

TECHNISCHE UNIVERSITÄT MÜNCHEN

Fakultät für Informatik

Prediction of Protein Function through Machine Learning

Dissertation

Maria Littmann



Fakultät für Informatik

Prediction of Protein Function through Machine Learning

Maria Littmann

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Nils Thuerey

Prüfende/-r der Dissertation:

1. Prof. Dr. Burkhard Rost
2. Prof. Christine Orengo, Ph. D., University College London, UK

Die Dissertation wurde am 17.02.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 25.05.2021 angenommen.



*To my late grandfather, Dr. Carl Preißer*



# Abstract

Knowledge about a protein's function is crucial for understanding the molecular mechanisms of life. However, function is not a well-defined concept but is determined by various factors and can be described through different annotations. Despite its importance, protein function remains unknown for most protein sequences. Computational methods bridge this sequence-annotation gap typically through homology-based inference by transferring annotations from sequence-similar proteins with known function or through prediction methods based on Machine Learning (ML).

This dissertation focuses on developing prediction methods for three aspects of protein function: Gene Ontology (GO) terms, binding residues, and sub-nuclear localization. Applying supervised learning and using manually selected features derived from evolutionary information, we succeeded in predicting sub-nuclear compartments and binding residues of proteins with high prediction performance.

In subsequent work, due to the advances in Deep Learning, we replaced hand-crafted features with data-driven inputs, namely sequence embeddings derived from language models. We used these embeddings to predict binding residues (transfer learning) improving the performance compared to the previous method based on evolutionary information. GO terms were predicted through annotation transfer following a concept similar to homology-based inference but using similarity between embeddings instead of sequence similarity to identify evolutionary related proteins. This new embedding-based annotation transfer clearly outperformed homology-based inference.

All methods developed in this dissertation provide reliable predictions of protein function, rely solely on sequence information, and are more broadly applicable than other methods. In addition, they demonstrate the benefit of combining various ML concepts (supervised learning, transfer learning, unsupervised learning) with biological data, highlighting the large potential of ML not only for protein function prediction but for computational biology in general.





# Zusammenfassung

Das Wissen über die Funktion von Proteinen ist wichtig, um die molekularen Mechanismen des Lebens zu verstehen. Proteinfunktion ist aber kein gut definiertes Konzept, sondern wird von mehreren Faktoren bestimmt und kann durch verschiedene Annotationen beschrieben werden. Trotz ihrer Wichtigkeit ist die Funktion der meisten Proteinsequenzen nicht bekannt. Computerbasierte Methoden schließen diese Lücke zwischen Sequenzen und Annotationen typischerweise durch Inferenz basierend auf Homologie, bei der Annotationen von Proteinen mit ähnlicher Sequenz und bekannter Funktion übertragen werden, oder durch Vorhersagemethoden basierend auf maschinellem Lernen (ML).

Diese Dissertation befasst sich mit der Entwicklung von Vorhersagemethoden für drei Aspekte von Proteinfunktion: Gene Ontology (GO) Terme, Bindungsstellen und Lokalisation von Proteinen im Zellkern. Durch die Nutzung von überwachtem Lernen (engl. Supervised Learning) und manuell ausgewählten Proteineigenschaften (engl. Features), die auf evolutionärer Information beruhen, haben wir erfolgreich und mit guter Qualität die Lokalisation von Proteinen im Zellkern und Bindungsstellen vorhergesagt.

In der nachfolgenden Arbeit konnten wir dank der Fortschritte im Deep Learning manuell bestimmte Eigenschaften durch datenbasierte Inputs ersetzen, und zwar durch Sequenzembeddings, die aus Sprachmodellen gewonnen werden können. Wir haben diese Embeddings genutzt, um Bindungsstellen vorherzusagen (Transfer Learning) und konnten die Vorhersagekraft im Vergleich zur vorhergehenden Methode, die evolutionäre Information genutzt hatte, klar verbessern. GO Terme wurden durch einen Annotationstransfer vorhergesagt, der einem ähnlichen Konzept wie Inferenz basierend auf Homologie folgt, allerdings die Ähnlichkeit zwischen Embeddings statt zwischen Sequenzen nutzt, um evolutionär verwandte Proteine zu finden. Dieser neue embedding-basierte Annotationstransfer liefert deutlich bessere Vorhersagen als Inferenz basierend auf Homologie.

Alle Methoden, die in dieser Dissertation entwickelt wurden, erlauben verlässliche Vorhersagen von Proteinfunktionen, nutzen ausschließlich Sequenzinformation und sind breiter anwendbar als andere Methoden. Außerdem zeigen sie auf, dass die Kombination von verschiedenen ML Konzepten (Supervised Learning, Transfer Learning, Unsupervised Learning) mit biologischen Daten Vorteile bringt. Das wiederum zeigt, dass ML nicht nur ein großes Potential für die Vorhersage von Proteinfunktion hat, sondern für die Bioinformatik im Allgemeinen.



# Acknowledgements

First and foremost, I would like to thank Burkhard Rost for inviting me into his lab and for giving me the opportunity to pursue a PhD. I am very grateful for his ongoing support and all the long walks and scientific discussions. The experience I gained during my time at the Rostlab made me grow both as a scientist and as a person. Also thank you for introducing me to TEDxTUM. Being involved in its organization team for the past years has been one of the best experiences of my life.

I wish to thank Christine Orengo who I had the pleasure to work with during the last years and who invited me to join her lab for a couple of weeks in 2017. Thanks to her and the entire Orengo Lab for welcoming me with open arms. Particular thanks to Nicola Bordin for working with me on the project on increasing functional purity of FunFams which resulted in a manuscript that we just submitted for peer review.

Also, thanks to Christine Orengo and Nils Thuerey for dedicating their time as members of my thesis committee.

I would also like to thank all of my colleagues from the Rostlab for many great scientific discussions and for creating a nice working environment in general. Special thanks go to Inga Weise and Tim Karl for their support with various administrative and technical matters throughout the years. Thanks to Tobias Olenyi, Michael Heinzinger, and Christian Dallago for their feedback on my dissertation.

Thanks to Katharina Selig for turning the vague result of a brainstorming session into a great publication. I really enjoyed working together.

I would like to thank all the awesome people I met during my time at TEDxTUM. Their dedication and passion for a job they do not get paid for has always inspired me. I am grateful for being part of such an amazing team and for all the friends I made along the way.

Thanks to all my friends and my family for always supporting me and being there whenever I needed some distraction from work. Special thanks to my mother who has been there for me my entire life, always accepted my personal goals, and always supported me in achieving those. I would not be where I am today without her.

Last, but not least, thanks to my best friend, my better half, my husband Richard Littmann for helping me with completing this thesis in multiple ways, for always having my back, and for making me a better person.



# Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>ix</b>
<b>Abbreviations</b>	<b>xi</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Annotation of Protein Function . . . . .	1
1.1.1. Describing Protein Function through Gene Ontology Terms . . . . .	3
1.1.2. Protein Function Determined through Ligand Binding . . . . .	6
1.1.3. Influence of Localization on Protein Function . . . . .	9
1.2. Concepts for Protein Function Prediction . . . . .	11
1.2.1. Homology-based Inference . . . . .	11
1.2.2. Machine Learning (ML) . . . . .	13
1.3. Pre-existing Methods to Predict Protein Function . . . . .	20
1.3.1. Prediction of GO Terms . . . . .	20
1.3.2. Prediction of Binding Residues . . . . .	21
1.3.3. Prediction of Subcellular Localization . . . . .	23
1.4. Outline of This Work . . . . .	24
<b>2. Validity of Machine Learning in Life Sciences Increased through Collaborations</b>	<b>27</b>
2.1. Preface . . . . .	27
2.2. Journal Article: Littmann, Selig <i>et al.</i> , Nature Machine Intelligence (2020)	28
2.3. Supplementary Material: Littmann, Selig <i>et al.</i> , Nature Machine Intelligence (2020) . . . . .	36
<b>3. Embeddings from Deep Learning Transfer GO Annotations beyond Homology</b>	<b>57</b>
3.1. Preface . . . . .	57
3.2. Journal Article: Littmann, Heinzinger <i>et al.</i> , Scientific Reports (2021) . . . . .	58
3.3. Supplementary Material: Littmann, Heinzinger <i>et al.</i> , Scientific Reports (2021) . . . . .	73
<b>4. Prediction of Protein Binding Residues from Sequence</b>	<b>99</b>
4.1. Evolutionary Couplings and Sequence Variation Effect Predict Protein Binding Sites . . . . .	99
4.1.1. Preface . . . . .	99

4.1.2. Journal Article: Schelling <i>et al.</i> , Proteins: Structure, Function, and Bioinformatics (2018) . . . . .	100
4.1.3. Supplementary Material: Schelling <i>et al.</i> , Proteins: Structure, Function, and Bioinformatics (2018) . . . . .	112
4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types . . . . .	130
4.2.1. Material and Methods . . . . .	130
4.2.2. Preliminary Results . . . . .	134
4.2.3. Conclusion . . . . .	143
<b>5. Detailed Prediction of Protein Sub-nuclear Localization</b>	<b>147</b>
5.1. Preface . . . . .	147
5.2. Journal Article: Littmann, Goldberg <i>et al.</i> , BMC Bioinformatics (2019) .	148
5.3. Correction to: Littmann, Goldberg <i>et al.</i> , BMC Bioinformatics (2019) . .	164
5.4. Supplementary Material: Littmann, Goldberg <i>et al.</i> , BMC Bioinformatics (2019) . . . . .	167
<b>6. Conclusion</b>	<b>183</b>
<b>References</b>	<b>187</b>
<b>A. Appendix</b>	<b>197</b>

## List of Figures

1.1. Example of relationships in GO for term “oxygen carrier activity” . . . . .	4
1.2. CAFA3 timeline . . . . .	6
1.3. Homology-based inference . . . . .	12
1.4. One-hot encoding . . . . .	14
1.5. Evolutionary profiles . . . . .	16
1.6. Embedding extraction using SeqVec . . . . .	19
4.1. Performance for bindPredictDL-binary . . . . .	135
4.2. Performance for bindPredictDL-multi . . . . .	138
4.3. Prediction performance for different thresholds . . . . .	140
4.4. Performance of homology-based inference for different E-value thresholds	142
4.5. Performance for bindPredictDL-multi combined with homology-based in- ference . . . . .	144

## List of Tables

1.1. Entries in BioLipP database . . . . .	8
4.1. Development set for bindPredictDL . . . . .	131
4.2. Test set performance by ligand type . . . . .	136
4.3. Performance comparison with bindPredictML17 . . . . .	137
4.4. Coverage and Negative Coverage for bindPredictDL-multi . . . . .	139
4.5. Test set performance with and without homology-based inference . . . . .	143





## Abbreviations

<b>ANN</b>	Artificial Neural Network
<b>BPO</b>	Biological Process Ontology
<b>CAFA</b>	The Critical Assessment of protein Function Annotation algorithms
<b>CCO</b>	Cellular Component Ontology
<b>CNN</b>	Convolutional Neural Network
<b>GO</b>	The Gene Ontology
<b>LM</b>	Language Model
<b>LSTM</b>	Long-Short-Term Memory Cell
<b>MCC</b>	Matthews Correlation Coefficient
<b>MFO</b>	Molecular Function Ontology
<b>ML</b>	Machine Learning
<b>MSA</b>	Multiple Sequence Alignment
<b>PDB</b>	Protein Data Bank
<b>PSSM</b>	Position-Specific Scoring Matrix
<b>SVM</b>	Support Vector Machine



# 1. Introduction

Proteins are involved in almost all cellular processes such as metabolic and information pathways, transport, DNA replication and transcription, and organization of structure [1]. Incorrect functioning of certain proteins can highly affect the organism. Therefore, determining the function of proteins is crucial to obtain insights into the molecular mechanisms of life. Knowing the function of a protein fosters the understanding of its overall role in the organism and helps to understand how a mutation in this protein might affect the fitness of the organism.

However, protein function is not a well-defined concept. Proteins can execute multiple functions and their functionality highly depends on the possibility to bind to other molecules called *ligands* [2]. Furthermore, a protein's function is influenced by its environment. Which function a protein executes depends on its localization in the cell, its interaction with other proteins, and conformational changes through post-translational modifications, among other factors [2]. To fully understand a protein's function, it is important to gain knowledge about all aspects influencing function.

In theory, protein function can be determined through specific experiments or computational methods. Though, in reality, it is often difficult to determine all the different facets of protein function and, especially, the complex interplay between them. Thus, we are usually restricted to only considering a few key aspects of a protein's function. This dissertation focuses on the development of computational methods using available experimental annotations to allow predictions of certain functional aspects for proteins without known annotations.

## 1.1. Annotation of Protein Function

The variety and flexibility of possible protein functions depending on multiple environmental factors make it difficult to annotate one specific function to a protein. Therefore,

diverse annotations, which allow to assign function on different interdependent levels, are available [3]. Considering the biological process or pathway a protein is involved in provides a functional assignment on a broader level. Annotations of a protein's subcellular localization describe the compartment or compartments of a cell in which the protein acts. A more specific assignment of protein function considers function on a molecular level, e.g., by determining the reaction catalyzed by a certain enzyme [3].

These three levels of annotations only consider function as a per-protein feature, i.e., one function is assigned to the entire protein. However, function can also be assigned to individual residues by identifying residues which are, for example, involved in binding to other molecules or are important for stabilizing the overall structure of the protein. Information about protein function on a per-residue level allows to determine those residues that are most essential for correct functioning of a protein and can reveal, for example, which mutations are most likely to disrupt protein function and consequently affect the overall fitness of the organism.

Functional annotations have often been made available as free text with a large spectrum of terminology and synonyms [3]. Such unstandardized and ambiguous annotations make it challenging to compare functions between different proteins, automatically analyze the available data, and to develop methods to predict protein function. While text-mining resources allow automatic retrieval of information even from unstructured text [4], especially the structuring and standardization of functional annotations facilitate computational processing of available data. Many excellent sources providing standardized functional annotations exist. They focus on different aspects of protein function considering function both as a protein-wide and a per-residue feature and include ENZYME [5], Swiss-Prot [6], KEGG [7], GO [8, 9], PDB [10, 11], and BioLiP [12, 13], among others [14–18].

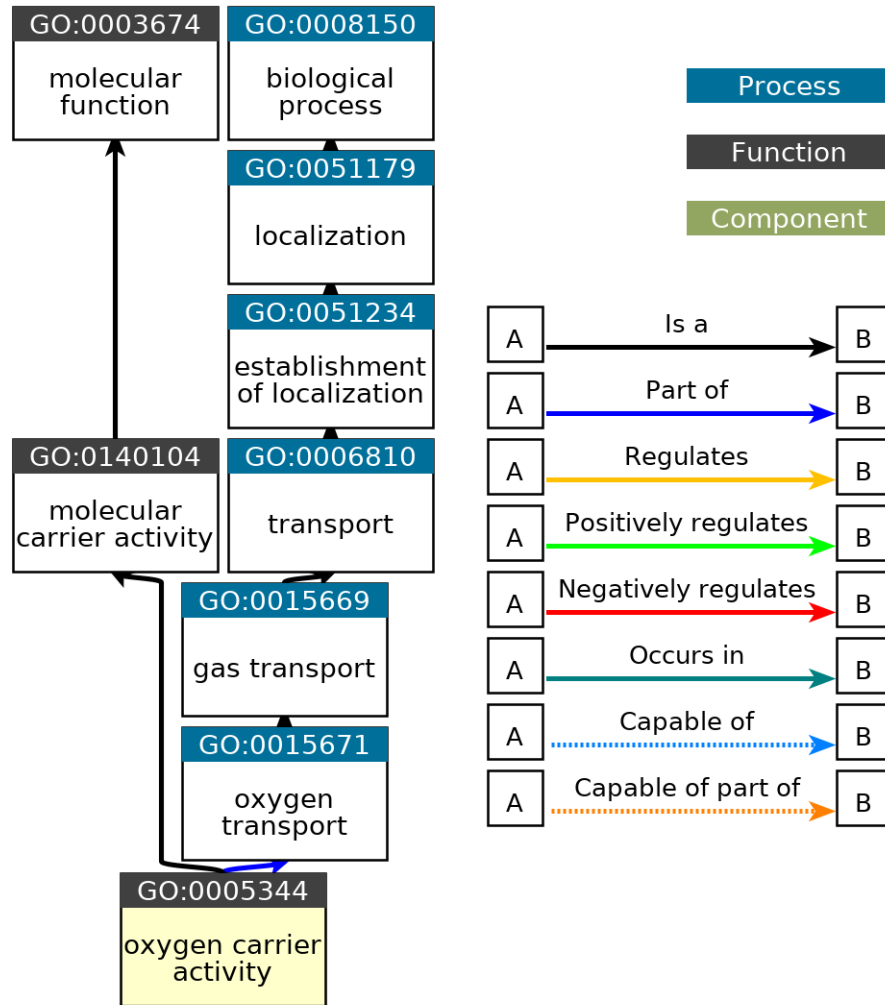
From this variety of available functional annotations, this dissertation focuses on three aspects: Gene Ontology (GO) terms, ligand binding, and sub-nuclear localization. How these aspects are determined and formally annotated is outlined in more detail below.

### 1.1.1. Describing Protein Function through Gene Ontology Terms

#### The Gene Ontology (GO)

GO [8, 9] is one of the most comprehensive resources and provides a structured and controlled vocabulary to describe protein function in a human- and machine-readable format. This allows easy manual annotation of protein function as well as computational processing of proteins and their annotations [8, 9]. GO separates different aspects of function into three hierarchies: (1) the Molecular Function Ontology (MFO) describes protein activity on the molecular level, (2) the Biological Process Ontology (BPO) focuses on the larger processes and pathways in a cell involving multiple molecular activities, and (3) the Cellular Component Ontology (CCO) refers to the cellular component(s) or subcellular localization(s) in which the protein acts. These three ontologies are organized as directed acyclic graphs with each node representing a functional annotation called a *GO term*. GO terms consist of a standardized name describing a certain function (e.g., “protein binding”) and a unique identifier (e.g., GO:0005515). GO terms are connected through “is a” relationships. For a certain relationship “*X* is a *Y*”, *X* is a more specific description of *Y* and is considered a child term of *Y* with the least specific term in each ontology being the corresponding root (molecular function, biological process, cellular component). However, GO is only loosely hierarchical because a term may have more than one parent term [9]. Other relations like “regulates”, “occurs in”, or “part of” do not follow a hierarchical structure but allow further specifications of functional connections and even relationships between terms from different ontologies. For example, the term “oxygen carrier activity” (GO:0005344) is a molecular carrier activity (GO:0140104) which is a molecular function (GO:0003674), i.e., this term belongs to MFO. However, it is also “part of” oxygen transport where “oxygen transport” (GO:0015671) belongs to BPO (Fig. 1.1).

A protein can be annotated to multiple functions by assigning a set of GO terms to it. The structure of the ontologies allows functional annotation with varying degree of specificity depending on, for example, the experimental evidence for a certain function. For a particular term annotated to a protein, all of its less specific parent terms are implicitly also assigned to this protein. In general, obtaining GO annotations through experimental or computational methods is difficult due to the large number of available terms with less specific terms usually being easier to annotate while providing less information than more specific terms. Therefore, many, particularly very specific GO terms



QuickGO - <https://www.ebi.ac.uk/QuickGO>

**Figure 1.1.: Example of relationships in GO for term “oxygen carrier activity”.** The GO term “oxygen carrier activity” (GO:0005344) belongs to MFO but is also connected to the term “oxygen transport” (GO:0015671), which is part of BPO. Terms in boxes with a black heading belong to MFO, terms in boxes with a blue heading to BPO. Black arrows indicate “is a” relationships while blue arrows depict “part of” relationships. The visualization has been retrieved from QuickGO [19].

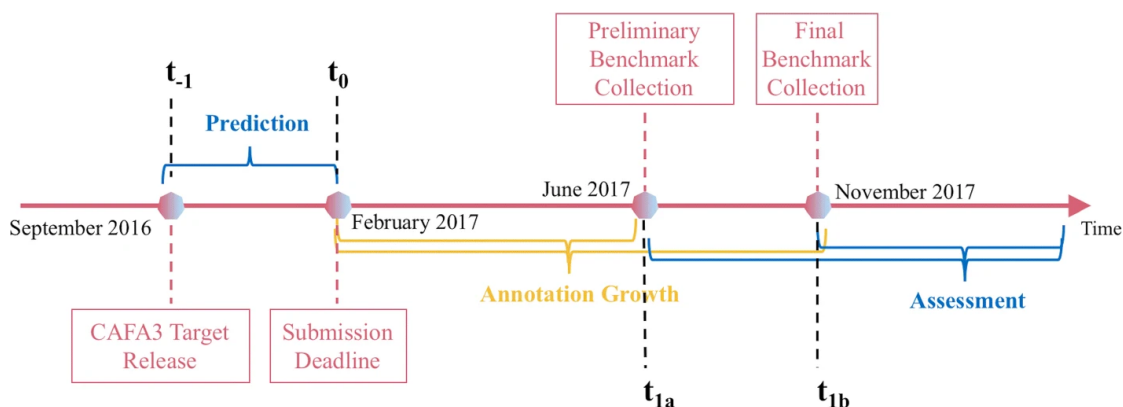
are only annotated to very few proteins [20]. Computational annotation of GO terms is further complicated by the hierarchical structure of the GO and the relationships between terms. Computational methods that treat terms independently of each other ignore that, if a certain term is annotated to a protein, other terms might implicitly also be annotated (e.g., parent terms) or cannot be annotated to this protein (e.g., mutually exclusive cellular compartments) [20].

The Gene Ontology Annotation database (GOA) [21–23] provides high-quality GO annotations for protein sequences in UniProt [24]. It contains manual annotations as well as computationally assigned GO terms. An evidence code indicates the annotation type. GOA is updated approximately every four weeks and version 201, released on 03 December 2020, contained GO annotations for 300 087 protein sequences from Swiss-Prot [6], of which 71 367 (24%) proteins had experimentally verified annotations (with evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC). Therefore, for 53% of all sequences in Swiss-Prot, any GO annotation is known, while experimentally verified annotations are only available for 13% of all proteins.

### **The Critical Assessment of protein Function Annotation algorithms (CAFA)**

To bridge the gap between many known sequences but only few with known GO annotations, multiple computational methods have been developed to determine protein function by predicting GO terms (more information in Section 1.3.1). The growing number of methods necessitated comparable performance evaluations. The Critical Assessment of protein Function Annotation algorithms (CAFA) [25–27] is an international collaboration designed to assess computational methods predicting GO terms, using a time challenge. The CAFA challenge takes place roughly every two to three years with the fourth instance (CAFA4) currently being evaluated. It follows a specific timeline and setup to allow an independent and fair evaluation of the prediction methods. Initially, at time point  $t_{-1}$ , a set of prediction targets is released (Fig. 1.2). Those targets usually comprise around 100 000 protein sequences which lack GO annotations for at least one of the three ontologies (BPO, MFO, CCO). CAFA participants set out to predict GO terms for those targets until time point  $t_0$  (Fig. 1.2). If any annotations are known in one ontology, predictions for that ontology are not considered in the evaluation. After the prediction phase, annotations for some of the targets are collected over the course of approximately nine months until time point  $t_{1b}$  (Fig. 1.2). Those annotations are obtained due to the natural annotation growth in public databases and, therefore, the

number of targets used for evaluation can vary between different instances of CAFA. The number of sequences with experimentally verified annotations after this phase is usually much smaller than the number of released targets. For example, for CAFA3, only 2.5% of the released targets gained annotations until the final assessment [27]. Using those newly obtained annotations, all participating methods are assessed applying the same evaluation criteria. The time-separated setup of the CAFA challenge ensures that the evaluation only relies on functional annotations which were not available during the development of any of the assessed methods. Even outside of the assessment phase of CAFA, the CAFA targets and evaluation are usually considered the standard for assessing performance of a method predicting GO terms.



**Figure 1.2.: CAFA3 timeline.** In the prediction phase between  $t_{-1}$  and  $t_0$ , participants submit predictions for the released targets. During the phase of annotation growth, the CAFA organizers collect newly gained annotations for the targets. Based on this final benchmark set, all participating methods are assessed and their performance is compared. This figure is a reprint from [27] and has been published under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

### 1.1.2. Protein Function Determined through Ligand Binding

Protein function is largely determined through the binding of proteins to specific ligands [2]. For example, proteins can bind to certain molecules to mark them for destruction or to catalyze a reaction between them. Proteins also bind to each other to assemble



into larger complexes, and the binding of certain molecules can activate or deactivate the execution of a certain protein function [2].

Proteins can bind various different ligands like metal ions (e.g.,  $\text{Ca}^{2+}$  or  $\text{Zn}^{2+}$ ), inorganic molecules (e.g.,  $\text{SO}_4^{2-}$  or  $\text{PO}_4^{3-}$ ), organic molecules (e.g., ATP or NAD), or macromolecules such as DNA, RNA, and other proteins. The biophysical properties of these ligands vary highly. For example, metal ions are usually positively charged while acid radicals like  $\text{SO}_4^{2-}$  or  $\text{PO}_4^{3-}$  are negatively charged. Also, metal ions and other small molecules are much smaller than DNA, RNA, or other proteins. Therefore, depending on the bound ligand, binding sites differ as well. More residues are involved in binding macromolecules like DNA or RNA than in binding small molecules. Positively charged ligands bind to negatively charged amino acids in the protein, while the binding site for negatively charged ligands rather contains positively charged amino acids [2]. Therefore, which residues in a protein can potentially be involved in binding depends on the specific ligand bound to the protein. The number of potential binding residues increases even further if other proteins are considered as possible ligands. Proteins highly vary in size, and biophysical properties like charge or hydrophobicity do not only differ between proteins, but also differ between different regions of one protein. Thus, a protein-protein binding site is not always formed by a similar structure which makes it hard to identify a consistent pattern.

The differences in size and biophysical properties of binding sites make protein binding a process of high specificity as one protein commonly binds to one or only a few selected ligands. Often only a few key residues in the protein determine this specificity. Those binding residues are experimentally determined by solving structures in complex with the respective ligand and identifying residues in close proximity to this ligand as binding residues (e.g.,  $\leq 5\text{\AA}$ ) [28]. In addition, catalytic residues can be identified as those residues in enzymes that are directly involved in the catalytic mechanism [29].

Protein-ligand complexes are available through the Protein Data Bank (PDB) [10, 11] and high-resolution structures (e.g., with a resolution  $\leq 2.5\text{\AA}$ ) obtained through X-ray crystallography [30] serve as a good starting point to obtain high-quality annotations of binding residues. However, many additives are used during protein purification and crystallization. Those additives appear as ligands bound to the protein in the crystallized structure, although they are not biologically relevant and do not bind to the protein under natural conditions. Evaluating which ligands are in fact biologically relevant is not trivial. Many resources have tried to address this issue [31–35]. BioLiP [12, 13], a

## 1. Introduction

---

database of biologically relevant ligand-protein interactions collected from the PDB, is one of those resources. The database is created using a combination of computational and manual assessment of ligand binding allowing a fast and high-quality determination of the biological relevance of ligands. Each entry in BioLiP provides a large list of annotations including not only the residues in a PDB structure involved in binding but also the bound molecule, ligand-binding affinity, catalytic residues obtained from the Catalytic Site Atlas (CSA) [29], Enzyme Commission (EC) numbers [36] describing the catalyzed reaction, GO terms, and cross-links to other popular databases [12]. BioLiP contains information on ligand binding for small (regular) ligands, metal ions, peptides, and DNA and RNA (combined as nucleic acids). It is updated weekly and the version released on 08 January 2021 contained data for 29 509 different ligands and 109 206 PDB structures (Table 1.1). The most frequently occurring ligands are nucleic acids, zinc ions, calcium ions, magnesium ions, and peptides accounting for 38% of the binding annotation data while 12 183 (41%) ligands are only annotated to one protein.

Number of	
Entries	525 197
PDB structures	109 206
DNA/RNA ligands	56 807
Peptide ligands	25 835
Metal ligands	146 126
Small (regular) ligands	296 421
Entries with binding affinity data	23 492

**Table 1.1.: Entries in BioLiP database.** BioLiP is updated weekly. In the version released on 08 January 2021, it contained binding information for 109 206 PDB structures. Most information was available for small ligands with 296 421 entries. Binding affinity data was available for 23 492 entries.

In addition to the presence of additives in protein-ligand complexes, the *cognate* ligand, i.e., the ligand binding to an enzyme in nature, can often not bind to the enzyme without the catalyzed reaction occurring. Therefore, compounds with a certain similarity to the cognate ligand are used as surrogates for crystallization of protein-ligand complexes. However, Tyzack et al. showed that the bound ligand is often not very similar to the cognate ligand with only 26.0% of all enzymatic structures in the PDB bound to a ligand with a similarity of  $\geq 0.7$  to the cognate ligand [37]. Therefore, considering the similarity between bound and cognate ligand in addition to the resolution of the structure can help

in identifying protein-ligand complexes that provide high-quality annotations of binding residues in enzymes.

Even with resources like BioLiP and information on similarity between cognate and bound ligand available, assessing the correctness of binding annotations remains an open issue. Binding annotations are not available for all proteins, and even if binding residues are known for a certain protein, other binding annotations can still be missing. Also, which residues are considered as binding is not well-defined. While defining all residues as binding which are close to the ligand is a commonly used approach, this definition still relies on a pre-defined threshold. Small variations of this threshold or of the overall protein structure can highly affect which residues are considered close to the ligand and, therefore, binding. These issues make it, in general, difficult to fully understand binding and protein function on a per-residue level. Computational methods to predict binding residues can assist experimental annotation by hinting towards potential binding sites. However high-quality prediction methods are difficult to develop because only a few residues in a protein are usually involved in binding (around 3% for metal ions to 15% for nucleic acids).

### 1.1.3. Influence of Localization on Protein Function

Proteins with a similar function often co-localize [38–40], and many proteins only execute their correct function in one particular compartment of the cell [41, 42]. Therefore, knowing the subcellular localization of a protein is one important aspect to describe its function.

Prokaryotic cells surround only one single compartment with a plasma membrane (with gram-negative bacteria having an additional outer membrane), while eukaryotic cells are organized into several membrane-bound compartments. The major intracellular compartments of an animal cell are the cytosol, endoplasmic reticulum, Golgi apparatus, nucleus, mitochondrion, lysosome, and peroxisome [2]. Plant cells have chloroplasts and vacuole as additional compartments. The compartments are separated from the extracellular space by the plasma membrane. More fine-grained distinctions between these compartments are also possible; e.g., we can distinguish between rough and smooth endoplasmic reticulum [2]. Additionally, some of the compartments are divided into sub-compartments. For example, the nucleus consists of the nucleolus, nucleoplasm, and the nuclear membrane [18].

Proteins are translated in the ribosomes located in the cytosol or attached to the endoplasmic reticulum and have to be transported to their final compartment afterwards [43]. To a large extent, they are sorted based on sorting signals, i.e., specific short sequences which are part of the whole protein sequence. For example, signal peptides are the targeting signal for the secretory pathway, i.e., the transport of proteins to the cell membrane or the extracellular space, and are usually located at the N-terminus of the protein sequence [44]. Transport into and out of the nucleus is mediated through nuclear localization signals (NLS; for import) and nuclear export signals (NES; for export), which are sorting signals that can occur anywhere in the protein sequence [45]. There exist different databases providing sorting signals associated with different compartments. For example, LocSigDB [43] collects experimentally known sorting signals for eight distinct subcellular localizations. NLSdb [46, 47] is a database for NLS and NES combining experimentally annotated signals and signals identified through a simple *in silico* mutagenesis.

Annotations for subcellular localization are, for example, available as part of GO through CCO [8, 9]. In total, CCO consists of 4 185 terms [48] with many terms describing very specific localizations but being annotated to none or very few protein sequences. Also, annotations in CCO seem to be incomplete. For example, only 0.2% of human proteins have an experimentally verified annotation in CCO. Other, probably more comprehensive resources like Swiss-Prot [6] and HPA [18] currently contain experimentally verified annotations of subcellular localization for 36% (with evidence code ECO:0000269 [49]) and 52% (“supported” and “approved” localizations) of the human proteome, respectively. These databases focus on a limited set of compartments and experimentally verified annotations of localization seem to be prevalently stored in those databases but are not reflected in CCO. Furthermore, specialized resources zoom into the localization in specific cellular compartments. For example, NMPdb [50], NOPdb [51], and NSort/DB [52] focus on more fine-grained localization annotations of nuclear proteins.

In general, determining localization on a more coarse-grained level is usually simpler than defining it on a more detailed level. Consequently, most experimentally verified annotations are available for compartments which can be easily determined in experiments with the most annotations usually being available for extracellular space and nucleus while less prominent compartments include peroxisome, lysosome, or vacuole [53, 54]. Prediction methods can help to infer annotations where experimental data are not available, but these methods tend to experience the same bias and are usually better

in predicting more frequently annotated compartments, while not providing predictions for less well-studied compartments or more fine-grained substructures.

## 1.2. Concepts for Protein Function Prediction

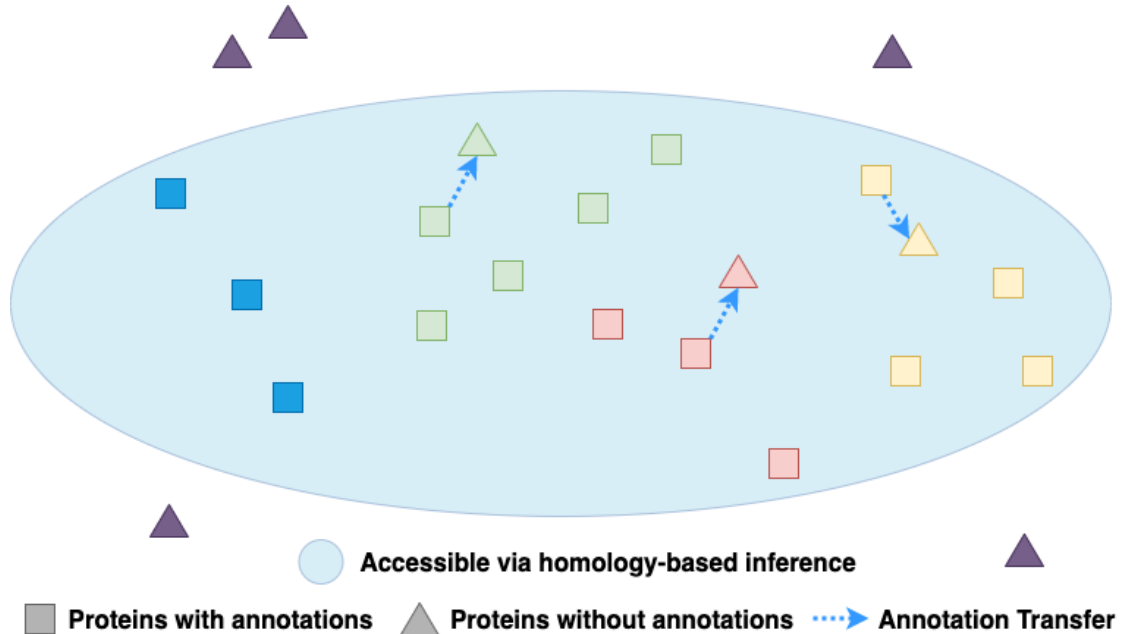
Advances in sequencing during the last years reducing costs and increasing speed have led to an exponential growth of publicly available protein sequences with more than 500 000 manually curated protein sequences currently available in Swiss-Prot [6]. However, most of these sequences lack functional annotations causing a substantial *sequence-annotation gap*. Computational biology has been trying to bridge this gap by developing prediction methods that allow fast and high-quality prediction of structural or functional protein features. In general, most pre-existing methods as well as the methods developed in this dissertation are based on two different core concepts: (1) Homology-based inference and (2) *de novo* prediction through Machine Learning (ML).

### 1.2.1. Homology-based Inference

In general, homology describes a similarity due to shared ancestry. In terms of proteins, we expect two homologous proteins to share a similar structure and function [55].

Homology-based inference is based on the assumption that sequence similarity between two proteins mostly stems from evolutionary relation and, therefore, sequence-similar proteins should share, for example, a common function [56]. Based on this relationship, homology-based inference transfers annotations between sequence-similar proteins. More specifically, for a given query protein  $Q$  without known functional annotations, the most sequence-similar protein  $H$  is identified from a set of proteins with known annotations. Then, the annotations from  $H$  are transferred to  $Q$  (Fig. 1.3, annotation transfer is indicated as blue arrows). Pairs of similar proteins can be found by using either simple sequence-to-sequence comparisons applying algorithms like BLAST [57] or more sophisticated sequence-to-profile comparisons through, e.g., PSI-BLAST [58] or MMseqs2 [59], which allow the identification of more distantly related proteins. The best hit for a given query protein can be simply identified as the protein with the highest sequence identity or sequence similarity to the query. However, taking aspects like the sequence length and the likelihood for the observed similarity to occur by chance into account usually

allows better distinction of protein hits which actually indicate evolutionary relation from random hits.



**Figure 1.3.: Homology-based inference.** This sketch shows the principal idea of homology-based inference. Triangles represent proteins without known annotations while squares indicate proteins with annotations. For some of the unannotated proteins, proteins with similar sequences and known annotations can be found, and annotations can be transferred between them as depicted by blue arrows. However, for some proteins indicated as purple triangles, no sequence-similar protein with known annotations exists in the data set. Therefore, no predictions can be made for those proteins using homology-based inference.

Homology-based inference has proven to work very well for the prediction of protein function as long as an annotated protein with high enough sequence similarity can be found [60–62]. There also exist evolutionary related protein pairs with low sequence similarity that still share a common function. However, if the similarity between two protein sequences is low, it is more likely that this similarity is observed by chance and not because the proteins descend from a common ancestor. Therefore, we cannot safely transfer annotations between such pairs anymore.

In addition, algorithms to perform sequence comparison rely on multiple parameters. Changing those parameters can affect which proteins are identified as similar and hence

potentially homologous to a certain query protein. Especially for low similarities, slight changes in the parameters can lead to missing homologous proteins or incorrectly identifying proteins as evolutionary related. For high similarity thresholds allowing a safe transfer of annotations, we cannot identify a sequence-similar protein with annotations for many queries (Fig. 1.3, purple triangles), and most proteins remain unannotated through homology-based inference. In particular for proteins from small, less studied protein families, homology-based inference often fails to provide protein function predictions.

### 1.2.2. Machine Learning (ML)

To predict the function of proteins for which homologs with annotations are not available, we usually rely on *de novo* prediction methods based on ML. ML automatically discovers patterns and regularities in a data set and makes predictions for new data based on these patterns [63, 64]. In the most common variant of supervised learning, an ML model is trained on a set of inputs or features for which the output is known. In the case of protein function prediction, protein structures can be a powerful input for these ML models. However, experimentally determined structures are not available for most proteins, and methods to predict protein structures with high quality and on large scale are not publicly available yet. Therefore, this dissertation focuses on the development of sequence-based methods. These methods solely rely on protein sequences as input, are therefore more broadly applicable than structure-based methods, and allow to obtain predictions for large sets of proteins.

Protein sequences are given as strings of varying length, built from 20 characters. Each character represents one of the 20 native amino acids, and each letter in the string represents one position in the protein sequence. For most ML algorithms, in order to be able to train them on protein sequences, we need to find a numerical encoding for those sequences.

#### One-hot Encoding

The simplest numerical encoding of a protein sequence is the *one-hot encoding* (Fig. 1.4). Here, each residue in the protein is represented as a vector of length 20, and each position in the vector stands for one of the 20 amino acids. For any position in the



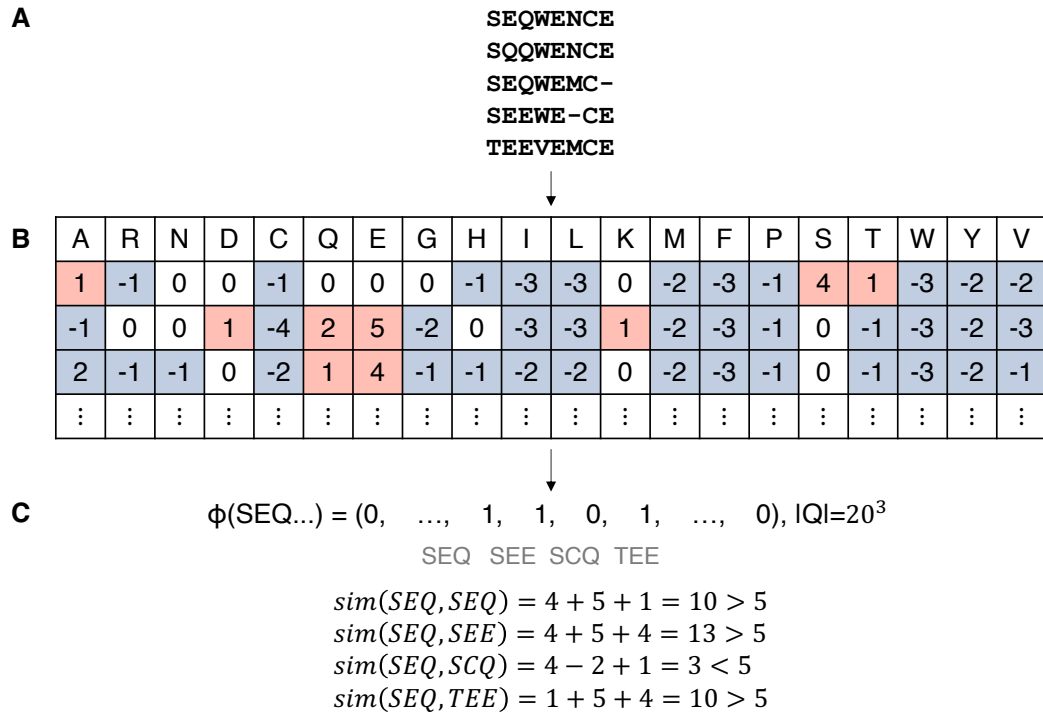


tion. Therefore, if a position is conserved, i.e., it is rarely changed between sequences from one family, it is likely to be structurally or functionally important. On the other hand, more variable positions which are different between proteins are probably less involved in ensuring correct functioning of the protein. Thus, using protein families as input for ML allows to implicitly provide the algorithm with information about sequence conservation and variability of single positions.

Protein families are represented through *evolutionary profiles*. Evolutionary profiles can be constructed using algorithms like PSI-BLAST [58]. They are represented through position-specific scoring matrices (PSSMs) which are built from multiple sequence alignments (MSAs) (Fig. 1.5A). For each position, a PSSM indicates how likely it is that a specific amino acid occurs at this position (Fig. 1.5B). If amino acids receive a positive score, they occur more often than expected while a negative value indicates that this amino acid occurs less frequently than expected. We can either assume an equal distribution for the expected frequency for each amino acid  $a$ , i.e.,  $f_{exp}(a) = \frac{1}{20} \forall a$ , or use more sophisticated distributions that consider, for example, the frequency with which an amino acid occurs in the entire MSA or in sequence sets from large databases.

PSSMs are of size  $L \times 20$  and therefore, have variable length depending on the protein sequence. To convert them into a vector of fixed length, for example, the Profile Kernel [65, 66] can be applied. The Profile Kernel counts the number of occurrences of  $k$ -mers (i.e., short protein sequences of length  $k$ ) and similar  $k$ -mers in the protein sequence resulting in a vector of length  $20^k$  where each element represents one  $k$ -mer (Fig. 1.5C). The value for  $k$  needs to be optimized by the user and, e.g.,  $k = 3$  has achieved best results for subcellular localization prediction [53, 60]. As for the amino acid composition, the Profile Kernel does not encode any positional information. To predict features on a protein level, the Profile Kernel can be used in conjunction with a Support Vector Machine (SVM). SVMs apply the Kernel trick to map input data which are not linearly separable into a higher dimensional feature space and find the optimal linear separation of the data in this new space. Therefore, SVMs rely heavily on the used Kernel function. For proteins, the Profile Kernel can serve as this function.

Evolutionary profiles are a powerful input used for many methods trying to predict all kinds of protein features [53, 62, 67, 68]. With a growing number of well performing prediction methods, we cannot only use the evolutionary profiles directly for predictions, but we can also develop prediction methods indirectly relying on evolutionary profiles by integrating protein features predicted through other methods. For example, the



**Figure 1.5.: Evolutionary profiles.** **A.** Small MSA with five sequences for protein sequence “SEQWENCE”. **B.** Potential resulting PSSM for the first three positions in the alignment. Negative values indicated as blue cells represent amino acids less observed at this position in the MSA than expected; positive values (red cells) represent amino acids more often observed than expected. **C.** The Profile Kernel is a vector of length  $20^k$ . For  $k = 3$ , each element in the vector represents one 3-mer. For a given 3-mer in the sequence (here: “SEQ”), all elements in the vector corresponding to a similar 3-mer are increased by one. Similarity is defined using a conservation threshold. For example, all 3-mers with a score  $> 5$  are considered similar. Then, “SEQ”, “SEE”, and “TEE” are similar to “SEQ” while “SCQ” is not.

web service PredictProtein [68] provides predictions for around 30 different structural and functional protein features and most of them are predicted through methods using evolutionary profiles as input. Then, those features are used as input for methods like SNAP2 [67] or ProNA2020 [62].

While evolutionary profiles and protein features predicted from them serve as input for many methods achieving good prediction performance, evolutionary profiles still rely on the construction of MSAs and the identification of sequence-similar proteins. As for the sequence search forming the basis of homology-based inference (see Section 1.2.1 for more information), building MSAs depends on many parameters which need to be optimized by experts, and for some proteins, it might not be possible to construct a meaningful MSA. Furthermore, using predicted features as input requires manual feature selection by experts since each feature might add beneficial information for different ML predictors.

### Sequence Embeddings

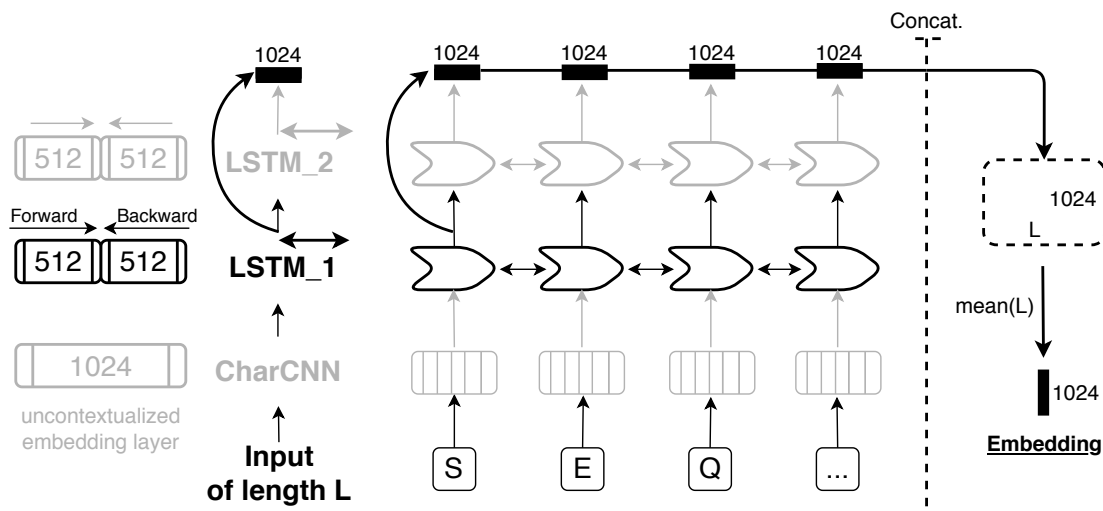
New data-driven techniques remove the need for the calculation of MSAs and manual selection of hand-crafted features. Adapting concepts from Natural Language Processing, new encodings for proteins referred to as *embeddings* are developed. They are derived from large sets of protein sequences without the need of any annotations for those sequences (self-supervised training). For example, ProtVec [69] is based on the concept of word2vec [70]. It defined 3-mers of amino acids as “words” and trained a Skip-gram neural network [70] using sequences from Swiss-Prot [6] to represent 3-mers as vectors of length 100. The length of the vectors is defined by the number of units in the hidden layer of the Skip-gram neural network. Therefore, the number of hidden units and consequently the size of the embeddings is one of the hyperparameters of the model, set prior to training. For ProtVec, a residue  $r_i$  in a sequence can be represented through the 100-dimensional embedding for the 3-mer  $(r_{i-1}, r_i, r_{i+1})$ . For example, in sequence “SEQW...”, the residue E would be represented through the embedding of “SEQ”. To obtain a fixed-length representation for an entire sequence, the vector representations of overlapping 3-mers for this sequence can, for example, be summed up and normalized (mean pooling) resulting in an embedding for the entire protein with 100 dimensions. It has been shown that ProtVec embeddings capture biophysical properties of amino acids and, e.g., 3-grams of hydrophobic amino acids are closer to each other in embedding space than to 3-grams of hydrophilic amino acids [69].

While ProtVec captures some context (the representation of a residue changes for different neighbors), “words”, i.e., 3-mers, are always depicted by the same vector independent of the rest of the sequence. Therefore, ProtVec is considered an uncontextualized approach [69, 70]. Contextualized approaches like SeqVec [71] or ProtBERT-BFD [72]

allow to derive representations for amino acids depending on their context, i.e., the rest of the protein sequence. SeqVec is based on the language model (LM) ELMo [73] (Embeddings from Language Models) using a stack of bi-directional long-short-term-memory cells (LSTMs) [74] and was trained to predict the next amino acid given the entire previous sequence (auto-regressive pre-training) [75, 76]. All sequences in UniRef50 (UniProt [24] clustered at 50% pairwise sequence identity) were used for training resulting in a set of around 33 million proteins. For ProtBERT-BFD, UniRef50 was replaced by BFD [77, 78] which contains around 2.1 billion sequences and is therefore 70 times larger than UniRef50. ProtBERT-BFD follows the idea of BERT [79] (Bidirectional Encoder Representations from Transformers [80]), which processes sequential data through the self-attention mechanism [81], and was trained to reconstruct masked out amino acids in a given protein sequence (masked language modeling). Transformers are better at capturing long-range dependencies than LSTMs because they explicitly compare each input token (amino acid) against all other input tokens (amino acids). While this allows to efficiently propagate information that is spread far apart in the sequence, it scales the memory requirement quadratically with the input length.

For both SeqVec and ProtBERT-BFD, 1024-dimensional embeddings for each residue in a protein sequence can be extracted from the hidden states of the pre-trained models (e.g., using the first LSTM layer of SeqVec, Fig. 1.6). Due to the extensive training time for both SeqVec and ProtBERT-BFD, the default embedding size of 1024 was chosen without further optimization of the number of dimensions. A fixed-size representation for a protein sequence of length  $L$  can be derived by, for example, averaging over all  $L$  residue embeddings (mean pooling) resulting in a 1024-dimensional embedding encoding the entire sequence (Fig. 1.6). Other approaches like maximum pooling or more complex methods [82, 83] can also be applied to derive per-protein embeddings from the per-residue representations. Since SeqVec and ProtBERT-BFD embeddings are contextualized, the per-residue embedding for a certain amino acid  $X$  looks different for two different sequences and also for two different occurrences of  $X$  in the same sequence. Therefore, unlike the Profile Kernel or the amino acid composition, per-protein embeddings encode positional information, and for two proteins with identical amino acid composition but different ordering of those amino acids, the per-protein embeddings will look differently.

Without the need to re-train the LMs for a specific task, the protein embeddings can be used as input to train prediction methods (transfer learning). Although the LMs were



**Figure 1.6.: Embedding extraction using SeqVec.** The example illustrated here shows how the first three residues (SEQ) of a hypothetical protein sequence (“SEQ...”) are processed with SeqVec in order to obtain a fixed-sized embedding. In general, the three layers of SeqVec (uncontextualized: CharCNN; contextualized: LSTM layer 1 and LSTM layer 2) project each residue to a vector space. The two LSTM layers process the sequence in both directions (“Forward” and “Backward”), each creating a vector of size 512. The vectors of both directions are concatenated for each LSTM independently, resulting in an embedding of size 1024 for each LSTM layer. Then, embeddings are extracted for each residue, e.g., as shown here, from the first LSTM layer. This results in 1024-dimensional embeddings for each residue. Per-protein embeddings are obtained by concatenating all per-residue embeddings (resulting in a  $L \times 1024$  matrix) and averaging over the length of the protein  $L$  ( $\text{mean}(L)$ ). This results in a per-protein embedding of length 1024. This figure is a reprint from [84] and has been published under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

never trained on any functional annotations, it has been shown that the embeddings capture rudimentary features of protein structure and function [71, 72, 85, 86]. While prediction methods using embeddings as input cannot compete with state-of-the-art methods yet [71, 72], they offer a fast and easy to compute alternative compared to evolutionary profiles. Future advances in the underlying LMs are presumably going to further improve the resulting embeddings offering the potential of developing easy yet powerful new methods for protein structure and function prediction.

## 1.3. Pre-existing Methods to Predict Protein Function

### 1.3.1. Prediction of GO Terms

Among the many existing methods to predict GO terms, the top performing methods from CAFA3 [27] being considered current state-of-the-art. For all three ontologies (BPO, MFO, CCO), the best performing method in CAFA3 was GOLabeler [87]. It combines five different components complementing each other: a naïve approach based on the frequency of GO terms in a database, homology-based inference, and three methods based on logistic regression using amino acid trigram, domains and motifs, and biophysical properties, respectively. The corresponding outputs are passed through a separate ranking model to generate the final list of predicted GO terms [87]. The performance of both the naïve approach and homology-based inference were reported as baseline methods for CAFA3 [27]. While the three logistic regression models require computationally intensive additional feature extraction, the results from CAFA3 as well as the independent performance evaluation from You et al. clearly showed that those three models improve performance upon just using the two baseline approaches. You et al. further extended their method after CAFA3 by adding another component harnessing data from protein-protein interactions resulting in the new method NetGO [88].

Methods published after CAFA3 with promising performance started to apply Deep Learning models. Two examples for such methods are DeepGOPlus [89] and a multi-task deep neural network (MTDNN) architecture developed by Fa et al. [90]. DeepGOPlus builds upon DeepGO [91] which learned sequence features using Convolutional Neural Networks (CNNs) and combined them with a protein-protein interaction network. For DeepGOPlus, Kulmanov and Hoehndorf replaced the original trigram embedding layer to represent the sequence with one-hot encoding, increased the number of CNN layers used, and switched from a hierarchical classification to a flat classification scheme. Kulmanov and Hoehndorf claimed that their method would have been among the top three methods in CAFA3 if they had competed.

MTDNN uses a multi-task architecture to tackle the multi-label problem of predicting GO terms, i.e., one protein can be annotated to multiple GO terms, and predicting whether one specific GO term is annotated to the respective protein or not is considered one task in this context. MTDNN combines a set of feedforward layers that are shared by all tasks with a set of layers specifically trained for each task. This setup allows

the predictor to utilize information from shared representations as well as task-specific characteristics during training. While not requiring time-consuming feature extraction steps, MTDNN could not be trained for each GO term individually due to too high demands for GPU memory. Instead, terms were grouped by their “is a” relationships in GO in order to allow training of single branches followed by subsequent predictions of the descendants in each branch [90].

In general, the best performing methods from CAFA3 have mainly relied on large feature sets and the combination of multiple ML models into one ensemble method [27]. Therefore, obtaining high-quality predictions of GO terms requires complex models, which are time-consuming to train, and input features which are difficult and sometimes even impossible to retrieve. With the rise of Deep Learning, time-consuming feature selection can be replaced. However, the resulting new models are even more complex, require training of many free parameters using only a small set of available data, and also trigger high demands for GPU memory to allow efficient training.

In this dissertation, I propose a new method which is simpler than other state-of-the-art methods while still achieving competitive performance, allowing fast and accurate predictions of GO terms for large sets of proteins (see Chapter 3).

#### 1.3.2. Prediction of Binding Residues

Protein-ligand binding is mainly determined through the three-dimensional structure of the protein. Therefore, structure-based methods have usually outperformed sequence-based methods [12, 92]. The method COACH [92] has been considered the state-of-the-art method to predict binding residues for many years [93–95]. It is an ensemble classifier combining five individual approaches. COACH was built upon two main components: the template-based methods TM-SITE and S-SITE. TM-SITE predicts binding residues by using structures of proteins with known annotations that are similar to the query protein, while S-SITE facilitates sequence profile-profile comparisons. COACH was further extended to also integrate COFACTOR [96], FINDSITE [97], and ConCavity [98], three previously developed prediction methods utilizing different structural features to predict binding residues.

While COACH is a powerful method, it highly relies on protein structures and the availability of templates, i.e., homologous proteins with known binding sites similar to the query protein. However, structures are not available for all sequences, and homologs

cannot be identified for each protein. Therefore, sequence-based *de novo* methods are needed to allow predictions for all available protein sequences. To not lose the good performance of structure- and template-based methods, some prediction methods apply a template-based approach whenever possible and only fall back to sequence-based *de novo* prediction if no structure or template is known. For example, IonCom [94] provides predictions for 13 metal and four acid radical ion ligands by combining a sequence-based model trained on sequence profiles using a modified AdaBoost [99] algorithm with predictions from COFACTOR, TM-SITE, S-SITE, and COACH.

ProNA2020 [62] is a method to predict protein-protein, protein-DNA, and protein-RNA binding both on a per-protein and a per-residue level. First, it predicts whether the given protein sequence binds to another protein, DNA, or RNA. For this step, ProNA2020 combines *de novo* prediction with homology-based inference. Annotations are transferred from a sequence-similar protein if available (homology-based inference); otherwise, the potential binding partner is predicted combining a Profile Kernel SVM [65, 66] and ProtVec [69]. The per-residue prediction is carried out through three Artificial Neural Networks (ANN) trained individually to predict residues binding to proteins, DNA, or RNA, respectively, using various predicted features available through PredictProtein [68].

New, completely sequence-based approaches facilitate concepts from Deep Learning. For example, DeepCSeqSite [95] consists of multiple CNN layers trained using seven different input features: PSSM, predicted structural features, conservation scores, residue type, and position embedding [95]. DeepCSeqSite is trained on 14 different ligands including small molecules and metal and acid radical ions without considering binding to other macromolecules (DNA, RNA, and other proteins).

In general, while the best-performing methods to predict binding residues require the availability of protein structures or similar proteins with known annotations, many well performing sequence-based *de novo* methods exist. Especially, applying homology-based inference or structure-based methods if available, and otherwise, relying on sequence-based *de novo* predictions offers a promising approach ensuring good performance and broad applicability. However, most of the currently existing methods focus on a specific set of ligands, e.g., there are predictors specialized for macromolecules like ProNA2020 [62] or only applicable to metal ions like IonCOM [94]. Even more specialized predictors focusing on only one specific ion exist [100, 101]. Therefore, the ligand of interest usually needs to be known prior to using these methods. In general, current methods to predict



binding residues are not applicable to all protein sequences because they are restricted to certain ligands or rely on features not available for all sequences.

To allow predictions for all proteins independently of the bound ligand, I propose two methods to predict binding residues which are less restrictive in terms of ligand type and are solely based on sequence information (see Chapter 4).

### 1.3.3. Prediction of Subcellular Localization

Different approaches exist to predict subcellular localization and are, for example, based on homology-based inference, the identification of sorting signals, the utilization of functional annotations, or the application of *de novo* prediction methods. Hybrid approaches combine different concepts into one method.

The identification of sorting signals provides a simple first step towards elucidating the subcellular localization of a protein because the transport of many proteins is mediated through the presence of such signals. For example, SignalP-5.0 [102] predicts signal peptides using a deep neural network approach, which allows the distinction of secreted proteins, i.e., proteins localized outside of the cell, from non-secreted proteins inside the cell. However, using known sorting signals only allows the identification of the localization of few proteins [103], indicating that many sorting signals remain unknown or that proteins are also transported through other mechanisms not requiring the presence of sorting signals.

Because of the strong connection of subcellular localization and protein function, text-based approaches allow the inference of subcellular localization from functional annotations of a protein. For example, in an eukaryotic cell, DNA only occurs in the nucleus, and a protein annotated as “DNA-binding” has to be localized to the nucleus to execute its function. LOCKey [104] is an example for a text-based method using this connection to predict subcellular localization. It created a rule library from proteins with known subcellular localizations to infer localization from annotations available in Swiss-Prot [6] for proteins without known localization. However, since text-based approaches rely on functional annotations which are not necessarily available for all sequences, they are usually not applicable to all proteins.

ML methods allow predictions from sequence alone and are generally applicable to all protein sequences. LocTree2 [53] uses a Profile Kernel SVM [65, 66] and a hierarchical

classification scheme to predict subcellular localization for eukaryotes, bacteria, and archaea, separately. Since prokaryotes lack larger membrane-bound compartments, only three and six classes are predicted for archaea and bacteria, respectively. For eukaryotes, LocTree2 distinguishes 18 different classes including ten major compartments of which some are further divided into membrane and non-membrane (e.g., LocTree2 considers nucleus and nuclear membrane as two different classes). The hierarchical prediction scheme allows to first separate soluble proteins from membrane-bound ones and then, follows the basic localization pathways (secretory and non-secretory pathway) in the cell [53]. LocTree3 [60] builds upon LocTree2 by combining it with homology-based inference, further improving the predictive power of the method.

DeepLoc [54] uses recurrent neural networks and an attention mechanism to predict for eukaryotic proteins (1) whether a protein is soluble or membrane-bound and (2) to which of ten compartments it localizes. While the hierarchical prediction scheme has improved performance for LocTree2 [53], a similar approach applied for DeepLoc could not increase its performance [54].

In general, current methods to predict subcellular localization focus on around ten major compartments without considering substructures, e.g., inside the nucleus. Also, most methods only predict one subcellular localization per protein. However, especially in dynamic compartments like the nucleus, where some substructures only form at certain time points, proteins could easily be associated with multiple localizations throughout the cell cycle. Therefore, while being very accurate, current prediction methods do not allow more fine-grained analysis through predictions of substructures or multi-localization predictions.

To overcome these limitations, I introduce a method specialized to predict sub-nuclear compartments also allowing the assignment of multiple compartments to one protein (see Chapter 5).

### 1.4. Outline of This Work

This dissertation aims at advancing prediction methods for three different aspects of protein function (GO terms, binding residues, and sub-nuclear localization) utilizing various ML techniques. First, in Chapter 2, I investigate the importance of ML applications in biology and medicine in general, with a strong focus on how technical correctness is

achieved, and how collaborations between authors from different fields of expertise influence the developed ML models. Chapter 3 presents *goPredSim*, a new method to predict GO terms using a simple yet effective approach which is similar to homology-based inference, but uses sequence embeddings derived from deep learned LMs instead of protein sequences. Chapter 4 focuses on the prediction of binding residues. First, I present the method *bindPredictML17* which predicts binding residues through an ANN trained on evolutionary information. This method relies on features which can be difficult to compute and was only trained on a subset of proteins with binding annotations (namely DNA-binding proteins and enzymes). I also describe a potential method to improve upon *bindPredictML17* by training a CNN and replacing the hand-crafted features with protein embeddings. Section 4.2 outlines preliminary results for the predictive performance of this method called *bindPredictDL*. In Chapter 5, I introduce *LocNuclei*, a method to predict sub-nuclear localizations of proteins by combining homology-based inference with a Profile Kernel SVM. *LocNuclei* distinguishes between 13 different sub-nuclear compartments and allows multiple compartments to be predicted for one protein. The analysis of GO terms and protein-protein interactions helps to assess whether the resulting predictions reveal insights about the nuclear mechanisms a protein is involved in and consequently about the overall function of a protein. Finally, this dissertation concludes with a summary of all presented results in Chapter 6.



## 2. Validity of Machine Learning in Life Sciences Increased through Collaborations

### 2.1. Preface

Quality and validity of Machine Learning (ML) models depend on two major factors: (1) size, quality, and universal validity of data, and (2) the correct development and assessment of the resulting models [64, 105]. To ensure that both aspects are met, the development and application of ML models to the life sciences requires expertise from both computational and biological or medical fields. Therefore, we hypothesized that interdisciplinarity should improve the validity of ML models, also leading to articles published in journals with higher impact factors and receiving higher number of citations. While previous literature discussed the influence of interdisciplinary collaborations on the number of citations and impact factor, the results were inconsistent [106–112].

In our study, we put emphasis on ML in life sciences and manually extracted information from 300 articles unavailable through automated assessment. The extracted information included the frequency of published data and software, the kind of applied evaluation, and the field of expertise of the authors. The field of expertise was determined by checking publicly available information about an author's scientific background and was grouped into computational sciences, biology, and medicine. In addition, impact factor, number of citations, and other metadata of the article were automatically retrieved. This allowed the separate analysis of scientific soundness and of impact as well as a more stringent and intuitive definition of interdisciplinarity: researchers from different disciplines co-authoring a work in contrast to purely looking at the number of disciplines citing a work.

Three results stood out: First of all, ensuring the validity of ML applications was impaired in many cases as only half of the articles shared their software, 64% shared data, and 81% applied any kind of evaluation. Only 26% met all three criteria. Secondly, the authors' scientific background highly influenced how technical aspects were addressed: Reproducibility and computational evaluation methods were more prominent if authors with a background in computational sciences were involved, while experimental verification was more often applied with experimentalists as co-authors. Thirdly, 73% of the ML applications resulted from interdisciplinary collaborations comprising authors from at least two of the three different disciplines. Our analysis suggested that, while collaborations between computational and experimental scientists were not associated with a higher impact factor, such collaborations led to more scientifically sound work. Both computational as well as experimental scientists benefit from working together: The first are given access to novel and challenging real-world biological data, increasing the scientific impact of their research, while the latter profit from computationally sound analyses improving the technical correctness of their work.

**Author contribution:** Katharina Selig and I performed the major part of data analysis and manuscript writing. I created and adapted the list of articles. Katharina Selig generated figures and performed statistical tests. Katharina Selig, Liel Cohen-Lavi, Yotam Frank, Peter Hönigschmid, Evans Kataka, Anja Mösch, Kun Qian, Avihai Ron, Sebastian Schmid, Adam Sorbie, Liran Szlak, and Ayana Dagan-Wiener collected data for the predefined list of articles. All authors read and approved the final manuscript.

### 2.2. Journal Article: Littmann, Selig *et al.*, *Nature Machine Intelligence* (2020)

**Reference:** Littmann, M., Selig, K., Cohen-Lavi, L., Frank, Y., Hönigschmid, P., et al. Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nature Machine Intelligence*, 2(1):18–24, 2020. doi:10.1038/s42256-019-0139-8

# Validity of machine learning in biology and medicine increased through collaborations across fields of expertise

Maria Littmann<sup>1,27\*</sup>, Katharina Selig<sup>2,27\*</sup>, Liel Cohen-Lavi<sup>3,4</sup>, Yotam Frank<sup>5</sup>, Peter Hönigschmid<sup>6</sup>, Evans Kataka<sup>6</sup>, Anja Mösch<sup>6</sup>, Kun Qian<sup>7,8</sup>, Avihai Ron<sup>9,10</sup>, Sebastian Schmid<sup>11</sup>, Adam Sorbie<sup>12</sup>, Liran Szlak<sup>13</sup>, Ayana Dagan-Wiener<sup>14</sup>, Nir Ben-Tal<sup>15</sup>, Masha Y. Niv<sup>14,16</sup>, Daniel Razansky<sup>9,10,17,18,19,20</sup>, Björn W. Schuller<sup>21</sup>, Donna Ankerst<sup>2</sup>, Tomer Hertz<sup>3,22,23</sup> and Burkhard Rost<sup>1,24,25,26</sup>

**Machine learning (ML) has become an essential asset for the life sciences and medicine. We selected 250 articles describing ML applications from 17 journals sampling 26 different fields between 2011 and 2016. Independent evaluation by two readers highlighted three results. First, only half of the articles shared software, 64% shared data and 81% applied any kind of evaluation. Although crucial for ensuring the validity of ML applications, these aspects were met more by publications in lower-ranked journals. Second, the authors' scientific backgrounds highly influenced how technical aspects were addressed: reproducibility and computational evaluation methods were more prominent with computational co-authors; experimental proofs more with experimentalists. Third, 73% of the ML applications resulted from interdisciplinary collaborations comprising authors from at least two of the three disciplines: computational sciences, biology, and medicine. The results suggested collaborations between computational and experimental scientists to generate more scientifically sound and impactful work integrating knowledge from both domains. Although scientifically more valid solutions and collaborations involving diverse expertise did not correlate with impact factors, such collaborations provide opportunities to both sides: computational scientists are given access to novel and challenging real-world biological data, increasing the scientific impact of their research, and experimentalists benefit from more in-depth computational analyses improving the technical correctness of work.**

Large amounts of experimental data triggered by technological advances are increasing the interaction between biology, medicine, and quantitative sciences<sup>1–3</sup>. For instance, the amount of genome sequencing data is growing exponentially while data storage capacity only grows linearly<sup>4</sup>. Numerous large databases in molecular biology and large clinical datasets increasing through electronic health records call for novel ways to interrogate, analyse and process biological and biomedical data for gaining biological and medical insights<sup>5</sup>.

Machine learning (ML) automatically identifies patterns and regularities in existing data to accurately predict for unseen data<sup>6</sup>.

Despite the complexity of the underlying mathematical concepts, ML has attracted broad attention even outside of the research community: querying Google Trends<sup>7</sup> with “machine learning” demonstrated an exponential increase over the past decade (January 2010–February 2019, data not shown). This general rise has been mirrored in many fields of biology and medicine—that is, the life sciences<sup>8–11</sup>—although keeping track with the rapid evolution of artificial intelligence (AI) challenges even those applying ML<sup>12</sup>. Typically, large biological or medical datasets enable the development of ML models that can be used to predict biological or clinical phenotypes through measurements from novel samples.

<sup>1</sup>Department of Informatics, Bioinformatics and Computational Biology, Technical University of Munich, Garching/Munich, Germany. <sup>2</sup>Department of Mathematics, Technical University of Munich, Garching/Munich, Germany. <sup>3</sup>National Institute for Biotechnology in the Negev, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>4</sup>Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>5</sup>The Blavatnik School of Computer Science, Tel-Aviv University, Ramat Aviv, Israel. <sup>6</sup>Department of Bioinformatics, Wissenschaftszentrum Weihenstephan, Technical University of Munich, Freising, Germany. <sup>7</sup>Chair of Human-Machine Communication, Technical University of Munich, Munich, Germany. <sup>8</sup>Educational Physiology Laboratory, Graduate School of Education, The University of Tokyo, Tokyo, Japan. <sup>9</sup>Institute for Biological and Medical Imaging, Helmholtz Center Munich, Neuherberg, Germany. <sup>10</sup>Faculty of Medicine, Technical University of Munich, Munich, Germany. <sup>11</sup>Chair of Food Chemistry and Molecular Sensory Science, Technical University of Munich, Freising, Germany. <sup>12</sup>Chair of Nutrition and Immunology, Technical University of Munich, Freising, Germany. <sup>13</sup>Weizmann Institute of Science, Rehovot, Israel. <sup>14</sup>The Institute of Biochemistry, Food and Nutrition, The Robert H. Smith Faculty of Agriculture, Food and Environment, The Hebrew University, Rehovot, Israel. <sup>15</sup>Department of Biochemistry and Molecular Biology, George S. Wise Faculty of Life Sciences, Tel Aviv University, Tel Aviv, Israel. <sup>16</sup>The Fritz Haber Center for Molecular Dynamics, The Hebrew University, Jerusalem, Israel. <sup>17</sup>Faculty of Medicine, University of Zurich, Zurich, Switzerland. <sup>18</sup>Institute of Pharmacology and Toxicology, University of Zurich, Zurich, Switzerland. <sup>19</sup>Institute for Biomedical Engineering, University of Zurich and ETH Zurich, Zurich, Switzerland. <sup>20</sup>Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland. <sup>21</sup>Group on Language, Audio and Music, Imperial College London, London, UK. <sup>22</sup>The Shraga Segal Department of Microbiology and Immunology, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>23</sup>Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research Center, Seattle, WA, USA. <sup>24</sup>Institute for Advanced Study, Garching/Munich, Germany. <sup>25</sup>School of Life Sciences, Technical University of Munich, Freising, Germany. <sup>26</sup>Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY, USA. <sup>27</sup>These authors contributed equally: Maria Littmann, Katharina Selig. \*e-mail: [littmann@rostlab.org](mailto:littmann@rostlab.org); [katharina.selig@tum.de](mailto:katharina.selig@tum.de)

Quality and validity of ML models hinge on two primary factors: (1) size, quality and universal validity of data; and (2) the correct development and assessment of the resulting models<sup>3,13</sup>. Successful ML applications extract generic principles from today's data, allowing the generalization—that is, accurate prediction—for tomorrow's data. This needs proper extraction and processing of data and features often requiring expert knowledge<sup>14–16</sup>. The development and application of ML models to the life sciences needs expertise from both computational and biological/medical fields. In contrast, ML applications to areas such as object and speech recognition or complex games (including chess and Go/Weiqi) for which task and success are more clearly defined require mainly expertise in ML.

### Collaborations across fields of expertise

Throughout science, interdisciplinarity has become important to break new grounds<sup>17,18</sup>. Several recent studies<sup>17,19–24</sup> investigated the role of interdisciplinarity by automatically extracting tens and hundreds of thousands of publications (for example, from *Web of Science* or the *Proceedings of the National Academy of Sciences*). Toward this end, one definition of interdisciplinarity is as follows: if an article is published and cited in different fields or subfields (for example, the US *National Science Foundation* classifies journals into 14 different disciplines and 143 subdisciplines<sup>17,21</sup>), the article is deemed 'interdisciplinary'<sup>17,21,24</sup>. Others define interdisciplinarity as articles published by authors from different disciplines, an approach so far limited to Italian scientists due to a public directory mapping Italian researchers to disciplines<sup>19,20</sup>.

The scientific impact of an article is usually measured by its number of citations<sup>17,24</sup>. To correct for field- and journal-specific effects, that number is normalized by time (years since publication) and by the journal's impact factor<sup>23,24</sup>. Since the impact factor is calculated from the number of citations of articles published in this journal<sup>25</sup>, articles from higher-ranked journals are expected to have higher citation counts.

All those automated studies allowed the assessment of many articles while being limited to the extraction of only a particular type of information. The studies disagree in their findings regarding the importance of interdisciplinary collaborations: one finds no consistent correlation between impact and interdisciplinarity from sampling over 750,000 publications: for some disciplines, interdisciplinarity was proportional to citations; for others (including physics) the relation was reversed<sup>24</sup>. Another work, focusing on more than 15,000 publications from physics, found interdisciplinarity was proportional to citation rates but only when published in journals with citation rates below average<sup>23</sup>. Yet other studies, based on 751,766<sup>17</sup> and 71,633 publications<sup>20</sup>, agreed that interdisciplinary work creates higher impact than non-interdisciplinary work. Also, specific collaborations between scientists from related fields lead to higher-impact publications than generic collaborations between scientists from very different fields<sup>20</sup>. Clearly, there is no simple common thread running through all of those findings. However, what made us revisit this question and begin our analysis were three other reasons: (1) the focus on ML and the life sciences, not explicitly covered by others; (2) the aim of separating the analysis of scientific quality (soundness) from impact; and (3) the introduction of a more rigorous definition of interdisciplinarity—instead of proxying by the number of disciplines citing a work, we require experts from different disciplines to co-author a work, a definition similar to the one used for the analysis of Italian authors<sup>19,20</sup>.

### Focus of this work

Here, we assessed several aspects of ML applications in the life sciences. We started with the selection of 17 journals representing computational/experimental biology and medicine (see Supplementary Information). Among all papers published in those 17 journals in the years 2011–2016, keyword searches (Supplementary Table 1)

matched 4,306 articles, where about 2,100 of those were deemed correct hits based on the observed false positive rate for a subset of articles. From those, initially 250 were randomly selected (see Supplementary Information; complete list in Supplementary Dataset 1, list of identified falsely extracted articles is provided in Supplementary Dataset 2). Subsequently, we applied the same selection process and chose another 50 papers from 2018 to verify that the major findings have not changed through the most recent advent of deep learning<sup>9,10</sup>. In contrast to previous studies<sup>17,19–24</sup>, our assessment focused on ML applications in the life sciences and all information was manually extracted from the articles. This allowed, for instance, to correct the 50% false positives from the keyword searches, and also to define interdisciplinarity through the authors' scientific backgrounds by reading partial CVs for 1,918 authors of the 250 papers. Each article was classified independently by two of us. These investments limited the number of papers analysed but allowed a more fine-grained assessment not accessible to automatic extraction.

Our focus had several implications, including that all papers reported applications of ML to the life sciences, as opposed to more theoretical treatments. In some sense, the application of ML (computational sciences) to the life sciences is by definition interdisciplinary. Thus, we could sharpen the perspective by distinguishing the expertise contributing to the application of ML to the life sciences with authors from potentially three disciplines: computational sciences, biology and medicine (expertise of author verified through CV, not through affiliation). The number of different disciplines presented in the author list proxied the level of interdisciplinarity with values from 1 to 3.

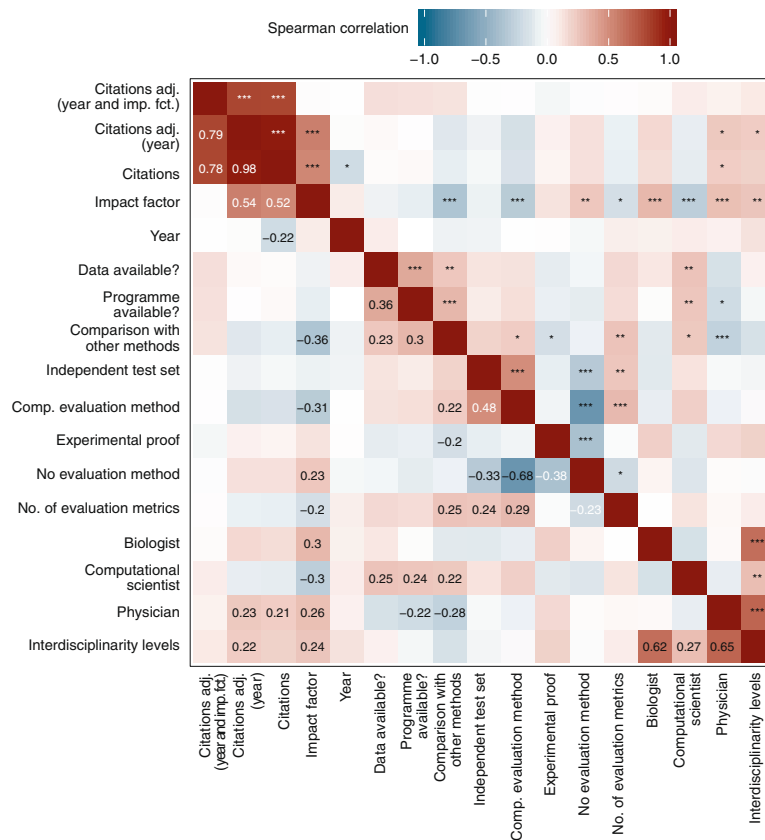
We proxied the validity of papers describing the application of ML methods to biology and medicine through six different indicators. The first four relate to whether the method was assessed in ways needed to ascertain that it works as promised (or at all). We asked: did the authors use cross-validation or other evaluation methods (V1: binary value), more than one single measure for performance (V2: integer), additional test sets (V3: binary value) or experimental verification (V4: binary value)? While method evaluation might correctly estimate performance for unseen data without V4, it appears impossible to accomplish this simple objective without V1–V3, let alone to develop the best possible method. The last two indicators related to sharing methods and results. These were sharing data (V5), programmes and codes (V6) through publicly available sites. Typically, reviewing ML applications by journal reviewers and the public at large requires availability of data and programmes in a form beyond what is available through description of methods.

The correct application of ML requires expertise from those familiar with ML and those familiar with the life sciences, that is, different disciplines. Thus, we hypothesized articles written by research teams from different disciplines to be more likely to report the necessary evaluation methods ensuring proper implementation of ML methods, to make their data publicly available so others could validate their results, and, subsequently, to be accepted in higher-ranked journals and have more citations.

### Results and discussion

**Three levels of interdisciplinarity.** By definition, all the papers analysed applied methods from computational fields to the life sciences—that is, were intrinsically interdisciplinary. All 250 papers analysed might have been considered interdisciplinary by automated analyses checking from which field/discipline the article was quoted. To generate a more detailed lens, we distinguished three disciplines (computational scientists, biologists and physicians) and introduced interdisciplinarity as a number ranging from one to three depending on how many disciplines were represented by the authors of the work. Most of the 250 papers were co-authored





**Fig. 1 | Spearman correlation coefficients for numeric and binary variables.** Correlation between the different criteria of 250 articles using the Spearman correlation tested at a significance level of 0.05. Significant *p*-values are displayed using \* for *p*-value < 0.05, \*\* for *p*-value < 0.01 and \*\*\* for *p*-value < 0.001 after adjusting for multiple testing using the Benjamin–Hochberg procedure. Blank squares denote that the correlation is non-significant. Citations adj. (year) and citations adj. (year and imp. fct.) denote the citations adjusted by year and by year and impact factor, respectively.

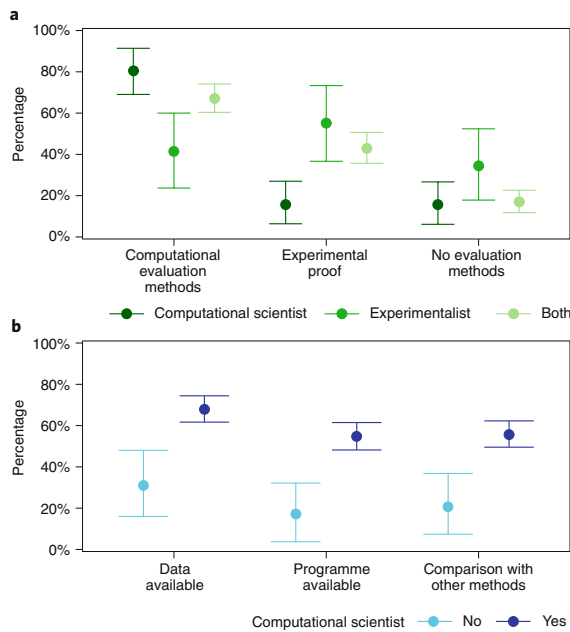
by two disciplines (one, 27%; two, 53%; three, 20%). Given these levels, we could classify all papers according to their level of interdisciplinarity and differentially analyse the key indicators: validity (evaluation and sharing) and impact (number of citations (NC); NC adjusted by year, equation (1) in Supplementary Information; impact factor, and NC adjusted by year and impact factor, equation (2) in Supplementary Information).

58% of the chosen 250 papers (see Supplementary Information for more details on how these articles were selected) appeared in only four of the 17 journals (by occurrence: *Bioinformatics*, *Proceedings of the National Academy of Sciences*, *PLOS Computational Biology* and *BMC Bioinformatics*; see additional results in Supplementary Information, including Supplementary Figs. 1, 2, 3 and 4, for more details)—that is, were 2.5-fold over-represented. While the disciplines of biologist and physician correlated positively with impact factor ( $\rho = 0.30/p\text{-value} < 0.001$ ,  $\rho = 0.26/p\text{-value} < 0.001$ , respectively), computational science correlated negatively ( $\rho = -0.30/p\text{-value} < 0.001$ ; Fig. 1). Computational scientists might focus more on methods, while biologists and physicians focus more on new data that tend to be highly cited in the life sciences.

**Scientific validity higher with experts participating in collaboration.** Evaluation methods (for example, cross-validation), usage of independent test sets, and/or experimental proofs reduce the

chance of overfitting and enhance the applicability of the model to future data. Indeed, 80% of the articles with only computational authors applied some evaluation methods or independent tests; compared to 41% of those written by ‘experimentalists’ (biologists and physicians; Fig. 2a). However, most articles written solely by experimentalists provided experimental proof (55%), so did 16% of those from only computational co-authors (Fig. 2a). The corresponding numbers for interdisciplinary collaborations between computational and experimental scientists (level of interdisciplinarity  $\geq 2$ ) were between these two extremes: 67% evaluated their methods and 43% provided experimental proof, suggesting that such collaborations facilitate experimental and computational validation. On the flip side, 19% of all articles did not provide any evaluation; this number rose as high as 34% without computational co-authors (Fig. 2a).

Several evaluation metrics are required to assess the performance of ML applications (for example, precision, recall, accuracy or confusion matrices). 6% of all articles used no evaluation metric, 53% used one or two, and 6% used over five (Supplementary Fig. 5). Although more metrics do not necessarily imply better assessment, even for binary predictions (separation of two classes/classifications), we have to consider the predictive power of the model for both classes separately—that is, minimally we need two evaluation metrics. More complex problems require more evaluation metrics.



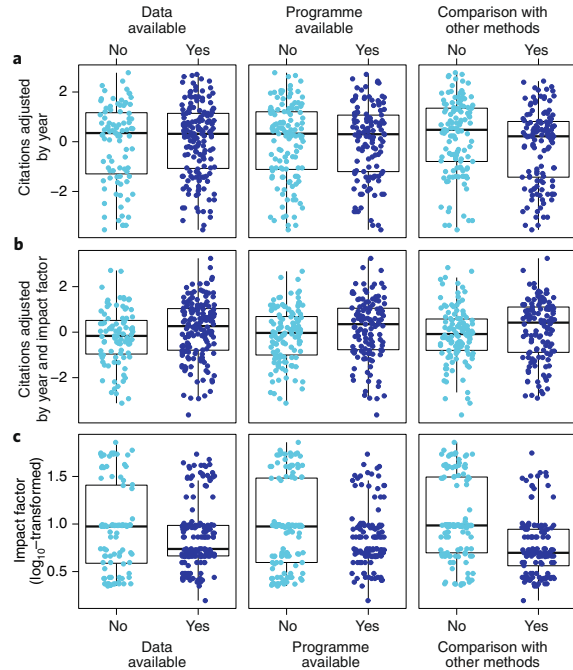
**Fig. 2 | Method validation, comparison and data and programme sharing depends on author expertise.** Percentages of 250 articles by evaluation methods, data or programme sharing and comparison with other methods split by authors' backgrounds are shown with 95% percentile bootstrap confidence intervals based on 1,000 bootstraps. **a**, Articles involving a computational scientist applied a computational evaluation method more often than articles with only an experimentalist (physician or biologist). Articles co-authored by experimentalists provided experimental proof more often than those without. Providing no evaluation method was more common among articles written solely by experimentalists. **b**, The involvement of a computational scientist was highly correlated with sharing the data, making the programme available, or performing a comparison with other methods.

Typically, clearly more than two metrics are needed to show different strengths and weaknesses of a prediction method.

About half (52%) of the methods were compared to others; this again dropped to 21% without computational co-authors ( $p$ -value = 0.001; Fig. 2b). Although crucial for validation, method comparisons might make descriptions more complex, leading to rejection from higher-ranked journals (Fig. 3c) and possibly to lower impact (Fig. 3c), although adjusting by impact factor as well suggested a slight pay-off from method comparisons in terms of citations (Fig. 3b).

Reproducibility is a pillar of science<sup>26–28</sup>, partially relying on making data and methods publicly available. It is particularly critical for ML applications because many minor technical details may invalidate results<sup>25</sup>. Overall, 64% of the articles shared their data (with large variation between journals: from *Nucleic Acids Research* = 89% to *New England Journal of Medicine* = 8%; Supplementary Fig. 6), reflecting the general trend that articles from medicine shared data the least (Supplementary Fig. 7). We could not establish whether this is related to sensitive patient data. While all journals encourage data sharing, many do not enforce it.

Overall, 68% of the articles with computational scientists shared data, opposed to 31% without ( $p$ -value < 0.001; Fig. 2b). 57% of the articles relied on data extracted from public resources or previous articles; however, 22% of those that did, did not publish their data.



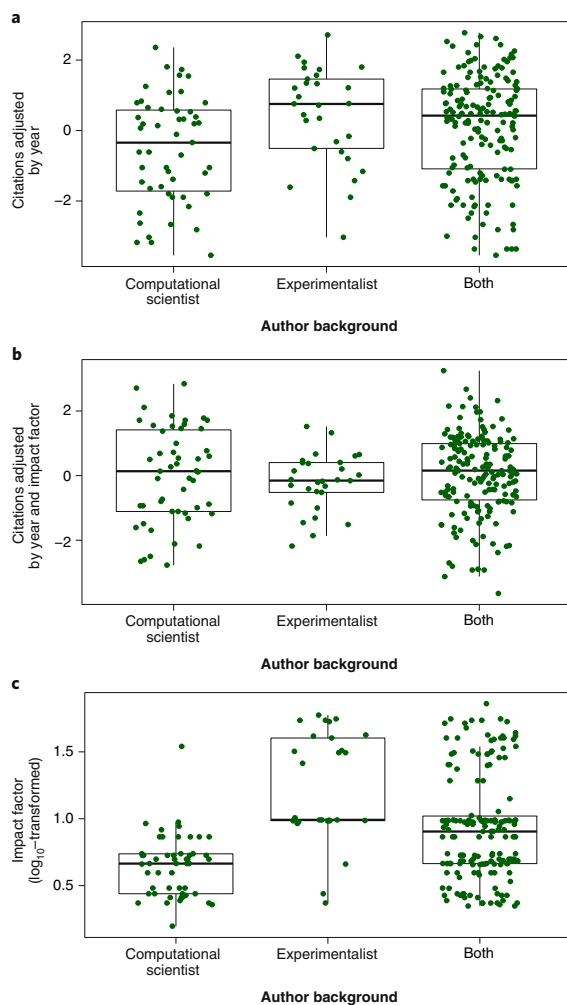
**Fig. 3 | Sharing and method comparison hardly impact citations.**

Boxplots of adjusted citations and  $\log_{10}$ -transformed impact factor of 250 articles split by data or programme sharing and comparison with other methods. Vertical bars indicate largest (smallest) value within 1.5 times the interquartile range above (below) the third (first) quartile. **a**, Number of citations adjusted by year were not influenced by data or programme availability. Comparing the developed method to others led to a small decrease in the number of citations. **b**, Adjusting by impact factor as well showed a small, but non-significant, trend towards higher citations when data or programme were available, or a comparison to other methods was performed. **c**, The impact factor was slightly higher for articles that did not make data or programme available, or compared their method to others, where only the last difference was statistically significant.

Data sharing was highest for collaborations with computer scientists (Fig. 2b).

Experimentalists might benefit from colleagues with knowledge in computer science to add evaluation methods, bring a greater variety of tools, and help with the interpretation of the scientific and statistical significance of results, therefore focusing more on technical aspects; while computational scientists benefit from the access to new data, domain knowledge and experimental verification of the results. Therefore, collaborative work will generate more scientifically sound and impactful work.

**Collaborations of scientists with different expertise were somewhat cited more often.** Interdisciplinary collaborations of researchers from different fields seem increasingly important to generate new ideas and results<sup>29,30</sup>. The higher the level of interdisciplinarity, the higher the NC adjusted by year ( $\rho = 0.22$ ,  $p$ -value = 0.02; Fig. 1, Supplementary Fig. 8) and the higher the impact factor ( $\rho = 0.24$ ,  $p$ -value = 0.002; Fig. 1, Supplementary Fig. 8). When adjusting NC by impact factor as well, the correlation was no longer significant (Fig. 1, Supplementary Fig. 8), suggesting that interdisciplinary articles were cited more mainly because they were published in higher-ranked journals (Supplementary Fig. 8). The correlation between



**Fig. 4 | Number of citations and impact factor not consistently higher for collaborations.** Boxplots of adjusted citations and  $\log_{10}$ -transformed impact factor of 250 articles split by authors' backgrounds. Vertical bars indicate largest (smallest) value within 1.5 times the interquartile range above (below) the third (first) quartile. **a**, The number of citations adjusted by year was slightly higher for articles solely written by experimentalists compared to articles involving computational scientists. **b**, Adjusting by impact factor as well removed this difference. This suggests that the higher number of citations for experimentalists was mainly caused by the fact that their work got accepted in higher-ranked journals. **c**, Impact factor was higher for articles only published by experimentalists (biologists and/or physicians) than for articles with computational scientists.

impact factor and level of interdisciplinarity (Supplementary Fig. 8) suggested that authors profit from collaborations.

Closer analysis of the correlation between interdisciplinarity and impact refined the message: distinguishing just two groups (computational and experimental), revealed NC to be higher for research teams of only experimental scientists (Fig. 4a). The results for impact factor and NC adjusted by impact factor and year suggested that the higher NC originated essentially from experimentalists publishing in higher-ranked journals (Fig. 4b,c). For research teams

with only computational expertise, contributions from experimentalists can help to add new data, find biologically relevant applications and interpretations of the results, and increase the relevance of ML applications leading to more visibility of conducted research because it might be accepted in higher-ranked journals.

Did scientific validity (evaluation and sharing) correlate with impact? Computational evaluations correlated negatively with the impact factor ( $\rho = -0.31$ ,  $p$ -value  $< 0.001$ ); using no evaluation method correlated positively with the impact factor ( $\rho = 0.23$ ,  $p$ -value = 0.004), but we could not detect a significant relationship between impact factor and experimental proof (Fig. 1). Since all articles analysed here focus on applications, the absence of proper evaluation—independent of the focus of a paper—clearly contradicts good scientific conduct.

Data sharing was not rewarded by increases in NC adjusted by year (Fig. 3a), although adjusting by impact factor as well hinted at a tendency for sharing to lead to more citations (Fig. 3b). Thus, although data sharing is crucial to ascertain validity and reproducibility, it is not incentivized by increased visibility. In fact, there was no significant difference in the impact factor (Fig. 3c).

Software sharing also did not correlate with NC adjusted by year (Fig. 3a); the trend changed toward more cited when adjusting NC by impact factor as well (Fig. 3b). On the contrary, not sharing software seemed to lead to acceptance of articles in higher-ranked journals, but again the difference was not significant (Fig. 3c). Certainly, method sharing is crucial for reproducibility and for the impact of a method on science. Therefore, we were surprised that programme sharing appeared neither crucial for visibility nor acceptance in the research community as proxied by citations and journal rank. Ultimately, this might shed light on the limitations of such measures to evaluate scientific impact.

**More computational scientists involved in 2018.** AI and ML are so rapidly evolving that papers published from 2011–2016 might simply not be up to date enough to capture the newest trends. We attempted to address this issue by analysing another 50 articles describing ML applications to the life sciences published in 2018 (selected and analysed largely by the same criteria as the other 250; see Supplementary Information for details; complete list in Supplementary Dataset 3). The major differences were: fewer publications without computational scientists (6% 2018 versus 12% 2011–2016), and programme sharing rose (70% versus 50%). Although data sharing did not change significantly (68% versus 64%), those papers that shared data were cited more often and accepted to higher-ranked journals, but we could not detect a significant difference (Supplementary Fig. 9). Other aspects also did not change, neither the fact that papers sharing programmes tended to be published in lower-ranked journals (Supplementary Fig. 6) nor the correlation between number of involved disciplines and the proxies for impact (for example, NC adjusted by year, impact factor, and NC adjusted by year and impact factor). Overall, the most substantial change was that computational scientists contributed more often in 2018. This might reflect the increasing complexity of realizing ever more popular deep learning-type solutions of ML.

**Limitations.** Although our analysis revealed interesting insights, some issues remain to be addressed in the future. First, thoroughly analysing more than 300 articles will render the conclusions more valid. Second, we proxied impact and visibility through number of citations and the impact factor. However, the number of citations can be influenced by other factors that can seem superficial and can be controlled by the authors<sup>31</sup>, and it is hard to compensate for these ones. Using the impact factor for measuring scientific impact has been criticized in the literature and the increasing use of social media might increase the visibility of research independent of the journal's impact factor<sup>32,33</sup>. Third, the scope of a journal

might influence the description of ML applications. Journals focusing on methodologies are more likely to require certain standards in ML; those focusing on biologically and medically relevant novelties are less likely to specifically ask for methodological details. Fourth, we considered any publicly available information to assign author disciplines but could not account for paid statisticians not listed as authors. A variety of medical scientists from pathologists to clinicians were all simplified as physician, ignoring large differences in scientific training. These simplifications might lead to underestimating computational expertise in publications. Furthermore, we considered data and programme availability as stated in the articles but did not attempt to contact authors to obtain those if not available. Finally, since several aspects in our analysis that correlated with the impact factor also correlated with each other, confounding factors might influence the results and these interrelationships are difficult to separate.

### Conclusions

We analysed 250 articles describing ML applications to the life sciences published 2011–2016 and another 50 articles published in 2018 in 17 journals from 24 different biological/medical fields (see Supplementary Information for more information). This diversity of fields was mirrored by the diversity of how ML was applied. Reproducibility and correct evaluation of results are crucial to ascertain validity and reliability of ML applications. Surprisingly, many articles did not focus on these aspects: 50% shared no software, 36% shared no data, and 19% applied no evaluation. In fact, an entire third (34%) of the articles only written by experimentalists described no evaluation. While we hypothesized that ensuring validity of ML applications would be necessary to achieve high visibility of the research, we found the opposite: more valid work was often published in lower-ranked journals, attracting fewer citations (Fig. 1, Fig. 3).

In general, how these technical aspects were addressed was highly influenced by the authors' scientific backgrounds: reproducibility and evaluation were more prominent with computational scientists as co-authors (Fig. 2), while articles co-authored by experimentalists more frequently provided experimental proof (Fig. 2). Thus, collaborations of authors from different disciplines provided more opportunity for higher-quality results integrating knowledge from various fields of expertise.

We hypothesized that collaborative research should also be cited more often and be accepted in higher-ranked journals. However, this was only true for computational scientists who profited from collaborating with experimentalists by getting accepted in higher impact factor journals (Fig. 4c).

One of the most substantial challenges for ML is a comprehensive, adequate evaluation; incorrect application of such tools can lead to drawing false conclusions or to overestimating the predictive power of a method. Collaborations between computational and experimental scientists substantially increased the correctness of evaluations and the likelihood of reproducibility. Thus, interdisciplinary collaborations increased the scientific validity of published research. As the enforcement of data and programme transparency will increase, ML methods in biology and medicine will have to be implemented more carefully. While using the impact factor to measure the success of a scientific article currently does not show an advantage of collaborations for experimental scientists (Fig. 4c), we suggest that these collaborations will become more frequent and impactful in the near future.

Received: 9 August 2019; Accepted: 6 December 2019;

Published online: 13 January 2020

### References

- Bleicher, K. H., Bohm, H. J., Muller, K. & Alanine, A. I. Hit and lead generation: beyond high-throughput screening. *Nat. Rev. Drug. Discov.* **2**, 369–378 (2003).
- Sulakhe, D. et al. High-throughput translational medicine: challenges and solutions. *Adv. Exp. Med. Biol.* **799**, 39–67 (2014).
- Howard, J. Quantitative cell biology: the essential role of theory. *Mol. Biol. Cell.* **25**, 3438–3440 (2014).
- Cook, C. E. et al. The European Bioinformatics Institute in 2016: data growth and integration. *Nucl. Acids Res.* **44**, D20–26 (2016).
- Chicco, D. Ten quick tips for machine learning in computational biology. *BioData Mining* **10**, 35 (2017).
- Cios, K. J., Kurgan, L. A. & Reformat, M. Machine learning in the life sciences. *IEEE Eng. Med. Biol. Mag.* **26**, 14–16 (2007).
- Google Trends. *Google* <https://trends.google.de/trends> (2019).
- Rost, B., Radivojac, P. & Bromberg, Y. Protein function in precision medicine: deep understanding with machine learning. *FEBS Lett.* **590**, 2327–2341 (2016).
- Webb, S. Deep learning for biology. *Nature* **554**, 555–557 (2018).
- Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.* **18**, 851–869 (2017).
- Larranaga, P. et al. Machine learning in bioinformatics. *Brief. Bioinform.* **7**, 86–112 (2006).
- Frank, M. R., Wang, D., Cebrian, M. & Rahwan, I. The evolution of citation graphs in artificial intelligence research. *Nat. Mach. Intell.* **1**, 79–85 (2019).
- Domingos, P. A few useful things to know about machine learning. *Commun. ACM* **55**, 78–87 (2012).
- Chou, K.-C. Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.* **273**, 236–247 (2011).
- Ioannidis, J. P. et al. Increasing value and reducing waste in research design, conduct, and analysis. *Lancet* **383**, 166–175 (2014).
- Gron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, 2017).
- Chen, S., Arsenault, C. & Larivière, V. Are top-cited papers more interdisciplinary? *J. Informetr.* **9**, 1034–1046 (2015).
- Cummings, J. & Kiesler, S. Organization theory and the changing nature of science. *J. Org. Des.* **3**, 1–16 (2014).
- Abramo, G., D'Angelo, C. A. & Di Costa, F. Authorship analysis of specialized vs diversified research output. *J. Informetr.* **13**, 564–573 (2019).
- Abramo, G., D'Angelo, C. A. & Di Costa, F. Do interdisciplinary research teams deliver higher gains to science? *Scientometrics* **111**, 317–336 (2017).
- Chen, S., Arsenault, C., Gingras, Y. & Larivière, V. Exploring the interdisciplinary evolution of a discipline: the case of biochemistry and molecular biology. *Scientometrics* **102**, 1307–1323 (2015).
- Xie, Z., Li, M., Li, J., Duan, X. & Ouyang, Z. Feature analysis of multidisciplinary scientific collaboration patterns based on PNAS. *EPJ Data Sci.* **7**, 5 (2018).
- Rinia, E. J., van Leeuwen, T. N. & van Raan, A. F. J. Impact measures of interdisciplinary research in physics. *Scientometrics* **53**, 241–248 (2002).
- Larivière, V. & Gingras, Y. On the relationship between interdisciplinarity and scientific impact. *J. Am. Soc. Inform. Sci. Technol.* **61**, 126–131 (2010).
- Wallach, J. D., Boyack, K. W. & Ioannidis, J. P. A. Reproducible research practices, transparency, and open access data in the biomedical literature, 2015–2017. *PLoS Biol.* **16**, e2006930 (2018).
- Berger, B. et al. ISCB's initial reaction to the New England Journal of Medicine editorial on data sharing. *PLoS Comput. Biol.* **12**, e1004816 (2016).
- Drazen, J. M. Data sharing and the journal. *N. Engl. J. Med.* **374**, e24 (2016).
- Longo, D. L. & Drazen, J. M. Data sharing. *N. Engl. J. Med.* **374**, 276–277 (2016).
- Mind meld. *Nature* **525**, 289–290 (2015).
- Nissani, M. Ten cheers for interdisciplinarity: the case for interdisciplinary knowledge and research. *Soc. Sci. J.* **34**, 201–216 (1997).
- van Wesel, M., Wyatt, S. & ten Haaf, J. What a difference a colon makes: how superficial factors. *Scientometrics* **98**, 1601–1615 (2014).
- Fitzgerald, R. T. & Radmanesh, A. Social media and research visibility. *Am. J. Neuroradiol.* **36**, 637 (2015).
- Patton, R. M., Stahl, C. G. & Wells, J. C. Measuring scientific impact beyond citation counts. *D-Lib Magazine* **22**, 5 (2016).

### Acknowledgements

Thanks to T. Karl and I. Weise (both TUM) for invaluable help with technical and administrative aspects of this work. Thanks to the TUM Graduate School (in particular Z. Zhang) for organizing the summer school, to the TUM (in particular H. Keidel and W. Herrmann) for substantial support on several levels including financing the summer school, to the Weizmann Institute, Tel Aviv University, Technion and Hebrew University for financial and general support; thanks also to the enlightening talks by D. Cremers (TUM), M. Linal (IAS Israel, Hebrew University), Y. Ofra (Bar-Ilan University); thanks to PubMed for providing easy access to published articles and supporting automatic access; thanks to the maintainers of Biopython for providing excellent code to access various databases and process biological data. Last, but not least, thanks to all maintainers of public databases and to all experimentalists who enabled this analysis by

making their data publicly available. This work was supported by grant no. 640508 from the Deutsche Forschungsgemeinschaft (DFG).

### Author contributions

M.L. and K.S. performed the major part of data analysis and of writing the manuscript. M.L. created and adapted the predefined list of articles. K.S. generated figures and performed statistical tests. L.C. assisted in finding interesting correlations in the data by performing complex analyses and statistical test and in generating figures. M.L., K.S., L.C., Y.F., P.H., E.K., A.M., K.Q., A.R., S.S., A.S., L.S. and A. D.-W. participated in the summer school where the idea for this work was developed, were involved in agreeing on the goals and analysis methods of this work, were involved in data analysis by collecting data from the predefined list of articles, and assisted in writing the manuscript. M.L., K.S. and A.M. collected the data for 2018. N.B.-T., M.Y.N., D.R. and B.W.S. supervised the work over the entire time and proofread the manuscript. D.A. provided valuable comments, especially regarding statistical analysis and was involved in manuscript writing. T.H. and B.R. initiated and supervised the summer school where the idea for this project was developed. T.H. provided important comments to refine the

analysis and contributed to manuscript writing. B.R. supervised and guided the work over the entire time and proofread the manuscript. All authors read and approved the final manuscript.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary information** is available for this paper at <https://doi.org/10.1038/s42256-019-0139-8>.

**Correspondence** should be addressed to M.L. or K.S.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

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

© The Author(s), under exclusive licence to Springer Nature Limited 2020

**2.3. Supplementary Material: Littmann, Selig *et al.*, Nature Machine Intelligence (2020)**

In the format provided by the authors and unedited.

# Validity of machine learning in biology and medicine increased through collaborations across fields of expertise

Maria Littmann<sup>1,27\*</sup>, Katharina Selig<sup>2,27\*</sup>, Liel Cohen-Lavi<sup>3,4</sup>, Yotam Frank<sup>5</sup>, Peter Hönigsmid<sup>6</sup>, Evans Kataka<sup>6</sup>, Anja Mösch<sup>6</sup>, Kun Qian<sup>7,8</sup>, Avihai Ron<sup>9,10</sup>, Sebastian Schmid<sup>11</sup>, Adam Sorbie<sup>12</sup>, Liran Szlak<sup>13</sup>, Ayana Dagan-Wiener<sup>14</sup>, Nir Ben-Tal<sup>15</sup>, Masha Y. Niv<sup>14,16</sup>, Daniel Razansky<sup>9,10,17,18,19,20</sup>, Björn W. Schuller<sup>21</sup>, Donna Ankerst<sup>2</sup>, Tomer Hertz<sup>3,22,23</sup> and Burkhard Rost<sup>1,24,25,26</sup>

<sup>1</sup>Department of Informatics, Bioinformatics and Computational Biology, Technical University of Munich, Garching/Munich, Germany. <sup>2</sup>Department of Mathematics, Technical University of Munich, Garching/Munich, Germany. <sup>3</sup>National Institute for Biotechnology in the Negev, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>4</sup>Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>5</sup>The Blavatnik School of Computer Science, Tel-Aviv University, Ramat Aviv, Israel. <sup>6</sup>Department of Bioinformatics, Wissenschaftszentrum Weihenstephan, Technical University of Munich, Freising, Germany. <sup>7</sup>Chair of Human-Machine Communication, Technical University of Munich, Munich, Germany. <sup>8</sup>Educational Physiology Laboratory, Graduate School of Education, The University of Tokyo, Tokyo, Japan. <sup>9</sup>Institute for Biological and Medical Imaging, Helmholtz Center Munich, Neuherberg, Germany. <sup>10</sup>Faculty of Medicine, Technical University of Munich, Munich, Germany. <sup>11</sup>Chair of Food Chemistry and Molecular Sensory Science, Technical University of Munich, Freising, Germany. <sup>12</sup>Chair of Nutrition and Immunology, Technical University of Munich, Freising, Germany. <sup>13</sup>Weizmann Institute of Science, Rehovot, Israel. <sup>14</sup>The Institute of Biochemistry, Food and Nutrition, The Robert H. Smith Faculty of Agriculture, Food and Environment, The Hebrew University, Rehovot, Israel. <sup>15</sup>Department of Biochemistry and Molecular Biology, George S. Wise Faculty of Life Sciences, Tel Aviv University, Tel Aviv, Israel. <sup>16</sup>The Fritz Haber Center for Molecular Dynamics, The Hebrew University, Jerusalem, Israel. <sup>17</sup>Faculty of Medicine, University of Zurich, Zurich, Switzerland. <sup>18</sup>Institute of Pharmacology and Toxicology, University of Zurich, Zurich, Switzerland. <sup>19</sup>Institute for Biomedical Engineering, University of Zurich and ETH Zurich, Zurich, Switzerland. <sup>20</sup>Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich, Switzerland. <sup>21</sup>Group on Language, Audio and Music, Imperial College London, London, UK. <sup>22</sup>The Shraga Segal Department of Microbiology and Immunology, Ben-Gurion University of the Negev, Be'er-Sheva, Israel. <sup>23</sup>Vaccine and Infectious Disease Division, Fred Hutchinson Cancer Research Center, Seattle, WA, USA. <sup>24</sup>Institute for Advanced Study, Garching/Munich, Germany. <sup>25</sup>School of Life Sciences, Technical University of Munich, Freising, Germany. <sup>26</sup>Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY, USA. <sup>27</sup>These authors contributed equally: Maria Littmann, Katharina Selig. \*e-mail: [littmann@rostlab.org](mailto:littmann@rostlab.org); [katharina.selig@tum.de](mailto:katharina.selig@tum.de)

# Supporting online material (SOM) for: Validity of machine learning in biology and medicine increased through collaborations across fields of expertise

Maria Littmann, Katharina Selig, Liel Cohen-Lavi, Yotam Frank,  
Peter Hönigschmid, Evans Kataka, Anja Mösch, Kun Qian, Avihai  
Ron, Sebastian Schmid, Adam Sorbie, Liran Szlak, Ayana Dagan-  
Wiener, Nir Ben-Tal, Masha Y. Niv, Daniel Razansky, Björn W.  
Schuller, Donna Ankerst, Tomer Hertz & Burkhard Rost

## Table of Contents for SOM

<b>Table of Contents for SOM</b> .....	<b>1</b>
<b>Short description of SOM</b> .....	<b>2</b>
<b>SOM: Material &amp; Methods</b> .....	<b>3</b>
Generation of journal list.....	3
Generation of keyword list .....	3
Table S1: List of keywords.....	3
Generation of article list .....	4
Data extraction and analysis.....	5
Table S2: List of biological fields. ....	5
Adjustment of number of citations for impact factor and year. 7	
Construction of 2018 dataset for comparison.....	7
<b>SOM: Additional Results</b> .....	<b>8</b>
Coverage of machine learning varies between journals and fields. .....	8



Scientific validity higher with experts participating in collaboration .....	13
Collaborations of scientists with different expertise somehow cited more often.....	16
More computational scientists are involved for articles published in 2018.....	17
<b>SOM References .....</b>	<b>18</b>

## Short description of SOM

In this Supporting Online Material (SOM), we show a more detailed analysis of the underlying data and minor aspects mentioned in the text. We begin by a more detailed description of the way we conducted our analysis (SOM Methods & Materials). We chose the 250+50 articles from a larger list of articles that was extracted from PubMed using a pre-defined list of keywords (Table S1). The articles were chosen from 17 journals (*Bioinformatics*, *BMC Bioinformatics*, *Cell*, *IEEE Transactions on Biomedical Engineering*, *IEEE Transactions on Medical Imaging*, *Journal of Cheminformatics*, *Lancet*, *Molecular Informatics*, *Molecular Systems Biology*, *NAR*, *Nature*, *Nature Medicine*, *Nature Methods*, *NEJM*, *PLOS Computational Biology*, *PNAS*, *Science*). The number of chosen articles differed between journals (Fig. S1) and citations were on average higher for older articles (Fig. S1). This trend is not specific for ML, but also holds true for all articles published in Nature and Science (Fig. S1). The chosen articles are from 24 different fields out of a list of 49 fields (Table S2). The five most frequent fields covered 76% of all articles. Some journals primarily publish papers from specific fields (Fig. S2) and the involvement of authors from a specific scientific background differed between fields (Fig. S3). We did not observe a difference in the number of citations adjusted by year or adjusted by year and impact factor (Fig. S4). The difference of the impact factor between the fields is not significant, but articles focusing on *medicine*, *neuroscience*, and *oncology* were, on average, published in journals with higher impact factors than articles from *genetics* and *molecular biology* (Fig. S4).

While all journals encourage or enforce data sharing, the number of articles actually sharing data or program was influenced by the journal (Fig. S5). Journals like NEJM that rarely publish articles sharing their data also often publish medical research and articles from *medicine* share data least often compared to other fields (Fig. S6).

## SOM: Material & Methods

### Generation of journal list

We considered different aspects of ML manually extracted from the first 250 scientific articles, which were selected from a list of pre-defined journals representing the *life sciences* based on a discussion during a scientific meeting organized between the co-authors from a great diversity of life science and computational fields. The final list consisted of the following 17 journals: Bioinformatics, BMC Bioinformatics, Cell, IEEE Transactions on Biomedical Engineering, IEEE Transactions on Medical Imaging, Journal of Cheminformatics, Lancet, Molecular Informatics, Molecular Systems Biology, Nature, Nature Medicine, Nature Methods, New England Journal of Medicine (NEJM), Nucleic Acids Research (NAR), PLOS Computational Biology, Proceedings of the National Academy of Sciences (PNAS), Science.

### Generation of keyword list

We refined the set of search keywords related to ML in the life sciences by the following process (complete list given in Table S1). Start with a published exhaustive [list of algorithms](#); exclude ‘dimensionality reduction’, ‘semi-supervised learning’, and ‘other machine learning methods and problems’ as well as terms that either describe approaches in a very general way or are more a notation of the underlying problem, for example ‘binary classifier’. We removed abbreviations such as ANN and SVM to decrease false positives from the keyword search. However, we did maintain abbreviations for algorithms such as LASSO that are generally used without mentioning the full name.

**Table S1: List of keywords.**

AdaBoost	2	Hopfield network	0
Artificial Neural Network	0	K-means clustering	0
Association rule learning	0	K-medians	0
Autoencoder	2	K-nearest neighbors algorithm	0
Bagging	1	LARS	1
Bayesian Belief Network	0	LASSO	7
Bayesian Network	10	Learning vector quantization	0
Belief Network	2	Linear discriminant analysis	4
BIRCH	1	Linear regression	35
Blending	0	Logistic regression	29
Boltzmann machine	0	Machine learning	100

Boosting	4	Multilayer perceptron	0
Bootstrap aggregating	0	Multinomial logistic regression	0
Classification and regression tree	2	Multivariate adaptive regression splines	0
Conditional Random Field	4	Naive Bayes	7
Convolutional neural network	8	Neural Network	28
Cross-validation	30	OPTICS algorithm	0
DBSCAN	0	Ordinary least squares regression	0
Decision tree	3	Perceptron	22
Deep belief networks	0	Principal component analysis	18
Deep Boltzmann Machine	0	Radial basis function network	0
Deep Learning	17	Random Forests	5
Dimension Reduction	3	Recurrent neural network	0
Ensemble Methods	13	Restricted Boltzmann machine	0
Fisher's linear discriminant	0	Ridge Regression	1
Fuzzy clustering	0	Self-organizing map	2
Gradient boosted regression tree	0	Stacked Auto-Encoders	0
Gradient boosting machine	1	Stacked Generalization	0
Hidden Markov Models	14	Stepwise regression	1
Hierarchical Clustering	7	Support vector machines	36
Hierarchical temporal memory	1		

List of keywords used to extract articles covering machine learning from PubMed with the number of the 250 articles that match each keyword. To decrease the number of wrongly identified articles containing one of the keywords, but not dealing with machine learning, abbreviations were excluded from the list.

### Generation of article list

We accessed PubMed<sup>1</sup> through the *Entrez* package from *Biopython*<sup>2</sup> and keyword-searched the 17 journals for articles from 2011 to 2016 matching at least one keyword. This retrieval yielded a list of 4,306 articles. 250 articles were chosen: the first 125 were the most cited articles and the second 125 were picked randomly. The number of citations in the Entrez package refer to the citations count as given in PubMed. Most cited 125: for each year we chose the top  $x$  articles with the most citations. We chose  $x$  proportional to the number of all articles that matched at least one keyword and were published in that year. Random 125: we picked  $y$  articles at random for each journal again matching  $y$  to the overall proportions of that journal overall years. Articles matching keywords unrelated to ML (e.g. *neural network* referring to brain), and those with ML applications in non-life sciences (e.g. language

recognition) were replaced by other articles using the above selection procedure. From the original list of 250 articles, 124 were still part of the final list and certain articles had to be replaced multiple times. This suggests a false-positive rate of at least 50%. Therefore, we expect that around 2,100 articles from the 4,306 originally retrieved are matching our criteria. With analysing 250 of those, we chose a sample of 12% from the original set.

### Data extraction and analysis.

Information extracted from the 250 articles included the impact factor of the journal (extracted from [scijournal.org](http://scijournal.org) and [Web of Science](http://Web of Science)) the number of citations (the *Biopython Entrez* package grepping the numbers from PubMed) and scientific field of an article (Table S2, extracted from [Wikipedia](http://Wikipedia)), the scientific background of the authors, the dataset (size, data type, and data source), the type of ML algorithm applied, and the types of model validations and assessments. Bioinformatics was excluded from the fields because the application of ML to biology essentially is one aspect of bioinformatics. We assigned the field of an article based on the actual topics covered there and one article could be assigned to more than one field.

**Table S2: List of biological fields.**

Anatomy	1	Medicine	34
Astrobiology	0	Microbiology	9
Bacteriology	0	Molecular biology	64
Biochemistry	5	Mycology	1
Bioengineering	5	Neuroscience	23
Biogeography	0	Oncology	25
Biomechanics	1	Paleobiology	0
Biophysics	2	Paleontology	0
Biotechnology	0	Parasitology	1
Botany	0	Pharmacology	5
Cell biology	3	Photobiology	0
Chronobiology	0	Phycology	0
Cognitive biology	2	Physiology	2
Comparative anatomy	0	Plant physiology	0
Cryobiology	0	Population biology	1
Developmental biology	0	Psychobiology	0

Ecology	1	Radiobiology	0
Embryology	0	Sociobiology	0
Epidemiology	5	Structural biology	8
Evolutionary biology	0	Synthetic biology	0
Evolutionary developmental	0	Systems biology	15
Genetics	59	Theoretical biology	0
Gerontology	0	Virology	0
Immunology	4	Zoology	4
Marine biology	0		

List of biological fields used to classify the articles by topic with the number of articles matched to that field.

The scientific background of the authors was categorized into computational, biological and medical based on publicly available information about their profession and education. *Computational scientists* included authors from computer sciences, statistics, mathematics, and bioinformatics; authors from medical fields were referred to as *physicians*; those from other disciplines of biology were considered as *biologists*. In our opinion, these three classifications should cover the most important fields of expertise for ML applications in the life sciences. In fact, all authors could be assigned to one of these categories.

We counted the number of different disciplines to assess the interdisciplinarity of the author list. Thereby the minimal number of one described articles with authors from a single field, two means authors from two different backgrounds contributed, and for the maximum number of three disciplines, at least one author of each background category (biology, medicine, and computer science) had to contribute to the article.

Two co-authors investigated each article in detail, extracting the predefined information into a specified data format. Inconsistent data or spelling errors were corrected. For numeric and binary variables, we calculated the Spearman correlation coefficient  $\rho$  and tested the correlation while controlling the false-discovery rate using the Benjamini–Hochberg procedure<sup>3</sup> adjusting for 136 multiple comparisons. We assessed the relationship between categorical variables using a chi-square test of independence and between numeric and categorical variables using the Wilcoxon rank sum test, while adjusting for 3 and 4 multiple comparisons, respectively. In addition to the chi-square test for categorical variables, we constructed a percentile bootstrap 95% - confidence interval for the proportions with 1000 bootstrap samples. The PubMed IDs of the bootstrap samples are available as additional file

*bootstrap\_pubmed\_ids.csv*. We performed all statistical tests at a 0.05 significance level.

### **Adjustment of number of citations for impact factor and year.**

To allow a comparison of the citations between articles published in different years, we adjusted the number of citations using a linear regression model of the log-transformed citations on year (Eqn. S1).

$$\log(\text{citations}) = \beta_0 + \beta_1 \cdot \text{year} + \varepsilon \quad (\text{Eqn. S1})$$

The adjusted citations were the residuals as calculated by the log-transformed number of citations minus the estimated log-transformed number of citations for the corresponding year (Eqn. S2).

$$\text{citations}_{\text{adjusted}} = \log(\text{citations}_{\text{actual}}) - (\hat{\beta}_0 + \hat{\beta}_1 \cdot \text{year}) \quad (\text{Eqn. S2})$$

Since the impact factor is calculated from the number of citations (as described in the [Web of Science](#)), impact factor and number of citations are correlated. Adding the log-transformed impact factor as a covariate to the regression model can be applied to remove this correlation.

### **Construction of 2018 dataset for comparison.**

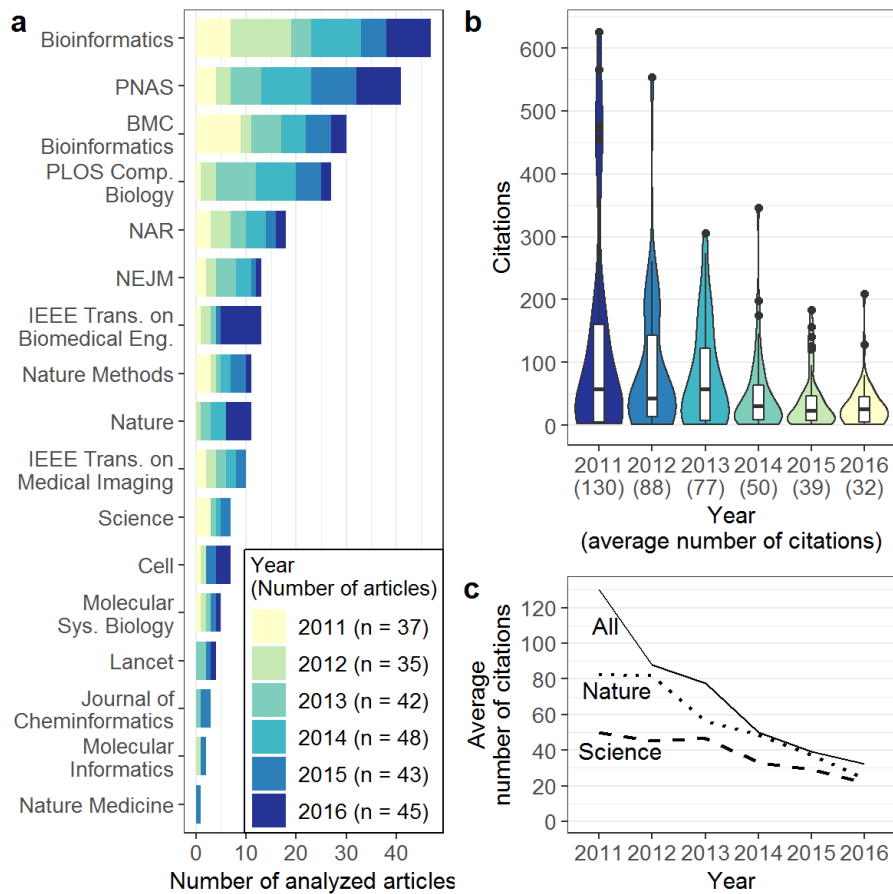
We extracted 50 articles from 2018, again 25 being the most cited ones and 25 being chosen at random. The random articles were extracted from the same 17 journals as the original list following the distribution of journals in there. We analysed these articles regarding the authors' scientific background and whether the data or program was made available. Again, we extracted the number of citations as well as the impact factor and the data is available in Supplementary Dataset 4.

## SOM: Additional Results

### Coverage of machine learning varies between journals and fields.

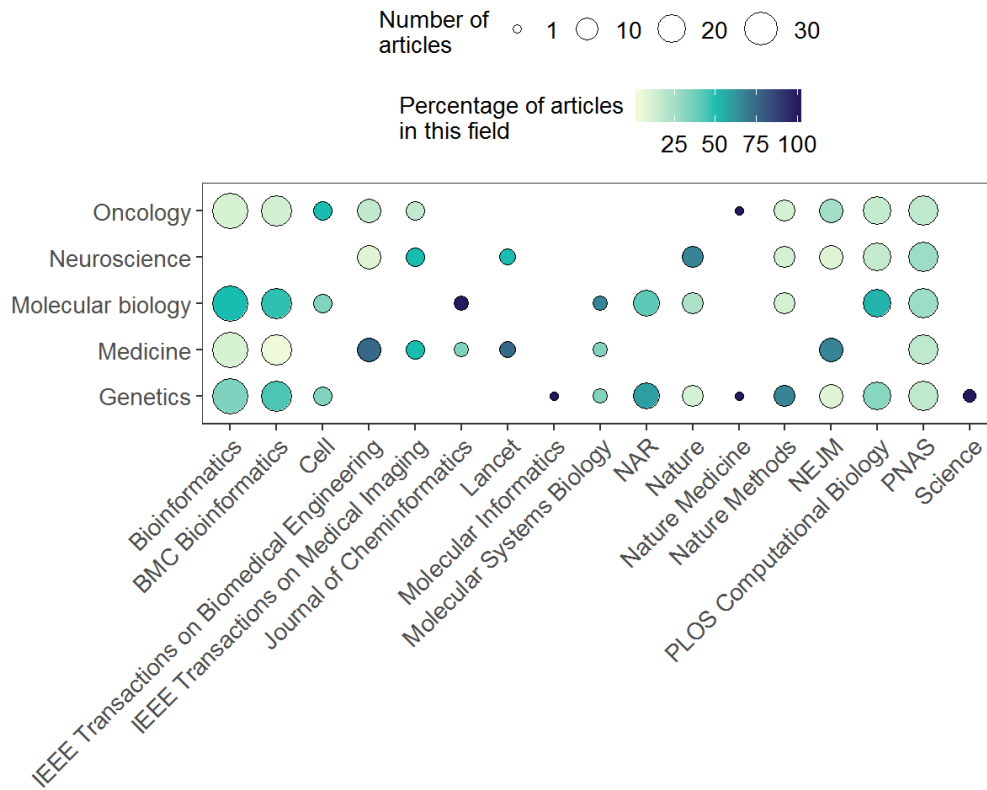
Most articles were cited fewer than 100 times, and the number of citations was proportional to time passed since publication (Spearman correlation coefficient  $\rho = -0.22$ ,  $p$ -value = 0.03; Fig. 1 main paper; Fig. S1). The average number of citations for articles from Nature and Science (2011-2016) showed the same trend as that for all 250 articles (Fig. S1). Since the time-dependency obfuscated inter-year comparisons, we adjusted by the number of years (*SOM Material & Methods*). As the number of citations correlated with the journal impact factor ( $\rho = 0.52$ ,  $p$ -value < 0.001, Fig. 1 main paper;), all aspects correlating with the impact factor trivially correlated with the number of citations. Normalizing by year and impact factor, removed this correlation. We continued also using the impact factor to assess the visibility of an article as publications in higher-ranked journals tend to be downloaded more often from bioRxiv<sup>4</sup>.

The number of articles differed highly between fields: the top five (*molecular biology* 26%, *genetics* 24%, *medicine* 14%, *oncology* 10% and *neuroscience* 9%) accounted for 76% of the 250 articles (Table S2, Fig. S2). Numbers varied even more by disciplines (author expertise): Computational scientists co-authored 88% of all articles, and 95% of those from *genetics* (Fig. S3). Biologists co-authored 70% of all and 59% in *medicine*. Physicians were primarily involved in articles from *medicine* and *oncology* (Fig. S3). Numbers of citations were largely similar for all fields (Fig. S4) but articles focusing on *medicine*, *neuroscience*, and *oncology* tended to be published in higher impact journals (Fig. S4).

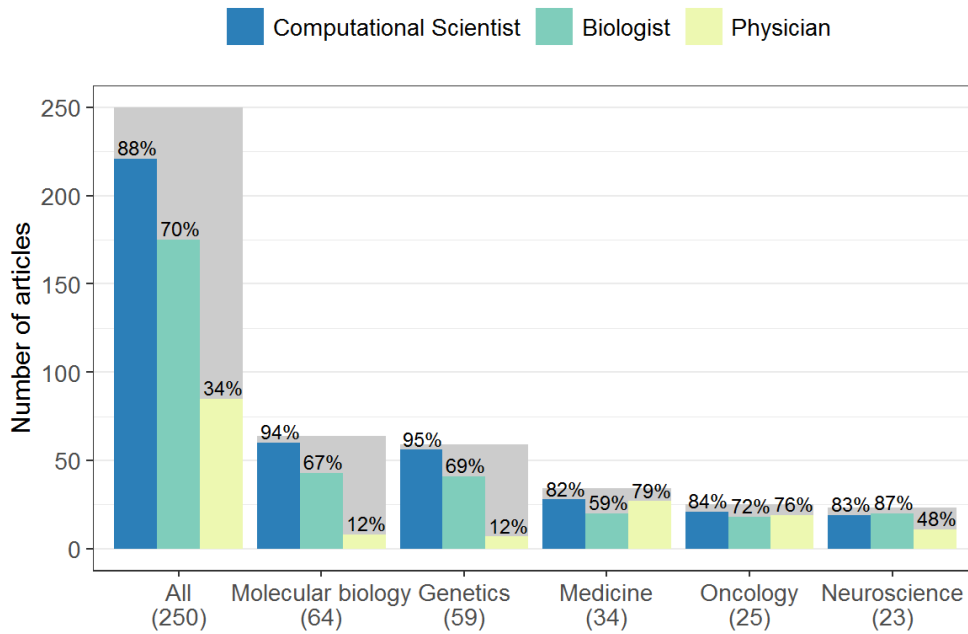


**Fig. S1: Citations per year and articles per journal per year** Overview of the distribution of articles across journals and citations across years in comparison with Nature and Science citations. **a.** Number of articles per journal per year. The number of articles varies highly between journals with the most articles contributed by the journal Bioinformatics and only one article contributed by the journal Nature Medicine. **b.** Violin plots with boxplots superimposed of number of citations per year. Articles are split by the year, they are published in. Older articles are on average more cited than newer articles, probably just because they have been available longer. Therefore, a higher number of citations does not necessarily imply a higher scientific relevance of the article. **c.** The increase of citations over time is not a trend specific to ML. The same holds true for all articles published in Nature and Science between 2011 and 2016.

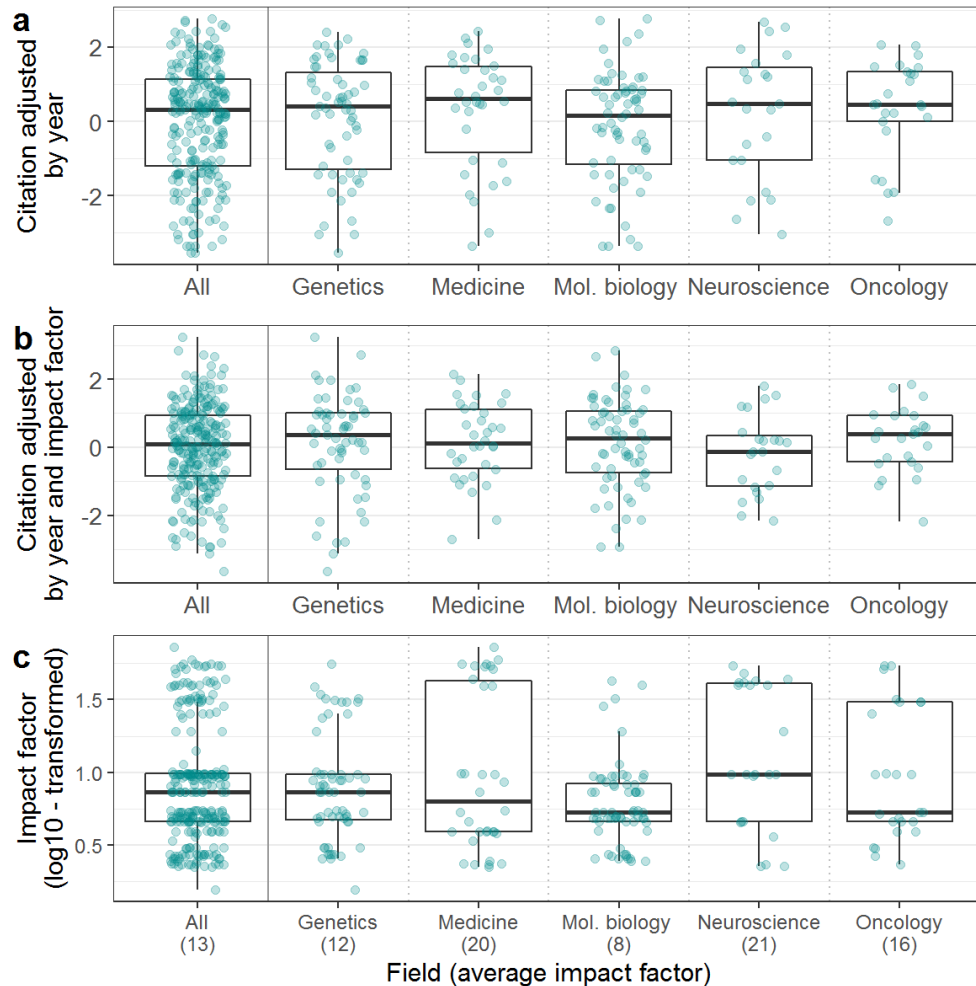




**Fig. S2: Fields represented in different journals** Certain fields are more common in some journals than in others. *Bioinformatics* focuses mainly on articles from the field of *genetics* or *molecular biology* while *New England Journal of Medicine* or *IEEE Transactions on Biomedical Engineering* primarily publish articles from medicine. The scope of a journal obviously influences whether an article from a certain field is more likely to get accepted or not.

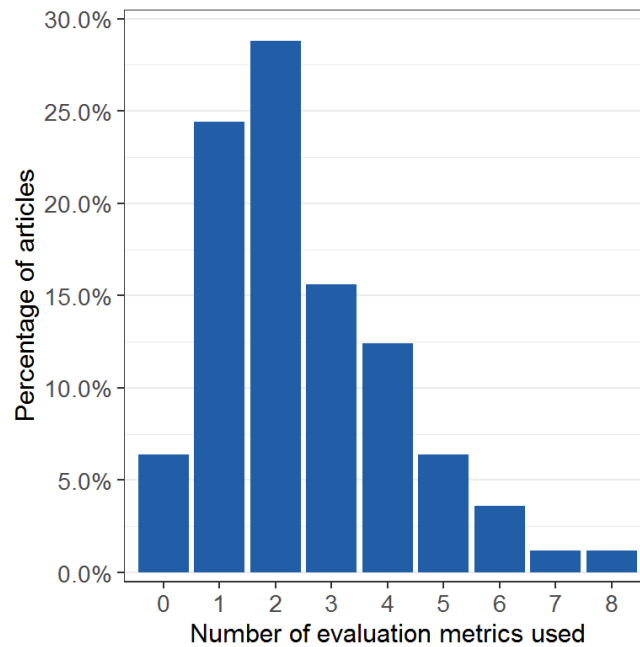


**Fig. S3: Authors' fields of expertise per field** Involvement of authors from different disciplines differs between fields. Computational scientists are involved in most of the articles with the highest involvement in articles from *genetics* and *molecular biology*. Not surprisingly, physicians are mainly involved in articles from *medicine* and *oncology* while only being rarely involved in articles from *genetics* or *molecular biology*.

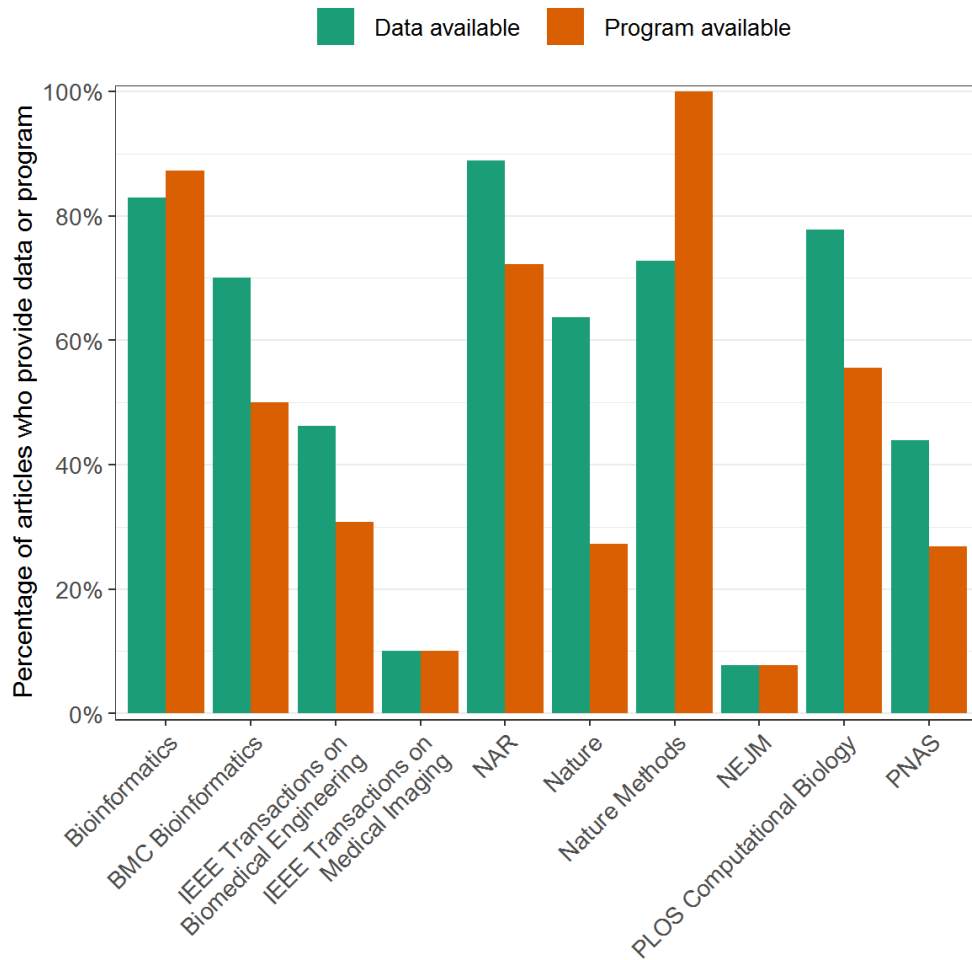


**Fig. S4: Adjusted number of citations and impact factor per field** Boxplots of adjusted citations and log<sub>10</sub>-transformed impact factor of 250 articles split by field. Vertical bars indicate largest (smallest) value within 1.5 times the interquartile range above (below) the third (first) quartile. We focus on the five major fields (*Molecular biology*, *genetics*, *medicine*, *oncology*, and *neuroscience*) covering 76% of all articles. The overall number of citations did not differ substantially between the five frequent fields, neither for the citations only adjusted by year nor the citations adjusted by both year and impact factor. **a.** Citations adjusted by year per field. **b.** Citations adjusted by year and impact factor per field. **c.** Impact factor per field on a log<sub>10</sub>-scale. The median impact factor was similar for all fields, but articles focusing on medicine, neuroscience, and oncology were, on average, published in journals with higher impact factors than articles from genetics and molecular biology.

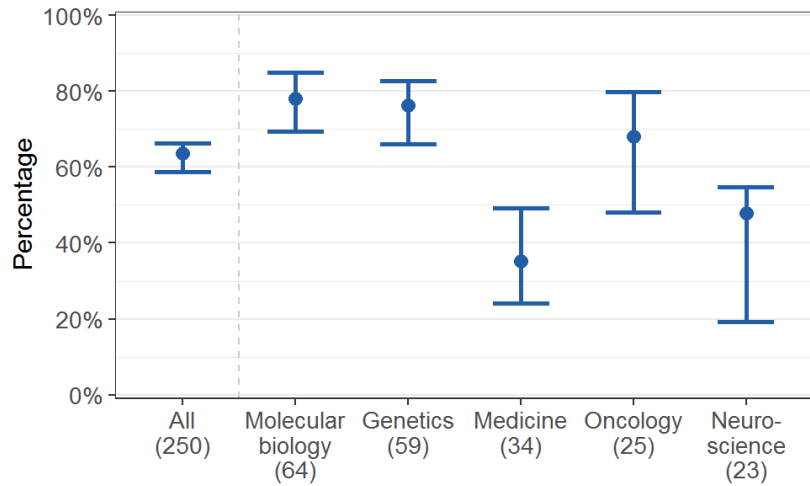
### Scientific validity higher with experts participating in collaboration



**Fig. S5: Number of evaluation metrics** One way to assess the validity of a method is achieving a certain performance for the given prediction task. This performance can be measured by a variety of evaluation metrics. The majority of articles (53%) apply one or two evaluation metrics, only 6% apply no metric, and also only 6% apply more than five metrics. While considering more evaluation metrics does not necessarily lead to a better assessment, applying at least two metrics is preferred to allow a valid assessment.

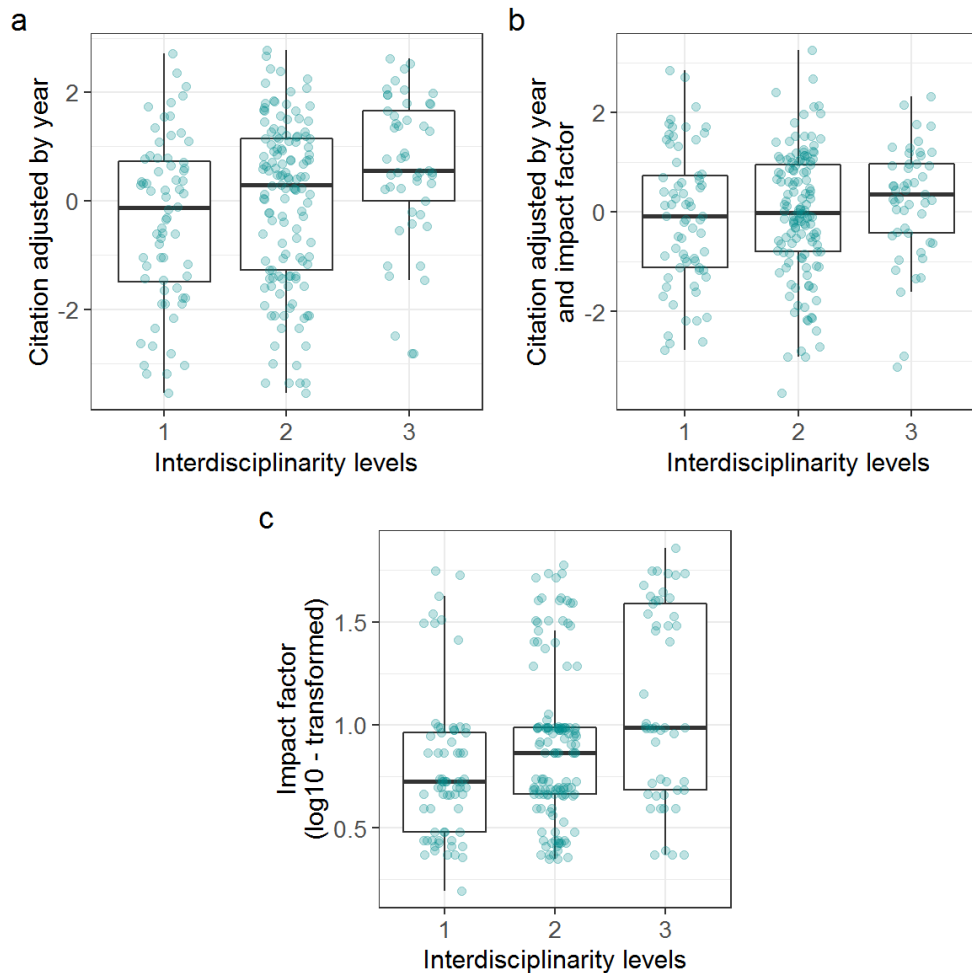


**Fig. S6: Percentage of articles sharing their data or program per journal** Considering journals with at least 10 articles in the data, we compare the percentage of those which provide their data or program. New England Journal of Medicine has the smallest fraction of articles sharing their data (8%) or program (8%). In Nature Methods all articles provide their program and 89% of articles published in NAR share their data.



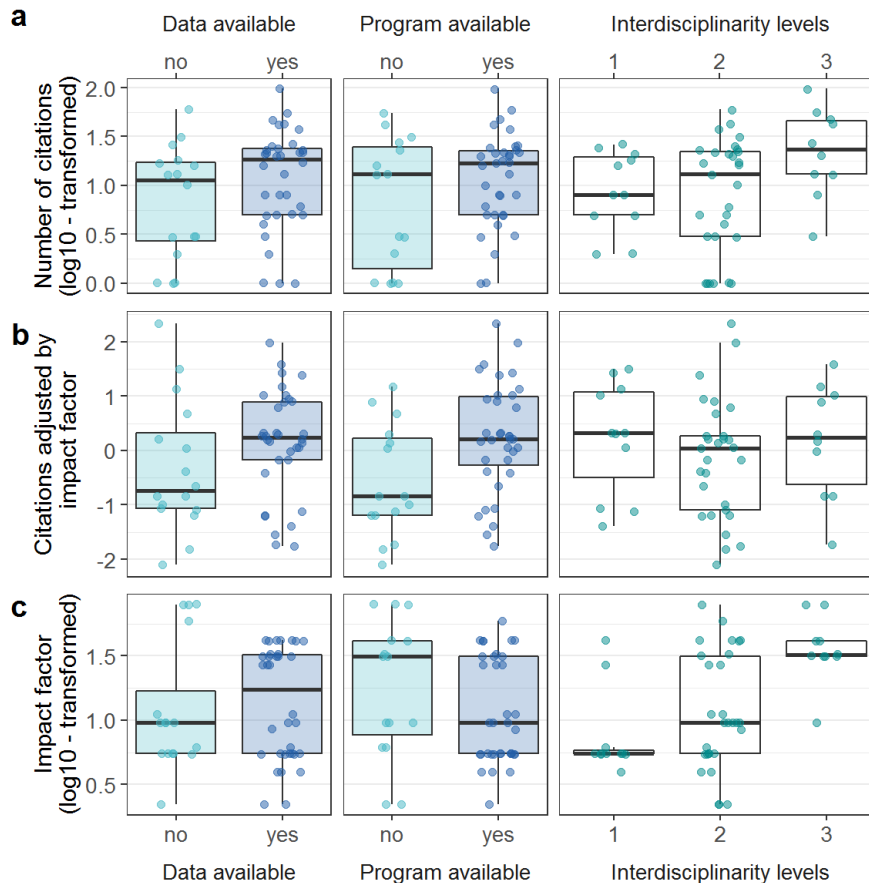
**Fig. S7: Data availability split by field** Percentages of 250 articles with data available split by field are shown with 95% percentile bootstrap confidence intervals based on 1000 bootstraps. The field of an article highly influences how often data is shared. Articles from *medicine* share data least often compared to other fields. This is probably caused by the fact that medical research often deals with sensitive patient data that cannot be shared publicly.

**Collaborations of scientists with different expertise somehow cited more often.**



**Fig. S8: Adjusted number of citations and impact factor for different collaborations**  
 Boxplots of adjusted citations and log<sub>10</sub>-transformed impact factor of 250 articles split by interdisciplinarity level. Vertical bars indicate largest (smallest) value within 1.5 times the interquartile range above (below) the third (first) quartile. **a.** The number of citations adjusted by year tended to be higher if researchers collaborated with other scientist outside their field of expertise. **b.** Adjusting also by impact factor led to smaller differences. This suggests that the higher number of citations for interdisciplinary teams was mainly caused by the fact that their work got accepted in higher-ranked journals. **c.** Articles written by authors of different specialities were on average published in journals with a higher impact factor than articles only written by authors from a single field of expertise.

## More computational scientists are involved for articles published in 2018.



**Fig. S9: Adjusted number of citations and impact factor for articles published in 2018 by data or program availability and for different collaborations** Boxplots of adjusted citations and log<sub>10</sub>-transformed impact factor of 250 articles split by data or program sharing or interdisciplinarity level. Vertical bars indicate largest (smallest) value with 1.5 times the interquartile range above (below) the third (first) quartile. **a.** The number of citations adjusted by year is not significantly different between articles sharing data and/or program and those that do not. As for articles from 2011 to 2016, citations are slightly higher for articles written by authors from different fields of expertise. **b.** Adjusting the citations also by impact factor reveals larger differences between the number of citations for sharing data and/or program and not sharing, but the differences are also not statistically significant. The number of citations adjusted by impact factor is similar independent of the number of different specialties among authors. **c.** It seems that articles sharing their data and not sharing their program are accepted to higher-ranked journals, but these differences are not statistically significant. Articles involving a large number of authors with different specialties are accepted in higher-ranked journals.



## SOM References

- 1 Coordinators, NR. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.* **44**, D7-19, doi:10.1093/nar/gkv1290 (2016).
- 2 Cock, PJ *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422-1423, doi:10.1093/bioinformatics/btp163 (2009).
- 3 Benjamini, Y & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)* **57** (1995).
- 4 Abdill, RJ & Blekhman, R. Meta-Research: Tracking the popularity and outcomes of all bioRxiv preprints. *Elife* **8**, doi:10.7554/eLife.45133 (2019).



# 3. Embeddings from Deep Learning

## Transfer GO Annotations beyond Homology

### 3.1. Preface

The Gene Ontology (GO) [8, 9] provides a standardized vocabulary to describe protein function in a human- and machine-readable format and separates different functional aspects into three hierarchies: Biological Process Ontology (BPO), Molecular Function Ontology (MFO), Cellular Component Ontology (CCO). A protein’s function can then be described by assigning certain GO terms to it. Experimentally verified GO annotations are not available for most proteins creating the need for prediction methods.

We developed *goPredSim*, a new and simple method to predict GO terms following a concept similar to homology-based inference: Annotations of an evolutionary related protein with annotations are transferred to a protein without known annotations. Instead of using sequence similarity to define evolutionary relation, our method relied on SeqVec embeddings [71] which were derived from language models adapting concepts from Natural Language Processing. We defined similarity between proteins as the Euclidean distance between the respective embeddings and considered two proteins as evolutionary related if their embeddings were similar.

The Critical Assessment of protein Function Annotation algorithms (CAFA) [25–27] is a community challenge taking place every two to three years to assess computational methods to predict GO terms. Replicating the conditions of CAFA3, which took place in 2017, *goPredSim* reached  $F_{max} = 37 \pm 2\%$ ,  $50 \pm 3\%$ , and  $57 \pm 2\%$  for BPO, MFO, and CCO, respectively. With this performance, our method would have been competitive

to the top ten CAFA3 competitors if we had participated. Preliminary evaluations for CAFA4 presented at ISMB2020 [114] supported those results. In addition, goPredSim clearly outperformed homology-based inference indicating that similarity between embeddings better captures functional similarity than sequence similarity does. Therefore, goPredSim is a simple yet effective method to predict GO terms, which is less complex than state-of-the-art methods and broader applicable than homology-based inference by allowing annotation transfer between more distantly related proteins. The method is available as a standalone web server (<https://embed.protein.properties/>) and as part of PredictProtein [68, 115].

**Author contribution:** I implemented the method goPredSim and performed evaluations. Michael Heinzinger provided SeqVec and ProtBERT embeddings. Christian Dallago implemented the web server. Tobias Olenyi computed the combination of sequence- and embedding-based transfer. All authors drafted the manuscript.


### 3.2. Journal Article: Littmann, Heinzinger *et al.*, Scientific Reports (2021)

**Reference:** Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., and Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific Reports*, 11(1):2045–2322, 2021. doi:10.1038/s41598-020-80786-0

**Copyright Notice:** Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>).



# OPEN Embeddings from deep learning transfer GO annotations beyond homology

Maria Littmann<sup>1,2,6</sup>, Michael Heinzinger<sup>1,2,6</sup>, Christian Dallago<sup>1,2</sup>, Tobias Olenyi<sup>1</sup> & Burkhard Rost<sup>1,3,4,5</sup>

Knowing protein function is crucial to advance molecular and medical biology, yet experimental function annotations through the Gene Ontology (GO) exist for fewer than 0.5% of all known proteins. Computational methods bridge this sequence-annotation gap typically through homology-based annotation transfer by identifying sequence-similar proteins with known function or through prediction methods using evolutionary information. Here, we propose predicting GO terms through annotation transfer based on proximity of proteins in the SeqVec embedding rather than in sequence space. These embeddings originate from deep learned language models (LMs) for protein sequences (SeqVec) transferring the knowledge gained from predicting the next amino acid in 33 million protein sequences. Replicating the conditions of CAFA3, our method reaches an  $F_{\max}$  of  $37 \pm 2\%$ ,  $50 \pm 3\%$ , and  $57 \pm 2\%$  for BPO, MFO, and CCO, respectively. Numerically, this appears close to the top ten CAFA3 methods. When restricting the annotation transfer to proteins with  $<20\%$  pairwise sequence identity to the query, performance drops ( $F_{\max}$  BPO  $33 \pm 2\%$ , MFO  $43 \pm 3\%$ , CCO  $53 \pm 2\%$ ); this still outperforms naïve sequence-based transfer. Preliminary results from CAFA4 appear to confirm these findings. Overall, this new concept is likely to change the annotation of proteins, in particular for proteins from smaller families or proteins with intrinsically disordered regions.

## Abbreviations

BERT	Bidirectional Encoder Representations from Transformers (particular deep learning language model)
BP(O)	Biological process (ontology) from GO
CAFA	Critical Assessment of Functional Annotation
CC(O)	Cellular component (ontology) from GO
ELMo	Embeddings from Language Models
GO	Gene ontology
GOA	Gene Ontology Annotation
k-NN	K-nearest neighbor
LK	Limited-knowledge
LM	Language model
LSTMs	Long-short-term-memory cells
M	Million
MF(O)	Molecular function (ontology) from GO
NK	No-knowledge
PIDE	Percentage pairwise sequence identity
RI	Reliability index
RMSD	Root-mean-square deviation

<sup>1</sup>Department of Informatics, Bioinformatics and Computational Biology, i12, TUM (Technical University of Munich), Boltzmannstr. 3, Garching, 85748 Munich, Germany. <sup>2</sup>TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany. <sup>3</sup>Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, Garching, 85748 Munich, Germany. <sup>4</sup>School of Life Sciences Weihenstephan (TUM-WZW), TUM (Technical University of Munich), Alte Akademie 8, Freising, Germany. <sup>5</sup>Department of Biochemistry and Molecular Biophysics, Columbia University, 701 West, 168th Street, New York, NY 10032, USA. <sup>6</sup>These authors contributed equally: Maria Littmann and Michael Heinzinger. ✉email: littmann@rostlab.org

**GO captures cell function through hierarchical ontologies.** All organisms rely on the correct functioning of their cellular workhorses, namely their proteins involved in almost all roles, ranging from molecular functions (MF) such as chemical catalysis of enzymes to biological processes or pathways (BP), e.g., realized through signal transduction. Only the perfectly orchestrated interplay between proteins allows cells to perform more complex functions, e.g., the aerobic production of energy via the citric acid cycle requires the interconnection of eight different enzymes with some of them being multi-enzyme complexes<sup>1</sup>. The Gene Ontology (GO)<sup>2</sup> thrives to capture this complexity and to standardize the vocabulary used to describe protein function in a human- and machine-readable manner. GO separates different aspects of function into three hierarchies: MFO (Molecular Function Ontology), BPO (biological process ontology), and CCO, i.e. the cellular component(s) or subcellular localization(s) in which the protein acts.

**Computational methods bridge the sequence-annotation gap.** As the experimental determination of complete GO numbers is challenging, the gap between the number of proteins with experimentally verified GO numbers and those with known sequence but unknown function (sequence-annotation gap) remains substantial. For instance, UniRef100<sup>3</sup> (UniProt<sup>3</sup> clustered at 100% percentage pairwise sequence identity, PIDE) contains roughly 220 M (million) protein sequences of which fewer than 1 M have annotations verified by experts (Swiss-Prot<sup>3</sup> evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC).

Computational biology has been bridging the sequence-annotation gap for decades<sup>4–11</sup>, based on two different concepts: (1) sequence similarity-based transfer (or *homology-based inference*) which copies the annotation from one protein to another if that is sequence similar enough because proteins of similar sequence have similar function<sup>12</sup>. In more formal terms: given a query Q of unknown and a template T of known function (F<sub>T</sub>): IF PIDE(Q,T) > threshold  $\theta$ , transfer annotation F<sub>T</sub> to Q. (2) *De-novo* methods predict protein function through machine learning<sup>5</sup>. If applicable, the first approach tends to out-perform the second<sup>13–16</sup> although it largely misses discoveries<sup>17</sup>. The progress of computational methods has been monitored by CAFA (*Critical Assessment of Functional Annotation*)<sup>9,18,19</sup>, an international collaboration for advancing and assessing methods that bridge the sequence-annotation gap. CAFA takes place every 2–3 years with its fourth instance (CAFA4) currently being evaluated.

Here, we introduce a novel approach transferring annotations using the similarity of embeddings from language models (LMs: SeqVec<sup>20</sup> and ProtBert<sup>21</sup>) rather than the similarity of sequence. Using embedding space proximity has helped information retrieval in natural language processing (NLP)<sup>22–25</sup>. By learning to predict the next amino acid given the entire previous sequence on unlabeled data (only sequences without any phenotype/label), e.g., SeqVec learned to extract features describing proteins useful as input to different tasks (transfer learning). Instead of transferring annotations from the labeled protein T with the highest percentage pairwise sequence identity (PIDE) to the query Q, we chose T as the protein with the smallest distance in embedding space (DIST<sup>emb</sup>) to Q. This distance also proxied the reliability of the prediction serving as threshold above which hits are considered too distant to infer annotations. Instead of picking the top hit, annotations can be inferred from the *k* closest proteins where *k* has to be optimized. In addition, we evaluate the influence of the type of LM used (SeqVec<sup>20</sup> vs. ProtBert<sup>21</sup>). Although the LMs were never trained on GO terms, we hypothesize LM embeddings to implicitly encode information relevant for the transfer of annotations, i.e., capturing aspects of protein function because embeddings have been shown to capture rudimentary features of protein structure and function<sup>20,21,26,27</sup>.

## Results and discussion

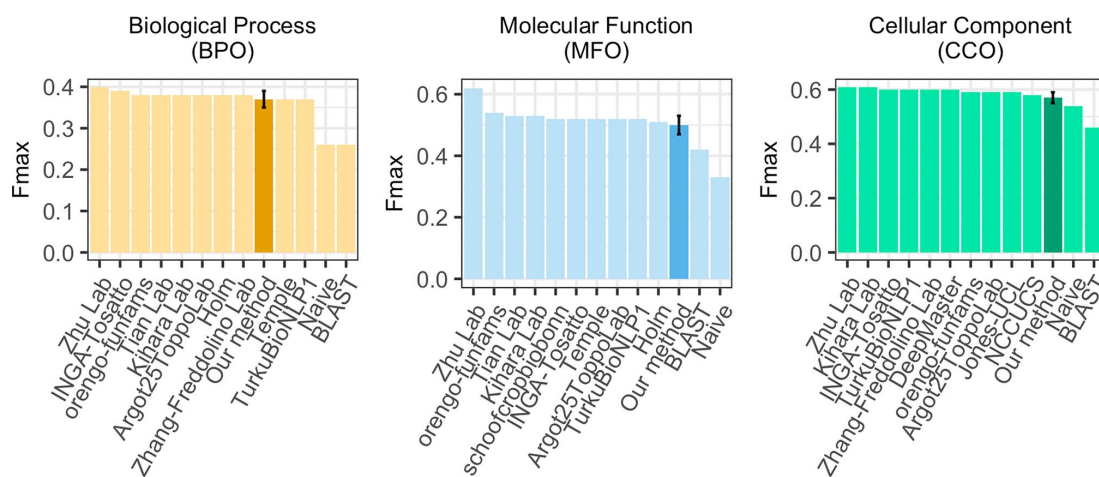
**Simple embedding-based transfer almost as good as CAFA3 top-10.** First, we predicted GO terms for all 3328 CAFA3 targets using the Gene Ontology Annotation (GOA) data set GOA2017 (Methods), removed all entries identical to CAFA3 targets (PIDE = 100%; set: GOA2017-100) and transferred the annotations of the closest hit (*k* = 1; closest by Euclidean distance) in this set to the query. When applying the *NK evaluation mode* (no-knowledge available for query, Methods/CAFA3), the embedding-transfer reached F<sub>max</sub> scores (Eq. 3) of 37 ± 2% for BPO (precision: P = 39 ± 2%, recall: R = 36 ± 2%, Eqs. 1/2), F1 = 50 ± 3% for MFO (P = 54 ± 3%, R = 47 ± 3%), and F1 = 57 ± 2% for CCO (P = 61 ± 3%, R = 54 ± 3%; Table 1, Fig. 1, Fig. S1). Errors were estimated through the 95% confidence intervals (± 1.96 stderr). Replacing the Euclidean by cosine distance (more standard amongst those working with embeddings, e.g., in NLP) changed little (Table 1; for simplicity, we only used Euclidean from here on). In the sense that the database with annotations to transfer (GOA2017) had been available before the CAFA3 submission deadline (February 2017), our predictions were directly comparable to CAFA3<sup>19</sup>. This embedding-based annotation transfer clearly outperformed the two CAFA3 baselines (Fig. 1: the simple *BLAST* for sequence-based annotation transfer, and the *Naïve* method assigning GO terms statistically based on database frequencies, here GOA2017); it would have performed close to the top ten CAFA3 competitors (in particular for BPO: Fig. 1) had the method competed at CAFA3.

Performance did not change when replacing global average with maximum pooling (Table 1). While averaging over long proteins could lead to information loss in the resulting embeddings, we did not observe a correlation between performance and protein length (Fig. S2, Table S1). In order to obtain the embeddings, we processed query and lookup protein the same way. If those have similar function and similar length, their embeddings might have lost information in the same way. This loss might have “averaged out” to generate similar embeddings.

Including more neighbors (*k* > 1) only slightly affected F<sub>max</sub> (Table S2; all F<sub>max</sub> averages for *k* = 2 to *k* = 10 remained within the 95% confidence interval of that for *k* = 1). When taking all predictions into account independent of a threshold in prediction strength referred to as the reliability index (RI, Methods; i.e., even low confidence annotations are transferred), the number of predicted GO terms increased with higher *k* (Table S3). The average number of GO terms annotated per protein in GOA2017 already reached 37, 14, 9 for BPO, MFO, CCO, respectively. When including all predictions independent of their strength (RI) our method predicted more

Data set	Embeddings	$F_{max}$		
		BPO	MFO	CCO
GOA2017	SeqVec	37 ± 2%	50 ± 3%	57 ± 2%
	SeqVec (Cosine)	37 ± 2%	50 ± 3%	58 ± 2%
	SeqVec (maximum pooling)	35 ± 2%	52 ± 3%	58 ± 2%
	ProtBert	36 ± 2%	49 ± 3%	59 ± 2%
	BLAST	26%	42%	46%
GOA2017X	SeqVec	31 ± 2%	51 ± 3%	56 ± 2%
GOA2020	SeqVec	51 ± 2%	61 ± 3%	65 ± 2%
	ProtBert	50 ± 2%	59 ± 2%	65 ± 2%
	BLAST	31%	53%	58%

**Table 1.** Performance for CAFA3 targets for simple GO annotation transfers\*. \*Mean  $F_{max}$  values for GO term predictions using embeddings from two different language models (*SeqVec* or *ProtBert*) or sequence similarity (*BLAST*) for the data sets *GOA2017-100* (2017), *GOA2017X* (2017), and *GOA2020-100* (2020) used for annotation transfer (note: the notation ‘-100’ implies that any entry in the data set with PIDE = 100% to any CAFA3 protein had been removed). By default, embedding distance was assessed by Euclidean distance (Eq. 4; exception marked *cosine*), and per-residue embeddings were pooled by global average pooling (exception marked maximum pooling). All values were compiled for picking the single top hit ( $k=1$ ) and using the CAFA3 targets from the NK and full evaluation mode<sup>19</sup>. For all simple annotation transfers (embedding- and sequence-based), performance was higher for the more recent data sets (*GOA2020* vs. *GOA2017*). Error estimates are given as 95% confidence intervals.  $F_{max}$  values were computed using the CAFA3 tool<sup>18,19</sup>.



**Figure 1.**  $F_{max}$  for simplest embedding-based transfer ( $k=1$ ) and CAFA3 competitors. Using the data sets and conditions from CAFA3, we compared the  $F_{max}$  of the simplest implementation of the embedding-based annotation transfer, namely the greedy ( $k=1$ ) solution in which the transfer comes from exactly one closest database protein (dark bar) for the three ontologies (BPO, MFO, CCO) to the top ten methods that—in contrast to our method—did compete at CAFA3 and to two background approaches “BLAST” (homology-based inference) and “Naive” (assignment of terms based on term frequency) (lighter bars). The result shown holds for the NK evaluation mode (no knowledge), i.e., only using proteins that were novel in the sense that they had no prior annotations. If we had developed our method before CAFA3, it would have almost reached the tenth place for MFO and CCO, and ranked even slightly better for BPO. Error bars (for our method) marked the 95% confidence intervals.

terms for CCO and BPO than expected from this distribution even for  $k=1$ . Only for MFO the average (11.7 terms) predicted was slightly lower than expected for  $k=1$  (number of terms exploded for  $k>1$ ; Table S3). While precision dropped with adding terms, recall increased (Table S3). To avoid overprediction and given that  $k$  hardly affected  $F_{max}$ , we chose  $k=1$ . This choice might not be best in practice: considering more than one hit ( $k>1$ ) might help when the closest hit only contains unspecific terms. However, such a distinction will be left to expert users.

When applying the *LK evaluation mode* (limited-knowledge, i.e., query already has some annotation about function, Methods/CAFA3), the embedding-based annotation transfer reached  $F_{max}$  scores of  $49 \pm 1\%$ ,  $52 \pm 2\%$ ,

	<i>PIDE</i>	$d_{\text{SeqVec}}$	$d_{\text{ProtBert}}$
<i>PIDE</i>	1	0.293	0.248
$d_{\text{SeqVec}}$		1	0.576
$d_{\text{ProtBert}}$			1

**Table 2.** Embedding and sequence similarity correlated\*. \**PIDE* percentage pairwise sequence identity,  $d_{\text{SeqVec}}$  similarity in SeqVec<sup>20</sup> embeddings (Eq. 5);  $d_{\text{ProtBert}}$  similarity in ProtBert<sup>21</sup> embeddings (Eq. 5). All values represent Spearman's correlation coefficients calculated for 434,001 sequence pairs. For all pairs, the significance was  $p$ -value  $< 2.2e-16$ , i.e., significant at the level of the precision of the software R<sup>28</sup>. The similarity between sequence and embeddings correlated less than the two different types of embeddings, namely SeqVec and ProtBert with each other. In order to highlight the trivial symmetry of the matrix, only the upper diagonal was given.

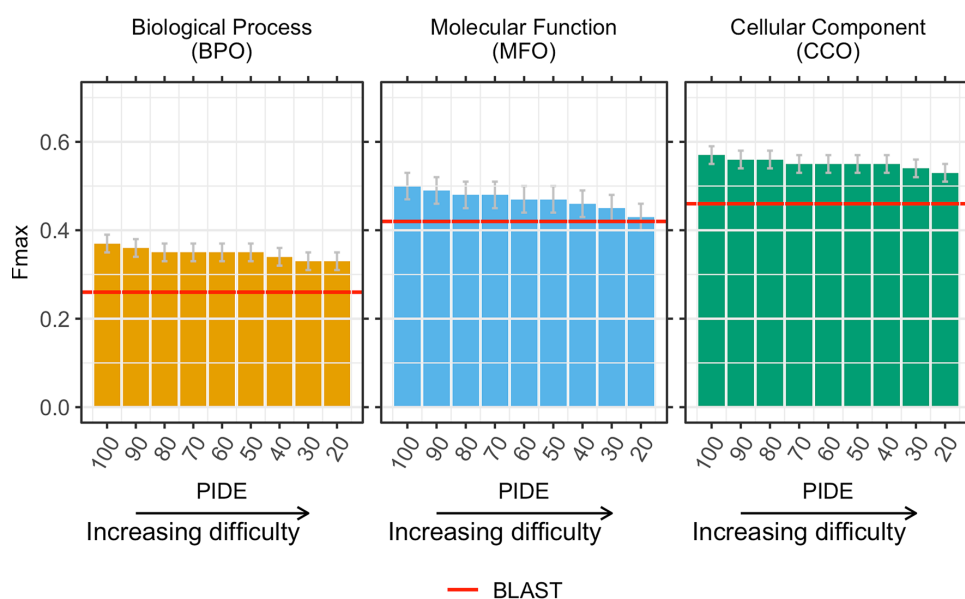
and  $58 \pm 3\%$  for BPO, MFO, and CCO, respectively (Fig. S3). Thus, the embedding-based annotation transfer reached higher values for proteins with prior annotations (LK evaluation mode) than for novel proteins without any annotations (NK evaluation mode; Table 1); the same was true for the CAFA3 top-10 for which the  $F_{\text{max}}$  scores increased even more than for our method for BPO and MFO, and less for CCO (Fig. 1, Fig. S3). In the LK mode, predictions are evaluated for proteins for which 1–2 GO ontologies had annotations while those for another ontology (or two) were added after the CAFA3 deadline<sup>9,19</sup>. While supervised training uses such labels; our approach did not since we had excluded all CAFA3 targets explicitly from the annotation transfer database (GOA2017). Thus, our method could not benefit from previous annotations, i.e., LK and NK should be identical. The observed differences were most likely explained by how  $F_{\text{max}}$  is computed. The higher  $F_{\text{max}}$  score, especially for BPO, might be explained by data set differences, e.g. LK average number of BPO annotations was 19 compared to 26 for NK. Other methods might have reached even higher by training on known annotations.

**Embedding-based transfer successfully identified distant relatives.** Embeddings condense information learned from sequences; identical sequences produce identical embeddings: if  $\text{PIDE}(Q,T) = 100\%$ , then  $\text{DIST}^{\text{emb}}(Q,T) = 0$  (Eq. 4). We had assumed a simple relation: the more similar two sequences, the more similar their embeddings because the underlying LMs only use sequences as input. Nevertheless, we observed embedding-based annotation transfer to outperform (higher  $F_{\text{max}}$ ) sequence-based transfer (Table 1, Fig. 1). This suggested embeddings to capture information beyond raw sequences. Explicitly calculating the correlation between sequence and embedding similarity for 431,224 sequence pairs from CAFA3/GOA2017-100, we observed a correlation of  $\rho = 0.29$  (Spearman's correlation coefficient,  $p$ -value  $< 2.2e-16$ ; Table 2). Thus, sequence and embedding similarity correlated at an unexpectedly low level. However, our results demonstrated that embedding similarity identified more distant relatives than sequence similarity (Figs. S1, S4).

In order to quantify how different embeddings for proteins Q and T can still share GO terms, we redundancy reduced the GOA2017 database used for annotation transfers at distance thresholds of decreasing *PIDE* with respect to the queries (in nine steps from 90 to 20%, Table S6). By construction, all proteins in GOA2017-100 had *PIDE*  $< 100\%$  to all CAFA3 queries (Q). If the pair (Q,T) with the most similar embedding was also similar in sequence, embedding-based would equal sequence-based transfer. At lower *PIDE* thresholds, e.g., *PIDE*  $< 20\%$ , reliable annotation transfer through simple pairwise sequence alignment is no longer possible<sup>14,29-32</sup>. Although embeddings-based transfer tended to be slightly less successful for pairs with lower *PIDE* (Fig. 2: bars decrease toward right), the drop appeared small; on top, at almost all *PIDE* values, embedding-transfer remained above BLAST, i.e., sequence-based transfer (Fig. 2: most bars higher than reddish line – error bars show 95% confidence intervals). The exception was for MFO at *PIDE*  $< 30\%$  and *PIDE*  $< 20\%$  for which the  $F_{\text{max}}$  scores from sequence-based transfer (BLAST) were within the 95% confidence interval (Fig. 2). This clearly showed that our approach benefited from information available through embeddings but not through sequences, and that at least some protein pairs close in embedding and distant in sequence space might function alike. In order to correctly predict the next token, protein LMs have to learn complex correlations between residues as it is impossible to remember the multitude of all possible amino acid combinations in hundreds of millions to billions of protein sequences. This forces models to abstract higher level features from sequences. For instance, secondary structure can directly be extracted from embeddings through linear projection<sup>26</sup>. The LMs (SeqVec & ProtBert) might even have learned to find correlations between protein pairs diverged into the “midnight zone” sequence comparison in which sequence similarity becomes random<sup>29,53</sup>. Those cases are especially difficult to detect by the most successful search methods such as BLAST<sup>34</sup> or MMseqs2<sup>35</sup> relying on finding similar seeds missing at such diverged levels.

Ultimately, we failed to really explain why the abstracted level of sequences captured in embeddings outperformed raw sequences. One attempt at addressing this question led to displaying cases for which one of the two worked better (Fig. S5). Looking in more detail at outliers (embeddings more similar than sequences), we observed that embedding-based inference tended to identify more reasonable hits in terms of lineage or structure. For instance, for the uncharacterized transporter YIL166C (UniProt identifier P40445) from *Saccharomyces cerevisiae* (baker's yeast), the closest hit in SeqVec embedding space was the high-affinity nicotinic acid transporter (P53322) also from *Saccharomyces cerevisiae*. Both proteins belong to the allantoin permease family while the most sequence-similar hit (with *PIDE* = 31%) was the gustatory and odorant receptor 22 (Q7PMG3) from the insect *Anopheles gambiae* belonging to the gustatory receptor family. Experimental 3D structures were





**Figure 2. Embedding-based transfer succeeded for rather diverged proteins.** After establishing the low correlation between embedding- and sequence-similarity, we tested how the level of percentage pairwise sequence identity (PIDE, x-axes) between the query (protein without known GO terms) and the transfer database (proteins with known GO terms, here subsets of *GOA2017*) affected the performance of the embedding-based transfer. Technically, we achieved this by removing proteins above a certain threshold in PIDE (decreasing toward right) from the transfer database. The y-axes showed the  $F_{\max}$  score as compiled by CAFA3<sup>19</sup>. If embedding similarity and sequence identity correlated, our method would fall to the level of the reddish lines marked by BLAST. On the other hand, if the two were completely uncorrelated, the bars describing embedding-transfer would all have the same height (at least within the standard errors marked by the gray vertical lines at the upper end of each bar), i.e., embedding-based transfer would be completely independent of the sequence similarity between query and template (protein of known function). The observation that all bars tended to fall toward the right implied that embedding and sequence similarity correlated (although for CCO,  $F_{\max}$  remained within the 95% confidence interval of  $F_{\max}$  for *GOA2017-100*). The observation that our method remained mostly above the baseline predictions demonstrates that embeddings capture important orthogonal information. Error bars indicate 95% confidence intervals.

not available for any of the three proteins. However, comparative modeling using Swiss-Model<sup>36</sup> revealed that both the target and the hit based on SeqVec were mapped to the same template (root-mean-square deviation (RMSD) = 0.3 Å) (Fig. S6a) while the hit based on sequence similarity was linked to a different structure (with RMSD = 16.8 Å) (Fig. S6b). Similarly, for the GDSL esterase/lipase At3g48460 (Q9STM6) from *Arabidopsis thaliana*, the closest hit in ProtBert embedding space was the GDSL esterase/lipase 2 (Q9SYF0) also from *Arabidopsis thaliana* while the most sequence-similar hit was the UDP-glucose 4-epimerase (Q564Q1) from *Caenorhabditis elegans*. The target and the embedding-based hit are both hydrolases belonging to the same CATH superfamily while the sequence-based hit is an isomerase and not annotated to any CATH superfamily. Comparative modeling suggested similar structures for target and embedding hit (RMSD = 2.9 Å) (Fig. S6c) while the structure found for the sequence-based hit similarity was very different (RMSD = 26 Å) (Fig. S6d). This suggested embeddings to capture structural features better than just raw sequences. Homology-based inference depends on many parameters that can especially affect the resulting sequence alignment for distantly related proteins. Possibly, embeddings are more robust in identifying those more distant evolutionary relatives.

**Embedding-based transfer benefited from non-experimental annotations.** Unlike the data set made available by CAFA3, annotations in our *GOA2017* data set were not limited to experimentally verified annotations. Instead, they included annotations inferred by computational biology, homology-based inference, or by “author statement evidence”, i.e., through information from publications. Using *GOA2017X*, the subset of *GOA2017-100* containing only experimental terms, our method reached  $F_{\max} = 31 \pm 2\%$  ( $P = 28 \pm 2\%$ ,  $R = 34 \pm 2\%$ ),  $51 \pm 3\%$  ( $P = 53 \pm 3\%$ ,  $R = 49 \pm 3\%$ ), and  $56 \pm 2\%$  ( $P = 55 \pm 3\%$ ,  $R = 57 \pm 3\%$ ) for BPO, MFO, and CCO, respectively. Compared to using *GOA2017-100*, the performance dropped significantly for BPO ( $F_{\max} = 37 \pm 2\%$  for *GOA2017-100*, Table 1); it decreased slightly (within 95% confidence interval) for CCO ( $F_{\max} = 57 \pm 2\%$  for *GOA2017-100*, Table 1); and it increased slightly (within 95% confidence interval) for MFO ( $F_{\max} = 50 \pm 3\%$  for *GOA2017-100*, Table 1). Thus, less reliable annotations might still help, in particular for BPO. Annotations for

BPO may rely more on information available from publications that is not as easily quantifiable experimentally as annotations for MFO or CCO.

Many of the non-experimental annotations constituted sequence-based annotation transfers. Thus, non-experimental annotations might have helped because they constituted an implicit merger of sequence and embedding transfer. Adding homologous proteins might “bridge” sequence and embedding space by populating embedding space using annotations transferred from sequence space. The weak correlation between both spaces supported this speculation because protein pairs with very similar sequences may differ in their embeddings and vice versa.

**Improving annotations from 2017 to 2020 increased performance significantly.** For CAFA3 comparisons, we only used data available before the CAFA3 submission deadline. When running new queries, annotations will be transferred from the latest GOA. We used *GOA2020-100* (from 02/2020 removing the CAFA3 targets) to assess how the improvement of annotations from 2017 to 2020 influenced annotation transfer (Table 1). On *GOA2020-100*, SeqVec embedding-based transfer achieved  $F_{\max} = 50 \pm 2\%$  ( $P = 50 \pm 3\%$ ,  $R = 50 \pm 3\%$ ),  $60 \pm 3\%$  ( $P = 52 \pm 3\%$ ,  $R = 71 \pm 3\%$ ), and  $65 \pm 2\%$  ( $P = 57 \pm 3\%$ ,  $R = 75 \pm 3\%$ ) for BPO, MFO, and CCO, respectively, for the NK evaluation mode (Table 1). This constituted a substantial increase over *GOA2017-100* (Table 1).

The large performance boost between *GOA2017* and *GOA2020* suggested the addition of many relevant GO annotations. However, for increasingly diverged pairs (Q,T), we observed a much larger drop in  $F_{\max}$  than for *GOA2017* (Fig. 2, Fig. S4). In the extreme, *GOA2020-20* ( $\text{PIDE}(Q,T) < 20\%$ ) with  $F_{\max} = 33 \pm 2\%$  (BPO),  $44 \pm 2\%$  (MFO), and  $54 \pm 2\%$  (CCO) fell to the same level as *GOA2017-20* (Figs. 2, S4). These results suggested that many of the relevant GO annotations were added for proteins sequence-similar to those with existing annotations. Put differently, many helpful new experiments simply refined previous computational predictions.

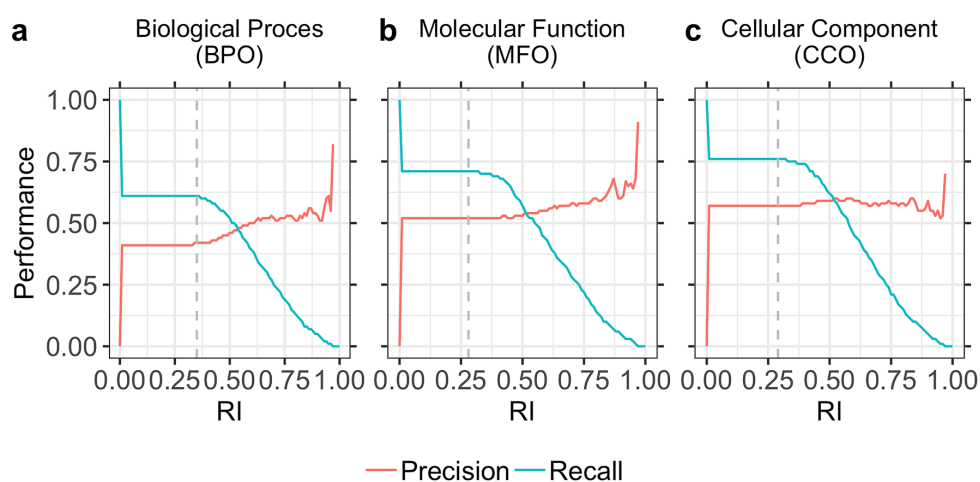
Running BLAST against *GOA2020-100* for sequence-based transfer (choosing the hit with the highest PIDE) showed that sequence-transfer also profited from improved annotations (difference in  $F_{\max}$  values for BLAST in Table 1). However, while  $F_{\max}$  scores for embedding-based transfer increased the most for BPO, those for sequence-based transfer increased most for MFO. Embedding-transfer still outperformed BLAST for the *GOA2020-100* set (Fig. S4c).

Even when constraining annotation transfer to sequence-distant pairs, our method outperformed BLAST against *GOA2020-100* in terms of  $F_{\max}$  at least for BPO and for higher levels of PIDE in MFO/CCO (Fig. S4c). However, comparing the results for BLAST on the *GOA2020-100* set with the performance of our method for subsets of very diverged sequences (e.g.  $\text{PIDE} < 40\%$  for *GOA2020-40*) under-estimated the differences between sequence- and embedding-based transfer, because the two approaches transferred annotations from different data sets. For a more realistic comparison, we re-ran BLAST only considering hits below certain PIDE thresholds (for comparability we could not do this for CAFA3). As expected, performance for BLAST decreased with PIDE (Fig. S4 lighter bars), e.g., for  $\text{PIDE} < 20\%$ ,  $F_{\max}$  fell to 8% for BPO, 10% for MFO, and 11% for CCO (Fig. S4 lighter bars) largely due to low coverage, i.e., most queries had no hit to transfer annotations from. At this level (and for the same set), the embedding-based transfer proposed here, still achieved values of  $33 \pm 2\%$  (BPO),  $44 \pm 2\%$  (MFO), and  $54 \pm 2\%$  (CCO). Thus, our method made reasonable predictions at levels of sequence identity for which homology-based inference (BLAST) failed completely.

**Performance confirmed by new proteins.** Our method and especially the threshold to transfer a GO term were “optimized” using the CAFA3 targets. Without any changes in the method, we tested a new data set of 298 proteins, *GOA2020-new*, with proteins for which experimental GO annotations have been added since the CAFA4 submission deadline (02/2020; Method). Using the thresholds optimized for CAFA3 targets (0.35 for BPO, 0.28 for MFO, 0.29 for CCO, Fig. 3), our method reached  $F_i = 50 \pm 11\%$ ,  $54 \pm 5\%$ , and  $66 \pm 8\%$  for BPO, MFO, and CCO, respectively. For BPO and CCO, the performance was similar to that for the CAFA3 targets; for MFO it was slightly below but within the 95% CI (Table 1). For yet a different set, submitted for MFO to CAFA4, the first preliminary evaluation published during ISMB2020<sup>37</sup>, also suggested our approach to make it to the top-ten, in line with the *post facto* CAFA3 results presented here.

**Embedding similarity influenced performance.** Homology-based inference works best for pairs with high PIDE. Analogously, we assumed embedding-transfer to be best for pairs with high embedding similarity, i.e., low Euclidean distance (Eq. 4). We used this to define a reliability index (RI, Eq. 5). For the *GOA2020-100* set, the minimal RI was 0.24. The CAFA evaluation determined 0.35 for BPO, 0.28 for MFO, and 0.29 for CCO as thresholds leading to optimal performance as measured by  $F_{\max}$  (Fig. 3 dashed grey lines marked these thresholds). For all ontologies, precision and recall were almost constant for lower RIs (up to ~0.3). For higher RIs, precision increased, and recall decreased as expected (Fig. 3). While precision increased up to 82% for BPO, 91% for MFO, and 70% for CCO, it also fluctuated for high RIs (Fig. 3). This trend was probably caused by the low number of terms predicted at these RIs. For CCO, the RI essentially did not correlate with precision. This might point to a problem in assessing annotations for which the trivial Naïve method reached values of  $F_{\max} \sim 55\%$  outperforming most methods. Possibly, some prediction of the type “organelle” is all that is needed to achieve a high  $F_{\max}$  in this ontology.

**Similar performance for different embeddings.** We compared embeddings derived from two different language models (LMs). So far, we used embeddings from SeqVec<sup>20</sup>. Recently, ProtBert, a transformer-based approach using a masked language model objective (Bert<sup>38</sup>) instead of auto-regression and more protein sequences (BFD<sup>39,40</sup>) during pre-training, was shown to improve secondary structure prediction<sup>21</sup>. Replacing



**Figure 3. Precision and recall for different reliability indices (RIs).** We defined a reliability index (RI) measuring the strength of a prediction (Eq. 5), i.e., for the embedding proximity. Precision (Eq. 1) and recall (Eq. 2) were almost constant for lower RIs (up to  $\sim 0.3$ ) for all three ontologies (BPO, MFO, CCO). For higher RIs, precision increased while recall dropped. However, due to the low number of terms predicted at very high RIs ( $> 0.8$ ), precision fluctuated and was not fully correlated with RI. Panel (a) shows precision and recall for BPO, panel (b) for MFO, and panel (c) for CCO. Dashed vertical lines marked the thresholds used by the CAFA3 tool to compute  $F_{\max}$ : 0.35 for BPO, 0.28 for MFO, and 0.29 for CCO. At least for BPO and MFO higher RI values correlated with higher precision, i.e., users could use the RI to estimate how good the predictions are likely to be for their query, or to simply scan only those predictions more likely to be correct (e.g.  $RI > 0.8$ ).

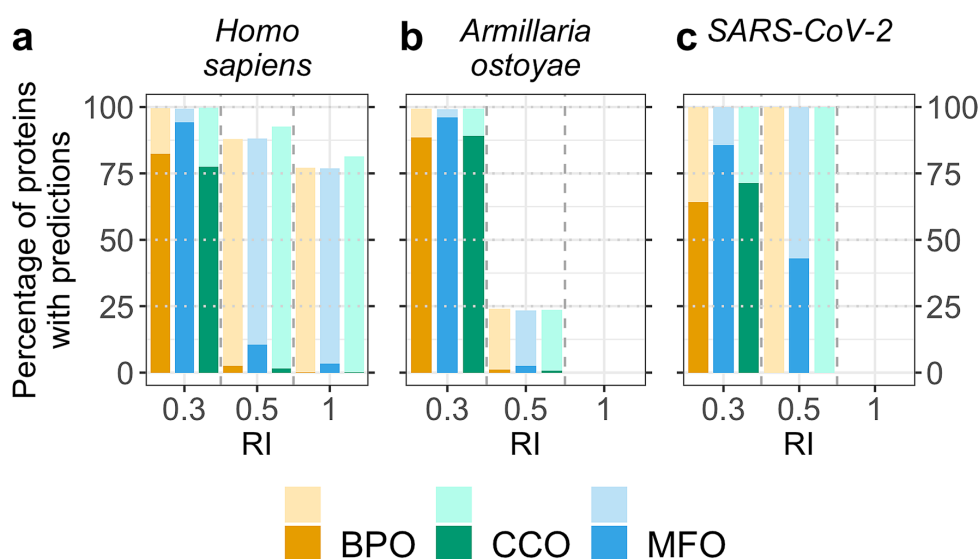
SeqVec by ProtBert embeddings to transfer annotations, our approach achieved similar  $F_{\max}$  scores (Table 1). In fact, the ProtBert  $F_{\max}$  scores remained within the 95% confidence intervals of those for SeqVec (Table 1). Similar results were observed when using GOA2017-100 (Table 1).

On the one hand, the similar performance for both embeddings might indicate that both LMs extracted equally beneficial aspects of function, irrespective of the underlying architecture (LSTMs in SeqVec, transformer encoders in ProtBert) or training set (SeqVec used UniRef50 with  $\sim 33$  M proteins, ProtBert used BFD with  $\sim 2.1$ B proteins). On the other hand, the similar  $F_{\max}$  scores might also highlight that important information was lost when averaging over the entire protein to render fixed-size vectors. The similarity in  $F_{\max}$  scores was less surprising given the high correlation between SeqVec and ProtBert embeddings ( $\rho = 0.58$ ,  $p$ -value  $< 2.2e-16$ ; Table 2). The two LMs correlated more with each other than either with PIDE (Table 2).

**No gain from simple combination of embedding- and sequence-based transfer.** All three approaches toward annotation transfer (embeddings from SeqVec or ProtBert, and sequence) had strengths; although performing worse on average, for some proteins sequence-transfer performed better. In fact, analyzing the pairs for which embedding-based transfer or sequence-based transfer outperformed the other method by at least four percentage points ( $|F_{\max}(\text{BLAST}) - F_{\max}(\text{embeddings})| \geq 4$ ) illustrated the expected cases for which PIDE was high and embedding similarity low, and vice versa, along with more surprising cases for which low PIDE still yielded better predictions than relatively high embedding RIs (Fig. S5). Overall, these results (Fig. S5) again underlined that LM embeddings abstract information from sequence that are relevant for comparisons and not captured by sequences alone. However, it also indicates that even protein pairs with low embedding similarity can share similar GO terms. In fact, embedding similarity for SeqVec embeddings only weakly correlated with GO term similarity (Spearman rank coefficient  $\rho = 0.28$ ,  $p$ -value  $< 2.2e-16$ ), but proteins with identical GO annotations were on average more likely to be close than proteins with more different GO annotations (Fig. S7). The similarity of GO terms for two proteins was proxied through the Jaccard index (Eq. 7). More details are provided in the SOM.

To benefit from the cases where BLAST outperformed our approach, we tried simple combinations: firstly, we considered all terms predicted by embeddings from either SeqVec or ProtBert. Secondly, reliability scores were combined leading to higher reliability for terms predicted in both approaches than for terms only predicted by one. None of those two improved performance (Table S4, method SeqVec/ProtBert). Other simple combinations also failed so far (Table S4, method SeqVec/ProtBert/BLAST). Future work might improve performance through more advanced combinations.

**Case study: embedding-based annotation transfer for three proteomes.** Due to its simplicity and speed, embedding-based annotation transfer can easily be applied to novel proteins to shed light on their potential functionality. We applied our method to the proteomes of three different proteomes: human (20,370 proteins from Swiss-Prot) as a well-researched proteome, the fungus *Armillaria ostoyae* (22,192 proteins, 0.01%



**Figure 4. Fraction of proteomes with predicted GO terms.** We applied our method to three proteomes (animal: *Homo sapiens*, fungus: *Armillaria ostoyae*, and virus: SARS-CoV-2) and monitored the fraction of proteins in each proteome for which our method predicted GO terms for different thresholds in embedding similarity (RI, Eq. 5). We show predictions for RI = 1.0 (“self-hits”), RI = 0.5 (with an expected precision/recall = 0.5), and RI = 0.3 (CAFA3 thresholds). Darker colored bars indicate predictions using GOA2020X as lookup set (only experimentally verified GO annotations) and lighter colors indicate predictions using GOA2020 as lookup set (using all annotations in GOA). (a) The human proteome is well-studied (all 20,370 proteins are in Swiss-Prot) and for most proteins, GO annotations are available, but those annotations are largely not experimentally verified (very small, dark-colored bars vs large, lighter-colored bars at RI = 1.0). (b) The proteome of the fungus *Armillaria ostoyae* appears more exceptional (0.01% of the 22,192 proteins were in Swiss-Prot); at RI  $\geq 0.5$ , predictions could be made only for 25% of the proteins when also using unverified annotations and none of the proteins already had any GO annotations. (c) While annotations were unknown for most proteins of the novel virus SARS-CoV-2 (no coverage at RI = 1), many annotations could be transferred from the human SARS coronavirus (SARS-CoV) and the bat coronavirus HKU3 (BtCoV) allowing GO term predictions for all proteins at reliability values  $\geq 0.5$ .

of these in Swiss-Prot), as one of the oldest (2500 years) and largest ( $4 \times 10^5$  kg/spanning over 10 km<sup>2</sup>) living organisms known today<sup>41</sup>, and SARS-CoV-2, the virus causing COVID-19 (14 proteins). At RI = 1.0, annotations were inferred from proteins of this organism (“self-hits”). Using only experimentally verified annotations (lookup data set GOA2020X), revealed both how few proteins were directly annotated (self-hits) in these organisms and how much of the sequence-annotation gap is gapped through embedding-based inference (Fig. 4: bars with darker orange, blue, green for BPO, CCO, and MFO respectively). In particular, for self-hits, i.e., proteins with 100% pairwise sequence identity (PIDE) to the protein with known annotation, it became obvious how few proteins in human have explicit experimental annotation (sum over all around 270), while through embedding-based inference up to 80% of all human proteins could be annotated through proteins from other organisms (light bars in Fig. 4 give results for the entire GOA2020 which is dominated by annotations not directly verified by experiment). For the other two proteomes from the fungus (*Armillaria ostoyae*) and the coronavirus (SARS-CoV-2), there were no inferences at this high level. On the other end of including all inferences as assessed through the data presented in all other figures and tables (i.e., at the default thresholds), for all three proteomes most proteins could be annotated directly from experimentally verified annotations through embeddings (three left-most bars in Fig. 4 for BPO, CCO, and MFO). In fact, when including all GO annotations from GOA (lookup set GOA2020), almost all proteins in all three proteomes could be annotated (Fig. 4: lighter colored left-most bars close to fraction of 1, i.e., all proteins). For SARS-CoV-2, our method reached 100% coverage (prediction for all proteins) already at RI  $\geq 0.5$  (Fig. 4c, lighter colors, middle bars) through well-studied, similar viruses such as the human SARS coronavirus (SARS-CoV). RI = 0.5 represent roughly a precision and recall of 50% for all three ontologies (Fig. 3). For *Armillaria ostoyae*, almost no protein was annotated through self-hits even when using unverified annotations (Fig. 4b: no bar at RI = 1). At RI = 0.5, about 25% of the proteins were annotated.

**Case study: embedding-based annotation transfer for SARS-CoV-2 proteome.** Given the relevance of SARS-CoV-2, we did not only apply our method to predict GO terms (BPO, MFO, and CCO) for all 14 SARS-CoV-2 proteins (taken from UniProt<sup>3</sup>; all raw predictions were made available as additional files named predictions\_semb\_sont.txt replacing the variables \$semb and \$sont as follows: \$semb = seqvec|probert,

and  $\$ont = bpo|mfo|cco$ ), but also investigated the resulting annotations further. While the two replicase polyproteins pp1a and pp1ab can also be split further into up to 12 non-structural proteins resulting in 28 proteins<sup>42</sup>, we used the definition from UniProt identifying 14 different proteins.

**Step 1: confirmation of known annotations.** Out of the 42 predictions (14 proteins in 3 ontologies), 12 were based on annotation transfers using proteins from the human SARS coronavirus (SARS-CoV), and 13 on proteins from the bat coronavirus HKU3 (BtCoV). CCO predictions appeared reasonable with predicted locations mainly associated with the virus (e.g., viral envelope, virion membrane) or the host (e.g., host cell Golgi apparatus, host cell membrane). Similarly, MFO predictions often matched well-known annotations, e.g., the replicase polyproteins 1a and 1ab were predicted to be involved in RNA-binding as confirmed by UniProt. In fact, annotations in BPO were known for 7 proteins (in total 40 GO terms), in MFO for 6 proteins (30 GO terms), and in CCO for 12 proteins (68 GO terms). Only three of these annotations were experimentally verified. With our method, we predicted 25 out of the 40 GO terms for BPO (63%), 14/30 for MFO (47%), and 59/68 for CCO (87%). Even more annotations were similar to the known GO annotations but were more or less specific (Table S5 summarized all predicted and annotated GO leaf terms, the corresponding names can be found in the additional files predictions\_semb\_ont.txt).

**Step 2: new predictions.** Since the GO term predictions matched well-characterized proteins, predictions might provide insights into the function of proteins without or with fewer annotations. For example, function and structure of the non-structural protein 7b (Uniprot identifier P0D7D8) are not known except for a transmembrane region of which the existence was supported by the predicted CCO annotation “integral component of the membrane” and “host cell membrane”. This CCO annotation was also correctly predicted by the embedding-based transfer from an *Ashbya gossypii* protein. Additionally, we predicted “transport of virus in host, cell to cell” for BPO and “proton transmembrane transporter activity” for MFO. This suggested non-structural protein 7b to play a role in transporting the virion through the membrane into the host cell. Visualizing the leaf term predictions in the GO hierarchy could help to better understand very specific annotations. For the BPO annotation of the non-structural protein 7b, the tree revealed that this functionality constituted two major aspects: The interaction with the host and the actual transport to the host (Fig. S10). To visualize the predicted terms in the GO hierarchy, for example the tool NaviGO<sup>43</sup> can be used which can help to interpret the GO predictions given for the SARS-CoV-2 proteins here.

Comparing annotation transfers based on embeddings from SeqVec and from ProtBert showed that 16 of the 42 predictions agreed for the two different language models (LMs). For five predictions, one of the two LMs yielded more specific annotations, e.g., for the nucleoprotein (Uniprot identifier P0D7C9) which is involved in viral genome packaging and regulation of viral transcription and replication. For this protein, SeqVec embeddings found no meaningful result, while ProtBert embeddings predicted terms such as “RNA polymerase II preinitiation complex” and “positive regulation of transcription by RNA polymerase II” fitting to the known function of the nucleoprotein. This example demonstrated how the combination of results from predictions using different LMs may refine GO term predictions.

## Conclusions

We introduce a new concept for the prediction of GO terms, namely the annotation transfer based on similarity of embeddings obtained from deep learning language models (LMs). This approach conceptually replaces sequence information by complex embeddings that capture some non-local information beyond sequence similarity. The underlying LMs (SeqVec & ProtBert) are highly involved and complex, and their training is time-consuming and data intensive. Once that is done, those pre-trained LMs can be applied, their abstracted understanding of the language of life as captured by protein sequences can be transferred to yield an extremely simple, yet effective novel method for annotation transfer. This novel prediction method complements homology-based inference. Despite its simplicity, this new method outperformed by several margins of statistically significance homology-based inference (“BLAST”) with  $F_{max}$  values of BPO + 11 ± 2% ( $F_{max}(\text{embedding}) - F_{max}(\text{sequence})$ ), MFO + 8 ± 3%, and CCO + 11 ± 2% (Table 1, Fig. 1); it even might have reached the top ten, had it participated at CAFA3 (Fig. 1). Embedding-based transfer remained above the average for sequence-based transfer even for protein pairs with PIDE < 20% (Fig. 2), i.e., embedding similarity worked for proteins that diverged beyond the recognition in pairwise alignments (Figs. S2 & S3). Embedding-based transfer is also blazingly fast to compute, i.e., around 0.05 s per protein. The only time-consuming step is computing embeddings for all proteins in the lookup database which needs to be done only once; it took about 30 min for the entire human proteome. GO annotations added from 2017 to 2020 improved both sequence- and embedding-based annotation transfer significantly (Table 1). Another aspect of the simplicity is that, at least in the context of the CAFA3 evaluation, the choice of none of the two free parameters really mattered: embeddings from both LMs tested performed, on average, equally, and the number of best hits (k-nearest neighbors) did not matter much (Table S2). The power of this new concept is generated by the degree to which embeddings implicitly capture important information relevant for protein structure and function prediction. One reason for the success of our new concept was the limited correlation between embeddings and sequence (Table 2). Additionally, the abstraction of sequence information in embeddings appeared to make crucially meaningful information readily available (Fig. S6). This implies that embeddings have the potential to revolutionize the way sequence comparisons are carried out.

## Methods

**Generating embedding space.** The embedding-based annotation transfer introduced here requires each protein to be represented by a fixed-length vector, i.e., a vector with the same dimension for a protein of 30 and another of 40,000 residues (maximal sequence length for ProtBert). To this end, we used SeqVec<sup>20</sup> to represent each protein in our data set by a fixed size embedding. SeqVec is based on ELMo<sup>44</sup> using a stack of LSTMs<sup>45</sup> for

auto-regressive pre-training<sup>46,47</sup> i.e., predicting the next token (originally a word in a sentence, here an amino acid in a protein sequence), given all previous tokens. Two independent stacks of LSTMs process the sequence from both directions. During pre-training, the two directions are joined by summing their losses; concatenating the hidden states of both directions during inference lets supervised tasks capture bi-directional context. For SeqVec, three layers, i.e., one uncontextualized CharCNN<sup>48</sup> and two bi-directional LSTMs, were trained on each protein in UniRef50 (UniProt<sup>3</sup> clustered at 50% PIDE resulting in ~33 M proteins). In order to increase regularization, the weights of the token representation (CharCNN) as well as the final Softmax layer were shared between the two LSTM directions, and a 10% dropout rate was applied. For SeqVec, the CharCNN as well as each LSTM has a hidden state of size 512, resulting in a total of 93 M free parameters. As only unlabeled data (no phenotypical data) was used (self-supervised training), the embeddings could not capture any explicit information such as GO numbers. Thus, SeqVec does not need to be retrained for subsequent prediction tasks using the embeddings as input. The hidden states of the pre-trained model are used to extract features. Corresponding to its hidden state size, SeqVec outputs for each layer and each direction a 512-dimensional vector; in this work, only the forward and backward passes of the first LSTM layer were extracted and concatenated into a matrix of size  $L * 1024$  for a protein with  $L$  residues. While the auto-regressive pre-training only allowed to gather contextual information from either direction, the concatenation of the representations allowed our approach to benefit from bi-directional context. A fixed-size representation was then derived by averaging over the length dimension, resulting in a vector of size 1024 for each protein (Fig. S11). This simple way of information pooling (also called *global average pooling*) outperformed in many cases more sophisticated methods in NLP<sup>49</sup> and showed competitive performance in bioinformatics for some tasks<sup>20,21,26</sup>. Based on experience from NLP<sup>49,50</sup>, we also investigated the effect of using a different pooling strategy, i.e., maximum pooling, to derive fixed size representations from SeqVec embeddings.

To evaluate the effect of using different LMs to generate the embeddings, we also used a transformer-based LM trained on protein sequences (ProtBert-BFD<sup>21</sup>, here simply referred to as *ProtBert*). ProtBert is based on the LM BERT<sup>38</sup> (Bidirectional Encoder Representations from Transformers<sup>51</sup>) which processes sequential data through the self-attention mechanism<sup>52</sup>. Self-attention compares all tokens in a sequence to all others in parallel, thereby capturing long-range dependencies better than LSTMs. BERT also replaced ELMO's auto-regressive objective by masked language modeling during pre-training, i.e., reconstructing corrupted tokens from the input, which enables to capture bi-directional context. ProtBert was trained with 30 attention layers, each having 16 attention heads with a hidden state size of 1024 resulting in a total of 420 M free parameters which were optimized on 2.1B protein sequences (BFD)<sup>39,40</sup> which is 70 times larger than UniRef50. The output of the last attention layer of ProtBert was used to derive a 1024-dimensional embedding for each residue. As for SeqVec, the resulting  $L * 1024$  matrix was pooled by averaging over protein length providing a fixed-size vector of dimension 1024 for each protein. Usually, BERT's special CLS-token is used for sequence-classification tasks<sup>38</sup> as it is already optimized during pre-training on summarizing sequence information by predicting whether two sentences are consecutive in a document or not. In the absence of such a concept for proteins, this second loss was dropped from ProtBert's pre-training rendering the CLS token without further fine-tuning on supervised tasks uninformative.

Embeddings derived from LMs change upon retraining the model with a different random seed, even using the same data and hyper-parameters. They are likely to change more substantially when switching the training data or tuning hyper-parameters. As retraining LMs is computationally (and environmentally) expensive, we leave assessing the impact of fine-tuning LMs to the future.

Generating the embeddings for all human proteins using both SeqVec and ProtBert allowed estimating the time required for the generation of the input to our new method. Using a single Nvidia GeForce GTX1080 with 8 GB vRAM and dynamic batching (depending on the sequence length), this took, on average, about 0.05 s per protein<sup>21</sup>.

**Data set.** To create a database for annotation transfer, we extracted protein sequences with annotated GO terms from the Gene Ontology Annotation (GOA) database<sup>53-55</sup> (containing 29,904,266 proteins from UniProtKB<sup>3</sup> in February 2020). In order to focus on proteins known to exist, we only extracted records from Swiss-Prot<sup>56</sup>. Proteins annotated only at the ontology roots, i.e. proteins limited to "GO:0003674" (molecular\_function), "GO:0008150" (biological\_process), or "GO:0005575" (cellular\_component) were considered meaningless and were excluded. The final data set *GOA2020* contained 295,558 proteins (with unique identifiers, IDs) described by 31,485 different GO terms. The GO term annotation for each protein includes all annotated terms and all their parent terms. Thereby, proteins are, on average, annotated by 37 terms in BPO, 14 in MFO, and 9 in CCO. Counting only leaves brought the averages to 3 in BPO, 2 in MFO, and 3 in CCO.

For comparison to methods that contributed to CAFA3<sup>19</sup>, we added another data set *GOA2017* using the GOA version available at the submission deadline of CAFA3 (Jan 17, 2017). After processing (as for *GOA2020*), *GOA2017* contained 307,287 proteins (unique IDs) described by 30,124 different GO terms. While we could not find a definite explanation for having fewer proteins in the newer database (*GOA2020* 295 K proteins vs. *GOA2017* with 307 K), we assume that it originated from major changes in GO including the removal of obsolete and inaccurate annotations and the refactoring of MFO<sup>2</sup>.

The above filters neither excluded GOA annotations inferred from phylogenetic evidence and author statements nor those based on computational analysis. We constructed an additional data set, *GOA2017X* exclusively containing proteins annotated in Swiss-Prot as experimental (evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS, or IC) following the CAFA3 definition<sup>19</sup>. We further excluded all entries with PIDE = 100% to any CAFA3 target bringing *GOA2017X* to 303,984 proteins with 28,677 different GO terms.

**Performance evaluation.** The targets from the CAFA3 challenge<sup>19</sup> were used to evaluate the performance of our new method. Of the 130,827 targets originally released for CAFA3, experimental GO annotations were obtained for 3328 proteins at the point of the final benchmark collection in November 2017<sup>19</sup>. This set consisted of the following subsets with experimental annotations in each sub-hierarchy of GO: BPO 2145, MFO 1101, and CCO 1097 (more details about the data set are given in the original CAFA3 publication<sup>19</sup>).

We used an additional data set, dubbed *GOA2020-new*, containing proteins added to GOA after February 2020, i.e., the point of accession for the GOA set used during the development of our method in preparation for CAFA4. This set consisted of 298 proteins with experimentally verified GO annotations and without any identical hits (i.e. 100% PIDE) in the lookup set *GOA2020*.

In order to expand the comparison of the transfer based on sequence- and embedding similarity, we also reduced the redundancy through applying CD-HIT and PSI-CD-HIT<sup>57</sup> to the *GOA2020* and *GOA2017* sets against the evaluation set at thresholds  $\theta$  of PIDE = 100, 90, 80, 70, 60, 50, 40, 30 and 20% (Table S6 in the Supporting Online Material (SOM) for more details about these nine subsets).

We evaluated our method against two baseline methods used at CAFA3, namely *Naïve* and *BLAST*, as well as, against CAFA3's top ten<sup>19</sup>. We computed standard performance measures. True positives (TP) were GO terms predicted above a certain reliability (RI) threshold (Method below), false positives (FP) were GO terms predicted but not annotated, and false negatives (FN) were GO terms annotated but not predicted. Based on these three numbers, we calculated precision (Eq. 1), recall (Eq. 2), and F1 score (Eq. 3) as follows.

$$P = \text{precision} = \frac{TP}{TP + FP} \quad (1)$$

$$R = \text{recall} = \frac{TP}{TP + FN} \quad (2)$$

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

The  $F_{\max}$  value denoted the maximum F1 score achievable for any threshold in reliability (RI, Eq. 5). This implies that the assessment fixes the optimal value rather than the method providing this value. Although this arguably over-estimates performance, it has evolved to a quasi-standard of CAFA; the publicly available CAFA Assessment Tool<sup>18,19</sup> calculated  $F_{\max}$  for the CAFA3 targets in the same manner as for the official CAFA3 evaluation. If not stated otherwise, we reported precision and recall values for the threshold leading to  $F_{\max}$ .

CAFA3 assessed performance separately for two sets of proteins for all three ontologies: (i) proteins for which no experimental annotations were known beforehand (no-knowledge, NK evaluation mode) and (ii) proteins with some experimental annotations in one or two of the other ontologies (limited-knowledge, LK evaluation mode)<sup>9,19</sup>. We also considered these sets separately in our assessment. CAFA3 further distinguished between *full* and *partial evaluation* with *full evaluation* penalizing if no prediction was made for a certain protein, and *partial evaluation* restricting the assessment to the subset of proteins with predictions<sup>19</sup>. Our method predicted for every protein; thus, we considered only the *full evaluation*. Also following CAFA3, symmetric 95% confidence intervals were calculated as error estimates assuming a normal distribution and 10,000 bootstrap samples estimated mean and standard deviation.

**Method: annotation transfer through embedding similarity.** For a given query protein Q, GO terms were transferred from proteins with known GO terms (sets *GOA2020* and *GOA2017*) through an approach similar to the k-nearest neighbor algorithm (k-NN)<sup>58</sup>. For the query Q and for all proteins in, e.g., *GOA2020*, the SeqVec<sup>20</sup> embeddings were computed. Based on the Euclidean distance between two embeddings  $n$  and  $m$  (Eq. 4), we extracted the  $k$  closest hits to the query from the database where  $k$  constituted a free parameter to optimize.

$$d(n, m) = \sqrt{\sum_{i=1}^{1024} (n_i - m_i)^2} \quad (4)$$

In contrast to standard k-NN algorithms, all annotations from all hits were transferred to the query instead of only the most frequent one<sup>58</sup>. When multiple pairs reached the same distance, all were considered, i.e., for a given  $k$ , more than  $k$  proteins might be considered for the GO term prediction. The calculation of the pairwise Euclidean distances between queries and all database proteins and the subsequent nearest neighbor extraction was accomplished very efficiently. For instance, the nearest-neighbor search of 1000 query proteins against *GOA20\** with about 300,000 proteins took on average only about 0.005 s per query on a single i7-6700 CPU, i.e., less than two minutes for all human proteins.

Converting the Euclidean distance enabled to introduce a reliability index (RI) ranging from 0 (weak prediction) to 1 (confident prediction) for each predicted GO term  $p$  as follows:

$$RI(p) = \frac{1}{k} \sum_{i=1}^l \frac{0.5}{0.5 + d(q, n_i)} \quad (5)$$

with  $k$  as the overall number of hits/neighbors,  $l$  as the number of hits annotated with the GO term  $p$  and the distance  $d(q, n_i)$  between query and hit being calculated according to Eq. (4).

Proteins represented by an embedding identical to the query protein ( $d=0$ ) led to  $RI=1$ . Since the RI also takes into account, how many proteins  $l$  in a list of  $k$  hits are annotated with a certain term  $p$  (Eq. 5), predicted terms annotated to more proteins (larger  $l$ ) have a higher RI than terms annotated to fewer proteins (smaller  $l$ ). As this approach accounts for the agreement of the annotations between the  $k$  hits, it requires the RI to be normalized by the number of considered neighbors  $k$ , making it not directly comparable for predictions based on different values for  $k$ . On top, if different embeddings are used to identify close proteins, RI values are not directly comparable, because embeddings might be on different scales.

Instead of assessing embedding proximity through the Euclidean distance, the embedding field typically uses the cosine distance (Eq. 6):

$$d_{\text{cosine}}(n, m) = 1 - \frac{\sum_{i=1}^{1024} n_i m_i}{\sqrt{\sum_{i=1}^{1024} n_i^2} \cdot \sqrt{\sum_{i=1}^{1024} m_i^2}} \quad (6)$$

Our initial assessment suggested cosine and Euclidean distance to perform alike, and we chose to use the metric more familiar to structural biologists, namely the Euclidean distance throughout this analysis.

**GO term similarity.** We measured the similarity between two sets of GO annotations  $A$  and  $B$  through the Jaccard index (Eq. 7) where  $|A \cap B|$  is the number of GO terms present in both sets and  $|A \cup B|$  is the number of GO terms present in at least one of the sets (duplicates are only counted once):

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (7)$$

**Correlation analysis.** We analyzed the correlation between sequence identity and embedding similarity through the Spearman's rank correlation coefficient because our data was neither distributed normally, nor were the two measures for similarity measures linear. In contrast to, e.g. Pearson correlation, Spearman does not assume a normal distribution and detects monotonic instead of linear relations<sup>59,60</sup>.

**Availability.** GO term predictions using embedding similarity for a certain protein sequence can be performed through our publicly available webserver: <https://embed.protein.properties/>. The source code along with all embeddings for GOA2020 and GOA2017, and the CAFA3 targets are also available on GitHub: <https://github.com/Rostlab/goPredSim> (more details in the repository). In addition to reproducing the results, the source code also allows calculating embedding similarity using cosine distance.

### Data availability

The source code and the embedding sets for target proteins and lookup databases are publicly available as a GitHub repository. GO term predictions for the SARS-CoV-2 proteins are provided as additional files and in the GitHub repository.

Received: 7 September 2020; Accepted: 24 December 2020

Published online: 13 January 2021

### References

- Krebs, H. A. & Johnson, W. A. Metabolism of ketonic acids in animal tissues. *Biochem J* **31**, 645–660. <https://doi.org/10.1042/bj0310645> (1937).
- The Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res* **47**, D330–D338. <https://doi.org/10.1093/nar/gky1055> (2019).
- UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.* **47**, D506–D515. <https://doi.org/10.1093/nar/gky1049> (2019).
- Hirst, J. D. & Sternberg, M. J. E. Prediction of structural and functional features of protein and nucleic acid sequences by artificial neural networks. *Biochemistry* **31**, 615–623 (1992).
- Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O. & Ofran, Y. Automatic prediction of protein function. *Cell. Mol. Life Sci.* **60**, 2637–2650 (2003).
- Leslie, C., Eskin, E., Weston, J. & Noble, W. S. Mismatch string kernels for SVM protein classification. *Bioinformatics*, in press (2003).
- Ofran, Y., Punta, M., Schneider, R. & Rost, B. Beyond annotation transfer by homology: novel protein–function prediction methods to assist drug discovery. *Drug Discov. Today* **10**, 1475–1482 (2005).
- Hamp, T. *et al.* Homology-based inference sets the bar high for protein function prediction. *BMC Bioinform.* **14**(Suppl 3), S 7. <https://doi.org/10.1186/1471-2105-14-S3-S7> (2013).
- Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nat. Methods* **10**, 221–227. <https://doi.org/10.1038/nmeth.2340> (2013).
- Cozzetto, D., Minneci, F., Curren, H. & Jones, D. T. FFPred 3: feature-based function prediction for all Gene Ontology domains. *Sci. Rep.* **6**, 31865. <https://doi.org/10.1038/srep31865> (2016).
- Kulmanov, M. & Hoehndorf, R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* **36**, 422–429. <https://doi.org/10.1093/bioinformatics/btz595> (2020).
- Zuckermandl, E. Evolutionary processes and evolutionary noise at the molecular level. *J. Mol. Evol.* **7**, 269–311 (1976).
- Nakai, K. & Horton, P. PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem. Sci.* **24**, 34–36 (1999).



14. Nair, R. & Rost, B. Sequence conserved for sub-cellular localization. *Protein Sci.* **11**, 2836–2847 (2002).
15. Goldberg, T. *et al.* LocTree3 prediction of localization. *Nucleic Acids Res.* **42**, W350–355. <https://doi.org/10.1093/nar/gku396> (2014).
16. Qiu, J. *et al.* ProNA2020 predicts protein-DNA, protein-RNA, and protein-protein binding proteins and residues from sequence. *J. Mol. Biol.* **432**, 2428–2443. <https://doi.org/10.1016/j.jmb.2020.02.026> (2020).
17. Goldberg, T., Rost, B. & Bromberg, Y. Computational prediction shines light on type III secretion origins. *Sci. Rep.* **6**, 34516. <https://doi.org/10.1038/srep34516> (2016).
18. Jiang, Y. *et al.* An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.* **17**, 184. <https://doi.org/10.1186/s13059-016-1037-6> (2016).
19. Zhou, N. *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.* **20**, 244. <https://doi.org/10.1186/s13059-019-1835-8> (2019).
20. Heinzinger, M. *et al.* Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinform.* **20**, 723. <https://doi.org/10.1186/s12859-019-3220-8> (2019).
21. Elnaggar, A. *et al.* ProtTrans: towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv* (2020).
22. Mikolov, T., Cheng, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings*.
23. Allen, C. & Hospedales, T. Analogies Explained: Towards Understanding Word Embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, 223–231 (PMLR).
24. Brokos, G.-I., Malakasiotis, P. & Androutsopoulos, I. Using Centroids of Word Embeddings and Word Mover's Distance for Biomedical Document Retrieval in Question Answering. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, 114–118 (Association for Computational Linguistics).
25. Kusner, M. J., Sun, Y., Kolkin, N. I. & Weinberger, K. Q. From Word Embeddings to Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*.
26. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 622803. <https://doi.org/10.1101/622803> (2020).
27. Vig, J. *et al.* BERTology meets Biology: Interpreting Attention in Protein Language Models. *arXiv* (2020).
28. R Core Team (R Foundation for Statistical Computing, 2017).
29. Rost, B. Twilight zone of protein sequence alignments. *Protein Eng.* **12**, 85–94 (1999).
30. Rost, B. Enzyme function less conserved than anticipated. *J. Mol. Biol.* **318**, 595–608 (2002).
31. Miika, S. & Rost, B. Protein–protein interactions more conserved within species than across species. *PLoS Comput. Biol.* **2**, e79. <https://doi.org/10.1371/journal.pcbi.0020079> (2006).
32. Clark, W. T. & Radivojac, P. Analysis of protein function and its prediction from amino acid sequence. *Proteins* **79**, 2086–2096. <https://doi.org/10.1002/prot.23029> (2011).
33. Rost, B. Protein structures sustain evolutionary drift. *Fold Des.* **2**, S19–S24 (1997).
34. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410. [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2) (1990).
35. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028. <https://doi.org/10.1038/nbt.3988> (2017).
36. Waterhouse, A. *et al.* SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* **46**, W296–W303. <https://doi.org/10.1093/nar/gky427> (2018).
37. El-Mabrouk, N. & Slonim, D. K. ISMB 2020 proceedings. *Bioinformatics* **36**, i1–i2. <https://doi.org/10.1093/bioinformatics/btaa537> (2020).
38. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186 (Association for Computational Linguistics).
39. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manifold. *Nat. Methods* **16**, 603–606. <https://doi.org/10.1038/s41592-019-0437-4> (2019).
40. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nat. Commun.* **9**, 2542. <https://doi.org/10.1038/s41467-018-04964-5> (2018).
41. Anderson, J. B. *et al.* Clonal evolution and genome stability in a 2500-year-old fungal individual. *Proc. Biol. Sci.* **285**, 20182233. <https://doi.org/10.1098/rspb.2018.2233> (2018).
42. O'Donoghue, S. I. *et al.* SARS-CoV-2 structural coverage map reveals state changes that disrupt host immunity. *bioRxiv* (2020).
43. Wei, Q., Khan, I. K., Ding, Z., Yerneni, S. & Kihara, D. NaviGO: interactive tool for visualization and functional similarity and coherence analysis with gene ontology. *BMC Bioinform.* **18**, 177. <https://doi.org/10.1186/s12859-017-1600-5> (2017).
44. Peters, M. E. *et al.* Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237 (Association for Computational Linguistics).
45. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735> (1997).
46. Mousa, A. & Schuller, B. Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models: a generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 1023–1032 (Association for Computational Linguistics).
47. Peters, M., Ammar, W., Bhagavatula, C. & Power, R. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1756–1765 (Association for Computational Linguistics).
48. Kim, Y., Jernite, Y., Sontag, D. & Rush, A. M. Character-aware Neural Language Models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. (AAAI Press).
49. Shen, D. *et al.* Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 440–450 (Association for Computational Linguistics).
50. Conneau, A., Douwe, K., Schwenk, H., Barrault, L. & Bordes, A. Supervised Learning of Universal Sentence Representations From Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 670–680 (Association for Computational Linguistics).
51. Vaswani, A. *et al.* Attention is All You Need. In *Neural information processing systems conference*. (eds I Guyon *et al.*) 5998–6008 (Curran Associates, Inc.).
52. Bahdanau, D., Cho, K. H. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate in *arXiv*.
53. Camon, E. *et al.* The Gene Ontology Annotation (GOA) database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res.* **32**, D262–266. <https://doi.org/10.1093/nar/gkh021> (2004).
54. Huntley, R. P. *et al.* The GOA database: gene Ontology annotation updates for 2015. *Nucleic Acids Res.* **43**, D1057–1063. <https://doi.org/10.1093/nar/gku1113> (2015).
55. GOA. <http://www.ebi.ac.uk/GOA> (2020).

56. Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M. & Bairoch, A. UniProtKB/Swiss-Prot. *Methods Mol. Biol.* **406**, 89–112. [https://doi.org/10.1007/978-1-59745-535-0\\_4](https://doi.org/10.1007/978-1-59745-535-0_4) (2007).
57. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152. <https://doi.org/10.1093/bioinformatics/bts565> (2012).
58. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13**, 21–27 (1967).
59. Dodge, Y. *The Concise Encyclopedia of Statistics 502–505* (Springer, New York, 2008).
60. Spearman, C. The Proof and Measurement of Association Between Two Things. *Am. J. Psychol.* **15**, 72–101 (1904).

## Acknowledgements

Thanks to Tim Karl and Inga Weise (both TUM) for invaluable help with technical and administrative aspects of this work. Thanks to the organizers and participants of the CAFA challenge for their crucial contributions, in particular to Predrag Radivojac (Northeastern Boston), Iddo Friedberg (Iowa State Ames), Sean Mooney (U of Washington Seattle), Casey Green (U Penn Philadelphia, USA), Mark Wass (Kent, England), and Kimberly Reynolds (U Texas Dallas). Particular thanks to the maintainers of the Gene Ontology (in particular to Michael Ashburner—Cambridge and Suzanna Lewis – Berkeley), as well as, the Gene Ontology Annotation database (maintained by the team of Claire O'Donovan at the EBI) for their standardized vocabulary and curated data sets. Last, but not least, thanks to all maintainers of public databases and to all experimentalists who enabled this analysis by making their data publicly available. This work was supported by the Bavarian Ministry of Education through funding to the TUM, by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium für Bildung und Forschung), by a grant from BMBF (Software Campus, 01IS17049) as well as by a grant from Deutsche Forschungsgemeinschaft (DFG-GZ: RO1320/4–1). We gratefully acknowledge the support of NVIDIA Corporation with the donation of one Titan GPU used for this research.

## Author contributions

M.L. implemented the method goPredSim and performed evaluations. M.H. provided SeqVec and ProtBert embeddings and contributed various ideas and comments to improve goPredSim and its assessment. M.L. and M.H. performed the major part of manuscript writing and figure generation. C.D. implemented the webserver allowing GO term predictions using our method through a web interface. T.O. had the initial idea to calculate correlation between sequence and embedding similarity and computed the combination of sequence- and embedding-based transfer. B.R. supervised and guided the work over the entire time and proofread the manuscript. All authors read and approved the final manuscript.

## Funding

Open Access funding enabled and organized by Projekt DEAL.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-020-80786-0>.

**Correspondence** and requests for materials should be addressed to M.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

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



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

© The Author(s) 2021

**3.3. Supplementary Material: Littmann, Heinzinger et al.,  
Scientific Reports (2021)**

# Supporting online material (SOM)

## for:

# Embeddings from deep learning transfer GO annotations beyond homology

**Maria Littmann** <sup>1,2,\*, $\diamond$</sup> , **Michael Heinzinger** <sup>1,2, $\diamond$</sup> , **Christian Dallago** <sup>1,2</sup>,  
**Tobias Olenyi**<sup>1</sup> & **Burkhard Rost** <sup>1,3,4</sup>

- 1 TUM (Technical University of Munich) Department of Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr. 3, 85748 Garching/Munich, Germany
- 2 TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
- 3 Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany & TUM School of Life Sciences Weihenstephan (TUM-WZW), Alte Akademie 8, Freising, Germany
- 4 Department of Biochemistry and Molecular Biophysics, Columbia University, 701 West, 168th Street, New York, NY 10032, USA
- \* Corresponding author: [littmann@rostlab.org](mailto:littmann@rostlab.org), <http://www.rostlab.org/>  
Tel: +49-289-17-814 (email rost: [assistant@rostlab.org](mailto:assistant@rostlab.org))

$\diamond$  These authors contributed equally

## Table of Contents for Supporting Online Material (SOM)

<b>TABLE OF CONTENTS FOR SUPPORTING ONLINE MATERIAL (SOM)</b> .....	1
SHORT DESCRIPTION OF SUPPORTING ONLINE MATERIAL .....	2
<b>MATERIAL</b> .....	3
Fig. S1: Precision and recall for different lookup sets based on <i>GOA2017</i> .....	3
Fig. S2: $F_{\max}$ of our method for different protein lengths .....	4
Fig. S3: $F_{\max}$ for our method and CAFA3 competitors using LK evaluation mode .....	5
Fig. S4: $F_{\max}$ , precision, and recall for different lookup sets based on <i>GOA2020</i> .....	6
Fig. S5: Proteins for which embedding-based or homology-based inference worked better dependent of RI and PIDE .....	7
Fig. S6: Comparative model for two target proteins and the corresponding hits found through embedding- or sequence similarity .....	8
Weak correlation between embedding similarity and GO term similarity .....	9
Fig. S7: Embedding similarity for different levels of GO term similarity .....	10
Fig. S8: Fraction of proteomes with predicted GO terms using lookup set <i>GOA2020</i> ... 11	11
Fig. S9: Fraction of proteomes with predicted GO terms using lookup set <i>GOA2020X</i> 13	13
Fig. S10: Visualization of predicted GO term for Nsp7b from SARS-CoV-2..... 15	15
Fig. S11: Visualization of the embedding generation using SeqVec..... 16	16
Table S1: Correlation between $F_{\max}$ and protein length .....	17
Table S2: $F_{\max}$ and average number of predicted GO terms for different values of $k$ ... 17	17
Table S3: Precision, recall, and average number of predicted GO terms for different values of $k$ ..... 18	18
Table S4: $F_{\max}$ for different combinations of embedding-based and homology-based annotation transfer..... 19	19
Table S5: Annotated and predicted GO terms for SARS-CoV-2 proteins .....	20
Table S6: Datasets for similarity lookup at different sequence identity thresholds..... 22	22
<b>REFERENCES FOR SUPPORTING ONLINE MATERIAL</b> .....	23

## Short description of Supporting Online Material

In this Supporting Online Material (SOM), we show a more detailed performance assessment of our method to predict GO terms and provide details regarding the used data sets as well as a sketch of how to extract embeddings via SeqVec (Fig. S11). While averaging over long proteins could lead to information loss in the resulting embeddings, the performance of our method did not correlate with the protein length (Fig. S2, Table S1).

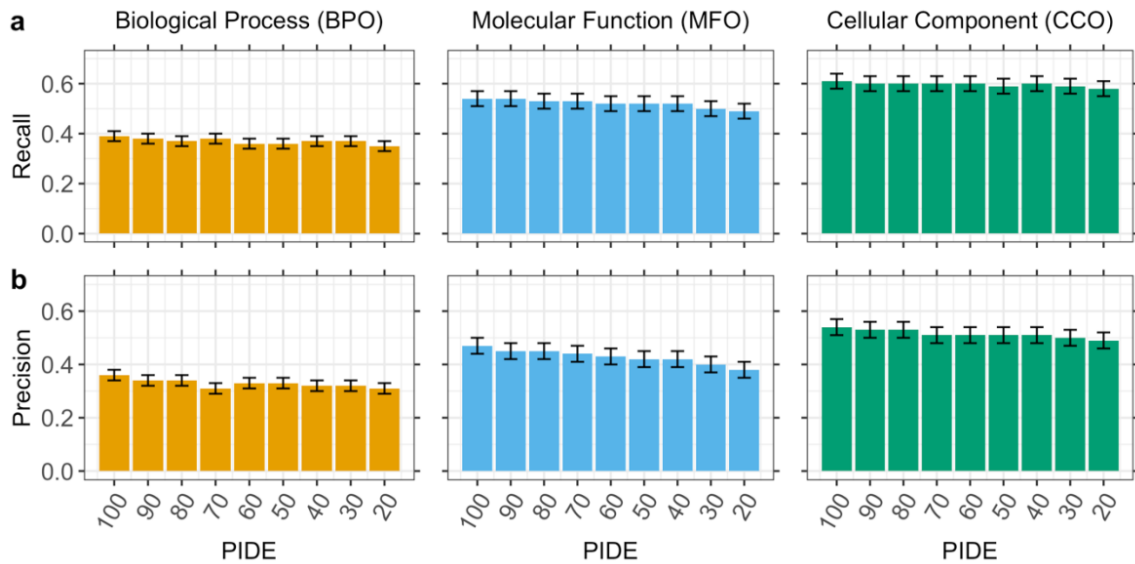
As  $F_{\max}$ , precision and recall decreased for lookup sets redundancy reduced at lower sequence identity thresholds for *GOA2017* and *GOA2020* (Fig. S1, S4). Performance generally increased for the top ten CAFA3 competitors as well as for our method for the *LK* evaluation mode (Fig. S3). In the *LK* evaluation mode, targets are evaluated for which some annotations were already known at point of submission for CAFA3 and that gained additional annotations since then. Comparing the performance of homology-based inference with our method did not show a clear correlation between embedding similarity, sequence identity, and which of the methods performed better (Fig. S5). A more detailed analysis of two example targets and the respective hits through embedding- and sequence similarity shows that embeddings seem to better capture structural relationship (Fig. S6). We also evaluate our underlying assumption that proteins close in embedding space should have similar GO annotations (Fig. S7).

We applied our method to three different organisms (human, the fungus *Armillaria ostoyae*, and virus: SARS-CoV-2) and show the fraction of proteins with a GO term prediction for different RI thresholds using *GOA2020* (Fig. S8) or *GOA2020X* (Fig. S9). For further analysis of the predictions, the predicted leaf terms can be visualized in the GO tree structure. Fig. S10 shows an example for the non-structural protein 7b of SARS-CoV-2.

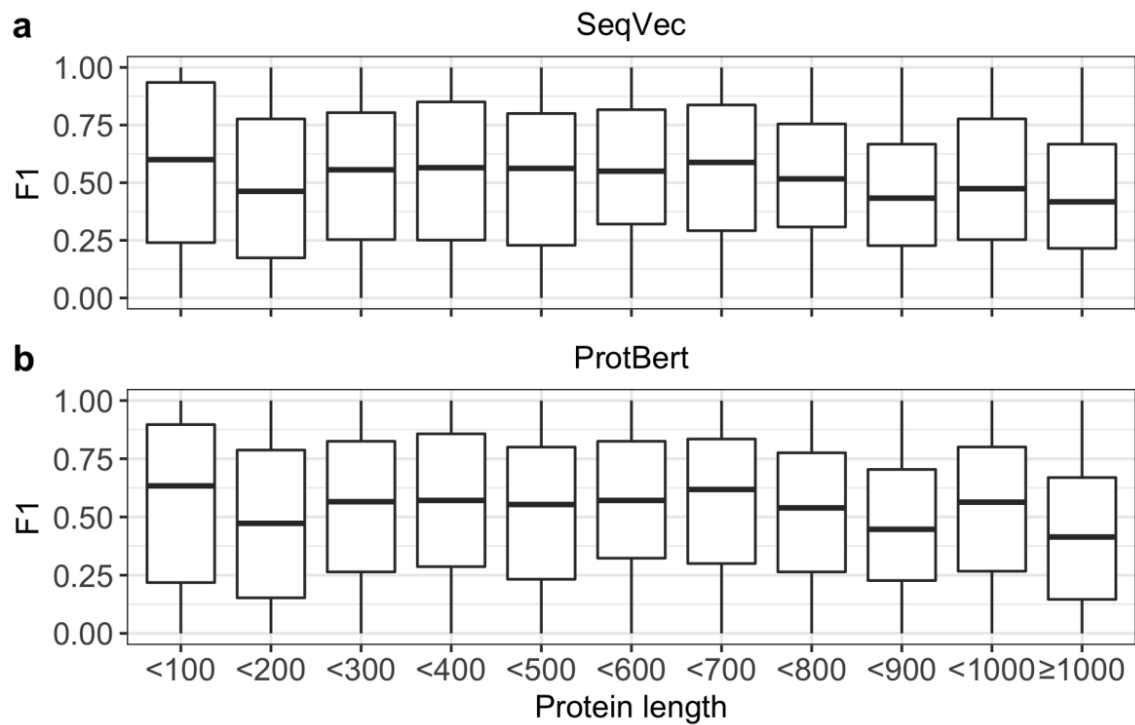
The choice of neighbors ( $k$ ) to include for annotation transfer did not affect  $F_{\max}$  (Table S2), however including more hits than the closest one could increase recall and the quality of predictions if e.g. only unspecific terms are annotated to the closest hit (Table S3). A detailed analysis of annotated and predicted GO terms for the 14 proteins from SARS-CoV-2 showed that only three of the annotated GO terms are experimentally verified and that our method predicted 83%, 47%, and 87% of the annotated terms for BPO, MFO, and CCO (Table S4). Surprisingly, the *GOA2017* set was larger than *GOA2020* in terms of sequences (Table S1), however not in terms of GO annotations. In general, the redundancy-reduced versions of the two sets had on average a sequence similarity of 43-44% while the number of identical sequences (by identifier) dropped for the sets reduced at 20 and 30% sequence identity compared to the other sets (Table S5).

## Material

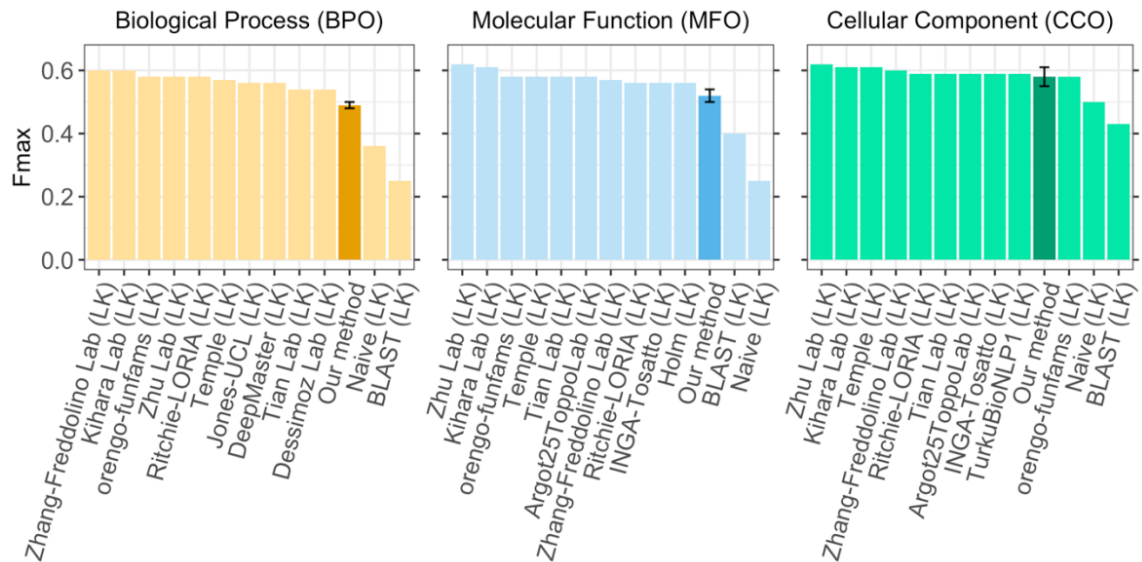
**Fig. S1: Precision and recall for different lookup sets based on GOA2017**



To test how the level of percentage pairwise sequence identity affects the performance of our method, we removed proteins above a certain pairwise sequence identity (as indicated on the x-axes) to the targets from our lookup set based on GOA version 2017. Panel **a** shows the recall and panel **b** the precision for BPO, MFO, and CCO, respectively, for lookup sets of <100, 90, 80, 70, 60, 50, 40, 30, and 20% sequence identity to the target proteins. Error bars indicate 95% confidence intervals.

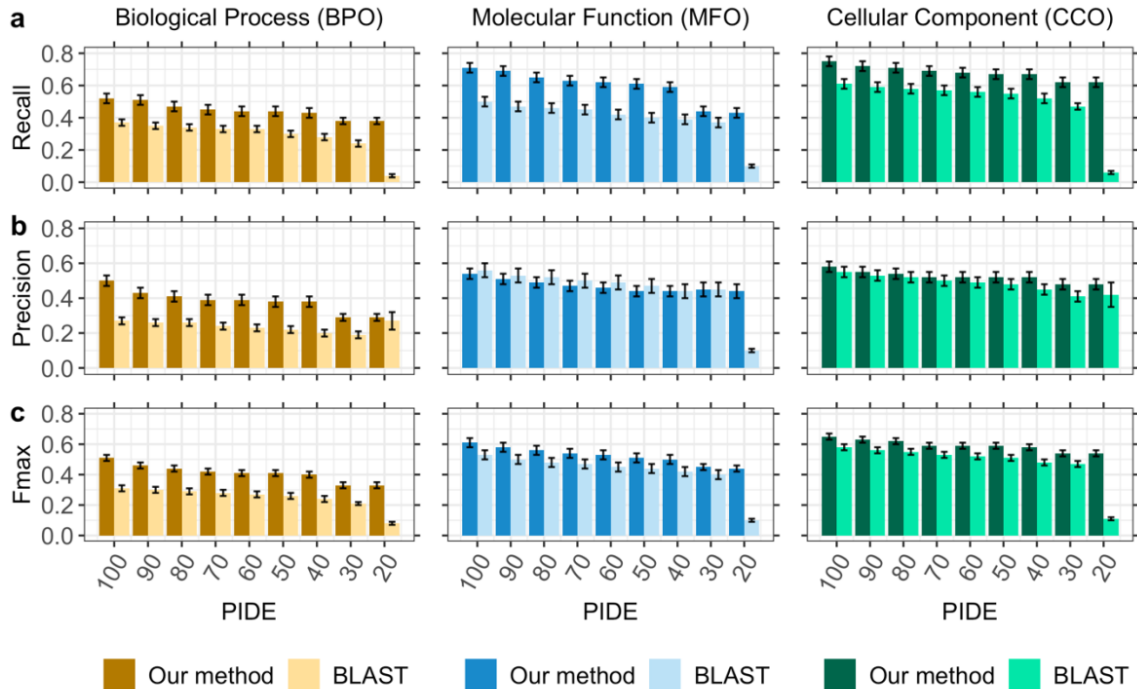
**Fig. S2:  $F_{\max}$  of our method for different protein lengths**

**Fig. S2: Performance not correlated with protein lengths.** We show the  $F_{\max}$  score for different intervals of varying protein length ranging from 100 to 1000 in steps of 100 for **a.** SeqVec and **b.** ProtBert. Protein embeddings for both models are derived by global average pooling over per-residue representations. This process could lead to an information loss for longer proteins. However, the performance of our method did not correlate with the length of the query protein.

**Fig. S3:  $F_{\max}$  for our method and CAFA3 competitors using LK evaluation mode**

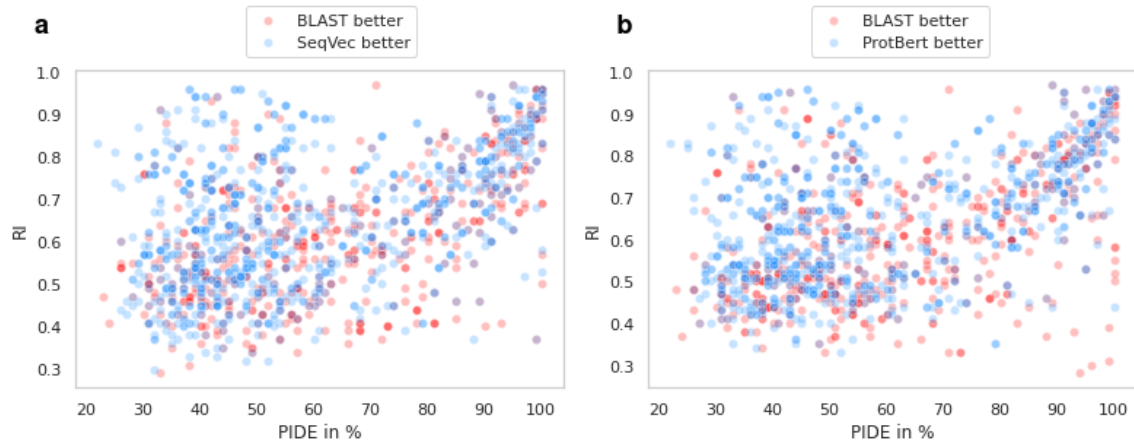
We compared the  $F_{\max}$  of our method (dark bar) for the three ontologies (BPO, MFO, CCO) to the top ten methods that did actually compete at CAFA3<sup>1</sup> and to two background approaches (lighter bars) also applied in the CAFA3 challenge for the LK evaluation mode using proteins for which some GO terms have been known before.



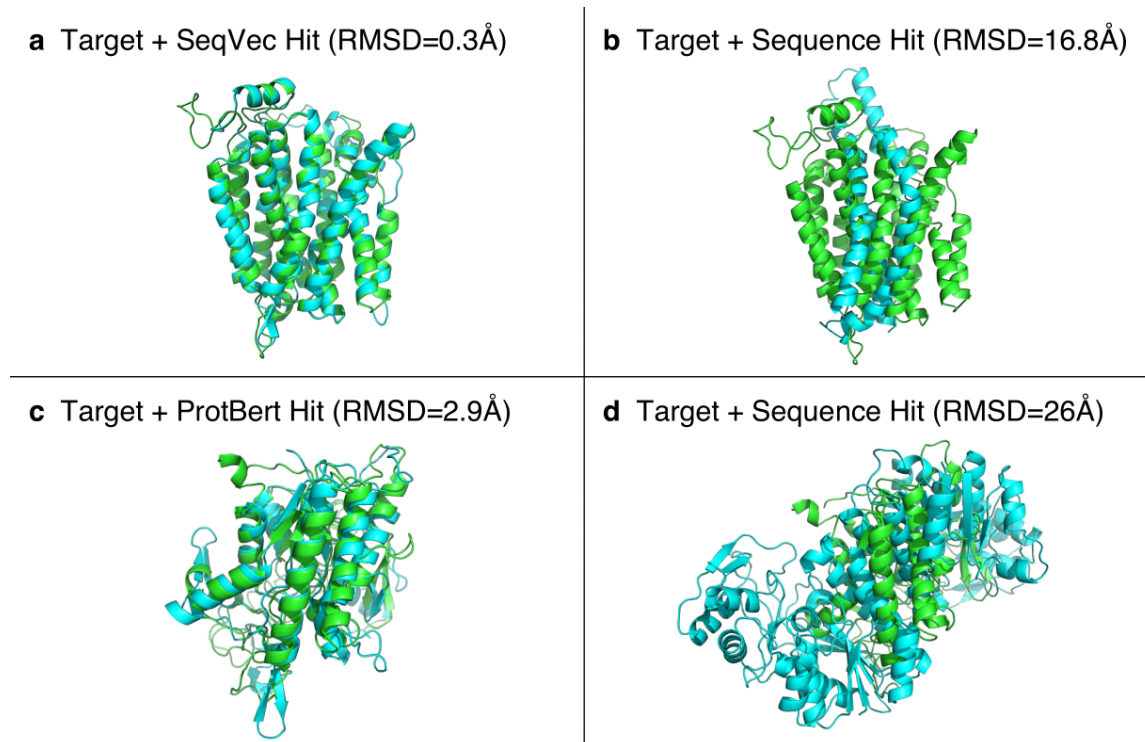
**Fig. S4:  $F_{\max}$ , precision, and recall for different lookup sets based on GOA2020**

To test how the level of percentage pairwise sequence identity (PIDE) affects the performance of our method and of homology-based inference (BLAST), we stepwise removed proteins from our lookup set (here: GOA-2020) if they shared more than a certain PIDE (as indicated on the x-axes) to any query protein. Panel **a** shows the recall, panel **b** the precision, and panel **c**  $F_{\max}$  for BPO, MFO, and CCO, respectively, for lookup sets of <100, 90, 80, 70, 60, 50, 40, 30, and 20% PIDE to the target proteins. Sequence-based transfer was accomplished by running BLAST against GOA2020 and transferring annotations from the hit with the highest PIDE (and PIDE <math>x\%</math> for the different lookup sets) in the local alignment (result marked by lighter colored bars, labeled as *BLAST*). Error bars indicate 95% confidence intervals.

**Fig. S5: Proteins for which embedding-based or homology-based inference worked better dependent of RI and PIDE**



Blue points indicate proteins for which embedding-based annotation transfer (**a.** SeqVec, **b.** ProtBert) worked better than homology-based inference (BLAST) (by at least four percentage points), red points indicate the reverse: proteins for which PIDE worked better than embeddings. The RI gives the embedding similarity, PIDE is percentage pairwise sequence identity. There is no clear relationship between PIDE, RI, and which of the two approaches worked better. Only for high RIs and low PIDE, embedding-based annotation transfer performed almost always better than PIDE.

**Fig. S6: Comparative model for two target proteins and the corresponding hits found through embedding- or sequence similarity**

We chose two example targets for which embedding-based inference worked better than homology-based inference and where the embedding similarity was high while the sequence similarity was low. This resulted in two protein triplets (target, embedding-similar hit, sequence-similar hit). For none of the six proteins, structures were available. **a.** Comparative modeling using Swiss-Model<sup>2</sup> mapped the first target with UniProt identifier P40445 and the corresponding hit found using SeqVec embeddings (P53322) to the same structural template of the D-galactonate-proton symporter of *E. coli* (6E9N<sup>3,4</sup>) with the resulting Swiss-Model models having a root-mean-square deviation (RMSD) of 0.3Å. **b.** The sequence-similar hit (Q7PMG3) for the same target mapped to the PDB structure 6C70 (Cryo-EM structure of Orco)<sup>3,5</sup> with RMSD=16.8Å to the structure of the target. **c.** For the second target (Q9STM6), the embedding-based hit (Q9SYF0) was found using ProtBert. Comparative modeling mapped target and hit to PDB structures 3KVN (autotransporter EstA from *Pseudomonas aeruginosa*)<sup>3,6</sup> and 5XTU (GDSL esterase of photobacterium sp. J15)<sup>3,7</sup> with RMSD=2.9Å. **d.** The modeled structure for the corresponding sequence-similar hit (Q564Q1) was based on the PDB structure 1KVQ (UDP-galactose 4-epimerase complexed with UDP-phenol)<sup>3,8</sup> with RMSD=26Å to the structure model of the target.

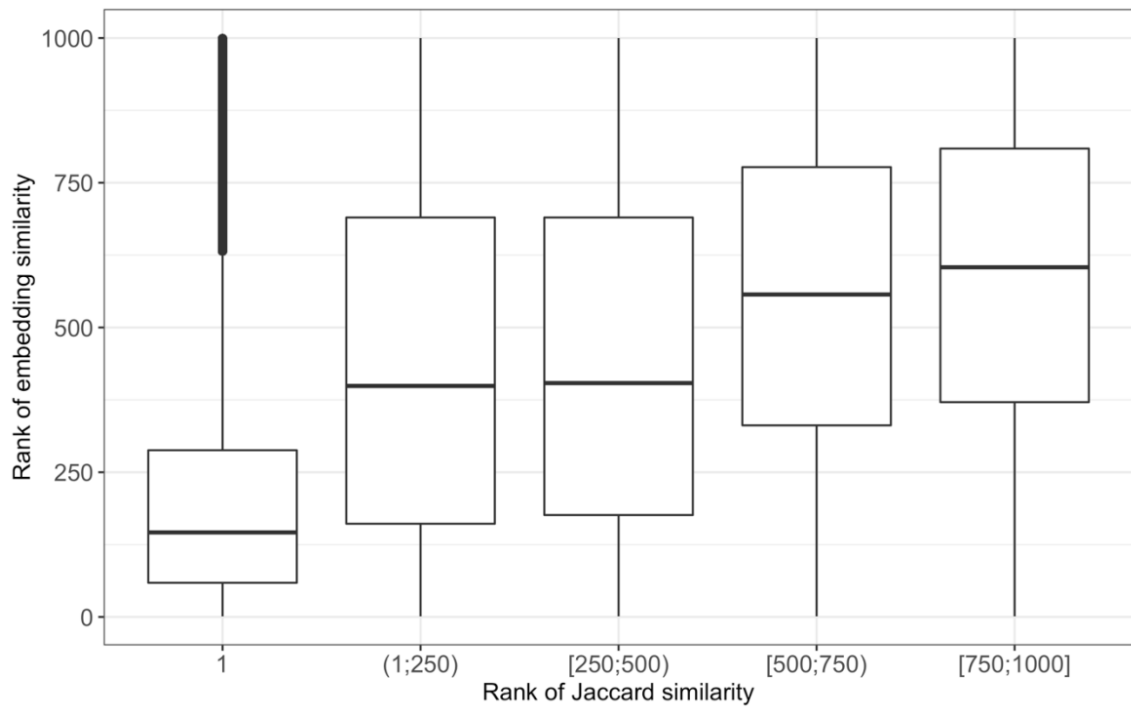
---

## Weak correlation between embedding similarity and GO term similarity

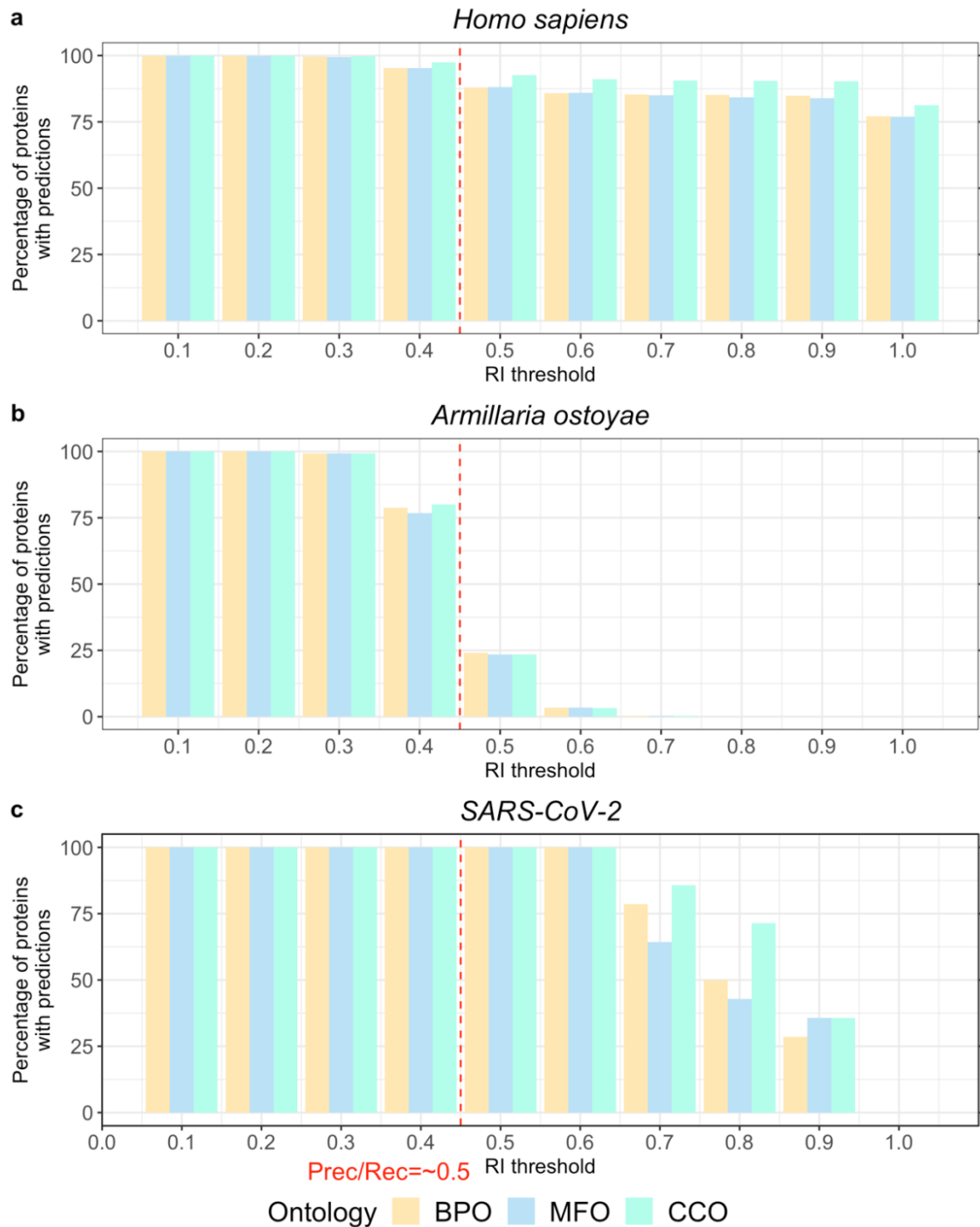
Our method relies on the assumption that proteins with similar embeddings share similar GO annotations. To evaluate this assumption, we randomly chose 5,000 proteins from the *GOA2017-100* set and calculated pairwise distances between these proteins and the remaining proteins in *GOA2017-100* (298,984 proteins) using SeqVec embeddings. We extracted the first 1,000 hits to avoid that our analysis is dominated by random noise for very distant protein pairs. This resulted in 5,000,000 pairs of proteins. For each of those pairs, we calculated the GO term similarity using the Jaccard index (Eqn. 7 in the main text).

The absolute embedding similarity does not necessarily have to correlate with the GO term similarity, i.e. if two proteins have a distance  $d_1$ , their GO annotations are not necessarily more similar than for two other proteins with a larger distance  $d_2$ . We rather assume that for a given query, the protein closest to it is more likely to have more similar GO terms than another hit farther away. Also, how similar the annotations of the most similar hit are differs between proteins. There are proteins for which we find hits in the lookup data set with identical annotations while for others, even the most similar hit might still be annotated to very different GO terms. Since both embedding distance and GO term similarity are not directly comparable between proteins, we converted our results to ranks, i.e. for a given query protein, we ordered all hits by distance and gave the hit with the smallest distance rank 1, the hit with the second-smallest distance rank 2, and so forth. We applied the same approach for the GO term similarity. For a perfect correlation, the closest protein (rank 1) should also have the most similar GO annotations (rank 1) and the most distant protein (rank 1000) should have the most dissimilar GO annotations (rank 1000).

For a query protein, its most similar hits (i.e. low ranks) are also more likely to be amongst the closest hits while more dissimilar hits also tend to be further away from the query in embedding space (Fig. S1). This observation corresponds to a weak correlation of  $\rho=0.28$  (Spearman's correlation coefficient, p-value  $< 2.2e-16$ ) between embedding similarity and GO term similarity. This weak correlation was expected to some extent. While we assume that proteins close in embedding space should have similar GO annotations, this does not necessarily imply that proteins far away in embedding space have dissimilar GO annotations.

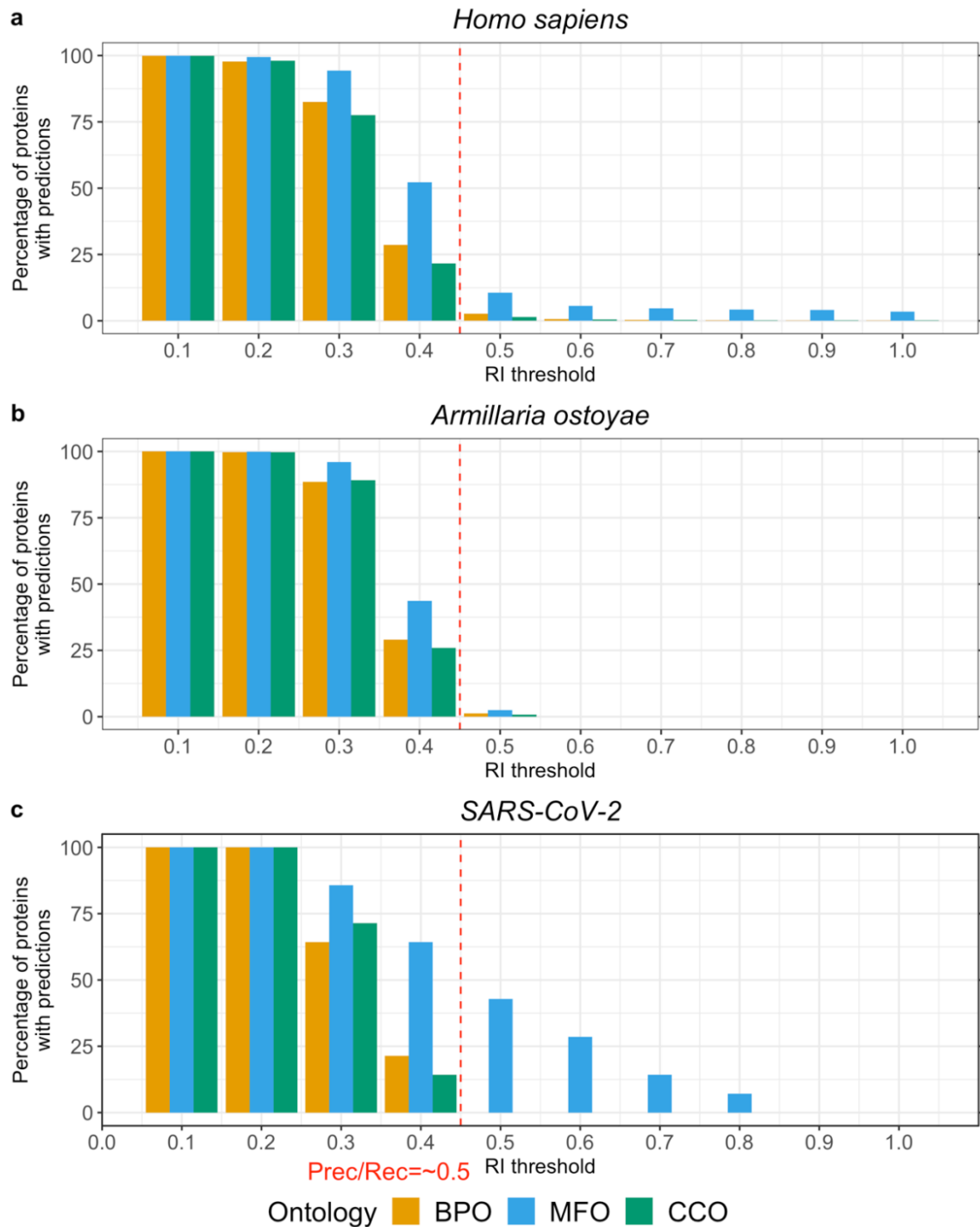
**Fig. S7: Embedding similarity for different levels of GO term similarity**

For each query protein, we extracted the 1000 closest proteins and assigned each hit a rank (i) according to its distance to the query (rank 1 = closest hit) and (ii) according to its Jaccard similarity between the GO annotations (rank 1 = most similar annotations). For a set of 5,000 proteins, the ranks for the Jaccard similarity were grouped as shown on the x-axis. On the y-axis, the ranks for the embedding distance are shown. For most proteins, the hit with the most similar annotation (most left box) is also one of the closest hits in embedding space.

**Fig. S8: Fraction of proteomes with predicted GO terms using lookup set GOA2020**

We applied our method to three proteomes (animal: *Homo sapiens*, fungus: *Armillaria ostoyae*, and virus: SARS-CoV-2) and monitored the fraction of proteins in each proteome

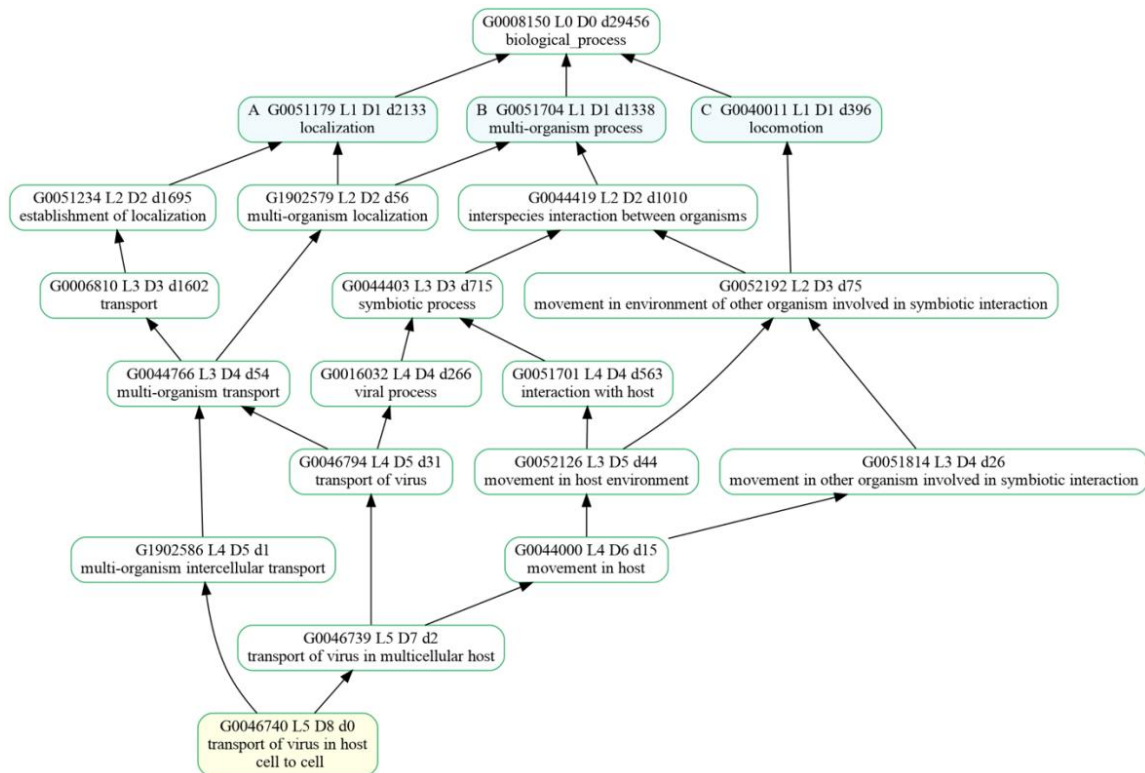
for which our method predicted GO terms for different thresholds in embedding similarity (RI, Eqn. 5 in main text). Intervals labelled by number ( $N=[0.1,1.0]$ ) average over all predictions (with  $N-0.1 < RI \leq N$ ). For  $RI=0.5$ , our method is expected to achieve roughly 50% precision and recall in all three ontologies (indicated by red dashed line). We used the set GOA2020 as lookup data set which also contains GO annotations not experimentally verified. **a.** The human proteome is well-studied (all 20,370 proteins are in Swiss-Prot) and for many proteins, GO term predictions can be obtained through self-hits, i.e. the annotations are taken from the query protein. Also, for proteins without any GO annotation, our method could predict GO terms at very high reliability probably because there are many well-studied model organisms with experimentally annotated orthologous proteins. **b.** The proteome of the fungus *Armillaria ostoyae* appears more exotic (0.01% of the 22,192 proteins were in Swiss-Prot); high-reliability ( $RI > 0.7$ ) predictions of GO terms were available for few proteins, and for  $RI > 0.5$ , GO terms were predicted for fewer than half of the proteome. **c.** While annotations were unknown for most proteins of the novel virus SARS-CoV-2 (no coverage at  $RI=1$ ), many annotations could be transferred from the human SARS coronavirus (SARS-CoV) and the bat coronavirus HKU3 (BtCoV) allowing GO term predictions for all proteins at reliability values as high as  $RI=0.6$ .

**Fig. S9: Fraction of proteomes with predicted GO terms using lookup set GOA2020X**

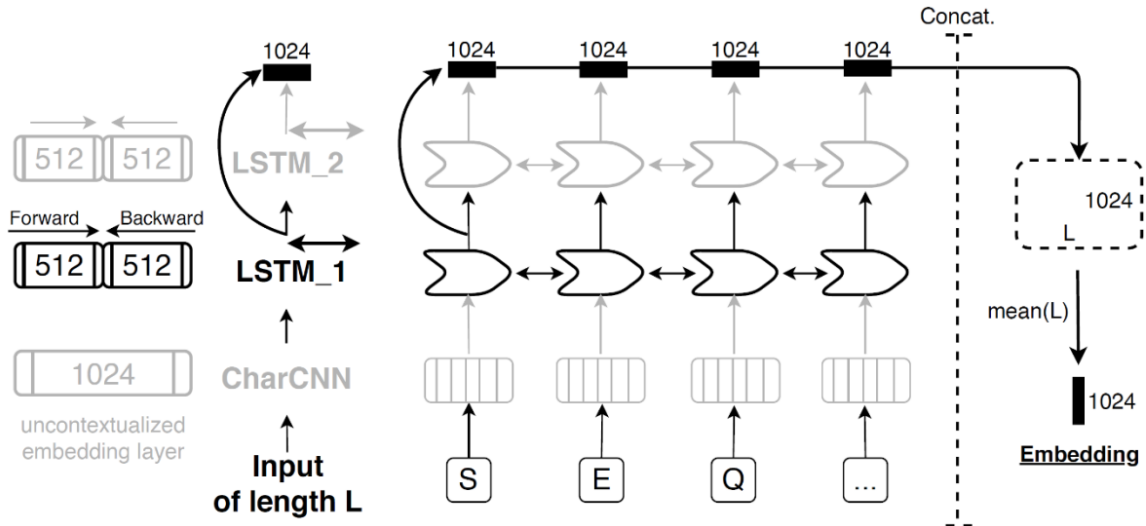
We applied our method to three proteomes (animal: *Homo sapiens*, fungus: *Armillaria ostoyae*, and virus: SARS-CoV-2) as described in Fig. S6. Instead of GOA2020, we used



GOA2020X as lookup data set which only contains experimentally verified annotations. **a.** While the human proteome is in general well-studied, most proteins lack GO annotations (almost no annotation transfer at  $RI=1.0$ ). **b.** For the proteome of the fungus *Armillaria ostoyae*, almost no experimental GO annotations could be transferred at high reliability ( $RI>0.5$ ). **c.** While SARS-CoV-2 is a very novel and therefore not well-studied proteome, for almost 50% of the proteins, GO annotations for MFO could be inferred at  $RI>0.5$  from the human SARS coronavirus (SARS-CoV) and the bat coronavirus HKU3 (BtCoV) which are both more well-studied.

**Fig. S10: Visualization of predicted GO term for Nsp7b from SARS-CoV-2**

To further analyze the predicted GO terms, the leaf terms can be visualized in the GO hierarchy. For the prediction of “GO:0046740” for the non-structural protein 7b from SARS-CoV-2, this visualization revealed that the functionality of this protein constituted two main components: The interaction with the host and the actual transportation.

**Fig. S11: Visualization of the embedding generation using SeqVec.**

We outline the process of generating fixed-size embeddings for protein sequences of variable length using SeqVec. The example illustrated here shows how the first three residues of protein sequence (“SEQ..”) are processed: first, the three layers of SeqVec (uncontextualized: CharCNN; contextualized: LSTM layer 1 and LSTM layer 2) project the protein sequence to vector space. The CharCNN generates vectors of size 1024 without considering neighboring residues. In contrast to this, the two LSTM layers process the sequence in both directions, each creating a vector of size 512. The vectors of both directions are concatenated for each LSTM independently, resulting in an embedding size of 1024 for each LSTM layers. In a second step, only embeddings of the first LSTM layer are extracted, concatenated and averaged over the length of the protein (global average pooling), resulting in a 1024-dimensional embedding.

**Table S1: Correlation between  $F_{\max}$  and protein length**

	<b>Spearman's correlation coefficient</b>	<b>P-value</b>
<b>SeqVec</b>	-0.03	0.27
<b>ProtBert</b>	-0.03	0.16
<b>BLAST</b>	0.05	0.06

\* Global average pooling for long sequences could lead to information loss because important information might be averaged out if the signal is not consistent over the protein length. However, we did not observe a correlation between protein length and performance for either SeqVec or ProtBert. The same holds true for homology-based inference ("BLAST"). Correlation was measured using Spearman's rank correlation coefficient.

**Table S2:  $F_{\max}$  and average number of predicted GO terms for different values of  $k$** 

	$F_{\max}$		
	<b>BPO</b>	<b>MFO</b>	<b>CCO</b>
<b>k=1</b>	37±2%	50±3%	57±2%
<b>k=2</b>	37±2%	51±2%	58±2%
<b>k=3</b>	37±2%	50±2%	58±2%
<b>k=4</b>	37±2%	51±2%	58±2%
<b>k=5</b>	36±2%	51±2%	59±2%
<b>k=10</b>	36±2%	49±3%	56±2%

\* The number of neighbors included for the annotation transfer did not affect  $F_{\max}$ . Evaluation was performed for the no-knowledge (NK) set of proteins for which no annotations in any ontology were available at the submission deadline of CAFA3. Error estimates indicate 95% confidence intervals.

**Table S3: Precision, recall, and average number of predicted GO terms for different values of  $k$** 

	Precision (shown as percentages)			Recall (shown as percentages)			Average number of predicted GO terms per protein		
	BPO	MFO	CCO	BPO	MFO	CCO	BPO	MFO	CCO
<b>k=1</b>	32±2	47±3	54±3	43±3	54±3	62±3	46.5	11.7	15.7
<b>k=2</b>	28±2	40±3	49±3	49±2	60±3	69±3	66.3	15.4	19.8
<b>k=3</b>	25±2	37±3	46±2	52±2	62±3	72±3	80.6	18.0	22.7
<b>k=4</b>	23±2	35±3	43±2	55±2	64±3	75±2	97.6	20.1	25.8
<b>k=5</b>	21±2	32±3	40±2	57±2	66±3	78±2	113.5	22.4	29.0
<b>k=10</b>	16±1	24±2	30±2	65±2	72±3	83±2	183.5	33.4	43.2

\* The average number of predicted GO terms per protein increased with increasing values of  $k$  when all predictions are taken into account. This increase in the number of predicted terms led to a decrease in precision, but to an increase in recall. Taking more proteins than the closest one into account can be beneficial to e.g., increase the specificity and quality of the predicted terms if only a few, unspecific terms are annotated to the closest hit. Evaluation was performed for the no-knowledge (NK) set of proteins for which no annotations in any ontology were available at the submission deadline of CAFA3. Error estimates indicate 95% confidence intervals.

**Table S4: F<sub>max</sub> for different combinations of embedding-based and homology-based annotation transfer**

	F <sub>max</sub>		
	BPO	MFO	CCO
SeqVec-2020	51±2%	61±3%	65±2%
SeqVec/ProtBert	51±2%	60±2%	66±2%
SeqVec/BLAST	50±2%	61±2%	65±2%
ProtBert/BLAST	49±2%	60±2%	65±2%
SeqVec/ProtBert/BLAST	51±2%	61±2%	66±2%

\* Combining embedding-based annotation transfer for different language models (SeqVec, ProtBert) and homology-based transfer (BLAST) did not improve performance over embedding-based transfer using SeqVec (SeqVec-2020). For this combination, every term predicted for either of the methods was also include in the final, combined prediction. Maybe this approach is too simple to reflect the complex relationship between embedding similarity, sequence similarity and prediction quality and only a more sophisticated combination could improve over the single method (SeqVec).

**Table S5: Annotated and predicted GO terms for SARS-CoV-2 proteins**

	Annotated			Predicted		
	BPO	MFO	CCO	BPO	MFO	CCO
<b>P0DTC1</b>	GO:0006508 GO:0016032 GO:0030683 GO:0039502 GO:0039503 GO:0039520 GO:0039548 GO:0039579 GO:0039595 GO:0039648 GO:0039657 GO:0090305	GO:0003723 GO:0004518 GO:0004519 GO:0008233 GO:0008234 GO:0016787 GO:0036459 GO:0046872	GO:0016020 GO:0016021 GO:0030430 GO:0033644 GO:0044220	GO:0001172 GO:0006508 GO:0019079 GO:0019082 GO:0039502 GO:0039520 GO:0039548 GO:0039579 GO:0039595 GO:0039648 GO:0090305	GO:0036459 GO:0003723 GO:0003968 GO:0004197 GO:0004519 GO:0008242 GO:0008270	GO:0016020 GO:0016021 GO:0030430 GO:0033644 GO:0044220
<b>P0DTD1</b>	GO:0001172 GO:0006508 GO:0016032 GO:0030683 GO:0032259 GO:0032508 GO:0039502 GO:0039503 GO:0039520 GO:0039579 GO:0039595 GO:0039644 GO:0039648 GO:0039657 GO:0090305	GO:0000166 GO:0003678 GO:0003723 GO:0003724 GO:0003968 GO:0004386 GO:0004518 GO:0004519 GO:0004527 GO:0005524 GO:0008168 GO:0008233 GO:0008234 GO:0016740 GO:0016779 GO:0016787 GO:0036459 GO:0046872	GO:0016020 GO:0016021 GO:0030430 GO:0033644 GO:0044172 GO:0044220	GO:0001172 GO:0006351 GO:0006508 GO:0019082 GO:0019083 GO:0032259 GO:0032508 GO:0039502 GO:0090503 GO:0039520 GO:0039579 GO:0039595 GO:0039644 GO:0039648 GO:0039694	GO:0000175 GO:0003678 GO:0003723 GO:0003724 GO:0003968 GO:0004197 GO:0004519 GO:0005524 GO:0008168 GO:0008242 GO:0008270 GO:0036459 GO:0042802	GO:0016020 GO:0016021 GO:0030430 GO:0033644 GO:0044172 GO:0044220
<b>P0DTC2</b>	GO:0009405 GO:0016032 GO:0019062 GO:0039654 GO:0039663 GO:0044650 <b>GO:0046718</b>	<b>GO:0005515</b> <b>GO:0046789</b>	GO:0016020 GO:0016021 GO:0019012 GO:0019031 GO:0020002 GO:0033644 GO:0044173 GO:0055036	GO:0009405 GO:0019064 GO:0039654 <b>GO:0046718</b> GO:0046813 GO:0075509	GO:0042802 <b>GO:0046789</b>	GO:0016021 GO:0019012 GO:0019031 GO:0020002 GO:0044173 GO:0055036
<b>P0DTC3</b>	No annotations		GO:0005576 GO:0016020 GO:0016021 GO:0019012 GO:0020002 GO:0030430 GO:0033644 GO:0044177 GO:0044178	GO:0034220 GO:0039707 GO:0051259	GO:0005216	GO:0005576 GO:0016020 GO:0016021 GO:0019012 GO:0020002 GO:0030430 GO:0044177 GO:0044178 GO:0044385

	Annotated			Predicted		
	BPO	MFO	CCO	BPO	MFO	CCO
<b>P0DTC4</b>	No annotations		GO:0016020 GO:0016021 GO:0033644 GO:0044177 GO:0044178	GO:0044662 GO:0046760	GO:0015078	GO:0016020 <u>GO:0016021</u> GO:0044172 GO:0044177 GO:0044178
<b>P0DTC5</b>	GO:0016032 GO:0030683	GO:0039660	GO:0016020 GO:0016021 GO:0019012 GO:0019031 GO:0033644 GO:0044177 GO:0044178 GO:0055036	GO:0019058 <u>GO:0030683</u>	<u>GO:0039660</u>	GO:0016021 <u>GO:0019012</u> <u>GO:0019031</u> GO:0030430 GO:0044177 GO:0044178 <u>GO:0055036</u>
<b>P0DTC6</b>	GO:0009405	No annotations	GO:0016020 GO:0033644 GO:0044165 GO:0044167 GO:0044177 GO:0044178	<u>GO:0009405</u>	GO:0005125 GO:0005126	GO:0016020 <u>GO:0044165</u> <u>GO:0044167</u> <u>GO:0044177</u> <u>GO:0044178</u>
<b>P0DTC7</b>	GO:0016032 GO:0039646 GO:0060153	No annotations	GO:0016020 GO:0016021 GO:0019012 GO:0033644 GO:0044165 GO:0044167 GO:0044173 GO:0044177 GO:0044178	<u>GO:0039646</u>	GO:0005515 GO:0005537	GO:0016020 <u>GO:0016021</u> <u>GO:0019012</u> <u>GO:0044165</u> <u>GO:0044167</u> <u>GO:0044173</u> <u>GO:0044177</u> <u>GO:0044178</u>
<b>P0DTD8</b>	No annotations		GO:0016020 GO:0016021 GO:0033644	GO:0046740	GO:0015078	GO:0016020 <u>GO:0016021</u> <u>GO:0033644</u>
<b>P0DTC8</b>	No annotations			GO:0006954 GO:0031666 GO:0045087	GO:0005515	GO:0005576 GO:0005615
<b>P0DTC9</b>	No annotations	GO:0003723	GO:0019012 GO:0019013 GO:0044172 GO:0044177	GO:0000413 GO:0006457 GO:0016567	<u>GO:0003723</u>	GO:0019012 <u>GO:0019013</u> GO:0030430 <u>GO:0044172</u> <u>GO:0044177</u> GO:0044220
<b>P0DTD2</b>	No annotations		GO:0016020 GO:0030430 GO:0033644 GO:0044161 GO:0044162	GO:0006412	GO:0000049 GO:0003735	GO:0016020 <u>GO:0030430</u> <u>GO:0044161</u> <u>GO:0044162</u>
<b>A0A663DJA2</b>	No annotations			GO:0009734 GO:0010930 GO:0048364	GO:0005506 GO:0009055 GO:0020037	GO:0016020 GO:0016021

\* Of the 14 proteins of SARS-CoV in UniProt, 7 have annotations in BPO, 6 in MFO and 12 in CCO. Of the 138 GO terms annotated, only three are experimentally verified. We predicted 83% of the annotations for BPO, 47% for MFO, and 87% for CCO. Bold terms indicate experimentally verified annotations; underlined terms indicate predicted terms which are also annotated.



**Table S6: Datasets for similarity lookup at different sequence identity thresholds.**

	<b>GOA2017</b>	<b>GOA2020</b>	<b>%identical proteins</b>	<b>Average sequence identity</b>
<b>Full</b>	307,287	295,558		
<b>100% Seq. ID.</b>	303,984	292,059	87%	44%
<b>90% Seq. ID.</b>	302,052	290,030	87%	44%
<b>80% Seq. ID.</b>	300,571	288,561	87%	44%
<b>70% Seq. ID.</b>	298,677	286,748	87%	44%
<b>60% Seq. ID.</b>	295,823	284,043	87%	43%
<b>50% Seq. ID.</b>	290,094	278,711	87%	43%
<b>40% Seq. ID.</b>	281,539	270,691	87%	43%
<b>30% Seq. ID.</b>	262,440	260,056	72%	43%
<b>20% Seq. ID.</b>	242,154	242,261	63%	43%

\* The first two columns show the size of data sets extracted from the Gene Ontology Annotation (GOA) database <sup>9-11</sup> in January 2017 (GOA2017) and January 2020 (GOA2020). The first row shows the full data set when only considering sequences from Swiss-Prot <sup>12</sup> and removing proteins only annotated to the roots of the three ontologies. The remaining rows represent data sets redundancy reduced against the CAFA3 targets at sequence identity thresholds of 100, 90, 80, 70, 60, 50, 40, 30 and 20%, respectively. Redundancy reduction was performed using CD-HIT and PSI-CD-HIT <sup>13,14</sup>. The last two columns show the agreement between the two GOA versions, *%identical proteins* gives the percentage of identical proteins in both sets (by UniProt identifier), and *Average sequence identity* states the average sequence identity of all protein pairs.

## References for Supporting Online Material

- 1 Zhou, N. *et al.* The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol* **20**, 244, doi:10.1186/s13059-019-1835-8 (2019).
- 2 Waterhouse, A. *et al.* SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res* **46**, W296-W303, doi:10.1093/nar/gky427 (2018).
- 3 Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res* **28**, 235-242, doi:10.1093/nar/28.1.235 (2000).
- 4 Leano, J. B. *et al.* Structures suggest a mechanism for energy coupling by a family of organic anion transporters. *PLoS Biol* **17**, e3000260, doi:10.1371/journal.pbio.3000260 (2019).
- 5 Butterwick, J. A. *et al.* Cryo-EM structure of the insect olfactory receptor Orco. *Nature* **560**, 447-452, doi:10.1038/s41586-018-0420-8 (2018).
- 6 van den Berg, B. Crystal structure of a full-length autotransporter. *J Mol Biol* **396**, 627-633, doi:10.1016/j.jmb.2009.12.061 (2010).
- 7 Mazlan, S. *et al.* Crystallization and structure elucidation of GDSL esterase of *Photobacterium* sp. J15. *Int J Biol Macromol* **119**, 1188-1194, doi:10.1016/j.ijbiomac.2018.08.022 (2018).
- 8 Thoden, J. B., Gulick, A. M. & Holden, H. M. Molecular structures of the S124A, S124T, and S124V site-directed mutants of UDP-galactose 4-epimerase from *Escherichia coli*. *Biochemistry* **36**, 10685-10695, doi:10.1021/bi9704313 (1997).
- 9 GOA, <<http://www.ebi.ac.uk/GOA>> (2020).
- 10 Camon, E. *et al.* The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. *Nucleic Acids Res* **32**, D262-266, doi:10.1093/nar/gkh021 (2004).
- 11 Huntley, R. P. *et al.* The GOA database: gene Ontology annotation updates for 2015. *Nucleic Acids Res* **43**, D1057-1063, doi:10.1093/nar/gku1113 (2015).
- 12 Boutet, E., Lieberherr, D., Tognolli, M., Schneider, M. & Bairoch, A. UniProtKB/Swiss-Prot. *Methods Mol Biol* **406**, 89-112, doi:10.1007/978-1-59745-535-0\_4 (2007).
- 13 Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150-3152, doi:10.1093/bioinformatics/bts565 (2012).

- 14 Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658-1659, doi:10.1093/bioinformatics/btl158 (2006).



## 4. Prediction of Protein Binding Residues from Sequence

### 4.1. Evolutionary Couplings and Sequence Variation Effect Predict Protein Binding Sites

#### 4.1.1. Preface

Binding of proteins to ligands such as small molecules, metal ions, or macromolecules like DNA, RNA, and other proteins is one important aspect of the molecular function of proteins [2]. However, with experimentally verified binding annotations unknown for most protein sequences, we have to rely on prediction methods.

We developed *bindPredictML17*, a method based on an Artificial Neural Network (ANN) that predicts binding residues mainly relying on evolutionary information derived from evolutionary couplings [116, 117] and variant effect predictions [67, 118]. Evolutionary couplings have proven to be a powerful concept to predict protein structure [116, 117] and it has also been hypothesized that they could capture functional information [117, 119]. We calculated evolutionary couplings using EVcouplings [116, 117, 120]. Variant effect predictors allow to predict whether a mutation from the native amino acid  $X$  at a certain position to any other amino acid  $Y$  will have an effect on the protein or not. Changing a protein residue involved in binding often leads to a change in this protein's function. Therefore, such mutations would be predicted to have a strong effect and, consequently, these predictions can be an indicator for the functional relevance of a residue. Variant effects were predicted using SNAP2 [67] and EVmutation [118].

*bindPredictML17* achieved a performance of  $F1 = 26.2 \pm 0.8\%$  clearly outperforming a random approach as well as a simple method combining the same input features as

a weighted sum instead of using Machine Learning (ML). Although our method was limited by the small data set and possibly missing annotations of binding residues, the predicted binding residues formed spatial clusters in the protein indicating that wrongly predicted binding residues could depict unknown binding sites. bindPredictML17 is solely based on sequence information allowing predictions for proteins without known structures. The source code of the method is made available as a GitHub repository: <https://github.com/Rostlab/bindPredict>.

**Author contribution:** I implemented bindPredictML17, performed the detailed performance assessment of the method, and did the majority of manuscript writing. Thomas A. Hopf helped with running EVcouplings and interpreting its results. All authors drafted the manuscript.

#### 4.1.2. Journal Article: Schelling *et al.*, **Proteins: Structure, Function, and Bioinformatics (2018)**

**Reference:** Schelling, M., Hopf, T. A., and Rost, B. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins: Structure, Function, and Bioinformatics*, 86(10):1064–1074, 2018. doi:10.1002/prot.25585

## RESEARCH ARTICLE

## Evolutionary couplings and sequence variation effect predict protein binding sites

Maria Schelling<sup>1</sup>  | Thomas A. Hopf<sup>1,2</sup> | Burkhard Rost<sup>1,3,4,5</sup><sup>1</sup>TUM (Technical University of Munich) Department of Informatics, Bioinformatics, & Computational Biology - i12, Garching/Munich, Germany<sup>2</sup>Department of Systems Biology & Department of Cell Biology, Harvard Medical School, Boston, Massachusetts<sup>3</sup>Institute for Advanced Study (TUM-IAS), Garching/Munich, Germany<sup>4</sup>TUM School of Life Sciences Weihenstephan (WZW), Freising, Germany<sup>5</sup>Department of Biochemistry and Molecular Biophysics & New York Consortium on Membrane Protein Structure (NYCOMPS), Columbia University, New York, New York**Correspondence**Maria Schelling,  
TUM (Technical University of Munich)  
Department of Informatics, Bioinformatics & Computational Biology - i12,  
Boltzmannstr. 3, 85748 Garching/Munich,  
Germany.  
Email: mschelling@rostlab.org**Abstract**

Binding small ligands such as ions or macromolecules such as DNA, RNA, and other proteins is one important aspect of the molecular function of proteins. Many binding sites remain without experimental annotations. Predicting binding sites on a per-residue level is challenging, but if 3D structures are known, information about coevolving residue pairs (evolutionary couplings) can predict catalytic residues through mutual information. Here, we predicted protein binding sites from evolutionary couplings derived from a global statistical model using maximum entropy. Additionally, we included information from sequence variation. A simple method using a weighted sum over eight scores substantially outperformed random (F1 = 19.3% ± 0.7% vs F1 = 2% for random). Training a neural network on these eight scores (along with predicted solvent accessibility and conservation in protein families) improved substantially (F1 = 26.2% ± 0.8%). Although the machine learning was limited by the small data set and possibly wrong annotations of binding sites, the predicted binding sites formed spatial clusters in the protein. The source code of the binding site predictions is available through GitHub: <https://github.com/Rostlab/bindPredict>.

**KEYWORDS**

binding site, coevolution, evolutionary couplings, machine learning, neural network, prediction, sequence variation

**1 | INTRODUCTION**

Determining protein function is crucial to understand the molecular mechanisms of life.<sup>1</sup> Nevertheless, molecular function remains experimentally unknown for most proteins and de novo predictions remain challenging.<sup>2</sup> One aspect of molecular function is binding: almost all key cell processes require proteins binding to other molecules.<sup>3</sup> These molecules are referred to as *ligands*, including ions, small molecules, or macromolecules such as DNA, RNA, and other proteins. Mostly one protein specifically binds selected ligands. Often only a few key residues determine this specificity. Mutations of those key residues tend

to disrupt function. We might distinguish three types of binding sites with very different features: (i) catalytic sites in enzymes that bind small molecules, (ii) binding interfaces for large molecules such as DNA or RNA, and (iii) binding of other proteins. In this work, we have focused on representatives from the first two classes for which we had reliable binding annotations: for enzymes and DNA-binding.

**1.1 | Evolutionary couplings can predict binding**

Genetic variation drives evolution; natural selection acts on the level of the corresponding phenotype.<sup>4</sup> Understanding how the genotype determines the phenotype is important for characterizing evolutionary processes and for identifying the phenotypic consequences of mutations. The first step toward understanding how the genotype determines the phenotype is the analysis of the effect of single sequence variants (SAV: Single Amino Acid Variant) on the protein. The effect of a mutation on the phenotype can be influenced by variants at other positions.<sup>5</sup> This phenomenon is referred to as epistasis. For instance,

Abbreviations: ANN, artificial neural network; AUC, area under the receiver operating curve (ROC-curve); CCS, cumulative coupling scores; DI, direct information; ECs, evolutionary couplings; EVmutation, method predicting impact of sequence variation using ECs; MI, mutual information; MSA, multiple sequence alignment; PDB, Protein Data Bank; PDIdb, Protein-DNA-Interface Database; RI, reliability index; ROC-curve, receiver operating characteristic-here plotted as true positive rate vs false positive rate; SAV, single amino acid variant

epistatic interactions could lead to a compensation of deleterious mutants by additional, compensatory mutations.

Evolutionary information can help to elucidate the genotype-phenotype relation and to identify epistatic interactions. Proteins with similar sequences belonging to the same family tend to be subject to similar constraints caused by natural selection. Residues conserved between all sequences are prone to be functionally important sites. Epistatic interactions between two residues suggest that the change of one residue has to be compensated by the change of the other to maintain function. In particular, residues close in space or residues involved in protein-ligand interactions are subject to coevolution.<sup>6</sup> Because these residues are linked in the constraints they together put upon evolution, they are referred to as *evolutionary couplings* (for brevity: ECs in this work). The correct identification of ECs from sequence information alone remains challenging, but if it succeeds, de novo predictions of protein structure and functional hotspots become possible.<sup>7</sup> ECs are often used to predict protein structure but can also help in the prediction of function.<sup>6,8</sup> Using mutual information (MI) as marker for ECs, it has been claimed that catalytic residues in enzymes can be predicted given protein sequence and structure as input.<sup>9</sup> It has, also, been shown that networks of residues constructed from spatial information and MI contain information about functional residues.<sup>10</sup>

Here, we present a new method predicting binding site residues through evolutionary couplings (ECs) from sequence alone. ECs are calculated using EVcouplings<sup>7</sup> which implements a global statistical model differing from previous approaches using MI. Enzymes and DNA-binding proteins serve as proxies for binding sites because for these proteins molecular function is largely defined by the binding site, and because they represent opposites of the spectrum, that is, the binding of small (enzymes) and large ligands (DNA-binding). Furthermore, we picked two diverse groups for which detailed experimental data was available for large-scale analysis. This was particularly important because the application of ECs already reduced the data set substantially due to its need for very large alignments. Different interpretations of the evolutionary couplings to extract and use information about the binding site are analyzed and combined with information derived from sequence variation. A simple average over relevant information provides a baseline prediction. This statistical baseline performed less well than a machine learning solution using a simple artificial neural network (ANN).

## 2 | METHODS

### 2.1 | Data set

The data set included only enzymes and DNA-binding proteins. Enzymes were extracted from Swiss-Prot<sup>11</sup> only using sequences with an EC number (Enzyme Commission number)<sup>12</sup> and at least one reference to a structure in the Protein Data Bank (PDB).<sup>13</sup> DNA-binding proteins were extracted from the Protein-DNA Interface Database (PDIdb)<sup>14</sup> containing only sequences for which at least one structure is available in the PDB. PDIdb contains 922 entries referring to 922 protein structures with known protein-DNA interfaces. These

922 structures can be mapped to 272 unique Swiss-Prot entries. The combined data set (enzymes + DNA-binding) was redundancy reduced with Unique-Prot<sup>15</sup> using an HVAL < 0 to obtain an unbiased data set. This implied a maximum pairwise sequence identity of 20% for alignments of more than 250 residues.<sup>16</sup>

For enzymes, all residues annotated in the PDB as binding (as given by the "SITE" and the "SITE\_DESCRIPTION" in the REMARK800 in PDBx files or the "struct\_site.gen" and "struct\_site.details" as given in mmCIF) were considered; this included catalytic sites, co-factors, or metal-binding sites. If multiple structures for a sequence were available, the binding site annotations of all structures were used. For DNA-binding proteins, PDIdb provided the binding sites by giving detailed interface data that can be downloaded for all 922 entries in PDIdb ([http://melolab.org/pdibd/web/download/pdibd\\_dat.tar.gz](http://melolab.org/pdibd/web/download/pdibd_dat.tar.gz)). Some sequences map to more than one structure in PDIdb, and for some structures, PDIdb describes more than one interface.<sup>14</sup> We took all structures and all described interfaces into account and annotated every residue as binding which is part of one of the DNA-protein interfaces given in PDIdb.

Some structures are oligomers also including heteromers (for both: enzymes and DNA-binding proteins). To annotate the binding site of a heteromer, only those chains of the structure mapping to the given sequence from Swiss-Prot were considered and their binding site annotations were used for this protein. All residues not labeled as binding were considered as non-binding in our evaluation.

The final sequence-unique data set contained 412 proteins (357 enzymes, 55 DNA-binding) corresponding to 3027 different PDB structures. These data corresponded to 80 385 residues: 9483 binding (12%) and 70 902 (88%) non-binding (detailed distribution for binding sites in the Supporting Information Figure S1).

### 2.2 | ECs: Evolutionary couplings

EVcouplings<sup>6-8,17</sup> infers ECs between pairs of residues in a protein from a multiple sequence alignment (MSA) by computing a maximum entropy model providing Direct Information (DI) scores that represent the ECs. EVcouplings calculates alignments using jackhammer.<sup>18</sup> To ensure high quality results for the calculation of EC scores, EVcouplings considers positions (residues) in the alignment for which at least 70% of all aligned sequences have an amino acid (as opposed to a gap).<sup>8</sup> Alignments also have to contain enough members and enough diversity, and to overall cover a substantial fraction of the query (ideally entire structural domains; short regions or motifs are not enough). Alignments for queries of length L (number of residues/positions used for the probability model) have to have  $\geq 3L$  sequences covering  $\geq 0.7L$  residues to provide reasonable results.<sup>8</sup> Proteins/families not meeting any of those criteria were excluded from the analysis. For the remaining proteins, the distribution of alignment lengths and diversity of sequences is given in Supporting Information Figure S2. Using the MSAs calculated with EVcouplings, DI scores were calculated through FreeContact.<sup>19</sup> DI scores are only one way to infer ECs from sequence alignments; many other methods exist (eg, Refs. 20 and 21). It has been shown that pseudo-likelihood maximization Direct Coupling Analysis (plmDCA) to infer ECs performs best for structure prediction.<sup>20</sup> Nevertheless, here we used mean-field DCA (mfDCA) because



it worked better to predict binding sites (plmDCA vs mfDCA in SOM Supporting Information Table S1).

The method EVcouplings provides scores for all possible pairs of residues in a protein, but not all of these pairs are actually evolutionary coupled. According to the developers of EVcouplings, the number of chosen pairs should scale monotonically with the length  $L$  of the protein.<sup>6</sup> We tested different numbers of pairs (0.5  $L$ ,  $L$ , 2  $L$ , 3  $L$ ) and observed that the calculation of our scores described below works best when using the top 2  $L$  scores (Supporting Information Table S1).

Binding residues tend to be on the protein surface. Thus, all coupled residue pairs for which both residues were in the protein core were removed. Solvent accessibility was predicted by PROFacc<sup>22,23</sup> and was normalized to relative solvent accessibility values using the maximum solvent accessibility value for each amino acid as given in Ref.<sup>24</sup> to obtain comparable values.

Most residues relevant for a binding site are confined to a local region in 3D. Thus, pairs with high EC scores distant in space were filtered out. Distances were inferred from structures of sequences similar to the query. Structures were chosen using a bitscore threshold. If more than one structure or chain was available for a protein sequence, the distance between two residues  $i$  and  $j$  was defined as the minimal distance over all structures. If no similar structure were found, no distances were provided for the query protein.

Another filter compiled cumulative coupling scores (CCS).<sup>8</sup> The CCS of residue  $i$  was calculated by summing the ECs over all highly coupled pairs  $P$  involving that residue (Equation (1)).  $P$  was either the original list of highly coupled pairs or a list of pairs filtered by solvent accessibility or distance.

$$CCS(i) = \sum_{\{i,j\} \in P} EC(\{i,j\}) \quad (1)$$

To ascertain comparability between proteins, CCS was normalized by dividing through the average EC score over all high-ranking pairs (Equation (2)):

$$CCS_{\text{norm}}(i) = \frac{CCS(i)}{\text{avg}(P)} \text{avg}(P) = \frac{1}{|P|} \sum_{\{i,j\} \in P} EC(\{i,j\}) \quad (2)$$

Next was a filter that identified EC clusters in a network, the nodes of which formed by all residues in a protein and the edges drawn between two nodes/residues with "high EC scores." The list with "high EC scores" was given by the original EC score ranking or by the original list filtered through solvent accessibility or by distance. The clustering coefficient for each node in the network measures how connected the neighborhood of a certain node is. Thereby, it identifies residues in locally dense clusters. Let  $G = (V, E)$  be a graph with  $V$  being the set of nodes and  $E$  being the set of edges. An edge  $e_{ij}$  connects nodes  $v_i$  and  $v_j$ . The neighborhood of a node  $v_i$  is given by  $N_i = \{v_j : e_{ij} \in E\}$ . The clustering coefficient  $C_i$  is then given by:

$$C_i = \frac{2 \cdot |\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{|N_i| \cdot (|N_i| - 1)} \quad (3)$$

### 2.3 | Effect predictions as filter

BLOSUM (block substitution matrix)<sup>25</sup> is a substitution matrix assessing how likely a mutation is observed in a protein family. BLOSUM

scores  $s_{ij}$  describe how the mutation of amino acid  $i$  into  $j$  compares to the expectation:  $s_{ij} = 0$  means "mutation as expected," while  $s_{ij} < 0$  implies "observed less often than expected," and  $s_{ij} > 0$  implies "observed more often than expected." The particular version BLOSUM62 can, therefore, be used to assess whether an observed mutation is likely to affect function,<sup>26</sup> for example, mutations with scores  $\geq 0$  can be considered as conservative and occurring in nature (essentially as neutral), those with scores  $< 0$  as non-conservative and less frequent due to purifying selection.<sup>27</sup>

Our filter combined BLOSUM62 with two methods predicting the effect of sequence variation upon molecular function, namely with SNAP2<sup>28</sup> and EVmutation.<sup>17</sup> For each residue in the protein, we determined the fraction of conservative mutations predicted as non-neutral by SNAP2 or EVmutation (taking ALL possible 19 non-native mutations into account). Residues with a high value for this filter were predicted to be binding. Both effect predictors were considered separately leading to two different scores. However, these scores are highly correlated (correlation coefficient = 0.65).

### 2.4 | Prediction based on weighted sum

Overall, eight different scores were used. Three reflected the CCS scores (Equation (2)) and three the clustering coefficients (Equation (3)). For both the list of pairs were compiled in three different ways: (1) take the full list of highly coupled pairs, (2) filter the list by solvent accessibility (keeping only pairs with at least one exposed residue), and (3) filter the list by spatial distance (keeping only those close). The scores derived from SNAP2 and EVmutation represented two additional scores. Each of those scores constituted a simple prediction of binding site. In order to combine all eight, CCS was normalized to range from 0 to 1 as all other scores. The combined score  $c_i$  for each residue  $i$  was the weighted sum over all scores  $c$  for this residue for the eight different predictions:

$$c_i = \sum_{j=1}^8 w_j \cdot c_{ij} \quad (4)$$

The weight for each score was simply given by its F1 score, that is, the harmonic mean between coverage and accuracy for that particular contribution. All residues for which  $c_i > c$  were predicted as binding where  $c$  is a predefined cut-off. To not overestimate the performance of the weighted sum the data set was split into five parts. The weights  $w_j$  and cut-off  $c$  were determined using four splits while the performance based on these weights and the threshold was estimated for the fifth split. This procedure was repeated so that every protein was tested exactly once.

### 2.5 | Application of machine learning

The eight scores to predict binding site residues can also be used as input to an artificial neural network (ANN). In addition to these eight, two other input units were used to reach a total of 10 input units, namely the solvent accessibility prediction and the conservation. Conservation scores were calculated based on the MSAs generated by EVcouplings. Using the Synthetic Minority Oversampling Technique (SMOTE)<sup>29</sup> compensated the imbalance between the class of binding

residues and non-binding residues. The ANN was implemented using the Python library `scikit-learn`<sup>30,31</sup> and had one hidden layer with 100 hidden units. Homologous sequences were included exclusively to train in order to increase the data set size (no homologs used for cross-training nor for testing). The ANN was trained and optimized using fivefold cross-validation splitting into three sets: training set, cross-training (or validation) set, and test set. Each protein in the redundancy-reduced data set was used for testing exactly once. The test sets were not used to make ANY decision: neither the best number of hidden units, nor early stopping, nor the choice of input. All of those parameters were optimized on the cross-training set. All sets contained roughly the same number of proteins, but not necessarily the same ratio of binding/non-binding residues.

## 2.6 | Homology-based inference

Homology-based inference of protein binding sites was performed based on the HVAL.<sup>16</sup> Assume a query protein Q and a protein with experimentally annotated binding sites E, then the binding sites for Q were homology-based inferred from E ("read off"), if:  $HVAL(E,Q) > \text{Threshold}$ . We tested different thresholds. The proteins were aligned locally using BLAST,<sup>32</sup> HVALs were calculated for these BLAST alignments and binding site annotations were inferred to the query protein from the BLAST hit with the highest HVAL.<sup>15</sup> Annotations were only inferred for the part of the query covered by the BLAST alignment.

## 2.7 | Comparison to other methods predicting binding sites in DNA-binding proteins

To compare the performance of our method to predict protein binding sites against other methods we chose three methods specialized on predicting DNA-binding residues. DP-Bind<sup>33</sup> uses the amino acid sequence, evolutionary profile and low-resolution structural information to apply a Support Vector Machine to predict DNA-binding residues. It was trained on 62 experimentally solved protein-DNA complexes. DRNAPred<sup>34</sup> considers a sliding window of 15 physicochemical and biochemical properties together with a hidden Markov model. This feature set serves as input to a logistic regression model to predict DNA-binding residues. They expand an existing benchmark dataset and train their model on 2827 DNA-binding proteins. someNA<sup>35</sup> uses a variety of features like amino acid composition, evolutionary profile, predicted secondary structure and solvent accessibility to train an ANN to predict DNA-binding residues. The model was trained on 144 proteins from PD1db.<sup>14</sup>

## 2.8 | Performance evaluation

The performance in predicting binding site residues was assessed through standard measures, namely *positive coverage* (or *sensitivity*:  $TP/TP + FN$ : residues correctly predicted as binding/all residues observed as binding), *positive accuracy* (or *precision*:  $TP/TP + FP$ : residues correctly predicted as binding/all residues predicted as binding), *negative coverage* (or *specificity*:  $TN/TN + FP$ ), and *negative accuracy* (or *negative predictive value*:  $TN/TN + FN$ ). To compile those numbers all residues NOT annotated as binding were considered as non-

binding (using the PDB annotation "SITE," ie, excluding protein-protein binding sites). The F1 score (F-measure) was compiled as the harmonic mean over positive coverage and positive accuracy. Each performance measure was calculated for each protein separately and the resulting distribution was used to calculate average performance values and SEs ( $\frac{\sigma}{\sqrt{n-1}}$  where  $\sigma$  refers to the SD and  $n$  to the number of proteins). If not stated otherwise, values for performance measurements are always given as averages along with  $\pm$ one SE.

The performance of the prediction using a weighted sum was compared with random predictions. The random background was generated by assigning the calculated ECs randomly to other pairs of residues in the same protein and by assigning the effect predictions from SNAP2 and *EVmutation* randomly to single residues. From this random assignment, CCS and clustering coefficients and the weighted sum were calculated as described above.

## 2.9 | Availability

The best machine learning-based prediction method is available as a Python project named *bindPredict* through GitHub: <https://github.com/Rostlab/bindPredict>. The repository contains the source code to run the trained model to predict binding sites in one or more query proteins. More detailed information on how to run the method is given in the repository.

# 3 | RESULTS AND DISCUSSION

## 3.1 | Weighted combination of single-feature prediction superior to random

The simplest method predicts binding sites based on a single feature. For instance, the cumulative coupling score (CSS Equation (2)) reflecting the evolutionary coupling of residue pairs, or the clustering coefficient of EC scores (Equation (3)), or information from the impact of sequence variation. While some of those simple methods did not perform very well, all outperformed random (Supporting Information Table S2): the F1 score reached by a single feature was highest for SNAP2 (F1 = ~24, Supporting Information Table S2), that is, comparing conservative mutations according to BLOSUM62 with effect predictions from SNAP2 (Methods). The second highest score was reached for another feature using an effect-prediction method, namely *EVmutation* (F1 = ~21, Supporting Information Table S2). However, if the worst method improved over random the least, this would also be *EVmutation* (factor of 2.3, Supporting Information Table S2). The reason was that *EVmutation* predicted more residues as binding, that is, had a higher chance to randomly hit on binding sites. In other words, the best single feature prediction would not use any EC or conservation score, instead, it would simply base the prediction on sites for which many conservative mutations according to BLOSUM62 are predicted to have an effect by SNAP2.

The combination of single methods through a simple weighted sum (Equation (4)) provided a better solution than any single feature (Supporting Information Table S2). Every residue above a certain threshold was classified as binding, all others as non-binding. The

threshold was determined through cross-validation. Comparing 10 thresholds between 0.1 and 0.9 with random always using 80% of the data to analyze performance, indicated highest improvements over random for a threshold of 0.3 (Supporting Information Table S3). Applying this threshold of 0.3 to the weighted sum (Equation (4)) resulted in a weighted-sum prediction method with  $F1 = 19.3\% \pm 0.7\%$  (sensitivity/positive coverage =  $35.2\% \pm 1.2\%$ ; precision/positive accuracy =  $16.5\% \pm 0.7\%$ ). The weighted sum clearly outperformed random predictions ( $F1 = 1.8\% \pm 0.7\%$ ) and served as baseline to measure further improvements against.

### 3.2 | Machine learning improved to F1 around 26%

The success of the relatively simple weighted sum over eight predictive features suggested the implementation of machine learning at least to optimize those weights. The challenge was the tiny data set of binding sites (redundancy reduced: 9483 binding residues). On top, the problem turned out to be very complex to learn (ie, the intrinsic complexity of the data was high): in order to learn, we needed neural networks with over 100 hidden units. Neural networks with fewer hidden units led to worse performance results (data not shown). Thus, even when only using the above eight features and three consecutive residues, the number of free parameters ( $100 * [3 * 8 + 2] = 2600$ ) was already slightly too high for the data set size (the cross-validation splits gave a training set size with about 5800 binding samples, that is, just twice the free parameters). Indeed, this rough back-of-an-envelope calculation was confirmed by the finding that using anything beyond information from a single residue was not supported by the data (data not shown).

Consequently, the input to the machine learning was carefully limited to: (i) the eight input features used for the weighted sum, (ii) the predicted per-residue solvent accessibility, and (iii) the conservation in the alignment generated by *EVcouplings*. Thus, the neural network (ANN) had 10 input and 100 hidden units (in one hidden layer), and two output units adding to 1200 free parameters. Although during training, the imbalance between binding (12%) and non-binding (88%) was corrected by over-sampling the binding residues (by a factor of eight), the non-binding still dominated. Therefore, the final decision was not performed by choosing "binding if output  $>0.5$ ," but rather by "binding if output  $>0.6$ ." Trying to slightly compensate the lack of data, homologs were added to the sequence-unique training set. Overlap between these training homologs and the test set were carefully avoided, that is, none of the added homologs was more sequence similar to any test protein than  $HVAL < 0$ . Performance was assessed by fivefold cross-validation using each protein exactly once for testing. The ANN reached F1 score of  $26.2\% \pm 0.8\%$  at a precision/positive accuracy of  $32.2\% \pm 1.1\%$  and a sensitivity/positive coverage of  $30.6\% \pm 1.1\%$ . The ANN achieved an area under the receiver operating characteristic (AUC) of 0.68 clearly outperforming a random approach (AUC = 0.45; Supporting Information Figure S3). The most significant improvement over random was reached at very low false positive rates (left-most region of ROC-curve, Supporting Information Figure S3). The performance differed highly between proteins (indicated by a very high median absolute deviation) with very high performance for a few proteins (Figure 1).

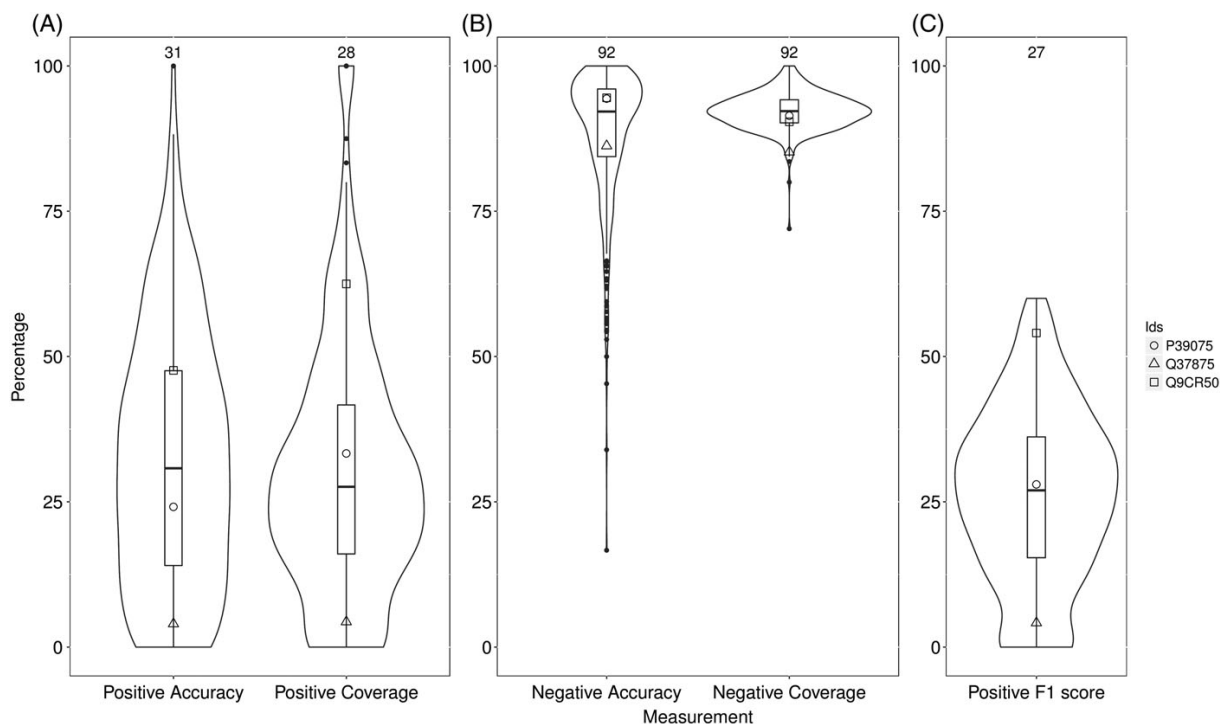
The best machine-learning solution presented in Figure 1 relied upon distance information inferred from known structures similar to the query protein. In other words, this method is applicable only if comparative modeling is applicable.<sup>36,37</sup> As this is not the case for all proteins, another method was tested that made no use of such information. For simplicity, this was tested by simply setting all values to 0, instead of by retraining without using that information. Without any information from known 3D structures, the method reached  $F1 = 25.6\% \pm 0.8\%$ . Thus, without distance information, performance dropped numerically (by 0.4 percentage points), but not significantly (by half a SE). Clearly then, the 3D information was not crucial.

Not using homologs for training by restricting the training to the original sequence-unique data set reduced performance to  $F1 = 25.5\% \pm 1.9\%$  (vs  $26.2\% \pm 0.8\%$  with homologs). Although the difference was not statistically significant, it held for all cross-validation-folds. Through the addition of homologs, the training data increased about threefold from roughly 6000 binding residues per cross-validation-fold to 18 000. The gain from using homologs might have been so limited because homologs might not contain enough new information, but only information similar to the data already presented. Including more non-redundant, unique data was impossible due to lack of data when focusing on enzymes and DNA-binding proteins with experimentally known structures. If the inclusion of more, but redundant data already improves substantially, the addition of more new data might be the most important key to major improvements.

Another limitation of the performance was likely noisy data. Wrong annotations of binding sites would make it more difficult for the neural network to learn the correct classification. Analyzing the size of the binding site suggested that certain annotations had severe issues (Supporting Information Figure S1). For instance, several enzymes had binding sites annotated for 60% of all their residues. Clearly not correct. In fact, all outliers, that is, all cases of proteins with most annotated binding residues were enzymes, not DNA-binding proteins. Correct annotations are likely to invert this finding: DNA-binding sites are, on average, larger than sites conveying enzymatic activity (this result is mostly due to many extreme outliers, Supporting Information Figure S1 violin plots under x-axis). Removing those cases from the testing would "beautified" the results reported (performance measures are higher for the set with fewer proteins); removing them from training would have reduced the valid points even further and thus been counterproductive. Therefore, for improving the method, not only more, but also better and more reliable data is needed. Surprisingly, the ANN learned from the experimental data to predict enzymes as having on average more binding sites than DNA-binding proteins (Supporting Information Figure S1, rightmost violin plots).

### 3.3 | Simple machine learning solution clearly outperformed weighted sum

The prediction of binding site residues using a weighted sum of *CCS*, clustering coefficients and disagreement between *BLOSUM* and *SNAP2* and *BLOSUM* and *EVmutation* served as one baseline prediction. Machine learning these eight plus solvent accessibility and conservation improved by roughly six percentage points from  $F1 = 19.3\%$



**FIGURE 1** Performance of neural network. The neural network using CCS, clustering coefficients and disagreement between BLOSUM and SNAP2 and BLOSUM and EVmutation as well as per-residue solvent accessibility and conservation as input features achieved a median accuracy of 31% with a median absolute deviation of (MAD) 25% corresponding to an average positive accuracy of  $32.2\% \pm 1.1\%$ , a median positive coverage of 28% with a MAD of 18% (average coverage of  $30.6\% \pm 1.1\%$ ) and a median positive F1 score of 27% with a MAD of 15% (average F1 score of  $26.2\% \pm 0.8\%$ ). Because the high imbalance between binding and non-binding residues, much higher values could be achieved for the negative accuracy and negative coverage than the (positive) accuracy and coverage. The performance varied highly between proteins and high values of the F1 score between 50% and 60% could be achieved for a few proteins. The performance of three examples is highlighted as colored dots. The structures of the chosen proteins and the binding site prediction are visualized in Figure 4

to  $F1 = 25.5\%$  (Figure 2). This increase was statistically significant by four standard errors (corresponding roughly to a  $P$ -value  $< 10^{-4}$ ).

The neural network outperformed the simple weighted sum for most proteins. However, despite the substantial difference in the averages, the weighted sum did better for many others (Supporting Information Figure S4: points below diagonal). Especially for the few examples for which the neural network failed completely to predict any binding site at the default threshold (Supporting Information Figure S4: points at  $F1 = 0$ ), the weighted sum at least predicted some binding site residues.

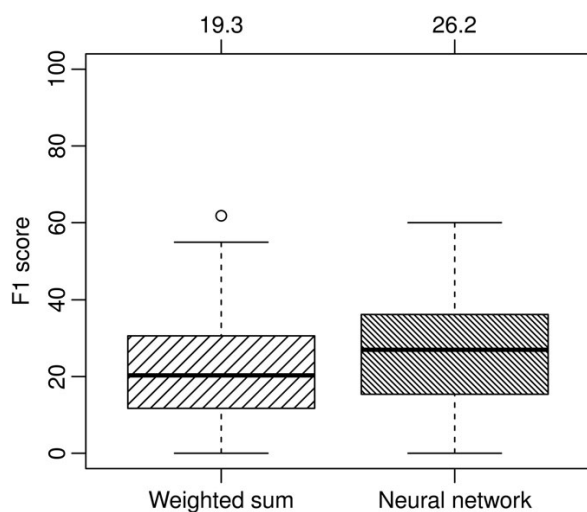
### 3.4 | Homology-based inference limited to a few proteins

Given a protein E with experimental binding site annotation that is sequence similar to a query protein Q, the binding sites for Q can be predicted by homology-based inference by simply transferring the annotation from E to Q, if, for example,  $HVAL(Q,E) < T$ , that is, the two are more similar than some threshold T. For instance, for  $T = 50$  ( $HVAL > 50$  implies over 70% pairwise sequence identity for long alignments), homology-based inference reached  $F1 = 46\% \pm 7\%$  (Supporting Information Figure S5) for the subset of proteins in our dataset for which this method could be applied. The machine learning

de novo prediction method reached  $F1 = 29\% \pm 3\%$  on the same subset of proteins (ie, better than  $F1 = 26.2\% \pm 0.8\%$  for the full data set). Using a higher HVAL thresholds dropped performance (eg,  $HVAL > 70$ :  $F1 = 36\% \pm 15\%$ ). Performance dropped for lower HVAL threshold. Nevertheless, at  $HVAL > 5$  (corresponding to  $>25\%$  PIDE for alignments over 250 residues), homology-based inference still slightly outperformed the de novo prediction ( $F1 = 29\% \pm 2\%$  for homology-based inference vs  $F1 = 27\% \pm 1\%$  for de novo, Supporting Information Figure S5).

By definition, homology-based inference is limited to finding sequence-similar proteins with reliable experimental annotations. Only for 27 of the 412 proteins in our data set (ie, 7%), we found a second protein with binding annotations at  $HVAL > 50$ . As always, however, the advantage of homology-based inference is that its power increases with growing databases. Another way to compare homology-based inference is by randomly (probability for binding = 0.12) predicting binding sites when no experimental information was available. This gave  $F1 = 12.9\% \pm 0.7\%$  for  $HVAL > 50$ , significantly below the de novo prediction for the full set ( $26.2\% \pm 0.8\%$ ).

These findings suggest combining homology-based inference and de novo prediction inferring binding annotations if a similar protein is available and predicting them using the ANN, otherwise. For  $HVAL >$

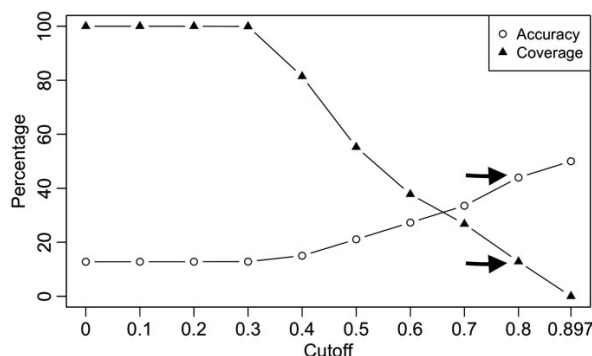


**FIGURE 2** Comparison of performance of weighted sum and of neural network. Using a weighted sum of CCS, clustering coefficients and disagreement between BLOSUM and SNAP2 and BLOSUM and EVmutation achieved an average F1 score of  $19.3 \pm 0.7\%$  while the neural network using these scores as well as per-residue solvent accessibility and conservation as input features achieved an average F1 score of  $26.2 \pm 0.8\%$ . Applying machine learning improved the predictive performance by seven percentage points and led mainly to an improvement of proteins with an average performance while the best performance achieved did not differ much between the application of the weighted sum and the neural network

10, the combined prediction reached the highest performance with  $F1 = 29\% \pm 1\%$ . For  $HVAL > 10$ , we have a homologous protein for 117 of 412 proteins. For these, homology-inference achieves  $F1 = 34\% \pm 3\%$  compared with  $F1 = 27.1\% \pm 0.8\%$  for de novo prediction. Assigning binding residues randomly when no experimental information was available leads to  $F1 = 19\% \pm 1\%$  for  $HVAL > 10$ . If homology-based inference outperformed de novo prediction down to  $HVAL > 5$ , why is the optimal combination not around  $HVAL = 5$ , then? Ultimately, because we are comparing apples and oranges: when citing the performance of homology inference vs de novo, we used different data set for both. When optimizing the threshold, the optimization implicitly used the entire set. Put differently, the above statement that homology inference works better down to  $HVAL > 5$  seemed to be based on overly optimistic data sets available for homology inference: for the same proteins, de novo prediction performed better than average.

### 3.5 | A few residues predicted reliably

Users of a binding site prediction method might be interested in considering only the most reliable predictions. More strongly predicted residues were also predicted more accurately (Figure 3), for example, 44% (arrow for Accuracy in Figure 3) of the 12% most strongly predicted binding residues (arrow for Coverage in Figure 3) were predicted correctly. Conversely, almost 20% of the residues were predicted at the lowest accuracy around 18% (saturation in accuracy curve below 0.4, Figure 3). The highest accuracy of 50% was reached



**FIGURE 3** Accuracy and coverage for different probability cut-offs. Using different cut-offs for the probability of the output unit “binding” to classify a prediction showed that the accuracy increased for higher probabilities while the coverage decreased. A small fraction of residues (12.8%) could be predicted at an accuracy of 44.0% for a probability cut-off of 0.8 while the highest precision of 50% could only be achieved for a very small fraction of residues (0.01%). There were no predictions of binding site residues for a probability  $\geq 0.9$

for only 1% of all residues (coverage of 0.01%, Figure 3). No residues were predicted with a probability  $\geq 0.9$  and only very few for  $\geq 0.8$ .

Accuracy did not increase much, for the subset of the most reliable predictions for each protein. However, zooming into the most reliable predictions, users could focus on some proteins for which very good predictions were obtained. Considering only the single most reliable prediction for each protein, this prediction was correct for 35% of all proteins (Table 1). Picking the five strongest predictions in each protein, at least one out of these five was correct for 70% of all proteins (Table 1); all five were correct for 5% of all proteins. Therefore, if only predictions with a high probability are picked, these could seed further (experimental and computational) investigations.

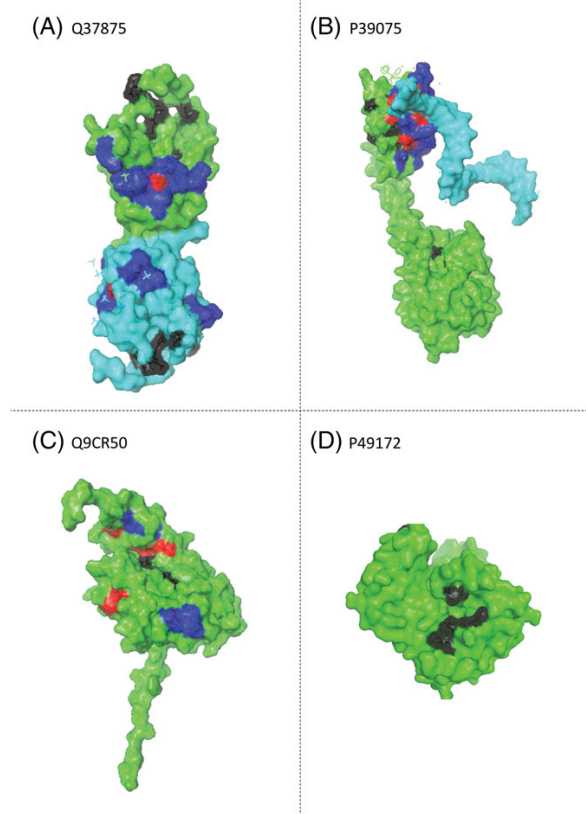
### 3.6 | Prediction method appeared to detect surface patches of binding

A closer inspection of three examples that were picked randomly from three different groups of proteins, namely those with low, average, or high F1 (Figure 4A-C; points marked in Figure 1) suggested that higher F1 scores originated from simultaneous reduction in residues mistakenly predicted as binding (FP) and as non-binding (FN). Examples with mediocre performance suggested that binding site residues missed

**TABLE 1** Choosing the most reliable prediction(s) in each protein

Number of most reliable predictions chosen per protein	Number of proteins with at least one correct prediction	Fraction of proteins with correct predictions
1	144	35%
2	213	52%
3	254	62%
4	277	67%
5	289	70%

When only considering the best prediction for every protein, the neural network made a correct prediction for 35% of the proteins. When picking the five best prediction, it could make at least one correct prediction for 70% of the proteins and for 5%, it was even completely right.

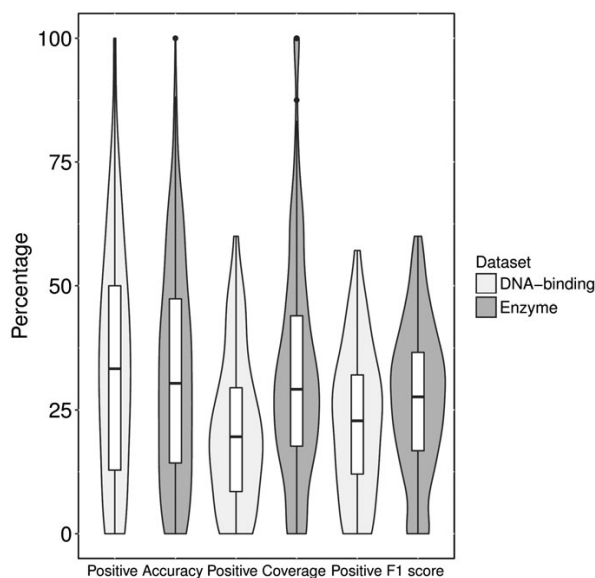


**FIGURE 4** Examples of predicted binding sites. Residues correctly predicted as binding are shown in red, residues wrongly predicted as non-binding in blue and residues wrongly predicted as binding in black. (A) The lysozyme of the bacteriophage P1 (UniProt identifier Q37875) is an example of an enzyme for which the prediction was rather bad with an F1 score of 4.2%. Especially binding residues around the center of the binding site were missed while certain regions were falsely predicted as binding. (B) The binding site of the multidrug-efflux transporter 1 regulator of *Bacillus subtilis* (UniProt identifier P39075) could be predicted with an F1 score of 29.2%. Again, the size of the binding site was underestimated leading to many missing residues around the center of the binding site while a second binding site in the lower part of the protein was predicted. (C) the prediction of the binding site of the RING finger and CHY zinc finger domain-containing protein 1 of the mouse (UniProt identifier Q9CR50) achieved an F1 score of 57.1%. The number of wrongly predicted residues was highly reduced compared with the other examples with a lower performance. (D) The 2-hydroxyruconate tautomerase from pseudomonas sp. (UniProt identifier P49172) is one of the proteins where none of the five most reliable predictions were correct. Visualizing these predictions on the structure (shown in black) made it obvious that they form a spatial cluster which can also be seen as a cavity that could act as a binding site. Therefore, it is possible that these five predictions are in fact not wrong, but the binding site is just not experimentally annotated

(FN) tended to spatially cluster around correct predictions. Also, residues wrongly predicted as binding tended to form small spatial clusters away from the binding site. Overall, the prediction method appeared to pick up spatial relations although only sequence information was used as input. For the examples chosen here, this even

remains true when not using any distance information to predict binding sites. The predictions only changed slightly (one new FP not predicted when using distance information) while the larger proportion remained the same.

In other words, the center of the binding site was often predicted correctly while its size was underestimated. This implied that residues in or at the periphery of the binding site were missed. The correctly predicted residues could help to identify the whole binding site by serving as a starting point for further analyses and experiments. Also, some residues were wrongly predicted as binding cluster. Especially if such clusters are big enough, it might be that instead of “false predictions,” these could be sites for which experimental results are missing. Put differently: the predictions might appear less accurate than they really were due to mistakes in annotations. For each protein, some residues were predicted at high reliability (Figure 3), and yet were labeled as incorrect. If only the 12% predictions with the highest reliability were considered, there was a prediction for every protein (Supporting Information Figure S6), but only 44% of those appeared correct (Figure 3). In fact, for 30% of the proteins, none of the five most reliable predictions were correct (Table 1). Randomly picking one of those proteins and visualizing the five most reliable predictions on the structure showed that these five predictions form a spatial cluster (Figure 4D). This cluster was also present when not using distance information to predict binding sites. In fact, these residues are located in a cavity that could function as a binding site. This suggested that these predictions were not wrong, but that a binding site annotation might be missing for this protein. We could not carry out such an analysis for all 30% of the above examples, nor can we



**FIGURE 5** Comparison of performance between DNA-binding proteins and enzymes. Comparing the performance for DNA-binding proteins and enzymes showed that binding sites of DNA-binding proteins are predicted at a lower accuracy while they are predicted at a slightly higher accuracy leading to an overall slightly lower F1 score for DNA-binding proteins. A more detailed analysis of the performance including negative accuracy and negative coverage is shown in Supporting Information Figure S7

ascertain that this protein does have an unknown binding site. However, given that we tested one single example and found something very reasonable, the suspicion that we substantially underestimated performance was clearly supported by this example. Overall, therefore, binding site residue predictions and especially very reliable predictions in proteins with known annotations might help to identify previously missed binding sites and might offer new directions for experiments to refine binding site annotations.

### 3.7 | Underestimating size of DNA-binding sites

Performance for DNA-binding proteins was slightly worse than for enzymes (Figure 5; Supporting Information Figure S7). Since DNA-binding sites tend to be larger than substrate binding sites of enzymes (Supporting Information Figure S1C), we had assumed the opposite, namely a higher performance for DNA-binding proteins than for enzymes. One problem for the prediction of DNA-binding might have been that 87% of the proteins in the data set were enzymes (357 enzymes vs 55 DNA-binding). The neural network largely learnt the average size (Supporting Information Figure S1: compare violin plots along y- and x-axes). Given the dominance of enzymes, the size prediction was also dominated by the smaller binding sites of those. Thus, DNA-binding sites tended to be more often under-predicted than those of enzymes (Supporting Information Figure S1: more black disks above than below dashed line). This under-prediction reduced the performance for DNA-binding proteins. DNA-binding proteins were also the most extreme cases for which the weighted sum outperformed the neural network (Supporting Information Figure S4). This added to the view that the abundance of enzymes in the training distracted the neural networks from learning DNA-binding sites.

### 3.8 | Active site residues predicted at higher reliability

Additional to the binding sites annotated in the PDB and in the PDIdb, Swiss-Prot provides explicit annotations for “active sites,” and alternative annotations for “binding sites.” However, only 64 of the 412 proteins (16%) had active/binding sites with an experimental evidence code (ECO:0000269) in Swiss-Prot/UniProt. On average, about 1% of the residues in a protein were annotated as binding by Swiss-Prot/UniProt compared with 12% for PDB and PDIdb. Obviously, Swiss-Prot/UniProt annotations were based on a narrower definition of binding, while PDB identified binding sites as a larger spatial region around the ligand. The fact that only 16% of the proteins had any binding/active site annotation in UniProt demonstrated that most annotations (84%) remained missing in UniProt even by their own definition. Annotations from the PDB and PDIdb, thus, rendered a more comprehensive data set for method development and assessment. Nevertheless, if de novo prediction works, it should also work for sites annotated in Swiss-Prot, in particular, since those—where available—might point to the most important binding residues in a protein.

The direct comparison between annotations that cover 1% of all residues and those covering 12% would be non-sense because the random odds of hitting 1 in 100 differ more than tenfold from those of hitting 12 in 100. Thus, the two sets of annotations needed to be

compared indirectly. For instance, binding sites annotated in Swiss-Prot were predicted at higher reliability than annotations from the PDB and PDIdb (Supporting Information Figure S8). In fact, when considering the 10% most reliable predictions for every protein, 9% of these predictions were Swiss-Prot binding sites as compared with 1% in the whole data set. Thus, Swiss-Prot binding site residues were overrepresented by a factor of 8 in the 10% most reliable predictions. PDB/PDIdb binding site residues were only over-represented by a factor of 2.5 (12% in the whole set vs 30% in the 10% most reliable predictions). The de novo machine learning method might have been trained on too inclusive (too large) binding sites from the PDB/PDIdb. However, the resulting method predicted the most important subset of these residues (narrower Swiss-Prot) even more reliably than those it had been trained on.

### 3.9 | Specialized DNA-binding predictions seemed better

Although, the machine learning had not been able to perform well for DNA-binding proteins, we still compared it to methods specialized on this task, namely on the three methods *DP-Bind*,<sup>33</sup> *DRNApred*,<sup>34</sup> and *someNA*.<sup>35</sup> Performance comparisons between these methods and our machine learning had limited value because all three methods most likely used many of the proteins in our data set (in fact, *someNA* used all DNA-binding proteins from our data set) for training their method. As we could not access their cross-validation results, we had to compare their training with our testing performance which will clearly over-estimate the performance for the specialists. Specialist could learn specifics about protein-DNA binding that generalist methods such as the one presented here will hardly be able to zoom into.

The 55 DNA-binding proteins in our dataset suggested that *DP-Bind* and *someNA* perform similar ( $F1 = 46\%$ ), while *DRNApred* appeared inferior ( $F1 = 17\%$ , Table 2). Again, as *DRNApred* was shown to be superior to state-of-the art prediction methods,<sup>34</sup> it might be that the other were optimized more on proteins identical or similar to the 55 used here, that is, that we inadequately compared training and testing when comparing *DP-Bind/someNA* on the one side (training) and *DRNApred* (testing) on the other. Also, the low performance of *DRNApred* is partially caused by a low coverage which suggests that *DRNApred* was trained on smaller binding sites leading to an underestimation of the size of the binding site. The generalist method introduced here fell between the two extremes ( $F1 = 22\% \pm 2\%$ ). Although some of the difference to the much better *DP-Bind* and *someNA* might root in the data set overlap issue (“used for training”), the difference appeared to be so substantial (more than factor of two in  $F1$ ) that a major reason might simply be the specialists to clearly outperform the generalist. More surprising was that although *DRNApred* most likely

**TABLE 2** DNA-binding specialist vs generalist

Method	DP-bind	someNA	DRNApred	Generalist-here
<i>F1 score</i>	46% ± 2%	46% ± 3%	17% ± 3%	22% ± 2%

The three specialists in predicting protein-DNA binding (*DP-bind*, *someNA*, and *DRNApred*) were applied to the 55 DNA-binding proteins in our dataset. The numbers reflect the cross-validated view only for the method introduced here (generalist) because the specialists all used either the same or similar proteins for training.

used some of those 55 proteins for training and although it specialized on DNA and RNA prediction, our generalist was competitive (improvement by 2.5 standard errors not impressive but significant). This showed that the generalist picked up crucial features about "binding" although not learning any features specific to nucleic acid binding.

## 4 | CONCLUSIONS

Our hypothesis was that we could use evolutionary couplings to directly predict protein ligand binding sites directly from sequence. In the most optimistic view, we proved that the signal carried by evolutionary couplings clearly outperformed random predictions and that our rigid decision to consider existing annotations as complete (not annotated as binding implied non-binding) was obviously too rigorous due to the incompleteness of and the noise in existing binding site annotations. The binding of proteins to DNA and to substrates upon which enzymes act, served as proxy for protein ligand binding.

Combining eight different scores derived from evolutionary couplings and sequence variation into a weighted sum and using this sum to predict binding site residues in enzymes and DNA-binding proteins resulted in a method outperforming a random approach by roughly 17 percentage points ( $F1 = 19.3 \pm 0.7$ , Figure 2). To put this score into perspective: picking a protein for which 10 residues are predicted as binding, 2 of 10 will be correct and this prediction will cover roughly half of the entire binding site. This combined prediction served as a baseline to assess whether machine learning can improve performance. Training a neural network using the same eight scores plus predicted solvent accessibility and conservation statistically significantly outperformed the baseline ( $F1 = 26.2\% \pm 0.8\%$ , Figures 1 and 2).

The strength of the neural network prediction reflects the reliability of the prediction (Figure 3). Choosing the five most reliable predictions for each protein gives at least one correctly predicted binding residue in 70% of all proteins (Table 1). Visualizing some examples suggested that especially very reliable, but wrong predictions of binding sites tended to form spatial clusters (Figure 4) which might imply that the predictions were more correct than the annotations. Therefore, the neural network can help in refining annotations of known binding sites or in finding experimental evidence for new binding sites. When homologs with experimentally known annotations of binding sites are available, reading those off is a better strategy than using the de novo prediction method (Supporting Information Figure S5). The simple combination "if an experimental annotation to a protein at HVAL > 10 is available: use homology inference, else: use de novo" gave the best performance boost, reaching  $F1 = 29\% \pm 1\%$ .

For our data, DNA-binding was, unexpectedly, predicted less well than active sites (Figure 5; Supporting Information Figure S1), and good DNA-binding predicting specialists seemed to clearly outperform the generalist approach introduced here (Table 2). Ultimately, the major limitation in performance appeared to originate from the insufficient amount of data available for our approach. This data shortage might also have been in the way of clearly answering our hypothesis:

evolutionary couplings enable the prediction of generic small binding sites (proxied by enzymes) and large binding sites (proxied by DNA-binding). At the same time, the evolutionary couplings did not provide the features most important for the success of the prediction method (Supporting Information Table S2). In particular, features generated by methods that predict the effect of sequence variation upon molecular function (eg, SNAP2 and EVmutation, Supporting Information Table S2) contributed more to the prediction success.

## ACKNOWLEDGMENTS

Thanks to Tim Karl (TUM) for invaluable help with hardware and software; to Inga Weise (TUM) without who we could not accomplish what we do; to Chris Sander and Deborah Marks for the development of EVcouplings and their ideas helping this project. Thanks to Cristina Marino Buslje (CONICET Buenos Aires) for her optimistic view that got us going onto this path; thanks to Christine Orengo (UCL London) for repeatedly encouraging us on this journey. Thanks also to both anonymous reviewers who helped importantly to improve this work. Last not least, thanks to Helen Bermann (PDB Rutgers), Ioannis Xenarios (Swiss-Prot, SIB, Geneva), Francisco Melo (PD1db, Santiago) and their crews for maintaining excellent databases and to all experimentalists who enabled this analysis by making their data publicly available.

## ORCID

Maria Schelling  <http://orcid.org/0000-0001-8533-8163>

## REFERENCES

- Whisstock JC, Lesk AM. Prediction of protein function from protein sequence and structure. *Q Rev Biophys*. 2003;36(3):307-340.
- Radivojac P, Clark WT, Oron TR, et al. A large-scale evaluation of computational protein function prediction. *Nat Methods*. 2013;10(3):221-227.
- Alberts B, Johnson A, Lewis JH, Morgan D. *Molecular Biology of the Cell*. New York, NY: Garland Science; 2015.
- Harms MJ, Thornton JW. Evolutionary biochemistry: revealing the historical and physical causes of protein properties. *Nat Rev Genet*. 2013;14:559-571.
- Lehner B. Genotype to phenotype: lessons from model organisms for human genetics. *Nat Rev Genet*. 2013;14(3):168-178.
- Marks DS, Colwell LJ, Sheridan R, et al. Protein 3D structure computed from evolutionary sequence variation. *PLoS One*. 2011;6(12):e28766.
- Marks DS, Hopf TA, Sander C. Protein structure prediction from sequence variation. *Nat Biotechnol*. 2012;30(11):1072-1080.
- Hopf TA, Colwell LJ, Sheridan R, Rost B, Sander C, Marks DS. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*. 2012;149(7):1607-1621.
- Teppa E, Wilkins AD, Nielsen M, Buslje CM. Disentangling evolutionary signals: conservation, specificity determining positions and coevolution. Implication for catalytic residue prediction. *BMC Bioinformatics*. 2012;13:235.
- Aguilar D, Oliva B, Marino Buslje C. Mapping the mutual information network of enzymatic families in the protein structure to unveil functional features. *PLoS One*. 2012;7(7):e41430.
- Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*. 2000;28(1):45-48.
- Webb EC. Enzyme nomenclature 1992: Recommendations of the Nomenclature committee of the International Union of Biochemistry and Molecular Biology; 1992.



13. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res.* 2000;28(1):235-242.
14. Norambuena T, Melo F. The protein-DNA interface database. *BMC Bioinform.* 2012;11(1):1-12.
15. Mika S, Rost B. UniqueProt: creating representative protein sequence sets. *Nucleic Acids Res.* 2003;31(13):3789-3791.
16. Rost B. Twilight zone of protein sequence alignments. *Protein Eng.* 1999;12(2):85-94.
17. Hopf TA, Ingraham JB, Poelwijk FJ, et al. Mutation effects predicted from sequence co-variation. *Nat Biotechnol.* 2017;35(2):128-135.
18. Finn RD, Clements J, Arndt W, et al. HMMER web server: 2015 update. *Nucleic Acids Res.* 2015;43(W1):W30-W38.
19. Kajan L, Hopf TA, Kalas M, Marks DS, Rost B. FreeContact: fast and free software for protein contact prediction from residue co-evolution. *BMC Bioinform.* 2014;15:85.
20. Ekeberg M, Lovkvist C, Lan Y, Weigt M, Aurell E. Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. *Phys Rev E Stat Nonlin Soft Matter Phys.* 2013;87(1):012707.
21. Jones DT, Buchan DW, Cozzetto D, Pontil M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics.* 2012;28(2):184-190.
22. Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. *Proteins.* 1994;20(3):216-226.
23. Rost B. PHD: predicting one-dimensional protein structure by profile based neural networks. *Methods Enzymol.* 1996;266:525-539.
24. Rost B, Sander C. Conservation and prediction of solvent accessibility in protein families. *Proteins.* 1994;20(3):216-226.
25. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A.* 1992;89(22):10915-10919.
26. Ng PC, Henikoff S. SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Res.* 2003;31(13):3812-3814.
27. Cargill M, Altshuler D, Ireland J, et al. Characterization of single-nucleotide polymorphisms in coding regions of human genes. *Nat Genet.* 1999;22(3):231-238.
28. Hecht M, Bromberg Y, Rost B. Better prediction of functional effects for sequence variants. *BMC Genom.* 2015;16(suppl 8):S1.
29. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. *J Artif Int Res.* 2002;16(1):321-357.
30. Buitinck L, Louppe G, Blondel M, et al. API desing for machine learning software: experiences from the scikit-learn project. ECML PKDD Workshop: Languages for Data Mining and Machine Learning; 2013:108-122.
31. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in python. *J Mach Learn Res.* 2011;12:2825-2830.
32. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *J Mol Biol.* 1990;215(3):403-410.
33. Kuznetsov IB, Gou Z, Li R, Hwang S. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins.* 2006;64(1):19-27.
34. Yan J, Kurgan L. DRNAPred, fast sequence-based method that accurately predicts and discriminates DNA- and RNA-binding residues. *Nucleic Acids Res.* 2017;45(10):e84.
35. Hönigschmid P. *Improvement of DNA and RNA Protein Binding Prediction.* Munich, Germany: Technical University of Munich; 2012.
36. Waterhouse A, Bertoni M, Bienert S, et al. SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Res.* 2018;31(13):3381-3385.
37. Webb B, Sali A. Protein structure modeling with MODELLER. *Methods Mol Biol.* 2017;1654:39-54.

#### SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

**How to cite this article:** Schelling M, Hopf TA, Rost B. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins.* 2018;1-11. <https://doi.org/10.1002/prot.25585>

**4.1.3. Supplementary Material: Schelling *et al.*, Proteins: Structure, Function, and Bioinformatics (2018)**

**Supporting online material**

**for:**

**Evolutionary couplings and sequence variation effect predict protein binding sites**

**Maria Schelling, Thomas Hopf & Burkhard Rost**

**Table of Contents for Supporting Online Material**

1. Comparison of actual size and predicted size of binding sites
2. Sequence diversity and length of MSA generated by EVcouplings
3. ROC and AUC for ANN
4. Comparison of performance of weighted sum and neural network
5. Performance of homology-based inference
6. Analysis of number of proteins with a prediction for different probabilities
7. Detailed performance comparison for DNA-binding proteins and enzymes
8. Comparison of reliability of prediction of binding sites annotated in UniProt and annotated in PDB/PDIdb
9. Performance comparison of mfDCA and plmDCA for different parameters
10. Performance comparison with random approach
11. Performance comparison for different cut-offs for weighted sum

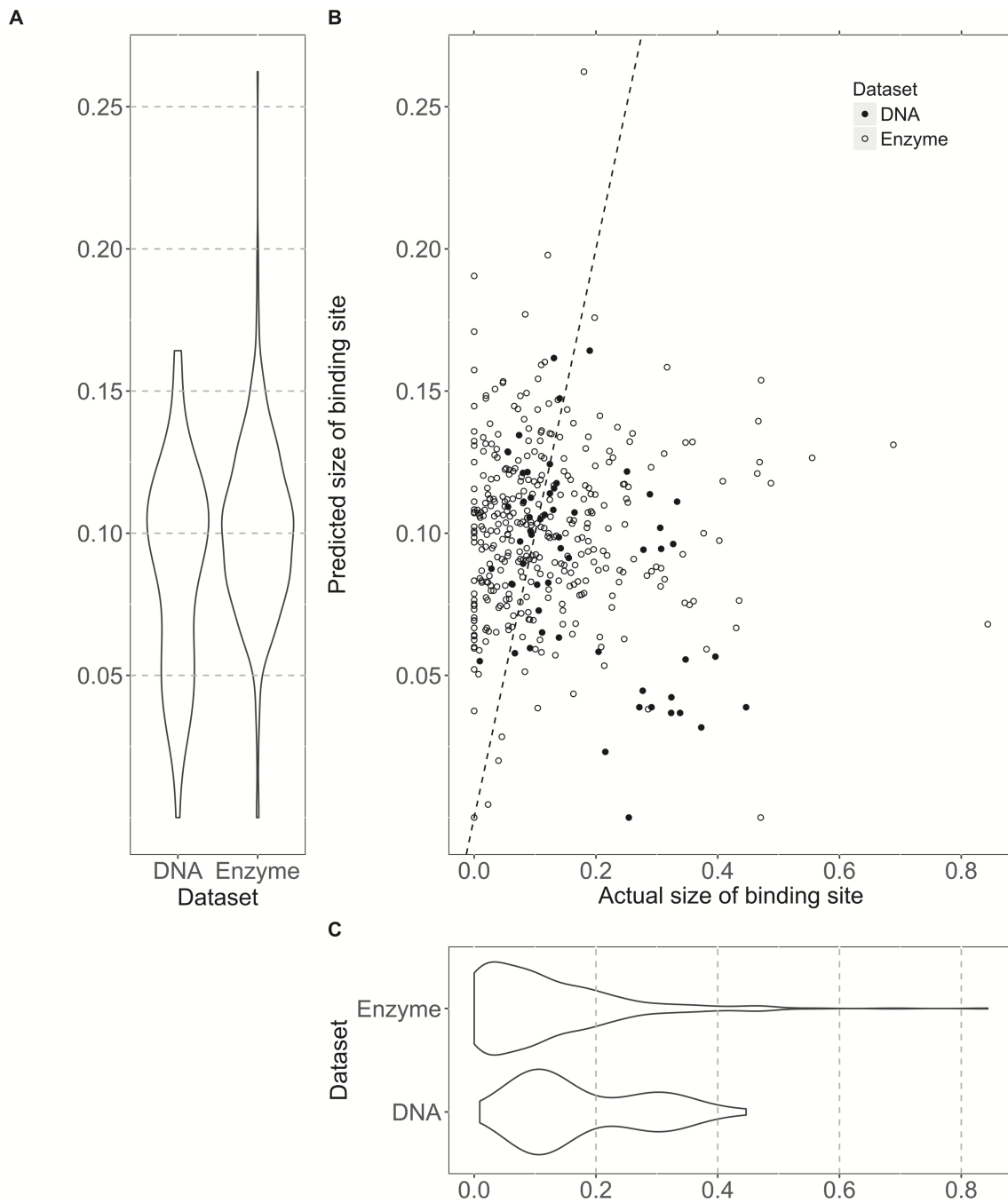
---

### Short description of Supporting Online Material

1. Figure showing the size of the actual binding site versus the size of the predicted binding site for enzymes and DNA-binding proteins and the distribution of number of actual and predicted binding site residues for enzymes and DNA-binding proteins
2. Figure showing the sequence diversity and length of MSA generated by EVcouplings
3. Figure showing the ROC and AUC for the ANN to predict binding site residues
4. Figure comparing the performance of the weighted sum and the neural network (given by the F1 score) for DNA-binding proteins and enzymes
5. Figure showing the performance of homology-based inference for different HVALs
6. Figure showing the number of proteins for with at least one prediction dependent from the chosen probability as cut-off
7. Figure showing the performance for DNA-binding proteins and enzymes by showing positive accuracy, coverage and F1 score and negative accuracy and coverage
8. Figure showing the distribution of reliability indices for the prediction of binding site residues annotated in UniProt and of those annotated in PDB/PDIdb
9. Table comparing performance of mfDCA and plmDCA for different parameters
10. Table showing the actual and random performance for the eight single scores
11. Table showing the performance for different cut-offs for the weighted sum combine all eight scores

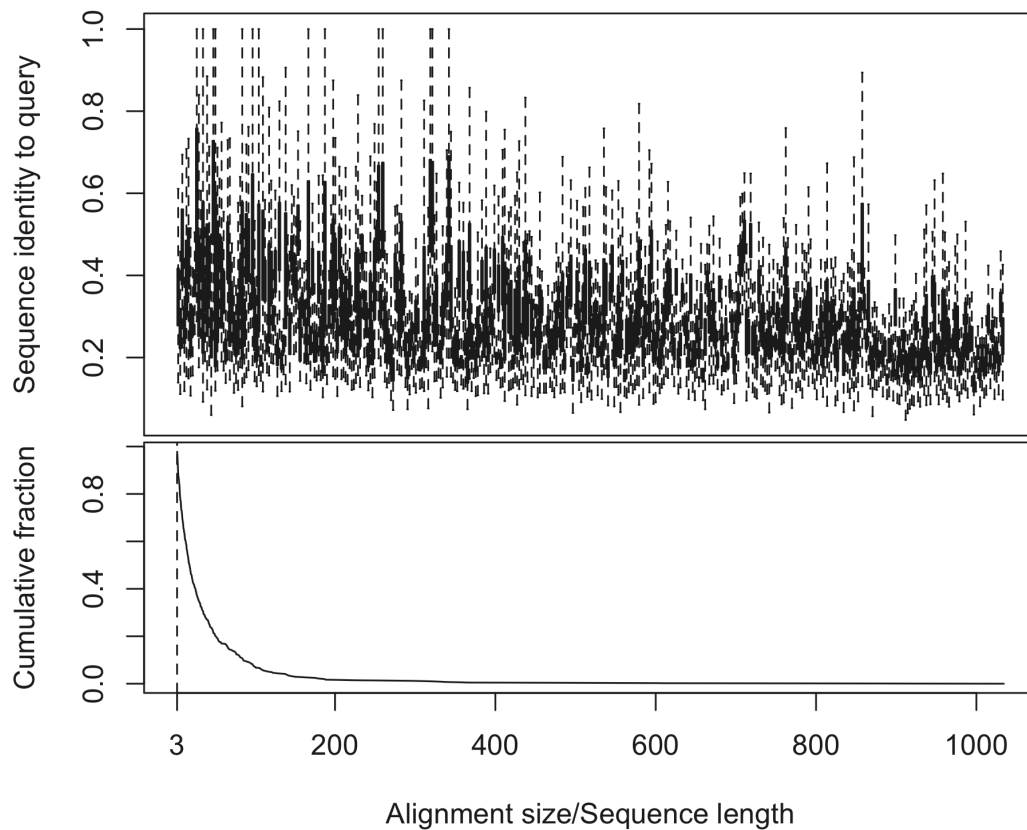
## Material

Fig. S1:



**Comparison of size of the actual binding site with size of the predicted binding site.**

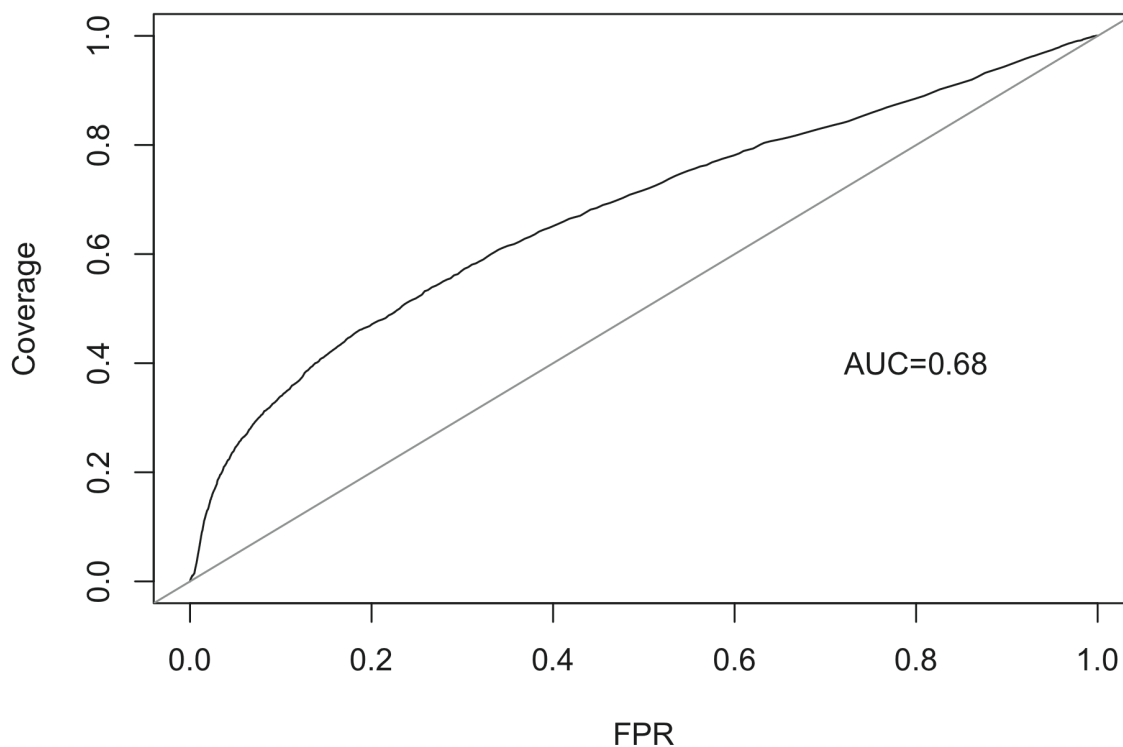
DNA-binding proteins tend to have a larger binding site than enzymes. However, the neural network learnt the average size of binding sites from the given data. The data is highly dominated by enzymes (357 enzymes vs 55 DNA-binding). Therefore, the neural network tended to predict binding sites with a similar size as expected for enzymes leading to an underestimation of the actual size of the binding site for DNA-binding proteins. Proteins represented by dots that are below the dotted line are proteins for which the size of the actual binding site were larger than of the predicted binding site.

**Fig. S2:**

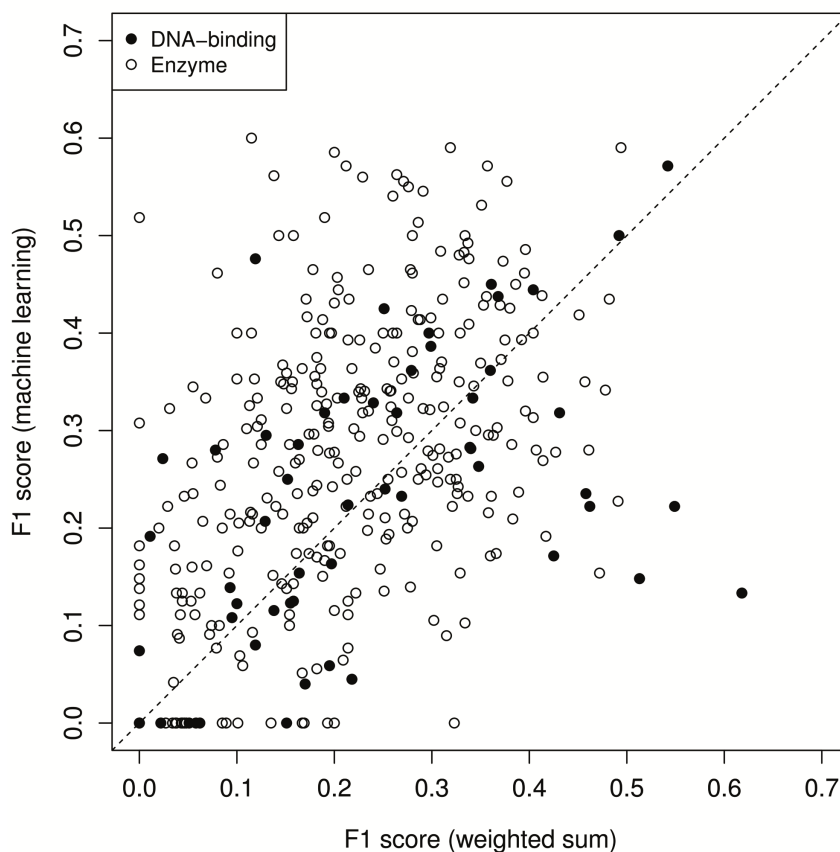
**Sequence diversity and length of MSAs generated by EVcouplings.** To ensure high quality results, the MSA should contain at least  $3L$  sequences with  $L$  being the number of positions in the query sequence used for the probability model and the sequence diversity within the alignment should be high enough, so sequences in the alignment should have a low sequence identity to the query. The upper part of the figure shows the sequence diversity for every MSA, only outliers are not shown for better visualization. It becomes clear that most sequences in the alignments have a sequence identity between 0.2 and 0.6 to the query sequence with an average sequence identity for all alignments of 0.26. Alignments in the

upper part of the figure are sorted by their size (number of sequences) as shown in the lower part. The size is given with respect to  $L$  and alignments with a size  $< 3L$  were excluded from the data set. The curve is a reverse cumulative curve. So, for every length  $x$ , the corresponding point on the curve gives the fraction of alignments with a size  $\geq x$ .



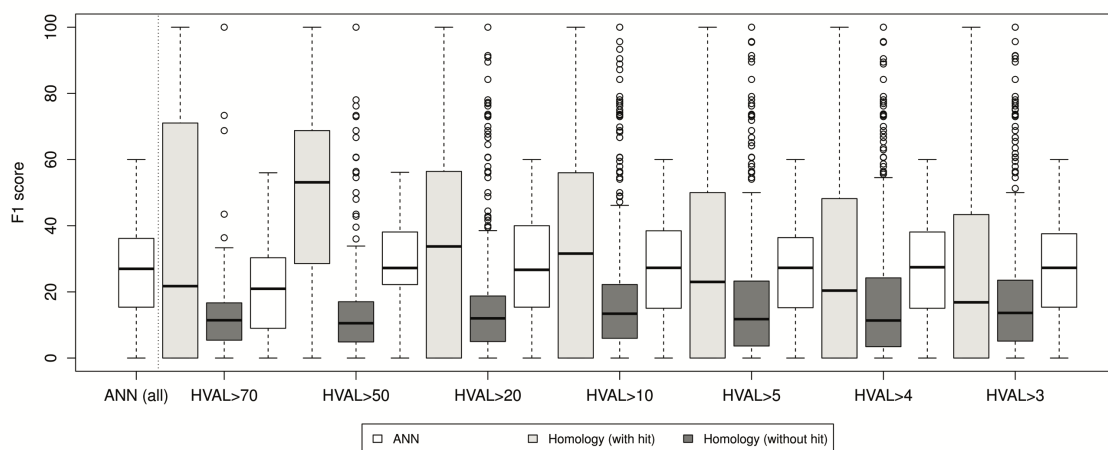
**Fig. S3:**

**ROC curve and AUC for ANN predicting binding site residues.** The receiver operation characteristic (ROC) sets the coverage and false-positive rate (FPR, number of residues falsely predicted as binding divided by the number of non-binding residues) into relation. The straight line represents the ROC for a random prediction. The area under the curve (AUC) is 0.68 showing that the ANN outperforms random (AUC=0.45)

**Fig. S4:**

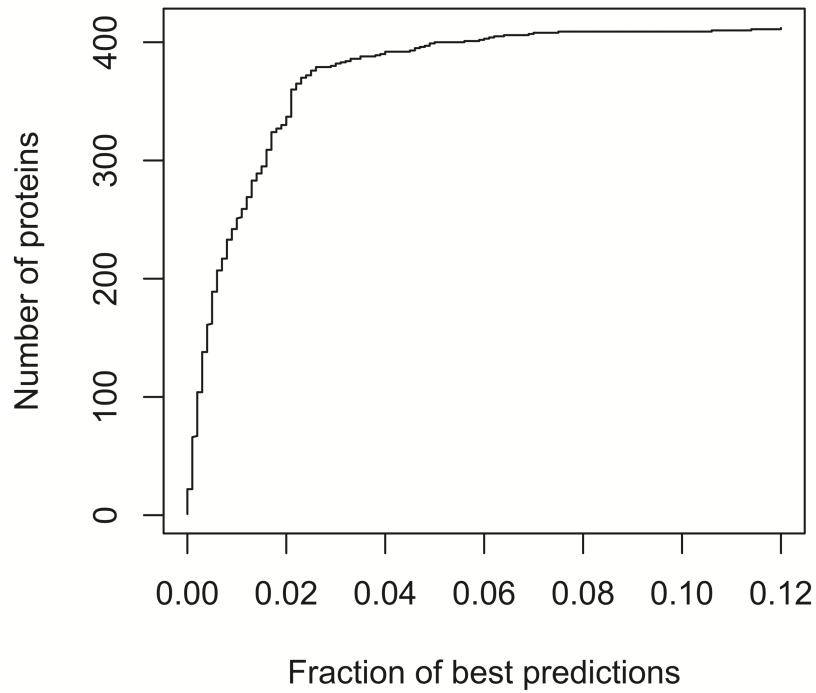
**Comparison of performance of weighted sum and neural network.** Comparing the F1 score of the neural network and the weighted sum on a per-protein basis showed that the neural network outperformed the weighted sum for most cases while there are some proteins for which the weighted sum achieved higher F1 scores. Every point below the dotted line represents a protein for which the weighted sum works better than the neural network. It became apparent that most of the extreme cases for which the weighted sum worked much better than the neural network were DNA-binding proteins suggesting that the neural network

had more problems in correctly predicting binding sites for DNA-binding proteins than the weighted sum.

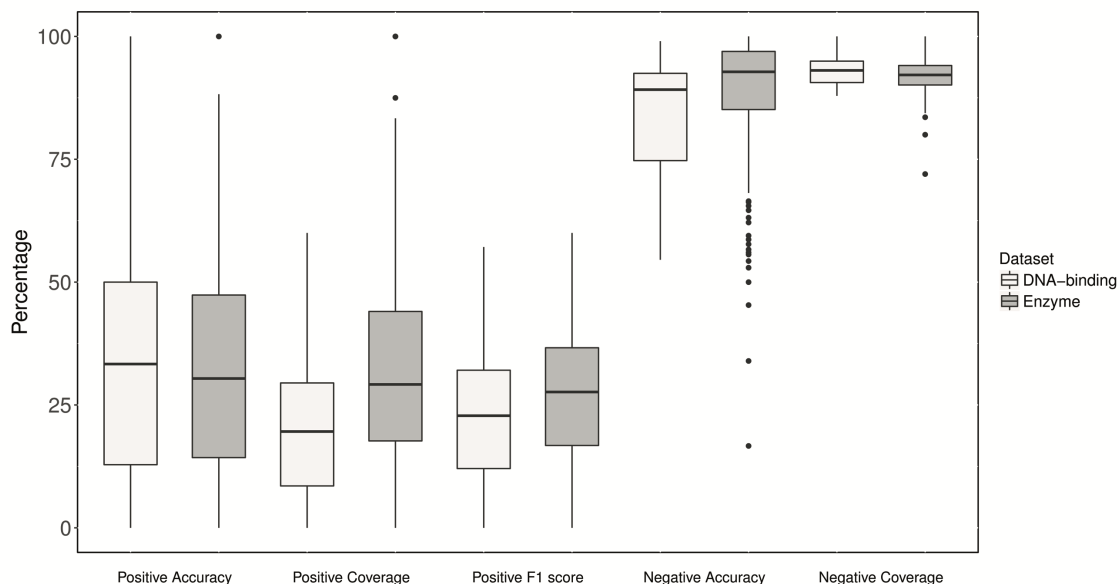
**Fig. S5:**

### Performance of homology-based inference limited by number of available homologs.

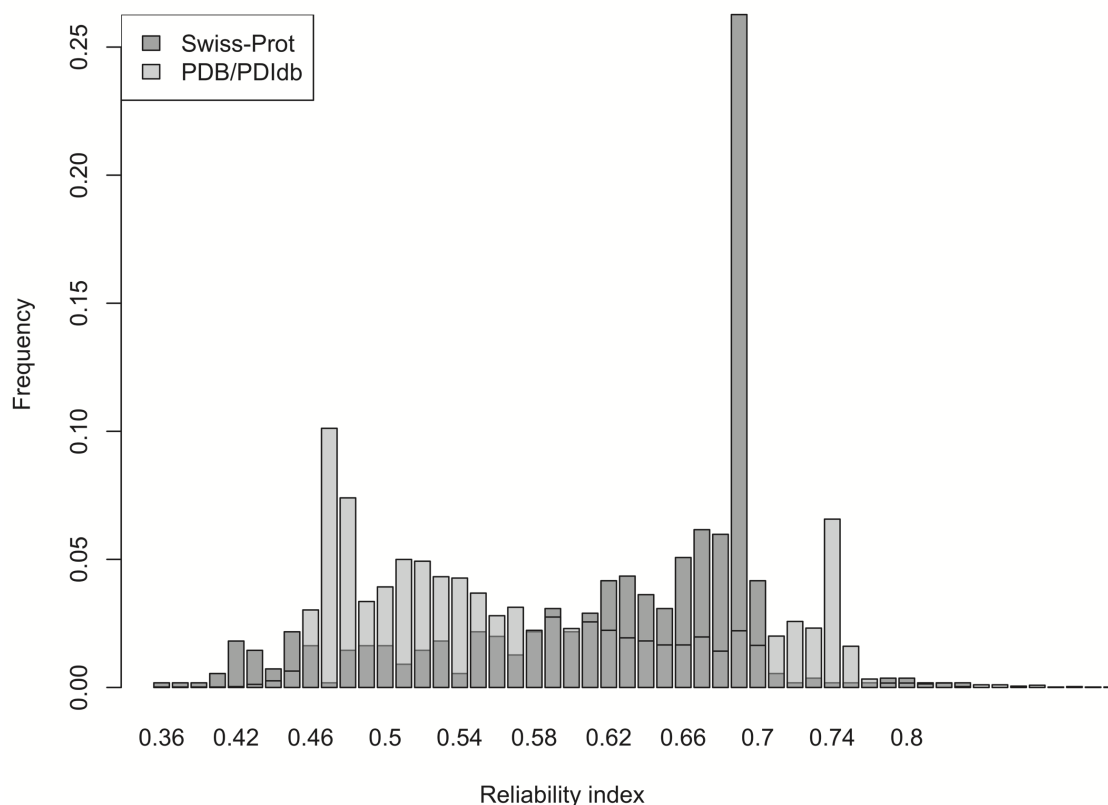
When predicting protein binding sites by transferring a known binding site annotation from a similar protein to the query protein (homology-based inference), this inference works best when two proteins are considered as similar if they have a pairwise HVAL > 50. For this threshold, homology-based inference on the set of proteins with a homolog achieves an average F1 score of  $46 \pm 7\%$  while machine learning can only achieve an average F1 score of  $29 \pm 3\%$  on the same set. However, the performance of homology-based inference is limited by the fact that there does not exist a homolog for every protein. Assigning binding residues randomly to the remaining proteins leads to an overall performance of  $12.9 \pm 0.7\%$  which is significantly worse than the performance of the de novo prediction ( $26.2 \pm 0.8\%$ ). Applying homology-based inference if a homolog exists and de novo prediction otherwise, improves the performance to  $F1 = 29 \pm 1\%$  when choosing  $HVAL > 10$  to define homology.

**Fig. S6:**

**Number of proteins with a prediction for most reliable predictions.** When only the 12% best predictions with the highest probability were considered, there was a prediction for every protein. This showed that reliable predictions were made for every protein although it is not clear whether these predictions are right or wrong.

**Fig. S7****Detailed comparison of performance between DNA-binding proteins and enzymes.**

Comparing the performance for DNA-binding proteins and enzymes showed that binding sites of DNA-binding proteins are predicted at a lower coverage while they are predicted at roughly the same accuracy leading to an overall slightly lower F1 score for DNA-binding proteins. Because of the high imbalance between binding and non-binding residues, negative accuracy and negative coverage achieve very high values for both enzymes and DNA-binding proteins. Most residues are correctly predicted as non-binding.

**Fig. S8**

**Comparison of prediction of binding site residues annotated in UniProt and of binding site residues annotated in PDB/PDIdb.** Comparing the reliability of prediction for binding site residues annotated in UniProt with binding site residues annotated in PDB/PDIdb shows that binding site residues from UniProt are predicted at higher reliability than binding site residues from PDB/PDIdb. However, since UniProt only has experimentally validated annotations for 64 of our 412 proteins, the UniProt annotations are far from complete (even enzyme and DNA-binding protein should have an annotated binding site). Therefore, UniProt annotations are not sufficient for a *de novo* prediction of protein binding sites.

**Table S1: Performance comparison for different parameters for plmDCA and mfDCA\***

	<b>plmDCA</b> <b>0.9</b>	<b>plmDCA</b> <b>0.999</b>	<b>plmDCA</b> <b>0.99999</b>	<b>mfDCA</b> <b>0.5L</b>	<b>mfDCA</b> <b>L</b>	<b>mfDCA</b> <b>2L</b>	<b>mfDCA</b> <b>3L</b>
<b>F1 score (CCS)</b>	6.0±0.5	3.8±0.4	3.2±0.4	4.8±0.4	9.4±0.5	9.3±0.5	10.2±0.5
<b>F1 score (CC)</b>	13.7±0.8	8.4±0.4	7.2±0.4	11.0±0.5	15.5±0.6	17.6±0.7	16.1±0.6

\* plmDCA provides a probability for each residue pair to reflect a real evolutionary coupling while mfDCA only provides a ranking given by the order of the scores. To determine how many scores should be considered as real evolutionary couplings, we tested different numbers of pairs. For plmDCA, best results are obtained both for CCS and CC when using a probability cutoff of 0.9 and considering every residue pair with a probability equal to 0.9 or higher as actually evolutionary coupled. For mfDCA, using the top 0.5L pairs leads to the worst results both for CCS and CC. Using the top 2L leads to the best performance for CC while using the top 3L leads to the best performance for CCS. However also the random performance increases for the top 3L, therefore we decided to use only the top 2L pairs for binding site prediction. For the best parameter settings (0.9 and 2L), mfDCA performs better than plmDCA when trying to predict binding sites. Therefore, mfDCA and a cutoff of 2L is used in our analysis.



**Table S2: Simple features outperform random in predicting binding sites. \***

<i>Single feature</i>	<i>F1 score</i>	<i>F1 score (random)</i>	<i>Improvement over random</i>
<b>CCS</b>	9.3±0.5	1.3±0.5	7.2
<b>CCS Dist</b>	12.5±0.6	3.4±0.6	3.7
<b>CCS Solv</b>	10.4±0.5	3.4±0.5	3.1
<b>CC</b>	17.6±0.7	2.6±0.7	6.8
<b>CC Dist</b>	16.4±0.7	2.1±0.7	7.8
<b>CC Solv</b>	16.8±0.7	1.2±0.7	14.0
<b>SNAP2</b>	24.2±0.8	5.6±0.7	4.3
<b>EVmutation</b>	21.3±0.8	9.2±0.7	2.3
<b>Weighted sum</b>	19.3±0.7	1.8±0.7	10.7

\* Predicting protein binding sites based on one single feature either derived from CCS or from clustering coefficients (CC) or from SNAP2 or EVmutation predictions always succeeded in outperforming a random approach. Performance was compared based on the F1 score given in percentage. The random prediction was calculated for each prediction differently always using the same scores as for the actual prediction assigned to random residues in the protein. The prediction based on SNAP2 achieved the best results compared to random.

**Table S3: Optimizing thresholds for single feature-based predictions. \***

<b>Cutoff</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>	<b>0.6</b>	<b>0.7</b>	<b>0.8</b>	<b>0.9</b>
	<b>F1 score for weighted sum</b>								
<b>Run 1</b>	41.7	39.6	29.3	16.0	6.0	0	0	0	0
<b>Run 2</b>	17.7	19.9	20.6	14.8	25.0	11.7	0	0	0
<b>Run 3</b>	31.6	26.0	36.1	12.5	15.4	0	0	0	0
<b>Run 4</b>	44.0	42.8	34.5	33.3	0	0	0	0	0
<b>Run 5</b>	19.3	14.9	10.5	10.0	0	0	0	0	0
<b>Average</b>	30.9	28.6	26.2	17.3	9.3	2.3	0	0	0
	<b>F1 score for random prediction</b>								
<b>Run 1</b>	27.7	28.8	12.2	0	0	0	0	0	0
<b>Run 2</b>	10.4	11.7	7.1	1.1	0	0	0	0	0
<b>Run 3</b>	22.2	16.5	8.4	0	0	0	0	0	0
<b>Run 4</b>	31.6	23.6	2.8	0	0	0	0	0	0
<b>Run 5</b>	15.8	13.6	12.2	0	0	0	0	0	0
<b>Average</b>	21.5	18.8	8.54	0.22	0	0	0	0	0

\* Combining all eight scores into a weighted sum resulted in a per-residue score ranging from 0 to 1. A residue can be classified as binding or non-binding depending on this score.

Choosing different cut-offs for the classification and comparing them to a random approach using a procedure similar to fivefold cross-validation showed that a cut-off of 0.3 did not lead to the highest F1 score in general but could improve the most over a random prediction for all five data splits. Therefore, this cut-off was chosen to predict binding site residues.

## 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

While `bindPredictML17` (see Section 4.1) achieved satisfactory performance, the evolutionary couplings serving as input are difficult to compute, and the method was mainly optimized for enzymes and DNA-binding proteins limiting its broader applicability. Replacing the hand-crafted features with data-driven inputs and extending the data set including more ligand types could improve performance and allow easier application to a larger number of sequences.

Here, we propose *bindPredictDL*, a method to predict binding residues which improves upon `bindPredictML17`. Instead of relying on evolutionary information, `bindPredictDL` uses embeddings derived from the language model (LM) ProtBERT-BFD [72] as input. Additionally, it replaces the development set of `bindPredictML17`, which used binding annotations from the Protein Data Bank (PDB) [10, 11] and the Protein-DNA Interface Database (PDIdb) [122], with a set which consists of more than twice as many proteins and contains more reliable binding annotations extracted from BioLiP [12, 13]. Using the distinction of different ligand types from BioLiP, `bindPredictDL` does not only predict whether a residue is binding or not, but also to what type of ligand (metal ions, nucleic acids, or small molecules) it binds.

### 4.2.1. Material and Methods

#### Data Set

Protein sequences with annotations of binding residues were extracted from BioLiP [12, 13]. BioLiP provides binding annotations for residues based on structural information from the PDB [10, 11], i.e., it is possible to have multiple annotations of binding residues for one sequence if there exist multiple structures for that sequence. To obtain binding annotations per sequence, we extracted binding information from BioLiP for all chains of high-resolution structures matching a given sequence and combined these annotations. Structures were considered as high-resolution if they were determined through X-ray crystallography [30] with a resolution of  $\leq 2.5\text{\AA}$ . All residues not annotated as binding were considered non-binding.

#### 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

BioLiP distinguishes four different ligand types: metal ions, nucleic acids (i.e., DNA and RNA), peptides, and small molecules (referred to as regular ligands in BioLiP). Here, we focused on predicting ligand binding excluding protein-protein binding. Therefore, we only considered proteins annotated to bind metal ions, nucleic acids, or small molecules and excluded peptides. At point of data set creation (26 November 2019), BioLiP consisted of 104 733 structures with high enough resolution and binding annotations which could be mapped to 14 894 sequences in UniProt [24]. This set was redundancy reduced using UniqueProt [123] with an H-VAL < 0. The final set of 1 314 proteins was split into a test set of 300 proteins and a training set of 1 014 proteins. The training set was used to train the Deep Learning model and to optimize hyperparameters applying five-fold cross-validation. The test set was used to evaluate the final model and to compare the method to bindPredictML17. More information on the data set is given in Table 4.1.

		Training	Test
Metal ions	No. of proteins	455	122
	No. of binding residues	2 374	881
	No. of non-binding residues	77 401	26 763
Nucleic acids	No. of proteins	108	66
	No. of binding residues	2 689	1 470
	No. of non-binding residues	15 582	14 689
Small molecules	No. of proteins	606	220
	No. of binding residues	9 281	3 906
	No. of non-binding residues	94 119	42 629
<b>All</b>	<b>No. of proteins</b>	<b>1 014</b>	<b>300</b>
	<b>No. of binding residues</b>	<b>13 999</b>	<b>5 869</b>
	<b>No. of non-binding residues</b>	<b>156 684</b>	<b>56 820</b>

**Table 4.1.: Development set for bindPredictDL.** The number of proteins, binding residues, and non-binding residues for the three ligand types (metal ions, nucleic acids, and small molecules) and for the entire data set are given. Values from the different ligand types do not sum to the number for “All” because some proteins are annotated to bind multiple ligands.

#### Protein Representation

We used ProtBERT-BFD [72] (in the following called ProtBERT) to create fixed-length vector representations for each residue in a protein sequence. ProtBERT uses the ar-

chitecture of the LM BERT [79], which applies a stack of self-attention [81] layers for masked language modeling. For ProtBERT, a stack of 30 attention layers, each having 16 attention heads with a hidden state size of 1024 (total number of free parameters: 420M) was trained on BFD with 2.1 billion protein sequences [77, 78]. Features learned during pre-training can be transferred to any task requiring protein representations by extracting the hidden states of the LM (transfer learning). To predict which residues in a protein are binding a ligand or not, we extracted 1024-dimensional vectors for each residue from ProtBERT.

#### Machine Learning Architecture

bindPredictDL consists of two independent methods: *bindPredictDL-binary* performs a binary prediction of whether a residue in a protein is binding or not; *bindPredictDL-multi* predicts whether a residue is non-binding or binding to a small molecule, a metal ion, or a nucleic acid (DNA or RNA). For both methods, a two-layer Convolutional Neural Network (CNN) implemented in PyTorch [124] was trained using the Adam optimizer, a learning rate of 0.01, early stopping, and a batch size of 406 resulting in two batches. ProtBERT embeddings with 1024 dimensions were used as input. The first CNN layer consisted of 128 feature channels each with a kernel (sliding window) size of  $k = 5$  mapping the input of size  $L \times 1024$  to an output of  $L \times 128$ . The second layer created the final predictions by applying a CNN with  $k = 5$  and one feature channel for bindPredictDL-binary (size of output:  $L$ ) and three feature channels for bindPredictDL-multi (size of output:  $L \times 3$ ). A residue was considered as non-binding if the output probability was  $< 0.5$  for bindPredictDL-binary, or if all of the output probabilities were  $< 0.5$  for bindPredictDL-multi. The two CNN layers were connected through an exponential linear unit (ELU) [125] and a dropout layer [126], with a dropout-rate of 50% for bindPredictDL-binary and 70% for bindPredictDL-multi.

To adjust for the high class imbalance (8% binding vs. 92% non-binding residues), we used weights in the loss function. For bindPredictDL-binary, positive samples (binding residues) were weighted with a factor of 4.2 simulating that 4.2 times more positive samples were in the training set than actually in there. For bindPredictDL-multi, individual weights were assigned for each ligand type with residues binding to metal ions being weighted with a factor of 8.9, residues binding to nucleic acids with 7.7, and residues binding to small molecules with 4.4. These weights allowed to adjust for the imbalance

between binding and non-binding residues within one class but not for the imbalance between classes, i.e., imbalance between the number of residues binding to small molecules, metal ions, or nucleic acids. Applying higher weights in the loss function increases recall (Eqn. 4.1), lower weights increase precision (Eqn. 4.2). The chosen weights led to optimal results in terms of F1 score (Eqn. 4.3) and MCC (Eqn. 4.4).

Both methods were trained using five-fold cross-validation. Every protein was used for validation exactly once.

### Homology-based Inference

To transfer annotations by homology, PSI-BLAST [58] alignments were used. For all proteins in the development set (training + test), we generated PSI-BLAST profiles with two iterations and E-value  $\leq 10^{-3}$  using an 80% non-redundant database combining UniProt [24] and PDB [10] following a standard protocol implemented also for other methods [60, 62, 68]. The resulting profiles were then aligned at E-value  $\leq 10^{-9}$  against all proteins with known binding annotations. For performance estimates, self-hits were excluded. Taking the hit of all retrieved alignments with the highest pairwise sequence identity to the query, a local alignment was calculated between query and hit using the Smith-Waterman algorithm [127]. Then, binding annotations were transferred between the aligned positions.

### Performance Evaluation

To assess whether a prediction was correct or not, we used the following standard annotations: True positives (TP) were residues correctly predicted as binding, false positives (FP) were incorrectly predicted as binding, but were annotated as non-binding, true negatives (TN) were correctly predicted as non-binding, and false negatives (FN) were not predicted as binding while being annotated as binding. Based on this classification for each residue, we evaluated performance using standard performance measurements, namely recall (or sensitivity) (Eqn. 4.1), precision (Eqn. 4.2), F1 score (Eqn. 4.3), and Matthews Correlation Coefficient (MCC) (Eqn. 4.4).

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$F1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \quad (4.3)$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4.4)$$

The coverage (Eqn. 4.5) indicated for how many proteins with a binding site any residue was predicted as binding. Accordingly, the negative coverage gave the fraction of proteins without a binding site for which also no binding site was predicted. Since our data set only consisted of proteins with a binding site, the negative coverage was only considered for the task of predicting binding residues for different types of ligands. In this case, the negative coverage gave the fraction of proteins without a binding site for a specific ligand type for which also no binding site for this ligand type was predicted (Eqn. 4.6).

$$Coverage = \frac{\text{No. of proteins with binding predictions}}{\text{No. of proteins with binding annotations}} \quad (4.5)$$

$$Neg. Coverage(l) = \frac{\text{No. of proteins without binding predictions for ligand } l}{\text{No. of proteins without binding annotations for ligand } l} \quad (4.6)$$

Each performance measurement was calculated for each protein individually, the mean was calculated over the resulting distribution, and standard errors defined as  $SE = SD/\sqrt{n-1}$  were calculated as error estimates, with  $n$  being the number of proteins and  $SD$  representing the standard deviation.

#### 4.2.2. Preliminary Results

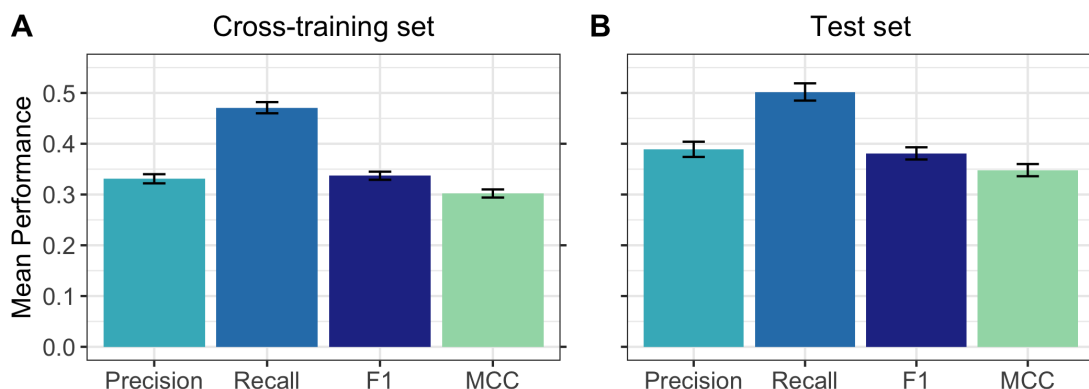
##### **bindPredictDL-binary predicted binding residues with F1=34-38%**

bindPredictDL-binary used ProtBERT [72] embeddings to predict, for each residue in a protein sequence, whether this residue is binding to a ligand or not. The method achieved  $F1 = 33.7 \pm 0.8\%$  on the cross-training set (Fig. 4.1A) and  $F1 = 38 \pm 1\%$



## 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

on the test set (Fig. 4.1B). For the cross-training set, the coverage (Eqn. 4.5) was 95%, i.e., at least one residue was predicted as binding for 959 proteins in the cross-training set, while it provided any binding prediction for 296 of the proteins in the test set corresponding to a coverage of 99%.



**Figure 4.1.: Performance for bindPredictDL-binary.** bindPredictDL-binary achieved **A.**  $Precision = 33.1 \pm 0.8\%$ ,  $Recall = 47 \pm 1\%$ ,  $F1 = 33.7 \pm 0.8\%$ , and  $MCC = 0.30 \pm 0.01$  on the cross-training set, and **B.**  $Precision = 39 \pm 2\%$ ,  $Recall = 50 \pm 2\%$ ,  $F1 = 38 \pm 1\%$ , and  $MCC = 0.35 \pm 0.01$  on the test set. While hyperparameters were optimized for the cross-training set, the method performed better on the test set. Error bars indicate standard errors.

The data set consisted of proteins binding to three major groups of ligands. bindPredictDL-binary was not trained to distinguish between these groups and was only provided information on whether a residue is binding or not, but not to what type of ligand it binds. However, it still performed differently for the different types with achieving better F1 scores for small molecules and nucleic acids as ligands than for metal ions (Table 4.2). Precision was especially low for metal ions ( $19 \pm 2\%$ ), while recall was higher than for the other ligands (Table 4.2). In fact, the binding sites for metal ions are much smaller than for the other ligands. On average, 3% of a protein’s residues form binding sites to metal ions, while 9% and 15% make binding sites for small molecules and nucleic acids, respectively (Table 4.1). The model apparently learned to predict larger numbers of binding residues mirroring the expected size of binding sites for small molecules and nucleic acids. This led to an over-prediction for metal ions and, consequently, to a decrease in precision compared to the other ligands.

#### 4. Prediction of Protein Binding Residues from Sequence

---

	Precision	Recall	F1	MCC
Overall	$39 \pm 2\%$	$50 \pm 2\%$	$38 \pm 1\%$	$0.35 \pm 0.01$
Metal ions	$19 \pm 2\%$	$61 \pm 2\%$	$25 \pm 2\%$	$0.28 \pm 0.02$
Nucleic acids	$43 \pm 3\%$	$43 \pm 3\%$	$39 \pm 3\%$	$0.34 \pm 0.03$
Small molecules	$40 \pm 2\%$	$49 \pm 2\%$	$39 \pm 2\%$	$0.36 \pm 0.01$

**Table 4.2.: Test set performance by ligand type.** The method achieved a higher precision for nucleic acids and small molecules than for metal ions, while recall was highest for metal ions. Standard errors are given as error estimates.

While parameters were optimized for the cross-training set, the model surprisingly performed better on the test set (Fig. 4.1). The test set was constructed to allow maximum overlap with the development set of bindPredictML17 which consisted solely of enzymes and DNA-binding proteins. Thus, the better performance on the test set could indicate that binding residues of enzymes were more accurately predicted because the binding site is better defined for enzymes than for other proteins.

#### **bindPredictDL-binary clearly outperformed bindPredictML17**

Of the 300 proteins in the test set, 225 were also part of the development set of bindPredictML17. Using the predictions for these 225 proteins from the respective cross-validation splits of bindPredictML17 allowed an unbiased comparison of both methods because there was no overlap between training and test set for either bindPredictML17 or bindPredictDL-binary.

bindPredictDL-binary clearly outperformed bindPredictML17. While bindPredictML17 achieved  $F1 = 34 \pm 1\%$  on this set of 225 proteins, bindPredictDL-binary achieved  $F1 = 41 \pm 1\%$  improving upon the old method by seven percentage points (Table 4.3). However, compared to bindPredictML17, bindPredictDL-binary did not make a binding prediction for all proteins resulting in a coverage of 99% (223 proteins).

Unlike bindPredictDL, bindPredictML17 was trained using annotations available through PDB [10, 11] for enzymes and through PDIdb [122] for DNA-binding proteins. However, as outlined in Section 1.1.2, binding annotations in the PDB do not necessarily reflect biologically relevant binding sites. Therefore, we used annotations from BioLiP [12, 13] to train bindPredictDL. Considering the predictions of bindPredictML17 for the 225 test proteins, we observed a better performance for the BioLiP annotations than for the PDB

#### 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

	Precision	Recall	F1	MCC	TP	FP	TN	FN	Coverage
bindPredictML17 (PDB)	39 ± 1%	31 ± 2%	29 ± 1%	0.23 ± 0.01	1 720	3 107	39 018	5 162	225 (100%)
bindPredictML17 (BioLiP)	37 ± 1%	42 ± 2%	34 ± 1%	0.30 ± 0.01	1 687	3 140	41 023	3 157	225 (100%)
bindPredictDL-binary	43 ± 2%	53 ± 2%	41 ± 1%	0.38 ± 0.01	2 252	3 533	40 630	2 592	223 (99%)
bindPredictDL-multi	45 ± 2%	48 ± 2%	39 ± 1%	0.37 ± 0.01	1 920	2 716	41 447	2 924	217 (96%)

**Table 4.3.: Performance comparison with bindPredictML17.** 225 of the 300 proteins in the test set were also part of the development set for bindPredictML17 which could be used to compare bindPredictML17 and bindPredictDL. Both bindPredictDL-binary and bindPredictDL-multi performed better than bindPredictML17. Also, bindPredictML17 managed to better predict the more reliable binding annotations from BioLiP than the ones from the PDB, although it was trained on PDB annotations. This indicated that the improvement of bindPredictDL upon bindPredictML17 was partially caused by training on more reliable annotations. Standard errors are given as error estimates.

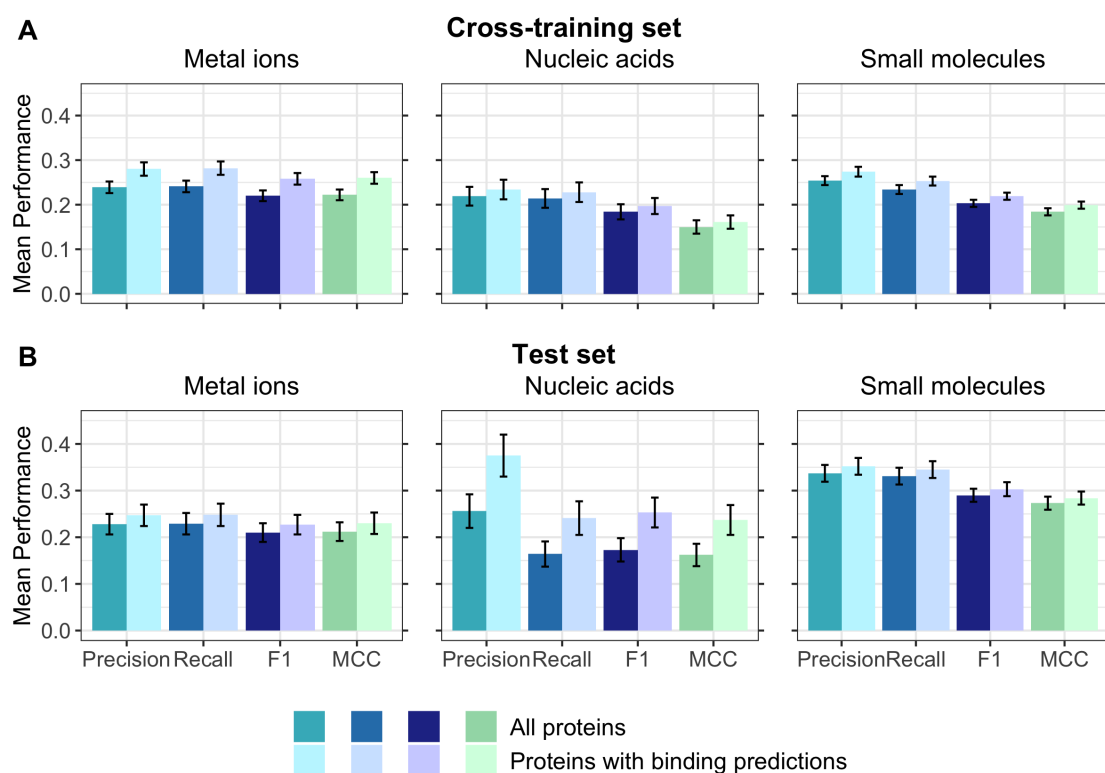
annotations although bindPredictML17 was trained on annotations from PDB (Table 4.3 “bindPredictML17 (PDB)” versus “bindPredictML17 (BioLiP)”). First of all, this showed, while being trained on noisy data, the seemingly false negative predictions of bindPredictML17 (FN in Table 4.3) were in fact often due to a wrong annotation in PDB. Without any re-training, the number of FN dropped by almost 40% when evaluating on annotations from BioLiP (Table 4.3). Secondly, these differences also highlighted the importance of using high-quality binding annotations. The binding annotations from PDB contain a lot of noise due to biologically irrelevant ligands. Training on those noisy data worsened performance, and the improvement of bindPredictDL upon bindPredictML17 could partially be attributed to training on annotations from BioLiP.

#### bindPredictDL-multi could distinguish between different ligand types

To extend bindPredictDL-binary, we proposed bindPredictDL-multi. Using the same input (ProtBERT embeddings) and data set (Table 4.1) as for bindPredictDL-binary, bindPredictDL-multi was trained to not only predict whether a residue is binding or not but also to which ligand type it binds. Binding residues were predicted with  $F1 = 22 \pm 1\%$  for metal ions,  $F1 = 18 \pm 2\%$  for nucleic acids, and  $F1 = 20 \pm 1\%$  for small molecules (Fig. 4.2A, darker colored bars). Performance was mainly limited by a low coverage

#### 4. Prediction of Protein Binding Residues from Sequence

(Eqn. 4.5) and low negative coverage (Eqn. 4.6) (Table 4.4). If no binding predictions were made for a protein binding ligand  $l$ , precision, recall, F1, and MCC for this prediction were set to 0. Same was true for the performance values for proteins not binding to a certain ligand but for which the method predicted some binding residues. If we only considered proteins with predictions for a specific ligand type for the performance evaluation, F1 rose to  $26 \pm 1\%$ ,  $20 \pm 2\%$ , and  $22 \pm 1\%$  for metal ions, nucleic acids, and small molecules, respectively (Fig. 4.2A, lighter colored bars).



**Figure 4.2.: Performance for bindPredictDL-multi.** Performance for prediction of residues binding to metal ions, nucleic acids, and small molecules for **A.** the cross-training set and **B.** the test set. Darker colored bars indicate performance values for all proteins with all values for the prediction of binding to ligand type  $l$  were set to 0 if no residue was predicted to bind to  $l$ . Lighter colored bars indicate prediction performance only for the proteins covered by the prediction, i.e., at least one residue is predicted to bind to ligand type  $l$ . In general, performance improved if only proteins with any binding prediction were considered (lighter colored bars). Error bars indicate standard errors.

#### 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

	Coverage	Negative Coverage
Metal ions	75%	48%
Nucleic acids	86%	86%
Small molecules	89%	33%

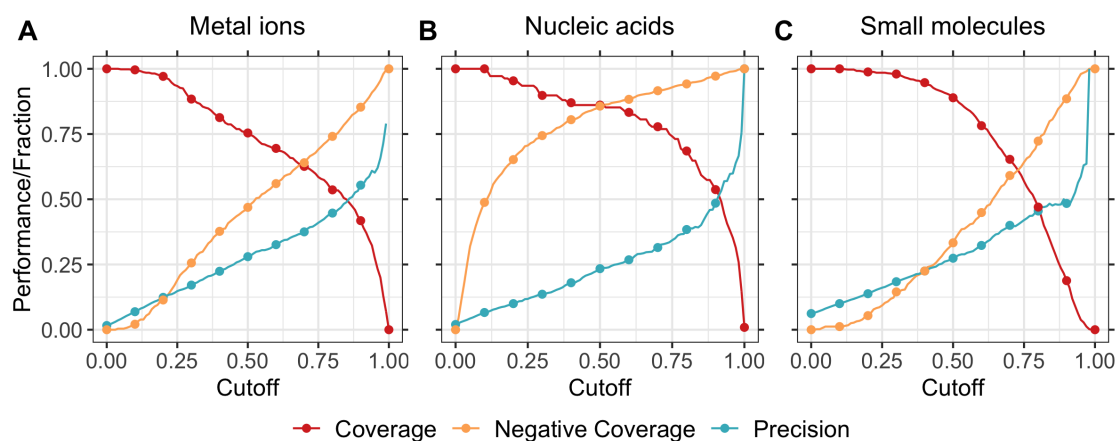
**Table 4.4.: Coverage and Negative Coverage for bindPredictDL-multi.** Coverage (Eqn. 4.5) indicates the fraction of proteins binding to ligand type  $l$  for which at least one residue was predicted to bind to  $l$  (without considering whether the prediction was correct or not). Negative Coverage (Eqn. 4.6) indicates the fraction of proteins not binding to ligand type  $l$  for which also no residue was predicted to bind to  $l$ . Coverage was lowest for proteins binding to metal ions, while negative coverage was lowest for proteins binding to small molecules.

Separately predicting whether a residue binds to a metal ion, a nucleic acid, or a small molecule was a more complicated prediction task than the binary classification of bindPredictDL-binary. Therefore, it was important to investigate how much predictive power we lost by performing this more complicated task. To compare bindPredictDL-multi with bindPredictDL-binary, we mapped the ligand-specific predictions to a binary output by considering every residue predicted as binding which was predicted to bind at least one of the three ligand types. For this binary prediction task, bindPredictDL-multi achieved  $32.0 \pm 0.8\%$  on the cross-training set performing only 1.7 percentage points worse than bindPredictDL-binary and almost remaining within two standard errors. Thus, the predictions of bindPredictDL-multi provided additional information about the bound ligand without losing predictive power compared to bindPredictDL-binary.

Applying bindPredictDL-multi to the test set, the method achieved  $F1 = 37 \pm 1\%$  being only one percentage point worse than bindPredictDL-binary. Residues binding to metal ions, nucleic acids, and small molecules could be predicted with  $F1 = 21 \pm 2\%$ ,  $F1 = 17 \pm 3\%$ , and  $F1 = 29 \pm 1\%$ , respectively (Fig. 4.2B). As bindPredictDL-binary, bindPredictDL-multi clearly outperformed bindPredictML17. It achieved  $F1 = 39 \pm 1\%$  and a coverage of 96% on the 225 proteins from the test set of bindPredictDL overlapping with the development set of bindPredictML17. Therefore,  $F1$  increased by five percentage points for bindPredictDL-multi compared to bindPredictML17 (Table 4.3).

**Focus on more reliable predictions increased precision**

On the cross-training set, bindPredictDL-multi achieved a precision of  $24 \pm 1\%$ ,  $22 \pm 2\%$ , and  $25 \pm 2\%$  for metal ions, nucleic acids, and small molecules, respectively (Fig. 4.2A, darker colored bars). Considering only proteins for which at least one residue was predicted as binding, precision rose to  $28 \pm 2\%$  (metal ions),  $23 \pm 2\%$  (nucleic acids), and  $27 \pm 1\%$  (small molecules) (Fig. 4.2A, lighter color bars). Precision could be further increased if stricter cutoffs were applied to define a residue as binding (Fig. 4.3). By default, all predictions with an output probability  $\geq 0.5$  were considered as binding residues. Increasing this cutoff led to an increase in negative coverage, i.e., for more proteins not binding to a specific ligand type, no residues were predicted to be binding (Fig. 4.3, orange line). If a binding prediction was made for a non-binding protein, all performance measurements were set to 0. Therefore, increasing the negative coverage reduced the number of proteins with a precision of 0. Also, the number of false positives was generally reduced leading to an overall increase of precision (Fig. 4.3, lighter blue line). While stricter prediction cutoffs allowed to focus on more reliable predictions for a few proteins, lower cutoffs led to a general increase of coverage (Fig. 4.3, red line). Therefore, while resulting in more false positive predictions, lower cutoffs provided predictions for more proteins.



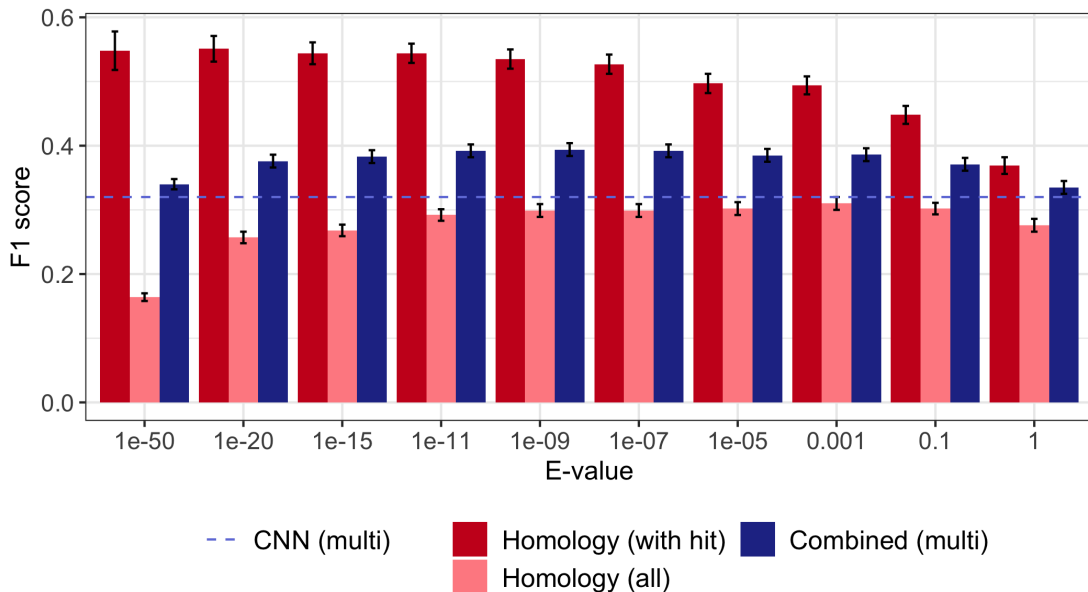
**Figure 4.3.: Prediction performance for different thresholds.** Residues were considered as binding if the output probability was greater or equal to a specific cutoff. Choosing larger values for this cutoff led to an increase in negative coverage and precision for **A.** metal ions, **B.** nucleic acids, and **C.** small molecules. On the other hand, lower cutoffs increased coverage allowing predictions for more proteins.

### Combination with homology-based inference further improved performance

Using homology-based inference to predict binding residues yielded very good results for low E-value thresholds, but hits at those thresholds were only found for very few proteins. For example, for E-value  $\leq 10^{-50}$ , homology-based inference achieved  $F1 = 55 \pm 3\%$  (Fig. 4.4, leftmost dark red bar), but a hit could only be identified for 102 of 1014 proteins. When only using homology-based inference to make a prediction for all proteins, a random decision would have to be made if no homolog with known binding annotations was available at the given threshold. This led to an immense drop in performance to  $F1 = 16.4 \pm 0.6\%$  for E-value  $\leq 10^{-50}$  (Fig. 4.4, leftmost light red bar). To harness the strong performance of homology-based inference while allowing better than random predictions for proteins without close homologs, we combined bindPredictDL with homology-based inference applying a simple protocol: Predict binding residues through homology-based inference if available; otherwise use the ML method. This combination achieved optimal performance at an E-value threshold of  $10^{-9}$  leading to  $F1 = 39 \pm 1\%$  for bindPredictDL-multi (when converting the three outputs to the binary prediction “binding/non-binding”) (Fig. 4.4, blue bar at E-value =  $10^{-9}$ ).

In the used protocol, homologs were identified by determining the most sequence-similar protein for an E-value below a certain threshold. For this protein, a local alignment was calculated, and binding annotations were inferred between aligned positions. If the alignment did not contain any binding annotations, the hit was discarded and the ML method was applied instead. While already reaching much higher performance than the ML method alone, the protocol for homology-based inference could be further improved to avoid discarding hits because the local alignment did not contain any binding annotations. Instead of using the most sequence-similar hit, we could use the hit with the local alignment covering most binding annotations. Alternatively, if the most sequence-similar hit did not contain any binding annotations in the aligned sequence part, we could choose the second best hit and continue for all hits that were found below the given E-value threshold. In this case, we would only switch to the ML method if none of the local alignments to any of the found hits contained binding annotations.

However, for simplicity, we used the standard protocol for now, i.e., if the local alignment of the most sequence-similar hit to the query contains binding annotations, transfer those to the query, otherwise, apply the ML method. Using this approach and the optimal E-value threshold of  $10^{-9}$  to combine homology-based inference and ML improved perfor-



**Figure 4.4.: Performance of homology-based inference for different E-value thresholds.** Performance as measured by the F1 score for homology-based inference varied with the E-value thresholds (red bars). The highest F1 was reached at E-value  $\leq 10^{-50}$ . However, if forcing predictions for all proteins by assigning binding residues at random if no homolog was available, F1 dropped to  $16.4 \pm 0.5\%$  (leftmost light red bar). The combination of homology-based inference and ML (blue bars) performed best for E-value  $\leq 10^{-9}$ . The dashed line indicates the performance for just using the CNN to predict binding residues by ligand type which are mapped to a binary output “binding/non-binding”. Error bars indicate standard errors.

mance on the test set. Compared to just using ML, the F1 score of bindPredictDL-multi improved by seven percentage points resulting in  $F1 = 44 \pm 2\%$  (Table 4.5).

Homology-based inference also improved performance for the individual ligands (Fig. 4.5) while still experiencing the same issues as the method solely based on ML, i.e., missing predictions for proteins annotated to bind this ligand (low coverage) and wrong predictions for proteins not annotated to bind this ligand (low negative coverage). Combining homology-based inference with ML resulted in the final method bindPredictDL-multi which allowed to predict whether a residue is binding to a metal ion, a nucleic acid, or a small molecule with  $F1 = 27 \pm 3\%$ ,  $F1 = 20 \pm 3\%$ , and  $F1 = 39 \pm 2\%$ , respectively (Fig. 4.5). The binary prediction of whether a residue is binding or not



## 4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types

	Precision	Recall	F1	MCC
bindPredictDL-binary (only ML)	$39 \pm 2\%$	$50 \pm 2\%$	$38 \pm 1\%$	$0.35 \pm 0.01$
bindPredictDL-multi (only ML)	$42 \pm 2\%$	$45 \pm 2\%$	$37 \pm 1\%$	$0.34 \pm 0.01$
bindPredictDL-binary (with HBI)	$54 \pm 2\%$	$48 \pm 2\%$	$45 \pm 2\%$	$0.43 \pm 0.02$
bindPredictDL-multi (with HBI)	$54 \pm 2\%$	$46 \pm 2\%$	$44 \pm 2\%$	$0.43 \pm 0.02$

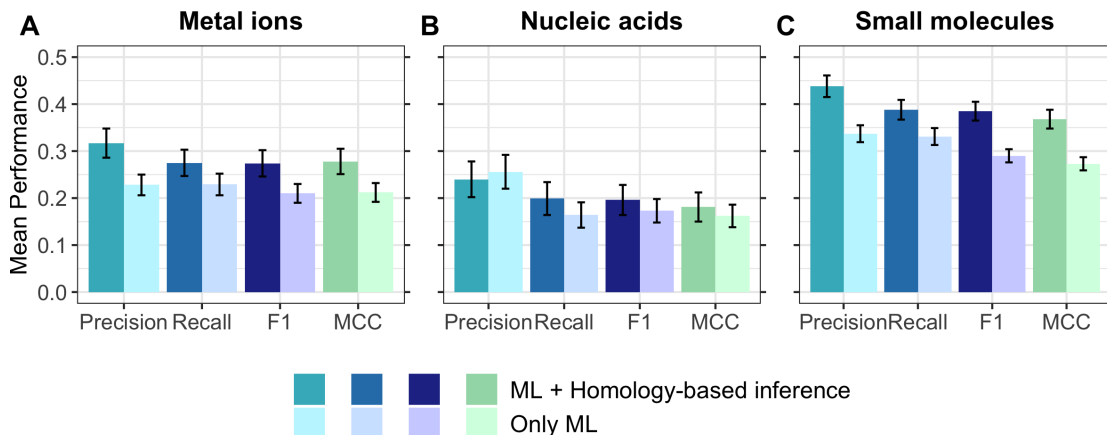
**Table 4.5.: Test set performance with and without homology-based inference.** Combining the ML method with homology-based inference improved  $F1$  by seven percentage points both for bindPredictDL-binary and bindPredictDL-multi. Standard errors are given as error estimates. HBI: Homology-based inference.

using the combination of ML and homology-based inference achieved  $F1 = 44 \pm 2\%$ . This performance was only slightly worse and still within the error margin compared to the combination of homology-based inference and bindPredictDL-binary, which was only trained on distinguishing between binding and non-binding residues without considering the bound ligand (Table 4.5).

### 4.2.3. Conclusion

With bindPredictDL-multi, we proposed a new method to predict whether a residue is binding metal ions, nucleic acids, or small molecules, or is non-binding. It used ProtBERT embeddings as input which were derived from a pre-trained LM and could easily be obtained for all protein sequences. A binary distinction of binding residues from non-binding residues could be derived from the ligand-specific predictions by mapping the three individual outputs to one prediction. In this case, each residue which was predicted to bind at least one of the three ligand types was considered as binding and all other residues as non-binding. This binary prediction of bindPredictDL-multi outperformed bindPredictML17 by five percentage points (Table 4.3). On the entire test set, bindPredictDL-multi achieved  $F1 = 37 \pm 1\%$  for the classification of residues into binding and non-binding; bindPredictDL-binary, which was trained explicitly for this binary task, achieved  $F1 = 38 \pm 1\%$ . While being one percentage point worse than bindPredictDL-binary, bindPredictDL-multi solved the more complicated task of predicting the type of bound ligand and, therefore, provided reliable predictions of binding residues as well as additional information about the bound ligand.

#### 4. Prediction of Protein Binding Residues from Sequence



**Figure 4.5.: Performance for bindPredictML-multi combined with homology-based inference.** We combined homology-based inference and ML and transferred annotations between close homologs if available, and run *de novo* prediction, otherwise. This improved performance for the prediction whether a residue binds a certain ligand or not for all ligands (**A.** Metal ions, **B.** Nucleic acids, **C.** Small molecules) compared to just applying the ML method (lighter colored bars). This resulted in the final version of bindPredictDL-multi achieving  $F1 = 27 \pm 3\%$ ,  $F1 = 20 \pm 3\%$ , and  $F1 = 39 \pm 2\%$  for metal ions, nucleic acids, and small molecules, respectively. Error bars indicate standard errors.

The performance of bindPredictDL was limited to a certain extent by a low coverage and low negative coverage (Table 4.4). For many proteins which were annotated to bind a specific ligand, no binding predictions were made, while residues were predicted as binding in proteins not bound to the ligand. Considering only proteins with at least one residue predicted as binding and applying a stricter cutoff to classify a residue as binding increased precision (Fig. 4.3).

The performance of bindPredictDL was improved through a simple combination of ML and homology-based inference: If a homologous protein with binding annotations was found, annotations of this protein were transferred to the query. Otherwise, the ML method was applied. This combination resulted in  $F1 = 27 \pm 3\%$ ,  $F1 = 20 \pm 3\%$ , and  $F1 = 39 \pm 2\%$  for the prediction of residues binding to metal ions, nucleic acids, and small molecules, respectively (Fig. 4.5). The binary prediction of whether a residue is binding or not achieved  $F1 = 44 \pm 2\%$  leading to an improvement of seven percentage points over the approach not using homology-based inference.

#### *4.2. Protein Embeddings Allow Prediction of Binding Residues for Various Ligand Types*

---

bindPredictDL represents a powerful method to predict binding residues. It relies solely on sequence information, and the usage of embeddings removes the need of hand-crafted features, which were used for, e.g., bindPredictML17. bindPredictDL allows the distinction of binding to three different ligand types while also achieving good performance for the binary prediction task of classifying residues as binding or non-binding. Since only a few residues in a protein sequence are binding, a more sophisticated approach to perform homology-based inference, which focuses more on the best local alignment than the overall most sequence-similar hit, could improve performance of the method further. Making the method publicly available as a web server will allow easy access to binding residue predictions also for non-expert users.



# 5. Detailed Prediction of Protein Sub-nuclear Localization

## 5.1. Preface

The nucleus contains several distinct substructures which are associated with different functions. Extensive knowledge about those substructures and which proteins are associated with them can help to better understand the interior nuclear mechanisms. However, nuclear substructures are very dynamic and some are exclusively formed during particular cell stages through interaction with DNA, RNA, and proteins [128, 129]. These dynamic rearrangements complicate experimental annotations of sub-nuclear localizations creating a need for accurate prediction methods.

We developed *LocNuclei*, a new method that predicts nuclear substructures from sequence alone. It distinguishes between 13 different sub-nuclear localizations and also predicts whether a protein only occurs in the nucleus or is also native to other sub-cellular compartments (i.e., is a *traveler protein*). *LocNuclei* applies homology-based inference to transfer annotations from a sequence-similar protein to the target protein if such a protein exists, and otherwise, relies on an SVM using the Profile Kernel [65, 66] to predict sub-nuclear compartments. Using this approach, sub-nuclear localization of a protein was predicted with  $Q_{13} = 62 \pm 3\%$ , and traveler proteins were identified with  $Q_2 = 72 \pm 2\%$ , where  $Q_n$  is defined as

$$Q_n = 100 * \frac{\sum_{i=1}^n \text{number of proteins correctly predicted in class } i}{\sum_{i=1}^n \text{total number of proteins observed in class } i} \quad (5.1)$$

and  $n$  is the number of classes. Since nuclear compartments are dynamic and sometimes only form during certain cell stages, it is likely that nuclear proteins are associated with

multiple compartments at different time points of the cell cycle. LocNuclei models this dynamic by allowing predictions of multiple compartments for one protein.

The analysis of GO term enrichment and protein-protein interactions showed that predictions of sub-nuclear structures through LocNuclei reveal functional insights of nuclear proteins. The source code and data sets are available in a GitHub repository: <https://github.com/Rostlab/LocNuclei>.

**Author contribution:** Sebastian Seitz collected the data and implemented the first version of LocNuclei together with Tatyana Goldberg and Mikael Bodén. I refined the existing implementation, performed analysis of GO enrichment and protein-protein interactions for nuclear proteins, and wrote the major part of the manuscript. All authors drafted the manuscript.

## 5.2. Journal Article: Littmann, Goldberg *et al.*, BMC Bioinformatics (2019)

**Reference:** Littmann, M., Goldberg, T., Seitz, S., Bodén, M., and Rost, B. Detailed prediction of protein sub-nuclear localization. *BMC Bioinformatics*, 20(205), 2019. doi:10.1186/s12859-019-2790-9

**Copyright Notice:** Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>).

METHODOLOGY ARTICLE

Open Access

# Detailed prediction of protein sub-nuclear localization



Maria Littmann<sup>1\*†</sup> , Tatyana Goldberg<sup>1†</sup>, Sebastian Seitz<sup>1</sup>, Mikael Bodén<sup>2</sup> and Burkhard Rost<sup>1,3,4,5</sup>

## Abstract

**Background:** Sub-nuclear structures or locations are associated with various nuclear processes. Proteins localized in these substructures are important to understand the interior nuclear mechanisms. Despite advances in high-throughput methods, experimental protein annotations remain limited. Predictions of cellular compartments have become very accurate, largely at the expense of leaving out substructures inside the nucleus making a fine-grained analysis impossible.

**Results:** Here, we present a new method (*LocNuclei*) that predicts nuclear substructures from sequence alone. *LocNuclei* used a string-based Profile Kernel with Support Vector Machines (SVMs). It distinguishes sub-nuclear localization in 13 distinct substructures and distinguishes between nuclear proteins confined to the nucleus and those that are also native to other compartments (traveler proteins). High performance was achieved by implicitly leveraging a large biological knowledge-base in creating predictions by homology-based inference through BLAST. Using this approach, the performance reached AUC = 0.70–0.74 and Q13 = 59–65%. Travelling proteins (nucleus and other) were identified at Q2 = 70–74%. A Gene Ontology (GO) analysis of the enrichment of biological processes revealed that the predicted sub-nuclear compartments matched the expected functionality. Analysis of protein-protein interactions (PPI) show that formation of compartments and functionality of proteins in these compartments highly rely on interactions between proteins. This suggested that the *LocNuclei* predictions carry important information about function. The source code and data sets are available through GitHub: <https://github.com/Rostlab/LocNuclei>.

**Conclusions:** *LocNuclei* predicts subnuclear compartments and traveler proteins accurately. These predictions carry important information about functionality and PPIs.

**Keywords:** Sub-nuclear localization, Traveler proteins, Prediction, Support vector machines (SVM), Profile kernel, GO enrichment, Evolutionary information, Predict protein function

## Background

The nucleus was the first sub-cellular organelle to be discovered as early as in the seventeenth century [1]. It is enclosed by a membrane and only found in eukaryotic cells (Greek “eu” εν: true, “karyon” καρυον: kernel, i.e. cells with a core, Latin: nucleus). The nucleus contains most of the genetic material, organized in chromosomes, and is the site for DNA replication and transcription. Nuclear proteins are synthesized mostly on the ribosomes in the cytoplasm and have to be transported back into the nucleus for proper function. Import into and

export out of the nucleus differ in several ways from the transport to other sub-cellular compartments. For instance, all proteins have to pass through a large structure in the nuclear envelope known as the nuclear pore complex (NPC) [2, 3]. Nuclear proteins can be transported in their fully folded conformation [3]. Transport is often regulated through binding to specific proteins, called *karyopherins*. Karyopherins bind by recognizing nuclear localization signals (NLS for import into the nucleus) or nuclear export signals (NES; for export from the nucleus) in the amino acid sequence of their cargo proteins [4]. Relying only on these NLS and NES fails to identify nuclear proteins because many known signals are too unspecific in sequence (match in many non-nuclear proteins) and for most known nuclear proteins such signals remain unknown [5–7].

\* Correspondence: [littmann@rostlab.org](mailto:littmann@rostlab.org); [assistant@rostlab.org](mailto:assistant@rostlab.org)

<sup>†</sup>Maria Littmann and Tatyana Goldberg these authors share first authorship

<sup>1</sup>Department of Informatics, Bioinformatics & Computational Biology - i12, TUM (Technical University of Munich), Boltzmannstr. 3, 85748 Garching/Munich, Germany

Full list of author information is available at the end of the article



The nucleus is a compartment separated by two membranes that contains several distinct sub-structures, each associated with distinct sets of function. These nuclear sub-structures are not enclosed by membranes and are very dynamic. Nuclear sub-structures can be in continuous flux; some are exclusively formed during particular cell stages through interaction with DNA, RNA and proteins [8, 9]. These dynamic rearrangements complicate experimental annotations. Translocation within the nucleus has been linked to NLS- and NES-like signals [10, 11]. However, this process is not well understood [8].

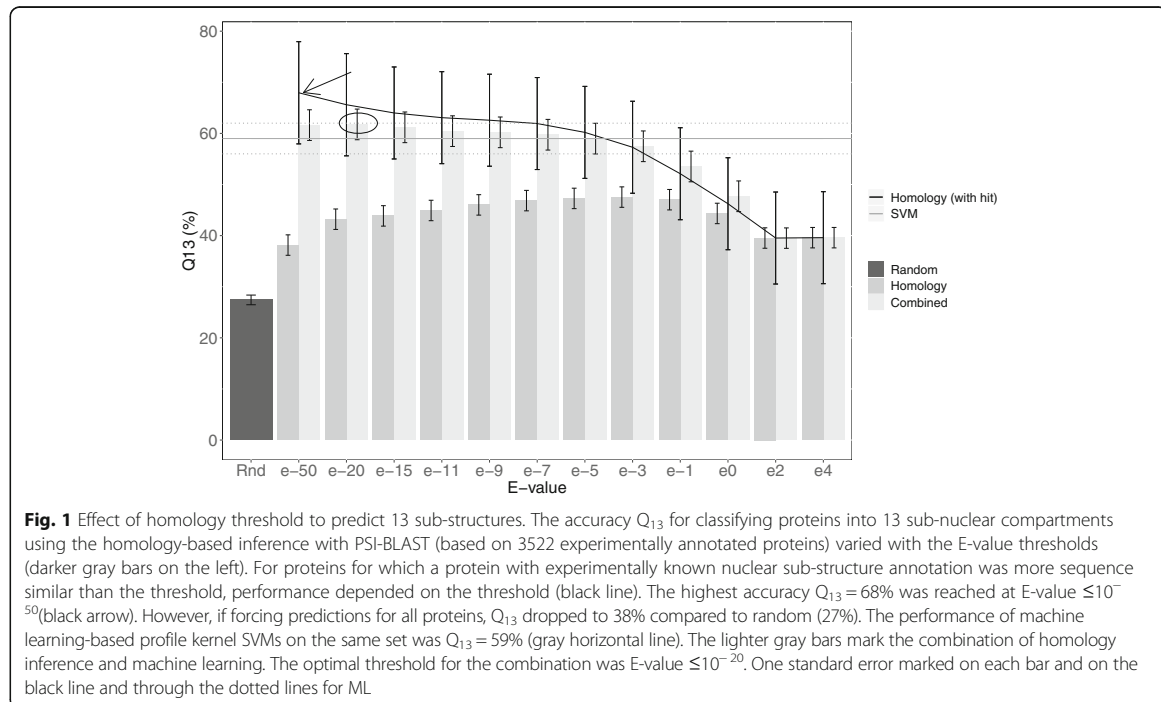
## Results

### High performance: Q13 = 62% and Q2 = 72%

LocNuclei describes two separate prediction methods: (1) predict one of 13 nuclear sub-structures and (2) distinguish proteins functional only in the nucleus vs. travel proteins, i.e. those functional in the nucleus and other compartments. Each of those two methods combines two different algorithms: (i) homology-based inference and (ii) machine learning-based prediction (through profile kernel SVMs). For the prediction task of 13 sub-nuclear compartments, the homology-based inference for proteins for which experimentally annotated homologs were available was most accurate with  $Q_{13} = 68\%$  at  $E\text{-value} \leq 10^{-50}$  (Fig. 1 black arrow). However, if only using homology-based inference, a random decision had to be made when no homolog of known localization was available at a given

threshold. Thus, the  $Q_{13}$  dropped to 38% (Fig. 1: left bar at  $E\text{-value} 10^{-50}$ ). This was still statistically significantly above random (Fig. 1: standard error bars substantially above random performance of 27% shown at the leftmost bar). On the same test set, the de novo-based inference employing a battery of 13 SVM classifiers achieved an almost three-fold higher level of  $Q_{13} = 59\%$  (Fig. 1: 2nd bar from the left). This result encouraged the application of a simple protocol: use homology-based inference when available, else use the machine learning method. The accuracy of homology-based inference decreased for less stringent  $E\text{-value}$  thresholds (Fig. 1: line decreases toward right). We chose the PSI-BLAST  $E\text{-value}$  of  $10^{-20}$  as the decision threshold between homology-based inference and machine learning based de novo prediction because the simple combination of homology-based and de novo was highest (the performance was determined using cross-validation/cross-training, i.e. NOT the testing set). The combined method, LocNuclei, outperformed both its components (Fig. 1: circle above bar for SVM and homology), reaching an overall accuracy of  $Q_{13} = 62 \pm 3\%$  (Fig. 1: circle).

In terms of relative contributions of HB vs. ML for our data set, the numbers were as follows. From the 1934 subnuclear proteins in our data set, 736 (38%) were predicted through homology-based inference (HB), and 1096 (57%) through the SVM Profile Kernel (ML). For 102 proteins (5%), neither HB nor any of the 13 SVMs





predicted any nuclear sub-compartment (note: this was only a subset of all prediction mistakes).

For the second prediction task (nuclear-only vs. traveler proteins) the final method combined homology-based inference and machine learning (again a Profile Kernel SVM) essentially in the same straightforward manner: take HB if possible. The final method (also referred to as *LocNuclei*) performed best at the PSI-BLAST  $E$ -value  $\leq 10^{-5}$  reaching an overall performance of  $Q_2 = 72 \pm 2\%$  (Additional file 1: Figure S1). In detail of the 1098 nuclear proteins in the corresponding data set, 419 proteins (38%) were predicted by homology-based inference, all other 679 (62%) using the SVM Profile Kernel.

#### Good predictions also for minority classes

*LocNuclei* distinguished between 13 different nuclear sub-structures. One crucial challenge for predicting many classes was the lack of experimental annotations for the *minority classes*, i.e. those with fewer known proteins. For instance, the SVM had to generalize from only 14 proteins in the *spindle apparatus* and from only 13 in the *perinucleolar sub-structure* (Additional file 1: Table S1). Nevertheless, *LocNuclei* succeeded in predicting for minority classes, e.g. 8 of the 14 samples for *spindle apparatus* were predicted correctly. The worst performance was observed for the *Cajal body*: 10 of the 42 predicted in this sub-structure were correctly predicted, while an equal number of 10 proteins were mis-predicted to be in the nucleoplasm (Table 1). All these ten mis-predictions originated from the SVM prediction. Using exclusively homology-based inference correctly predicted 8 of 42 Cajal bodies and no misclassification to nucleoplasm would occur (Additional file 1: Table S2).

#### Reliability index allows focus on best predictions

For each prediction, *LocNuclei* also provides a reliability index (RI) that reflects the prediction strength. The RI was scaled to values between 0 (uncertain prediction) and 100 (reliable prediction). Although the RI scaling did not correlate with performance throughout its entire interval, it enables users to focus on reliably predicted proteins: e.g. of the 25% most strongly predicted proteins, 76% were correctly predicted (RI > 50, Fig. 2a: dashed lines).

For the second prediction task (traveler), the reliability index correlated slightly better with performance in the sense that with increasing RI  $Q_2$  increased (albeit not significantly above values of RI = 50, Fig. 2b). For RI > 50, *LocNuclei* predicted for 45% of the proteins and 77% of these were predicted correctly (Fig. 2b: dashed line).

#### Performance of *LocNuclei* confirmed for independent data set of novel proteins

The only method for predicting nuclear sub-structures available during the development of our new method was

*NSort* [12]. Comparing the two methods back-to-back using values published was meaningless due to the differences in data sets. Being no longer available, *NSort* could not be run on new data. Thus, the only meaningful benchmark required training and testing *LocNuclei* on the sets used for *NSort*. Towards this end, we downloaded the *NSort* data set from <http://bioinf.scmb.uq.edu.au:8080/nsort/db> and split into five subsets, trained on four and tested on the remaining one. These sets were rotated five times, so that each protein in the *NSort* set was tested exactly once. The area under the ROC curve (AUC) calculated from the test proteins proxied performance for comparability. For training, we used the same parameters as for the original method. The data set of *NSort* contained proteins from eight sub-nuclear localizations; *LocNuclei-NSort* performed equally well as or even better than *NSort* except for proteins located in the perinucleolar (Table 2). Comparing the original version of *LocNuclei* predicting 13 classes with the version re-trained on eight using the *NSort* data set using common proteins showed that *LocNuclei* performed on average equally well (Additional file 1: Table S3).

#### Spectra of sub-nuclear distributions predicted between organisms differ

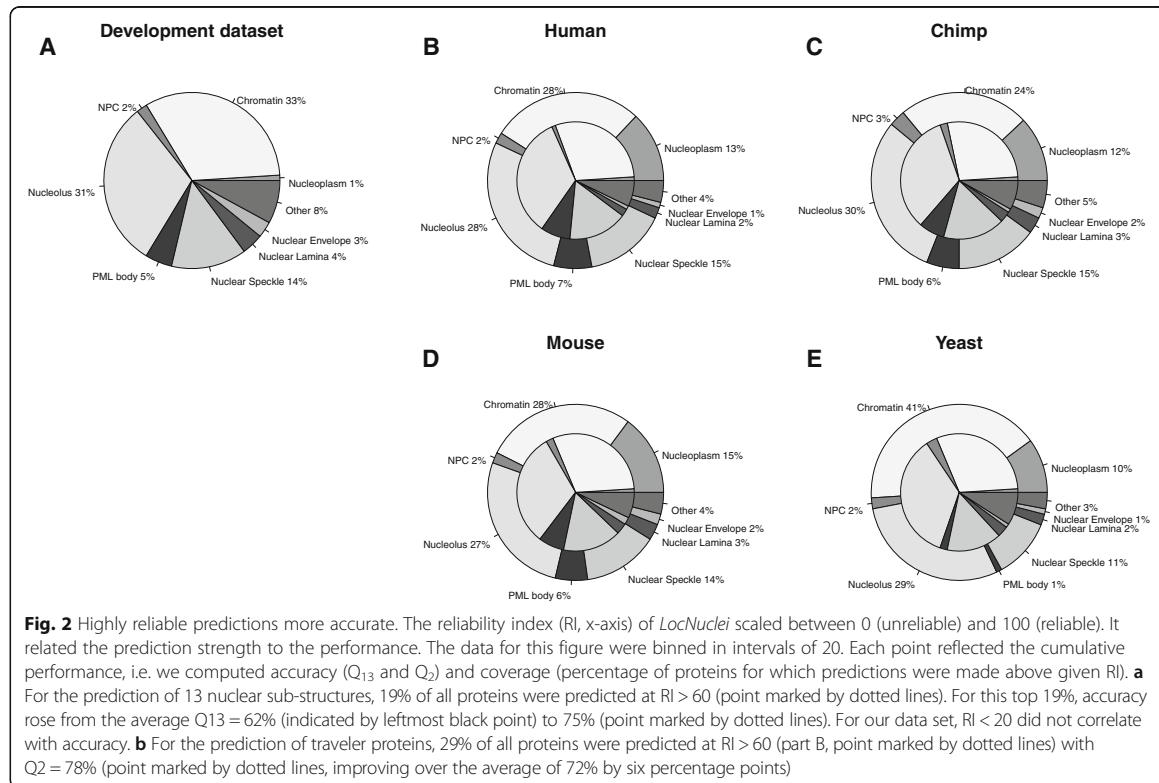
After completing the development, *LocNuclei* was applied to predicting the nuclear sub-structures for entire proteome in *Homo sapiens* (human), *Pan troglodytes* (chimp), *Mus musculus* (mouse) and *Saccharomyces cerevisiae* (baker's yeast). Human, mouse and baker's yeast contribute the most proteins to the development set (341, 961, and 101, respectively). Chimp was only chosen because we expect it to be very close to human. *LocTree3* [13] provided the whole proteome predictions for all four organisms (<https://roslab.org/services/loctree3/proteomes>). All proteins predicted as nuclear and nuclear membrane were used. The resulting datasets contained 6123 proteins for human, 7358 proteins for mouse, 4761 proteins for chimp and 2107 for yeast.

Most machine learning tools have some kind of prediction bias overestimating some classes while underestimating others. To correct for this bias, it was proposed to use the confusion matrix of the tool based on the development set [14]. This leads to an estimation of the overall class distribution that is closer to the truth than the actual predicted distribution. The compositions of the predicted sub-nuclear compartments, i.e. the sub-nuclear spectra were very similar for all organisms for the part only using homology inference (Fig. 3b, c, d and e inner circles). When applying the bias correction to the whole dataset, the composition for human, mouse and chimp remained similar (Fig. 3b, c and d). For human, chimp and mouse, the distributions were also close

**Table 1** LocNuclui confusion matrix for 13 nuclear sub-structures

Observed-> Predicted:	Chromatin	Nucleolus	Nuclear speckle	Nuclear lamina	Nuclear matrix	Nuclear envelope	Cajal body	Nuclear pore complex	Nucleoplasm	Kinetochore	Spindle apparatus	Perinuclear	SUM predicted
Chromatin	<b>506</b>	32	11	7	0	6	1	1	4	6	1	1	579
Nucleolus	49	<b>461</b>	29	11	4	12	6	2	3	2	1	2	585
Nuclear speckle	14	19	<b>153</b>	2	2	4	1	0	2	1	0	0	199
PMI body	12	13	9	<b>38</b>	2	3	2	1	0	1	0	0	82
Nuclear lamina	5	6	3	2	<b>41</b>	3	7	0	0	1	0	0	70
Nuclear matrix	7	9	5	4	2	<b>25</b>	1	0	1	0	0	0	54
Nuclear envelope	4	4	1	0	3	1	<b>34</b>	0	0	1	0	0	54
Cajal body	3	5	2	0	1	1	0	<b>10</b>	1	0	0	0	23
Nuclear pore complex	4	4	1	2	3	0	6	0	<b>15</b>	1	0	0	36
Nucleoplasm	39	38	30	11	7	8	5	10	3	<b>13</b>	2	1	169
Kinetochore	4	5	0	1	1	0	1	2	2	<b>5</b>	0	0	22
Spindle apparatus	32	27	19	8	10	9	7	5	1	1	<b>8</b>	1	129
Perinuclear	0	3	0	0	0	0	1	0	0	0	0	<b>4</b>	8
None	18	27	29	9	4	2	6	1	2	4	3	3	110
% observed	33	31	14	4	4	3	3	2	2	1	1	1	
SUM observed	697	653	292	95	80	74	72	42	34	25	14	13	

The confusion matrix for LocNuclui predictions on the development set with the columns showing the number of observed and the rows the number of predicted proteins (as shown by the sums provided in the last column and the last row). Correct predictions shown on the diagonal are highlighted in bold. The sub-structures are sorted by the number of available annotations (smallest classes at the bottom/right). The dataset was highly imbalanced in the sub-structures, e.g. only 27 proteins were annotated in spindle apparatus and perinuclear and the smallest seven of the 13 classes together accounted for only 11% of all annotated, unique proteins (percentage values are given in the row "% observed"). Performance was largely proportional to the class size, i.e. worse for smaller. Nevertheless, LocNuclui succeeded to predict compartments with only a few samples in the training set, e.g. 8 of the 14 proteins located in the Spindle apparatus are correctly predicted



to the one of the development set (Fig. 3). Only the distribution for yeast differed with a higher number of proteins localized to the chromatin than for the other organisms (Fig. 3e). For all organisms, most proteins were predicted to be either in the chromatin or in the nucleolus (Fig. 3b, c, d and e). Chromatin is a structure built from the interaction with DNA and its role is the maintenance of DNA and the regulation of its transcription. It is known

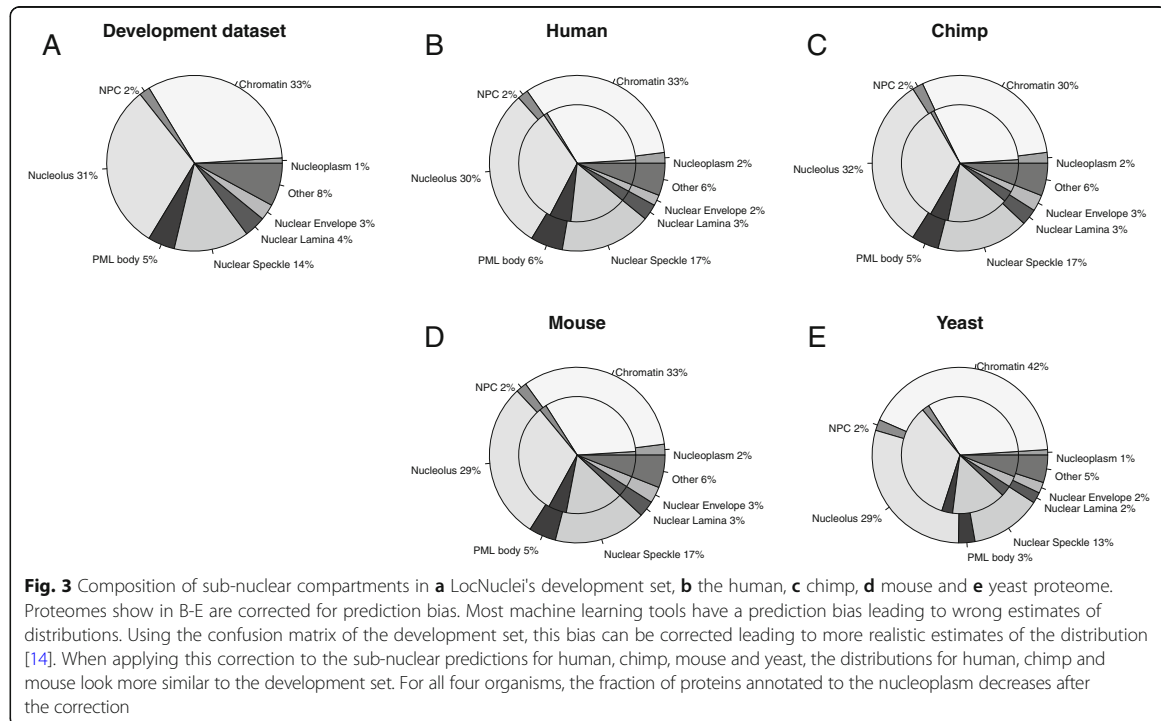
that many proteins that compose the chromatin are exchanged with other sub-nuclear compartments, such as the nucleolus [15, 16].

Using the given distribution, we can also calculate the Euclidean distance between these distributions and use them as a proxy for the distance between the organisms. In our lab, it has been shown that the simple predicted location spectra using all subcellular localizations

**Table 2** Comparison between *LocNuclei* and *NSort*

Sub-nuclear compartment	Number of proteins	AUC <i>NSort</i>	AUC <i>LocNuclei-NSort</i>
Perinucleolar	24	$0.80 \pm 0.05$	$0.73 \pm 0.03$
Cajal body	49	$0.60 \pm 0.03$	$0.62 \pm 0.02$
Nuclear pore complex	51	$0.79 \pm 0.05$	$0.88 \pm 0.02$
Nuclear lamina	77	$0.70 \pm 0.01$	$0.82 \pm 0.01$
PML bodies	91	$0.77 \pm 0.03$	$0.75 \pm 0.01$
Chromatin	323	$0.71 \pm 0.01$	$0.78 \pm 0.01$
Nuclear speckle	403	$0.71 \pm 0.01$	$0.77 \pm 0.01$
Nucleolus	598	$0.60 \pm 0.01$	$0.72 \pm 0.01$
Sum/Mean	1285	$0.71 \pm 0.03$	$0.76 \pm 0.02$

For this comparison, *LocNuclei* was re-trained using the development data of *NSort*, comprising 1285 sequence-unique proteins annotated in eight sub-nuclear localization classes. On proteins from all eight classes, *LocNuclei* performed equally well as or better than *NSort* except for proteins located in the perinucleolar. The overall cross-validated AUC of *LocNuclei* was 0.76 compared to 0.71 for *NSort*. The values for *NSort* were taken from its publication [12]



capture evolutionary aspects of cross-species comparisons [17]. Applying the same concept to the subnuclear location spectra suggested yeast to be most distant from human, chimp and mouse while the distance between human and mouse was smaller than that between human and chimp (Table 3). These differences were statistically significant. If we consider de novo prediction and homology-based inference separately, the relation between organisms based on the distances of the sub-nuclear location spectra did not change for de novo prediction while the location spectra predicted through homology-based inference reflected the expected

**Table 3** Euclidean distance between organisms based on predicted subnuclear location spectra

Overall	Human	Chimp	Mouse	Yeast
Human	0	4.0 ± 0.6	1.8 ± 0.3	9.7 ± 0.7
Chimp	4.0 ± 0.6	0	4.0 ± 0.5	12.7 ± 0.8
Mouse	1.8 ± 0.3	4.0 ± 0.5	0	9.9 ± 0.7
Yeast	9.7 ± 0.7	12.7 ± 0.8	9.9 ± 0.7	0

We calculate the Euclidean distance between predicted subnuclear location spectra and use that distance as proxy to identify evolutionary relationships. As expected, yeast is most distant from the other organisms. However, according to the subnuclear location spectra, human is closer to mouse than to chimp which is opposite what we would expect from known evolutionary relationships. Predicted subnuclear location spectra help in identifying certain aspects of evolution while they cannot capture all evolutionary relations in detail

relation, i.e. human appeared closest to chimp and most distant to yeast (Additional file 1: Table S4).

#### Predictions for homologous protein pairs from different organisms agreed

For a more fine-grained analysis, we also compared predictions for pairs of homologous proteins. For each of the six possible organism pairings, we identified all pairs of homologous proteins in the same way used for LocNuclei (PSI-BLAST at  $E\text{-Values} \leq 10^{-20}$ ). The resulting number of homologous protein pairs mirrored the distance between the predicted subnuclear location spectra (Table 3) for these organisms: For 70% of the human nuclear proteins, we found a homologous protein in mouse (Table 4); the distance between these two organisms based on the location spectra was also the smallest. For yeast, which was most distant to the other organisms, we only found homologous proteins for 20–23% of the proteins (Table 4). For all organism pairs, most protein pairs were predicted by homology-based inference (Table 4, third column). For only a few protein pairs (2% or 6%), one of the proteins was predicted using homology-based inference while the other one is predicted de novo (Table 4, fourth column).

For pairs of homologous proteins, we expect similar predictions from *LocNuclei*. The similarity in predictions between two proteins was measured through the fraction of agreement (Eq. 5; note: for some proteins more than

**Table 4** Homologous protein pairs between four different organisms

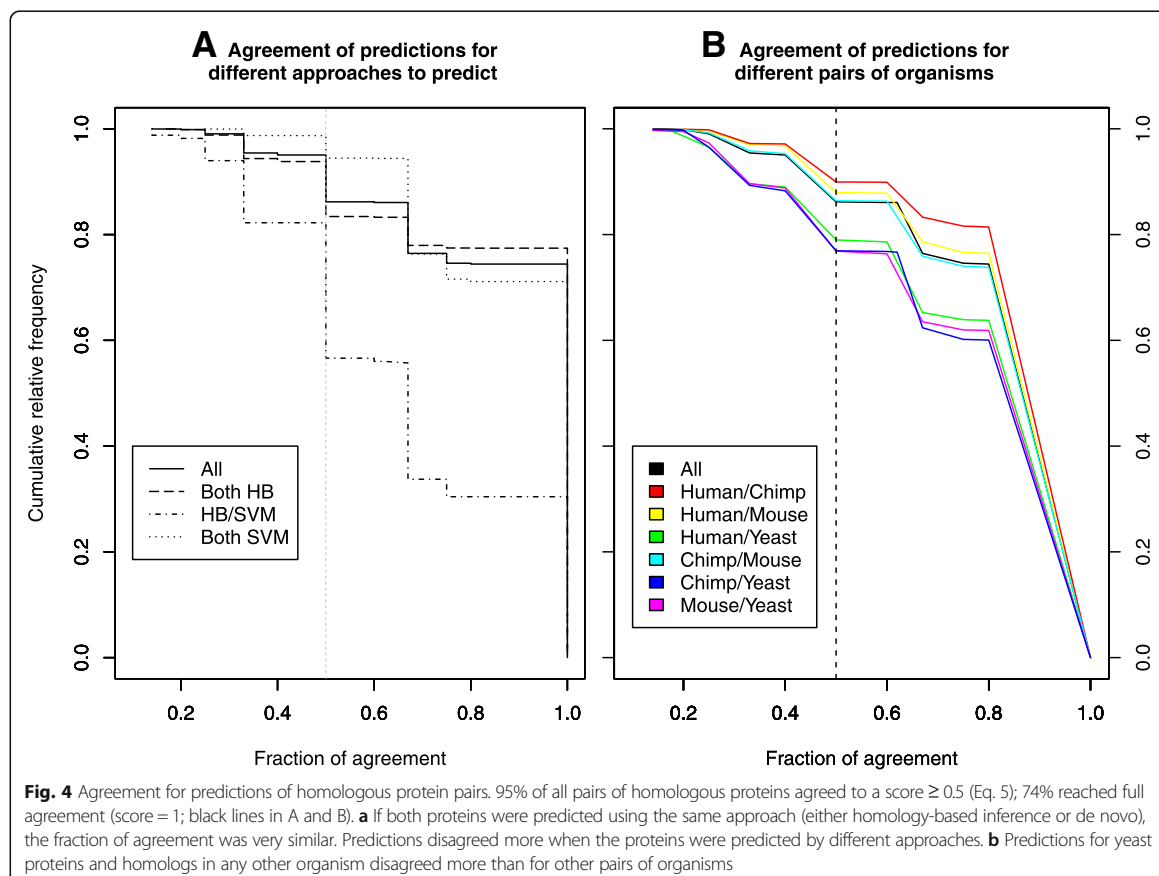
	Overall	Both HB	HB/SVM	SVM	% of proteins (organism1)	% of proteins (organism2)
Human/Chimp	3663	2510 (68%)	58 (2%)	1095 (30%)	60%	62%
Human/Mouse	4316	2609 (60%)	67 (2%)	1640 (38%)	70%	50%
Human/Yeast	809	638 (79%)	50 (6%)	121 (15%)	13%	21%
Chimp/Mouse	4041	2667 (66%)	65 (2%)	1309 (32%)	65%	55%
Chimp/Yeast	776	608 (78%)	42 (6%)	126 (16%)	16%	20%
Mouse/Yeast	973	742 (76%)	50 (5%)	181 (19%)	13%	23%

We identified pairs of homologous proteins between human, chimp, mouse, and yeast. The second column in the table gives the overall numbers of pairs for these two organisms, the next three columns refer to pairs of proteins where both were predicted using homology-based inference, one was predicted with homology-based inference and the other one de novo, and both were predicted de novo. The last two columns give the percentage of proteins in the respective organisms for which a homolog was found

one class was predicted). For almost three fourth (74%) of all protein pairs this agreement was 1, i.e. all classes were predicted identically; while for over 95% of the pairs the agreement scores were  $\geq 0.5$  (Fig. 4a). Surprisingly, the agreement was essentially the same if both proteins were predicted by homology-based inference (HB) and de novo by machine learning (ML, Fig. 4a dashed and dotted lines). Only for mixed protein pairs (one predicted by HB, the other by ML) predictions

agreed much less (Fig. 4a: lowest line with dots and dashes). However, these pairs constituted a small fraction of the overall set of protein pairs (Table 4, fourth column; 2% of all pairs of homologous proteins).

For the set of four model organisms (human, chimp, mouse, and yeast), predictions for homologous proteins agreed most between human and chimp, slightly less between human-mouse or chimp-mouse, and least for human-yeast, mouse-yeast and chimp-yeast (Fig. 4b). As



yeast is the most distant from the other organisms, it is most likely that yeast proteins have different sub-nuclear locations although related in evolution.

Overall, homologous protein pairs, obviously share sub-nuclear locations, otherwise, homology-based inference would not work for our predictions. Nevertheless, for some protein pairs predictions agreed poorly, with the minimal agreement of 0.14 and with 21 protein pairs having agreement  $\leq 0.2$ . Of these 21 proteins, only eight include proteins from yeast; most (15) include a protein from mouse. The agreement score inversely correlated with the number of compartments predicted differentially between the two organisms. For instance, for the four worst predictions (agreement = 0.14), seven compartments were predicted for one organism, but only two for the other. The second protein with two predicted compartments was always the probable E3 ubiquitin-protein ligase HUL4 from yeast (Uniprot identifier P40985) while the other four proteins seemed to belong to the same family. Three proteins were from mouse (genes *Herc6*, *Herc4*, and *Herc3*; Uniprot identifier F2Z461, Q6PAV2, A6H6S0) and the fourth protein was from the chimp gene *HERC6* (Uniprot identifier H2QPV8).

#### GO enrichment of sub-nuclear predictions

Subcellular localization is one aspect of protein function. Thus, the Gene Ontology (GO) [18, 19] reserves one of its three ontologies for function to *Cellular Component* (the other two being *Molecular Function* and *Biological Process*). This does not strictly imply that the *LocNuclei* predictions correlate with function as described by the BFO (Biological Process Ontology of GO). Nevertheless, we hypothesized that there is a correlation.

To address this hypothesis, we performed a GO enrichment analysis of terms from the BFO for the human nuclear proteins predicted by *LocNuclei*. Experimental annotations were available for 4667 of the 5088 (92%) predicted human nuclear proteins. For each of the 13 nuclear sub-structures, we identified the BFO-terms enriched at highest statistical significance ( $p$ -value  $< 0.01$ , Additional file 1: Table S5). Only for 10 of the 13, more than 10 BFO terms reached  $p$ -values  $< 0.01$  (only 2 for peri-nucleolar, only one for nucleoplasm, and none for the spindle apparatus, Additional file 1: Table S5).

The *nucleolus* is involved in ribosomal biogenesis [20] and *LocNuclei* predicted 1856 of the 5088 (36%) human nuclear proteins at the nucleolus. For these proteins, the BFO terms “rRNA processing”, “rRNA metabolic process”, “RNA modification” and “ribonucleoprotein complex biogenesis” were prominent amongst the ten terms with the lowest  $p$ -value (highest significance, Additional file 1: Table S5). *Chromatin* packages DNA and

regulates the access of DNA-binding proteins [21]. For the 1901 proteins predicted to locate to the chromatin (37% of all nuclear proteins) enriched BFO terms included “chromatin organization”, “regulation of RNA biosynthetic process” and “regulation of transcription, DNA-templated” (Additional file 1: Table S5). *Kinetochores* are protein complexes that form when a cell divides; they are located at the centromere and attach the duplicated chromosomes to the mitotic spindle to allow their separation [2]. Only 42 proteins were predicted to locate to the *kinetochores*. For these proteins enriched BFO terms included “cell division”, “chromosome segregation”, and “attachment of spindle microtubules to kinetochores” (Additional file 1: Table S5). Although only few (42) proteins were predicted for kinetochores, the GO enrichment analysis revealed a clear link between the predicted localization and function. Overall, the results of the enrichment analysis for nucleolus, chromatin and kinetochore clearly supported the hypothesis that the predicted sub-nuclear location provided important new evidence for inferring protein function. The results for other compartments such as *nuclear pore complex* and *nuclear envelope* also supported the hypothesis (Additional file 1: Table S5).

For other sub-structures, the signal was less clear. One extreme negative example was the *spindle apparatus* for which not a single BFO term was enriched statistically significantly. The problem might have been that only 13 proteins were predicted in this sub-structure (Additional file 1: Table S5) limiting the power of an enrichment analysis. Another extreme example was the *nucleoplasm* for which 852 proteins (17% of all) were predicted but only one BFO term was statistically significant (namely *Keratinization*, Additional file 1: Table S5). The problem here might have originated from the diversity of this sub-structure that might also result in many prediction mistakes (Table 1). The third sub-structure for which we found fewer than 10 BFO terms enriched at  $P$ -values  $< 10^{-2}$  was the *perinucleolar* (two terms enriched in 33 predicted proteins, Additional file 1: Table S5). For another sub-structure full of a variety of very different proteins [22], the *PML bodies*, our hypothesis was also not supported making it difficult to clearly infer function from enrichment of GO terms.

Performing the same analysis for traveler proteins shows that the most significantly enriched BFO terms for traveler proteins are all associated with transport and localization (Additional file 1: Table S5) suggesting that traveler proteins travel in and out of the nucleus to transport molecules and guide protein localization. Less, but still significantly enriched terms also include involvement in signal transduction (e.g. GO35556 – intracellular signal transduction, GO0023051 – regulation of signaling, or GO0010646 – regulation of cell communication).

### Protein-protein interactions (PPI) related to predicted sub-nuclear localizations

Another way to proxy biological processes is through monitoring physical protein-protein interactions (PPIs<sup>1</sup>) [23]. In analogy to the BFO enrichment analysis, we tested whether or not proteins predicted in nuclear sub-structures by *LocNuclei* contained information about PPIs. More explicitly, we analyzed whether the experimentally annotated PPIs are overrepresented for certain compartments. Overrepresentation is described by the odds ratio that sets the number of observed PPIs between proteins in two compartments (or the same one) into relation with the expected number of PPIs between these compartments. An odds ratio below 1 indicates less PPIs than expected, 1 indicates as many PPIs as expected and values above 1 indicate more PPIs than expected.

Toward this end, the set of human proteins with predicted sub-nuclear localizations were mapped to a dataset of binary, direct interactions from multiple sources used in a different context by our group [24]. In this set, more PPIs than expected are observed within all compartments with especially high values for PPIs between proteins within the kinetochore and the spindle apparatus (Fig. 5) indicating that the formation of compartments and the functionality of proteins performed in these compartments highly relies on interaction between proteins. PPIs between proteins in different compartments are either underrepresented or close to expected except for interactions between proteins in the kinetochore and the spindle apparatus as well as between proteins in the nuclear pore complex, the nuclear lamina and the nuclear envelope (Fig. 5).

Another way to analyze PPIs within nuclear proteins is to compare them to proteins outside the nucleus. To do so, we constructed a PPI network from the human PPI data with proteins being the nodes and an edge drawn between proteins when they interact. The network consists of 15,634 nodes in 569 connected components. Only 142 of these components consist of more than one node. Of the 15,634 proteins in the network, 2037 are solely located in the nucleus, 1283 are traveler proteins travelling between the nucleus and other compartments and 12,314 are proteins located outside the nucleus. On average, nuclear proteins in this network have an average degree of 18 for non-traveler and of 20 for traveler while non-nuclear proteins only have a degree of 10. Considering only the largest connected component with 14,875 does not significantly change the average degree. So, on average nuclear proteins have a higher degree, i.e. they are interacting with more other proteins, than non-nuclear proteins. Also, traveler proteins have a slightly higher degree than non-traveler proteins indicating that they need to

interact with other proteins to move in and out of the nucleus. Also, most of the nuclear proteins (97%) are located in the largest connected component, so they are an important part of the PPI network.

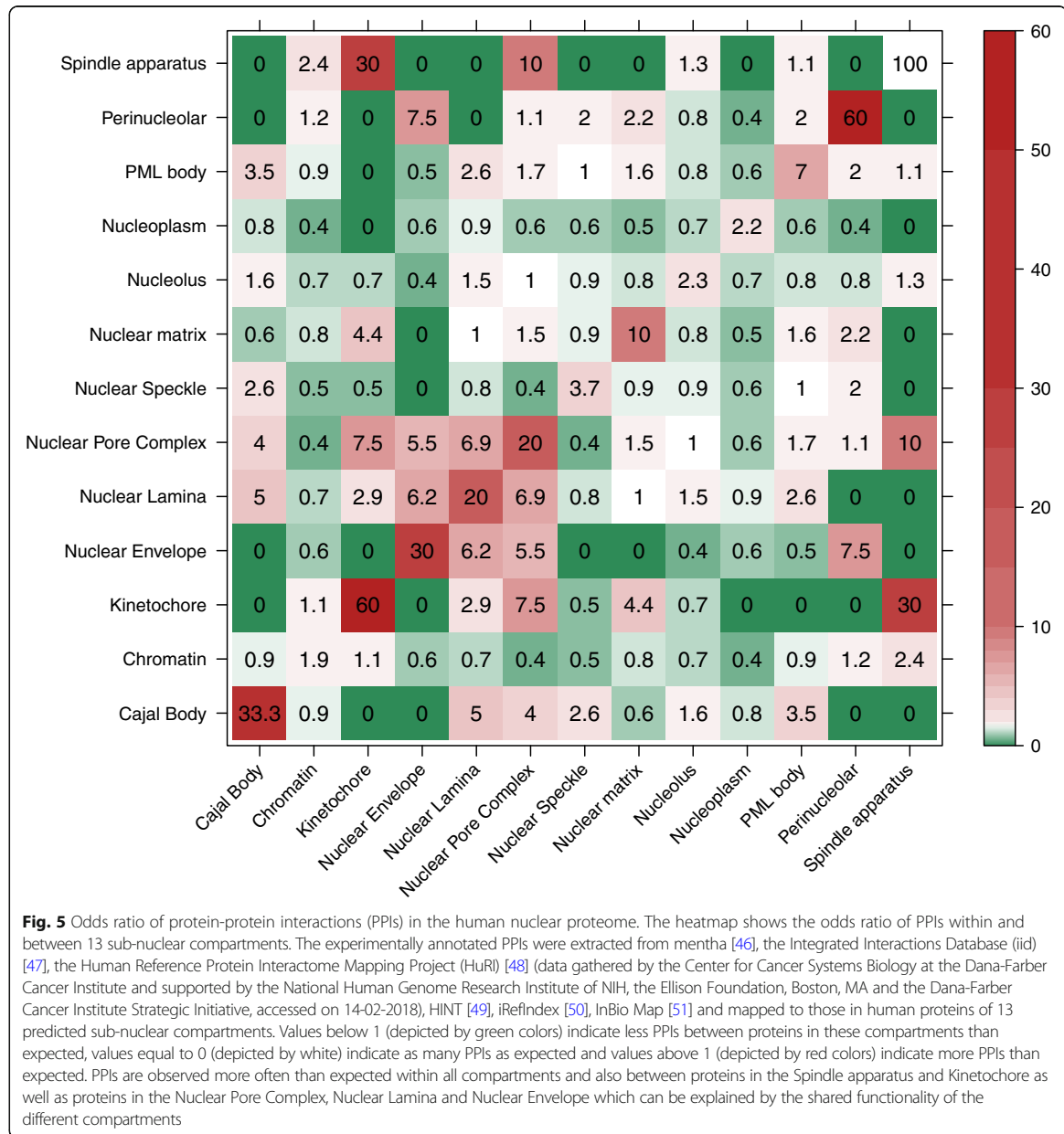
### Discussion

*LocNuclei* predicts sub-nuclear localization at a high accuracy. It combines homology-based inference and de novo prediction to achieve the highest performance. The relatively conservative threshold at which the combination was best (Fig. 1:  $E$ -value  $\leq 10^{-20}$ ) was surprising due to its extremity (e.g. thresholds down to  $E$ -values  $\leq 10^{-3}$  are often used to infer functional similarity), and due to the fact that the performance for lower values was still higher than that of the machine learning (Fig. 1: “Homology (with hit)” vs. SVM). In fact, the curve remained numerically higher down to  $E$ -values  $\leq 10^{-5}$  (straight gray line at  $Q_{13} = 59\%$  vs. dark line in Fig. 1). Given the simple algorithm for the combination of homology-based inference (HB) and machine learning (ML) (if  $\exists$  HB, take HB, else take ML) the combined algorithm could never be worse than its constituents ( $HB \& ML > \max(HB, ML)$ ). Thus, the optimality in  $Q_{13}$  of a conservative threshold suggested that some of the proteins for which HB was available were also predicted above average for ML. Conversely, the cases added at lower thresholds of HB were predicted better by ML than by HB thereby reducing performance by choosing HB over ML (Fig. 1 threshold between  $10^{-20}$  and  $10^{-5}$  all have HB above the ML performance).

Trained on the *NSort* training data, *LocNuclei-NSort* outperforms *NSort*, a predictor for eight sub-nuclear localization classes. On the one hand, it appeared that *LocNuclei* did not gain much from more recent data. On the other hand, it appeared not to have lost from distinguishing more classes.

Spectra for subnuclear compartments calculated from the distribution of the actual predictions show that none of the spectra for human, chimp, mouse, or yeast resembled that for the development set (Additional file 1: Figure S2A) suggesting that the new method was not completely biased by its development set and could discover important aspects in the nuclear proteomes of human, chimp, mouse, and yeast. The biggest difference was for the nucleoplasm for which a much large fraction was predicted in all organisms than in the development set. Since the fraction of proteins predicted in the nucleoplasm decreases when applying a correction (Fig. 3), this suggests a bias in the prediction towards overestimating the number of proteins located to that compartment.

The Euclidean distance between subnuclear location spectra is used to discover evolutionary relationships between organisms. However, the discovered relations between human, chimp, mouse, and yeast are not all as expected (e.g. human closest to mouse instead of



chimp). So, while the comparison of subnuclear location spectra can reveal some insights into the evolutionary relationship between organisms (e.g. human, chimp and mouse closer to each other than to yeast), not all evolutionary aspects can be uncovered completely. Either the subnuclear spectra do not carry enough information to capture evolutionary relationships between these organisms fully or the de novo method makes too many mistakes when predicting subnuclear compartments so that not enough

information is left to reconstruct the evolutionary relationships correctly.

For pairs of homologous proteins, the predicted sub-nuclear compartments often agree. However, there are some pairs where the predictions are very different, especially in terms of number of predicted compartments. We could not find any evidence in public databases or the literature that the difference in predicted compartments for these protein pairs is reasonable. Therefore, the major reason for a disagreement in



predictions between homologous proteins seems to be that *LocNuclei* predicts too many compartments for certain proteins. In fact, this observation is also true for the development set: For 34% of the proteins, the correct number of compartments is predicted, for 61%, at least one compartment more is predicted than annotated, and for 44%, even at least two compartments more are predicted than annotated.

Predicted subnuclear compartments can reveal insights into a protein's functionality. GO enrichment analysis revealed a clear link between the predicted localization and function for many compartments (e.g. nucleolus and kinetochores) while the signal was less clear for other compartments (e.g. nucleoplasm and PML bodies). Overall, the inference of function (as proxied by BFO) from *LocNuclei* predictions worked best for compartments with a stable structure and a clearly defined function.

Monitoring PPIs provides another way to proxy biological processes. As expected, the number of PPIs between proteins within the same compartment is always high while the number of PPIs between proteins in different compartments is much lower. There are only a few exceptions (PPIs between kinetochores and spindle apparatus, and between nuclear pore complex, nuclear lamina, and nuclear envelope occur more often than expected) and these ones can be explained by the shared functionality of the proteins in these compartments. The kinetochore is responsible for attaching the duplicated chromosomes to the spindle apparatus [2] making PPIs between these two compartments inevitable for proper functionality. Nuclear pore complex, nuclear lamina and nuclear envelope are all part of the nuclear membrane suggesting that interactions between proteins of these compartments are needed for stability and proper functionality of the nuclear membrane. As the GO enrichment analysis, the analysis of PPIs between sub-nuclear human proteins showed that the predicted nuclear sub-structures related to the expected functionality of sets of proteins. Therefore, being able to correctly predict subnuclear compartments can help in identifying probable PPIs and functionality.

## Conclusions

*LocNuclei* is an easy-to-use new method predicting sub-nuclear localization; it combined homology-based inference (using PSI-Blast) and de novo prediction (machine learning through an SVM Profile Kernel) to predict the most likely of 13 sub-nuclear compartments in which a nuclear protein functions. It used a similar technology to distinguish between proteins functional only in the nucleus and those also functional in other non-nuclear compartments (dubbed *traveler proteins*). Fivefold stratified

cross-validation yielded  $Q_{13} = 0.62 \pm 0.03$  (one standard deviation) for the sub-structure prediction and  $Q_2 = 0.72 \pm 0.02$  for the traveling proteins. These high values constituted another example for the scientific merit of the Profile Kernel technology [25].

Six thousand one hundred twenty-three proteins of 20,248 of the human proteins (30%) were predicted by *LocTree3* to be located in the nucleus. Here we introduced a set of new methods, referred to as *LocNuclei* that mapped these proteins onto 13 sub-nuclear structures. Most of the nuclear proteins (57%) were predicted to function in the chromatin or the nucleolus. *LocNuclei* also distinguished between traveler and non-traveler proteins. This method suggested only about one third of all nuclear proteins to also function outside the nucleus.

GeneOntology (GO) enrichment analyses focusing on the BFO (Biological Process Ontology) suggested that BFO terms can be inferred from the predicted sub-nuclear locations, at least for stable localizations with a clearly defined role. By cross-referencing the mapped human nuclear proteome protein-protein interaction (PPI) data, an overrepresentation of interactions of proteins within a compartment as well as between proteins located to the kinetochores and the spindle apparatus or proteins located to the nuclear lamina, nuclear envelope, and nuclear pore complex were observed. Like the BFO enrichment, the PPI enrichment suggested that *LocNuclei* predictions might help in annotating protein networks.

## Methods

### Data set for development and evaluation

Experimentally annotated nuclear proteins and annotations for their sub-nuclear localization were combined from six databases: HPRD [26], NMPdb [27], NOPdb [28], NPD [29], NSort/DB [30], and Swiss-Prot [31]. These databases differ in some of their annotation terms for sub-nuclear compartments. We "normalized" these differences through a set of 13 distinct keywords describing the sub-nuclear data set (Additional file 1: Table S6).

Of 12,055 proteins experimentally annotated as nuclear, only 3522 (29%) were associated with one or more nuclear sub-structure. *UniqueProt* [32] generated a non-redundant subset for these by only accepting pairs with  $HVAL < 20$  [33, 34] (implying less than 40% pairwise sequence identity for alignments over 250 residues). At lower HVALs, the data set became too small for meaningful performance estimates. The final sequence-unique sub-nuclear set comprised 1934 proteins (Additional file 1: Table S1).

Four thousand seven hundred twenty-two of the same 12,055 nuclear proteins were also annotated in at least one other non-nuclear sub-cellular compartment (e.g. the mitochondria). The complete set of 12,055 nuclear proteins was redundancy-reduced at  $HVAL < 0$  yielding 1098

sequence-unique proteins, of which 559 (51%) were annotated to exclusively localize to the nucleus, 539 (49%) to be in the nucleus and some other compartment.

The resulting prediction method was trained to differentiate between (i) proteins localized solely to the nucleus and proteins localized to the nucleus and other sub-cellular compartments (traveler proteins), as well as between (ii) proteins of the 13 sub-nuclear localization classes.

### Prediction methods

*LocNuclei* combined homology-based inference and machine learning-based *de novo* predictions in the same way LocTree3 [13] does: if a sequence similar to a protein of experimentally known localization is available that annotation is transferred, if not, the machine learning-based prediction is returned. Stratified fivefold cross-validation was used to determine all parameters and to assess the performance. In a stratified cross-validation, the distribution of classes is approximately equal in every subset [35].

### Homology-based inference

PSI-BLAST [36] alignments are used to transfer annotations by homology. For all proteins of known localization, PSI-BLAST profiles were generated with two iterations and E-value  $\leq 10^{-3}$  using an 80% non-redundant database combining UniProt [37] and PDB [38]. These profiles were then aligned at E-value  $\leq 10^{-20}$  (for prediction of sub-nuclear compartments) or  $\leq 10^{-5}$  (for prediction of traveler proteins) against non-redundant proteins in the development set. For performance estimates, PSI-BLAST self-hits were excluded. The annotation from the hit with the highest pairwise sequence identity of all retrieved alignments was transferred to the query protein.

### De novo prediction

The SVM [39] implementation of LibSVM [40] and the Profile Kernel Function [25, 41] was used to train 13 different SVM classifiers to predict 13 sub-nuclear localizations, where each classifier was trained to discriminate between all the proteins in one particular nuclear sub-structure and all proteins in any of the other 12 nuclear sub-structures. Another profile kernel SVM learned to distinguish between proteins exclusively observed in the nucleus and those observed in the nucleus and other sub-cellular compartments (referred to as *traveler proteins*).

The Profile Kernel algorithm maps each evolutionary profile to a  $20^k$ -dimensional vector of integers. Each dimension represents one *k-mer*, a string of *k* consecutive residues and a particular value gives the number of times this *k-mer* is conserved in an evolutionary profile (multiple sequence alignment). Conservation is calculated as the sum of substitution scores for each residue in the *k-mer* and has to fall below a certain threshold  $\sigma$  [25, 41].  $\sigma$  and *k* are user defined parameters that we

optimized during training. For the SVMs, we focused on optimizing *C*, the penalty parameter of the error term, and *tol*, the tolerance for the stopping criterion. For each Profile Kernel SVM, we optimized these four parameters independently. Also, class weights inversely proportional to class frequencies in the input data were applied for the subnuclear prediction to correct for class imbalance. The traveler dataset was almost balanced; thus, we did not apply class weights for this prediction task. All chosen parameter settings for the 14 different SVMs are listed in Additional file 1: Table S7.

### Reliability index (RI)

Prediction strength correlated with performance (Fig. 2) allowing users to focus on more reliable new predictions through a reliability index (RI) ranging from 0 (weak prediction) to 100 (confident prediction). For the homology-based inference, the percentage pairwise sequence identity (PIDE) from PSI-BLAST was used to define the RI ( $RI = \text{int}(10 \cdot (\text{PIDE} - 20) / 8)$ ). To convert the raw SVM score to a reliability index, this score is normally transferred to a probability using Platt scaling [42]. However, the implementation of Platt scaling in LibSVM [40] failed for our dataset. Typically, SVM scores  $> 0$  should give probability values  $> 0.5$ . For our dataset, this was only observed for the prediction of some sub-structures (classes). For others, Platt scaling transferred the scores to probabilities  $< 0.5$ . Therefore, we had to renormalize the raw SVM scores (Eq. 1) as follows:

$$RI_{\text{svm}} = \text{raw}_{\text{svm}} \cdot \frac{100}{\max(\text{raw}_{\text{svm}})} \quad (1)$$

### Performance evaluation

The performance of *LocNuclei* was assessed through standard measures. For each localization class, every prediction can be classified as either true positive (TP, the sample is predicted and observed in this class), false positive (FP, the sample is predicted in this class, but observed in another), false negative (FN, the sample is predicted not to be in this class but observed in it) and true negative (TN, the sample is predicted and observed in another class). From this classification, the overall accuracy follows:

$$Q(n) = 100 \cdot \frac{\sum_{i=1}^n \text{number of proteins correctly predicted in class } i}{\sum_{i=1}^n \text{total number of proteins observed in class } i} \quad (2)$$

with *n* as the number of localization classes (here: 13). To simplify, this measure calculates the total number of correct predictions divided by the total number of proteins in the test set.

The receiver operating characteristic (ROC) curve and the derived area under the curve (AUC) are combined performance measures connecting true positive rate (TPR, Eq. 3) and false positive rate (FPR, Eq. 4) [43]. The ROC-curve shows FPR versus TPR.

$$TPR = 100 \cdot \frac{TP}{TP + FN} \quad (3)$$

$$FPR = 100 - 100 \cdot \frac{TN}{TN + FP} \quad (4)$$

The curve is often simplified into a single number, the Area Under the Curve (AUC) [43].

#### Comparison of LocNuclei predictions between proteins

*LocNuclei* might predict more than one sub-nuclear compartment for a particular protein. This implies that the comparison of predictions between, e.g. two similar/homologous proteins requires the introduction of additional parameters. Toward this end, we used the fraction of agreement in two predictions  $A^n$  and  $B^m$  defined as follows:

$$\begin{aligned} \text{agree}(A^n, B^m) &= \frac{1}{n} \cdot \sum_{i=1}^n z_i \cdot z_i \\ &= \begin{cases} 1, & \text{if } a_i \in (b_1, \dots, b_m) \text{ and } n \geq m \text{ w.l.o.g.} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

where  $A$  and  $B$  are two proteins and  $n$  and  $m$  are the number of predicted compartments for  $A$  and  $B$ , respectively. In the limit of a single prediction per protein, this agreement is identical to the percentage of correct predictions; in the limit of predicting all sub-compartments for one protein, the value falls below random (1/13 which is lower than random given the difference in the size distribution of the 13 compartments).

#### GO enrichment analysis

Gene Ontology (GO) [18, 19] provides a controlled vocabulary (GO terms) of annotated functions for a protein. It consists of three separate ontologies: “Biological Process”, “Molecular Function” and “Cellular Compartment”. To analyze whether certain GO terms are statistically enriched for proteins annotated in a particular nuclear sub-structure, we used the webserver *GORilla* (<http://cbl-gorilla.cs.technion.ac.il/>) [44]. *GORilla* analyzes the enrichment of a certain set of proteins through a hypergeometric distribution. It compares the number of known experimental annotations of a GO term in all proteins within a compartment (positive class) and those in all proteins not in the compartment (negative class). The resulting p-value gives the probability to observe the given annotations under the assumption that the annotations for proteins from both classes do not differ. A

small p-value indicates that this assumption is not true and that the corresponding GO term is overrepresented in the positive class. *GORilla* also offers correction for multiple testing by giving a p-value adjusted using the Benjamini-Hochberg method [45]. We only considered the adjusted p-value when analyzing the significance of results. We considered all terms with p-values < 0.01 as significantly enriched in the positive. The GO enrichment analysis was carried out exclusively for GO ontology “biological process”.

#### Protein-protein interactions (PPI) for nuclear proteins

To analyze the map between nuclear sub-structures and protein-protein interactions (PPIs) in human proteins, we merged a dataset containing information from six original resources, namely: (1) mentha [46], (2) the Integrated Interactions Database (iid) [47], (3) the Human Reference Protein Interactome Mapping Project (HuRI) [48] (data gathered by the Center for Cancer Systems Biology at the Dana-Farber Cancer Institute and supported by the National Human Genome Research Institute of NIH, the Ellison Foundation, Boston, MA and the Dana-Farber Cancer Institute Strategic Initiative, accessed on 14-02-2018), (4) HINT [49], (5) iRefIndex [50], and from (6) InBio Map [51]. For each database, only binary, direct interactions were considered (often also referred to as transient physical interactions), i.e. we excluded associations. Furthermore, only interactions determined by an experiment and validated by a yeast two-hybrid (Y2H) experiment or interactions supported by two independent Pubmed IDs were considered.

To analyze whether proteins between or within a compartment interact more often than we would expect, we calculate an odds ratio for an interaction to happen between compartment  $i$  and  $j$  (Eq. 5).

$$\text{odds}(PPI_{ij}) = \frac{\text{num}_{\text{obs}}(PPI_{ij})}{\text{num}_{\text{exp}}(PPI_{ij})} \quad (6)$$

where  $\text{num}_{\text{exp}}(PPI_{ij})$  is the number of expected PPIs between proteins in these compartments and is calculated as

$$\text{num}_{\text{exp}}(PPI_{ij}) = \frac{\text{num}_{\text{pos}}(PPI_{ij})}{\sum_{ij} \text{num}_{\text{pos}}(PPI_{ij})} \cdot \text{num}_{\text{obs}}(PPI) \quad (7)$$

Where  $\text{num}_{\text{pos}}(PPI_{ij})$  is the number of possible PPIs between proteins in compartment  $i$  and  $j$  in the whole PPI dataset and  $\text{num}_{\text{obs}}(PPI)$  is the overall number of observed PPIs in our data set.

*NSort* [12] is a framework with eight Bayesian Network-based classifiers that predict protein sub-nuclear localization in eight classes (nucleolus, perinucleolar region, PML bodies, nuclear speckle, Cajal bodies, chromatin and nuclear pore complexes). Each classifier

operates from biological features including protein sequence, protein interactions, domain and post-translational modification. Each prediction of *NSort* can be traced back to the feature contributing most to the result. As *NSort* is the only method available to accomplish some of the objectives aimed at by *LocNuclei*, we compared the performance of *LocNuclei* to that of *NSort*.

#### Availability

*LocNuclei* is a Python project and is available on GitHub: <https://github.com/Rostlab/LocNuclei>. The datasets of sub-nuclear and traveler proteins used for development as well as sub-nuclear and traveler predictions for all proteins from the development set are also available. More detailed information on how to run *LocNuclei* is given in the repository.

#### Endnotes

<sup>1</sup>Operationally, we defined transient, physical protein-protein interactions (PPI) as cases of two different proteins that come so close in space that they “bind” (physical interaction as opposed to association; closest  $C\text{-}\alpha\leq 6\text{ \AA}$ ) and that this binding is shorter than the “life”-time of either of the two (transient). This simple definition implies in particular that (i) PPIs are formed only between different proteins, (ii) no transitivity:  $PPI(A,B) \cap PPI(B,C) \nrightarrow PPI(A,C)$ , and (iii) no molecular machines: just as most associated proteins do not bind, most members of the same molecular machine, or large physical complex do not bind.

#### Additional file

**Additional file 1: Figure S1.** Effect of E-value thresholds on combined prediction of traveler proteins. **Figure S2.** Composition of sub-nuclear compartments in the human, chimp, mouse and yeast proteome and *LocNuclei*'s development set. **Table S1.** Composition of the sub-nuclear development set for *LocNuclei*. **Table S2.** *LocNuclei* confusion matrix for homology-based inference and machine learning prediction. **Table S3.** Comparison between *LocNuclei* and *LocNuclei-NSort*. **Table S4.** Euclidean distance between organisms based on subnuclear location spectra predicted with SVM Profile Kernel (de novo) or homology-based inference. **Table S5.** Top ten statistically enriched GO terms for each sub-nuclear compartment. **Table S6.** Normalization of sub-nuclear localization terms. **Table S7.** Chosen hyperparameters for the 14 different SVM Profile Kernels. (DOCX 43170 kb)

#### Abbreviations

AUC: Area under the ROC curve; FPR: False positive rate; GO: Gene Ontology; HB: Homology-based inference; HuRi: Human Reference Protein Interactome Mapping Project; iid: Integrated Interactions Database; ML: Machine learning; NES: Nuclear export signals; NLS: Nuclear localization signals; NPC: Nuclear pore complex; PPI: Protein-protein interaction; ROC: Receiver operating characteristic; SVM: Support vector machine; TPR: True positive rate

#### Acknowledgements

Thanks to Tim Karl (TUM) for invaluable help with hardware and software; to Inga Weise (TUM) for support with many other aspects of this work; and to Michael Heinzinger for providing us with the dataset for human PPIs. Thanks also to the anonymous reviewer who helped substantially to improve the

paper. Last, not least, thanks to Marc Vidal (Harvard, Dana Faber, Cambridge), Igor Jurisica (Princess Margret Cancer Centre, Toronto), Miguel Andrade (HIPPIE, Mainz University), and their colleagues and crews for maintaining excellent databases and to all experimentalists who enabled this analysis by making their data publicly available.

#### Funding

This work was supported by the Bavarian Ministry for Education through funding to the TUM paying for the positions of the authors and publication charges. The funding body did not play any role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

#### Availability of data and materials

The datasets generated and analyzed in this article are available in the *LocNuclei* repository, <https://github.com/Rostlab/LocNuclei>.

#### Authors' contributions

SS collected the data and implemented the first version of *LocNuclei* together with TG and MB. ML refined the existing implementation, performed GO enrichment analysis and analysis of protein-protein interactions for nuclear proteins and did the major part of writing the manuscript. TG, MB and BR supervised the work over the entire time and proofread the manuscript. All authors read and approved the final manuscript.

#### Ethics approval and consent to participate

Not applicable.

#### Consent for publication

Not applicable.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

#### Author details

<sup>1</sup>Department of Informatics, Bioinformatics & Computational Biology - i12, TUM (Technical University of Munich), Boltzmannstr. 3, 85748 Garching/Munich, Germany. <sup>2</sup>School of Chemistry and Molecular Biosciences, UQ (University of Queensland), Cooper Rd, Brisbane City, QLD 4072, Australia. <sup>3</sup>Institute for Advanced Study (TUM-IAS), Lichtenbergstr 2a, 85748 Garching/Munich, Germany. <sup>4</sup>TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany. <sup>5</sup>Department of Biochemistry and Molecular Biophysics & New York Consortium on Membrane Protein Structure (NYCOMPS), Columbia University, 701 West, 168th Street, New York, NY 10032, USA.

Received: 27 September 2018 Accepted: 2 April 2019

Published online: 23 April 2019

#### References

- Erhardt M, Adamska I, Franco OL. Plant nuclear proteomics—inside the cell maestro. *FEBS J.* 2010;277(16):3295–307.
- Alberts B, Johnson A, Lewis JH, Morgan D. *Molecular biology of the cell.* New York, NY: Garland Science; 2015.
- Sampathkumar P, Kim SJ, Upla P, Rice WJ, Phillips J, Timney BL, Pieper U, Bonanno JB, Fernandez-Martinez J, Hakhverdyan Z, et al. Structure, dynamics, evolution, and function of a major scaffold component in the nuclear pore complex. *Structure.* 2013;21(4):560–71.
- Freitas N, Cunha C. Mechanisms and signals for the nuclear import of proteins. *Curr Genomics.* 2009;10(8):550–7.
- Cokol M, Nair R, Rost B. Finding nuclear localisation signals. *EMBO Rep.* 2000;1:411–5.
- Bernhofer M, Goldberg T, Wolf S, Ahmed M, Zaugg J, Boden M, Rost B. NLSdb - major update for database of nuclear localization signals and nuclear export signals. *Nucleic Acids Res.* 2018;46(D1):D503–8.
- Marfori M, Mynott A, Ellis JJ, Mehdi AM, Saunders NF, Curmi PM, Forwood JK, Boden M, Kobe B. Molecular basis for specificity of nuclear import and prediction of nuclear localization. *Biochim Biophys Acta.* 2011;1813(9):1562–77.

8. Carmo-Fonseca M. The contribution of nuclear compartmentalization to gene regulation. *Cell*. 2002;108(4):513–21.
9. Chubb JR, Bickmore WA. Considering nuclear compartmentalization in the light of nuclear dynamics. *Cell*. 2003;112(4):403–6.
10. Lohrum MA, Ashcroft M, Kubbutat MH, Vousden KH. Identification of a cryptic nucleolar-localization signal in MDM2. *Nat Cell Biol*. 2000;2(3):179–81.
11. Eilbracht J, Schmidt-Zachmann MS. Identification of a sequence element directing a protein to nuclear speckles. *Proc Natl Acad Sci U S A*. 2001;98(7):3849–54.
12. Bauer DC, Willadsen K, Buske FA, Le Cao KA, Bailey TL, Dellaire G, Boden M. Sorting the nuclear proteome. *Bioinformatics*. 2011;27(13):17–14.
13. Goldberg T, Hecht M, Hamp T, Karl T, Yachdav G, Ahmed N, Altermann U, Angerer P, Ansorge S, Balasz K, et al. LocTree3 prediction of localization. *Nucleic Acids Res*. 2014;42(Web Server issue):W350–5.
14. Marot-Lassauzaie V, Bernhofer M, Rost B. Correcting mistakes in predicting distributions. *Bioinformatics*. 2018;34(19):3385–3386.
15. McKeown PC, Shaw PJ. Chromatin: linking structure and function in the nucleolus. *Chromosoma*. 2009;118(1):11–23.
16. Bickmore WA, Sutherland HGE. Addressing protein localization within the nucleus. *EMBO J*. 2002;21(6):1248–54.
17. Marot-Lassauzaie V. Cross-species comparison of protein subcellular localization annotation. *Bachelor Thesis*. Munich: Technical University of Munich; 2017.
18. The Gene Ontology C. Expansion of the Gene Ontology knowledgebase and resources. *Nucleic Acids Res*. 2017;45(D1):D331–8.
19. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, et al. Gene ontology: tool for the unification of biology. The Gene Ontology consortium. *Nat Genet*. 2000;25(1):25–9.
20. Andersen JS, Lam YW, Leung AK, Ong SE, Lyon CE, Lamond AI, Mann M. Nucleolar proteome dynamics. *Nature*. 2005;433(7021):77–83.
21. Comings DE. The structure and function of chromatin. *Adv Hum Genet*. 1972;3:237–431.
22. Lallemand-Breitenbach V. PML nuclear bodies. *Cold Spring Harb Perspect Biol*. 2010;2(5):a000661.
23. Ofran Y, Rost B. Protein-protein interaction hotspots carved into sequences. *PLoS Comput Biol*. 2007;3(7):e119.
24. Heinzinger M. Predicting protein contacts and interactions using co-evolution and deep learning. Munich: Technical University of Munich; 2017.
25. Hamp T, Goldberg T, Rost B. Accelerating the original profile kernel. *PLoS One*. 2013;8(6):e68459.
26. Keshava Prasad TS, Goel R, Kandasamy K, Keerthikumar S, Kumar S, Mathivanan S, Telikicherla D, Raju R, Shafreen B, Venugopal A, et al. Human protein reference database—2009 update. *Nucleic Acids Res*. 2009;37(Database):D767–72.
27. Mika S, Rost B. NMPdb: database of nuclear matrix proteins. *Nucleic Acids Res*. 2005;33(Database issue):D160–3.
28. Leung AK, Trinkle-Mulcahy L, Lam YW, Andersen JS, Mann M, Lamond AI. NOPdb: nucleolar proteome database. *Nucleic Acids Res*. 2006;34(Database issue):D218–20.
29. Dellaire G, Farrall R, Bickmore WA. The nuclear protein database (NPD): sub-nuclear localisation and functional annotation of the nuclear proteome. *Nucleic Acids Res*. 2003;31(1):328–30.
30. Willadsen K, Mohamad N, Boden M. NSort/DB: an intranuclear compartment protein database. *Genomics Proteomics Bioinformatics*. 2012;10(4):226–9.
31. Bairoch A, Apweiler R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*. 2000;28(1):45–8.
32. Mika S, Rost B. UniqueProt: creating representative protein sequence sets. *Nucleic Acids Res*. 2003;31(13):3789–91.
33. Sander C, Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*. 1991;9(1):56–68.
34. Rost B. Twilight zone of protein sequence alignments. *Protein Eng*. 1999;12(2):85–94.
35. Olson DL, Delen D. Advanced data mining techniques, 1 edn. eBook: Springer-Verlag Berlin Heidelberg; 2008.
36. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389–402.
37. Bairoch A, Apweiler R. The SWISS-PROT protein sequence data bank and its supplement TrEMBL. *Nucleic Acids Res*. 1997;25(1):31–6.
38. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The Protein Data Bank. *Nucleic Acids Res*. 2000;28(1):235–42.
39. Cortes C, Vapnik VN. Support vector networks. *Mach Learn*. 1995;20:273–97.
40. Chang C-C, Lin C-J. LIBSVM: A library for Support Vector Machines. *ACM Trans Intell Syst Technol*. 2011;2(3):27:21–7.
41. Kuang R, Wang K, Wang K, Siddiqi M, Freund Y, Leslie C. Profile-based string kernels for remote homology detection and motif extraction. *Journal of bioinformatics and computational biology*. 2005;3(3):527–550.
42. Platt JC. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in large margin classifiers*: MIT Press; 1999;10(3):61–74.
43. Gribskov M, Robinson NL. Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput Chem*. 1996;20(1):25–33.
44. Eden E, Navon R, Steinfeld I, Lipsion D, Yakhini Z. GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinf*. 2009;10:48.
45. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B Methodol*. 1995;57(1):289–300.
46. Calderone A, Castagnoli L, Cesareni G. Mentha: a resource for browsing integrated protein-interaction networks. *Nat Methods*. 2013;10(8):690–1.
47. Kotlyar M, Pastrello C, Sheahan N, Jurisica I. Integrated interactions database: tissue-specific view of the human and model organism interactomes. *Nucleic Acids Res*. 2016;44(D1):D536–41.
48. Rolland T, Tasan M, Charlotiaux B, Pevzner SJ, Zhong Q, Sahni N, Yi S, Lemmens I, Fontanillo C, Mosca R, et al. A proteome-scale map of the human interactome network. *Cell*. 2014;159(5):1212–26.
49. Das J, Yu H. HINT: high-quality protein interactomes and their applications in understanding human disease. *BMC Syst Biol*. 2012;6:92.
50. Razick S, Magklaras G, Donaldson IM. iRefIndex: a consolidated protein interaction database with provenance. *BMC Bioinf*. 2008;9:405.
51. Li T, Wernersson R, Hansen RB, Horn H, Mercer J, Slodkowitz G, Workman CT, Rigina O, Rapacki K, Staerfeldt HH, et al. A scored human protein-protein interaction network to catalyze genomic interpretation. *Nat Methods*. 2017;14(1):61–4.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more [biomedcentral.com/submissions](https://biomedcentral.com/submissions)



### **5.3. Correction to: Littmann, Goldberg *et al.*, BMC Bioinformatics (2019)**

An incorrect figure has been published as Fig. 2 in the original publication. A correction with the correct Fig. 2 has been published and is shown below.

CORRECTION

Open Access

# Correction to: Detailed prediction of protein sub-nuclear localization



Maria Littmann<sup>1†</sup>, Tatyana Goldberg<sup>1†</sup>, Sebastian Seitz<sup>1</sup>, Mikael Bodén<sup>2</sup> and Burkhard Rost<sup>1,3,4,5</sup>

**Correction to: BMC Bioinformatics (2019) 20:205**  
<https://doi.org/10.1186/s12859-019-2790-9>

Following publication of the original article [1], the author reported that an incorrect figure has been published as Fig. 2. The correct Fig. 2 is shown below.

The publisher apologizes to the authors and readers for the inconvenience.

#### Author details

<sup>1</sup>Department of Informatics, Bioinformatics & Computational Biology - i12, TUM (Technical University of Munich), Boltzmannstr. 3, 85748 Garching/Munich, Germany. <sup>2</sup>School of Chemistry and Molecular Biosciences, UQ (University of Queensland), Cooper Rd, Brisbane City QLD 4072, Australia. <sup>3</sup>Institute for Advanced Study (TUM-IAS), Lichtenbergstr 2a, 85748 Garching/Munich, Germany. <sup>4</sup>TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany. <sup>5</sup>Department of Biochemistry and Molecular Biophysics & New York Consortium on Membrane Protein Structure (NYCOMP), Columbia University, 701 West, 168th Street, New York, NY 10032, USA.

Published online: 20 December 2019

#### Reference

1. Littmann M, et al. Detailed prediction of protein sub-nuclear localization. *BMC Bioinformatics*. 2019;20:205. <https://doi.org/10.1186/s12859-019-2790-9>.

The original article can be found online at <https://doi.org/10.1186/s12859-019-2790-9>

\* Correspondence: [littmann@rostlab.org](mailto:littmann@rostlab.org); [assistant@rostlab.org](mailto:assistant@rostlab.org)

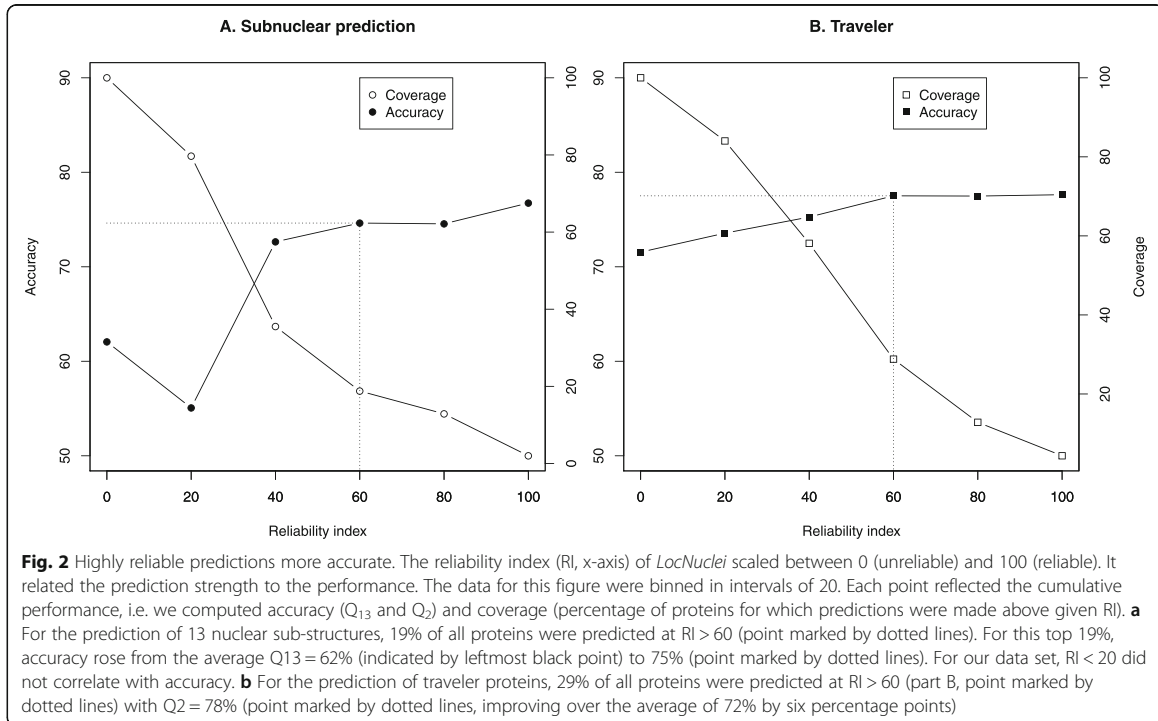
<sup>†</sup>Maria Littmann and Tatyana Goldberg contributed equally to this work.

<sup>1</sup>Department of Informatics, Bioinformatics & Computational Biology - i12, TUM (Technical University of Munich), Boltzmannstr. 3, 85748 Garching/Munich, Germany

Full list of author information is available at the end of the article



© The Author(s). 2019 **Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated.





**5.4. Supplementary Material: Littmann, Goldberg et al., BMC Bioinformatics (2019)**

**Supporting online material  
for:  
Detailed prediction of protein sub-nuclear  
localization**

**Maria Littmann, Tatyana Goldberg, Sebastian Seitz, Mikael Bodén  
& Burkhard Rost**

**Table of Contents for Supporting Online Material**

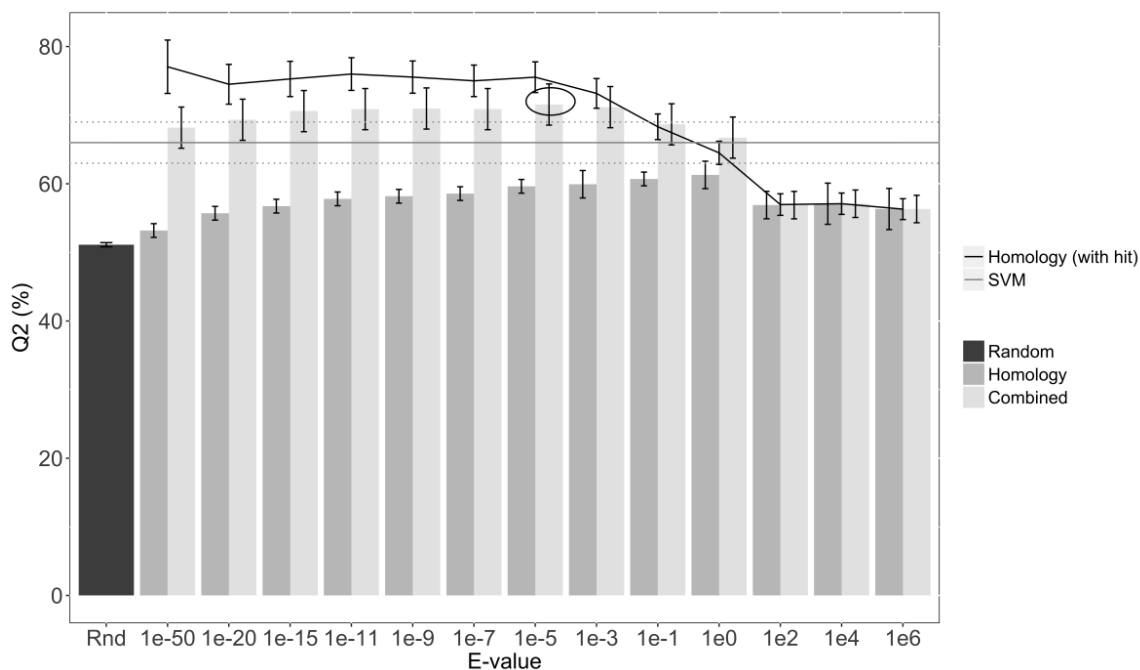
1. Effect of E-value thresholds on combined prediction of traveler proteins
2. Composition of sub-nuclear compartments for different organisms and LocNuclei's development set
3. Normalization of sub-nuclear localization terms
4. Composition of the sub-nuclear development set
5. Comparison between LocNuclei and LocNuclei-NSort
6. Euclidean distance between organisms based on subnuclear location spectra
7. Top ten statistically enriched GO terms

**Short description of Supporting Online Material**

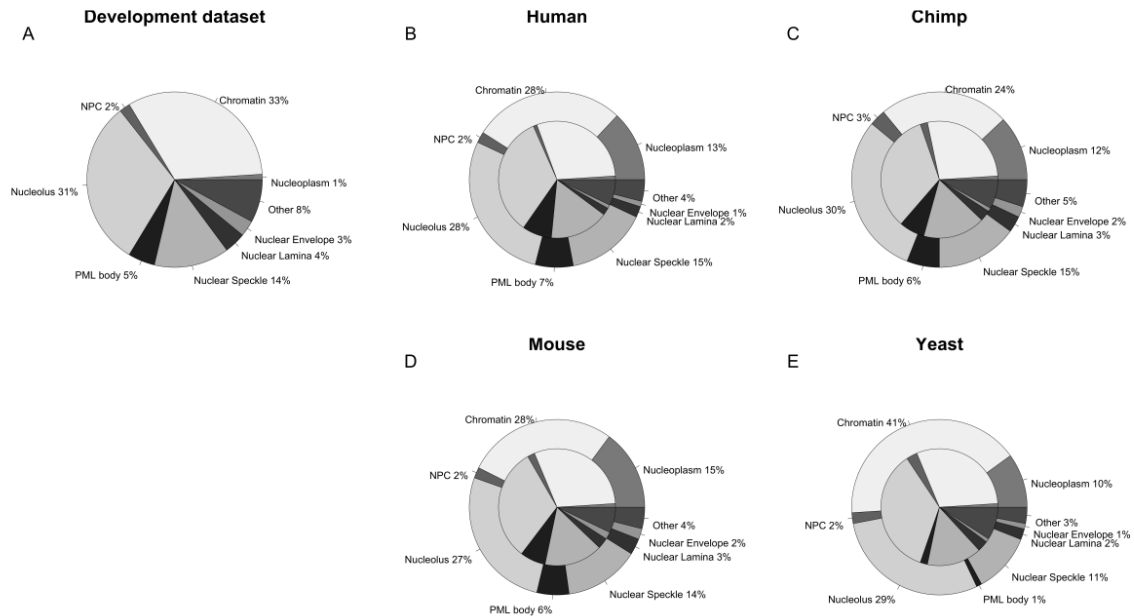
1. Figure showing Q2 for different E-value thresholds for the homology-based component of LocNuclei for traveler predictions
2. Figure showing the composition of sub-nuclear compartments (without correction for prediction bias) for different organisms and LocNuclei's development set
3. Table showing the normalization of sub-nuclear terms
4. Table showing the composition of the sub-nuclear development set for LocNuclei across 13 sub-nuclear classes
5. Table showing AUC values for LocNuclei and LocNuclei-NSort
6. Table showing the Euclidean distance between organisms based on predicted subnuclear location spectra
7. Table showing the top ten statistically enriched GO terms for every sub-nuclear compartment from the GO enrichment analysis

## Material

**Fig. S1:**



**Fig. S1: Effect of E-value thresholds on combined prediction of traveler proteins.** The accuracy  $Q_2$  for classifying nuclear proteins as travelers or not using homology-based inference with PSI-BLAST (based on 12,055 experimentally annotated nuclear proteins) varies at different E-value thresholds (darker grey bars on the left). For proteins for which a protein with experimentally known nuclear sub-structure annotation is more sequence similar than the threshold, performance depends on the threshold (black line). Homology based inference reaches the highest accuracy of  $Q_2 = 77\%$  at the stringent E-value  $\leq 10^{-50}$ . However, when evaluated on the entire test set (*i.e.* also on proteins for which no homolog is available), the performance drops significantly to  $Q_2 = 53\%$  compared to a random prediction of  $Q_2 = 49\%$ . The performance of the SVM on the same set, however, reaches  $Q_2 = 66\%$  (the performance is marked by grey lines). The lighter grey bars mark the combination of homology inference and machine learning. The optimal threshold for the combination was E-value  $\leq 10^{-5}$ . One standard error marked on each bar and on the black line and through the dotted lines for ML.

**Fig. S2:**

**Fig. S2: Composition of sub-nuclear compartments in the human, chimp, mouse and yeast proteome and LocNuclei's development set.** A. Composition of LocNuclei's development set (assembled from nuclear proteins of various organisms) B. Composition of 5,088 human nuclear C. Composition of 4,067 nuclear proteins from chimp. D. Composition of 6,041 nuclear proteins from mouse E. Composition of 1,790 nuclear proteins from yeast. All nuclear proteome sets were obtained by taking nuclear and nuclear membrane proteins from LocTree3 [29] whole proteome prediction. Differences in the number of nuclear proteins identified from LocTree3 and numbers given here originate from the fact that LocNuclei was not able to predict any sub-nuclear compartment for some of the proteins. For each organism, the outer circle shows the composition of the whole dataset while the inner circle only shows the composition of the proteins predicted using homology inference. Human, mouse and chimp are very similar because they also share a large amount of homologous proteins while yeast with a more different proteome in general also shows a different composition of the nuclear proteome.

**Table S1: Composition of the sub-nuclear development set for LocNuclei**

	Chromatin (697)	Nucleolus (653)	Nuclear speckle (292)	PML body (95)	Nuclear lamina (80)	Nuclear matrix (74)	Nuclear envelope (72)	Cajal body (42)	Nuclear pore complex (35)	Nucleoplasm (29)	Kinetochore (25)	Spindle apparatus (14)	Perinucleolar (13)
Chromatin (697)	<b>584</b>												
Nucleolus (653)	68	<b>483</b>											
Nuclear speckle (292)	22	79	<b>176</b>										
PML body (95)	23	18	9	<b>49</b>									
Nuclear lamina (80)	5	8	2	3	<b>51</b>								
Nuclear matrix (74)	4	3	3	2	1	<b>63</b>							
Nuclear envelope (72)	2	0	0	1	3	1	<b>63</b>						
Cajal body (42)	3	15	14	4	2	0	0	<b>15</b>					
Nuclear pore complex (35)	5	6	3	2	17	0	0	2	<b>12</b>				
Nucleoplasm (29)	3	8	3	1	0	2	2	0	0	<b>13</b>			
Kinetochore (25)	3	4	0	0	1	0	0	0	2	1	<b>15</b>		
Spindle apparatus (14)	2	1	0	1	4	1	1	0	1	0	3	<b>6</b>	
Perinucleolar (13)	3	4	6	2	1	0	0	1	0	0	3	0	<b>2</b>

The table displays numbers of sequence-unique proteins (HVAL [7, 8]  $\leq 20$ ) across 13 sub-nuclear localization classes in the development set of LocNuclei. Only proteins with experimental annotations extracted from HPRD [1], NMPdb [2], NOPdb [3], NPD [4], NSort/DB [5] and Swiss-Prot [6] are used. The numbers of unique sequences per localization are given in parentheses. The numbers on the diagonal describe sequences with the annotation of one localization class (e.g. 584 sequences in the set were annotated to localize at the chromatin only). Other numbers are annotations of two sub-nuclear compartments. Note that some sequences had annotations of more than two compartments.

**Table S2: LocNuclei confusion matrix for homology-based inference and machine learning prediction**

Observed:-> Predicted:	<i>Chromatin</i>	<i>Nucleolus</i>	<i>Nuclear speckle</i>	<i>PML body</i>	<i>Nuclear lamina</i>	<i>Nuclear matrix</i>	<i>Nuclear envelope</i>	<i>Cajal body</i>	<i>Nuclear pore complex</i>	<i>Nucleoplasm</i>	<i>Kinetochore</i>	<i>Spindle apparatus</i>	<i>Perinucleolar</i>	<i>SUM predicted</i>
<i>Chromatin</i>	<b>208</b>	19	4	4	0	3	1	0	0	3	4	0	1	247
	<b>298</b>	13	7	3	0	3	2	1	1	1	2	1	0	332
<i>Nucleolus</i>	28	<b>212</b>	17	8	0	5	0	2	1	2	1	0	1	277
	21	<b>249</b>	12	3	4	7	3	4	1	1	1	1	1	308
<i>Nuclear speckle</i>	4	11	<b>66</b>	1	0	3	0	0	0	1	1	0	0	87
	10	8	<b>87</b>	1	2	1	1	1	0	1	0	0	0	112
<i>PML body</i>	1	2	3	<b>21</b>	0	0	0	0	1	0	1	0	0	29
	11	11	6	<b>17</b>	2	3	2	1	0	0	0	0	0	53
<i>Nuclear lamina</i>	1	1	2	1	<b>12</b>	2	5	0	1	0	1	0	0	26
	4	5	1	1	<b>29</b>	1	2	0	1	0	0	0	0	44
<i>Nuclear matrix</i>	3	6	4	3	1	<b>16</b>	0	0	0	0	0	0	0	33
	4	3	1	1	1	<b>9</b>	1	0	0	1	0	0	0	21
<i>Nuclear envelope</i>	1	0	0	0	2	0	<b>14</b>	0	5	0	1	0	0	23
	3	4	1	0	1	1	<b>20</b>	0	1	0	0	0	0	31
<i>Cajal body</i>	0	2	0	0	0	0	0	<b>8</b>	0	1	0	0	0	11
	3	3	2	0	1	1	0	<b>2</b>	0	0	0	0	0	12
<i>Nuclear pore complex</i>	0	1	0	1	2	0	5	0	<b>10</b>	0	1	0	0	20
	4	3	1	1	1	0	1	0	<b>5</b>	0	0	0	0	16
<i>Nucleoplasm</i>	1	3	2	2	0	0	0	0	0	<b>5</b>	0	0	0	13
	38	35	28	9	7	8	5	10	3	<b>8</b>	2	1	2	156
<i>Kinetochore</i>	2	3	0	1	0	0	0	0	2	1	<b>3</b>	0	0	12
	2	2	0	0	1	0	1	1	0	1	<b>2</b>	0	0	10
<i>Spindle apparatus</i>	0	1	0	0	0	0	0	0	0	0	0	<b>1</b>	0	2
	32	26	19	8	10	9	7	5	1	1	1	<b>7</b>	1	127
<i>Perinucleolar</i>	0	2	0	0	0	0	0	0	0	0	0	0	<b>4</b>	6
	0	1	0	0	0	0	0	1	0	0	0	0	<b>0</b>	2
<i>None</i>	18	27	29	9	4	2	2	6	1	2	4	3	3	110
<i>% observed</i>	<i>33</i>	<i>31</i>	<i>14</i>	<i>4</i>	<i>4</i>	<i>3</i>	<i>3</i>	<i>2</i>	<i>2</i>	<i>1</i>	<i>1</i>	<i>1</i>	<i>1</i>	
<i>SUM observed</i>	<b>697</b>	<b>653</b>	<b>292</b>	<b>95</b>	<b>80</b>	<b>74</b>	<b>72</b>	<b>42</b>	<b>34</b>	<b>29</b>	<b>25</b>	<b>14</b>	<b>13</b>	

The confusion matrix for LocNuclei predictions on the development set with the columns showing the number of observed and the rows the number of predicted proteins. In each cell, the upper numbers are always predictions through homology-based inference and lower numbers are predictions with the SVM Profile Kernel.

**Table S3: Comparison between LocNuclei and LocNuclei-NSort**

<i>Sub-nuclear compartment</i>	<i>Number of proteins</i>	<i>AUC LocNuclei-NSort</i>	<i>AUC LocNuclei</i>
Perinucleolar	13	0.71±0.04	0.83±0.03
Cajal body	35	0.64±0.02	0.55±0.03
Nuclear pore complex	30	0.85±0.02	0.83±0.02
Nuclear lamina	41	0.78±0.02	0.76±0.02
PML bodies	68	0.74±0.01	0.73±0.02
Chromatin	204	0.75±0.01	0.73±0.01
Nuclear speckle	250	0.76±0.01	0.74±0.01
Nucleolus	409	0.72±0.01	0.70±0.01
<b>Sum/Mean</b>	<b>849</b>	<b>0.74±0.02</b>	<b>0.73±0.02</b>

Comparing the original version of *LocNuclei* trained on 13 compartments and *LocNuclei-NSort* trained on the development set of NSort and predicting 8 compartments shows that *LocNuclei* can perform equally well. It is even better on proteins in the perinucleolar than *LocNuclei-NSort* while being worse on proteins located in the cajal bodies.



**Table S4: Euclidean distance between organisms based on subnuclear location spectra predicted with SVM Profile Kernel (*de novo*) or homology-based inference**

	De novo prediction				Homology-based inference			
	Human	Chimp	Mouse	Yeast	Human	Chimp	Mouse	Yeast
Human	0	7.2	1.9	12.8	0	1.7	2.8	3.9
Chimp	7.2	0	6.1	18.7	1.7	0	2.7	4.0
Mouse	1.9	6.1	0	13.3	2.8	2.7	0	4.6
Yeast	12.8	18.7	13.3	0	3.9	4.0	4.6	0

We calculate the Euclidean distance between predicted subnuclear location spectra and use that distance as proxy to identify evolutionary relationships. Comparing the results for location spectra predicted with an SVM Profile Kernel (*de novo*) to those predicted with homology-based inference shows that *de novo* prediction cannot entirely capture the expected relationships, i.e. the distance between human and mouse is smaller than between human and chimp. Location spectra predicted with homology-based inference succeed in reflecting the expected evolutionary relationship, i.e. the distance between human and chimp is the smallest

**Table S5: Top ten statistically enriched GO terms for each sub-nuclear compartment**

<i>Compartment</i>	<i>Top 10 enriched GO terms</i>	<i>Description</i>	<i>No. of enriched GO terms</i>	<i>No. of proteins in this compartment</i>
<i>Nuclear envelope</i>	GO:0006998 GO:0061024 GO:0007077 GO:0030397 GO:0051081 GO:0006409 GO:0051031 GO:0097064 GO:0075733 GO:0046794	Nuclear envelope organization Membrane organization Mitotic nuclear envelope disassembly Membrane disassembly Nuclear envelope disassembly tRNA export from nucleus tRNA transport ncRNA export from nucleus Intracellular transport of virus Transport of virus	99	58
<i>Chromatin</i>	GO:0006325 GO:2001141 GO:1903506 GO:0006355 GO:0034645 GO:0097659 GO:0006351 GO:0031326 GO:0009889 GO:0010556	Chromatin organization Regulation of RNA biosynthetic process Regulation of nucleic acid-templated transcription Regulation of transcription, DNA-templated Cellular macromolecule biosynthetic process Nucleic acid-templated transcription Transcription, DNA-templated Regulation of cellular biosynthetic process Regulation of biosynthetic process Regulation of macromolecule biosynthetic process	97	1901
<i>Nuclear pore complex</i>	GO:0006606 GO:0017038 GO:0006913 GO:0051169 GO:0051170 GO:0051168 GO:0006409 GO:0051031 GO:0034504 GO:0071705	Protein import into nucleus Protein import Nucleocytoplasmic transport Nuclear transport Import into nucleus Nuclear export tRNA export from nucleus tRNA transport Protein localization to nucleus Nitrogen compound transport	77	141
<i>Nuclear speckle</i>	GO:0008380 GO:0006397 GO:0016071 GO:0000375 GO:0000377 GO:0000398 GO:0006396 GO:0043484 GO:0048024 GO:0050684	RNA splicing mRNA processing mRNA metabolic process RNA splicing via transesterification reactions RNA splicing via transesterification reactions with bulged adenosine mRNA splicing via spliceosome RNA processing Regulation of RNA splicing Regulation of mRNA splicing via spliceosome Regulation of mRNA processing	71	976
<i>PML body</i>	GO:0043401 GO:0009755 GO:0070936	Steroid hormone mediated signalling pathway Hormone-mediated signalling pathway Protein K48-linked ubiquitination	64	470

<i>Compartment</i>	<i>Top 10 enriched GO terms</i>	<i>Description</i>	<i>No. of enriched GO terms</i>	<i>No. of proteins in this compartment</i>
	GO:0006357 GO:0006355 GO:0045944 GO:1903506 GO:2001141 GO:0030522 GO:0048522	Regulation of transcription by RNA polymerase II Regulation of transcription, DNA-templated Positive regulation of transcription by RNA polymerase II Regulation of nucleic acid-templated transcription Regulation of RNA biosynthetic process Intracellular receptor signalling pathway Positive regulation of cellular process		
<i>Cajal body</i>	GO:0060333 GO:0060337 GO:0016074 GO:0043170 GO:0006807 GO:0044238 GO:0071704 GO:0044237 GO:0000387 GO:0031118	Interferon-gamma-mediated signalling pathway Type I interferon signalling pathway snoRNA metabolic process Macromolecule metabolic process Nitrogen compound metabolic process Primary metabolic process Organic substance metabolic process Cellular metabolic process Spliceosomal snRNP assembly rRNA pseudouridine synthesis	62	67
<i>Nucleolus</i>	GO:0006364 GO:0016072 GO:0034470 GO:0034660 GO:0006396 GO:0044085 GO:0022613 GO:0009451 GO:0030490 GO:0006399	rRNA processing rRNA metabolic process ncRNA processing ncRNA metabolic process RNA processing Cellular component biogenesis Ribonucleoprotein complex biogenesis RNA modification Maturation of SSU-rRNA tRNA metabolic process	54	1856
<i>Kinetochores</i>	GO:0007062 GO:0051301 GO:0007059 GO:0051276 GO:0000819 GO:0008608 GO:0000070 GO:0098813 GO:0071173 GO:0071174	Sister chromatid cohesion Cell division Chromosome segregation Chromosome organization Sister chromatid segregation Attachment of spindle microtubules to kinetochore Mitotic sister chromatid segregation Nuclear chromosome segregation Spindle assembly checkpoint Mitotic spindle checkpoint	38	42
<i>Nuclear matrix</i>	GO:0021515 GO:0009725 GO:0009719 GO:0097485 GO:0007411 GO:0021527 GO:0045944 GO:1904903 GO:1904896 GO:0045935	Cell differentiation in spinal cord Response to hormone Response to endogenous stimulus Neuron projection guidance Axon guidance Spinal cord association neuron differentiation Positive regulation of transcription by RNA polymerase II ESCRT III complex disassembly ESCRT complex disassembly Positive regulation of nucleobase-containing compound metabolic process	38	120

<i>Compartment</i>	<i>Top 10 enriched GO terms</i>	<i>Description</i>	<i>No. of enriched GO terms</i>	<i>No. of proteins in this compartment</i>
<i>Nuclear lamina</i>	GO:0006998 GO:0007010 GO:0061024 GO:0051225 GO:0007030 GO:0007051 GO:0000226 GO:0090286 GO:0007017 GO:0051179	Nuclear envelope organization Cytoskeleton organization Membrane organization Spindle assembly Golgi organization Spindle organization Microtubule cytoskeleton organization Cytoskeletal anchoring at nuclear membrane Microtubule-based process Localization	25	130
<i>Peri-nucleolar</i>	GO:0048010 GO:0045445	Vascular endothelial growth factor receptor signalling pathway Myoblast differentiation	2	33
<i>Nucleoplasm</i>	GO:0031424	Keratinization	1	852
<i>Spindle apparatus</i>			0 (lowest p-value= 0.0247)	13
<i>Traveler</i>	GO:0051179 GO:0051234 GO:0006810 GO:0008104 GO:0015833 GO:0042886 GO:0015031 GO:0033036 GO:0045184 GO:0016192	Localization Establishment of localization Transport Protein localization Peptide transport Amide transport Protein transport Macromolecule localization Establishment of protein localization Vesicle-mediated transport	207	2248

The table displays the overall number of enriched GO terms and the top ten enriched GO terms for each sub-nuclear compartment and for proteins predicted as traveler. GO enrichment analysis was performed for the 5,088 proteins of the human nuclear proteome and their sub-nuclear localizations as predicted by LocNuclei. A GO term is considered as enriched if the p-value, adjusted for multiple testing, is  $< 0.01$ . Terms are ranked by p-value where the first term is the term with the lowest p-value. For Spindle apparatus, there are no enriched terms which is probably due to the low number of proteins prediction for this localization (13). For compartments with many proteins and a clearly defined function like nucleolus, chromatin, kinetochores, nuclear envelope or nuclear pore complex, the enriched GO terms reflect the expected functionality. Also for traveler proteins, the enriched GO terms all associated with transport and localization give evidence about the functionality of these proteins.

**Table S6: Normalization of sub-nuclear localization terms**

<b>Databases term</b>	<b>Normalized term</b>
Cajal body, cajal bodies, gem	Cajal bodies
Chromatin, centromere, chromosome, heterochromatin, telomere, unsynapsed chromosome axes	Chromatin
Nuclear envelope, nuclear membrane, nucleus membrane	Nuclear envelope
Nuclear lamina, nuclear periphery, nucleus lamina	Nuclear lamina
Nuclear matrix, nucleus matrix	Nuclear matrix
Nuclear pore	Nuclear pore complex
Nuclear speckle	Nuclear speckles
Nucleolus, nucleolar	Nucleolus
Nucleoplasm	Nucleoplasm
Perinucleolar	Perinucleolar compartment
PML body, nuclear dots, PML-NBs, PML/ND10 bodies	PML bodies
Kinetochores	Kinetochores
Spindle apparatus, spindle microtubules, spindle midzone, spindle poles	Spindel apparatus

Databases HPRD [1], NMPdb [2], NOPdb [3], NPD [4], NSort/DB [5] and Swiss-Prot [6] annotate sub-nuclear proteins using synonyms for some terms. We extracted these terms and normalized them to 13 sub-nuclear localization classes. The normalization was done case-insensitive; terms of the same class are separated by comma.

**Table S7: Chosen hyperparameters for the 14 different SVM Profile Kernels**

	k	$\sigma$	C	tol
Cajal Body	4	7	1.0	0.0001
Chromatin	4	7	2.0	0.0001

Kinetochores	5	8	1.0	0.0001
Nuclear Envelope	4	7	2.0	0.1
Nuclear Lamina	4	7	2.0	0.1
NPC	3	5	2.0	0.1
Nuclear speckle	3	5	2.0	0.1
Nuclear matrix	4	6	1.0	0.0001
Nucleolus	4	9	2.0	0.0001
Nucleoplasm	3	7	0.5	0.0001
PML body	4	8	2.0	0.0001
Perinucleolar	4	7	2.0	0.0001
Spindle apparatus	4	7	1.0	0.0001
Traveler	3	6	0.1	0.1

For the Profile Kernel, we optimized the parameters  $k$ , the k-mer length, and  $\gamma$ , the conservation threshold. For the SVMs, we focused on optimizing  $C$ , the penalty parameter of the error term, and  $tol$ , the tolerance for the stopping criterion. We optimized all parameters for the 14 SVMs independently.

## References for Supporting Online Material

1. Keshava Prasad, T.S., et al., *Human Protein Reference Database--2009 update*. *Nucleic Acids Res*, 2009. **37**(Database issue): p. D767-72.
2. Mika, S. and B. Rost, *NMPdb: Database of Nuclear Matrix Proteins*. *Nucleic Acids Res*, 2005. **33**(Database issue): p. D160-3.
3. Leung, A.K., et al., *NOPdb: Nucleolar Proteome Database*. *Nucleic Acids Res*, 2006. **34**(Database issue): p. D218-20.
4. Dellaire, G., R. Farrall, and W.A. Bickmore, *The Nuclear Protein Database (NPD): sub-nuclear localisation and functional annotation of the nuclear proteome*. *Nucleic Acids Res*, 2003. **31**(1): p. 328-30.
5. Willadsen, K., N. Mohamad, and M. Boden, *NSort/DB: an intranuclear compartment protein database*. *Genomics Proteomics Bioinformatics*, 2012. **10**(4): p. 226-9.
6. Bairoch, A. and R. Apweiler, *The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000*. *Nucleic Acids Res*, 2000. **28**(1): p. 45-8.

7. Sander, C. and R. Schneider, *Database of homology-derived protein structures and the structural meaning of sequence alignment*. *Proteins*, 1991. **9**(1): p. 56-68.
8. Rost, B., *Twilight zone of protein sequence alignments*. *Protein Eng*, 1999. **12**(2): p. 85-94.





## 6. Conclusion

Elucidating a protein's function is crucial to understand its overall role in the organism and to shed light on the molecular mechanisms of life. Despite its importance, protein function is not a well-defined concept because it is not only determined through the protein sequence but is also influenced by various other factors. Furthermore, determining protein function experimentally remains difficult, while protein sequencing has become cheaper and easier in the last couple of years, leading to a rapidly growing number of available protein sequences without any functional annotations. Computational biology has focused on bridging this sequence-annotation gap through prediction methods.

This dissertation focused on three different aspects to describe protein function, namely Gene Ontology (GO) terms, binding residues, and sub-nuclear localization, and developed new methods to predict those aspects. To predict GO terms, we proposed *goPredSim* (see Chapter 3), a new method applying an unsupervised approach to transfer annotations. It follows a similar concept as homology-based inference but uses embedding similarity instead of sequence similarity to identify evolutionary related proteins. Embeddings are derived from language models which were pre-trained on large sets of sequences without using any annotations (self-supervised training). *goPredSim* clearly outperformed homology-based inference indicating that embedding similarity captures functional relations better than sequence similarity and allows for the identification of more distant relatives. Therefore, *goPredSim* is a method to predict GO terms which is much simpler than existing state-of-the-art methods while still achieving good performance. In addition, it enables annotation transfer between proteins not captured through sequence similarity making it broader applicable than homology-based inference.

Further, we developed *bindPredictML17* (see Section 4.1), a method to predict binding residues. It is based on an Artificial Neural Network (ANN) using evolutionary informa-

## 6. Conclusion

---

tion derived from evolutionary couplings and mutation effect predictions. Being solely based on sequence information, bindPredictML17 achieved  $F_1 = 26.2 \pm 0.8\%$ , and the predicted binding residues often formed spatial clusters in the protein structure. Such predictions indicate that residues incorrectly predicted as binding could still be close to the binding site and stabilize it, or could point towards missing binding annotations.

While bindPredictML17 achieved good performance, it relies on evolutionary information that cannot be computed for all proteins. To improve upon bindPredictML17, we proposed *bindPredictDL* (see Section 4.2) which replaced the ANN with a Convolutional Neural Network (CNN) and does not rely on evolutionary information as input. Instead, it uses a transfer learning approach utilizing embeddings. bindPredictDL distinguishes three different types of binding residues (binding to small molecules, metal ions, or nucleic acids). It clearly outperformed bindPredictML17 by five percentage points. Combining bindPredictDL with homology-based inference increased performance further, leading to  $F_1 = 44 \pm 2\%$  for the binary task of predicting whether a residue binds a ligand or not and  $F_1 = 27 \pm 3\%$ ,  $F_1 = 20 \pm 3\%$ ,  $F_1 = 39 \pm 2\%$  for the prediction of binding residues for metal ions, nucleic acids, and small molecules, respectively. bindPredictDL constitutes a method for binding residue prediction which is based on embeddings that can easily be obtained for all available protein sequences. Therefore, bindPredictDL is solely based on sequence information and does not require structural information or hand-crafted features as input. Compared to other existing binding residue predictors, it is not restricted to a certain ligand or set of ligands but allows predictions for various different types of binding and can even distinguish between three major groups of ligands.

To predict sub-nuclear localization, *LocNuclei* (see Chapter 5) uses a combination of homology-based inference and a Profile Kernel SVM. It allows distinction of 13 different compartments and also the assignment of multiple localizations to one protein. LocNuclei achieved an overall accuracy ( $Q_{13}$ , Eqn. 5.1) of  $62 \pm 3\%$ , and the predicted compartments matched the expected functions as described through GO terms. Therefore, LocNuclei allows to predict protein localization on a fine-grained level for one specific compartment, namely the nucleus. To account for the dynamic organization of the nucleus, LocNuclei can predict multiple sub-nuclear compartments for one protein. The resulting predictions allow to draw conclusions about a protein's function.

In general, the methods presented in this dissertation reflect the diversity of protein function by leveraging various Machine Learning (ML) techniques, each offering specific

---

strengths. Homology-based inference remains an approach which achieves very high performance for protein function prediction but is limited to a small subset of proteins. Because of their complementarity, combining homology-based inference with ML achieves the best performance for function prediction. While ML models often relied on evolutionary information and difficult to compute features in the past, the rise of Deep Learning introduces new possibilities and features that allow fast and accurate predictions for large sets of protein sequences. Especially protein embeddings represent a powerful new input replacing hand-crafted features. They can, for example, serve as input to train supervised ML algorithms. Additionally, the inference based on embedding similarity introduced in this dissertation can complement homology-based inference and enable annotation transfer between evolutionary related proteins with low sequence similarity. Future advances in the quality of protein embeddings could further improve their predictive power and, therefore, boost performance of methods to predict protein function. However, the applied concepts and developed methods do not only show that ML can predict protein function but also highlight the large potential of ML for any prediction task in computational biology. In fact, the study presented in Chapter 2 showed the relevance of ML in biology and medicine through an assessment of published literature highlighting the vast number of possibilities for applications of ML in biology and medicine, and the importance of collaborative research at the intersection of computer science and life sciences.

In conclusion, this dissertation advanced prediction methods for three main aspects of protein function: GO terms, binding residues, and sub-nuclear localization. To do so, we developed different prediction methods utilizing heterogeneous ML concepts by applying supervised learning based on hand-crafted features (LocNuclei, bindPredictML17), transfer learning utilizing embeddings from pre-trained language models (bindPredictDL), or unsupervised learning performing a simple annotation transfer between proteins with similar embeddings (goPredSim). Those methods achieve high prediction performance, rely solely on sequence information, and are more broadly applicable than other methods. In addition, the diversity of the applied concepts shows the large potential of ML applications not only for protein function prediction, but for computational biology in general.



## References

- [1] Whisstock, J. C. and Lesk, A. M. Prediction of protein function from protein sequence and structure. *Quarterly Reviews of Biophysics*, 36(3):307, 2003.
- [2] Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., et al. *Molecular Biology of the Cell*. Garland Science, Taylor and Francis Group, 2018.
- [3] Lee, D., Redfern, O., and Orengo, C. Predicting protein function from sequence and structure. *Nature Reviews Molecular Cell Biology*, 8(12):995–1005, 2007.
- [4] Krallinger, M. and Valencia, A. Text-mining and information-retrieval services for molecular biology. *Genome Biology*, 6(7):1–8, 2005.
- [5] Bairoch, A. The ENZYME database in 2000. *Nucleic Acids Research*, 28(1):304–305, 2000.
- [6] Boutet, E., Lieberherr, D., Tognolli, M., and Bairoch, A. UniProtKB/Swiss-Prot. *Methods in Molecular Biology*, 406:89–112, 2007.
- [7] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., et al. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(suppl\_1):D354–D357, 2006.
- [8] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
- [9] Gene Ontology Consortium. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, 47(D1):D330–D338, 2019.
- [10] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., et al. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.
- [11] PDB. <https://www.rcsb.org/>, 2021. Accessed: 2021-01-12.
- [12] Yang, J., Roy, A., and Zhang, Y. BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research*, 41(D1):D1096–D1103, 2012.
- [13] BioLiP. <https://zhanglab.ccmb.med.umich.edu/BioLiP/>, 2021. Accessed: 2021-01-12.

- [14] Vastrik, I., D'Eustachio, P., Schmidt, E., Joshi-Tope, G., Gopinath, G., et al. Reactome: a knowledge base of biologic pathways and processes. *Genome Biology*, 8(3):1–13, 2007.
- [15] Caspi, R., Foerster, H., Fulcher, C. A., Hopkinson, R., Ingraham, J., et al. MetaCyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 34(suppl\_1):D511–D516, 2006.
- [16] Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., et al. The funCat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 32(18):5539–5545, 2004.
- [17] Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., et al. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4(1):1–14, 2003.
- [18] Thul, P. J., Åkesson, L., Wiking, M., Mahdessian, D., Geladaki, A., et al. A subcellular map of the human proteome. *Science*, 356(6340), 2017.
- [19] Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'Donovan, C., et al. QuickGO: a web-based tool for Gene Ontology searching. *Bioinformatics*, 25(22):3045–3046, 2009.
- [20] Zhao, Y., Wang, J., Chen, J., Zhang, X., Guo, M., et al. A Literature Review of Gene Function Prediction by Modeling Gene Ontology. *Frontiers in Genetics*, 11:400, 2020.
- [21] Camon, E., Magrane, M., Barrell, D., Lee, V., Dimmer, E., et al. The Gene Ontology Annotation (GOA) database: sharing knowledge in uniprot with gene ontology. *Nucleic Acids Research*, 32(Database issue):D262–D266, 2004.
- [22] Huntley, R. P., Sawford, T., Mutowo-Meullenet, P., Shypitsyna, A., Bonilla, C., et al. The GOA database: gene ontology annotation updates for 2015. *Nucleic Acids Research*, 43(D1):D1057–D1063, 2015.
- [23] GOA. <http://www.ebi.ac.uk/GOA>, 2021. Accessed: 2021-01-11.
- [24] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 2019.
- [25] Radivojac, P., Clark, W. T., Oron, T. R., Schnoes, A. M., Wittkop, T., et al. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013.
- [26] Jiang, Y., Oron, T. R., Clark, W. T., Bankapur, A. R., D'Andrea, D., et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, 17(1):1–19, 2016.

- 
- [27] Zhou, N., Jiang, Y., Bergquist, T. R., Lee, A. J., Kacsoh, B. Z., et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology*, 20(1):1–23, 2019.
- [28] Schmidt, T., Haas, J., Cassarino, T. G., and Schwede, T. Assessment of ligand-binding residue predictions in CASP9. *Proteins: Structure, Function, and Bioinformatics*, 79(S10):126–136, 2011.
- [29] Furnham, N., Holliday, G. L., de Beer, T. A., Jacobsen, J. O., Pearson, W. R., et al. The Catalytic Site Atlas 2.0: cataloging catalytic sites and residues identified in enzymes. *Nucleic Acids Research*, 42(D1):D485–D489, 2014.
- [30] Smyth, M. and Martin, J. x Ray crystallography. *Molecular Pathology*, 53(1):8, 2000.
- [31] Lopez, G., Valencia, A., and Tress, M. FireDB—a database of functionally important residues from proteins of known structure. *Nucleic acids research*, 35(suppl\_1):D219–D223, 2007.
- [32] Dessailly, B. H., Lensink, M. F., Orengo, C. A., and Wodak, S. J. LigA-Site—a database of biologically relevant binding sites in proteins with known apo-structures. *Nucleic Acids Research*, 36(suppl\_1):D667–D673, 2007.
- [33] Benson, M. L., Smith, R. D., Khazanov, N. A., Dimcheff, B., Beaver, J., et al. Binding MOAD, a high-quality protein–ligand database. *Nucleic Acids Research*, 36(suppl\_1):D674–D678, 2007.
- [34] Liu, T., Lin, Y., Wen, X., Jorissen, R. N., and Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Research*, 35(suppl\_1):D198–D201, 2007.
- [35] Wang, R., Fang, X., Lu, Y., and Wang, S. The PDBbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of Medicinal Chemistry*, 47(12):2977–2980, 2004.
- [36] Webb, E. C. et al. *Enzyme nomenclature 1992. Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Ed. 6. Academic Press, 1992.
- [37] Tyzack, J. D., Fernando, L., Ribeiro, A. J., Borkakoti, N., and Thornton, J. M. Ranking enzyme structures in the PDB by bound ligand similarity to biological substrates. *Structure*, 26(4):565–571, 2018.
- [38] Andrade, M. A., O’Donoghue, S. I., and Rost, B. Adaptation of protein surfaces to subcellular location. *Journal of Molecular Biology*, 276(2):517–525, 1998.

- [39] Jensen, L. J., Gupta, R., Blom, N., Devos, D., Tamames, J., et al. Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology*, 319(5):1257–1265, 2002.
- [40] Rost, B., Liu, J., Nair, R., Wrzeszczynski, K. O., and Ofran, Y. Automatic prediction of protein function. *Cellular and Molecular Life Sciences CMLS*, 60(12):2637–2650, 2003.
- [41] Cokol, M., Nair, R., and Rost, B. Finding nuclear localization signals. *EMBO reports*, 1(5):411–415, 2000.
- [42] Durand, E., Verger, D., Rêgo, A. T., Chandran, V., Meng, G., et al. Structural biology of bacterial secretion systems in Gram-negative pathogens-potential for new drug targets. *Infectious Disorders-Drug Targets (Formerly Current Drug Targets-Infectious Disorders)*, 9(5):518–547, 2009.
- [43] Negi, S., Pandey, S., Srinivasan, S. M., Mohammed, A., and Guda, C. LocSigDB: a database of protein localization signals. *Database*, 2015, 2015.
- [44] Imai, K. and Nakai, K. Tools for the Recognition of Sorting Signals and the Prediction of Subcellular Localization of Proteins From Their Amino Acid Sequences. *Frontiers in Genetics*, 11:1491, 2020.
- [45] Freitas, N. and Cunha, C. Mechanisms and signals for the nuclear import of proteins. *Current genomics*, 10(8):550, 2009.
- [46] Nair, R., Carter, P., and Rost, B. NLSdb: database of nuclear localization signals. *Nucleic Acids Research*, 31(1):397–399, 2003.
- [47] Bernhofer, M., Goldberg, T., Wolf, S., Ahmed, M., Zaugg, J., et al. NLSdb—major update for database of nuclear localization signals and nuclear export signals. *Nucleic Acids Research*, 46(D1):D503–D508, 2018.
- [48] Carbon, S., Ireland, A., Mungall, C. J., Shu, S., Marshall, B., et al. AmiGO: online access to ontology and annotation data. *Bioinformatics*, 25(2):288–289, 2009.
- [49] Jupp, S., Burdett, T., Leroy, C., and Parkinson, H. E. A new Ontology Lookup Service at EMBL-EBI. In *SWAT4LS*, pages 118–119. 2015.
- [50] Mika, S. and Rost, B. NMPdb: database of nuclear matrix proteins. *Nucleic Acids Research*, 33(suppl\_1):D160–D163, 2005.
- [51] Leung, A. K. L., Trinkle-Mulcahy, L., Lam, Y. W., Andersen, J. S., Mann, M., et al. NOPdb: nucleolar proteome database. *Nucleic Acids Research*, 34(suppl\_1):D218–D220, 2006.
- [52] Willadsen, K., Mohamad, N., and Bodén, M. NSort/DB: an intranuclear compartment protein database. *Genomics, Proteomics & Bioinformatics*, 10(4):226–229, 2012.



- 
- [53] Goldberg, T., Hamp, T., and Rost, B. LocTree2 predicts localization for all domains of life. *Bioinformatics*, 28(18):i458–i465, 2012.
- [54] Almagro Armenteros, J. J., S nderby, C. K., S nderby, S. K., Nielsen, H., and Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017.
- [55] Koonin, E. V. Orthologs, paralogs, and evolutionary genomics. *Annu. Rev. Genet.*, 39:309–338, 2005.
- [56] Mahlich, Y., Steinegger, M., Rost, B., and Bromberg, Y. HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics*, 34(13):i304–i312, 2018.
- [57] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
- [58] Altschul, S. F., Madden, T. L., Sch ffer, A. A., Zhang, J., Zhang, Z., et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.
- [59] Steinegger, M. and S ding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, 2017.
- [60] Goldberg, T., Hecht, M., Hamp, T., Karl, T., Yachdav, G., et al. LocTree3 prediction of localization. *Nucleic Acids Research*, 42(W1):W350–W355, 2014.
- [61] Hamp, T., Kassner, R., Seemayer, S., Vicedo, E., Schaefer, C., et al. Homology-based inference sets the bar high for protein function prediction. In *BMC Bioinformatics*, volume 14, page S7. Springer, 2013.
- [62] Qiu, J., Bernhofer, M., Heinzinger, M., Kemper, S., Norambuena, T., et al. ProNA2020 predicts protein-DNA, protein-RNA and protein-protein binding proteins and residues from sequence. *Journal of Molecular Biology*, 2020.
- [63] Cios, K. J., Kurgan, L. A., and Reformat, M. Machine learning in the life sciences. *IEEE Engineering in Medicine and Biology Magazine*, 26(2):14–16, 2007.
- [64] Chicco, D. Ten quick tips for machine learning in computational biology. *BioData Mining*, 10(35), 2017.
- [65] Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., et al. Profile-based string kernels for remote homology detection and motif extraction. *Journal of Bioinformatics and Computational Biology*, 3(03):527–550, 2005.
- [66] Hamp, T., Goldberg, T., and Rost, B. Accelerating the Original Profile Kernel. *PLOS ONE*, 8(6), 2013. 10.1371/journal.pone.0068459.

- [67] Hecht, M., Bromberg, Y., and Rost, B. Better prediction of functional effects for sequence variants. *BMC Genomics*, 16(S8):S1, 2015.
- [68] Yachdav, G., Kloppmann, E., Kajan, L., Hecht, M., Goldberg, T., et al. PredictProtein—an open resource for online prediction of protein structural and functional features. *Nucleic Acids Research*, 42(W1):W337–W343, 2014. 10.1093/nar/gku366.
- [69] Asgari, E. and Mofrad, M. R. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PLOS ONE*, 10(11):e0141287, 2015.
- [70] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [71] Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., et al. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(723), 2019.
- [72] Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., et al. ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv preprint arXiv:2007.06225*, 2020.
- [73] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., et al. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [74] Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [75] Mousa, A. and Schuller, B. Contextual Bidirectional Long Short-Term Memory Recurrent Neural Network Language Models: a generative approach to sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1023–1032. 2017.
- [76] Peters, M. E., Ammar, W., Bhagavatula, C., and Power, R. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.
- [77] Steinegger, M. and Söding, J. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):1–8, 2018.
- [78] Steinegger, M., Mirdita, M., and Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manifold. *Nature Methods*, 16(7):603–606, 2019.
- [79] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

- 
- [80] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., et al. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008. 2017.
- [81] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [82] Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., et al. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. *arXiv preprint arXiv:1805.09843*, 2018.
- [83] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.
- [84] Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T., and Rost, B. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific Reports*, 11(1):2045–2322, 2021. doi:10.1038/s41598-020-80786-0.
- [85] Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, page 622803, 2019.
- [86] Vig, J., Madani, A., Varshney, L. R., Xiong, C., Socher, R., et al. BERTology Meets Biology: Interpreting Attention in Protein Language Models. *arXiv preprint arXiv:2006.15222*, 2020.
- [87] You, R., Zhang, Z., Xiong, Y., Sun, F., Mamitsuka, H., et al. GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics*, 34(14):2465–2473, 2018. 10.1093/bioinformatics/bty130.
- [88] You, R., Yao, S., Xiong, Y., Huang, X., Sun, F., et al. NetGO: improving large-scale protein function prediction with massive network information. *Nucleic Acids Research*, 47(W1):W379–W387, 2019.
- [89] Kulmanov, M. and Hoehndorf, R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2019. 10.1093/bioinformatics/btz595.
- [90] Fa, R., Cozzetto, D., Wan, C., and Jones, D. T. Predicting human protein function with multi-task deep neural networks. *PLOS ONE*, 13(6):e0198216, 2018.
- [91] Kulmanov, M., Khan, M. A., and Hoehndorf, R. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, 2018.
- [92] Yang, J., Roy, A., and Zhang, Y. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

- [93] Hu, X., Wang, K., and Dong, Q. Protein ligand-specific binding residue predictions by an ensemble classifier. *BMC Bioinformatics*, 17(1):470, 2016.
- [94] Hu, X., Dong, Q., Yang, J., and Zhang, Y. Recognizing metal and acid radical ion-binding sites by integrating ab initio modeling with template-based transferals. *Bioinformatics*, 32(21):3260–3269, 2016.
- [95] Cui, Y., Dong, Q., Hong, D., and Wang, X. Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC Bioinformatics*, 20(1):93, 2019.
- [96] Zhang, C., Freddolino, P. L., and Zhang, Y. COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information. *Nucleic Acids Research*, 45(W1):W291–W299, 2017.
- [97] Brylinski, M. and Skolnick, J. A threading-based method (findsite) for ligand-binding site prediction and functional annotation. *Proceedings of the National Academy of Sciences*, 105(1):129–134, 2008.
- [98] Capra, J. A., Laskowski, R. A., Thornton, J. M., Singh, M., and Funkhouser, T. A. Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure. *PLoS Comput Biol*, 5(12):e1000585, 2009.
- [99] Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [100] Shu, N., Zhou, T., and Hovmöller, S. Prediction of zinc-binding sites in proteins from sequence. *Bioinformatics*, 24(6):775–782, 2008.
- [101] Xia, C.-Q., Pan, X., and Shen, H.-B. Protein–ligand binding residue prediction enhancement through hybrid deep heterogeneous learning of sequence and structure data. *Bioinformatics*, 36(10):3018–3027, 2020.
- [102] Armenteros, J. J. A., Tsirigos, K. D., Sønderby, C. K., Petersen, T. N., Winther, O., et al. SignalP 5.0 improves signal peptide predictions using deep neural networks. *Nature Biotechnology*, 37(4):420–423, 2019.
- [103] Wrzeszczynski, K. and Rost, B. Annotating proteins from endoplasmic reticulum and golgi apparatus in eukaryotic proteomes. *Cellular and Molecular Life Sciences CMLS*, 61(11):1341–1353, 2004.
- [104] Nair, R. and Rost, B. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, 18(suppl\_1):S78–S86, 2002.
- [105] Domingos, P. A Few Useful Things to Know about Machine Learning. *Commun. ACM*, 55(10):78–87, 2012. ISSN 0001-0782. 10.1145/2347736.2347755.

- 
- [106] Chen, S., Arsenault, C., and Larivière, V. Are top-cited papers more interdisciplinary? *Journal of Informetrics*, 9(4):1034 – 1046, 2015. ISSN 1751-1577. <https://doi.org/10.1016/j.joi.2015.09.003>.
- [107] Abramo, G., D’Angelo, C. A., and Di Costa, F. Authorship analysis of specialized vs diversified research output. *Journal of Informetrics*, 13(2):564–573, 2019.
- [108] Abramo, G., D’Angelo, C. A., and Di Costa, F. Do interdisciplinary research teams deliver higher gains to science? *Scientometrics*, 111(1):317–336, 2017.
- [109] Chen, S., Arsenault, C., Gingras, Y., and Larivière, V. Exploring the interdisciplinary evolution of a discipline: the case of Biochemistry and Molecular Biology. *Scientometrics*, 102(2):1307–1323, 2015.
- [110] Xie, Z., Li, M., Li, J., Duan, X., and Ouyang, Z. Feature analysis of multidisciplinary scientific collaboration patterns based on PNAS. *EPJ Data Science*, 7:1–17, 2018.
- [111] Rinia, E., van Leeuwen, T., and van Raan, A. Impact measures of interdisciplinary research in physics. *Scientometrics*, 53(2):241–248, 2002.
- [112] Larivière, V. and Gingras, Y. On the relationship between interdisciplinarity and scientific impact. *Journal of the American Society for Information Science and Technology*, 61(1):126–131, 2010.
- [113] Littmann, M., Selig, K., Cohen-Lavi, L., Frank, Y., Hönigschmid, P., et al. Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nature Machine Intelligence*, 2(1):18–24, 2020. doi:10.1038/s42256-019-0139-8.
- [114] El-Mabrouk, N. and Slonim, D. K. ISMB 2020 proceedings. *Bioinformatics*, 36(Supplement\_1):i1–i2, 2020. 10.1093/bioinformatics/btaa537.
- [115] PredictProtein. <https://predictprotein.org>, 2021. Accessed: 2021-01-23.
- [116] Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., et al. Protein 3D structure computed from evolutionary sequence variation. *PLOS ONE*, 6(12):e28766, 2011.
- [117] Marks, D. S., Hopf, T. A., and Sander, C. Protein structure prediction from sequence variation. *Nature Biotechnology*, 30(11):1072–1080, 2012.
- [118] Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P., Springer, M., et al. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, 2017.
- [119] Teppa, E., Wilkins, A. D., Nielsen, M., and Buslje, C. M. Disentangling evolutionary signals: conservation, specificity determining positions and coevolution. implication for catalytic residue prediction. *BMC Bioinformatics*, 13(1):235, 2012.

- [120] Hopf, T. A., Green, A. G., Schubert, B., Mersmann, S., Schärfe, C. P. I., et al. The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, 2018. ISSN 1367-4803. 10.1093/bioinformatics/bty862.
- [121] Schelling, M., Hopf, T. A., and Rost, B. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins: Structure, Function, and Bioinformatics*, 86(10):1064–1074, 2018. doi:10.1002/prot.25585.
- [122] Norambuena, T. and Melo, F. The protein-DNA interface database. *BMC Bioinformatics*, 11(1):1–12, 2010.
- [123] Mika, S. and Rost, B. UniqueProt: creating representative protein sequence sets. *Nucleic Acids Research*, 31(13):3789–3791, 2003.
- [124] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [125] Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv preprint arXiv:1511.07289*, 2015.
- [126] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [127] Smith, T. F., Waterman, M. S., et al. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.
- [128] Carmo-Fonseca, M. The contribution of nuclear compartmentalization to gene regulation. *Cell*, 108(4):513–521, 2002.
- [129] Chubb, J. R. and Bickmore, W. A. Considering nuclear compartmentalization in the light of nuclear dynamics. *Cell*, 112(4):403–406, 2003.
- [130] Littmann, M., Goldberg, T., Seitz, S., Bodén, M., and Rost, B. Detailed prediction of protein sub-nuclear localization. *BMC Bioinformatics*, 20(205), 2019. doi:10.1186/s12859-019-2790-9.

# A. Appendix

## List of Publications

The publication-based dissertation at hand is based on the following four peer-reviewed and published publications:

- **Maria Littmann**, Katharina Selig, Liel Cohen-Lavi, Yotam Frank, Peter Hönig-schmid, et al. Validity of machine learning in biology and medicine increased through collaborations across fields of expertise. *Nature Machine Intelligence*, 2 (1):18–24, 2020. doi:10.1038/s42256-019-0139-8
- **Maria Littmann**, Michael Heinzinger, Christian Dallago, Tobias Olenyi, and Burkhard Rost. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific Reports*, 11(1):2045–2322, 2021. doi:10.1038/s41598-020-80786-0
- **Maria Schelling**, Thomas A Hopf, and Burkhard Rost. Evolutionary couplings and sequence variation effect predict protein binding sites. *Proteins: Structure, Function, and Bioinformatics*, 86(10):1064–1074, 2018. doi:10.1002/prot.25585
- **Maria Littmann**, Tatyana Goldberg, Sebastian Seitz, Mikael Bodén, and Burkhard Rost. Detailed prediction of protein sub-nuclear localization. *BMC Bioinformatics*, 20(205), 2019. doi:10.1186/s12859-019-2790-9

The results discussed in Section 4.2 of this dissertation are part of a manuscript in preparation.

In addition to the publications above, I co-authored the following publications not discussed in this dissertation:

- Yannick Mahlich, Jonas Reeb, Maximilian Hecht, **Maria Schelling**, Tjaart Andries Petrus de Beer, et al. Common sequence variants affect molecular function more than rare variants? *Scientific Reports*, 7(1):1-13, 2017. doi:10.1038/s41598-017-01054-2
- Linus Scheibenreif, **Maria Littmann**, Christine Orengo, and Burkhard Rost. FunFam protein families improve residue level molecular function prediction. *BMC Bioinformatics*, 20(400). 2019. doi:10.1186/s12859-019-2988-x

## A. Appendix

---

- Christian Dallago, Konstantin Schütze, Michael Heinzinger, Tobias Olenyi, **Maria Littmann**, et al. Streamlining value of protein embeddings through bio\_embeddings. Presented at LMRL, NeurIPS2020 and published online at <https://www.lmr1.org/papers>

Also, I have co-authored the following publication that has been submitted to peer-review and has been published as pre-print on bioRxiv. The results are not discussed in this dissertation.

- **Maria Littmann**, Nicola Bordin, Michael Heinzinger, Christine Orengo, and Burkhard Rost. Clustering FunFams using sequence embeddings improves EC purity. bioRxiv, 2021. doi: 10.1101/2021.01.21.427551

Another publication which I co-authored but was not discussed in this dissertation has also been submitted to peer-review.

- Christian Dallago, Konstantin Schütze, Michael Heinzinger, Tobias Olenyi, **Maria Littmann**, et al. Learned embeddings from deep learning to visualize and predict protein sets. Submitted, 2021