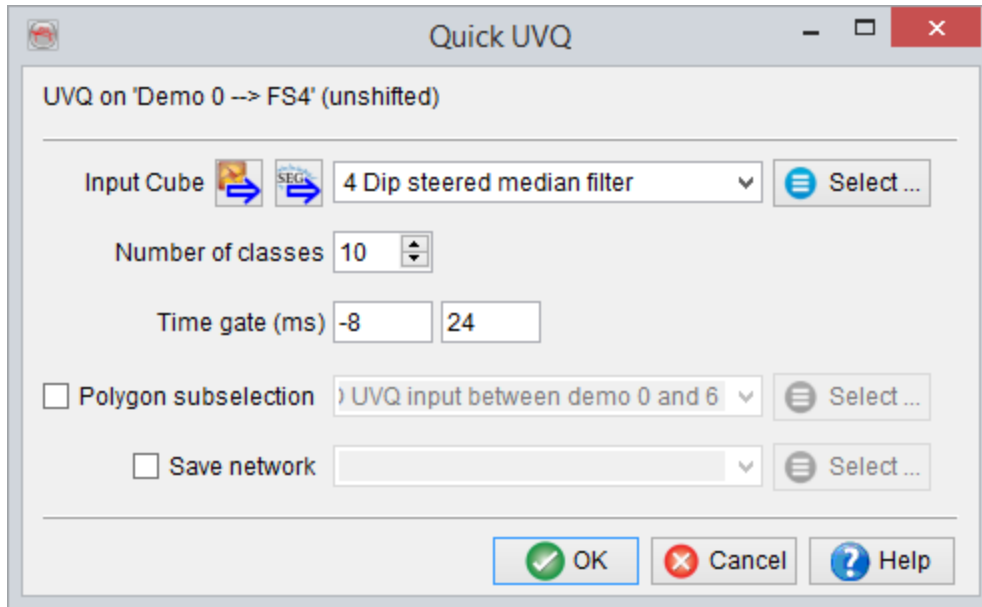
The colors of the input nodes in an unsupervised network will also change during training. In unsupervised mode, these colors do not indicate that one attribute is more important than another. All attributes in a clustering experiment are equally important.

Optionally, the neural network can be stored immediately on pressing OK. To do this, enter a neural network name in the appropriate field at the bottom of the NN training window.



3.7.5.1.1 Quick UVQ

The *Quick UVQ* horizon option can be used to quickly segment the seismic waveform along the interpreted horizon. The attributes and data selection are made automatically, contrarily to the standard NN training mode that request the user to define attribute and training locations (picks) before going into the [NN training module](#).



The sole window requests the user to select the input seismic data (2D or 3D according to the type of interpretation). Around 2000 input traces are randomly chosen to train the neural network. The waveform in a given time window is the main parameter to input, together with the output number of classes. It is highly recommended to save the output network.

The time window should match the length of the target of interest, and be extended on either sides by a few samples.

Example: You have a top reservoir horizon and the target is 50m = 32ms thick. An appropriate time window would be [-12, 44] ms along the horizon.

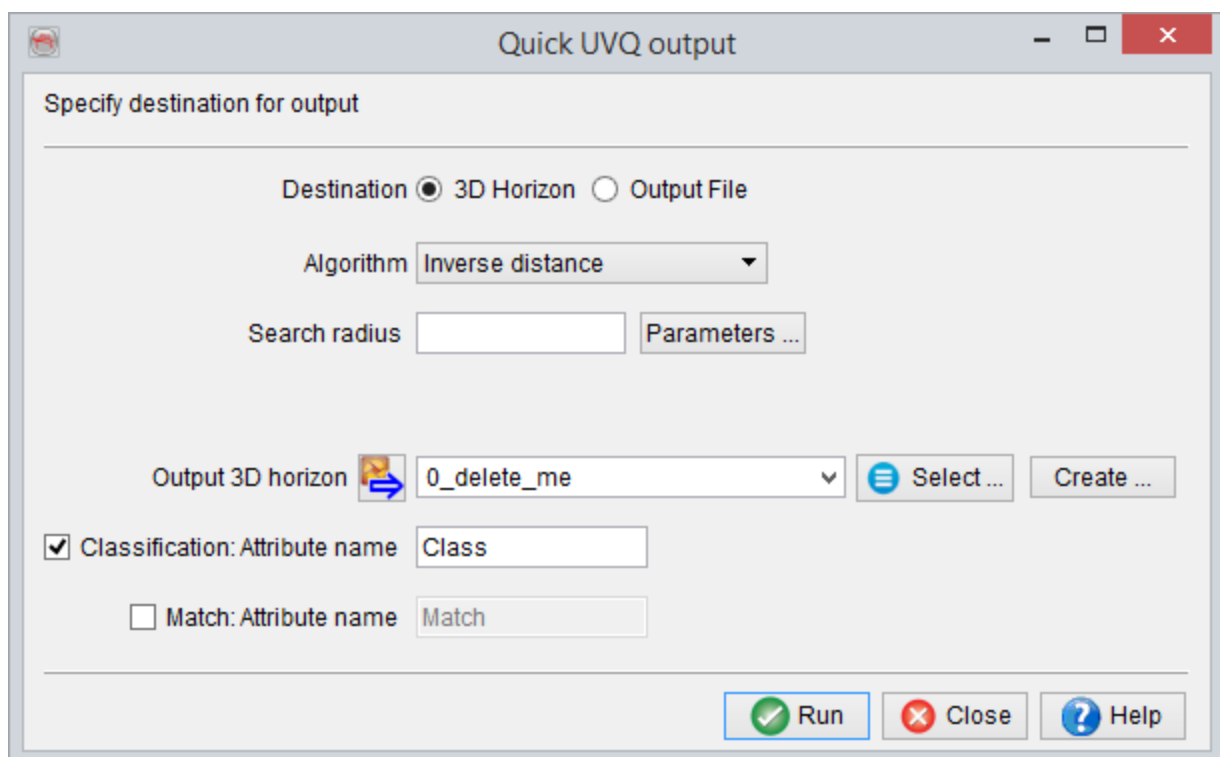
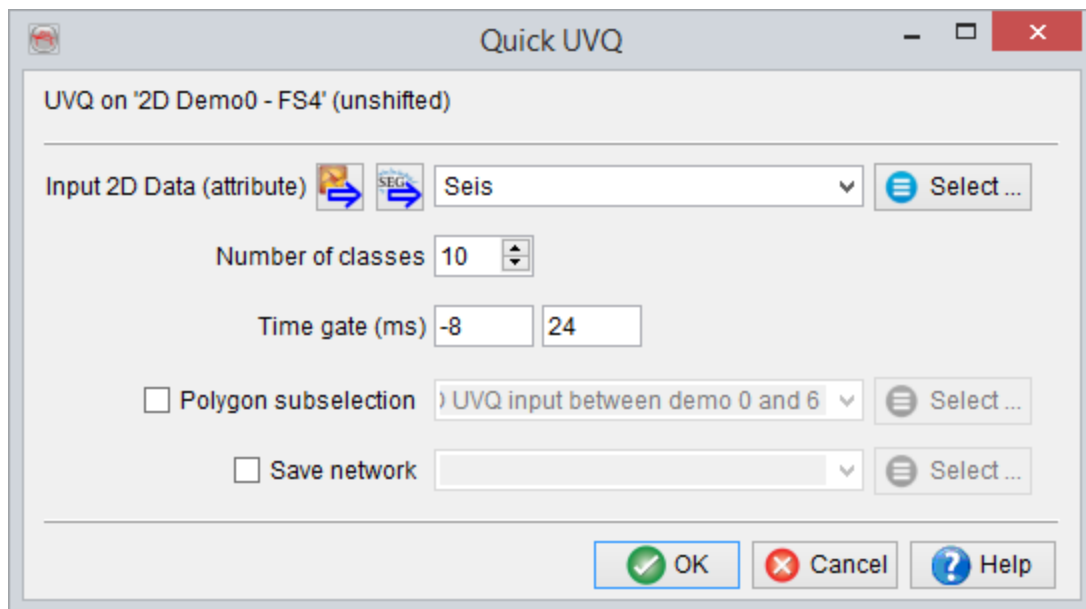
The number of class is hard to determine in advance. Too many classes will give redundant class centres, but too few classes will not match the seismic.

The 'OK' button will start the unsupervised Neural Network training. The training that results above 90% match could then be accepted to create output maps. The 'OK' button in the Neural Network Training window will automatically start processing the two outputs (Segmentation and match grids) on the 3D horizon. The processed grids are only saved to the database using the right mouse click on the processed attributes (Class, Match) of the horizon.

After training an additional Neural Network Info window will pop-up (see above). This dialog gives a detailed report about the Neural Network training. The Display button in the Neural Network info dialog is used to pop-up a class center display in a 2D viewer (see below). The classes can be color-coded using the Color Table, and the color table can be displayed in the 2D viewer by clicking on the 'show classification' button.

2D QUICK UVQs

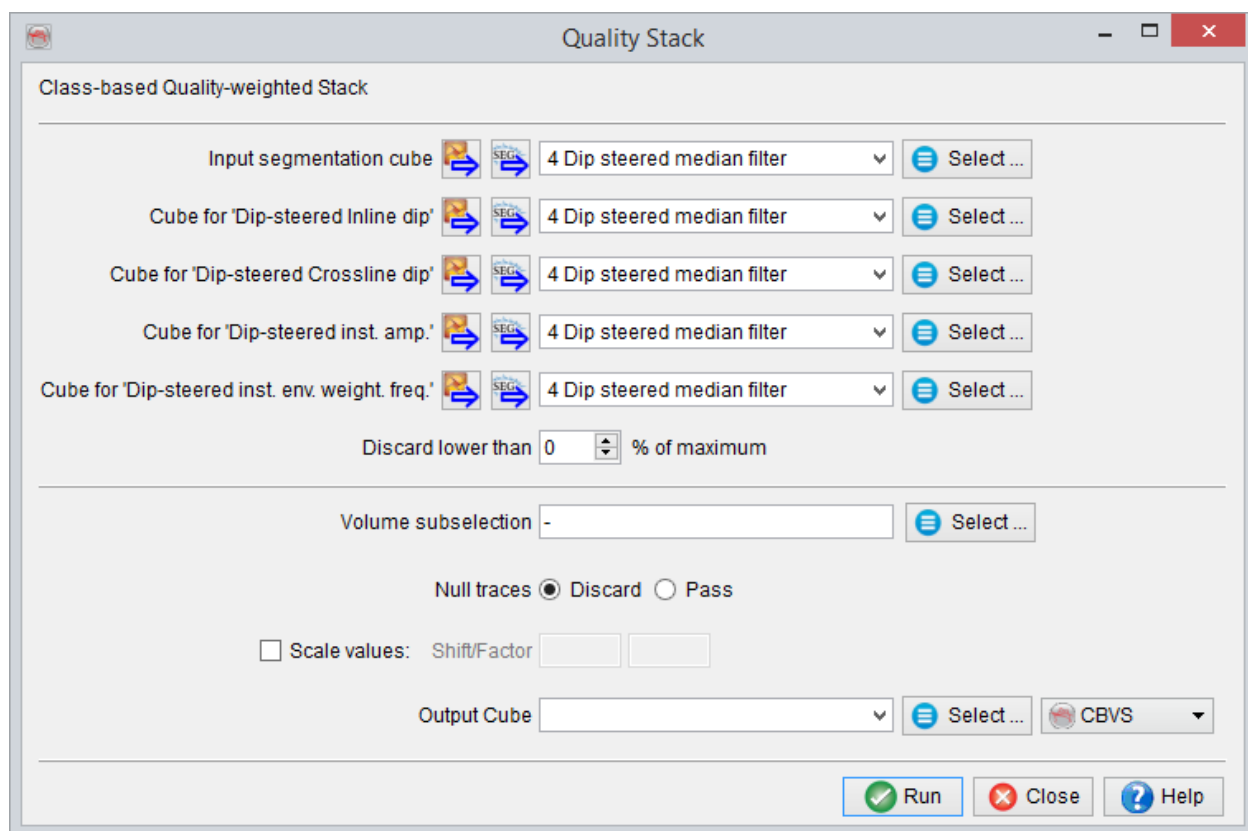
The similar workflow is also available for a mapped 2D seismic horizon. Similar settings are required and the processing steps are also the same (as explained previously). After the Neural Network training is finished, the system will prompt to grid and save the classification and the match grids to a 3D horizon. Select or create a 3D horizon to process the classification and match maps.



3.7.5.1.2 Quality-based UVQ Stacking

It is possible to use the output of an unsupervised segmentation for the stacking of seismic data. This functionality allows stacking of multiple cubes based on class number. The input nodes must be a measure of quality for each of these cubes.

This function is only available if the environment variable `OD_DGB_QUALITY_STACKING` is set to "Yes". The quality stack button appears in the "NN info" window of the loaded neural network. It will not appear if a single volume is used as input to all attributes of the neural network, like for a UVQ classification.



The segment cube must have been previously processed and saved on disk. It represents the first input. The other inputs are the volumes to be stacked, one for each

attribute (so the input of the attributes used in the neural network). Once again only stored volumes can be selected.

The output will be a weighted stacking of the input volumes. The weights are represented by the class center values (one for each attribute), and vary according to the best fitting class center as represented in the segment volume. Optionally low weight (relative to the maximum weight in a given class) may discard the corresponding volumes locally. Use a low value for the smoothest output, and a high value for the best discrimination.

3.7.5.2 Supervised Training from Picksets

In *Supervised mode*, the network's performance is tracked during training in two graphs: *Normalized RMS* and *% Misclassification*:

The *Normalized RMS* error curves (see network training picture below) indicate the overall error on the train and test sets, in red and blue respectively on a scale from 0 (no error) to 1 (maximum error). Both curves should go down during training. When the test curve goes up again the network is over fitting. Training should be stopped when (preferably before) this happens. Typically a RMS value in the 0.8 range is considered reasonable, between 0.8 and 0.6 is good, between 0.6 and 0.4 is excellent and below 0.4 is perfect. The normalized error is calculated as follows:

$$rms = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - e_i)^2}$$

and

$$normrms = rms / \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - mean)^2}$$

where

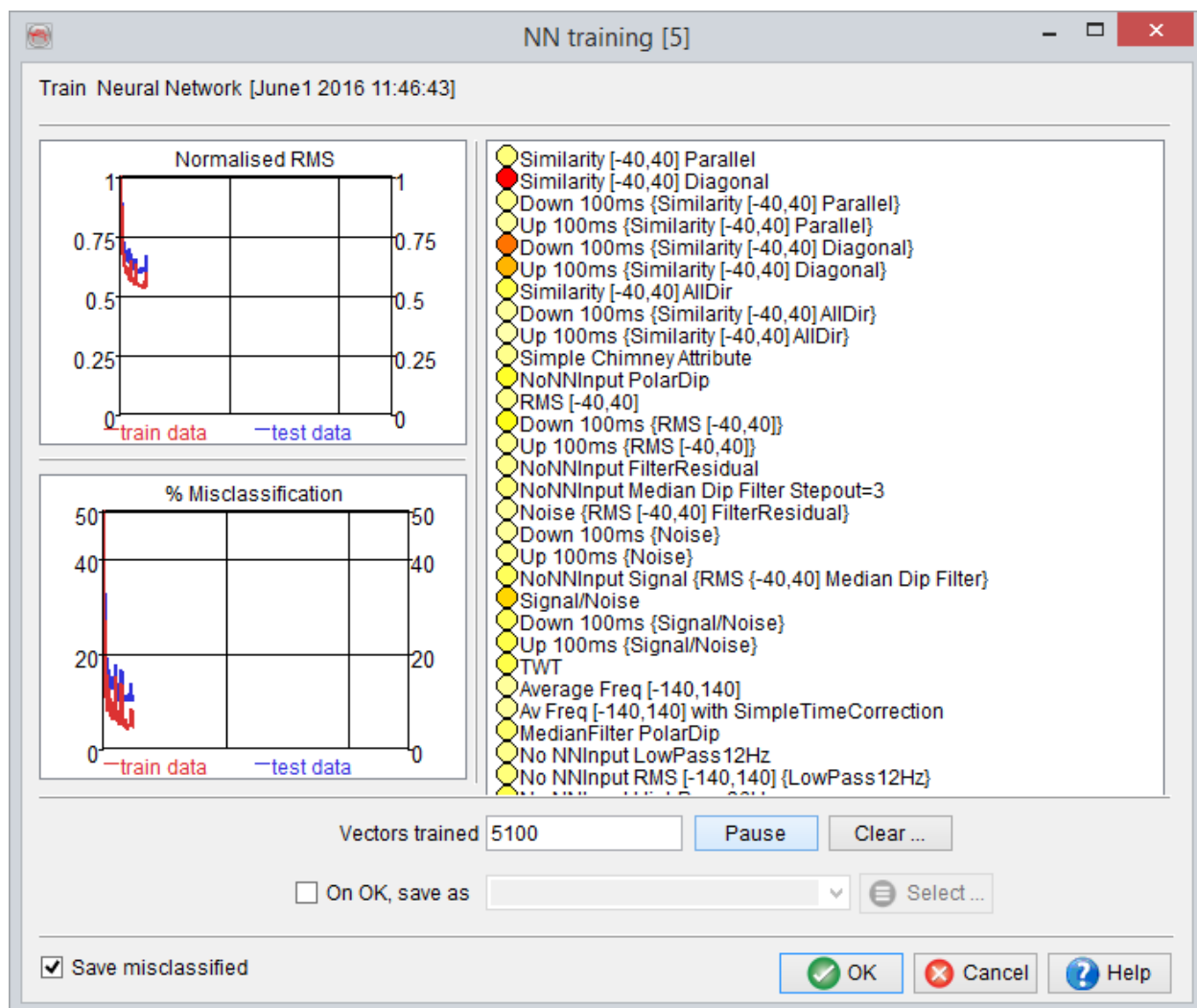
$$mean = \frac{1}{n} \sum_{i=1}^n t_i$$

The *percentage misclassification* shown in the lower left corner is a much easier quality control parameter to interpret. It simply shows how the percentage of the training and test set that is classified in the wrong class.

On the right-hand side of the window a graphical representation of the input attributes is shown. The circle in front of the attribute name changes color during training. The colors reflect the weights attached to each input node and are therefore indicative for the relevant importance of each attribute for the classification task at hand. Colors range from red (high weight means high importance) via yellow to red (relative small weights, less important). This feature is very useful when you wish to design small networks to increase processing speed.

Optionally, the neural network can be stored immediately by pressing the OK button. First, enter a neural network name in the appropriate field at the bottom of the NN training window.

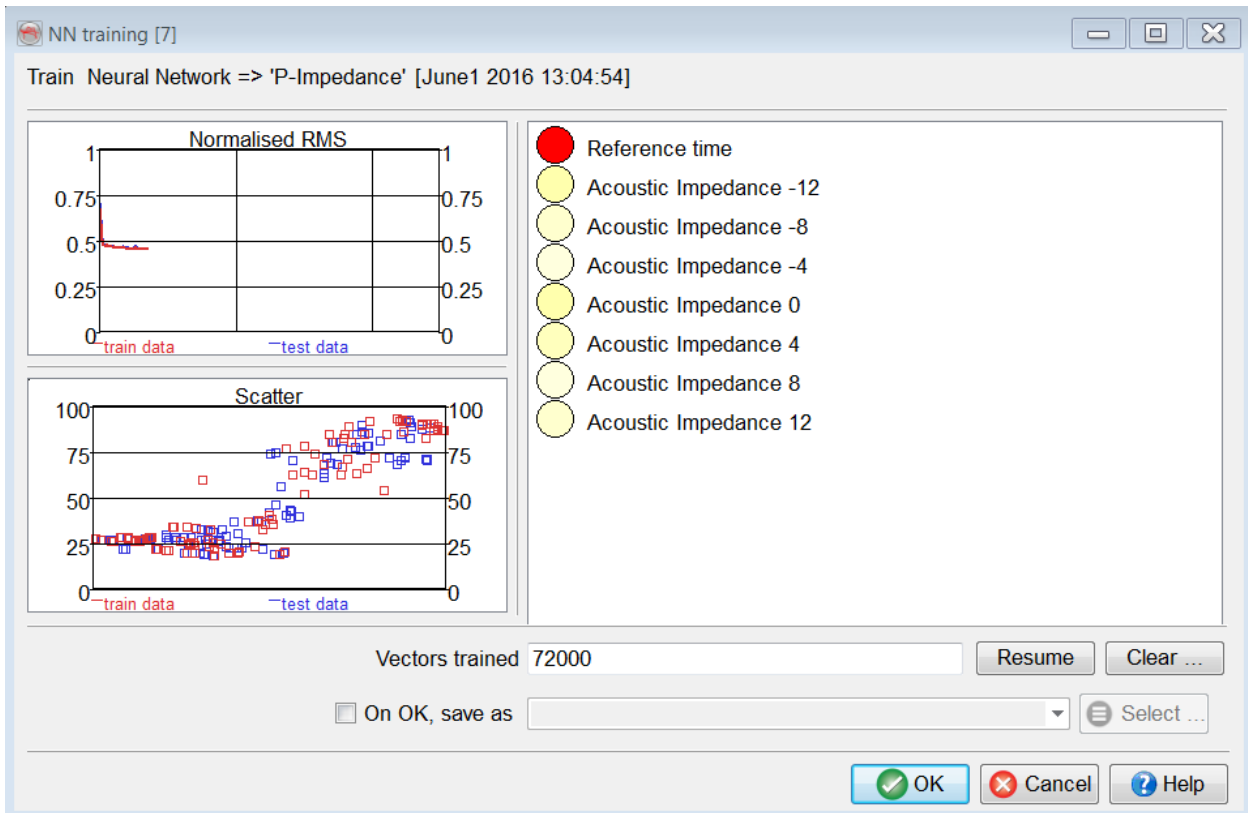
The *Save misclassified* toggle allows saving the misclassified picks in a new Pickset. This Pickset is automatically loaded in OpendTect again. The Pickset can be indicative of picking errors. It is not recommended to bluntly remove the misclassified picks from a Pickset, since good picks, although misclassified during training, still help neural network training.



3.7.5.3 Supervised Training from Well Data

The Supervised training window from well data is very similar to the training window from a Pickset (see below). The only difference is the display of a *scatter plot* instead of a *% Misclassification plot*. A scatter plot shows the actual target data on the horizontal axis and the predicted target data by the neural network, as it is at that moment, on the vertical axis. Not all nodes are plotted. Only a random selection of the used train and test data is shown. Ideally, after sufficient training, all data points should be on the diagonal. That would mean that the trained neural network predicted all examples correctly.

However, this will rarely be the case. In most cases, the data will cluster along the diagonal. The narrower this cloud, the better the neural network is trained. Overtraining occurs when the Normalized RMS of the test data increases, while the Normalized RMS of the train set decreases. This usually also means that the cloud of train nodes becomes narrower, while the cloud of test nodes becomes wider again.



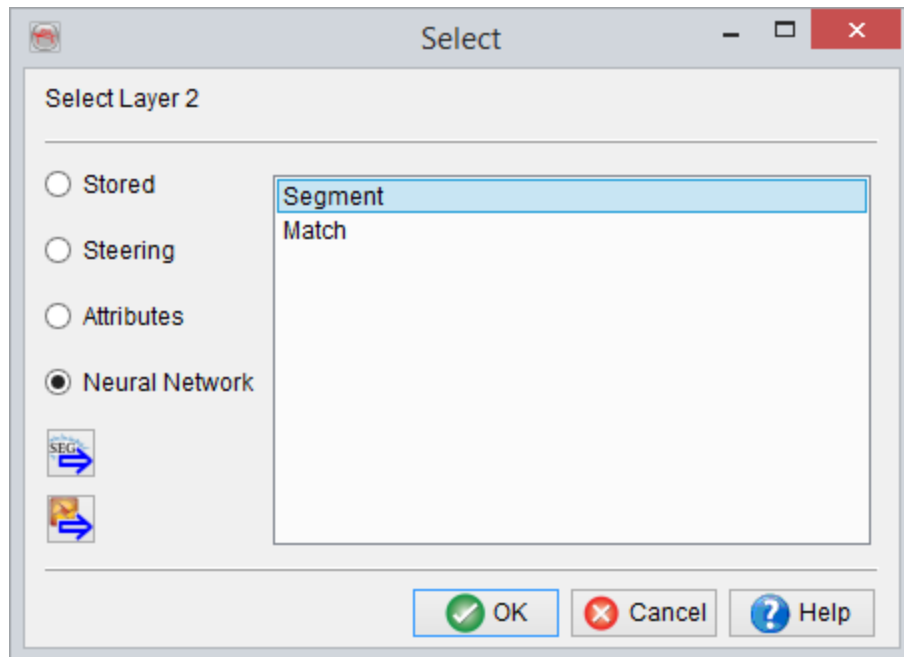
3.7.5.4 Application

Neural networks should be applied like standard attributes:

On-the-fly application

Load a slice in the scene, use the option "Add > attribute". Select the "Neural Network" list of attributes. The type of output depends on the type of neural network:

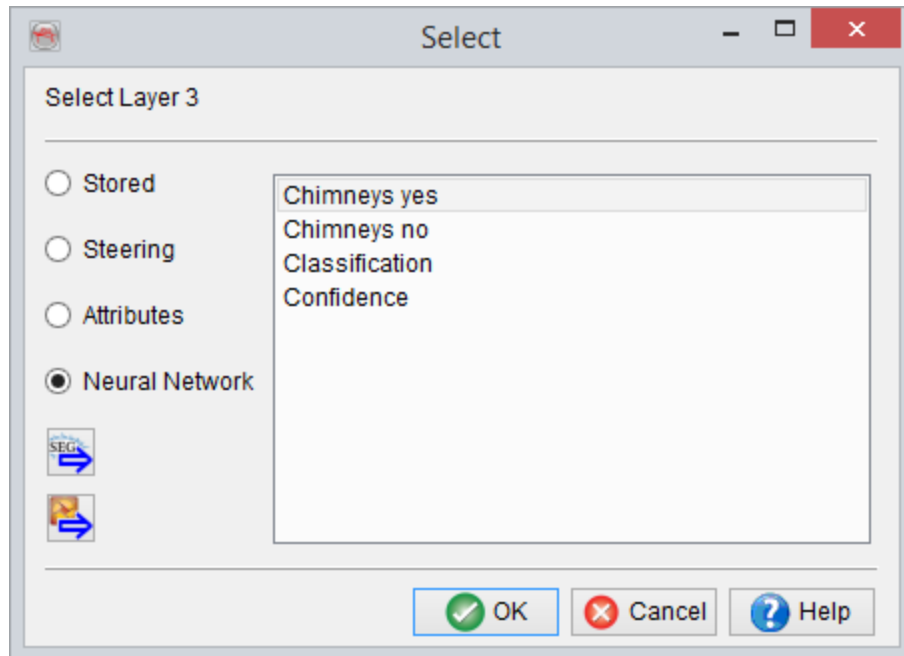
1. Pattern recognition, unsupervised mode:



Segment: Index of the most likely class center (as seen in the [Neural Network Information](#))

Match: Measure of similitude between the most likely class center and the actual attributes. Between 0 and 1 (identical).

2. Pattern recognition, supervised mode:



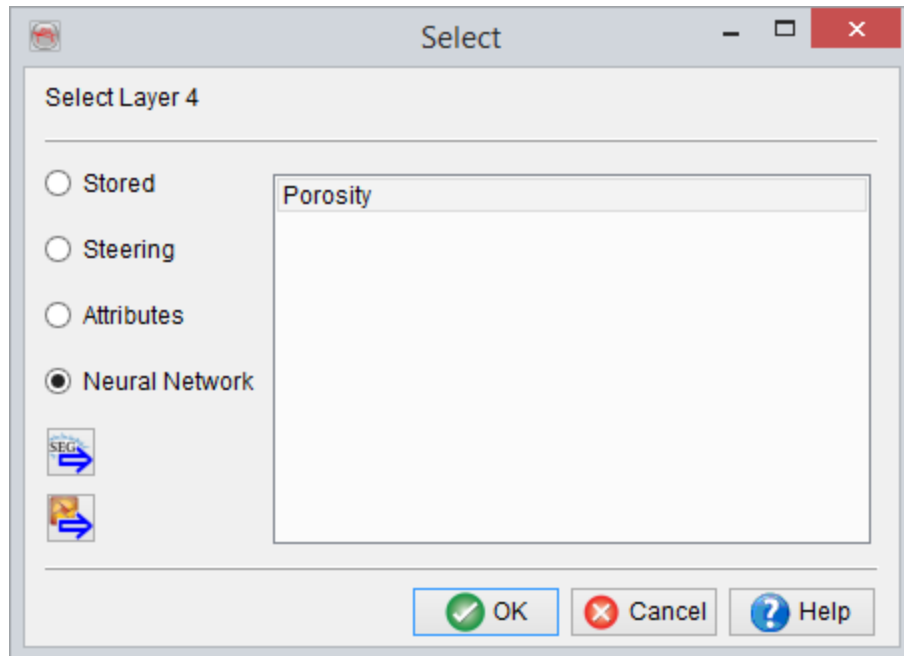
[Pattern 1]: Neural network probability that a given sample matches the interpreted "Pattern 1": Returns amplitudes around 1 if the sample is similar to the picks of the corresponding first pickset. Returns values around 0 otherwise.

[Pattern 2]: Neural network probability that a given sample matches the interpreted "Pattern 2": Returns amplitudes around 1 if the sample is similar to the picks of the corresponding second pickset. Returns values around 0 otherwise.

Classification: Returns the index of the most likely pattern in the order seen above.

Confidence: Returns the difference between the largest and second largest NN probabilities in the set (You can train with more than two picksets)

3. Property prediction



[Output log]: Returns the predicted value for the target log. The uncertainty on the predicted value is given in the [training report](#).

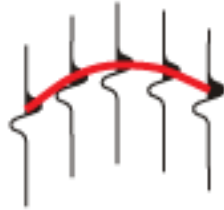
Batch processing

The batch processing on the elements described above should be started from the menu Processing > Create Seismic Output > Attributes > Single Attribute > (2D/3D).



You need to launch one batch process for each element under the list "Neural networks".

4 Dip-Steering



4.1 Background

Dip-Steering is defined as the process of following seismic reflectors by auto-tracking the pre-calculated dip-field from a given starting position. The pre-calculated dip-field in OpenTect is stored in a Steering-Cube, which contains two values at each seismic sample position: the dip in the inline direction and the dip in the cross-line direction.

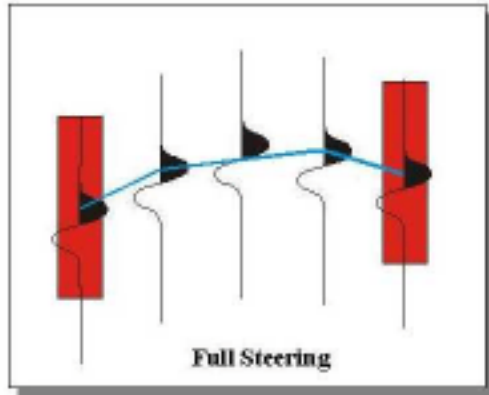
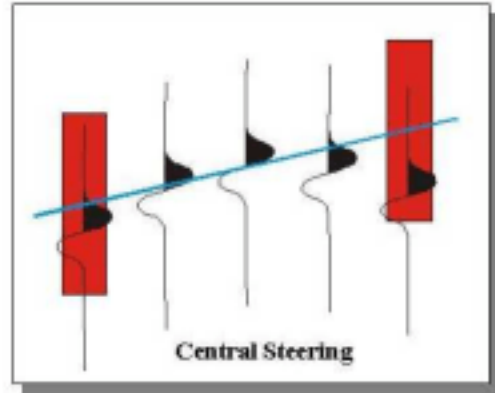
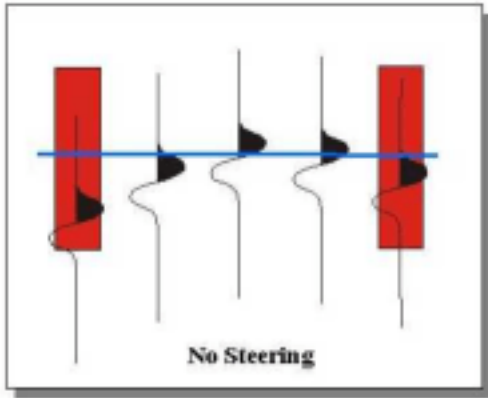
Dip-Steering is a key concept in OpenTect that is used to:

- Compute dip-steered multi-trace attributes. Examples are: Similarity, Texture, Volume Statistics, ... Also Neural Network-based 'probability' volumes such as ChimneyCubes and FaultCubes include many dip-steered attributes.
- Compute attributes directly from the Steering-Cube. Examples are: the set of Volume Curvature attributes and (geologic) dip and azimuth.
- Apply dip-steered filters. Examples are: dip-steered median filter and dip-steered fault-enhancement filter.

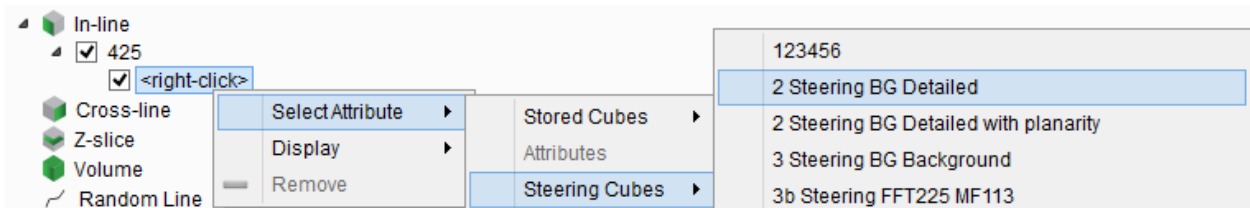
- Auto-track horizons. Examples are the HorizonCube auto-tracker which tracks thousands of horizons directly from the dip-field and OpendTect's conventional amplitude and similarity tracker that can optionally include dip to constrain the tracker.
- Grid loosely mapped horizons. The dip-steered gridding algorithm fills in holes and unmapped parts to generate a horizon that exists at every seismic location.

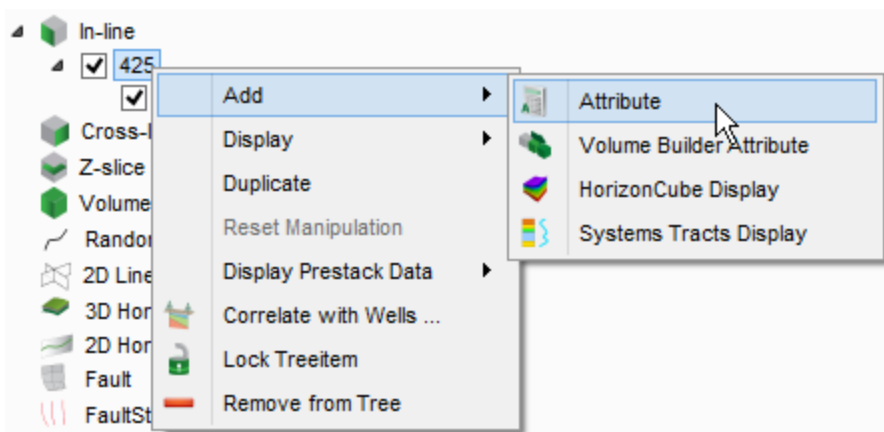
Depending on the process it is often useful to work with more than one Steering-Cube. Typically, we use two Steering Cubes. The 'Detailed SteeringCube' contains dips as calculated by the dip-computation algorithm (OpendTect supports multiple algorithms, see ...). The Detailed SteeringCube is used if we need to preserve details in the data, e.g. when computing Volume Curvature attributes in fracture detection studies. The 'Background SteeringCube' is a smoothed (usually median-filtered) version of the Detailed SteeringCube. The Background SteeringCube contains less noise and the dips follow larger structural trends. It is typically favoured for dip-steered filtering operations.

The Dip-Steering plugin for OpendTect supports two different modes of data-driven steering: Central Steering and Full Steering. In Central steering the dip / azimuth at the evaluation point is followed to find all trace segments needed to compute the attribute response / filter / track the event. In Full steering the dip/azimuth is updated at every trace position. The difference between 'no steering', 'central steering' and 'full steering' is shown in the following figures. Note that these figures show the 2D equivalent of steering, which in actual fact is a 3D operation.

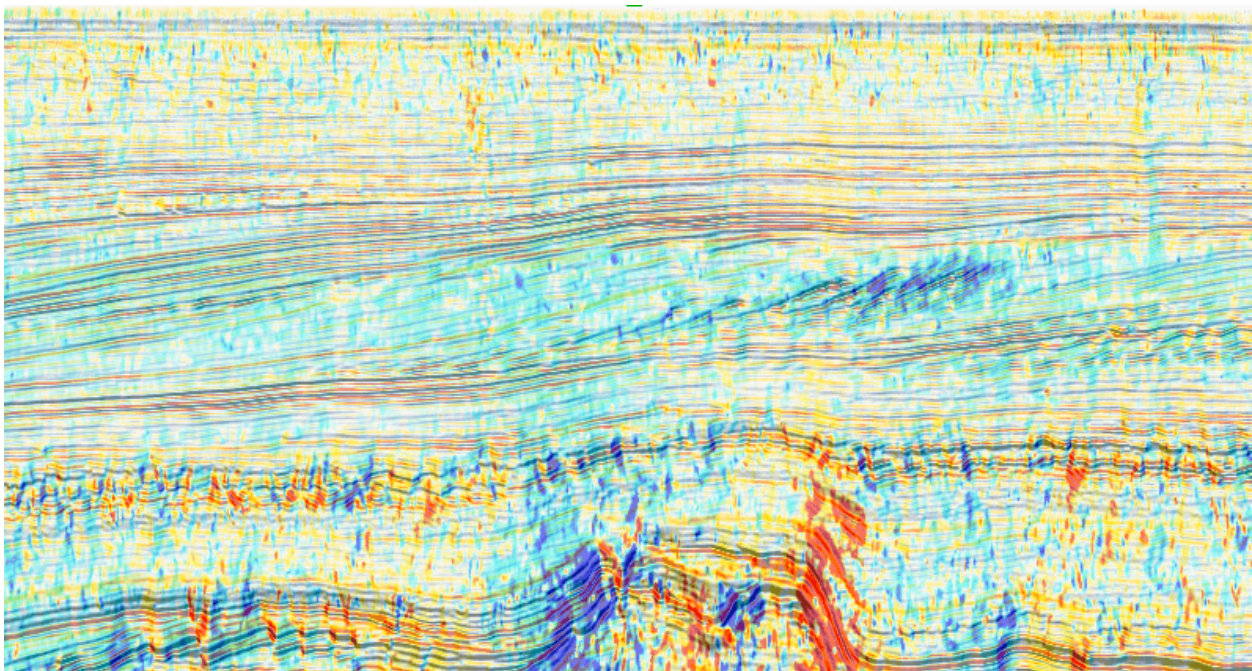


SteeringCube information can be displayed in a 3D scene directly from the OpendTect tree, in two ways, as shown in the pictures below. This option displays the cross-line dip along an inline, or the inline dip along a cross-line.



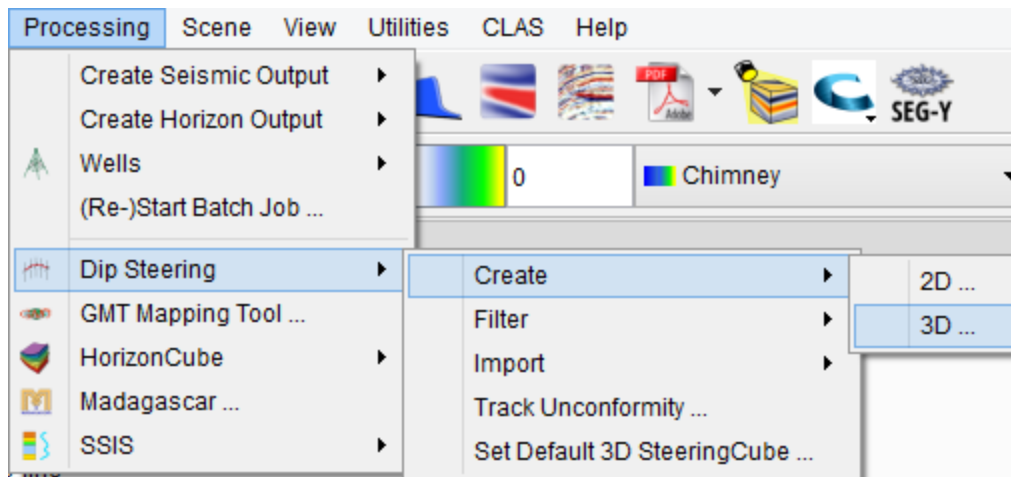


Detailed SteeringCube of inline 425 in F3-Demo. For display purposes the (cross-line) dip image was made (50%) transparent:



Dip-Steering options are launched from the Processing -> Dip Steering Menu. Dip-Steering is available for 2D seismic and 3D seismic. A SteeringCube for a 2D line stores at every sample position the dip in the line direction. A SteeringCube for a 3D

volume stores two values per sample position: the dip in the inline direction and the dip in the cross-line direction.



💡 *The windows pictured and described in this manual are the ones pertaining to 3D processing. 2D processing windows are similar and the same description can therefore be used for 2D dip-steering.*

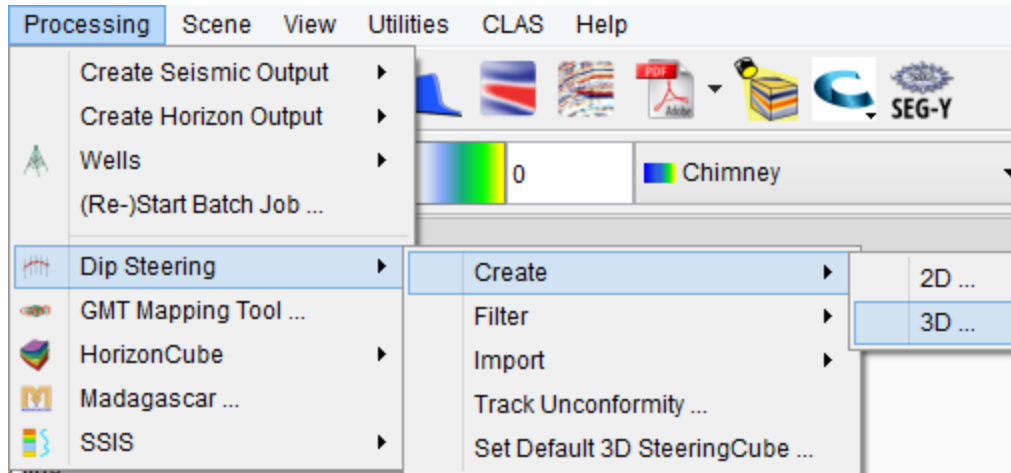
References

The following references are available from the dGB website. Although slightly out of date, the concepts are still valid:

- [Benchmark of HorizonCube algorithms](#)
- [Introduction to SteeringCubes](#)
- [SteeringCube workflows](#)

4.2 Create Steering Data

The processing main menu is use to start (2D/3D)-SteeringCube creation and filtering.



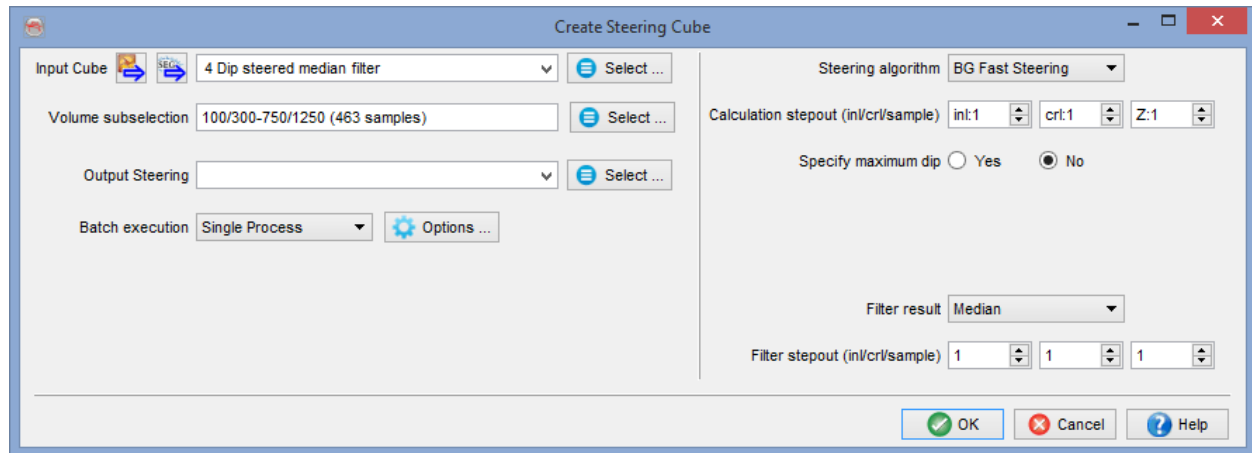
Select input data cube (usually a seismic volume) and optionally the sub-volume to process.

Tree types of *Steering algorithms* are supported:

- PCA
- Phase-gradient (former BG)
- FFT

PCA (Principal Component Analysis) is an alternative method to estimate smoothed reflection dips. It produces relatively smoother dip/azimuth estimates of a seismic reflection. The PCA technique has an advantage that it is faster with a built-in structural oriented smoothing. One may not require additional median filtering, which consumes time on larger size datasets.

Phase- gradient Steering is a quick algorithm developed by BG. It is based on analysis of the gradient of the amplitude data, both vertically and horizontally. The BG fast algorithm is prone to noise, this can be overcome by adding a median or average filter.



FFT Steering computes dip in a small sub-volume defined by the step-outs by searching for the highest energy in the 3D Fourier transformed domain.

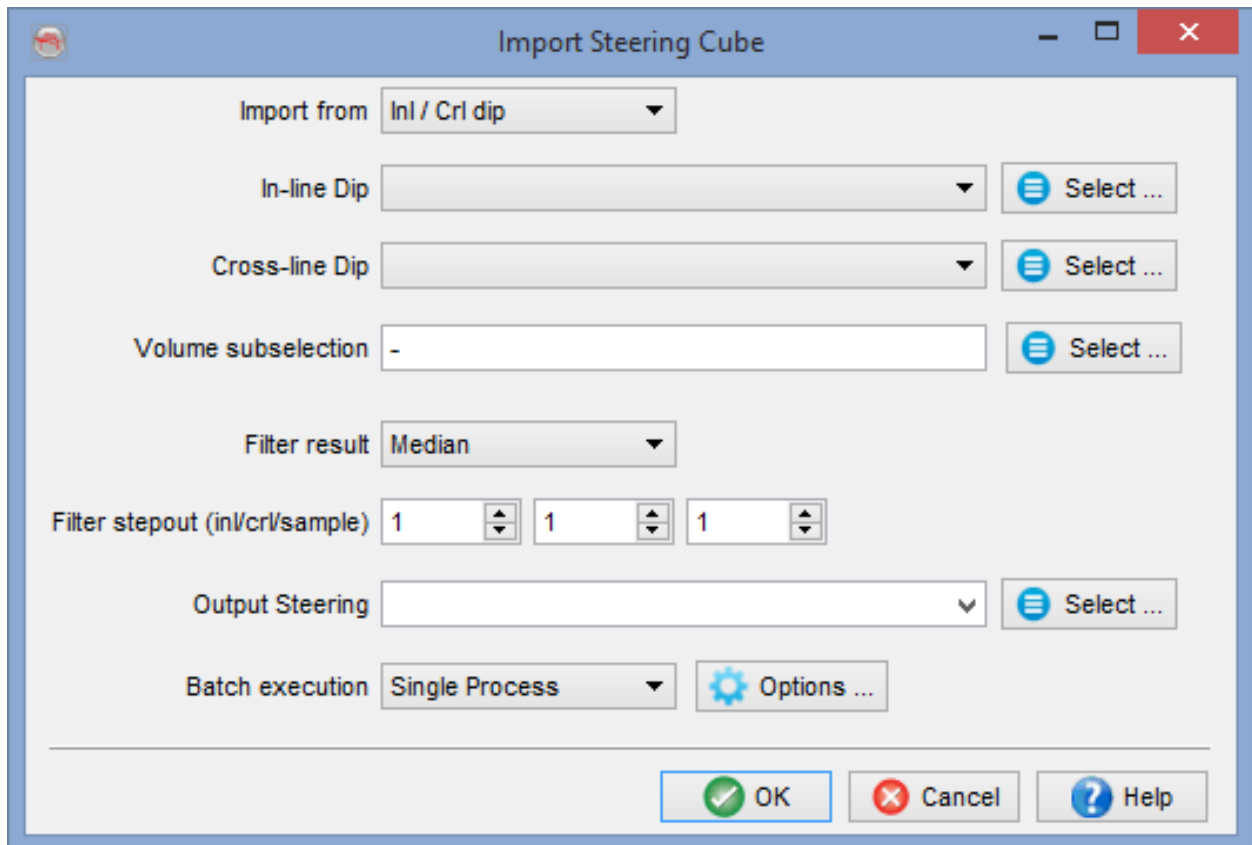
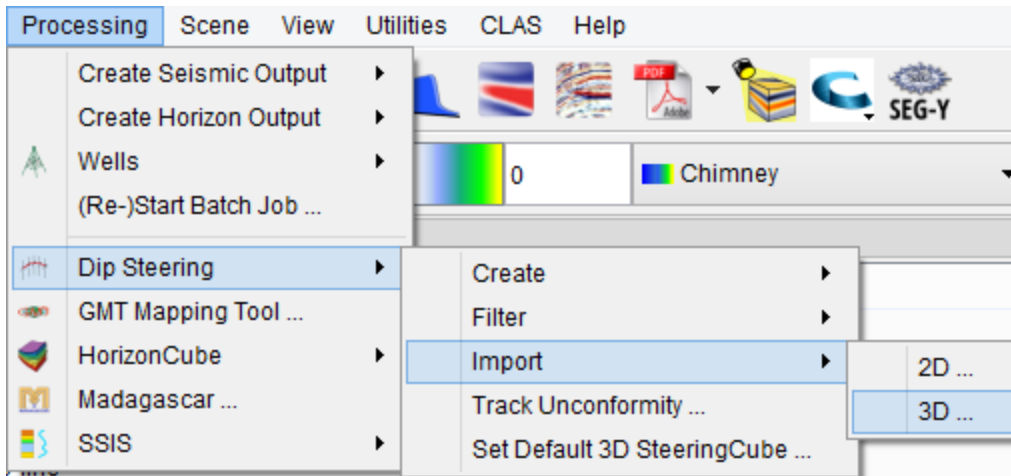
More background information, including a benchmark of the different algorithms and visual examples are presented in the [benchmark section](#).

The optional *Specify maximum dip* limits dip values derived from the input data. Another option to avoid extreme dip values is to filter the steering data with a *Median filter*. This removes the outliers from the steering data. The stepouts are defined in samples, regardless sampling rate.

The processing specifications as defined in the window can be saved (optional). Provide a file name in the textbox to store the processing specification. If this space is left empty, the processing specification is not saved. If, for any reason, the processing is aborted, the process can be re-started with this parameter file using the *Re-start* option under the *Processing* menu.

The *Proceed* button starts the single-machine or Distributed Computing mode. For more information on single-machine and Distributed processing, open the help menu from the Distributed Computing window.

4.2.1 Import Steering Data



Subject to Dip-Steering licensing permissions Steering data created in other software packages can be imported into OpendTect, Third-party steering data is supplied in the form of seismic volumes: two volumes for a 3D SteeringCube (inline dip and cross-line dip, or dip and azimuth), or one volume for 2D Steering Data (the dip in the line direction). The import functionality converts the 3rd-party data sets into OpendTect compatible Steering Cubes for 3D, or 2D, respectively.

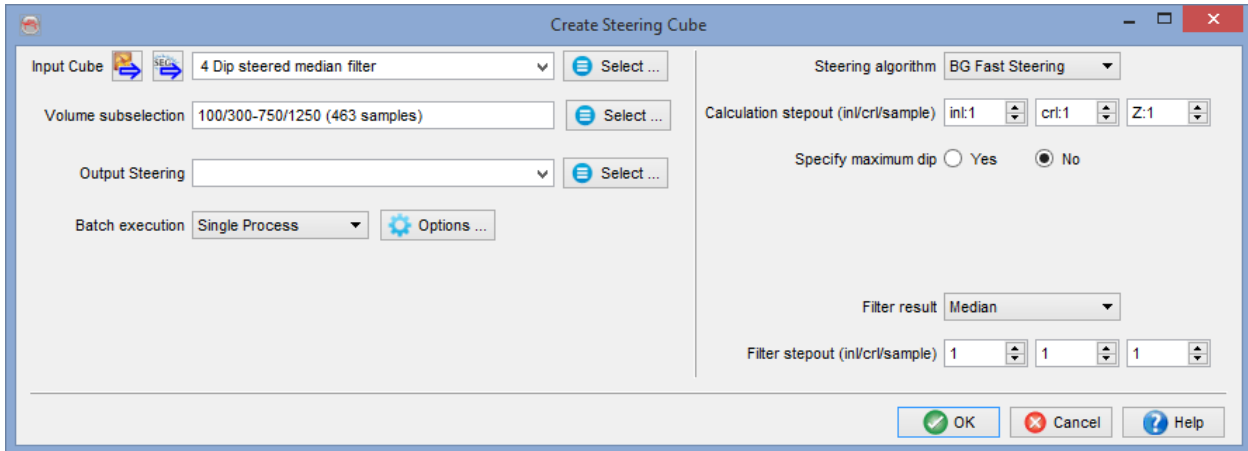
The input data must adhere to the following criteria:

- Dip values should not be negative and given in usec/m, or in mm/m if the survey is in depth.
- Azimuth should be given in degrees from -180 to 180. Positive azimuth is derived from the inline in the direction of increasing cross-line numbers. Azimuth = 0 indicates that the dip is in the direction of increasing cross-line numbers. Azimuth = 90 indicates that the dip is in the direction of increasing in-line numbers.

Optionally the imported data can be smoothed by application of a median, or average *filter*. The default is not to smooth as smoothing filters can also be applied in a separate step afterwards (see [Filter Steering Cubes](#)).

The import function can be executed in *batch*, on a single machine, or distributed over several computers

4.2.2 Create SteeringCube




Select the *Input Cube* (usually a seismic volume) and optionally specify the *Volume subselection* to process.

Give the SteeringCube to be computed a name under *Output Steering*.

Under *Batch execution* specify where the processing is to be done. *Single Process* means the SteeringCube is processed on the machine you are currently working on. Under *Options ...* you can set the job priority (Linux nice level) from -19 (low) to 19 (top).

Distributed Computing is the recommended mode as computing a SteeringCube is a relatively CPU intensive process. In this mode batch processing is distributed over all computers (workstations and clusters) that are available to the user. The Distributed Computing window is launched after all parameters were set and OK is pressed.

 *Distributed Computing (processing) is a unique selling point of OpendTect that is not automatically available after a default installation. This mode depends on the network environment and needs to be installed by the*



system administrator. For details, please see: Chapter 5.5 of the System Administrator's manual.

OpendTect supports three types of *Steering algorithms*:

- *PCA (new in 6.2)*
- *Phase-gradient (former BG)*
- *FFT*

PCA (new in 6.2)

Dip Estimation using Principal Component Analysis (PCA)

Reflection dips are estimated using Principal Component Analysis from the seismic amplitude field. This specific PCA uses an Eigen Value Decomposition (EVD) in order to extract the direction (as a 3 component vector) of maximum amplitude changes, as the largest eigen value of the decomposition.

This direction is the normal to the wavefront at any position of the seismic reflector, thus perpendicular to the dip. The second and third eigen values will be located within the plane tangent to the seismic reflection.

The EVD is applied on a sample by sample basis, using the following six amplitude gradient products, previously computed and smoothed volumetrically using recursive gaussian filters:

Gradient inline direction	*	Gradient inline direction
Gradient crossline direction*		Gradient crossline direction
Gradient vertical direction	*	Gradient vertical direction
Gradient inline directions	*	Gradient crossline direction
Gradient inline directions	*	Gradient vertical direction
Gradient crossline direction*		Gradient vertical direction

The smoothness of the output dip field is entirely determined by the strength of the filters applied to the gradient product attributes prior to the SVD decomposition. Eventually it is not one but three filters that are applied in the vertical, crossline and inline directions consecutively. It is thus possible to adjust each of the three filter strengths independently. Note however that applying a minimum smoothing of 1 in the three directions is essential for a stable estimation of the dip field. Post-processing filtering is usually not required.

Two filtering algorithms are available: the Deriche filter is applied for standard filter sizes (below 32), and the Van Vliet filter for bigger sizes ($\sigma = 32$ and higher). The runtime, for both filters, is independent of σ : Running it with $\sigma = 1$ takes as much time as with $\sigma = 20$. For example a processing done with filter size [2, 6, 40] will first apply a Van Vliet filter of $\sigma = 40$ in the vertical direction. Then a Deriche filter of $\sigma = 6$ is applied to the previous output in the crossline direction. Finally a Deriche filter of $\sigma = 2$ is applied in the inline direction to the output.

Because of filters being recursive filters, a larger number of samples in the direction where the filter is applied is required in order to get stable results. As a result the processing engine will automatically load up to 50 additional samples when available in order to stabilize the result. Running the PCA processing in chunks should thus deliver an almost identical result as running the entire dataset at once. However there will be a runtime penalty if the chunk size is set much smaller than 50 inlines per chunk, for instance using Multi-Machine distributed processing.

The EVD may optionally output the relative difference between the first two Eigen Values. When applied on seismic reflections, this attribute is proportional to the "planarity" of the reflection: It will reach a maximum of 1 if the reflection locally coincides with the plane tangent to it, have with be always remain larger or equal to zero.

This optional attribute is stored as a third component of the output SteeringCube. The planarity attribute, similar to a similarity attribute, can be used to weight the inversion-based horizon tracking such as Unconformity Tracker or Multi-horizon HorizonCube.

Phase-gradient (former BG)

Phase-gradient Steering is the recommended algorithm for applying dip-steered filters and to compute dip-steered attributes. Phase-gradient Steering is a quick algorithm developed by the BG-Group. It computes dip in a small sub-volume defined by the step-outs from the gradient of the instantaneous phase. The Phase-gradient algorithm is prone to noise, which can be overcome by adding a median or average filter. The default is to apply a mild *median filter* with step out 1,1,1.

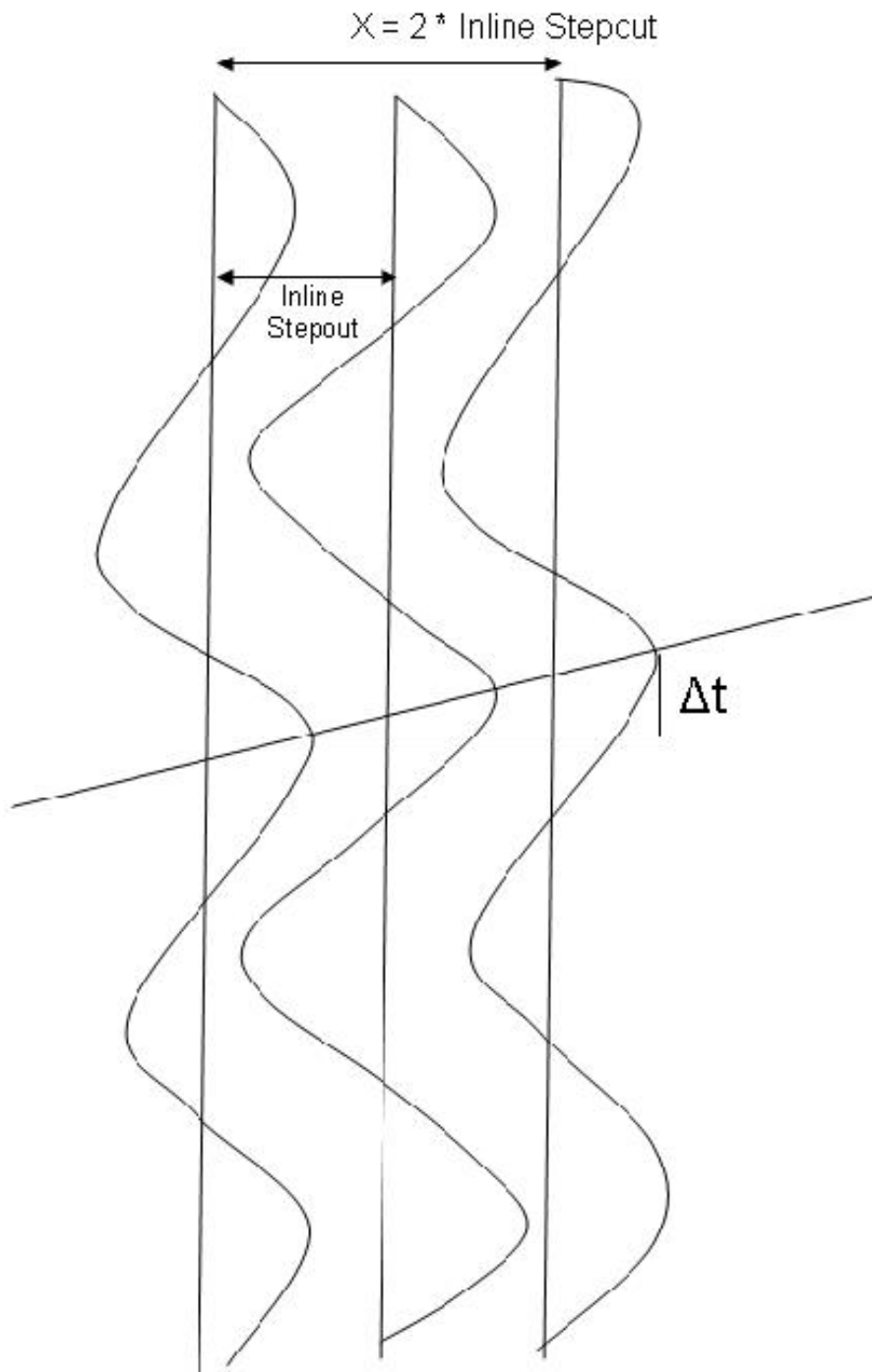
Optionally, dip computation can be limited to a maximum dip (in us/m).

FFT

FFT Steering computes dip in a small sub-volume defined by the step-outs by searching for the highest energy in the 3D Fourier transformed domain.

Also with *FFT Steering*, dip computation can optionally be limited to a *maximum dip* (in us/m) and the results can optionally be smoothed with a median or average *filter*. The default is not to smooth as smoothing filters can also be applied in a separate step afterwards (see [Filter SteeringCubes](#)).

Steering calculates the dip in the inline direction and in the cross-line direction from extrema. All Maxima and Minima are determined on the central trace and two neighboring traces located at the *stepout* positions. Maxima and minima with the smallest Δt separation (see image) are considered to represent the same seismic event. This results in dip values at all extrema positions. Resampling these values to the seismic sampling rate yields the Steering data at the central trace position.



Inline dip = $\Delta t/x$

Finding dip in Inline direction

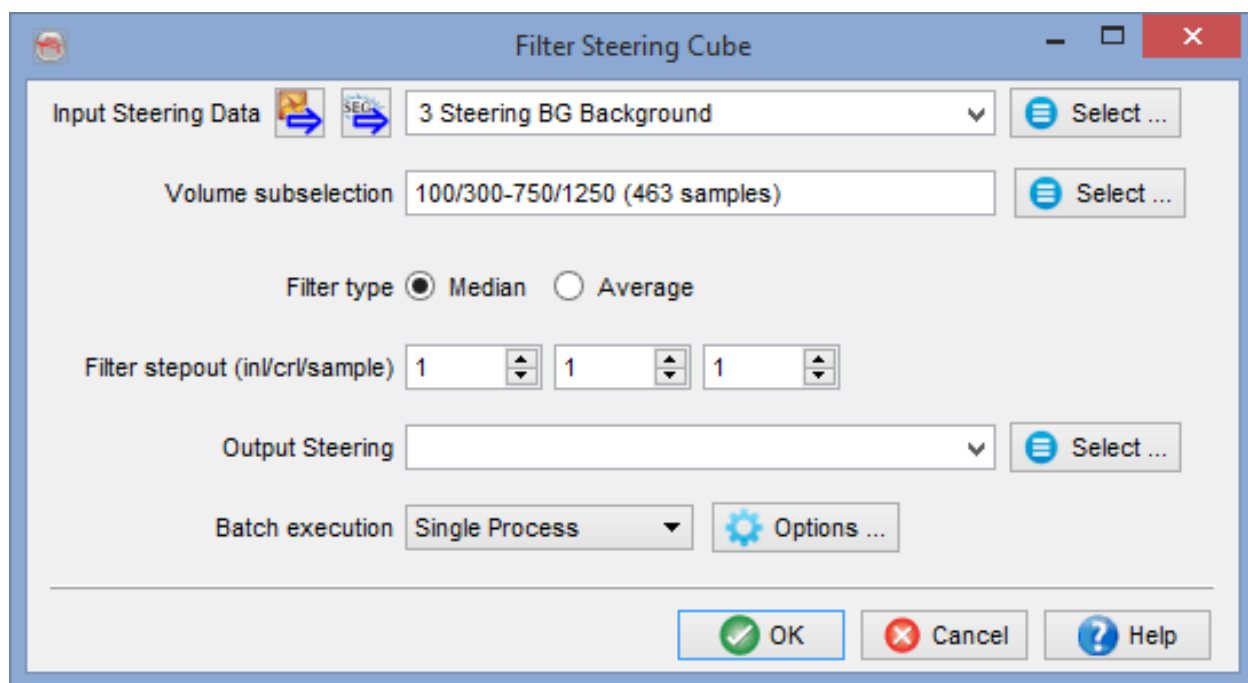
The *maximum dip (us/m)* parameter determines the search gate on the neighboring traces within which a maximum, or minimum must be found. The parameter ensures unrealistic dips are found, which may happen in noisy areas.

Also Steering supports optional application of smoothing *filters*. The default is not to smooth as smoothing filters can also be applied in a separate step afterwards (see [Filter SteeringCubes](#)).

The OK button starts the single machine, or multi-machine processing mode. For more information on single and Distributed Computing, open the help menu from the Distributed Computing window.

4.3 Filter

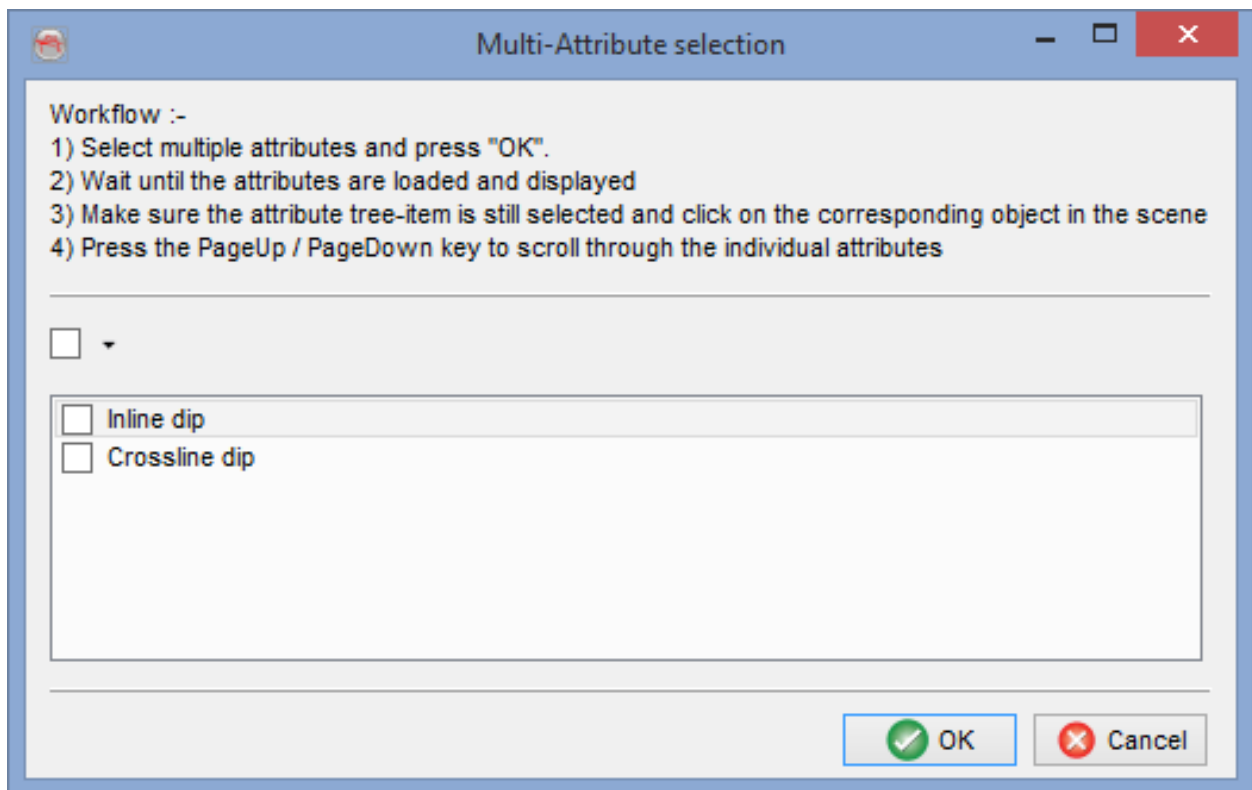
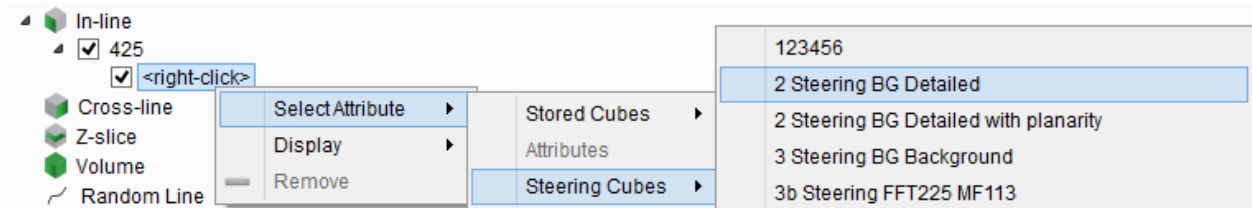
SteeringCubes can be smoothed by a median or average filter during creation, or post-creation. Typical filter settings to create a Background SteeringCube from a Detailed SteeringCube are: *Median* filter with *stepouts*: 3,3,3, or 5,5,5, or 3,3,7 (to emphasize smoothing in the vertical direction).



Select the input SteeringCube and, optionally, the *sub-volume* to process. The *median* filter *stepout* settings are defined in samples, regardless of sampling rate. *Batch jobs* can be processed on a single machine or distributed over multiple machines. For more information on single-machine and distributed processing, open the help menu from the Distributed Computing window.

4.4 Display SteeringCube

Once the SteeringCube has been calculated and filtered (optional), the results can be displayed in the OpendTect scene as a multi-component attribute, i.e on inline dip and/or cross-line dip.



4.5 Attributes with Steering

See the following chapters below.

4.5.1 Curvature Analysis

Curvature can be used as input to other attributes. The *Volume Statistics* attribute, in particular, proves to give very useful outputs. Just select the *Curvature* attribute as input and select the output statistic. For Fault Detection, '*Variance*' is a suitable output statistic.

A local surface is constructed at the evaluation point by following the dip information from the SteeringCube. The *Curvature* attribute specified in *Output* is then calculated according to Roberts (2001). In his Feb. 2001 *First Break* article, Roberts defines Curvature as a two-dimensional property of a curve that describes how bent a curve is at a particular point in the curve, i.e. how much the curve deviates from a straight line. The same concept is used to describe the Curvature of a surface. Curvature is measured on the curve which is the intersection between a plane and the surface. Since this can be done in numerous ways, there is an infinite number of Curvature attributes that can be calculated for any plane. The subset implemented here relates to the most useful subset of Curvatures that are defined by planes that are orthogonal to the surface and which are called normal Curvatures. A positive Curvature corresponds to an anticline and a negative Curvature indicates a syncline. A flat plane has zero Curvature. The application suggestions are from Roberts (First Break, Feb 2001).

Outputs

Mean Curvature: The average of any two orthogonal Normal Curvatures through a point on the surface is constant and is defined as the Mean Curvature. Minimum and Maximum Curvature (see below) are orthogonal surfaces, therefore the Mean

Curvature is also the sum of Minimum and Maximum Curvature divided by two. The Mean Curvature is not considered a very important attribute for visualization purposes, but it is used to derive some of the other attributes.

Gaussian Curvature: is defined as the product of the Minimum and Maximum Curvature. It is sometimes referred to as the Total Curvature. The Gaussian Curvature is not considered a very important attribute for visualization purposes, but it is used to derive some of the other attributes.

Maximum Curvature: From the infinite number of Normal Curvatures there exists one curve, which defines the largest absolute Curvature. This is called the *Maximum Curvature*. The plane in which Maximum Curvature is calculated is orthogonal to the plane of the Minimum Curvature. This attribute is very effective at delimiting faults and fault geometries.

Minimum Curvature: is the smallest absolute Curvature from the infinite number of Normal Curvatures that exist. The plane in which Minimum Curvature is calculated is orthogonal to the plane of the Maximum Curvature. The Minimum Curvature is often quite noisy, but it can sometimes be a good diagnostic in identifying fractured areas. Also, it is used to compute other Curvature attributes.

Most Positive Curvature: returns the most positive curvature from the infinite number of Normal Curvatures that exist. The attribute reveals faulting and lineaments. The magnitude of the lineaments is preserved but the shape information is lost. This attribute can be compared to first derivative based attributes (dip, edge, and azimuth).

Most Negative Curvature: returns the most negative curvature from the infinite number of Normal Curvatures that exist. The attribute reveals faulting and lineaments. The magnitude of the lineaments is preserved but the shape information is lost. This attribute can be compared to first derivative based attributes (dip, edge, and azimuth).

Shape Index (S_i): is a combination of Maximum and Minimum Curvature that describes the local morphology of the surface independent of scale. The attribute may reflect e.g. whether the surface corresponds to a bowl ($S_i=-1$), a valley ($S_i=-1/2$), ridge ($S_i=+1/2$), a dome ($S_i=1$) or it is flat ($S_i=0$). Because the attribute is not affected by the absolute magnitude of Curvature, it is reported to be useful for picking up subtle fault and surface lineaments, as well as other patterns.

Dip Curvature: returns the Curvature of the intersection with the plane that defines the dip direction of the surface. This plane is orthogonal to the plane for the Strike Curvature. This Curvature method tends to exaggerate local relief contained within the surface and can be used to enhance differential compacted features such as channeled sand bodies and debris flows.

Strike Curvature: (also known as Tangential Curvature) returns the Curvature of the intersection with the plane that defines the strike direction of the surface. This plane is orthogonal to the plane for the Dip Curvature. The attribute describes the shape of the surface. It is extensively used in terrain analysis, e.g. to study soil erosion and drainage patterns. The attribute reveals how shapes are connected, e.g. how ridges are connected to the flanks of anticlines. It may be useful for fluid-flow studies.

Contour Curvature: (also known as Plan Curvature) is not a Normal Curvature. It is very similar to the Strike Curvature and effectively represents the Curvature of the map contours associated with the surface. Contour Curvature values are not very well constrained, and large values can occur at the culmination of anticlines, synclines, ridges, and valleys.

Curvedness: describes the magnitude of Curvature of a surface independent of its shape. The attribute gives a general measure of the amount of Total Curvature within the surface.

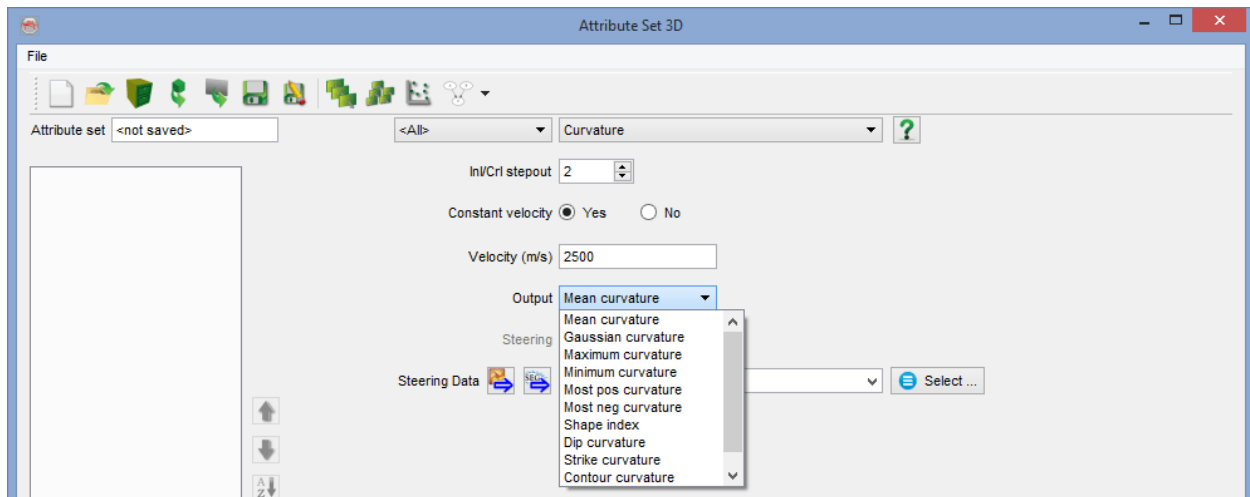
4.5.1.1 Volumetric Curvature

Returns *Curvature* properties from a SteeringCube

At every sample location, a 'virtual local horizon' is constructed on which the curvature is calculated (from the SteeringCube). This 'local horizon' is converted to depth in time surveys, which is why a velocity input is required. The output attribute is a 2D dataset or 3D volume that may be applied (projected) along horizons. Using a large stepout with a detailed SteeringCube in this attribute gives similar results as using a strongly filtered SteeringCube but with a small stepout. However the attribute runtime will be different.

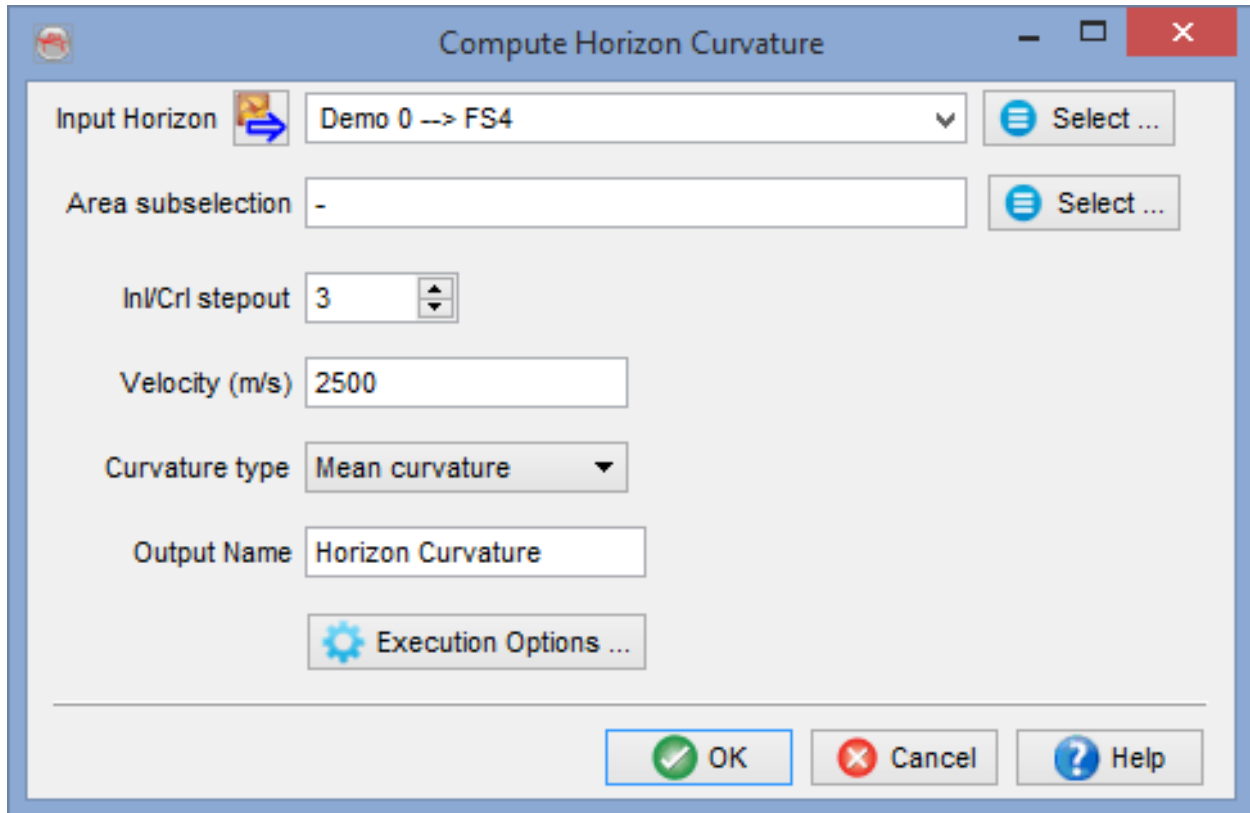
For further detail, including descriptions of the output options, below, please refer to [Curvature Analysis](#).

The *Curvature* options in OpenText are shown below:

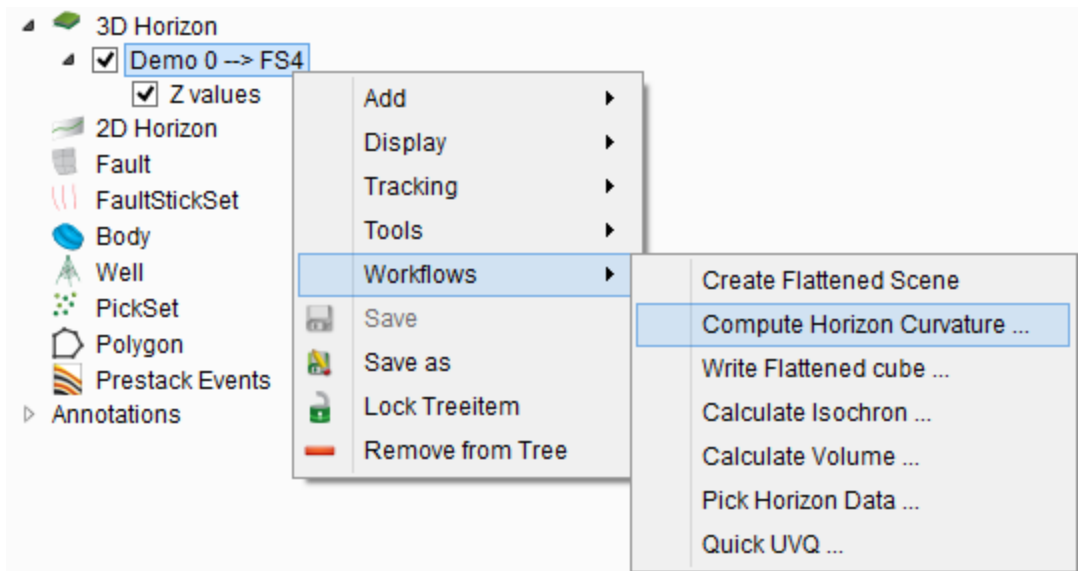


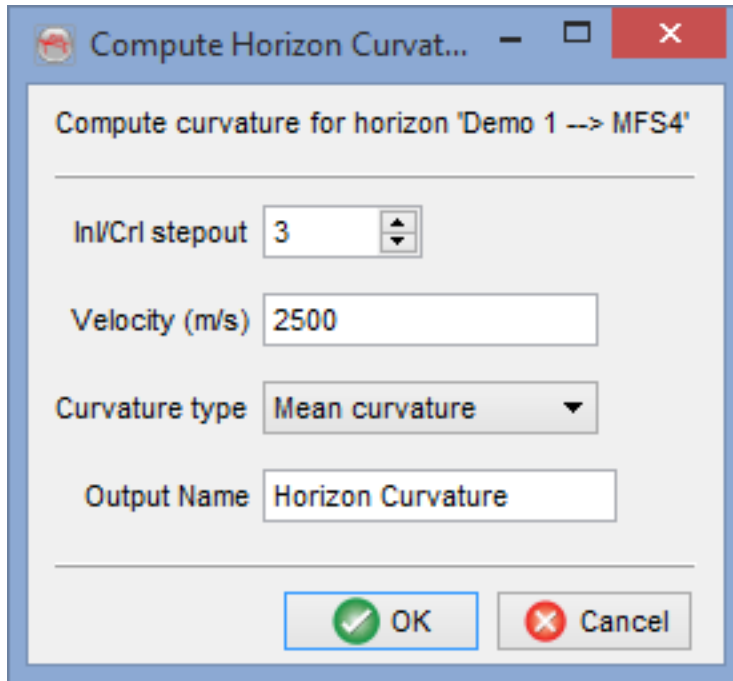
4.5.1.2 Horizon-Based Curvature

Horizon Curvature can be calculated along horizons either via *Processing > Create Horizon Output > Horizon Curvature...*:



or via the tree, directly onto the selected horizon:

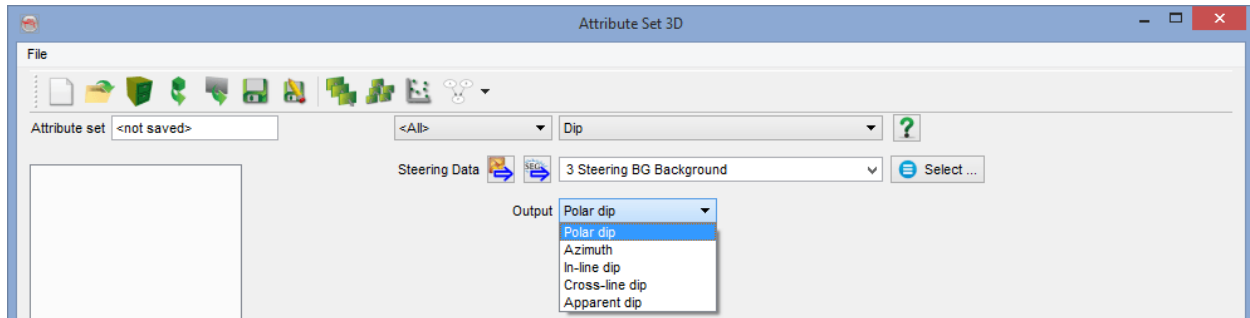




This input horizon is converted to depth in time surveys, which is why a velocity input is required. The output attribute is added on the horizon when using the tree menu, or as a grid in the database for that horizon when using the processing menu. Note that the first calculation is done within the current session, while it is done in batch from the processing menu (recommended for large horizons and/or large stepouts). For further detail, including descriptions of the output options, below, please refer to [Curvature Analysis](#).

4.5.2 Dip

Returns dip/azimuth from a SteeringCube



Description: The inline and crossline dips of a SteeringCube are transformed to the requested *Output* type (Polar dip, Azimuth, Inline dip, Crossline dip). When the SteeringCube was computed from seismic data sampled in time, the dips in a SteeringCube are apparent dips (slowness), and the returned attribute will also represent apparent dips. To compute real dips, please use the *dip angle* attribute.

Outputs

Polar dip: attribute converts extracted inline and crossline dips to polar dip, or true (geological) dip. The polar dip is the square root of the sum of (inline dip)² and (crossline dip)². The polar dip is thus larger or equal to zero.

Dips are given in μ seconds/meters in time surveys (millimeters/meters in depth survey), since they are a ratio between a vertical length and a horizontal distance. The dip angle attribute may be used to convert the polar dip output into degrees.



Along a 2D line the polar dip will return the absolute value of the line dip, the dip along the 2D line.

Azimuth: attribute returns the Azimuth of the dip direction in degrees ranging from -180 to +180. Positive azimuth is defined from the inline in the direction of increasing cross-line numbers. Azimuth = 0 indicates that the dip is dipping in the direction of increasing cross-line numbers. Azimuth = 90 indicates that the dip is dipping in the direction of increasing in-line numbers.

 *This output is not available in 2D.*


Inline dip: returns the dip along the inline direction as extracted by the steering algorithm. It is the first stored component of the SteeringCube, in $\mu\text{s}/\text{m}$ or mm/m .

 *This output is not available in 2D.*

Crossline dip: attribute returns the dip along the crossline direction as extracted by the steering algorithm. It is the second stored component of the SteeringCube, in $\mu\text{s}/\text{m}$ or mm/m .

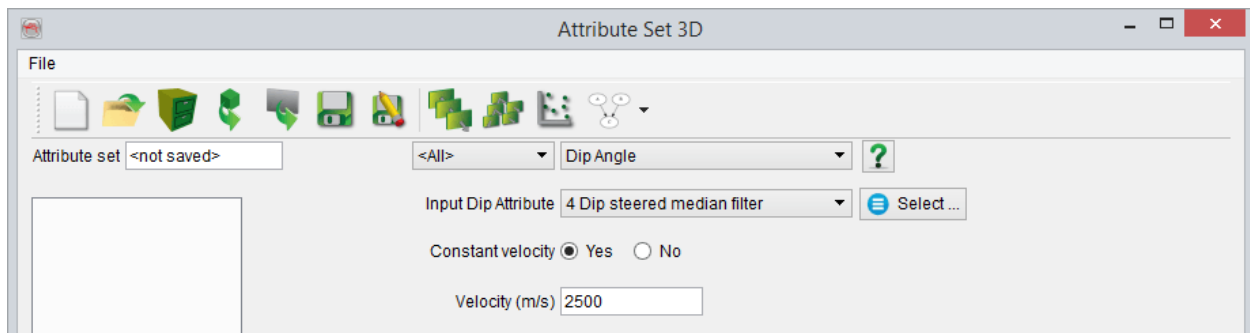
 *This output is not available in 2D (use line dip for 2D survey instead).*

Line dip: returns the dip along the 2D line. Computed steering lines are also two-component files with the first component equal to zero for all samples. The line dip is the second component.

 *This output is not available in 3D. (use inline dip and/or crossline dip instead)*

4.5.3 Dip Angle

The *dip angle* attribute converts the apparent dip (slowness-us/m) into degrees. It calculates the true dip from the apparent dip (slowness). If no velocity model is available, specify a constant *velocity* in the velocity field. The velocity should be given in meters per second (m/s).



The dip angle is calculated as follow.

$$\text{Dip Angle} = \text{Tan}^{-1} \left(\frac{\text{TWT} - \text{dip}}{2E + 06} \times V \right) \text{ in time domain}$$

$$\text{Dip Angle} = \text{Tan}^{-1} \left(\frac{Z - \text{dip}}{1000} \right) \text{ in depth domain}$$

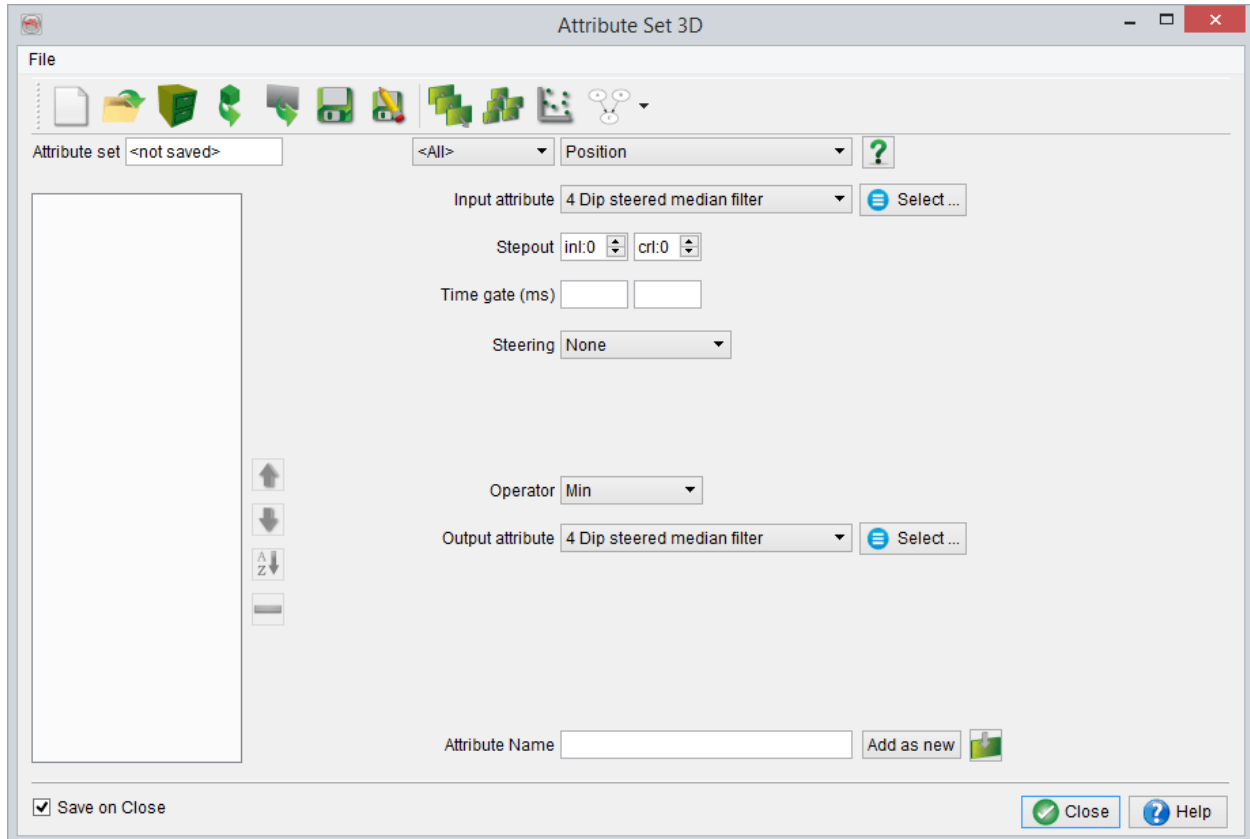
where TWT-dip is in micro seconds per meter and Z-dip is in millimeters per meter.



OpenTect uses SI-unit systems.

4.5.4 Position

Position attribute returns any attribute calculated at the location where an other attribute has its minimum, maximum, or median within a small volume.



Description: The input attribute defines the attribute that is used to determine the position at which the output attribute has to be calculated. The stepouts, time gate, and steering define the volume in which the input attribute is evaluated. The Operator determines which position is returned from this analysis; The position of the minimum, maximum or median of the input attribute. This position is the position at which the output attribute will be calculated.

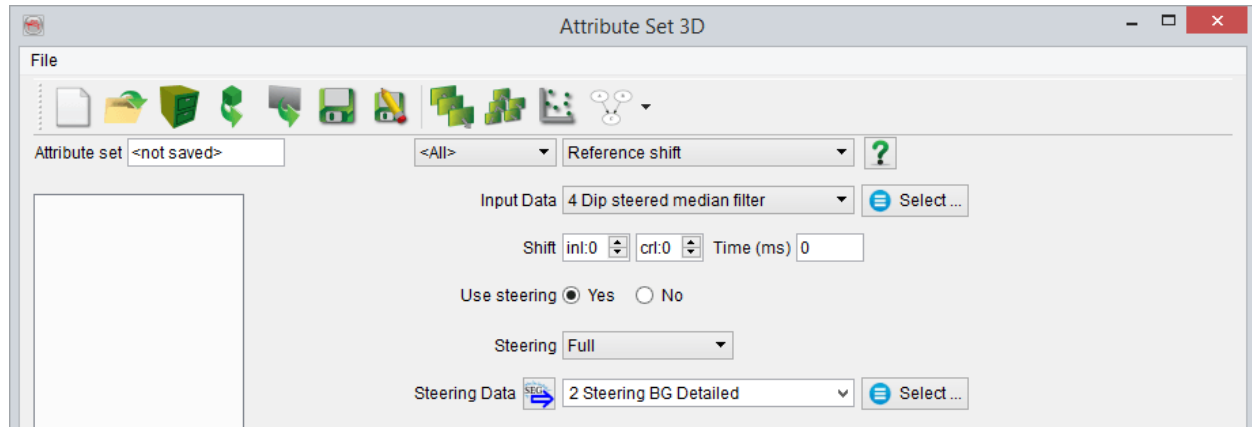
For example, one can determine where in a small volume the energy is minimal, and output the frequency at the location of this lowest energy. Another way of applying this attribute is to output Dip-steered filtered data at minimum values in areas where faults are present. In this way, the noise is reduced and the faults are sharpened.

Steering: In *Central Steering* the local dip information at the reference point is followed from trace to trace until all samples in the specified search radius are found. Central steering thus collects the input values along a dipping plane. In *Full steering* the dip information of the reference point is used only to find the position (and value) of the adjacent trace. The dip information at this new position is then used to find the position (and value) of the next trace and so on, until all samples in the specified search radius are found. Full steering thus corresponds to collecting the input values along a curved surface. In *Constant direction* the steering information is user-specified. The range of the *Apparent dip* is more than zero, and the *Azimuth* is defined from the inline, in the direction of increasing crossline-numbers. The azimuth ranges from -180 to 180 degrees.

In all forms of Steering the amplitude values at the intersection of trace and Steering surface are determined by interpolation.

4.5.5 Reference Shift

The *Reference shift* attribute moves the extraction position in 3D space.



Description: The *Input attribute* is extracted at the shifted position. Original reference (extraction) point has inline crossline coordinates (0,0). Relative number 1 means the next inline or crossline, respectively. The vertical shift is specified in *ms* using the *Time* option, or can be derived using *Steering*.

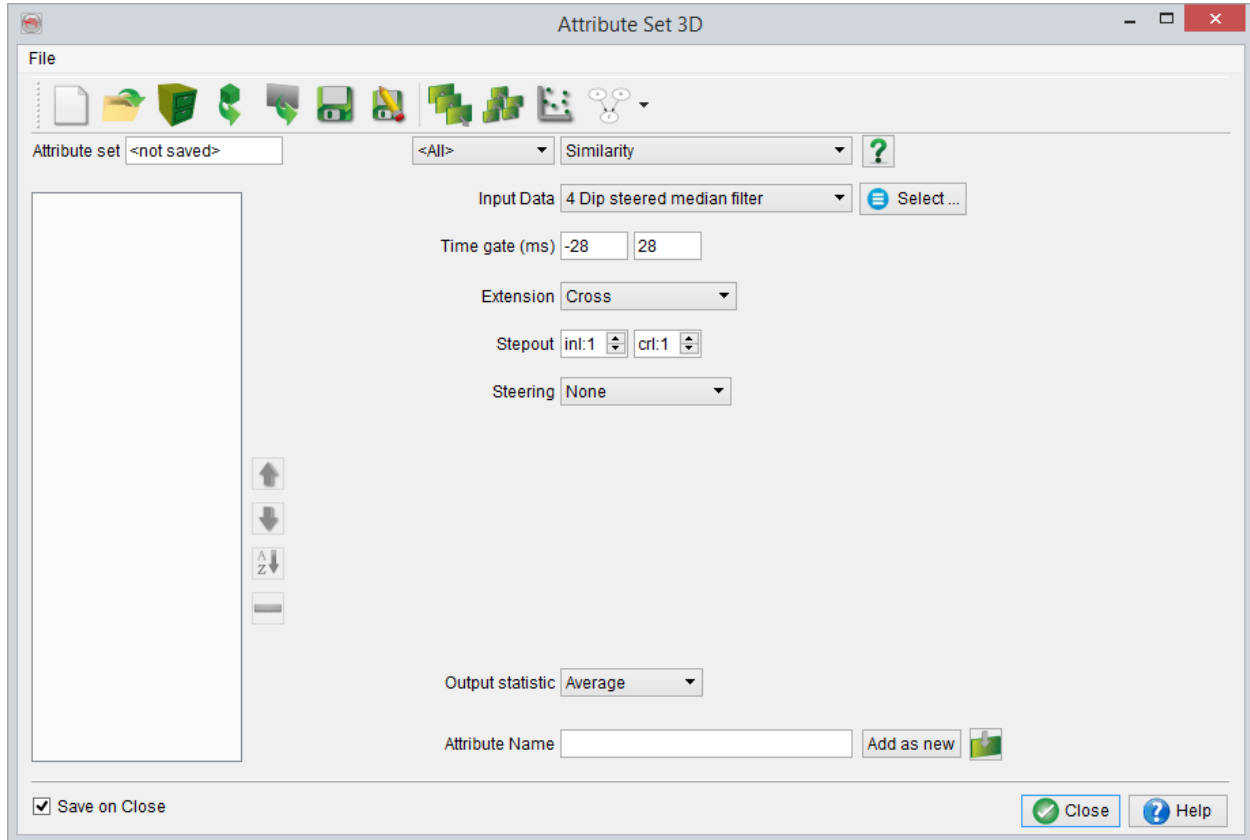
Steering is specified in one of the following ways:

- *None*: The reference position is not shifted vertically.
- *Central*: The reference position is vertically shifted according to the dip and azimuth information at (0,0,0) from the SteeringCube.
- *Full*: The reference position is vertically shifted according to the dip and azimuth from the SteeringCube, from trace to trace from the starting position (0,0,0) to the position specified at *Inl/Crl shift*.
- *Constant direction*: The reference position is vertically shifted according to a user defined *Apparent dip* and *Azimuth*.

Shifting the reference position is a form of *directivity* that is useful in multi-attribute analysis. For example, to highlight flat spots, one may consider training a neural network on attributes extracted in three horizontally aligned windows.

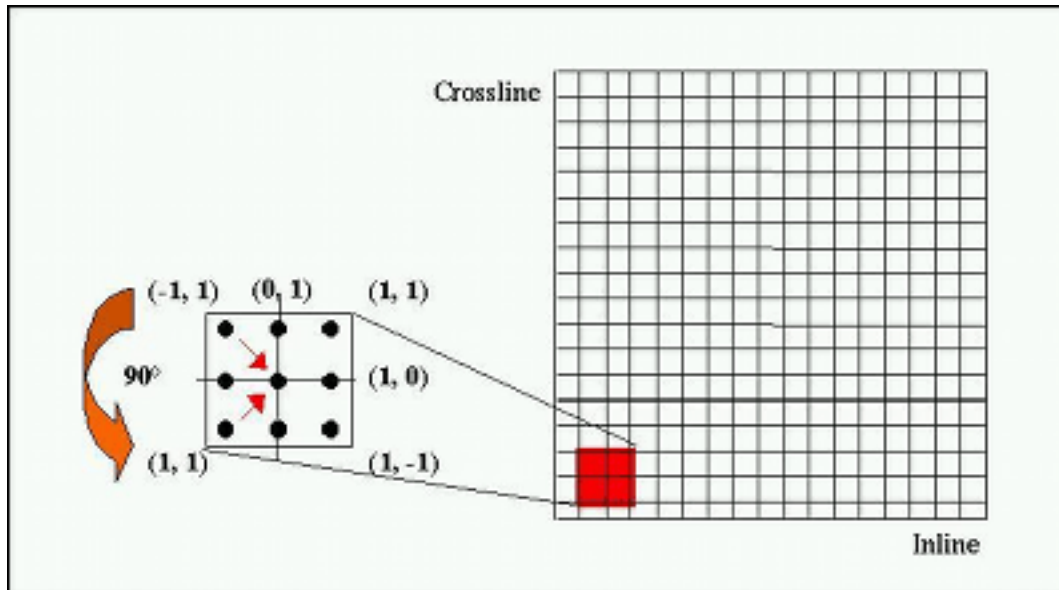
4.5.6 Similarity

The *Similarity* attribute returns trace-to-trace similarity properties



Description: *Similarity* is a form of "coherency" that expresses how much two or more trace segments look alike. A similarity of 1 means the trace segments are completely identical in waveform and amplitude, while a similarity of 0 means they are completely dis-similar. Consider the trace segments to be vectors in hyperspace, similarity is then defined as the euclidean distance between the vectors, normalized over the vector lengths. The trace segments as defined by the *Time gate* in ms and are found by *Steering* from the reference point to the specified *trace positions*. Positions are specified in

relative numbers (see figure). The *Extension* parameter determines how many trace pairs are used in the computation, see below.



Definition of trace positions relative to the reference point at (0,0)

Extension: With *None* specified, only the trace pairs specified in *trace positions* are used to compute the output. *Mirror at 90 degrees* and *Mirror at 180 degrees* means that two similarities are computed: for the specified trace pair and for the pair that is obtained by 90 or 180 degrees rotation, respectively. The average, minimum or maximum of these pairs as specified in *Output statistic* is returned. In *Full block* all possible trace pairs in the rectangle defined by *Inl/Crl stepout* are computed. The statistical property specified in *Output statistic* is returned.

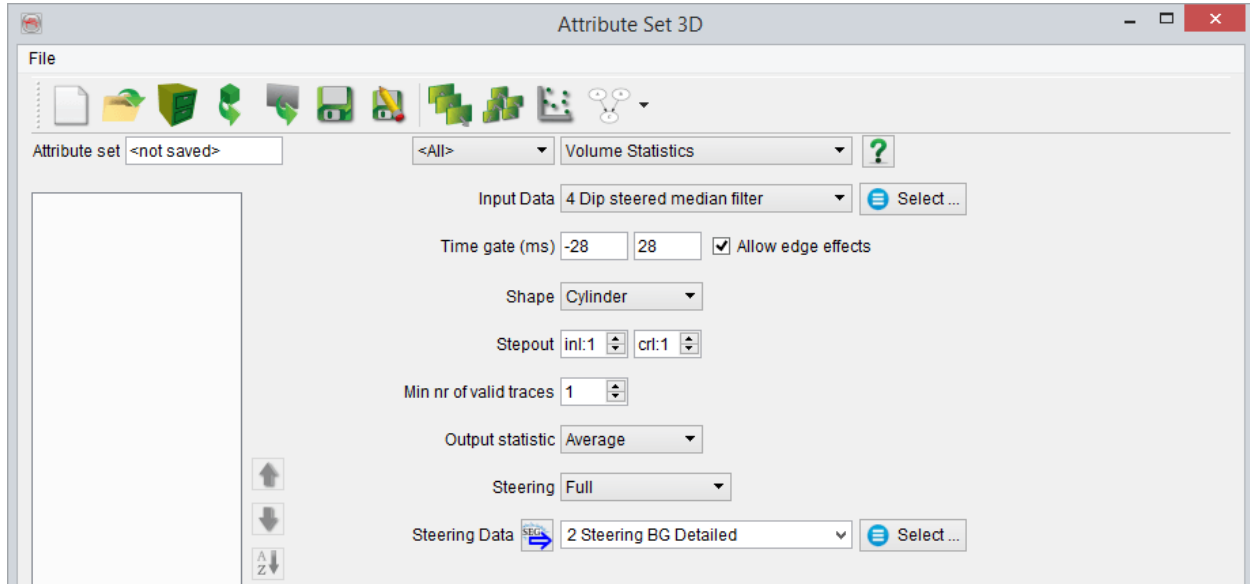
Steering: In *Central Steering* the local dip information at the reference point is followed from trace to trace until all samples in the specified search radius are found. Central Steering thus collects the input values along a dipping plane. In *Full steering* the dip information of the reference point is used only to find the position (and value) of the adjacent trace. The dip information at this new position is then used to find the position

(and value) of the next trace and so on, until all samples in the specified search radius are found. Full steering thus corresponds to collecting the input values along a curved surface. In *Constant direction* the steering information is user-specified. The range of the *Apparent dip* is more than zero, and the *Azimuth* is defined from the inline, in the direction of increasing crossline-numbers. The azimuth ranges from -180 to 180 degrees.

In all forms of steering the amplitude values at the intersection of trace and steering surface are determined by interpolation.

4.5.7 Volume Statistics

The *Volume Statistics* attribute returns statistical properties from a small sub-volume



Description: The statistical property specified in *Output statistic* is returned. The input values are collected from a cube (rectangle) or cylinder (ellipsoid) around the reference point defined by parameters: *Time gate*, *Shape* and *Inl/Crl stepout*. Optionally *Steering* is used to obtain the trace segments of the input sub-volume.

Steering: In *Central Steering*, the local dip information at the reference point is followed from trace to trace until all samples in the specified search radius are found. Central Steering thus collects the input values along a dipping plane. In *Full steering* the dip information of the reference point is used only to find the position (and value) of the adjacent trace. The dip information at this new position is then used to find the position (and value) of the next trace and so on, until all samples in the specified search radius are found. Full steering thus corresponds to collecting the input values along a curved surface. In *Constant direction* the steering information is user-specified. The range of

the *Apparent dip* is more than zero, and the *Azimuth* is defined from the inline, in the direction of increasing crossline-numbers. The azimuth ranges from -180 to 180 degrees.

In all forms of steering, the amplitude values at the intersection of trace and steering surface are determined by interpolation.

4.5.8 Perpendicular Dip Extractor

This attribute works similarly to the [volume statistics](#) attribute, with one very important exception:

The time gate used for the extraction of the amplitudes is not vertical, but is perpendicular to the layer dip. This layer dip is read from a SteeringCube, which is why it is not possible to toggle off the full steering for this attribute.

Using this attribute instead of the standard volume statistics will most often return more "correct" amplitudes, because the extraction is done in the same geological layer with a real constant thickness, whereas a vertical time gate would represent an apparent thickness, varying through the survey.

4.6 Benchmark SteeringCube Creation

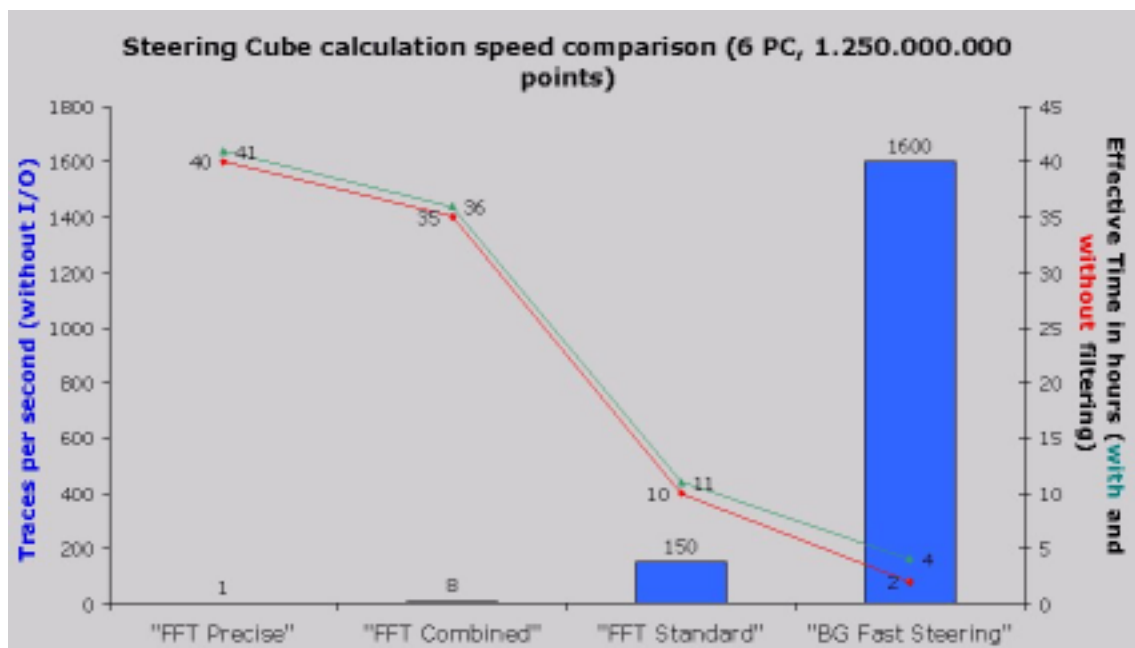
In this chapter the quality and speed of the available *steering algorithms* are compared in order to assist in your choice of the correct SteeringCube algorithm. First, the results of a test on a standard data set are presented to give an indication of the speed of SteeringCube calculation versus the chosen algorithm and *calculation cube size* parameter (where applicable). Then, a visual quality check on one of the inlines is presented. This is done by checking the crossline dip, which is directly extracted from the SteeringCube, and by reviewing the steered similarity, which uses information derived from the SteeringCube.

The final output quality is determined not only according to the steering algorithm. The size of the calculation cube also plays an important role. Next, the application of additional dip-steered median filter influences on the output quality are presented and finally, the use of dip limit.

4.6.1 Speed vs Algorithm and Calculation Cube Size

In *Figure 1* the relative calculation speed is displayed for the different algorithms. The test was done on a cluster of 6 computers using distributed computing. The input cube has 817811 traces. Each trace has 1551 samples with step of 4 ms, making a total of 6200 ms. Calculation speed is measured in traces per second.

Figure 1. Comparison of relative speed of the different steering algorithms.



4.6.2 Visual Quality Check

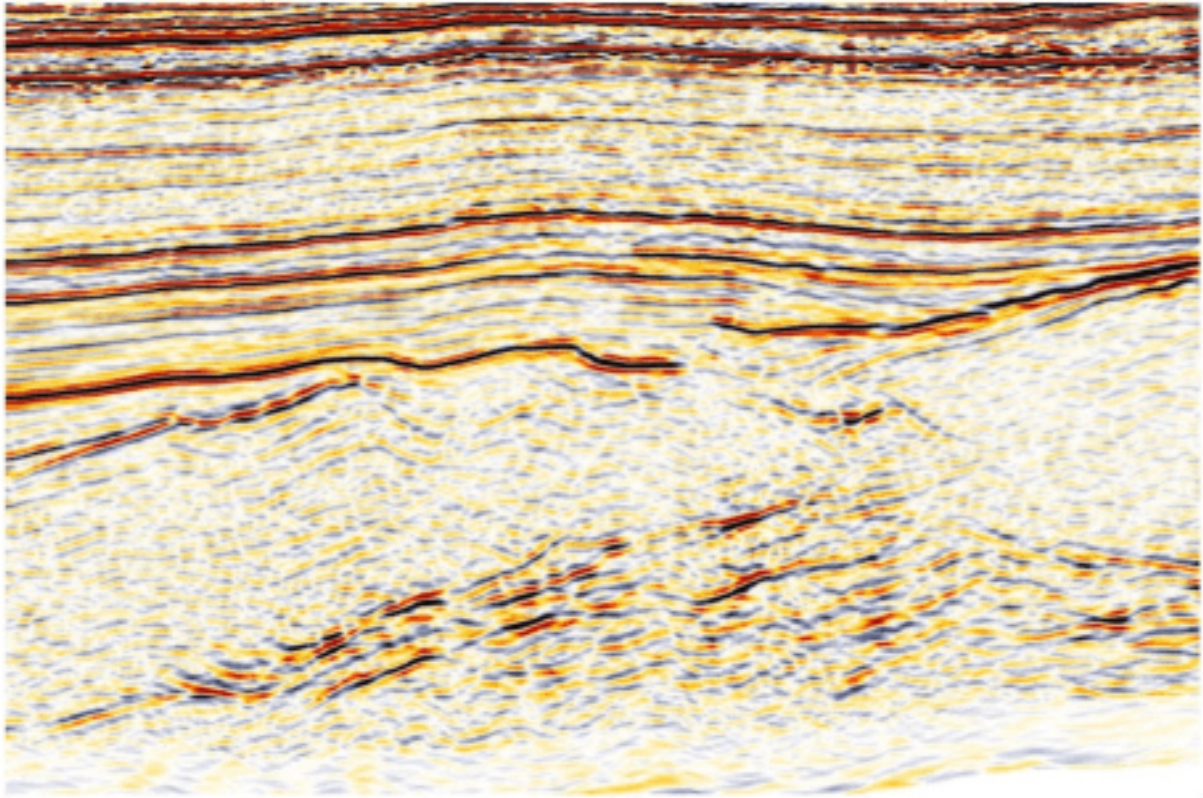
In the following sections, the crossline dip component of a SteeringCube created with the different algorithms is presented. The figures contain the results from the *Precise FFT* algorithm, the *Combined FFT* algorithm, the *Standard FFT* algorithm and the *BG Fast Steering* algorithm (for technical details see [Create SteeringCube](#)) for a cube size of 3.

Notice that the nomenclature convention used is the following:

- The FFT precise is called FFT+.
- The FFT combined is called FFT++.
- The standard steering algorithm is called FFT+++.
- The calculation cube size is specified after the algorithm (e.g. FFT7, BG5, etc.).
- The limit of dip is called maxdipXXX, where XXX is the limit in $\hat{\mu}\text{s/m}$.
- Additional filtering are called medXYZ, where X is the inline stepout, Y the crossline stepout and Z the sample stepout.

The inline itself is displayed in *Figure 2* for reference. The inline was selected because many geological and seismic features are visible, which enable evaluation of the performance of the algorithm in different environments.

Figure 2. Seismic data of the test inline.



4.6.3 Crossline Dip Attribute

The crossline dip is one of the two SteeringCube components (together with inline dip) and is related to the dips projected in the crossline direction. In *Figure 3* to *Figure 5* the crossline dip is displayed for the algorithms mentioned in *Figure 1* with cube sizes of 3, 5, and 7.

Figure 3. Crossline dip with (calculation cube size=3): precise steering (A), combined steering (B), standard steering (C) and BG steering (D).

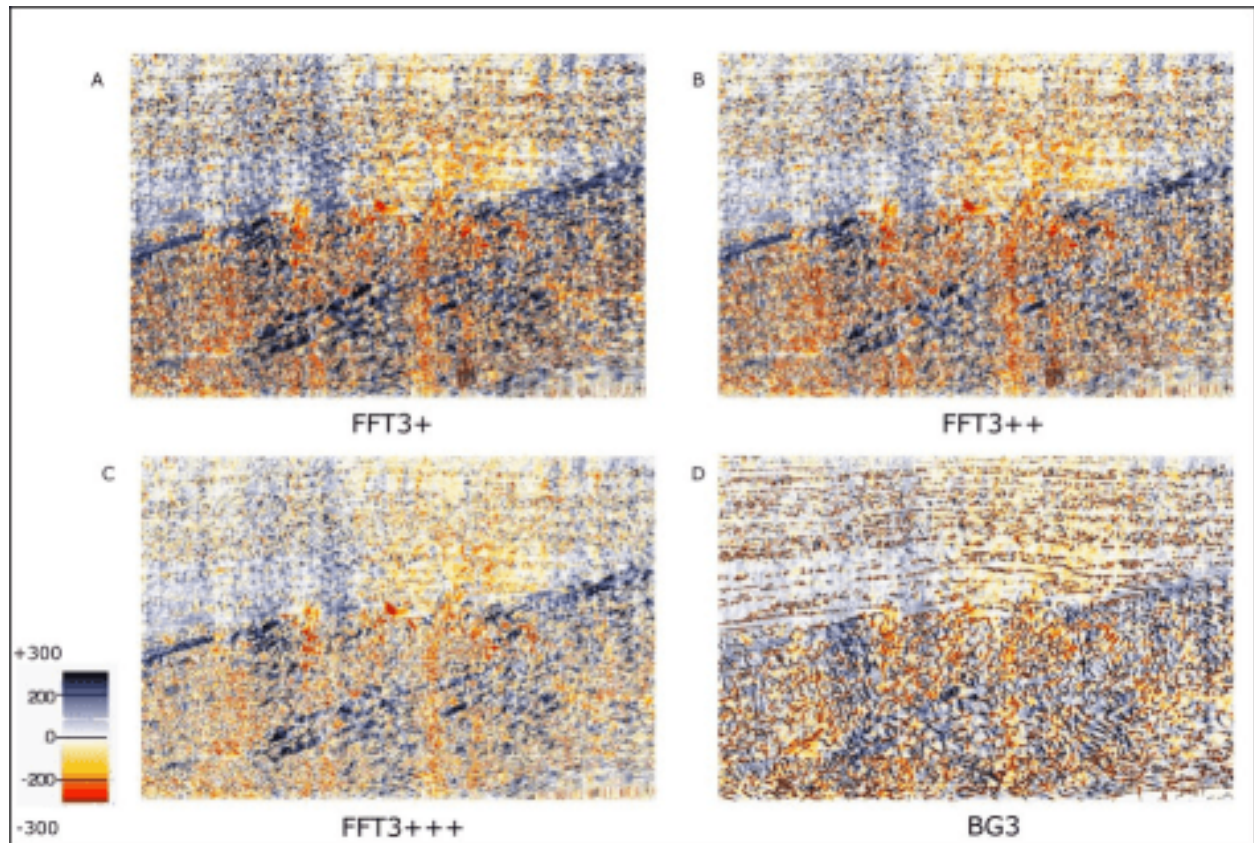


Figure 4. Crossline dip with (calculation cube size=5): precise steering (A), combined steering (B), standard steering (C) and BG steering (D).

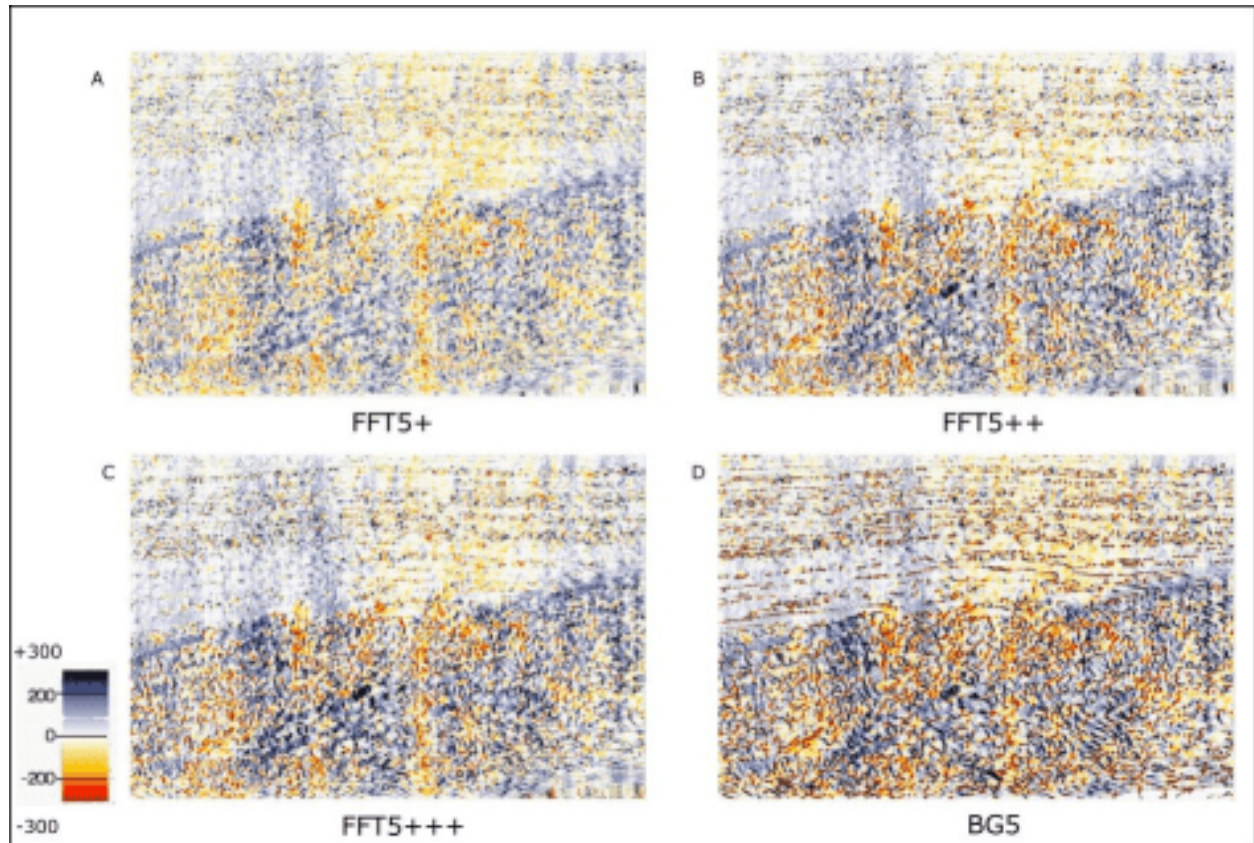
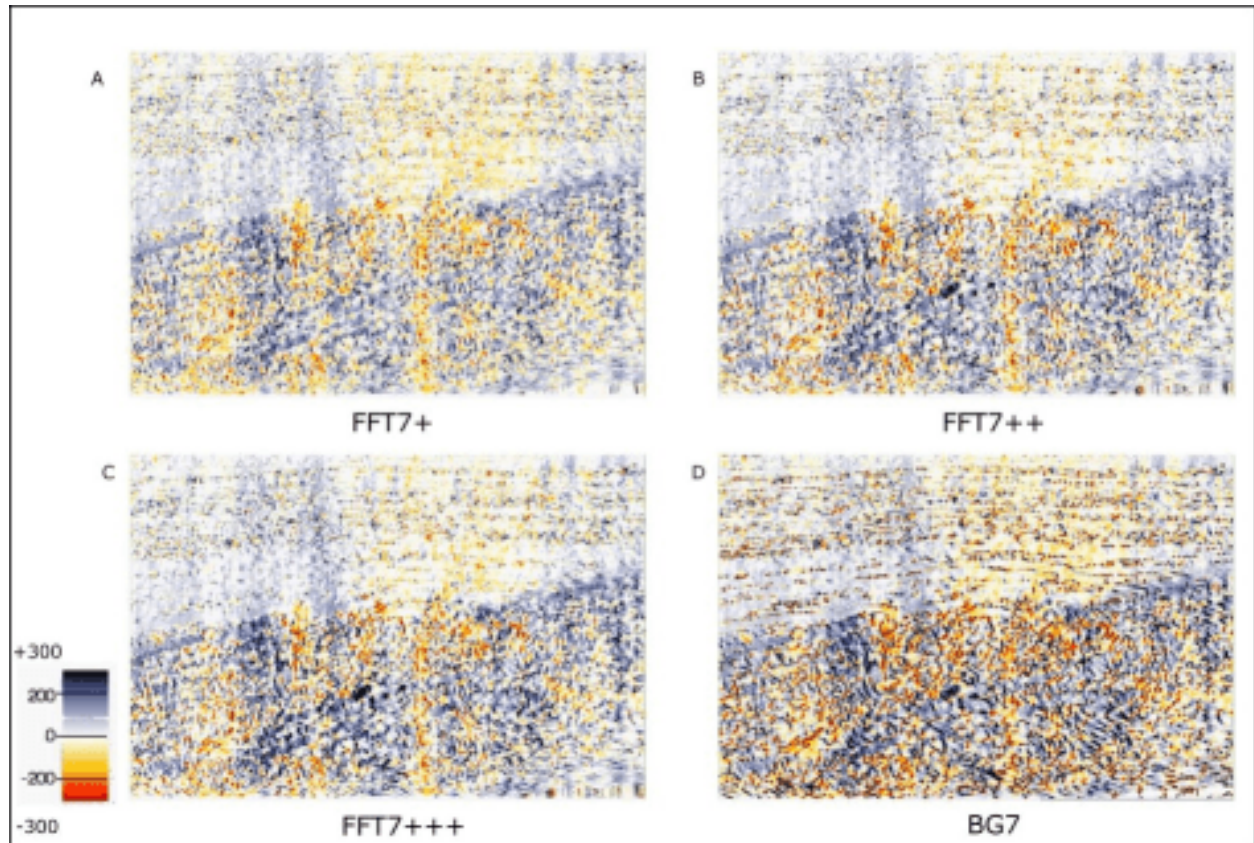


Figure 5. Crossline dip with (calculation cube size=7): precise steering (A), combined steering (B), standard steering (C) and BG steering (D).



From *Figure 3* to *Figure 5* it is notable that there is no significant difference between the combined steering algorithm- (FFT++) and precise steering algorithm (FFT+) speed and accuracy. The standard steering algorithm (FFT+++) is fast but apparently often produces erroneous results in high dip areas. In order to avoid getting a noisy SteeringCube the calculation cube-size of the FFT algorithms has to be at least 5 or 7. The latter is the default setting. The BG algorithm has a different behaviour: a cube size of 3 seems to be sufficient, but the raw SteeringCube is useless and a median filtering appears to be mandatory.

4.6.4 Filtering of the SteeringCubes

Figure 6 and *Figure 7* show the results after applying a median filter with different step-outs to the steering cubes. It is apparent that no lateral filtering and only a small vertical filtering, gives the best result with the FFT algorithm.

The BG steering needs to be filtered in the lateral and vertical direction. The best results were obtained with median filter with step-outs 1 1 3. After filtering, the outputs of the precise steering with cube size 7 and median filtered with step-out of 2 only in the vertical direction (FFT7+ med002) and Fast BG steering median filtered with the step-outs 1 1 and 3 (BG3 med113) are very similar in accuracy. However, the latter is produced 10 times faster.

Figure 6. Filtering of FFT+ using calculation cube size = 7: raw (A), median filter with step outs 0 0 2 (B), median filter with step-outs 0 0 4 (C), median filter with step outs 1 1 2 (D).

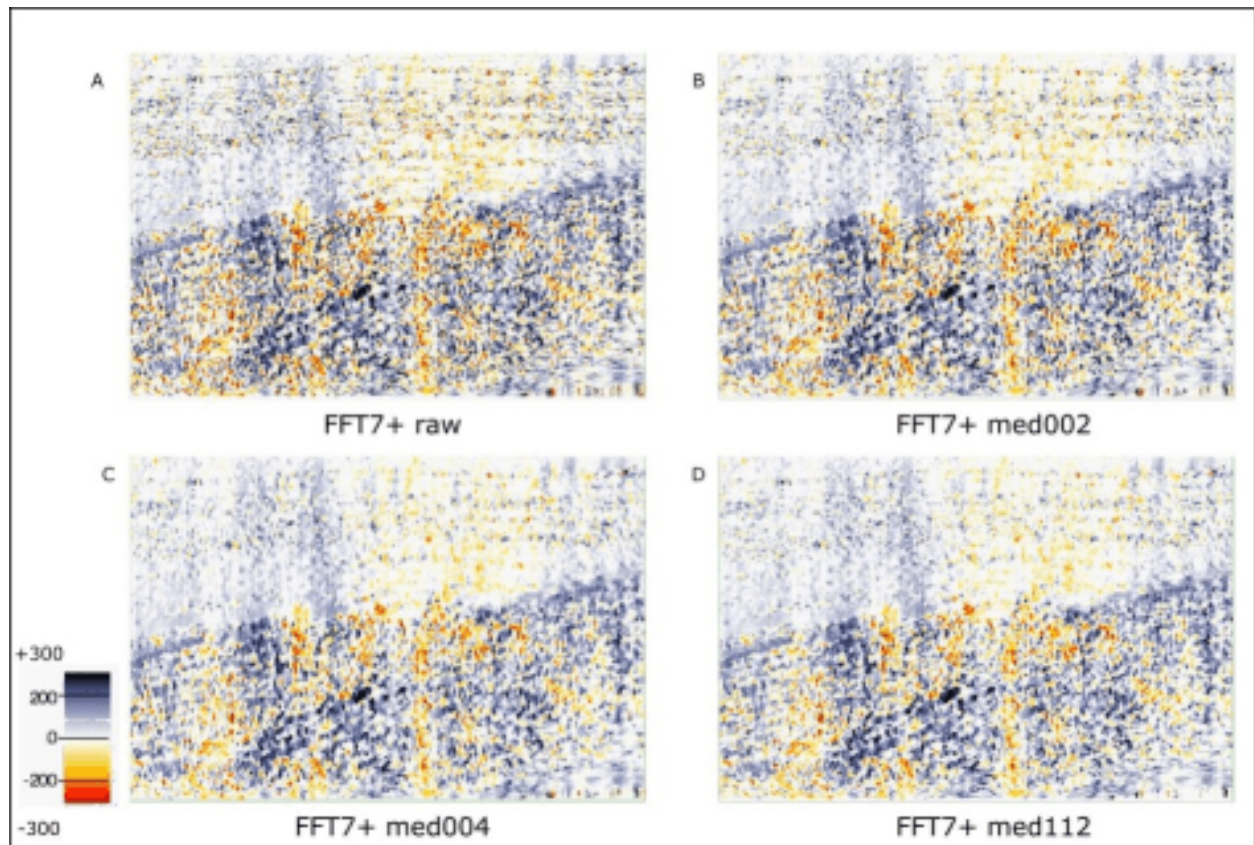


Figure 7. Filtering of BG steering using calculation cube size=3: raw (A), median filter with step-outs 0 0 2 (B), median filter with step outs 1 1 1 (C), median filter with step outs 1 1 3 (D).

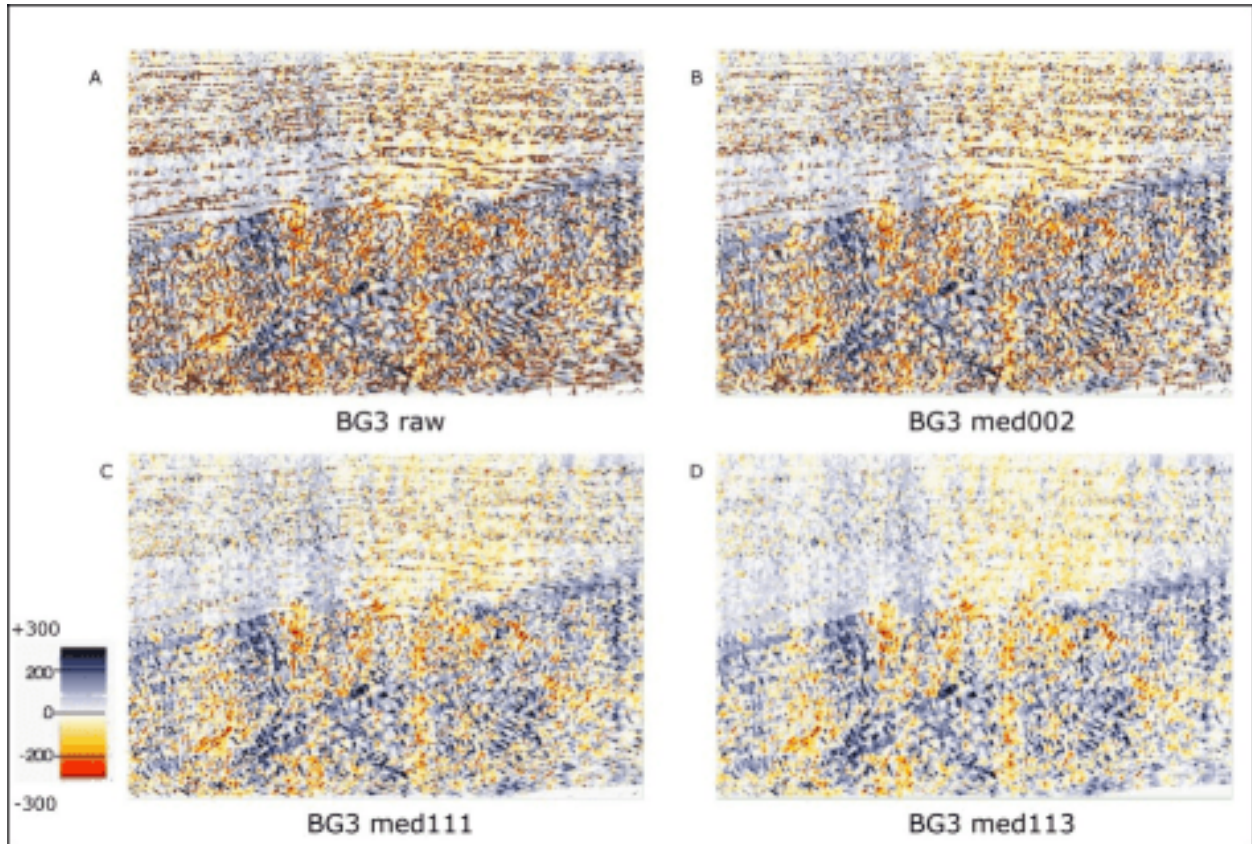
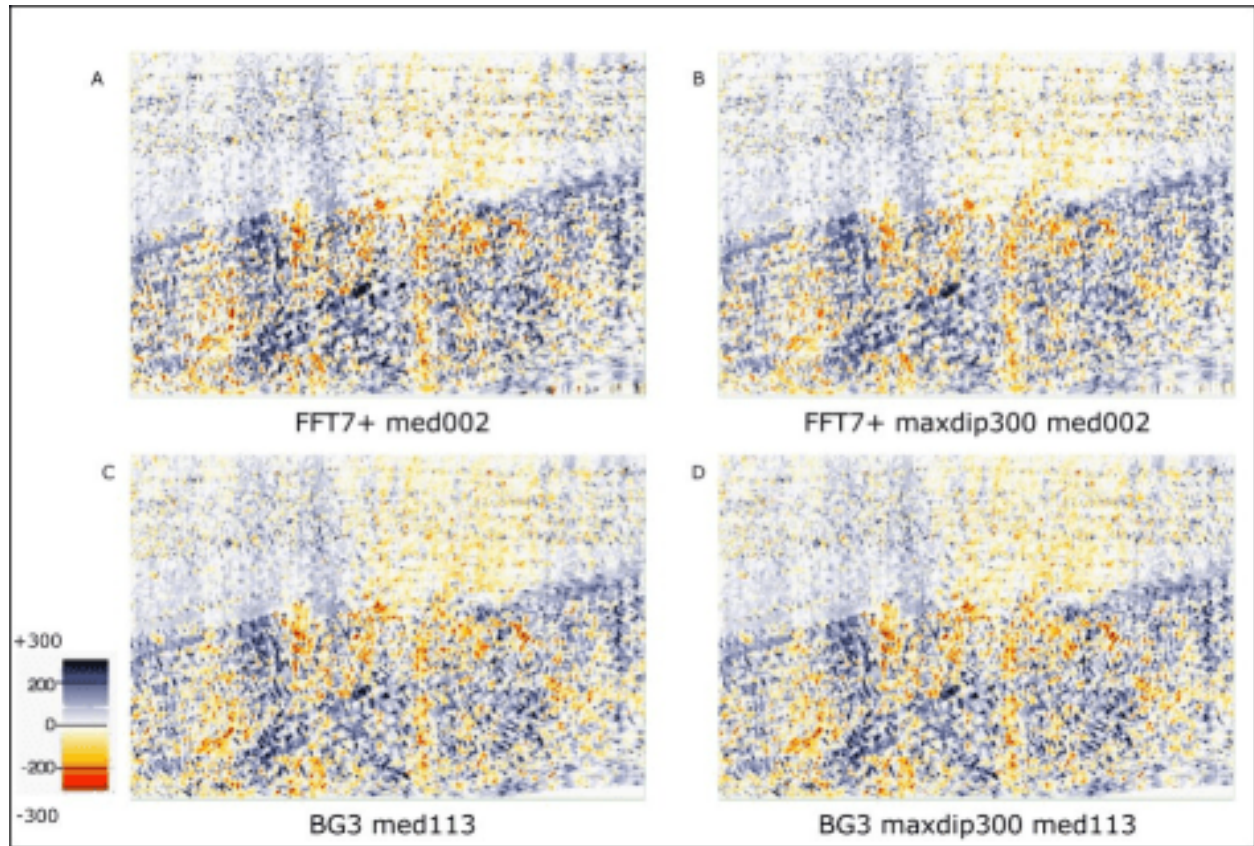


Figure 8 shows that adding a dip limit during the processing does not affect the speed of the algorithms. The final result is strictly identical when the dip is lower than the limit L , and the extreme values are rounded toward L . Using a limit requires a priori knowledge and is in the end a choice of the interpreter.

Figure 8. Filtering of FFT+ using calculation cube size=7: median filter with step-outs 0 0 2 (A) median filter with step-outs 0 0 2 and maximum dip of 300 (B), and filtering of BG steering using calculation cube size=3: median filter

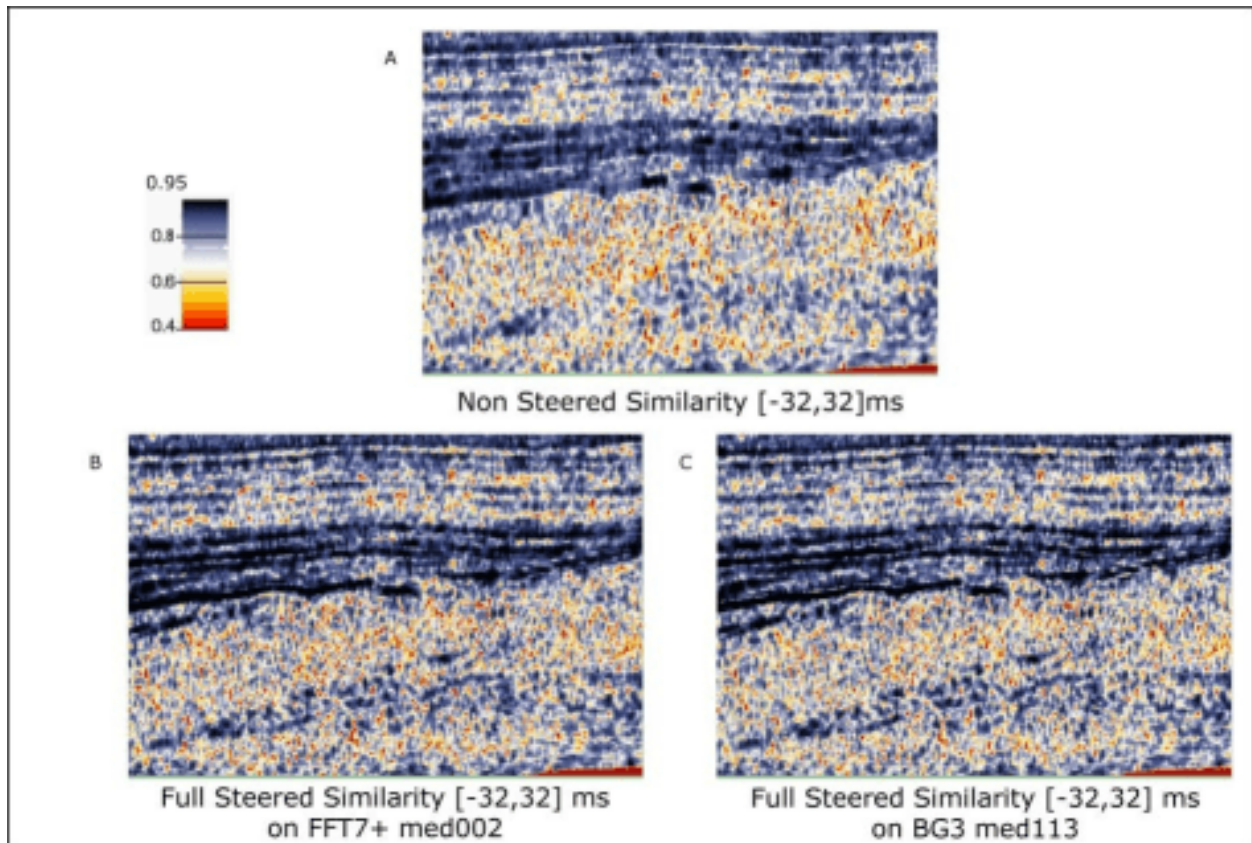
with step-outs 1 1 3 (C), median filter with step-outs 1 1 3 and maximum dip of 300 (D).



4.6.5 Steered Similarity Attribute

Figure 9 displays the Similarity attribute for the algorithms FFT+ and BG. As an extra reference, the non-steered similarity is added. All figures were calculated with the time gate $[-32,32]$ ms.

Figure 9. Positive curvature using perfect FFT (time gate from $[-32,32]$ ms): no steering (A), precise steering with cube size =7 (B), BG steering with cube size =3 (C).



4.6.6 Choosing a Steering Algorithm

Different Steering algorithms are available. The *precise FFT* algorithm yields an almost perfect SteeringCube at the costs of considerably longer calculation times. The *BG Fast Steering* algorithm seems to fit 95% of the situations and is very fast. dGB recommends the use of this algorithm, using its default calculation cube size of 3 and additional median filtering 113.

Although depending on the geology, data quality, available computation time and purpose other choices can be made. A number of examples are presented:

- For a dataset of good quality with only small and low variance in dips, the *BG Fast Steering* method performs well enough. With a median filtering with step-out 1 1 3 applied the result is already very acceptable.
- For a poor quality dataset, one of the *FFT* algorithms should be used because the *BG Fast Steering* algorithm is sensitive to noise and will produce too much outliers. Minor vertical filtering might also help improve the results.
- For detailed studies at target level, the *precise FFT* algorithm can be considered for a sub-volume within the area of interest.
- When creating chrono-stratigraphy, the *Steering* is optimal because it looks for similar events (min, max) on neighboring traces in inline/crossline directions.

It is always possible to go back and spend much more time in producing a SteeringCube using the *FFT precise* algorithm.

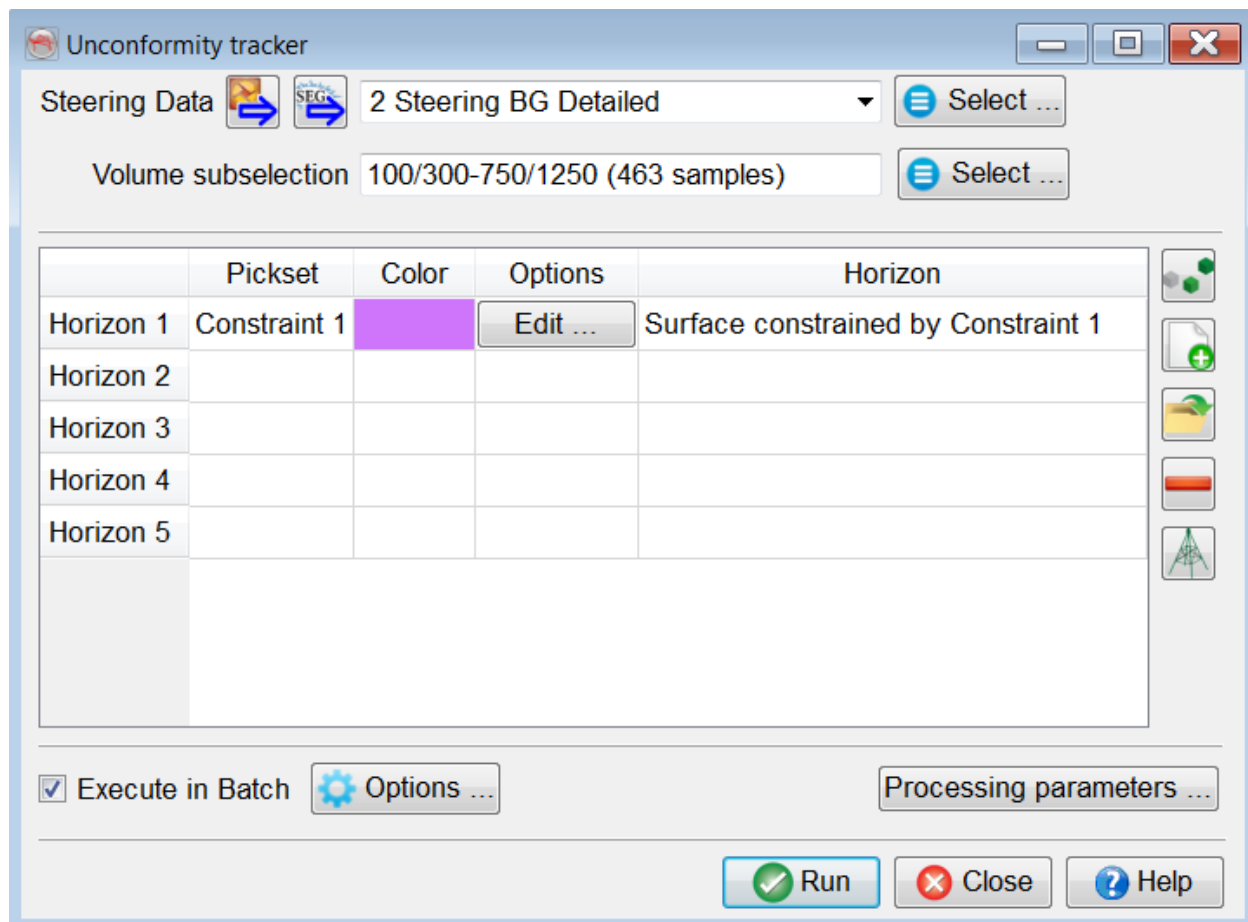
4.7 Track Using Inversion

This seismic horizon tracker generates one or more horizons from a Steering Cube (dip volume). The tracker uses a constrained, inversion-based algorithm that globally minimizes the error between horizon dips and seismic dips (Wu and Hale, 2015).

A confidence weight volume is recommended for faulted intervals. A good confidence weight volume is given by the “Planarity” attribute. Planarity is computed in the “Faults & Fractures” plugin, It is a measure of “local flatness” of a seismic event. When used as a confidence weight volume in the unconformity tracker it assigns low weights at fault positions and high weights at good reflectors. A flipped discontinuity attribute like similarity, semblance, or Thinned Fault Likelihood can be considered as alternatives to planarity.

The unconformity tracker is a general-purpose horizon tracker that can be used among others to:

- Map unconformities (events that change laterally and are therefore difficult to map using conventional amplitude / similarity trackers)..
- Map horizons at intra-reservoir scale levels that tie exactly at given well markers.
- Fast-track mapping of seismic events, e.g. for constructing a structural framework, to constrain a HorizonCube, or to build low-frequency models for seismic inversion.



Main window for the Unconformity Tracker

The workflow can be started in three ways:


- Insert a new horizon in the table for picking
- Click on the “Pick Seeds” icon and manually pick a set of locations.
- Click on the “Load existing Pick Set” icon to open an existing pick set.
- Click on the “From Wells” icon and select one or several marker sets and the wells.

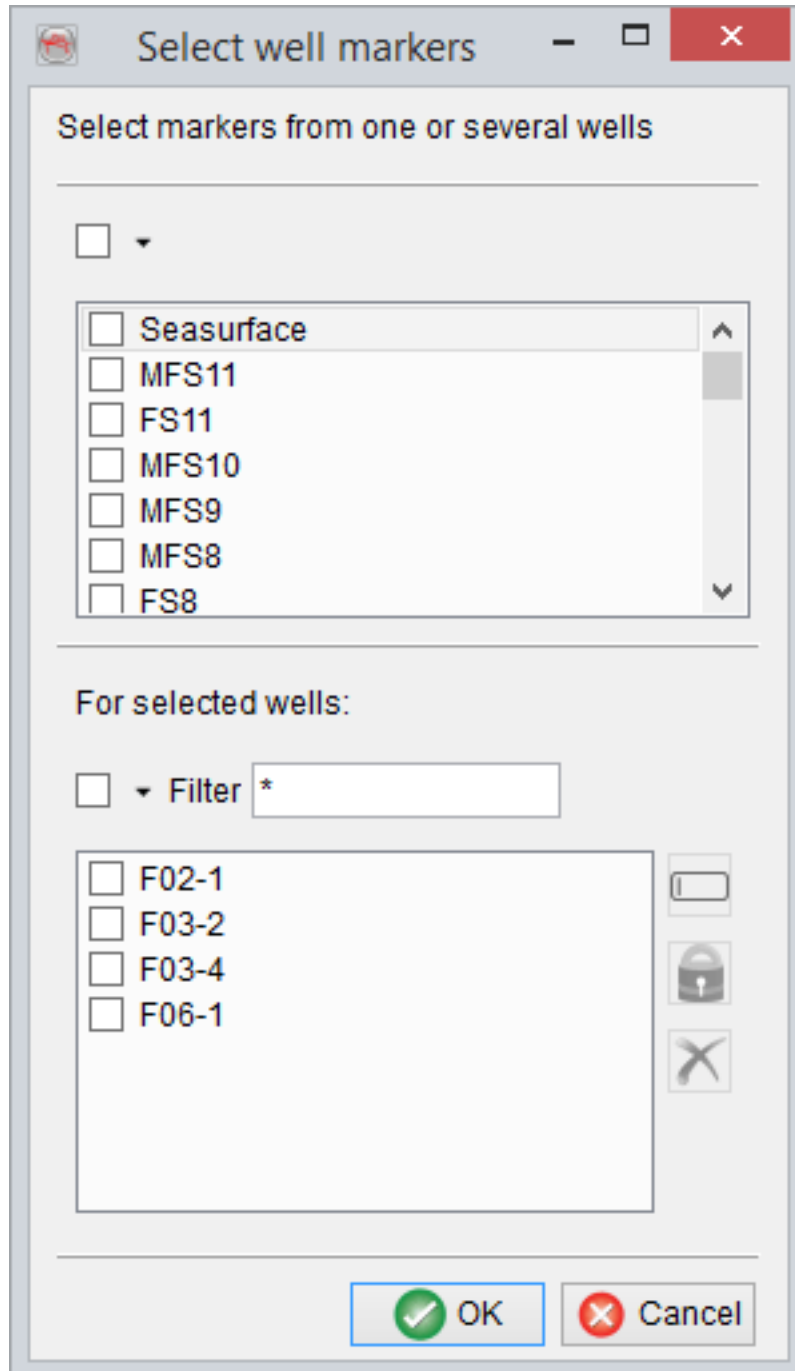
Seeds can be added or removed thereafter regardless of the origin of the pickset, either interactively or using the table editor under the Edit options. At least one seed is required for each horizon to be tracked using this tracker.

Inputs

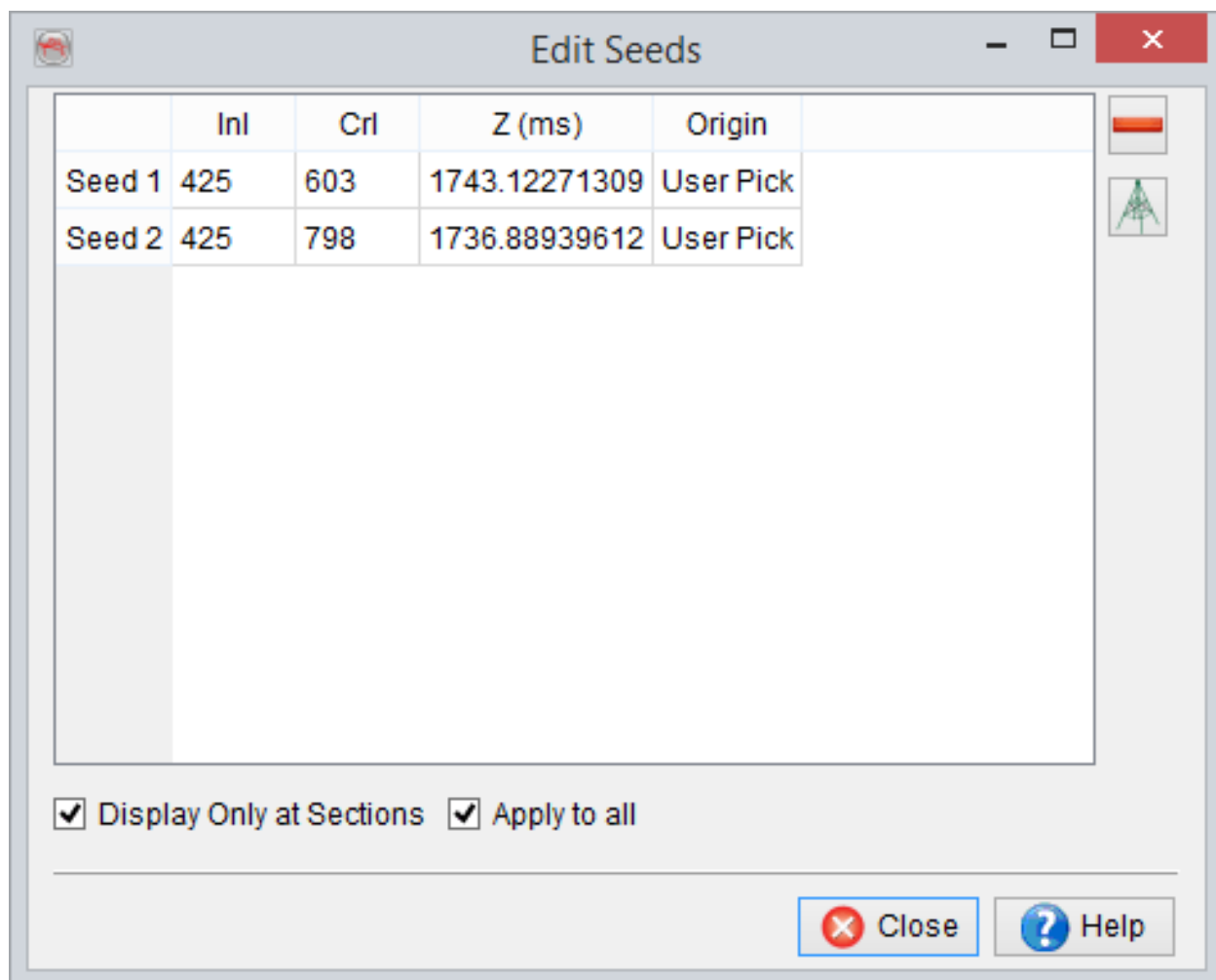
- A pre-processed SteeringCube is mandatory. This SteeringCube must be filtered to have less noise. A typical Detailed SteeringCube should be a good starting point.
- Set of locations (from well makers and/or manually picked positions).
- Optional: confidence weight volume (planarity, flipped Thinned Fault Likelihood, ...).

Adding Well Markers

Multiple markers from multiple wells can be selected in the Select well markers window . Each selected marker adds a row to the table in the main window. The set of (manually-picked / markers) locations can be QC-ed and edited by pressing the Edit button. To add more picks, select the relevant row, press the Pick Seed button and manually pick positions to add.

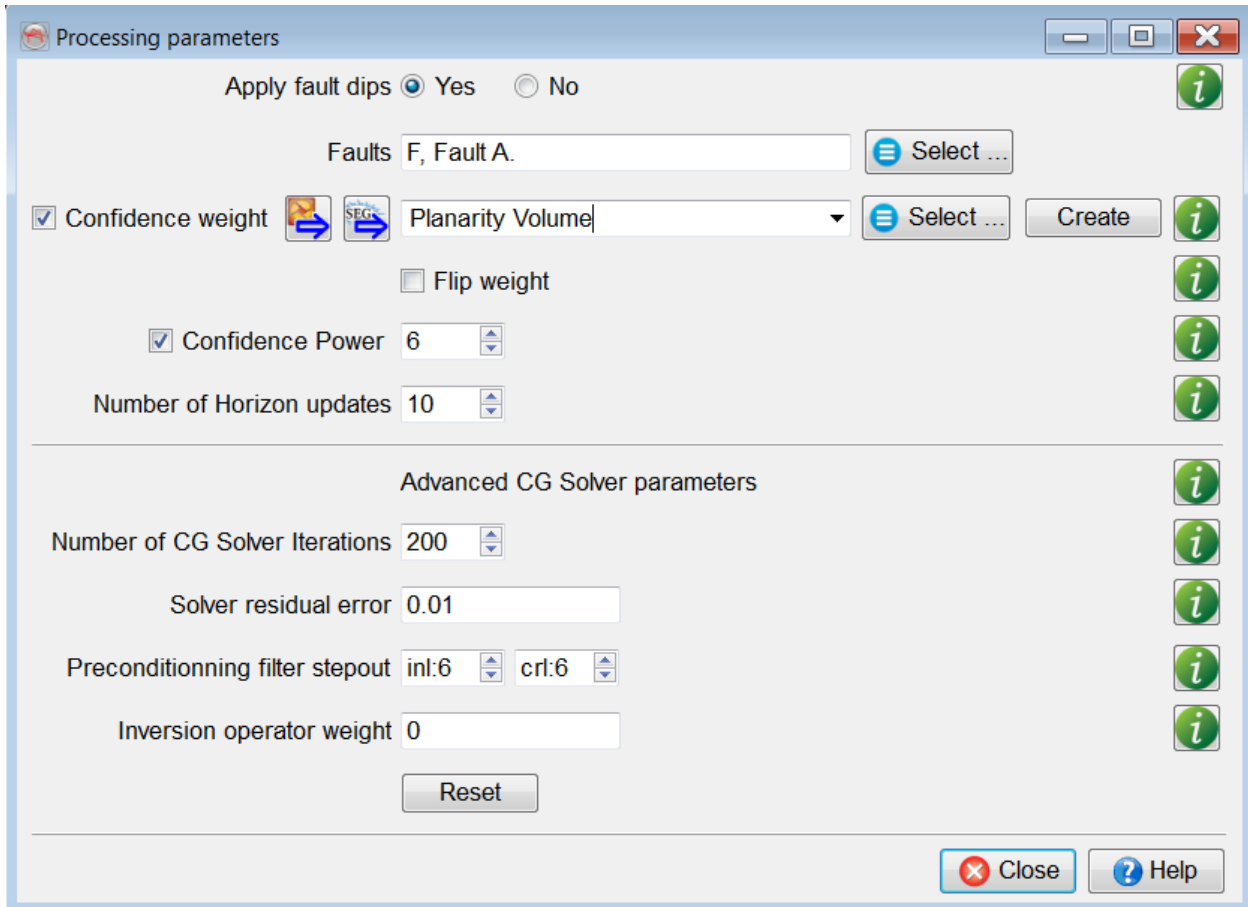


Selection of well markers is done here. Multiple markers can be selected. Each marker will create a horizon.



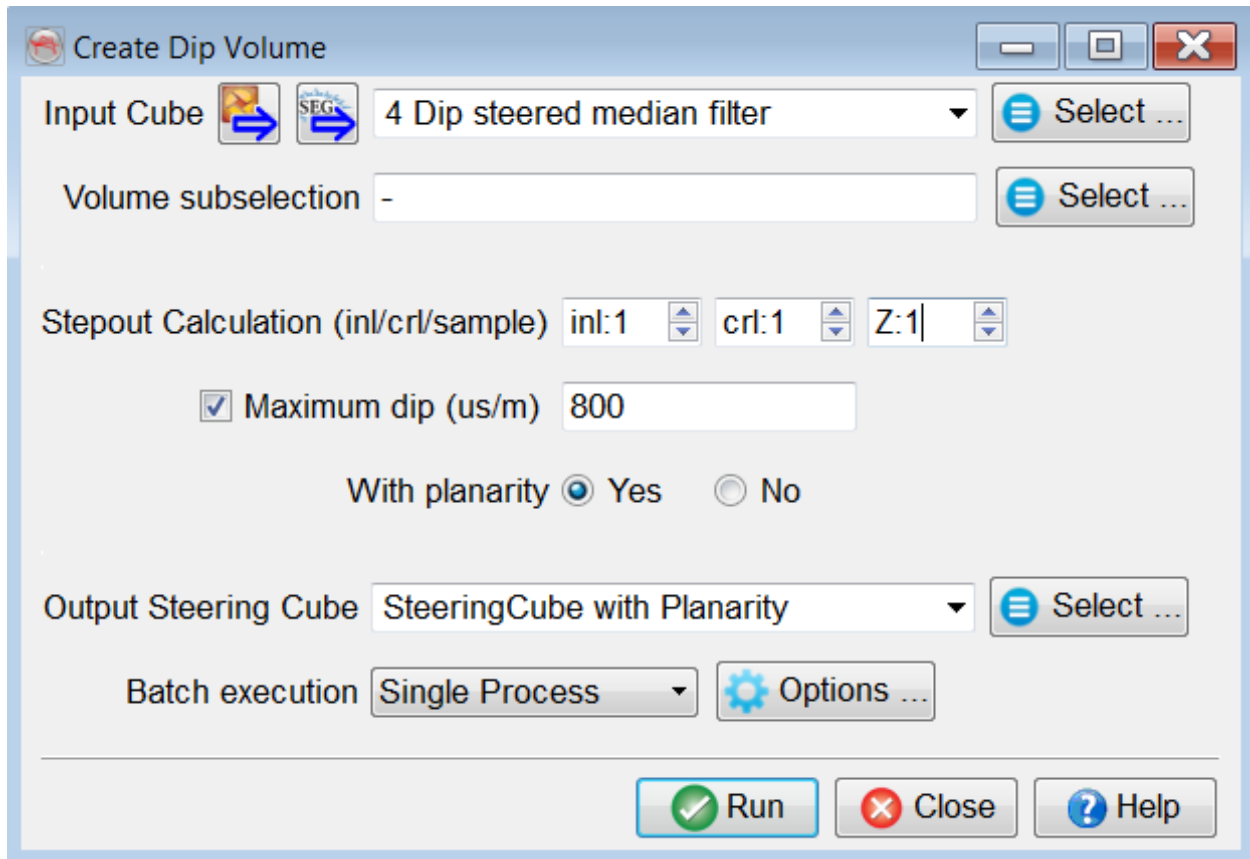
Edit Seeds window will show the origin (user defined or wells) of the picks. By default the seeds are displayed on sections (e.g. inlines/crosslines only). To view in 3D, you may want to toggle this feature off.

Processing Parameters

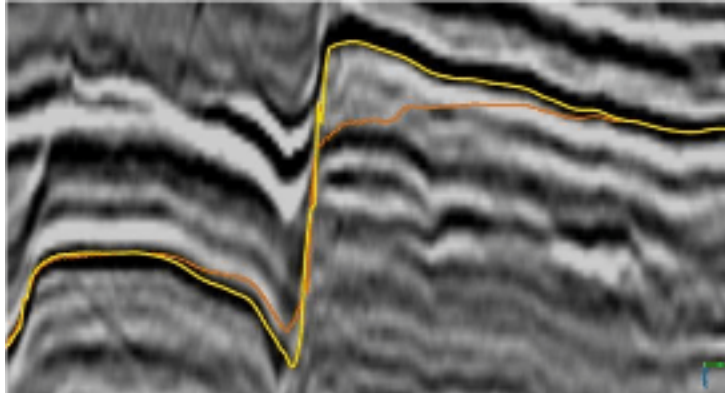


- Manually interpreted faults are provided by setting *apply fault dips* to *yes*, which will perform two tasks:
 - Calculate fault dips and update the SteeringCube.
 - Update the planarity cube at fault location by setting a minimum value of 0.01.
- It is not recommended to use the faults without a confidence cube.
- A confidence weight volume is optional. It is only required if the interval is faulted. If the given volume has higher weights (e.g. near to 1) at the fault weights, you may flip the weights ($1 - \text{input}$) to minimize the effect of erroneous dips at fault locations.

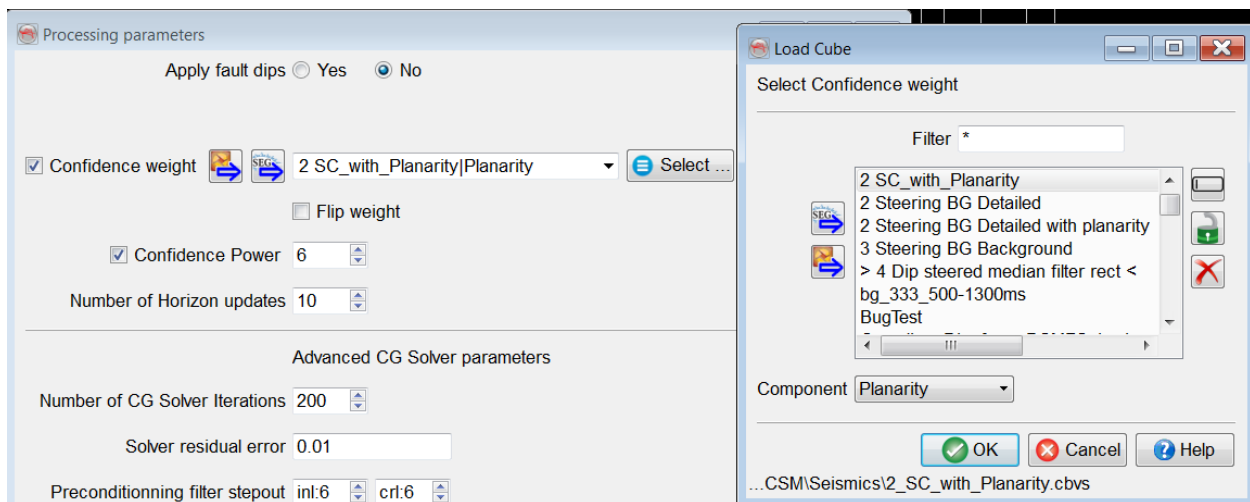
- A planarity volume – as a confidence weight – can be created by pressing the *Create* button. This will create a SteeringCube with a planarity volume as a third component.



The following result is produced with a confidence weight volume. With a power of 8 (yellow horizon) the event is correctly located when compared with the power of 2 (orange horizon).



- When the batch processing finishes, the right component is chosen to select as a confidence cube (see below):



- A power function is applied to the confidence weights (planarity). Its value is trivial. However, we recommend using a power of 6 or higher. A higher value increases the contrast between faults and continuous reflectors (see below).
- The *Number of Horizon updates* field (max. 100) is the number of times the horizon is changed and horizon dips are re-calculated. The quality of the results increase logarithmically with increasing number of horizon updates. After a certain number of updates, typically 10, the horizon will effectively not change further. If such a situation occurs, we consider that the results are fully converged

and the best possible solution is achieved.

- The core of the algorithm is a preconditioned Conjugate Gradient ‘CG’ solver, which attempts to solve a system of linear equations. This algorithm is controlled with the following parameters:
 - **Iterations:** The maximum number of iterations the GC solver is run to solve the inverse problem.
 - **Residual error:** The threshold value below which the GC solver stops.
 - **Preconditioning filter stepout:** Is a recursive filter to smooth the horizon and minimizes abrupt jumps (max. 10).
 - **Inversion Operator Weight:** This parameter is trivial and helps in overriding the dips in bad data quality areas. If it is larger than 0 (max. 1), the weighted dip derivatives are subtracted from the dips obtained from horizons to minimize the erroneous dip effects. By default, we do not recommend adding such weights. The Confidence Weight volume mostly controls such effects.

Larger Surveys

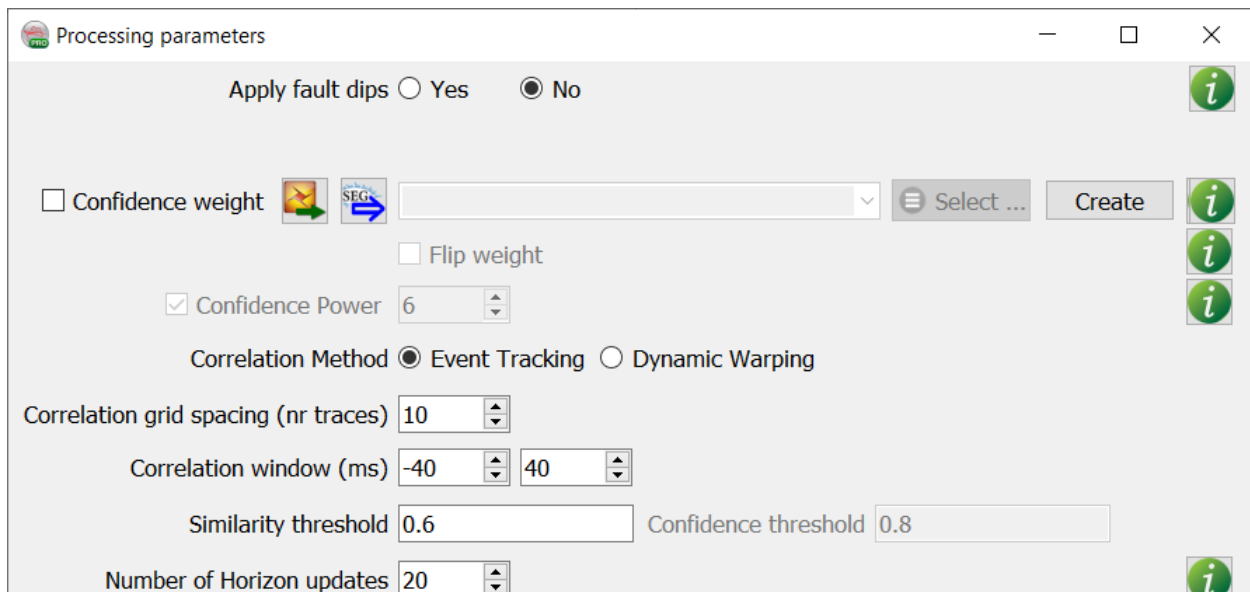
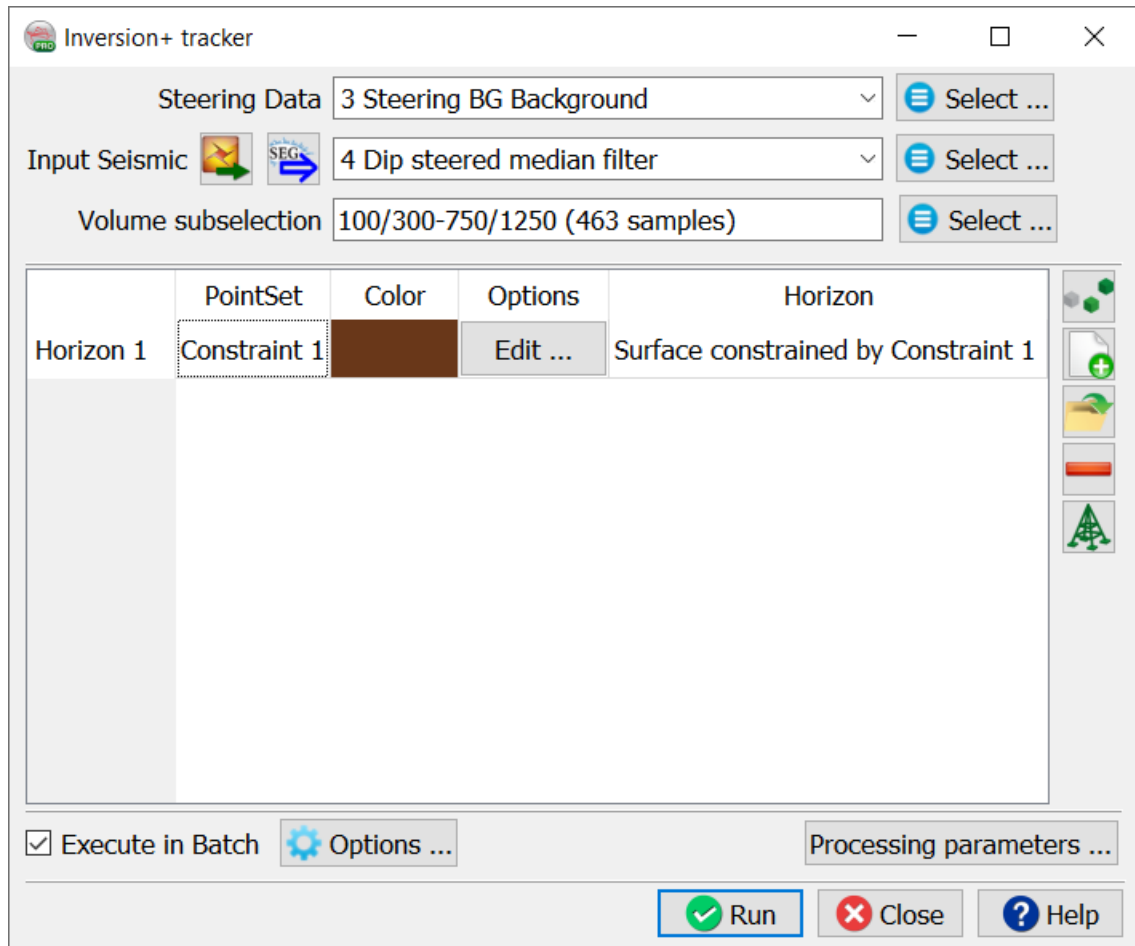
For larger datasets (e.g. volume size > 5GB) with limited RAM, we recommend decimation of the results. This can be done in the volume sub-selection. One may upscale the lateral stepouts (Inline/Crossline) to produce a horizon.

Reference

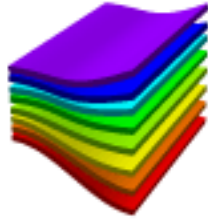
Wu, X., and Hale, D. (2015) Horizon Volume with Interpreted Constraints, *Geophysics*, v80, Issue 2, IM21-IM33.

4.8 Track Using Inversion+

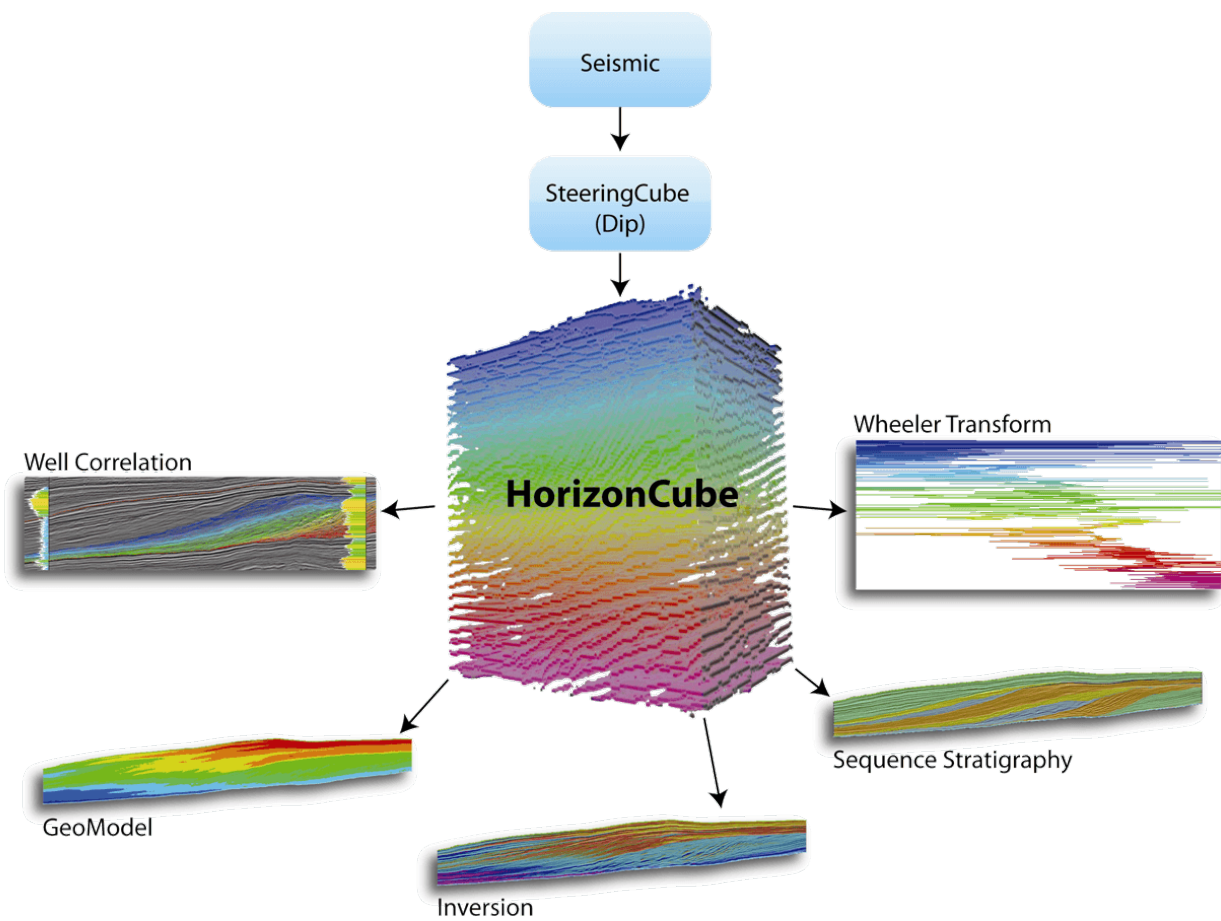
This seismic horizon tracker combines amplitude tracking and tracking from seismic dips.



5 HorizonCube



5.1 Introduction



HorizonCube: A plugin that auto-tracks a dense set of mapped 3D stratigraphic surfaces. The HorizonCube impacts all aspects of seismic interpretation work and allows the interpreter to extract more geology from the data. The HorizonCube is used for: detailed geologic model building, improving seismic inversion, sequence stratigraphic interpretation (SSIS) and correlating wells (Well Correlation Panel).

Purpose: The HorizonCube will impact the entire seismic interpretation workflow, leading to significant improvements, including:

- More accurate low frequency model building and robust geological models
- Superior quantitative rock property predictions
- Easy detection of stratigraphic traps (sequence stratigraphy)

Background: The HorizonCube plugin was first launched as the Chronostratigraphy that was part of the OpendTect SSIS (Sequence Stratigraphic Interpretation System) plugin. It consists of a dense set of correlated 3D stratigraphic surfaces that are assigned a relative geological age, with a corresponding colour. It was not long before we realized the potential of the HorizonCube and that the number of applications derived from HorizonCube exceeded just the sequence stratigraphy domain and had applications across the interpretation workflow. The stand-alone HorizonCube plugin was born with the 'HorizonCube' separated from SSIS.

Today, users can look forward to the following benefits:

Low Frequency Model Building and More Accurate, Robust Geological Models

In standard inversion workflows, the low-frequency model is considered the weakest link. Now, users can create highly accurate low frequency models by utilizing all the extracted seismic events from the HorizonCube, allowing a detailed initial model to be built.

In a similar fashion rock properties can be modeled. Instead of using only a few horizons, all horizons of the HorizonCube are used, resulting in greatly improved rock property models.

The HorizonCube and other Plugins .

Rock Property Predictions

The highly accurate low frequency models can be used to create geologically correct Acoustic Impedance (AI) and Elastic Impedance (EI) cubes using the Deterministic and Stochastic Inversion plugins. To complete the workflow, the Neural Networks plugin is used to predict rock properties from the Acoustic Impedance volume, avoiding the use of oversimplified linear models which can not accurately describe most rock property relations.

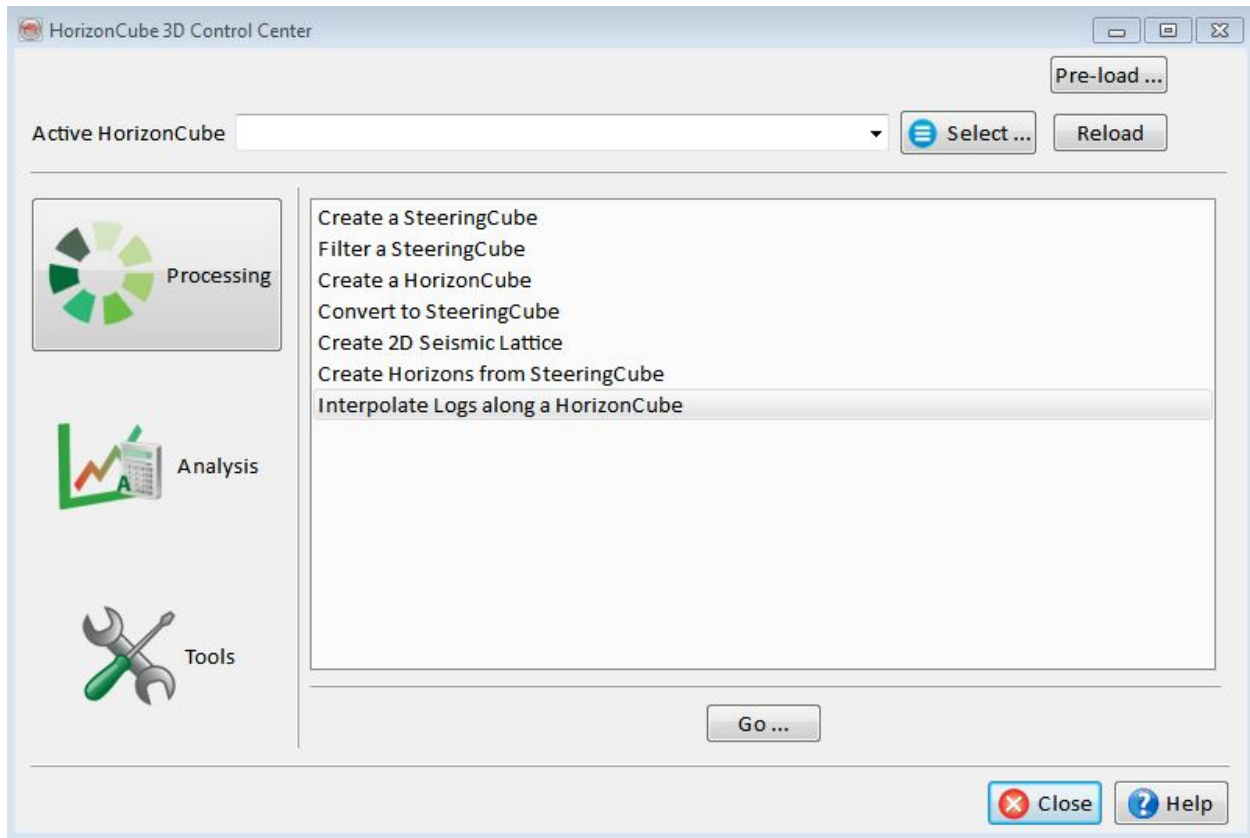
These advanced tools bring a high degree of precision to traditional seismic workflows, resulting in better seismic predictions and more accurate input into the reservoir management decision-making process.

Sequence Stratigraphy (SSIS plugin)

The SSIS plugin works on top of the HorizonCube plugin. Users can interactively reconstruct the depositional history in geological time using the HorizonCube slider, flatten seismic data in the Wheeler domain, and make full system tracts interpretations with automatic stratigraphic surfaces identification and base-level reconstruction.

In the near future the SSIS plugin will be integrated with the Well Correlation Panel, enabling the HorizonCube slider and systems tracts interpretation to be integrated with the well correlation display and its interactive functionalities.

5.2 HorizonCube Control Center



It is a selection menu that can be placed anywhere on the desktop while working with OpenTect. It is designed as a control box that contains several sub-menus inside. The control box is used to run the following:

1. [Data preparation](#)
2. [Horizons from SteeringCube](#)
3. [Create a New HorizonCube](#)
4. [HorizonCube Attributes](#)
5. [HorizonCube Tools](#)
6. [3D Slider](#)

5.2.1 Processing

See the following chapters below.

5.2.1.1 Create A SteeringCube

SteeringCube is a necessary input for a HorizonCube. For the full article refer to [4.2.2 Create SteeringCube](#) chapter.

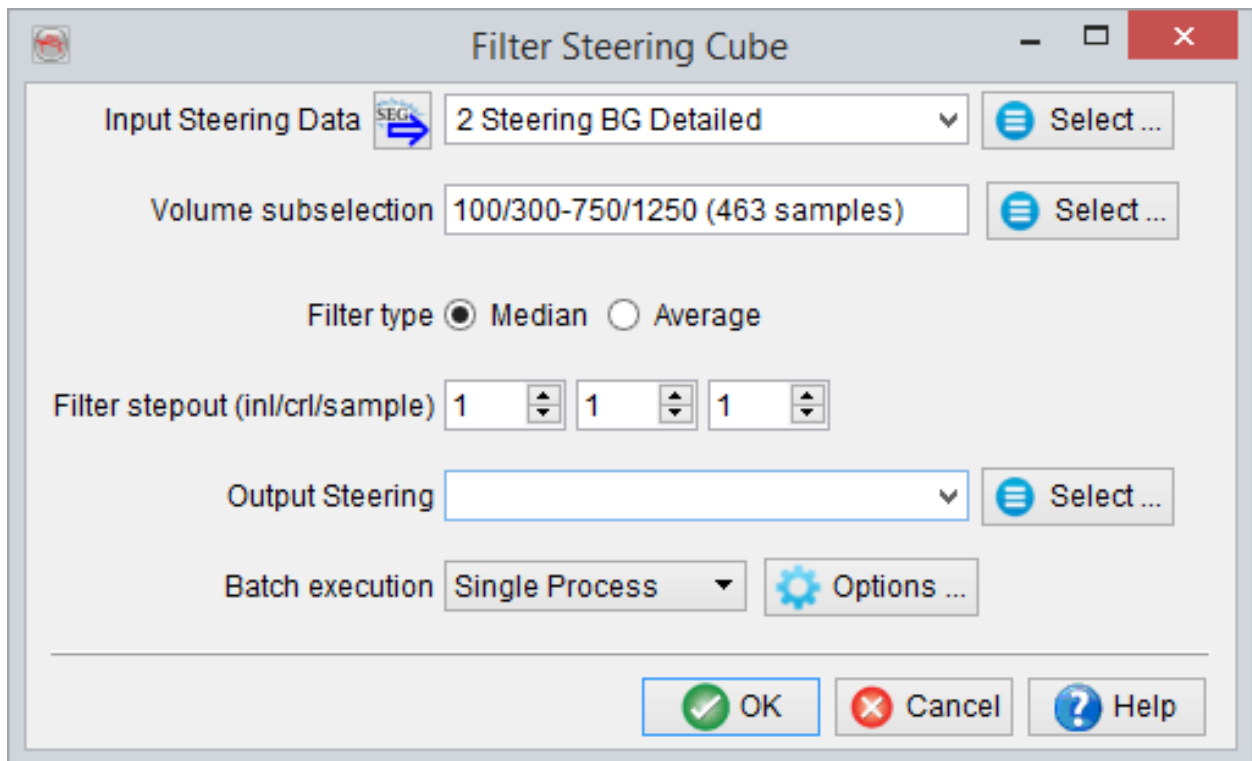
Create Steering Cube

4 Dip steered median filter	Select ...	Steering algorithm	BG Fast Steering
100/300-750/1250 (463 samples)	Select ...	Calculation Stepout (in/cr/sample)	in:1
	Select ...	Specify Maximum dip	<input type="radio"/> Yes <input checked="" type="radio"/>
Single Process	Options ...	Filter result	Median
		Filter stepout (in/cr/sample)	1

OK

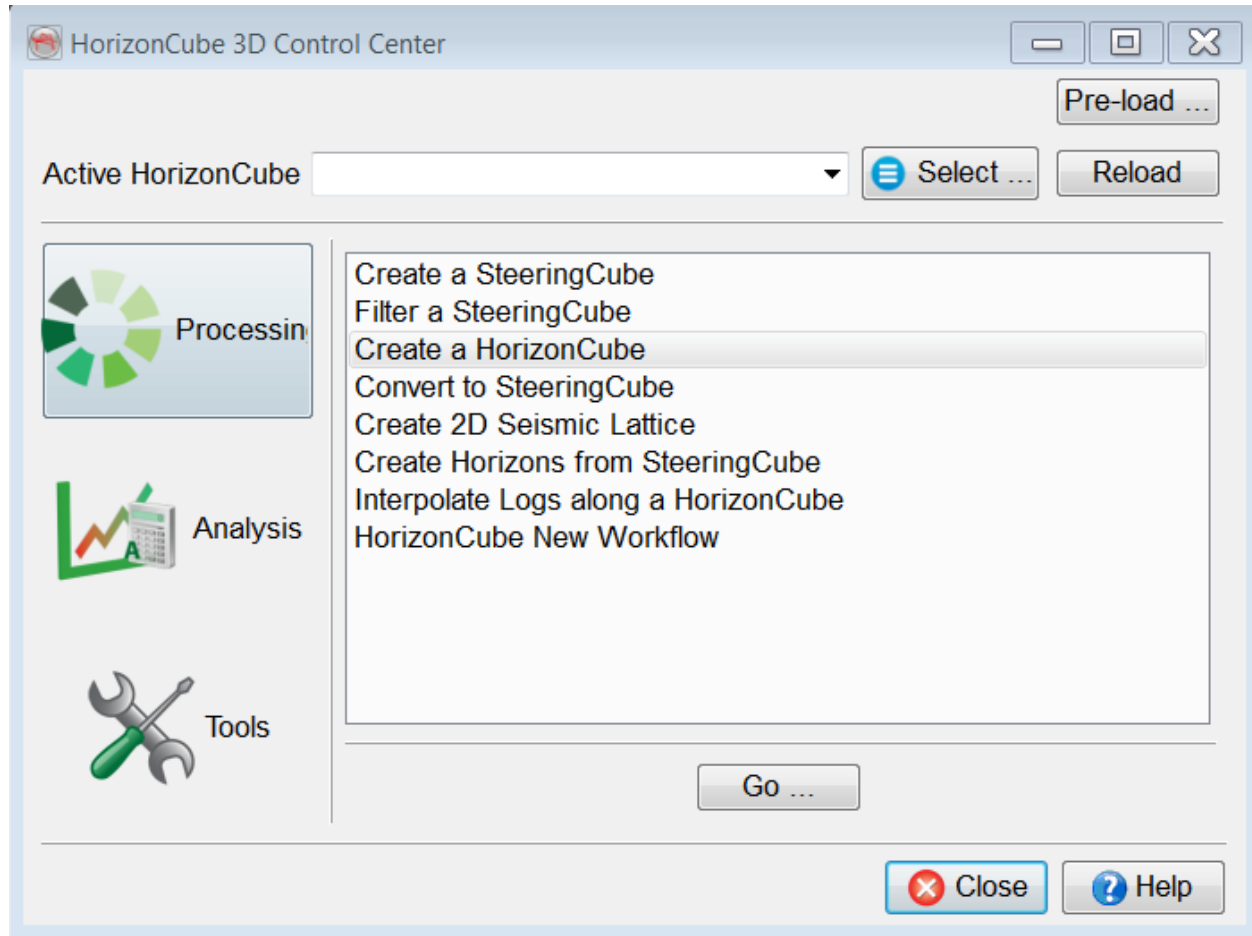
5.2.1.2 Filter A SteeringCube

The SteeringCube can be improved through smoothing. A good SteeringCube improves the HorizonCube calculations. Use the option *Filter SteeringCube...* to apply a median filter to the SteeringCube. Adjust the needed parameters: inline/crossline step-out, time gate size, and maximum dip according to your dataset. Filtering of the SteeringCube is described in [4.3 Filter](#).



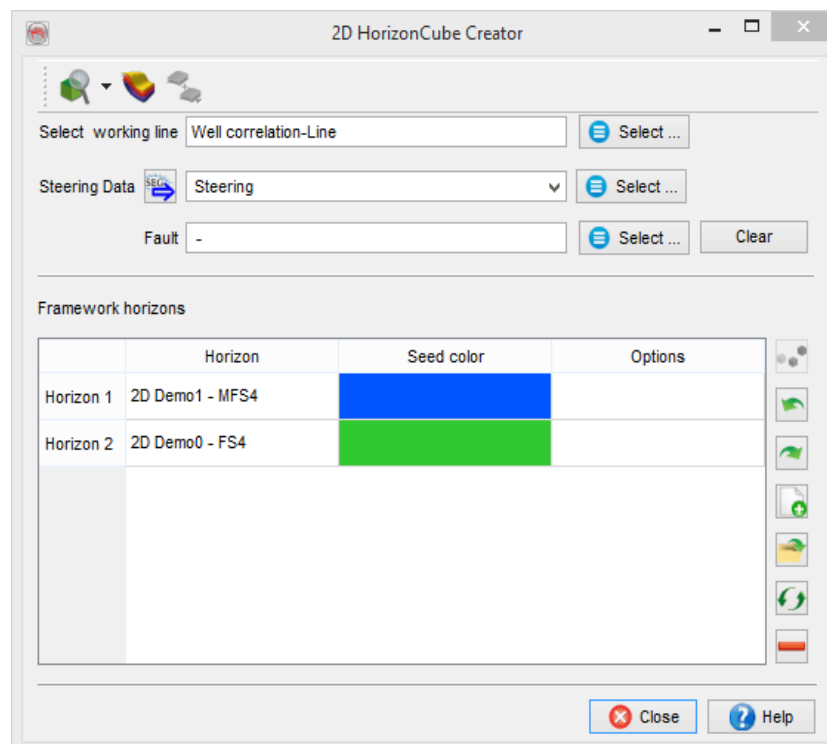
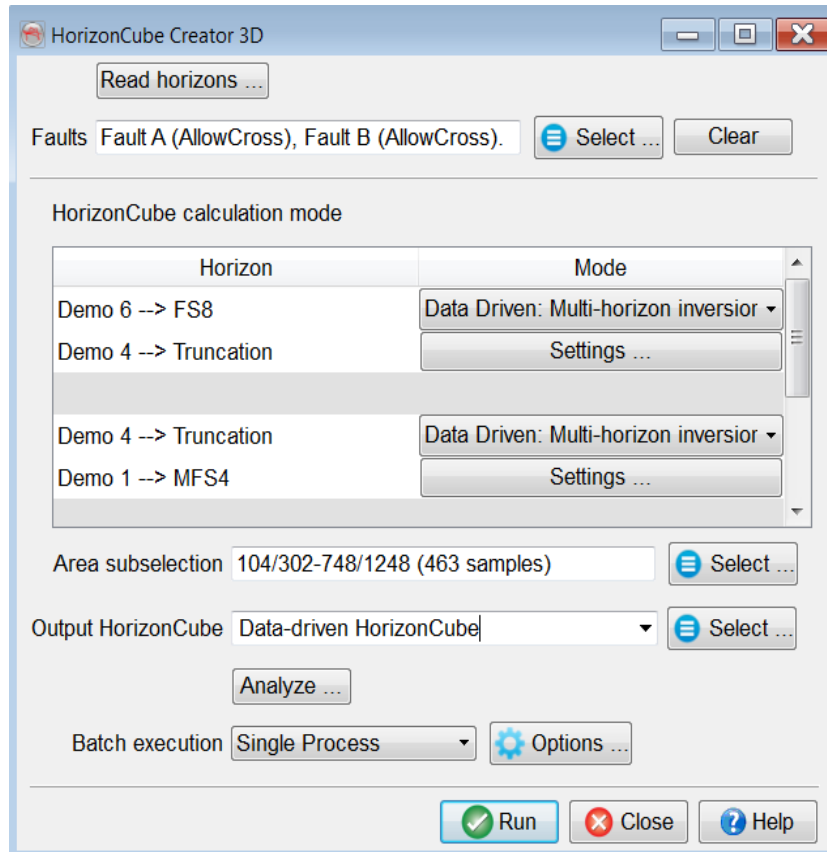
5.2.1.3 Create HorizonCube

A new 2D/3D HorizonCube is created from the [Control Center](#).



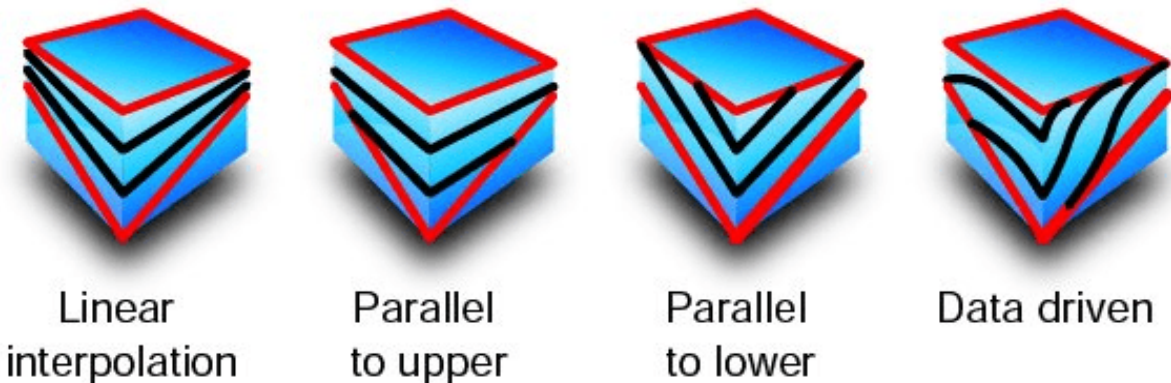
The HorizonCube is created between the given horizons. These horizons are selected (minimum of 2, top and bottom) by clicking the *Read horizons* button for 3D and *Load Existing Horizons* icon for 2D. When the horizons are read, they are automatically placed in stratigraphic order and the corresponding packages are defined. Each package is defined by two horizons (a top and base of the package). In case of 2D HorizonCube, the input lineset/linename are selected to get the input geometrical

information. This 2D lines selection is made by pressing the *Select* button next to the LineSet/LineName field in the HorizonCube Creator 2D window.



HorizonCube Models

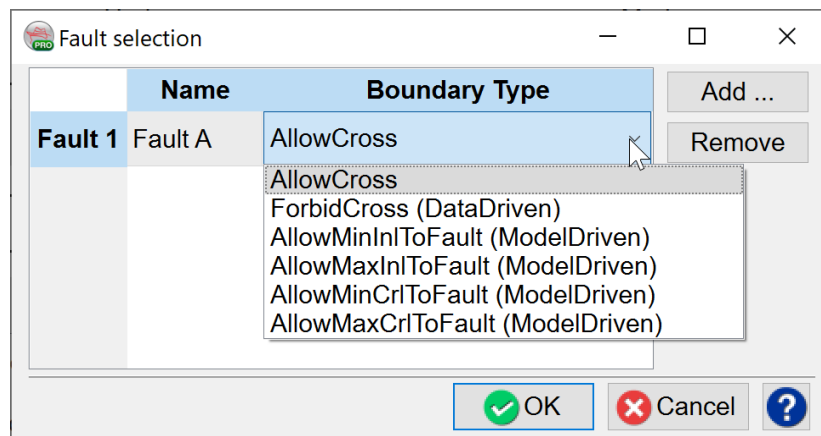
Three models are based on interpolation and one model uses a data driven approach:



- [Proportional](#): The proportional model can also be used for 3D Stratal Slicing. In settings, the user can specify the spacing between two HorizonCube horizons.
- [Parallel to upper](#): This model best depicts the layout patterns, such as onlaps. In settings, the user can specify vertical spacing between HorizonCube horizons.
- [Parallel to lower](#): This model relates to upward truncation patterns. In settings, the user can specify vertical spacing between HorizonCube horizons.
- [Data driven](#): This model is driven from data based upon steering information. The HorizonCube will follow the dip and azimuth information read from a [SteeringCube](#).
 - Tracking based algorithm – It tracks seismic horizons from a single start position by following a SteeringCube along a line or in 3D.
 - A global inversion scheme to create a HorizonCube. This is quite similar to the concept of the [Unconformity Tracker](#).

HorizonCube Creator 3D Options

Fault: 3D faults or 2D faultsticksets are selected from the main HorizonCube creator dialog. The user can select more than one fault / faultstickset.



AllowCross means the faults are completely ignored and the HorizonCube is created as if there were no faults.

ForbidCross is meant for the data driven HorizonCube trackers that can detect a fault and avoid crossing it. So, the HorizonCube events experience a throw proportional to the top and bottom horizons when they cross the fault.

The other options are for Model Driven methods. Since these methods do not do any tracking, it is difficult to apply a throw. Therefore, the user can generate a partial HorizonCube on one side of the fault. The different options allow choice of which side of the fault plane the HorizonCube will be generated. For example, if a Fault plane is running parallel to inlines, you have two fault blocks, one with lower crossline numbers and the other with higher crossline numbers. If you want to create the HC in the lower crossline number block, you would choose the option **AllowMinCrIToFault**. For the higher crossline number block, the option would be **AllowMaxCrIToFault**. The same rules apply for a fault plane running more or less parallel to a crossline. Then the two options to choose from would be **AllowMinInlToFault** and **AllowMaxInlToFault**.

Area Subselection: An area sub selection can be made to restrict the HorizonCube calculation. Please note that the HorizonCube can also be calculated within a polygonal area of interest. To know how to create a polygon in OpendTect, please refer to OpendTect Help documentation. However, this feature is not supported for a 2D HorizonCube.

Output HorizonCube: This field is used to give an output name for the HorizonCube.

Analyze: There are several checks being designed to quality control the failure of a batch program running to create a HorizonCube. The cause of failure could be a bad start position e.g. edge of survey, a trace defining a fault location, issues with the framework horizons etc. Therefore, it is always suggested to Analyze the HorizonCube processing before starting the actual processing (Go button).

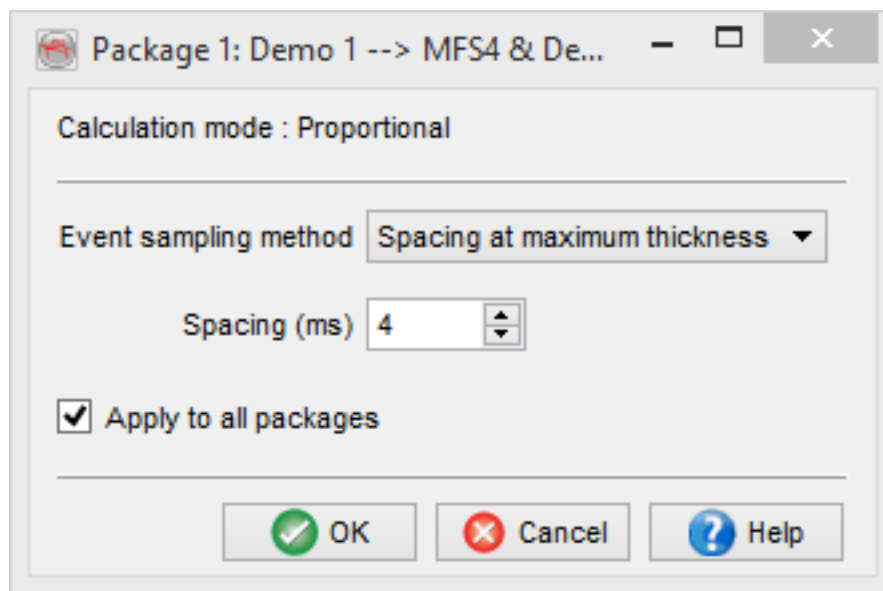
Go: It starts the HorizonCube processing in a pop-up window. The batch processing window will provide you the instantaneous progress of HorizonCube calculation. Once the batch program prompts "Finished batch processing", the output is ready to be visualized. Press "Show options" to get the possibility for remote processing (after having pressed "Go").

For more information about 2D HorizonCube Creator window refer to [5.2.1.4 2D HorizonCube Processing](#).

5.2.1.3.1 Model Driven Settings

In the Specify Parameters window a user can choose between two sampling methods for the HorizonCube events:

- Spacing at maximum thickness, for which the spacing parameter (in ms / m /ft) needs to be specified. This sets the sampling rate of the data in the Wheeler domain.
- Number of events that defines the number of Chronostratic events that can be exported as horizons.



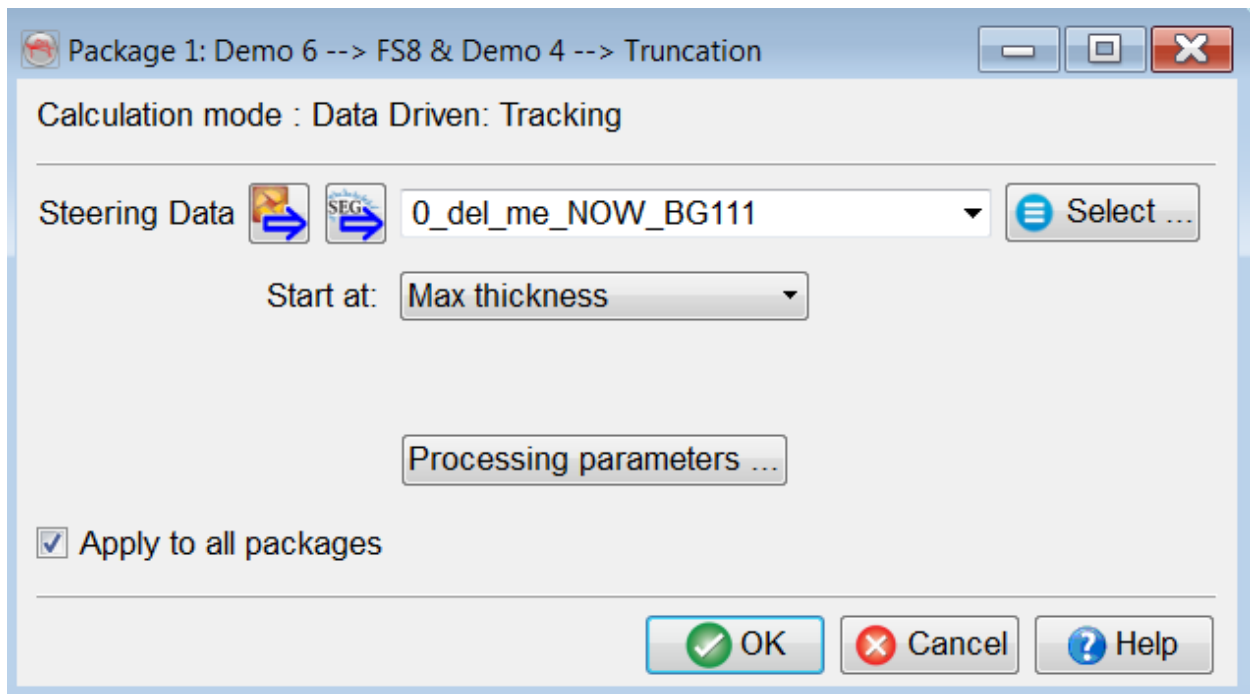
The option "Apply to all sequences" takes effect only when pressing "OK" and cannot be undone.

5.2.1.3.2 Data Driven Settings

In the Data Driven calculation mode the settings can be edited separately for every package.

Parameters

- *Steering Data*, which is a necessary input HorizonCube creation. The quality of the HorizonCube depends on the quality of SteeringCube itself.
- Start position, another key parameter to control the quality of HorizonCube, defined in the *Start at* option in the settings dialog. The start position also defines indirectly the number of events to be initiated in the first iteration / pass, since the package thickness varies laterally.



- *Max thickness*: Ideally the start position should be at a maximum thickness defined by the given framework horizons. Sometimes the trace defining the maximum thickness is either at the edge of a survey or in a poor seismic quality area.

In such cases, the start position should not be set to maximum thickness. The best practice is to create isochron (or isopach) maps for the given horizons and find out the thicker areas with good quality of seismic data.

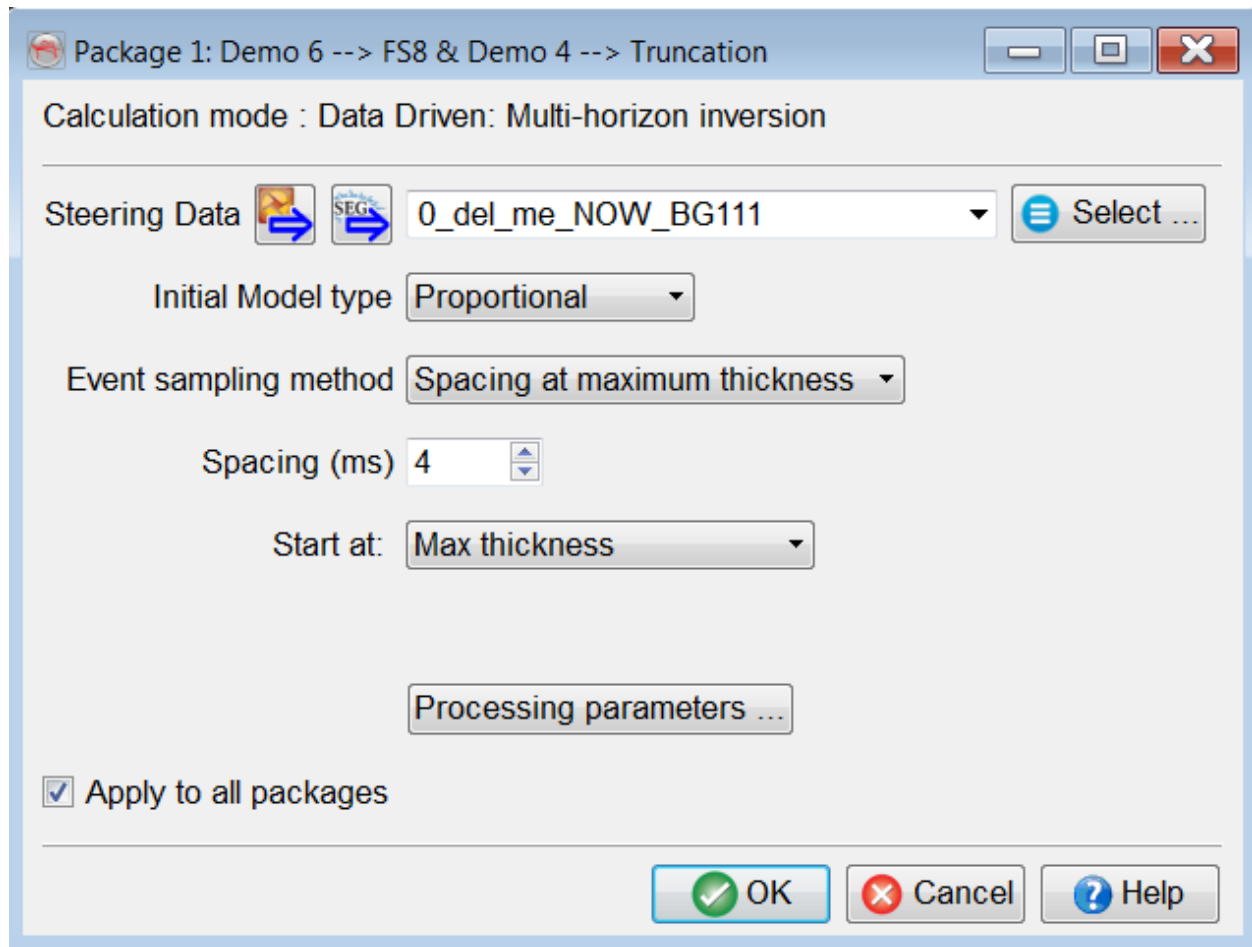
- *Center*: Center refers to a trace that lies in the middle of a survey.
- *Inline center / maximum thickness*: It defines a trace position that lies at a location of maximum thickness but it uses the central trace of that particular inline.
- Processing parameters for Continuous Events and Truncated Events are available and explained in the followings sub-sections.

Parameters (Multi-horizon Inversion)

Multi-horizon inversion is a data-driven technique similar to the [Unconformity Tracker](#). The method first creates a set of model-driven horizons between the framework (input) horizons, which are provided by a user. A constrained position (seed) per model-driven horizon is defined by the event sampling method. These model-driven horizons are then iteratively updated through the unconformity tracker. Once all horizons are finished, the algorithm automatically checks for crossings between horizons and fixes them prior to storing the horizons as a HorizonCube.

Recommendations

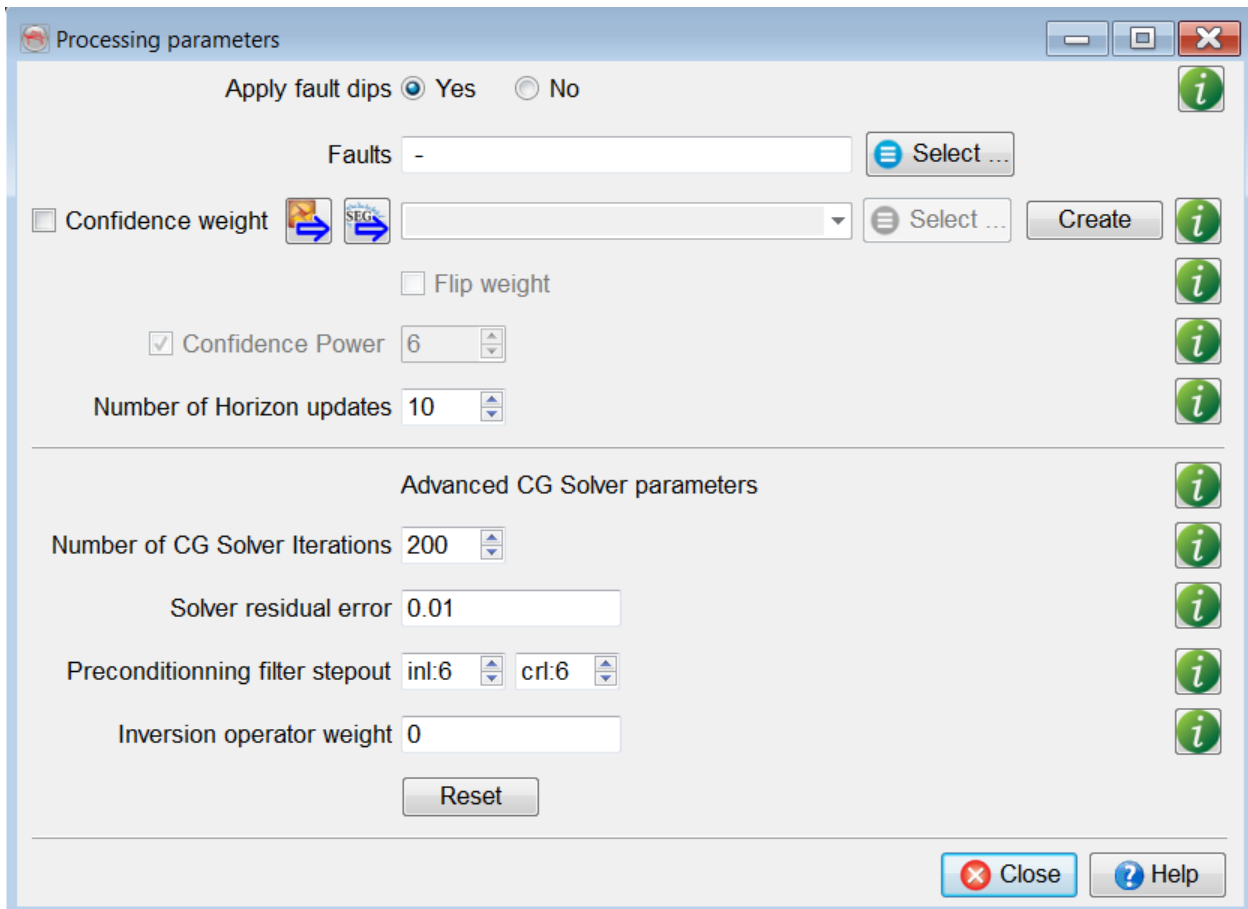
Since this algorithm needs to invert lots of horizons at once, we recommend running a HorizonCube within a package. One may run a HorizonCube per package on different machines and do vertical merging later on (via Tools which are available in the HorizonCube Control Center).



- Input *Steering Data* is a requirement for data-driven HorizonCube.
- *Initial Model Type*: Choose a right model that best describes the interval between the framework horizons. For instance, if the zone contains an angular unconformity where intra-horizons are following the base, then the optimal model would be parallel to the lower else it could either be proportional or vice versa.
- *Spacing*: This controls the number of events to be created in a package. Finer the spacing the more horizons will be produced at a cost of processing speed.
- *Start at*: This parameter defines the location to initiate the model based horizons. Generally speaking, this could be a location defining the major thickness in a package resulting in more horizons.

- *Processing parameters* for the multi-horizon inversion are identical to the ones defined for [Unconformity Tracker](#).

The option "Apply to all packages" takes effect only when pressing "OK" and cannot be undone.



5.2.1.3.2.1 Advanced Options

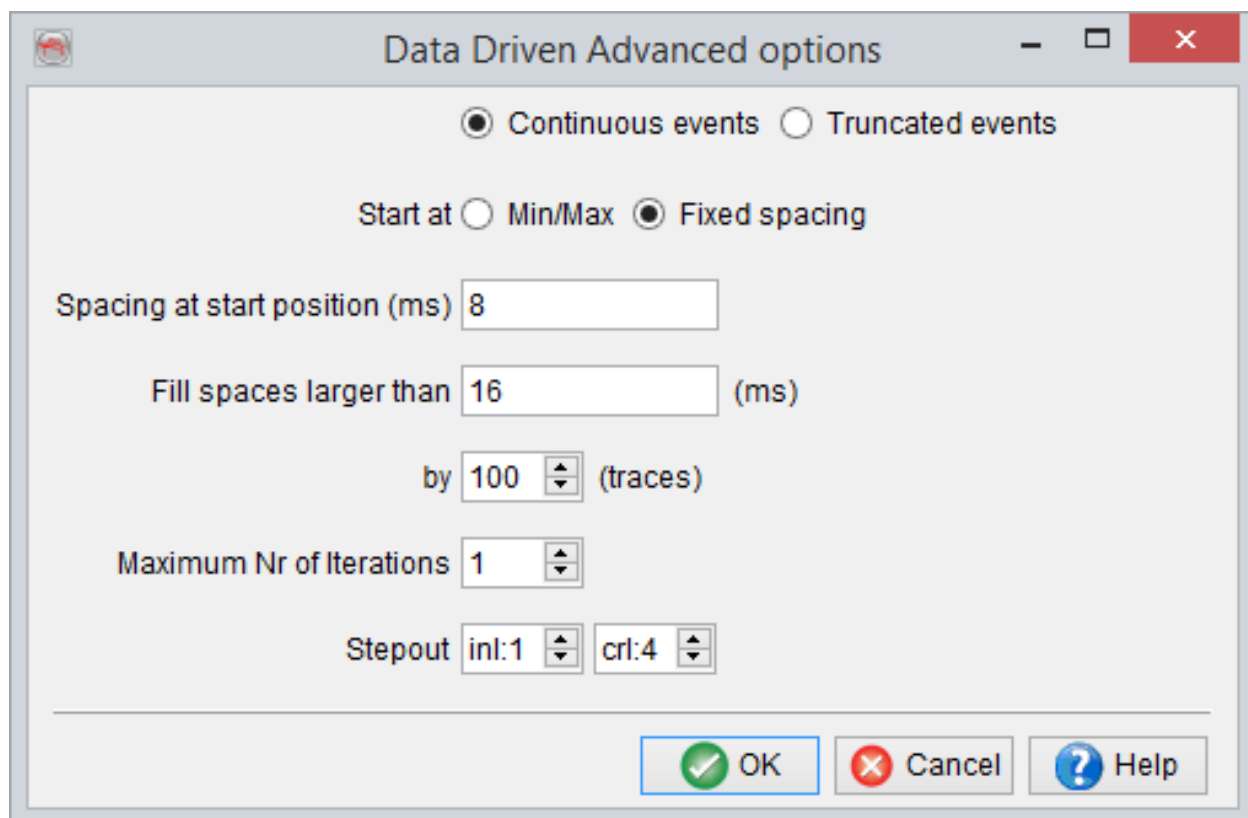
Continuous / Truncated Events: The continuous events are fully mapped events in 2D/3D that converge / diverge with each other but are not allowed to cross each other. The truncated events are diachronous in 2D/3D i.e. when two horizons come close to each other, the tracking is stopped and a new horizon is initiated afterwards. The continuous HorizonCube is good for GeoModel building or for low frequency model building for seismic inversion. On the other hand, truncated HorizonCube is useful for Sequence Stratigraphic (SSIS) interpretation e.g. wheeler transformation.

The following most important parameters apply to both HorizonCube types:

Spacing at start position: Used at the start position only. Vertical spacing between the seeds from which the HorizonCube events will be initiated. Implies a regular sampling of the events at the start position. The continuous HorizonCube proposes an alternative mode (see corresponding section below).

Smallest spacing (e.g. 4) will result into a dense HorizonCube and a largest spacing (e.g. 16) will result into a coarse HorizonCube.

Stepouts: The stepout (inline : crossline) parameters control the spatial quality of horizon tracking in 3D. It defines the number of z-values (of an event) to be used to forecast the z-value at a new trace position. By default the inline steps are set to 1 (i.e. 3 z-values on a crossline plane) and the crossline steps are set to 4 (i.e. 9 z-values on inline plane). The smaller stepout mean faster and detailed dip field tracking and the largest stepout are preferable for a regionally continuous event. The best practices are to test it with asymmetric parameters (e.g. 1:4 or 4:8 or 1:12). The symmetric steps (e.g. 4:4, 8:8 or even higher 12:12) are useful to average-out very small details / noisy trails while tracking. This parameter is a key to be tested through this utility prior to creating any HorizonCube. It is recommended to find optimal stepout values by varying it for an individual horizon.

The image shows a dialog box titled "Data Driven Advanced options". It has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The dialog contains several settings: "Continuous events" is selected with a radio button, while "Truncated events" is unselected. "Fixed spacing" is selected for the "Start at" option, while "Min/Max" is unselected. The "Spacing at start position (ms)" is set to 8. "Fill spaces larger than" is set to 16 (ms). The "by" value is set to 100 (traces). The "Maximum Nr of Iterations" is set to 1. The "Stepout" is set to "inl:1" and "crf:4". At the bottom right, there are three buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a blue question mark icon).

Data Driven Advanced options

Continuous events Truncated events

Start at Min/Max Fixed spacing

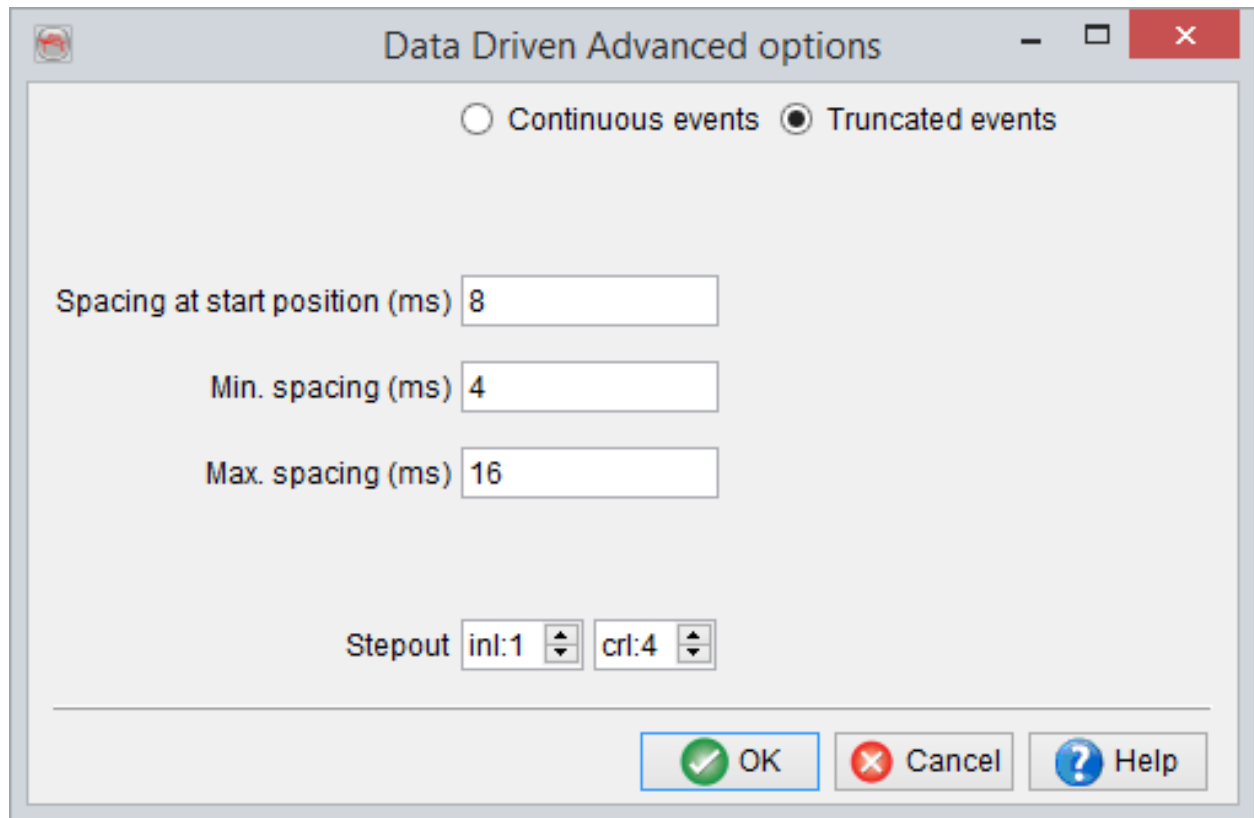
Spacing at start position (ms)

Fill spaces larger than (ms)

by (traces)

Maximum Nr of Iterations

Stepout

A dialog box titled "Data Driven Advanced options" with a standard Windows window border. It contains two radio buttons for event types, three text input fields for spacing values, and two spinner boxes for stepout values. At the bottom are three buttons: "OK" with a green checkmark, "Cancel" with a red X, and "Help" with a blue question mark.

Continuous events Truncated events

Spacing at start position (ms)

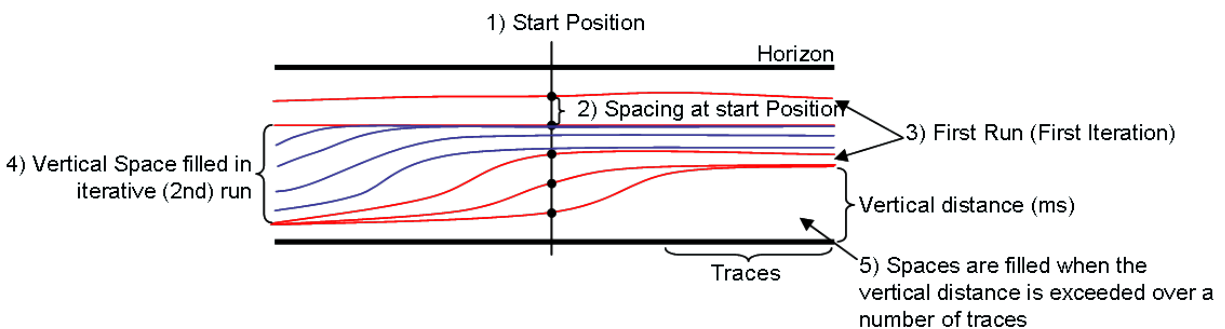
Min. spacing (ms)

Max. spacing (ms)

Stepout

5.2.1.3.2.2 Continuous Events

At the start position (1) numerous horizons are initiated at a user defined interval (2). Normally, the sample rate is used here in order to initiate a horizon at every sample. These horizons are tracked from the start position outward within the whole extend of the survey (3). When the horizons diverge large vertical spaces between the horizons are created, which are filled in iterative runs (4). To prevent very small vertical spaces to be filled (with horizons that are present in the whole survey) the spaces are defined by a vertical setting as well as a horizontal (5). A vertical space is filled when the vertical distance between the horizons exceeds a user specified amount over a lateral extend - the user defined number of traces-.



Start at: The (Start at) radio box (for Continuous HorizonCube) is used to define the trace position from which the horizons will be tracked. By default, *Fixed spacing* is used and the corresponding constant value is filled in the Spacing at start position field. The *Min/Max* relates to a given seismic cube (positive / negative amplitudes) i.e. the HorizonCube events will be initiated at a start position defined by the Min/Max amplitudes and this will not yield an evenly spaced HorizonCube at start position.

Fill spaces larger than (ms) or by (traces): This is used to specify the minimum allowed gap vertical (in ms / m / ft) or by distance (m / ft) to be filled in the subsequent defined iterations.

Max. Nr. of iterations Depending on the geologic thickness variations within a defined package, often the gaps are found after the first iteration of a HorizonCube. To fill the gaps in a HorizonCube the initial iteration value should be defined (either 1 or 2). The best practice is to create a HorizonCube with 1 iteration initially and then at later stages the gaps could be filled using the HorizonCube tools (Add more iterations). This is suggested as a quality control step because the HorizonCube calculation is slower for subsequent iterations. For instance, the HorizonCube with 1 iterations and smaller step outs could be generated in 1 or two hours. However, if the iterations are 2, the HorizonCube calculation time exponentially increases.

Data Driven Advanced options

Continuous events Truncated events

Start at Min/Max Fixed spacing

Spacing at start position (ms)

Fill spaces larger than (ms)

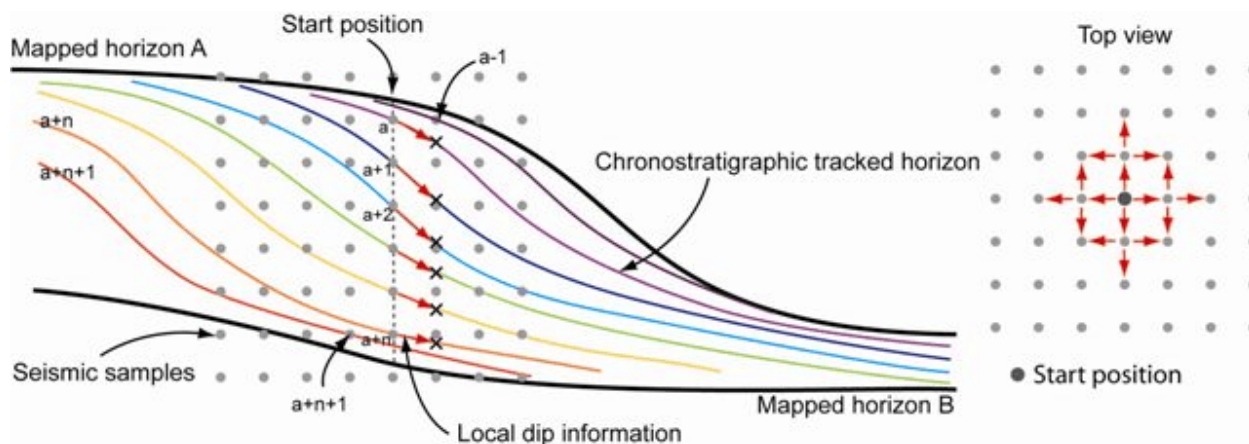
by (traces)

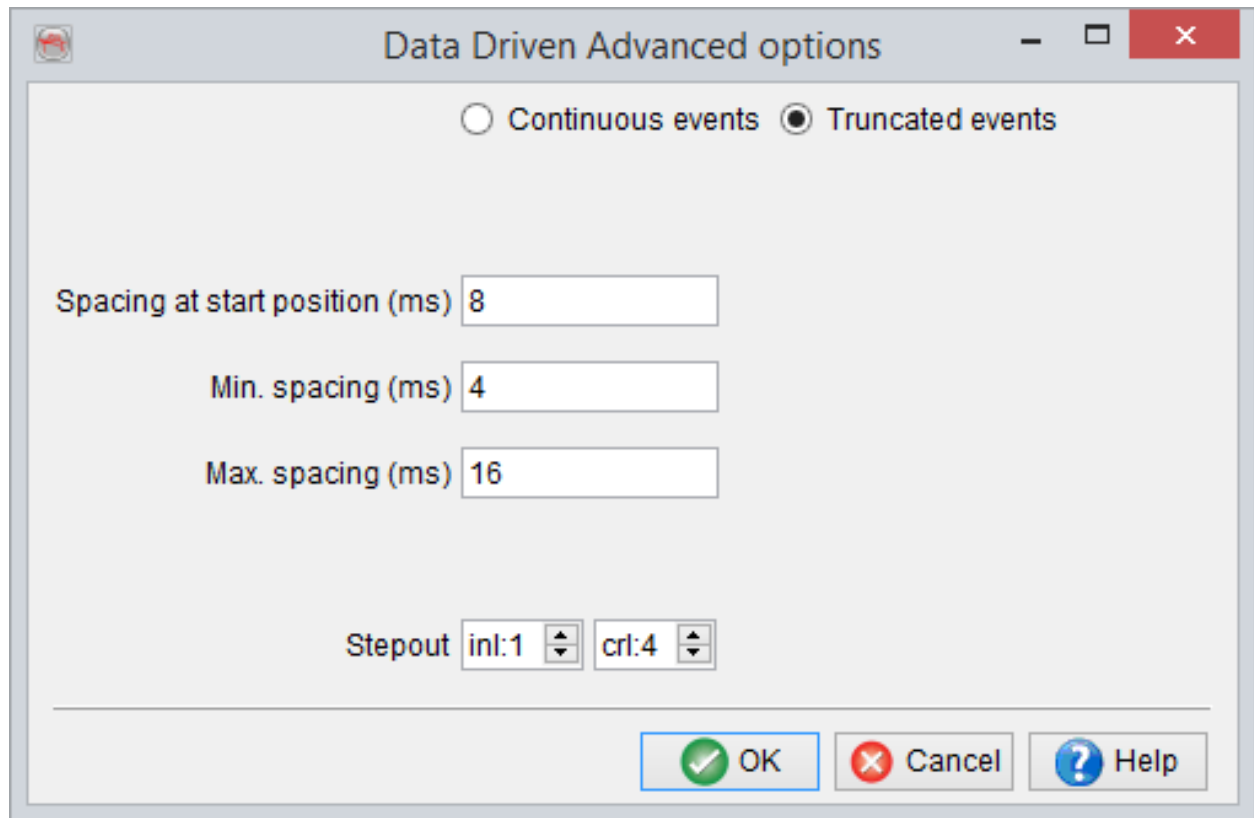
Maximum Nr of Iterations

Stepout

5.2.1.3.2.3 Truncated Events

Min or *Max. spacing* (ms / m / ft): This parameter is defined for a HorizonCube with truncated events. Converging events cause one of the two events to be stopped if the vertical distance becomes smaller than the minimum thickness. Diverging events cause one additional event to be added between the two diverging events when their vertical distance becomes larger than the maximum thickness.



A dialog box titled "Data Driven Advanced options" with a standard Windows window border. It contains two radio buttons for event types, three text input fields for spacing values, and two spinner boxes for stepout values. At the bottom are three buttons: "OK" with a green checkmark, "Cancel" with a red X, and "Help" with a blue question mark.

Data Driven Advanced options

Continuous events Truncated events

Spacing at start position (ms)

Min. spacing (ms)

Max. spacing (ms)

Stepout

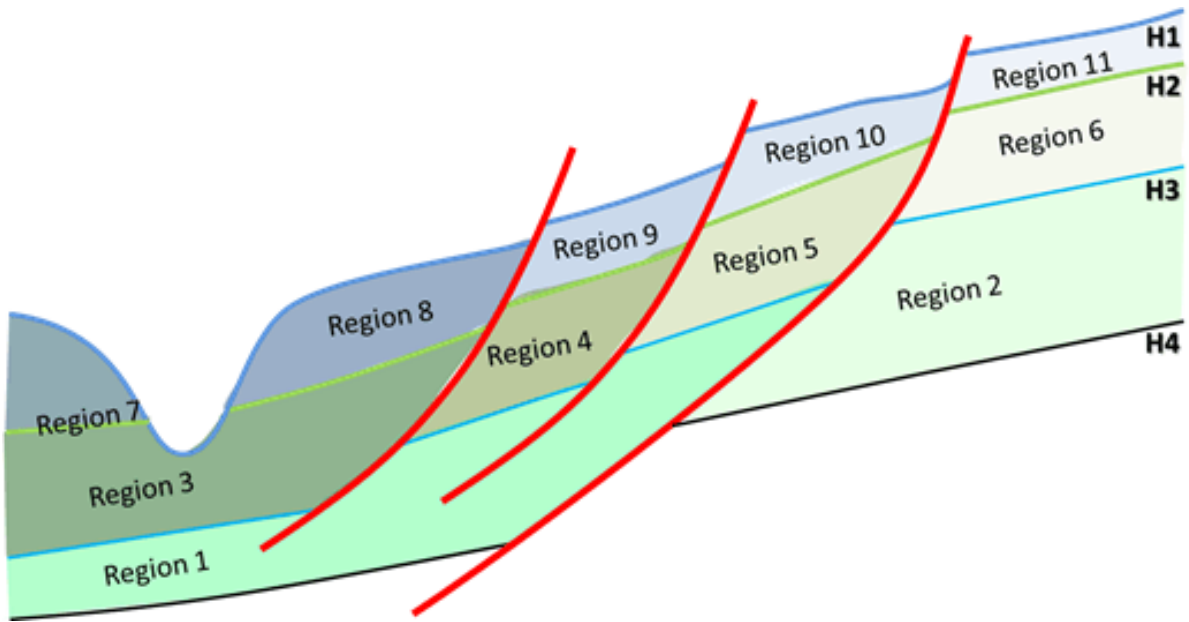
5.2.1.4 2D HorizonCube Processing

Introduction

Since version 5.0 OpendTect supports a new way of processing a 2D HorizonCube. The workflow is interactive and operates in regions. A region is defined as a vertical segment bounded by horizons and faults. HorizonCube processing is performed on a line-by-line basis and per region. Next the user interactively correlates HorizonCube segments from region to region over the entire line. This results in a merged HorizonCube in which all segments are correlated and the number of horizons per segment is equal. To process grids of 2D lines the user correlates and merges separately processed HorizonCubes into a final HorizonCube for the grid.

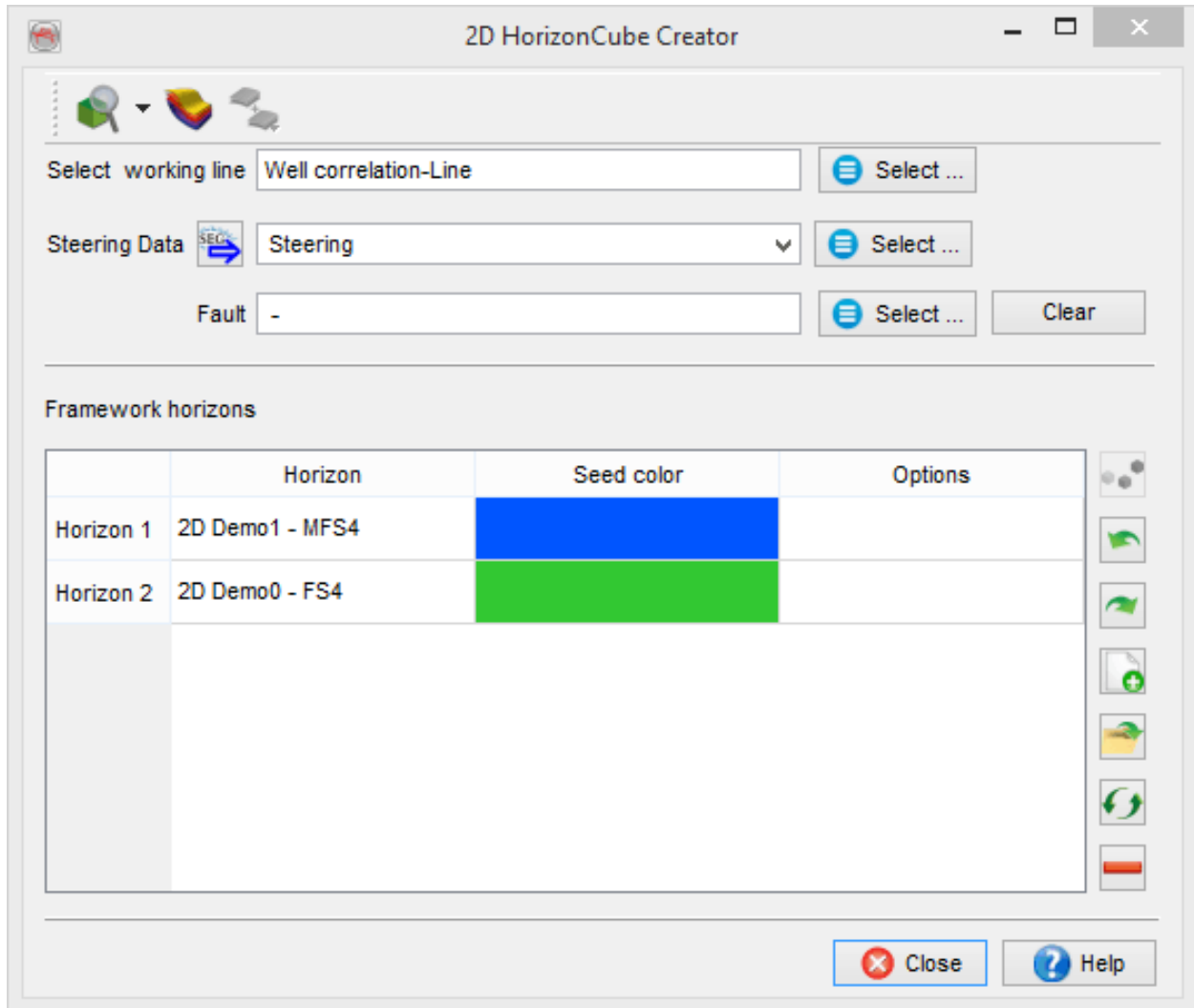
HorizonCube processing for one line is performed in three steps:

1. [Constructing the Framework.](#)
 - a. Select an existing set of horizons and faults.
 - b. Optionally, add more framework horizons using the new multi-seed, dip steered tracking algorithm.
2. [Processing of HorizonCube segments per region.](#)
3. [Correlation of HorizonCubes between regions.](#)



HorizonCube regions are automatically constructed as closed polygons from the given set of horizons and faults. The illustration above shows a set of geological regions that are automatically constructed by the algorithm.

5.2.1.4.1 Preparing the Framework



The user interface for the HorizonCube creator supports the following:

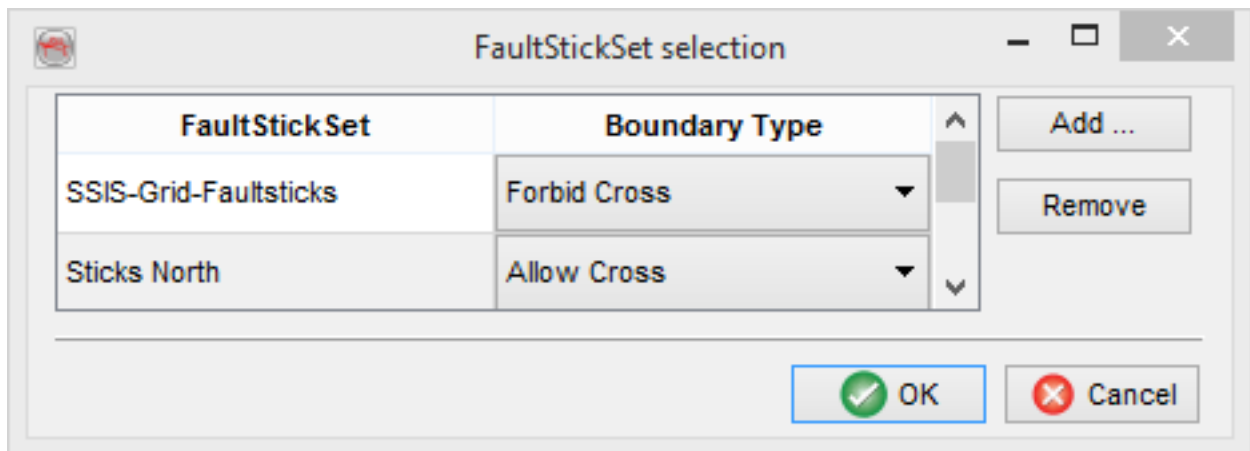
1. *Select working line*: Selection of a single 2D line on which a HorizonCube is processed.
2. *Steering Data*: The steering data corresponding to selected 2D line should be provided.

3. *Faults*: This option is used to select multiple fault stick sets. The fault sticks will be automatically displayed in the scene. The tracking algorithm can treat faults in different ways:

- a. Forbid Crossing – Tracking of Framework horizons and HorizonCube events stops when a fault stick is encountered. This option generates HorizonCube segments that will require correlation to make a final merged output.
- b. Allow Crossing – Tracking of Framework horizons and HorizonCube events will continue across fault sticks. This option generates a HorizonCube that does not require merging. It approximates results produced in older versions of OpendTect.



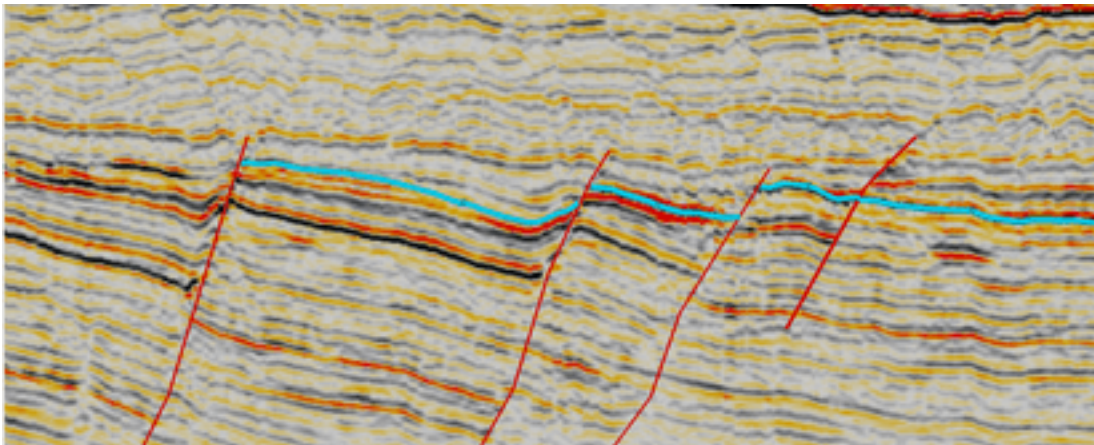
A user may draw channels and other geological terminations as (fake) fault sticks and select these as constraints to construct regions. This combined with manual correlation can improve the results.



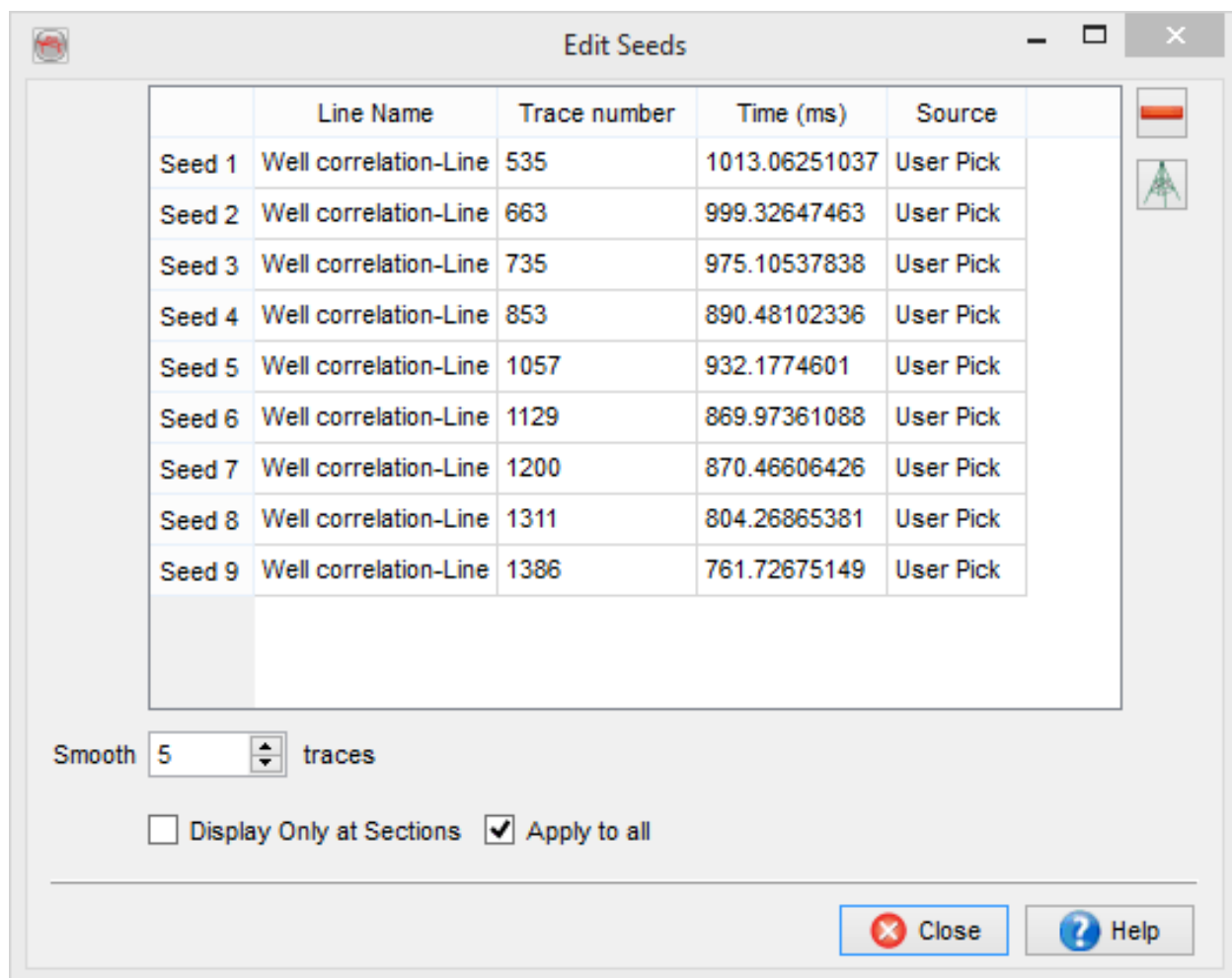
Framework horizons: Framework horizons are either imported, or they are created on-the-fly. To select existing seismic horizons press the open icon.

To create a new dip steered horizon proceed as follows:

- a. Press the + icon to add a new horizon.
- b. Press the seed icon and start picking in the scene. The auto-tracker follows the dip until it reaches the end of the line, or a fault stick it is not allowed to cross.
Optionally tracking is controlled by adding more seeds. Options:
 - I. Insert seed: Left mouse click.
 - II. Delete a seed: CTRL + Left mouse click.
 - III. Rename a framework horizon by double clicking the horizon (column) cell.
 - IV. Edit button in the table: controls the range over which the tracking error of the tracked events is smoothed in the mid-section between two seeds.
- c. Framework horizons can be removed from the table using the trash button.
- d. Pick mode is indicated by the mouse pointer and can be toggled ON/OFF with keyboard ESC. If pick mode is OFF the display can be changed:
 - I. Pan: click and drag with CTRL-Left, SHIFT-Left, or Middle mouse.
 - II. Rotate: click and drag with Left mouse.
 - III. Zoom: Rotate with middle mouse wheel.



Multi-seed picking of a horizon with the dip-steered auto-tracker.



A table (above) listing the picked seeds for a dip steered seismic horizon. It also allows changing the seed position (trace and z-value) and smoothing the tracking error between seeds. You can select multiple rows and delete all seeds (CTRL+A plus trash button, or select multiple entries while left mouse button is pressed followed by clicking the trash button).

Preview Regions

This feature is allows HorizonCube regions to be QC-ed. All regions are displayed as polygons overlying the mapped horizons and fault sticks. To update a region, either

manually adjust fault sticks (using OpendTect's fault picking tools), or adjust framework horizons using the dip-steered auto-tracker in the 2D HorizonCube Creator window.

You can customize the display settings for this region previewer through the settings sub-menu or icon.

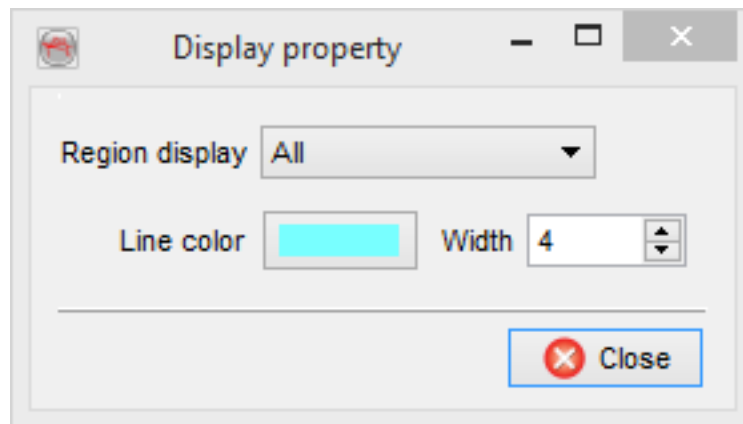
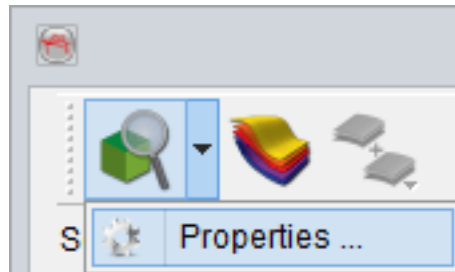


Regions can also be previewed from the data-driven settings dialog, see

[5.2.1.4.2 Processing 2D HorizonCube](#)

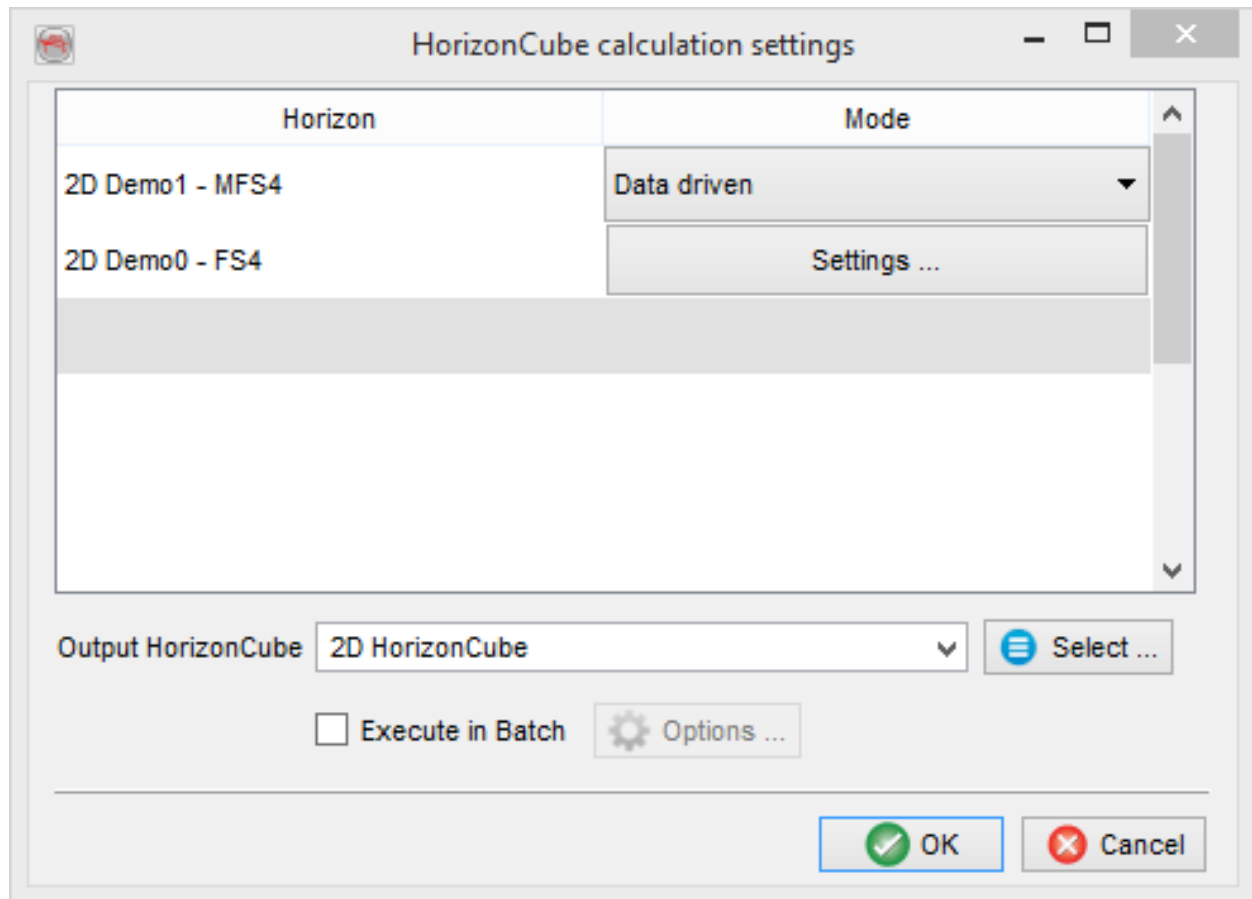
5.2.1.4.2 Processing 2D HorizonCube

Region display properties: visualization of the regions can be adjusted by changing the line color and line width. The changes will be reflected in the 3D scene.



Create/Update HorizonCube

To process a 2D HorizonCube, press the HorizonCube icon. The HorizonCube calculation settings dialog window is launched.



Model-driven Modes:

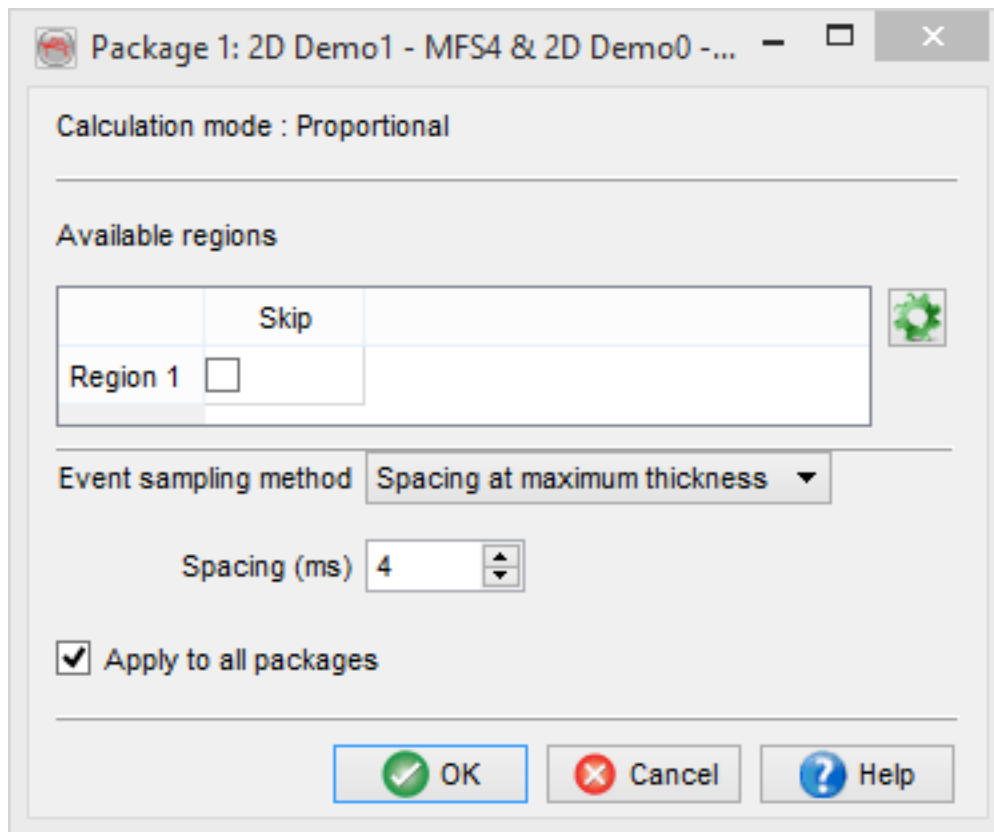
The three model-driven modes are:

- Parallel to Upper
- Parallel to Lower
- Proportional

Settings:

Pressing the *Settings ...* button launches the settings dialog window. Two Event sampling methods are supported: *Spacing at maximum thickness* and the *number of*

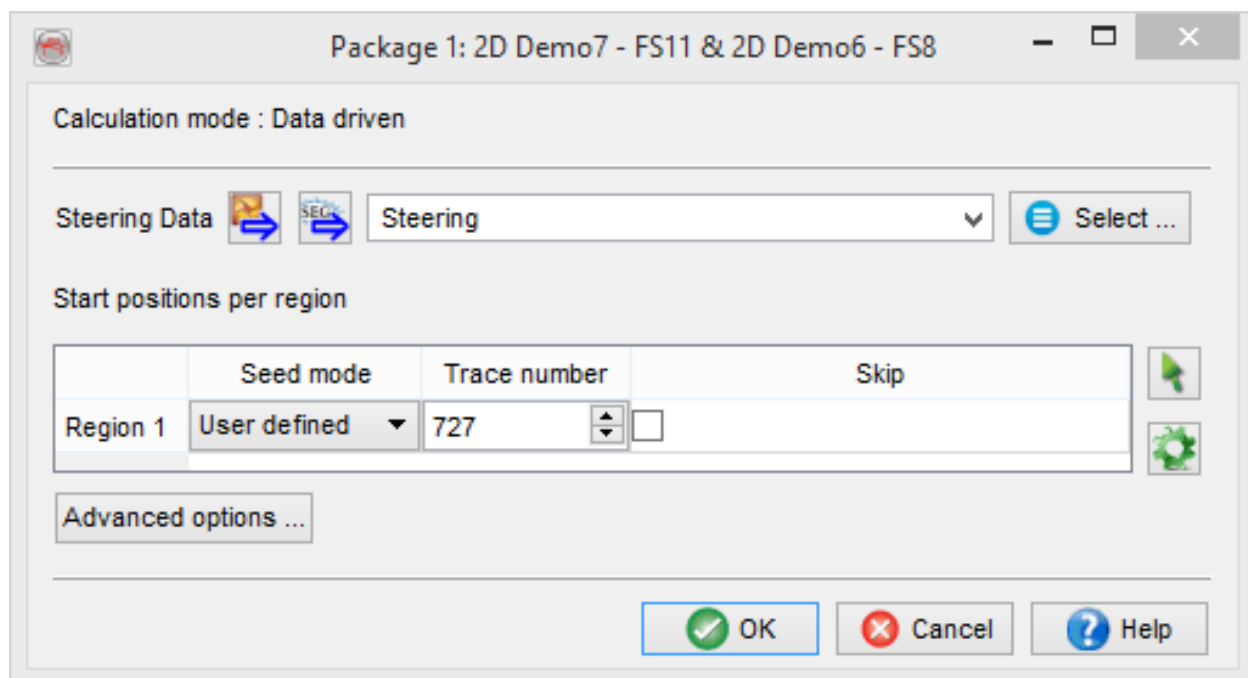
events for the package. More information can be found on [5.2.1.3.1 Model Driven Settings](#).



Data-driven Mode: In this mode dip steered horizons are auto-tracked to produce data-driven HorizonCube segments per region. The settings for this mode are defined per package.

Settings

Pressing the *Settings ...* button launches the settings dialog window for a particular package and updates the 3D scene by showing regions and the proposed starting positions for the tracker. Hovering over the scene highlights the active region.



Options

Steering – Select the steering input per package.

Start position – This is a crucial input parameter. The dip-steered auto-tracker simultaneously tracks all horizons from this position at the sampling interval specified in Advanced Settings. The starting position is displayed as a vertical line. This can be changed in the table but the recommended way to change the position is to interactively change this in the 3D scene. To do this press the green arrow icon. Now hover over the regions and QC and adjust the starting positions. Starting position should be selected in the thicker part of a region in good quality data areas and preferably not too close to a fault. The green wheel icon is used to change display settings of regions and start positions.

Skip a region – Regions can be skipped, e.g. in bad data areas. No HorizonCube is processed in skipped areas.

Advanced Options – See the section on [Advanced options](#).

Processing (on-the-fly/batch mode):

HorizonCube processing can be done:

On-the-fly:

Recommended for smaller 2D lines (less than 20Km) with few regions only.

A Wheeler Scene is launched automatically once HorizonCube processing is finished.

Batch-mode:

Recommended for long 2D lines (greater than 20Km), or lines with many regions.

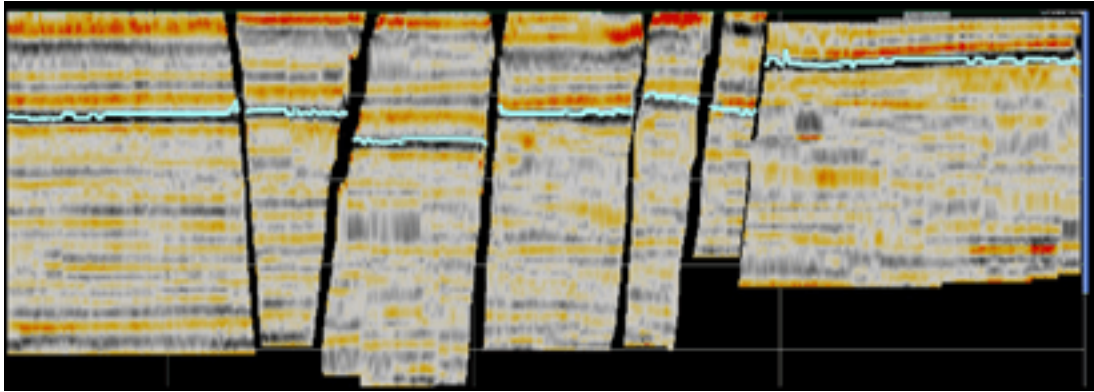
To inspect the data in the Wheeler domain a Wheeler scene must be added and filled manually.

Updating the Results after Processing:

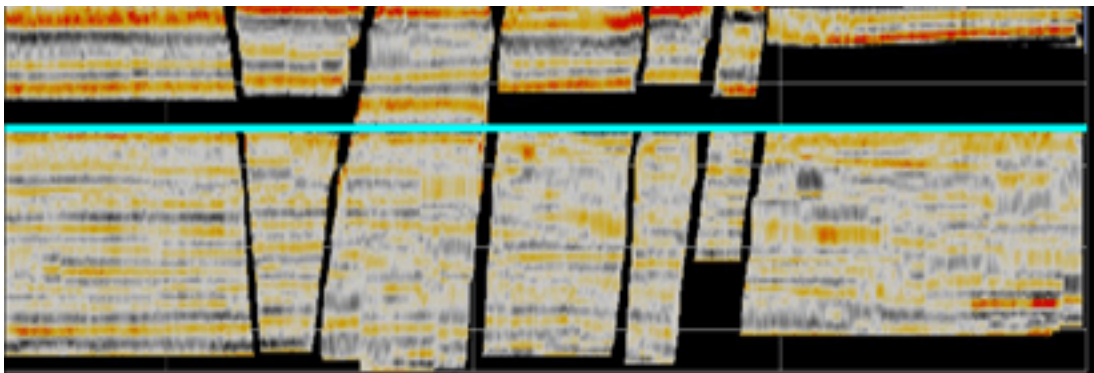
The Wheeler scene is an excellent tool to QC the HorizonCube processing results. Remember that segments are processed independently so that it cannot be expected that events are correlated over segments. Neither will the number of events be identical per segment. Correlation and merging will be done in a later step (see Correlating 2D HorizonCube Regions).

In this phase the Wheeler scene is used to study the flatness of seismic events. Dipping events in the Wheeler scene imply that the auto-tracker deviated from the actual seismic reflection patterns. HorizonCube processing can in that case be improved by adjusting the framework and rerunning the processing.

Framework horizons can be picked and update in the normal scene as well as in the Wheeler scene.




Picking a (dip-steered) seismic horizon in the Wheeler domain.

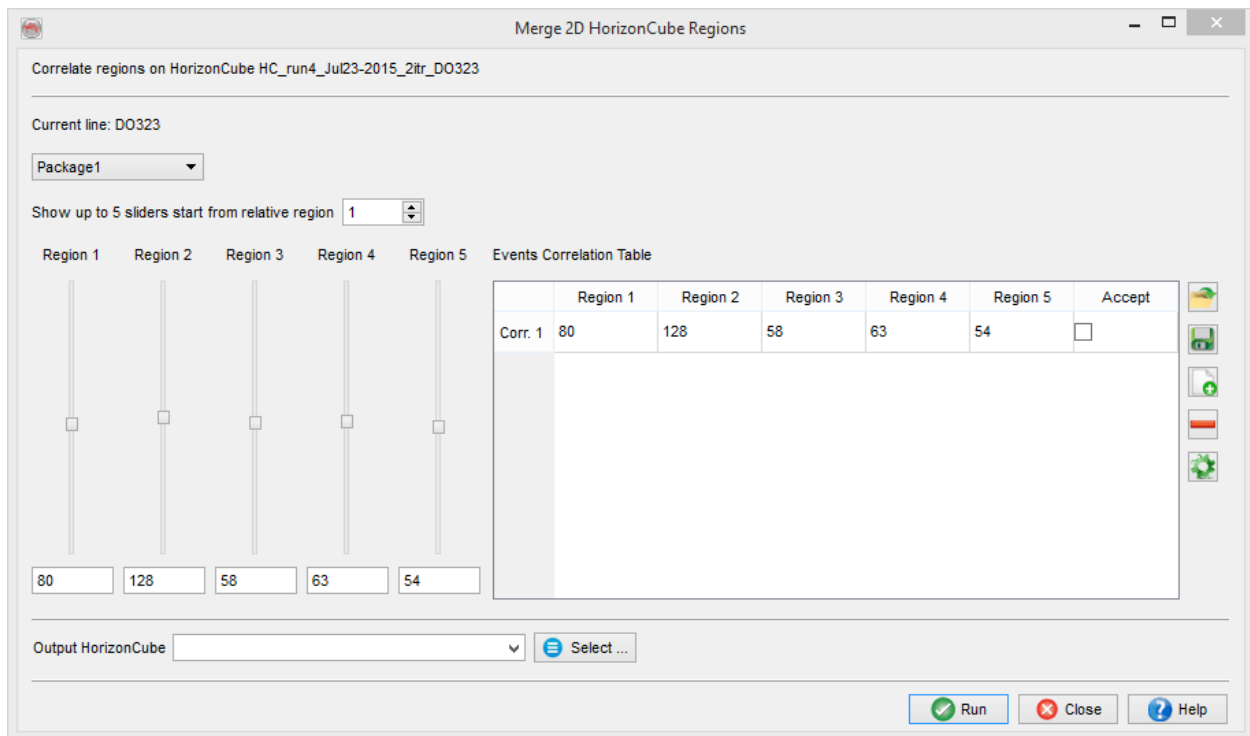


The updated Wheeler scene after reprocessing the HorizonCube.

5.2.1.4.3 Correlate 2D HorizonCube Regions

HorizonCube segments must be correlated laterally. This interactive process is performed using a set of sliders that control HorizonCube displays per region. The user correlates one or more events over all regions after which a new, merged HorizonCube is computed. The merged HorizonCube has an equal number of horizons per package.

The correlation window is launched from the correlation window icon  or from the Tools tab of the HorizonCube Control Center.



The sliders control the number of events per region to display. The sliders operate simultaneously in the normal scene and the Wheeler scene. Depending on the performance of the graphics card and the size of the 2D line to be manipulated

simultaneously updating both scenes might be too slow for comfort. If so, it is possible to work in one scene only and to iconize, or close the other one.

Correlation is performed per *Package* on a Region by Region basis. Move the first slider and position it on a prominent seismic reflector. Now move the second slider and position it on the same reflector in the adjacent region. Repeat this for the first five regions. Next reposition the sliders laterally by adjusting the *start from region* position in the spinbox. Continue the process until the event is correlated across all regions. When satisfied check the *Accept* toggle such that the correlation cannot be changed accidentally.

To add further correlations press the *plus* icon and repeat the process. To change an already accepted correlation uncheck the *Accept* toggle and adjust the sliders. To remove a correlation uncheck the *Accept* toggle and press the *trash* icon.

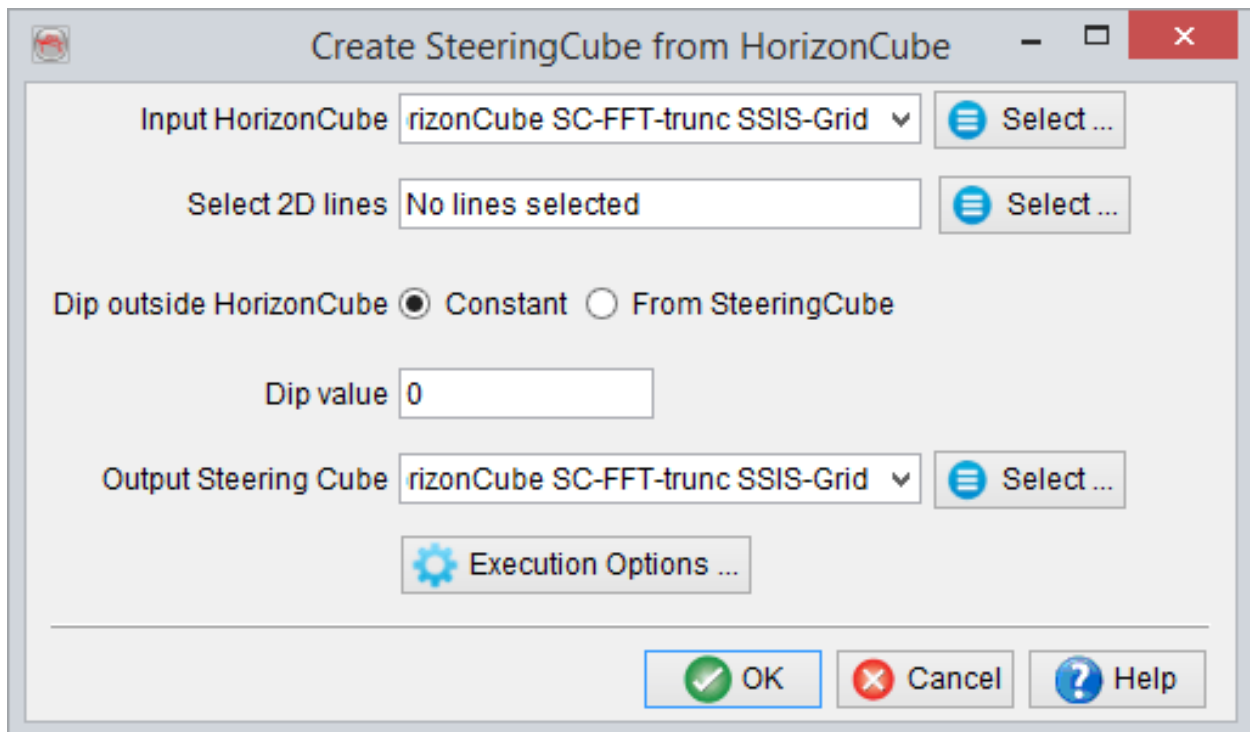
5.2.1.5 Create HorizonCube - Multiple Lines

Similar to 3D this tool allows to create one HorizonCube per 2D line. If the start position is set to user defined, it is recommended to select one line at a time, only one inline-crossline pair can be specified per line. Further the existing HorizonCube can be overwritten by adding more lines.

For more information refer to [Create a HorizonCube](#) section.

5.2.1.6 Convert To SteeringCube

A HorizonCube can be transformed into a SteeringCube using this tool. The input HorizonCube is selected first. The Volume sub-selection is optional. The dip values outside the area of the selected HorizonCube can either be filled with a constant dip value or read from another SteeringCube. The output name for the SteeringCube is provided in the output field.

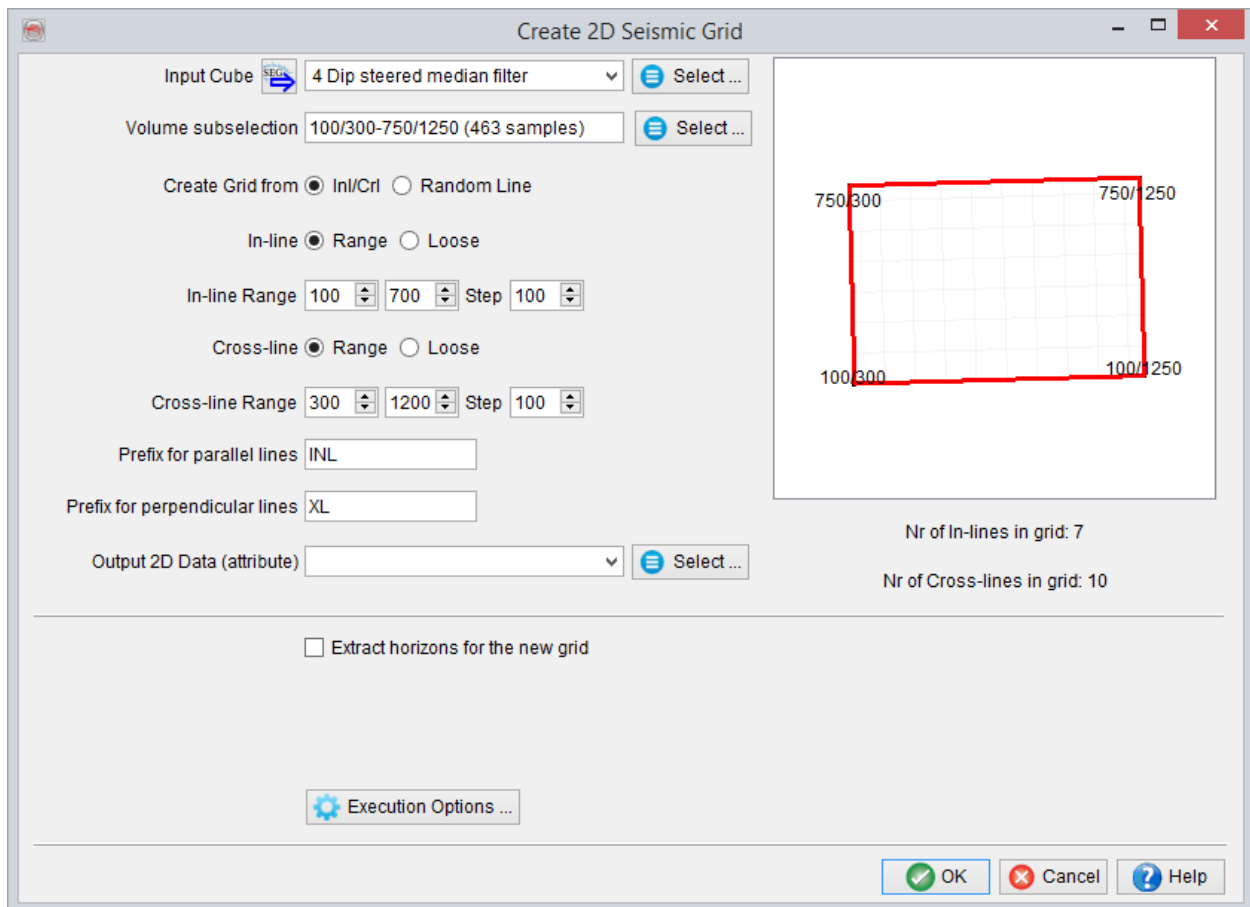


The screenshot shows a dialog box titled "Create SteeringCube from HorizonCube". It contains the following fields and controls:

- Input HorizonCube:** A dropdown menu showing "rizonCube SC-FFT-trunc SSIS-Grid" and a "Select ..." button.
- Select 2D lines:** A text field containing "No lines selected" and a "Select ..." button.
- Dip outside HorizonCube:** Two radio buttons: "Constant" (selected) and "From SteeringCube".
- Dip value:** A text field containing "0".
- Output Steering Cube:** A dropdown menu showing "rizonCube SC-FFT-trunc SSIS-Grid" and a "Select ..." button.
- Execution Options ...:** A button with a gear icon.
- Buttons:** "OK", "Cancel", and "Help" buttons at the bottom right.

5.2.1.7 Create 2D Seismic Lattice

This tool is used to create a 2D lattice/grid from 3D seismic data. This feature is released as a Multi2D workflow in the SSIS plugin (Chapter 7). The workflow is simple i.e. convert 3D seismic data into a coarse 2D grid with preferred orientation and geometry. After the conversion, the next step is to prepare a HorizonCube for all 2D lines. Using the HorizonCube for each 2D line, interpret sequence stratigraphic surfaces. The extracted sequence stratigraphic surfaces are then used to do detailed interpretations on the 3D seismic volume.



2D Seismic Grid options

- **Input Cube:** Selected 3D seismic data to be used in this field.
- **Volume Subselection:** Optionally, the 3D volume can be restricted to an objective area. This subselection is made here by restricting inline/crossline/time ranges for the selected volume.
- **Create Grid from:** Define the output grid geometry and orientation of the 2D lines to be created. If Inl/Crl is selected for this option, the general inline/crossline orientation (or geometry) of the selected 3D data would be used within the defined inline/crossline range and the corresponding steps. Optionally, the inline/crossline range can be edited manually by setting loosely spaced inline/crossline numbers separated by commas.

Create Grid from Inl/Crl Random line

Inline Range Loose

Inlines (comma separated)

Crossline Range Loose

Crosslines (comma separated)

Another option is to define a 2D grid geometry using a *Random Line*. Using the selected random line, the grid can be created in both (parallel and perpendicular) directions of the randomline. This fixed spacing needs to be given in the *line spacing (m)* fields.

The screenshot shows a software interface for creating a 2D grid. At the top, there are two radio buttons: "Inl/Crl" (unselected) and "Random line" (selected). Below this, there is a text input field labeled "Input RandomLine" with a dropdown arrow and a "Select ..." button. Underneath, there are two spinners: "Parallel line spacing (m)" set to 500 and "Perpendicular line spacing (m)" set to 500.

Random Line options

- **Prefix:** Label the 2D-line names in the fields.
- **Output Lineset:** Output for the 2D lineset. Please provide the lineset name and the name for the seismic data (attribute).
- **Extract Horizons:** Convert the 3D interpreted horizons into 2D horizons by checking this box (optional). In the *Select horizons list*, select one or more horizons.

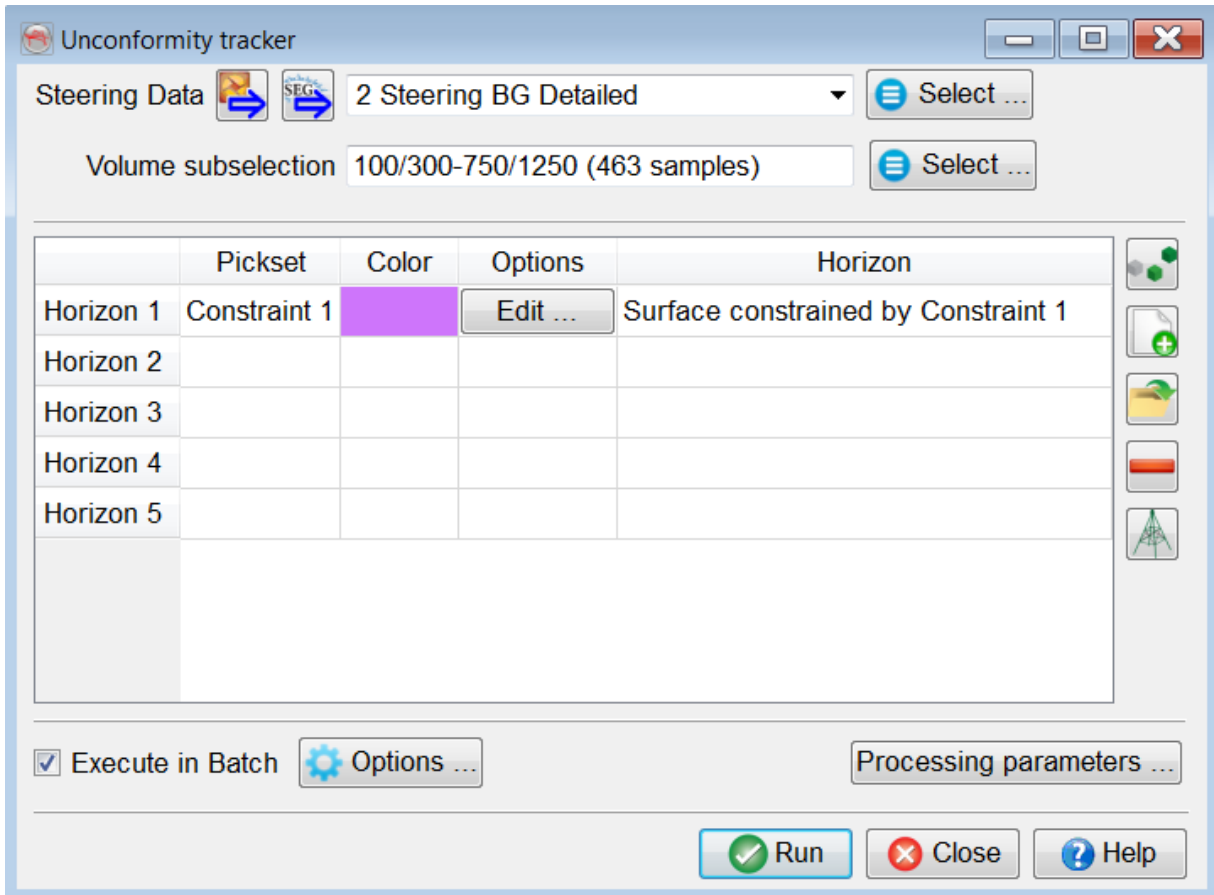
5.2.1.8 Create Horizons From A SteeringCube

This seismic horizon tracker generates one or more horizons from a Steering Cube (dip volume). The tracker uses a constrained, inversion-based algorithm that globally minimizes the error between horizon dips and seismic dips (Wu and Hale, 2015).

A confidence weight volume is recommended for faulted intervals. A good confidence weight volume is given by the “Planarity” attribute. Planarity is computed in the “Faults & Fractures” plugin, It is a measure of “local flatness” of a seismic event. When used as a confidence weight volume in the unconformity tracker it assigns low weights at fault positions and high weights at good reflectors. A flipped discontinuity attribute like similarity, semblance, or Thinned Fault Likelihood can be considered as alternatives to planarity.





The unconformity tracker is a general-purpose horizon tracker that can be used among others to:

- Map unconformities (events that change laterally and are therefore difficult to map using conventional amplitude / similarity trackers)..
- Map horizons at intra-reservoir scale levels that tie exactly at given well markers.
- Fast-track mapping of seismic events, e.g. for constructing a structural framework, to constrain a HorizonCube, or to build low-frequency models for seismic inversion.



Main window for the Unconformity Tracker

The workflow can be started in three ways:


- Insert a new horizon in the table for picking .
- Click on the “Pick Seeds” icon  and manually pick a set of locations.
- Click on the “Load existing Pick Set” icon  to open an existing pick set.
- Click on the “From Wells” icon  and select one or several marker sets and the wells.

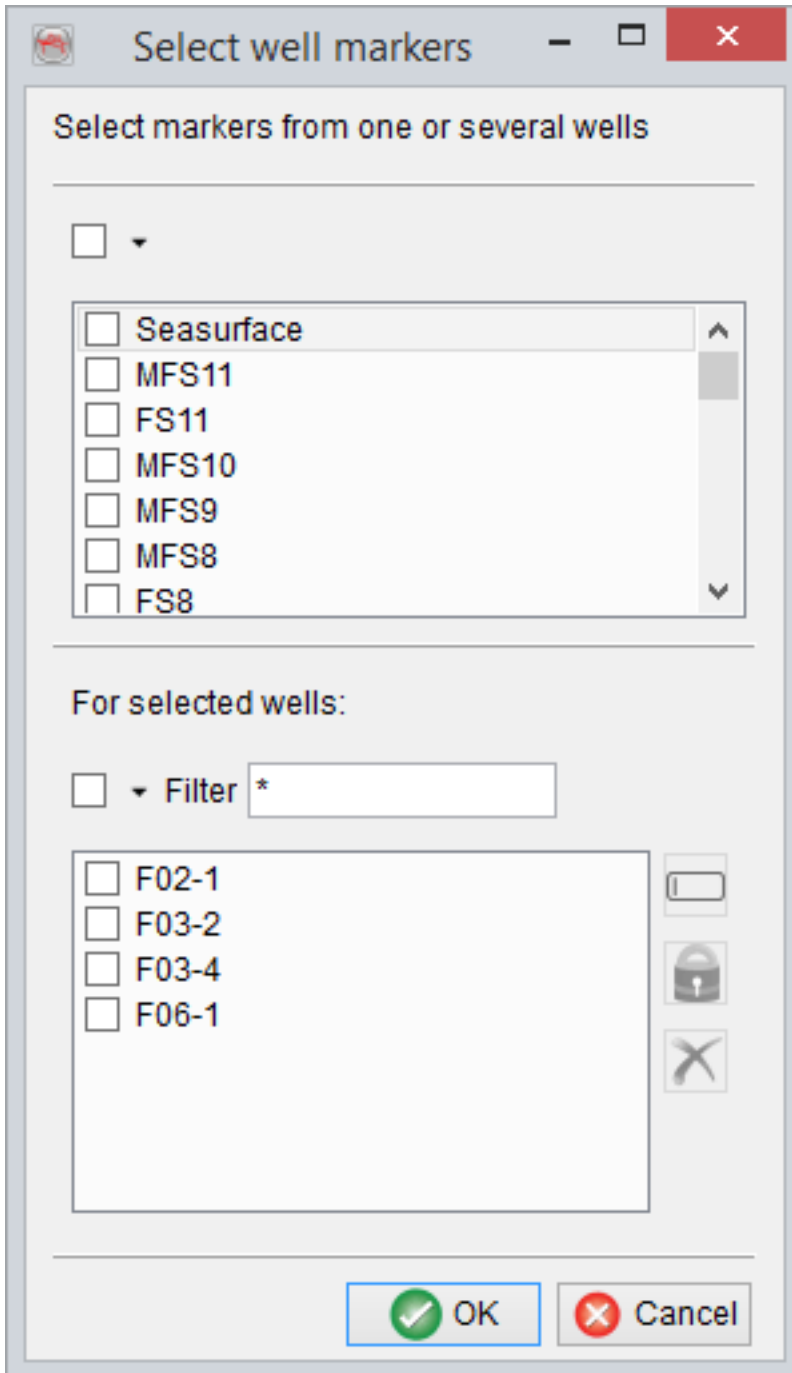
Seeds can be added or removed thereafter regardless of the origin of the pickset, either interactively or using the table editor under the Edit options. At least one seed is required for each horizon to be tracked using this tracker.

Inputs

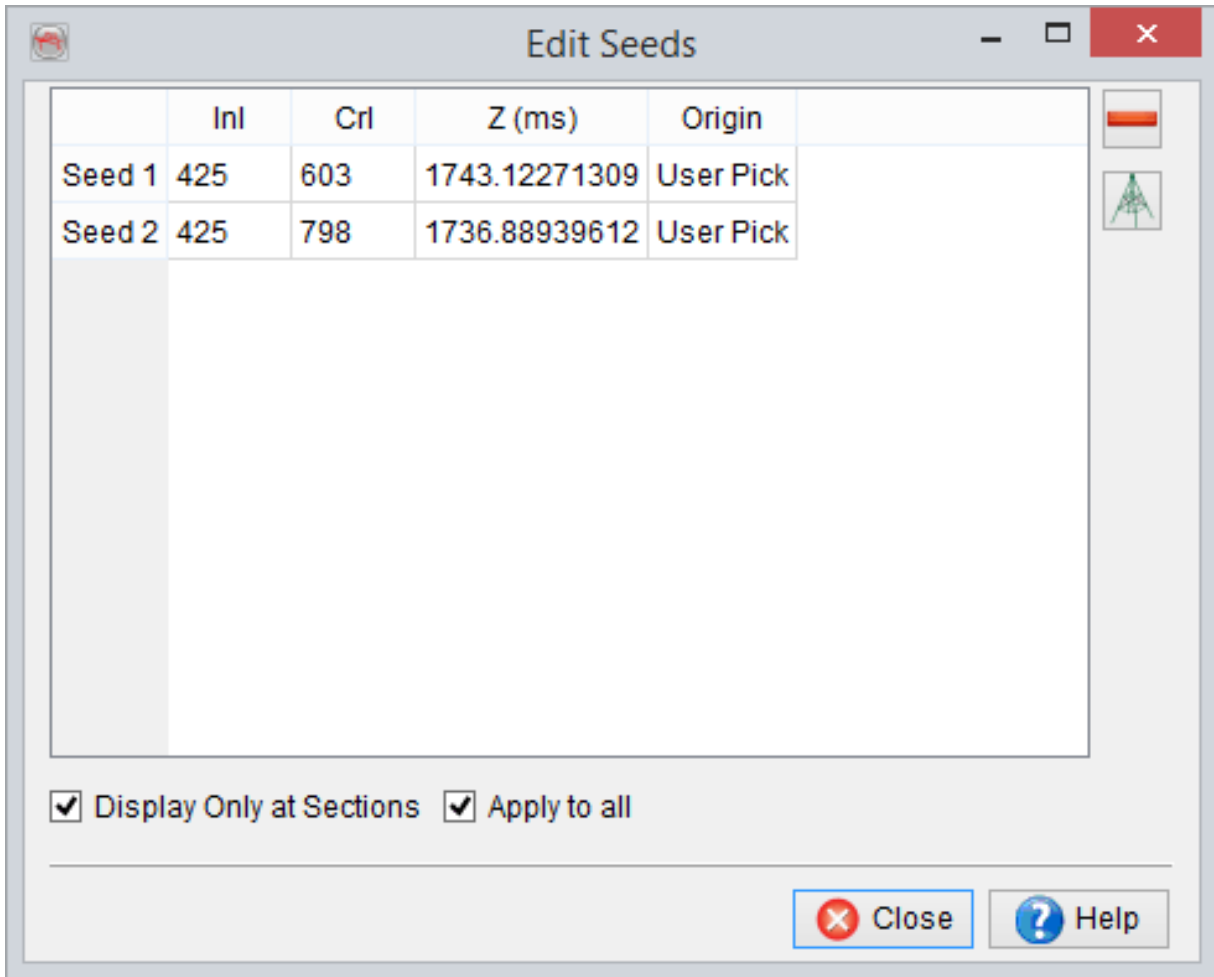
- A pre-processed SteeringCube is mandatory. This SteeringCube must be filtered to have less noise. A typical Detailed SteeringCube should be a good starting point.
- Set of locations (from well makers and/or manually picked positions).
- Optional: confidence weight volume (planarity, flipped Thinned Fault Likelihood, ...).

Adding Well Markers

Multiple markers from multiple wells can be selected in the Select well markers window . Each selected marker adds a row to the table in the main window. The set of (manually-picked / markers) locations can be QC-ed and edited by pressing the Edit button. To add more picks, select the relevant row, press the Pick Seed button and manually pick positions to add.

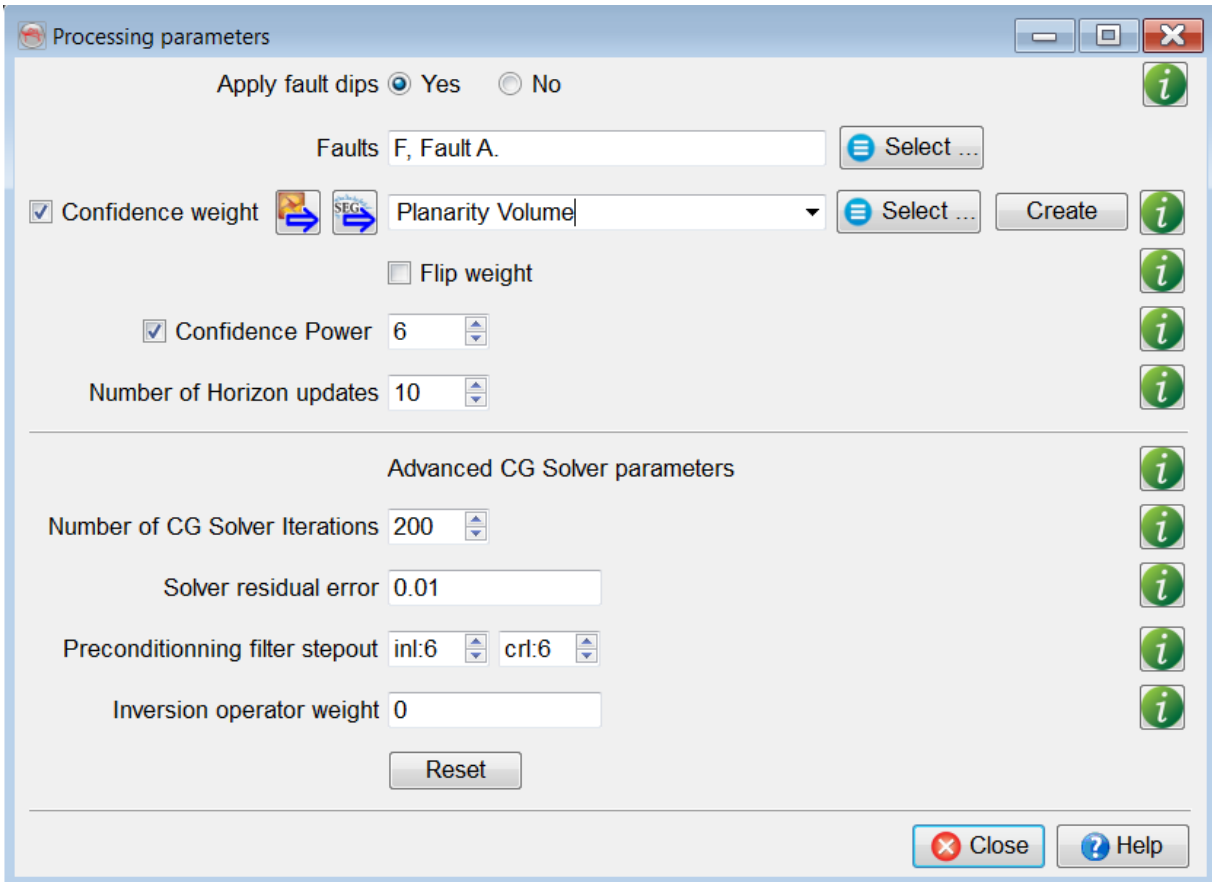


Selection of well markers is done here. Multiple markers can be selected. Each marker will create a horizon.

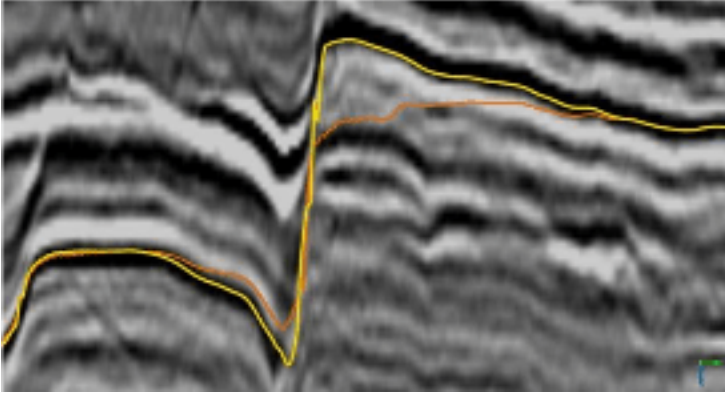


Edit Seeds window will show the origin (user defined or wells) of the picks. By default the seeds are displayed on sections (e.g. inlines/crosslines only). To view in 3D, you may want to toggle this feature off.

Processing Parameters



- A confidence weight volume is optional. It is only required if the interval is faulted. If the given volume has higher weights (e.g. near to 1) at the fault weights, you may flip the weights ($1 - \text{input}$) to minimize the effect of erroneous dips at fault locations.
- A power function is applied to the weights. A higher value increases the contrast between faults and continuous reflectors.



This result is produced with a confidence weight volume. With a power of 8 (yellow horizon) the event is correctly located when compared with the power of 2 (orange horizon).

- The *Number of Horizon updates* field (max. 100) is the number of times the horizon is changed and horizon dips are re-calculated. The quality of the results increase logarithmically with increasing number of horizon updates. After a certain number of updates, typically 10, the horizon will effectively not change further. If such a situation occurs, we consider that the results are fully converged and the best possible solution is achieved.
- The core of the algorithm is a preconditioned Conjugate Gradient ‘CG’ solver, which attempts to solve a system of linear equations. This algorithm is controlled with the following parameters:
 - **Iterations:** The maximum number of iterations the GC solver is run to solve the inverse problem.
 - **Residual error:** The threshold value below which the GC solver stops.
 - **Preconditioning filter stepout:** Is a recursive filter to smooth the horizon and minimizes abrupt jumps (max. 10).
 - **Inversion Operator Weight:** This parameter is trivial and helps in overriding the dips in bad data quality areas. If it is larger than 0 (max. 1), the weighted dip derivatives are subtracted from the dips obtained from

horizons to minimize the erroneous dip effects. By default, we do not recommend adding such weights. The Confidence Weight volume mostly controls such effects.

Larger Surveys

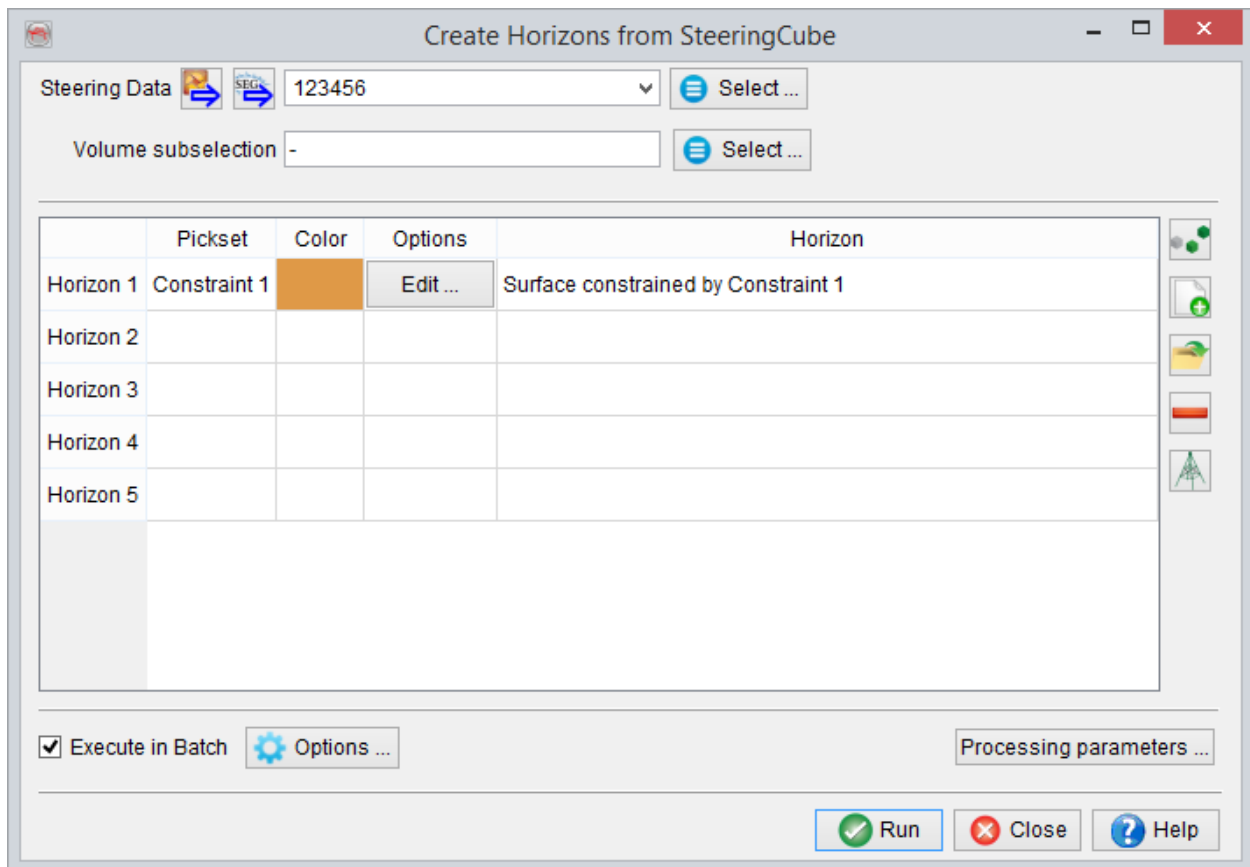
For larger datasets (e.g. volume size > 5GB) with limited RAM, we recommend decimation of the results. This can be done in the volume sub-selection. One may upscale the lateral stepouts (Inline/Crossline) to produce a horizon.

Reference

Wu, X., and Hale, D. (2015) Horizon Volume with Interpreted Constraints, *Geophysics*, v80, Issue 2, IM21-IM33.

5.2.1.9 Edit Horizons From SteeringCube

To fill holes or to re-track an existing horizon using SteeringCube, the similar utility can be launched from the HorizonCube Sub menu (Processing > HorizonCube > *Edit horizons with SteeringCube*). This will bring the following window. This window is almost identical to the Create Horizon(s) from SteeringCube window. The SteeringCube is selected for the *Input SteeringCube* field. Additionally, the faults can also be selected. The *Edit Options* are used to either fill the holes/gaps in an horizon using SteeringCube or to re-track a horizon using SteeringCube.



Edit existing horizon(s) using the selected SteeringCube.

5.2.1.10 Interpolate Logs Along A HorizonCube

This gridding step is used to populate a 3D volume using well log(s). The HorizonCube provides the necessary steering to guide the interpolation of the well logs. This is equivalent to conventional gridding, but that would take place in the [Wheeler domain](#) where all seismic events are flat.

The screenshot shows a software dialog box titled "Edit step" with a standard Windows-style title bar (minimize, maximize, close buttons). The main content area is titled "HorizonCube-Well Interpolator".

At the top, there is a dropdown menu for "Input HorizonCube" with a "Select ..." button to its right. Below this is another dropdown menu for "Vertical Extension" set to "By parallel to top/base".

Underneath, there are radio buttons for "Log extension if needed", with "No" selected.

The central part of the dialog features a table with two columns: "Well" and "Log". The table is currently empty. To the right of the table are two buttons: "Add ..." and "Remove ...".

Below the table, there is a dropdown menu for "Algorithm" set to "Inverse distance".

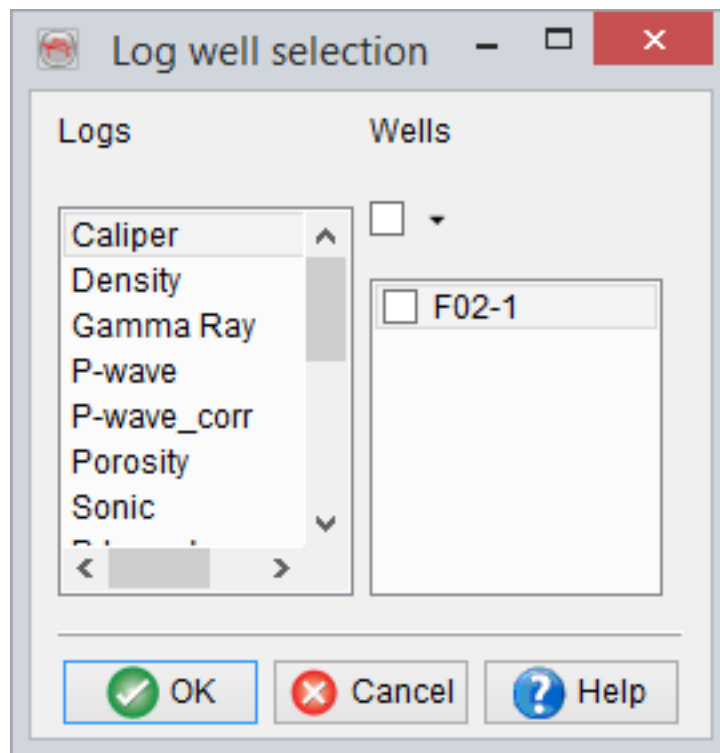
There is a checkbox for "Search radius (m)" which is currently unchecked, followed by an empty text input field.

At the bottom of the dialog, there is a text input field labeled "Name for this step".

The bottom right corner contains three buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a blue question mark icon).

The HorizonCube well log interpolator includes two fields to be provided by the user: a HorizonCube and a table of well logs. Select a 3D HorizonCube by pressing the 'Select' button.

To select wells and corresponding logs, press 'Add' button. To remove a selected well/-log, select it and press the 'Remove' button. To change the log, select and press the 'Change log' button. It is easier to first select the wells; The list of logs will then get updated and present only the logs that are common to all selected wells.



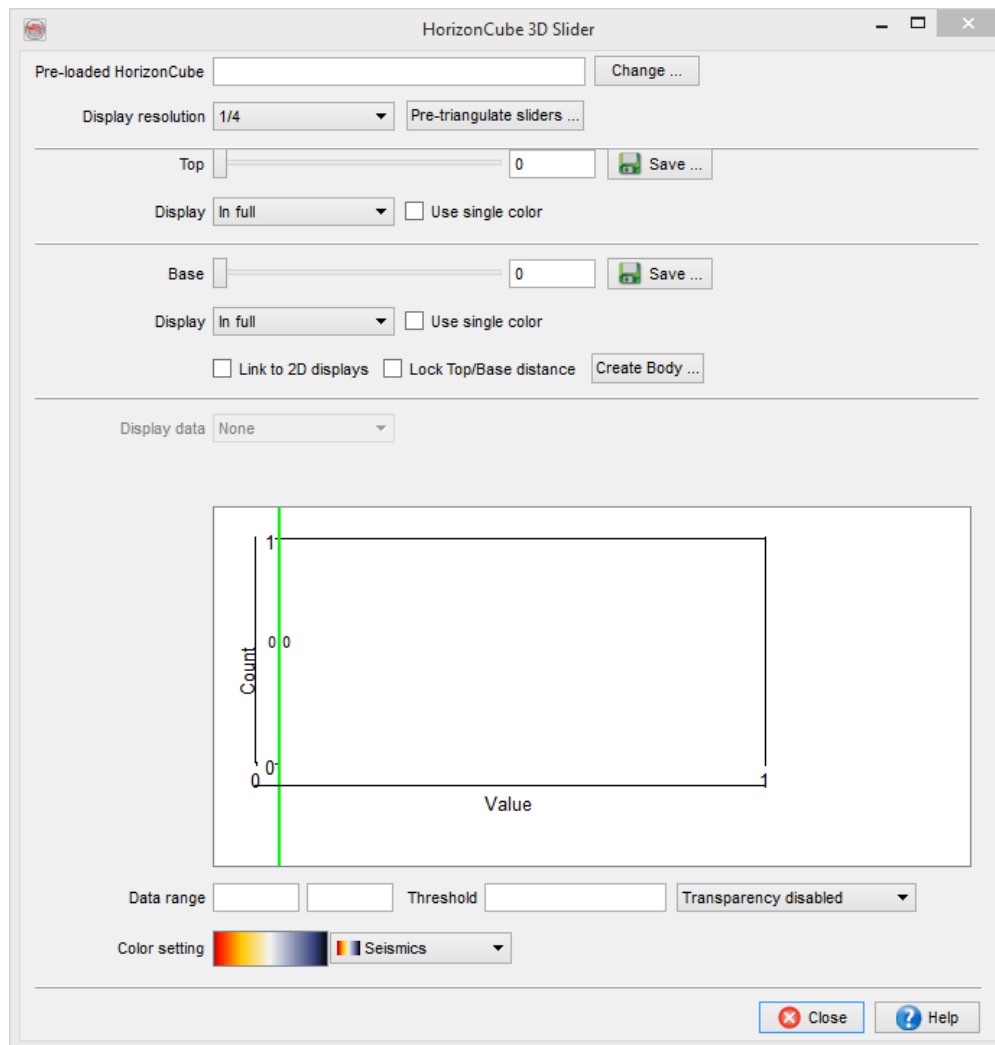
After the selection of both wells and the HorizonCube, provide a name for this step at the bottom and proceed to the Volume Builder by pressing OK.

5.2.2 Analysis

See the following chapters below.

5.2.2.1 3D Slider - HorizonCube Visualization

A tool that is not only used to visualize the HorizonCube events in 3D but also to interpret depocenter, pinchouts, geologic bodies based on thickness /attribute map views along the events. The thicknesses are computed on the fly and are displayed as a grid on the horizons. From the isopach maps one may furthermore create *bodies*. Another benefit of the 3D slider is to save the key events as conventional OpendTect horizons.



Preload HorizonCube

A user must preload a HorizonCube into the memory.



If the HorizonCube is big and there is insufficient RAM installed on the system, you may preload a package.

Top/Bottom Slider(s)

There are two sliders available to display the corresponding events in the scene and also to compute the isopach thicknesses between the events. The user has an option to hide either event.

Link to 2D displays

It links the (top/bottom) sliders movement to the sections (inlines/crosslines) on which the preloaded HorizonCube is displayed. So, if the slider is moved, the corresponding HorizonCube overlay on sections will also be moved.

Lock top-bottom distance

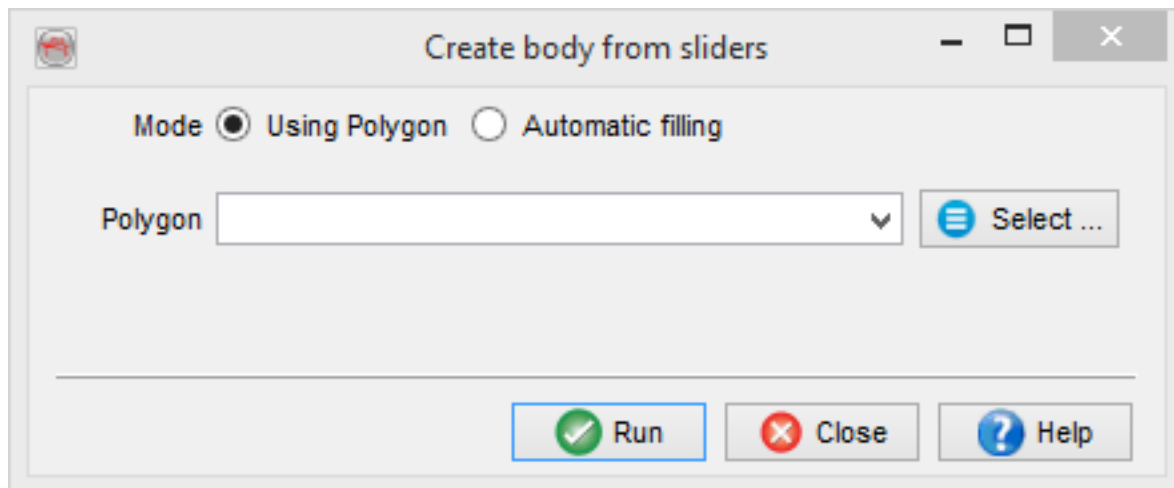
It is used to lock the number of events between the sliders positions. Once the number of events are locked, one may use only one slider to compute the isopach between them.

Body

It is used to create a body object within OpendTect. The body can be created either within a polygon or by using mentioning the threshold value. The threshold value is the position of the red line in the histogram of the 3D Slider. The polygon is selected if a pre-defined polygon exists. If not, you will have to create a polygon first (for details please read the pickset/polygon section of the User Documentation). The second mode i.e. *Automatic filling* requires a threshold value. For instance, if one wants to create the bodies of all isopach values that have 0.25sec thickness, 0.25 should be given as a

threshold. Optionally, one could move the red line in the slider to provide that value. The body is filled either with values that are below the threshold or with the values that are above the threshold. These are set in the *Body value* radio boxes.

The body is created within a selected polygon and between the two HorizonCube events positioned using the slider.



Create bodies from the 3D Slider

Surface data

💡 *Nothing will be computed on the displayed (Top/base) events in the scene.*

Depth: Will display the Z-values (TWT / Depth) along the events.

Top-Base Isopach: It *computes* the isopach thickness between the two events selected by a user at particular sliders position. Once the user presses the *Calculate* button, the histogram of the thickness is displayed. The histogram display can be used to define the transparency on the thickness map. The green lines on the top of the his-

tograms define the clipping of the histogram. The red line on the histogram is used to define the transparency.

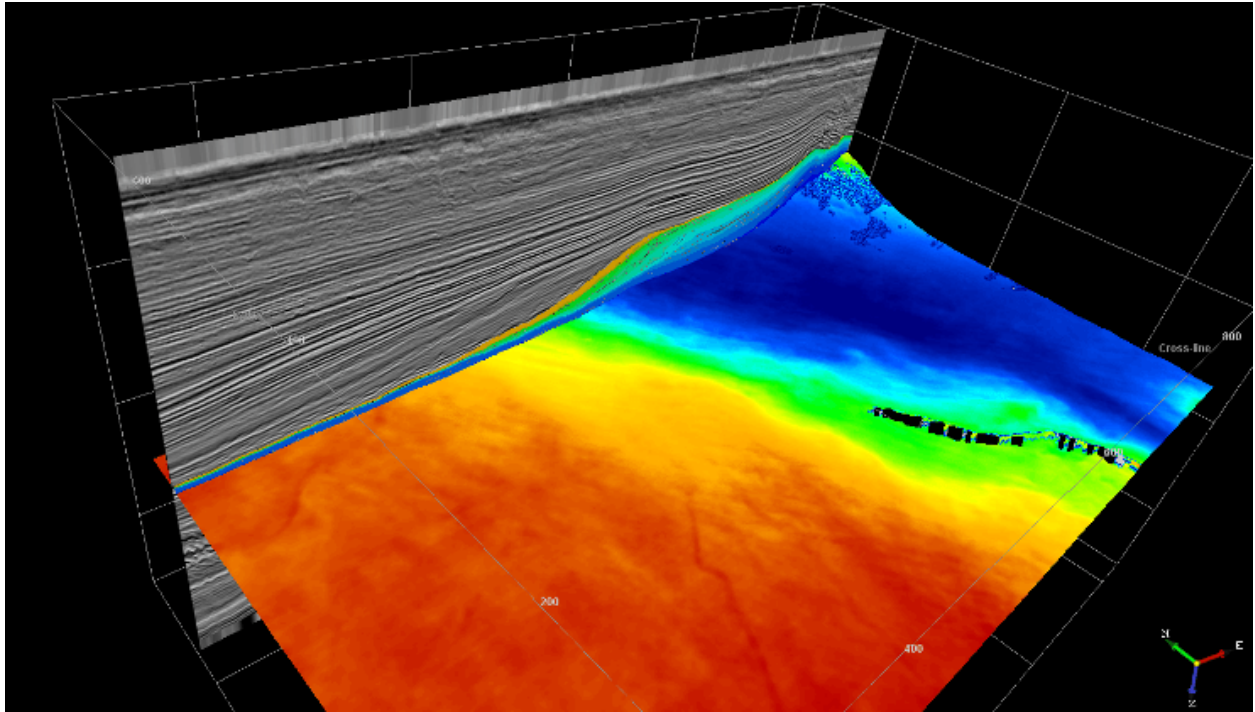
New Attribute: It is used to compute a stored seismic volume or a defined attribute along the horizons.

Transparency

This field defines the transparency range on a colour bar. The first two fields adjacent to the transparency are the colour ranges. The third field is the transparency cut-off value. Disable option in the list box defines no transparency, above defines transparency above a given cut-off value and below defines transparency below the cut-off value.

Colour bar

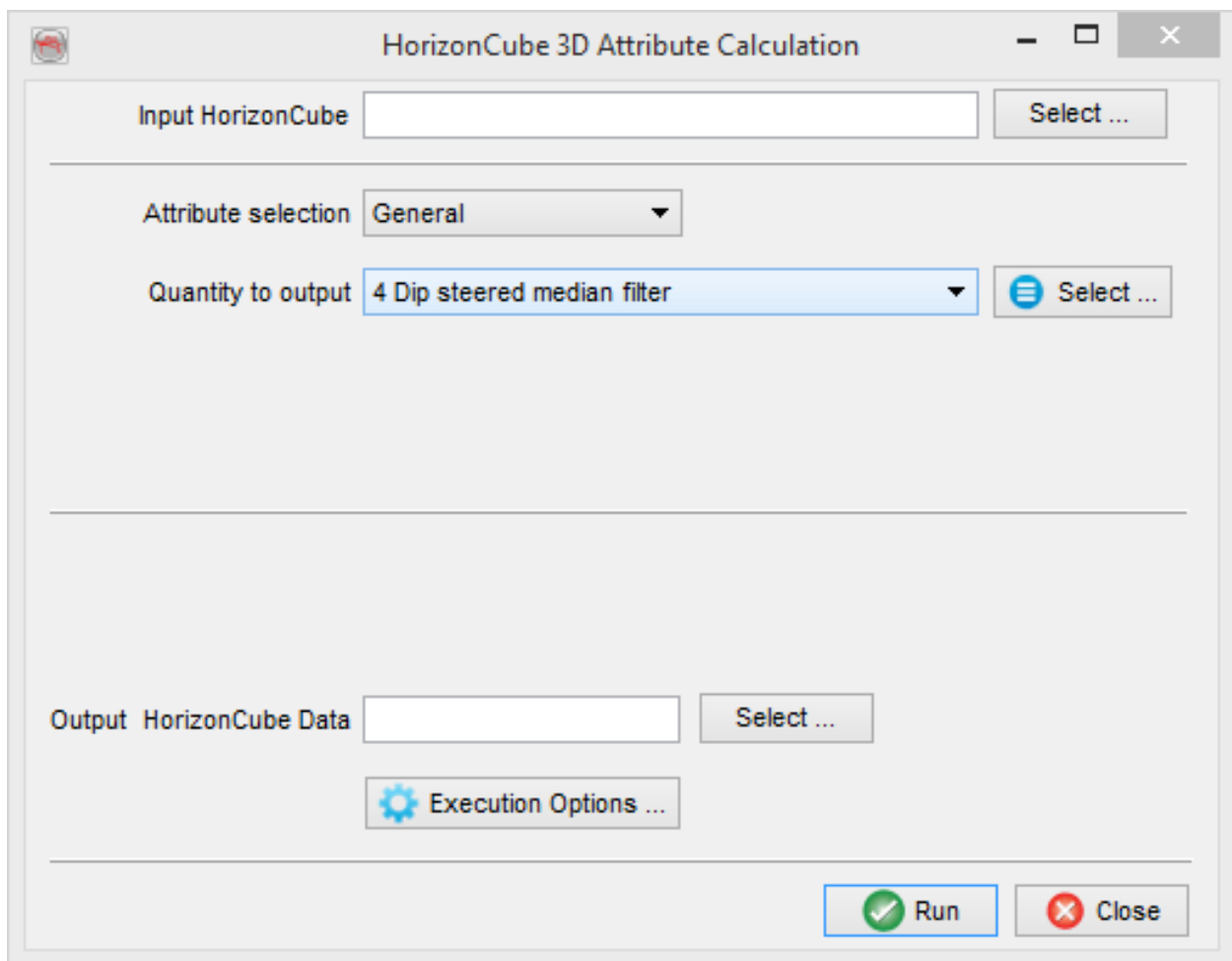
The colour ranges and the corresponding selected colour bars are available at the bottom. The active colour bar is used to display the thickness map on the displayed horizons. The colour bar can interactively be changed by scrolling the available list of pre-defined colour tables. Furthermore, one may right click on the colour bar and use the pop-up menu that work similar to the general OpendTect colour bar.



5.2.2.2 Extract Attributes Along HorizonCube

These attributes are computed from the [HorizonCube Control Center](#). The output of all of these attributes is a set of grids, which are collectively saved as [HorizonCube Data](#). These attribute grids can be visualized using [3D Slider](#).

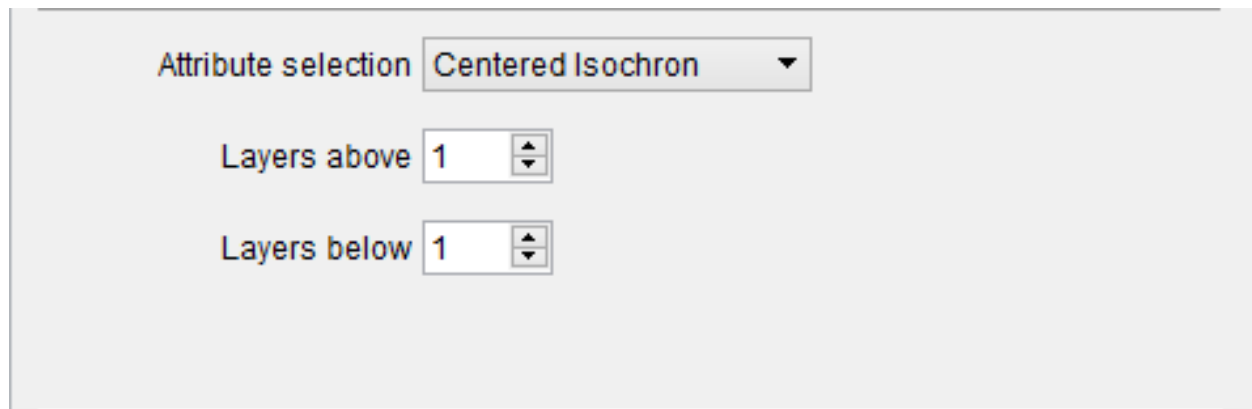
In the *Select HorizonCube* option, you can select the entire HorizonCube, or a package or an event.



The screenshot shows a software dialog box titled "HorizonCube 3D Attribute Calculation". The dialog has a standard Windows-style title bar with a minimize button, a maximize button, and a close button. The main content area is divided into several sections by horizontal lines. The first section contains an "Input HorizonCube" text box followed by a "Select ..." button. The second section contains an "Attribute selection" dropdown menu currently set to "General". The third section contains a "Quantity to output" dropdown menu currently set to "4 Dip steered median filter", followed by a "Select ..." button with a blue circular icon. The fourth section contains an "Output HorizonCube Data" text box followed by a "Select ..." button. Below this is a button with a gear icon labeled "Execution Options ...". At the bottom right of the dialog are two buttons: "Run" with a green checkmark icon and "Close" with a red X icon.

General: This category is used to compute any existing attribute (stored or defined) along HorizonCube events. For instance, if you already have a stored AI or PHI volume and you would like to compute it along HorizonCube, you can use this option.

Centered Isopach: This attribute calculates the vertical thickness (TWT/depth) between two events such that they are centered by an event. In this manner, the thickness is computed along the centered event. Number of layers above/below define the relative number of events to compute the thickness. For instance, if the value for layers above/below is set to 1, for each central event, the thickness is computed by subtracting the TWT/Depth of an underlying event from the overlying event and storing the result for the central one.



The screenshot shows a user interface for selecting an attribute. It features a dropdown menu labeled "Attribute selection" with "Centered Isochron" selected. Below this are two spinners: "Layers above" and "Layers below", both set to the value "1".

Topographical Curvature: This attribute defines geometrical curvature of an event, attribute or HorizonCube Data. The stepouts are number of inlines/crosslines to be used to compute the curvature. These attributes are mostly equivalent to the [conventional curvature attributes](#) that are directly computed from the SteeringCube.

Attribute selection TopoLogical Curvature ▼

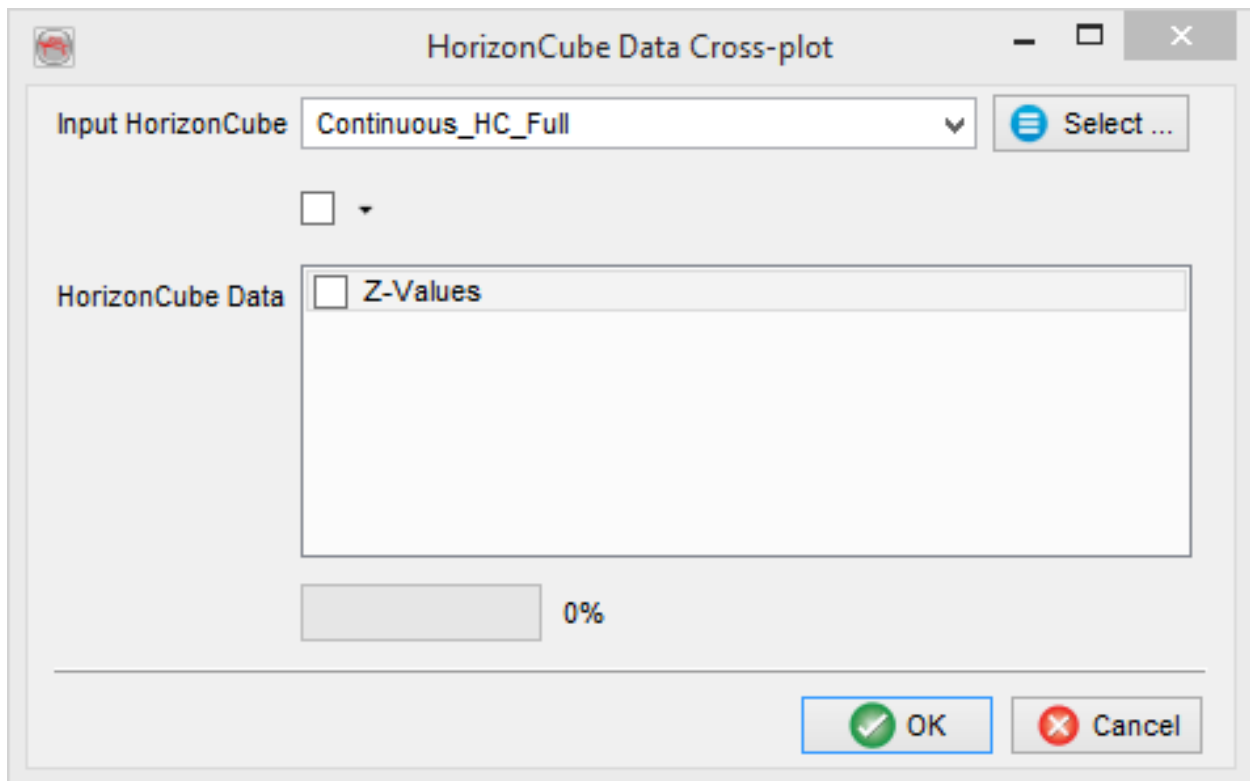
Inl/Crl stepout 1

Algorithm Mean curvature ▼

5.2.2.3 Make Crossplots Of HorizonCube Data
















This tool is used for creating crossplots out of HorizonCube data (see how to extract attributes along HorizonCube at [Extract Attributes Along HorizonCube](#) page).

In the *HorizonCube Data Cross-plot* window select the input HorizonCube and the data for cross-plotting.



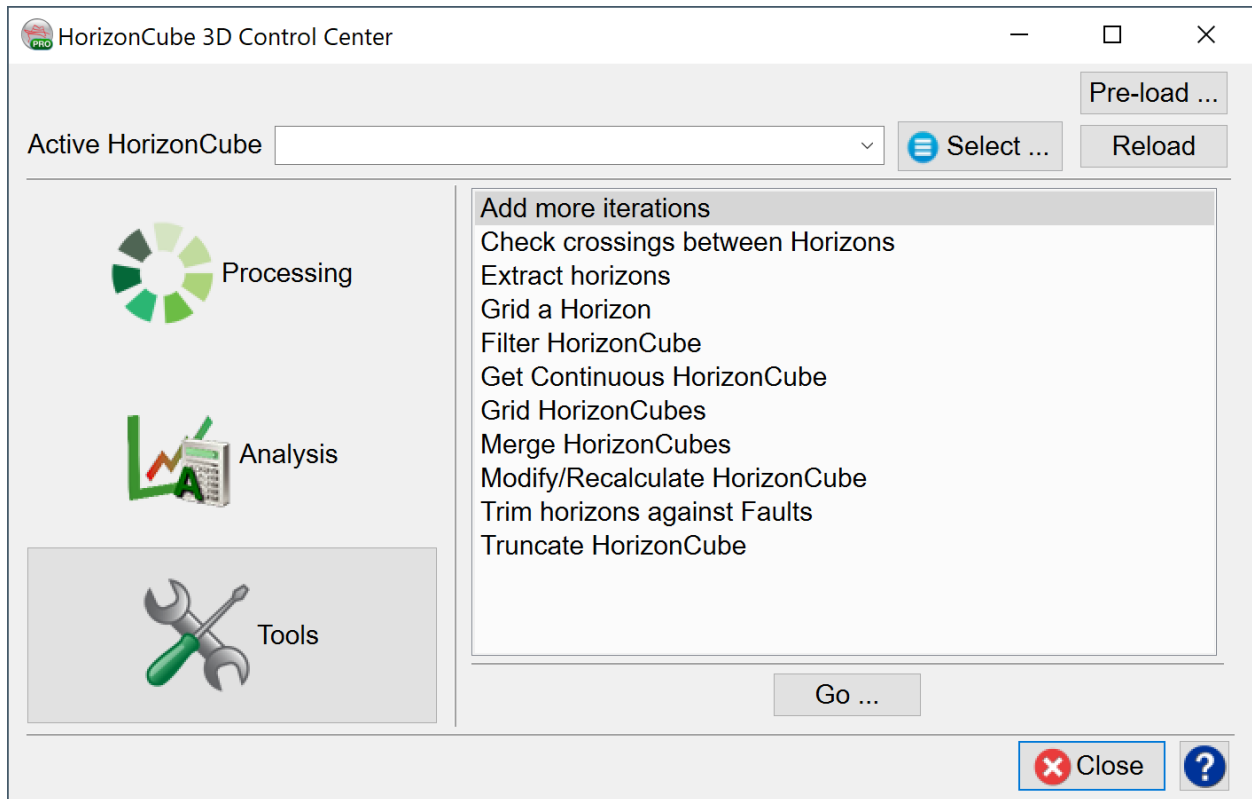
Proceed to launch the cross-plot table window, which is used to edit and plot the data for a cross-plot. For more information about cross-plotting refer to this document.

Continuous_HC_Full












 Show 0.7%
 

 Show In/Cri
 


	X-Coord	Y-Coord	Z (ms)	* Event index	Z-Values
105/303	605906.99736378	6073683.42579284		2.1098765e9	105
105/454	609680.52567399	6073788.88907666		2.1098765e9	105
105/605	613454.0539842	6073894.35236049		2.1098765e9	105
105/756	617227.58229441	6073999.81564431		2.1098765e9	105
105/907	621001.11060462	6074105.27892813		2.1098765e9	105
105/1058	624774.63891483	6074210.74221196		2.1098765e9	105
105/1209	628548.16722504	6074316.20549578		2.1098765e9	105
106/415	608705.20776206	6073786.63987045		2.1098765e9	106

5.2.3 Tools

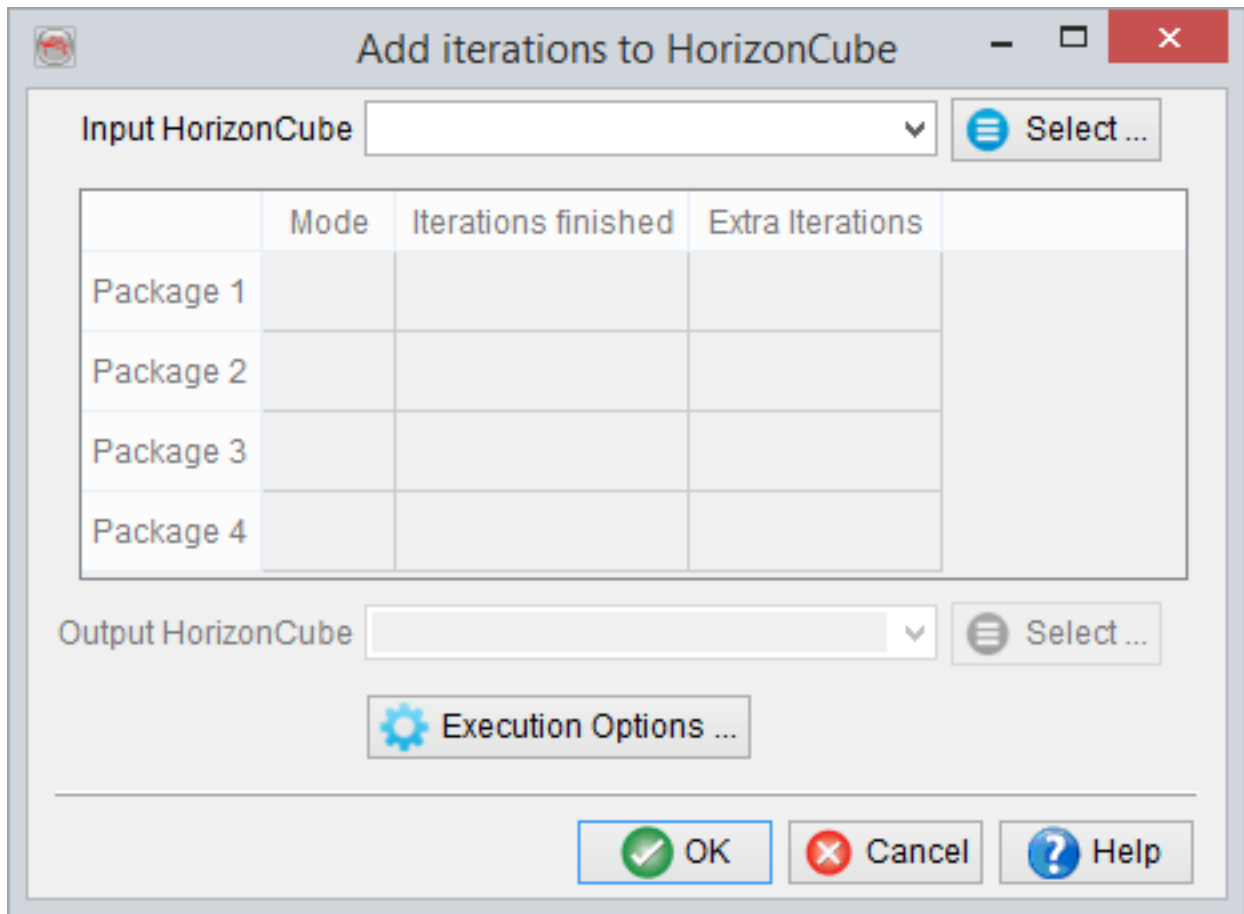


5.2.3.1 Add More Iterations

If a 2D/3D HorizonCube is created and there are still large areas with unwanted holes or gaps, more iterations can be inserted in a package to fill these spaces. There are four modes supported to fill the gaps:

1. **Data Driven:** This option fills the gaps using the SteeringCube i.e. a data driven HorizonCube with the same parameters as that were defined in the previous iteration.
2. **Proportional:** A model driven option to fill the holes proportional to upper and lower events of existing HorizonCube.
3. **Parallel to upper:** A model driven option to fill the holes using the geometry of the upper event of a HorizonCube..
4. **Parallel to lower:** Another model driven approach to fill the holes parallel to the geometry of the lower event of a HorizonCube.

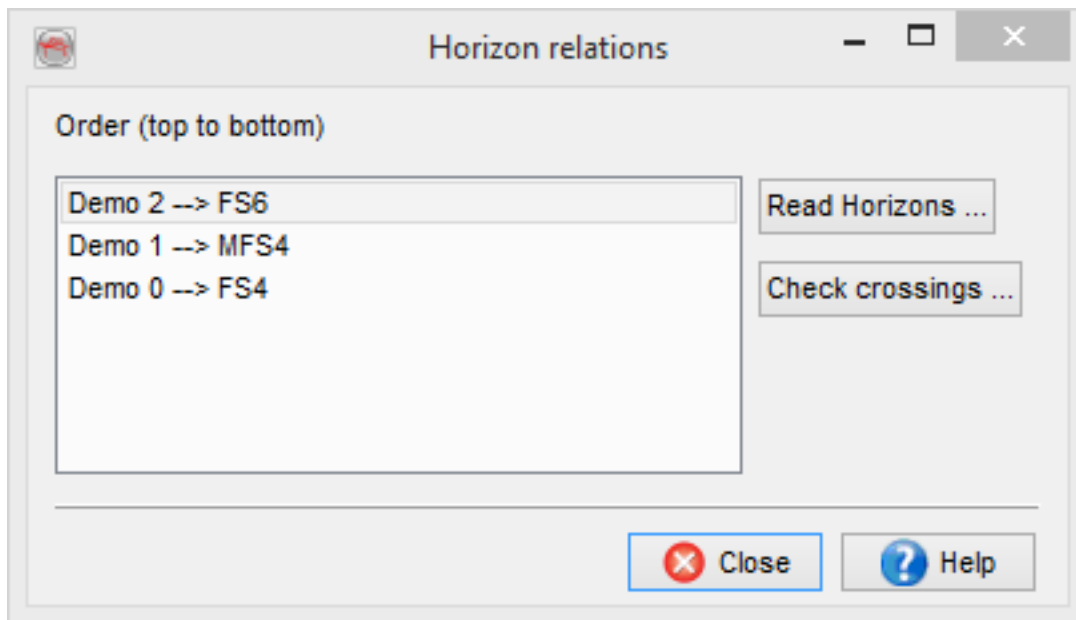
In the 'Input HorizonCube' field, the stored HorizonCube is selected whereafter the corresponding previously finished iterations are displayed in the table. The extra iterations in the next column are set to fill the gaps in a HorizonCube. The results of this processing can be stored into a new HorizonCube.



User interface of 3D HorizonCube to add more iterations.

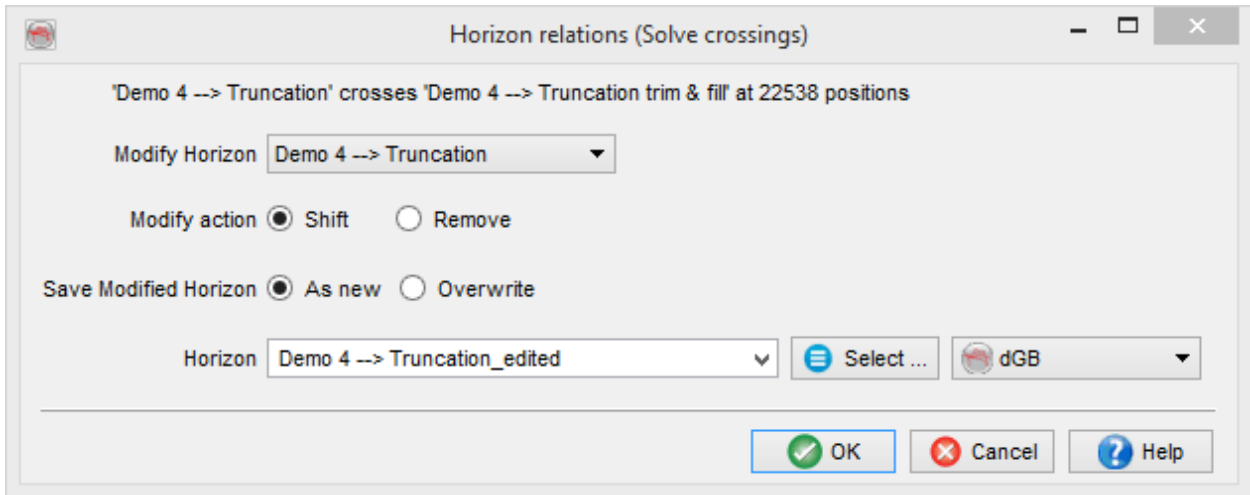
5.2.3.2 Check Crossings Between Horizons

It is used to resolve conflicts between horizons crossing each other. Press the Go button for the Data preparation step to launch the Horizon relation window. The later window is used to select all horizons that need checking (*Read Horizons ...*). The horizons are sorted automatically from top to bottom. The *Check crossings... button is used to check the crossings between the listed horizons automatically and resolve them.*



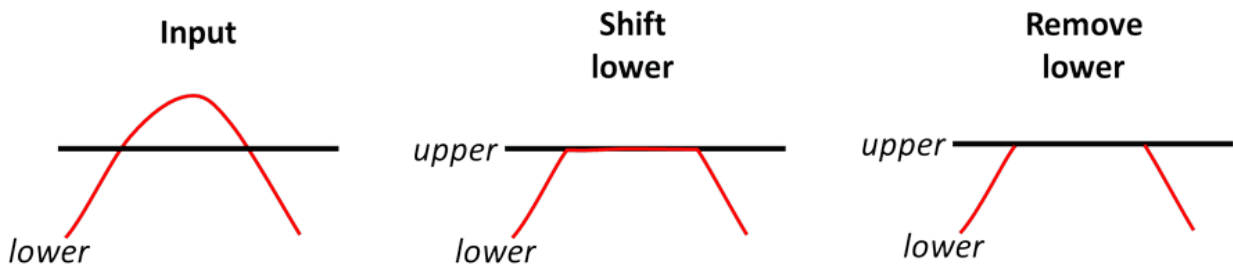
Solving crossing conflicts

To solve crossing conflicts select the horizon that will be modified. The software will check the number of positions where a conflict exists and modify the horizon by removing the conflict points or by changing the values to be equal to the overlying/underlying horizon.

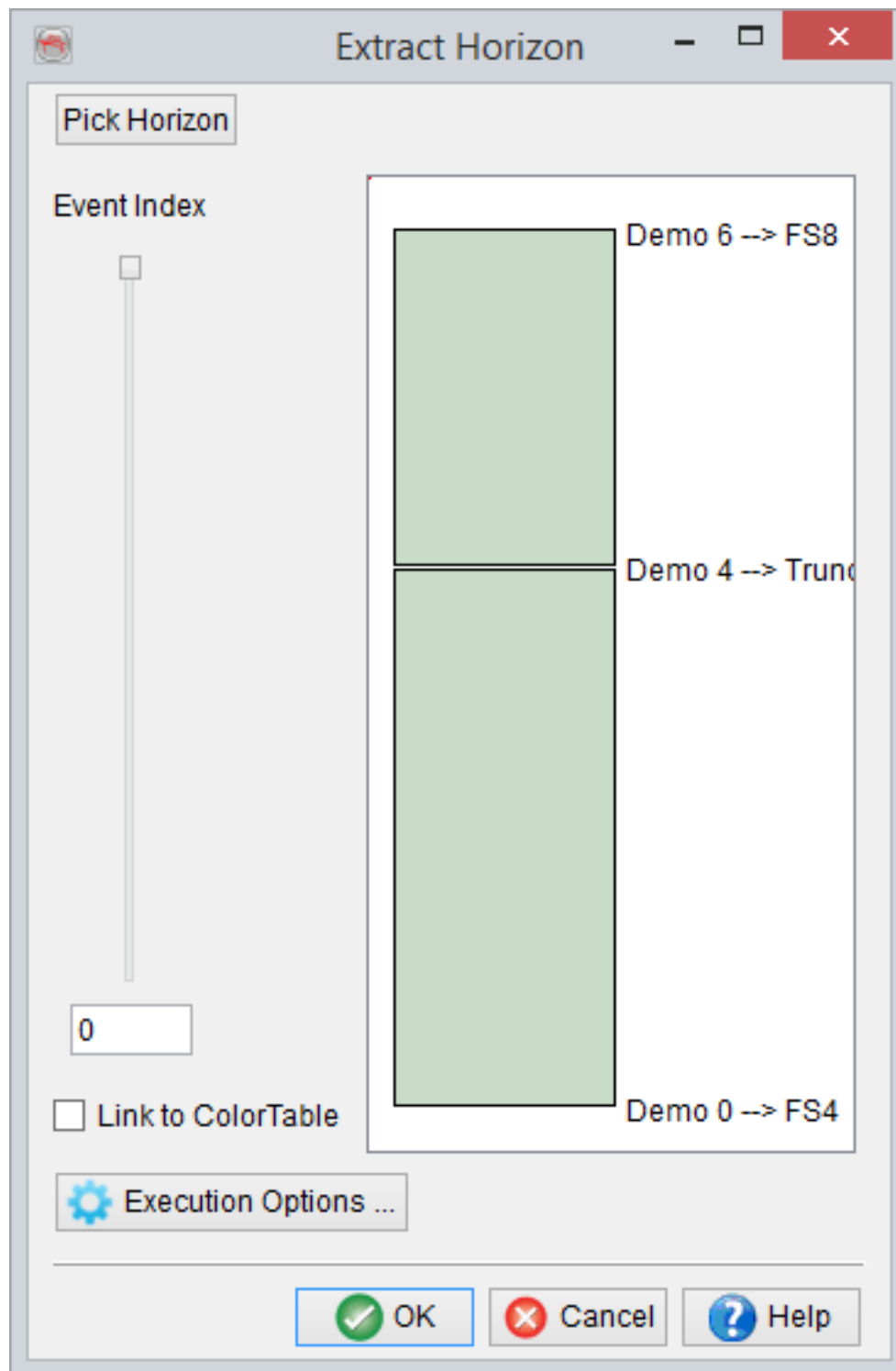


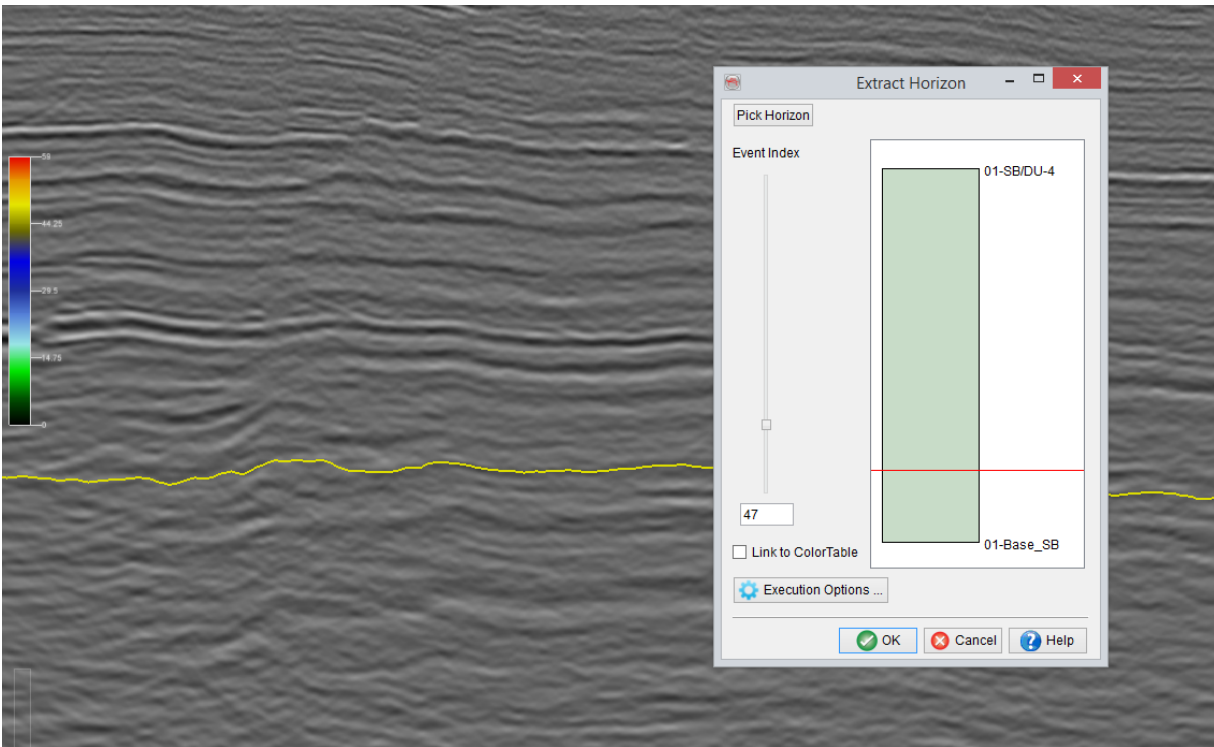
💡 *To honor the requirement that horizons cannot coincide, the actual values are not exactly equal, but they are within one sample position accuracy.*

The figure below sketches what will happen to the lower (red) horizon if you select shift or remove. The software verifies that removing and shifting operations are executed properly and the correct HorizonCube calculation results are reached.



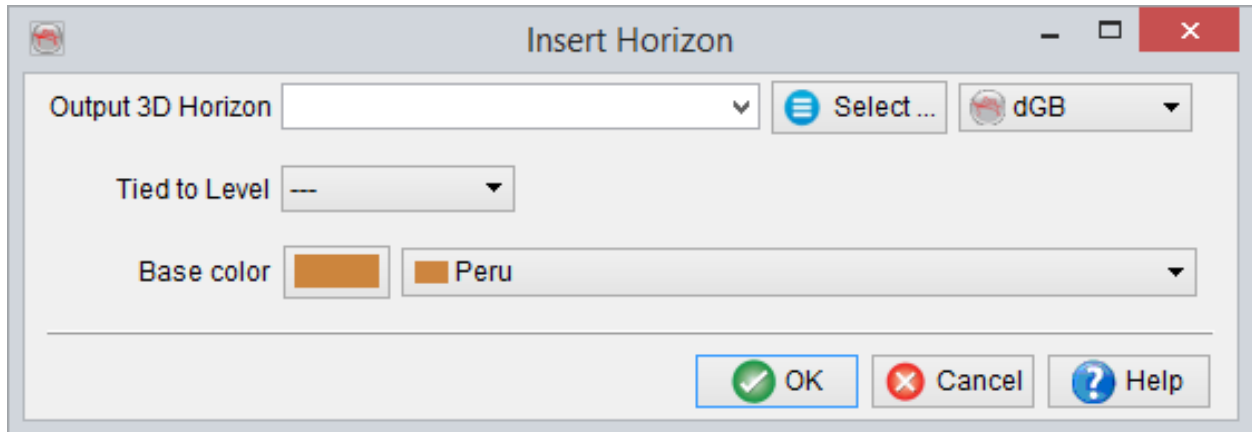
5.2.3.3 Extract Horizons





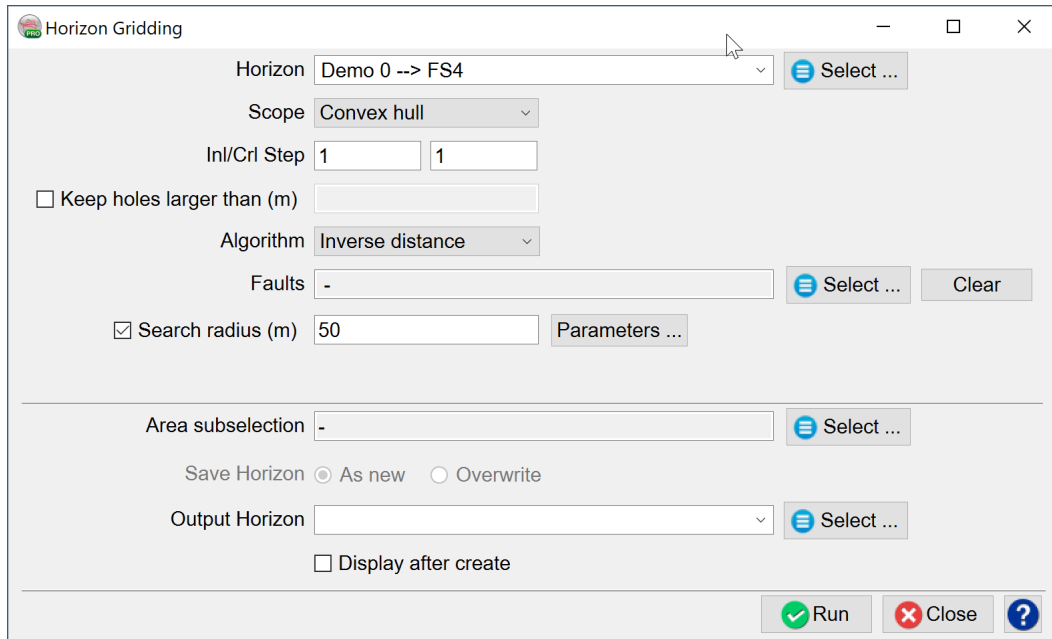
This tool is used to extract seismic horizons from the [selected HorizonCube](#). The slider is first positioned to a point where the user is interested in extracting a full 3D horizon. The relative position of the slider is always presented as a red-line in the graphical display of the packages in a HorizonCube. After setting the slider position, the Pick Horizon button is pressed to extract a horizon at the selected position.

The output name, color (and optional stratigraphic level) for the new horizon is specified in the pop-up window. This process can be repeated for any number of horizons. Clicking on 'Proceed' in the main window extracts and saves all picked horizons in a batch process.



Pick Horizon: Creates an output horizon at a defined slider position.

5.2.3.4 Grid A Horizon

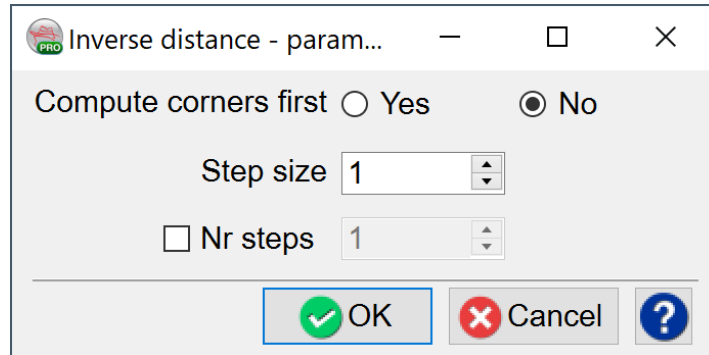


This utility is used to grid/interpolate a horizon having gaps/holes or to filter (average/median) a horizon grid. There are several gridding algorithms supported in OpendText.

Gridding Parameters:

- **Geometry:** There are different types of geometries that are used to do interpolation. The Full survey is used to interpolate (in-/out-wards) the Horizon-Z values within the entire survey box. The Bounding Box defines the rectangle fitting the horizon geometry, which is generally smaller than the survey box. The Convex hull type of area fitting also restricts the gridding geometry within the horizon boundaries. To grid the gaps or holes in a horizons, the Only holes type of gridding geometry is used.

- **Inl/Crl step:** The default steps correspond to the sampling rate of the input horizon. The step can be decreased up to the survey sampling rate to get a higher resolution horizon.
- **Algorithm(s):**



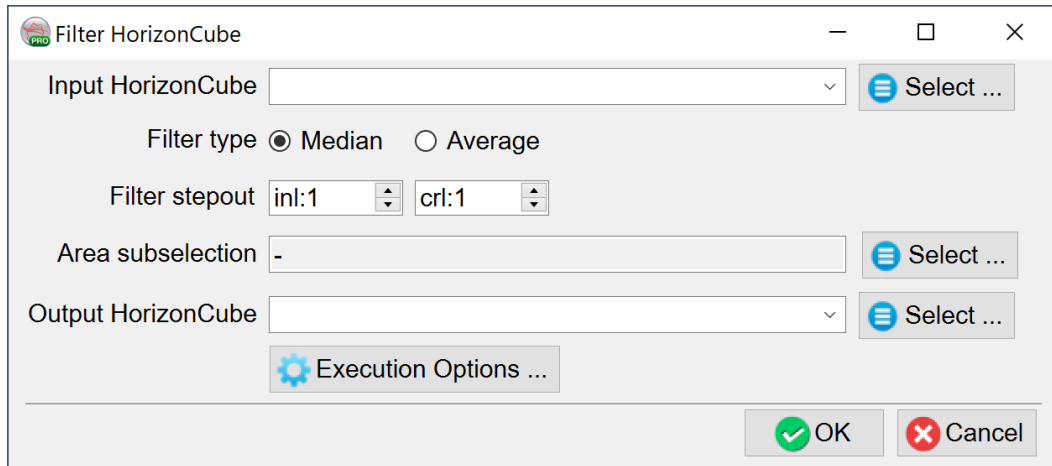
- *Inverse distance* algorithm uses an inverse distance method of interpolation. Inverse distance requires the search radius with optional parameters (step-size and number of steps). The step size of '1' means that one bin would be used in all directions to interpolate the horizon Z-values. Whereas the number of steps define the number of concentric circles for inverse distance computation. For these steps, the grid computation can be set to the corner points for the defined radius or not (default option).
- *Triangulation* is a fast gridding algorithm, which uses triangulation method of interpolation. The interpolation can also be defined by providing an optional maximum distance (radius) by setting interpolate option checked.
- *Extension* uses a simple linear interpolation algorithm to extend the horizon Z-values outward using the number of steps (bins), which need to be

defined in the following parameter field (Number of steps).

- *Continuous Curvature (GMT)* is a continuous curvature algorithm of interpolation, which is a part of the GMT Plugin of OpendTect. Please check the [GMT website](#) for further details. This algorithm only requires the tension parameter (ranges from 0-1), which controls the smoothing. The tension 0 gives minimum curvature type of surface interpolation, while the tension of 1 gives a harmonic surface.
- *Nearest Neighbour (GMT)* is also another interpolation algorithm coming from the GMT Plugin of OpendTect. This algorithm requires the search radius to be defined. It is mostly useful for a regularly spaced grid data. Please check the [GMT website](#) for further details.

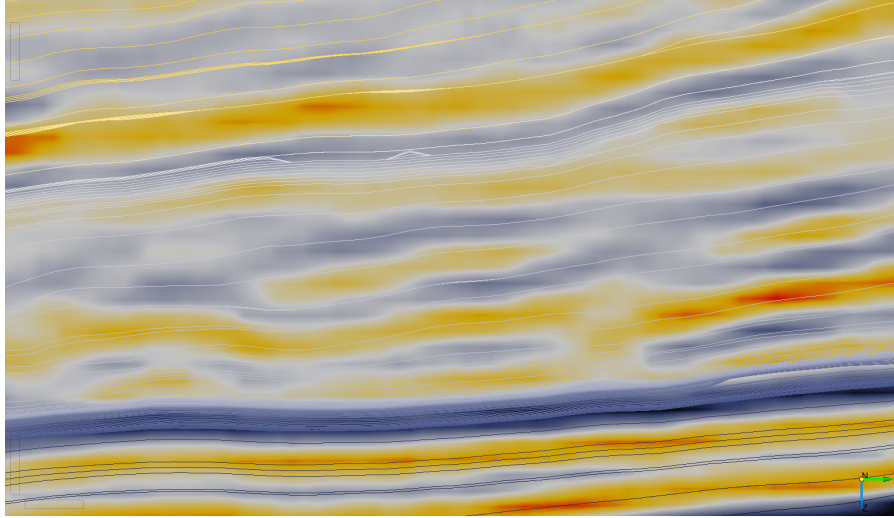
© dGB Beheer B.V. 2002 - 2021

5.2.3.5 Filter A HorizonCube

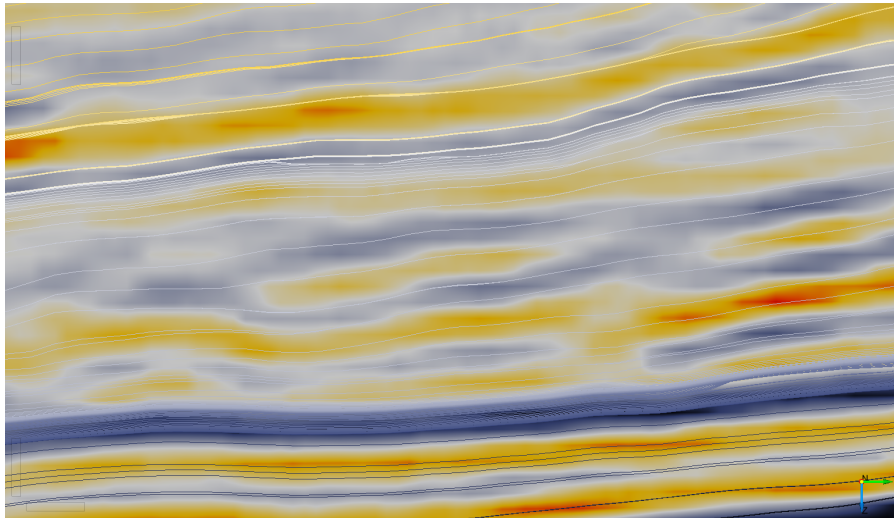


This option can be very handy if a user finds spikes in his/her HorizonCube. Two filter types are supported (median & average), each with a stepout in both the inline and crossline directions.

Before filtering the HorizonCube:

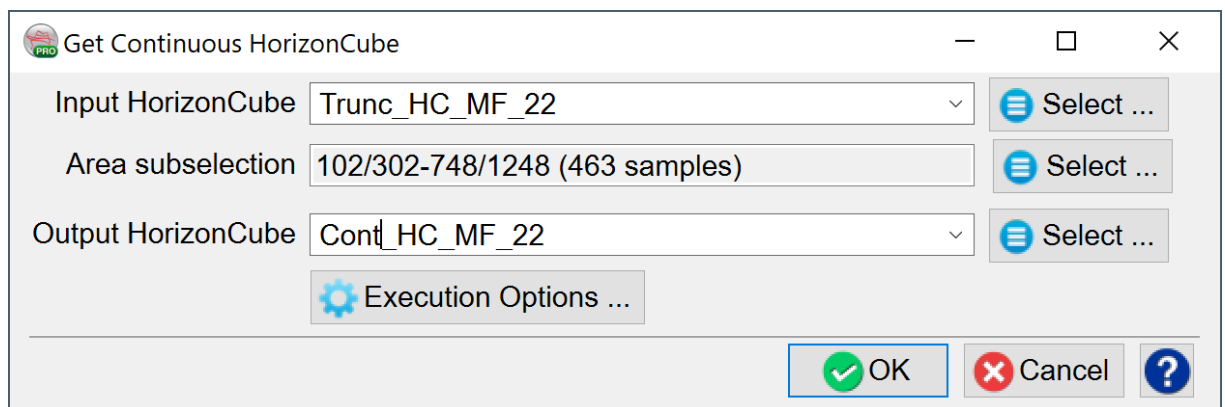


After applying median filtering, see that the spike is nicely smoothed:



5.2.3.6 Get Continuous HorizonCube

An existing [truncated HorizonCube](#) can be converted to a HorizonCube with [Continuous Events](#) using this tool. In the following window, the data selection is very simple. Select the input HorizonCube (truncated) in the input field and type the output HorizonCube name in the output field. Optionally, the area sub-selection can also be made to do the conversion within a sub-volume only.

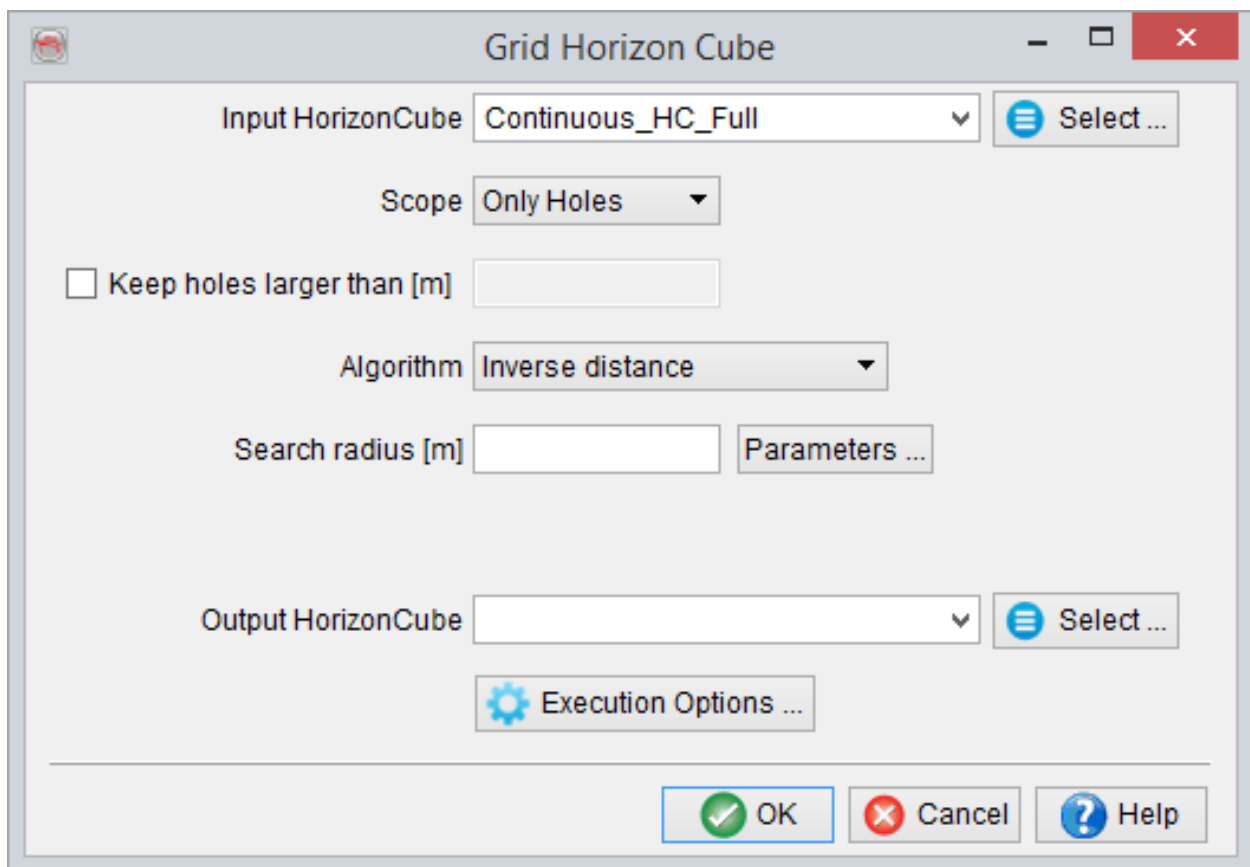


5.2.3.7 Grid HorizonCubes

After processing the HorizonCube, if the events have many holes, one can grid the output result.

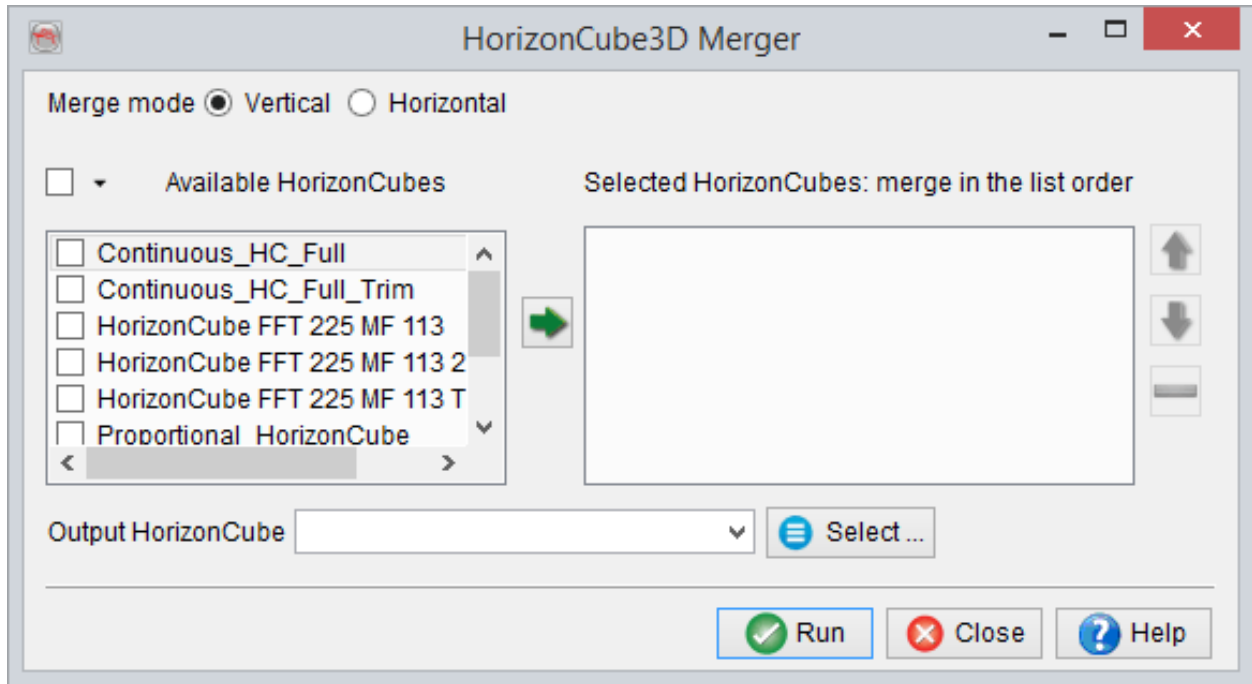
This is done through the HorizonCube control center.

Different gridding algorithms can be used: Inverse distance, Triangulation, Extension, GMT, and the Dip-Steered gridding as shown below.



5.2.3.8 Merge HorizonCubes

The merge tool allows for selection of multiple HorizonCubes to be sorted into a list for merging:



The process may be performed vertically or horizontally. The named output HorizonCube will be created and be available for all future processing.


5.2.3.9 Modify / Recalculate HorizonCube

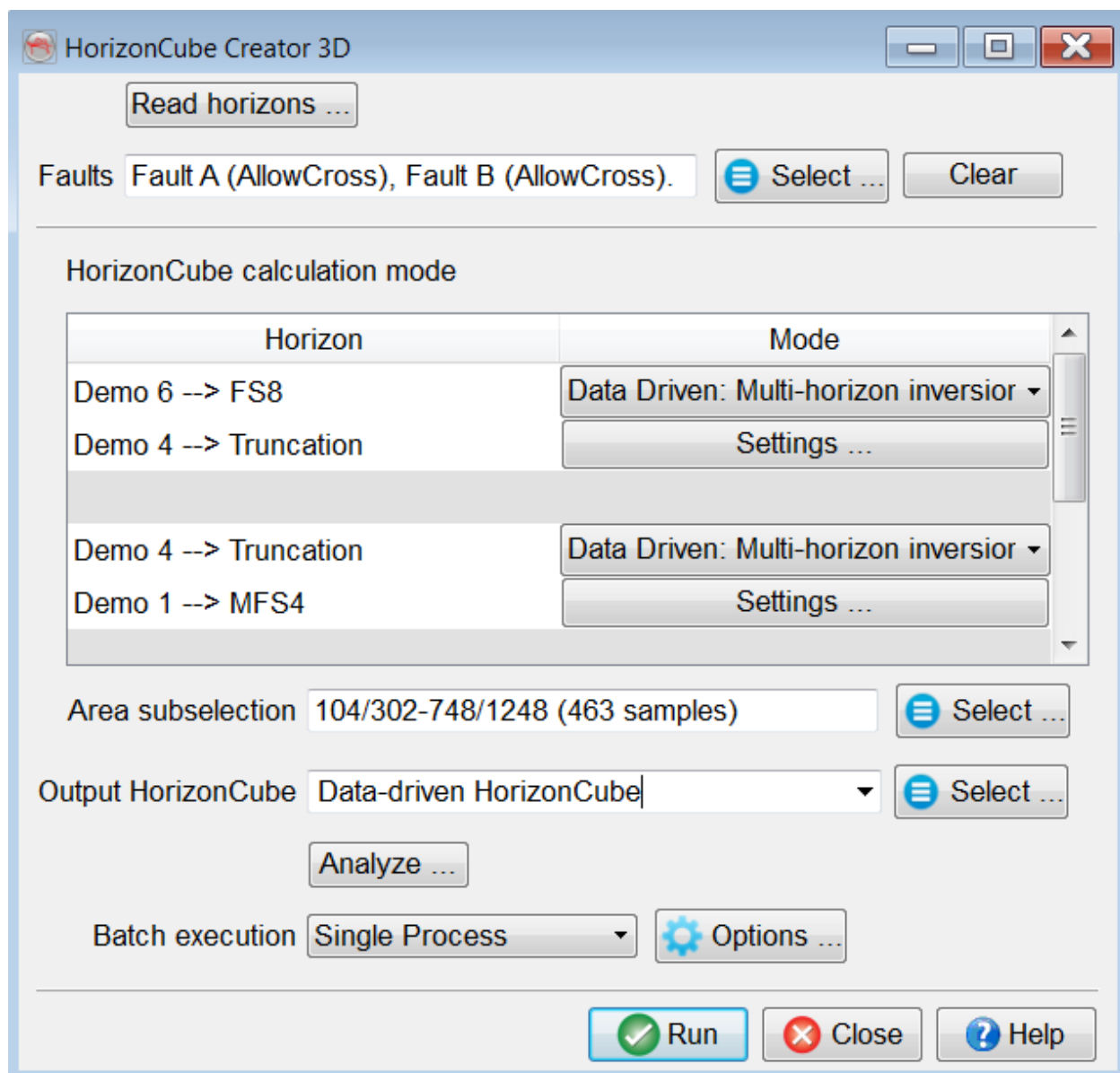
This tool is used to modify the HorizonCube (either 2D or 3D) with a new 'mode' or alternatively by using another SteeringCube. This tool is valuable in complex geologic intervals or when dealing with very noisy seismic data. In such cases changing parameters/settings for a package might be beneficial. Specifically, the calculation mode might be changed from 'data driven' to 'model driven' or vice versa. SteeringCubes can also be set individually for each package. Thus, this tool includes two options for any package of a HorizonCube: *Keep* or *Re-calculate*.

In the following window, the input 3D HorizonCube is provided. The table for HorizonCube calculation mode is populated according to the previous settings for the selected HorizonCube. The *Read Horizons* button can also be used to add more horizons (packages) to the selected HorizonCube. Note: If *Read Horizons* is clicked, even wanted existing horizons in the HorizonCube have to be selected.

In the table, set calculation *Mode* and set the *Action* to re-compute. The corresponding settings can also be changed. Additionally, more interpreted faults can also be added to the existing HorizonCube by click on the *Select* button for the *Faults* field. The *Analyze* button is used to check the inconsistencies and problems that might be encountered during the calculation of the HorizonCube.

Clicking the *Go* button starts the HorizonCube modify/re-calculate batch program for the selected HorizonCube with the new settings.

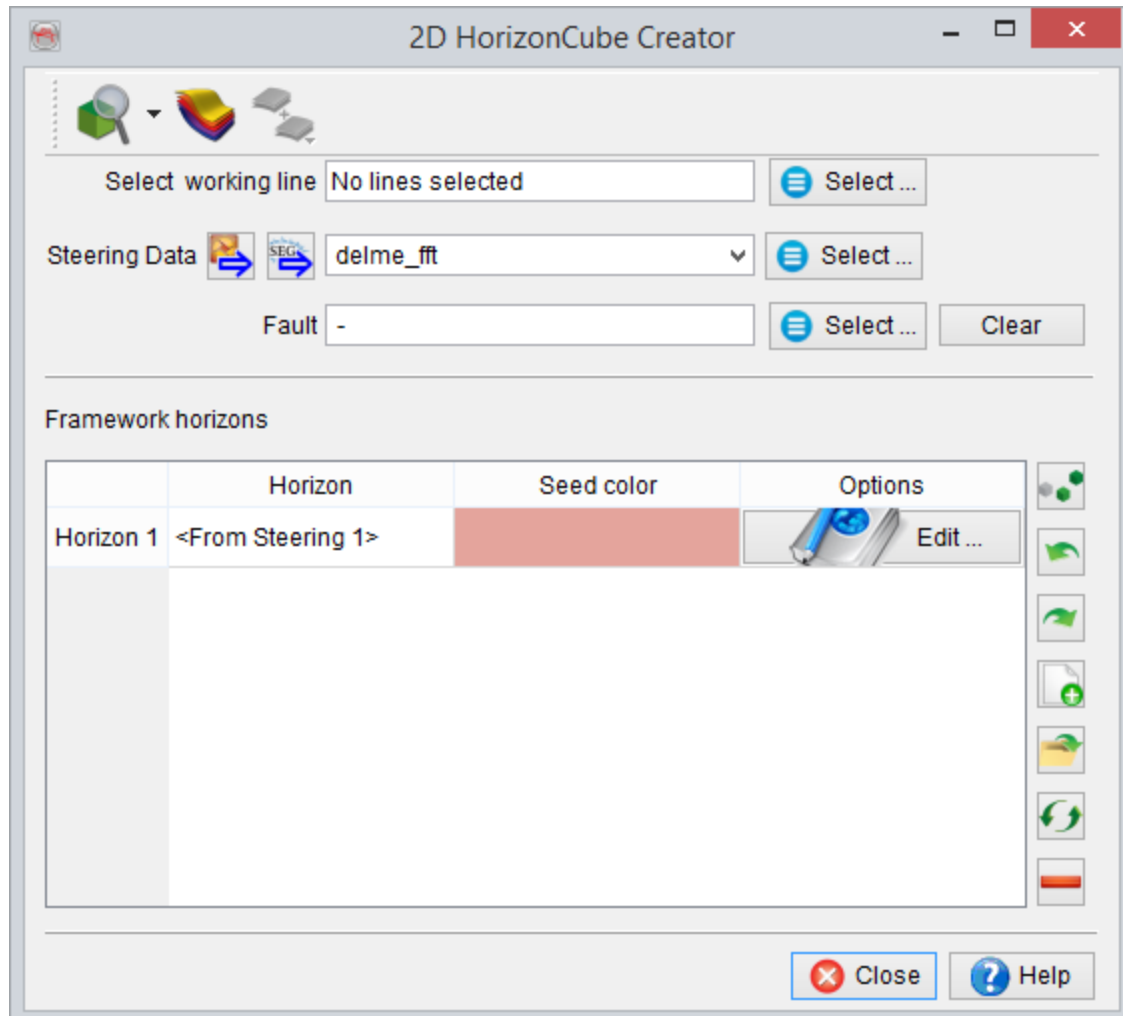
 *The Show Option check-box is an optional tool for the batch program configuration. By default (unchecked) the batch program runs in a pop-up window. Further, a batch program can be run in a log file (without showing the pop-up window). Remote connections to other computers for Distributed Computing (processing) is also accessed here.*



Modify a previously calculated HorizonCube with another Mode/SteeringCube with different settings.

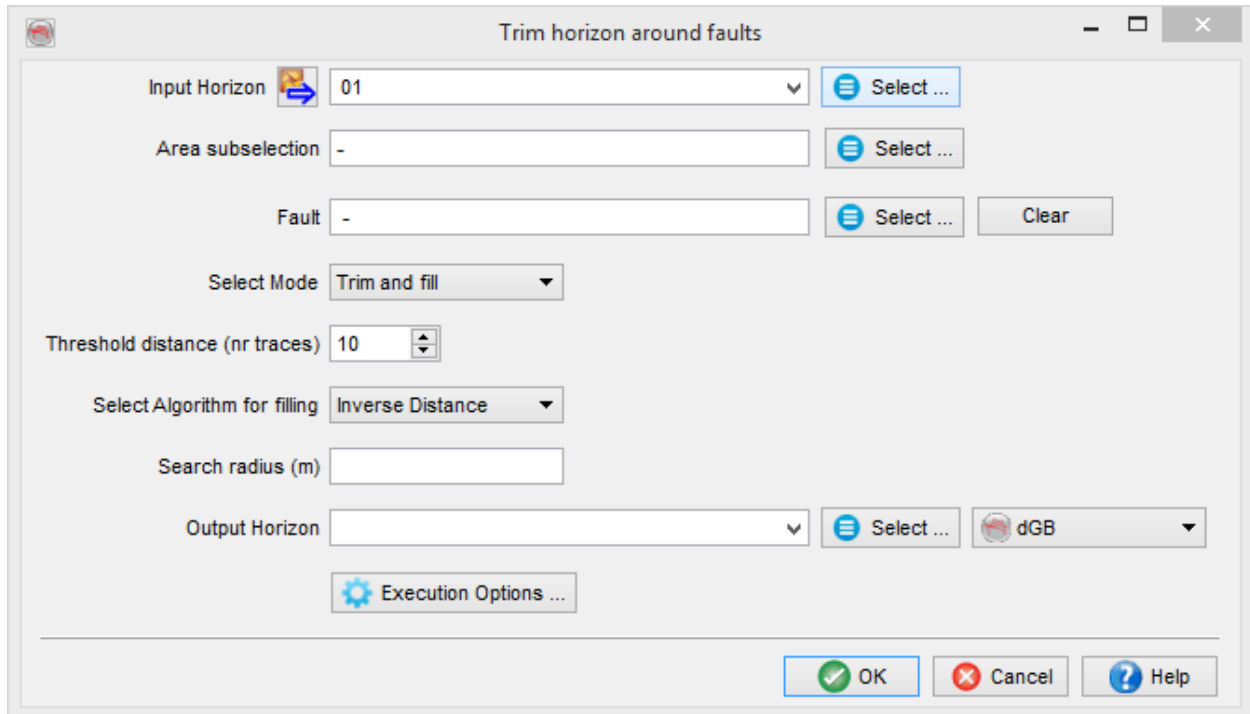
Add / Recalculate line

The workflow for modifying/re-calculating 2D HorizonCubes is in many ways similar to the 3D workflow. Set 'Input HorizonCube', choose which lines to add/modify ('Select lines to add/modify'). Packages and faults can then be added, modified or kept as-is.



5.2.3.10 Trim Against Faults

The trimming feature in OpendTect improves the fault/horizon relation.



There are different modes to trim a horizon at given faults location.

Trim and Track

It is an option to remove an area defined by the trace threshold away from a fault plane. The removed part is then tracked back to the fault plane using a selected SteeringCube. Please note that on an average seismic data close to a fault plane, seismic data could be very noisy. Thus you can expect that a computed SteeringCube could also be very noisy. Therefore, it is suggested to use a background (heavily filtered) SteeringCube for this purpose.

Trim only

This will only remove the parts of a given horizon within a defined trace radius close to a fault plane.

Re-track only

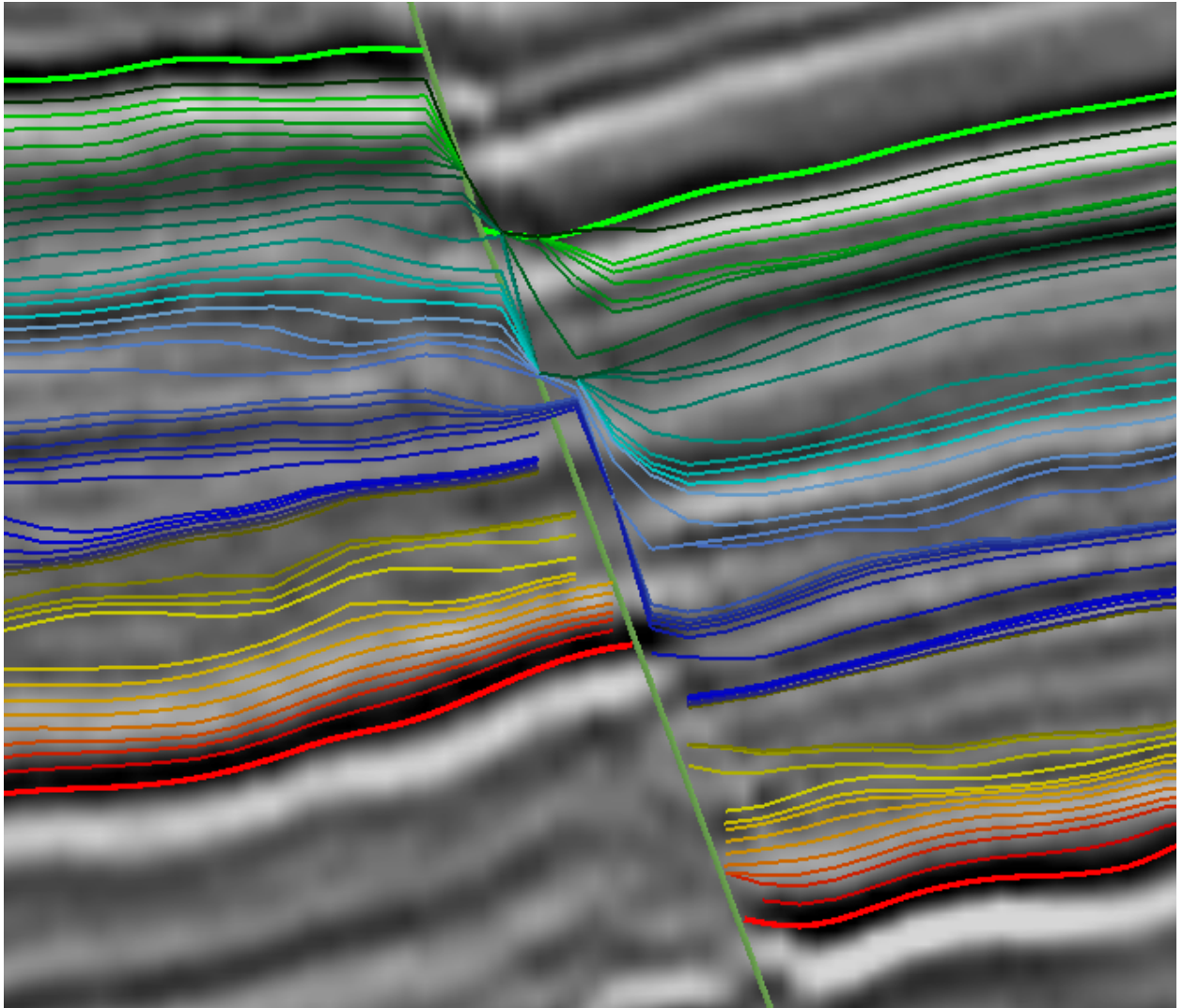
This option will track a horizon close to a fault.

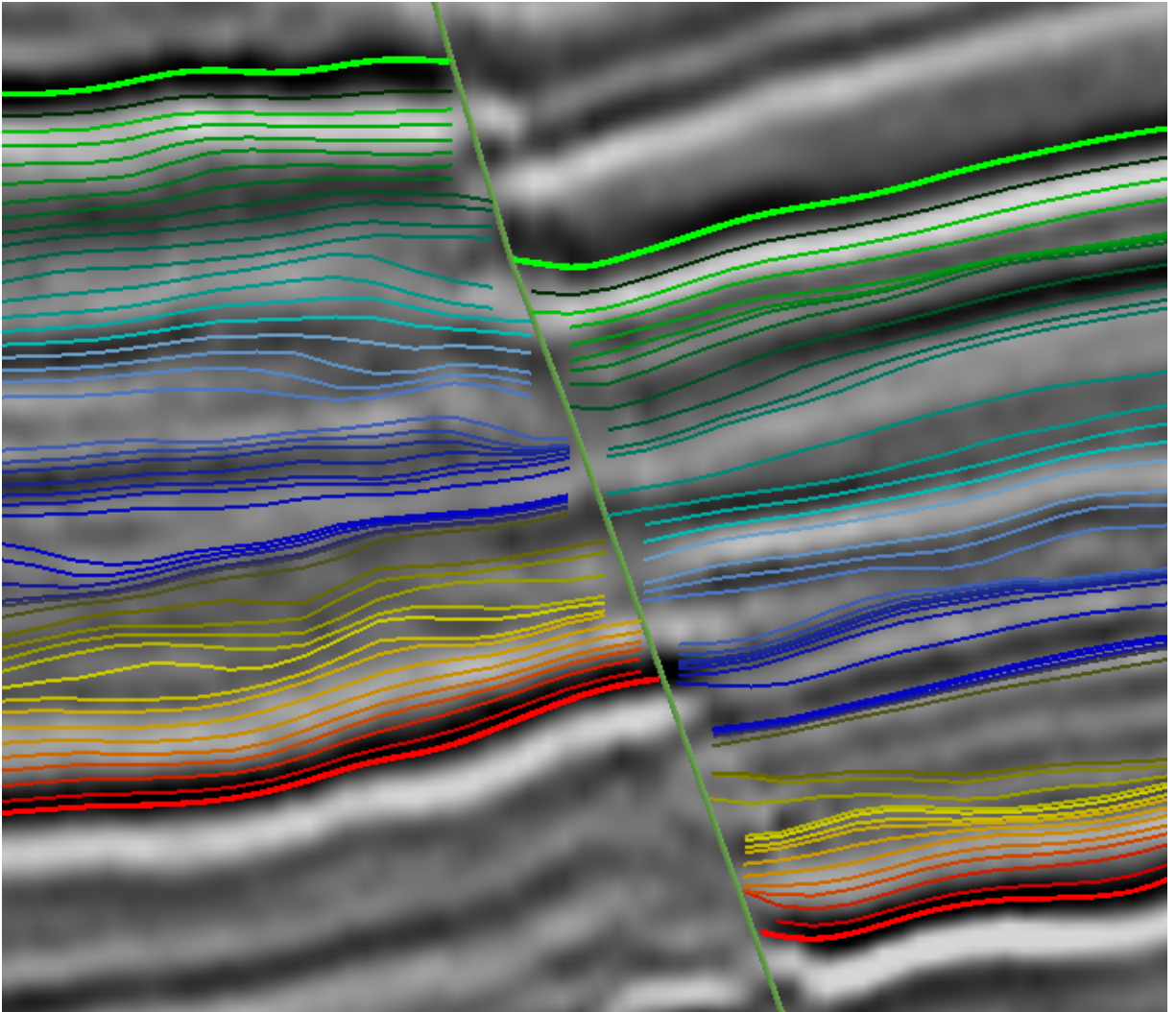
The threshold should be chosen according to 'neatness' of the horizon around the fault. If a unnecessarily large threshold is chosen, a large part of the horizon will be deleted and re-tracked, thus making it prone to errors. Ideally, a threshold of 10 is sufficient (default value), whereas for an 'indisciplined' horizon a threshold of 30 or above may be needed. The same horizon may behave differently around different faults. Different thresholds for the same horizon but different faults or fault-sets may be chosen.

In this case the trimming should be done in steps:

- Original horizons ----> using 1st set of faults and a small threshold --> Horizon 1
- Horizon 1 ----> using 2nd set of faults and slightly larger threshold --> Horizon 2
- Horizon 2 ----> using 3rd set of faults and a large threshold --> Final output Horizon

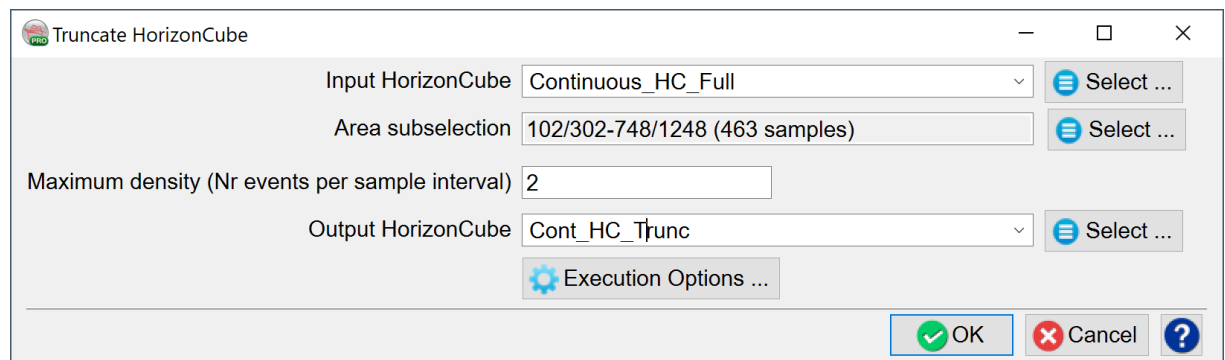
An example of before- and after trimming is shown below:





5.2.3.11 Truncate HorizonCube

An existing [continuous HorizonCube](#) often contains very closely spaced events. The HorizonCube can later be filtered by defining a maximum density (threshold) as a number of closely spaced events. In the following window the data selection is very simple. Select the input HorizonCube (continuous) in the input field and type the output HorizonCube name in the output field. The *Maximum density* defines a threshold for two horizons in the HorizonCube. The value '1' defines a threshold of one event per sample of a seismic data. Optionally, the area sub-selection can also be made to do the conversion within a sub-volume only.



To learn more about the benefit of applying a density filter to the continuous HorizonCube, please read the [Wheeler Transformation](#).

5.3 HorizonCube Attributes

HorizonCube attributes are new geometrical attributes introduced by dGB Earth Sciences. These are purely stratigraphic attribute that help in identification and characterization of stratigraphic traps. These attributes can be computed directly from HorizonCube and stored as volume (as the Attribute Set Window does) or stored as a separate element such as HorizonCube Data, which contains grid based attributes. The advantage of computing HorizonCube attributes as a volume is that one could easily render it in 3D as one cannot easily render 100s of grids in 3D scene. Further advantage of this approach is that one could also integrate the HorizonCube attributes with conventional seismic attributes by using logical expressions or neural networks to characterize seismic data, to predict hydrocarbon traps etc.

Attribute Set Window

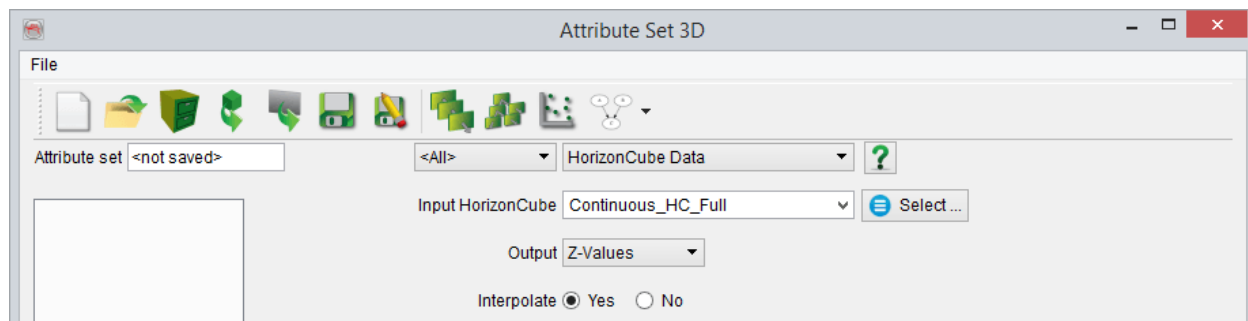
These attributes can be found in general OpendTect Attribute Set window, which can be launched via Analysis > Attributes > 3D/2D.

Using Attribute set window, one can define the following HorizonCube attributes:

HorizonCube Data

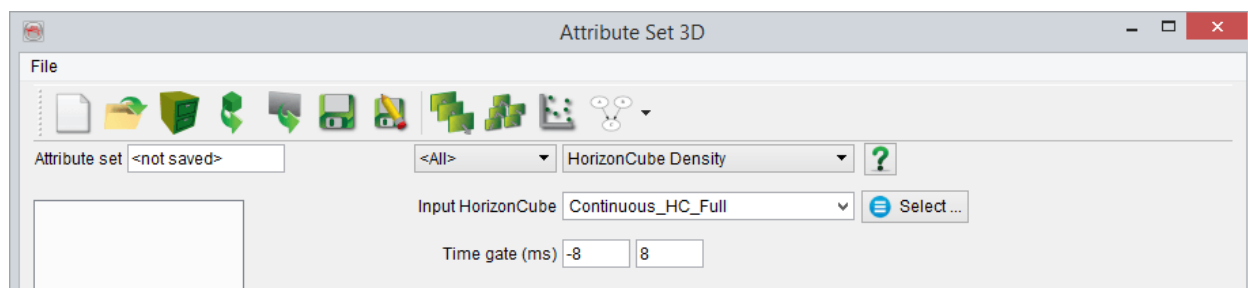
This attribute is used to convert the attributes that are stored under HorizonCube data into a volume. The input HorizonCube should be a 3D HorizonCube for this case as it is not supported for a 2D HorizonCube. The output field is populated if the selected 3D HorizonCube contains pre-computed [HorizonCube Data](#). By default, the *Interpolate* option is set to yes as there could be gaps that you might want to fill. If you want to keep the gaps empty, set this option to No.

It is equivalent to OpendTect's basic attribute i.e. horizon. To read further, please follow the section below on *Calculate attributes along HorizonCube*.



HorizonCube Density

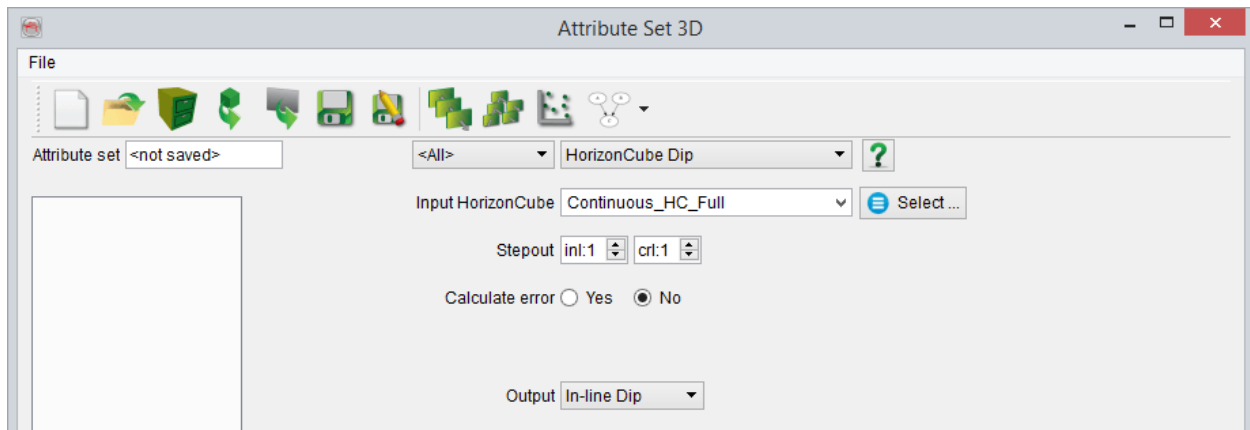
It is an 2D/3D attribute that defines the density of HorizonCube events within a defined time gate. It returns an *event count*, which defines total number of events within a time gate. The output values are number of events.



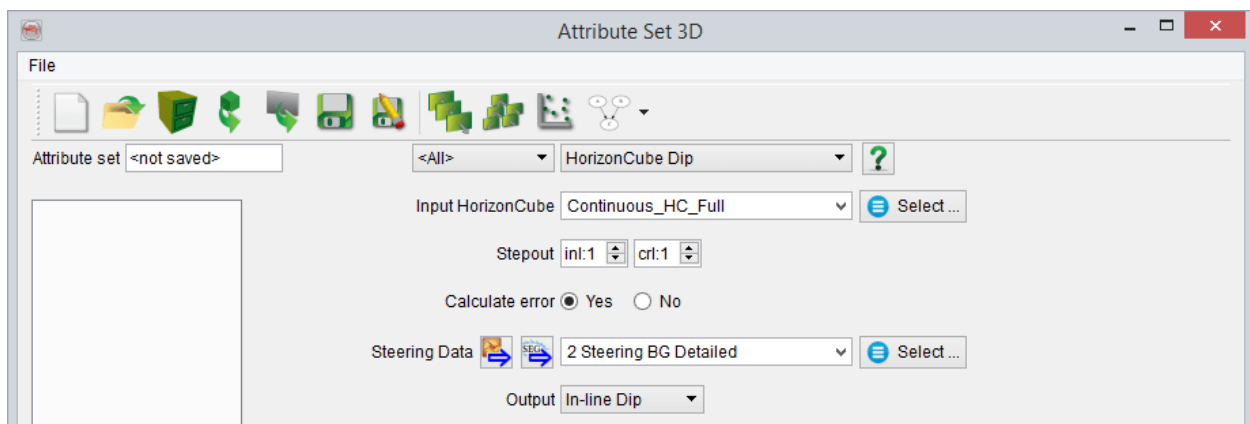
The attributes can be use to map or visualize pinchouts, unconformities, condensed intervals, etc. These objects will appear as high density values.

HorizonCube Dip

This attribute returns the dips of 2D/3D HorizonCube as a volume. The stepouts are the radii to compute the dips along each event. Larger stepouts define the regional dips along an event.



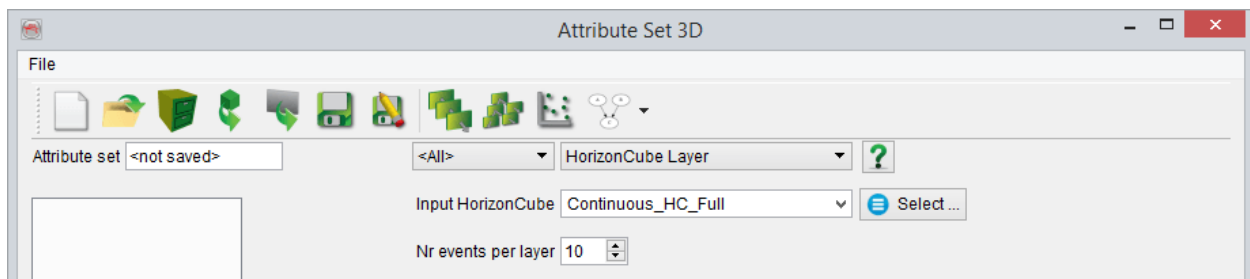
In order to QC the dips of HorizonCube vs. input SteeringCube, one may set the *Calculate error* field to Yes. This would allow you to select the parent SteeringCube to further evaluate the HorizonCube results. The errors are computed for each seismic sample by subtracting the dips of HorizonCube from the selected SteeringCube. The result is a dip difference. Lower dip errors mean that HorizonCube events do not deviate much from the selected SteeringCube and vice versa.



HorizonCube Layers

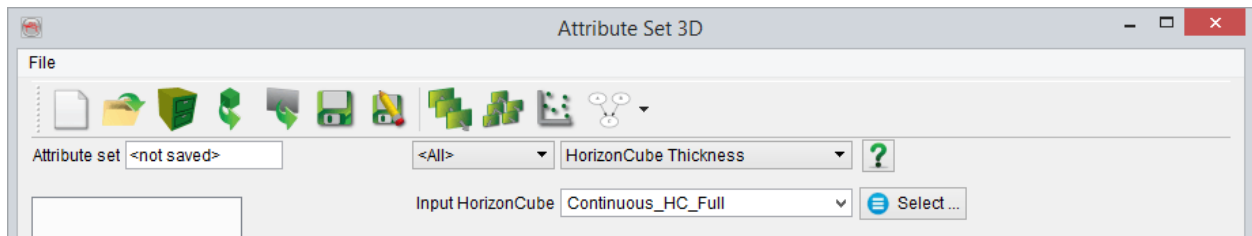
It is very convenient to visualize 100s of HorizonCube events in 3D scene. However, one could convert them into a volume and could easily render it in 3D. This attribute does such a job, which primarily governs to QC HorizonCube results.

The number of events per layer field allows to select how many events should be considered to defined a layer. The default value is set to 10, which means that each resultant layer contains at least 10 HorizonCube events. The output of this attribute is a range of values defining number of layers.

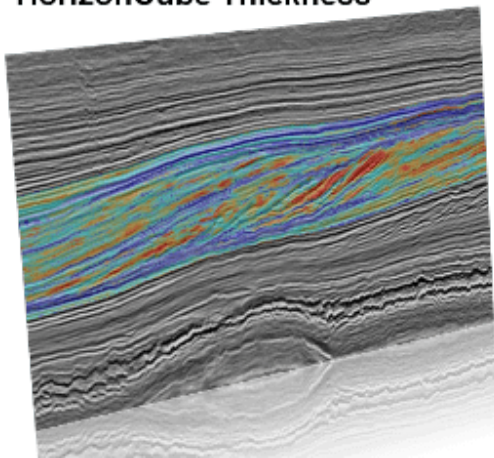


HorizonCube Thickness

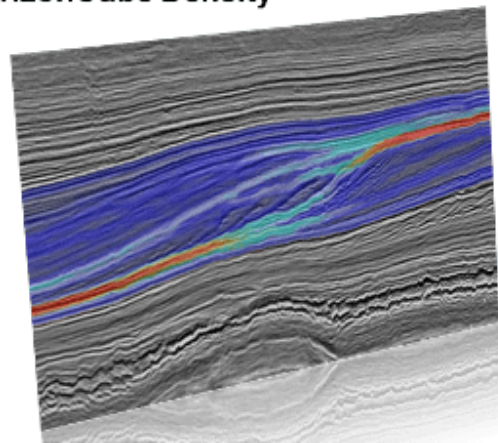
It is a vertical thickness (isochron) between two HorizonCube events. It is also a trace attribute along the HorizonCube events. Such an attribute can also be used to visualize pinchouts or unconformities (zones of near zero thickness).



HorizonCube Thickness



HorizonCube Density

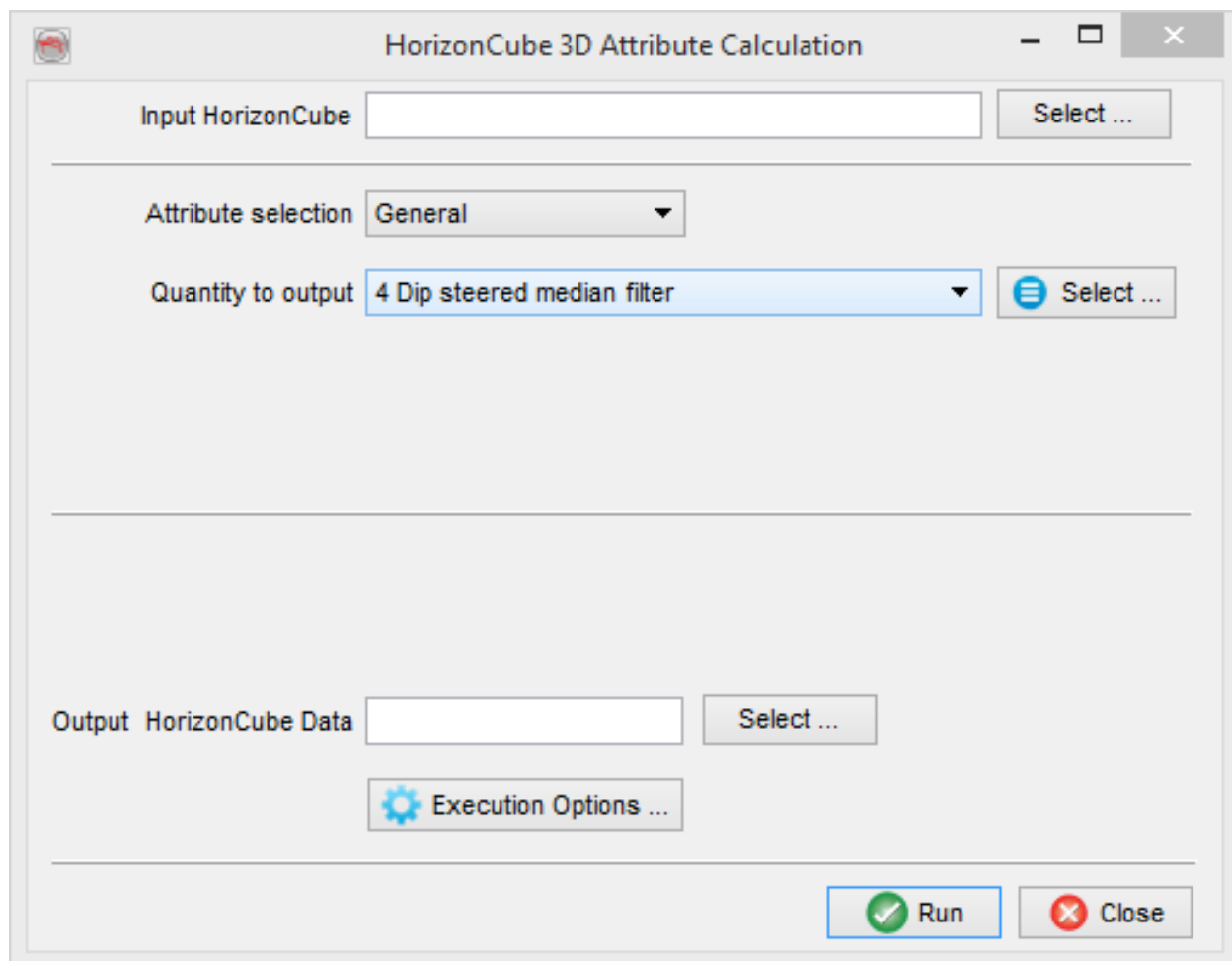


Example of HorizonCube attributes overlain on seismic data.

Calculate Attributes along 3D HorizonCube

These attributes are computed from the [HorizonCube Control Center](#). The output of all of these attributes is a set of grids, which are collectively saved as [HorizonCube Data](#). These attribute grids can be visualized using [3D Slider](#).

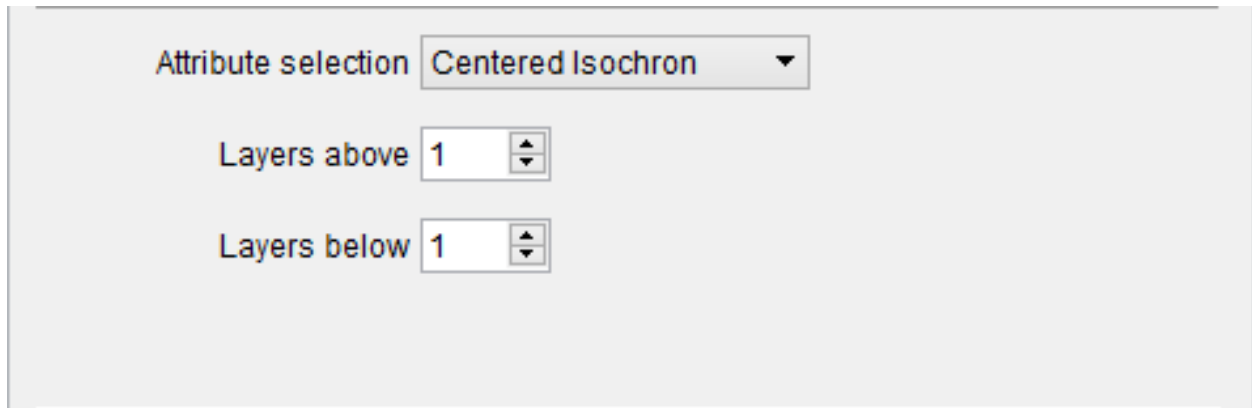
In the *Select HorizonCube* option, you can select the entire HorizonCube, or a package or an event.



General: This category is used to compute any existing attribute (stored or defined) along HorizonCube events. For instance, if you already have a stored AI or PHI volume and you would like to compute it along HorizonCube, you can use this option.

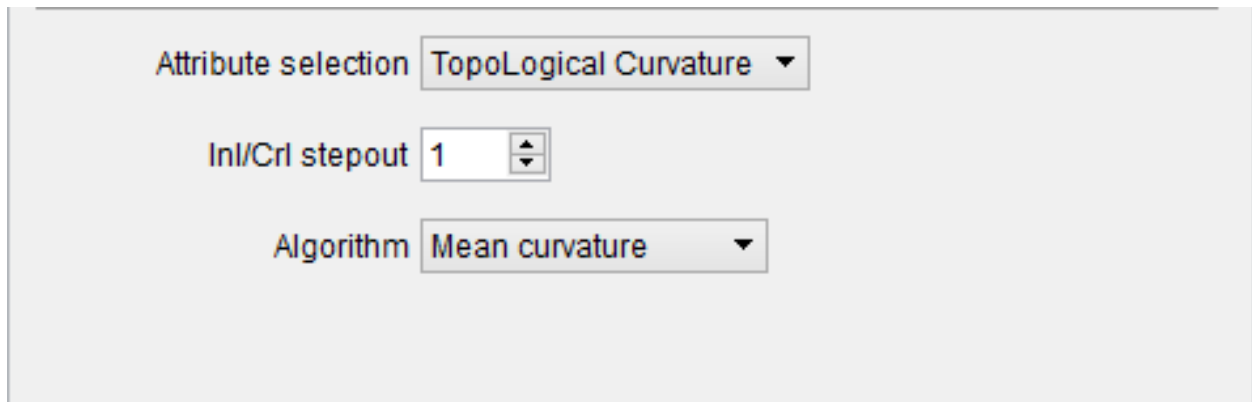
Centered Isopach: This attribute calculates the vertical thickness (TWT/depth) between two events such that they are centered by an event. In this manner, the thickness is computed along the centered event. Number of layers above/below define the relative number of events to compute the thickness. For instance, if the value for layers

above/below is set to 1, for each central event, the thickness is computed by subtracting the TWT/Depth of an underlying event from the overlying event and storing the result for the central one.



The screenshot shows a software interface with three controls. At the top, there is a dropdown menu labeled 'Attribute selection' with the value 'Centered Isochron' and a downward arrow. Below this, there are two spinners. The first is labeled 'Layers above' and has the value '1'. The second is labeled 'Layers below' and also has the value '1'. Both spinners have up and down arrows for adjustment.

Topographical Curvature: This attribute defines geometrical curvature of an event, attribute or HorizonCube Data. The stepouts are number of inlines/crosslines to be used to compute the curvature. These attributes are mostly equivalent to the [conventional curvature attributes](#) that are directly computed from the SteeringCube.



The screenshot shows a software interface with three controls. At the top, there is a dropdown menu labeled 'Attribute selection' with the value 'TopoLogical Curvature' and a downward arrow. Below this, there are two controls. The first is a spinner labeled 'Inl/Crl stepout' with the value '1'. The second is a dropdown menu labeled 'Algorithm' with the value 'Mean curvature' and a downward arrow.

References

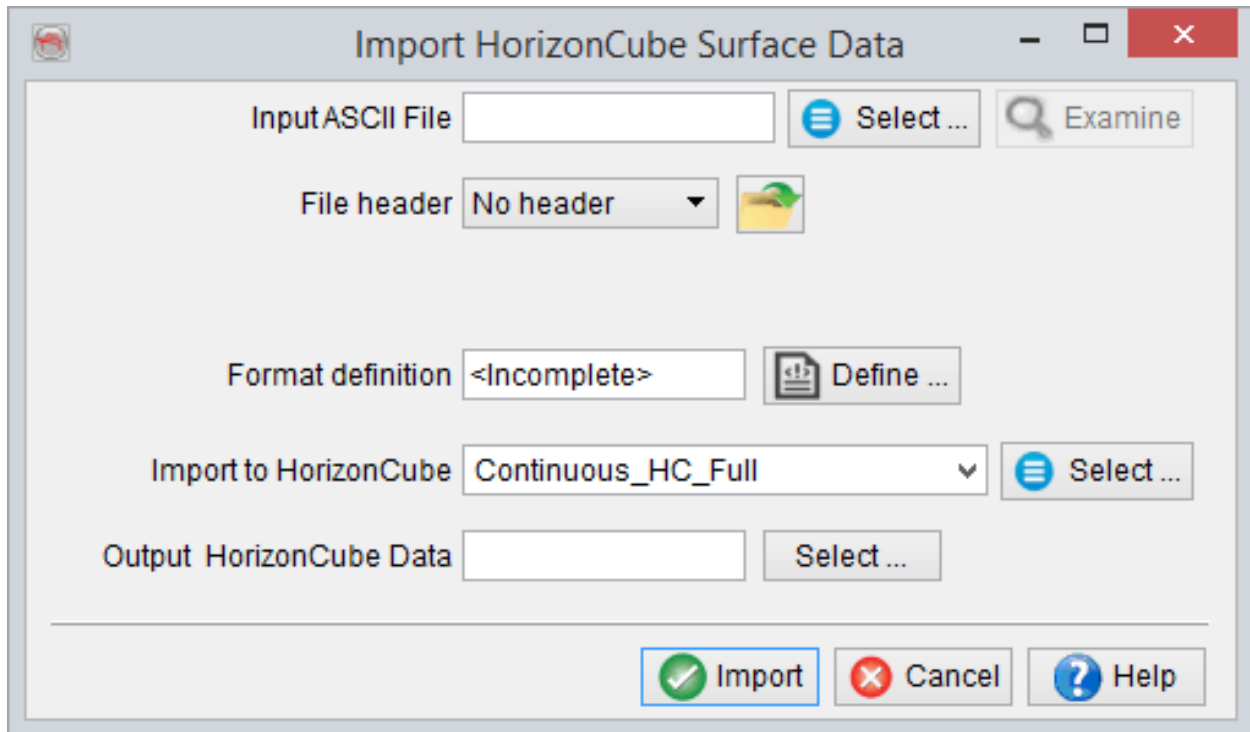
de Groot, P., Qayyum, F., On the Horizon, Oilfield Technology, p. 24-28, January 2013.

de Groot, P., Qayyum, F., Attributes play important role in seismic interpretation, Hart's E&P Magazine, p. 31-34, October 2012.

Weblink: [OpendTect Attributes Table](#)

5.4 Import HorizonCube Surface Data

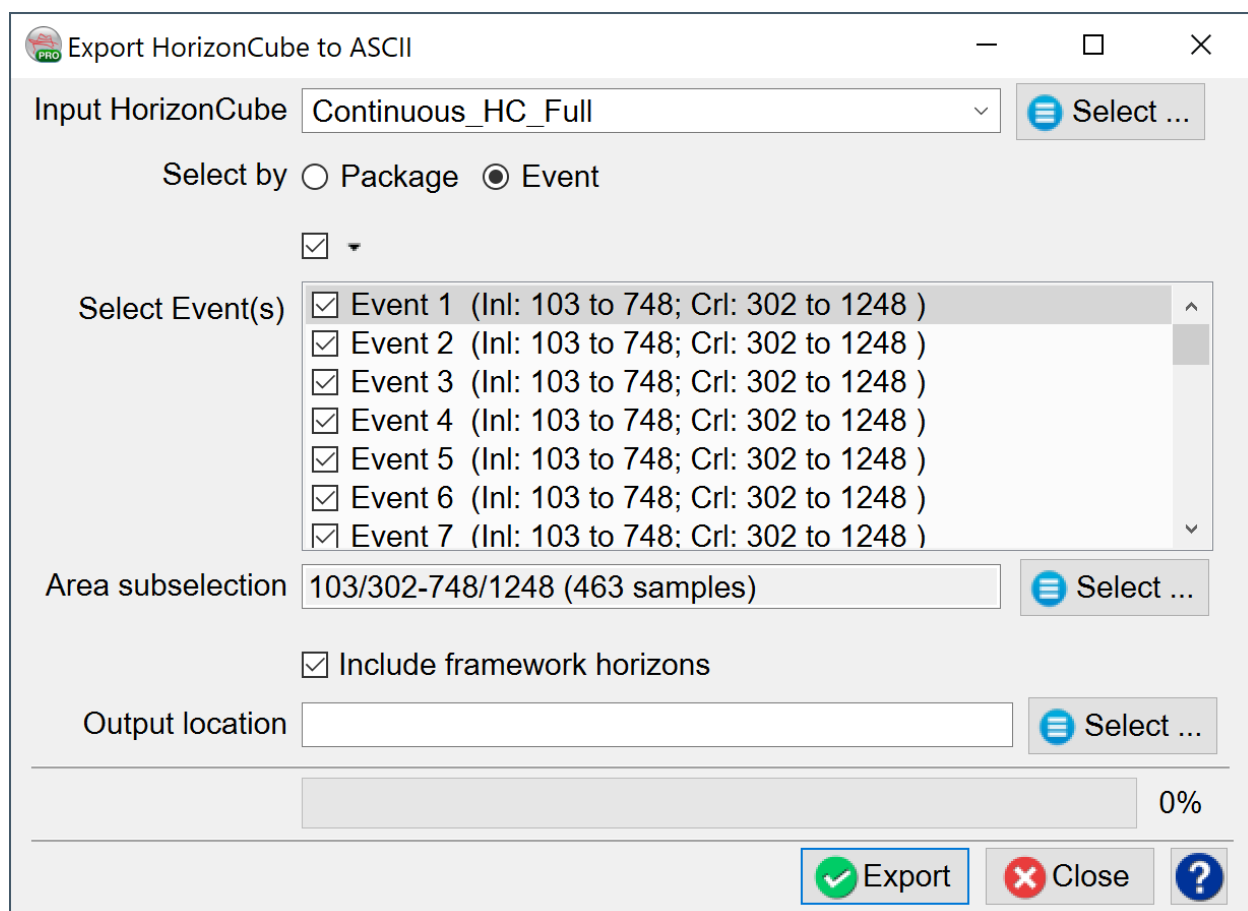
The HorizonCube surface data can be imported in OpendText as an ascii file. This is done via:



Some few specifications are needed such as: File header, the format definition. The surface data is then imported as HorizonCube.

5.5 Export HorizonCube

The 3D HorizonCube can be exported as a single/multiple horizon, into an ASCII file. This can be done via export sub menu (Survey > Export > HorizonCube 3D). In the 'Export HorizonCube to ASCII' window, select the HorizonCube and export either the package or the events.

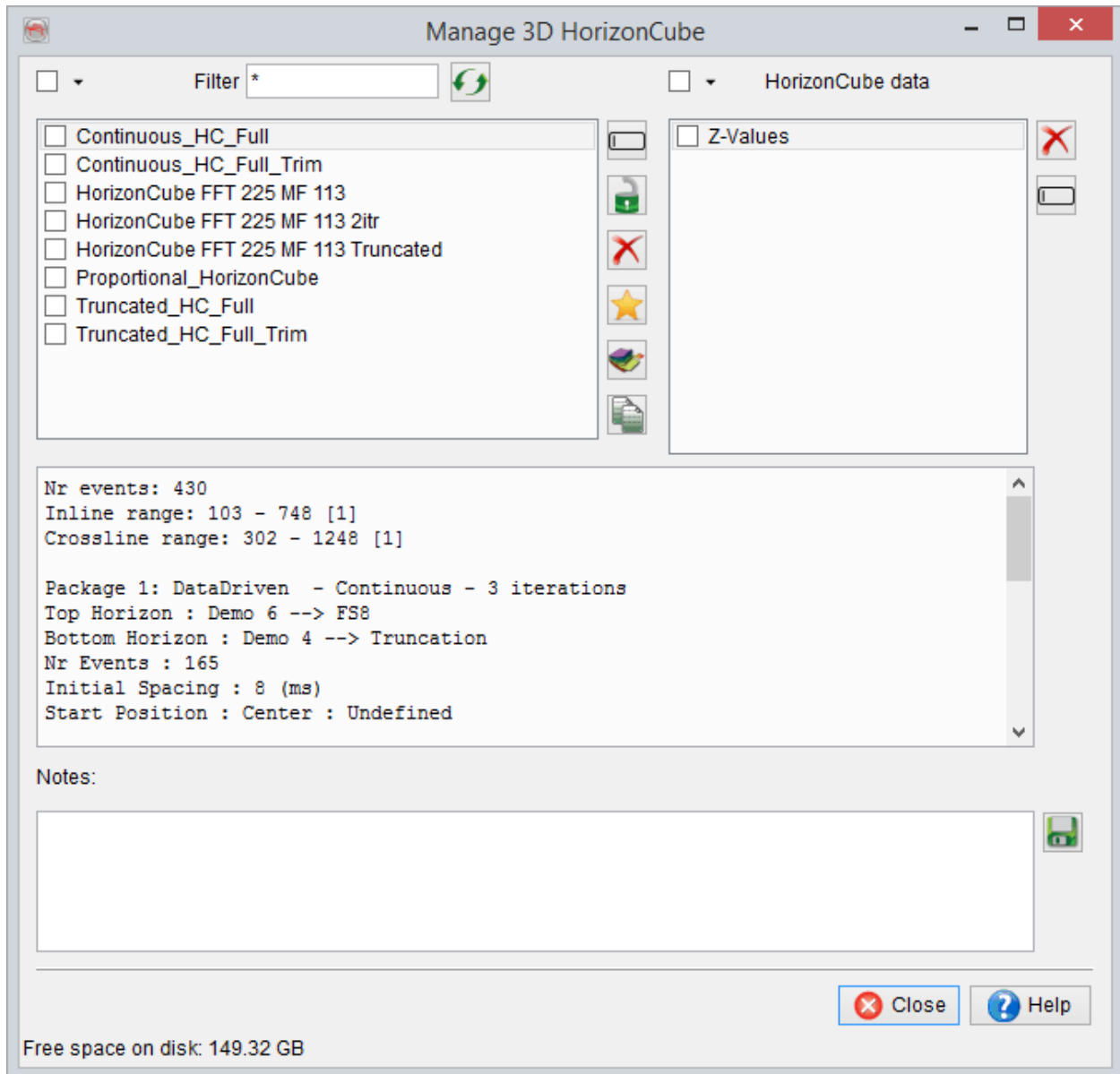


Options








- **Output Events:** It is used to export a selected event(s) to an ASCII file. Multiple selection could be made by dragging the mouse over a range of events.

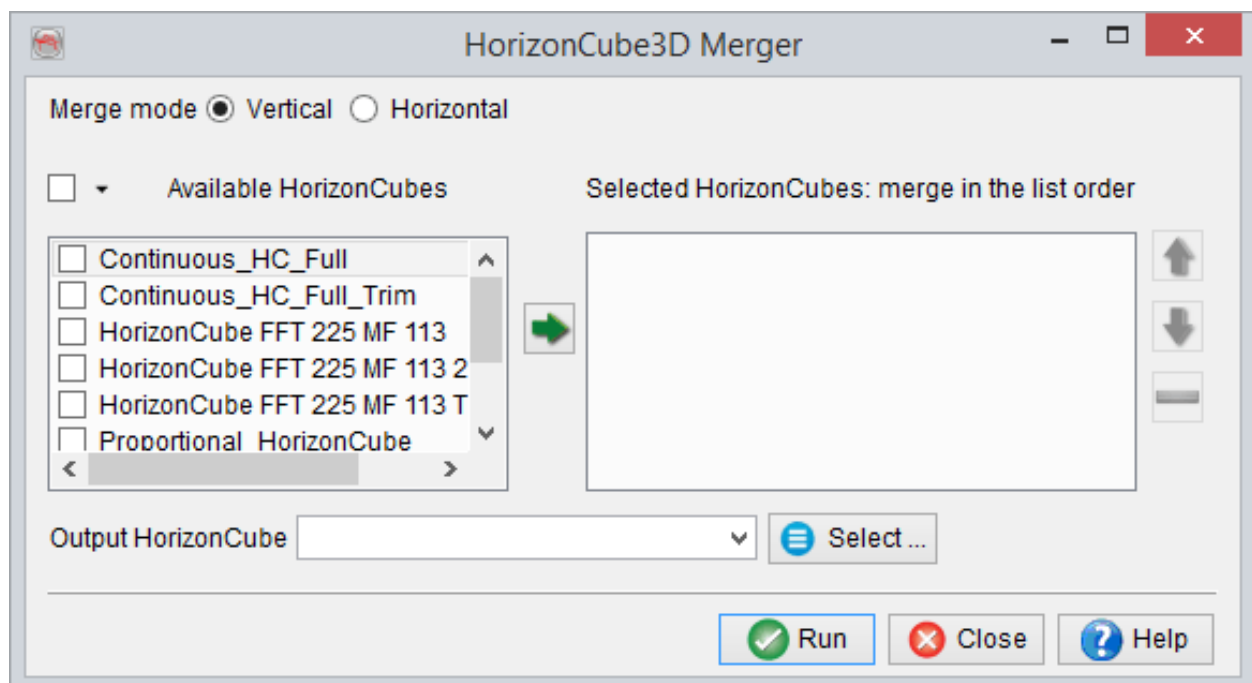
- **Output Package:** It is used to export the events in a selected package(s) to ASCII files. Multiple selection of the packages could also be made by dragging the mouse over multiple packages. To export a package, the output location must be a directory.
- **Area sub-selection:** It is an optional sub-selection to export the events. The user can restrict the output files within an area of interest defined by a range/polygon/table.

5.6 Manage HorizonCube



Options

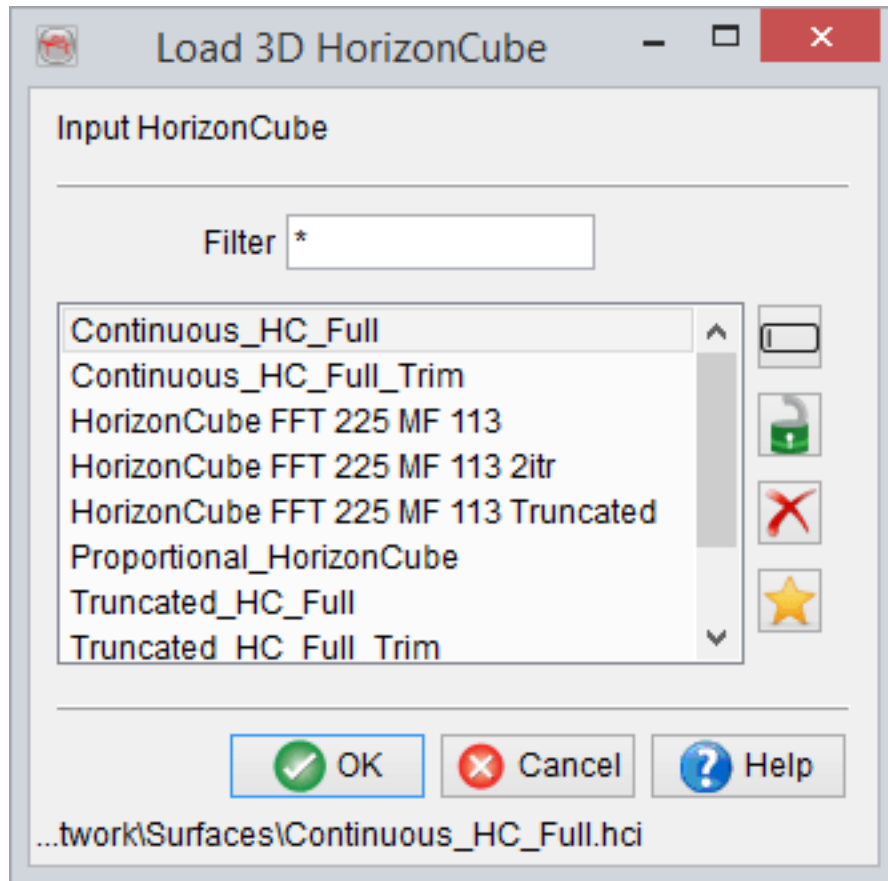
-  *Rename*: Rename the selected HorizonCube.
-  *Lock*: Sets the selected HorizonCube to a read-only mode and disables editing.
-  *Remove/Delete*: Removes the selected HorizonCube from the survey.
-  *Set as Default*: sets the HorizonCube to default.
-  *Copy Horizons*: Extracts horizons (or patches) from the selected HorizonCube. The 'copy to horizon' button extracts one or more seismic horizons from the HorizonCube. This launches the above window. Either a horizon geometry from one sequence (i.e. Package) or from individual automated horizons (Event) can be extracted. The area sub-selection can also be used to restrict the output horizon geometry either within a volume or within a polygon. The output horizon can then be used for further analysis.
-  is used to *copy the horizon to a new object*.
-  is used to *save the file*.



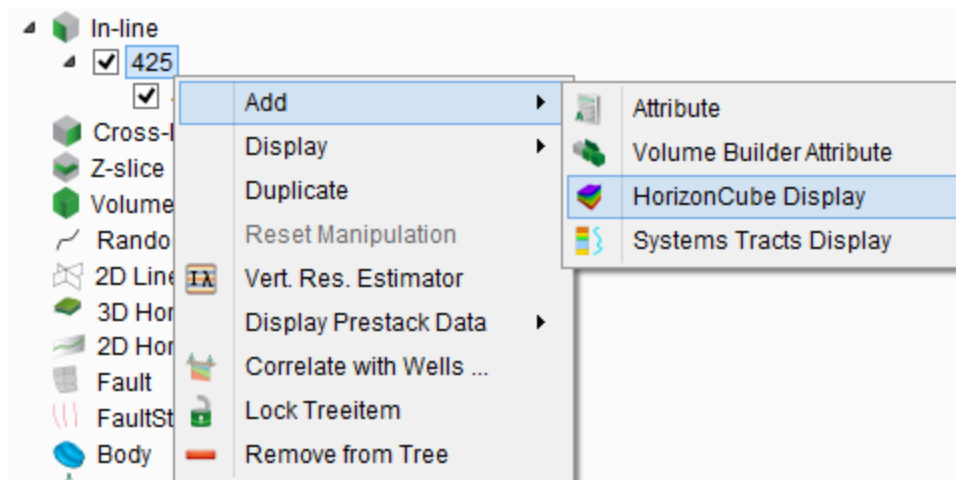
HorizonCube Data: It contains the list of all attributes that are computed and stored along HorizonCube using [HorizonCube_Control_Center](#). A HorizonCube data is equivalent to "horizon data". It contains set of grid based attributes that are computed along selected/entire events of HorizonCube. This data can also be converted to a volume using [HorizonCube Data attribute](#).

5.7 HorizonCube Display Options

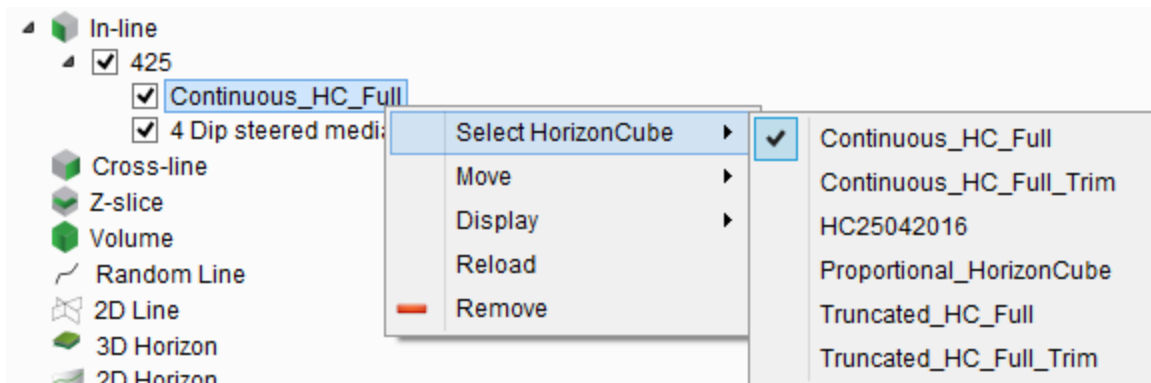
A HorizonCube is selected from the OpenTect Processing Menu (Processing > HorizonCube > 3D/2D..). The HorizonCube Control Center pops up. In this window, the active HorizonCube can be selected via the select button, from the window below:



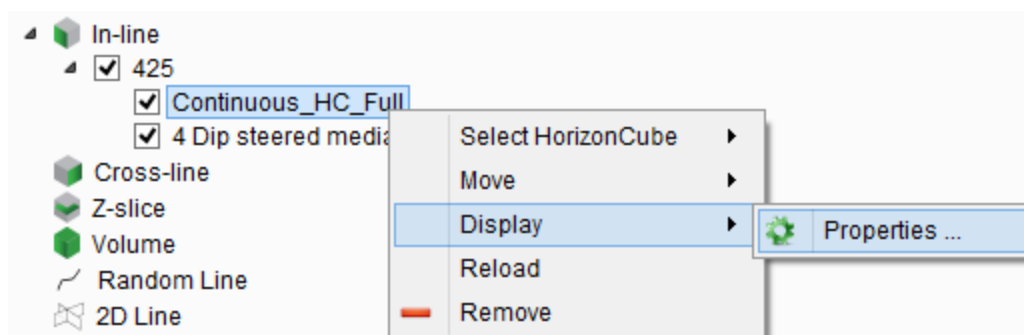
Alternatively, a HorizonCube can be selected directly on an element (e.g. Inline/Cross-line/2D Line):

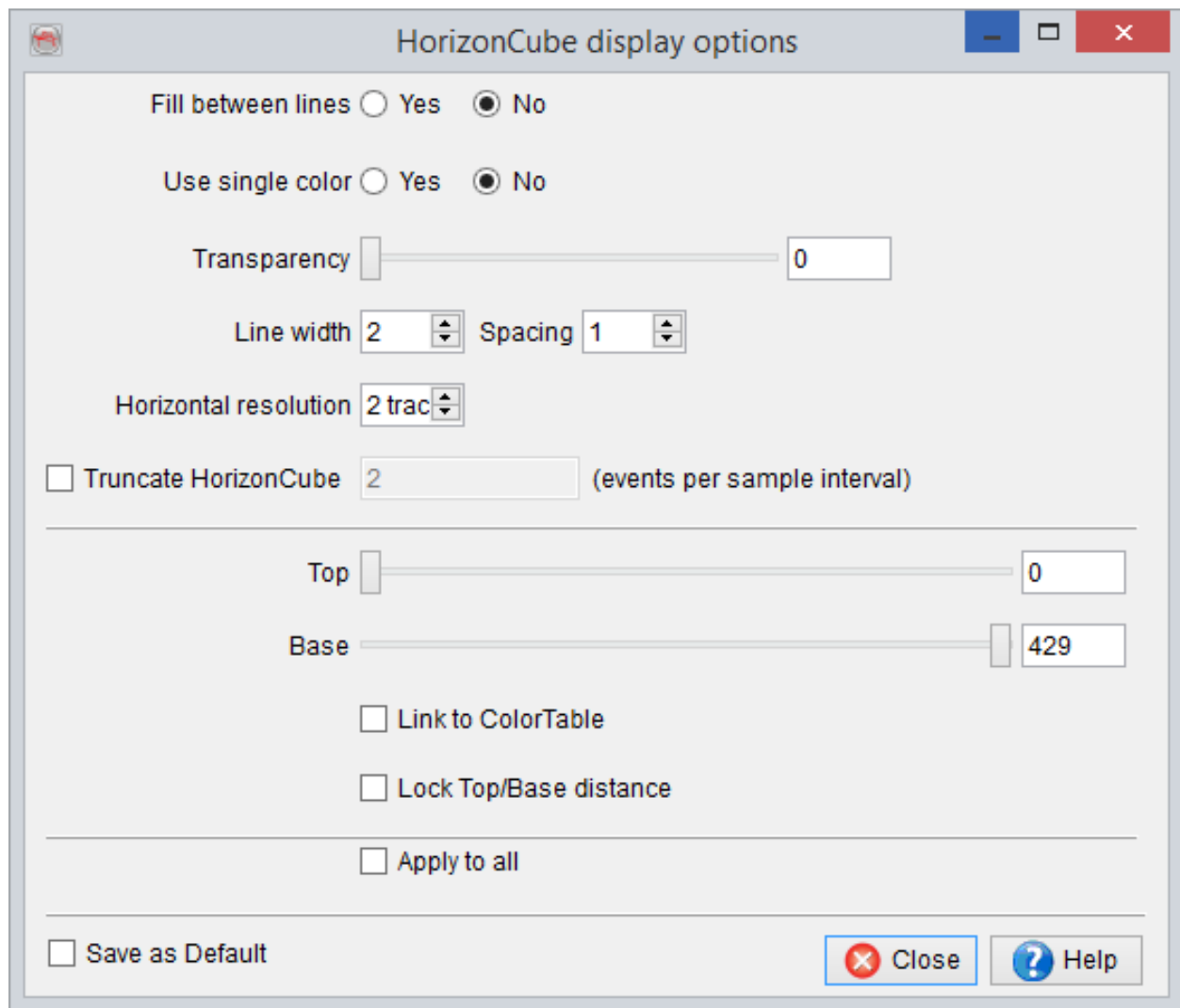


Changing the selected HorizonCube can be done via the control centre (as mentioned above), or directly via the element in the tree. Here we see an example of changing the HorizonCube displayed on an inline:



HorizonCube display options





The display properties of an active HorizonCube in the scene.

From the tree, right-click on the displayed HorizonCube and select 'Display > Properties' menu item. It will launch a new dialog appears as shown above:

In this particular window, you can change the line styles (fill/ single color display), spacing between horizons, line width. Importantly, a slider (top of bottom) is also available to slide the horizons. For instance, if you have displayed thick horizon lines and you want to see that how the layers are developed, you can move the slider within same

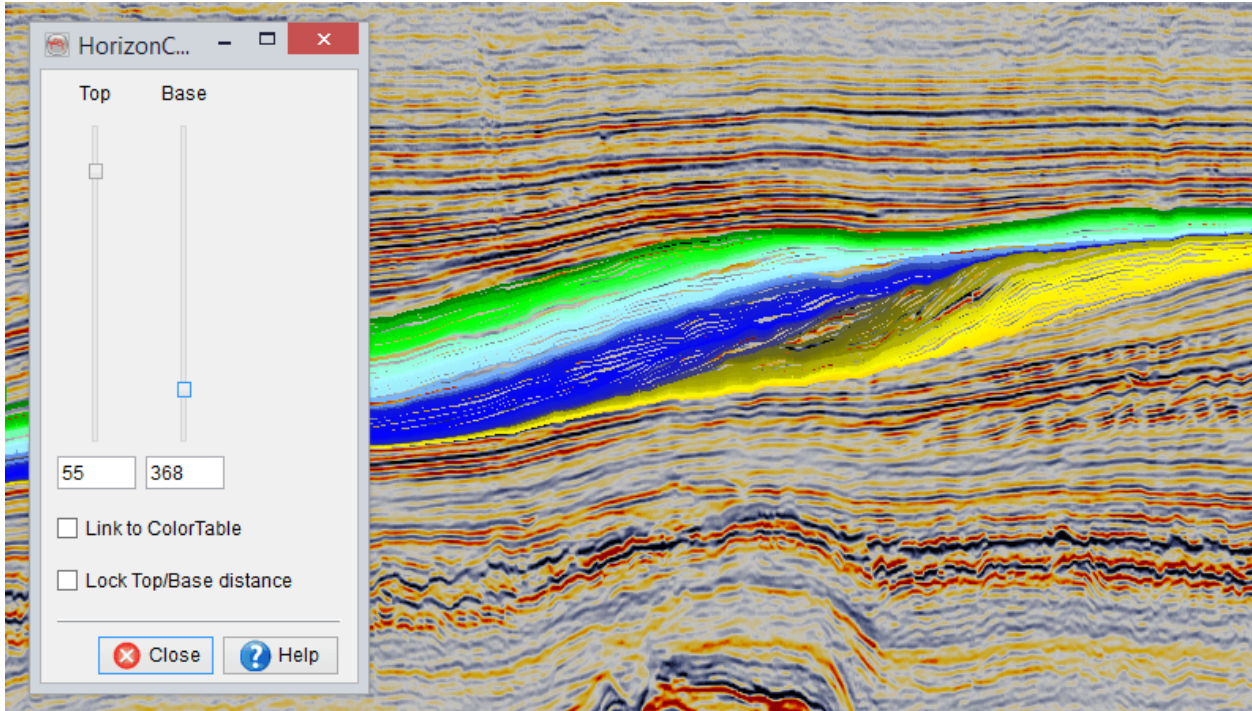
window. The horizontal resolution option will increase or decrease horizontal smoothing according to the selected traces.

Truncated HorizonCube is an option to apply on-the-fly truncation filter on a Continuous HorizonCube. The value of 2 means that only 2 events would be kept within a sampling interval and the remaining would be removed in the display. Note that the truncated HorizonCube is optimum display for interpreting depositional trends in a Wheeler scene. Therefore, this option is a valuable tool to make [SSIS Interpretation](#).

Another useful visualization tool is the HorizonCube Slider:



Once you have the HorizxonCube displayed on, for example, an inline, pressing the HorizxonCube slider button brings up a simple dialogue which allows for control of the horizons displayed in the scene:



The top and base horizons can be manipulated via the slider or set by entering a Z value into either field. Additional options allow the user to '*Link to ColorTable*' and/or '*Lock Top/Base Distance*' to aid browsing.

6 Sequence Stratigraphic Interpretation System



6.1 Introduction

Automated horizon-tracking, Wheeler transforms, and systems tracts interpretations are unique 3D seismic interpretation capabilities that are supported in OpendTect SSIS; dGB's new Sequence Stratigraphic Interpretation System (SSIS for short).

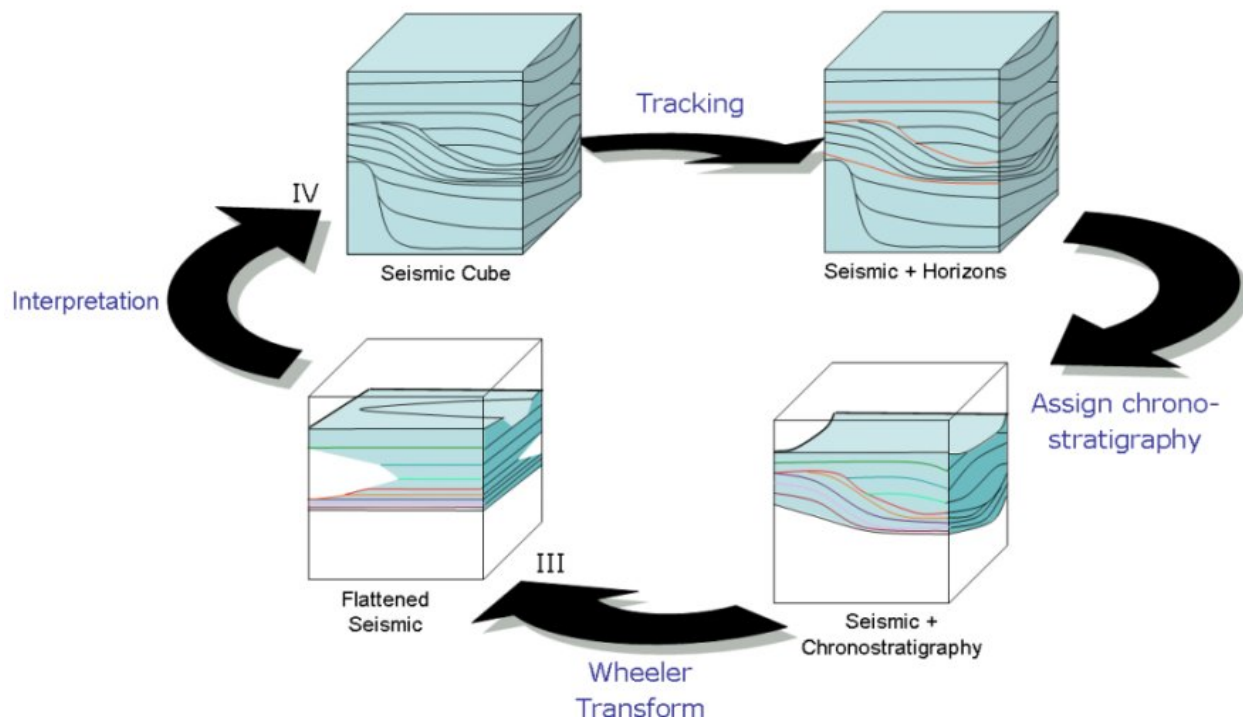
In SSIS, seismic interpreters are offered new ways of visualizing and analyzing seismic data, which leads to better insights of sediment deposition, erosion, and timing. In combination with OpendTect's neural networks plugin, users can follow up with seismic facies clustering and study the resulting patterns and bodies, and their spatial distribution, in relation to geologic timing and systems tracts.

The OpendTect SSIS workflow is an *iterative process* that consists of four basic steps:

First, major sequence boundaries are interactively mapped with OpendTect's horizon trackers (or these are imported from other interpretation systems). Next, all possible intermediate horizons are auto-tracked with sub-sample accuracy. Each intermediate

horizon corresponds to a geologic time line, i.e. a chrono-stratigraphic event, and can be identified by a unique index indirectly to relative geologic age of the event. The index is automatically assigned to each horizon in stratigraphic order according to its relative thickness. Horizons that form the base of one sequence and the top of the sequence below it, are assigned the same index. This index can be manually overruled for both indexes separately. In this way the duration of deposition and duration of hiatuses can be changed. When a horizon is diachronous (time transgressive), the index is the youngest index for the top of a sequence and the oldest index for the base of a sequence.

Two auto-track modes are supported: *model driven* and *data driven*. In the model driven approach the HorizonCube is calculated by interpolation or by adding horizons parallel to upper or lower bounding surfaces. In the data-driven mode, seismic horizons are auto-tracked by following the local dip and azimuth of the seismic events. This mode requires OpendTect's dip-steering plugin to pre-calculate the SteeringCube containing the dip and azimuth information.



SSIS Workflow

The third step in the process is the actual *Wheeler Transform*. Basically, this flattens the seismic data (or derived attributes) along the auto-tracked horizons and honors truncations and erosional/depositional hiatuses. Studying the data in the Wheeler transformed domain increases our understanding of the *spatial distribution and timing of sediment deposition*. To quote sequence stratigrapher Peter Vail: "*You never fully appreciate the implications of a sequence stratigraphic interpretation until you've transformed it to a Wheeler diagram*".



The fourth step in the SSIS workflow is *systems tract interpretation*. Inspecting the spatial distribution of the sequences and lap-out patterns of seismic events, in both the normal domain and the Wheeler transformed domain, enables the user to segment the seismic sequences into systems tract. Systems tracts are specified per HorizonCube range. To accommodate different naming conventions, the software allows the default systems tract names (*High Stand, Falling Stage, Low Stand, Transgressive*) to be replaced by user-defined names.

With the HorizonCube calculated and optionally systems tracts interpreted, the user can continue the sequence stratigraphic analysis with seismic facies interpretation. Visualizing systems tract-interpretations together with the HorizonCube and overlaying the normal- and Wheeler transformed domains helps in identifying depositional features of interest. More advanced seismic facies analysis is possible with OpendTect's neural network plugin. Waveform segmentation along any HorizonCube event is a simple and straightforward approach for visualizing seismic patterns per stratigraphic event. Similarly, seismic attributes can be clustered by a neural network to reveal 3D bodies. Further, the user can train a neural network to recognize seismic bodies in 3D from user-defined examples (supervised mode). These approaches have already been supported in OpendTect's neural network plugin. The new version of SSIS allows the results to be visualized and analyzed in the stratigraphic domain, i.e. without distortion

by the structure. To appreciate this, you could load (a subset of) a Wheeler-transformed seismic volume (e.g. an attribute, neural network clustered result, or AI cube) in the volume viewer of the Wheeler scene and use the time-slice display to scroll the data. Time slicing in the Wheeler domain corresponds to horizon-slicing in the normal (structural) domain, hence all patterns observed in each slice belong to the same geological event.

6.1.1 SSIS Toolbar

The SSIS toolbar contains the icons that are used by a user to manage the HorizonCube and corresponding stratigraphic interpretations:


-  Launch [SSIS interpretation module](#). In the SSIS interpretation window the user can interpret sequence stratigraphy by using the HorizonCube. Note that the correct HorizonCube needs to be selected first.
-  The HorizonCube-slider is used to visualize the seismic horizons in a stratigraphic manner i.e. slide the horizons from top to bottom to visualize the depositional patterns.

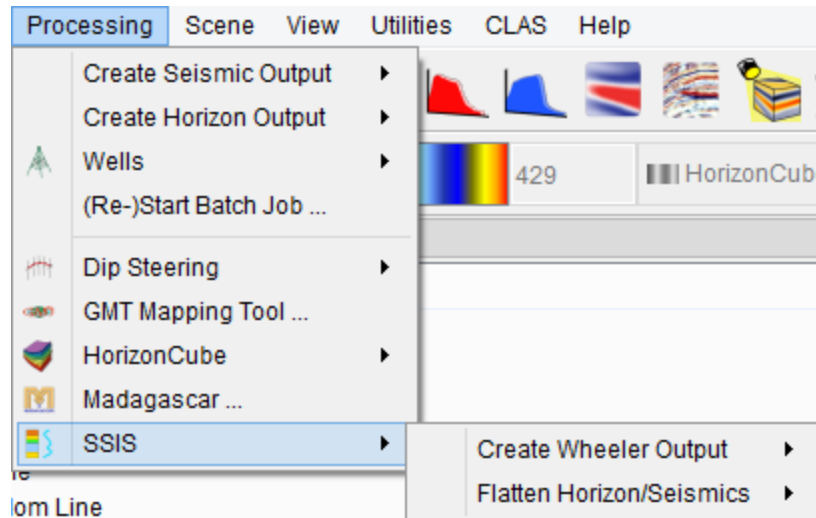
6.2 Interpretation Window

See following chapters below.

6.2.1 Overview

In OpendTect-SSIS, an interactive module is used to interpret a sequence in details. This is done by displaying the HorizonCube on an element (inline/crossline/2D line) and launching the interpretation setup:

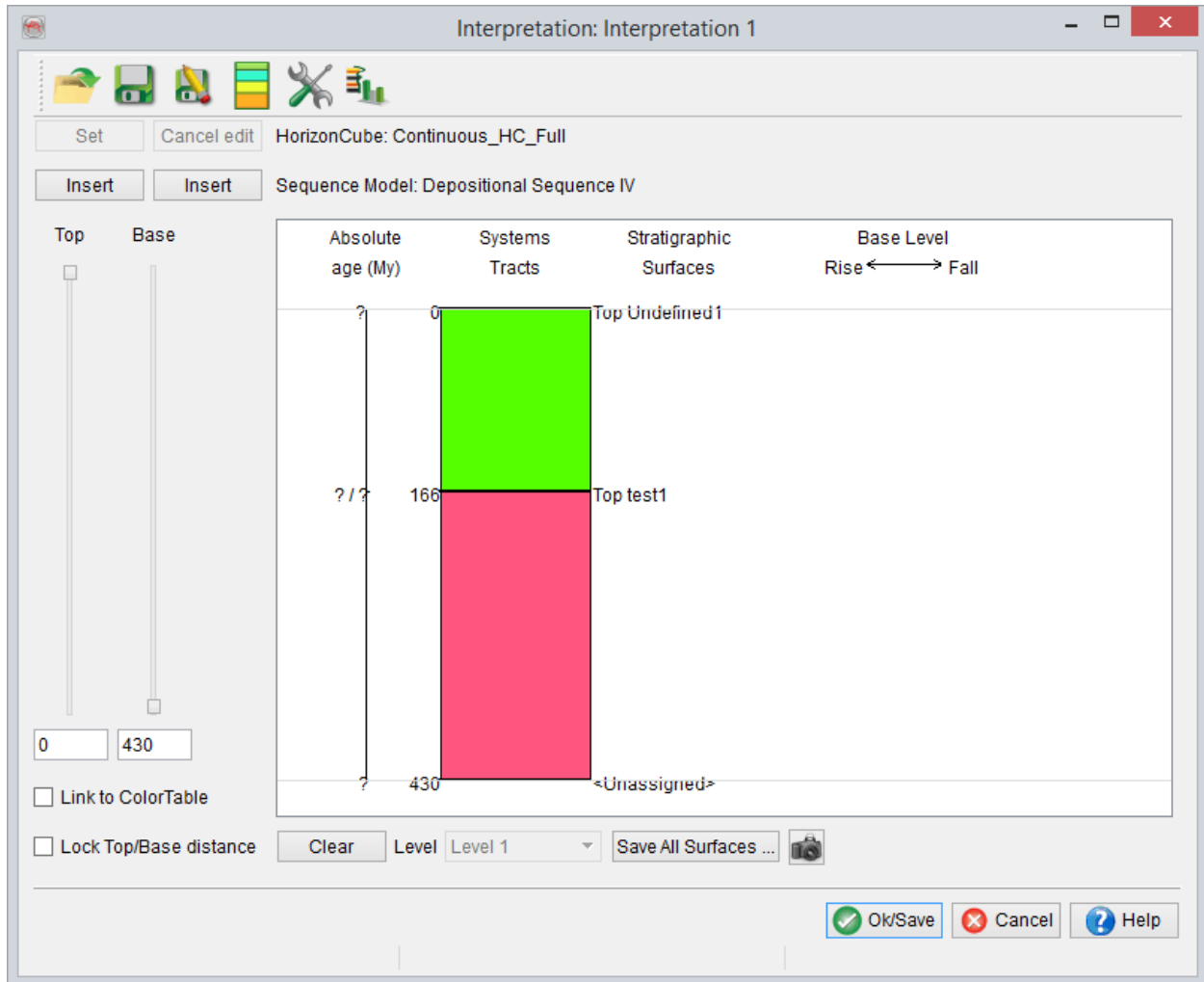
- Select the HorizonCube.
- Add the HorizonCube display in the tree.
- From the SSIS menu select *SSIS > Interpretation > New/Edit* or use the short-cut button  from the SSIS toolbar.





Launch the interpretation window




This window is used to perform a sequence stratigraphic interpretation from the HorizonCube (2D/3D). The interpretation is done in an interactive manner i.e. by sliding the [HorizonCube-slider](#) up and down (right panel) and visualizing the depositional changes in the OpendTect scene. When a shift in depositional pattern is apparent, a sequence stratigraphic surface can be inserted by pressing the *Insert* button. This is also

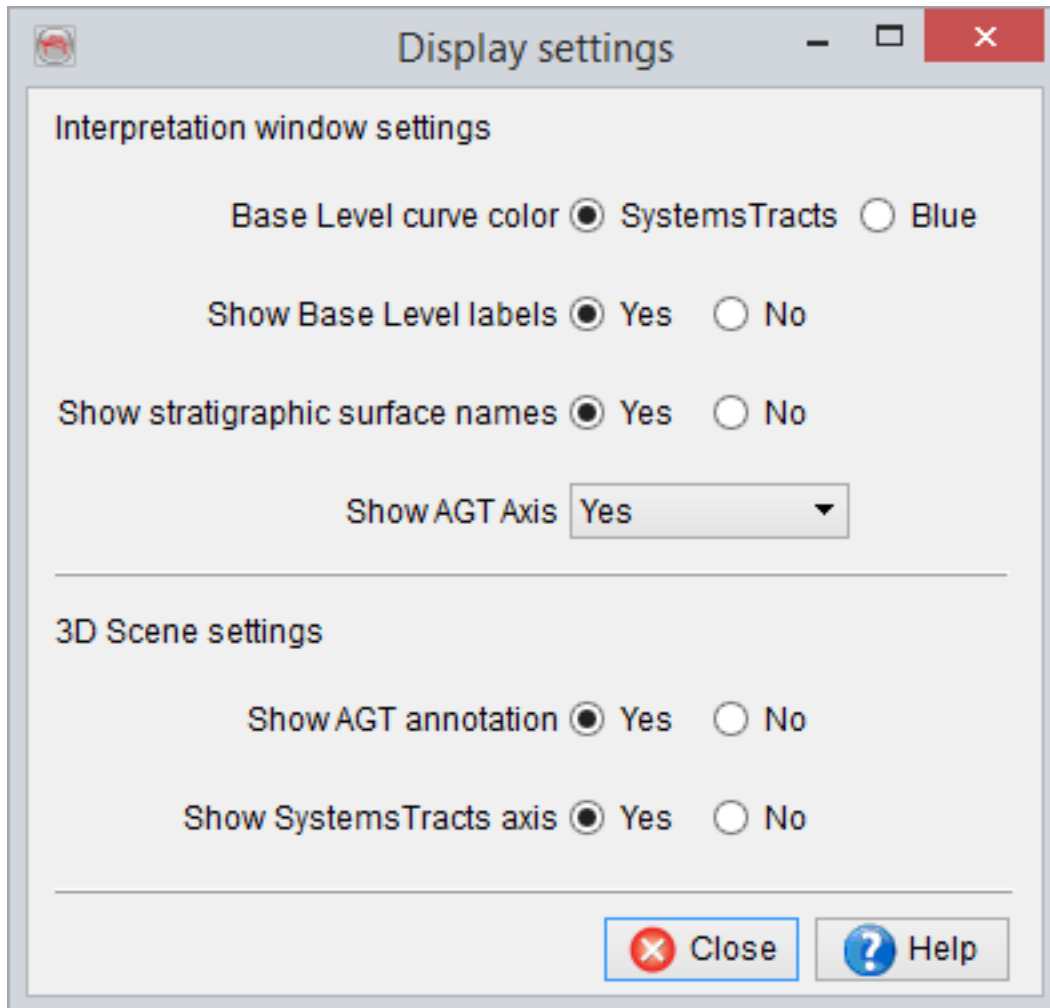
interactive, i.e. the slider will update the display in the OpendText scene and the surfaces will be added in the graphical area (left panel) of the Interpretation window. An example of this interpretation window with a sequence stratigraphic interpretation is shown in the figure below.



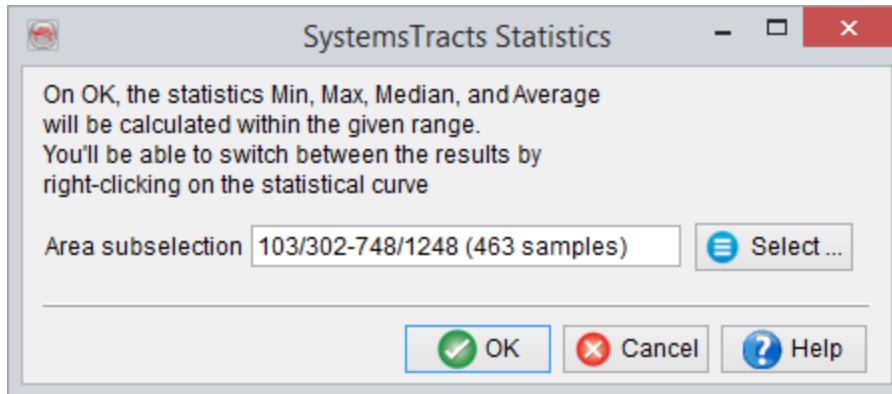
Interpretation window for SSIS


-  Open the stored interpretation. Note that more than one interpretation can be stored for any calculated HorizonCube.
-  Save the interpretation.

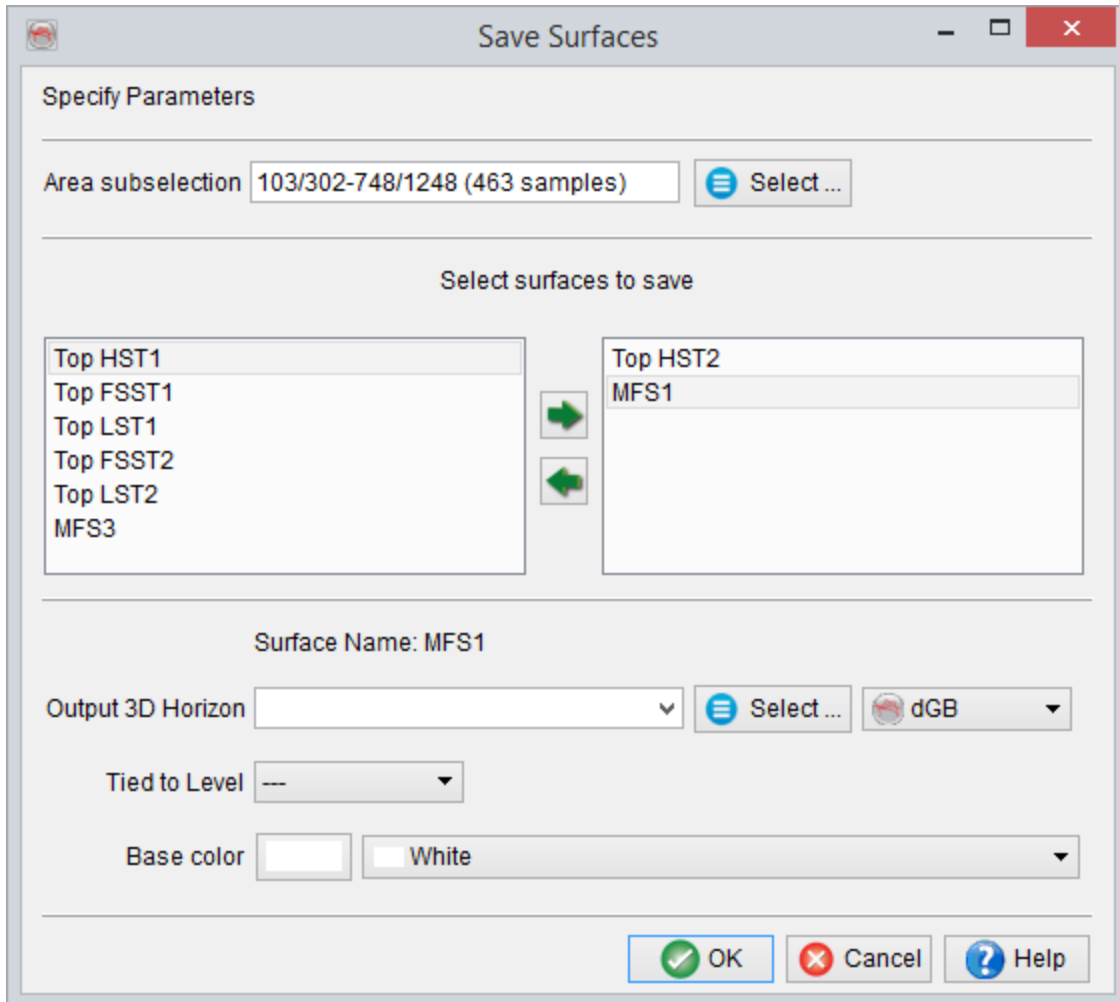
-  Save the interpretation with a new name i.e. Save As..
-  [Manage the Sequence Model.](#)
-  Display settings:



-  Calculate the statistics for your model:



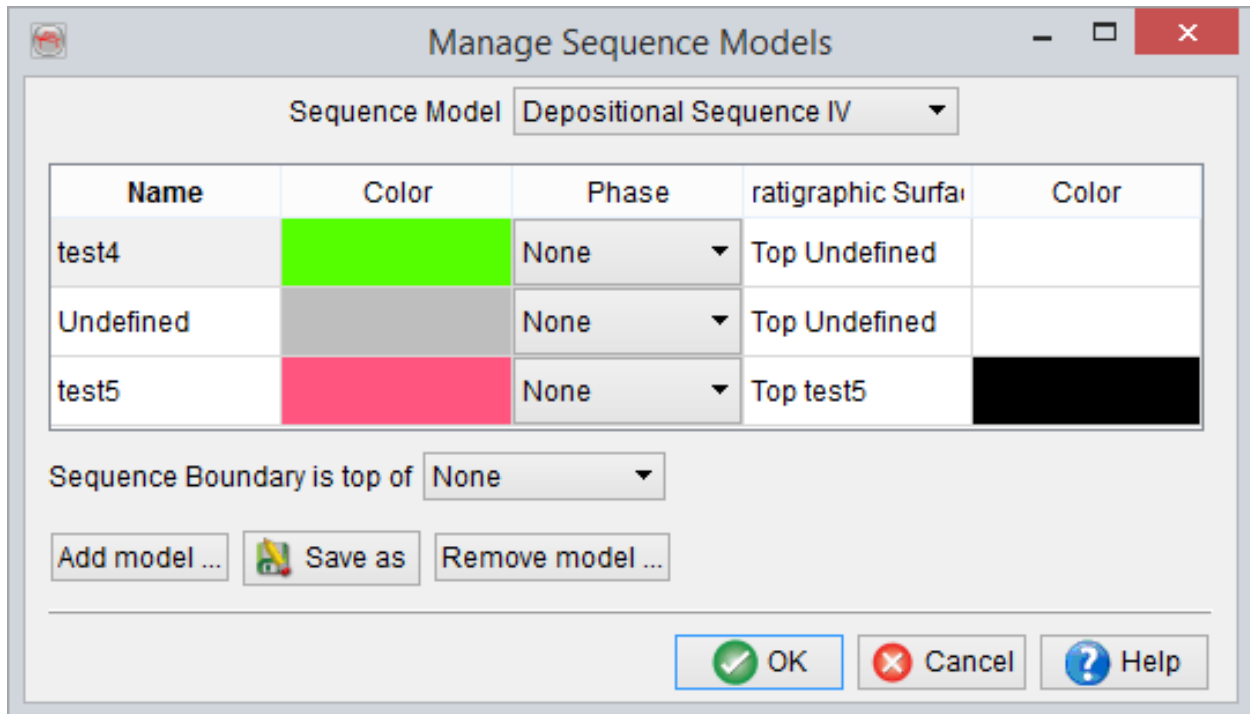
-  Take a snapshot of the graphical area and save it.
- *Link to ColorTable*: This options is used to lock the color table while moving the slider position up and down.
- **Save All Surfaces ...** will present you with a dialogue where you may select the surfaces you wish to save and set basic options for these:




Colour-coded Base level Curve

The base level curve colour could be changed in such a way that a segment bounded by two nodes/surfaces represents the corresponding systems tract colour. This is done by right-clicking on the curve and selecting the option colour coded.

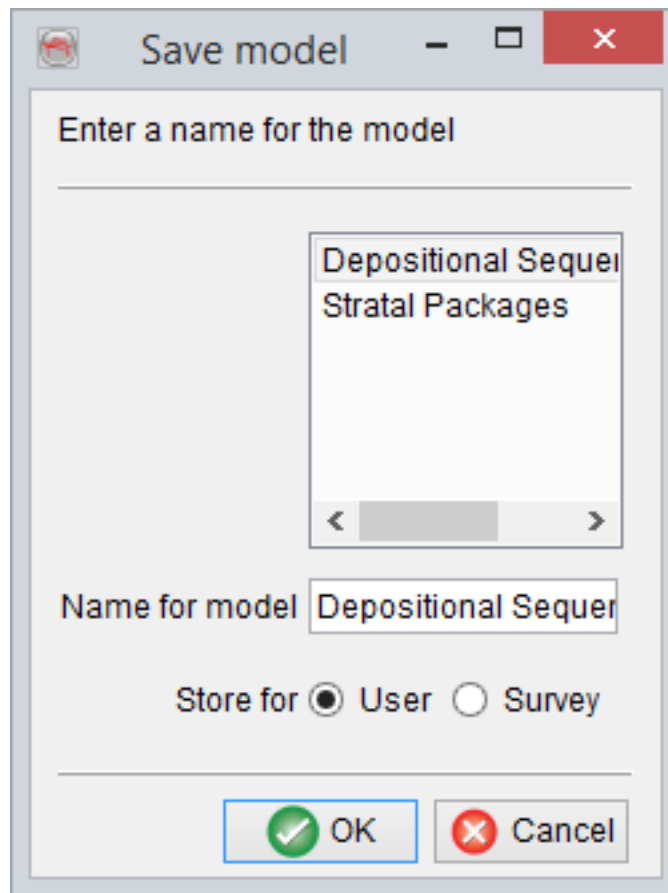
6.2.2 Select, Define a Depositional Model



Sequence Models Dialog

Seismic sequence stratigraphic interpretations in SSIS are started by selecting/creating a depositional model. The models are color-coded representations of systems tracts in OpenTect and the corresponding base level curve is automatically drawn in the Interpretation system. To select/define one of such model, click the *Select Sequence Model*  icon. By default, there are five different standard depositional sequence models available in the setup. You can either select any one of them or press *Add Model* to add new model in the *Systems tract* setup window. The model can be saved at a user or at a survey level. The user level is used to save the model for the current OpenTect user name only and it will be accessible by that user name only in all

surveys. However, you can also save it at a survey level, which means that the model would be available to all surveys in the current surveys directory.



A user defined Sequence Model

In order to define a new model, press the 'Add Model' button and type the name (*Name for Model*) in the pop-up window. Store the newly created model either for the current user or for the entire survey. On 'OK', a blank model will be added in the systems tracts Setup. Right-click on the *Undefined* text in the *Name Column* and select *Insert row after* sub menu. This will add a blank line below the undefined row. Type a name for the systems tract of the newly created model. To assign a color for the newly added systems

tract, double click on the white cell of the same row. Select an appropriate color for the systems tract.

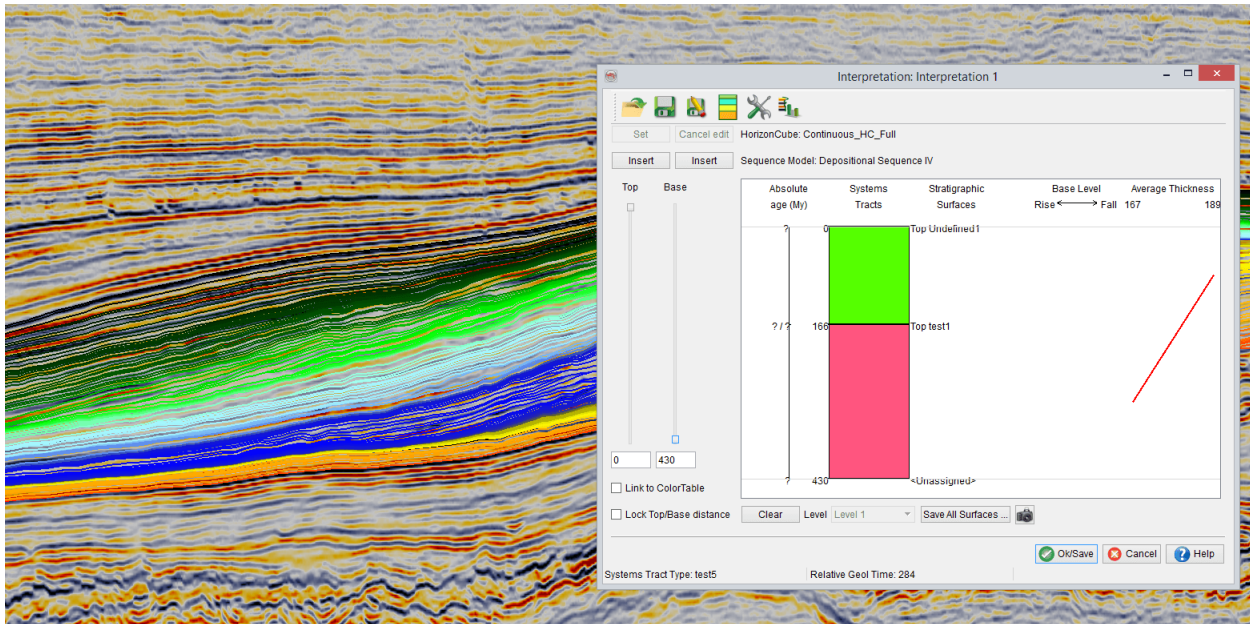
The sequence boundary could also be changed for a depositional model. You may put the sequence boundary at a selected systems tract name. This sequence boundary will appear in the base level curve.



Most of the fields (name, color) are editable by double clicking on the corresponding cell. You may also override the color and name as you like. The standard colors and names of the surfaces are used while converting the surfaces into general OpendTect horizons.

6.2.3 Interpretation Workflow

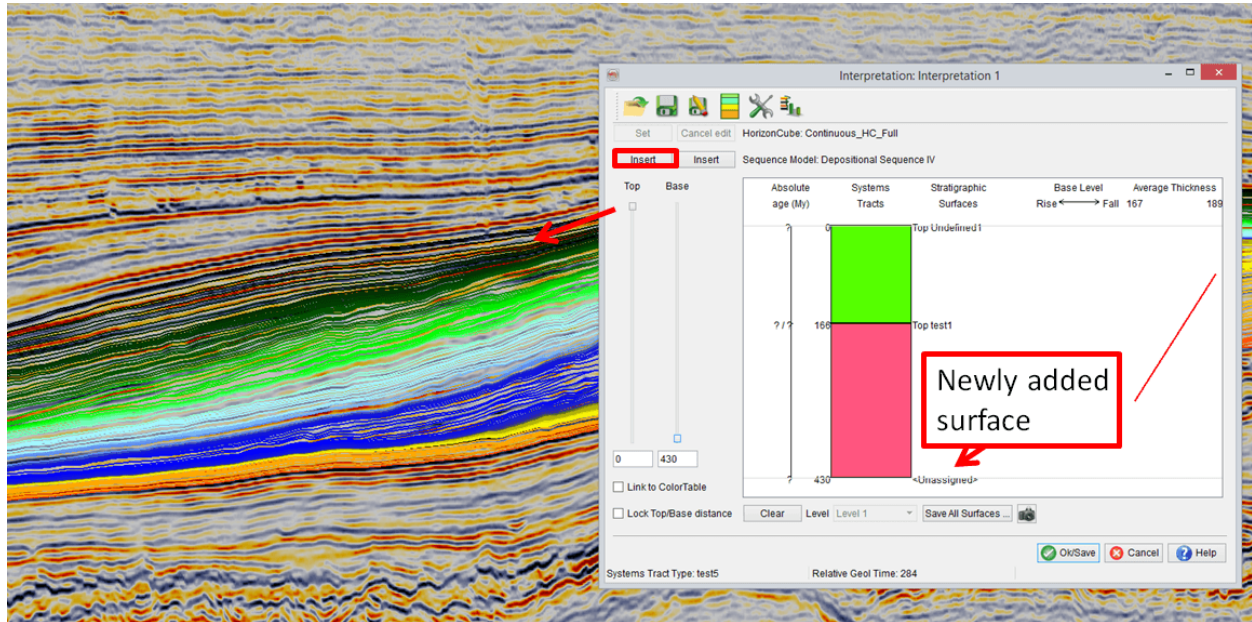
After defining/selecting the right [depositional model](#), you are ready to start interpreting sequence stratigraphic surfaces and the systems tracts. First, create an overlay of the HorizonCube on an inline/crossline/2D line, and position the scene in a way that you will notice main depositional changes (see below).



The displayed seismic data is on an inline and the displayed HorizonCube is created for the entire survey range. The interpretation window is placed to the left so that the identified sequence stratigraphic surfaces can be inserted easily.

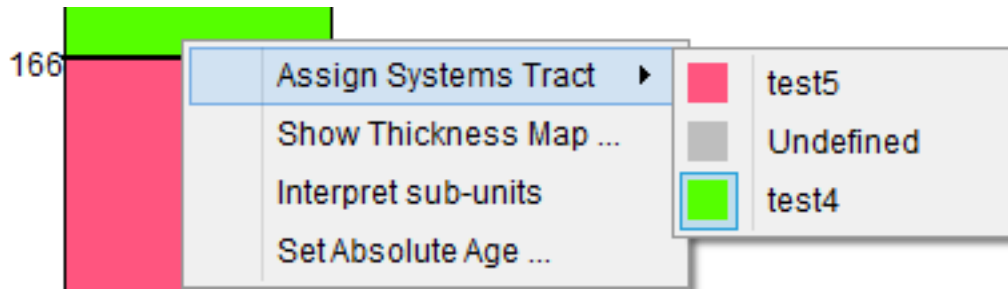
Check the *Link to ColorTable* option to optimize (stretch/squeeze) the colors for the HorizonCube relative to the slider position. Now, move the *Top* slider slowly down/up to observe the depositional changes. Start your interpretation from the bottom events of the HorizonCube. Move the *Top* slider up from the lowest position and position it where

you observe a depositional shift or a sequence stratigraphic surface. At this point, you can add a surface to the current interpretation. Press *Insert* button to add a sequence stratigraphic surface in the display panel. This will add an surface in the Systems Tracts (grey) column (as shown below).



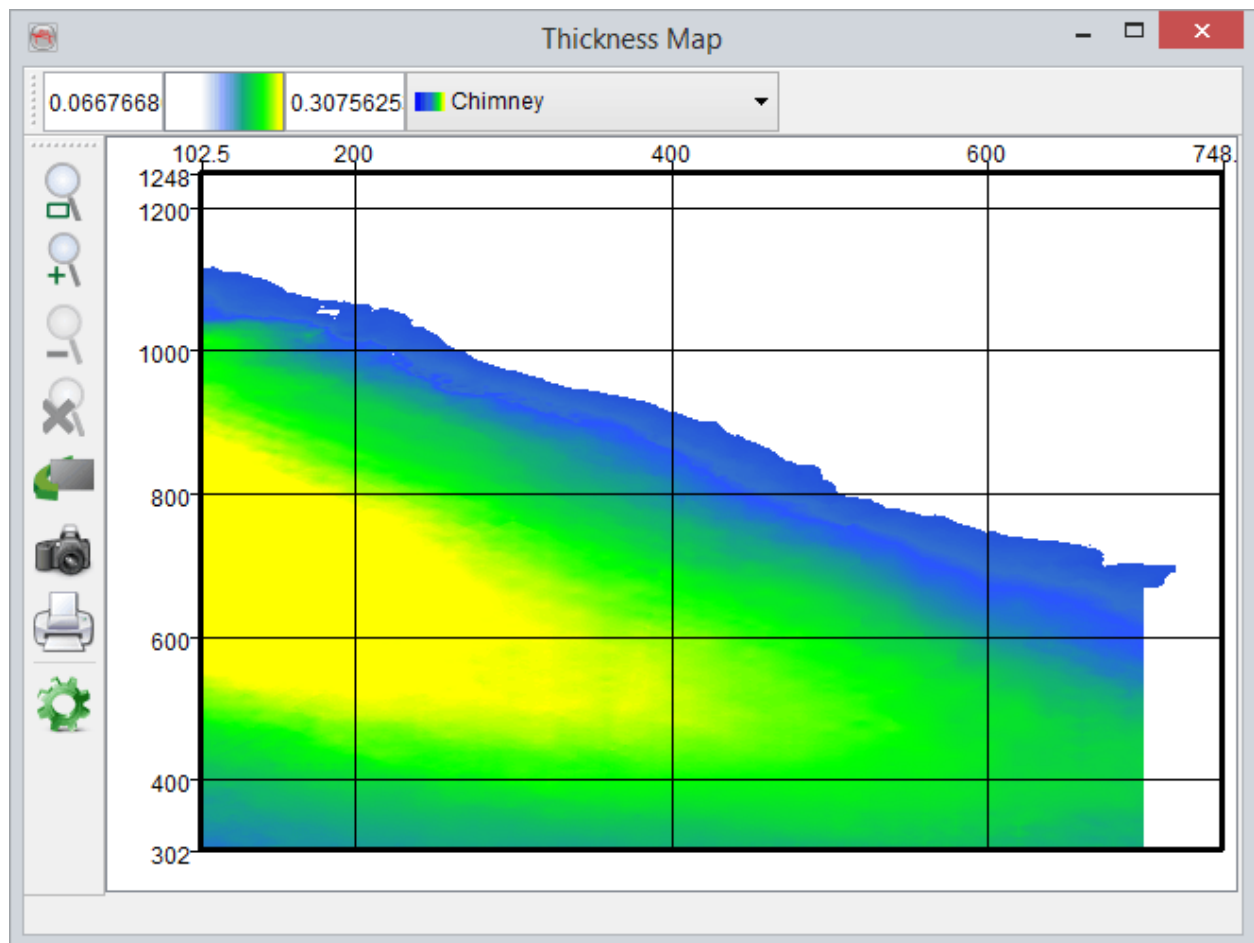
The top slider position is adjusted to view at where the depositional changes are observed (e.g. in this case from transgression to normal regression).

Now, the next step is to assign a name to the sequence stratigraphic surface. Right-click within the gray area i.e. just below the newly inserted stratigraphic surface. In the pop-up tree, you will see four options. From the 'Assign Systems Tract' option, select a sequence systems tract. Once you have selected this, the package will be filled with the corresponding color. Note also, that the relative change in base level is automatically plotted according to the depositional sequence model. Now you can continue interpreting the entire section by positioning the slider to next position(s) and following the same steps.





The pop-up menu is briefly explained below:

- *Assign Systems Tract*: Note that each depositional model has its own systems tracts-definition. Thus, if you have selected depositional model IV the corresponding systems tract that you can assign will be listed as above.
- *Edit Boundaries*: If you are not satisfied with the inserted stratigraphic surface for the systems tract, select the 'edit boundaries' option from the tree. On the right panel you can move the HorizonCube-slider to a new position. When done, press Set button or Cancel edit.
- *Show Thickness Map*: This will show the time-thickness variations.

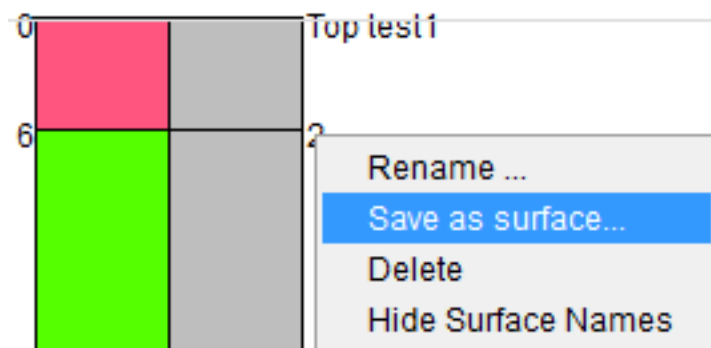


- *Set Absolute Age:* Set the geological absolute age (My) to any stratigraphic boundary. In the next pop-up window, give absolute age for top and bottom of the system tract.

Save Interpretation

Once you complete your interpretation on the entire section, you are ready to save the interpretation. This can be done by pressing the save button . By default, the interpretation is saved with a default name (Interpretation 1). Any HorizonCube can have more than one Interpretations. Therefore, you can also save the interpretation with another name by pressing the Save As button .

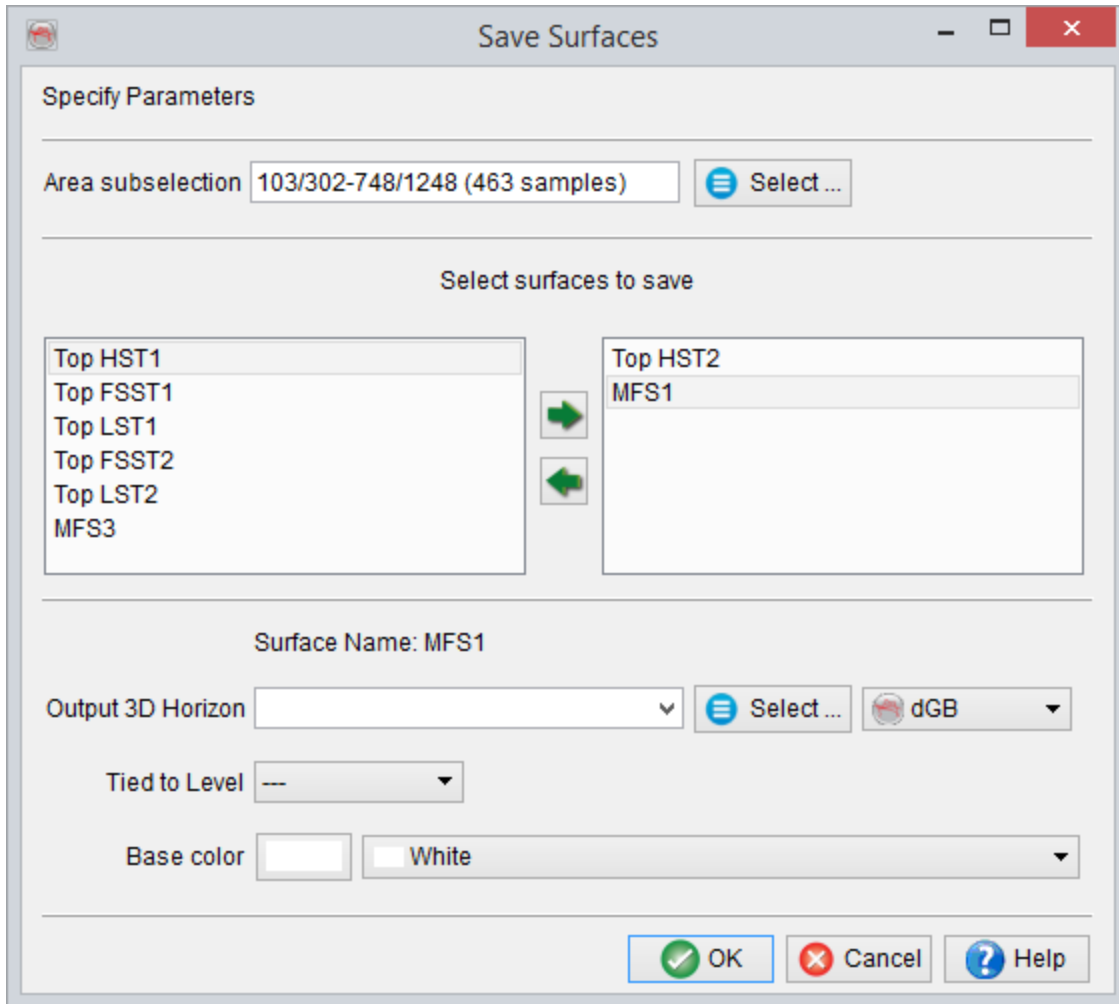
Safe Surfaces



A *stratigraphic surface* can be saved as an output surface. Either right click on the stratigraphic surface name, dots of base level curve or at numeric values of absolute age (if defined).

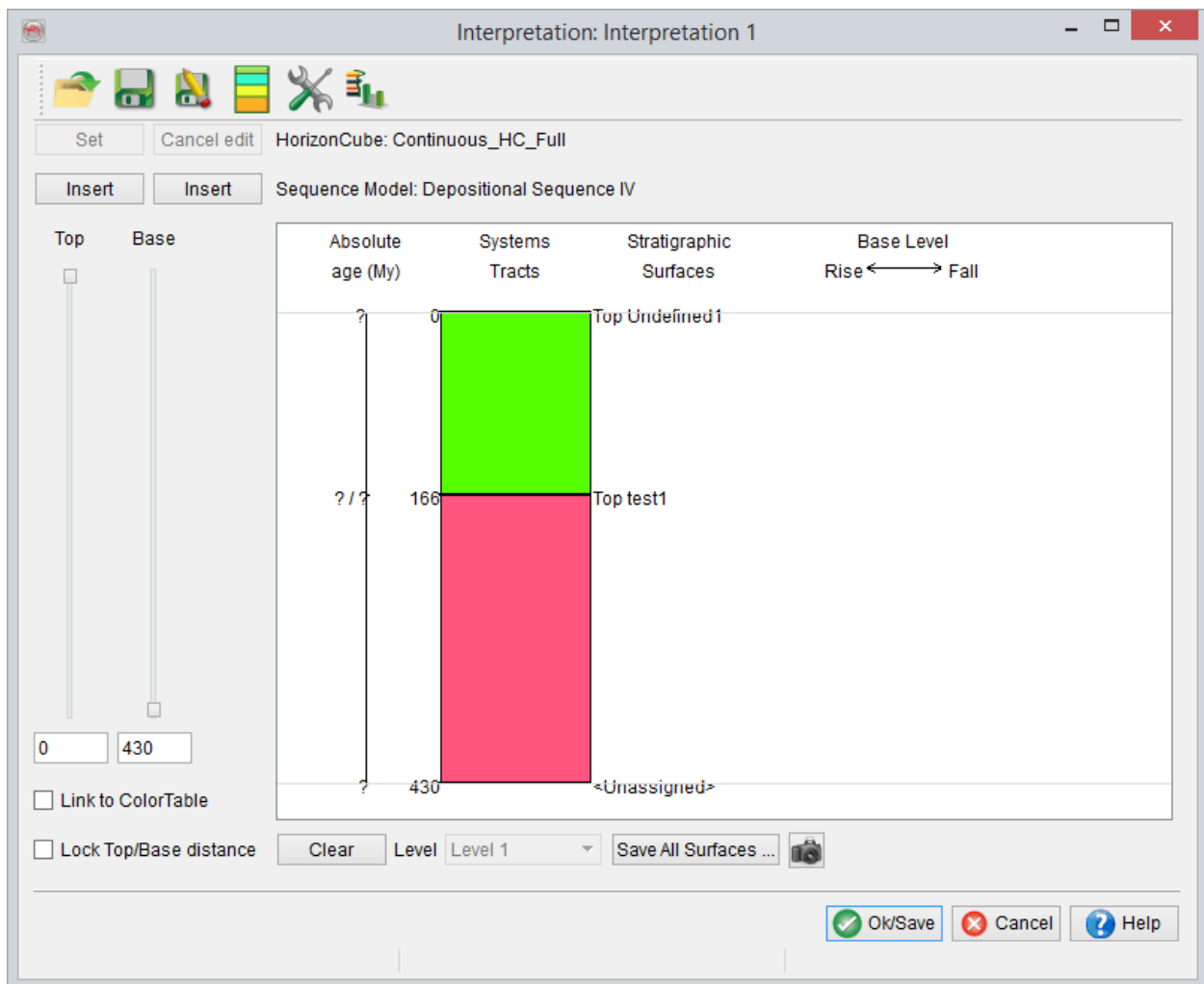
Once you are satisfied with the interpretation, press *save* or *save as* button to save the interpretation.

Select the area and write the name of output surface. This output surface can be displayed later from the Horizon tree.



You can also save all surfaces by pressing *Save all surfaces* button.

6.2.4 2D Interpretation Window

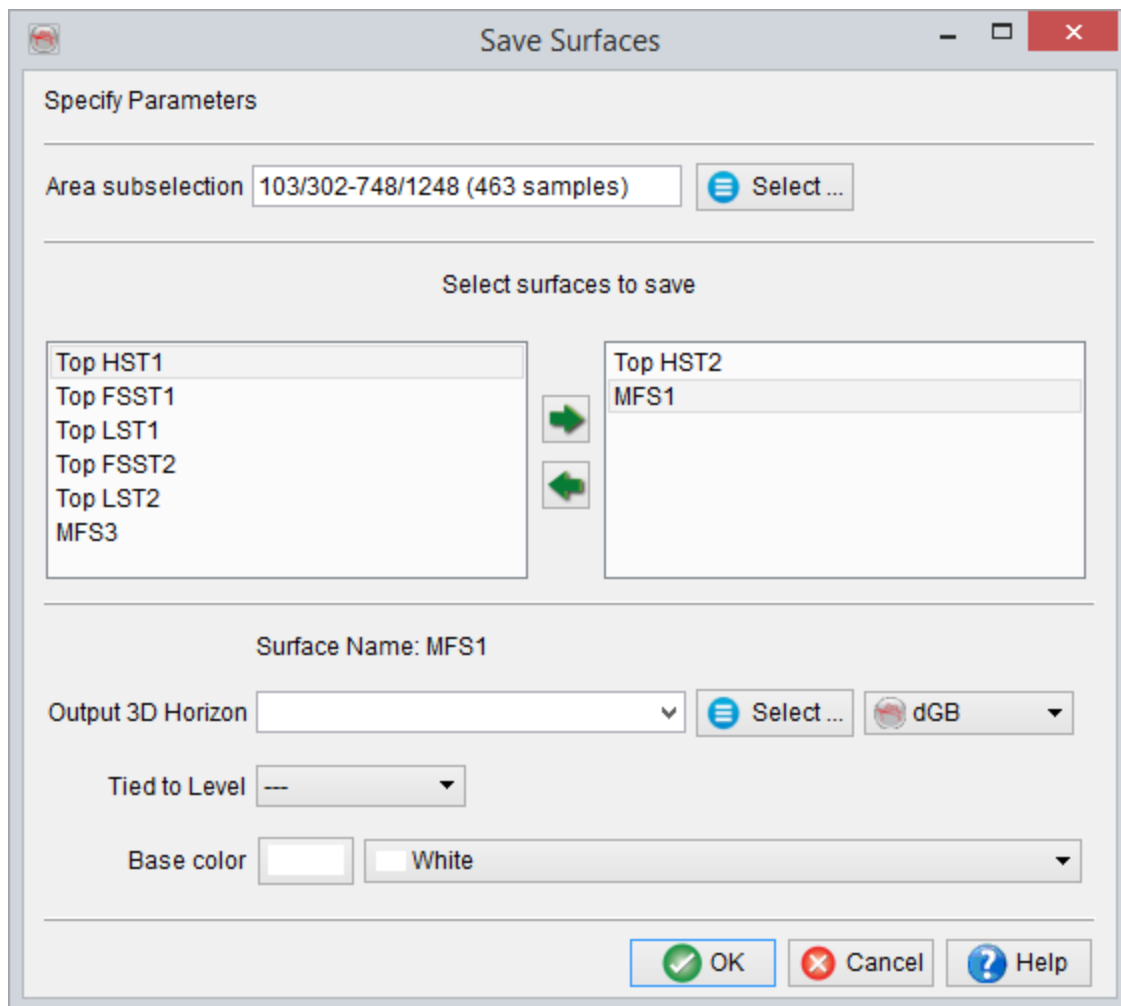


2D Interpretation windows is somewhat identical to the conventional 3D window. The only difference is that you need to select the relevant 2D line in this window on which the interpretations would be made. Note that the slider (Top / Bottom) works individually for a selected line only.

It is also supported to copy one line's interpretation to another line using the *Copy* button. If the interpretation has already been made on an intersecting line, you could copy

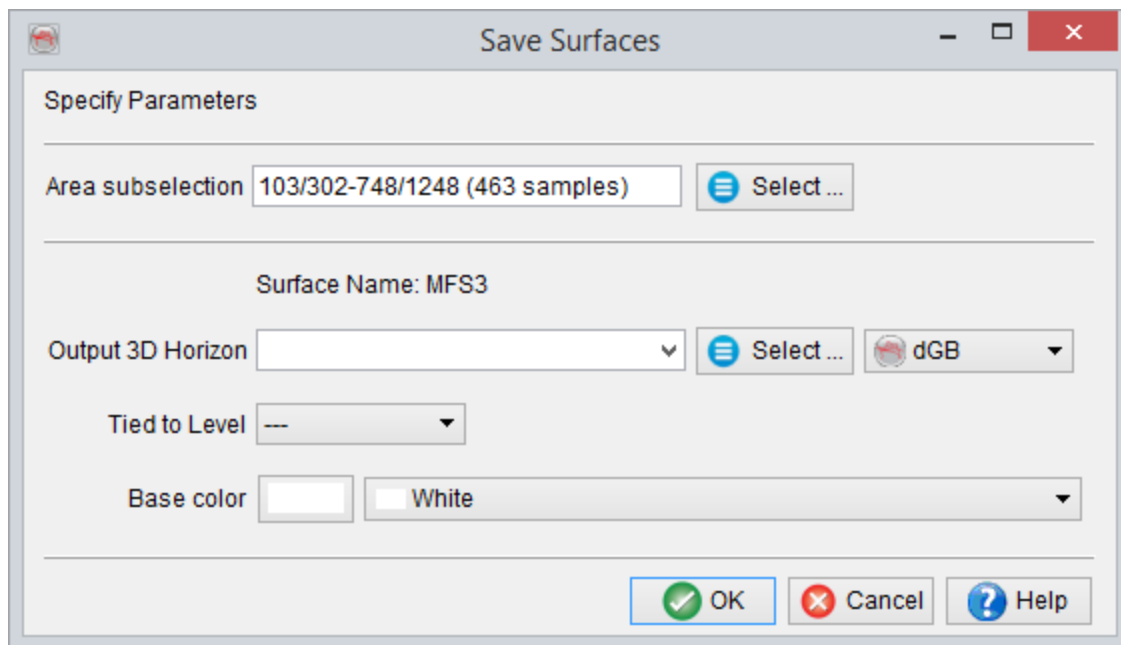
that interpretation to another intersecting line. In this manner, you would keep the same number of systems tracts on the new intersection and the boundaries would be placed automatically. This will also avoid the mis-ties at the intersections.

Save All Surfaces button will allow you to save the interpreted sequence stratigraphic surfaces as 2D horizons. The surfaces that you want to save are added on to the right list box using the arrows. The corresponding lines 2D lines on which the interpretation was made for the selected lines must also be selected by the user. The output name / colour should be mentioned prior to pressing the OK button.



Note that you could also save/rename/delete each surface individually by right clicking on the surface name in the graphical area. It will launch the pop-up menu with the following options:

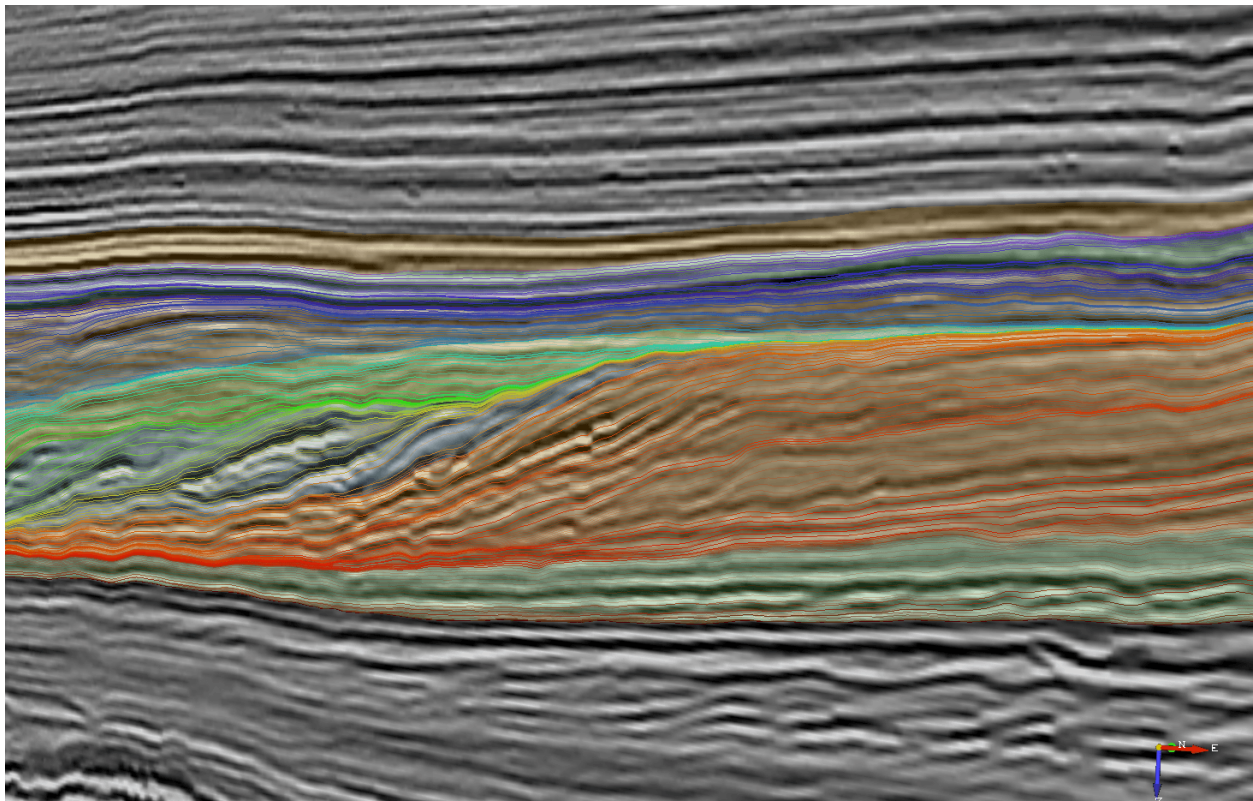
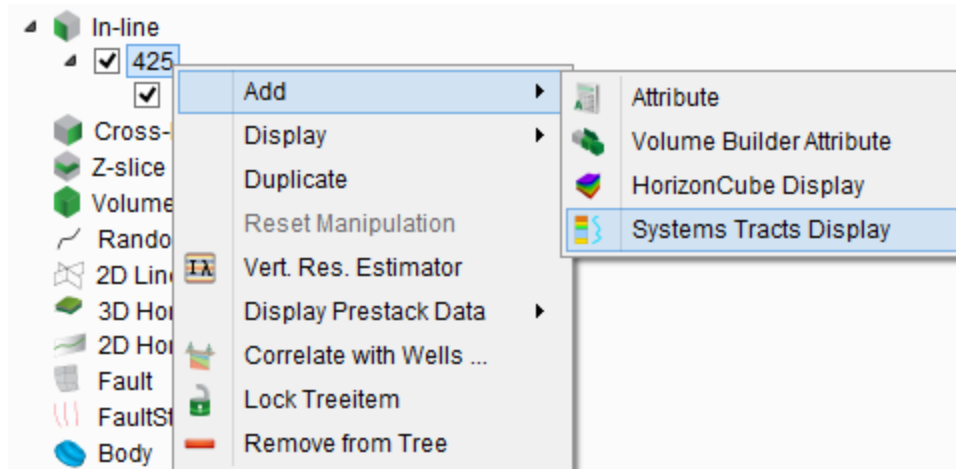
- **Rename:** It will rename the surface either at a user-defined manner or Auto-set. In the user-defined manner, the provide name would be used. If the name is set to Auto-set, it will automatically number the default surfaces according to the selected model.
- **Save as surfaces:** It is used to save the surface as a 2D horizon. In the pop-up window, the corresponding lines are listed on which the surface has been interpreted. The line selection must be made by the user in the *Select 2D lines* list. The output name / colour should also be provided.



- **Delete:** It will remove the selected boundary and would extend the lower interpreted systems tract up.
- **Hide surface names:** This is used to hide the surface label.

6.2.5 Display System Tracts

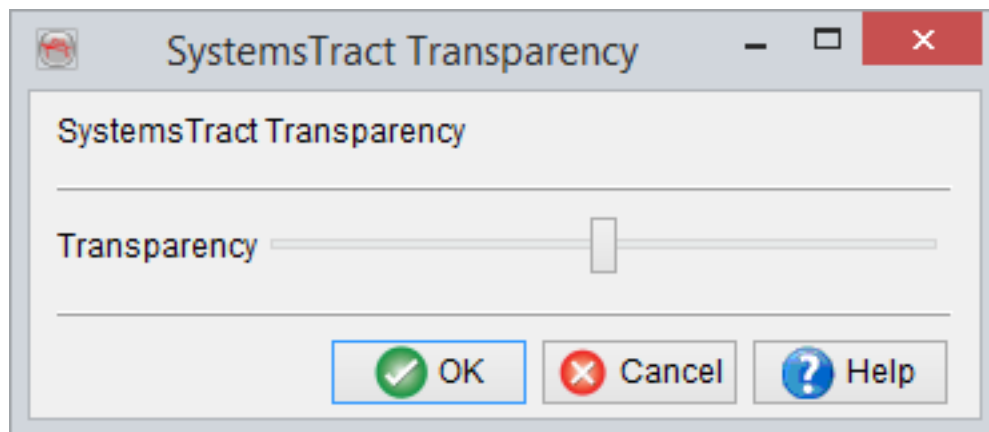
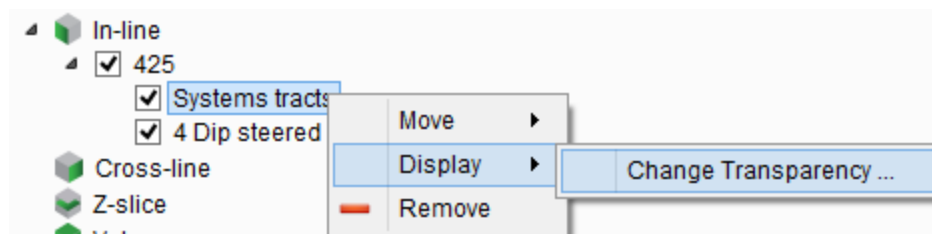
The interpreted systems tracts in the interpretation window can be displayed on the inline/crossline/2D line using a pop-up menu. Right-click on the element on which you want to display the systems tracts and select the *Add Systems Tracts display option*.



An example overlay of the interpreted systems tracts (semi-transparent) on an inline from a 3D HorizonCube.

Change Transparency

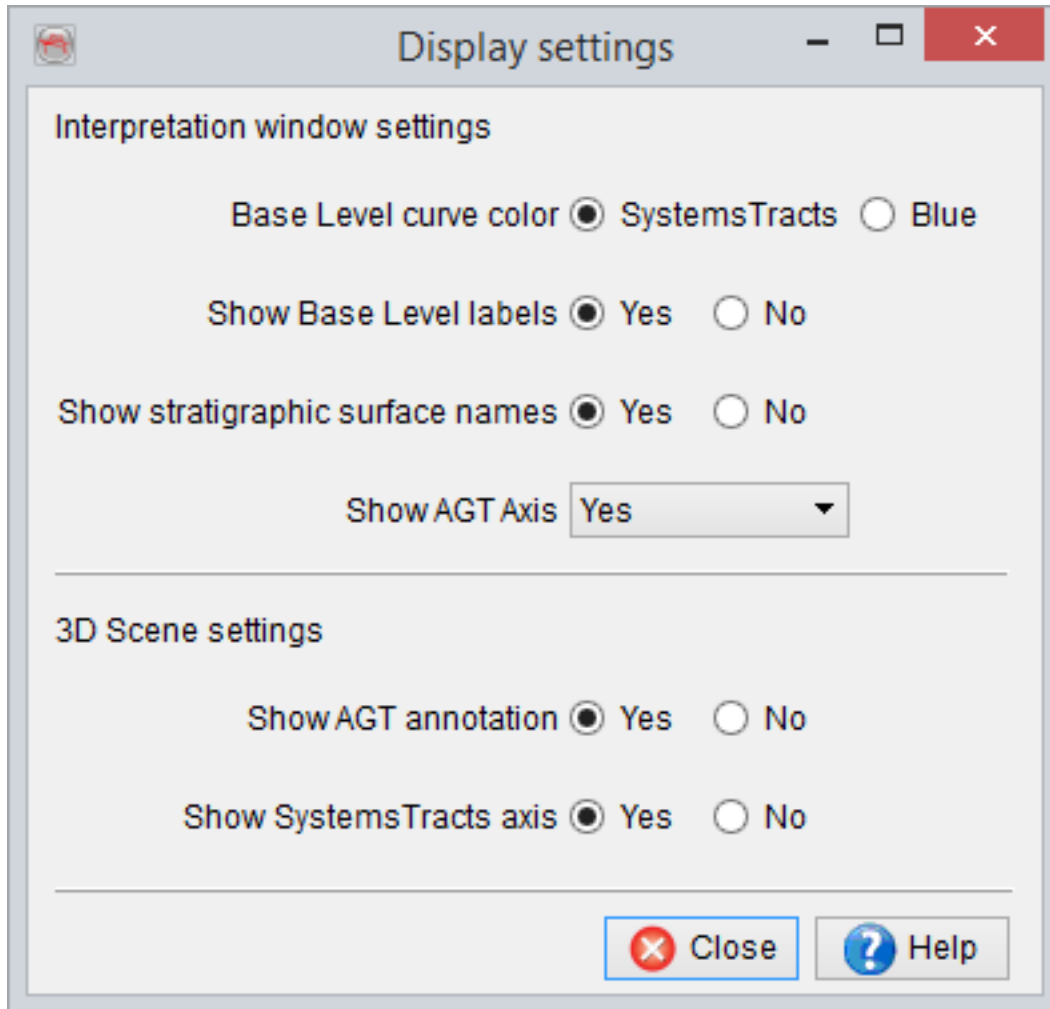
To change the transparency of the systems tracts overlay, right click on the systems tracts element from the tree (as shown below) and select the transparency-option from the pop-up menu. From the pop-up window (SystemTracts transparency), the transparency can be set by moving the slider towards right.



Moreover, the *Move* option in the pop-up menu for the System Tracts changes the order of the system tracts within the element. However, it doesn't change the display layout in the OpendTect scene. Select *Remove* item to remove the display of the System Tracts from the scene.

6.2.6 SSIS Display Settings

The display settings of the interpretation window are changed using this dialog.




Options

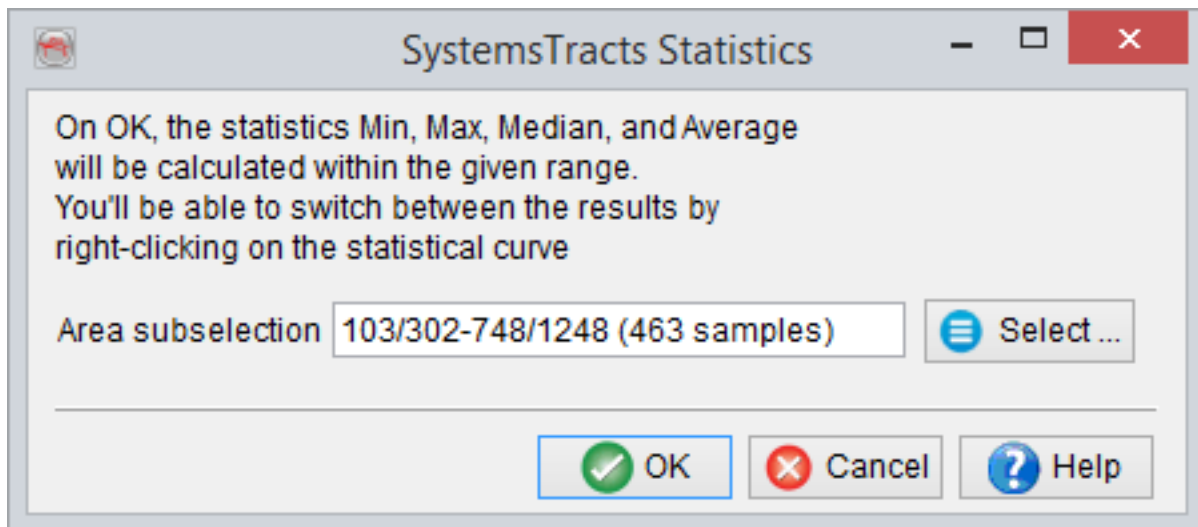
- *Base Level curve color* is by default set to the color of a systems tract. If a systems tract is blue, the color of the base level curve segment should be blue also. If the default is changed to Blue, the entire curve will appear as blue curve.

- *Show Base Level labels* allows to annotate the labels such as End of Regression, Onset of Base level fall, End of Transgression etc.
- *Show stratigraphic surfaces names* allows to annotate the name of sequence stratigraphic surfaces as they appear in the selected [Depositional Model](#).
- *Show AGT annotation* is an option that interacts with the 3D Scene and survey box. If this option is set to No, the AGT annotation will not appear in the scene.
- *Show Systems Tract axis* also interacts with the 3D Scene. If this option is set to No, the systems tracts' color codes will not appear in the scene.

6.2.7 Create Statistical Curves

Statistical curves are the thickness based curves that are defined based on various statistics (Min, Max, Median, average) for each systems tract . These curves are valuable to graphically illustrate the thickness variations adjacent to base level curve. Note that the base level curve is mostly a conceptual illustration that is not derived from the data. It is this thickness curve, which is an add-on to the base level curve.

The statistical curves are computed by launching the *Systems Tracts Statistics Dialog* using the icon  .




It is recommended to use more than one trace location, or an area defining the depocenters.

After proceeding, the curve is displayed on the right-most side of the base level curve. The default curve is average, which can be changed by right clicking on the newly displayed curve and selecting various statistics only the fly.



At present, these curves are not stored in the interpretation window and are temporarily displayed while the dialog is open.

6.3 HorizonCube Slider

The HorizonCube slider  is launched from the [SSIS toolbar](#), which is used to visualize the displayed HorizonCube and to understand the depositional patterns/geometries. The slider in the SSIS interpretation is of vital importance as it is being used in the SSIS interpretation module.



Link to ColorTable: If this is checked, the active HorizonCube colour bar will be dynamically stretching/squeezing relative to the slider position.

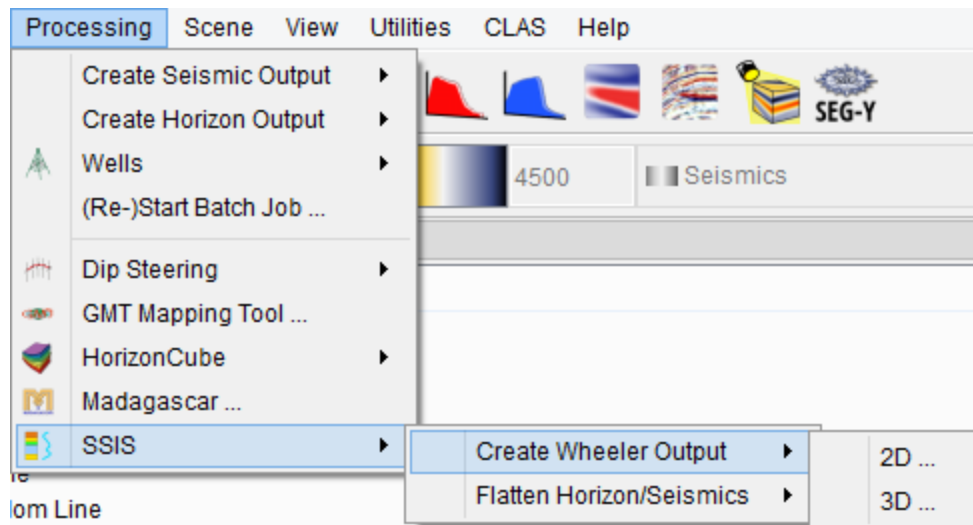
Lock Top/Bottom distance: It is used to display fixed number of events. For instance, if you want to visualize one event at a time, you may set the slider gap of zero i.e. the top and bottom slider positions are reflecting the same event. Thereafter check this option and move either the top or bottom slider.

6.4 Wheeler Transform, Wheeler Scene

See following chapters below.

6.4.1 Wheeler Scene

The Wheeler Scene in OpendTect is another scene, which is added from the SSIS menu. It may be noted that the Wheeler Scene is a transformation (flatenning) of HorizonCube into relative geological time (RGT). Therefore, before adding a Wheeler Scene, the HorizonCube should be selected.

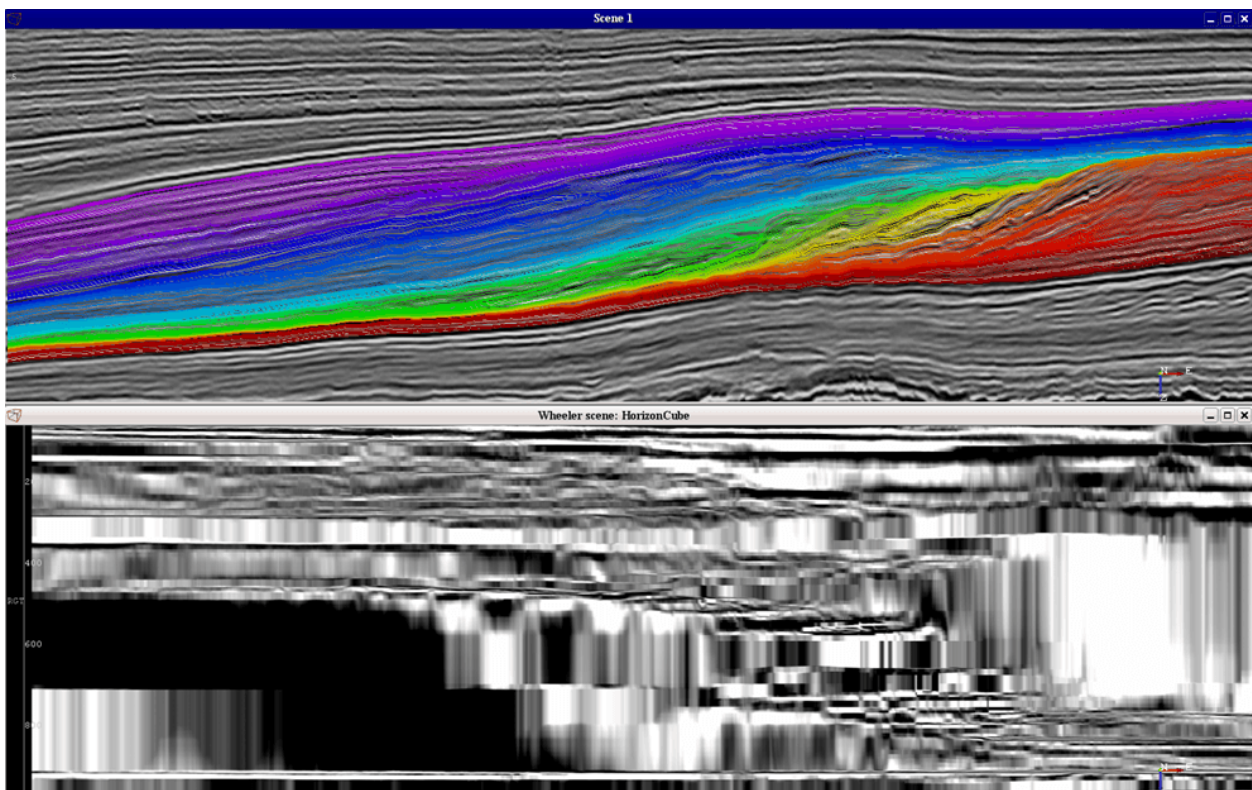


In the Wheeler scene, flattened seismic data can be displayed by adding elements (inline, crossline, and timeslice) in the tree. By displaying flattened seismic data, one can assess the quality of the selected HorizonCube. Making a good HorizonCube is an iterative process. It will probably take several runs before the result is satisfactory. In order to improve HorizonCube the user can try to:

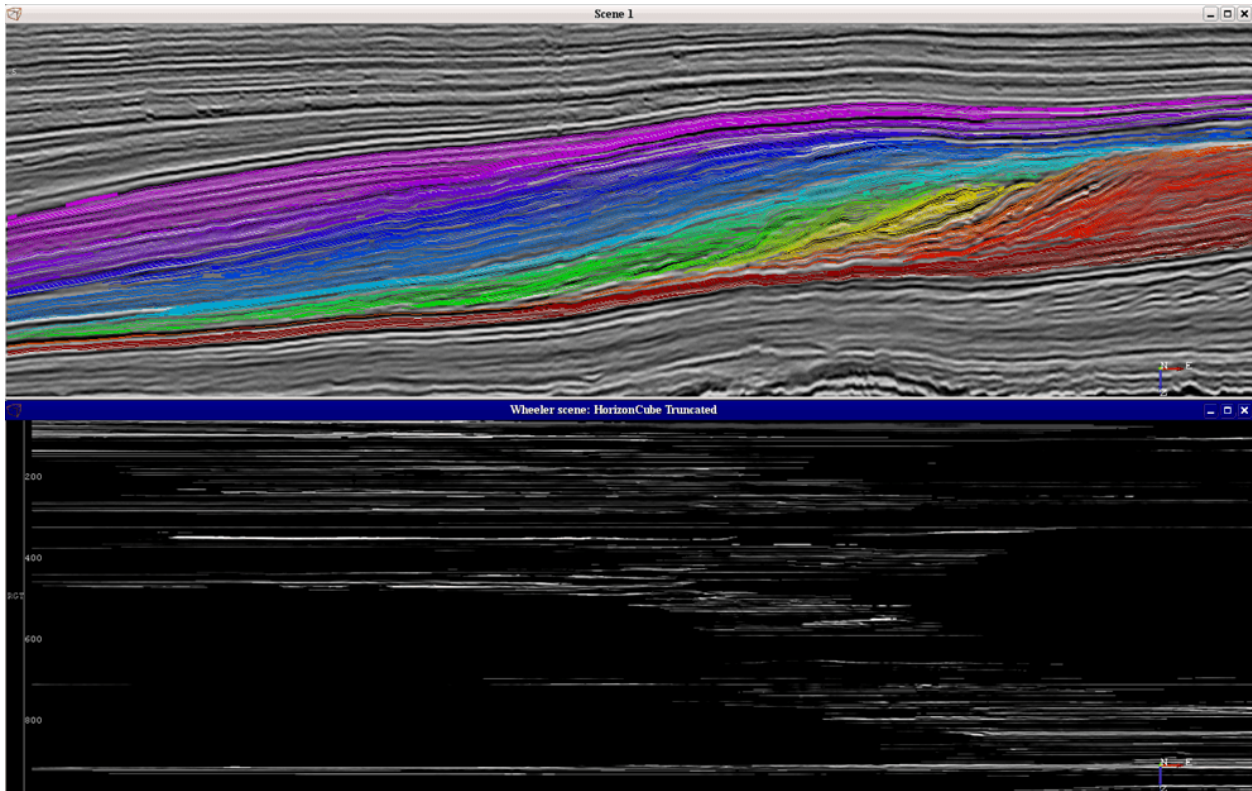
- *map additional horizons (sequence boundary)*
- *fill holes in mapped horizons*
- *make horizons continuous and watertight*
- *filter the SteeringCube (data-driven approach)*

In the Wheeler scene, stored data is transformed on-the-fly to the flattened domain. All attributes and neural network outputs are calculated in the normal domain first and then transformed to the Wheeler domain. Because of this transformation, quickly scanning through your Wheeler is only possible after [creating a Wheeler cube](#). On-the-fly calculations are not supported for the volume viewer, which only accepts input from pre-calculated Wheeler cubes.

Below, the two types of Wheeler transformation are displayed. One with [Continuous HorizonCube](#) and the other is the same HorizonCube with a [density filter](#). Note that for Wheeler Transformation, the closely spaced events in a continuous HorizonCube needs to be filtered out so that the hiatus, and stratal terminations become prominent and one can interpret base-level changes.

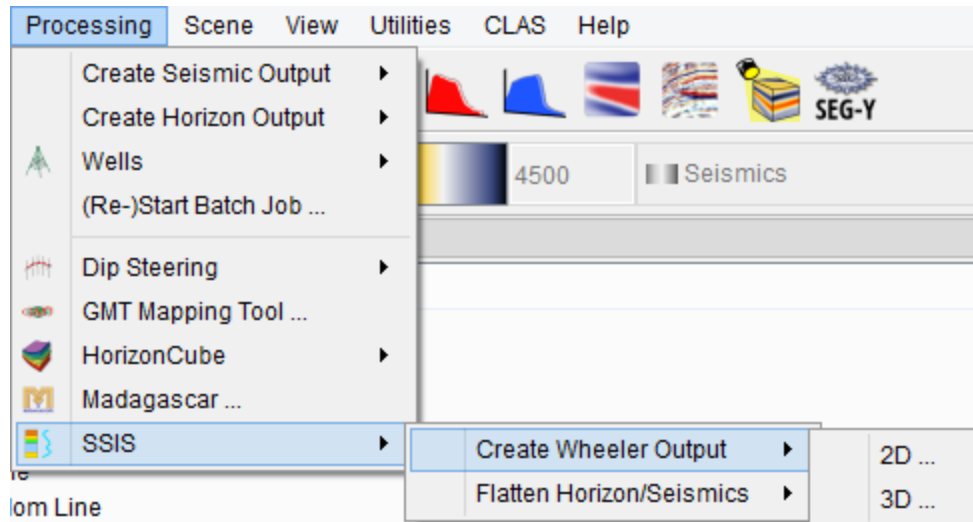


Top scene is a display of a seismic data and the HorizonCube with continuous events. Bottom display is the Wheeler Transformed Scene of the HorizonCube. The same seismic data is displayed in the wheeler domain.



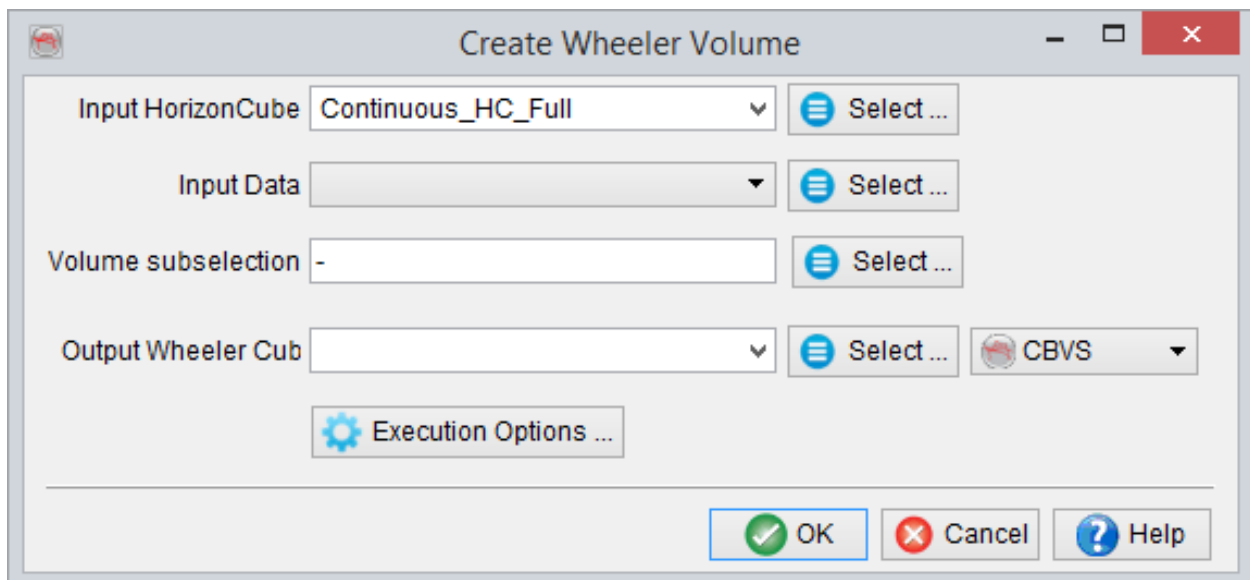
Top scene is a display of a seismic data and the continuous HorizonCube with a density filter applied. Bottom display is the Wheeler Transformed Scene of the displayed HorizonCube. The same seismic data is displayed in the wheeler domain.

6.4.2 Create Wheeler Output (2D, 3D)



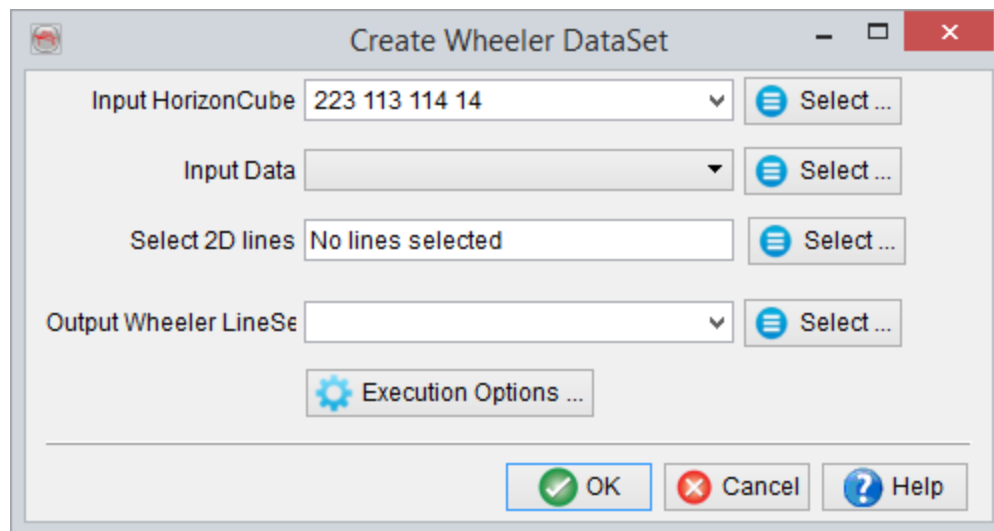
A WheelerCube is a seismic volume in the Wheeler domain. When a WheelerCube is displayed on an element in the Wheeler scene, no transformation is needed as the Wheeler volume is created on-the-fly.

 *Before creating a WheelerCube a HorizonCube must be selected.*



Create Wheeler output volume for a 3D volume

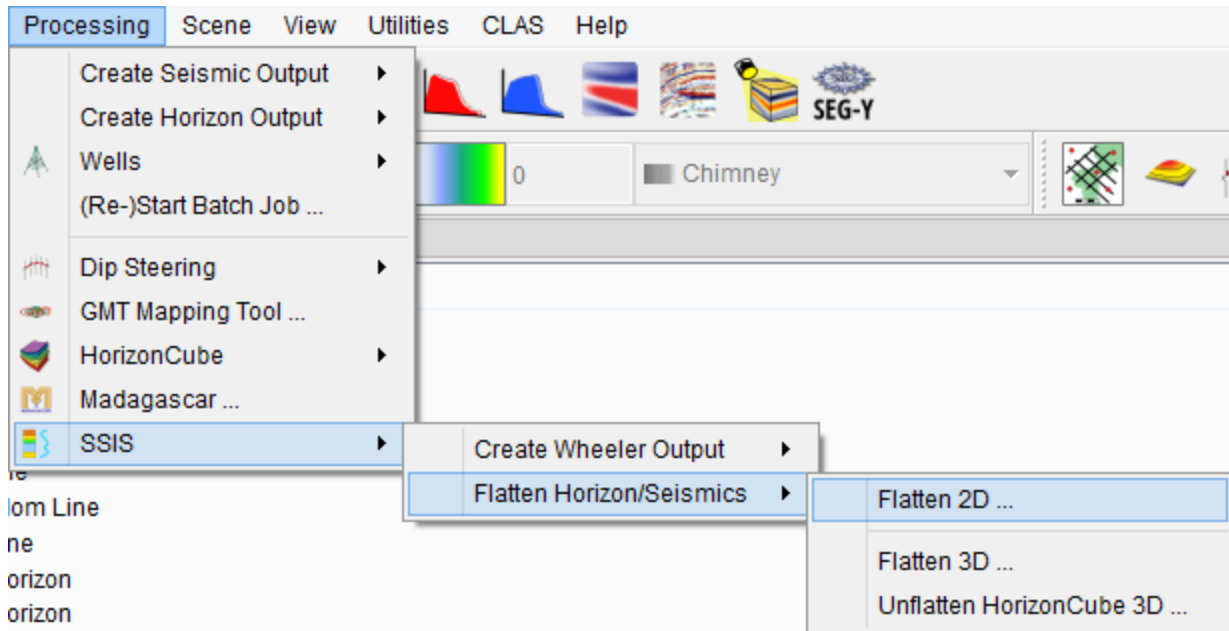
Any stored 3D-data (seismic, attribute, neural network output) can be converted into the Wheeler domain. Select the stored volume you want to transform and specify a name for the WheelerCube. Moreover, the wheeler volume can be reduced by specifying the [area-sub-selection](#). The Wheeler Cube is created in batch-mode (see corresponding section in the OpendTect User Documentation). A stored WheelerCube can be displayed by right-clicking an element in the tree, then click select *attribute* > *Wheeler Cubes*.



Create Wheeler output volume for a 2D Lineset/Line

For a 2D type of Wheeler transformation, 2D HorizonCube is required. Select the 2D HorizonCube in the input. This will be used as a transformation matrix for the Wheeler transformation. Next, provide the input 2D data e.g. seismic/AI data. Optionally, you may create it for one line only and also restrict the output to a limited number of traces (*Traces sub-selection*). After this, the Wheeler output is stored to the written line-set and the attribute (format: Lineset_Name|Output_Wheeler_Attribute_Name).

6.5 Flatten Horizon, Seismics



The flattening and un-flattening workflow can be used for SSIS interpretations specifically in structurally distorted regions. The workflow starts with flattening a volume/line using a reference horizon. The selected seismic data and horizons are flattened into a new survey. In this flattened survey, the SteeringCube can be prepared, followed by processing of the HorizonCube. After the HorizonCube in the flattened survey is satisfactory, it can be un-flattened in the parent (un-flattened) survey for further SSIS interpretation.

6.5.1 Flatten

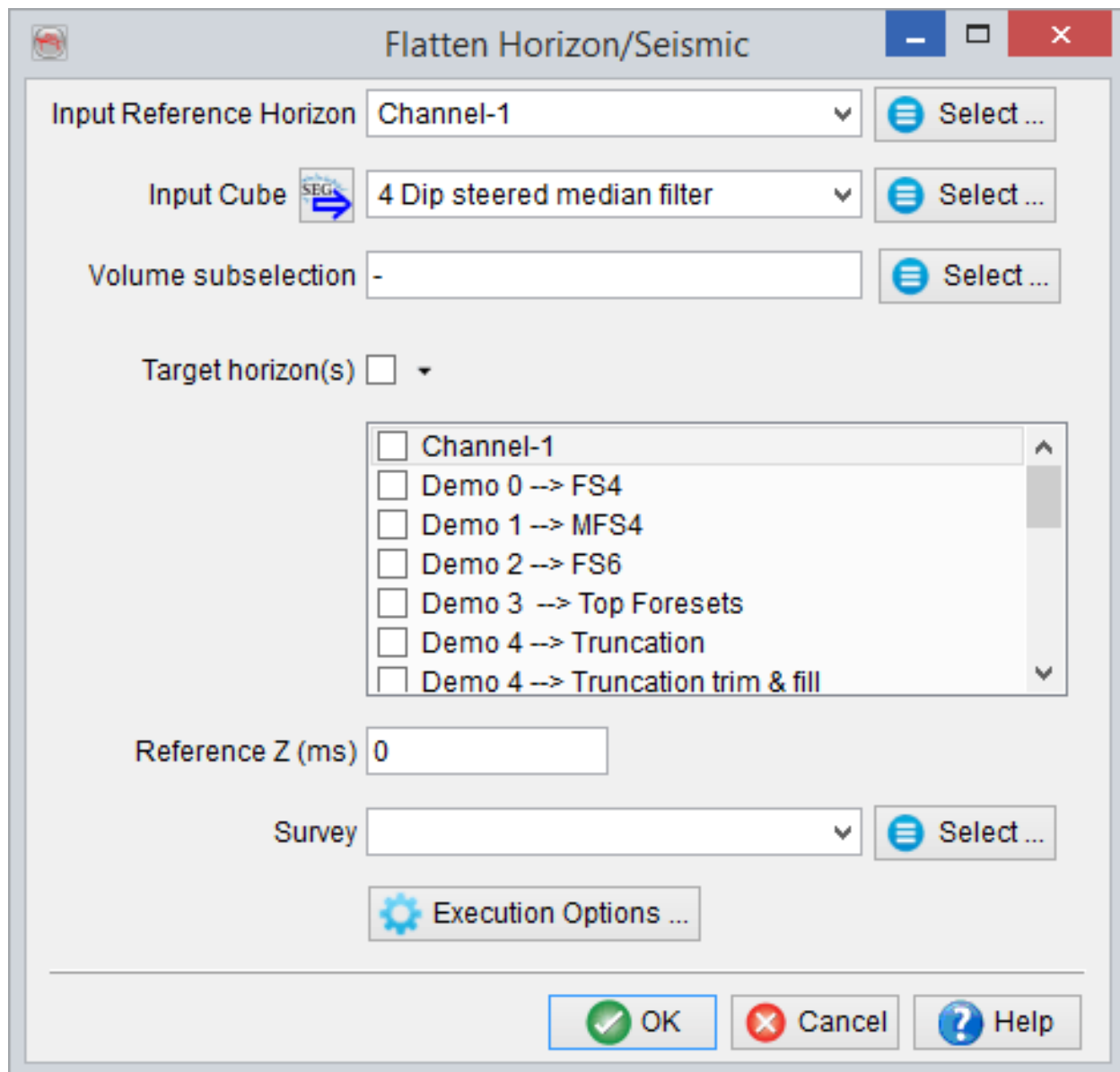
3D Survey

To flatten a 3D survey using a reference horizon, this window is used. It can be launched from the SSIS menu. The input *Reference Horizon* is selected from the active survey, which would be used as a datum for the flattened survey. This datum is a *Reference Z(ms)* value. The corresponding seismic volume is selected in the *Input Cube* field. This volume would be used to prepare SteeringCube (for HorizonCube) in a flattened survey. Optionally, volume sub-selection can also be made to do work in a sub-volume.

The *Target Horizon(s)* are selected as these horizons are used as input in the HorizonCube, for building the framework. In the list one or more horizons can be transformed in the selected output i.e. a flattened survey (give a name in the *Select Survey* field).

Mode: Flattening / un-flattening mode. If flattening is set here, the selected data from the active survey will be flattened in a new (flattened) survey. The un-flattening mode is used to un-flattened a flattened (active) survey.

After providing the right selection(s), the user may proceed further by clicking on the *Proceed* button. This will start a batch program. Optionally, the user may run the batch program on a remote host by checking the *Show Option(s)* box before pressing the *Proceed* button.



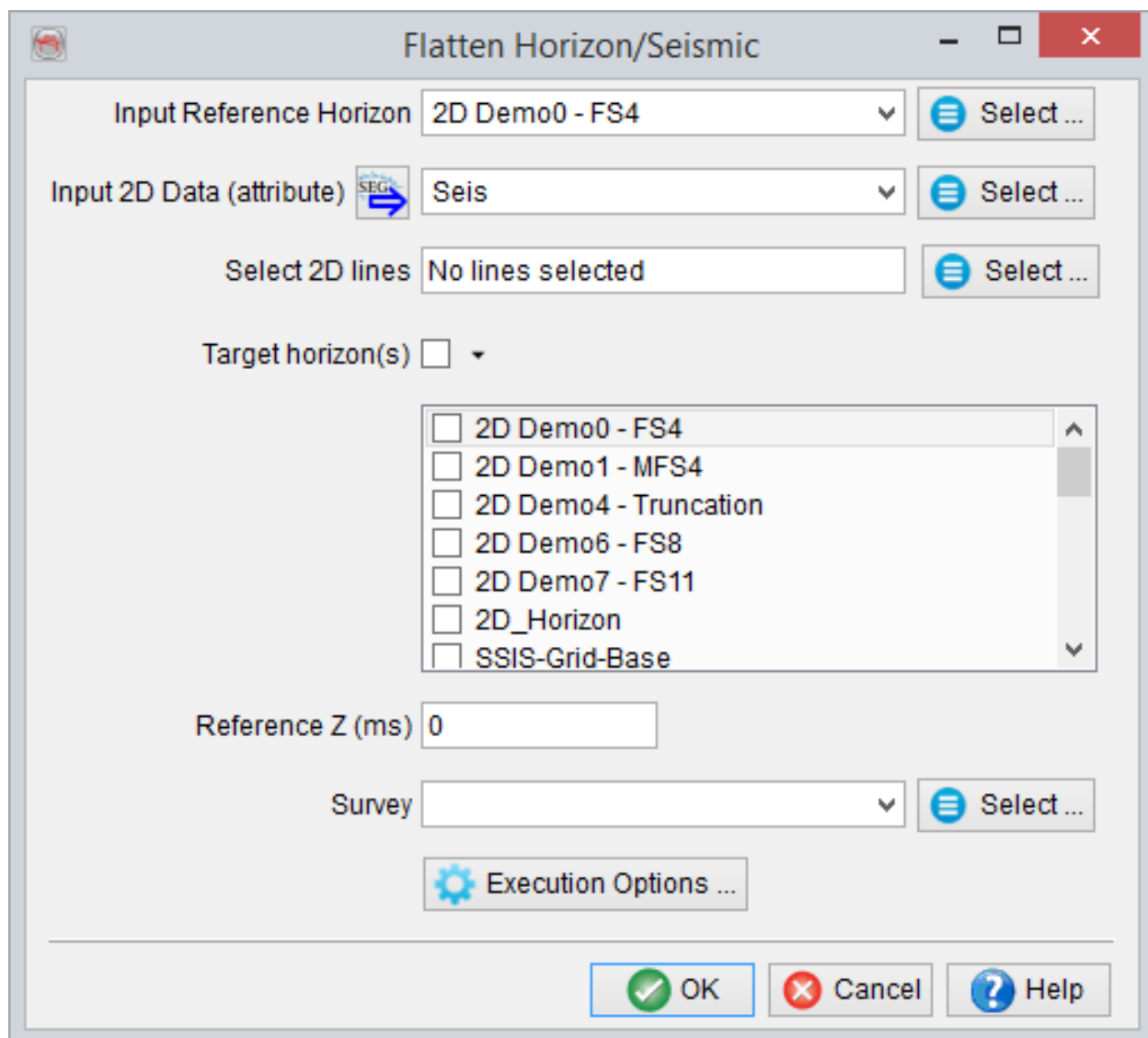
Flattening a 3D seismic data and the horizons to prepare a HorizonCube in a flattened survey.

2D Survey

The same workflow can be applied for a 2D survey. If the window is launched in an active 2D survey, the 2D lines and the corresponding 2D horizons can be flattened using a

reference horizon. The reference horizon defines a flattening datum with a given *Reference Z value*. This is selected from the *Input Reference Horizon* field. The 2D lines available in a *Lineset* are selected for the *Lineset/Line Name* option by pressing the select button.

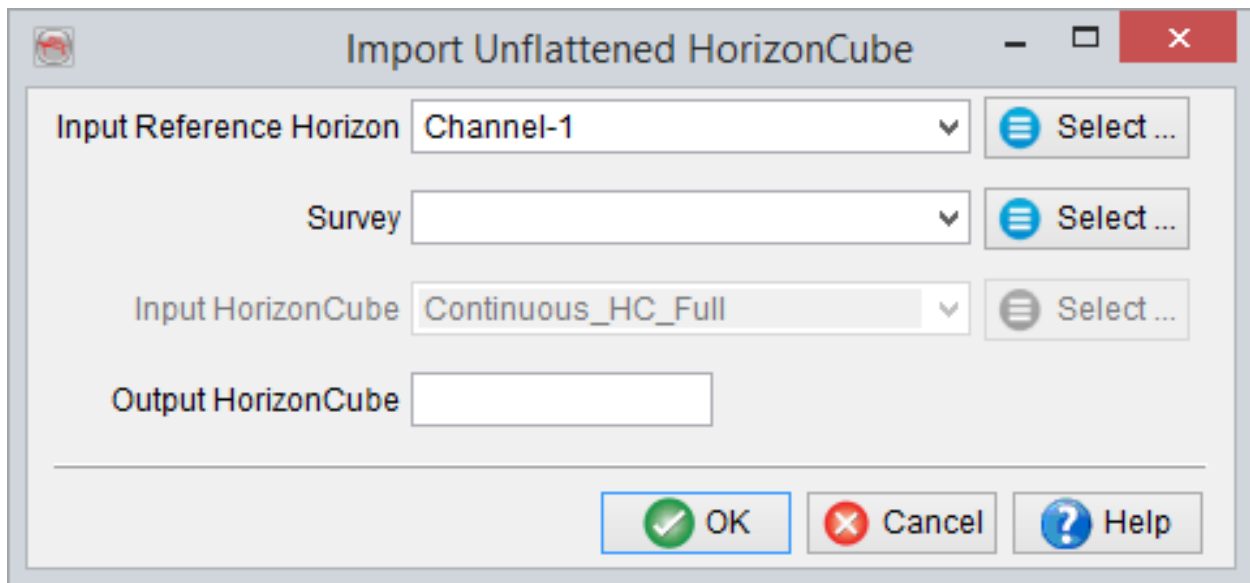
Together with the seismic data, the corresponding 2D horizons from the active survey may also require flattening. These horizons are selected from the *Target Horizon(s)* field. Finally, select the flattening mode, and give an appropriate output survey name and press the *Proceed* button.



Flattening 2D seismic data and the horizons for preparing a HorizonCube in a flattened survey.

6.5.2 Unflatten HorizonCube

The HorizonCube calculated in a flattened survey can be transformed back to its parent (un-flattened) survey. A input reference-horizon for the 2D/3D case needs to be provided. The horizon should be the same as the one selected to create a flattened survey. The input flattened 2D/3D survey, in which the HorizonCube is prepared, is selected from the *Select Survey* field. The input flattened HorizonCube available in the selected survey is selected in the *Input HorizonCube* field. Finally the output name for the HorizonCube is given. If the 'OK' button is pressed, the batch program will start to transform the flattened HorizonCube (2D/3D) into a un-flattened HorizonCube in the active survey.



Import Unflattened HorizonCube

Input Reference Horizon Channel-1 Select ...

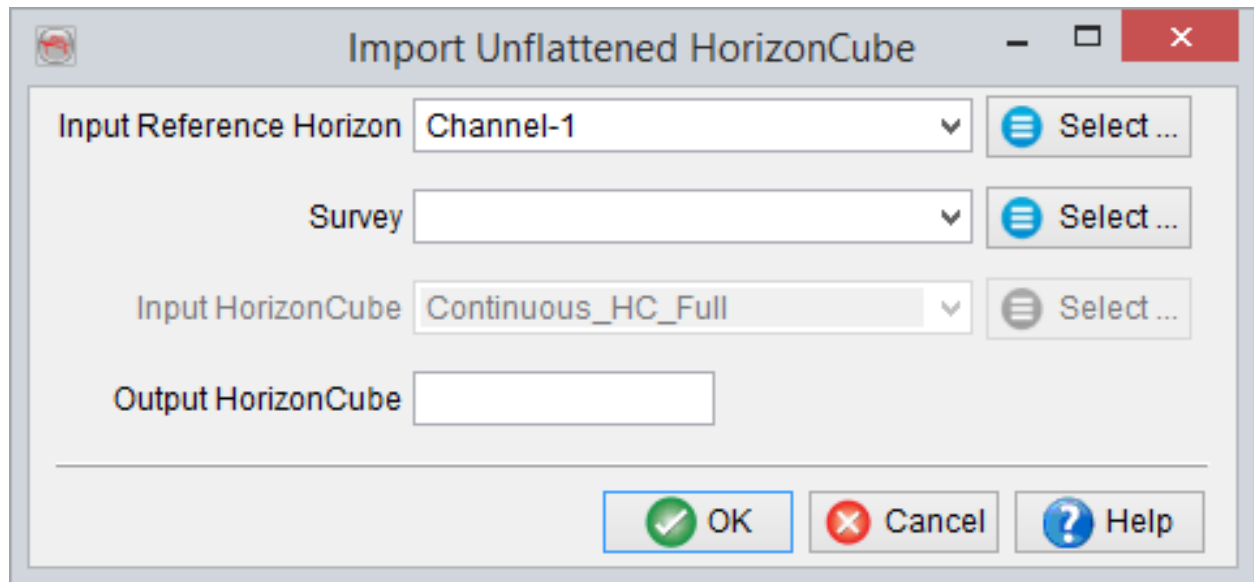
Survey Select ...

Input HorizonCube Continuous_HC_Full Select ...

Output HorizonCube

OK Cancel Help

Un-flatten a HorizonCube from a 3D flattened survey.



Un-flatten a HorizonCube from a 2D flattened survey.

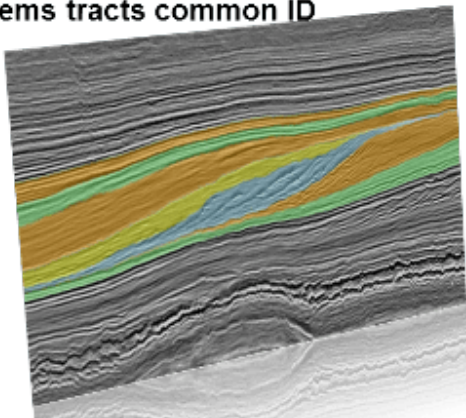
6.6 System Tracts Attributes

A systems tract attribute defines a volume that contains the sequence stratigraphic interpretation. Once an interpreter has completed 3D SSIS interpretation, the interpreted systems tracts can be stored as an attribute volume. This is normally done using this attribute. Once this attribute is defined using the Attribute set window, one may either store it as an output volume or can visualize it in 3D. This attribute can also be used for other purposes e.g. reserve estimation, prediction of seismic facies bounded by the systems tracts etc.

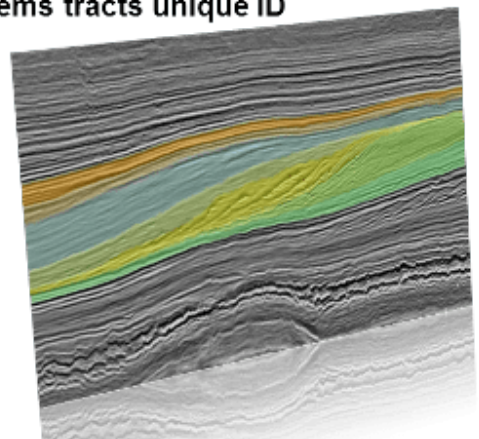
This attribute returns three outputs:

1. **Common ID:** It is a common ID for a systems tract within a volume. For instance, if there are four HSTs interpreted, all four HSTs will have same common ID (i.e. 0). Such an attribute is chiefly useful for 3D visualizations of a systems tract with a common colour. However, for seismic prediction, it will have little usage.
2. **Unique ID:** It returns a unique number to each systems tract. Such an attribute is a valuable tool to do seismic prediction using [Neural Networks](#).

Systems tracts common ID

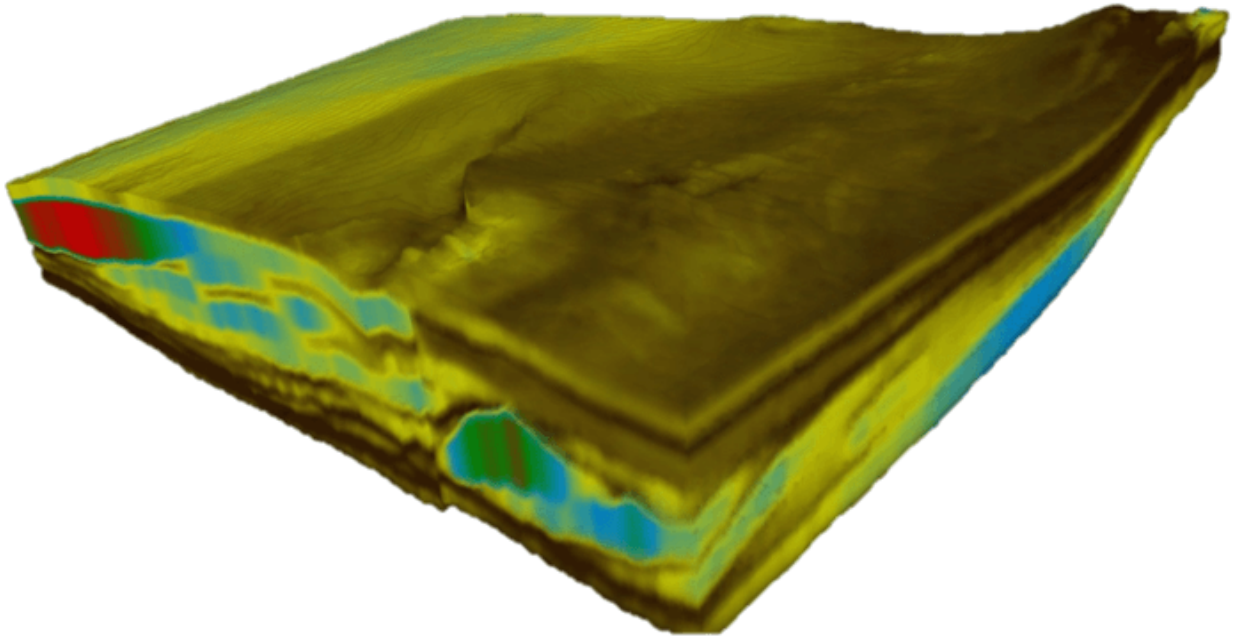


Systems tracts unique ID

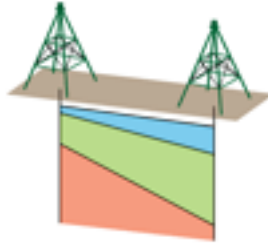


An example of systems tracts sub-attribute.

3. **Isochron:** It is an isochron between the top and bottom of a systems tract. It can be used to interpret the preserved thickness within a systems tract. It can also be calibrated with absolute geologic time to interpret the actual sedimentation rate within a systems tract.

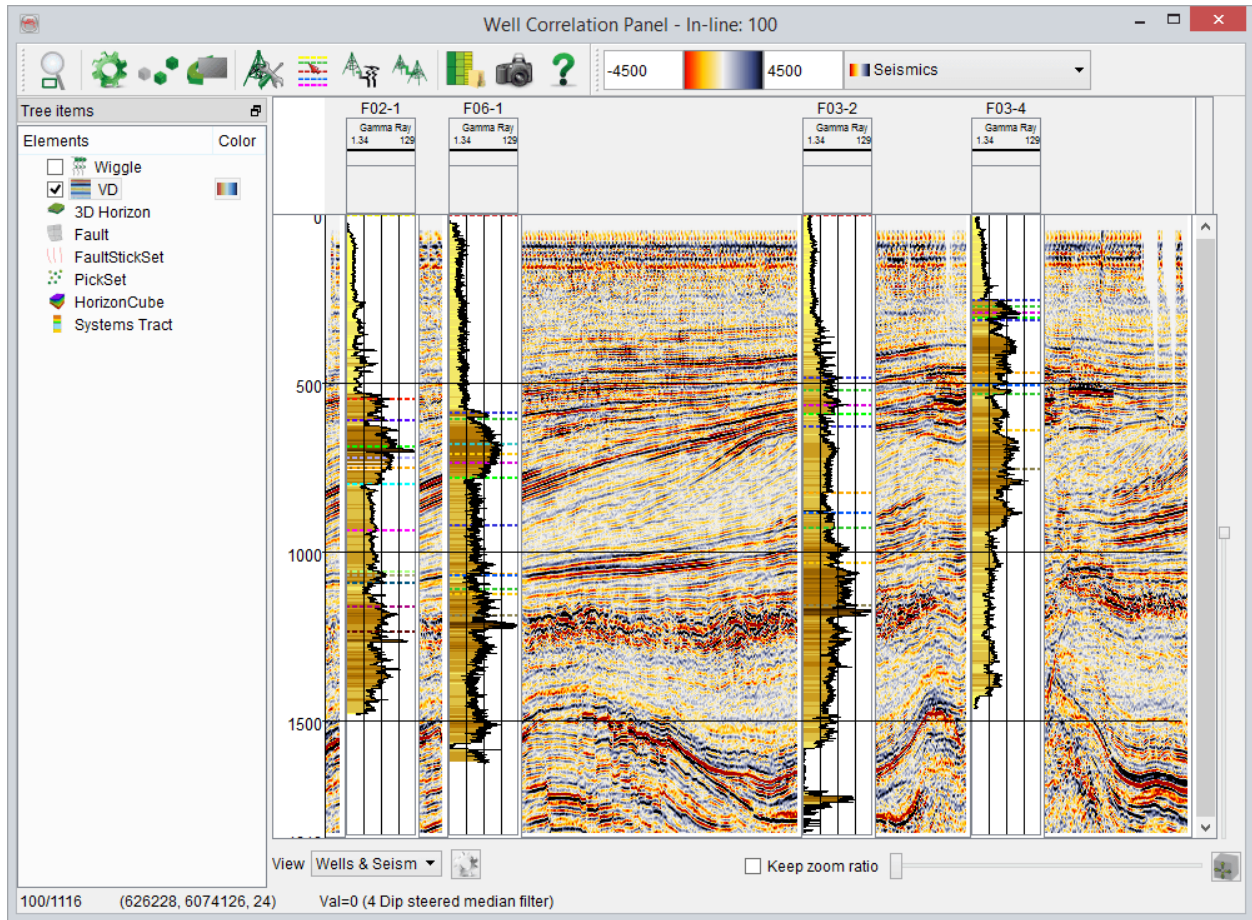


7 Well Correlation Panel



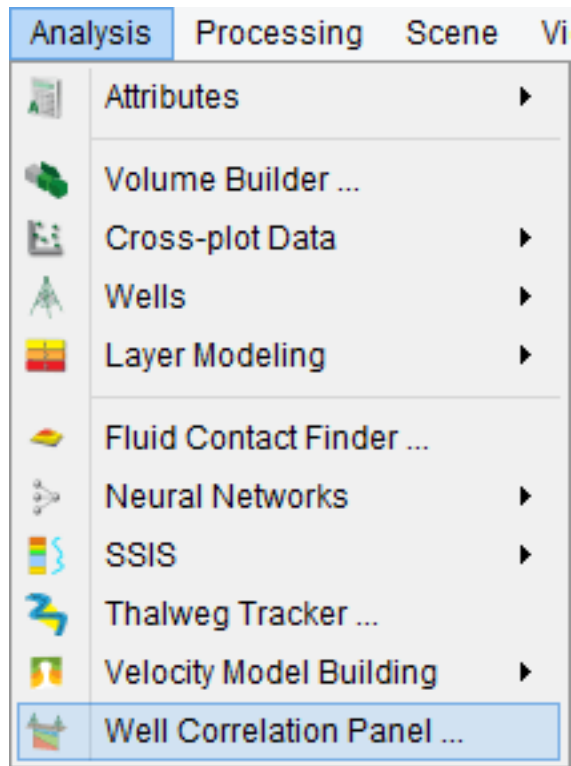
7.1 Introduction


The Well Correlation Panel (WCP) is used to create multi-well correlation(s) sections using the interactive overlays of the HorizonCube, seismic data and SSIS Interpretation (systems tracts). The aim of using WCP in OpendTect is to build a sequence stratigraphic framework by connecting wells using the HorizonCube. This is done easily by picking new stratigraphic markers while visualizing densified Horizons. Additional options are available, such as editing the well marker database by moving the marker vertical position using the mouse. Moreover, the interpreter can also create a transect (2D Line) from a given 3D seismic data by connecting several wells. The following section explains these options in detail.

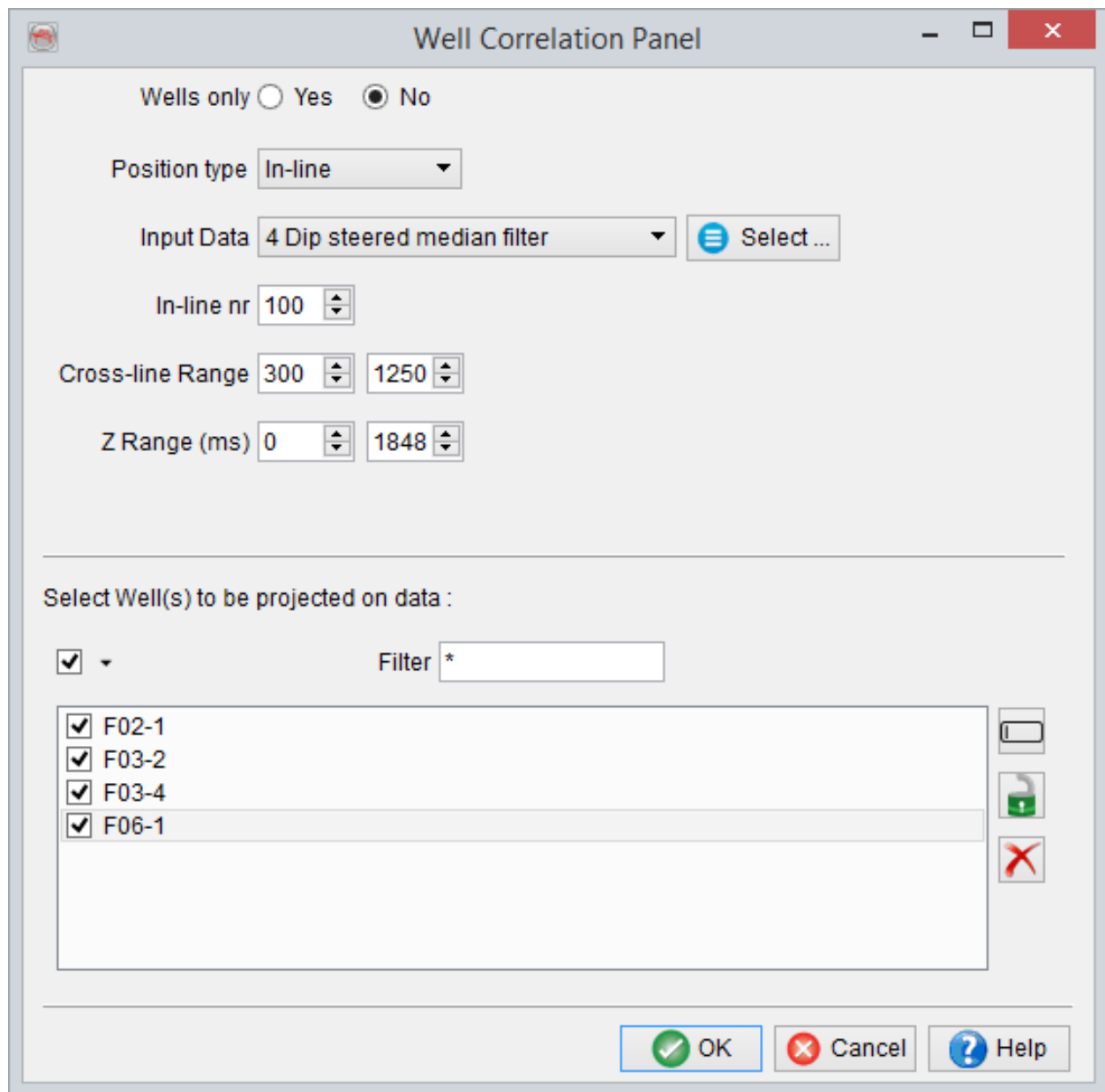


7.2 WCP Main Window

The 'Well Correlation Panel' is launched from the Analysis menu:



The application can also be launched in the OpenTect main menu (Analysis > Well Correlation Panel) or by clicking the 'WCP' icon  in the main toolbar. In the following window, input data-type is needed, e.g. Inline/Crossline or a 2D line.



Select Data Type:

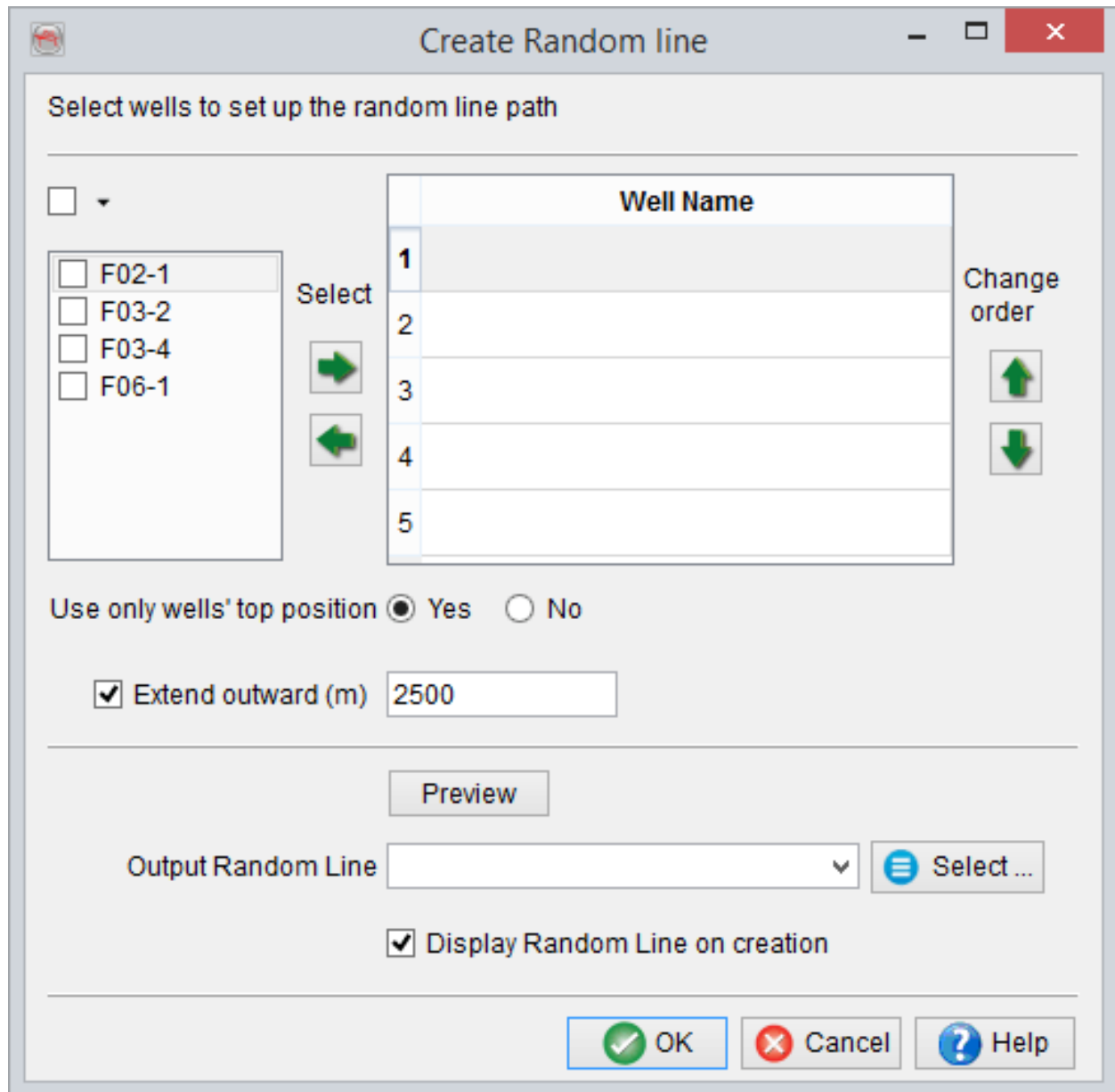
- 2D Line: Correlates a 2D line close to selected wells. A sub-selection can be made for the line-set and the corresponding line. Furthermore, a trace sub-selection can also be made if the line is too big and the wells lie within the defined trace range.

- Inline/Crossline Used to select a 3D seismic data (inline/cross-line) close to the selected well.

Selection of key wells:

- The selected wells are projected onto selected data (2D/3D line). One or several wells can be selected in the well list. To select all, please use left mouse button (select and drag down the list). Optionally, use CTRL key plus left mouse click(s).

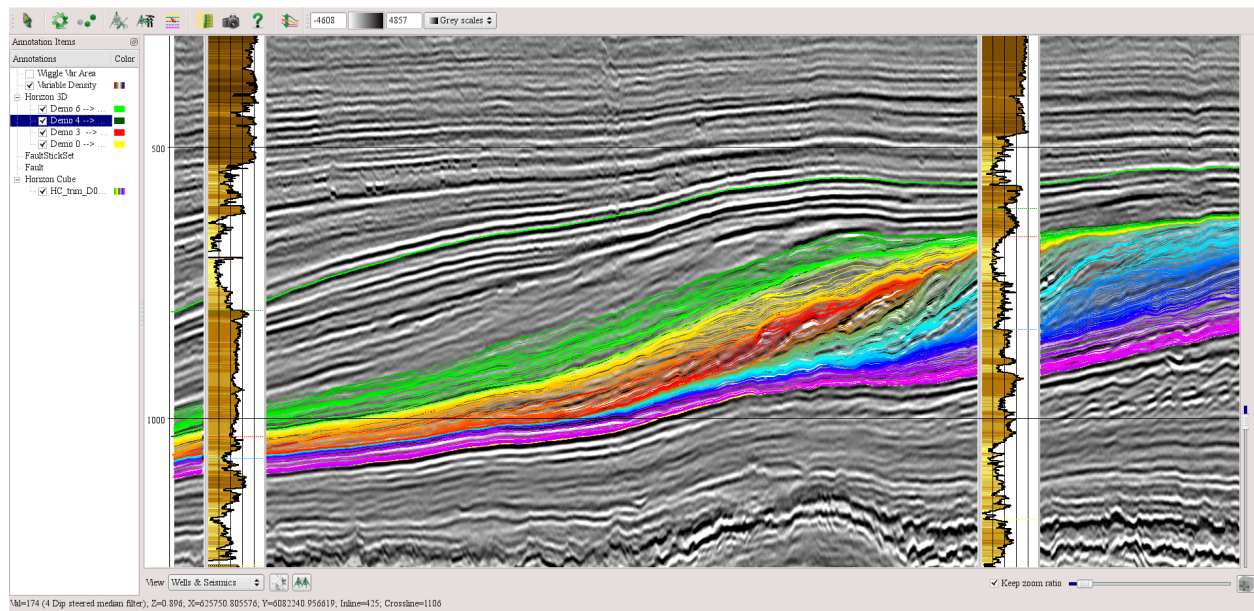
Create a 2D line (from 3D) between wells



A 2D line created from 3D seismic data can be created directly in the WCP input data selection window. To define a random line geometry connecting the wells, selected the wells from the left panel using the arrow(s). The table is filled and ordered according to the selection made. However, the order of the wells can be changed using the top/bottom arrows available for the *Change Order* option. Optionally, the geometry can be

visualized in a 3D Scene by pressing the *Preview* button. The geometry of the random line (*Start at*) can either be defined by the top position of a well or by a bottom position of the well (deviated wells). The *extend outward* field is by default set to 2500m such that the line is extend 2500m away from the first and the last well positions.








Once the geometry of the random line is defined, select the 3D seismic volume and provide a name to this line and corresponding line-set. This stores the geometry of random line as a 2D line, which can be loaded later. Now, this stored 2D line can be used as an input seismic data type for the WCP.










WCP Toolbar Icons

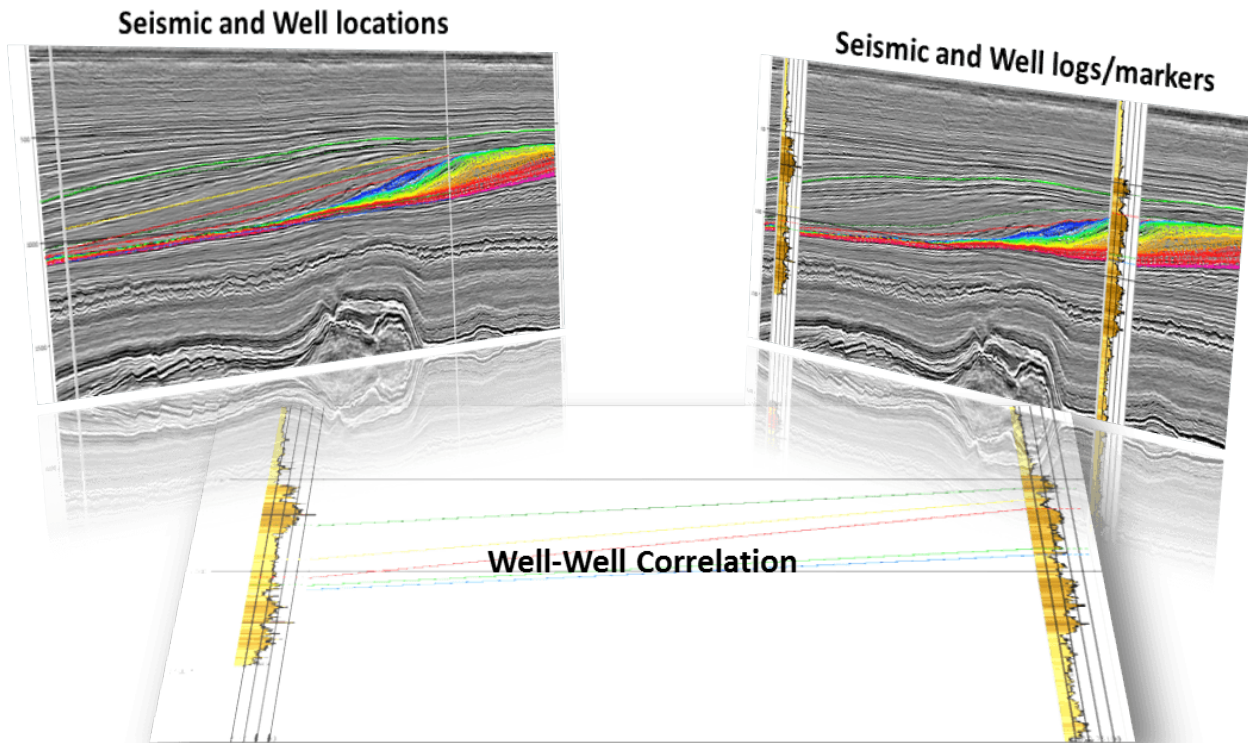


The top toolbar for the well correlation panel.

-  Pan the display and pick horizons/markers on the seismics/well.
-  Set the color properties (change/clip the colour and its range) of the seismic data.
-  *Seed Mode*: Start 3D/2D horizon picking on the displayed seismic data.
-  Set the well properties, e.g. select logs for a well, fill logs with colours, display stratigraphic column, or add more log panels.
-  Launch the seismic-well tie module of OpendTect. Please press F1 key and browse to section 6.4 for further details.
-  *Pick Markers*: Pick new/modify well markers in the displayed wells.
-  *Manage Stratigraphy* Launch the stratigraphy management window of OpendTect.

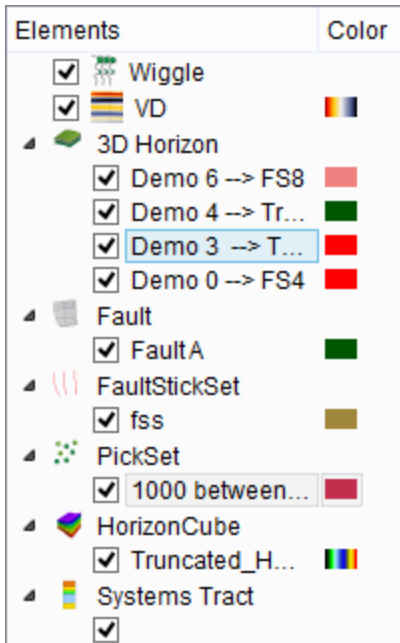
-  *Snapshot tool*: Grab an image of the correlation panel.
-  Help documentation on Well Correlation Panel.
-  *HorizonCube Slider*: Active when the HorizonCube has already been displayed in the panel.
-  Toggle WCP view e.g. Display wells only, seismic only seismic and well together.
-  Create various displays: wells on top of seismic, wells as separate panel, equidistant wells correlation.
-  Correlate markers by drawing a marker connection.
-  Re-sets the zoom of the display to a default preview i.e. entire seismic transect is shown.
- *Keep zoom ratio* Associated with the zoom (in/out) sliders. It is checked by default to keep the vertical and the horizontal ratio equal while either slider (vertical/horizontal) is moved to zoom/unzoom the display.

7.3 Correlation Displays and Settings



Several displays for WCP

WCP supports several displays for correlations; Equidistant display, in which the distance between the selected wells is set to an equal distance or normal distance display, in which the real well-to-well distance is scaled and is displayed in the correlation panel. Moreover, the well data can also be overlain on top of seismic or with a gap between seismic. Any wells can also be displayed without using seismic. These display settings can be accessed from the `WCP` and from the tree.



Variable Density: Right-click to display seismic data.

Horizon3D/2D: Display a 3D horizon if an inline is displayed in the panel or a 2D horizon if a 2D line is displayed in the panel. Additionally, start interpreting a new 2D/3D horizon.

FaultStickSet: Display a fault-stick set or interpret a new fault stick set.


Fault: Display/Interpret a 3D fault.

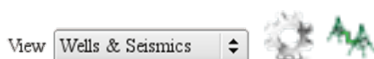
HorizonCube: Add a HorizonCube display.

Systems Tracts: Add an overlay of SSIS interpretation i.e. systems tracts.

Correlation Views

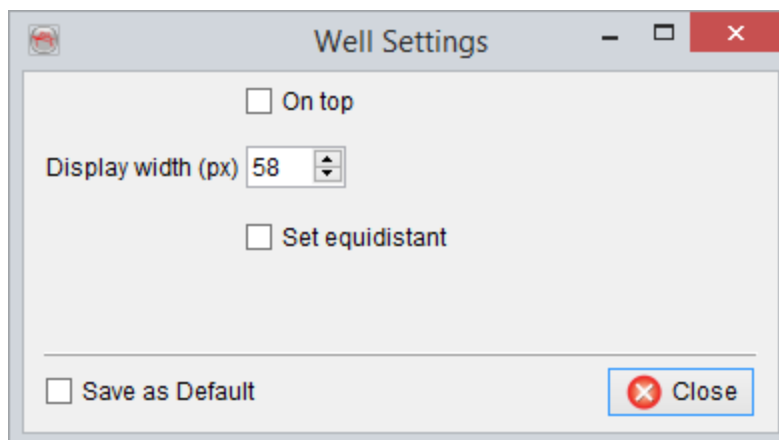
The toolbar, available at the bottom of a well correlation panel, is used to prepare different type of correlation views e.g. well-well only without seismic, well-seismic correlations without displaying logs and the well-seismic correlations using well logs and seismic. The combo-box (shown below) is used for setting these properties.


The well settings  are used to display a well panel either as a gap between seismic transect or on top of the seismic transect. By default, the well panel is displayed as a separate panel connecting the seismic transect. Once the *On top* box is checked, the well logs panel is displayed on top of the seismic data. Additionally, the top log information header may be toggled off/on.

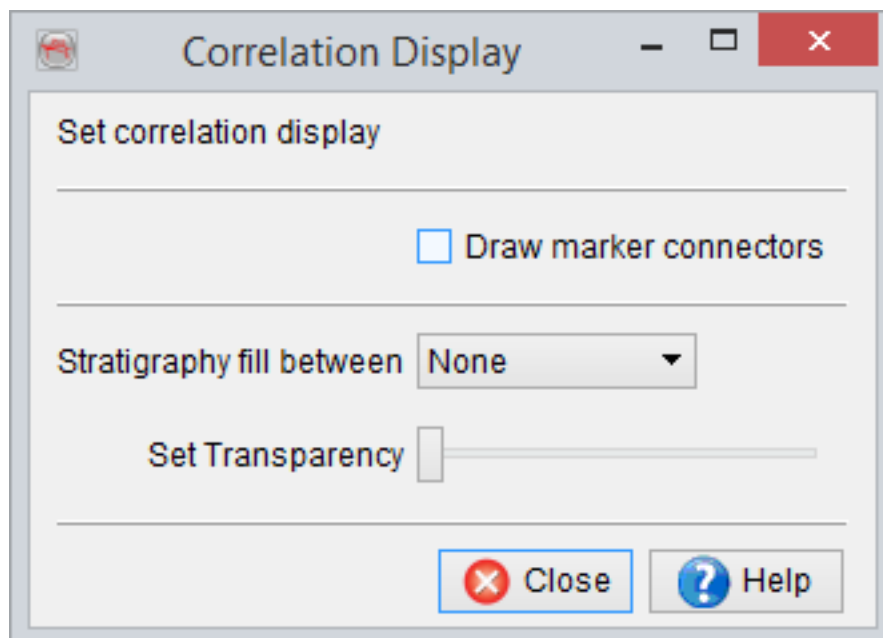


The width (pixels) is used to control the well panel width. The minimum range is 35pixels.


Set an *Equidistant Correlation View* check the *Set Equidistant* box.

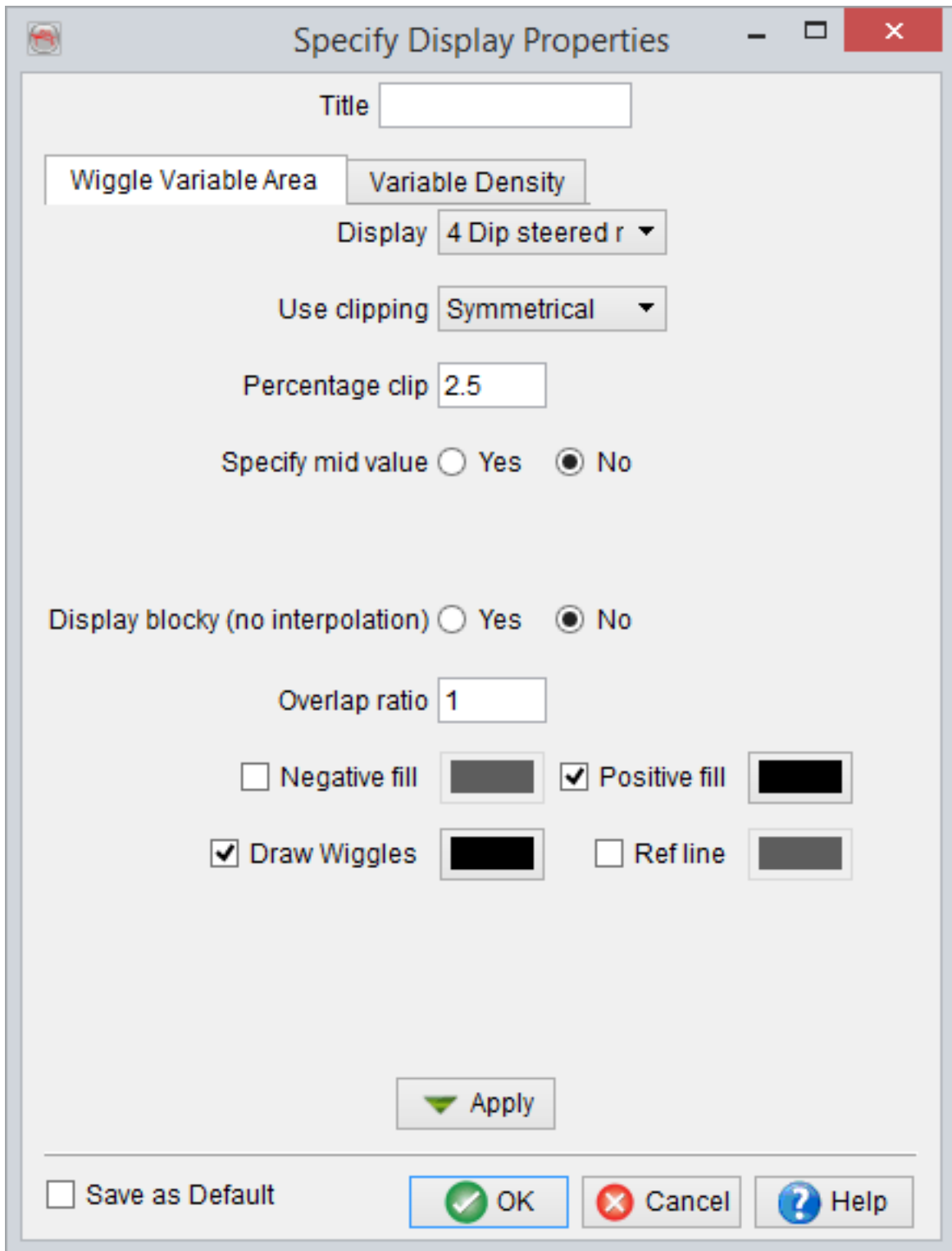


Draw Marker Correlation(s) by pressing this icon . In the pop-up window i.e. 'Correlation Display', the *Draw marker connectors* box should be checked in order to draw a straight line marker connection between the wells. In addition, the section between the markers/horizons with the stratigraphy may be filled.




Seismic Display Properties

The icon  launches the seismic display properties dialog. This window (below) is used to change the color spectrum of the seismic and to define amplitude clipping range. The clipping range can be set to 'none' for no clipping. Once the parameters are set, press the *Apply* button to see changes of the seismic display in the WCP.



The color settings for the displayed seismic line are changed here.

Well Display Properties

The well display properties are launched by pressing this icon . The well display properties window opens the settings for individual well or for all wells. By default the settings are defined only for an individual well, as seen on the top of this window. To apply same settings to all wells displayed in the WCP, use the *Apply to all wells* button. Moreover, the tabs *Log 1* and *Log 2* are normally defined independently. However, both tabs have identical options.

By default, only one panels is displayed for the log display. The *Panel -- Number* field on the top of this window is used to add more panel and its corresponding settings. For instance, if *Number* is 2 and *Panel* is 1, and the user defines the settings in Log 1 tab, then the settings are only defined for the Panel 1. To apply new settings for another log panel, please change the *Panel* to 2.

In the *Log 1* tab, the log is selected from the drop down list of *Select log field*. The *Specify* option enables specification of a clipping/non-clipping range. If the *Specify* field is set to 'data range', the min/max log values are used to display the log curve (i.e. non-clipped). A user may over-rule this by setting *Log range (min/max)* manually. The selected log ranges can also be flipped by checking the *Mirror* option. The *logarithmic* option can be used to change the log curve display into a logarithmic display. The line *thickness* and *color* properties can also be modified. The log curve can be filled by setting the *Fill* properties. There are three types of filling properties currently supported: filling the curve on the left- or right side of the log or the full panel. Optionally, the log may be filled with a single user-defined color. Another option is to flip the color table, which reverses the color spectrum. Similar settings can be set for the other log from the tab *Log 2*.

Display properties of: F02-1

Select Well F02-1

Panel 1 Number 1

Display well stratigraphy

Log 1 | Log 2 | Markers

Select log Gamma Ray

Specify data range Flip

Log range (min/max) 1.3392 129.177 Logarithmic

Style Well log Seismic

Line thickness 1

Line color Black

Fill Left of log single color

Fill with Gamma Ray flip color table


SandShale 1.3392 129.177

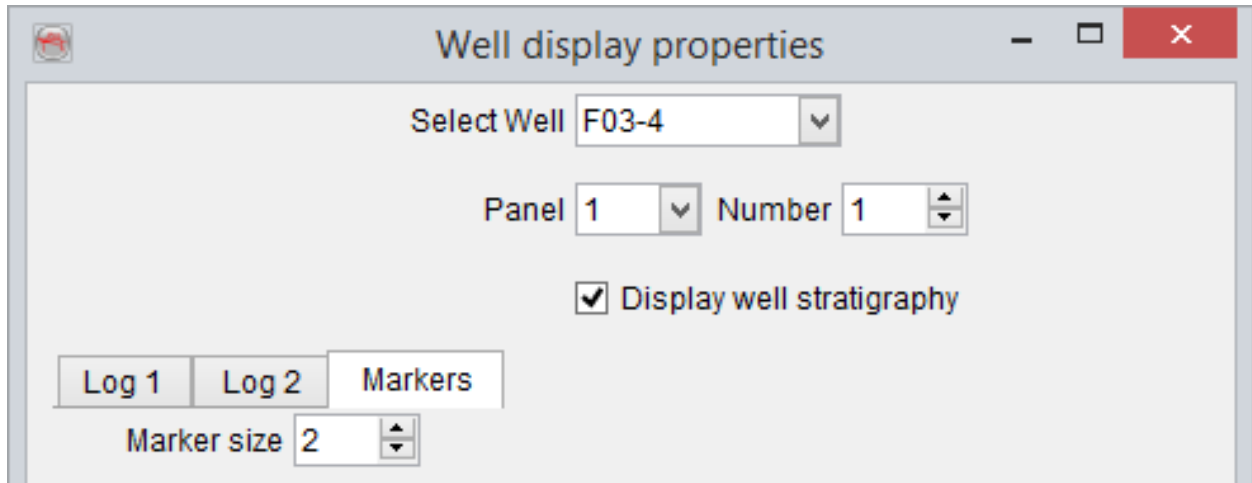
Apply to all wells

Save as Default

Close Help


Well logs are selected and their display settings are changed in this window.

Another WCP display option is to overlay the WCP with Stratigraphy  defined in OpenTect. This is added from the WCP well display properties (*Stratigraphy* tab)




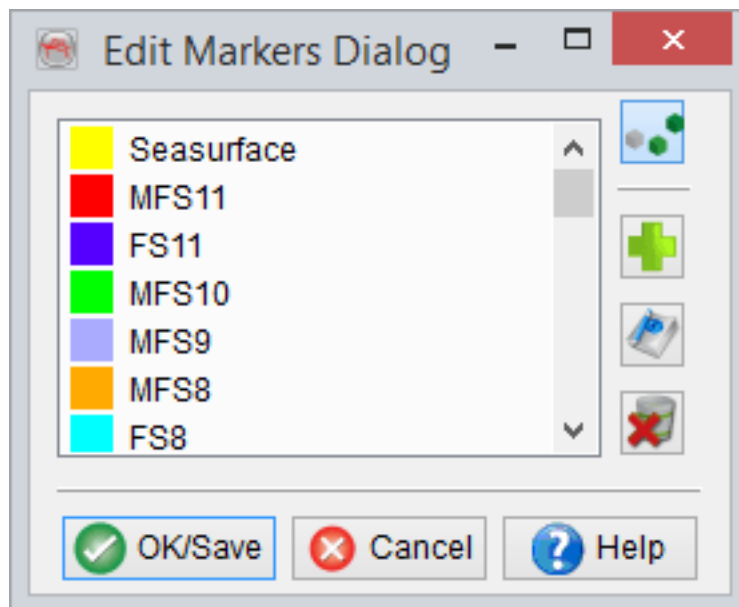
Display Stratigraphy in WCP

7.4 Pick Markers and Correlate

New well-markers are defined or alternatively, the existing markers can be modified using WCP. The purpose of this functionality is to build a new stratigraphic framework of the area. The *Edit Makers dialogue* is launched by clicking on this  icon.

How to pick markers?

By keeping this dialog box open, markers are added to the well using a mouse click on the well-panel displayed in the WCP. Before starting picking, the 'interact' (Pick) mode should be toggled on . Once markers are interpreted and added to all wells, press the *OK/Save* button to save the markers.



This dialog box is used when picking new markers.

8 SynthRock



8.1 Introduction



The SynthRock plugin is a forward pseudo-well modeling and probabilistic inversion package to achieve qualitative and quantitative seismic interpretation. It supports wedge models, stochastic models, pre- and poststack synthetic seismograms generation and cross-matching (HitCube) inversion. This commercial plugin extends the capabilities of the [Basic Layer Modeling](#) module of OpenTect.

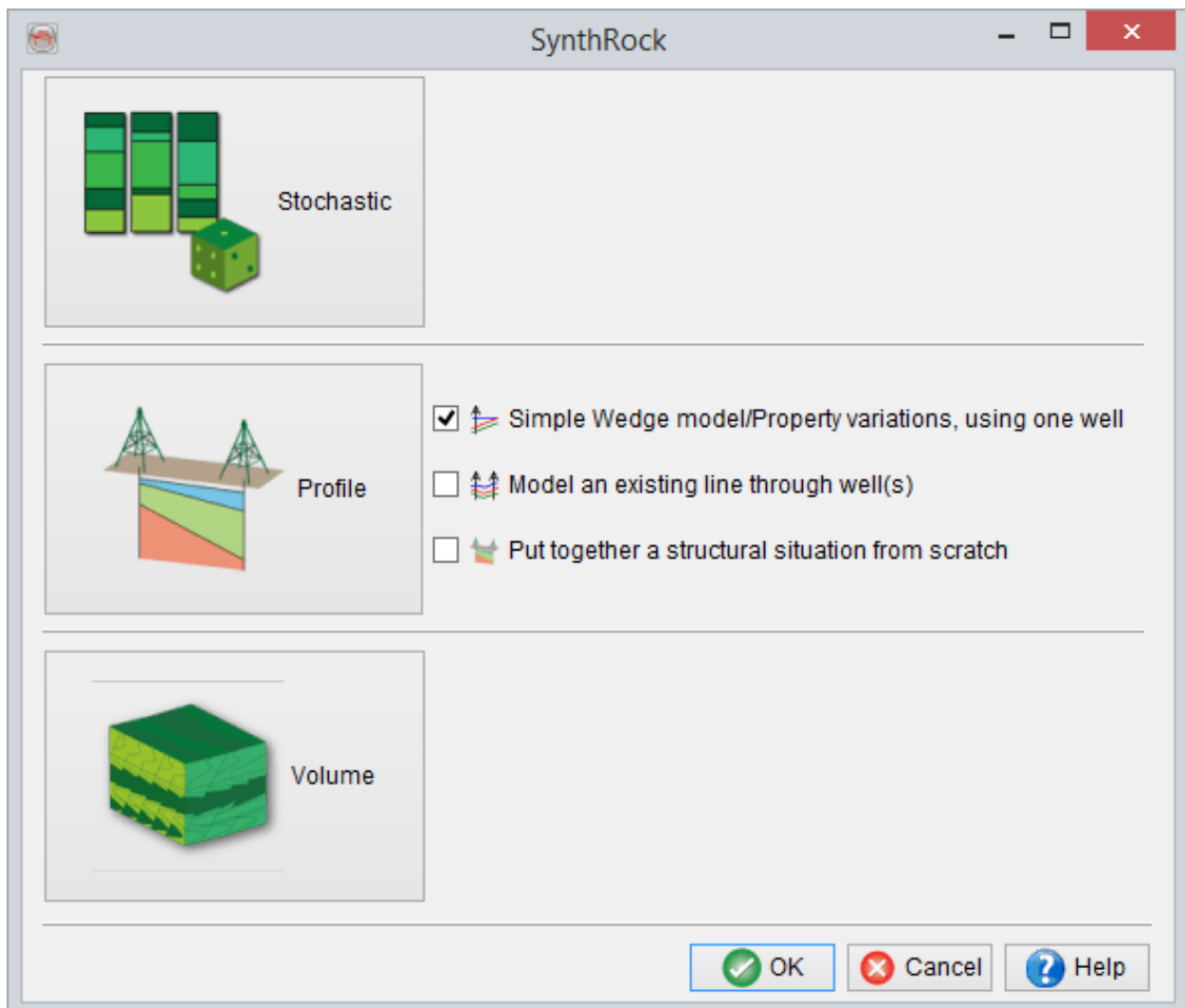
Forward modeling of synthetic seismic data plays a key role in understanding the seismic response of a target interval. A thorough understanding of the response enables interpreters to separate true hydrocarbon anomalies from false positives and to make more accurate reservoir predictions. In a forward model, geologic information and synthetic seismic data is integrated through a rock physics model.

Sophisticated workflows in which model parameters are varied stochastically can be run to create a data base of pseudo-wells, representative of the expected geologic and seismic variations at target level. Such models are then used to predict rock properties with uncertainties from pre- and poststack seismic volumes.

The new “Volume” module of SynthRock allows the generation of poststack and prestack synthetic seismic data from 3D rock property volumes (AI-SI, Vp-Vs-Rho), for comparing the synthetic seismic data with the recorded seismic data.

SynthRock can either be started in two ways:

- From the corresponding icon in the main OpendTect toolbar .
- From within the stratigraphy manager with the icon .



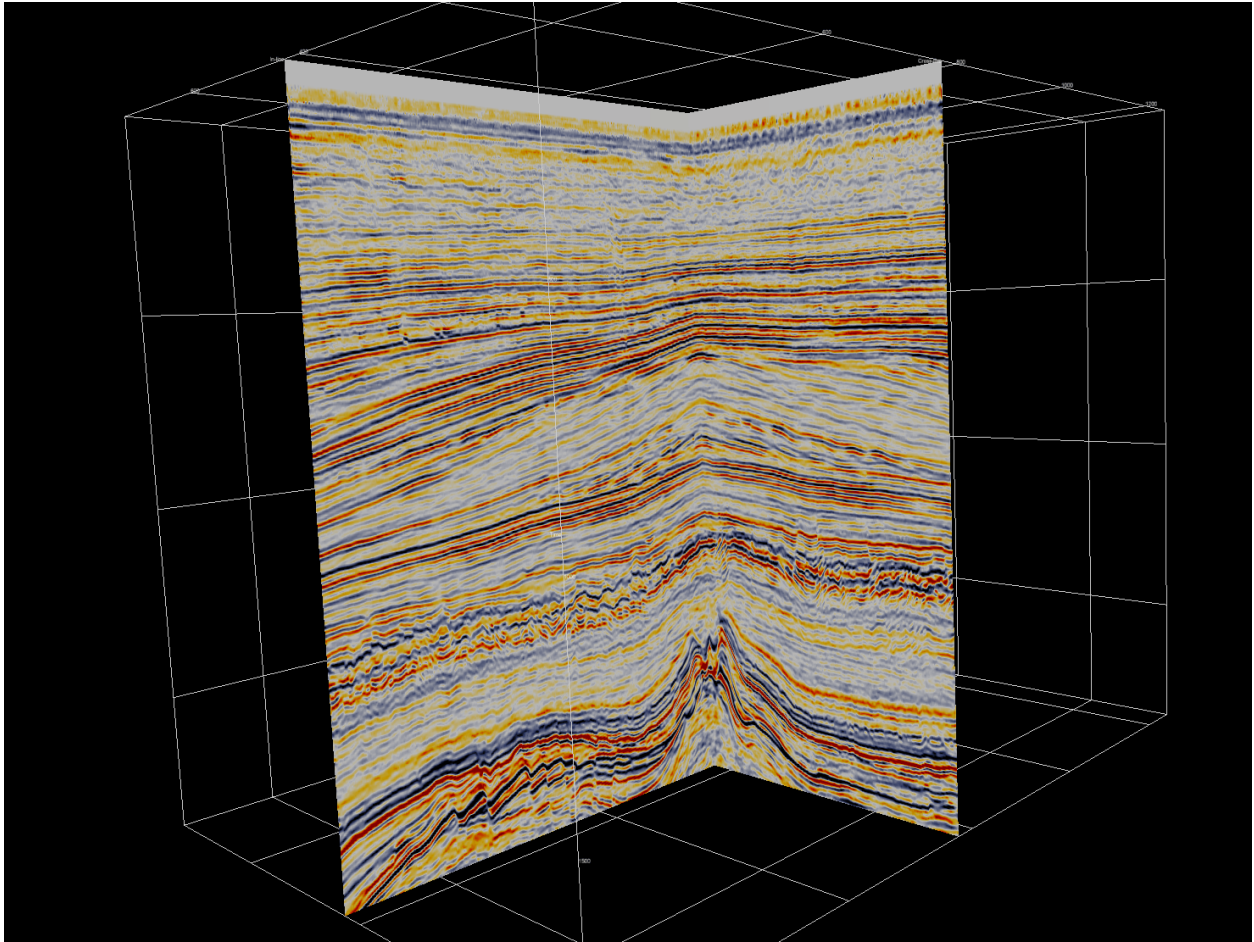
SynthRock can be used to do the following tasks:

1. Perform a stochastic pseudowell modeling.
2. Create pseudowells profiles by interpolating between/from existing wells.
3. Applying Gassmann fluid substitution on the modeled pseudowells.
4. Generate synthetic prestack seismic using full zoeppritz from either the pseudowells or 3D rock property volumes.
5. Extract synthetic seismic attributes and/or well layer attributes from the modeled pseudowells and optionally export them along a 2D line.
6. Run the HitCube stochastic inversion from the modeled wells.

Once a layer modeling window is started, you cannot change its type between basic, stochastic and profile. To use another model type you need another window, to be launched similarly from either the analysis menu or from the main toolbar icon. Only one layer modeling window (thus SynthRock window) can be used at the time. That is because the stratigraphy object needs to be locked during the modeling. A full uncoupling will be implemented in later releases. For the time being you will need to launch additional OpendTect main windows if you want to work with several models simultaneously (at your own risk).

A stand-alone SynthRock application will be made available post 5.0.

8.2 Volume Synthetics



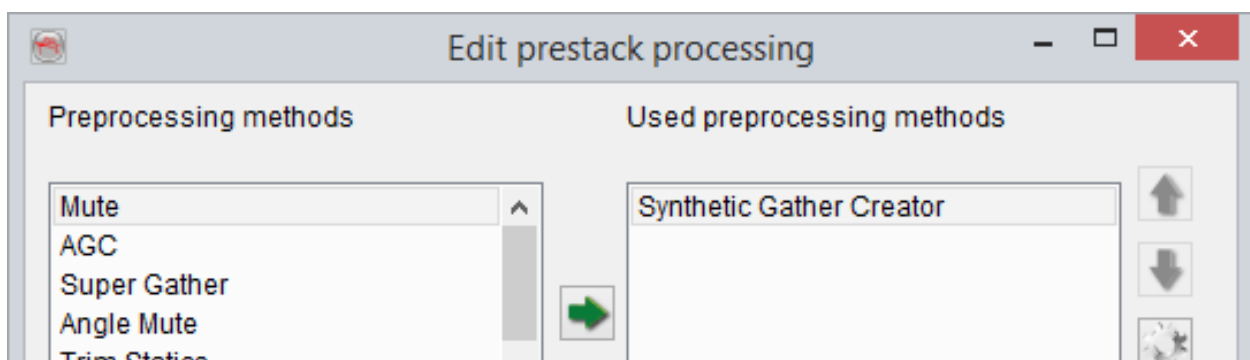
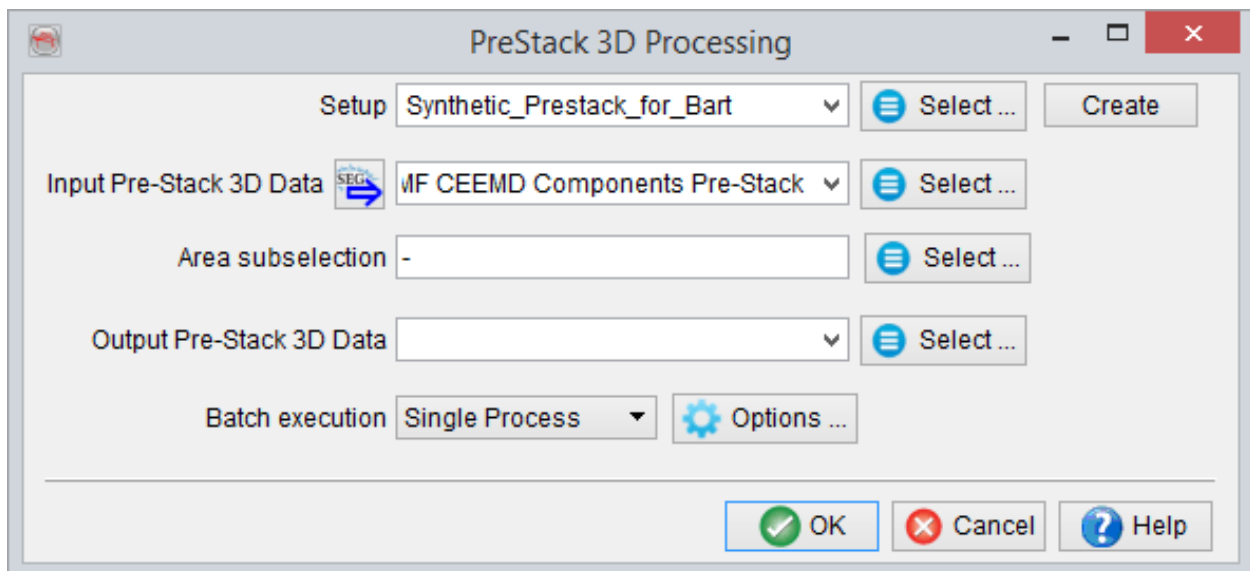
Introduction

Volume Synthetics is used to create Acoustic, or Elastic synthetic seismograms from user-supplied elastic parameters. For acoustic modelling the user either inputs volumes of P-wave and Density, or a volume of Acoustic Impedance values with Density as an optional input. For elastic modelling the inputs are: P-wave, S-wave and Density, or Acoustic Impedance and Shear Impedance with Density as optional input.

Synthetics are calculated by ray-tracing through a horizontally layered isotropic earth model. SynthRock uses a sophisticated, ray-trace based, synthetics generating algorithm that accurately solves Zoeppritz' equations. It supports generation of: PP, PS; near, mid, far, full and angle stacks, from which other properties such as Gradient Impedance, AvO attributes and Extended Elastic Impedance, can be derived. Option-ally multiples can be generated.

Volume Synthetics is integrated in the Prestack processing system under the option Synthetic Gather Creator. This option can only be included as the first step in a pre-processing workflow.

Volume synthetics is either launched from the SynthRock launch menu, or from the Processing -> Prestack Processing ... menu. From the latter menu the Prestack Processing window is launched. Here the locations where synthetics are to be computed are specified under *Area sub-selection*. To define a pre-processing work flow press *Create*.



Input parameters are specified in the SyntheticGatherCreator setup Window, which is launched from the Properties ... button.

Synthetic Data Creator setup

Input type Acoustic Elastic

Input Source PWave, Density

P-wave Velocity cube Velocity model (INT) Select ... Edit ...

Density ID=<100010.200> Select ...

Ray-Tracer Advanced RayTracer

offset range (m)(start/stop) 0 3000 Step 100 Zero Offset

Downward wave-type P S

Upward wave-type P S

Blocking threshold 1 %

Select spreading geometry None

Wavelet stat 120 Manage Wavelets ... Extract ...

Apply NMO corrections Yes No

Advanced ...

OK Cancel


In this window select the synthetics modelling type (Acoustic, or Elastic), the input volumes, the wavelet, and what output to generate. The various options are described in [Appendix A - Synthetics Generation](#).

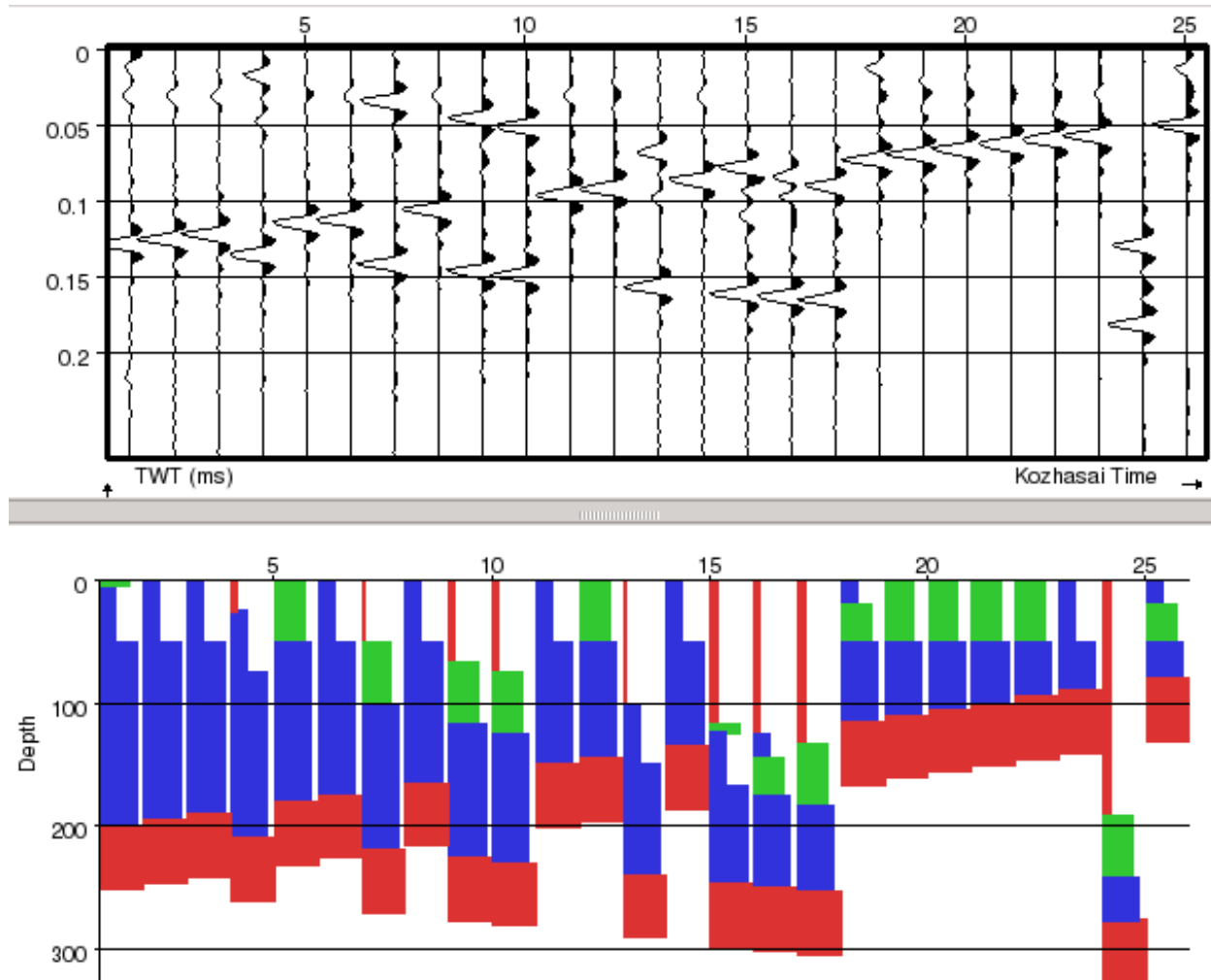
8.3 Stochastic Pseudowell Modeling

The stochastic modeling is performed using the following workflow:

1. First the [modeling nodes](#) must be selected. They can be single layers, formations, or meta-formations.
2. For each member of each node the [stochastic parameters](#) must be set. Default constant parameters are provided.
3. Pressing "Go" will draw a user defined number of pseudowells that honor the modeling description.
4. Then [fluid substitution](#) may be applied. This process duplicates the pseudowells into a brine set and a fluid-filled set.

The modeling itself will take place on the left side of the layer definition window. The modeling description can be saved, but not the stochastically derived pseudowells, except by extracting the corresponding data using the [crossplot extractor](#).

The pseudowells are re-drawn by either pressing on the  button or resizing the layer modeling window (sic). It is advised to start with a relatively low number of pseudowells while building the model, and then to increase it for the final application (several hundreds for a [HitCube inversion](#) are needed). An example is given below.

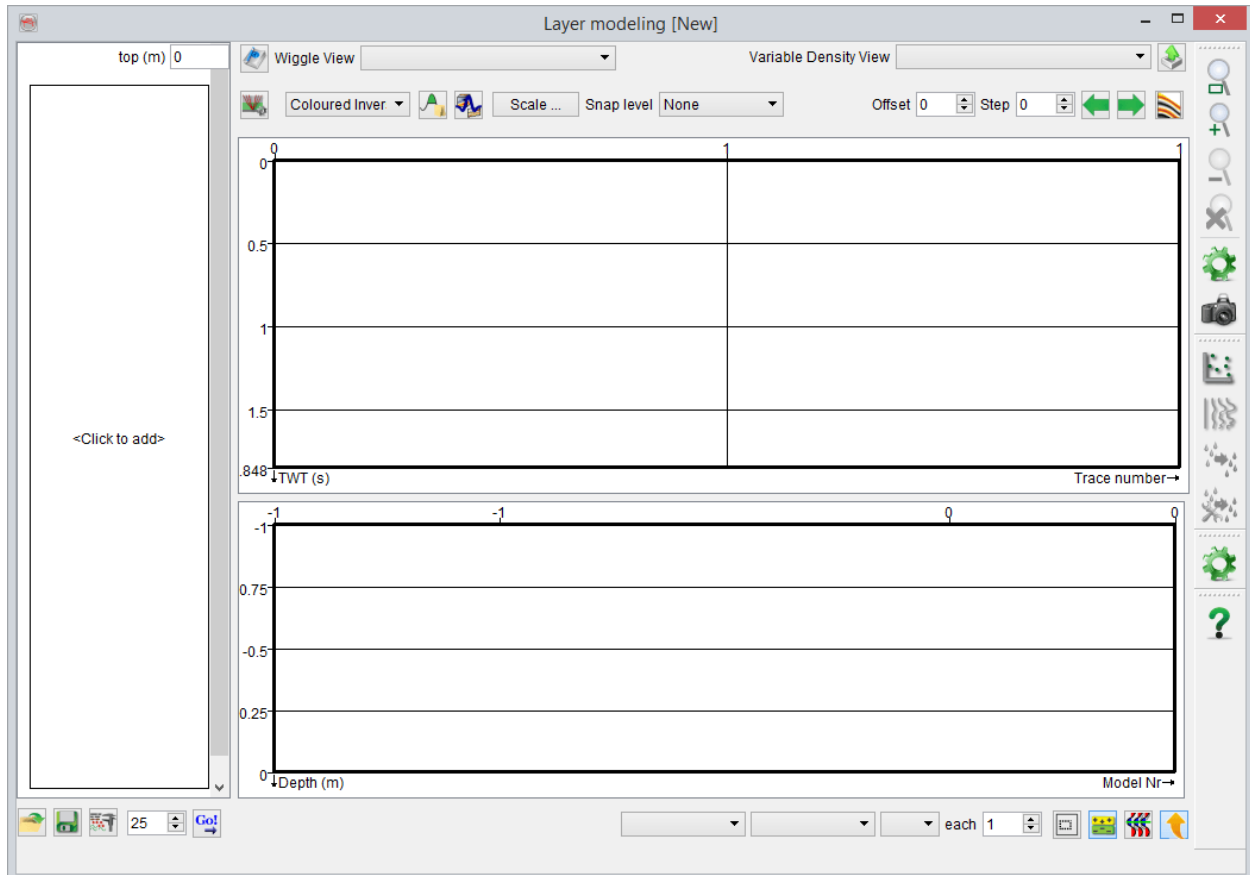


8.3.1 Add New Modeling Node

The modeling description is typically composed of a number of nodes that are added in a given order in the layer description. The nodes are taken from the current stratigraphic framework, thus this framework must first be filled with the formations and lithologies that will be used during the modeling, prior to the modeling.

How-to add a node

Click on the left side of layer modeling window for the clean start, or right-click in this column when there is already at least one top node:




top (m) 0

ene.Messinian.Undef

- Edit ...
- Add Above ...
- Add Below ...
- Remove


e.Tortonian.Undef[Oil]

Empty vs existing


For the first clean run of the simulation (i.e. before any node has been added), specification of layer properties to be used in the modeling is mandatory. Various rock properties can be selected from the available list. It is also possible to define a new property by pressing . For more details go to [Manage Layer Properties](#).

Simulation units can be:


- Single layers, one lithology of a given formation. This is useful only if you want to insert a single, most often thick, blocky formation into the modeling.

 *This option may be chosen to model relatively thick caprock shale above a (sand) reservoir as well as for modelling the underlying shale.*

- Sequence of layers, it could either be a single formation composed of all the lithologies it can contain or a meta-formation. Meta-formations regroup all the formations (and their corresponding lithologies) underneath, under a single tree structure, as individual simulation units (example below).

 *The above option is useful while modeling multiple lithologies within a formation, e.g. an interbedded sand-shale sequence in the reservoir with a particular Net-to-Gross ratio or Sand-to-Shale ratio.*

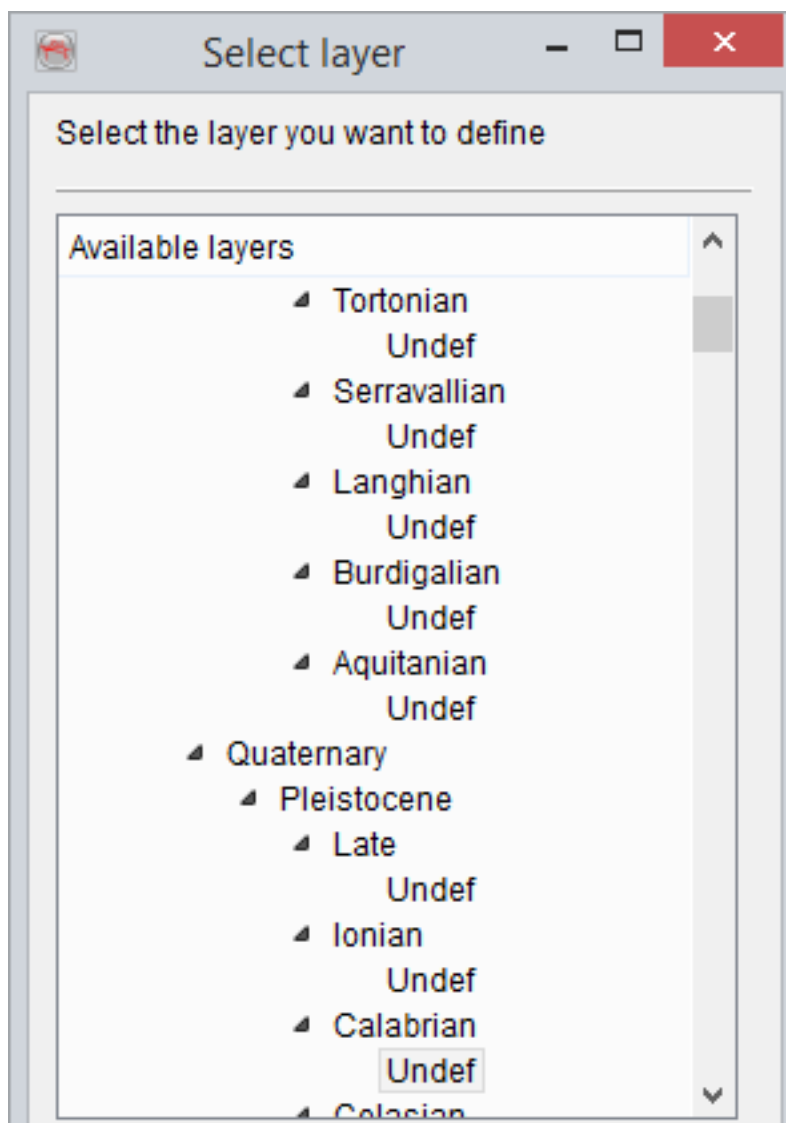
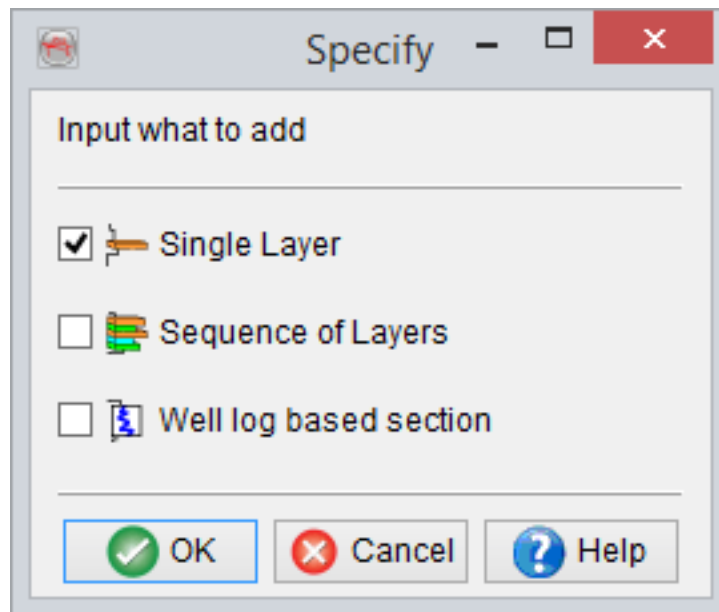
- Well-log based section, it can contain either full or part of well log data, which has been upscaled/blocked at a constant size.

 *This option is of particular use when a detailed overburden model is required, e.g. for generating pre stack synthetic gathers, where in the ray-tracing phase requires specification of the elastic model from ground-level down to the interval of interest. This could simply be achieved by choosing*

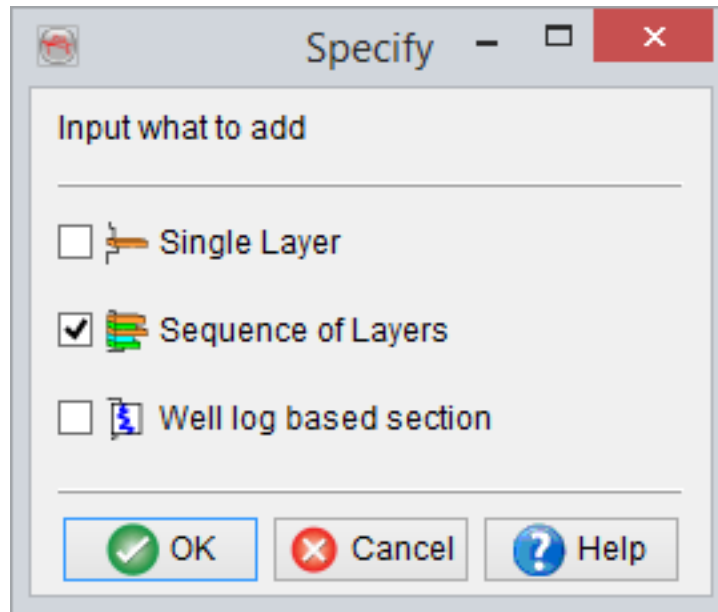


V_p, V_s and Density logs from an available well, where the logging is done from ground level down to the interval of interest.

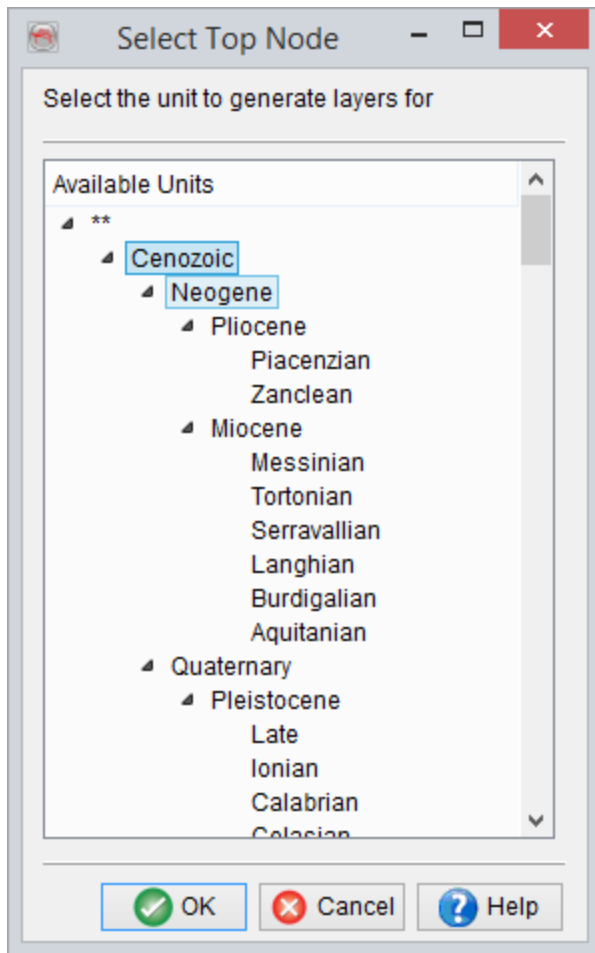
Adding a Single Layer:



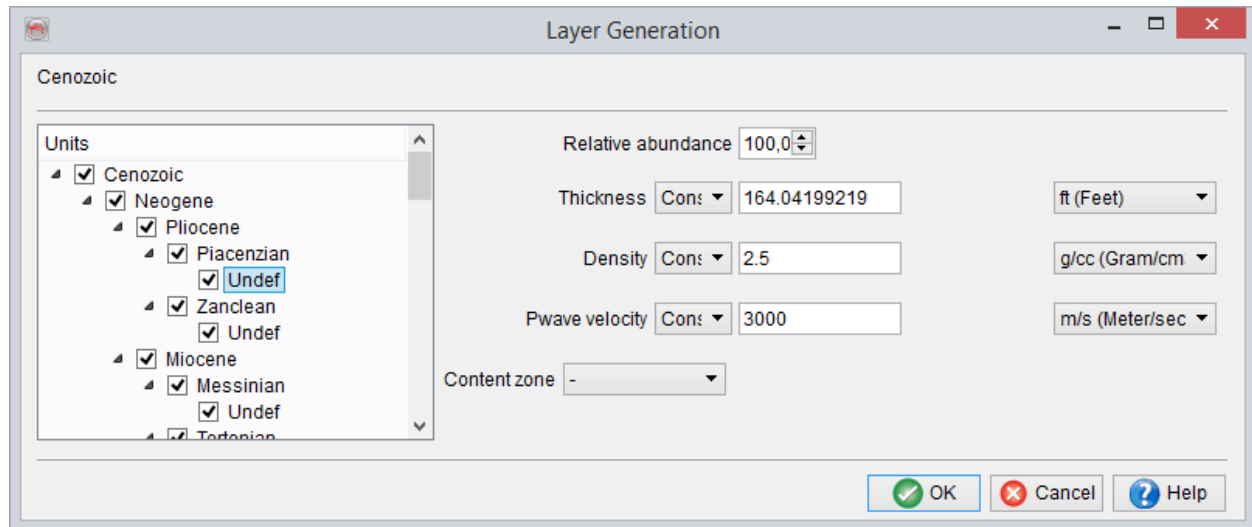
Adding a Multiple Layers:



On first-time addition of multiple layers, you will be prompted to select the top node from the following window:



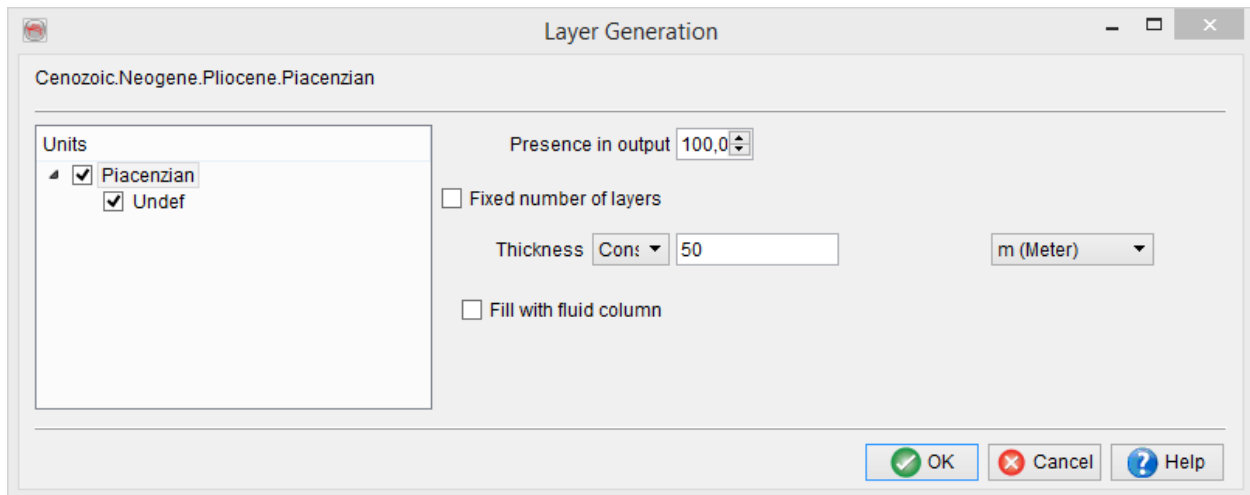
Before being taken to the Layer Generation window:



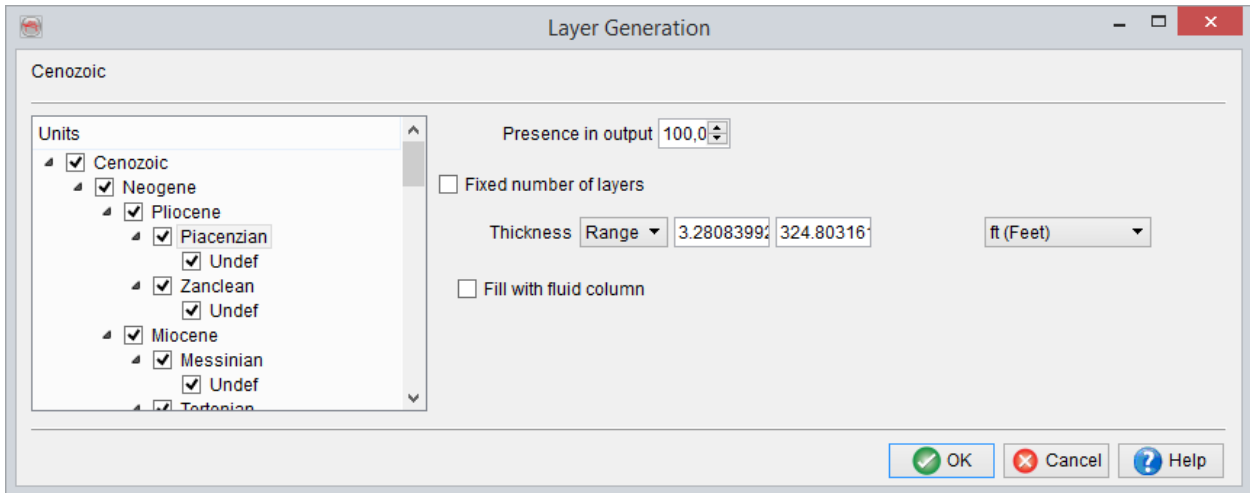
8.3.2 Model Definition

The modeling will distinguish three types of nodes:

1. **Meta-formations** These nodes are specific by the fact that they do *not* have lithologies on the level below, but formations. They are used to **control the order of appearance** of the formations during the simulation. For instance in the example below the top node "Chalk" is used to make sure that the "Ommenlanden" formation is always modeled on top of the "Texel" formation. Optionally one can also set the probability of presence of the entire node.

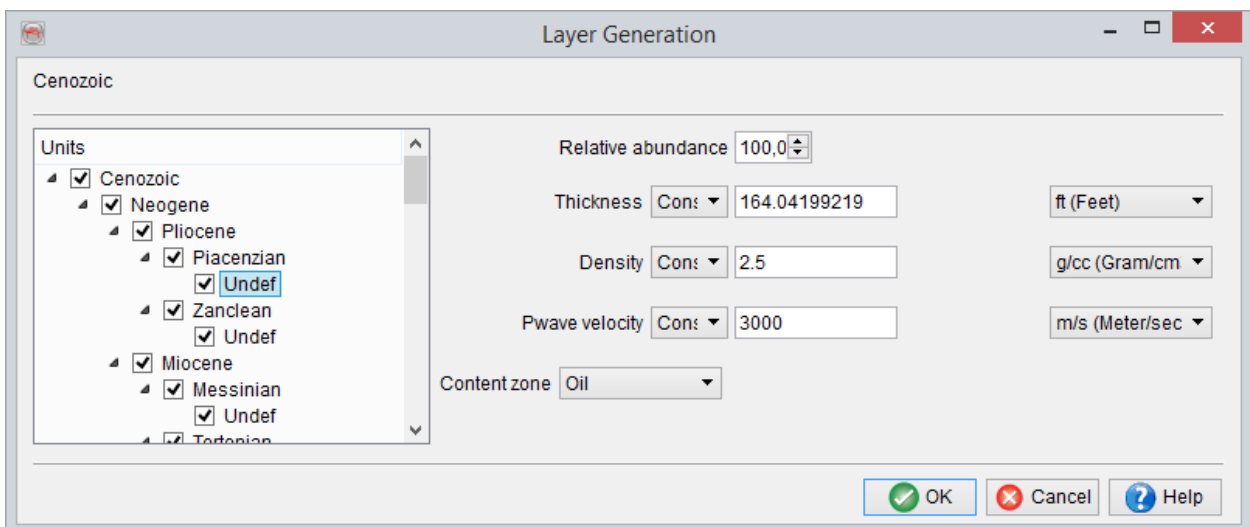


2. **Formations** These nodes are specific by the fact that they only have lithologies on the level below. The simulation will add blocks taken from the leaves from bottom upwards according to the thickness and probability of presence settings of the leaves, until the thickness of the formation is reached. See example below:



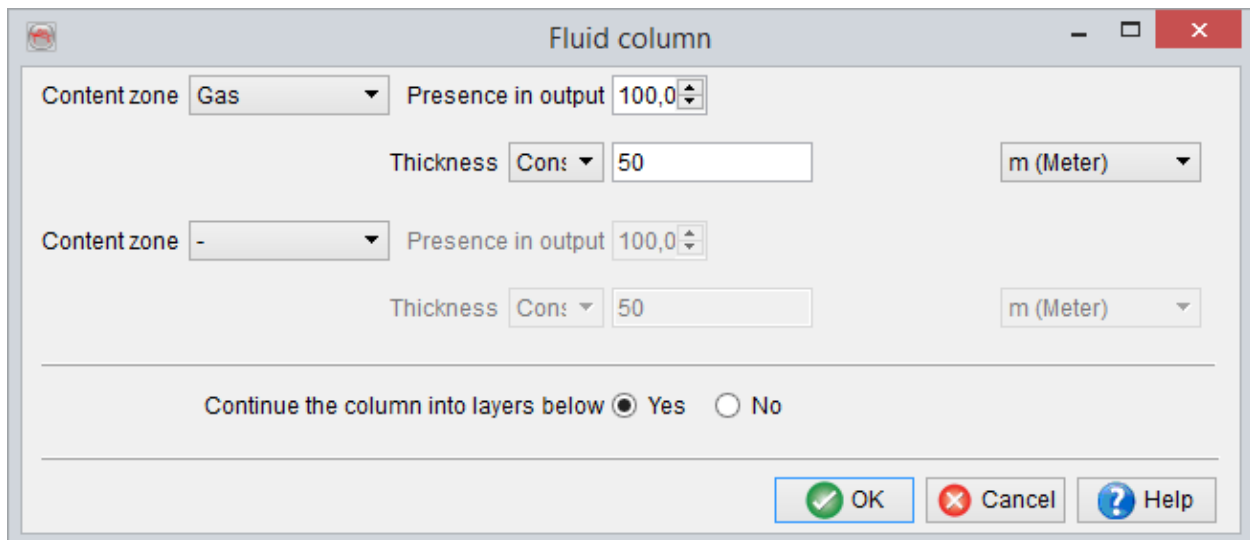
Optionally the thickness of the formation can be constrained by the number of blocks, instead of the sum of their thickness. See example below:

3. **Leaves** The leaves are the key item of the pseudowell modeling. Each leaf that is modeled is a small layer, stacked on top of the previously created layers of the pseudowell. It belongs to a given formation, has a top depth (Z), thickness, log values, and a given fluid content. See example below:



The relative abundance defines the probability to find a given lithology in a formation. It is used to draw the lithology type to add when inserting a new block in a formation, after the previous. It can be used to model a net-to-gross ratio, but the user will need to make sure that the sum of the relative abundances of all leaves is equal to 100%.

It is possible to specify if some fluid is present or not by selecting the box *Fill with fluid column* for the meta-formations and formations. If the option is selected, the characteristics of the fluid column have to be *Defined*. It will consist in at least one fluid. There can be as many fluids present as you defined in the Manage Contents window. This fluid column will be present in every generated pseudo-wells if the *presence* in output is 100%. In addition to the probability of occurrence, its thickness can be set to constant or varied randomly or within a user-defined range. The fluid column can be continued into the layers below if for example the layer below is also a reservoir for the case if the thickness of the fluid column exceeds the total thickness of the layer where it is defined in some pseudo-wells.



Fluid column

Content zone Gas Presence in output 100,0

Thickness Cons 50 m (Meter)

Content zone - Presence in output 100,0

Thickness Cons 50 m (Meter)

Continue the column into layers below Yes No

OK Cancel Help

At the lithology level, the fluid content can be simply one of the fluids available in the Content manager or a fluid column can also be defined.

All numerical values can be set in the following ways (select the desired method and press set if applicable):

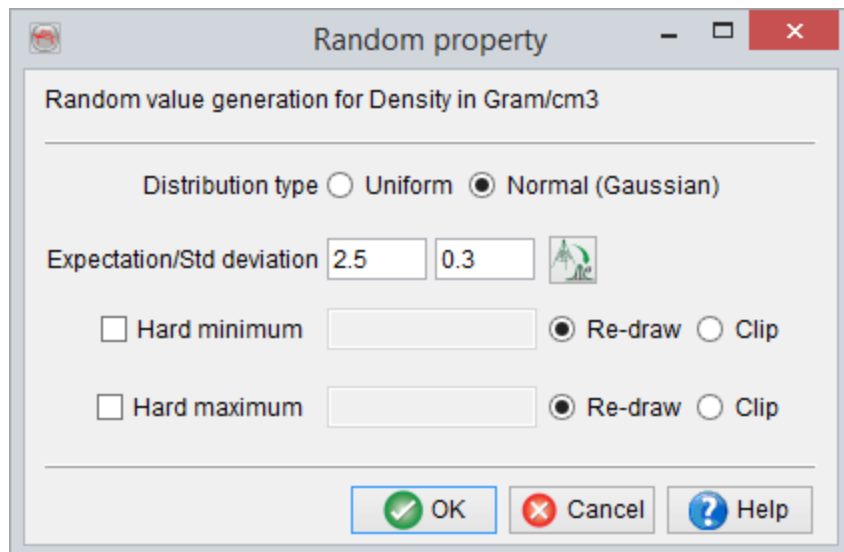
- Constant value (default when starting).
- Range: A linear variation with increasing pseudowell index
- Random: Gaussian or uniform distributions.
- PDF: A distribution drawn from a saved probability density function.
- Math: Value computed from other quantities.

8.3.2.1 Random Distribution

The modeled parameter value can be drawn from a uniform or normal (gaussian) distribution. This later is the most realistic for most logs, and formation thicknesses. It can nevertheless happen that the thickness distribution is uniform throughout the survey. A good way to estimate what distribution could be used is plot the histogram of the available data: Well logs for the logs, isopach maps for the thicknesses when the top and base have been interpreted. Displaying the histogram of the available data is more important than one could expect: If the log is spiky, the distribution will be skewed and the extracted average will be shifted from the real position.

It is possible to constraint the drawn value with hard extrema. There are two modes to do this:

- Either the value is re-drawn until it falls within the boundaries. You would typically do this for a sonic or density log.
- Or the value is clipped to the extrema. This is typical of a porosity log at its minimum (0%) or a water saturation log at its maximum (100%), since you can expect many of the layers to have exactly the extrema value.



8.3.2.2 PDF Distribution

The modeled parameter value can be drawn from the distribution read in a stored probability density function. Stored PDFs have multiple dimensions, thus you will need to select what dimension to use. A single PDF can thus be used to provide the input for multiple quantities. Also a transformation can be applied on the values stored in the PDF, to simulate sonic when the PDF contains P-wave, or to simulate $\log(K)$ when the distribution contains K (permeability).

PDF property

PDF-based value generation for Density

Probability Density Function

PDF dimension

Transform values (example '1/x')

\

Output unit of measure

Hard minimum Re-draw Clip

Hard maximum Re-draw Clip

8.3.2.3 Math-based Layer Properties

Mathematics equations can be set to derive pseudowell properties from other modeled properties, or using simple rock property models. For instance, this first example computes the P-wave velocity value in each layer from the modeled density.

Calculate new logs for 'F02-1'

/ MathFunctions sqrt (Square ro Insert p=vd

Formula (like 'den / son') (Rhob / 0.31) ^ (1 / 0.25) Set

For 'Rhob' use Density convert to: kg/m3 (Kg/m3)

Formula result is Velocity -

Output sample distance 0.15124556 Feet

Inter/extrapolate input logs? Unless all und

Name for new log Density

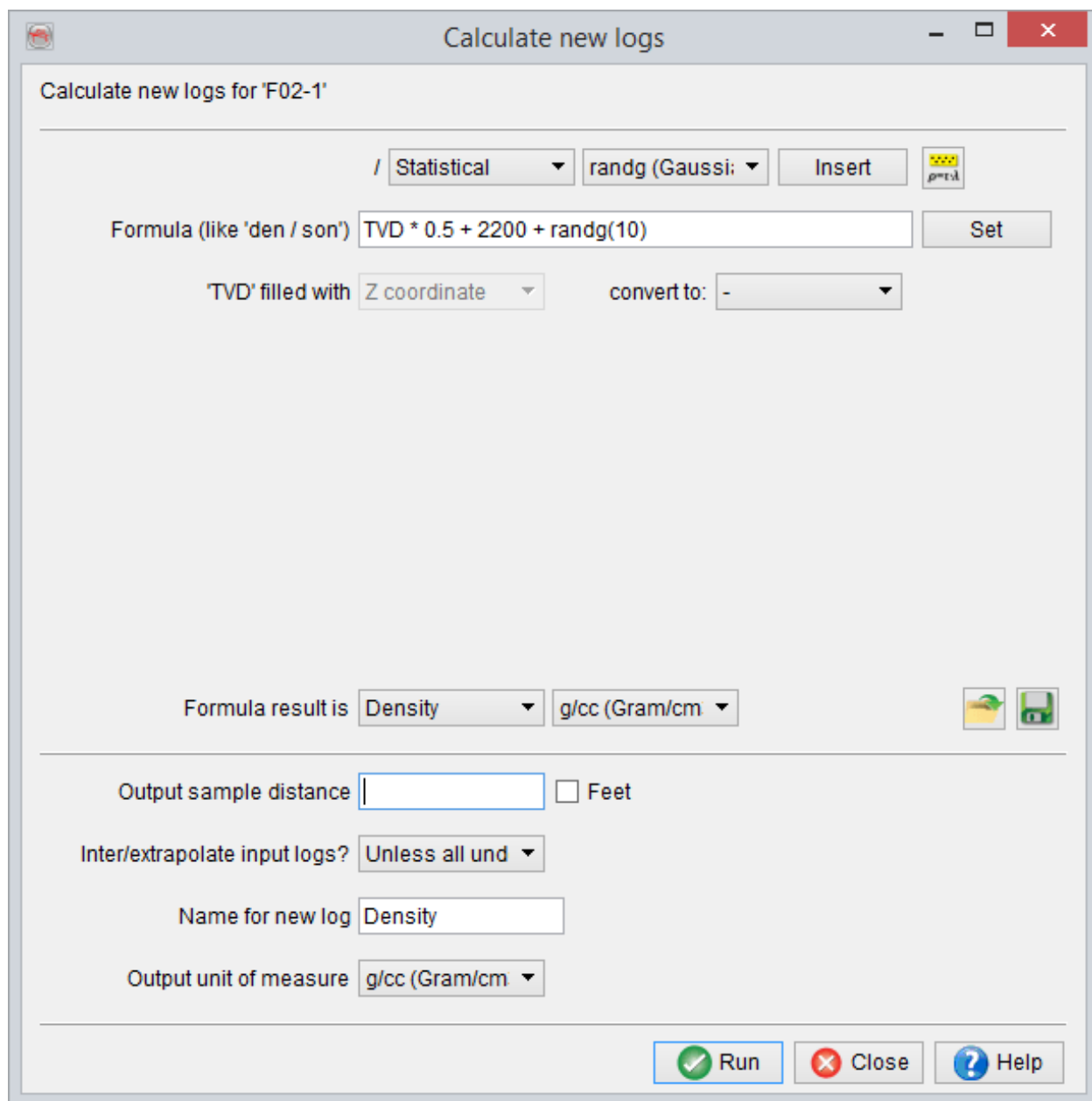
Output unit of measure s (Seconds)

Run Close Help

Input properties are converted to the selected unit of measure before the equation is computed. The Formula output unit have to be specified, since it might be different than the default property unit of measure. Example: The formula delivers a result in g/cm³, but the preferred display unit is set to kg/m³ (in the layer properties manager for the selected property).

Layer features like thickness and the depth at the top of the block can be used in the mathematical equation as input. Note that thickness is available from the drop-down property selection menu. While TVD (Depth) is available through the *Insert* option of type '*Other*'.


The second example creates a vertical velocity gradient starting from 2200 m/s at zero depth with a slope of 0.5 (m/s)/m and added Gaussian noise with a standard deviation of 10 m/s.

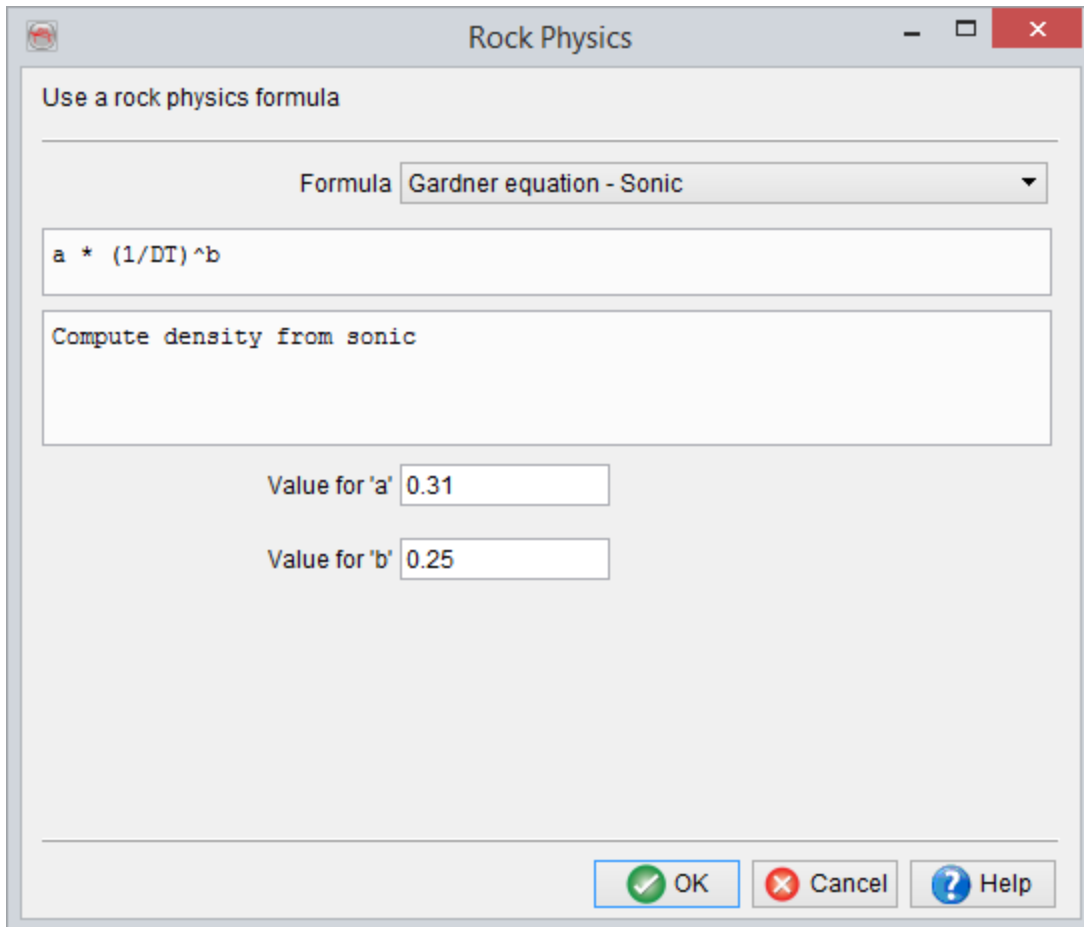


User-defined formula can be saved () and retrieved ()



The same syntax as in the mathematics attributes should be used.

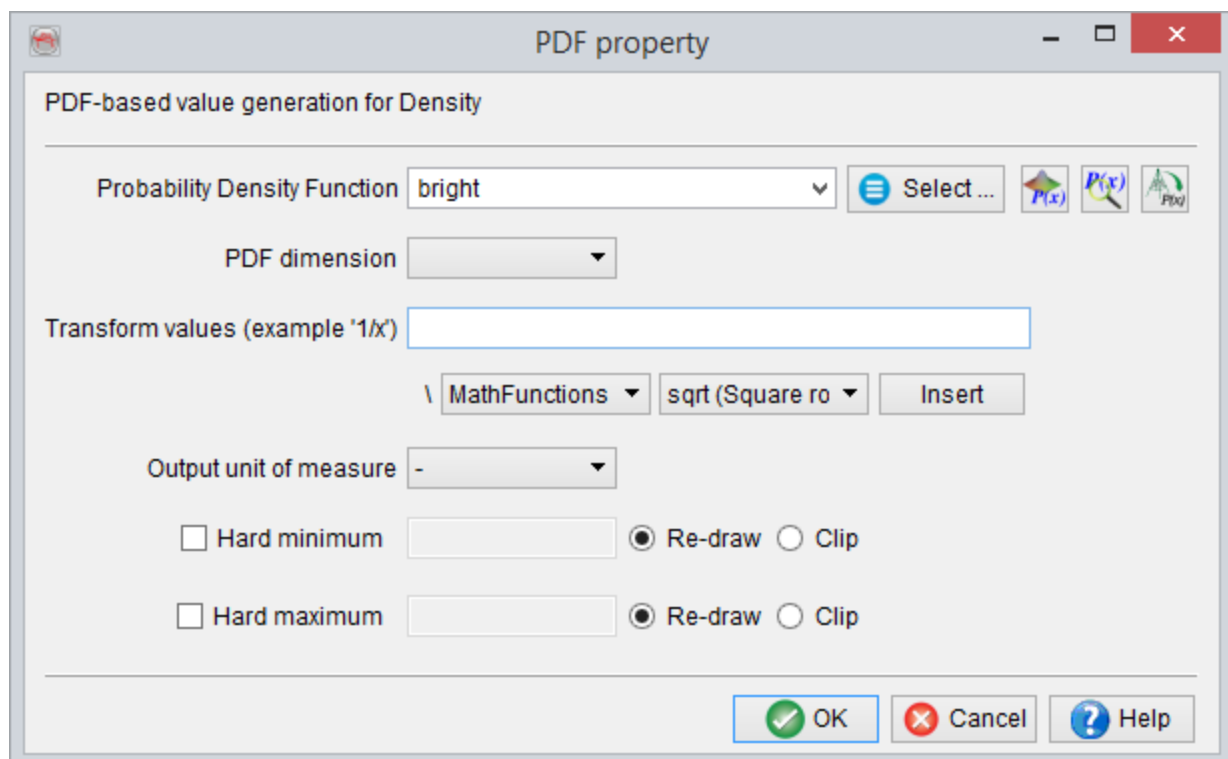
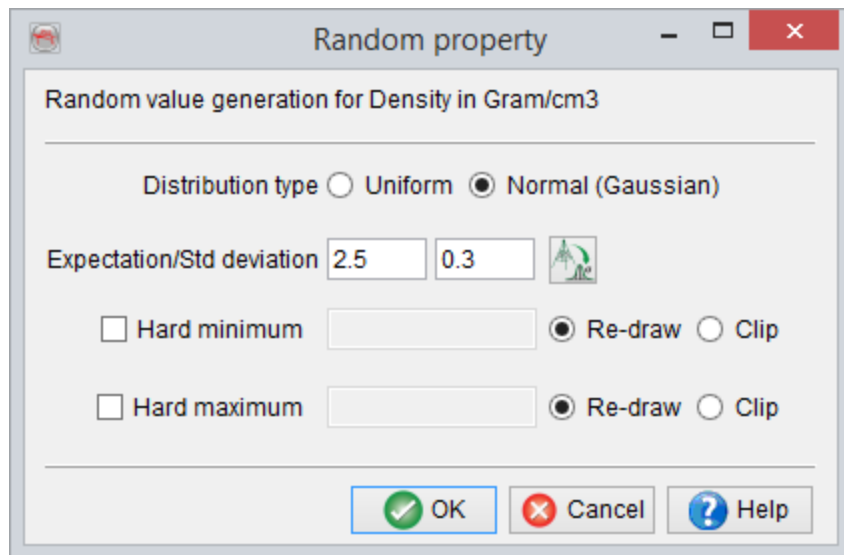
Rock physics link

The rock physics library, via  **Rock Physics ...**, provides a comprehensive set of equations that can be used to derive logs from others. Please refer to the OpendTect User Documentation.



8.3.3 Analysis of the Existing Wells

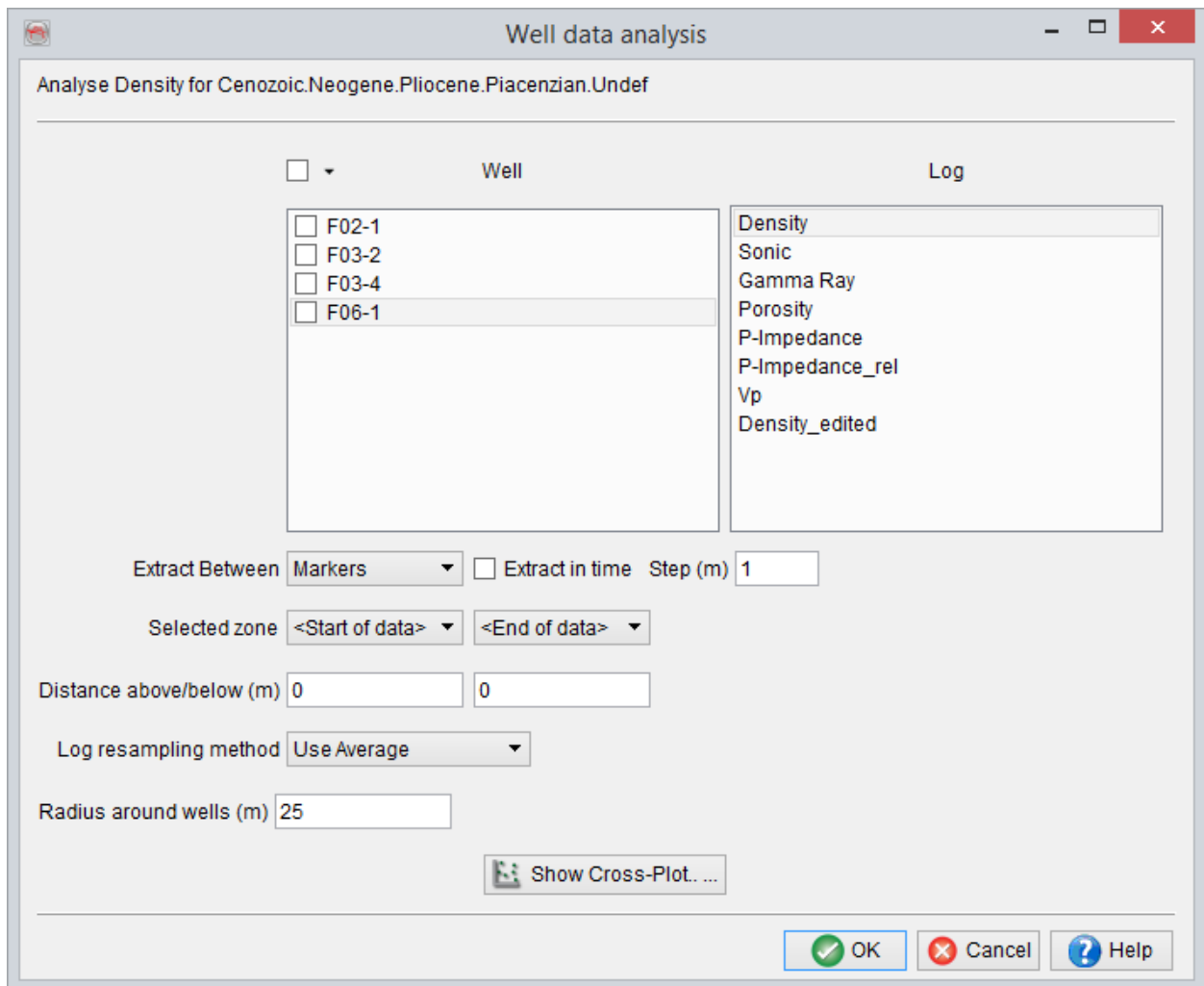
Well data analysis is a module that feeds the modeling description with stochastic parameters, either uniform/normal distributions or probability density functions, derived at existing wells. This module is accessed by clicking on the  icon when defining a [Random property](#) or on the  icon when defining a [PDF property](#).




Well data analysis window looks very similar to *Cross-plot tool* in both cases and have many similar options and features.

Random property

For a *Random property*, a single log (Density in the example below) is selected and analysed in the specified interval.



It is highly recommended, even though not required, to click on Show Cross-Plot button in order to extract a table, display and edit the data if necessary. For example, viewing

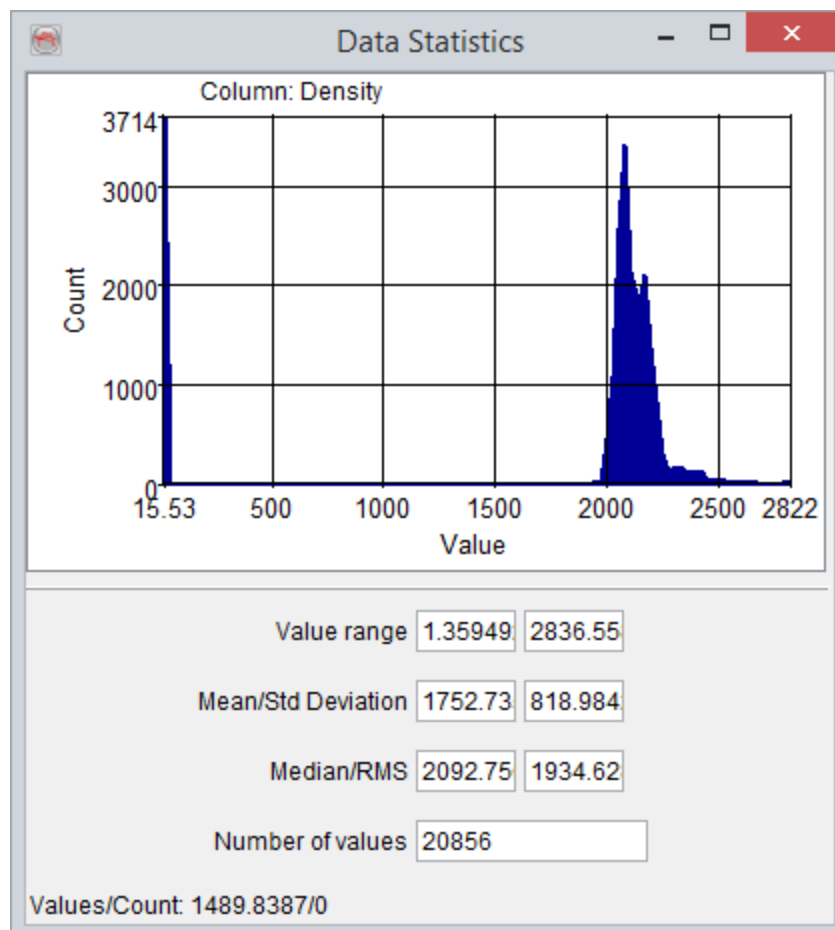
a log histogram by clicking on the  icon in the cross-plot table window can help to determine if a given property follows Gaussian or Uniform distributions in a certain interval and refine the automatically derived parameters:

Well data analysis Cross-plot

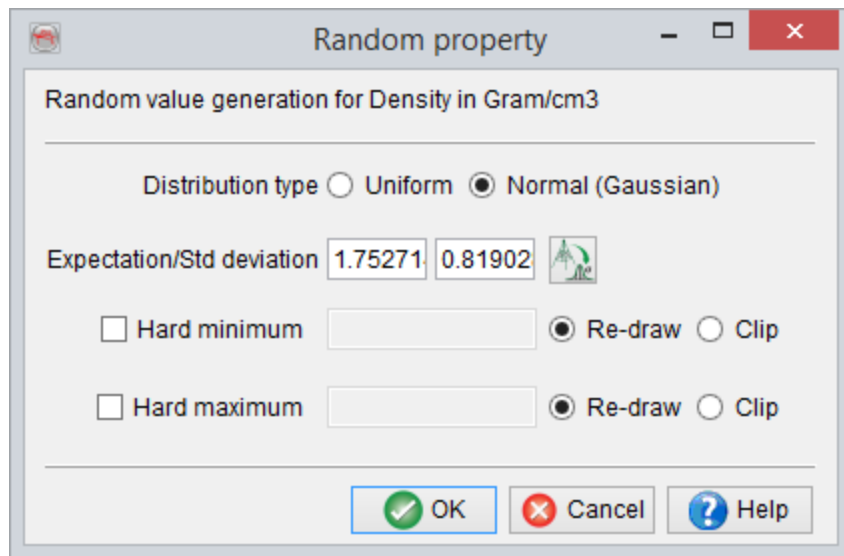
Well data:F02-1,F03-2,F03-4,F06-1

	X-Coord	Y-Coord	[X] Z (ms)	* MD(m)	[Y] Density
F06-1	607909.85229611	6077190.66652313	0.48	29.29	2049.03564453
F03-4	623255.97999955	6082586.86999977	-1.68	31.15	2077.89501953
F03-4	623239.91391156	6082595.9375224	-1.68	31.15	2077.89501953
F06-1	607903.00000011	6077213	2.67	32.29	2049.16650391
F03-4	623240.6119288	6082570.94786723	-0.54	33.15000153	2078.19018555
F03-4	623265.60218251	6082571.64629957	-0.54	33.15000153	2078.19018555
F06-1	607884.16402516	6077214.95774596	3.4	33.29	2049.25610352
F03-2	619100.99999987	6089491.00000011	4.47	34.15000153	2049.21362305

Close Help

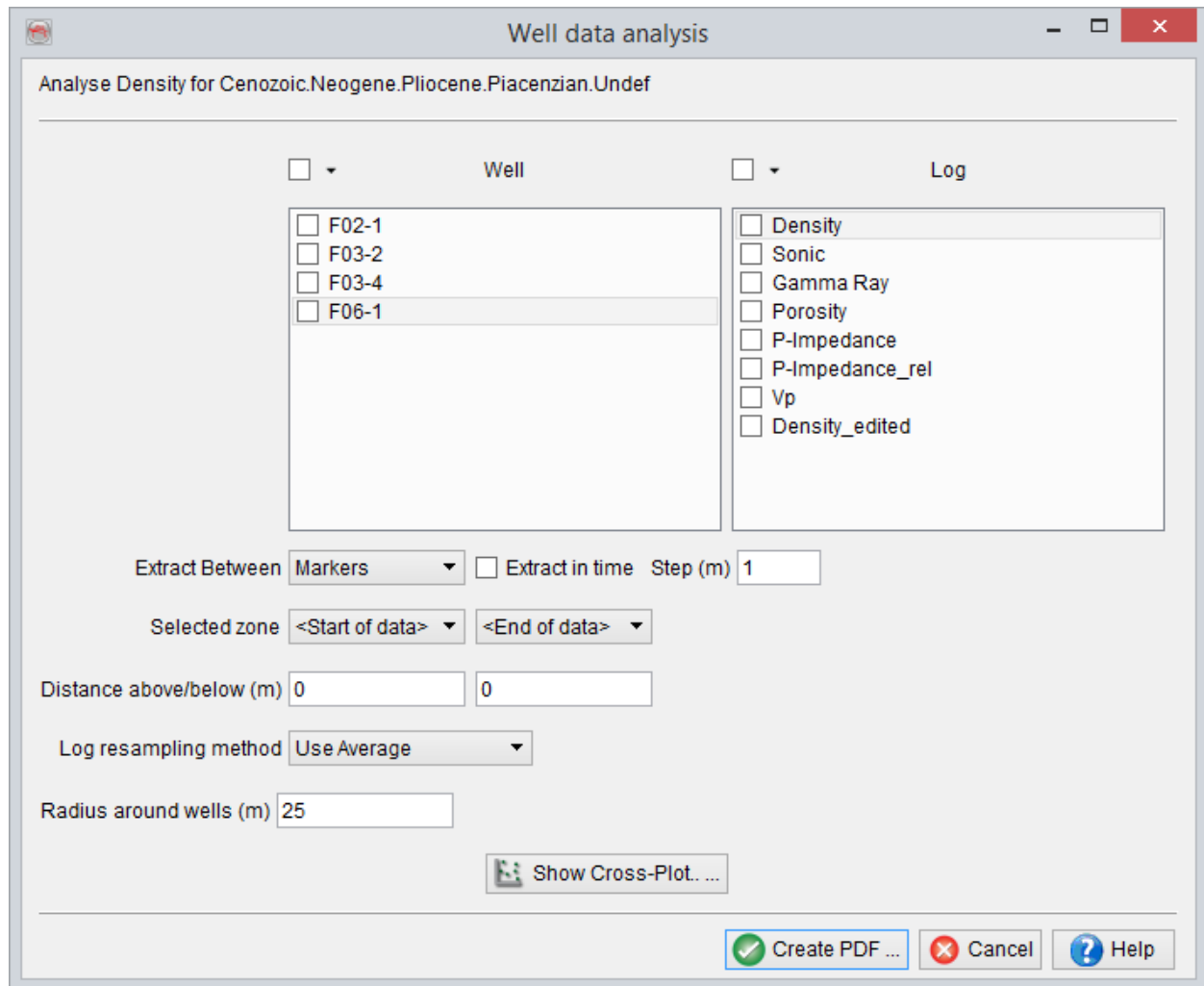



Upon OK click in the *Well data analysis* window, the derived distribution parameters are automatically fed to the *Random property* window:



PDF property

For a PDF property, several logs can be selected and analysed in the specified interval.



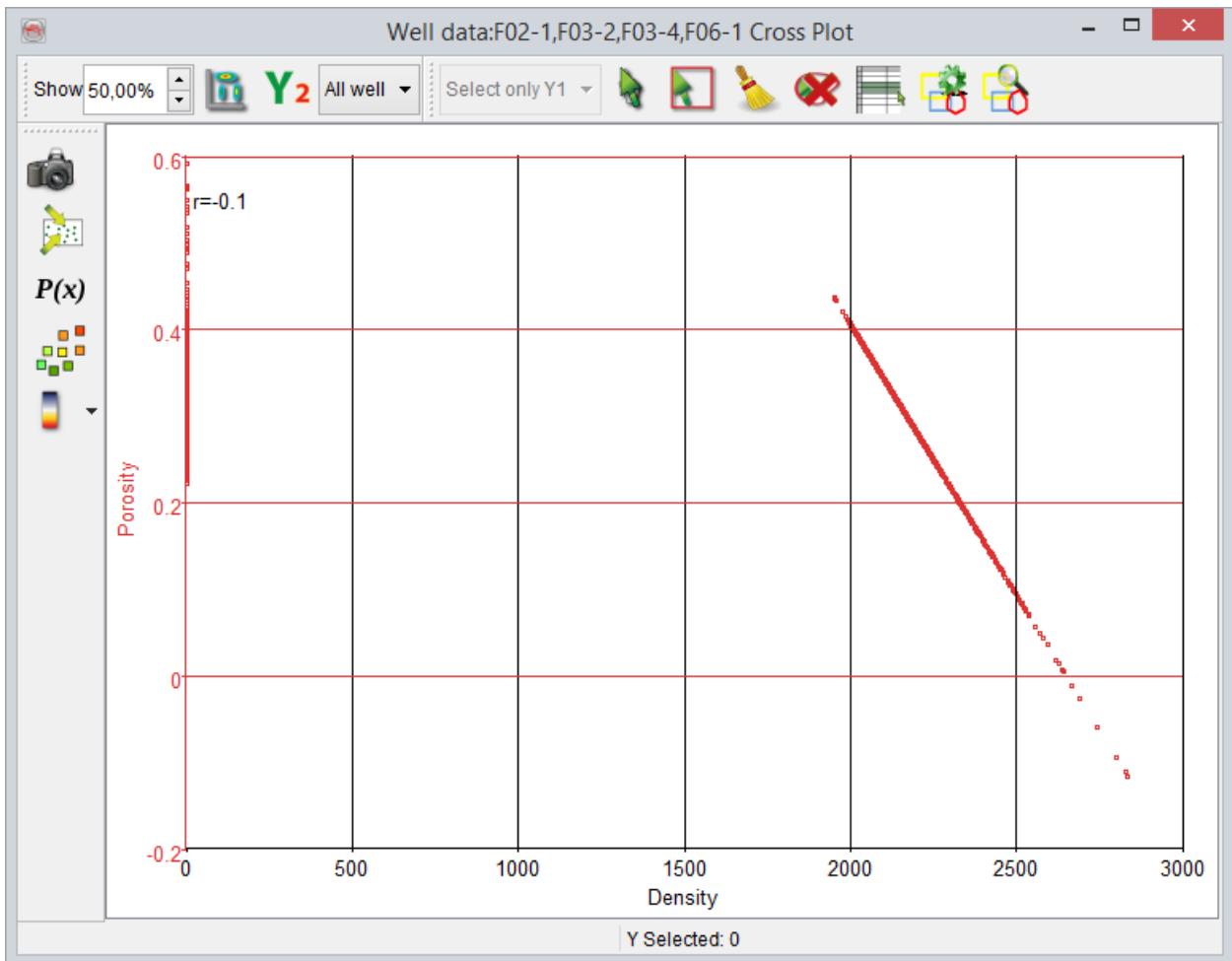
It is highly recommended, but not required, to click on Show Cross-Plot button in order to extract a table and graph properties of interest by clicking on the  icon which can help to modify automatically determined PDF parameters:

Well data analysis Cross-plot

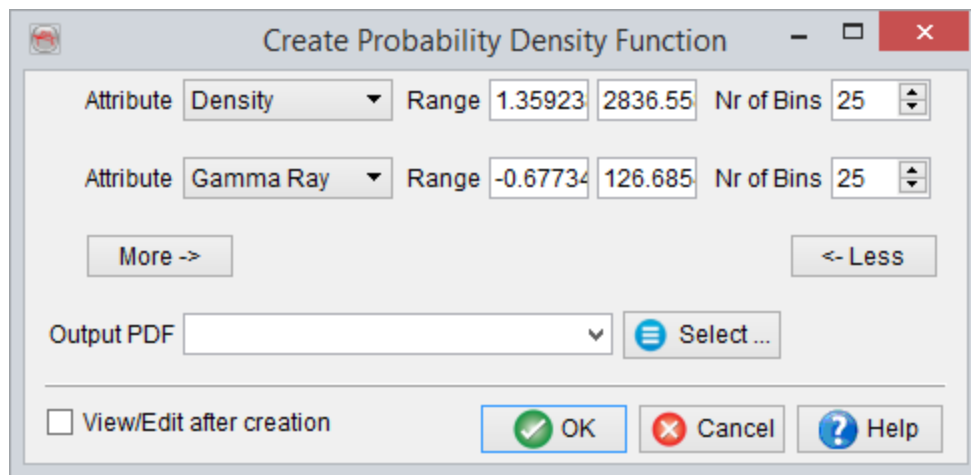
Well data:F02-1,F03-2,F03-4,F06-1


	X-Coord	Y-Coord	Z (ms)	* MD(m)	[X] Density	Gamma Ray	[Y] Porosity
F06-1	607909.85229611	6077190.66652...	0.48	29.29	2049.03564453		0.37560004
F03-4	623255.97999955	6082586.86999...	-2.25	30.15	2077.96264648		0.35752001
F03-4	623239.91391156	6082595.9375224	-2.25	30.15	2077.96264648		0.35752001
F03-4	623240.6119288	6082570.94786...	-1.11	32.15000153	2078.04907227		0.35746428
F03-4	623265.60218251	6082571.64629...	-1.11	32.15000153	2078.04907227		0.35746428
F06-1	607903.00000011	6077213	2.67	32.29	2049.16650391		0.37551543
F03-2	619120.9992515	6089483.38709...	3.39	33.15000153	2049.13134766	14.52392387	0.3755385
F03-2	619095.31098055	6089507.67831	3.39	33.15000153	2049.13134766	14.52392387	0.3755385

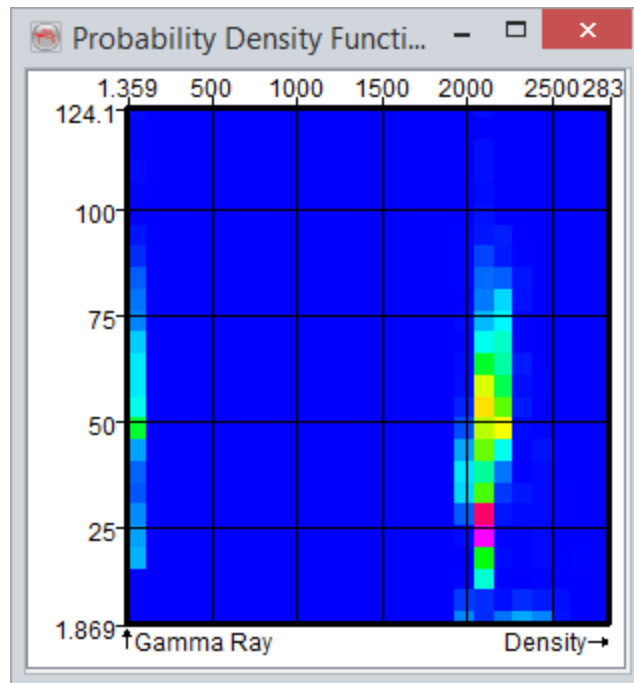
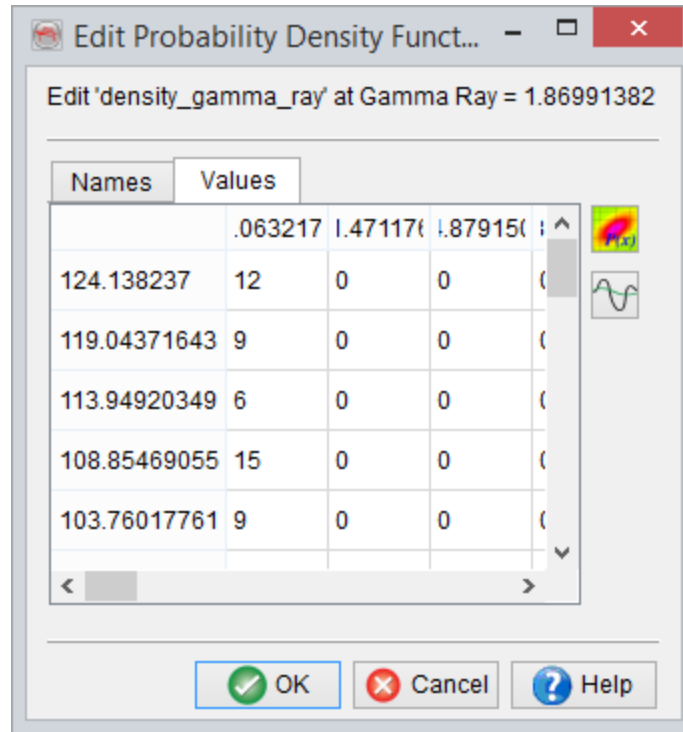
Close Help




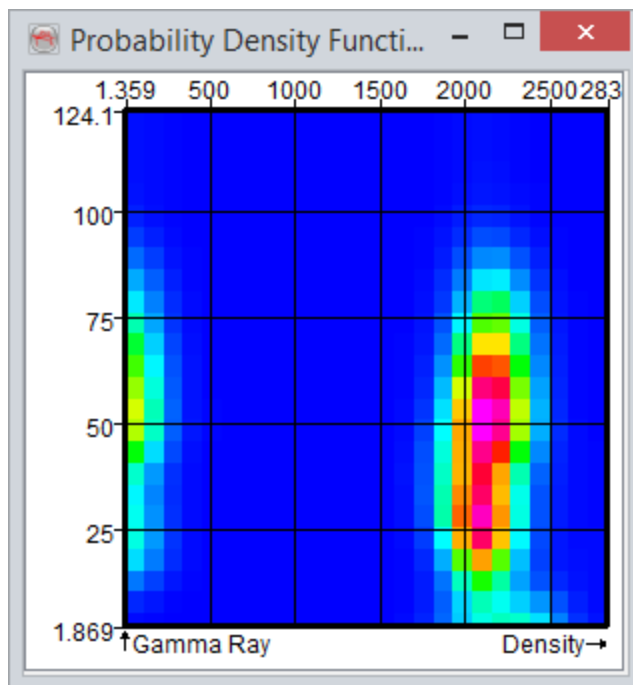
After clicking on the *Create PDF* button in *Well data analysis* window, the Create PDF window allows to define the number of dimensions, a property for each dimension, their ranges and number of bins. Remember that only up to three properties can be stored in a discrete PDF. Wells commonly contain a small number of data points, it is therefore recommended to check *View/Edit* after creation option.



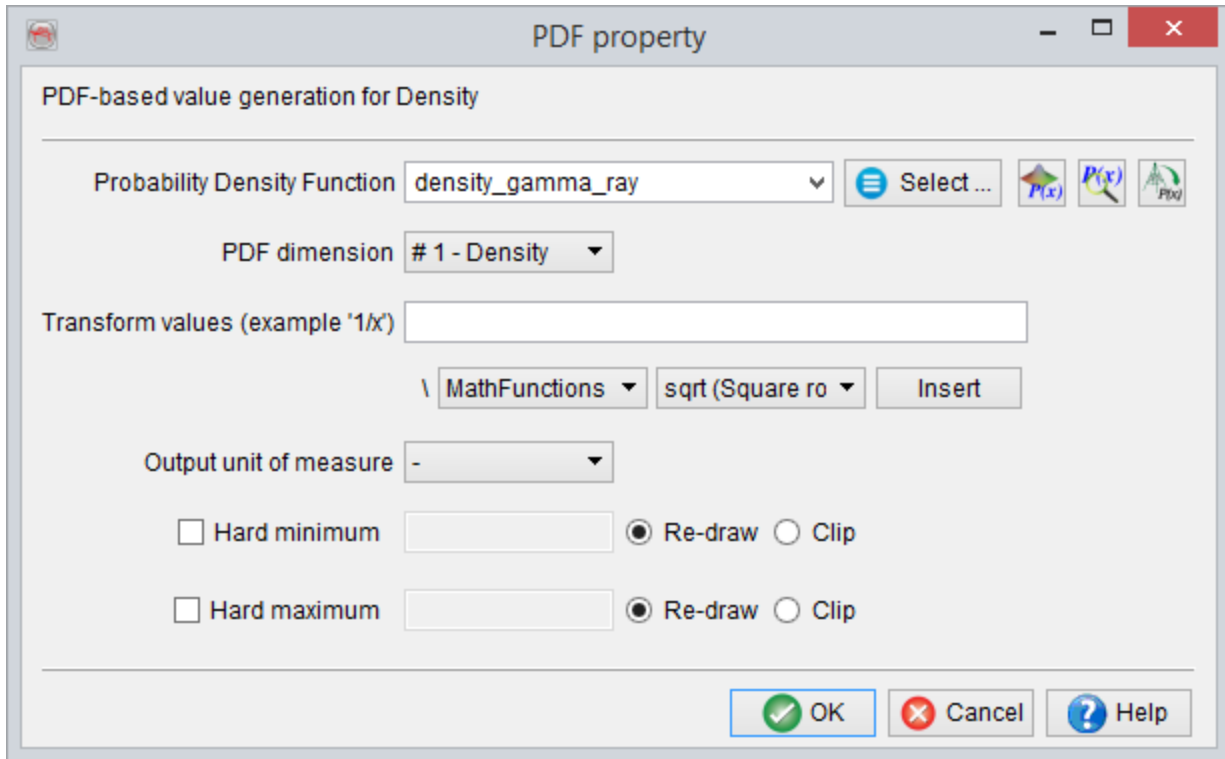
It allows to browse and edit a PDF in a table, as well as display it graphically by clicking on the  icon.



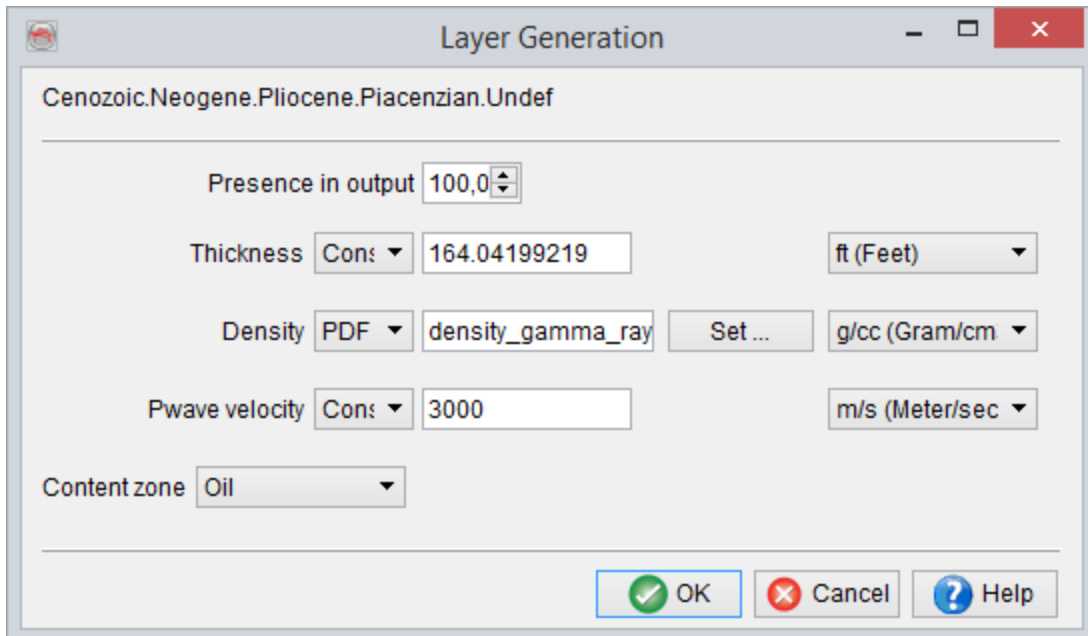
Editing of a PDF can be done manually in a table or smoothed by clicking on the  icon. For other details check the Manage PDF section of the OpendTect User Documentation



Upon clicking OK button in either the *Create PDF* or *Edit PDF* window, the derived PDF is automatically fed to the PDF property window. Stored PDF can have multiple dimensions, thus you will need to select what dimension to use. Consequently, a single PDF can be used to provide the input for multiple quantities. Also, a transformation can be applied on the values stored in the PDF: for example, to simulate sonic when the PDF contains P-wave, or to simulate $\log(K)$ when the distribution contains K (permeability).



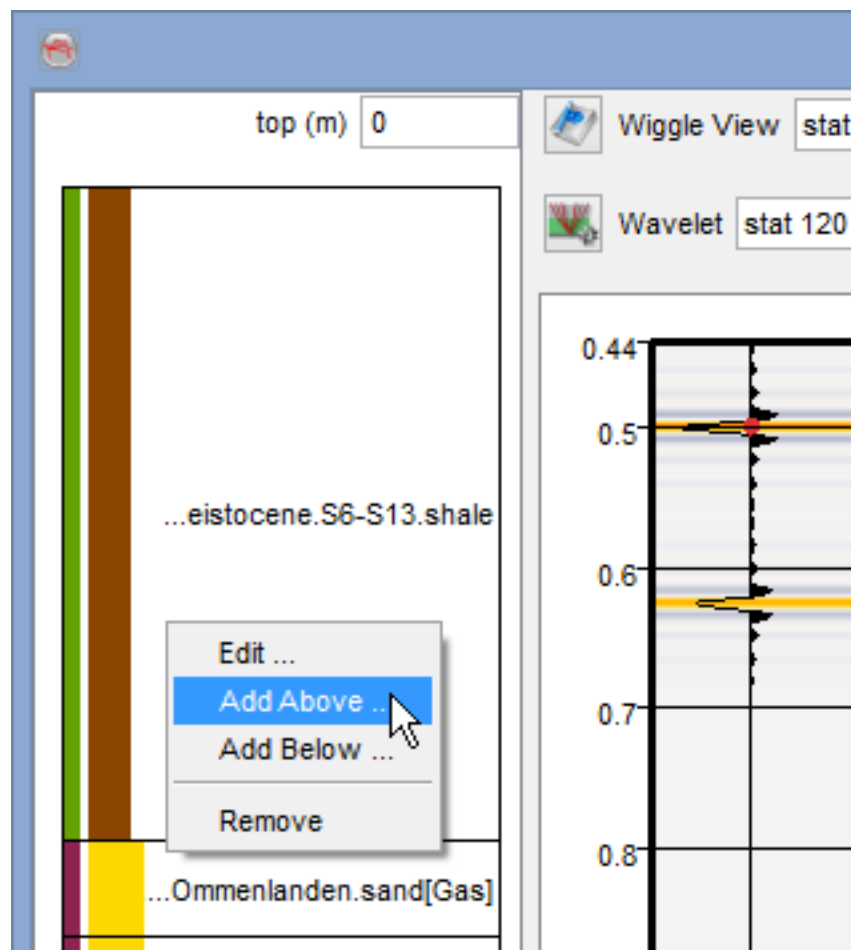
Finally, confirmation by clicking OK button in either the *Random property* or *PDF property* windows updates the corresponding fields in *Layer Generation window* as shown below for Density and Porosity.



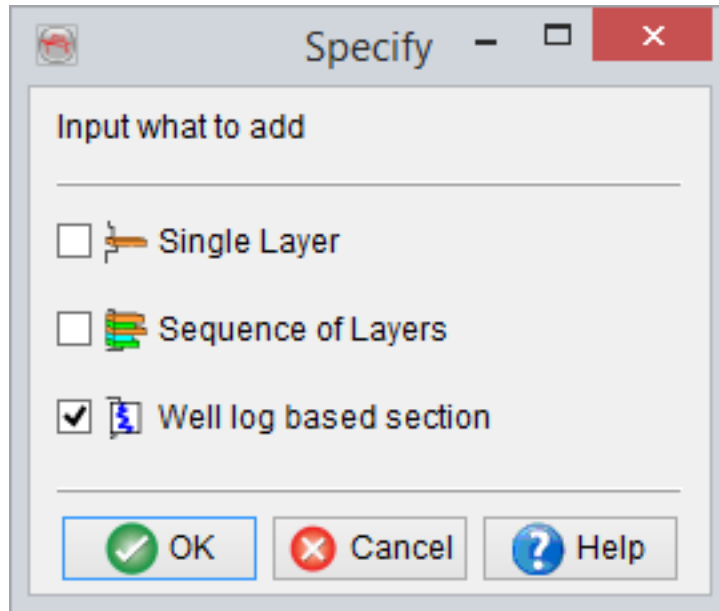
8.3.4 Well-Log - based Generation

Well-log based section can be either full or partly full of well log data, which has been upscaled/blocked at a constant size. This option is recommended for modeling a realistic overburden which is required for ray-tracing when working with prestack data.

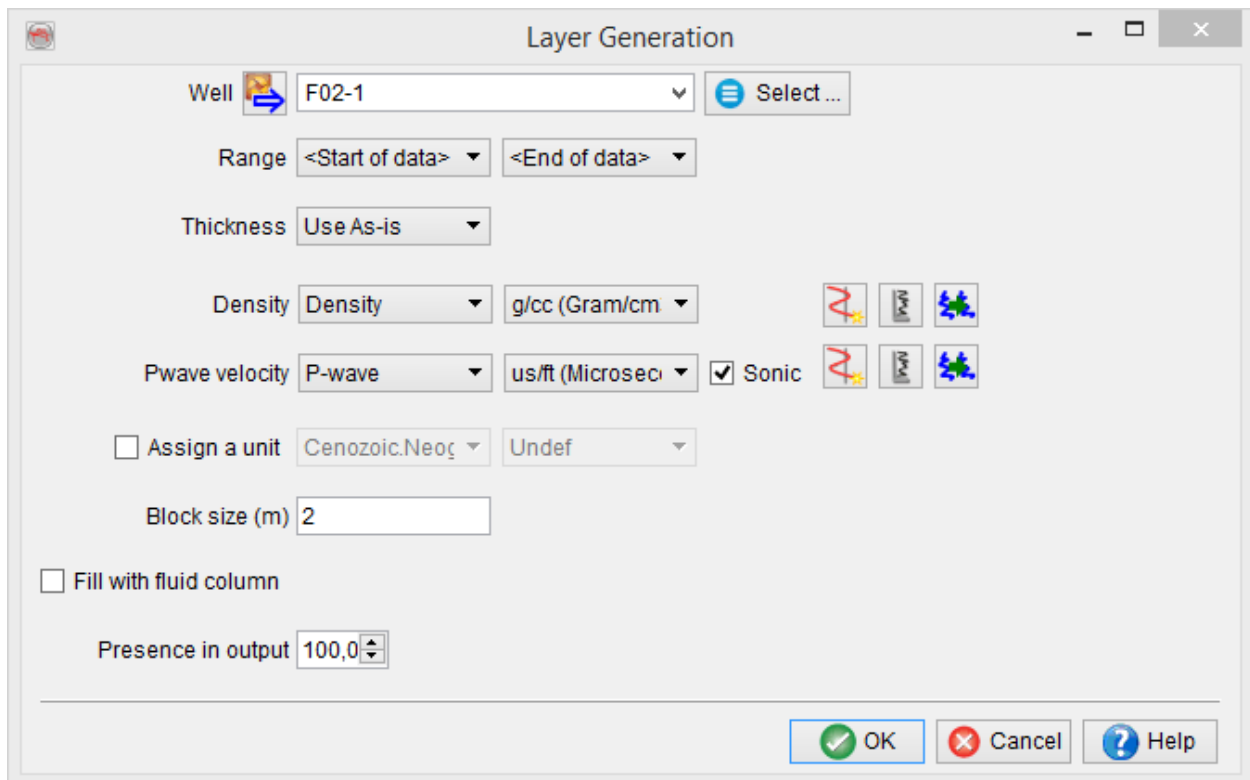
Right-clicking inside the *Layer Description* panel of the *Layer Modeling* window, will bring up the following popup:



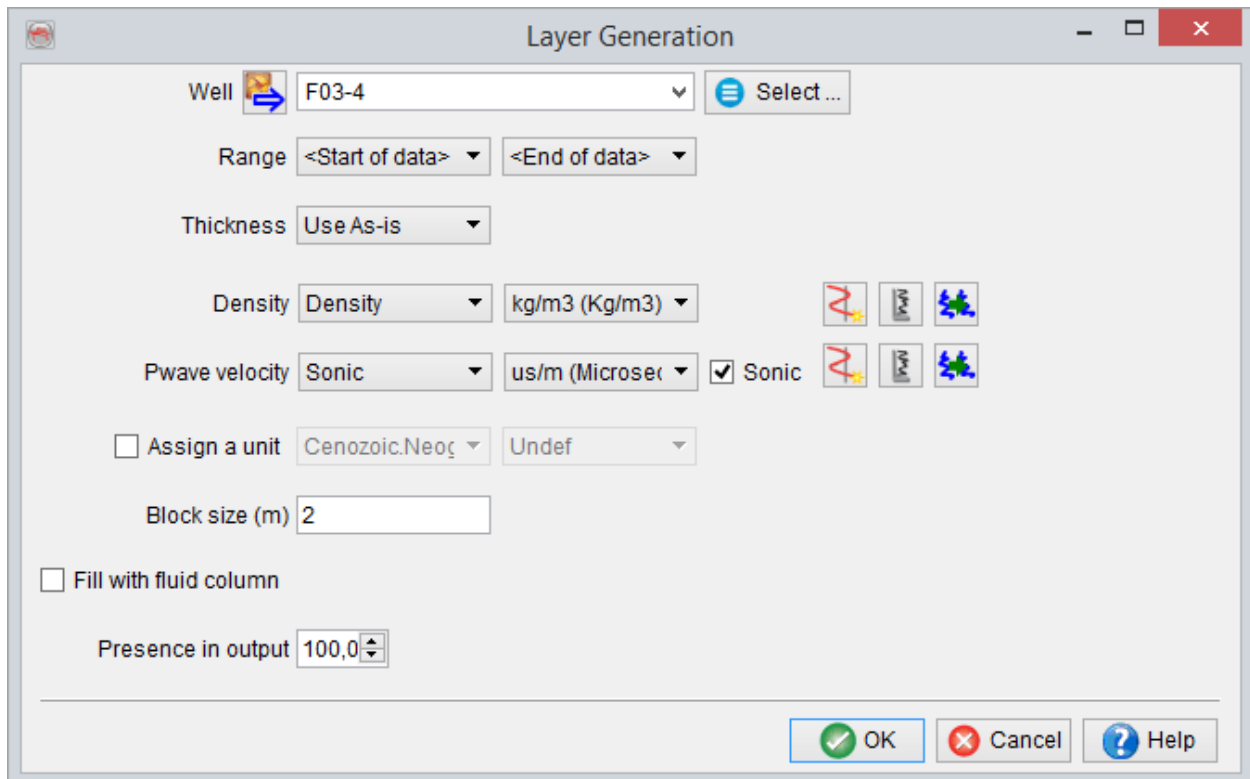
Clicking either 'Add Above...' or 'Add Below...' will bring up the following selection option:






Choosing '*Well-Log based section*' will bring you here:



And selecting a well will allow auto-input of any available logs. Well-log based section can be either full or part of well log data, which has been upscaled/blocked at a constant size. This option is recommended for modeling a realistic overburden which is required for ray-tracing when working with prestack data. See following example for F03-4:



The example above shows Layer Generation window for adding a well-log based section to a model description. Once the well is selected, define the interval bound by formation tops to be used. Thickness of the modeled interval can be Used as-is, or modified using one of the three options Cut from Top, Cut from Bottom or Stretch/Squeeze. If modified it can be set to a Constant, Range or Random (Uniform or Gaussian distribution). For example, Cut from Top in conjunction with Range or Random can be used to model subcropping strata, and Stretch/Squeeze along with Range or Random - to model thickness variations.

The next section of this window allows to choose well logs corresponding to the layer properties used in the modeling. Note that all used properties are required to have a log selected. Missing logs can be either created () without having to exit SynthRock or imported using the Well Manager. Selected log can be viewed (), or changed ().

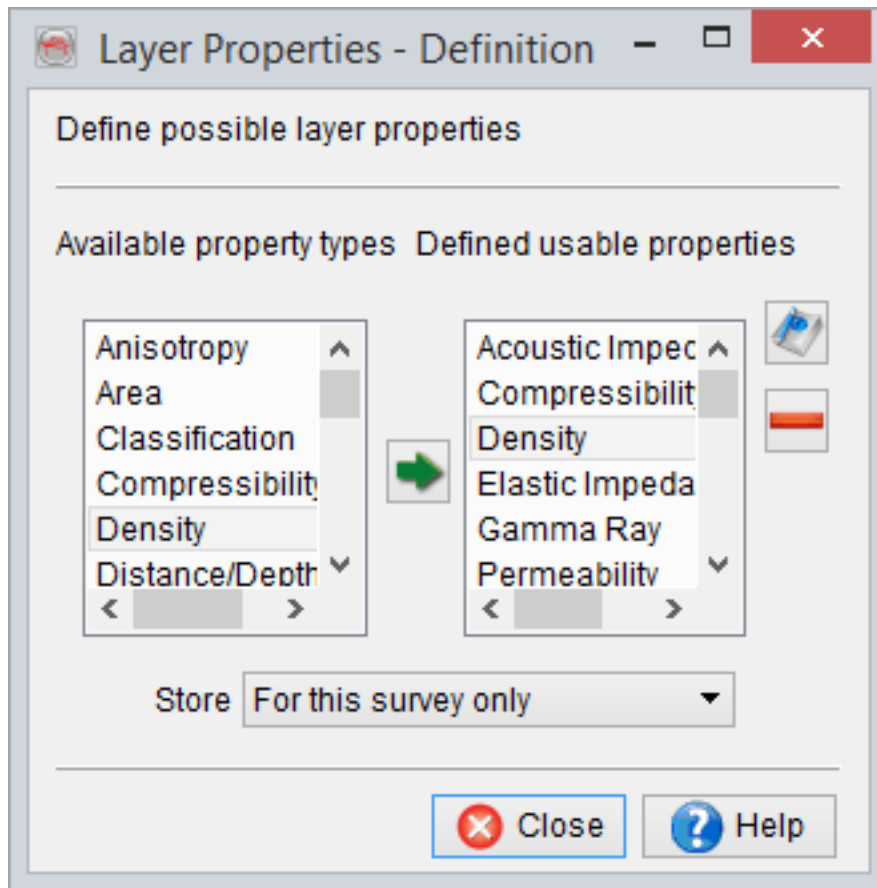
Optionally, a well-log section can be assigned to a specific unit by selecting both formation and lithology member. If no unit is assigned, a name derived from bounding formation tops is displayed in the modeling description part of the Layer modeling window, but an actual layer name as well as lithology are left undefined ('-'). Default colors for undefined formations and lithologies are white and grey respectively. Note that if no unit name is assigned, the subsequently created layers can not be used for plotting Attributes vs model properties or be a target quantity during the HitCube inversion.

All the logs are upscaled/blocked at the user-defined Block size. Note, that when modeling large intervals, Block size should be chosen with care as block sizes which are too small significantly increase time to generate pseudowells and memory consumption.

If Fill with fluid column is checked, the fluid column has to be Defined. Presence in output controls the percentage of wells having the given well-log based section.


8.3.5 Manage layer Properties

This window is also accessible from the *Layer property selection* window in the Layer Modeling module. In this manager, different usable layer properties are listed with their corresponding type. For example Delta and Epsilon properties belong to the Anisotropy type. Click on the usable property in the right column to see on the left column what is the type associated (highlighted).

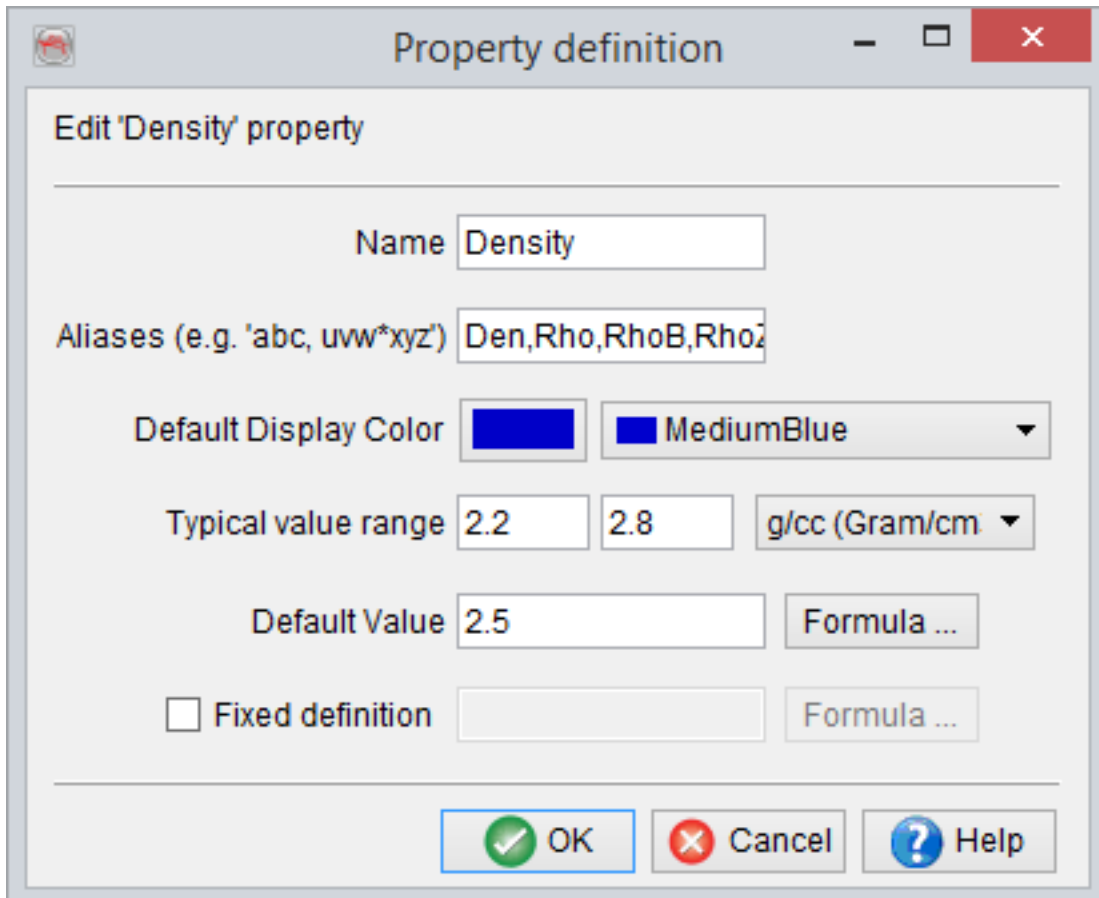


The two lists are hardcoded. However the user can ask for adding an extra property or type in sending an email to [support](#). Usable properties listed can be removed of the list if they have no use in the current project.

Define/Edit usable properties

One usable property can correspond only to one property type. For changing the property type of a usable property, the usable property needs to be removed and added again in association to the selected property type. To add a usable property, select a property type and click on the  in between the two list : it opens the property definition window. For each property needs a name. Possible aliases can be specified. It is useful to associate the correct log to a property: logs with different names can thus be related to the same property. Default display parameters are set up : the colour and the typical range value with the associated units.


Already defined usable properties can also be edited on clicking on the  icon.

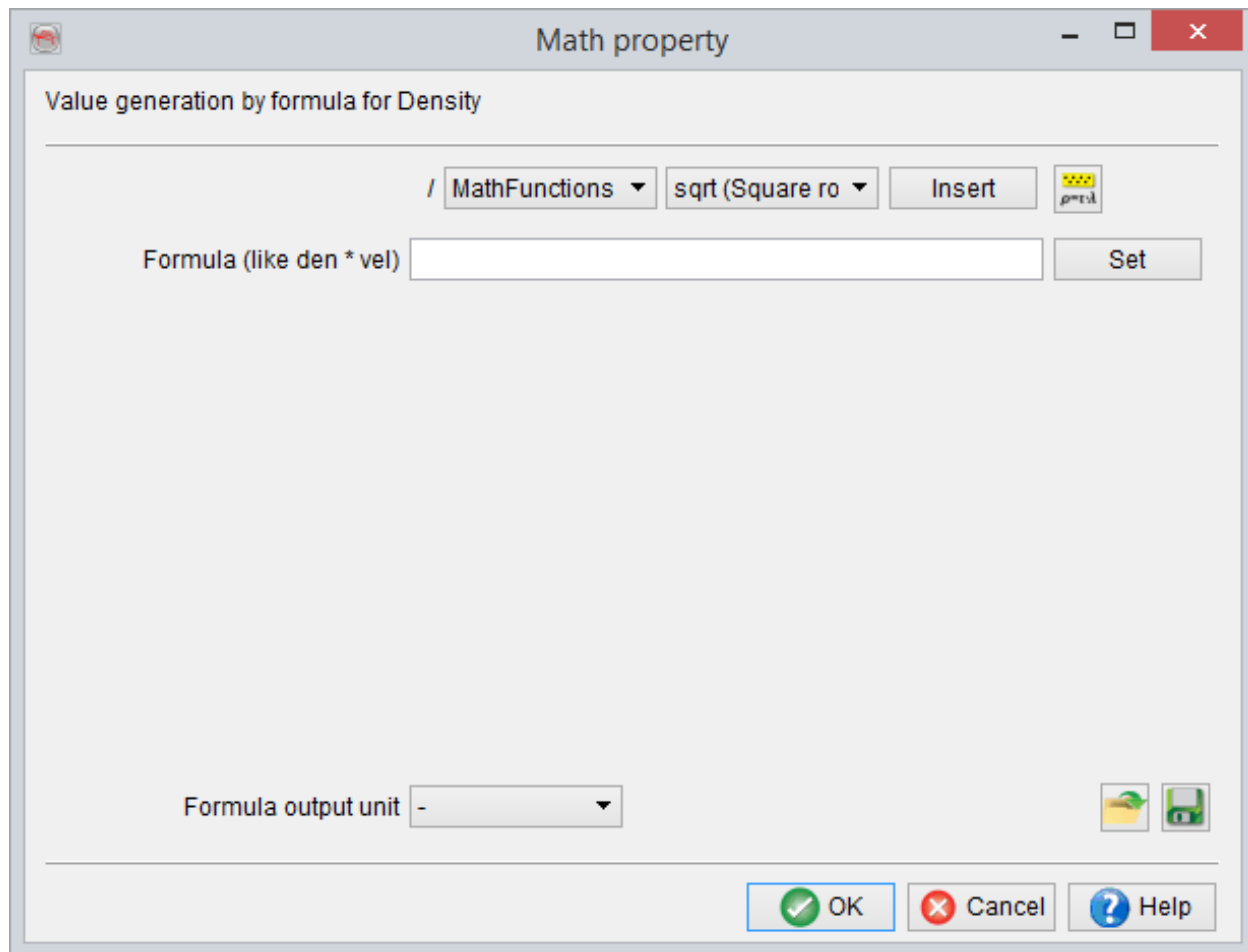




The screenshot shows a dialog box titled "Property definition" with a close button (X) in the top right corner. The main content area is titled "Edit 'Density' property". It contains several input fields and controls:

- Name:** A text box containing "Density".
- Aliases (e.g. 'abc, uw*xyz'):** A text box containing "Den,Rho,RhoB,Rho2".
- Default Display Color:** A color selection control showing a blue swatch and a dropdown menu with "MediumBlue" selected.
- Typical value range:** Two text boxes containing "2.2" and "2.8", followed by a dropdown menu showing "g/cc (Gram/cm" with a downward arrow.
- Default Value:** A text box containing "2.5" and a button labeled "Formula ...".
- Fixed definition:** A checkbox that is unchecked, followed by an empty text box and a button labeled "Formula ...".

At the bottom of the dialog, there are three buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a blue question mark icon).

Optionally, the default value of any property can be defined by a specific equation that can either be typed in or retrieved from a list of saved formulas on clicking on *Formula...* Further formulas and equations may be utilized from the RockPhysics library, reached via the  icon.



The edited Layer Properties are saved when clicking on *Ok*. There are also options here to either *Load*  or *Save*  Maths Formulas.

Fixed Definition: A property that will never be modeled directly but will be auto-computed in the background from other modeled properties. Formula for such properties

always remain consistent, irrespective of any geological setting: Acoustic Impedance, Shear Impedance, Vp/Vs Ratio, Poisson's Ratio, Lambda-Rho, Mu-Rho etc.

The 'fixed definition' for these properties can be specified by ticking the 'Fixed definition' box and clicking on the Formula button, bringing up the RockPhysics library:

The 'Property definition' dialog box is titled 'Property definition' and contains the following fields and controls:

- Add 'Elastic Ratio' property**
- Name:** Poissons Ratio
- Aliases (e.g. 'abc, uvw*xyz'):** Poi,PR
- Default Display Color:** MediumOrchid
- Typical value range:** 0.1, 0.5, (Ratio)
- Default Value:** 0.3
- Fixed definition:** $((Vp/Vs/Vp/Vs/Vp) ^ 2 - 2)$ Formula ...
- Buttons:** OK, Cancel, Help

The 'Math property' dialog box is titled 'Math property' and contains the following fields and controls:

- Value generation by formula for Elastic Ratio**
- MathFunctions:** sqrt (Square ro) Insert
- Formula (like den * vel):** $((Vp/Vs) ^ 2 - 2) / (2 * ((Vp/Vs) ^ 2 -))$ Set
- For 'Vp' use:** Poissons Ratic convert to:
- For 'Vs' use:** Poissons Ratic convert to:


Storing levels

The level where it is saved is specified in the main window:

- For this survey only (default): save at the root of the survey, applicable for all users only for this survey (if no parameters saved for the user).
- As default for all surveys: save in dTect/data (where are all the surveys), applicable for all users and all surveys (if no parameters saved for the user or this survey)
- As default for my user ID only: save in home/.od. If it exists, it has the priority on the two others.

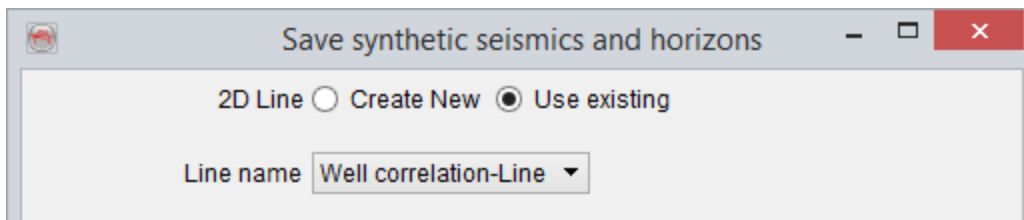
Once the parameters saved, the only way to access the default parameters provided by OpendTect is to delete the Properties file where it is located. OpendTect will then access its default properties file.

8.3.6 Export Synthetic Dataset

The synthetic seismic data (both poststack and prestack), the layer property synthetics in Time (e.g. AI, Density etc.) and the stratigraphic levels/markers, from all modeling modules (i.e. Basic, Profile and Stochastic) can be exported along 2D lines. The stratigraphic levels/markers in the modeled pseudo-wells are essentially exported as 2D horizons. This is achieved by clicking on the  icon at the top right of the modeling window (see below).



Export of the synthetics can be done onto an already existing 2D line or a new line created on the fly.



Selecting an existing 2D line

If the 2D line is created on the fly, the *Geometry for line* has to be defined as well. It can be done by defining a straight line between two X-Y coordinate pairs.

Save synthetic seismics and horizons

2D Line Create New Use existing

New Line Name

Geometry for line

Coordinates: from

to

Creating a straight 2D line between two X-Y coordinate pairs

The 2D line can also be created, on the fly, along an existing polygon.

Save synthetic seismics and horizons

2D Line Create New Use existing

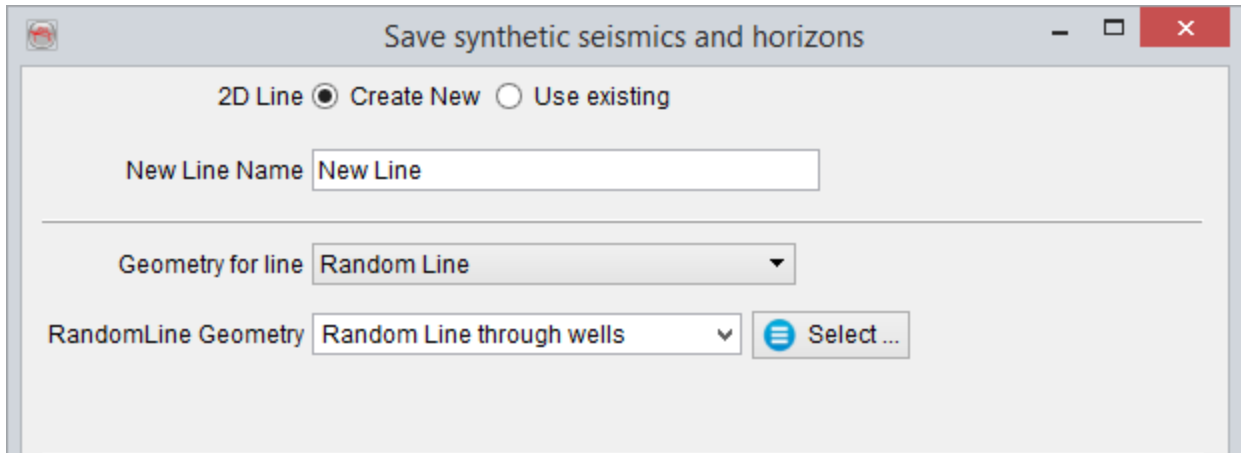
New Line Name

Geometry for line

Polygon

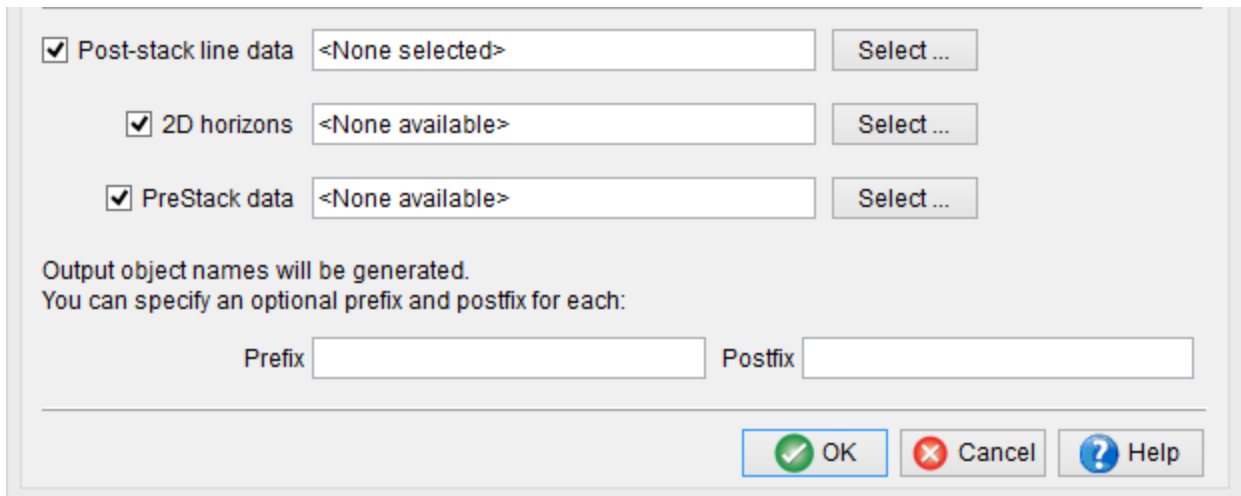
Creating a 2D line along a polygon

Finally, the 2D line can also be created along an existing random line.

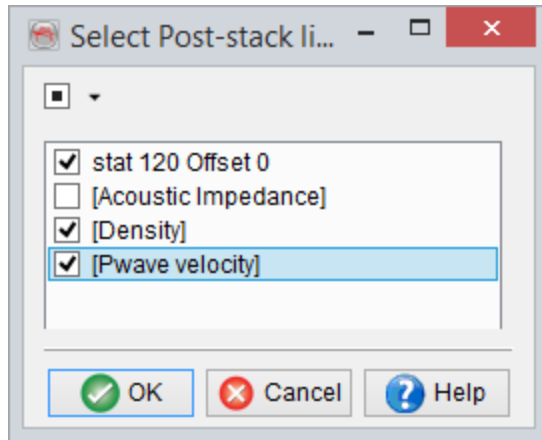


Creating a 2D line along a random line

Now, a selection on poststack data, 2D horizons and prestack data can be made for exporting along the 2D line.

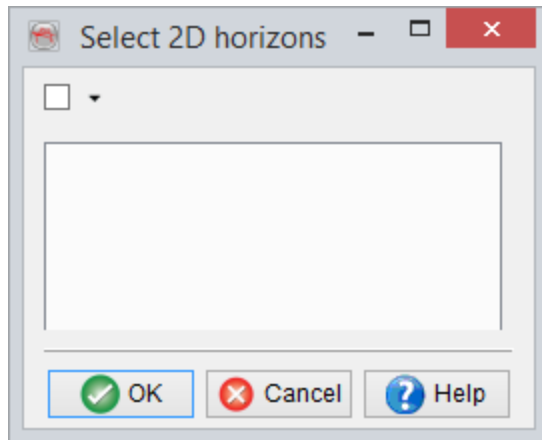


For poststack data, user can select synthetic seismic and various layer property synthetics (e.g. Acoustic Impedance, Density etc.).



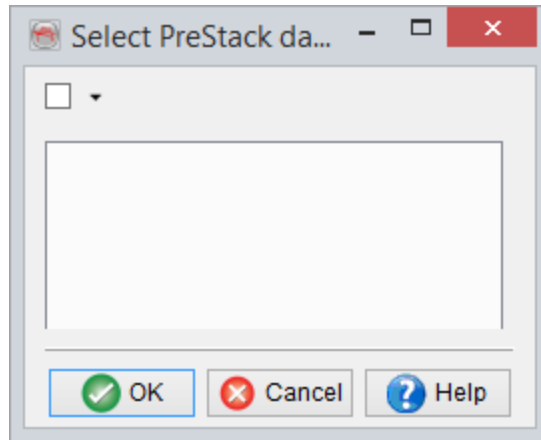
Poststack data selection for export

Similarly for 2D horizons, various levels present in the pseudo-wells can be selected.



2D horizon data selection for export

and finally (if any) prestack data can be selected.

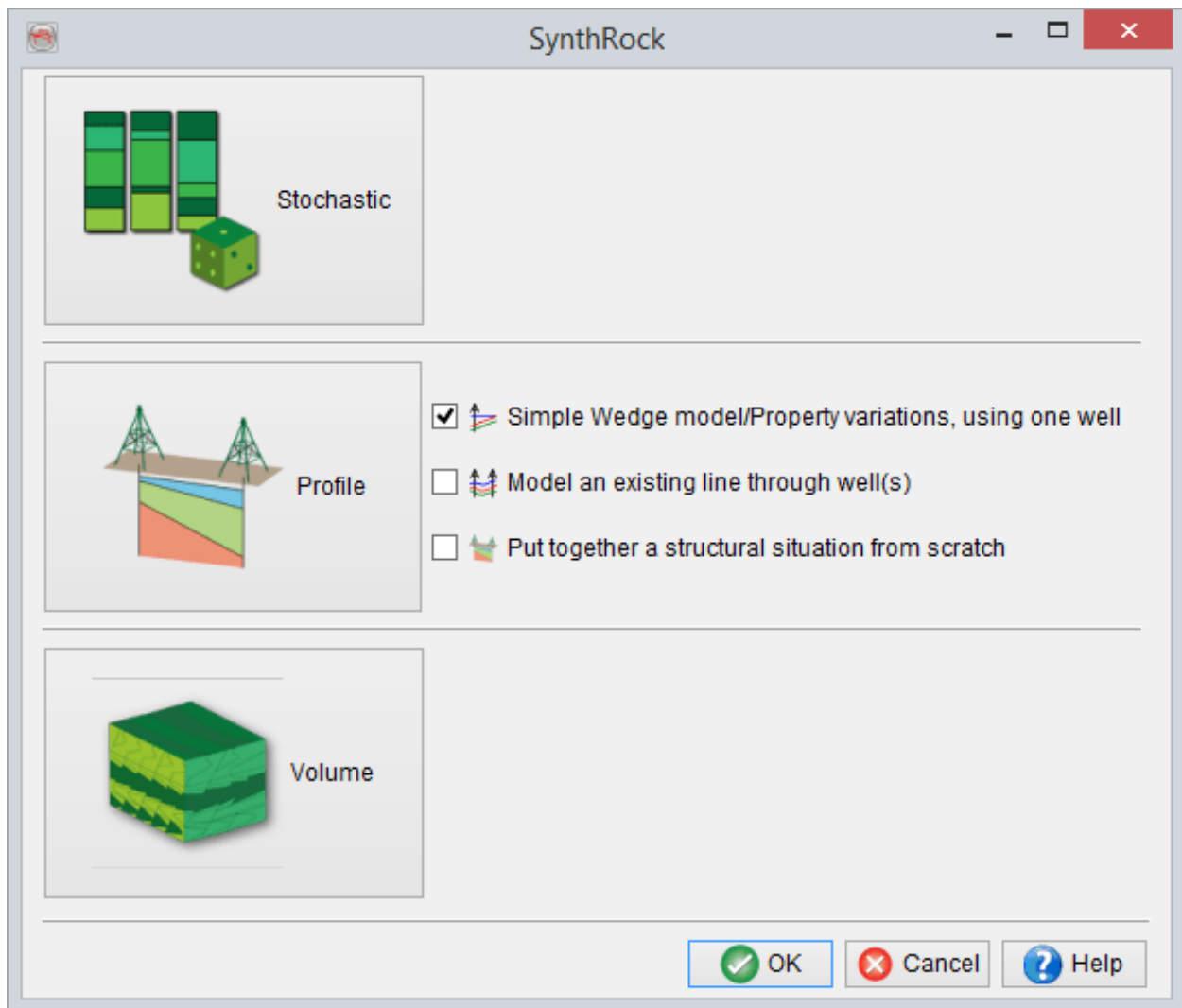


Prestack data selection for export

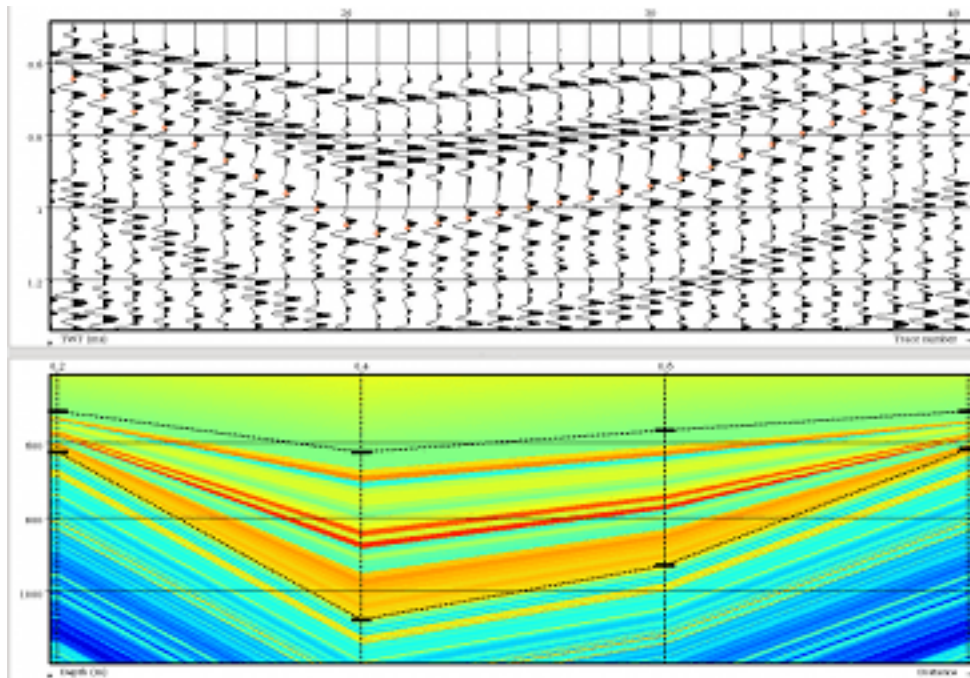
Optionally, a prefix and/or postfix can be specified for various data items. Pressing *Ok* will export the selected data items along the 2D line.

8.4 Profile Modeling

The profile modeling is used to create pseudowells by interpolating well logs between control points. The control points are inserted from real wells loaded in the project, with well-log data upscaled/blocked into layers of user-defined size, or added as control profiles during the modeling.



The modeling is performed in two steps by adding the control points and modifying the model parameters . An output example is given below. The modeling itself takes place in the bottom part of Layer modeling window. The modeling description can be saved, while interpolated pseudowells are not saved along with it. The pseudowells can be either dumped to a text file, or the corresponding data can be extracted using the cross-plot extractor. The pseudowells are re-drawn by pressing on button. The number on the left of button defines the number of pseudowells to be created. It is advised to start with a relatively low number while building the model, and then increase it for the final application (several hundreds for a HitCube inversion are needed).



Profile modeling offers three options to start modeling

Simple Wedge model/Property variations, using one well and Model an existing line through well(s) are two pre-defined workflows that allow user to quickly set up most commonly used models. These workflows are pre-defined in a sense that a user is guided through the required sequence of windows by the software. They allow to build

simple models much faster than doing so from scratch. Once a simple model is built, further modeling can be continued using the classic Profile mode functionality in the Layer modeling window.

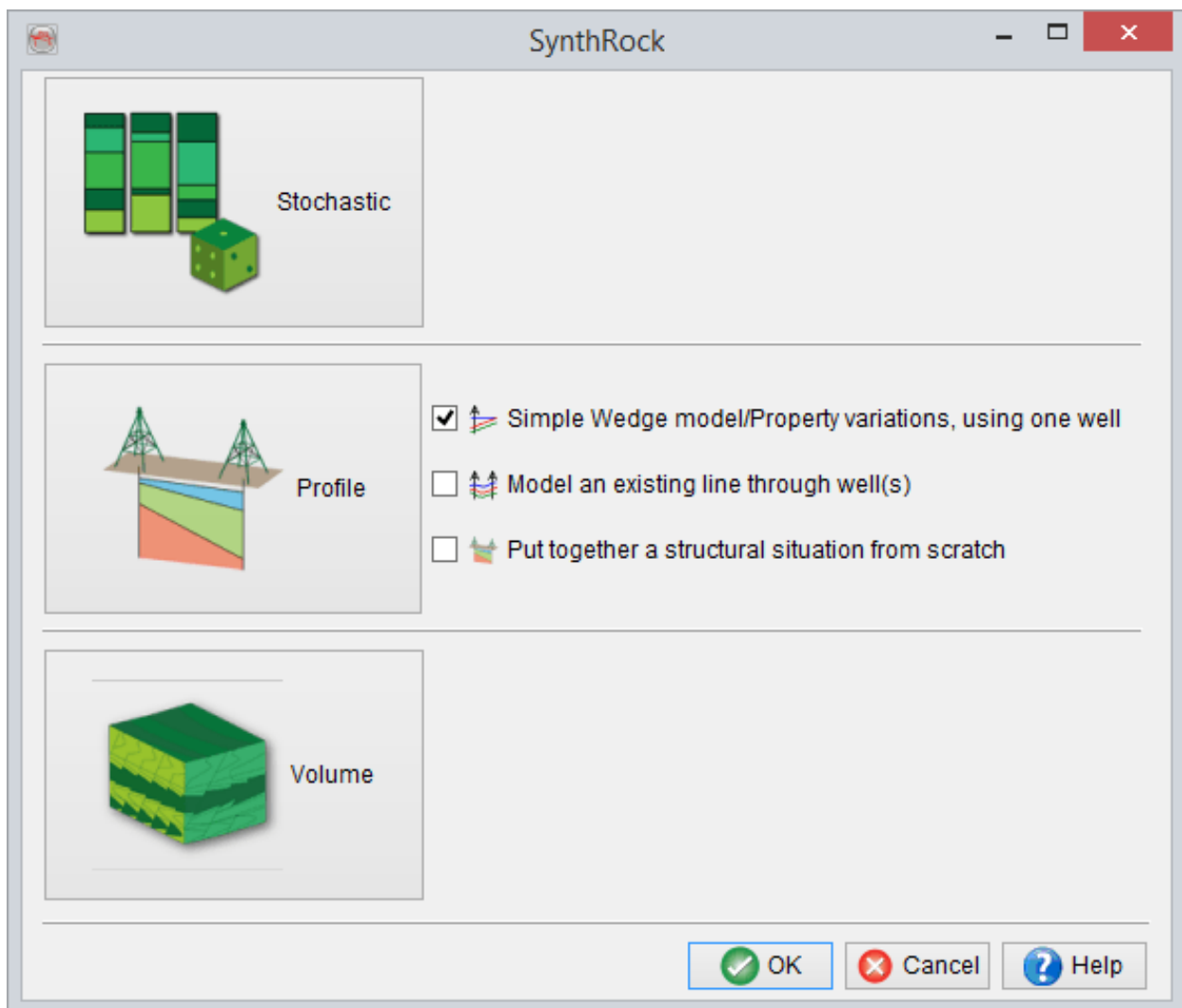
Note, that some windows in the pre-defined workflows are unique and might not be accessible from Layer modeling window if accidentally closed by a user. In such case, a user either have to continue in the Layer modeling window with the classic Profile mode functionality, which allows to do the same things but requires more user input, or start the pre-defined workflow again.



The third option, the classic Profile mode, is accessible either by clicking on the large Profile icon or by selecting Put together a structural situation from scratch option and clicking OK.

- [Simple Wedge model/Property variations, using one well](#)
- [Model an existing line through well\(s\)](#)
- [Put together a structural situation from scratch](#)

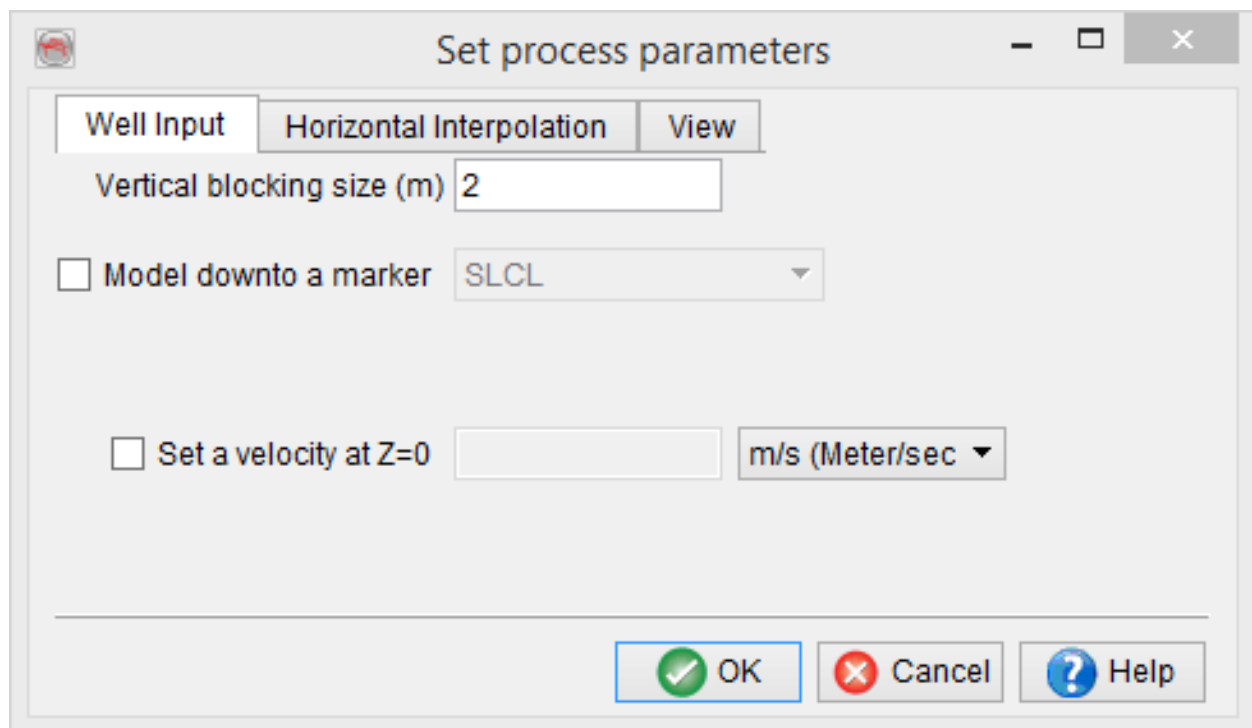
8.4.1 Simple Wedge Model/Property Variations Using one Well

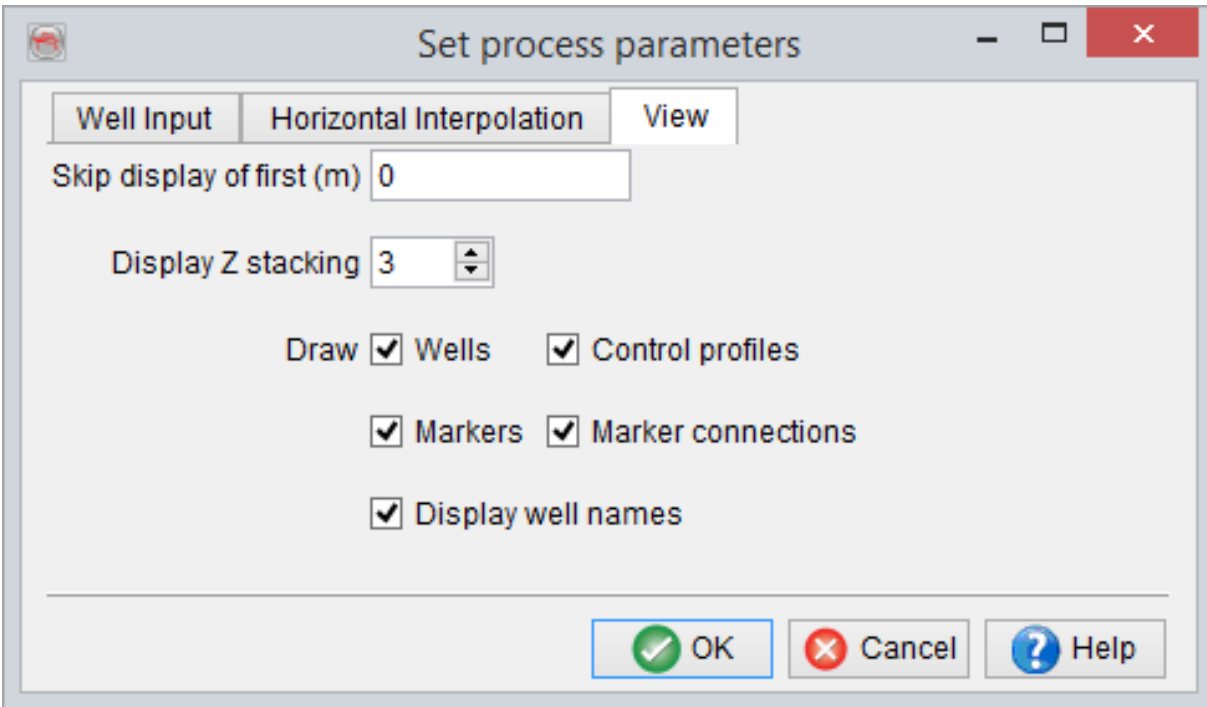
Simple Wedge model/Property variations, using one well is the first pre-defined workflow in SynthRock profile modeling which allows to quickly build a simple wedge model and/or model lateral property variations. It is launched by checking the corresponding box in *SynthRock* start window and clicking OK.



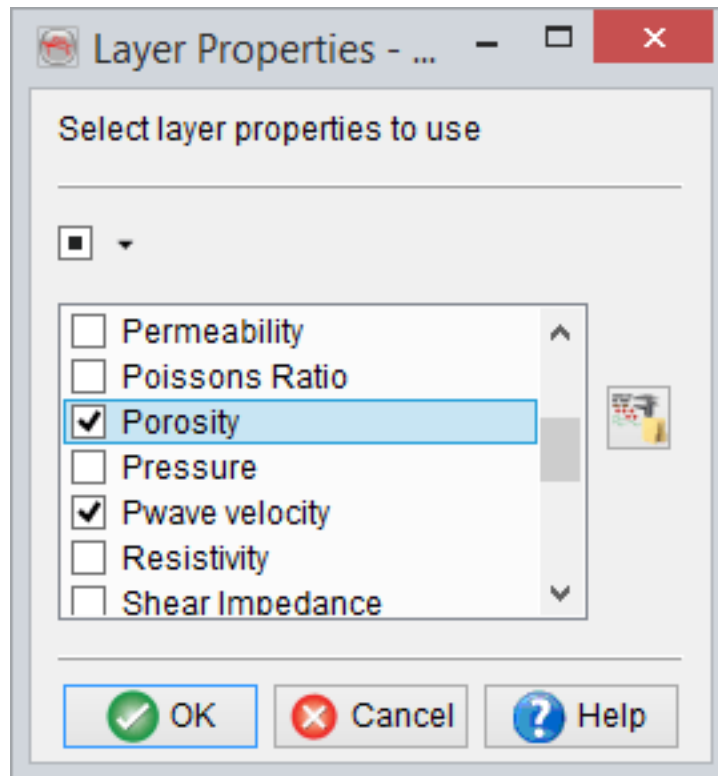
A user is expected to go through the pre-defined sequence of windows. The workflow starts, as any profile modeling in SynthRock, with setting modeling parameters in Set process parameters window and selecting properties to be modeled in Layer Properties - Selection window. Once the workflow is completed, both sets of parameters can be modified by clicking on  and  icons of the *Layer modeling window* respectively.



In the example below, few changes of the default parameters are made in the first window in order to limit the modeled interval.

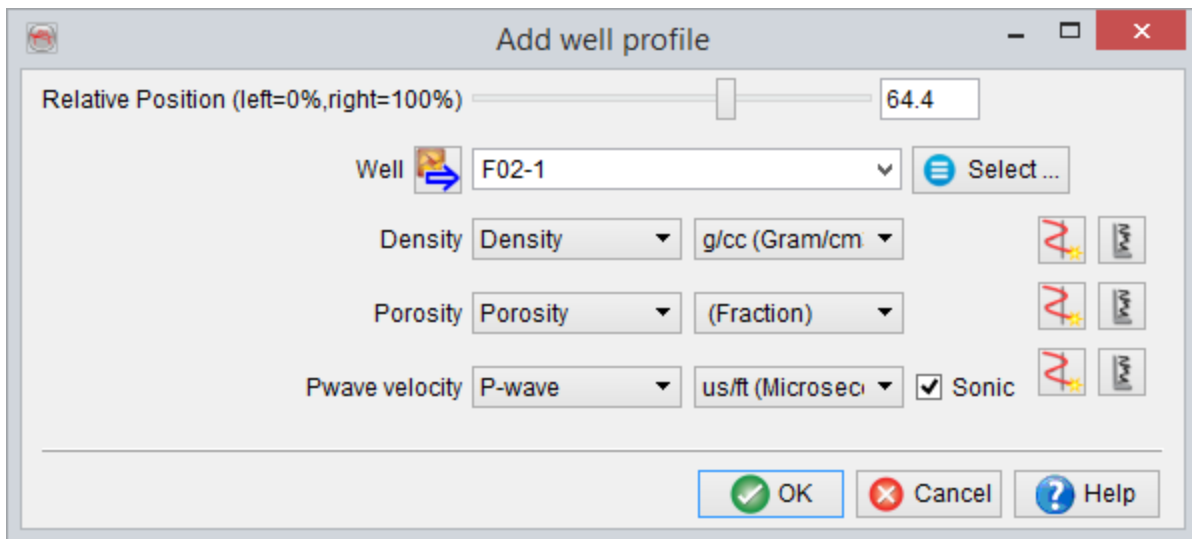




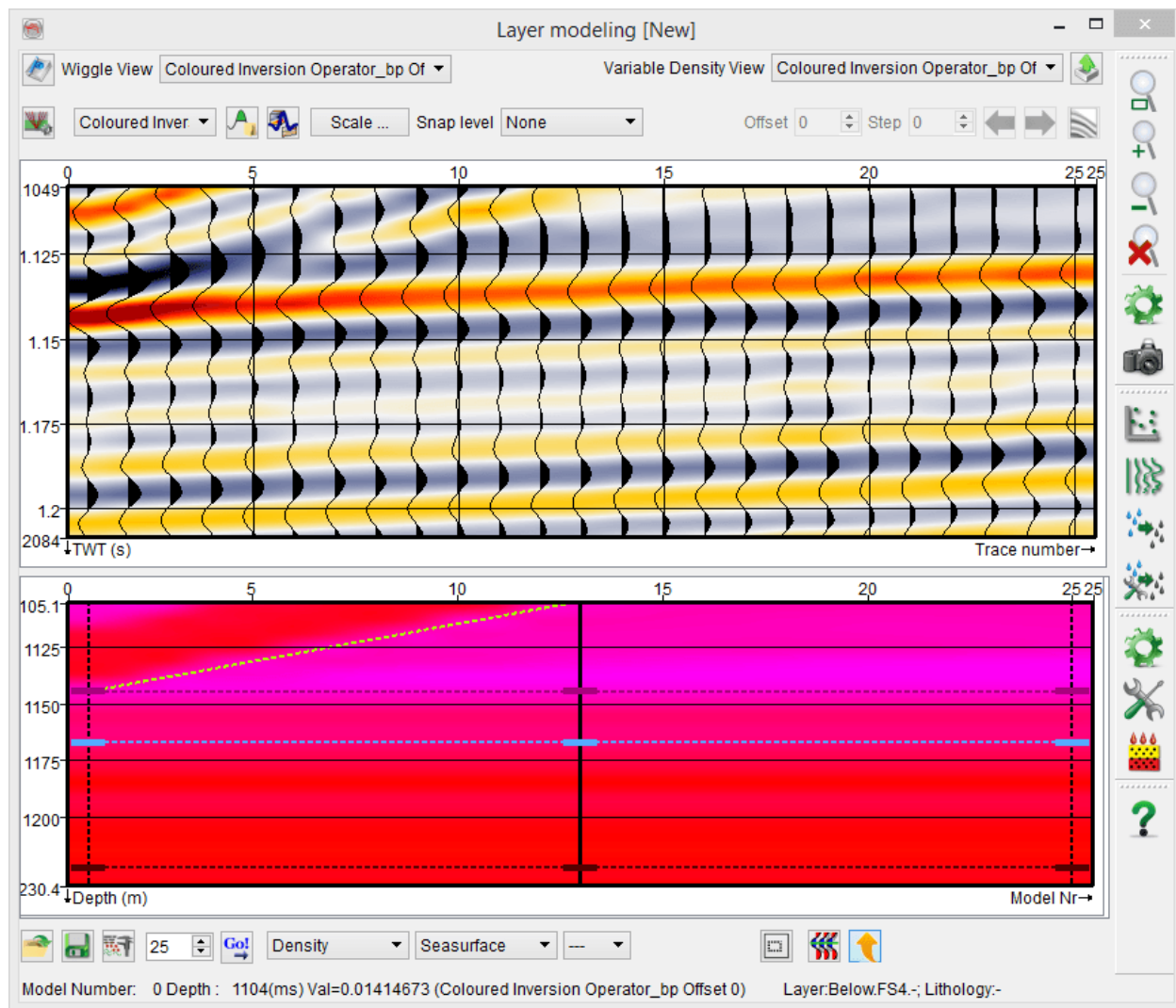
Porosity is added in addition to the default properties in the second window.



Once the modeling parameters and the property list are confirmed, Add well window allows to choose an existing well from a project database. In the example below *F02-1* is selected. An attempt to automatically determine well logs matching the modeled properties is made by the software, but it is recommended to always confirm the correct logs are selected. Note, that all properties chosen in *Layer Properties - Selection window* are required to have a log selected. Missing logs can be either created by clicking on the  icon without having to exit SynthRock or imported using Well Manager. Selected log can be displayed by clicking on the  icon.



Before proceeding further observe changes in the *Layer modeling window* after clicking on the *Apply* button. A default number of pseudowells has been created, which are in fact simply duplicates of the selected well with logs upscaled according to Set process parameters window. The original well is positioned at the far-left side of the profile and displayed as a solid vertical line. The control profile is positioned at the far-right side of the profile and displayed as a dashed vertical line. Corresponding synthetic seismic section has been generated using the chosen wavelet and displayed in the top part of *Layer modeling window*.

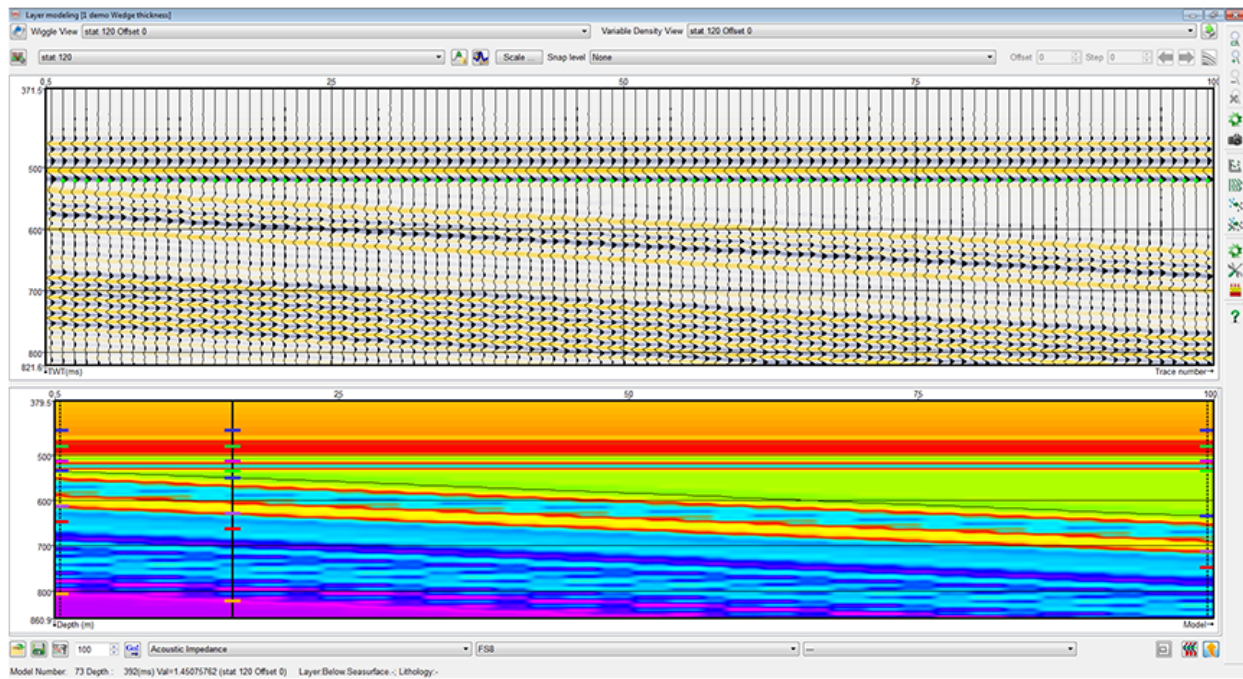
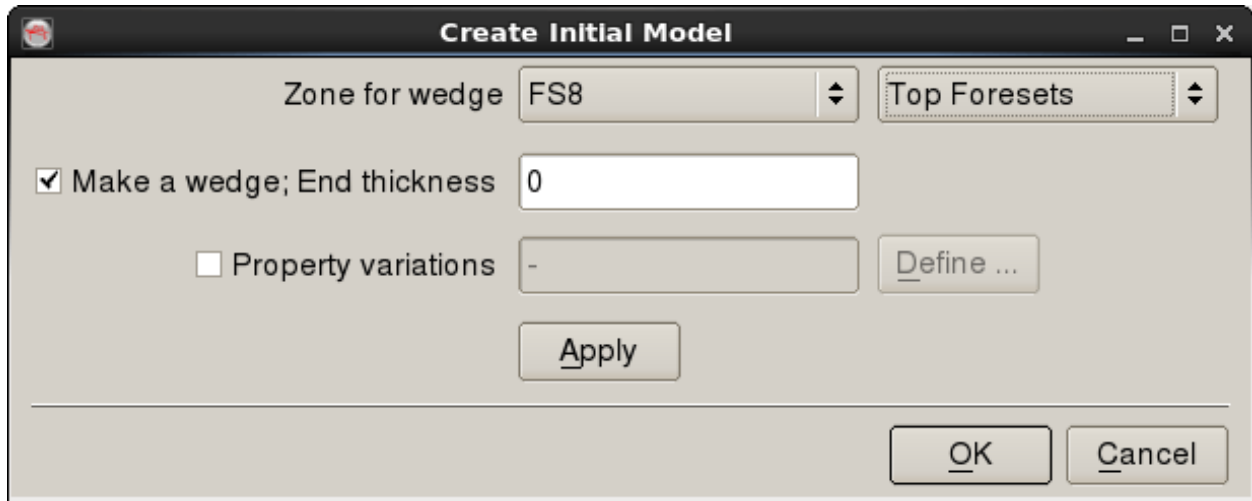


Clicking on OK button prompts to *Create Initial Model* window where a wedge thickness change and/or lateral property variations can be defined. First, a user must specify a zone by choosing existing well tops.

A wedge model

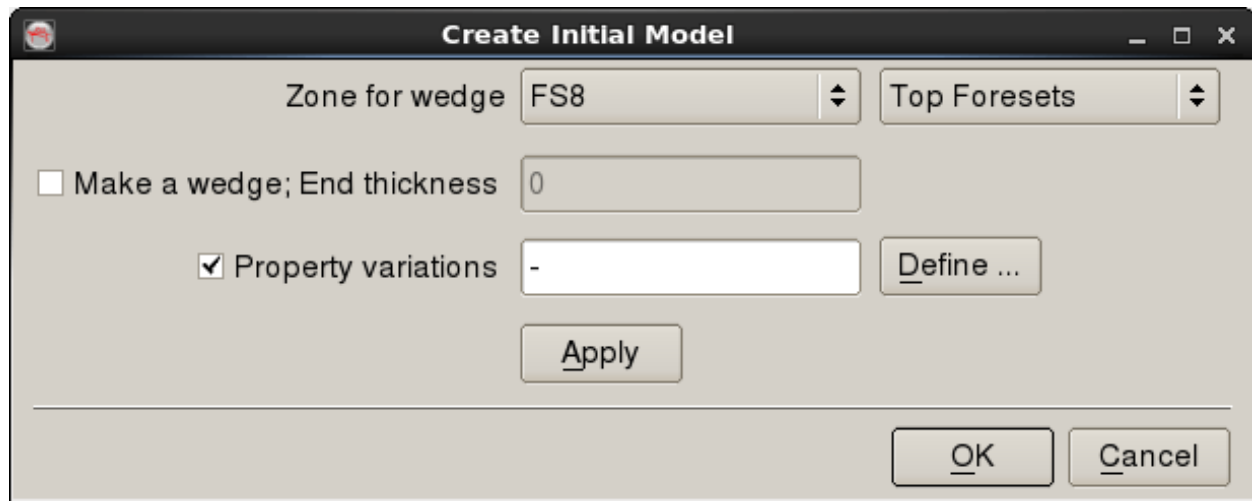
If thickness changes are modeled in the selected zone, *Make a wedge* option has to be checked, and *End thickness*, corresponding to the control profile at far right, has to be set. In the example below, wedge model of the *FS8 - Top Foresets* interval is defined

by varying its thickness from the original value at the well (128 m) to 0 m. Clicking OK generates the defined model.

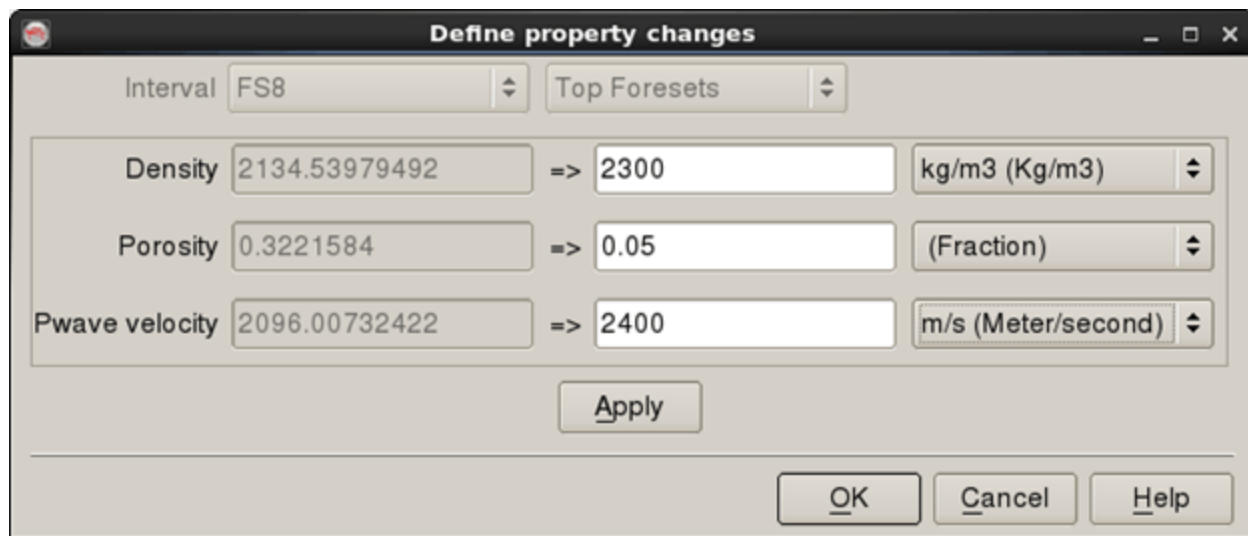


Define property changes

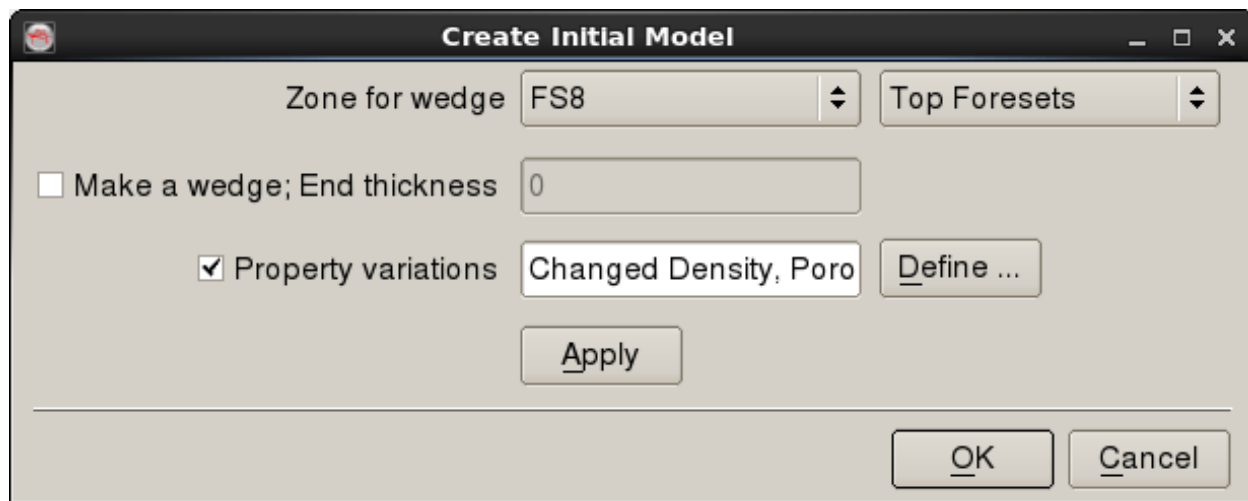
If property changes are modeled alone, *Property variations* option must be checked and *Make a wedge* option must be unchecked. *Property variations* have to be specified by clicking on Define button.

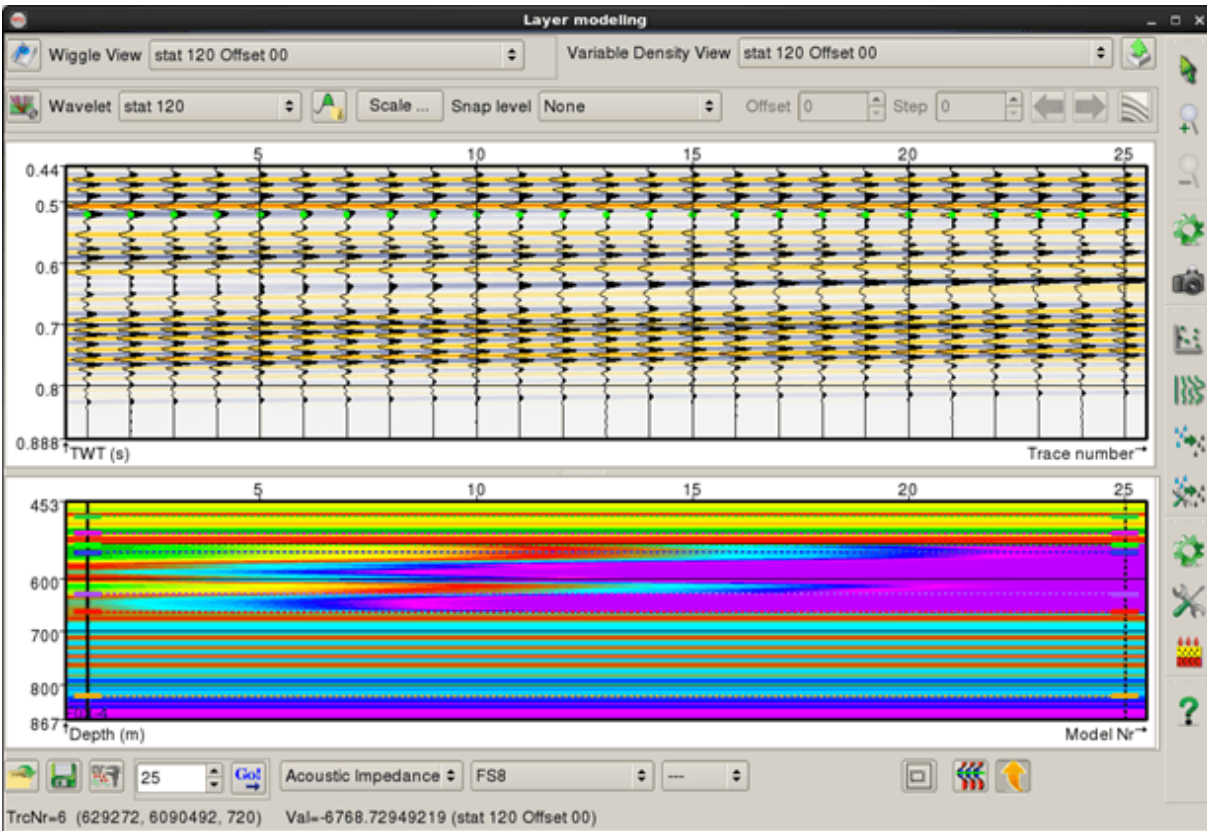


In *Define property changes* window, properties of a selected interval can be changed in such a manner that individual samples of a property are scaled to have the user-specified average in the given interval. The greyed-out left column contains initial average property values, and new average values must be specified in the right column. In the example below, *Porosity* is decreased to 5 percent with corresponding increases in *Density* and *Pwave velocity*. Click OK to confirm the changes.



The *Create Initial Model* window is updated as shown below. Click *Apply* or OK to generate the model.



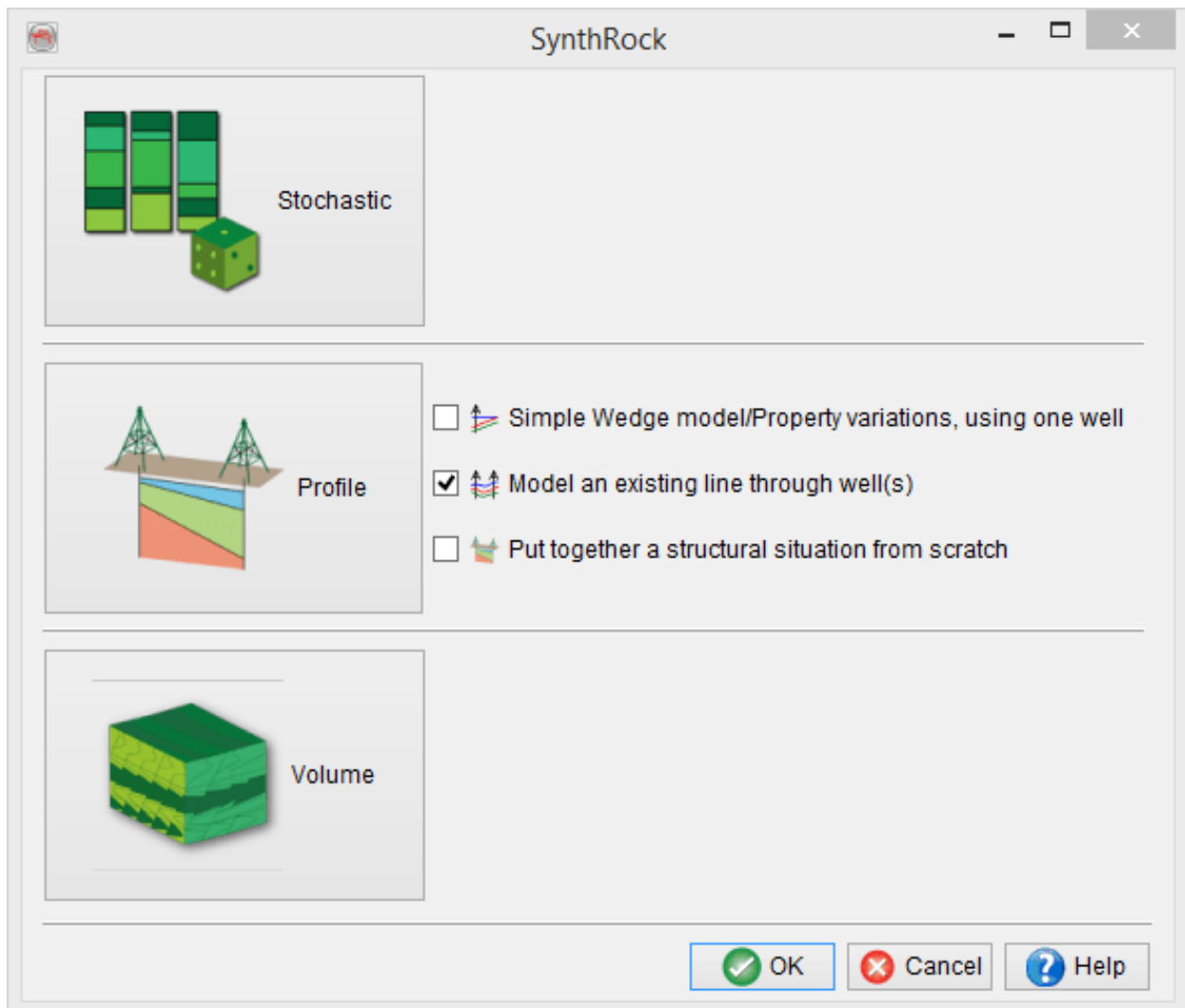




A wedge model with property variations

Finally, a wedge model can be combined with property variations by checking both options and defining the parameters as described above.

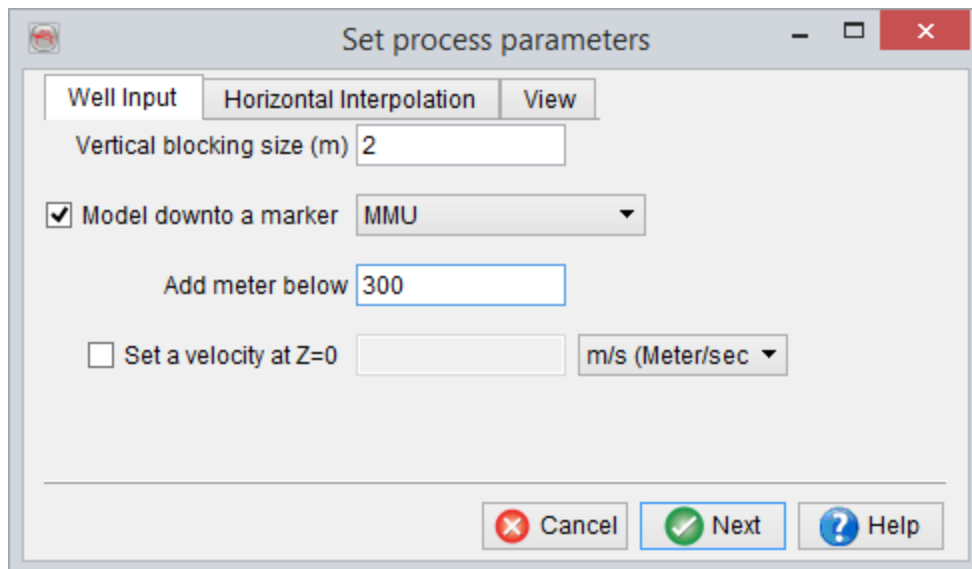
8.4.2 Model an existing line through well(s)

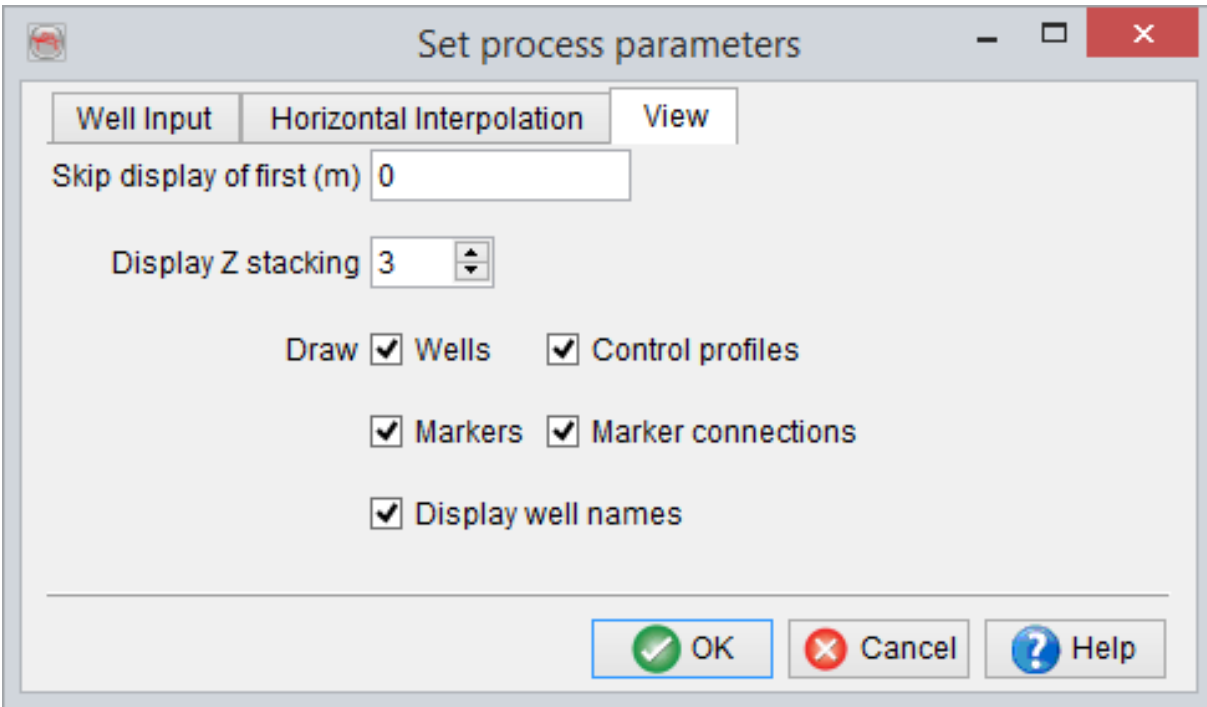
Model an existing line through well(s) is the second pre-defined workflow in SynthRock profile modeling which allows to quickly build a model through the existing wells in the project database. It is launched by checking the corresponding box in *SynthRock* start window and clicking OK.



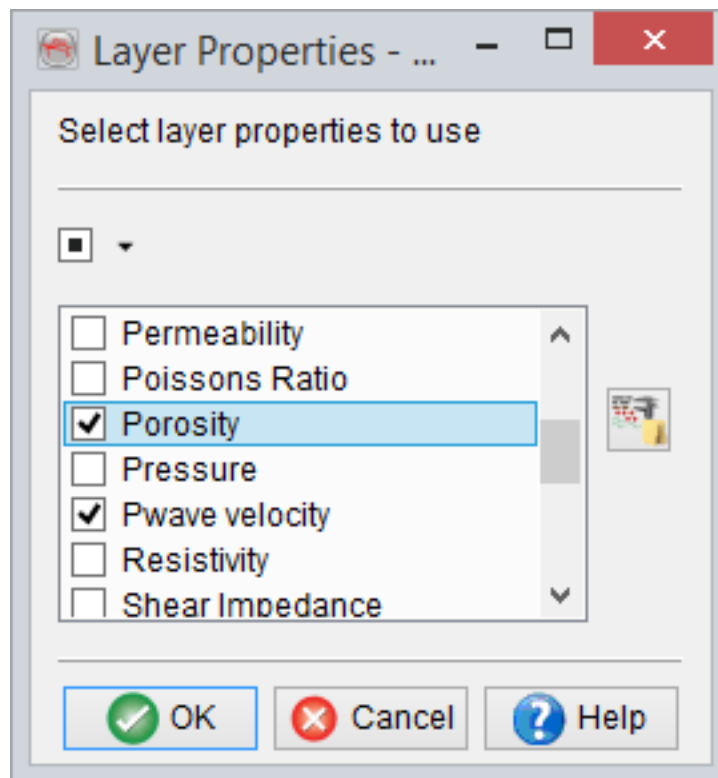
A user is expected to go through the pre-defined sequence of windows. The workflow starts, as any profile modeling in SynthRock, with setting modeling parameters in Set process parameters window and selecting properties to be modeled in Layer Properties - Selection window. Once the workflow is completed, both sets of parameters can be modified by clicking on the  and  icons of *Layer modeling* window respectively.

In the example below, few changes of the default parameters are made in the first window in order to limit the modeled interval to [300 m; MMU].

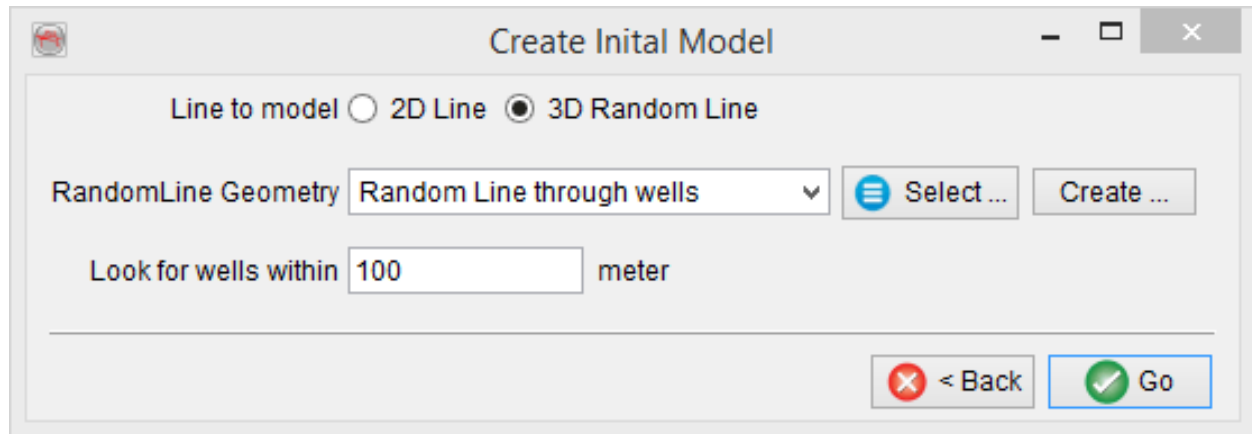




Porosity is added in addition to the default properties in the second window.

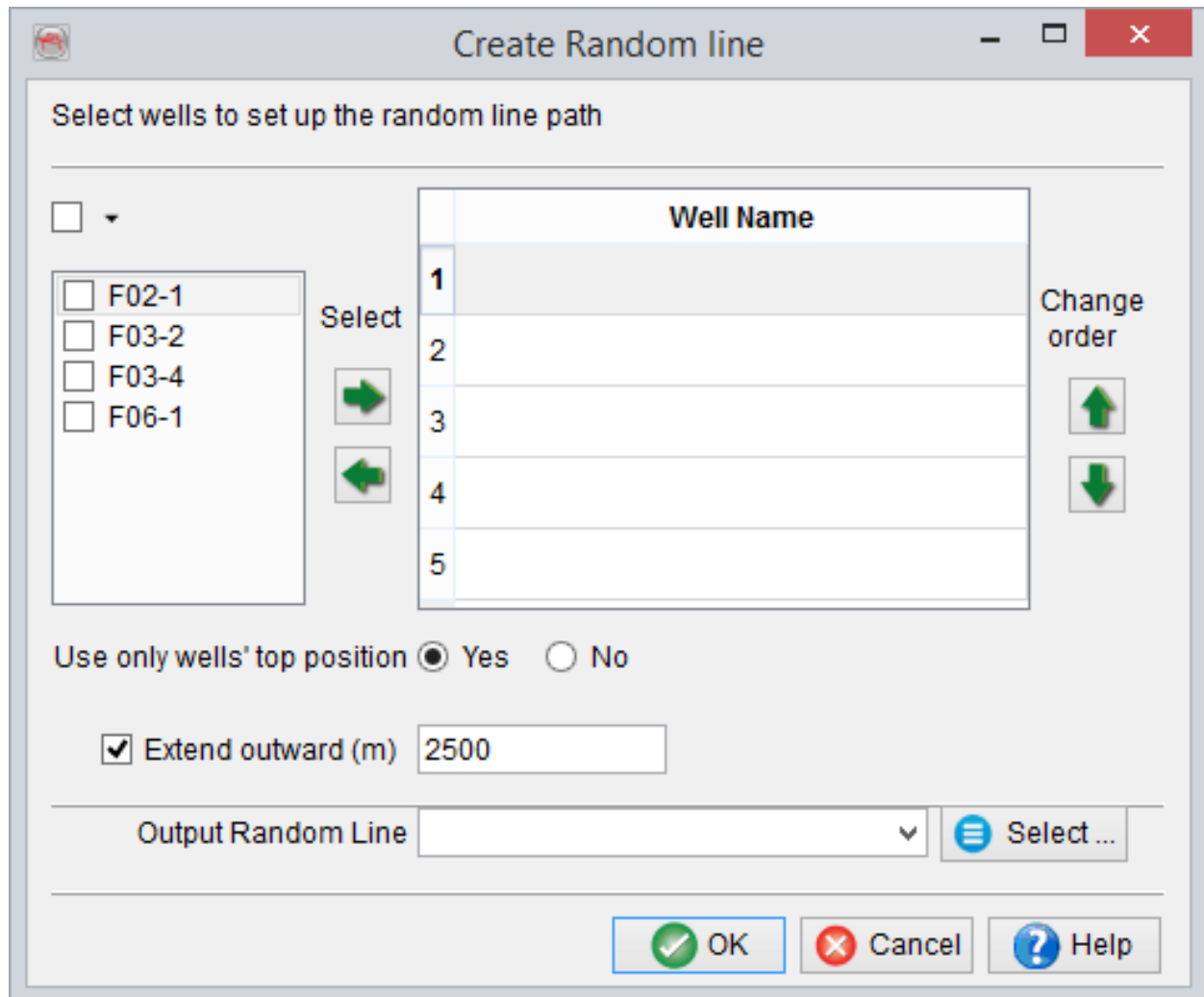


Once the modeling parameters and the property list are confirmed, *Create Initial Model* window appears. First, the *Line to model* option must be selected as either *2D line* or *3D Random line*.



Create 3D Random Line

Geometry of a 3D random line through the existing wells can be created by clicking on *Create* button.



Wells to be used must be first selected and then added/removed using the 'Select' arrows. Order of the wells can be defined using the 'Change order' arrows and this order defines the shape of a random line. If *Use only well's top position* is set to *Yes*, the random line is drawn through the well tops. Alternatively, *No* gives an option to choose *Top* or *Bottom* of each well. A random line can be optionally extended beyond the locations of the first and the last wells by checking the box and specifying the distance. Finally, after giving a name click OK to create the random line and return to *Create Initial Model* window.

Create Random line

Select wells to set up the random line path

F02-1
 F03-4

Select

Well Name	
1	F06-1
2	F03-2
3	
4	
5	

Change order

Use only wells' top position Yes No

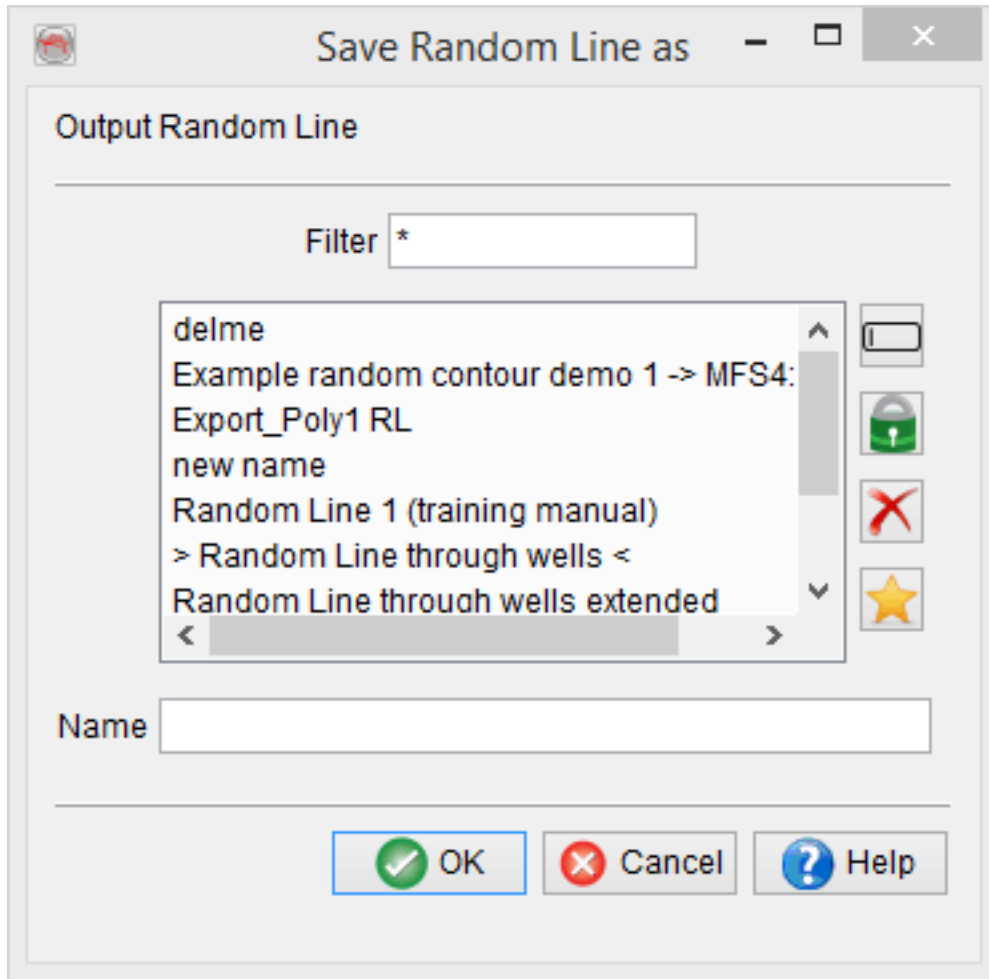
Extend outward (m)

Output Random Line

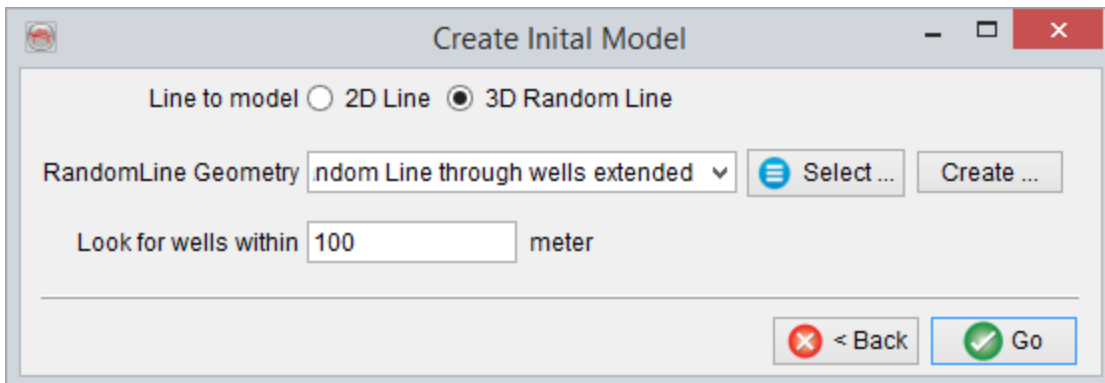
Existing 3D Random line

If a model is created along an existing 3D random line, click on *Select* button to choose from the list of existing random lines in the project database.

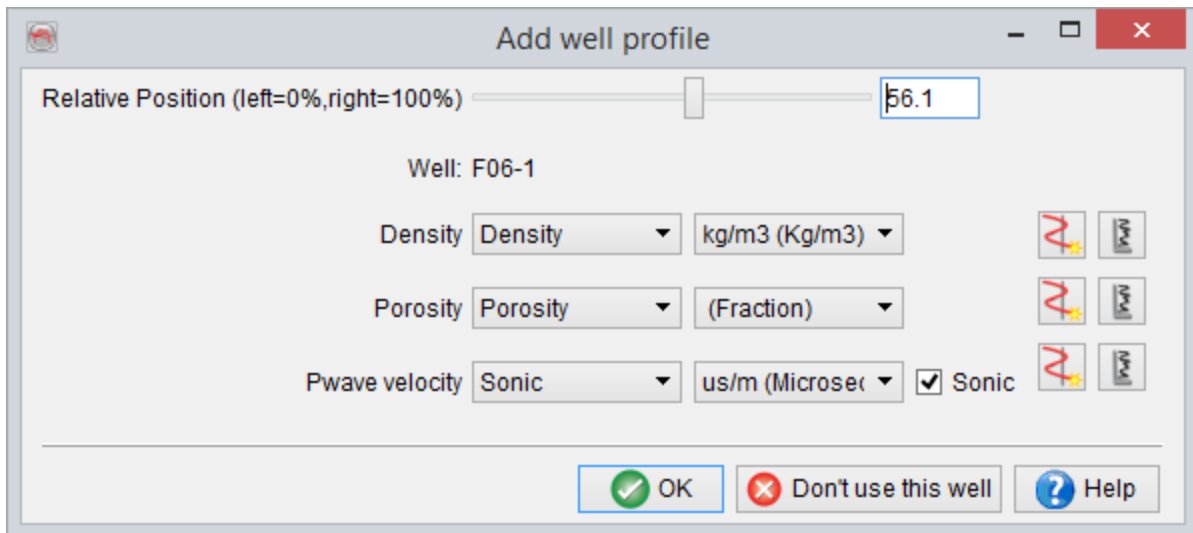
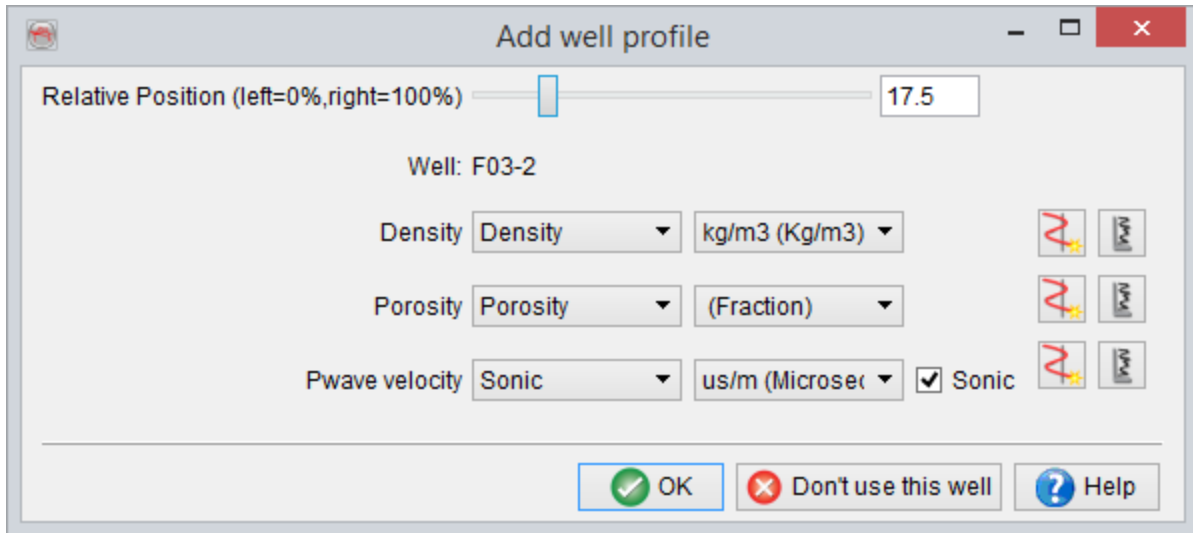
As an example, a model is created along a *Random Line through wells extended* which passes through all 4 wells, *F06-1*, *F02-1*, *F03-2* and *F03-4* and extends outwards on both sides.

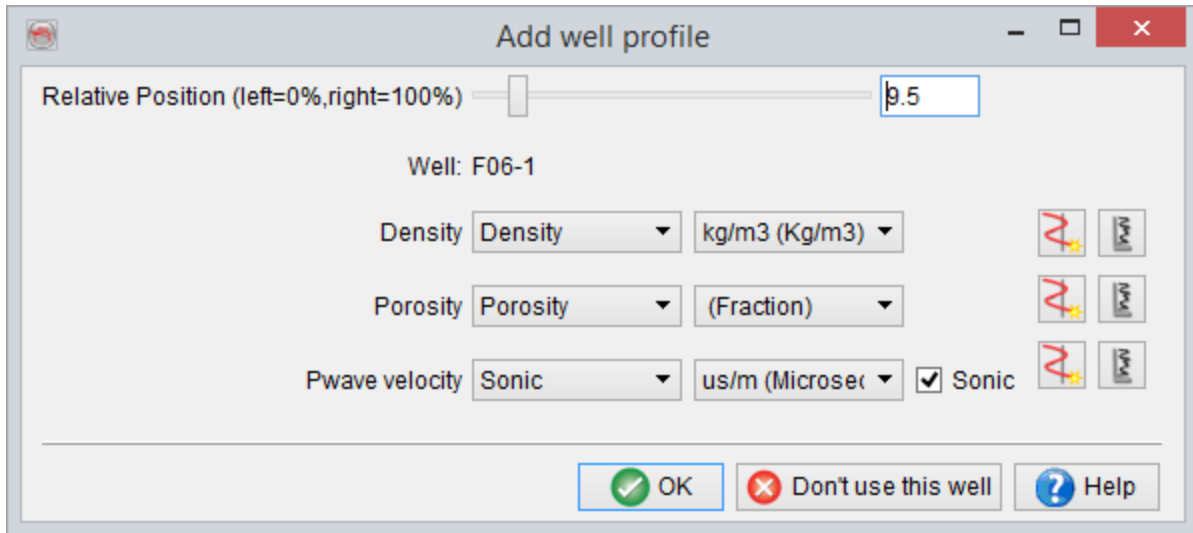
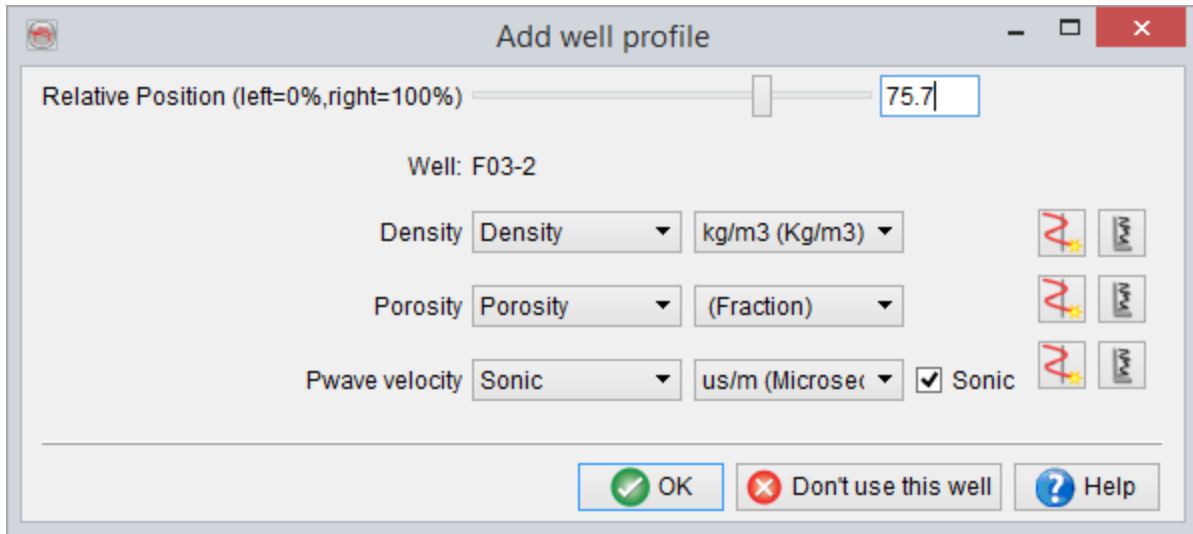


Once the line is selected, its name appears in the *Create Initial Model* window. *Look for wells within* allows to include wells that are not positioned exactly on the selected line.

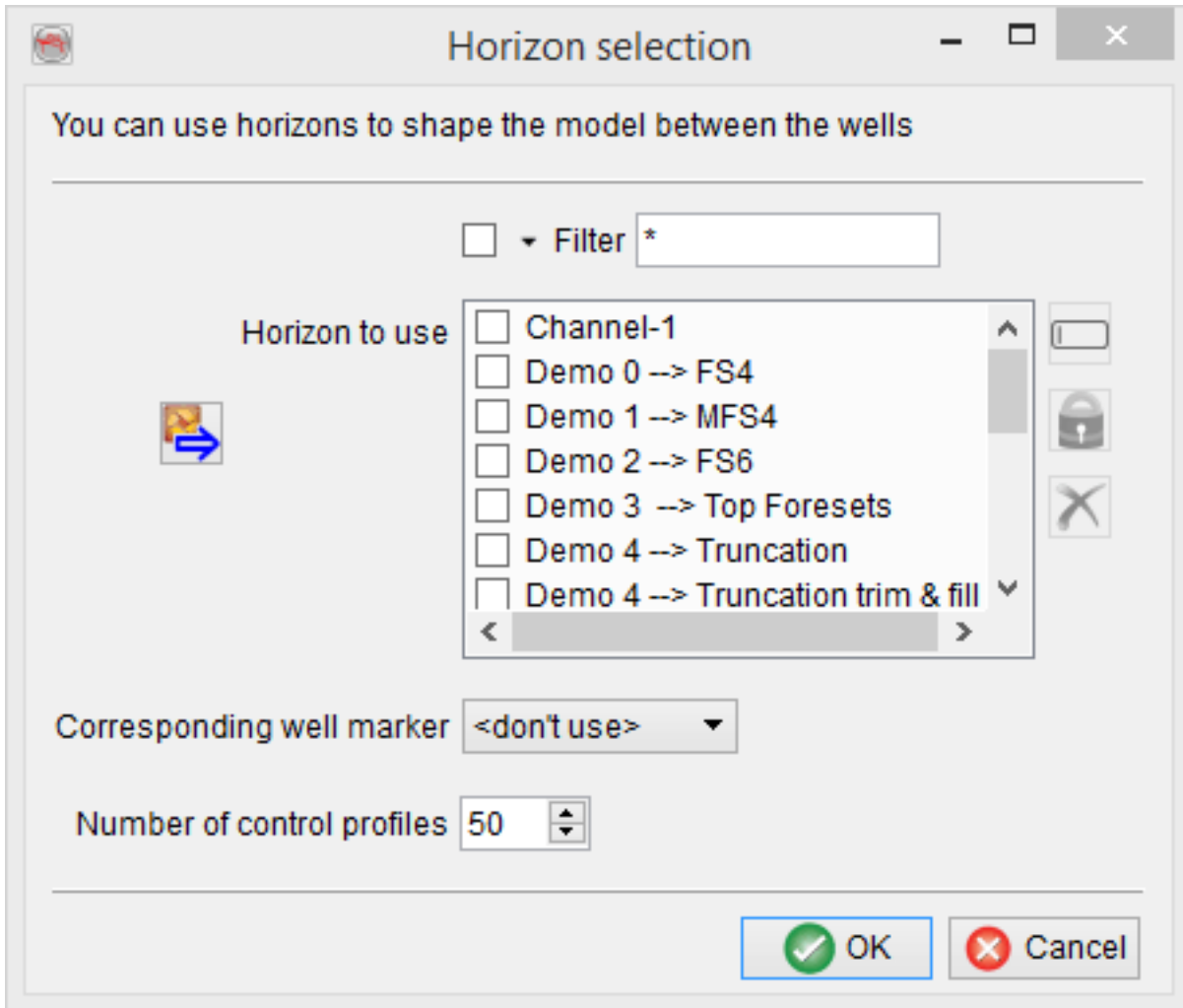


Click OK to proceed, and then go through a number of *Add well* windows. Such window is shown for every well located within the specified distance from the selected line. Automatically determined *Relative Position* and logs can be modified if needed. A user can also select not to use a particular well by clicking on *Don't use this well* button.

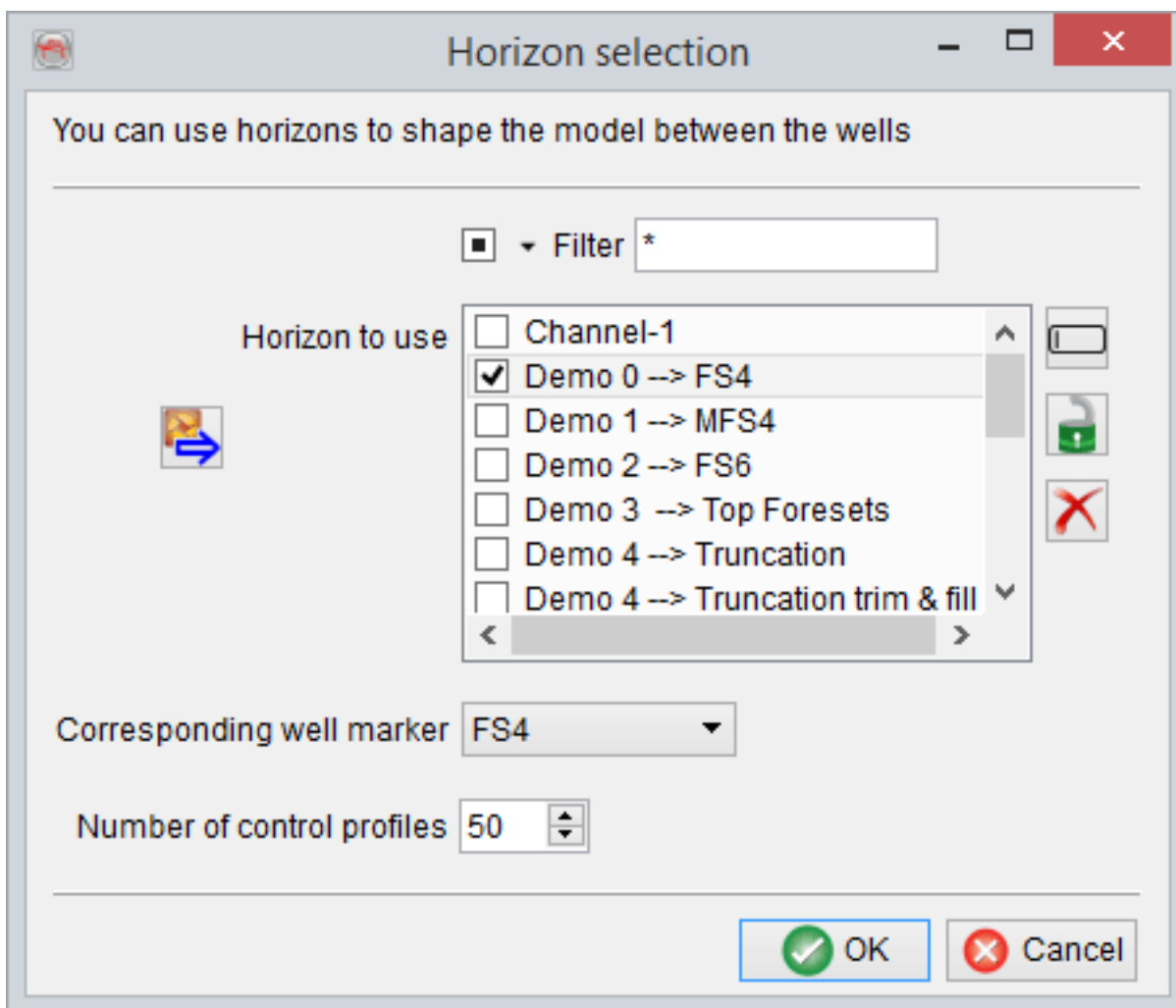




Next, the *Horizon selection* window allows to choose horizons and corresponding formation tops to steer the horizontal interpolation. Select any horizon in *Horizons* to use list and note that the *Corresponding well marker* is set to *<don't use>* by default. This means that the selected horizon is not used in modeling.

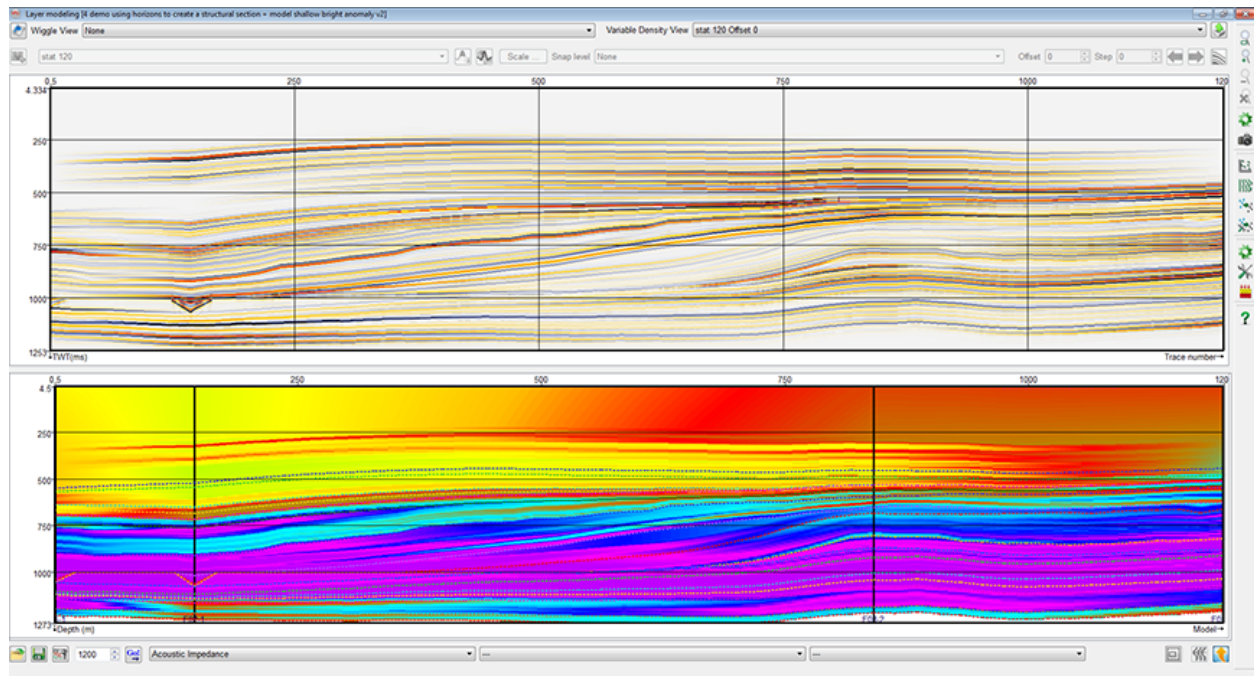


Select the first horizon to be used in the modeling and set the corresponding marker as shown below for '*Demo 0 --> FS4*' horizon with a corresponding *FS4* well top. Similarly, well tops have to be selected for all the horizons to be used in the modeling. Finally, *Number of control profiles* can be specified. Clicking OK generates a model with default number of pseudowells.

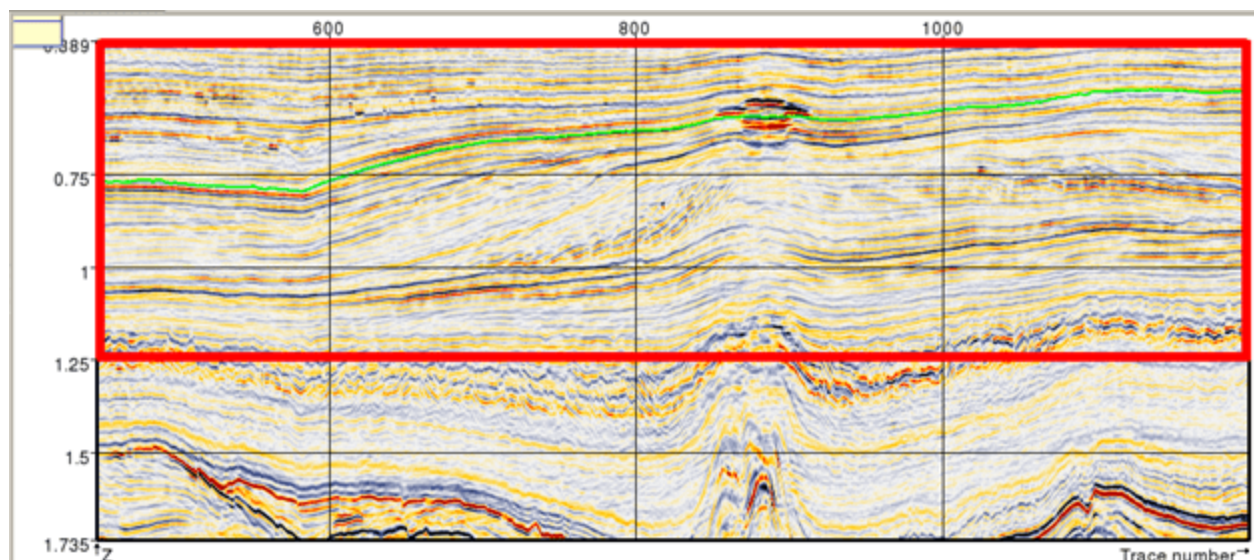


At this point the pre-defined workflow is finished, and further modeling can be continued using the classic Profile mode functionality in *Layer modeling* window.

An example below shows the final model along the *Random Line through wells extended*. Five horizons and corresponding well tops are set as following: 'Demo 0 --> FS4' - FS4; 'Demo 1 --> MFS4' - MFS4; 'Demo 4 --> Truncation' - Truncation; 'Demo 5 --> FS7' - FS7; and 'Demo 6 --> FS8' - FS8. Note, that default number of pseudowells is changed to 1200, and some display parameters are modified to create this image. The green line overlying synthetic seismic corresponds to FS8 well top.

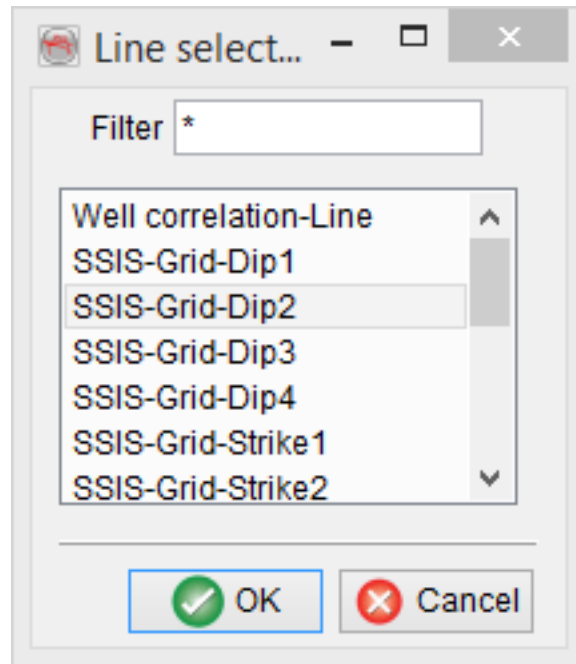


Below is the corresponding seismic section along the selected random line. The red box outlines the modeled interval, and the light green line corresponds to 'Demo 6 --> FS8' horizon. The constructed model is a good start for further detailed modeling. The detailed modeling can be done to support existing interwell interpretation. For example, by modifying properties of the control profiles in the vicinity of the high-amplitude anomaly seen in the middle of the seismic section along the green horizon.

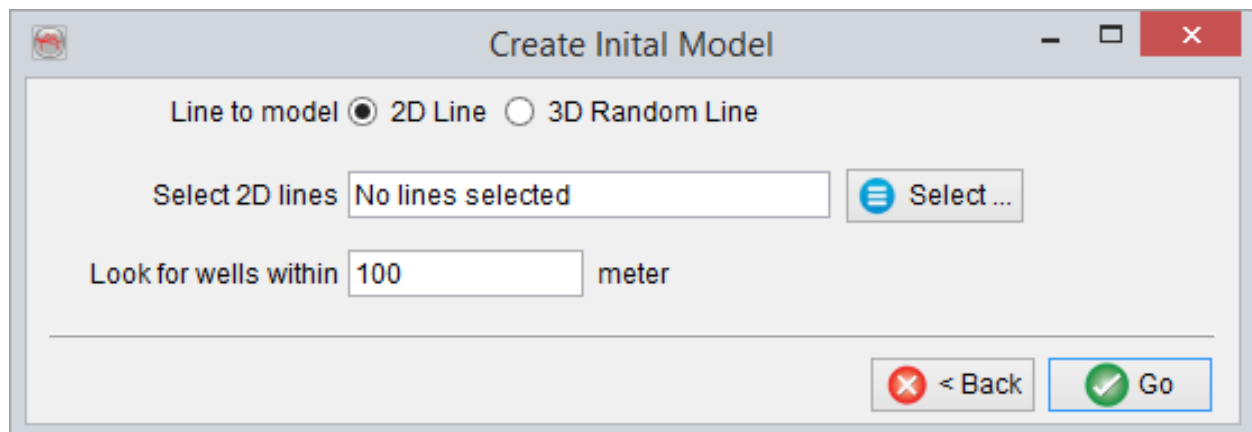


Existing 2D line

Switch Line to model to 2D line option and click *Select*. In the *Line selection* window choose the existing 2D line in the project database along which modeling is to be done.



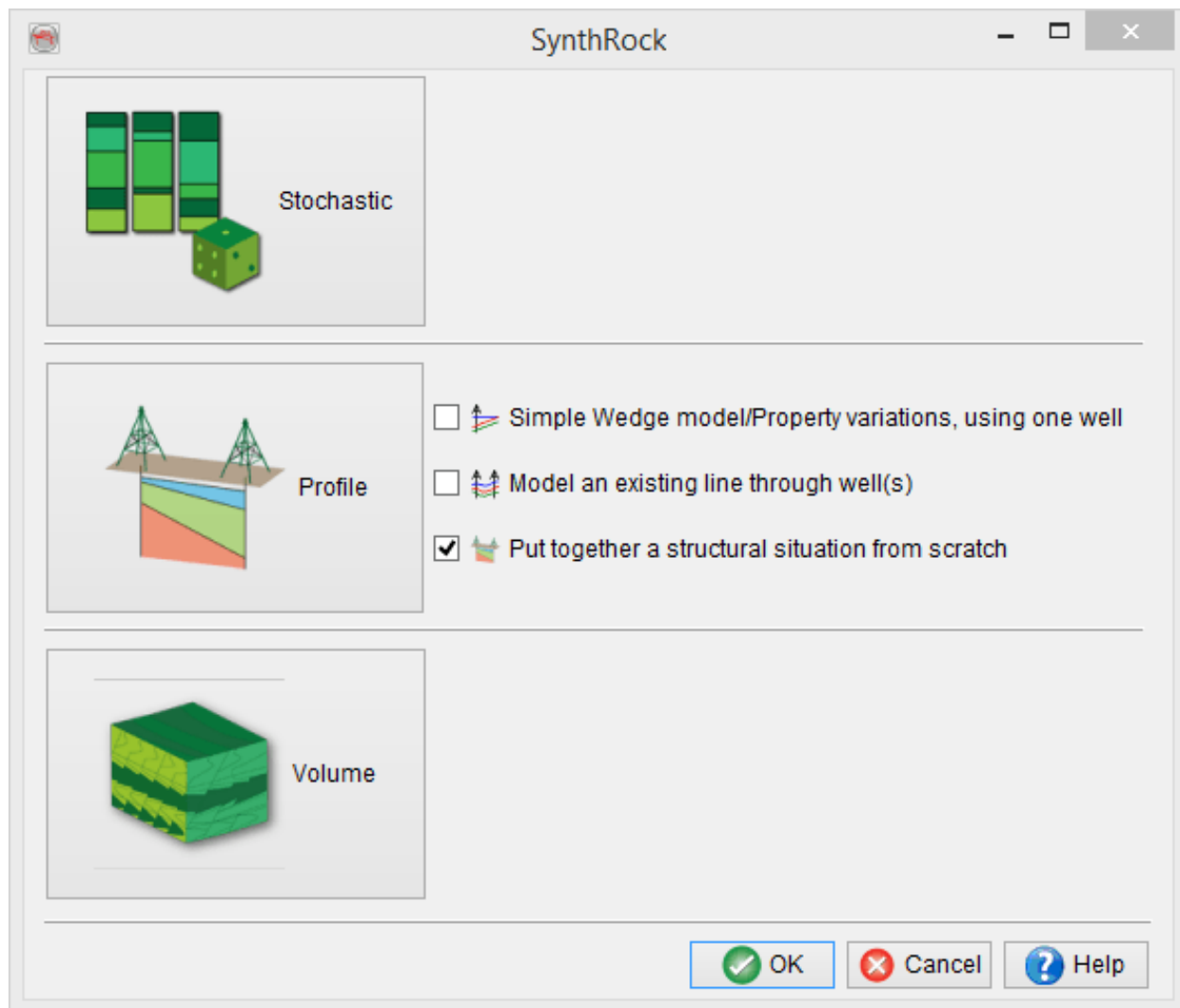
Once the 2D line is selected, specify the appropriate distance from the line to include all required wells for the modeling and click OK. Afterwards, follow the instructions described for Existing 3D Random line.





8.4.3 Put Together a Structural Situation from Scratch

Put together a structural situation from scratch is a powerful modeling tool with all classic Profile mode functionality. In this mode a user starts building a model from scratch. It gives all the flexibility available, but also requires more time than the pre-defined workflows. Therefore, it is recommended to start with pre-defined workflows if possible and, once a simple model is built, continue with a classic Profile mode functionality as described below.

Put together a structural situation from scratch can be launched from *SynthRock* start window either by checking the corresponding box and clicking OK or simply by clicking on the large Profile icon.



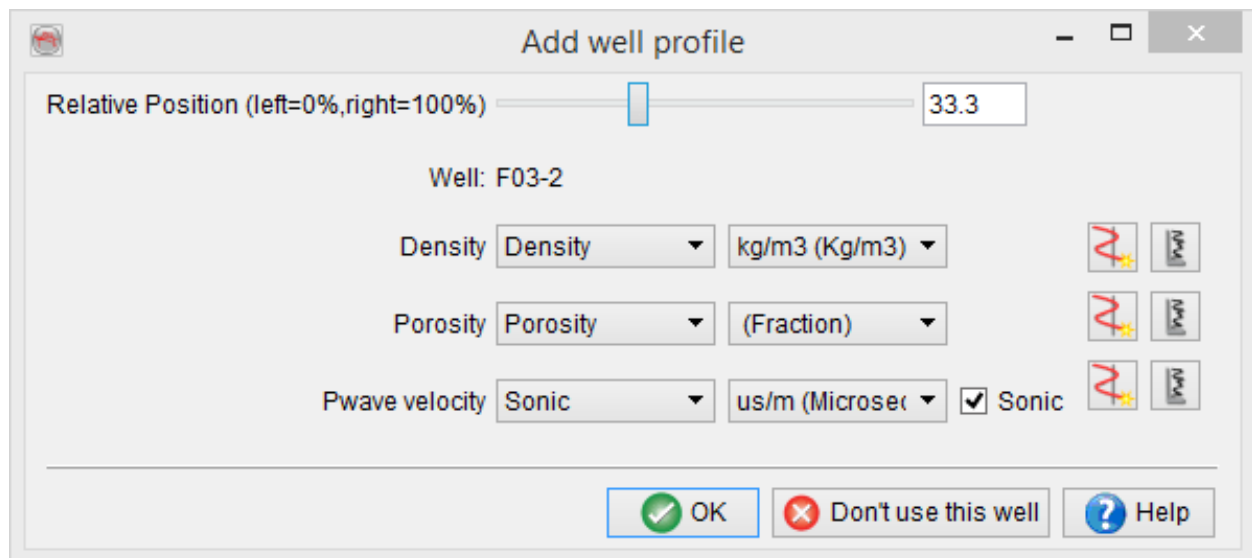
Add/edit well and control profile




Control points for the profile modeling are added with a right-click in the bottom part of Layer modeling window. Before the first control point is added, a pop-up window *Set process parameters* requires to confirm/specify modeling parameters and pop-up window *Layer Properties - Selection* allows to select properties to be used in modeling. Later during the modeling both sets of parameters can be modified by clicking on the  and  icons respectively. The first control point must be an existing well in the

project. Additional control points can be created by adding more wells (Add well) or by setting existing pseudowells as control profiles (*Add Control Profile*).

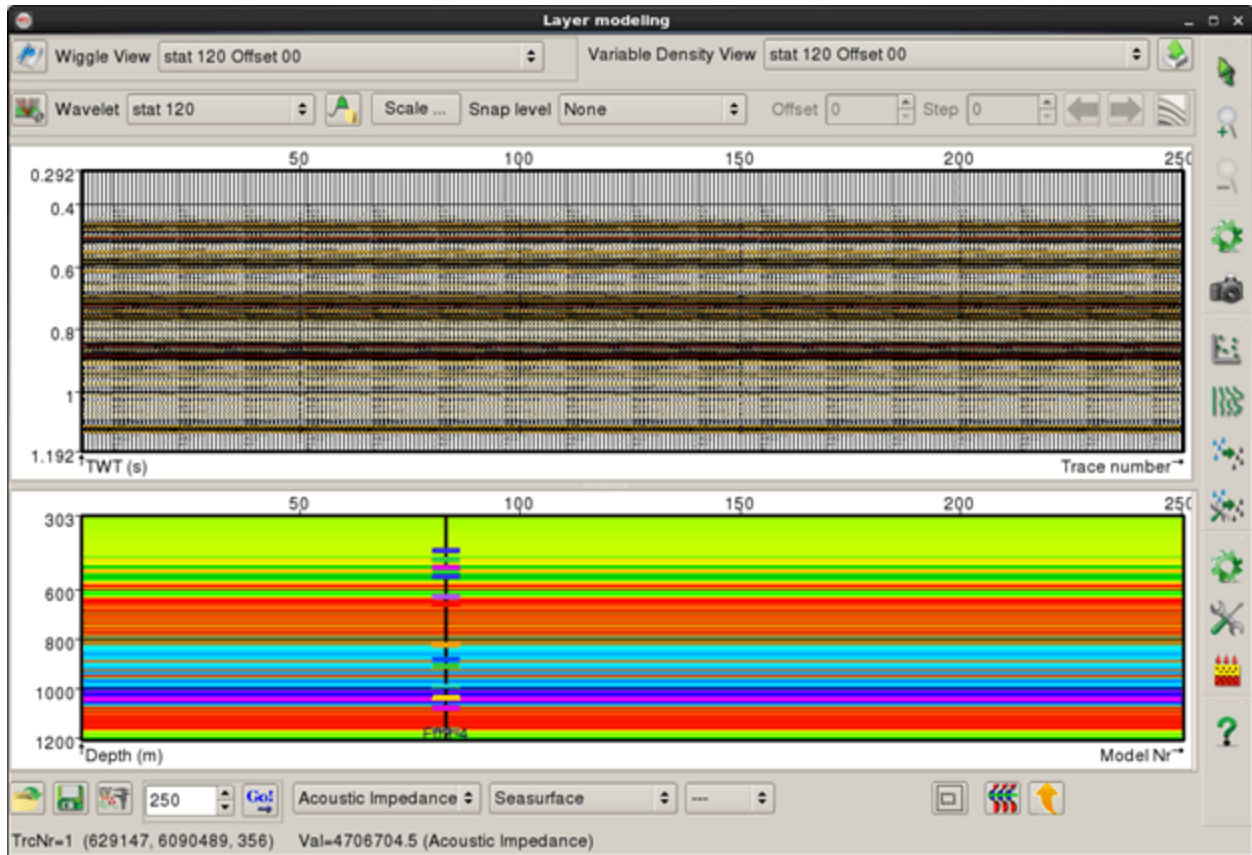
Add the first well

Click in the area where it says *<Click to add a well>*. Once the modeling parameters and the property list are confirmed, the first control point can be added. *Add well* window allows to choose the well to be added.



The *Relative Position* element specifies where the new well is added in the profile. Here, 0 and 100% correspond to the left and right ends of the profile respectively. Thus, the position of the added well is determined as $(N * Relative Position)/100$ where N is the number of pseudowells to be generated (specified left of the  button). Note, that all properties chosen in *Layer Properties - Selection* window are required to have a log selected. Missing logs can be either created by clicking on the  icon without having to exit SynthRock or imported using the *Well Manager*. The selected log can be displayed by clicking on the  icon. Pressing Ok or *Apply* creates a specified number of

pseudowells that have been interpolated/extrapolated from the selected well with logs upscaled according to *Set process parameters* window. The original well is displayed as a solid vertical line. When the first well is added, the generated pseudowells are simply duplicates of the original one. Corresponding synthetic seismic section is generated using the chosen wavelet and displayed in the top part of *Layer modeling* window.

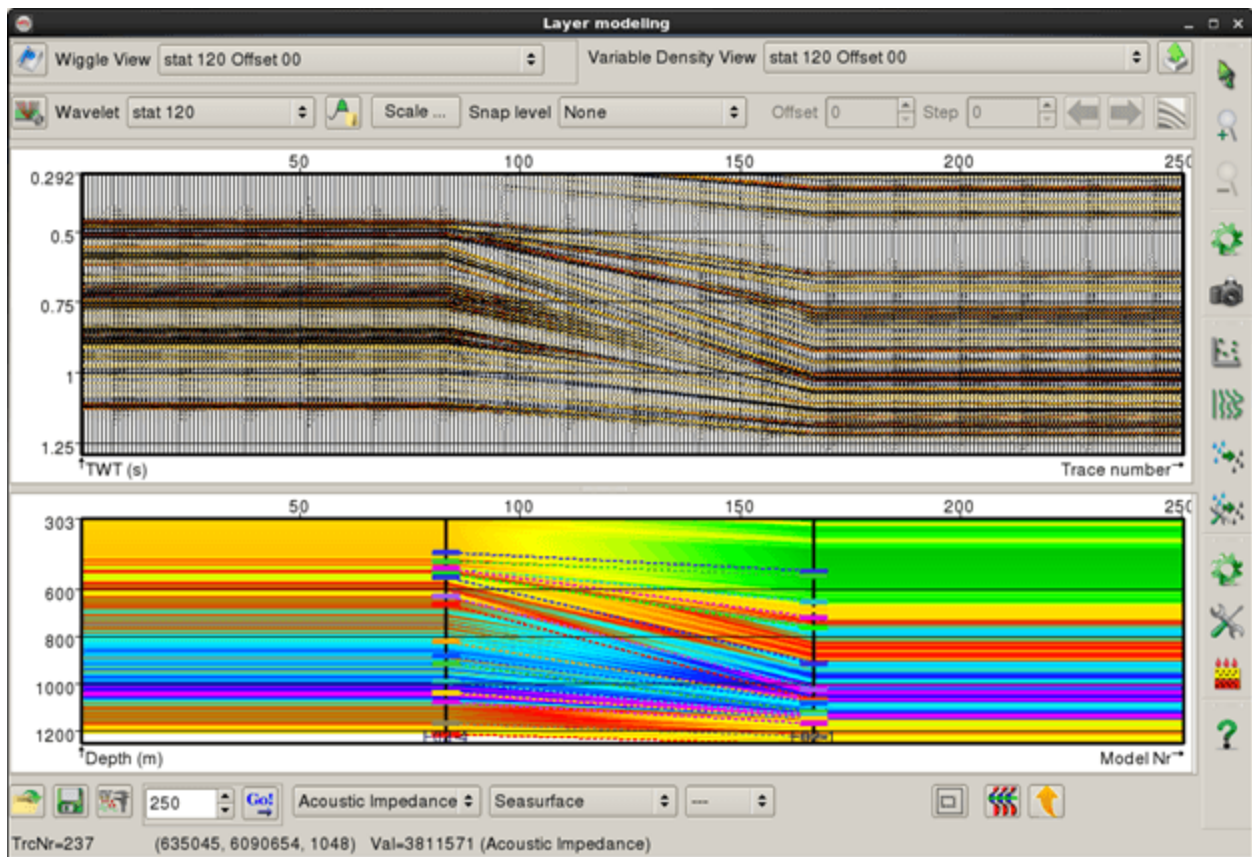


Add more wells

Once the first well is added, right click in the lower part of the *Layer modeling* window brings up several options as shown below.

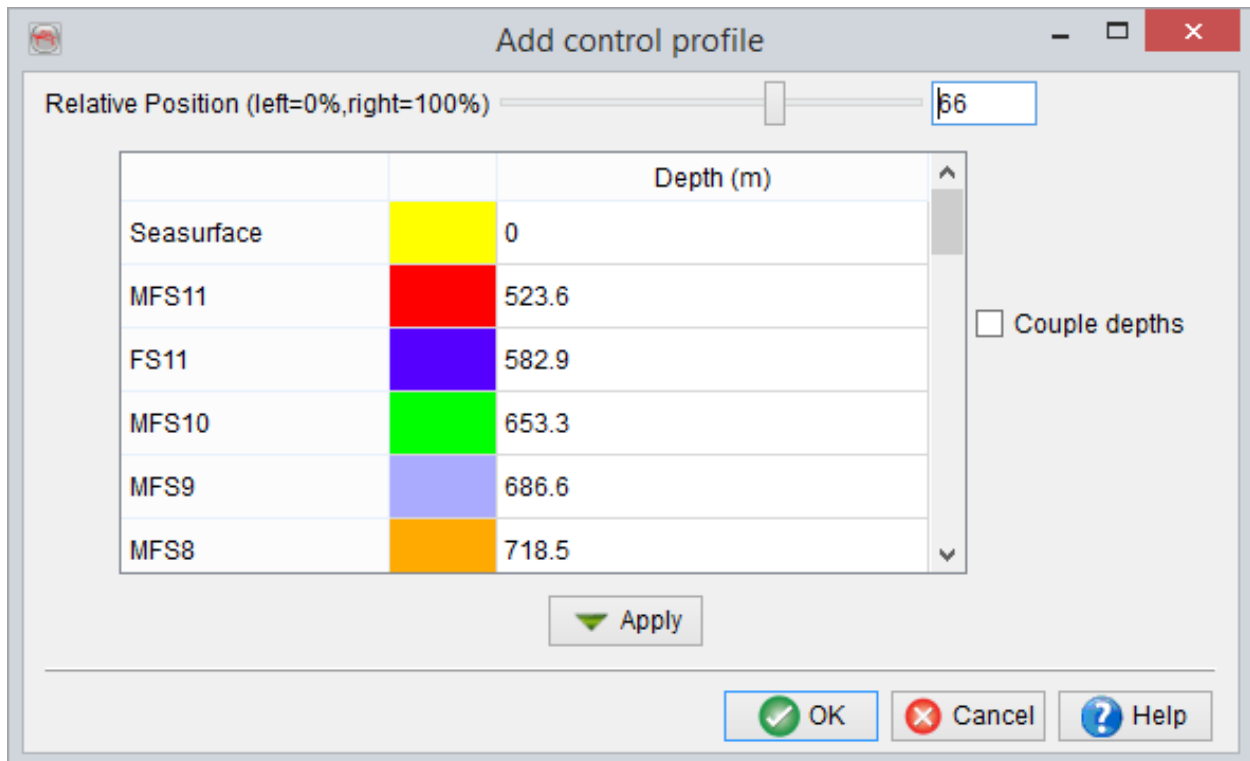
- Edit ...
- Remove
- Inspect values
- use Horizon
- change Properties
- Remove all Control Profiles
- Dump model to file

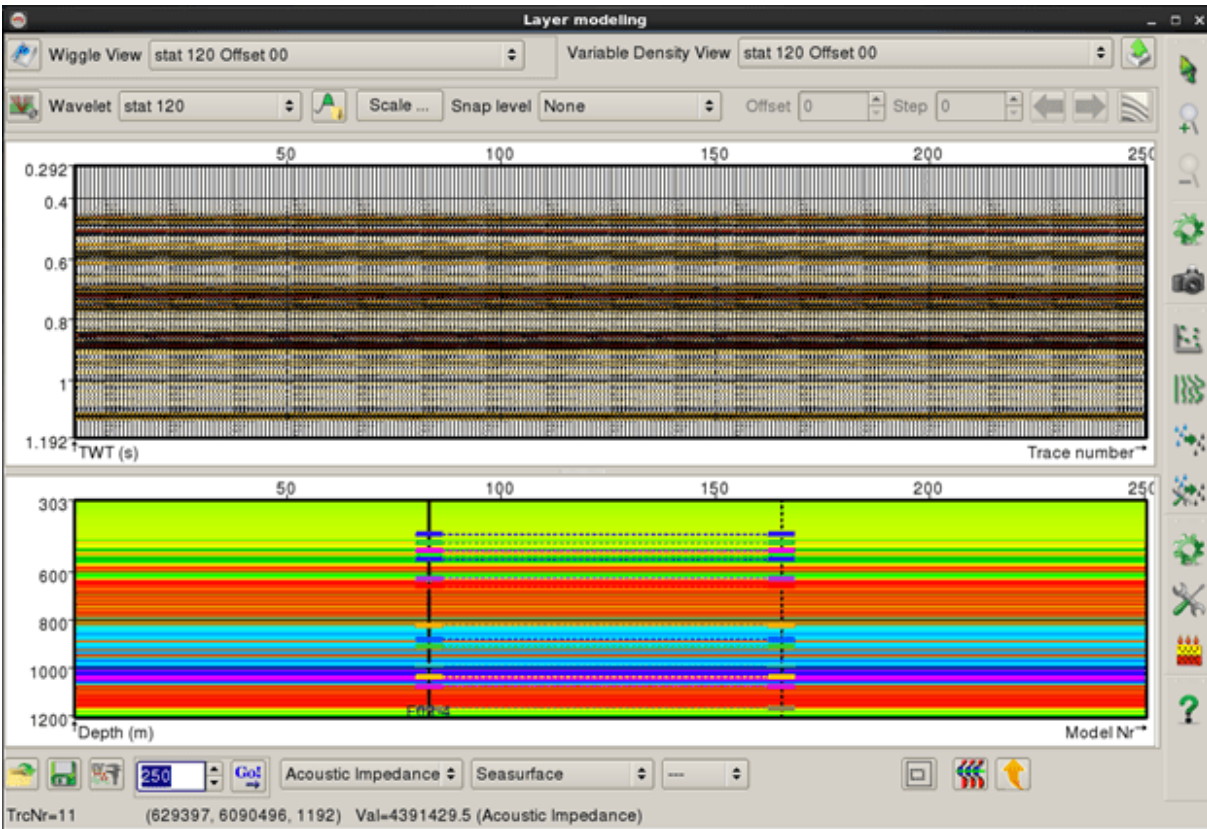
The *Add Well* option brings exactly the same window as the first time. After adding a new well, the pseudowells are re-generated by horizontal interpolation between the original wells in the profile.



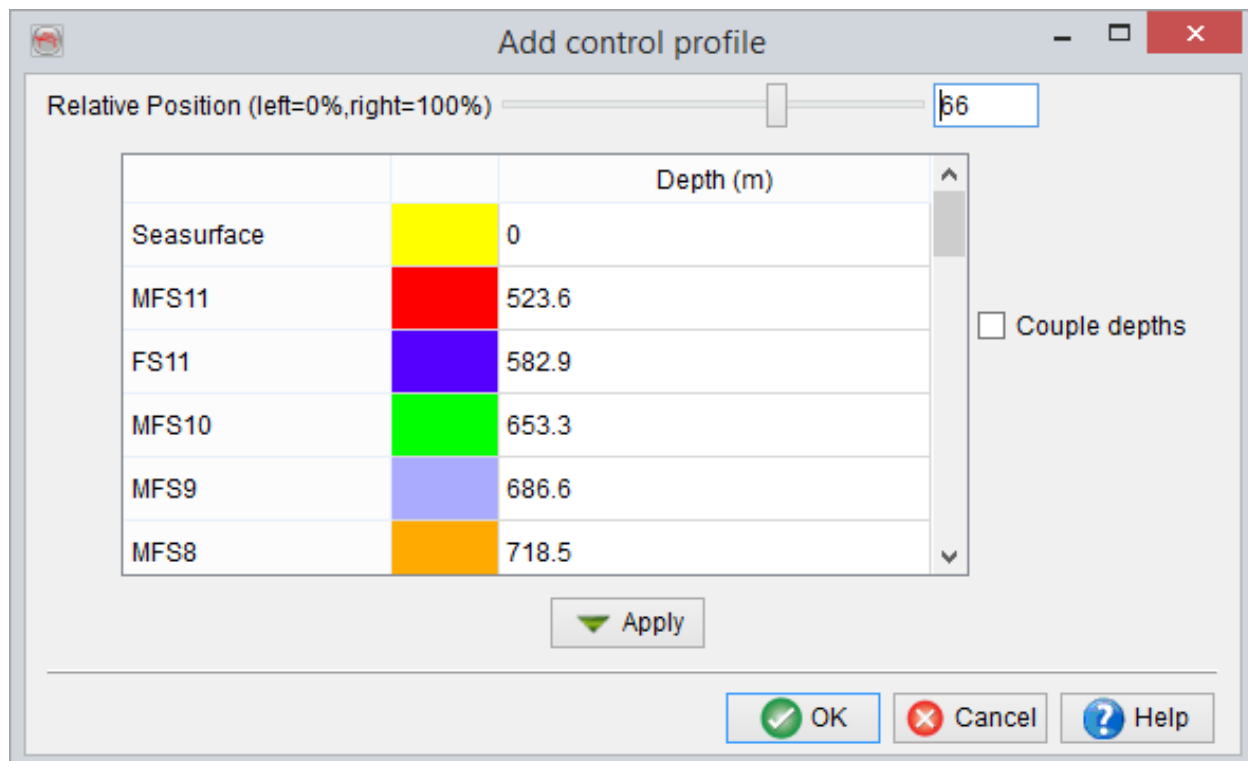
Add control profile

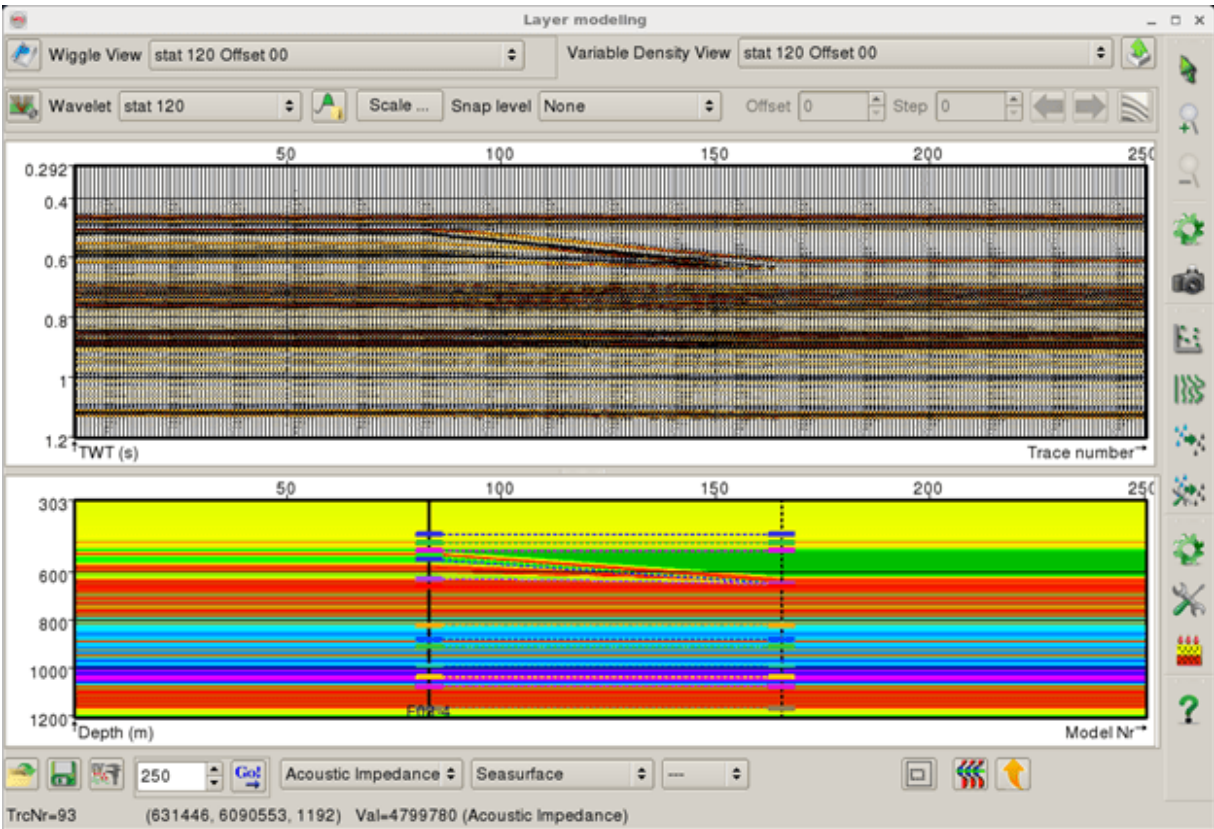
Right click on existing profile outside added wells and select to *Add Control Profile*. Specify *Relative Position* and click *Apply* to find out that a new control profile is added to the *Layer modeling* window without thickness or depth changes so far. In the example below, a control profile is added to the model containing only one original well *F03-4*. The control profile is displayed with a dashed line.



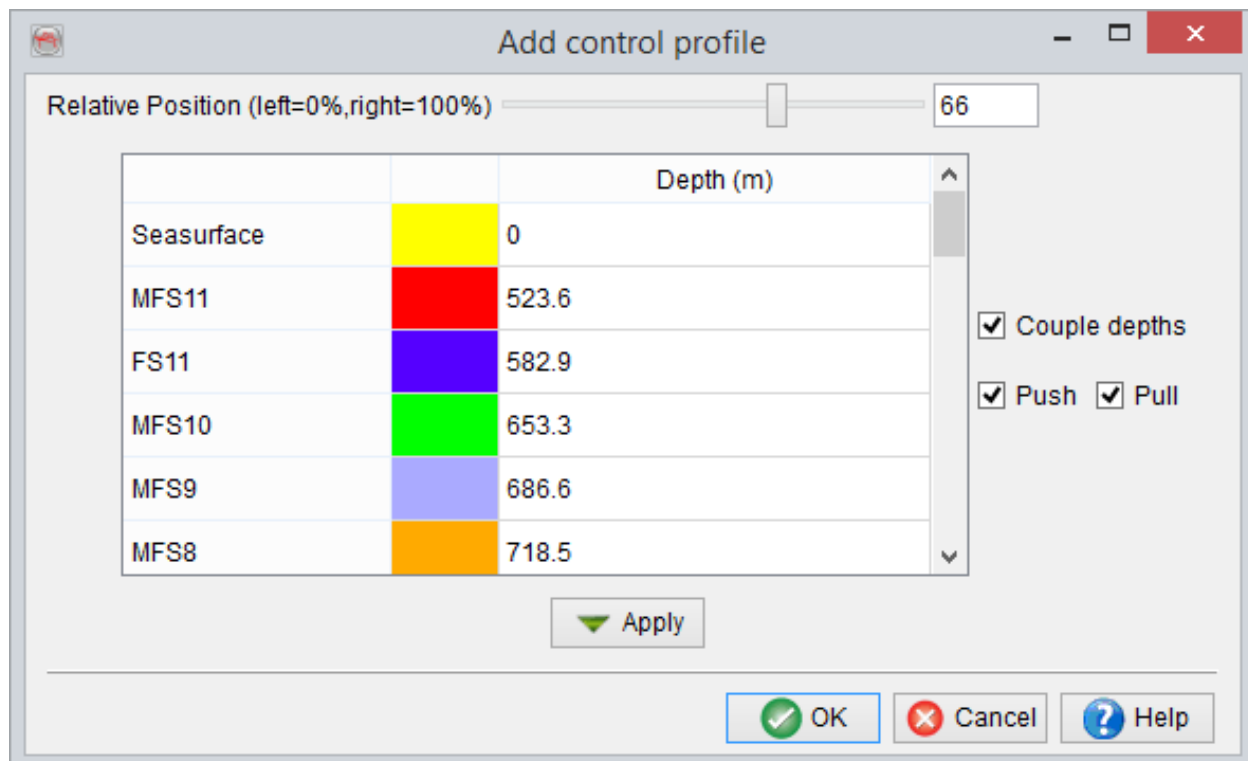


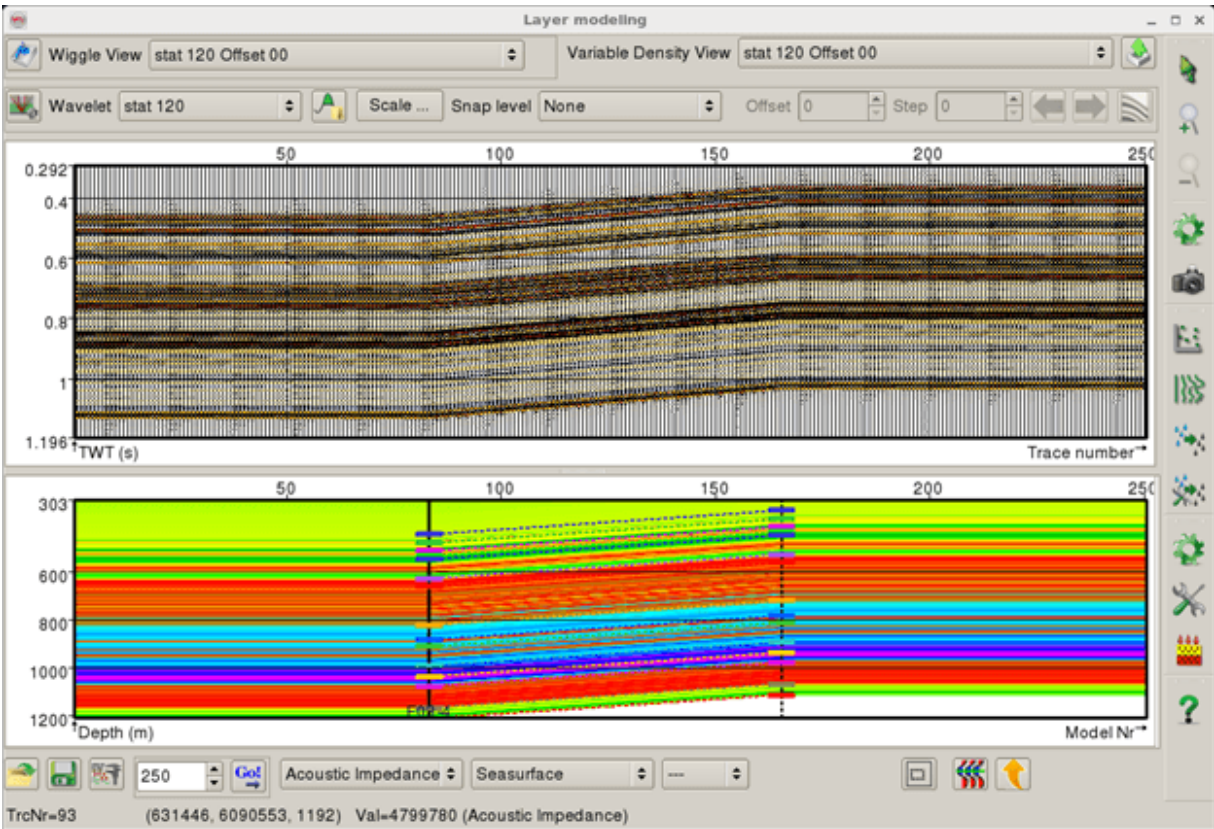
When adding a control profile, depths of a single top or of several tops simultaneously can be changed in *Add control profile* window. The depth of a single top can be modified when *Couple depths* option is unchecked. Note, that the same modified depth is assigned to those tops whose stratigraphic order is broken by the modification, while the depths of all other tops are kept unchanged. In the example below *FS8* depth is changed from 535 to 650 . Note, that *FS7* and *Truncation* tops are assigned the same depth in order to preserve the stratigraphic order of the tops. All the other depths are kept unchanged.



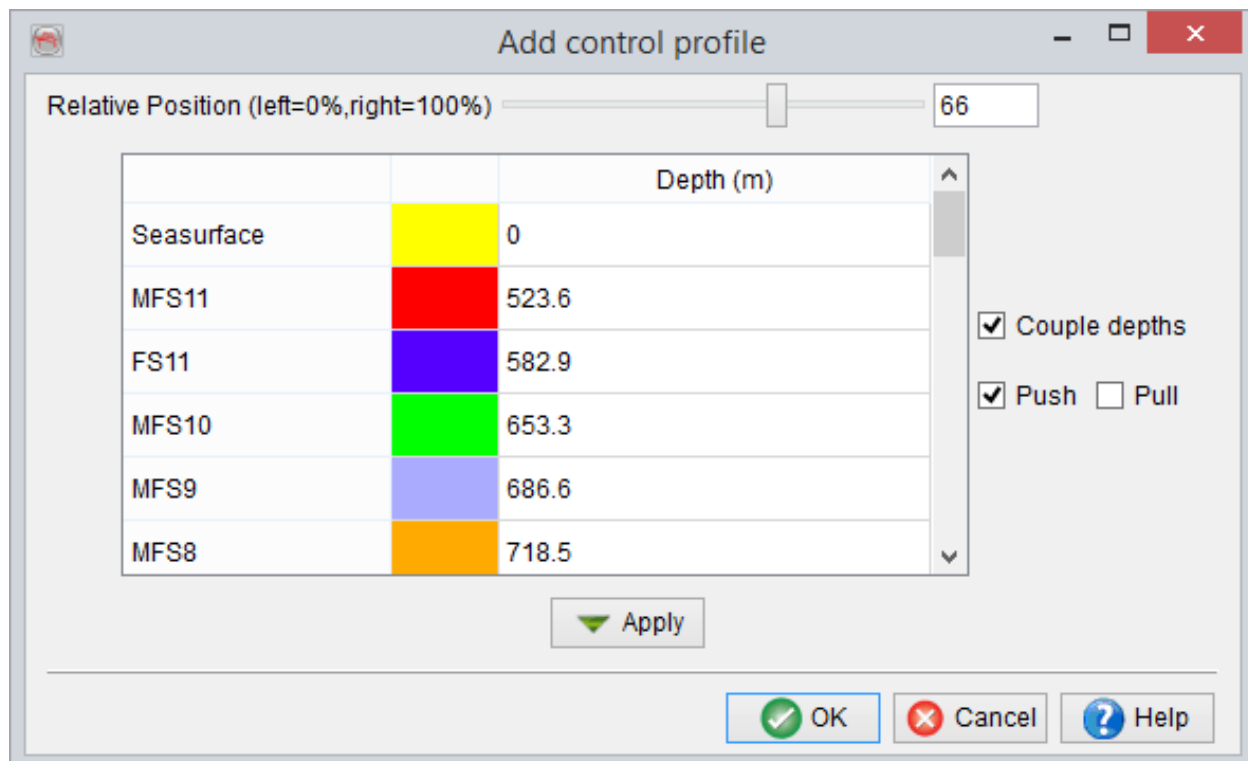


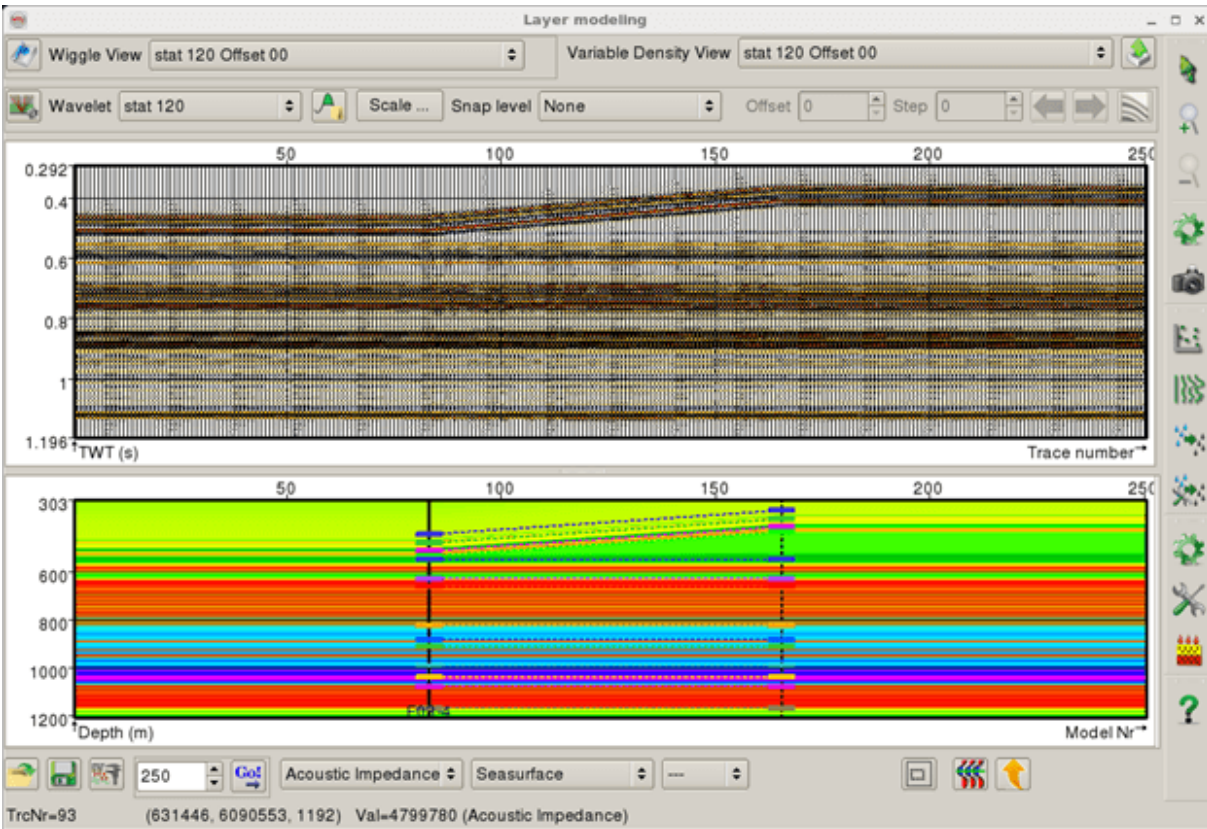
If the *Couple depths* option is checked, depths of several tops can be modified simultaneously. If both *Push* and *Pull* are checked, all tops can be shifted up or down by changing one depth by the desired shift. Note, that the stratigraphic order of tops is never broken in *Push* and *Pull* case. In the example below, the depth of *FS8* top is decreased by 100 m, thus shifting all formation tops 100 m up.



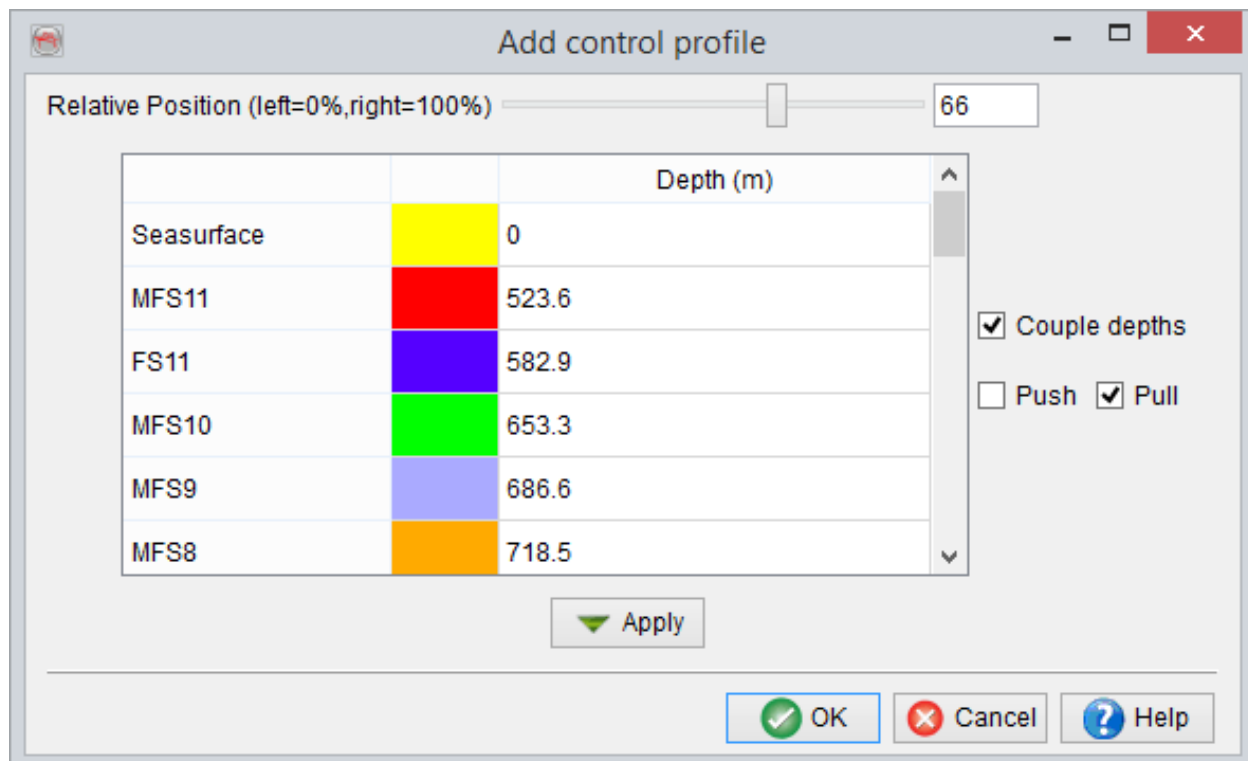


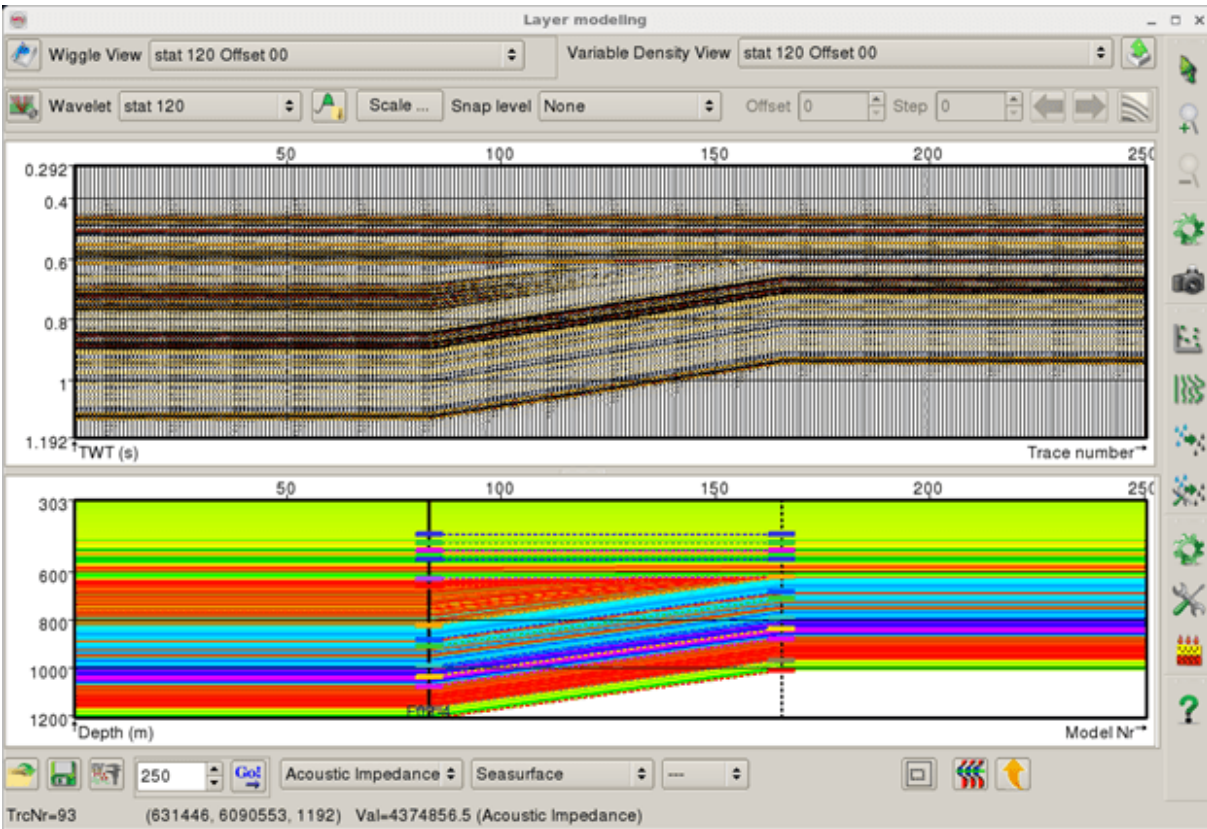
If only *Push* is checked and the modified top is shifted up, then the tops above it are pushed up by the same amount. Similarly, if only *Push* is checked and the modified top is shifted down, then the tops below it are pushed down by the same amount. Note, that the stratigraphic order of tops is never broken in the *Push* only case. In the example below, the depth of *FS8* top was decreased by 100 m, thus shifting it and all the tops above it 100 m up, while the tops below it are kept unchanged.





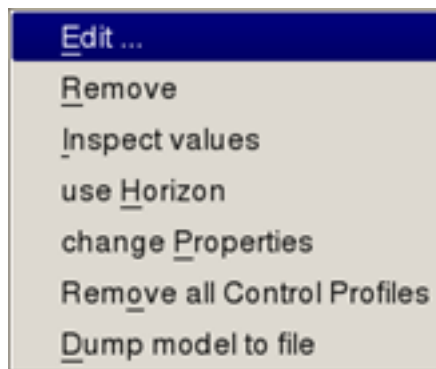
If only *Pull* is checked and the modified top is shifted up, then the tops below it are pulled up by the same amount. Similarly, if only *Pull* is checked and the modified top is shifted down, then the tops above it are pulled down by the same amount. Note, that the stratigraphic order of tops can be broken in the *Pull* only case. In the example below, the depth of *FS6* top was decreased by 200 m, thus shifting it and all the tops below it 200 m up. Note, that two tops above *FS6*, *Truncation* and *Top Foresets*, are assigned the same depth as *FS6* in order to preserve the stratigraphic order of the top. All the other tops above it are kept unchanged.





Edit well or control profile

Right click on added wells or control profiles allows to choose the *Edit* option.



Edit well parameters and *Edit control profile* windows are almost identical to the *Add well* and *Add control profile* windows respectively and allow changes of already added objects. Optionally, a single well or control profile can be removed by choosing *Remove* option.

Define Property Changes

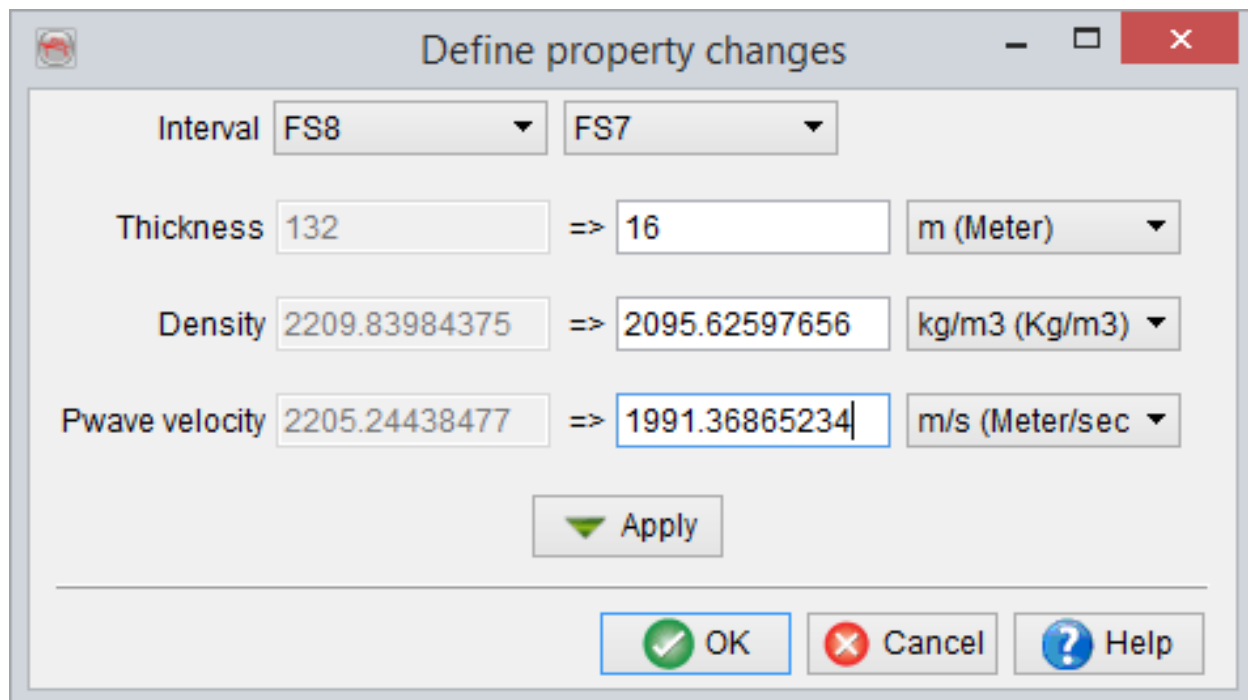
Right click change Properties option brings up a Define property changes window. Properties of a selected interval can be changed in such a manner that individual samples of a property are scaled to have the user-specified average in the given interval. Note, that this window for added wells does not allow thickness to be changed, whereas the same window for a control profile does allow.

Property	Current Value	Target Value	Unit
Interval	Lower Low Sor	FS6	
Thickness	98	160	m (Meter)
Density	2233.19238281	2171.50634766	kg/m3 (Kg/m3)
Pwave velocity	2319.39013672	2197.44921875	m/s (Meter/sec)

Apply

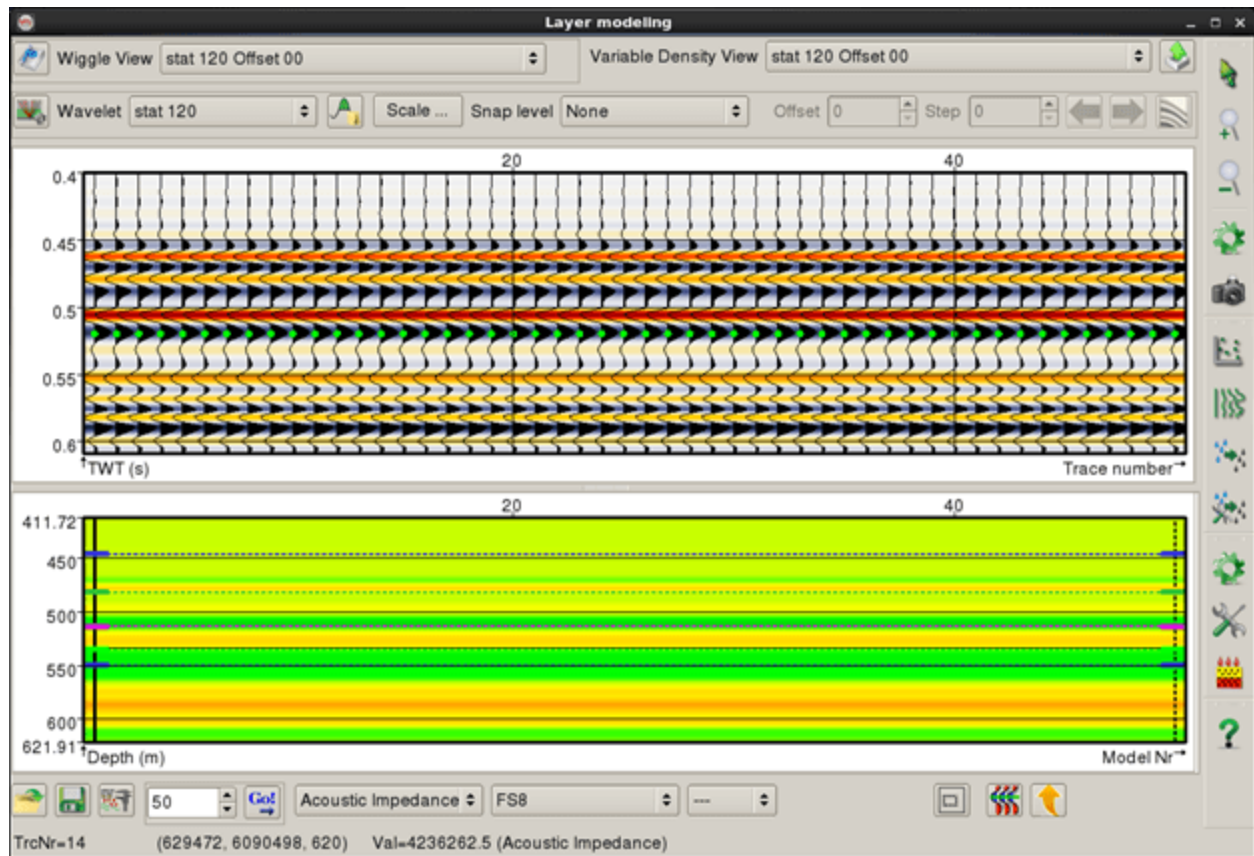
OK Cancel Help

Define property change for a well



Define property change for a control profile

The first example shows construction of a simple model of a lateral change in porosity and acoustic impedance. The model consists of one well at the far-left side and one control profile at the far-right side of the profile. The two lower markers, light green and blue, correspond to *FS8* and *FS7* tops respectively. Green dots found on synthetic seismic panel correspond to *FS8* top. So far, the properties are not varied along the profile.



Right click on the control profile and choose the *change Properties* option. Once the interval *FS8-FS7* is selected in the *Define property change* window, its greyed-out current properties are shown in the left column. User-modified properties are typed in the right column. Clicking *Apply* or *OK* re-generates pseudowells and synthetic seismic to reflect the changes.

Define property changes

Interval FS8 FS7

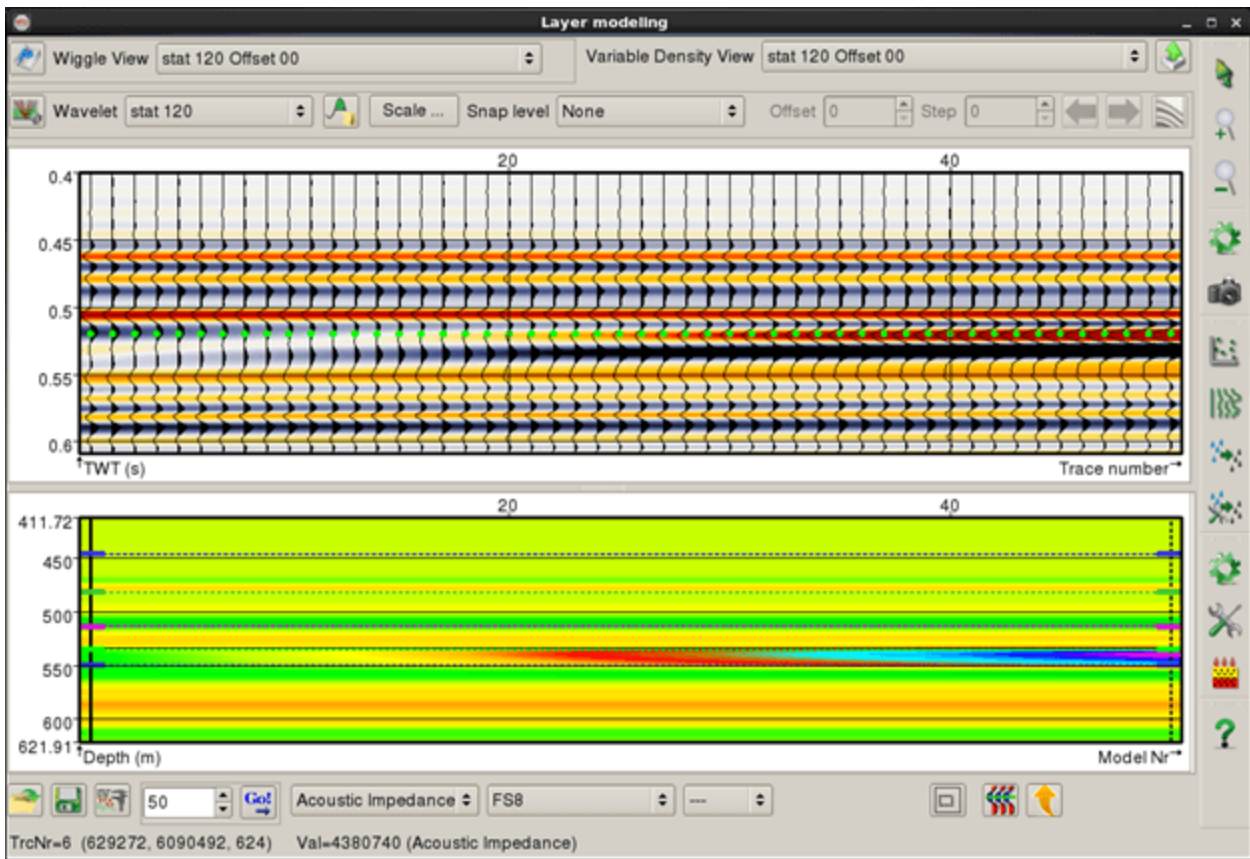
Thickness 132 => 16 m (Meter)

Density 2209.83984375 => 2300 kg/m3 (Kg/m3)

Pwave velocity 2205.24438477 => 2500 m/s (Meter/sec)

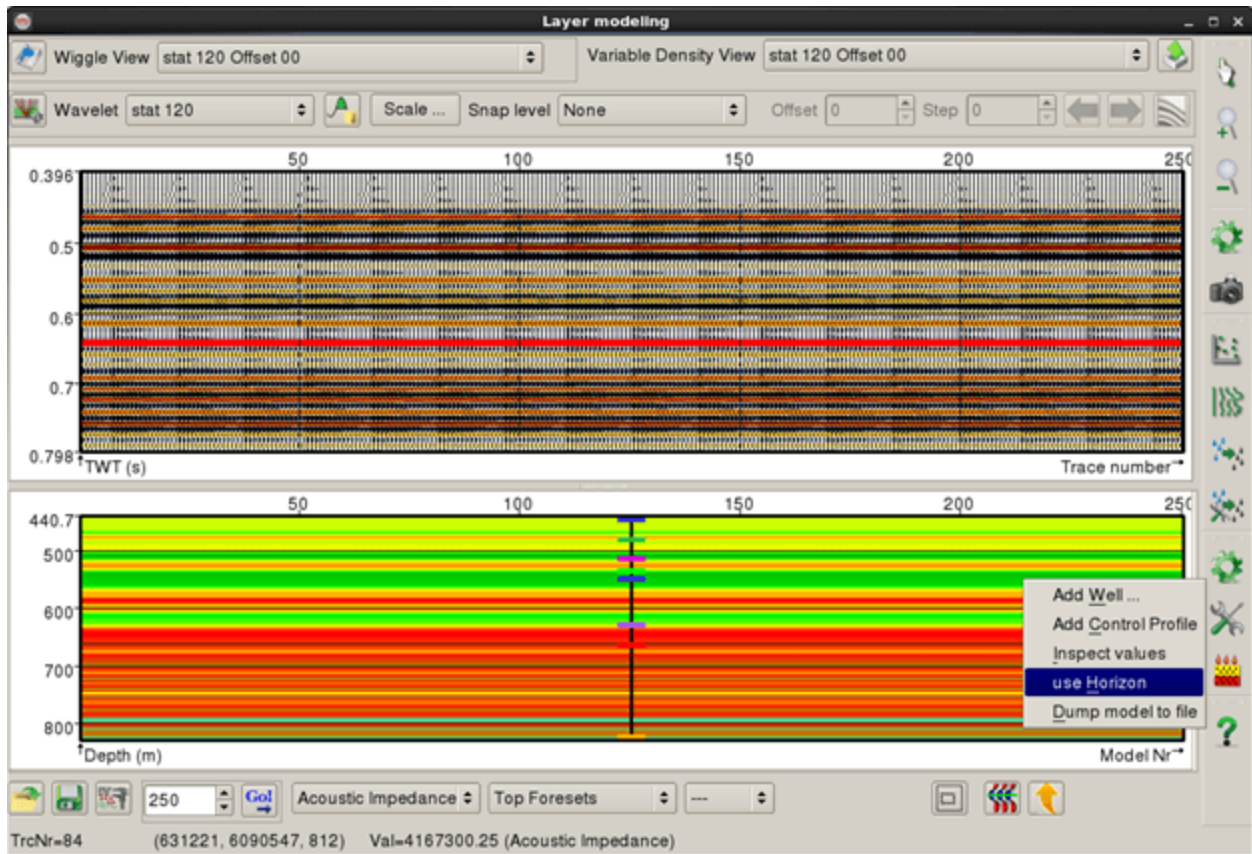
Apply

OK
Cancel
Help

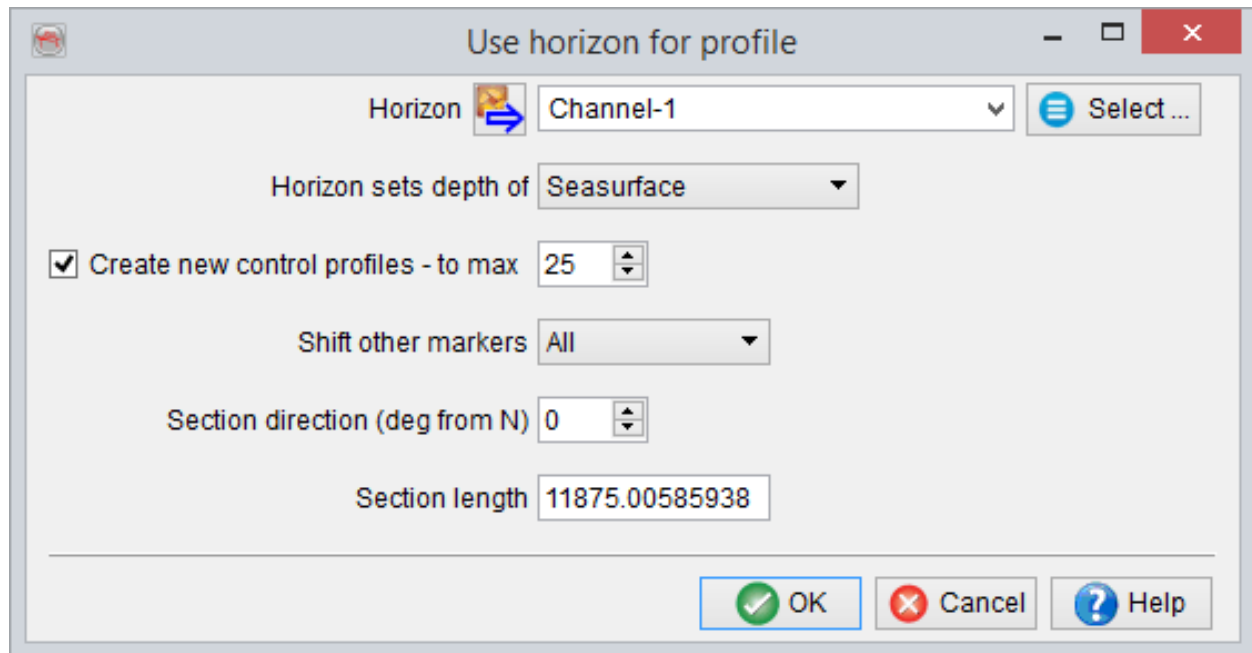


Use Horizon

Existing horizons from the project database can be used to easily and quickly mimic an interpreted (3D or 2D) structure and align the pseudo-wells along it in the profile modeling. This workflow is based on an automated creation of a number of *Control Profiles* that follow the depth trend of the selected horizon. It can be started by right clicking in the bottom part of *Layer modeling* window and selecting use *Horizon* option. The layout of *Use horizon for profile* window depends on the number of wells used in the modeling.

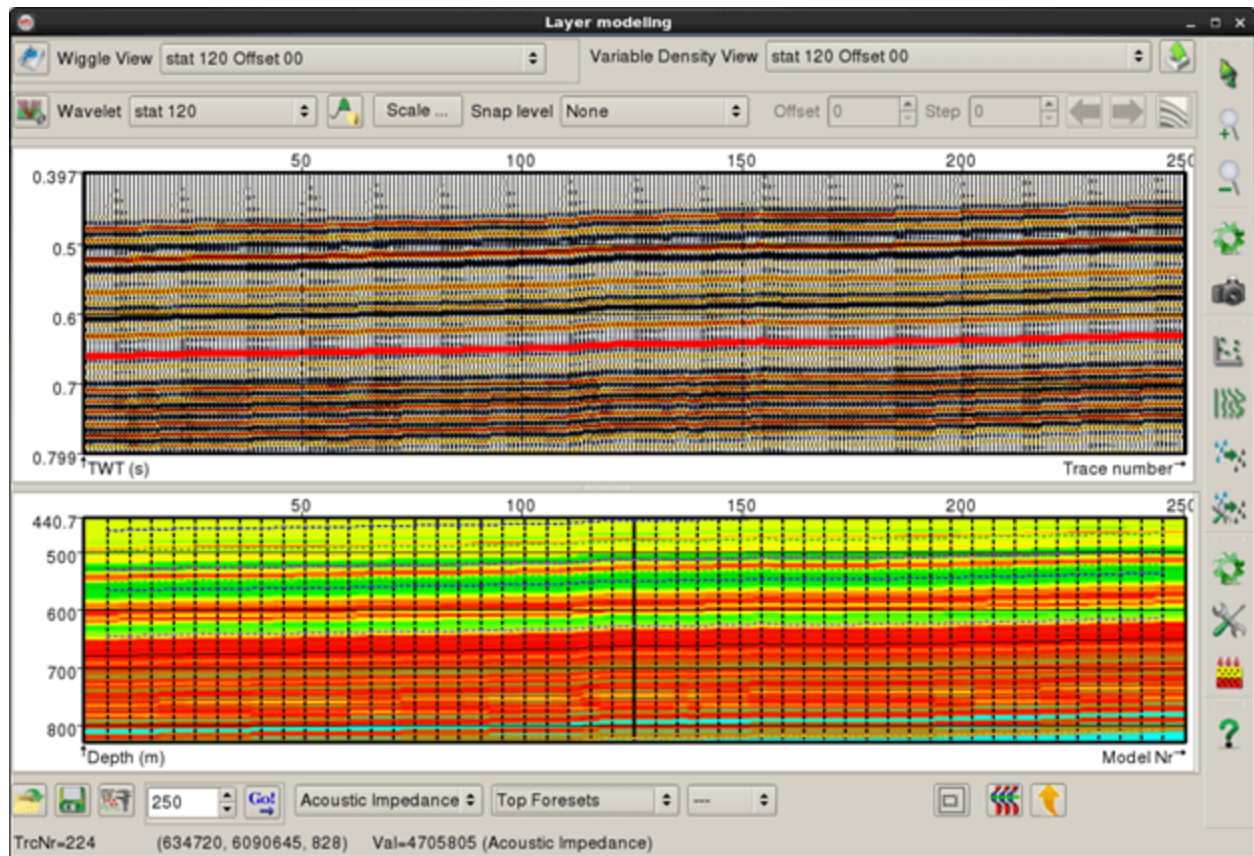


If there is only one (real) well loaded in the layer, the following dialogue window will appear:

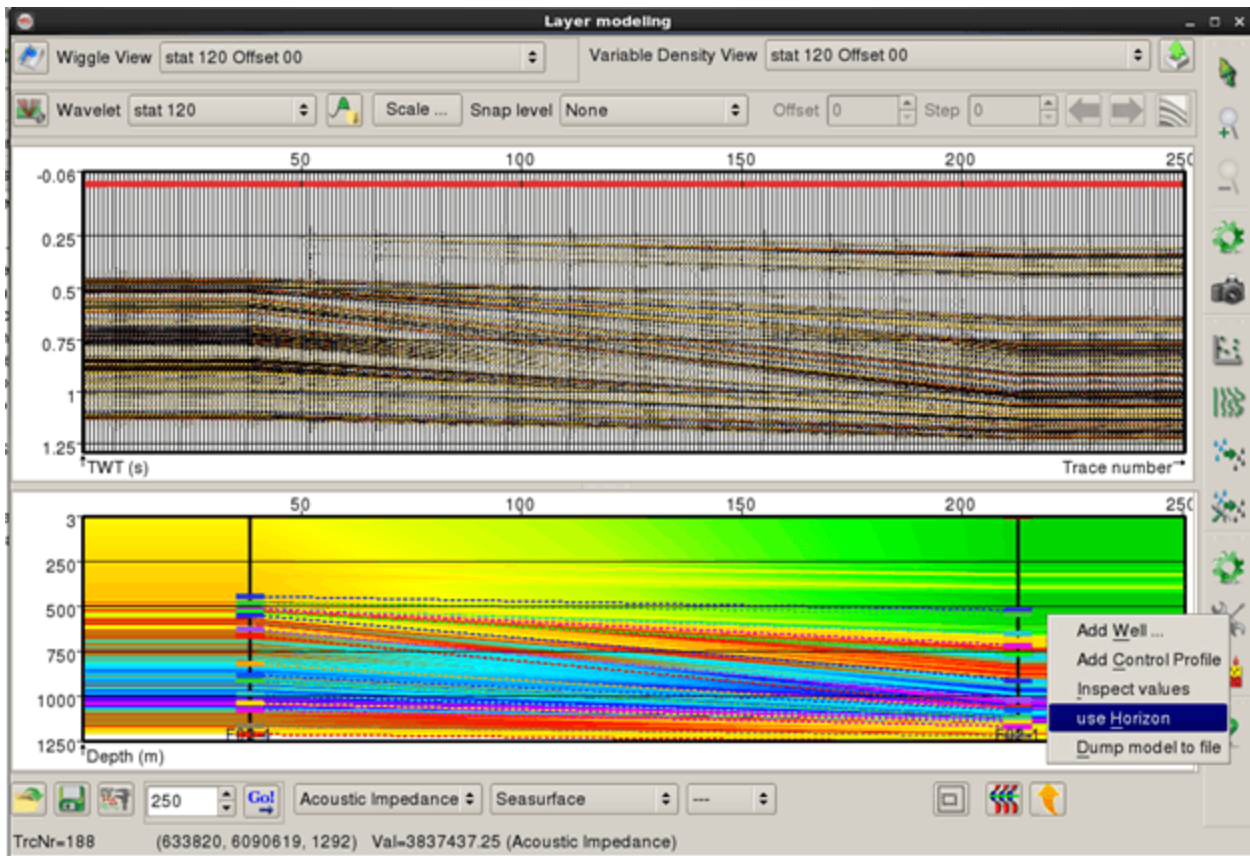


In the *Use horizon for profile* window the user must select a horizon (2D or 3D) on which the simulation of the structure is based. Then, the corresponding formation top has to be chosen. The algorithm adds a user-defined or automatically computed (if unchecked) number of control profiles. At these control profiles, depths of the chosen top are changed using the selected horizon. Linear interpolation of the top depths is performed between them. Handling of other formation tops is set with Shift other markers option to one of the following: *All*, *No*, *Only Above* or *Only Below*. *Section direction* and *section length* should be thought of as an orientation and a length of a 2D profile through the well along which the specified horizon is extracted.

The example below shows the case when a single *F03-4* well is used in the modeling. Note, that it is a zoom-in, red line on synthetic seismic corresponds to *Top Foresets*. Specifying parameters, as shown and described above, and clicking OK re-generates the pseudowells. Control profiles shown with dashed line guide the structural modeling.

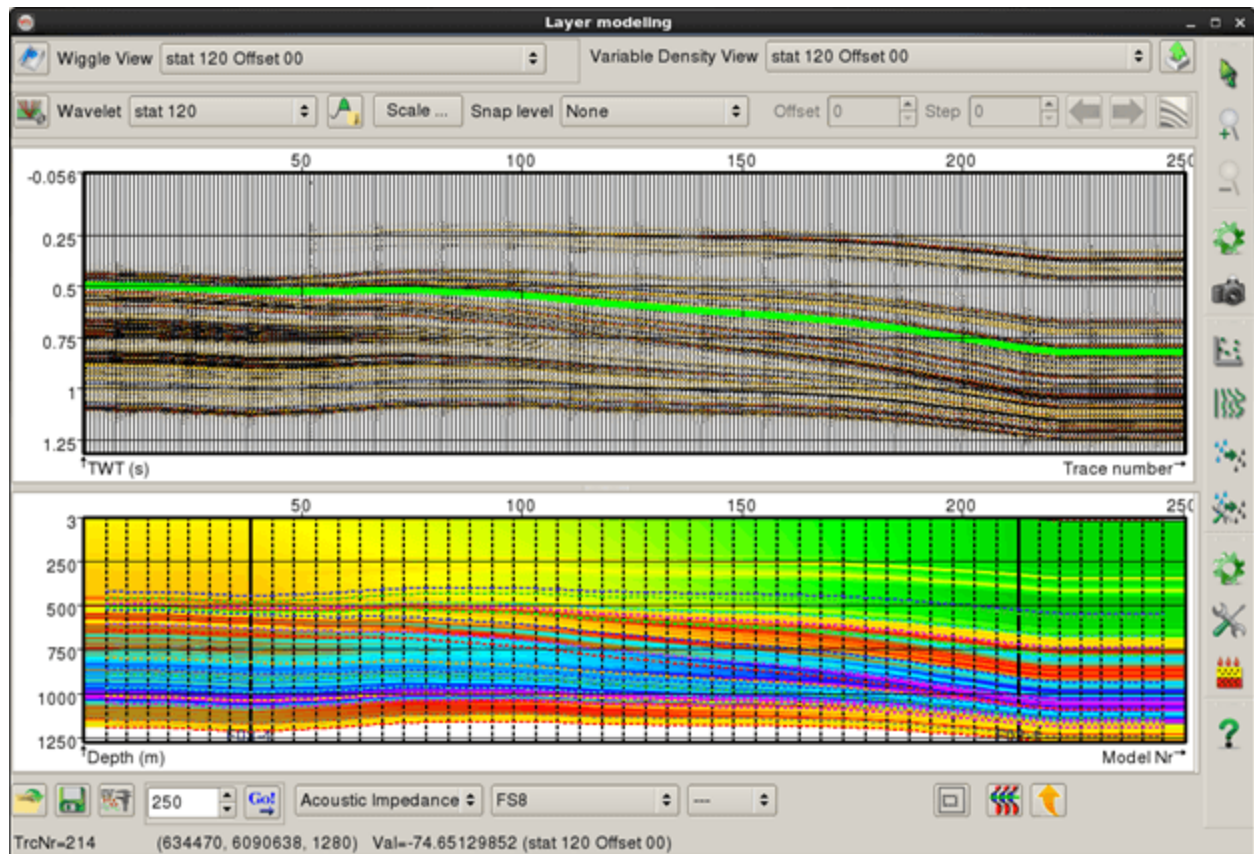


If there are more than one (real) wells loaded in the layer, the use *Horizon* option will bring-up a slightly different dialogue. Two last options from a single well case are eliminated, because well locations determine the direction of the profile along which the specified horizon is extracted.



The screenshot shows the 'Use horizon for profile' dialog box. It contains several configuration options: 'Horizon' set to 'Demo 6 --> FS8', 'Horizon sets depth of' set to 'FS8', 'Create new control profiles - to max' checked with a value of 50, 'Shift other markers' set to 'All', 'Section direction (deg from N)' set to 0, and 'Section length' set to 11875.00585938. Buttons for 'OK', 'Cancel', and 'Help' are at the bottom.

The example below shows the case when two wells, *F03-4* and *F02-1*, are used in the modeling. Parameters are set as in the figure above. Upon clicking OK in the above window, control profiles are added as dashed lines, and pseudowells are re-generated as shown below. The green line on synthetic seismic display corresponds to *FS8* top.



Finally, it is useful to note that depth trends of multiple horizons can be used to build a single profile model, by repeating the above workflow for different horizons and shifting only the corresponding marker each time. Perhaps, in such a case a much quicker option is to use one of the pre-defined workflows called [Model an existing line through well\(s\)](#).

8.4.4 Add, Edit Existing Well

Control points for the profile modeling are added with a right-click where it says "Click to add a well". Before the first control point is added you will need to [specify the list of logs](#) that will be interpolated.

Add the first control point: Blocking a real well to generate the pseudo-well

- Clicking in the bottom part of the main layer modeling window to add a well, will first pop-up the modeling [parameter selection](#) window. Once the process parameters are set, the program is ready to start loading the first real well for generating pseudo-well models. The following dialogue window is used for adding (and later editing) a well.

Fluid column

Content zone: Gas Presence in output: 100.0%

Thickness: Constan 50 m (Meter)

Content zone: - Presence in output: 100.0%



Thickness: Constan 50 m (Meter)

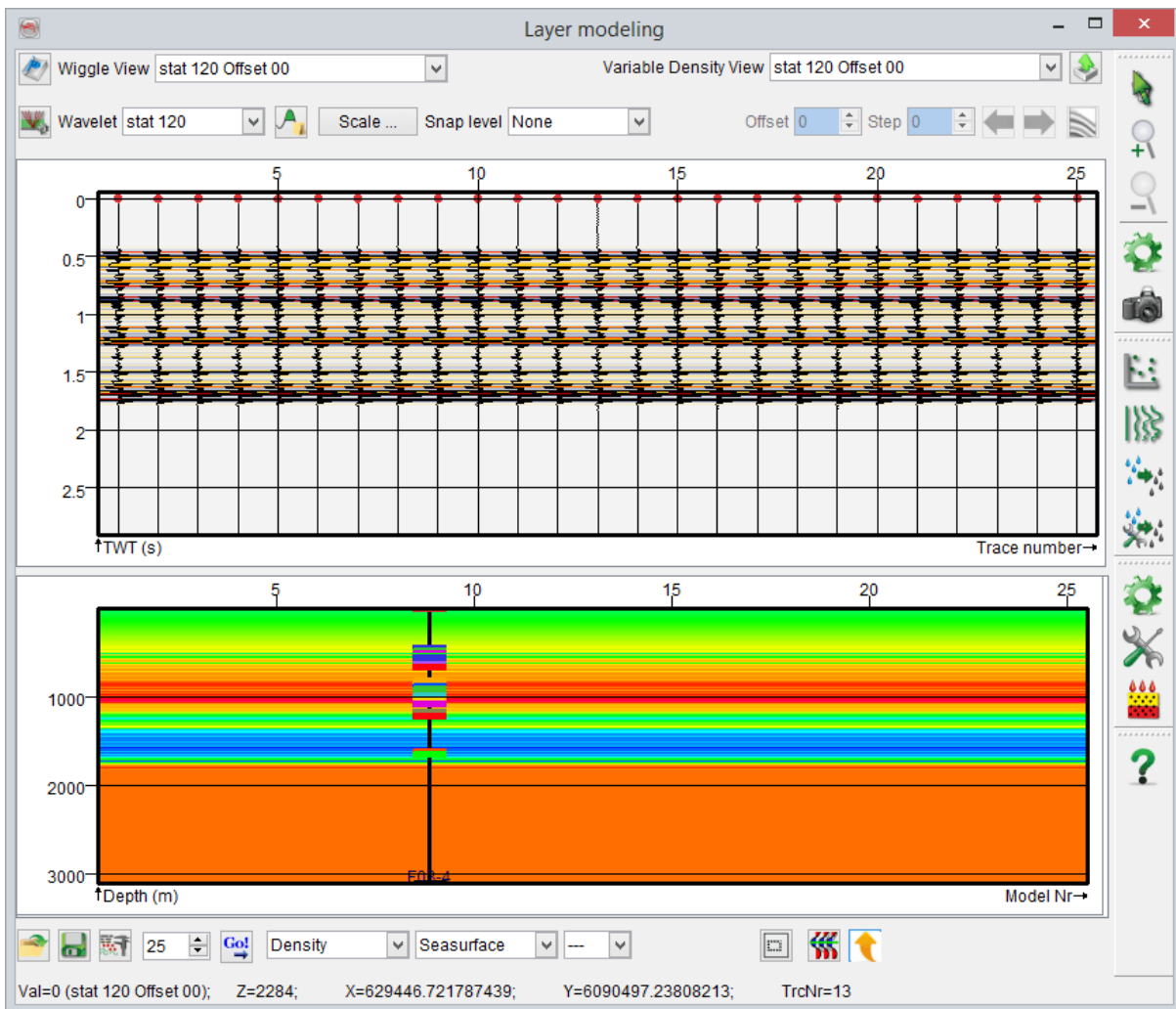
Continue the column into layers below Yes No

OK Cancel Help

Definition of the first control point from well F03-4

The *Relative Position* element specifies where the new well will be added in the profile. Here, 0% corresponds to the left-most end of the profile whereas 100% corresponds to the right-most end. For example, if N is the total number of pseudo-wells to be generated (displayed on the bottom left corner, left of the Go button), using '33%' as the

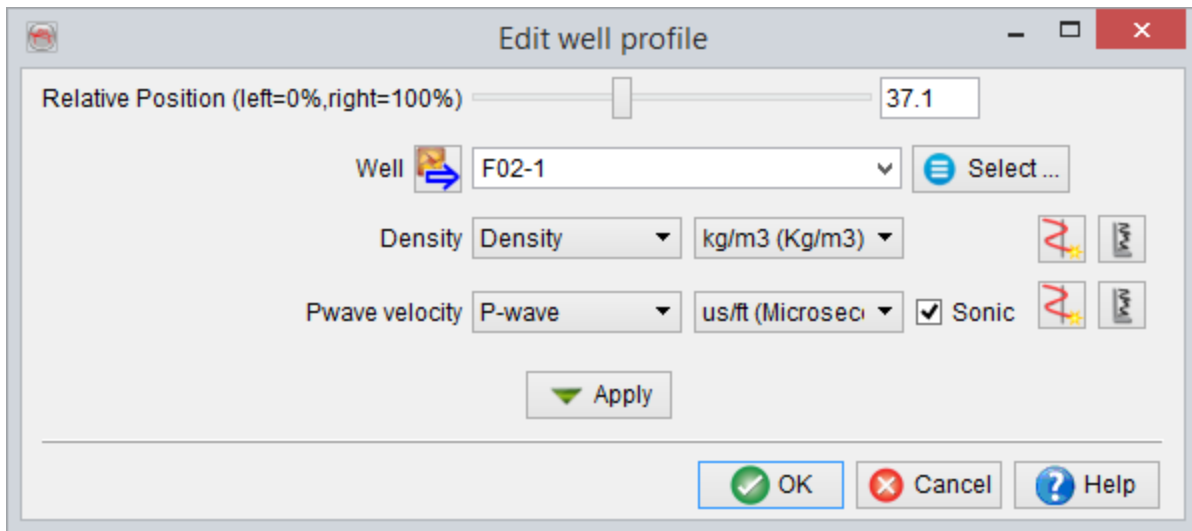
relative position will add the new well at position number Nx33/100. Afterward, select an existing real well from the OpendTect database for the blocking, and a stored log corresponding to various properties, with the corresponding unit. The log units are read from the project database and should not be changed; SynthRock will not convert the units automatically. Here, there is also a functionality to create new input logs from the existing well logs (e.g. Density from Sonic), on the fly, by clicking on . Additionally, a log can be viewed for QC purposes by pressing this  button. Pressing *Ok* or *Apply* will create a number of pseudo-wells by blocking the real well logs. An already added real well is displayed as a solid vertical line.



25 pseudo-wells created from F03-4 well

Editing an existing well:

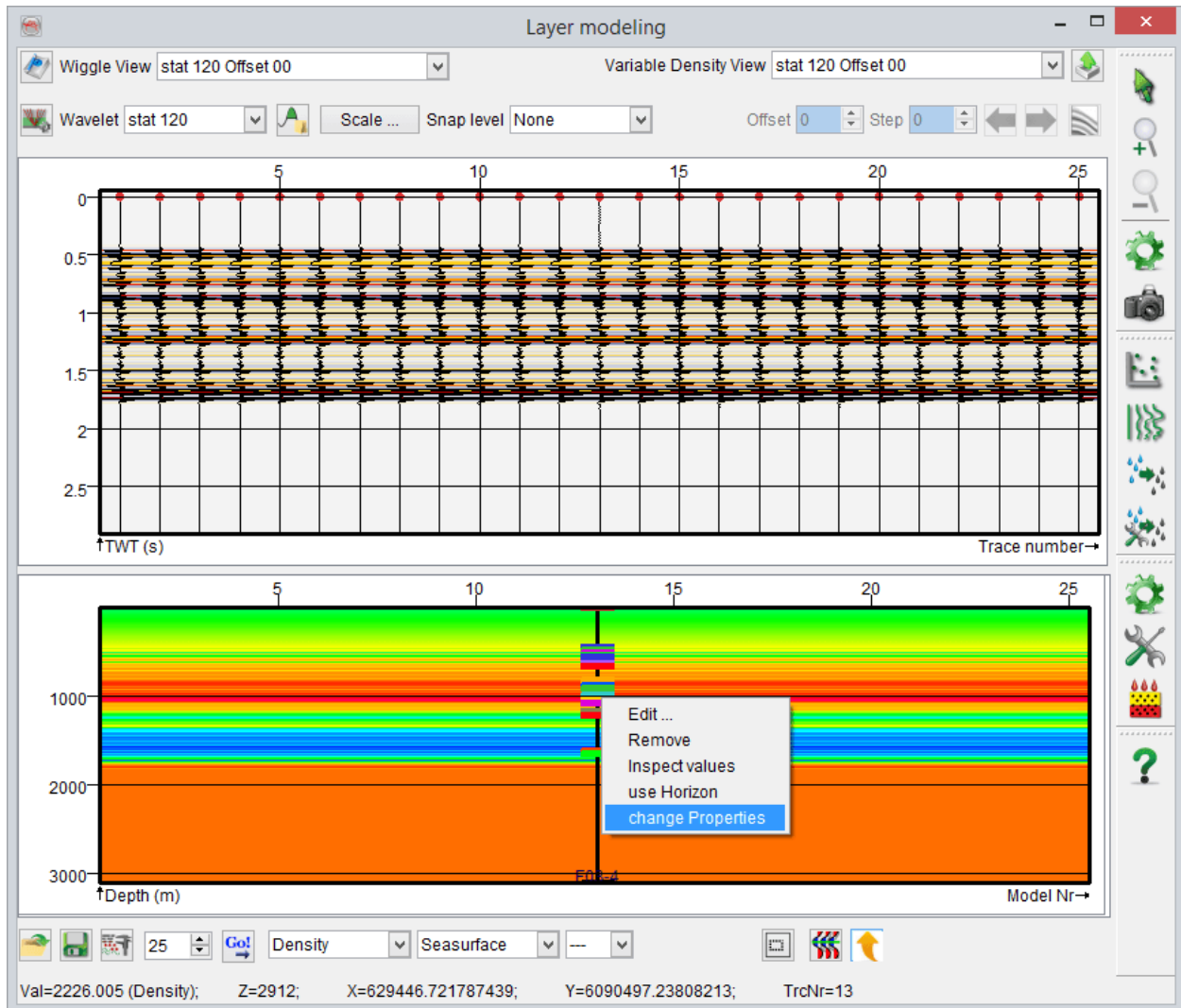
- Using the above mentioned method more wells can be added. The edition of the added (real) well logs or of their relative position, can be done by right-clicking close to the solid vertical line (corresponding to the well) and then selecting *Edit*, to pop-up the dialogue window explained in the section above.



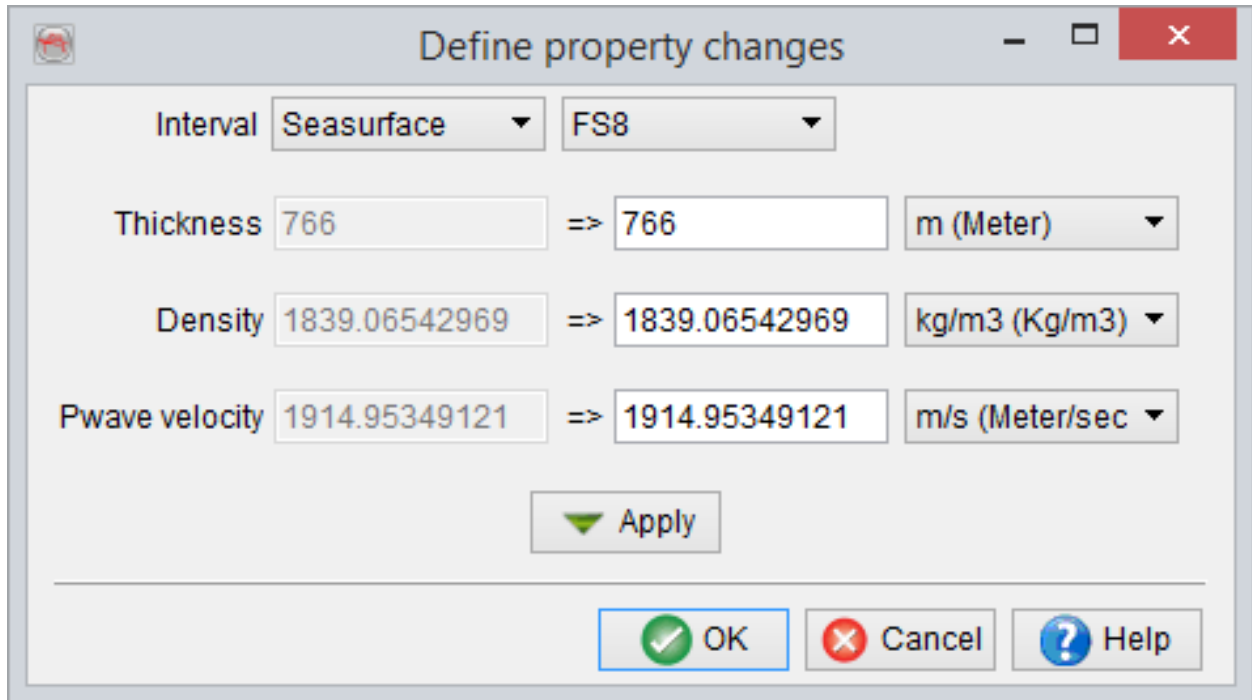
Editing the relative position of well F03-4

8.4.4.1 Define Property Changes

Right clicking on an added well, in the lower part of *Profile* module window, can be used to *Edit Properties* of a part of that well. The image below shows a very simple model in the *Profile* module, having 25 identical pseudo-wells based on one real well. The lower window here displays a Density profile.

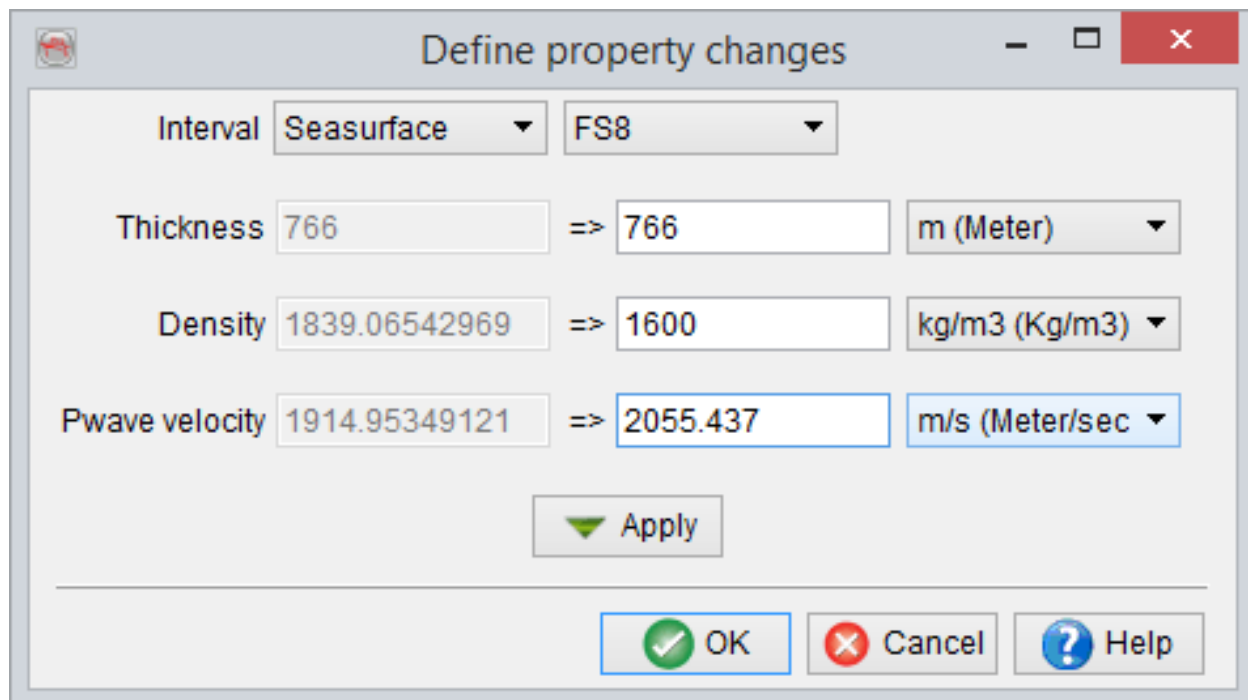


The *Define property changes* dialogue shows thickness and average values of various layer properties (e.g. Density, Vp, Al etc.) in a given (vertical) interval, for a real well.



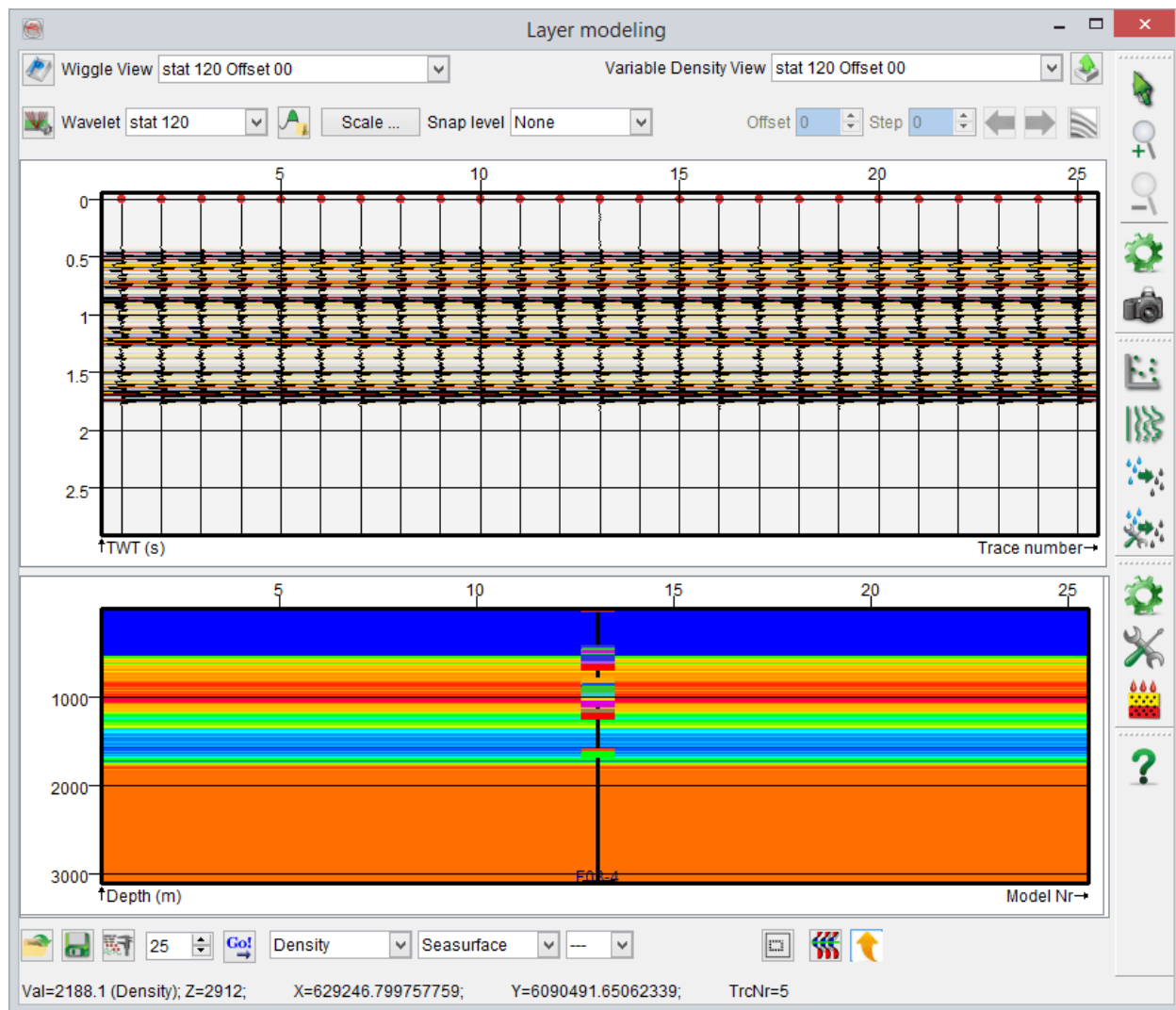
Original average property values in a particular interval

Here, the user can scale all the samples of one or more properties in that interval by specifying a different average value. This essentially means that the individual samples of a property will be scaled, so as to have the user-specified average in the given interval.



Edited average property (Density) values in a particular interval

Pressing *Apply* or *Ok* will update the layer properties of the well, and consequently of the pseudo-wells and their corresponding synthetic seismic data.

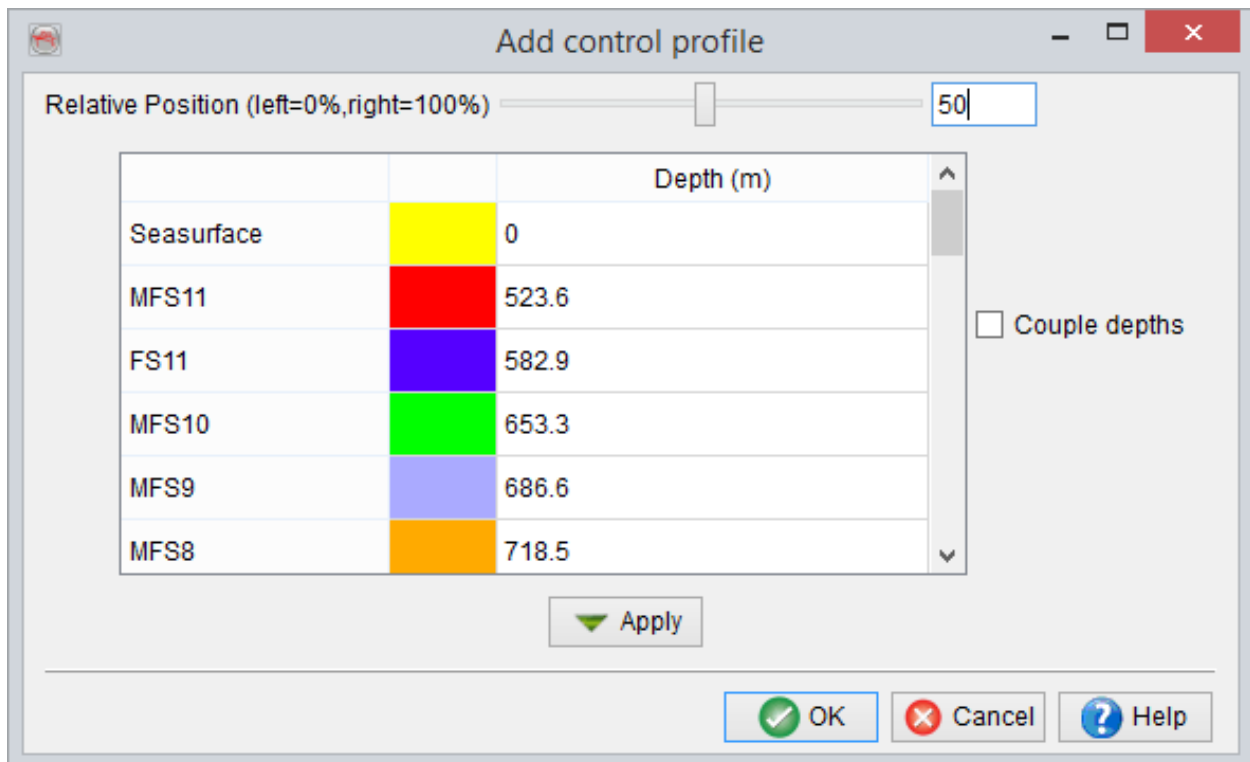


In real world scenarios, where multiple wells are used in modeling, the layer properties for the pseudo-wells between the edited wells and the unedited wells, are recomputed using horizontal interpolation.

8.4.5 Add, Edit Control Profile

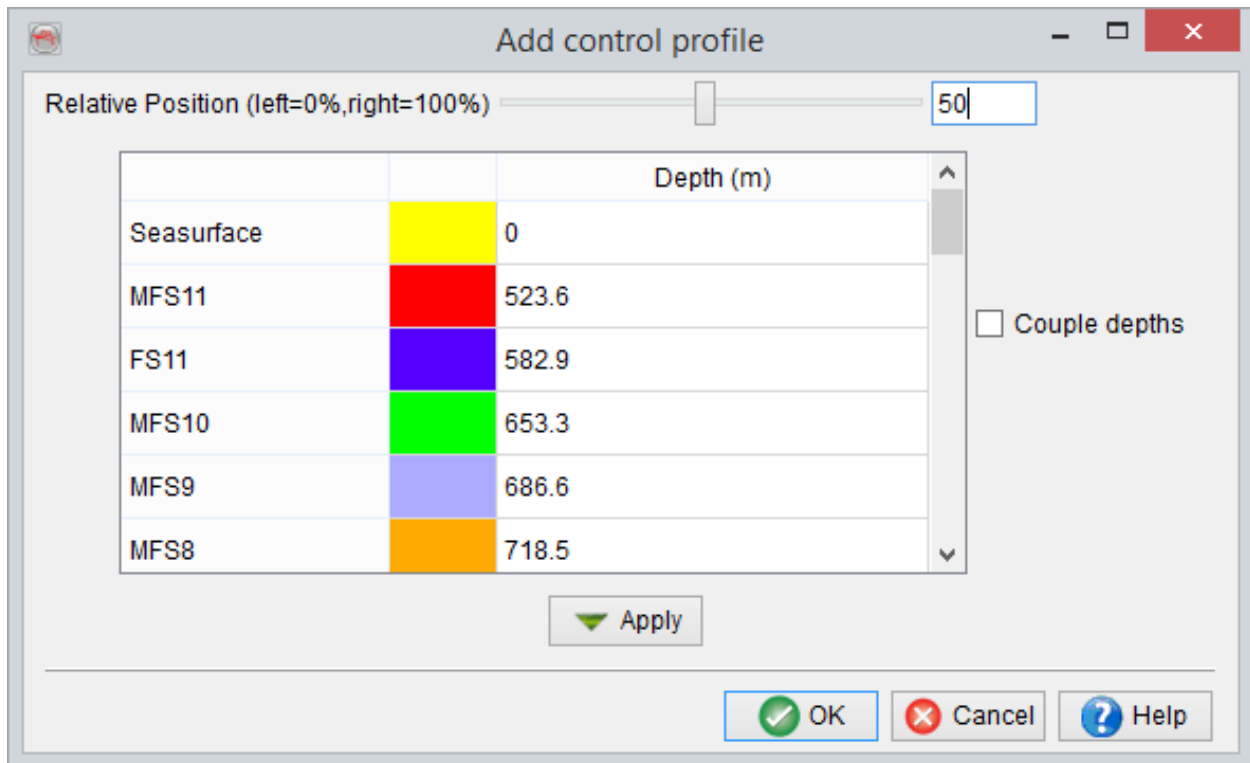
The second type of available control points in the profile modeling, other than the (real) wells are called *Control Profiles*. They are basically used to introduce a structural element in the modeling; to essentially steer the log data interpolation along a well defined structure, which is in fact defined by changing depths of a number of well markers at various positions in the profile.

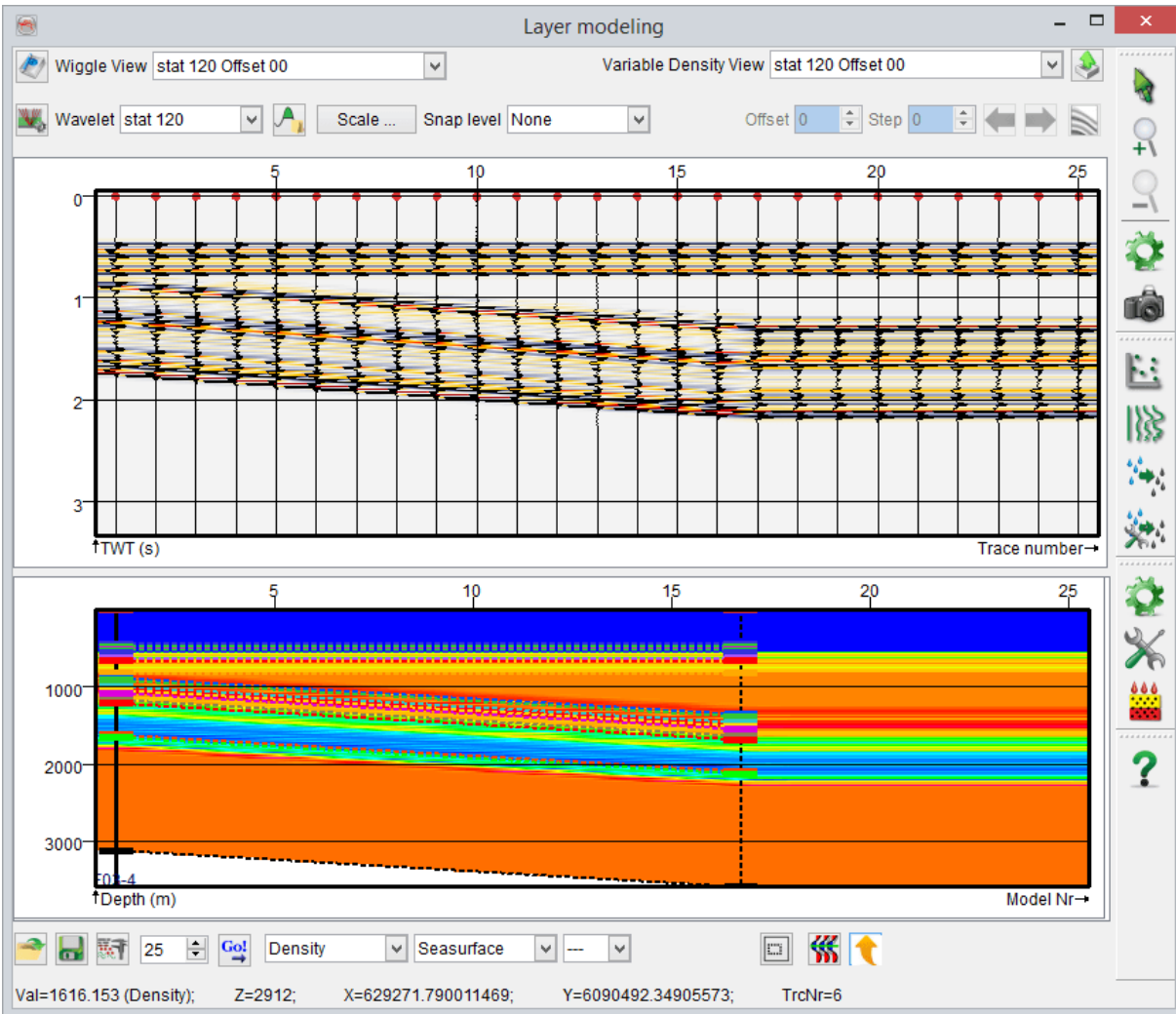
- **Adding a new control profile:** Control profiles are added with a right-click where it says Add Control Profile. Just like for the well based control points, the Relative Position element specifies where the new well will be added in the profile. Here, 0% corresponds to the left-most end of the profile and 100% corresponds to the right-most end. After specifying the position, the depth of various markers can be edited at this position.



Defining a control profile

If *Couple depths* option is kept unchecked, then editing any particular marker's depth is not going to change any other marker's depth; except when post-edition depth of that marker becomes lower than the depth of markers above. If this is indeed the case, all those markers will also be given the (post-edition depth) of the marker in consideration, for example below.

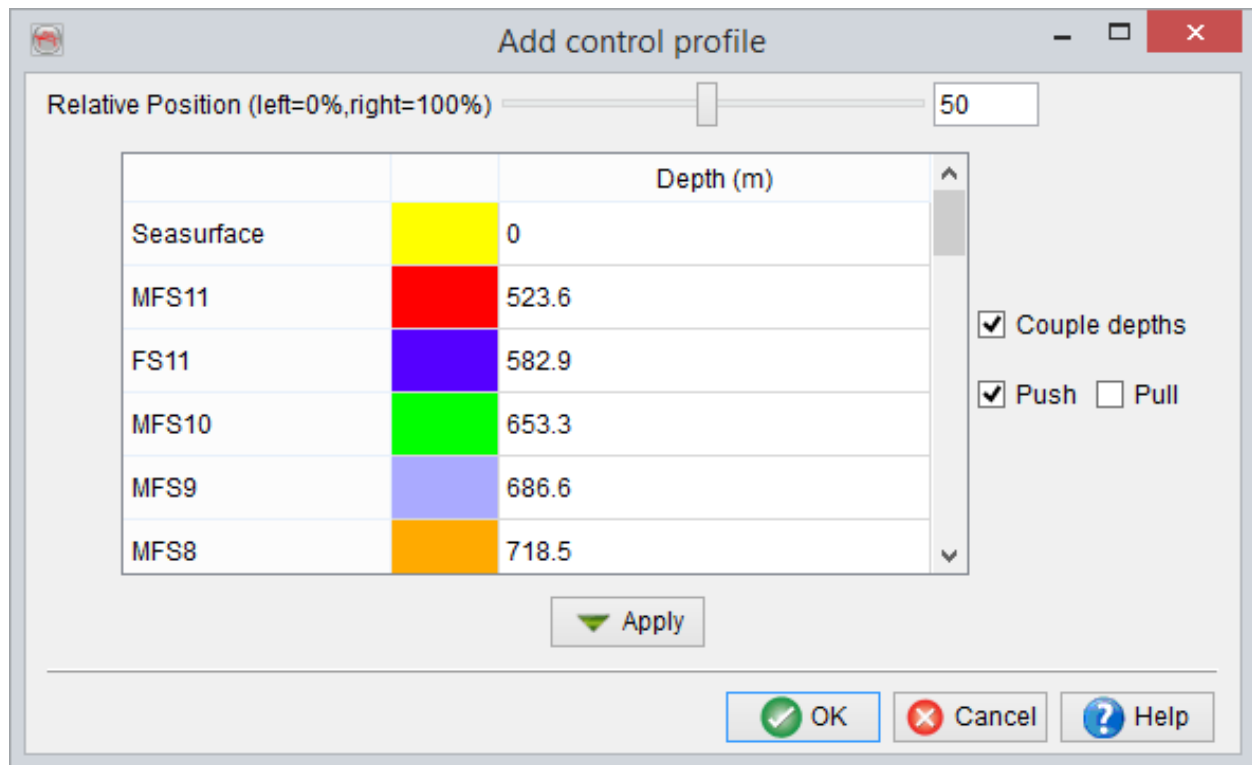


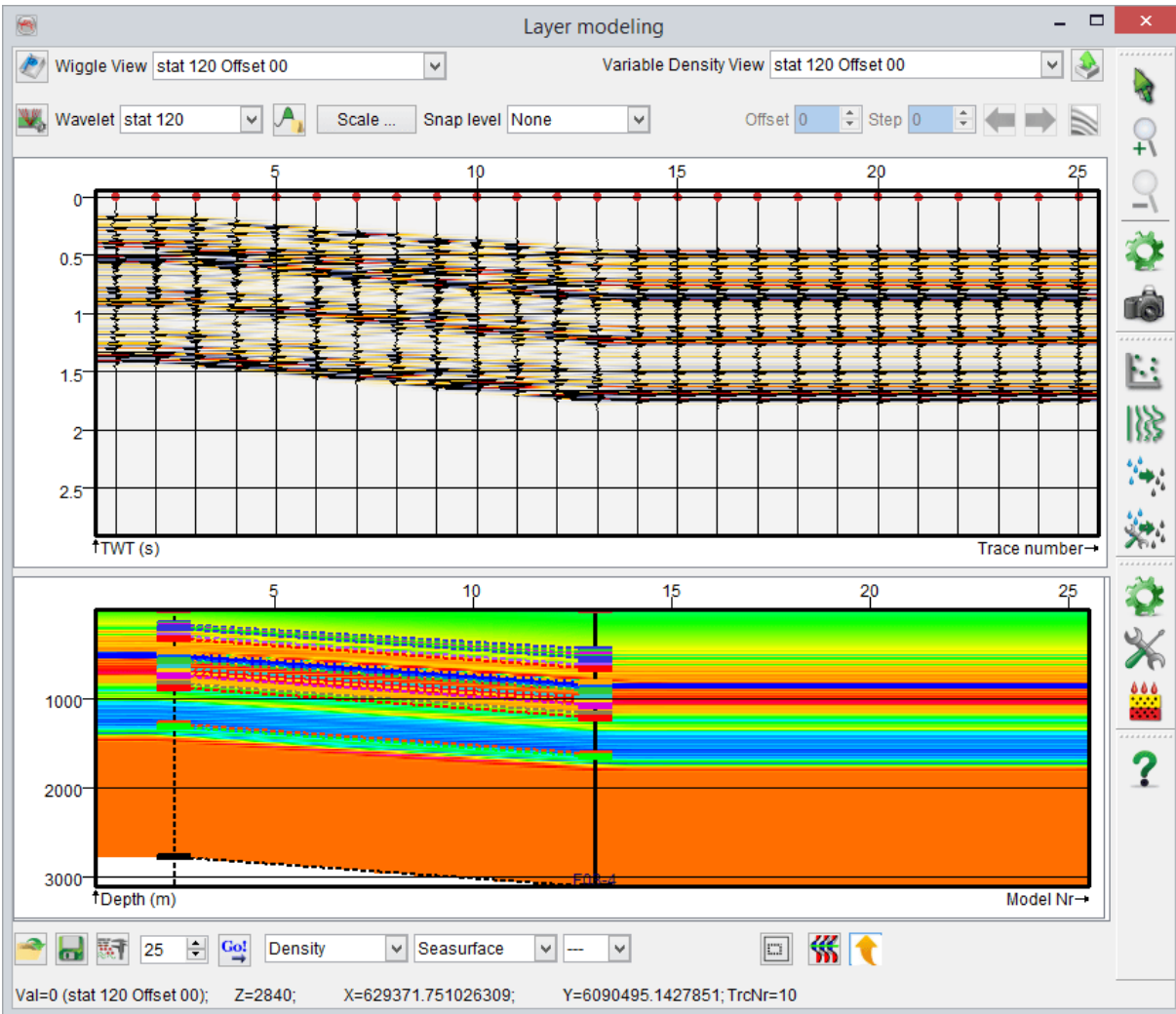


Defining a control profile by editing depths of FS8 marker, without coupling depths of other markers

If *Couple depths* option is checked and only *Push* is selected - Editing any marker's depth is going to push all other markers in the direction of the change, starting from the point of change. For the given example, the direction of change is from deep (initial depth of FS8 marker 535m) to shallow (post-edition depth of FS8 marker 400m) and the point of change is FS8 marker. Thus, all the markers above FS8 have also been

pushed upwards (i.e. from deep to shallow), by the same amount as the FS8 marker itself, i.e. 135 meters.

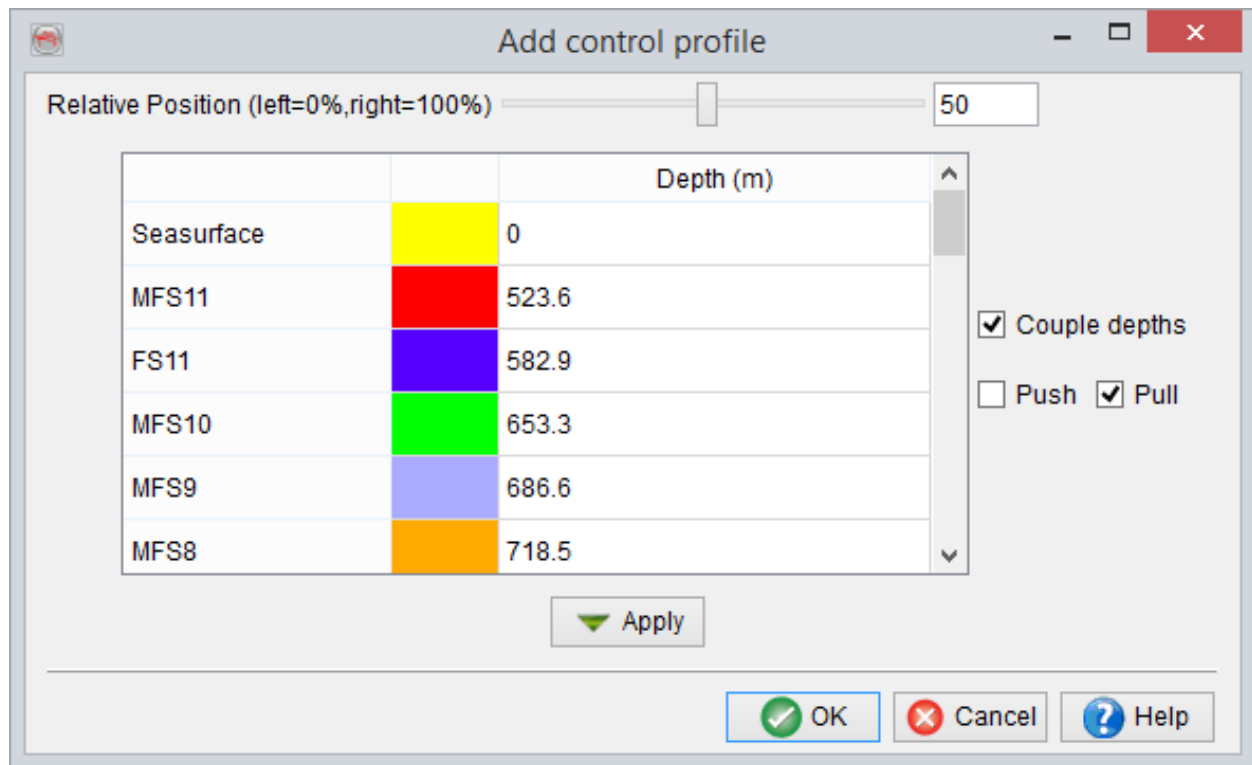


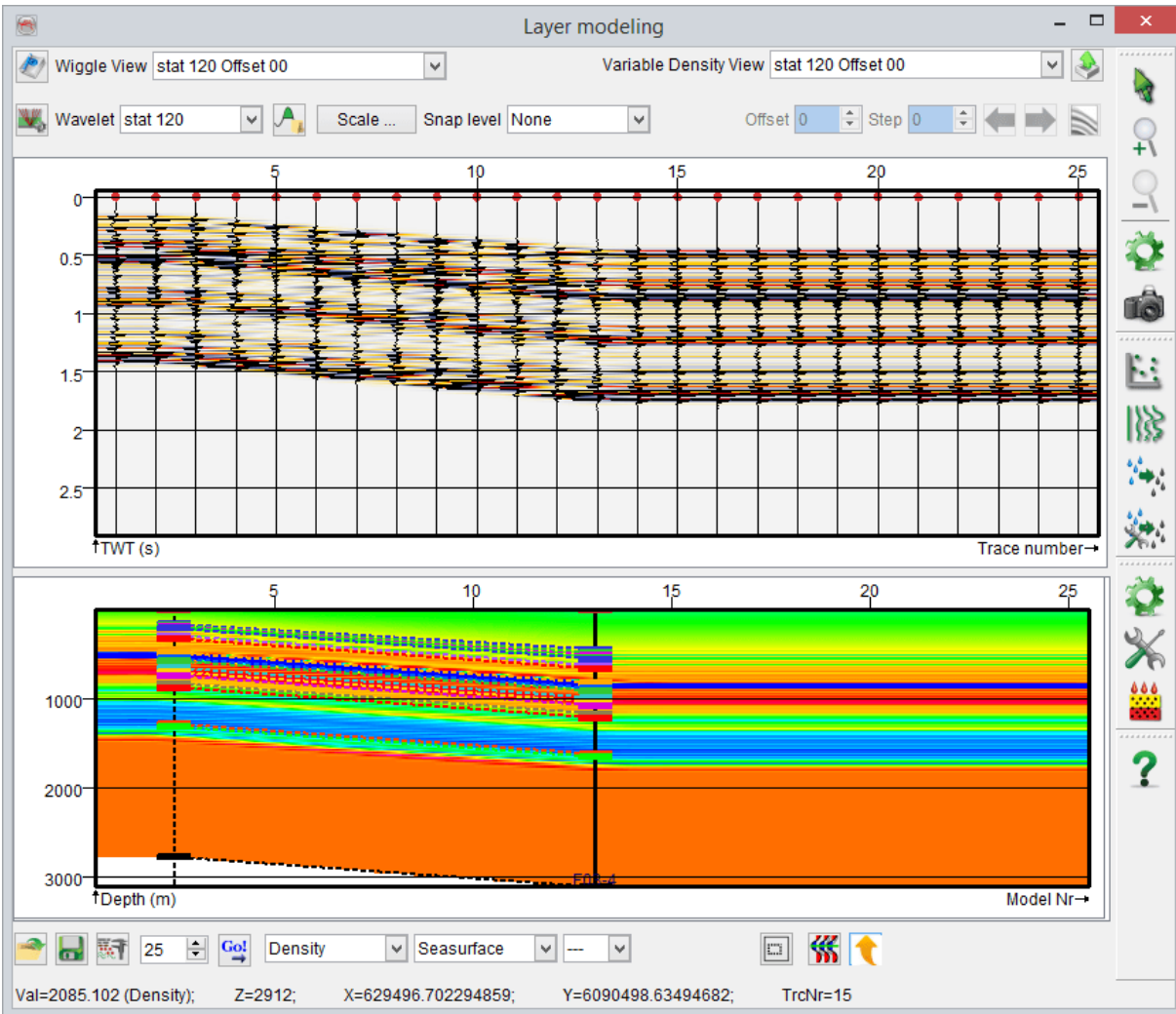


Defining a control profile by editing depths of FS8 marker and pushing markers above upwards

Now if *Couple depths* option is checked but only *Pull* is selected - Editing any marker's depth is going to pull all other markers in the direction of the change, starting from the point of change. For the given example, the direction of change is from deep (initial depth of FS8 marker 535m) to shallow (post-edition depth of FS8 marker 400m) and the point of change is FS8 marker. Thus, all the markers below FS8 have been pushed upwards (i.e. from deep to shallow), by the same amount as the FS8 marker itself, i.e.

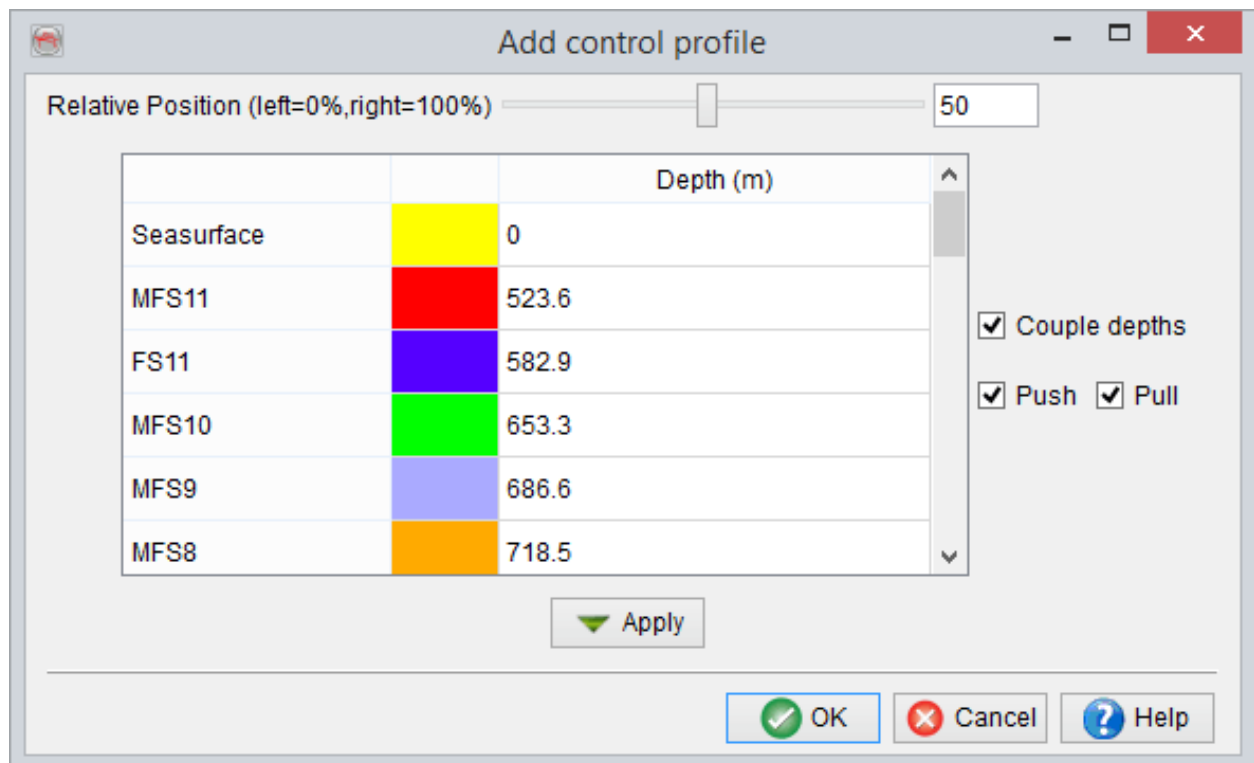
135 meters. It should be noted here that the change in the depths of markers above FS8 is because of the aforementioned exception.

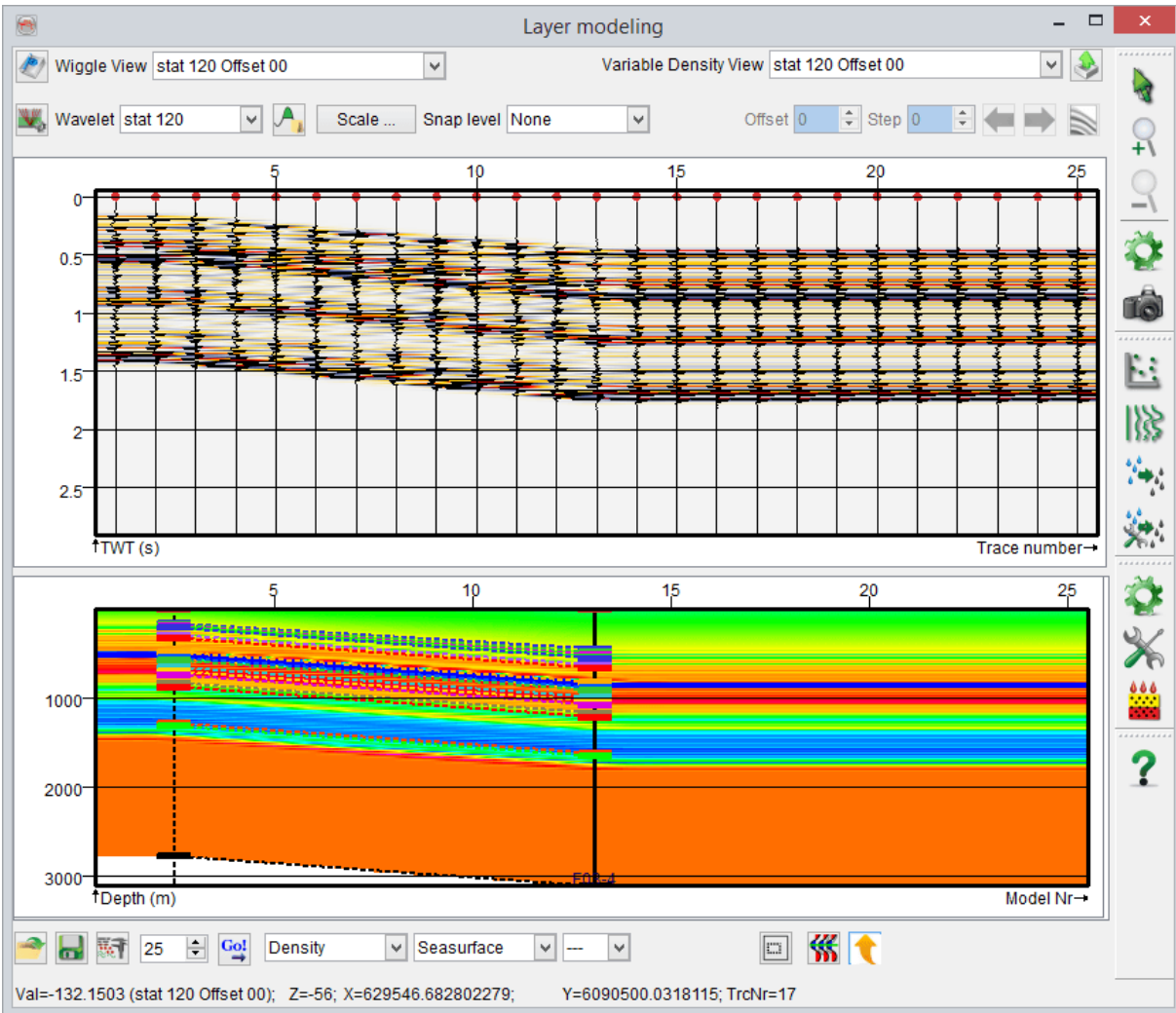




Defining a control profile by editing depths of FS8 marker and pulling markers below upwards

Finally, if *Couple depths* option is checked and both *Push* and *Pull* are selected - Editing any marker's depth is going to move (both push and pull) all other markers in the direction of the change. For our example, the direction of change is from deep to shallow. Thus, all the markers have been moved upwards by the same amount as the FS8 marker itself, i.e. 135 meters.





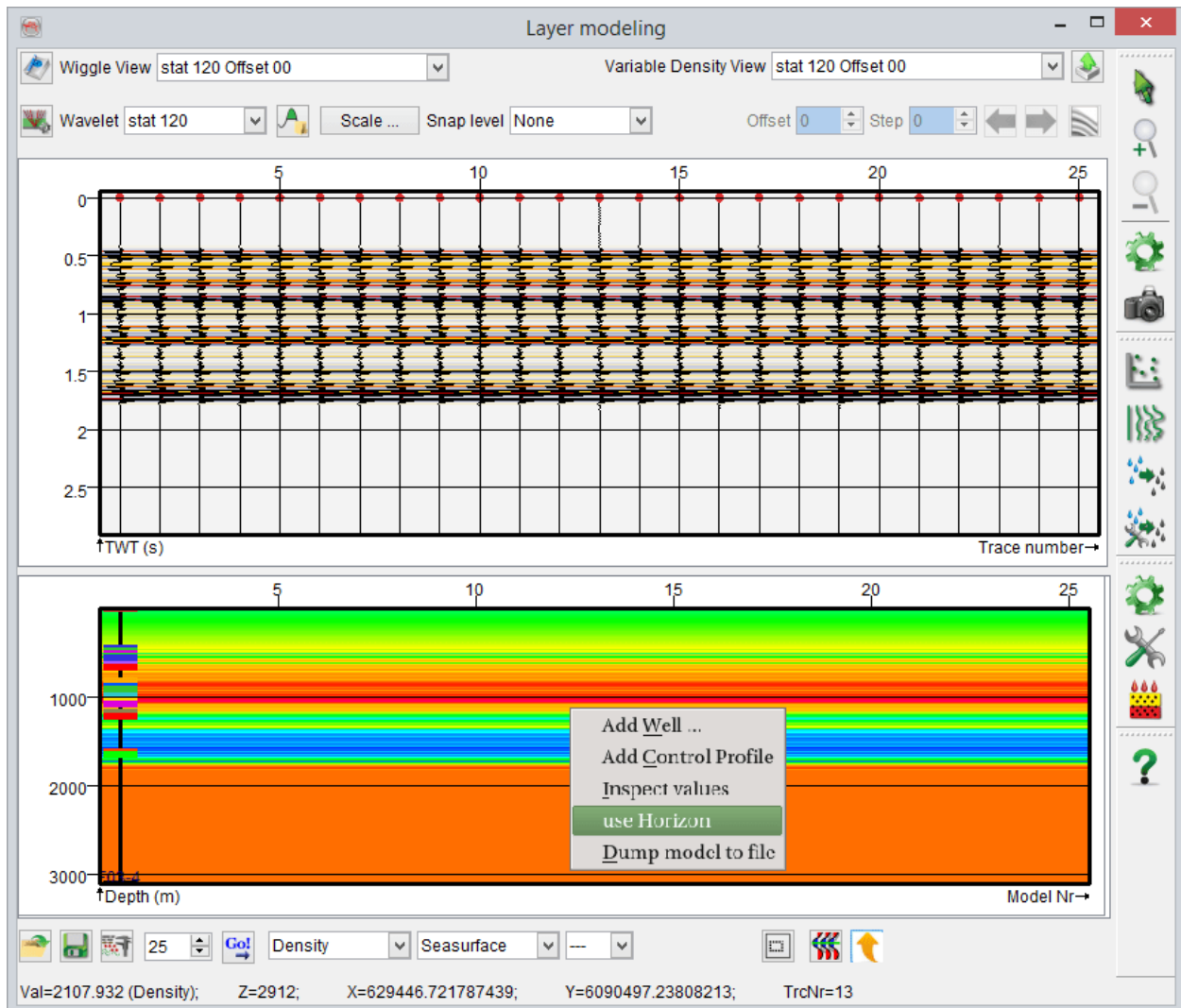
Defining a control profile by editing depths of FS8 marker and moving all other markers upwards

The added control profiles are displayed as dashed vertical lines.

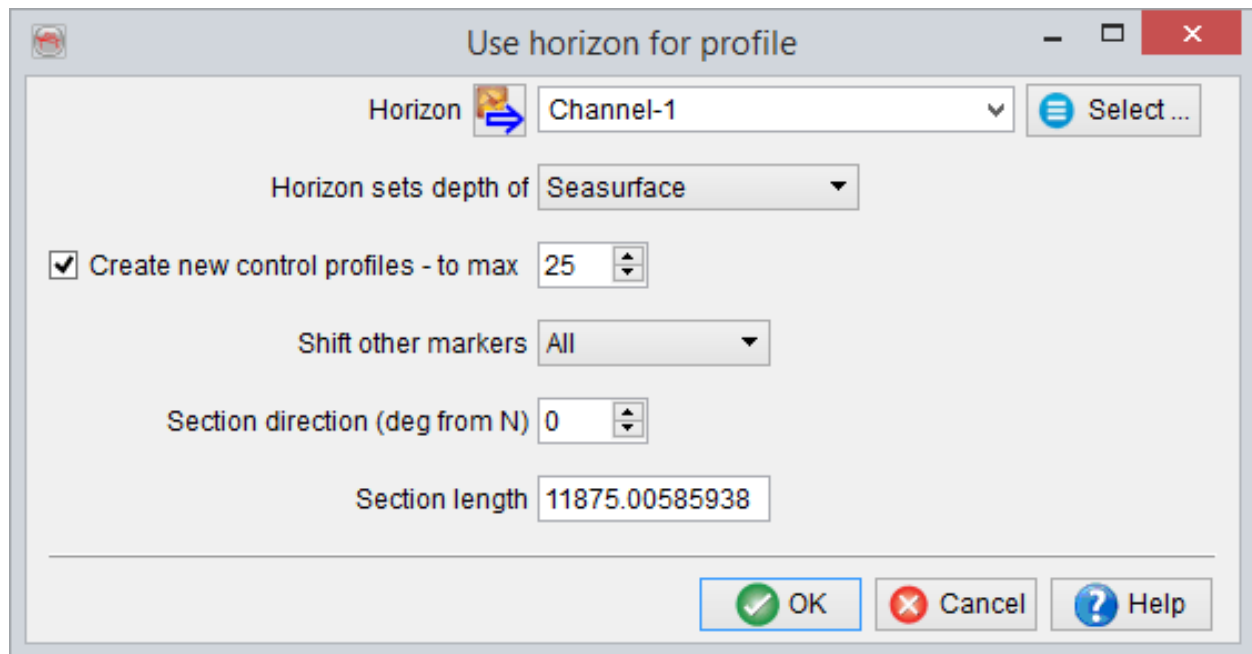
- **Editing an existing control profile:** The edition of an existing control profile is done by right-clicking close to the dashed vertical line (corresponding to the control profile) and then selecting Edit, to pop-up a dialogue window similar to the one explained in the section above. Just like adding a new control profile, both relative position and well marker depths can be edited, of an existing control profile.

8.4.6 Use Horizon

Existing horizons from the project database, can be used to easily and quickly mimic an interpreted (3D or 2D) structure and align the pseudo-wells along it, in the profile modeling. This is essentially achieved by an automated creation of a number of *Control Profiles*, that follow the depth trend of the selected horizon. This can be achieved by right clicking on the bottom and selecting *Use Horizon*.



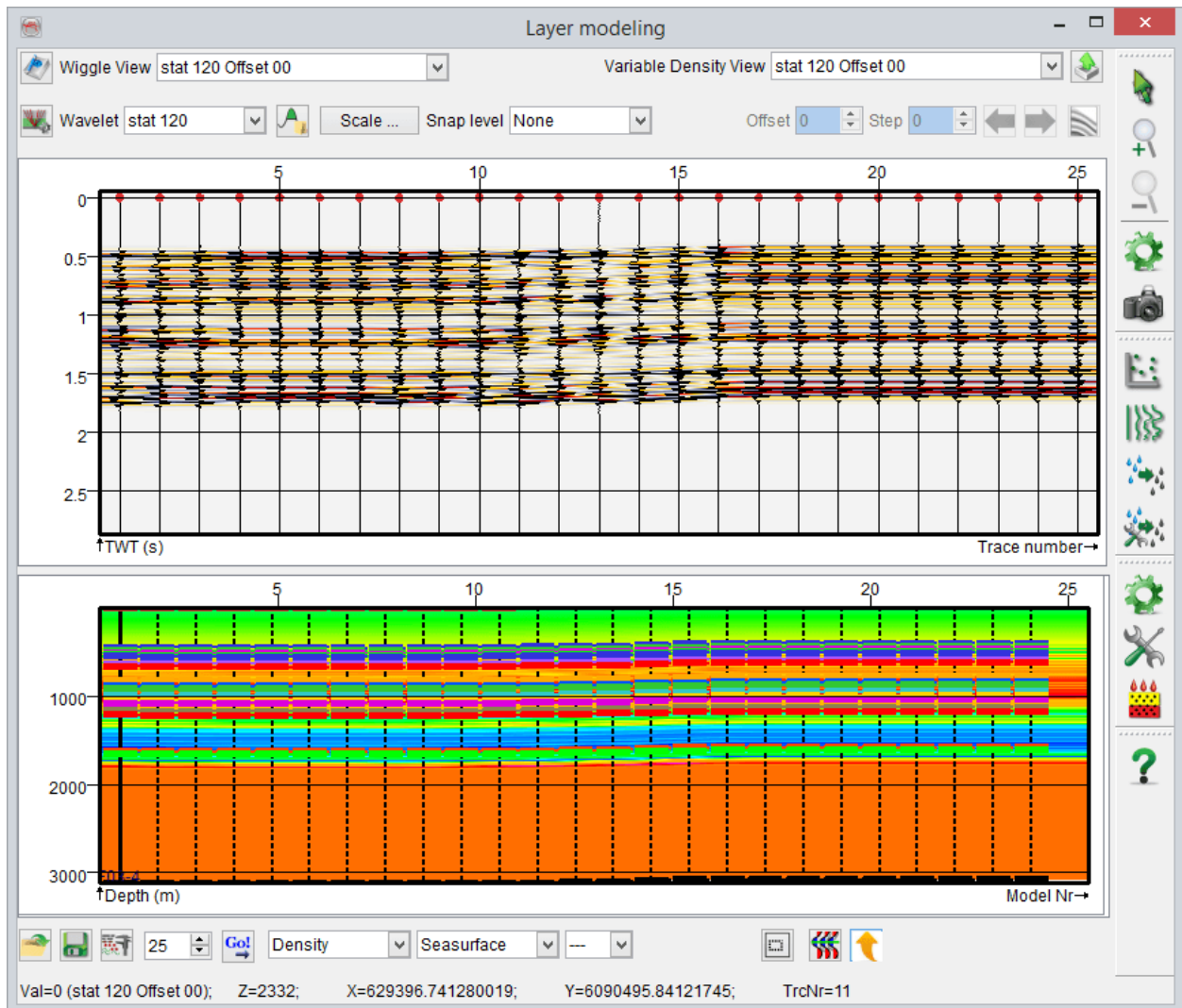
If there is only one (real) well loaded in the layer, the following dialogue window will appear after clicking the link.



Here, the user can select an appropriate horizon on which the simulation of the structure will be based. This selected time horizon is converted into depth using the velocity-depth trend of the pseudo-well models.

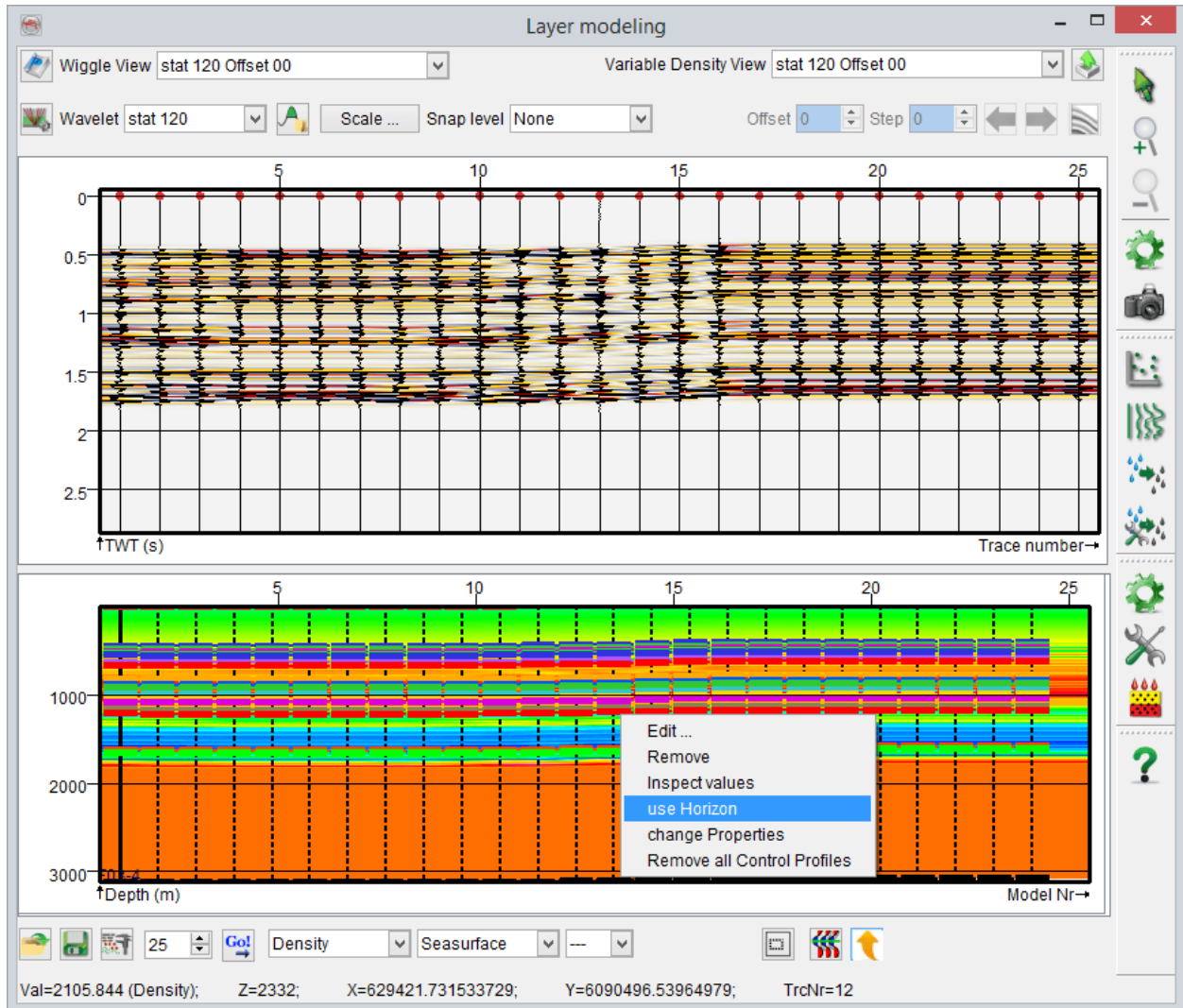
Next thing is to specify the corresponding marker, whose depths will be changed using the depth values of the (depth-converted) horizon. Afterward, there is an option to limit the maximum number of new control profiles to be added, in between the existing pseudo-wells. It is actually through the creation of these control profiles, the program effectively changes the depth of the marker, as to follow the depth trend of the selected horizon. Further, it is possible to also shift other markers relative to the marker in consideration. 'All markers' or 'Markers only above' or 'Markers only below' or 'No marker' are the options available here.

Finally, it is possible to use the depth trend of the horizon along a (pseudo) section, oriented in a particular direction (relative to North) and extending upto a certain distance from the well location. The *Section direction* is specified in Degrees relative to North and the *Section length* in X/Y units of the survey. Below is an example of a pseudo-well profile based on 'Demo0 FS4' horizon.



Using 'Demo0 FS4' horizon to create pseudo-well profile

If there are more than one (real) wells loaded in the layer, for example below, Use Horizon option will bring-up a slightly different dialogue.

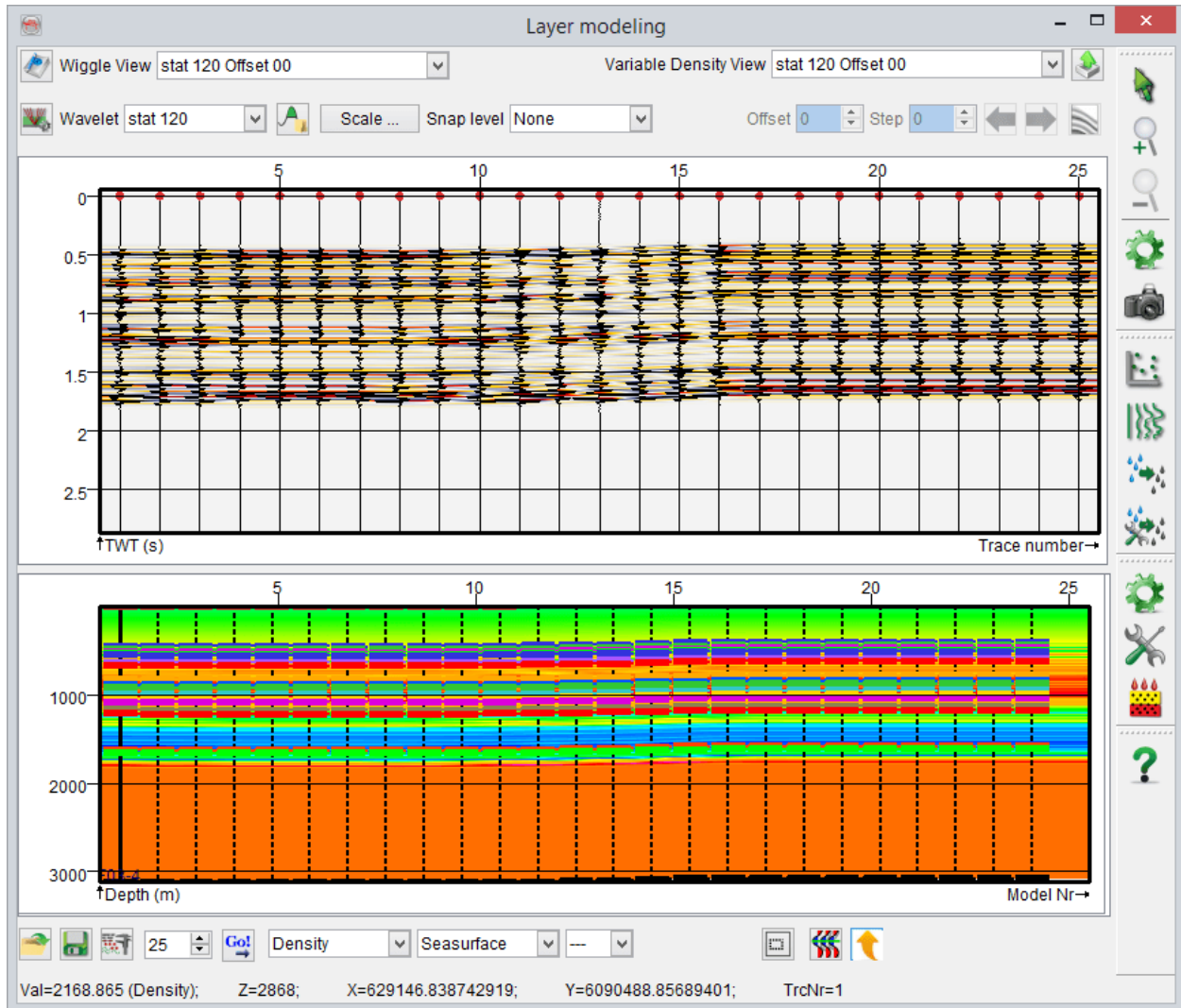


The screenshot shows the 'Use horizon for profile' dialog box. It contains the following fields and options:

- Horizon: Channel-1 (with a 'Select ...' button)
- Horizon sets depth of: Seasurface
- Create new control profiles - to max: 25
- Shift other markers: All
- Section direction (deg from N): 0
- Section length: 11875.00585938


At the bottom, there are buttons for 'OK', 'Cancel', and 'Help'.

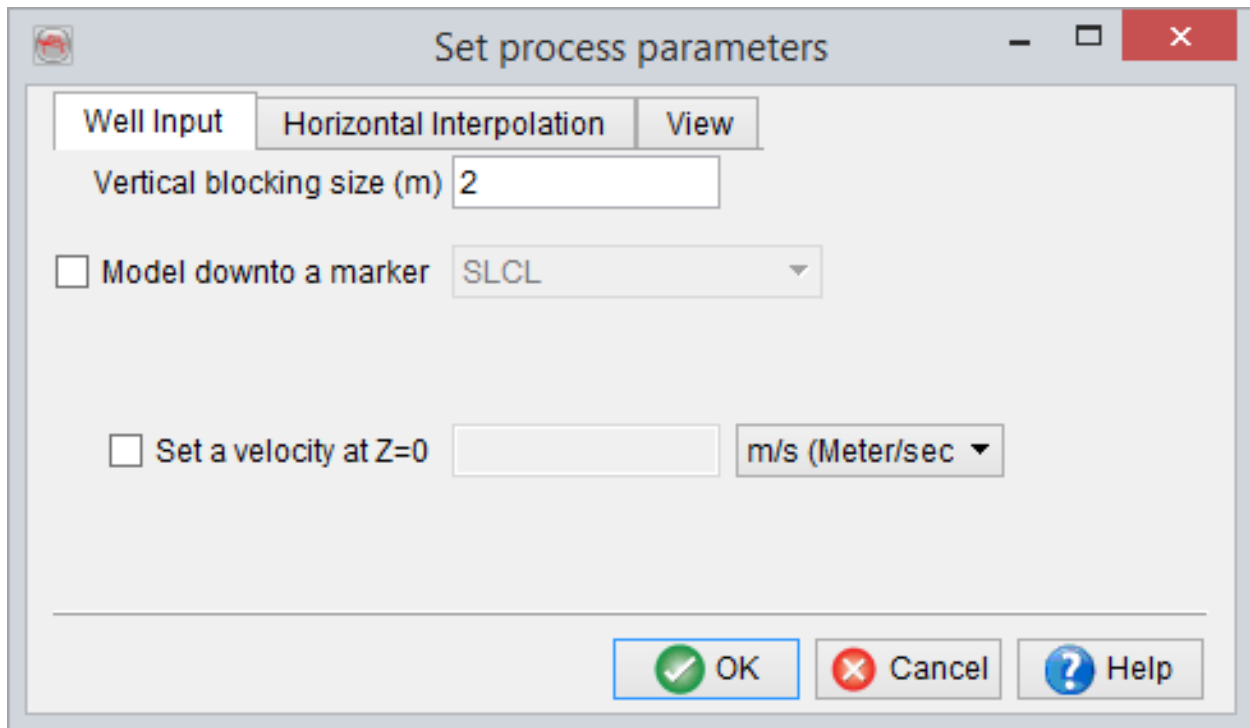
Here, all the options are same as before, except that there is no need to specify the *Section direction* and the *Section length*. This is simply because the program will automatically compute the structural trend in depth of a particular horizon, between the wells (see below).



Finally, it is useful to note that depth trends of multiple horizons can be used to build a single profile model, by repeating the above work-flow for different horizons and perhaps shifting *only* the corresponding marker each time.

8.4.7 Set Process Parameters

The layer modeling settings and various annotations in the profile module can be edited by setting the process parameters. The Set process parameters window can be accessed by clicking on , from the vertical list of icons on the right hand side of the layer modeling window. Also, clicking in the bottom part of the main layer modeling window to add the very first well, pops-up this modeling process parameter selection window.



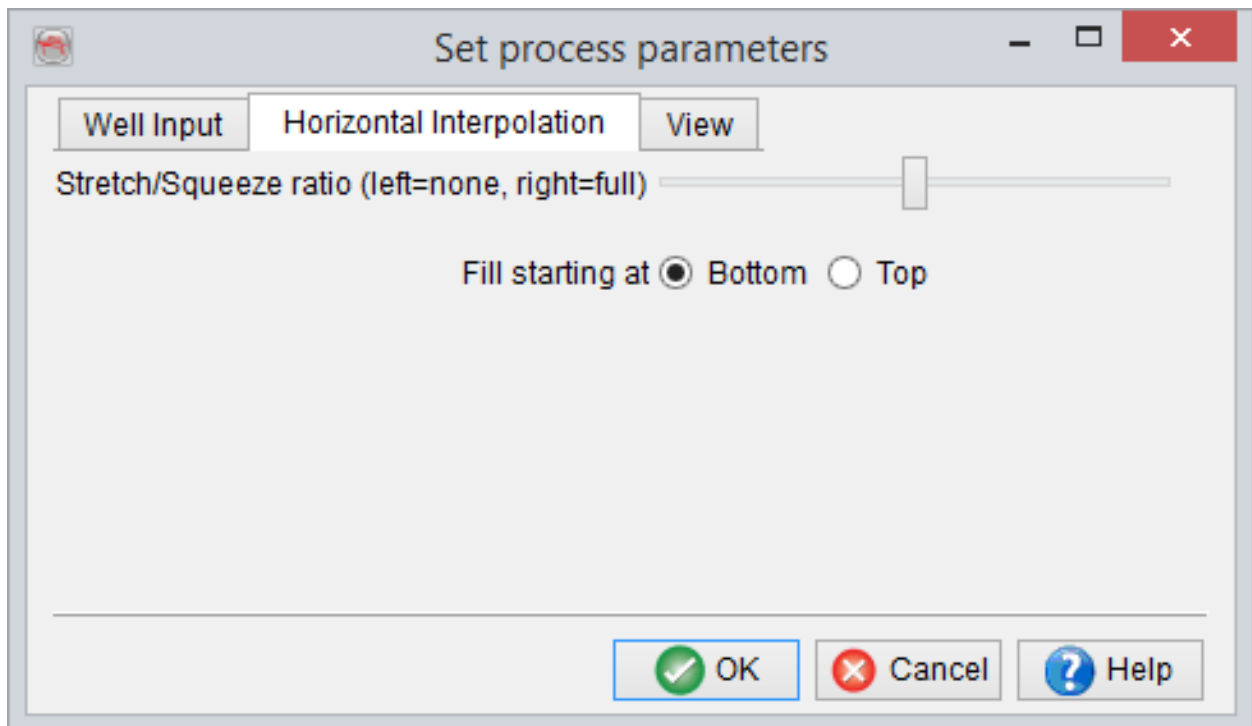
In the first tab (above) of the parameter selection window, a *Vertical blocking size* for various logs can be specified. Optionally, the model can be limited *down* to a particular well marker with some additional depth below it. Further, it is also possible to set a user defined surface velocity (i.e. velocity at Z=0). This options particularly comes in handy, when the input (real) Vp/Sonic logs do not start from the surface. Using this option, a

reasonable overburden velocity model can be created, as the program will interpolate the velocity information between the surface (i.e. $Z=0$) and the first velocity sample in the well logs. If this option is not used, the top most velocity value from the well logs will be assigned to all the layers above, up to the surface.

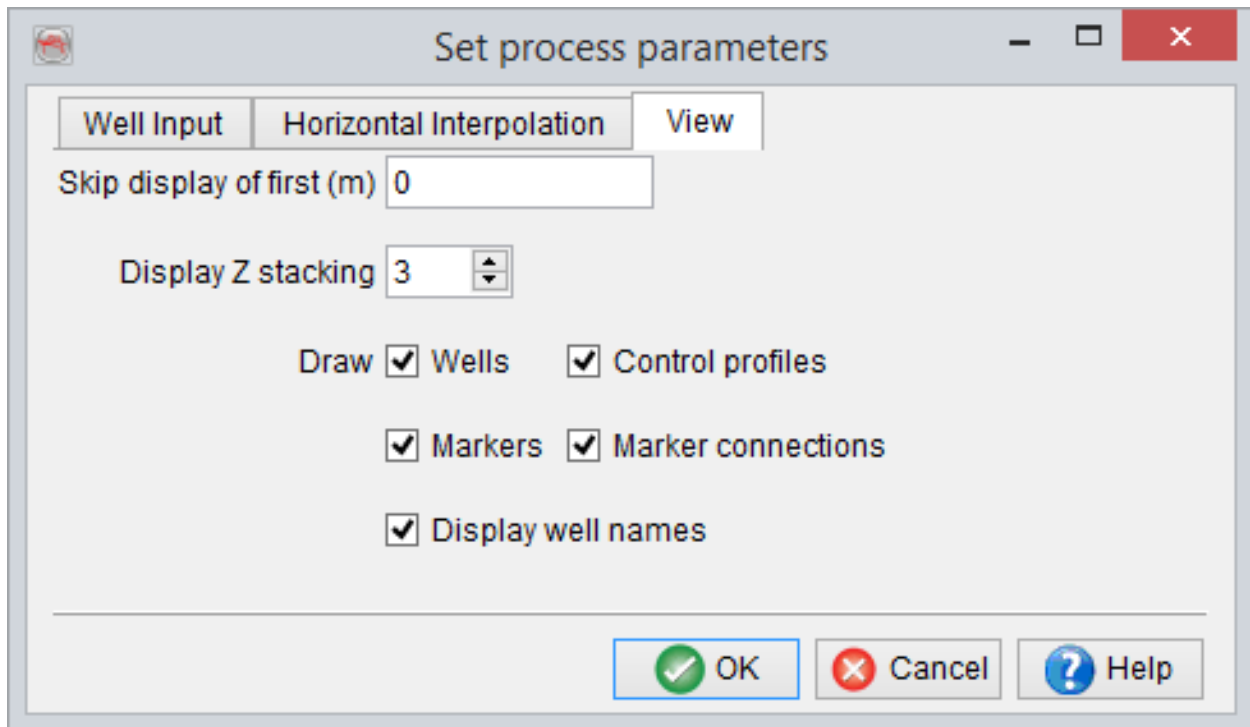
The second tab (below) parametrizes how the horizontal interpolation is done between the wells (and/or control profiles) of various log data. At a certain point in time, the program needs to generate a new profile between two 'fixed' profiles - Two wells, or a well and a control profile, or two control profiles. In general the thickness, at the new profile location, will not be the same as any of the two controlling profiles. If a full stretch/squeeze ratio is used, then all layers from both 'fixed' profiles will be present. If no stretch/squeeze ratio is used, then two things are possible - Either cut the extra layers from top (*Fill starting at Bottom*) or cut layers from the bottom (*Fill starting at Top*). However, the using full stretch/squeeze may be rather non-geological and no stretch/squeeze is somewhat non-flexible. Therefore, the default option is a combination of the two and is the use of 50% stretch/squeeze ratio, for the horizontal interpolation.

For example, consider a scenario, where there are 10 layers on the left and 5 layers on the right (both sides blocked at 1m), and the program has to draw a profile in the middle which obviously is going to have a layer thickness of 7.5 meters. Now, the use of full stretch/squeeze ratio option is going to fill this 7.5m thickness with all the layers, by squeezing them in. If no squeezing/stretching is used, either the top 7.5 layers or bottom 7.5 layers can be outputted. If the default 50% stretch/squeeze ratio option is used, either the top 8.5 layers or bottom 8.5 can be squeezed in the total thickness of 7.5 meters, at the new profile location.

It may also be noted that geologically speaking, cutting the top layers out or *Fill starting at Bottom* is a more sensible option.

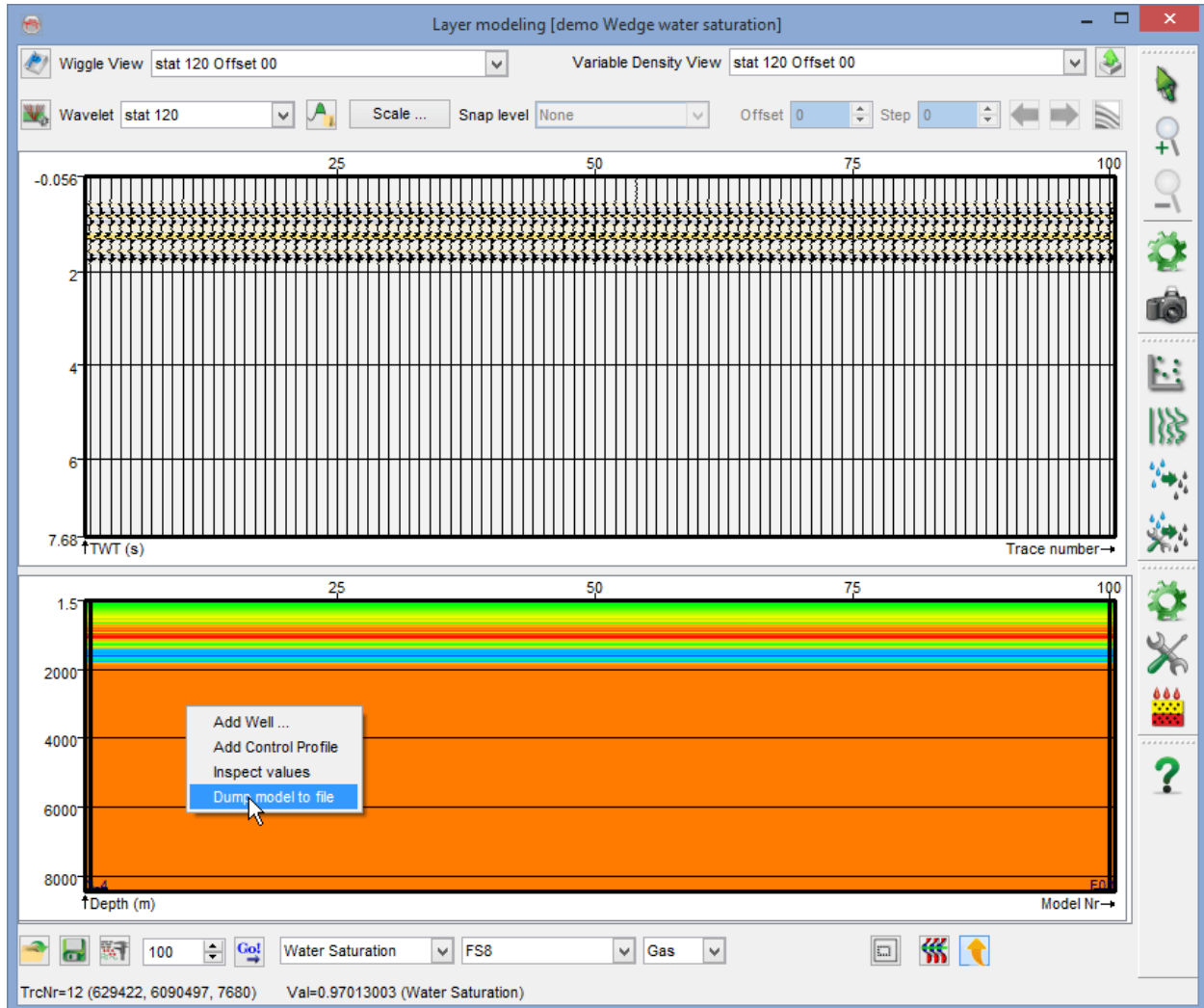


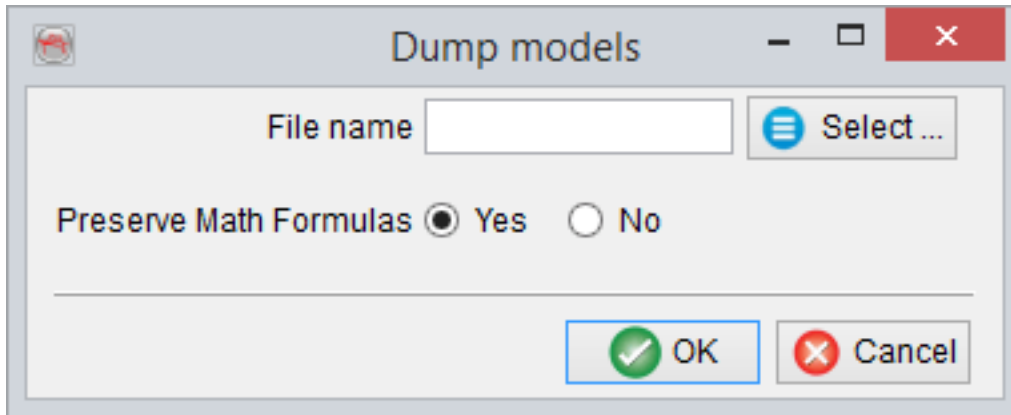
Finally, the third tab of the parameter selection window, controls the display options for the layer model. Here, it is possible to *Not* display the layer model down to a user defined depth, starting from the surface (i.e. $Z=0$). Also, it is possible to *display* the layer model in a blocky manner, by stacking a number of vertical blocks (block size defined in the first tab), using the *Display Z stacking* option. For example, if vertical blocking size of the well log data is defined as '2' meters and the default Z stacking of '3' is used, three adjacent vertical blocks (of 2 m thickness each) will be stacked together, only for display. Next, it can be chosen, whether to draw or not, the annotations for various elements of the layer model - Wells, Control Profiles, Markers, Marker connections and Well names.



8.4.8 Dump Model to File

All the pseudo-well models forming a profile simulation can be saved to a *.txt* file using this option available in the right-click menu.






If there are some layer quantities present in the pseudo-well models that are computed from other quantities using Math formulas, it is possible to save those formulas as well in the output dump file.

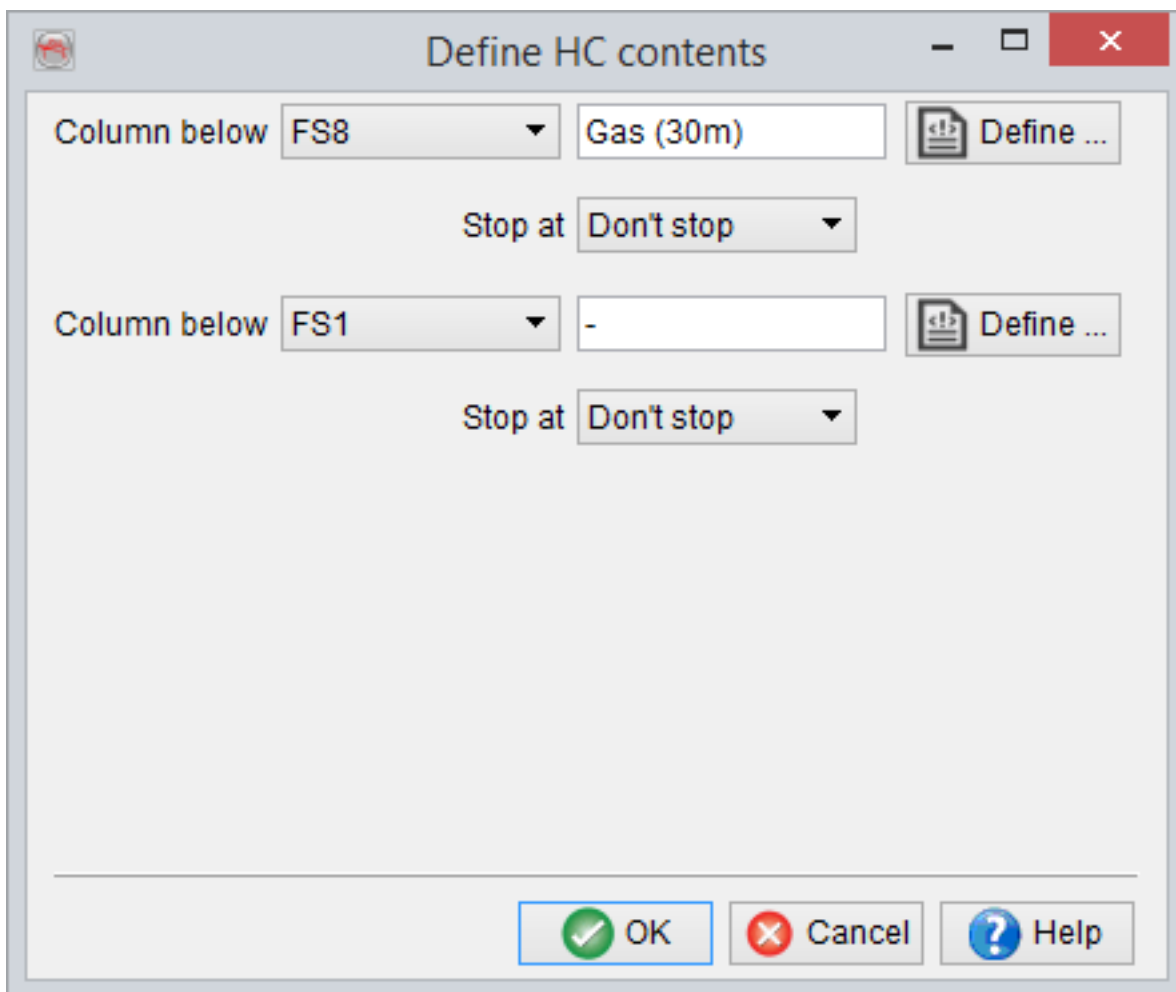
9.5 Fluid Replacement

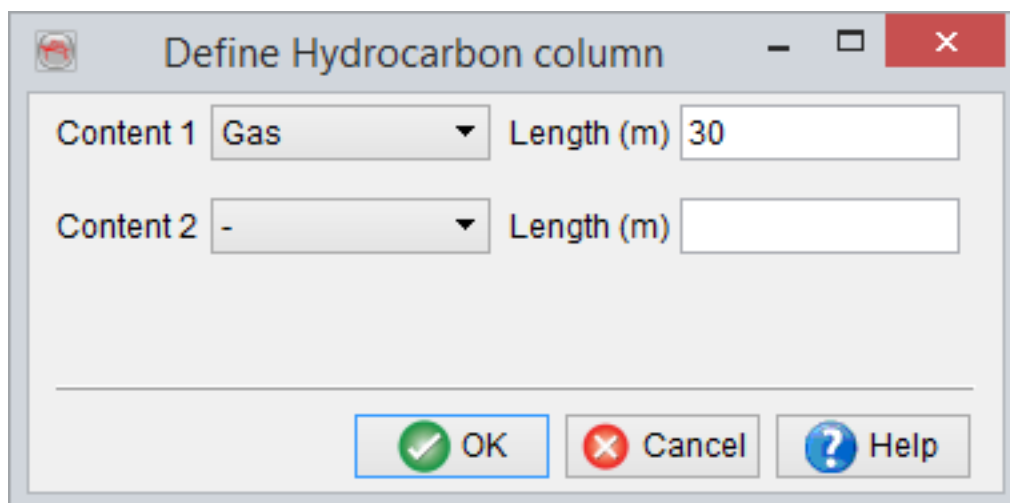
This module is used to apply fluid substitution to the modeled pseudo-wells. It computes the changes in the elastic properties (e.g. P-wave velocity, S-wave velocity and Density) of the pseudo-wells when fluid A (e.g. brine) is replaced by fluid B (e.g. oil).

8.5.1 Define Fluid Contents


The fluid replacement algorithm in SynthRock is based on Gassmann's equation. Prior to any simulation, fluid possibly present can be specified in the *Contents Manager*.

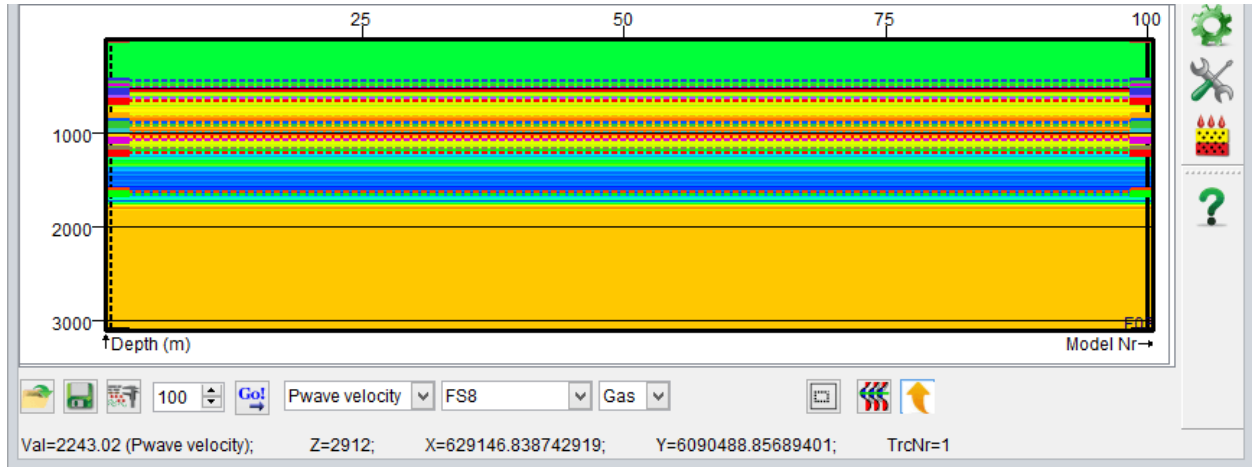
In the *Profile* module, the hydrocarbon column has to be defined by particular thickness relative to a well marker(s). This is done by clicking on the bottom right  icon, in the layer modeling window of the *Profile* module. Then a number of fluid contents, relative to various markers, can be defined in clicking on *Define*: below any individual marker, multiple fluid columns (e.g. oil column, gas column) of (possibly) different thickness can be specified.





Optionally, the user can choose to limit the HC column down to another marker, below the original marker relative to which the HC column is defined. Pressing *Ok* will set-up the HC content.

The gas column of the above example can be visualized by zooming in at the appropriate level and specifying *Gas* as fluid content, in the bottom part of the main layer modeling window. It appears as a rectangle made of red colored lines (see below). The *Outline color* and/or the *Pattern* of the fluid visualization can be changed under Manage Contents by clicking on the  in the Manage Stratigraphy window.




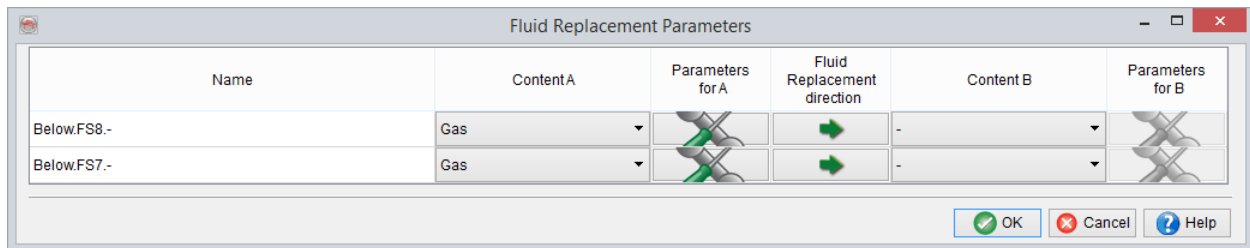
In the Stochastic or Basic Layer modeling, the fluid column has to be defined in the [model building](#) itself. The display is done in the same way as in the Profile mode.



8.5.2 Fluid Replacement Parameters

Once a fluid column has been defined in the pseudo-wells, the fluid replacement can be carried out by specifying the parameters for the Gassmann's equation.

Define Fluid Replacement Parameters

Click on  from the vertical list of icons on the right side pane, in the layer modeling window, to specify the fluid replacement parameters for various layers with fluid content.



Here, the user can specify, for various layers, the fluid Content A (initial) and the fluid Content B (after replacement). The parameters for individual fluids before/after replacement are specified by clicking on the  icon. Now, in the above example, the initial fluid content in the two intervals (below the two well markers) is brine and is denoted by a '-' sign, by default. As the initial fluid content A is brine, the parameter specification icon has grayed out. The final fluid content, B, is gas and therefore clicking on the above mentioned icon, pops-up the following window. It should be noted that the direction of fluid replacement is from brine to gas, the direction can be changed by clicking on the  : this will also change the direction of the arrow.

Fluid Replacement Parameters per unit

Fluid Replacement Parameters...

Bulk modulus hydrocarbon (GN/m2)

Bulk modulus water (GN/m2)

Bulk modulus solid (GN/m2)

Density Hydrocarbons (g/cm3)

Density Water (g/cm3)

Poisson's Ratio


Porosity (%)

Water Saturation (%)

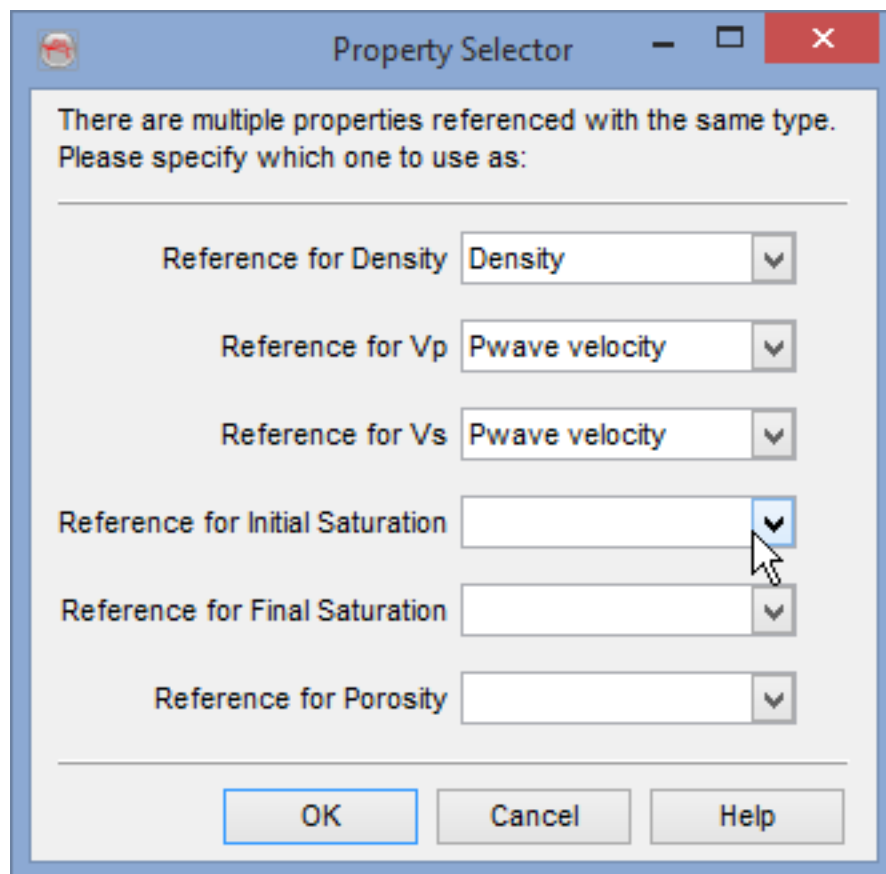
Apply these values to

The top five parameters have to be filled in. If the Poisson's Ratio, Porosity and Water Saturation information is available in the layer model, they can be left blank. In absence of this information, a constant values can be filled in for these three parameters. Finally, these parameters can be defined for the layer in consideration, similar layers or all the layers having fluid content defined.

Apply Fluid Replacement

Now, once the fluid replacement parameters have been defined, the  icon can be pressed to apply the fluid replacement. This will update the elastic properties (e.g. V_p , V_s , Density etc.) of the pseudo-well layers with fluid content, which will in turn impact the synthetic seismic response.

When the aforementioned icon is pressed the first time, the user has to specify various rock properties defined in the pseudo-wells, as required by the Gassmann's fluid replacement algorithm. They are V_p , V_s , Density, initial Water Saturation, final Water Saturation and Porosity.



If only fluid type is changed, while keeping a constant water saturation, e.g. oil to gas (at constant $S_w = x$) or from oil/gas (at $S_w = x$) to brine (i.e. $S_w = 1$) same water

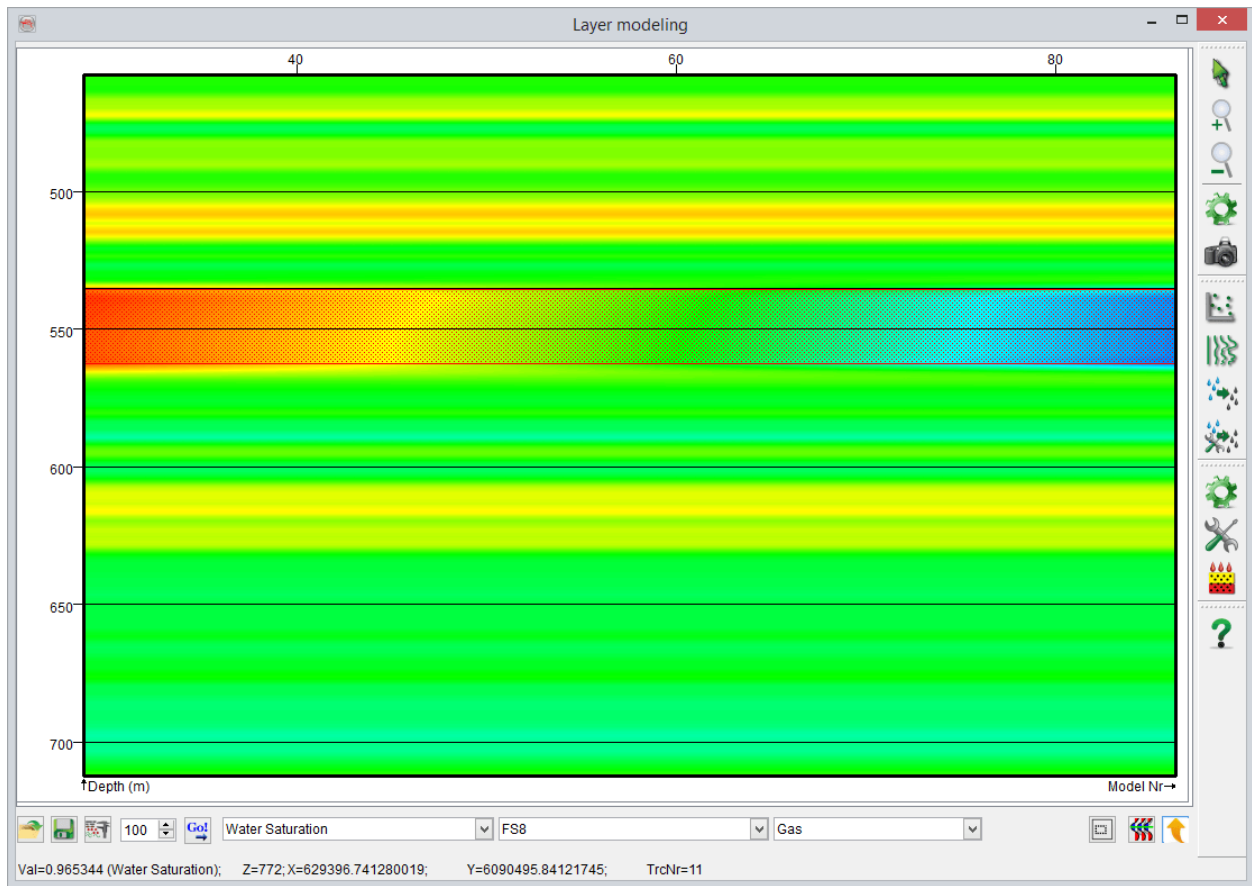
saturation property should be specified as reference for initial and final saturations. If the final saturation is kept empty, the software automatically uses the initial saturation as final saturation as well. It is also possible to specify constant values of initial and final water saturations as well as porosity, in the Fluid Replacement Parameters dialogue. If that is done, any references defined for saturations and porosity in the above window will not be honoured.

However, if the target is to compute elastic properties at different water saturations with/without changing the fluid type, e.g. oil (at $S_w = x$) to oil (at $S_w = y$) or oil (at $S_w = x$) to gas (at $S_w = y$), one needs to specify separate initial and final saturations. Note that in order to do so, two layer properties of the same type (i.e. Volumetrics → Water Saturation) need to be predefined in the [Manage Layer Properties](#) and selected in the simulation in question.

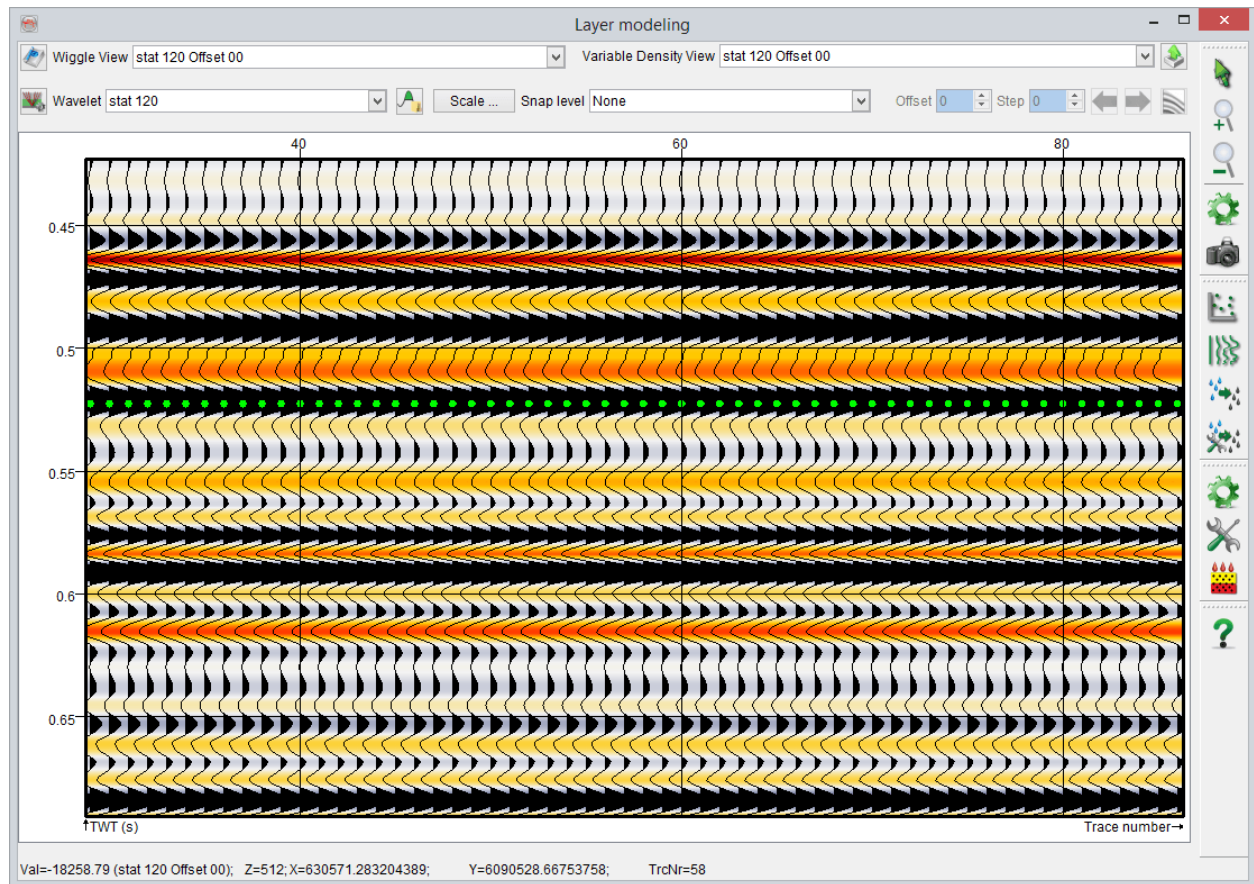


For shaly sands, use effective porosity and effective saturations, in the fluid replacement.

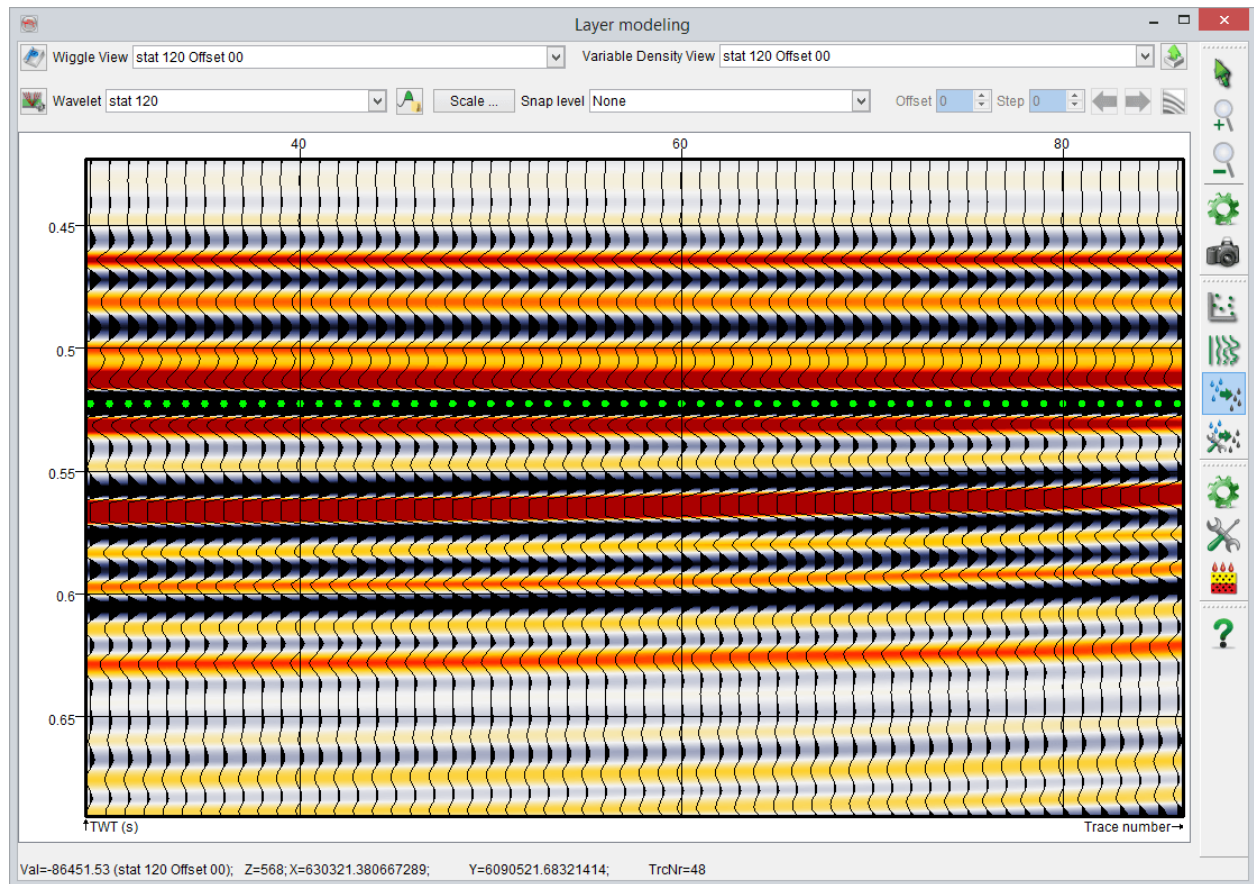
Below is an example of water saturation modeling, based on F03-4 well of F3 Demo data-set, wherein a gas column of thickness 30m is attached below the FS8 marker. Here modeling is done to decrease the logged water saturation from 100% to 90%, from left to right to replace with gas (i.e. gas saturation increases from 0% to 10%). Afterward, fluid replacement is applied to compute the change in seismic response corresponding to the change in fluid fill.



Example profile of water saturation variation from 90% to 100%, from left to right



Original synthetic seismic, corresponding to fully brine case (i.e. water saturation of 100%)



Synthetic seismic after replacing brine with gas, corresponding to varying water saturation from 90% to 100%, left to right

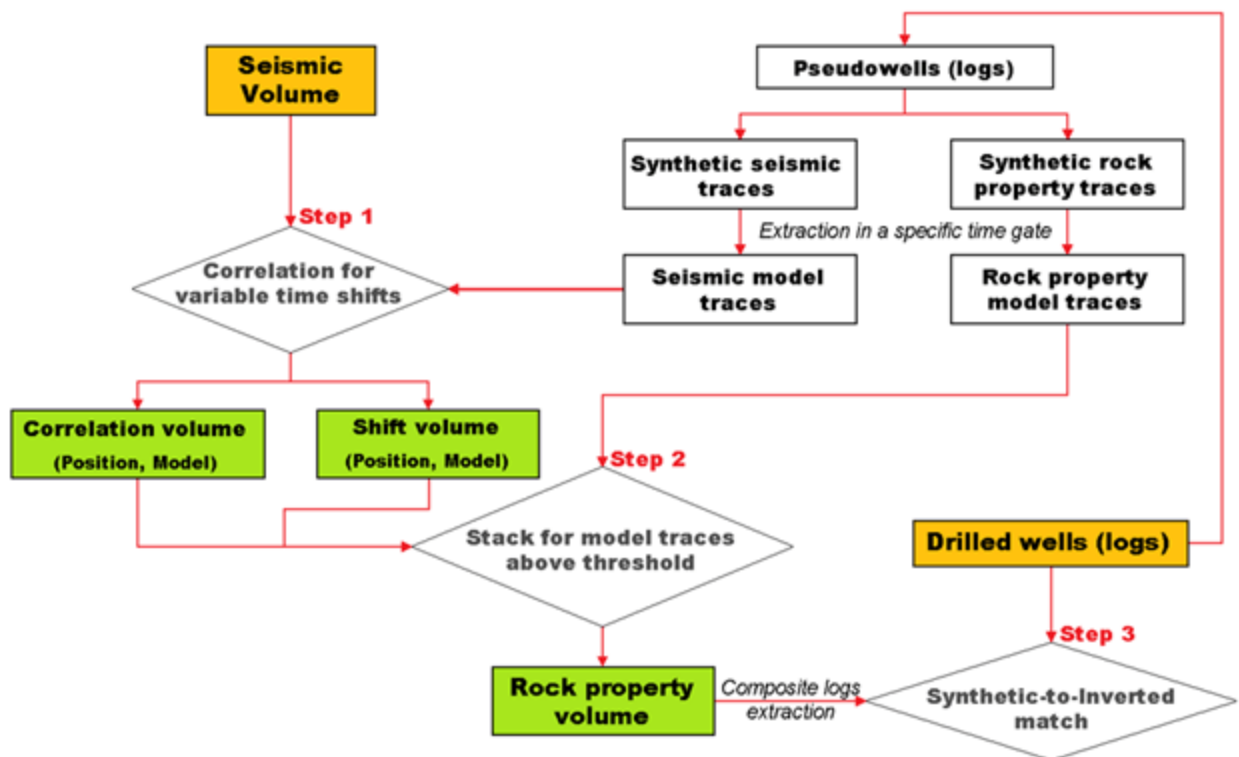
8.6 HitCube Stochastic Inversion

The HitCube is a stochastic inversion process. It assigns a spatial location to the simulated pseudo-wells such that rock properties can be computed from the tied models. The objective is indeed to predict reservoir properties with their relative uncertainties.

The HitCube workflow is divided in two steps:

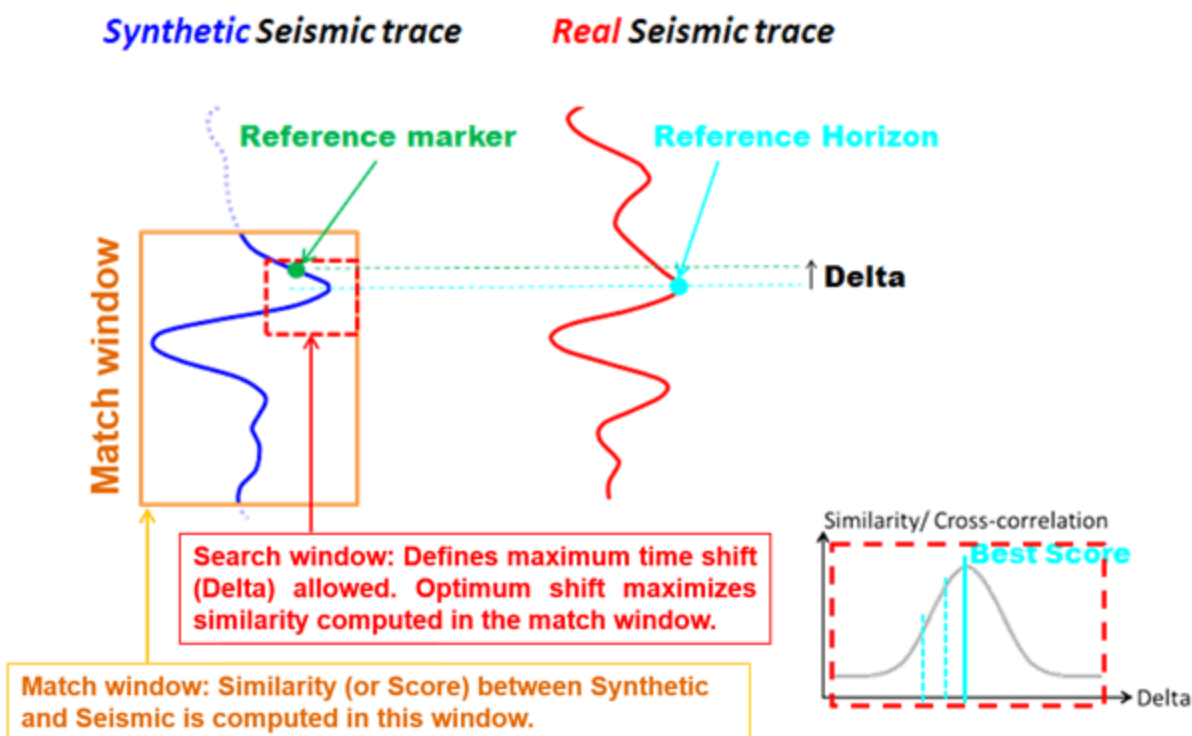
1. The actual seismic traces in the volume are correlated with synthetic seismic from modeled wells.
2. The property traces from corresponding models with a good correlation are stacked to build the output probability grids.

The inversion can be proceed regarding a selected horizon or within the entire volume. The workflow is very similar in the two cases. The inputs are the reference seismic and, for *along horizon*, a target/reference horizon.



Step 1 : Matching

The synthetic and real seismic are compared using a selected *matching method* (Similarity, Cross-correlation or Amplitude spectrum). The similarity is often preferred as it is scale sensitive and thus enables to distinguish between identical waveforms of different amplitude levels. The matching is achieved in a *matching gate* defined regarding the selected horizon if *along horizon* or if in *volume*, this time window is moved in a *Searching* time range that need to be specified. The matching window is determined in looking at the frequency spectrum of the data and QC in looking at the histogram of the output. The output of this first phase is a similarity and time shift pair for each model and each trace of the data, that is stored (internally, not visible to user) in two volumes called *ScoreCube* and *DeltaCube*. In these volumes, each sample corresponds to a model. The best matching window is the one where most of the traces correlate, i.e the less undefined values in the final outputs. It has to be selected carefully as a too large matching window will result in similarity between two different events. The matching gate can be QCed in looking at the histograms : the mean of the maximum Score (*Use best 1 models*) and the number of values used in the Delta cube at maximum Score must be optimized and the Delta cube at maximum Score must have a symmetrical histogram.



Step 2 : Stacking

The stacking step will generate the end products of the inversion, using the ScoreCube and DeltaCube as inputs, together with output features and/or logs from the pseudo-wells. For each model correspond a set of property logs that have been modeled. First the number of best models that sufficiently match the seismic data is selected. These models provide the dataset that will be stacked in order to compute the output target. The stacking can be done in taking the average or the median of the values. The final result is a predicted property volume based on probability. Additionally, various statistics (e.g. average) on score and delta values of the best matching pseudo-well models can be outputted.

Step 3: QC

The QC of HitCube inverted rock properties (e.g. AI, porosity) against available well log data can be done in OpendText using cross-plots and visual comparisons.

Discrete property prediction

Usually property logs are continuous. But it exists logs like lithology logs which are discrete. In a litholog, the log will output a number for each lithology, for example : 1= sand, 2= shale, 3= limestone. To achieve a property prediction, each discrete value has to be considered on its own. In the case of the lithology prediction, a binary log will have to be computed for each lithology (1= one lithology, 0= the others). Then the HitCube process can be run for each of them. The result would be a probability of occurrence of the lithology at every location. The facies cube is computed in combining these probability volumes : the most likely lithology is output at each location. A confidence cube can also be created as the difference between the probability of the most likely lithology and the second most likely. The more this difference is important, the more trustable is the facies distinction at this location.

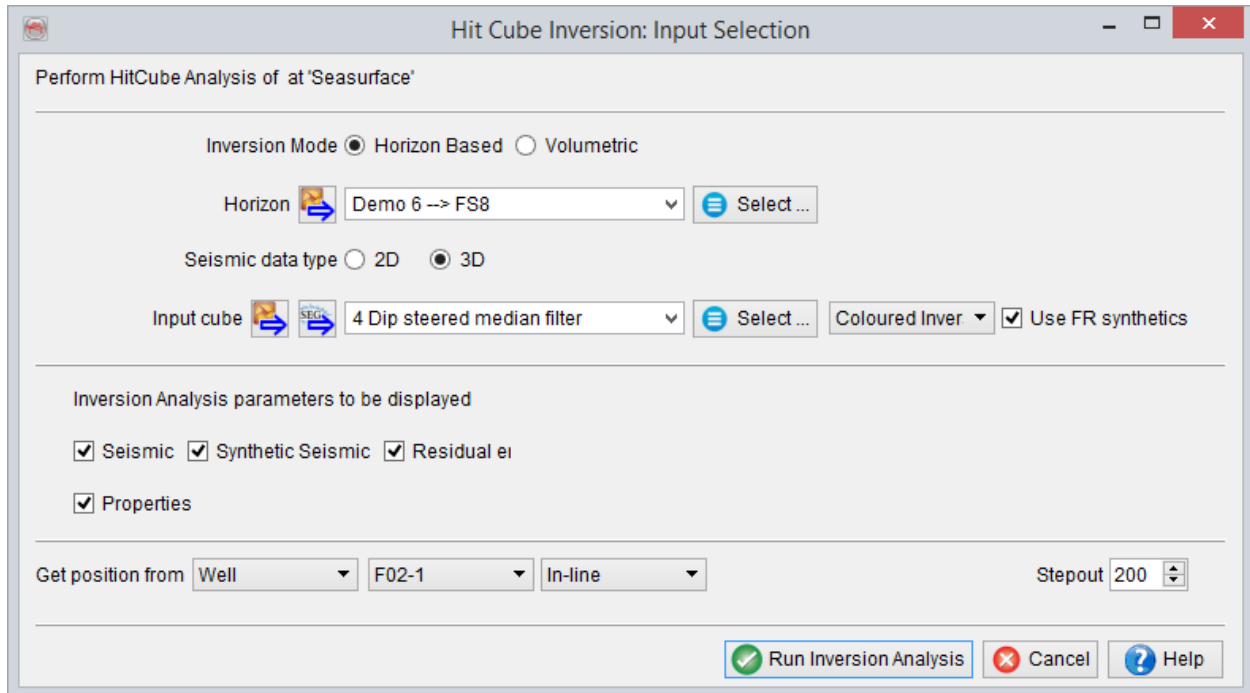
Discrete logs can be computed using any continuous log. For instance, using the Acoustic Impedance log, a new log can be computed that is equal to 1 when the AI value is higher than a given value and 0 otherwise. The HitCube will then give the probability that, at one location, AI is higher than this given value. The prediction of any log with more than two possible value follows the same workflow as the facies prediction.

8.6.1 Input Data and Inversion Type

The input window firstly allows you to choose the inversion type. The first one is the mode which proceeds the scoring regarding a selected horizon. The second one proceeds the scoring within the entire volume.

Then, before selecting any input data, you have to select the type of this one if it is a Lineset (2D) or a Cube (3D). And finally, the synthetic which will be used to match the data has to be selected, if your choice is not the default one.


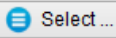
When these fields are selected click on Ok to go to the Analysis window.





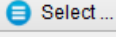
Hit Cube Inversion: Input Selection

Perform HitCube Analysis of at 'Seasurface'

Inversion Mode Horizon Based Volumetric

Horizon  Demo 6 --> FS8 

Seismic data type 2D 3D


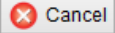
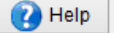
Input cube   4 Dip steered median filter  Coloured Inver Use FR synthetics

Inversion Analysis parameters to be displayed

Seismic Synthetic Seismic Residual e1

Properties

Get position from Well F02-1 In-line Stepout 200

8.6.2 HitCube Analysis

The Analysis window is maybe the most important window in the HitCube, because it allows you to set the parameters and to evaluate their QC.

The first part of this window allows you to select some parameters. Here, you can select the *Matching part of the models* by setting a time range. Notice the button on the right of this time range selection field. This button allows you to watch the matching traces defined with the previously selected time range. Then you can chose the matching method between four proposals :

- Similarity which is Amplitude and phase sensitive
- Cross-correlation which is only phase sensitive
- Similarity of amplitude spectra
- Cross-correlation of amplitude spectra

Then, you can select or not the *Search for the best match mode*. This mode gives you the access to a searching method you can apply before calculation and which will proceed the scoring at each sub-seismic step in a range around an *initial shift*. This range is defined as plus and minus the aperture value you can set. The output score will be the highest found in this range. So this best match can be shifted in time. You will be able to select as output these time shifts.

Then, you are able to chose the outputs and to set the related parameters. So you can set the time gate of interest around the horizon you want to output if the *Along horizon* mode was selected in the input window. The step will be the sampling path of your output.

Just above, you can chose the *Matching type*. This option proposes two methods.

The first is the *Fixed window* one. This method will process only one score at each trace. This score is calculated around the horizon in the range set in the Matching part of the models field.

The second is the *Sliding window* method. This one will process a score at each step of the *Output time gate around the horizon*. The value is also calculated the time range defined in the *Matching part of the models field*.

Just below the output time get field, you chose the number of best models you want to use.

And finally, you have several output selection fields for which you can evaluate the QC of their related parameters with an evaluation window. The access to this evaluation window is the button at the end of each selected line. With these fields, you can select different outputs you will store :

- Score : stores the average, the median or the minimum value of the best pseudo-wells at each trace for the fixed window method and at each sample for the sliding window method, which number is set just above ;
- Snap delta Z : stores, for each trace or sample, the average or median of Z shifts, only available with the search mode ;
- Number of hits : stores the number of pseudo-wells which score value is higher than the one selected at each trace or sample ;
- Top model number ;
- Model properties : stores the average weighted by score values, the median, the most frequent value (advised for discrete log) or the standard deviation of the selected property at each trace or sample. You can also normalize the score values by squeezing them in the range which can be set.

HitCube Inversion Analysis [Evaluation Needed]

Inversion parameters

Matching Gate: -12 | 32 (ms)

Matching method: Similarity (Amplitude and Phase sensitive)

Accept best: 5 models

Accept only scores above: 0.75

Advanced ...

Inversion Window (ms): -8 | 48

Define properties ...

Calculate: Average

Display will be enabled after Evaluation

Color Bar for seismic/synthetic

-5200 | 5200 | Seismics

Real Seismics : 4 Dip steered median filter

Synthetic seismic : Coloured Inversion Operator_bp Offset 0

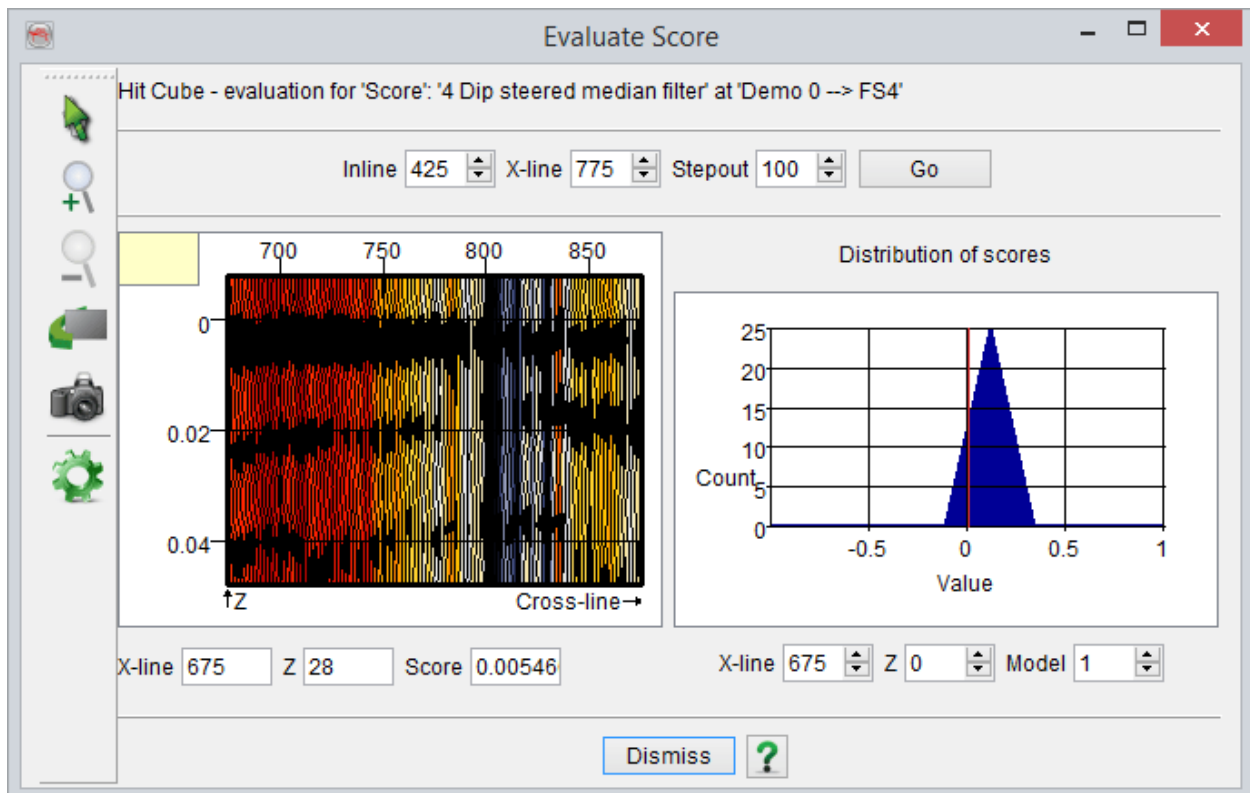
Residual Error

Score vs CrossLine Range

Get position from: Define | In-line | 425 | Central Position: 418 | Stepout: 118

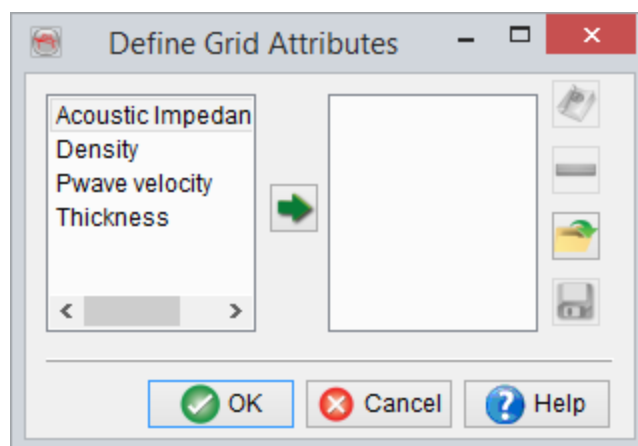
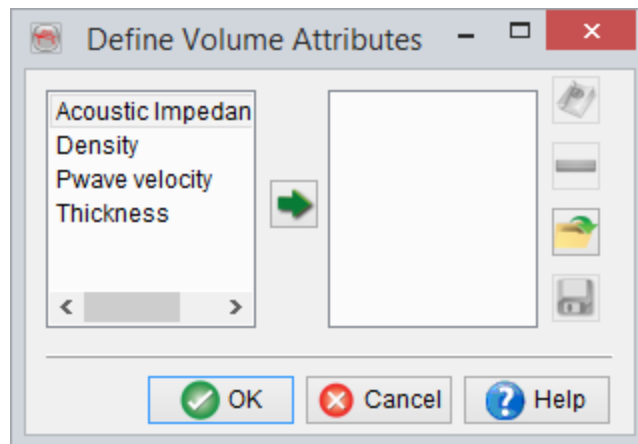
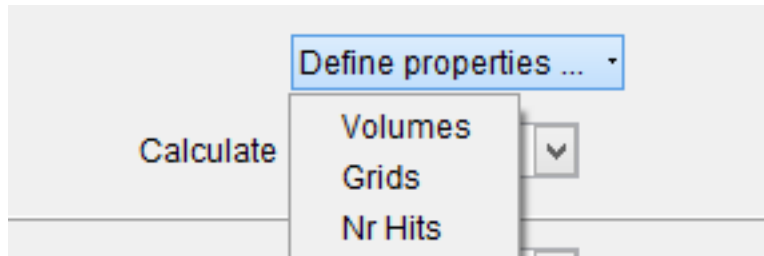
8.6.2.1 HitCube Parameters QC Evaluation

The evaluation windows will allow you to test different sets of parameters and to watch a preview regarding these sets with graphs in different inlines and crosslines. You can also change the scale of the left preview window by changing the stepout.



8.6.2.2 Define Output Layer Properties

For the last field ("Store model properties"), you need to select a property before QC it.



8.6.3 Output Data - Inversion Batch Processing

The last window, the output one, allows you to set the final parameters for the *Matching type*, *Output time gate around horizon* (if *Along horizon* mode was selected) and the *Area subselection*. Then you have the name generation options, like prefix, postfix or the wavelet name if it is available.


Hit Cube Inversion: Output selection

Quantities to output

Defined Properties Score

Synthetic Seismics Top Model Number

Residual Error Delta Z

Inversion parameters Match gate : (-12, 32) (ms) 
Accept best 5 models
Accept all scores.

Output names will be generated.
Specify what to use for name generation.

Layer generator name Prefix

Cube name Postfix

Wavelet name

Value outside calculated part Undefined

Area subselection

9 Velocity Model Building



9.1 Introduction

This plugin will work only on a 3D survey

The basic concept of *velocity model building* is to use the travel time of the acoustic waves to image the subsurface. Unlike the amplitudes, which are linked to acoustic properties (density and Lamé parameters), the travel time and ray geometry between source and receiver are a function of the velocity (and anisotropy) field(s). This is a consequence of the ray geometry obeying Fermat's principle of least time.

The velocity model building requires migrated *Common Image Gathers (CIG)*, as input. CIG are prestack seismic gathers migrated to depth, thus NMO corrected. The velocity field used for this migration will also be a necessary input for semblance calculation and thus RMO picking, and is linked to OpenTect CIG data store. The corresponding post-stack volume might also be appended to the pair CIG/Velocity model to create a link between the post stack horizons and the prestack events and to perform horizon-based velocity analysis.

Picking velocities is possible since the velocity model determines the travel times and ray geometry. A correct travel time and ray geometry will allow the image gather


(migrated seismic in depth) to be stacked constructively. The constructiveness of the gathers is measured via a semblance function which outputs a semblance value for each possible RMS velocity. The picking of the high semblances enables, therefore, the interpreter to retrieve the correct velocity function.

The process is made available on a semi-automated base in two modes:

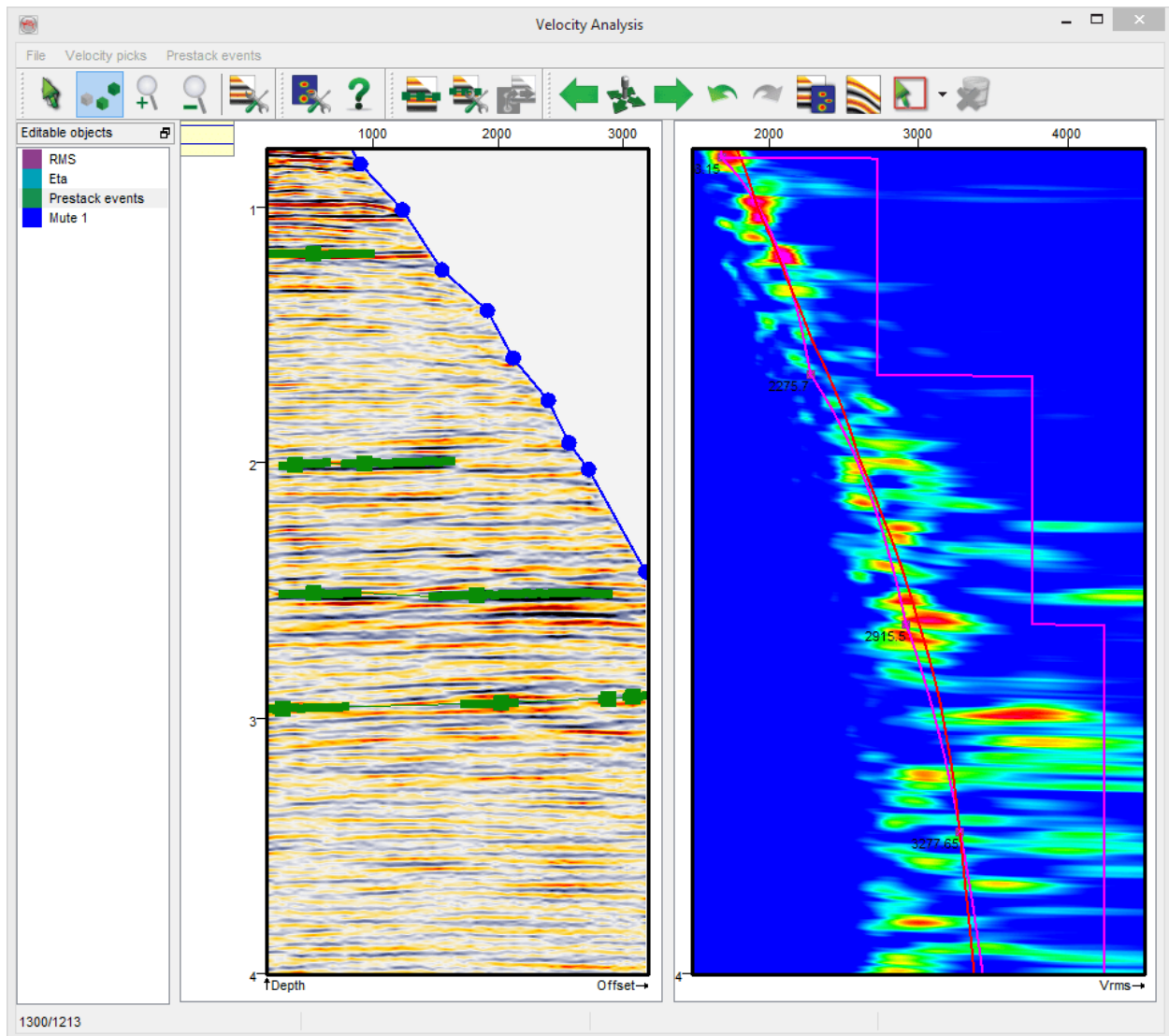
1. [Vertical velocity update](#)
2. [Horizon-based velocity update](#).

The vertical velocity update presents the semblance panel for a single trace. The picked RMO's generate a new velocity function with depth, which can be applied on-the-fly on the corresponding common image gather (CIG). Therefore, the flatness of the CIG can be very quickly appreciated during the picking phase. The maximum of semblance, therefore the change in the velocity function, is tracked between the picked positions and the next position. Therefore, the picking is called semi-automated. The picking may also be done along a poststack horizon instead of on a trace-by-trace basis. A standard workflow would be to track horizons on a migrated volume, pick the RMO's along each horizon performing the horizon-based velocity update, and QC them in the vertical velocity update window.







9.2 Vertical Velocity Analysis




From the VMB toolbar, launch the vertical velocity update window by pressing the corresponding icon .

The window is composed by a Common Image Gather from a single trace on the left, and the corresponding semblance plot on the right:




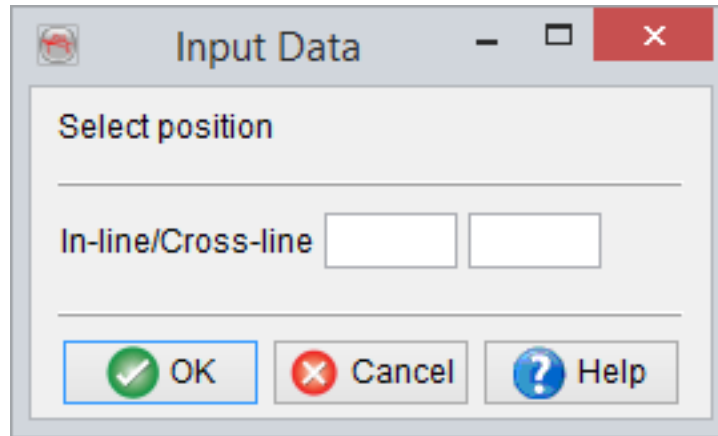
The display is annotated by separate display settings for both 2D viewers. The 2D viewer settings are similar to the standard 2D viewer, except that the semblance cannot be displayed using wiggles. Furthermore the semblance panel features one or two additional curves: The *migration velocity* (always shown) and the *picked velocity*, based on the velocity picks and migration velocity. Thus this second curve is shown once a velocity pick is made. If horizons are loaded in the main scene they will be shown as well in this vertical analysis window as horizontal coloured segments.

The velocity picks, mute definitions and tomography events are editable objects. As such they can be edited using the same tools as for the picksets and polygons: *left click to add a new, hold left click to drag and drop an existing pick, ctrl+left click to remove*. In all cases you must be in edit mode  and make sure that the object you would like to edit is highlighted/selected in the list of editable objects to apply those actions. Selection and removal tools are available to quickly remove a large number of pick on either the CIG (for mute and prestack events) or the semblance gather (for RMOs). You will first need to make a square  or polygonal  selection before to press the trash icon , that will remove the selected picks. Before removing a large number of picks you might want to use the undo  and/or redo  buttons.

Back in the view mode  you are able to zoom in  and out  on the panels, that always remain synchronized.


When moving the cursor over the semblance panel you will see an overview of the RMO correction that is linked to your cursor position.

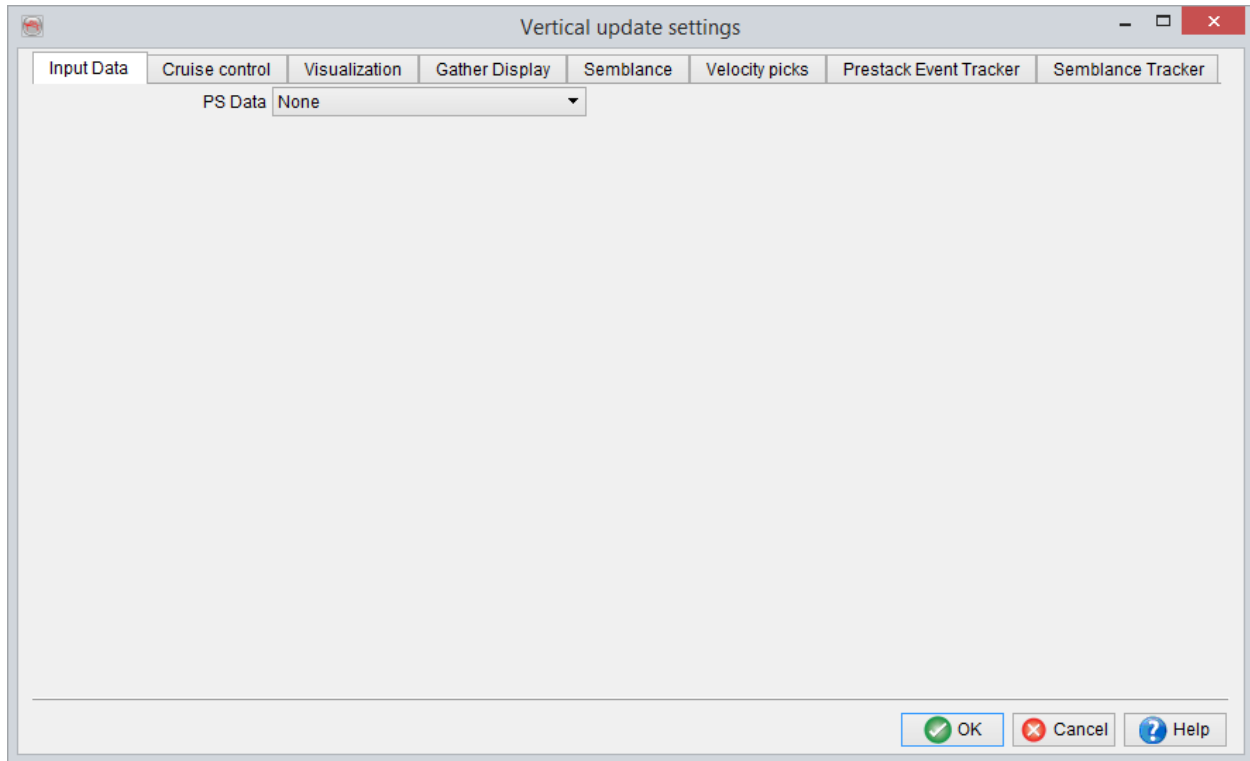
In the two viewer you can jump to any position using this toolbar icon , that launches a CDP selection window:







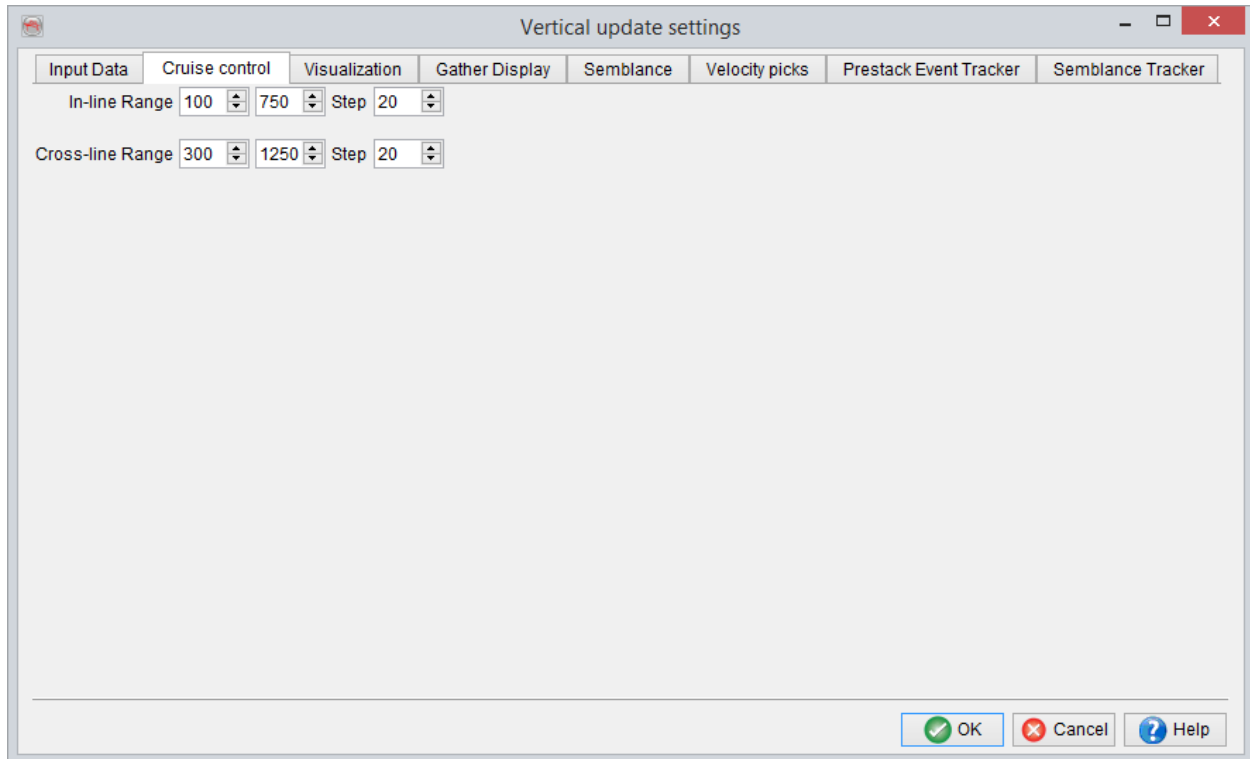
It is highly recommended to browse at start-up through the settings before picking a mute, RMO or tomography events. Those settings will be presented here below:

Settings: (available via *File--> Settings...*)

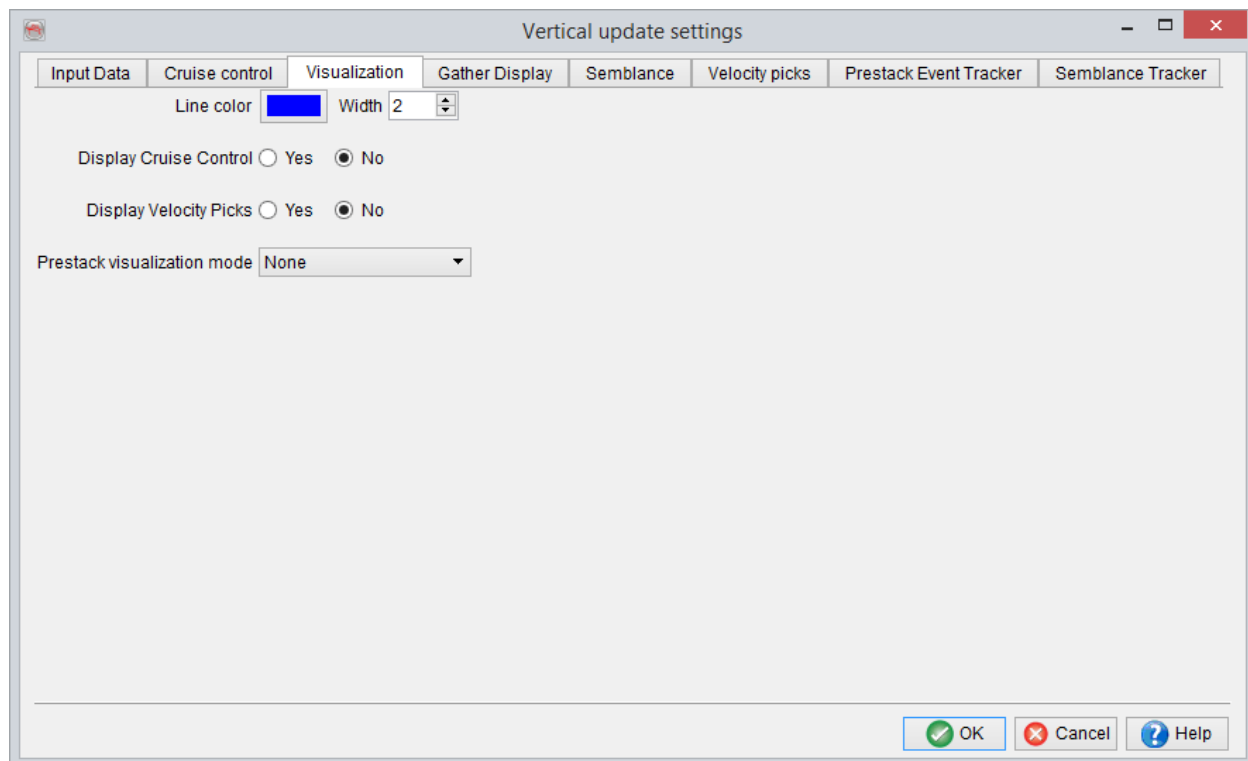
- **Input data:** You are requested to provide the prestack seismic data at the first launch, and specify the corresponding velocity model. The association will be stored and remembered for the next time. The prestack data, velocity, and post-stack seismic data can be in time or depth. As mentioned in the base documentation the velocity type must be set between Interval velocities (time or depth) and RMS velocity (time only). In the time domain the migrated gather can be NMO corrected or not. This velocity correction can be applied using the corresponding pre-processing step. The CIG display settings are directly available using the icon .





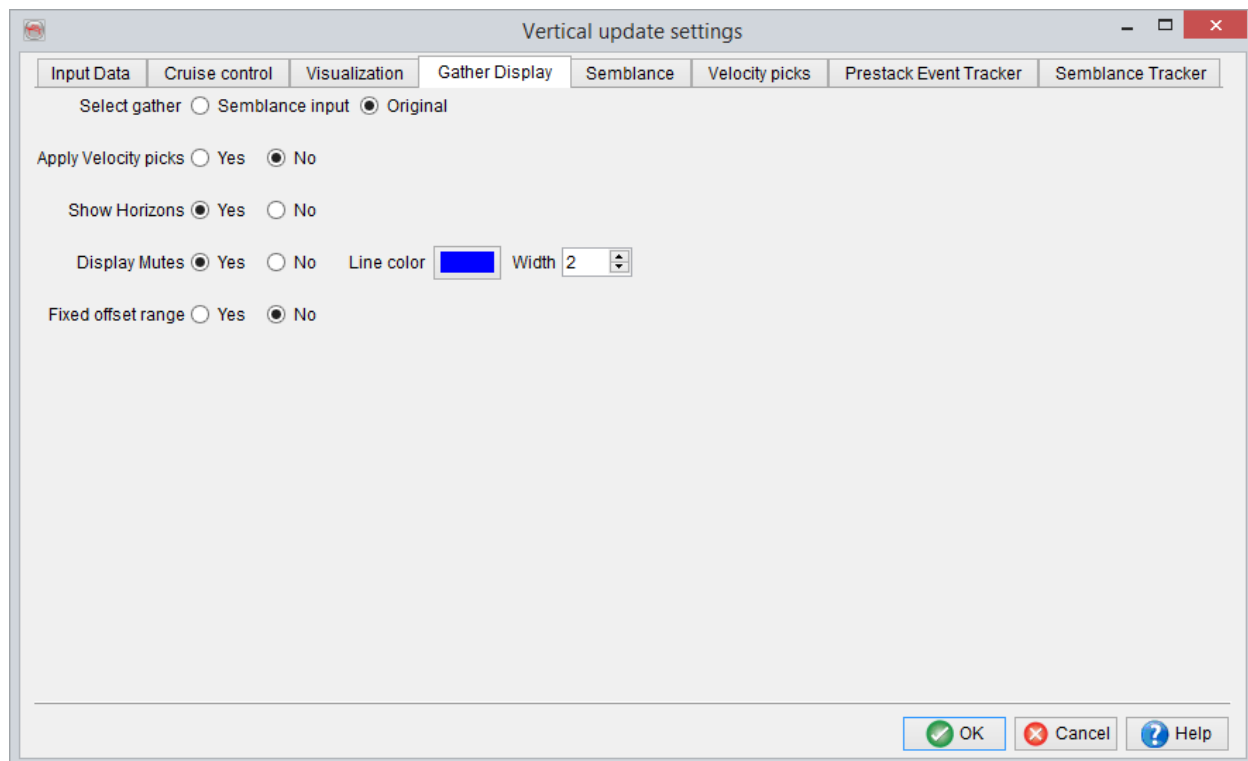
- **Cruise control:** Shortcuts are available during picking to go from one position to another. The cruise control settings define the positions that will be used for the quick browsing to the next  or previous  CIG. Nevertheless the other positions remain accessible for the analysis via the pick  icon in the main 3D scene and set position icons  . When picking your location in the main OpendTect scene you need to be in edit mode in main scene and to pick on an inline or crossline.






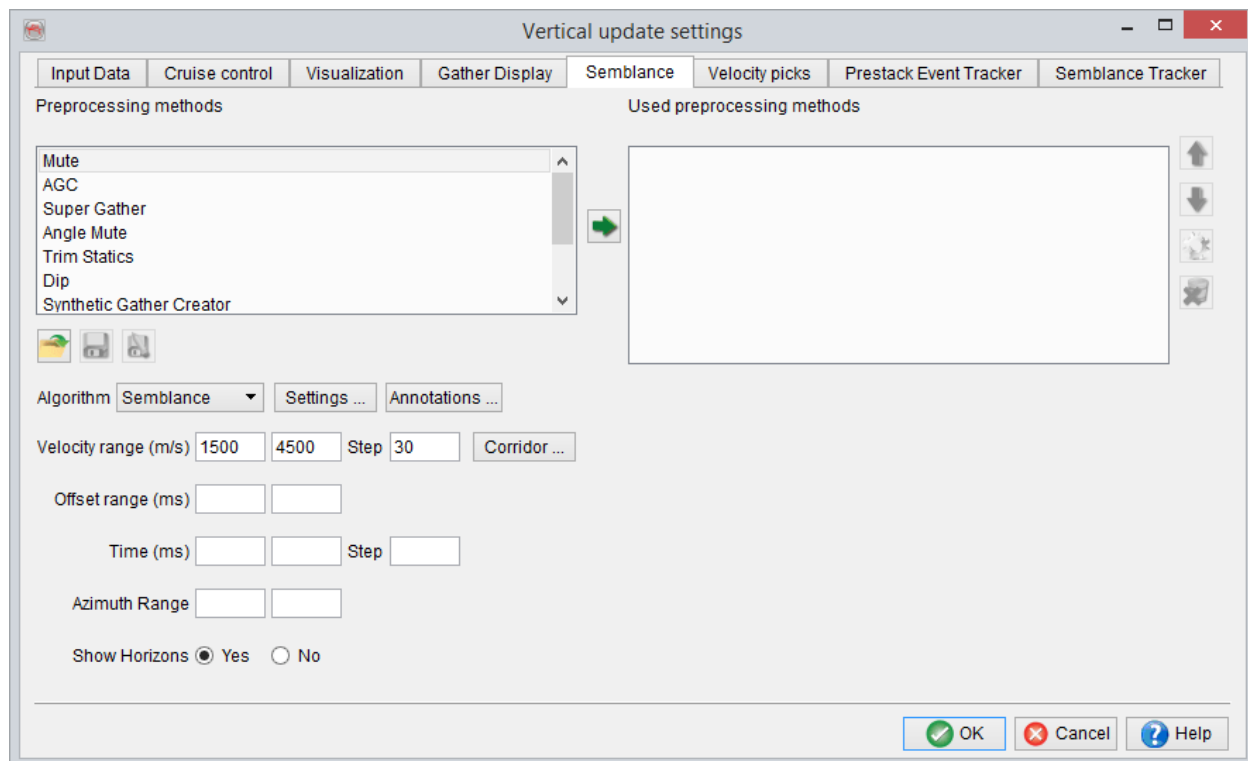
- **Visualization:** The visualization options enable you to display or hide the current analysis positions, all the positions defined from the cruise control settings, and the positions where RMO correction are picked. Those positions will be shown as vertical lines in the main OpendTect 3D scene. Optionally the cruise control positions and velocity picks may only be displayed at section, i.e. on the selected inlines and/or crosslines loaded in the scene. The current position will always remain visible. The last option allows the 3D visualization of the picked prestack events in the 3D scene



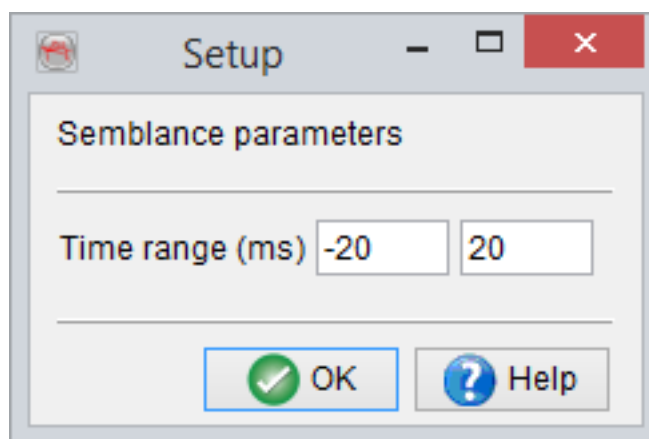
- Gather Display:** Those settings will affect the CIG display. Two toggles are available: The gather may be displayed as existing on disk (*original*) or after the semblance pre-processing (*semblance* input, see semblance settings tab). The gather may be *uncorrected*  (no application of RMO picks) or *RMO corrected*  if at least one pick is available in the survey. The mute and horizon annotations may be switched on/off. Please note that the mute display is toggled off if the RMO correction is applied, and that the horizons will only be shown in the nearest offsets.



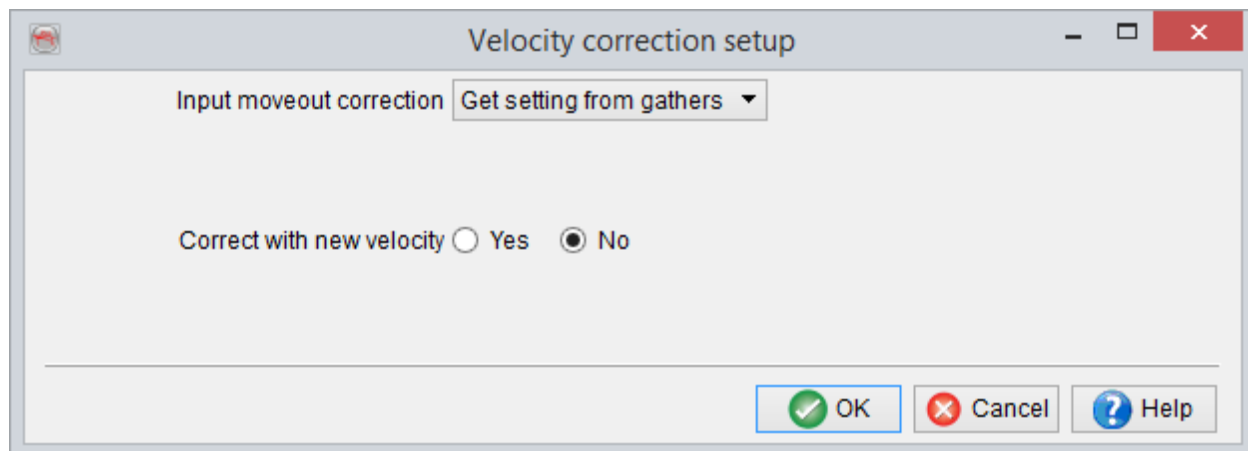
- Semblance:** This setting tab is used to set the reference offset, RMO and velocity ranges, toggle on/off the horizon display, but more important it is meant to set and list the pre-processing methods applied to the gathers on disk before the semblance calculation: Mute, Automatic Gain Control and Vertical Stack are available. Please refer to the base documentation for a description of these pre-processing methods. Empty mutes might also be created from this tab by adding a mute step without name. When one or more pre-processing methods are used you can switch the seismic display between the prestack data as existing on disk  or after pre-processing , i.e. the input to the semblance computation. The other semblance display settings are directly available using the icon .



The following settings define the length of the time/depth gate used for the computation of the semblance:

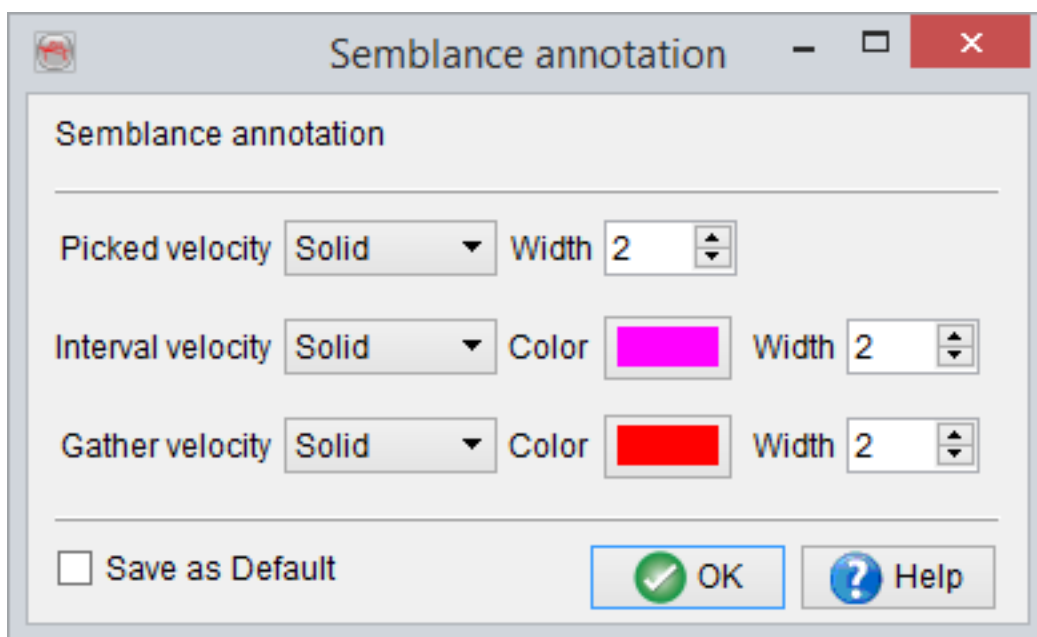


Additional VMB-specific pre-processing method are available, like Velocity correction:

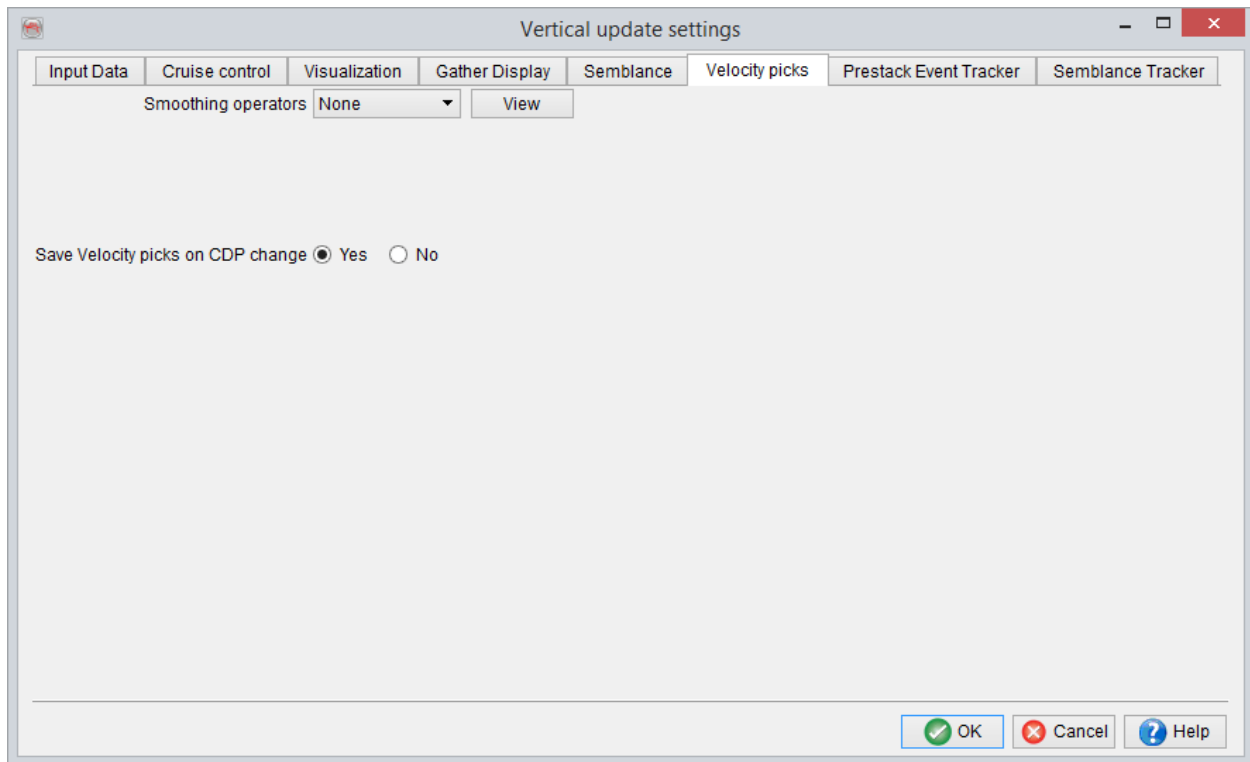


This pre-processing method will apply the velocity correction defined by the gridded velocity picks to the prestack data.

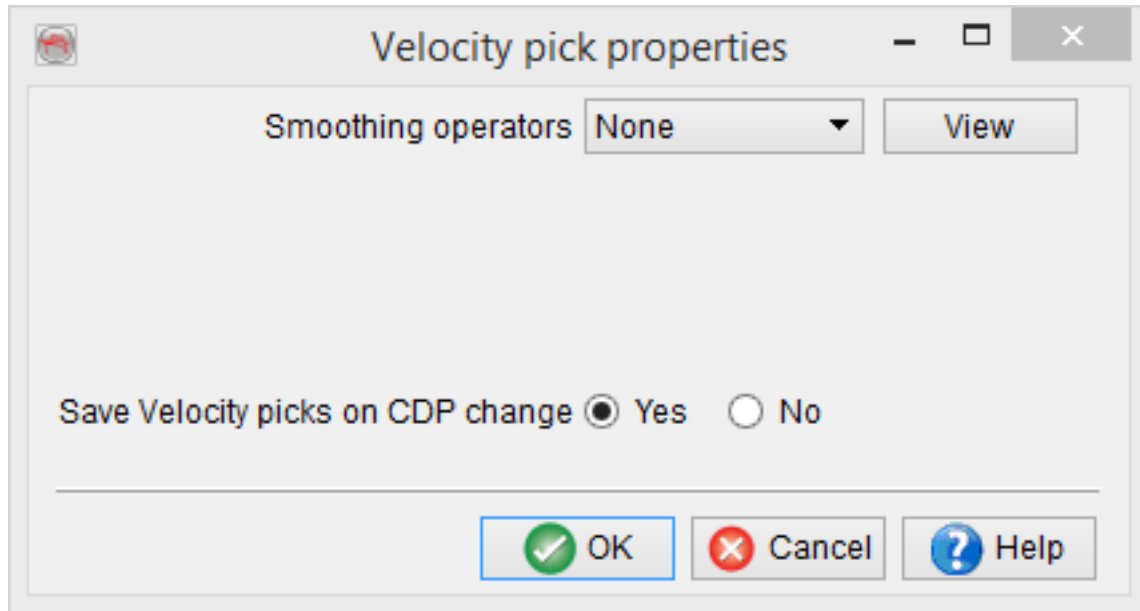
Semblance annotations Annotations for the semblance display can be set and saved as user default settings in the following window:



- **Velocity picks:** In this tab you can specify the tapering level and saving mechanism for the velocity picks. The main option is to set the tapering function and parameters used when deriving the new velocity curve from the velocity picks and migration velocity. The view option enables you to visualize the tapering functions.





Those settings can be retrieved from the *Velocity picks* menu with the option *Properties*:




- **Prestack Event Tracker:** The prestack event tracker presents similar options to the poststack horizon tracker. The tracking is based on the absolute or relative amplitude changes from trace to trace given a search window, optionally using the trace-to-trace similarity. Nevertheless given a single prestack event tracking you will be able to track both peaks and troughs on the CIG. This decision will be made based on the position of the first pick of each prestack event.

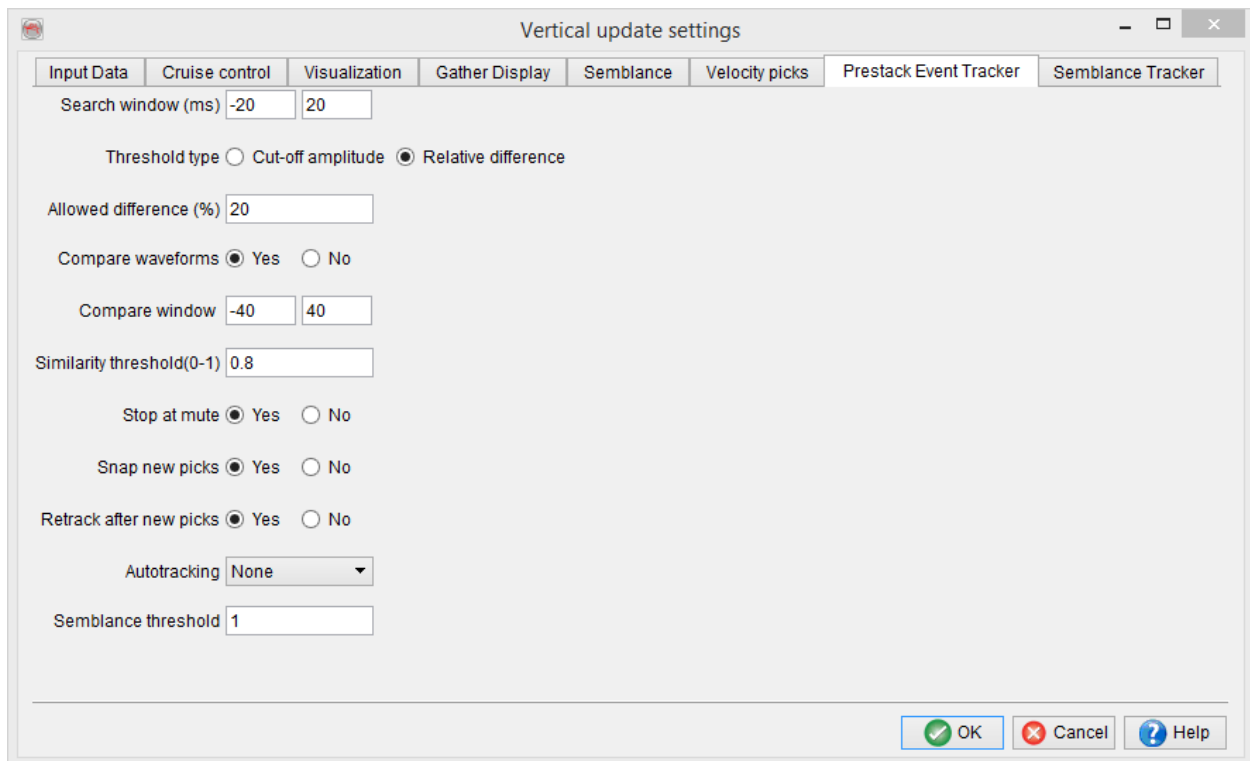
The seeds appear larger than the tracked positions, although both types are editable (movable or removable).

Futhermore there is always one active prestack event displayed using a thicker bold line, except if the cursor is outside of the search window of all existing seeds of the CIG. Any new pick will be append to the active prestack event if the cursor is within the search window of an existing pick, except if the keys control+shift are pressed when picking. In that case a new prestack event will be created.

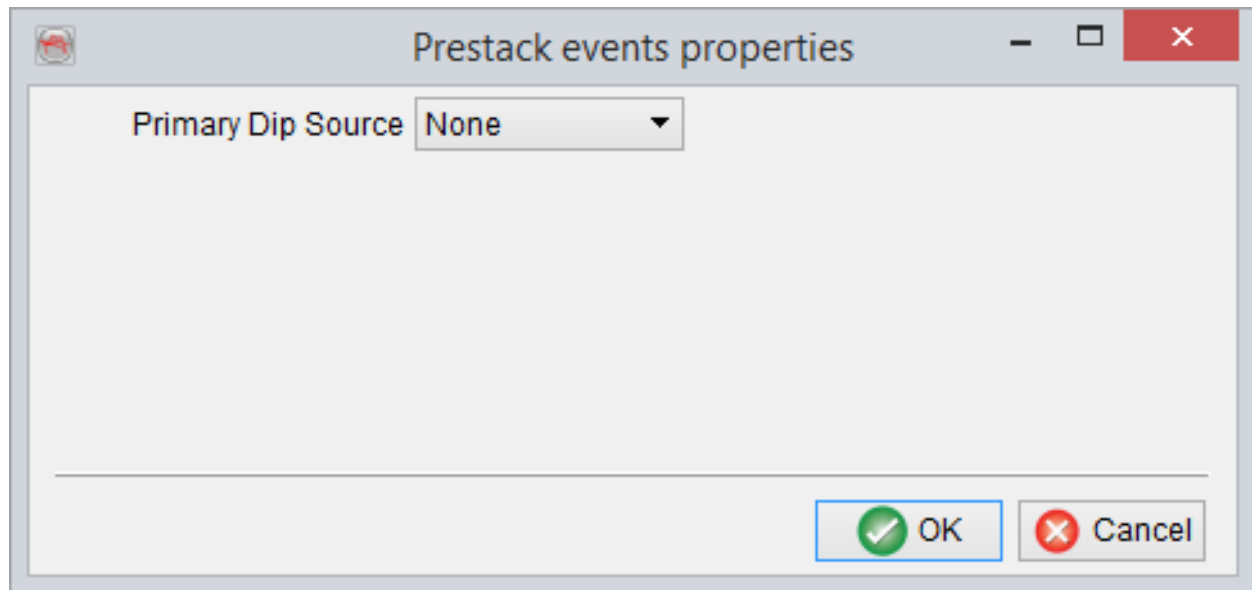
On a new prestack gather the prestack may be picked and tracked  or autotracked . The autotracking is to be used at a new locations as it will become unavailable if

one or more prestack events are picked. On the contrary at least one seed must exist before pressing the track option.

The autotracking option is meant to use either the calculated semblance or the post-stack horizons to create seeds for the autotracking of the prestack events. This option must be set in the tracking settings  before using the corresponding icons in the velocity analysis windows. All autotracked events can be edited afterwards.



Those settings can be retrieved from the *Prestack event* menu with the option *Properties*:



- **Semblance Tracker:** The Velocity picks can be tracked by following the picked RMOs between the previous location and the current position using the semblances panels at both locations, or optionally using all intermediate semblance panels. Most of the settings are once again very similar to the horizon tracker and prestack events tracker.

Vertical update settings

Input Data | Cruise control | Visualization | Gather Display | Semblance | Velocity picks | Prestack Event Tracker | Semblance Tracker


Track semblance Yes No

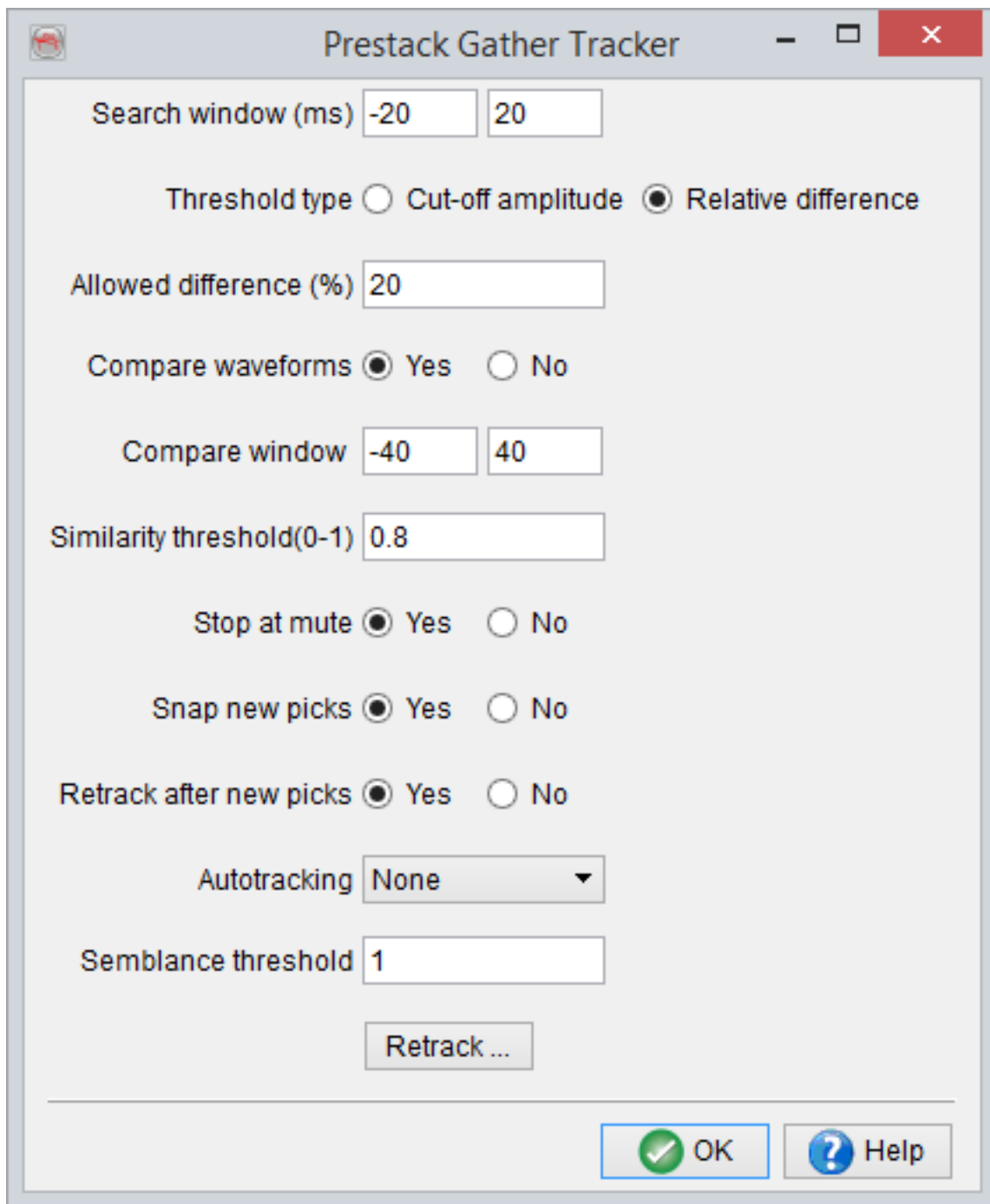
Use intermediate gathers Yes No

Velocity search range m/s

Depth search range (ms)

Compare bullseye's shape Yes No

Those settings can be retrieved from this icon  in the toolbar:




The image shows a screenshot of the 'Prestack Gather Tracker' settings dialog box. The window title is 'Prestack Gather Tracker'. The settings are as follows:

- Search window (ms): -20 | 20
- Threshold type: Cut-off amplitude Relative difference
- Allowed difference (%): 20
- Compare waveforms: Yes No
- Compare window: -40 | 40
- Similarity threshold(0-1): 0.8
- Stop at mute: Yes No
- Snap new picks: Yes No
- Retrack after new picks: Yes No
- Autotracking: None (dropdown menu)
- Semblance threshold: 1
- Retrack ... (button)

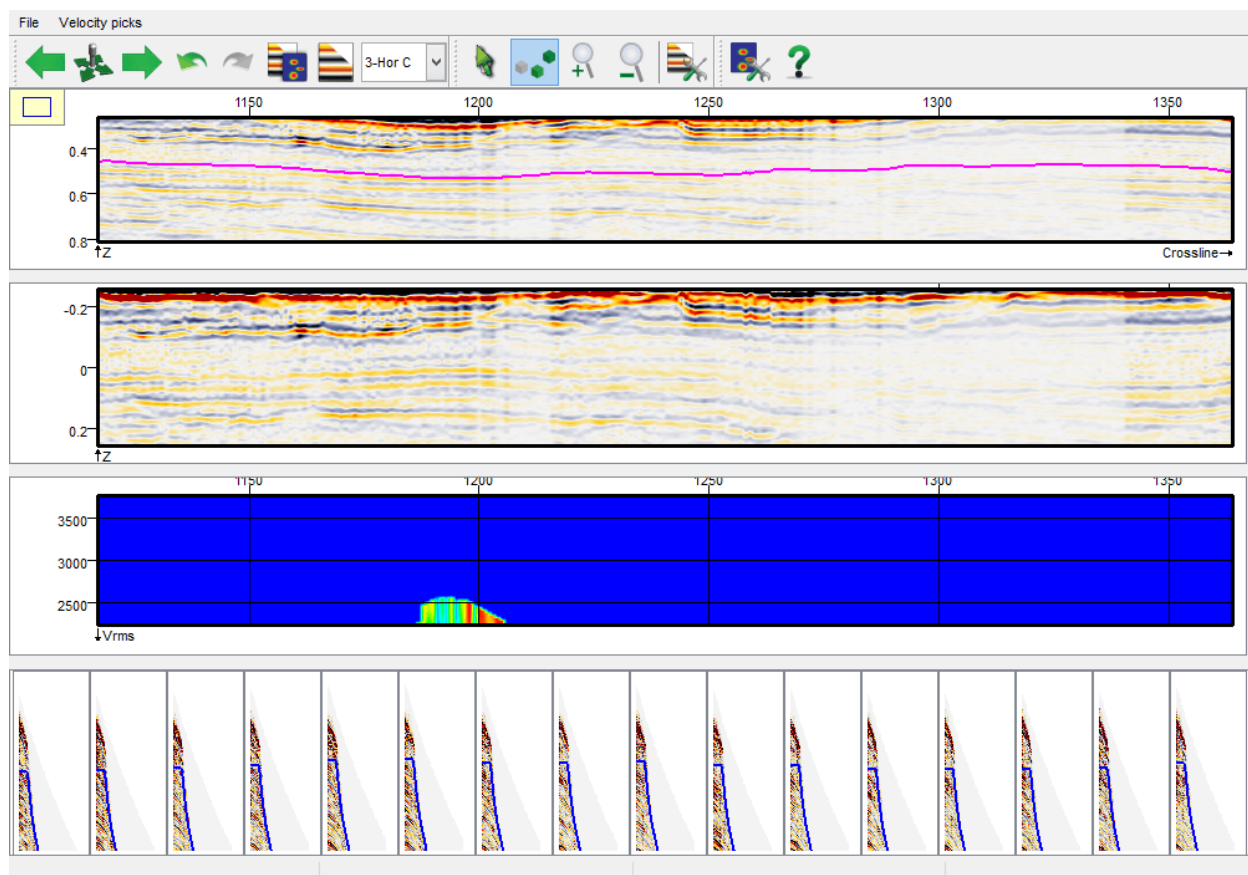
At the bottom right, there are two buttons: 'OK' (with a green checkmark icon) and 'Help' (with a blue question mark icon).

9.3 Horizon-based Velocity Update


From the VMB toolbar, launch the horizon-based velocity analysis using the appropriate icon .

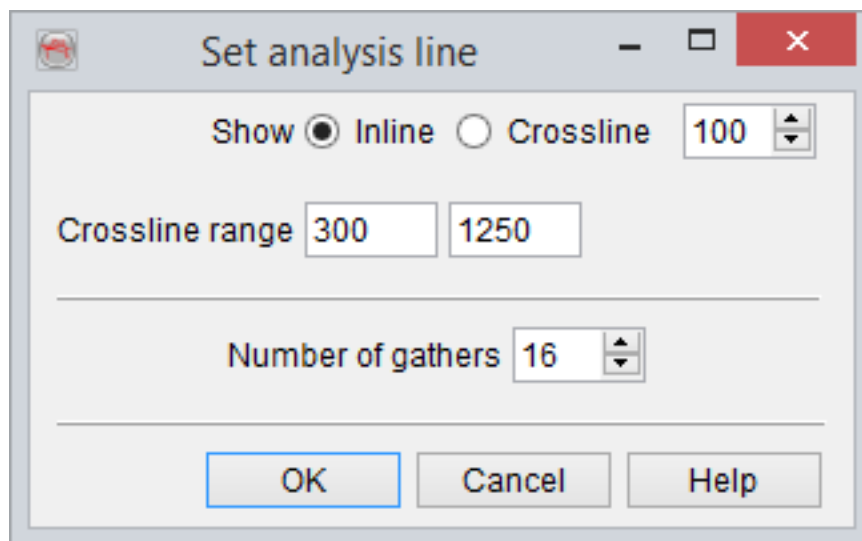
The window is setup using four 2D viewers, presented from top to bottom:

1. The migrated image on an inline or crossline
2. The migrated image on that same inline or crossline, flattened with respect to the active horizon
3. The semblance calculated at the intersection of the line and the active horizon
4. A set of CIG along the selected line, before or after semblance pre-processing, with or without RMO correction



The horizon-based analysis works similarly to the vertical velocity analysis, although only Velocity picks can be picked, and neither migration velocity nor picked velocity are displayed. The mutes and prestack events must be picked in the vertical velocity analysis window. The RMO function defined by the picks will be interpolated laterally over the entire section, providing RMO corrections at each CIG of the line. A combo box allows the quick toggle between the horizons loaded in the settings. Please note that the cursor as well as the velocity picks are synchronized between both velocity analysis windows.

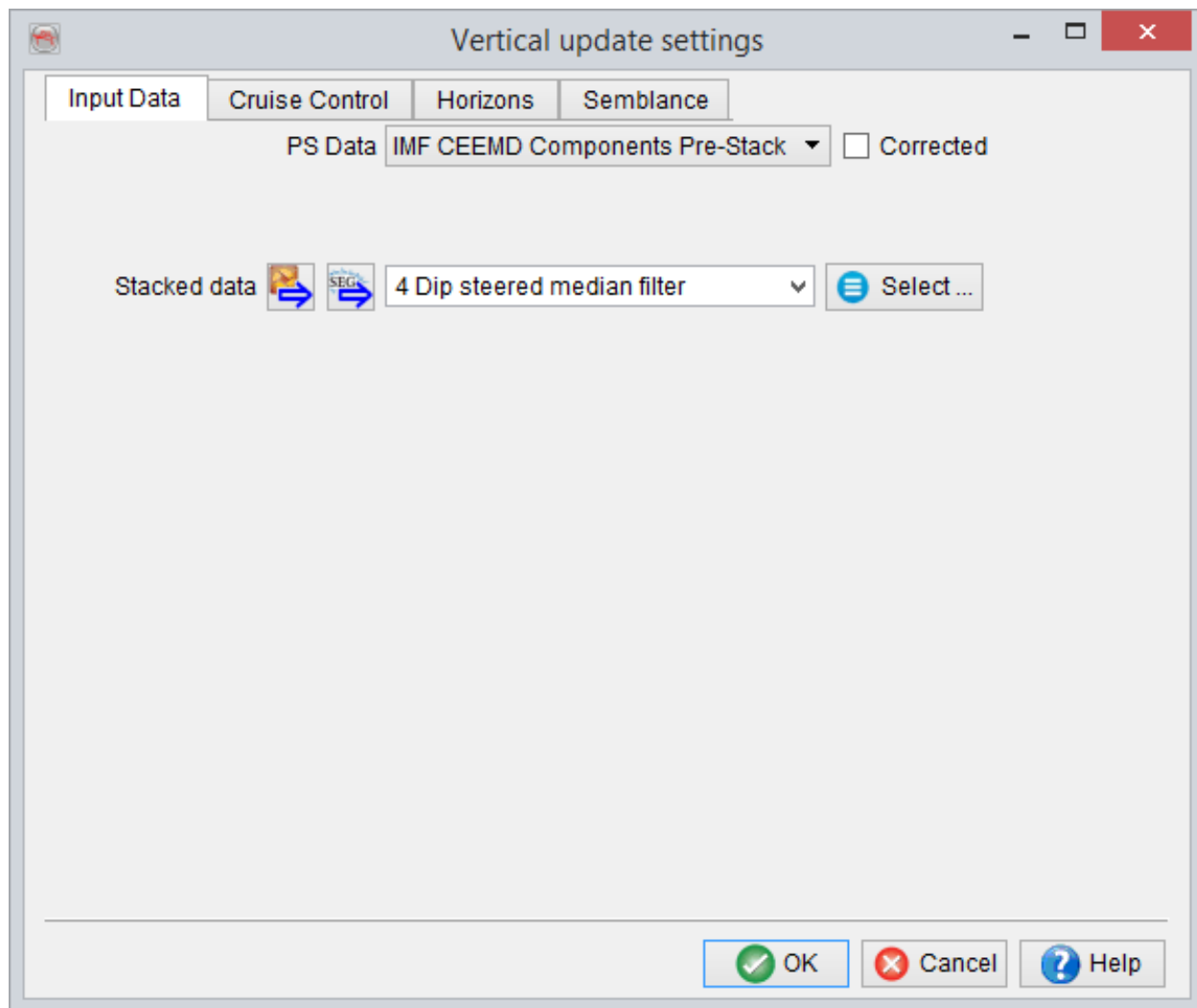
In the two viewer you can jump to any inline or crossline any position using this toolbar icon , that launches a line selection window:



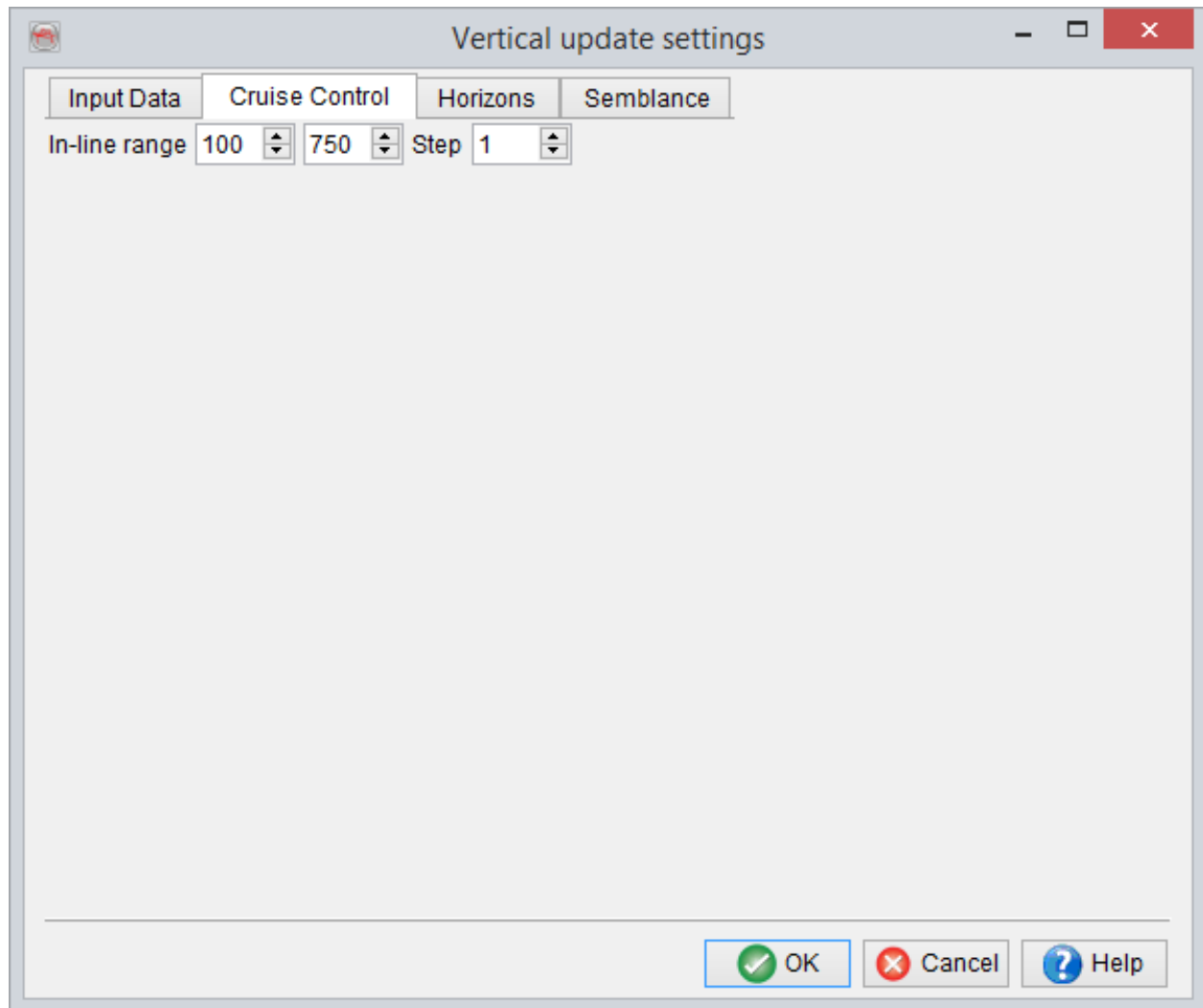
Settings: (available via *File--> Settings...*)

This section presents only the settings that are different from the vertical velocity analysis settings.

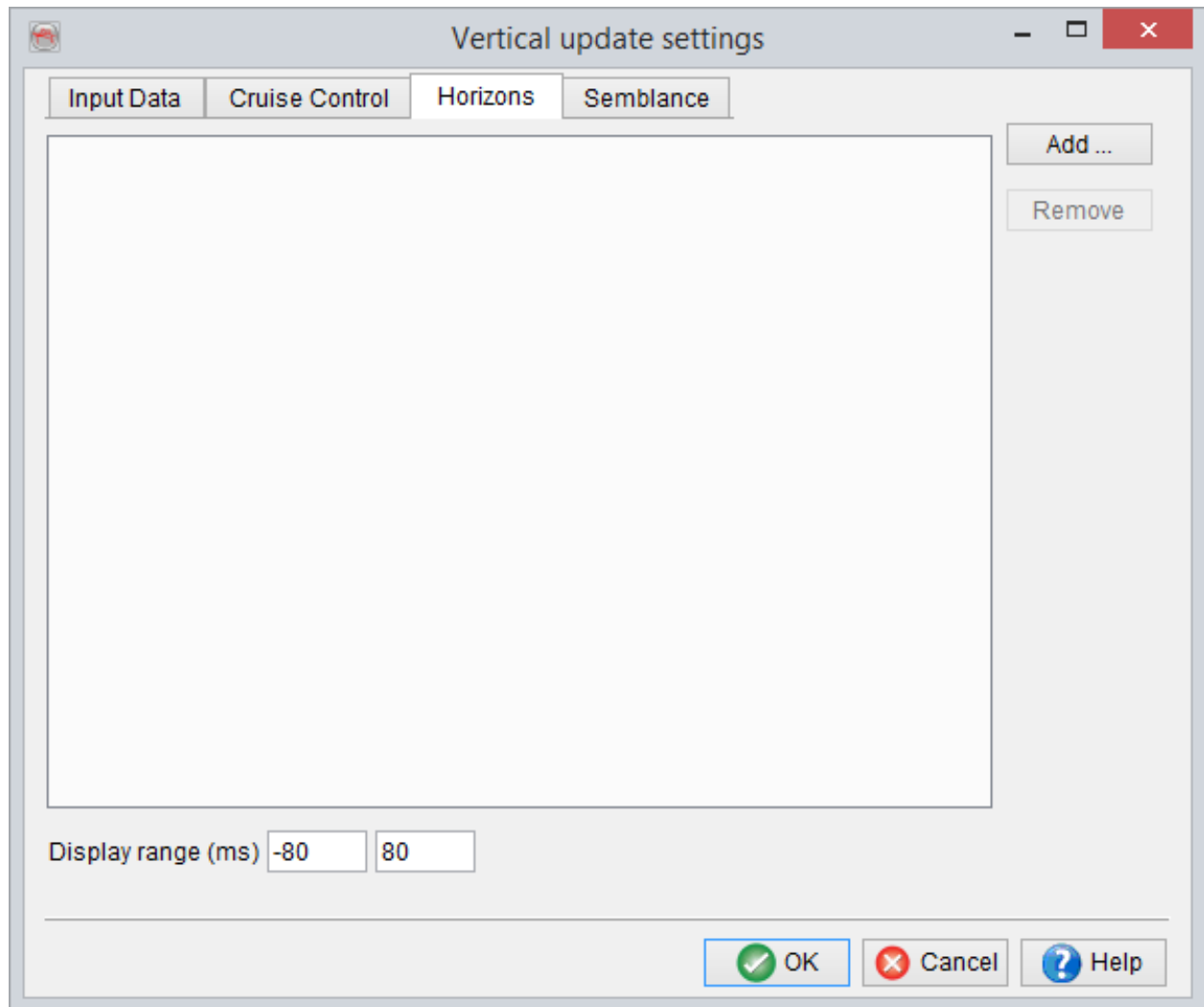
- **Input:** The input will link the prestack data and migration velocity to the corresponding stacked data visible in the two upper 2D viewers.



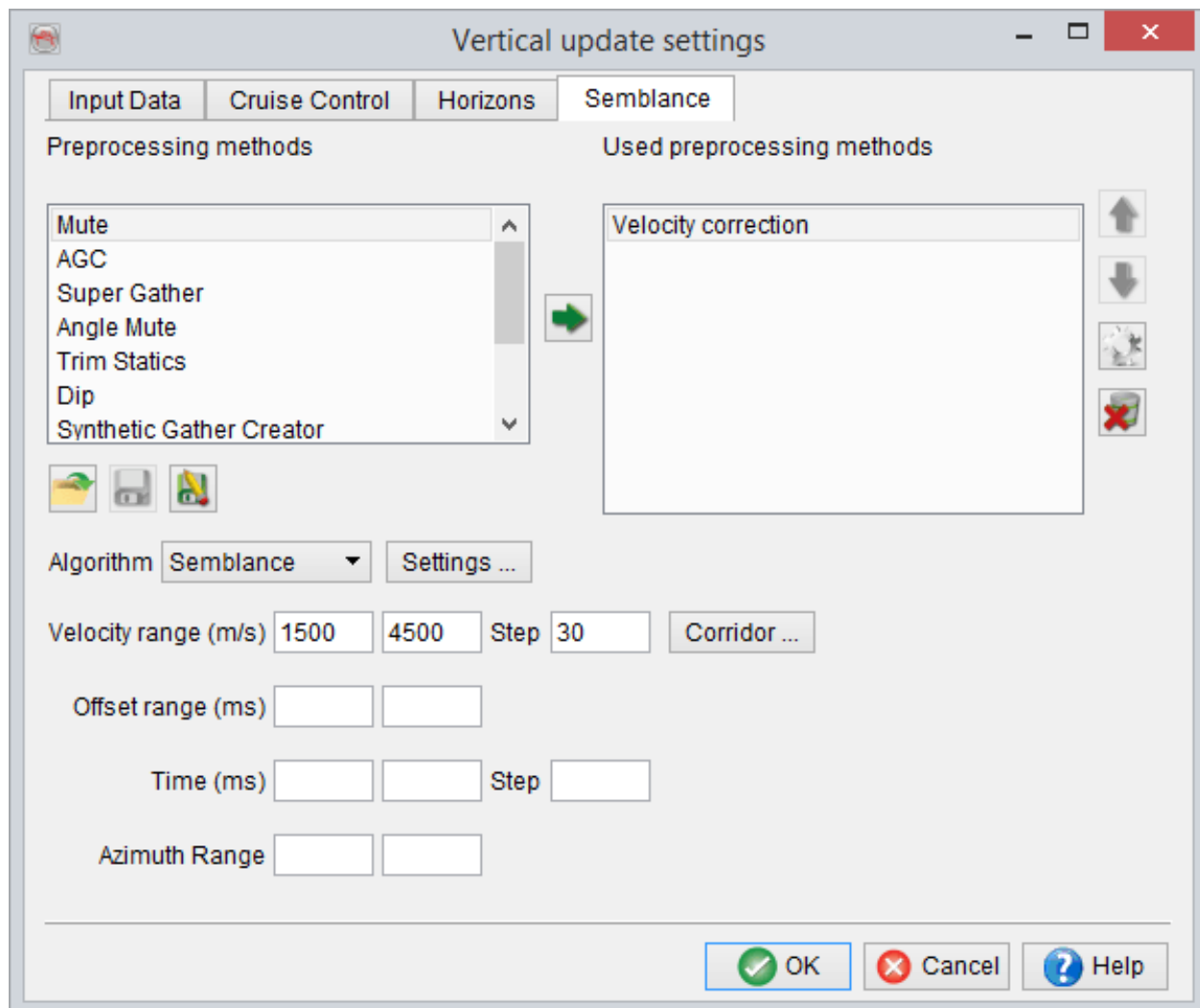
- **Cruise control:** The available item is either an inline or crossline range, depending on the choice of active line. This can be changed using the "Go to any position" button where the analysis line (number and type) can be set.



- **Horizons:** The horizons used for the analyzes must be defined here and are loaded by pressing the add button. You can select as many horizons as you want, and specify the display range of the flattened section (second 2D viewer from the top).



- **Semblance:** The semblance settings are fully similar as for the vertical velocity analysis window.



9.4 Input-Output

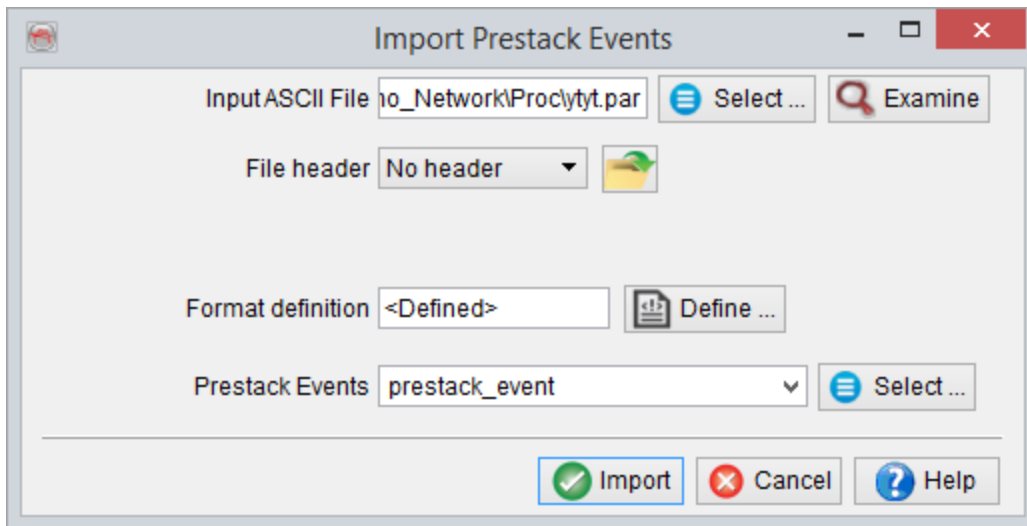
Most of the processing in the Velocity Model Building Plugin will not require data import and export:

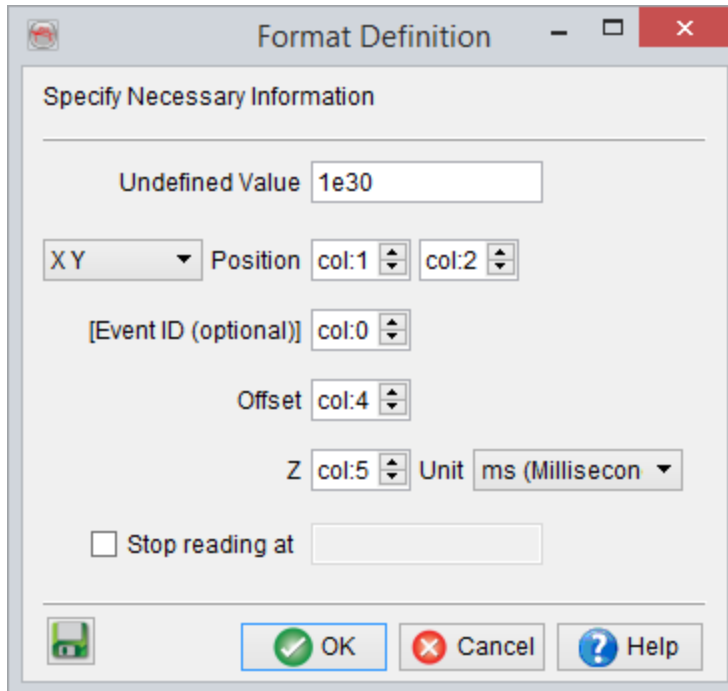
The velocity volumes are built, and remain, in OpendTect, such as the picked mute definitions, RMO and tomography events.

Nevertheless, for convenience it is possible to export the volumes to SEG-Y files and the mute definition and prestack events to text files (ASCII). Please refer to the base documentation for the export of volumes and mute definitions.

9.4.1 Prestack Event Import

The import of prestack data is similar to the other import. After selecting the proper file, in ascii format, the presence/absence of header has to be specified. It can be of fixed or variable length. If variable, the "word" defining the end of the header has to be given. then the format needs to be define : the columns for each listed quantities have to be indicated.



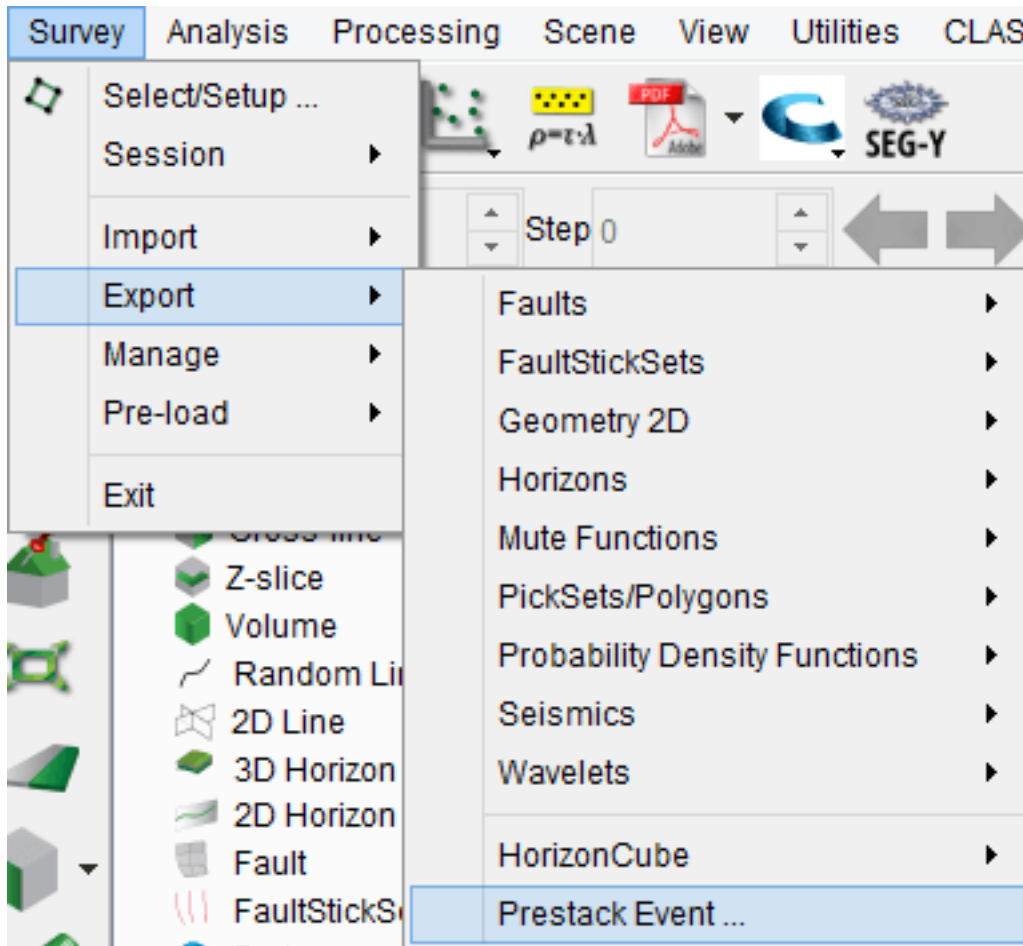


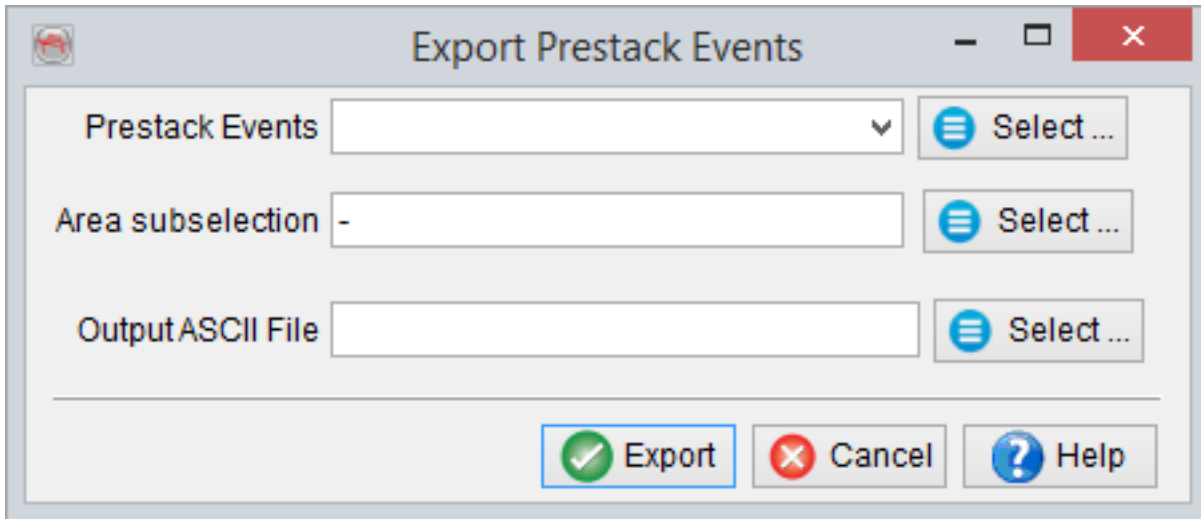
If the Event ID is not mentioned in the file, associate the *col:0* (first column in the file is *col:1*).

The data import can be done until a given row, you just need to specify it in *Stop reading at*.

9.4.2 Prestack Events Export

The export menu enables exporting of picked prestack events to text (ASCII) files. The output file will be column-sorted with one row per pick. The following data will be found in each column:





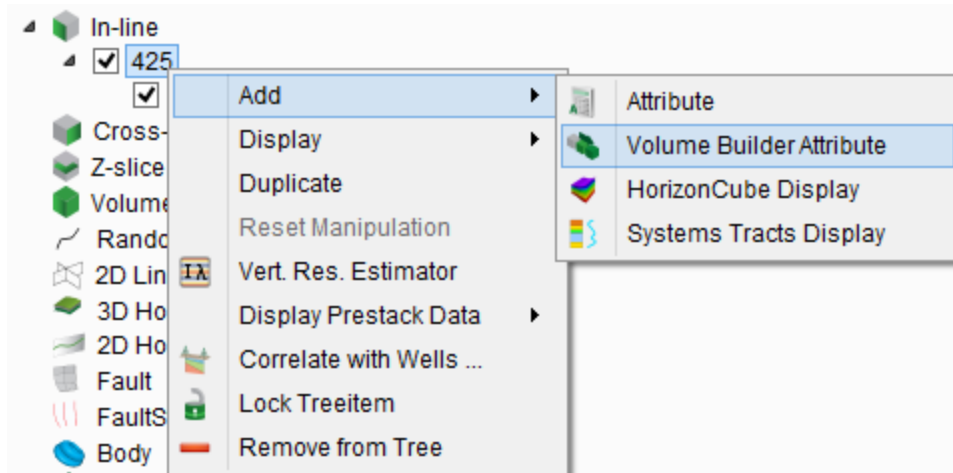
Options

- Inline
- Crossline
- Event index (0-N)
- Dip, going to increasing inlines. 0 is written if dip is not available.
- Dip, going to increasing crosslines. 0 is written if dip is not available.
- Event quality, 0-255 (higher is better)
- Azimuth (0-2PI)
- Offset
- Depth
- Pick quality, 0-255 (higher is better).

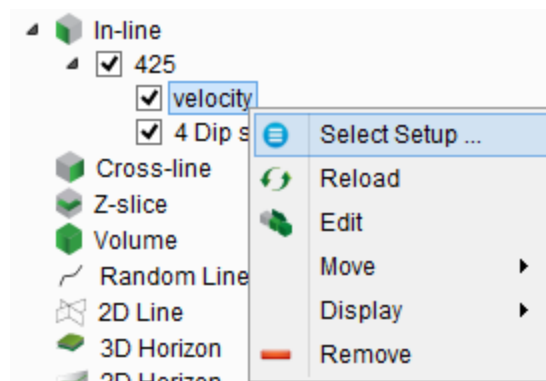
The dip comes either from a horizon (if the event is picked from an horizon) or from the SteeringCube. The pick quality is determined while tracking depending on the strengths of the event and how well the individual pick fits the rest.

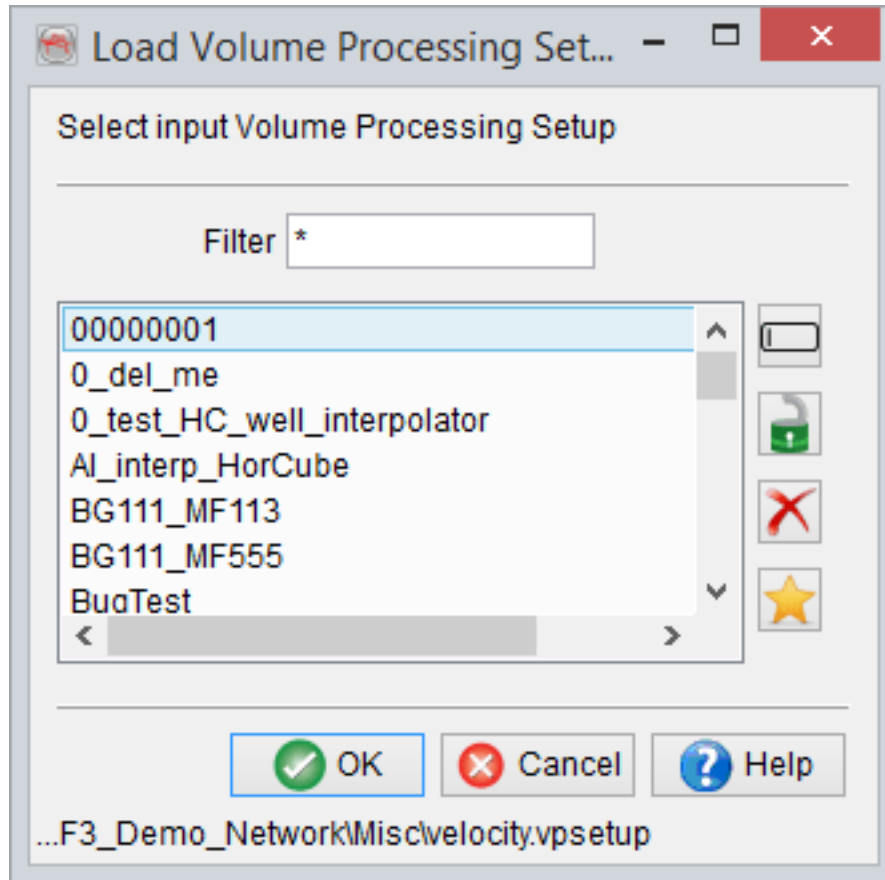
9.5 Velocity Display

Once velocity picks have been made they can be gridded using the volume processing tools. The VMB plugins contains a special tree item that enables a quick application of the volume processing workflow without the need to process the entire volume:



Pressing "Add velocity display" in the right-click menu of a slice (inline, crossline, time-slice) adds a special layer. Its right-click menu does not allow the selection of stored volumes or attributes, but the selection of a stored volume processing (gridding) flow.





The selection of a flow will launch its computation on the current slice only. If you already added a velocity display ,you can either reload it in case of modification or select a different processing setup.



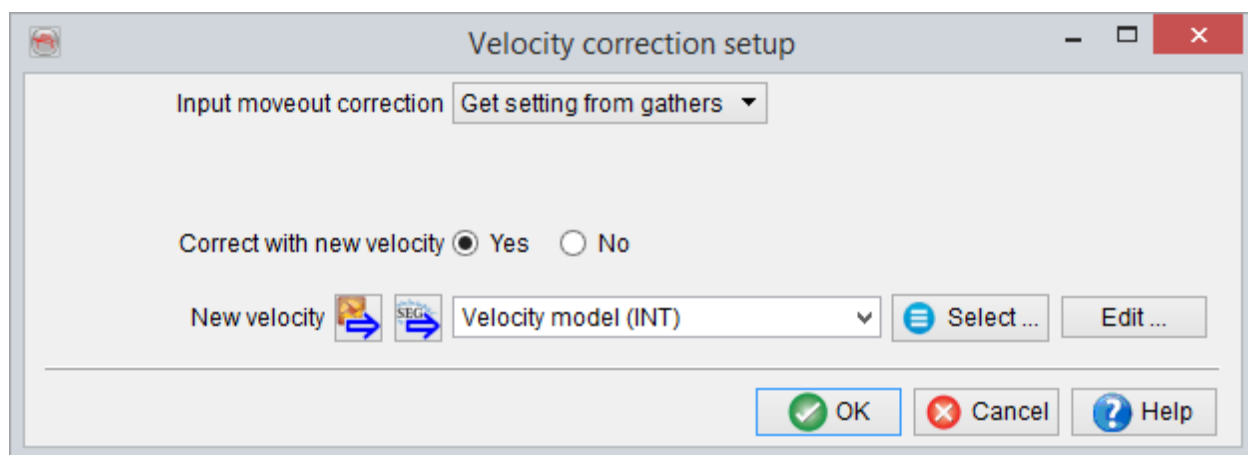
Although the display is for a single slice the computation is still volume-based. Therefore it can take some time to process and will make OpenText unresponsive during that period of time. For large jobs batch processes may still be preferred.

9.6 Velocity Correction

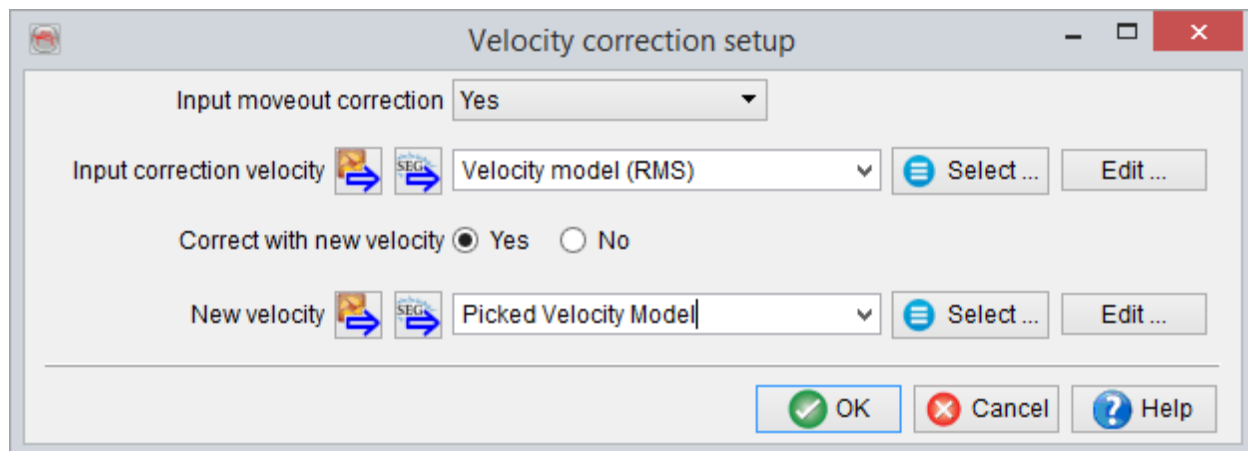
In the Prestack processing steps available in *Processing > Create Seismic Output > prestack processing*, the Velocity correction method will (un-)apply a normal moveout correction based either the migration velocity and/or a new velocity model. An hyperbolic moveout is applied (non-hyperbolic moveout will be implemented later).

The following combinations can be performed:

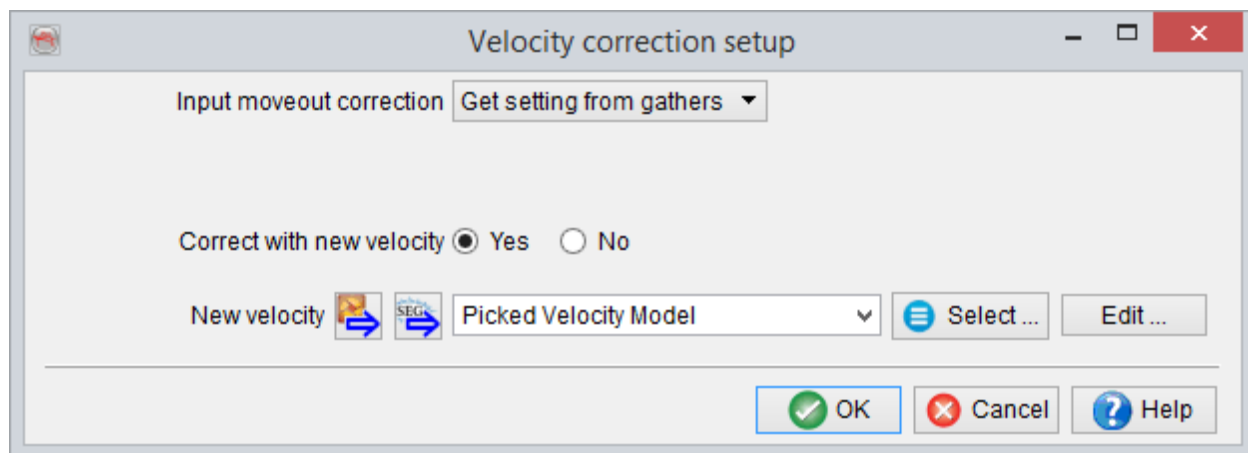
1. The gathers **are not NMO corrected** and the objective is to **apply a NMO correction** with either the migration velocity volume or any other picked velocity model. In that case the following setup must be used:



2. The gathers **are NMO corrected** and the objective is to **undo and apply a a NMO correction with ANOTHER velocity model**. In that case the velocity model used for applying the NMO correction on the stored gathers must be specified (such that it can be undone), and the new velocity model must be selected:



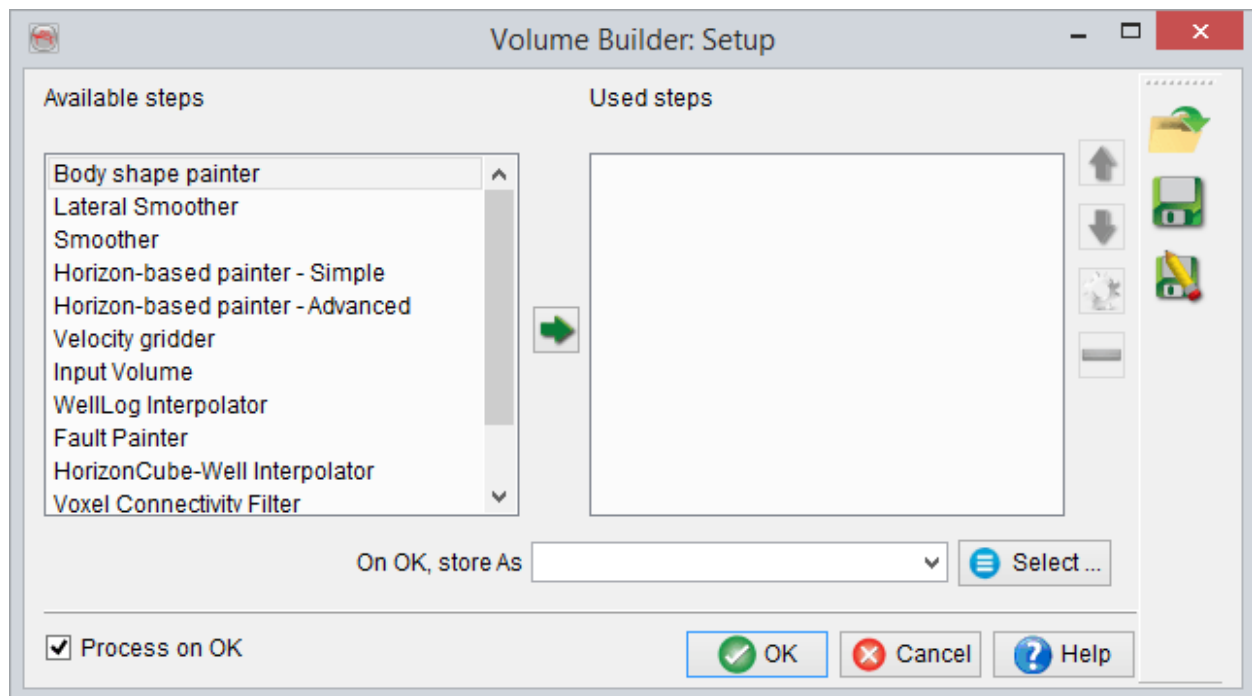
Users of the VMB plugin may have already linked the gathers to a migration velocity volume, and set a flag saying that the gathers are NMO corrected or not. If that is that case the first part can be set to "Get setting from gathers" and one needs only to specify the correction velocity:

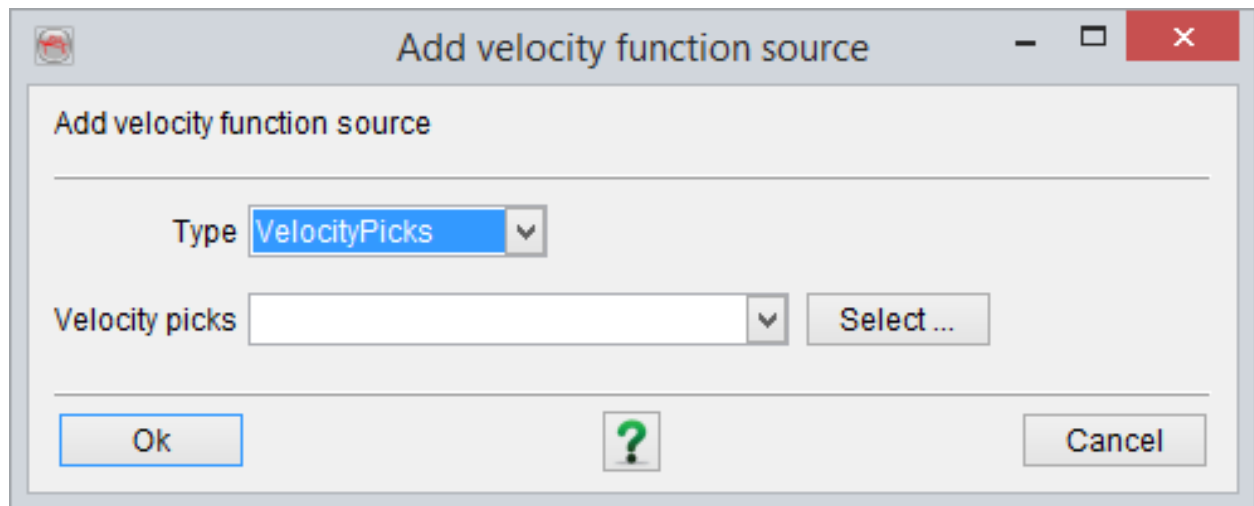
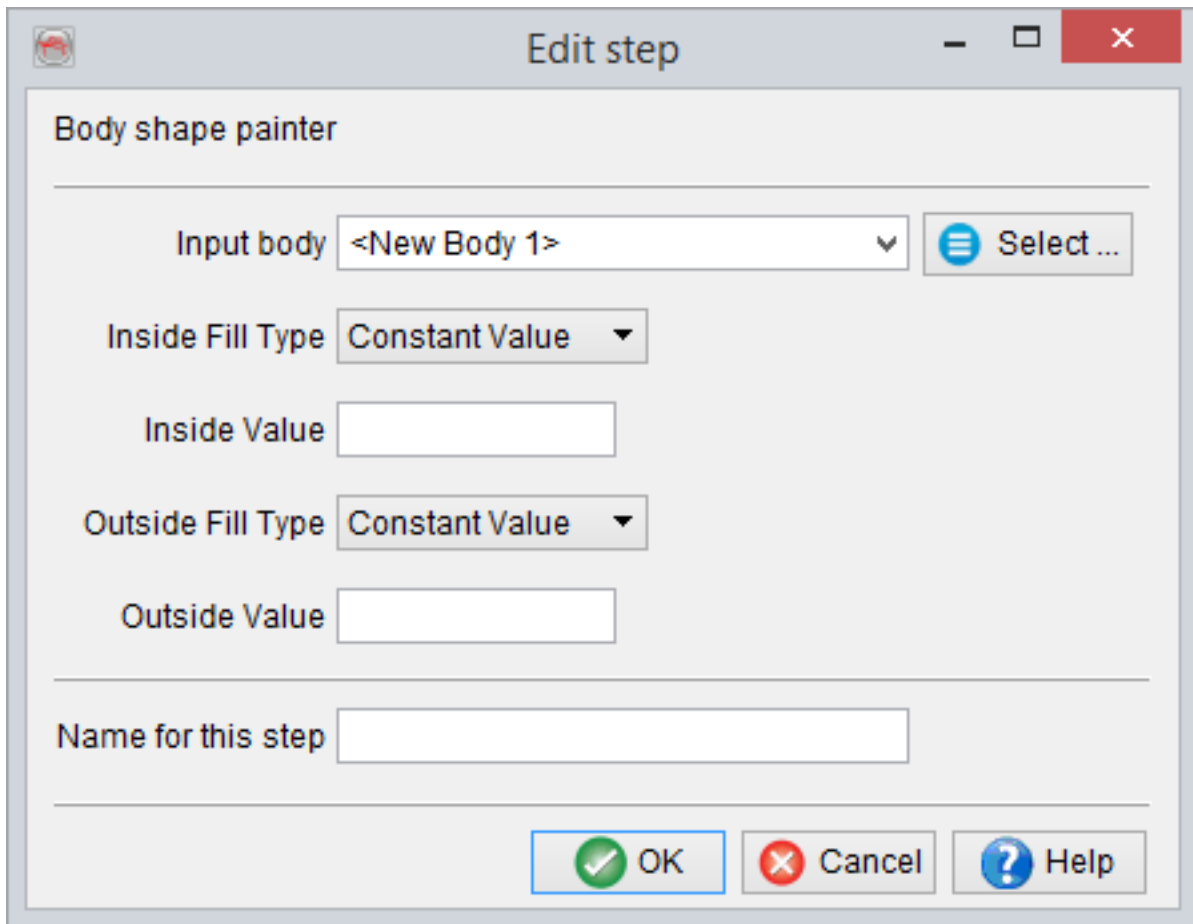


9.7 VMB Specific Gridding Step - Gridding of Velocity Picks

The volume processing tool of OpendText has a step called "Velocity gridder". It can grid either velocity functions or volumes.

The VMB plugin enables the selection of a third velocity source, the velocity picks made in the VMB analysis windows (vertical/horizontal). No parameter is required since the velocity picks are attached to the migration velocity during the [analysis](#). The stored RMO will be converted to interval velocities and gridded in the volume. Please note that the gridding of velocities is done in order to maintain the time-depth relation hold by the velocity function.





9.8 VMB Specific Gridding Step - Surface-limited Filler

The surface limited filler intends to paint velocities in a 3D area whose geometry is defined by one or more 3D horizons. This step may look like the inter-horizon filler but it is actually more powerful.

The Add and Remove buttons should be used to select the 3D horizons to be used in the actual step. The side defines the relative position of the horizon with respect to the area to be painted with velocities. For instance one of the horizon could be a salt flank loaded as a 3D horizon.

The painted velocities are referenced to a specific time. This time can be either constant (user-defined), or retrieved from a 3D horizon and not necessarily from one of the horizons defining the limit of the body. An horizon in between could for instance be used.

Then velocities are painted from that reference time. The velocity must be provided as a velocity/gradient pair. The values are once again either user-defined or extracted from a surface data (grid) attached to a 3D horizon.

Surface limited filler

Name	Side	Color
3-Hor C	Below	
4-Hor D	Below	
5-Hor E	Below	

Start value Constant From Horizon Data

Start value constant

Gradient Constant From Horizon Data

Gradient constant [/ms]

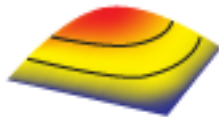
Reference time Constant Horizon

Horizon

Name for this step

10 Fluid Contact Finder

(formerly CCB)




Based on ideas of Jan-Gabe van der Weide and Andries Wever of *Wintershall Noordzee BV*. Wintershall Noordzee BV has granted dGB the usage of the material on which they hold Intellectual Property claims.

10.1 Introduction

This plugin will work only in a 3D survey

Fluid Contact Finder (FCF for short) is a seismic filtering workflow that will stack the seismic traces with respect to the depth of a surface.

It is based on the principle that seismic traces that penetrate the reservoir at the same depth have identical hydrocarbon columns. Seismic traces with identical horizon depth (along a single contour line) may be display using random lines created along contours.

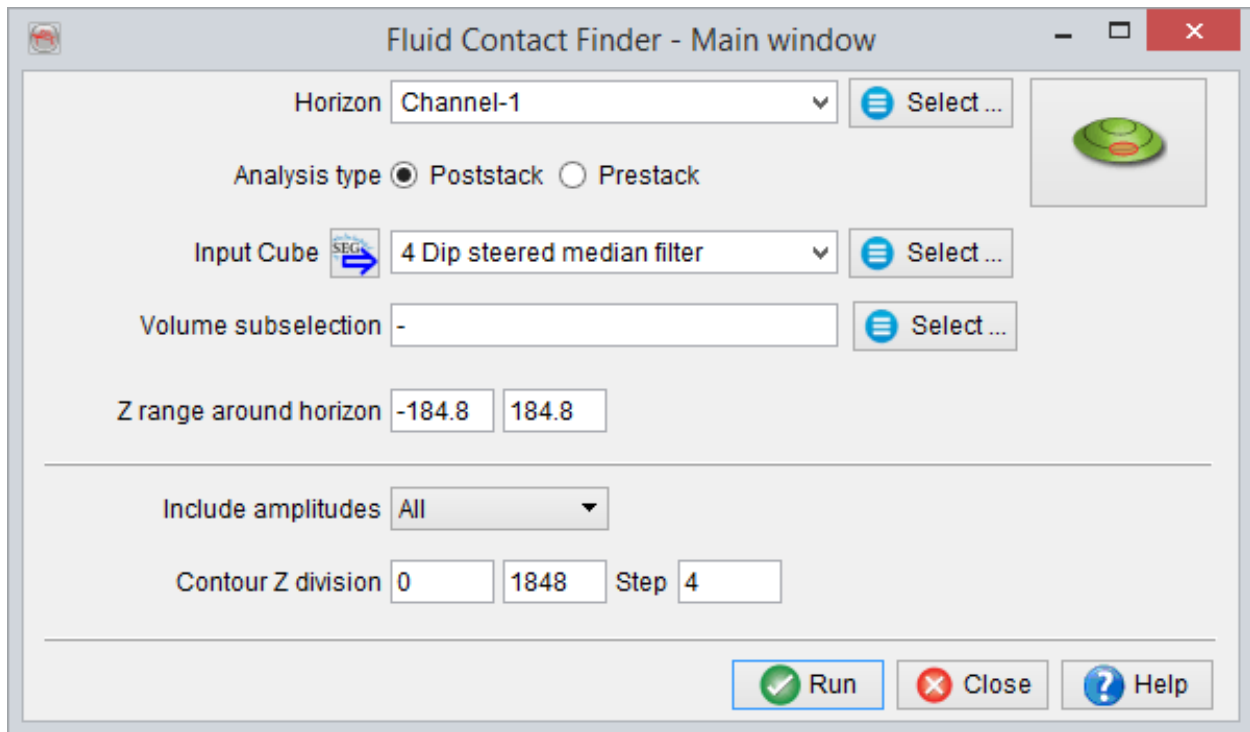
The Fluid Contact Finder may be started from the corresponding icon in the "OpendTect tools" toolbar  , or via the "Analysis" menu using the option "Fluid Contact Finder".

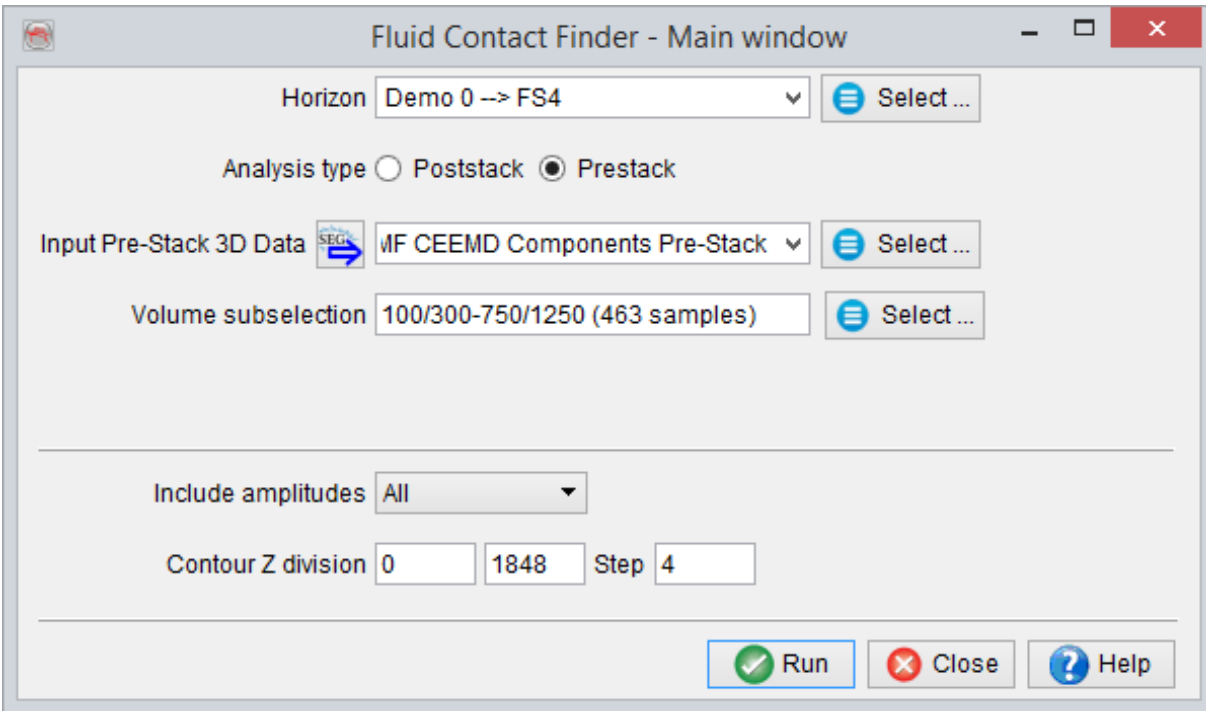
10.2 FCF Main Window

The FCF main window is used to specify the inputs and parameters required for the stacking.

Inputs are:

1. Surface defining the contour lines geometry
2. Seismic volume to stack, post- or prestack.





Parameters are:

1. Volume subselection: Area defining the traces that will be used for stack. This can be an entire volume, a rectangular sub-selection (possibly decimated), a table sub-selection, or a polygonal sub-selection.
2. Contour Z division: start, stop and step value for the bin selection: A bin is a time/depth gate around a contour line $[-\text{step}/2, +\text{step}/2]$. It will be the X-axis of the 2D FCF display. Start and stop values are absolute time/depth values.
3. Z range around horizon: time/depth interval used the extraction of the seismic data prior to stacking. This is a relative gate with respect to the time/depth of the contour.

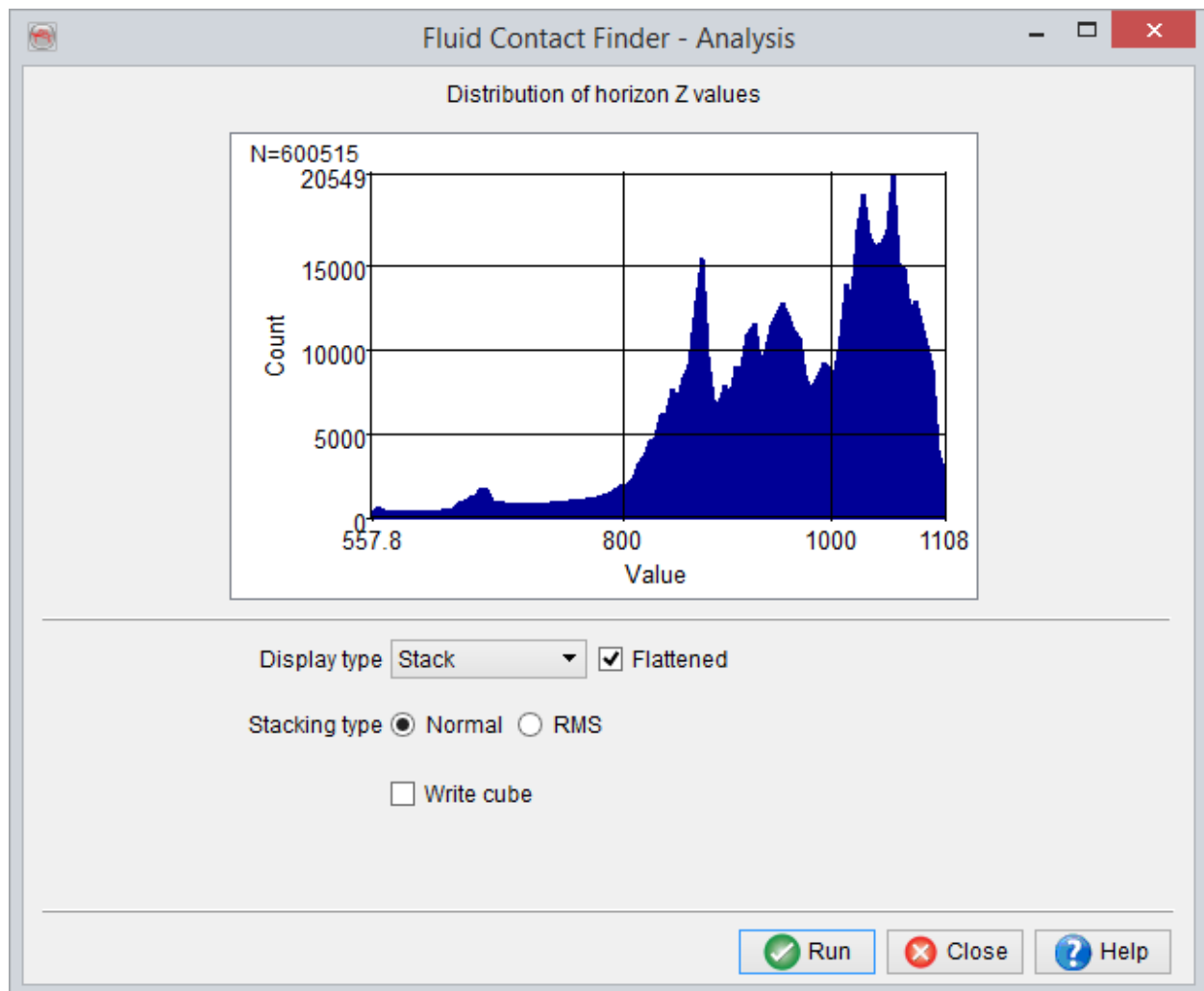
Numbers must be entered in Z unit of the survey: meters, seconds or milliseconds.

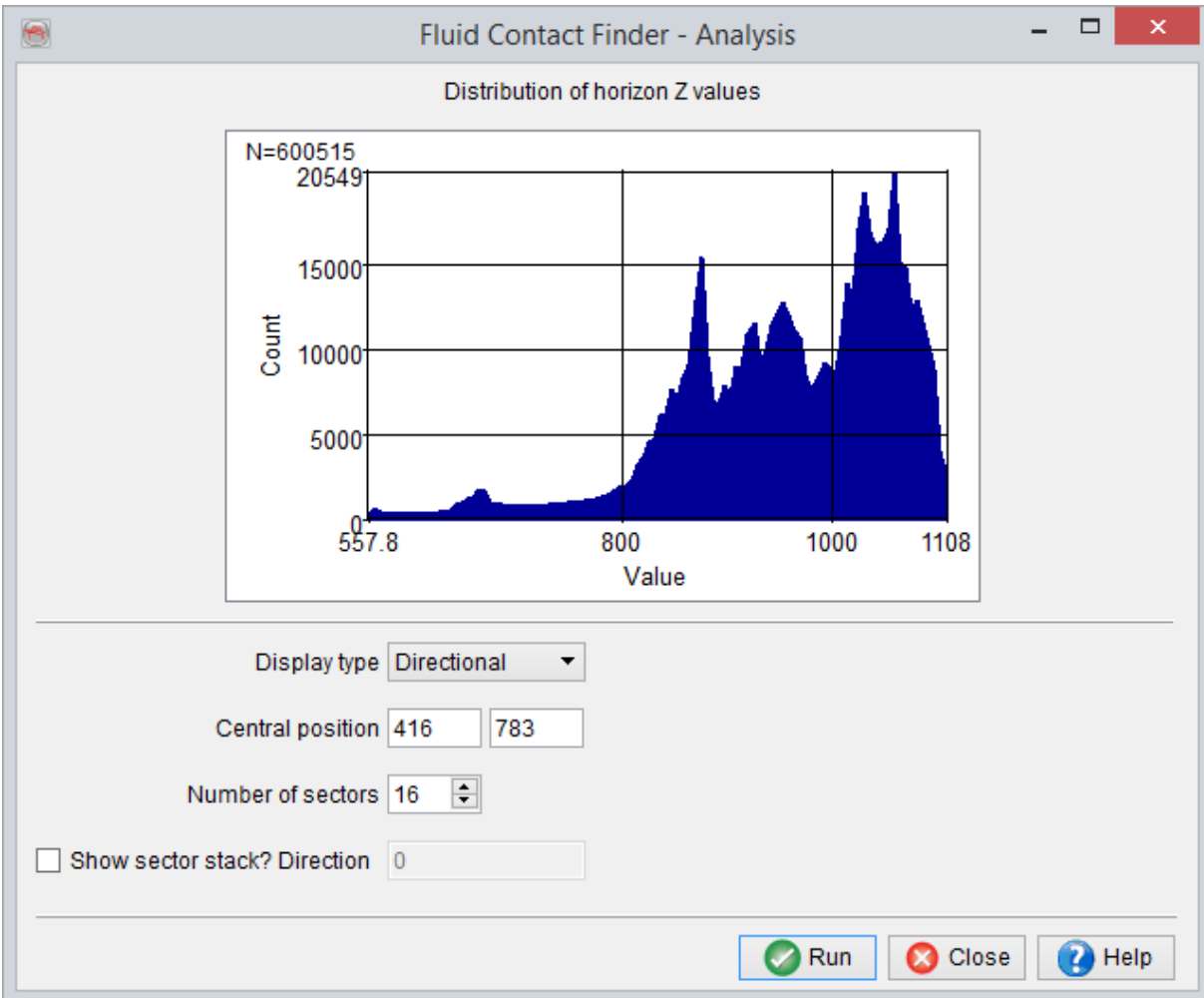
Based on the selected data the traces will be collected based on the time/depth (bin) of the horizon intersection with each trace.

10.3 FCF Analysis

The FCF analysis window has the two following functions:

1. It displays an histogram presenting the number of collected traces per bin (post-stack), or the histogram of amplitudes along the horizon (prestack).
2. It allows choosing the type of display for the output stack, the stacking type used, and outputting the stacked traces to a 3D cube.





The X-axis of the histogram presents bin values in the Z unit of the survey. The Y-axis is the number of traces collected per bin (the maximum count value is shown in the top middle of the histogram).

The display (when pressing the "Go" button) will be in a 2D viewer.

- 'Stack' will display the stacked traces wrt the bin depth (poststack), or the amplitude along the horizon in a crossplot bin depth vs. offset (prestack).
- 'Single Z' will display all traces of a single bin *before* stack.
- 'Directional' will display a FCF-stacked amplitude map along the horizon as a function of the distance to a central position and the azimuth sector. The central

position is pre-computed as the shallowest position with the selected area. It can be modified by the user, as well as the azimuthal sectors parameters.

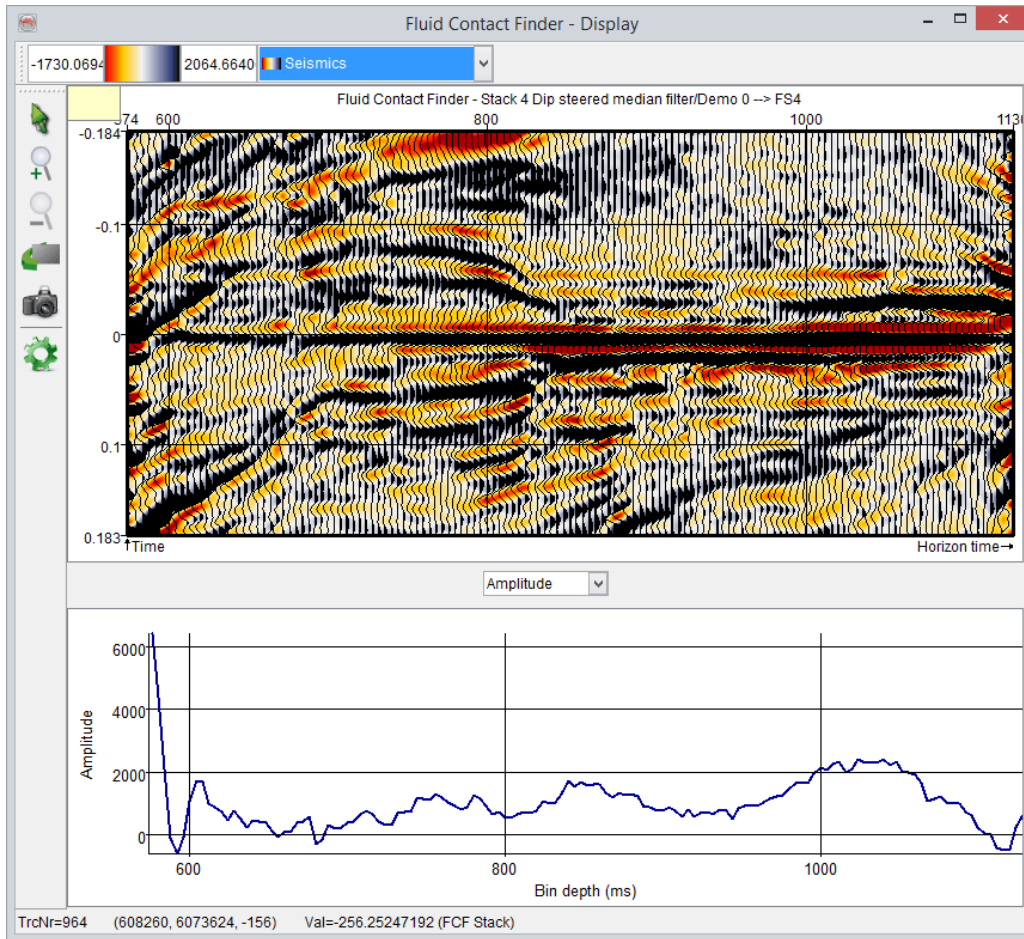
Optionally, a 3D volume may be output. In this volume each input trace is replaced by the stacked trace of its corresponding bin. The output volume can then be used to make crossplots and update amplitude maps for instance.



The FCF main window remains open when using and after closing this FCF analysis window. Multiple analysis windows can be created, with different volume subselections for instance. Nevertheless the main FCF window must remain opened to perform the stacking.

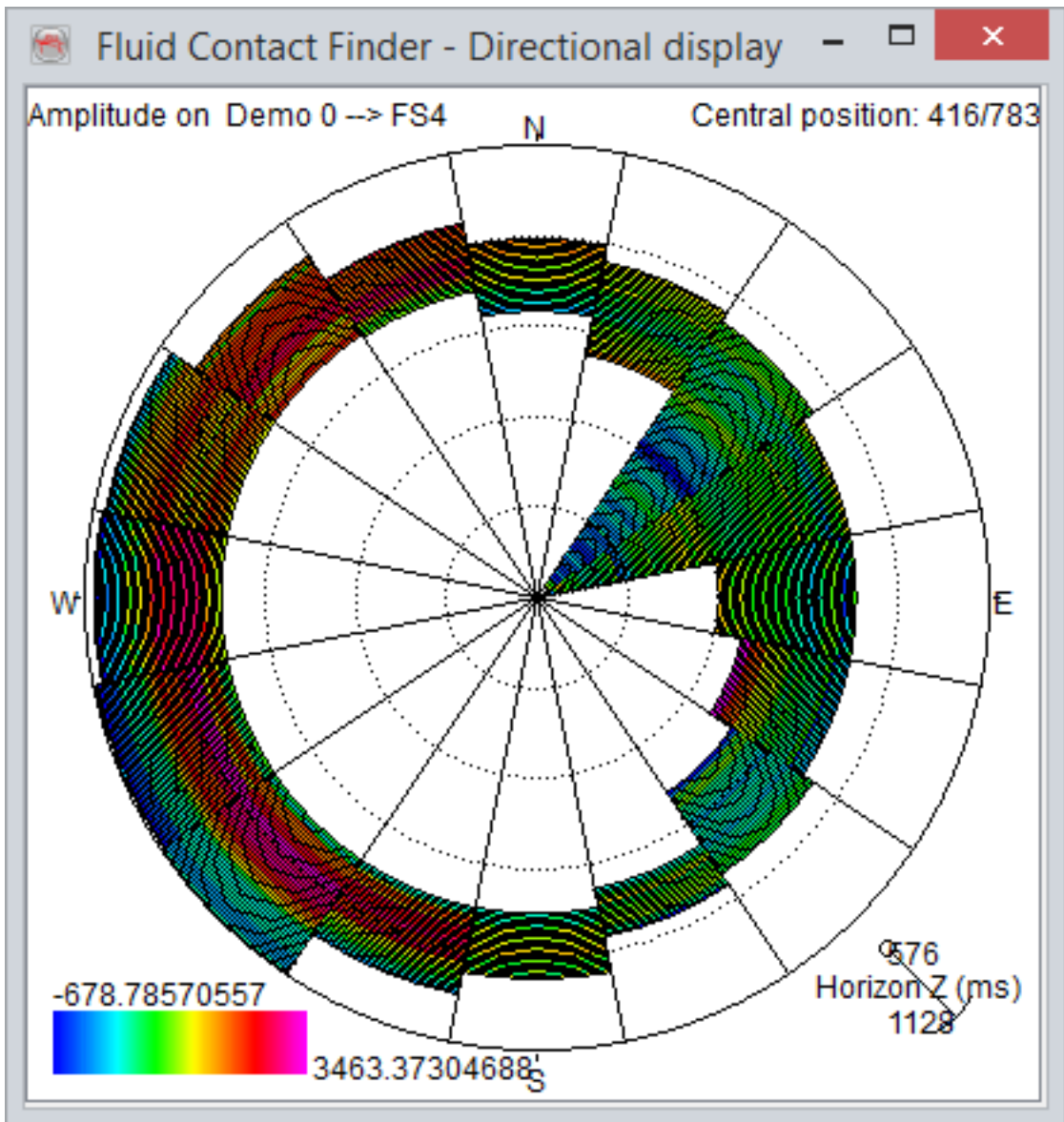
Example of 2D stack

The following figure presents an example of 2D stack. Mind the increase of amplitude at 2130 ms. A crossplot of amplitude, frequency and phase is presented vs. bin depth is presented in the lower part. In the prestack FCF the crossplot shows AVO attributes



Example of FCF directional display

The following figure is an example of FCF directional plot. The colours represent the amplitude along the horizon. The grid represents the grid used for the binning of the seismic trace before stack.



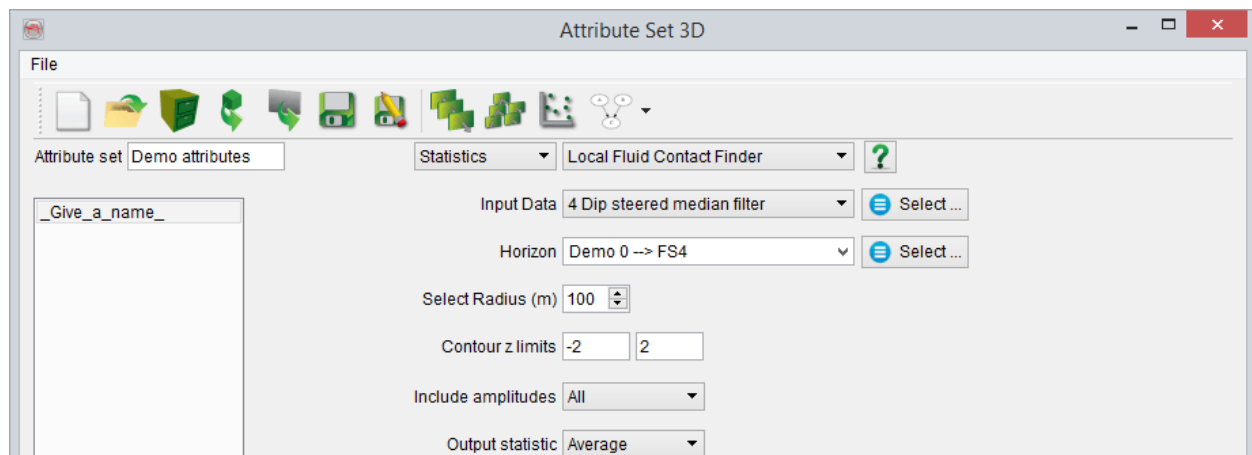
10.4 Local FCF Attribute

This statistical attribute makes local averages according to a given 3D horizon. Its primary application is the filtering of seismic data in the depth domain with the aim of removing the structural footprint and enhancing hydrocarbon-related flat spots. The output is a 3D volume with the same geometry as the input volume.

This attribute is comparable to the volume statistics attribute: The data is extracted in a user-defined radius within a time gate, and a statistic is output (average, median, RMS). The default time gate is +/- half the survey sampling rate.

The main difference is that only positions that have the same distance (+/- the time gate) between the actual sample and the pick of the 3D interpretation will be kept for the computation of the statistic. All other will be discarded. Therefore the samples extracted could be looked as a portion of contour lines computed from the 3D horizons, and shifted at the depth of the actual sample.

The radius of the data selection corresponds to the aperture of the migration: It limits the amount of data in a lateral sense collected and used by the stacking operator. Its unit should be a real world length in the same unit as the survey coordinates.



11 Applications

11.1 How to Make the ChimneyCube

In the following, we will assume that you have selected a survey, imported a Seismic Cube, and calculated the SteeringCube. We also assume you have generated a default Chimney Cube attribute set. (For Default attribute-set see Chapter below)

11.1.1 Workflow

Creating a ChimneyCube is a simple process. The only input needed is picks from example locations, where chimneys have been identified along with counter examples, i.e. points that do not belong to a chimney. At these example locations, attributes will be extracted to train a neural network in order to classify the data into chimneys and non-chimneys. Typically, all attributes in the set are used but it is possible to select a subset only, e.g. to speed up the processing time. The ChimneyCube network has two output nodes representing chimneys and non-chimneys. Each node is given the value one (true) or zero (false). In other words, the network tries to predict a vector (1,0) when the example is a chimney and (0,1) when the example location is not a chimney. When the trained network is applied, it is sufficient to output only the chimney node. A value close to one describes a high chimney-probability at that specific location, while a value close to zero indicates low probability.

The procedure to create the ChimneyCube is not a true classification process. This, due to the output not being binary but rather continuous.



However, the neural networks plugin for OpendTect does support a true classification output. This procedure is especially useful for classifying data into more than two classes, e.g. for supervised facies classification. If

Classify output in the Create Neural Network window is set to "Yes", the network output consists of two nodes: Classification and Confidence. The Classification output returns the winning class, i.e. an integer number between 1 and N, where N is the number of classes. The Confidence indicates how close the output vector is to the optimal vector representing the winning class (e.g. 0,1,0,0 represents the second class in a 4 class problem). The match has a value between 1 (perfect) and 0 (very poor).

11.1.2 Picking Example Locations

Picking chimneys and non-chimneys is done in the graphics window using the plane viewers. Unless you know your data so well that you can immediately go to the inlines, crosslines, or timeslices (where you wish to pick chimneys), we recommend starting by making a few *similarity slices* at different times. Select the Time-plane viewer and add a new element (right-hand mouse button, select *Add*). Position the plane: click on the interact mode icon, click on the plane followed by a click and drag on one of the frame arrows. The position is displayed in the status bar below the scene. Click outside the frame when you have reached the desired position. The select view data menu pops up. Load one of the similarity attributes (option *Attributes*). If the *Attributes* option is not available and you can only select from the "Stored" data cubes, you have no active attribute set; create one in the attribute set editor from the *Processing* menu. Loading a similarity attribute from the set means that the attribute is calculated on the fly. This may take a few seconds depending on the data size and the hardware you use. To use similarity slices in context of processing a chimney cube check the display and look for circular patterns or other anomalous dots.

Display a seismic inline through one of the possible chimney locations (inline viewer). Let us assume that you have indeed identified a chimney on this line. You may want to display one or more attributes in a small window around the chimney (add another inline viewer, position it and resize it) to check how the chimney appears on single-attributes. (To compare different attributes you can add more viewers that can be toggled on/off in the tree, or you can add another scene (Windows menu, option *New*). You can also use the "Swift" icon to apply attributes from an open attribute set directly to an active element.

Having done this you are now ready to start picking chimneys and non-chimneys. The first action required is to create two new picksets: one for chimneys and one for non-

chimneys. Use the right-hand pop-up menu from the picksets entry in the OpendTect tree to do this. Type the name of the pickset you wish to create, e.g. *chimneys_yes* and start picking locations (left-hand mouse button) inside the chimney. Picks can be removed by pressing *Ctrl* and left-hand mouse clicking the pick. Click on the data element in the OpendTect tree to be able to move the element to another position and repeat the process. Repeat this exercise until you have sampled all chimney points you need. Now select the *non-chimneys* pickset from the OpendTect tree and start picking non-chimney locations. When you are done save each pickset in a separate pickset Group (*Store* pop-up menu option from the tree).

Picking strategy: Picking example locations is the key step in this procedure. You should aim to create representative sets for both chimneys and non-chimneys. Try not to limit yourself to one chimney if there are more chimneys in your data set. Try to sample these consistently and over as wide a time-range as possible.

The default chimney attribute set is tuned to find chimneys that are characterized by energy variations, low trace-to-trace similarity and chaotic reflection patterns (variations in local dip). The default attribute set extracts attributes in 3 separate windows of 80ms that are aligned above, around and below the evaluation point. This arrangement will help the network to distinguish between a vertical disturbance (of approx. 240ms) and a local disturbance that is not vertical, hence should be classified as non-chimney.

So, if you use the default chimney set, you should select example chimney locations inside zones where the energy and the reflection pattern are disturbed over a vertical zone. non-chimneys should be picked at locations where you see good reflectors but also at locations of non-vertical disturbance (e.g. at faults). A typical set for training consists of several hundred to a few thousand points.

11.1.3 Neural Network Training

The next step is the training of a neural network. Click the corresponding icon or start the [network module](#) from the Processing menu. Create a New neural network, and select the attributes you wish to use (normally all) and the pickset groups containing the chimney and non-chimney locations. In general not all locations are used to train the network but a percentage (10 to 20) of the examples is used to avoid overfitting the network. The network will extract the attributes of your choice at the locations you specified, it will randomly split the data into train and test sets and it starts the training phase.

Training performance is tracked during training and presented in two figures. The normalized RMS error curves indicate the overall error on the train and test sets, respectively on a scale from 1 (maximum error) to 0 (no error). Both curves should go down during training. When the test curve goes up again the network is overfitting. Training should be stopped preferably before this happens. Typically an RMS value in the 0.8 range is considered reasonable, between 0.8 and 0.6 is good, between 0.6 and 0.4 is very good and below 0.4 is excellent.

A better feel for the network's performance is presented in the lower figure, which shows the percentage mis-classified for train and test sets as a function of training cycles.

Finally, you will notice that the nodes of the network change colors during training. The colors indicate how important each node (each input attribute) is for the classification. The colors run from red (most important) via yellow to white (least important).

Overtraining: Overfitting occurs when a network starts to recognize individual examples from the training set. The network performs better on the training set, but the performance on the test set decreases. For optimal results network training is stopped when the performance on the test set is maximum (minimum error). The point to stop can be seen from the performance graphics in the network training window.

11.1.4 Evaluation and Application of the Trained Neural Network

Now that you have a trained neural network, you are ready to create TheChimneyCube®. Before applying it to an entire volume, you may want to check the result on a few selected planes. Pop up the Select attribute data menu by clicking with the right-hand mouse button on a plane element in the tree. Note that the neural network option is now active. The network has two output nodes: chimney and non-chimney. Select the chimney node to generate the desired display.

When satisfied you can continue by applying the trained network to the entire data volume. This is done in the *Create Volume* module that is launched from the *Processing* menu. Instead of processing an entire cube it is also possible to limit the output range to a sub-volume. To increase speed, run the job on several machines simultaneously in multi machine mode. OpendTect will split the jobs over the specified machines and will combine the output at the end of the processing.

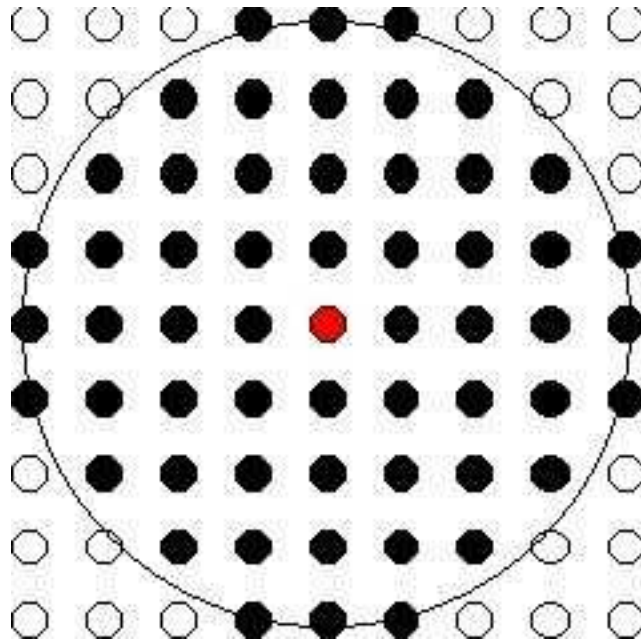
For more details on Chimney analysis workflow, look at : [dGB's How To Videos on YouTube](#).

11.2 Dip-Steered Median Filter

The dip-steered median filter is a data-driven tool that yields a cleaned-up seismic data volume in which coherent events are enhanced and randomly distributed noise is reduced. The filter increases the general interpretability of the seismic data and improves the performance of automatic horizon trackers. Basically the filter collects all amplitudes inside a disc with user-specified search radius and replaces the value at the center by the median value of the amplitudes. The search disk (see graphic below) follows the local dips from the [SteeringCube](#).

The filter, in combination with the SteeringCube works as follows:

1. A search radius is defined.
2. From a starting position (**red** dot) we extract the first amplitude.
3. The local dip and azimuth is followed to the next trace.
4. The interpolated amplitude at this point is extracted.
5. Step 3 and 4 are repeated for all traces inside the search radius (see Figure).
6. The amplitude at the starting position is replaced with the median value of all extracted amplitudes.
7. Points 2 to 6 are repeated for all samples in the cube.



Filter input for a 4-trace radius which corresponds to 57 points. Note that the disk is neither flat, nor horizontal but follows the seismic events from trace to trace.

A median value can be defined as the value associated with the central position of a ranked series. So, if we rank all N amplitudes from smallest to largest number than we find the median value by taking the value at position $(N+1)/2$, where N is an odd number. To understand the effect of a median filter, let us assume we are filtering a seismic event with a 3-points median filter. The event, e.g. the amplitudes along a horizon is given by the following series:

...0,0,1,0,0,1,1,3,1,0,1,1,1,.....

The 3-points median filtered response is given by:

....0,0,0,0,0,1,1,1,1,1,1,1,1,.....

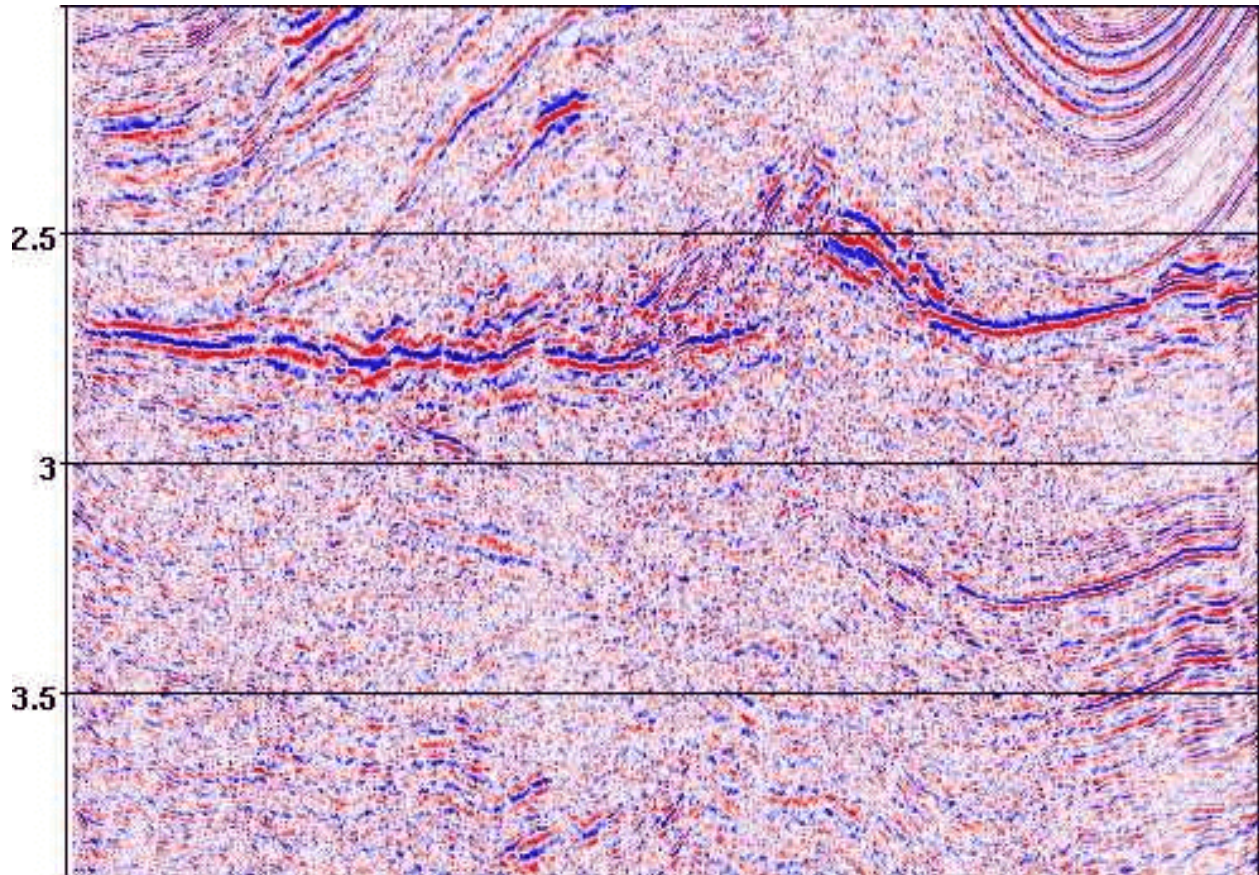
To check this take 3 consecutive input numbers, rank them and output the value in the middle, then slide your input set one position and repeat the exercise.

Please observe that:

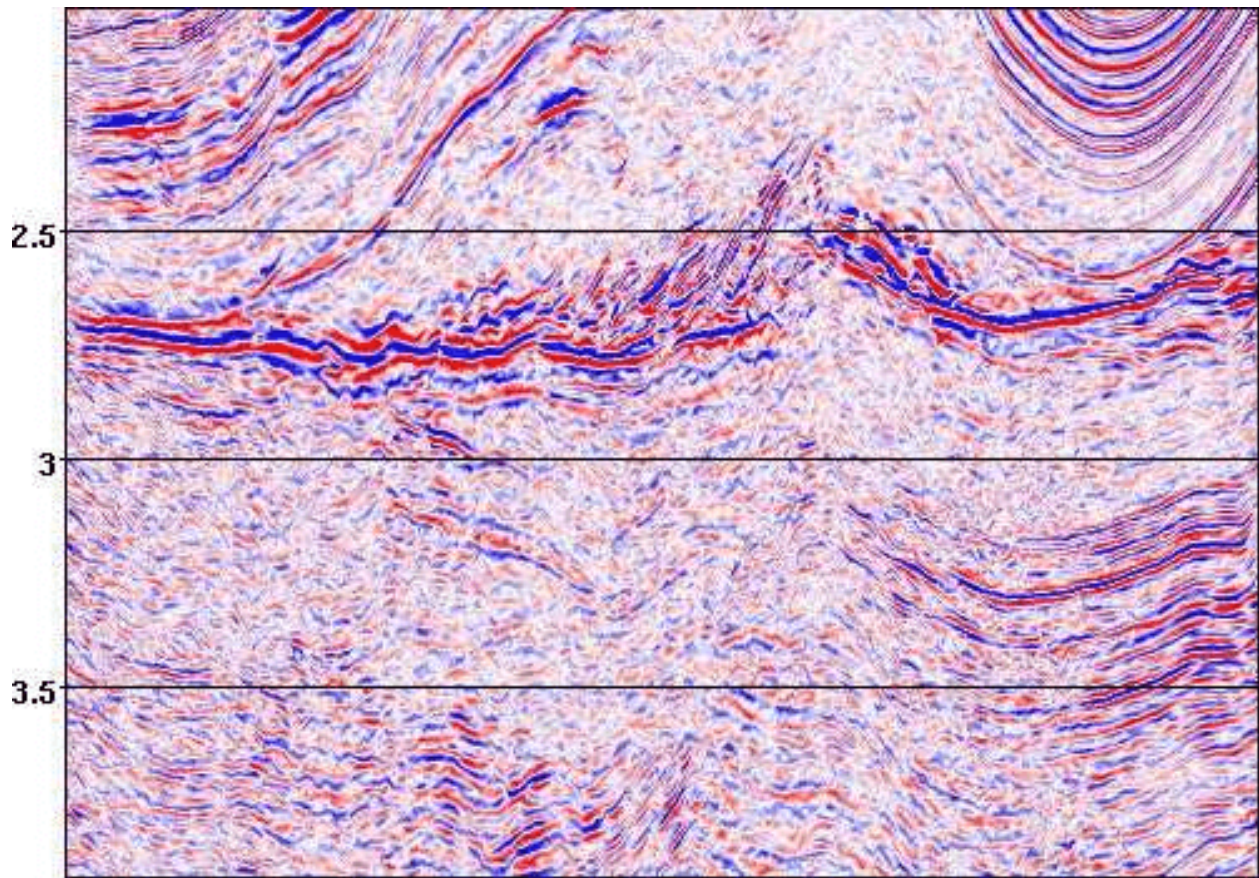
1. Events smaller than half the filter length are removed (e.g. the 1 on the left and 0 on the right)
2. Noise bursts are also removed (the value 3)
3. Edges are preserved (the break from mainly zeros to mainly ones stays exactly at the same position. In other words no filter tails are introduced).

11.2.1 Example Results

The following figures show an example of the effect of a 57-points (4 trace radius) filter.



Original seismic.



Seismic section from a 3D volume after 57-points dip-steering-median-filtering.

11.2.2 Create a Dip-Steered Median Filter

In OpendTect, filters are integrated with attribute extractions. The advantage of this approach is that all attributes can be filtered separately without creating intermediate results first. To apply a median filter to a seismic data set, you define the median filter as an attribute in the *active attribute set*. Start the attribute definition module (icon or Attributes option, Processing menu). Select an existing attribute set, or create a new one. Specify *Volume statistics* as attribute type, select the input seismic and specify *output = median*. Set the search radius e.g. 4x4 and specify the time gate as [0,0]. Select the *SteeringCube* and specify Full steering as the steering mode. A time-gate of [0,0] means that effectively the filter input is collected along a disk. Full steering means that the disk is curved according to the local dip information (see also Create Steering Data).

To apply the filter interactively, use the plane viewers in one or more scenes. For a good comparison between filtered and unfiltered displays, scale the data similarly (clipping option, right-hand mouse button on the color bar). To apply to a volume, select Volume output from the Processing menu, specify the dip steered median filter attribute as '*Quantity to output*'.

11.2.3 Note

As an alternative, one could also use the *Edge Preserving filter* described by Li You et al. in *The Leading Edge* of February 2002. Just like the dip-steered median filter, this Edge Preserving filter is included in the *Evaluate Attributes* default attribute set. It uses the [Position](#) attribute to locate the area where the variance in seismic amplitude is lowest, and outputs the average amplitude at that location to the current sample location.

12 Default Attribute Sets

The steering and neural networks plugins for OpendTect are provided with *default attribute sets* to help get the user started. The sets have proven their value in several studies, and generally deliver good results in their respective applications. Attribute sets starting with NN are meant as input for a neural network and are optimized to detect certain geological features (ChimneyCube, SaltCube etc ...), The other attribute sets have different other purposes.



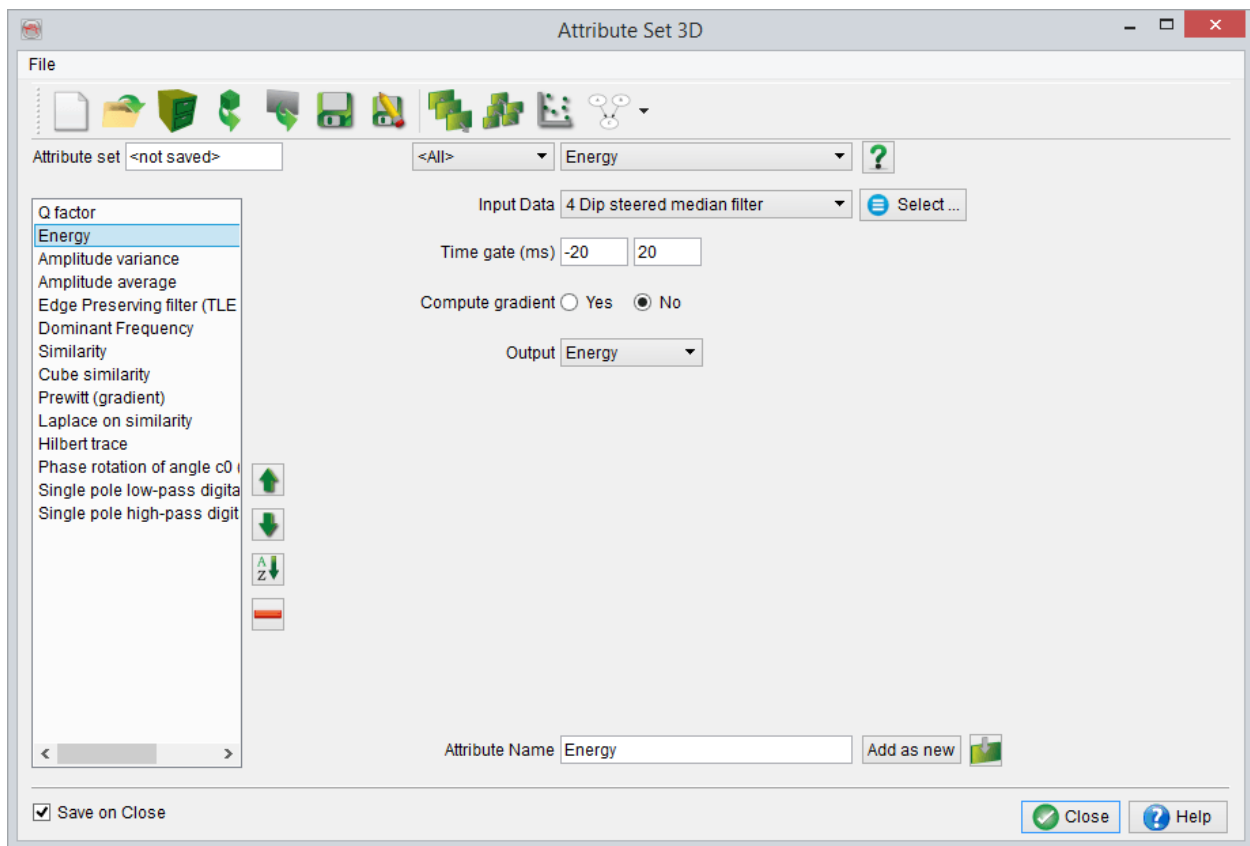
All default attribute sets need a SteeringCube.

In general, default sets give satisfactory results and therefore inexperienced user can use the sets without modifications. Experienced users can use the sets as starting point for attribute analysis. Fine-tuning is done by modifying attribute parameters, and/or adding or removing attributes. To give the user a good idea about the applicability of the default attribute set, typical examples are provided in the next sections of this appendix. All examples contain a short description of the attribute set and its characteristics, an example of seismic data and the result after applying the default attribute set to this data.

12.1 Evaluate Attributes

The "*Evaluate attributes*" is a default attribute set which gives the user the possibility to find the best parameter setting for a particular attribute. To evaluate attributes, use your visual inspection, common sense, seismic knowledge.

Only for this particular default attribute set, the use of dGB plugins (SteeringCube and/or Neural Network) is not needed.



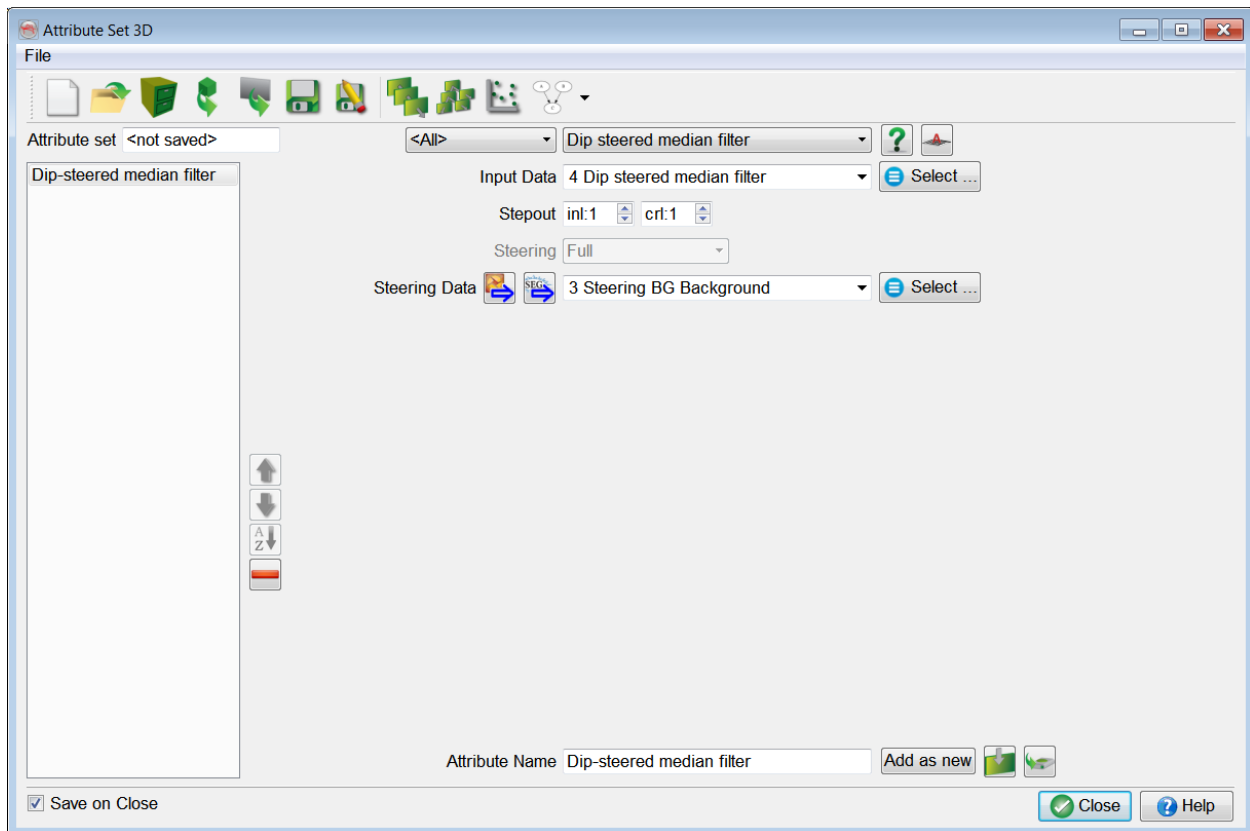
"Evaluate Attribute" default attribute-set

For more details on this topic, please look at [dGB's How To Videos on YouTube](#).

12.2 dGB Evaluate Attributes

The "*dGB Evaluate Attributes*" contains a general selection of various attributes, including steered attributes (polar dip, similarity, curvature etc.) and filters (dip-steered median filter and edge preserving filter). It is intended as a guide or starting point for a scan through the wide range of different attributes, and may function as a starting point for a custom made attribute set.

The difference with "Evaluate Attributes" is that dGB Evaluate Attributes use dGB plugins (with SteeringCube).

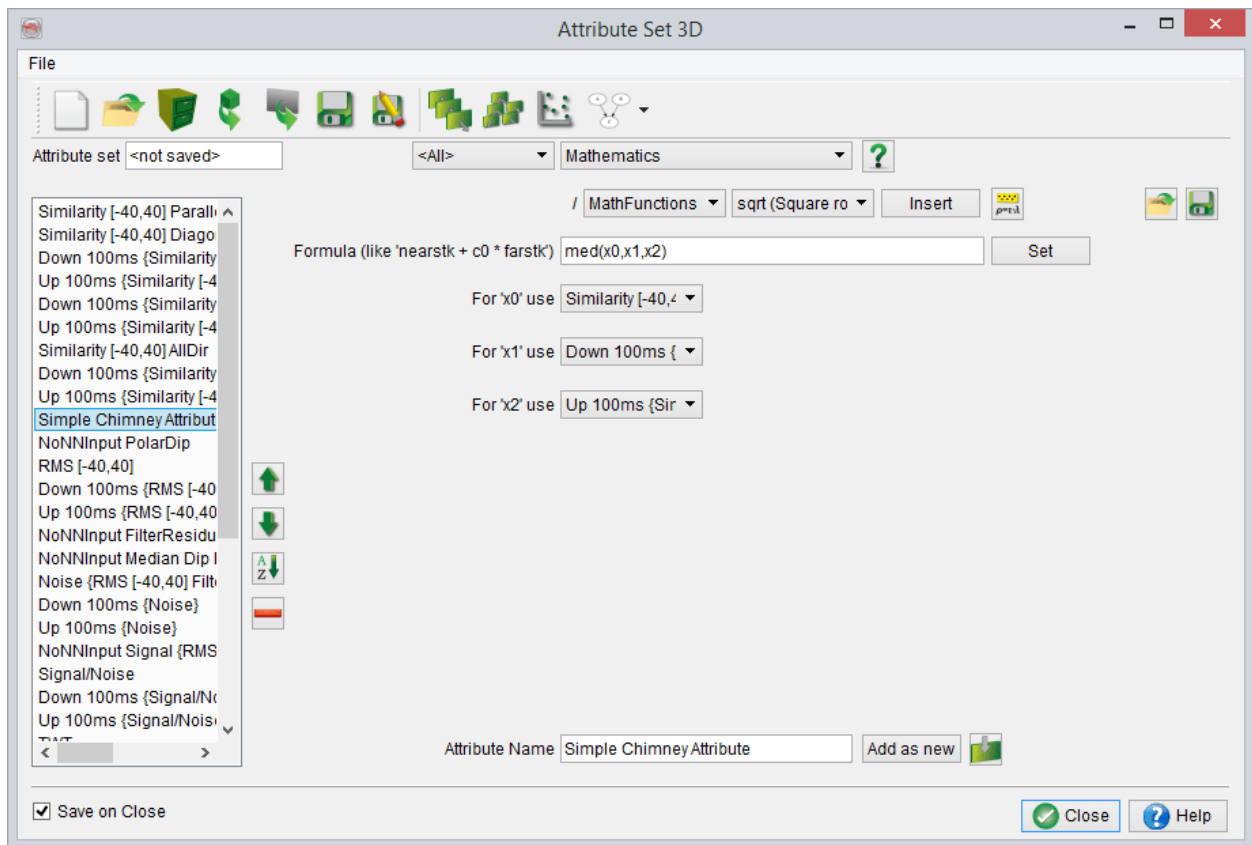


"dGB Evaluate Attribute" default attribute-set

12.3 NN Chimney Cube

This attribute set is meant for usage in a neural network. A key feature of this set is that most attributes are extracted in three separate time windows: one above, one centered around, and one below the point of investigation. In this way, we utilize the fact that chimneys are vertical bodies with a certain dimension. It is expected that similar seismic characteristics of Chimneys are present in all three windows and thus a correlation exists between these three windows. The neural network will recognize this and will thus be able to distinguish between real chimneys and other (more localized) features.

In this section, we compare the original seismic in Figure 1 with the results of chimney detecting neural network displayed in Figure 3. The main body of the chimney and its sidetracks are picked up, while other features (faults, low similarity/low energy layers) are rejected. In addition we compare a similarity attribute in Figure 2 with the neural network results in Figure 3. The similarity attribute highlights the chimney, but also other (unwanted) features are enhanced. The multi-attribute neural network is able to make a clear distinction between chimney and non-chimney. Neural networks with the NN ChimneyCube attribute set as input can (and in general will) also detect other fluid migration paths, e.g. dewatering structures.



"Chimney Cube" default attribute-set

Figure 1. Chimney in seismic data

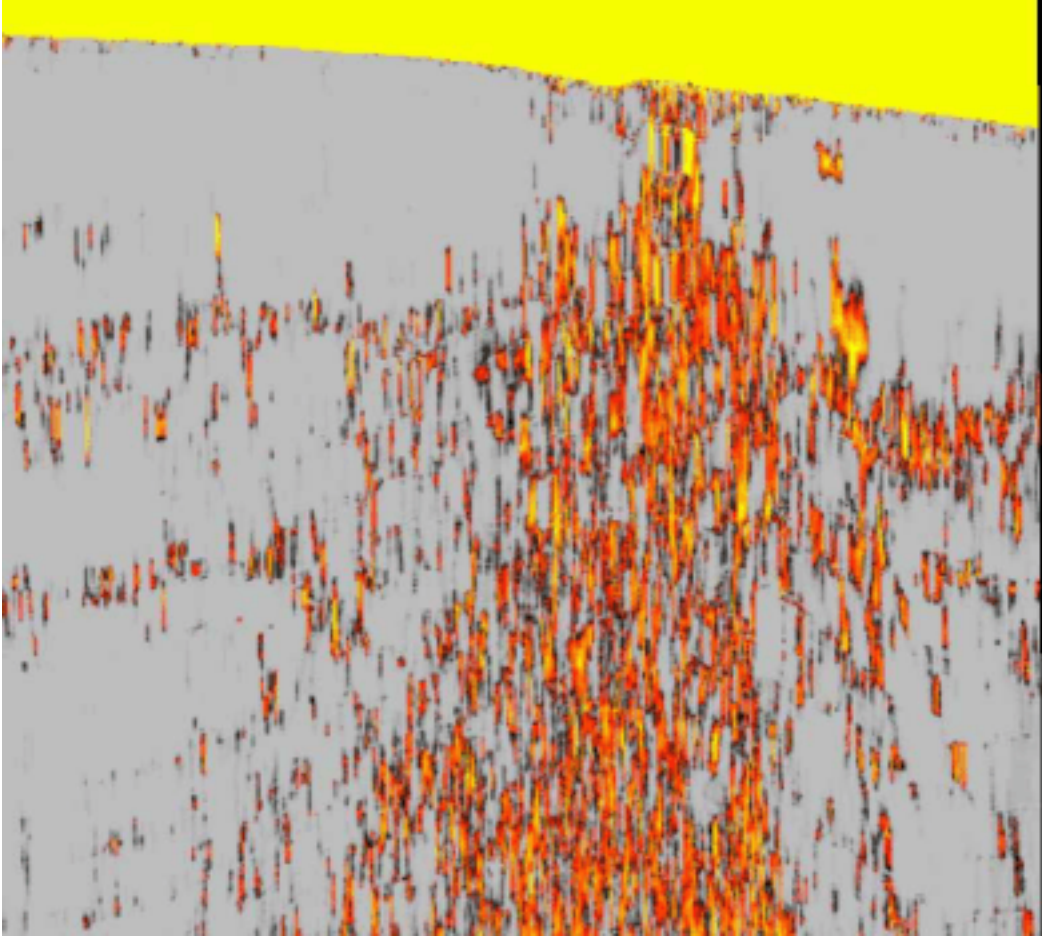


Figure 2. The similarity attribute.

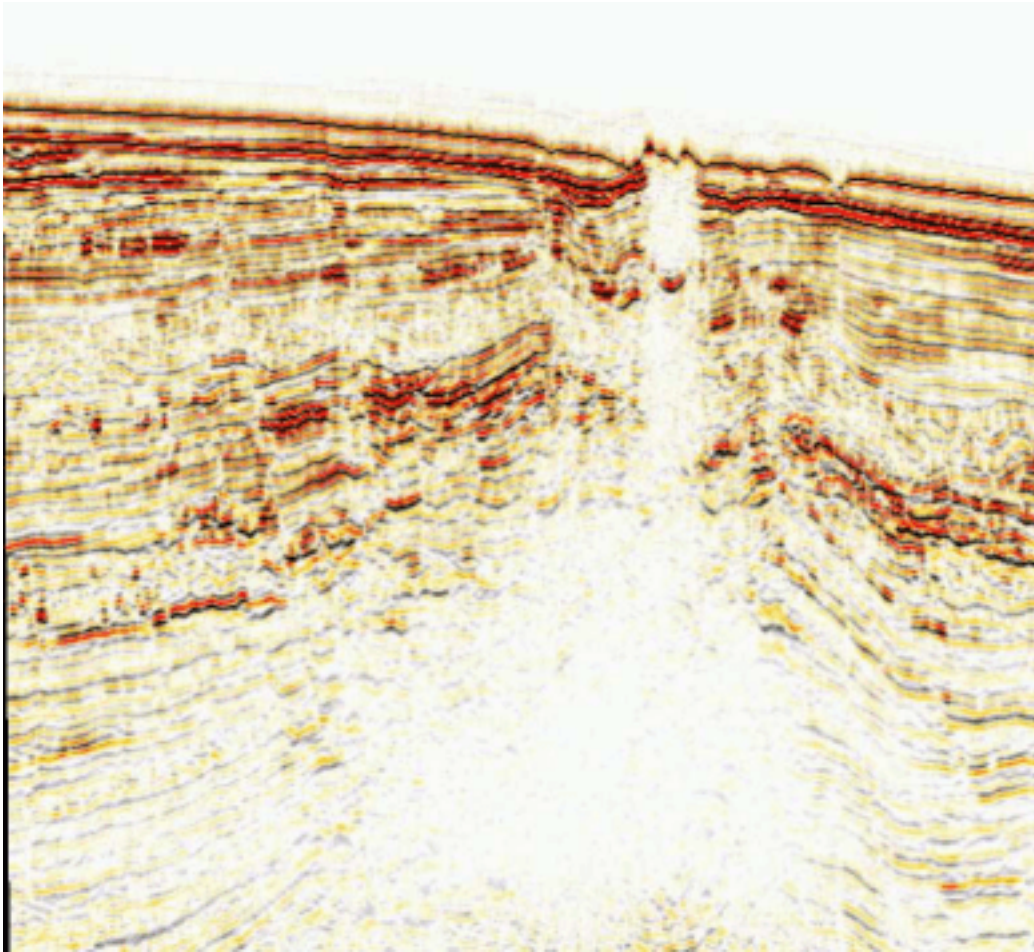
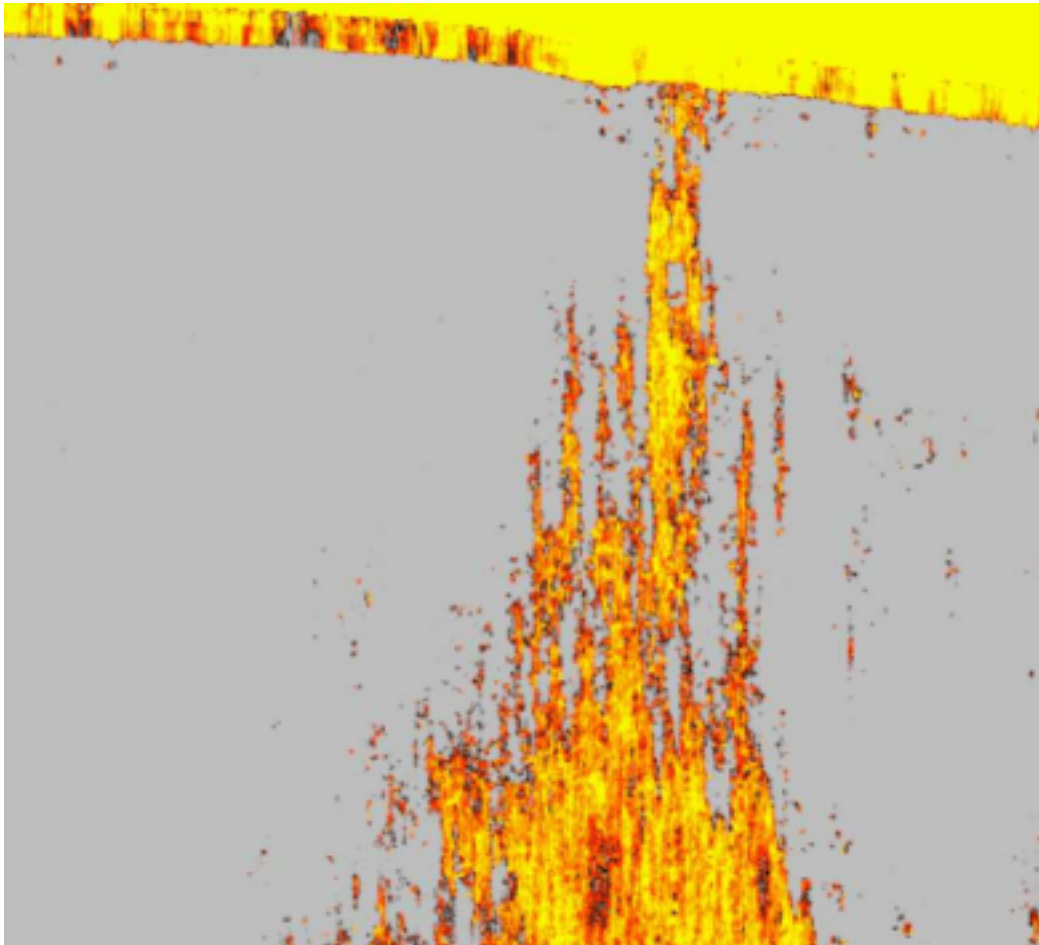


Figure 3. Result after applying a neural network with the NN Chimney Cube as input.

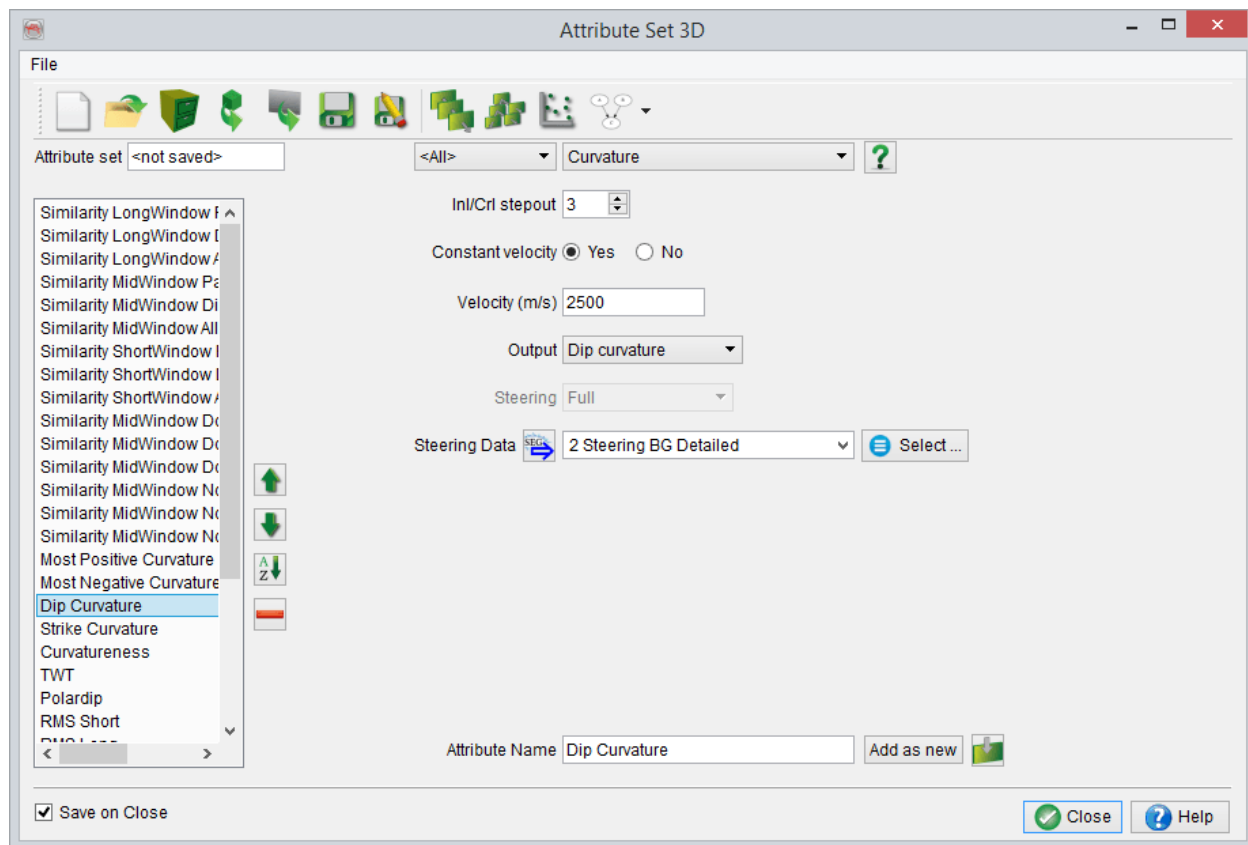


For more details on chimney analysis workflow, look at : [dGB's How To Videos on YouTube](#).

12.4 NN Fault Cube

This attribute set is meant for usage in a neural network. The attributes are tuned to pick up larger and smaller lateral discontinuities in the data. Depending on the character of faults on seismic, the parameters of the attributes can be modified. The defaults provide the best detection of steeply dipping faults of 1 to 3 traces wide. With wider faults or faulted zones longer windows and larger step-outs may improve the results. More flat lying faults are better detected using smaller (vertical) windows

Figure 4 shows seismic data with steeply dipping faults and Figure 5 shows the neural network generated fault cube result. Similarity is the most important attribute in fault detection, but the other attributes (energy, polar dip, dipvariance) enable the neural network to distinguish between faults and other low similarity features, e.g. chimneys or salt layers (provided enough counter-examples are picked for training). Also, they increase the fault continuity. For example, the more chaotic character of seismic data at a fault location is detected by the dipvariance attributes; at a fault location, local dip may vary much more from sample to sample even if the (steered) similarity remains high, and this is exactly what the dipvariance attributes detect. Sometimes the NN fault cube also tends to pick up the acquisition footprint of a survey. A good extension to the NN fault cube default set is adding one or two of the curvature attributes, see [Curvature](#) for further explanation on curvature.



"Fault Cube" default attribute-set

Figure 4. Faults in seismic data.

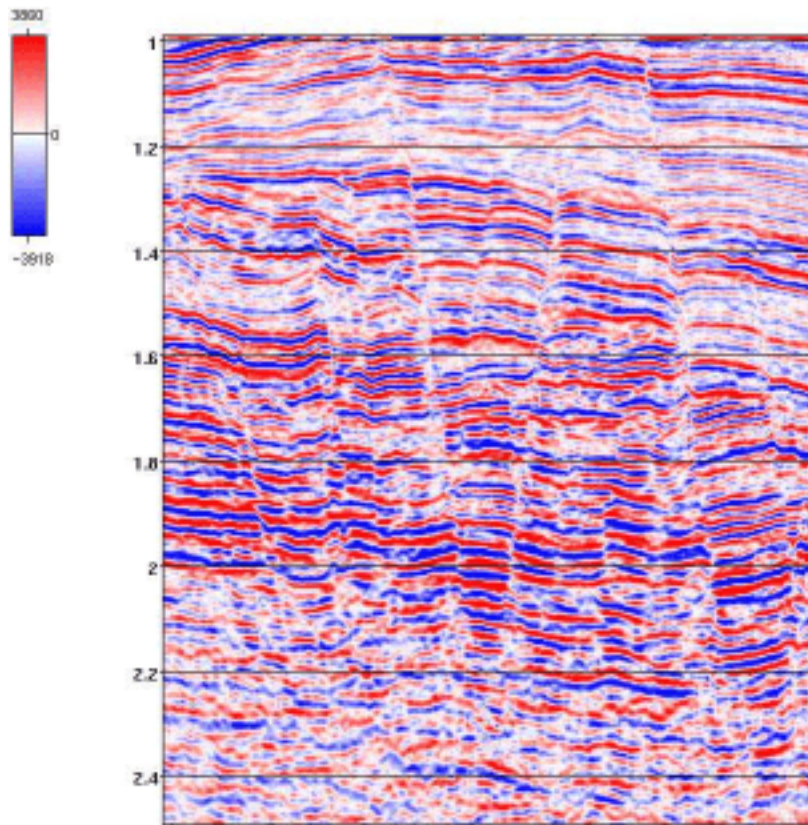
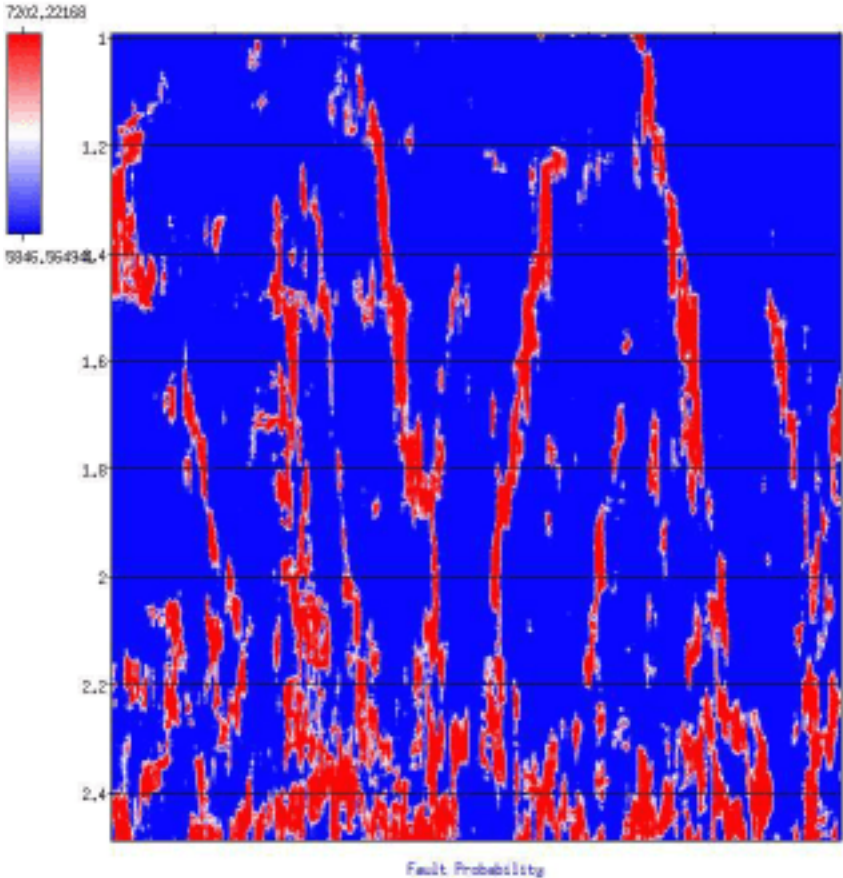
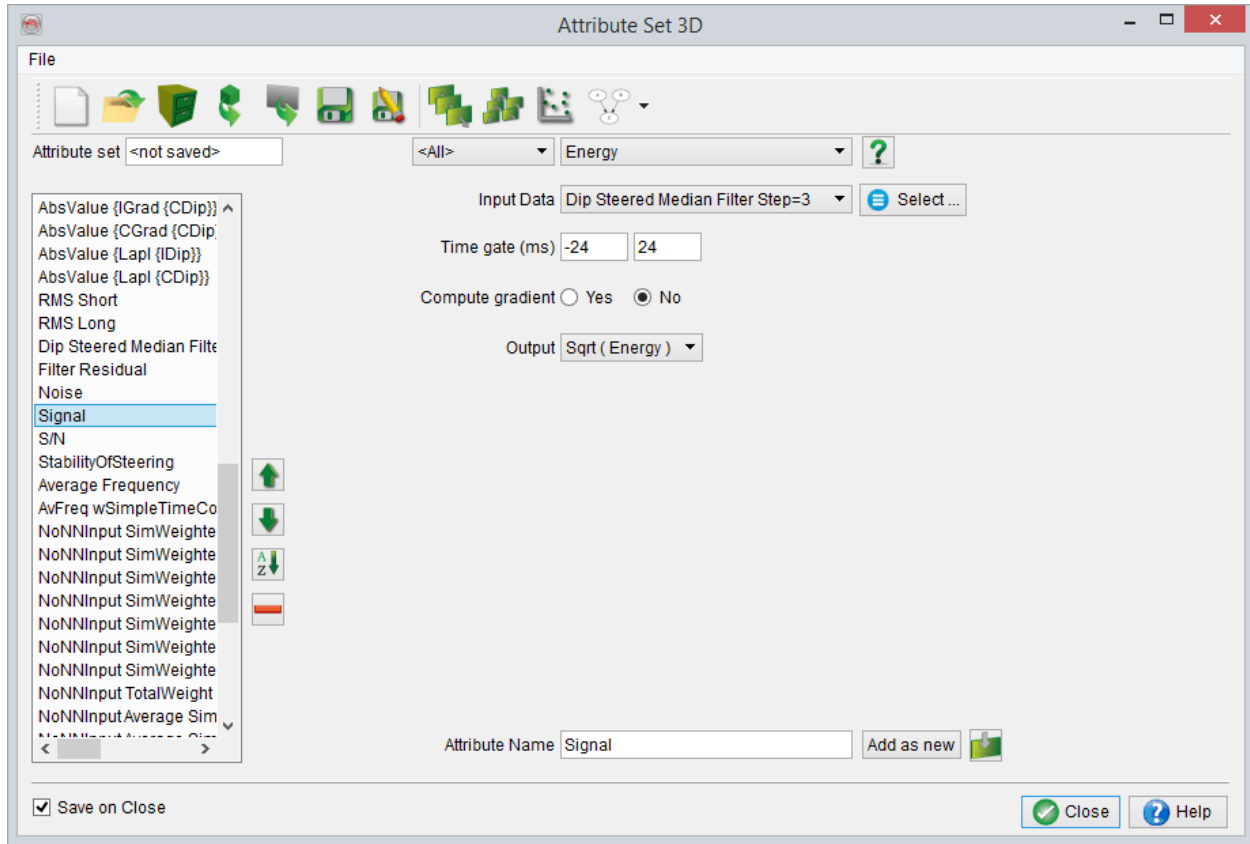


Figure 5. Result after applying a neural network with the NN Fault Cube as input. The output probability is re-scaled from 0-1 to 0-10000 and then clipped between 5850 and 7200.



12.5 NN Fault Cube Advanced

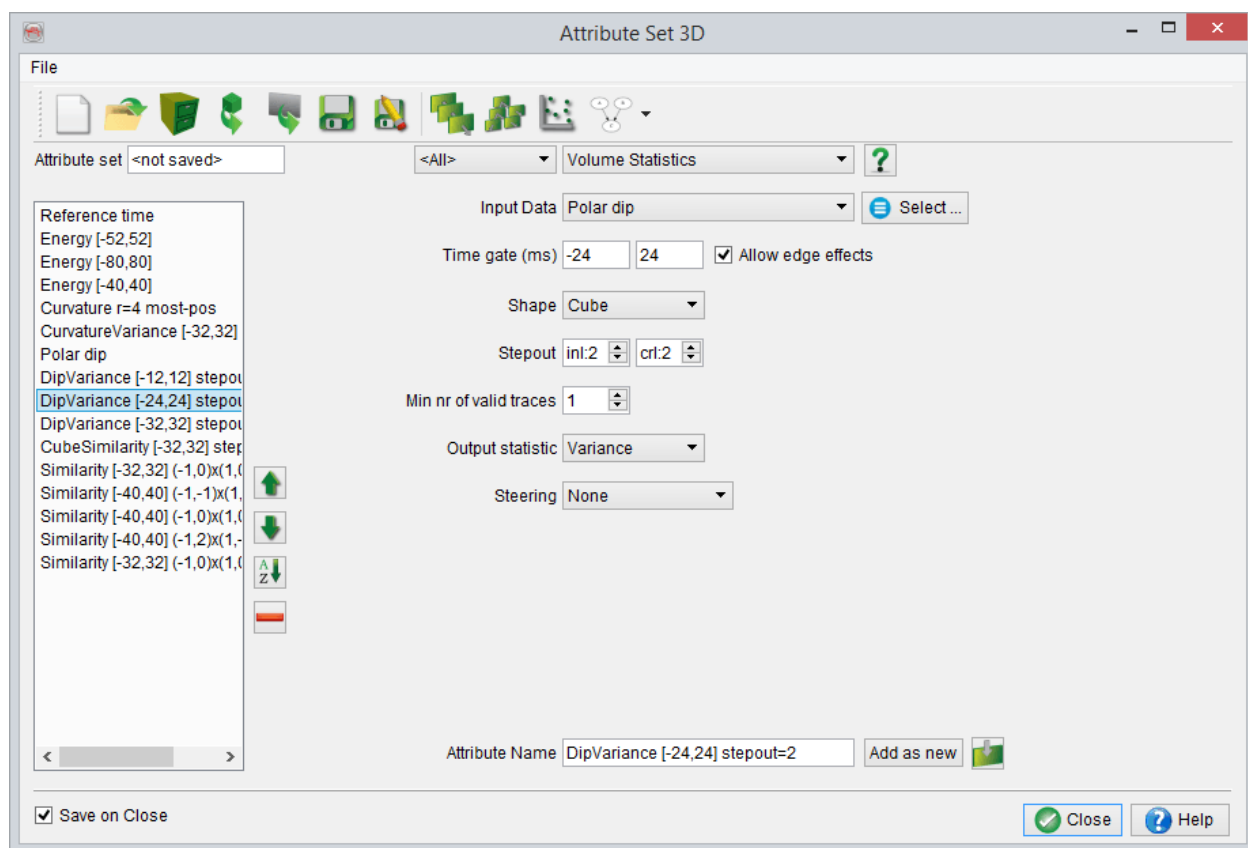
Using this default attribute-set, the user can create a Fault "probability" Cube for fault interpretation with more advanced parameters/attributes.



"Fault Cube advanced" default attribute-set

12.6 NN Salt Cube

This attribute set is meant for usage in a neural network. It is focused on detecting the generally chaotic, low-energy character of salt on seismic data. Because salt has many appearances, there are several possibilities for optimization. For example, one can focus on salt layers (horizontally oriented) or salt domes (vertically oriented). For layers, you can consider to decrease vertical window lengths while for domes the vertical window lengths can be increased. Adding frequency attributes can also be considered since most salt bodies exhibit a different frequency content as compared to their surroundings.



"SaltCube" default attribute-set

In general, the NN SaltCube tends to pick up both salt layers and salt diapirs. In the example below, the bottom of the salt is very well resolved.

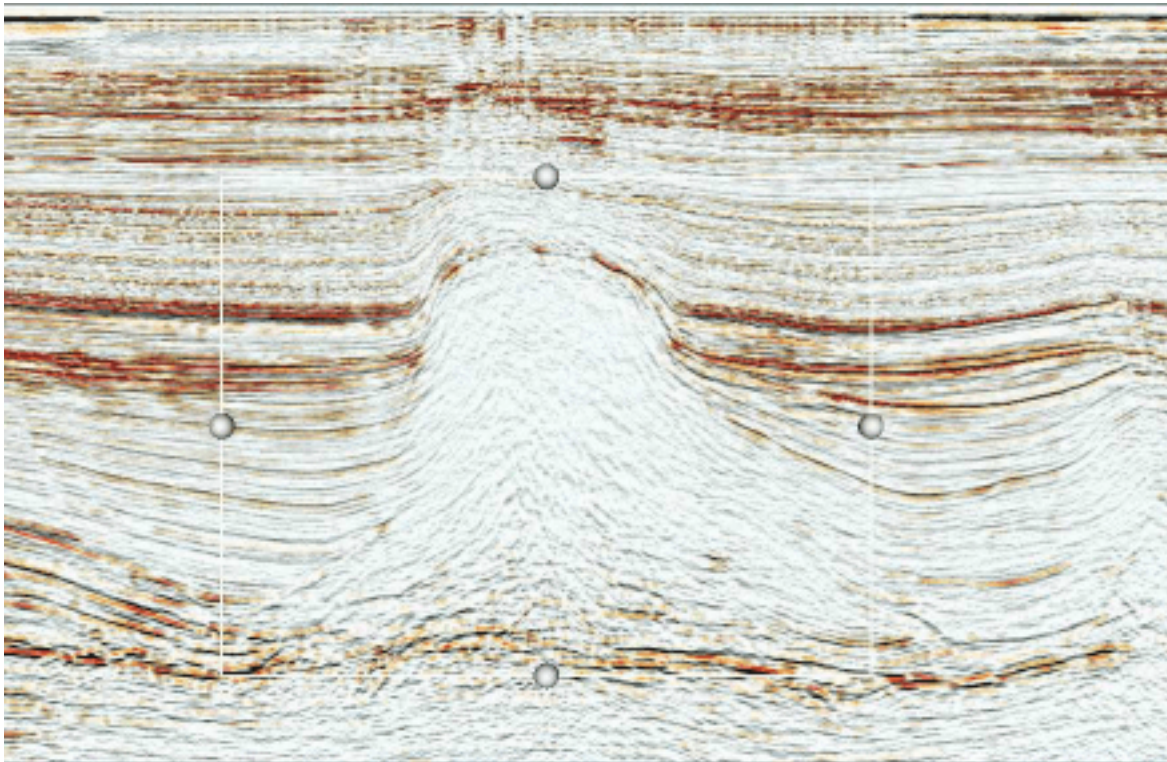


Figure 6. Salt in seismic data

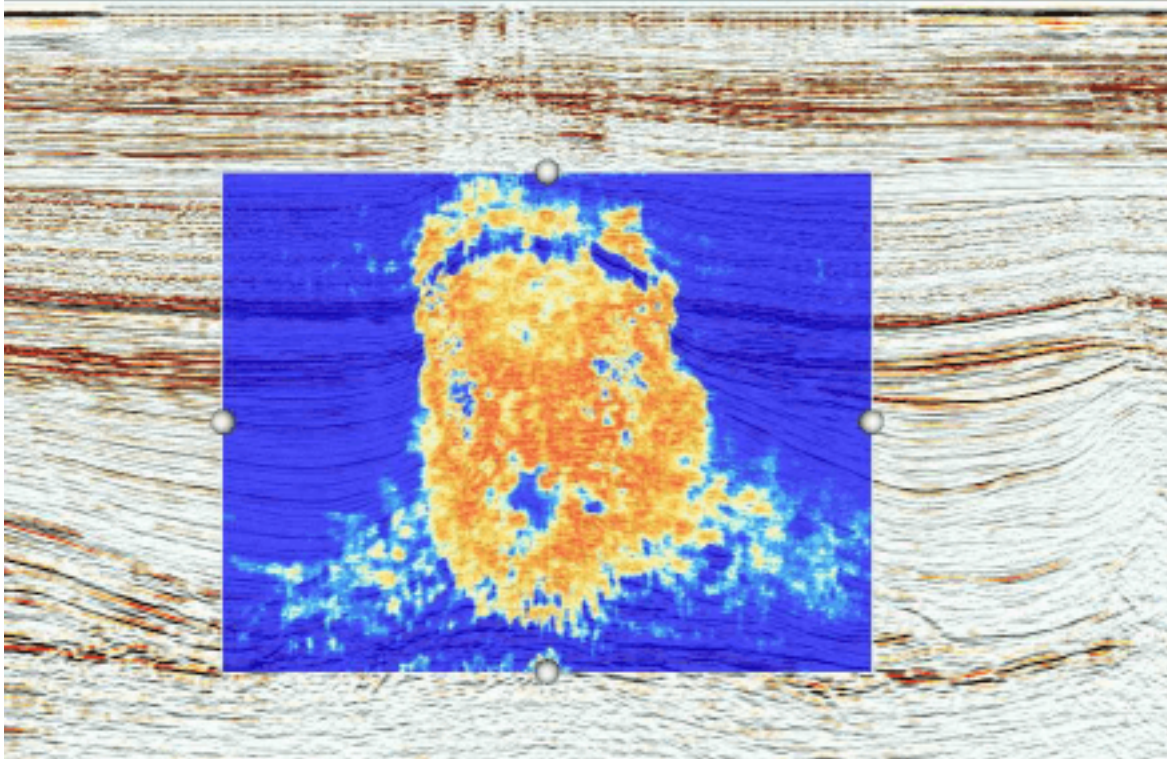
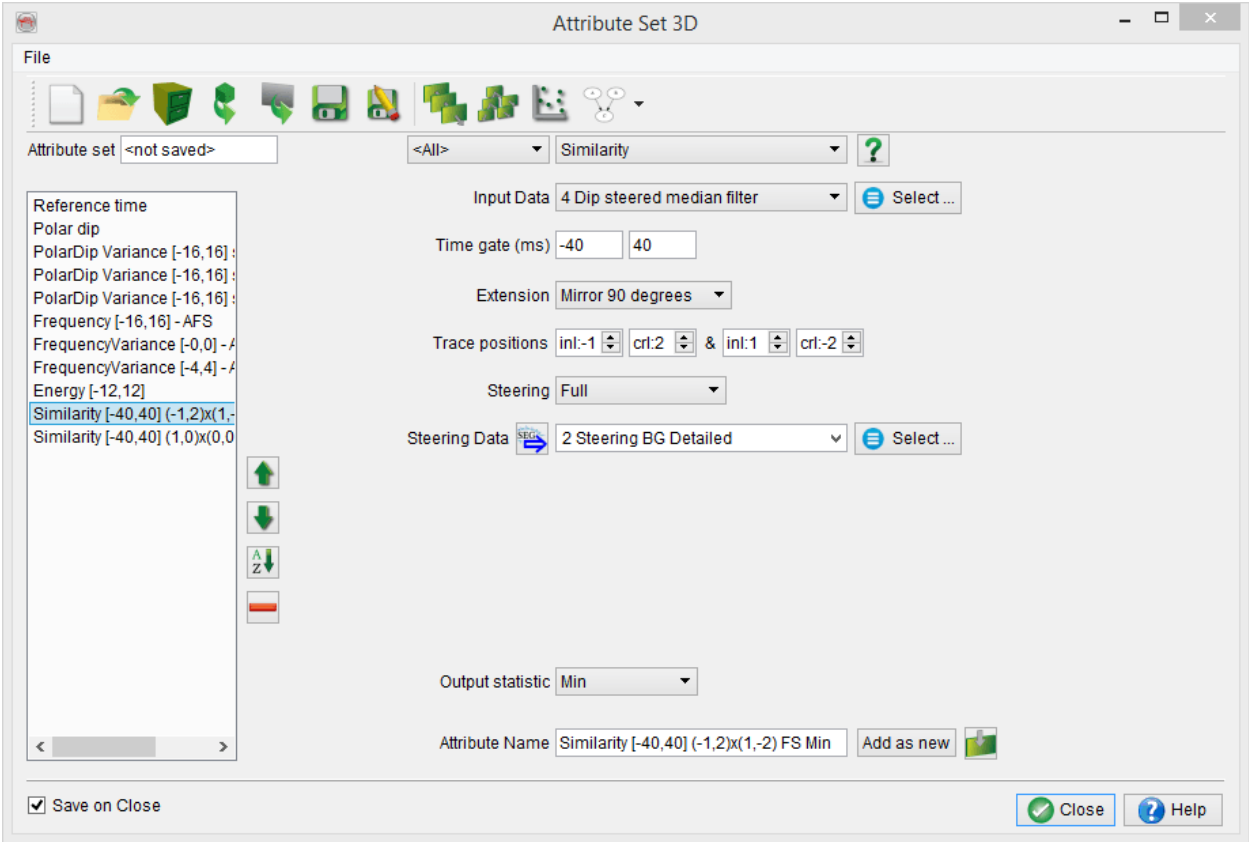


Figure 7. Result after applying a neural network with the NN SaltCube as input. The salt probability is displayed as overlay over the seismic data, with blue indicating low salt probability and yellow to red indicating a high salt probability.

12.7 NN Slump Cube

This attribute set was created by Andrew Wilson of *BG International* for quickly mapping turbidite slumps. Slumps are characterized by chaotic reflection patterns and frequency losses; therefore, tops and bottoms are difficult to map using conventional techniques. In Figure 8 a conventional view of the seismic data (left) and slump detection results (right) is shown, and in Figure 9 the slump detection results are presented as a 3D body.



"SlumpCube" default attribute-set

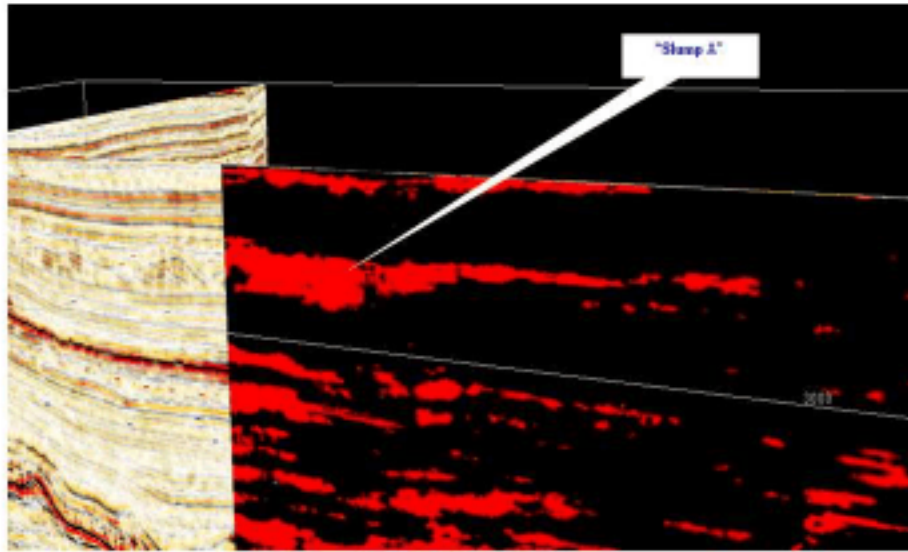


Figure 8: Left seismic data with slumps, on the right the result of the neural network with the NN Slump Cube attribute set as input. Courtesy Andrew Wilson BG.

Geobody for “slump A” shown in red extracted by voxel picking on scaled, smoothed slump classification volume (grey) in GeoViz.

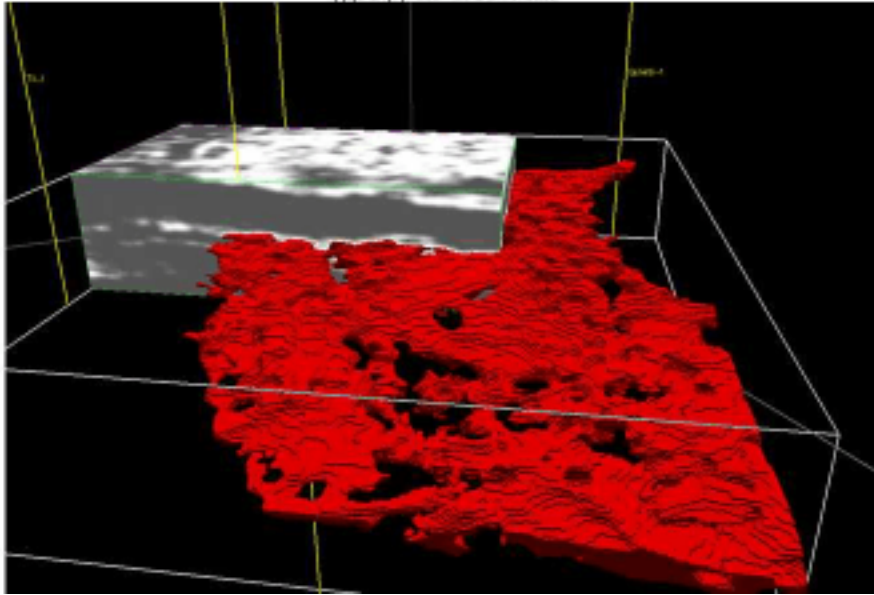
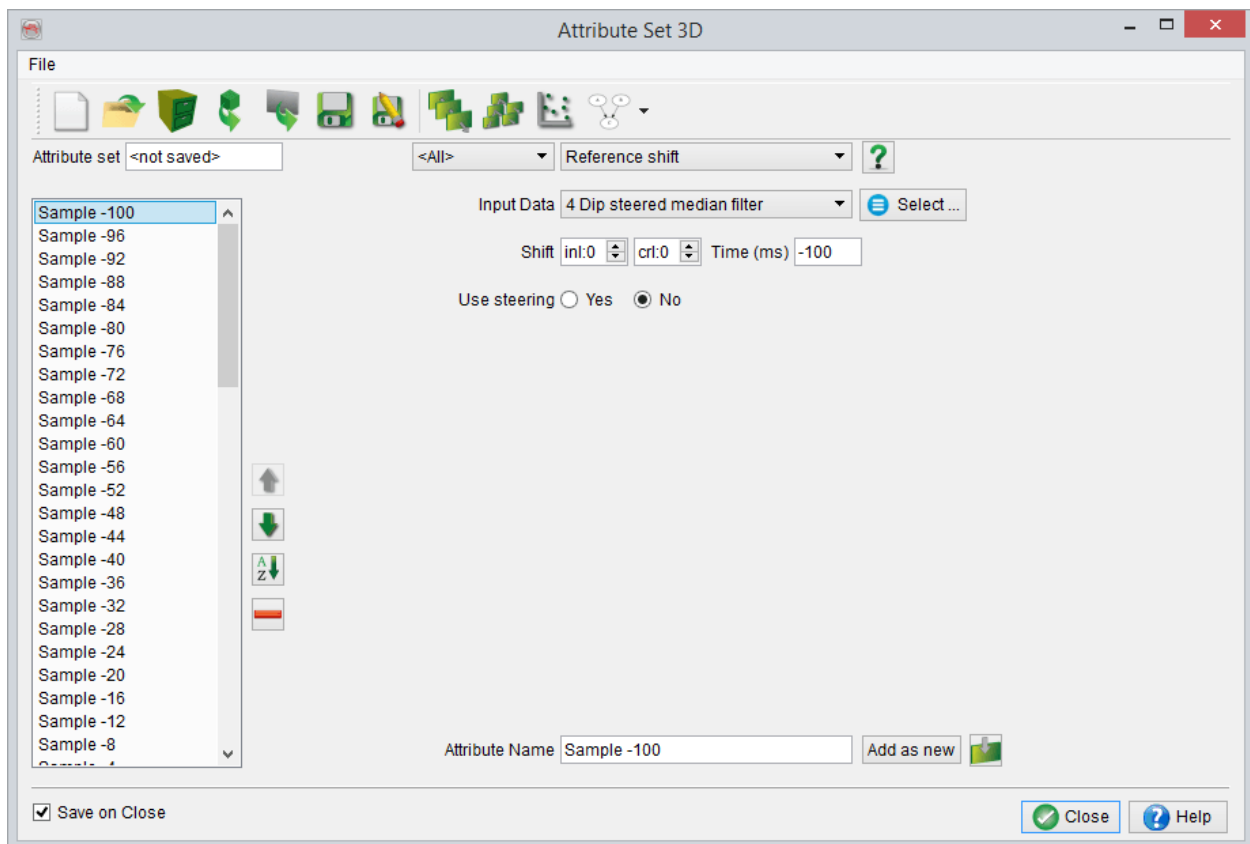


Figure 9: Result after applying a neural network with the NN Slump Cube as input. The slump volume is displayed as a three dimensional body. Courtesy Andrew Wilson BG.

12.8 Unsupervised Waveform Segmentation

This attribute set contains a number of samples from the seismic data volume above and below the sample position. The set of samples describes the seismic waveform, and can be used in horizon based unsupervised segmentations. The workflow is as follows: 1) create a set of random picks along the horizon (Pickset menu), 2) train a UVQ network on examples extracted at the random pick locations (use (part of) the waveform as input), and 3) apply the trained network to the horizon (horizon menu).



"UVQ" default attribute-set

A horizon may cover a number of geological (sedimentary) environments. Generally each geological environment will generate a specific seismic response. An unsupervised neural network learns to segment the different seismic responses into different classes. Operating in this way channels, sand bars, bright spots, and other geological bodies might be detected. Note that the default set should not be used for segmentation of volumes because the input changes dramatically when we modify the extraction time position. For segmentation of 3D bodies, you should use phase-independent attributes (e.g. energy, similarity etc). In Figure 8-10 an example of horizon based segmentation in 8 classes is shown.

The default set should be modified such that the sample rate of the attribute set corresponds with the sample rate of the data, and that the sampled window covers the seismic response of the level of interest. In the default attribute set the segmentation is based on the waveform; another approach would be to segment on basis of a number of attributes such as energy, frequency, etc.

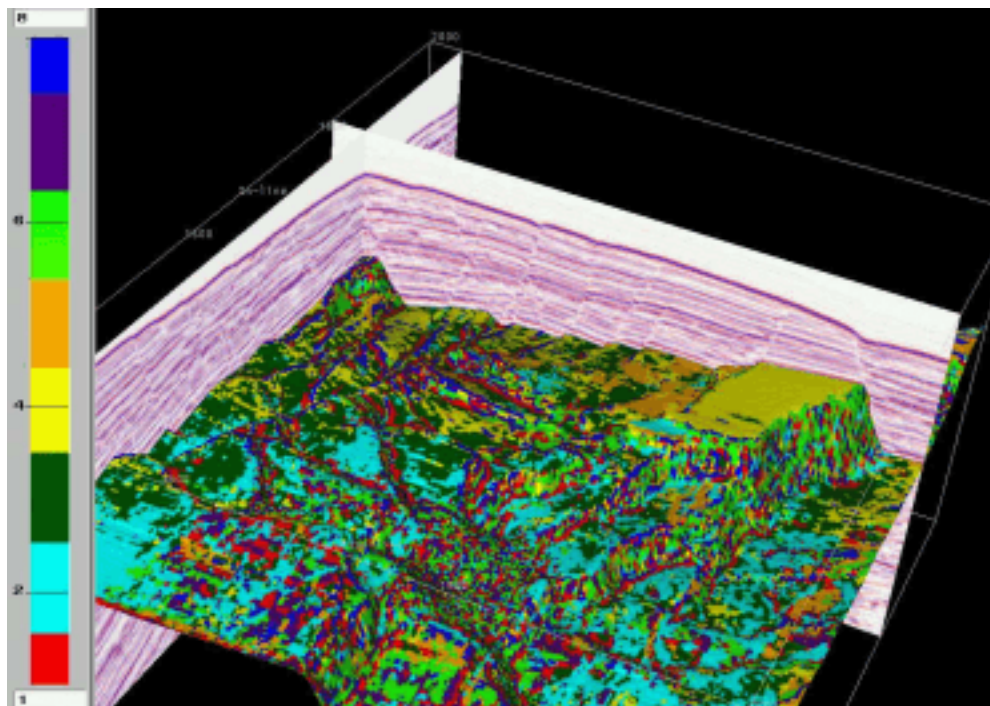


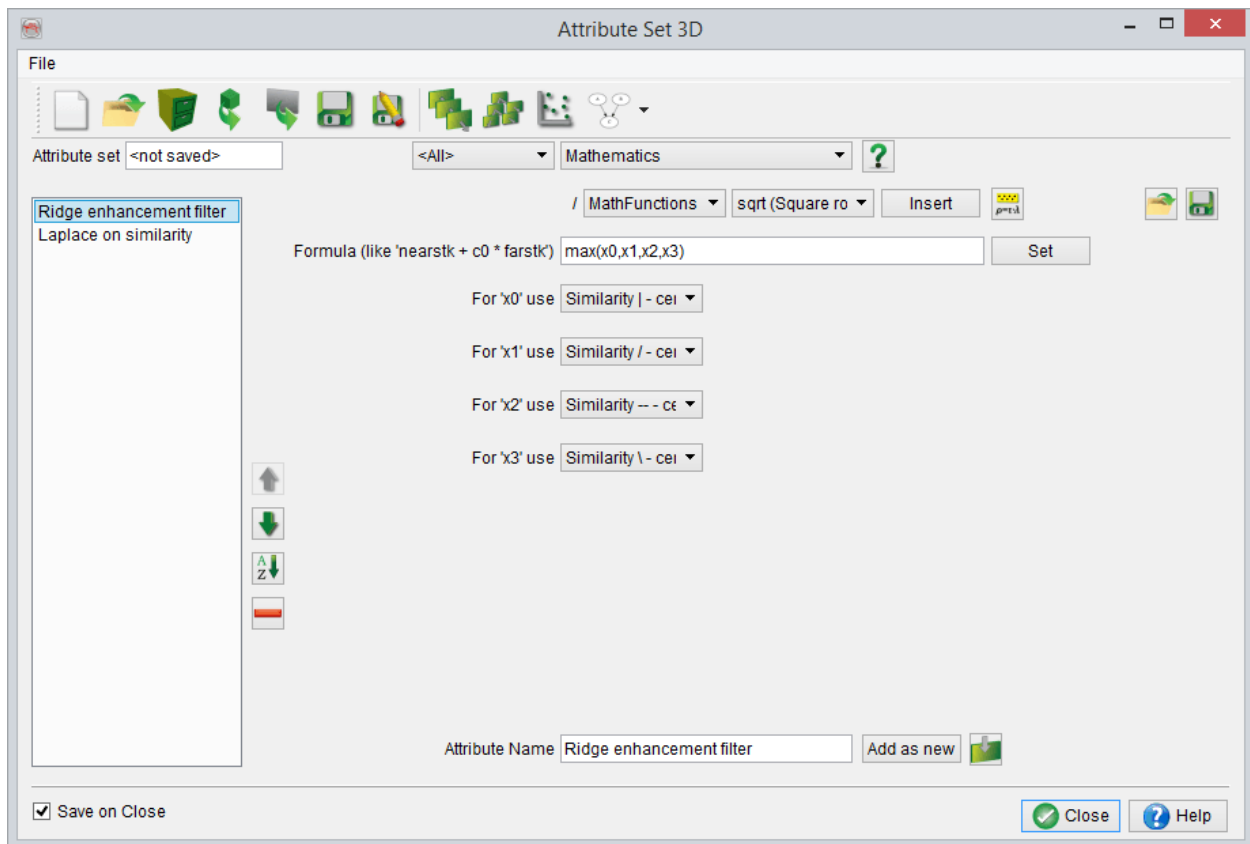
Figure 10. Horizon based unsupervised segmentation

To quickly create, apply and display an UVQ network, use this link: [Quick UVQ](#).

12.9 Ridge-enhancement Filter

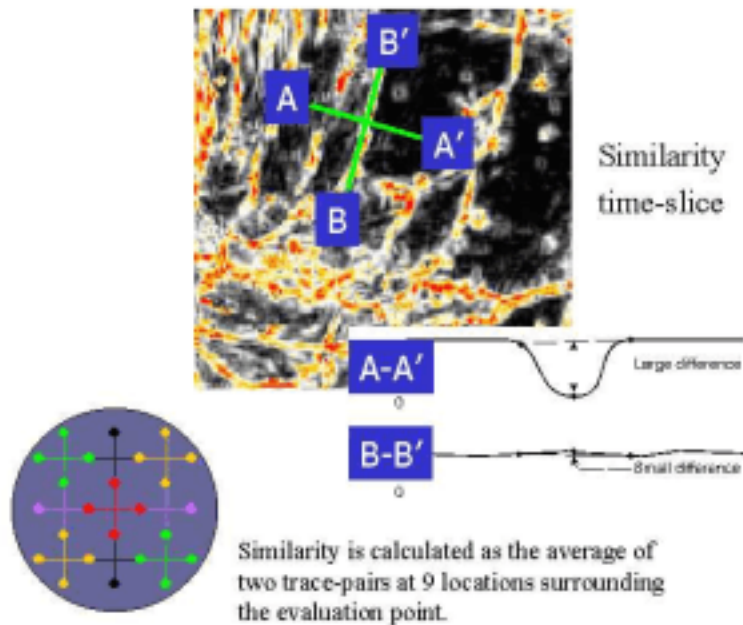
This filter sharpens ridges in a similarity cube.

The filter compares, in the time-slice domain, neighboring similarity values in four different directions, (inline direction, crossline, 45 degrees and 135 degrees), then outputs the largest ridge value. The ridge in each direction is the: $\text{sum}(\text{values on either side}) / 2 - \text{center value}$. In most evaluation points, there are no ridges and the values, thus, tend to be small; but when you cross a fault, there will be a large ridge perpendicular to the fault direction. The filter outputs the largest value, i.e. the ridge corresponding to the perpendicular direction.



"Ridge Enhancement Filter" Attribute-set

The output of this attribute set is the meta attribute *Ridge-enhancement attribute* at the bottom of the attribute set. All other attributes are intermediate attributes, used in the calculation of the final attribute. The construction of the attribute detects lateral lineaments in time slices of steered similarity, yet ignores bodies of low similarity. The idea behind the ridge-enhancement attribute is explained in figure below.

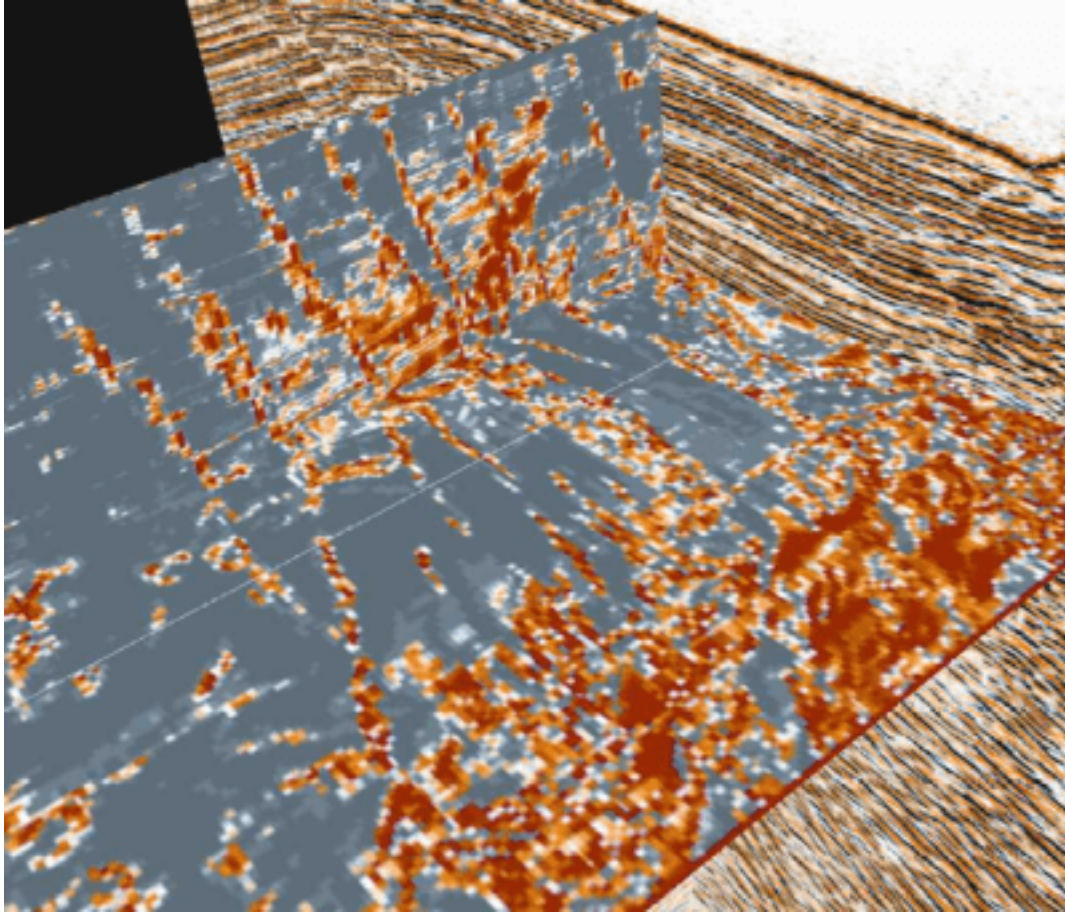


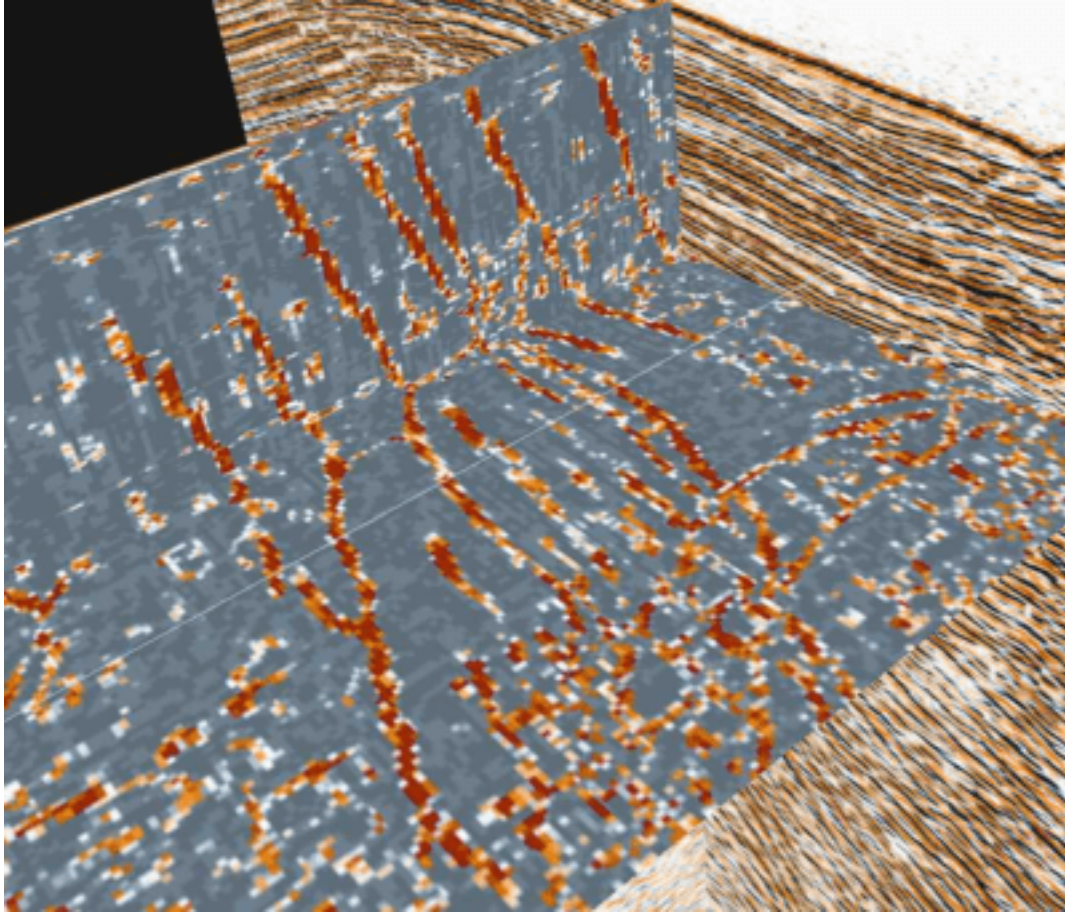
When we slice through a similarity cube and we cross a fault we observe a large difference in attribute response between the value at the fault position and the values on either side. In the ridge enhancement set we calculate 9 similarity attributes surrounding the evaluation point. We then scan in different directions to find the largest difference, which is the desired output. In following figures the output of the similarity attribute is compared with the output of the ridge-enhancement cube. The bodies of

high similarity have disappeared and the faults are also sharper. The users can optimize this attribute by fine tuning the parameters of the similarity attribute such that faults are optimally detected. For example parameters can be adjusted to the width and orientation of the faults, according to: wider faults or faulted zones must have longer windows and larger step-outs. More flat lying faults are better detected using smaller (vertical) windows.



You can decrease the processing time by almost a factor 9 if you store a similarity cube first and use this as input. Instead of calculating the similarity attribute 9 times on the fly, you calculate and store the similarity once and retrieve it. After you have stored your similarity cube can either trick the system by changing the first attribute (for instance by making it a Mathematics, Formula: $x0$ and as input your similarity cube) or by removing the first attribute and changing the attribute input of the following attributes from the removed attribute to your stored similarity cube.

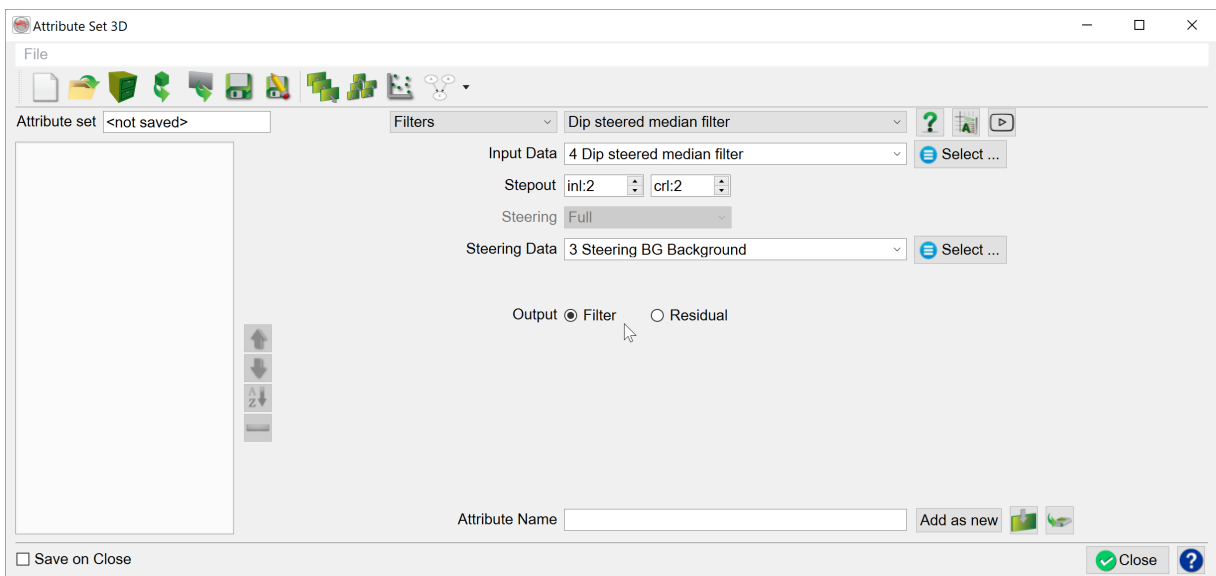




12.10 Dip-Steered Median Filter

The *Dip-Steered median filter* removes random noise and enhances laterally continuous seismic events by filtering along the structural dip. In median filtering, the center amplitude in a dip-steered circle is replaced by the median amplitude within the extraction circle. The effect is an edge-preserving smoothing of the seismic data.

Alternatively, the **residual** “difference between the original and dip steered median filter” seismic can be computed as output.

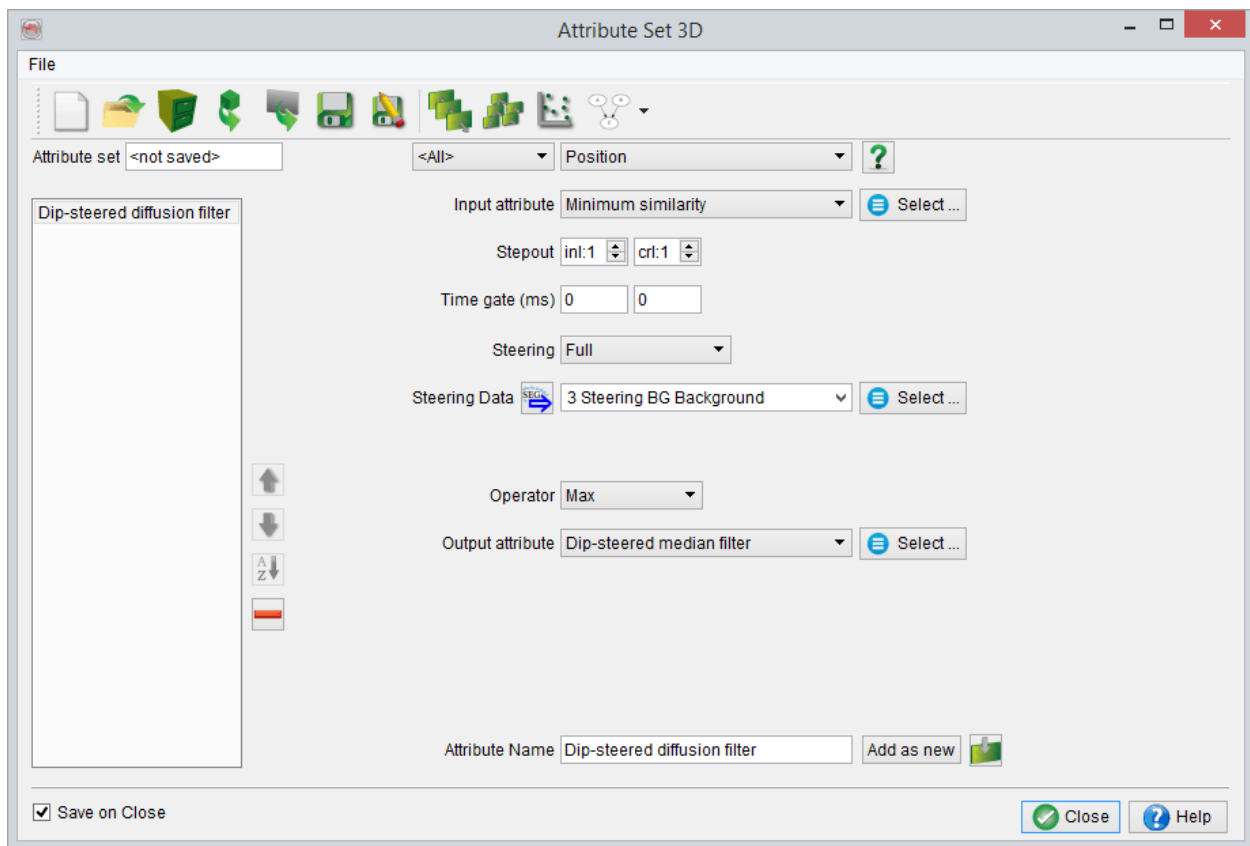


"Dip-Steered median filter" default attribute-set

12.11 Dip-Steered Diffusion Filter

A "*Dip-Steered Diffusion Filter*" is a default attribute set used to sharpen faults in Fault/Fracture analysis.

"Position" is the attribute to use; it is an important step for *Fault Enhancing Filtering*. In this case, the user takes the *Minimum Similarity* as Input attribute and as Output, for example, filtered data (using Max as Operator).

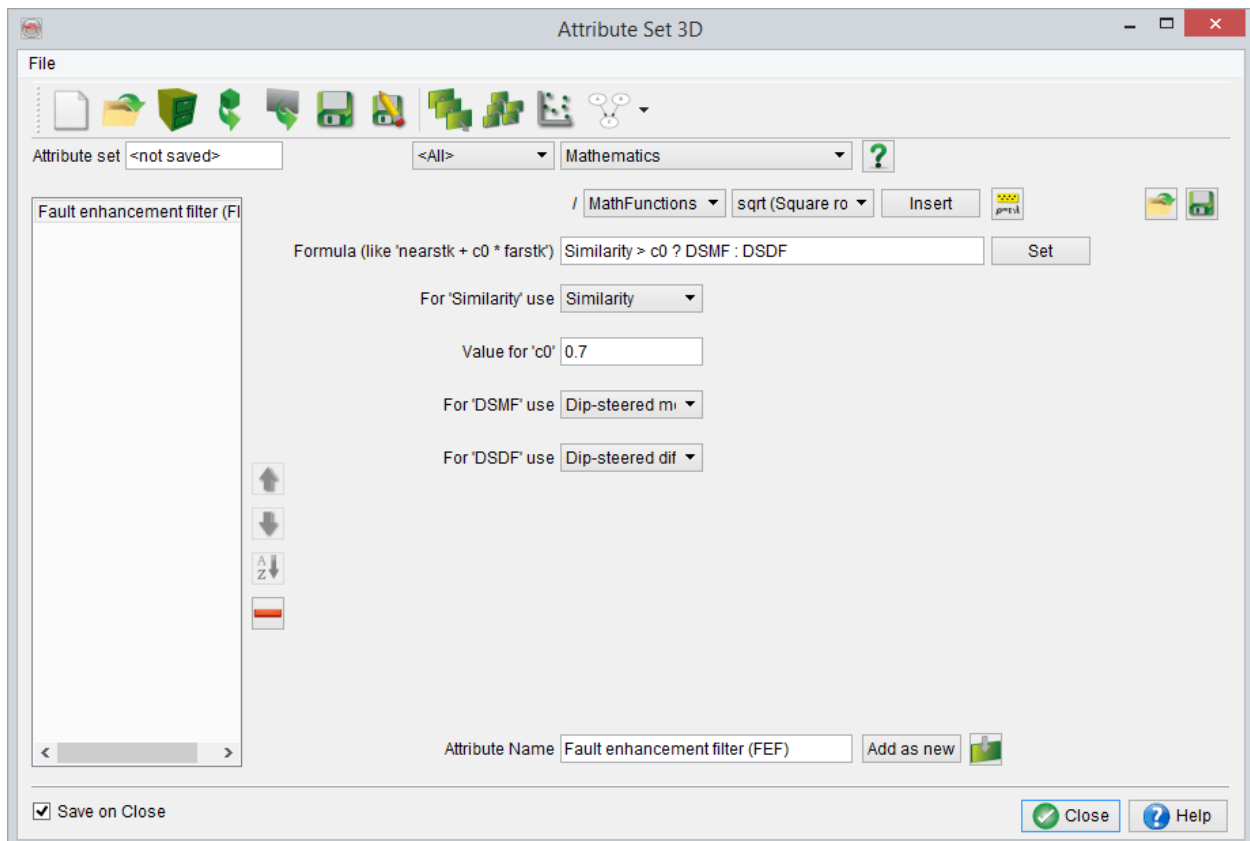


"Dip-Steered Diffusion Filter" default attribute-set

12.12 Fault Enhancement Filter

The "*Fault Enhancement Filter*" sharpens edges (faults) by means of median or diffusion filtering, along the structural dip.

In Fault enhancement filtering the quality of the seismic data in a dip-steered circle is evaluated. If the quality is good (Similarity is high) a dip-steered median filter can be applied. If the quality is low (near faults) a dip-steered diffusion filter should be used. The effect is smoothed seismic with sharp fault breaks.

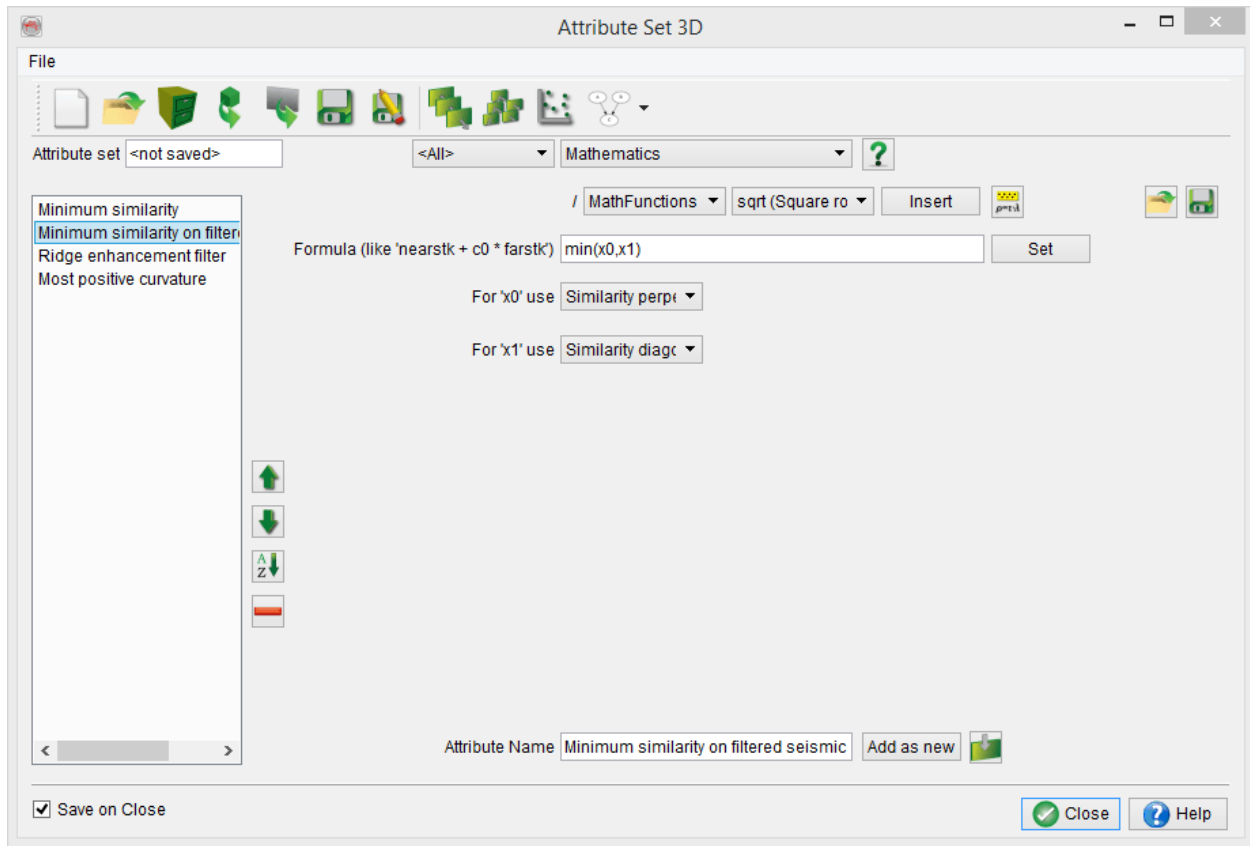


"Fault Enhancement Filter" default attribute set

For more detail information about workflows and tips look at: [dGB's How To Videos on YouTube.](#)

12.13 Fault Enhancement Attributes

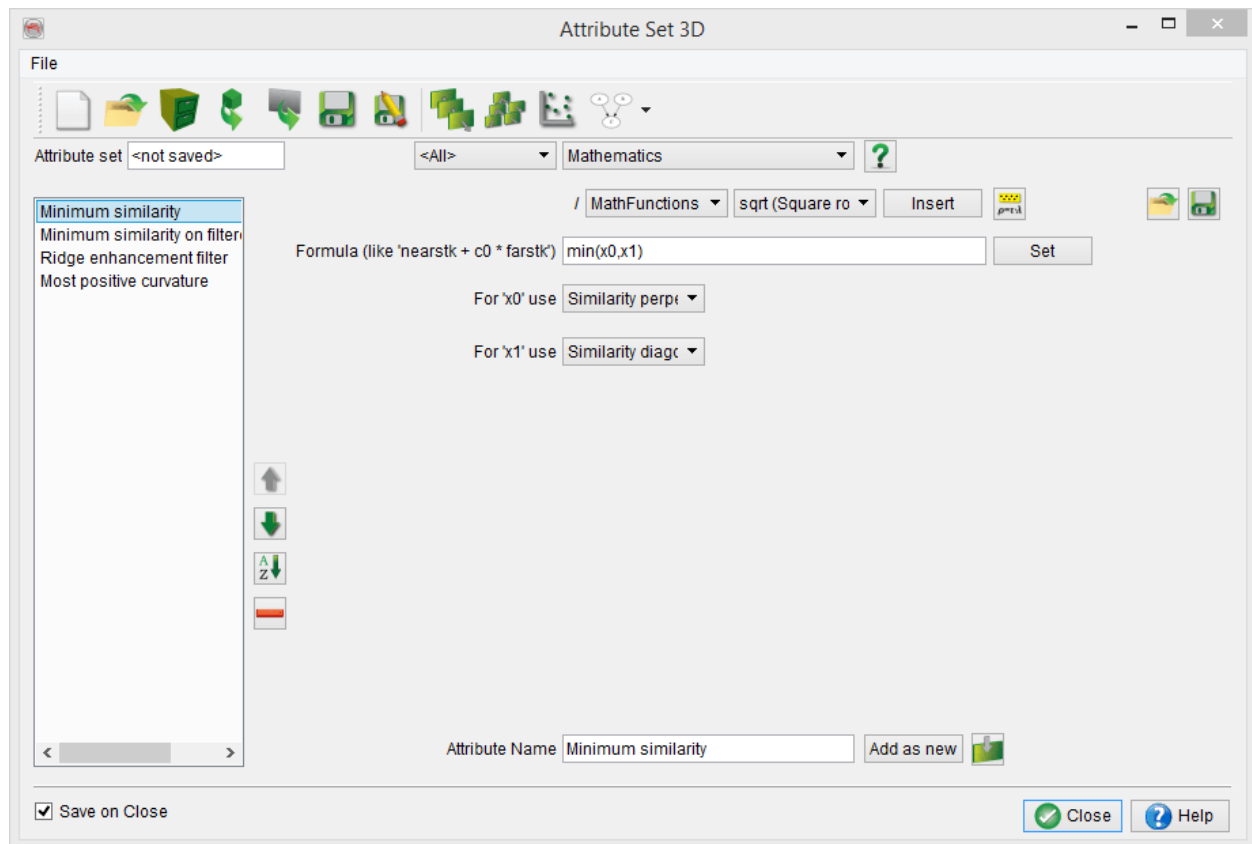
The "*Fault Enhancement Attributes*" default attribute set is a setup of four attributes for Fault/Fracture analysis.



"Fault Enhancement Attributes" default attribute set.

12.14 Fault Enhancement Filter (Expert)

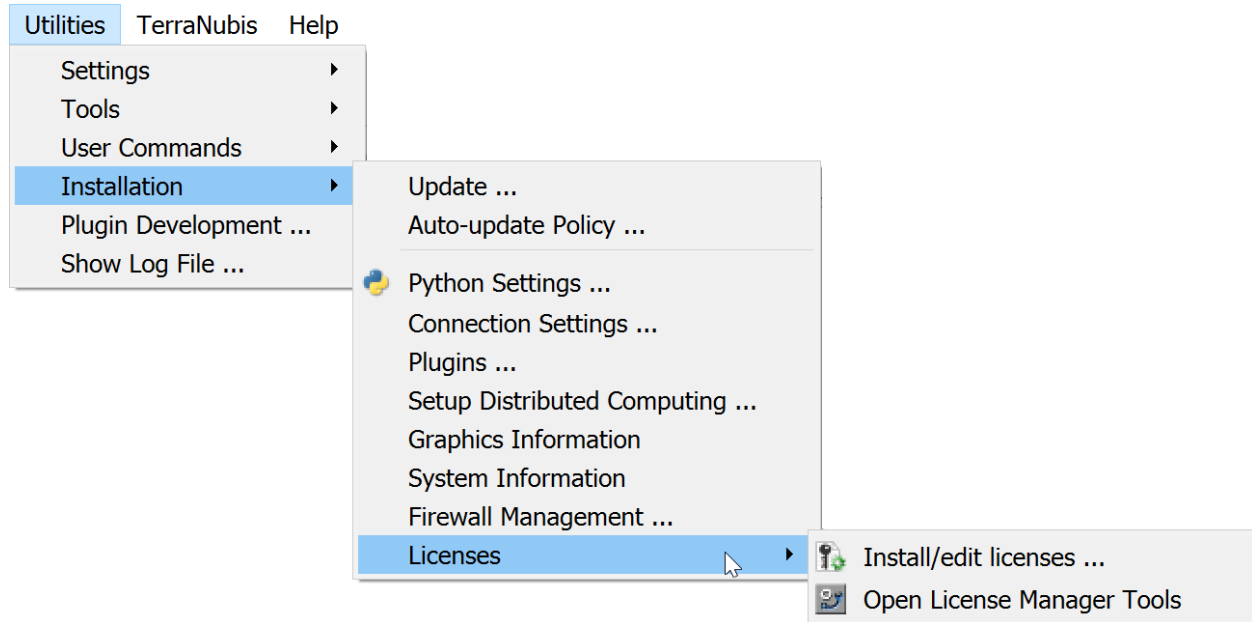
This is the same as the "Fault Enhancement Filter", but it enables the user to visualize and modify the parameters of the dip-steered median filter, dip-steered diffusion filter, and Fault Enhancement Filter; on the other hand, the "Fault Enhancement Filter" is a "ready to use" default attribute set.



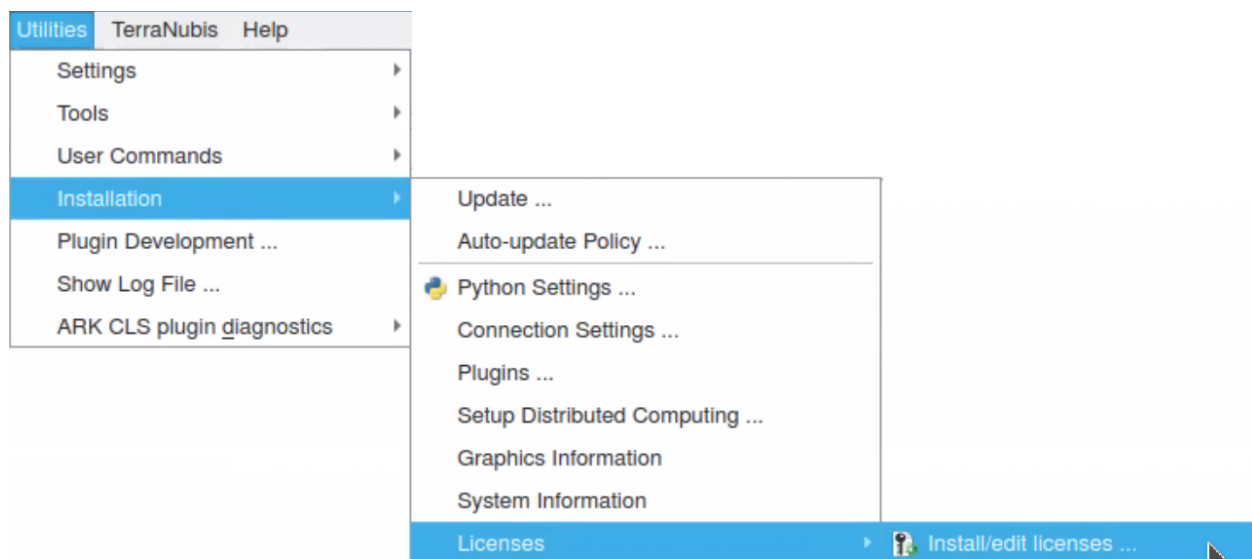
Seismic Filters Median-Diffusion-Fault-Enhancement" default attribute set.

13 Licenses

Under *Utilities > Installation > Licenses* you will see two options, differing per platform:



License options under Windows



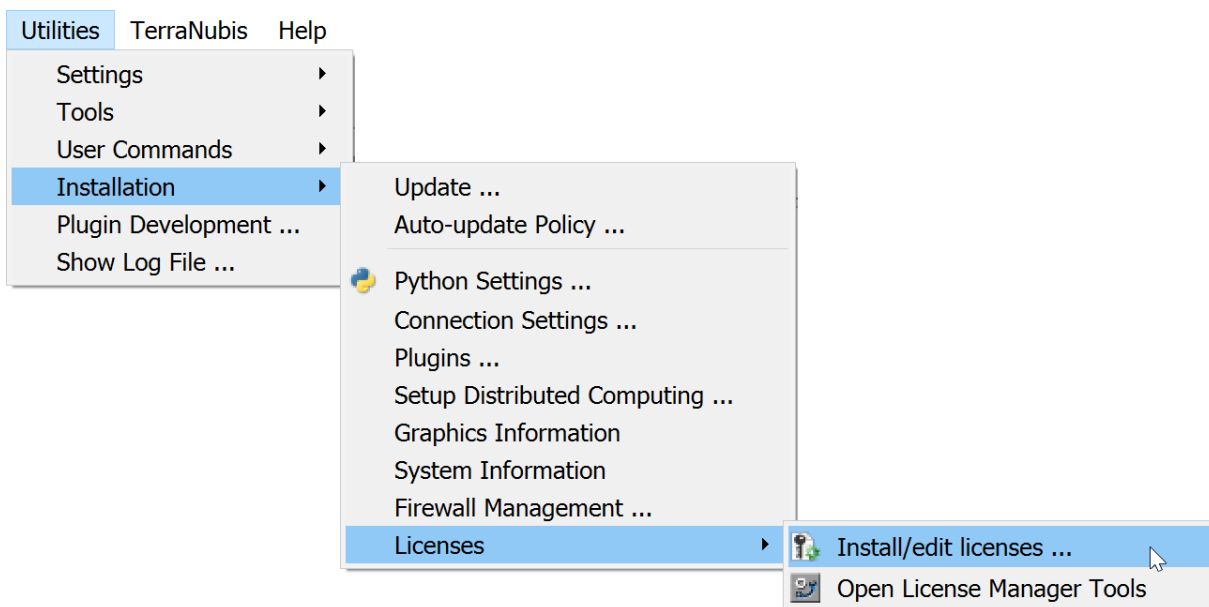
License options under Linux

These three options are explained in the following sub-sections:

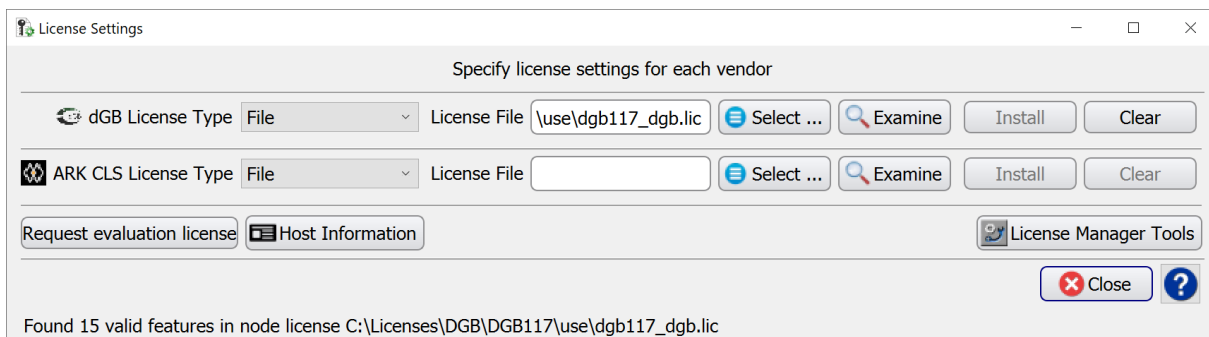
- For information about floating or server-based licenses, please refer to: [Flexnet Installation Guide](#)
- For more general information about OpendTect licensing options, please see: [Licenses](#)
- A more complete explanation of OpendTect license Installation can be found in the [Webinar](#), available on [OpendTect's YouTube Channel](#) and the [dgbes.com video resources page](#).

13.1 Install or Edit Licenses

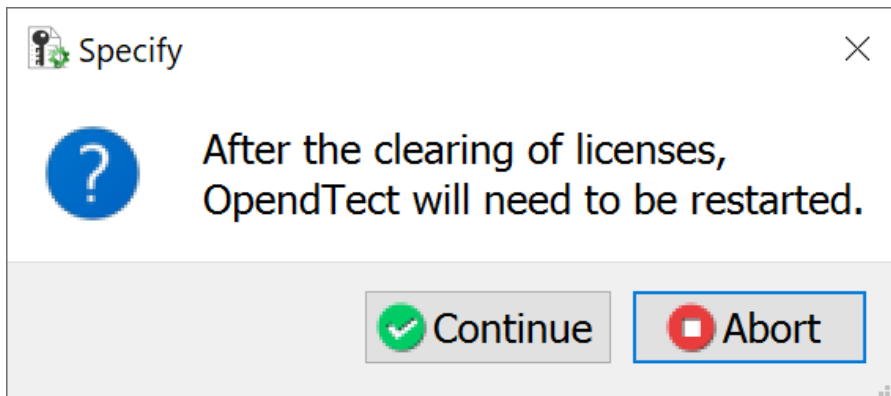
Plugins to OpenText can be run either by using a license server or node-locked licenses.



Use the following window to specify the type of license and the path to the license files:

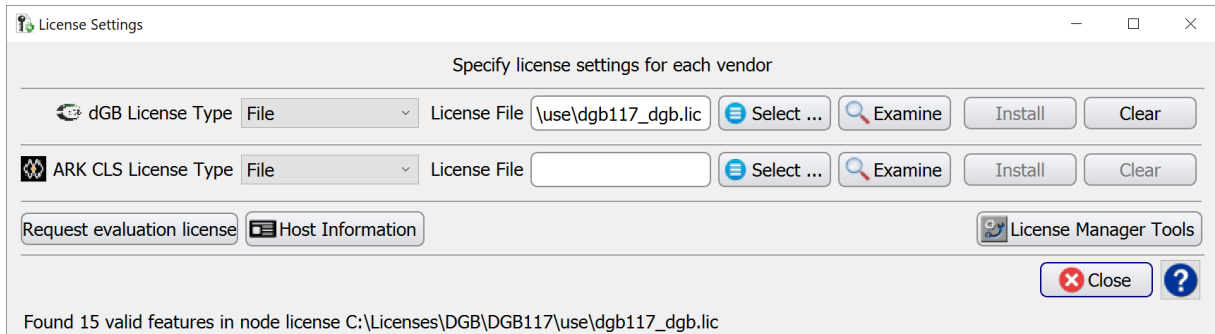


Here you can install each of the licenses by simply clicking 'Select', choosing the appropriate license file and clicking 'Ok' in the file selection window. Once you have selected all the licenses you are evaluating, click 'Ok'. Your installation will be confirmed and you will be prompted to re-start OpendTect:



For more detailed info, please refer to: [this UserDoc section](#) and this [AdminDoc section](#).

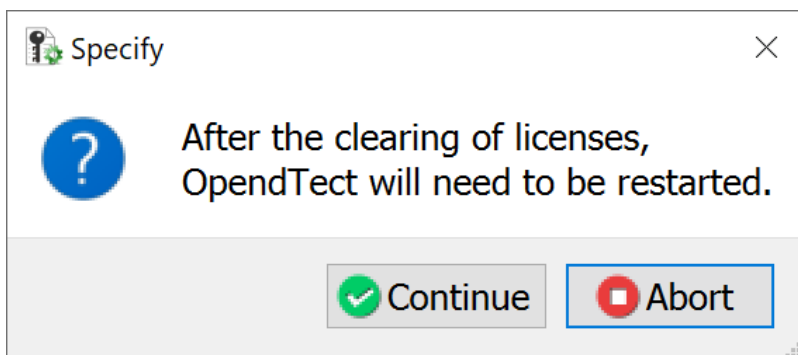
13.2 Clear License Installation



This option can be found via *Utilities -> Installation -> Licenses -> Install/Edit Licenses* and will clear:

- Node-locked licenses installed via any route.
- Floating (or 'server') licenses from the client machine (without stopping the license server).

Once cleared, you will be prompted to restart:

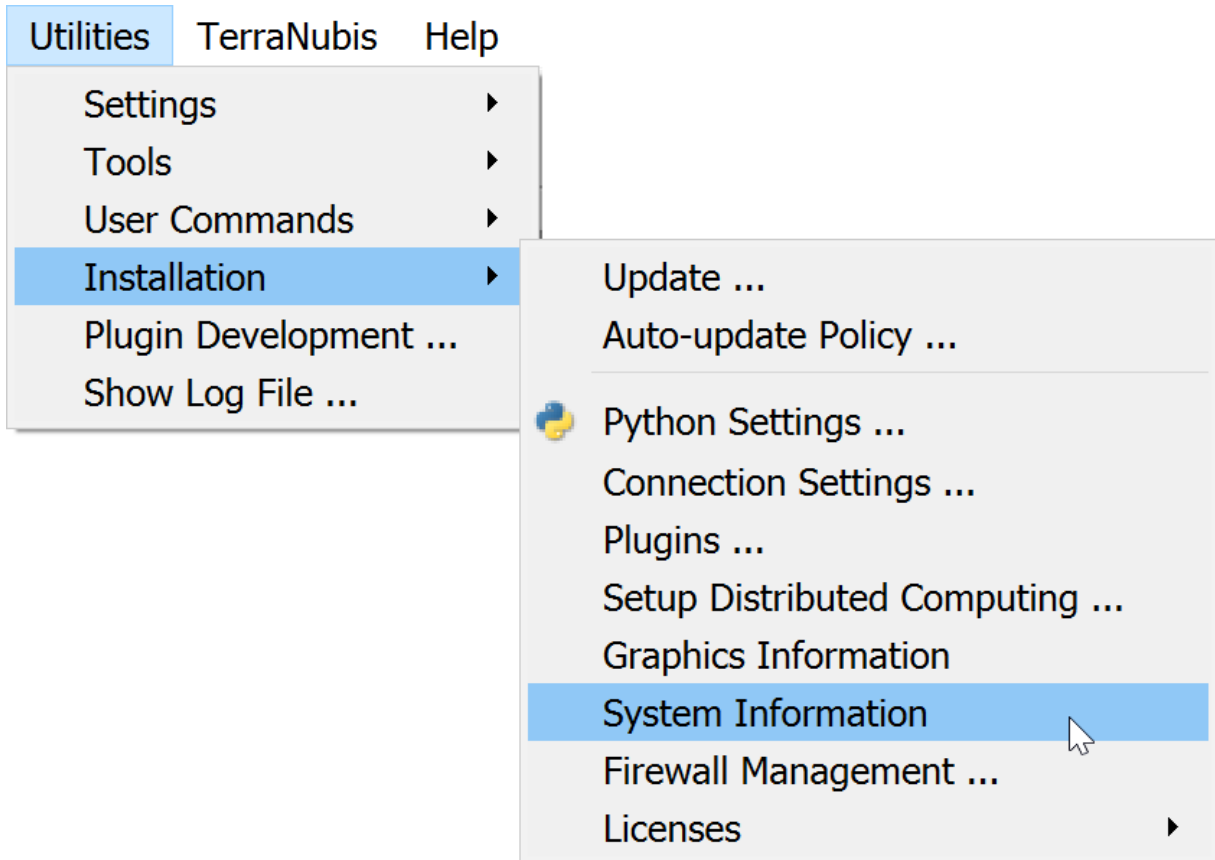


Users of Linux systems wishing to clear their license installation may choose to do the following:

- Locate the .flexlmrc file in your HOME directory (eg: \$HOME/.flexlmrc)
- Check in the file for specific lines referring to the OpendTect vendors (dGB, ARKCLS)
- If the file contains lines relevant to other software, then just delete the individual lines. Otherwise, you may choose to delete the file.

This method also applies to both demo/node-locked and floating licenses and will also not stop the server.

13.3 System Information



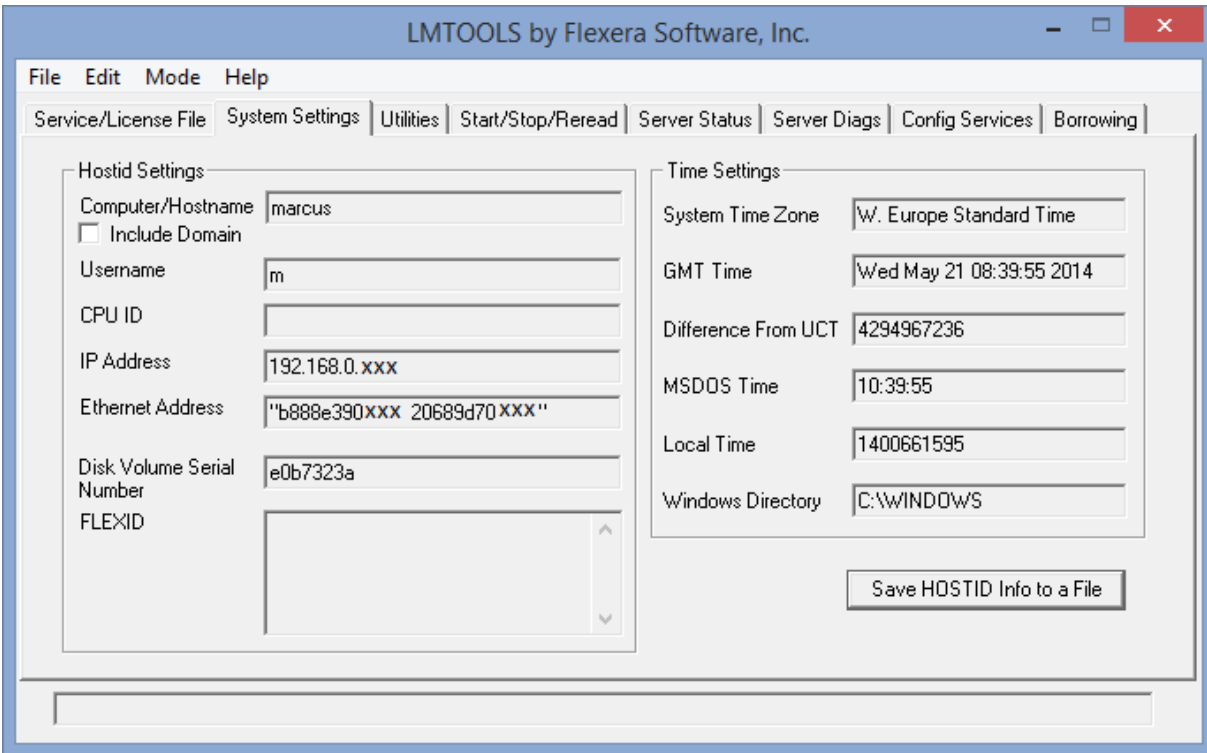
This option scans your system, and lists relevant System Information (HostID, Name, Operating System, User Name).

System Information

HostID(s)	abc123def456
Computer/Host Name	dgb456
<input type="checkbox"/> Computer/Host Name Overrule	
Computer/Host Address	192.111.1.1
Domain name	ad.domain.dgbes.com
Time Zone	(UTC+01:00) Amsterdam, Berlin, Bern, Ror
Operating System	Windows (64 bits)
OS Product name	Windows 10 Version 22H2
User name	mark
OpendTect Settings folder	C:\Users\mark\.od
OpendTect Installation folder	C:\OPENDT~1_v7\70B1B6~1.3_I\70FE6E~1

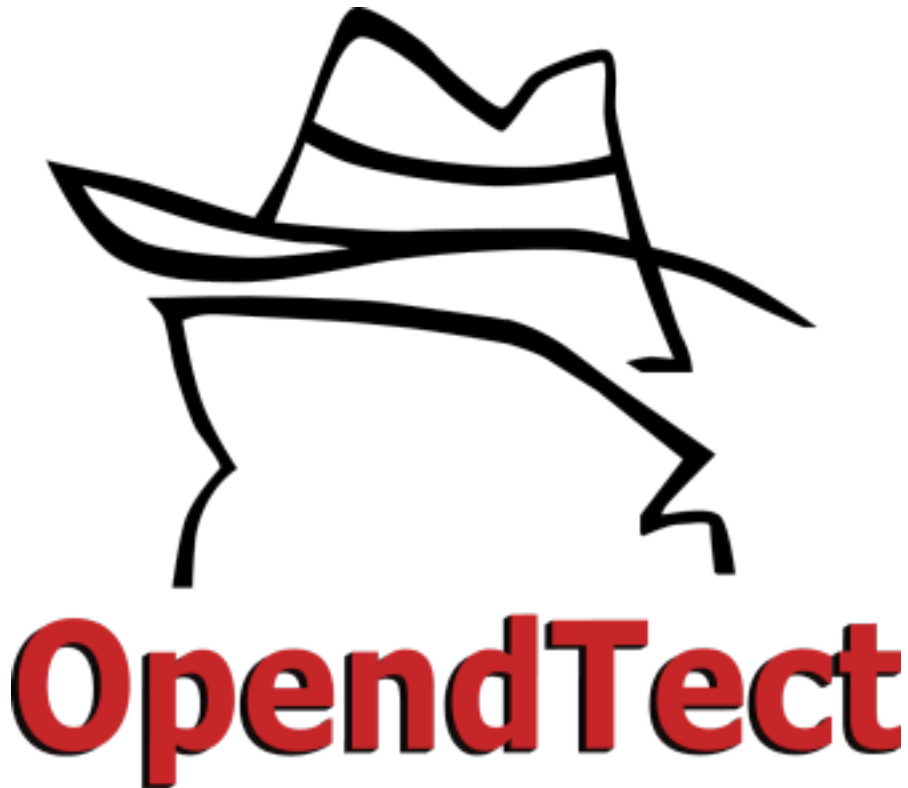
This information can be copied using the Copy to Clipboard button, convenient for contact with support when requesting licenses etc.

On Windows, accessing the HostID of the machine can be done via the LM Tools (available via the Start Menu or directly from ..\OpendTect\7.0.0\bin\win64\lm.dGB\lmttools.exe):



The option 'Save HOSTID Info to a file' will simply save the information displayed above into a .txt file for reference.

14 References




- Aminzadeh, F., de Groot, P., Berge, T. and Valentini, G., 2001. Using Gas Chimneys as an exploration tool. Part I - concepts and procedures, Part II - examples. World Oil, May 2001 and June 2001.
- de Groot, P.F.M., Ligtenberg, H., Heggland, R. and Meldahl, P., 2001. Selecting and combining attributes to enhance the detection of seismic objects. 63d. EAGE conference, Amsterdam. 11-15 June 2001.
- Heggland, R., Meldahl, P., Groot, P. and Bril, B., 2000. Detection of seismic chimneys by neural networks - A new prospect evaluation tool. 62nd EAGE conference, 29 May - 2 June 2000. Glasgow.

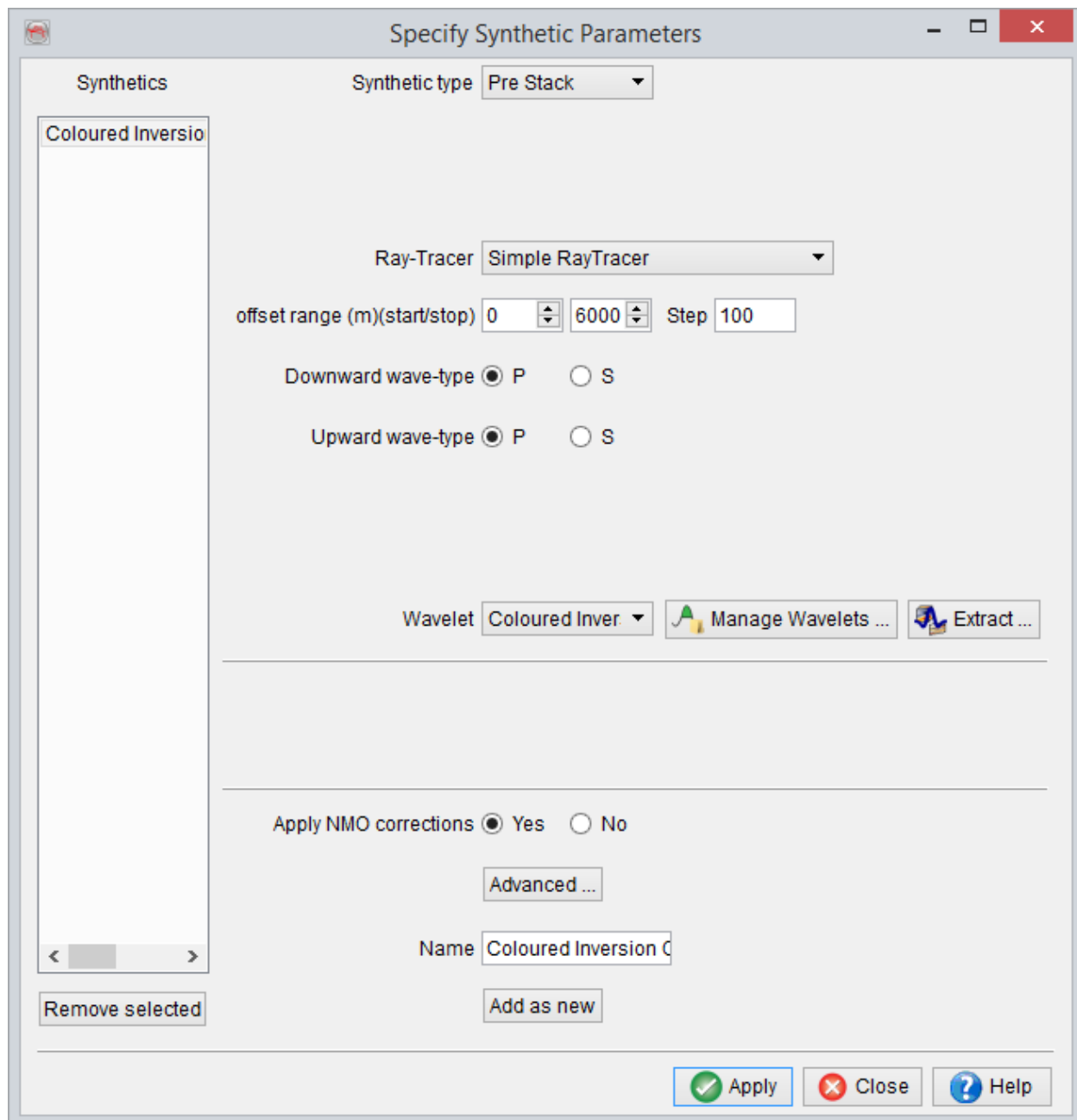
- Heggland, R., Meldahl, P., de Groot, P. and Aminzadeh, F., 2000. Seismic chimney interpretation examples from the North Sea and the Gulf of Mexico. The American Oil & Gas Reporter, Feb. 2000.
- Heggland, R., Meldahl, P., Bril, B, and de Groot, P., 1999. The chimney cube, an example of semi-automated detection of seismic objects by directive attributes neural networks: Part II; Interpretation. 69th SEG conference, Oct. 31 - Nov . 5, 1999. Houston.
- Marfurt, K.J., and R. L. Kirlin, 2000, 3D Broadband estimates of reflector dip and amplitude: Geophysics, 65, 304-320.
- Meldahl, P., Heggland, R., Bril, B, and de Groot, P., 2001. An iterative method for identifying seismic objects by their texture, orientation and size. SEG international Exposition and 71st. Annual Meeting. San Antonio, Texas, USA. 9-14 Sep. 2001.
- Meldahl, P., Heggland, R., Bril, B, and de Groot, P., 2000. Semi-automated detection of seismic objects by directive attributes and neural networks, method and application. 62nd EAGE conference, 29 May - 2 June 2000, Glasgow.
- Meldahl, P., Heggland, R., Bril, B, and de Groot, P., 1999. The chimney cube, an example of semi-automated detection of seismic objects by directive attributes neural networks: Part I; Methodology. 69th SEG conference, Oct. 31 - Nov. 5, 1999. Houston.
- Roberts, Andy, 2001, Curvature attributes and their application to 3D interpreted horizons, First Break, 19(2), 85-100, First Break, 19(2), 85-100.

Appendix A - Synthetic Data Generation

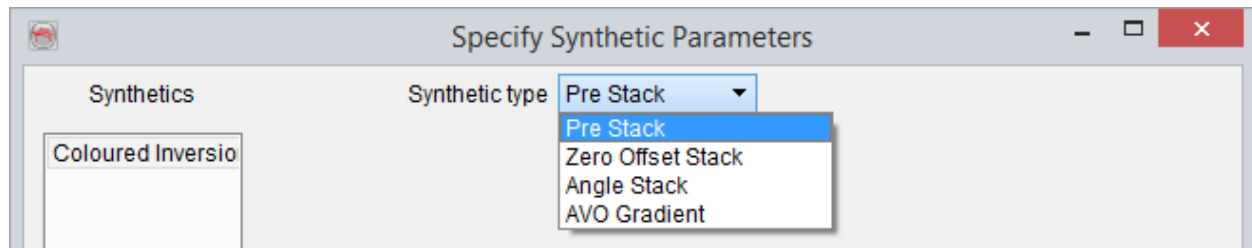
Options

- Ray Tracing
- Computation of the Zero Offset Reflection Coefficient
- Computation of the Reflection Coefficient at any non-zero offset
- Elastic Model

Synthetic seismic data is generated in *SynthRock* by clicking on the edit icon () in the top-left corner of the main Layer Modeling Interface. This will bring up the following window:



Here various types of synthetic data can be generated: Zero Offset Stack, Pre Stack gathers, Angle Stack and AVO Gradient:



Ray tracing

While Zoeppritz equations are used to compute angle-dependent reflectivity; ray tracing is required to compute the angle of incidence, at various interfaces of elastic model, of seismic rays recorded at various offsets. The 2D ray-tracing can be performed in two ways:

Simple Ray Tracer:

The ray is going directly from the source to the depth of the target layer, and up to the receiver in the same way. This does not account for ray bending, or velocity inversions. Here the user has to specify the offset range and the step for creating pre stack gathers; they could in theory be same as defined in acquisition/processing of the seismic data. It can model both Downgoing and Upgoing P-waves and S-waves. Now, ray tracer and Zoeppritz equations have produced angle-dependent or offset-dependent reflectivity traces, which can be convolved with user defined wavelet to produce pre stack gathers. It may be noted that in SynthRock, the conversion from offset domain to angle domain and vice-versa is done using the V_p of the Elastic Model [[hyperlink with Elastic Model](#)] (which is essentially the upscaled and time converted V_p log of pseudo-wells).

Ray-Tracer Simple RayTracer

offset range (m)(start/stop) Step

Downward wave-type P S

Upward wave-type P S

Simple RayTracer parameters

Advanced Ray Tracer (not in the GPL version):

It works in a more sophisticated way than the simple ray tracer. It honours the ray bending according to Snell's law and thus velocity inversions as well. To reduce the processing time, the Elastic Model [hyperlink with Elastic Model] layers may be blocked: Consecutive layers with similar V_p , V_s and Density values are concatenated together, as defined by the threshold. For example the default threshold is 1%, which means if there is less than 1% difference in the elastic model values of two layers, they will be blocked. The ray is propagated in a straight line inside a concatenated layer.

It is also possible to compute internal multiples in the advanced ray tracer. Furthermore, incorporation of spherical divergence, is also possible, by defining the spreading geometry as either "Distance" or "Distance*Vint" .

Ray-Tracer Advanced RayTracer

offset range (m)(start/stop) Step

Downward wave-type P S

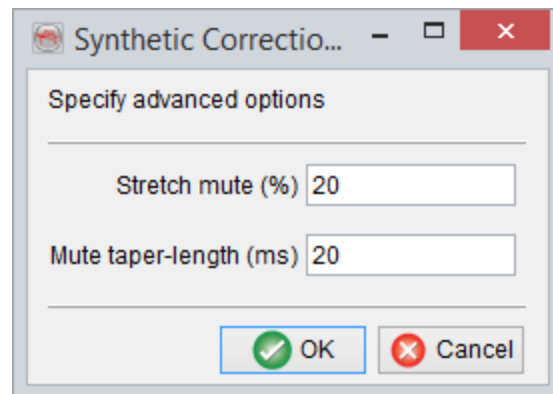
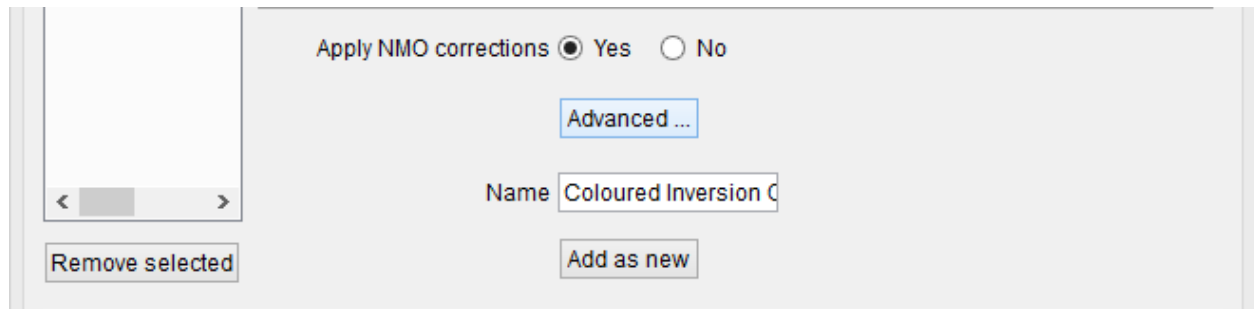
Upward wave-type P S

Blocking threshold %

Select spreading geometry None

Advanced RayTracer parameters

Afterward, NMO corrections can also be applied to create NMO corrected synthetic gathers. Here in the Advanced options, one can specify the % stretch mute typically applicable at far offsets. If the length of a full seismic waveform increases by more than the mute %, it will get muted. Moreover, the taper-length of the muting function, can be defined under this advanced options menu of NMO corrections:



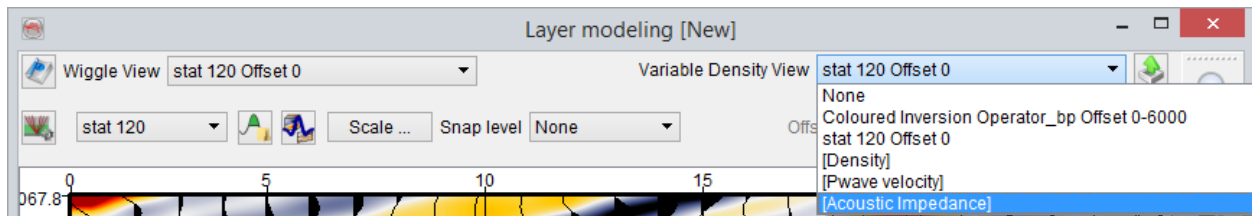
Advanced RayTracer: Advanced corrections options

Computation of the Zero Offset Reflection Coefficient For the simplest Zero Offset Stack, calculation of reflection coefficient at any interface is done using the simple formula:

$$R = \frac{(Z_1 - Z_0)^2}{(Z_1 + Z_0)^2}$$

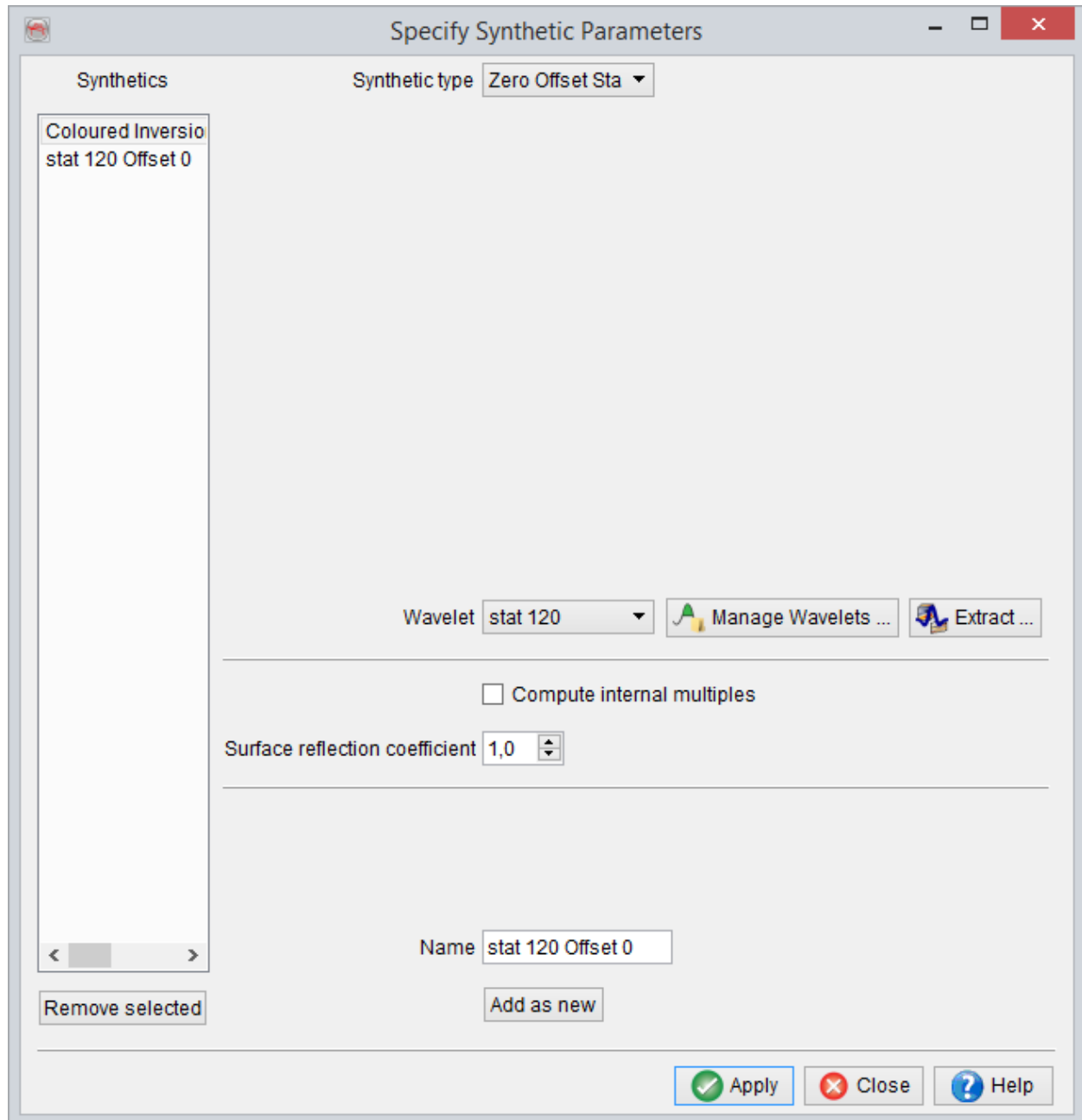
where Z_1 and Z_0 are the impedance of the top and bottom layers, respectively. These layers are basically upscaled and time converted version of various logs (Rho, Vp and Vs) in pseudo-well models, and as such comprise the Elastic Model [hyperlink with Elastic Model] for synthetic seismic generation. The upscaling is done using the Backus averaging algorithm in depth, but at a (variable) depth sampling rate which is equivalent to the seismic sample rate in time. Depth-to-time conversion of the pseudo-well logs, is done using the velocity model of the pseudo-wells itself.

Backus upscaling is done only for Vp, Vs and Density logs (and other logs based on them e.g. AI, LambdaRho, MuRho etc.). All other logs e.g. Phi, Sw etc. are upscaled using thickness weighted averaging (i.e. weights used for the averaging are the thicknesses of various pseudo-well layers) and are afterwards converted into time (using the velocity model of the pseudo-wells), at survey sample rate. A Nyquist filter, as defined by the survey sample rate is also applied on these time converted rock property traces; e.g. if seismic survey sampling is at 4 ms, Nyquist filter will allow a maximum frequency of 125Hz. These are accessible to user in real-time on the Variable Density View:



Now, computation of above described reflection coefficient, at all the possible acoustic impedance contrasts in upscaled pseudo-well layers, gives rise to a reflectivity trace in time.

This reflectivity trace is then convolved with a user defined wavelet, to create the Zero Offset Stack for all the pseudo-well models:

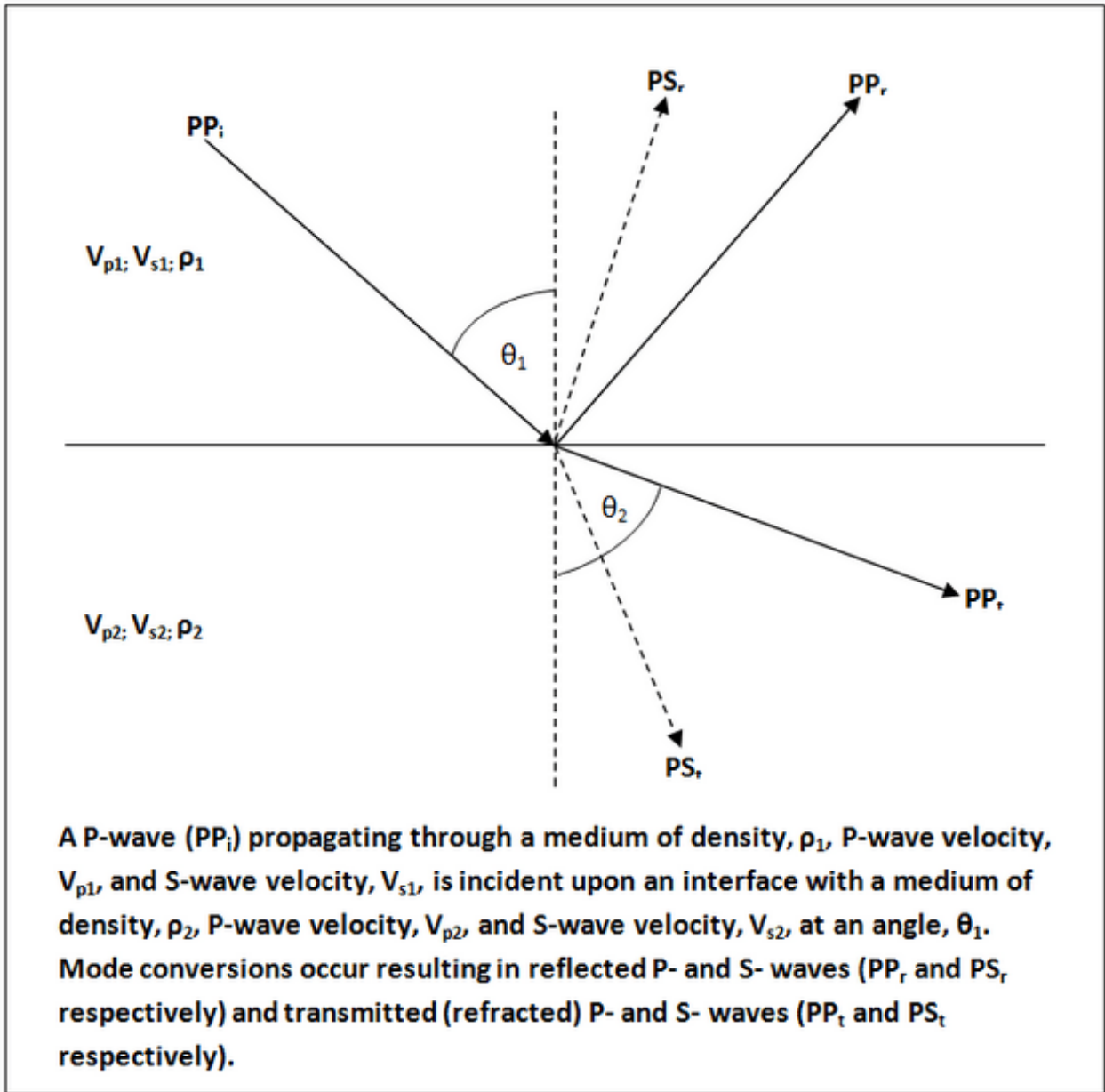


Computation of the Reflection Coefficient at any non-zero offset Pre Stack data (i.e. offset gathers) can be generated in OpendTect using full Zoeppritz equations and ray tracing (simple or advanced).

Full Zoeppritz equations are used to compute angle-dependent reflectivity from the elastic model (i.e. upscaled and time converted version of various logs (Rho, Vp and Vs) from pseudo-wells) at various interfaces as:


(above images are from Wikipedia)

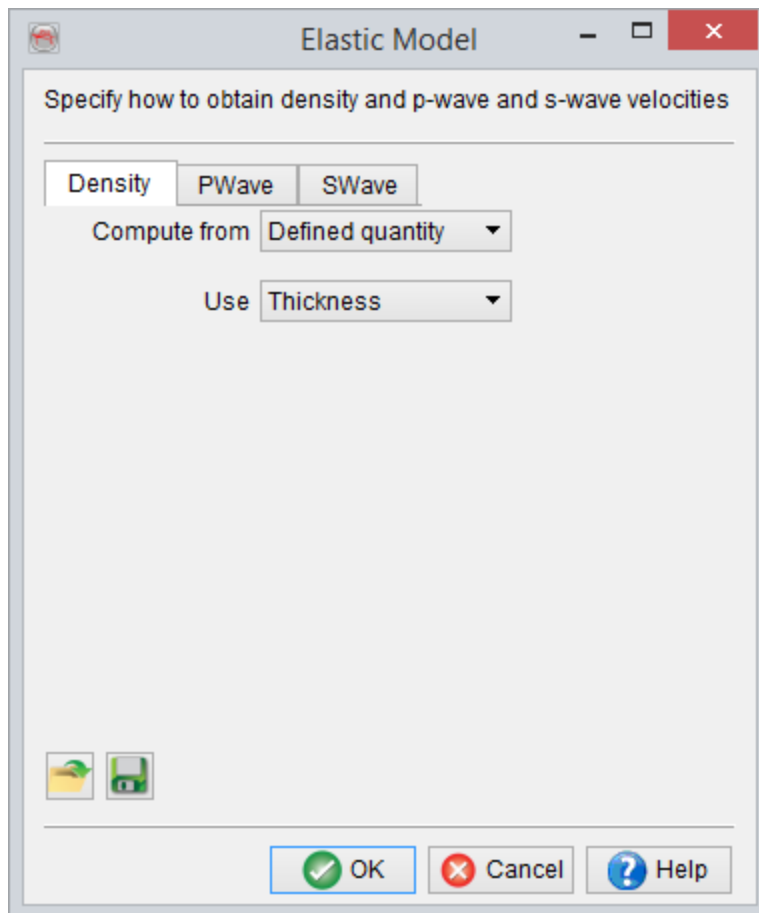
$$\begin{bmatrix} R_P \\ R_S \\ T_P \\ T_S \end{bmatrix} = \begin{bmatrix} -\sin \theta_1 & -\cos \phi_1 & \sin \theta_2 & \cos \phi_2 \\ \cos \theta_1 & -\sin \phi_1 & \cos \theta_2 & -\sin \phi_2 \\ \sin 2\theta_1 & \frac{V_{P1}}{V_{S1}} \cos 2\phi_1 & \frac{\rho_2 V_{S2}^2 V_{P1}}{\rho_1 V_{S1}^2 V_{P2}} \cos 2\phi_1 & \frac{\rho_2 V_{S2} V_{P1}}{\rho_1 V_{S1}} \cos 2\phi_2 \\ -\cos 2\phi_1 & \frac{V_{S1}}{V_{P1}} \sin 2\phi_1 & \frac{\rho_2 V_{P2}}{\rho_1 V_{P1}} \cos 2\phi_2 & \frac{\rho_2 V_{S2}}{\rho_1 V_{P1}} \sin 2\phi_2 \end{bmatrix}^{-1} \begin{bmatrix} \sin \theta_1 \\ \cos \theta_1 \\ \sin 2\theta_1 \\ \cos 2\phi_1 \end{bmatrix}$$



(above images are from Wikipedia)

Elastic Model

This Elastic Model can be accessed by clicking the  icon, just left of 'Wavelet'. This model is required by OpendTect for generating synthetic seismic data (both zero offset stack and pre stack gathers). The elastic model essentially tells the software which quantities to use for the reflection coefficient computation and ray tracing, in terms of Density, V_p and V_s :



If "Compute from: Defined quantity" is chosen, OpendTect can use appropriate (upscaled and time converted) quantities from pseudo-wells. User can also choose to

compute missing quantities (not modeled in pseudo-wells) using pre-filled rock-physics relations, e.g. Vs from Vp using Castagna's equation:

Elastic Model

Specify how to obtain density and p-wave and s-wave velocities

Density PWave **SWave**

Compute from Castagna's equation

Formula $c_0 \cdot V_p + c_1$

For c_0 Use Constant Value 0.8618999

For V_p Use Pwave velocity

For c_1 Use Constant Value -1172

Quantity name: SWave

OK Cancel Help

Glossary

A

Absolute Impedance

Full-bandwidth impedance inversion response in which the "missing" low-frequency part of the spectrum has been added by the inversion method. For example in model-driven inversions the low-frequency model is typically created by interpolating impedance well logs guided by mapped seismic horizons.

Accomodation Space

The available space for sediments to fill (measured from seafloor to base-level).

AI

Acoustic Impedance: the product of seismic velocity and density

Attribute

An attribute is a derived quantity from a seismic input set. Attributes in OpendTect are defined by a name, a value, and a position in 3D space (inline, cross-line and Z (2WT or depth)). Attributes can be calculated from single-trace, multi-trace, and multi-volume inputs. They can be steered and/or chained. Steered attributes are multi-trace attributes in which the trace segments are found by following a (pre-)calculated dip-azimuth. Chained attributes are attributes derived from other attributes. For example, Similarity and Energy are separate attributes that can be chained to calculate the Similarity of the Energy using the "Position" attribute.

Attribute Set

An attribute set is an entity consisting of a group of attributes. Usually attributes in a defined attribute set have something in common. For example, all attributes in a set have the potential to highlight an object type of interest, or a combined attribute, using all other attributes as intermediate results. This would be a desirable output.

B

Base level

The surface at which sediment supply, relative sea level changes and wave energy are in balance. This is the surface at which the accommodation space equals zero: there is neither deposition, nor erosion.

Body

A body is an element that defines an arbitrary three dimensional geological shape (or a geo-body). The body can also be created manually or by using polygons.

c

ChimneyCube

A volume that highlights vertical disturbances in seismic data. The cube is used in fluid migration path studies, in prospect ranking and for fault seal analysis. A ChimneyCube is generated by a neural network that was trained on picked examples (chimneys and non-chimneys). It gives at every sample location the "chimney probability" i.e. the likelihood of belonging to the class of identified seismic chimneys.

Chrono-stratigraphy

A set of relative geologic time lines as stored in a HorizonCube.

CLAS

A plugin for petrophysical analysis. CLAS stands for Computer Log Analysis System.

Closed Source

Software that is released in binary form only. The commercial plugins to OpendTect are released as closed source extensions. Such extensions are only permitted if OpendTect is run under a commercial (or academic) license agreement.

Color Blending

Combined display of three (four) attributes that are displayed in the Red Green and Blue color channels. Optionally the fourth channel (alpha) displays transparency. Color blending is aka as RGB (RGBA) blending.

Crossline Dip

Dip in the direction of the Crossline axis, or in the direction of increasing crosslines.

D

Dip-Steering

The process of auto-tracking seismic data by following the pre-calculated, local dip and azimuth of the seismic. Dip-steering is used for: a) extracting seismic trace segments along seismic reflectors as input to multi-trace attribute calculations, b) computing special attributes such as polar dip, azimuth, and volume curvature attributes, c) filtering seismic data (known as dip-steered

filtering, aka structurally oriented filtering), and d) auto-tracking chrono-stratigraphic horizons in the creation of a HorizonCube.

Dip-Steering Cube

A volume computed from seismic data with at every sample position information about the local dip and azimuth of the seismic data. In a 3D Steering Cube this information is stored in two attributes per sample: inline dip and cross-line dip. On 2D seismic only one value is stored: the line-dip. Dips in a Steering Cube are measured in the line direction and expressed in us/m or mm/m, for time and depth data, respectively.

E

EEI

Extended Elastic Impedance. Scaled and rotated impedance response at a particular angle. Rotation is typically optimized to predict a certain well log property of interest.

EI

Elastic Impedance. Impedance response at a particular angle of incidence.

Element

An element is a sub-division of various items (of the tree) that are displayed in a 3D scene. Inline, crossline, timeslices, horizon, wells etc are some elements. Each element is sub-divided into a sub-element. For instance an inline element can have further sub-elements e.g. inline # 120 that can contain upto eight different attributes.

Eustatic sea-level

Sea-level relative to center of earth.

Explicit Representation

A representation of a 3D object in OpendTect in the form of a triangulated surface.

F

Fault Stickset

The faults are interpreted on a section as a stick, and all sticks that belong to one fault are grouped in one sticksets. Therefore, a fault stickset contains an unordered collection of the interpreted sticks.

Forced regression

Deposition characterized by progradation and incision. Base-level is falling decreasing accomodation space, forcing the system to prograde. Forced regression occurs during the Falling stage systems tract.

G

GMT

An open source mapping package developed and maintained by the University of Hawaii (<http://gmt.soest.hawaii.edu/>). GMT stands for Generic Mapping Tools.

GPL License

Gnu General Public License (<http://www.gnu.org/licenses/gpl.html>) is an open source license under which OpendTect can be run. The license allows redistribution of (modified) source code under the same licensing conditions (copy left principle). It is not allowed to combine the open source part with closed source plugins, which is why OpendTect is also licensed under a commercial license agreement and under an Academic license agreement.

H

Horizon Data

It refers to a stored attribute grid in a horizon. An attribute is calculated on-the-fly or in a batch process. On-the-fly, a user needs to store by right-clicking on it an selecting Save attribute... option. The saved attribute can also be managed in the Manage horizon. It may be noted that a horizon can contain unlimited stored attribute/horizon data.

HorizonCube

A dense set of auto-tracked (or modeled) seismic horizons that is indexed and sorted according to relative geologic time (= chrono-stratigraphy).

I

Implicit Representation

A representation of a 3D object in OpendTect in the form of an iso-surface through a cube of values.

Incision

Depositional feature caused by erosion.

Inline Dip

Dip in the direction of the Inline axis, or in the direction of increasing inline numbers.

M

Madagascar

An open source seismic processing package. See: [http://en.wikipedia.org/wiki/Madagascar_\(software\)](http://en.wikipedia.org/wiki/Madagascar_(software))

Meta Attribute

A meta-attribute is an attribute created from multiple input attributes. In OpendTect, a meta attribute is created either through neural networks, or through mathematical manipulations and/or logical operations. For example, TheChimneyCube and TheFaultCube are meta-attributes. See the Ridge enhancement filter attribute set from the Default attribute sets for an example of a meta-attribute created through math and logic. The meta-attribute in this set is the last attribute in the list.

MLP Neural Network

Multi-Layer-Perceptron type of neural network. The network is used for seismic object detection (creating Chimney Cubes, Fault Cubes, Salt Cubes etc.) and for predicting rock properties from seismic data (Porosity, Vshale, Sw etc). An MLP network is trained on a data set with known examples (supervised learning). In the training phase the network aims to find the optimal, non-linear mapping between input attributes and target attributes. The network in OpendTect is a fully-connected, three-layer MLP (input layer, hidden layer, output layer). The non-linear transformation takes place in the hidden layer.

MPSI

A plugin for stochastic acoustic impedance inversion. MPSI stands for Multi-Point Stochastic Inversion.

N

Normal regression

Deposition characterized by aggradation and progradation. The base level is rising but the consumption of accommodation space by sedimentation exceeds the creation of accommodation space by the base level rise. Normal regression occurs during high stand and low stand systems tracts.

O

Open Source

Software that is released with its source code. OpendTect is released as open source product that can be extended with closed source plugins. Such extensions are only permitted if OpendTect is run under a commercial (or academic) license agreement.

P

PDF

PDF is Probability Density Functions. In OpendTect these are created in the cross-plot tool by selecting a desired area in the cross-plot domain. The density of the points in the selected area is a measure for the probability of the desired target variable that can then be predicted by applying the derived PDF function to (scaled) input volumes in a Bayesian classification scheme.

Pickset

A Pickset is a collection of picked locations, i.e. inline-crossline-Z information. Picksets are part of a Pickset Group. For example a Pickset Group containing picks at fault locations may consist of different fault Picksets to differentiate between large faults and small faults, or to reflect picks on different inlines.

Pickset Group

A Pickset group is a collection of different Picksets. Usually Picksets are grouped because they refer to the same object, e.g. Chimney_yes or Chimney_no.

R

Regression

Seaward shoreline and facies shift. Regression can be Normal (base level rises) or Forced (base level falls).

Relative Impedance

Band-limited impedance inversion response computed by methods such as colored inversion.

Relative sea-level

The net effect of eustatic sea level changes and local tectonic fluctuations.

Retrogradation

Depositional trend characterized by sediments building landwards aka back-stepping.

S

SEG-Y

A file format for exchanging seismic or seismic-like data. It is used for both 2D and 3D pre- or poststack data. A file being SEG-Y compliant does not mean that it can be loaded into OpendTect. There are several possible problems. One of these is missing trace identification and/or positioning. Another issue is lack of true compliance (->SEG-Y Rev 0, -> SEG-Y Rev 1). The different types of SEG-Y are shown below: * SEG-Y Rev 0: The initial SEG-Y specification in 1975. It is very precise in some areas but totally unspecified in other, crucial areas. This has led to an almost uncountable number of variants. Some are sort-of SEG-Y standard, others blatantly non-compliant. * SEG-Y Rev 1: In 2002 the Revision 1 document made an end to the most obvious shortcomings of ->SEG-Y Rev 0,

especially in the area of ->trace positioning and ->trace identification. Still many SEG-Y files or files claimed to be SEG-Y are Rev 0 or badly (i.e. not) compliant with Rev 1. This is why OpendTect has numerous options for the SEG-Y reading process. * SEG-Y Textual header: The first 3200 bytes of a SEG-Y file must be filled with textual comment on the contents of the SEG-Y file. Older textual headers are encoded in EBCDIC rather than ASCII, which makes them impossible to read in a standard text editor. * SEG-Y EBCDIC header: -> SEG-Y Textual header. * SEG-Y Tape Header: The part of a SEG-Y file that gives information about all traces in the file. This information is in the ->SEG-Y Textual header and ->SEG-Y Binary header. * SEG-Y Binary header: The second part of the SEG-Y Tape header contains binary information about, amongst others, values for number of samples per trace, byte encoding, sample interval, and SEG-Y Revision. * Trace identification: Every trace in OpendTect needs to have an identification in form of a trace number (2D data) or inline/crossline (3D data). For prestack data the offset forms and extra trace identification. * Trace positioning: In OpendTect, every seismic trace needs to be located in 3D space. For 3D data, the position can be derived from the->Trace identification (inline and crossline numbers). Traces in 2D lines have their own, separate X- and Y-coordinate. For prestack data there must also be an offset available.

SSIS

A plugin to perform a sequence stratigraphic analysis (systems tracts, Wheeler transforms) from seismic data using HorizonCube input. SSIS stands for Sequence Stratigraphic Interpretation System.

Stratal Slicing

The process of cutting through a seismic volume along surfaces that are computed proportionally between mapped top and bottom horizons, aka proportional slicing.

Systems Tracts

Subdivisions of sequences that consist of discrete depositional units that differ in geometry from other systems tracts and have distinct boundaries on seismic data. Different systems tracts are considered to represent different phases of baselevel changes.

T

Trace Identification

Every trace in OpendTect needs to have an identification in form of a trace number (2D data) or inline/crossline (3D data). For prestack data the offset forms and extra trace identification.

Trace Positioning

In OpendTect, every seismic trace needs to be located in 3D space. For 3D data, the position can be derived from the->Trace identification (inline- and crossline numbers). Traces in 2D lines have their own, separate X- and Y-coordinate. For prestack data there must also be an offset available.

Transgression

Landward shoreline and facies shift characterized by aggradation and retrogradation. The base-level is rising and more accommodation space is created than is consumed by sedimentation

Tree

The tree is the docking window, which is detachable and movable. This is used to display the data into a scene. The tree is attached to a scene and is labeled as Tree Scene 1. Where '1' is the scene number. Each tree has its own elements that are displayed in corresponding scene.

U

UVQ Neural Network

Unsupervised Vector Quantizer type of neural network. This network is used for clustering (segmenting) data into a user-defined number of clusters. Cluster centers are found in a training run on a subset of the data. In the application phase the network generates two outputs: 1) the index number of the winning cluster and 2) the match, a value between 0 and 1 indicating how close the input vector is to the vector representing the winning cluster. UVQ segmentation can be performed in 2D mode (waveform segmentation along mapped horizons) and in 3D mode (generates 3D bodies). A display of the cluster centers is a useful diagnostic in waveform segmentation (Neural Network module: Info button).

V

VMB

A plugin for picking velocities from semblance gathers, and in a surface-consistent-manner. VMB stands for Velocity Model Building.

W

Waveform Segmentation

Process of clustering seismic trace segments with a UVQ network along a mapped horizon into a user-defined number of clusters.

WCP

A plugin to pick and QC well log markers with the help of seismic data and (optionally) the HorizonCube. WCP stands for Well Correlation Panel.

Wheeler Transform

Process of flattening seismic data (or attributes) according to the chrono-stratigraphic horizons in a HorizonCube. In a Wheeler scene the vertical axis represent relative geologic time.