dom2vec: Capturing domain structure and function using self-supervision on protein domain architectures

Damianos P. Melidis (damianosmel@gmail.com)
Leibniz Universitat Hannover
Brandon Malone
NEC Laboratories Europe
Wolfgang Nejdl
Leibniz Universitat Hannover

Research article

Keywords: Protein domain architectures, InterPro; Machine learning, Neural networks, Word embeddings, Quality assessment, SCOPe secondary class, Enzymatic Commission class

Posted Date: August 28th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-58816/v1

Version of Record: A version of this preprint was published at Algorithms on January 19th, 2021. See the published version at https://doi.org/10.3390/a14010028.

RESEARCH

dom2vec: Capturing domain structure and function using self-supervision on protein domain architectures

Damianos P. Melidis[1*], Brandon Malone[2] and Wolfgang Nejdl[1,3]

Abstract

Background: Word embedding approaches have revolutionized natural language processing (NLP) research. These approaches aim to map words to a low-dimensional vector space, in which words with similar linguistic features cluster together. Embedding-based methods have also been developed for proteins, where words are amino acids and sentences are proteins. The learned embeddings have been evaluated qualitatively, via visual inspection of the embedding space and extrinsically, via performance comparison on downstream protein prediction tasks. However, these sequence embeddings have the caveat that biological metadata do not exist for each amino acid, in order to measure the quality of each unique learned embedding vector.

Results: Here, we present *dom2vec*, an approach for learning protein domain embeddings using *word2vec* on InterPro annotations. In contrast to sequence embeddings, biological metadata do exist for protein domains, related to each domain separately. Therefore, we present four *intrinsic* evaluation strategies to quantitatively assess the quality of the learned embedding space. To perform a reliable evaluation in terms of biology knowledge, we selected the metadata related to the most distinctive biological characteristics of domains. These are the structure, enzymatic and molecular function of a given domain. These evaluations allow us to assess the quality of learned embeddings independently of a particular downstream task. Notably, *dom2vec* obtains adequate level of performance in the intrinsic assessment, therefore we can draw an analogy between the local linguistic features in natural languages and the domain structure and function information in domain architectures. Moreover, we demonstrate the *dom2vec* applicability on protein prediction tasks, by comparing it with state-of-the-art sequence embeddings in three downstream tasks. We show that *dom2vec* outperform sequence embeddings for toxin and enzymatic function prediction and is comparable with sequence embeddings in cellular location prediction.

Conclusions: We report that the application of *word2vec* on InterPro annotations produces domain embeddings with two significant advantages over sequence embeddings. First, each *unique* dom2vec vector can be quantitatively evaluated towards its available structure and function metadata. Second, the produced embeddings can outperform the sequence embeddings for a subset of downstream tasks. Overall, *dom2vec* embeddings are able to capture the most important biological properties of domains and surpass sequence embeddings for a subset of prediction tasks. Hence, researchers can reliably use them for domain architecture and protein prediction tasks.

Keywords: Protein domain architectures; InterPro; Machine learning; Neural networks; Word embeddings; Quality assessment; SCOPe secondary class; Enzymatic Commission class

Background

A primary way of how proteins evolve is through rearrangement of their functional and structural units, known as *protein domains* [1, 2]. The domains are independent folding and functional modules and exhibit conserved sequence segments. Prediction algorithms exploited this information and use the domain composition of a protein as input features for various tasks. For example, [3] classified the cellular location and [4, 5] predicted the associated Gene Ontology (GO) terms. One common way to represent the do-

*Correspondence: melidis@l3s.uni-hannover.de
[1]L3S Research Center, Leibniz University Hannover, Hannover, Germany
Full list of author information is available at the end of the article

main composition of a protein is by the linear order in a protein, *domain architecture* [6].

Moreover, [7] investigated if the "language" of domain architectures has a grammar as a natural spoken language. They compared the bi-gram entropy of *domain architecture* for Pfam domains [8] to the respective entropy of the English language, showing that although it was lower than the English language, it was significantly different from a language produced after shuffling the domains. Prior to this result, methods had exploited *domain architecture* representation for various applications, such as fast homology search [9] and retrieval of similar proteins [10].

Word embeddings are unsupervised learning methods that have as input large corpora and they output a dense vector representation of words contained in the sentences of these documents based on the distributional semantic hypothesis. By this assumption, the meaning of a word can be understood by its context. Thus a word vector encapsulates local linguistic features, such as lexical or semantical information, of the respective word. Several methods to train word embeddings have been established [11, 12, 13]. These representations have shown to hold several properties such as analogy and grouping of semantically similar words [14, 15]. Word embeddings are currently the mainstream input for neural networks in NLP as their use improved the performance on most of the tasks.

Various methods to create embeddings for proteins were proposed [16, 17, 18, 19, 20, 21, 22]. ProtVec fragmented the protein sequence in 3-mers for all possible starting shifts. Then they learned embeddings for each 3-mer and represented the respective protein as the average of its constituting 3-mer vectors [16]. SeqVec utilized and extended the Embeddings from Language Models (ELMo) [23] to learn a dense representation per amino acid residue. SeqVec embedding resulted in matrix representations of proteins, created by concatenating their learned residue vectors [20].

The previous embedding approaches evaluated the learned representations qualitatively and quantitatively. For qualitative evaluation, they averaged out the whole protein amino acid embeddings to compute the aggregated vector. Then they used known biological characteristics of proteins, (biophysical, chemical, structural, enzymatic and taxonomic), as distinct colors in a reduced 2-D embedding space. In such visualizations, they reported the appearance of distinct clusters of proteins, each consisting of proteins with similar properties. For quantitative evaluation, they measured the improvement of performance in downstream tasks.

Focusing on qualitative evaluation of existing protein embeddings, we confirm two caveats. First, researchers averaged out the protein amino acid vectors, consequently this qualitative evaluation is not related in a straightforward way with each learned embedding vector trained per amino-acid. To add on, this averaging-out operation may not reveal the function of the most important sites of a protein; making the comparison result holding a low degree of biological significance. Second, we argue that the presented qualitative evaluations lack the ability to assess different learned embeddings in a sophisticated manner. This is because there is no a systematic way to compare *quantitatively* 2-D plots of reduced embedding spaces, each produced by a protein embedding method in investigation.

Indeed for word embeddings, there is an increase in methods to evaluate word representations intrinsically and in quantitative manner such as [24, 25]. Having such evaluation metrics allows us to validate the knowledge acquired *per each word vector* and use the best performing space for downstream tasks. However, intrinsic evaluations of current amino acid embedding representations are prevented by not complete biological metadata at amino acid level, for all disposed proteins, in UniProtKnowledgeBase (UniProtKB) [26].

To address this evaluation shortcoming of protein sequence embeddings we make five major contributions:

1 We propose *dom2vec*, in which words are InterPro annotations and sentences are the domain architectures. Then we use *word2vec* method, to learn the embedding vector representation for each InterPro annotation.

2 We established a novel intrinsic evaluation method based on the most significant biological information for a domain; its structure and function. First, we evaluated the learned embedding space by domain hierarchy. Then, we investigated the performance of a nearest neighbor classifier, $C^d_{nearest}$, to predict the secondary structure class provided by SCOPe secondary structure class [27] and the Enzyme Commission (EC) primary class. Finally, we equally examined the performance of $C^d_{nearest}$ classifier to predict the GO molecular function class for three example model organisms and one human pathogen.

3 Strikingly, we observed that $C^d_{nearest}$ reaches adequate accuracy, compared to $C^d_{nearest}$ on randomized domains vectors, for secondary structure, enzymatic and GO molecular function. Thus we hypothesized an analogy between word embedding clustering by local linguistic features and protein domains clustering by domain structure and function.

4 To evaluate our embeddings extrinsically, we inputted the learned domains embeddings to simple neural networks and compared their performance

with state-of-the-art protein sequence embeddings in three full-protein tasks. We surpassed both SeqVec and ProtVec for the toxin presence and enzymatic primary function prediction task and we reported comparable results in the cellular location prediction task.

5 We make available the trained domains embeddings to be used by the research community.

Methods

Our approach is summarized in Figure 1, in the following we explain the methodology for each part of the approach.

Building domain architecture

InterPro database contains functional annotations for superfamily, family and single domains as well as functional protein sites. Hereafter, we will refer to all such functional annotations as InterPro annotations. Furthermore, we will denote *domain architecture* the ordered arrangement of domains in a protein. We consider two distinct strategies to represent a protein based on it domain architecture consisting of either *non-overlapping* or *non-redundant* annotations. These annotation types are defined as:

Non-overlapping annotations. For each region with overlapping InterPro annotations, all InterPro annotations except the longest are removed.

Non-redundant annotations. For each region with overlapping InterPro annotations, all InterPro annotations with the same InterPro identifier, except the longest InterPro annotations with unique identifier, are removed.

To efficiently resolve the annotation overlaps, an interval tree was built for each protein to detect overlapping domains, and each protein was split into regions with overlapping domains. After resolving overlaps, the admitted annotations in each protein were sorted by start position to construct its domain architecture. Following the approach of [5] we also added the "GAP" domain to annotate more than 30 amino acid subsequence that does not match any InterPro annotation entry.

Training domain embeddings

Given a protein, we assumed that words are its resolved InterPro annotations and sentences are the protein domain architecture. By this assumption, we learned task-independent embeddings for each InterPro annotation using two variants of *word2vec*: continuous bag of words and skip-gram model; hereafter denoted as CBOW and SKIP. See [12] for technical details on the difference between these approaches. Through this training, each InterPro annotation is associated with a task-independent embedding vector.

Novel intrinsic evaluation methods

Previous works evaluated the quality of embeddings only indirectly by measuring performance on downstream tasks. Nevertheless, in NLP, the quality of a learned word embedding space is often evaluated *intrinsically* by considering relationships among words, such as analogies. Such an evaluation is important, as it ensures the learned embeddings are meaningful without choosing a specific downstream task.

In the following, we used the metadata for the most characteristic properties of domains, in order to evaluate the learned embedding space for various hyperparameters of *word2vec*. We propose four intrinsic evaluation approaches for domain embeddings: domain hierarchy based on the family/subfamily relation, SCOPe secondary structure class, EC primary class, and GO molecular function annotation.

We refer to the embedding space learned by *word2vec* for a particular set of hyperparameters as V_{emb}. We refer to the k nearest neighbors of a domain d as $C^d_{nearest}$ found using the Euclidean distance.

Random *Interpro annotation* vectors To inspect the relative performance of V_{emb} on each of the following evaluations, we *randomized* all domain vectors and run each evaluation task. That is, we assigned to each domain vector a newly created random vector, for each unique dimensionality of embedding space, irrespective of all other embedding method parameters.

Domain hierarchy

InterPro defines a strict family-subfamily relationship among domains. This relationship is based on sequence similarity of the domain signatures. We refer to the children of domain p as S_p. We use these relationships to evaluate an embedding space, posing the following research question,

$RQ_{hierarchy}$: Did vectors of hierarchically closely domains form clusters in the V_{emb}?

Evaluation We predicted the closest $|S_p|$ domains on cosine similarity of their vector to the parent vector and we denote this predicted set as $\hat{S}_p$. For all learned embedding spaces, we measured their recall performance, $Recall_{hier}$ defined as follows:

$$Recall_{hier} = \sum_p \frac{|S_p \cap \hat{S}_p|}{|S_p|}. \quad (1)$$

SCOPe secondary structure class

We extracted the secondary structure of Interpro domains from the SCOPe database and form the following research question,

RQ_{SCOPe}: Did vectors of domains, with same enzymatic primary class, form clusters in the V_{emb}?

Evaluation We evaluated V_{emb} by retrieving $C^d_{nearest}$ of each domain. Then, we applied stratified 5-fold cross validation and measured the performance of a k-nearest neighbor classifier to predict the structure class of each domain. The intrinsic evaluation performance metric is the average accuracy across all folds, $Accuracy_{SCOPe}$.

EC primary class
The enzymatic activity of each domain is given by its primary EC class [28] and pose the following research question,

RQ_{EC}: Did vectors of domains, with enzymatic primary class, form clusters in the V_{emb}?

Evaluation We again evaluate V_{emb} using k nearest neighbors in a stratified 5-fold cross validation setting. Average accuracy across all folds is again used to quantify the intrinsic quality of the embedding space.

GO molecular function
For our last intrinsic evaluation, we aimed to assess V_{emb} using the molecular function GO annotation. We extracted all molecular function GO annotations associated with each domain. In order to account for differences in specificity of different GO annotations, we always used the depth-1 ancestor of each annotation, i.e. children of the root molecular function term, GO:0003674.

Since model organisms have the most annotated proteins we created GO molecular function data sets for one example prokaryote (*Escherichia coli* denoted *E.coli*), one example simple eukaryote (*Saccharomyces cerevisiae* denoted *S.cerevisiae*) and one complex eukaryote (*Homo sapiens* denoted *Human*). To assess our embeddings also for not highly annotated organisms, we included a molecular function data set for an example human pathogen (*Plasmodium falciparum*, denoted as *Malaria*). Finally, we pose the following research question,

RQ_{GO}: Did vectors of domains, with same GO molecular function, form clusters in the V_{emb}?

Evaluation We again evaluate an embedding space using k nearest neighbors in a stratified 5-cross validation setting. Average accuracy across all folds is again used to quantify performance.

Qualitative evaluation

As a preliminary evaluation strategy, we used qualitative evaluation approaches adopted in existing work. To follow the qualitative approach of ProtVec and SeqVec we also visualized the embedding space for selected domain superfamilies, to answer the following research question,

$RQ_{qualitative}$: Did vectors of each domain superfamily form a cluster in the V_{emb}?

Evaluation To find out, we added the vector of each domain in a randomly chosen domain superfamily to an empty space. Then we performed principle component analysis (PCA) [29] to reduce the space in two dimensions and observed the formed clusters.

Extrinsic evaluation

In addition to assess the learned V_{emb}, we also examined the performance change in downstream tasks. For the three supervised task TargetP, Toxin, and NEW, we feeded the domain representations in simple neural networks and compare the performance of our model with the state-of-the-art protein embeddings, ProtVec and SeqVec.

TargetP
This data set is about predicting the cellular location of a given protein. We downloaded the TargetP data set provided by [30] and we also used the non-plant data set. This data set consists of 2 738 proteins accompanying with their uniprot id, sequence and the cellular location label which can be nuclear, cytosol, pathway or signal and mitochondrial. Finally, we removed all instances with duplicate set of domains, resulting in total of 2 418. This is a multi-class task and its class distribution is summarized in Supplementary section E.

Evaluation For the TargetP we used the mc-AuROC performace metric.

Toxin
[31] introduced a data set associating protein sequence to toxic or other physiological content. We used the hard setting which provides uniprot id, sequence and the label toxin content or non-toxin content, for 15 496 proteins. Finally, we kept only the proteins with unique domain composition, resulting to 2 270 protein instances in total. This is a binary task and the class distribution is shown in Suppl. E.

Evaluation As Toxin data set is binary task, we used AuROC as performance metric.

NEW
The NEW data set [32] contains the data for predicting the enzymatic function of proteins. For each of the 22 618 proteins the data set provides sequence and the EC number class. The primary enzyme class, first digit of EC number, is our label on this prediction task, resulting in a multi-class task. Finally, we removed all instances with duplicate domain composition, resulting in a total of 14 434 protein instances. The possible classes are 6 and the class distribution is shown in supplementary section E.

Evaluation NEW data set is a multi-class task, thus we used mc-AuROC as performance metric.

Data partitioning
We divided each data set into 70/30% train and test splits. To perform model selection, we created inner three-fold cross validation sets on each train split.

Out-of-vocabulary experiment We observed that the performance of classifier depending on protein domains is highly dependent on the out-of-vocabulary (OOV) domains, as first discussed in [33]. OOV domains are all the domains contained in the test set, *but* not in the train. For TargetP, Toxin and NEW we observed that approximately 60%, 20%, 20% of test proteins contain *at least one* OOV domain, respectively.

For TargetP, containing the highest OOV we experimented to compensate on high degree of OOV. We split the test set into shorter sets by an increasing degree of OOV, namely 0%, 10%, 30%, 50%, 70%, 100%. Then we trained models for the whole train set and benchmarked the performance on each of these test subsets.

Generalization experiment For Toxin and NEW data set, experiencing low OOV, we sought to investigate the generalization of the produced classifier. We increased the number of training examples that the model was allowed to learn from and we benchmarked *always* in the entire test set. To do so, we created training splits of size, 10%, 20%, 50% of the whole train set. To perform significance testing we trained on 10 random subsamples for each training split percentage and then test on the separate step set. We used the paired sample t-test, Benjamini-Hochberg multiple-test, to compare the performance between a pair of classifiers on the test set.

Simple neural models for prediction
We consider a set of simple, well-established neural models to combine the InterPro annotation embeddings for each protein to perform downstream tasks, that is, for *extrinsic* evaluation tasks. In particular, we use FastText [34], convolutional neural networks (CNNs) [35], and recurrent neural networks (RNNs) with long-short term memory (LSTM) cells [36] and bi-directional LSTMs.

Results

Building domain architecture

We used the domain hits for UniProt proteins from the InterPro version 75, containing 128 660 257 proteins with InterPro signature, making up the 80.9% of the total UniProtKB proteome (version 2019_06). For all these proteins, we extracted the non-overlapping and non-redundant sequences, which we processed in the next section. The number of unique non-overlapping sequence was (35 183 + 1) plus the "GAP" domain and non-redundant was (36 872 + 1) plus the "GAP". Comparing to the total number of domains in InterPro version 75, which was 36 872, we observed that non-overlapping InterPro annotations captured 95.42% and the non-redundant captured 100% of the InterPro annotation entries. To enable visual comparison of the created type of domain architectures versus the downloaded InterPro annotations, in Figure 2 we illustrated the non-overlapping and non-redundant domain architectures of *Diphthine synthase* protein. This same protein, *Diphthine synthase*, was picked as example illustration for annotations in the latest InterPro work [37].

Training domain embeddings

Domain architecture
Before applying the *word2vec* method we examined the histograms of number of non-overlapping and non-redundant InterPro annotations per protein in Figure 3. We observed that these distributions are long-tailed with mode equal to 1 and 3 respectively. Then, we used both CBOW and SKIP algorithms to learn domain embeddings. We used the following parameters sets. Based on the histograms, we selected the context window parameter for word to be 2 or 5, $w = \{2, 5\}$. For the number of dimensions, we used common values from the NLP literature, $dim = \{50, 100, 200\}$. We trained the embeddings from 5 to 50 epochs with step size 5 epochs $ep = \{5, 10, 15, \ldots, 50\}$. Finally, all other parameters were set to their default values. For example, the negative sampling parameter was left to default, ng=5.

Novel intrinsic evaluation

In the following, we evaluated each instance of learned embedding space V_{emb} for both non-overlapping and non-redundant representations of domain architectures. An instance of V_{emb} space is the embeddings space learned for each combination of the product $non\_overlap \times w \times dim \times ep$. Consequently, the total number of embeddings space instances is $|non\_overlap| \times |w| \times |dim| \times |ep| = 2 \times 2 \times 3 \times 10 = 120$. Let V^i_{emb} denote such embedding space instance. In the following subsection, we evaluated each V^i_{emb} instance for domain hierarchy, secondary structure, enzymatic primary class and GO molecular function. Finally, all reported performances are shown *for the best performing epoch value (ep).*

$RQ_{hierarchy}$: Did vectors of hierarchically closely domains form clusters in the V_{emb}?
For the first research question, we loaded the parent-child tree T_{hier}, provided by InterPro, consisting of

2 430 parent domains. Then for each V^i_{emb} we compared the actual and predicted children of each parent, and we averaged out the recall for all parents. For ease of presentation, we show only the results for non-redundant InterPro annotations at Table 1a and we provide the complete results in the Suppl. A.

From Tables S1 and 1a (Suppl. A), we observed that SKIP performed better overall, and the embeddings learned from non-redundant InterPro annotations always have better average recall values compared to non-overlapping. The best performing V^i_{emb} achieved average $Recall_{hier}$ of 0.538. We compared this moderate performance of V_{emb} with the performance of the randomized spaces, which was equal to 0. We concluded that our embedding spaces greatly outperformed each randomized space for domain hierarchy relation. Therefore, we admitted that the *majority* of domains of the same hierarchy were placed in close proximity in the embedding space.

RQ_{SCOPe}: Did vectors of domains, with same secondary structure class, form clusters in the V_{emb}?
We extracted the SCOPe class for each InterPro domain. This resulted to 25 196 domains with unknown secondary structure class, 9 411 with single secondary structure class and 2 265 domains with more than one assigned classes (multi-label). For clarity, we removed all multi-label and unknown instances resulting to 9 411 single-labeled instances. The class distribution of the resulting data set is shown in the Suppl. B.

We measured the performance of $C^d_{nearest}$ classifier in each V^i_{emb} to examine the homogeneity of the space with respect to the SCOPe class. We split the 9 411 domains in 5-fold stratified cross validation sets. To test the change in prediction accuracy for an increasing number of neighbors, we used different sets of neighbors, namely, $k = \{2, 5, 20, 40\}$. We summarized the results for the best performing $C^d_{nearest}$ which was for $k = 2$ for non-redundant InterPro annotations in Table1b. We show the respective table for non-overlapping InterPro annotations in the Suppl. B. We compared these accuracy measurements to the respective ones of the random spaces, and we found that the lowest accuracy values, achieved for CBOW with w=5 using non-overlapping domains, are twice as high as the accuracy values of the random spaces for all possible dimensions. Consequently, we concluded that domain embeddings of the same secondary structure class formed distinct clusters in the learned embedding space.

RQ_{EC}: Did vectors of domains, with same enzymatic primary class, form clusters in the V_{emb}?
We processed the EC primary class, resulting in 29 354 domains with unknown EC, 7 248 domains with only one EC and 721 with more than one EC. As before, we removed all multi-label and unknown instances leaving 7 428 domains with known EC. We augmented a domain instance with its vector representation for each V^i_{emb}, and then we used $C^d_{nearest}$to predict the EC label. See Suppl. C for the class distribution of EC task.

We reported the average $Accuracy_{EC}$ obtained on embedding space learned using non-redundant InterPro annotations in Table 1c. We show the respective table for non-overlapping in the Suppl. C. We compared these accuracy measurements to the respective ones of the random spaces. We found that the minimum average $Accuracy_{EC}$ value was equal to 60.51 and was achieved using CBOW w=5 for non-overlapping InterPro annotations. That value was approximately twice as large than the accuracy values of the random spaces for all possible dimensions; the maximum average $Accuracy_{EC}$ for random space with dim=100 was 32.64. Hence, we were able to accept that domain embeddings of the same EC primary class formed distinct clusters in a learned embedding space.

RQ_{GO}: Did vectors of domains, with same GO molecular function, form clusters in the V_{emb}?
We parsed the GO annotation file of InterPro to extract first-level GO molecular function for domains for the four organisms. We followed the same methodology to examine the homogeneity of a V_{emb} with respect to GO molecular function annotations. For each V^i_{emb}, we augmented each domain by its vector and its GO label, and we classified each domain using $C^d_{nearest}$. As before, we used 5-fold stratified cross-validation for evaluation. In our experiments, we varied the number of neighbors $k = \{2, 5, 20, 40\}$ to test its influence to the change of performance.

For space limitations, we summarized the performances showing only the best average accuracy over the number of neighbors. For the ease of presentation we omit the resulted tables for the three first organisms and show only for *Human*, but we discuss the results for all organisms. See Suppl. D for full results.

For *Malaria*, the best average accuracy was 76.86 and the minimum was 56.94. We compared this moderate minimum accuracy to the maximum accuracy obtained by the randomized embedding space, which was 47.57. Therefore, we concluded that *dom2vec* embeddings outperformed the random baseline by at least ten percent.

For *E.coli*, the best accuracy was 81.72 and the minimum was 67.34. By comparing with the random baseline, achieving best accuracy of 64.46, we observed that again *dom2vec* was able to surpass the random baseline.

For *Yeast*, the best accuracy was 75.10 and the minimum accuracy value was 59.82. We contrasted to

the maximum accuracy obtained in a random space, which was 53.73, to report that *dom2vec* vectors in $V_{emb}^{E.coli}$ captured GO molecular function classes in much higher degree than randomized vectors.

For *Human*, the best average performance for non-redundant InterPro annotations are shown in Table 1d. The best average accuracy was 75.96, scored by 2-NN for V_{emb}^{human} (non-redundant,SKIP,w=5,dim=50, ep=40). We compared the minimum accuracy values, obtained by CBOW with w=5, to accuracy values of the random spaces, and we found that the worst-performing *dom2vec* was 20 percentage values higher than the random baseline.

For all four example organisms, we observed that the SKIP on non-redundant InterPro annotations produced V_{emb} where $C_{nearest}^{d}$ achieved the best average accuracy in. For the three out of the four organisms, the best performances were achieved for the lowest number of dimensions (dim=50). In all cases, we found that the worst-performing *dom2vec* embeddings outperformed the random baselines. By these findings, we affirmed that domain embeddings of the same GO molecular function class formed distinct clusters in the learned embedding space.

Concluding on novel intrinsic evaluation

Based on the previous four experiments, we aimed to evaluate the learned V_{emb} spaces and select the best domain embedding space for downstream tasks. In all experiments, the non-redundant InterPro annotations created the better performing embedding spaces compared to non-overlapping annotations. We reasoned for this finding, by comparing the modes of number of annotations per protein for the two annotation types, Figure 3. We hypothesized that by the very low mode for non-overlapping annotations, mode equal to one annotation, the *word2vec* method could not produce embeddings for even the stringent context window value of two. In contrast, 52% of proteins contain less than or equal to three non-redundant InterPro annotations.

This makes *word2vec* able to produce embedding spaces producing the best intrinsic performance. From the individual results, we saw that the configuration of parameters (non-redundant, SKIP, w=5, dim=50) brought best results in $C_{nearest}^{d}$ performance for SCOPe, EC and GO for *E.coli*, *Yeast*, *Human*, second best for *Malaria* and the sixth best recall (0.507) for the domain hierarchy relation. Therefore, we will denote as $V_{emb}^{best\ intrinsic}$, the space produced by (non-redundant, SKIP, w=5, dim=50, ep=50).

$RQ_{qualitative}$: Did vectors of each domain superfamily form a cluster in the V_{emb}?

To explore the V_{emb} in terms of the last research question, $RQ_{qualitative}$, we randomly selected five InterPro domain superfamilies to perform the visualization experiment. The selected domain superfamilies were *PMP-22/EMP/MP20/Claudin superfamily* with parent InterPro id IPR004031, *small GTPase superfamily* with parent InterPro id IPR006689, *Kinase-pyrophosphorylase* with parent InterPro id IPR005177, *Exonuclease, RNase T/DNA polymerase III* with parent InterPro id IPR013520 and *SH2 domain* with parent InterPro id IPR000980.

We loaded the parent-child tree T_{hier}, provided by InterPro, and for each domain superfamily starting from the parent domain we included recursively all domains that have subfamily relation with this parent domain. For example, the *Kinase-pyrophosphorylase* domain superfamily had domain parent IPR005177, which in turn had two immediate domain subfamilies IPR026530, IPR026565. The IPR026565 domain contained a subfamily domain with id IPR017409, consequently the set of domains for *Kinase-pyrophosphorylase* domain superfamily was {IPR005177, IPR026530, IPR026565, IPR017409}. We retrieved the vectors for each domain in a domain superfamily using the $V_{emb}^{best\ intrinsic}$, the best performing *dom2vec* space selected previously.

Finally, we visualized the two-dimensional PCA-reduced space at Figure4. We recognized that domains embeddings of each superfamily organized well-separated clusters. From these cluster, the cluster of *Exonuclease, RNase T/DNA polymerase III* superfamily had the highest dispersion of all presented superfamilies. By this finding, we could answer to the research question: embedding vectors of the same superfamily clustered well in the learned V_{emb}.

Extrinsic evaluation

Extracting domain architecture

For each data set that contained the UniProt identifier for the protein instance, we extracted the domain architecture for non-redundant InterPro annotations, already created in Section "Building domain architecture". For all proteins whose UniProt identifier could not be matched, or for data sets not providing the protein identifier, we used InterProScan [38] to find the domain hits per protein. For proteins without a domain hit after InterProScan, we created a protein-specific, artificial protein-long domain; for example, we assigned to the protein G5EBR8, a protein-long domain named "G5EBR8_unk_dom".

Model selection

To select which simple neural model we should compare to the baselines, we performed hyperparameter selection using an inner, three-fold cross validation on the training set; the test set was not used to select hyperparameters. We used common parameters, dropout

of 0.5, batch size of 64, the Adam optimizer [39] with learning rate of 0.0003, weight decay for the last fully connected layer of 0 and number of epochs equal to 300. As a final hyperparameter, we allowed updates to the learned domain embeddings, initialized by selected *dom2vec* embeddings. The results are shown in Suppl. E.

Running baselines

Then, we used the same network as the one in right side of Figure 5 of [20]; we refer to this network as SeqVecNet. Namely, the network first averages the 100 (ProtVec) or 1 024 (SeqVec) dimensional embedding vector for a protein; it then applies a fully connected layer to compress a batch of such vectors into 32 dimensions. Next, a ReLU activation function (with 0.25 dropout) was applied to that vector, followed by batch normalization. Finally, another fully connected layer was followed by the prediction layer. As third baseline, we added the *1-hot* of domains in order to investigate the performance change compared to *dom2vec* learned embeddings.

Evaluation

For TargetP, we sought to investigate the effect of OOV on the produced classifier compared to sequence-based embeddings classifiers, which do not experience OOV as their used sequence features are highly common in both train and test set. For Toxin and NEW datasets, we benchmarked the generalization of the produced classifier compared to the sequence-based embeddings classifiers. Finally, for both kinds of experiments, we used the trained models on each test set. Hence, this evaluation shows how differences in the training set affect performance on the test set. The resulted performances are shown in Figure 5.

Out-of-vocabulary experiment For TargetP we validated that OOV will affect the performance of domains dependent classifiers. That is, for OOV in the range of $0 - 30\%$ the *dom2vec* classifier was comparable to the best performing model, SeqVec. However, when OOV increased more, then the performance of our model dropped, but still being competitive with the SeqVec. *dom2vec* greatly outperformed the *1-hot* representation, validating the NLP assumption that unsupervised embeddings improve classification on unseen words, in this context protein domains, compared to *1-hot* word (domain) vectors.

Generalization experiment For both Toxin and NEW, *dom2vec* significantly outperformed SeqVec, ProtVec and domains *1-hot* vectors, Benjamini--Hochberg multiple-test corrected p-value < 0.05. In Toxin data set, we observed that ProtVec learned the less variant model, but with the trade-off obtaining the lowest performance (mc-AuROC). For NEW data set, the *dom2vec 1-hot* representation was the second best representation outperforming SeqVec and ProtVec, allowing us to validate the finding that domain composition is the most important feature for enzymatic function prediction as concluded by [32].

Discussion

Comparison with Pfam domain embeddings

From all proposed protein embeddings works, only [22] developed intrinsic quantitative benchmarks. They applied *word2vec* for Pfam domain annotations for only eykaryotic proteins. They used the following three experiments to benchmark their Pfam domain embeddings. First, they benchmarked nearest neighbor, $C^d_{nearest}$, classifier performance on predicting the three main GO ontologies of a Pfam using its embedding vector. Second, they assessed the Matthew's correlation coefficient [40] between Pfam embedding and first-order Markov encodings. They also investigated if vector arithmetic holds by comparing each time two pairs of Pfam domains with mutually exclusive GO binary assignment, for example one pair consisting of two domains annotated as intra-cellular (GO:0005622) and the contrasting pair contained two domains annotated as extra-cellular (GO:0005615).

Building and evaluating Pfam domain embeddings is the closest approach to *dom2vec*. Our approach differs in four main points. First, we trained embeddings for all domain annotations of all proteins available in InterPro. We included all available InterPro annotations, consisting of super-family, family, single domains and functional sites, as "words" input to the *word2vec* method. Therefore, we used a broader set of annotations and for the whole spectrum of organisms. Besides, *word2vec* was developed for sentences of corpora of natural languages, which have a moderate number of words and a large number of sentences. To cope with the sentence length assumption, we resolved overlapping and redundant annotations, so as to increase the number of InterPro annotations, making our input more suitable for the *word2vec* method. To follow the second assumption on number of sentences, we inputted all proteins with InterPro annotation. This increased the number of sentences, used as *word2vec* input, about 14-fold, from 9 030 650 (Pfam domains) to 128 660 257 (*dom2vec*). Second, we benchmarked over the two *word2vec* models (CBOW and SKIP) and their parameters for each experiment of our intrinsic evaluation step. Then we used our established intrinsic evaluation to choose the best embedding space. Third, we established unique intrinsic evaluation benchmarks for the characteristic biological features of domains which are the secondary structure and the enzymatic function. We also formed intrinsic evaluation for another

important biological domain feature which is the hierarchy of domains. Last, *dom2vec* was also extrinsically evaluated on three downstream prediction tasks. We have shown that *dom2vec* embeddings could surpass established sequence embeddings for toxin and enzymatic function prediction and it was comparable to these embeddings for cellular location prediction. This downstream evaluation revealed that *dom2vec* not only capture the biological features of domains adequately, but it could also improve the performance on protein prediction tasks.

Was domain architecture informative enough for word2vec?

Our InterPro annotations histograms have shown skewed empirical distributions of the number of annotations (Figure 3). For non-overlapping, the mode is one annotation, *but* for non-redundant only 10% of proteins contained only one annotation. The latter form of annotations had a mode of three annotations, consequently *word2vec* was presented with an input corpus of applicable number of words per sentence, even for the shortest window of two (w=2).

We argue that despite the low modes of these distributions *word2vec* inputs were informative. To do so, we refer to the information gain distribution of the grammar created by Pfam domain architectures found by [7]. We see that even if the grammar distribution was computed for bi-grams (low window for n-grams compared to a natural language), the difference with the grammar distribution of randomized architectures was still significant. In our intrinsic evaluation, we did validate that such a corpus with a low mode of the number of words, domains, could still allow *word2vec* to produce embeddings that capture biological features known for domains.

The analogy between a natural language and protein domain architectures

We have shown that *dom2vec* captured adequately the domain SCOPe structural information, EC enzymatic function and the GO molecular function of each domain with such available metadata information. However, *dom2vec* produced moderate results in the domain hierarchy evaluation task. After investigating the properties of domain families that *dom2vec* produces these moderate results, we concluded that *dom2vec* cannot capture the domain hierarchy mostly for domain families of low cardinality. We argue that using more complex classifiers compared to $C^d_{nearest}$, we could gain in hierarchy performance, but this was not the scope of our evaluation.

Importantly, we did discovered that *dom2vec* embeddings captured the most distinctive biological characteristics of domains, secondary structure, enzymatic and molecular function, for an individual domain. That is *word2vec* produced domain embeddings that clustered sufficiently well by their structure and function class. Therefore, our finding supported the accepted modular evolution of proteins [1], in a data-driven way. Also, it made possible a striking analogy between words in natural language that clustered together in *word2vec* space [14] and domains in domain architectures that clustered together in *dom2vec* space. Therefore, we parallel the semantic and lexical similarity of words to the functional and structural resemblance of domains. This analogy may augment the research on understanding the nature of rules underlying the domain architecture grammar [7]. We are confident that this interpretability aspect of *dom2vec* will allow researchers to apply it, reliably, to predict biological features of novel domain architectures and proteins with identifiable InterPro annotation(s).

Boosting downstream prediction performance

In downstream task evaluation, *dom2vec* outperformed significantly domain 1-hot vectors and state-of-the-art sequence-based embeddings for Toxin and NEW data sets. For the TargetP *dom2vec* was comparable to the best performing sequence-based embedding, Seqvec, for OOV up to 30%.

As a consequence, if the protein prediction task enables the inclusion of additional input features, more than the mere amino-acid sequence, we recommend to extract InterPro annotations for the proteins at hand and use *dom2vec* in combination with sequence embeddings to boost predictive performance. For example, as researchers use domain information successfully for protein function prediction, we foresee that combining sequence embeddings and *dom2vec* will boost performance. If the prediction task does not allow the use of domain information, for example CASP (Critical Assessment of protein Structure Prediction), we suggest using *dom2vec* as a hard-to-beat baseline for sequence embedding classifiers.

Conclusions

We presented *dom2vec*, protein domain embeddings. We processed InterPro annotations to create domain architectures and then applied *word2vec* to these architectures. We introduced a novel intrinsic evaluation, based on metadata related to the most critical biological characteristics of an individual domain. In this evaluation, we found that *dom2vec* vectors cluster sufficiently well by secondary structure, enzymatic function and molecular function.

We then used *dom2vec* embeddings as input of simple neural networks for three different protein prediction tasks to compare their performance with

state-of-the-art sequence embeddings. We found that *dom2vec* models surpassed ProtVec and SeqVec models in two tasks, toxin and enzymatic function prediction and were comparable to sequence embeddings models in the task of cellular localization. We believe that *dom2vec* can be used reliably by the research community, to boost prediction performance for individual domains and whole proteins.

Acknowledgements
DPM would like to thank Sofia K. Forslund for the helpful suggestions on selecting metadata related to the most important biological characteristics of domains.

Author's contributions
DPM conceptualized the work, implemented the software for the methodology and evaluation parts. BM proposed the generalization experiment for the downstream evaluation. WN proposed the qualitative, visualization, experiment. DPM wrote the manuscript. BM and WN suggested improvements for the manuscript. All authors read and approved the final manuscript.

Funding
This study was funded by the Ministry for Science and Culture of Lower Saxony Germany (MWK: Ministerium fuer Wissenschaft und Kultur) within the project "Understanding Cochlear Implant Outcome Variability using Big Data and Machine Learning Approaches", project id: ZN3429.

Availability of data and materials
The code is available on github:
`https://github.com/damianosmel/dom2vec`.
Processed metadata for each intrinsic evaluation benchmark and downstream prediction data sets containing the protein domain architecture per protein instance are found in the same repository.

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]L3S Research Center, Leibniz University Hannover, Hannover, Germany. [2]NEC Laboratories Europe, Heidelberg, Germany. [3]Knowledge-based Systems Laboratory, Leibniz University Hannover, Hannover, Germany.

References

1. Moore, A.D., Björklund, Å.K., Ekman, D., Bornberg-Bauer, E., Elofsson, A.: Arrangements in the modular evolution of proteins. Trends in Biochemical Sciences **33**(9), 444–451 (2008)
2. Forslund, S.K., Kaduk, M., Sonnhammer, E.L.: Evolution of protein domain architectures, 469–504 (2019)
3. Chou, K.-C., Cai, Y.-D.: Using functional domain composition and support vector machines for prediction of protein subcellular location. Journal of Biological Chemistry **277**(48), 45765–45769 (2002)
4. Forslund, K., Sonnhammer, E.L.: Predicting protein function from domain content. Bioinformatics **24**(15), 1681–1687 (2008)
5. Doğan, T., MacDougall, A., Saidi, R., Poggioli, D., Bateman, A., O'Donovan, C., Martin, M.J.: UniProt-DAAC: domain architecture alignment and classification, a new method for automatic functional annotation in UniProtKB. Bioinformatics **32**(15), 2264–2271 (2016)
6. Scaiewicz, A., Levitt, M.: The language of the protein universe. Current Opinion in Genetics & Development **35**, 50–56 (2015)
7. Yu, L., Tanwar, D.K., Penha, E.D.S., Wolf, Y.I., Koonin, E.V., Basu, M.K.: Grammar of protein domain architectures. Proceedings of the National Academy of Sciences **116**(9), 3636–3645 (2019)
8. Sonnhammer, E.L., Eddy, S.R., Birney, E., Bateman, A., Durbin, R.: Pfam: multiple sequence alignments and HMM-profiles of protein domains. Nucleic Acids Research **26**(1), 320–322 (1998)
9. Terrapon, N., Weiner, J., Grath, S., Moore, A.D., Bornberg-Bauer, E.: Rapid similarity search of proteins using alignments of domain arrangements. Bioinformatics **30**(2), 274–281 (2013)
10. Marchler-Bauer, A., Bo, Y., Han, L., He, J., Lanczycki, C.J., Lu, S., Chitsaz, F., Derbyshire, M.K., Geer, R.C., Gonzales, N.R., *et al.*: CDD/SPARCLE: functional classification of proteins via subfamily domain architectures. Nucleic Acids Research **45**(D1), 200–203 (2016)
11. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25^{th} International Conference on Machine Learning (2008)
12. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the 1^{st} International Conference on Learning Representations (2013)
13. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26 (2013)
15. Drozd, A., Gladkova, A., Matsuoka, S.: Word embeddings, analogies, and machine learning: Beyond king-man+woman=queen. In: Proceedings of the 26^{th} International Conference on Computational Linguistics: Technical Papers (2016)
16. Asgari, E., Mofrad, M.R.K.: Continuous distributed representation of biological sequences for deep proteomics and genomics. PLOS ONE **10**(11), 0141287 (2015)
17. Yang, K.K., Wu, Z., Bedbrook, C.N., Arnold, F.H.: Learned protein embeddings for machine learning. Bioinformatics **34**(15), 2642–2648 (2018)
18. Bepler, T., Berger, B.: Learning protein sequence embeddings using information from structure. In: Proceedings of the 7^{th} International Conference on Learning Representations (2019)
19. Asgari, E., McHardy, A.C., Mofrad, M.R.K.: Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). Scientific Reports **9**(3577) (2019)
20. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., Rost, B.: Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinformatics **20**(723) (2019)
21. Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M., Church, G.M.: Unified rational protein engineering with sequence-based deep representation learning. Nature methods **16**(12), 1315–1322 (2019)
22. Buchan, D.W., Jones, D.T.: Learning a functional grammar of protein domains using natural language word embedding techniques. Proteins: Structure, Function, and Bioinformatics **88**(4), 616–624 (2020)
23. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 16^{th} Annual Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (2018)
24. Schnabel, T., Labutov, I., Mimno, D., Joachims, T.: Evaluation methods for unsupervised word embeddings. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (2015)
25. Lastra-Díaz, J.J., Goikoetxea, J., Taieb, M.A.H., García-Serrano, A., Aouicha, M.B., Agirre, E.: A reproducible survey on word embeddings and ontology-based methods for word similarity: linear combinations outperform the state of the art. Engineering Applications of Artificial Intelligence **85**, 645–665 (2019)
26. The UniProt Consortium: UniProt: the universal protein knowledgebase. Nucleic Acids Research **45**(D1), 158–169 (2017)
27. Fox, N.K., Brenner, S.E., Chandonia, J.-M.: SCOPe: Structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. Nucleic Acids Research **42**(D1), 304–309 (2013)
28. Fleischmann, A., Darsow, M., Degtyarenko, K., Fleischmann, W., Boyce, S., Axelsen, K.B., Bairoch, A., Schomburg, D., Tipton, K.F., Apweiler, R.: Intenz, the integrated relational enzyme database.

Nucleic acids research **32**(suppl_1), 434–437 (2004)
29. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901)
30. Emanuelsson, O., Nielsen, H., Brunak, S., Von Heijne, G.: Predicting subcellular localization of proteins based on their n-terminal amino acid sequence. Journal of Molecular Biology **300**(4), 1005–1016 (2000)
31. Gacesa, R., Barlow, D.J., Long, P.F.: Machine learning can differentiate venom toxins from other proteins having non-toxic physiological functions. PeerJ Computer Science **2**, 90 (2016)
32. Li, Y., Wang, S., Umarov, R., Xie, B., Fan, M., Li, L., Gao, X.: DEEPre: sequence-based enzyme EC number prediction by deep learning. Bioinformatics **34**(5), 760–769 (2017)
33. Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. In: Proceedings of the 53^{rd} Annual Meeting of the Association for Computational Linguistics (2015)
34. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15^{th} Conference of the European Chapter of the Association for Computational Linguistics (2017)
35. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., *et al.*: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
37. Mitchell, A.L., *et al.*: InterPro in 2019: improving coverage, classification and access to protein sequence annotations. Nucleic Acids Research **47**(D1), 351–360 (2019)
38. Jones, P., *et al.*: InterProScan 5: genome-scale protein function classification. Bioinformatics **30**(9), 1236–1240 (2014)
39. Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: Proceedings of the 3^{rd} International Conference on Learning Representations (2015)
40. Matthews, B.W.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure **405**(2), 442–451 (1975)

Figures

Figure 1: Summary of our approach divided in four parts, building two forms of domains architecture, training domain embeddings, performing intrinsic and extrinsic evaluation of *dom2vec* embeddings.

Figure 2: Non overlapping and non-redundant domain architectures of *Diphthine synthase* protein. Because all domains are overlapping with the largest one, colored in blue, the non-overlapping annotation is just the single longest domain (IPR035966). All other domains have a unique InterProid, so the set of non-redundant InterPro annotations includes all presented domains sorted by starting position; we colored all these other domains in fading hues of gray based on their starting position.

Tables

Additional Files

Additional file 1 — supplementary
The supplementary material is described in supplementary.pdf.

Figure 3: Histograms of number of InterPro annotations per protein. **a**: Non-overlapping and **b**: non-redundant annotations.

(a) Domains hierarchy

Model╲ Dimension	dim=50	dim=100	dim=200
CBOW, w=2	0.406 (ep=10)	0.412 (ep=10)	0.414 (ep=5)
CBOW, w=5	0.405 (ep=30)	0.402 (ep=35)	0.382 (ep=10)
SKIP, w=2	0.512 (ep=5)	0.53 (ep=5)	**0.538** (ep=5)
SKIP, w=5	0.507 (ep=5)	0.525 (ep=5)	0.524 (ep=5)
random	0	0	0

(b) SCOPe Secondary structure

Model╲ Dimension	dim=50	dim=100	dim=200
CBOW, w=2	77.09 (ep=5)	76.35 (ep=5)	75.77 (ep=5)
CBOW, w=5	78.15 (ep=5)	76.94 (ep=5)	76.84 (ep=5)
SKIP, w=2	84.42 (ep=45)	84.42 (ep=40)	84.08 (ep=30)
SKIP, w=5	**84.56** (ep=25)	84.06 (ep=45)	83.72(ep=10)
random	23.39 (k=40)	23.49 (k=40)	22.76 (k=20)

(c) EC primary class

Model╲ Dimension	dim=50	dim=100	dim=200
CBOW, w=2	76.88(ep=5)	75.85(ep=5)	75.39(ep=5)
CBOW, w=5	80.89(ep=5)	79.89(ep=5)	77.16(ep=5)
SKIP, w=2	89.47(ep=35)	89.06(ep=40)	88.86(ep=5)
SKIP, w=5	**90.85**(ep=30)	90.41(ep=15)	90.2(ep=5)
random	33.62 (k=40)	32.06 (k=40)	32.28 (k=40)

(d) GO molecular function (*Human*)

Model╲ Dimension	dim=50	dim=100	dim=200
CBOW, w=2	66.94(ep=5)	66.32(ep=5)	66.32(ep=5)
CBOW, w=5	67.77(ep=5)	65.87(ep=5)	65.77(ep=5)
SKIP, w=2	74.77(ep=40)	74.18(ep=5)	73.14(ep=5)
SKIP, w=5	**75.96**(ep=40)	75.53(ep=10)	74.98(ep=5)
random	37.05 (k=40)	37.03 (k=20)	37.05 (k=40)

Table 1: Intrinsic evaluation performance. **a**: $Recall_{hier}$ for non-redundant InterPro annotations. **b**,**c**&**d**: $C^{d}_{nearest}$ average accuracy over all folds: **b:** $Accuracy_{SCOPe}$, **c:** $Accuracy_{EC}$ and **d:** $Accuracy_{GO}$ for non-redundant InterPro annotations. For all tables, results shown for the best performing ep value; if k not shown then k=2. Best performance of an evaluation task shown in bold case

Figure 4: Visualization of domain vectors for five domain superfamilies in the *dom2vec* space with parameters ($V_{emb}^{best\ intrinsic}$). The percentage of variance explained by a principle component dimension, is shown in parenthesis.

Figure 5: Downstream performance. **a:** TargetP, OOV experiment: learning in whole train and benchmark in test splits of increasing out-of-vocabulary degree. **b**&**c:** Toxin and NEW, generalization experiment: learning in increasing train splits, 10 replicates each, and benchmark in whole test sets. The marked points represent the mean performance on the test set and the shaded regions show one standard deviation above and below the mean.

Figures

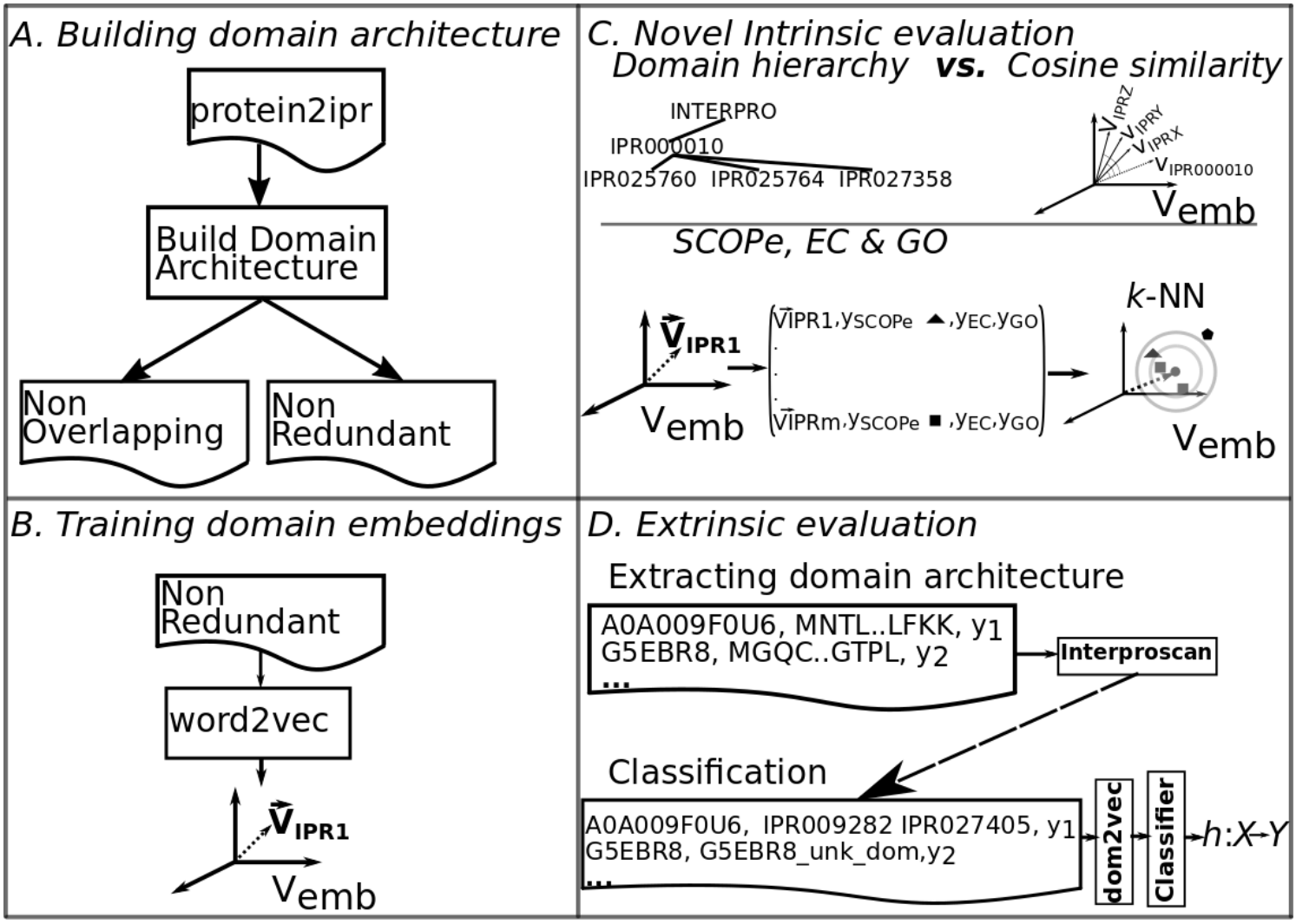

Figure 1

Summary of our approach divided in four parts, building two forms of domains archi- tecture, training domain embeddings, performing intrinsic and extrinsic evaluation of dom2vec em- beddings.

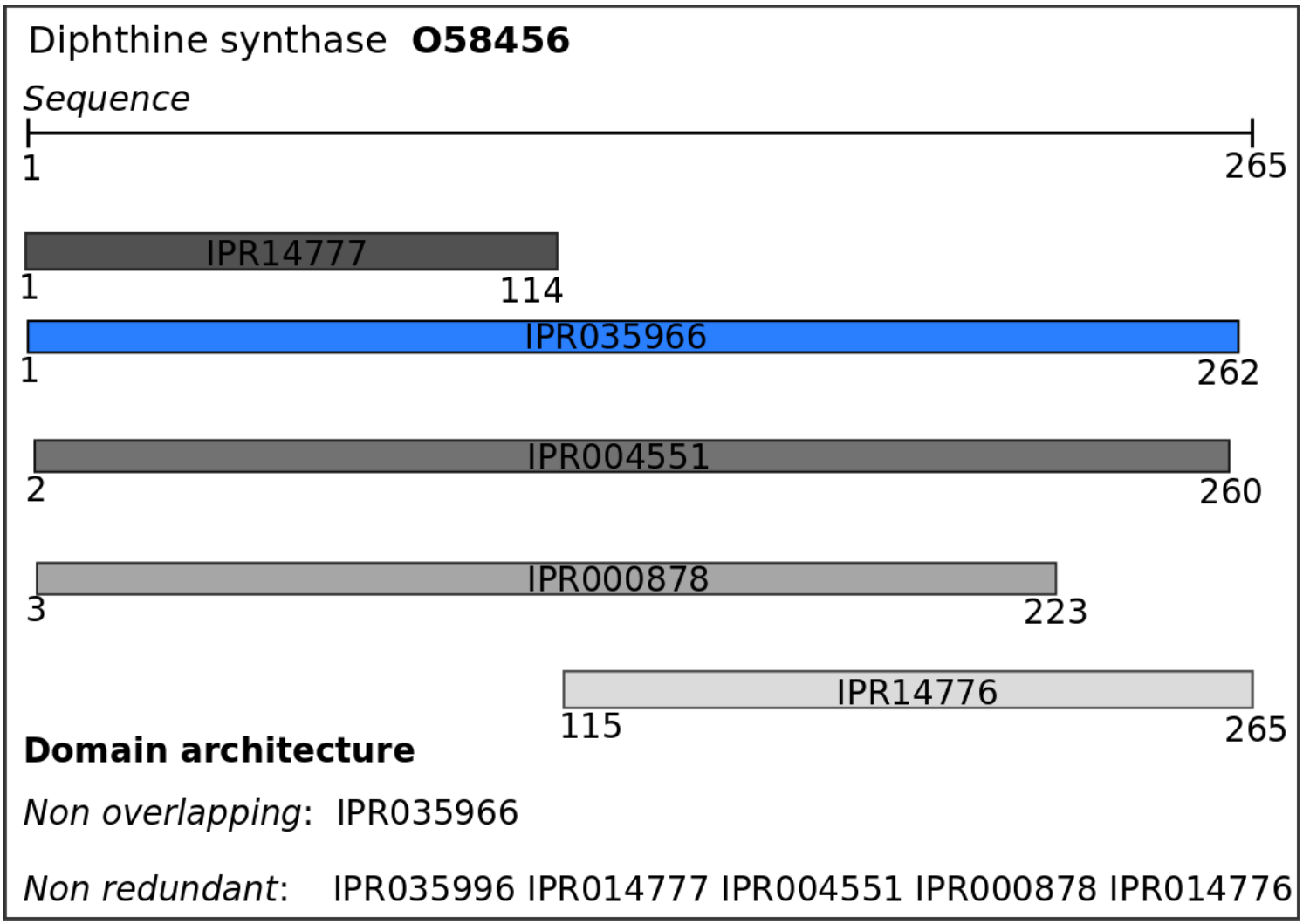

Figure 2

Non overlapping and non-redundant domain architectures of Diphthine synthase pro- tein. Because all domains are overlapping with the largest one, colored in blue, the non- overlapping annotation is just the single longest domain (IPR035966). All other domains have a unique InterProid, so the set of non-redundant InterPro annotations includes all presented do- mains sorted by starting position; we colored all these other domains in fading hues of gray based on their starting position.

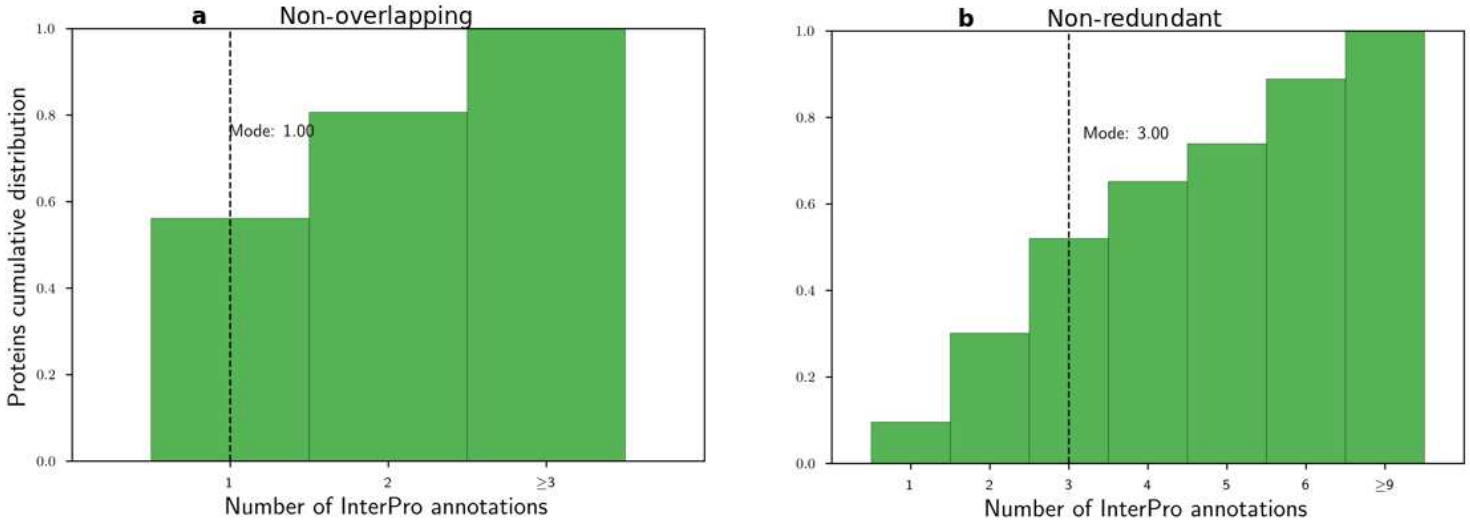

Figure 3

Histograms of number of InterPro annotations per protein. a: Non-overlapping and b: non-redundant annotations.

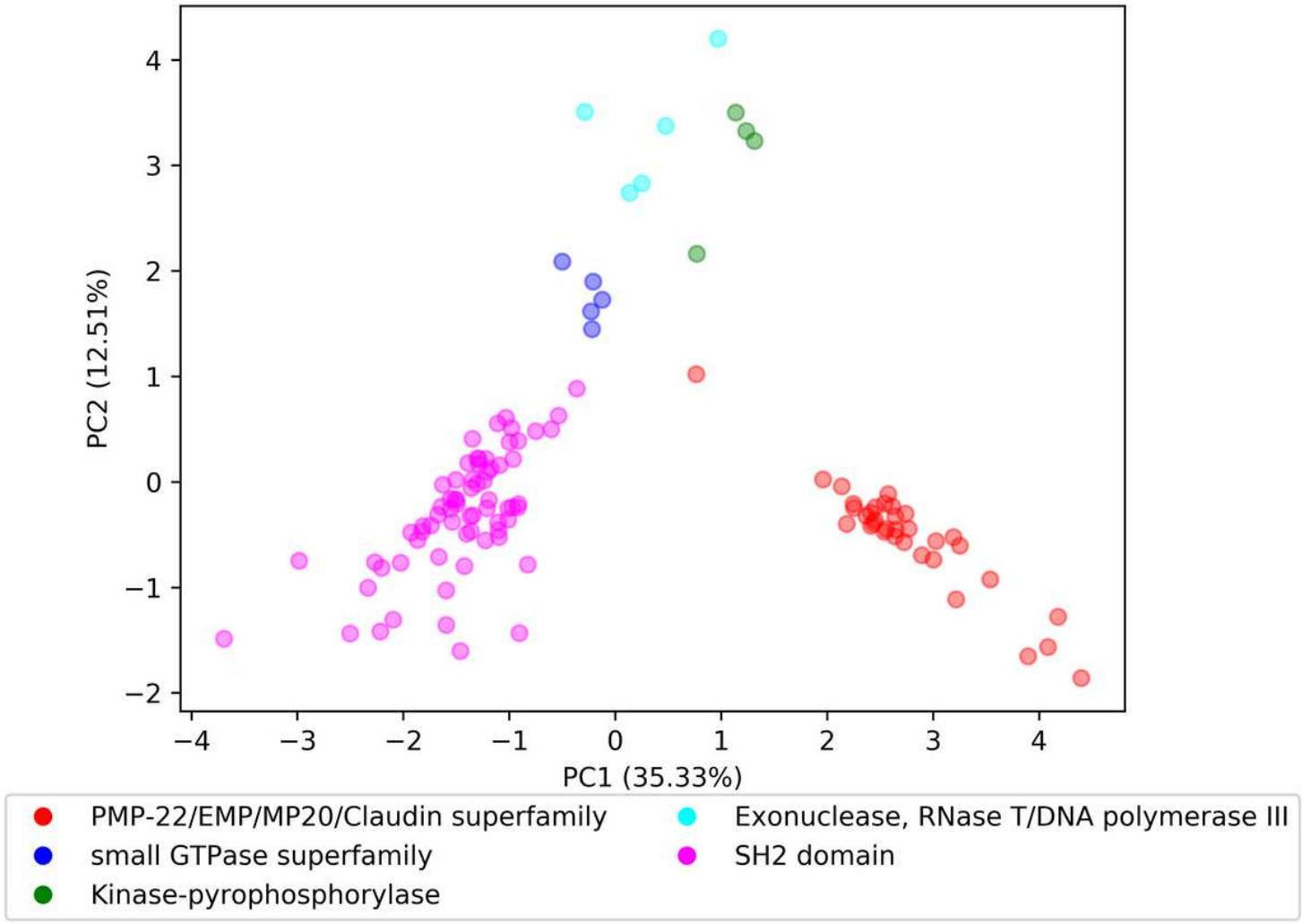

Figure 4

Visualization of domain vectors for ve domain superfamilies in the dom2vec space with parameters (V best intrinsic emb). The percentage of variance explained by a principle component di- mension, is shown in parenthesis.

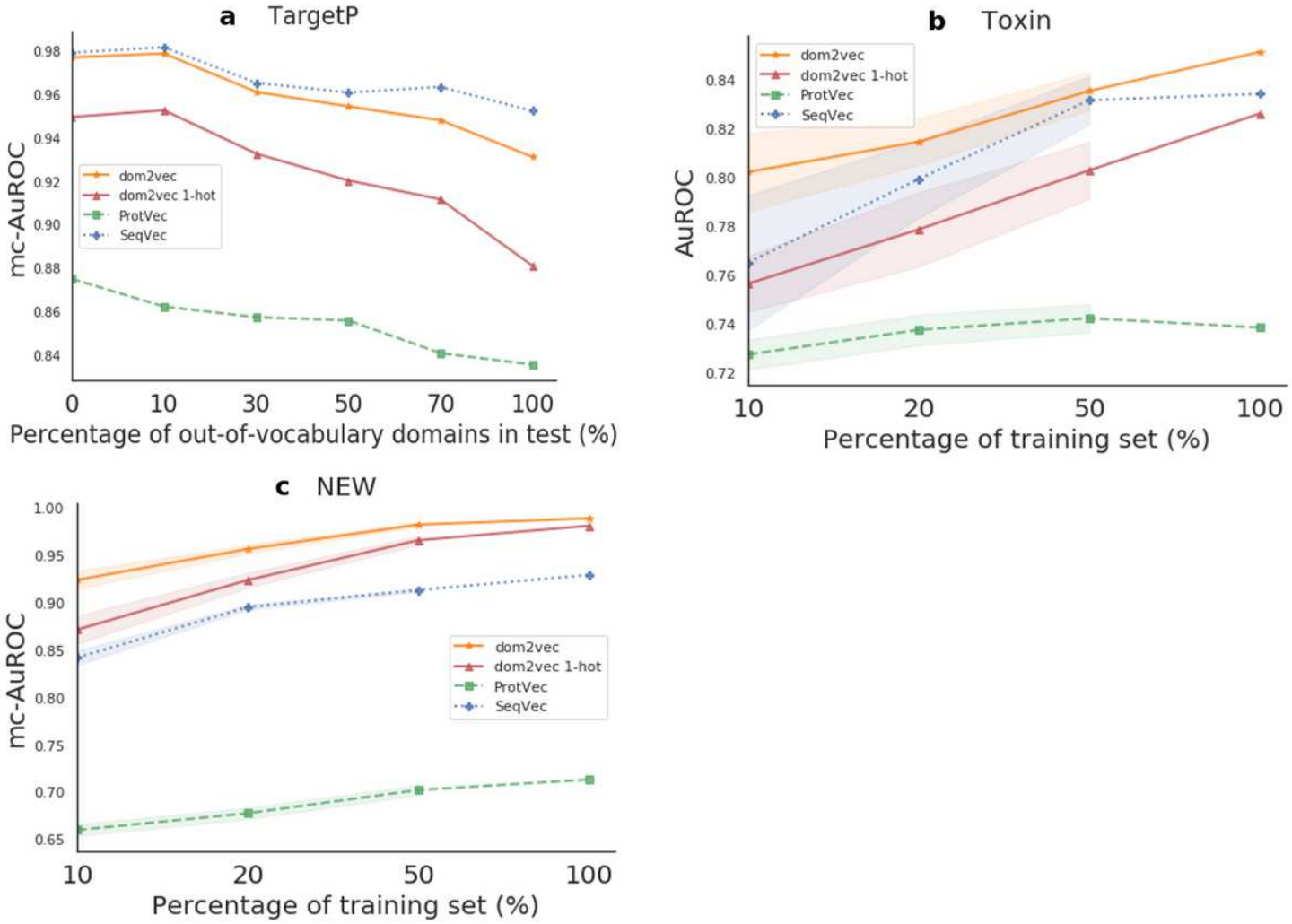

Figure 5

Downstream performance. a: TargetP, OOV experiment: learning in whole train and benchmark in test splits of increasing out-of-vocabulary degree. b&c: Toxin and NEW, generalization experiment: learning in increasing train splits, 10 replicates each, and benchmark in whole test sets. The marked points represent the mean performance on the test set and the shaded regions show one standard deviation above and below the mean.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- dom2vecsupplementary.pdf