3D PROBABILISTIC SEISMIC FACIES ANALYSIS – PROGRAM gtm3d (Generative Topographic Mapping)

Overview

Like Self-organizing Maps (SOM), Generative Topographic Mapping (GTM) maps high-dimensional data (e.g. five or more attributes per voxel or 12 amplitude samples about a picked horizon) to a lower-dimensional (usually 2D or 3D) latent space, which are then mapped to a continuous 2D or 3D color bar. There are two major limitations to the popular Self-organizing Maps (SOM) clustering algorithm. First, there is no clear rule for selecting the training radius and the learning rate; these parameters are data dependent. Second, because of the absence of any defined cost function there is no measure of "confidence" in the final clustering results to indicate the convergence at the final iteration. GTM is a probabilistic reformulation of SOM that takes care of most of these shortcomings.

First introduced by Bishop et al. (1998), GTM generates a probability density model that describes the distribution of *N D*-dimensional (multi-attribute) input data vectors, $\mathbf{x_n}$, in terms of a relatively smaller number, *K*, *L*-dimensional latent variables, $\mathbf{u_k}$, where $L<D$. Each latent variable in GTM defines the mean location, $\mathbf{m_k}$, of a *D*-dimensional Gaussian distribution. The "mixture" or sum of these probability density functions statistically describes the input data. The model parameters (the means of the Gaussian probability density functions) are determined by maximizing the likelihood estimation of the summed probability density function that predicts the input data vectors, $\mathbf{x_n}$.

In K-means and SOM clustering each data vector $\mathbf{x_n}$ is assigned to the nearest cluster center or prototype vector, $\mathbf{p_k}$. In GTM, the "prototype vectors" are replaced by the uniformly placed grid points (nodes) $\mathbf{u_k}$, each of which share a certain "responsibility", R_{nk}, in representing each data vector $\mathbf{x_n}$. Once found, there are several ways to display this relationship. The analogue to SOM would be to assign each data vector $\mathbf{x_n}$, to a cluster value, *k*, (and corresponding color) to the Gaussian $\mathbf{u_k}$ component which is most responsible. The posterior probability of the data value $\mathbf{x_n}$ is projected in the 2D latent space. We may choose to find the expected value (pdf-weighted mean) of the $\mathbf{x_n}$ vectors or the most likely position (mode) of the data value and assign it a color corresponding to its location in *L*-dimensional latent space.

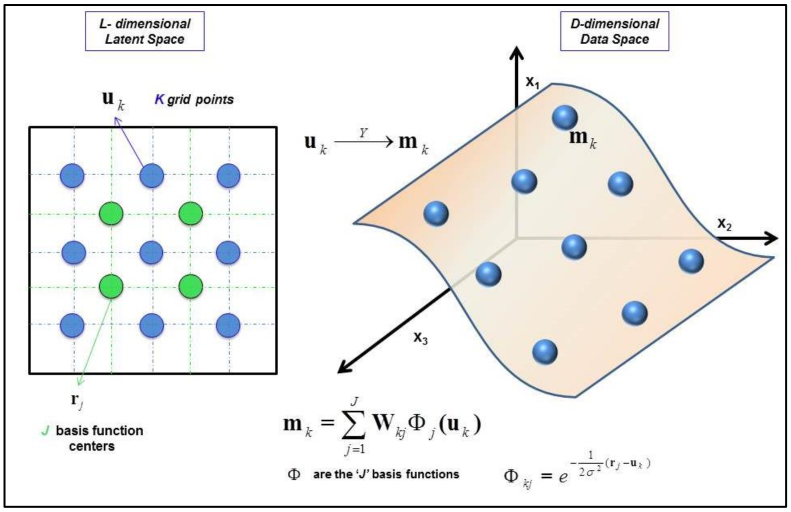


Figure 1: The prior distribution consists of latent space variables ordered on a regular grid (blue circles) residing in an *L*-dimensional latent space. In this figure *L*=2. A smaller number of *J* non-linear non-orthogonal Gaussian basis functions are used to interpolate latent space grid points to the *D*-dimensional data-space. ϕ_j consists of a regular array of Gaussian functions with a predefined constant standard deviation. Thus with the linear combination of these basis functions the latent space (blue circles) are mapped to the data-space (blue spheres) on the 2D non-Euclidean manifold S. Thus, each node $\mathbf{u}_k$ is then mapped to a corresponding point $\mathbf{m}_k$ in data-space, given by $\mathbf{m}_\mathrm{k} = \sum_{j=1}^{J} \mathbf{W}_{kj} \phi_j(\mathbf{u}_\mathrm{k})$ (equation 1).

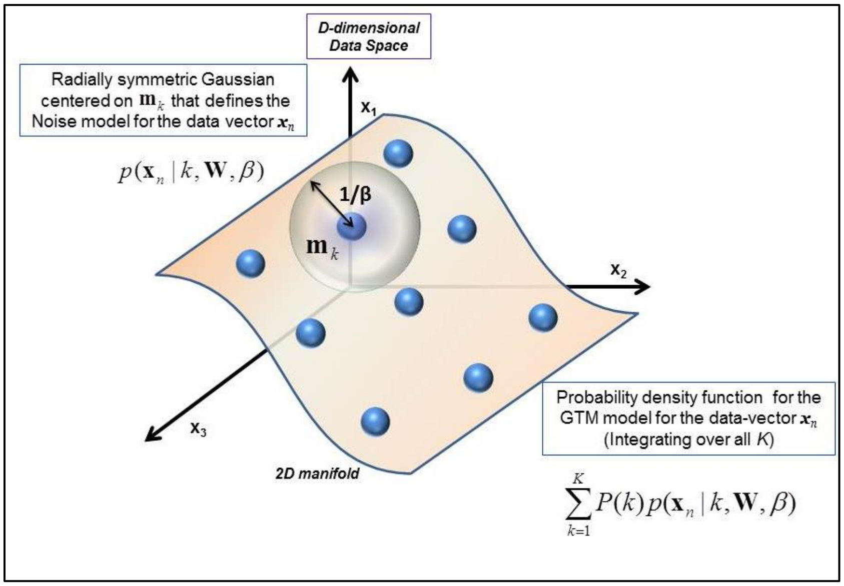


Figure 2: The 2D non-Euclidean manifold S with the mapped reference grid points in the data-space $\mathbf{m}_k$. A pdf is defined for a data-vector $\mathbf{x}_n$ with a radially symmetric Gaussian functions with centers at $\mathbf{m}_k$ and having a variance of 1/β given by

$p(\mathbf{x}_n|k, \mathbf{W}, \beta)$. The final probability density function of a GTM model is obtained by summing the contribution of each weighted pdf.

GTM Theory

In general, we wish to find a nonlinear projection $\mathbf{m}_k$ $(\mathbf{u}_k;\mathbf{W})$ which maps the K points $\mathbf{u}_k$ ($k=1,2,3,\ldots,K$) in the two- or three-dimensional latent space into the K points in the D-dimensional data-space. This non-linear transformation is given by

$$\mathbf{m}_\mathrm{k} = \sum_{j=1}^{J} \mathbf{W}_{kj}\phi_j(\mathbf{u}_\mathrm{k}) \ , \qquad (1)$$

where $\mathbf{W}$ is a D x J matrix of unknown weights, $\phi_j(\mathbf{u}_\mathrm{k})$ is a set of J non-linier basis functions, and $\mathbf{m}_k$ are reference vectors in the data space. A noise model (the probability of the existence of a particular data vector $\mathbf{x}_n$ given weights $\mathbf{W}$ and inverse variance β) is introduced for each measured data vector $\mathbf{x}_n$**.** The probability density function, p, is represented by a suite of K radially symmetric D-dimensional Gaussian functions centered about $\mathbf{m}_k$ with variance of $1/\beta$:

$$p(\mathbf{x}_n|\mathbf{W},\beta) = \sum_{k=1}^{K} \frac{1}{K}\left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} e^{-\frac{\beta}{2}\left\|\mathbf{m}_k-\mathbf{x}_n\right\|^2} \ . \qquad (2)$$

The prior probabilities of each of these components are assumed to be equal with a value of $1/K$, for data vectors $\mathbf{x}_n$ that range from $n=1,2,\ldots, N$. Figure 1 illustrates the GTM mapping from an L=2-dimensional latent space to the 3-dimensional data space.

The probability density model (GTM model) is fit to a dataset $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3\ldots\ldots\mathbf{x}_N\}$ to find the parameters $\mathbf{W}$ and β using a maximum likelihood estimation. One of the popular techniques used in parameter estimations is the Expectation Maximum (EM) algorithm. We calculate the N x K posterior probability or *responsibility*, R_{nk}, which each of the K components in latent space takes for every data-vector using the current values of the GTM model parameters $\mathbf{W}$ and β and Bayes theorem:

$$R_{nk} = \frac{e^{-\frac{\beta}{2}\left\|\mathbf{m}_k-\mathbf{x}_n\right\|^2}}{\sum_j e^{-\frac{\beta}{2}\left\|\mathbf{m}_j-\mathbf{x}_n\right\|^2}} \ . \qquad (3)$$

Equation 3 forms the "E-step" or Expectation step in the EM algorithm. The E-step is followed by the Maximum or "M-step", that uses these responsibilities to update the model for a new weight matrix $\mathbf{W}$ by solving a set of linear equations,

$$\left(\mathbf{\Phi}^T\mathbf{G}\mathbf{\Phi} + \frac{\alpha}{\beta}\boldsymbol{I}\right)\mathbf{W}^T{}_{new} = \mathbf{\Phi}^T\mathbf{R}\mathbf{X} \qquad (4)$$

where

$\mathrm{G}_{kk} = \sum_{n=1}^{N} \mathrm{R}_{nk}$ are the non-zero elements of the K x K diagonal matrix $\mathbf{G}$**,**

$\mathbf{\Phi}$ is a K x J Matrix with elements $\mathbf{\Phi} = \phi_j(\mathbf{u}_k)$,

α is regularization constant to avoid division by zero, and

$\mathbf{I}$ is the J x J identity matrix.

The updated value of β is given by

$$\frac{1}{\beta_{new}} = \frac{1}{ND}\sum_{n=1}^{N}\sum_{k=1}^{K} R_{nk} \left\|\mathbf{W}_{kj_{new}}\phi_j(\mathbf{u}_\mathrm{k}) - \mathbf{x}_\mathrm{n}\right\|^2 \ . \qquad (5)$$

The initialization of $\mathbf{W}$ is done so that the initial GTM model approximates the principal components (largest eigenvectors) of the input data, $\mathbf{x}_\mathbf{n}$. The value of β^{-1} is initialized to be the larger of the $(L+1)^{th}$ eigenvalue from PCA where L is the dimension of the latent space. In Figure 1, L=2, such that we initialize β^{-1} to be the inverse of the third eigenvalue. Figure 2 summarizes this workflow.

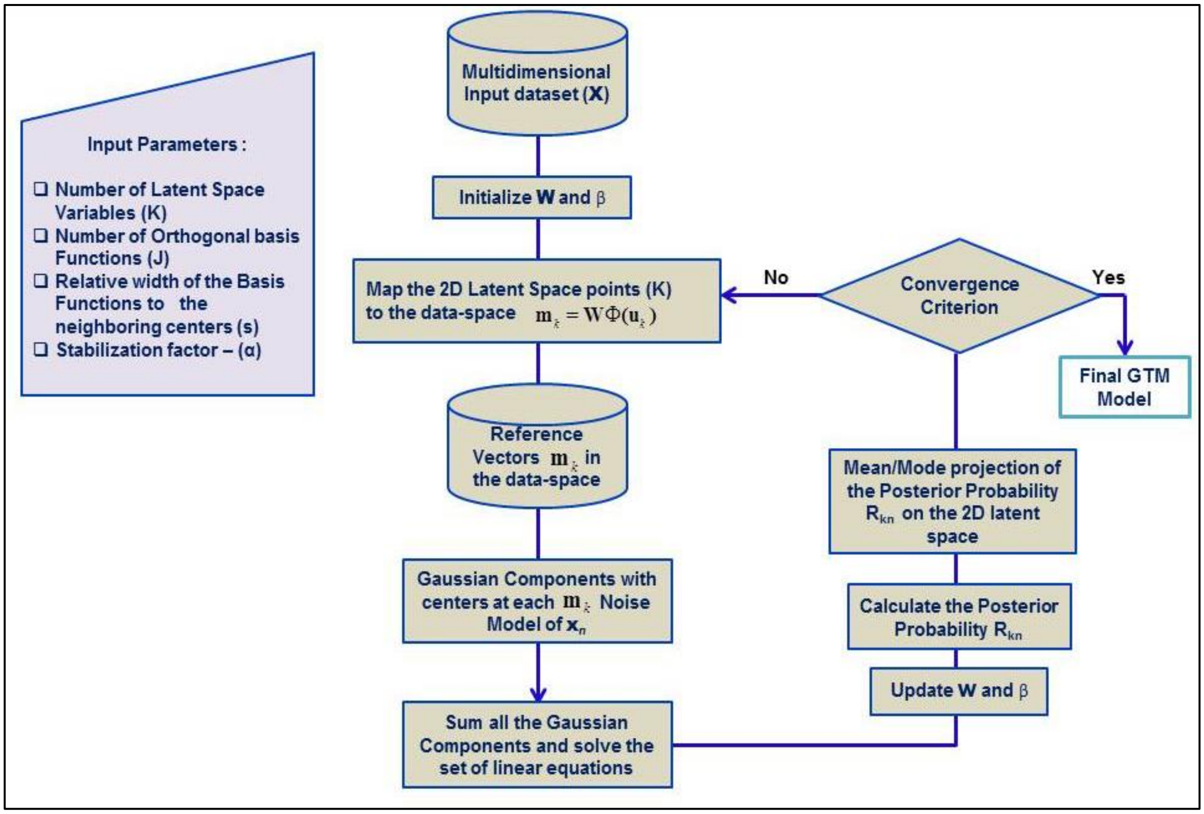


Figure 2. Generative topographic mapping (GTM) workflow.

Data Visualization in GTM

We will use the responsibility, R_{nk} , (also called the posterior probabilities) to determine the mode or the pdf-weighted mean of each input data vector, $\mathbf{x}_n$, in the lower dimensional latent space (Figure 2). The mode represents the grid point in the latent space having the maximum posterior probability value

$\mathbf{U}_{mode}(\mathbf{x}_n) = \mathrm{MAX}(\mathbf{u}_k)$.

The mean is simply

$$\mathbf{U}_{mean}(\mathbf{x}_n) = \sum_{k=1}^{K} R_{kn}\mathbf{u}_k$$

where $\mathbf{u}_k$ are the grid positions in the 2D latent space.

We use two ways of displaying our clusters. For simplicity, let's assume we are using a 2D latent space. The more conventional way is to color the mode or mean of each data point using a 2D color bar (e.g. Strecker and Uden, 2002; Matos and Marfurt, 2010). Unfortunately, most commercial interpretation software does not allow for simple 2D color bar manipulation. The second method uses crossplotting tools that are found in most commercial interpretation software packages. Rather than assign an integer label to a cluster as we do in SOM, we extract the GTM x and y (distances along eigenvectors 1 and 2) components in the latent space and output two 'attributes' that can then be crossplotted and edited using geoprobe technology. This second approach facilitates the generation of one or more geobodies.

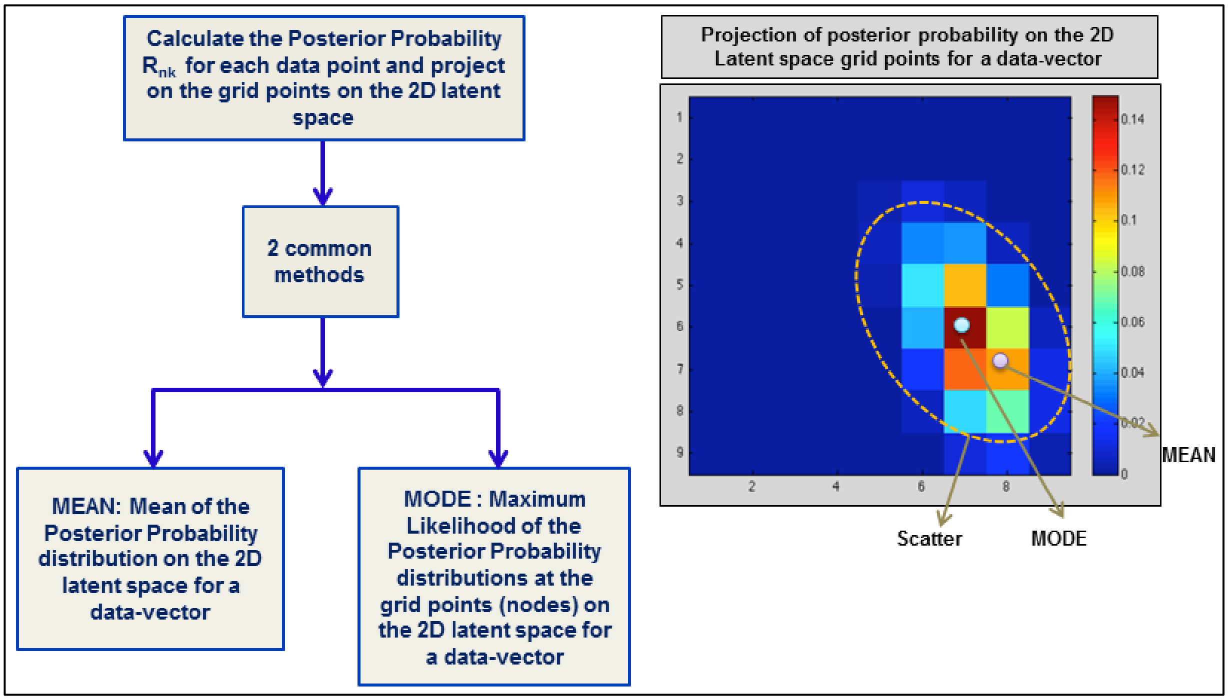


Figure 3: Workflow for the data visualization by GTM. Bishop et al.(1998) used the posterior distribution "responsibility" matrix **R**, to compute either the mode or the mean of the projected *N*-dimensional data-vector $\mathbf{x_n}$, onto the lower 2D latent space nodes $\mathbf{u}_k$, The mean location will assign the value $\mathbf{U}_{mean}(\mathbf{x}_n)$ to be the weighted average of the posterior distribution values and will in general fall in between neighboring values of $\mathbf{u}_k$. The mode will assign the value $\mathbf{U}_{mode}(\mathbf{x}_n)$ to be the location of the greatest posterior distribution value in the 2D latent space and will always correspond to a discrete gridded value of $\mathbf{u}_k$.

Example 1: GTM data visualization of a reservoir completion problem

Our first application of a GTM workflow uses engineering "attributes" rather than seismic attributes and is discussed in detail by Roy et al. (2012). The input data vectors, $\mathbf{x_n}$, correspond to n=137 horizontal wells from the Haynesville shale. The spatial distribution of the pilot holes are shown in Figure 4. Each of the 137 well has 13 engineering and 2 geologic parameters:

1. Total clean volume of sand,
2. Total proppant volume
3. Total 100-mesh sand
4. Total non-100-mesh sand
5. Daily peak rate
6. Cluster spacing
7. Number of hydraulic fracture stages

8. Total perforations,
9. Total perforation cluster,
10. Total perforation length,
11. Contour permeability
12. Average treating rate
13. Average proppant concentration,
14. Thickness of the formation, and
15. Porosity,

describing a *D*=15-dimensional data space. Each of these 15 parameters are hypothesized to affect the EUR of each well:

Each of the 15 input components are normalized using a *z*-score algorithm to remove the impact of measurement units and to precondition the data to be better represented by our Gaussian probability density functions. Applying the GTM technique to these data results in posterior probabilities (responsibility) in a 2D latent space that can be mapped to form a mode- or mean-distribution map of each input data vector in the output latent space. These projected points are then colored by the scaled estimated ultimate recovery (EUR) values.

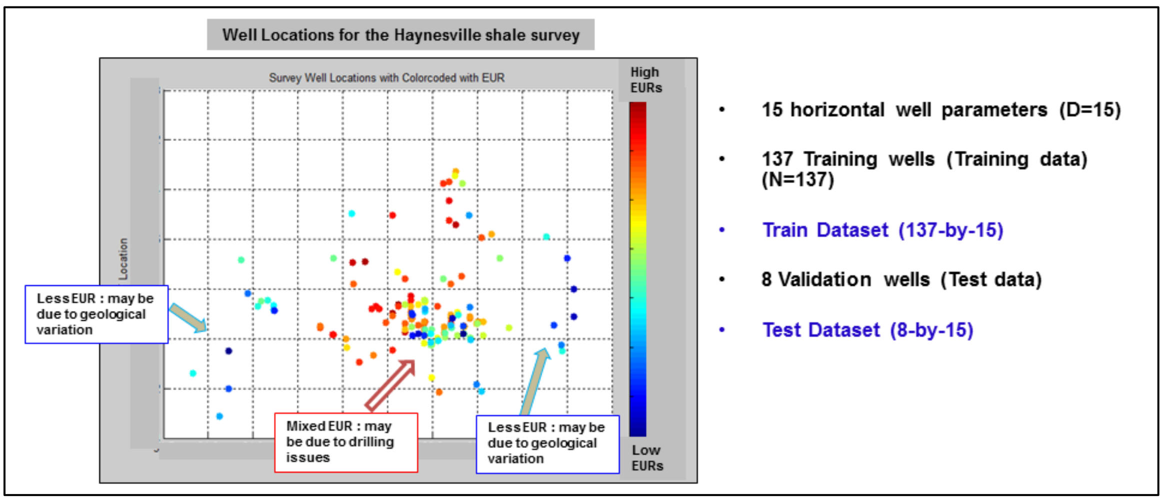


Figure 4: Spatial distribution of the pilot holes of 145 wells drilled in a Haynesville shale play of an area roughly 1000 km^2. Colors correlate to EUR with blues indicating low EUR, cyan and yellow intermediate EUR, and red high EUR. 137 wells will be used to train and eight wells will be used to validate the GTM. Note high- and low-EUR wells are not spatially separated.

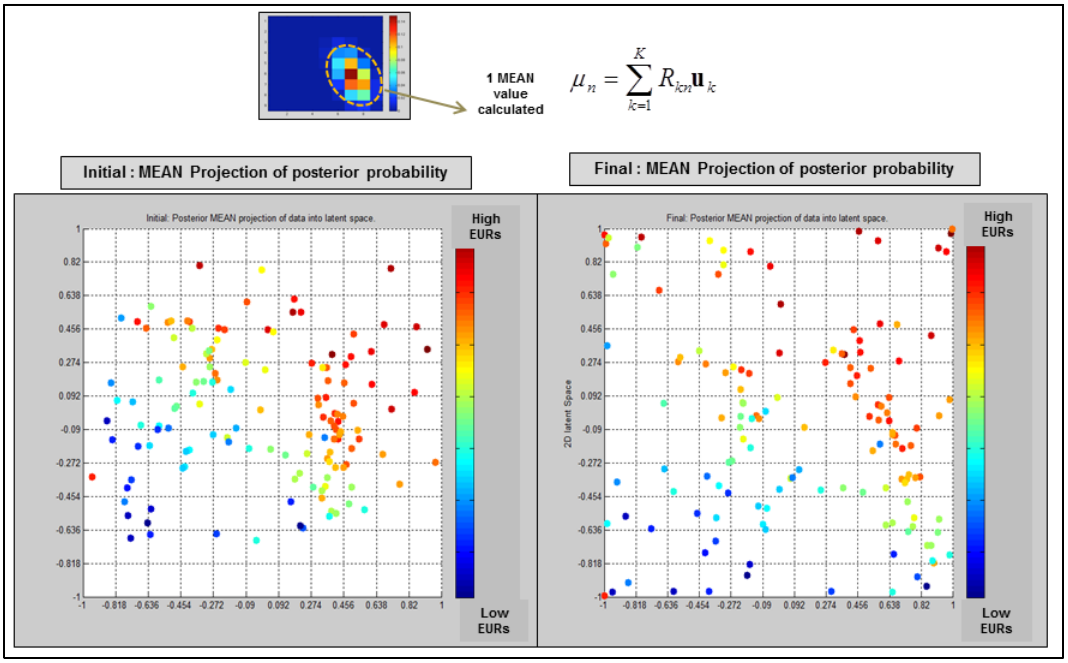


Figure 5: The mean posterior distribution map of the "*responsibilities*" of the data in the 2D Latent space. Here the data-vectors are projected onto the mean locations calculated from their posterior probability values at the grid points. (a) Initial distribution of the posterior mean projections of the data in the latent space. (b) Final distribution of the posterior mean projections of the data in the latent space after 100 iterations. The plot is color-coded by the scaled EUR values. Note the separation between the good (red) and the bad (blue) EUR values in (b).

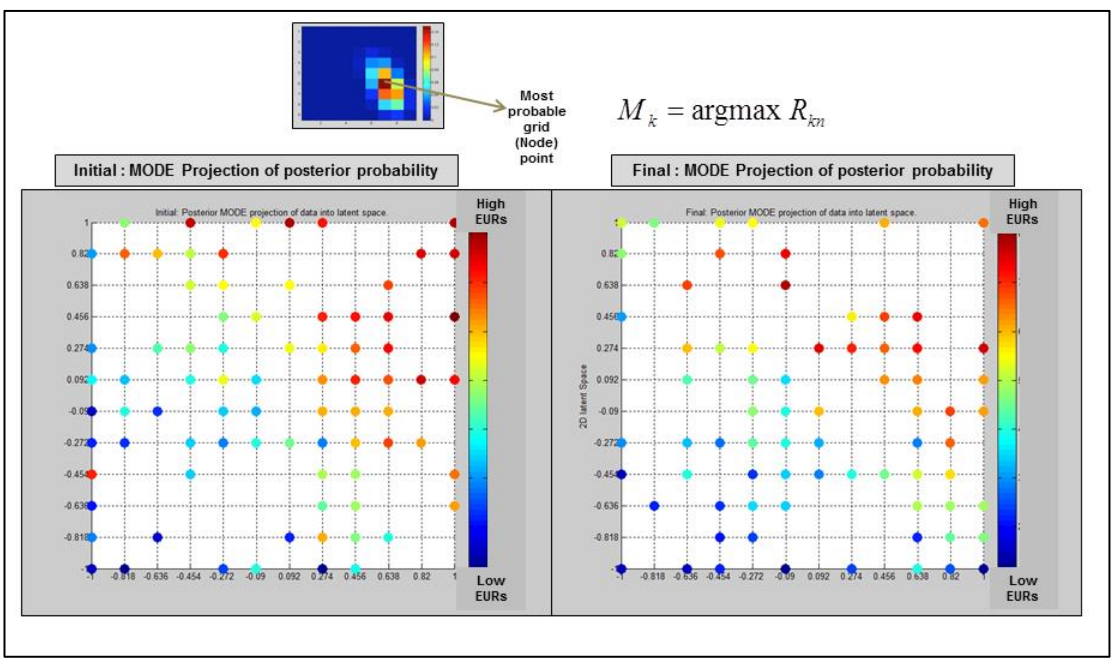


Figure 6. The mode posterior distribution map or the "responsibility" of the data in the 2D latent space. The data-vectors are projected onto the most likely grid points (grid points with the highest value of R_{nk}). (a) Initial distribution of the posterior mode

projections of the data in the latent space. (b) Final distribution of the posterior mode projections of the data in the latent space after 100 iterations. The plot is color-coded from low to high EUR values. Note the latent space shows a more orderly separation between the good, moderate and the bad EURs for the final iteration.

The mean or the mode of the posterior probability (responsibility) distribution map of every data-vector is plotted in the 2D latent space before and after convergence shown in Figures 5 and 6. Figure 5 depicts the distribution of the mean posterior probabilities for all the data vectors in the latent space. The mode values represent the most probable index location (location of the grid points in the latent space) having the maximum posterior probability values for all of the data-vectors. The color-coding varies from the high (red) to low (blue) EUR values. Both the mean and the mode projections show better clustering or separation in the final iteration. Analysis of the components according to their importance still needs to be done.

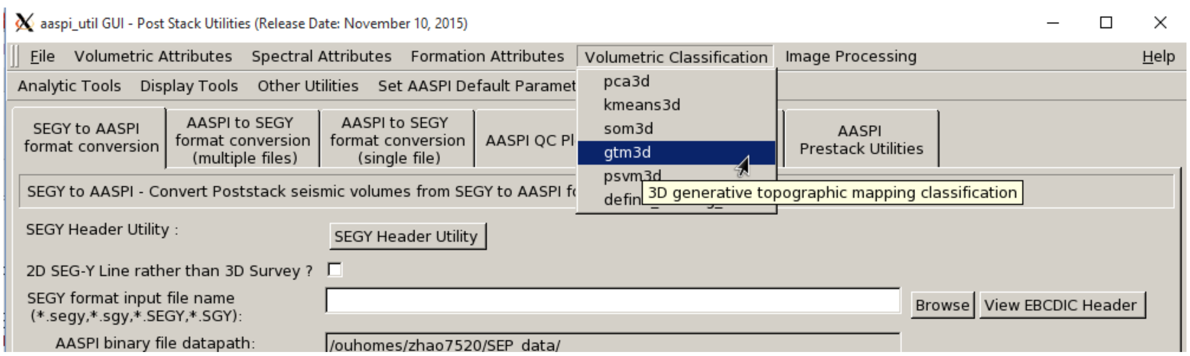
Volumetric Classification: Program **gtm3d**

AASPI implementation

In the AASPI software, the GUI for GTM can be invoked by typing

aaspi_gtm3d &

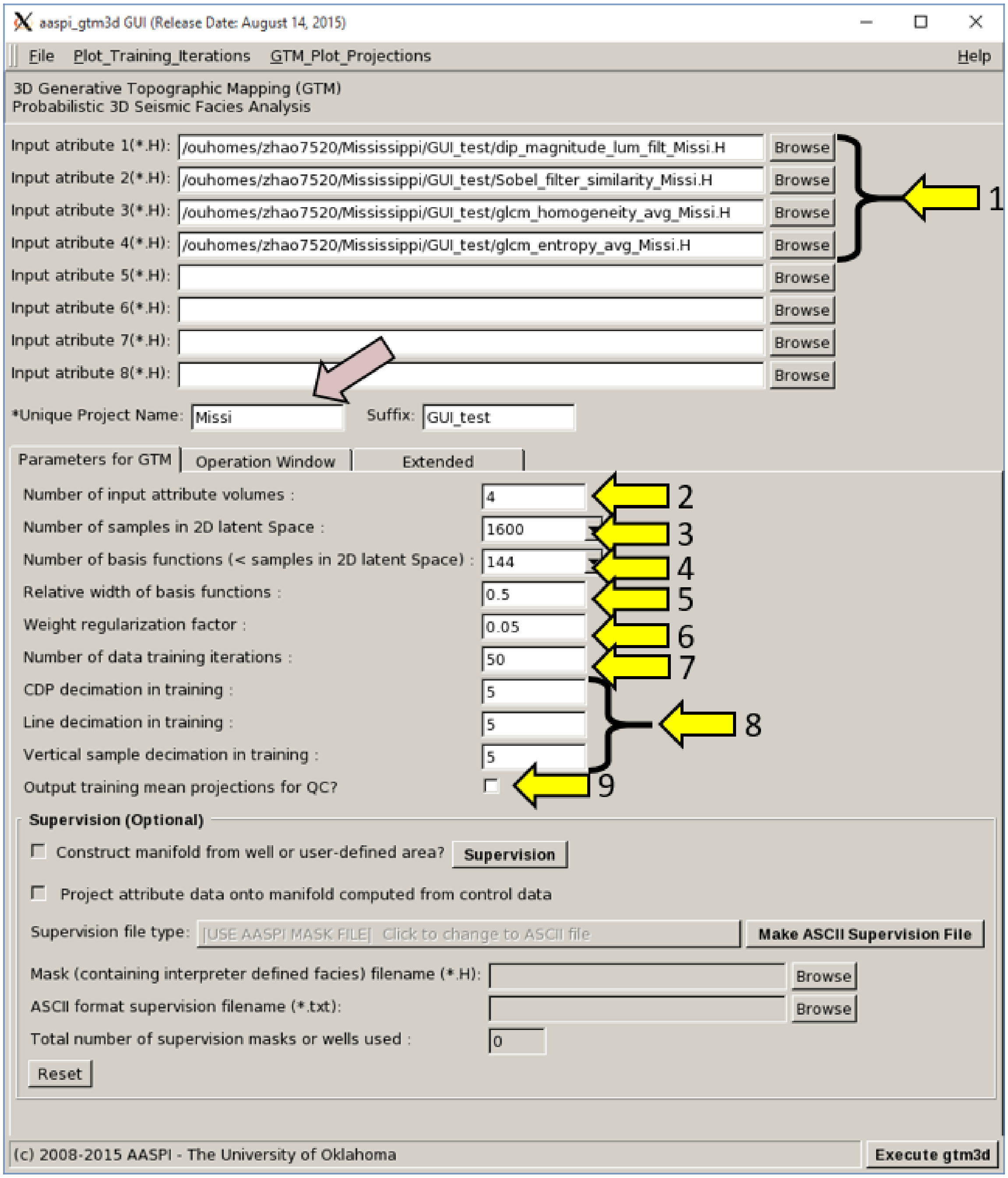
or from the main **aaspi_util** window by selecting the *Volumetric Attributes* drop down menu:



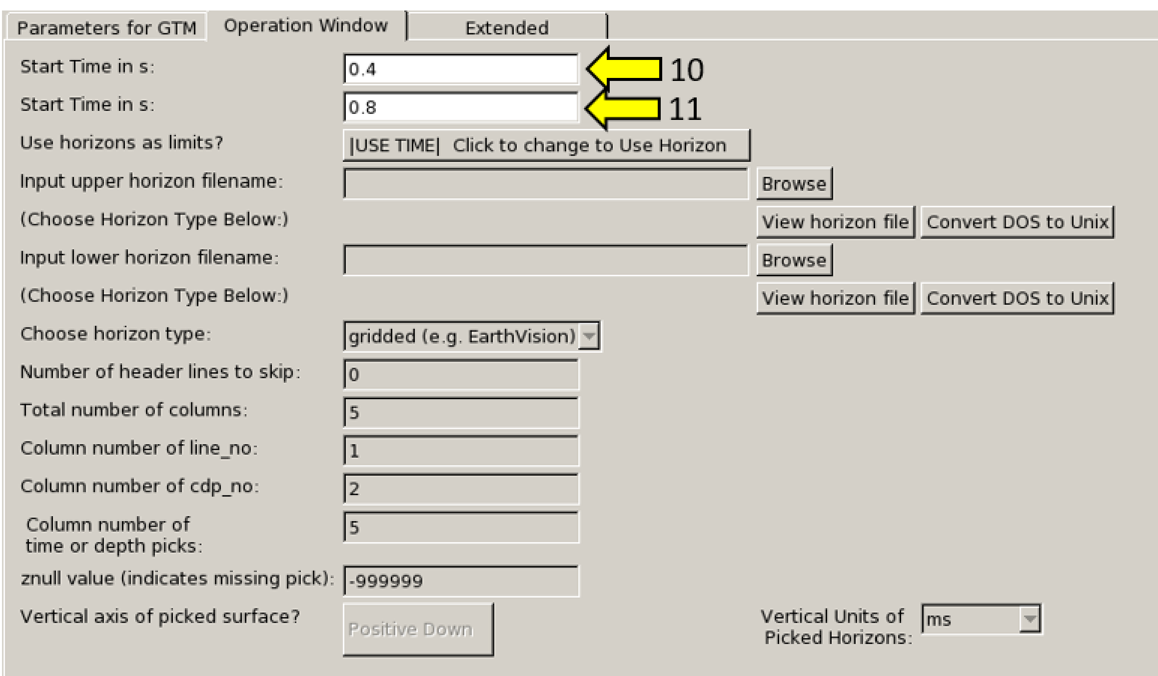
Note :
*Since we are using a 2D manifold in N-dimensional space that is mapped to a 2D latent space, programs **gtm3d** and **som3d** require three or more input attribute volumes. Each attribute volume needs to be the same size. In the current version of software the time window (start time and end time) needs to be smaller (at least one sample) than the time window of input data.*

The following GUI will pop up:

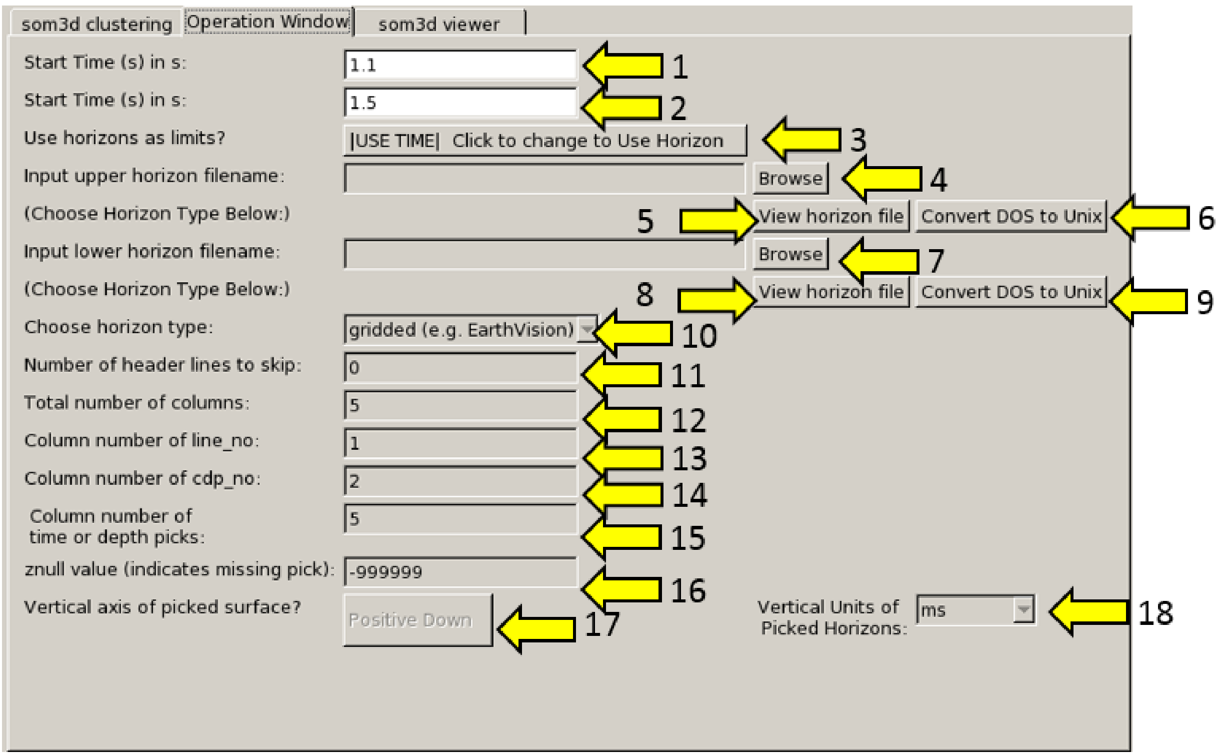
Volumetric Classification: Program **gtm3d**

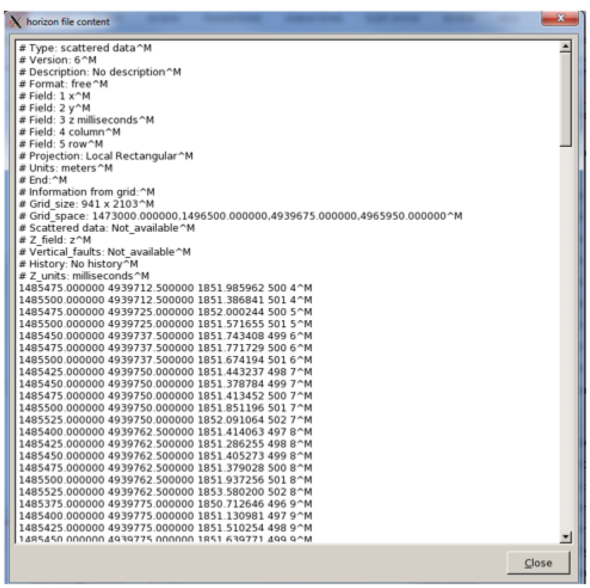


As with SOM, the input consists of a suite of **(1)** seismic attribute volumes (3 or more volumes) that the interpreter has chosen to differentiate different seismic facies, rock types, lithologies, or other clusters. For example, a mass transport complex may be characterized by relatively low coherence, strongly converging reflectors, and high entropy (measured by the GLCM algorithm). Surrounding marine shales may be characterized by moderate coherence, low reflector convergence (i.e. parallel reflectors) and low GLCM entropy. Next, **(2)** enter the number of input volumes represents the dimensionality of the dataset (automatically updated). Then **(3)** select the number of grid points to span the 2D latent space, *K*, (GTM Theory equation 2). These points are mapped to the data-space. Then **(4)** select the number of non-linear basis functions, *J*, – (GTM Theory equation 1) that form a regular array of Gaussian functions. A linear combination of these basis functions is used to map the points in the latent space to the data space. Both the latent space samples and the basis functions should be squared integers value, (e.g. (256…400…625…900 etc.….). They can be automatically selected from some pre-defined values in dropdown menu. Care should be taken so that the number of basis functions (*J*) should be less than the number of grid points in the 2D latent space (*K*). Next, **(5)** Enter the width of the basis functions relative to the distance between two neighboring basis function centers.. This width is used to define the standard deviation of the non-linear basis functions, which is constant for a GTM model. If *s*=2 the basis functions will have widths (std. dev) equals to two times the distance between two neighboring basis function centers. Initially the code runs a multiattribute PCA to initialize the starting values of **W** and β (see GTM Theory). Next, **(6)** enter the regularization factor, α, (GTM Theory equation 4) used to stabilize the linear equation for solving the new **W**. This prevents any division by zero. Next, **(7)**, enter the number of iterations to run GTM. To minimize run times, only a fraction of the input dataset is used for training. Therefore **(8)** enter the factor to decimate the dataset, which will be used for training. For example the values 5, 5, 5 mean every 5th data-vector in a trace, inline, and crossline is used for training, such that we train on every 1 out of 125 samples from the input data. Finally, **(9)** choose if to output the mean projections for all training steps for QC purpose.



In the *Operation Window* tab, similar to **som3d**, one can choose between defining the window using constant time or two horizons. Enter the **(10)** start time and the **(11)** end time of the data for GTM seismic facies classification. Here, we suggest the user to take a one sample less than the actual size of the data, e.g. if the input data is 0.000s to1.000s and the sampling rate is 4ms (0.004s). Then take the start time ≥ 0.004s and end time ≤0.996s. We recommend limiting GTM to the target area since it is computationally intensive, and a horizon-based window is better for analyzing in a relatively constant deposition environment. Below is how to define an operation window using horizons. The panel shown is from **som3d**, and the **gtm3d** has an identical operation window panel.





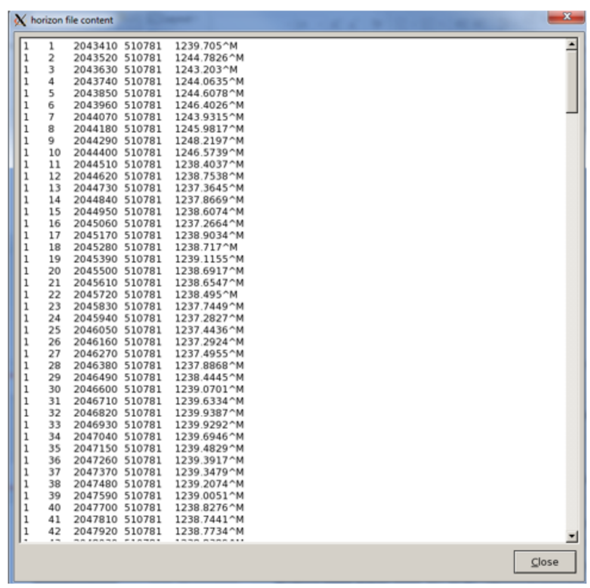


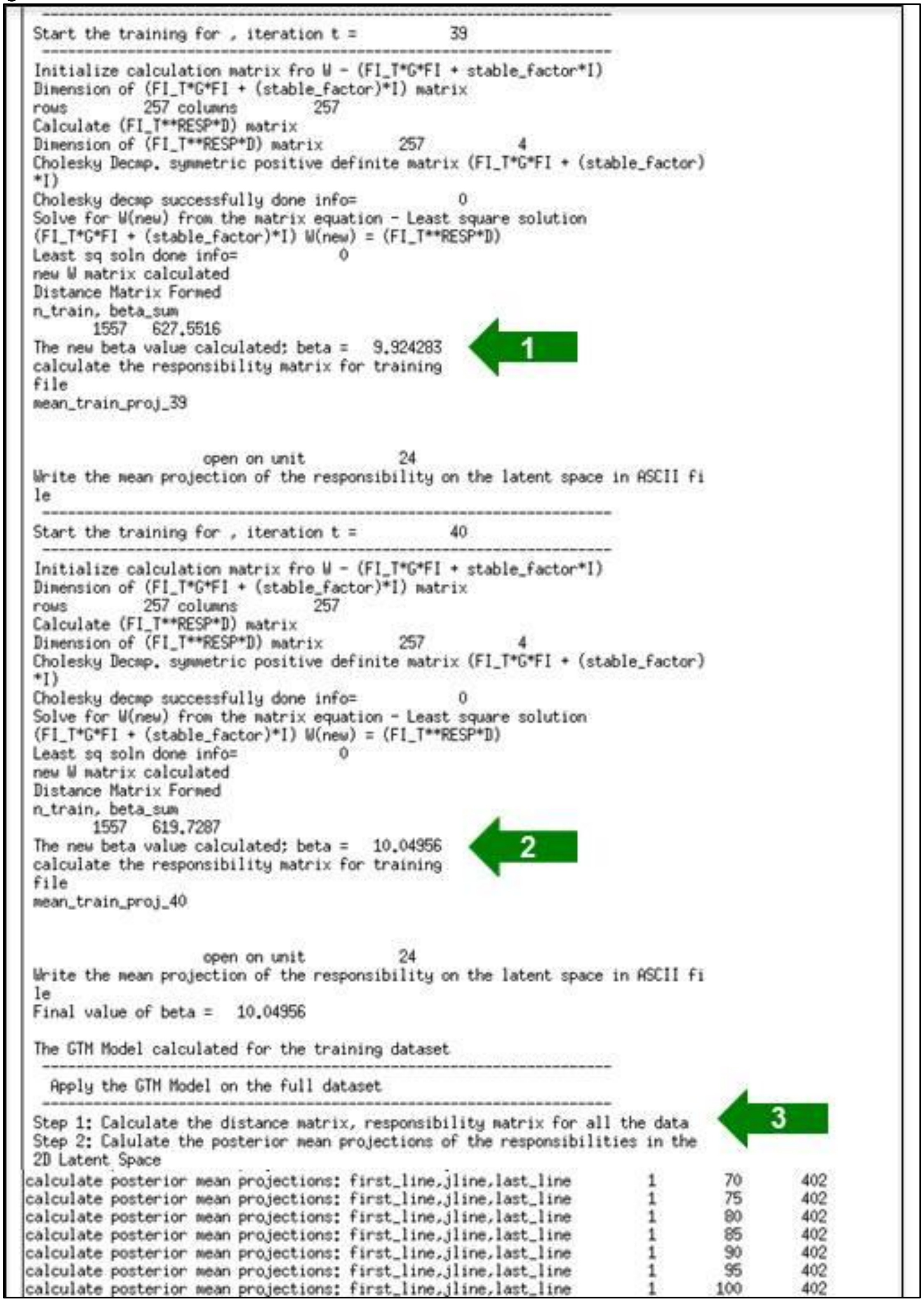
Figure 1. (left) A gridded horizon file (EarthVision format). (right) An interpolated horizon file with five columns (ASCII free format).

Horizon definition

The horizon definition panel will look the same for almost all AASPI GUIs:

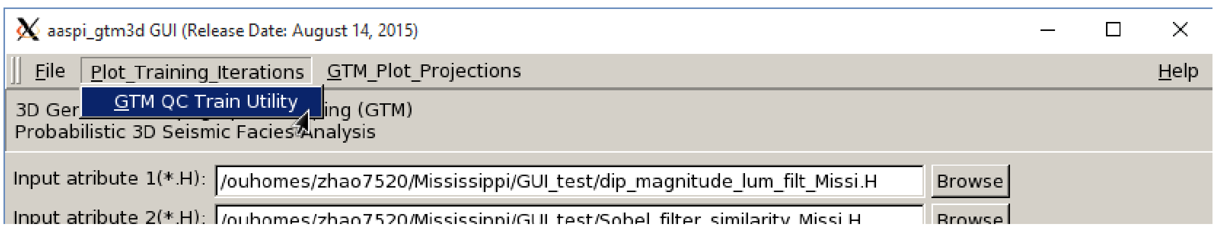
1. Start time (upper boundary) of the analysis window.
2. End time (lower boundary of the analysis window.
3. Toggle that allows one to do the analysis between the top and bottom time slices described in 1 and 2 above, or alternatively between two imported horizons. If *USE HORIZON* is selected, all horizon related options will be enabled. If the horizons extend beyond the window limits defined in 1 and 2, the analysis window will be clipped.
4. Browse button to select the name of the upper (shallower) horizon.
5. Button that displays the horizon contents (see Figure 1).
6. Button to convert horizons from Windows to Linux format. If the files are generated from Windows based software (e.g. Petrel), they will have the annoying carriage return (^M) at the end of each line (Shown in Figure 1). Use these two buttons to delete those carriage returns. Note: This function depends on your Linux environment. If you do not have the program **dos2unix** it may not work. In these situations, the files may have been automatically converted to Linux and thus be properly read in.
7. Browse button to select the name of the lower (deeper) horizon.
8. Button that displays the horizon contents (see Figure 1).
9. Button to convert horizons from Windows to Linux format. (see 6 above).
10. Toggle that selects the horizon format. Currently *gridded* (e.g. EarthVision in Petrel) and *interpolated* (ASCII free format, e.g. SeisX) formats are supported. The gridded horizon are nodes of B-splines used in mapping and have no direct correlation to the seismic data survey. For example, gridded horizons may be computed simply from well tops. The x and y locations are aligned along north and east axes. In contrast interpolated horizons have are defined by *line_no, cdp_no (crossline_no)* and *time* triplets for each trace location. Examples of both format are shown in Figure 1. If *interpolated* is selected, the user needs to manually define each column in the file.
11. Number of header lines to skip in the *interpolated* horizon files.
12. Total number of columns in the *interpolated* horizon files.
13. Enter the column number containing the *line_no (inline_no)* of the interpolated data triplet.
14. Enter the column number containing the *cdp_no (crossline_no)* of the interpolated data triplet.
15. Enter the column number containing the *time* or *depth* value of the interpolated data triplet.
16. *Znull* value (indicate missing picks) in the horizon files.
17. Toggle to choose between positive down and negative down for the horizon files (e.g. Petrel uses negative down).
18. Choose the vertical units used to define the horizon files (either *s*, ms, *kft, ft, km,* or *m*).

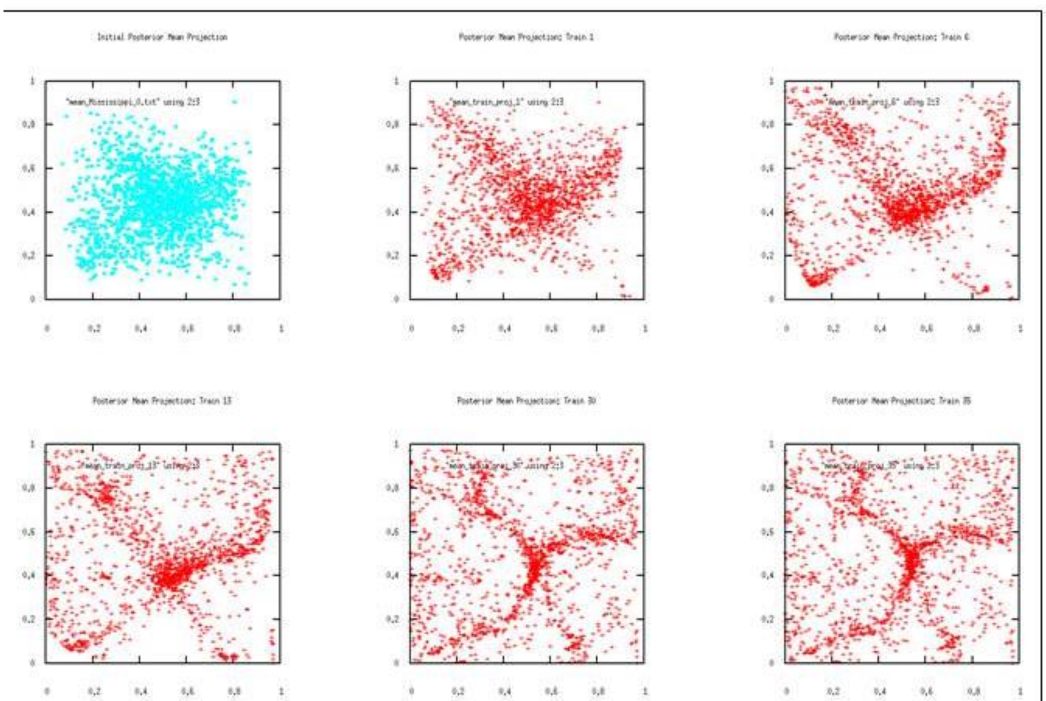
After providing all these parameters click "*Execute GTM*".
If the verbose option is chosen, the output to your xterm will look like the following image:



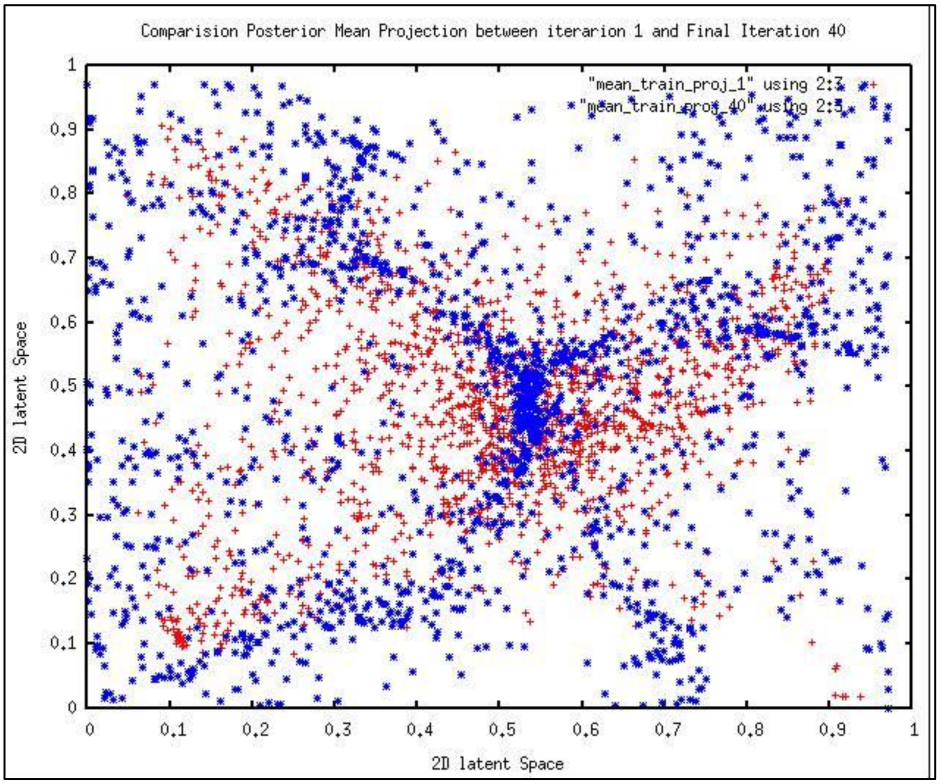
```
-----------------------------------------------------------------------
Start the training for , iteration t =          39
-----------------------------------------------------------------------
Initialize calculation matrix fro W - (FI_T*G*FI + stable_factor*I)
Dimension of (FI_T*G*FI + (stable_factor)*I) matrix
rows        257 columns          257
Calculate (FI_T**RESP*D) matrix
Dimension of (FI_T**RESP*D) matrix          257           4
Cholesky Decmp. symmetric positive definite matrix (FI_T*G*FI + (stable_factor)
*I)
Cholesky decmp successfully done info=              0
Solve for W(new) from the matrix equation - Least square solution
(FI_T*G*FI + (stable_factor)*I) W(new) = (FI_T**RESP*D)
Least sq soln done info=              0
new W matrix calculated
Distance Matrix Formed
n_train, beta_sum
        1557   627.5516
The new beta value calculated: beta =   9.924283
calculate the responsibility matrix for training
file
mean_train_proj_39


                    open on unit           24
Write the mean projection of the responsibility on the latent space in ASCII fi
le
-----------------------------------------------------------------------
Start the training for , iteration t =          40
-----------------------------------------------------------------------
Initialize calculation matrix fro W - (FI_T*G*FI + stable_factor*I)
Dimension of (FI_T*G*FI + (stable_factor)*I) matrix
rows        257 columns          257
Calculate (FI_T**RESP*D) matrix
Dimension of (FI_T**RESP*D) matrix          257           4
Cholesky Decmp. symmetric positive definite matrix (FI_T*G*FI + (stable_factor)
*I)
Cholesky decmp successfully done info=              0
Solve for W(new) from the matrix equation - Least square solution
(FI_T*G*FI + (stable_factor)*I) W(new) = (FI_T**RESP*D)
Least sq soln done info=              0
new W matrix calculated
Distance Matrix Formed
n_train, beta_sum
        1557   619.7287
The new beta value calculated: beta =   10.04956
calculate the responsibility matrix for training
file
mean_train_proj_40


                    open on unit           24
Write the mean projection of the responsibility on the latent space in ASCII fi
le
Final value of beta =   10.04956

The GTM Model calculated for the training dataset
-----------------------------------------------------------------------
  Apply the GTM Model on the full dataset
-----------------------------------------------------------------------
Step 1: Calculate the distance matrix, responsibility matrix for all the data
Step 2: Calulate the posterior mean projections of the responsibilities in the
2D Latent Space
calculate posterior mean projections: first_line,jline,last_line          1          70         402
calculate posterior mean projections: first_line,jline,last_line          1          75         402
calculate posterior mean projections: first_line,jline,last_line          1          80         402
calculate posterior mean projections: first_line,jline,last_line          1          85         402
calculate posterior mean projections: first_line,jline,last_line          1          90         402
calculate posterior mean projections: first_line,jline,last_line          1          95         402
calculate posterior mean projections: first_line,jline,last_line          1         100         402
```

The number of iterations in the above example was set to be 40, such that the iteration will stop after 40 iterations (green arrow 1). The GTM training step ends (green arrow 2) and the results applied to the complete dataset (green arrow 3) to create two attributes that provide the cluster locations (projection of the mean posterior probability of the data-vectors) along the two axes in the latent space to be used in subsequent cross plotting.
After the GTM classification has completed (shown by the green arrow 2 in the terminal window), one can QC the mean distribution of the posterior probability (responsibility) projections of the training data on the 2D latent space by clicking on "GTM QC Train Utility" (Presently this utility uses ***gnuplot***). Gnuplot needs to be installed to view this utility. Otherwise the ASCII files generated can be viewed in any other graph utility manually. After clicking this flowing two *gnuplot* windows will pop up.

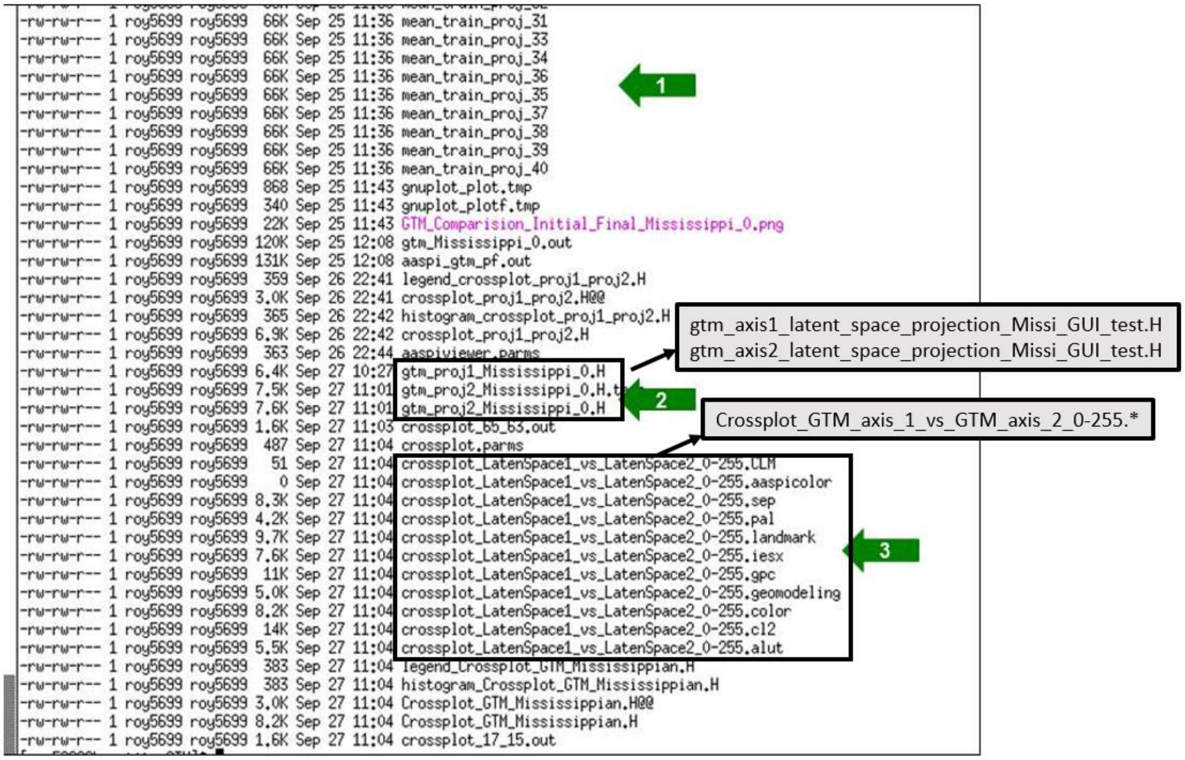




A suite of plots to QC GTM convergence. The initial projection onto a 2D PC plane is shown in cyan. Subsequent iterations are of the mean distribution of the posterior probability (*responsibilities*) projections of the training data onto the 2D latent space are shown in red. Note the projections of $\mathbf{x_n}$ onto $\mathbf{m_k}$ is nearly identical in the last two iterations (30 and 35), indicating convergence.

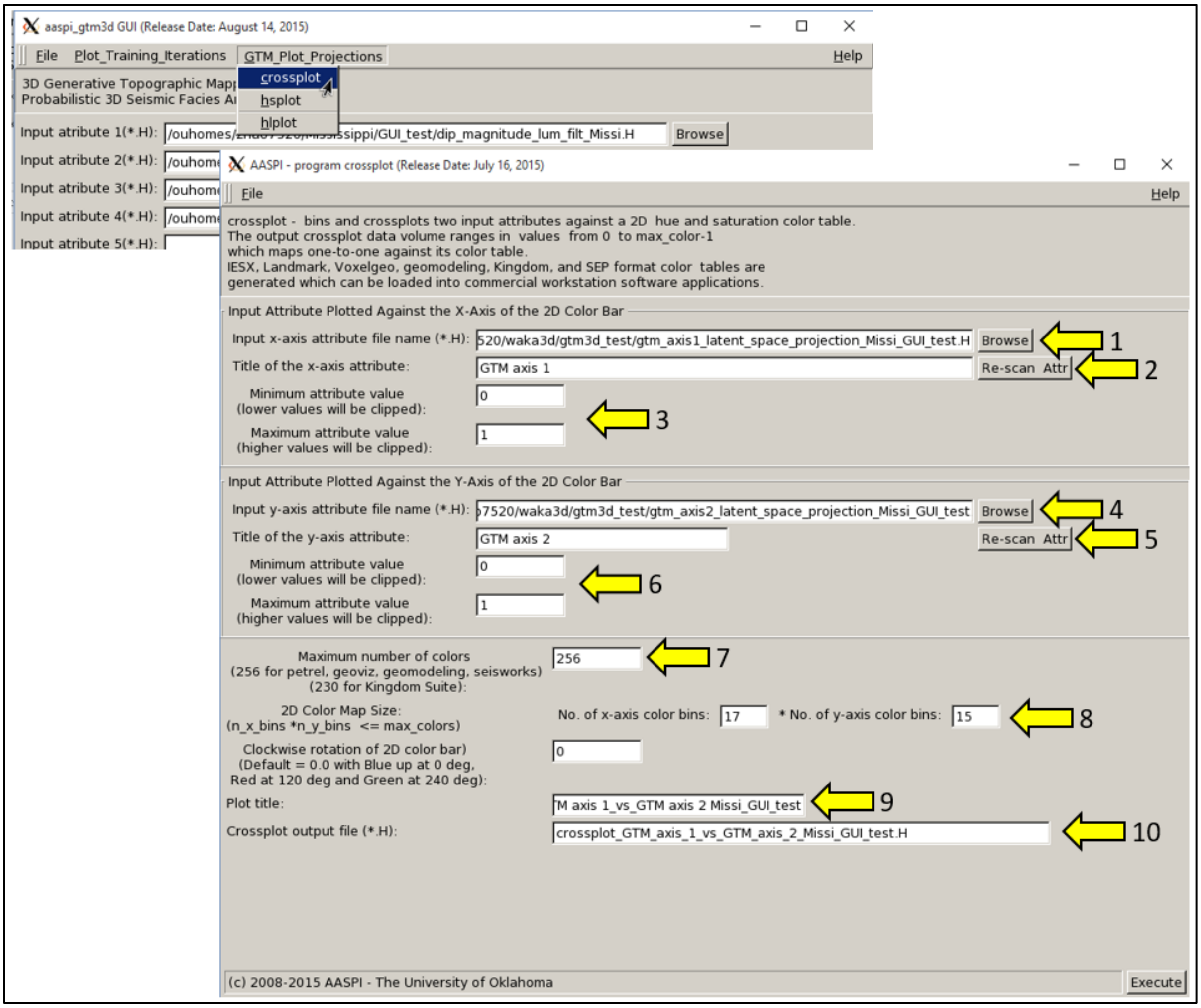


The comparative plot of the posterior probability mean projections of the training data after iteration 1 (in red) and after iteration 40 (in blue). The initial projection is computed by projecting each data vector against the first two principal components.

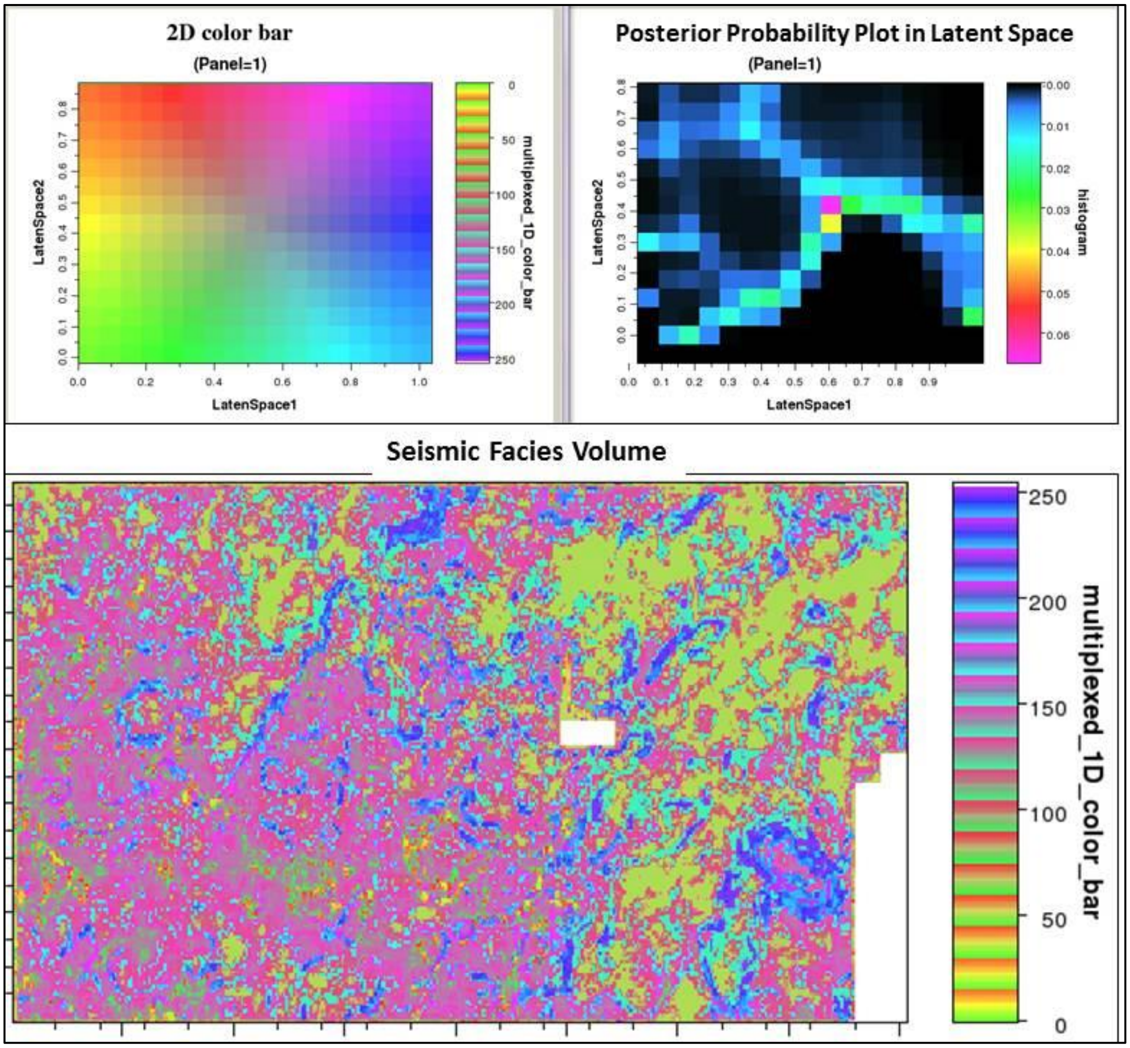


The above terminal window shows the output created by the GTM program. The mean distributions of the posterior probability projections of the training data on the 2D latent space are saved in ASCII files and are given by (green arrow 1). They are plotted by the *gnuplot* utility as shown earlier. The mean distribution of the posterior probability (*responsibility*) projections of the entire data volume onto the 2D latent space are saved as two separate AASPI-format *.H files given by (green arrow 2, names have been changed in the latest release) - *gtm_axis1_latent_space_projection_${project_name}_${sufffix}* and *gtm_ axis2_latent_space_projection_${project_name}_${sufffix}*). These two mean projection volumes can be colored by the *crossplot*, *hsplot* and the *hlplot* utility, which will generate the *crossplot* file and the *sep* colorbar and the *Petrel* colorbar (green arrow 3, names have also been changed in the latest release).

The *crossplot*, *hsplot* and the *hlplot* utility GUI can be invoked by clicking on the “GTM Plot Menu” (yellow arrow) as shown in the GUI. Here the example is shown using the *crossplot* utility.

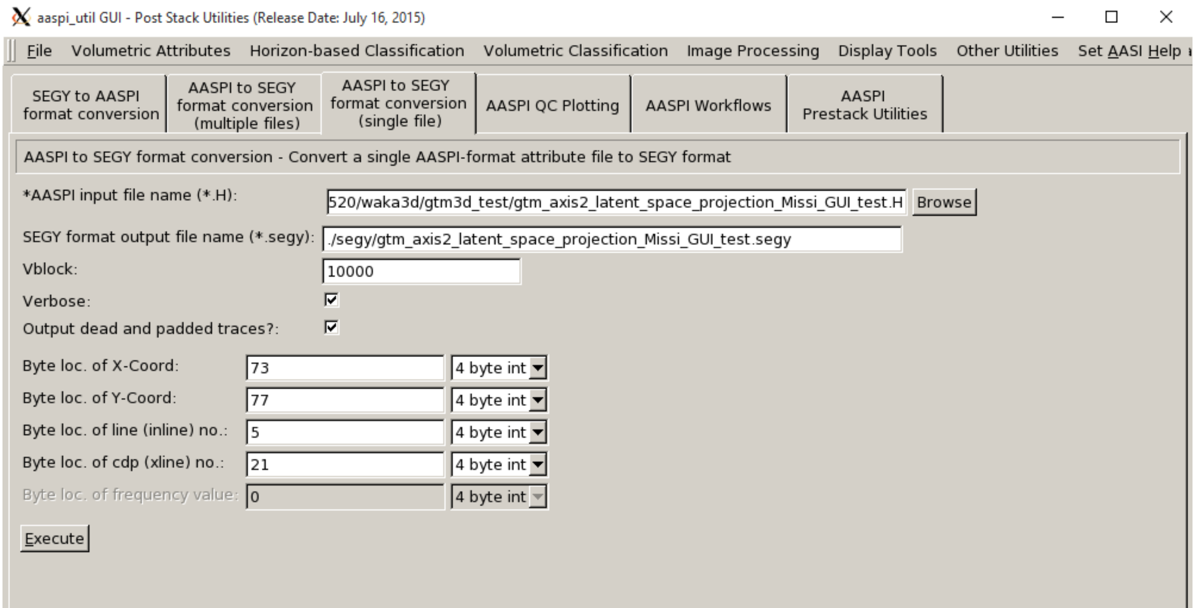


In the *crossplot* GUI the two mean distribution of the posterior probability projections (*responsibilities*) are plotted along (1) the x and (4) the y axes. The range of the two volumes should always be between 0 and 1(arrows 3 and 6). Press the “Scan” button (arrows 2 and 5) to find the amplitude range in the volumes. (7) Enter the maximum number of colors used for visualization. Remember that several of the major commercial workstation software packages limit you to 256 colors. Enter (9) the plot title and (10) the output crossplot file name more details on the crossplot workflow can be found in the program **crossplot** documentation. The output is generated as shown below.



Upper left figure shows a 2D color bar and upper right figure shows a 2D histogram mapped to the 2D GTM latent space. Note the clustering in the histogram into several “arms”. Bottom figure shows a time slice through the classified data volume at the Mississippi Lime level. White zones indicate dead no-permit areas which give rise to the anomalous histogram location in pink.

The mean of the posterior probability in the latent space is plotted as a 2D histogram plot. The 2D colorbar is used to color seismic data-vectors according to their spatial position in the 2D latent space. This crossplot volume generated from GTM clustering and its corresponding colorbar can be imported into commercial interpretation software for more sophisticated visualization and better integration with well data. Simply convert the volumes *gtm_axis1_latent_space_projection_${project_name}_${sufffix}* and *gtm_axis1_latent_space_projection_${project_name}_${sufffix}*) as exported to segy format using the AASPI to SEGY (single file) utility as shown below:

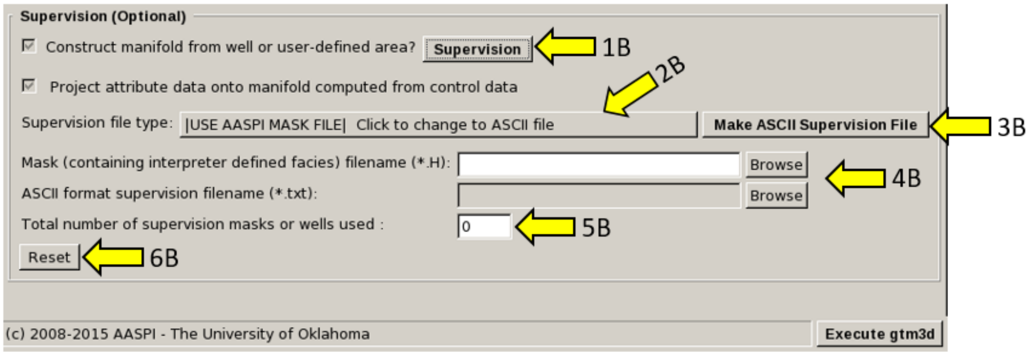


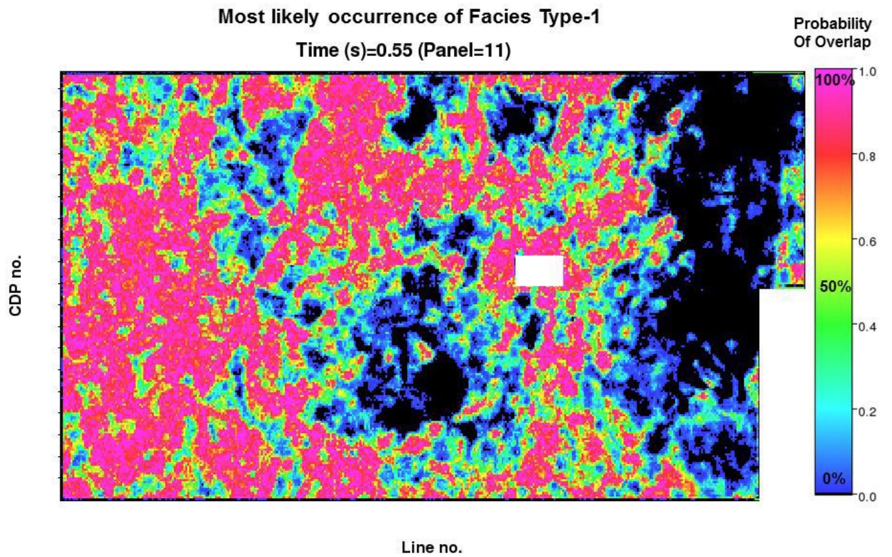
Supervision:

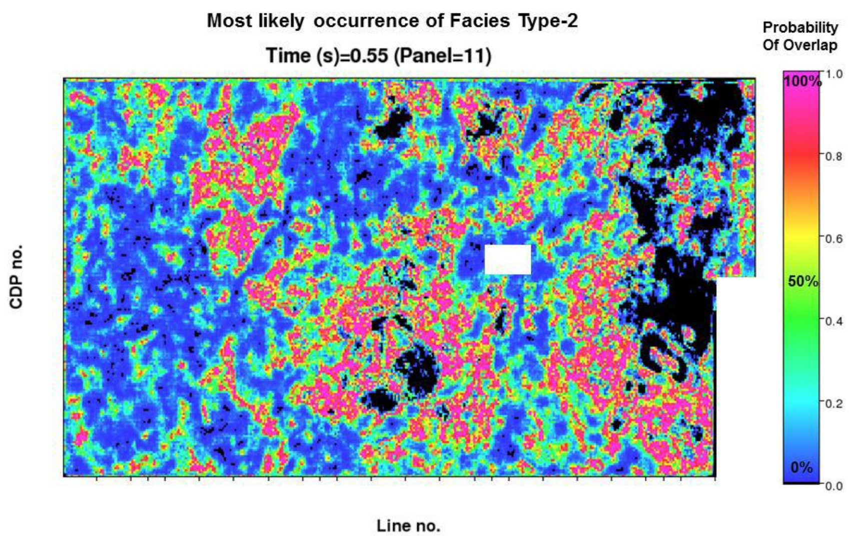
For a better analysis of the how likely one type of facies occurs we need to input some model wavelets from picked within the survey or areas of interest. Such analysis is performed by calculating the *Bhattacharyya distance* (Bhattacharyya, 1943) between the averaged responsibility pdf of supervision data in a particular facies and a data sample. In this way the similarity between this data sample and the particular facies is measured. This is done by clicking the “supervision” button (Arrow 1B). After which a few buttons becomes active. Currently two different types of supervision files are supported, being AASPI .H format mask files or ASCII format text file. More details about these two file types will be given in the later part of this section. The .H mask file is generated using utility ***aaspi_plot***, and the ASCII text file is generated using utility ***aaspi_make_training_clusters***. Use the button 2B to switch between these two types of supervision file. If ASCII file type is selected, the user can use button 3B to invoke ***aaspi_make_training_clusters*** and generate a supervision file. For how to use ***aaspi_make_training_clusters***, please refer to the documentation of utility ***aaspi_make_training_clusters***.

An AASPI .H format mask file is a 3D volume of facies labels and shares the same geometry as input seismic (attribute) volumes. Each sample point is assigned an integer number representing a facies, and the program ***gtm3d*** will pair the input seismic attribute with such facies label at the same spatial location to form a supervision dataset. An ASCII format text supervision file is a list of supervision vectors. If *N* is the total number of columns in that file, then the first *N-1* columns are input attributes (one attribute a column), and column *N* is the facies label. In this way, each row of this file represents an *N* dimensional data vector, which consists of *N-1* attributes and *1* facies label.

Then the corresponding supervision file should be provided using one of the Browses (4B), and fill in the number of facies (masks) at 5B. The reset button (6B) can be used to start over unsupervised GTM analysis.







The above two figures show the most likely occurrence of facies Type 1 and Type 2. The output files are names as *gtm_bhattacharyya_distance_mask_n_${project_name}_${suffix}*, where *n* is the facies label index. The magenta color highlights regions with the highest probability (90-100 %) of occurrence of the facies similar to the facies around an input well. The blue regions have very low likely that the facies is similar to the facies around the input well. And the black regions have no similarity to the facies type around the input well.

Creation of volume probes in Petrel using the latent space projections onto eigenvectors 1 and 2

The generated GTM projections can be then imported to any commercial seismic data interpretation packages for better visualization. In this document we use Petrel as an example for illustration purpose. The *gtm_axis1_latent_space_projection_${project_name}_${sufffix}* and *gtm_axis2_latent_space_projection _${project_name}_${suffix}* files are converted into .segy and are imported into Petrel. Generate a volume probe with these two volumes around the selected horizon as shown below. Crossplot utility which is common in most of the commercial software can be used to crossplot these two volumes and then the clusters on the latent space can be picked manually and colored which then can be simultaneously visualized in the seismic volume.

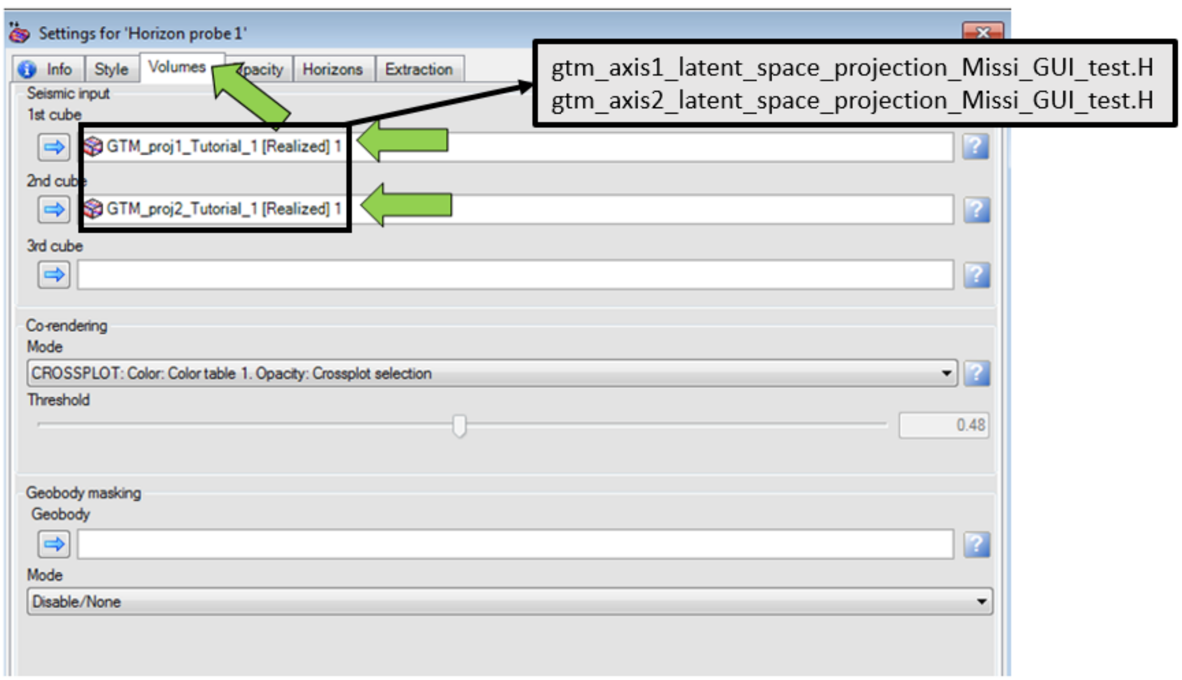


Figure above shows the crossplotting tool as it appears in Petrel. Note the file names have changed in the recent release and are shown in the gray box.

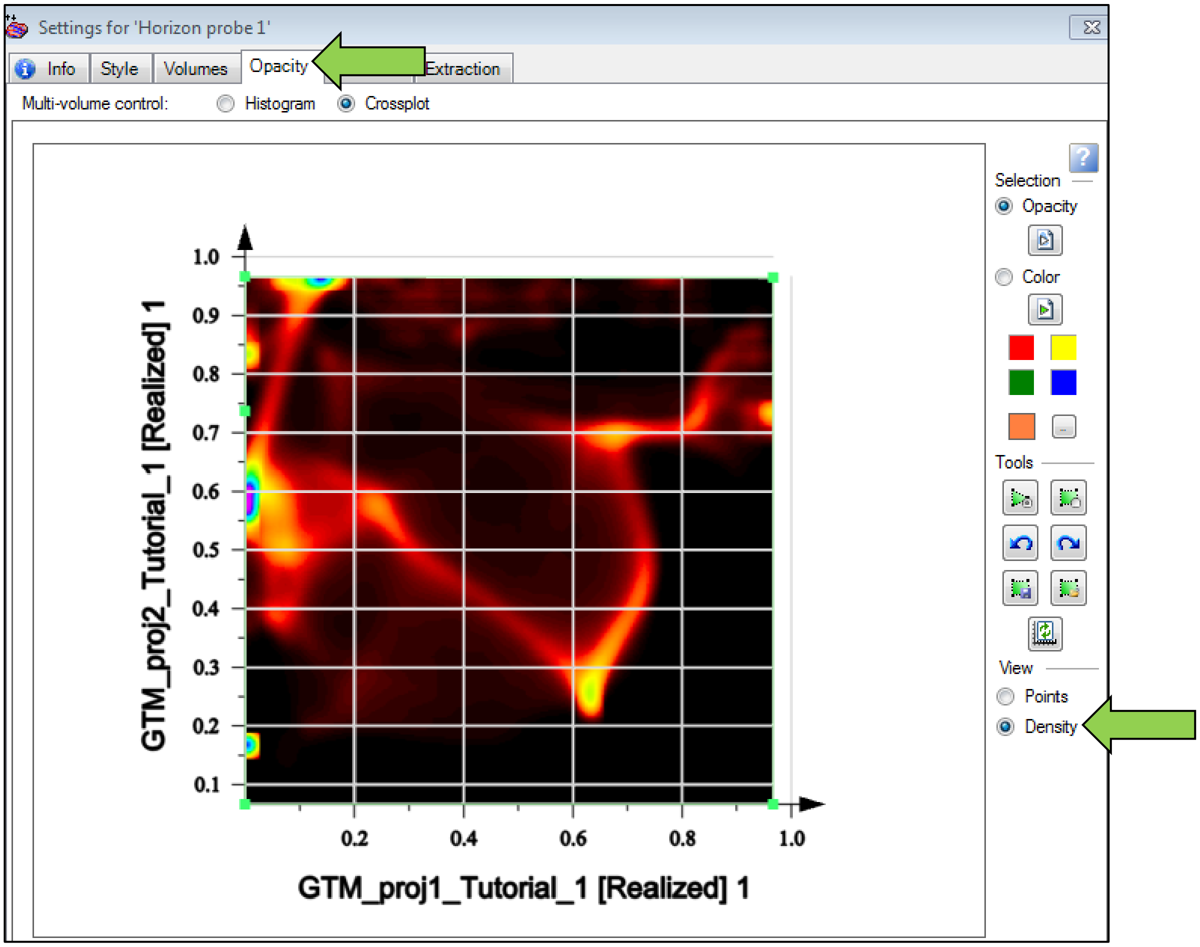


Figure 11. The pdfs on the latent space as it appears on Petrel's crossplot tool.

In Petrel's "*Opacity*" tab click on *Crossplot* to generate a 2D histogram of the two volumes as shown above. Also click on *density* as marked with the green arrow. The resulting image shows the mean distribution of the posterior probability (*responsibility*) projections of the *volume probe selected* on the 2D latent space. The "high-density" of the histogram appear in the crossplot as bright colors (light red, yellow, green, blue, and violet). Once visualized, the interpreter draws polygons around hypothesized clusters which are then plotted using Petrel's volume probe tool.

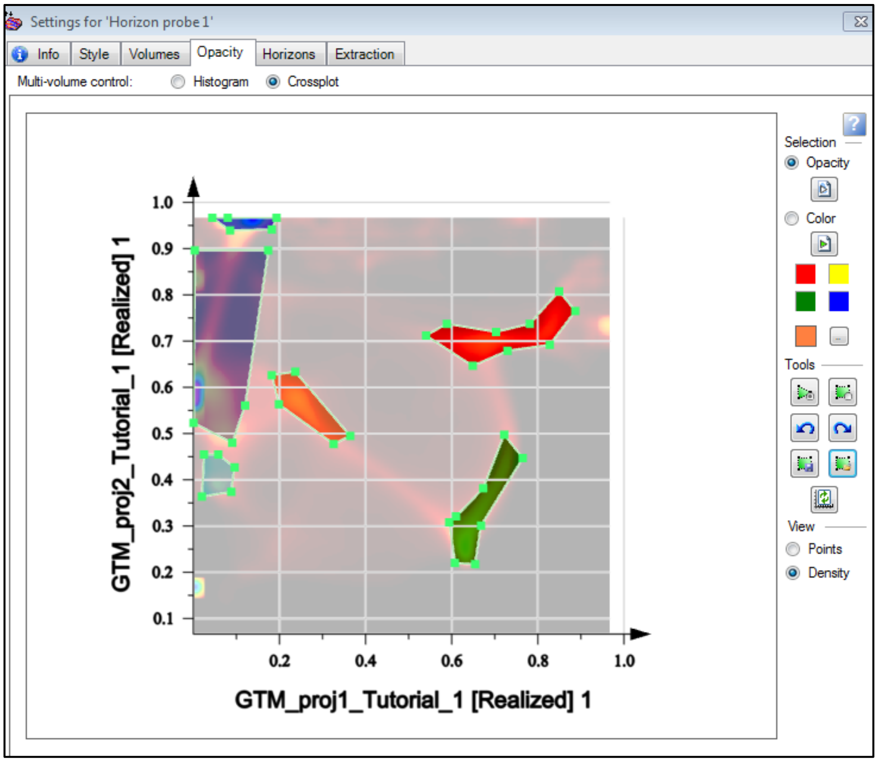


Figure 12. Interpreter-generated polygons displayed on top of the 2D historgram shown in the previous image. In this manner, the interpreter can select clusters of interest and see how they relate to well control or classic interpretation (e.g. salt bodies, mass transport complexes, gas-charged fans) in the 3D volume.

Pressing "Ctrl" key to select multiple polygons on the cross-plot. In the Figure 12 we have select six polygons representing most of the data distribution. Figure 13 shows a horizon slice along the top Mississippi Lime through the interpreter-generated clusters.

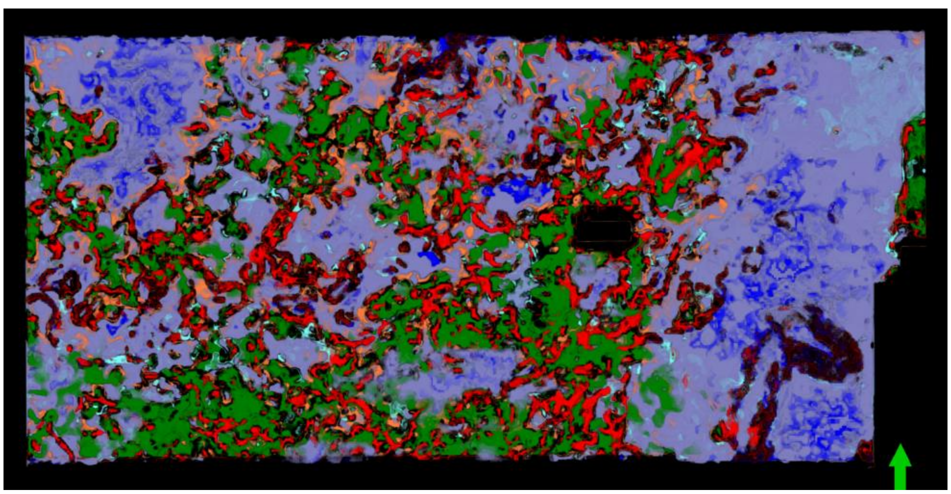


Figure 13. Interpreter generated clusters. Image logs indicate that the blue colors correspond to tight lime and tight chert while red and green correspond to porous tripolite and fractured chert.

Example 2: Veracruz Basin, Southern Mexico

The following example comes from a carbonate wash play in the Veracruz Basin, Mexico, described by Roy et al. (2014).

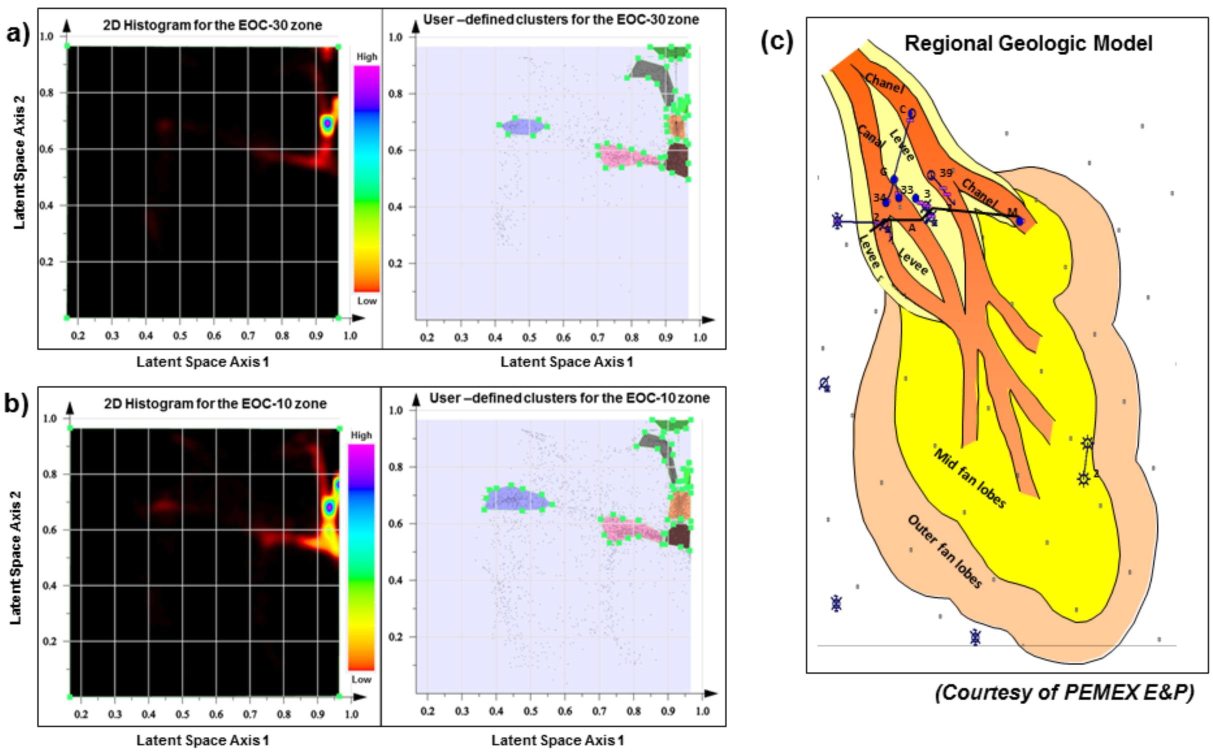


Figure 14. The 2D cross- plot of the mean posterior distribution map of the "*responsibilities*" of the data onto the 2D Latent space for the reservoir units EOC-30 and EOC-10. The cross plot is generated by cross plotting to GTM projection volumes. **(a)** The projection of the mean "*responsibilities*" of EOC-30 unit. The 2D histogram is on the right and the scatter crossplot is on the left. Seven clusters are visible on the latent pace corresponding to the high-density points. These clusters are delineated by polygons with different colors and in the subsequent figure will help to visualize the different classes in the seismic data. **(b)** The projection of the mean "responsibilities" of EOC-10 unit. The mineralogy content and porosity distribution for the EOC-10 and the EOC-30 reservoir units being similar the clusters for both of these reservoir units lie on the same location in the 2D latent space. They are also color-coded similarly since both reservoir units have similar rock type. **(c)** Regional conceptual sedimentary model (Roy, 2013).

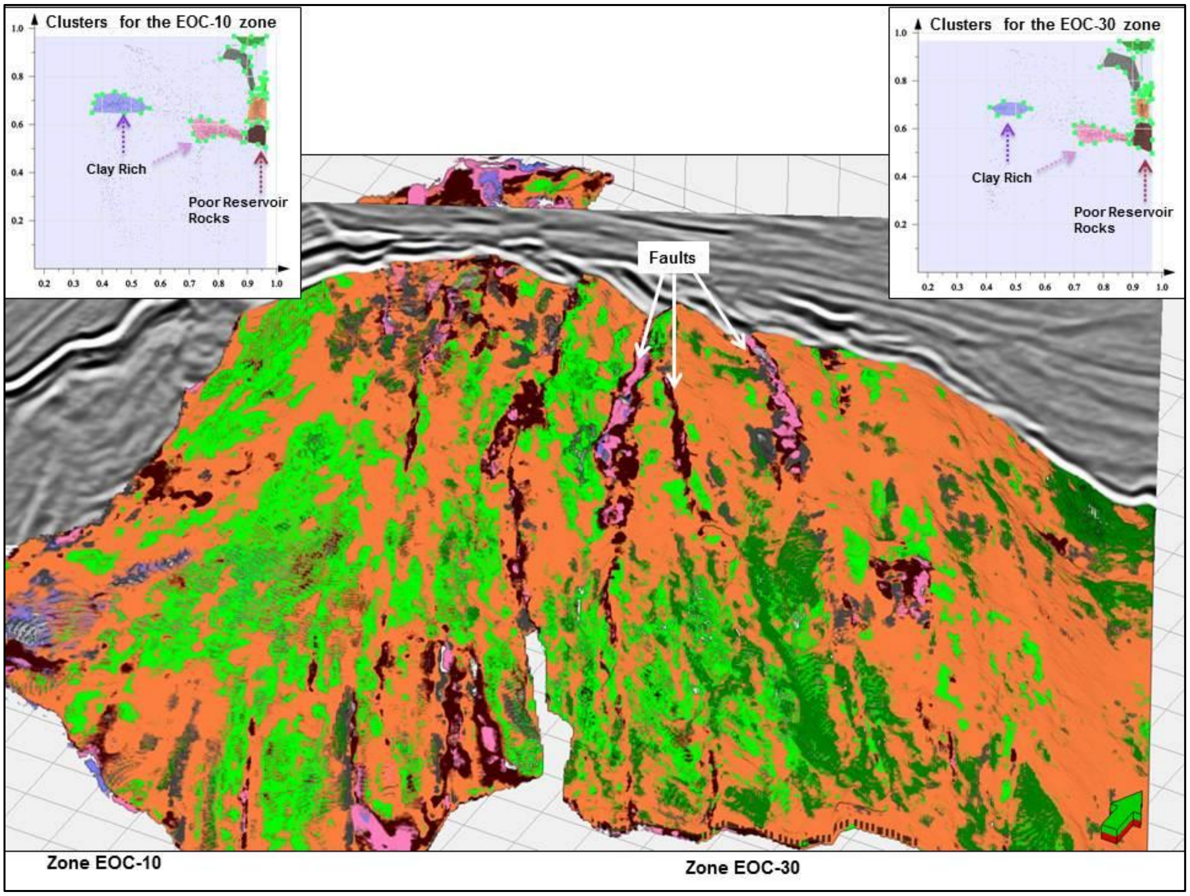


Figure 14. Generating the seismic facies volume (geobodies) from GTM clustering within the reservoir units EOC-10 and EOC-30, considering the input seismic volumes - lambda-rho (λρ) vs. mu-rho (μρ) and P-wave impedance (Zp) vs. Vp to Vs ratio (Vp/Vs). Different polygons around classes signify rock types for reservoir units **(a)** EOC-10 and **(b)** EOC-30. Seven different facies class have been identified from the clusters in the latent space and are delineated by polygons of different colors. **(c)** The horizon probe generated for the EOC -10 and the EOC-30 reservoir units after the unsupervised GTM analysis. The white arrows highlight the faults. The most abundant facies are the orange facies (Roy, 2013).

Example 3: Barnett Shale

This example comes from a paper by Roy (2013), applying GTM to the Barnett Shale play.

The inputs to our GTM algorithm are different seismic inversion volumes (P-impedance, lambda-rho, mu-rho) which help in understanding the highly heterogeneous nature of the Barnett shale. For the above attribute generations the seismic data between the Marble faults horizon and the Viola limestone is considered. The impedance volumes better reflect a heterogeneous shale reservoir based rock type.

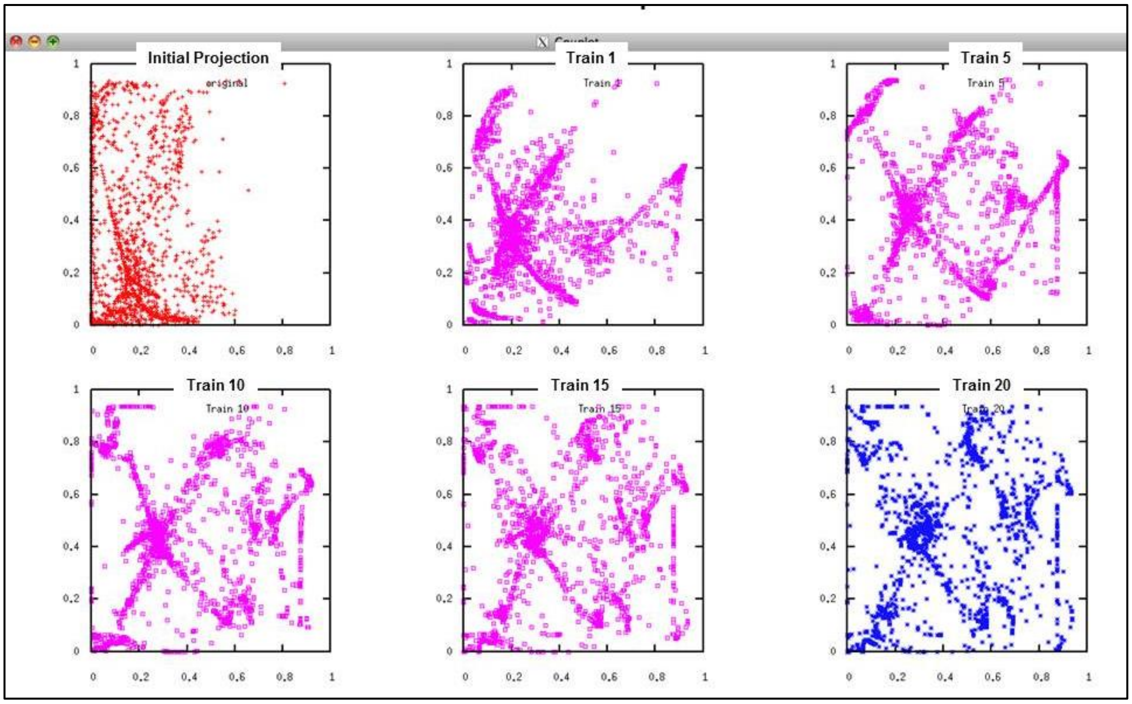
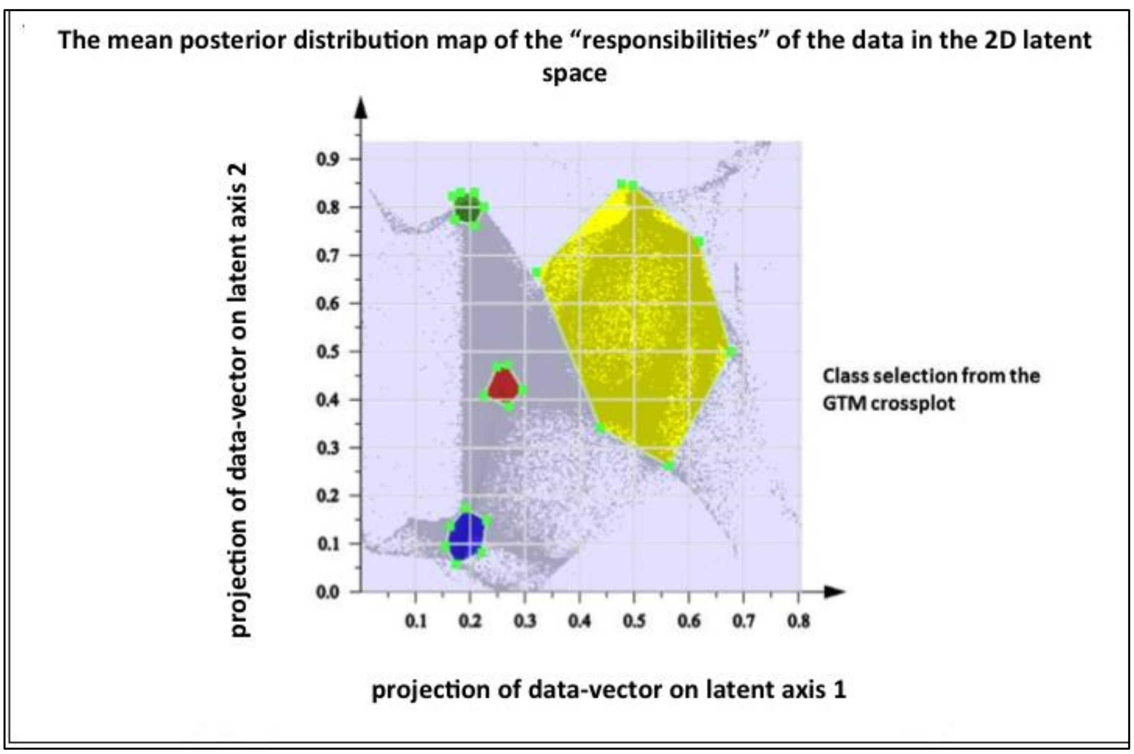


Figure 15. The mean posterior distribution map of "responsibilities" of the training data in 2d latent space. Here, the first graph shows initial projection, on the two projection axis. Here, you can see that the separation in the data increases and data is getting clustered.



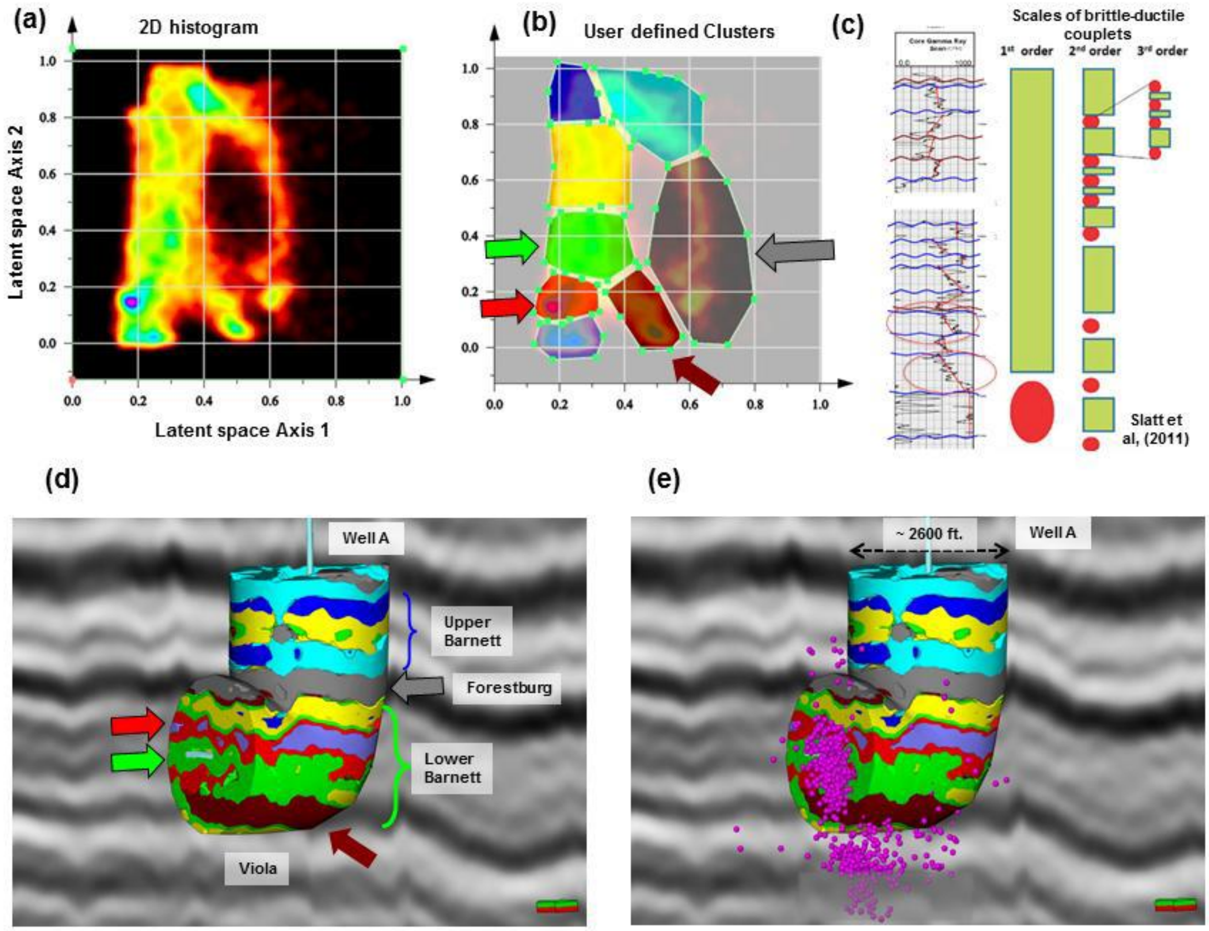


Figure 5.13. Well-probes for Well A generated by cross-plotting the two GTM projection volumes. **(a)** The 2D histogram generated from the cross-plot of the two GTM projection volumes. **(b)** Eight user-defined polygons drawn around the clusters seen in (a). **(c)** Brittle-ductile couplets proposed by Slatt et al, (2011). **(d)** Well-probe data colored by the clusters selected in (b). The Upper Barnett, the Lower Barnett exhibit a different cluster composition and are in turn different from the intervening Forestburg Limestone (in gray). **(e)** The microseismic events from this well are plotted along with the well-probe. Note the microseismic events are more localized in the red and light green facies and misses the brown facies, thus the red and light green facies *125* are interpreted as brittle and brown facies to be ductile. The results are consistent with the 2nd order brittle-ductile couplets proposed by Slatt et al, (2011).

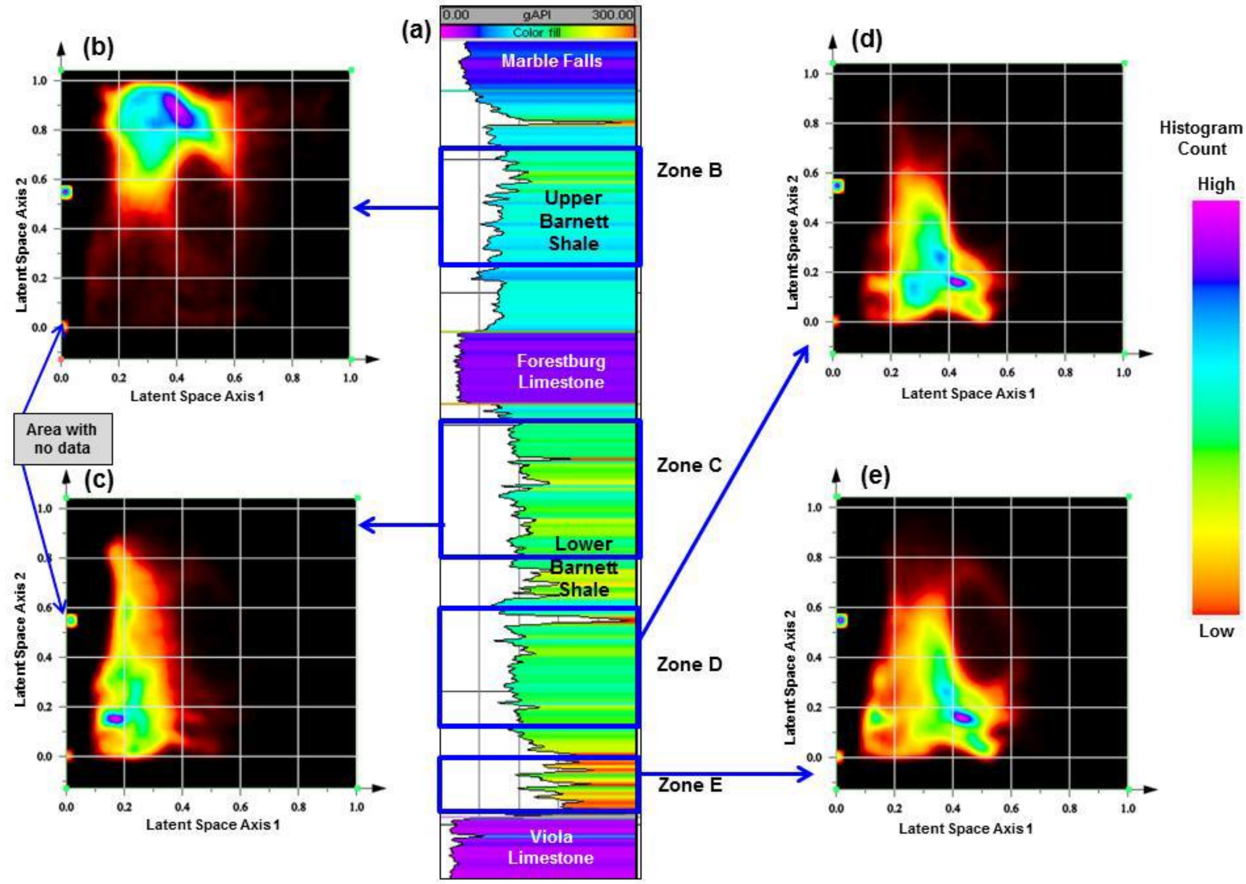


Figure 16. Four different zones within the Barnett Shale selected using a gamma ray log from a well within the survey. The corresponding 2D histograms of the mean posterior probability projections for **(b)** the Upper Barnett (zone B), **(c)** the top of the Lower Barnett (zone C), **(d)** the middle of the Lower Barnett (zone D) and **(e)** the bottom of the Lower Barnett Shale (zone E) are shown.

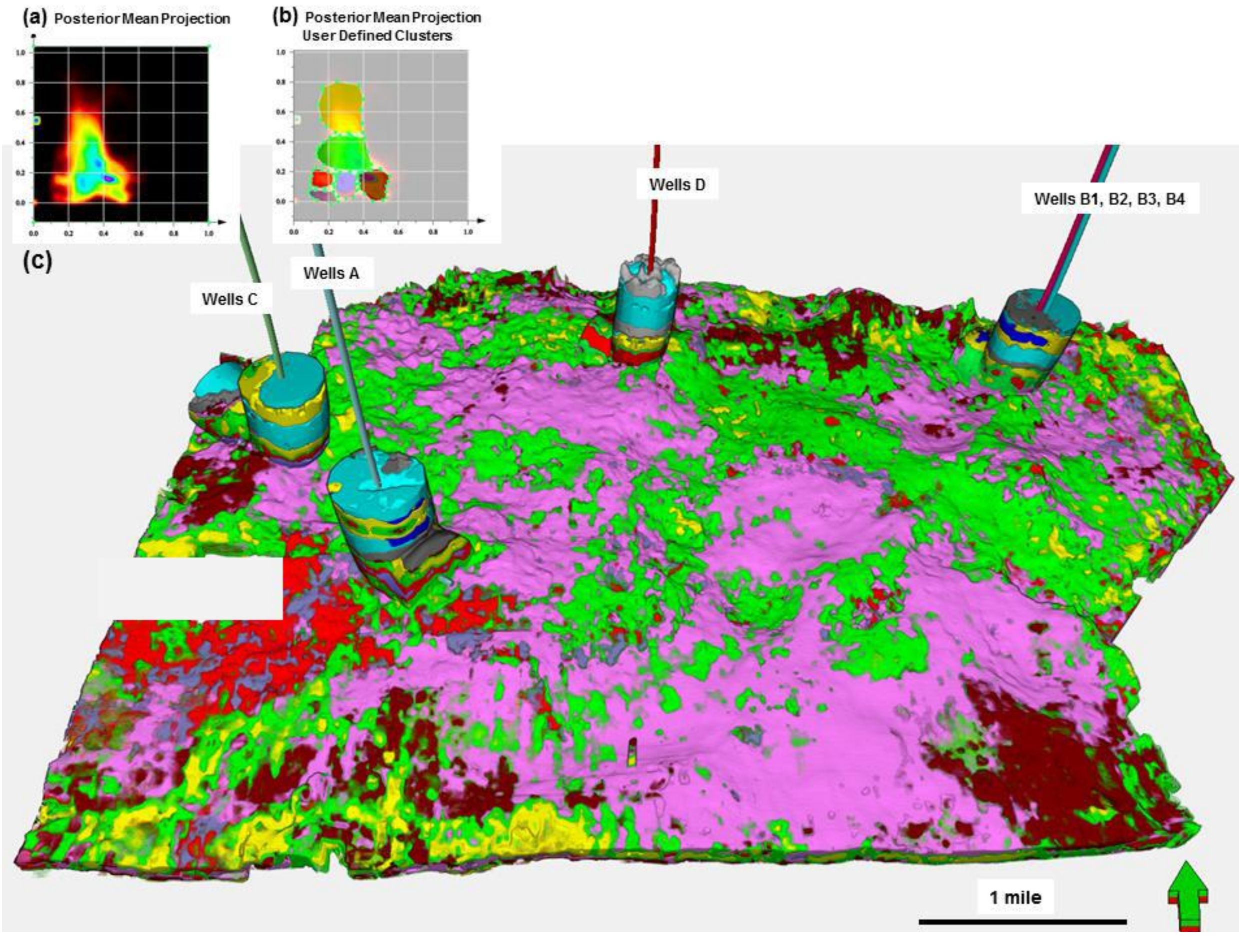


Figure 17: 2d histogram of zone D defined above and below by stratal slices corresponding to the well log picks in Figure 5.15a. **(b)** User defined polygons are created and are colored consistent with the well-probes in Figures 5.13 and 5.14. **(c)** The facies volume probe of middle section of the Lower Barnett Shale zone D visualized along with the well-probes with the colors selected according to the clusters in (b). Note that this zone has least similarity to the Upper Barnett Shale. With more of the microseismic events concentrated in the pink, light green and red facies as seen in the well-probes, and the dominance of siliceous non-calcareous shale lithofacies (Singh, 2008), this zone 3 is interpreted as brittle with results consistent with Perez (2013).

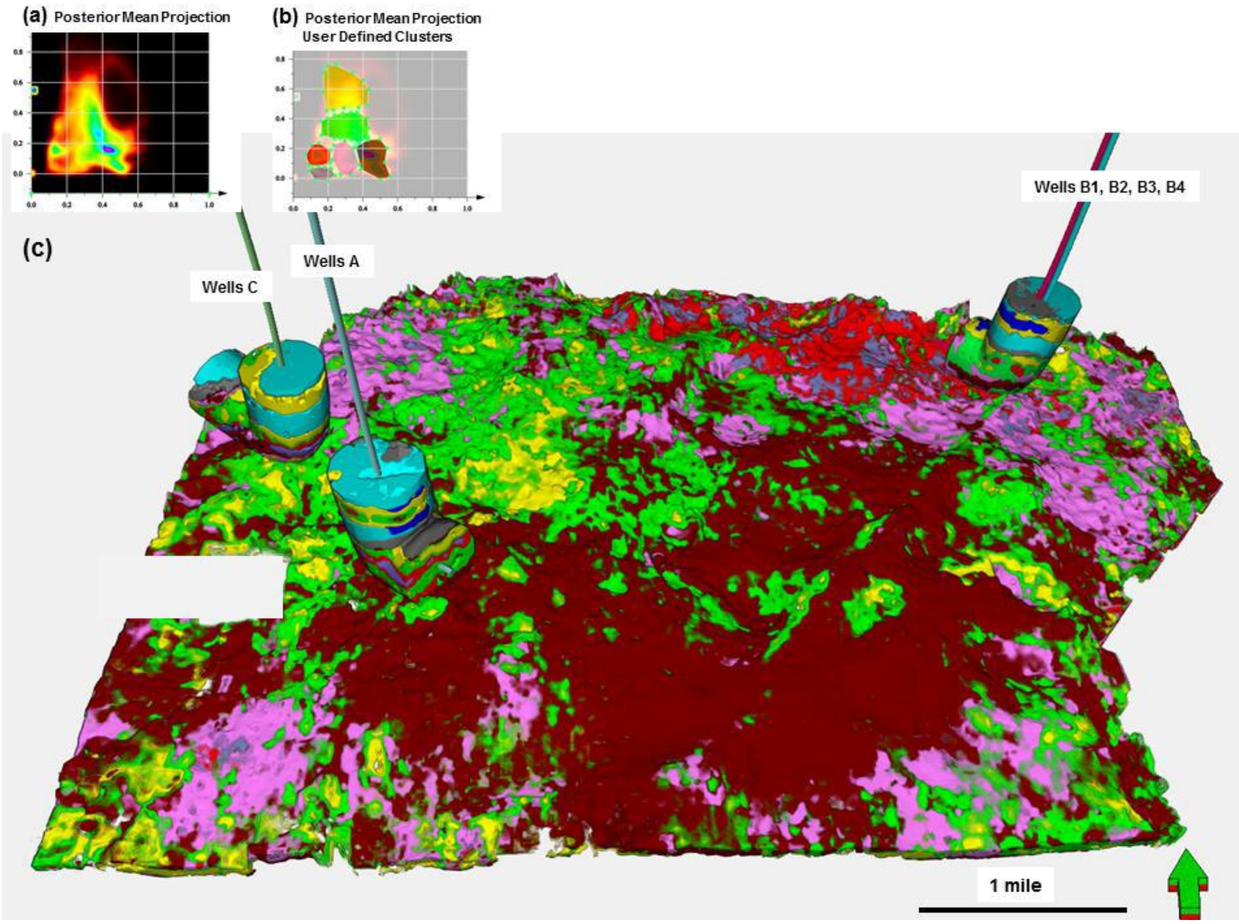


Figure 18. 2d histogram of zone E defined above and below by stratal slices corresponding to the well log picks in Figure 5.15a. **(b)** User defined polygons are created and are colored consistent with the well-probes in Figures 5.13 and 5.14. **(c)** The facies volume probe of bottom section of the Lower Barnett Shale zone E visualized along with the well-probes with the colors selected according to the clusters in (b). This zone corresponds to the hot gamma ray zone (Pollastro et al., 2007). Six clusters are also identified from the mean posterior probability projections (in the top inset) are polygons are drawn and are colored consistent with the well-probes. With very few of the microseismic events in the brown colored facies we interpret from (Singh, 2008 and Perez 2013) the brown colored rock to be ductile and high in TOC content. The pink, light green and red facies are the regions with brittle shale.

References

Bhattacharyya, A., 1943, On a measure of divergence between two statistical populations defined by their probability distributions: Bulletin of the Calcutta Mathematical Society, **35**, 99–109.

Bishop, C. M., M. Svensen, and C. K. I. Williams, 1998, The generative topographic mapping: Neural Computation, **10**, No. 1, 215-234.

Bishop, C. M., M. Svensen, and C. K. I. Williams, 1998, Development of the generative topographic mapping: Neurocomputing, **21**, No. 1, 203-224.

Matos, C. M., K. J. Marfurt, and P. Johann, 2009, 2D color display of Self Organizing Maps: 79th Annual International Meeting Society of Exploration Geophysicists, Expanded Abstracts.

Roy, A., V. Jayaram, J.T. Kwiatkowski, and D. Devegowda, 2012, Learning from past experience in shale gas reservoirs: A predictive machine-learning tool based on a generative topographic model to estimate EURs and well productivity: Submitted to 2013 SPE Reservoir Simulation Symposium.

Perez, Altamar, R., 2013, Correlation of surface seismic measurements to completion quality: Application to the Barnett Shale, PhD dissertation.

Roy, A., Araceli S. Romero-Peláez, Tim J. Kwiatkowski, Kurt J. Marfurt , 2014, Generative topographic mapping for seismic facies estimation of a carbonate wash, Veracruz Basin, southern Mexico , Interpretation, **2**, No. 1, pp. SA31-SA47.

Slatt, M. R., and Y. Abousleiman, 2011, Merging sequence stratigraphy and geomechanics for unconventional gas shales. The Leading Edge, 274-282.

Singh, P., 2008, Lithofacies and Sequence Stratigraphic Framework of the Barnett Shale, Northeast Texas: Ph.D. dissertation, University of Oklahoma.

Strecker, U., and R. Uden, 2002, Data mining of 3D poststack attribute volumes using Kohonen self-organizing maps: The Leading Edge, **21**, 1032-1037.

Wallet, B. C., M. C.de Matos, J. T. Kwiatkowski, and Y. Suarez, 2009, Latent space modeling of seismic data: An overview: The Leading Edge, **28** , 1454-1459.