

Supporting the Integrated Visual Analysis of Input Parameters and Simulation Trajectories

Martin Luboschik^a, Stefan Rybacki^b, Fiete Haack^b, Hans-Jörg Schulz^a

^aComputer Graphics Group, University of Rostock, Germany
^bModelling and Simulation Group, University of Rostock, Germany

Abstract

The visualization of simulation trajectories is a well-established approach to analyze simulated processes. Likewise, the visualization of the parameter space that configures a simulation is a well-known method to get an overview of possible parameter combinations. This paper follows the premise that both of these approaches are actually two sides of the same coin: Since the input parameters influence the simulation outcome, it is desirable to visualize and explore both in a combined manner. The main challenge posed by such an integrated visualization is the combinatorial explosion of possible parameter combinations. It leads to insurmountably high simulation runtimes and screen space requirements for their visualization. The Visual Analytics approach presented in this paper targets this issue by providing a visualization of a coarsely sampled subspace of the parameter space and its corresponding simulation outcome. In this visual representation, the analyst can identify regions for further drill-down and thus finer subsampling. We aid this identification by providing visual cues based on heterogeneity metrics. These indicate in which regions of the parameter space deviating behavior occurs at a more fine-grained scale and thus warrants further investigation and possible re-computation. We demonstrate our approach in the domain of systems biology by a visual analysis of a rule-based model of the canonical Wnt signaling pathway that plays a major role in embryonic development. In this case, the aim of the domain experts was to systematically explore the parameter space to determine those parameter configurations that match experimental data sufficiently well.

Keywords: parameter space visualization, trajectory visualization, multi-scale visualization, interactive computational steering

1. Introduction

Exploring the parameter space as well as exploring simulation trajectories both struggle with the question, which respective regions should be observed for interesting behavior. And even if these regions were known, it is still open at which level of granularity to look at these regions in order to guarantee that all crucial information is captured. The first question relates to the *extent* of the observation, while the second relates to the *grain* of observation – together they define the observation's *scale* [1, pp.55-65]. When analyzing the mere outcome of a single simulation, the problem of finding the right scale of observation is already challenging [2]. As one can independently drill down in the two aspects of input parameters and simulation output, their combined exploration increases the search space for a suitable scale dramatically. As a result of this, screen space requirements and computation times increase as well, because it requires to precompute simulations for the multitude of parameter combinations to be able to show them in concert.

In this paper, we contribute a Visual Analytics approach that addresses this challenge and aims to achieve such an integrated visualization of parameter space and resulting simulation trajectories. This approach is based on a coarsely sampled parameter space to address the challenge of high computation times, for which the simulation output is visualized at an even coarser scale to address the challenge of high screen space re-

quirements. To aid in navigating this multi-scale setup, we further provide means to inform the analyst in a coarse-grained overview where it is worthwhile to drill down towards more fine-grained scales. At first, such a drill-down will show the more fine-grained simulation results at lower levels that have already been computed. When the drill-down has exhausted the computed results, it can be used to make an informed decision on whether to run additional simulations for a more fine-grained subsampling. This way, only those regions that actually exhibit behavior of interest are re-simulated at a more fine-grained scale. Through this close tie-in between computation and visual exploration, we effectively avoid an impossible exhaustive simulation, visualization, and exploration of the entire parameter space.

The following Section 2 will briefly discuss the existing approaches for parameter space visualization, before we will introduce our overall Visual Analytics approach in Section 3. This approach is given as a generic workflow that is applicable as a domain-independent solution strategy and then detailed for our specific context that yields time-series data (1-dimensional trajectories) as outcome of a stochastic simulation. Our realization of this approach is further exemplified by a use case from the domain of systems biology in Section 4. This example investigates an 11-dimensional continuous parameter space for which stochastic simulations with 4096 sampled time steps and 50 replications each are performed. The advantages that our ap-

proach has over traditional approaches for inspecting such data, as well as its inherent limitations are discussed in Section 5, before giving some concluding remarks and an outlook on future work in Section 6.

2. Related Work

The dependency between input parameters and computation outcome has been subject of visualization research for quite some time. The simplest kind of parameter visualization poses a concrete parameter combination and seeks to visualize the corresponding computation outcome. Examples for such visualizations range from early ones, such as the Influence Explorer [3], to more recent and more sophisticated ones, such as the system described in [4]. While the earlier example features most prominently a number of interactive sliders to set the desired parameter combination, the setting of parameters is already more advanced in the later example, in which variations of control parameters are shown in histograms, scatterplots and alike. Some techniques even integrate the computation outcome into the parameter setting interface as with Image Graphs [5] or spreadsheet interfaces [6].

Computationally and visually more challenging are techniques that have the opposite take on the problem: given a desired computation outcome, they want to determine and visualize combinations of input parameter values that will (approximately) lead to this outcome. To ask the question this way around is a more recent challenge that has led to the development of a number of novel visualization approaches. Among them are result-driven explorations of visual effects [7, 8, 9], the accuracy-driven exploration of simulation results [10], and the multi-dimensional optimization of quantification indicators proposed in the Vismon system [11].

All of these approaches have in common, that they aim at individual parameter combinations of simulation results. Thus, visualizations of the entire parameter space that show the simulation outcome for a large number of parameter combinations fall into a third category of parameter visualization. Examples for this category are the parameter space visualization for image analysis [12] and the 2-dimensional embedding of high-dimensional parameter spaces for navigation and interaction [13]. This third category faces the same challenges of high computation time and high screen space requirements as we have outlined them in Section 1. In most cases, visualizations of this category thus apply a sampling of the parameter space as well, in order to cope with these challenges. One solution that enables the user to focus on known regions of interest and prevents him from spending time exploring meaningless parameter combinations is the use of presets, as it has been utilized by [13]. Beyond presets, first approaches like [14] or the Tuner system [15] aim to point the user towards locations in the parameter space that might be of interest but are not yet covered by the current parameter sampling. They use estimation and objective functions to compute these locations, albeit noting that this may not be possible to extend to all application domains – in particular not to those that underlie random processes, like stochastic simulation does.

Drilling down and refining an initial sampling of the parameter space without any estimation – as our approach aims to achieve it – is an application of the domain of *multi-scale visualization*. There are several approaches that aim to provide solutions for investigating data at different scales. These can be divided into approaches that permit their interactive exploration by switching between scales and approaches that mix different scales by partially embedding high resolution data into coarser scales.

The interactive approaches rely on common hierarchical navigation techniques, such as drill-down and roll-up [16], to switch seamlessly between different scales. Sophisticated examples utilizing this navigation scheme are the SignalLens [17], ChronoLenses [18], or the Stack Zooming approach [19]. As such an interactive approach requires some time to fully explore the data, one can employ methods that detect and show regions or points of interest where a drill-down might be worthwhile. For example, the approach of [2] computes the local differences between subsequent scales and depicts them in so-called *heterogeneity bands* alongside the visualization. These alert users where deviating behavior from the currently shown time-plot can be observed by drilling down into scales that are currently below pixel-size and thus not discernable.

The embedding approaches use often very similar measures to the interscale differences described above, but they utilize this information differently by embedding the detected patterns on the finer scale into the coarser scale of the overall behavior of the data [20]. Depending on what patterns these measures detect and subsequently embed, these approaches are called, for example, *outlier-preserving* [21] or *peak-preserving* [22]. Others techniques, such as the Clustering Visualization Spreadsheet [23] try to reintroduce data characteristics that may have been missed due to masking effects back into the visualization.

So far, to the best of our knowledge, both approaches – parameter space visualization and multi-scale visualization – have not been brought together. Yet we see a clear need to do so, as multi-scale visualization can be used to address the challenges that parameter space visualization poses beyond the mere support via presets. They can aid the human analyst in choosing just the right scale to run the simulations at and to look at the resulting outcome, so that computation time and screen space can be allotted to exactly those regions in the parameter space and those parts of the simulation results that show behavior of interest. The next section outlines how we propose to combine these two visualization approaches in order to achieve this.

3. Our Principal Approach and Visual Analytics Setup

Our principal approach consists of three stages which each relate to a different aspect of the Visual Analytics process – the data, the visualization, and the insight gained by the human analyst. This overall approach is described in the following section, before the three stages and their realization in our visual analytics setup are introduced in detail in the sections thereafter.

3.1. Our Principal Approach

Visual Analytics is all about the back and forth between computational analysis, visual exploration, and the human in the loop who is steering this process. In our case, the subject of analysis is a model of a real system and its multi-dimensional parameter space, whereas the computational method of analysis is its simulation. While its purely computational analysis would be too runtime-intensive to be practical, the combination of (I) computational analysis, (II) visual exploration, and the (III) decision making abilities of the human in the loop can actually generate meaningful insights even in such a complex setting. This is achieved by iteratively carrying out the following three stages:

Stage I: Computational Analysis. This stage performs the required computations. It encompasses the three steps of (1) sampling the parameter space, (2) running the simulation for each sampled parameter combination, and (3) recording the simulation results for a given duration at given intervals. The main concept of this stage is that it only runs the time-intensive simulations for a coarse-grained subdivision of the parameter space and logs their output at rather coarse time intervals, so that it can be carried out in reasonable time.

Stage II: Visual Exploration. This stage depicts the generated coarse-grained simulation trajectories by combining methods from parameter space visualization and multi-scale visualization. To first generate and then interactively explore this visualization, we follow the three steps of (1) delineating the multi-dimensional parameter space, (2) choosing a sequence of scales, and performing a (3) targeted drill-down along the chosen scales. The core idea of this stage is that it allows the human analyst to systematically identify those regions of the input parameter space and those time intervals of the simulation trajectory that exhibit behavior of interest and thus warrant further investigation.

Stage III: Decision Making. This stage captures the informed choices the analyst can make based on his observations to further refine the simulation of future configurations. As the analyst can reconfigure the simulation with respect to the parameter space and to the simulation trajectories, two independent decisions have to be made on (1) refining the input parameter space and (2) refining the recorded simulation output. The aim behind this stage is that the analyst can narrow down the extent of the parameter space to scan and of the time interval to observe, in order to be able to increase their grain in the next iteration of these three stages.

Different application scenarios require different simulation approaches, which in turn require different sets of parameters and produce different kinds of outputs. As a result, the principal approach presented above must be tailored to each individual scenario and its particular parameters and simulation results by instantiating it with concrete analysis and visualization methods. The outcome is a procedure that follows the overall workflow of the principal approach, but is customized to the

case at hand. The following sections detail our particular realization of the principal approach for 1-dimensional simulation trajectories produced by stochastic simulation runs.

3.2. Stage I: Computational Analysis

This stage assumes a model with n parameters and an upper limit of s simulations to be run. As the used simulation method is stochastic, it is repeated for r simulation runs whose results are recorded for the duration of p time points.

1. Sampling the Parameter Space. Since we are only able to run a limited number s of simulations, yet have to take into account that each simulation is repeated r times due to the stochasticity, effectively only $s \div r$ simulations of different parameter combinations are possible. Nevertheless, we want these simulations to be evenly distributed across the parameter space, so that their combined outcome captures the array of all potential outcomes as best as possible.

Naïvely, one can sample the n -dimensional parameter space evenly at k points per parameter to yield k^n simulations to be run. Yet, depending on the number of repetitions r , this often leads to the effect that too few simulations can actually be performed to adequately cover the parameter space – i.e., with only 1 or 2 samples per parameter. To resolve this issue, we instead use an alternative sampling method based on Latin Hypercubes (LHS) [24, 25]. It aims to pick c parameter combinations from the n -dimensional parameter space, so that an even coverage is achieved and at the same time, $c \leq s \div r$ is ensured.

2. Running the Simulation. For each of the combinations determined in the first step, a simulation is performed. To speed-up their computation, we can parallelize the simulation to be run across multiple machines. For the particular case of stochastic simulation, we can subdivide the computational load by the repetitions [26] or by the parameter combinations [27]. For greatest flexibility, we utilize a combined approach that takes both variants into account.

3. Recording the Simulation Trajectory. Considering that storing the output data from a simulation may require a substantial part of the overall simulation time warrants to single out this step [28]. We record the simulation trajectories of all simulation runs at a fixed time interval that defines the temporal grain of our observation and yields p points in time at which the simulation state is captured. It is important that the time intervals are fixed across all simulation runs, so that the repetitions of the simulations are recorded at the same time points and can thus be averaged into a single resulting trajectory to even out the effects of the stochasticity. In this context, one has to note that even though a simulation may run for a certain duration at a certain level of granularity (e.g., step size), the duration and granularity that are required for later analysis may be only a subset of these. For example, in many cases the simulation state is not recorded during an initial setting time of a given duration as it bears no meaning until the simulation becomes reasonably stable [29, p.101].

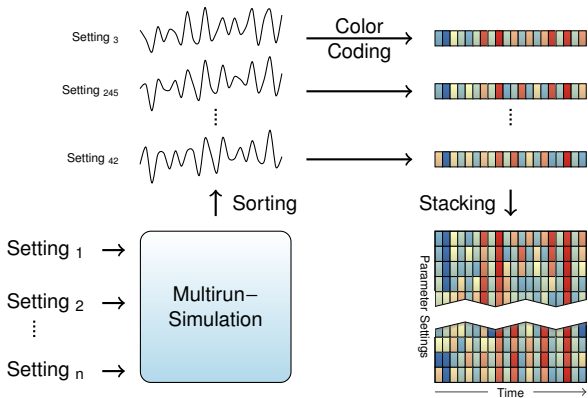


Figure 1: 2D visualization of simulation runs. After sorting the simulation trajectories, each trajectory is color-coded into a pixel row and then stacked to form a 2D image.

After processing these three steps, we have a data basis of c simulation trajectories that were averaged from their r repetitions, each consisting of p time points. The next stage addresses the issue of their visualization and interactive exploration.

3.3. Stage II: Visual Exploration

This stage aims to make the data resulting from Stage I interactively available in a visual analysis setup. In the following discussion, we assume that c and p each exceed by far the number of horizontally or vertically available pixels in the visual setup, as it is not only the worst case, but also the most common case.

1. Delineating the Parameter Space. Since the parameter space is n -dimensional, but we want to depict it on the screen, it has to be transformed/projected into 2-dimensional space. Such a projection is a key part of all parameter space visualizations, which use different ways to do so. For example, in [12] a tree structure is used to map the different parameter combinations to a linear sequence represented by the tree's leaves, whereas [13] uses a 2-dimensional embedding. In our case, we use a 1-dimensional embedding (i.e., linear sorting) that keeps the second dimension for displaying the simulation trajectories. While the order of the time points of these trajectories is given by their inherent temporal sequence, the order of the parameter combinations can be freely arranged (see Figure 1).

To arrange them in a manner that will permit the user to perceive regularities and relationships in the visualization, we seek to define an order that places parameter combinations close together either according to the simulation trajectories or according to the parameters themselves. The first sorting option allows for analyzing the influence of parameters, while the second ordering focuses on the detection of interesting behavior and on a subsequent refinement of the sampling.

For both orderings, we have experimented with several delineation approaches among which are Euclidean distance, gradient distances, space-filling curves, Hausdorff distance, and Fréchet distance. In the end, we achieved the best results by using an averaged squared distance (cp. [30]) for sorting by the

simulation trajectories and an averaged Euclidean distance for the parameter sorting. Nevertheless, to handle alternative setups the user can interactively select between the above distances and chose the simulation trajectories, a certain interval of them, the parameters or an according parameter subset to be the base of sorting.

2. Choosing a Sequence of Scales. As a result of the last step, we have a 1-dimensional embedding of the parameter space mapped to the vertical axis as well as the simulation trajectory for each sampled parameter combination mapped to the horizontal axis. As c and p exceed the dimensions of the viewing area, we need to show an overview at a coarser scale first, in which the analyst can then drill down to look at more detailed scales. For this coarsening, we apply a bisection of the resolution, which is performed by recursively sampling the parameter combinations and recursively aggregating the temporal scale.

As both aspects can be navigated independently (i.e., we can change the scale of the parameters separately from the scale of the simulation trajectories), this results in a 2-dimensional array of all possible scale combinations (see Figure 2(a)). A zoom-in/drill-down operation would usually scale both aspects simultaneously – i.e., when the temporal granularity of the simulation trajectory doubles, one would also expect the resolution of the parameter space to double. This means, that such a traditional drill-down would only use the scales on the diagonal of that array and thus limit the exploratory analysis to viewing only these. To increase the flexibility of this process, we allow the user to interactively define his own drill-down path through that array (see Figure 2(b)).

In order to support the user by hinting at the most promising paths across the scales, we utilize the heterogeneity-based approach from [2]. Its main idea is, to use heterogeneity information between neighboring scales (see Figure 2(a)) as an indicator for noteworthy information at the more fine grained scales. This heterogeneity information can be generated by various metrics – each of which capturing different notions of difference or divergence between subsequent scales. We provide a number of such metrics (e.g., the absolute difference and the gradient) for the user to interactively choose from. We aggregate the computed, detailed heterogeneity information between neighboring scales and encode it into the so called *scale selection matrix* (see Figure 2(b)). Using a color-coding, high heterogeneities are represented by saturated colors along the borders between the matrix' entries. Based upon this interactive matrix, the user can select a sequence of scales (see Figure 2(b)) to be used for the exploratory drill-down of the data (see Figure 2(c)). This selection is supported by means that allow the user to select column-wise sequences, row-wise sequences, and arbitrary sequences along straight lines drawn onto the matrix. Figure 3(a) shows the matrix display within our application.

3. Targeted Drill-down along the Chosen Scales. Once the path through the scales is chosen, we show the actual numerical values of the simulation trajectories by mapping them onto color by using a bivariate color scale from [31] (cp. Figure 1). It encodes positive (red) and negative (blue) deviation from a

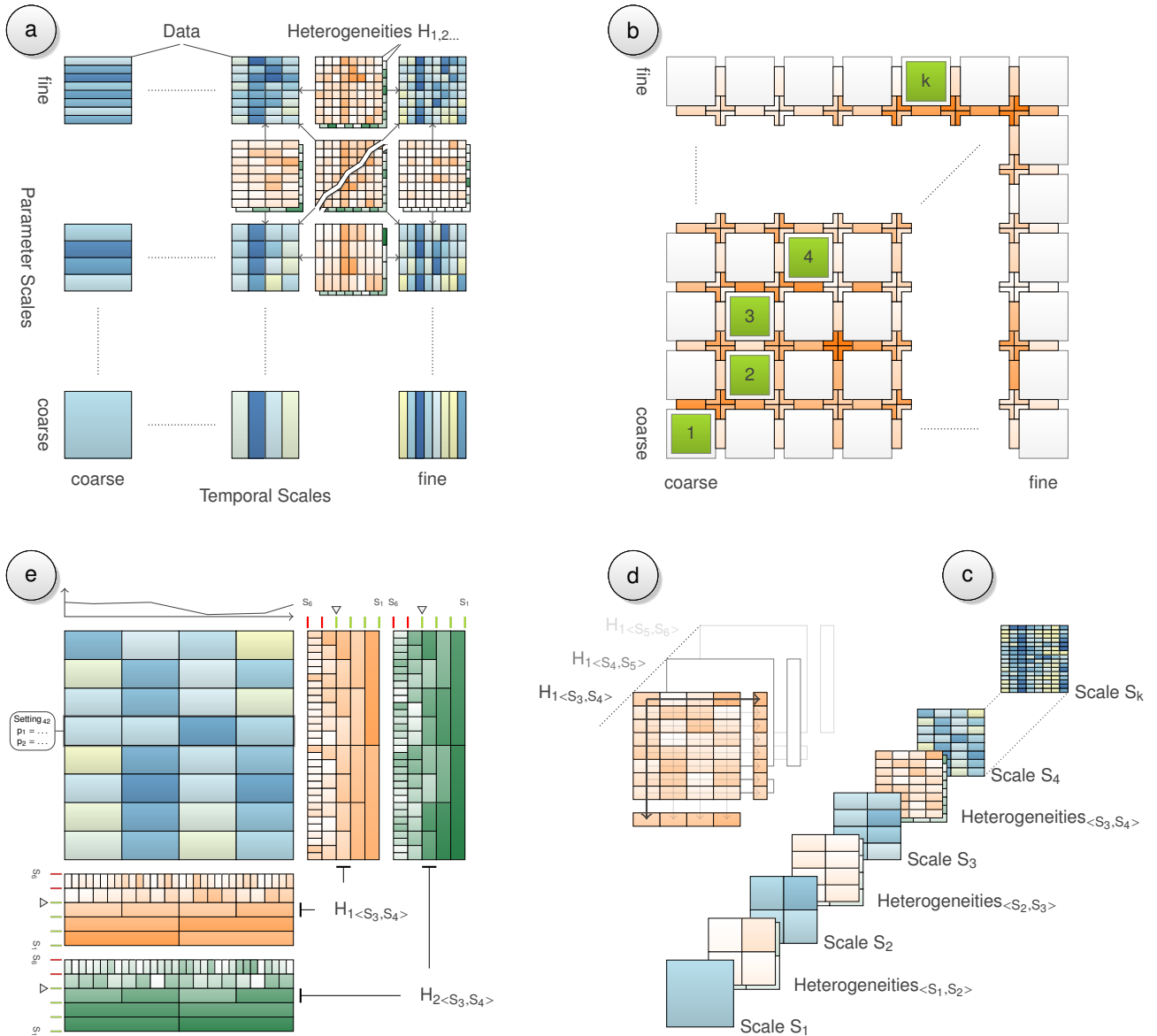


Figure 2: The main conceptual components of our approach. Different heterogeneities $H_{1,\dots,n}$ between neighboring scales are calculated (a). To determine an exploration path through the scale combinations, one of the computed heterogeneities is selected, aggregated, and displayed between neighboring cells of the *scale selection matrix* (b). Selecting a path in the matrix determines a sequence of scales combined with the corresponding interscale heterogeneities (c). That heterogeneity information $H_{1,\dots,n}$ is projected row- and column-wise (d) to depict it in heterogeneity bands along the main view of the currently selected scale (e). These bands, as well as an interactively displayable line chart, overplotting indicators, and further details on demand support the exploration of the multiple scales.

reference value measured in the lab. The result can be seen in Figure 3(b), showing areas of the parameter space that lead to positive deviations at the bottom and areas that lead to values lesser than the measured reference at the top. Outcomes that match the reference value and show no deviation at all are located close to the lower border colored entirely in yellow.

The analyst can then start to drill into the data along the sequence of scales he has chosen. As always the same number of pixels are available, a drill-down into a more fine-grained scale will automatically require to reduce the extent the analyst is investigating. To aid the analyst in making an informed choice on where to drill down (extent) and how far to drill down (grain), we compute and show measures of interestingness to indicate

possible exploration targets. As before with the scale selection matrix, we make use of interscale heterogeneities, which quantify how much data on a more fine-grained scale deviates from a more coarse-grained scale according to [2]. As there is no screen space to show all 2-dimensional heterogeneities in full resolution, we accumulate the calculated heterogeneities between all consecutive scales vertically and horizontally (see Figure 2(d)). To provide guidance according to multiple metrics at once (e.g., the absolute difference or gradient), this procedure is repeated for each of them. The results are color-coded heterogeneity bands on both sides of the main view that shows the stacked trajectories (see Figure 2(e)). For a clear differentiation, each metric is represented in its own color scale, mapping

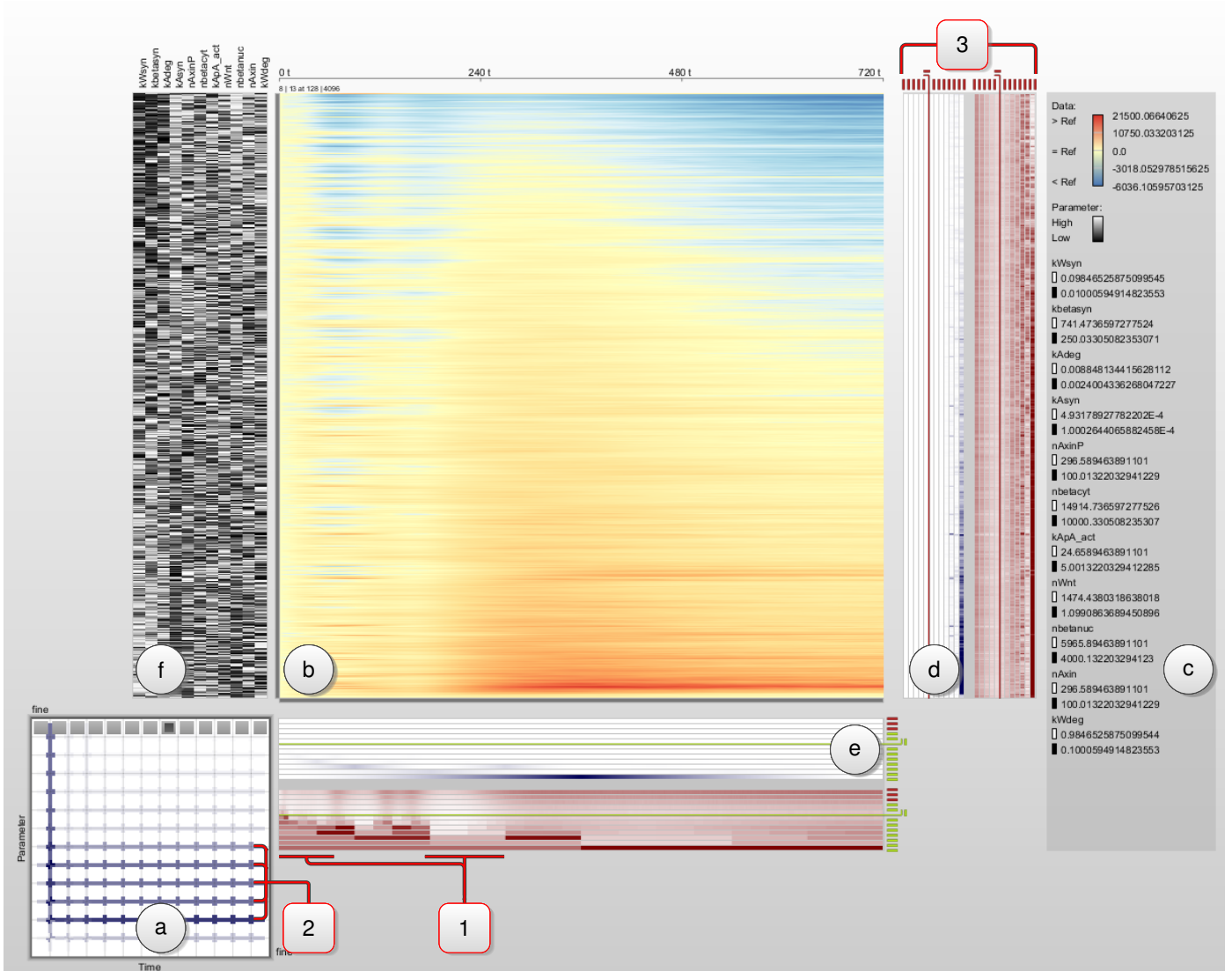


Figure 3: Instance of our Visual Analytics setup. The setup consists of several interlinked components: (a) The matrix of scales for selecting a drill-down order. The colored bands between neighboring matrix cells indicate the interscale heterogeneity between them and thus hint at promising candidates to zoom into. The currently chosen order is indicated by filling the corresponding cells gray. In the shown case, the drill-down order is defined only across temporal scales (from left to right), whereas the scale of the parameters remains at the finest resolution (top row). (b) The main view that depicts the vertically aligned simulation trajectories. The actual numerical value along the trajectory is color-coded according to the bivariate color palette shown in (c). The view can be read horizontally from left to right tracing the simulation trajectories over time for the parameters associated with each row. Yet, it can also be looked at vertically to compare the different simulation states across the parameter space at the simulation time that is associated with each column. The currently shown scale in the main view is indicated in the matrix (a) with a dark gray inset in the corresponding matrix cell. (d) The vertical heterogeneity view indicates with colored bands where in the parameter space a drill-down or roll-up would uncover deviating behavior from the currently shown scale of the simulation outcome. Each band (red/blue) stands for a different heterogeneity metric. Additional metrics can be added, so that a user can base his decision on multiple metrics at the same time. (e) The horizontal heterogeneity view indicates the same for the temporal scales. Both heterogeneity views make use of the algorithm presented in [2] to compute and visualize these bands. (f) The parameter space view encodes each parameter as a band of black-and-white stripes. The brighter they are, the higher is their parameter value in a given parameter combination. The highlighted data aspects (1) through (3) are detailed further in Sections 3.4 & 4.3.

a zero heterogeneity to white and high heterogeneity values to saturated colors.

In our current analysis setup, we are interested in parameter settings, whose simulation trajectories represent local outliers concerning numerical values as well as the trajectory's trend. Thus, we use a point-wise absolute Euclidian distance and a 1D-gradient as heterogeneity metrics. The 1D-metric captures trend differences only in the horizontal direction as we are

more interested in changes in the temporal direction at different scales, than trends along neighboring parameter combinations. Moreover, the latter vertical trends depend on the applied sorting and thus are of less meaningful.

The purpose of all these heterogeneity metrics is to point the user towards more fine-grained data that diverges from the currently shown coarse-grained overview, so that he can target his drill-down to these regions (e.g., between the intervals

of Figure 3(1)). For this purpose, we provide common navigation techniques like fisheye distortions, pan and zoom, drill-down and roll-up along the selected traversal order of scales, as well as direct scale selection in the selection matrix. During the interactive exploration process, indicators next to the heterogeneity bands (Figure 1(e)) continuously inform the user, if the current view is affected by overplotting to advise a further reduction of the extent (zoom, cp. [2]). As all views are tightly interconnected, such actions update all other views immediately. To provide conclusive information for the *Decision Making*-stage, we further show details on demand for selected parameter settings. These give the precise numerical parameter values for a parameter combination of interest and depict its simulation trajectory along the reference data in a more conventional time-value-plot above the main view (Figure 1(e)).

Equipped with such an interactive exploratory technique to investigate the simulation outcome across the parameter space on several scales, the analyst can make observations from the data that lead to decisions for future simulation runs.

3.4. Stage III: Decision Making

The drill-down can only be repeated as long as simulation results are available for the more fine-grained levels. If at some point during the exploration of the data the user wants to drill down further, more simulations at finer temporal scales and a finer sampling of the parameter space have to be run for the region of interest. Informed by the visual exploration of the parameter space, the user can now decide on suitable refinements for such future simulation runs.

1. Refining the Input Parameter Space. After investigating how the current simulation results behave across the parameter space, the analyst may observe differing degrees of influence for different parameters. One parameter may not influence the simulation outcome much and can either be sampled more coarsely or it can even be left out completely in the future. Whereas another parameter may exhibit interesting behavior that could not be fully explored at its current sampling rate and thus demands a more fine-grained subsampling in future simulation runs. If both cases are observed in conjunction, the analyst can shift computing time from one parameter to the other in order to better balance future simulation experiments.

For example, in the parameter space view in Figure 3(f), parameter kA_{syn} has predominantly high values in the regions that exhibit a positive deviation from the measured reference value, whereas the parameters kW_{syn} , $k\beta_{syn}$, and kA_{deg} appear to have the opposite effect. Hence, they seem to play a major role in how the simulation trajectories behave and would be worthwhile to be subsampled further in future simulation runs to confirm this observation and to investigate it in more detail. Yet other parameters do not exhibit such a clear pattern with their values being almost randomly distributed. As apparently these other parameters do not have a significant effect on the simulation outcome, they can either be neglected entirely in future simulation runs (i.e., set to some constant value) or they can be sampled more coarsely. This way, the analyst can shift computing time from parameters that show no effect on the

simulation outcome to others that do in order to better balance future simulation experiments towards parameters of interest.

2. Refining the Recorded Simulation Trajectory. Likewise, the analyst could observe time intervals in the simulation trajectories, in which not much behavior of interest occurs, while other intervals show such behavior and would thus be interesting to look at more closely. Hence, intervals of the simulation trajectory that do not exhibit much dynamics can either be recorded more coarsely or omitted altogether in the recording.

In Figure 3(1) for example, no changes appear before 70min, as well as between 150min and 300min of the simulated time span. Thus, the resources can be reallocated to run region-specific simulations on more or less fine-grained scales, according to the dynamic behavior that was observed.

Having gained this knowledge about which of the parameters and which intervals of the simulation trajectories are of high interest, any subsequent simulation can be targeted directly at them, while leaving out those of lesser interest. Thus, another analysis cycle starting with Stage I would be carried out, but this time taking the gained knowledge into account and shifting the initially even subdivision of the parameter space and evenly recorded simulation trajectories so that they better capture the identified areas of high interest. The following section illustrates with a use case example, how such an iterative analysis is performed in order to gain insight in a real-world scenario.

4. Use Case: Exploring Temporal and Parameter Scales

This section demonstrates our approach with an example from the domain of systems biology. Based on a biological model, which is described in Section 4.1, simulation runs are executed for a few thousand different parameter combinations in order to get a good grasp of the model's overall behavior. From these simulation runs, the concentration of a key molecule is recorded and matched against experimental lab data.

There exist several different approaches to calculate the parameter configuration that fits available lab data best [32, 33]. For the computational model that we use for simulation, such approaches have been used in the past to compute and provide parameter values (initial molecule counts and reaction rate constants) that lead to plausible trajectories [34]. However, for developing new hypotheses and regarding potential model extensions, an automated optimization method does not suffice and a detailed exploratory analysis of alternative parameter configurations in regard to deviations from the measured reference data is desired. As the following sections illustrate, with our Visual Analytics setup, the parameter space can be explored accordingly and lead to parameter configurations that give concrete guidance on where to refine future simulation runs to yield further insights.

4.1. Biological Background

The main focus of the study lies on neuronal differentiation in human stem cells, in particular *progenitor cells*. Progenitor cells are already more specialized than stem cells, but they

still have the ability to transform (differentiate) into different cell types. For example, neuronal progenitor cells can still differentiate into various neuronal cell types, such as neurons or glial cells. Interestingly the fraction of neurons evolving from a set of progenitor cells increases significantly, when the cells are treated with *Wnt* molecules. This process might play an essential role in the treatment of neurodegenerative diseases, like Alzheimer’s disease. Therefore it is important to know the detailed cellular mechanisms that take place when stimulating progenitor cells with *Wnt* molecules (*Wnt signaling*). Additionally, *Wnt* signaling is also connected to a number of human cancers and developmental disorders.

To study *Wnt* signaling in neuronal progenitor cells, a computational stochastic model has been established, based on experimental data. The model is defined in the modeling formalism ML-Rules [35]. It comprises 3 hierarchical levels (cell, nucleus, protein), 6 (attributed) molecules, and 13 reaction rate constants and is based upon [34]. The model describes major components of the canonical *Wnt* signaling pathway and their intracellular dynamics.

The key component of *Wnt* signaling is a protein called β -catenin. It is mainly located in the intracellular fluid (cytosol). When progenitor cells are stimulated by extracellular *Wnt* molecules, a cascade of chemical reactions is triggered inside the cytosol. As a result, β -catenin is transported into the cell nucleus, where it activates specific genes. The relocation of β -catenin into the nucleus is the main indicator for the pathway activation. However, the detailed mechanism by which *Wnt* mediates β -catenin relocation is still poorly understood and subject of intensive study [36].

4.2. Stage I: Computational Analysis

The data set is produced by simulating the previously described model using a τ -leaping implementation of the stochastic simulation algorithm [37].

Sampling the Parameter Space. For the sampling of the parameter space, we used a subset of $n = 11$ parameters that are supposed to have the strongest impact on the signaling pathway. Typically these parameters are “bottlenecks” in the pathway and characterized by a low molecule count or a slow reaction rate. Consequently, already small changes in these parameter values may have a major impact on the simulation outcome. Sampling the 11 parameters, each with 10 values, results in $c = 10^{11}$ possible parametrizations when doing a full factorial parameter scan. Due to time constraints, we chose to perform only $c = 4096$ configurations, performing $r = 50$ replications for each configuration, which results in a total of $s = 204800$ simulation runs. The number of possible samples per parameter would thus be $2.13 \rightarrow k \sim 2$ in this setting, basically meaning that only the *min* and *max* of each parameter range would be used as sample points. Thus, we chose the Latin Hypercube Sampling combined with permutations: In our setting each parameter has 10 samples, which in return would result in only 10 configurations generated by a classical LHS approach. To achieve 4096 configurations and therefore a better coverage

of the parameter space, we further generated unique configurations by randomly permuting the generated LHS design.

Running the Simulation. Having generated 4096 configurations in the previous step, the actual simulations for each of those configurations were performed. All the simulations were calculated on a single multi-core processor machine, distributing the calculation across available processors. For the particular case of stochastic simulation, we distributed the computation by repetitions [26] and parameter combinations [27].

Recording the Simulation Trajectory. Each replication of each configuration is simulated until the simulation time reached 720min (12h), which is the approximate time span to determine the cell’s fate, i.e., its prospective cell type (“cell fate commitment”). To aggregate replications, it is important to sample each replication at the same time points. Therefore, we sampled the molecule count (trajectories) using a fixed time interval of 0.17578125 simulation time, yielding 4096 sample points. For each of these sample points, we recorded the critical molecule count of the β -catenin in the nucleus ($n\beta nuc$) and averaged them on a per configuration basis, resulting in one averaged simulation trajectory per parameter configuration.

4.3. Stage II: Visual Exploration

This stage deals with the visualization and exploration of the recorded simulation trajectories for β -catenin. Note that we do not visualize β -catenin’s molecule count directly, but as a difference value as compared to the experimental data from the lab. As the visual exploration itself is an interactive process, we address the steps of this stage twice to make their interplay in an iterative refinement more explicit.

1st Iteration: Delineating the Parameter Space. The visual exploration started with delineating the parameter combinations according to the trajectory data (Figure 3). The used average squared distance results in a smooth transition between the different outcome trajectories that allows for the detection of correlations concerning the parameters. As observed in Section 3.4, the deviation towards the reference data seems to be correlated to the following four reaction rate constants: kA_{syn} , kW_{syn} , $k\beta_{syn}$, and kA_{deg} .

1st Iteration: Choosing a Sequence of Scales. Delineating the parameter combinations according to the trajectory data results in an alignment of trajectories, where the heterogeneities between consecutive trajectories are rather small as a result of the sorting. Thus, a coarsening of the parameter space (vertical axis) and the accordingly calculated heterogeneities mainly capture sorting artifacts as well as the continuously increasing distances between trajectories due to coarser sampling. This effect becomes visible in the scale selection matrix, by continuously increasing heterogeneities towards the coarsest parameter scale (Figure 3(2)). Consequently, an investigation of different scales in parameter space will yield no insight on the parameters themselves.

For this reason, we first chose to investigate the temporal scales. In order not to miss outstanding simulation trajectories concerning the temporal behavior, we first chose the finest available parameter space sampling (Figure 3(a)) although it is affected by overplotting. The according overplotting indicators (Figure 3(3)) remind us to reduce the extent for a faithful data representation. In this case, any recorded heterogeneity is only due to a reduced temporal resolution providing insight on temporal scales for future simulation trajectory recordings. The respective sequence of scales – the top row of the scale selection matrix – is selected interactively.

1st Iteration: Targeted Drill-down along the Chosen Scales. Figure 3(d) & (e) show the calculated heterogeneities absolute Euclidean distance (dark blue) and 1D-gradient distance (dark red). Our first observation (later referred to as *O1*) is, that absolute numerical differences between consecutive temporal scales are close to zero, after the sixth finest temporal scale – becoming visible by white heterogeneity bands in Figure 3(d) & (e). The same holds true for the 1D-gradient metric. But we also see, that gradient changes slightly increase again towards the finest temporal resolution. In combination with the low Euclidean distances in these areas of increase, this suggests a slight shifting of the curves along the temporal axis, which is only due to a more precise sampling at highest temporal resolutions. As a second observation (*O2*), we can also determine temporal regions where the data is constant and no changes seem to appear at all – the white regions inside the dark red heterogeneity bands (Figure 3(1)).

Drilling down to a temporal scale of high resolution, zooming into the specific regions (e.g., into Figure 3(1)) as well as the interactively displayable line chart (cp. Figure 4(7)) confirm both observations, the very slight shifting of curves and the practical absence of changes before 70min and between 150min and 300min. As a first result, *O1* suggests to limit further analysis investigations to the sixth finest temporal scale. Moreover, both observations are supposed to affect future simulation runs as they recommend to reconfigure the recording resolution of the simulation trajectories.

2nd Iteration: Delineating the Parameter Space. According to the correlation hypothesis, the sorting is altered in the second exploration run. Since *kWsyn*, *kbetasyn*, and *kAdeg* seem to influence the deviation in a combined manner, we focus on these parameters and sort the data according to them (Figure 4). By doing so, the heterogeneity bands should point to outliers in the simulation trajectories that deviate from surrounding similar parameter settings. Again the correlation between those parameters and the difference to the measured reference values becomes visible, whereas the remaining parameters are masked out.

2nd Iteration: Choosing a Sequence of Scales. As we already determined the essential temporal resolution, we now focus on the different parameter space sampling scales. For this reason, we interactively select all entries of the scale selection matrix,

along the 6th column from the right to match the aforementioned temporal scale from *O1* below which no more changes can be observed.

2nd Iteration: Targeted Drill-down along the Chosen Scales. Figure 4 shows the visualization during the second iteration. In this case, all captured heterogeneities are due to the coarsening of the parameter space sampling and points to changes between consecutive parameter settings at the different scales. At a first glance, the dark red heterogeneity bands (Figure 4(1)) reveal the uniform sampling of the parameter space, as well as a uniform distribution of gradient changes up to the highest resolutions. In contrast, the Euclidean distance metric hints at a non-uniform distribution of heterogeneities, as most of them are located at the top (Figure 4(2)). Using an adapted local color scale supports this observation (*O3*) of heterogeneities correlating with the chosen three parameters in particular at the finer scales (Figure 4(3)). Figure 4(4) again shows the uneven distribution of changes along the temporal axis (see *O2*).

Beside these first impressions, the slightly darker blue bars within the heterogeneity bands (Figure 4(5)) indicate the existence of outliers due to bigger differences between subsequent simulation outcomes although their three parameters are deemed to be nearly equal. An interactive grain refinement (drill-down) and extent reduction (zoom) to the scales and intervals where those bars appear, finally confirm this observation (Figure 4(6)). To support the stage of decision making, we interactively select these parameter settings that yield outliers. This brings up their details on demand by showing their concrete numerical parameter values as well as their behavior over time in the time-value-plot to ease their comparison with neighboring simulation trajectories (Figure 4(7)). As a result, both observations *O3* and *O4* advise a reconfiguration of the parameter sampling towards a non-uniform sampling focusing the found high heterogeneity intervals.

4.4. Stage III: Decision Making

Basically, the computed 4096 configurations already achieve a good coverage of the parameter space. We observed approximately the same fraction for higher (red) and lower (blue) and some equivalent values for nuclear β -catenin concentration when comparing the simulation trajectories to the experimental data (Figure 3(b)). Based on our new visual exploration approach, we were able to further distinguish between sensitive and non-sensitive parameters, identifying potential regions for subsequent refinements – with respect to parameter space and the temporal scale – and to find remarkable parameter configurations that motivate further fine-grained evaluation.

Refining the Input Parameter Space. The automated sensitivity analysis performed by Mazemondet et al. [34] determined *n β nuc*, *nAxin*, and *kbetasyn* as the most significant parameters. Even though *kbetasyn* is the only parameter which exactly matches the correlated parameters identified by our visual approach (*kAsyn*, *kWsyn*, *kbetasyn*, and *kAdeg*), the results are still plausible, when considering the biological context. The parameter *n β nuc* determines the number of β -catenin molecules

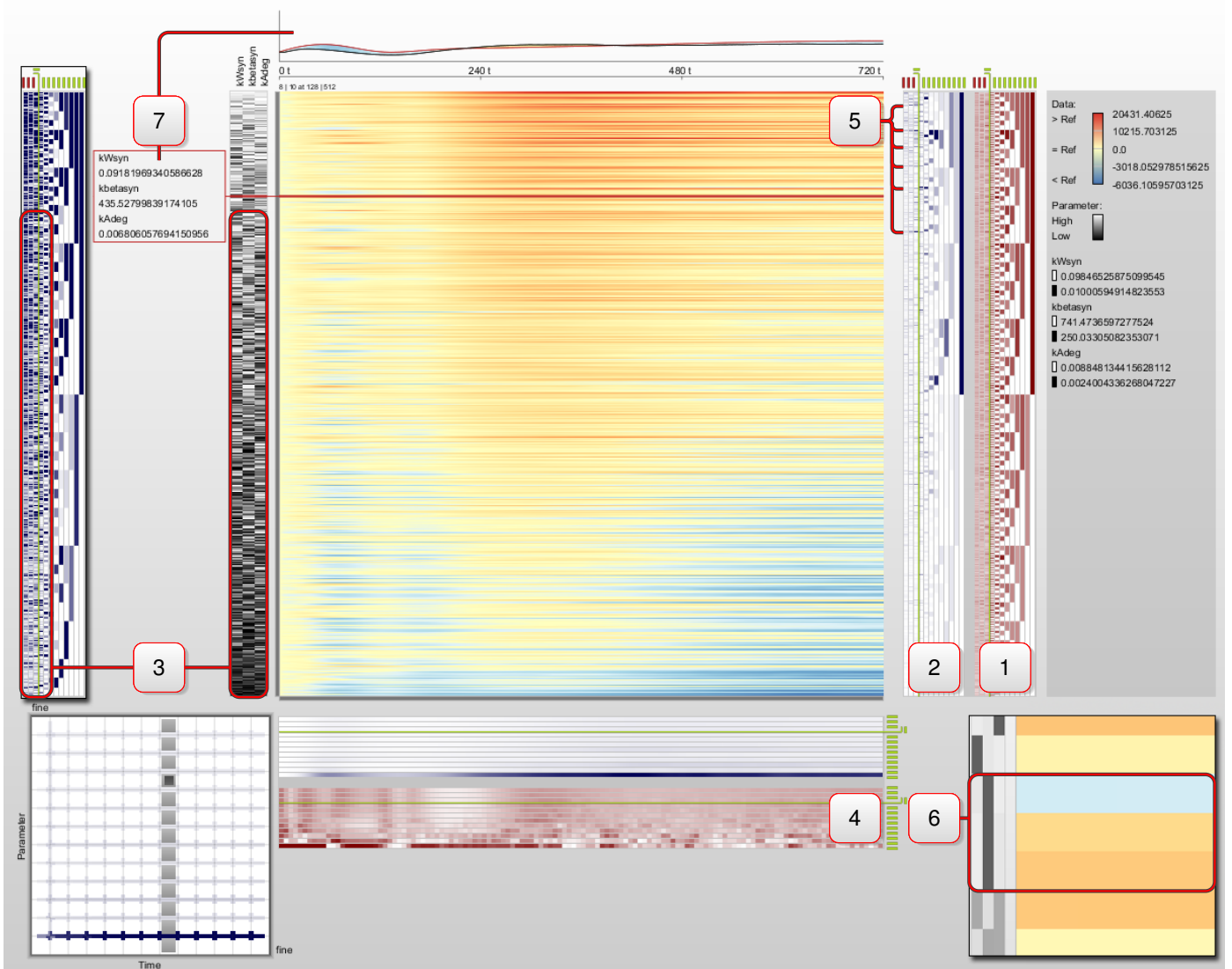


Figure 4: Result of sorting the data according to a reduced set of parameters to explore the parameter sampling scales at the sixth finest temporal resolution. Details on the highlighted data aspects (1) through (7) are given in Section 4.3.

and is regulated by synthesis and degradation as well as the transport between nucleus and cytosol. Since transport parameter values have been kept constant, the parameter with highest impact on the amount of $n\beta nuc$ is $k\beta etasyn$. The same applies to $nAxin$, which defines the number of another important molecule of the pathway – Axin. The count $nAxin$ directly depends on the parameters $kAsyn$ and $kAdeg$, as they define the synthesis and degeneration rates of Axin. Consequently, $kAsyn$ and $kAdeg$ directly influence the number of Axin molecules in the system. Therefore the automated analysis of Mazemondet et al. coincides very well with our insights gained from visual exploration.

However, the parameter space view allows not only to determine parameter sensitivity, but also parameters having a promoting and repressing effect. As seen in Figure 3, $kWsyn$, $k\beta etasyn$, and $kAdeg$ obviously have promoting effects, i.e., high parameter values lead to higher simulated values as com-

pared to the reference. On the other hand, $kAsyn$ has the opposite effect, which is exactly the case in Wnt signaling. The choice of the three parameters $kWsyn$, $k\beta etasyn$, and $kAdeg$ is therefore not only consistent with the visual cues but also matches the biological background.

Based on the sorting of the three most significant parameters, regions of higher heterogeneity could be revealed. In these regions, neighboring trajectories vary more, even though the parameter configurations are generally similar. In our model, the heterogeneity is particularly high when all three parameters have high values and vice versa (O3). Accordingly, the parameter space is not sufficiently covered here and thus prompts for a more fine-grained sampling or a non-uniform sampling to keep the simulation efforts constant.

On the other hand, similar parameter settings lead to different/opposite simulation outcomes in certain cases (O4). These parameter configurations need further evaluation as well, as

they imply outliers that are not associated with the main set of significant parameters. A reconfiguration of sampling can be useful to determine, whether this effect is due to the selected parameters themselves or to the remaining (masked) parameters. If finer parameter samplings in these parameter intervals result in a smooth transition between neighboring configurations without outliers, the parameter sampling in that interval was too coarse. The masked parameters may be the cause, if the outliers remain. Therefore such areas are also prominent candidates for further fine-grained parameter samplings.

Refining the Recorded Simulation Trajectories. The τ -leaping implementation of the stochastic simulation algorithm, that has been used to carry out the simulations, naturally employs the best temporal resolution in terms of accuracy and speed [37]. However, the (horizontal) heterogeneity also sheds light on the dynamics of the signaling pathway under study. The experimental data in [34] indicates a biphasic signal after stimulating the cells. That means the cell response is characterized by a first, short (transient) activation, which is followed by a continuous, long-term activity. The activity of the pathway is indicated by high $n\beta_{nuc}$ values. Therefore the very low heterogeneity values observed between 150min and 300min correspond to the time lag (delay) between the first and the second activation of the pathway. Thereby the visualization remarkably exposes that this delay is parameter independent (*O2*), which allows for a coarser temporal recording in this case.

Even more significantly, we make use of observation *O1*. It exposes, that 128 samples at the temporal scale suffice to reflect the behavior of the simulation trajectories neglecting minor temporal shifting. This results in a temporal resolution of approximated 5.5min which in our scenario is fine enough to capture changes in a cell's Wnt pathway. Consequently, data recording of the simulation trajectories can be dramatically reduced to 3% of the former highest resolution without losing any crucial information.

5. Discussion and User Feedback

By following our approach in such an interactive iterative manner, as it was described in the above use case, a user gains an understanding of the data on all levels – from a global overview via interesting local regions to seeking individual points in the data. The overview is supported by the initial coarse-grained, color-coded trajectories per configuration shown in the main view. Given a suitable delineation of the parameter space, this view alone conveys already a “big picture” that gives insights in how the different courses that the trajectories take are distributed across the parameter space – which kinds of trajectories occur often and which are rare. After having investigated the overall extent of the data at a coarse scale, the heterogeneity displays provide the means to look into regions of interest. They help to target a drill-down towards interesting regions in the parameter space and in time by indicating the extent of such regions (i.e., where to drill down) and the grain of such regions (i.e., how far to drill down). Drilling down further, the user would finally be presented with individual configurations

and be able to single them out. If one is looking for individual parameter configurations, for example, as it is the case in parameter fitting, this is the level that would allow the user to identify a result. While applying an automatic method, as it is the state-of-the-art in this field, would most likely yield such a result equally well, this result would be an isolated point in the parameter space generated by an algorithm that remains largely a “black box” to the user. With our interactive visual approach, the user knows exactly how the result came about, why he chose which delineation and which scales on the way to this result, and how this result relates to the whole of the parameter space. This yields a result that makes sense, that the user trusts in, and that he can argue for. This result can of course always be verified with an automated method if needed.

This added benefit over purely automatic analysis approaches was also reflected in the feedback that we gathered in evaluation sessions with our application partners from the Systems Biology domain. At first our approach with all its required interactivity did not appeal to the domain experts as much as simply running an algorithmic analysis. Yet, once they understood that for instance different delineations will result in entirely different views of the parameter space, each with its own semantics and insights to be found, they were very positive about it, as this clearly surpasses what they can learn from their data with automatic approaches. The need to explore visualizations and to interact with them is simply the price that has to be paid for gaining this additional understanding instead of an isolated, singular result. Thus, it may not be surprising that soon after this realization, our users actually demanded more interactivity and more visualizations to be added. For example, as they are very accustomed to time-value-plots, they suggested to include the possibility to select an individual color-coded trajectory in the main view and to show its behavior in detail in a linked time-value-plot. As a result of this feedback, we have added this particular feature, as it was shown in Figures 1(e) & 4(7).

Another suggestion that will be highly useful and that is conceptually well within reach is a direct integration of our approach as a steering interface into their simulation software. This way, new simulation experiments could be triggered directly from within our visualization – a certainly tempting prospect for future work that shows that our approach goes actually further than a mere analysis of simulation results. The users suggested furthermore to show data different from the simulation trajectories, which might be useful in other scenarios – e.g., the explicit simulation processing times or the uncertainty inherent in the used simulation method. Such an integration, for example of uncertainty visualization, can be realized in a straightforward manner by simply displaying each trajectory's accuracy in the main view, as it was already implemented in [10].

As our approach was specifically constructed to scale to a large number of trajectories and time points by using methods from multi-scale visualization, its limitations lie not so much on the side of the visual representation, but rather on the time spent to compute the steps that lead up to it. Despite the fact that we only sample the parameter space at a coarse scale, its simulation and subsequent delineation nevertheless take a con-

siderable amount of time. These long-running computing times stand in contrast to the high amount of interactivity that we aim to provide. For example, simply switching to a different delineation will trigger the appropriate sorting algorithms and possibly lock the Visual Analytics setup for several hours, depending on the number of parameter combinations to be sorted and the number of time points of the trajectories according to which to sort them. As this challenge is inherent to the problem we deal with, we only have the options of either reducing the computational load by reducing the number of samples, or of shifting it to a pre-computation. We circumvent this problem by utilizing the latter approach and pre-computing all the different delineations to have them stored and readily available as soon as the user wants to switch. While this adds to the computing time before the visualization can be shown, it keeps the visualization interactive once it is ready.

6. Conclusion

With the presented general approach for bringing together parameter space visualization and multi-scale visualization, we have taken a first step towards giving an overview of large parameter spaces and the resulting simulation trajectories at the same time. Passing through the 3-stage approach allows the user to make well-founded decisions that take into account the entirety of the sampled parameter combinations and simulation results with a confidence that is directly related to the granularity of the sampling. The user can increase his confidence in the simulation result by sampling regions showing unclear and ambiguous behavior on a more fine-grained scale in future simulation runs.

Further analytical means could be employed to additionally strengthen the user's confidence in his decisions. On the one hand, if a parameter appears to have no influence on the simulation result, this could be confirmed analytically by running a sensitivity analysis for this particular parameter, before deciding whether to eliminate it from future simulations. On the other hand, if a parameter seems to have a high influence on the outcome, a parameter estimation could be run to narrow down the interval in which the sought after parameter value must lie according to the measurements from the lab. This would cut down on the extent of this parameter's scale and thus free resources to increase its grain in future scans of this parameter's value range. While being highly desirable for performing such an *informed optimization*, such a tighter integration with confirmatory methods remains a subject for future work in this area.

Acknowledgements

The authors wish to thank the domain experts who participated in the feedback sessions, as well as the anonymous reviewers for their helpful input. The authors gratefully acknowledge funding from the German Research Foundation (DFG) as well as the implementation support by Martin Röhlig.

References

- [1] Ahl V, Allen T. Hierarchy Theory: A Vision, Vocabulary, and Epistemology. Columbia University Press; 1996.
- [2] Luboschik M, Maus C, Schulz HJ, Schumann H, Uhrmacher AM. Heterogeneity-based guidance for exploring multiscale data in systems biology. In: Proceedings of IEEE BioVis'12. IEEE Press; 2012, p. 33–40. doi:10.1109/BioVis.2012.6378590.
- [3] Tweedie L, Spence B, Dawkes H, Su H. The Influence Explorer (video) – a tool for design. In: Proceedings of CHI'96. ACM Press; 1996, p. 390–1. doi:10.1145/257089.257390.
- [4] Matković K, Gračanin D, Jelović M, Ammer A, Lež A, Hauser H. Interactive visual analysis of multiple simulation runs using the simulation model view: Understanding and tuning of an electronic unit injector. IEEE Transactions on Visualization and Computer Graphics 2010;16(6):1449–57. doi:10.1109/TVCG.2010.171.
- [5] Ma KL. Image graphs - a novel approach to visual data exploration. In: Proceedings of IEEE Vis'99. IEEE Press; 1999, p. 81–8. doi:10.1109/VISUAL.1999.809871.
- [6] Jankun-Kelly T, Ma KL. A spreadsheet interface for visualization exploration. In: Proceedings of IEEE Vis'00. IEEE Press; 2000, p. 69–76. doi:10.1109/VISUAL.2000.885678.
- [7] Marks J, Andalman B, Beardsley PA, Freeman W, Gibson S, Hodgins J, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In: Proceedings of SIGGRAPH'97. ACM Press; 1997, p. 389–400. doi:10.1145/258734.258887.
- [8] Bruckner S, Möller T. Result-driven exploration of simulation parameter spaces for visual effects design. IEEE Transactions on Visualization and Computer Graphics 2010;16(6):1468–76. doi:10.1109/TVCG.2010.190.
- [9] Brochu E, Brochu T, de Freitas N. A Bayesian interactive optimization approach to procedural animation design. In: Proceedings of SCA'10. Eurographics Association; 2010, p. 103–12. doi:10.2312/SCA/SCA10/103-112.
- [10] Luboschik M, Rybacki S, Ewald R, Schwarze B, Schumann H, Uhrmacher AM. Interactive visual exploration of simulator accuracy: A case study for stochastic simulation algorithms. In: Proceedings of WSC'12. IEEE Press; 2012, p. 419:1–12. doi:10.1109/WSC.2012.6465190.
- [11] Booshehrian M, Möller T, Peterman RM, Munzner T. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. Computer Graphics Forum 2012;31(3):1235–44. doi:10.1111/j.1467-8659.2012.03116.x.
- [12] Pretorius AJ, Bray MA, Carpenter AE, Ruddle RA. Visualization of parameter space for image analysis. IEEE Transactions on Visualization and Computer Graphics 2011;17(12):2402–11. doi:10.1109/TVCG.2011.253.
- [13] Wijk JJ, Overveld CWAM. Preset based interaction with high dimensional parameter spaces. In: Post FH, Nielson GM, Bonneau GP, editors. Data Visualization: The State of the Art. Springer; 2003, p. 391–406. doi:10.1007/978-1-4615-1177-9_27.
- [14] Berger W, Piringer H, Filzmoser P, Gröller E. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. Computer Graphics Forum 2011;30(3):911–20. doi:10.1111/j.1467-8659.2011.01940.x.
- [15] Torsney-Weir T, Saad A, Möller T, Hege HC, Weber B, Verbavatz JM. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. IEEE Transactions on Visualization and Computer Graphics 2011;17(10):1892–901. doi:10.1109/TVCG.2011.248.
- [16] Elmqvist N, Fekete JD. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Transactions on Visualization and Computer Graphics 2010;16(3):439–54. doi:10.1109/TVCG.2009.84.
- [17] Kincaid R. SignalLens: Focus+context applied to electronic time series. IEEE Transactions on Visualization and Computer Graphics 2010;16(6):900–7. doi:10.1109/TVCG.2010.193.
- [18] Zhao J, Chevalier F, Pietriga E, Balakrishnan R. Exploratory analysis of time-series with ChronoLenses. IEEE Transactions on Visualization and Computer Graphics 2011;17(12):2422–31. doi:10.1109/TVCG.2011.195.
- [19] Javed W, Elmqvist N. Stack zooming for multi-focus interaction in time-series data visualization. In: Proceedings of IEEE PacificVis. IEEE Press; 2010, p. 33–40. doi:10.1109/PACIFICVIS.2010.5429613.

- [20] Hao M, Dayal U, Keim D, Schreck T. Multi-resolution techniques for visual exploration of large time-series data. In: Proceedings of EuroVis'07. Eurographics Association; 2007, p. 27–34. doi: 10.2312/VisSym/EuroVis07/027-034.
- [21] Novotny M, Hauser H. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics* 2006;12(5):893–900. doi:10.1109/TVCG.2006.170.
- [22] Hao MC, Janetzko H, Mittelstädt S, Hill W, Dayal U, Keim DA, et al. A visual analytics approach for peak-preserving prediction of large seasonal time series. *Computer Graphics Forum* 2011;30(3):691–700. doi: 10.1111/j.1467-8659.2011.01918.x.
- [23] Woodring J, Shen HW. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Transactions on Visualization and Computer Graphics* 2009;15(1):123–37. doi: 10.1109/TVCG.2008.69.
- [24] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 1979;21(2):239–45. doi: 10.1080/00401706.2000.10485979.
- [25] Hernandez AS, Lucas TW, Sanchez PJ. Selecting random latin hypercube dimensions and designs through estimation of maximum absolute pairwise correlation. In: Proceedings of WSC'12. IEEE Press; 2012, p. 25:1–12. doi:10.1109/WSC.2012.6465112.
- [26] Leye S, Himmelspach J, Jeschke M, Ewald R, Uhrmacher AM. A grid-inspired mechanism for coarse-grained experiment execution. In: Proceedings of DS-RT'08. IEEE Press; 2008, p. 7–16. doi:10.1109/DS-RT.2008.33.
- [27] Himmelspach J, Ewald R, Leye S, Uhrmacher AM. Enhancing the scalability of simulations by embracing multiple levels of parallelization. In: Proceedings of HiBi'10. IEEE Press; 2010, p. 57–66. doi: 10.1109/PDMC-HiBi.2010.17.
- [28] Perumalla KS, Protopopescu VA. Reversible simulations of elastic collisions. *ACM Transactions on Modeling and Computer Simulation* 2013;23(2):12:1–25. doi:10.1145/2457459.2457461.
- [29] Asmussen S, Glynn PW. *Stochastic simulation: Algorithms and analysis*. Springer; 2007. doi:10.1007/978-0-387-69033-9.
- [30] Garland M, Heckbert PS. Surface simplification using quadric error metrics. In: Proceedings of SIGGRAPH'97. ACM Press; 1997, p. 209–16. doi:10.1145/258734.258849.
- [31] Harrower MA, Brewer CA. ColorBrewer.org: An online tool for selecting color schemes for maps. *The Cartographic Journal* 2003;40(1):27–37. doi:10.1179/000870403235002042.
- [32] Glover F. Tabu search – Part I. *ORSA Journal on Computing* 1989;1(3):190–206. doi:10.1287/ijoc.1.3.190.
- [33] Glover F. Tabu search – Part II. *ORSA Journal on Computing* 1990;2(1):4–32. doi:10.1287/ijoc.2.1.4.
- [34] Mazemondet O, John M, Leye S, Rolfs A, Uhrmacher AM. Elucidating the sources of beta-catenin dynamics in human neural progenitor cells. *PLoS ONE* 2012;7(8):e42792:1–12. doi:10.1371/journal.pone.0042792.
- [35] Maus C, Rybacki S, Uhrmacher AM. Rule-based multi-level modeling of cell biological systems. *BMC Systems Biology* 2011;5:166:1–20. doi: 10.1186/1752-0509-5-166.
- [36] Metcalfe C, Bienz M. Inhibition of GSK3 by Wnt signalling – two contrasting models. *Journal of Cell Science* 2011;124(21):3537–44. doi: 10.1242/jcs.091991.
- [37] Helms T, Luboschik M, Schumann H, Uhrmacher AM. An approximate execution of rule-based multi-level models. In: Proceedings of CMSB'13. 2013, To appear.