# Visualization System Requirements for Data Processing Pipeline Design and Optimization

Tatiana von Landesberger, Dieter W. Fellner, and Roy A. Ruddle

**Abstract**—The rising quantity and complexity of data creates a need to design and optimize data processing pipelines—the set of data processing steps, parameters and algorithms that perform operations on the data. Visualization can support this process but, although there are many examples of systems for visual parameter analysis, there remains a need to systematically assess users' requirements and match those requirements to exemplar visualization methods. This article presents a new characterization of the requirements for pipeline design and optimization. This characterization is based on both a review of the literature and first-hand assessment of eight application case studies. We also match these requirements with exemplar functionality provided by existing visualization tools. Thus, we provide end-users and visualization developers with a way of identifying functionality that addresses data processing problems in an application. We also identify seven future challenges for visualization research that are not met by the capabilities of today's systems.

**Index Terms**—Visualization systems, requirement analysis, data processing pipelines

## 1 Introduction

Visualization has historically been used to derive new findings from data, and to communicate those findings to a wider audience. Today, the rising quantity and complexity of the data give rise to an important third usage: to design and optimize data processing pipelines, especially those where scientists are faced with a large space of pipeline and/or parameter choices. This is the case in diverse domains such as medical imaging and business intelligence, chemistry and security.

A *pipeline* is a sequence of *computations*, i.e., *steps*. Each computation is implemented with certain algorithms and executed on input data using specific algorithm parameters. The computation produces outputs (results) from input data (see Fig. 1a). When a pipeline has several steps, the initial inputs are used to compute intermediate outputs, and the intermediate outputs are used as inputs to the next step of the pipeline. This is repeated until the final output is produced (see Fig. 1b).

To *design* a pipeline users choose between different computation steps, or their execution order. During *optimization* users often keep the computation steps fixed but choose different algorithms or their parameter settings. Note, we use the term *workflow* to encompass the whole analytical process, from data acquisition, through application of the data processing pipeline, to investigation and explanation of the results.

Both the design and optimization of pipelines can be performed in various ways, from fully programmatic (e.g., programs written in R, Matlab, or other language), via a combination of programmatic and visualization, to a fully visual way (e.g., using KNIME, VTK). This paper focuses on the possibilities and opportunities of visualization support in designing and optimizing pipelines. Visualization has the potential to help scientists understand the trade-offs between different models, processing methods (e.g., one algorithm versus another), the consequences of choices or assumptions made during one step in a pipeline on later steps (failing to do this leads to a phenomenon termed *broken workflow* [1]), and assess outputs against objective and subjective criteria. The net result will be pipelines that are more effective for both automated and human-in-the-loop processing.

A systematic assessment is needed to show how current visualization functionality matches user's requirements for pipeline design and optimization, and which new functionality still needs to be developed. Previous research has made steps in this direction (e.g., Sedlmair et al.'s conceptual framework [2]), but the focus was on parameter space analysis rather than pipeline design and optimization.

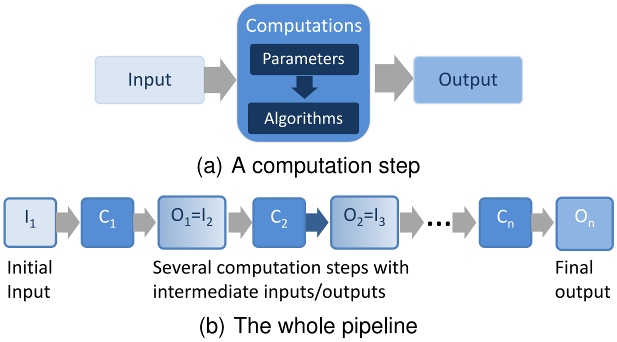
The present article addresses this need by making four important contributions. First, through eight case studies we describe a breadth of application challenges for the design and optimization of data processing pipelines (see Section 4). Second, by combining those case studies with a thorough review of literature, we characterize users' requirements (see Section 5). Third, we match users' requirements to the functionality that is provided in exemplar visualization systems (see Section 6). This can help users to define their problems and find appropriate visualization systems, and developers to profit from proven functionality and solve larger problems by combining that functionality. Fourth, we identify challenges for future visualization research (see Section 7.2).

## 2 Related Work

This section is divided into two parts. First, we review the wide variety of methods that are used to design and

- *T. von Landesberger and D. W. Fellner are with TU Darmstadt, Darmstadt 64289, Germany.*
*E-mail: {tatiana.landesberger, dieter.fellner}@gris.tu-darmstadt.de.*
- *R. A. Ruddle is with the University of Leeds, Leeds LS2 9JT, United Kingdom. E-mail: R.A.Ruddle@leeds.ac.uk.*

*Manuscript received 18 Oct. 2015; revised 7 July 2016; accepted 27 July 2016. Date of publication 25 Aug. 2016; date of current version 30 June 2017.*
*Recommended for acceptance by H. Qu.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TVCG.2016.2603178*



(a) A computation step

(b) The whole pipeline

Fig. 1. Data processing pipeline schema.

optimize pipelines. Then we briefly summarize existing frameworks and characterizations that underpin our research. Exemplars of visualization systems that have been developed to support pipeline optimization and design are presented in Section 6.

### 2.1 Methods for Pipeline Design and Optimization

Current methods for pipeline design and optimization range from fully automatic to methods that require substantial user input. Fully automatic methods are best suited to problems that are well-understood, and have established analytical pipelines and pre-defined outcomes that can be automatically evaluated. Examples range from single-criterion optimization problems [3], [4] to problems with multiple criteria that adopt techniques such as multi-criteria decision analysis [5].

The user-based methods may be subdivided according to the manner in which users provide input: iteration, parameter sampling, and whole workflow. In the first of these, users iteratively choose pipeline parameters and examine the outputs, until the solution is satisfactory. Some techniques are primarily computational, often incorporating data mining methods, and examples include dimension reduction [6], clustering [7] and classification [8]. However, visualization is an inherent part of others approaches, for example, using statistical learning techniques to make a real-time prediction of the results for regions of parameter settings [9].

Parameter sampling systems help users to investigate the outputs for many combinations of parameter settings. The ability to visualize those outputs brings several important benefits, which include allowing users to save considerable time [10], [11], conduct a far more rigorous review of outputs than is possible with the iteration approach [12], and gain a high-level understanding of how different parameters interact to affect the outputs [13], [14].

Whole-workflow systems capture the provenance of data analysis, and provide visual support for multiple lines of inquiry that is particularly beneficial when analysis takes place over an extended period of time or involves multiple end-users [15], [16]. Visualization brings benefits that include significantly improving the process of prototyping engineering designs [17] and comparing flood-control strategies [18].

The pipelines that are used with the above methods involve one or more computation steps (see Fig. 1). Most applications involve the running of a simulation model or data processing algorithms, which may be bespoke or utilize existing packages. In some applications visualization tools are just used to investigate output from computation steps, with examples being Vismon for multiple Monte Carlo runs of a fisheries model [19] and Orchestral for copy number calculations in genomics [1]). In other applications a user both runs computations and visualizes output with a single tool, with examples being World Lines [18] and the visual steering during design prototyping [17].

### 2.2 Frameworks and Characterizations

Our research builds on Sedlmair et al.'s conceptual framework [2], which characterizes data flows, navigation strategies and analysis tasks that take place during visual parameter space analysis. Sedlmair et al.'s analysis shows that most users adopt a global-to-local navigation strategy that is consistent with Shneiderman's well-known mantra [20]. At a global level, users are concerned with acquiring a 'big picture' of the data by understanding trends, outliers, clusters, distributions and correlations. At a local level users wish to understand the details of particular parameter choices and outputs.

Sedlmair et al. [2] identify six user tasks: optimization, model output partitioning, model fitting, finding outliers, assessment of output uncertainty, and model sensitivity. Outliers are one aspect of the veracity of a pipeline's inputs, and uncertainty and sensitivity are included as distinct factors in our characterization of users' requirements (see Section 5). However, optimization and model fitting concern the overall purpose of conducting visual parameter analysis and partitioning involves understanding outputs in the context of parameter settings. These tasks complement our characterization.

Our characterization of user's requirements is concerned with the factors that influence the construction and optimization of data processing pipelines. Our paper takes a different perspective on visual parameter analysis than that theoretically summarized by Sedlmair et al. [2] or analyzed in previous scenarios in the literature (see Table 1). Sedlmair et al. focus on the tasks performed during data modeling and analyze types of parameters, inputs and outputs of modeling. They put focus also on user's navigation strategies. We focus more on the burdens and requirements encountered in design and optimization, for example, users' comprehension of pipeline steps or output assessment time (see Section 5). Nevertheless, we note that the basic calculation step, the computation of derived data, and the analysis of output sensitivity feature in both works. Therefore, we see our work as complementary extension of the framework provided by Sedlmair et al. [2].

A number of other studies have characterized the tasks that users perform with visualization systems. Of particular relevance to our research is a study that interviewed 35 data analysts from 25 organizations to investigate the challenges and barriers that analysts face [21]. Common issues were provenance, the validity and consistency of assumptions, and the sensitivity of findings to choices made during analysis (e.g., parameter settings). All of these feature in our requirements characterization. Other visualization task characterizations are more abstract (e.g., [22]) and cover the full scope of usage of visualization systems, from pipeline design/optimization to deriving/communicating new findings.

TABLE 1
A Mapping of Users' Requirements Onto the Eight Application Case Studies (See Section 4) and the 28 Literature Review Papers ('*' Indicates the Seven Papers that were not Part of Sedlmair et al.'s Review [2])

| Requirement/ scenario | Software name | Inputs | | | | Computations | | | | | | Outputs | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Volume | Velocity | Variety | Veracity | Unknown pipeline | Assumptions | Number of combinations | Comprehension | Compute time | Provenance | Number of outputs | Subjectivity | Sensitivity | Uncertainty | Assessment time | Context |
| ***First-hand Application Case Studies*** | | | | | | | | | | | | | | | | | |
| Comparative genomics [1] | | | | | x | x | | ■ | | x | | x | | ■ | | ■ | ■ |
| 3D image segmentation [24] | | | | | | ■ | x | ■ | x | | | ■ | ■ | x | | | x |
| Histopathology segmentation [25] | | | | | | ■ | | ■ | | | | | ■ | ■ | | ■ | |
| Chemical process models [26] | | | | | | x | | ■ | ■ | | | | | | x | | |
| Economic modelling [27] | | | | | | ■ | ■ | x | | x | | | ■ | | ■ | x | ■ |
| Aircraft engine design [28] | | | | | | | | ■ | ■ | x | | | ■ | ■ | | | |
| Phylogenetic trees [29,30] | | | | | | | x | ■ | | x | | x | | ■ | | ■ | |
| Molecular evolution [31,32] | | x | | | | | | | | x | | ■ | | x | ■ | x | |
| ***Literature Review*** | | | | | | | | | | | | | | | | | |
| * Raidou et al. [75] | iCoCooN | | | | | | ■ | ■ | | | | ■ | | | | | |
| * Ruppert et al. [46] | -- | | | | | | | x | ■ | | | x | | | | | |
| Luboshik et al. [38] | -- | | | | | | | x | | | | ■ | | ■ | ■ | | |
| Bruckner et al. [10] | -- | | | | | | | x | | | | ■ | x | ■ | | | |
| * Beham et al. [43] | Cupid | | | | | | | ■ | | | | ■ | x | x | | | |
| Konyha et al. [39] | -- | | | | | | | x | | | | ■ | x | | | | |
| Pretorius et al. [12] | Paramorama | | | | | | | ■ | | | | x | ■ | | | | |
| Afzal et al. [36] | RVF | | | | | | x | ■ | | | ■ | | x | | | | |
| Bergner et al. [13] | ParaGlide | | | | | | | x | | x | | | ■ | | | | |
| Torsney-Weir et al. [33] | Tuner | | | x | | | | ■ | | x | | | x | | | | |
| *Padua et al. [11] | -- | | | | | | | x | ■ | | | | x | | | | |
| * Bögl et al. [42] | TiMoVA | | | | | ■ | | ■ | ■ | | x | | x | ■ | | | |
| Spence et al. [73] | -- | | | | | | | ■ | | | | | | ■ | | | |
| Berger et al. [9] | -- | | | | | | | ■ | | x | | | | x | x | | |
| Piringer et al. [40] | HyperMoVal | | | | | x | | x | | | | | | ■ | ■ | | ■ |
| Matkovic et al. [17] | -- | | | | | ■ | | x | | | | | | | | | |
| Coffey et al. [34] | -- | | | x | | | | x | ■ | | | | | | | | |
| Matkovic et al. [50] | -- | | | | | | | | | | | x | | | | | |
| Potter et al. [49] | Ensemble-Vis | | | | | | | | | | | x | | | ■ | | |
| Booshehrian et al. [19] | Vismon | | | | | | | | | | | ■ | x | ■ | x | | |
| Brecheisen et al. [52] | -- | | | | | | | | | | | | x | ■ | x | | |
| Unger et al. [53] | -- | | | | | | | | | | | | ■ | | ■ | | ■ |
| Amirkhanov et al. [73] | -- | | | | | | | | | | | | x | | | | |
| Marks et al. [48] | Design Galleries | | | | | | | | | ■ | | | x | | | | |
| Waser et al. [18] | World Lines | | | | | | | | ■ | x | | | | | x | | |
| * Martins et al. [35] | -- | | | x | | | | | | | | | | | ■ | x | |
| * Wu [72] | -- | | | | x | | | | | | | | | x | x | | |
| Guo et al. [41] | -- | | | | | ■ | | | | | | | | | | | |

Each colored cell indicates a requirement that is important for a given case study/paper, and an 'x' indicates that the requirement remains unmet by users' current tools.

## 3 METHODOLOGY

Our research was divided into four parts. The first part fits with the first layer of Munzner's nested model of visualization design (characterize the domain problem) [23]. The second and third parts of our approach fit with the second layer of Munzner's model (abstracting the domain problems in a manner that informs the design of visualization systems).

First, we investigated eight widely diverse application case studies to gain a first-hand understanding of users' requirements (see Section 4). Each case study started with a stakeholder completing a questionnaire that was designed to capture their aim, current analysis methods and aspirations (see supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2016.2603178), and the authors reading the responses and a key paper about the work. Then we conducted a semi-structured interview with the stakeholder, to probe important issues, and discuss limitations of today's tools for designing and optimizing data processing pipelines.

The second part was to characterize users' requirements in a manner that captured the breadth of the case studies (see Section 5). This took place in workshop-type sessions that involved the paper authors and some colleagues.

To validate and fine-tune the characterization, two of the authors then independently reviewed a broad set of papers, selecting 28 representative papers dealing with visualization techniques for pipeline design and optimization (see Table 1). Differences between the authors' characterizations were resolved by discussion. The 28 papers include all 21 that were selected as a core subset in Sedlmair et al.'s review of visual parameter space analysis [2], as representatives of a much wider body of visualization research.

The fourth part of our method was to identify exemplar solutions for certain aspects of the characterization (see Section 6) and future research challenges (see Section 7.2). The exemplars were drawn from visualization systems that were developed to address some of the challenges identified in the case studies, or described in the 28 papers or more recent related work.

## 4 Application Case Studies

This section describes eight application case studies, highlighting the aim of each and key challenges that users face. These case studies come from a range of application domains, and allowed us to obtain an in-depth understanding of users' requirements from first-hand experience. In three of the case studies (Section 4.2, 4.3, and 4.8) users currently employ scientific visualization techniques, whereas information visualization predominates in the other case studies. In the first five case studies users need to design a data processing pipeline, contrasting with previous work which has primarily involved pipeline optimization (e.g., see the studies reviewed by Sedlmair et al. [2]).

### 4.1 Comparative Genomics

In *comparative genomics* users wish to identify patterns of genetic mutation that are characteristic of factors (e.g., disease, organ and tumor stage) that vary across a collection of hundreds or thousands of DNA samples [1]. This is one of two case studies that clearly involve 'big' data. In this case it is due to veracity. Noise masks interesting features in the data, established processing methods perform aggressive smoothing that removes noise and some features, and output is sensitive to small changes in the thresholds used to differentiate normal regions of DNA from mutated regions. Biologists and bioinformaticians want to develop new methods to detect cross-sample similarities and trends, but the space of possibilities is large and data processing takes hours for a single pipeline run on high-performance computing (HPC) facilities. The output from a single processing run also takes a considerable time to assess, because hundreds of thousands of DNA regions often need to be considered individually. Users also need to take into account the large body of prior research that has identified DNA regions associated with particular diseases.

### 4.2 3D Image Segmentation

The aim of *3D image segmentation* research is to develop methods that automatically identify a structure from volumetric data. One example comes from Steger et al. [24], who developed a pipeline to segment radial-based lymph nodes from a CT scan for cancer diagnosis. The pipeline has multiple steps, with dozens of quantitative parameter choices, and several choices of the used algorithms. The results are assessed via a set of quantitative criteria, viewing 3D graphical output, and making comparisons with multiple references (the subjective nature of segmentation means that two experts are unlikely to produce the same ground truth). In Steger et al.'s pipeline, the algorithms used in certain steps make assumptions about the shape and size of the segmented objects (lymph nodes). Current tools prevented the users from validating that these assumptions were consistent with the characteristics of the final output. Even though each segmentation result is fast to compute (10 seconds), the sheer number of processing steps and parameters, coupled to the sensitivity of results to specific choices, makes pipeline design and optimization difficult.

### 4.3 2D Image Segmentation

Another class of segmentation involves 2D images, for example, *histopathology segmentation* aims to robustly detect contiguous regions of tissue in virtual slides [25]. With a standard pipeline, no single set of parameter settings is optimal for all input images (optimal settings for one image lead to poor segmentation of others). Therefore, users need to design a more sophisticated pipeline. But this requires them to better understand the composite effect of different parameters and to be able to review and make judgements about the segmented output from many images. Some aspects of output assessment are straightforward, but others require input from a domain expert which can lead to delays due to work commitments.

### 4.4 Chemical Engineering

Chemical engineers want to scale-up *chemical process models* to an industrial-scale to make manufacturing efficient. The models are formed by integrating experimental data gathered in laboratory experiments with a theoretical understanding of the chemical reactions, and knowledge of numerical simulations [26]. Each simulation typically only takes a few seconds to compute. However, there are a large number of possible models and variants, each taking the form of a network of chemical reactions. Models are compared using a multitude of graphical plots (e.g., showing time versus concentration), but the engineers do not have the tools to determine confidence intervals for individual model components. This means that the engineers are unable to identify which components are essential to include. And thus, to improve engineers' overall understanding of chemical processes.

### 4.5 Economic Modeling

In *economic modeling*, the goal is to develop a new model based on new economic theory. Modeling experts work with prior assumptions about model parameters and their distributions, to find the model that best matches real-world data from a broad set of basis model specifications [27]. For this purpose, several model variants are analyzed using stochastic simulations. The evaluation of a model is subjective, with modeling experts needing to visually assess a set of probabilistic output functions. This whole process is highly iterative and time-consuming. Users wish to be able to model several trends in parallel and create more complex models. This is, however, limited by current computational tools.

### 4.6 Aircraft Engine Design

*Aircraft engine design* involves parameter tweaking constrained both by high computational time and risk-averseness to making large design changes [28]. The aim is to make small improvements in engine performance, as measured across a basket of up to five measures (fuel efficiency, stall speed, etc.). The engine is modeled using approximately 100 parameters that are highly abstracted from its physical characteristics. It is straightforward for users to identify a set of feasible solutions (a Pareto set), but experts have to use considerable tacit knowledge to determine which is the best out of candidate designs. Even using HPC, it takes days to compute the model, meaning that it is only possible to perform the computation for a small number of parameter settings.

### 4.7 Phylogenetic Trees

Complex output analysis after performing several pipeline steps is the main bottleneck during the comparison of *phylogenetic trees* [29]. Biologists want to determine the ‘true’ evolutionary dependency of species. This is approximated by so-called phylogenetic trees, calculated from the DNA (or other data) of species using a set of algorithms (e.g., sequence alignment, clustering). The calculation of the evolutionary tree has several steps. Each has a set of parameters of different types (quantitative, nominal, type of data used, and even the choice of algorithm as a parameter). Although rules of thumb exist for the parameter settings, the right parameter setting depends on the dataset at hand. Biologists wish to analyze the sensitivity of the output tree to the input parameters, datasets and algorithms that are used [29], [30]. They calculate the trees, which takes up to an hour per tree, and compare the tree structures to analyze: (1) parameter sensitivity, and (2) core structures within trees (i.e., evolutionarily stable subtrees). This is difficult, as algorithmic tree distance functions do not take detailed differences into account and visual exploration of thousands of trees is not feasible. Therefore new visual tree comparison and parameter sensitivity exploration tools needed to be developed [29]. The tools showed that the construction of phylogenetic trees depends—contrary to folk wisdom in the community—to a large extent on clustering and scoring schemes assumptions, but to a lesser extent on the detailed parameters of the underlying evolutionary model.

### 4.8 Molecular Evolution

To study *molecular evolution*, scientists run nanoscale simulations [31]. To understand the results and how the simulation models may be improved, scientists need to be able to view the overall molecular structure, and emergent large and small-scale features that are scattered throughout. However, a ‘big’ volume of data is involved (e.g., 1 million data points, with 1000 dimensions), so HPC resources are typically needed [32], and sometimes a single simulation may take weeks so it is not possible to run a large number of simulations. The high computational demand leads to the use of reduced models that only approximate the true molecular dynamics. Moreover, assessment is complicated by the fact that there may be unknown or counter-intuitive connections between different dimensions in the data. Users wish for new visual analytics methods that would allow them to compare several models and would display the differences between outcomes.

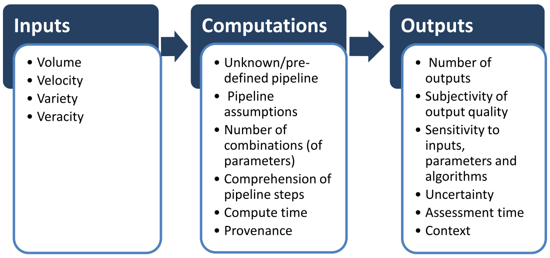

Inputs
• Volume
• Velocity
• Variety
• Veracity
Computations
• Unknown/pre-defined pipeline
• Pipeline assumptions
• Number of combinations (of parameters)
• Comprehension of pipeline steps
• Compute time
• Provenance
Outputs
• Number of outputs
• Subjectivity of output quality
• Sensitivity to inputs, parameters and algorithms
• Uncertainty
• Assessment time
• Context

Fig. 2. Characterization of users’ requirements for pipeline design and optimization. The main categories are inputs, computations and outputs. Further information is given in Sections 5.1, 5.2, and 5.3.

## 5 Characterization of Requirements

This section characterizes users’ requirements for the design and optimization of data processing pipelines. The characterization is derived from the eight case studies (see Section 4). It was refined and validated by reviewing 28 previously published papers that describe application examples and visualization systems for pipeline design and optimization. We use the collective term *scenarios* for all of case studies and papers.

Each requirement in our characterization represents a fundamental barrier to end-users’ ability to design and optimize high-quality data processing pipelines in certain applications. The mapping between requirements and scenarios is summarized in Table 1, with further detail provided in the online supplementary material, available online. Exemplar solutions are provided in Section 6. In analyzing the scenarios, we consider details of the application requirements which sometimes extended beyond the capabilities of the tools that the authors of a given paper were able to provide. The requirement remaining unmet by the presented tools are also indicated in Table 1, with key open challenges summarized in Section 7.2.

The requirements are grouped as follows:

1) *Inputs* cover aspects of the data that is that is fed into a pipeline, and are inspired by the ’Vs’ of ‘big data’.
2) *Computations* cover choices that a user makes in the design of a pipeline and execution of the computational steps.
3) *Outputs* cover requirements that are based on the difficulty of choosing between designs or parameter sets, either on completion of a pipeline or between steps.

### 5.1 Inputs

Our inputs requirements are the *volume, velocity, variety and veracity* of the data (see Fig. 2). They were inspired by the ‘Vs’ of big data. While there are various notions of big data, we focus on those that are most relevant to pipeline design and optimization.

A high data *volume* necessitates using distributed systems architectures, storing the data and performing computations remotely from users, and transmitting outputs over a network to users. Even with massively scalable computations (e.g., using MapReduce) and data structures that allow direct access to multiple abstractions of the data, response times are typically slower than those needed for truly interactive visualization. However, as the molecular evolution case study shows (see Section 4.8), it is sometimes possible to achieve real-time interaction (a latency of 100 ms, or less).

*Velocity*. introduces the requirement to process data as it arrives, and is most challenging when it is impossible to store all of the data. This shifts the challenge to being one of designing a processing pipeline to filter or abstract the data, after which further pipelines are used for detailed data analysis. This is considered as future challenge (see Section 7.2).

*Variety*. comes mainly in three forms, which may be combined to define the requirements of a particular scenario. First, unstructured data is inherently more difficult to analyze than data that are structured into the rows and columns of a conventional database. Examples include free text, images from a multitude of patients [33], multimedia, and the designs of 3D models [34]. Second, as the number of variables increases (e.g., [35]), it is more challenging to deal with inputs. Third, as more sophisticated relationships (e.g., additional factors in the spread of disease in an epidemic [36]) are sought then processing time may increase exponentially.

*Veracity*. It is common for data to be incomplete (missing values) or contain erroneous items (noise, bias, duplicates or errors). Missing data is often treated conservatively, discarding the records concerned or giving missing fields a zero score [37]. In other applications, veracity involves noise that masks patterns that users wish to find and understand, with an example being the comparative genomics case study (see Section 4.1). Veracity brings with it the requirement for visualizations that help users to understand missingness and noise in their data, and their effect on pipeline outputs.

## 5.2 Computations

This set of requirements covers choices that users make about a pipeline and its execution (see Fig. 2). The pipeline may need to be designed as part of the data analysis (i.e., an unknown pipeline), and the steps of the pipeline may need to adhere to certain (external) assumptions. Pipeline execution involves calculating outputs from inputs using certain algorithms and their parameters, which all contribute to the outputs. The complexity of the analysis may mean that the pipeline needs to be iteratively refined and, in multi-step pipelines (see Fig. 1b), users may need to assess outputs and make additional choices between pipeline steps.

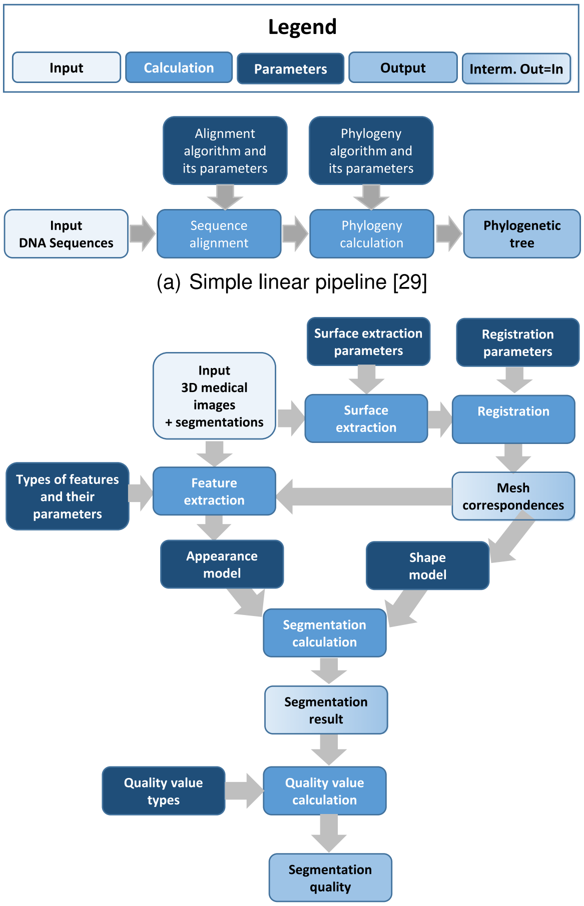
*Unknown pipeline*. Users need to choose a pipeline before processing data with it. Sometimes the pipeline is well-established (e.g., [38], [39]), but in other situations users need to choose between algorithms (e.g., the 3D image segmentation case study in Section 4.2), improve the sophistication of an existing pipeline [17], [40], make a pipeline robust to the characteristics of the input data [12], or design the pipeline from scratch [27], [41], [42]. The latter is particularly true in exploratory analysis, where users are analyzing a new form of data or looking for new patterns (e.g., [9], [10], [11], [43]).

Pipeline steps often make certain *assumptions*. Sometimes these assumptions are either known or may be checked a priori (e.g., the economic modeling case study in Section 4.5). However, on other occasions users need to be able to rigorously check assumptions. For example, implicit choices made for one step of a pipeline may prove to be incompatible with outputs produced by subsequent steps (e.g., the 3D image segmentation case study in Section 4.2). Another case is when assumptions are historic and difficult to update, due to a lack of new data (e.g., basing influenza infection rates on those that occurred during the 1918 pandemic [36]). The rigorous checking of assumptions is impeded by a lack of tools that allow users to holistically assess the analysis workflow [1].

The *number of combinations* is dictated by the range of possibilities being considered for the pipeline (see Unknown Pipeline, above) and the size of the parameter space for each pipeline (the number of samples category of Sedlmair et al. [2]). The latter depends on the number of parameters that are non-trivial to choose (typically only a subset of all the parameters [12]), the range of possible values, and how the values are sampled. As the number of parameters increases, sampling generally becomes sparser. Even where sampling is performed at regular intervals [12] or using stochastic techniques [19], users would benefit from help to make choices once the number of combinations becomes non-trivial. Greater help is required in scenarios where the number of combinations currently forces users to techniques such as a Latin Hypercube [38], or approaches have yet to be developed for choosing a range of combinations (e.g., the comparative genomics and phylogenetic trees case studies in Sections 4.1 and 4.7).

Ease of *comprehension* is also an important requirement when users are designing or optimizing processing pipelines (see Fig. 3). Many pipelines are linear, but as the number of steps increases (e.g., see phylogenetic tree case study in Section 4.7) so does the cognitive complexity of the system being modeled [45]. Another way in which pipelines may be usually complex includes the steps being interrelated (i.e., the pipeline forms a network rather than a linear pipeline). Exemplars include the chemical processes case study, the provision of many options in an environmental flooding scenario [18], and. Alternatively, the pipeline may contain a number of ‘what if’ branches [42]. Sometimes individual parameters are easy for users to understand (e.g., the number of clusters for k-means), but other parameters are abstractions of physical systems [34] or the parameters are difficult to relate to output consequences [11], [46]. Sometimes, users tend to treat the pipeline as a ‘black box’ and accept the default settings [47].

*Compute time* refers to the time required to perform the calculations of a pipeline, so that a user may assess the outputs by using a visualization system or another mechanism. The time is dictated by the quantity of data being analyzed, the resources available and the computational complexity of the algorithms (linear, exponential, etc). Compute time affects the ways in which it is feasible to use visualization to design and optimize processing pipelines. For example, in the aircraft engine design case study (see Section 4.6) the calculation takes days and users employ considerable tacit knowledge to make optimization decisions. When compute time is excessive, alternative approaches include the use of surrogate models that seek a trade-off between accuracy and speed [9], selective sampling of the parameter space [13], [33], and off-line post-processing of output to calculate pipeline alternatives or derived measures [18], [48]. By contrast, in the 3D image



(a) Simple linear pipeline [29]

(b) Complex pipeline [44]

Fig. 3. Simple versus complex pipeline. a) Calculation of phylogenetic trees [29]. b) Pipeline for medical image segmentation [44].

segmentation case study (see Section 4.2), the pipeline calculation takes only seconds and it would be feasible to batch process millions of parameter combinations with an HPC facility. If computation is interactive then users may interactively analyze the effect of parameter changes.

*Provenance* involves recording all of a user's choices, as well as information about the inputs, and is particularly valuable when pipeline design or optimization is iterative, involves several people, or takes place over an extended period of time. However, although capturing provenance is a central tenet of good data analysis practice, it is not often stated as an explicit requirement. Notable exceptions are [36], [42].

### 5.3 Outputs

The outputs of a pipeline's calculations either need to be directly assessed or need to be used as an input to the next step of the pipeline (see Fig. 1b). The requirements for both cases are discussed.

First, the *number of outputs* becomes important when the output is very detailed (e.g., hundreds of thousands of DNA regions; see the comparative genomics case study in Section 4.1) or thousands of simulation results are computed from a single input and parameter set (e.g., using a stochastic model [19]). As the number of outputs increases, so does the requirement for sophisticated visualization techniques to assess the outputs. Those techniques become less scalable as the outputs become more complex and involve, for example, geographic visualizations [49], animations [10] or 3D graphics where users need to inspect many different views [50].

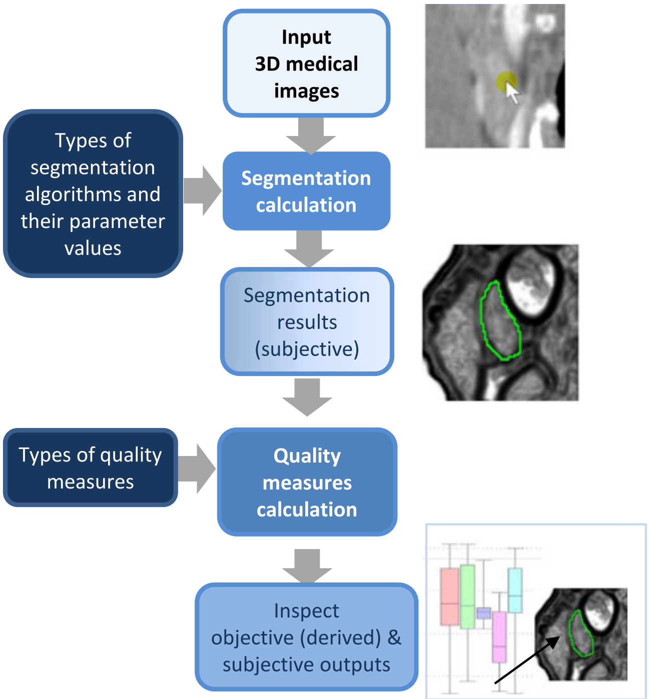

Input 3D medical images
Types of segmentation algorithms and their parameter values
Segmentation calculation
Segmentation results (subjective)
Types of quality measures
Quality measures calculation
Inspect objective (derived) & subjective outputs

Fig. 4. Computation of derived outputs (i.e., quality values) for quantitative evaluation of medical image segmentations [24].

The *subjectivity* of an output's interpretation and assessment decreases the ease with which users may judge the suitability of a given pipeline design or parameter settings. Users need to make non-trivial subjective assessments of outputs that range from competing objective criteria (e.g., [39]), to maps (e.g., [36]), images (e.g., [13]), animations (e.g., [10]), and 3D models (e.g., [43]). Objective measures (derived measures) are sometimes used as a proxy for subjective assessment, to simplify and quantify the output so that low-quality parameter settings may be ruled out. Examples include measures of segmentation quality (e.g., the 3D image segmentation case study in Section 4.2, see Fig. 4), and cell segregation, where metrics for the number and mean area of cells could highlight regions of the parameter space that have similar scores but on inspection result in low- versus high-quality segregation [12].

*Sensitivity* is the size of the change of outputs with respect to the size of changes in inputs and calculation parameters (e.g., a small change in inputs or in parameters may result in large changes in outputs). Together with uncertainty (see below), sensitivity is one of six recurring analysis tasks that were identified in the review by Sedlmair et al. [2]. They are also a common requirement in the case studies described in Section 4. Where sensitivity is important in multi-step pipelines then tools are needed to integrate out from the steps so that users may adopt a rigorous approach to pipeline design and optimization (e.g., the comparative genomics case study).

*Uncertainty* may relate to the precision (the exactness of outputs), completeness (e.g., the effect of missing data), consistency (agreement of interrelated outputs), timeliness (certainty about the currency of data), and credibility (trustfulness of data sources) [51]. Uncertainty of the outputs needs to be considered when assessing the quality of computational outputs. For example, in the economic modeling case study (see Section 4.5), stochastic algorithms result in uncertain results that require detailed inspection by the expert. Despite the frequency with which sensitivity and uncertainty were highlighted as important in an application (see Table 1), it is rare that they appear together as requirements (exceptions are [9], [19], [38], [40], [52], [53] and the molecular evolution case study).

*Assessment Time.* The time that users take to assess output may be a challenge. Sometimes this is due to the high-dimensionality of the data (e.g., [35] and the phylogenetic trees case study in Section 4.7). In other applications the challenge centers on users needing to make a thorough comparison of computed output with baseline or ground-truth data, for example at multiple levels of detail, across widely differing spatial locations, or over time (e.g., [40], [54] and the molecular evolution case study).

It is sometimes important to view output in the *context* of the results of previous research, for example, to validate a new model (e.g., [40], [54]) or to help users interpret new data (e.g., the comparative genomics case study). On other occasions the context is provided by reference data or ground truth. Occasionally the subjectivity of the task may mean that there are several competing ground truths (e.g., the 3D image segmentation case study), which all need to be considered when optimizing data processing.

## 6 Visualization Functionality

This section summarizes the functionality that existing interactive visualization systems provide to help users design and optimize data processing pipelines. The section is divided into three main parts, which map onto the three groups of requirements (Inputs, Computations, and Outputs).

Interactive visualization is an intrinsic part of the solution for big data analysis and, hence, for the four Inputs requirements. Sometimes that solution is achieved by appropriate systems engineering (see Section 6.1), and on other occasions by visualization methods that support computation or output exploration (see Sections 6.2 and 6.3, respectively).

Interactive visualization benefits the Computations requirements in a number of ways. One is by providing an overview of alternative pipeline designs (*unknown pipeline*) or the steps that were taken during analysis (*provenance*) (see Section 6.2.4), and these could be combined to help users understand the consequences of different *assumptions*. Visualization is used directly in the *comprehension* of computations (see Section 6.2.2), but plays only a supporting role in reducing *compute time* (see Section 6.2.3). The *number of combinations* requirement benefits from combining on-the-fly computation with visual exploration (e.g., see the hybrid approach in Section 6.2.1).

Regarding Outputs, interactive visualization has clear benefits for helping users assess *subjectivity*, which is central to many real-world data analysis problems (e.g., [24], [25], [27], [28]). The *sensitivity* and *number of outputs* requirements benefit from visual exploration techniques as described in Sections 6.3.1 and 6.3.2, respectively). *Uncertainty* is a long-standing research topic in visualization [55], but also where some innovative solutions have been produced (e.g., see Fig. 10). *Context* is addressed by the multiple levels of detail and view perspectives provided by many visualization systems, and the ability of some systems to leverage display real estate to show detail in context (e.g., [1], [32]). However, *assessment time* is mainly limited by users and, therefore, only addressed indirectly by visualization.

Fig. 5. Example of a scalable visualization system for molecular dynamic simulations. One of the benefits that users gained was being able to identify fracture modes [32].

The choice and combination of functionality, of course, depends on details of an application's user requirements.

Almost all of the visualization functionality is interactive, from the user input that is needed to select parameters in user-defined and hybrid approaches, to brushing, filtering and other operations when users are investigating visualization input-output correspondence, navigating views to explore outputs, and reviewing the provenance of the pipeline design process.

### 6.1 Visual Assessment of Input Data

Data analysis pipeline creation and optimization often starts with the visual assessment of the input data for their suitability in subsequent calculations. Many interactive visualization approaches for various data types could be used, and a comprehensive overview is beyond the scope of this paper. The reader is referred to reviews, e.g., [56], [57], [58].

We do, however, need to consider how visualization systems can help scale-up processing pipelines to deal with big volumes of data. Such data is typically stored remotely from a user, and so requires distributed visualization systems. Web-based solutions inevitably compromise interactive responsiveness for bandwidth usage, but are well-established in domains such as bioinformatics (e.g., [59]). However, it is possible to achieve real-time interaction with remote rendering through the usage of dedicated graphics cluster (see Fig. 5) or using incremental visualization approaches [60], [61].

Data volume, veracity, variety and velocity still pose challenges for visualization (e.g., see [61], [62]). We discuss this under future research (see Sections 7.2.3, 7.2.5 and 7.2.7).

### 6.2 Visual Support for Computation

Interactive visualization offers support for dealing with a large number of parameters, comprehending computations and computation steps, overseeing time-intensive calculations, creating pipelines and analyzing result provenance.

#### 6.2.1 Large Number of Parameter Combinations

Parameter value selection is widely supported across today's systems. The support ranges from iterative user-defined

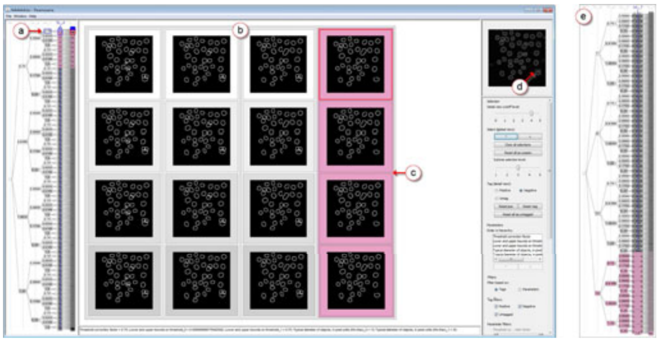


Fig. 6. Automated parameter sampling and user-driven filtering of relevant parameters for 2D image segmentation. This increased the rigor with which a user investigated parameter combinations, and led to a slight but meaningful increase in the quality of the results [12].

selection of model parameters [42], [46], to automated parameter selection (e.g., by random or regular sampling [12], [13], [29]). Moreover, hybrid approaches combine user steering and automatic parameter value sampling [63].

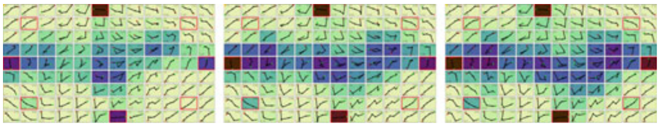
*User-defined selection* offers the user full control over model creation and refinement. Users iteratively choose computation parameters and examine the outputs, until the solution is satisfactory. An example is the system for model selection in time series analysis by Bögl et al. [42]. User-defined selection may be very time consuming and often requires expertise for selecting good parameters for the next iteration. Therefore, it is suited to cases where the calculation of outputs is computationally intensive (e.g., the aircraft engine design case study in Section 4.6) or outputs take a considerable time to assess (e.g., the economic modeling case study in Section 4.5).

By contrast, *automated parameter sampling* offers the possibility to explore a large number of outputs at one time, with minimal need for manual intervention. This option is best suited to cases where individual outputs may be calculated rapidly, and one thing that this allows is the analysis of output sensitivity (e.g., in image segmentation [12] (see Fig. 6) and the phylogenetic trees case study [29]).

*A hybrid approach* is advantageous in large or sparsely sampled parameter spaces, with the user and system working together to choose regions of the space that should be explored in greater detail. One notable exemplar is a system that pre-computes heterogeneity information (e.g., gradients), to provide hints about the most promising paths in time and parameter scale, from which users interactively make specific choices for refinement [63]. In another approach, users explore one or two parameters at a time, by interactively calculating the output variation in those parameter dimensions [9].

### 6.2.2 Comprehension of Computations

Comprehension is supported by 'opening the black box' of computations [64]. This allows the user to examine intermediate calculation results and thus to get a better understanding of the calculation progress and the transformation of inputs into outputs [11]. Two main approaches stand out: (a) progressive visual analytics showing intermediate results during calculation and (b) explanatory visualization showing the progress of calculation after the computation has finished.



(a) Start (b) Progress 1 (c) Progress 2

Fig. 7. Progressive trajectory clustering for movement analysis [65]. User can see intermediate results and adjust further calculation.

*Progressive visual analytics* functionality is often tightly coupled with the option of computational steering [64], [65], [66]. We elaborate on this in the next section as it is often used also for time-intensive computations.

One example of *explanatory visualization* addresses a requirement of the medical image segmentation case study (see Section 4.2) by presenting a visualization of the quality improvement during the iterative 3D medical image segmentation [67].

### 6.2.3 Time Intensive Computations

Time intensive computations such as those in the aircraft engine design case study (see Section 4.6) can be supported by off-line computation [48], steering [68] or the progressive visual analytics approach that was outlined above. They show information about a running computation and incremental results during the computation. Moreover, they offer control over pipeline execution and results, allowing users to adjust parameters during computation instead of waiting for final result [65]. Such approaches are systematized by Mühlbacher et al. [64]. As an example, Schreck et al. [65] introduce the visualization and steering of self-organizing map calculations (see Fig. 7), Stolper et al. [66] show the progress of K-Means calculation and Hellerstein et al. [69] presented a so-called Control Project, which includes incremental calculation and steering of association rule mining algorithms.

### 6.2.4 Pipeline Design and Provenance

*Visual pipeline design* systems provide interactive visual means for creating pipelines (also often referred to as workflows) by combining calculation steps and setting calculation parameters. Some work focused on allowing users to make iterative changes in a pipelines design [17], [70]. Recently, these pipeline creation tools were enhanced with pipeline simplification and workflow suggestions [71], [72] allowing users to create pipelines faster and in a more informed way.

*Provenance* involves recording all of a user's choices (inputs, parameters, computation algorithms) during pipeline design or optimization. As analysis workflows become more complex, capturing provenance is likely to become increasingly important, even though it was only explicitly highlighted in a small amount of the research listed in Table 1. One well-known exemplar is the Vistrails scientific workflow and provenance management system [15], which allows the user to create and to reuse pipelines for visualization and data exploration. Another is the TiMoVA system [42], which integrated a model's history with the visualization of model outputs to show all of a pipeline's steps. Waser et al. [18] propose a system 'World Lines' that shows the history of simulations and allows the user to steer the simulation and compare the simulation runs (see Fig. 8).

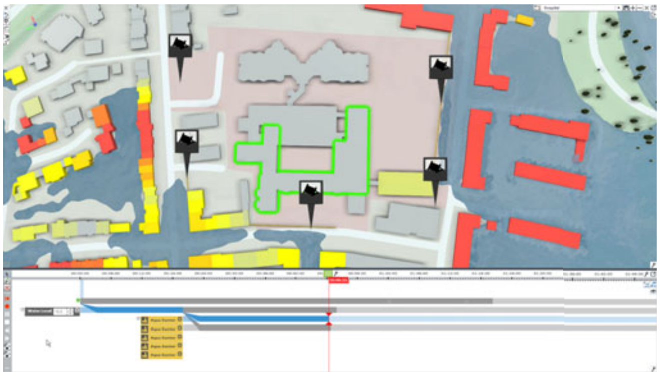


Fig. 8. A World Lines view (bottom) shows the history of a flood simulation (top) and allows the user to steer and compare the simulations. This greatly enhances the ease with which users can compare management strategies [18].

## 6.3 Visual Exploration of Outputs

Dedicated interactive visualization approaches help the user in gaining insights into the sensitivity of outputs to inputs and calculation setting. Other visualization tools support exploration of large number of outputs as well as dealing with subjective output assessment. Some visualization approaches also offer the user with the possibility to make comprehensive output assessments while computation is taking place.

### 6.3.1 Assessing Sensitivity of Outputs to Inputs

Input-output correspondence visualization is often employed to explore the sensitivity of outputs to input parameter values (e.g., [9] and the phylogenetic trees case study in Section 4.7), to enable interactive refinement of parameters to optimize outputs (e.g., [12], [40], [46]), to assess the effect of inputs on outputs ('understanding the black box' [38]) or to interactively analyze the influence of input uncertainty on the output uncertainty through the pipeline [53]. All these cases require that users have the possibility to interactively assess subjective outputs in the context of input parameters. The visualization of input-output correspondence poses a challenge as both inputs and outputs need to be shown simultaneously.

One possibility is to show inputs and outputs in an *integrated view*. For example, 'extended' parallel coordinate plots can be employed for quantitative inputs and outputs, which are treated as variables in the plots [9]. Cupid [43] overlays examples of 3D geometry within parallel coordinate plots of the input parameter values, and uses the same 3D geometry to depict nodes in trees showing output clusters (see Fig. 9).

Alternatively, inputs and outputs can be shown in *linked views* with brushing and filtering [63]. The Influence Explorer allows correspondences to be investigated from opposite perspectives, via a Parameters Window (an input perspective) and a Performances Window (an output perspective) [73].

Input-output correspondence is often shown by the *position of outputs according to the values of input parameters*. The Design Galleries approach uses graphical miniatures as data points in XY plots of two input parameters [48]. Luboschik et al. [38] show inputs on a plot X axis and outputs on the Y axis, thereby indicating the influence of inputs on outputs. Both Ruppert et al. [46] and Booshehrian et al. [19] use a grid-based visualization of outputs. The grid is produced by discretizing the input parameters into a set of intervals, with outputs shown inside the grid cells. Paramorama implements a hierarchical ordering of input parameters, together with miniatures of the output images for user-selected parameter regions [12], [14].

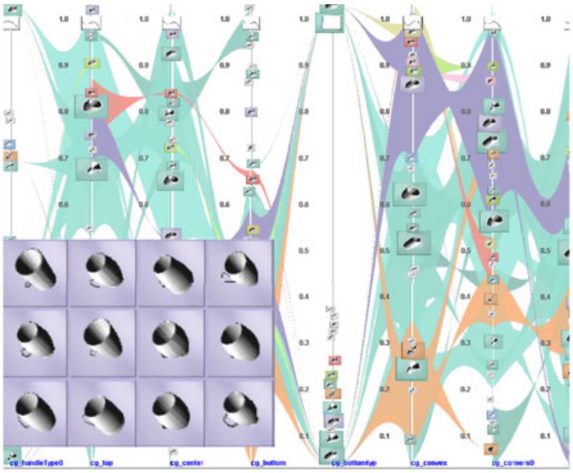


Fig. 9. Cupid [43] system shows the correspondence of input parameters to the outputs via clustering and overlays in a parallel coordinates plot. Cupid now allows users to detect relationships between parameters and identify sensitive parameter ranges.

### 6.3.2 Large Number of Outputs

Visualization systems allow users to inspect outputs in a variety of forms. This is important whenever the output is partly or wholly subjective, a situation encountered in half of the scenarios that are listed in Table 1. Some of the systems focus on individual outputs, and others offer the possibility of exploring a set of outputs or comparing the outputs.

For *exploring individual outputs* we highlight three visualization types. The first is *overlaying* graphical output on top of the ground truth ('perfect output') so that any differences are shown directly (e.g., [12], [14], [52]). A second involves *multiple linked views* [73], where each view is designed to allow questions about the output to be answered from a particular perspective. Examples include environmental modeling, where users need to assess multivariate, image and spatio-temporal output [49], [54], and the linking of histograms, scatterplots, parallel coordinates and function graphs to optimize a fuel injection system [39]. The third involves *computing derived measures* from outputs, because it is easier to visualize those measures for a large parameter space than to show all of the underlying subjective output. (e.g., [33], [42]). The derived measures for outputs of various parameter combinations can then be inspected for assessing output quality (see Fig. 10).

To allow users to *explore a large set of outputs*, a system needs to provide a step change in the quantity of output that may be visualized. One way is by providing visualizations for *multiple levels of detail*. Exemplars structure outputs to facilitate exploration [43], or use one view to provide an overview (e.g., a contour plot or histogram) and others to show the detailed output for a given combination of parameters that a user chooses interactively (e.g., [13], [19]). Alternatively, flexible user interfaces, allow the user to filter out or to focus on

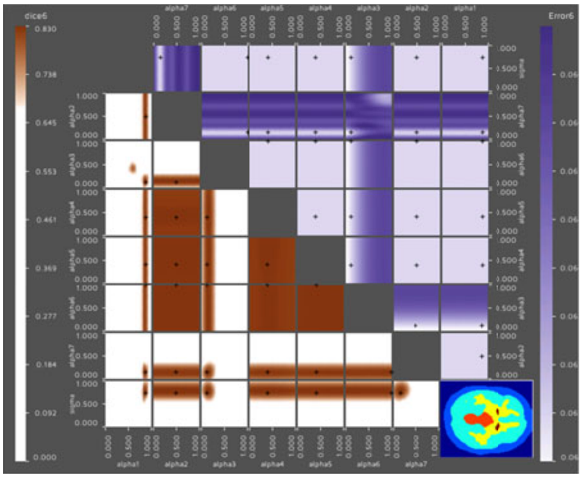


Fig. 10. Tuner: Visualization of output values (lower triangle) together with their uncertainty (upper triangle) for various parameter value combinations. This helps users to understand trade-offs between parameters and select starting locations for data analysis [33].

interesting parts of the dataset. Exemplars include allowing users to specify output constraints to rule out parts of the input parameter space [19], to identify and exclude outliers [41], and filters chosen using one dataset to be applied to others [13]. A third option is to group similar outputs and then show only representatives of each group [29], [43].

The circumstances under which users need to *visually compare outputs* include making fine-grained assessments of sensitivity or understanding the validity of multiple models. Comparison is often supported by *small multiples* [10], [18], [41], [49], [74], [75]. Other exemplars allow users to make comparisons at *multiple levels of detail* (e.g., a plot of time versus lives saved, and maps showing the geographic distribution of lives saved [36]), or *clustering outputs* (e.g., [29], [43]).

## 7 Discussion

We first discuss our characterization of the requirements and then we identify challenges for future visualization research.

### 7.1 Requirement Characterization

Our requirement characterization focuses on data processing and user's involvement in construction and optimization. It covers the three main factors: inputs, computations and outputs. The construction of the factors was challenging as we had to strike a balance between complexity, generality and broadness of coverage of characterization.

As Table 1 highlights, there are some notable differences between the first-hand case studies (see Section 4) and application examples from the literature. Factors that were more prevalent in the case studies were an unknown pipeline (i.e., needing to design rather than simply optimize it), assumptions, assessment time and context. Of particular note is how often assessment time was identified as an important factor in the end-user scenarios. In economic modeling this was due to the difficulty of comprehending the implications of a single output, but in the other scenarios it was primarily due to a combination of the quantity and subjectivity of the output. By contrast, it was notable how often the number of parameter combinations, number of outputs, subjectivity and uncertainty appear as key requirements in our case studies as well as in the literature.

The wider context in which the design process is conducted also influences the choices that users make during pipeline design and optimization (e.g., see [21]). This wider context is outside the scope of the present paper, and we limit our discussion to highlighting three key factors. The first factor comprises constraints on time, budget or resources provided by the organization when performing pipeline decisions. For example, in the economic modeling case study (see Section 4.5), the time constraint influenced the number of parameter settings that could be analyzed by the user in a given time frame for the final result. The second factor is the expertise and diversity of roles of the people who contribute to the analysis. If there are several users then they may have different roles such as bioinformaticians, computer scientists and biologists working together in comparative genomics or phylogenetic tree analysis (see Section 4.1 and 4.7). The third factor is the intended audience (e.g., analysts, managers or the general public), which plays a role when selecting an appropriate communication of the pipeline design strategy.

### 7.2 Future Challenges for Visualization Research

We conclude the discussion by presenting seven challenges that visualization research needs to address. Previous authors have presented challenges at a high level (e.g., data, users, design, and technology [76]) or focusing on particular types of visualization (e.g., scientific [55]). By analyzing users' unmet requirements (see Table 1), and building on the work by Johnson [55], we identify challenges that are key to the successful exploitation of visualization in the design and optimization of data processing pipelines. The challenges start with users' workflow, and then focus on computations and outputs. The challenges associated with display real estate and user interaction cut across the user requirements that are shown in Table 1. We conclude by summarizing additional challenges that visualization systems face for processing big data.

#### *7.2.1 Transform Users' Workflow*

Data processing pipeline is often executed in discrete steps due to the processing time that is involved, meaning that the consequences of choices made in one step on its successors are not rigorously assessed. Situations where users have highlighted concerns include trade-offs between noise removal and feature suppression (see Section 4.1), validating whether the characteristics of 3D image segmentation outputs are consistent with assumptions that are inherent with the methods used in certain pipeline steps (see Section 4.2), and checking whether assumptions about co-evolutionary distance proved true (see Section 4.7). A challenge is to create visualization systems that can 'un-break' users' workflow [1] and that allow the users to holistically assess the consequences of decisions made in each pipeline step on the other steps.

#### *7.2.2 Assist in Parameter Choice*

Some pipelines have tens, or even hundreds, of parameters (e.g., [13]). Research is needed into how visualization systems can provide users with support to select of regions of interest in a parameter space [17], and guidance for choosing parameters that are difficult to comprehend (e.g., training parameters for machine learning [40]). Alternatively, one could research new visual methods with which users could specify output characteristics so that a visualization system may automatically derive suitable parameters [10].

#### 7.2.3 Represent Error and Uncertainty

This is a long-standing challenge [55]. For example, users can only assess sensitivity for a subset of parameter combinations [9], and want to check the parameter settings against diverse inputs in 3D image segmentation or compare the outputs of molecular evolution hypotheses. Users also want to understand uncertainty 'stack-up' over the pipeline steps [52], [53] and need innovative visualizations to understand the interplay of simulation parameters and outcomes [18]. In multi-step pipelines (see Fig. 1), errors and uncertainty in the outputs of one step may increase the veracity of the inputs to the next step. Analyzing and visualizing the flow of uncertainty in pipelines is an ongoing research challenge.

#### 7.2.4 Exploit the Power of Derived Measures

It is often only practical to view a subset of subjective output [36], due to the number of outputs, output size (e.g., large images) or output complexity (e.g., epidemic model outputs). Derived measures offer a solution that speeds up assessment by allowing coarse judgments and comparisons to be made objectively. Yet, the usage of these measures is arguably in its infancy. In particular, users want greater flexibility to adapt derived measures on-the-fly to meet particular needs [19], [42], to be provided with measures that capture qualitative perceptual differences (e.g., [48]) and, when suitable measures are unknown, bring rigor by comparing new measures [11].

#### 7.2.5 Leverage Large Amounts of Display Real Estate

It is common for people to use multiple monitors on their desktop, ultra-high definition (UHD) displays have become a commodity, and tiled displays ('powerwalls') may be created for modest cost. Increasing the display real estate allows users to visualize detail in context (see [55]), and show orders of magnitude than is possible with ordinary displays [77].

One key research challenge is to develop guidelines about how to exploit that real estate. We need to: (a) determine the useful size of a display, taking account of both the momentary capacity of our eyes' photoreceptors [78] and the benefits of physical navigation [79], (b) know how to construct information-rich visualizations that show many variables in a single view, and (c) explore usability issues that are related to the manageability of many views [39].

Another challenge is to gather convincing evidence about the benefits of large amounts of real estate in an application setting. The evidence is largely anecdotal (e.g., [1], [32]).

#### 7.2.6 Improve User Interaction

Interactive visualization allows users to generate a sequence of visualizations answering a particular component of the overall research question. This raises design challenges:

First, how should a visualization tool guide users toward an analysis strategy that progressively simplifies the data (e.g., use histograms and descriptive statistics to exclude variables with low sensitivity, apply dimension reduction techniques to collapse the variable space, and identify factors to be subdivide heterogeneous data into homogeneous sets).

Second, the interaction cost needs to be substantially reduced. One study with well-known tools found that users had to perform an average of 13 motor actions to complete each application-level task (e.g., filter data or format a visualization) [80]. Tools such as Tableau improve the situation, but are still cumbersome for exploring the effect of sets of variables.

Third, back-end computation needs to be seamlessly integrated with user interaction (e.g., to leverage user input in pattern recognition [81], and drill-down to important parameter subspaces [46]). This will require new interfaces and implementations that 'open' black box algorithms [64].

#### 7.2.7 Up-scale for Big Data

The four aspects of big data clearly present challenges for visualization, some of which are the same as those listed above. Large amounts of display real estate will help to address the problems posed by big volume data, by increasing the capability of visualizations to show detail in context and multiple abstractions. Data that is big in terms of variety will benefit from derived measures and interaction strategies that help users to simplify high-dimensional data.

Where veracity is an issue, assessing data quality is an inherent part of analysis. Research is needed to determine how multi-dimensional data visualization techniques may be exploited and integrated within users' analysis workflow.

Finally, high-velocity data compounds the challenges identified above and raises the need for processing pipelines to be simplified, which cuts to the core of the use of visualization systems for pipeline design.

## 8 Conclusion

This article described the requirements for visualization systems supporting pipeline design and optimization. Through eight practical case studies and a review of representative literature we identified users' requirements when designing and optimizing data processing pipelines. We matched user requirements with the functionality that previous visualization systems have provided and derived open challenges for visualization research. The result is a framework that developers can use to relate user requirements to techniques exemplified by those systems, and to implement effective solutions to given application requirements. Visualization researchers will profit from our comprehensive overview of user requirements and unmet visualization challenge for future research.

## Acknowledgments

The work has been partially supported by DFG. The authors are grateful to Hans-Jörg Schulz, Johannes A. Pretorius and Arjan Kuijper for their helpful suggestions to the paper. We also thank the interviewed experts for their insights. All images are reused with authors' permission.

## REFERENCES

[1] R. Ruddle, W. Fateen, D. Treanor, P. Sondergeld, and P. Ouirke, "Leveraging wall-sized high-resolution displays for comparative genomics analyses of copy number variation," in *Proc. IEEE Symp. Biological Data Visualization*, Oct. 2013, pp. 89–96.

[2] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller, "Visual parameter space analysis: A conceptual framework," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2161–2170, Dec. 2014.

[3] N. Ansari and E. Hou, *Computational Intelligence for Optimization*. Berlin, Germany: Springer, 2012.

[4] C. Onwubiko, *Introduction to Engineering Design Optimization*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.

[5] I. B. Huang, J. Keisler, and I. Linkov, "Multi-criteria decision analysis in environmental sciences: Ten years of applications and trends," *Sci. Total Environment*, vol. 409, no. 19, pp. 3578–3594, 2011.

[6] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller, "DimStiller: Workflows for dimensional analysis and reduction," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2010, pp. 3–10.

[7] T. von Landesberger, M. Görner, and T. Schreck, "Visual analysis of graphs with multiple connected components," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2009, pp. 155–162.

[8] J. Choo, H. Lee, J. Kihm, and H. Park, "iVisClassifier: An interactive visual analytics system for classification based on supervised dimension reduction," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2010, pp. 27–34.

[9] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller, "Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction," *Comput. Graph. Forum*, vol. 30, no. 3, pp. 911–920, 2011.

[10] S. Bruckner and T. Möller, "Result-driven exploration of simulation parameter spaces for visual effects design," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1467–1475, Oct. 2010.

[11] L. Padua, H. Schulze, K. Matkovic, and C. Delrieux, "Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections," *Comput. Graph.*, vol. 41, no. 2, pp. 99–113, 2014.

[12] A. J. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle, "Visualization of parameter space for image analysis," *IEEE Trans. Vis. and Comput. Graph.*, vol. 17, no. 12, pp. 2402–2411, Dec. 2011.

[13] S. Bergner, M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad, "ParaGlide: Interactive Parameter Space Partitioning for Computer Simulations," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 9, pp. 1499–1512, Sep. 2013.

[14] A. J. Pretorius, Y. Zhou, and R. A. Ruddle, "Visual parameter optimisation for biomedical image processing," *BMC Bioinformatics*, vol. 16, no. Suppl. 11, 2015, Art. no. S9.

[15] L. Bavoil, et al., "Vistrails: Enabling interactive multiple-view visualizations," in *Proc. IEEE Vis.*, 2005, pp. 135–142.

[16] A. Singh, L. Bradel, A. Endert, R. Kincaid, C. Andrews, and C. North, "Supporting the cyber analytic process using visual history on large displays," in *Proc. Int. Symp. Vis. Cyber Security*, 2011, Art. no. 3.

[17] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser, "Interactive visual steering - rapid visual prototyping of a common rail injection system," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1699–1706, Nov. 2008.

[18] J. Waser, R. Fuchs, H. Ribicic, B. Schindler, G. Bloschl, and E. Gröller, "World lines," *IEEE Trans. Vis. and Comput. Graph.*, vol. 16, no. 6, pp. 1458–1467, Dec. 2010.

[19] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner, "Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making," *Comput. Graph. Forum*, vol. 31, no. 3, pp. 1235–1244, Jun. 2012.

[20] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Proc. IEEE Symp. Visual Languages*, 1996, pp. 336–343.

[21] S. Kandel, A. Päpcke, J. M. Hellerstein, and J. Heer, "Enterprise data analysis and visualization: An interview study," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2917–2926, Dec. 2012.

[22] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Vis. and Comput. Graph.*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013.

[23] T. Munzner, "A nested model for visualization design and validation," *IEEE Trans. Vis. and Comput. Graph.*, vol. 15, no. 6, pp. 921–928, Nov. 2009.

[24] S. Steger, Y. N. Bozoglu, A. Kuijper, and S. Wesarg, "Application of radial ray based segmentation to cervical lymph nodes in CT images," *IEEE Trans. Med. Imag.*, vol. 32, no. 5, pp. 888–900, May 2013.

[25] A. J. Pretorius, D. R. Magee, D. Treanor, and R. A. Ruddle, "Visual parameter optimization for biomedical image analysis: A case study," in *Proc. SIGRAD Interactive Visual Anal. Data*, 2012, pp. 67–75.

[26] C. S. MacLeod and F. L. Muller, "On the fracture of pharmaceutical needle-shaped crystals during pressure filtration: Case studies and mechanistic understanding," *Organic Process Res. Develop.*, vol. 16, no. 3, pp. 425–434, 2012.

[27] P. Levine, P. McAdam, and P. Welz, "On habit and the socially efficient level of consumption and work effort," Univ. Surrey, Discussion Papers Economics, *Tech. Rep. DP 07/13*, Sep. 2013.

[28] T. W. Simpson, V. Toropov, V. Balabanov, and F. A. Viana, "Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not," in *Proc. AIAA/ISSMO Multidisciplinary Anal. Optimization Conf.*, 2008, pp. 10–12.

[29] M. Hess, et al., "Visual exploration of parameter influence on phylogenetic trees," *IEEE Comput. Graph. Appl.*, vol. 34, no. 2, pp. 48–56, Mar. 2014.

[30] S. Bremm, T. von Landesberger, M. Hess, T. Schreck, P. Weil, and K. Hamacher, "Interactive visual comparison of multiple trees," in *IEEE Conf. Visual Analytics Sci. Technol.*, Oct. 2011, pp. 31–40.

[31] S. Weißgräber, F. Hoffgaard, and K. Hamacher, "Structure-based, biophysical annotation of molecular coevolution of acetylcholinesterase," *Proteins Structure Function Bioinformatics*, vol. 79, no. 11, pp. 3144–3154, 2011.

[32] K. Reda, A. Knoll, K.-I. Nomura, M. E. Papka, A. E. Johnson, and J. Leigh, "Visualizing large-scale atomistic simulations in ultra-resolution immersive environments," in *Proc. IEEE Symp. Large-Scale Data Anal. Vis.*, 2013, pp. 59–65.

[33] T. Torsney-Weir "Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 1892–1901, Dec. 2011.

[34] D. Coffey, C.-L. Lin, A. G. Erdman, and D. F. Keefe, "Design by dragging: An interface for creative forward and inverse design with simulation ensembles," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2783–2791, Oct. 2013.

[35] R. M. Martins, D. B. Coimbra, R. Minghim, and A. Telea, "Visual analysis of dimensionality reduction quality for parameterized projections," *Comput. Graph.*, vol. 41, no. 2, pp. 26–42, 2014.

[36] S. Afzal, R. Maciejewski, and D. Ebert, "Visual analytics decision support environment for epidemic modeling and response evaluation," in *Proc. IEEE Conf. Visual Analytics Sci. Technol.*, Oct. 2011, pp. 191–200.

[37] T. Gschwandtner, J. Gärtner, W. Aigner, and S. Miksch, "A taxonomy of dirty time-oriented data," in *Multidisciplinary Research and Practice for Information Systems*. Berlin, Germany: Springer, 2012, pp. 58–72.

[38] M. Luboschik, S. Rybacki, F. Haack, and H.-J. Schulz, "Supporting the integrated visual analysis of input parameters and simulation trajectories," *Comput. Graph.*, vol. 39, pp. 37–47, 2014.

[39] Z. Konyha, K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser, "Interactive visual analysis of families of function graphs," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 6, pp. 1373–1385, Nov. 2006.

[40] H. Piringer, W. Berger, and J. Krasser, "HyperMoVal: Interactive visual validation of regression models for real-time simulation," *Comput. Graph. Forum*, vol. 29, no. 3., pp. 983–992, 2010.

[41] Z. Guo, M. O. Ward, and E. A. Rundensteiner, "Model space visualization for multivariate linear trend discovery," in *Proc. IEEE Symp. Visual Analytics Sci. Technol.*, 2009, pp. 75–82.

[42] M. Bögl, W. Aigner, P. Filzmoser, T. Lammarsch, S. Miksch, and A. Rind, "Visual analytics for model selection in time series analysis," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2237–2246, Dec. 2013.

[43] M. Beham, W. Herzner, M. E. Gröller, and J. Kehrer, "Cupid: Cluster-based exploration of geometry generators with parallel coordinates and radial trees," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1693–1702, 2014.

[44] T. Von Landesberger, S. Bremm, M. Kirschner, S. Wesarg, and A. Kuijper, "Visual analytics for model-based medical image segmentation: Opportunities and challenges," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 4934–4943, 2013.

[45] M. W. Golay, P. H. Seong, and V. P. Manno, "A measure of the difficulty of system diagnosis and its relationship to complexity," *Int. J. Gen. Syst.*, vol. 16, no. 1, pp. 1–23, 1989.

[46] T. Ruppert, J. Bernard, A. Ulmer, H. Lücke-Tieke, and J. Kohlhammer, "Visual access to an agent-based simulation model to support political decision making," in *Proc. Int. Conf. Knowl. Technol. Data-Driven Business*, 2014, pp. 16:1–16:8.

[47] S. Bottomley, "Bioinformatics: Smartest software is still just a tool," *Nature*, vol. 429, no. 6989, pp. 241–241, 2004.

[48] J. Marks, et al., "Design galleries: A general approach to setting parameters for computer graphics and animation," in *Proc. Annu. Conf. Comput. Graph. Interactive Techn.*, 1997, pp. 389–400.

[49] K. Potter, et al., "Ensemble-Vis: A framework for the statistical visualization of ensemble data," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2009, pp. 233–240.

[50] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser, "Interactive visual analysis of complex scientific data as families of data surfaces," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1351–1358, Nov./Dec. 2009.

[51] J. Thomson, E. Hetzler, A. MacEachren, M. Gahegan, and M. Pavel, "A typology for visualizing uncertainty," *Electron. Imag.*, vol. 5669, pp. 146–157, 2005.

[52] R. Brecheisen, A. Vilanova, B. Platel, and B. ter Haar Romeny, "Parameter sensitivity visualization for DTI fiber tracking," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1441–1448, Nov. 2009.

[53] Y. Wu, G. X. Yuan, and K. L. Ma, "Visualizing flow of uncertainty through analytical processes," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2526–2535, Dec. 2012.

[54] A. Unger, S. Schulte, V. Klemann, and D. Dransch, "A visual analysis concept for the validation of geoscientific simulation models," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2216–2225, Dec. 2012.

[55] C. Johnson, "Top scientific visualization research problems," *Comput. Graph. Appl.*, vol. 24, no. 4, pp. 13–17, 2004.

[56] J. Kehrer and H. Hauser, "Visualization and visual analysis of multifaceted scientific data: A survey," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 3, pp. 495–513, Mar. 2013.

[57] T. Von Landesberger, "Visual analysis of large graphs: State-of-the-art and future research challenges," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1719–1749, 2011.

[58] K. Brodlie, R. A. Osorio, and A. Lopes, "A review of uncertainty in data visualization," in *Expanding the Frontiers of Visual Analytics and Visualization*. Berlin, Germany: Springer, 2012, pp. 81–109.

[59] B. Giardine, et al., "Galaxy: A platform for interactive large-scale genome analysis," *Genome Res.*, vol. 15, no. 10, pp. 1451–1455, 2005.

[60] S. Frey, F. Sadlo, K.-L. Ma, and T. Ertl, "Interactive progressive visualization with space-time error control," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2397–2406, Dec. 2014.

[61] H. J. Schulz, M. Angelini, G. Santucci, and H. Schumann, "An enhanced visualization process model for incremental visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 7, pp. 1830–1842, Jul. 2016.

[62] N. Boukhelifa and D. J. Duke, "Uncertainty visualization: Why might it fail?" in *Proc. CHI Extended Abstracts Human Factors Comput. Syst.*, 2009, pp. 4051–4056.

[63] K. Matkovic "Visual analytics for complex engineering systems: Hybrid visual steering of simulation ensembles," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1803–1812, Nov. 2014.

[64] T. Mühlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit, "Opening the black box: Strategies for increased user involvement in existing algorithm implementations," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1643–1652, Dec. 2014.

[65] T. Schreck, J. Bernard, T. Von Landesberger, and J. Kohlhammer, "Visual cluster analysis of trajectory data with interactive kohonen maps," *Inf. Vis.*, vol. 8, no. 1, pp. 14–29, 2009.

[66] C. D. Stolper, A. Perer, and D. Gotz, "Progressive visual analytics: User-driven visual exploration of in-progress analytics," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 1653–1662, Nov. 2014.

[67] T. von Landesberger, "Opening up the black box of medical image segmentation with statistical shape models," *Visual Comput.*, vol. 29, no. 9, pp. 893–905, 2013.

[68] J. D. Mulder, J. J. van Wijk, and R. van Liere, "A survey of computational steering environments," *Future Generation Comput. Syst.*, vol. 15, no. 1, pp. 119–129, 1999.

[69] J. M. Hellerstein, "Interactive data analysis: The control project," *Computer*, vol. 32, no. 8, pp. 51–59, 1999.

[70] K.-L. Ma, "Image graphs: A novel approach to visual data exploration," in *Prco. IEEE Conf. Vis.*, 1999, pp. 81–88.

[71] E. Santos, L. Lins, J. P. Ahrens, J. Freire, and C. T. Silva, "VisMashup: Streamlining the creation of custom visualization applications," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1539–1546, Nov.\Dec. 2009.

[72] D. Koop, C. E. Scheidegger, S. P. Callahan, J. Freire, and C. T. Silva, "Viscomplete: Automating suggestions for visualization pipelines," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1691–1698, Nov.\Dec. 2008.

[73] B. Spence, L. Tweedie, H. Dawkes, and H. Su, "Visualisation for functional design," in *Proc. IEEE Inf. Vis.*, 1995, pp. 4–10.

[74] A. Amirkhanov, C. Heinzl, M. Reiter, and E. Gröller, "Visual optimality and stability analysis of 3DCT scan positions," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1477–1486, Nov.\Dec. 2010.

[75] R. G. Raidou, M. Breeuwer, A. Vilanova, U. A. van der Heide, and P. J. van Houdt, "The iCoCooN: Integration of cobweb charts with parallel coordinates for visual analysis of DCE-MRI modeling variations," in *Proc. Eurographics Workshop Visual Comput. Biol. Med.*, 2014, pp. 11–20.

[76] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, *Mastering The Information Age-Solving Problems with Visual Analytics*. Goslar, Germany: EU, 2010.

[77] C. Andrews, A. Endert, B. Yost, and C. North, "Information visualization on large, high-resolution displays: Issues, challenges, and opportunities," *Inf. Vis.*, vol. 10, no. 4, pp. 341–355, 2011.

[78] C. Ware, *Information Visualization: Perception for Design*. Amsterdam, Netherlands: Elsevier, 2012.

[79] R. Ball, C. North, and D. A. Bowman, "Move to improve: Promoting physical navigation to increase user performance with large displays," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2007, pp. 191–200.

[80] C. Rooney and R. A. Ruddle, "A new method for interacting with multi-window applications on large, high resolution displays," in *Theory and Practice of Computer Graphics*. Goslar, Germany: Eurographics, 2008, pp. 75–82.

[81] A. Endert, P. Fiaux, and C. North, "Semantic interaction for visual text analytics," in *Proc. SIGCHI Conf. Human factors Comput. Syst.*, 2012, pp. 473–482.

**Tatiana von Landesberger** received the PhD degree in 2010. She is a group leader in Interactive Graphics Systems Group, Technische Universität Darmstadt. She focuses on visual analytics of complex and large data in various applications.

**Dieter W. Fellner** is a professor with TU Darmstadt, Germany, and a director with the Fraunhofer IGD. He chairs the CGV Institute, TU Graz, is CEO of the Fraunhofer Austria Research and board member of Fraunhofer Project Centre for Interactive Digital Media, NTU, Singapore. He focuses on visual computing.

**Roy A. Ruddle** received the PhD degree in psychology from the University of Wales. He is a professor of computing with the University of Leeds. He focuses on navigation in virtual and information spaces, and data visualization on high-resolution displays.