“THEME ARTICLE”, “FEATURE ARTICLE”, or “COLUMN” goes here: The theme topic or column/department name goes after the colon.

A Query-based Framework for Searching, Sorting, and Exploring Data Ensembles

Garrett Aldrich[1, 3]

Jonas Lukasczyk[2]

Jeffrey D Hyman[3]

Gowri Srinivasan[3]

Hari Viswanathan[3]

Christoph Garth[2]

Heike Leitte[2]

James Ahrens[3]

Bernd Hamann[1]

[1]University of California, Davis

[2]Technische Universität Kaiserslautern

[3]Los Alamos National Laboratory

We introduce an innovative ensemble analysis framework for organizing, searching and comparing results produced by hundreds of physical simulations. Our web-based approach is built on standard technologies, utilizes a scalable and modular design, and is suitable for displaying the results of in-situ analysis and extreme-scale simulations.

To model real-world systems at a high-degree of accuracy, computer simulations must produce massive amounts of complex and multivariate data. When the parameters of these simulations are uncertain or non-deterministic behavior occurs, no single simulation result can be used to accurately predict the behavior of a system. However, massive increases in computational power have given scientists the ability to run these simulations repeatedly. The resulting dataset from each run is referred to as a realization, and the collection of these realizations is a data ensemble. Understanding the extent and variability in possible outcomes represented by an ensemble is of key importance for predicting the behavior of the modeled physical system. Data and visualization scientists apply a variety of analysis techniques that transform the simulation results into a format that can be understood by the intended audience. We have developed new tools that are broadly applicable to multiple scientific domains, with the goal of simplifying this complex process. Our effort’s key contribution is the development of the Database Optimized Relationship Analysis (DORA) framework.

DORA includes several tools for exploring, searching, and comparing realizations of a data ensemble. From a user perspective, DORA is primarily a way to browse, search, and interact with

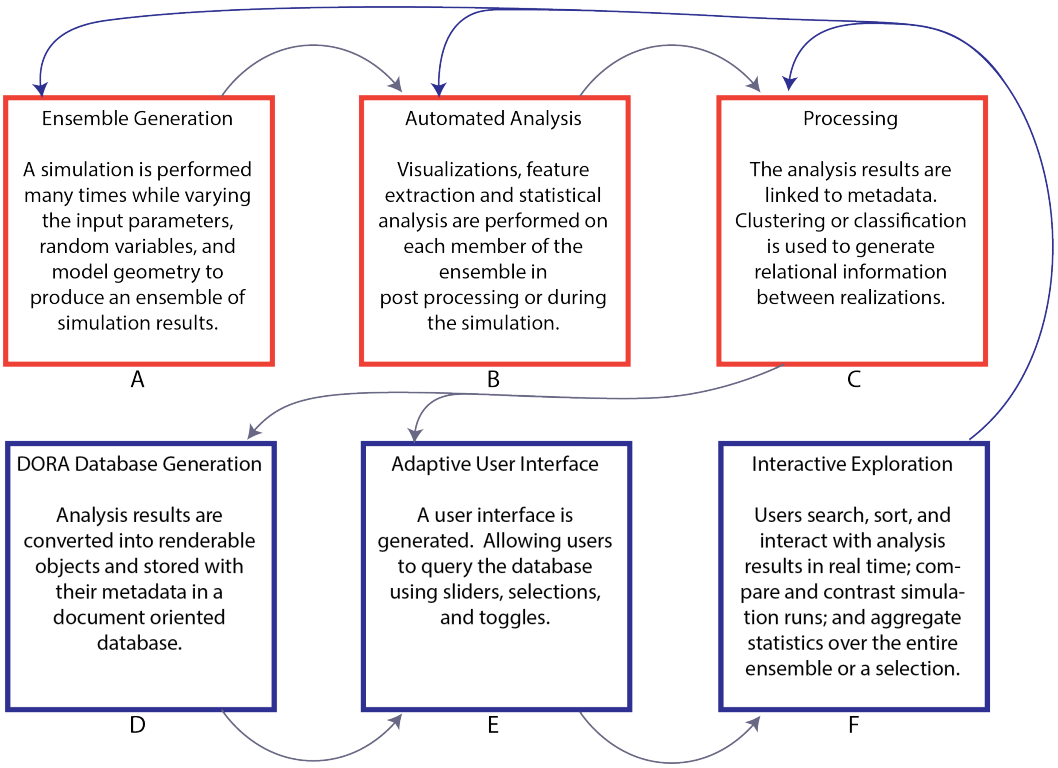

Figure 1 An example workflow for creating and analyzing complex scientific simulations that require multiple realizations for predicting real-world behavior. Ensemble generation and analysis (A-C) starts by running a simulation multiple times. The results are analyzed for each simulation to produce visualizations, graphs, and plots. These analysis results are processed, and relational analysis is performed to compare each simulation output. The DORA framework (D-F) then combines this analysis into a database and creates an intuitive user interfaces, which allows scientists and domain experts to explore the ensemble and find salient results. Finally, the insights gained from analyzing an ensemble with DORA may lead to new ideas that can influence how future ensembles are created, influence the type of analysis done for these ensembles, or determine the methods used to produce relational information between the realizations.

very large sets of analysis data produced by multiple physics-based simulations. While our main contribution is the presented framework, it is important to understand the typical workflow used to simulate a physical system. Figure 1 (A-C) provides an overview of this process. First, a domain expert generates an ensemble by executing a simulation multiple times, producing different results due to changing parameters, non-deterministic behavior, different model geometry, or a combination of these factors. Determining the number of necessary realizations is difficult, and it is often an iterative process, but several hundred simulations may be needed. The simulation results are often too complex to be easily understood in an unprocessed form. Automated analysis is performed, either via post-processing or while the simulation is running (*in situ* analysis), resulting in a set of "analysis products" for each realization. The algorithms required are domain-specific, and include visual, feature-based, and statistical methods. Each of these methods generally has its own parameters and may create many products, such as 2D and 3D renderings, statistical plots, or more abstract types like graphs and diagrams. These analysis products act as a proxy for the simulation output while requiring only a fraction of the original storage. To make them available to our framework, each analysis product is linked to metadata, including descriptions of the simulation and analysis methods, and the parameters for both. This includes the creation of relational information obtained by clustering or classification of the realizations. In this process, machine learning methods are applied to the metadata of each realization, grouping them into similar sets (groups or pre-defined classes); a numerical similarity or distance metric is produced for each realization in a set. The specific algorithms used varies for a given application domain, but many machine learning algorithms produce the needed type of information. While

we have developed specific tools for each of these three components, they are intentionally kept separate from the DORA framework described in this paper. This flexibility allows DORA to be integrated into existing applications where analysis methods are already well defined.

The DORA framework is described in Figure 1 (D-F). It combines several tools that allow users to access the analysis data and information produced from a large data ensemble, find specific realizations that meet user-defined criteria, and compare simulation results. To create an ensemble database, DORA first converts the analysis products into "render objects" that are compatible with the framework. DORA does not directly render large complex data types, like simulation meshes or 3D images, due to computational constraints. Instead, real-time interaction is emulated for these data types using a pre-rendering technique. Less complex data types, like graphs, plots and images, can be rendered directly by the framework. These render objects and their metadata are used to populate the analysis database that can be queried directly by a user. However, a user must know the query language and have experience with database operations. Alternatively, DORA generates a dynamic web-based interface using sliders, lists, and toggles as appropriate. This interface allows users to explore an ensemble by searching for simulation results that meet particular requirements to identify simulations that produced similar or dissimilar outputs, and aggregate statistical information from a current selection, or the entire ensemble. This query can be done with a database stored locally or accessed remotely on a server. DORA enables scientists to view and search through huge amounts of data in real-time to discover salient and unexpected behavior in an ensemble.

BACKGROUND

Scientist have used data ensembles as a way of predicting complex physical systems for decades. For example, meteorologists often generate hypotheses for a large storm's trajectory by analyzing multiple possible paths associated with various probabilities. To generate this set of paths, many simulations are performed and similar results are grouped together and averaged (clustered), creating a hierarchy. Each group is represented by a storm's possible path, usually visualized as a collection of paths overlaid on a map of the storm's geographic location. While these types of ensembles have been generated for years, the accuracy, complexity, and storage size of simulations has rapidly increased, requiring new methods for their analysis.[3]

Database creation is one of the preferred techniques for dealing with large ensembles. A recent example was demonstrated by Li et al.[1] who presented a framework for data-intensive climate ensembles. Their method optimizes a database to efficiently access the raw simulation grid for analysis and visualization. While their system is specialized for a particular domain and application, it demonstrates the effectiveness of utilizing database technology for accessing vast amounts of scientific data. Slycat[5] is a general framework that has been applied to multiple domains outside of climate data. It provides specific analysis methods focusing on finding correlations in parameter space. DORA is a more flexible framework that leverages a variety of analysis methods and is not targeted at a specific domain. This portability comes with a cost: Users cannot directly interact with the original simulation data; instead, they can only access the products of analysis methods applied to that data. However, these analysis products are typically orders of magnitude smaller[2] than the original data, making DORA extremely scalable regarding the number of simulations that can be analyzed at once.

Our motivation for using the paradigm of relying on analysis results directly is the increased prevalence of *in situ* analysis for extreme-scale simulations. *In situ* methods perform automated analysis during a simulation, essentially combining steps A and B from Figure 1, meaning that the raw simulation data does not have to be transferred from system memory to storage[4]. For extreme-scale simulations, bandwidth to storage devices can become the limiting factor and, in many cases, simulations calculate significantly more data than can be practically stored. These and other factors have led to the increased necessity of *in situ* analysis in modern scientific simulation. Data ensembles produced by these extreme-scale simulations continue to increase in size, motivating us to design our framework with *in situ* analysis in mind.

We predict that *in situ* methods will become the standard for dealing with large-scale ensembles. DORA can help tackle two of the main issues that arise from this type of analysis. First, all of the data analysis methods must be chosen ahead of time, as the simulation data is not stored for later analysis. It is therefore preferable to apply many different techniques and produce an abundance of analysis results. DORA is capable of organizing millions of analysis results and efficiently finding the analysis products that a scientist needs to answer specific hypotheses. Second, DORA makes it possible to visually interact with the simulation data in real time. We identified the Paraview Cinema[1] standard as an appropriate model for adding visualizations to our database. Cinema uses a set of images from a spherical camera model to effectively emulate 3D visualizations. Rendering can take place *in situ*, or as a post processing step. DORA enables users to rotate, zoom, and pan 3D visualizations as well as move forward and backward in time as if the data was being rendered in real time. The types of visualizations and the features being rendered must be chosen ahead of time; however, large-scale datasets can be remotely explored at interactive rates using this technique.

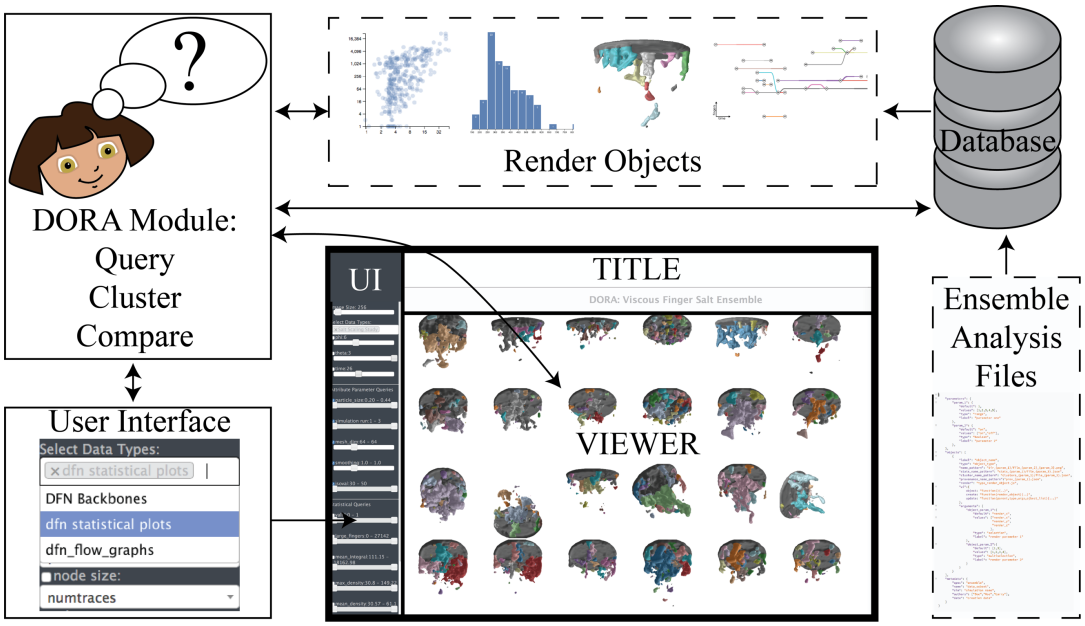

Figure 2. Diagram of the DORA framework. A database is generated from the analysis files produced from in situ or post processed analysis of simulations. The visual components of these objects are stored as render objects that can be displayed in the Web-based viewer. The Query Cluster Compare (QCC) module uses the database information to generate a user interface. Selections made by users are interpreted as queries on the database by the QCC. Results are displayed in the viewer.

ENSEMBLE EXPLORATION AND ANALYSIS FRAMEWORK

Figure 2 illustrates the DORA framework and how the database is created from analysis objects for exploring an ensemble of simulation results. Each solid block represents a module of the framework and arrows indicate communication between modules. To analyze a data ensemble, the set of analysis objects, indicated by dashed lines, is sent to the Database module. We use a JavaScript implementation of the Apache CouchDB® standard, PouchDB®, which enables both

local and remote database access. The Database module creates an entry for each analysis object with associated parameters, statistics, and clustering information. Every entry is sent to the Render Objects module, where the analysis objects are converted into extendable JavaScript based render objects for displaying and interacting with the analysis results. The Database module compiles the extents for statistics, parameters, and relational information, which are sent to the Query Cluster Compare (QCC) module. The QCC passes this information to the User Interface (UI) module, which dynamically creates the interface on the viewer. Interactions made by the user are sent from the UI back to the QCC that translates these interactions into database queries and displays the results on the viewer.

The ability to query analysis objects using a dynamically created user interfaces is what makes DORA a powerful tool for exploring large data ensembles. Each object in the database has information about the simulation that created it including: simulation parameters used in that simulation, both global and feature-specific statistical information, relational information, feature types, and analysis parameters. The combination of this metadata makes it possible to quickly search, organize and compare the results generated by different simulations or analysis processes using an intuitive interface. Users search the Database by choosing keywords and selecting either a specific or range of numerical values from sliders, strings from list of options, or a toggle to indicate that a specific keyword is linked to all analysis objects returned. These selections are converted into queries on the database, which returns the set of render objects which meet the selected criteria. Users can interact with each render object directly or as a linked group. They can also choose to explore all related objects which belong to the same group, and search within a current selection.

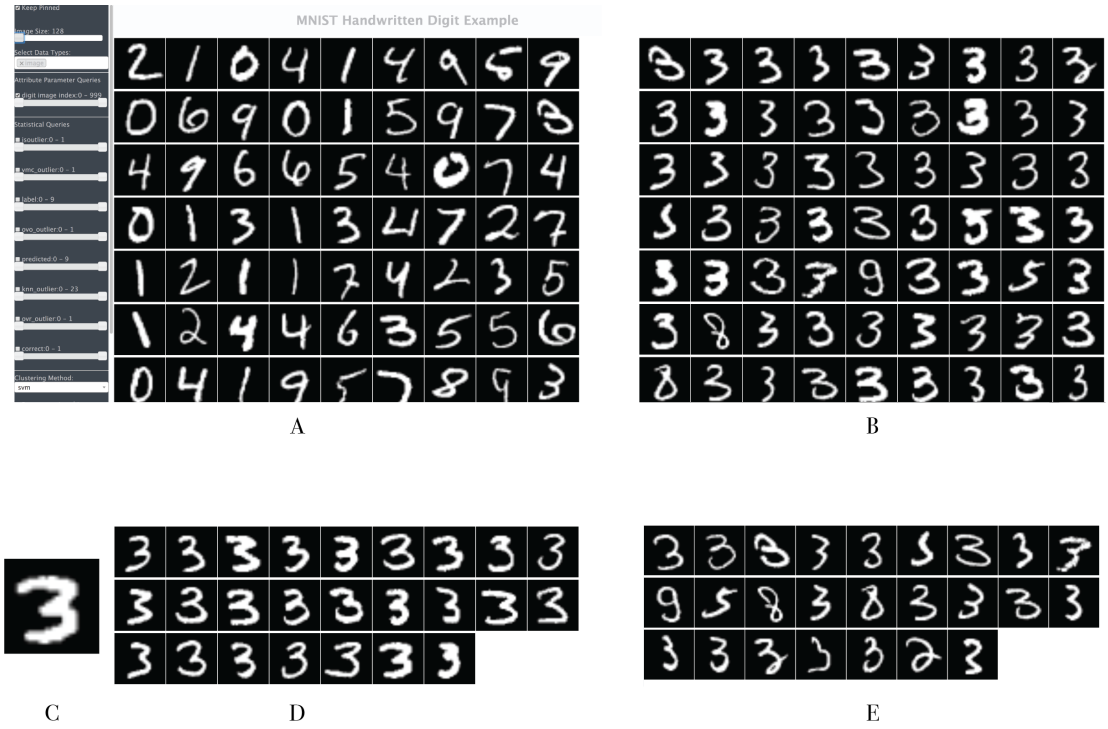

Figure 3. Example of cluster-based exploration of the MNIST handwritten digit database. A) The database can be viewed with the standard web interface. B) The user selects “3” and images belonging to the same cluster are shown. C) The most representative image, i.e., the one closest to the center of the cluster, is displayed. D) The 25 images most similar to (C) can be selected or E) the 25 least similar to (C) can be generatd. These can be interperated as outliers for that cluster.

We describe how users interact with DORA by discussing an example dataset consisting of images from a commonly used machine learning training set. The MNIST[6] handwritten digits are

images of the digits zero through nine, as shown in Figure 3(A) within the DORA interface. Each image in the database contains the correct label and machine learning id. To compute relational information for each image, we use a standard implementation of a support vector machine (SVM).[7] SVM is a method that compares images that are unknown to pre-labeled groups or classes, and predicts which class they should belong to. This approach accurately groups an image of a handwritten digits with the correct labels (i.e. a handwritten "8" will be classified as an 8) 97% of the time. To create an ensemble database, each image is added as an analysis object. The metadata for each image includes the correct label, a predicted class from the SVM, and different distance metrics. The predicted classification information is used by DORA as relational information, so that users can find related images in the database.

To explore the ensemble of handwritten digits, a user manipulates sliders to select all images classified by the SVM as a "3," see Figure 3(B). By selecting any one of these images, the user can perform a query for the representative image in that group; in this case, the image closest to the respective support vector, shown in Figure 3(C). Being able to directly compare to a representative object, allows the user to better understand the variability that can occur within a group. In a scientific setting, this type of operation is useful for discovering outliers, and comparing them to an average or typical case. The system allows the user to easily query for the objects most similar to the representative object. In this case the 25 most similar images are shown in Figure 3(D). Alternately, the user can select the least similar images to a representative object (E) in the cluster, which is a useful tool for discovering outliers. In this case, several poorly written digits are returned as well as several mislabeled images. In a more general setting, a scientist could further explore and investigate related analysis objects to improve their algorithm and gain better understanding of why outliers occur.

Filtering the analysis objects generated and extracted from a data ensemble allows the user to select simulation results meeting specific criteria. Our system generates potential selection criteria by accumulating statistical and parameter information for a current DB selection, and it represents the accumulated information as plots created dynamically. Users can create histograms for a single variable, scatter plots for two variables, or line plots for two variables where one is a function of the other. This functionality supports the process of identifying outliers, selecting query bounds, or identifying trends among previously selected data objects. Direct comparisons can also be made for groups of data objects. For example, plots of the same type, representing the same variables, can be combined and overlaid. Users can highlight lines, points or bars to emphasize specific details. Cinema-style image data can be linked so that views and temporal values are associated. For more advanced comparisons, our framework can be extended by including JavaScript plugins, making it possible to compare more abstract data/data representations such as graphs.

ENSEMBLE OF SIMULATED FLOW IN FRACTURED ROCK AND EXTRACTION OF NETWORK BACKBONES

In low permeability rocks, e.g., granite and shale, fractures are the principal pathway for flow and the associated transport of dissolved chemicals. Characterizing fluid flow in these formations is important to many civilian, government, and industry applications including: aquifer management, hydrocarbon extraction, and long-term storage of spent nuclear fuel.[11] One tool that geoscientists use to model these phenomena are discrete fracture network (DFN), where intersecting polygons represent the cracks in subsurface rock in which fluid and gas are transported. Due to inherent uncertainty associated with the subsurface, ensembles of DFN models are stochastically generated with the same statistical properties, but different network geometry.

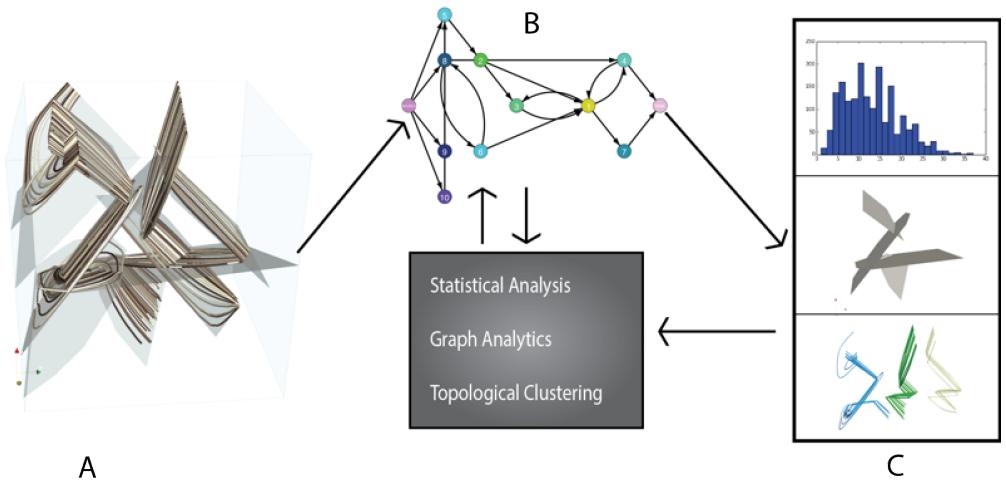

Figure 4 To automate flow and transport analysis for discrete fracture networks (A), we use a graph representation (B) from which we extract statistical properties, paths of concentrated flow and clustered particle traces (C). This method allows us to automate analysis of large numbers of flow simulations and store them in a database for use with DORA. (Image courtesy of Aldrich et al.[12])

To demonstrate the utility of DORA in these simulations, we examine particle transport behavior in an ensemble of 100 flow simulations within independent identically distributed DFNs. Coalescence of particle path lines indicates the occurrence of flow channeling within the network and the existence of backbones, a subnetwork where most of the flow occurs. A key question is how do these channelized paths influence transport behavior, and, in particular, how long it takes for a particle to exit the domain (breakthrough time). We use an automated graph-based method for flow network analysis to extract these backbones. Figure 4 depicts the workflow for the analysis system that is described in detail in Aldrich et al.[12] A key component of the analysis is generating a flow topology graph (FTG), where each fracture is represented by a vertex in the graph, and edges represent flow from one fracture to another. Analysis of the FTG produces the following: a hierarchy of channelized paths, ordered by the amount of flow over each set of fractures; several statistical properties about the rate of flow, breakthrough times, etc; and particle clustering information.

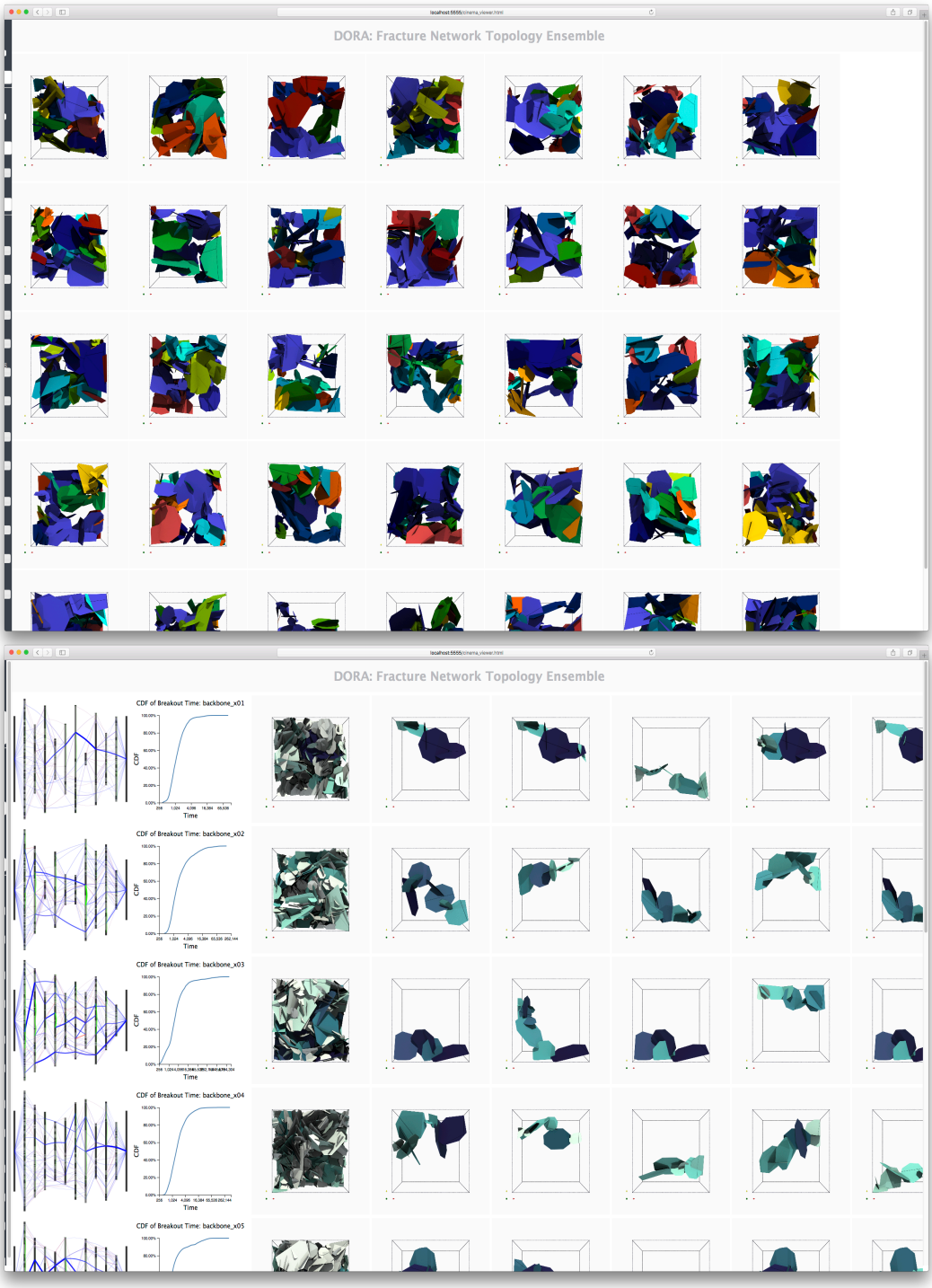

Figure 5. Ensemble of 100 discrete fracture network (DFN) simulations, each one presented with several analysis results, viewed with the viewer for DORA. Top: A gallery-style view allows the user to quickly browse through the entire ensemble, here the set of fractures forming channelized paths, shown for each network. Each path is rendered in a unique color. Bottom: For more detail, users can select a grid view presenting analysis results for each simulation, enabling direct comparison of selected simulations. A user can access over 4000 database entries, containing over 500,000 images.

For the DFN ensemble, automated analysis is applied to each of the 100 members and used to generate the ensemble database. Figure 5 shows two examples of how this database can be viewed using the DORA framework. In the top view, each simulation result is represented by a single analysis object and renderings of the channelized path hierarchy, where each path is independently colored. This relatively small subset of the total network carries the majority of flow and is generally more useful than viewing the entire network. The bottom view in Figure 5 uses a grid layout, which is better for comparing the results between multiple simulations. In this case the user has chosen to display a dynamic rendering of the flow topology graph for each simulation, a statistical plot, a rendering of the entire network, and renderings of each individual channelized path. The user can compare several simulation runs at once, however it is difficult to see details about the ensemble as a whole.

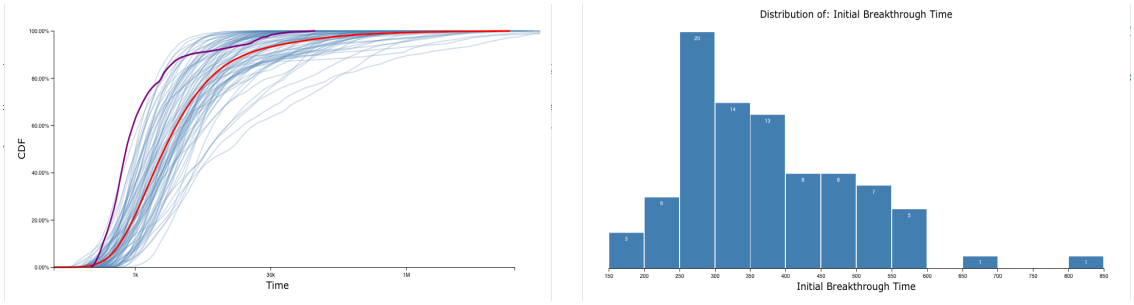

Figure 6 Breakthrough time is the amount of time required by each simulated particle to traverse the flow network and exit the system. The CDF of breakthrough times (breakthrough curve) can be accumulated into a single plot (left) where the average is shown as a red line. The user can perform queries to find the simulation results that produced a specific curve by selecting it in the plot (purple line). Users can display statistical values, e.g. distributions of intial breakthrough, the minimum time required by a particle to traverse the system (right). The ability to automatically generate these plots from a current selection allows users to understand the variability of possible outcomes and identify outliers for further investigation.

DORA can also be used to compare results across an ensemble and identify outliers by accumulating plots and statistical values from the database. Figure 6 shows how these results can be overlaid to give an indication of the variability present in an ensemble and reveal trends in the data. In Figure 6(A) the cumulative breakthrough times are shown for all 100 simulations. An average line is calculated (red), and users can select individual curves (purple). The selected curve(s) can be used to query the database for those simulations. Figure 6(B) shows a histogram first breakthrough times, which appears to have some outliers at the upper end.

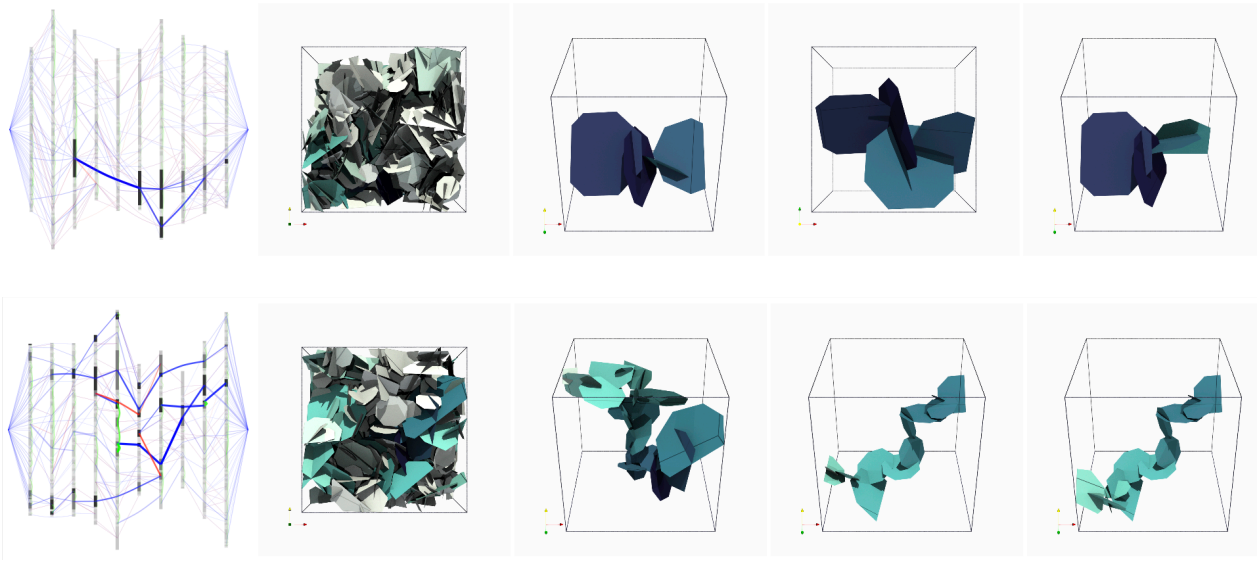

Figure 7 Fracture networks with fastest (top row) and slowest (bottom row) initial breakthrough times can be selected from the 100 network ensemble. The generated visualizations allow us to evaluate the top three backbone paths for each network, i.e., the paths carrying the most flow in the network. The paths are shown in the flow topology graph for each simulation and in the geometric representation of each path. Feature-based analysis is necessary as detailed structures can be occluded in the network when rendering a DFN representation of each fracture network.

DORA can also be used to investigate the disparity between the two extremes of initial breakthrough. Fig. 7 (top) shows backbone information in a network with one of the fastest breakthroughs and Fig. 7 (bottom) shows the same information for a network with very slow travel times. In the fast network, the backbone consists of larger fractures oriented in primary direction of flow (from left to right). In the slow network, the backbone is composed of relatively small fractures with variable orientations, resulting in slower breakout times. The differences between these two networks highlight how the random topological configuration of fracture networks, even when using the same underlying statistical properties, can drastically influence simulation results.

Flow in fractured rock is one of the many examples where inherent uncertainty requires and ensemble of models to capture the full range of possible results. Analyzing DFN simulations is computationally expensive, and interactively visualizing the results can be difficult due to their complexity and size. DORA enables scientists to efficiently discover outlier behavior within an ensemble and interactively explore and compare results in real time for up to tens of thousands of fractures.

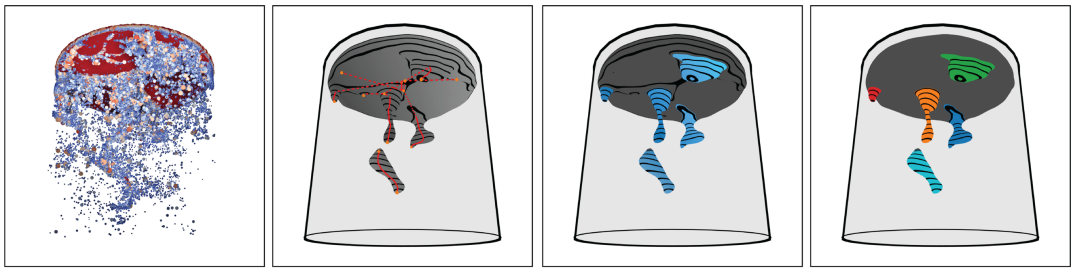

Figure 8. A finite particle simulation of salt water being injected into a can of water. (A) Blue indicates low consentration of salt, and red indicates high concentration. We estimate the density function of salt as a function of volume and compute a graph of the highly concentrated areas (B) to extract the viscous finger structures from the salt base (C) and separate them into components (D). Each component is assigned a random color that remains the same over time, so that the user can see these fingers evolve. These structures are important to scientists for understanding the complex mixing behavior of fluids arising in numerous engineering (drug delivery) and natural systems (carbon sequestration and hydrocarbon extraction). (Image courtesy of Lukasczyk et al.[10])

ENSEMBLE STUDY OF VISCOUS FINGER EXTRACTION IN MIXING OF FLUIDS

DORA's main purpose is a tool for domain experts to aid in evaluating their scientific work, however it is also a beneficial tool for data and visualization scientists in rapid algorithm development. Recently DORA was used to develop feature extraction and analysis algorithms for the 2016 IEEE Science Visualization Contest. The goal of this contest was to analyze an ensemble of 50 finite particle method (FPM) simulations in which salt water solution is suspended over a water solution, as shown in Figure 8(A). As the two fluids mix, the dense salt water forms finger like clumps that penetrate the boundary between the two liquids.[8] Understanding this phenomenon is critical for multiple applications including determining how much oil (high viscosity) can be recovered from a reservoir partially saturated with water (low viscosity)[9].

DORA allowed us to test multiple algorithms, apply them to the entire simulation ensemble, and evaluate the results in a meaningful way. The feature extraction method, which earned 2nd place in the contest, is described in Figure 8(B-D). First, a concentration function is estimated over the domain's volume (can of water) and a graph is constructed of the regions with greatest density. This graph allows us to extract the viscous fingers as structures representing a subset of the volume where salt is concentrated beyond a given threshold. These regions are assigned a color and then tracked over time so that scientists can see how they nucleate, evolve, and disperse. During the analysis, several statistical values are also calculated and stored, for example, the total salt concentration, volume, depth and center of mass for each viscous finger. For a detailed description of the algorithm see Lukasczyk et al.[10]

This algorithm is effective for identifying viscous fingering in the volume, extracting the structure of these fingers, and tracking their evolution over time. However, several analysis parameters needed to be adjusted to produce the desired results. For example, there are several threshold values that can be used to separate the concentrated salt regions that represent viscous fingers from the rest of the liquid (Figure 8B). The analysis algorithm is also scale-dependent, and different parameters produce differently sized features. Figure 9(A) shows how different parameter values can produce remarkably different results from the same realization. Furthermore, different simulation parameters used to generate the ensemble required different analysis parameters to properly extract features at a reasonable scale. A final complication is the fact that the algorithms we developed were both time- and space-intensive: 15-20 minutes was the required run time, and 20GB of storage space was required per analysis run.

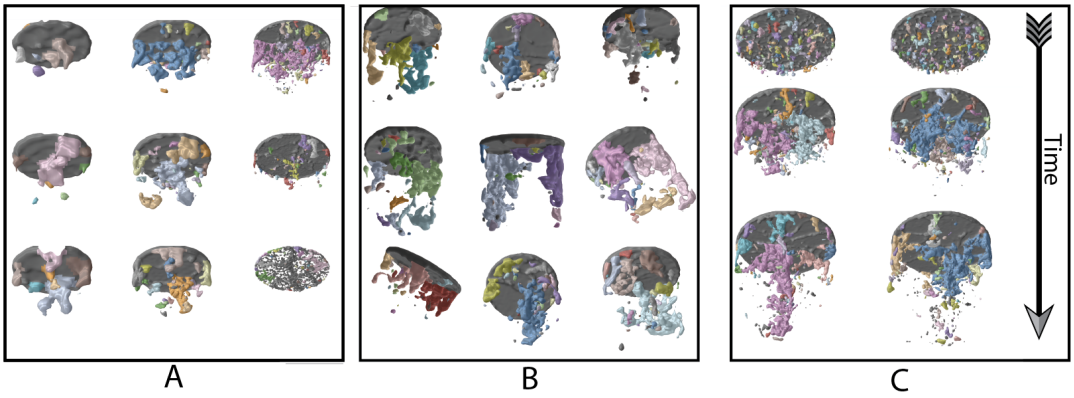

Figure 9 We use DORA to rapidly develop a novel analysis method for extracting viscous fingers from a finite point method fluid simulation. DORA enabled us to evaluate different analysis parameters and their effectiveness for extracting the desired structures (A). Each structure is rendered in a unique color remaining the same during the simulation. Once proper analysis parameters are identified, the user can display every member of the ensemble with these parameters (B). Multiple ensemble members can be linked so that they share the same view and time settings (C), supporting effective visual comparison between simulations.

The tools of the DORA framework allowed us to overcome these issues. First, we automated the feature extraction algorithm so that it could be applied multiple times with different parameters. We applied 64 iterations of the analysis on each of the 50 ensemble realizations, which resulted in 320 analysis products, each of which includes 3D time-dependent visualizations and the statistical information for each. Figure 9 gives an example of these analysis products and how they are evaluated in the DORA framework. Figure 9(A) shows that some of the results produced structures that were not useful for scientific evaluation, as they were invalid. However, examination of the statistical properties in cases where the analysis did produce good results allowed us to search the database using these properties. Queries based on these properties returned a set of analysis products that properly represented the viscous finger structures, figure 9(B). Using DORA, we evaluated these results individually and compared multiple simulations at the same time, both spatially and temporally, cf. figure 9(C).

This example highlights how DORA provides can be used to evaluate analysis techniques for ensemble datasets. Multiple methods and analysis parameters can be evaluated for each member of the ensemble, which is especially useful in cases where different groups are analyzing the same simulation results, or where multiple hypotheses need to be tested and each may need a different type of analysis. DORA users can visually evaluate the results and query the ensemble metadata to select simulation results with particular properties. While analysis methods may be computationally expensive, only the rendered images and statistical properties are stored long term. Once the database has been created, users are able to visualize, query, and interact with the results in real time, both locally and remotely. DORA enables domain and data scientists to rapidly develop, test, and compare multiple analysis methods across large ensembles.

CONCLUSION

For large ensembles produced from multiple simulation runs, automated analysis is often necessary to produce images, extract features, transform the data into abstract types, and/or produce statistical descriptions of the data. One issue is that the best features for a specific domain, the number of images needed, or the type of statistics that could be useful are unknown when

implementing the automated analysis. Our solution is to run many different analysis methods, either *in situ* or in post processing, from which the most useful results are then included in a database for ensemble analysis. For a large ensemble, this can easily produce more results than is possible to effectively evaluate.

We developed the DORA framework to allow users to search, sort and compare these analysis results and probe specific scientific hypothesis about the system being modeled. Increases in computational ability and simulation complexity have drastically increased the number of data ensembles and the need for analysis tools. We leveraged recent advances in web-technologies to address this need in the development of DORA. The ensemble database stores the simulation parameters, analysis parameters, metrics, derived statistics, and clustering information alongside each analysis objects. This design allows the user to search for specific simulation results, features, or metrics by querying the database. We designed an interface that lets the user build queries in a natural way by combining the selections of particular values, ranges of values, or the existence of keywords in the database. DORA was designed from the bottom up, to be a collaborative tool that runs in a web-based environment and can be deployed both locally or using a server-client model.

We demonstrated the utility of DORA using two case studies from vastly different domains. In each case study, we highlighted an important capability of our system including the ability to evaluate automated analysis and identify analysis parameters which provide the most insight into a particular dataset, using clustering information to identify outliers and compare analysis results, and using statistical information to search for simulation results with features that exhibit particular properties of interest. The combination of these analysis methods makes DORA a powerful tool for providing insight in the complex behavior of data-intensive simulations.

REFERENCES

1. Li, Z., Huang, Q., Carbone, G. J., & Hu, F. (2017). A high performance query analytical framework for supporting data-intensive climate studies. Computers, Environment and Urban Systems, 62, 210–221.
2. Ahrens, J., Jourdain, S., O'Leary, P., & Patchett, J. (2014). An image-based approach to extreme scale in situ visualization and analysis (pp. 424–434). An image-based approach to extreme scale in situ visualization and analysis.
3. Wilson, A. T., & Potter, K. C. (2009). Toward visual analysis of ensemble data sets (pp. 48–53). Presented at the the 2009 Workshop, New York, New York, USA: ACM Press.
4. Ayachit, U., Bauer, A., Geveci, B., OLeary, P., Moreland, K., Fabian, N., & Mauldin, J. (2015). ParaView Catalyst (pp. 25–29). Presented at the the First Workshop, New York, New York, USA: ACM Press.
5. Crossno, P. J., Shead, T. M., Sielicki, M. A., & Hunt, W. L. (2015). Slycat ensemble analysis of electrical circuit simulations. Topological and Statistical Methods for Complex Data. Tackling Large-Scale, High-Dimensional, and Multivariate Data Spaces, Springer, 279-294.
6. Li Deng. (2012). The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. IEEE Signal Processing Magazine, 29(6), 141–142.
7. Duan, K., & Keerthi, S. S. (2005). Which Is the Best Multiclass SVM Method. An Empirical Study. Multiple Classifier Systems.
8. De Wit, A., Bertho, Y., & Martin, M. (2005). Viscous fingering of miscible slices. Physics of Fluids.
9. Jiménez Martínez, J., Porter, M. L., Hyman, J. D., Carey, J. W., & Viswanathan, H. S. (2016). Mixing in a three-phase system: Enhanced production of oil-wet reservoirs by CO2 injection. Geophysical Research Letters, 43(1), 196–205.
10. Jonas Lukasczyk, Garrett Aldrich, Michael Steptoe, Guillaume Favelier, Charles Gueunet, Julien Tierny, Ross Maciejewski, Bernd Hamann, and Heike Leitte. (2017). Viscous Fingering: A Topological Visual Analytic Approach. (To Appear) Applied Mechanics and Materials 869 - Proceedings of the 1st Conference on Physical Modeling for Virtual Manufacturing Systems and Processes (2017): S. 9-19

11. Hyman, J. D., Jiménez-Martínez, J., Viswanathan, H. S., Carey, J. W., Porter, M. L., Rougier, E., et al. (2016). Understanding hydraulic fracturing: a multi-scale problem. Philosophical Transactions of the Royal Society a: Mathematical, Physical and Engineering Sciences, 374(2078), 20150426–16.
12. Aldrich, G., Hyman, J. D., Karra, S., Gable, C. W., Makedonska, N., Viswanathan, H., et al. (2017). Analysis and Visualization of Discrete Fracture Networks Using a Flow Topology Graph. IEEE Transactions on Visualization and Computer Graphics, 23(8), 1896–1909.