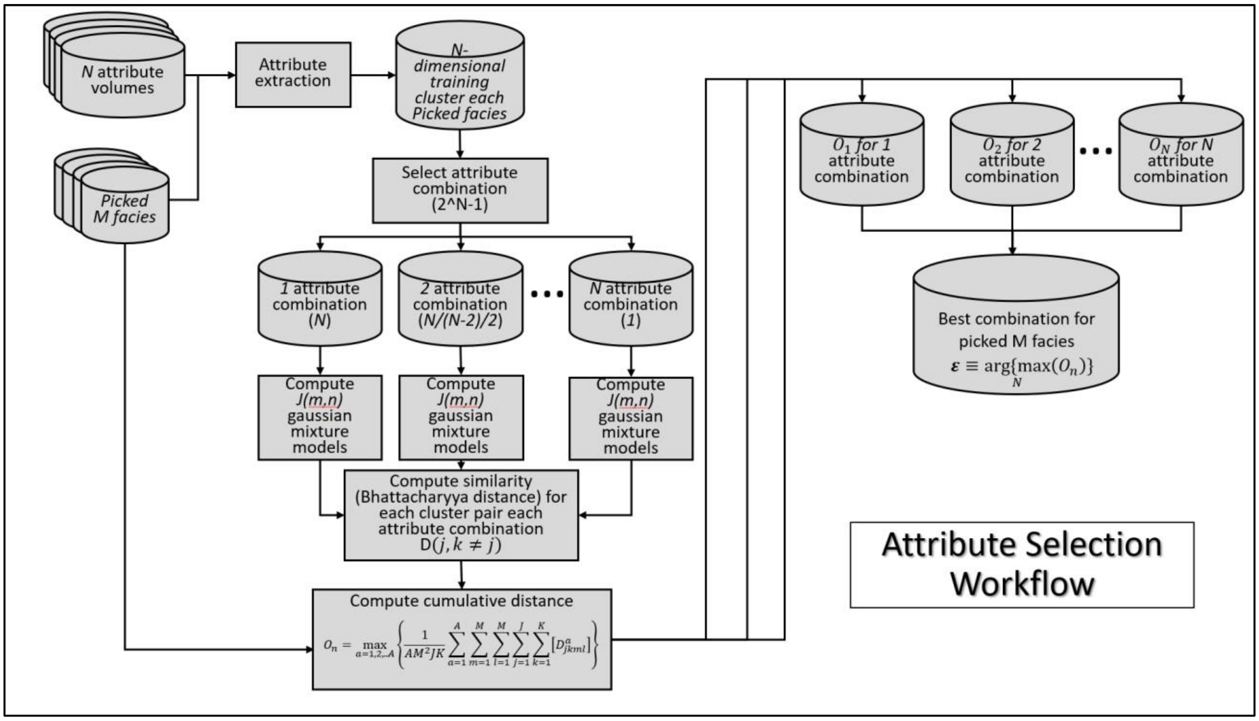
ATTRIBUTE SELECTION FOR 3D SEISMIC FACIES ANALYSIS – PROGRAM attribute_selection

Contents

Computation flow chart

Interpreters face two main challenges in seismic facies analysis. The first challenge is for a human interpreter to define, or "label", the facies of interest. Accurately defining the 3D extent of a given seismic facies takes an understanding of geologic processes and the limits of seismic acquisition, processing, and imaging. Machine learning is based on accurate training data, in which this application is provided by a skilled interpreter defining polygons about facies of interest. The second challenge is to select a suite of attributes that can differentiate a target facies from the background reflectivity. Unfortunately, there are relatively few interpreters who possess both a deep understanding of the geology of a given exploration play and a deep understanding of the sensitivity of an ever-expanding collection of seismic attributes to geology.

This GMM-based attribute selection program is a tool to select the best attribute combinations from input candidate attributes. The GMMs use PDFs to represent rather than to discriminate between facies. Gaussian mixture models are based on probability theory, and by construction, provides a posterior probability that any particular voxel belongs to a given mixture model.



The Workflow illustrates the steps used in our GMM-based attribute selection. The workflow starts with the facies of interest picking and extraction of training voxels from the *N* candidate attributes. The GMM clusters are computed for each facies attribute combination. Then, the Bhattacharyya distance is computed for each cluster pair that is in the same attribute dimension. The winning attribute combination in each attribute dimension needs to calculate the average cumulative distance for the comparison of attribute combinations that are in different attribute dimensions. The best attribute combination is the one has the highest average cumulative distance.

Output file naming convention

Program **attribute_selection** will always generate the following output files:

Output file description	File name syntax
program log information	attribute_selection_*unique_project_name_suffix*.log
program error/completion information	attribute_selection_*unique_project_name_suffix*.err

where the values in red are defined by the program GUI. The errors we anticipated will be written to the **.err* file and be displayed in a pop-up window upon program termination. These errors, much of the input information, a description of intermediate variables, and any software trace-back errors will be contained in the **.log* file.

Theory

The GMMs use PDFs to represent rather than to discriminate between facies. Gaussian mixture models are based on probability theory, and by construction provide a posterior probability that any particular voxel belongs to a given mixture model. In statistics, a multivariate distribution of data vector $\mathbf{x}_i$ on the parameters ψ modeled by Gaussian mixture models is,

$$p(\mathbf{x}_i|\psi) = \sum_{k=1}^{K} \varphi_k f\,(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1}$$

where $\boldsymbol{\mu}_k$ is the mean and $\boldsymbol{\Sigma}_k$ is the covariance matrix for the multivariate case. φ_k is the weight, which is that

$$\sum_{k=1}^{K} \varphi_k = 1. \tag{2}$$

The multi-dimensional Gaussian mixture probability function is,

$$f(\mathbf{x}_i \mid \mu_k, \boldsymbol{\Sigma}_k) = \frac{1}{|\boldsymbol{\Sigma}_k|^{\frac{1}{2}}(2\pi)^{\frac{N}{2}}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\right), \tag{3}$$

where the symbol $^{\mathrm{T}}$ indicates the transpose of a matrix. *N* is the number of candidate attribute, where *i=1 … N*. In n-dimensional attribute space, the k*th* PDF is defined by its mean $\boldsymbol{\mu}_k$ and its covariance matrix, $\boldsymbol{\Sigma}_k$. Hardisty (2017) showed how to compute the optimum number of GMMs to represent multiattribute data in a seismic survey, where the objective is to determine if different seismic facies naturally clump into different areas of n-dimensional space, allowing them to be color-coded and displayed.

Similar to Hardisty (2017), in our application, we will use the k-means clustering to generate initial clustering models. We apply 300 iterations of a refinement technique to cluster attribute vectors into an attribute space defined by the first two eigenvectors of *N*-by-*N* covariance matrix. Thus, k-means technique results the initial multivariate means $\boldsymbol{\mu}_k^i$ and the covariance matrices $\boldsymbol{\Sigma}_k^i$, and the weights φ_k^i is defined by the fraction of attribute vectors assigned to each k-means cluster. Because the number of component *K* is known, expectation maximization (EM) is able to be used to determine the best model's parameters. Classification Expectation-Maximization (CEM) as an alternative EM algorithm is better to classify data vectors, when data vectors are assumed to be generalized into a single Gaussian mixture model (Celeux and Govaert, 1992; Hardisty, 2017). K-means clustering is a popular cluster analysis for data mining, which is able to find clusters of comparable spatial extent. In general, Expectation-maximization algorithm for mixture Gaussian distributions involves both K-means and Gaussian mixture models. The conjunction of the Classification Expectation-Maximization (CEM) and the Stochastic Expectation-Maximization (SEM) is employed to learn mixture parameters in this paper. Like the conventional EM algorithm, both two algorithms require to define a partition of the input data, then compute the posterior probability according to the responsibility matrix. Each element w_{ik} (the posterior probability) of the *N*×*M* responsibility matrix is given by

$$w_{ik} = \frac{\varphi_k f(\mathbf{x}_i \mid \mu_k, \boldsymbol{\Sigma}_k)}{p(\mathbf{x}_i|\psi)}. \tag{4}$$

Accumulating the responsibility matrix, the CEM creates K-Partitions by assigning each data component to the cluster that provides the highest posterior probability according to the responsibility matrix. For each cluster, the mixture parameters are updated by the respective partition that associating with the maximum log-likelihood estimates, which is defined as

$$L(\psi) = \sum_{k=1}^{K} \sum_{i=1}^{N} \mathrm{z_{ik}} \log\{\varphi_k f(\mathbf{x}_i \mid \mu_k, \boldsymbol{\Sigma}_k)\}, \tag{5}$$

where $\mathrm{z_{ik}}$ is an indicator that is equal to 1 only if the data vector $\mathbf{x}_i$ belongs to cluster k (Hardisty, 2017). The SEM algorithms randomly assigns partitions to a cluster associated with the posterior probabilities in the responsibility matrix, which will help avoid sub-optimal solutions provided by the CEM. Thus, we use 200 iterations of the SEM to initialize the CEM algorithm that gives a final partition and GMM. The parameterizations of each covariance matrix associated with each Gaussian mixture models can be controlled so that resulting in reduction of the number of parameters. We consider nine modules of covariance matrices that introduced by Celeux and Govaert (1993) and Hardisty (2017), and use the Bayesian Information Criterion (BIC) proposed by Schwarz (1978) to compare models of differing complexity. The BIC is defined as:

$$BIC = \log\big(L(\psi)\big) - \frac{1}{2}\varepsilon \log(\alpha), \tag{6}$$

where ε is the number of estimated parameters and α is the number of training voxels. The higher BIC value indicates more confidence on covariance parameterizations.

As before, some insight into the attribute expression of a given facies or “what works” reduces the number of combinations to be evaluated. For each collection of attributes (*n*=2, *n*=3, *n*=4, …) we generate *n*-dimensional GMMs of each user-defined seismic facies. We then multiply the Gaussian mixture model for each facies against the others, summing the results. The attribute selection that provides the largest summed distance (or least overlap) is the best combination for that value of n. We then validate this combination in predicting facies not used in constructing the original GMMs. Next, we increase *n* to determine if we can significantly increase the amount of overlap distances (or confidence) by increasing the dimensionality of the problem.

In the workflow, interpreters only need to define *M* facies of interests and *N* candidate attributes for those facies. We pick *M* facies by drawing a suite of polygons on time slices and vertical slices of the seismic amplitude images, and each facies can be represented by multiple polygons. Therefore, there are *N**M* individual volumes that represent the *N* candidate attributes and the *M* facies, which will be inputted into our workflow. The workflow iteratively selects different number of attributes from the *N* candidate attributes. When only considering 1 selected attribute, there are *N* probabilities of attribute combinations. For 2 selected attributes, there are *N!/(N-2)!/2!* attribute combinations. The total combination number of different selected attributes from the *N* candidate attributes will be $2^N - 1$. We compute GMMs $[\varphi_k, \mu_k, \boldsymbol{\Sigma}_k]$ of picked voxels to represent supervised voxels of each attribute combination and each facies. Although interpreters can accurately draw one facies by one or multiple polygons, the facies like MTDs and channels may contain multiple GMMs. For this case, we need to setup a maximum cluster number, which is equal to 2 in our two applications.

After computing GMMs for each facies each attribute combination, we employ the Bhattacharyya Distance (Mak and Barnard, 1996) to measure the similarity between each cluster. For two GMMs *j* and *k* residing in *n*-dimensional attribute space, the distance D_{jk}^n between them is:

$$D_{jk}^n = \frac{1}{8}\left(\boldsymbol{\mu_k} - \boldsymbol{\mu_j}\right)^T \left[\frac{\boldsymbol{\Sigma_k}+\boldsymbol{\Sigma_j}}{2}\right]^{-1} \left(\boldsymbol{\mu_k} - \boldsymbol{\mu_j}\right) + \frac{1}{2}\ln\frac{\left|\frac{\boldsymbol{\Sigma_k}+\boldsymbol{\Sigma_j}}{2}\right|}{\sqrt{|\boldsymbol{\Sigma_k}||\boldsymbol{\Sigma_j}|}}, \tag{7}$$

where $\boldsymbol{\mu_k}$, $\boldsymbol{\mu_j}$ and $\boldsymbol{\Sigma_k}$, $\boldsymbol{\Sigma_j}$ are the mean and covariances of the GMM cluster *k* and *j*. The measured distance overall expresses the properties that include the difference of size, shape, distance in the GMM space, and the larger the value, the more dissimilarity between two clusters. The similarity of all cluster pairs is measured by the Bhattacharyya Distance. Because the Bhattacharyya Distance can only work for the cluster pairs that are in the identical dimensional space (n-dimensional attribute space), other GMM clusters (in n-2, n-1, n+1, n+2, … dimensional attribute space) will be unavailable to be compared with the GMM clusters in n-dimensional or each other. Therefore, we define a new distance that measures average cumulative distance of each attribute combination,

$$O_{a_n}^n = \max_{A_n}\left\{\frac{1}{A_n M^2 JK}\sum_{a=1}^{A_n}\sum_{m=1}^{M}\sum_{l=1}^{M}\sum_{j=1}^{J}\sum_{k=1}^{J}\left[D_{mljk}^n\right] \mid n = 1, 2, \dots, N\right\}, \tag{8}$$

where M is the number of the picked facies, and A_n is the number of probable numbers of attribute combinations within *n* selected attributes (in *n*-dimension). *J* is the number of GMM clusters about A_n attribute combinations in all facies, respectively. The index *n* is from 1 to *N*, where *N* is the number of input candidate attributes. Next, we will find the optimum attribute combinations in each *n* selected attribute combination by comparing the value of the average cumulative distance. The optimum attribute combinations associated with 1, 2, …, *N* selected attribute space are as O^1, O^2, …, O^N. Then, we will find the most optimum attribute combination $\boldsymbol{\varepsilon}$ for the picked *M* facies by:

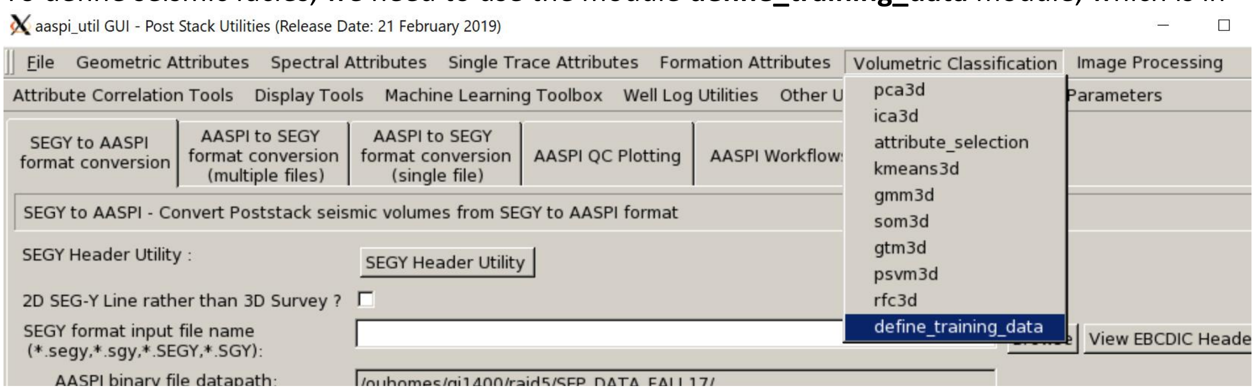
$$\boldsymbol{\varepsilon} \equiv \arg\{\max_N\left(O_{a_n}^n\right)\}, \tag{9}$$

Which means that the average cumulative distance of the most optimum attribute combination $\boldsymbol{\varepsilon}$ will be the highest among all attribute combinations.

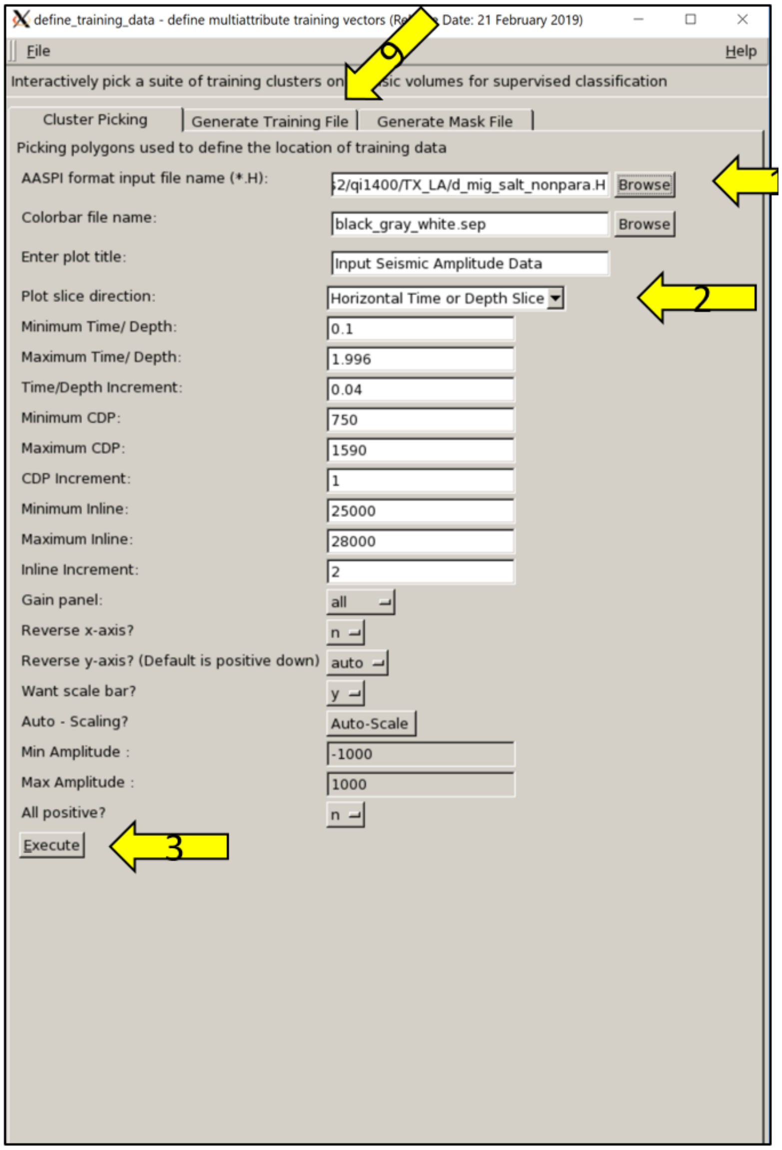
Application of attribute_selection module for seismic facies analysis

We first picked facies of interest by drawing multiple polygons on a seismic amplitude volume. The candidate seismic attributes are selected based on the interpreter's experience. The training voxels are then extracted on the picked polygons from all candidate seismic attributes. Next, we compute GMM clusters for each picked facies for each possible attribute combination, then compute the Bhattacharyya distance to measure the similarity between each cluster under the attribute combinations that are in the same dimension of attribute space. The higher Bhattacharyya distance between each facies cluster indicates the more easily the pair of clusters are separable. To evaluate attribute combinations between different dimension space, we define the average cumulative distance. In different numbers of attribute spaces, there is one optimum attribute combination, and the attribute combination that has the highest average cumulative distance is the best combination among all possible attribute combinations. Next, we filter the selected seismic attributes through the proposed 3D adaptive Kuwahara filter to suppress the effects of seismic noise, smoothen interior textures, and sharpen the edges of seismic facies. The filtered attributes are then mapped onto the GTM latent space to generate unsupervised PDFs of all voxels. The picked voxels of each facies associated with the selected attributes are then mapped onto the latent space to generate supervised PDFs of training voxels. Then, we compute the likelihood between training and all voxels, which results in probability volume of training facies.

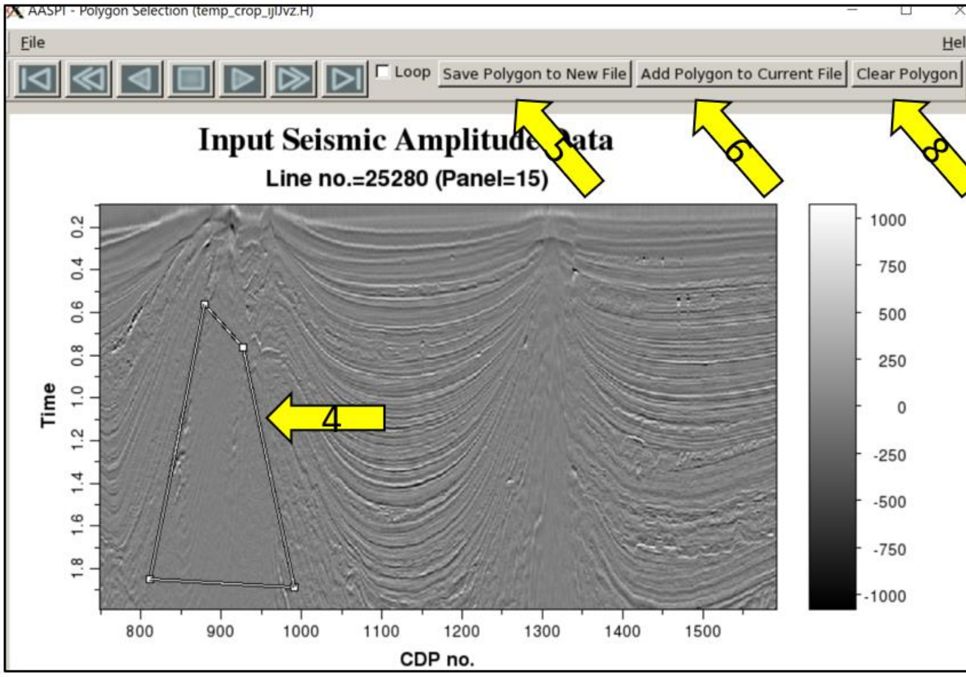
To define seismic facies, we need to use the module **define_training_data** module, which is in

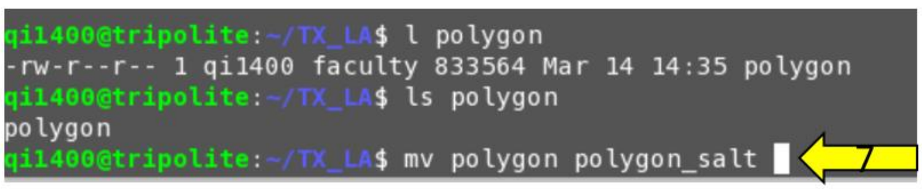


Click **define_training_data** and the program will be displayed (see next page).

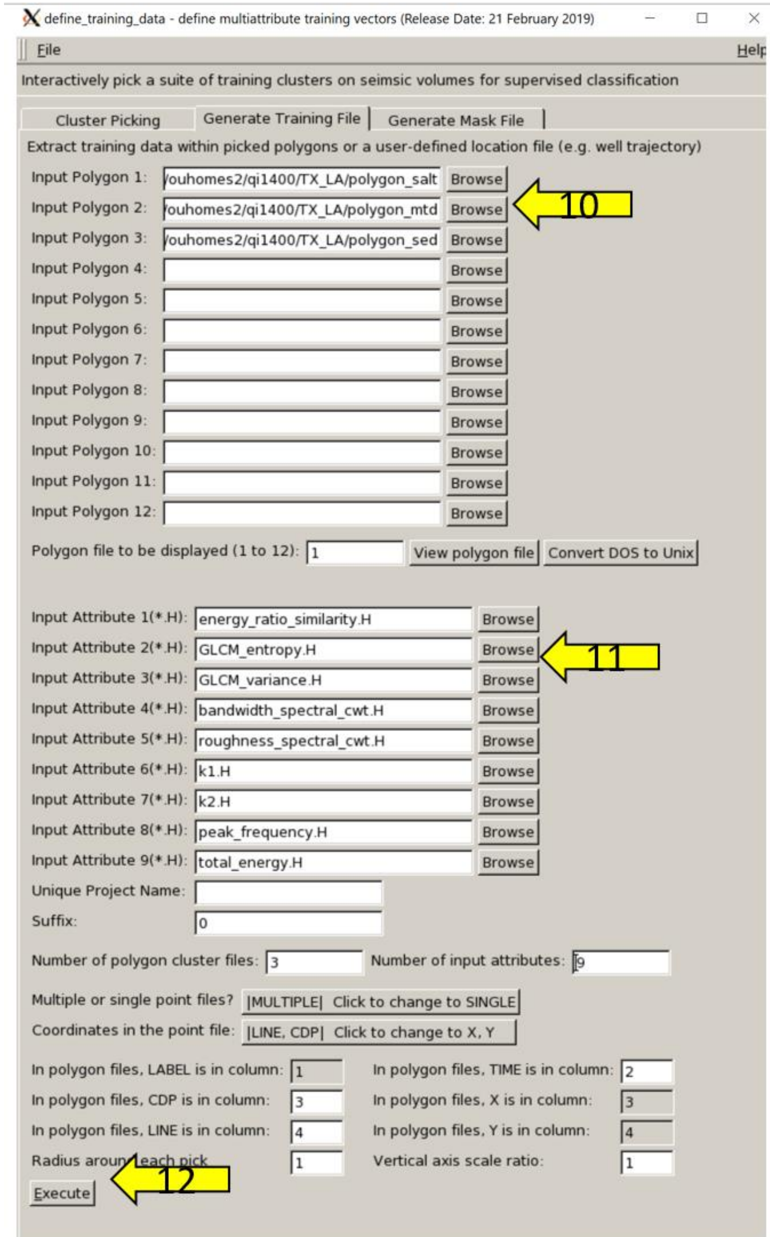


In using **define_training_data**, we can use seismic amplitude or attribute volume to draw polygon. (1) select seismic amplitude or attribute volume; (2) we can either plot a time slice or a vertical slice to draw a polygon; if we want to change plot direction, click (3) execute, then a plot will display.





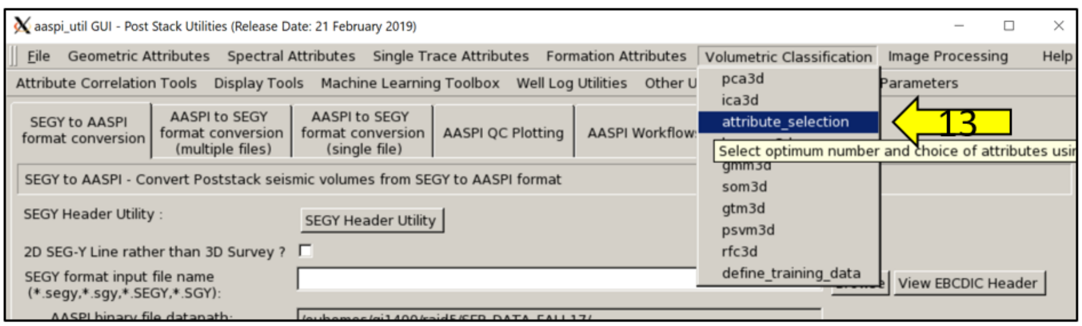
We plot the inline slice. (4) after figure plotting, we can draw polygons on the facies we wish to pick; (5) after the first polygon drawn on a slice, we need to click **save_polygon_to_new_file** to save this polygon to a new file; for other polygons drawn on the same slice or other slice, we only need to click (6) **add_polygon_to_current_file** to save picked voxels. After picking the first facies through slices, we need to (7) change the polygon name in the project folder, figure shows the case in Linux system, and we change the default name "polygon" to be "polygon_salt". Before picking another facies, we need to click (8) clear_polygon, then we draw polygons on other facies. By clicking (9) at the **define_training_data** module, the window will be changed to the following image (see next page).



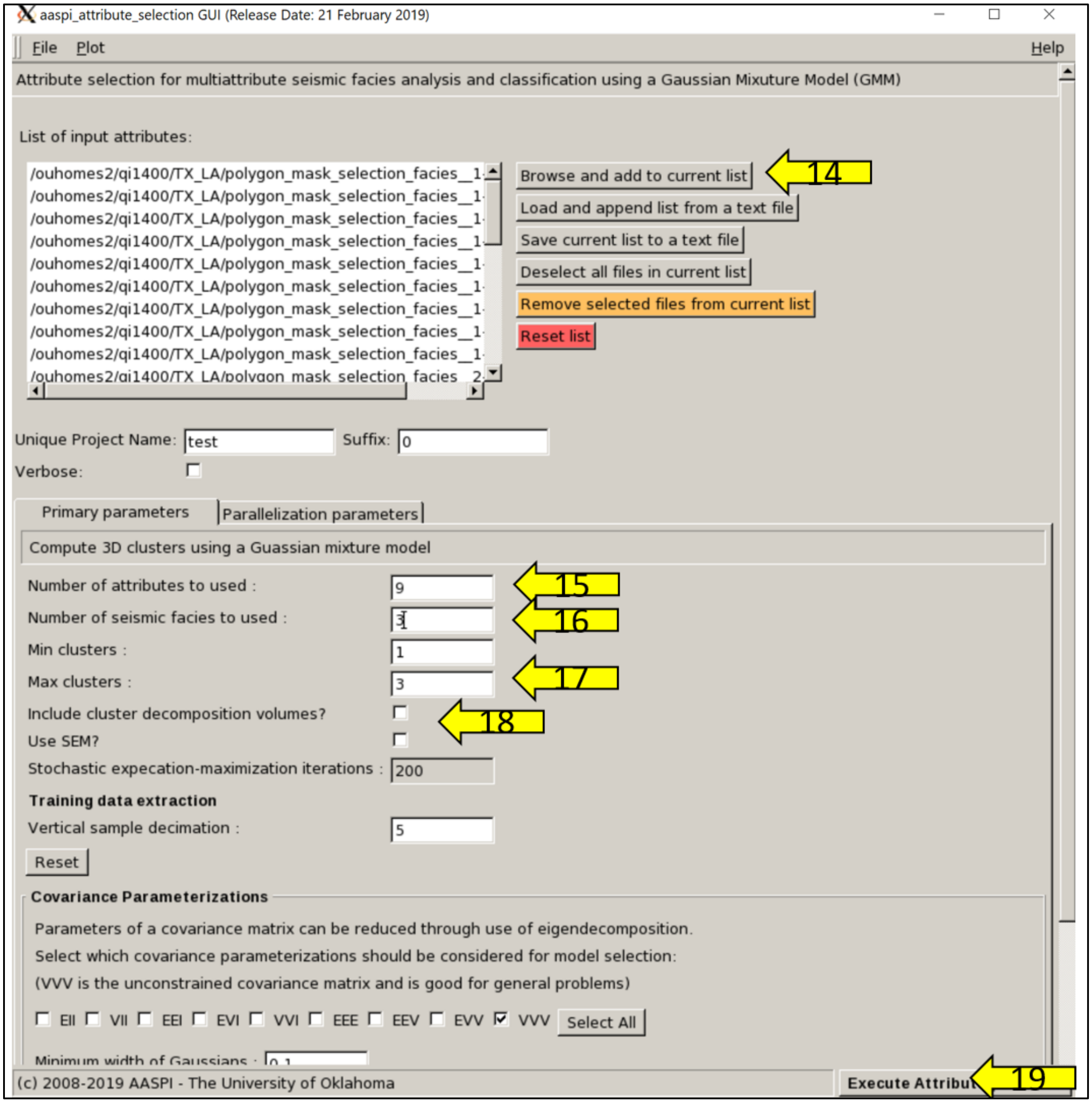
(10) Input picked facies (salt, mtd, background sediment); (11) input candidate attributes; please remember the input orders of facies and attributes, this information will be used in attribute selection. After inputing, click (12) execute.

```
polygon_mask_selection_facies__1-attr__1.H  polygon_mask_selection_facies__2-attr__6.H
polygon_mask_selection_facies__1-attr__2.H  polygon_mask_selection_facies__2-attr__7.H
polygon_mask_selection_facies__1-attr__3.H  polygon_mask_selection_facies__2-attr__8.H
polygon_mask_selection_facies__1-attr__4.H  polygon_mask_selection_facies__2-attr__9.H
polygon_mask_selection_facies__1-attr__5.H  polygon_mask_selection_facies__3-attr__1.H
polygon_mask_selection_facies__1-attr__6.H  polygon_mask_selection_facies__3-attr__2.H
polygon_mask_selection_facies__1-attr__7.H  polygon_mask_selection_facies__3-attr__3.H
polygon_mask_selection_facies__1-attr__8.H  polygon_mask_selection_facies__3-attr__4.H
polygon_mask_selection_facies__1-attr__9.H  polygon_mask_selection_facies__3-attr__5.H
polygon_mask_selection_facies__2-attr__1.H  polygon_mask_selection_facies__3-attr__6.H
polygon_mask_selection_facies__2-attr__2.H  polygon_mask_selection_facies__3-attr__7.H
polygon_mask_selection_facies__2-attr__3.H  polygon_mask_selection_facies__3-attr__8.H
polygon_mask_selection_facies__2-attr__4.H  polygon_mask_selection_facies__3-attr__9.H
polygon_mask_selection_facies__2-attr__5.H
```

Then, supervised 1D volumes within picked facies associated with different candidate attributes are generated as shown in the figure above. We have 3 facies and 9 attributes, so the total inputs are 27 volumes. The first number in the name indicates facies, and the second number indicates attributes.



Go back to **aaspi_util** GUI and click (13) **attribute_selection**:



Using (14) *browse_and_add_to_current_list* to select 1D volumes generated from define_training_data; (15) write number of attributes; (16) write number of seismic facies; (15) and (16) should be identical to define_training_data; (17) define how many GMM clusters are generated through expectation maximization algorithms. For chaotic facies, there are in general more than one cluster. (18) Use stochastic expectation maximization or not; this is the same parameter as in **GMM3d**; after click (19) *Execute*, the process will start and show as (see next page):

```
[input_fn__3=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__3.H]
[input_fn__4=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__4.H]
[input_fn__5=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__5.H]
[input_fn__6=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__6.H]
[input_fn__7=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__7.H]
[input_fn__8=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__8.H]
[input_fn__9=/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__9.H]
[max_clusters=2]
[min_clusters=1]
[min_width=0.1]
[nattr=9]
[nfacies=3]
[nvolume=27]
[output_fn=aaspi_attribute_selection_2_22_0.out]
[sample_decimation=5]
[suffix=0]
[unique_project_name=2_22]
[use_NEM=n]
[use_SEM=n]
[use_mpi=y]
[verbose=n]



 ===========================================================================
 | PROGRAM: attribute_selection                                            |
 | COPYRIGHT 2019 ATTRIBUTE-ASSISTED SEISMIC PROCESSING AND INTERPRETATION |
 | THE UNIVERSITY OF OKLAHOMA                                              |
 | ROYALTY FREE USE TO AASPI SPONSORS AND COINVESTIGATORS                  |
 | SOFTWARE DISTRIBUTION TO OTHERS PROHIBITED WITHOUT COMMERCIALIZATION    |
 ! AGREEMENT                                                               |
 ===========================================================================
 software release date = 8 February 2019

 run time date = 20190222
                         month      year
        current date        2      2019
     expiration date       12      2099

 The number of input facies is           3
 The number of input attribte is           9
 The number of input volume is          27
 command line argument for file           1  input_fn__1
 file read in:/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__1.H
 command line argument for file           2  input_fn__2
 file read in:/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__2.H
 command line argument for file           3  input_fn__3
 file read in:/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__3.H
 command line argument for file           4  input_fn__4
 file read in:/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__4.H
 command line argument for file           5  input_fn__5
 file read in:/ouhomes2/qi1400/TX_LA/polygon_mask_selection_facies__1-attr__5.H
 command line argument for file           6  input_fn__6
```

After the entire process, the best attribute combination will be shown as:

```
The highest Bhattacharyya distance of each number of attribute combination is
 0.0000000E+00   9.707500       14.66341       15.01278       15.07014
  15.62346       16.17454       15.69815       10.61667
The attribute comb associated with the highest Bhattacharyya distance
          0           0           0           0           0           0
          0           0           0           1           4           0
          0           0           0           0           0           0
          1           4           6           0           0           0
          0           0           0           1           2           4
          6           0           0           0           0           0
          1           2           4           5           6           0
          0           0           0           1           3           4
          5           8           9           0           0           0
          1           2           4           5           6           7
          9           0           0           1           2           3
          4           6           7           8           9           0
          1           2           3           4           5           6
          7           8           9
The best attribute combination is           1           2           4
          5           6           7           9           0           0
Best extracted attribute number is           7
```

(20) indicates the highest cumulative Bhattacharyya distance of each attribute number of attribute combinations; (21) indicates the attribute index associated with those highest cumulative distance, attribute No. 1 is the first input attribute in **define_training_data** that is **energy_ratio_similarity** attribute, and attribute No. 4 is the fourth input attribute in **define_training_data** that is **bandwidth_spectral** attribute; (22) indicates the index of the best attribute combination; (23) indicates the attribute number of the best attribute combination.

Displaying the results

We first validate our attribute selection workflow by applying to the deep-water dataset, acquired from the Gulf of Mexico (GOM). The seismic data is located offshore Louisiana shelf edge and cover approximately 8000 km^2 with 37.5 m by 25 m bins. Salt domes rise from the deepest sections of the basin, where complex masses of shale and mud slides migrate toward and deposits at a minibasin. Multiple facies can be observed in the dataset which include undeformed shale, interbedded sand and siltstone, salt domes, and mass transport complexes.

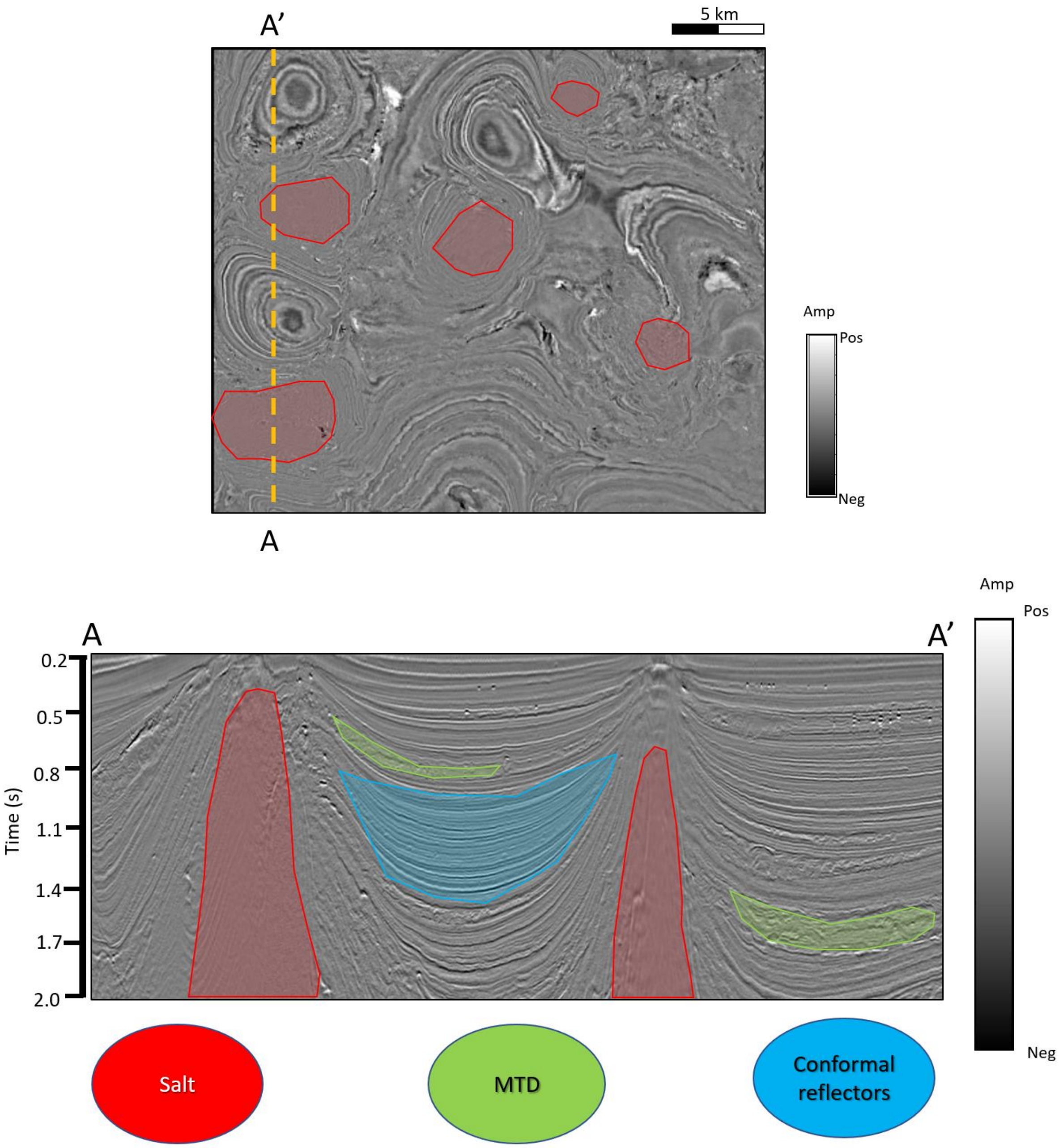


Figure 1. (a) Time slice at t = 1.22 s, and (b) vertical slice along line AA' through the seismic amplitude volume. Red polygons indicate salt diapirs voxels that will be used to define the location of this facies in multiattribute space. Green polygons indicate MTD facies, and blue polygons indicate conformal reflectors. Salt facies exhibit weak envelope, low frequency, and deviating boundaries of the reflector dip. When defining training data, interpreters only pick those voxels in which they have the greater confidence.

Seismic amplitude played as the initial attribute in seismic interpretation is able to identify most of large geologic features by the spatial variation of seismic amplitude and phase. Other seismic attributes derived from seismic amplitude provide quantitative measure of statistical and geometric patterns of geologic features. Figure 1a shows the time slice at t = 1.22 s through the seismic amplitude volume. Figure 1b shows the line AA' through the seismic amplitude volume. Note the polygons indicate the three picked facies of interest (M=3), which are salt diapirs, conformal reflectors, and MTD. Seismic expression of salt in the GOM data is vertically and laterally chaotic, and incoherent. Because the data is prestack time migrated, parts of reflectors such as the boundaries of salt are mis-migrated and also able to be observed on salt diapirs. Seismic expression of mass transport complexes is incoherent and chaotic as well, however mass transport complexes exhibit mixed energy and frequency rather than low energy and frequency in seismic expression of salt. In the picked salt facies (Figure 2a), it contains mixtures of seismic noise that are incoherent subsequent facies, and migration artifacts that are coherent subsequent facies. MTD facies can also be seen as coherent, rotated reflectors (Figure 2b), which are depositions of mudstone or siltstone blocks, or sliding gravity flows of shale formation. The zoomed conformal background is shown in Figure 2c, and consists of coherent sediment and shale layers. Figure 3 indicates GMMs of the MTD within two subsequent facies on the 2D attribute space.

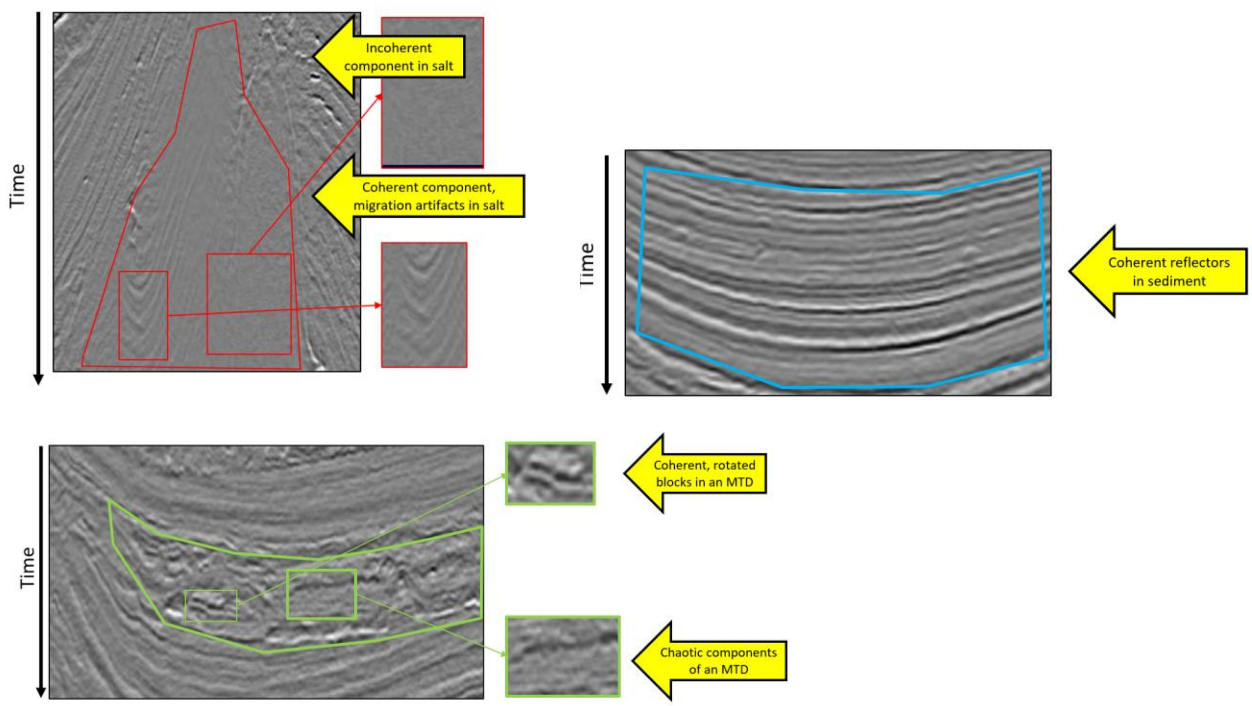


Figure 2. Zoomed vertical slices from Figure 5b of picked (a) salt, (b) MTD, and (c) conformal background facies. Note salt and MTD may contain coherent and incoherent subsequent facies.

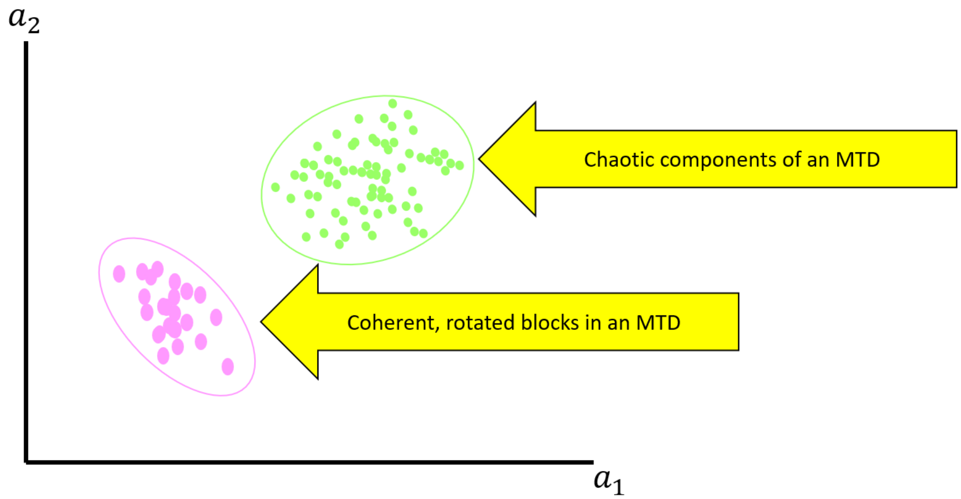


Figure 3. A cartoon of two GMMs represents chaotic components and coherent, rotated blocks in an MTD.

To avoid redundant use of attributes, we identify salt features of dip, amplitude, and frequency variation, using the coherence, grey-level co-occurrence matrix (GLCM), nonparallelism attributes, and statistic measures of frequency, totaling nine candidate attributes (N=9). The candidate attributes selected for the GTM should successfully measure different seismic responses of salt, conformal reflectors, and MTD's. Figure 4 shows the list of the input candidate attributes. Note that the deviation of vector dip, the deviation of energy gradient attributes, and the covariance of vector dip and energy gradient highlight chaotic salt areas, high energy salt boundaries, and nonparallelism and randomness of seismic reflectors, respectively. The spectrum bandwidth attribute, and the spectrum roughness attributes belong to statistic measures of spectrum that are used to detect frequency variation between salt and other facies. Additionally, the spectrum bandwidth of salt rapidly changes and exhibits chaotic anomalies. The roughness of spectrum salt is much lower than other facies. These two statistic measures of spectrum describe the frequency variation of salt diapirs in two different ways.

Input attributes
1. Coherence
2. Spectral bandwidth
3. Covariance of dip and energy
4. GLCM entropy
5. GLCM variance
6. Energy deviation
7. Spectral roughness
9. Reflector convergence
9. Dip deviation

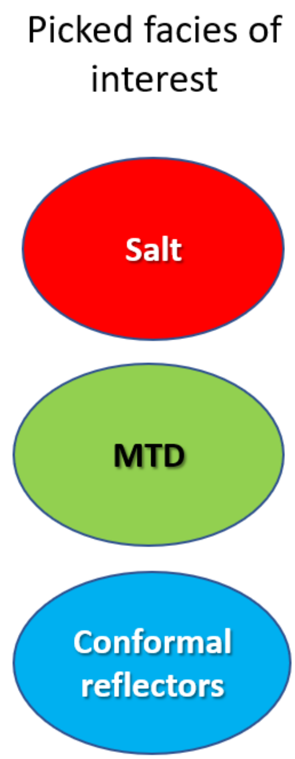


Figure 4. List of the candidate attributes and the picked facies of interest. Coherence measures the similarity between traces, which is also sensitive to strong random noise. Spectral bandwidth and spectral roughness are statistical measures of spectrum. GLCM entropy and GLCM variance are texture attributes and measure texture variations of seismic amplitude images. Dip deviation, energy deviation, and covariance of dip and energy measure lateral changes of reflector dip, lateral changes of reflector energy, and lateral changes of covariance of dip and energy. Reflector convergence measures vertical changes of reflector dip.

***n* selected attributes**	**Number of combinations** A_n	**Number of clusters** J_n	**Number of the comparisons**
1	9	54	1434
2	36	216	23220
3	84	504	126756
4	126	756	285390
5	126	756	285390
6	84	504	126756
7	36	216	23220
8	9	54	1434
9	1	6	15

Figure 5. List of each *n (n=1,2,…,9)* selected attributes associated with the number of combinations, the number of clusters, and the number of the comparisons in each sub-group combinations.

Extracting the training voxels of the three picked facies from the nine candidate attributes is the first step of the attribute selection workflow. The total number of combinations for a nine-attribute combination is 511. For each combination A_n n of n selected attribute each facies m, we compute J GMM clusters. Different attribute combinations can have different numbers of GMM clusters. While a "homogeneous" seismic facies like salt may be well represented by a single GMM, more heterogeneous seismic facies like MTD's may require two or more GMMs. To decrease computation cost and the influence of seismic random noise, the maximum cluster number of each attribute combination of each facies is set to two. The training voxels is generated to multiple GMMs. The subsequent combination number of each n combination A_n and the number of GMM clusters that associate with the subsequent combination number are shown on Figure 5. The Bhattacharyya Distance between each GMM cluster in each n selected attribute combination is computed to compare the similarity of each two-cluster pair of GMM. The number of the Bhattacharyya Distance computed about each n selected attribute combination is J!/(J-2)!/2!. Next, we compute the commutative distance to evaluate attribute combinations, then average the distance by the number of the sub-group possible combination, the number of picked facies, and the number of the computed GMM clusters in the identical number of the attribute space. Figure 6 shows the average cumulative distance of each possible number of selected attributes from the total nine attribute. The maximum distance is within the seven selected attributes and those attributes are coherence, spectral bandwidth, GLCM entropy, GLCM variance, energy deviation, spectral roughness, and dip deviation (Figure 7). We examine histograms of the seven selected attributes (Figure 8). The red masks indicate most of the salt voxels located on the histograms of the selected attributes. Note that histograms of selected attributes can be approximately separated salt voxels from other facies using simple thresholds.

n	1	2	3	4	5	6	7	8	9
Attributes	3	1,4	1,4,6	1,2,4,6	1,2,4,5,6	1,3,4,5,8,9	1,2,4,5,6,7,9	1,2,3,4,6,7,8,9	1,2,3,4,5,6,7,8,9
Distance O_n	9.65	9.71	14.66	15.01	15.07	15.62	16.14	15.70	10.62

Figure 6. List of the average cumulative distance of each n selected attributes. Note that the attribute combination associated with the highest distance is 1, 3, 4, 5, 8, 9 (attribute index in **define_training_data**).

Selected attributes
1. Coherence
2. Spectral bandwidth
4. GLCM entropy
5. GLCM variance
6. Energy deviation
7. Spectral roughness
9. Dip deviation

Figure 7. List of the attributes in the selected attribute combination.

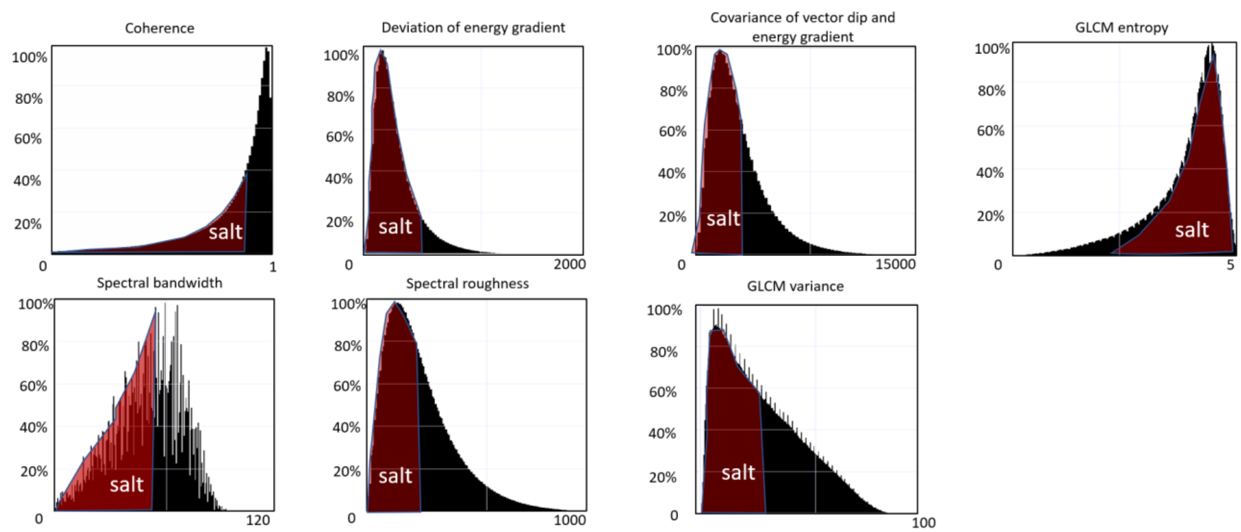


Figure 8. Histograms of the seven selected attributes. The red masks indicate most of the salt voxels located on the histograms of different attributes. Note the histograms of salt voxels can be approximately separated from other facies by thresholds.

References

Celeux, G. and G. Govaert, 1993, Gaussian Parsimonious clustering models: Pattern Recognition, **28**, 781-793.

Hampson, D. P., J. S. Schuelke, and J. A Quirein, 2001, Use of multiattribute transforms to predict log properties from seismic data: Geophysics, 66, 220-236.

Hardisty, R., 2017, Unsupervised seismic facies using Gaussian mixture models, Thesis.

Mak, B., and E. Barnard, 1996, Phone clustering using the Bhattacharyya distance, Proc. Int. Conf. Spoken Language Processing, **4**, 2005-2008.

Schwarz, G., 1978, Estimating the dimension of a model: The Annals of Statistics, **6**, 461–464