

Slycat Ensemble Analysis of Electrical Circuit Simulations

Patricia J. Crossno, Timothy M. Shead, Milosz A. Sielicki, Warren L. Hunt, Shawn Martin, and Ming-Yu Hsieh

1 Ensembles and Sensitivity Analysis

With recent advances in computational power, scientists can now run thousands of related simulations to explore a single problem. We refer to such a group of related simulations as an *ensemble*. More generally, an ensemble can be thought of as a set of samples or observations, each consisting of the same set of variables, in a shared high-dimensional space describing a particular problem domain. Practically, an ensemble is a collection of data sets with common attributes that we wish to analyze as a whole. Thus *ensemble analysis* is a form of meta-analysis that looks at the combined behaviors and features of the ensemble in an effort to understand and describe the underlying problem space. By looking at the collection as a whole, higher level patterns emerge beyond what can be seen by examining individual simulation runs.

As an example, *sensitivity analysis* is a type of ensemble analysis that evaluates how changes in simulation input parameters correlate with changes in simulation results. In addition to revealing the types and strengths of relationships between inputs and outputs, sensitivity analysis can be used to verify that simulation results are within expected ranges and to validate that the underlying model is behaving correctly. Unexpected results could point to unexplored physical regimes, poorly understood parameter spaces, or something as mundane as flaws in simulation codes. Input parameters form the set of independent variables, and outputs the set of dependent variables. Commonly, sensitivity analyses are performed using either simple regression (correlating a single input to a single output at a time), or multiple regression (correlating a group of inputs to a single output). However, neither of

P.J. Crossno (✉) • T.M. Shead • M.A. Sielicki • W.L. Hunt • S. Martin • M.-Y. Hsieh
Sandia National Laboratories, PO Box 5800, Albuquerque, NM 87185, USA
e-mail: pjcross@sandia.gov; tshead@sandia.gov; masieli@sandia.gov; wlhunt@sandia.gov;
smartin@sandia.gov; myhsieh@sandia.gov

these approaches provides a means for evaluating the collective relationships among multiple inputs and multiple outputs.

Our introduction to this work began while evaluating the likely impacts on workflows in analysis and visualization from proposed architectural changes for exascale computing. As part of the evaluation, we interviewed analysts working at Sandia National Laboratories in a variety of simulation domains, including thermal, solid mechanics, and electrical circuit analysis. Sensitivity analysis is a common component within each domain's work flow, although the simulation results vary widely, ranging from simple tables of metrics to time series and finite element models. The analysts typically use Dakota [2] to manage ensemble creation, using custom scripts to extract scalar metrics from finite element outputs or time series. The metrics are merged with tables of original input parameters and analyzed using Dakota, JMP, Matlab, Minitab, Excel, and other tools. Existing visualization tools such as ParaView [13] and EnSight [7] are used for remote visualization of large simulation results. However, these tools are fundamentally designed to visualize individual simulations, or handfuls of simulations that are loaded into memory simultaneously and visually superimposed. Ensembles containing hundreds or thousands of simulations require a different type of analysis, a different visual abstraction, and a different system architecture to effectively integrate so many results.

Our investigation led to the creation of Slycat, a system designed to meet the needs of ensemble analysis. For sensitivity analysis, and parameter studies in particular, Slycat provides a visual interface to answer the following questions about a given ensemble:

- Which input parameters are most strongly correlated with specific outputs?
- Which input parameters exhibit little or no impact on output responses?
- Which simulations are anomalous, and in what ways are they different?

We use Canonical Correlation Analysis (CCA) to model the relationships between input and output parameters because it maps well to the structure of our problem, especially in its ability to correlate multiple inputs against multiple outputs. Although powerful and ideally suited to the problem, CCA results can be difficult to interpret; making CCA accessible to domain experts through the tight integration of useful visualizations with iterative exploratory analysis is the central contribution of this work.

2 Related Work

The work that most closely aligns with our own is that of Degani et al., which applies CCA to the analysis of correlations between the operating environment of a Boeing aircraft and the actions and responses of the pilots [8]. Presenting the CCA correlations in a circular layout called a heliograph, the positive or negative correlation weights of different variables are represented as outward or inward

facing bars, respectively. The concentric stacking of CCA components in the same plot leads to overlapping bars from adjacent components, potentially leading to misinterpretations of the results. Our system provides a simpler, easier to understand visualization of CCA results as applied to multiple inputs and outputs.

Much of the earlier ensemble research deals with data sets that are either geospatial or spatiotemporal in nature. Consequently, their analysis and visualization approaches rely on this, making them unsuitable for ensembles that lack a spatial component. For example, Wilson and Potter explore geospatial weather data and discuss how ensembles mitigate uncertainty in simulations [31]. A similar approach from Potter employs isocontours over spatial domains [20, 21]. Noodles, another tool for modeling and visualizing uncertainty in ensembles of weather predictions, displays the spatial distributions of inter-simulation uncertainty through a combination of ribbons and glyphs drawn as map overlays [22]. Waser et al. integrate computational steering controls into a spatiotemporal flood simulation framework, enabling users to steer parameter studies and generate ensembles on demand [29, 30].

Another branch of ensemble visualization research uses feature extraction. However, these techniques can still have spatial dependencies. In the technique of Smith et al., clustering based on feature identification is performed on time-varying, spatial data [24]. A suite of feature detection techniques, including CCA, is used by Sukharev et al. to reveal structure in multivariate, time-varying climate data sets. Once their data has been clustered, segmented, and correlations computed, the results are geo-spatially overlaid on the weather prediction region for visualization and interpretation [27]. Hummel et al., classifies fluid transport variance within neighborhoods over flow field ensembles. Linked views enable selection in the fluid feature space to produce a visualization over the physical domain [11]. Another system, EDEN, incorporates several multivariate feature detection techniques in a single interface [26]. Piringer et al. visualize multivariate data using downsampling, 3D surface plots, extracted scalar features, and glyph-based visualizations to explore an ensemble of 2D functions. In addition to comparing ensemble member functions against each other, this work attempts to illustrate the distribution of features across the ensemble [19].

CCA has also been used to analyze spatial data. To more clearly identify the relationships in functional magnetic resonance imaging data, Karhunen et al. have developed a method that exploits CCA prior to applying blind source separation techniques [12]. The research of Ge et al. demonstrates CCA correlations that reflect the spatial correlations in multiple sensor arrays, even in the presence of noise [9]. In a recent paper by Marzban et al., CCA is shown to capture complex weather relationships between model parameters and forecast quantities [15].

More broadly, parameter studies have a relationship to our work, though their intent is often either based on the design of experiments or on directing simulation results towards a particular outcome, neither of which aligns with our particular system goals. Examples include systems for exploring the relationship between input parameters and aesthetic visual effects in animations [5], and the steering of designs [6].

Matkovic et al. recognize the need for advanced tools to support engineers in visualizing and understanding ensembles, and incorporate in their system multivariate visualization techniques such as parallel plotting, multiple linked views, and scatterplots to display one-to-one correlations [16]. However, one-to-one correlation analysis is insufficient for evaluating the complex, multivariate relationships inherent in our user's data.

Within sensitivity analysis, sampling tools are typically relied on to provide coverage of the simulation parameter space [1, 2]. Even with a sampling method in place, much system behavior is unknown and there is more work to do to uncover input-output relationships. A study by Song and Zhao employs a variance-based method to identify the first-order model sensitivities when applied to forest growth simulations [25]. Other sensitivity analysis work applies classic statistical methods, such as ANOVA, to U.S. immigration model results [23], and statistical aggregation to models of large distributed computing systems [17]. In contrast, our research is focused on understanding the behavior of an ensemble with the intent of exposing hidden relationships between the simulation input parameters and the results, without the emphasis on numerical quantification of uncertainty.

3 System Architecture

To support answering the questions outlined in Sect. 1, and to support additional analysis types in the future, we designed Slycat around the following general requirements:

- Remote ensemble analysis, in which large data is analyzed in-place to minimize data movement.
- Ubiquitous access to analysis results regardless of the availability of the original data sets or source platforms.
- Desktop delivery providing interactive exploration of ensemble analysis results, and collaborative sharing with appropriate access controls.

The need for remote ensemble analysis is driven by the ever widening gap between high performance computing (HPC) computational performance and I/O performance. Practically speaking, we have reached a point where computation is effectively “free” while data movement has become very expensive, and moving raw ensemble data from the HPC platform where it was generated to the host running Slycat would take significantly more time than the analysis computations to follow! Better instead to perform those computations on the machine where the ensemble data is located, so that only the model—typically orders of magnitude smaller than the original data—is moved across the network to the Slycat host. This leads to the Slycat architectural design of Fig. 1.

An important practical consideration for users of HPC platforms is that ensemble results may often become temporarily or permanently unavailable—login nodes come and go due to resource contention, users often must archive or delete their

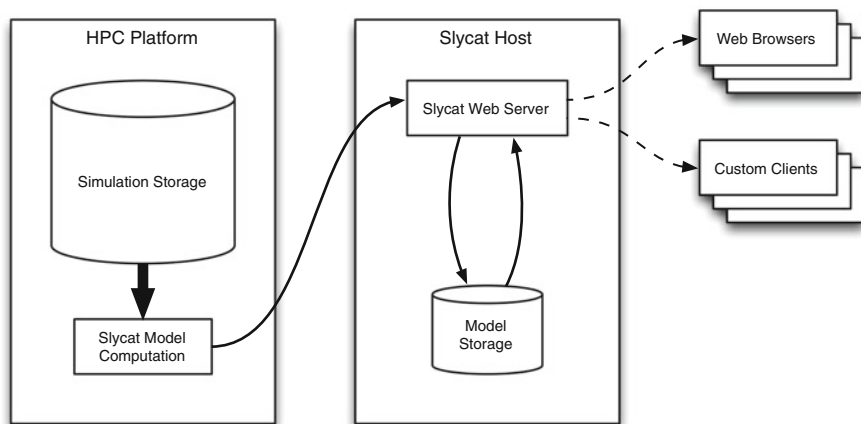


Fig. 1 Slycat system diagram depicting how large data on an HPC platform is analyzed in-place to produce greatly reduced model artifacts that are stored by the Slycat web server. Later, these artifacts are delivered—incrementally and on-demand—to interactive clients

data as scratch filesystems near capacity, and so on. Because Slycat stores its own greatly-reduced models of the underlying raw data, and only those models are necessary to produce a visualization, users can continue to access their Slycat analysis results even when the original HPC resources are unavailable.

Finally, we want a system architecture that supports easy desktop delivery and collaboration, enabling users to share results seamlessly with colleagues across the network without requiring any software downloads or installation. That means using existing web standards and clients, dictating much of the subsequent design and derived requirements. It means adopting a web server as the front-end for the system and standard web browsers as clients (or custom clients using standard web protocols to communicate). In turn, interactions and visualizations must be developed using only the set of technologies that are widely available within web browsers, such as HTML5, JavaScript, AJAX, SVG, and Canvas.

Unlike dedicated visualization tools such as ParaView or Ensight, we cannot rely on the client to perform serious calculations. This necessitates Slycat’s pre-computation of visualization artifacts, organized for rapid, incremental retrieval from the server on-demand. As an example, we allow users to interact with data tables that can contain thousands of columns and millions of rows - which would cause unacceptable slowdowns if they had to be transferred from server to client in their entirety. Instead, only the data needed to display the currently-visible table rows are transferred as the user scrolls through the table. This “just-in-time” approach to data is used throughout the client, minimizing total bandwidth consumption and keeping the interface responsive.

Although working around the constraints of web browsers has been a challenge, the rewards have been significant, enabling Slycat users to “bookmark” the state of a visualization and share it with colleagues simply by sharing a hyperlink.

4 Canonical Correlation Analysis

H. Hotelling, who was also instrumental in developing Principal Component Analysis, proposed Canonical Correlation Analysis (CCA) [10]. CCA is a method that can be used to understand relationships between two sets of multivariate data. Writing our sets as matrices, $X = [\mathbf{x}_1, \dots, \mathbf{x}_n]_{p \times n}$ is presumed to be independent, and $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n]_{q \times n}$ to be dependent, where the \mathbf{x}_i and \mathbf{y}_j lie in \mathbb{R}^p and \mathbb{R}^q , respectively (i.e. each vector \mathbf{x}_i has p components and each vector \mathbf{y}_j has q components). CCA attempts to find projections \mathbf{a} and \mathbf{b} such that $R^2 = \text{corr}(\mathbf{a}^T X, \mathbf{b}^T Y)$ is maximized, where $\text{corr}(\bullet, \bullet)$ denotes the standard Pearson correlation, which in our case is given by

$$\text{corr}(\mathbf{a}^T X, \mathbf{b}^T Y) = \frac{\mathbf{a}^T C_{xy} \mathbf{b}}{\sqrt{\mathbf{a}^T C_{xx} \mathbf{a} \mathbf{b}^T C_{yy} \mathbf{b}}},$$

where C_{xx} is the $p \times p$ covariance matrix $\frac{1}{n} X X^T$, C_{yy} is the $q \times q$ covariance matrix $\frac{1}{n} Y Y^T$, and C_{xy} is the $p \times q$ covariance matrix $\frac{1}{n} X Y^T$. (Note that we are assuming for convenience that X and Y are mean-subtracted and unit variance.)

Thus to find the projections \mathbf{a} and \mathbf{b} we want to solve

$$\begin{aligned} &\text{maximize } \mathbf{a}^T C_{xy} \mathbf{b} \\ &\text{subject to } \mathbf{a}^T C_{xx} \mathbf{a} = \mathbf{b}^T C_{yy} \mathbf{b} = 1. \end{aligned}$$

This problem reduces to a generalized eigenvalue problem, and thus has a unique, global minimum (up to subspace isomorphism and assuming adequate matrix ranks). CCA is a linear method and is a direct generalization of several standard statistical techniques, including PCA, multiple linear regression (MLR), and Partial Least Squares (PLS) [3, 4].

The vectors $\mathbf{a}^T X$ and $\mathbf{b}^T Y$ are known as the first pair of canonical variables. Further pairs of canonical variables are orthogonal and ordered by decreasing importance. In addition to the canonical variables, the R^2 value for each variable pair is obtained, and various statistics can be computed to determine the significance of the correlation. A common statistic used in this context is the p -value associated with Wilks' λ [14].

Once the canonical variables are determined, they can be used to understand how the variables in X are related to the variables in Y , although this should be done with some caution. The components of the vectors \mathbf{a} and \mathbf{b} can be used to determine the relative importance of the corresponding variables in X and Y . These components are known as canonical coefficients. However, the canonical coefficients are considered difficult to interpret and may hide certain redundancies in the data. For this reason, it is more typical to analyze the canonical loadings, also known as the structure coefficients. The structure coefficients are given by the correlations between the canonical variables and the original variables (e.g. $\text{corr}(\mathbf{a}^T X, X)$). The structure coefficients are generally preferred over the

canonical coefficients due to the fact that they are more closely related to the original variables.

CCA is a powerful method that is particularly useful for certain problems because it acts on pairs of data sets (unlike PCA, which acts on a single data set). However, it is also complex and difficult to interpret. One of our central goals in this work has been to provide CCA-based analysis for data ensembles, but in a framework more understandable to domain experts.

5 Visualization

In Slycat, sensitivity analysis is performed through an iterative cycle of designating which input and output variables to include in the analysis, performing CCA, and visually exploring the resulting model. Analysis typically starts with an all-to-all evaluation to get an initial sense of the data, revealing the most strongly correlated combinations of variables. Some cases require iterative refinement to tease apart disparate groups of inputs and outputs.

To provide integrated perspectives, Slycat combines three levels of detail in a single web page using multiple linked views, as shown in Fig. 2. In the upper left, the *Correlation View* shows the relationships found for the ensemble as a whole, displaying structure coefficients in tabular form, grouped by correlation components

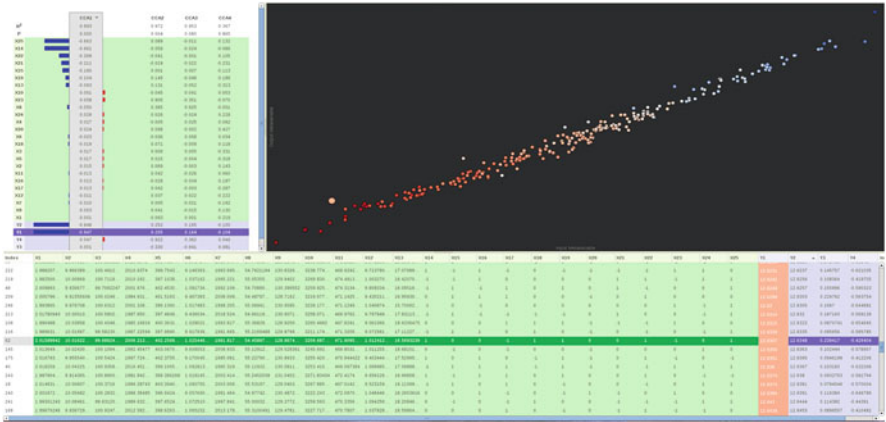


Fig. 2 Slycat visualization showing an all-to-all CCA analysis of a 250 simulation electrical circuit ensemble. As seen through the lengths and shared colors/directions of the bars in the *Correlation View* in the *upper left*, the first CCA component exhibits a strong positive correlation predominantly between the combined inputs X25 and X14 and both of the outputs Y1 and Y2, an example of a many-to-many relationship. In the *Simulation View* scatterplot on the *upper right*, the highlighted simulation (enlarged point) on the *lower left* can be seen to be anomalous given its offset from the diagonal and its lower Y1 value (*pink*) relative to other simulations with similar inputs (*reds*)

into columns and by ensemble variables into rows. The top two rows display each component's R^2 and p -value. Variable names are shown along the left edge of the view. Rows for input variables are colored green, while output variable rows are lavender. This green/purple color coding is used consistently throughout the interface to designate inputs and outputs. The rows in each column can be sorted by the correlation strengths of the variables in either ascending or descending order.

Users select a component by clicking its column header, expanding the coefficients into an inline vertical bar chart. The bars visually encode the signed values of the structure coefficients for each variable, with left-facing blue bars representing negative values and right-facing red bars representing positive ones. Color coding the bars visually reinforces the relationship types. Variables with matching colors are positively correlated, and variables with mismatched colors are negatively correlated. Bar length indicates a variable's relative importance. Sorting a component in descending order separately displays the input and output variables in order of significance with the longest bars at the top and the shortest at the bottom. This ordering makes it simple to evaluate which variables exhibit strong correlations, whether the correlations represent positive or negative relationships between the variables, and which inputs are driving a particular group of results.

The scatterplot in the upper right of Fig. 2 is the *Simulation View*. It displays how well individual simulations are described by the correlations for the ensemble as a whole. The axes are the canonical variables, $\mathbf{a}^T X$ and $\mathbf{b}^T Y$, and each simulation is rendered as a point with x and y coordinates computed as sums of input and output values, respectively. This coordinate space is highly abstract, with the x -axis representing a metavariable that is the sum of all the inputs and the y -axis a metavariable of all the outputs. Consequently, the scatterplot changes whenever a new canonical component is selected, since each component's structure coefficients are different. If the model finds a perfect linear correlation between inputs and outputs, the scatterplot points will form a diagonal line. Anomalous simulations will appear offset from the diagonal as positional outliers.

Points can be color coded by the values of input or output variables, providing another way to identify outliers. Selecting a row in the *Correlation View* or a column header in the *Variable Table* (see below) selects that variable's values for color coding in the scatterplot. We use a diverging blue/white/red color map [18] to encode the values, where blue is at the low end of the scale, and red is at the high end. While we do not assign any particular meaning to the central values shown in white, we find in practice that the diverging color map makes it easier to interpret values of nearby points.

Across the bottom of the page, the *Variable Table* displays raw data with a row for each individual simulation and a column for every ensemble variable. Column (variable) selections in the *Variable Table* are linked with row selections in the *Correlation View* and the scatterplot color map. The selected column is color coded using the same color map as the scatterplot to visually correlate the two views. The table columns can be sorted in ascending or descending order. Additionally, row (simulation) selections are linked with point selections in the scatterplot. Darker green/purple backgrounds highlight selected table rows, while selected points in the scatterplot are drawn with a larger radius.

6 Electrical Simulation Sensitivity Analysis

Our users model electrical circuits using Xyce, an open source, high-performance circuit simulator developed by Sandia National Laboratories as part of the Advanced Simulation and Computing (ASC) Program [28]. We will examine two circuit ensembles of differing scales: a small ensemble of 250 runs, and a large ensemble of 2,641 runs. In both cases, some of the input variables take a restricted set of values (-1 , 0 , or 1). These values are used to select different simulation models whose responses are *low*, *nominal*, and *high*, respectively. The models act to encapsulate groups of input variables, thereby reducing the number of variables and the size of the ensemble.

6.1 Small Ensemble

The small ensemble has 250 simulations, each with 25 input variables and 4 output variables. Outputs Y1 and Y2 measure voltage responses, while Y3 and Y4 measure current. The goal of this analysis is to answer the first two questions from the list in Sect. 1: Which inputs are most strongly correlated with specific outputs, and which inputs exhibit little or no impact on the outputs?

As seen in the *Correlation View* bar chart in Fig. 2, the first CCA component shows a positively correlated relationship that is mostly between the input parameters X25 and X14 and both of the voltage outputs. The inputs are listed in decreasing order of importance, so the parameters at the bottom of the green region in the first column exhibit little or no impact on voltage responses.

In Fig. 3, the sorted variables for the second CCA component reveal a strong negative correlation predominantly between the input X23 and the current response, Y4. Color coding the simulations first by the input values (upper image), then by the outputs (lower image), we can see a strong correspondence between the *low*, *nominal*, and *high* values of X23 and groups in the Y4 response values. Although the other current response, Y3, is present in the second CCA component, it is more strongly described by the third component, as shown in Fig. 4. The central relationship there is a negative correlation between the input X8 and the output Y3.

6.2 Large Ensemble

The large ensemble is roughly two orders of magnitude larger than the small ensemble, containing 2,641 simulations of a different electrical circuit with 266 input variables and 9 outputs. The outputs for this circuit are more varied than the previous circuit, capturing events and features rather than simple voltages or currents. Slycat easily scales to handle this data, and our largest ensemble to date has contained more than 500,000 simulation runs.

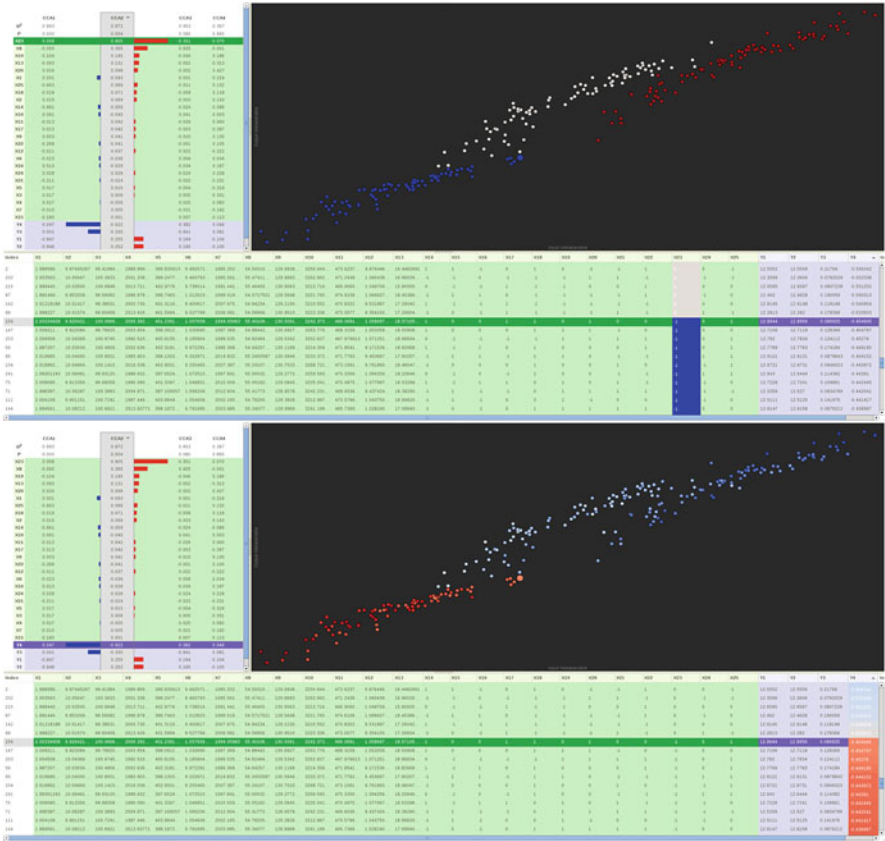


Fig. 3 In the second CCA component, the correlation is predominantly a negative relationship between input X23 and output Y4. Color coding the scatterplot by the values of X23 (*top image*) and Y4 (*bottom image*), we can see a one-to-one correspondence between the *low* values of X23 and the *high* Y4 responses. The *nominal* and *high* values of X23 combine into a single low-valued group in Y4. In both images, table rows are sorted by Y4 value. Note the selected (enlarged) simulation point near the scatterplot center, which lies on the boundary between the two groups

Given the large number of input variables, an initial analysis goal is to reduce the number of variables required to drive the simulations. Using a process similar to that of the previous section, we are able to cull the number of input variables needed to drive future simulations of this circuit from 266 down to 21.

Finally, we demonstrate how Slycat can be used to answer the third analysis question from Sect. 1: which simulations are anomalous, and in what ways do they differ from the norm? In our initial all-to-all analysis, we noticed four anomalous simulations in the upper part of the scatterplot, the highlighted red points shown in Fig. 5. Distance from the diagonal is a metric for how well the linear correlation found by CCA describes a particular simulation, so these four runs immediately stand out as anomalous. What sets them apart? Since the Y-axis in the scatterplot is

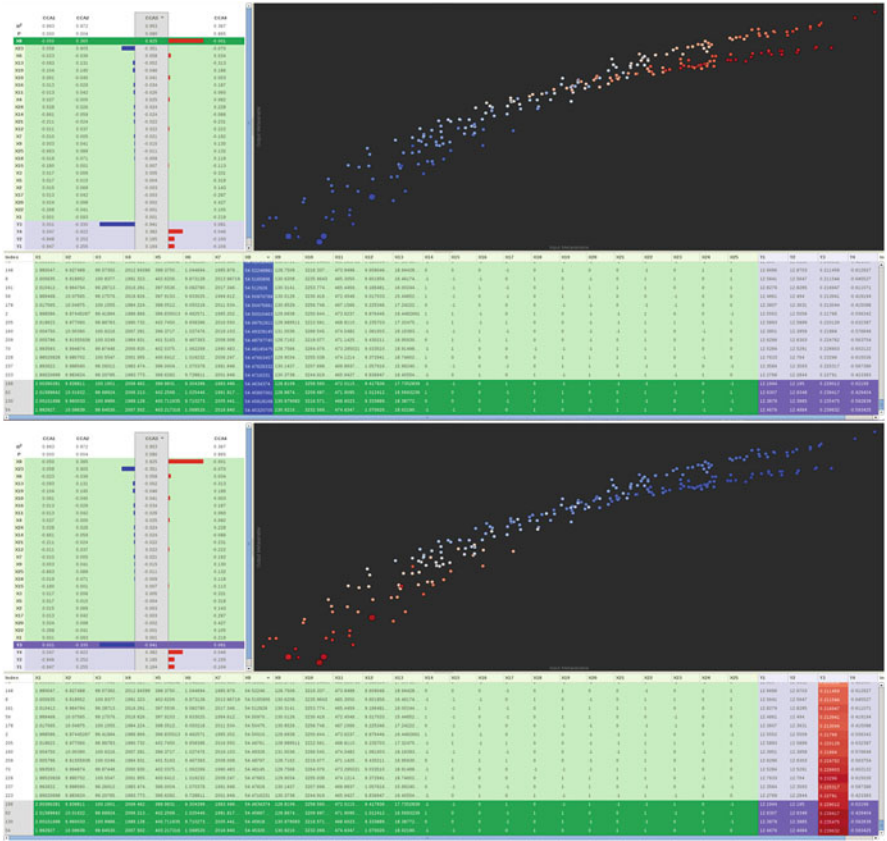


Fig. 4 In the third CCA component, the input parameter X8 is negatively correlated with the current response Y3. Color coding the scatterplot by the values of X8 (*top image*) and Y3 (*bottom image*), we can see the negative relationship between low values (*dark blue*) in the input parameter and high values (*dark red*) in the output. In both images, the table rows are sorted by decreasing X8 values and we have selected the four simulations with the lowest values in X8

a metavariable based on the simulation outputs, vertical placement is a function of output variable values. Interactively switching the color coding between the various outputs, we discover that the Y4 values for these four simulations are at the high end of the scale. Sorting the Y4 values in the table, we see that these four simulations have Y4 values that are distinctly higher than any of the others (notice that they are in red, while the next largest values are in orange).

Next we investigate similarities amongst the four simulations' inputs, hypothesizing a common factor leading to the higher responses. We perform a many-to-one CCA analysis between all of the inputs and Y4. The two most highly correlated input variables, X248 and X255, both have identical values for all four simulations. However, each of these variables provides a range of other Y4 responses for the same input values, as seen in Fig. 6, so neither variable in isolation is the

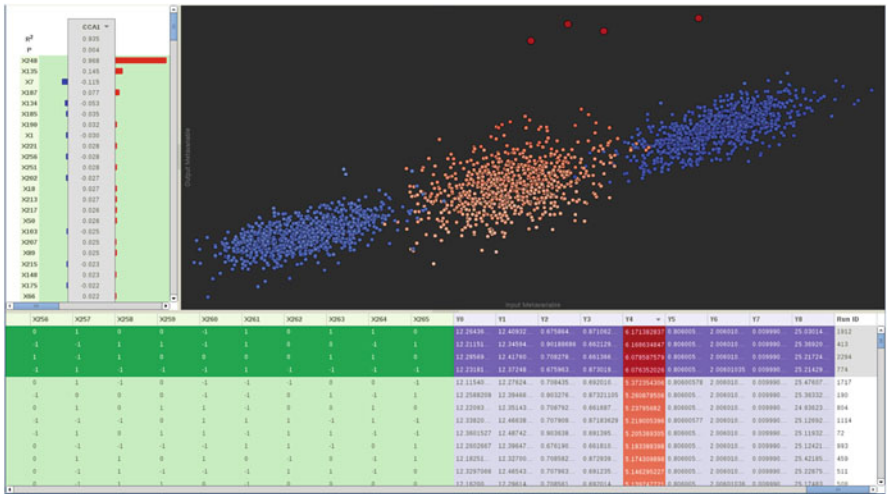


Fig. 5 In the first CCA component of the all-to-all analysis of the large ensemble, anomalous runs (in red) are highlighted near the top of the scatterplot. We initially notice them based on their position. The vertical position indicates that the difference between these simulations and the others is based on one of the outputs. *Color coding*, combined with sorting the table, shows that these four simulations have much higher values in Y4 than any of the other simulations. Note that since color encodes value, selection in the scatterplot is shown by increased point radius

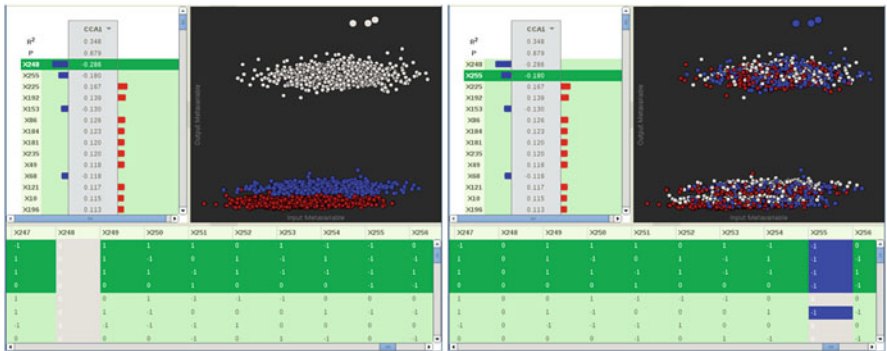


Fig. 6 We perform an all-to-one analysis to discover which input variables are most highly correlated with Y4, and which could be driving the four anomalous output values. The top two variables, X248 (*left image*) and X255 (*right image*), have identical input values for the four runs, but neither is the sole driver, since both demonstrate the same inputs driving a variety of outputs. The anomalous outputs must stem from a combination of inputs acting in concert

cause. Using the table, we find nine variables that share identical values for all four simulations (X248, X255, X224, X175, X176, X187, X196, X213, and X229). Given X248's strong correlation with Y4, it is definitely involved. Further simulation by our collaborators is needed to isolate which additional input variables

are involved, but Slycat allowed us to narrow down the possibilities to a handful of variables.

7 Performance

The use cases that we have presented each took less than a minute to compute. To better characterize the performance of Slycat’s CCA implementation, we synthesized a series of test ensembles, varying the number of observations and variables in each, then timed how long it took to upload the data into Slycat and compute a CCA model.

The test hardware included a Mac OSX workstation client that generated the ensembles, uploading them via a 10 Gb ethernet network to a Linux server running the Slycat software with eight 2.4 GHz Quad-Core AMD Opteron processors and 78 GB of RAM. Each test was run three times and the timings averaged. Note that some combinations of observations and variables could not be tested, either because there were too few observations for a given number of variables, or because the CCA computations for the combination would have exceeded the available memory on the server.

During ingestion, the data was uploaded one-variable-at-a-time to avoid exceeding request size limits imposed by HTTP. Reading from left-to-right in Table 1, the ingestion times are almost perfectly linear in the number of variables uploaded. Reading top-to-bottom, we would expect similarly linear behavior as we increase the number of observations; however, the timings are complicated by other factors, including overhead for HTTP request handling, database access, and disk I/O for each upload.

Each test dataset contained an even number of variables, and we configured the CCA computations to split them into equal numbers of inputs and outputs. This configuration ensured that the CCA computation would produce the maximum number of CCA components possible for a given dataset. From Table 2, we see that increasing the number of variables (reading left-to-right) has a larger impact on runtimes than increasing the number of observations (reading top-to-bottom).

Table 1 Data ingestion times (s)

Observations	Variables								
	4	8	16	32	64	128	256	512	1024
10	1.901	2.6							
100	1.897	2.583	4.114	7.019	13.17				
1,000	1.967	2.607	4.13	7.112	13.08	25.96	51.61	104.7	
10,000	2.191	2.998	4.735	8.577	16.13	31.94	63.48	133.8	276.9
100,000	3.238	5.567	7.996	15.43	29.29	57.35	116.9	233.7	
1,000,000	12.53	24.66	48.54	96.02	193	381.7	761.4		

Table 2 CCA compute times (s)

Observations	Variables								
	4	8	16	32	64	128	256	512	1024
10	2.515	2.724							
100	2.516	2.71	3.289	4.489	7.025				
1,000	2.642	2.73	3.326	4.562	7.146	13.82	30.35	85.84	
10,000	2.667	2.827	3.576	4.907	8.023	18.13	74.87	279.4	966
100,000	2.812	3.417	5.952	14.06	35.13	120.4	419.6	1791	
1,000,000	4.727	9.361	26.58	86.04	317.2	1176	5419		

This is consistent with our expectations for the CCA implementation. Since CCA solves an eigenvalue problem based on a covariance matrix, its expected complexity is $O(n) + O(p^3)$, where n is the number of observations and p is the number of variables.

Conclusions and Future Work

We have demonstrated how Slycat meets our design goals, illustrated its utility on two real-life electrical circuit analysis examples of varying scale, and presented performance results for a series of synthetic test ensembles. Slycat’s linked views display multiple levels of abstraction from high level ensemble-wide context to deep exploration of relationships at the level of individual simulation inputs and outputs. This combination of iterative analysis and visualization makes CCA more approachable, allowing users to interactively develop and test hypotheses about relationships among variables.

As of this writing, Slycat includes a second analysis type based on time series clustering, which allows us to directly analyze time series data such as the voltage and current waveforms that were reduced to individual output features in our examples. We are also considering a new type of analysis model based on factoring of arbitrary-dimension tensors. The Slycat source code and documentation are freely available under an open source license at <https://github.com/sandialabs/slycat>, and we welcome collaborators interested in incorporating their own models into Slycat.

Acknowledgements The authors wish to thank Ben Long at Sandia National Laboratories for sharing his work processes, expertise, and data. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. Abdellatif, A.S., El-Rouby, A., Abdelhalim, M.B., Khalil, A.: Hybrid Latin hypercube designs. In: The 7th International Conference on Informatics and Systems (INFOS), pp. 1–5 (2010)
2. Adams, B.M., Ebeida, M.S., Eldred, M.S., Jakeman, J.D., Swiler, L.P., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Hu, K.T., Vigil, D.M., Bauman, L.E., Hough, P.D.: Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.3.1 user's manual. Tech. Rep. SAND2010-2183, Sandia National Laboratories (2013)
3. Anderson, T.W.: An Introduction to Multivariate Statistical Analysis, 3rd edn. Wiley, New York (2003)
4. Borga, M.: Learning multidimensional signal processing. PhD thesis, Linköping University, Linköping (1998)
5. Bruckner, S., Möller, T.: Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1468–1476 (2010)
6. Coffey, D., Lin, C.L., Erdman, A., Keefe, D.: Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE Trans. Vis. Comput. Graph.* **19**(12), 2783–2791 (2013)
7. Computational Engineering International I: URL <http://www.ceissoftware.com/> (2013)
8. Degani, A., Shafto, M., Olson, L.: Canonical correlation analysis: Use of composite heliographs for representing multiple patterns. In: Diagram 2006, Lecture Notes in Artificial Intelligence, vol. 4045. Springer, pp. 93–97 (2006)
9. Ge, H., Kirsteins, I., Wang, X.: Does canonical correlation analysis provide reliable information on data correlation in array processing? In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2009 (ICASSP 2009)
10. Hotelling, H.: Relations between two sets of variates. *Biometrika* **28**(3/4), 321–377 (1936)
11. Hummel, M., Obermaier, H., Garth, C., Joy, K.: Comparative visual analysis of Lagrangian transport in CFD ensembles. *IEEE Trans. Vis. Comput. Graph.* **19**(12), 2743–2752 (2013)
12. Karhunen, J., Hao, T., Ylipaavalniemi, J.: Finding dependent and independent components from related data sets: A generalized canonical correlation analysis based method. *Neurocomputing* **113**, 153–167 (2013)
13. Kitware, I.: URL <http://www.paraview.org/> (2013)
14. Krzanowski, W.J.: Principles of Multivariate Analysis. A User's Perspective. Oxford University Press, London (1988)
15. Marzban, C.: Model tuning with canonical correlation analysis. *Mon Wea Rev (Conditionally Accepted)*. URL <http://faculty.washington.edu/marzban/cca.pdf> (2013)
16. Matkovic, K., Gracanin, D., Jelovic, M., Ammer, A., Lez, A., Hauser, H.: Interactive visual analysis of multiple simulation runs using the simulation model view: Understanding and tuning of an electronic unit injector. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1449–1457 (2010)
17. Mills, K., Filliben, J., Dabrowski, C.: An efficient sensitivity analysis method for large cloud simulations. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 724–731 (2011)
18. Moreland, K.: Diverging color maps for scientific visualization. In: *Advances in Visual Computing*, vol. 5876, pp. 92–103. Springer, Berlin (2009)
19. Piringer, H., Pajer, S., Berger, W., Teichmann, H.: Comparative visual analysis of 2d function ensembles. *Comput Graph. Forum* **31**(3 Pt 3), 1195–1204 (2012)
20. Potter, K., Wilson, A., Bremer, P.T., Williams, D., Doutriaux, C., Pascucci, V., Johnson, C.: Visualization of uncertainty and ensemble data: Exploration of climate modeling and weather forecast data with integrated visus-cdat systems. *J. Phys. Conf. Ser.* **180**(1), 012, 089 (2009)
21. Potter, K., Wilson, A., Bremer, P.T., Williams, D., Doutriaux, C., Pascucci, V., Johnson, C.: Ensemble-vis: A framework for the statistical visualization of ensemble data. In: IEEE International Conference on Data Mining Workshops, 2009 (ICDMW '09), pp. 233–240 (2009)

22. Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., Moorhead, R.: Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1421–1430 (2010)
23. Shearer, N., Khasawneh, M., Zhang, J., Bowling, S., Rabadi, G.: Sensitivity analysis of a large-scale system dynamics immigration model. In: *Systems and Information Engineering Design Symposium (SIEDS)*, 2010, pp. 78–81. IEEE (2010)
24. Smith, K.M., Banks, D.C., Druckman, N., Beason, K., Hussaini, M.Y.: Clustered ensemble averaging: A technique for visualizing qualitative features of stochastic simulations. *J. Comput. Theor. Nanosci.* **3**(5), 752–760 (2006)
25. Song, X., Zhao, G.: Sensitivity analysis for a forest growth model: A statistical and time-dependent point of view. In: *Plant Growth Modeling, Simulation, Visualization and Applications (PMA)*, 2012 IEEE 4th International Symposium (2012)
26. Steed, C.A., Ricciuto, D.M., Shipman, G., Smith, B., Thornton, P.E., Wang, D., Shi, X., Williams, D.N.: Big data visual analytics for exploratory earth system simulation analysis. *Comput. Geosci.* **61**, 71–82 (2013)
27. Sukharev, J., Wang, C., Ma, K.L., Wittenberg, A.: Correlation study of time-varying multivariate climate data sets. In: *Visualization Symposium, 2009 (PacificVis '09)*, IEEE Pacific, pp. 161–168 (2009)
28. Thornquist, H.K., Keiter, E.R., Rajamanickam, S.: Electrical modeling and simulation for stockpile stewardship. *XRDS* **19**(3), 18–22 (2013)
29. Waser, J., Fuchs, R., Ribicic, H., Schindler, B., Bloschl, G., Groller, M.: World lines. *IEEE Trans. Vis. Comput. Graph.* **16**(6), 1458–1467 (2010). DOI 10.1109/TVCG.2010.223
30. Waser, J., Ribicic, H., Fuchs, R., Hirsch, C., Schindler, B., Bloschl, G., Groller, M.: Nodes on ropes: A comprehensive data and control flow for steering ensemble simulations. *IEEE Trans. Vis. Comput. Graph.* **17**(12), 1872–1881 (2011). DOI 10.1109/TVCG.2011.225
31. Wilson, A.T., Potter, K.C.: Toward visual analysis of ensemble data sets. In: *Proceedings of the 2009 Workshop on Ultrascale Visualization (UltraVis '09)*, pp. 48–53. ACM, New York (2009)

Topological and Statistical Methods for Complex Data
Tackling Large-Scale, High-Dimensional, and
Multivariate Data Spaces

Bennett, J.C.; Vivodtzev, F.; Pascucci, V. (Eds.)

2015, XV, 297 p. 120 illus., 101 illus. in color.,

Hardcover

ISBN: 978-3-662-44899-1