Enhancing Comparative Model Analysis using Persistent Homology

Bastian Rieck, *Student Member, IEEE*, and Heike Leitte, *Member, IEEE*

Abstract—Mathematical models are widely used to replicate natural phenomena. They represent the growing universe, as well as the spreading of diseases. By concentrating on the essentials of these systems, mathematical models are perfectly suited to study system behavior under altered conditions. Defining a model for a given system is a challenging task and is commonly an advancing process, where current models are updated and competing ones designed. Quality quantification is hence a central task in model development. In this paper, we focus on the comparative analysis of competing models. We integrate state-of-the-art techniques and propose a novel topology-based measure to quantify model quality. Our novel measure particularly concentrates on structural stability in parameter space. Additionally, we design a model landscape that communicates similarities among model candidates. Both methods are demonstrated and evaluated using an example from drug development.

Index Terms—Model analysis, comparison, persistent homology, quality measure

1 Introduction

Mathematical models commonly describe a large variety of systems from numerical simulations in engineering, over chemical processes in the life sciences, to business simulations in the social sciences. The goal of the models is to describe the essential features of the system to facilitate in-depth analysis. All models have in common that they take a set of input variables and, based on these, compute output variables. The task of the models is to predict the output as reliably as possible. As models are simplified versions of the real world and as measured data is inherently erroneous, predictions and real data hardly ever match perfectly.

Hence, several competing models are developed and choosing the best one is not always straightforward—there are both constraints in accuracy as well as computational complexity and/or ease of implementation. Model quality is commonly assessed using some sort of mean error. This is a highly-simplified piece of information that requires a lot of experimenting and user knowledge to accurately judge the quality of a model. Several visualization methods have been developed to facilitate quality validation—see Sedlmair et al. [17] for a concise overview. Among the earliest of such approaches is a method by Spence et al. [18], who designed visualizations for engineering design. Using scatterplot matrices and scatterplots of a predefined set of parameter values (such as tolerance specifications), they facilitated the exploration process. Mühlbacher and Piringer [14] proposed a framework for measuring the quality of regression models. For geoscientific simulations with few parameters, Unger et al. [20] developed a method for completely enumerating and exploring the parameter space via standard statistical graphics. In a similar vein, Bruckner and Möller [5] focus on exploring the parameter spaces of visual effects in 3D scenes. For known and fully-enumerated parameter spaces in the context of visual prototyping, Matkovic et al. [13] developed a visual steering approach using coordinated views of scatterplots. By contrast, Pretorius et al. [15] present an enumeration and exploration workflow for parameter spaces in image analysis. Among others, their system recalculates the resulting images for different parameter values in classification tasks. More generally, Bergner et al. [2] partition multivariate parameter spaces into regions with distinct output behaviour. This can be used to understand qualitative differences in model outputs. Rheingans and desJardins [16] focus on visualizing the predictive qualities of different models, using both visualizations of probability distributions and self-organizing maps. In contrast to our approach, their method is geared towards class prediction problems. Furthermore, they rely on data projections, which are often not well-defined in modelling tasks. Recently, research focused on general quality metrics for model visualization [3]. Our method falls into the category of measuring "complex patterns" within the data space, although our purpose is not the visualization of the data itself.

In this paper, we extend existing work in model validation. We present the *model landscape* (ML), a visualization that illustrates the quality of existing models. For this purpose, we compare the output of a model to the measured data and among each other. We also propose a novel performance measure based on the topological analysis of models that is more sensitive towards the predictions of models than both the existing measures RMSE and R^2.

2 Quality measures for regression analysis

In the following, we assume that we are given a data set containing n instances (measurements) with d attributes, such that each instance can be represented as a vector $x_i = (x_{i1}, \dots, x_{id}) \in \mathbb{R}^d$. Every instance x_i has a corresponding scalar value $s_i \in \mathbb{R}$. Regression analysis now involves deriving a functional relationship between the values of the d attributes and the set of values $S = \{s_1, \dots, s_n\}$. There are numerous methods (e.g. support vector machines) for this purpose and each one results in a different set of predicted values.

We call each set S of predicted values a *model* of the scalar function. Since different algorithms for finding these models have different complexities and behave differently, we need to judge their quality. To this end, the data is commonly partitioned into a larger *training data set* and a smaller *test data set*. The algorithm is then applied to the training set in order to obtain a model. The quality of the model is judged by calculating statistics on the test data set.

In the following, we will compare different quality measures on a simple 1-dimensional example. Fig. 1, left, depicts the model (red) and the measured function values (grey).

2.1 State of the art

The *root-mean-square error* (RMSE) and the *correlation coefficient* (R^2) are the two most common methods for judging the quality of a model. Given a model with n values $m_i \in \mathbb{R}$ and original values $s_i \in \mathbb{R}$ in the test data set, the RMSE is defined as

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} (m_i - s_i)^2 / n} \tag{1}$$

and aggregates the errors in the predicted values of the model. The RMSE is not particularly sensitive because its aggregation and mean calculation tends to mask errors in the model. By contrast, R^2 measures how well the model and the original values correlate with each

- *Bastian Rieck and Heike Leitte are with the Interdisciplinary Center for Scientific Computing. E-mail: {bastian.rieck,heike.leitte}@iwr.uni-heidelberg.de.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

other, i.e.

$$R^2 = \operatorname{cor}\left(\{m_1, \dots, m_n\}, \{s_1, \dots, s_n\}\right)^2, \tag{2}$$

where cor refers to *Pearson's correlation*. R^2 has the known weakness of being unable to detect systematic over- and underpredictions of a model [12, pp. 95–97]. It is also not suitable to describe the *accuracy* of the model. Both measures are commonly used in conjunction to outweigh the individual shortcomings.

2.2 Persistent homology

We complement these measures with one that quantifies structural stability in the high-dimensional parameter space. Going back to the example in Fig. 1, we do not want to quantify the small individual errors (which are already captured by RMSE), but assess how well major structures are preserved. In our example, we want to retain the links between local minima and maxima. Such large-scale structures are well captured using topological analysis. Persistent homology is an algorithm from computational topology that summarizes data sets using topological features. Such features are, for example, *connected components* (order 0), *tunnels* (order 1), and *voids* (order 2). Topology thus focuses on the connectivity information of a data set. To describe the connectivity of a 1D function, we look at its connected components, i.e. features of order 0. To finish our example, we will treat the 1-dimensional case first and touch only briefly upon the multivariate case. For detailed reading, we refer the reader to Edelsbrunner and Harer [8].

The 1-dimensional case Given a function $f\colon D \subseteq \mathbb{R} \to \mathbb{R}$, persistent homology describes the connectivity changes in the *sublevel sets* of f, i.e. sets of the form $L_c^-(f, c) = \{x \mid f(x) \leq c\}$. For discrete data, the function only has a finite number of (local) extrema, which we bring in ascending order, i.e. $e_1 \leq e_2 \dots$, where each e_i is either a minimum or a maximum. Starting from e_1, we then perform a "sweep" through the function values. We stop at each e_i and consider the number of connected components of f. If e_i is a local minimum, it will create a new component and we identify it with $c = f(e_i)$. Similarly, if e_i is a local maximum, it will destroy an existing component and we identify it with $d = f(e_i)$. We then look up the value c of the component that was destroyed and store the pair (c, d). The tuples (c, d) summarize the connectivity changes of f. By treating each pair (c, d) as a point in $\mathbb{R}^2$, we obtain a diagram in the plane—the *persistence diagram* (see Fig. 1, right, for an illustration). Each red point in the diagram corresponds to a min-max-connection of the red model function. Likewise, we computed those topological features for the measured data (grey points).

To compare persistence diagrams we have to define a distance metric. Given two diagrams X and Y, their *bottleneck distance* is

$$W_\infty(X, Y) = \inf_{\eta\colon X \to Y} \sup_{x \in X} \|x - \eta(x)\|_\infty, \tag{3}$$

where $\eta : X \to Y$ denotes a bijection and $\|x - y\|_\infty$ the *maximum norm*. The distance between X and Y is thus the smallest supremum over all bijections (see Fig. 1, right, for the respective matching). The bottleneck distance is bounded from above by the maximum distance between the functions [6], making it very stable and robust against noise. We have

$$W_\infty(X, Y) \leq \|f - g\|_\infty \tag{4}$$

for persistence diagrams X, Y corresponding to functions f, g, respectively. This property is known as the *bottleneck stability* (see Fig. 1 for both distances).

The bottleneck distance is very insensitive to details of the bijection between the two diagrams. In practice, we thus calculate the *qth Wasserstein distance* between diagrams X, Y as

$$W_q(X, Y) = \sqrt[q]{\left(\inf_{\eta\colon X \to Y} \sum_{x \in X} \|x - \eta(x)\|_\infty^q\right)}, \tag{5}$$

for which similar stability results hold [7]. Calculating both distances involves calculating maximum weighted matchings in bipartite graphs [8, pp. 229–236].

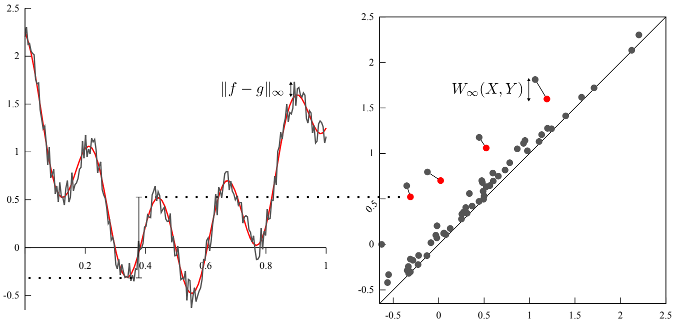


Fig. 1: An example for the bottleneck distance W_∞ between two persistence diagrams. Left: Model function superimposed with a noisy sampling or the original function. We keep track of the connected components of f while sweeping from top to bottom. In the figure, we marked $(c, d) \approx (-0.3121, 0.5224)$ and show the local minimum and maximum required for its calculation. Right: Superimposed persistence diagrams and the best bijection. The bottleneck distance is the distance between the two upper-most connected points. The remaining points correspond to components that are created from the noise. In all optimal bijections, these points are being assigned to the diagonal.

The multivariate case If we are given a multivariate data set with a set of corresponding scalar values, we need to extend the calculations from above. This requires a distance measure $\operatorname{dist}(\cdot, \cdot)$, such as the Euclidean distance, and a threshold ε—unlike the 1-dimensional case, we need to approximate the domain of our function first. To this end, we calculate a proximity graph on the data, connecting points x and y if $\operatorname{dist}(x, y) \leq \varepsilon$. By searching cliques in this graph, we expand it to a special simplicial complex (i.e. a generalized graph structure), the *Vietoris-Rips complex*. We then assign the simplices in the complex the scalar values of the input data (which corresponds to describing the domain of the function) and use an algorithm by Zomorodian and Carlsson [21] to calculate persistent homology.

3 Implementation

In the following, we assume that we are given a multivariate regression task and several models. More precisely, we require a set of input data in the form of an unstructured point cloud with points from $\mathrm{D} \subseteq \mathbb{R}^n$ and a set S of values from $\mathbb{R}$. We want to learn the functional relationship between D and S, i.e. we assume that we have a scalar function $f\colon \mathrm{D} \to \mathbb{R}$ from which our input data are noisy samples. As model values, we permit any set of values in $\mathbb{R}$. Commonly, model values are obtained using algorithms such as linear regression, support vector machines, and so on. However, we do not pose any restrictions on the source of the model values—any set of values from $\mathbb{R}$ is admissible. We furthermore require a distance metric $\operatorname{dist}\colon \mathrm{D} \times \mathrm{D} \to \mathbb{R}$ such as the Euclidean distance. We use dist to approximate the unknown domain of the function. Last, we require a distance threshold ε that is used to define neighbourhoods in the data. This distance threshold controls the coarseness of the approximation. High values of ε yield data sets in which almost all data points are considered to be similar, while low values of ε are more fine-grained. ε can be chosen efficiently by calculating the longest edge length in the *minimum spanning tree* of D, for example.

Using these data, we calculate the *Vietoris-Rips complex* of the input data. We use the model values to assign each 0-simplex (each vertex) in the complex a weight w. Higher-dimensional simplices are assigned the maximum weight of their vertices. We then sort the complex in ascending order to obtain a *filtration* of the input data. For each model M, we thus obtain a simplicial complex S_M. We calculate persistent homology of each S_M, resulting in a set of persistence diagrams $\{D_1, \dots, D_k\}$, where k denotes the number of models. We also calculate a persistence diagram D of the original input data.

In the following, we will refer to the Wasserstein distance measure as PH+W. By calculating the PH+W between D and each D_i, we obtain a measure of how well the value of a specific model approximates

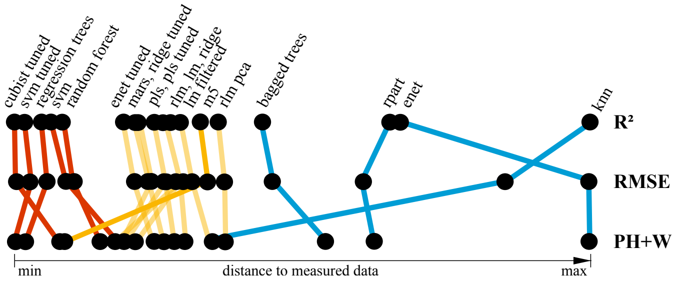


Fig. 2: Comparison of three distance measures: For each model, the graph shows the distance to the measured data using one of the three distance measures. Models are linked across distances by lines coloured according to model quality (red: good, yellow: medium, blue: poor).

the original data. We then calculate the PH+W for each pair D_i, D_j of persistence diagrams, resulting in a real-valued $k \times k$ matrix. This matrix describes the pairwise topological distances between the instances of the models. Using multidimensional scaling, we obtain a set of coordinates in $\mathbb{R}^2$ that best approximates the distances. The resulting point set describes the *model landscape* (ML) of the given models. Each point in the ML corresponds to an instance of single model. We colour-code nodes by the quality of their respective models. Distances in the ML directly encode differences in the topological behaviour of the models. If two models tend to result in the same predictions, their corresponding nodes will be placed in close vicinity to each other.

4 Results

In the following, we will demonstrate the model analysis in detail using an example from drug development. An important task in this field is to predict the *solubility* of chemical compounds, i.e. how easily a compound is being dissolved in a solvent. Solubility is is of paramount importance if a substance is to be administered as a drug (e.g. orally or through injection). Tetko et al. [19] and Huuskonen [11] investigated a number of chemical compounds with known solubility values. They derived a complex set of descriptors and used linear regression models (among others) to obtain predictors for the solubility values. In the following, we will work with their database of 1267 compounds.

Each compound is described by a 300-dimensional vector of measured properties. As suggested by Kuhn and Johnson [12, p. 103 ff.], we perform data cleaning and model training as a preprocessing step. We compare the following models:

bagged trees	Bagged model trees
cubist	Cubist regression trees
enet	Regularized regression w/ penalties
knn	*k*-nearest neighbours
lm	Linear regression
mars	Multivariate adaptive regression splines
m5	Model trees
pls	Partial least squares
svm	Support vector machines
random forest	Random forests
rpart	Single regression trees
regression trees	Boosted regression trees
ridge	Ridge regression w/ penalties
rlm	Robust lin. reg.
rlm pca	Robust lin. reg. w/ preprocessing

In future annotations, the "tuned" variants refer to variants in which parameter tuning has been applied (e.g. cross-validation) to improve results. We use the ground truth of Kuhn and Johnson [12, pp. 221–223] for our analysis. Using a combination of R^2, RMSE, and manual inspection, they partitioned the models into three performance groups (Fig. 2).

The distance measure for the data points in $\mathbb{R}^{300}$ with mixed attribute types is a combination of the *Hamming distance* (for binary attributes) and the *Euclidean distance* (for continuous attributes). We compute persistent homology for $\varepsilon = 30$ and use the solubility values of each model as weights for the 0-simplices (see Sec. 3). Persistent homology computation for each model takes about 6 s (with more efficient implementations [1] available). PH+W calculations then take an additional 5 s in total.

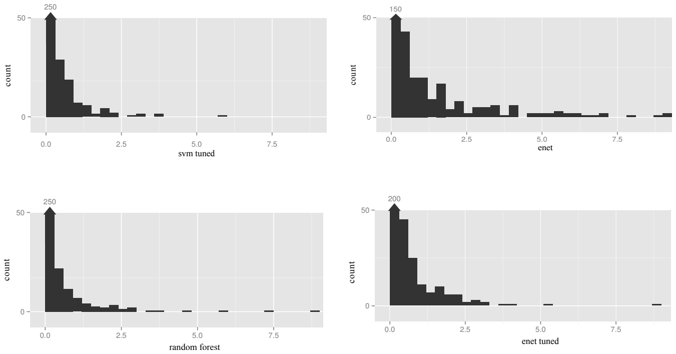
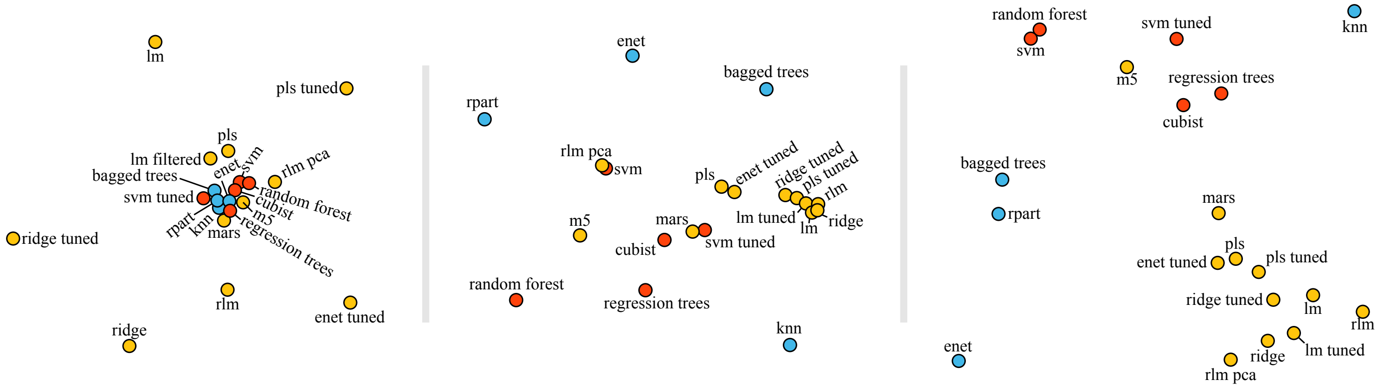


Fig. 3: Histograms of distances between selected models and measured data: *svm tuned* (one of the best models), *enet* (one of the worst models), *enet tuned* (to show the effects of tuning), *m5* (judged differently by PH+W)

We first analyze how well the models agree with the measured data. Fig. 2 illustrates the respective distances for the two classical measures R^2 and RMSE, as well as our new distance PH+W. Overall, the ranking is fairly consistent. We observe homogeneous groups for the models of varying quality (red, yellow, and blue). While R^2 and RMSE have very similar distances for most models, we observe alterations for PH+W: There are several changes in the order of models within the groups; the clustering of the high-performance models is more diverse. Two models (*m5* and *knn*) feature distances that belong to better model classes, i.e. *m5*: yellow → red, *knn*: blue → yellow.

The larger diversity between model qualities in PH+W can be partly explained by the distribution of errors (Fig. 3). The top row compares one of the best (*svm tuned*) to one of the worst (*enet*) models. The overall shape of the two histograms is similar. We observe a general tendency towards larger errors in *enet*. Variance increases from 0.44 for *svm tuned* to 5.23 for *enet*. There are also substantially more outliers, i.e. distances greater than 2.5. *enet tuned* demonstrates how both error types (high variance and many outliers) are significantly reduced by altered algorithm parameters (variance 0.78). *enet tuned* is ranked by both R^2 and RMSE as the best medium-performance model. Comparing the histogram of *enet tuned* to *random forest* (the worst model in the high-performance group), we observe a significant increase of low-error samples, while there are also more outliers. The influence of this comparatively small group of larger errors is diminished by the mean computation and makes differences between models very blurred. In the application scenario, this implies that the prediction for many samples is very good, but that significant outliers may occur using *random forest*. By contrast, the amount of outliers for *svm tuned*, the best model according to R^2 and RMSE, is substantially reduced.

To better understand the changes in ranks, we investigate *m5* more closely. With its average distribution of distances (variance 1.24), *m5* is a model of medium quality. This is reflected by the values for R^2 and RMSE. PH+W rates *m5* much better and places it among the high-quality models. To further explore model similarities, we calculate the *model landscapes* using relative pairwise distances for R^2, RMSE, and PH+W (Fig. 4). Using the R^2 correlation-based distance, most of the models form a dense cluster surrounded by multiple outliers (Fig. 4a). There is no clear distinction between the quality classes. The model landscape for RMSE already exhibits more structure (Fig. 4b). Many models of medium quality (yellow) are clustered. The good models (red) also form a loose cluster with medium models interspersed (*m5*, *mars*, and *rlm pca*). PH+W (Fig. 4c) features the clearest distinction of performance classes. The red and yellow clusters are clearly distinguishable, and the low-performance models in blue are distributed along the periphery. Similar to RMSE, *m5* is close



(a) R^2 (b) RMSE (c) Wasserstein

Fig. 4: Inter-model distances for the three quality measures: The 2D *model landscapes* reveal for each measure the pairwise model distances in high-dimensional space. Spatial proximity reflects short distances. The models are coloured according to a manual classification (red: good, yellow: medium, blue: poor).

to the high-quality models while *mars* lies on the boundary between red and yellow. The placement of *m5* is caused by known instabilities in the model [4]—its predictive quality is going to increase with more data points.

A further advantage of the *model landscape* is that it facilitates model selection for competing models. This is especially necessary in situations where the model yielding the best overall performance has a prohibitive computational cost. In our example, this holds for *cubist* and *regression trees*, who take several minutes to be calculated even on small data sets. Kuhn and Johnson [12] thus conclude that the medium-performance models, such as *lm* or *mars*, may actually be more suited for real-world data because they scale better to thousands of data points.

5 Conclusion & future work

In this paper, we enhanced visual methods for comparative model analysis. We introduced a novel quality measure (PH+W) for multivariate models based on the Wasserstein distance of the persistence diagrams of models. The novel measure describes structural stability in the high-dimensional space and is very robust against noise. We also employed the *model landscape*, a 2D representation of pairwise model distances, to explore relationships between models. We found that PH+W is more sensitive in discriminating different models than the commonly-used measures R^2 and RMSE.

There are several possible enhancements to our system that would further strengthen the analytic power. Gosink et al. [10] use *Bayesian model averaging* to obtain a measure of the uncertainty of predictive models. Integrating similar information into our visualization would facilitate choosing a suitable model. With arbitrary model algorithms, however, uncertainty analysis is more complicated because we usually do not know the proper domain of the input function. Another aspect for future research thus involves investigating how to strengthen the domain approximation by persistent homology, e.g. via Morse-Smale complexes [9].

Acknowledgments

This research was supported by a stipend of the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS MathComp) and the German Research Foundation (DFG).

References

[1] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner. PHAT– persistent homology algorithms toolbox. In *Mathematical Software – ICMS 2014*, volume 8592 of *LNCS*, pages 137–143. Springer, 2014.

[2] S. Bergner, M. Sedlmair, T. Möller, S. N. Abdolyousefi, and A. Saad. ParaGlide: Interactive parameter space partitioning for computer simulations. *IEEE T. Vis. Comput. Gr.*, 19(9):1499–1512, 2013.

[3] E. Bertini, A. Tatu, and D. Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE T. Vis. Comput. Gr.*, 17(12):2203–2212, Dec. 2011.

[4] L. Breiman. Heuristics of instability and stabilization in model selection. *Ann. Statist.*, 24(6):2350–2383, 1996.

[5] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE T. Vis. Comput. Gr.*, 16(6):1467–1475, Oct. 2010.

[6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, Jan. 2007.

[7] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and Y. Mileyko. Lipschitz functions have L_p-stable persistence. *Found. Comput. Math.*, 10(2):127–139, Apr. 2010.

[8] H. Edelsbrunner and J. Harer. *Computational topology: An introduction*. American Mathematical Society, 2010.

[9] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE T. Vis. Comput. Gr.*, 16(6):1271–1280, Nov. 2010.

[10] L. Gosink, K. Bensema, T. Pulsipher, H. Obermaier, M. Henry, H. Childs, and K. Joy. Characterizing and visualizing predictive uncertainty in numerical ensembles through bayesian model averaging. *IEEE T. Vis. Comput. Gr.*, 19(12):2703–2712, Dec. 2013.

[11] J. Huuskonen. Estimation of aqueous solubility for a diverse set of organic compounds based on molecular topology. *J. Chem. Inf. Comput. Sci.*, 40(3):773–777, May 2000.

[12] M. Kuhn and K. Johnson. *Applied predicitive modeling*. Springer, 2013.

[13] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - Rapid visual prototyping of a common rail injection system. *IEEE T. Vis. Comput. Gr.*, 14(6):1699–1706, Nov. 2008.

[14] T. Mühlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE T. Vis. Comput. Gr.*, 19(12):1962–1971, Dec. 2013.

[15] J. A. Pretorius, M.-A. Bray, A. E. Carpenter, and R. A. Ruddle. Visualization of parameter space for image analysis. *IEEE T. Vis. Comput. Gr.*, 17(12):2402–2411, Dec. 2011.

[16] P. Rheingans and M. desJardins. Visualizing high-dimensional predicitive model quality. In *Proc. IEEE Visualization '00*, pages 493–496, 2000.

[17] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE T. Vis. Comput. Gr.*, 99:1, 2014. To appear.

[18] B. Spence, L. Tweedie, H. Dawkes, and H. Su. Visualization for functional design. In *Proc. Symp. InfoVis*, pages 4–10, 1995.

[19] I. V. Tetko, V. Y. Tanchuk, T. N. Kasheva, and A. E. P. Villa. Estimation of aqueous solubility of chemical compounds using E-state indices. *J. Chem. Inf. Comput. Sci.*, 41(6):1488–1493, Nov. 2001.

[20] A. Unger, S. Schulte, V. Klemann, and D. Dransch. A visual analysis concept for the validation of geoscientific simulation models. *IEEE T. Vis. Comput. Gr.*, 18(12):2216–2225, Dec. 2012.

[21] A. Zomorodian and G. Carlsson. Localized homology. *Comput. Geom.*, 41(3):126–148, Nov. 2008.