

OPEN

Assessing phenotype order in molecular data

Ludwig Lausser, Lisa M. Schäfer, Lyn-Rouven Schirra, Robin Szekely, Florian Schmid & Hans A. Kestler 

Biological entities are key elements of biomedical research. Their definition and their relationships are important in areas such as phylogenetic reconstruction, developmental processes or tumor evolution. Hypotheses about relationships like phenotype order are often postulated based on prior knowledge or belief. Evidence on a molecular level is typically unknown and whether total orders are reflected in the molecular measurements is unclear or not assessed. In this work we propose a method that allows a fast and exhaustive screening for total orders in large datasets. We utilise ordinal classifier cascades to identify discriminable molecular representations of the phenotypes. These classifiers are constrained by an order hypothesis and are highly sensitive to incorrect assumptions. Two new error bounds, which are introduced and theoretically proven, lead to a substantial speed-up and allow the application to large collections of many phenotypes. In our experiments we show that by exhaustively evaluating all possible candidate orders, we are able to identify phenotype orders that best coincide with the high-dimensional molecular profiles.

Assessing the correspondence between observable phenotypes and their underlying molecular background is a challenging task in molecular biology. Even for pairwise comparisons it is not straight forward to confirm hypothesised relations in high-dimensional marker representations.

This becomes even more evident for higher order relations among multiple phenotypes. In this case, local events and global processes might be confused as they both can lead to the same pattern of observable phenotypes. While local - pairwise - events might be reflected by any type of pairwise differences, an overall connecting pattern is required for global processes. An example, for such higher level relations are ordinal phenotypes of type $phenotype_1 < phenotype_2 < phenotype_3$, as they might occur in developmental processes¹⁻⁴, like embryogenesis⁵, phylogenetic reconstruction⁶⁻⁸ or diagnostic stagings or gradings⁹⁻¹². Their observable representations suggest an order of the phenotypes ($<$), which might lead to hypotheses on a connecting “ordinal” relation or process on a molecular level (Fig. 1B). Providing evidence for these hypotheses is quite challenging due to the high dimensionality of molecular profiles. Being defined for univariate categorical variables, the concept of ordinality can be embedded in many different ways in a multivariate real-valued feature representation. There might also be several ordinal relations that coexist in parallel.

In the research field of ordinal classification usually, a known order is used to improve classification performance. The assumption is that the given order between the classes (phenotypes) can be mapped to the given representation and hence also holds in the feature space. In this work, we instead propose a method that can check whether the reflection is provided, by elaborating a performance-based criterion for detecting and comparing ordinal structures in multivariate feature representations. We present an algorithm (CASCADES) that allows for systematic and exhaustive screens through the search space of all phenotype orders. It is applicable for extracting a small set of candidate orders from a feature representation that fulfils a minimal generalisation ability of a predictive model. Based on supervised classification, our method uses the canonical paradigm for learning relationships between raw uninterpreted feature representations and semantically meaningful phenotypes (classes, categories, concepts, etc.)¹³⁻¹⁵. Utilising feature representations and class memberships these techniques allow the extraction of phenotype-specific patterns and the construction of phenotype separating boundaries. In this way, classifiers identify characteristics of phenotypes or even learn the key attributes of their concepts. Mainly designed for discrimination the learning processes of classification algorithms often neglect the semantic relationships among classes. Standard training algorithms would neither request nor reconstruct such dependencies explicitly¹⁶⁻¹⁸.

Institute of Medical Systems Biology, Ulm University, 89069, Ulm, Germany. Ludwig Lausser, Lisa M. Schäfer and Lyn-Rouven Schirra contributed equally. Correspondence and requests for materials should be addressed to H.A.K. (email: hans.kestler@uni-ulm.de)

Received: 18 January 2019

Accepted: 29 July 2019

Published online: 13 August 2019

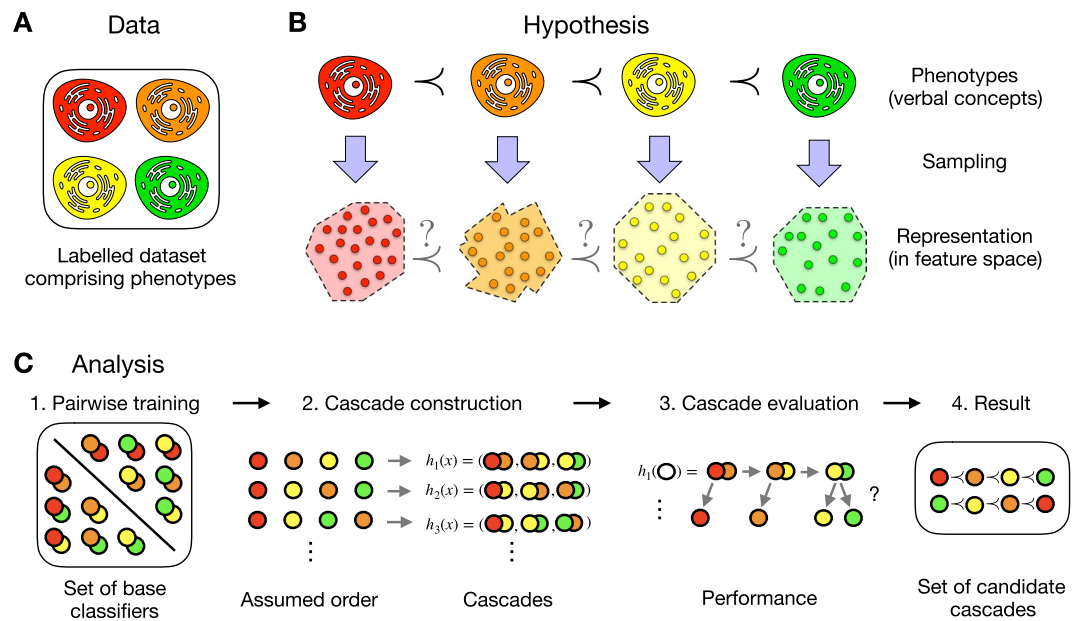


Figure 1. Assessment of phenotype order in molecular data (four phenotype/class example). **(A)** (Data): The data consists of a set of molecular measurements labelled with their phenotype names. **(B)** (Hypothesis): Is an ordinal relation on the phenotypes reflected in feature space? A trend in observable characteristics of phenotypes often induces hypotheses about ordinality on their molecular background. Its confirmation requires a reflection in the corresponding feature representation. **(C)** (Analysis): The analysis incorporates four different steps and evaluates which hypotheses are reflected in the specific feature representation. First, for all pairwise combinations binary base classifiers are trained in a pairwise manner. Second, for each possible order a cascade is built using the previously trained base classifiers. Third, the performance of these ensemble classifiers is analysed. The cascade classifier analyses each sample in a stepwise manner. For each base classifier it holds that if the sample belongs to its first class it is labelled and if not it is transferred to the next classifier. By doing so, regions of the feature space are defined and labelled dependent on the order of the base classifiers' first class. Finally, a set of possible candidate cascades that show a good performance measured by the class-wise minimal sensitivity is returned.

For this reason, we focus on ordinal classifier cascades¹⁹ of binary base classifiers. They are a specialisation of general decision lists^{20–22}. A predefined order of phenotypes constrains the learning algorithm of an ordinal classifier cascade. Initially designed for guiding the learning process, we showed that wrong assumptions on the class order can lead to severely decreased detection rates of an ordinal classifier cascade²³.

In our approach (Fig. 1), we separate the training and evaluation of binary base classifiers from the construction of the classifier cascade. For the training, no order information is used, and each base classifier is trained independently. But the assumed phenotype order defines the evaluation sequence of these pairwise base classifiers. In each step of the evaluation a feature space region is labelled as decision region for a specific class, and the remaining space stays unlabelled. If the order is wrong, then samples of classes that are later in the order already lie within this region, whereas if it reflects the order they lie within the unlabelled region. Although trained only pairwise, the base classifiers show good performance when used to distinguish between a class and all following classes in the order.

Here, we utilise this susceptibility as a clear cut criterion for discriminating between class orders that allow a high generalisation ability or not. We provide theoretical upper bounds on the class-wise sensitivities of ordinal classifier cascades, which would enable the proposed algorithm to scale up to large collections of phenotypes. The combination of the pairwise training scheme and these bounds lead to a complexity reduction, as the number of base classifier trainings, in a single train-test experiment, for n classes is reduced from $(n-1)n!$ to $(n-1)n$, and the number of cascade constructions and trainings is in the worst case $n!$ but decreases by the number of cascades that do not pass this bound. We could show the utility of our method to identify reflected orderings in experimental evaluations on artificial data and gene expression profiles of developmental and ageing phenotypes.

Results

We evaluated the ability of the CASCADES algorithm to detect reflected orders in feature space based on artificial and existing gene expression datasets (see Methods). For our analysis a linear support vector machine (SVM)¹⁸ was chosen as a base classifier for the ordinal cascades due to its superior performance²³. The SVM was imported from the LIBSVM library²⁴. Its cost parameter was fixed to a value of one.

The performance of the ordinal cascades as well as their base classifiers were evaluated in 10×10 cross-validation (CV) experiments²⁵. The 10×10 CV experiments were repeated for all class orders ($\mathcal{J}!$ experiments) and the performance was measured in terms of minimal class-wise sensitivity. All classification experiments were performed with help of the TunePareto software²⁶.

Id	Name	Ordinal Cascades	Bound	Min Sens.
d_1 :	linear (10!–2 cascades rejected)	(a) $y_1 \prec \dots \prec y_{10}$	100%	100%
		(b) $y_{10} \prec \dots \prec y_1$	100%	100%
d_2 :	curved (10!–2 cascades rejected)	(a) $y_1 \prec \dots \prec y_{10}$	100%	100%
		(b) $y_{10} \prec \dots \prec y_1$	100%	100%
d_3 :	non-ordinal (all cascades rejected)	—	—	—
d_4 :	Drosophila melanogaster (21 cascades rejected)	(a) <i>embryo</i> \prec <i>larva</i> \prec <i>pupa</i> \prec <i>adult</i>	94.4%	89.4%
		(b) <i>adult</i> \prec <i>pupa</i> \prec <i>larva</i> \prec <i>embryo</i>	72.3%	72.3%
		(c) <i>adult</i> \prec <i>pupa</i> \prec <i>embryo</i> \prec <i>larva</i>	71.0%	71.0%
d_5 :	Danio rerio (117 cascades rejected)	(a) <i>adult</i> ₂ \prec <i>adult</i> ₁ \prec <i>embryo</i> ₃ \prec <i>embryo</i> ₂ \prec <i>embryo</i> ₁	85.0%	85.0%
		(b) <i>adult</i> ₂ \prec <i>adult</i> ₁ \prec <i>embryo</i> ₃ \prec <i>embryo</i> ₁ \prec <i>embryo</i> ₂	85.0%	85.0%
		(c) <i>adult</i> ₂ \prec <i>adult</i> ₁ \prec <i>embryo</i> ₁ \prec <i>embryo</i> ₂ \prec <i>embryo</i> ₃	54.7%	54.7%
d_6 :	Human muscle adaptation (20 cascades rejected)	(a) <i>age</i> ₁ \prec <i>age</i> ₂ \prec <i>age</i> ₃ \prec <i>age</i> ₄	73.1%	62.5%
		(b) <i>age</i> ₁ \prec <i>age</i> ₂ \prec <i>age</i> ₄ \prec <i>age</i> ₃	73.1%	62.5%
		(c) <i>age</i> ₃ \prec <i>age</i> ₄ \prec <i>age</i> ₂ \prec <i>age</i> ₁	50.5%	46.1%
		(d) <i>age</i> ₃ \prec <i>age</i> ₄ \prec <i>age</i> ₁ \prec <i>age</i> ₂	50.5%	31.6%
d_7 :	Caenorhabditis elegans, stages (116 cascades rejected)	(a) <i>stage</i> ₁ \prec <i>stage</i> ₂ \prec <i>stage</i> ₃ \prec <i>stage</i> ₄ \prec <i>stage</i> ₅	91.7%	91.7%
		(b) <i>stage</i> ₅ \prec <i>stage</i> ₄ \prec <i>stage</i> ₃ \prec <i>stage</i> ₂ \prec <i>stage</i> ₁	75.0%	66.7%
		(c) <i>stage</i> ₁ \prec <i>stage</i> ₂ \prec <i>stage</i> ₃ \prec <i>stage</i> ₅ \prec <i>stage</i> ₄	58.6%	58.6%
		(d) <i>stage</i> ₅ \prec <i>stage</i> ₄ \prec <i>stage</i> ₃ \prec <i>stage</i> ₁ \prec <i>stage</i> ₂	50.0%	50.0%
d_8 :	Caenorhabditis elegans, time (10!–1 cascades rejected)	(a) $t_1 \prec t_2 \prec t_3 \prec t_4 \prec t_5 \prec t_6 \prec t_7 \prec t_8 \prec t_9 \prec t_{10}$	66.7%	66.7%
d_9 :	Various cancer cell lines (all cascades rejected)	—	—	—

Table 1. Evaluation of the CASCADES algorithm on the artificial datasets d_1, \dots, d_3 and the real datasets d_4, \dots, d_9 . Screening experiments were performed for sensitivity thresholds $t \in \{1.00, 0.95, \dots, 0.50\}$. Each screening experiment comprises all $|\mathcal{Y}|!$ class orders. All candidate cascades that pass a minimal sensitivity threshold bound $t \geq 0.5$ are reported. The bound and the real minimal class-wise sensitivities are reported. The highest bounds and minimal class-wise sensitivities are marked in boldface.

Artificial datasets. We performed experiments on three different kinds of artificial datasets, two of them reflect an ordered sequence of sample clouds in the feature space (d_1 and d_2) and one does not (d_3) (Supplementary Fig. S1). Each dataset comprises $|\mathcal{Y}| = 10$ classes ($i = 1, \dots, 10$) of 100 samples. The instances of class y are drawn independent and identically distributed from a normal distribution $\mathbf{x} \sim \mathcal{N}(\mathbf{x}_y, sd)$ in dependence of a class specific centroid $\mathbf{m}_y \in \mathbb{R}^2$. The standard deviation sd was identical for all classes. Experiments were performed for standard deviations $sd \in \{0.1, 0.2, \dots, 1.0\}$.

The analysis on the simple artificial datasets show that our method can distinguish between phenotype orders that are reflected and are not reflected in the two dimensional data. The ordinal assumption is imposed by a common increase in both features. The results for $sd = 0.2$ are given in Table 1. It can be seen that for d_1 and d_2 the correct order and its inverse are returned. All other possible orders show a minimal class-wise sensitivity lower than 50%. For d_3 (non-ordinal) no order passed the threshold of 50%.

We additionally analysed the performance in dependency of the standard deviation of the artificial data clouds. For the dataset d_1 (linear) and d_2 (curved) the sensitivities under the correct assumption decline with increasing standard deviations (Supplementary Fig. S2). The corresponding bounds lie above the real sensitivities. For the wrong order, the sensitivity of at least one class drops. In the given example the sensitivities of classes y_1 to y_5 and the corresponding bounds are mainly identical to those of the correct class order but largest changes can be observed for classes y_6 and y_7 . For the non-ordinal dataset d_3 the minimal class-wise sensitivity is zero independent of the standard deviation (Supplementary Fig. S2).

For each dataset and each setting, all $10! \approx 3.6 \cdot 10^6$ possible class orders are screened and the number of remaining class orders is reported (Supplementary Fig. S3). Datasets d_1 and d_2 show comparable results. Our method returned at most four candidate cascades for each experiment. With increasing standard deviation the distinction of classes became harder. Candidate cascades could only fulfil lower sensitivity thresholds t . The bounds of all rejected cascades predicted minimal sensitivities below 0.5. With lower thresholds the chance of finding more than two candidate cascades increased. As expected, no candidates were proposed for dataset d_3 (non-ordinal). Evaluating the real minimal class-wise sensitivities of the remaining cascades revealed that additional candidates were rejected.

Gene expression datasets. Furthermore, experiments on existing gene expression data were performed (see Methods). We chose ordinal multi-class expression data for which the classes correspond to specific points in time of a process. In three datasets (d_4, d_5, d_7) the classes correspond to developmental stages of *Drosophila*

*melanogaster*¹, *Danio rerio*², and *Caenorhabditis elegans*⁴. Additionally, d_7 was used with a different labelling. This further labelling was not based on stages but based on the point in time at which the sample was taken. The further dataset in our analysis d_6 comprises transcriptome samples of human muscles³. The data was categorised into four classes according to the age (in years) of the participants. For all these datasets it is expected that the assumed order based on the order of points in time is reflected within the expression profiles. To test our method (CASCADES algorithm) on real data for which no order is assumed, we included gene expression profiles from cell lines that are derived from 9 different cancer tissue types²⁷.

The results of the real datasets are shown in Table 1. As the performance of the cascade depends on a sensitivity bound which depends only on the performance of the independent base classifiers, first those candidate cascades are reported that pass a sensitivity bound $t \leq 0.5$. For the temporal ordinal datasets d_4 – d_8 , the CASCADES algorithm rejected at least 83.3% of all candidate cascades. No candidate passed the CASCADES algorithm for the non-ordinal dataset d_9 . The number of candidates is further depleted by analysing the minimal class-wise sensitivity of the full cascades. For dataset d_4 , the highest minimal class-wise sensitivity (89.4%) was achieved by the correct class order. It was followed by the correct inverse class order (72.3%) and an incorrect class order (71.0%).

Three candidates passed the CASCADES algorithm for dataset d_5 . The highest minimal class-wise sensitivity was achieved by two candidate cascades (85.0%). Both reflect the inverse class order. The first one corresponds to the correct inverse class order. The second one assumes an incorrect class order $embryo_3 \prec embryo_1 \prec embryo_2$. The third candidate achieved a minimal class-wise sensitivity of 54.7%. A general division between the adult and embryo samples can be observed. The order might be explained by the duration between the different states. Whereas all three embryonic samples cover a range of 10 days after birth, the first adult class comprises samples taken at month 3 and the $adult_2$ class is 1–2 years after birth. As a result the order assumption $embryo_1 \prec embryo_2 \prec embryo_3$ might only be reflected ambiguously.

Four class orders passed the CASCADES algorithm on dataset d_6 . Two of these candidates dropped out due to a minimal class-wise sensitivity lower than 50.0%. The remaining two achieved minimal class-wise sensitivities of 62.5%. One of these class orders corresponds to the correct class order. The second one proposes a partially consistent class order ($age_4 \prec age_3$). As age_3 and age_4 comprise 10 years each and age_1 and age_2 20 years it can be argued, similar to the cascades of d_5 . The two classes age_3 and age_4 might be too similar under the order assumption leading to comparable results.

For dataset d_7 , the minimal class-wise sensitivity of the correct class order outperformed the performance of other candidate cascades (91.7%). All other candidate cascades achieved a minimal class-wise sensitivity of at most 66.7%. Only one candidate cascade passed the CASCADES algorithm when analysed on the level of points in time (d_8). The correct class order gained a minimal class-wise sensitivity of 66.7%, which was achieved for class t_2 . All other classes achieved class-wise sensitivities of at least 80.0%. Especially this dataset shows that our analysis does not aim at improving the classification performance as much as possible, but rather finding the order that outperforms other orders and this independent of a specific performance, as long as the performance is better than 50%.

Discussion

Ordinal relations between phenotypes are often defined on a semantic level. These relations are assumed to be reflected in a given feature representation without evaluating whether these assumptions hold. It might be the case that independent causes lead to ordinal phenotype characteristics or that the order is not reflected in the chosen feature space because the measured features are not responsible for the observed order.

In this work, we present ordinal classification as an example of a supervised learning task that incorporates semantic relations in the training process of classification models. By constraining the learning process, ordinal classification results in a restricted model class, which is no longer able to separate an arbitrary landscape of classes. This property is used to falsify wrong assumptions on the dependencies of the classes and the chosen feature space.

We provide two theoretical upper bounds on the minimal class-wise sensitivity, which are utilised for accelerating the training of ordinal classifier cascades and allow an exhaustive evaluation of all possible class orders. In this way, ordinal classifier cascades are used as an explorative tool to screen for unknown ordinal dependencies. In our experiments, we give examples for up to 10 different classes resulting in the evaluation of over 3.6 million class orders. Although our algorithm requires pairwise training of the ensemble members, both the bounds and the algorithm are independent of the chosen type of base classifier, the binary training type and might be transferred to alternative ensembles.

Our experiments on the artificial data showed that only if any ordinality is reflected in the feature space possible sets of candidate orders are returned. If there is no ordinal sequence reflected no cascade passes the bound of 0.5. No order was detected for non-ordinal datasets. Always the correct order and its reverse were found as dominant order if cascades were returned for the artificial ordinal datasets.

For all datasets independent of the chosen standard deviation, at least 80% of all candidate cascades could be rejected due to minimal class-wise sensitivities lower than 50%. However, although the procedure can reconstruct the correct class order for all datasets, alternative ordinal class structures might be detected. In our experiments, these alternatives differ from the assumed class order in the position of the last two classes. A reason might be the lower number of constraints for these classes.

For biological applications, we evaluated our method on observable ordinal phenotypes for which a reflection in gene expression levels can be assumed. For three different model organisms we analysed developmental stages characterised by their morphology (*D. melanogaster*)¹, age (*Danio rerio*)² and number of C-lineage cells (*C. elegans*)⁴. For *C. elegans* also the sampling time points were used in the analysis.

Our screening procedure allowed to reveal ordinal structures within the gene expression profiles of all three model organisms. The hypothesised time relation or its inverse is always included in the set of best-performing cascades. In three out of those four datasets the hypothesised relation dominates with a performance gap towards all other cascades. This strongly indicates a reflection of these orderings in the profiles.

For the *Danio rerio* dataset (d_5) two cascades rank first before a performance gap. There is a swap observed between the two youngest embryo phenotypes. This might be caused either by a data intrinsic reason that those classes are not distinct enough, as staging by days post fertilization has been shown to exhibit high variation in growth rate²⁸, or by the technical aspect of the lower number of constraints for later classes.

In contrast to the developmental processes, of which the order of stages is tightly regulated by a genetic program^{5,29}, ageing is influenced by multiple factors³⁰. Nevertheless we resulted in comparable results for the dataset that measures age related gene expression changes. On the human muscle adaptation dataset 91.7% of all candidate orders were rejected. Among the remaining two candidates the expected order and only one false positive can be found. For non-ordinal phenotypes, as given by the collection of distinct cancer cell lines, no candidate cascades were observed. This indicates that ordinal relations are not a common phenomenon among multiple phenotypes.

Our method can, however, not only be used to confirm proposed hypotheses but also to explore the feature space for potential ordinal structures. This might become relevant if the relation is not easily accessible due to sampling. In surgery, for example, histologically distinguishable tissue regions can be defined in the same biopsy or in single-cell experiments various cell types are extracted from one sample. Within these feature spaces our procedure allows for screening of total ordinal cascades and additionally of ordinal sub-cascades embedded in a larger set of non-ordinal classes. It can hence be used to screen for intrinsic molecular ordinal structures and hypothesis relational axes, which might not be detected in a standard multi-class analysis.

Methods

We will use following notation throughout the description of the methodology behind the algorithm. An object is represented by a vector of real-valued measurements $\mathbf{x} = (x^{(1)}, \dots, x^{(n)})^T \in \mathbb{R}^n$. Each object is assumed to be categorisable in one out of $|\mathcal{Y}| \geq 2$ predefined classes $y \in \mathcal{Y}$, where $\mathcal{Y} = \{y_i\}_{i=1}^{|\mathcal{Y}|}$ denotes the space of all class labels. The general classification task will be to identify a function mapping c , a classifier, that allows the accurate prediction of the class labels of new unseen objects $c: \mathbb{R}^n \rightarrow \mathcal{Y} = \{y_1, \dots, y_{|\mathcal{Y}|}\}$.

As quality measures, we utilise the conditional prediction rates of c . These estimate the probability of classifier c to predict the class label y_j for samples of class y_i based on a set of test samples \mathcal{X}_i . In its basic version, a conditional prediction rate can be calculated as

$$p_c(y_j|\mathcal{X}_i) = \frac{1}{|\mathcal{X}_i|} \sum_{x \in \mathcal{X}_i} \mathbb{I}_{\{c(x)=y_j\}}$$

where $\mathbb{I}_{\{ \cdot \}}$ denotes the indicator function. Other (re-) sampling strategies might be used for determining conditional prediction rates. However, they will not alter the theoretical characteristics discussed in this work. We distinguish between three types of conditional prediction rates: 1. *sensitivities* if $y_i = y_j$ and $y_i, y_j \in \mathcal{Y}$, 2. *confusions* if $y_i \neq y_j$ and $y_i, y_j \in \mathcal{Y}$, 3. *external rates* if $y_i \notin \mathcal{Y}$ and $y_j \in \mathcal{Y}$.

While (class-wise) sensitivities and confusions build the standard quality measures of a confusion matrix³¹, the external rates describe the classifiers behaviour on foreign classes. They will especially be of interest when dealing with different label spaces.

In the basic multi-class classification scenario, a classifier is typically adapted in a data-driven training procedure based on a set of training samples $\mathcal{S}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{S}_{tr}|}$. The basic assumption of this scenario are pairwise distinct classes $y_1 \neq \dots \neq y_{|\mathcal{Y}|}$, which can be separated in the chosen feature space. In the ordinal classification scenario it is additionally assumed that the labels in \mathcal{Y} are totally ordered $y_{(1)} \prec \dots \prec y_{(|\mathcal{Y}|)}$. In this context, the symbol $y_{(i)}$ denotes the i th class of the order. We utilise the symbol \prec to indicate that the order relationship is only known for the label space. It is unclear, if this relationship is reflected by the chosen measurements. Nevertheless, ordinal classifiers rely on this assumption. The order of the classes is utilised for guiding the construction of the decision regions. It is provided as additional information to the training algorithm.

Ordinal classifier cascades. In the following, we will discuss ordinal classifier cascades of type

$$h_{i,j}: \mathbb{R}^n \rightarrow \{y_{(i)}, y_{(i+1)}, \dots, y_{(j+1)}\}. \quad (1)$$

The cascade $h_{i,j}$ can be seen as a late-aggregation multi-classifier system³², where indices $1 \leq i \leq j < |\mathcal{Y}|$ indicate the base classifiers of the corresponding ensemble $\mathcal{E} = \{c_{(i)}, \dots, c_{(j)}\}$. The members of the ensemble are designed for separating two neighboured classes $c_{(i)}: \mathbb{R}^n \rightarrow \{y_{(i)}, y_{(i+1)}\}$.

An ordinal cascade will be called a *full cascade* if it is designed for predicting all labels of the label space \mathcal{Y} . Full cascades will be denoted by $h = h_{1,|\mathcal{Y}|-1}$. Other ordinal cascades will be called *partial cascades*.

The fusion strategy of an ordinal cascade can be interpreted as a sequence of logical conjunctions of its base classifiers

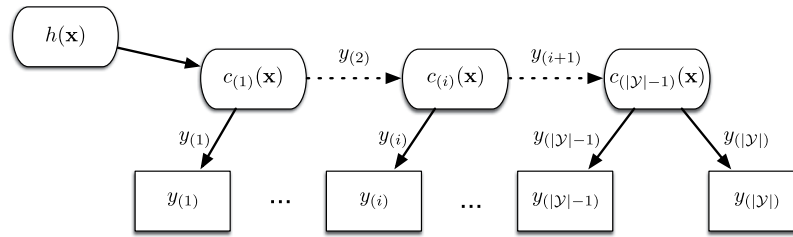


Figure 2. Structure of a full ordinal classifier cascade h . The cascade consists of a fixed order of base classifiers $c_{(i)}, 1 \leq i < |\mathcal{Y}|$. A sample \mathbf{x} is passed sequentially from one base classifier to another. If a base classifier $c_{(i)}(\mathbf{x})$ predicts class label $y_{(i)}$, the procedure stops and the ensemble h predicts $y_{(i)}$. Otherwise the ensemble h predicts $y_{(|\mathcal{Y}|)}$.

$$h_{i,j}(\mathbf{x}) = \begin{cases} y_{(i)} & \text{if } c_{(i)} = y_{(i)} \\ y_{(k)} & \text{if } \left(\bigwedge_{i \leq l < k \leq j} \left(c_{(l)}(\mathbf{x}) = y_{(l+1)} \right) \right) \wedge c_{(k)}(\mathbf{x}) = y_{(k)}. \\ y_{(j+1)} & \text{else } \bigwedge_{l=i}^j \left(c_{(l)}(\mathbf{x}) = y_{(l+1)} \right) \end{cases} \quad (2)$$

A scheme of this architecture can be found in Fig. 2. For classifying a sample \mathbf{x} , the ensemble members $c_{(k)}(\mathbf{x})$ are evaluated sequentially according to the assumed order of classes. If a base classifier $c_{(k)}(\mathbf{x})$ predicts its first class label $y_{(k)}$, the procedure stops and $h_{i,j}(\mathbf{x}) = y_{(k)}$. If it predicts class label $y_{(k+1)}$ the sample is passed to the subsequent base classifier $c_{(k+1)}$. This fusion scheme implies following three characteristics on $h_{i,j}$:

1. Each class $y_{(k)}, i < k < j + 1$ can be predicted by two base classifiers.
2. The lowest class $y_{(i)}$ can only be predicted by the first classifier $c_{(i)}(\mathbf{x})$. The highest classes $y_{(j+1)}$ can only be predicted by the last classifier $c_{(j)}(\mathbf{x})$.
3. A sample \mathbf{x} will only be passed to a base classifier $c_{(k)}(\mathbf{x}), 1 < k \leq j$, if all its predecessors $c_{(l)}(\mathbf{x}), l < k$, decide for their second class $y_{(l+1)}$.

Training algorithms for ordinal classifier cascades mainly focus on the training of the base classifiers. In the following, we utilise a pairwise inductive training, in which the training set $\mathcal{S}_{(k)}$ of a base classifier $c_{(k)}$ consists of the samples of classes $y_{(k)}$ and $y_{(k+1)}$

$$\mathcal{S}_{(k)} = \{(\mathbf{x}, y) | (\mathbf{x}, y) \in \mathcal{S}_{tr}, y \in \{y_{(k)}, y_{(k+1)}\}\}. \quad (3)$$

In a previous study, this type of training showed to induce the highest susceptibility to incorrect assumptions on the class order²³.

Upper bounds on class-wise sensitivities. The structural properties of the ordinal classifier cascade allow for the construction of upper limits on their empirical class-wise sensitivities. These bounds are based on the training of the cascade’s base classifiers and postulated in Theorem 1. Although this theorem is formulated for full cascades, the corresponding bounds can directly be applied for partial cascades.

Theorem 1 Let h denote an ordinal classifier cascade $h: \mathbb{R}^n \rightarrow \mathcal{Y} = \{y_{(1)}, \dots, y_{(|\mathcal{Y}|)}\}$ with base classifiers $\mathcal{E} = \{c_{(1)}, \dots, c_{(|\mathcal{Y}|-1)}\}$. Let furthermore $\mathcal{X}_{(i)}$ be a non-empty set of samples of class $y_{(i)}$. Then the sensitivity of h for $y_{(i)}$ is limited by

$$P_h(y_{(i)} | \mathcal{X}_{(i)}) \leq P_{c_{(i)}}(y_{(i)} | \mathcal{X}_{(i)}) \quad (4)$$

$$P_h(y_{(i)} | \mathcal{X}_{(i)}) \leq \min_{k < i} P_{c_{(k)}}(y_{(k+1)} | \mathcal{X}_{(i)}). \quad (5)$$

Proof. The theorem is a direct consequence of Lemmata 1 and 2 (see Supplementary).

Theorem 1 states that the sensitivities of an ordinal classifier cascade h can be upper bounded by several conditional prediction rates of its base classifiers. For class $y_{(i)}$, the sensitivity of the cascade is limited by the corresponding sensitivity of its i th base classifier $c_{(i)}$ (Eq. 4). It is also bounded by the predictions of all previous base classifiers $c_{(k)}, k < i$ (Eq. 5). A sample of class $y_{(i)}$ will not be classified correctly, if it is classified as $y_{(k)}$ by $c_{(k)}$. The sensitivity of the cascade for class $y_{(i)}$ is therefore also limited by the conditional prediction rate of $c_{(k)}$ for predicting class label $y_{(k+1)}$ for samples of class $y_{(i)}$. A detailed theoretical proof can be found in the Supplementary.

Detection of ordinal class structures. Ordinal classifier cascades can be used for detecting wrong assumptions about the ordinality of the real class structures. Due to their susceptibility, these classifiers will fail when the real feature structures reflect a different class order or no class order at all. In a screening process,

Number of Classes	Number of Base Classifiers	
	Precalculation	De Novo Training
$ \mathcal{Y} $	$(\mathcal{Y} - 1) \mathcal{Y} $	$(\mathcal{Y} - 1) \mathcal{Y} !$
2	2	2
3	6	12
4	12	72
5	20	480
10	90	$>3 \cdot 10^7$
20	380	$>4 \cdot 10^{19}$
30	870	$>7 \cdot 10^{33}$

Table 2. A comparison between the precalculation strategy and a de novo calculation is given. Number of base classifiers required for an exhaustive training of all $|\mathcal{Y}|!$ ordinal classifier cascades is shown. The table gives the number of base classifiers that are required to be trained and evaluated within the screening process based on a single training-test training.

ordinal classifier cascades can be used for revealing unknown ordinal class structures. We have proposed the minimal class-wise sensitivity p^* of an ordinal classifier cascade as a measure of the correctness of the assumed class order $p^* = \min_{1 \leq i \leq |\mathcal{Y}|} p_h(y_{(i)} | \mathcal{X}_{(i)})$. A sensitivity threshold $t \leq p^*$ is used for determining whether an ordinal class structure can be assumed or not. The criterion can be evaluated for each order of the classes in \mathcal{Y} .

The findings of Theorem 1 allow an alternative evaluation of this criterion. As a direct consequence of Theorem 1, the value of p^* can again be upper bounded by conditional prediction rates of the base classifiers

$$\forall i: p^* \leq p_h(y_{(i)} | \mathcal{X}_{(i)}) \leq p_{c_{(i)}}(y_{(i)} | \mathcal{X}_{(i)}) \quad (6)$$

$$\forall i: p^* \leq p_h(y_{(i)} | \mathcal{X}_{(i)}) \leq \min_{k < i} p_{c_{(k)}}(y_{(k+1)} | \mathcal{X}_{(i)}). \quad (7)$$

Ordinal classifier cascades that are based on wrong assumptions about the ordinality of the classes can therefore be sorted out by the training of the corresponding base classifiers. A graphical illustration describing this sorting out based on a four class example and dependent on Eqs 6 and 7 can be found in Supplementary Fig. S4.

Coupled to a pairwise inductive training of the base classifiers (Eq. 3) the bounds of Theorem 1 can reduce complexity of screens for ordinal structures. As the training of a base classifier $c_{(i)}$ is only based on the samples of classes $y_{(i)}$ and $y_{(i+1)}$, it is no longer dependent on the position of the base classifiers within the cascade h . Cascades trained on different orders of \mathcal{Y} will therefore consist of common building blocks. The exhaustive training of all $|\mathcal{Y}|!$ cascades, each consisting of $|\mathcal{Y}| - 1$ base classifiers, can therefore be accelerated by precalculating and evaluating all possible $(|\mathcal{Y}| - 1)|\mathcal{Y}|$ base classifiers $c_{i,j}: \mathcal{R}^n \rightarrow \{y_i, y_j\}$. Note that symbols y_i, y_j and $c_{i,j}$ no longer rely on an assumed class order.

In any case, the complexity of the exhaustive evaluation is mainly determined by the training and evaluation complexity of the base classifiers. A comparison of the precalculation scheme and a de novo calculation of all cascades in dependency on the numbers of classes $|\mathcal{Y}|$ can be found in Table 2. For the presented numbers we assume an evaluation via a single training-test split. For three classes, the de novo strategy already requires twice the number of base classifier trainings and evaluations than the precalculation strategy. For ten classes, the de novo strategy trains more than $3 \cdot 10^7$ base classifiers while the precalculation scheme only demands 90 base classifiers. The number of base classifier trainings and evaluations might be increased by a constant factor if resampling strategies are applied.

The following quality measures are needed for the application of Theorem 1:

$$FC_{i,j} = p_{c_{i,j}}(y_i | \mathcal{X}_i) \quad \text{and} \quad SC_{i,j}(r) = p_{c_{i,j}}(y_j | \mathcal{X}_r). \quad (8)$$

Here, $FC_{i,j}$ denotes the class-wise sensitivity of $c_{i,j}$ for predicting its first class label y_i . The term $SC_{i,j}(r)$ denotes the conditional prediction rates of $c_{i,j}$ for samples of class y_r that are classified as y_j . Both quantities can be precalculated and memorised for all binary base classifiers (Table 3).

The CASCADES algorithm. We propose the recursive enumeration scheme CASCADES($\mathcal{Y}, \mathcal{C}, y_i, t$) for the exhaustive training of all orders of \mathcal{Y} (Fig. 3). It can be seen as a filter routine that rejects ordinal cascades that will not achieve a minimal class-wise sensitivity $t \leq p^*$ according to the bounds of Theorem 1. The remaining cascades will be returned as a set of candidates \mathcal{C} .

The CASCADES algorithm is based on the evaluation of an extended confusion table as shown in Table 3. It will replace the training procedure of each base classifier (training-test split or resampling strategy) by looking up $|\mathcal{Y}| - k$ conditional prediction rates, where k is the base classifiers position within the cascade. As there exist $\binom{|\mathcal{Y}|}{k+1}$ possibilities of constructing subcascades (prefixes) of k base classifiers, at most $f(|\mathcal{Y}|) = \sum_{k=2}^{|\mathcal{Y}|} k \binom{|\mathcal{Y}|}{k+1}$ comparisons are required in a worst case scenario. Nevertheless, this number will rapidly break down by utilising early stopping criteria.

$c_{i,j}(x)$		First Class $FC_{i,j}$	Second Class $SC_{i,j}(r)$				
class i	class j	y_i	y_1	...	y_r	...	$y_{ \mathcal{Y} }$
\vdots	\vdots	\vdots	\vdots		\vdots		\vdots
y_i	y_j	$P_{c_{i,j}}(y_i \mathcal{X}_i)$	\vdots		$P_{c_{i,j}}(y_j \mathcal{X}_r)$		\vdots
\vdots	\vdots	\vdots	\vdots		\vdots		\vdots

Table 3. Training table of binary classifiers $c_{i,j}$ in a multi-class scenario ($|\mathcal{Y}| > 2$). Each classifier $c_{i,j}$ is trained on the samples of classes $y_i, y_j \in \mathcal{Y}$ and evaluated on the samples of all classes in \mathcal{Y} . The sensitivity of $c_{i,j}$ on detecting samples of y_i is denoted by $FC_{i,j}$. The conditional prediction rates of $c_{i,j}$ of predicting class y_j for samples of class y_r is given by $SC_{i,j}(r)$.

```

CASCADES( $\mathcal{Y}, \mathcal{C}, y_i, t$ )
1: if  $|\mathcal{Y}| = 0$  then
2:   return  $\mathcal{C}$ 
3: else
4:    $\mathcal{C}_s := \emptyset$ 
5:   if  $y_i = \emptyset$  then
6:     for  $y_j \in \mathcal{Y}$  do
7:        $\mathcal{C}_c := \text{CASCADES}(\mathcal{Y} \setminus \{y_j\}, \{y_j\}, y_j, t)$ 
8:        $\mathcal{C}_s := \mathcal{C}_s \cup \mathcal{C}_c$ 
9:     end for
10:    return  $\mathcal{C}_s$ 
11:   else
12:     for  $y_j \in \mathcal{Y}$  do
13:       if  $FC_{i,j} \geq t$  and  $\forall y_r \in \mathcal{Y} : SC_{i,j}(r) \geq t$  then
14:          $\mathcal{C}_r := \{(y_{(1)}, \dots, y_{(k)}, y_j) \mid (y_{(1)}, \dots, y_{(k)}) \in \mathcal{C}\}$ 
15:          $\mathcal{C}_c := \text{CASCADES}(\mathcal{Y} \setminus \{y_j\}, \mathcal{C}_r, y_j, t)$ 
16:       else
17:          $\mathcal{C}_c := \emptyset$ 
18:       end if
19:        $\mathcal{C}_s := \mathcal{C}_s \cup \mathcal{C}_c$ 
20:     end for
21:   return  $\mathcal{C}_s$ 
22: end if
23: end if
    
```

Figure 3. CASCADES algorithm for an exhaustive screening of all orders of class labels $y \in \mathcal{Y}$. The function parameters are the set of class labels \mathcal{Y} , a set of candidate cascades \mathcal{C} , the threshold t and y_i which specifies the class label that was selected in the previous recursion.

A single cascade is represented as an ordered vector of class labels $(y_{(1)}, \dots, y_{(|\mathcal{Y}|)}) \in \mathcal{C}$. Each candidate cascade is constructed sequentially. It is extended by a new class label in each recursive call of the algorithm. The construction stops, if the performance measures of the current base classifier falls under the chosen threshold t . In this case the candidate cascade is rejected.

The sequential extension of a partial cascade $h_{1,k-1}$ improves the runtime of the exhaustive search. If the k th (candidate) base classifier does not fulfil the minimal criteria, all full ordinal cascades that utilise $h_{1,k}$ as prefix can be withdrawn. This corresponds to $(|\mathcal{Y}| - k - 1)!$ full ordinal cascades. A single early stopping will reduce the number of lookups by at least $f(|\mathcal{Y}| - k)$.

The algorithm is initialised with the full set of labels \mathcal{Y} , an empty set of candidate cascades $\mathcal{C} = \emptyset$, the chosen threshold t and $y_i = \emptyset$. The parameter y_i will later on indicate the class label selected in the previous recursion. In each recursion, the class labels $y_j \in \mathcal{Y}$ are tested as possible extensions of the candidate cascades in \mathcal{C} . If $FC_{i,j} \geq t$ and $\forall y_r \in \mathcal{Y} : SC_{i,j}(r) \geq t$ the current base classifier $c_{i,j}$ fulfils the bounds on p^* . In this case, class label y_j can be added to the current candidate cascade and can be removed from the set of remaining labels \mathcal{Y} . The next base classifier will be chosen by a recursive call $\text{CASCADES}(\mathcal{Y} \setminus \{y_j\}, \mathcal{C}_r, y_j, t)$. If the current base classifier does not fulfil the minimal criteria, the corresponding (partial) candidate cascades are erased and an empty set \emptyset is returned. All suitable candidate cascades are collected at the end of the recursive call. Although the algorithm CASCADES rejects cascades with too low minimal class-wise sensitivities, the remaining candidates are not guaranteed to fulfil the minimal criterion $t \leq p^*$. Each of the final candidates must therefore be cross-checked by an evaluation of the full cascade. CASCADES can directly be applied for the evaluation of partial ordinal cascades. By replacing the initial set of class labels \mathcal{Y} by a subset $\mathcal{Y}' \subset \mathcal{Y}$, the algorithm will evaluate all orders of the class labels in \mathcal{Y}' .

Datasets. An overview of the characteristics of all datasets can be found in Supplementary Table S1. The datasets d_5-d_9 were collected from the gene expression omnibus repository³³ (GSE13371, GSE47881, GSE2180, GSE32474) and processed using the robust multi-array average (rma) normalisation as implemented in the affy package³⁴. For d_4 the processed data was downloaded.

Linear dataset (d_1). For a first series of experiments, the centroid of the i th class y_i is chosen as $\mathbf{m}_{y_i} = (i, i)^T$. In this way, the class centroids lie on a line.

Curved dataset (d_2). For the second dataset, the class centroids were chosen depending on their predecessors. $\mathbf{m}_{y_i} = (m_{y_i}^{(1)}, m_{y_i}^{(2)})^T = (m_{y_{i-1}}^{(1)} + u_i^{(1)}, m_{y_{i-1}}^{(2)} + u_i^{(2)})^T$, where $u_i^{(1)}, u_i^{(2)} \sim \mathcal{U}(0.5, 2)$. As a starting point $\mathbf{m}_{y_0} = (0, 0)^T$ was chosen. This dataset has a curved shape.

Non-ordinal dataset (d_3). The third artificial dataset is designed to be non-ordinal. The centers of the classes are arranged on a predefined two dimensional grid in the range $[1, 4]^2$ (Supplementary Table S2).

***Drosophila melanogaster* (d_4).** The drosophila dataset generated by Arbeitman *et al.*¹ consists of gene expression profiles of the fruit fly *Drosophila melanogaster*. These profiles consist of 4028 measurements and were collected at different points in time during the life cycle of the model organism. They can be categorised according to the developmental stages of *Drosophila melanogaster*: *embryo* < *larva* < *pupa* < *adult*. Overall the dataset contains profiles for 31 *embryos*, 10 *larvae*, 18 *pupae* and 8 *adults*. We use the natural order of the developmental stages as ordinal class labels for our experiment.

***Danio rerio* (d_5).** The dataset collected by Toyama *et al.*² consists of gene expression profiles of the pineal glands of zebrafish (*Danio rerio*). The expression profiles were collected at five different time points: *embryo*₁ < *embryo*₂ < *embryo*₃ < *adult*₁ < *adult*₂, where *embryo*₁–*embryo*₃ were collected 3, 5 and 10 days after birth and *adult*₁ and *adult*₂ were collected at an age of 3 months and 1–2 years. The dataset comprises 12–15 samples for each class. The age of the samples will be used as class order.

Human muscle adaptation (d_6). Philips *et al.*³ have compared the transcriptome of human muscle cells before and after 20 weeks of supervised resistance-exercise training (RET). The corresponding dataset consists of paired gene expression profiles. For our experiments, the data was categorised into four classes according to the age (in years) of the participants: *age*₁ < *age*₂ < *age*₃ < *age*₄. The class labels denote age intervals of [20; 40) years (18 samples), [40; 60) years (38 samples), [60; 70) years (16 samples) and [70; 80) years (16 samples). In order to avoid overoptimistic results, we ensured that the profiles of a subject are not used for training and testing the classifier simultaneously.

***Caenorhabditis elegans* (d_7 and d_8).** Baugh *et al.*⁴ analysed the influence of the homeodomain protein PAL-1 of the C-lineage-specific gene regulatory network in the model organism *C. elegans*. They gathered gene expression data of samples of wild-type embryos and mutant embryos with additional C blastomeres, as well as on samples of mutants without any C blastomeres. For our experiments we used data of the C-cell-free organisms, taken at 10 points in time after the 4-cell-stage of the embryo. We labelled these samples in two different ways. In the first experiment (d_7) the samples were labelled according to the developmental stages proposed in the original publication: *stage*₁ < *stage*₂ < *stage*₃ < *stage*₄ < *stage*₅, where 0 and 23 minutes samples were merged in the *stage*₁ class, 41 and 53 minutes samples in *stage*₂, and samples taken at 66, 83 and 101 minutes after the 4-cell stage in the class *stage*₃. *stage*₄ only consists of samples taken at 122 minutes, and *stage*₅ contains time points 143 and 186 minutes. In the second experiment (d_8), the points in time were analysed solely: $t_1 < t_2 < t_3 < t_4 < t_5 < t_6 < t_7 < t_8 < t_9 < t_{10}$.

Various cancer cell lines (d_9). Pfister *et al.*²⁷ collected gene expression profiles from cell lines that derived from 9 different cancer tissue types (*breast* (15 samples), *central nervous system* (18 samples), *colon* (21 samples), *leukemia* (18 samples), *melanoma* (26 samples), *non-small cell lung* (26 samples), *ovarian* (21 samples), *prostate* (6 samples), *renal* (23 samples)). In contrast to d_1 – d_8 , the classes of this dataset are not assumed to fulfil an ordinal relationship as each group originates from a different tissue type: $line_1 \neq line_2 \neq line_3 \neq line_4 \neq line_5 \neq line_6 \neq line_7 \neq line_8 \neq line_9$.

Data Availability

The drosophila dataset is available from <http://flygenome.yale.edu/Lifecycle/>. The other datasets are available from the GEO repository <https://www.ncbi.nlm.nih.gov/gds>: GSE13371, GSE47881, GSE2180, GSE32474.

References

- Arbeitman, M. N. *et al.* Gene expression during the life cycle of *Drosophila melanogaster*. *Sci.* **297**, 2270–2275, <https://doi.org/10.1126/science.1072152> (2002).
- Toyama, R. *et al.* Transcriptome analysis of the zebrafish pineal gland. *Dev. Dyn.* **238**, 1813–1826, <https://doi.org/10.1002/dvdy.21988> (2009).
- Phillips, B. E. *et al.* Molecular networks of human muscle adaptation to exercise and age. *PLOS Genet.* **9**, 1–15, <https://doi.org/10.1371/journal.pgen.1003389> (2013).
- Baugh, L. R. *et al.* The homeodomain protein PAL-1 specifies a lineage-specific regulatory network in the *C. elegans* embryo. *Dev.* **132**, 1843–1854, <https://doi.org/10.1242/dev.01782> (2005).
- Long, C., Li, W., Liang, P., Liu, S. & Zuo, Y. Transcriptome comparisons of multi-species identify differential genome activation of mammals embryogenesis. *IEEE Access* **99**, 1–1, <https://doi.org/10.1109/ACCESS.2018.2889809> (2018).
- Kearney, P. E., Hayward, R. B. & Meijer, H. Evolutionary trees and ordinal assertions. *Algorithmica* **25**, 196–221, <https://doi.org/10.1007/PL00008274> (1999).
- Guénoche, A. Ordinal properties of tree distances. *Discret. Math.* **192**, 103–117, [https://doi.org/10.1016/S0012-365X\(98\)00068-5](https://doi.org/10.1016/S0012-365X(98)00068-5) (1998).
- Kannan, S. & Warnow, T. Tree Reconstruction from Partial Orders. *SIAM J. on Comput.* **24**, 511–519, <https://doi.org/10.1137/S0097539793252195> (1995).
- Buchholz, M. *et al.* Transcriptome analysis of microdissected pancreatic intraepithelial neoplastic lesions. *Oncogene* **24**, 6626–6636, <https://doi.org/10.1038/sj.onc.1208804> (2005).
- Feldmann, U. & Steudel, I. Methods of ordinal classification applied to medical scoring systems. *Stat. Medicine* **19**, 575–586, [doi:10.1002/\(SICI\)1097-0258\(20000229\)19:4<575::AID-SIM357>3.0.CO;2-A](https://doi.org/10.1002/(SICI)1097-0258(20000229)19:4<575::AID-SIM357>3.0.CO;2-A) (2000).
- Bender, R. & Grouven, U. Ordinal logistic regression in medical research. *J. Royal Coll. Physicians Lond.* **31**, 546–551 (1997).
- Cardoso, J. S., da Costa, J. F. P. & Cardoso, M. J. Modelling ordinal relations with svms: An application to objective aesthetic evaluation of breast cancer conservative treatment. *Neural Networks* **18**, 808–817, <https://doi.org/10.1016/j.neunet.2005.06.023> (2005).
- Bishop, C. *Pattern Recognition and Machine Learning* (Springer, New York, 2006).

14. Hastie, T., Tibshirani, R. & Friedman, J. H. *The Elements of Statistical Learning* (Springer, New York, 2001).
15. Webb, A. R. *Statistical Pattern Recognition*, 2nd edn (John Wiley & Sons Ltd., Chichester, 2002).
16. Fix, E. & Hodges, J. L. Discriminatory analysis: Nonparametric discrimination: Consistency properties. Tech. Rep. Project 21-49-004, Report Number 4, USAF School of Aviation Medicine, Randolph Field, Texas (1951).
17. Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. *Classification and Regression Trees*. The Wadsworth statistics/probability series (Chapman and Hall/CRC, Boca Raton, 1984).
18. Vapnik, V. N. *Statistical Learning Theory* (John Wiley & Sons Ltd., New York, 1998).
19. Frank, E. & Hall, M. A simple approach to ordinal classification. In Raedt, L. D. & Flach, P. (eds) *Machine Learning: ECML 2001 - 12th European Conference on Machine Learning. Proceedings*, vol. 2167, 145–156, <https://doi.org/10.1007/3-540-44795-413> (Springer, Berlin, 2001).
20. Rivest, R. L. Learning decision lists. *Mach. Learn.* **2**, 229–246, <https://doi.org/10.1007/BF00058680> (1987).
21. Kestler, H. A., Lausser, L., Lindner, W. & Palm, G. On the fusion of threshold classifiers for categorization and dimensionality reduction. *Comput. Stat.* **26**, 321–340, <https://doi.org/10.1007/s00180-011-0243-7> (2011).
22. Meyer, L. H. *et al.* Early Relapse in ALL Is Identified by Time to Leukemia in NOD/SCID Mice and Is Characterized by a Gene Signature Involving Survival Pathways. *Cancer Cell* **19**, 206–217, <https://doi.org/10.1016/j.ccr.2010.11.014> (2011).
23. Lattke, R., Lausser, L., Müssel, C. & Kestler, H. A. Detecting ordinal class structures. In Schwenker, F., Roli, F. & Kittler, J. (eds) *Multiple Classifier Systems (MCS 2015)*, vol. 9132, 100–111, <https://doi.org/10.1007/978-3-319-20248-89>: (Springer International Publishing, Cham, 2015).
24. Chang, C.-C. & Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intell. Syst. Technol.* **2**, 27:1–27:27, <https://doi.org/10.1145/1961189.1961199> (2011).
25. Japkowicz, N. & Shah, M. *Evaluating Learning Algorithms: A Classification Perspective* (Cambridge University Press, New York, 2011).
26. Müssel, C., Lausser, L., Maucher, M. & Kestler, H. A. Multi-objective parameter selection for classifiers. *J. Stat. Softw.* **46**, 1–27, <https://doi.org/10.1007/1179085329> (2012).
27. Pfister, T. D. *et al.* Topoisomerase I levels in the NCI-60 cancer cell line panel determined by validated ELISA and microarray analysis and correlation with indenoisoquinoline sensitivity. *Mol. Cancer Ther.* **8**, 1878–1884, <https://doi.org/10.1158/1535-7163.MCT-09-0016> (2009).
28. Parichy, D. M., Elizondo, M. R., Mills, M. G., Gordon, T. N. & Engeszer, R. E. Normal table of postembryonic zebrafish development: Staging by externally visible anatomy of the living fish. *Dev. Dyn.* **238**, 2975–3015, <https://doi.org/10.1002/dvdy.22113> (2009).
29. Dong, P. & Liu, Z. Shaping development by stochasticity and dynamics in gene regulation. *Open Biol.* **7**, 170030, <https://doi.org/10.1098/rsob.170030> (2017).
30. Rodriguez-Rodero, S. *et al.* Aging Genetics and Aging. *Aging Dis.* **2**, 186–195, http://www.aginganddisease.org/EN/abstract/article_147277.shtml (2011).
31. Zuo, Y.-C. *et al.* Predicting peroxidase subcellular location by hybridizing different descriptors of Chou' pseudo amino acid patterns. *Anal. Biochem.* **458**, 14–19, <https://doi.org/10.1016/j.ab.2014.04.032> (2014).
32. Lorena, A. C., de Carvalho, A. C. P. L. F. & Gama, J. M. P. A review on the combination of binary classifiers in multiclass problems. *Artif. Intell. Rev.* **30**, 19–37, <https://doi.org/10.1007/s10462-009-9114-9> (2009).
33. Edgar, R., Domrachev, M. & Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* **30**, 207–210, <https://doi.org/10.1093/nar/30.1.207> (2002).
34. Gautier, L., Cope, L., Bolstad, B. M. & Irizarry, R. A. affy-analysis of Affymetrix GeneChip data at the probe level. *Bioinforma.* **20**, 307–315, <https://doi.org/10.1093/bioinformatics/btg405> (2004).

Acknowledgements

The research leading to these results has received funding from the German Research Foundation (DFG, SFB 1074 project Z1, and GRK 2254 HEIST), and the Federal Ministry of Education and Research (BMBF, Gerontosys II, Forschungskern SyStaR, id 0315894 A; e:Med, SYMBOL-HF, id 01ZX1407A; conFirm, id 01ZX1708C) all to H.A.K.

Author Contributions

L.L. designed the algorithm and conceived the experiments, L.M.S. and L.-R.S. performed data acquisition and conducted the experiments, L.L., L.M.S. and L.-R.S. analysed the results, L.L., R.S. and F.S. implemented the algorithm, L.L. and F.S. drafted the manuscript, H.A.K. supervised and guided the study. L.L., L.M.S., L.-R.S. and H.A.K. wrote the manuscript. All authors reviewed the manuscript.

Additional Information

Supplementary information accompanies this paper at <https://doi.org/10.1038/s41598-019-48150-z>.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019