Using Maximum Topology Matching to Explore Differences in Species Distribution Models

Jorge Poco, New York University
Harish Doraiswamy, New York University
Marian Talbert, U.S. Geological Survey
Jeffrey Morisette, U.S. Geological Survey
Cláudio T. Silva, New York University

ABSTRACT

Species distribution models (SDM) are used to help understand what drives the distribution of various plant and animal species. These models are typically high dimensional scalar functions, where the dimensions of the domain correspond to predictor variables of the model algorithm. Understanding and exploring the differences between models help ecologists understand areas where their data or understanding of the system is incomplete and will help guide further investigation in these regions. These differences can also indicate an important source of model to model uncertainty. However, it is cumbersome and often impractical to perform this analysis using existing tools, which allows for manual exploration of the models usually as 1-dimensional curves. In this paper, we propose a topology-based framework to help ecologists explore the differences in various SDMs directly in the high dimensional domain. In order to accomplish this, we introduce the concept of maximum topology matching that computes a locality-aware correspondence between similar extrema of two scalar functions. The matching is then used to compute the similarity between two functions. We also design a visualization interface that allows ecologists to explore SDMs using their topological features and to study the differences between pairs of models found using maximum topological matching. We demonstrate the utility of the proposed framework through several use cases using different data sets and report the feedback obtained from ecologists.

Keywords: Function similarity, computational topology, species distribution models, persistence, high dimensional visualization.

1 INTRODUCTION

Species distribution models (SDM) combine observations of species occurrence or abundance with environmental layers. They are used to gain ecological insights and to predict distributions across various landscapes including terrestrial, freshwater, and marine realms [16]. They help ecologists answer questions about the relationship between the species and environmental variables.

There are multiple modeling approaches used for SDMs, some relying more on the traditional least-square and maximum likelihood methods to relate predictors data to the observed data; while some use an iterative, machine learning and Monte Carlo resampling techniques to explore the relationship. The different modeling techniques have varying degrees of complexity and the model to use depends on the goals of the study and the data (primarily the response and the assumptions). However, considering the application of multiple modeling techniques to a common data set can provide insight into the behavior of each modeling approach [44].

Forecast of species' distributions presents substantial discrepancies based on the predictive modeling approach used [10, 23] highlighting the uncertainties associated with these predictions [10, 32]. Several studies show that these algorithms can predict substantially different future potential ranges even if current predictions are largely congruent [7, 40, 44]. Such model disagreement helps ecologists understand areas where their data or understanding of the system is incomplete and to either guide further investigation of a model, or to identify an implausible model. These discrepancies can also be an important source of uncertainty in model projections to new spatial or temporal extents. It can also help the improvement of methods and / or parameters used for the models. It is therefore important for ecologists to be able to explore in detail the different models and be able to study the differences between them.

Problem Definition. SDMs are typically high dimensional scalar functions, where the dimensions correspond to different model algorithms and environment variables, also known as predictors. Ecologists are interested in studying the behavior of these models over their parameter space comprised of their predictors. However, their current approach resorts to visualizing 1-dimensional (1D) slices of the models. That is, in considering the influence of one specific predictor, the common technique is to select a predictor of interest and fix the values of the other predictors to their mean values, and compare the variation of the models with respect to the selected predictor. This results in a 1D curve known as *response curve* [43] (e.g., see Fig. 8). The main shortcoming of restricting the analysis to considering only one predictor at a time is that it is not possible to obtain an accurate view of the model. This is because, features resulting from the interactions between the other predictors are lost through such dimensionality reduction. More importantly, even when looking at 1D slices, the response curves are restricted to the fixed value of the other predictors. While there has been some work where ecologists analyze two-dimensional slices of the models [17, 44], the above problems still hold.

Contributions. The goal of this work is to help ecologists understand the interactions between the predictors in SDMs, and thus have a better understanding of what drives the various species. To this end, we propose the use of computational topology to help explore and compare SDMs directly in the high dimensional domain. In particular, we use the extrema of the corresponding scalar functions to *guide* the users towards interesting features of the SDM.

While such exploration of the SDMs will provide more flexibility to the ecologists, manual comparison between the two models is still a time consuming and often impractical process. To overcome this, we propose a novel technique that can be used to compare two scalar functions in a locality-aware manner. We do this by first creating a bipartite graph where the edges correspond to possible correspondences between the extrema of the two functions. The edge weights are defined such that they reflect both the spatial locality of the extrema, as well as the likeness in terms of their function values. The maximum weight matching of the bipartite graph is then computed to obtain the correspondences between the set of extrema. These correspondences are then used to compute a topological similarity measure between the two functions. We also show through experiments the robustness of the matching and the resulting topological similarity measure.

We design a visualization interface to help ecologists explore SDMs and analyze the differences between them. Finally, working together with ecologists we demonstrate the effectiveness of our technique and the user interface through several use case scenarios involving SDMs of different species.

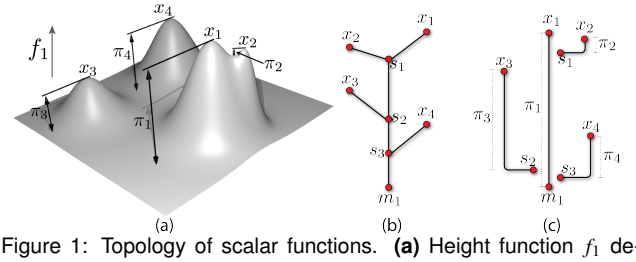


Figure 1: Topology of scalar functions. **(a)** Height function f_1 defined on a 2-dimensional manifold having 4 maxima. π_i represents the persistence of a maximum x_i. **(b)** The join tree tracks the connectivity of the super-level sets of a scalar function. **(c)** Each branch in the branch decomposition of the join tree corresponds to the path between a creator-destroyer critical point pair.

2 Background

We now provide the necessary background on concepts from computational topology that form the mathematical and algorithmic basis of this work. We refer the reader to the following textbooks [15, 26] for a comprehensive discussions on these concepts.

Morse functions and Species Distribution Models. Let $\mathbb{M}$ denote a d-manifold with or without boundary. Given a smooth, real-valued function $f : \mathbb{M} \to \mathbb{R}$ defined on $\mathbb{M}$, the *critical points* of f are exactly where the gradient becomes zero. The function f is a *Morse function* if it satisfies the following conditions [26]: (1) All critical points of f are non-degenerate and lie in the interior of $\mathbb{M}$; (2) All critical points of the restriction of f to the boundary of $\mathbb{M}$ are non-degenerate; and (3) All critical values are distinct *i.e.*, $f(p) \neq f(q)$ for all critical points $p \neq q$. For a Morse function f defined on a d-manifold $\mathbb{M}$, there are $d+1$ types of critical points indexed from 0 to d. In this work, we are interested in the two most familiar types – *minimum* (with index 0) and *maximum* (with index d), corresponding to a point p whose function value is smaller, or larger, than all other points within a sufficiently small neighborhood of p, respectively. Fig. 1(a) shows a height function, f_1, defined on a 2-manifold. This function consists of 4 maxima – x_1, x_2, x_3, and x_4.

A species distribution model is a d-dimensional function $m : \mathbb{R}^d \to \mathbb{C}$, where $\mathbb{C} = [0,1]$ denotes the unit interval. It assigns a probability for the presence of a given species based on the values of its d predictors. In the remaining discussion, we assume that the input SDMs are Morse functions. In case the above conditions do not hold, simulated perturbation of the function [14, Section 1.4] ensures that no two critical values are equal.

Topological persistence. A sub-level set of a function f, $\mathbb{M}^{(-\infty,a]} := \{x \in \mathbb{M} \mid f(x) \leq a\}$, is the set of all points having function value less than or equal to a. A super-level set is similarly defined as the preimage of the interval $\mathbb{M}^{[a,+\infty)}$.

Consider the sweep of the function f in increasing order of function value. The topology of the sub-level sets changes when this sweep passes a critical point. In particular, at a critical point, either new topology is generated or some topology is destroyed, where topology is quantified by a class of 'cycles'. For example, a 0-dimensional cycle represents a connected component, a 1-dimensional cycle is a loop that represents a tunnel, and a 2-dimensional cycle bounds a void. A critical point is a creator if new topology appears and a destroyer otherwise. One can pair up each creator v_1 uniquely with a destroyer v_2 that destroys the topology created at v_1. The persistence value of v_1 and v_2 is defined as $f(v_2) - f(v_1)$, which intuitively indicates the lifetime of the feature created at v_1, and thus the importance of v_1 and v_2.

The function in Fig. 1(a) consists of three creator-destroyer pairs – (x_2, s_1),(x_3, s_2), and (x_4, s_3). While the global maximum x_1 has a persistence value of ∞, we use a notion of extended persistence where in addition to the above pairs, the global maximum is paired with the global minimum [1]. The persistence values of the set of maxima x_i of the function in Fig. 1(a) is highlighted as π_i.

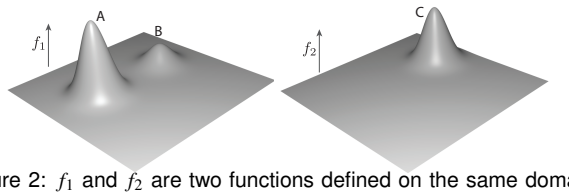


Figure 2: f_1 and f_2 are two functions defined on the same domain. Existing techniques identify peak C to be similar to A instead of B even though B and C are in the same neighborhood of the domain.

Topological persistence of a feature measures the amount of simplification required to smooth the input function in order to remove that feature. This property is later used to define a distance measure between two SDMs.

As mentioned above, in this paper we only consider extreme points of the input function as features. Given an input domain of size n, the persistence of such features can be computed efficiently in $O(n \log n + n\alpha(n))$ time using the union-find data structure [1].

Merge trees. A *join tree* tracks the topology of the super-level sets of the input function, while the *split tree* tracks the topology of the sub-level sets [9]. The join tree and split tree are together known as *merge trees*. Fig. 1(b) shows the join tree of the function shown in Fig. 1(a). The join / split tree is computed using the union-find data structure to keep track of the connected components of the super-level set (or the sub-level set). This procedure also returns the set of creator-destroyer pairs corresponding to the topological features.

A merge tree can be decomposed into a set of *branches* using the obtained critical point pairs [31]. Each branch corresponds to the path in the merge tree between a creator-destroyer critical point pair. Thus, the height of a branch represents the persistence of the corresponding critical points. Fig. 1(c) shows the branch decomposition of the join tree in Fig. 1(b). The smoothing of a function obtained by removing an extremum can be represented abstractly by removing the branch corresponding to that extremum together with all its sub-branches. This observation is key in our algorithm that computes the topological similarity measure between two SDMs.

3 Related Work

In this section, we first briefly discuss related work that are used to explore high dimensional functions. Next, we survey topology based techniques that are used for comparing two scalar functions.

Exploring High Dimensional Functions There are multiple visual analytic techniques to explore the parameter space of high dimensional scalar functions (also referred to as models). Most of these methods are based on sampling the parameter space or using regression algorithms to approximate / predict output from unknown configurations. Matkovic et al. [25] proposed to visualize multirun data as families of data surfaces (with respect to pairs of independent dimensions) in combination with projections and aggregation of the data surfaces at different levels. The same authors [24] also proposed to generate new sample points by interactively narrowing down the control parameters in the visualization via brushing to support visual steering of a simulation. Along the same lines HyperMoVal [33] was designed to visually relate one or more high-dimensional scalar functions with validation data. Later, Berger et al. [4] extended HyperMoVal using regression models for a continuous exploration of the sample parameter space. Similarly, we can find applications of parameter exploration in other domains such as image segmentation [41]. Other approaches partitioned the input space and provided visual analytics strategies for exploration of the input space using one or two parameters at the time [5, 28]. Some work has been done in exploring and understanding the differences in model simulations. For example, Poco et al. [34] propose the SimilarityExplorer tool for a visual comparison of the model output but they do not explore the input parameter space. However, all of these approaches require users to manually explore the space in order to identify interesting regions.

Topological abstractions have also been used to create visual representations of high dimensional functions. Topological Landscapes [42] provided a 2D terrain representation having the same contour tree as the input high-dimensional scalar function. When the input are point clouds Oesterling et al. [30] proposed to reconstruct the scalar function using density kernels and used topological landscapes to visualize the density of points using a 2D terrain. Geber et al. [20] segmented the input domain using an approximate Morse-Scale complex on a cloud of point samples. Then each segment was represented by a curve using a regression. Finally those curves were visualized in 2D space using dimensionality reduction algorithms. While these techniques helped users understand the topology of the involved function, it was difficult to use these methods to compare scalar functions as the neighborhood information was lost in the transformation to a 2D representation.

In the ecology domain, while there has been some work on trying to study two-dimensional slices of SDMs [17, 44], ecologists mostly use the SAHM package [43] which supports exploration through 1D response curves.

Comparing Scalar Functions Early methods of comparing scalar functions directly used the persistence of the critical points of the functions to do so. A distance function, usually bottleneck distance, between the persistence diagrams [11] of two functions are used to compare them. Using an alternate representation, called barcode, Carlsson et al. [8] represented the persistence of the features as intervals on a real line. They then defined a metric to compute the similarity between two barcodes. A disadvantage of using a pure persistence based measure is that they do not capture the neighborhoods of the features.

More recent methods for comparing scalar functions used some form of topological abstraction of the scalar functions to compare them. Morozov et al. [27] defined the interleaving distance between two merge trees as the minimum cost of shifting points in one tree to obtain a mapping of one tree to the other. Beketayev et al. [3] defined a distance between two merge trees by comparing all possible branch decompositions of the two trees. Bauer et al. [2] extended the interleaving distance between two merge trees to Reeb graphs and proposed the functional distortion distance to compare two Reeb graphs, where a Reeb graph is a topological structure which tracks the connectivity of level sets of a scalar function with increasing function value. More recently Narayanan et al. [29] proposed a distance measure between two scalar functions based on the maximum common subgraph between complete extremum graphs, where an extremum graph is a topological data structure that captures proximity between extreme points in a scalar field [12]. While extremum graphs naturally encode locations of extrema and saddles, it is with respect to a single function. The usefulness of this locality information is not clear when comparing two functions. Alternative to computing distance measures, topological structures have also been used to structurally compare two functions. Multi-resolution Reeb graphs [22] as well as Morse-Smale complexes [18] have been used for comparing two shapes. Saikia et al. [35] introduced a data structure called extended branch decomposition graphs using which they could compare between all sub-trees of two merge trees. Topological abstractions have also been used to identify similar structures within a scalar function [37, 38].

While the above methods capture adjacency based on the connectivity between level sets, they still suffer from two shortcomings. First, it is possible for two adjacent features (adjacent edges) to actually be far from each other. Second and more importantly, the actual locality of the features identified as similar need not be located in the same locality of the domain, which is a requirement for the target application. For example, consider the two functions shown in Fig. 2. the above techniques would identify maximum A in f_1 with C in f_2 even though the two maxima are far from each other. However, given that B and C are in a similar locality of the domain, we are interesting in identifying B with C.

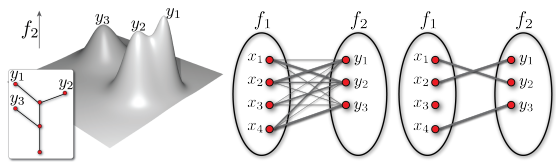


Figure 3: Computing the maximum topology matching. **(a)** 2-dimensional scalar function f_2 that is compared with the function f_1 in Fig. 1(a). Join tree of f_2 is shown in the bottom left side. **(b)** The constructed bipartite graph between the maxima of the two functions. **(c)** The computed matching between the maxima.

Instead of an abstraction, level sets and their properties have also been used for comparing scalar functions [6, 36, 39]. Since these techniques require computing the level sets, extending them to work for high dimensional functions is non-trivial.

4 Scalar Function Similarity

We now describe our technique to compare two scalar functions that are defined on the same domain. The main idea is to identify the best match, in terms of the location and function value, between the set of extrema (of the same type) of the two functions. This matching is then used to compute the similarity measures between the functions.

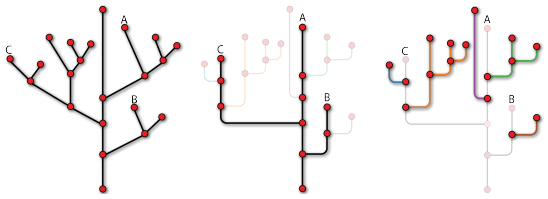
In this section, we first describe the procedure to identify the correspondence between the set of extrema of two functions. Next we define two similarity measures between the functions and describe how they are computed using the found correspondences. Without loss of generality, the techniques in this section are described with respect to the set of maxima of the functions. The same procedures apply to the set of minima as well.

4.1 Maximum Topology Matching

The first stage in identifying the similarity between two scalar functions f_1 and f_2 is to identify the correspondence between the extrema of the functions. Intuitively, two similar functions will have the same topology and hence the same number of extrema. Thus, the goal is to get the "best" match possible in the sense that there is one-to-one mapping between the extrema of the two functions. Without loss of generality, we assume that the two functions are normalized between 0 and 1.

Let M_1^+ and M_2^+ be the set of maxima of f_1 and f_2 respectively. We first create a complete weighted bi-partite graph $G_T(M_1^+, M_2^+, E^+)$ in which the two partitions corresponds to the maxima of the two functions respectively. Consider a pair of maxima $a \in M_1^+$ and $b \in M_2^+$. Let the difference between their function values be $\delta_{a,b} = |f_1(a) - f_2(b)|$. Let $d_g(a,b)$ denote the distance between the pair of maxima. Since the SDM is defined on $\mathbb{R}^d$, we use the Euclidean distance for this purpose. We assign a weight $w_{a,b}$ to the edge corresponding to the pair of maxima a and b as follows: $w_{a,b} = (1 - \delta_{a,b}) \times \exp(d_g(a,b)^2/r^2)$. Here, r is a cut-off radius, which acts as a knob to define the neighborhood sensitivity.

The weight $w_{a,b}$ essentially consists of two parts. A high value of $(1 - \delta_{a,b})$ implies a high similarity between the two maxima in terms of their function value. The weighting term $\exp(d_g(a,b)^2/r^2)$ ensures that importance is given to pairs of maxima that are closer to each other, thus preserving the neighborhood locality. Thus a high weight between a pair of maxima implies that they are *similar* not only in terms of their function value, but are also within the same locality of the domain. For example, in order to compare function f_2 shown in Fig. 3(a) with function f_1 from Fig. 1(a), we create the bipartite graph shown in Fig. 3(b). The thickness of the edges represents their weights. Note that the edges corresponding



(a) (b) (c)

Figure 4: Computing topological similarity. **(a)** Example join tree of a function that is being compared. Let maxima A, B and C be matched to maxima of the other function. **(b)** The first step of the algorithm computes the matching tree T'_S, a connected sub-tree of the join tree T_S induced by the matched maxima. **(c)** The connected components of $T_S \setminus T'_S$ corresponds to the regions of the function that has to be simplified to obtain a perfect matching. The minimum amount of simplification required to do so measures the topological similarity between the two models.

to maxima pair that are nearby in the function domain have weight higher than those that are further away.

We next compute the maximum weighted matching [19] on the graph G. A *matching* is defined as a set of pairwise non-adjacent edges. A maximum weighted matching is defined as a matching where the sum of values of the edges in the matching has maximum value. The resultant matching provides the correspondence between the set of maxima of the two functions. For the example of functions f_1 and f_2, the obtained matching is shown in Fig. 3(c). Note that our technique matches the maxima x_1 to y_2 and x_2 to y_1 due to their proximity. This is unlike existing techniques that do not use the locality information to match features, which would have matched x_1 to y_1 and x_2 to y_2. Also, these techniques would have matched x_3 to y_3 since they use the relative persistence of features when computing similarity.

4.2 Similarity Measures

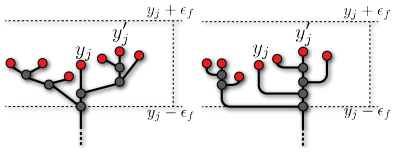
Topological Similarity. The topological similarity between f_1 and f_2 is defined as the effort required to make the two functions have the same number of maxima in the same neighborhood of the domain. Such functions will produce a *perfect matching* in G, that matches all vertices of the bipartite graph. This quantity is measured as the minimum amount of simplification that is to be performed to attain such a perfect matching.

Consider the function f_1 having the set M_1^+ as its maxima. Let $C \subseteq M_1^+$ be the set of maxima that have a corresponding match in M_2^+. Then $\overline{C} = M_1^+ \setminus C$ is the set of maxima that have to be simplified. The join tree and the appropriate branch decomposition is used to compute, τ_1, the amount of simplification required as follows. Let r_o be the root of the join tree T_S. That is, r_o is the global minimum of the function f_1. In the first step, we construct the *matching tree* T'_S, the join tree of f'_1 which is the function f_1 in which the set of maxima $\overline{C}$ are removed (simplified). This tree is constructed as follows:

1. For each maximum $m \in C$, construct the path L_m, which is the unique path from the leaf corresponding to m to the root r_o.
2. the matching tree $T'_S \subset T_S$ is the tree induced by the paths L_m computed above, *i.e.*, $\{T'_S = \bigcup_{m \in C} L_m\}$

Fig. 4(b) shows the matching tree computed from the join tree in Fig. 4(a). Here, three maxima were matched in the bipartite graph.

Let $\overline{T_S} = T_S \setminus T'_S$. Consider the connected components K of $\overline{T_S}$. Simplifying the set of maxima in $\overline{C}$ is equivalent to removing each of these connected components from T_S. These components corresponds to a connected sub-tree of T_S. The effort τ^k required for simplifying a given component k is equal to the *height* of the largest branch of the corresponding sub-tree. Fig. 4(c) shows the different components that have to be simplified for the example in Fig. 4(a).



(a) (b)

Figure 5: Effect of noise in the neighborhood of a maximum. **(a)** Presence of noise could introduce multiple extraneous extrema in the neighborhood of a relevant maximum y_j. **(b)** In case one of the noisy maximum y'_j is matched in the maximum matching, then the amount of simplification needed to remove y_j is bounded by ε, while the simplification needed for other extraneous maxima is bounded by 2ε.

τ_1 is then computed as the maximum value of τ^k over all components $k \in K$. τ_2, the minimum amount of simplification required for function f_2 is computed in a similar manner. The *topological similarity* $\tau = \max(\tau_1, \tau_2)$ is the minimum simplification required to obtain a perfect matching between the two functions.

Functional Similarity Given a perfect matching between the maxima of the two topologically similar functions, it is still possible that the matched maxima could differ in their function values. The functional similarity measures this difference. Formally, the functional similarity ϕ is the maximum $\delta_{a,b}$ over all edges (a,b) that are part of the matching. Intuitively, this quantity is used to measure the maximum amount of change required to construct functionally similar functions from topologically similar functions.

4.3 Implementation

In order to efficiently compute the topology of a species distribution model $m : \mathbb{R}^d \to \mathbb{C}$, the high-dimensional domain of m is approximated as a nearest-neighbor graph, G, of a set of points sampled uniformly, using Latin Hypercube Sampling, from the domain of $\mathbb{R}^d$. m is then represented as piece-wise linear (PL) function defined on $G - m : G \to \mathbb{C}$. The function is defined on the vertices of the graph and linearly interpolated within each edge.

We use the sweep algorithm by Carr et al. [9] to compute the merge trees, which can be accomplished in $O(n \log n + m\alpha(m))$ time. Here, n and m are the number of vertices and edges, respectively, in G and α is the inverse Ackermann function. Let the number of extrema in the two functions, f_1 and f_2, be $t_1 = O(n_1)$ and $t_2 = O(n_2)$ respectively. The created bipartite graph has $n_v = t_1 + t_2$ nodes and $n_e = t_1 \times t_2$ edges. Computing the maximum weight matching can be accomplished in $O(n_v^2 \log n_v + n_v n_e)$ using Dijkstra's algorithm with a Fibonacci heap [19]. Even though the time complexity is cubic, we achieve interactive running times in practice for computing the matching (see supplemental document).

4.4 Effect of Noise

Noise-based artifacts are common in real world data sets. It is therefore important to consider the effect of noise to the stability of the matching, and the resulting similarity measures. If the original matching remains even with noise, then given the low persistence of the noisy extrema that are added, there is no significant change to the similarity measures. So let us assume that the matching is different from the original. Consider a matched pair of maxima (x_i, y_j) between functions f_1 and f_2. Without loss of generality, assume that there was noise introduced into the function f_2. This would potentially create additional maxima in the neighborhood of y_j. Let the effect on the function value variation due to the noise be bounded by ε_f. Depending on the changes in the weights, the following scenarios are possible.

1. The matching (x_i, y_j) does not change due to noise.
2. The matching algorithm pairs x_i with a maximum y'_j in the neighborhood of y_j. In this case, both τ and ϕ change by a maximum of ε_f. This is because, y'_j is in the resulting matching tree, and y_j has to be simplified. The persistence of y_j in

the new configuration is then bounded by the change in function value (See Fig. 5), which in the worst case is $2\varepsilon_f$.

3. The matching pairs x_i with y_k not in the neighborhood of y_j. This implies that weight of the edge (x_i, y_k) managed to increase past the weight of edge (x_i, y_j), *i.e.*, $w_{x_i,y_j} \approx w_{x_i,y_k}$. While the weight of the matching in this case would not significantly change, the values of τ and ϕ could be affected.

We are interested in further exploring Case 3 above when the maximum y_j as well as the set of maxima y'_j that was created (due to noise) in the neighborhood of y_j remains unmatched. If at least one of them is matched to another maximum x'_i, then the change to τ would be similar to Case 2 above.

Since the function values are between 0 and 1, the weights of the edges in G is always between 0 and 1. When the weights of the edges under consideration are low, there are three possibilities:

1. Both y_j and y_k are far away from x_i; or
2. One of the maxima, say y_j is far from x_i but has $\delta \approx 0$, and for y_k, δ is high while it is within the neighborhood of x_i; or
3. Both maxima are in the neighborhood of x_i, but have very high δ (close to 1).

All the above three cases produce an *uneven match*, *i.e.*, the matched pair significantly differ in function values, or are not in the neighborhood of each other. In order to avoid such matches, we perform an additional pruning step to remove such low weight edges from the bipartite graph. Thus this step ensures that there is no significant change in the similarity measures in such cases. Note that we use a value of 10^{-6} in this filtration step, thus ensuring that significant matched pairs are not removed.

On the other hand, let the weights of the edges under consideration be high. Given the exponential decrease in the weights with respect to distance between the maxima, we can safely assume that the two maxima are in the neighborhood of x_i. Assuming that r is small (we use 0.1 in our experiments), we can safely infer that the two function values are similar (and high). Thus, there is no effect on ϕ. Let s be a saddle that can be reached through a descending path from both y_j and y_k. If there are no other matches in both the sub-trees, from y_j to s and from y_k to s, then there is no change to τ. In case there are other matches, then the persistence of the two maxima in their respective sub-trees decides the maximum change in τ, which is bounded by $|\pi_{y_k} - \pi_{y_j}|$. As we show next, we found that in practice the changes to τ was indeed small due to noise.

Experiments In order to test the robustness to noise, we perform three types of experiments. In the first experiment, we fix a function f_1, and artificially induce noise to f_1 to obtain a noisy function f_1^*. The amount of noise induced was bounded by $\varepsilon = 10^{-4}$. We then compute the similarity measures between f_1 and f_1^*. Ideally the topological similarity τ should be *zero*. We performed this experiment for the different models across three data sets. The mean and standard deviation of τ across these tests were 1.18×10^{-4} and 4.05×10^{-5} respectively. Note that this is less than the 2ε bound.

In the second experiment, we consider pairs of functions, f_1 and f_2. We induce noise into one of the functions, say $f_2^* = f_2 + \texttt{noise}$. We then computed the similarity between f_1 and f_2^*. Again, τ between f_1 and f_2 should be the same as f_1 and f_2^* (*i.e.* the difference should be 0). In this scenario, we found the mean difference in the topological similarity to be 6.42×10^{-5} and standard deviation to be 6.29×10^{-5}.

The final experiment considers the effect of noise to the locations of the extrema. We perturb the locations of the underlying graph (bounded by 10^{-4}), and compute the similarity between the different pairs of functions. We then measure the difference between τ before and after perturbation. The mean and standard deviation of the difference in τ in this case was 7.8×10^{-4} and 8×10^{-4} respectively. When looking at individual errors in all of the above experiments, we found that in several cases, there was no change in τ demonstrating the robustness of the measure to noise.

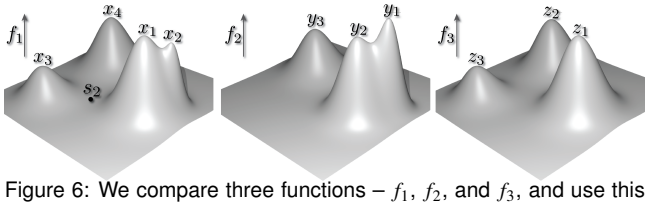


Figure 6: We compare three functions – f_1, f_2, and f_3, and use this comparison to demonstrate the visualization interface. f_1 and f_2 are the same functions as used in the earlier examples.

5 Exploration Framework

We design a visual interface to help ecologists explore multiple SDMs. We accomplish this through the use of multiple visualizations. The interface consists of 4 views.

5.1 Properties View

A matrix is used to represent various properties of different models, as well as the difference between pairs of models. The diagonal of this matrix represents the properties of the individual models. The functional distance ϕ between the pairs of models is represented in the upper triangular matrix, while the topological distance τ is represented in the lower triangular matrix. Fig. 7(a) shows the properties view for the three sample functions shown in Fig. 6. In case of functions f_1 and f_2, the difference is the presence of peak x_3 in f_1, which contributes to the topological similarity. In case of f_2 and f_3, peak equivalent to z_3 is missing in f_2, while a peak equivalent to y_1 is missing in f_3. However, the simplification required to remove z_3 is greater than that required for y_1, which is denoted by their topological similarity.

5.2 Features View

This view visualizes the topological features of the selected model(s) as a scatter plot. The choice of scatter plot was motivated by the simplicity of the persistence diagram and the fact that the ecologists were familiar with scatter plots. Each point in the scatter plot corresponds to a topological feature (maximum or minimum). The axes of the scatter plot are defined based on what the user wants to explore, and the maxima and minima are represented as upward pointing and downward pointing triangles respectively.

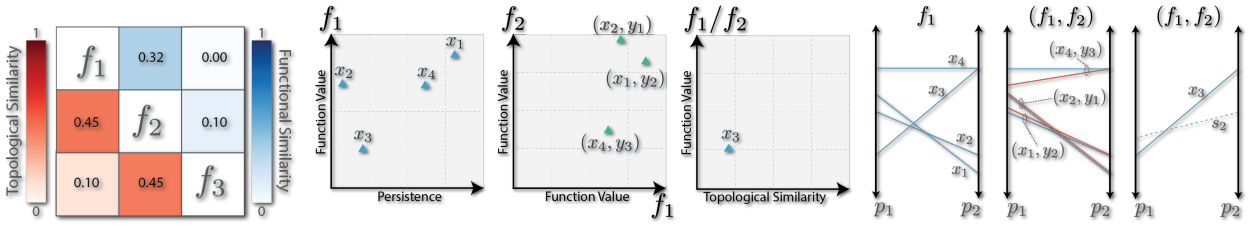
Explore single model. In this case, the x-axis of the model corresponds to the persistence (topological significance) of the extrema, while the y-axis corresponds to its function value. This allows the user to choose features during the exploration. For example, in case users are not interested in extrema with a small function value, then they can focus at the appropriate portion of the plot. The extrema of the function f_1 is shown in Fig. 7(b)-left.

Explore similarities between two models. In this case, each point in the scatter plot corresponds to a pair of extrema that are similar, that is, the pair of extrema that match. The axes corresponds to the function values of the two extrema. This view also provides the intuition for the functional similarity. A functionally similar pair of functions should have all points along the diagonal in this plot. Divergence from the diagonal denotes a disparity in the function values between the two functions in the parameter space in the neighborhood of the extreme points. Fig. 7(b)-middle illustrates the different matches found between f_1 and f_2 (also see Fig. 3(c)).

Explore differences between two models. In this case, each point in the scatter plot corresponds to an extremum that is present in one function but absent in the other. The color of the point denotes the function it is part of. The x-axis corresponds to the topological similarity measure, while the y-axis corresponds to the function value. Fig. 7(b)-right shows the difference between functions f_1 and f_2.

5.3 Parallel Coordinates View

Once features of interest are chosen, the spatial region in the domain corresponding to the selected features is visualized using the



(a) Properties view (b) Features view (c) Parallel coordinates view

Figure 7: **(a)** The properties view summarizes the topological similarity and functional similarity between the three sample functions (a lower value is better). **(b)** A scatter plot is used to denote the similar and dissimilar extrema of a given pair of functions. The maxima and minima are represented as upward and downward pointing triangles respectively. When exploring a single function (left), each point corresponds to an extremum of the function. Here, we show the set of maxima of f_1. When exploring the similarities between two functions (middle), each point corresponds to a pair of extrema that are matched. The figure shows the matching between the maxima of f_1 and f_2. When exploring differences between two functions (right), each point corresponds to an extremum that is present in one function, but not in the other. When comparing f_1 and f_2, the maximum x_3 is absent from f_2 **(c)** Parallel coordinates is used to represent the location of extrema of interest in the high dimensional predictor space. The points corresponding to the set of maxima of f_1 (left), the matched maxima between f_1 and f_2 (middle), and the maximum that is absent in f_2 (right) are shown. when exploring the differences, corresponding critical point pair (saddle s_2) is also shown.

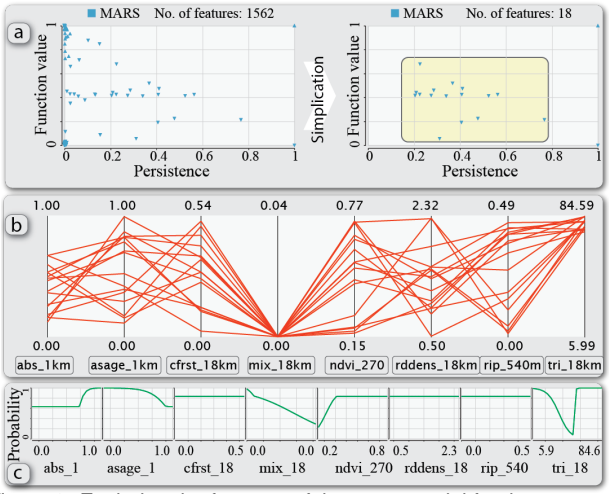


Figure 8: Exploring the features of the MARS model for the Brewers Sparrow data set. **(a)** Given the set of all extrema, the user simplifies to remove all those extrema having persistence less than 0.2. Note that this removes all maxima except the global maximum (at location $(1,1)$ in the scatter plot). **(b)** The locations of the selected set of minima of the MARS model are shown using the parallel coordinates view. **(c)** Note that it is difficult to grasp the presence of high persistent minima (deep valleys) using the default response curves that is common in the analysis of this data.

parallel coordinates view. This view provides information on the location of the selected extrema in the high dimensional space. Fig. 7(c)-left illustrates the locations of all maxima of the function f_1. Fig. 7(c)-middle and Fig. 7(c)-right show the matched maxima and the differing maximum respectively.

5.4 Response Curve View

As mentioned in Section 1, a response curve represents a one dimensional slice of the function. We include this view in our interface since it helps the ecologists understand the different features as they are familiar with this representation. By selecting a feature and a predictor of interest from the parallel coordinate view, the user can view the response curves with respect to the selected predictor. The values of the other predictors are set to those corresponding to the selected extremum. We also show the response curve of the critical point pair corresponding to an extremum. This helps users understand how the function changes. For example, when viewing a minimum-saddle pair, the upward movement of the response curve indicates the approximate shape of the corresponding "valley" in the high dimensional space.

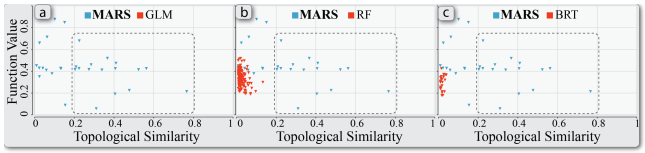


Figure 9: Comparing MARS with other models for the Brewers Sparrow species. Note that multiple significant minima that are present in MARS are not present in the other models. Also, these constitute the significant differences between these models.

6 Case Studies

In this section we describe two use case scenarios that are of interest to ecologists. The first case shows how the extrema in the different models can be used to guide ecologists towards interesting features of the model. The second case demonstrates how our similarity comparison technique can be used to identify differences between the models that are otherwise difficult to find. We use two 8-dimensional data sets, Brewers Sparrow and Sagebrush, for the experiments in this section [43](see Supplemental document for a description of the data).

Exploring an SDM. In this use case, the user is interested in exploring the properties of a single SDM. Using the visual interface, the user first selects the species and model algorithm of interest. In this experiment, the user chooses the MARS model for the Brewers Sparrow species [21]. Fig. 8(a) shows the set of extrema of this model. An initial simplification is performed to remove noise/less significant extrema. Note that for the MARS model, there are a high number of significant minima. Fig. 8(b) shows the different predictors corresponding to the set of selected minima. It is interesting to note that all of these minima occur when combination of values of mix_18km is low and tri_18km is high. Such a behavior is clearly not visible using the default response curves [44] shown in Fig. 8(c).

Exploring differences between given pair of models of a fixed Species. In this experiment, the user first selects the pair of models that are to be compared from the Properties view. The user can now view either the non-matched features or the matched features. The first experiment considers the differences between MARS and other models for the Brewers Sparrow species. Users can filter features (extrema) having low topological similarity. As shown in the previous use case, the MARS model for Brewers Sparrow contains a large number of significant minima. It can be seen than these minima do not match with any minima of the other models, i.e., there exists no minima in the other models in the corresponding locations. This is illustrated in Fig. 9 where we look at the different extrema in the features view. Let us now select a significant difference between GLM and MARS (having high value of τ). Fig. 10

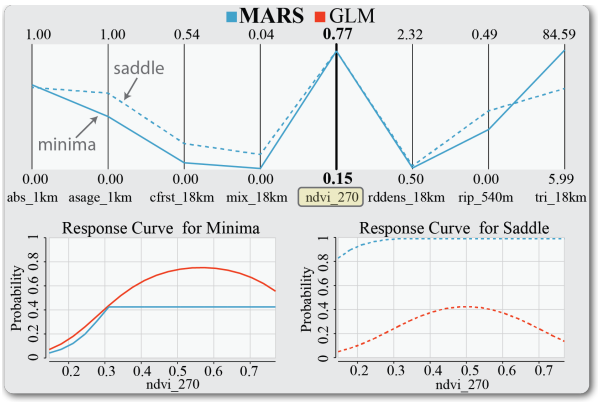


Figure 10: Locations of a significant minimum-saddle pair in MARS is shown using parallel coordinates. Note the moving up of the response curves of MARS from the minimum to the saddle. At the same location, we see a different behavior for the GLM model.

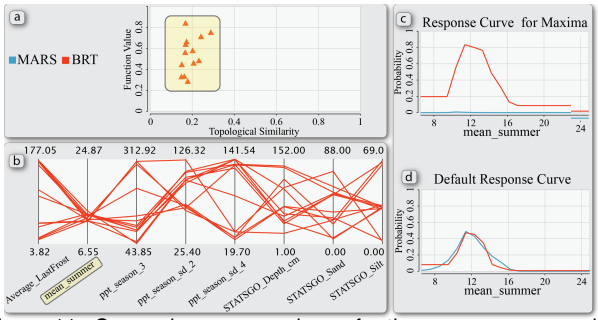


Figure 11: Comparing MARS and BRT for the Sagebrush species. **(a)** Selecting all significant maxima that are present in BRT but not in MARS. **(b)** Note that such difference mainly occurs at a relatively low value of the mean_summer predictor. **(c)** The response curve at one of the maximum. **(d)** This behavior is counter intuitive to the default response curve, in which we see both models having the same pattern.

shows the coordinates of the minimum-saddle pair (intuitively the lowest and highest point of the valley corresponding to the minimum) that is present in MARS, but absent in GLM. The response curves varying the predictor ndvi_270 at the minimum and saddle points shows a significant increase in the shape of the curve indicating a "valley"-like structure in MARS. However, we see a slight decrease in the response curve for GLM indicating the absence of a minimum in that region (and thus the difference).

In the next experiment, the user compares the differences between MARS and BRT for Sagebrush [13]. In particular, the user selects the set of significant maxima (having topological similarity > 0.15) in MARS that are not present in BRT (Fig. 11(a)). Fig. 11(b) shows the coordinates corresponding to these maxima. Note that all of these differences occur when the value of mean_summer is low. This is counter intuitive when one looks at the default response curves of these two models (Fig. 11(d)).

Feedback from Ecologists. When we initially provided our tool to the ecologists, they found the results to be a little too abstract and had difficulty in comprehending them. To help them get familiar and better understand the utility of working directly in the high dimensional space, we used a two dimensional slice of the different models, and setup the software to work with this data. Their familiarity with the features in low dimensions allowed them to relate to the results from our tool. Also, since they could easily visualize the 2D data, the different features were directly apparent.

An immediate advantage they found using our tool was in being able to see patterns that was not possible before. This utility is reflected in the following comment – *"the following is an example in which the 1D response curves were inadequate: With the spruce fir is that one of the predictors (mean_gt_38) has a very large point mass of presence at 0 and all other values for that predictor are associated with absence so if I look at response curves in SAHM holding all predictors constant at their mean then I get flat lines at 0 because as long as mean_gt_38 is at its mean value the predicted value is 0 no matter what the other predictors are."*

The examples presented in the previous section highlight the complexity of the response surface when considering an eight dimensional space (that is, using eight predictor variables) and clearly provide new information about the various models used by the ecologists. However, the implications of some of these results was not immediately apparent, which we plan to explore further in the future. As an ecologist collaborator mentioned during one of our interactions, *"looking at models this way is interesting for me. With our current tools for the example in Fig. 11, we wouldn't have otherwise known that it was at low values of mean_summer that the models differed and that might be of interest in an in depth study"*.

In some cases when there is a difference between two models, it is possible that this is due to missing data. In such cases ecologists would have to collect additional data from regions having the differences. So, such a tool can also help in identifying these regions of discrepancies.

7 Discussions and Conclusions

Discretization of a high dimensional function. Identifying an ideal sample size to represent a high dimensional function is a difficult problem. For all the experiments in this paper, we used a sample size of 10^5 points. We chose this size since we found that the similarity measure computed did not significantly change even on increasing the sample size to above 10^5. This is because increasing the sample size only created noisy extrema which did not affect the similarity measures.

Each dimension of an SDM corresponds to an environmental variable, which have a well defined range of values. Given this, the domain space of the SDM would correspond to the Euclidean subspace corresponding to these ranges. However, since the units of these variables differ, we normalize the ranges between 0 and 1 in order to be consistent. Alternatively it will be interesting to explore other possibilities, such as standardizing the inputs instead.

Neighborhood radius. The neighborhood radius used for weighing the edge weights of the bipartite graph is largely dependent on the application and domain expertise. We used a neighborhood radius $r = 0.1$ for this purpose, and was based on discussions with the ecologists, who did not want the matching features to be far away. Typically, with increasing r, the number of matches between similar (in terms of function value) extrema that are farther away would increase. This would potentially decrease the functional similarity between two models. In case of topological similarity, it could both increase as well as decrease. It would increase if the larger r causes a nearby but high persistent extrema that was earlier matched to not be matched. It could also decrease, since more matches are possible, thus causing less number of extrema to be simplified. We plan to investigate this further in the future, to be able to automatically identify an appropriate radius.

Conclusions. With the focus on helping ecologists better understand SDMs, we design a topology based framework that helps guide them towards interesting features of the model. We also propose the concept of maximum topology matching that can be used to identify similarities and differences between a given pair of SDMs. Even though the focus was on the ecology domain, our technique is general and can be applied in cases requiring a locality-aware way of comparing scalar functions. While we found that our similarity measures were stable under the influence of noise in practice, it is still possible to obtain discontinuities in rare cases. We plan to investigate this further and try to obtain a stable measure. It will also be interesting to consider topological splines as a visual

metaphor to represent SDMs, and using it to show the identified similarities and differences. We are currently working with our collaborators in an ecology related paper that further investigates the implications of the results obtained. In future, we are also interested in exploring the generality of our similarity technique.

Acknowledgments. This work was supported in part by a Google Faculty Award, an IBM Faculty Award, the Moore-Sloan Data Science Environment at NYU, the NYU School of Engineering, the NYU Center for Urban Science and Progress, AT&T, NSF award CNS-1229185, DOE, and the NASA Biodiversity Program award NNH11AS091. MT and JM's contribution was funded by the Department of the Interior North Central Climate Science Center. Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

REFERENCES

[1] P. K. Agarwal, H. Edelsbrunner, J. Harer, and Y. Wang. Extreme Elevation on a 2-manifold. *Disc. Comput. Geom.*, 36(4):553–572, 2006.

[2] U. Bauer, X. Ge, and Y. Wang. Measuring distance between reeb graphs. In *Proc. SOCG*, pages 464:464–464:473. ACM, 2014.

[3] K. Beketayev, D. Yeliussizov, D. Morozov, G. Weber, and B. Hamann. Measuring the distance between merge trees. In *TopoInVis*, pages 151–165. Springer, 2014.

[4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *CGF*, pages 911–920, 2011.

[5] S. Bergner, M. Sedlmair, T. Moller, S. Nabi Abdolyousefi, and A. Saad. Paraglide: Interactive parameter space partitioning for computer simulations. *IEEE TVCG*, 19(9):1499–1512, 2013.

[6] S. Bruckner and T. Möller. Isosurface similarity maps. *CGF*, 29(2):773–782, 2010.

[7] L. Buisson, W. Thuiller, N. Casajus, S. Lek, S., and G. Grenouillet. Uncertainty in ensemble forecasting of species distribution. *Global Change Biology*, 16(4):1145–1157, 2010.

[8] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In *Proc. SGP*, pages 124–135, 2004.

[9] H. Carr, J. Snoeyink, and U. Axen. Computing Contour Trees in All Dimensions. *Comput. Geom. Theory Appl.*, 24(2):75–94, 2003.

[10] A. Cheaib, V. Badeau, J. Boe, I. Chuine, C. Delire, E. Dufrêne, C. François, E. S. Gritti, M. Legay, C. Pagé, et al. Climate change impacts on tree ranges: model intercomparison facilitates understanding and quantification of uncertainty. *Ecology letters*, 15(6):533–544, 2012.

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

[12] C. Correa, P. Lindstrom, and P.-T. Bremer. Topological spines: A structure-preserving visual representation of scalar fields. *IEEE TVCG*, 17(12):1842–1851, 2011.

[13] K. Decker and M. Fink. Colorado wildlife action plan enhancement: Climate change vulnerability assessment. 2014.

[14] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge Univ. Press, England, 2001.

[15] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., Providence, Rhode Island, 2009.

[16] J. Elith and J. R. Leathwick. Species distribution models: Ecological explanation and prediction across space and time. *Annu. Rev. Ecol. Evol. Syst.*, 40(1):677–697, 2009.

[17] J. Elith, J. R. Leathwick, and T. Hastie. A working guide to boosted regression trees. *Journal of Animal Ecology*, 77(4):802–813, 2008.

[18] W. Feng, J. Huang, T. Ju, and H. Bao. Feature correspondences using morse smale complex. *The Visual Computer*, 29(1):53–67, 2013.

[19] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, 1987.

[20] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE TVCG*, 16(6):1271–1280, 2010.

[21] S. E. Hanser, M. Leu, S. T. Knick, and C. L. Aldridge. *Sagebrush ecosystem conservation and management: ecoregional assessment tools and models for the Wyoming Basins*. Allen Press.

[22] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. SIGGRAPH*, pages 203–212, 2001.

[23] T. Keenan, J. Maria Serra, F. Lloret, M. Ninyerola, and S. Sabate. Predicting the future of forests in the mediterranean under climate change, with niche-and process-based models: Co2 matters! *Global Change Biology*, 17(1):565–579, 2011.

[24] K. Matkovic, D. Gracanin, M. Jelovic, and H. Hauser. Interactive visual steering - rapid visual prototyping of a common rail injection system. *IEEE TVCG*, 14(6):1699–1706, 2008.

[25] K. Matkovic, D. Gracanin, B. Klarin, and H. Hauser. Interactive visual analysis of complex scientific data as families of data surfaces. *IEEE TVCG*, 15(6):1351–1358, 2009.

[26] J. Milnor. *Morse Theory*. Princeton Univ. Press, New Jersey, 1963.

[27] D. Morozov, K. Beketayev, and G. Weber. Interleaving distance between merge trees. In *TopoInVis*, 2013.

[28] T. Muhlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE TVCG*, 19(12):1962–1971, 2013.

[29] V. Narayanan, D. M. Thomas, and V. Natarajan. Distance between extremum graphs. In *Proc. PacificVis*, 2015.

[30] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. Visualization of high dimensional point clouds using their density distribution's topology. *IEEE TVCG*, 99(11):1547–1559, 2011.

[31] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. The TOPORRERY: computation and presentation of multi-resolution topology. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 19–40. 2009.

[32] H. M. Pereira, P. W. Leadley, V. Proença, R. Alkemade, J. P. Scharlemann, J. F. Fernandez-Manjarrés, M. B. Araújo, P. Balvanera, R. Biggs, W. W. Cheung, et al. Scenarios for global biodiversity in the 21st century. *Science*, 330(6010):1496–1501, 2010.

[33] H. Piringer, W. Berger, and J. Krasser. Hypermoval: Interactive visual validation of regression models for real-time simulation. *CGF*, 29(3):983–992, 2010.

[34] J. Poco, A. Dasgupta, Y. Wei, W. Hargrove, C. Schwalm, R. Cook, E. Bertini, and C. Silva. SimilarityExplorer: A visual inter-comparison tool for multifaceted climate data. *CGF*, 33(3):341–350, 2014.

[35] H. Saikia, H.-P. Seidel, and T. Weinkauf. Extended branch decomposition graphs: Structural comparison of scalar data. *CGF*, 33(3):41–50, 2014.

[36] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE TVCG*, 14(6):1475–1482, 2008.

[37] D. Thomas and V. Natarajan. Symmetry in scalar field topology. *IEEE TVCG*, 17(12):2035–2044, 2011.

[38] D. Thomas and V. Natarajan. Detecting symmetry in scalar fields using augmented extremum graphs. *IEEE TVCG*, 19(12):2663–2672, 2013.

[39] D. Thomas and V. Natarajan. Multiscale symmetry detection in scalar fields by clustering contours. *IEEE TVCG*, 20(12):2427–2436, 2014.

[40] W. Thuiller. Patterns and uncertainties of species' range shifts under climate change. *Global Change Biology*, 10(12):2020–2027, 2004.

[41] T. Torsney-Weir, A. Saad, T. Moller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE TVCG*, 17(12):1892–1901, 2011.

[42] G. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE TVCG*, 13(6):1416–1423, 2007.

[43] N. Young. Tutorial for the Software for Assisted Habitat Modeling (SAHM) package in VisTrails. Technical report, US Geological Survey, Fort Collins Science Center, 2012.

[44] D. Zurell, J. Elith, and B. Schröder. Predicting to new environments: tools for visualizing model behaviour and impacts on mapped distributions. *Diversity and Distributions*, 18(6):628–634, 2012.