Scalable embedding of multiple perspectives for indefinite life-science data analysis

Munch, Maximilian; Heilig, Simon; Vath, Philipp; Schleif, Frank Michael

Published in:
2021 IEEE Symposium Series on Computational Intelligence, SSCI 2021 - Proceedings

DOI:
10.1109/SSCI50451.2021.9659914

Scalable embedding of multiple perspectives for indefinite life-science data analysis

Maximilian Münch
Bernoulli Institute for Mathematics,
Computer Science and Artificial Intelligence
University of Groningen
Groningen, The Netherlands
m.a.munch@rug.nl

Simon Heilig, Philipp Väth, Frank-Michael Schleif
Department of Computer Science and
Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany
{simon.heilig, philipp.vaeth, frank-michael.schleif}@fhws.de

2021 IEEE Symposium Series on Computational Intelligence (SSCI) | 978-1-7281-9048-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/SSCI50451.2021.9659914

***Abstract*—Life science data analysis frequently encounters particular challenges that cannot be solved with classical techniques from data analytics or machine learning domains. The complex inherent structure of the data and especially the encoding in non-standard ways, e.g., as genome- or protein-sequences, graph structure or histograms, often limit the development of appropriate classification models. To address these limitations, the application of domain-specific expert similarity measures has gained a lot of attention in the past. However, the use of such expert measures suffers from two major drawbacks: (a) there is not one outstanding similarity measure that guarantees success in all application scenarios, and (b) such similarity functions often lead to indefinite data that cannot be processed by classical machine learning methods. In order to tackle both of these limitations, this paper presents a method to embed indefinite life science data with various similarity measures at the same time into a complex-valued vector space. We test our approach on various life science data sets and evaluate the performance against other competitive methods to show its efficiency.**

***Index Terms*—Indefinite learning, complex-valued embedding, life science data, multi-perspective embedding, multimodal data**

I. Introduction

The demand for robust and reliable models in life science data analysis, like bioinformatics, biochemistry, environmental research, medicine, and others, has never been greater: not only the pure amount of data but also the intrinsic complexity of the data is increasing steadily, such that classical techniques are not applicable. A common way to capture the complexity of the data is the application of so-called structured data formats like encodings as sequence data, graph structures, or image-based data. Naturally, a downside of these data representations arises from most machine learning algorithm's constraints to numerical fixed-length input vectors, which are not given by these structured formats.

A common strategy to transform such structured input data into a vectorial representation is given by embedding techniques from deep learning [1]–[3]. However, these techniques require immense quantities of input data and are unsupervised, so there is no guarantee that the essential structure of the input is preserved after the embedding. Due to the lack of sufficiently large data sets, proximity-based measures are a powerful alternative to handle non-vectorial input data [4], [5]. Such a proximity-based measure is usually grounded on the domain-specific knowledge of a domain expert and produces a proximity-score for each pair of data points from the input data.

Depending on the type of proximity function, this score characterises the degree of *similarity* or *dissimilarity* between each input data point. Considering, e.g., for a set of protein sequences, a proximity function can either measure the relatedness (in case of a similarity function) or the difference (in case of a dissimilarity function) between all sequences of the set. Nevertheless, the majority of machine learning methods require the respective proximity functions to satisfy strict mathematical properties to guarantee well-performing and robust models. However, most domain-specific (dis-)similarity measures typically used for life science applications are not fulfilling the mathematical requirements [5], and the representations are still very costly without approximations [4].

Another challenging task in life science data analysis is the high complexity inherent in the analysis task itself. In general, there is not one outstanding proximity measure that fits perfectly, regardless of the task and the data's structure. In fact, selecting an appropriate proximity measure remains a challenging task requiring a substantial amount of time and computational power [7]. To overcome this limitation, various methods combining multiple (dis-)similarity functions have recently been proposed [7]–[9]. The underlying idea of these methods is to consider the given problem from multiple perspectives to better handle the intrinsic complexity. In literature, this strategy is also referred to as *multi-view learning* - see e.g. [10] for an in-depth analysis of this domain.

In summary, both the high complexity of life science tasks and the compositional nature of the given data result in two main challenges in life science data analysis:

1) A translation of non-vectorial data into a vectorial representation with low computational costs.
2) The combination of multiple proximity functions to consider the given problem from multiple perspectives.

MM is supported by the Bavarian HighTech agenda and the Würzburg Center for Artificial Intelligence and Robotics (CAIRO). SH is supported by the ESF (WiT-HuB 4/2014-2020), project KI-trifft-KMU, StMBW-W-IX.4-6-190065.

978-1-7281-9048-8/21/$31.00 ©2021 IEEE

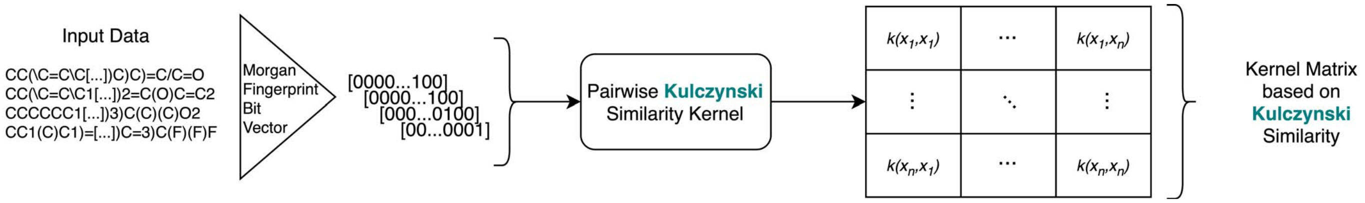
Input Data
CC(\C=C\C[...])C)C)=C/C=O
CC(\C=C\C1[...])2=C(O)C=C2
CCCCCC1[...])3)C(C)(C)O2
CC1(C)C1)=[...])C=3)C(F)(F)F
Morgan Fingerprint Bit Vector
[0000...100]
[0000...100]
[000...0100]
[00...0001]
Pairwise Kulczynski Similarity Kernel
$k(x_1,x_1)$
⋯
$k(x_1,x_n)$
⋮
⋱
⋮
$k(x_n,x_1)$
⋯
$k(x_n,x_n)$
Kernel Matrix based on Kulczynski Similarity

Fig. 1. Preprocessing workflow for creating the Tox-21 data sets. Chemicals represented as SMILE codes are translated to Morgan Fingerprints. The kernel is created by using an application related pairwise similarity measure on the Morgan Fingerprints, in this case so-called *Kulczynski*. Pairwise calculated similarities are stored in the proximity kernel matrix on the right [6].

For this purpose, this paper provides an extension of own previous research from [11] to integrate various (dis-) similarity functions at low costs with moderate approximations. At first, we embed the data into several (potentially complex-valued) vector spaces and combine these spaces. During the the model's training, we apply a technique called *relevance learning* which captures each perspective's importance and the impact of the respective proximity function. At the end of the training, the obtained model provides not only information about the importance of the proximity function, but we also obtain information about the most important reference points in the data set by means of a prototype-based classifier model. All elements of our approach are described in more detail in the following. Subsequently, the effectiveness of our method is evaluated on a variety of benchmark data. We conclude with a detailed discussion of the results and an outlook on further research on this topic.

II. Learning classification models from structured data

Learning a classification model from structured input data is a challenging task. Recently, for structured data, in particular for sequence and graph data, deep learning-based techniques such as ProtVec [12], Node2Vec [13], or Graph2Vec [3] became highly competitive for learning a vectorial representation of non-vectorial input data.

On the one hand, these models perform very well and solve outstanding challenges [14], but on the other hand, they are extraordinary computationally expensive and require a lot of training data. Especially in life sciences, data collection can be an expensive task leading to limited input data. Additionally, particularly in life sciences, interpretability is a crucial requirement for classification models. Moreover, life science data are traditionally characterized by a high degree of complexity and heterogeneity. For these reasons, deep learning methods are no longer considered in this paper. Instead, we will focus on multi-modal similarity-based expert functions to describe multiple perspectives of the particular data. For this purpose, this section comprises a brief overview of similarity-based learning, learning with multiple similarity functions, and how to deal with indefinite data.

A. Mathematical Background and Basic Notation

Consider a collection of N objects $\{\mathbf{x}_i\}$, $i = 1, 2, \ldots, N$ in some input space $\mathcal{X}$, where $\mathbf{x}_i$ need not to be in a vectorial form. A proper Mercer kernel acting on pairs of $\mathcal{X}$ can be constructed starting with a given similarity function or inner product on $\mathcal{X}$. For example, if $\mathcal{X}$ is a finite-dimensional vector space, a classical similarity function is the Euclidean inner product (corresponding to the Euclidean distance). Additionally, $\phi : \mathcal{X} \mapsto \mathcal{H}$ is a mapping from $\mathcal{X}$ to a Hilbert space $\mathcal{H}$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Usually, the transformation ϕ is a non-linear mapping to a high-dimensional space $\mathcal{H}$ and may not be given in an explicit form (meaning without an explicit calculation). Instead, a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is given, which encodes the inner product in $\mathcal{H}$. The kernel k is a positive (semi-)definite (psd) function such that $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$, for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. The matrix $\mathbf{K}_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ is an $N \times N$ kernel (Gram) matrix derived from the training data. For more general similarity measures, subsequently, we also use $\mathbf{S}$ to describe a similarity matrix. This procedure is motivated by the non-linear transformation of input data into higher dimensional $\mathcal{H}$, allowing linear techniques in $\mathcal{H}$. Kernelized methods process the embedded data points in a feature space utilizing only the inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ [15], without the need to explicitly calculate ϕ. This technique leads to great success and is referred to as the *kernel trick*. In general, the kernel function can be very generic. Most prominent are the linear kernel with $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ where $\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ is the Euclidean inner product and ϕ is the identity mapping, or the RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\sigma^2}\right)$, with $\sigma > 0$ as a free scale parameter. In any case, most kernel methods require the kernel function $k(\mathbf{x}, \mathbf{x}')$ to be positive semi-definite. A matrix $\mathbf{K}$ is positive semi-definite if $x^T\mathbf{K}x \geq 0$ for all $x \in R^n$, respectively, if all eigenvalues of $\mathbf{K}$ are greater or equal to zero. Note that domain-specific measures derived from expert knowledge do not always satisfy this requirement, as the applied similarity measure may not imply a metric dissimilarity and hence does not lead to a Mercer kernel. As an example, the creation of such an $N \times N$ kernel matrix is illustrated in Fig. 1, following the preprocessing pipeline from [6]. The collection of N data points is given in the input space $\mathcal{X}$ as SMILE codes in a non-vectorial form. At first, the input data is translated into bit-vectors, so-called Morgan fingerprints. Next, a similarity measure (in this case Kulczynski) calculates for each pair of bit-vectors a similarity score and stores this value for each pair in a similarity matrix $\mathbf{S}$ of size $N \times N$.

Consequently, such non-positive semi-definite (non-psd) similarity measures cause *indefinite* kernels, resulting in complications with methods developed for Mercer kernels. Nevertheless, indefinite expert measures are enjoying considerable popularity as they constantly outperform their metric counterparts [16]. Due to their excellent results, there is a wide variety of expert measures today [5].

B. Multiple Kernel Learning

Considering highly complex problems from different perspectives, particularly in life science disciplines, a given problem and its associated data are frequently of such complexity that a multi-dimensional consideration of the problem, i.e. from various perspectives, is the only way to solve it. A multi-dimensional treatment of objects in similarity-based learning is usually done by *Multiple Kernel Learning* (MKL). Generally, MKL aims to derive one strong kernel as a combination of multiple weak base kernels as an input to an arbitrary kernel method. For this reason, MKL is frequently used in information fusion where each kernel was derived by a different similarity measure or came from different input sources (e.g., for analyzing text, video, and audio data simultaneously) [17]. Nowadays, there is a wide variety of techniques for combining multiple kernel functions. A highly convenient framework is provided by [7].

Nevertheless, multiple kernel learning is still highly limited by strong mathematical constraints of kernel methods [15], as indicated above. Hence, the immediate usage of arbitrary distance or similarity measures in MKL leads to a guarantee loss in the optimization procedure as the problem is not strictly convex anymore. For example, the famous support vector machine (SVM) can only be used to a limited extent since the convexity of the optimization can no longer be guaranteed [18]. In case of a psd input matrix, the underlying convex optimization can be solved by standard numerical solvers, approaching the global optimum [19]. However, if the input matrix is indefinite, there might be no global minimum, and only a local optimum can be found, or the solver does not converge at all [18]. Therefore, employing a non-psd measure in SVM is a heuristic approach without any guarantees prohibiting in practical applications.

C. Learning with Indefinite Similarity Functions

As a consequence, several correction and adjustment procedures were designed to continue working with indefinite similarity measures. Following the taxonomy of [5], there are two main directions that allow to keep on working with non-metric proximity data despite the problems of indefiniteness: (1) leave the data non-psd and develop models that can handle non-psd data, and (2) modify the data to become psd in order to apply solid models with a solid mathematical foundation. An in-depth survey on correction techniques to process non-psd data is given in [5] and [20]. By applying such techniques, MKL can be used, but choosing an appropriate correction approach is not straightforward. Additionally, there is still no memory and computationally efficient technique to combine multiple perspectives employing multiple similarity functions.

III. Embedding indefinite kernels with multiple perspectives

Recently, an embedding technique for non-metric proximity data at low cost and moderate approximation error, resulting in vectorial representation, was introduced in [11]. Initially, the method was applied to only a single similarity function. In this paper, however, the input data are processed by multiple similarity functions. For each similarity function referred to as *perspective*, we will create an individual vectorial embedding matrix following the procedure in [11]. Subsequently, all individual perspectives are combined in one (potentially complex-valued) vector space. By employing a metric relevance learning technique within a prototype-based classification model, the various perspectives can be weighted during a training procedure. Consider a collection of F similarity functions $\{f_i\}$, $i = 1, 2, ..., F$. Let $\mathbf{S}_i \in \mathbb{R}^{N \times N}$ be the symmetric pairwise similarity matrix derived from the similarity function f_i. Next, the embedding matrix $\mathbf{M}_i$ is obtained by either embedding $\mathbf{S}_i$ using the real-valued embedding variant of [4] or using the complex-valued embedding from [11].

The real-valued embedding can be calculated by an eigen decomposition and a projection:

$$\mathbf{S}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \qquad \mathbf{V} = \mathbf{U}\left|\mathbf{\Lambda}\right|^{\frac{1}{2}}, \tag{1}$$

resulting in an embedding matrix $\mathbf{M}_i$ in a real-valued vector space. The so-called flipping strategy (the $|\cdot|$ operator in Eq. (1)), as suggested in [4], now ensures eigenvalues ≥ 0 and hence no complex values in the projection matrix. A complex-valued embedding is obtained by removing the $|\cdot|$ operator after the eigen decomposition and projection:

$$\mathbf{S}_i = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \qquad \mathbf{V} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}, \tag{2}$$

which leads to a complex-valued representation of the original matrix $\mathbf{S}_i$. Additionally to the embedding projection, we also recommend a Nyström approximation to ensure low computational costs. In previous research [21], the Nyström approximation has been proven to remain valid for generic proximity data, in particular non-psd similarities. Hence the Nyström approximation becomes available to approximate a non-psd matrix. Our work helps twofold to permit an effective embedding of multiple proximities: (1) the input does not need to be a kernel but can also be a similarity matrix, and with respect to [21] also a dissimilarity matrix and (2) the Nyström matrix approximation can also be applied on non-psd similarities which reduces the costs of the embedding In the Nyström approximation, one has to specify the number l of landmarks with $l \ll N$. The landmarks can be selected for non-psd matrices randomly or with a clustering strategy such as kmeans++ as shown in [22].

Depending on whether real-valued or complex-valued embedding is to be performed, the complete pipeline of our so-called *multi-perspective embedding* (MPE) is modeled according to Algorithm 1.

Algorithm 1 Multi-perspective Embedding

```
Input: $S := \{\mathbf{S}_1, \ldots, \mathbf{S}_F\}, l$
Output: $\mathbf{P}^*$ with $\mathbf{x}_i :=$ i-th row of $\mathbf{P}^*$
  $\mathbf{P}^* := []$
  for $\mathbf{S}_i$ in $S$ do
    $\mathbf{K}_1, \mathbf{K}_2 :=$ Nyström-Approximation($\mathbf{S}_i, l$)    ▷ using [5] and [22]
    $[\mathbf{C}, \mathbf{A}] :=$ eig($\mathbf{K}_2$)
    if complex-valued then
      $\mathbf{W} :=$ diag(sqrt(1./diag($\mathbf{A}$))) $\cdot$ $\mathbf{C}^T$    ▷ complex-valued embed.
    else
      $W :=$ diag(sqrt(1./diag($|\mathbf{A}|$))) $\cdot$ $\mathbf{C}^T$    ▷ real-valued embed
    end if
    $\mathbf{M} := \mathbf{W} \cdot \mathbf{K}_1$
    $\mathbf{P}^* := [\mathbf{P}^*, \mathbf{M}]$    ▷ perspective concatination
    $\mathbf{K}^* := \mathbf{M}' \cdot \mathbf{M}$    ▷ reconstruction (optional)
  end for
```

With the suggested approximation techniques and following further ideas discussed in [5], the aforementioned procedure can be done within linear costs. For a better intuition of Alg. 1, the complete process for our multi-perspective embedding is illustrated in Fig. 2 with the Morgan Fingerprint bit-vectorial representation as input data.

For comparing two embedded vectors, the application of a norm operator provides a dissimilarity score between the vectors:

$$d(\mathbf{x}, \mathbf{x}') = \mathbf{\Omega}(\mathbf{x} - \mathbf{x}'), \tag{3}$$

with $\mathbf{\Omega}$ a linear projection matrix. This $\mathbf{\Omega}$-matrix can be learned as outlined in [23]. This matrix permits interpretability of the single perspectives, as shown later in Sec. V-D.

Since the embedding of the input data causes complex-valued data, we simply need a classification model that can handle complex-valued matrices.

IV. Complex-valued classification with Generalized Learning Vector Quantization

Currently, there are only a few classification models that can handle complex-valued data like the complex-valued support vector machine (cSVM) [24], the complex-valued generalized learning vector quantization (cGMLVQ) [23], or a complex-valued neural network (cNNet) [25]. Further, a nearest neighbour (NN) classifier can be used by employing a standard norm operator. While cSVM, cGMLVQ, cNNet are parametric methods, the NN classifier is parameter-free and can be used directly.

Models from the Learning Vector Quantization (LVQ) family are defined by a set of labeled prototypes and a distance measure $d(\cdot,\cdot)$. New data is classified according to the nearest prototype's label using the distance measure $d(\cdot,\cdot)$. In contrast to the NN classifier in which the entire data set is used, the classes in LVQ schemes are represented by only very few prototypes. Hence, after training, LVQ methods require less computational effort and storage. Moreover, LVQ is often praised for its white-box character, which is beneficial in many applications [26].

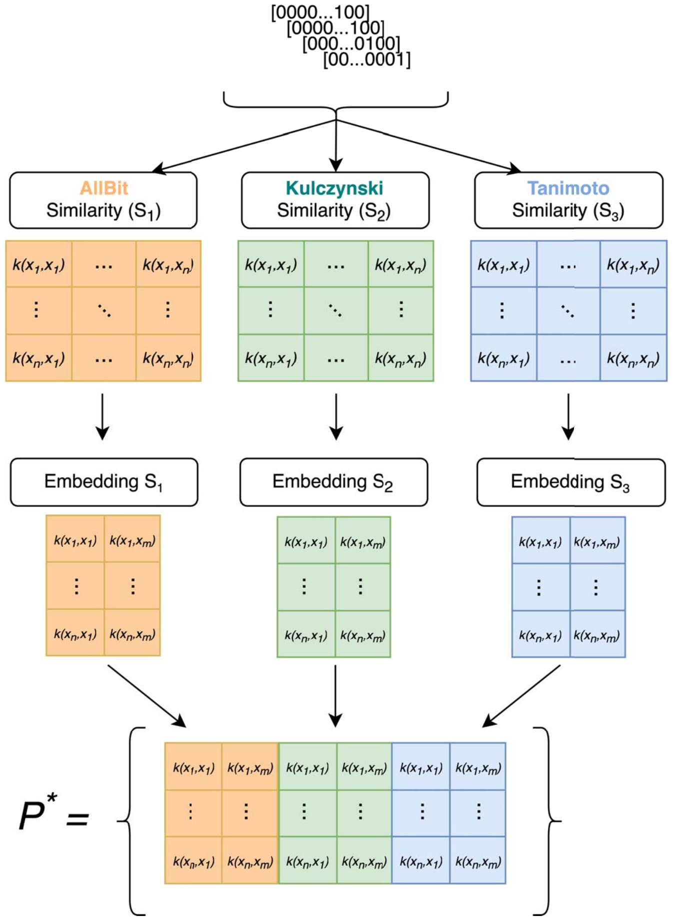

Fig. 2. Graphical illustration of the multi-perspective embedding procedure: starting with the input data, various similarity functions $\mathbf{S}_i$ are applied to the data resulting in F similarity matrices (in the actual processing pipeline of Alg. 1, the similarity matrices are not explicitly calculated, only the necessary score values). Next, the similarities are embedded according to the embedding strategy. Finally, all embedded similarities are concatenated to one large feature vector in $\mathbf{P}^*$.

A. Training an LVQ Classifier

Given a training data set of N labeled inputs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, in which $\mathbf{x}_i \in \mathbb{R}^d$ is an input vector and $y_i \in \{1, 2, ..., K\}$ its class label. The aim of the training procedure is the adaptation of M labeled prototypes $\{(\mathbf{w}_k, y_k)\}_{k=1}^M$ to the training data, such that the resulting classification scheme gives high classification accuracy with respect to the hold-out test data and new unseen data. Like in kernel machines, the proximity measure (here a dissimilarity measure) of choice $d(\cdot,\cdot)$ is of central importance for the model's performance. A common choice is squared Euclidean distance measure $(\mathbf{x} - \mathbf{w})^T(\mathbf{x} - \mathbf{w})$. In [27], a valid cost function for the LVQ heuristic was proposed that can be minimized by, e.g., gradient descent:

$$E_{GLVQ} = \sum_{i=1}^{N} \Phi(\mu_i), \text{ with } \mu_i = \frac{d_+(\mathbf{x}_i) - d_-(\mathbf{x}_i)}{d_+(\mathbf{x}_i) + d_-(\mathbf{x}_i)}. \tag{4}$$

The argument μ_i is based on the difference between the distance $d_+(\mathbf{x}_i)$ from its position to the closest prototype with the same label and the distance $d_-(\mathbf{x}_i)$ to the closest prototype with a different label, normalized to the range $\mu_i \in [-1, 1]$. The function $\Phi(\cdot)$ is monotonically increasing and is usually chosen to be identity $\Phi(x) = x$ or the logistic function $\Phi(x) = 1/(1+\exp(-x))$. The standard Euclidean distance does not account for differences in the classification importance of the dimensions. To improve classification accuracy, matrix *relevance learning* was introduced in [28]. A full matrix of adaptive relevances $\mathbf{\Lambda} = \mathbf{\Omega}^T\mathbf{\Omega}$ is introduced in the distance measure:

$$d^{\Lambda}(\mathbf{w}, \mathbf{x}_i) = (\mathbf{x}_i - \mathbf{w})^T \mathbf{\Omega}^T \mathbf{\Omega} (\mathbf{x}_i - \mathbf{w}) \tag{5}$$

The linear projection defined by the matrix $\mathbf{\Omega}$ is adapted during training to reflect each feature's importance.

The cost function in Eq. (4) is minimized with respect to the prototypes $\{\mathbf{w}_k\}_{k=1}^M$ and the linear projection matrix $\mathbf{\Omega}$ by either batch- or stochastic gradient descent. To formulate the gradient descent update rules with respect to $\mathbf{w}_\pm$ and $\mathbf{\Omega}$ for an example $\mathbf{x}_i$, one applies the chain rule:

$$\begin{aligned}\mathbf{w}_\pm &= \mathbf{w}_\pm - \alpha\Phi'(\mu_i)\frac{\partial \mu_i}{\partial d_\pm}\frac{\partial d_\pm}{\partial \mathbf{w}_\pm},\\ \mathbf{\Omega}_\pm &= \mathbf{\Omega}_\pm - \beta\Phi'(\mu_i)\frac{\partial \mu_i}{\partial d_\pm}\frac{\partial d_\pm}{\partial \mathbf{\Omega}_\pm},\end{aligned} \tag{6}$$

with the learning rates α and β. For all results reported in the following, we have set $\alpha = 0.01$ and $\beta = 0.001$.

B. Learning Rules for Complex-Valued Data

When the data has been embedded in a complex-valued space and one uses the Hermitian transpose in Eq. (5), the distance is always real-valued since it is a sum of squared magnitudes. Hence, only the innermost derivatives of the distance measure in Eq. (6) have to be considered with respect to the complex-valued variables. Complex-valued derivations can be done using the Wirtinger differential operators [29] as proposed in [23]:

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \; \frac{\partial}{\partial z^*} = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right), \tag{7}$$

in which $z = x + iy$ and $z^* = x - iy$, the complex conjugate. Using the differential operator with respect to z^*, the innermost derivatives in Eq. (6) are as follows:

$$\begin{aligned}\frac{\partial d}{\partial \mathbf{w}_\pm^*} &= -\mathbf{\Omega}^H\mathbf{\Omega}(\mathbf{x}_i - \mathbf{w}_\pm),\\ \frac{\partial d}{\partial \mathbf{\Omega}^*} &= \mathbf{\Omega}(\mathbf{x}_i - \mathbf{w}_\pm)(\mathbf{x}_i - \mathbf{w}_\pm)^H,\end{aligned} \tag{8}$$

which are conceptually similar to the derivatives of the real-valued variables. Finally, it is noteworthy that the (c)GMLVQ model can be trained with linear costs based on the vectorial data.

V. Experiments

In this section, we evaluate the proposed multi-perspective embedding on various data sets from life science domains. The data sets are briefly described in the following to provide intuition on their properties. In order to show the effectiveness of our approach, we compare the techniques. At the end of this section, a more in-depth review of the subsample size is given, as it may significantly contribute to the quality of the multi-perspective embedding.

TABLE I
Properties of the different data sets.
Details are given in the textual description.

Data set	#perspectives	#samples	#classes	source
FlowCyto	4	612	3	[30]
Sugar	3	1350	9	[31]
Swiss-Prot	8	14991	15	[32]
Tox 21: NR-AhR	10	8164	2	[33]
Tox 21: NR-AR	10	9357	2	[33]
Tox 21: NR-ER	10	7693	2	[33]
Tox 21: SR-ARE	10	7164	2	[33]
Tox 21: SR-HSE	10	8146	2	[33]
Tox 21: SR-MMP	10	7316	2	[33]
Tox 21: SR-p53	10	8629	2	[33]

A. Data sets

Each data set in this experimental setup consists of multiple $N \times N$ similarity matrices according to the number of used proximity functions. All similarity matrices of a data set had different spectral properties and consequently differed in their degree of indefiniteness. For a brief overview of the properties of the individual data sets, see Tab. I.

Each data set and its corresponding preprocessing pipeline are now briefly described in more detail:

1) The **FlowCyto** data set is based on 612 FL3-A DNA flow cytometer histograms from breast cancer tissues in 256 resolution, divided into three classes.In total, this data set consists of 4 proximity matrices of size 612×612, each representing the same original histogram data but with different parameterisations in the L1 norm proximity measure [30]. The proximity matrices are given as dissimilarities and must be translated to similarity matrices by a procedure called double centering[1].
2) **Sugar** data set is a benchmark data set for multi-modal data evaluation, taken from [31]. This data set offers multiple descriptions of sugar data taken by different optical sensors at various wavelengths. In our experimental setup, the information for each of the 1350 data points was available in 3 different channels divided into 9 classes. We used the currently prominent Wasserstein distance [34] to calculate the proximity matrices, followed by double centering.
3) **Swiss-Prot** consists of 14991 protein sequences taken as a subset of the famous Swiss-Prot database [32]. The sequence data are categorised according to their primary chemical characteristics into 15 classes. The generation of the proximity matrices was done by the alignment functions Smith-Waterman algorithm and Needleman-Wunsch algorithm [35]. Each alignment function was processed with different parameterisation, resulting in a total of 10 various perspectives. It is worth noting that

[1] $\mathbf{S} = -\mathbf{JDJ}/2$ with $\mathbf{J} = (\mathbf{I} - \mathbf{1}\mathbf{1}^\top/N)$, identity matrix $\mathbf{I}$ and vector of ones $\mathbf{1}$.

the intrinsic complexity of this data set is exceptionally high, which makes it very difficult to approximate in general.

4) The original intention of the **Tox21** challenge was to improve the development of computational methods for toxicity prediction of certain chemical compounds and their effects on particular physical processes. The main task here was to decide whether certain substances have a toxic effect on specific body regions (2 class classification). Initially, the challenge contained 12 physical effects, such as stress response effects (SR) or effects on nuclear receptors (NR). Exemplary, we used the following six out of the original 12 assays of the challenge to evaluate our approach: **NR-AhR**, **NR-AR**, **NR-ER**, **SR-ARE**, **SR-HSE**, **SR-MMP**. For more details on the challenge and the data itself, see [33]. For each of these 6 data sets, we used 10 different proximity functions from the RDKIT framework [36] to compare chemical compounds following the pipeline from Fig. 1. As shown in Tab. I, the sizes of the Tox 21 data sets vary as not all compounds are available in each physical process.

B. Evaluation Models

In the experimental setup, we tested various models in order to evaluate the benefit of our multi-perspective embedding compared to a baseline classifier and methods from the MKL domain:

- Nearest neighbour (NN): As a baseline model, we used the nearest neighbour classifier with the respective proximity function as a similarity measure. Here, the proximity matrices are averaged into one single proximity matrix to infer an overall label prediction. This approach is computationally expensive since either all distances have to be recalculated after each iteration or the entire dissimilarity matrix needs to be cached in memory.
- Easy and Average Multiple Kernel Learning: EasyMKL and AverageMKL are two methods from the multiple kernel learning domain, implemented in MKLpy [7] used in combination with a Support Vector Machine (SVM) whose parameters were optimized via a grid search. The MKL models have been calculated on the various $\mathbf{S}_i$.
- Nearest neighbour on multi-perspective embedded data (MPE-NN & MPE-cNN): These two classification models are included in the experiments to illustrate the benefits of a plain multi-perspective embedding without any advanced learning method for complex-valued data. Subsequently, we employed a simple nearest-neighbour classifier on this (potentially complex-valued) vectorial data. Similar to the NN classifier above, this classification method is still not efficient, but as long as the embedding process preserved the neighbourhood relationships of the data, the model yields accurate results.
- Complex-valued Generalized Learning Vector Quantization: We employ the cGMLVQ algorithm as presented in Sec. IV as a variant of a prototype-based learning algorithm for classifications. To interlink the multi-perspective embeddings and to scale the importance of the various contributions, relevance learning is emplyed. For simplicity, we employ one prototype per class.

C. Results

We evaluate the performance of our proposed multi-perspective embedding on the aforementioned data sets from Sec. V-A using all classification models from Sec. V-B and their respective hyperparameters. All classifiers are evaluated in a five-fold cross-validation with a hold-out test set with accuracy and standard deviation as performance metric. The subsample size for the (complex-valued) multi-perspective embedding of the complete $N \times N$ similarity matrix was 5% of the original $N \times N$ - matrix. The performances of the considered algorithms on the particular data sets are shown in Tab. II.

In our experiments, the baseline model NN performed competitively against the other methods in general. Actually, its accuracy was best on the data sets Swiss-Prot, NR-AhR, and SR-MMP. When dealing with a dataset as complex as SwissProt, using a nearest neighbour classifier that has access to the entire data as reference data may be beneficial. The performance of NN on the remaining data sets was similar to that of the MKL-models EasyMKL and AverageMKL. By averaging all perspectives, this method becomes attractive on the accuracy side, but considering the computational effort, as a lazy learner, this model is highly inefficient for multimodal analysis. Besides Swiss-Prot, NR-AhR, and SR-MMP, MKL-models performed similarly to the baseline classifier and only in case of the Swiss-Prot data set, both EasyMKL and AverageMKL outperformed the (c)GMLVQ variants with MPE. However, it is essential that these models are only valid for psd kernels. Accordingly, there is no guarantee of an optimal solution during the optimisation process.

Although the two models MPE-NN and MPE-cNN were primarily employed only to show that the MPE did not destroy the neighbourhood relationships, the two models achieved competitive results in some cases compared to the baseline classifier NN. This is particularly noteworthy because only 5% of the similarity scores had to be calculated for the MPE instead of the complete $N \times N$ similarity matrices. Overall, the two classifiers showed a slightly weaker - in some cases slightly better - performance than NN, EasyMKL and AverageMKL. The best results over most data sets have been obtained by the prototype-based classification models, especially by the complex-valued multi-perspective embedded variant. Using the classical embedding (MPE-GMLVQ) according to [4] resulted in only slightly inferior results than the complex-valued GMLVQ (MPE-cGMLVQ). Overall, Swiss-Prot remained the only data set where both MPE-GMLVQ and cGMLVQ struggled, probably grounded in the complex inherent structure of this data set. In summary, our multi-perspective embedding approach provided very promising results on most data sets while demonstrating memory- and runtime-efficiency.

TABLE II
CLASSIFICATION RESULTS (MEAN ± STANDARD-DEVIATION)

	No Embedding			Real Embedding		Complex Embedding	
Data set	NN	EasyMKL	AverageMKL	MPE-NN	MPE-GMLVQ	MPE-cNN	MPE-cGMLVQ
FlowCyto	0.62 ± 0.03	0.63 ± 0.04	0.63 ± 0.02	0.61 ± 0.04	0.68 ± 0.02	0.61 ± 0.03	**0.69 ± 0.02**
Sugar	0.60 ± 0.03	0.63 ± 0.04	0.63 ± 0.02	0.59 ± 0.04	0.91 ± 0.01	0.61 ± 0.02	**0.92 ± 0.02**
Swiss-Prot	**0.90 ± 0.01**	0.62 ± 0.05	0.72 ± 0.03	0.90 ± 0.01	0.84 ± 0.01	0.90 ± 0.00	0.83 ± 0.01
Tox21: NR-AhR	**0.92 ± 0.01**	0.87 ± 0.00	0.89 ± 0.00	0.91 ± 0.01	0.91 ± 0.01	0.91 ± 0.01	**0.92 ± 0.01**
Tox21: NR-AR	0.96 ± 0.00	0.97 ± 0.00	0.97 ± 0.00	0.97 ± 0.01	**0.98 ± 0.00**	0.96 ± 0.00	**0.98 ± 0.00**
Tox21: NR-ER	0.87 ± 0.01	0.87 ± 0.01	0.88 ± 0.00	0.86 ± 0.01	**0.90 ± 0.00**	0.86 ± 0.01	**0.90 ± 0.00**
Tox21: SR-ARE	0.87 ± 0.01	0.85 ± 0.00	0.84 ± 0.00	0.86 ± 0.01	0.86 ± 0.00	0.86 ± 0.01	**0.88 ± 0.01**
Tox21: SR-HSE	0.95 ± 0.00	0.95 ± 0.00	0.95 ± 0.00	0.94 ± 0.01	**0.96 ± 0.00**	0.94 ± 0.01	**0.96 ± 0.00**
Tox21: SR-MMP	**0.90 ± 0.01**	0.84 ± 0.00	0.85 ± 0.00	0.89 ± 0.01	0.89 ± 0.00	**0.90 ± 0.01**	**0.90 ± 0.01**

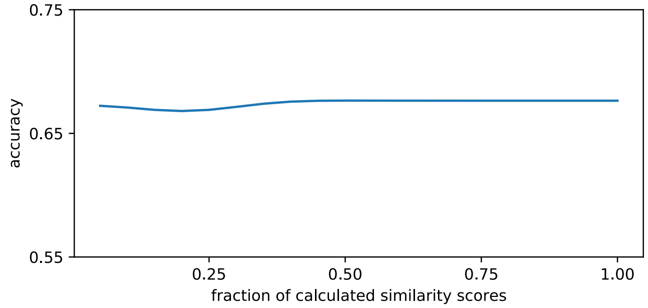

Fig. 3. Progression of accuracy with increasing percentage of input data. The x-axis indicates the subsample percentage of all n input data points.

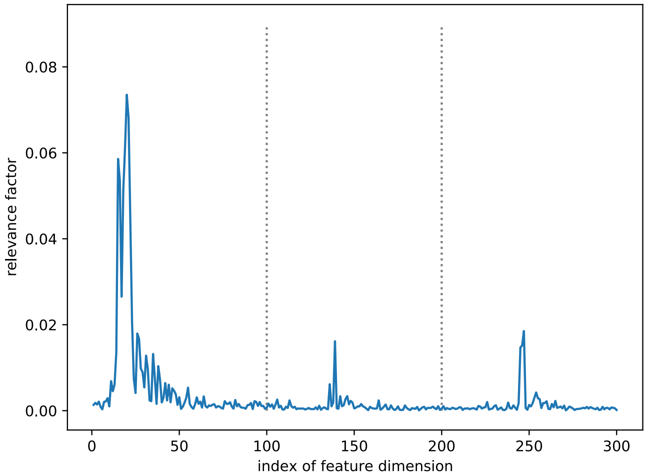

Fig. 4. Relevance profiles of the embedded feature vectors of the sugar data set. The y-axis illustrates the assigned weights during GMLVQ training for each feature vector (captured by its index on the x-axis). The dotted vertical lines indicate the sections of the 3 different similarity functions.

D. Scalability and Interpretability

In addition to its runtime and memory efficiency, our approach also provides scalability and interpretability.

Scalability and approximation techniques are indispensable in life science data analysis, especially in the context of Big Data. Working with dense matrices and not approximating the proximity matrices leads to an explosion in runtime [21]. With the highly efficient Nyström approximation in Algorithm 1, our approach requires only a small fraction of all similarity scores to be calculated. The only critical aspect when applying approximation techniques is whether they suffer from evaluation metrics. Commonly, any dimensionality reduction or low-rank approximation implies a reduction in accuracy, resulting in a less effective classifier. Fig. 3 illustrates the accuracy progression for the FlowCyto data set based on the subsample percentage of calculated similarity scores. Starting with 5% of the input data, we increased the percentage after each five-fold cross-validation by 5% until the entire input data is used.

The reached accuracy scores remained in a small range across all tested subsample percentages. Consequently, the accuracy behaved almost independently to the size of the subsample taken from the original data. We observed this behaviour in both the FlowCyto data set and other data sets, which are not included in this paper due to spatial limitations.

As **interpretability** also plays an increasingly important role in the development of classification models, we will have a brief look at the interpretability of the MPE-cGMLVQ model. Like all LVQ methods, employing a GMLVQ model allows the identification of representative and important data points. New data points are classified based on closest prototype's label. This allows any false prediction to be investigated and interpreted based on the prototypes coordinates.

Additionally, employing relevance learning allows the interpretation of the input data's features. Since relevance learning assigns weights to the dimensions of the input elements, it is possible to identify which dimensions were particularly important in the classification process. Although the dimensions are no longer directly related to the input data due to the embedding of the data in a new vector space, nevertheless they allow an interpretation of the importance of the similarity function. Exemplary, we chose a relevance profile that emerged during GMLVQ training using the Sugar data set. The weights assigned to the embedded vectors by cGMLVQ during the training process are shown in Fig. 4.

For convenient visualisation, we set the embedding size to 100 dimensions per similarity function. The dashed lines separate the individual fractions of P^*, created with the respective proximity functions. Overall, the relevance learning process revealed only a few very relevant features in the data. Considering both the summed and average relevance scores of the three similarity function segments, the sections of similarity function 2 and 3 are significantly lower than those of similarity function 1. Consequently, the most important perspective rated by relevance learning is the first similarity function.

In summary, both the interpretation of the model's prototypes and the relevance profiles provide convenient ways for interpretability. In addition to the promising experimental results, interpretability and scalability by landmark subsampling increase the applicability of our approach.

VI. Conclusions

In this paper, we presented a fast and efficient strategy for learning from multiple non-vectorial sources by means of indefinite proximity functions. By embedding all available perspectives in a complex-valued vector space, we are not only able to transfer structured data into a vectorial representation using indefinite similarity functions. This approach also enables the training of a model with several different data descriptions (i.e. kernel or similarity functions) at the same time. Compared to other methods from multi-modal data analysis, our approach is competitive and, in some cases, significantly better. Moreover, our approach provides high interpretability, memory efficiency, and less computational complexity. Our initial findings on this approach seem promising, although there is still much potential for further improvements, in particular a detailed comparison with embedding methods from deep learning. In this paper, we focused exclusively on life science data analysis, however, there are numerous other domains in which non-metric or indefinite proximity functions are commonly used. As our approach is widely adaptable for similar purposes, there will be further opportunities to apply our complex-valued multi-perspective embedding approach in the future.

References

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st ICLR 2013*, 2013.

[2] I. J. Goodfellow, Y. Bengio, and A. C. Courville, *Deep Learning*, ser. Adaptive computation and machine learning. MIT Press, 2016.

[3] M. Grohe, "Word2vec, node2vec, graph2vec, X2vec: Towards a Theory of Vector Embeddings of Structured Data," in *Proc. of the ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, 2020.

[4] E. Pekalska and R. P. W. Duin, *The Dissimilarity Representation for Pattern Recognition - Foundations and Applications*. WorldScientific, 2005.

[5] F. Schleif and P. Tiño, "Indefinite proximity learning: A review," *Neural Comput.*, vol. 27, no. 10, pp. 2039–2096, 2015.

[6] M. Münch, C. Raab, M. Biehl, and F.-M. Schleif, "Data-driven supervised learning for life science data," *Frontiers in Applied Mathematics and Statistics*, vol. 6, p. 56, 2020.

[7] I. Lauriola and F. Aiolli, "Mklpy: a python-based framework for multiple kernel learning," *CoRR*, vol. abs/2007.09982, 2020.

[8] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[9] F. Aiolli and M. Donini, "Easymkl: a scalable multiple kernel learning algorithm," *Neurocomputing*, vol. 169, pp. 215–224, 2015.

[10] Y. Li, M. Yang, and Z. Zhang, "A Survey of Multi-View Representation Learning," *IEEE Transact. on Knowledge and Data Eng.*, 2019.

[11] M. Münch, M. Straat, M. Biehl, and F. Schleif, "Complex-valued embeddings of generic proximity data," in *S+SSPR 2020*, ser. LNCS, vol. 12644. Springer, 2020, pp. 14–23.

[12] E. Asgari and M. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, 2015.

[13] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA*. ACM, 2016, pp. 855–864.

[14] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, 2021.

[15] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[16] M.-P. Dubuisson and A. Jain, "A modified hausdorff distance for object matching," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, 1994, pp. 566–568 vol.1.

[17] I. Lauriola, M. Polato, and F. Aiolli, "The minimum effort maximum output principle applied to multiple kernel learning," in *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*, 2018.

[18] G. Loosli, S. Canu, and C. S. Ong, "Learning svm in krein spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 6, pp. 1204–1216, June 2016.

[19] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA, USA: MIT Press, 1999, pp. 185–208.

[20] S. Mehrkanoon, X. Huang, and J. A. K. Suykens, "Indefinite kernel spectral learning," *Pattern Recognit.*, vol. 78, pp. 144–153, 2018.

[21] A. Gisbrecht and F. Schleif, "Metric and non-metric proximity transformations at linear costs," *Neurocomputing*, vol. 167, pp. 643–657, 2015.

[22] D. Oglic and T. Gärtner, "Nyström method with kernel k-means++ samples as landmarks," in *Proc. of the 34th Int. Conf. on Machine Learning, ICML 2017, Sydney, NSW, Australia, 2017*, ser. Proc. of Machine Learning Research, vol. 70. PMLR, 2017, pp. 2652–2660.

[23] M. Straat, M. Kaden, M. Gay, T. Villmann, A. Lampe, U. Seiffert, M. Biehl, and F. Melchert, "Learning vector quantization and relevances in complex coefficient space," *Neur. Comp. and Applications*, Mar 2019.

[24] L. Zhang, W. Zhou, and L. Jiao, "Complex-valued support vector classifiers," *Digital Signal Processing: A Review Journal*, 2010.

[25] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *6th ICLR 2018*, 2018.

[26] R. van Veen, V. Gurvits, R. Kogan, S. Meles, G. de Vries, R. Renken, M. Rodriguez-Oroz, R. Rodriguez-Rojas, D. Arnaldi, S. Raffa, B. de Jong, K. Leenders, and M. Biehl, "An application of generalized matrix learning vector quantization in neuroimaging," *Comp. Methods and Programs in Biomedicine*, vol. 197, p. 105708, 2020.

[27] A. Sato and K. Yamada, "Generalized learning vector quantization," in *Advances in Neur. Inf. Processing Systems*, vol. 8. MIT Press, 1996.

[28] P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural Comput.*, vol. 21, no. 12, p. 3532–3561, Dec. 2009.

[29] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Mathem. Annalen*, vol. 97, pp. 357–376, 1927.

[30] R. P. Duin, "PRTools," 2012. [Online]. Available: http://www.prtools.org

[31] F. Melchert, A. Matros, M. Biehl, and U. Seiffert, "The sugar dataset - a multimodal hyperspectral dataset for classification and research," 2016.

[32] B. Boeckmann, A. Bairoch, R. Apweiler, M.-C. Blatter, A. Estreicher, E. Gasteiger, M. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, and M. Schneider, "The swiss-prot protein knowledgebase and its supplement trembl in 2003,," *Nucleic Acids Research*, vol. 31, pp. 365–370, 2003.

[33] T. in the 21st Century program, "Tox21 challenge," 2014. [Online]. Available: https://tripod.nih.gov/tox21/challenge/index.jsp

[34] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *31st Int. Conference on Machine Learning, ICML 2014*, 2014.

[35] D. Gusfield, *Algorithms on strings, trees, and sequences: computer science and computational biology*, 1997.

[36] G. Landrum, "Rdkit: Open-source cheminformatics." [Online]. Available: http://www.rdkit.org