

## Journal Pre-proof

Assessing feature scorer results on high-dimensional datasets with t-SNE

Bruno Iochins Grisci, Mario Inostroza-Ponta, Márcio Dorn

PII: S0925-2312(25)01233-0

DOI: <https://doi.org/10.1016/j.neucom.2025.130561>

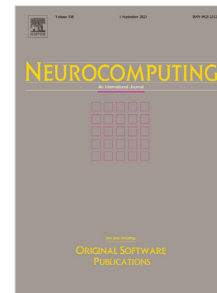
Reference: NEUCOM 130561

To appear in: *Neurocomputing*

Received date: 30 November 2023

Revised date: 18 February 2025

Accepted date: 21 May 2025



Please cite this article as: B.I. Grisci, M. Inostroza-Ponta and M. Dorn, Assessing feature scorer results on high-dimensional datasets with t-SNE, *Neurocomputing* (2025), doi: <https://doi.org/10.1016/j.neucom.2025.130561>.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2025 Published by Elsevier B.V.

# Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

Bruno Iochins Grisci<sup>a</sup>, Mario Inostroza-Ponta<sup>b</sup> and Márcio Dorn<sup>a,c,d,\*</sup>

<sup>a</sup>Institute of Informatics, Federal University of Rio Grande do Sul, Av. Bento Gonçalves 9500, Porto Alegre, 91501-970, RS, Brazil

<sup>b</sup>Computer Engineering Department, University of Santiago de Chile, Av. Libertador Bernardo O'Higgins, Santiago, 9170022, RM, Chile

<sup>c</sup>Center of Biotechnology, Federal University of Rio Grande do Sul, Av. Bento Gonçalves, Porto Alegre, 91501-970, RS, Brazil

<sup>d</sup>National Institute of Science and Technology - Forensic Science, Porto Alegre, RS, Brazil

## ARTICLE INFO

### Keywords:

dimensionality reduction  
data visualization  
feature scoring  
machine learning interpretability

## ABSTRACT


While vast literature on high-dimensional data visualization is available, there are not many works regarding the visualization of feature scorers and their results. Feature scorers are algorithms that assign numerical importance to each feature of multi-dimensional datasets. These importance scores can be used in several applications, such as feature selection, knowledge discovery, and machine learning interpretability. There are several feature scorers to choose from, and often no single metric or ground truth is available to guarantee the quality of their results. In this scenario, visualization can become valuable to support the decision of which method to choose and how good its results are. For this goal, this work presents “weighted t-SNE.” It modifies the relationship between data points in the embedded 2D space to reflect the importance of each dimension of the original datasets as assessed by a feature scorer. This research discusses how to implement weighted t-SNE, proposes the silhouette coefficient as a numerical evaluation of the results, and shows several examples of its use in practice. Synthetic and real-world tabular datasets are used in the experiments together with nine feature scorers, ranging from Mutual Information to neural networks. Each feature scorer produces unique visualizations, and weighted t-SNE can be used to compare and choose the one that better suits a given dataset and task. Weighted t-SNE can also visually show the importance of features learned by machine learning models and help us see how they are organizing the data, increasing their interpretability.


## 1. Introduction

High-dimensional data refers to datasets whose dimensionality is comparable to or even higher than their sample size. It is often present in tables containing information regarding the sciences or business [1]. Some examples are data of genes expression from cancer patients [2, 3], SNPs in forensic science [4], COVID-19 patients' hemogram exams [5, 6, 7], astrophysics [8], and sales from e-commerce [9]. One of the key aspects of those large tabular datasets is the presence of dozens to millions of columns, each representing a *feature*, *input*, or *dimension* of the data being studied [10].

Data analyses such as clustering, classification, regression, or outlier detection rely first on the transformation, selection, and removal of features [1]. Each feature represents one attribute of a data sample or instance. Dimensionality reduction and feature selection allow large datasets to become manageable by keeping only the relevant information, making the data more easily understandable, and freeing storage space. Feature scorers are a large and diverse group of algorithms that play a crucial role in this process. They use distinct strategies, metrics, and criteria to determine the importance score of each feature. Once obtained, these scores are numerical values that can be used to filter the data or even for knowledge discovery. Because several machine learning models can output feature importance as well, these scores can even help with machine learning interpretability [10, 11, 12]. It is important to note that the feature importance is always concerning a specific task and data, and it should not be assumed that this value will be the same if the context for the feature changes.

\*Corresponding author

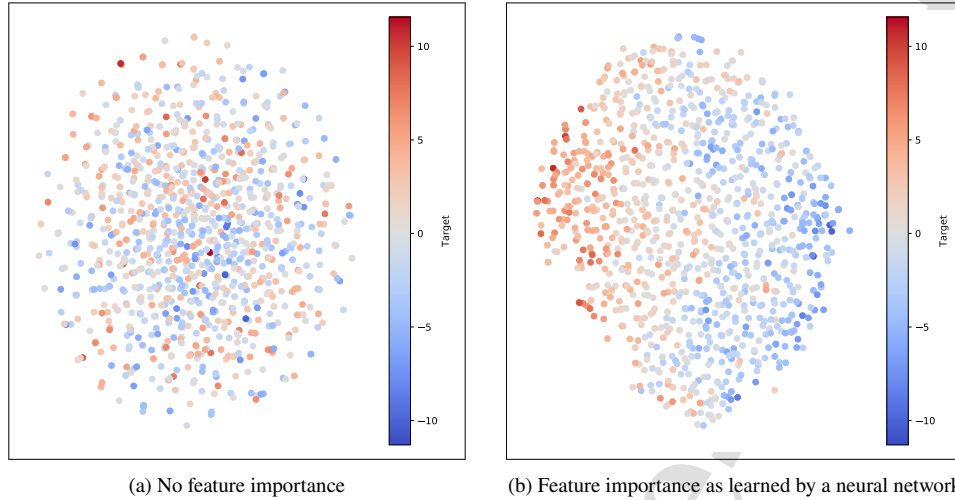
 bigrisci@inf.ufrgs.br (B.I. Grisci); mario.inostroza@usach.cl (M. Inostroza-Ponta); mdorn@inf.ufrgs.br (M. Dorn)

 <https://brunogrisci.github.io/> (B.I. Grisci); <https://informatica.usach.cl/academico/mario-inostroza-ponta/>

(M. Inostroza-Ponta); <https://sbc.inf.ufrgs.br/> (M. Dorn)

ORCID(s): 0000-0003-4083-5881 (B.I. Grisci); 0000-0003-1295-8972 (M. Inostroza-Ponta); 0000-0001-8534-3480 (M. Dorn)

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



**Figure 1: Comparison between the visualization using weighted t-SNE of a synthetic regression dataset before and after considering the feature importance as learned by a neural network.** Each figure is the visualization of the 2D embedding of a regression dataset (negative target values in blue, positive values in red) composed mostly of irrelevant features. On the left is the regular visualization, in which the samples are mixed. On the right, it is possible to visualize how the feature importance learned by a neural network disentangles the samples. This experiment is further detailed in Section 4. Interactive versions of all the plots in this publication are available at <https://sbcblab.github.io/wtsne/>.

Since there are several feature scorers and no single metric guarantees the quality of the results, visualization becomes a valuable tool to help in the method selection and show how good the results are. An inadequate set of scores can harm the subsequent steps of data analysis. Moreover, the visualization of a low-dimensional embedding of the high-dimensional data using methods like the t-Distributed Stochastic Neighbor Embedding (t-SNE) [13] is already an established aspect of exploratory data analysis [1]. Visual cluster analysis often employs these techniques to project the high-dimensional data into 2D scatterplots, assuming that the projection faithfully matches the actual clusters [14].

In this scenario, visualizing the clusters affected by the feature importance scores in a 2D scatterplot would be a valuable tool for comparing feature scorers. The algorithms with better results (a more significant and representative set of scores) would lead to better clusters in the projection. However, due to the challenges of visualizing high-dimensional tabular data and the algorithms being overlooked as a pre-processing step, they are usually applied as a black-box at the beginning of an analysis [15]. In the vast literature on high-dimensional data visualization, few works are interested in displaying the effects of feature scoring [10]. Most of the research interest is in the evaluation of feature selection [16, 17, 18] or the use of visualization to allow the user to interact with the selection process [19, 20, 21].

To tackle this problem, we propose an extension of t-SNE to visualize the results from feature scoring algorithms: *weighted t-SNE*. It modifies the relationship between data points to reflect the importance of each dimension. The desired outcome is that relevant features have more influence on the position of the points in the final projection. Weighted t-SNE can be used as an inspection tool to compare and choose feature scorers in different datasets. Additionally, combined with machine learning, it could better represent the patterns learned by the models. Fig. 1 shows an example of this application. In Fig. 1a, a regular projection is made with t-SNE of a synthetic dataset. After training a neural network to perform regression on this dataset, the importance score of each feature was extracted from the model and used to create the projection shown in Fig. 1b. It is possible to see how the feature importance learned by the neural network sorted the samples by comparing the two scatterplots. This work focuses on continuous or numerical features, which can also be ordinal or categorical. The requirement for running t-SNE on the data is that it must be meaningful. For instance, a suitable distance measure for the high-dimensional space must exist. This requirement is often not reached if the feature types are mixed.

A preliminary version of weighted t-SNE appeared in the research by Grisci et al. [10]. It focused in *relevance aggregation*, a method for neural network interpretability. In that study, authors used weighted t-SNE to visualize the

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

patterns learned by the networks trained on tabular data. Despite those results showing the potential of this kind of visualization, Grisci et al. [10] only used weighted t-SNE to validate the main method, relevance aggregation. Junior and Lopes [22] and Zhao et al. [23] later used weighted t-SNE to visualize neural network results for absenteeism prediction and to evaluate data transformation techniques on software defect prediction models, respectively. Lu and Yan [24] and Ma et al. [25] also explored the idea of scaling features for t-SNE visualizations, albeit for distinct purposes. The present research presents a thorough description, critical analysis, and discussion of an updated weighted t-SNE, together with a large set of experiments on several datasets and feature scorers.

The remainder of this paper has the following organization. Section 2 presents and discusses feature scorers, related visualization methods, and the silhouette coefficient metric. Section 3 describes the proposed weighted t-SNE. Section 4 details the experiments and results, as well as a discussion on machine learning interpretability. Finally, Section 5 concludes this work.

### Statement of significance

Understanding the effectiveness of feature scorers is a fundamental challenge in machine learning and data science. While prediction accuracy is commonly used to evaluate feature scoring and selection, it does not fully capture how feature importance influences data representation. Visualization techniques such as t-SNE provide a powerful tool for exploring high-dimensional data, yet they do not inherently account for feature relevance. This paper introduces weighted t-SNE, a novel extension that integrates feature importance scores into the dimensionality reduction process, enabling direct inspection of how different scorers structure data.

**Problem:** Selecting the best feature scorer is often difficult due to the lack of ground truth for feature importance. Existing evaluation methods typically rely on classification or regression performance, but these metrics do not reveal how well a scorer identifies relevant features. Additionally, feature scoring is distinct from feature selection. While feature selection reduces dimensionality by removing features, feature scoring assigns relative importance to all features, and this process has received less attention from visualization tools. Without proper inspection methods, irrelevant or misleading feature scores may negatively impact downstream tasks, such as model interpretability, classification, or regression. For high-dimensional data, the manual inspection of thousands of numerical values is infeasible. Moreover, two models with identical predictive accuracy may assign different importance scores to features, making numerical comparison difficult, a challenge that weighted t-SNE helps resolve by providing an intuitive visual representation.

**What is already known:** Traditional t-SNE or similar tools like UMAP are widely used for exploratory data analysis, often assuming that clusters in the visualization correspond to meaningful patterns in the data. However, it does not incorporate feature importance, making it unsuitable for evaluating feature scorers. Visualization techniques for feature selection exist, such as RadViz, Self-Organizing Maps (SOM), and SmartStripes, but they focus on selecting a subset of features rather than analyzing the distribution of importance scores across all dimensions. Machine learning interpretability tools such as SHAP and LIME provide insights into model decisions but do not offer direct visualization of feature scorers across an entire dataset.

**What this paper adds:** Weighted t-SNE is a visualization tool specifically designed for analyzing feature scorers, bridging the gap between feature scoring and visual interpretability. It allows users to visually assess how different scoring methods affect data representation. It can be used in labeled datasets, providing insight into how feature importance influences different types of machine learning models. Beyond classification accuracy, weighted t-SNE has multiple practical applications: It helps revealing outliers in high-dimensional datasets where irrelevant features might otherwise obscure them. It visually tracks how a model transforms feature representations internally, making it easier to analyze learned feature importance compared to raw numerical scores. Weighted t-SNE is a valuable tool for machine learning and data science pipelines. By allowing visualization of feature scorers, it helps users select the most suitable scorer for a given dataset, ultimately improving downstream tasks such as feature selection, classification, or regression. By incorporating feature importance into dimensionality reduction, weighted t-SNE provides a new perspective on feature scoring, offering insights that neither accuracy-based evaluation nor traditional visualization methods provide.

## 2. Related work

### 2.1. Feature scoring

Feature scorers are algorithms or methods that, given a dataset, assign to each of its features (dimensions) a score measuring its importance or relevance based on specific criteria or metrics. For example, given a tabular dataset with two classes and  $n$  features, a standard feature scorer would return  $n$  scores corresponding to each dimension of the dataset in a way that those features with larger scores are better at distinguishing the two classes. These importance scores can be used in several applications, such as dimensionality reduction, feature selection, knowledge discovery, outlier detection, and machine learning interpretability [26, 27].

Because there is a large overlap in methods and application, it is important to distinguish between feature scoring and feature selection as two related but distinct tasks:

- **Feature scoring:** This task focuses on assigning numerical importance scores to features based on their relevance to a specific task (e.g., classification, regression, statistical significance, domain relevance). The scores provide a quantitative measure of how useful or relevant each feature is, but they do not directly reduce the dimensionality of the dataset. Feature scoring allows the comparison of features and enables feature ranking and feature selection.
- **Feature selection:** This task involves using the importance scores from feature scoring (or other selection criteria) to select a subset of features for downstream tasks. Typically, a threshold is applied to the scores, and features below the threshold are removed from the dataset. The goal of feature selection is to reduce dimensionality, improve computational efficiency, and enhance the interpretability of the data. Usually, after feature selection the features are used as equally important.

For example, if the features are ranked by their scores, a threshold can be set to remove all features below it from the dataset. After the reduction, the remaining features will ideally be more informative for training predictors, can be more easily analyzed by humans, or take less processing time and memory space for algorithms. Moreover, knowing which features are more important and which are less according to given criteria can be exploited for analyzing and interpreting the data [28, 10]. Further differences between feature scoring and feature selection are discussed in the experiments of Section 4.

A feature on its own may seem irrelevant, but when combined with other features, it could become highly relevant [29, 27]. Ideally, the features with high scores should be strongly relevant, but sometimes they can be weakly relevant if non-redundant features are helpful in improving the predictions. Meanwhile, the irrelevant, redundant, or noisy features receive lower scores [27]. Feature scorers are considered *univariate* if each feature is analyzed independently, and *multivariate* if the interactions and correlations between features are taken into account. These methods are usually divided in the five groups presented below [27, 26] with examples that will be used in the experiments of Section 4.

**Filter:** these methods take into account the inherent characteristics of the data in conjunction with evaluation criteria (information, distance, consistency, dependency, etc.), and are not limited to being classifiers. Most filters treat the problem as a ranking problem and are univariate. They do not rely on specific learning algorithms, providing more general solutions that different classifiers can use. Filters are faster and more computationally efficient than the other groups of selectors, but they may ignore relationships between different features or the effects that they have when combined [26, 27]. An example is the Kruskal-Wallis one-way analysis of variance [30], a non-parametric statistical test for discovering if samples originate from the same distribution. The Kruskal-Wallis test can be used as a simple filter feature scorer. In this case, every feature is individually tested to check if it belongs to the same distribution for the different groups (or classes), and their scores come from the statistical significance (the more certain of a feature belonging to different distributions for each group, the larger its importance) [28, 10]. Another example is the Minimum Redundancy Maximum Relevance (mRMR) [31], a filter algorithm that iteratively selects the features that maximize mutual information to the target class and minimize the redundancy regarding all the features previously selected. Mutual Information is a measure of how much information one random variable has in relation to another variable [32]. It is possible to quantify a feature's relevance based on how much information it holds with respect to the target class. As a final example, ReliefF scores each feature according to how different they are from nearby instances [33]. For all samples in a dataset, it looks for the  $k$ -neighbor samples of each class and then weights how much each feature differs between samples, thus being an efficient algorithm that considers feature-correlation [34].

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

**Wrapper:** the scoring is made using some optimization algorithm, often a metaheuristic, and then wrapping a classifier (or regressor) around the selected features [27]. The accuracy is the criteria of evaluation. The set of most discriminating features is found by minimizing the classification error, which often results in better accuracy than filters. However, wrappers are highly dependent on the learning algorithm being used as a classifier, so the solutions do not generalize. There is no guarantee that the quality of performance of the selected features will be transferable for other classifiers. Wrappers are more likely to suffer from overfitting and to present huge computational costs since the training of the classifier needs to be performed for each new subset being evaluated [26, 27].

**Embedded:** the scoring process is integrated within the learning algorithm and is conducted simultaneously with the classification (or regression). This approach is more efficient than wrappers because it avoids the repetition of training a classifier and is less prone to overfitting while achieving similar performance. Despite this, the computational complexity in high-dimensional data remains a challenge. [27]. Examples of a simple embedded algorithm are the least absolute shrinkage and selection operator (Lasso) [35] and decision trees, but even support-vector machines (SVM) and neural networks can be treated as embedded scorers. In these cases, some metric is used to evaluate the impact of each feature inside the model. This idea is further discussed in Subsection 4.1.

**Ensemble:** multiple classifiers or regressors are trained and their predictions are combined to achieve a better result. A popular example is the random forest [36], which is an ensemble of decision trees.

**Hybrid:** a combination of different methods (which may or may not belong to the same group), different scorer algorithms, or different criteria, in an attempt to capitalize on their unique strengths. The most common combination is that of wrappers and filters [27].

Unlike filters, wrapper and embedded methods rely on the accuracy of a classifier during evaluation, while also utilizing strategies to search the feature space to complete the selection process [26]. Hybrid methods usually have the best results, as they combine the strengths of the other approaches, decrease the computational costs narrowing the search space, and reduce overfitting [27]. Feature scorers are usually global in that the features' importance is computed over the entire data space [37]. If the importance must be computed locally (different subspaces of the data have different feature scores), an alternative method is subspace clustering [37, 38]. More details on how particular feature scorers work, including mathematical formulation, are present in the **Supplementary material**.

Feature scoring can be applied to several domains, some of them listed in Section 1. One of the most critical and prevalent applications of feature scorers is in gene selection. Due to its relevance, it will be described in more detail in this section and used in some experiments in Section 4. Feature selection is applied to genomic data (a popular example being gene expression from microarray experiments) to discover subsets of genes capable of separating samples from different populations [39, 2, 40, 41]. Genomic data can be effectively used for reliable cancer diagnosis, prognosis, or clinical treatment, but it often contains irrelevant, redundant, or noisy values [42, 27]. The discovery of genes capable of differentiating samples from different target annotations (the samples' classes) is an essential aspect in the analysis of microarray data [26]. These informative genes are used in the identification of diseases or as potential drugs targets [26]. In contrast to regular feature selection, the elimination of redundant features can lose informative genes with highly correlated expressions in gene selection.

Most gene selection studies are focused on filters due to their efficiency and generability [27]. Nevertheless, many challenges persist, such as the presence of technical defects in the experiments [27], class imbalance [43, 26], data bias [44], stability [45], and the difference between the several analyses standards. More importantly, retrieving domain-specific information from the data is not an easy task, and determining the relevance or redundancy of a feature is difficult, leading to unexpected biases and mistakes in conclusions. Despite a large number of methods available, there is still much room for innovations and improvements [27, 41, 28, 45]. Gene selection is an open problem with many challenges and new alternatives rising, with several methods for differentially expressed genes discovery being only slightly different among them [26, 46]. There is also the need for visualization to make sense of clusters of samples in these large high-dimensional datasets [47, 48, 49, 10]. In this context, creating analytics tools for the comparison and inspection of feature scorers is fundamental.



## 2.2. Visualization methods

Several algorithms are used to visualize multi-dimensional data in the 2D or 3D spaces. One of the most common approaches is to compute the Principal Component Analysis (PCA) of the data and display the first components [50]. The PCA extracts the principal components of the data through the covariance between dimensions so that each component is a linear transformation of the original dimensions that maximizes the variance information. Unlike t-SNE, PCA is deterministic and tends to preserve the distance of the points in the lower dimensions, keeping the global structure of the data but losing its local structure. PCA also cannot faithfully represent non-linearity in the data, which would defeat the purpose of using it to visualize feature selectors that are non-linear. As discussed by Kobak and Berens [47], PCA can be used to initialize the positions of the points for the t-SNE.

A more recent option to t-SNE is the Uniform Manifold Approximation and Projection (UMAP) [51]. Previous experiments conducted by Xia et al. [14] showed that UMAP and t-SNE are the best dimensionality reduction methods for cluster identification and membership identification. UMAP uses a sampling-based approach to optimize repulsive forces between data points [47]. While UMAP is said to be faster than regular t-SNE, it was observed that it is slower than optimized implementations of t-SNE and does not present better results for transcriptomics data [47].

A distinct approach for non-linear multi-dimensional 2D visualization is RadViz [52]. This algorithm places each feature as equally spaced positions around the perimeter of a circle to serve as “anchor points.” The data samples are represented as points inside the circle, their positions determined by the values of their features. Each data point is connected to the anchor points as if by a spring whose stiffness is proportional to the value of the corresponding feature (scaled between zero and one). The final data point is in the equilibrium position for all its springs, and points close to a variable anchor have a higher value than for the other variables. RadViz has been used to visualize gene expression data [49] and has also been expanded into attribute-RadViz [20] as a tool for user-guided feature selection. One disadvantage of RadViz is that its complexity grows and clarity decreases as the number of dimensions to be represented increases, which is an impediment in the use-cases with thousands of features present in Section 4.

Other methods of user-guided feature selection are the FDive [19], which uses Self-Organizing Maps (SOM) [53] to determine feature relevance, and the SmartStripes [15], which uses a color data table to convey information about the features. Once again, although the topics are closely related, the focus of the present work is the visualization of feature scoring, not feature selection. Another approach that resembles the results of SmartStripes but was developed to show the importance scores of features is the table heatmap [10]. In this approach, the original dataset is displayed, and each cell is colored with hue value-intensity proportional to the feature importance. However, table heatmap is unfeasible for large datasets.

## 2.3. t-Distributed Stochastic Neighbor Embedding

The t-Distributed Stochastic Neighbor Embedding (t-SNE) [13] is one of the most popular algorithms for visualizing high-dimensional datasets. It is a non-linear dimensionality reduction technique that allows the embedding of high dimensions in a much smaller space, usually 2D, to be used in plots. The core idea of t-SNE is to use methods from machine learning to optimize the embedding in a way that preserves the local neighborhood of points. Because of that, the local structure of points clustered together is usually kept in the embedding space. However, the global structure of the dataset can be lost. It is important to note that in t-SNE the low-dimensional embedding space does not necessarily represent any meaningful dimension of the original space, which is different from other methods, some of them, like PCA, described in Subsection 2.2.

The algorithm is now presented in more detail, based on the definitions of Maaten and Hinton [13] and Kobak and Berens [47]. Several practical improvements were proposed in the past years that made t-SNE more efficient regarding memory consumption and computing time [47, 54, 55, 56, 57]. Considering that we desire to visualize a  $N$ -dimensional dataset  $\mathbf{X} \in \mathbb{R}^N$ , t-SNE will create a lower dimensional embedding  $\mathbf{Y} \in \mathbb{R}^n$ , in which  $n \ll N$ . The desired result is that two points  $\chi_i, \chi_j \in \mathbf{X}$  that are close to each other should be represented by lower-dimensional points  $\gamma_i, \gamma_j \in \mathbf{Y}$  that are also close to each other. The similarities between the original and the embedding space points can be modeled as probability densities. Similarities in  $\mathbf{X}$  can be computed by a Gaussian distribution as shown in Equation 1. These are conditional probabilities that can be symmetrized to obtain joint probabilities  $p_{ij}$  as in Equation 2.

$$p_{j|i} = \frac{\exp(-\|\chi_i - \chi_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\chi_i - \chi_k\|^2 / 2\sigma_i^2)} \quad (1)$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2} \quad (2)$$

The similarities in the embedding space  $\mathbf{Y}$  are computed similarly. However, instead of the Gaussian distribution, the Student's t-distribution is used, as shown in Equation 3. This change allows some distances to be less faithfully preserved in the embedding thanks to the fatter tails of the t-distribution. This is needed because there is less available space in the embedding (because there are fewer dimensions), so without this freedom from the fatter tails, the points would end up crowded together.

$$q_{ij} = \frac{(1 + \|\gamma_i - \gamma_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\gamma_k - \gamma_l\|^2)^{-1}} \quad (3)$$

Now that the distributions  $\mathbf{P}$  and  $\mathbf{Q}$  are defined from  $p_{ij}$  and  $q_{ij}$ , the goal of t-SNE is to make  $\mathbf{P}$  and  $\mathbf{Q}$  as similar to each other as possible. The Kullback–Leibler (KL) divergence is an alternative to measure the similarity between two probability distributions, shown in Equation 4. The divergence can be seen as a cost  $KL(\mathbf{P}||\mathbf{Q})$  to be minimized. Because it is differentiable, it can be optimized with gradient descent.

$$KL(\mathbf{P}||\mathbf{Q}) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (4)$$

The Gaussian kernels centered over each data point in  $\mathbf{X}$  have a bandwidth  $\sigma_i$  (Equation 1). As the data density varies, each data point has its optimal value for  $\sigma_i$ . For instance, a larger value of  $\sigma_i$  will be adequate in sparser regions. These values are defined as the perplexity value, and the Gaussian kernels fit the nearest neighbors within one standard deviation of the probability density. The perplexity is a tunable parameter, as the user should determine how many close neighbors each point has. When it was first proposed, it was stated that the performance of t-SNE is robust to changes in the perplexity between the typical values of 5 and 50 [13]. However, the choice of perplexity and number of iterations during optimization can significantly impact the result of t-SNE [58]. Another point raised is that there is no guarantee that distances between well-separated clusters are meaningful without fine-tuning the perplexity. A rule-of-thumb that resulted in good empirical results is to set the perplexity to  $\max(30, m/100)$ ,  $m$  being the number of samples in the dataset [47]. This work employs this strategy to avoid choosing an arbitrary perplexity. One issue of SNE-like methods is that information on scales beyond the chosen perplexity tends to be excluded. One possible solution is using generalized similarities, computed as averages of softmax ratios with varying bandwidths, encompassing all neighborhood sizes. The inclusion of different scales causes a slight increase in computational complexity, but these novel multi-scale similarities do not require additional parameters [59].

Some works have already used the t-SNE as a feature extractor for several applications [16, 17, 18]. In those applications, the features of the low-dimensional embeddings obtained from t-SNE are used instead of the original features from the high-dimensional data [16]. The t-SNE, among other methods, can also be straightforwardly used to visualize feature selection results. This requires the creation of two plots, one with the complete set of features and another with only the subset of selected features [60]. These are not the goals of the present work, which uses t-SNE to visualize the impact of feature importance scores and not to create the scores themselves or to only show a subset of features with no regard to their importance. The difference will be further discussed in Section 4.

#### 2.4. Silhouette coefficient

The silhouette coefficient is a metric used to indicate the separation between clusters of points [61]. The silhouette measures how close each point in a cluster is to the points in the nearest clusters. Its range is  $[-1.0, 1.0]$ . A value close to 1 indicates that the point is far from other clusters and is well matched to its assigned cluster, a value of 0 suggests the point is near the decision boundary between clusters, and a value close to  $-1$  indicates that the point is closer to some other cluster than the one it is assigned and may have been wrongly assigned to its cluster. The silhouette coefficient is the average silhouette over all samples.

The silhouette coefficient in a visualization of a dataset can be used as an approximation on how well distinct classes are separated in the final plot [20]. In this case, the silhouette coefficient is computed using the actual class labels of



## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

each data point. For instance, if the points in the low-dimensional embedding space have a silhouette coefficient close to zero or less, this embedding failed to divide the classes. Of course, for this analysis to make sense, it always needs to consider the silhouette coefficient of the original high-dimensional space, as the embedding is not expected to segregate classes that do not have clear boundaries in the original space.

Considering that there are two or more clusters and that  $\phi \in C_i$  is a data point assigned to the cluster  $C_i$ ,  $a(\phi)$  is, as defined in Equation 5, the mean distance between  $\phi$  and every other data point in  $C_i$ .  $|C_i|$  is the number of points in  $C_i$  and  $d(\phi, \psi)$  is the distance  $d(\cdot)$  between the points  $\phi, \psi \in C_i$ . The value  $a(\phi)$  measures how well  $\phi$  belongs to  $C_i$  (the smaller the value the better).

$$a(\phi) = \frac{1}{|C_i| - 1} \sum_{\psi \in C_i, \phi \neq \psi} d(\phi, \psi) \quad (5)$$

Analogously, the value  $b(\phi)$ , defined in Equation 6, is the smallest mean distance between the point  $\phi$  and all points in a cluster  $C_j \neq C_i$ . The value of  $b(\phi)$  measures the dissimilarity of  $\phi$  to all points from other clusters.

$$b(\phi) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{\chi \in C_j} d(\phi, \chi) \quad (6)$$

The silhouette of the point  $\phi \in C_i$ , called  $s(\phi)$ , is computed combining the values  $a(\phi)$  and  $b(\phi)$  as in Equation 7. The  $s(\phi)$  will be close to 1 when  $a(\phi) \ll b(\phi)$ , meaning that  $\phi$  has a low dissimilarity to its own cluster (small  $a(\phi)$ ) and a high dissimilarity to the other clusters (large  $b(\phi)$ ).

$$s(\phi) = \begin{cases} \frac{b(\phi) - a(\phi)}{\max\{a(\phi), b(\phi)\}} & \text{if } |C_i| > 1 \\ 0 & \text{if } |C_i| = 1 \end{cases} \quad (7)$$

Finally, the silhouette coefficient  $SC$  is defined as the mean  $s(\phi)$  over the entire dataset. This is shown in Equation 8, in which  $C$  is the set of all clusters combined.

$$SC = \frac{1}{|C|} \sum_{\phi \in C} s(\phi) \quad (8)$$

## 2.5. Trustworthiness

The trustworthiness  $T$  is used to measure how much the local patterns in a projection (for instance, the embedded 2D space from t-SNE) mirror the actual patterns in the high dimensional data [62]. It is the proportion of points that are close in both the high dimensional and embedded space, computed according to Equation 9 [62, 63].

$$T(k) = 1 - \frac{2}{mk(2m - 3k - 1)} \sum_{i=1}^m \sum_{j \in \mathcal{N}_i^k} \max(0, (r(i, j) - k)) \quad (9)$$

In which for every sample  $\phi_i$ ,  $\mathcal{N}_i^k$  are its  $k$  nearest neighbors in the embedded space, and every sample  $\phi_j$  is its  $r(i, j)$ -th nearest neighbor in the high dimensional space. Nearest neighbors in the embedded space that are not neighbors in the high dimensional space are penalized proportionally to their rank in the high dimensional space [63]. The range of values for  $T(k)$  is in  $[0, 1]$ , with 1 being the best. Following the work of Espadoto et al. [62] and Martins et al. [64], the parameter  $k$  is set to be  $k = 7$ . The trustworthiness is used to assess the quality of the 2D projections generated by the visualization methods.

### 3. Weighted t-SNE

The proposed weighted t-SNE visualization is a method for inspecting the results from feature scoring algorithms. From a practical standpoint, we identified some desired properties that weighted t-SNE should have to be helpful for the analysis and comparison of feature scorers. It should: (i) account for non-linearity in the data; (ii) be model-agnostic (work for any scorer); (iii) use the importance score of each feature to produce a corresponding embedding; and (iv) consider all features in the original dataset if they all have an importance score larger than zero. Regarding the first item, many datasets contain complex structures in high-dimensional space, and the visualization must be able to reveal it in the projection space [1]. Moreover, non-linear and local methods like t-SNE are preferred in cluster and membership identification [14]. Considering all features with non-zero scores is important to distinguish between different scorers' behavior. For instance, to distinguish a scorer that concentrates all importance on a few features from a scorer that spreads the relevance among more features, even those that, in practice, are irrelevant. Moreover, this avoids the issue of having to define a hard threshold on what is a large enough score for a feature to be considered relevant.

Weighted t-SNE does not change the t-SNE algorithm itself but changes the relationship between data points (their distance) to reflect the importance of each dimension (feature). It leverages that feature scorers offer some feature relevance value, indicating the feature's contribution to classes or clusters identification. These values are the *weights* that give *weighted* t-SNE its name. All weights are scaled so that the maximum importance value is one and the minimum value (for an irrelevant feature) is zero.

The projection regarding the importance of the features is accomplished by incorporating the weights in the distance calculation. The weighted Euclidean distance  $d_w(\cdot)$  [65, 66] (Equation 10) is used in Equation 1 from Subsection 2.3 to compute the distance between two  $N$ -dimensional points  $\phi$  and  $\psi$ , using the  $N$  weights in  $\omega$ .

$$d_w(\phi, \psi, \omega) = \sum_{i=1}^N \sqrt{(\omega[i](\psi[i] - \phi[i]))^2} \quad (10)$$

Considering a  $N$ -dimensional dataset  $\mathbf{X} \in \mathbb{R}^N$  with  $m$  points, the difference of each dimension  $i$  between the points  $\phi$  and  $\psi$  is multiplied by weight  $\omega[i]$  (the importance score of that feature). As shown in Equation 11, this is equivalent to multiplying each point by  $\omega$  or to perform the Hadamard product (element-wise product) between a matrix composed of  $m$  columns repeating the  $N$  (the number of features) values of  $\omega$  and  $\mathbf{X}$  to obtain a new scaled dataset  $\mathbf{X}' \in \mathbb{R}^N$  and use it to create the visualization.

$$\omega[i](\psi[i] - \phi[i]) = (\psi[i] \cdot \omega[i]) - (\phi[i] \cdot \omega[i]) \quad (11)$$

Because the weights range from zero (irrelevant) to one (relevant), scaling the dimensions by their weights allows features of higher importance to have a greater impact on the distance between points. The desired outcome is that relevant features influence the point position more in the final visualization.

Because the feature scaling can be performed before the embedding optimization, the additional computing cost of weighted t-SNE is negligible. With the current improvements in the implementation of t-SNE, the code runs in a matter of seconds or minutes. It should not significantly impact the total time of the data analysis pipeline.

It is worth noting that the method faces the same challenges as regular t-SNE regarding hyperparameter tuning [58]. It should be possible to adopt this strategy to create variants of other visualization methods, for instance, a "weighted UMAP." Such variants would inherit the advantages and drawbacks of their original algorithms. This research focuses on t-SNE due to the advantages discussed in Subsection 2.2 and 2.3 and because narrowing on a single variant allows for a more diverse set of experiments.

The main goal of weighted t-SNE is to provide an inspection tool to compare and choose feature scorers in different datasets. Additionally, it can be one more item in the toolset of interpretable machine learning (Subsection 4.1). Weighted t-SNE was not designed for the individual visualization of specific features nor to score features themselves. Instead, it provides a global visualization of the impact scorers have on the entire dataset and how it relates to data clusters. As such, the analysis of the results of a feature scorer using weighted t-SNE only makes sense in a comparative manner, meaning that the visualization of the dataset after the scoring needs to be contrasted to a visualization of the dataset before the scoring. The single 2D projection of weighted t-SNE for a feature scorer is meaningless if the 2D projection of the original data is not present, allowing the user to perceive or measure the difference between the two.

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

Weighted t-SNE focuses on feature scoring rather than feature selection, meaning it visualizes feature importance scores rather than selecting a subset of features. While feature selection and scoring are related, they serve different purposes. Feature selection typically receives more attention in visualization tools, whereas feature scoring lacks intuitive inspection methods. Weighted t-SNE fills this gap but does not replace feature selection techniques.

Weighted t-SNE can be used with other visualization tools, specifically those designed to visualize individual features' importance, for a richer evaluation. An example of this application was demonstrated using a preliminary version of weighted t-SNE and the table heatmaps to analyze neural networks on multiple datasets in a previous study [10]. The same research also explored using weighted t-SNE to visualize different scores for different clusters of the same dataset. It highlighted the learning of distinct importance scores for each data class by neural networks.

Recognizing that weighted t-SNE is an "indirect way" to evaluate feature scorers relying on a visual approximation of the data separation after scoring is essential. However, evaluating feature selection by classification or regression metrics (training a model on the top  $n$  features) is also indirect. Even though feature scoring and feature selection are typical parts of classification or regression pipelines (mainly during preprocessing), these are not the same tasks, nor is the only purpose of feature selection to improve classification accuracy. Thus, using an independent classifier to evaluate the selection of features, while recommended as additional validation, is not a perfect measure.

Evaluating feature selectors based on classification metrics would apply to feature selection algorithms (the top  $n$  features would be used to train a classifier) but not necessarily to feature scorers, for which all features receive a score. Although deeply connected, feature scoring and selection are different problems, as discussed in the next section. The results from feature scorers can be converted into feature selection. However, in the process, it is necessary to define a cutting threshold external to the algorithm and to discard features with scores under that threshold. The scores themselves are not considered in this context. Weighted t-SNE was designed with this issue in mind, thus visualizing all the scores. Some of the experiments in the next section highlight these differences. Another point is that not all feature scorers are based on classification or regression models, such as the Kruskal-Wallis Filter or the mRMR. While an improvement in classification or regression metrics given a specific selection of features can indirectly measure the selection quality, it will not always be the case. Machine learning models are known to learn biases [67] and shortcuts [68] from the data that lead to better metrics without necessarily discovering the rules or representations desired by the user. In this scenario, a scorer that follows a different strategy (for instance, a statistical or information-based algorithm) may find a better set of relevant features from a knowledge discovery standpoint but that does not maximize the prediction accuracy of a model. Because all these methods, including weighted t-SNE, approximate the measurement of the scoring quality, it is recommended that they be used together to combine their strengths and mitigate their drawbacks.

#### 4. Experiments and discussion

Six datasets were used for the experiments with weighted t-SNE and feature scorers. They are fully described in Table 1. Two datasets are synthetic classification tasks, so the number of relevant and irrelevant features can be controlled. The first dataset used in the study was modeled after the exclusive-OR (XOR), using two out of a total of  $n$  inputs [69, 10]. The XOR function in this problem is determined by two specific binary features in the input data, which are known to the user but not to the machine learning models. The other  $n - 2$  features, where  $n = 50$ , are randomly generated binary values with no effect on the output. Samples are assigned either to class 0 or class 1 according to the following values of their two relevant features:  $0 \oplus 0 = 0$ ,  $1 \oplus 1 = 0$ ,  $0 \oplus 1 = 1$ , and  $1 \oplus 0 = 1$ . The other 48 features are zero or one and do not impact the sample's class. Despite its apparent simplicity, this problem is made more difficult by its non-linearity and the presence of many irrelevant features.

Another synthetic dataset with two classes and 100 samples created with *scikit-learn* [70, 71]<sup>1</sup> was also used. This dataset has 5,000 input features, with only 50 being informative for class separation (unknown to the algorithms), and the rest being irrelevant. Two other datasets are genes expression from cancer microarray experiments due to their importance, as discussed in Subsection 2.1. They are from liver and prostate cancer microarray experiments obtained from the CuMiDa database [2]<sup>2</sup> under the accession codes GSE22405 and GSE6919\_U95B. These datasets are composed of two classes, one with samples from healthy tissue and the other from tumorous tissue. The choice of gene expression datasets followed the recommendations of Grisci et al. [44] regarding the data quality and up-to-dateness.

<sup>1</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make\\_classification.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html)

<sup>2</sup><https://sbcb.inf.ufgrs.br/cumida>

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

**Table 1**

Description of the datasets used in the weighted t-SNE experiments. Relevant is the number of features which are relevant to the task. All datasets are numerical and tabular.

Dataset	Samples	Features	Relevant	Classes	Origin	Reference
XOR	500	50	2	2	Synthetic	[69, 10]
Synth	100	5000	50	2	Synthetic	[70, 71]
Liver	48	22284	Unknown	2	Microarray	[2]
Prostate	115	12647	Unknown	2	Microarray	[2]
Regression	1000	100	4	-	Synthetic	[72, 71]
Mouse cortex	23822	45769	Unknown	23	RNA-seq	[73, 47]

Two additional datasets are used to illustrate the points made in Subsection 4.1. The first is a synthetic regression dataset (Fig. 1) with 100 features and 1,000 samples with continuous target values ranging between  $-12$  and  $12$  created with *scikit-learn* [72, 71]<sup>3</sup>. It was only tested with the neural network because some of the other algorithms are not directly suited for regression. The second is an RNA-seq dataset of 23,822 cells from adult mouse cortex [73], divided by the hierarchy of cell types (23 classes). This larger dataset was used for the machine learning experiments. Except for the XOR dataset, all the data was standardized. It is a best practice only to apply feature selection or scoring in balanced datasets. All datasets used in our experiments have balanced classes to avoid issues. There is no general way to describe the effects of class imbalance in all feature scorers because each algorithm has distinct strategies, advantages, and drawbacks. However, the work of Kamalov et al. [74] further discusses the problem.

The weighted t-SNE was implemented using Python 3, the *openTSNE* library<sup>4</sup>, and *NumPy* [75]. The perplexity was computed as described in Subsection 2.3, and the positions were initialized using PCA, following the recommendations from Kobak and Berens [47]. The appropriate learning rate is selected according to  $\max(200, n/12)$  [76]. All experiments were conducted with 500 iterations. The silhouette coefficient was computed using the implementation from *scikit-learn* [71]<sup>5</sup>.

Nine feature scorers were chosen to generate the visualizations: Kruskal-Wallis Filter, Mutual Information, mRMR, ReliefF, Lasso, Decision Tree, Random Forest, Linear SVM, and Neural Network. These algorithms represent the several scoring strategies discussed in Subsection 2.1, as summarized in Table 2. All their outputs are feature importance weights between zero (irrelevant) and one (relevant), so the method presented in Section 3 can be applied. The machine learning-based methods are further described and discussed in Subsection 4.1. The neural network was trained using *Keras* with the *TensorFlow* backend. The experiment with the XOR dataset had two hidden layers of 20 neurons each, ReLU activation and L1 regularization with a factor of 0.001. The network trained for the regression task had two hidden layers with 128 and 64 neurons, and the network for the mouse cortex dataset had four hidden layers with 100, 200, 100, and 100 neurons. For the other experiments, the neural network had four hidden layers with 32 neurons each and L1 regularization with a factor of 0.01. Because the goal of the experiments is only to show the functionality of weighted t-SNE, the other scorers were trained using the recommended hyperparameters of their implementations from *scikit-learn* [71]. Further details about each feature scorer and how feature importance is computed are present in the **Supplementary material**. Weighted t-SNE can even be used to compare different sets of hyperparameters for the same feature scorer. In addition to the scorers, the results for “no scoring” are also shown, corresponding to a standard t-SNE visualization of the original high-dimensional data.

The results for each binary classification dataset are shown in the Fig. 2, Fig. 3, Fig. 4, and Fig. 5. As an example of the utility of the weighted t-SNE, the user can compare the visualizations of each feature scorer to the original data and each other and inspect which ones have better results. For instance, it is clear from Fig. 2a that the XOR dataset with all the original features does not have distinguishable classes (as expected because most of the features are random). Considering the data modified to reflect the weights of the feature scorers, some structure starts to appear in the visualizations. However, for this dataset only ReliefF (Fig. 2e), Random Forest (Fig. 2h), and Neural Network (Fig. 2j) were able to capture the correct structure of XOR. The two classes in these figures appear separated into four clusters because each cluster contains samples of one of the four possible combinations of the two relevant features. For instance, the class 0 (in red) is split between two clusters ( $0 \oplus 0$  and  $1 \oplus 1$ ). The challenge of this dataset is to identify

<sup>3</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make\\_sparse\\_uncorrelated.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_sparse_uncorrelated.html)

<sup>4</sup><https://opentsne.readthedocs.io/en/latest/>

<sup>5</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html)

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

**Table 2**

Properties of the feature scorers used in the experiments. The type refers to how the scorer works and the correlation to how interactions between features are treated.

Scorer	Type	Correlation	Reference
Kruskal-Wallis	Filter	Univariate	[30]
Mutual Information	Filter	Univariate	[32]
mRMR	Filter	Multivariate	[31]
ReliefF	Filter	Multivariate	[33]
Lasso	Embedded	Multivariate	[35]
Decision Tree	Embedded	Multivariate	[77]
Random Forest	Ensemble	Multivariate	[36]
Linear SVM	Embedded	Multivariate	[78]
Neural Network	Embedded	Multivariate	[10]

the non-linear relationship between the variables among very noisy data. The three scorers with the best visualization were the ones that gave larger importance scores to the only two relevant features. For instance, the random forest scored the two relevant features with a 0.99 and a 1.0, and all other features were scored with less than 0.2.

The same happens for the Synth dataset in Fig. 3, which like the XOR dataset, is composed mainly of random features. However, this time the two algorithms with the apparent best results are the Kruskal-Wallis Filter (Fig. 3b) and the ReliefF (Fig. 3e). The algorithm used to create the Synth dataset [70, 71] builds the classes by initially creating clusters of points normally distributed, which may explain why a simple statistical method like the Kruskal-Wallis Filter was able to obtain a good result. The size of this dataset also presents a great challenge for models based on classifiers, as it is hard to learn the correct relationships with only 100 data points in the space of five thousand features, of which only 50 are relevant. Interestingly, Kruskal-Wallis Filter and ReliefF were able to “outperform” a visualization with “perfect” selection (Fig. 6a). In this case, a “perfect” selection was created by weighing all fifty relevant features with a score of one and all remaining features with zero. However, this scenario did not create a clear class separation in the final visualization (Fig. 6b). This example also illustrates the point made at the end of Subsection 2.3. In this case, the “perfect” selection is equivalent to the visualization of a feature selection applied to the Synth dataset, in which fifty features were selected. In feature selection, the importance score of each feature is disregarded, generating different results from the weighted t-SNE approach. This result shows that incorporating importance scores can reveal a “hidden” structure in the visualization. While not equivalent, both strategies can be used to make complementary visualizations of feature scoring and selection.

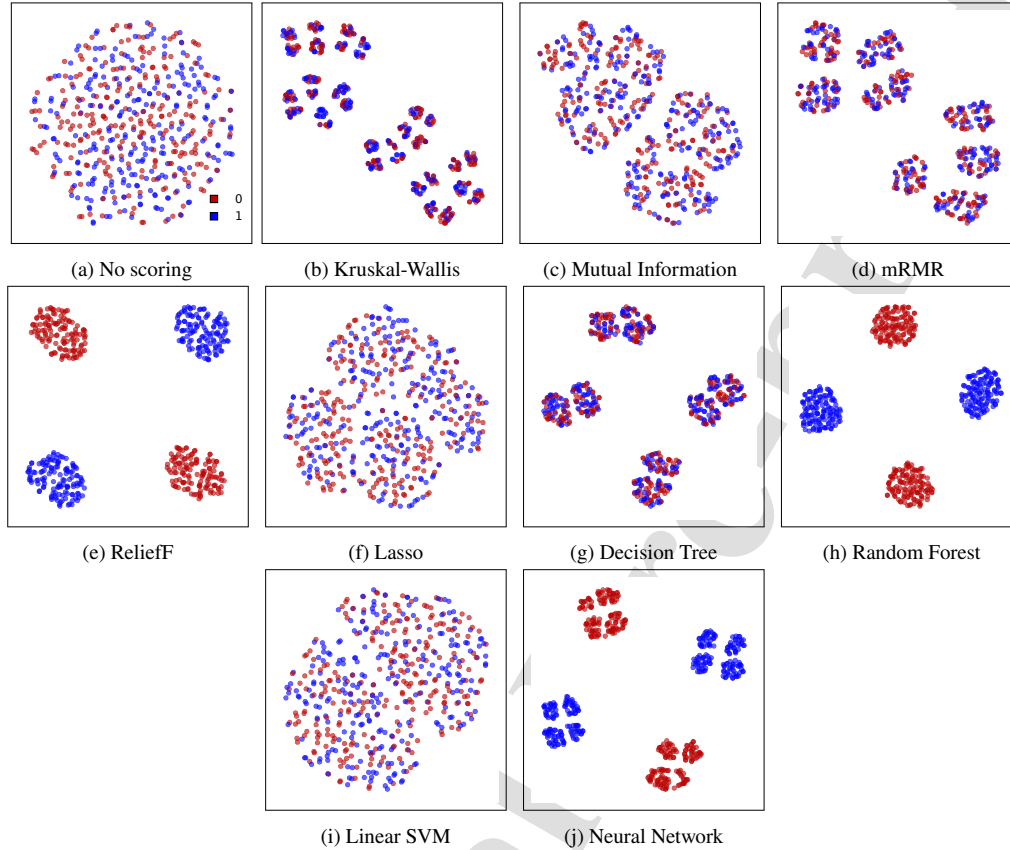
The most relevant features (genes) are unknown for the two cancer datasets in Fig. 4 and Fig. 5. Moreover, the dimensionality is one order of magnitude larger than the synthetic datasets with fewer samples. These characteristics make for a more challenging analysis, as is often the case with real-world data. Even so, by inspecting the visualizations of the liver cancer dataset, it is possible to distinguish between scorers that are splitting the two groups (for instance, the mRMR in Fig. 4d and Neural Network in Fig. 4j) and those that are not (Fig. 4f, Fig. 4i). For the prostate cancer data in Fig. 5, most visualizations do not split the two classes. The use of weighted t-SNE shows that the scorers are not able to score features in a way that visually improved data classification. In this case, the visualization helps inform the user that further experimentation may be needed. This outcome is actually expected because this dataset is described as a hard classification and clustering task in the database by Feltes et al. [2].

Another interesting application of weighted t-SNE is in outlier detection. In the regular visualization of the liver cancer dataset (Fig. 4a), it is impossible to identify any outliers because the two classes are entangled. However, an outlier can be spotted once the features are weighted using their importance scores and the classes appear separated in the visualization. In Fig. 4d, 4h, and 4j, it is possible to see a single hepatocarcinoma sample (red point) clustered together with the normal (healthy) samples (blue points). This sample that appears out of place has the accession code GSM557108 and is identified as “liver tissue of subject 15, tumor”<sup>6</sup>. The healthy samples in this dataset were obtained from the adjacent tissue of primary hepatocarcinoma samples<sup>7</sup>, so it is possible that this outlier was actually a healthy sample that was mislabeled, or perhaps a tumor sample with contamination from neighboring healthy tissue.

<sup>6</sup><https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM557108>

<sup>7</sup><https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE22405>

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



**Figure 2: Weighted t-SNE visualization for the XOR dataset.** Each figure is the visualization of the 2D embedding with weighted t-SNE for each of the nine feature scorers, plus the original data in “no scoring.” Each red or blue point represents one sample from one of the two classes. The color legend is shown in Fig. 2a and omitted in the other figures.

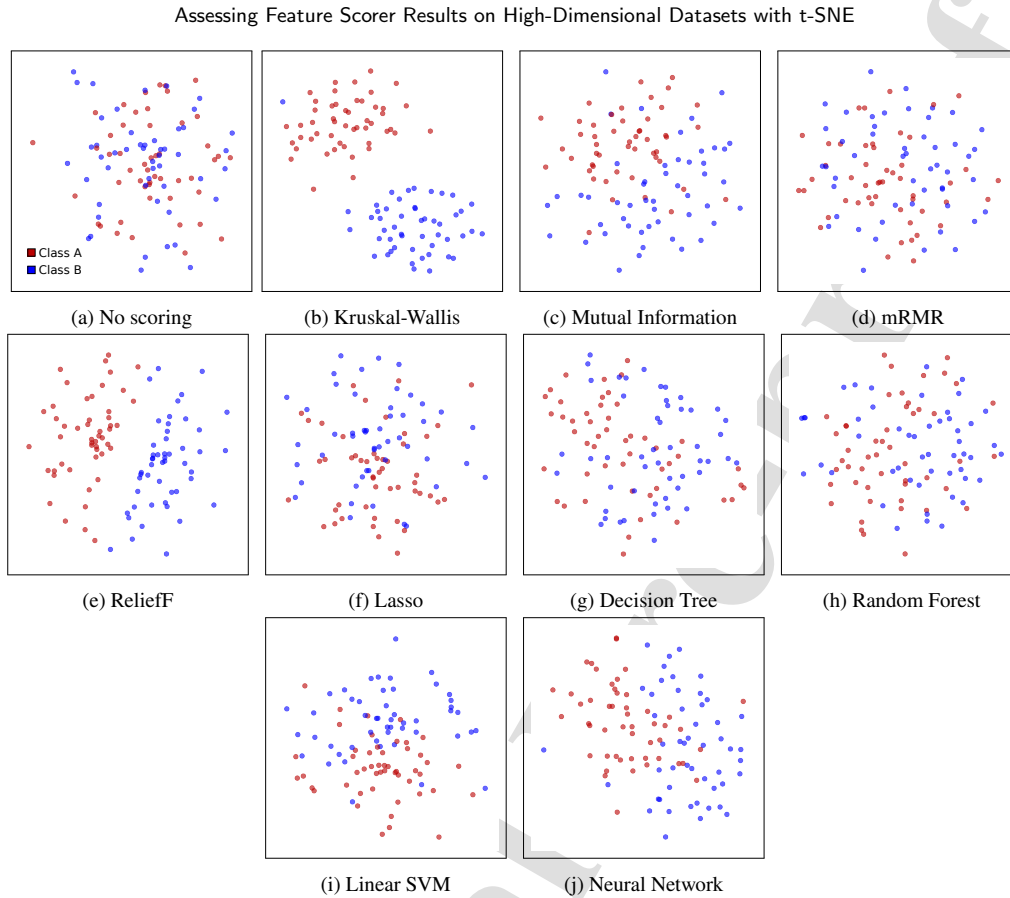
As mentioned in Subsection 2.4, the silhouette coefficient can be used to summarize the results of the visualization of feature importance [20] (Fig. 2, Fig. 3, Fig. 4, and Fig. 5). The silhouette coefficient for each scorer and dataset is shown in Table 3. The scorers with the best visual split between classes have the larger silhouette coefficients. Table 3 also shows the Kullback–Leibler divergence of each t-SNE optimization in the KL columns. The complimentary Table 4 presents the trustworthiness of each 2D visualization presented in this work, as discussed in Subsection 2.5. It was computed using the *scikit-learn* implementation<sup>8</sup>.

The first analysis to be made is to compare the silhouette coefficients of each scoring to the coefficient obtained without scores as a baseline to be beaten. Table 3 also allows comparing the original high-dimensional space and the 2D embedded space from t-SNE. Because of the behavior of distance metrics in high dimensions [79], the silhouettes in the original space tend to be closer to zero than in the embedding space. The expected behavior is for the patterns of scorers with larger or smaller coefficients to be preserved in the high-dimensional and embedded spaces.

A user interested in incorporating a feature scorer in a larger analysis pipeline can then use weighted t-SNE to decide which of the available options better suits their specific dataset or task. The user would select a few scorers they wish to inspect, get the importance of the features from each of them, and then generate the weighted visualizations for each scorer being tested and a regular t-SNE plot for the dataset. Then, they can use the visualization and the corresponding silhouette coefficients to select the scorer that produced the best or expected results. The following

<sup>8</sup><https://scikit-learn.org/stable/modules/generated/sklearn.manifold.trustworthiness.html>





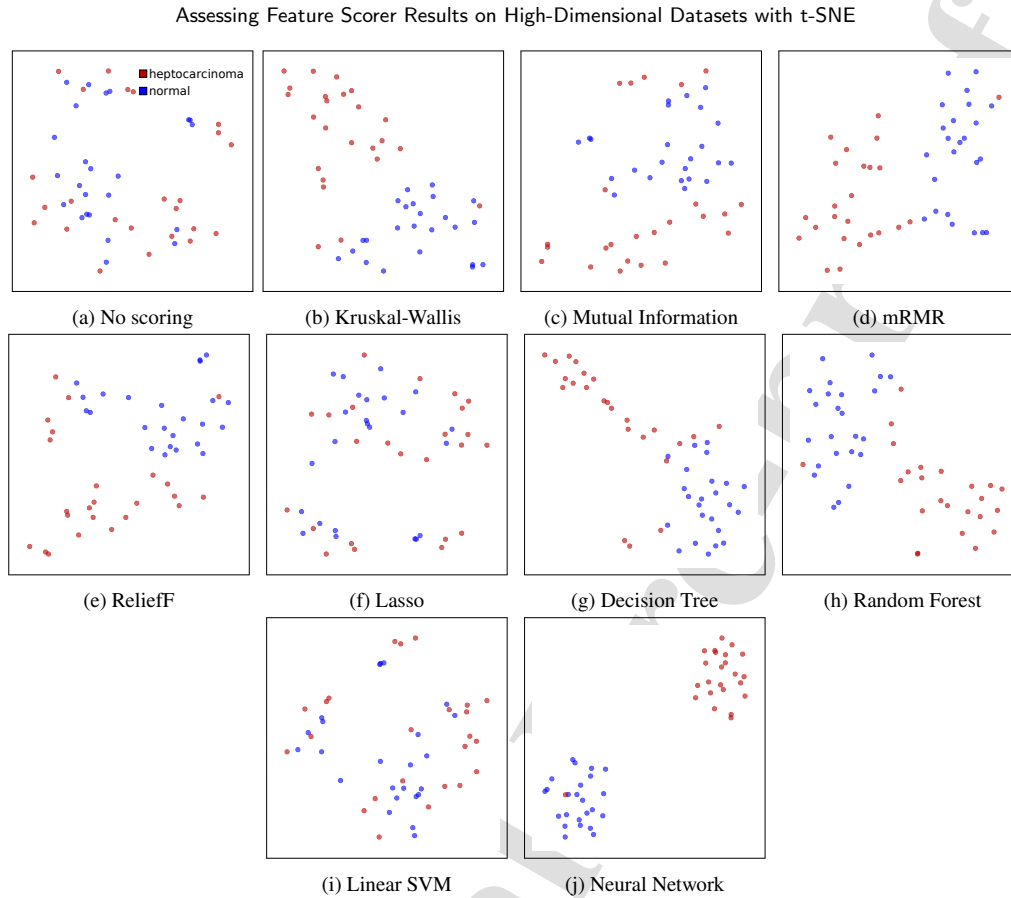
**Figure 3: Weighted t-SNE visualization for the synthetic 2-classes dataset.** Each red or blue point represents one sample from one of the two classes. The color legend is shown in Fig. 3a and omitted in the other figures.

experiment illustrates and validates this idea. A simple k-nearest neighbors (kNN) [80] classifier ( $k = 3$ ) was trained on each of the datasets using the different scorers (including the baseline case with no scores). The kNN was chosen to avoid biases in the results from other classifiers used as scorers, such as SVM or decision tree. The results in Table 3 are the F1-scores [81] of the kNN on the test sets (a 33% split from the complete dataset). As can be seen, the feature scorers with better silhouette coefficients achieved higher classification performance. Thus, together with other metrics and analyses, the weighted t-SNE visualization can be used to select the best feature scorers for a classification pipeline.

#### 4.1. Machine learning interpretability

Even though machine learning has become widespread in the past years and many models achieved state-of-the-art results in challenging tasks, the more complex models, especially deep learning or large ensemble models such as random forests, are still widely regarded as “black-boxes” [11, 10]. Their intrinsically complex structures make explaining or predicting their behaviors hard. The learned features are only implicitly described by many internal model parameters [82]. The lack of explainable decisions can lead to under-performance, distrust, or machine bias [83, 11, 84]. The techniques that help humans to understand the cause of a decision made by a machine learning model are known as interpretable or explainable machine learning.

The experiments above show that weighted t-SNE can be used to visualize the feature importance learned by a few popular machine learning models, such as decision trees, random forest, Lasso, SVM, and neural networks. For the first two, the feature importance measures the total reduction of the impurity brought by that feature (Gini importance),

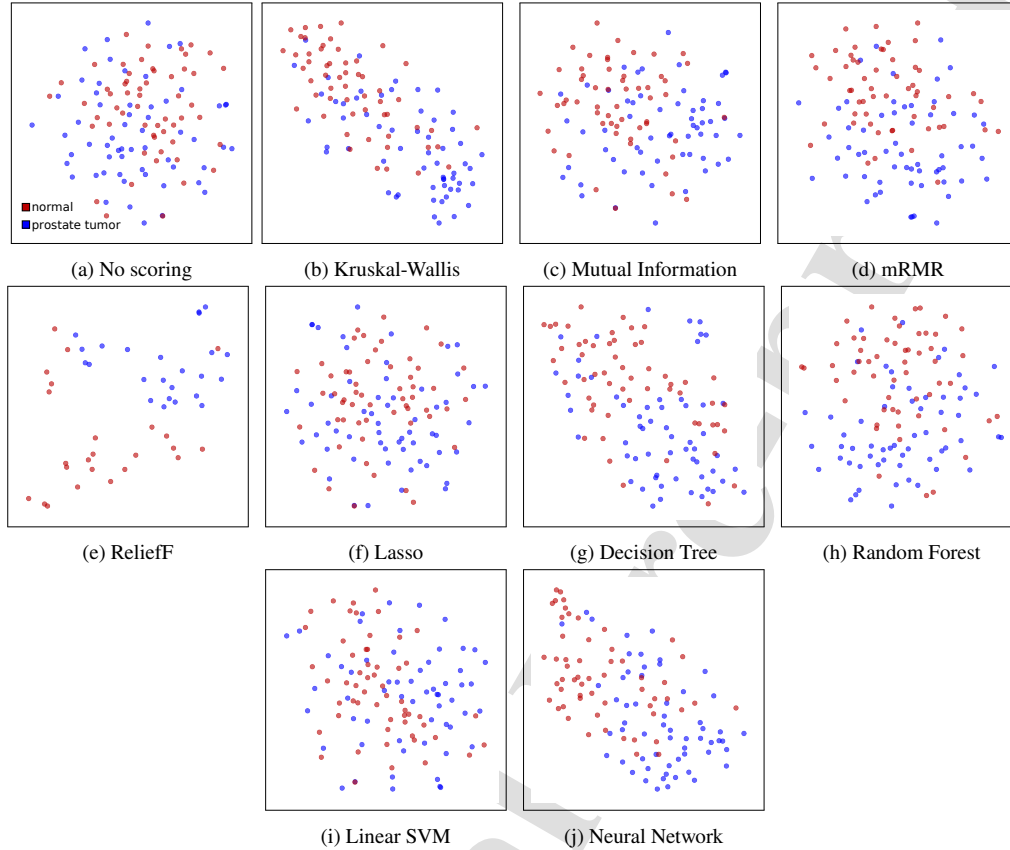


**Figure 4: Weighted t-SNE visualization for the liver cancer dataset.** Each red or blue point represents one sample from one of the two classes. The color legend is shown in Fig. 4a and omitted in the other figures.

or in other words, how well that feature is splitting the classes inside the model [85, 11]. The reduction of variance or Gini index in relation to the parent node should be summed for all splits using the feature and scaled to 100 to compute the overall importance of a feature in a decision tree [11]. For the linear SVM and Lasso, both being linear models, the coefficients assigned to the features during training time can be used to measure their importance to the model. For neural networks, the feature importance was computed using relevance aggregation [10], an algorithm suited for tabular data that uses Layer-wise Relevance Propagation (LRP) [86] to measure the impact each feature has in the output of the network. The calculation of feature importance for each model is described in more detail in the **Supplementary material**.

Weighted t-SNE can help interpret these methods by visually displaying the impact of the importance of the learned features. In the example shown in Fig. 1, it is visible that the neural network is giving more importance to features that better separate the samples across their target values. The synthetic regression dataset has only four truly relevant features out of 100, and the neural network in Fig. 1 gave the largest importance scores to three of them (0.44, 0.40, 0.25, respectively, while the next largest score of 0.20 was given to an irrelevant feature and the fourth truly relevant feature received a score of 0.11). In the example of Fig. 2, one can visually intuit that the neural network and the random forest learned a correct representation of the problem and that the decision tree and the linear SVM did not before even seeing the accuracy of these models. It is important to highlight that the linear SVM was already expected not to be able to perform well on the XOR dataset, a non-linear problem, as it, by definition, cannot find non-linear relations between features. However, as the goal of the experiments is to analyze the visualizations and not the scorers, the weighted

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



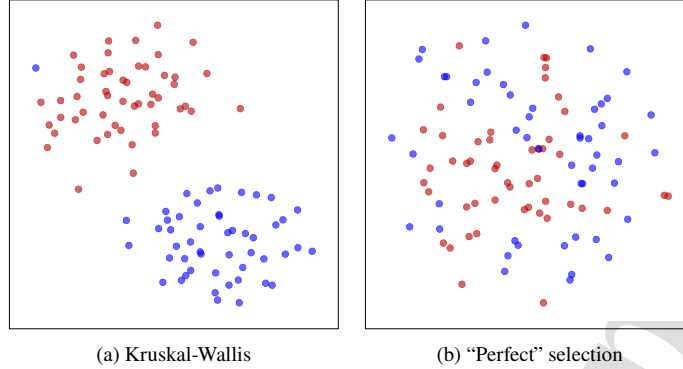
**Figure 5: Weighted t-SNE visualization for the prostate cancer dataset.** Each red or blue point represents one sample from one of the two classes. The color legend is shown in Fig. 5a and omitted in the other figures.

t-SNE visualization of the linear SVM results is accomplishing its goal of showing that this is not a suitable scorer for the task.

Fig. 7 shows the results for larger real-world datasets. The mouse cortex dataset has over 20 thousand samples, over 40 thousand features, and 23 classes. Visualizing the scores learned by three distinct machine learning models, namely Lasso (Fig. 7b), random forest (Fig. 7c), and neural network (Fig. 7d) using weighted t-SNE reveals that the models with better performance (larger F1-score) also produced a better split of samples in the 2D embedding (larger silhouette coefficient). The regular t-SNE visualization of the dataset (Fig. 7a) may appear different from other publications, for instance, from its depiction by Kobak and Berens [47], because we purposely did not perform feature selection and other filtering protocols running the t-SNE algorithm. This is necessary because we wish to compare feature scorers without biases introduced by other algorithms.

Regarding the visualization of the importance scores obtained with neural networks, especially in the case of weighted t-SNE (Fig. 8a), it is essential to differentiate it from the visualization made by projecting the layers activation (Fig. 8b) (the internal state of the network for a given input) [87]. In the case discussed here, what is visualized is the feature importance of the original data dimensions as learned by the neural networks. So it is a direct representation of the input features. In the second case, the visualization is of the neural network's internal data representation and does not directly map to the input features. One option does not invalidate the other, and both can be used together to increase the understanding of a model, as presented in Fig. 8. While both visualizations show that the neural network

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



**Figure 6: Comparison between the Kruskal-Wallis Filter and a "perfect" selection.** Each figure is the visualization of the 2D embedding with weighted t-SNE for the results of Kruskal-Wallis Filter and a "perfect" selection. This visualization refers to the synthetic dataset with 5000 features, in which only 50 are relevant. The "perfect" solution weights all relevant features with a score of one and all noisy features with a score of zero. Details as in Fig. 3.

**Table 3**

**Silhouette coefficient, KL divergence, and F1-score of a kNN trained on the data for four distinct datasets and nine feature scorers.** The HD columns show the silhouette coefficient for the original high-dimensional space, the 2D columns show the silhouette coefficient for the 2D embedding representation learned by t-SNE, and the KL columns show the Kullback–Leibler divergence from the t-SNE optimization. For each scorer, the silhouette coefficient is computed considering the scalarization of the dimensions by their importance, except for "no scoring." The kNN column shows the F1-score on a test set (33% of samples) of a kNN trained on the datasets using the importance scores.

	XOR				Synth				Liver				Prostate			
	HD	2D	KL	kNN	HD	2D	KL	kNN	HD	2D	KL	kNN	HD	2D	KL	kNN
No scoring	0.000	0.003	2.12	0.58	0.001	-0.009	0.16	0.63	0.038	0.038	0.14	0.68	0.008	0.046	0.51	0.58
Kruskal Wallis	0.004	0.010	0.48	0.51	0.041	0.609	0.56	1.00	0.183	0.449	0.15	0.94	0.072	0.196	0.35	0.71
Mutual Information	0.000	-0.001	1.19	0.54	0.006	0.180	0.69	0.66	0.133	0.234	0.15	0.94	0.025	0.121	0.49	0.68
mRMR	0.000	-0.002	0.74	0.54	0.008	0.030	0.77	0.56	0.228	0.560	0.18	0.94	0.031	0.156	0.54	0.66
ReliefF	0.104	0.204	0.80	1.00	0.005	0.433	0.33	0.97	0.128	0.333	0.18	0.94	0.018	0.126	0.48	0.68
Lasso	0.000	0.005	1.54	0.55	0.001	0.052	0.16	0.60	0.039	0.033	0.14	0.68	0.008	0.027	0.51	0.55
Decision Tree	-0.001	-0.002	0.65	0.58	0.050	0.058	0.48	0.61	0.358	0.422	0.07	0.81	0.078	0.122	0.42	0.71
Random Forest	0.089	0.204	0.91	1.00	0.014	0.045	0.74	0.67	0.235	0.490	0.21	0.94	0.052	0.162	0.51	0.73
Linear SVM	0.000	-0.001	1.59	0.56	0.002	0.180	0.19	0.64	0.040	0.037	0.16	0.81	0.008	0.049	0.55	0.60
Neural Network	0.148	0.192	0.49	1.00	0.014	0.200	0.65	0.85	0.387	0.782	0.17	0.94	0.098	0.200	0.40	0.73

learned to split the two classes of the XOR dataset, weighted t-SNE presents the samples regarding the input features of the dataset so that the final plot resembles the XOR function.

Weighted t-SNE can also be applied at different moments during the training of the neural networks, allowing the visualization of the learning process according to the current feature importance scores. Fig. 9 and Fig. 10 show how the visualization changes at different training epochs and how the networks "learn" to split the samples into their clusters as the training progress. Table 5 shows that the training loss of the neural networks is inversely correlated to the corresponding silhouette coefficient, indicating that improving the prediction also improves the feature scoring, thus resulting in better clustering in the 2D embedding.

So far, our experiments have focused on the relationship between cluster visualization in weighted t-SNE, silhouette coefficient, and prediction performance. The key takeaway was that feature scorers producing clearer clusters in weighted t-SNE tend to correlate with higher classification performance, as evidenced by the kNN classifier results in Table 3, the alignment of silhouette coefficients with F1-scores in Fig. 7, and the correlation between neural network loss and silhouette scores in Fig. 10 and Table 5. These findings suggest that weighted t-SNE can be a useful tool for selecting feature scorers that improve downstream prediction in supervised learning pipelines. This focus was due to the relevance of feature scoring in supervised learning and the need to somehow quantify an intrinsically visual and interpretative result such as weighted t-SNE visualizations. Because labeled datasets were used in the experiments, it is also fair to assume that the data is clusterable, as the samples are already sorted into predefined classes. However,

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

**Table 4**Trustworthiness  $T(7)$  of the 2D t-SNE and weighted t-SNE projections for all datasets and feature scorers.

	XOR	Synth	Liver	Prostate	Regression	Mouse cortex
No scoring	0.814	0.653	0.939	0.809	0.714	0.548
Kruskal-Wallis	0.997	0.863	0.949	0.896	-	-
Mutual Information	0.978	0.728	0.949	0.843	-	-
mRMR	0.990	0.792	0.937	0.821	-	-
ReliefF	0.978	0.701	0.954	0.826	-	-
Lasso	0.934	0.648	0.943	0.815	-	0.995
Decision Tree	0.987	0.921	0.984	0.930	-	-
Random Forest	0.974	0.806	0.937	0.871	-	0.998
Linear SVM	0.930	0.647	0.937	0.797	-	-
Neural Network	0.993	0.834	0.956	0.921	0.681	0.979

**Table 5**

Pearson correlation between training loss of neural network and the corresponding silhouette coefficient of the weighted t-SNE 2D embedding.

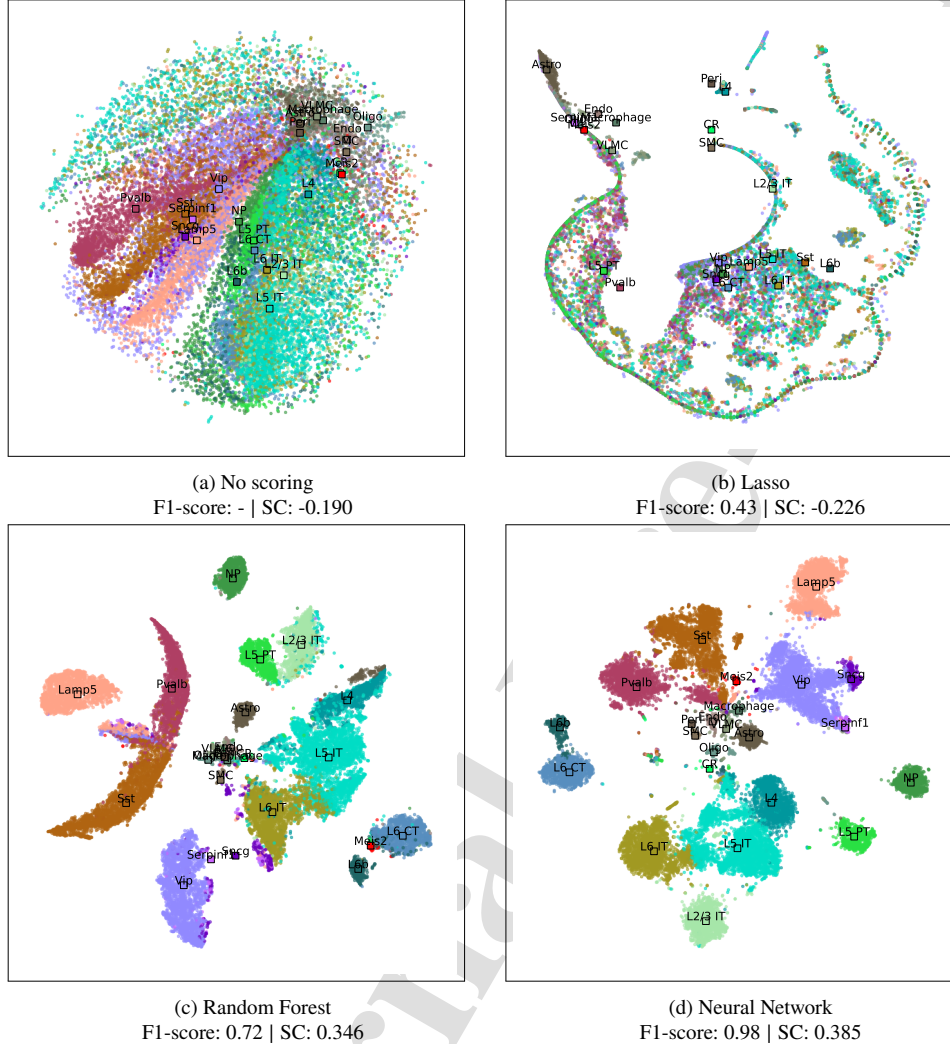
Dataset	Correlation
XOR	-0.85
Synth	-0.37
Liver	-0.81
Prostate	-0.67
Regression	-0.74
Mouse cortex	-0.98

this strong connection between cluster clarity and classification accuracy raises an important question: *if accuracy correlates with better feature scoring, is visualization even necessary, or would prediction metrics alone suffice?*

To show that weighted t-SNE reveals insights beyond what prediction metrics can capture, we devised the following experiment. We created a synthetic dataset with three classes, 500 samples, and 150 features, where each class is defined by five relevant features (15 relevant features in total), while the remaining features contain only noise. The original dataset contained three distinct classes: red, blue, and cyan, as shown in Fig. 11. However, before training, we manually merged the blue and cyan samples into a single class, reducing the task to a binary classification problem between red and blue. We then trained two random forest models (A and B) on this binary-labeled dataset using the same hyperparameters. Both models achieved identical classification performance: F1-score of 0.99 on training and 0.93 on test. However, because of the random initialization of the algorithm, each random forest assigned different feature importance scores, despite reaching the same F1-scores. The weighted t-SNE projections for both models are shown in Fig. 11. Random Forest A (Fig. 11a) groups the samples into two clusters, red and blue, strictly following the binary classification labels, ignoring internal structure within the blue class. Random Forest B (Fig. 11b) instead produces three clusters, preserving the original three-class structure, even though it was trained on a binary task. These differences arise because each model relied on different features, as seen in Table 6, which lists the top ten most important features for both models. Even though they achieved the same F1-scores, they assigned importance to different features, ultimately influencing how the samples were structured in the visualization.

Distinguishing between these two models based solely on F1-scores would be impossible. The models are equally predictive, yet one preserves meaningful structure in the data while the other does not. This highlights a limitation of using prediction metrics alone to evaluate feature scorers: they do not capture how scorers organize the data or whether they reveal hidden structure. This raises the interesting question of which model is better. We argue that there is no single correct answer. If the decision to merge blue and cyan was intentional, then Random Forest A is preferable because it enforces this labeling choice. If the user was unaware that blue contained two distinct subgroups, then Random Forest B provides a valuable insight, revealing that the merged class actually contains two coherent clusters. In either case, weighted t-SNE provides an intuitive way for users and domain experts to inspect feature scoring, offering insights that would be difficult to extract from raw feature importance values alone, especially in high-dimensional datasets.

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



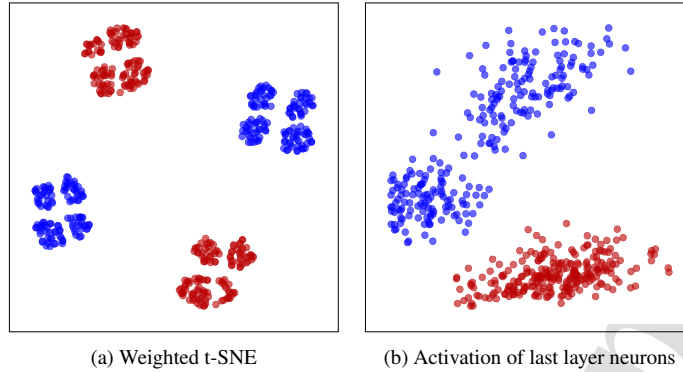
**Figure 7: Weighted t-SNE visualization for the mouse cortex dataset.** Each figure is the visualization of the 2D embedding with weighted t-SNE for three machine learning models, plus the original data in “no scoring.” The captions describe the F1-score of each model and the silhouette coefficient (SC) of each 2D embedding. Each color corresponds to a different class. Following the scheme from Kobak and Berens [47], warm colours correspond to inhibitory neurons, cold colours correspond to excitatory neurons, and brown or grey colours correspond to non-neural cells.

While this is a synthetic example, similar situations frequently occur in real-world applications where interpretability is critical. In fraud detection, one model may prioritize transaction history while another focuses on user behavior. Both may achieve the same fraud classification accuracy, but a domain expert might prefer one feature set over the other depending on interpretability needs.

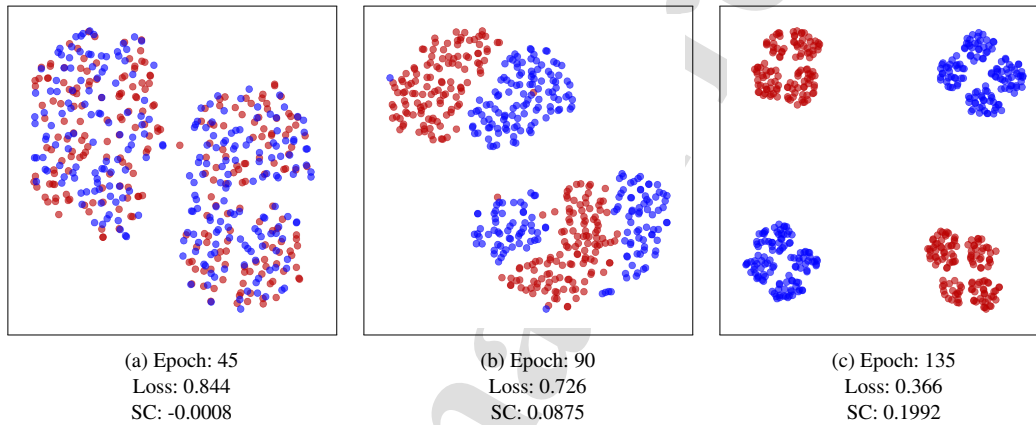
Weighted t-SNE can be especially useful for the interpretability of neural networks applied to tabular datasets such as the examples in Table 1. Empirical evidence has shown that providing good explanations for networks with many fully connected layers, commonly used in tabular datasets, is hindered by a lack of selectivity [88]. In works based on transformers, the attention layer is restricted to categorical features and is not applied to continuous features



Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



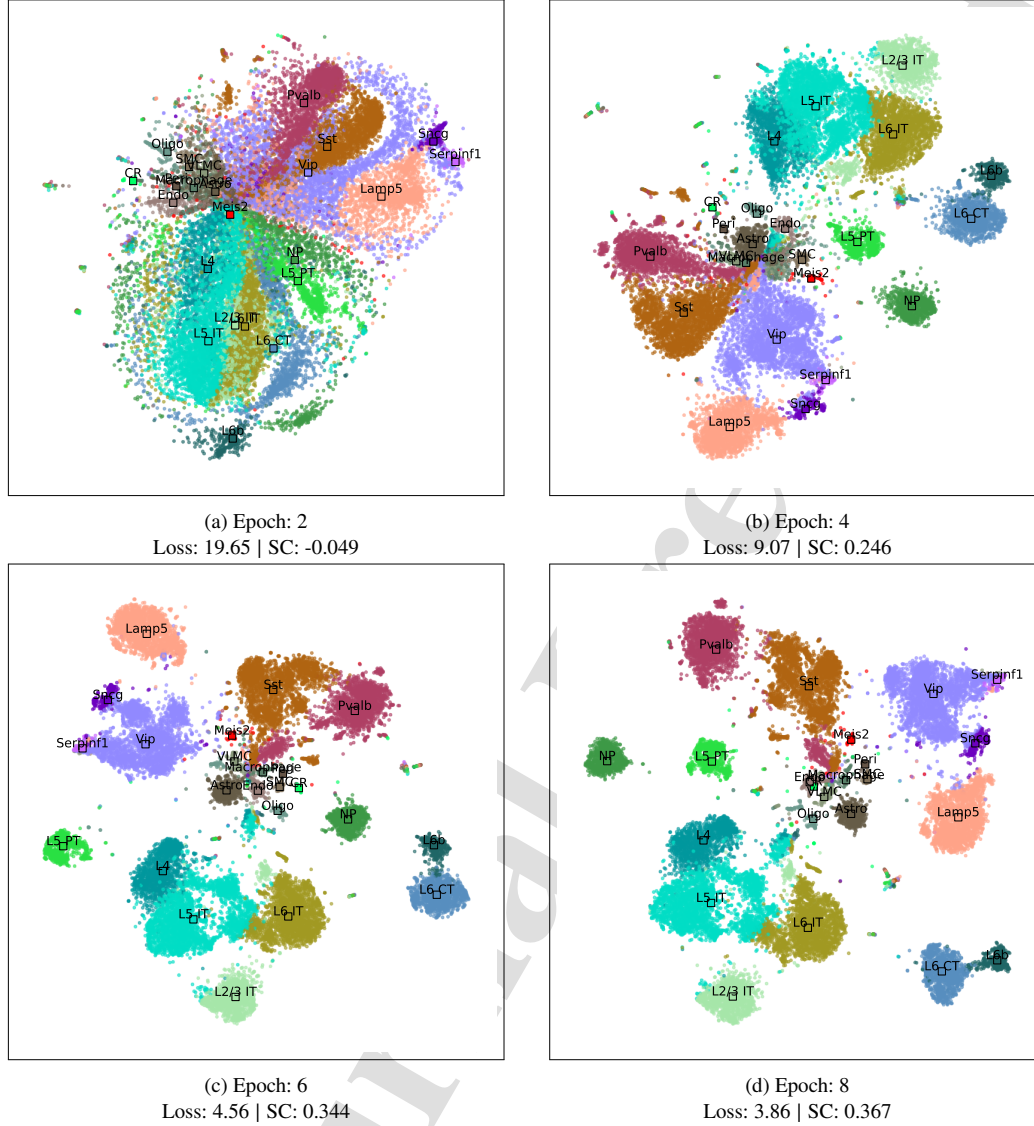
**Figure 8: Comparison between the weighted t-SNE and the activation of the neurons in the last layer of a neural network.** On the left is the same weighted t-SNE presented in Fig. 2j, and on the right is the neurons' activation of the last layer (before the softmax function) of the same neural network. Because this layer only has two neurons, the activation of the first neuron is on the x-axis and the activation of the second neuron is on the y-axis. Visualization details of the XOR dataset as in Fig. 2.



**Figure 9: Visualization of the training of a neural network with the XOR dataset.** Each figure shows the weighted t-SNE for the feature scores of a specific training epoch of a neural network. Each caption describes the training epoch, the training loss, and the silhouette coefficient (SC) of the 2D embedding. Visualization details of the XOR dataset as in Fig. 2. An animation of this plot is available at <https://sbcblab.github.io/wtsne/>.

[89], which is a large short-back for interpreting results for data comprising mainly of this kind of features, such as gene expression data. Meanwhile, surrogate models such as Local Interpretable Model-Agnostic Explanations (LIME) [90] rely on defining a neighborhood, which is not well defined for tabular data [11], even though newer methods are exploring solutions for this issue [91, 92]. In this context, the use of weighted t-SNE in conjunction with methods like relevance aggregation can be a new and helpful tool for inspecting neural networks trained on tabular datasets [10]. Together with other metrics, such as the training loss, it can help researchers and practitioners decide if the model is fulfilling their expectations. While outside the scope of the current work, the visualization of deep features using weighted t-SNE is also a possible future direction discussed in Section 5. It would include adapting weighted t-SNE for unstructured data types, such as images and text, by integrating gradient-based feature attributions or deep learning interpretability techniques.

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

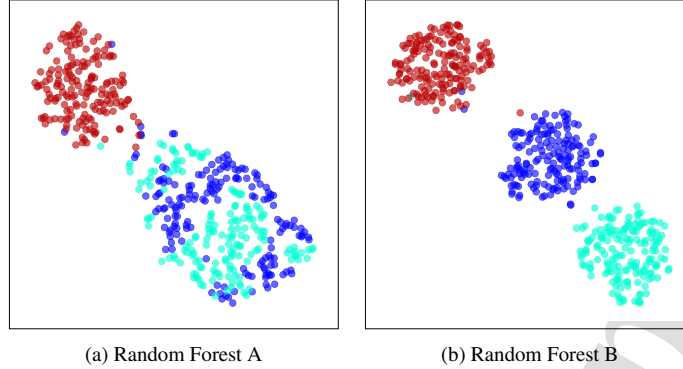


**Figure 10: Visualization of the training of a neural network with the mouse cortex dataset.** Each figure shows the weighted t-SNE for the feature scores of a specific training epoch of a neural network. Each caption describes the training epoch, the training loss, and the silhouette coefficient (SC) of the 2D embedding. Visualization details of the mouse cortex dataset as in Fig. 7. An animation of this plot is available at <https://sbcblab.github.io/wtsne/>.

#### 4.2. Limitations

While weighted t-SNE provides a novel approach to visualizing feature scorer results, it has certain limitations that should be acknowledged and contextualized. Unlike traditional t-SNE, which works directly on raw data, our method requires a precomputed set of scores from a feature scorer. If these scores are unreliable or poorly computed, the visualization may misrepresent the structure of the data. However, this is an inherent challenge in feature scoring itself rather than a flaw in weighted t-SNE. Moreover, this limitation is mitigated by the very purpose of weighted

Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE



**Figure 11: Comparison using weighted t-SNE between two random forests with the same hyperparameters.** Both of them achieved the same training F1-score of 0.99 and test F1-score of 0.93. Even so, they scored features differently, resulting in distinct visualizations. The two random forests were trained to label the red samples from the blue and cyan samples in a binary classification.

**Table 6**

feature importance of the ten top ranking features of the two random forests from Fig. 11.

Random Forest A		Random Forest B	
Feature	Importance	Feature	Importance
rel14	0.150922	rel13	0.126705
rel11	0.144392	rel15	0.112081
rel15	0.116410	rel11	0.111083
rel12	0.100376	rel14	0.075461
rel7	0.046047	rel9	0.068682
rel13	0.036134	rel6	0.046066
rel10	0.033920	rel12	0.043192
rel6	0.027337	rel8	0.040205
rel3	0.021719	rel10	0.037155
rel4	0.019296	rel4	0.026757

t-SNE: it allows users to compare different feature scorers visually, helping identify cases where a scorer produces misleading importance rankings. Unlike standard t-SNE, which is typically used in unsupervised exploratory analysis, weighted t-SNE depends on labeled data to obtain feature importance scores. While this may contradict the usual unsupervised nature of dimensionality reduction, this is a necessary property of most feature scorers, as feature importance usually requires a target variable to be defined. Nonetheless, future work could investigate unsupervised feature scoring techniques to expand the applicability of weighted t-SNE to unlabeled datasets. Like traditional t-SNE, weighted t-SNE inherits some sensitivity to hyperparameters such as perplexity and learning rate. However, weighted t-SNE itself does not significantly increase computational cost beyond standard t-SNE, since the feature weighting step is a simple transformation.

Feature selection methods are often evaluated by classification or regression accuracy, and weighted t-SNE does not directly optimize for these metrics. This is by design, as our goal is not to replace accuracy-based evaluation but to complement it with interpretability and visualization. Feature importance scores affect model behavior in ways that accuracy alone cannot reveal: for example, two models with similar predictive performance may use completely different feature sets, which would be difficult to analyze numerically but can be visually assessed with weighted t-SNE. In future work, a hybrid evaluation strategy could be developed that incorporates both accuracy-based metrics and visualization to provide a more holistic assessment of feature scorers.

Not all high-dimensional datasets exhibit clear cluster structures, making it difficult to interpret whether a visualization is “better” based on apparent separability. However, weighted t-SNE does not assume that data must

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

be clusterable, rather, it shows how feature scorers influence the structure of the data visualization. In cases where no meaningful separation appears, this is itself useful information: it suggests that the scorer may not be effectively capturing feature relevance. Future research could explore alternative evaluation metrics beyond the silhouette coefficient to assess the quality of weighted t-SNE projections in non-clusterable datasets. We used the silhouette score to provide a numerical approximation of the separability observed in the visualizations. However, the silhouette score is not without limitations. It is usually higher for convex clusters than for other types of clusters. Other clustering validity indices, such as the Davies-Bouldin index or Dunn index, could be explored in future work to complement or refine this approach. Since weighted t-SNE is fundamentally a visualization tool, numerical cluster metrics should be interpreted with caution, and user discretion is always recommended.

Weighted t-SNE is, by design, a comparative and contrastive tool. The visualizations are only meaningful when interpreted in relation to a baseline, typically, a regular t-SNE projection of the original dataset without feature scores. A single weighted t-SNE visualization from one scorer, without comparison to a baseline, provides limited insight into how the feature importance scores affect the data structure. Users should always compare weighted t-SNE outputs to the original unweighted t-SNE, as only then can the changes introduced by feature scores be properly assessed.

Our experiments were conducted exclusively on tabular datasets, and the adaptation of weighted t-SNE to other data types, such as text or image data, remains an open challenge. In structured tabular data, feature importance is typically well-defined, making weighted t-SNE straightforward to apply. However, feature importance in unstructured data, such as deep learning feature maps for images or attention weights for text, follows different paradigms.

## 5. Conclusion

This work introduced weighted t-SNE, a novel visualization technique that enhances the inspection of feature scoring algorithms. The method allows users to visually assess feature scorers, providing an intuitive alternative to numerical importance values and predictive metrics. A series of experiments demonstrated the effectiveness of weighted t-SNE across nine feature scorers, two synthetic datasets, and two cancer microarray datasets. The key findings were: Feature scorers with higher classification performance tended to produce clearer clusters in weighted t-SNE visualizations, as confirmed by silhouette coefficient analysis (Table 3, Fig. 7 Table 5). The visualization of feature scores using weighted t-SNE is different from the visualization of feature selection (Fig. 6) or internal model's parameters (Fig. 8). Weighted t-SNE enabled direct visual comparison between scorers, revealing differences in how they structured the data, which would be difficult to discern from raw importance values (Fig. 2, Fig. 3, Fig. 4, Fig. 5). The visualizations generated with weighted t-SNE are trustworthy in regard of the high-dimensional data (Table 4). Outlier detection was improved, as weighted t-SNE made it easier to visually identify samples that were otherwise obfuscated by irrelevant features in high-dimensional space (Fig. 4). In regression tasks, weighted t-SNE successfully reflected the importance of continuous target variables, extending its applicability beyond classification (Fig. 1). Tracking model learning over time was made possible, as weighted t-SNE was used to visualize changes in feature representations across neural network epochs, helping assess how models evolve during training (Fig. 9, Fig. 10). Weighted t-SNE successfully guided feature scorer selection, allowing us to identify feature scorers that improved the F1-score of an independently trained kNN classifier (Table 3). An experiment with two random forests showed that even models with identical predictive performance can assign importance to different features, leading to distinct visualizations, an insight that accuracy-based evaluation alone fails to provide (Fig. 11, Table 6). These results establish weighted t-SNE as a valuable tool for feature importance analysis in machine learning. It enhances feature importance analysis by providing a contrastive, visual tool that reveals differences between feature scorers, aids in outlier detection, tracks model learning, and helps select feature scorers that improve downstream tasks. Unlike accuracy-based evaluation or raw importance values, it uncovers hidden structure, preserves meaningful feature relationships, and integrates seamlessly into machine learning workflows.

More ideas based on weighted t-SNE can be developed in the future. Techniques such as joint t-SNE [93] could be used to improve the comparability between the 2D embeddings of several scorers, and alternatives exploring other algorithms like the UMAP instead of the t-SNE can be tested. One possible future research is regarding the use of weighted t-SNE to visualize the importance of deep features in modern deep learning models. Here we define deep features as abstract features extracted from deep learning models, such as convolutional neural networks or transformers. However, this would require modifications to account for the unique characteristics of deep features, such as their abstract nature and context-dependence (e.g., image, text, or time-series data). The method would need to select a specific deep learning model and specialized feature scorers (e.g., gradient-based methods like Grad-CAM

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

[94] or attention mechanisms [95]), and find the interpretation of high-dimensional embeddings, maybe through sparse autoencoders [96]. Another research direction is investigating the use of weighted t-SNE in unsupervised learning contexts by incorporating unsupervised feature importance metrics or exploring interactive visual analytics tools that allow dynamic comparison of multiple feature scorers on the same dataset.

Feature importance is not just about improving accuracy. It is about understanding what a model has learned. Weighted t-SNE provides a powerful interpretability tool by making feature importance visually accessible, helping users detect outliers, uncover hidden structure, track model learning, and align models with domain knowledge. As deep learning and explainable AI research continue to focus on image and text data [11], many real-world datasets, particularly in science and business, remain tabular [97]. Weighted t-SNE is well-suited for high-dimensional tabular datasets, such as gene expression data, where model interpretability is essential for drawing meaningful scientific conclusions [2, 28, 7]. By integrating weighted t-SNE into machine learning pipelines, researchers and practitioners can gain deeper insights into feature importance, ensuring that models not only perform well but also make decisions that align with human understanding.

## 6. Data and code availability

Interactive versions of all the figures and animations of Fig. 9 and Fig. 10 can be accessed in: <https://sbcblab.github.io/wtsne/>. The necessary source code, hyperparameters, and datasets used in the experiments and results can be accessed on GitHub: [https://github.com/sbcblab/weighted\\_tSNE](https://github.com/sbcblab/weighted_tSNE). The microarray data used in this work is available in the CuMiDa repository: <https://sbcblab.inf.ufrgs.br/cumida>.

## CRedit authorship contribution statement

**Bruno Iochins Grisci:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Writing - Review and Editing, Visualization. **Mario Inostroza-Ponta:** Methodology, Validation, Writing - Review and Editing. **Márcio Dorn:** Methodology, Validation, Resources, Writing - Review and Editing, Supervision, Project administration, Funding acquisition.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work was supported by grants from the Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) [24/2551-0001392-0; 23/2551-0001894-2; 19/2551-0001906-8; 24/2551-0000725-3], Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) [314082/2021-2; 440279/2022-4; 408154/2022-5 and 404319/2024-6], and the Emerging Leaders in the Americas Program Scholarship with the support of the Government of Canada. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## References

- [1] J.-T. Sohns, M. Schmitt, F. Jirasek, H. Hasse, H. Leitte, Attribute-based explanation of non-linear embeddings of high-dimensional data, *IEEE Transactions on Visualization and Computer Graphics* 28 (2021) 540–550.
- [2] B. C. Feltes, E. B. Chandelier, B. I. Grisci, M. Dorn, Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research, *Journal of Computational Biology* 26 (2019) 376–386.
- [3] B. C. Feltes, J. D. F. Poloni, M. Dorn, Benchmarking and Testing Machine Learning Approaches with BARRA:CuRD, a Curated RNA-Seq Database for Cancer Research, *Journal of Computational Biology* 28 (2021) 931–944.
- [4] E. Avila, A. B. Felkl, P. Graebin, C. P. Nunes, C. S. Alho, Forensic characterization of brazilian regional populations through massive parallel sequencing of 124 snps included in hid ion ampliseq identity panel, *Forensic Science International: Genetics* 40 (2019) 74–84.
- [5] E. Avila, A. Kahmann, C. Alho, M. Dorn, Hemogram data as a tool for decision-making in covid-19 management: applications to resource scarcity scenarios, *PeerJ* 8 (2020) e9482.
- [6] V. Formica, M. Minieri, S. Bernardini, M. Ciotti, C. D'Agostini, M. Roselli, M. Andreoni, C. Morelli, G. Parisi, M. Federici, et al., Complete blood count might help to identify subjects with high probability of testing positive to sars-cov-2, *Clinical Medicine* 20 (2020) e114–9.



## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

- [7] M. Dorn, B. I. Grisci, P. H. Narloch, B. C. Feltes, E. Avila, A. Kahmann, C. S. Alho, Comparison of machine learning techniques to handle imbalanced covid-19 cbc datasets, *PeerJ Computer Science* 7 (2021) e670.
- [8] R. J. Lyon, B. Stappers, S. Cooper, J. Brooke, J. Knowles, Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach, *Monthly Notices of the Royal Astronomical Society* 459 (2016) 1104–1123.
- [9] C. O. Sakar, S. O. Polat, M. Katircioglu, Y. Kastro, Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and lstm recurrent neural networks, *Neural Computing and Applications* 31 (2019) 6893–6908.
- [10] B. I. Grisci, M. J. Krause, M. Dorn, Relevance aggregation for neural networks interpretability and knowledge discovery on tabular data, *Information Sciences* 559 (2021) 111–129.
- [11] C. Molnar, *Interpretable Machine Learning*, 2 ed., 2022. URL: <https://christophm.github.io/interpretable-ml-book>.
- [12] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, Sydney NSW Australia, 2017, pp. 3145–3153.
- [13] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, *Journal of Machine Learning Research* 9 (2008) 2579–2605.
- [14] J. Xia, Y. Zhang, J. Song, Y. Chen, Y. Wang, S. Liu, Revisiting dimensionality reduction techniques for visual cluster analysis: An empirical study, *IEEE Transactions on Visualization and Computer Graphics* 28 (2021) 529–539.
- [15] T. May, A. Bannach, J. Davey, T. Ruppert, J. Kohlhammer, Guiding feature subset selection with an interactive visualization, in: *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2011, pp. 111–120.
- [16] S. Mukherjee, t-sne based feature extraction technique for multi-layer perceptron neural network classifier, in: *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, IEEE, 2017, pp. 660–664.
- [17] M.-a. Li, X.-y. Luo, J.-f. Yang, Extracting the nonlinear features of motor imagery eeg using parametric t-sne, *Neurocomputing* 218 (2016) 371–381.
- [18] W. Xi, Z. Li, Z. Tian, Z. Duan, A feature extraction and visualization method for fault detection of marine diesel engines, *Measurement* 116 (2018) 429–437.
- [19] F. L. Dennig, T. Polk, Z. Lin, T. Schreck, H. Pfister, M. Behrisch, Fdive: Learning relevance models using pattern-based similarity measures, in: *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2019, pp. 69–80.
- [20] E. Artur, R. Minghim, A novel visual approach for enhanced attribute analysis and selection, *Computers & Graphics* 84 (2019) 160–172.
- [21] J. Krause, A. Perer, E. Bertini, Infuse: Interactive feature selection for predictive modeling of high dimensional data, *IEEE Transactions on Visualization and Computer Graphics* 20 (2014) 1614–1623.
- [22] J. M. G. Junior, F. M. Lopes, Interpretability with relevance aggregation in neural networks for absenteeism prediction, in: *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, IEEE, 2022, pp. 01–04.
- [23] Y. Zhao, Z. Huang, L. Gong, Y. Zhu, Q. Yu, Y. Gao, Evaluating the impact of data transformation techniques on the performance and interpretability of software defect prediction models, *IET Software* 2023 (2023).
- [24] W. Lu, X. Yan, Variable-weighted fda combined with t-sne and multiple extreme learning machines for visual industrial process monitoring, *ISA transactions* 122 (2022) 163–171.
- [25] S. Ma, G. Cheng, Y. Li, R. Zhao, Dimension reduction method of high-dimensional fault datasets based on c\_m\_t-sne under unsupervised background, *Measurement* 214 (2023) 112835.
- [26] C. Lazar, J. Taminiau, S. Megack, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, A. Nowe, A survey on filter techniques for feature selection in gene expression microarray analysis, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9 (2012) 1106–1119.
- [27] J. C. Ang, A. Mirzal, H. Haron, et al., Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13 (2016) 971–989.
- [28] B. I. Grisci, B. C. Feltes, M. Dorn, Neuroevolution as a tool for microarray gene expression pattern identification in cancer research, *Journal of Biomedical Informatics* 89 (2019) 122–133.
- [29] I. A. Gheyas, L. S. Smith, Feature subset selection in large dimensionality domains, *Pattern Recognition* 43 (2010) 5–13.
- [30] W. H. Kruskal, W. A. Wallis, Use of ranks in one-criterion variance analysis, *Journal of the American statistical Association* 47 (1952) 583–621.
- [31] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27 (2005) 1226–1238.
- [32] J. R. Vergara, P. A. Estévez, A review of feature selection methods based on mutual information, *Neural Computing and Applications* 24 (2014) 175–186.
- [33] I. Kononenko, E. Šimec, M. Robnik-Šikonja, Overcoming the myopia of inductive learning algorithms with relief, *Applied Intelligence* 7 (1997) 39–55.
- [34] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relief and rrelief, *Machine learning* 53 (2003) 23–69.
- [35] V. Fonti, E. Belitser, Feature selection using lasso, *VU Amsterdam Research Paper in Business Analytics* 30 (2017) 1–25.
- [36] L. Breiman, Random forests, *Machine Learning* 45 (2001) 5–32.
- [37] H.-P. Kriegel, P. Kröger, A. Zimek, Subspace clustering, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2 (2012) 351–364.
- [38] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, *ACM sigkdd explorations newsletter* 6 (2004) 90–105.
- [39] V. Bolón-Canedo, N. Sánchez-Marono, A. Alonso-Betanzos, J. M. Benítez, F. Herrera, A review of microarray datasets and applied feature selection methods, *Information Sciences* 282 (2014) 111–135.
- [40] B. I. Grisci, B. C. Feltes, M. Dorn, Microarray classification and gene selection with fs-neat, in: *2018 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Rio de Janeiro, Brazil, 2018, pp. 1–8.
- [41] B. C. Feltes, B. I. Grisci, J. de Faria Poloni, M. Dorn, Perspectives and applications of machine learning for evolutionary developmental biology, *Molecular Omics* 14 (2018) 289–306.



## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

- [42] A.-L. Boulesteix, C. Strobl, T. Augustin, M. Daumer, Evaluating microarray-based classifiers: an overview, *Cancer Informatics* 6 (2008) 77–97.
- [43] R. Powers, M. Goldszmidt, I. Cohen, Short term performance forecasting in enterprise systems, in: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, ACM, Chicago Illinois USA, 2005, pp. 801–807.
- [44] B. I. Grisci, B. C. Feltes, J. de Faria Poloni, P. H. Narloch, M. Dorn, The use of gene expression datasets in feature selection research: 20 years of inherent bias?, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2023) e1523.
- [45] M. C. Barbieri, B. I. Grisci, M. Dorn, Analysis and comparison of feature selection methods towards performance and stability, *Expert Systems with Applications* (2024) 123667.
- [46] S.-O. Shim, M. H. Alkinani, L. Hussain, W. Aziz, Feature ranking importance from multimodal radiomic texture features using machine learning paradigm: A biomarker to predict the lung cancer, *Big Data Research* 29 (2022) 100331.
- [47] D. Kobak, P. Berens, The art of using t-sne for single-cell transcriptomics, *Nature Communications* 10 (2019) 1–14.
- [48] A. Diaz-Papkovich, L. Anderson-Trocmé, S. Gravel, A review of umap in population genetics, *Journal of Human Genetics* 66 (2021) 85–91.
- [49] M. Mramor, G. Leban, J. Demšar, B. Zupan, Visualization-based cancer microarray data classification analysis, *Bioinformatics* 23 (2007) 2147–2154.
- [50] K. Pearson, On lines and planes of closest fit to systems of points in space, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (1901) 559–572.
- [51] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, *arXiv preprint arXiv:1802.03426* (2018).
- [52] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, E. Stanley, Dna visual and analytic data mining, in: *Proceedings. Visualization'97 (Cat. No. 97CB36155)*, IEEE, Phoenix, AZ, USA, 1997, pp. 437–441.
- [53] T. Kohonen, Self-organized formation of topologically correct feature maps, *Biological cybernetics* 43 (1982) 59–69.
- [54] P. G. Polícar, M. Stražar, B. Zupan, Embedding to reference t-sne space addresses batch effects in single-cell classification, *Machine Learning* (2021) 1–20.
- [55] L. Van Der Maaten, Accelerating t-sne using tree-based algorithms, *The Journal of Machine Learning Research* 15 (2014) 3221–3245.
- [56] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, Y. Kluger, Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data, *Nature Methods* 16 (2019) 243–245.
- [57] E. Z. Macosko, A. Basu, R. Satija, J. Nemesh, K. Shekhar, M. Goldman, I. Tirosh, A. R. Bialas, N. Kamitaki, E. M. Martersteck, et al., Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets, *Cell* 161 (2015) 1202–1214.
- [58] M. Wattenberg, F. Viégas, I. Johnson, How to use t-sne effectively, *Distill* (2016).
- [59] J. A. Lee, D. H. Peluffo-Ordóñez, M. Verleysen, Multi-scale similarities in stochastic neighbour embedding: Reducing dimensionality while preserving both local and global structure, *Neurocomputing* 169 (2015) 246–261.
- [60] X. Chen, F. Kopsaftopoulos, Q. Wu, H. Ren, F.-K. Chang, Flight state identification of a self-sensing wing via an improved feature selection method and machine learning approaches, *Sensors* 18 (2018) 1379.
- [61] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics* 20 (1987) 53–65.
- [62] M. Espadoto, R. M. Martins, A. Kerren, N. S. Hirata, A. C. Telea, Toward a quantitative survey of dimension reduction techniques, *IEEE transactions on visualization and computer graphics* 27 (2019) 2153–2173.
- [63] L. Van Der Maaten, Learning a parametric embedding by preserving local structure, in: *Artificial intelligence and statistics*, PMLR, 2009, pp. 384–391.
- [64] R. M. Martins, R. Minghim, A. C. Telea, Explaining Neighborhood Preservation for Multidimensional Projections, in: R. Borgo, C. Turkey (Eds.), *Computer Graphics and Visual Computing (CGVC)*, The Eurographics Association, 2015.
- [65] C. B. Horan, Multidimensional scaling: Combining observations when individuals have different perceptual structures, *Psychometrika* 34 (1969) 139–165.
- [66] J. D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition, *Psychometrika* 35 (1970) 283–319.
- [67] M. Roberts, D. Driggs, M. Thorpe, J. Gibbey, M. Yeung, S. Ursprung, A. I. Aviles-Rivero, C. Etmann, C. McCague, L. Beer, et al., Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans, *Nature Machine Intelligence* 3 (2021) 199–217.
- [68] R. Geirhos, et al., Shortcut learning in deep neural networks, *Nat. Mach. Intell.* 2 (2020) 665–673.
- [69] M. Tan, M. Hartley, M. Bister, R. Deklerck, Automated feature selection in neuroevolution, *Evolutionary Intelligence* 1 (2009) 271–292.
- [70] I. Guyon, Design of experiments of the nips 2003 variable selection benchmark, in: *NIPS 2003 Workshop on Feature Extraction and Feature Selection*, volume 253, Whistler, 2003, pp. 1–30.
- [71] F. Pedregosa, G. Varoquaux, E. Duchesnay, et al., Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12 (2011) 2825–2830.
- [72] G. Celeux, M. El Anbari, J.-M. Marin, C. P. Robert, et al., Regularization in regression: comparing bayesian and frequentist methods in a poorly informative situation, *Bayesian Analysis* 7 (2012) 477–502.
- [73] B. Tasic, Z. Yao, L. T. Graybuck, K. A. Smith, T. N. Nguyen, D. Bertagnolli, J. Goldy, E. Garren, M. N. Economo, S. Viswanathan, et al., Shared and distinct transcriptomic cell types across neocortical areas, *Nature* 563 (2018) 72–78.
- [74] F. Kamalov, F. Thabtah, H. H. Leung, Feature selection in imbalanced data, *Annals of Data Science* (2022) 1–15.
- [75] C. R. Harris, K. J. Millman, R. Gommers, T. E. Oliphant, et al., Array programming with NumPy, *Nature* 585 (2020) 357–362.
- [76] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, J. E. Snyder-Cappione, Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets, *Nature communications* 10 (2019) 1–12.
- [77] C. J. Stone, *Classification and Regression Trees*, 1st ed., Taylor & Francis Group, LLC, Boca Raton, FL, 1984.

## Assessing Feature Scorer Results on High-Dimensional Datasets with t-SNE

- [78] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning* 20 (1995) 273–297.
- [79] C. C. Aggarwal, A. Hinneburg, D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, in: J. Van den Bussche, V. Vianu (Eds.), *Database Theory — ICDT 2001*, Springer, Berlin, Heidelberg, 2001, pp. 420–434.
- [80] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician* 46 (1992) 175–185.
- [81] A. Tharwat, Classification assessment methods, *Applied Computing and Informatics* 17 (2021) 168–192.
- [82] R. Garcia, A. C. Telea, B. C. da Silva, J. Tørresen, J. L. D. Comba, A task-and-technique centered survey on visual analytics for deep learning model engineering, *Computers & Graphics* 77 (2018) 30–49.
- [83] J.-B. Lamy, B. Sekar, G. Guezennec, J. Bouaud, B. Séroussi, Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach, *Artificial Intelligence in Medicine* 94 (2019) 42–53.
- [84] M. O. Prates, P. H. Avelar, L. C. Lamb, Assessing gender bias in machine translation: a case study with google translate, *Neural Computing and Applications* 32 (2020) 6363–6381.
- [85] T. Hastie, R. Tibshirani, J. H. Friedman, J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, 2 ed., Springer, New York, NY, USA, 2009.
- [86] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, *Plos One* 10 (2015) 1–46.
- [87] G. D. Cantareira, E. Etemad, F. V. Paulovich, Exploring neural network hidden layer activity using vector fields, *Information* 11 (2020) 426.
- [88] G. Montavon, W. Samek, K.-R. Müller, Methods for interpreting and understanding deep neural networks, *Digital Signal Processing* 73 (2018) 1–15.
- [89] X. Huang, A. Khetan, M. Cvitkovic, Z. Karnin, Tabtransformer: Tabular data modeling using contextual embeddings, *arXiv preprint arXiv:2012.06678* (2020).
- [90] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, San Francisco California USA, 2016, pp. 1135–1144.
- [91] A. Björklund, A. Henelius, E. Oikarinen, K. Kallonen, K. Puolamäki, Sparse robust regression for explaining classifiers, in: *International Conference on Discovery Science*, Springer, 2019, pp. 351–366.
- [92] A. Björklund, K. Mäkelä, K. Puolamäki, Slisemap: Supervised dimensionality reduction through local explanations, *Machine Learning* (2022).
- [93] Y. Wang, L. Chen, J. Jo, Y. Wang, Joint t-sne for comparable projections of multiple high-dimensional datasets, *IEEE Transactions on Visualization and Computer Graphics* 28 (2021) 623–632.
- [94] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, Venice, Italy, 2017, pp. 618–626.
- [95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: *Advances in Neural Information Processing Systems*, NIPS, Long Beach, CA, USA, 2017, pp. 5998–6008.
- [96] C. O'Neill, C. Ye, K. Iyer, J. F. Wu, Disentangling dense embeddings with sparse autoencoders, *arXiv preprint arXiv:2408.00657* (2024).
- [97] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, *IEEE Transactions on Neural Networks and Learning Systems* (2022).

## Highlights

### **Weighted t-SNE: Assessing Feature Scorer Results on High-Dimensional Datasets**

Bruno Iochins Grisci, Mario Inostroza-Ponta, Márcio Dorn

- We propose an extension of t-SNE to visualize the results of feature scorers.
- Relevant features have more influence on the position of points in the projection.
- We perform experiments on nine feature scorers and six datasets.
- Weighted t-SNE can be used to compare and choose the best feature scorer visually.
- It can be used to increase the interpretability of machine learning models.

**Bruno Iochins Grisci**

Bruno Iochins Grisci is a Professor in the Department of Theoretical Informatics at the Federal University of Rio Grande do Sul (UFRGS). He holds a PhD, MSc, and BSc in Computer Science from UFRGS. His research focuses on explainable AI, machine learning, metaheuristics, bioinformatics, and data visualization. He has conducted research at Unisinos (with Dell Inc.) and completed exchange programs at the University of Birmingham, the University of Santiago de Chile, Karlsruhe Institute of Technology, and Dalhousie University. He is an alumnus of the Heidelberg Laureate Forum and the Emerging Leaders in the Americas Program.

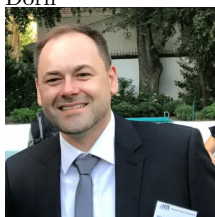
**Mario Inostroza-Ponta**

Mario Inostroza-Ponta received his Computer Engineering degree from the Universidad de Santiago de Chile, Santiago, Chile, in 2001, and his Ph.D. in Computer Science from the University of Newcastle, Callaghan, NSW, Australia, in 2008. He is currently an Associate Professor with the Departamento de Ingeniería Informática, Universidad de Santiago de Chile. He spent a year as a Post-Doctoral Researcher at the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil. His current research interests include the development of ad-hoc computational techniques for bioinformatics and other areas, using data mining, metaheuristics, and multiobjective optimization.

**Márcio Dorn**

Márcio Dorn received his M.Sc. degree in Computer Science from the Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil, in 2008, and his Ph.D. degree in Computer Science from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2012. He is a Professor at the Institute of Informatics, UFRGS, where he also leads the Structural Bioinformatics and Computational Biology Laboratory. He was a Research Associate at the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2008, 2009, and 2017. His current research interests include bioinformatics, structural bioinformatics, machine learning, metaheuristics, artificial intelligence, and high-performance computing. Prof. Dorn is a CNPq (Brazilian National Research Council) Advanced Fellow and an Alexander von Humboldt Research Fellow.

Dorn



Inostroza



Grisci



**Declaration of interests**

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: