# Projection Pursuit Multivariate Transform

Ryan M. Barnett, John G. Manchuk, and Clayton V. Deutsch

*Transforming complex multivariate geological data to be multivariate Gaussian is an important and challenging problem in geostatistics. A variety of transforms are available to accomplish this goal, but may struggle with data sets of high dimensional and sample sizes. Projection Pursuit Density Estimation (PPDE) is a well-established non-parametric method for estimating the joint PDF of multivariate data. A central component of the PPDE algorithm involves the transformation of the original data towards a multivariate Gaussian distribution. Rather than use the PPDE for its original intended purpose of density estimation, this convenient data transformation will be utilized to map complex data to a multivariate Gaussian distribution within a geostatistical modeling context. This approach is proposed as the Projection Pursuit Multivariate Transform (PPMT), which shows the potential to be very effective on data sets with a large number of dimensions. The PPMT algorithm is presented along with considerations and case studies.*

**Introduction**

The multivariate Gaussian distribution is very commonly adopted within geostatistics for describing geologic variables. This is due to the mathematical tractability of the multiGaussian distribution [10], which may be either convenient or necessary for geostatistical modeling. As geologic variables are often non-Gaussian in nature, a wide variety of techniques [1] are available to transform them to a multiGaussian form, with their associated back-transforms to reintroduce the original distributions.

The widely applied normal score transformation [5,10] will guarantee that variables are made univariate Gaussian; however, multivariate complexities such as heteroscedastic, non-linear, and constraint features may still exist. Multivariate Gaussian transforms such as the Stepwise Conditional Transform [11], kernel based methods [12], and the Multivariate Standard Normal Transformation (MSNT) [3,4] may successfully transform complex data to be multiGaussian, but could suffer from issues revolving around the number of dimensions or observations in the data.

Likely to excel on data sets that cause stress for the above techniques, the Projection Pursuit Multivariate Transform (PPMT) is proposed here for facilitating multiGaussian geostatistical modeling. Adapted from the Projection Pursuit Density Estimation (PPDE) algorithm [6,7,8,9], the PPMT iteratively searches for and Gaussianizes highly non-Gaussian 1-D projections in the data. Following a requisite number of iterations, the complex data are transformed to be multiGaussian. The success of the PPMT is judged based on maintenance of the original multivariate structure [3] and degree of Gaussianity in the transformed data.

An inverse-distance technique originally developed for the MSNT [3,4] is used to back-transform simulated Gaussian realizations to the original complex space based on the PPMT mapping. Additional observations may be retrieved using Gibbs sampling [2] in order to better define the PPMT mapping and facilitate its multivariate extrapolation when back-transforming.

A review of the essential PPMT theory is presented, followed by a discussion on its major considerations. A 2-D synthetic dataset is used throughout these sections for demonstration. A 3-D Nickel Laterite dataset of greater non-linear and stoichiometric complexity will then be used to demonstrate the PPMT in a case study.

**Projection Pursuit Concept**

First introduced by Freidman and Tukey [7], PPDE may be used to determine the joint probability density function (PDF) of a multivariate distribution. Non-parametric in nature, the PPDE is particularly well suited and effective relative to other techniques when applied to complex data of a high number of dimensions [7,9].

The overall premise is to detect linear projection vectors in the data[1] that are the most complex or interesting, where the Gaussian distribution is treated as the least complex [8] because the projection

[1] Friedman [6] also developed the PPDE algorithm for working with 2-D projections along planes.

of a multiGaussian distribution is also Gaussian [10]. The assumption is made that non-Gaussian structures in the higher dimensions will be exhibited in the lower dimensional projection. Friedman [6] discusses that projections exhibit a smoothed shadow of what are likely more marked complexities in the higher dimensions.

Once the most interesting projection vector as been determined, the individual high dimensional components may be transformed to normalize their projection (termed Gaussianize by the literature). Iterating this search and Gaussianize algorithm, the high dimensional data is gradually transformed to a multiGaussian distribution. The final step of the PPDE algorithm involves estimating the multivariate density through combining the 1-D projections, but that process is not relevant to the PPMT.

**Projection Pursuit Theory**

The presented PPDE algorithm in this section was almost entirely developed by Friedman [6], although its syntax largely follows Hwang [9]. An initial normal score transform and a modification to the data sphering are the only changes that are made from these sources.

*Normal Score and Data Sphering*

Before the iteration steps of projection pursuit may be applied, the data matrix $z_{k\,x\,n}$ of $k$ dimensions or variables and $n$ observations must first be transformed to have suitable properties. The familiar normal score transform (Equation 1) is first applied, where quantile matching is performed between the empirical data CDF, $F$, and the standard normal CDF, $G$. This is done so that the PPMT transformed variables will reside in standard normal units.

$$y = G^{-1}(F(z)) \tag{1}$$

An orthogonal covariance matrix and unit variance between the variables will also be critical at several stages of the iteration algorithm, which is achieved by data sphering in Equation 2. Here the linear combination of the normal scored data, $y$, and the sphering matrix $S^{-1/2}$, produces the transformed data, $x$, that is suitable for projection pursuit. The sphering matrix is calculated according to Equation 2, where the eigenvector matrix $U$ and the eigenvalue matrix $D$ are attained from spectral decomposition of $y$'s covariance matrix. Note that traditional data sphering [6] will also center the data through subtracting $E\{y\}$. This is not necessary here, however, since the data have already been centered by the normal score transform.

$$x = S^{-1/2} y, \; where \; S^{-1/2} = UD^{-1/2}U^T \tag{2}$$

*Projection Index*

Central to PPDE is the test statistic (termed projection index by the literature) that is used to measure the deviation of each projection away from the Gaussian distribution. Friedman's projection index [6] will be presented here due to the promising initial results that it produced. It was specifically designed to place greater emphasis on the body of the distribution as opposed to the tails. This was done because complex structures such as multi-modality and non-linearity will most often occur near the distribution's center. Many of the more commonly applied test statistics are more highly sensitive to the tails of a distribution, and therefore were deemed less suitable [6].Excellent sources [6,9] are available for the somewhat lengthy development of the projection index through its conceptual, integral, and numerical forms, but only the latter will be presented here.

Given a directional vector $\alpha_{k\,x\,1}$ and the associated projection $p = \alpha^T x$, the projection index $I(\alpha)$ may be calculated according to Equation 3.

$$I(\alpha) = \sum_{j=1}^{d} \frac{2j+1}{2} E_r^2\{\psi_j(r)\} \tag{3}$$

Here the Legendre polynomials are denoted by $\psi_j(r)$, whose calculation is given by Equation 4.

$$\psi_0(r)=1,\ \psi_1(r)=r,\ \text{and}\ \psi_j(r)=\left[(2j-1)r\psi_{j-1}(r)-(j-1)\psi_{j-2}(r)\right]\Big/ j,\ \text{for}\ j\geq 2 \tag{4}$$

The Legendre polynomials are a function of *r*, which is a transformed version of the projection, *p*, according to Equation 5.

$$r=2G(p)-1,\qquad r\in[-1,1] \tag{5}$$

*Optimized Projection Search*

To quickly determine the vector $\alpha$ yielding the maximum projection index $I(\alpha)$, an optimized search is utilized that begins with a course stepping along combinations of the principal component axes[2] [6]. Once a maximum $I(\alpha)$ is determined along one of these major axes, $\alpha$ is then fine-tuned using steepest ascent optimization. This requires the derivative of Equation 3, which is given by Equation 6 under the constraint $\alpha^T\alpha=1$.

$$\frac{\partial I(\alpha)}{\partial\alpha}=\frac{2}{\sqrt{2\pi}}\sum_{j=1}^{d}rE_r^2\{\psi_j(r)\}\psi'_j(r)e^{-p^2/2}\left(x-\alpha p\right) \tag{6}$$

Here $\psi'_j(r)$ is the derivative of the Legendre polynomials, which is calculated according to Equation 7.

$$\psi'_1(r)=1,\ \text{and}\ \psi'_j(r)=r\psi'_{j-1}(r)+j\psi_{j-1}(r),\quad \text{for}\ j>1 \tag{7}$$

*Gaussianize*

With the vector $\alpha$ yielding the maximum $I(\alpha)$ determined, the final step of each iteration is to transform the high dimension data *x* so that its projection *p* along $\alpha$ is normalized according to Equation 8.

$$\tilde{p}=G^{-1}(F_\alpha(p)) \tag{8}$$

To accomplish this, the orthonormal matrix *U* (Equation 9) is first determined, where the beta coefficients are calculated using the Gram-Schmidt algorithm [9].

$$U=[\alpha,\beta_1,\beta_2,\ldots,\beta_{k-1}]^T \tag{9}$$

The linear combination of *U* and the data matrix *x* (Equation 10), results in a transformation where the first row is the projection $p=\alpha^T x$.

$$Ux=[\alpha^T x,\beta_1^T x,\beta_2^T x,\ldots,\beta_{k-1}^T x]^T \tag{10}$$

Next, let $\Theta$ be a vector that transforms the first row of *Ux* to the standard normal distribution according to Equation 8, but leaves the remaining orthogonal directions intact (Equation 11).

$$\Theta(Ux)=[\tilde{p},\beta_1^T x,\beta_2^T x,\ldots,\beta_{k-1}^T x]^T \tag{11}$$

Equation 12 is finally applied to transform the data in a manner that Gaussianizes the projection, but does not alter the orthogonal directions.

$$\tilde{x}=U^T\Theta(Ux) \tag{12}$$

Following Gaussianization, the projection index along this direction will be zero. The optimized search for the maximum projection index may then be reapplied to find other complex directions if they exist. Iteratively applying this search and Gaussianize procedure, the multivariate distribution will eventually approach a multiGaussian one.

A 2-D synthetic case study is introduced here to demonstrate the PPMT. The synthetic model and sampled data are displayed in Figure 1, where complex multivariate features and distinctive spatial structures are observed in the original Z1 and Z2 variables. Following the normal score and data sphering of these variables according to Equations 2 and 3, the projection pursuit iterations are demonstrated in Figure 2, where each iteration panel displays the histogram of the most complex projection and scatter plots before and after Gaussianizing. The projection vector with the maximum projection index for each

[2] This course stepping is first executed to minimize the potential of the subsequent gradient based optimization becoming trapped in a local maxima.

iteration is displayed by the solid line in the scatter plots, where the points are colored by their associated unaltered $X$ value (Equation 2 to serve as a visual reference for the shift of each observation through the iterations. This is referred to as dimension coloring [3], which is expanded on in the following section. Note the marked shift towards a multivariate distribution after only four iterations.

**Judging the Results**

Due to projection based dimension reduction residing at the core of PPDE, it was designed to be effective with high dimensional data. Relative to other forms of density estimation such as kernel methods, the PPDE has proven to be robust against the curse of dimensionality [6,9]. Judging how successfully the PPMT is applied for geostatistical frameworks will be based on the transform distortion and multivariate Gaussianity of the transformed data.

*Transform Distortion*

First, how well the original multivariate structure of the data is maintained through the transformation. A successful Gaussian mapping should minimize the changes that occur to the distances between observations in original and transformed space [3,4]. If observations that are very near to one another in original space are mapped far apart in transformed space (and vice-versa), it is likely that the multivariate structure will have been unreasonably distorted. This will result in the destruction of spatial structure that will be apparent in the variogram of the transformed variables. While the MSNT calculates this transform distortion in its objective function, its authors also proposed that the distortion may be intuitively judged according to dimension coloring [3]. Through coloring the transformed variable scatterplots according to their untransformed values, one gains insight into the nature of spatial shifting that has occurred as a result of the transform. Most important, is that if little distortion has occurred, a smooth gradient of color will be observed [3].

As discussed in the previous section, Figure 2 displays the first four iterations of the PPMT being applied to the synthetic data. Figure 3 then displays the transformed scatterplots following 25 iterations, where the points are colored by their associated Y values (following the normal score but prior to sphering and iterations as seen in Equation 1). A smooth gradient is seen, indicating that very little distortion has occurred since all of the neighbors in original space remain very close in the transformed space. The normal score and PPMT transformed variograms are also displayed in Figure 3, where very little loss of spatial structure has occurred. Based on work with a variety of datasets, including the two presented in this paper, the PPMT appears to produce minimal transform distortion.

*Multivariate Gaussianity and Stopping Criteria*

Also important in judging the PPMT success is the Gaussianity of the transformed data following the final projection pursuit iteration. Having decided on the form of the projection index $I(\alpha)$ in Equation 3, the test statistic that is used to define Gaussianity has already been determined. Choosing the value to which the maximum $I(\alpha)$ must descend before the data are deemed Gaussian enough (stopping criteria) may be somewhat subjective and is not yet fully defined.

Complicating this decision is that the PPMT will not likely achieve the same degree of Gaussianity in data of decreasing observations and increasing dimensions. The $I(\alpha)$ in Equation 3 is standardized by the number of observations and dimensions used in its calculation, allowing for the direct comparison of its maxima across datasets of differing observations and dimensions (Figure 4). It may be observed that increasing the number of observations leads to far faster descent towards Gaussianity, as well as a final superior form of Gaussianity. This is in line with previous literature [6,9], where Hwang even suggested that 400 observations may be required to achieve a reasonably Gaussian form. Likewise, decreasing the number of dimensions results in improved convergence towards a final Gaussian form. Visual inspection of scatterplots for each iteration and dataset in Figure 4 would suggest that anything less than a maximum $I(\alpha)$ of 0.05 could be considered reasonably Gaussian. The presented results in this paper, however, are simply based on a constant 25 iterations due to the profile of maximum $I(\alpha)$ that is apparent in Figure 4. Stopping criteria that may require additional iterations beyond 25 is not anticipated

to be an issue since the transform distortion of the PPMT tends not to degrade with increasing iterations, and execution time is minimal (Figure 5).

**Projection Pursuit Back-Transform**

Following any arbitrary modeling framework in Gaussian space, a method is required for back-transforming the simulated realizations to their original distribution. The MSNT back-transform [3,4] may be appropriately applied to a PPMT mapping. Though the PPMT and MSNT are entirely different algorithms, the end results of a Gaussian mapping is achieved by both.

Since the simulated grid nodes will not possess identical values to the mapped observations, an interpolation method is used by the MSNT to infer their location in original space based on their proximity to the nearest mapped observations in Gaussian space. This concept is represented by Equation 13, where the $i^{th}$ simulated location is back-transformed based on its euclidean distance in transformed space $d_{t(i\,j)}$ , to the $j^{th}$ mapped observation. This amounts to inverse distance weighting, where $d_{t(i\,j)}$ will determine the weight attributed to the original values of the j$^{th}$ mapped observation on the $y_i$ estimate. Any number of nearest mapped observations could be used for this interpolation, but it is advocated [3] that this should be chosen based on the k number of multivariate dimensions, where $k+1$ number of observations is optimal. Using less than this does not adequately constrain the multivariate interpolation, but increasing beyond $k+1$ will begin to converge the back-transformed results towards the mean.

$$y_i = \sum_{j=1}^{k+1} \lambda_j y_j, \text{ where } \lambda_j = \frac{1}{d_{t(i\,j)}} \text{ and } \sum_{j=1}^{k+1} \lambda_j = 1 \tag{13}$$

Returning to the PPMT transformed synthetic data from Figure 5, SGSIM is used for simulating the Gaussian variables before back-transforming according to Equation 13. Selected model validation plots of the back-transformed Z1 and Z2 realizations are shown in Figure 6 where good reproduction of the univariate, multivariate, and spatial statistics is observed. As it is difficult to gain a sense of whether the joint density of the data is being reproduced by the realizations based on the scatterplots, bivariate Gaussian kernel density estimation (KDE) plots are displayed in Figure 7. Excellent reproduction of the joint density is observed according to this figure. Note that this synthetic dataset was composed of 280 observations, and therefore represents a somewhat challenging form for the PPMT. A potential concern with the back-transformed scatterplot is that no extrapolation takes place, with the realizations entirely constrained within the convex hull of the data. Further, there are somewhat aesthetically unpleasing strings of data that occur where few original mapped observations exist to define the mapping. While these issues may not be a concern for most applications, potential solutions will be the focus of the next section.

*Gibbs Sampler for Improved Mapping and Extrapolation*

It is apparent from Figure 1 that multivariate tails exist in the True model that the 280 sample data do not capture. This is a common sampling phenomenon that is often handled by allowing some extrapolation beyond the limits of the sample data where such a practive is justified. This extrapolation is straight forward in the univariate case (Figure 8), where transforms such as the normal score [5] perform interpolation between the extreme observations of the sample data and user specified tail values. Multivariate extrapolation is not straight forward, because the location of these tail values cannot be described by a single number; rather they take on the form of a multidimensional contour. These contours are known in the case of parametric distributions such as the Gaussian model (Figure 8), but are not often understood for non-parametric data.

One potential solution is to perform a sampling of the multivariate distribution in order to obtain a greater number of observations for the forward mapping. So long as the utilized sampling algorithm defines an accurate density for the tails of the distribution, a great number of sampled observations (ideally matching the number of simulated nodes) will provide the forward mapping with points in this extrapolation space.

As seen in Figure 7, areas in the multivariate space where there are a sparse number of mapped observations will result in unattractive stringing of back-transformed realizations. This multivariate

sampling approach also presents a solution to this issue, as a great number of mapping observations are introduced to these sparse regions, reducing the occurrence of this stringing effect.

Clearly the success of this approach depends on the chosen multivariate sampling algorithm for generating a set of observations that are representative of the original density and critical statistics. A non-parametric algorithm based on the Gibbs sampler is utilized [2], which is particularly well suited for high dimensional and complex multivariate data. The Z synthetic variables are normal scored to their Y equivalents, before applying the multivariate sampler [2] to define an additional 10,000 observations (Figure 9). These data are then combined and transformed via the PPMT, with the first and last iterations shown in Figure 10. Scatterplots of the combined data are displayed beside the isolated original data to show that both sets approach Gaussianity. This is to be expected so long as the Gibbs observations truly reproduce the original data density. Using only the original data to condition the subsequent SGSIM simulation, the Gaussian realizations are then back-transformed using the full combination of Gibbs and original mapped observations. The scatterplot of one back-transformed realization is presented in Figure 11, alongside scatterplots of the True model and a PPMT realization without Gibbs observations to define the mapping. The original data is overlain on each scatterplot for reference. The Gibbs observation mapping results in a back-transformation that reasonably reproduces the tails observed in the True model. Very few string artifacts are also observed.

**Case Study**

Nickel laterite data composed of 933 observations and 3 variables (Ni, Fe, and $SiO_2$) will be used to demonstrate a geostatistical PPMT based modeling framework. As observed in the scatterplots of Figure 12, highly complex multivariate features are present including non-linearity, and stoichiometric constraints. Since the variables are compositional, the workflow uses a logratio transform [1]. This is followed by the PPMT, with the first and last iterations shown in Figure 13. Based on dimension coloring and variography in Figure 14, the PPMT has transformed the data with minimal distortion to the multivariate structure. This is reflected in the minimal loss of spatial structure that is observed in the transformed variograms, relative to the original normal score variograms. SGSIM is then used to independently simulate the uncorrelated Gaussian variables, with the original data used for mapping the realizations back to original space. Validation plots are displayed in Figures 15 and 16, where good reproduction of the univariate, multivariate, and spatial structure is observed. The slight decrease in spatial correlation of the realizations relative to the data and deviations of the Q-Q plots are mainly attributed to stationarity concerns rather than PPMT related issues.

Additional variables from the Nickel laterite dataset were used to assess the applicability of the PPMT in higher dimensions. MgO, Co, and $Al_2O_3$ were added in varying amounts to form a total of 4 and 6 variables (from which Figure 4 was constructed). While a slight degradation in the final Gaussianity of the transformed variables was observed (Figure 4), the PPMT was found to be successfully applied to geostatistical frameworks with this increasing number of variables.

**Conclusion**

The PPMT developed in this work is a promising new approach for facilitating multivariate geostatistical modeling. The PPMT transforms any multivariate dataset to be uncorrelated and multivariate Gaussian. An inverse-distance based interpolation method is used for back-transforming the realizations. This back-transformation may be aided by multivariate sampling to better define the sparsely mapped regions, as well as to allow for extrapolation. Although it performs best with an increasing number of observations and a decreasing number of variables, initial testing found that the PPMT may be successfully applied to datasets of varying size and dimension.

Future PPMT research is likely to center on the consideration of alternative projection indices than the one employed by Friedman [6]. Additionally, a more definitive stopping criteria is required, with back-fitting [6,9] applied for improving multiGaussianity of the transformed data.

**References**


1 Barnett, R. (2011). *Guidebook on Multivariate Geostatistical Tools.* Edmonton, Alberta: Centre for Computational Geostatistics.

2 Barnett, R., & Deutsch, C. (2012). Non-Parametric Gibbs Sampler with Kernel Based Conditional Distributions. *CCG Annual Report 14*, Paper 102.

3 Barnett, R., & Deutsch, C. (2012). MSNT Advances and Case Studies. *CCG Annual Report 14*, Paper 101.

4 Deutsch, C. (2011). Multivariate Standard Normal Transformation. *CCG Annual Report 13*, Paper 101.

5 Deutsch, C., & Journel, A. (1998). *GSLIB: A geostatistical software library and user's guide, second edition.* Oxford University Press.

6 Friedman, J. (1987). Exploratory Projection Pursuit. *Journal of the American Statistical Association, vol.82*, pp.249-266.

7 Friedman, J., & Tukey, J. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE, C-23*, 881-890.

8 Huber, P. (1985). Projection pursuit. *Annals of Statistics, vol.13*, pp.435-475.

9 Hwang, J., Lay, S., & Lippman, A. (1994). Nonparametric multivariate density estimation: a comparative study. *IEEE Transactions on Signal Processing, vol.42*, pp.2795-2810.

10 Johnson, R., & Wichern, D. (1988). Applied Multivariate Statistical Analysis. New Jersey: Prentice Hall.

11 Leuangthong, O., & Deutsch, C. (2003). Stepwise conditional transformation for simulation of multiple variables. *Mathematical Geology, vol.35, no.2*, pp.155-173.

12 Manchuk, J., & Deutsch, C. (2011). A program for data transformations and kernel density estimations. *CCG Annual Report 13*, Paper 116.

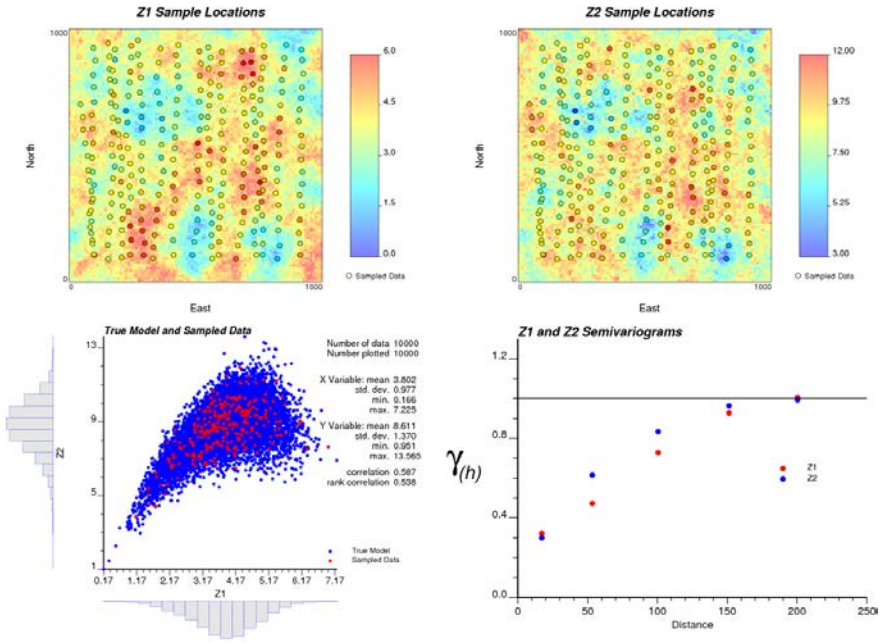




**Figure 1:** Overview of the synthetic model construction and resultant properties of Z1 and Z2. The top maps display the True model values (gridded) and data sampling locations (circles). The bottom left scatter plot displays the True model (blue) and sampled data (red) values. The experimental omni-directional semi-variograms are shown in the bottom right for Z1 (red) and Z2 (blue).

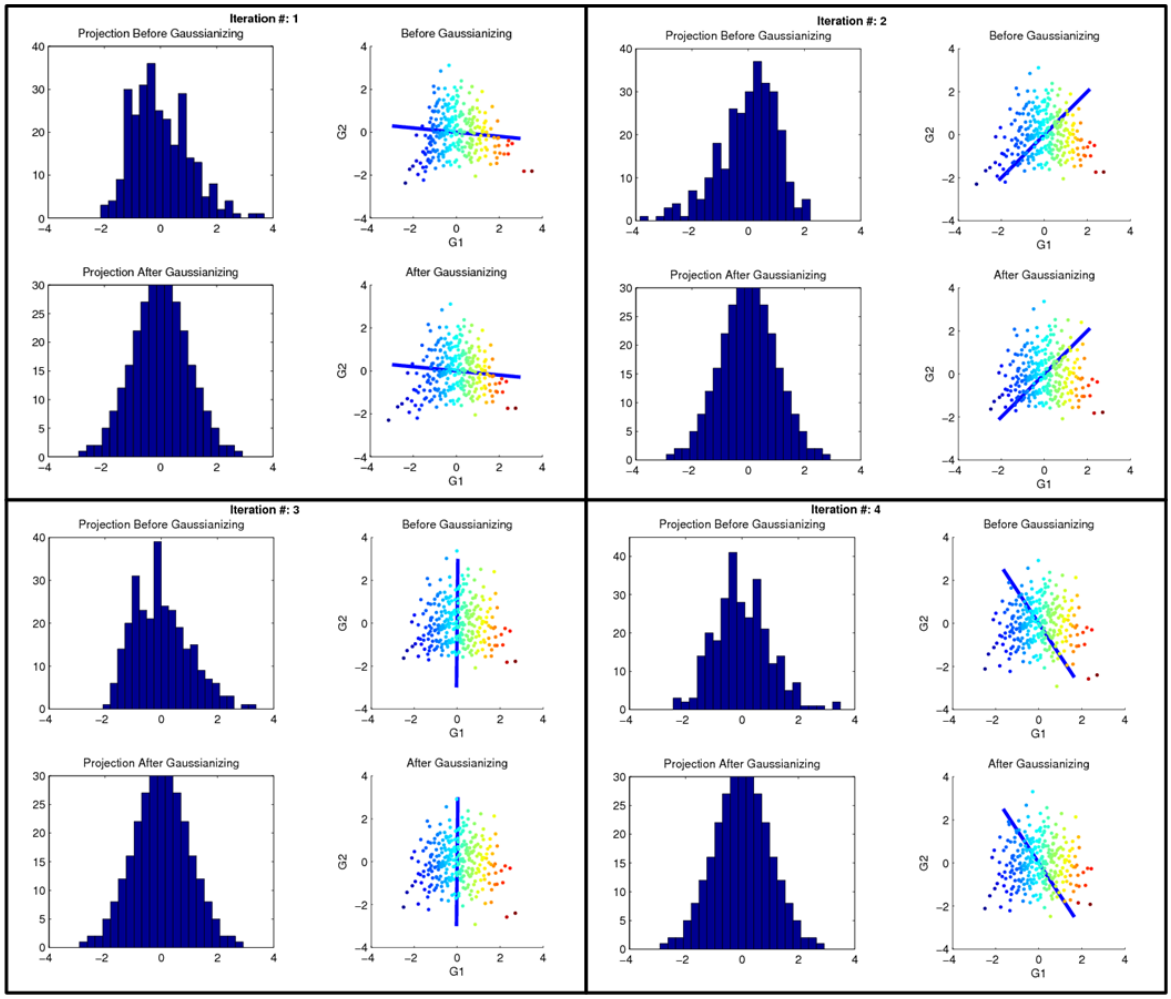



**Figure 2:** Illustration of the first four iterations of the PPMT on the 2-D synthetic data. Points are colored by their associated $X$ value.

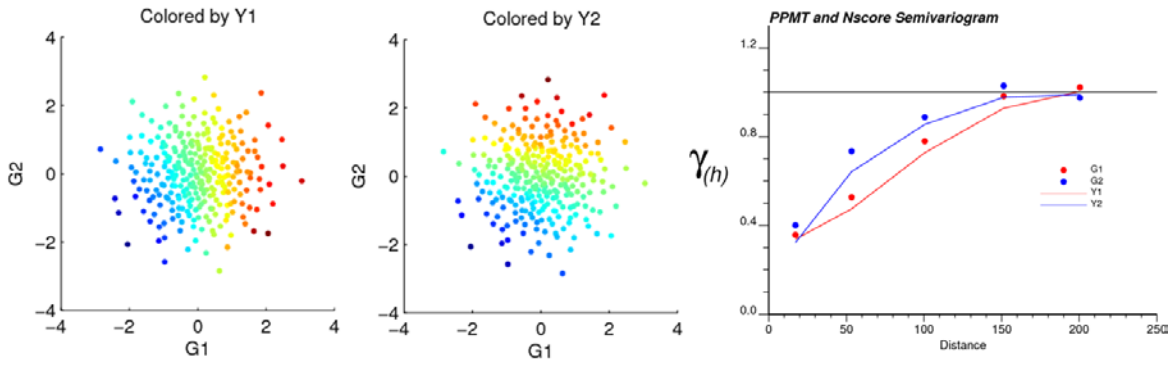
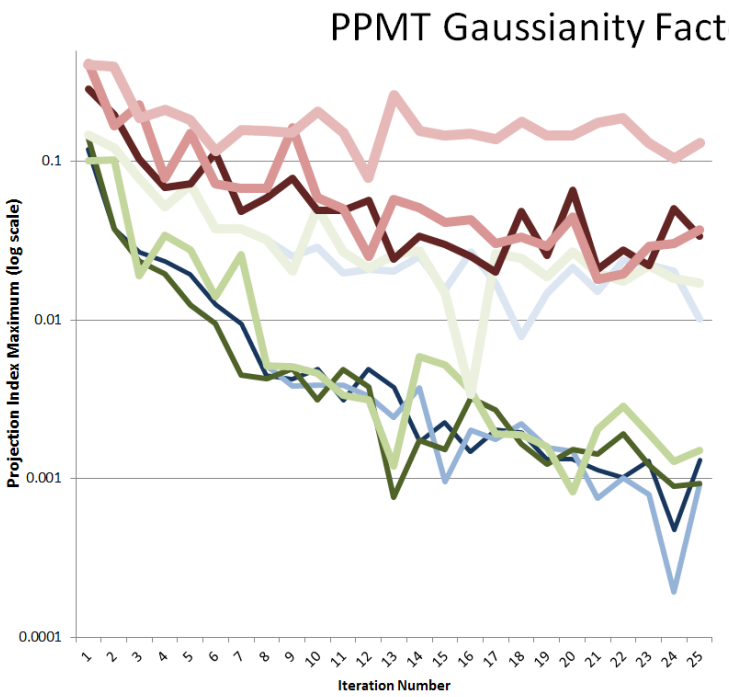



**Figure 3:** Scatterplots of G1 and G2, where each point is colored by its associated Y1 (left) and Y2 (middle) values. The omni-directional experimental semivariogram before and after transformation are shown on the right.




| Predicted Difficulty Contours (Used in Line Scaling) | | | |
|---|---|---|---|
| Number of Variables | Number of Data | | |
| | 900 | 500 | 200 |
| 2 | 1 | 2 | 5 |
| 4 | 4 | 5 | 6 |
| 6 | 5 | 6 | 7 |

| Actual Difficulty Contours (Weighted value of Iterations 20-25) | | | |
|---|---|---|---|
| Number of Variables | Number of Data | | |
| | 900 | 500 | 200 |
| 2 | 1.0 | 1.0 | 17.2 |
| 4 | 1.2 | 1.6 | 18.5 |
| 6 | 33.6 | 27.2 | 132.9 |

**Figure 4:** Summary of the PPMT Gaussianizing difficulty with a changing number of observations and dimensions. Projection index maximum value for each PPMT iteration is displayed, where lines are colored, shaded, and scaled according to the top table. Based on a weighted average value of the final five iterations, the actual Gaussianizing difficulty is given in the second table.

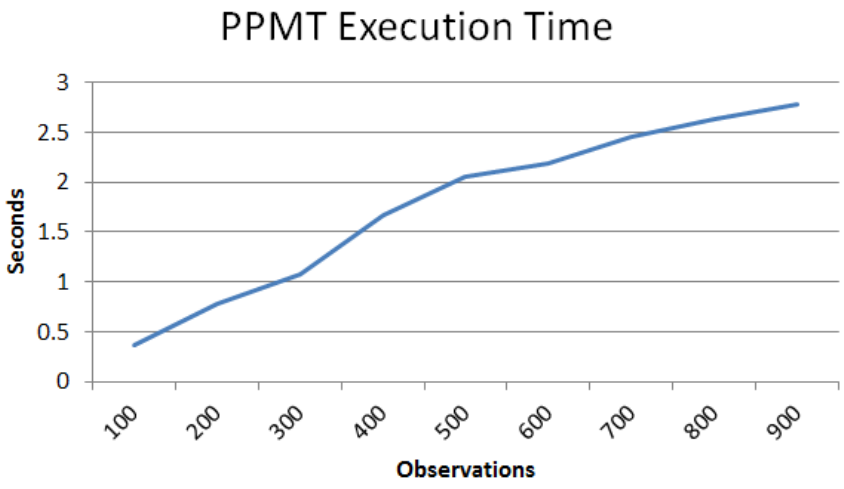



**Figure 5:** PPMT execution time (25 iterations) for a changing number of observations.

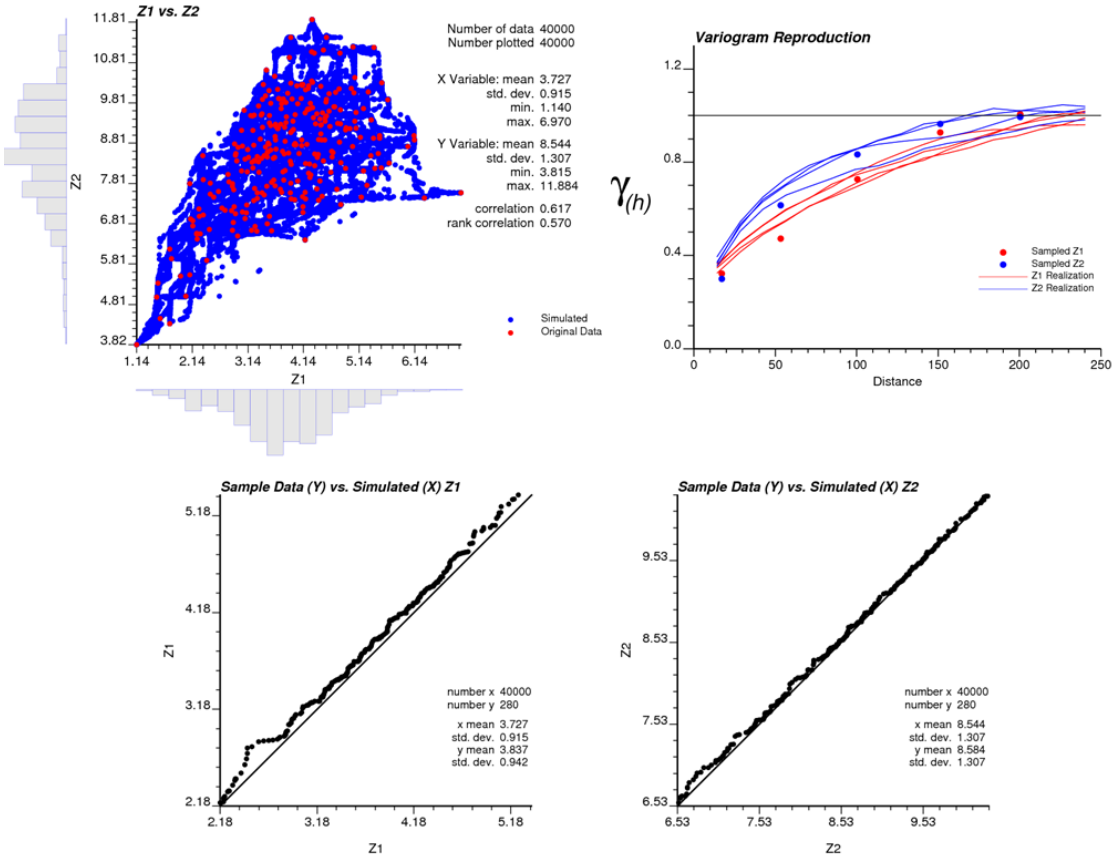



**Figure 6:** Selection of model validation plots. Scatterplots between the original data (red) and simulated values (blue) in the top left. Experimental semivariogram of the data (dots) and the simulated realizations (lines) for Z1 (red) and Z2 (blue) in the top right. Q-Q plots between the declustered data and simulated realizations for Z1 (bottom left) and Z2 (bottom right).

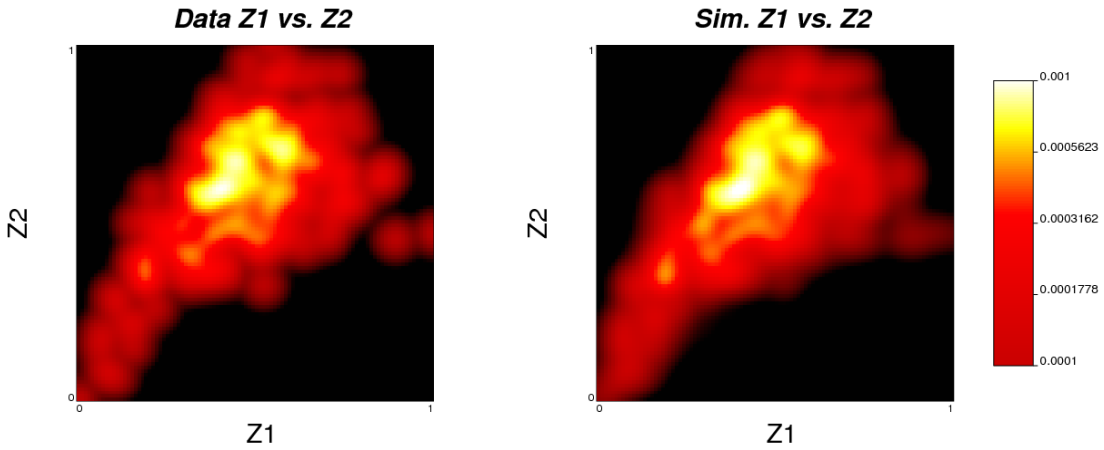



Figure 7: Gridded bivariate Gaussian KDE of the original data (left) and simulated realization (right).

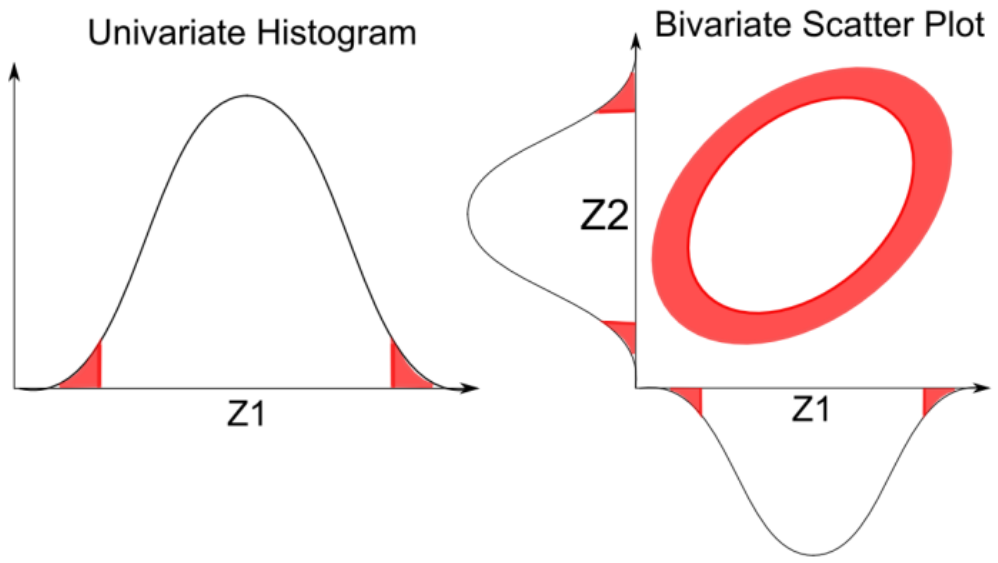



**Figure 8:** Schematic illustration of the univariate (left) and bivariate (right) Gaussian distribution regions that will require extrapolation in the case of simulated nodes exceeding the sampled data.

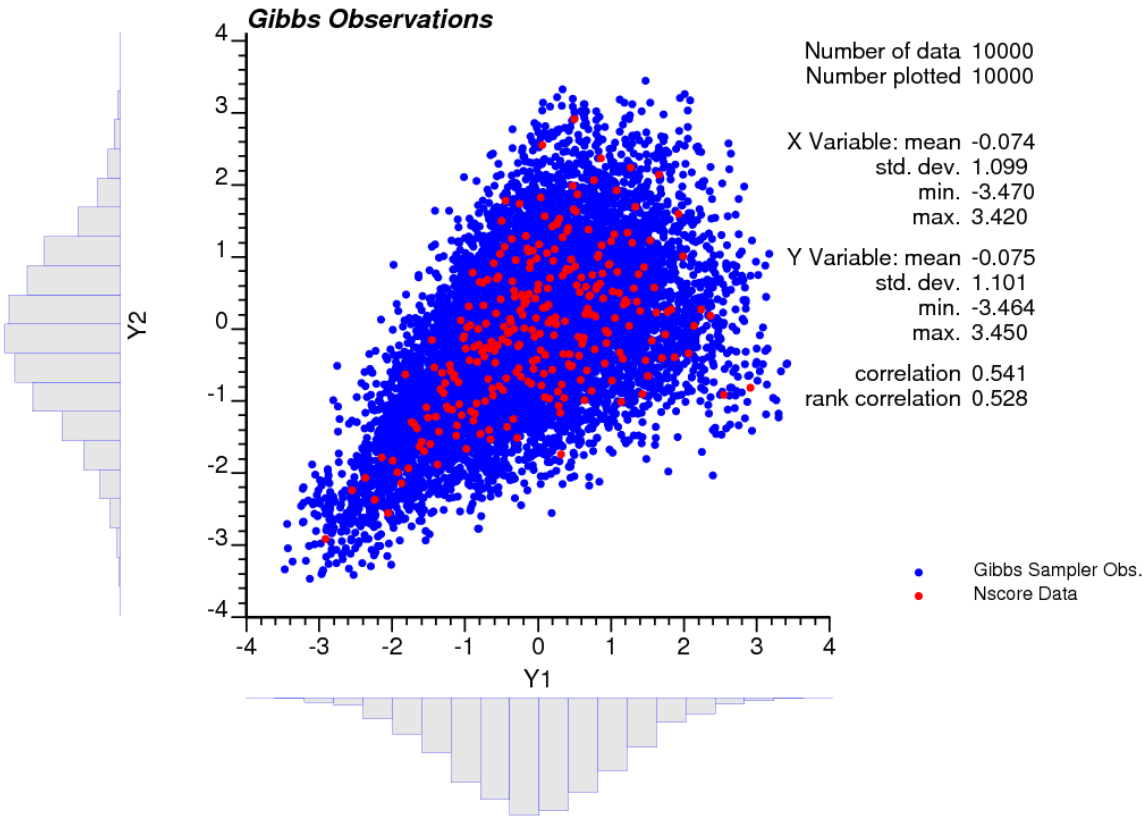



**Figure 9:** The 280 original normal scored Y data (red) overlain on 10,000 Gibbs sampling observations (blue).

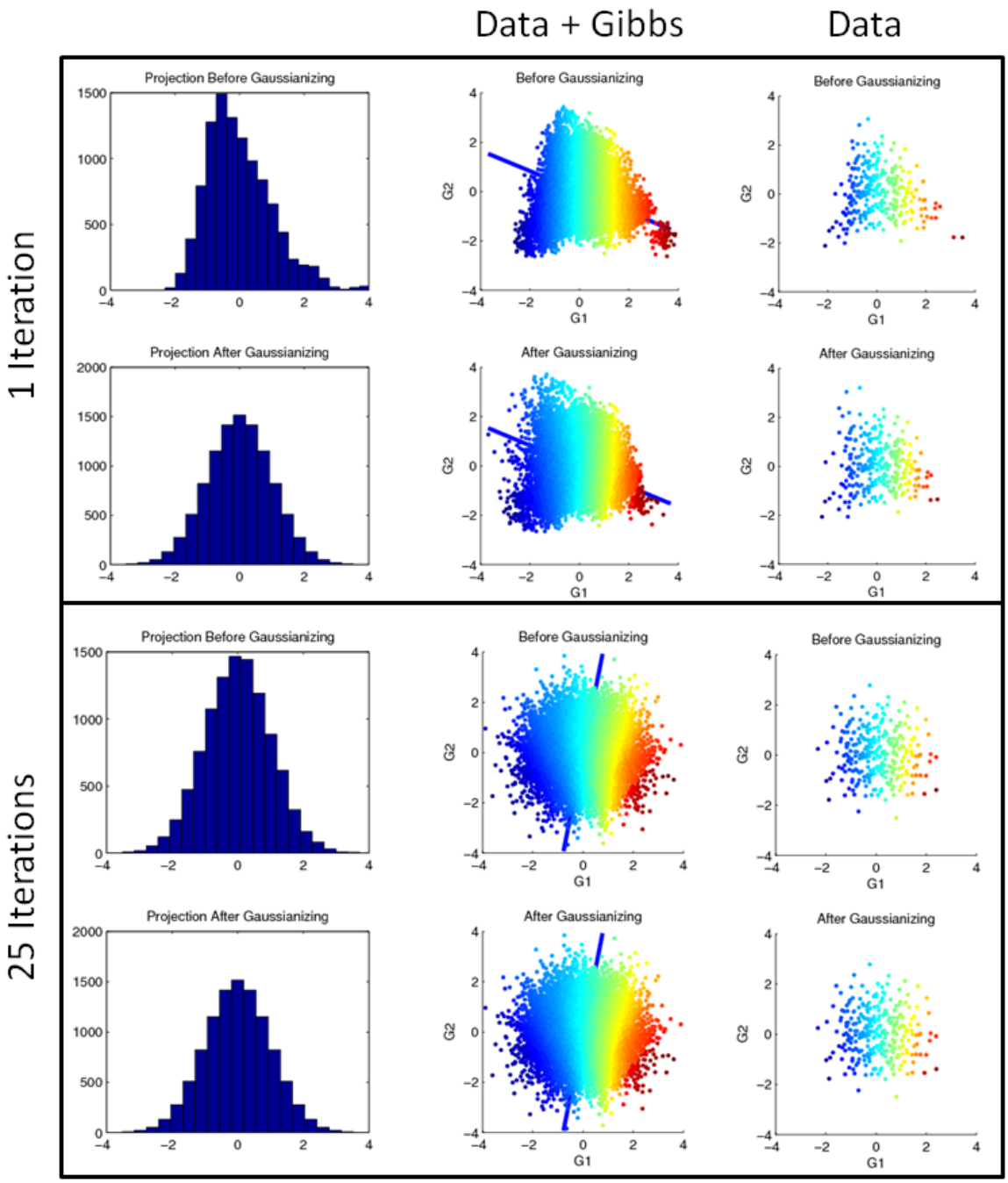



**Figure 10:** Projection histograms, and $X$ colored scatterplots for the first and twenty-fifth iteration of the PPMT where Gibbs sampled observations are included in the transformation along with the original data.

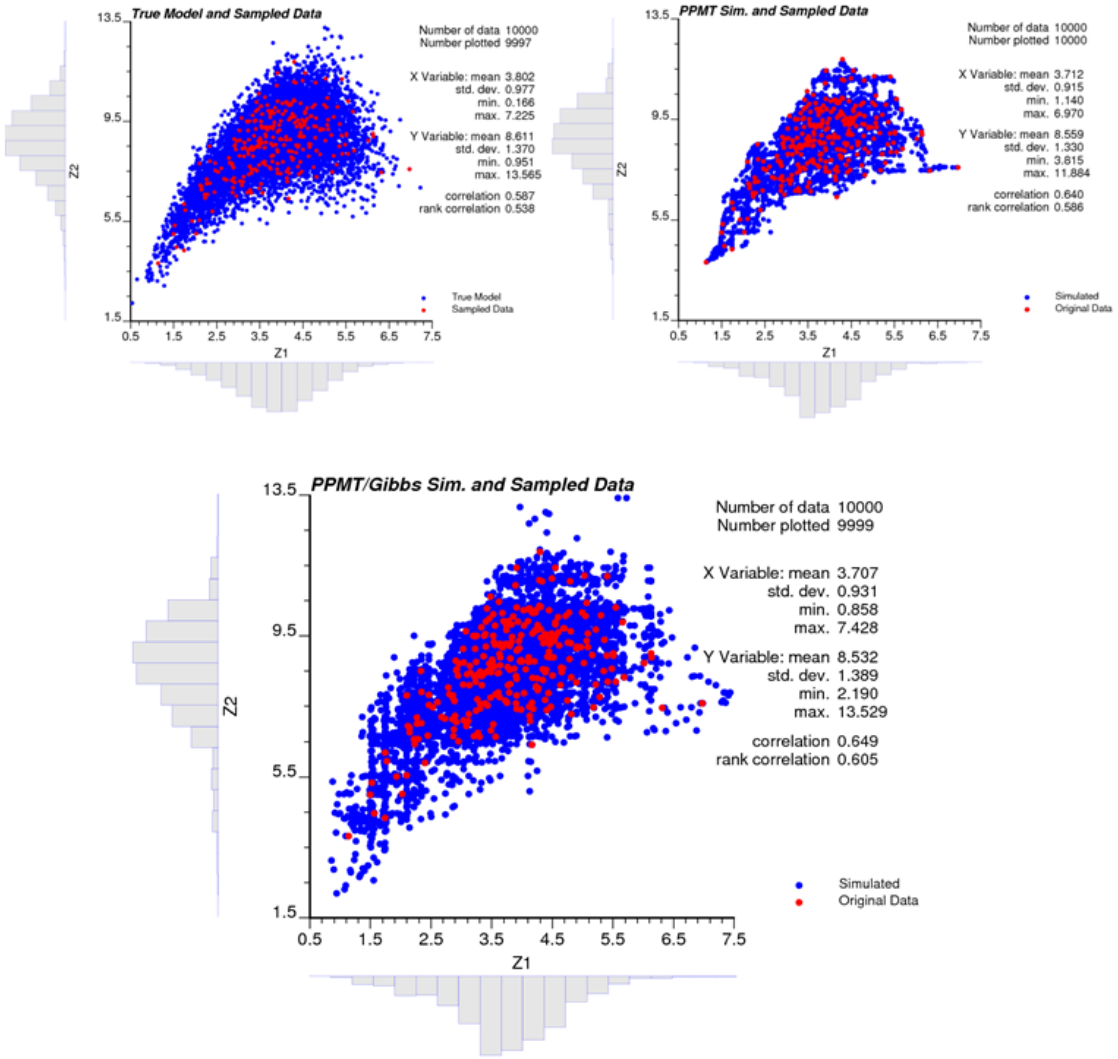



**Figure 11:** Scatterplots between the (i) True model (blue) and sampled data (red) (top right), (ii) simulated values *without* the use of Gibbs sampling observations (blue) and sampled data (red) (top left), and (iii) simulated values *with* the use of Gibbs sampling observations (blue) and sampled data (red) (bottom).

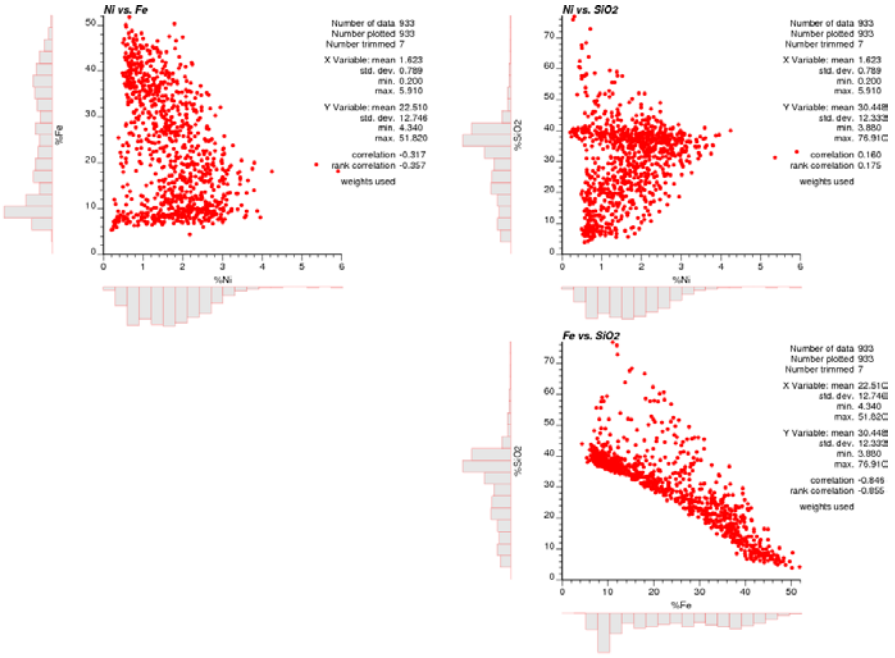



**Figure 12:** Scatterplots between Ni, Fe, and SiO2 for the Nickel laterite data.

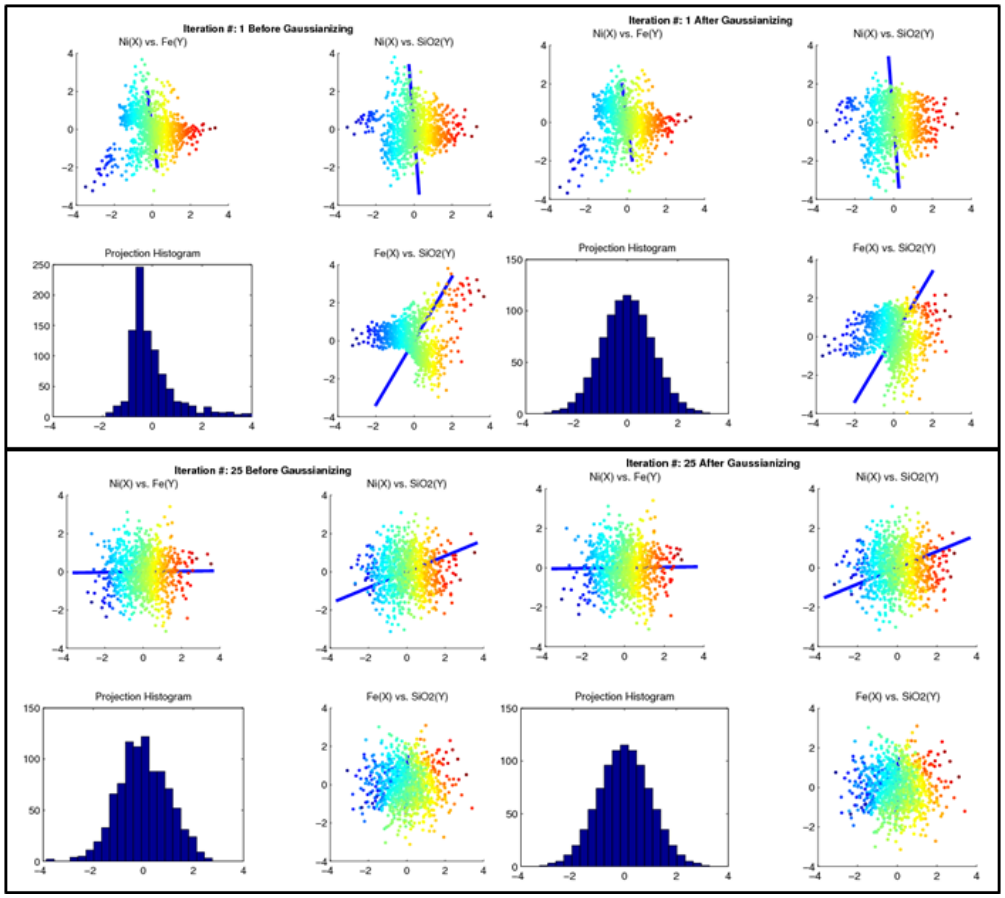



**Figure 13:** Each panel displays the projection histograms and scatterplots before and after Gaussianizing, for the first and 25$^{th}$ PPMT iteration. Each scatterplot is colored by the $X$ value of its respective x-axis variable, with the orientation of the maximum projection index vector given by the solid line.

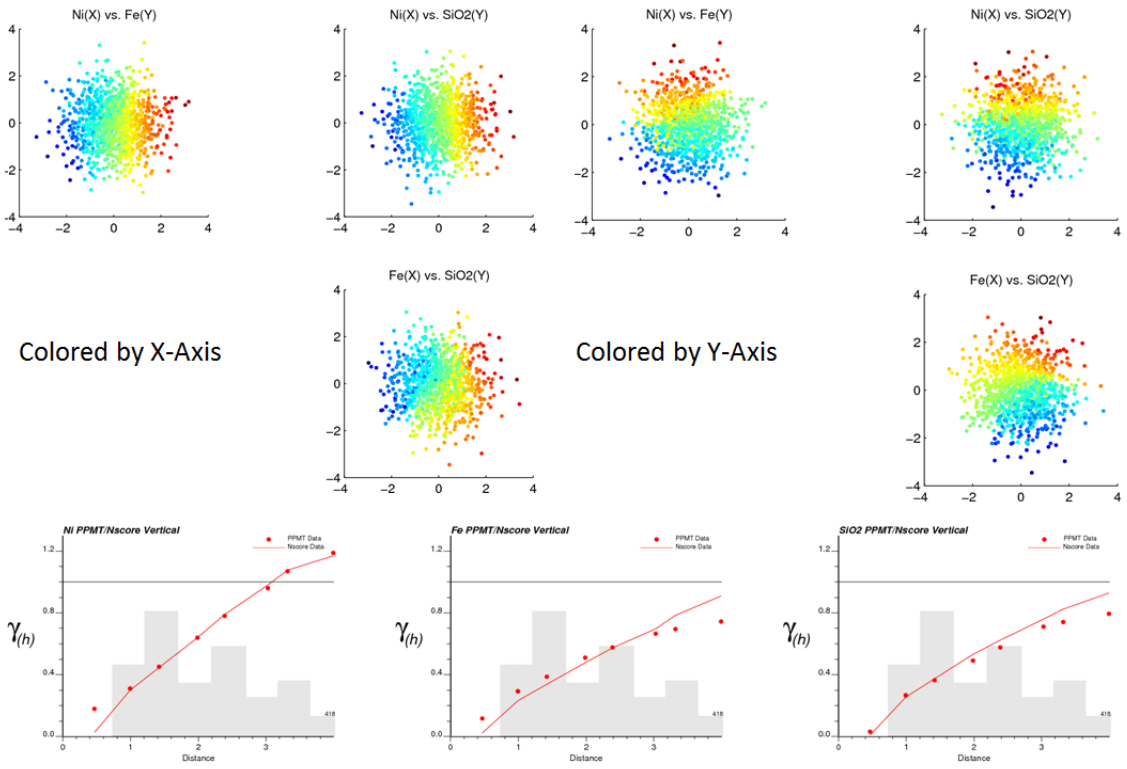



**Figure 14:** Scatterplots between the transformed Nickel laterite variables, where each point is colored by the $Y$ its respective x-axis variable. The omni-directional experimental semivariogram before (line) and after (dots) transformation are shown on the bottom. The relative number of pairs used in the calculation of each variogram is given by the grey histogram.

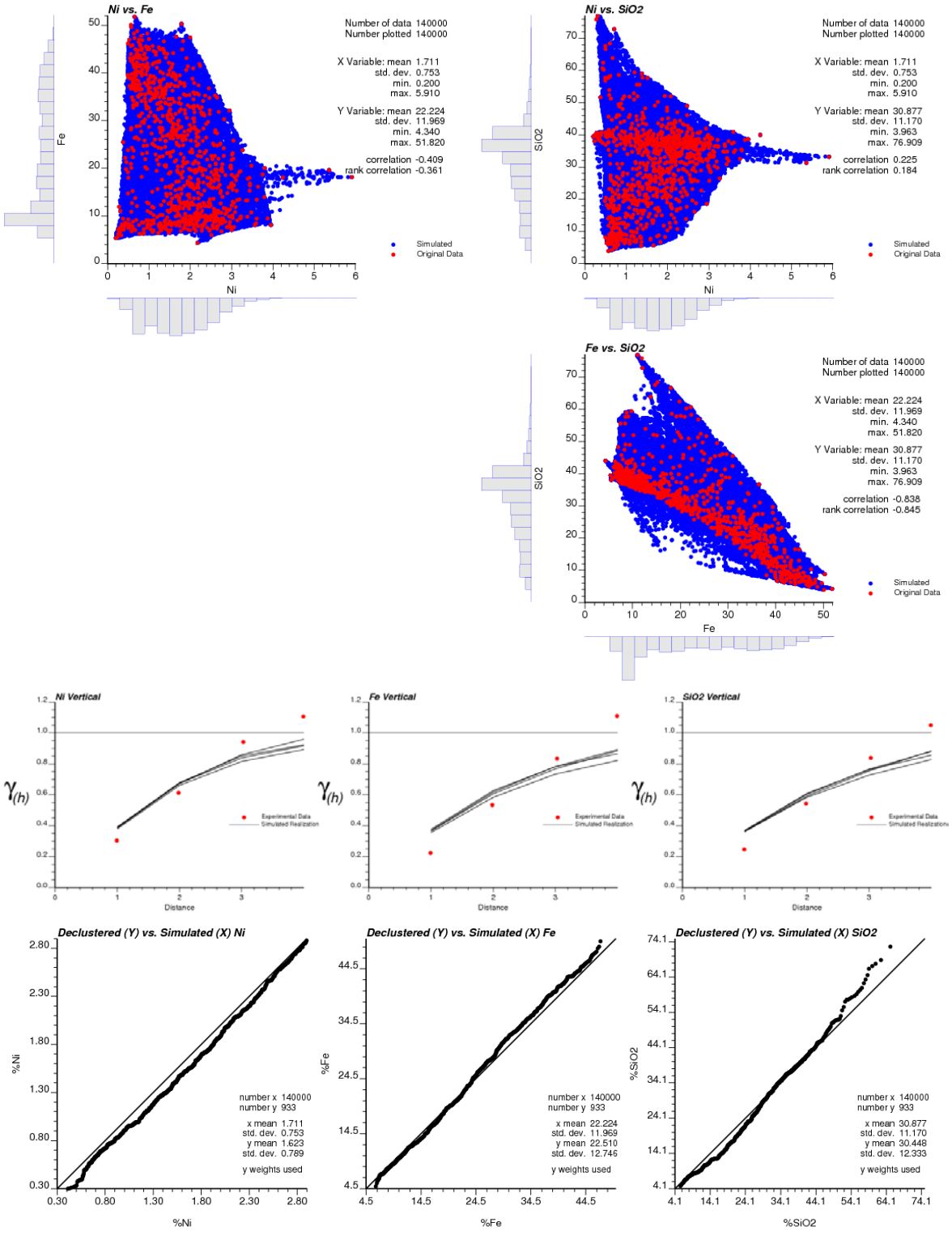



**Figure 15:** Selection of model validation plots. Scatterplots between the original data (red) and simulated values (blue) (top). Experimental semivariogram of the data (dots) and the simulated realizations (lines) (middle). Q-Q plots between the declustered data and simulated realizations (bottom).

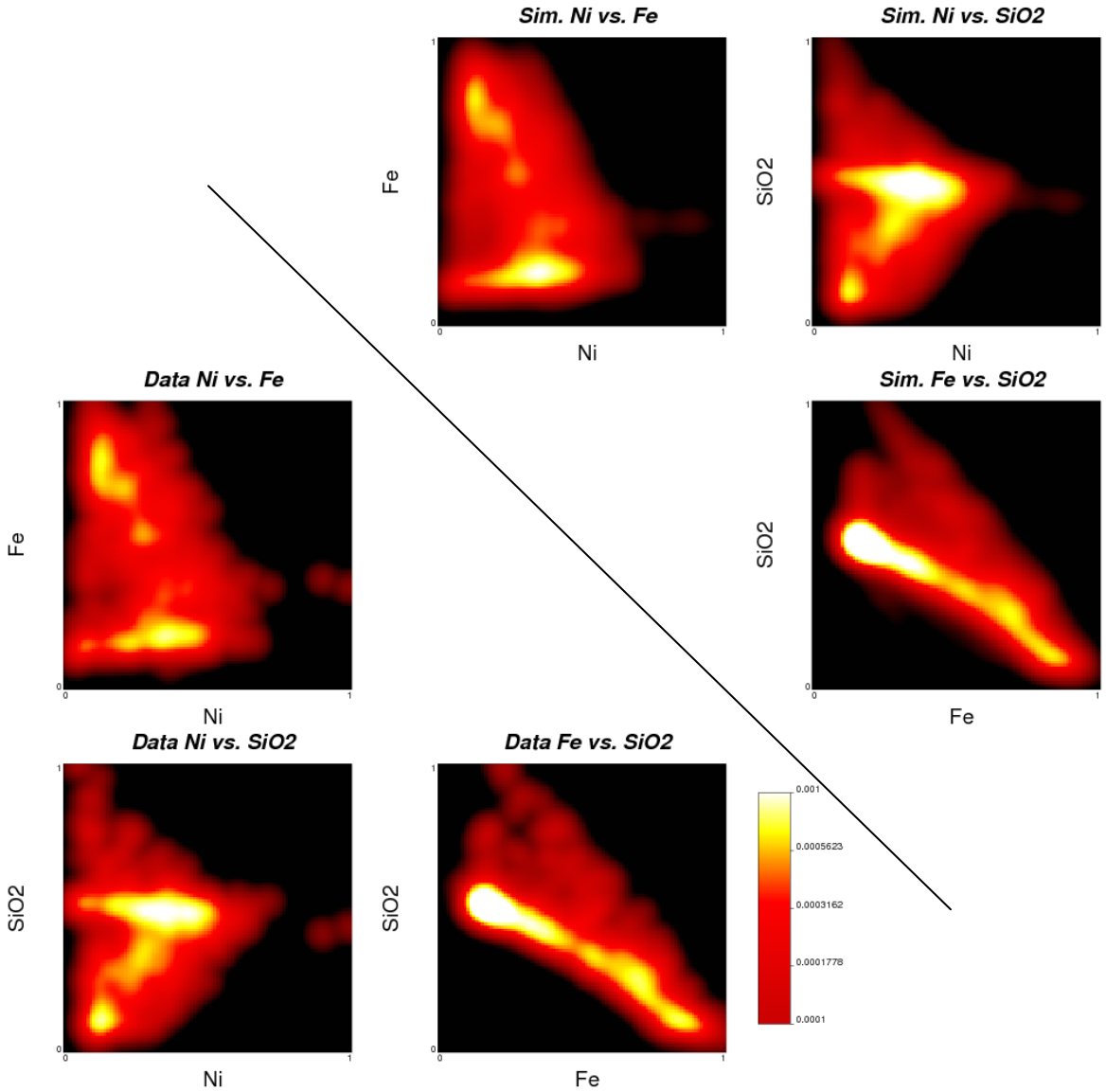



Figure 16: Gridded bivariate Gaussian KDE of the original data (bottom covariance matrix triangle) and four back-transformed realizations (upper covariance matrix triangle).

**Appendix**

The PPMT MATLAB function was written as prototype code for a FORTRAN based standalone executable. This function is seen being called from the script in Figure 17 (line 15), with the lines above specifying the required input parameters in a format mimicking CCG executables. Likewise, input and output data files are Geo-EAS formatted files. Treating this function calling script as an executable par file, the input parameters are described below:

- **datafl:** file with the input data to be transformed
- **vcols(i), i=1,…,nvars:** column locations within the **datafl** for the variables to transformed. The **nvars** number of variables is calculated based on the length of this input vector
- **tmin,tmax:** trimming limits that applied to all variables
- **maxiters:** the number of PPMT iterations (since stopping criteria is not yet defined)
- **igibb:** toggles whether Gibbs observations will be included with data observations in the PPMT mapping **(0=no,1=yes)**
- **gibbfl:** file containing Gibbs observations (required if **igibb=1**)
- **outfl:** file containing input data with the transformed variables appended (only the data with no Gibbs observations)
- **trnfl:** file containing the original and transformed observations (data and Gibbs observations). This must be referenced for the back-transform if **igibb=1**

```
1
2                    %Parameters for PPMT%
3                    %%%%%%%%%%%%%%%%%%%%
4
5      %START OF PARAMETERS:
6 -    datafl='nscore.out';                  % file with data
7 -    vcols=[3 4];                          % variable columns
8 -    tmin=-5; tmax=5;                      % trimming limits
9 -    maxiters=25;                          % maximum number of iterations
10 -   igibb=0;                              % use gibbs observations?
11 -   gibbfl='nofile';                      % file with gibbs observations data
12 -   outfl='ppmt.out';                     % output file for transformed data
13 -   trnfl='ppmt.trn';                     % output file for transform table
14
15 -   ppmt(datafl,vcols,tmin,tmax,maxiters,...
16                                  igibb,gibbfl,outfl,trnfl);
```

**Figure 17:** PPMT Matlab function parameters.