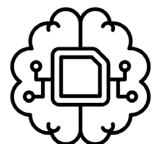
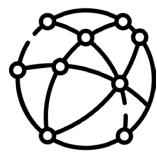




Revealing function,  
interactions, and localization  
of peroxisomal proteins  
using Deep Learning-based  
approaches



Marco **Anteghini**







## Propositions

1. Deep Learning-based embeddings of protein sequences are adaptable to predict sub-organelle protein localization.  
(this thesis)
2. Peroxisomal research requires the development of *ad-hoc* computational methods.  
(this thesis)
3. Quantum computing will revolutionize Molecular Dynamics simulations.
4. Exploring the human microbiome holds great potential for advancing personalized medicine.
5. The teaching of Bioinformatics should be done in high school.
6. Humanity must invest in research towards space colonization
7. The digital education of elderly individuals should be prioritized.

Propositions belonging to the thesis, entitled

Revealing function, interactions, and localization of peroxisomal proteins using Deep Learning-based approaches

Marco Anteghini  
Wageningen, 22 September 2023



# **Revealing function, interactions, and localization of peroxisomal proteins using Deep Learning-based approaches**

**Marco Anteghini**

## **Thesis committee**

### **Promotor**

Prof. Dr V.A.P. Martins dos Santos  
Personal chair, Bioprocess Engineering  
Wageningen University & Research

### **Co-promotor**

Dr E. Saccenti  
Assistant professor at the Laboratory of Systems and Synthetic Biology  
Wageningen University & Research

### **Other members**

Dr A. Rosato, University of Florence, Italy  
Prof. Dr M.H. Medema, Wageningen University & Research  
Dr G. Roshchupkin, Erasmus MC Medical Center  
Prof. Dr I.J. van der Klei, University of Groningen

This research was conducted under the auspices of VLAG Graduate School (Biobased, Biomolecular, Chemical, Food and Nutrition Sciences).

# **Revealing function, interactions, and localization of peroxisomal proteins using Deep Learning-based approaches**

**Marco Anteghini**

## **Thesis**

submitted in fulfilment of the requirements for the degree of doctor

at Wageningen University

by the authority of the Rector Magnificus,

Prof. Dr A.P.J. Mol,

in the presence of the

Thesis Committee appointed by the Academic Board

to be defended in public

on Friday 22 September 2023

at 11 a.m. in Omnia Auditorium.

Marco Anteghini

Revealing function, interactions, and localization of peroxisomal proteins using Deep Learning-based approaches,  
236 pages.

PhD thesis, Wageningen University, Wageningen, the Netherlands (2023)  
With references, with summary in English

ISBN: 978-94-6447-810-5

DOI: <https://doi.org/10.18174/635214>

The research described in this thesis was financially supported by The Dutch Financer.

The Financial support from the PerICo International Training Network from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968 is gratefully acknowledged.

Cover design by Francesca Menghi





# Contents

1	Introduction: of peroxisomes and deep-learning	3
2	Computational approaches for peroxisomal protein localization	37
3	In-Pero: exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins	47
4	OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal targeting signal detection	71
5	Is-mPTS: an experimentally validated tool for Pex19 - membrane peroxisomal targeting signal (mPTS) binding domain identification from protein sequence	85
6	P-PPI: accurate prediction of peroxisomal protein-protein interactions using deep learning-based protein sequence embeddings	105
7	PortPred: exploiting deep learning embeddings of protein sequences to accurately identify transporter proteins and their substrate category	125
8	Representing semantified biological assays in the Open Research Knowledge Graph	157
9	Easy semantification of bioassays	167
10	Bioinforming: training the youth	185
11	Discussion	201
12	Summary	219
13	Acknowledgements	225
14	Ringraziamenti	227
15	List of Publications	229
16	Overview of the completed training activities	233



---

# Introduction: of peroxisomes and deep-learning

1

## 1.1 The peroxisome

First described as microbodies in 1954 [1], peroxisomes are ubiquitous membrane-bound organelles [2] with a diameter ranging from 0.1 to 1  $\mu\text{m}$  [3]. Peroxisomes contain enzymes involved in a variety of reactions, particularly connected to energy metabolism [4, 5]. Some of the metabolic pathways that involve peroxisomes include phospholipid biosynthesis, fatty acid beta-oxidation, bile acid synthesis, fatty acid alpha-oxidation, glyoxylate metabolism, amino acid degradation, and Reactive Oxygen/Nitrogen species (ROS/RNS) metabolism [4]. Peroxisomes are also involved in non-metabolic functions such as cellular stress responses, response to pathogens and antiviral defense, and cellular signaling [5]. Because of this, they have gained the appellation of "protective" organelles [5], and dysfunctions in peroxisomal proteins have been associated with severe metabolic disorders [4–6].

### 1.1.1 Peroxisome biogenesis

The process of peroxisomal biogenesis involves proteins called peroxins that are encoded by a set of genes named PEX genes [7]. Peroxins are also involved in peroxisome proliferation which is a process regulated by nucleus-based signaling cascades involving transcription factors that promote the expression of peroxisomal genes, resulting in increased peroxisome quantity [8]. Peroxisomes are formed through two pathways: *de novo* biogenesis and growth/division of existing peroxisomes [9, 10]. During *de novo* biogenesis, two groups of vesicles (pre-peroxisomal vesicles) containing peroxins fuse to form a vesicle that contains all the peroxins required for matrix protein recruitment. These two groups of vesicles are usually formed in the membrane of the Endoplasmic Reticulum, or from both Endoplasmic Reticulum and Mitochondria (e.g. in mammals) [11]. This early peroxisome then imports more proteins to eventually form a mature peroxisome. Mature peroxisomes can continue to import both membrane and matrix proteins [10]. Peroxisomes can also emerge from other existing peroxisomes. These peroxisomes grow to a certain size after obtaining their peroxisomal membrane proteins (PMPs) and matrix proteins directly from the cytosol [11].

### 1.1.2 Main metabolic functions of peroxisomes in mammals

**Fatty acid beta-oxidation.** Very long-chain fatty acids (VLCFAs) such as C22:0, C24:0, and C26:0, are oxidized in peroxisomes and cannot be oxidized in mitochondria. Medium and long-chain fatty acids can also be oxidized in peroxisomes [11, 12], however, under normal conditions, the mitochondrial beta-oxidation system handles these oxidation processes [6]. In addition to VLCFAs, other metabolites whose oxidation is dependent upon peroxisomal beta-oxidation system include pristanic acid, di and trihydroxycholestanoic acid, tetracosanoic acid, and long-chain dicarboxylic acids. [12, 13].

**Phospholipids biosynthesis.** Peroxisomes synthesize ether-linked phospholipids, which are important membrane components found in various tissues, including the brain and lungs [14]. This process involves products from fatty acid  $\beta$ -oxidation.

**Bile acid biosynthesis.** Bile acids are synthesized in peroxisomes by performing a beta-oxidation step required for the formation of mature C24-bile acids from C27-bile acid intermediates. Additionally, *de novo* synthesized bile acids are conjugated within the peroxisome [15].

**Fatty acid alpha-oxidation.** Oxidative decarboxylation of 3-methyl branched-chain fatty acids occurs solely in peroxisomes through a series of enzymatic steps [13]. Fatty acid alpha-oxidation in peroxisomes is also dependent on functional interactions with other organelles [13, 16].

**Glyoxylate metabolism.** Peroxisomes detoxify the toxic metabolite glyoxylate via transamination into glycine, which prevents the formation of oxalate, a molecule that cause tissue damage [13]. The enzyme alanine glyoxylate aminotransferase (AGT) mediates this process, and peroxisomes are necessary for its function. Glyoxylate metabolism in peroxisomes involves functional cross talk with other organelles [13, 16].

**Amino acid degradation.** Liver and kidney have peroxisomes containing d-amino acid oxidase which produce imino acids and removes ammonia, a toxic molecule [17]. Additionally, there are small amounts of l-amino acid oxidase in these peroxisomes which can come into play when there is an excess of amino acids for all other pathways [17].

**Reactive Oxygen/Nitrogen species metabolism.** Peroxisomes are a major source of ROS/RNS species in mammalian cells, generated as byproducts of various metabolic pathways [18, 19]. Many enzymes involved in these pathways generate specific ROS or RNS as byproducts of their normal catalytic function [20]. Peroxisomes possess intricate protective mechanisms to counteract oxidative stress and maintain redox balance, but an imbalance between peroxisomal ROS/RNS production and removal may damage biomolecules [18].

Given their metabolic role, non-functional peroxisomes have been associated to several diseases like:

**Zellweger syndrome.** This is a rare genetic disorder that affects the development of many organs and systems, including the brain, liver, and kidneys. It is characterized by the absence or reduction of peroxisomes in cells, resulting in the accumulation of very-long-chain fatty acids and other substances that are normally broken down by peroxisomes [21].

**Neonatal adrenoleukodystrophy (NALD).** NALD is a milder form of Zellweger syndrome, with symptoms that usually appear in infancy or early childhood. It is also characterized by the absence or reduction of peroxisomes, leading to the accumulation of very-long-chain fatty acids and other substances [22].

**Infantile Refsum disease.** This is another rare genetic disorder that affects the breakdown of certain types of fat in the body, particularly phytanic acid. It is caused by a defect in peroxisome function, resulting in the accumulation of phytanic acid and other substances in the body [6].

**X-linked adrenoleukodystrophy (X-ALD).** X-ALD is a genetic disorder that affects the breakdown of very-long-chain fatty acids, resulting in their accumulation in the brain, nervous system, and adrenal glands. Peroxisomes are involved in the breakdown of very-long-chain fatty acids, and defects in peroxisome function can contribute to the development of the disorder [23].

### 1.1.3 The peroxisome proteome

#### Localization of peroxisomal proteins

Peroxisomes are single-membrane bound organelles and peroxisomal proteins only have two possible sub-organelle localization: membrane and matrix [2, 24]. A peculiarity of many peroxisomal proteins is that most of them have one or more consensus motifs known as Peroxisomal Targeting Signals (PTSs) [25–29]. Specific receptors recognize a PTS and bind to a region of the peroxisomal protein [25], thus allowing the peroxisomal protein import process. The known PTSs are as follows:

1. **PTS1.** The PTS1 receptor is encoded by the PEX5 gene [26] and is defined as the final dodecamer with a focus on the terminal tripeptide [27];
2. **PTS2.** It is an N-terminal targeting signal, and its receptor is encoded by the PEX7 gene (a co-receptor is also involved in peroxisomal import) [28];
3. **mPTS.** It is a cis-acting targeting signal specific for peroxisomal membrane proteins. The C-terminal domain of the Peroxisomal membrane protein import receptor PEX19 (Pex19), known as the mPTS (PMP-targeting signal) binding site, forms a three-helical bundle and plays a crucial role in the assembly and maintenance of the membrane [30]. It acts as a receptor and chaperone for peroxisomal membrane proteins (PMPs) [29–31].

In Figure 1.1, it is shown a simplified representation of the known PTSs.

Novel use-case-specific computational methods, combined with more established computational algorithms (such as those relying on Peroxisome Targeting Signal (PTS) and membrane Peroxisome Targeting Signals (mPTS) motif detection) and experimental methodologies, can facilitate the discovery and annotation of new peroxisomal proteins. This is particularly useful when investigating sequences from entire proteomes [24, 32].

#### Peroxisomal protein functions

Due to their significant participation in numerous metabolic and non-metabolic processes, as elaborated in section 1.1.2 and supported by references [4, 5], it can be inferred that peroxisomal proteins are associated with a wide range of functions [33]. However, despite ongoing research efforts, there remain numerous unidentified peroxisomal proteins and their corresponding functions [5, 19]. Recently, systematic,

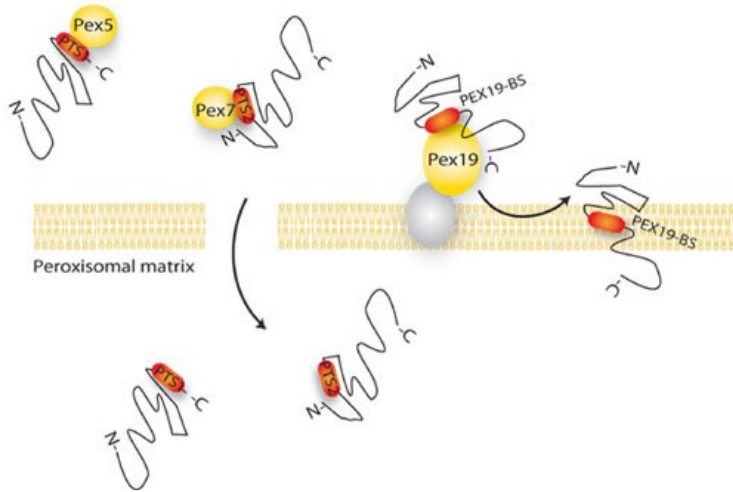


Figure 1.1: Schematic representation of the known Peroxisome Targeting Signals (PTSs). PTS1 binds to the Pex5 protein thus allowing protein import inside the peroxisome. PTS2 binds to the Pex7 protein, also allowing protein import inside the peroxisomal matrix. The Pex19 Binding Site (Pex19-BS) instead is essential for attaching peroxisomal membrane proteins to the membrane. The image is retrieved from the PeroxisomeDB webserver (<http://www.peroxisomedb.org/>)

whole-organelle proteome assays have provided new insights into the functions of the peroxisomal proteome [33]. In Yifrach *et al.* (2022), 33 newly verified peroxisomal proteins were identified in yeast, expanding the current protein count of peroxisomes by approximately 40% [33]. These newly identified proteins include enzymes and putative enzymes, structural and regulatory proteins, and uncharacterized proteins of unknown function. This indicates the need for further investigations, and the development of novel computational approaches could be essential for this purpose [24, 34, 35]. An example of possible peroxisomal protein functions in humans is shown in Table 1.1.

### Peroxisomal proteins interactions

The essential role of peroxisomes in eukaryotic cells is only possible through continued interaction with other subcellular organelles [16]. These interactions occur at Membrane Contact Sites (MCs) [36]. The list of interactive organelles includes lipid droplets, lysosomes, the endoplasmic reticulum [37], and mitochondria [38]. An overview of peroxisomes as interaction hubs can be seen in Figure 1.2 from Schrader *et al.* (2019) [16]. The metabolic interaction between peroxisomes and other organelles is often mediated by tethering proteins that bring the organelles physically closer together, facilitating efficient transfer of metabolites [13]. Given the high complexity and the extent of interactions between peroxisomes and other organelles, it is essential to discover new peroxisomal proteins, their functions, and their interactions. This underscores the importance of the algorithms explained in this thesis.

Table 1.1: Example of ten peroxisomal proteins in humans with their UniProt ID, Protein Name, Gene Name, and related function description. The table represents the variety of functions of the peroxisomal proteins.

UniProt ID	Protein Name	Gene Name	Function
O60683	Peroxisome biogenesis factor 10	PEX10	Protein ligase component of a retro-translocation channel required for peroxisome organization by mediating export of the PEX5 receptor from peroxisomes to the cytosol, thereby promoting PEX5 recycling
O75381	Peroxisomal membrane protein PEX14	PEX14	Component of the PEX13-PEX14 docking complex, a translocon channel that specifically mediates the import of peroxisomal cargo proteins bound to PEX5 receptor
O75521	Enoyl-CoA delta isomerase 2	ECI2	Metabolic role. Able to isomerize both 3-cis and 3-trans double bonds into the 2-trans form in a range of enoyl-CoA species. Has a preference for 3-trans substrates
O95573	Fatty acid CoA ligase Acsl3	ACSL3	Acyl-CoA synthetases (ACSL) activate long-chain fatty acids for both synthesis of cellular lipids, and degradation via beta-oxidation
O96011	Peroxisomal membrane protein 11B	PEX11B	Involved in peroxisomal proliferation
P01189	Pro-opiomelanocortin	COLI	Stimulates the adrenal glands to release cortisol
P08670	Vimentin	VIME	Intermediate filament involved in the stabilization of type I collagen mRNAs
P0C024	Peroxisomal coenzyme A diphosphatase NUDT7	NUDT7	Fatty acyl-coenzyme A (CoA) diphosphatase
P12268	Inosine-5'-monophosphate dehydrogenase 2	IMDH2	Catalyzer involved in <i>de novo</i> synthesis of guanine nucleotides. Plays an important role in the regulation of cell growth
P14735	Insulin-degrading enzyme	IDE	Plays a role in the cellular breakdown of insulin and other peptides. Plays a role in intercellular peptide signaling

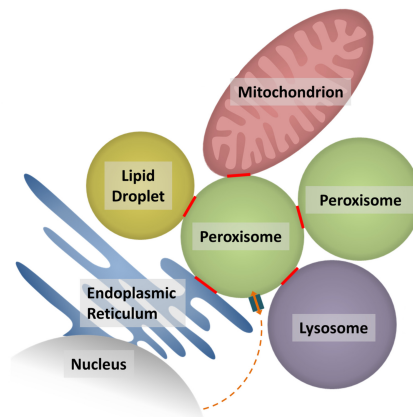


Figure 1.2: Peroxisome-organelle MCs in mammalian cells and their suggested functions. Communication between Peroxisome and the nucleus is also indicated. Figure adapted from Schrader *et al.* (2019).



## 1.2 Peroxisomal proteins function, localization and interactions determination

Peroxisomal proteins can be studied adopting different approaches which include both experimental and computational methods. Among the experimental methods that can be used for studying peroxisomal protein localization, interactions, and functions are immunofluorescence microscopy, which involves using specific antibodies to visualize peroxisomal proteins within cells, and subcellular fractionation that isolates peroxisomes to assess protein localization [39]. Co-immunoprecipitation (Co-IP) can be employed for identifying protein-protein interactions [40]. Other techniques include proximity labeling techniques such as BioID [41] or APEX [42] for identifying proteins in close proximity to bait proteins in peroxisomes, yeast two-hybrid assay [43] for detecting protein-protein interactions, fluorescence resonance energy transfer (FRET) [44] for studying protein-protein interactions in living cells, functional assays including enzyme activity assays and metabolite analysis to investigate peroxisomal protein functions [45, 46], and genetic approaches such as gene knockout [47], knockdown [48], or overexpression [49] to study the impact of specific genes on peroxisome biology [50].

Computational approaches for studying peroxisomal protein localization, interactions, and functions are less explored, or in general not re-adapted for the peroxisomal protein use case. As an example, computational methods can be based on sequence analysis [24, 34, 51], homology modeling [52, 53], protein-protein interaction prediction [54], pathway and network analysis [55], and molecular dynamics simulations [56].

Sequence analysis involves analyzing protein sequences to predict subcellular localization signals or targeting motifs indicative of peroxisomal localization [24, 51]. Homology modeling, in its simplest form, entails predicting the structure of a protein (e.g. peroxisomal protein) with only its sequence information by aligning it with a related protein whose structure is already known [53]. Protein-protein interaction prediction algorithms identify potential interactions (e.g. involving peroxisomal proteins) [54]. Pathway and network analysis explores metabolic pathways and protein interaction networks associated with peroxisome biology [55]. Molecular dynamics simulations investigate the dynamic behavior of peroxisomal proteins at the atomic level, revealing conformational changes, protein-lipid interactions, and ligand binding processes [56].

By employing these computational approaches, researchers complement experimental studies and gain a deeper understanding of peroxisomal protein localization, interactions, and functions in a systematic and efficient manner. However, despite of the recent advancements in Artificial Intelligence (AI), there is still a lack of computational methods tailored to peroxisomal research, thus the necessity of developing novel approaches such as the ones presented in this thesis.

### 1.3 An increasing interest: Artificial Intelligence in biomedical sciences

The pioneering Artificial Intelligence works can be traced back approximately to the mid-20th century, with the first step being the development of the electronic computer. In 1950, Alan Turing proposed the famous Turing Test as a measure of a machine's ability to exhibit intelligent behavior that is indistinguishable from that of a human [57]. Seven years later we had the first Perceptron, a type of linear classifier, (i.e. a classification algorithm based on a linear function which combines a set of weights with the feature vector), able to recognize simple visual patterns and was the basis for future advancements in machine learning (ML) and the development of more complex neural networks (NNs) architectures [58].

From 1957 to 1974, AI flourished. Computers could store more information and became faster, cheaper, and more accessible. As scientists started selecting appropriate algorithms for their specific problems, machine learning algorithms also advanced. Early AI demonstrations, such as Newell and Simon's General Problem Solver and Joseph Weizenbaum's ELIZA [59], were encouraging, particularly in regards to problem solving and spoken language interpretation. These accomplishments, combined with the support of prominent researchers, persuaded government agencies like Defense Advanced Research Projects Agency (DARPA) to invest in AI research across various institutions.

ELIZA [59] can be considered a pioneer work in Natural Language Processing (NLP), which is a field of AI concerned with the interaction between computers and human languages [60]. The primary goal of NLP is to enable machines to understand natural language text or speech as humans do. This involves the use of various techniques such as statistical modeling and ML, to develop algorithms and models that can automatically extract meaning from unstructured text. NLP finds applications in a wide range of areas such as machine translation, sentiment analysis, text summarization, speech recognition, and information retrieval, among others[60].

A boost to the AI-related research took place with the introduction of algorithms that had multiple layers of non-linear features. These works were first introduced by the Group Method of Data Handling in 1965 [61] and paved the way to what is nowadays known as Deep Learning (DL) [62]. They were in fact the first DL systems of the feed-forward multilayer perceptron type [62–64]. The principle behind these type of DL architectures is to use deep models (with several layers) with polynomial activation functions. In each layer, the best features are selected through statistical methods and forwarded to the next layer [61, 63, 64]. Schematic representations of the perceptron, the feed-forward neural network (FNN) and the feed-forward multilayer perceptron architectures are shown in Figure 1.3.

After the pioneering works of Ivakhnenko [61, 62] several DL architectures were developed. Here a list of the ones treated in this thesis:

**Recurrent neural network (RNN).** RNNs are a variation to feed-forward networks where connections between nodes can generate cycles, thus allowing output from some nodes to affect subsequent input to the same node, thus allowing a temporal dynamic behavior [65, 66].

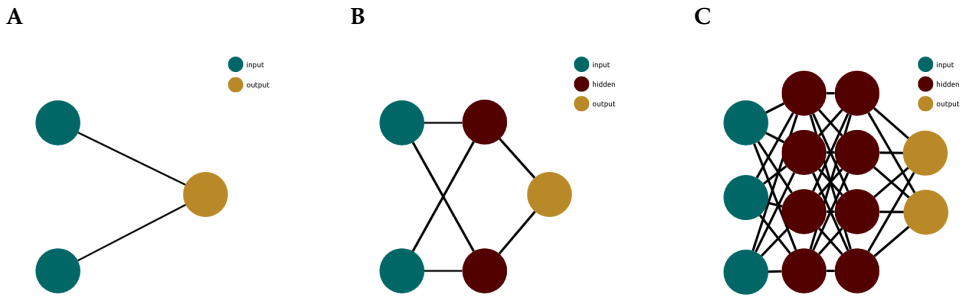


Figure 1.3: **A.** Perceptron. This neural network contains only two layers: input (petrol) and output (ocra). **B.** Feed-forward neural network. All of the perceptrons include in the architecture are arranged in inputs layers and output layers; every perceptron is connected with each node in the next layer. This structure also present hidden layers (bordeaux) which have no connection with the outer world. **C.** Multilayer Perceptron. It is a feed-forward network that uses more than one hidden layer.

**Long short-term memory (LSTM).** The term "LSTM" is derived from its resemblance to a standard RNN's ability to retain both "long-term memory" and "short-term memory". The primary objective of the LSTM architecture is to provide RNNs with a short-term memory that can persist over thousands of timesteps, hence the name "long short-term memory" [67].

**Bidirectional recurrent neural networks (BRNN).** This architecture establishes connections between two hidden layers that operate in opposite directions and share the same output. This enables the output layer to access information from both past (backwards) and future (forward) states concurrently [68].

**Transformers.** The transformer model uses self-attention technique [69]. In particular, it assigns varying levels of importance to different components of the input data (including the recursive output). Unlike RNNs, transformers process the entire input simultaneously. The attention mechanism enables the model to consider the context of any position within the input sequence. For instance, when dealing with a natural language sentence as input, the transformer is not constrained to processing one word at a time. This characteristic allows for greater parallelization compared to RNNs, leading to reduced training times [69–71].

The trend of publications related to Artificial Intelligence, Natural Language Processing, Machine Learning, and Deep Learning has been on a steady rise since 1980. From a modest 90 publications by the end of 1985, the number of publications has grown consistently year by year, reaching a peak of 103,962 publications in 2023. It is relevant to note that in a time range of 2 years and 3 months (2020-2023 31st March) the number of publications increased more than two times with respect to the time frame 2015-2020. This trend reflects the increasing importance of these fields in the development of modern technology and the interest they have

garnered in the academic community. These publications include papers with titles or abstracts that mention the aforementioned topics, indicating a growing interest in these areas of research from PubMed (<https://pubmed.ncbi.nlm.nih.gov/>). These data are visualized in Figure 1.4. These numbers highlight the relevance that AI development is having in biomedical sciences.

## 1.4 AI in bioinformatics

In Bioinformatics, the applications of deep learning is becoming common practice [72]. For example in domains like the various omics (e.g. genomics), biomedical imaging, and biomedical signal processing, biological molecule sequencing, we can see various applications of deep learning, such as gene expression regulation, protein structure prediction, cancer diagnosis and prognosis, drug discovery, and medical image analysis RNNs ([72–76]. An example is visible in the synergy between AI and the CRISPR technologies applied to vaccine design, therapeutic treatment improvement and RNA guide activities [74, 75, 77].

In recent years, we had the breakthrough of AlphaFold, a deep learning NN developed by Google’s DeepMind, which demonstrated to be a significant milestone in the field of protein folding [78]. AlphaFold uses cutting-edge artificial intelligence techniques to predict the three-dimensional structure of a protein based solely on its amino acid sequence with remarkable accuracy. This breakthrough is particularly significant because it had long been thought impossible to predict a protein’s structure with such precision based solely on its sequence [78].

Given the increasing amount of sequences available for the scientific community, which is proportional to the cost drop in generating DNA sequences and protein sequences [79–81], it is essential to focus our effort on how to analyse, integrate, process and extract knowledge out of this large data sets. UniProt release 2022\_03 contained over 227 million sequence records in UniProtKB [81]. On 13-04-2023 on the Uniprot web-server were present 246,440,937 sequences thus proving a continuous increase. In addition, the amount of automatically generated sequences compared to the manually reviewed ones, is shown in Figure 1.5. This increment requires the development of novel strategy to digest the amount of data, thus the application of DL algorithms in sequence analysis [24, 34, 82–85] as explained in the next sections and in the chapters 3,4,5 and 7.

## 1.5 DL-based embeddings for biological sequences

Protein embeddings are ways to encode functional and structural properties of a protein from its sequence, in a vectorized representation (<https://www.uniprot.org/help/embeddings>). ML and DL models trained on protein sequences can generate these embeddings, which can capture biological properties [82, 83]. These embeddings can then be used to infer these properties in unseen sequences without relying on the knowledge of underlying physicochemical or biological mechanisms [86].

Transformer technologies (e.g. ChatGPT) [69–71] that use text files, as well as embedding generator architectures for biological sequences (e.g. protein sequences), share a common approach in processing and analyzing text data. They both utilize

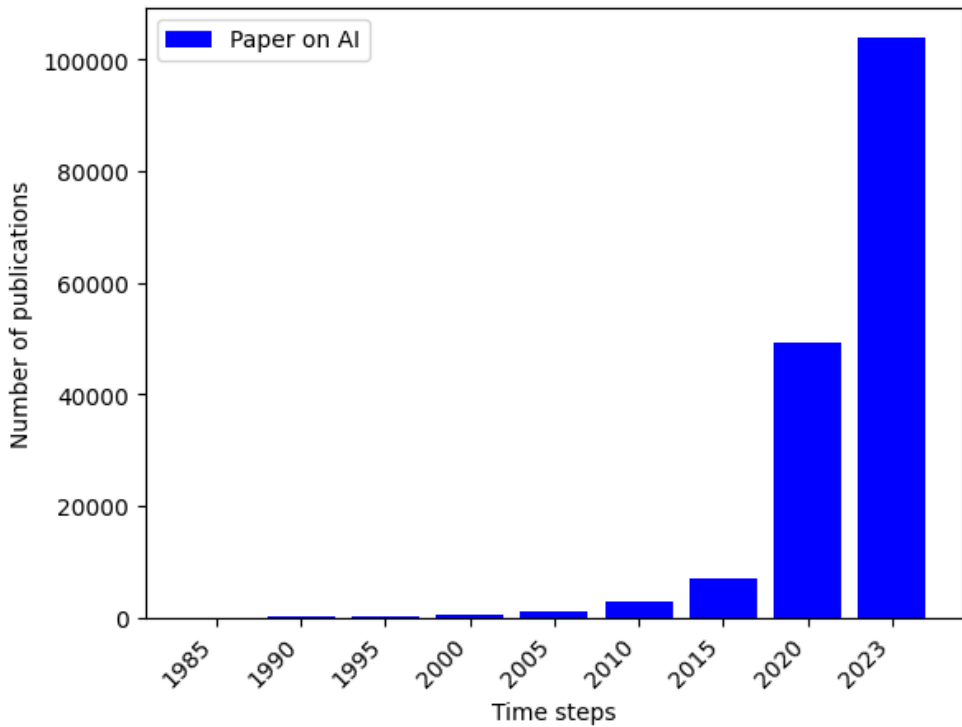


Figure 1.4: The trend of the number of papers published in a specific time range mentioning one of the following words: 'Artificial Intelligence', 'Natural Language Processing', 'Machine Learning' or 'Deep Learning'

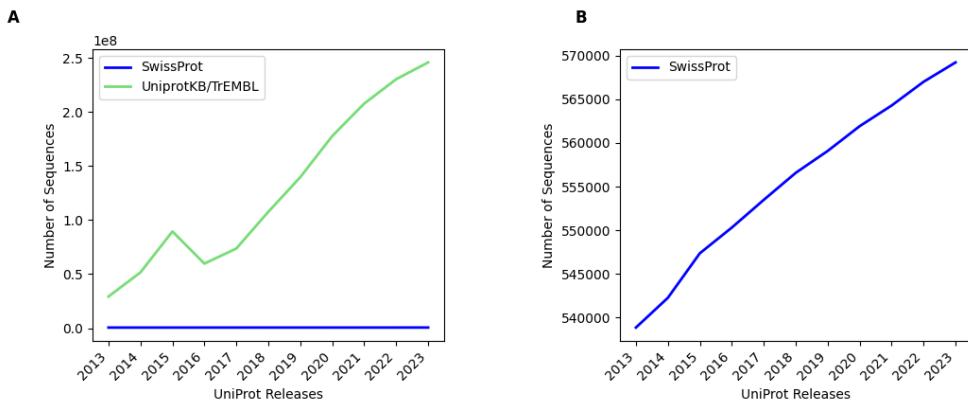


Figure 1.5: **A.** Trend of the number of sequences available against the UniProt release version. The blue line represent the sequences from SwissProt (manually curated). The green line represent the sequences available in the whole UniProtKB. Comparing the two trends on this scale (up to 250 millions) we see a constant number of manually reviewed protein sequences. **B.** The line shows the slow incremental growth of the sequences available on SwissProt.

DL techniques to process sequences of symbols and transform them into meaningful representations [70, 71, 84, 85, 87–92]. In the case of Transformer technology and other DL approaches applied to NLP, text files are fed into NNs and transformed into embeddings, which are numerical representations of the original text data [70, 71, 87–93]. These embeddings can then be used for various downstream tasks such as text classification or language generation [70, 71, 87–93].

Similarly, in the case of biological sequences, for example in protein sequences, the primary structure of proteins can be represented as a sequence of amino acids, which can be encoded as text data. These sequences can then be processed by NNs and transformed into embeddings, which can be used for tasks such as predicting protein structure, localization or function [24, 34, 82–85]. For clarification, we can see a text document as a biological sequence, where the words in the text are like nucleic acid basis or amino acids in a biological sequence. Practically, we would input a FASTA sequence which for a computer is no different than a text file. That would be embedded in a numerical representation and then processed by the classification algorithms [24, 34]. Overall, these technologies demonstrate the power of deep learning and NNs in processing and analyzing complex sequences of data, whether they are text documents or biological sequences.

Five recently proposed methods for the embedding of protein sequences based on deep-learning approaches are presented in this thesis, namely: The Unified Representation (UniRep)[94], The Sequence-to-Vector embedding (SeqVec)[95], Protein-BERT (PROTBERT)[96], The Evolutionary Scale Modelling - 1b (ESM-1b)[85], The Evolutionary Scale Modelling - 2 (ESM2) [97]. These embeddings are explained in the sections below and a summary is shown in Table 1.2.

### 1.5.1 The Unified Representation (UniRep)

UniRep [94] is based on a 1900-hidden unite RNN architecture, able to capture evolutionary, chemical and biological information encoded in the protein sequence starting from 24 million UniRef50 sequences [98] where UniRef50 is a non-redundant sub-cluster of Uniprot [99]. In UniRep, the protein sequence is modelled by using a hidden state vector, which is recursively updated based on the previous one. The method learns by scanning a sequence of amino acids, predicting the next one based on the sequence it has seen before. Using UniRep, a protein sequence can be represented by an embedding with a length of 64, 256, or 1900 units, depending on the NNs architecture.

#### UniRep application example 1 - Proteochemometric modeling

Proteochemometric modeling is a method used in drug discovery to predict the binding affinity of small molecules with proteins [100]. The use of UniRep has shown promising results in generating embeddings that outperform classical human-engineered representations [101]. In Kim et al. 2021, UniRep [82] and other DL-based embeddings [83, 85], where used to represent 1226 unique human proteins from a large-scale benchmark PCM dataset created in Lenselink et al. [102]. This dataset comprises 310 k compound–protein bioactivity measurements taken exclusively from the highest-confidence bioactivity assay data in ChEMBL [103]. Results were also reported on a large dataset of protein–ligand binding activities, con-

taining 500 k unique compounds and 1 k unique proteins [101]. The hypothesis that descriptors generated via unsupervised representation learning are more powerful than handcrafted protein and compound descriptors was supported by the results. It was observed that unsupervised descriptors consistently outperformed handcrafted ones across all splits and models, with a statistically significant difference.

### **UniRep application example 2 - Predicting stability of naturally occurring and de novo designed proteins.**

In the UniRep original paper [82], the performance of UniRep was compared to established approaches for predicting protein stability and function, trained on experimental data on top of other baseline representations, as well as the widely used Rosetta [104] structural stability prediction method. The Rosetta method relies on computational modeling and energy minimization algorithms to predict the three-dimensional structure of proteins based on sequence information and known protein structures[104]. Despite lacking explicit physical knowledge and structural data, UniRep Fusion outperformed Rosetta on rank order correlation with measured stability on a held-out test set, as well as on each fold topology subset. UniRep also outperformed all baseline models in the suite of experiments conducted in the study.

## **1.5.2 The Sequence-to-Vector embedding (SeqVec)**

SeqVec [83] is obtained by training ELMo [105], on UniRef50 [98]. ELMo is a word embedding method that models both complex characteristics of word use (e.g., syntax and semantics) and how these vary across linguistic contexts. It consists of a 2-layer bidirectional LSTM [106] backbone pre-trained on a large text corpus. The SeqVec embedding can be obtained based on either a per-residue level (word level) or a per-protein level (sentence level). The per-residue level protein sequence embedding is informative in predicting the secondary structure or intrinsically disordered region. The per-protein level embedding is useful to predict subcellular localisation and to distinguish membrane-bound vs water-soluble proteins [95]. In per-protein level embedding, the protein sequence is represented by an embedding of length 1024.

### **SeqVec application example 1 - Cross-species Protein Function Prediction**

In van den Bent *et al.* (2021), A SeqVec-based molecular function prediction model was trained on one species and tested on several other species with varying evolutionary distance [107]. Seven well-annotated species from different evolutionary classes, phyla, and kingdoms were considered, including *Mus musculus* (Mouse), *Rattus norvegicus* (Rat), *Homo sapiens* (Human), *Danio rerio* (Zebrafish), *Caenorhabditis elegans* (C. elegans), *Saccharomyces cerevisiae* (Yeast), and *Arabidopsis thaliana* (A. thaliana). Mouse was selected as the training species, and the remaining species had increasing divergence time from Mouse, creating an "evolutionary staircase." The Mouse data was split into 8977 mouse training, 1801 mouse validation, and 1790 mouse test proteins for optimal tuning and assessing a classifier [107]. As results is has been demonstrated that SeqVec-based molecular function prediction (using

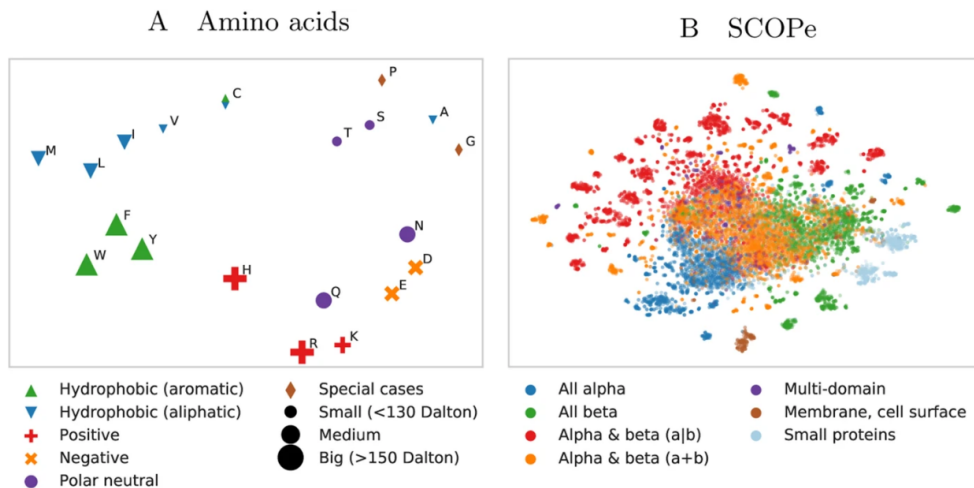


Figure 1.6: 2D t-SNE projections of unsupervised SeqVec embeddings highlight different realities of proteins and their constituent parts, amino acids in the UniRef50 data set. **A.** The embedding space confirms: the 20 standard amino acids are clustered according to their biochemical and biophysical properties, i.e. hydrophobicity, charge or size. The unique role of Cysteine (C, mostly hydrophobic and polar) is conserved. **B.** SeqVec embeddings capture structural information as annotated in the main classes in SCOPe without ever having been explicitly trained on structural features. Figure adapted from Heinzinger *et al.* (2019) [83]

a SeqVec-based Multilayer Perceptron classifier) can extract information from one well-annotated training species for predictions in various other eukaryotic species.

### SeqVec application example 2 - Per-residue level properties classification

An interpretable visualization of the SeqVec per-residue embedding can be done in a two dimensions projection using t-SNE [108]. The projection of the embedding space confirms that the model successfully captures bio-chemical and bio-physical properties on the 20 standard amino acids [83] as shown in Figure 1.6 (adapted from the original publication [83]). For example, aromatic amino acids (W, F, Y) are well separated from aliphatic amino acids (A, I, L, M, V) and small amino acids (A, C, G, P, S, T) are well separated from large ones (F, H, R, W, Y). Moreover, at the per-residue level, secondary structure and regions with intrinsic disorder were predicted significantly better than through one-hot encoding or through Word2vec-like approaches [83, 109].

### 1.5.3 ProteinBert (PROTBERT)

PROTBERT [96] is inspired by the Bidirectional Encoder Representations from Transformers (BERT) which is a deep learning model that utilizes a transformer architecture to pretrain on large amounts of unlabeled text data, enabling it to generate high-quality contextualized word representations for various NLP tasks [89]. PROTBERT was instead pretrained on the raw protein sequences available in Uniref100



(~106 million proteins) [89, 98]. The original BERT model is trained on two tasks: 1) language modelling where 15% of tokens are masked and the model predicts the masked tokens from context; 2) next sentence prediction where BERT is trained to predict the probability of a chosen next sentence given the first sentence. BERT learns contextual embeddings for words and can be finetuned on small data sets for optimized predictions on specific tasks [89]. In PROTBERT sequences are treated as separate documents, where the 'next' sentence prediction is not used. The masking procedure works by training randomly masked protein sequences, similar to the original BERT model. In particular, the model takes a sequence (sentence) as input, masks 15% of the amino acids (words) from it and is asked to output the complete sequence. ProteinBert was pretrained on two simultaneous tasks. 1) bidirectional language modelling of protein sequences 2) Gene Ontology (GO) annotation prediction, which captures diverse protein functions [110]. The final embedding has a length of 1024. For a detailed explanation of how to retrieve the ProteinBert embedding, we refer the reader to the specific GitHub [https://github.com/nadavbra/protein\\_bert](https://github.com/nadavbra/protein_bert) or the bio-embeddings repository [https://github.com/sacdallago/bio\\_embeddings](https://github.com/sacdallago/bio_embeddings).

### **PROTBERT application example 1 - Protein structure refinement**

In the recently published Deep Accuracy Network model (DeepAccNet), PROTBERT embeddings [96] were used along with several structural features to guide and improve protein structure refinement [111]. The DeepAccNet framework is designed to estimate, per-residue accuracy and residue-residue distance signed error in a protein model. Per-residue accuracy refers to the accuracy of each individual residue in a protein model. Residue-residue distance signed error refers to the difference between the predicted distance between two residues and their actual distance in the native protein structure. Both of these quantities for protein models and use them to guide Rosetta-based protein structure refinement [104]. In this approach, PROTBERT was used as a feature and its importance was analyzed by combining it with a distance map during training and analyzing the loss of predictions on a held-out test protein set. The addition of BERT features to DeepAccNet causes an improvement in the performance of the network in predicting per-residue accuracy and residue-residue distance signed error in protein models.

### **PROTBERT application example 2 - Evolutionary conservation profiles of proteins**

In Marquet *et al.* (2022), three classifiers were trained using PROTBERT (together with other embeddings) as input to predict family conservation based on nine conservation classes (1 to 9, where 9: highly conserved and 1: highly variable) introduced by ConSurf-DB [112]. In particular a non-redundant subset of proteins with experimentally known structures (data set ConSurf10k) was used. First, the capabilities of PROTBERT in reconstructing corrupted amino acid were tested. In the case of corrupting and reconstructing all residues in ConSurf10k, PROTBERT assigned a probability to the native amino acid and each of the 19 non-native amino acids for each position in the protein. Based on these "substitution probabilities," PROTBERT accurately predicted the native amino acid in 45.3% of cases. Secondly, Logistic Regression, FNNs, and standard convolutional neural network (CNN) [113–

115] models were trained using PROTBERT as input to predict family conservation based on the nine conservation classes. The authors demonstrated the capabilities and adaptability of these embeddings in predicting conserved regions.

### 1.5.4 The Evolutionary Scale Modelling - 1b (ESM-1b)

The Evolutionary Scale Modelling - 1b (ESM-1b) [85] was trained on 250 million sequences of the UniParc database [99] and relies on a transformer architecture [69, 116]. The peculiarity of the transformer architecture is that it is able to return for each amino acid (word) of the sequence (sentence), an embedding with contextual information. In other terms, it compares every amino acid (word) in the sequence (sentence) to every other amino acid (word) in the sequence (sentence), including itself, and reweighs the embeddings of each word. The modules responsible for this process are called self-attention blocks and consist of three main steps:

1. Dot product similarity and alignment scores;
2. Scores normalization and embedding weight;
3. Reweighting of the original embeddings.

In ESM-1b, the transformer processes inputs through a series of blocks that alternate self-attention with feed-forward connections. In this case, since it has been trained on proteins, the self-attention blocks construct pairwise interactions between all positions in the sequence, so that the transformer architecture represents residue-residue interactions. In addition, ESM-1b was trained using the masked language modelling objective [116] which forces the model to identify dependencies between the masked site and the unmasked parts of the sequence in order to make the prediction of the masked parts. Finally, the model was optimized scaling the identified hyperparameters to train a model with ~650 M parameters (33 layers) on the UR50/S data set, resulting in the ESM-1b Transformer [117]. The final length of the ESM-1b vector is 1280.

#### ESM-1b application example - Global protein homolog detection

ESM-1b proved to be a valid embedding in improving the global protein homolog detection with a consistent gain in protein function identification [118]. In Kilinc *et al.* (2023), has been developed a novel model called PROST combining ESM-1b [85] and Raimondi *et al.* (2018) model that utilize inverse direct cosine transform (iDCT) quantizations and dynamic time wrappings (DTW) to find protein similarities. Quantization is a method of reducing high-dimensional data to a low dimension representation. Quantization has important advantages over a larger dense layer, including not requiring further training and allowing for the use of the entire benchmark for testing. This results as PROST model being an alignment-free method that simply compares the embedding vectors for a high level of efficiency [118]. PROST was compared against CSBLAST [119], PHMMER [120], NCBI-BLAST [121], and FASTA scores [122] from a previous benchmarking paper [123]. An additional comparison was done against the homology detection method used in the ESM-1b paper [85]. PROST performs significantly better than other tools.

### 1.5.5 The Evolutionary Scale Modelling - 2 (ESM2)

Like for ESM-1b, the ESM2 model is a language model that uses evolutionary patterns linked to protein structure to predict high resolution protein structures directly from the primary protein sequence [97]. It eliminates the need for external evolutionary databases, multiple sequence alignments, and templates. The ESM2 model is trained over sequences in the UniRef protein sequence database [98]. During training, sequences are sampled with even weighting across approximately 43 million UniRef50 training clusters from approximately 138 million UniRef90 sequences so that over the course of training the model sees approximately 65 million unique sequences [98, 124]. The model generates state-of-the-art three-dimensional structure predictions while maintaining high resolution accuracy and results in a speed improvement for structure prediction of more than an order of magnitude. The ESM2 model is based on a BERT-style encoder-only transformer architecture with modifications [85, 89]. ESM2, as for the ESM-1b, was trained using the masked language modelling objective [116] which forces the model to identify dependencies between the masked site and the unmasked parts of the sequence in order to make the prediction of the masked parts [85]. The new family of transformer protein language models, ESM-2, was trained at scales from 8 million parameters up to 15 billion parameters. The main differences with the ESM-1b, are that ESM-1b had dropout both in hidden layers and attention which was removed completely to free up more capacity. Additionally, a query and key vector inside the self-attention with a sinusoidal embedding was introduced in ESM2, which improved model quality for small models. However, it was observed that the performance improvements start to disappear as the model size and training duration get bigger. The final length of the ESM2 vector is 1280.

#### ESM-1b and ESM2 application example - Estimation of sequence conservation for identifying functional sites

The work proposed by Yeung *et al.* (2023) [125] focuses on the alignment-free estimation of sequence conservation for identifying functional sites using protein sequence embeddings, and it is performed on a data set of multiple sequence alignments from the Pfam database [126]. In this work the authors benchmark several protein embeddings, and in particular, ESM-1b and ESM2 based on the Jensen-Shannon divergence score [127]. According to the benchmarks, the ESM2 protein language models provide the best performance-to-computational-cost ratio as shown in Figure 1.7 from the original publication [125]. Although larger models perform better, they require more computational resources. Among the ESM2 models (with different training parameters), it is observed that the increase in model size corresponds to a linear increase in performance.

## 1.6 Aim and outline of the Thesis

DL-based protein sequence embeddings can be used for various prediction tasks but their usage for specific use case scenarios (especially for peroxisome-related research) still needs to be validated. DL-based embeddings proved to be informative

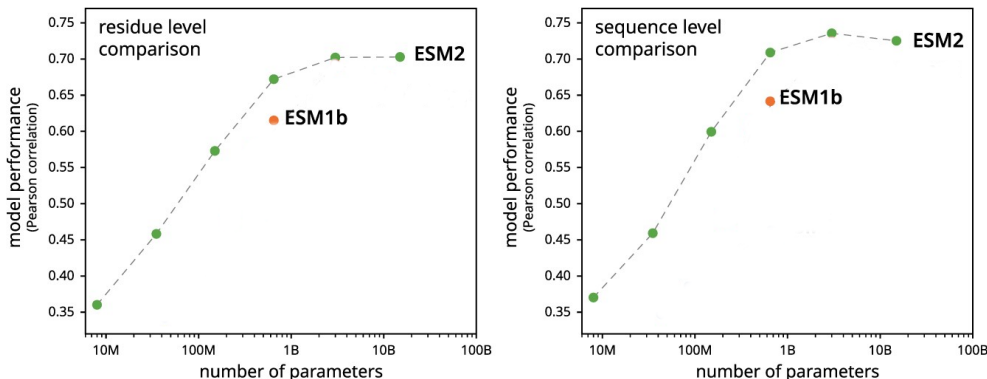


Figure 1.7: **A.** Comparison of DL-based protein embeddings at the individual residue level, showcasing the tradeoff between accuracy of conservation score predictions (measured by Pearson correlation) and computational resources required (number of model parameters). Each data point represents a distinct protein language model, with colors indicating models from the same family. Dotted lines connect models within the same family. **B.** Normalized performance comparison of DL-based protein embeddings by sequence, highlighting the relationship between accuracy of conservation score predictions (measured by Pearson correlation) and computational resources required (number of model parameters). Data points represent different protein language models, with colors denoting models from the same family.

Table 1.2: The protein sequence embeddings discussed in this thesis. The column "Embedding Name" shows the acronym of each embedding. The column "Architecture" lists the deep learning architecture behind each embedding. The column "Training Set" contains the subset of protein sequences from UniProtKB (<https://www.uniprot.org/>) used to train each embedding. The column "Vector Length" shows the length of the vector representing the embedded protein sequences that was used in the works presented in this thesis.

Embedding Name	Architecture	Training Set	Vector Length
UniRep	RNN	UniRef50	1900
SeqVec	LSTM	UniRef50	1024
PROTBERT	BRNN	UniRef100	1024
ESM-1b	Transformer	UniParc	1280
ESM2	Transformer	UniRef50/90	1280

in predicting the stability of natural and *de novo* designed proteins, in predicting the functionality of molecularly diverse mutants, in protein engineering, in protein structure prediction and in subcellular localization [82–84, 97, 128].

This thesis focuses on the capabilities of DL-based protein sequence embeddings for several tasks. In particular, their applicability for specific prediction algorithms (e.g. sub-organelle localization, transporter substrate prediction) is highlighted. The deployment of DL-based protein sequence embedding enabled to gain insights into Protein-Protein Interactions, Transporter substrate and function, and sub-organelle localization in the case of peroxisomal proteins [24, 34].

The necessity of investigating the peroxisomal protein use case, thus providing the scientific community with novel and accurate tools for peroxisomal research, is connected to the fact that well-known computational resources, such as the PeroxisomeDB, are of limited accuracy and/or are not updated with recent discoveries and the specific servers are often not functional [51]. Also, general computational methods can be useful only if re-adapted for specific use cases [129–134].

The **Introduction** provides contextualization for the research conducted in this thesis. It highlights the significance of Artificial Intelligence in Bioinformatics and Biomedical disciplines through descriptive statistics. The parallel between AI applied to text files and biological sequences is thoroughly explored, serving as the foundation for the thesis. DL-based sequence embeddings are introduced, showcasing their potential applications in subsequent chapters, including text file semantification.

Chapters 3-7 demonstrate the application of DL-based embeddings in specialized frameworks, enhancing computational approaches in peroxisomal-related research.

**Chapter 2** provides an overview of the computational methods and tools that can be used to investigate peroxisomal proteins and their functions. The chapter then discusses the limitations of these methods and highlights the need for more accurate and efficient approaches.

**Chapter 3** gives a detailed analysis of the In-Pero algorithm, which is a deep learning-based approach for predicting the localization of sub-peroxisomal proteins, specifically proteins that can be located inside the peroxisomal matrix or in the peroxisomal membrane. The chapter presents a detailed analysis of this algorithm, which combines protein-sequence embeddings with classical machine learning techniques to address the problem of predicting the sub-localization of peroxisomal proteins. Furthermore, the algorithm's performance is evaluated for sub-mitochondrial localization, resulting in the development of a highly effective predictor called In-Mito, surpassing most existing classifiers in accuracy.

**Chapter 4** presents the OrganelX e-Science Web Server, which is a user-friendly implementation of several deep learning-based algorithms for predicting sub-peroxisomal proteins localization. The novel Is-PTS1 algorithm, used to detect potential peroxisomal proteins carrying a PTS1 signal sequence, is also introduced. In this chapter, the reproducibility of the methods developed is highlighted in **Chapter 3**.

The procedure is extended and further applied in **Chapter 5**, which focuses on the detection of the membrane peroxisomal targeting signal (mPTS). These signals play a crucial role in the targeting of proteins to peroxisomes.

**Chapter 6** proposes a novel approach (in line with the ones presented in the other chapters) for protein-protein interaction prediction, where now the interaction is represented as a concatenation of the two embeddings representing the interacting protein sequences. Here, an algorithm called P-PPI is presented that is fine-tuned for peroxisomal protein-protein interaction prediction (thus the acronym). The chapter describes the usage of this algorithm to present potential candidates to be further checked either experimentally or with complementary computational approaches. In particular, it is reported a use case that highlights the synergy between P-PPI and the multimer predictor implemented within AlphaFold.

An additional application of DL-based sequence embeddings is presented in

**Chapter 7**, where it is shown the adaptation of the embeddings methods to predict transporter proteins and classify them according to their specific substrate. Here, the peroxisomal protein dataset is only used as proof of concept since the algorithm is perfectly scalable for every protein sequence.

To further provide practical benefits to the peroxisomal-research community, and in particular to the experimentalists, part of this thesis work focuses on developing semi-automatic a sematification system to upload and share biological assays.

In particular, **Chapter 8** and **Chapter 9** are examples of how similar technologies can be applied and tested for different inputs and topics. The change of topic here is strong but useful since it shows how a similar technology can be applied to different cases. The knowledge acquired from this application was then used to further improve the presented predictive algorithms.

**Chapter 8** focuses on the application of a SciBERT-based model (similar to DL-based sequence embeddings algorithms) for semantifying biological assays. The chapter begins by discussing the importance of representing biological assays (bioassays) in a standardized and machine-readable format to facilitate data integration and knowledge discovery.

**Chapter 9** further examines the SciBERT-based model for the semantification of bioassays by considering by comparing it to an approach which focuses on clustering bioassays before semantification. Surprisingly, the powerful transformer-based labeling method exhibited lower accuracy compared to the clustering solution (54% F1 vs. 83% F1), and labeling with a large set of labels took significantly more time due to per-label classifications.

Knowledge become valuable only if it can be passed to next generations. **Chapter 10** delves into the need for better promotion of bioinformatics and science for future generations, at the local level. The chapter revolves around the development of a teaching framework comprising a structured curriculum and a 5-day training school format, aimed at nurturing young scientists and raising awareness about the existing limitations in bioinformatics. The content of this thesis have been re-adapted to be included in the design of novel training schools.

The thesis ends with a **Discussion** section about the improvements that this set of scientific contributions brought to the scientific community. An in-depth discussion is presented about challenges and limitations of the current approaches adopted in artificial intelligence applied to biomedical sciences in general, and specifically for the representation of protein sequences and peroxisomal research. These limitations are also discussed within the training sessions explained in **Chapter 10**, where the trainers aims to raise awareness about these topics for the next generations of scientists from an early stage of their careers. This chapter presents an improved method for designing training schools, incorporating insights gathered from a pilot project and a survey conducted among young students.

The key contributions presented in the Thesis are outlined and visualized in the infographic in Figure 1.8.

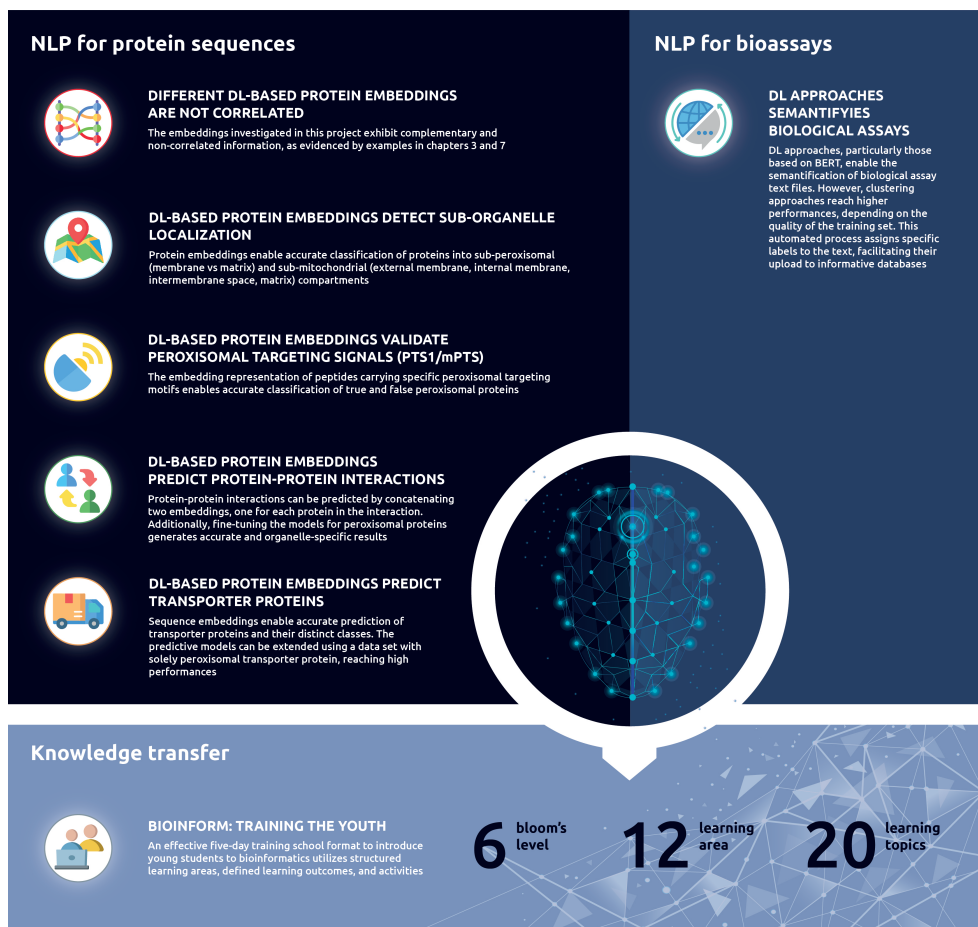


Figure 1.8: Key contributions of the Thesis. The infographic highlights two parallel lines of research where Natural Language Processing (NLP) is applied to protein sequences or biological assays (bioassays). NLP applied to protein sequences treats the sequences as text files. NLP approaches generate informative embeddings that can be used for several prediction tasks. The key contributions are as follows: 1) Different protein embeddings are not correlated; 2) Protein embeddings detect sub-organelle localization; 3) Different protein embeddings validate peroxisomal targeting signals; 4) Different protein embeddings predict protein-protein interactions; 5) Protein embeddings predict transporter proteins. On the other hand, NLP techniques, particularly deep learning (DL) models applied to biological assays, allow for the automatic semantification of bioassays, as shown in the associated key contribution titled 'DL approaches semantify biological assays' (top-right corner). At the bottom of the infographic, the last key contribution is presented as 'Bioinforming the youth.' It conceptualizes a five-day training school format focused on bioinformatics for young students.

## References

- [1] Rhodin, J. “Correlation of ultrastructural organization and function in normal and experimentally changed proximal convoluted tubule cells of the mouse kidney”. In: *Doctoral Thesis., Karolinska Institutet, Stockholm, Aktiebolaget Godvil 1* (1954).
- [2] Duve, C. d. “The peroxisome: a new cytoplasmic organelle”. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 173.1030 (1969), 71–83.
- [3] Smith, J. J. and Aitchison, J. D. “Peroxisomes take shape”. In: *Nature Reviews Molecular Cell Biology* 14.12 (2013), 803–817. DOI: 10.1038/nrm3700. URL: <https://doi.org/10.1038/nrm3700>.
- [4] Wanders, R. J. A., Waterham, H. R., and Ferdinandusse, S. “Metabolic Interplay between Peroxisomes and Other Subcellular Organelles Including Mitochondria and the Endoplasmic Reticulum”. In: *Frontiers in Cell and Developmental Biology* 3 (2016), 83.
- [5] Islinger, M. et al. “The peroxisome: an update on mysteries 2.0”. In: *Histochemistry and Cell Biology* 150 (2018), 1–29. DOI: 10.1007/s00418-018-1722-5.
- [6] Wanders, R. J. “Peroxisomal Disorders”. In: *Emery and Rimoin’s Principles and Practice of Medical Genetics*. Elsevier, 2013, 1–22. DOI: 10.1016/b978-0-12-383834-6.00110-5. URL: <https://doi.org/10.1016/b978-0-12-383834-6.00110-5>.
- [7] Jansen, R. L. M. et al. “Comparative Genomics of Peroxisome Biogenesis Proteins: Making Sense of the PEX Proteins”. In: *Frontiers in Cell and Developmental Biology* 9 (2021). DOI: 10.3389/fcell.2021.654163. URL: <https://doi.org/10.3389/fcell.2021.654163>.
- [8] Gurvitz, A. and Rottensteiner, H. “The biochemistry of oleate induction: Transcriptional upregulation and peroxisome proliferation”. In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006), 1392–1402. DOI: 10.1016/j.bbamcr.2006.07.011. URL: <https://doi.org/10.1016/j.bbamcr.2006.07.011>.
- [9] Eckert, J. H. and Erdmann, R. “Peroxisome biogenesis”. In: *Reviews of Physiology, Biochemistry and Pharmacology*. Springer Berlin Heidelberg, 75–121. DOI: 10.1007/s10254-003-0007-z. URL: <https://doi.org/10.1007/s10254-003-0007-z>.
- [10] Kim, P. “Peroxisome Biogenesis: A Union between Two Organelles”. In: *Current Biology* 27.7 (2017), R271–R274. DOI: 10.1016/j.cub.2017.02.052. URL: <https://doi.org/10.1016/j.cub.2017.02.052>.
- [11] Farré, J.-C. et al. “Peroxisome biogenesis, membrane contact sites, and quality control”. In: *EMBO reports* 20.1 (2018). DOI: 10.15252/embr.201846864. URL: <https://doi.org/10.15252/embr.201846864>.
- [12] Wanders, R. J. “Metabolic functions of peroxisomes in health and disease”. In: *Biochimie* 98 (2014), 36–44. DOI: 10.1016/j.biochi.2013.08.022. URL: <https://doi.org/10.1016/j.biochi.2013.08.022>.



- [13] Wanders, R. J. A. et al. "The physiological functions of human peroxisomes". In: *Physiological Reviews* 103.1 (2023), 957–1024. DOI: 10.1152/physrev.00051.2021. URL: <https://doi.org/10.1152/physrev.00051.2021>.
- [14] Lodhi, I. J. and Semenkovich, C. F. "Peroxisomes: A Nexus for Lipid Metabolism and Cellular Signaling". In: *Cell Metabolism* 19.3 (2014), 380–392. DOI: 10.1016/j.cmet.2014.01.002. URL: <https://doi.org/10.1016/j.cmet.2014.01.002>.
- [15] Ferdinandusse, S. and Houten, S. M. "Peroxisomes and bile acid biosynthesis". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006), 1427–1440. DOI: 10.1016/j.bbamcr.2006.09.001. URL: <https://doi.org/10.1016/j.bbamcr.2006.09.001>.
- [16] Schrader, M., Kamoshita, M., and Islinger, M. "Organelle interplay - peroxisome interactions in health and disease". In: *Journal of Inherited Metabolic Disease* 43.1 (2019), 71–89. DOI: 10.1002/jimd.12083. URL: <https://doi.org/10.1002/jimd.12083>.
- [17] Litwack, G. "Metabolism of Amino Acids". In: *Human Biochemistry*. Elsevier, 2018, 359–394. DOI: 10.1016/b978-0-12-383864-3.00013-2. URL: <https://doi.org/10.1016/b978-0-12-383864-3.00013-2>.
- [18] Fransen, M. et al. "Role of peroxisomes in ROS/RNS-metabolism: Implications for human disease". In: *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease* 1822.9 (2012), 1363–1373. DOI: 10.1016/j.bbadis.2011.12.001. URL: <https://doi.org/10.1016/j.bbadis.2011.12.001>.
- [19] Costa, C. F. et al. "The Mystery behind Hydrogen Peroxide Permeation across the Peroxisomal Membrane". In: *Peroxisporins: Redox Signal Mediators in and Between Cells*. Ed. by I. Medraño-Fernández, R. Sitia, and P. Bienert. CRC Press/Taylor & Francis, 2022.
- [20] Tiew, T. W.-Y., Sheahan, M. B., and Rose, R. J. "Peroxisomes contribute to reactive oxygen species homeostasis and cell division induction in Arabidopsis protoplasts". In: *Frontiers in Plant Science* 6 (2015). DOI: 10.3389/fpls.2015.00658. URL: <https://doi.org/10.3389/fpls.2015.00658>.
- [21] Breitling, R. "Pathogenesis of peroxisomal deficiency disorders (Zellweger syndrome) may be mediated by misregulation of the GABAergic system via the diazepam binding inhibitor". In: *BMC Pediatrics* 4.1 (2004). DOI: 10.1186/1471-2431-4-5. URL: <https://doi.org/10.1186/1471-2431-4-5>.
- [22] Kelley, R. I. et al. "Neonatal adrenoleukodystrophy: New cases, biochemical studies, and differentiation from Zellweger and related peroxisomal polydystrophy syndromes". In: *American Journal of Medical Genetics* 23.4 (1986), 869–901. DOI: 10.1002/ajmg.1320230404. URL: <https://doi.org/10.1002/ajmg.1320230404>.
- [23] Engelen, M. et al. "X-linked adrenoleukodystrophy (X-ALD): clinical presentation and guidelines for diagnosis, follow-up and management". In: *Orphanet Journal of Rare Diseases* 7.1 (2012), 51. DOI: 10.1186/1750-1172-7-51. URL: <https://doi.org/10.1186/1750-1172-7-51>.

- [24] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. "In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins". In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
- [25] Gould, S. G., Keller, G. A., and Subramani, S. "Identification of a peroxisomal targeting signal at the carboxy terminus of firefly luciferase." In: *Journal of Cell Biology* 105.6 (1987), 2923–2931.
- [26] Kiel, J. A. et al. "Ubiquitination of the Peroxisomal Targeting Signal Type 1 Receptor, Pex5p, Suggests the Presence of a Quality Control Mechanism during Peroxisomal Matrix Protein Import". In: *Journal of Biological Chemistry* 280.3 (2005), 1921–1930.
- [27] Brocard, C. and Hartig, A. "Peroxisome targeting signal 1: Is it really a simple tripeptide?" In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006), 1565–1573.
- [28] Kunze, M. "The type-2 peroxisomal targeting signal". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1867.2 (2020), 118609.
- [29] Van Ael, E. and Fransen, M. "Targeting signals in peroxisomal membrane proteins". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006). Peroxisomes: Morphology, Function, Biogenesis and Disorders, 1629–1638. DOI: <https://doi.org/10.1016/j.bbamcr.2006.08.020>. URL: <https://www.sciencedirect.com/science/article/pii/S0167488906002308>.
- [30] Schueller, N. et al. "The peroxisomal receptor Pex19p forms a helical mPTS recognition domain". In: *The EMBO Journal* 29.15 (2010), 2491–2500. DOI: 10.1038/emboj.2010.115. URL: <https://doi.org/10.1038/emboj.2010.115>.
- [31] Rottensteiner, H. et al. "Peroxisomal Membrane Proteins Contain Common Pex19p-binding Sites that Are an Integral Part of Their Targeting Signals". In: *Molecular Biology of the Cell* 15.7 (2004), 3406–3417. DOI: 10.1091/mbc.e04-03-0188. URL: <https://doi.org/10.1091/mbc.e04-03-0188>.
- [32] Kamoshita, M. et al. "Insights Into the Peroxisomal Protein Inventory of Zebrafish". In: *Frontiers in Physiology* 13 (2022).
- [33] Yifrach, E. et al. "Systematic multi-level analysis of an organelle proteome reveals new peroxisomal functions". In: *Molecular Systems Biology* 18.9 (2022). DOI: 10.15252/msb.202211186. URL: <https://doi.org/10.15252/msb.202211186>.
- [34] Anteghini, M. et al. "OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection". In: *Computational and Structural Biotechnology Journal* 21 (2022), 128–133. DOI: 10.1016/j.csbj.2022.11.058. URL: <https://doi.org/10.1016/j.csbj.2022.11.058>.
- [35] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. "PortPred: exploiting deep learning embeddings of amino acid sequences for the identification of transporter proteins and their substrates". In: (2023). DOI: 10.1101/2023.01.26.525714. URL: <https://doi.org/10.1101/2023.01.26.525714>.
- [36] Chen, C. et al. "Peroxisomal Membrane Contact Sites in Mammalian Cells". In: *Frontiers in Cell and Developmental Biology* 8 (2020). DOI: 10.3389/fcell.2020.00512. URL: <https://doi.org/10.3389/fcell.2020.00512>.

- 
- [37] Wu, F. and Klei, I. J. van der. "Structure - function analysis of the ER - peroxisome contact site protein Pex32". In: *Frontiers in Cell and Developmental Biology* 10 (2022). DOI: 10.3389/fcell.2022.957871. URL: <https://doi.org/10.3389/fcell.2022.957871>.
  - [38] Schrader, M. et al. "Peroxisome-mitochondria interplay and disease". In: *Journal of Inherited Metabolic Disease* 38.4 (2015), 681–702.
  - [39] Mandrell, R. E., Griffiss, J. M., and Macher, B. A. "Lipooligosaccharides (LOS) of *Neisseria gonorrhoeae* and *Neisseria meningitidis* have components that are immunochemically similar to precursors of human blood group antigens. Carbohydrate sequence specificity of the mouse monoclonal antibodies that recognize crossreacting antigens on LOS and human erythrocytes." In: *Journal of Experimental Medicine* 168.1 (1988), 107–126. DOI: 10.1084/jem.168.1.107. URL: <https://doi.org/10.1084/jem.168.1.107>.
  - [40] Lin, J.-S. and Lai, E.-M. "Protein-Protein Interactions: Co - Immunoprecipitation". In: *Methods in Molecular Biology*. Springer New York, 2017, 211–219. DOI: 10.1007/978-1-4939-7033-9\_17. URL: [https://doi.org/10.1007/978-1-4939-7033-9\\_17](https://doi.org/10.1007/978-1-4939-7033-9_17).
  - [41] Sears, R. M., May, D. G., and Roux, K. J. "BioID as a Tool for Protein-Proximity Labeling in Living Cells". In: *Methods in Molecular Biology*. Springer New York, 2019, 299–313. DOI: 10.1007/978-1-4939-9546-2\_15. URL: [https://doi.org/10.1007/978-1-4939-9546-2\\_15](https://doi.org/10.1007/978-1-4939-9546-2_15).
  - [42] Nguyen, T. M. T. et al. "APEX Proximity Labeling as a Versatile Tool for Biological Research". In: *Biochemistry* 59.3 (2019), 260–269. DOI: 10.1021/acs.biochem.9b00791. URL: <https://doi.org/10.1021/acs.biochem.9b00791>.
  - [43] Young, K. H. "Yeast Two-hybrid: So Many Interactions, (in) So Little Time..." In: *Biology of Reproduction* 58.2 (1998), 302–311. DOI: 10.1095/biolreprod58.2.302. URL: <https://doi.org/10.1095/biolreprod58.2.302>.
  - [44] Sekar, R. B. and Periasamy, A. "Fluorescence resonance energy transfer (FRET) microscopy imaging of live cell protein localizations". In: *Journal of Cell Biology* 160.5 (2003), 629–633. DOI: 10.1083/jcb.200210140. URL: <https://doi.org/10.1083/jcb.200210140>.
  - [45] Metz, J., Castro, I., and Schrader, M. "Peroxisome Motility Measurement and Quantification Assay". In: *BIO-PROTOCOL* 7.17 (2017). DOI: 10.21769/bioprotoc.2536. URL: <https://doi.org/10.21769/bioprotoc.2536>.
  - [46] Dahan, N. et al. "Current advances in the function and biogenesis of peroxisomes and their roles in health and disease". In: *Histochemistry and Cell Biology* 155.4 (2021), 513–524. DOI: 10.1007/s00418-021-01982-1. URL: <https://doi.org/10.1007/s00418-021-01982-1>.
  - [47] Yan, R. et al. "Peroxisome Proliferator-Activated Receptor Gene Knockout Promotes Podocyte Injury in Diabetic Mice". In: *BioMed Research International* 2022 (2022). Ed. by D. Rokaya, 1–8. DOI: 10.1155/2022/9018379. URL: <https://doi.org/10.1155/2022/9018379>.

- [48] Wei, X. et al. “Knockdown of PEX16 Induces Autophagic Degradation of Peroxisomes”. In: *International Journal of Molecular Sciences* 22.15 (2021), 7989. DOI: 10.3390/ijms22157989. URL: <https://doi.org/10.3390/ijms22157989>.
- [49] Bagattin, A., Hugendubler, L., and Mueller, E. “Transcriptional coactivator PGC-1 $\alpha$  promotes peroxisomal remodeling and biogenesis”. In: *Proceedings of the National Academy of Sciences* 107.47 (2010), 20376–20381. DOI: 10.1073/pnas.1009176107. URL: <https://doi.org/10.1073/pnas.1009176107>.
- [50] Chorny, S., Koster, J., and Waterham, H. R. “Applying CRISPR-Cas9 Genome Editing to Study Genes Involved in Peroxisome Biogenesis or Peroxisomal Functions”. In: *Methods in Molecular Biology*. Springer US, 2023, 233–245. DOI: 10.1007/978-1-0716-3048-8\_17. URL: [https://doi.org/10.1007/978-1-0716-3048-8\\_17](https://doi.org/10.1007/978-1-0716-3048-8_17).
- [51] Schlüter, A. et al. “PeroxisomeDB 2.0: an integrative view of the global peroxisomal metabolome”. In: *Nucleic Acids Research* 38.suppl\_1 (2009), D800–D805.
- [52] Muhammed, M. T. and Aki-Yalcin, E. “Homology modeling in drug discovery: Overview, current applications, and future perspectives”. In: *Chemical Biology & Drug Design* 93.1 (2018), 12–20. DOI: 10.1111/cbdd.13388. URL: <https://doi.org/10.1111/cbdd.13388>.
- [53] Greer, J. “Model for haptoglobin heavy chain based upon structural homology.” In: *Proceedings of the National Academy of Sciences* 77.6 (1980), 3393–3397. DOI: 10.1073/pnas.77.6.3393. URL: <https://doi.org/10.1073/pnas.77.6.3393>.
- [54] Canzler, S. et al. “ProteinPrompt: a webserver for predicting protein-protein interactions”. In: *Bioinformatics Advances* 2.1 (2022). Ed. by M. Gromiha. DOI: 10.1093/bioadv/vbac059. URL: <https://doi.org/10.1093/bioadv/vbac059>.
- [55] Gambardella, G. et al. “Differential network analysis for the identification of condition-specific pathway activity and regulation”. In: *Bioinformatics* 29.14 (2013), 1776–1785. DOI: 10.1093/bioinformatics/btt290. URL: <https://doi.org/10.1093/bioinformatics/btt290>.
- [56] Liu, X.-Y. et al. “Docking and Molecular Dynamics Simulations of Peroxisome Proliferator Activated Receptors Interacting with Pan Agonist Sodelglitazar”. In: *Protein & Peptide Letters* 18.10 (2011), 1021–1027. DOI: 10.2174/092986611796378701. URL: <https://doi.org/10.2174/092986611796378701>.
- [57] TURING, A. M. “I.—COMPUTING MACHINERY AND INTELLIGENCE”. In: *Mind* LIX.236 (1950), 433–460. DOI: 10.1093/mind/lix.236.433. URL: <https://doi.org/10.1093/mind/lix.236.433>.
- [58] Rosenblatt, F. “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological Review* 65.6 (1958), 386–408. DOI: 10.1037/h0042519. URL: <https://doi.org/10.1037/h0042519>.
- [59] Weizenbaum, J. “ELIZA—a computer program for the study of natural language communication between man and machine”. In: *Communications of the ACM* 9.1 (1966), 36–45. DOI: 10.1145/365153.365168. URL: <https://doi.org/10.1145/365153.365168>.

- 
- [60] Khurana, D. et al. "Natural language processing: state of the art, current trends and challenges". In: *Multimedia Tools and Applications* 82.3 (2022), 3713–3744. DOI: 10.1007/s11042-022-13428-4. URL: <https://doi.org/10.1007/s11042-022-13428-4>.
  - [61] Ivakhnenko, A. G. and Lapa, V. G. *Cybernetic Predicting Devices*. CCM Information Corporation, 1965.
  - [62] Farlow, S. J. "The GMDH Algorithm of Ivakhnenko". In: *The American Statistician* 35.4 (1981), 210. DOI: 10.2307/2683292. URL: <https://doi.org/10.2307/2683292>.
  - [63] LeCun, Y., Bengio, Y., and Hinton, G. "Deep learning". In: *Nature* 521.7553 (2015), 436–444. DOI: 10.1038/nature14539. URL: <https://doi.org/10.1038/nature14539>.
  - [64] Schmidhuber, J. "Deep learning in neural networks: An overview". In: *Neural Networks* 61 (2015), 85–117.
  - [65] Sutskever, I., Martens, J., and Hinton, G. "Generating Text with Recurrent Neural Networks". In: 2011, 1017–1024.
  - [66] Reczko, M. and Hatzigerrorgiou, A. "Prediction of the subcellular localization of eukaryotic proteins using sequence signals and composition". In: *PROTEOMICS* 4.6 (2004), 1591–1596.
  - [67] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
  - [68] Schuster, M. and Paliwal, K. "Bidirectional recurrent neural networks". In: *Signal Processing, IEEE Transactions on* 45 (1997), 2673–2681. DOI: 10.1109/78.650093.
  - [69] Vaswani, A. et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, 5998–6008.
  - [70] Radford, A. et al. "Language Models are Unsupervised Multitask Learners". In: 2019.
  - [71] Brown, T. B. et al. "Language Models are Few-Shot Learners". In: *ArXiv abs/2005.14165* (2020).
  - [72] Min, S., Lee, B., and Yoon, S. "Deep learning in bioinformatics". In: *Briefings in Bioinformatics* (2016), bbw068. DOI: 10.1093/bib/bbw068. URL: <https://doi.org/10.1093/bib/bbw068>.
  - [73] Zou, Q., Wan, S., and Zeng, X. "HPTree: Reconstructing phylogenetic trees for ultra-large unaligned DNA sequences via NJ model and Hadoop". In: *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2016, 53–58. DOI: 10.1109/BIBM.2016.7822492.
  - [74] Kim, H. K. et al. "Deep learning improves prediction of CRISPR-Cpf1 guide RNA activity". In: *Nature Biotechnology* 36.3 (2018), 239–241. DOI: 10.1038/nbt.4061. URL: <https://doi.org/10.1038/nbt.4061>.
  - [75] Maserat, E. "Integration of Artificial Intelligence and CRISPR/Cas9 System for Vaccine Design". In: *Cancer Informatics* 21 (2022), 117693512211401. DOI: 10.1177/11769351221140102. URL: <https://doi.org/10.1177/11769351221140102>.

- [76] Gakii, C., Mireji, P. O., and Rimiru, R. “Graph Based Feature Selection for Reduction of Dimensionality in Next-Generation RNA Sequencing Datasets”. In: *Algorithms* 15.1 (2022), 21. DOI: 10.3390/a15010021. URL: <https://doi.org/10.3390/a15010021>.
- [77] Bhat, A. A. et al. “Integration of CRISPR/Cas9 with artificial intelligence for improved cancer therapeutics”. In: *Journal of Translational Medicine* 20.1 (2022). DOI: 10.1186/s12967-022-03765-1. URL: <https://doi.org/10.1186/s12967-022-03765-1>.
- [78] Jumper, J. et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), 583–589. DOI: 10.1038/s41586-021-03819-2. URL: <https://doi.org/10.1038/s41586-021-03819-2>.
- [79] Alfaro, J. A. et al. “The emerging landscape of single-molecule protein sequencing technologies”. In: *Nature Methods* 18.6 (2021), 604–617. DOI: 10.1038/s41592-021-01143-1. URL: <https://doi.org/10.1038/s41592-021-01143-1>.
- [80] Wetterstrand, K. *DNA sequencing costs: Data*. URL: <https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data>.
- [81] Alex Bateman, and et al. “UniProt: the Universal Protein Knowledgebase in 2023”. In: *Nucleic Acids Research* 51.D1 (2022), D523–D531. DOI: 10.1093/nar/gkac1052. URL: <https://doi.org/10.1093/nar/gkac1052>.
- [82] Alley, E. et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16 (2019). DOI: 10.1038/s41592-019-0598-1.
- [83] Heinzinger, M. et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20 (2019).
- [84] Elnaggar, A. et al. “ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing”. In: *bioRxiv* (2020).
- [85] Rives, A. et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118. DOI: 10.1073/pnas.2016239118. URL: <https://doi.org/10.1073/pnas.2016239118>.
- [86] Yang, K. K. et al. “Learned protein embeddings for machine learning”. In: *Bioinformatics* 34.15 (2018). Ed. by J. Wren, 2642–2648. DOI: 10.1093/bioinformatics/bty178. URL: <https://doi.org/10.1093/bioinformatics/bty178>.
- [87] Clark, K. et al. “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators”. In: *ArXiv abs/2003.10555* (2020).
- [88] Rahali, A. and Akhloufi, M. A. “End-to-End Transformer-Based Models in Textual-Based NLP”. In: *AI* 4.1 (2023), 54–110. DOI: 10.3390/ai4010004. URL: <https://doi.org/10.3390/ai4010004>.
- [89] Devlin, J. et al. “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)*. 2019, 4171–4186.

- 
- [90] Beltagy, I., Lo, K., and Cohan, A. “SciBERT: Pretrained Language Model for Scientific Text”. In: *EMNLP*. 2019.
  - [91] Lan, Z. et al. “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations”. In: *ArXiv abs/1909.11942* (2019).
  - [92] Raffel, C. et al. “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *ArXiv abs/1910.10683* (2019).
  - [93] Liu, Y. et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *ArXiv abs/1907.11692* (2019).
  - [94] Alley, E. C. et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16.12 (2019), 1315–1322. DOI: 10.1038/s41592-019-0598-1. URL: <https://doi.org/10.1038/s41592-019-0598-1>.
  - [95] Heinzinger, M. et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20 (2019).
  - [96] Brandes, N. et al. “ProteinBERT: A universal deep-learning model of protein sequence and function”. In: *bioRxiv* (2021). DOI: 10.1101/2021.05.24.445464.
  - [97] Lin, Z. et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), 1123–1130. DOI: 10.1126/science.ade2574. URL: <https://doi.org/10.1126/science.ade2574>.
  - [98] Suzek, B. E. et al. “UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches”. In: *Bioinformatics* 31.6 (2014), 926–932.
  - [99] Consortium, T. U. “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
  - [100] Westen, G. J. P. van et al. “Proteochemometric modeling as a tool to design selective compounds and for extrapolating to novel targets”. In: *MedChemComm* 2.1 (2011), 16–30. DOI: 10.1039/c0md00165a. URL: <https://doi.org/10.1039/c0md00165a>.
  - [101] Kim, P. T., Winter, R., and Clevert, D.-A. “Unsupervised Representation Learning for Proteochemometric Modeling”. In: *International Journal of Molecular Sciences* 22.23 (2021), 12882. DOI: 10.3390/ijms222312882. URL: <https://doi.org/10.3390/ijms222312882>.
  - [102] Lenselink, E. B. et al. “Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set”. In: *Journal of Cheminformatics* 9.1 (2017). DOI: 10.1186/s13321-017-0232-0. URL: <https://doi.org/10.1186/s13321-017-0232-0>.
  - [103] Mendez, D. et al. “ChEMBL: towards direct deposition of bioassay data”. In: *Nucleic Acids Research* 47.D1 (2018), D930–D940. DOI: 10.1093/nar/gky1075. URL: <https://doi.org/10.1093/nar/gky1075>.
  - [104] Ovchinnikov, S. et al. “Protein structure prediction using Rosetta in CASP12”. In: *Proteins: Structure, Function, and Bioinformatics* 86 (2017), 113–121. DOI: 10.1002/prot.25390. URL: <https://doi.org/10.1002/prot.25390>.

- [105] Peters, M. E. et al. “Deep contextualized word representations”. In: *Proc. of NAACL*. 2018.
- [106] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [107] Bent, I. van den, Makrodimitris, S., and Reinders, M. “The Power of Universal Contextualized Protein Embeddings in Cross-species Protein Function Prediction”. In: *Evolutionary Bioinformatics* 17 (2021), 117693432110626. DOI: 10.1177/11769343211062608. URL: <https://doi.org/10.1177/11769343211062608>.
- [108] Hinton, G. and Maaten, L. van der. “Visualizing data using t-SNE”. In: *J. Mach. Learn. Res.* 9.Nov (2008), 2579–2605.
- [109] Mikolov, T. et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. NIPS’13*. Lake Tahoe, Nevada: Curran Associates Inc., 2013, 3111–3119.
- [110] Ashburner, M. et al. “Gene Ontology: tool for the unification of biology”. In: *Nature Genetics* 25.1 (2000), 25–29. DOI: 10.1038/75556. URL: <https://doi.org/10.1038/75556>.
- [111] Hiranuma, N. et al. “Improved protein structure refinement guided by deep learning based accuracy estimation”. In: *Nature Communications* 12.1 (2021). DOI: 10.1038/s41467-021-21511-x. URL: <https://doi.org/10.1038/s41467-021-21511-x>.
- [112] Chorin, A. B. et al. “ConSurf-DB: An accessible repository for the evolutionary conservation patterns of the majority of PDB proteins”. In: *Protein Science* 29.1 (2019), 258–267. DOI: 10.1002/pro.3779. URL: <https://doi.org/10.1002/pro.3779>.
- [113] Cramer, J. “The Origins of Logistic Regression”. In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (2002). DOI: 10.2139/ssrn.360300.
- [114] Murtagh, F. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2.5-6 (1991), 183–197. DOI: 10.1016/0925-2312(91)90023-5. URL: [https://doi.org/10.1016/0925-2312\(91\)90023-5](https://doi.org/10.1016/0925-2312(91)90023-5).
- [115] Lecun, Y. et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), 2278–2324. DOI: 10.1109/5.726791.
- [116] Harris, Z. S. “Distributional structure”. In: *Word* 10.2-3 (1954), 146–162.
- [117] Rives, A. et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021).
- [118] Kilinc, M., Jia, K., and Jernigan, R. L. “Improved global protein homolog detection with major gains in function identification”. In: *Proceedings of the National Academy of Sciences* 120.9 (2023). DOI: 10.1073/pnas.2211823120. URL: <https://doi.org/10.1073/pnas.2211823120>.



- 
- [119] Biegert, A. and Söding, J. "Sequence context-specific profiles for homology searching". In: *Proceedings of the National Academy of Sciences* 106.10 (2009), 3770–3775. DOI: 10.1073/pnas.0810767106. URL: <https://doi.org/10.1073/pnas.0810767106>.
  - [120] Finn, R. D., Clements, J., and Eddy, S. R. "HMMER web server: interactive sequence similarity searching". In: *Nucleic Acids Research* 39.suppl (2011), W29–W37. DOI: 10.1093/nar/gkr367. URL: <https://doi.org/10.1093/nar/gkr367>.
  - [121] Camacho, C. et al. "BLAST+: architecture and applications". In: *BMC Bioinformatics* 10.1 (2009).
  - [122] Pearson, W. R. "[5] Rapid and sensitive sequence comparison with FASTP and FASTA". In: *Methods in Enzymology*. Elsevier, 1990, 63–98. DOI: 10.1016/0076-6879(90)83007-v. URL: [https://doi.org/10.1016/0076-6879\(90\)83007-v](https://doi.org/10.1016/0076-6879(90)83007-v).
  - [123] Saripella, G. V., Sonnhammer, E. L. L., and Forslund, K. "Benchmarking the next generation of homology inference tools". In: *Bioinformatics* 32.17 (2016), 2636–2641. DOI: 10.1093/bioinformatics/btw305. URL: <https://doi.org/10.1093/bioinformatics/btw305>.
  - [124] Suzek, B. E. et al. "UniRef: comprehensive and non-redundant UniProt reference clusters". In: *Bioinformatics* 23.10 (2007), 1282–1288.
  - [125] Yeung, W. et al. "Alignment-free estimation of sequence conservation for identifying functional sites using protein sequence embeddings". In: *Briefings in Bioinformatics* 24.1 (2023). DOI: 10.1093/bib/bbac599. URL: <https://doi.org/10.1093/bib/bbac599>.
  - [126] Mistry, J. et al. "Pfam: The protein families database in 2021". In: *Nucleic Acids Research* 49.D1 (2020), D412–D419. DOI: 10.1093/nar/gkaa913. URL: <http://doi.org/10.1093/nar/gkaa913>.
  - [127] Capra, J. A. and Singh, M. "Predicting functionally important residues from sequence conservation". In: *Bioinformatics* 23.15 (2007), 1875–1882. DOI: 10.1093/bioinformatics/btm270. URL: <https://doi.org/10.1093/bioinformatics/btm270>.
  - [128] Almagro Armenteros, J. J. et al. "DeepLoc: prediction of protein subcellular localization using deep learning". In: *Bioinformatics* 33.21 (2017), 3387–3395.
  - [129] Thumulari, V. et al. "DeepLoc 2.0: multi-label subcellular localization prediction using protein language models". In: *Nucleic Acids Research* (2022).
  - [130] Horton, P. et al. "WoLF PSORT: protein localization predictor". In: *Nucleic Acids Research* 35.suppl\_2 (2007), W585–W587.
  - [131] Krogh, A. et al. "Predicting transmembrane protein topology with a hidden markov model: application to complete genomes<sup>11</sup>Edited by F. Cohen". In: *Journal of Molecular Biology* 305.3 (2001), 567–580. DOI: <https://doi.org/10.1006/jmbi.2000.4315>. URL: <https://www.sciencedirect.com/science/article/pii/S0022283600943158>.
  - [132] Small, I. et al. "Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences". In: *PROTEOMICS* 4.6 (2004), 1581–1590.

- [133] Almagro Armenteros, J. J. et al. “Detecting sequence signals in targeting peptides using deep learning”. In: *Life Science Alliance* 2.5 (2019). DOI: 10.26508/lsa.201900429.
- [134] Käll, L., Krogh, A., and Sonnhammer, E. L. “A Combined Transmembrane Topology and Signal Peptide Prediction Method”. In: *Journal of Molecular Biology* 338.5 (2004), 1027–1036.





---

# Computational approaches for peroxisomal protein localization

2

This chapter is based on:

Marco Anteghini & Vitor A.P. Martins dos Santos. Computational Approaches for Peroxisomal Protein Localization.

*Published in: Schrader, M. (eds) Peroxisomes. Methods in Molecular Biology, 2643 (2023). Humana, New York.*

DOI: 10.1007/978-1-0716-3048-8\_29

## Abstract

Computational approaches are practical when investigating putative peroxisomal proteins and for sub-peroxisomal protein localisation in unknown protein sequences. Nowadays, advancements in computational methods and Machine Learning (ML) can be used to hasten the discovery of novel peroxisomal proteins and can be combined with more established computational methodologies. In this chapter, we explain and list some of the most used tools and methodologies for novel peroxisomal protein detection and localisation.

## 2.1 Introduction

Advancements in organelle-specific research are possible also thanks to use case-specific tools such as for sub-peroxisomal and sub-mitochondrial protein localisation [1–5]. These tools, nowadays easily accessible and user-friendly, allow researchers to perform fast and accurate screening while looking for new peroxisomal and mitochondrial proteins [1–5]. Alternatively, general methods for protein sequence localisation can be handy if re-adapted for specific use cases [6–11].

For example, a general peroxisomal protein search from a given set of FASTA sequences can start by detecting the predicted subcellular localisation using DeepLoc-2.0 [6]. After filtering for predicted peroxisomal protein, a researcher can either look for known Peroxisomal Targeting Signals (PTSs) to further filter the dataset [3], or retrieve a list of candidates for future analysis or experimental validations [12].

PTSs are consensus motifs found in many peroxisomal proteins. Specific receptors recognise a PTS and bind to a region of the peroxisomal protein [13]. The known PTSs are: 1) PTS1. The PTS1 receptor is encoded by the PEX5 gene [14] is defined as the final dodecamer with a focus on the terminal tripeptide [15]; 2) PTS2. It is an N-terminal targeting signal and its receptor is encoded by the PEX7 gene (a co-receptor is also involved in the peroxisomal import) [16]; 3) mPTS. It is a cis-acting targeting signal specific for peroxisomal membrane proteins. Its mechanism is still poorly understood [17]. The algorithms defined in Schülter et al. (2009) [3] can detect these different PTSs, and the PTS1 can now be easily and accurately detected also on [https://organelx.hpc.rug.nl/fasta/compute\\_in\\_pts](https://organelx.hpc.rug.nl/fasta/compute_in_pts), as described in recent works [1, 5].

In this chapter, we list a number of practical tools accompanied by specific use cases and a workflow on how to perform a complete peroxisomal protein localisation search. The workflow presented here is supported by a service bundle and a practical study example [18].

## 2.2 Materials

### 2.2.1 Use case-specific tools

- PeroxisomeDB. The PEROXISOME DATABASE (PeroxisomeDB) organise and integrates curated information about peroxisomes. That includes genes, proteins, molecular functions, metabolic pathways and their related disorders [3].

Related prediction tools are also available at

<http://www.peroxisomedb.org/>. In the scope of this chapter, we report three main tools for different PTSs detection: 1) PTS1 binding sites; 2) PTS2 binding sites; 3) Pex19BS binding sites. All the three programs rely on multiple sequence alignments where the input sequence or the input BLOCK is aligned with a predefined BLOCK that contains a specific category of proteins (e.g. proteins containing PTS1).

- **In-Pero.** A computational pipeline that discriminates between matrix and membrane proteins [1]. In-Pero relies on a Support Vector Machine classifier trained on the statistical representation of protein sequences obtained by combining two deep-learning embeddings (UniRep + SecVec) [19, 20]. In-Pero can be executed locally following the instruction available at <https://github.com/MarcoAnteghini/In-Pero> or on the dedicated web server available at [https://organelx.hpc.rug.nl/fasta/compute\\_in\\_pero](https://organelx.hpc.rug.nl/fasta/compute_in_pero).
- **In-Mito.** A computational pipeline that allows classifying the four sub - mitochondrial compartments: matrix, internal-membrane, inter-membrane space and external membrane [1]. In-Mito relies on a Support Vector Machine classifier trained on the statistical representation of protein sequences obtained by combining two deep-learning embeddings (UniRep + SecVec) [19, 20]. In-Mito can run locally following the instruction available at <https://github.com/MarcoAnteghini/In-Mito> or on the dedicated web server available at [https://organelx.hpc.rug.nl/fasta/compute\\_in\\_mito](https://organelx.hpc.rug.nl/fasta/compute_in_mito).
- **DeepMito.** A computational method for predicting sub-mitochondrial localisation based on a convolutional neural network architecture [2]. Given an input protein, DeepMito can discriminate the four sub-mitochondrial compartments: matrix, internal-membrane, inter-membrane space and external membrane. DeepMito is available at <http://busca.biocomp.unibo.it/deepmito/>.

## 2.2.2 General tools for subcellular localisation and transmembrane detection

- **TMHMM2.0 and DeepTMHMM.** TMHMM2.0 is a membrane protein topology prediction method based on a hidden Markov model (HMM) [8]. It predicts transmembrane helices and discriminates between soluble and membrane proteins. The tool is available at <https://services.healthtech.dtu.dk/service.php?TMHMM-2.0>. DeepTMHMM is a novel version of the TMHMM predictor. It is the most complete and currently the best-performing method for the membrane protein topology prediction [21]. The model encodes the primary amino acid sequence by a pre-trained language model and decodes the topology by a state-space model to produce topology and type predictions. DeepTMHMM is available at <https://dtu.biolib.com/DeepTMHMM>.
- **Phobius.** Combined transmembrane topology and signal peptide predictor [11]. The predictor relies on a HMM that models the different sequence re-

gions of a signal peptide and the different regions of a transmembrane protein in a series of interconnected states. Phobius is available at <https://phobius.sbc.su.se/>

- DeepLoc-2.0 [6]. Multi-localization prediction tool based on a pre-trained protein language model that uses a three-stage deep learning approach for sequence classification. 1) The feature representation for each amino acid in the sequence is generated. 2) Attention-based pooling stage produces a single representation for the whole sequence. 3) the prediction stage uses a classifier to output the subcellular labels. DeepLoc-2.0 is available at <https://services.healthtech.dtu.dk/service.php?DeepLoc-2.0>
- PSORT. A computer program that predic protein localisation sites in cells and its last version is WoLF PSORT[7]. WoLF PSORT converts protein amino acid sequences into numerical localisation features; based on sorting signals, amino acid composition and functional motifs. A k-nearest neighbour classifier is used for the final prediction. The webserver is available at <https://psort.hgc.jp/>
- TargetP-2.0. Deep Learning method to identify N-terminal sorting signals, which direct proteins to the secretory pathway, mitochondria, and chloroplasts or other plastids [10]. The method relies on Bi-directional Recurrent Neural Networks (BiRNN) with Long short-term memory (LSTM) cells and a multi-attention mechanism [22]. TargetP-2.0 is available at <https://services.healthtech.dtu.dk/service.php?TargetP-2.0>.

## 2.3 Methods

### 2.3.1 Peroxisomal protein candidates detection

The workflow for a typical analysis represented as a service bundle is visible in Figure 2.1 and also available (with functional links for each tool) at <https://tess.elixir-europe.org/workflows/peroxisomal-candidates-detection>. For an accurate analysis, it is recommended to first look for known PTSs when available and then proceed with further filtering steps. After the PTS detection, we can investigate the presence of transmembrane regions in the aminoacid sequence and filter the results according to the detected PTS. In particular, we can exclude membrane proteins while checking for PTS1 or PTS2. Afterwards, if stringent filtering is required, it is recommended to analyse the candidates with other subcellular localisation tools (see the ‘General tools for subcellular localisation and transmembrane detection’ section) and remove the proteins with unexpected predicted localisation.

Alternatively, we can start our analysis directly from the subcellular localisation prediction and then run the predicted peroxisomal proteins with a sub-peroxisomal classification tool that does not consider PTS motifs [1]. As reported in Figure 2.1, after the subcellular localisation prediction, if we obtain mitochondrial proteins, it is possible to either run DeepMito or In-Mito, while we can execute In-Pero for the peroxisomal proteins [1, 2]. These tools discriminate the sub-organelle compartments,



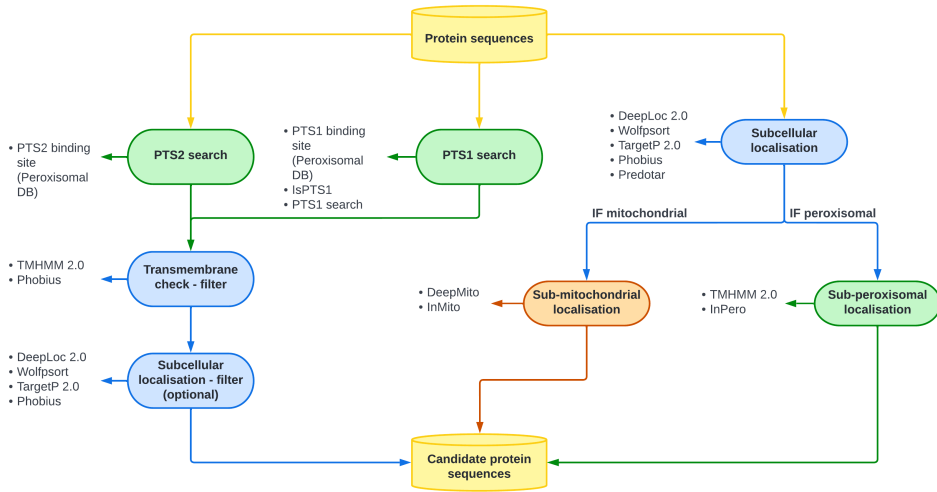


Figure 2.1: Workflow and Service Bundle of a standard peroxisomal and mitochondrial protein localisation analysis. The workflow starts with the initial dataset containing protein sequences in FASTA format. The starting point is the root of the graph ‘Protein sequences’. Each node represents a step of the analysis. Its associated tools are visible on the left of the node. The workflow converges in the final node ‘Candidate protein sequences’, where candidate protein sequences are selected for future analyses or experimental validation.

which are 4 in case of mitochondria (matrix, internal-membrane, inter-membrane space, external membrane) and 2 in case of peroxisomes (matrix, membrane) [1, 2].

As a final step for further validation, selected sequences can be screened for conservation of the potential PTS1/PTS2 using BLAST (the last version while writing this chapter is BLAST+ 2.13.0) [23, 24].

An example of a complete pipeline was performed in the recent work of Kamoshita et al. (2022) [18]. For simplicity we report here a summarised computational pipeline: 1) The Danio rerio proteome was downloaded from UniProt (<https://www.uniprot.org/>) [25] and screened for proteins carrying a PTS1 at the very C-terminus matching the consensus motif [ASGNPHTG]-[RKHQNSL]-[LMIVF]; 2) Among 46,848 proteins, 2,638 proteins matching the pattern were identified and filtered for non membrane proteins with TMHMM Server v. 2.02 [8] (1966 protein left); 3) The 1,966 protein sequences were further analysed with WoLF PSORT (Package Command Line Version 0.2) [7] and entries with Endoplasmic Reticulum as possible subcellular localisation were removed (1,171 sequences left); 4) The identified proteins were further analysed by PTS1 predictor algorithms [3] and sequences which produced no hit with the “metazoa” or “general” modus of the software were removed (371 proteins left); 5) Finally, the obtained entries were manually curated, integrating information from Zebrafish specific datasets and considered for experimental validation.

## Notes:

1. Most of the tools presented in this chapter are designed for Eukaryotes. Some of them can be used for prokaryotic organisms as well (e.g. DeepTMHMM [21]). Note that peroxisomes are only present in Eukaryotes. We advise the user to check the specifications of each tool in the original web server or paper.
2. In this chapter, we list some of the available tools for mitochondrial protein detection. Important components of the organelle division machinery present a dual localisation (peroxisomal and mitochondrial). Moreover, both organelles have proven to be in continuous interplay [26]. For an accurate peroxisomal protein localisation search, it is advised to look into mitochondrial localisation too.

## Acknowledgements

This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. "In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins". In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
- [2] Savojardo, C. et al. "DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks". In: *Bioinformatics* 36.1 (2019), 56–64.
- [3] Schlüter, A. et al. "PeroxisomeDB 2.0: an integrative view of the global peroxisomal metabolome". In: *Nucleic Acids Research* 38.suppl\_1 (2009), D800–D805.
- [4] Claros, M. G. and Vincens, P. "Computational Method to Predict Mitochondrially Imported Proteins and their Targeting Sequences". In: *European Journal of Biochemistry* 241.3 (1996), 779–786.
- [5] Anteghini, M. et al. "OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection". In: *Computational and Structural Biotechnology Journal* 21 (2022), 128–133. DOI: 10.1016/j.csbj.2022.11.058. URL: <https://doi.org/10.1016/j.csbj.2022.11.058>.
- [6] Thumulari, V. et al. "DeepLoc 2.0: multi-label subcellular localization prediction using protein language models". In: *Nucleic Acids Research* (2022).
- [7] Horton, P. et al. "WoLF PSORT: protein localization predictor". In: *Nucleic Acids Research* 35.suppl\_2 (2007), W585–W587.
- [8] Krogh, A. et al. "Predicting transmembrane protein topology with a hidden markov model: application to complete genomes<sup>1</sup>Edited by F. Cohen". In: *Journal of Molecular Biology* 305.3 (2001), 567–580. DOI: <https://doi.org/10.1006/jmbi.2000.4315>. URL: <https://www.sciencedirect.com/science/article/pii/S0022283600943158>.
- [9] Small, I. et al. "Predotar: A tool for rapidly screening proteomes for N-terminal targeting sequences". In: *PROTEOMICS* 4.6 (2004), 1581–1590.
- [10] Almagro Armenteros, J. J. et al. "Detecting sequence signals in targeting peptides using deep learning". In: *Life Science Alliance* 2.5 (2019). DOI: 10.26508/lsa.201900429.
- [11] Käll, L., Krogh, A., and Sonnhammer, E. L. "A Combined Transmembrane Topology and Signal Peptide Prediction Method". In: *Journal of Molecular Biology* 338.5 (2004), 1027–1036.
- [12] Schrader, T. A., Islinger, M., and Schrader, M. "Detection and Immunolabeling of Peroxisomal Proteins". In: *Methods in Molecular Biology*. Springer New York, 2017, 113–130.
- [13] Gould, S. G., Keller, G. A., and Subramani, S. "Identification of a peroxisomal targeting signal at the carboxy terminus of firefly luciferase." In: *Journal of Cell Biology* 105.6 (1987), 2923–2931.

- [14] Kiel, J. A. et al. "Ubiquitination of the Peroxisomal Targeting Signal Type 1 Receptor, Pex5p, Suggests the Presence of a Quality Control Mechanism during Peroxisomal Matrix Protein Import". In: *Journal of Biological Chemistry* 280.3 (2005), 1921–1930.
- [15] Brocard, C. and Hartig, A. "Peroxisome targeting signal 1: Is it really a simple tripeptide?" In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006), 1565–1573.
- [16] Kunze, M. "The type-2 peroxisomal targeting signal". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1867.2 (2020), 118609.
- [17] Van Ael, E. and Fransen, M. "Targeting signals in peroxisomal membrane proteins". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006). Peroxisomes: Morphology, Function, Biogenesis and Disorders, 1629–1638. DOI: <https://doi.org/10.1016/j.bbamcr.2006.08.020>. URL: <https://www.sciencedirect.com/science/article/pii/S0167488906002308>.
- [18] Kamoshita, M. et al. "Insights Into the Peroxisomal Protein Inventory of Zebrafish". In: *Frontiers in Physiology* 13 (2022).
- [19] Alley, E. et al. "Unified rational protein engineering with sequence-based deep representation learning". In: *Nature Methods* 16 (2019). DOI: 10.1038/s41592-019-0598-1.
- [20] Heinzinger, M. et al. "Modeling aspects of the language of life through transfer-learning protein sequences". In: *BMC Bioinformatics* 20 (2019).
- [21] Hallgren, J. et al. "DeepTMHMM predicts alpha and beta transmembrane proteins using deep neural networks". In: *bioRxiv* (2022). DOI: 10.1101/2022.04.08.487609.
- [22] Lin, Z. et al. "A structured self-attentive sentence embedding". In: *arXiv preprint arXiv:1703.03130* (2017).
- [23] Altschul, S. F. et al. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (1990), 403–410. DOI: 10.1016/S0022-2836(05)80360-2. URL: [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2).
- [24] Camacho, C. et al. "BLAST+: architecture and applications". In: *BMC Bioinformatics* 10.1 (2009).
- [25] Consortium, T. U. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
- [26] Schrader, M. et al. "Peroxisome-mitochondria interplay and disease". In: *Journal of Inherited Metabolic Disease* 38.4 (2015), 681–702.





---

# In-Pero: exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins

3

This chapter is based on:

Marco Anteghini, Vitor A.P. Martins dos Santos, Edoardo Saccenti. In-Pero: Exploiting Deep Learning Embeddings of Protein Sequences to Predict the Localisation of Peroxisomal Proteins.

*Published in: Int. J. Mol. Sci. 22(12), 6409 (2021)*

DOI: 10.3390/ijms22126409

## Abstract

Peroxisomes are ubiquitous membrane-bound organelles, and aberrant localisation of peroxisomal proteins contributes to the pathogenesis of several disorders. Many computational methods focus on assigning protein sequences to subcellular compartments, but there are no specific tools tailored for the sub-localisation (matrix vs. membrane) of peroxisome proteins. We present here In-Pero, a new method for predicting protein sub-peroxisomal cellular localisation. In-Pero combines standard machine learning approaches with recently proposed multi-dimensional deep-learning representations of the protein amino-acid sequence. It showed a classification accuracy above 0.9 in predicting peroxisomal matrix and membrane proteins. The method is trained and tested using a double cross-validation approach on a curated data set comprising 160 peroxisomal proteins with experimental evidence for sub-peroxisomal localisation. We further show that the proposed approach can be easily adapted (In-Mito) to the prediction of mitochondrial protein localisation obtaining performances for certain classes of proteins (matrix and inner-membrane) superior to existing tools.

## 3.1 Introduction

In eukaryotes, there are ten main subcellular localisations which can be further subdivided into intra-organellar compartments (see Figure 3.1A). These organelles perform one or more, and often complementary, specific tasks in the cellular machinery. Examples of organelles are the nucleus, for the storage of genetic (DNA) material, mitochondria for the production of energy and the peroxisome.

The organelles provide suitable biological conditions for proteins and the correct transport of a protein to its final destination is crucial to its function. Failure in protein transport systems has been associated with several disorders including Alzheimer's and cancers [1–3].

It has been observed that proteins from different organelles show signatures, in their amino acid composition, that associate with their subcellular localisation [4]. This has led to the hypothesis that each protein has evolved to function optimally in a given subcellular compartment, and to the idea that the information encoded in the sequence can be used to predict the subcellular localisation.

Since the pioneering work of Nakashima and Nishikawa, who used the amino acid composition to discriminate between intra- and extra-cellular proteins [5], several studies have been proposed to predict protein localisation (see [6] for comprehensive reviews). A list of the most common tools for subcellular localisation includes BaCello [7] a predictor based on different Support Vector Machines (SVM) organised in a decision tree; Phobius [8], a combined transmembrane topology and signal peptide predictor; WoLF PSORT [9] a  $k$ -nearest neighbors based classifier; TPpred3 [10], an SVM predictor exploiting N-terminal targeting peptides.

Nowadays, many bioinformatics methods for subcellular and sub-organelle localisation are easily findable and accessible [7, 9–11]. Moreover, the recent applications of machine learning (ML) and deep-learning (DL) approaches to encode protein sequences, has shown promising results in several tasks, including subcellular



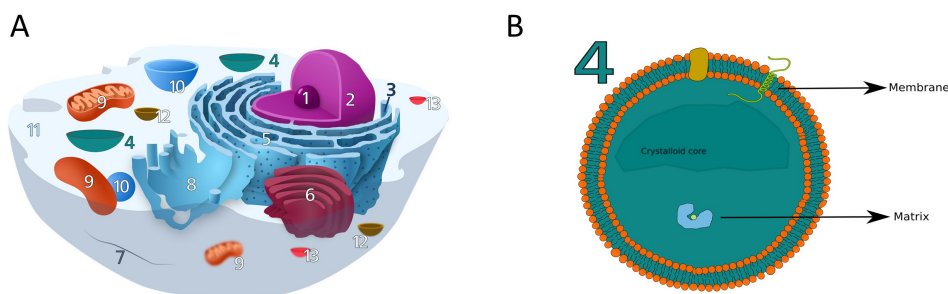


Figure 3.1: (A) The eukaryotic cell and its organelles and compartments: (1) Nucleolus, (2) Nucleus, (3) Ribosome, (4) Peroxisome, (5) Rough Endoplasmic Reticulum, (6) Golgi apparatus, (7) Cytoskeleton, (8) Smooth endoplasmic reticulum, (9) Mitochondrion, (10) Vacuole, (11) Cytoplasm, (12) Lysosome, (13) Vesicles. (B) The peroxisome and its structure, showing the lipidic bilayer membrane, the inner matrix and crystalloid core (not always present). Peroxisomal proteins can be divided into two groups, matrix and membrane proteins, depending on the localisation. Membrane proteins are found attached on the inner and outer surface or can span through the layer (trans-membrane proteins). Panel A is partially adapted from [en.wikipedia.org/wiki/Ribosome#/media/File:Animal\\_Cell.svg](https://en.wikipedia.org/wiki/Ribosome#/media/File:Animal_Cell.svg) accessed on 02-2021.

classification [12–18].

In these approaches, protein sequences are commonly transformed to numerical representations that can be mathematically manipulated. Classically, these representations are referred to as ‘encodings’ and can be broadly subdivided in four categories (i) binary encoding, (ii) encoding based on physical-chemical properties, (iii) evolution-based encoding and, (iv) structural encoding [19]. Examples are the one-hot encoding (1HOT) [19], the residue physical-chemical properties encoding (PROP) [20], the Position-Specific Scoring Matrix (PSSM) [21, 22].

Recently, deep learning methods have also been proposed and applied to extracting fundamental features of a protein and to embed them into a statistical representation that is semantically rich and structurally, evolutionary, and bio-physically grounded [12]. These statistical representations are known as deep-learning embeddings (DL-embeddings) and are a multidimensional transformation of the protein sequence obtained using DL to extract and learn the information from the huge amount of protein sequences available in biological databases.

We can take advantage of these embeddings for several tasks, especially subcellular localisation [14]. Two of the most promising DL-embeddings are the Unified Representation (UniRep) [12] and the Sequence-to-Vector (SeqVec) [13] embeddings. UniRep [12] provides amino-acid embedding containing meaningful physicochemically and phylogenetic clusters and proved to be efficient for distinguishing proteins from various SCOP (structural classifications of proteins) classes. SeqVec showed similar results and optimal performance for predicting subcellular localisation, including peroxisomes [13].

Peroxisomes (see Figure 3.1B) are ubiquitous organelles surrounded by a single biomembrane that are relevant to many metabolic pathways like phospholipid biosynthesis, fatty acid beta-oxidation, bile acid synthesis, docosahexaenoic acid

synthesis, fatty acid alpha-oxidation, glyoxylate metabolism, amino acid degradation, and ROS/RNS metabolism [23]. Peroxisomes are also involved in non-metabolic functions, like cellular stress responses, response to pathogens and antiviral defence, and cellular signalling [24]. Because of this they gained the appellation of "protective" organelles [24] and dysfunctions in peroxisomal proteins have been associated with metabolic disorders [23, 24].

However, the full extent of their functions is still largely unknown [24] and the discovery of new peroxisomal proteins can facilitate further knowledge acquisition.

This leads to the problem of determining the localisation of peroxisome proteins. For instance, both membrane contact site (MCS) proteins [25] and peroxisomal transporters (PT) [26] are found on the membrane: that is, distinguishing between proteins located on the peroxisomal membrane or in its granular matrix is thus a fundamental step for the characterization of unknown peroxisomal proteins.

The problem of protein sub-peroxisomal localisation has received limited attention: as for today, the only way to retrieve information about the sub-peroxisomal localisation is to check for short conserved sequence motif known as signal motifs, or protein targeting signals (PTS) as implemented in the PeroxisomeDB server [27] ([www.peroxisomedb.org](http://www.peroxisomedb.org), accession date 06-2020). Through PeroxisomeDB, given a FASTA sequence as input, it is possible to identify PEX19BS, PTS1 and PTS2 targeting signals: more precisely, PTS1 and PTS2, can identify peroxisomal matrix proteins while PEX19BS can identify peroxisomal membrane proteins.

In this study, we address the problem of predicting the sub-localisation of peroxisomal protein using a computational strategy that combines protein-sequence embedding with classical machine learning. We reviewed and compared four different machine learning approaches, namely Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), and Partial Least Square Discriminant Analysis (PLS-DA) in combination with five protein embedding approaches: residue one-hot encoding (1HOT), residue physical-chemical properties (PROP), Position Specific Scoring Matrices (PSSM), Unified Representation, Sequence-to-Vector.

Based on our comparative study, we built a computational pipeline (In-Pero), which is based on Support Vector Machines and the combination of UniRep and SeqVec embedding. We also tested our approach for sub-mitochondrial localisation, obtaining a predictor (In-Mito) that outperformed most of the existing classifiers.

## 3.2 Results

### 3.2.1 Selection of the Best Classifier for Sub-Peroxisomal Prediction

We compared four commonly used machine learning approaches (Logistic Regression, Partial Least Squares Discriminant analysis, Random Forest and Support Vector Machines) in combination with different protein sequence encodings and embeddings to select the best classification strategy to predict the sub-localisation of peroxisomal proteins. Results are summarised, per classification algorithm, in Table 3.1 where different metrics for model quality quantification are given. All results were obtained with repeated double cross-validation to avoid model overfitting and bias.

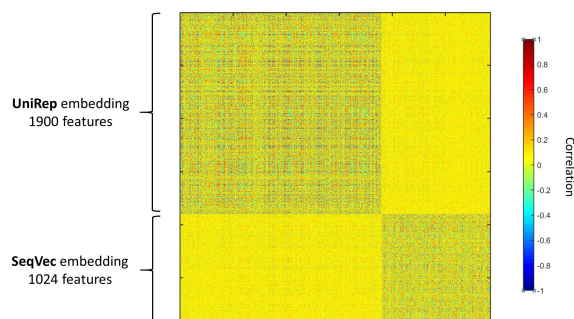


Figure 3.2: Correlation among the UniRep (1900 features) and the SeqVec (1024 features) protein sequence embeddings. Pearson's linear correlation is used and are calculated over 160 protein sequences. The two embeddings are uncorrelated.

3

In general, Logistic regression (Table 3.1a) and Support vector machines (Table 3.1b) showed similar performance, superior to PLS-DA 3.1c) and Random Forest (Table 3.1d). However, the prediction model built using SVM has a smaller standard deviation, indicating higher stability.

We observed that combining two different encodings and/or embeddings gives a better prediction of the peroxisomal sub-localisation. In particular, concatenating UniRep and SeqVec showed a noticeable improvement in the performances. That indicates that the two embeddings carry different and complementary information about the properties of the protein sequence, as given in Figure 3.2, that show how the two embeddings are not correlated.

### 3.2.2 In-PERO a Tool for the Prediction of Peroxisomal Protein Sub-Localisation

Based on the results obtained and discussed in Section 3.2.1, we developed In-Pero, a computational pipeline to predict the Intra-Peroxisomal localisation of a proximal protein, that is, to discriminate between matrix and membrane proteins.

In-Pero is based on a Support Vector Machine classifier trained on the statistical representation of protein sequences obtained by the combination of two deep-learning embeddings (UniRep + SecVec).

In-Pero consists of four main steps (see Figure 3.3A)

1. Input of the protein sequence in FASTA format.
2. Calculation of the statistical representation of the protein sequence using the UniRep ( $1 \times 1900$ ) and the SeqVec ( $1 \times 1024$ ) embeddings.
3. Merging of the two statistical representation to obtain a 2924-dimensional representation of the protein sequence.
4. Prediction of the subcellular localization using the trained SVM prediction model.

Table 3.1: Step Forward Feature Selection for each of the compared methods . (a) Logistic Regression (LR) performances; (b) Support Vector Machines (SVM) performances; (c) Partial Least Square–Discriminant Analysis (PLS–DA) performances; (d) Random Forest (RF) performances. The analysed encodings and embeddings are protein one hot encoding (1HOT), residue physical-chemical properties encoding (PROP), position specific scoring matrix (PSSM), Unified Representation (UniRep) and Sequence to Vector (SeqVec). The results refer to the Double Cross Validation (DCV) procedure performed for each iteration of the forward feature selection (Section 3.4.5). The  $F_1$  (inner) score refers to the inner loop of the DCV while  $F_1$  (outer) refers to the outer loop. The performances are reported in terms of F1 score, BACC, MCC and ACC (see Section 3.4.8 and SM).

(a) LR					
	$F_1$ (inner)	$F_1$ (outer)	BACC	MCC	ACC
1HOT	0.577	0.623 $\pm$ 0.071	0.618 $\pm$ 0.075	0.269 $\pm$ 0.143	0.809 $\pm$ 0.036
PROP	0.607	0.595 $\pm$ 0.109	0.591 $\pm$ 0.093	0.213 $\pm$ 0.222	0.794 $\pm$ 0.054
PSSM	0.615	0.575 $\pm$ 0.067	0.604 $\pm$ 0.089	0.177 $\pm$ 0.144	0.719 $\pm$ 0.040
<b>UniRep</b>	<b>0.765</b>	<b>0.749 <math>\pm</math> 0.068</b>	<b>0.755 <math>\pm</math> 0.077</b>	<b>0.501 <math>\pm</math> 0.137</b>	<b>0.856 <math>\pm</math> 0.032</b>
SeqVec	0.792	0.712 $\pm$ 0.068	0.726 $\pm$ 0.079	0.427 $\pm$ 0.140	0.825 $\pm$ 0.042
UniRep + 1HOT	0.636	0.648 $\pm$ 0.103	0.650 $\pm$ 0.111	0.312 $\pm$ 0.204	0.806 $\pm$ 0.061
UniRep + PROP	0.614	0.595 $\pm$ 0.104	0.589 $\pm$ 0.093	0.234 $\pm$ 0.217	0.812 $\pm$ 0.040
UniRep + PSSM	0.634	0.615 $\pm$ 0.100	0.615 $\pm$ 0.100	0.201 $\pm$ 0.166	0.738 $\pm$ 0.042
<b>UniRep + SeqVec</b>	<b>0.844</b>	<b>0.851 <math>\pm</math> 0.055</b>	<b>0.847 <math>\pm</math> 0.075</b>	<b>0.715 <math>\pm</math> 0.113</b>	<b>0.919 <math>\pm</math> 0.032</b>

(b) SVM					
	$F_1$ (inner)	$F_1$ (outer)	BACC	MCC	ACC
1HOT	0.624	0.693 $\pm$ 0.130	0.713 $\pm$ 0.139	0.396 $\pm$ 0.261	0.819 $\pm$ 0.070
PROP	0.634	0.616 $\pm$ 0.108	0.606 $\pm$ 0.094	0.274 $\pm$ 0.226	0.819 $\pm$ 0.041
PSSM	0.631	0.602 $\pm$ 0.087	0.623 $\pm$ 0.102	0.217 $\pm$ 0.178	0.750 $\pm$ 0.044
UniRep	0.775	0.768 $\pm$ 0.077	0.755 $\pm$ 0.099	0.544 $\pm$ 0.162	0.869 $\pm$ 0.036
<b>SeqVec</b>	<b>0.778</b>	<b>0.777 <math>\pm</math> 0.046</b>	<b>0.813 <math>\pm</math> 0.052</b>	<b>0.567 <math>\pm</math> 0.090</b>	<b>0.856 <math>\pm</math> 0.038</b>
SeqVec + 1HOT	0.68	0.757 $\pm$ 0.079	0.774 $\pm$ 0.103	0.527 $\pm$ 0.166	0.856 $\pm$ 0.042
SeqVec + PROP	0.648	0.597 $\pm$ 0.114	0.589 $\pm$ 0.099	0.218 $\pm$ 0.229	0.812 $\pm$ 0.044
SeqVec + PSSM	0.634	0.614 $\pm$ 0.091	0.639 $\pm$ 0.110	0.244 $\pm$ 0.188	0.756 $\pm$ 0.041
<b>SeqVec + UniRep</b>	<b>0.825</b>	<b>0.859 <math>\pm</math> 0.031</b>	<b>0.863 <math>\pm</math> 0.042</b>	<b>0.721 <math>\pm</math> 0.060</b>	<b>0.919 <math>\pm</math> 0.015</b>

(c) PLS-DA					
	$F_1$ (inner)	$F_1$ (outer)	BACC	MCC	ACC
1HOT	0.452	0.452 $\pm$ 0.005	0.500 $\pm$ 0.001	0.001 $\pm$ 0.001	0.825 $\pm$ 0.015
PROP	0.551	0.582 $\pm$ 0.086	0.575 $\pm$ 0.065	0.249 $\pm$ 0.198	0.831 $\pm$ 0.032
PSSM	0.542	0.592 $\pm$ 0.133	0.582 $\pm$ 0.092	0.277 $\pm$ 0.290	0.844 $\pm$ 0.044
<b>UniRep</b>	<b>0.743</b>	<b>0.782 <math>\pm</math> 0.060</b>	<b>0.782 <math>\pm</math> 0.060</b>	<b>0.568 <math>\pm</math> 0.117</b>	<b>0.875 <math>\pm</math> 0.034</b>
SeqVec	0.759	0.707 $\pm$ 0.081	0.695 $\pm$ 0.080	0.419 $\pm$ 0.160	0.844 $\pm$ 0.044
UniRep + 1HOT	0.478	0.471 $\pm$ 0.051	0.502 $\pm$ 0.034	0.002 $\pm$ 0.112	0.806 $\pm$ 0.023
UniRep + PROP	0.478	0.471 $\pm$ 0.051	0.502 $\pm$ 0.034	0.267 $\pm$ 0.128	0.825 $\pm$ 0.032
UniRep + PSSM	0.564	0.616 $\pm$ 0.110	0.599 $\pm$ 0.075	0.326 $\pm$ 0.233	0.850 $\pm$ 0.041
<b>UniRep + SeqVec</b>	<b>0.806</b>	<b>0.792 <math>\pm</math> 0.078</b>	<b>0.773 <math>\pm</math> 0.074</b>	<b>0.599 <math>\pm</math> 0.166</b>	<b>0.888 <math>\pm</math> 0.042</b>

(d) RF					
	$F_1$ (inner)	$F_1$ (outer)	BACC	MCC	ACC
1HOT	0.569	0.401 $\pm$ 0.077	0.523 $\pm$ 0.050	0.046 $\pm$ 0.089	0.450 $\pm$ 0.124
PROP	0.631	0.572 $\pm$ 0.016	0.564 $\pm$ 0.012	0.203 $\pm$ 0.090	0.812 $\pm$ 0.020
PSSM	0.618	0.585 $\pm$ 0.110	0.567 $\pm$ 0.088	0.261 $\pm$ 0.261	0.819 $\pm$ 0.064
<b>UniRep</b>	<b>0.732</b>	<b>0.741 <math>\pm</math> 0.051</b>	<b>0.779 <math>\pm</math> 0.079</b>	<b>0.503 <math>\pm</math> 0.104</b>	<b>0.838 <math>\pm</math> 0.023</b>
SeqVec	0.695	0.691 $\pm$ 0.035	0.720 $\pm$ 0.053	0.407 $\pm$ 0.790	0.800 $\pm$ 0.042
UniRep + 1HOT	0.728	0.703 $\pm$ 0.063	0.765 $\pm$ 0.089	0.443 $\pm$ 0.139	0.794 $\pm$ 0.032
UniRep + PROP	0.710	0.692 $\pm$ 0.093	0.731 $\pm$ 0.113	0.403 $\pm$ 0.192	0.806 $\pm$ 0.041
UniRep + PSSM	0.699	0.743 $\pm$ 0.100	0.776 $\pm$ 0.128	0.501 $\pm$ 0.209	0.844 $\pm$ 0.052
<b>UniRep + SeqVec</b>	<b>0.778</b>	<b>0.764 <math>\pm</math> 0.135</b>	<b>0.790 <math>\pm</math> 0.141</b>	<b>0.540 <math>\pm</math> 0.267</b>	<b>0.850 <math>\pm</math> 0.087</b>
UniRep + SeqVec + 1HOT	0.774	0.721 $\pm$ 0.108	0.738 $\pm$ 0.121	0.456 $\pm$ 0.214	0.844 $\pm$ 0.044
<b>UniRep + SeqVec + PROP</b>	<b>0.720</b>	<b>0.787 <math>\pm</math> 0.134</b>	<b>0.793 <math>\pm</math> 0.144</b>	<b>0.581 <math>\pm</math> 0.261</b>	<b>0.888 <math>\pm</math> 0.061</b>
UniRep + SeqVec + PSSM	0.741	0.733 $\pm$ 0.123	0.754 $\pm$ 0.136	0.480 $\pm$ 0.242	0.850 $\pm$ 0.054

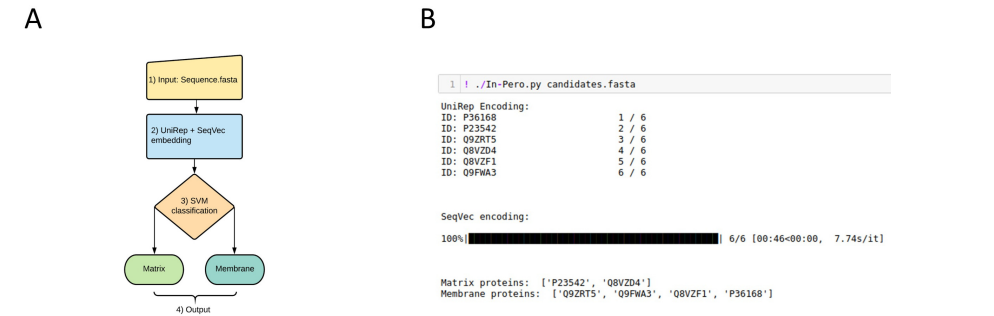


Figure 3.3: **(A)** In-Pero workflow (1) Input as protein fasta sequence. (2) Sequence representation via DL encoding, in particular concatenating UniRep and SeqVec. (3) Support Vector Machines based classification. (4) Output of the sub-peroxisomal location of the queried protein. **(B)** Example of a typical execution, with 6 sequences contained in the candidates.fasta file: the sub-peroxisomal classification of each protein is give.

In-Pero is implemented in Python and work in command line modality. An example of the input command line and output is given in Figure 3.3B.

### 3.2.3 Validation of Sub-Peroxisomal Membrane Protein Prediction

The In-Pero prediction tool was trained and validated using a double cross-validation strategy (see Section 3.4.7) with a stratified 5-fold splitting. The predictive capability of the model was assessed on the data that have not been used for model calibration (i.e., the selection of meta-parameters to obtain the best prediction quality). This approach is a proxy for the use of an external data set for experimental validation, and ensures unbiased model assessment and reduces the risk of over-fitting.

Despite all precautions, we believe it is important to benchmark In-Pero against existing tools. However, at the time of this writing, there are no existing computational specifically designed tools for the sub-localisation of peroxisomal protein. As a work-around, we compared the prediction of In-Pero with those of TMMHMM server (see Section 3.4.9) using a set of 116 peroxisomal protein of unknown sub-peroxisomal localisation (see Section 3.4.2) which have not been used to train the In-Pero classifier.

When In-Pero is run on these 116 proteins, we obtained membrane localisation for 48 and matrix localisation for 68. We tested the 48 protein classified as membrane proteins using TMHMM: 7 were predicted as transmembrane proteins while 13 have characteristics compatible with transmembrane localisation (a value  $\geq 1$  for at least one among the ExpAA, First60 and PredHel scores, see Section 3.4.9). Prediction results are given in Table 3.2.

For two of the proteins predicted as transmembrane proteins (064883 and P20138), there is experimental evidence of them being involved into various cellular membranes [28, 29], while the sub-peroxisomal location is not reported. Among

Table 3.2: Transmembrane and Membrane proteins found with both methods (In-Pero and TMHMM). The seven Transmembrane proteins showing high prediction scores with TMHMM are in bold.

Membrane		Transmembrane
O82399	Q8H191	<b>O64883</b>
P36168	Q8K459	<b>P20138</b>
P90551	Q8VZF1	<b>Q84P23</b>
Q12524	Q9LYT1	<b>Q84P17</b>
Q4WR83	Q9NKW1	<b>Q84P21</b>
Q75LJ4	Q9S9W2	<b>Q9M0X9</b>
Q9SKX5		<b>P08659</b>

the others, four (Q84P23, Q84P17, Q84P21, and Q9M0X9) are present in *Arabidopsis thaliana* while P08659 is present in *Photinus pyralis*. For these five proteins, neither membrane localisation nor the sub-peroxisomal location are given, making them candidates for a more precise annotation.

It should be noted that TMHMM method focuses on predicting transmembrane regions, not predicting subcellular location. However, we can speculate that also peroxisome membrane protein may share some structural and physico-chemical properties similar to cell membrane proteins; thus confronting the results of TMHMM with In-Pero can provide an independent, albeit only partial, validation of the In-Pero classifier.

### 3.2.4 Extending In-Pero to Predict Sub-Mitochondrial Proteins

To explore further the applicability of the combination of machine learning and deep-learning protein sequence embeddings to other problems related to the prediction of protein localisation, we applied In-Pero for sub-mitochondrial classification.

Mitochondrial proteins are physiologically active in different compartments (the matrix, the internal membrane, the inter-membrane space and the external membrane) and their aberrant localisation contributes to the pathogenesis of human mitochondrial pathologies [30]. By adapting In-Pero to a multiclass classification problem, we obtained the In-Mito predictor. We considered both an SVM and a multinomial Logistic Regression as classification algorithms since, in this case, they performed similarly.

There are several tools available for the prediction of sub-mitochondrial localisation. We compared In-Mito against SubMitoPred [31], DeepMito [15], and DeepPred-SubMito [32].

We tested our model with the SM424-18 and SubMitoPred data sets (see Section 3.4.2 for more details). Results are given in Table 3.3.

In-Mito compared favourably to existing predictors, especially in respect to methods designed to classify all four mitochondrial compartments. Moreover, In-Mito shows a well-balanced capability to predict all four different compartments. In particular, In-Mito shows excellent performance in the prediction of matrix proteins and inner membrane proteins, which are the two most abundant subcellular

Table 3.3: Comparison with DeepMito and DeepPred-SubMito (DP-SM) based on the SM424-18 data set (Data A) and the SubMitoPred data set (Data B). The results are reported in terms of Matthews Correlation Coefficient (MCC). The four mitochondrial compartments are outer membrane (O), inner membrane (I), intermembrane space (T) and matrix (M). Given the similar performances of our predictor (In-Mito) implemented with Logistic Regression (LR) and Support Vector Machines (SVM), we report both. The best performances are highlighted in bold font.

Data A	MCC(O)	MCC(I)	MCC(T)	MCC(M)
DeepMito	0.460	0.470	0.530	0.650
DP-SM	<b>0.850</b>	0.490	<b>0.990</b>	0.560
In-Mito (LR)	0.680	<b>0.730</b>	0.690	<b>0.820</b>
In-Mito (SVM)	0.640	0.690	0.620	0.800

Data B	MCC(O)	MCC(I)	MCC(T)	MCC(M)
SubMitoPred	0.420	0.340	0.190	0.510
DeepMito	0.450	0.680	0.540	0.790
DP-SM	<b>0.920</b>	0.690	<b>0.970</b>	0.730
In-Mito (LR)	0.690	0.750	0.620	<b>0.850</b>
In-Mito (SVM)	0.650	<b>0.760</b>	0.540	0.840

compartments (80% of the SubMitoPred data set).

For this multi-class problem, we obtained better prediction performance using either logistic regression (for matrix protein) or SVM (for inter-membrane proteins). This supports the idea of possibly combining different predictors for better classification.

Given the accuracy of the classifications obtained with our approach, we also implemented the tool In-Mito for sub-mitochondrial classification, which works in the same way as In-Pero (Figure 3.3). In particular, the final output here consists of one among the four possible sub-mitochondrial compartments.

### 3.3 Discussion

With this work, we covered a less explored area of bioinformatics analysis of protein sequences, namely the computational prediction of the localisation of peroxisome proteins.

Building on existing approaches, we addressed the problem by combining machine learning algorithms with different combinations of protein encodings and embeddings.

We found that the (combination of) deep learning embeddings Seq-Vec [13] and UniRep [12] outperformed classical encodings when applied to sub-peroxisomal classification. Our newly proposed prediction tool In-Pero obtained a (double cross-validated) classification accuracy of 0.92.

We also adopted the approach deployed in In-Pero to predicting the subcellular localisation of mitochondrial proteins, resulting in the In-Mito classifier. We

found In-Mito to compare favourably with state-of-the-art approaches and for certain classes of proteins (matrix and intermembrane) to outperform existing prediction tools like DeepMito [33] and SubMitoPred [31].

These results suggest that (i) the evolutionary, biochemical and structural information encoded in a protein amino acid sequence cannot be fully captured by one single embedding and that different approaches need to be combined, (ii) deep-learning embeddings are highly versatile and could become a standard for protein sequence representation and analysis and (iii) the possibility of extending In-Pero and In-Mito for the characterisation of other sub-organelles proteins.

Moreover, while in this work we utilised machine learning approaches, we anticipate that our method can be extended to the use of deep-learning methods also for the prediction, such as convolutional neural networks, recurrent neural networks or a combination thereof.

The lack of predictors and tools specifically dedicated to the prediction of sub-localisation of peroxisomal protein makes our work the very first on this subject and presents a complete method and benchmark that can be used as a base for future studies.

## 3.4 Materials and Methods

### 3.4.1 Overview of the Full Comparison Workflow

A complete overview of the comparison strategy for the selection of the best classification strategy to predict the sub-localisation of peroxisomal proteins is given in Figure 3.4. The comparison pipeline consists of three main steps:

1. Data curation: Retrieval of peroxisome protein sequence from UniProt, clustering and filtering.
2. Feature extraction: Transformation of the protein sequences into numerical representations capturing protein characteristics using classical encodings (1HOT, PROP and PSSM) and deep-learning embeddings (UniRep and SeqVec).
3. Full comparison. Double cross-validated assessment of the prediction capability of different combination of machine-learning approaches (Logistic Regression, Support Vector Machines, Partial Least Square Discriminant Analysis and Random Forest) and protein sequence encodings and embeddings using Step Forward Feature Selection.

All methods and approaches used are detailed in the following sections.

### 3.4.2 Data Sets

Amino acid sequences for peroxisomal membrane and matrix proteins were retrieved in December 2019 from the UniprotKB/SwissProt database ([www.uniprot.org](http://www.uniprot.org)) [34].



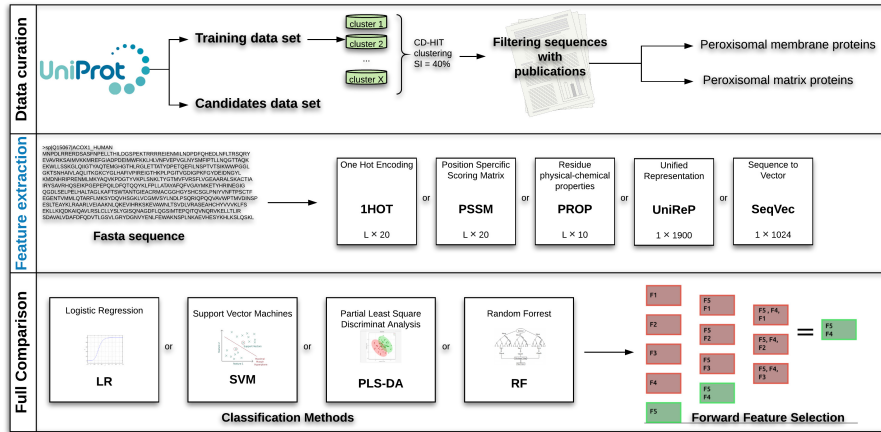


Figure 3.4: Overview of the full analysis for the predictor pipeline development. Data curation: retrieval and selection of peroxisomal protein sequences (see Sections 3.4.2). Feature extraction: conversion of protein sequences to standard encodings, namely: one-hot encoding (1HOT), residue physical-chemical properties encoding (PROP), position specific scoring matrix (PSSM), unified representation (UniRep), sequence-to-vector (SeqVec). Full Comparison: application of classification algorithms (Section 3.4.6) and selection of the best combination(s) of sequence encodings and embeddings using step forward feature selection (see Section 3.4.5)

### Retrieval of Peroxisomal Membrane Proteins

Peroxisomal membrane proteins were retrieved using the query ‘fragment:no locations:(location: “Peroxisome membrane [SL-0203]”) AND reviewed:yes’ with peroxisomal membrane sub-cellular location (SL-0203) to select reviewed, non-fragmented membrane protein sequences.

We obtained 327 non-fragmented protein sequences which were then clustered using Cd-hit [35], with sequence identity of 40%. The representative (i.e., the longest protein sequence in the cluster) of each cluster was chosen resulting in 162 sequences. We used a 40% similarity threshold consistently with DeepMito [15].

We restricted further the selection only to those proteins with at least one associated publication specific for the sub-cellular localization, obtaining 135 highly curated peroxisomal membrane protein sequences. Additionally, three sequences were removed from the data set, since they were not available for the UniRep embedding. The final data set contains 132 membrane proteins.

### Retrieval of Peroxisomal Matrix Proteins

Reviewed, non-fragmented matrix peroxisomal protein sequences were obtained with the query ‘fragment:no locations:(location: “Peroxisome matrix [SL-0202]”) AND reviewed:yes’.

We obtained 60 entries that were reduced to 22 after clustering for similarity and further reduced to 19 after selecting only those proteins with at least one publication specific for the subcellular localisation.

Due to the low number of matrix proteins in comparison to the number of membrane proteins (132), we performed another advanced search in Uniprot with query: ‘fragment:no locations:(location:“Peroxisome [SL-0204]”) NOT locations:(location:“Peroxisome membrane [SL-0203]”) AND reviewed:yes’.

That allowed us to select reviewed, non-fragmented protein sequences, with peroxisomal location (SL-0204), and not peroxisomal membrane location (SL-0203).

We obtained 721 non membrane protein sequences, 202 after clustering, which were reduce to 22 after applying the same filtering procedure. There were 13 common entries between the two subsets; clustering using a 40% sequence similarity threshold gave 28 unique peroxisomal matrix protein sequences.

### Retrieval of Candidate Peroxisomal Proteins

Further peroxisomal protein candidates were retrieved from Uniprot (June 2020). We looked for peroxisomal proteins (SL-0204 and GO:5777) with a non-specific sub-peroxisomal location (SL-0203, SL-0202, GO:5778, GO:5782) and experimental evidence. We then excluded the peroxisomal proteins also found in mitochondria (SL-0173, GO:5739) and in the endoplasmic reticulum (SL-0095, GO:10168), obtaining 116 reviewed entries.

### Data Sets for Sub-Mitochondrial Protein Classification

To assess the applicability of our prediction tool to the prediction of other sub-organelles protein localisation we considered two well-curated data sets containing mitochondrial proteins.

*SM424-18 data set*: this data set was used to build the DeepMito predictor [15] and contains 424 mitochondrial proteins collected using stringent conditions, in particular only non-fragmented proteins with an experimentally determined subcellular localisation in one of the four sub-mitochondrial compartments (outer membrane, inter-membrane space, inner membrane and matrix). Clustering using Cd-hit [35], with a 40% sequence identity threshold was used to select representative sequences. We refer the reader to the original publication for more details [15]. *Sub-MitoPred data set*: this data set was used to build the SubMitoPred predictor [31]. It contains 570 mitochondrial proteins collected using stringent conditions, in particular only non-fragmented proteins with and experimentally determined subcellular localisation in one of the four sub-mitochondrial compartments (outer membrane, inter-membrane space, inner membrane and matrix). Clustering using Cd-hit [35], with a 40% sequence identity threshold was used to select representative sequences. We refer the reader to the original publication for more details [31].

### 3.4.3 Classic Protein Sequence Encoding Methods

We considered three of the most commonly used method for the encoding of the amino acid protein sequences.

- Residue one-hot encoding. The one-hot encoding (1-HOT) [36] is the most used binary encoding method. A residue  $j$  is represented by a  $1 \times 20$  vector containing 0 s except in the  $j$ -th position; for instance alanine (A) is represented as

100,000,000,000,000,000,000. A protein sequence constituted by  $L$  amino acid is thus represented by an  $L \times 20$  matrix.

- Residue physical-chemical properties encoding. Akinori et al. devised a way to represent an amino-acid with ten factors [20] summarising different amino acid physico-chemical properties. This encoding method, often abbreviated as PROP, is the most commonly used physico-chemical encoding [36]. Any given residue  $j$  in the protein sequence is represented by a  $1 \times 10$  vector containing real number. Each number summarise different amino-acid properties and it is an orthogonal property obtained after multivariate statistical analysis applied to a starting set of 188 residue-specific physical properties. A protein sequence constituted by  $L$  amino acid is thus represented by an  $L \times 10$  matrix.
- The Position-specific scoring matrix (PSSM) [21, 22] takes into account the evolutionary information of a protein. This scoring matrix is at the basis of protein BLAST searches (BLAST and PSI-BLAST) [37] where residues are translated into substitution scores. A residue  $j$  in the protein sequence is represented by a  $1 \times 20$  vector containing the 20 specific substitution scores. Amino acid substitution scores are given separately for each position of the protein multiple sequence alignment (MSA) after running PSI-BLAST [37] against the Uniref90 data set (release Oct 2019) for three iterations and e-value threshold set to 0.001. We used a sigmoid function to map the values extracted from the PSI-BLAST checkpoint file in the range [0–1], as in DeepMito [15]. Basically, PSSM captures the conservation pattern in the alignment and summarises evolutionary information of the protein. In PSSM a protein sequence constituted by  $L$  amino acid is thus represented by an  $L \times 20$  matrix.

### 3.4.4 Deep Learning Protein Sequence Embeddings

We considered two recently proposed methods for the embedding of protein sequences based on deep-learning approaches:

1. Unified Representation. The Unified Representation (UniRep) [12] is based on a recurrent neural network architecture (1900-hidden unite) able to capture chemical, biological and evolutionary information encoded in the protein sequence starting from  $\sim 24$  million UniRef50 sequences [38]. Technically, the protein sequence is modelled by using a hidden state vector, which is recursively updated based on the previous hidden state vector. This means that the method learns scanning a sequence of amino acids, predicting the next one based on the sequence it has seen so far. Using UniRep a protein sequence can be represented by an embedding of length 64, 256, or 1900 units depending on the neural network architecture used. In this study, we used the 1900 units long (average final hidden array). For a detailed explanation on how to retrieve the UniRep embedding, we refer the reader to the specific GitHub repository: <https://github.com/churchlab/UniRep> (06-2021).
2. Sequence-to-Vector embedding. The Sequence-to-Vector embedding (SeqVec) [13] embeds biophysical information of a protein sequence taking a natural

language processing approach considering amino acids as words and proteins as sentences. SeqVec is obtained by training ELMo [39], a deep contextualised word representation that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts, which consists of a 2-layer bidirectional LSTM [40] backbone pre-trained on a large text corpus, in this case, UniRef50 [38]. The SeqVec embedding can be obtained by training ELMo at the per-residue (word-level) and per-protein (sentence-level). With the per-residue level it is possible to obtain a protein sequence embedding that can be used to predict the secondary structure or intrinsically disordered region; with the per-protein level embedding it is possible to predict subcellular localisation and to distinguish membrane-bound vs. water-soluble proteins [13]. Here we use the per-protein level representation, where the protein sequence is represented by an embedding of length 1024. For a detailed explanation on how to retrieve the SeqVec embedding, we refer the reader to the specific GitHub repository: <https://github.com/mheininger/SeqVec> (06-2021).

### 3.4.5 Step Forward Feature Selection

Step Forward Feature Selection was used to select the best combination of features (predictors) that is, protein encodings or embeddings to be used as input for classification algorithms [41].

It is a wrapper method that evaluates subsets of variables, in our case, combinations of protein encodings/embeddings. It starts with the evaluation of each individual encoding, and selects that which results in the best performing selected algorithm model. Next, it proceeds by iteratively adding one encoding/embedding to the current best performing features and evaluating the performance of the classification. The procedure is halted when performance worsens and the best combination of embeddings/encodings is retained. A schematic representation of this approach given in Figure 3.4 (Step 3: Full Comparison).

### 3.4.6 Classification Algorithms

The determination of the sub-localisation of peroxisomal (membrane *vs* matrix) protein is easily translated into a two-group classification problem. For this task we considered four widely used machine learning methods (hyperparameters optimisation details in Table 3.4).

- Support Vector Machines (SVM) is an algorithm for two-group classification which aims to find the maximal margin hyperplane separating the points in the feature space [42].
- Random Forest [43, 44] is an ensemble learning method that, in the case of a classification task construct a multitude of decision trees and output the mode of the classes of the individual trees.
- Partial least squares discriminant analysis (PLS-DA) is a partial least squares regression [45, 46] where the response vector  $Y$  contains dummy variables indicating class labels (0-1 in this case). Sample predicted with  $Y \geq 0.5$  are clas-

Table 3.4: Hyperparameters for the grid searches

Hyperparameters	
SVM	<ul style="list-style-type: none"> <li>• C:logspace(− 2,10,13)</li> <li>• gamma:logspace(−9,3,13)</li> <li>• kernel:['linear','poly','rbf','sigmoid']</li> </ul>
RF	<ul style="list-style-type: none"> <li>• n_estimators:[15,25,50,75,100,200,300]</li> <li>• criterion:['gini','entropy']</li> <li>• max_depth:[2,5,10,None]</li> <li>• min_samples_split:[2,4,8,10]</li> <li>• max_features:['sqr','auto','log2']</li> </ul>
PLS-DA	<ul style="list-style-type: none"> <li>• n_components:[2,5,10,15,20,25,30]</li> </ul>
LR	<ul style="list-style-type: none"> <li>• penalty:['l1','l2']</li> <li>• solver:['liblinear','saga']</li> <li>• C:logspace(−3,9,13)</li> </ul>

sified as belonging to class 1 and to class 0 other wise. PLS finds combinations of the original variable maximizing the covariance between the predictor variable and response  $Y$  by projecting the data in a  $k$ -dimensional space with  $k$  possibly much smaller than the original number of variables.

- Logistic Regression (LR). We used a penalised implementation of multivariable logistic regression [47].

### 3.4.7 Model Calibration and Validation

We used double cross validation (DCV) [48, 49] for (i) optimising the hyperparameters of the different classification algorithm used (i.e., for model calibration) and (ii) for an unbiased estimation of prediction errors when the model is applied to new cases (that are within the population of the data used). This strategy is particularly well suited for small data sets.

The DCV strategy consists of two nested cross-validation loops. In the outer loop data is first split in  $k$  folds. One fold is used as Validation set while the remaining  $k - 1$  folds are used as calibration set. The inner loop is applied to the Calibration set which is again split in a test and training set using  $k$ -fold split. In our work we used 5 folds for both inner and outer loop. The inner loop is used to optimise the hyperparameters of the different classification algorithms through a (hyper)grid

search: for each set of hyperparameters, the average classification score is computed across the folds. The hyperparameters corresponding to the best classification score are then used to fit a classification model whose quality is assessed on the Validation set obtaining unbiased model evaluation since the validation data has not been used to train the classification model.

The step forward feature selection procedure described in Section 3.4.5 was included in the calibration loop so that model calibration involved also the selection of the best combination (with respect with model predictive ability) of protein sequence encodings and embeddings.

Given the unbalance of the two class of proteins, different weights were applied to the two class. Class weights were considered to be metaparameters and optimised in the inner calibration loop.

### 3.4.8 Metrics for Model Classification Accuracy

We used several metrics to quantify the quality of the classification models, namely: accuracy (ACC), F1 score [50], balanced accuracy (BACC) [51], Matthews correlation coefficient (MCC) [52]. Formulas are defined as follows:

Accuracy (ACC) that is, the classification error, defined as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

where:  $TP$  is the number of true positives,  $FP$  is the number of false positives;  $TN$  and  $FN$  are the number of true and false negatives, respectively.

The  $F_1$  score [50]:

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}, \quad (3.2)$$

where  $PPV$  is the as positive predicted value (or precision)

$$PPV = \frac{TP}{TP + FP}, \quad (3.3)$$

and  $TPR$  is the true positive rate (recall or sensitivity):

$$TPR = \frac{TP}{TP + FN}. \quad (3.4)$$

The  $F_1$  score is the harmonic mean of recall and precision and varies between 0, if the precision or the recall is 0, and 1 indicating perfect precision and recall.

The balanced accuracy  $BACC$  [51]:

$$BACC = \frac{TPR + TNR}{2}, \quad (3.5)$$

$$TNR = \frac{TN}{TN + FP} \quad (3.6)$$

is the true negative rate or specificity. The  $BACC$  is an appropriate measure when data is unbalanced and there is no preference for the accurate prediction of one of the two classes.

---

The Matthews correlation coefficient (MCC) [52]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (3.7)$$

MCC is the correlation coefficient between the true and predicted class: it is bound between  $-1$  (total disagreement between prediction and observation) and  $+1$  (perfect prediction);  $0$  indicates no better than random prediction. The MCC is appropriate also in presence of class unbalance [53].

### 3.4.9 Prediction of Trans-Membrane Proteins

We used TMHMM (Trans-Membrane Hidden Markov Model) for the prediction of trans-membrane proteins [54, 55] available at: <http://www.cbs.dtu.dk/services/TMHMM/>

TMHMM returns the most probable location and orientation of trans-membrane helices in the protein sequence, summarised in several output parameters: ExpAA, the expected number of amino acids in transmembrane helices. If this number is larger than 18 it is very likely to be a transmembrane protein; First60, the expected number of amino acids in transmembrane helices in the first 60 amino acids of the protein; PredHel, the number of predicted transmembrane helices. We refer the reader to the original publications for more details.

We used TMHMM to predict the localisation of peroxisomal protein with unknown sub-localisation (see data sets description in Sections 3.4.2)

### 3.4.10 Data Availability Statement

All presented tools and data are free to use and available online. The data sets used in this study are available at <https://github.com/MarcoAnteghini/In-Pero/tree/master/Dataset>. Standalone versions of In-Pero and In-Mito are available at <https://github.com/MarcoAnteghini/In-Pero> and <https://github.com/MarcoAnteghini/In-Mito> (accessed on 6 June 2021). The data set containing the 116 investigated proteins as candidates and the related prediction is available for experimental classification and proper UniProt annotations at <https://github.com/MarcoAnteghini/In-Pero/tree/master/Candidates>.

### 3.4.11 Software

For all classification algorithms we used the implementation available in the scikit-learn python library (version 0.22.1) [56]. We obtained the PLS-DA algorithm by adapting the PLS regression algorithm to perform a regression with a dummy variable. All data sets and codes are available at <https://github.com/MarcoAnteghini> and at [www.systemsbiology.nl](http://www.systemsbiology.nl).

## Acknowledgment

We thank Katarina Elez (FU Berlin) for the in-depth discussion about deep-learning encoding procedures.

### **Funding**

This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.



## References

- [1] Hartmann, T. et al. "Alzheimer's disease  $\beta$ A4 protein release and amyloid precursor protein sorting are regulated by alternative splicing". In: *Journal of Biological Chemistry* 271.22 (1996), 13208–13214.
- [2] Shurety, W. et al. "Localization and post-Golgi trafficking of tumor necrosis factor-alpha in macrophages". In: *Journal of interferon & cytokine research* 20.4 (2000), 427–438.
- [3] Bryant, D. M. and Stow, J. L. "The ins and outs of E-cadherin trafficking". In: *Trends in cell biology* 14.8 (2004), 427–434.
- [4] Andrade, M. A., O'Donoghue, S. I., and Rost, B. "Adaptation of protein surfaces to subcellular location 1 Edited by F. E. Cohen". In: *Journal of Molecular Biology* 276.2 (1998), 517–525. DOI: 10.1006/jmbi.1997.1498. URL: <https://doi.org/10.1006/jmbi.1997.1498>.
- [5] Nakashima, H. and Nishikawa, K. "Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies". In: *Journal of molecular biology* 238.1 (1994), 54–61.
- [6] Dönnes, P. and Höglund, A. "Predicting protein subcellular localization: past, present, and future". In: *Genomics, proteomics & bioinformatics* 2.4 (2004), 209–215.
- [7] Pierleoni, A., Martelli, P. L., and Fariselli, P. "BaCelLo: a Balanced subCellular Localization predictor." In: *Bioinformatics (Oxford, England)* 22 (2006), e408–16. DOI: 10.1093/bioinformatics/btl122.
- [8] Käll, L., Krogh, A., and Sonnhammer, E. L. "A Combined Transmembrane Topology and Signal Peptide Prediction Method". In: *Journal of Molecular Biology* 338.5 (2004), 1027–1036.
- [9] Horton, P. et al. "WoLF PSORT: protein localization predictor". In: *Nucleic Acids Research* 35.suppl\_2 (2007), W585–W587.
- [10] Savojardo, C. et al. "TPpred3 detects and discriminates mitochondrial and chloroplastic targeting peptides in eukaryotic proteins". In: *Bioinformatics* 31.20 (2015), 3269–3275.
- [11] Jiang, Y. et al. *MULocDeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation*. 2020.
- [12] Alley, E. et al. "Unified rational protein engineering with sequence-based deep representation learning". In: *Nature Methods* 16 (2019). DOI: 10.1038/s41592-019-0598-1.
- [13] Heinzinger, M. et al. "Modeling aspects of the language of life through transfer-learning protein sequences". In: *BMC Bioinformatics* 20 (2019).
- [14] Elnaggar, A. et al. "ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing". In: *bioRxiv* (2020).

- [15] Savojardo, C. et al. “DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks”. In: *Bioinformatics* 36.1 (2019), 56–64.
- [16] Almagro Armenteros, J. J. et al. “DeepLoc: prediction of protein subcellular localization using deep learning”. In: *Bioinformatics* 33.21 (2017), 3387–3395.
- [17] Ho Thanh Lam, L. et al. “Machine Learning Model for Identifying Antioxidant Proteins Using Features Calculated from Primary Sequences”. In: *Biology* 9.10 (2020).
- [18] Le, N. Q. K. and Huynh, T.-T. “Identifying SNAREs by Incorporating Deep Learning Architecture and Amino Acid Embedding Representation”. In: *Frontiers in Physiology* 10 (2019), 1501.
- [19] Jing, X. et al. “Amino Acid Encoding Methods for Protein Sequences: A Comprehensive Review and Assessment”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17.6 (2020), 1918–1931. DOI: 10.1109/TCBB.2019.2911677.
- [20] Kidera, A. et al. “Statistical Analysis of the Physical Properties of the 20 Naturally Occurring Amino Acids”. In: *Journal of Protein Chemistry* 4 (1985), 23–55. DOI: 10.1007/BF01025492.
- [21] Attwood, T. “Profile (Position-Specific Scoring Matrix, Position Weight Matrix, PSSM, Weight Matrix)”. In: *Dictionary of Bioinformatics and Computational Biology*. American Cancer Society, 2004. ISBN: 9780471650126.
- [22] Stormo, G. D. et al. “Use of the ‘Perceptron’ algorithm to distinguish translational initiation sites in *E. coli*”. In: *Nucleic Acids Research* 10.9 (1982), 2997–3011.
- [23] Wanders, R. J. A., Waterham, H. R., and Ferdinandusse, S. “Metabolic Interplay between Peroxisomes and Other Subcellular Organelles Including Mitochondria and the Endoplasmic Reticulum”. In: *Frontiers in Cell and Developmental Biology* 3 (2016), 83.
- [24] Islinger, M. et al. “The peroxisome: an update on mysteries 2.0”. In: *Histochemistry and Cell Biology* 150 (2018), 1–29. DOI: 10.1007/s00418-018-1722-5.
- [25] Farré, J.-C. et al. “Peroxisome biogenesis, membrane contact sites, and quality control”. In: *EMBO reports* 20.1 (2019), e46864.
- [26] Baker, A. et al. “Peroxisomal ABC transporters: functions and mechanism”. In: *Biochemical Society Transactions* 43.5 (2015), 959–965.
- [27] Schlüter, A. et al. “PeroxisomeDB 2.0: an integrative view of the global peroxisomal metabolome”. In: *Nucleic Acids Research* 38.suppl\_1 (2009), D800–D805.
- [28] Lipka, V. et al. “Pre- and Postinvasion Defenses Both Contribute to Nonhost Resistance in *Arabidopsis*”. In: *Science* 310.5751 (2005), 1180–1183.
- [29] Siddiqui, S. S. et al. “The Alzheimer’s disease-protective CD33 splice variant mediates adaptive loss of function via diversion to an intracellular pool”. In: *Journal of Biological Chemistry* 292.37 (2017), 15312–15320.

- 
- [30] Schapira, A. H. "Mitochondrial disease". In: *The Lancet* 368.9529 (2006), 70–82.
  - [31] Kumar, R., Kumari, B., and Kumar, M. "Proteome-wide prediction and annotation of mitochondrial and sub-mitochondrial proteins by incorporating domain information". In: *Mitochondrion* 42 (2018), 11–22.
  - [32] Wang, X., Jin, Y., and Zhang, Q. "DeepPred-SubMito: A Novel Submitochondrial Localization Predictor Based on Multi-Channel Convolutional Neural Network and Dataset Balancing Treatment". In: *International Journal of Molecular Sciences* 21.16 (2020), 5710.
  - [33] Savojardo, C. et al. "Large-scale prediction and analysis of protein sub-mitochondrial localization with DeepMito". In: *BMC Bioinformatics* 21.S8 (2020). DOI: 10.1186/s12859-020-03617-z. URL: <https://doi.org/10.1186/s12859-020-03617-z>.
  - [34] Consortium, T. U. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
  - [35] Li, W. and Godzik, A. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22 (2006), 1658–1659.
  - [36] Jing, X. et al. "Amino Acid Encoding Methods for Protein Sequences: A Comprehensive Review and Assessment". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 17.6 (2020), 1918–1931. DOI: 10.1109/TCBB.2019.2911677.
  - [37] Altschul, S. et al. "Gapped blast and psi-blast: A new generation of protein database search programs". In: *Nucl. Acids. Res.* 25 (1996), 3389–3402.
  - [38] Suzek, B. E. et al. "UniRef: comprehensive and non-redundant UniProt reference clusters". In: *Bioinformatics* 23.10 (2007), 1282–1288.
  - [39] Peters, M. E. et al. *Deep contextualized word representations*. 2018.
  - [40] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
  - [41] Meyer-Baese, A. and Schmid, V. "Chapter 2-feature selection and extraction". In: 2014, 21–69.
  - [42] Boser, B. E., Guyon, I. M., and Vapnik, V. N. "A Training Algorithm for Optimal Margin Classifiers". In: COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401. URL: <https://doi.org/10.1145/130385.130401>.
  - [43] Tin Kam Ho. "The random subspace method for constructing decision forests". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), 832–844. DOI: 10.1109/34.709601.
  - [44] Breiman, L. "Random forests". In: *Machine learning* 45.1 (2001), 5–32.

- [45] Wold, H. "11 - Path Models with Latent Variables: The NIPALS Approach - NIPALS = Nonlinear Iterative Partial Least Squares." In: *Quantitative Sociology*. Ed. by H. Blalock et al. International Perspectives on Mathematical and Statistical Modeling. Academic Press, 1975, 307–357. ISBN: 978-0-12-103950-9. DOI: <https://doi.org/10.1016/B978-0-12-103950-9.50017-4>. URL: <http://www.sciencedirect.com/science/article/pii/B9780121039509500174>.
- [46] Wold, S. et al. "The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses". In: *SIAM Journal on Scientific and Statistical Computing* 5.3 (1984), 735–743.
- [47] Cramer, J. "The Origins of Logistic Regression". In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (2002). DOI: 10.2139/ssrn.360300.
- [48] Cawley, G. C. and Talbot, N. L. C. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation". In: *Journal of Machine Learning Research* 11.70 (2010), 2079–2107. URL: <http://jmlr.org/papers/v11/cawley10a.html>.
- [49] Filzmoser, P., Liebmann, B., and Varmuza, K. "Repeated double cross validation". In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 23.4 (2009), 160–171.
- [50] Rijsbergen, C. J. V. *Information Retrieval*. 2nd. Butterworth-Heinemann, 1979.
- [51] Brodersen, K. H. et al. "The balanced accuracy and its posterior distribution". In: *2010 20th International Conference on Pattern Recognition*. IEEE. 2010, 3121–3124.
- [52] Matthews, B. W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), 442–451.
- [53] Boughorbel, S., Jarray, F., and El-Anbari, M. "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric". In: *PloS one* 12.6 (2017), e0177678.
- [54] Sonnhammer, E. L., Von Heijne, G., Krogh, A., et al. "A hidden Markov model for predicting transmembrane helices in protein sequences." In: *Ismb*. Vol. 6. 1998, 175–182.
- [55] Krogh, A. et al. "Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes". In: *Journal of molecular biology* 305.3 (2001), 567–580.
- [56] Pedregosa, F. et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), 2825–2830.





---

# OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal targeting signal detection

4

This chapter is based on:

Marco Anteghini, Asmaa Haja, Vitor A.P. Martins dos Santos, Lambert Schomaker and Edoardo Saccenti. OrganelX Web Server for Sub-Peroxisomal and Sub-Mitochondrial Protein Localization and Peroxisomal Target Signal detection.

*Published in: Computational and Structural Biotechnology Journal 21, 128-133 (2023)*

DOI: 10.1016/j.csbj.2022.11.058

## Abstract

We present the OrganelX e-Science Web Server that provides a user-friendly implementation of the In-Pero and In-Mito classifiers for sub-peroxisomal and sub-mitochondrial localization of peroxisomal and mitochondrial proteins and the Is-PTS1 algorithm for detecting and validating potential peroxisomal proteins carrying a PTS1 signal sequence. The OrganelX e-Science Web Server is available at <https://organelx.hpc.rug.nl/fasta/>.

## 4.1 Introduction

Signatures in the amino acid sequences of proteins have been associated with domains, family functional sites and their sub-cellular localization [1–4]. These sequences can be used in association with machine learning (ML) approaches to develop prediction tools, that nowadays are easily findable and accessible [5–8].

Deep-learning approaches have been recently used to embed (encode) the protein sequences, which showed promising results for several tasks, including sub-cellular classification [9–15]. The Unified Representation (UniRep) [9] and the Sequence-to-Vector (SeqVec) [10] are two of the most promising and already used protein sequence embeddings. UniRep provides an amino-acid embedding that summarizes physico-chemical properties and phylogenetic clusters and has been shown to be efficient for distinguishing proteins from various structural classifications of protein classes [9]. SeqVec showed optimal performance for predicting sub-cellular localisation [10]. The potential of these embeddings has been recently explored for highly specific tasks, such as sub-organelle localisation: in particular, they have been used for sub-peroxisomal and sub-mitochondrial protein localisation [16].

Peroxisomes and mitochondria are ubiquitous organelles surrounded by a single (peroxisomes) or a double (mitochondria) biomembrane that is relevant to many metabolic and non-metabolic pathways [17, 18]. The full extent of the functions of peroxisomes, mitochondria and of the involved pathways is still largely unknown [18]: in this light, the discovery of new peroxisomal and mitochondrial proteins can facilitate further knowledge acquisition.

Here we present the OrganelX Web Server (available at <https://organelx.hpc.rug.nl/fasta/>) which hosts two existing algorithms designed to predict sub-peroxisomal (In-Pero) and sub-mitochondrial (In-Mito) localization of a (set of) protein(s) starting from the amino acid sequence(s). The In-Pero and In-Mito algorithm have been introduced in [16] and can be used to predict the sub-cellular localization of known or putative peroxisomal and mitochondrial proteins whose localization is unknown. We also introduce a new functionality (the Is-PTS1 algorithm) for the classification of protein sequences as peroxisomal (*i.e.* proteins that can be imported in the peroxisome) or non-peroxisomal starting from the detection of a specific peroxisomal targeting signal (PTS1)[19].

To our knowledge, there are no online resources that allow simple and fast prediction of the sub-peroxisomal and sub-mitochondrial localization or the prediction of peroxisomal proteins through identification of the PST1 signal starting from the amino acid sequence. These tools offered online through the OrganelX server facil-



itate research on peroxisomes and mitochondria by making the prediction of protein sub-cellular localisation easy to perform (only the upload of protein FASTA sequences is needed). OrganelX can be used without the need for programming skills and significantly reduces the number of bioinformatic steps that should have been otherwise performed to extract relevant information from the protein sequences of interest [20].

## 4.2 Materials and Methods

### 4.2.1 In-Pero and In-Mito classifiers

The In-Pero and In-Mito algorithms and the prediction models implemented in the OrganelX Web Server have been introduced and described in Anteghini *et al.* (2021) [16]. We give here a brief account of the most important characteristics. We refer the reader to the original publication for full details on algorithm development, training and validation.

The In-Pero prediction model was originally trained on a curated, non-redundant (40% of sequence identity) data set of 160 peroxisomal proteins [16] with validated sub-cellular localization; the In-Mito model was trained on a curated, non-redundant (40% of sequence identity) data set of 424 mitochondrial proteins [16] also with validated sub-cellular localization.

Both algorithms start by encoding the protein amino acid sequence using the concatenation of two deep learning-based sequence embeddings [16]: UniRep [9] and Seqvec [10, 16].

The classification problems are solved using Support Vector Machines [21]. The In-Pero algorithm predicts whether a proximal protein belongs to the matrix or is a (trans)membrane protein, resulting in a binary classification problem. The In-Mito algorithm predicts the possible localization of mitochondrial proteins: matrix, inner-membrane, inter-membrane and outer-membrane, resulting in a four-class classification problem.

### 4.2.2 The Is-PTS1 classifier

The proteins that are imported into peroxisomes (peroxisomal proteins) are directed to the peroxisome through the PEX5 receptor that recognizes a specific region of the peroxisomal protein called a peroxisomal targeting signal 1 (PTS1) [22]. Operationally, the PTS1 is defined as dodecamer sequences at the C-terminal ends of the protein sequence which accommodate physical contacts with both the surface and the binding cavity of PEX5 and ensure accessibility of the extreme C-terminus [19]. However, the presence of the PTS1 is not a guarantee for the import of proteins across the peroxisomal membrane [19, 23, 24]. The problem is then to predict whether a protein carrying the PTS1 is a peroxisomal protein or not.

The Is-PTS1 algorithm first searches putative PTS1 signals by matching a regular expression [ASCNPHGTGEQ][RKHQNSL][LAMIVF] in the C-terminal part of the sequence (last 3 amino acids) [20, 25].

Due to some limitations in the embedding generation procedure, we recommend the user upload sequences with less than 1200 residues [9, 10]. When dealing with

longer sequences we recommend conserving the C-terminal part and eventually removing the N-terminal part.

If the PTS1 signal is found, the full amino acid sequence is encoded using the concatenation of UniRep [9] and Seqvec [10] protein embeddings [9, 10] as in the case of the In-Pero and In-Mito algorithms [16]. The binary classification of the protein sequence as peroxisomal or not-peroxisomal is carried over using a Support Vector Machine classifier [21] trained on a non-redundant (40% of sequence identity) data set consisting of 72 peroxisomal proteins (positives) and 155 non-peroxisomal proteins (negatives) all carrying a putative PTS1 signal.

An additional data set of 5 different proteomes of five organisms (*Saccharomyces cerevisiae*, *Homo sapiens*, *Danio rerio*, *Mus musculus* and *Bos taurus*) was assembled to assess how many proteins contain a putative PTS1 signal. The protein sequence was downloaded from UniProt [26] (release 04\_2022) and only the reviewed sequences were considered. An overview of the number of proteins containing a PTS1 signal is reported in Table 4.1. Considering the proteins from all the species, 6.4% of the reviewed protein carrying a putative PTS1 signal are also annotated as peroxisomal.

Table 4.1: Summary statistics (per organism) of proteins with the putative PTS1 signals retrieved from UniProt. ‘n. protein’ indicates the total number of proteins retrieved per organism, the peroxisomal proteins are in brackets; ‘n. matches’ the number of proteins containing a putative PTS1 signal in the C-terminal part of the sequence; ‘true matches’ (TM) indicates how many among the ‘n. of matches’ are annotated as peroxisomal.

Organism	n. proteins (pero)	n. matches	true matches	% TM
<i>S. cerevisiae</i>	6050 (85)	74	21	28
<i>Homo sapiens</i>	20360 (143)	1180	59	5
<i>Danio rerio</i>	3216 (22)	158	8	5
<i>Mus musculus</i>	17085 (146)	976	63	6
<i>Bos taurus</i>	6015 (53)	297	22	7

### 4.2.3 Model optimization

The training, hyper-parameters optimization and validation procedures of the In-Pero, In-Mito and Is-PTS prediction models were carried over using a repeated double cross-validation approach [27, 28] as detailed in [16].

### 4.2.4 Prediction results

The results of the prediction (Peroxisomal and Mitochondrial sub-cellular localization, presence of the PTS1/peroxisomal protein) are given with an associated probability. For the binary classifiers (In-Pero and Is-PTS1) the class probability is calibrated using Platt scaling [29] from the logistic regression on the SVM scores, fit by additional cross-validation on the training data. For the multi-class classifier (In-Mito), the class probability was calculated using the improved version of the coupling approach [30, 31].

### 4.2.5 Data sets for extra validation

We assembled two additional data sets for extra validation of the In-Pero and Is-PTS algorithms (Web server implementation). The In-Mito algorithm was already externally validated in the original publication against two existing tools: DeepMito [32] and DeepPred-SubMito [33] (see Table 3 in [16]).

For the validation of In-Pero, we queried UniProt [34] for reviewed proteins with a clear sub-peroxisomal annotation in the membrane ("SL-0203" and "GO:0005778") or matrix ("SL-0202" and "GO:0005782"). The resulting sequences were then clustered for 40% of sequence identity with CD-hit [35]. Sequences overlapping with our original training set were removed, obtaining 85 membrane proteins and 59 matrix proteins.

To validate the Is-PTS1 algorithm we retrieved from UniProt (and processed in a similar way) 15 peroxisomal proteins carrying the PTS1 signal (true positives) and 15 non-peroxisomal proteins carrying the PTS1 signal (true negatives).

### 4.2.6 Software

The OrganelX Web Server was implemented using Django, a high-level Python web framework [36] (<https://www.djangoproject.com/>).

The internal services for running the classification algorithms are located on Peregrine, the high-performance computing cluster at the University of Groningen, the Netherlands. For more info see <https://www.rug.nl/society-business/centre-for-information-technology/research/services/hpc/facilities/peregrine-hpc-cluster?lang=en>.

## 4.3 Results

### 4.3.1 Performance and validation of the prediction algorithms

#### Performance of In-Pero and In-Mito

The performance and benchmarking of the In-Pero and In-Mito algorithm are exhaustively illustrated and discussed in [16]. For convenience we give in Table 4.2 a summary of the validation results from [16].

#### Validation of the Performance of the Is-PTS1 algorithm

The Is-PTS1 predictor is a newly implemented algorithm. Its overall performance was assessed against the data set containing peroxisomal protein carrying a PTS1 (see Table 4.1). The yeast peroxisome is the organelle with the highest protein concentrations which partially explain the high quantity of annotated peroxisomal protein carrying a PTS1 signal found in Uniprot [38]. Also, peroxisomal proteins are often studied on yeast as a model organism [39]. Is-PTS1 performance on the indicated data set is excellent: ACC=  $0.92 \pm 0.01$  (Accuracy),  $F_1 = 0.91 \pm 0.01$  ( $F_1$  score), AUC=  $0.92 \pm 0.02$  (Area Under the Curve) and MCC=  $0.92 \pm 0.01$  (Matthews' Correlation Coefficient). Results are averaged over 5 cross-validation splits.

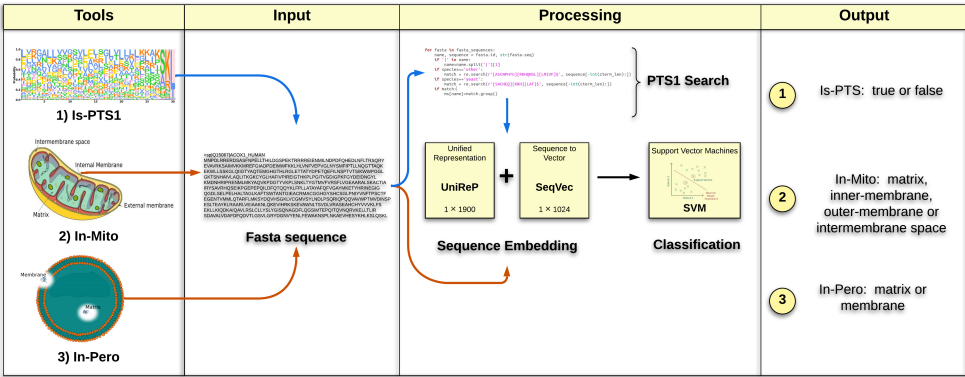


Figure 4.1: Overview of the OrganelX e-Science Web Server. The workflow of prediction of protein sub-cellular localization (In-Pero and In-Mito) and Peroxisomal Target Signal detection (Is-PTS1) is organized in four main steps: (1) Selection of the appropriate prediction algorithm; (2) Input (upload) of FASTA file containing one or more protein sequence; (3) Embedding of the protein sequence(s) and SVM-based classification; (4) Generation and presentation of the result output.

### 4.3.2 Extra validation of the Web server implementation

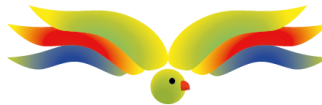
The performance of the In-Pero predictor on the extra validation data set is given in Table 4.3: the quality metrics are in line with what observed in the original publication [16]. The performance of the Is-PTS1 predictor is consistent with the results obtained in the training data set (see Section 3.1.2).

### 4.3.3 Using the OrganelX Web Server

An overview of the functionalities available OrganelX Web Server is shown in Figure 4.1. The different prediction tools (In-Pero, In-Mito and Is-PTS1) are accessible from the homepage as shown in Figure 4.2.

#### OrganelX Web Server: input

The input for the In-Pero (sub-cellular localization of peroxisomal proteins), In-Mito (sub-cellular localization of mitochondrial proteins) and Is-PTS1 (detection of a peroxisomal targeting signal) algorithms available on the OrganelX Web Server is a FASTA text file containing one or more protein sequence. Each sequence begins with a single-line description, followed by amino-acid sequence data. The single-line description consists of **>sp|ID|Desc** symbols, where **>sp** is a fixed prefix, **ID** is the sequence name, and **Desc** is a descriptive text, followed by tokens of the FASTA sequence on the next lines. Alternatively, the single-line description can be **>ID** as a basic FASTA file. The input window of the OrganelX Web Server is shown in Figure 4.3A.



Welcome !

This page is part of the organelx website and present 3 predictive tools

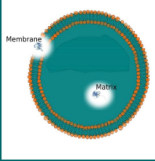
- (1) In-Pero for sub-peroxisomal protein localisation
- (2) In-Mito for sub-mitochondrial protein localisation
- (3) Is-PTS1 for detection and validation of potential protein targeting signal in peroxisomes

Please contact our [Team](#) for any questions, remarks or improvements

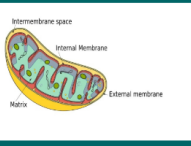
4

**SKL**  
  
r'[EQASCNPHTG][RKHQNSL][LMIVFA]S'

**Is-PTS1**  
Search and validate a PTS1 signal.  
[Go to page](#) [GitHub code](#)



**In-Pero**  
Sub-peroxisomal localisation detection.  
[Go to page](#) [GitHub code](#)



**In-Mito**  
Sub-mitochondrial localisation detection.  
[Go to page](#) [GitHub code](#)

**EXAMPLE**

**Test example**  
Test for Is-PTS1, In-Pito and In-Mito.  
[Go to page](#)

Figure 4.2: Homepage of the OrganelX Web Server (<https://organelx.hpc.rug.nl/fasta/>). From the homepage, the user can access the three prediction tools via the: Is-PTS1 (prediction of peroxisomal proteins based on the presence of the Peroxisome Target Signal), In-Pero (sub-cellular localization of peroxisomal proteins) and In-Mito (sub-cellular localization of peroxisomal proteins). An example is also available. Is-PTS1, In-Pero, In-Mito predictor tools as well as visualize an example. The blue buttons 'Go to page' redirect the user to the specific tool.

Table 4.2: Performance of the In-Pero and In-Mito prediction algorithms from [16]. Results are given as mean  $\pm$  standard deviation over a 5-fold Double Cross Validation. Prediction quality metrics:  $F_1$  score ( $F_1$ , the harmonic mean of precision and recall), Accuracy (ACC), Matthews' Correlation Coefficient (MCC) [37] and the Area Under the Curve (AUC). The performance of the In-Mito classifier are quantified using MCC for each sub-cellular mitochondrial compartment: outer membrane (O), inner membrane (I), inter-membrane space (T) and matrix (M). The In-Mito performances are benchmarked with two other methods namely DeepMito [32] and DeepPred-SubMito (DP-SM) [33].

Method	$F_1$	ACC	MCC	AUC
In-Pero	$0.86 \pm 0.03$	$0.92 \pm 0.01$	$0.72 \pm 0.06$	$0.91 \pm 0.02$
	MCC (O)	MCC (I)	MCC (T)	MCC (M)
DeepMito	0.46	0.47	0.53	0.65
DP-SM	0.85	0.49	0.99	0.56
In-Mito	0.64	0.69	0.62	0.80

Table 4.3: Performances of the In-Pero and Is-PTS1 predictor on two extra validation data sets. Performance quality metrics:  $F_1$  score ( $F_1$ ), Accuracy (ACC), Matthews' Correlation Coefficient (MCC) [37] and Area under the curve (AUC).

Tools	$F_1$	ACC	MCC	AUC
In-Pero	0.83	0.88	0.74	0.86
Is-PTS1	0.84	0.83	0.67	0.83

### OrganelX Web Server: submitting a job

When submitting a job, the user can specify a username, and an email address (optional) and upload a FASTA file. The user can either wait for the results via email or refresh the result web page. The result page is automatically refreshed every 3 minutes. The computation time may change depending on the file size and the traffic on the website. If an email address has been provided, the user will receive a message including the results attached in a .csv file (e.g. Figure 4.4). An example can be accessed at [https://organelx.hpc.rug.nl/fasta/test\\_example](https://organelx.hpc.rug.nl/fasta/test_example), from where an example FASTA file can be downloaded.

### OrganelX Web Server: output

The results are given in a .csv file which allows easy manipulation and re-use for further analysis. The .csv file contains the classification results and the probabilities for each predicted class. The results for each class are reported under its specific column, while each row contains the IDs of the corresponding classified entries. The output window of the OrganelX Web Server is shown in Figure 4.3A.

(a) Input window.

(b) Output window.

Figure 4.3: Input and output windows of the Organelx Web Server. (A) the user can specify an arbitrary username, an email address where to receive the results. The FASTA file is uploaded by clicking on the the 'Choose file' button; (B) the probabilities for each protein in the FASTA file will appear next to a specific class (e.g. 'pred membrane' or 'pred matrix'). In case of errors during the embedding generation, the protein ID will be flagged as 'not encoded'.

ProteinID	Membrane	Matrix
A1L259	0.445	0.554
Q6NV34	0.202	0.797
P41903	0.784	0.215

Figure 4.4: Output file in .csv format obtained from a FASTA containing 3 sequences. The column 'ProteinID' shows the specific UniProt ID for each entry; the columns 'Membrane' and 'Matrix' show the probability associated to the 'Membrane' and 'Matrix' classes.

## 4.4 Conclusions

The In-Pero predictor allows for accurately classifying membrane and matrix proteins inside the peroxisomes. Is-PTS1 predictor detects peroxisomal proteins carrying a PTS1 signal. The In-Mito predictor can be used as a complementary tool when investigating ambiguous or double localization in mitochondrial proteins. These tools proved to be accurate and a valid alternative to the commonly used pipelines, which are less precise, fragmented and time demanding. These three prediction algorithms are now made easily accessible and simple to use through the OrganelX Web server. OrganelX provides a solution to the problem of accurately performing sub-organelle classification and contributes to improving the lack of specific computational methods in peroxisomal research and will facilitate the work of the many groups working on peroxisome and mitochondria research.

## Availability

OrganelX e-Science Web Server can be reached at:

<https://organelx.hpc.rug.nl/fasta/>. The data sets and stand-alone versions of the predictors (Python code) are available at:

<https://github.com/MarcoAnteghini/In-Pero> (In-Pero); <https://github.com/MarcoAnteghini/In-Mito> (In-Mito) and <https://github.com/MarcoAnteghini/Is-PTS1> (Is-PTS1).

## Acknowledgement

We thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster. The e-Science server was realized with the support and nurturing of Ger Strikwerda. This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.



## References

- [1] Andrade, M. A., O'Donoghue, S. I., and Rost, B. "Adaptation of protein surfaces to subcellular location 1 1Edited by F. E. Cohen". In: *Journal of Molecular Biology* 276.2 (1998), 517–525. DOI: 10.1006/jmbi.1997.1498. URL: <https://doi.org/10.1006/jmbi.1997.1498>.
- [2] Hunter, S. et al. "InterPro: the integrative protein signature database". In: *Nucleic Acids Research* 37.Database (2009), D211–D215. DOI: 10.1093/nar/gkn785. URL: <https://doi.org/10.1093/nar/gkn785>.
- [3] Scott, M. S., Thomas, D. Y., and Hallett, M. T. "Predicting Subcellular Localization via Protein Motif Co-Occurrence". In: *Genome Research* 14.10a (2004), 1957–1966. DOI: 10.1101/gr.2650004. URL: <https://doi.org/10.1101/gr.2650004>.
- [4] Almagro Armenteros, J. J. et al. "Detecting sequence signals in targeting peptides using deep learning". In: *Life Science Alliance* 2.5 (2019). DOI: 10.26508/lsa.201900429.
- [5] Horton, P. et al. "WoLF PSORT: protein localization predictor". In: *Nucleic Acids Research* 35.suppl\_2 (2007), W585–W587.
- [6] Pierleoni, A., Martelli, P. L., and Fariselli, P. "BaCellLo: a Balanced subCellular Localization predictor." In: *Bioinformatics (Oxford, England)* 22 (2006), e408–16. DOI: 10.1093/bioinformatics/btl222.
- [7] Savojardo, C. et al. "TPpred3 detects and discriminates mitochondrial and chloroplastic targeting peptides in eukaryotic proteins". In: *Bioinformatics* 31.20 (2015), 3269–3275.
- [8] Jiang, Y. et al. *MULocDeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation*. 2020.
- [9] Alley, E. et al. "Unified rational protein engineering with sequence-based deep representation learning". In: *Nature Methods* 16 (2019). DOI: 10.1038/s41592-019-0598-1.
- [10] Heinzinger, M. et al. "Modeling aspects of the language of life through transfer-learning protein sequences". In: *BMC Bioinformatics* 20 (2019).
- [11] Elnaggar, A. et al. "ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing". In: *bioRxiv* (2020).
- [12] Rives, A. et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences". In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118. DOI: 10.1073/pnas.2016239118. URL: <https://doi.org/10.1073/pnas.2016239118>.
- [13] Savojardo, C. et al. "DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks". In: *Bioinformatics* 36.1 (2019), 56–64.
- [14] Almagro Armenteros, J. J. et al. "DeepLoc: prediction of protein subcellular localization using deep learning". In: *Bioinformatics* 33.21 (2017), 3387–3395.

- [15] Ho Thanh Lam, L. et al. "Machine Learning Model for Identifying Antioxidant Proteins Using Features Calculated from Primary Sequences". In: *Biology* 9.10 (2020).
- [16] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. "In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins". In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
- [17] Wanders, R. J. A., Waterham, H. R., and Ferdinandusse, S. "Metabolic Interplay between Peroxisomes and Other Subcellular Organelles Including Mitochondria and the Endoplasmic Reticulum". In: *Frontiers in Cell and Developmental Biology* 3 (2016), 83.
- [18] Islinger, M. et al. "The peroxisome: an update on mysteries 2.0". In: *Histochemistry and Cell Biology* 150 (2018), 1–29. DOI: 10.1007/s00418-018-1722-5.
- [19] Brocard, C. and Hartig, A. "Peroxisome targeting signal 1: Is it really a simple tripeptide?" In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1763.12 (2006), 1565–1573.
- [20] Kamoshita, M. et al. "Insights Into the Peroxisomal Protein Inventory of Zebrafish". In: *Frontiers in Physiology* 13 (2022).
- [21] Cortes, C. and Vapnik, V. "Support-vector networks". In: *Machine Learning* 20.3 (1995), 273–297. DOI: 10.1007/bf00994018. URL: <https://doi.org/10.1007/bf00994018>.
- [22] Baker, A. et al. "Peroxisomal ABC transporters: functions and mechanism". In: *Biochemical Society Transactions* 43.5 (2015), 959–965.
- [23] Aitchison, J., Murray, W. W., and Rachubinski, R. "The carboxyl-terminal tripeptide Ala-Lys-Ile is essential for targeting *Candida tropicalis* trifunctional enzyme to yeast peroxisomes." In: *Journal of Biological Chemistry* 266.34 (1991), 23197–23203.
- [24] De Hoop, M. and Ab, G. "Import of proteins into peroxisomes and other microbodies." In: *Biochemical Journal* 286.Pt 3 (1992), 657.
- [25] Schlüter, A. et al. "PeroxisomeDB 2.0: an integrative view of the global peroxisomal metabolome". In: *Nucleic Acids Research* 38.suppl\_1 (2009), D800–D805.
- [26] Alex Bateman, and et al. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic acids research* 49.D1 (2021), D480–D489.
- [27] Cawley, G. C. and Talbot, N. L. C. "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation". In: *Journal of Machine Learning Research* 11.70 (2010), 2079–2107. URL: <http://jmlr.org/papers/v11/cawley10a.html>.
- [28] Filzmoser, P., Liebmann, B., and Varmuza, K. "Repeated double cross validation". In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 23.4 (2009), 160–171.
- [29] Platt, J. "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods". In: *Adv. Large Margin Classif.* 10 (2000).

- 
- [30] Refregier, P. and Vallet, F. "Probabilistic approach for multiclass classification with neural networks". In: *Artificial Neural Networks*. Elsevier, 1991, 1003–1006.
  - [31] Wu, T.-F., Lin, C.-J., and Weng, R. C. "Probability Estimates for Multi-Class Classification by Pairwise Coupling". In: *J. Mach. Learn. Res.* 5 (2004), 975–1005.
  - [32] Savojardo, C. et al. "DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks". In: *Bioinformatics* 36.1 (2020), 56–64.
  - [33] Wang, X., Jin, Y., and Zhang, Q. "Deeppred-submito: a novel submitochondrial localization predictor based on multi-channel convolutional neural network and dataset balancing treatment". In: *International Journal of Molecular Sciences* 21.16 (2020), 5710.
  - [34] Consortium, T. U. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
  - [35] Li, W. and Godzik, A. "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences". In: *Bioinformatics* 22 (2006), 1658–1659.
  - [36] Forcier, J., Bissex, P., and Chun, W. J. *Python web development with Django*. Addison-Wesley Professional, 2008.
  - [37] Matthews, B. W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), 442–451.
  - [38] Kohlwein, S. D., Veenhuis, M., and Klei, I. J. van der. "Lipid Droplets and Peroxisomes: Key Players in Cellular Lipid Homeostasis or A Matter of Fat—Store 'em Up or Burn 'em Down". In: *Genetics* 193.1 (2013), 1–50. URL: <https://doi.org/10.1534/genetics.112.143362>.
  - [39] Sibirny, A. A. "Yeast peroxisomes: structure, functions and biotechnological opportunities". In: *FEMS Yeast Research* 16.4 (2016). Ed. by J. Nielsen, fow038. DOI: 10.1093/femsyr/fow038. URL: <https://doi.org/10.1093/femsyr/fow038>.



---

# **Is-mPTS: an experimentally validated tool for Pex19 - membrane peroxisomal targeting signal (mPTS) binding domain identification from protein sequence**

5

Marco Anteghini, Chethan K. Krishna, Vishal C. Kalel, Vitor A.P. Martins dos Santos,  
Edoardo Saccenti and Ralf Erdmann

*Manuscript in preparation*

## Abstract

The targeting of peroxisomal membrane proteins (PMPs) is a complex process that involves the recognition of PMPs in the cytosol and their subsequent insertion into the peroxisomal membrane. PMP targeting signals (mPTS) play a crucial role in this process, but their intricate nature and poor definition present challenges for understanding their functionality.

In this study, we are developing a novel pipeline and leveraging advancements in machine learning and deep learning approaches to accurately predict protein sub-cellular localization. We are combining these techniques with algorithms for Peroxisome Targeting Signal (PTS) motif detection. Specifically, we are emphasizing the need for improved detection of mPTS motifs, which are essential for targeting peroxisomal membrane proteins.

To address this challenge, we are introducing a new algorithm for detecting mPTS motifs in protein sequences and experimentally validating its effectiveness. Additionally, we are proposing the development of easily accessible tools that incorporate novel approaches and recent discoveries. Our overall aim is to provide the peroxisomal research community with a reliable computational method for accurate PEX-19 binding site detection.

## 5.1 Introduction

Peroxisomes are ubiquitous cellular organelles, enclosed by a single membrane, known as ‘protective’ organelles due to their involvement in both metabolic and non-metabolic pathways [1–3]. Malfunctions in peroxisomal proteins have been associated with metabolic disorders in humans [1, 2].

Despite their significance, the full spectrum of peroxisomal functions remains largely unknown [1], and the discovery of novel peroxisomal proteins can greatly contribute to expanding our knowledge in this field [1, 2, 4].

Peroxisomal proteins are initially synthesized in the cytoplasm and then directed to the peroxisome for specific functions [5]. Unlike other organelles that primarily receive their proteins from the endoplasmic reticulum (ER), peroxisomes possess dedicated mechanisms for importing matrix proteins and inserting membrane proteins [6]. The most common PTS motif, known as PTS1, is found in many peroxisomal matrix proteins and is recognized by the PEX5 receptor [7–9]. Another PTS motif, PTS2, is present in a few matrix proteins and is recognized by the PEX7 receptor [10]. Additionally, peroxisomal membrane proteins have their own targeting motif, called the “mPTS” motif, recognized by the PEX19 receptor [11–13]. The discovery of the PEX gene family, including PEX5, PEX7, and PEX19, has significantly advanced our understanding of peroxisomal biogenesis and function, offering opportunities for further research to explore the complete range of peroxisomal functions and their significance in various cellular processes [1, 2, 14, 15].

The targeting of peroxisomal membrane proteins (PMPs) involves a complex multistep process, which includes the recognition of PMPs in the cytosol and their subsequent insertion into the peroxisomal membrane [12]. This complexity is reflected in the intricate nature of the targeting signals, known as peroxisomal mem-

brane protein targeting signals (mPTS), associated with PMPs. Pex19p, a protein of interest, has been proposed as a potential player in the recognition of PMPs, as it interacts with a majority of them [12].

Novel use-case-specific computational methods, combined with more established computational and experimental methodologies, can facilitate the discovery and annotation of new peroxisomal proteins, for example, when investigating sequences from entire proteomes [16, 17]. Recent advancements in Machine Learning (ML) and Deep Learning (DL) approaches have enabled researchers to accurately predict the subcellular and sub-organelle localization of proteins [16, 18, 19]. These techniques can be combined with algorithms that rely on Peroxisome Targeting Signal (PTS) motif detection for comprehensive analyses [20].

In Anteghini *et al.* 2023, DL approaches have been used to embed (encode) the protein sequences and predict PTS1 signals [18]. Given the analyses from recent works on DL-based embeddings, we adopted the Evolutionary Scale Modelling - 2 (ESM2) embedding [21] to represent the Pex19-binding peptides.

Classic PTS detection algorithms, utilize the consensus sequences of type 1 (PTS1) and type 2 (PTS2) signals, which target proteins into the peroxisomal matrix. However, peroxisomal membrane proteins contain membrane protein targeting signals (mPTS) that are essential for their targeting. Since mPTS signals are poorly defined, the development of novel algorithms is necessary to acquire new knowledge in this field [12].

Existing computational resources in the peroxisomal research community, such as PeroxisomeDB, have not been updated with recent discoveries, and specific servers are often non-functional [20]. Moreover, algorithms for PTS1, PTS2, and especially mPTS, have not been updated with the latest consensus motifs. Consequently, there is a need for new, easily accessible tools that incorporate novel approaches and recent discoveries to facilitate the peroxisomal research community.

In this contribution, we present our ongoing study focused on optimizing the detection of mPTS motifs in protein sequences using a new algorithm and experimental validation, which combined can form an informative pipeline. In particular, we test our tool on a data set of detected PEX-19 binding site from unpublished data.

## 5.2 Methods

### 5.2.1 Consensus motif

PEX19-binding sites showed a similar aminoacid composition [11–14, 22]. Overall, this similarity can be defined with the consensus motif ‘XXX[CFILTVW]XX[ACFILQVWY][CILV]XX[ACFILVWY][ILQRV]XXX’, where X is any aminoacid, and the letters between squared brackets represent the possible aminoacids in that position. We used the reported consensus motif for identifying putative PEX19-binding sites.

### 5.2.2 Training data set

The peptides used during the training of the Is-mPTS model, are manually curated and come with experimental validation from this study and previous publi-

cations [11–14]. Peptides from *Homo sapiens* (Human) and *S. cerevisiae* (Yeast) were used in the training set. The training set consists of 56 positives and 109 negatives.

### Human peptides

The PEX19-binding peptides from Human were retrieved from the evidences in Schueller *et al.* (2010), Emmanoulidis *et al.* (2017) and Halbach *et al.* (2005) [11, 13, 14]. In total we selected 19 positives and 29 negatives.

### Yeast peptides

The PEX19-binding peptides from Yeast were retrieved from Rottensteiner *et al.* (2004) [12]. In total we selected 37 positives and 80 negatives.

## 5.2.3 Validation data set - *trypanosoma brucei* peptides

To test the Is-mPTS model performances in a real-world scenario we validated our algorithm against 8 unpublished (but experimentally validated) PEX19-binding sites. Note that this version of the algorithm was never trained on peptides from *Trypanosoma brucei*.

## 5.2.4 The Evolutionary Scale Modelling - 2 (ESM2) embedding

The ESM2 model is a language model designed to predict high-resolution protein structures directly from the primary protein sequence. It leverages evolutionary patterns associated with protein structure, eliminating the need for external evolutionary databases, multiple sequence alignments, and templates [21]. During training, the ESM2 model is exposed to a diverse set of sequences sampled from the UniRef protein sequence database, which comprises approximately 138 million UniRef90 sequences organized into approximately 43 million UniRef50 training clusters [23, 24]. This sampling strategy ensures that the model encounters around 65 million unique sequences during training.

ESM2 is based on a transformer architecture. The transformer model is based on the self-attention technique [25]. This technique dynamically assigns varying degrees of importance to different elements of the input data, including the recursive output. Unlike recurrent neural networks (RNNs), transformers have the advantage of processing the entire input simultaneously. Through the attention mechanism, the model can effectively consider the contextual information from any position within the input sequence. This characteristic is especially beneficial when dealing with natural language sentences, as the transformer is not limited to processing one word at a time. Consequently, the transformer model exhibits improved parallelization capabilities compared to RNNs, resulting in reduced training times [25–27].

The ESM2 model achieves high-resolution accuracy in three-dimensional structure prediction while significantly improving the speed of structure prediction [28, 29]. The model is trained using a masked language modelling objective, which requires it to capture dependencies between masked and unmasked parts of the sequence to make accurate predictions [29, 30]. The training process for ESM2 in-



volves scaling the model size from 8 million parameters to as large as 15 billion parameters. Notably, the inclusion of a query and key vector within the self-attention mechanism, along with sinusoidal embedding, enhances the quality of the model, particularly for smaller architectures. The resulting ESM2 vector has a length of 1280.

In this study, we adopted a per-residue embedding representation, where each peptide is embedded using a matrix with dimensions of 1280 times the number of aminoacids in the sequence. This approach allows for comprehensive encoding of the peptide characteristics and facilitates downstream analysis and prediction tasks.

### 5.2.5 Classification algorithms

The determination of true and false PEX19-binding sites is easily translated into a binary classification problem. We considered four widely used classification algorithms.

**Support Vector Machines (SVM)** is a supervised learning algorithm for two-group classification which aims to find the maximal margin hyperplane separating the points in the feature space [31, 32]. SVMs also perform non-linear classifications applying the kernel trick, thus implicitly mapping their inputs into high-dimensional feature spaces. In the case of multiple classes, multiple binary classification problems are performed. It can be done in two ways [33]: 1) *One-vs-One*, a binary classifier per each pair of classes; 2) *One-vs-Rest*, a binary classifier per class. In this study, we used the One-vs-Rest approach.

**Random Forest (RF)** is an ensemble learning method that, in the case of a classification task constructs a multitude of decision trees and outputs the mode of the classes of the individual trees [34, 35].

**Multilayer Perceptron (MLP)** is a class of feed-forward artificial neural networks that can distinguish among non-linearly separable data and uses backpropagation for training [36, 37]. Each node in an MLP, with the exception of the input node, uses a nonlinear activation function. In this study, we used the ReLu activation function [38].

**Logistic Regression (LR)** estimates the parameters of a logistic model [39]. In binary classifications, the corresponding probability of the values associated with two different labels can vary between 0 and 1. The multinomial LR model, for K possible outcomes, runs K-1 independent binary logistic regression models, in which one outcome is chosen as a "pivot" and then the other K-1 outcomes are separately regressed against the pivot outcome. We used a penalised implementation of multivariable logistic regression [40].

### 5.2.6 Experimental score

We integrated the Is-mPTS model together with an experimentally defined score to build a comprehensive pipeline. The score relies on the matrix presented in Rotten-

Table 5.1: Hyperparameters for the grid searches. The `logspace` function, as available on NumPy [41] returns numbers spaced evenly on a log scale. In `logspace(start,stop,numbers)`, the sequence starts at base *start* (base to the power of start), ends with base *stop* and *numbers* is the number of samples to generate. The listed methods are: Support Vector Machines (SVM); Random Forest (RF); Multilayer Perceptron (MLP); Logistic Regression (LR)

method	hyperparameters	description	search space
SVM	C	Inverse of regularization strength	<code>logspace(-2,10,13)</code>
	gamma	Kernel coefficient	<code>logspace(-9,3,13)</code>
	kernel	Specifies the kernel type to be used in the algorithm	<code>'linear','poly','rbf','sigmoid'</code>
RF	n_estimators	The number of trees in the forest	15,25,50,75,100,200,300
	criterion	The function to measure the quality of a split	<code>'gini','entropy'</code>
	max_depth	The maximum depth of the tree	2,5,10,None
	min_samples_split	The minimum number of samples required to split an internal node	2,4,8,10
MLP	max_features	The number of features to consider when looking for the best split	<code>'sqrt','auto','log2'</code>
	hidden_layer_sizes	The number of neurons in each hidden layer	(50,),(100,),(150,),(200,),(100,50,)
	activation	Activation function for the hidden layer	<code>'relu'</code>
	solver	The solver for weight optimization	<code>'lbfgs'</code>
LR	alpha	Strength of the L2 regularization term	1.0
	learning_rate	Learning rate schedule for weight updates	<code>'constant'</code>
	penalty	Specify the norm of the penalty	<code>'l1','l2'</code>
	solver	Algorithm to use in the optimization problem	<code>'liblinear','saga'</code>
	C	Inverse of regularization strength	<code>logspace(-3,9,13)</code>

steiner *et al.* (2004) [12] and further reported in Table 5.2. Note that the table focuses on the 9 central positions (aminoacids) of the mPTS motif. The raw prediction score of a peptide was calculated as the exponential of the sum of nine respective aminoacid scores at specific positions as reported in the following function:

$$\text{raw score} = \exp\left(\sum_{i=1}^9 a_i\right) \quad (5.1)$$

where  $a$  is the experimental score of an aminoacid, corresponding to the intensity pick. The intensities were quantified using the LumiImager (Roche, Basel, Switzerland).

### Experimental scores normalization

Given the high variability of the reported scores, in this study, we normalized them (Table 5.3) using the following function:

$$z = \frac{a - \min(a)}{\max(a) - \min(a)} \quad (5.2)$$

where  $z$  is the normalized score and  $a$  is the experimental score of an aminoacid (from Table 5.2).

### Experimental score analysis

We downloaded the yeast proteome (*Saccharomyces cerevisiae* - strain ATCC 204508 / S288c) from Uniprot (<https://www.uniprot.org/proteomes/UP000002311>) [42]. We analyzed the matching peptides computing the experimental score for each of them.

Table 5.2: Scores for different aminoacids at nine positions. The raw prediction score of a peptide is the exponential of the sum of nine respective aminoacid scores. Four preceding and two succeeding aminoacids in 15-mer peptides do not contribute to the score. Rows represent aminoacids, and columns represent aminoacid positions.

	1	2	3	4	5	6	7	8	9
A	0.253	-0.193	0.437	-0.490	2.637	0.187	0.400	1.020	0.231
C	0.267	-0.628	-1.308	0.733	2.429	0.092	0.003	0.833	0.182
D	0.265	-1.068	-0.924	-0.510	-1.160	-0.295	0.029	-0.983	-0.684
E	-0.181	-0.386	0.064	-0.046	-1.755	-1.857	0.089	-0.056	-0.490
F	1.582	0.992	-0.601	0.839	2.072	0.608	0.388	1.184	0.144
G	-0.980	-0.317	-0.660	-0.762	-1.796	-0.064	-0.585	-0.951	-0.395
H	-0.762	0.022	0.490	-0.248	-2.261	-0.114	0.148	0.309	-0.263
I	1.178	0.686	0.272	0.936	2.870	0.581	0.013	1.102	0.717
K	-2.758	0.521	0.557	-0.642	-2.226	0.441	0.271	-1.483	-0.293
L	1.657	0.749	0.185	1.201	2.927	0.693	0.615	1.473	0.900
M	0.243	0.499	-0.221	-0.025	-0.012	-0.179	0.083	-0.011	-0.497
N	-0.453	-2.062	-0.521	-0.662	-1.504	-1.218	-1.135	-0.166	-0.283
P	-1.919	-0.590	-0.246	-3.189	-1.771	-1.176	-0.397	-3.798	-1.060
Q	-0.841	0.340	0.066	0.574	-2.149	0.259	0.420	0.408	0.922
R	-0.470	0.648	0.695	-0.428	-1.471	0.263	0.362	0.101	1.011
S	-0.258	-0.357	0.822	0.115	-0.997	0.179	0.179	0.253	0.099
T	1.284	0.067	0.290	0.239	-0.026	0.174	-0.227	0.126	-0.091
V	0.977	0.777	0.184	1.263	2.767	0.778	-0.122	0.803	0.695
W	0.754	0.372	0.426	0.802	0.976	0.612	0.113	1.154	-0.304
Y	0.163	-0.072	-0.005	0.299	0.448	0.035	-0.647	-1.317	-0.541

Table 5.3: Normalized scores for different aminoacids at nine positions. Rows represent aminoacids, and columns represent aminoacid positions.

	1	2	3	4	5	6	7	8	9
<b>A</b>	0.238	0.095	0.296	0.000	1.000	0.217	0.285	0.483	0.231
<b>C</b>	0.421	0.182	0.000	0.546	1.000	0.375	0.351	0.573	0.399
<b>D</b>	1.000	0.065	0.166	0.456	0.000	0.607	0.834	0.124	0.334
<b>E</b>	0.861	0.756	0.987	0.931	0.052	0.000	1.000	0.925	0.702
<b>F</b>	0.817	0.596	0.000	0.539	1.000	0.452	0.370	0.668	0.279
<b>G</b>	0.471	0.854	0.656	0.597	0.000	1.000	0.699	0.488	0.809
<b>H</b>	0.545	0.830	1.000	0.732	0.000	0.780	0.876	0.934	0.726
<b>I</b>	0.408	0.236	0.091	0.323	1.000	0.199	0.000	0.381	0.246
<b>K</b>	0.000	0.989	1.000	0.638	0.160	0.965	0.914	0.385	0.744
<b>L</b>	0.537	0.206	0.000	0.371	1.000	0.185	0.157	0.470	0.261
<b>M</b>	0.743	1.000	0.277	0.474	0.487	0.319	0.582	0.488	0.000
<b>N</b>	0.849	0.000	0.813	0.738	0.294	0.445	0.489	1.000	0.938
<b>P</b>	0.529	0.903	1.000	0.171	0.571	0.738	0.957	0.000	0.771
<b>Q</b>	0.426	0.810	0.721	0.887	0.000	0.784	0.837	0.833	1.000
<b>R</b>	0.403	0.854	0.873	0.420	0.000	0.699	0.739	0.633	1.000
<b>S</b>	0.406	0.352	1.000	0.611	0.000	0.647	0.647	0.687	0.603
<b>T</b>	1.000	0.195	0.342	0.308	0.133	0.265	0.000	0.234	0.090
<b>V</b>	0.380	0.311	0.106	0.479	1.000	0.312	0.000	0.320	0.283
<b>W</b>	0.726	0.464	0.501	0.759	0.878	0.628	0.286	1.000	0.000
<b>Y</b>	0.839	0.705	0.743	0.916	1.000	0.766	0.380	0.000	0.440

### 5.2.7 Model training and validation

**Training.** Each model was evaluated on the training data sets respectively using 10-fold cross-validation (10-CV) [43]. In every iteration, a single fold was kept as the testing set, and the remaining nine sets were used to train the respective model. The trained model was then tested using the test set. The procedure stops when all 10 subsets are used as a test once. The average performance for each model was considered as a single estimation. To obtain a stable error estimation, we repeated the 10-CV ten times with different random splits. The variations between runs were highlighted by the standard deviation. The cross-validation performances are reported as mean  $\pm$  standard deviation (SD) of the ten different runs of the 10-CV.

The cross-validation procedures include a (hyper)grid search: for each set of hyperparameters, the average classification score is computed across the folds. The hyperparameters corresponding to the best classification score are then used to fit a classification model whose quality is assessed on the validation set. The reference metric is the  $F_1$  score. The Hyperparameters optimisation details are shown in Table 5.1).

**Validation.** The *Trypanosoma brucei* data sets was used to perform additional validations (Section 5.2.3). The data in the independent validation set were not used during the cross-validation processes and are completely unknown to the models.

### 5.2.8 Evaluation Metrics

To assess the quality of the classification models, we employed several evaluation metrics: sensitivity (SEN), specificity (SPE), accuracy (ACC), F1 score [44], Matthews correlation coefficient (MCC) [45], and the area under the curve (AUC) of the receiver operating characteristic (ROC). The formulas for these metrics are defined as follows:

Sensitivity (*SEN*) or True Positive Rate (TPR) is calculated as

$$SEN/TPR = \frac{TP}{TP + FN} \quad (5.3)$$

Specificity (*SPE*) is calculated as

$$SPE = \frac{TN}{TN + FP} \quad (5.4)$$

Accuracy (*ACC*) is calculated as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.5)$$

$F_1$  score [44] is defined as

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}, \quad (5.6)$$

where *PPV* represents the Positive Predicted Value (or precision), defined as

$$PPV = \frac{TP}{TP + FP}, \quad (5.7)$$

The  $F_1$  score is the harmonic mean of recall and precision and ranges between 0 (indicating no precision or recall) and 1 (indicating perfect precision and recall).

Matthews correlation coefficient (MCC) [45] is calculated as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (5.8)$$

The MCC measures the correlation coefficient between the true and predicted classes, ranging from -1 (indicating total disagreement) to +1 (indicating perfect prediction), with 0 indicating random prediction. The MCC is suitable for evaluating performance in the presence of class imbalance [46].

The area under the curve (AUC) of the receiver operating characteristic (ROC) curve, which plots the true positive proportion or sensitivity against the false positive proportion or specificity, is defined as

$$AUC = \int_0^1 TPRd(FPR) \quad (5.9)$$

The AUC analysis enables the evaluation of binary classifier performance by considering variations in the discrimination threshold. A perfect prediction corresponds to an AUC score of 1.0, while an AUC of 0.5 indicates randomness [47].

## 5.3 Results

### 5.3.1 Is-mPTS pipeline development

The Is-mPTS pipeline takes a FASTA sequence (or multiple sequences) as input and performs a motif search. The motif search relies on detecting the putative PEX19-binding sites matching with the consensus motif

‘XXX[CFILTVW]XX[ACFILQVWY][CILV]XX[ACFILVWY][ILQRV]XXX’. Once the matching motifs, with a length of 15 aminoacids, are found, the protein embeddings are generated and submitted to the Is-mPTS model.

The Is-mPTS model calculates a probability score for each matching peptide, indicating the likelihood of it being an actual PEX19 binding site. The Is-mPTS model is built by selecting the best model out of four classifiers: logistic regression (LR), random forest (RF), support vector machines (SVM), and multilayer perceptron (MLP). Each model undergoes a 10-fold cross-validation using a training set that contains experimentally detected binding sites. The peptides in the training set are embedded using the ESM2 embedding. Additionally, an additional score based on experimental data is incorporated into the Is-mPTS pipeline.

As a result, the Is-mPTS pipeline outputs two scores. First, the classifier distinguishes between true and false PEX19-binding sites using the Is-mPTS model. Secondly, the information of the experimentally based score is added. The overview is visible in Figure 5.1.

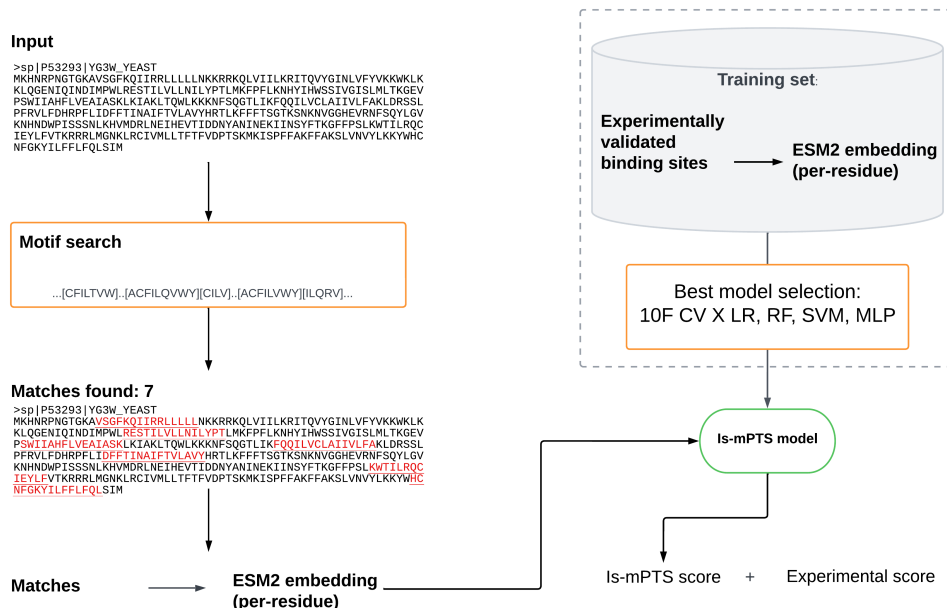


Figure 5.1: Overview of the Is-mPTS pipeline. It takes as input a FASTA sequence (or more) and performs a motif search. Once the matching motifs are found (15 aminoacids length), the protein embeddings are generated and submitted to the Is-mPTS model to obtain a probability score (for each matching peptide) which informs about the probability of a given peptide being an actual PEX19 binding site. The top right corner (in the dashed line square) shows the procedure to generate the Is-mPTS model which consists of selecting the best model out of four classifiers, namely: logistic regression (LR), random forest (RF), support vector machines (SVM), and multilayer perceptron (MLP). For each model, a 10-fold cross-validation (10F CV) is performed. The adopted training set contains experimentally detected binding sites which are then embedded using the ESM2 embedding. Finally, an additional score (based on experimental data) is added.

### 5.3.2 PEX-19 binding sites prediction based on the ESM2 embedding

The performance of four machine learning (ML) classifiers, namely logistic regression (LR), multilayer perceptron (MLP), random forest (RF), and support vector machine (SVM), was evaluated for predicting PEX-19 binding sites. The peptides used for the training set are reported in Section 5.2.2.

Table 5.4 presents the results obtained from these classifiers in terms of various performance metrics. The accuracy (ACC) values ranged from 0.8001 to 0.8177, indicating the overall correctness of the predictions. The balanced accuracy (BACC) scores, which consider imbalanced class distributions, ranged from 0.7413 to 0.7964, highlighting the classifiers' ability to perform well across different classes. The Matthew correlation coefficient (MCC) values ranged from 0.5559 to 0.6058, providing an assessment of the classifiers' overall performance that considers true positives, true negatives, false positives, and false negatives. The area under the receiver operating characteristic curve (AUC) scores, which measure the classifiers' ability to distinguish between positive and negative samples, ranged from 0.7413 to 0.7964, with higher values indicating better discrimination. The  $F_1$  scores, combining precision and recall, ranged from 0.7487 to 0.7936, reflecting the classifiers' ability to achieve a balance between these two metrics. Specificity (SPE) values ranged from 0.8394 to 0.9433, indicating the classifiers' ability to correctly identify negative samples, while sensitivity (SEN) values ranged from 0.5393 to 0.7277, representing the classifiers' ability to correctly identify positive samples.

Overall, these results demonstrate the effectiveness of the ML classifiers in predicting PEX-19 binding sites and provide valuable insights into their performance across different evaluation metrics. The best performances were obtained with the MLP classifier.

### 5.3.3 Experimental score analysis on *S. cerevisiae* proteome

We analyzed the matching peptides computing the experimental score for each of them. The results of the matching peptides which overlapped with our yeast training set of positives (Section 5.2.2) are shown in Table 5.5. We can notice that the scores range from 0.54 to 0.89, with 15 peptides presenting a score  $\geq 0.78$ . Only 2 have a lower score.

We then repeated the analysis for the peptides present in our negative dataset of yeast peptides (80 negatives). We observed that none of these peptides had a score greater than or equal to 0.78. Therefore, in the yeast proteome, false positives do not have a score of 0.78 or higher.

### 5.3.4 PEX-19 binding sites prediction on the validation set

After generating the embeddings of the peptides in the validation set (Section 5.2.3), we computed the experimental score and the Is-mPTS score for each peptide. Results are visible in Table 5.2

Considering just the Is-mPTS score, we can notice that 6 out of 8 peptides were correctly predicted (accuracy of 0.75). Results are visible in Figure 5.2. Despite some disparities between the Is-mPTS scores and experimental scores there is a high level



Table 5.4: Performance of the four machine learning (ML) classifiers analysed for predicting PEX-19 bunding sites. The ML classifiers are logistic regression (LR), multilayer perceptron (MLP), random forest (RF) and support vector machine (SVM). The scores are reported in terms of accuracy (ACC), balanced accuracy (BACC), matthew correlation coefficient (MCC), area under the receiver operating characteristic curve (AUC),  $F_1$  score, specificity (SPE) and sensitivity (SEN).

(a) LR		(b) MLP	
metrics	score	metrics	score
ACC	$0.81 \pm 0.01$	ACC	$0.82 \pm 0.01$
BACC	$0.78 \pm 0.01$	BACC	$0.80 \pm 0.01$
MCC	$0.58 \pm 0.03$	MCC	$0.61 \pm 0.03$
AUC	$0.78 \pm 0.01$	AUC	$0.80 \pm 0.01$
$F_1$	$0.78 \pm 0.01$	$F_1$	$0.79 \pm 0.01$
SPE	$0.85 \pm 0.02$	SPE	$0.86 \pm 0.03$
SEN	$0.72 \pm 0.02$	SEN	$0.73 \pm 0.05$

(c) RF		(d) SVM	
metrics	score	metrics	score
ACC	$0.81 \pm 0.01$	ACC	$0.80 \pm 0.01$
BACC	$0.74 \pm 0.02$	BACC	$0.78 \pm 0.01$
MCC	$0.56 \pm 0.03$	MCC	$0.57 \pm 0.02$
AUC	$0.74 \pm 0.02$	AUC	$0.78 \pm 0.01$
$F_1$	$0.75 \pm 0.02$	$F_1$	$0.78 \pm 0.01$
SPE	$0.94 \pm 0.02$	SPE	$0.84 \pm 0.01$
SEN	$0.54 \pm 0.05$	SEN	$0.72 \pm 0.02$

Table 5.5: Experimental scores of 15 aa peptides for the matching yeast peptides.

protein ID	gene name	15 aa peptide	experimental score
P32800	PEX2	GSALSGVVFQCRKRT	0.76
P40155	PEX17	LLRLRRIIAQLQKRL	0.89
Q06497	ANT1	GVIVQGLLFAFRGEL	0.78
Q08580	PEX27	LQHTLGLLSLLLLTR	0.82
Q08580	PEX27	LRLVIQQLSLFRYYL	0.80
Q08580	PEX27	AINLYKIIKRFRWLR	0.84
Q04370	PEX12	FLRIYPIFKLLALS	0.76
Q04370	PEX12	SQFFPTFIFVLRVYQ	0.78
P41909	PXA1	KVYFKLLIRHLLQIS	0.85
P41909	PXA1	FLLLTAQIFFLVMRT	0.83
P41909	PXA1	GIFVNYFITGFILRK	0.54
Q02969	PEX25	ITVIKVLSSLLRNF	0.86
Q02969	PEX25	GSGLTGLVKLWITTK	0.78
P80667	PEX13	IFAIMKFLKKILYRA	0.83
P33760	PEX6	NMGCVRLVKLFVLLL	0.82
P28795	PEX3	LFTTGSVVVFFVKRW	0.84
P34230	PXA2	KKCLILFITQAILLN	0.82
Q12462	PEX11	VLRLQLYLARFLAVQ	0.80

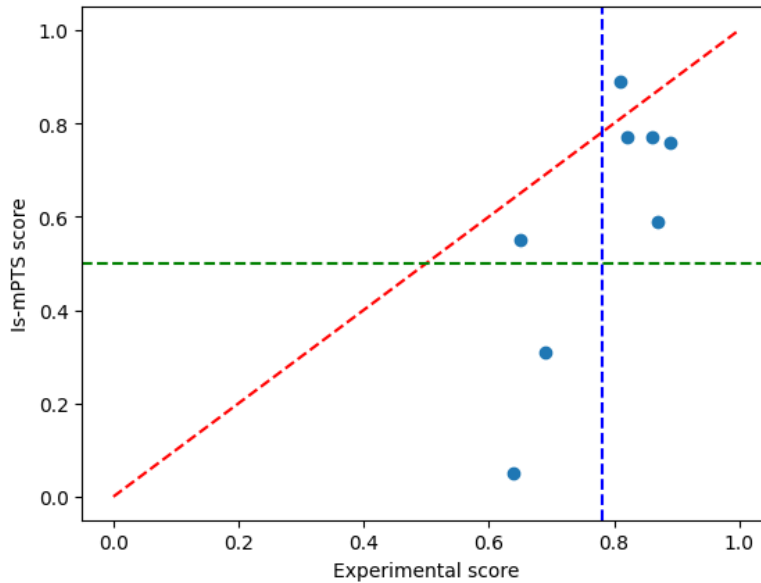


Figure 5.2: Experimental scores (x axis) and corresponding Is-mPTS scores (y axis) for the 8 peptides in the validation set. Point closer to the red diagonal show similar values between experimental score and Is-mPTS score. The blue vertical line represent the threshold of 0.78 for the experimental score. The green horizontal line represent the threshold of 0.50 for the Is-mPTS score. The majority of the dots are in the upper right square, showing high correspondence between true prediction.

of agreement between the Is-mPTS scores and the experimental scores. Overall, these results highlight the potential of the Is-mPTS score as a predictive metric for PEX-19 binding sites. However, the variations observed between the Is-mPTS scores and experimental scores underscore the importance of considering other factors and validation methods to ensure accurate predictions.

## 5.4 Discussion

The objective of this study was to develop a computational tool, called Is-mPTS, for accurately predicting PEX-19 binding sites. We integrated experimentally derived measurements from previous studies with a novel classification algorithm to design the Is-mPTS pipeline. Our novel model achieved an accuracy of 0.82.

We further conducted an experimental score analysis using the yeast proteome. The experimental scores for the matching peptides ranged from 0.54 to 0.89, with the majority having scores equal to or higher than 0.78.

Notably, the predictions on the validation set, which included peptides from an organism (*trypanosoma brucei*) not included in the training set, were mostly accurate (0.75). These results suggest that the algorithm can be extended to different organisms.

However, it is important to acknowledge that there were some disparities between the Is-mPTS scores and the experimental scores. This highlights the need to consider other factors and validation methods to ensure accurate predictions.

In general, we propose to select candidates with an experimental score  $\geq 0.78$  and an Is-mPTS score  $> 0.5$  to minimize the false positives, thus reducing the cost of experimental validations.

To provide further guidance to future users of our pipelines, additional analyses are required. We are currently analyzing the predictions of our pipeline against the human proteome and conducting validations and experiments to visualize the effective binding signals from the predicted peptides.

Using this preliminary version of the pipeline allowed us to discover novel PEX19-binding sites, which will be presented in future publications.

In conclusion, our study demonstrates the effectiveness of machine learning techniques, specifically DL-based embeddings and MLP classifiers, in predicting PEX-19 binding sites. The Is-mPTS score shows promise as a predictive metric, although further validation and refinement are necessary. This work contributes to our understanding of PEX-19 binding sites and lays the foundation for future studies on protein localization and protein-protein interactions in peroxisomal biogenesis.

## Acknowledgement

This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Islinger, M. et al. "The peroxisome: an update on mysteries 2.0". In: *Histochemistry and Cell Biology* 150 (2018), 1–29. DOI: 10.1007/s00418-018-1722-5.
- [2] Wanders, R. J. A., Waterham, H. R., and Ferdinandusse, S. "Metabolic Interplay between Peroxisomes and Other Subcellular Organelles Including Mitochondria and the Endoplasmic Reticulum". In: *Frontiers in Cell and Developmental Biology* 3 (2016), 83.
- [3] Schrader, M., Kamoshita, M., and Islinger, M. "Organelle interplay - peroxisome interactions in health and disease". In: *Journal of Inherited Metabolic Disease* 43.1 (2019), 71–89. DOI: 10.1002/jimd.12083. URL: <https://doi.org/10.1002/jimd.12083>.
- [4] Wanders, R. J. A. et al. "The physiological functions of human peroxisomes". In: *Physiological Reviews* 103.1 (2023), 957–1024. DOI: 10.1152/physrev.00051.2021. URL: <https://doi.org/10.1152/physrev.00051.2021>.
- [5] Gould, S. J. and Collins, C. S. "Peroxisomal-protein import: is it really that complex?" In: *Nature Reviews Molecular Cell Biology* 3.5 (2002), 382–389. DOI: 10.1038/nrm807. URL: <https://doi.org/10.1038/nrm807>.
- [6] Kim, P. and Hettema, E. "Multiple Pathways for Protein Transport to Peroxisomes". In: *Journal of Molecular Biology* 427.6 (2015), 1176–1190. DOI: 10.1016/j.jmb.2015.02.005. URL: <https://doi.org/10.1016/j.jmb.2015.02.005>.
- [7] Gould, S. G., Keller, G. A., and Subramani, S. "Identification of a peroxisomal targeting signal at the carboxy terminus of firefly luciferase." In: *Journal of Cell Biology* 105.6 (1987), 2923–2931.
- [8] Kiel, J. A. et al. "Ubiquitination of the Peroxisomal Targeting Signal Type 1 Receptor, Pex5p, Suggests the Presence of a Quality Control Mechanism during Peroxisomal Matrix Protein Import". In: *Journal of Biological Chemistry* 280.3 (2005), 1921–1930.
- [9] Ghosh, D. and Berg, J. M. "A Proteome-Wide Perspective on Peroxisome Targeting Signal 1(PTS1)-Pex5p Affinities". In: *Journal of the American Chemical Society* 132.11 (2010), 3973–3979. DOI: 10.1021/ja9109049. URL: <https://doi.org/10.1021/ja9109049>.
- [10] Kunze, M. "The type-2 peroxisomal targeting signal". In: *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research* 1867.2 (2020), 118609.
- [11] Halbach, A. et al. "Function of the PEX19-binding Site of Human Adrenoleukodystrophy Protein as Targeting Motif in Man and Yeast". In: *Journal of Biological Chemistry* 280.22 (2005), 21176–21182. DOI: 10.1074/jbc.m501750200. URL: <https://doi.org/10.1074/jbc.m501750200>.
- [12] Rottensteiner, H. et al. "Peroxisomal Membrane Proteins Contain Common Pex19p-binding Sites that Are an Integral Part of Their Targeting Signals". In: *Molecular Biology of the Cell* 15.7 (2004), 3406–3417. DOI: 10.1091/mbc.e04-03-0188. URL: <https://doi.org/10.1091/mbc.e04-03-0188>.

- 
- [13] Schueller, N. et al. “The peroxisomal receptor Pex19p forms a helical mPTS recognition domain”. In: *The EMBO Journal* 29.15 (2010), 2491–2500. DOI: 10.1038/emboj.2010.115. URL: <https://doi.org/10.1038/emboj.2010.115>.
  - [14] Emmanouilidis, L. et al. “Allosteric modulation of peroxisomal membrane protein recognition by farnesylation of the peroxisomal import receptor PEX19”. In: *Nature Communications* 8.1 (2017). DOI: 10.1038/ncomms14635. URL: <https://doi.org/10.1038/ncomms14635>.
  - [15] Jansen, R. L. M. et al. “Comparative Genomics of Peroxisome Biogenesis Proteins: Making Sense of the PEX Proteins”. In: *Frontiers in Cell and Developmental Biology* 9 (2021). DOI: 10.3389/fcell.2021.654163. URL: <https://doi.org/10.3389/fcell.2021.654163>.
  - [16] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. “In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins”. In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
  - [17] Kamoshita, M. et al. “Insights Into the Peroxisomal Protein Inventory of Zebrafish”. In: *Frontiers in Physiology* 13 (2022).
  - [18] Anteghini, M. et al. “OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection”. In: *Computational and Structural Biotechnology Journal* 21 (2022), 128–133. DOI: 10.1016/j.csbj.2022.11.058. URL: <https://doi.org/10.1016/j.csbj.2022.11.058>.
  - [19] Savojardo, C. et al. “DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks”. In: *Bioinformatics* 36.1 (2020), 56–64.
  - [20] Schlüter, A. et al. “PeroxisomeDB 2.0: an integrative view of the global peroxisomal metabolome”. In: *Nucleic Acids Research* 38.suppl\_1 (2009), D800–D805.
  - [21] Lin, Z. et al. “Evolutionary-scale prediction of atomic-level protein structure with a language model”. In: *Science* 379.6637 (2023), 1123–1130. DOI: 10.1126/science.ade2574. URL: <https://doi.org/10.1126/science.ade2574>.
  - [22] Saveria, T. et al. “Conservation of PEX19-Binding Motifs Required for Protein Targeting to Mammalian Peroxisomal and Trypanosome Glycosomal Membranes”. In: *Eukaryotic Cell* 6.8 (2007), 1439–1449. DOI: 10.1128/ec.00084-07. URL: <https://doi.org/10.1128/ec.00084-07>.
  - [23] Suzek, B. E. et al. “UniRef: comprehensive and non-redundant UniProt reference clusters”. In: *Bioinformatics* 23.10 (2007), 1282–1288.
  - [24] Suzek, B. E. et al. “UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches”. In: *Bioinformatics* 31.6 (2014), 926–932.
  - [25] Vaswani, A. et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, 5998–6008.
  - [26] Radford, A. et al. “Language Models are Unsupervised Multitask Learners”. In: 2019.

- [27] Brown, T. B. et al. “Language Models are Few-Shot Learners”. In: *ArXiv abs/2005.14165* (2020).
- [28] Devlin, J. et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *CoRR abs/1810.04805* (2018).
- [29] Rives, A. et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021), e2016239118. DOI: 10.1073/pnas.2016239118. URL: <https://doi.org/10.1073/pnas.2016239118>.
- [30] Harris, Z. S. “Distributional structure”. In: *Word* 10.2-3 (1954), 146–162.
- [31] Boser, B. E., Guyon, I. M., and Vapnik, V. N. “A Training Algorithm for Optimal Margin Classifiers”. In: *COLT ’92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992*, 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401. URL: <https://doi.org/10.1145/130385.130401>.
- [32] Cristianini, N. and Ricci, E. “Support Vector Machines”. In: *Encyclopedia of Algorithms*. Boston, MA: Springer US, 2008, 928–932. ISBN: 978-0-387-30162-4. DOI: 10.1007/978-0-387-30162-4\_415. URL: [https://doi.org/10.1007/978-0-387-30162-4\\_415](https://doi.org/10.1007/978-0-387-30162-4_415).
- [33] Seliya, N., Zadeh, A. A., and Khoshgoftaar, T. M. “A literature review on one-class classification and its potential applications in big data”. In: *Journal of Big Data* 8.1 (2021). DOI: 10.1186/s40537-021-00514-x. URL: <https://doi.org/10.1186/s40537-021-00514-x>.
- [34] Tin Kam Ho. “The random subspace method for constructing decision forests”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), 832–844. DOI: 10.1109/34.709601.
- [35] Breiman, L. “Random forests”. In: *Machine learning* 45.1 (2001), 5–32.
- [36] Murtagh, F. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2.5-6 (1991), 183–197. DOI: 10.1016/0925-2312(91)90023-5. URL: [https://doi.org/10.1016/0925-2312\(91\)90023-5](https://doi.org/10.1016/0925-2312(91)90023-5).
- [37] Linnainmaa, S. “Taylor expansion of the accumulated rounding error”. In: *BIT* 16.2 (1976), 146–160. DOI: 10.1007/bf01931367. URL: <https://doi.org/10.1007/bf01931367>.
- [38] Fukushima, K. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (1980), 193–202. DOI: 10.1007/bf00344251. URL: <https://doi.org/10.1007/bf00344251>.
- [39] Tolles, J. and Meurer, W. J. “Logistic Regression”. In: *JAMA* 316.5 (2016), 533. DOI: 10.1001/jama.2016.7653. URL: <https://doi.org/10.1001/jama.2016.7653>.
- [40] Cramer, J. “The Origins of Logistic Regression”. In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (2002). DOI: 10.2139/ssrn.360300.
- [41] Harris, C. R. et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), 357–362. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.

- 
- [42] Consortium, T. U. “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
- [43] Stone, M. “Cross-Validatory Choice and Assessment of Statistical Predictions”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2 (1974), 111–133. DOI: <https://doi.org/10.1111/j.2517-6161.1974.tb00994.x>. URL: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x>.
- [44] Rijsbergen, C. J. V. *Information Retrieval*. 2nd. Butterworth-Heinemann, 1979.
- [45] Matthews, B. W. “Comparison of the predicted and observed secondary structure of T4 phage lysozyme”. In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), 442–451.
- [46] Boughorbel, S., Jarray, F., and El-Anbari, M. “Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric”. In: *PloS one* 12.6 (2017), e0177678.
- [47] Melo, F. “Area under the ROC Curve”. In: *Encyclopedia of Systems Biology*. Springer New York, 2013, 38–39. DOI: 10.1007/978-1-4419-9863-7\_209. URL: [https://doi.org/10.1007/978-1-4419-9863-7\\_209](https://doi.org/10.1007/978-1-4419-9863-7_209).





---

# **P-PPI: accurate prediction of peroxisomal protein-protein interactions using deep learning-based protein sequence embeddings**

## Abstract

Protein-protein interactions (PPIs) are crucial for various biological processes, and their prediction is typically accomplished through experimental methods, which can be time-consuming and costly. Computational methods provide a faster and more cost-effective approach, leveraging protein sequences and other data sources to infer PPIs. Deep learning (DL) approaches have shown promising results in various protein-related tasks, including PPI prediction. However, DL-based embeddings are often not thoroughly compared or evaluated against state-of-the-art tools. Additionally, existing PPI predictors incorporate different types of information beyond protein sequence representation, making it important to assess the effectiveness of DL-based embeddings solely relying on protein sequences. In this work, we benchmark and compare commonly used DL-based embeddings for PPI prediction based solely on protein sequence information. We utilize high-quality training data, including experimentally validated negative interactions from the Negatome database. The best model, obtained through double cross-validation and hyperparameter optimization, is selected and evaluated to predict peroxisomal PPIs. The resulting tool, P-PPI, is further enhanced by combining AlphaFold2-Multimer predictions with the P-PPI model, leveraging DL-based embeddings and protein structure predictions for a comprehensive analysis of peroxisomal PPIs. This integrated approach holds significant potential to advance our understanding of complex protein networks and their functions.

## 6.1 Introduction

Protein-protein interactions (PPIs) are specific physical contacts that occur between two or more proteins in a cell or living organism. They involve biochemical events where interactions such as electrostatic forces, hydrogen bonding, and the hydrophobic effect take place [1]. PPIs play crucial roles in various biological processes, including signal transduction (a chemical or physical signal is transmitted through a cell as a series of molecular events), enzymatic reactions [2], and gene regulation [3].

Protein-protein interaction can be determined by experimental methods that include techniques such as:

**Yeast two-hybrid (Y2H)** assay [4]. This technique involves the use of a yeast strain engineered to express two hybrid proteins - a DNA-binding domain (DBD) and an activation domain (AD) - fused to the proteins of interest. If the proteins interact, the DBD and AD come in close proximity, leading to the activation of a reporter gene;

**Co-immunoprecipitation (Co-IP)** [5]. It relies on the immunoprecipitation of a target protein using an antibody specific to that protein. If interacting proteins are present in the sample, they can be co-immunoprecipitated and detected using techniques such as Western blotting or mass spectrometry;

**Affinity purification coupled with mass spectrometry (AP-MS)** [6]. It is a method that analyzes tagged proteins expressed in cells. It entails purifying the tagged proteins and their associated binding partners through affinity chromatography. Fol-

---

lowing purification, the protein complexes are eluted and separated using gel electrophoresis or liquid chromatography. Mass spectrometry is then employed to identify the proteins;

**Fluorescence Resonance Energy Transfer (FRET)** [7]. Proteins of interest are labeled with fluorophores, and when they interact, energy transfer occurs, leading to a measurable change in fluorescence.

While these experimental methods provide valuable information, they are often time-consuming, expensive, and limited to specific experimental conditions.

Computational methods offer a faster and more cost-effective approach to predict PPIs, in particular if benchmarked against experimental data. These methods leverage various data sources, including protein sequences, structures, evolutionary information, and functional annotations, to infer protein interactions.

The application of deep learning (DL) approaches to encode protein sequences has shown promising results in several tasks such as subcellular [8] and sub-organelle [9–11] classification, protein structure [12, 13] and function prediction [14–17], and protein-protein interactions (PPI) prediction [18, 19].

However, these embeddings are often not compared and evaluated against themselves for highly specific tasks and are not compared against the state-of-the-art tools available. In particular, PPI prediction tools are well-established and already provide highly accurate predictions, but they often utilize different types of information (e.g., structural, evolutionary) or complex algorithms beyond protein sequence representation [18, 20–25]. DL-based embedding predictors offer several advantages. Firstly, they do not require lengthy pre-processing steps, as only the FASTA sequence is necessary for input [9, 10, 17]. This simplifies the prediction process and saves time. Additionally, DL-based embeddings are pre-computed, making the overall procedure computationally efficient [9, 10, 17]. Lastly, these predictors consistently yield highly accurate results [9, 10, 17]. Among the established PPI predictors that incorporate various types of information beyond just the sequence, notable examples include:

**PIPE4** [21]. It is a network-based algorithm that predicts PPIs based on network topology and sequence similarity. It uses a graph-based approach to represent proteins as nodes and their interactions as edges in a network. The algorithm then applies machine learning techniques to predict new interactions based on the properties of the network.

**PEPPI** [24]. It is built using a combination of structural similarity, sequence similarity, functional association data, and machine learning-based classification through a naive Bayesian classifier model. The algorithm consists of several independent modules, including a neural network trained on conjoint triads (where seven classes of amino acids are clustered according to their dipoles and volumes of the side chains, and any three continuous amino acids are regarded as a unit, from which 343 features can be extracted) [26], a STRING database lookup module (which checks for known interacting proteins), a threading-based module [27] (a protein modeling method for proteins that have the same fold as proteins of known structures but do not have homologous), and a sequence-based module using the Basic Local Alignment Search Tool BLAST [28, 29]. Scores from each module are transformed into a ratio of likelihood based on pre-trained score probability distributions, and the final likelihood ratio is calculated as the product of likelihood ratios from each indepen-

dent module.

**PPI-Affinity [30].** It is built using a dataset of protein-protein complexes with known binding affinity (BA) data from the PDBbind database [31]. Moreover, molecular descriptors using ProtDCal were used. The model was trained using Support Vector Machines (SVM) with ensemble learning, optimizing hyperparameters using a grid search, and selecting the best ensemble model based on performance measures.

Moreover, the AlphaFold2 algorithm has demonstrated remarkable accuracy in predicting protein structures, including residues involved in protein-protein interactions [12]. Recent studies applying AlphaFold2-Multimer, have highlighted its suitability and limits for predicting interacting complexes [19, 32, 33]. Again, these complex algorithms use various types of information beyond protein sequence representation. That emphasizes the need to thoroughly evaluate the effectiveness of DL-based embeddings in predicting PPIs, thus solely relying on the protein sequences. It is important to demonstrate the extent to which these embeddings can be employed by the scientific community to achieve quick and precise predictions, ultimately reducing the reliance on extensive initial information and pre-processing steps for the predictor.

It is important to note that computational methods often have PPI training sets which include high quality positive interactions but lack of reliable negative interactions. Instead, negative samples are often generated by randomly pairing proteins that are not known to interact. However, these negative samples may not accurately represent the absence of interactions, leading to potential biases in the prediction models. The Negatome database [1] provides a curated collection of negative PPIs, which can be useful for training and evaluating PPI prediction models.

In this work, we aim to benchmark and compare the most commonly used DL-based embeddings [13–15] for predicting PPIs solely based on protein sequence information, relying on high quality training data (that includes experimentally validated negative protein-protein interactions).

The best model, obtained through double cross-validation (DCV) [34] and hyperparameter optimization, is selected and further evaluated to predict peroxisomal protein-protein interactions. The resulting tool is referred to as Peroxisomal Protein-Protein Interaction (P-PPI). Additionally, this study explores the complementary use of the P-PPI tool and AlphaFold2 [12]. By combining AlphaFold2-Multimer predictions with P-PPI, we aim to enhance the assessment of protein-protein interactions by incorporating structural insights. This integrated approach leverages the strengths of both computational methods, employing DL-based embeddings and precise protein structure predictions to enable a comprehensive and accurate analysis of peroxisomal protein-protein interactions. The integration of these approaches has significant potential to help us advance our understanding of complex protein networks and their functions.

---

## 6.2 Methods

### 6.2.1 Data set

#### Training and validation set

The PPIs are retrieved from the DIP database at <https://dip.doe-mbi.ucla.edu/dip/Main.cgi> (human dataset - 2017). The positive dataset contains 5970 encoded PPIs. The non-PPIs are retrieved from NEGATOME 2.0 at <http://mips.helmholtz-muenchen.de/proj/ppi/negatome/manual.html>. The negative dataset contains 2087 encoded non-PPIs. As a representative and more balanced sample, we retrieved positive and negative training sets of size 1500 and a validation set of size 1000.

#### Peroxisomal PPI data set

We selected non overlapping protein-protein interactions (from the protein left out from the DIP data set) where at least one of the two proteins was a peroxisomal protein. In total we obtained 77 peroxisomal interactions. 60 were used for fine-tuning the model. 17 were used as test set.

### 6.2.2 Deep Learning Protein Sequence Embeddings

We considered three recently proposed methods for the embedding of protein sequences based on deep-learning approaches:

**Unified Representation (UniRep).** UniRep [35] employs a recurrent neural network architecture with 1900 hidden units to capture chemical, biological, and evolutionary information from protein sequences. It utilizes a hidden state vector that is iteratively updated based on the previous state. This allows the model to learn by predicting the next amino acid in the sequence based on the preceding context. UniRep generates protein sequence embeddings of length 64, 256, or 1900 units, depending on the chosen neural network architecture. In this study, we utilized the 1900-unit embedding (average final hidden array). For a detailed explanation on how to obtain the UniRep embedding, please refer to the specific GitHub repository: <https://github.com/churchlab/UniRep>.

**Sequence-to-Vector (SeqVec).** The Sequence-to-Vector embedding (SeqVec) [36] adopts a natural language processing approach, treating amino acids as words and proteins as sentences, to capture biophysical information from protein sequences. SeqVec is obtained by training ELMo [37], a deep contextualized word representation that models complex characteristics of word use and their variations across contexts, using a 2-layer bidirectional LSTM [38] backbone. In this case, ELMo is pretrained on a large text corpus, specifically UniRef50 [39]. SeqVec can generate embeddings at both the per-residue (word-level) and per-protein (sentence-level) levels. The per-residue level embedding allows prediction of secondary structure or intrinsically disordered regions, while the per-protein level embedding enables prediction of subcellular localization and differentiation between membrane-bound

and water-soluble proteins [36]. In this study, we utilized the per-protein level representation, where the protein sequence is represented by a 1024-unit embedding. For detailed instructions on retrieving the SeqVec embedding, please refer to the specific GitHub repository: <https://github.com/mheinzingner/SeqVec>.

**Evolutionary Scale Modelling - 1b (ESM-1b).** ESM-1b is trained on 250 million sequences from the UniParc database [40] and utilizes a deep transformer architecture [41, 42], a powerful model for representation learning and generative modeling in natural language processing (NLP). The transformer architecture in ESM-1b provides contextual information for each amino acid (word) in the sequence (sentence) by comparing it to every other amino acid (word) in the sequence, including itself, using self-attention blocks. These blocks consist of three steps: 1) Computing dot product similarity and alignment scores; 2) Normalizing the scores and weighting the embeddings; 3) Re-weighting the original embeddings based on the scores. In ESM-1b, the transformer processes inputs through blocks that alternate self-attention with feed-forward connections. Since it is trained on proteins, the self-attention blocks capture pairwise interactions between all positions in the sequence, representing residue-residue interactions. ESM-1b is trained using masked language modeling objective, which requires the model to predict masked parts of the sequence by understanding dependencies between the masked site and the unmasked parts. The model is optimized with hyperparameters to train a 650 million-parameter (33 layers) model on the UR50/S dataset, resulting in the ESM-1b Transformer [13]. The resulting ESM-1b vector has a length of 1280 units.

### 6.2.3 Step Forward Feature Selection

As reported in our previous study where we analysed DL-based sequence embeddings, also here we adopted a Step Forward Feature Selection (SFFS) approach [9]. SFFS was used to select the best combination of features (predictors) that is, protein encodings or embeddings to be used as input for classification algorithms [43]. It is a wrapper method that evaluates subsets of variables, in our case, combinations of protein embeddings. It starts with the evaluation of each individual encoding, and selects that which results in the best performing selected algorithm model. Next, it proceeds by iteratively adding one embedding to the current best performing features and evaluating the performance of the classification. The procedure is halted when performance worsens and the best combination of embeddings is retained.

### 6.2.4 Logistic Regression (LR)

We used a penalised implementation of multivariable logistic regression [44]. Penalized multivariable logistic regression is a technique used to handle high-dimensional data with multicollinearity and overfitting issues. It incorporates a penalty term to control model complexity and select relevant predictors. One commonly used penalized approach is ridge regression, which adds a squared L2 penalty to the objective function. Another popular penalized method is lasso regression, which adds an L1 penalty to the objective function. Ridge regression shrinks coefficients towards zero, lasso regression performs variable selection, and elastic net combines both methods.

---

Penalized logistic regression improves model generalizability, mitigates overfitting, and enhances interpretability [44, 45].

### 6.2.5 Model Calibration and Validation

We employed a double cross-validation (DCV) technique [34, 46] for two main purposes: (i) optimizing the hyperparameters of the classification algorithms used, enabling model calibration, and (ii) obtaining unbiased estimates of prediction errors when applying the model to new cases within the dataset population. This approach is particularly suitable for small datasets.

The DCV strategy involves two nested cross-validation loops. In the outer loop, the data is initially divided into  $k$  folds, with one fold serving as the validation set while the remaining  $k - 1$  folds are used for model calibration. The inner loop is applied to the calibration set, further splitting it into training and test sets using a  $k$ -fold division. In our study, we utilized 5 folds for both the inner and outer loops.

The inner loop is responsible for optimizing the hyperparameters of the different classification algorithms. A (hyper)grid search is performed, evaluating the average classification score across the folds for each set of hyperparameters. The hyperparameters corresponding to the best classification score are then used to fit a classification model, and its quality is assessed on the validation set. This ensures unbiased evaluation of the model since the validation data was not used during the training of the classification model.

Additionally, we incorporated a step-forward feature selection procedure, as described in Section 6.2.3, within the calibration loop. This means that model calibration also involved selecting the best combination of protein sequence encodings and embeddings, based on their predictive ability.

Given the imbalance between the two classes of proteins, different weights were applied to each class. These class weights were considered metaparameters and were optimized within the inner calibration loop.

### 6.2.6 AlphaFold2-Multimer

The AlphaFold system [12], described in Jumper *et al.* (2021), is a protein structure prediction method that integrates various sources of information such as the amino acid sequence, multiple sequence alignments (MSA) [47, 48], and homologous structures. It utilizes a neural network called Evoformer, which incorporates a neural representation of the MSA and pairwise relations between amino acids [12].

The pairwise representation provides information about the relative positions of amino acids in the chain and is used to estimate the relative distances between them. The first row of the MSA embedding, together with the pair embedding, is used to generate the final structure prediction. The model is trained end-to-end (all of the parameters are trained jointly), with gradients propagating from the predicted structure through the entire network [12].

AlphaFold2-Multimer is an extension of the original AlphaFold system that specifically addresses the prediction of protein complexes with multiple chains [32]. One of the key challenges in modeling protein complexes is accounting for permutation symmetry [32, 49]. Since a protein sequence can appear multiple times

in a complex, the model cannot rely on predicting the chains in a predetermined order [49]. AlphaFold2-Multimer incorporates an optimization process to identify the optimal permutation of predicted homomer chains that aligns with the ground truth. This process utilizes a simple heuristic that greedily seeks a suitable ordering for the chains [33].

The approach for constructing multiple sequence alignments (MSAs) in AlphaFold2-Multimer [32] follows a similar methodology to that employed in AlphaFold [12, 47, 48]. For homomeric complexes, the MSA is replicated  $X$  times, where  $X$  denotes the number of chain repeats, and aligned in a left-to-right stacking manner. In the case of heteromeric complexes, the sequences between the MSAs of each chain are paired whenever possible, utilizing UniProt species annotation [40, 50, 51]. The ranking of candidate rows for each chain is determined based on their similarity to the respective target sequence, and pairs with the same rank are concatenated. If partial alignments exist, with sequence pairings between certain chains but not others, gaps are introduced between the paired sequences as padding. The remaining unpaired MSA sequences for each chain are stacked in a block-diagonal fashion below the set of paired sequences, with padding applied on the off-diagonal regions. During training, sampling of the MSA cluster centers is biased to ensure an equal probability of sampling each chain's unpaired MSA, regardless of the number of sequences it contains. The sampling of paired and unpaired sequences is unbiased and proportional to their relative occurrence. During testing, unbiased MSA sampling is performed to enhance performance slightly, as it introduces greater diversity across recycling iterations [32].

AlphaFold2-Multimer system employs a cropping procedure aimed at maximizing chain coverage and crop diversity, while also ensuring a balanced representation of interface and non-interface regions [32]. Due to memory and compute limitations, the number of residues that can be trained on is restricted, and thus the cropping process involves selecting contiguous blocks of residues up to a maximum length of 384 residues. The cropping procedure is specifically designed to involve multiple chains within a complex and give priority to binding interfaces between the chains, as these interfaces play a critical role in accurately modeling protein complexes [32].

The training protocol for AlphaFold2-Multimer closely adheres to that of AlphaFold. The training dataset comprises protein structures sourced from the Protein Data Bank (PDB) up to a maximum release date of 2018-04-30 [32, 52].

## AlphaFold metrics

### The Local Distance Difference Test (IDDT) [53]

It is a superposition-free score that evaluates local distance differences of atoms in a model, including validation of stereochemical plausibility [53]. IDDT reports the domain accuracy without requiring a domain segmentation of chain structures [32]. The distances are either computed between all heavy atoms (IDDT) or only the  $C\alpha$  atoms to measure the backbone accuracy (IDDT- $C\alpha$ ).

### Predicted aligned error (PAE) [32]

The position error of AlphaFold is assessed for each pair of residues, providing an estimation of the distance error between the predicted and true structures when



aligned on specific residues. The range of values for the position error is typically from 0 to 35 Angstroms. This information is commonly visualized as a heatmap, where the residue numbers are displayed along the vertical and horizontal axes, and the color at each pixel represents the position error value (PAE) for the corresponding pair of residues. When the relative position of two domains is accurately predicted, the PAE values will be low (typically less than 5 Angstroms) for residue pairs with one residue in each domain.

### 6.2.7 Metrics

The evaluation metrics used for assessing the performance of the models in binary classification (true vs. false interaction) were accuracy (ACC), F1 score [54], Matthews correlation coefficient (MCC) [55], and the area under the curve (AUC) of the receiver operating characteristic (ROC). These metrics are calculated based on the number of true positives ( $TP$ ), false positives ( $FP$ ), true negatives ( $TN$ ), and false negatives ( $FN$ ) using the following formulas:

True positive rate (TPR) is defined as

$$SEN/TPR = \frac{TP}{TP + FN} \quad (6.1)$$

False positive rate (FPR) is defined as

$$FPR = \frac{FP}{TN + FP} \quad (6.2)$$

Accuracy (ACC) is defined as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.3)$$

$F_1$  score [54] is defined as

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}, \quad (6.4)$$

where  $PPV$  is the positive predicted value (or precision)

$$PPV = \frac{TP}{TP + FP}, \quad (6.5)$$

The  $F_1$  score is the harmonic mean of recall and precision and varies between 0, if the precision or the recall is 0, and 1 indicating perfect precision and recall.

Matthews correlation coefficient (MCC) [55] is defined as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (6.6)$$

$MCC$  is the correlation coefficient between the true and predicted class: it is bound between  $-1$  (total disagreement between prediction and observation) and  $+1$  (perfect

prediction); 0 indicates no better than random prediction. The *MCC* is appropriate also in presence of class unbalance [56].

The area under the curve (AUC) of the receiver operating characteristic (ROC) curve which plots the true positive proportion or the Sensitivity against the Specificity, is defined as

$$AUC = \int_0^1 TPRd(FPR) \quad (6.7)$$

The AUC analysis enables the evaluation of the performance of a binary classifier system according to the variation of the discrimination threshold. A perfect prediction has an AUC score of 1.0 while an AUC of 0.5 indicates randomness [57].

### 6.2.8 Approach

We evaluated three different DL-based embeddings, SeqVec, UniRep, ESM-b1 for correctly predicting PPIs. To do so we performed 5-fold Double Cross Validation (DCV) with a Logistic Regression (LR) classifier. The best model was then tested against the validation set.

## 6.3 Results

### 6.3.1 P-PPI pipeline overview

The Figure 6.1 provides an overview of the pipeline developed in this study, showcasing the integration of the P-PPI model and an optional analysis using AlphaFold2-Multimer.

### 6.3.2 P-PPI model performances with different protein embeddings

The classification performance of various embeddings, including their concatenation, in distinguishing true and false interactions was evaluated using a step-forward feature selection approach. The results are presented in Table 6.1. Esm-1b demonstrated superior performance compared to SeqVec and UniRep, achieving accuracy scores of 0.8703, 0.8656, and 0.8654, respectively. The performance of the concatenated Esm-1b and SeqVec embeddings was found to be very similar to the performance of the Esm-1b single embedding (0.8703 vs 0.8716). Consequently, we utilized the single embedding representation (Esm-1b) for the final P-PPI model. Overall, the performances of all embeddings are very close.

### 6.3.3 Performances of the peroxisomal proteins fine-tuned model

We fine-tuned the P-PPI model by retraining it on a data set consisting of 60 peroxisomal protein-protein interactions, which was non-overlapping with the original training set. Subsequently, we evaluated the model's performance on a separate test

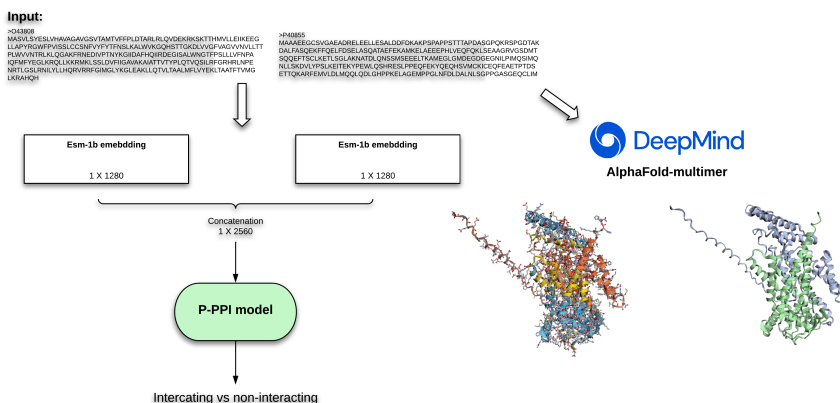


Figure 6.1: Overview of the P-PPI pipeline workflow: The protein sequences of the two putative interacting proteins (in FASTA format) enter the pipeline and are initially encoded using the ESM-1b embedding. The resulting embeddings are then concatenated to represent the protein interaction. The P-PPI model, consisting of the ESM-1b embedding and Logistic Regression, predicts whether the interaction is true or false. Additionally, as an optional step, the model can be validated against AlphaFold2-Multimer to assess the quality of complexes that involve the two interacting proteins. A robust and precise model indicates the reliability of the predicted interaction.

set comprising 17 peroxisomal interactions. The fine-tuning process resulted in significant improvements, with an accuracy of 0.9118. Detailed results can be found in Table 6.2.

### 6.3.4 AlphaFold2-Multimer model use case

As a proof of concept, we selected a positively predicted interaction from the P-PPI model and tested it against the AlphaFold2-Multimer algorithm to generate the structure of the interacting complex. Among the 17 PPIs from the peroxisomal test set, we chose a specific interaction between the Pex19 and PM34 proteins in Human (UniProt IDs: P40855-O43808). This interaction comes with 4 experimental validations (e.g. Co-IP); for details we invite to check the UniProt ID O43808 at <https://www.uniprot.org/>.

To generate the structure of the interacting complex, we inputted the FASTA sequences of the interacting proteins into the AlphaFold Google Colab script, available at <https://colab.research.google.com/github/deepmind/alphafold/blob/main/notebooks/AlphaFold.ipynb>. The results of the structure prediction are presented in Figure 6.2 and the corresponding evaluation metrics are shown in Figure 6.3.

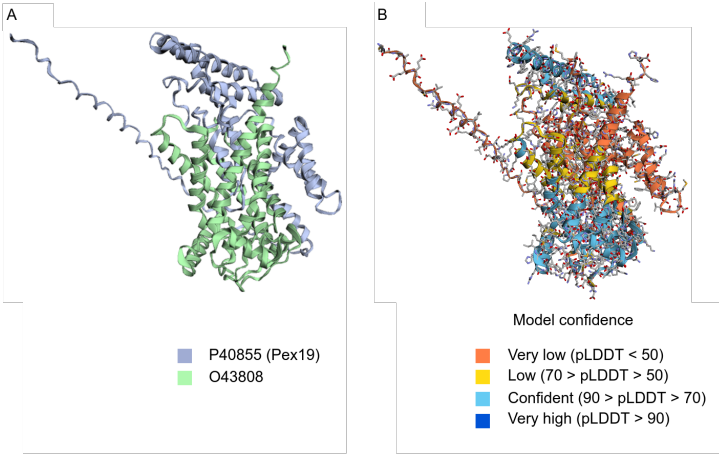


Figure 6.2: Predicted interacting complex (UniProt IDs: P40855-O43808). Subfigure A, highlights the two proteins in grey (P40855) and light green (O43808). Subfigure B shows the different portions of the complex colored by model confidence level expressed with predicted Local Distance Difference Test (pLDDT)

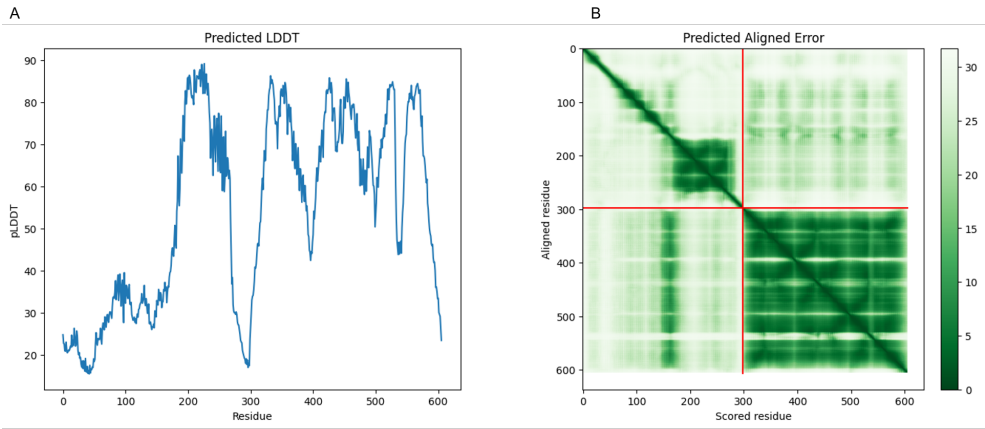


Figure 6.3: Visualization of the AlphaFold metrics for model evaluation. Subfigure A shows the performances in terms of predicted Local Distance Difference Test (pLDDT) from residue 0 to residue 606 of the modelled complex. Subfigure B shows the performances in terms of Predicted Aligned Error (PAE) as an heatmap from residue 0 to residue 606 of the complex.

Table 6.1: Results obtained using SeqVec, UniRep, and ESM-b1 embeddings are reported in terms of accuracy (ACC), Matthew’s Correlation Coefficient (MCC),  $F_1$  macro score ( $F_1$ ), and area under the curve (AUC) for both training and validation sets. The best performing embedding is highlighted in bold. The used classifier is Logistic Regression.

	ACC		MCC		$F_1$		AUC	
	CV	Val.	CV	Val.	CV	Val.	CV	Val.
SeqVec	0.8656 $\pm$ 0.0024	0.6760	0.6114 $\pm$ 0.0104	0.3573	0.7834 $\pm$ 0.0058	0.6731	0.7453 $\pm$ 0.0049	0.6760
UniRep	0.8654 $\pm$ 0.0021	0.6560	0.6100 $\pm$ 0.0120	0.3144	0.7840 $\pm$ 0.0043	0.6548	0.7467 $\pm$ 0.0017	0.6560
<b>ESM-1b</b>	0.8703 $\pm$ 0.0144	0.6840	0.6277 $\pm$ 0.0096	0.3713	0.7883 $\pm$ 0.0033	0.6827	0.7475 $\pm$ 0.0102	0.6840
UniRep-SeqVec	0.8579 $\pm$ 0.0022	0.6630	0.5906 $\pm$ 0.0133	0.3400	0.7591 $\pm$ 0.0011	0.6631	0.7179 $\pm$ 0.0032	0.6630
<b>ESM-1b-SeqVec</b>	0.8716 $\pm$ 0.0073	0.6860	0.6319 $\pm$ 0.0059	0.3773	0.7910 $\pm$ 0.0061	0.6913	0.7502 $\pm$ 0.0019	0.6860
ESM-1b-UniRep	0.8556 $\pm$ 0.0080	0.6560	0.5819 $\pm$ 0.0049	0.3115	0.7543 $\pm$ 0.0097	0.6522	0.7143 $\pm$ 0.0029	0.6548

Table 6.2: Performances of the Logistic Regression model with the ESM-1b embedding against the peroxisomal test set (17 interactions) after fine-tuning. Results are reported in terms of accuracy (ACC), Matthew’s Correlation Coefficient (MCC)  $F_1$  macro score ( $F_1$ ) and area under the curve (AUC)

Metric	Value
ACC	0.9118
$F_1$	0.9189
MCC	0.8367
AUC	0.9250

## 6.4 Discussion and Conclusion

The results obtained from our PPI prediction method, as demonstrated in Tables 6.1 and 6.2, show promising outcomes and highlight the potential of employing pre-trained deep learning models for protein sequence embedding in the prediction of protein-protein interactions (PPIs). These findings hold true not only for PPIs in general but also for peroxisomal PPIs specifically, thus we implemented the P-PPI model, which accurately predicts peroxisomal protein-protein interactions with an accuracy of 91%.

This opens up possibilities for further analysis and improvements in the field. Our approach involved using the Esm-1b embedding, which proved to be the most effective. We incorporated Esm-1b in the final model, along with the fine-tuned version, resulting in accurate prediction of peroxisomal PPI.

An additional computational validation was performed using AlphaFold2, which provided additional confidence in our predictions as presented in Figures 6.2 and 6.3.

It is important to emphasise that our training data set relies on experimentally validated positive and negative interactions. PPI predictors are often built using negative interactions, where the non-interacting proteins are selected from different subcellular locations. These interactions do not occur in a living cell but can be verified in vitro, thus adding noise to the predictive models. The models can inadvertently learn different subcellular locations instead of true and false interactions. Moreover, due to the nature of peroxisomes and their high interconnection with several organelles, we believe it is more relevant to focus on the interactions rather than the subcellular locations.

Moving forward, there are several avenues for future work. Firstly, we plan to test additional deep learning embeddings, such as the novel ESM2, to explore their performance in PPI prediction. Secondly, we aim to create a negative set of non-PPIs based on proteins from different subcellular locations, allowing for a more comprehensive comparison and providing experimental validation to our theory of preferring experimentally validated negative interactions over non-PPIs based on proteins from different subcellular locations. It is also important to consider comparing the Negatome data set with other negative datasets to understand the limitations of alternative data sources in capturing negative interactions.

Furthermore, we intend to compare our predictor against state-of-the-art methods and train the model on PPIs from other organisms. Lastly, we will suggest candidate PPIs and validate the predictions experimentally.

## Acknowledgement

This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Blohm, P. et al. “Negatome 2.0: a database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis”. In: *Nucleic Acids Research* 42.D1 (2013), D396–D400. DOI: 10.1093/nar/gkt1079. URL: <https://doi.org/10.1093/nar/gkt1079>.
- [2] Mahankali, M., Alter, G., and Gomez-Cambronero, J. “Mechanism of enzymatic reaction and protein-protein interactions of PLD from a 3D structural model”. In: *Cellular Signalling* 27.1 (2015), 69–81. DOI: 10.1016/j.cellsig.2014.09.008. URL: <https://doi.org/10.1016/j.cellsig.2014.09.008>.
- [3] Sogaard-Andersen, L. and Valentin-Hansen, P. “Protein-protein interactions in gene regulation: The cAMP-CRP complex sets the specificity of a second DNA-binding protein, the CytR repressor”. In: *Cell* 75.3 (1993), 557–566. DOI: 10.1016/0092-8674(93)90389-8. URL: [https://doi.org/10.1016/0092-8674\(93\)90389-8](https://doi.org/10.1016/0092-8674(93)90389-8).
- [4] Young, K. H. “Yeast Two-hybrid: So Many Interactions, (in) So Little Time...” In: *Biology of Reproduction* 58.2 (1998), 302–311. DOI: 10.1095/biolreprod58.2.302. URL: <https://doi.org/10.1095/biolreprod58.2.302>.
- [5] Lin, J.-S. and Lai, E.-M. “Protein–Protein Interactions: Co - Immunoprecipitation”. In: *Methods in Molecular Biology*. Springer New York, 2017, 211–219. DOI: 10.1007/978-1-4939-7033-9\_17. URL: [https://doi.org/10.1007/978-1-4939-7033-9\\_17](https://doi.org/10.1007/978-1-4939-7033-9_17).
- [6] Dunham, W. H., Mullin, M., and Gingras, A.-C. “Affinity-purification coupled to mass spectrometry: Basic principles and strategies”. In: *PROTEOMICS* 12.10 (2012), 1576–1590. DOI: 10.1002/pmic.201100523. URL: <https://doi.org/10.1002/pmic.201100523>.
- [7] Sekar, R. B. and Periasamy, A. “Fluorescence resonance energy transfer (FRET) microscopy imaging of live cell protein localizations”. In: *Journal of Cell Biology* 160.5 (2003), 629–633. DOI: 10.1083/jcb.200210140. URL: <https://doi.org/10.1083/jcb.200210140>.
- [8] Thumhuri, V. et al. “DeepLoc 2.0: multi-label subcellular localization prediction using protein language models”. In: *Nucleic Acids Research* (2022).
- [9] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. “In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins”. In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
- [10] Anteghini, M. et al. “OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection”. In: *Computational and Structural Biotechnology Journal* 21 (2022), 128–133. DOI: 10.1016/j.csbj.2022.11.058. URL: <https://doi.org/10.1016/j.csbj.2022.11.058>.
- [11] Savojardo, C. et al. “DeepMito: accurate prediction of protein sub - mitochondrial localization using convolutional neural networks”. In: *Bioinformatics* 36.1 (2019), 56–64.

- [12] Jumper, J. et al. “Highly accurate protein structure prediction with AlphaFold”. In: *Nature* 596.7873 (2021), 583–589. doi: 10.1038/s41586-021-03819-2. URL: <https://doi.org/10.1038/s41586-021-03819-2>.
- [13] Rives, A. et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021).
- [14] Alley, E. C. et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16.12 (2019), 1315–1322. doi: 10.1038/s41592-019-0598-1. URL: <https://doi.org/10.1038/s41592-019-0598-1>.
- [15] Heinzinger, M. et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20 (2019).
- [16] Elnaggar, A. et al. “ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 01 (2021), 1–1. doi: 10.1109/TPAMI.2021.3095381.
- [17] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. “PortPred: exploiting deep learning embeddings of amino acid sequences for the identification of transporter proteins and their substrates”. In: (2023). doi: 10.1101/2023.01.26.525714. URL: <https://doi.org/10.1101/2023.01.26.525714>.
- [18] Nambiar, A. et al. “Transforming the Language of Life: Transformer Neural Networks for Protein Prediction Tasks”. In: *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. BCB ’20. Virtual Event, USA: Association for Computing Machinery, 2020. ISBN: 9781450379649. doi: 10.1145/3388440.3412467. URL: <https://doi.org/10.1145/3388440.3412467>.
- [19] Bryant, P., Pozzati, G., and Elofsson, A. “Improved prediction of protein-protein interactions using AlphaFold2”. In: *Nature Communications* 13.1 (2022). doi: 10.1038/s41467-022-28865-w. URL: <https://doi.org/10.1038/s41467-022-28865-w>.
- [20] Singh, R. et al. “Struct2Net: a web service to predict protein-protein interactions using a structure-based approach”. In: *Nucleic Acids Research* 38.Web Server (2010), W508–W515. doi: 10.1093/nar/gkq481. URL: <https://doi.org/10.1093/nar/gkq481>.
- [21] Dick, K. et al. “PIPE4: Fast PPI Predictor for Comprehensive Inter- and Cross-Species Interactomes”. In: *Scientific Reports* 10.1 (2020). doi: 10.1038/s41598-019-56895-w. URL: <https://doi.org/10.1038/s41598-019-56895-w>.
- [22] Chen, K.-H., Wang, T.-F., and Hu, Y.-J. “Protein-protein interaction prediction using a hybrid feature representation and a stacked generalization scheme”. In: *BMC Bioinformatics* 20.1 (2019). doi: 10.1186/s12859-019-2907-1. URL: <https://doi.org/10.1186/s12859-019-2907-1>.
- [23] Li, Y. and Ilie, L. “SPRINT: ultrafast protein-protein interaction prediction of the entire human interactome”. In: *BMC Bioinformatics* 18.1 (2017). doi: 10.1186/s12859-017-1871-x. URL: <https://doi.org/10.1186/s12859-017-1871-x>.



- 
- [24] Bell, E. W. et al. “PEPPI: Whole-proteome Protein-protein Interaction Prediction through Structure and Sequence Similarity, Functional Association, and Machine Learning”. In: *Journal of Molecular Biology* 434.11 (2022), 167530. DOI: 10.1016/j.jmb.2022.167530. URL: <https://doi.org/10.1016/j.jmb.2022.167530>.
  - [25] Canzler, S. et al. “ProteinPrompt: a webserver for predicting protein-protein interactions”. In: *Bioinformatics Advances* 2.1 (2022). Ed. by M. Gromiha. DOI: 10.1093/bioadv/vbac059. URL: <https://doi.org/10.1093/bioadv/vbac059>.
  - [26] Wang, X. et al. “A novel conjoint triad auto covariance (CTAC) coding method for predicting protein-protein interaction based on amino acid sequence”. In: *Mathematical Biosciences* 313 (2019), 41–47. DOI: 10.1016/j.mbs.2019.04.002. URL: <https://doi.org/10.1016/j.mbs.2019.04.002>.
  - [27] Rost, B., Schneider, R., and Sander, C. “Protein fold recognition by prediction-based threading”. In: *Journal of Molecular Biology* 270.3 (1997), 471–480. DOI: 10.1006/jmbi.1997.1101. URL: <https://doi.org/10.1006/jmbi.1997.1101>.
  - [28] Altschul, S. F. et al. “Basic local alignment search tool”. In: *Journal of Molecular Biology* 215.3 (1990), 403–410. DOI: 10.1016/s0022-2836(05)80360-2. URL: [https://doi.org/10.1016/s0022-2836\(05\)80360-2](https://doi.org/10.1016/s0022-2836(05)80360-2).
  - [29] Camacho, C. et al. “BLAST+: architecture and applications”. In: *BMC Bioinformatics* 10.1 (2009).
  - [30] Romero-Molina, S. et al. “PPI-Affinity: A Web Tool for the Prediction and Optimization of Protein–Peptide and Protein–Protein Binding Affinity”. In: *Journal of Proteome Research* 21.8 (2022), 1829–1841. DOI: 10.1021/acs.jproteome.2c00020. URL: <https://doi.org/10.1021/acs.jproteome.2c00020>.
  - [31] Wang, R. et al. “The PDBbind Database: Methodologies and Updates”. In: *Journal of Medicinal Chemistry* 48.12 (2005), 4111–4119. DOI: 10.1021/jm048957q. URL: <https://doi.org/10.1021/jm048957q>.
  - [32] Evans, R. et al. “Protein complex prediction with AlphaFold-Multimer”. In: (2021). DOI: 10.1101/2021.10.04.463034. URL: <https://doi.org/10.1101/2021.10.04.463034>.
  - [33] Bryant, P. et al. “Predicting the structure of large protein complexes using AlphaFold and Monte Carlo tree search”. In: *Nature Communications* 13.1 (2022). DOI: 10.1038/s41467-022-33729-4. URL: <https://doi.org/10.1038/s41467-022-33729-4>.
  - [34] Cawley, G. C. and Talbot, N. L. C. “On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation”. In: *Journal of Machine Learning Research* 11.70 (2010), 2079–2107. URL: <http://jmlr.org/papers/v11/cawley10a.html>.
  - [35] Alley, E. et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16 (2019). DOI: 10.1038/s41592-019-0598-1.
  - [36] Heinzinger, M. et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20 (2019).

- [37] Peters, M. E. et al. *Deep contextualized word representations*. 2018.
- [38] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), 1735–1780. doi: 10.1162/neco.1997.9.8.1735.
- [39] Suzek, B. E. et al. “UniRef: comprehensive and non-redundant UniProt reference clusters”. In: *Bioinformatics* 23.10 (2007), 1282–1288.
- [40] Consortium, T. U. “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
- [41] Vaswani, A. et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, 5998–6008.
- [42] Harris, Z. S. “Distributional structure”. In: *Word* 10.2-3 (1954), 146–162.
- [43] Meyer-Baese, A. and Schmid, V. “Chapter 2-feature selection and extraction”. In: 2014, 21–69.
- [44] Cramer, J. “The Origins of Logistic Regression”. In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (2002). doi: 10.2139/ssrn.360300.
- [45] Yan, Y. et al. “Comparison of standard and penalized logistic regression in risk model development”. In: *JTCVS Open* 9 (2022), 303–316. doi: 10.1016/j.xjon.2022.01.016. URL: <https://doi.org/10.1016/j.xjon.2022.01.016>.
- [46] Filzmoser, P., Liebmann, B., and Varmuza, K. “Repeated double cross validation”. In: *Journal of Chemometrics: A Journal of the Chemometrics Society* 23.4 (2009), 160–171.
- [47] Chowdhury, B. and Garai, G. “A review on multiple sequence alignment from the perspective of genetic algorithm”. In: *Genomics* 109.5-6 (2017), 419–431. doi: 10.1016/j.ygeno.2017.06.007. URL: <https://doi.org/10.1016/j.ygeno.2017.06.007>.
- [48] Baltzis, A. et al. “Highly significant improvement of protein sequence alignments with AlphaFold2”. In: *Bioinformatics* 38.22 (2022). Ed. by P. L. Martelli, 5007–5011. doi: 10.1093/bioinformatics/btac625. URL: <https://doi.org/10.1093/bioinformatics/btac625>.
- [49] Bliven, S. E. et al. “Analyzing the symmetrical arrangement of structural repeats in proteins with CE-Symm”. In: *PLOS Computational Biology* 15.4 (2019). Ed. by A. E. Darling, e1006842. doi: 10.1371/journal.pcbi.1006842. URL: <https://doi.org/10.1371/journal.pcbi.1006842>.
- [50] Alex Bateman, and et al. “UniProt: the Universal Protein Knowledgebase in 2023”. In: *Nucleic Acids Research* 51.D1 (2022), D523–D531. doi: 10.1093/nar/gkac1052. URL: <https://doi.org/10.1093/nar/gkac1052>.
- [51] Suzek, B. E. et al. “UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches”. In: *Bioinformatics* 31.6 (2014), 926–932.
- [52] Berman, H. M. et al. “The Protein Data Bank”. In: *Nucleic Acids Research* 28.1 (2000), 235–242.

- 
- [53] Mariani, V. et al. "IDDT: a local superposition-free score for comparing protein structures and models using distance difference tests". In: *Bioinformatics* 29.21 (2013), 2722–2728. DOI: 10.1093/bioinformatics/btt473. URL: <https://doi.org/10.1093/bioinformatics/btt473>.
- [54] Rijsbergen, C. J. V. *Information Retrieval*. 2nd. Butterworth-Heinemann, 1979.
- [55] Matthews, B. W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), 442–451.
- [56] Boughorbel, S., Jarray, F., and El-Anbari, M. "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric". In: *PloS one* 12.6 (2017), e0177678.
- [57] Melo, F. "Area under the ROC Curve". In: *Encyclopedia of Systems Biology*. Springer New York, 2013, 38–39. DOI: 10.1007/978-1-4419-9863-7\_209. URL: [https://doi.org/10.1007/978-1-4419-9863-7\\_209](https://doi.org/10.1007/978-1-4419-9863-7_209).



---

# PortPred: exploiting deep learning embeddings of protein sequences to accurately identify transporter proteins and their substrate category

This chapter is based on:

Marco Anteghini, Vitor A.P. Martins dos Santos, Edoardo Saccenti. PortPred: exploiting deep learning embeddings of protein sequences to accurately identify transporter proteins and their substrate category.

*Preprint: bioRxiv - Manuscript Submitted*

DOI: 10.1101/2023.01.26.525714

## Abstract

The physiology of every living cell is regulated at some level by transporter proteins which constitute a relevant portion of membrane-bound proteins and are involved in the movement of ions, small and macromolecules across bio-membranes. The importance of transporter proteins is unquestionable. The prediction and study of previously unknown transporters can lead to the discovery of new biological pathways, drugs and treatments. Here we present PortPred, a tool to accurately identify transporter proteins and their substrate starting from the protein amino acid sequence. PortPred successfully combines pre-trained deep learning-based protein embeddings and machine learning classification approaches and outperforms other state-of-the-art methods. In addition, we present a comparison of the most promising protein sequence embeddings (Unirep, SeqVec, ProteinBERT, ESM-1b) and their performances for this specific task.

Transporter proteins are considered essential for the functioning of all living organisms; malfunctioning of transporters is often associated with diseases and they are frequently studied as drug targets [1–3].

A transporter or membrane transport protein is a protein involved in the transport of ions, small molecules and macro-molecules across a biological membrane [4, 5]. Transporter proteins are continuously identified and characterized. Nowadays, they are represented and classified in the transporter classification system (<http://www.tcdb.org/>) that systematically classifies transport proteins according to their mode of transport, energy coupling mechanism, molecular phylogeny, and substrate specificity [6, 7].

The biological relevance of transporter proteins is reflected by the expanding body of research literature on the topic: between 2018 and 2022 18.295 papers appear on PUBMED (<https://pubmed.ncbi.nlm.nih.gov/>) containing the words ‘transporter protein’ or ‘transporters’ in their title or abstract, while 16.365 were found as published in the previous four years. The protein structure database PDB (<https://www.rcsb.org/>) [8], contains 5150 structures with resolution  $\leq 1.5 \text{ \AA}$  (14-12-2022), corresponding to 3424 proteins identified as transporters. UniProt reports 7391 reviewed sequences annotated as a transporter (14-12-2022); if the search is related to sequences automatically annotated, the number of transporters increases to a staggering 373.477. The huge amount of sequence data compared to structural data concerning transporter proteins indicates the necessity to rely on efficient sequence-based predictors to accurately identify transporter proteins.

In recent years, have been presented several tools that predict transporter proteins starting from the amino acid sequence based on machine learning approaches. At the time of this writing, the following tools were published: 1) Transporter Substrate Specificity Prediction (TrSSP) [9]; 2) SCMMTP [10]; 3) Li et al. approach [11]; 4) FastTrans [12]; 5) TooT-T [13]; 6) TranCEP [14].

All these tools exploit the amino acid composition of the protein sequences. However, the application of deep learning (DL) approaches to encode protein amino acid sequences has shown promising results for several tasks such as subcellular and sub-organelle classification, protein structure and function prediction, and protein-protein interactions (PPI) [15–21].

Following our previous works on the use of sequence embeddings for the pre-

diction of the sub-cellular localization of peroxisomal proteins, [17, 21] we apply a similar framework to the development of PortPred, a prediction tool for the accurate identification of transporter proteins and multi-class classification of their transported substrates.

We reviewed and compared the most recent and frequently used DL-based protein embeddings (namely: UniRep [15], SeqVec [16], PROTBERT [20], and ESM-1b [18]) in predicting transporter proteins and their relative substrates, in combination with several machine learning approaches.

PortPred was developed by testing both the single embeddings and their combination thereof and finding the best protein representation and machine learning classifier. PortPred was also tested against the state-of-the-art transporters predictors ([9–14]) for either binary classification (transporter vs not-transporter) and multi-class classification related to the transporter substrates namely: cation, anion, electron, lipid, aminoacid, protein/mRNA, sugar, others. Details about the substrate are shown in Table 7.1.

We found PortPred to outperform the state-of-the-art in predicting transporter proteins and being accurate in predicting different substrates.

Table 7.1: Descriptions of substrate-specific transporters for all the classes considered in this study.

Substrate	Transporter Description
Amino Acid	Transporters for amino acid molecules that are mainly of the solute carrier family. Examples are transporters from the Amino Acid-Polyamine-Organocation (APC) Superfamily [22].
Anion	The organic anion transporter (OAT) subfamily constitutes roughly half of the SLC22 (solute carrier 22) transporter family.[23]
Cation/Hydrogen Ion	Protein involved in the transport of hydrogen ions across a membrane. Used to power processes such as ATP synthesis and bacterial flagellar rotation.[24]
Electron	Electron transporter proteins form chains (ETCs) where each ETC is a series of protein complexes and other molecules that transfer electrons from electron donors to electron acceptors via redox reactions [25].
Protein/mRNA	Membrane proteins involved in the movement of macromolecules, such as another protein or mRNA.[26, 27]
Sugar	The sugar transporters are responsible for the binding and transport of carbohydrates, organic alcohols, and acids in a wide range of organisms[28].
Lipid	ATP-dependent ABC and P4-ATPase lipid transporters are known to contribute to lipid translocation across the lipid bilayers on the cellular membranes[29].
Other	This category includes all transporters that are not represented in the other classes. For example transporter proteins that move metal ions like iron, nickel, copper, and zinc [30].

## 7.1 Methods

### 7.1.1 Database search queries

We obtained the number of transporter proteins available in biological databases, mentioned in the Introduction (Section 1), with the following queries.

**PubMed query 1:** ((transporter protein[Title/Abstract]) OR (transporters[Title/Abstract])) AND ((“2014”[Date - Publication] : “2018”[Date - Publication])).

**PubMed query 2:** ((transporter protein[Title/Abstract]) OR (transporters[Title/Abstract])) AND ((“2018”[Date - Publication] : “3000”[Date - Publication])).

**PDB query:** (Structure Keywords HAS ANY OF WORDS "transport, transporter, transporters, transporter protein") AND  
(Refinement Resolution = [ 0 - 1.5 ]).

**UniProt query:** (cc\_function:transporter)

### 7.1.2 Overview of existing methods for the prediction of transporter proteins and their substrate

**Transporter Substrate Specificity Prediction (TrSSP) [9].** It implements Support Vector Machines classifier on six prediction modules considering the following features respectively: 1) amino acid composition [31]; 2) AAIndex that considers the biochemical composition of the amino acid residues [32, 33]. In particular, a subset of the AAIndex database which has 49 selected physical, chemical, energetic, and conformational properties [33]; 3) The position-specific scoring matrix profile (PSSM) [34, 35] run on the Swissprot data set [24]. PSSM captures the conservation pattern in the alignment and summarises evolutionary information of the protein where the scoring matrix is at the basis of protein BLAST searches (BLAST and PSI-BLAST) [36]; 4) combination of AAIndex/PSSM with the Swissprot based PSSM; 5) PSSM run on UniRef90 [37]; 6) a combination of AAIndex/PSSM (UniRef90).

**Scoring Card Method for Membrane Transport Proteins (SCMMTP) [10].** It implements a scoring card method (SCM) based on the dipeptide composition of the amino acid sequence to identify putative membrane transport proteins. In SCMMTP, the first step is the creation of a matrix of (20x20) 400 dipeptides which represents the normalized dipeptide propensity scores of the Membrane Transport Proteins (MTPs). This matrix is then optimized using the Improved Genetic Algorithm (IGA) for maximum satisfiability (MAX-SAT) problems, [38]. IGA optimizes the dipeptide propensity scores maximizing the prediction accuracy and conserving the original sequence information. The fitness function of IGA is concerned with the area under the ROC curve (AUC) [39] and Pearson's correlation coefficient between the initial and optimized propensity scores of 20 amino acids.

**Li et al. [11].** This approach first creates a hybrid feature representation of the amino acid sequence which integrates the Position Specific Scoring Matrix (PSSM) [34], the amino acid composition, biochemical properties from the PROFEAT (Protein Features) [40], and Gene Ontology (GO) terms. The hybrid feature is created by recursively selecting features using an SVM-based backward feature extraction model which is used to predict the substrate class of transmembrane transport proteins.

**FastTrans [12].** It generates a word-embedding representation of the protein sequence implementing a natural language processing approach. First, biological words are generated by splitting the amino acid sequence into overlapping fragments of the same length. Secondly, a word embedding vector for each biological word is generated using Skip gram [41] or Continuous Bags of Words (CBWO) models [42]. The classification is performed using SVM [43].



---

**TooT-T [13].** It is an ensemble classifier that combines the predictions from homology annotation transfer and machine-learning classifiers. The ensemble classifier uses six predictions (three from the homology annotation transfer and three from SVM classifiers) and outputs the final binary prediction (transporter vs non-transporter). It is implemented using the Gradient Boosting Machine (GBM), as available by caret package in R <https://CRAN.R-project.org/package=caret>. Given a query protein, this method starts with a homology search of the Transporter Classification Database (TCDB) [7] using BLAST [44]. The query sequence is classified as a transporter if a hit is found using three predetermined sets of thresholds, thus generating the three homology modelling annotation transfer predictions. Secondly, three variations of newly generated features called psi-composition feature psiAAC, psiPAAC, and psiPseAAC are computed [13]. Psi-composition combines amino acid composition with alignment results from PSI-BLAST [36]. These psi-composition features are then used as input for three SVM classification models.

**TranCEP [14].** It uses the pair amino acid composition (PAAC) encoding scheme, the TM-Coffee algorithm for generating multiple sequence alignments [45], and its relative transitive consistency score (TCS) [46]. The predictor relies on eight SVM classifiers, one for distinguishing between each pair of classes of substrates.

## Software

We report the links to the tools mentioned and tested in this study (if available).

- TrSSP - <https://www.zhaolab.org/TrSSP/>
- SCMMTP - [http://iclab.life.nctu.edu.tw/iclab\\_webtools/SCMMTP/](http://iclab.life.nctu.edu.tw/iclab_webtools/SCMMTP/)
- FastTrans - <http://bio216.bioinfo.yzu.edu.tw/fasttrans/>
- TranCEP - <https://github.com/bioinformatics-group/TranCEP>

## 7.1.3 Deep Learning Based Protein Sequence Embeddings

We considered four recently proposed methods for the embedding of protein sequences based on deep-learning approaches and protein sequences:

**The Unified Representation (UniRep) [15]** is based on a 1900-hidden unite recurrent neural network architecture, able to capture evolutionary, chemical and biological information encoded in the protein sequence starting from 24 million UniRef50 sequences [37] where UniRef50 is a non-redundant sub-cluster of Uniprot [24]. In UniRep, the protein sequence is modelled by using a hidden state vector, which is recursively updated based on the previously hidden state vector. That means the method learns by scanning a sequence of amino acids, predicting the next one based on the sequence it has seen before. Using UniRep, a protein sequence can be represented by an embedding with a length of 64, 256, or 1900 units, depending on the neural network architecture. In this study, we used the 1900 units length (average final hidden array). For a detailed explanation of how to retrieve the UniRep embedding, we refer the reader to the specific GitHub repository

<https://github.com/churchlab/UniRep>(11.2021) or the bio-embeddings GitHub repository [https://github.com/sacdallago/bio\\_embeddings](https://github.com/sacdallago/bio_embeddings).

**The Sequence-to-Vector embedding (SeqVec) [16]** is based on a natural language processing (NLP) approach. It embeds biophysical information of a protein sequence where amino acids are words and proteins are sentences. SeqVec is obtained by training ELMo [47], on UniRef50 [37]. ELMo is a deep contextualised word representation that models both complex characteristics of word use (e.g., syntax and semantics) and how these vary across linguistic contexts. It consists of a 2-layer bidirectional LSTM [48] backbone pre-trained on a large text corpus. The SeqVec embedding can be obtained based on either a per-residue level (word level) or a per-protein level (sentence level). The per-residue level protein sequence embedding is informative in predicting the secondary structure or intrinsically disordered region; The per-protein level is useful to predict subcellular localisation and to distinguish membrane-bound vs water-soluble proteins [16]. Here we use the per-protein level representation, where the protein sequence is represented by an embedding of length 1024. For a detailed explanation of how to retrieve the SeqVec embedding, we refer the reader to the specific GitHub repository <https://github.com/mheinzingner/SeqVec> or the bio-embeddings repository [https://github.com/sacdallago/bio\\_embeddings](https://github.com/sacdallago/bio_embeddings).

**ProteinBert (PROTBERT) [49]** is inspired by the Bidirectional Encoder Representations from Transformers (BERT) which is a deep learning model that utilizes a transformer architecture to pretrain on large amounts of unlabeled text data, enabling it to generate high-quality contextualized word representations for various NLP tasks [50]. PROTBERT was instead pretrained on the raw protein sequences available in Uniref100 (~106 million proteins) [37, 50]. The original BERT model is trained on two tasks: 1) language modelling where 15% of tokens are masked and the model predicts the masked tokens from context; 2) next sentence prediction where BERT is trained to predict the probability of a chosen next sentence given the first sentence. BERT learns contextual embeddings for words and can be finetuned on small data sets for optimized predictions on specific tasks [50]. In PROTBERT sequences are treated as separate documents, where the 'next' sentence prediction is not used. The masking procedure works by training randomly masked protein sequences, similar to the original BERT model. In particular, the model takes a sequence (sentence) as input, masks 15% of the amino acids (words) from it and is asked to output the complete sequence. ProteinBert was pretrained on two simultaneous tasks. 1) bidirectional language modelling of protein sequences 2) Gene Ontology (GO) annotation prediction, which captures diverse protein functions [51]. The final embedding has a length of 1024. For a detailed explanation of how to retrieve the ProteinBert embedding, we refer the reader to the specific GitHub [https://github.com/nadavbra/protein\\_bert](https://github.com/nadavbra/protein_bert) or the bio-embeddings repository [https://github.com/sacdallago/bio\\_embeddings](https://github.com/sacdallago/bio_embeddings).

**The Evolutionary Scale Modelling - 1b (ESM-1b)** was trained on 250 million sequences of the UniParc database [24] and relies on a deep transformer architecture [52, 53], a powerful model architecture for representation learning and generative

---

modelling in NLP. The peculiarity of the transformer architecture is that it is able to return for each amino acid (word) of the sequence (sentence), an embedding with contextual information. In other terms, it compares every amino acid (word) in the sequence (sentence) to every other amino acid (word) in the sequence (sentence), including itself, and reweighs the embeddings of each word. The modules responsible for this process are called self-attention blocks and consist of three main steps: 1) Dot product similarity and alignment scores; 2) Scores normalization and embedding weight; 3) Reweighing of the original embeddings. In ESM-1b, the transformer processes inputs through a series of blocks that alternate self-attention with feed-forward connections. In this case, since it has been trained on proteins, the self-attention blocks construct pairwise interactions between all positions in the sequence, so that the transformer architecture represents residue–residue interactions. In addition, ESM-1b was trained using the masked language modelling objective [53] which forces the model to identify dependencies between the masked site and the unmasked parts of the sequence in order to make the prediction of the masked parts. Finally, the model was optimized scaling the identified hyperparameters to train a model with ~650 M parameters (33 layers) on the UR50/S data set, resulting in the ESM-1b Transformer [18]. The final length of the ESM-1b vector is 1280.

### 7.1.4 Overview of PortPred development and benchmarking

The overall strategy for the development of the PortPred tool for the prediction of transporter proteins and their substrates is schematized in Figure 7.1. It consists of 4 main steps: 1) Curation of protein sequence data; 2) Generation of the embeddings (ESM-b1, UniRep, SeqVec, PROTBERT) of the amino acid sequence; 3) Evaluation of different ML approaches; 4) Benchmarking with available tools.

### 7.1.5 Data sets

Our ML architecture was trained on three different training sets. Training set 1 is a newly generated data set (the PortPred data set); Training set 2 is the TrSSP training set and Training set 3 is the FastTrans training set [9, 12]. It was then tested against three different validation sets. Validation set 1 is an independent data set containing Peroxisomal proteins, Validation set 2 is an independent data set from the TrSSP predictor [9], and Validation set 3 is an independent data set from the FastTrans predictor [12].

Training set 1, Training set 2 as well as Validation set 1 and Validation set 2 were used as benchmarks. See Sections 7.1.5, 7.1.5 and 7.1.5 for details. The newly generated data set, that contains peroxisomal proteins, was used as a specific real-world use case (see section 7.1.5). A complete summary of the used data sets is available in Table 7.2.

#### Training set 1, the PortPred data set

Given the level of redundancy present in the data set available in the literature (see 7.1.5), and to have an unbiased comparison of the tools, we defined a novel data set that we consider more reliable for the final model training. The proteins were retrieved from Uniprot (02-10-2021) [24] obtaining 6631 transporter protein

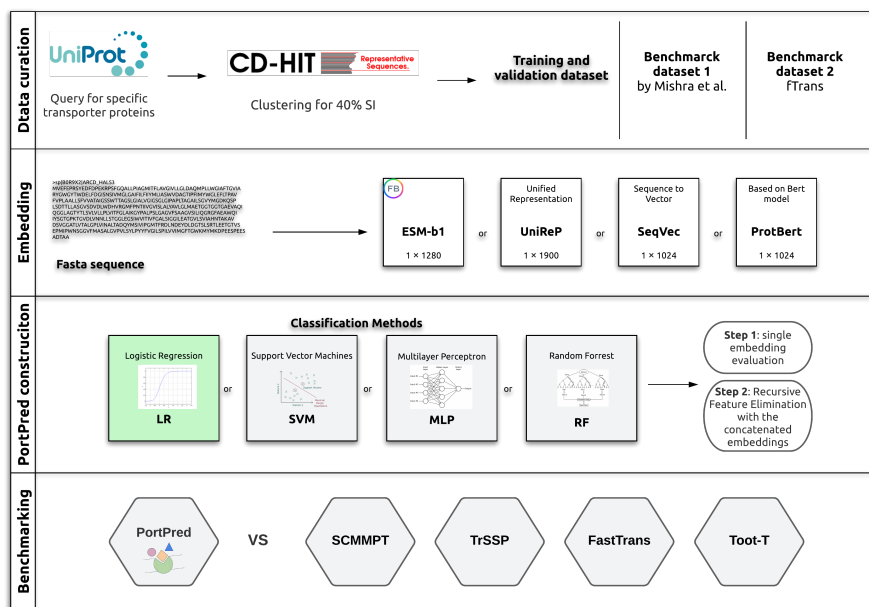


Figure 7.1: Overview of the PortPred development. Data curation: retrieval and selection of protein sequences (see Section 7.1.5). Embedding: conversion of protein sequences to standard encodings, namely: ESM-b1, unified representation (UniRep), sequence-to-vector (SeqVec), and PROTBERT. PortPred construction: application of different classification algorithms (Section 7.1.6), evaluation and selection of the best single embedding (step 1), evaluation and selection of the best combination of sequence embeddings using recursive feature elimination (RFE) (see Section 7.1.7) (step 2). Benchmarking: comparison of PortPred tool and or data set with the transporter classifiers available in literature: Scoring Card Method for Membrane Transport Proteins (SCMMPT); Transporter Substrate Specificity Prediction (TrSSP); FastTrans; TooT-T

sequences and 19139 non-transporter sequences. The data set was clustered using cd-hit [54] for 40% of sequence identity and filtered to not overlap with the Training set 2 7.1.5. We obtained a data set containing 1781 transporter proteins divided into 8 classes, namely: cation transporters, electron transporters, lipid transporters, aminoacid transporters, protein/mRNA transporters, sugar transporters, others transporters which include calcium, cobalt, copper, porin, iron, potassium, sodium, zinc, nickel, neurotransmitter, oxygen, phosphate, sulfate, ammonia and er-Golgi located transporters. We also retrieved 1781 non-transporter proteins as negatives among our random sample of non-transporter proteins. Given some limitations in the generation of the embeddings for very long protein sequences (e.g. ESM-b1 does not embed proteins longer than 1024 residues), we removed them from the data set, which finally consists of a balanced and non-redundant data set of 1580 positives entries and 1621 negatives entries. The data set is available at <https://github.com/MarcoAnteghini>.

Table 7.2: The six data sets used in this study. In bold are Training set 1 and Validation set 1 which are newly generated data sets from this work. Validation set 1 is an independent data set which contains peroxisomal proteins. Note that the Validation set 1 does not contain information about the specific transporter substrates and it is used as a real-world case scenario. Training set 2 is the training set used in the Transporter Substrate Specificity Prediction (TrSSP) paper. The Validation set 2 is an independent data set also from the TrSSP paper. Training set 3 is the training set used in the FastTrans paper. The Validation set 3 is an independent data set also from the FastTrans paper.

	amino acid	anion	cation/hydrogen ion	electron	protein/mRNA	sugar	lipid	other	positives	negatives	total
Training set 2	70	60	260	60	70	60	N.A.	200	780	600	1380
Validation set 2	15	12	36	10	15	12	N.A.	20	120	60	180
Training set 3	61	N.A.	73	184	380	71	66	165	1000	875	1875
Validation set 3	12	N.A.	15	37	75	13	12	33	197	167	372
<b>Training set 1</b>	92	N.A.	116	262	656	125	65	465	1781	1781	3562
<b>Validation set 1</b>	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	N.A.	173

### Validation set 1, a peroxisomal proteins data set

A data set for a specific use case scenario was created using peroxisomal protein only. We searched on Uniprot (20/03/2022) for reviewed peroxisomal proteins correlated to the GO term ‘transport’, obtaining 172 entries using the query ‘locations:(location:"Peroxisome membrane [SL-0203"]) goa:("peroxisomal membrane [5778"]) goa:("transport [6810"]) AND reviewed:yes’. The complete list is available at [https://github.com/MarcoAnteghini/PortPred/blob/main/peroxisomal\\_proteins\\_dataset/protein\\_id\\_list.txt](https://github.com/MarcoAnteghini/PortPred/blob/main/peroxisomal_proteins_dataset/protein_id_list.txt)

### Training set 2 and Validation set 2 from TrSSP data set

The data set (composed of training and independent validation sets) for the model benchmarking with other available predictors was the same used in all of them [10, 12, 55]. This benchmarking data set (Training set 2 and Validation set 2) provided by Mishra et al., is collected from the Swiss-Prot database [9, 24] release 2013\_03. The TrSSP data set [9] contains a total of 1,560 sequences, divided into Training set 2 and Validation set 2 as shown in Table 7.2. The categories related to the 900 transporters present in the data set are 85 amino acid/oligopeptide transporters, 72 anion transporters, 296 cation transporters, 70 electron transporters, 85 protein/mRNA transporters, 72 sugar transporters, 220 other transporters. Also, 660 non-transporters were included as negatives. The data set can either be found at <https://www.zhaolab.org/TrSSP/?dowhat=datasets> or on GitHub at <https://github.com/MarcoAnteghini>.

### Training set 3 and Validation set 3 from FastTrans data set

As an additional data set for benchmarking our approach with the multiclass prediction, we used the same data set used for FastTrans by Nguyen et al. [12]. This data set is divided into Training set 3 and Validation set 3 (see Table 7.2). The protein sequence in this data set was retrieved from UniProt [24] (release 2018\_10) and contained proteins involved in the biological process of transporting ions/molecules. The data set does not contain fragmented sequences and sequences annotated with

more than two substrate specificities. In addition, sequences with more than 20% similarity were removed using PSI Blast [36]. The data set consists of 1050 membrane proteins (negatives) and 1197 transporters (positives). Note that the hydrogen ion substrate category from Nguyen et al. [12] is either called hydrogen ion or cation and represents the same set of proteins.

### 7.1.6 Classification Algorithms

The determination of transporter and non-transporter proteins is easily translated into a binary classification problem, while to distinguish among substrate categories we used a multi-class classification approach. For both tasks, we considered four widely used classification algorithms.

**Support Vector Machines (SVM)** is a supervised learning algorithm for two-group classification which aims to find the maximal margin hyperplane separating the points in the feature space [43, 56]. SVMs also perform non-linear classifications applying the kernel trick, thus implicitly mapping their inputs into high-dimensional feature spaces. In the case of multiple classes, multiple binary classification problems are performed. It can be done in two ways [57]: 1) *One-vs-One*, a binary classifier per each pair of classes; 2) *One-vs-Rest*, a binary classifier per class. In this study, we used the *One-vs-Rest* approach.

**Random Forest (RF)** is an ensemble learning method that, in the case of a classification task constructs a multitude of decision trees and outputs the mode of the classes of the individual trees [58, 59].

**Multilayer Perceptron (MLP)** is a class of feed-forward artificial neural networks that can distinguish among non-linearly separable data and uses backpropagation for training [60, 61]. Each node in an MLP, with the exception of the input node, uses a nonlinear activation function. In this study, we used the ReLu activation function [62].

**Logistic Regression (LR)** estimates the parameters of a logistic model [63]. In binary classifications, the corresponding probability of the values associated with two different labels can vary between 0 and 1. The multinomial LR model, for K possible outcomes, runs K-1 independent binary logistic regression models, in which one outcome is chosen as a "pivot" and then the other K-1 outcomes are separately regressed against the pivot outcome. We used a penalised implementation of multivariable logistic regression [64].

### 7.1.7 PortPred implementation

#### Model Training and Validation

In this study, we used three different training sets with no overlap between them and three independent validation sets.

Table 7.3: Hyperparameters for the grid searches. The `logspace` function, as available on NumPy [65] returns numbers spaced evenly on a log scale. In `logspace(start,stop,numbers)`, the sequence starts at base *start* (base to the power of start), ends with base *stop* and *numbers* is the number of samples to generate. The listed methods are: Support Vector Machines (SVM); Random Forest (RF); Multilayer Perceptron (MLP); Logistic Regression (LR)

method	hyperparameters	description	search space
SVM	C	Inverse of regularization strength	<code>logspace(-2,10,13)</code>
	gamma	Kernel coefficient	<code>logspace(-9,3,13)</code>
	kernel	Specifies the kernel type to be used in the algorithm	<code>'linear','poly','rbf','sigmoid'</code>
RF	n_estimators	The number of trees in the forest	15,25,50,75,100,200,300
	criterion	The function to measure the quality of a split	<code>'gini','entropy'</code>
	max_depth	The maximum depth of the tree	2,5,10,None
	min_samples_split	The minimum number of samples required to split an internal node	2,4,8,10
	max_features	The number of features to consider when looking for the best split	<code>'sqrt','auto','log2'</code>
MLP	hidden_layer_sizes	The number of neurons in each hidden layer	(200,),(100,),(50,),(200,100,6,1)
	activation	Activation function for the hidden layer	<code>'relu'</code>
	solver	The solver for weight optimization	<code>'lbfgs'</code>
	alpha	Strength of the L2 regularization term	1.0
LR	learning_rate	Learning rate schedule for weight updates	<code>'constant'</code>
	penalty	Specify the norm of the penalty	<code>'l1','l2'</code>
	solver	Algorithm to use in the optimization problem	<code>'liblinear','saga'</code>
	C	Inverse of regularization strength	<code>logspace(-3,9,13)</code>

**Training.** To be consistent with the other methods, each model was evaluated on the training data sets respectively using 10-fold cross-validation (10-CV) [66]. In every iteration, a single fold was kept as the testing set, and the remaining nine sets were used to train the respective model. The trained model was then tested using the test set. The procedure stops when all 10 subsets are used as a test once. The average performance for each model was considered as a single estimation. To obtain a stable error estimation, we repeated the 10-CV ten times with different random splits. The variations between runs were highlighted by the standard deviation. The cross-validation performances are reported as mean  $\pm$  standard deviation (SD) of the ten different runs of the 10-CV.

The cross-validation procedures include a (hyper)grid search: for each set of hyperparameters, the average classification score is computed across the folds. The hyperparameters corresponding to the best classification score are then used to fit a classification model whose quality is assessed on the validation set. The reference metric is the F1 score. The Hyperparameters optimisation details are shown in Table 7.3).

**Validation.** The independent data sets were used to perform additional validations. The data in the independent validation sets were not used during the cross-validation processes and are completely unknown to the models.

### Concatenation of Embeddings

To obtain a comprehensive overview of the single embeddings capabilities, we first evaluated each model using a single embedding and finally, we run a training and test procedure where every protein was represented with a concatenation of all the

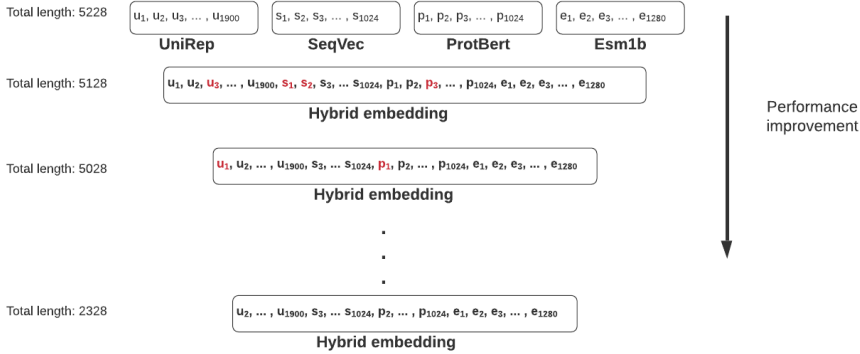


Figure 7.2: Schematic representation of the recursive feature elimination process. The initial data set contains a vector of length 5228. Each iteration remove a fixed number of random features, in this case, 100. The performance is then evaluated with the reduced embedding and the process continues until it worsens or the minimum number of features to consider has been reached. The final vector (length 2328) representation is saved.

available embeddings.

### Recursive Feature Elimination

Recursive Features Elimination (RFE) defines an optimal subset of informative features with respect to a given task. It starts considering all features in the training data set (the 4 concatenated embeddings in our case) and successfully removes one or more of them until the performance worsens or an arbitrary number of features remains. The performance is evaluated through a CV (10-CV here) classification. The approach creates a model where the desired input is a hybrid version of all the analysed embeddings. In particular, just the relevant features (values) of the concatenated embedding are kept (e.g. 2328 out of 5228). We used the RFECV function, available on scikit-learn that automatically selects the number of features chosen by RFE [67]. We adopted Logistic Regression as an estimator within the RFECV function, given its consistency during our initial estimations and its capability of working with both binary and multiclass classification tasks. In the RFECV function, the number of features to remove at each iteration must be specified, we used 100 in order to have a granular but fast process. The chosen metric for the performance optimisation was the F1 score. A detailed explanation of the metric can be seen in Section 7.1.8. An overview of the process is shown in Figure 7.2.

#### 7.1.8 Metrics

We used several metrics to quantify the quality of the classification models, namely: sensitivity (SEN), specificity (SPE), accuracy (ACC), F1 score [68], Matthews correlation coefficient (MCC) [69] and the area under the curve (AUC) of the receiver operating characteristic (ROC). Given that  $TP$  is the number of true positives,  $FP$  is the



number of false positives;  $TN$  and  $FN$  are the numbers of true and false negatives respectively, the following formulas are defined as:

Sensitivity ( $SEN$ ) or True positive rate ( $TPR$ ) is defined as

$$SEN/TPR = \frac{TP}{TP + FN} \quad (7.1)$$

Specificity ( $SPE$ ) is defined as

$$SPE = \frac{TN}{TN + FP} \quad (7.2)$$

Accuracy ( $ACC$ ) is defined as

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (7.3)$$

$F_1$  score [68] is defined as

$$F_1 = 2 \times \frac{PPV \times TPR}{PPV + TPR}, \quad (7.4)$$

where  $PPV$  is the positive predicted value (or precision)

$$PPV = \frac{TP}{TP + FP}, \quad (7.5)$$

The  $F_1$  score is the harmonic mean of recall and precision and varies between 0, if the precision or the recall is 0, and 1 indicating perfect precision and recall.

Matthews correlation coefficient ( $MCC$ ) [69] is defined as

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (7.6)$$

$MCC$  is the correlation coefficient between the true and predicted class: it is bound between  $-1$  (total disagreement between prediction and observation) and  $+1$  (perfect prediction); 0 indicates no better than random prediction. The  $MCC$  is appropriate also in presence of class unbalance [70].

The area under the curve ( $AUC$ ) of the receiver operating characteristic ( $ROC$ ) curve which plots the true positive proportion or the Sensitivity against the Specificity, is defined as

$$AUC = \int_0^1 TPRd(FPR) \quad (7.7)$$

The  $AUC$  analysis enables the evaluation of the performance of a binary classifier system according to the variation of the discrimination threshold. A perfect prediction has an  $AUC$  score of 1.0 while an  $AUC$  of 0.5 indicates randomness [71].

### 7.1.9 Software

A stand-alone version of the tool is available at <https://github.com/MarcoAnteghini/PortPred> together with the used data sets. Moreover, the data sets, together with an explanatory Jupyter notebook, are available at [https://drive.google.com/drive/folders/1L\\_zdaDa2EoPTWQz0dNqSHCweQFfixcsHe](https://drive.google.com/drive/folders/1L_zdaDa2EoPTWQz0dNqSHCweQFfixcsHe).

### 7.1.10 Embeddings correlation

Different embeddings store different information and in some cases, concatenating two or more embeddings can improve the performances [17]. In this study we first checked for a possible correlation between the embeddings using Pearson's correlation coefficient [72]. We observed that combining four different encodings and/or embeddings gives a better prediction of the peroxisomal sub-localisation. In particular, concatenating UniRep, SeqVec, PROTBERT and ESM-1b showed a noticeable improvement in the performances. That indicates that the four embeddings carry different and complementary information about the properties of the protein sequence, as given in Figure 7.3, which shows how the four embeddings are not correlated.

### 7.1.11 PortPred development

The best models for the prediction of transporter protein and their substrate, trained on the PortPred data set, were used for the final tool. It consists of two classification steps. First, the classifier distinguishes between transporter and non-transporter proteins. Secondly, a multiclass classification is performed. The final output is a specific transporter category (lipid, sugar, protein/mRNA, electron, hydrogen ion, amino acid and other). A scheme of the tool functionalities is shown in Figure 7.4.

#### Transporter vs Non Transporter prediction on Training set 1

We analysed the performances of each embedding for a binary classification task ('transporter' vs 'non-transporter') on the Training set 1 and the Validation set 2 (reported in table as Ind.), explained in Section 7.1.5. As classifiers, we tested LR, RF, SVM and MLP (see Section 7.1.6). The results in terms of sensitivity, specificity, accuracy, Matthew-correlation-coefficient, area under the curve and f1 score are shown in Table 7.4 (see Section 7.1.8 for details about the metrics). The ESM-1b embedding coupled with an SVM classifier reaches the best performances with an F1 score of 84.65% and ACC of 84.67% during cross-validation.

Secondly, we analysed the concatenated embeddings performances, and the hybrid embeddings performances (details in Section 7.1.7) on the same data set. Results are shown in Table 7.5. In this case, the hybrid embeddings obtained with an RFE procedure slightly outperformed the concatenated embeddings. The best classifier in handling the hybrid embedding is SVM, reaching ACC of 84.27% in the cross-validation.

We reported the results with and without the ESM-1b embedding to have an additional comparison since it cannot handle long protein sequences (longer than 1024 residues).

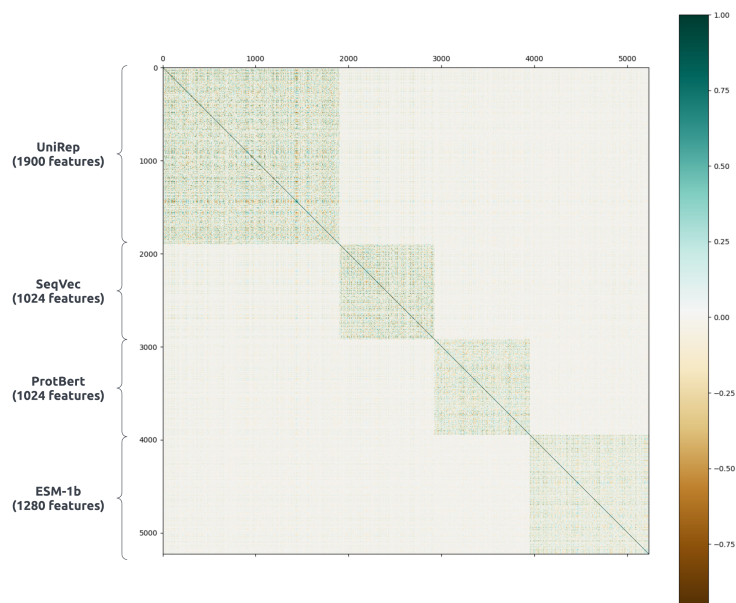


Figure 7.3: Correlation among UniRep (1900 features), SeqVec (1024 features), PortPred (1024 features) and ESM-1b (1280 features) protein sequence embeddings. Pearson's linear correlation is used and are calculated over 1580 transporter protein sequences of the PorPred training set. The four embeddings are uncorrelated

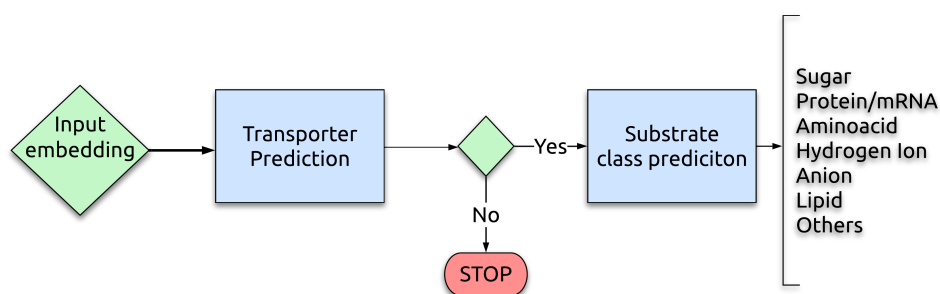


Figure 7.4: A schematic representation of the PortPred tool functionalities in three steps. 1) The algorithm takes as input the embedding of a protein sequence; 2) IF condition. If the protein is predicted as transporter the algorithm proceed, otherwise it stops; 3) The algorithm predicts which substrate the transporter carries.

Table 7.4: Performances of each embeddings in predicting transporter proteins in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient (MCC), area under the curve (ROC AUC) and F1 score (F1). Each classifier, namely: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), Multilayer Perceptron (MLP) was evaluated with a 10-fold Cross-Validation (see columns CV). Each model was then validated on an independent data set (see columns Ind.). The independent data set is the one described in Validation set 2 (Section 7.1.5). The subtables represent the single embedding performances: a) UniRep; b) SeqVec; c) PROTBERT; d) ESM-1b.

a) UniRep										
classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	84.17	76.70 ± 0.47	70.00	76.61 ± 0.49	79.44	76.65 ± 0.03	0.5395	0.5333 ± 0.0061	77.08	76.65 ± 0.03
RF	89.17	75.81 ± 0.28	78.33	78.63 ± 0.32	85.56	77.22 ± 0.21	0.6750	0.5451 ± 0.0042	83.75	77.22 ± 0.21
<b>SVM</b>	83.33	77.58 ± 0.82	73.33	77.20 ± 0.66	80.00	77.39 ± 0.27	0.5581	0.5482 ± 0.0053	78.33	77.39 ± 0.27
MLP	90.00	76.24 ± 0.51	75.00	75.08 ± 0.56	85.00	75.66 ± 0.19	0.6587	0.5136 ± 0.0038	82.5	75.66 ± 0.19

b) SeqVec										
classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	89.17	78.62 ± 0.38	73.33	79.65 ± 0.55	83.89	79.13 ± 0.35	0.6334	0.5830 ± 0.0070	81.25	79.13 ± 0.35
RF	90.00	74.99 ± 0.50	78.33	81.20 ± 0.56	86.11	78.10 ± 0.33	0.6862	0.5633 ± 0.0067	84.17	78.10 ± 0.33
<b>SVM</b>	91.67	80.65 ± 0.43	81.67	80.16 ± 0.31	88.33	80.40 ± 0.30	0.7365	0.6084 ± 0.0060	86.67	80.40 ± 0.30
MLP	90.83	79.87 ± 0.49	73.33	78.61 ± 0.51	85.00	79.24 ± 0.36	0.6567	0.5850 ± 0.0071	82.08	79.24 ± 0.36

c) PROTBERT										
classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	90.83	78.40 ± 0.41	78.33	79.02 ± 0.50	86.67	78.71 ± 0.29	0.6977	0.5745 ± 0.0059	84.58	78.71 ± 0.29
RF	90.83	78.40 ± 0.39	70.00	76.73 ± 0.33	83.89	77.56 ± 0.22	0.6292	0.5515 ± 0.0044	80.42	77.56 ± 0.22
<b>SVM</b>	93.33	78.70 ± 0.45	73.33	81.09 ± 0.44	86.67	79.89 ± 0.45	0.6934	0.5984 ± 0.0089	83.33	79.89 ± 0.45
MLP	89.17	79.19 ± 0.32	76.67	78.62 ± 0.60	85.00	78.91 ± 0.43	0.6611	0.5784 ± 0.0088	82.92	78.91 ± 0.43

d) ESM-1b										
classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	89.72	82.42 ± 0.40	76.36	83.16 ± 0.54	85.19	82.80 ± 0.38	0.6670	0.6560 ± 0.0074	83.04	82.79 ± 0.37
RF	92.52	80.23 ± 0.55	85.45	81.30 ± 0.39	90.12	80.78 ± 0.24	0.7798	0.6157 ± 0.0050	88.99	80.76 ± 0.24
<b>SVM</b>	96.26	84.25 ± 0.58	80.00	85.06 ± 0.57	90.74	84.67 ± 0.41	0.7909	0.6935 ± 0.0080	88.13	84.66 ± 0.41
MLP	90.65	81.44 ± 0.47	74.55	81.61 ± 0.66	85.19	81.53 ± 0.39	0.6648	0.6307 ± 0.0076	82.60	81.52 ± 0.38

## Validation on the peroxisomal proteins data set

We tested the tool with a subset of peroxisomal proteins called Validation set 1 (Section 7.1.5). The predictor produced promising results, thus allowing us to suggest new transporter protein candidates in peroxisomes. In particular, we analyzed the predictor performances in highlighting transporter proteins in a generic subset of peroxisomal proteins that have been associated with transport functions in Uniprot. 26 proteins out of 167 were identified as non-transporter proteins. Looking into this predicted negative data set we realised that just 3 out of 26 had a clear transporter function (True Negatives) while the remaining are part of more complex machinery not directly connected with transporter function. For example PEX12 (UniprotID: Q8VC48), predicted as negative, is a peroxisome assembly protein. More precisely, it is a component of a retrotranslocation channel required for peroxisome organization. This proteins only forms a channel once assembled with PEX2 and PEX10. The complete list of predictions is available at [https://drive.google.com/drive/folders/1XKn0Rs8uEb\\_T61Nhg10aCx8Pzsc0rRNA](https://drive.google.com/drive/folders/1XKn0Rs8uEb_T61Nhg10aCx8Pzsc0rRNA).

We performed a manual curation for all the entries (available at [https://github.com/MarcoAnteghini/PortPred/blob/main/peroxisomal\\_proteins\\_dataset/Validation\\_set1.csv](https://github.com/MarcoAnteghini/PortPred/blob/main/peroxisomal_proteins_dataset/Validation_set1.csv)). From the manual curation, the estimated performances are ACC: 70.66%, ROC AUC: 82.62%, MCC: 0.4756, SEN: 65.24 %, SPE: 1.0%, F1:

classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	90.08	79.35 ± 0.46	81.36	80.96 ± 0.44	87.22	80.15 ± 0.21	0.7114	0.6035 ± 0.0043	85.72	80.16 ± 0.21	90.46	80.15 ± 0.21
RF	90.32	78.16 ± 0.51	85.71	80.96 ± 0.41	88.89	79.56 ± 0.36	0.7467	0.5917 ± 0.0070	88.02	79.56 ± 0.36	91.8	79.55 ± 0.36
SVM	92.31	79.82 ± 0.41	80.95	81.97 ± 0.37	88.33	80.89 ± 0.26	0.7412	0.6183 ± 0.0052	86.63	80.90 ± 0.26	91.14	80.89 ± 0.26
MLP	89.34	80.26 ± 0.43	81.03	79.43 ± 0.39	86.67	79.85 ± 0.20	0.6977	0.5973 ± 0.0040	85.19	79.84 ± 0.19	90.08	79.84 ± 0.19

classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	90.00	84.18 ± 0.38	80.00	84.89 ± 0.24	86.67	84.54 ± 0.24	0.7000	0.6909 ± 0.0047	85.00	84.54 ± 0.24	90.00	84.53 ± 0.24
RF	91.20	78.17 ± 0.42	89.09	81.54 ± 0.37	90.56	79.86 ± 0.30	0.7846	0.5978 ± 0.0060	90.15	79.86 ± 0.30	93.06	79.85 ± 0.30
SVM	92.31	84.60 ± 0.36	80.95	83.93 ± 0.61	88.33	84.27 ± 0.36	0.7412	0.6857 ± 0.0073	86.63	84.26 ± 0.36	91.14	84.26 ± 0.36
MLP	87.50	83.60 ± 0.46	84.62	82.98 ± 0.50	86.67	83.29 ± 0.38	0.6934	0.6661 ± 0.0078	86.06	83.29 ± 0.38	90.32	83.29 ± 0.38

classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	89.19	83.49 ± 0.30	84.31	82.65 ± 0.28	87.65	83.07 ± 0.26	0.7209	0.6617 ± 0.0052	86.75	83.07 ± 0.26	90.83	83.07 ± 0.26
RF	90.09	80.02 ± 0.50	86.27	80.72 ± 0.46	88.89	80.37 ± 0.28	0.7490	0.6078 ± 0.0056	88.18	80.37 ± 0.29	91.74	80.36 ± 0.28
SVM	90.00	84.47 ± 0.50	84.62	83.84 ± 0.41	88.27	84.16 ± 0.39	0.7356	0.6834 ± 0.0078	87.31	84.15 ± 0.39	91.24	84.15 ± 0.39
MLP	88.24	82.75 ± 0.58	71.67	80.82 ± 0.51	82.10	81.80 ± 0.38	0.6109	0.6363 ± 0.0076	79.95	81.78 ± 0.38	86.12	81.78 ± 0.38

classifier	SEN %		SPE %		ACC %		MCC		ROC AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	89.19	83.74 ± 0.42	84.31	83.12 ± 0.60	87.65	83.43 ± 0.43	0.7209	0.6690 ± 0.0088	86.65	83.43 ± 0.43	90.83	83.42 ± 0.43
RF	90.91	80.46 ± 0.52	86.54	81.10 ± 0.50	89.51	80.78 ± 0.38	0.7635	0.6159 ± 0.0077	88.72	80.78 ± 0.38	92.17	80.77 ± 0.38
SVM	90.00	84.18 ± 0.33	84.62	84.37 ± 0.47	88.27	84.27 ± 0.30	0.7356	0.6859 ± 0.0061	87.31	84.27 ± 0.31	91.24	84.27 ± 0.30
MLP	87.27	81.52 ± 0.43	78.85	80.56 ± 0.68	84.57	81.05 ± 0.37	0.6519	0.6212 ± 0.0075	83.06	81.05 ± 0.37	88.48	81.04 ± 0.37

Table 7.5: Performances of the concatenated embeddings based models, trained on the Train-set 1 (PortPred data set). Performances reflect the models capability in predicting transporter proteins in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient (MCC), area under the curve (ROC AUC) and F1 score (F1). Each classifier, namely: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), Multilayer Perceptron (MLP) was evaluated with a 10-fold Cross-Validation (see columns CV). Each model was then validated on an independent data set (see columns Ind.). The independent data set is the one described in Validation set 2 (Section 7.1.5). The subtables represent the concatenated embedding performances. Table a) represent the concatenated embeddings performances with the exclusion of ESM-1b; Table b) represent the concatenated embeddings performances of all the embeddings; Table c) represent the concatenated embeddings performances after performing a Recursive Feature Elimination and with the exclusion of ESM-1b; Table d) represent the concatenated embeddings performances after performing a Recursive Feature Elimination.

65.23%.

### **Transporter vs Non-Transporter prediction on Training set 2 and benchmarking**

We first analysed the performances of each embedding for a binary classification task ('transporter' vs 'non-transporter') on the Training set 2 and the Validation set 2 (TrSSP benchmark data set). As classifiers, we tested LR, RF, SVM and MLP (see Section 3.4.6). The results in terms of sensitivity, specificity, accuracy, Matthew-correlation-coefficient, area under the curve and f1 score are shown in Table 7.6 (see Section 7.1.8 for details). The ESM-1b embedding coupled with an SVM classifier reaches the best performances with an F1 score of 85.54% and ACC of 88.70%.

Secondly, we analysed the concatenated embeddings performances and the hybrid embeddings performances (details in Section 7.1.7) on the same benchmark data set (TrSSP). Results are shown in Table 7.7. In this case, the hybrid embeddings obtained with an RFE procedure outperformed the concatenated embeddings. Both, in general, outperform the single embeddings' performances. The best classifier in handling the hybrid embedding is LR, reaching an F1 score of 94.45% and ACC of 94.53% in the cross-validation.

We reported the results with and without the ESM-1b embedding to have an additional comparison since it cannot handle long protein sequences (longer than 1024 residues). Nevertheless, the average length of a transporter protein sequence found on Swissprot (20.01.2023) is 447 residues and the median is 347. We computed this value by averaging the length of 7420 proteins. These proteins were found with the query '(cc\_function:transporter) AND (length:[50 TO \*]) AND (reviewed:true) AND (fragment:false)'.

Finally, we report the performances of our best classifier trained on the Training set 2 against the results of recently published works [9, 10, 12, 13]. Results are visible in Table 7.8. Our model outperforms the state-of-the-art in predicting transporter proteins with an ACC of 94.53% on the independent validation set.

### **Transporter substrate categories prediction and benchmarking**

Our method's capability in predicting substrate-specific transporter proteins is shown in Table 7.9. The results in terms of F1 score and MCC show the consistency of our method when trained on two different data sets (Training set 1 and Training set 3) and validated on the same independent data set (Validation set 3). The Training set 3 comes from the work of Nguyen et al. (2019), reported as FastTrans [12]. Training set 1 is a newly generated PortPred data set. For details about the data set refer to Section 7.1.5.

As an additional comparison, Table 7.10 reports similar performances of PortPred trained with Training set 1 and Training set 3 in classifying specific kinds of transporter proteins. Moreover performances of our PortPred model validated on the Validation set 3, retrieved from the FastTrans paper, are visible as confusion matrix in Figure 7.5.

Table 7.6: Performances of each embeddings in predicting transporter proteins in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient (MCC), area under the curve (AUC) and F1 score (F1). Each classifier, namely: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), Multilayer Perceptron (MLP) was evaluated with a 10-Fold Cross-Validation (see columns CV) on the Training set 2. Each model was then validated on the Validation set 2 (see columns Ind.). The subtables represent the single embedding performances: a) UniRep; b) SeqVec; c) PROTBERT; d) ESM-1b.

a) UniRep												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	87.50	83.73 ± 0.78	70.00	82.47 ± 0.60	81.67	83.18 ± 0.42	0.5827	0.6604 ± 0.0080	78.75	83.10 ± 0.39	86.42	82.95 ± 0.42
RF	89.17	81.47 ± 0.81	81.67	87.48 ± 0.62	86.67	84.09 ± 0.33	0.7027	0.6849 ± 0.0058	85.42	84.48 ± 0.29	89.92	83.98 ± 0.33
SVM	86.67	83.58 ± 1.520	76.67	83.80 ± 0.148	83.33	83.67 ± 0.31	0.6283	0.6717 ± 0.0048	81.67	83.69 ± 0.23	87.39	83.47 ± 0.29
MLP	88.33	84.67 ± 0.68	66.67	80.63 ± 0.88	81.11	82.91 ± 0.60	0.5658	0.6533 ± 0.0119	77.50	82.65 ± 0.61	86.18	82.62 ± 0.61

b) SeqVec												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	87.50	85.01 ± 0.63	88.33	85.02 ± 1.04	87.78	85.01 ± 0.55	0.7373	0.6979 ± 0.0114	87.92	85.01 ± 0.58	90.52	84.82 ± 0.56
RF	87.50	82.22 ± 0.42	83.33	88.18 ± 0.78	86.11	84.81 ± 0.37	0.6952	0.6990 ± 0.0078	85.42	85.20 ± 0.40	89.36	84.70 ± 0.38
SVM	86.67	82.35 ± 0.168	91.67	89.85 ± 1.39	88.33	85.61 ± 0.47	0.7556	0.7170 ± 0.0079	89.17	86.10 ± 0.36	90.83	85.52 ± 0.45
MLP	90.00	85.54 ± 0.64	90.00	83.33 ± 0.121	90.00	84.57 ± 0.71	0.7826	0.6874 ± 0.0146	90.00	84.42 ± 0.75	92.31	84.33 ± 0.73

c) PROTBERT												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	93.33	85.77 ± 0.47	83.33	82.97 ± 0.71	90.0	84.55 ± 0.46	0.7734	0.6871 ± 0.0091	88.33	84.37 ± 0.47	92.56	84.30 ± 0.47
RF	91.67	84.63 ± 0.62	78.33	77.58 ± 0.86	87.22	81.57 ± 0.43	0.7094	0.6249 ± 0.0089	85.00	81.11 ± 0.45	90.53	81.17 ± 0.44
SVM	95.00	84.87 ± 0.13	81.67	84.90 ± 0.91	90.56	84.88 ± 0.57	0.7846	0.6955 ± 0.0106	88.33	84.89 ± 0.50	93.06	84.69 ± 0.55
MLP	95.83	86.00 ± 0.79	81.67	82.70 ± 0.76	91.11	84.57 ± 0.59	0.7972	0.6871 ± 0.0118	88.75	84.35 ± 0.59	93.50	84.30 ± 0.60

d) ESM-1b												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	94.39	88.70 ± 0.31	90.91	85.60 ± 0.37	93.21	87.34 ± 0.28	0.8493	0.7441 ± 0.0052	92.65	87.15 ± 0.28	94.84	87.14 ± 0.28
RF	95.33	88.53 ± 0.43	85.45	78.42 ± 0.88	91.98	84.08 ± 0.37	0.8194	0.6769 ± 0.0078	90.39	83.47 ± 0.41	94.01	83.70 ± 0.39
<b>SVM</b>	92.52	89.47 ± 0.81	90.91	87.71 ± 0.46	91.98	88.70 ± 0.35	0.8241	0.7718 ± 0.0070	91.72	88.59 ± 0.30	93.84	88.54 ± 0.34
MLP	92.52	88.34 ± 0.109	87.27	83.65 ± 0.131	90.74	86.28 ± 0.36	0.7945	0.7224 ± 0.0073	89.9	86.00 ± 0.38	92.96	86.05 ± 0.37

Table 7.7: Performances of the concatenated embeddings based models, trained on the Training set 2. Performances reflect the models capability in predicting transporter proteins in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient (MCC), area under the curve (ROC AUC) and F1 score (F1). Each classifier, namely: Logistic Regression (LR), Random Forest (RF), Support Vector Machines (SVM), Multilayer Perceptron (MLP) was evaluated with a 10-Fold Cross-Validation (see columns CV). Each model was then tested on the Validation set 2 (see columns Ind.). The subtables represent the concatenated embedding performances. Table a) represent the concatenated embeddings performances with the exclusion of ESM-1b; Table b) represent the concatenated embeddings performances of all the embeddings; Table c) represent the concatenated embeddings performances after performing a Recursive Feature Elimination and with the exclusion of ESM-1b; Table d) represent the concatenated embeddings performances after performing a Recursive Feature Elimination.

a) UniRep + SeqVec + PROTBERT												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	91.60	85.70 ± 0.70	81.97	85.50 ± 0.5	88.33	85.60 ± 0.40	0.7386	0.7090 ± 0.0007	86.78	85.6 ± 0.40	91.21	85.40 ± 0.40
RF	94.83	82.17 ± 0.57	84.38	88.28 ± 0.93	91.11	84.83 ± 0.53	0.8043	0.6996 ± 0.0112	89.6	85.23 ± 0.56	93.22	84.72 ± 0.54
<b>SVM</b>	93.28	85.6 ± 0.70	85.25	88.17 ± 0.42	90.56	86.72 ± 0.40	0.7884	0.7340 ± 0.0077	89.26	86.88 ± 0.37	92.89	86.58 ± 0.4
MLP	90.83	86.44 ± 0.61	81.67	84.10 ± 0.93	87.78	85.42 ± 0.42	0.7250	0.7049 ± 0.0082	86.25	85.27 ± 0.45	90.83	85.19 ± 0.44

b) RFE - UniRep + SeqVec + PROTBERT												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	90.16	85.60 ± 0.58	82.76	85.48 ± 0.61	87.78	85.55 ± 0.45	0.7229	0.7086 ± 0.0090	86.46	85.54 ± 0.45	90.91	85.36 ± 0.46
RF	95.65	82.95 ± 0.23	84.62	88.65 ± 0.69	91.67	85.43 ± 0.32	0.8179	0.7110 ± 0.0071	90.13	85.8 ± 0.36	93.62	85.32 ± 0.33
<b>SVM</b>	90.16	84.99 ± 0.93	82.76	88.72 ± 0.48	87.78	86.61 ± 0.62	0.7229	0.7328 ± 0.0119	86.46	86.85 ± 0.58	90.91	86.48 ± 0.62
MLP	94.07	86.23 ± 0.85	85.48	84.28 ± 1.23	91.11	85.38 ± 0.74	0.8019	0.7043 ± 0.0150	89.78	85.26 ± 0.76	93.28	85.16 ± 0.75

c) UniRep + SeqVec + PROTBERT + ESM-1b												
classifier	SEN %		SPE %		ACC %		MCC		AUC %		F1 %	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
LR	94.34	88.46 ± 0.38	87.50	86.95 ± 0.70	91.98	87.79 ± 0.37	0.8219	0.7546 ± 0.0077	90.92	87.70 ± 0.37	93.90	87.62 ± 0.37
RF	95.33	84.42 ± 0.45	90.91	87.84 ± 0.70	93.83	85.92 ± 0.38	0.8624	0.7200 ± 0.0074	93.12	86.13 ± 0.39	95.33	85.80 ± 0.38
<b>SVM</b>	94.44	89.59 ± 0.51	90.74	88.04 ± 0.59	93.21	88.90 ± 0.30	0.8480	0.7764 ± 0.0064	92.59	88.75 ± 0.30	94.88	88.81 ± 0.31
MLP	95.15	88.84 ± 0.45	84.75	85.42 ± 0.73	91.36	87.34 ± 0.47	0.8118	0.7445 ± 0.0094	89.95	87.13 ± 0.49	93.33	87.13 ± 0.48

d) RFE - UniRep + SeqVec + PROTBERT + ESM-1b												
classifier	SEN		SPE		ACC		MCC		ROC AUC		F1	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
<b>LR</b>	95.24	95.06 ± 0.37	87.72	93.85 ± 0.31	92.59	94.53 ± 0.26	0.8366	0.8896 ± 0.0054	91.48	94.46 ± 0.26	94.34	94.45 ± 0.27
RF	94.39	85.61 ± 0.52	89.09	89.22 ± 0.84	92.59	87.19 ± 0.57	0.8348	0.7452 ± 0.0114	91.74	87.41 ± 0.59	94.39	87.09 ± 0.57
SVM	94.29	95.02 ± 0.48	85.96	93.84 ± 0.48	91.36	94.50 ± 0.33	0.8093	0.8890 ± 0.0067	90.13	94.42 ± 0.33	93.4	94.43 ± 0.32
MLP	95.24	95.01 ± 0.31	87.72	93.71 ± 0.36	92.59	94.44 ± 0.20	0.8366	0.8877 ± 0.0039	91.48	94.36 ± 0.20	94.34	94.35 ± 0.20

Table 7.8: Performance comparison between the proposed method (PortPred) and those of recently published works in predicting transporter proteins trained on the Training set 2. Performances are measured in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient(MCC), area under the curve (ROC AUC) and F1 score (F1). The evaluation was performed on 10-Fold Cross-Validation data (see CV columns) and on an independent data set (see Ind. columns).

tool	SEN %		SPE %		ACC %		MCC	
	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV
<b>PortPred</b>	95.24	<b>92.59</b>	87.72	<b>93.85</b>	<b>92.59</b>	<b>94.53</b>	<b>0.84</b>	<b>0.89</b>
SCMMTP	80.00	83.76	68.33	77.68	76.11	81.12	0.47	0.62
TrSSP	76.67	76.67	81.67	78.46	80.00	78.99	0.57	0.58
FastTrans	<b>10.00</b>	83.14	77.50	84.48	85.00	83.94	0.73	0.68
TooT-T	94.17	90.15	<b>88.33</b>	89.97	92.22	90.07	0.82	0.80



Table 7.9: Performances of the PortPred model, trained on the Training set 1 and the Training set 3 in terms of F1 score (macro average) and Matthew-correlation-coefficient (MCC). The first column indicates the data set; the second column indicates the classifier among Logistic Regression (LR), Random forest (RF), Support vector machines (SVM) and Multilayer perceptron (MLP); from the third column the performances are shown in terms of F1 and MCC, whit the CV indicating the cross-validation process on the specific Training set (1 or 3) and the column Ind. indicating the performances of the model trained on the specific Training set (1 or 3) but tested only against Validation set 3.

data set	classifier	F1 %		MCC	
		CV	Ind.	CV	Ind.
1	LR	88.24 $\pm$ 0.57	89.14	0.9184 $\pm$ 0.0031	0.9071
3		96.65 $\pm$ 0.24	90.26	0.9706 $\pm$ 0.0024	0.9137
1	RF	64.13 $\pm$ 0.29	66.63	0.7406 $\pm$ 0.0064	0.7512
3		61.42 $\pm$ 0.37	65.04	0.7099 $\pm$ 0.0067	0.7619
1	SVM	88.15 $\pm$ 0.61	88.92	0.9183 $\pm$ 0.0044	0.8980
3		88.44 $\pm$ 0.67	89.89	0.8967 $\pm$ 0.0058	0.9135
1	MLP	86.55 $\pm$ 0.64	88.51	0.9057 $\pm$ 0.0032	0.9274
3		87.45 $\pm$ 0.66	91.56	0.8900 $\pm$ 0.0058	0.9283

Table 7.10: Performances of the PortPred model in a multiclass prediction task. The model was trained on Training set 1 and Training set 3. Results are shown in terms of sensitivity (SEN), specificity (SPE), accuracy (ACC), matthew-correlation-coefficient (MCC) and area under the curve (ROC AUC) (F1). The first column indicates the data set; the second column indicates the substrates; from the third column the performances are shown, whit the CV indicating the cross-validation process on the specific Training set (1 or 3) and the column Ind. indicating the performances on the model trained on the specific Training set (1 or 3) but tested only against Validation set 3.

Training set	substrate	SEN		SPE		ACC		AUC		MCC	
		CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.	CV	Ind.
1	Amino acid	0.90 $\pm$ 0.10	0.83	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.98	0.95 $\pm$ 0.05	0.91	0.91 $\pm$ 0.06	0.86
3		0.89 $\pm$ 0.12	0.75	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.98	0.94 $\pm$ 0.04	0.87	0.93 $\pm$ 0.07	0.81
1	Electron	0.94 $\pm$ 0.02	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.98 $\pm$ 0.02	0.99	0.95 $\pm$ 0.02	0.99
3		0.99 $\pm$ 0.02	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.98	0.99 $\pm$ 0.01	0.99	0.98 $\pm$ 0.02	0.95
1	Hydrogen ion	0.90 $\pm$ 0.08	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.95 $\pm$ 0.04	0.99	0.93 $\pm$ 0.03	0.99
3		0.91 $\pm$ 0.11	0.93	0.99 $\pm$ 0.01	0.98	0.99 $\pm$ 0.01	0.99	0.95 $\pm$ 0.06	0.96	0.93 $\pm$ 0.09	0.93
1	Lipid	0.50 $\pm$ 0.16	0.99	0.99 $\pm$ 0.01	0.99	0.98 $\pm$ 0.01	0.99	0.75 $\pm$ 0.08	0.99	0.64 $\pm$ 0.12	0.99
3		0.87 $\pm$ 0.11	0.78	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.93 $\pm$ 0.06	0.89	0.92 $\pm$ 0.08	0.88
1	Protein/mRNA	0.98 $\pm$ 0.01	0.99	0.98 $\pm$ 0.01	0.99	0.98 $\pm$ 0.01	0.99	0.98 $\pm$ 0.01	0.99	0.95 $\pm$ 0.01	0.99
3		0.99 $\pm$ 0.01	0.99	0.98 $\pm$ 0.01	0.98	0.99 $\pm$ 0.01	0.98	0.99 $\pm$ 0.01	0.98	0.98 $\pm$ 0.03	0.96
1	Sugar	0.93 $\pm$ 0.09	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.96 $\pm$ 0.05	0.99	0.92 $\pm$ 0.09	0.96
3		0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.99	0.96 $\pm$ 0.03	0.93
1	Others	0.94 $\pm$ 0.02	0.97	0.97 $\pm$ 0.19	0.99	0.96 $\pm$ 0.01	0.99	0.95 $\pm$ 0.01	0.98	0.90 $\pm$ 0.02	0.96
3		0.98 $\pm$ 0.03	0.91	0.99 $\pm$ 0.01	0.99	0.99 $\pm$ 0.01	0.98	0.98 $\pm$ 0.02	0.95	0.95 $\pm$ 0.04	0.92

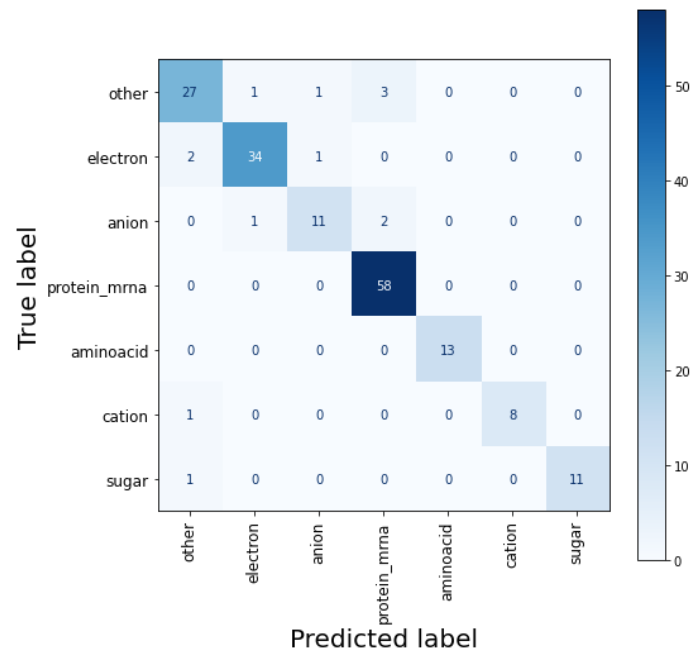


Figure 7.5: Confusion matrix of the best performing models for the different transporter categories. This matrix represent the results of the PortPred model trained on the Training set 3 and tested on Validation set 3

### 7.1.12 Discussion

Transporter proteins play a crucial role in the transport of ions, small molecules, and macromolecules across biological membranes. They are essential for the functioning of all living organisms and are frequently studied as drug targets due to their association with various diseases. The research on transporter proteins has significantly increased since their first discovery and characterization. In this study, we focused on developing PortPred, a prediction tool for transporter proteins, using deep learning (DL) approaches and protein sequence embeddings.

DL-based sequence embeddings have shown promising results in various bioinformatics tasks, including subcellular and sub-organelle classification, protein structure and function prediction, and protein-protein interactions. Inspired by these advancements, we explored the use of DL-based sequence embeddings for the accurate identification and classification of transporter proteins.

To develop PortPred, we reviewed and compared several DL-based protein embeddings, including UniRep, SeqVec, PROTBERT, and ESM-1b. These embeddings capture the underlying features and representations of protein sequences and can be used to encode protein information for downstream prediction tasks. We evaluated the performance of each embedding and their combination in predicting transporter proteins and differentiating among various categories of transporters.

Our comprehensive analysis revealed that hybrid embeddings, which combine multiple embeddings with a feature selection procedure, generally outperformed single embeddings alone. The combination of embeddings provided a more informative representation of transporter proteins, leading to improved prediction performance. However, we observed that ESM-1b embedding alone showed comparable or even higher performances than the hybrid embedding, demonstrating the effectiveness of this DL-based approach.

PortPred outperformed existing methods in predicting transporter proteins, achieving an accuracy of 94.53%. It also demonstrated excellent performance in classifying different categories of transporter proteins, with an average accuracy of 98.71%. These results highlight the robustness and reliability of PortPred for transporter protein prediction.

Furthermore, we emphasize the importance of critically evaluating and validating DL methodologies in bioinformatics. It is essential to avoid the trend of simply improving the state-of-the-art without thoroughly assessing the reliability and interpretability of the results. Therefore, we performed consistent benchmark analyses to validate the performance of PortPred and ensure the reproducibility of our findings in a FAIR (Findable, Accessible, Interoperable, and Reusable) manner.

In practical applications, PortPred shows promising potential. When applied to a real-world case scenario involving peroxisomal proteins, PortPred achieved an accuracy of 82.62% and exhibited high specificity, making it a reliable tool for avoiding false positives. This highlights the practical utility of PortPred in identifying transporter proteins and can aid in the understanding of their biological functions and implications.

In conclusion, our study demonstrates the adaptability and effectiveness of DL-based sequence embeddings for transporter protein prediction. We encourage the scientific community to make informed choices when selecting and utilizing DL-based pre-trained representations, considering the specific requirements and char-

acteristics of their prediction tasks. Furthermore, we advocate for rigorous validation, adaptation, and reporting of the limitations of these embeddings to ensure their reliability and usefulness in extracting meaningful biological insights.

## 7.2 Conclusion

The continuous application of DL approaches in bioinformatics increases the risk of underestimating the biological focus when testing new DL methodologies. This is linked to the fact that extracting biological insights from neural network architectures is not directly possible. However, it is relevant to validate our discoveries through consistent analyses. In particular, pre-trained DL models for protein sequence embedding are practical resources that require validation for specific prediction tasks.

When dealing with these methodologies, it is essential to be as critical as possible and not promote the trend of improving the state-of-the-art with a new tool that has 1% accuracy more than the others without first checking how reliable the results are. For this reason, we performed consistent benchmark analyses to prove the reliability of our discoveries in a FAIR manner.

We explored the usage of pretrained DL-based sequence embeddings for a specific use case. In particular, we tackled the problem of accurately classifying transporter proteins and differentiating among various categories of transporters. For this task, we developed the PortPred classifier.

Our comprehensive and unbiased analysis showed that in most of the cases, hybrid embedding is more informative than a single embedding alone. In addition, hybrid embedding obtained with a feature selection procedure is tailored to use-case-specific prediction tasks, thus improving the performances.

However, ESM-1b embedding alone, shows similar or even higher performances (Table 7.4 vs Table 7.5) than the hybrid embedding, demonstrating the effectiveness of this DL-based approach.

PortPred outperforms existing methods when predicting transporter proteins (ACC of 94.53%) and reaches optimal performances in classifying different categories of transporter proteins (average ACC of 98.71%). Nevertheless, single embeddings alone can still be used for accurate predictions, as shown by our analysis.

Ultimately, our predictor shows promising applications in a real-world case scenario. Running PortPred against the peroxisomal proteins in Validation set 1, we reached an accuracy of 82.62%. The model shows very high specificity 0.99%, thus making it a reliable tool for avoiding false positives.

## Acknowledgement

This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Hediger, M. A. et al. "The ABCs of membrane transporters in health and disease (SLC series): Introduction". In: *Molecular Aspects of Medicine* 34.2 (2013). The ABCs of membrane transporters in health and disease (SLC series), 95–107.
- [2] Sahoo, S. et al. "Membrane transporters in a human genome-scale metabolic knowledgebase and their implications for disease". In: *Frontiers in Physiology* 5 (2014), 91.
- [3] Robey, R. W. et al. "Revisiting the role of ABC transporters in multidrug-resistant cancer". In: *Nature Reviews Cancer* 18.7 (2018), 452–464.
- [4] Dahl, S. G., Sylte, I., and Ravna, A. W. "Structures and Models of Transporter Proteins". In: *Journal of Pharmacology and Experimental Therapeutics* 309.3 (2004), 853–860. DOI: 10.1124/jpet.103.059972.
- [5] Saier, M. H. "A Functional-Phylogenetic Classification System for Transmembrane Solute Transporters". In: *Microbiology and Molecular Biology Reviews* 64.2 (2000), 354–411.
- [6] Busch, W. and Saier, M. H. "The Transporter Classification (TC) System, 2002". In: *Critical Reviews in Biochemistry and Molecular Biology* 37.5 (2002), 287–337.
- [7] Saier Milton H, J. et al. "The Transporter Classification Database (TCDB): 2021 update". In: *Nucleic Acids Research* 49.D1 (2020), D461–D467.
- [8] Berman, H. M. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (2000), 235–242. DOI: 10.1093/nar/28.1.235. URL: <https://doi.org/10.1093/nar/28.1.235>.
- [9] Mishra, N. K., Chang, J., and Zhao, P. X. "Prediction of Membrane Transport Proteins and Their Substrate Specificities Using Primary Sequence Information". In: *PLoS ONE* 9.6 (2014). Ed. by D. Fotiadis, e100278. DOI: 10.1371/journal.pone.0100278. URL: <https://doi.org/10.1371/journal.pone.0100278>.
- [10] Liou, Y.-F. et al. "SCMMTP: identifying and characterizing membrane transport proteins using propensity scores of dipeptides". In: *BMC Genomics* 16.S12 (2015).
- [11] Li, L. et al. "Prediction the Substrate Specificities of Membrane Transport Proteins Based on Support Vector Machine and Hybrid Features". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 13.5 (2016), 947–953.
- [12] Nguyen, T.-T.-D. et al. "Using word embedding technique to efficiently represent protein sequences for identifying substrate specificities of transporters". In: *Analytical Biochemistry* 577 (2019), 73–81.
- [13] Alballa, M. and Butler, G. "TooT-T: discrimination of transport proteins from non-transport proteins". In: *BMC Bioinformatics* 21.S3 (2020). DOI: 10.1186/s12859-019-3311-6. URL: <https://doi.org/10.1186/s12859-019-3311-6>.

- [14] Alballa, M., Aplop, F., and Butler, G. “TranCEP: Predicting the substrate class of transmembrane transport proteins using compositional, evolutionary, and positional information”. In: *PLOS ONE* 15.1 (2020). Ed. by A. G. de Brevern, e0227683. DOI: 10.1371/journal.pone.0227683. URL: <https://doi.org/10.1371/journal.pone.0227683>.
- [15] Alley, E. C. et al. “Unified rational protein engineering with sequence-based deep representation learning”. In: *Nature Methods* 16.12 (2019), 1315–1322. DOI: 10.1038/s41592-019-0598-1. URL: <https://doi.org/10.1038/s41592-019-0598-1>.
- [16] Heinzinger, M. et al. “Modeling aspects of the language of life through transfer-learning protein sequences”. In: *BMC Bioinformatics* 20 (2019).
- [17] Anteghini, M., Santos, V. A. M. dos, and Saccenti, E. “In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins”. In: *bioRxiv* (2021). DOI: 10.1101/2021.01.18.427146.
- [18] Rives, A. et al. “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences”. In: *Proceedings of the National Academy of Sciences* 118.15 (2021).
- [19] Nambiar, A. et al. “Transforming the Language of Life: Transformer Neural Networks for Protein Prediction Tasks”. In: *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. BCB '20. Virtual Event, USA: Association for Computing Machinery, 2020. ISBN: 9781450379649. DOI: 10.1145/3388440.3412467. URL: <https://doi.org/10.1145/3388440.3412467>.
- [20] Elnaggar, A. et al. “ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Deep Learning and High Performance Computing”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 01 (2021), 1–1. DOI: 10.1109/TPAMI.2021.3095381.
- [21] Anteghini, M. et al. “OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection”. In: *Computational and Structural Biotechnology Journal* 21 (2022), 128–133. DOI: 10.1016/j.csbj.2022.11.058. URL: <https://doi.org/10.1016/j.csbj.2022.11.058>.
- [22] Vastermark, A. et al. “Expansion of the APC superfamily of secondary carriers”. In: *Proteins: Structure, Function, and Bioinformatics* 82.10 (2014), 2797–2811. DOI: 10.1002/prot.24643. URL: <https://doi.org/10.1002/prot.24643>.
- [23] Nigam, S. K. et al. “The Organic Anion Transporter (OAT) Family: A Systems Biology Perspective”. In: *Physiological Reviews* 95.1 (2015), 83–123. DOI: 10.1152/physrev.00025.2013. URL: <https://doi.org/10.1152/physrev.00025.2013>.
- [24] Consortium, T. U. “UniProt: the universal protein knowledgebase in 2021”. In: *Nucleic Acids Research* 49.D1 (2020), D480–D489.
- [25] Lyall, F. “Biochemistry”. In: *Basic Science in Obstetrics and Gynaecology*. Elsevier, 2010, 143–171. DOI: 10.1016/b978-0-443-10281-3.00013-0. URL: <https://doi.org/10.1016/b978-0-443-10281-3.00013-0>.

- 
- [26] Hata, Y., Slaughter, C. A., and Südhof, T. C. "Synaptic vesicle fusion complex contains unc-18 homologue bound to syntaxin". In: *Nature* 366.6453 (1993), 347–351.
  - [27] Huang, Y. et al. "Membrane Transporters and Channels". In: *Cancer Research* 64.12 (2004), 4294–4301. DOI: 10.1158/0008-5472.can-03-3884. URL: <https://doi.org/10.1158/0008-5472.can-03-3884>.
  - [28] Mueckler, M. et al. "Sequence and Structure of a Human Glucose Transporter". In: *Science* 229.4717 (1985), 941–945. DOI: 10.1126/science.3839598. URL: <https://doi.org/10.1126/science.3839598>.
  - [29] Ristovski, M. et al. "Lipid Transporters Beam Signals from Cell Membranes". In: *Membranes* 11.8 (2021), 562. DOI: 10.3390/membranes11080562. URL: <https://doi.org/10.3390/membranes11080562>.
  - [30] Ma, Z., Jacobsen, F. E., and Giedroc, D. P. "Coordination Chemistry of Bacterial Metal Transport and Sensing". In: *Chemical Reviews* 109.10 (2009), 4644–4681. DOI: 10.1021/cr900077w. URL: <https://doi.org/10.1021/cr900077w>.
  - [31] Agarwal, S. et al. "Identification of mannose interacting residues using local composition". In: *PloS one* 6.9 (2011), e24039.
  - [32] Chen, S.-A. et al. "Prediction of transporter targets using efficient RBF networks with PSSM profiles and biochemical properties". In: *Bioinformatics* 27.15 (2011), 2062–2067.
  - [33] Kawashima, S. et al. "AAindex: amino acid index database, progress report 2008". In: *Nucleic acids research* 36.suppl\_1 (2007), D202–D205.
  - [34] Attwood, T. "Profile (Position-Specific Scoring Matrix, Position Weight Matrix, PSSM, Weight Matrix)". In: *Dictionary of Bioinformatics and Computational Biology*. American Cancer Society, 2004. ISBN: 9780471650126.
  - [35] Stormo, G. D. et al. "Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli". In: *Nucleic Acids Research* 10.9 (1982), 2997–3011.
  - [36] Altschul, S. et al. "Gapped blast and psi-blast: A new generation of protein database search programs". In: *Nucl. Acids. Res.* 25 (1996), 3389–3402.
  - [37] Suzek, B. E. et al. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches". In: *Bioinformatics* 31.6 (2014), 926–932.
  - [38] Boughaci, D., Benhamou, B., and Drias, H. "IGA: an Improved Genetic Algorithm for MAX-SAT Problems". In: *Proceedings of the 3rd Indian International Conference on Artificial Intelligence, Pune, India, December 17-19, 2007*. Ed. by B. Prasad. IICAI, 2007, 132–150.
  - [39] Bradley, A. P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms". In: *Pattern Recognition* 30.7 (1997), 1145–1159. DOI: 10.1016/S0031-3203(96)00142-2. URL: [https://doi.org/10.1016/S0031-3203\(96\)00142-2](https://doi.org/10.1016/S0031-3203(96)00142-2).

- [40] Li, Z. R. et al. "PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence". In: *Nucleic Acids Research* 34.Web Server (2006), W32–W37. DOI: 10.1093/nar/gkl305. URL: <https://doi.org/10.1093/nar/gkl305>.
- [41] Guthrie, D. et al. "A Closer Look at Skip-gram Modelling". In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*. Genoa, Italy: European Language Resources Association (ELRA), 2006. URL: [http://www.lrec-conf.org/proceedings/lrec2006/pdf/357\\_pdf.pdf](http://www.lrec-conf.org/proceedings/lrec2006/pdf/357_pdf.pdf).
- [42] Mikolov, T. et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. DOI: 10.48550/ARXIV.1301.3781. URL: <https://arxiv.org/abs/1301.3781>.
- [43] Boser, B. E., Guyon, I. M., and Vapnik, V. N. "A Training Algorithm for Optimal Margin Classifiers". In: COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 1992, 144–152. ISBN: 089791497X. DOI: 10.1145/130385.130401. URL: <https://doi.org/10.1145/130385.130401>.
- [44] Altschul, S. F. et al. "Basic local alignment search tool". In: *Journal of Molecular Biology* 215.3 (1990), 403–410. DOI: 10.1016/S0022-2836(05)80360-2. URL: [https://doi.org/10.1016/S0022-2836\(05\)80360-2](https://doi.org/10.1016/S0022-2836(05)80360-2).
- [45] Tommaso, P. D. et al. "T-Coffee: a web server for the multiple sequence alignment of protein and RNA sequences using structural information and homology extension". In: *Nucleic Acids Research* 39.suppl (2011), W13–W17. DOI: 10.1093/nar/gkr245. URL: <https://doi.org/10.1093/nar/gkr245>.
- [46] Chang, J.-M., Tommaso, P. D., and Notredame, C. "TCS: A New Multiple Sequence Alignment Reliability Measure to Estimate Alignment Accuracy and Improve Phylogenetic Tree Reconstruction". In: *Molecular Biology and Evolution* 31.6 (2014), 1625–1637. DOI: 10.1093/molbev/msu117. URL: <https://doi.org/10.1093/molbev/msu117>.
- [47] Peters, M. E. et al. "Deep contextualized word representations". In: *Proc. of NAACL*. 2018.
- [48] Hochreiter, S. and Schmidhuber, J. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [49] Brandes, N. et al. "ProteinBERT: A universal deep-learning model of protein sequence and function". In: *bioRxiv* (2021). DOI: 10.1101/2021.05.24.445464.
- [50] Devlin, J. et al. "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)*. 2019, 4171–4186.
- [51] Ashburner, M. et al. "Gene Ontology: tool for the unification of biology". In: *Nature Genetics* 25.1 (2000), 25–29. DOI: 10.1038/75556. URL: <https://doi.org/10.1038/75556>.
- [52] Vaswani, A. et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, 5998–6008.
- [53] Harris, Z. S. "Distributional structure". In: *Word* 10.2-3 (1954), 146–162.



- 
- [54] Li, W. and Godzik, A. “Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences”. In: *Bioinformatics* 22.13 (2006), 1658–1659. doi: 10.1093/bioinformatics/btl158. URL: <https://doi.org/10.1093/bioinformatics/btl158>.
  - [55] Li, Y. and Ilie, L. “SPRINT: ultrafast protein-protein interaction prediction of the entire human interactome”. In: *BMC Bioinformatics* 18.1 (2017). doi: 10.1186/s12859-017-1871-x. URL: <https://doi.org/10.1186/s12859-017-1871-x>.
  - [56] Cristianini, N. and Ricci, E. “Support Vector Machines”. In: *Encyclopedia of Algorithms*. Boston, MA: Springer US, 2008, 928–932. ISBN: 978-0-387-30162-4. doi: 10.1007/978-0-387-30162-4\_415. URL: [https://doi.org/10.1007/978-0-387-30162-4\\_415](https://doi.org/10.1007/978-0-387-30162-4_415).
  - [57] Seliya, N., Zadeh, A. A., and Khoshgoftaar, T. M. “A literature review on one-class classification and its potential applications in big data”. In: *Journal of Big Data* 8.1 (2021). doi: 10.1186/s40537-021-00514-x. URL: <https://doi.org/10.1186/s40537-021-00514-x>.
  - [58] Tin Kam Ho. “The random subspace method for constructing decision forests”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), 832–844. doi: 10.1109/34.709601.
  - [59] Breiman, L. “Random forests”. In: *Machine learning* 45.1 (2001), 5–32.
  - [60] Murtagh, F. “Multilayer perceptrons for classification and regression”. In: *Neurocomputing* 2.5-6 (1991), 183–197. doi: 10.1016/0925-2312(91)90023-5. URL: [https://doi.org/10.1016/0925-2312\(91\)90023-5](https://doi.org/10.1016/0925-2312(91)90023-5).
  - [61] Linnainmaa, S. “Taylor expansion of the accumulated rounding error”. In: *BIT* 16.2 (1976), 146–160. doi: 10.1007/bf01931367. URL: <https://doi.org/10.1007/bf01931367>.
  - [62] Fukushima, K. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (1980), 193–202. doi: 10.1007/bf00344251. URL: <https://doi.org/10.1007/bf00344251>.
  - [63] Tolles, J. and Meurer, W. J. “Logistic Regression”. In: *JAMA* 316.5 (2016), 533. doi: 10.1001/jama.2016.7653. URL: <https://doi.org/10.1001/jama.2016.7653>.
  - [64] Cramer, J. “The Origins of Logistic Regression”. In: *Tinbergen Institute, Tinbergen Institute Discussion Papers* (2002). doi: 10.2139/ssrn.360300.
  - [65] Harris, C. R. et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), 357–362. doi: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
  - [66] Stone, M. “Cross-Validatory Choice and Assessment of Statistical Predictions”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2 (1974), 111–133. doi: <https://doi.org/10.1111/j.2517-6161.1974.tb00994.x>. URL: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x>.
  - [67] Pedregosa, F. et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), 2825–2830.

- [68] Rijsbergen, C. J. V. *Information Retrieval*. 2nd. Butterworth-Heinemann, 1979.
- [69] Matthews, B. W. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), 442–451.
- [70] Boughorbel, S., Jarray, F., and El-Anbari, M. "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric". In: *PloS one* 12.6 (2017), e0177678.
- [71] Melo, F. "Area under the ROC Curve". In: *Encyclopedia of Systems Biology*. Springer New York, 2013, 38–39. DOI: 10.1007/978-1-4419-9863-7\_209. URL: [https://doi.org/10.1007/978-1-4419-9863-7\\_209](https://doi.org/10.1007/978-1-4419-9863-7_209).
- [72] Saccenti, E., Hendriks, M. H. W. B., and Smilde, A. K. "Corruption of the Pearson correlation coefficient by measurement error and its estimation, bias, and correction under different error models". In: *Scientific Reports* 10.1 (2020). DOI: 10.1038/s41598-019-57247-4. URL: <https://doi.org/10.1038/s41598-019-57247-4>.





---

# Representing semantified biological assays in the Open Research Knowledge Graph

This chapter is based on:

Marco Anteghini, Jennifer D'Souza, Vitor A.P. Martins dos Santos, Sören Auer. Representing Semantified Biological Assays in the Open Research Knowledge Graph. *Published in: Ishita, E., Pang, N.L.S., Zhou, L. (eds) Digital Libraries at Times of Massive Societal Transition. ICADL 2020. Lecture Notes in Computer Science, 12504 (2020).* DOI: 10.1007/978-3-030-64452-9\_8

## Abstract

In the biotechnology and biomedical domains, recent text mining efforts advocate for machine-interpretable, and preferably, semantified, documentation formats of laboratory processes. This includes wet-lab protocols, (in)organic materials synthesis reactions, genetic manipulations and procedures for faster computer-mediated analysis and predictions. Herein, we present our work on *the representation of semantified bioassays in the Open Research Knowledge Graph (ORKG)*. In particular, we describe a semantification system work-in-progress to generate, automatically and quickly, the critical semantified bioassay data mass needed to foster a consistent user audience to adopt the ORKG for recording their bioassays and facilitate the organisation of research, according to FAIR principles.

## 8.1 Introduction

More and more scholarly digital library initiatives aim at fostering the digitalization of traditional document-based scholarly articles [1–8]. This means structuring and organizing, in a fine-grained manner, knowledge elements from previously unstructured scholarly articles in a Knowledge Graph. These efforts are analogous to the digital transformation seen in recent years in other information-rich publishing and communication services, e.g., e-commerce product catalogs instead of mailorder catalogs, or online map services instead of printed street maps. For these services, the traditional document-based publication was not just digitized (by making digitized PDFs of the analog artifacts available) but has seen a comprehensively transformative digitalization.

Of available scholarly knowledge digitalization avenues [1–7], we highlight the Open Research Knowledge Graph (ORKG) [9]. It is a next-generation digital library (DL) that focuses on ingesting information in scholarly articles as machine-actionable knowledge graphs (KG). In it, an article is represented with both (bibliographic) metadata and semantic descriptions (as subject-predicate-object triples) of its *contributions*. ORKG has a number of advantages as: 1) it enables flexible semantic content modeling (i.e., ontologized or not, depending on the user or domain); 2) it semantifies *contributions* at various levels of granularity from shallow to fine-grained; and 3) it publishes persistent KG links per article contribution that it contains. For further technical details about the platform, we refer the reader to the introductory paper [9].

The ORKG DL aims to integrate and interlink contributions' KGs for Science at large, i.e. multidisciplinary. Thus far, ongoing efforts are in place for integrating scholarly contributions from at least two disciplines, viz. Math [10] (e.g., <https://www.orkg.org/orkg/paper/R12192>) and the Natural Language Processing subdomain in AI [11] (e.g., <https://www.orkg.org/orkg/paper/R44253>). Moreover, the ORKG also has a separate feature to automatically import individual articles' contributions data found tabulated in survey articles [12]. E.g., an ORKG object for Earth Science articles' contributions surveyed: <https://www.orkg.org/orkg/comparison/R38484>. Since surveys are written in most disciplines, this latter feature directly targets the ORKG aim; however, its sole limitation is that it is restricted only to those papers that have been

---

surveyed. On the other hand, with the per-domain semantification models, articles not surveyed can be also modeled in the ORKG.

In this paper, we describe our ongoing work in extending the ORKG to integrate biological assays from the Biochemistry discipline. For bioassays, a semantification model already exists as the BioAssay Ontology (BAO) [13]. However, we need to design a pragmatic workflow for integrating bioassays semantified by the BAO in the ORKG DL. To this end, we discuss the manual and automatic process of integrating such semantified data in the ORKG DL. Furthermore, we show how these semantified data integrated in the ORKG is amenable to advanced computational processing support for the researcher.

With the volume of research burgeoning [14], adopting a finer-grained semantification as KG for scholarly content representation is compelling. Better semantification means better machine actionability, which in turn means innumerable possibilities of advanced computational functions on scholarly content. One function especially poignant in this era of the publications deluge [15], is computational support to alleviate the manual information ingestion cognitive burden. This is precisely the computational support showcase we depict from the ORKG DL over our integrated bioassay KGs, consequently highlighting the benefits of digitalizing bioassays and of the ORKG DL platform.

## 8.2 Our Work-In-Progress Aims and Motivations

*Allowing practitioners to easily search for similar bioassays as well as compare these semantically structured bioassays on their key properties.*

**Why integrate bioassays in a knowledge graph?** Until their recent semantification in an expert-annotated dataset of 983 bioassays [16–18] based on the BAO [13], bioassays were published in the form of plain text. Integrating their semantified counterpart in a KG facilitates their advanced computational processing. Consider that key assay concepts related to biological screening, including Perturbagen, Participants, Meta Target and Detection Technology, will be machine-actionable. This widens the potential for relational enrichment and interlinking when integrated with machine-interpretable formats of wet lab protocols and inorganic materials synthesis reactions and procedures [19–22]. Furthermore, in this era of neural-based ML technologies, KG-based word embeddings foster new inferential discovery mechanisms given that they encode high-dimensional semantic spaces [23] with bioassay KGs so far untested for.

**Why the ORKG DL ?** [2] The core of the setup of knowledge-based digitalized information flows is the distributed, decentralized, collaborative creation and evolution of information models. Moreover, vocabularies, ontologies, and knowledge graphs to establish a common understanding of the data between the various stakeholders. And, importantly, the integration of these technologies into the infrastructure and processes of search and knowledge exchange toward a research library of the future. The ORKG DL is such a solution. Implemented within TIB, as a central

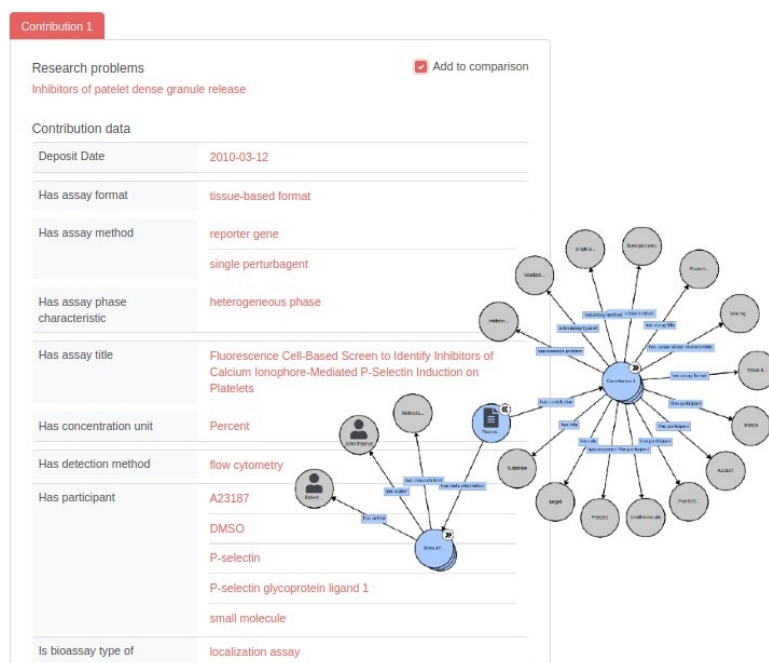


Figure 8.1: An ORKG representation of a semantified Bioassay with an overlaid graph view of the assay. Accessible at: <https://www.orkg.org/orkg/paper/R48178>

library and information centre for science and technology, it also promises development longevity: the Leibniz Association institutional networks presents a critical mass of application domains and users to enhance the infrastructure and continuously integrate new knowledge disciplines.

With these considerations in place, the work described in the subsequent sections is being carried forth. Next, we describe our approach in the context of two main research questions.

### 8.3 Approach: Digitalization of Biological Assays

**RQ1:** *What are steps for manually digitalizing a Bioassay in the ORKG?* The digitalization is based on the prior requirement that text-based bioassays are semantified based on the BioAssay Ontology (BAO) [13]. This is the manual aspect of the digitalization process involving domain experts or the assay authors themselves. In Figure 8.1, we show an example of a manually pre-semantified bioassay integrated in ORKG. This bioassay was semantified on eight properties based on the BAO. It was drawn from an expert-annotated set of 983 bioassays [17, 18]. In terms of salient features, the bioassays in this dataset have 53 triple semantic statements on average with a minimum of 5 and a maximum of 92 statements; there are 42 different types of bioassays (e.g., luciferase reporter gene assay, protein-protein interaction assay—see in appendix the full list); and there are 11 assay formats (e.g., cell-based,



---

biochemical). Thus, the manual semantification task complexity can be viewed as 53 modeling decisions.

In gist, the manual digitalization of a bioassay in the ORKG includes: 1) *a BAO-based semantification step*: forming subject-predicate-object triples of the bioassay text content based on the BAO. E.g., for the assay in Fig. 8.1, a few of its semantic triples are: (Contribution, Has assay format, tissue-based format), (Contribution, Has assay method, reporter gene), among others. And as a recommended step, 2) associating each ontologized resource (i.e., a subject, a predicate, an object) with a URI as its defining class in the original ontology, which for bioassays is the BAO.

Having just described the manual digitalization workflow, we next present our hybrid workflow that is currently in development. In this, we decide to incorporate automated semantification which levies pragmatic considerations in the digitalization of bioassays in the ORKG. Relatedly, there is an existing hybrid system [16] for semantifying bioassays involving machine learning and expert interaction which inspires our work. Nonetheless, we differ. While their learning-based component relies heavily on explicitly encoded syntactic features of the text, ours relies on neural networks based on the current state-of-the-art transformer models [24] trained on millions of scientific articles [25]. Such systems by encoding high-dimensional semantic spaces of the underlying text, obviate the need to make explicit considerations for features of the text. Moreover, they significantly outperform systems designed based on explicit features [26]—with due credit to the system by Clark et al. [16] designed prior to the onset of this revolutionary technology. Next, our hybrid workflow is designed toward a practical end—to be integrated in the ORKG DL which has a predominant focus on the digitalization of scholarly knowledge content multidisciplinarily, thus setting it apart from any existing DL.

**RQ2:** *What are the modules needed in the hybrid digitalization of Bioassays in the ORKG?* Essentially, given a new bioassay text input, we are implementing two modules in a two-step workflow as follows: 1) an automated semantifier; and 2) a human-in-the-loop curation of the predicted labels either by the assay author or a dedicated curator. Unlike the manual workflow, this presents a much easier and less time-intensive task for the human. They would be merely selecting the correctly predicted triples, deleting the incorrect ones, or defining new ones as needed. Assuming a well-trained machine learning module, the latter two steps may be entirely omitted. Toward this hybrid workflow, as work in progress, the automated semantifier is in development, and we are also implementing extensions in the ORKG infrastructure to include additional front-end views as assay curation interfaces.

## 8.4 Solving the Cognitive Information Ingestion Hurdle: Comparison Surveys across KG-based Bioassays

*Premise: We need an information processing tool that can be used by biomedical practitioners to quickly comprehend bioassays' key properties.*

The ORKG DL has a computational feature to generate and publish surveys in the form of a tabulated comparisons of the KG nodes [12]. To demonstrate this

Properties	SAR analysis of tumor necrosis factor alpha (TNF-alpha) induced IL-8 secretion in MCF-7/NOD1 cells Contribution 1	Fluorescence Cell-Based Screen to Identify Inhibitors of Calcium Ionophore-Mediated P-Selectin Induction on Platelets Contribution 1	Allosteric Modulators of D1 Receptors: Dose-dependent Assay Contribution 1
Deposit date	2010-01-13	2010-03-12	2007-11-19
Has assay format	Interleukin-8 cell-based format	tissue-based format	cell-based format
Has assay method	ELISA binding assessment method single perturbagent stable transfection	reporter gene single perturbagent	luciferase induction reporter gene method stable transfection
Has assay phase characteristic	heterogeneous phase	heterogeneous phase	heterogeneous phase
Has assay title	SAR analysis of tumor necrosis factor alpha (TNF-alpha) induced IL-8 secretion in MCF-7/NOD1 cells	Fluorescence Cell-Based Screen to Identify Inhibitors of Calcium Ionophore-Mediated P-Selectin Induction on Platelets	Allosteric modulators of D1 Dopamine Receptors
Has concentration unit	micromolar	Percent	micromolar molar
Has detection method	absorbance	flow cytometry	luminescence induction
Has endpoint	IC50	Empty	EC95
Has incubation time value	30 min	20 min	2.5 h
Has measured entity	TNF-alpha measured entity	measured entity	Allosteric modulators of D1 Dopamine Receptors Dopamine

Figure 8.2: Comparisons of semantified bioassays in the ORKG digital library. Online <https://www.orkg.org/orkg/comparison?contributions=R48195,R48179,R48147>

feature, we manually entered the data of three semantified bioassays in the ORKG DL. Applying then the ORKG survey feature on the three assays aggregates their semantified graph nodes in tabulated comparisons across the assays. This is depicted in Figure 8.2. With such structured computations enabled, we have a novel approach to uncovering and presenting information relying on aggregated scholarly knowledge. The computation shown in Fig. 8.2 aligns closely with the notion of the traditional survey articles, except it is fully automated and operates on machine-actionable knowledge elements. The BAO-semantified assays are compared side-by-side on their graph nodes. Thus, tracking the progress on bioassays, can be eased from a task of several days to a few minutes.

---

## 8.5 Conclusion

Thus in this paper, we outlined a vision in two separate workflows for integrating bioassay knowledge in the ORKG DL and our ongoing work to this end. The implications of bioassay structured and machine-actionable knowledge are broad.

To mention just one in the particular context of the current Covid-19 pandemic: The discovery of cures for diseases can be greatly expedited if scientists are given intelligent information access tools, and our work toward automatically semantifying bioassays are a step in this direction.

To this end, the workflows prescribed in this work offer the possibilities to chose between a manual or a semi-automatic strategy for bioassays' semantification within a real-world digital library.

We would like to invite interested researchers to collaborate with us on the following topics: 1) generating a large dataset of semantically structured bioassays; 2) user evaluation of our semi-automated system for semantically structuring bioassay data.

We deem this as a starting point for a discussion in the community ultimately leading to more clearly defined technical requirements, and a roadmap for fulfilling the potential of the ORKG as a next-generation digital library for fine-grained semantified access to scholarly content.

## Aknowledgement

Supported by TIB Leibniz Information Centre for Science and Technology, the EU H2020 ERC project ScienceGraph (GA ID: 819536) and the ITN PERICO (GA ID: 812968). This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Aryani, A. et al. “A Research Graph dataset for connecting research data repositories using RD-Switchboard”. In: *Scientific data* 5 (2018), 180099.
- [2] Auer, S. *Towards an Open Research Knowledge Graph*. Version 1. 2018. doi: 10.5281/zenodo.1157185.
- [3] Baas, J. et al. “Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies”. In: *Quantitative Science Studies* 1.1 (2020), 377–386.
- [4] Birkle, C. et al. “Web of Science as a data source for research on scientific and scholarly activity”. In: *Quantitative Science Studies* 1.1 (2020), 363–376.
- [5] Fricke, S. “Semantic scholar”. In: *Journal of the Medical Library Association: JMLA* 106.1 (2018), 145.
- [6] Hendricks, G. et al. “Crossref: The sustainable source of community-owned scholarly metadata”. In: *Quantitative Science Studies* 1.1 (2020), 414–427.
- [7] Manghi, P. et al. *OpenAIRE Research Graph Dump*. Version 1.0.0-beta. Zenodo, 2019. doi: 10.5281/zenodo.3516918. URL: <https://doi.org/10.5281/zenodo.3516918>.
- [8] Wang, K. et al. “Microsoft academic graph: When experts are not enough”. In: *Quantitative Science Studies* 1.1 (2020), 396–413.
- [9] Jaradeh, M. Y. et al. “Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge”. In: *Proceedings of the 10th International Conference on Knowledge Capture*. 2019, 243–246.
- [10] Runnwerth, M., Stocker, M., and Auer, S. “Operational Research Literature as a Use Case for the Open Research Knowledge Graph”. In: *Mathematical Software - ICMS 2020 - 7th International Conference, Braunschweig, Germany, July 13-16, 2020, Proceedings*. Ed. by A. M. Bigatti et al. Vol. 12097. Lecture Notes in Computer Science. Springer, 2020, 327–334. doi: 10.1007/978-3-030-52200-1\_32.
- [11] D’Souza, J. and Auer, S. *NLPContributions: An Annotation Scheme for Machine Reading of Scholarly Contributions in Natural Language Processing Literature*. 2020.
- [12] Oelen, A. et al. “Generate FAIR Literature Surveys with Scholarly Knowledge Graphs”. In: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020. JCDL ’20. Virtual Event, China: Association for Computing Machinery*, 2020, 97–106. ISBN: 9781450375856. doi: 10.1145/3383583.3398520.
- [13] Visser, U. et al. “BioAssay Ontology (BAO): a semantic description of bioassays and high-throughput screening results”. In: *BMC bioinformatics* 12.1 (2011), 257.
- [14] Johnson, R., Watkinson, A., and Mabe, M. “The STM report”. In: *An overview of scientific and scholarly publishing. 5th edition October* (2018).
- [15] Jinha, A. E. “Article 50 million: an estimate of the number of scholarly articles in existence”. In: *Learned Publishing* 23.3 (2010), 258–263.

- 
- [16] Clark, A. M. et al. “Fast and accurate semantic annotation of bioassays exploiting a hybrid of machine learning and user confirmation”. In: *PeerJ* 2 (2014), e524.
  - [17] Schürer, S. C. et al. “BioAssay ontology annotations facilitate cross-analysis of diverse high-throughput screening data sets”. In: *Journal of biomolecular screening* 16.4 (2011), 415–426.
  - [18] Vempati, U. D. et al. “Formalization, annotation and analysis of diverse drug and probe screening assay datasets using the BioAssay Ontology (BAO)”. In: *PloS one* 7.11 (2012), e49198.
  - [19] Kononova, O. et al. “Text-mined dataset of inorganic materials synthesis recipes”. In: *Scientific data* 6.1 (2019), 1–11.
  - [20] Kulkarni, C. et al. “An Annotated Corpus for Machine Reading of Instructions in Wet Lab Protocols”. In: *NAACL: HLT, Volume 2 (Short Papers)*. New Orleans, Louisiana, 2018, 97–106. DOI: 10.18653/v1/N18-2016.
  - [21] Kuniyoshi, F. et al. “Annotating and Extracting Synthesis Process of All-Solid-State Batteries from Scientific Literature”. In: *LREC*. 2020, 1941–1950.
  - [22] Mysore, S. et al. “The Materials Science Procedural Text Corpus: Annotating Materials Synthesis Procedures with Shallow Semantic Structures”. In: *Proceedings of the 13th Linguistic Annotation Workshop*. 2019, 56–64.
  - [23] Bianchi, F. et al. “Knowledge Graph Embeddings and Explainable AI”. In: *arXiv preprint arXiv:2004.14843* (2020).
  - [24] Vaswani, A. et al. “Attention is all you need”. In: *Advances in neural information processing systems*. 2017, 5998–6008.
  - [25] Beltagy, I., Lo, K., and Cohan, A. “SciBERT: Pretrained Language Model for Scientific Text”. In: *EMNLP*. 2019.
  - [26] Devlin, J. et al. “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)*. 2019, 4171–4186.



---

# Easy semantification of bioassays

This chapter is based on:

Marco Anteghini, Jennifer D'Souza, Vitor A.P. Martins dos Santos, Sören Auer. Easy Semantification of Bioassays.

*Published in: Bandini, S., Gasparini, F., Mascardi, V., Palmonari, M., Vizzari, G. (eds) AIXIA 2021 – Advances in Artificial Intelligence. AIXIA 2021. Lecture Notes in Computer Science, 13196 (2021)*

DOI: 10.1007/978-3-031-08421-8\_14

## Abstract

Biological data and knowledge bases increasingly rely on Semantic Web technologies and the use of knowledge graphs for data integration, retrieval and federated queries. We propose a solution for automatically *semantifying biological assays*. Our solution contrasts the problem of automated semantification as labeling versus clustering where the two methods are on opposite ends of the method complexity spectrum. Characteristically modeling our problem, we find the clustering solution significantly outperforms a deep neural network state-of-the-art labeling approach. This novel contribution is based on two factors: 1) a learning objective closely modeled after the data outperforms an alternative approach with sophisticated semantic modeling; 2) automatically *semantifying biological assays* achieves a high performance  $F1$  of nearly 83%, which to our knowledge is the first reported standardized evaluation of the task offering a strong benchmark model.

## 9.1 Introduction

Semantifying scholarly communication within the next-generation Knowledge-Graph-based Scholarly Digital Libraries, such as the Open Research Knowledge Graph<sup>1</sup> (ORKG) [1], relies on core semantic techniques such as ontologized formalizations and Web resource identifiers [2]. This supports the mainstream *Knowledge representation and reasoning* vision in AI. Further, semantified data can enable knowledge-based interoperability between multiple databases simply by reusing identifiers and utilizing no-SQL query languages such as SPARQL [3] that can perform distributed queries over the various data sources. Obtaining improved machine interpretability of scientific findings has seen keen interest in the Life Sciences [4] domain. Many major bioinformatics databases such as UniProt [5], KEGG [6], REACTOME [7] and the NCBI database [8] which includes the PubChem BioAssay database now make their data available as Linked Data in which both biological entities and connections between them are ontologized with standardized relations and are identified through a unique identifier (an Internationalized Resource Identifier or IRI). In a parallel Computational Linguistics ecosphere, many recent interdisciplinary data collection and annotation efforts [9–12] are focused on the shallow semantic structuring of unstructured text based on the Life Sciences ontologies. E.g., instructional content in lab protocols, descriptions of chemical synthesis reactions, or bioassays. Thus information described otherwise in *ad hoc* ways within scholarly documents attain machine-actionable, structured representations. Such datasets inadvertently facilitate the development of automated machine readers.

In this work, we take up the problem of the automated semantification of Biological Assays (Bioassays). This problem has both Life Science-specific solutions as the Bioassay Ontology [13] and Computational Linguistics-based semantified unstructured text annotations [14–16]. A bioassay is, by definition, a standard biochemical test procedure used to determine the concentration or potency of a stimulus (physical, chemical, or biological) by its effect on living cells or tissues [17, 18]. It is

---

<sup>1</sup><https://www.orkg.org/>



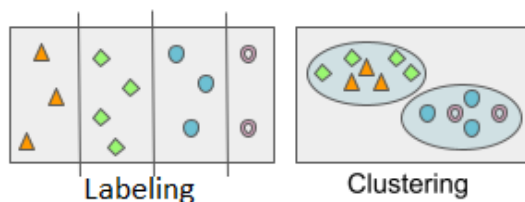


Figure 9.1: Illustration of labeling versus clustering to aggregate data points

described with relevant information on basic procedures such as determining the signal that indicates biological activity, determining doses used during the test, calculation methods etc. Also, bioassays are always qualified and validated [19] to highlight their accuracy, repeatability, and adequacy for use in the measurement of relative potency. Thus, a semantic description of the assay represented as logical annotations consisting of property and value pairs is the semantic equivalent of the unstructured bioassay text. They would enable their large-scale analysis in diverse systems. Bioassay texts are semantified based on the BioAssay Ontology (BAO) [13, 20]. The BAO describes chemical and biological screening assays and their related results to facilitate their categorization and data analysis. On the BioPortal<sup>2</sup> where the BAO is hosted, the BAO showed 7513 classes and 227 properties dated June 3, 2021. Thus the semantification of an assay is a tedious human annotation task since they have to: 1) decide which ontologized class relation pair applies to a bioassay; and 2) given a sentence from the bioassay text, decide whether it is expressible as a logical statement by the BAO. This results in a large decision space for the human annotator making it a time-consuming endeavor. Computational techniques fitted appropriately with the problem semantics can fully alleviate the tedious human annotation task.

In this paper, we examine the computational aspects of the automated semantification of biological assays (bioassays) in light of two different approaches and their evaluations. We first formulate a labeling objective for bioassay semantification. This we recently proposed as a work-in-progress idea leveraging a transformer-based supervised classifier [21, 22]. Herein, we carry out in detail the experiments we began and further examine a novel clustering objective to bioassays semantification. Labeling and clustering are two methods of pattern identification used in machine learning. Although both techniques have certain similarities, the difference lies in the fact that labeling relies on a predefined set of labels assigned to objects, while clustering identifies similarities between objects, which it groups according to those characteristics in common and which differentiate them from other groups of objects. This is illustrated in Figure 9.1. On the one hand, we identify each logical statement of a semantified bioassay as a potential label. On the other hand, we observed that bioassays with similar text descriptions also had similar semantic representations. Thus a fine-grained clustering of the assays themselves could mean a cluster as a whole can be semantified by a standard set of labels. If it takes a classifier multiple passes to fully label an assay, it takes a clustering model just one pass over the data to semantify clusters. Via our experiments, we observed that labeling

<sup>2</sup><https://bioportal.bioontology.org/>

and clustering have contrasting score and time footprints. As a surprising result, the powerful transformer-based labeling method proves to be less accurate than a clustering solution at 54% F1 vs. 83% F1; and labeling with a large labels set has a significantly longer prediction time accounting for per-label classifications.

In summary, the contributions of our work are:

1. we formalize two machine learning objectives, i.e. labels classification and clustering, for the automated semantification of bioassays. Relatedly, we discuss the dataset characteristics and its adaptations. To our knowledge, these standardized machine learning tasks over a corpus of bioassays are discussed for the first time.
2. we empirically evaluate the approaches and report unconventional findings that favor k-means clustering over the more resource-intensive transformers;
3. we present an application of bioassay semantification within the Open Research Knowledge Graph scholarly contributions knowledge digitalization platform. The workflow allows scientists to upload bioassays, obtain automated semantified bioassays as results, and curate the semantic annotations.

## 9.2 A Motivating Example for Bioassay Semantification

**Assay ID 1960** An example sentence from the assay is ‘Finally, fluorescence polarization can be used to effectively monitor the in vitro RNA-binding activity of both proteins using a standard fluorescence plate reader.’ This sentence as it is is not computable. In other words, the terms ‘fluorescence polarization’, ‘in vitro RNA-binding activity’ or ‘standard fluorescence plate reader’ in the unstructured text have no semantic interpretation to a computer. However, in the context of the standardized Bioassay terminology, the sentence is annotated with the following logical statement: ‘has detection method’  $\rightarrow$  ‘fluorescence polarization’ from the BioAssay ontology [13] grounded to the identifiers (bao:BAO\_0000207, bao:BAO\_0000003). This semantic annotation is now computable by machines, e.g., within reasoning tasks. But these annotations need to be manually curated by an expert who reads from context information in the phrase ‘Finally, fluorescence polarization can be used to effectively monitor’ and is also familiar with the experimental setting of the assay. To semantify the above statement, the expert deduces that ‘high polarization’ in ‘protein-probe complex’ was detected by the method ‘fluorescence polarization.’ However, making such decisions is an expensive human annotation task and nearly impossible at scale. Nevertheless, if such logical statements are annotated for a small set of bioassays, they can be easily annotated at scale via machine learning which is the focus of this work. Another motivating example is available in Supplementary Materials (SM)<sup>3</sup>.

---

<sup>3</sup><https://github.com/MarcoAnteghini/Easy-Semantification-of-Bioassays-SM>

---

## 9.3 Related Work

### 9.3.1 Corpora of Semantified Life Science Publications

Increasingly, text mining initiatives are seeking out recipes or formulaic semantic patterns to automatically mine machine-actionable information from scholarly articles [9–12]. In [10], they annotate wet lab protocols, covering a large spectrum of experimental biology, including neurology, epigenetics, metabolomics, cancer and stem cell biology, with actions corresponding to lab procedures and their attributes including materials, instruments and devices used to perform specific actions. Thereby the protocols then constituted a prespecified machine-readable format as opposed to the ad hoc documentation norm. Kulkarni et al. [10] release a large human-annotated corpus of semantified wet lab protocols to facilitate machine learning of such shallow semantic parsing over natural language instructions. Within scholarly articles, such instructions are typically published in the Materials and Method section in Biology and Chemistry fields. Along similar lines, inorganic materials synthesis reactions and procedures continue to reside as natural language descriptions in the text of journal articles. There is a growing need in such fields to find ways to systematically reduce the time and effort required to synthesize novel materials that presently remains one of the grand challenges in the field. In [9, 11], to facilitate machine learning models for automatic extraction of materials syntheses from text, they present datasets of synthesis procedures annotated with semantic structure by domain experts in Materials Science. The types of information captured include synthesis operations (i.e. predicates), and the materials, conditions, apparatus and other entities participating in each synthesis step.

In this work, we leverage a similar semantically annotated corpus in the Life Science domain, but the knowledge theme tackled in our corpus is that of semantifying bioassays [13]. Normally, bioassays can be stored and accessed on PubChem [23, 24] which now contains more than 1.3M bioassays (22-06-2021). Only considering the period between 2015 and 2021, 389,835 new bioassays have been added to PubChem. To semantify a single bioassay is expert-specific and time-consuming. However, the process is not scalable for large-scale analyses, e.g. searching databases for related assays and comparisons or clustering similar entries. This requires the creation of new approaches to favor bioassays semantification, analysis, comparison and facilitate knowledge sharing. The ultimate goal would be to obtain a fully-automated software that can easily transform a human-friendly unstructured bioassay text report to a computer-friendly version as their semantic equivalent in the form of a set of logical statements.

### 9.3.2 AI-based Scholarly Knowledge Graph Construction

Early scholarly knowledge graph (SKG) construction initiatives were based on the sentences' information granularity. For this, ontologies and vocabularies were created [25–28] from diverse aspects of the publication including discourse and specific themes as experiments; corpora were annotated [29, 30], and symbolic features-based ML techniques were implemented [31]. Recent scientific search technology led to new annotated corpora focusing on phrases with three or six types of generic sci-

Table 9.1: Four example logical statements (from 50 total) for the semantified PubChem Assay with ID 360 (<https://pubchem.ncbi.nlm.nih.gov/bioassay/360>). Note, these statements are triples with subject ‘Bioassay.’

---

HAS PARTICIPANT	→	DMSO
HAS ASSAY PHASE CHARACTERISTIC	→	HOMOGENEOUS PHASE
HAS TEMPERATURE VALUE	→	25 DEGREE CELSIUS
HAS INCUBATION TIME VALUE	→	20 MINUTE

---

entific concepts in articles across up to ten different scholarly disciplines [32–34], for which neural systems were developed [35–37]. In SKG creation, relation extraction has also raised keen interest, thanks also to community challenges such as ScienceIE 2017 [38], SemEval 2018 Task 7 [39] and NlpContributionGraph 2021 [40], where participants tackled the problem of detecting semantic relations; newer advanced methods employed attention-based bidirectional long short-term memory networks (BiLSTM) [41] or used dynamic span graph framework based on BiLSTMs [42]. Recently, strategically designed neural-symbolic hybrid approaches have proven effective [43].

For the scholarly knowledge theme of structuring Bioassays, specifically, the only prior machine learning approach was a morpho-syntactic features-based Bayes classifier [16]. This early system, however, had unreplicable human-engineered aspects and non-standard evaluations. We focus on machine learning that entails no additional hand-engineering and report standardized evaluations.

## 9.4 Materials and Methods

### 9.4.1 An Expert-Annotated Semantified Bioassays Corpus

To develop our automated semantifiers, we leverage a corpus comprising an expert-annotated collection of 983 semantified bioassays [14, 15]. In Table 9.1, we show four logical statements of a semantified bioassay (ID 360 in PubChem) as an example. Each logical statement is expressed as a predicate and value pair. In the chosen example, the first two statements are *ontologized* statements, i.e. the predicate and value pair are in the Bioassay Ontology (BAO) [13]. These annotations are made by a domain expert based on comprehensive knowledge of the BAO which contains thousands of predicate value pairs as semantification candidates. The next two statements are *partially ontologized*, i.e. their predicates can be found in the BAO but the values are directly from the bioassay text description and hence are bioassay-specific. These statements report the various specific measurements made in the course of the bioassay. Semantified Bioassays contain both *ontologized* and *partially ontologized* statements. For the semantification task addressed in this paper, we restrict ourselves only to the *ontologized* statements of each semantified bioassay. For this, we prune all *partially ontologized* statements from each semantified assay. In Table 9.2, we summarize the dataset statistics for the original corpus with all the statements and the corpus we use after pruning. We can see that prior to pruning, the original

Table 9.2: Semantified bioassays corpus statistics shown before (‘original’ row) and after (‘pruned’ row) pruning its *partially ontologized* statements. Note the corpus used for the work in this paper is the ‘pruned’ version.

	AVERAGE	MINIMUM	MAXIMUM	TOTAL
original	56	7	162	5524
pruned	37	2	87	1906

Table 9.3: Fine-grained pruned semantified corpus statistics in terms of the top 10, 20, 30, etc., most common predicates seen in the statements. E.g., the top 10 column contains the ten most frequently occurring predicates in the 1906 statements. Note the last column (‘top 80’) reflects the total unique predicates in the corpus. The rows show the number of the unique statements with the corresponding frequent predicates. The parenthesized numbers show the statements’ proportion in the overall corpus.

top 10	top 20	top 30	top 40	top 50	top 60	top 70	top 80
795	959	1492	1804	1866	1879	1896	1906
(41.7)	(50.3)	(78.3)	(94.6)	(97.9)	(98.6)	(99.5)	(100.0)

corpus had 5524 total unique statements overall, which after pruning are reduced to 1906 statements. In the pruned corpus, bioassays have between 2 minimum and 87 maximum statements at an average of 37 statements. Considering only the predicates in these 1906 total statements, some predicates apply to semantify a bioassay more commonly than others. This is shown via the predicates statistics reported in Table 9.3. In particular, 94% of the semantic statements comprise only the 40 most commonly occurring predicates from a total of 80 unique predicates. Note this labels repetition detail of the corpus is critical since the labeling of bioassays with semantic statements are only among those observed in the data. In our previous works [21, 22] we adopted different pruning strategies. A comparison with this final version is available in SM.

### Corpus Formalization.

Let  $B$  be the overall semantified bioassays collection. A bioassay  $b$  from  $B$  is semantified with a set of *ontologized* logical statements  $sls$  (or semantic statements) which is  $sls = \{ls_1, ls_2, ls_3, \dots, ls_k\}$  where  $ls_x$  is a logical statement  $\in LS$  such that  $LS$  is the collection of all the distinct *ontologized* logical statements used for semantification seen in the training data. And  $sls$  has  $k$  different statements when taken together form the semantic equivalent of bioassay  $b$ . Across bioassays, their corresponding  $sls$  sizes vary.

As shown in Table 9.2, the corpus we use has  $|LS| = 1906$  unique statements (after pruning the *partially ontologized* statements).

Two semantification machine learning objectives are contrasted next.

### 9.4.2 Labeling - Task Definition for Bioassay Semantification

Bioassays semantification can be addressed as a labeling problem. In this scenario, each logical statement can be treated within a binary classification task as applicable or not. On average in our data, a bioassay could then have around 37 applicable logical statements from  $LS$ . The task can be formalized as follows.

#### Task Formalism.

Each input data instance is the pair  $(b, ls; c)$  where  $c \in \{true, false\}$  is the classification of the label  $ls$ . Thus, specifically, our semantification problem entails classifying labels:  $(b, ls)$  is *true* if  $ls \in$  logical statements set of  $b$ , else *false*. The *false* instances are formed by pairing  $b$  with any other label not in the logical statements set  $sls$  of  $b$ .

Intuitively, this task formulation is meaningful because it emulates the way the human expert annotates the data. Basically, the expert, from their memory of all logical statements  $LS$ , simply assigns  $ls$  to a given  $b$  if they deem it as *true*; irrelevant statements are not considered, thus implicitly deemed *false*.

#### Task Model.

Our machine learning system is the state-of-the-art, bidirectional transformer-based SciBERT [44], pre-trained on millions of scientific articles. For bioassay semantification, we use the SciBERT classification architecture. In each data instance  $(b, ls; c)$ , the classifier input representation for the pair ' $b, ls$ ' is the standard SciBERT format, treating them as sentence pairs separated by the special [SEP] token; the special classification token ([CLS]) remains the first token of every instance. Its final hidden state is used as the aggregate sequence representation for classification and is fed into a linear classification layer.

### 9.4.3 Clustering - Task Definition for Bioassay Semantification

We define clustering as the second machine learning strategy. This is from corpus observations wherein bioassays with similar text descriptions were semantified with similar sets of logical statements. Thus, bioassays could be clustered based on their text descriptions into semantic groups and each cluster group could be collectively semantified for its bioassays. This task formalism is as follows.

#### Task Formalism.

Let  $K$  be the total number of clusters of bioassays represented by the set  $C = \{c_1, c_2, \dots, c_K\}$ .  $B_{train} = \{b_1, b_2, \dots, b_n\}$  corresponds to the total bioassays in the training set used to obtain optimal cluster centroids; and  $V_{train} = \{v_1, v_2, \dots, v_n\}$  is the vectorized representation of each bioassay to fit the clustering model. Note,  $K < n$ . Further, each cluster  $c_x$  is associated with all the distinct logical statements of the bioassays in the respective cluster group. If cluster  $c_x$  is fitted with two bioassays  $b_p$  and  $b_q$  in the training set, then  $c_x$  is associated with  $sls_{c_x} = sls_{b_p} \cap sls_{b_q}$ . Thus, new logical statements sets are formed as  $\{sls_{c_1}, sls_{c_2}, \dots, sls_{c_K}\}$  associated with the  $K$  clusters. After the clustering semantification model is fitted with  $V_{train}$ , semantification is performed.

Table 9.4: Bioassay semantification results by SciBERT-based labels classification. The first column shows the number of *false* statements (RF) that each bioassay was labeled with—the rows report 3 different experiments (170RF as optimal).

	P	R	F1
160RF	0.33	0.94	0.49
<b>170RF</b>	<b><u>0.37</u></b>	<b><u>0.94</u></b>	<b><u>0.54</u></b>
180RF	0.35	0.94	0.51

Each new bioassay  $b_{test}$  is assigned based on  $v_{test}$  to its closest cluster and semantified with the logical statements set of that cluster.

Clustering has the following alternative semantification task intuition. The domain expert tries to repeat their semantification decisions as much as possible based on similar bioassays they already annotated. In other words, for a new bioassay, they would copy as many logical statements from a similar already semantified bioassay and then decide if additional logical statements were needed. While this latter aspect is not modeled within the clustering problem, our results show that just copying the logical statements between similar bioassays is a significantly accurate automatic semantification strategy.

### Task Model.

Each bioassay text is represented based on the TF-IDF [45] vectorized format. The clustering approach we employ is the K-means algorithm [46]. To determine the optimal clusters  $K$ , we employ the elbow optimization strategy that tries to select the smallest number of clusters accounting for the largest amount of variation in the data [47]<sup>4</sup>.

## 9.5 Bioassay Semantification Experiments

### 9.5.1 Experimental Setup

**1. Labeling Task-specific Settings.** Unlike clustering, the labeling task entails defining *false* logical statement semantification candidates as well. Since each assay had on average 37 *true* logical statements, we experimented with a random set of *false* (RF) statements in the range between 100 and 200 in increments of 10. The values were set to avoid biasing the classifier on only *false* inferences but also to be sufficiently representative.

**2. Three-fold Cross Validation.** For both labeling and clustering, we performed 3-fold cross validation experiments with a training/test set distribution of 600 and 300 assays, respectively. The test set assays were selected such that they were unique between the folds. **3. Evaluation Metrics.** We measure the standard precision, recall, and F1 scores for bioassay semantification per fold experiment. The final scores are then averaged over the three folds.

<sup>4</sup><https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

Table 9.5: Rate of semantifying bioassays on various corpus subsets using SciBERT

	top10	top20	top30	top40	top50	top60	top70	full
TPU	34s	38s	42s	1m 10s	1m 24s	1m 22s	1m 12s	1m 20s
CPU	28m 10s	29m 15s	34m 7s	58m 3s	1h 6m	1h 6m 14s	1h 6m 4s	1h 6m 8s

Table 9.6: SciBERT-based bioassay semantification on corpus subsets starting with only the statements containing the 10 most common predicates (*top 10* row) until the full corpus (*all 80* row). In these experiments, the optimal 170RF was used.

predicates	P	R	F1		predicates	P	R	F1
<i>top 10</i>	<u>0.53</u>	0.94	<u>0.67</u>		<i>top 50</i>	0.36	0.95	0.52
<i>top 20</i>	0.50	0.89	0.64		<i>top 60</i>	0.41	0.92	0.57
<i>top 30</i>	0.45	<u>0.95</u>	0.61		<i>top 70</i>	0.32	0.95	0.48
<i>top 40</i>	0.37	0.94	0.52		<i>all 80</i>	<b>0.37</b>	<b>0.94</b>	<b>0.54</b>

Table 9.7: Bioassay semantification results by K-means clustering

Num. of Clusters	Labels freq $\geq 5$			Labels freq $\geq 4$			Labels freq $\geq 3$			Labels freq $\geq 2$			Labels freq $\geq 1$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
50	0.54	<b>0.75</b>	<b>0.63</b>	0.48	<b>0.80</b>	0.60	0.40	<b>0.84</b>	0.54	0.32	<b>0.89</b>	0.47	0.19	<b>0.94</b>	0.31
100	0.69	0.59	0.63	0.66	0.66	<b>0.66</b> ↑	0.62	0.76	0.68 ↑	0.53	0.85	0.66 ↑	0.32	0.92	0.47 ↑
150	0.83	0.40	0.54 ↓	0.80	0.49	0.61 ↓	0.76	0.63	<b>0.69</b> ↑	0.70	0.79	<b>0.74</b> ↑	0.54	0.90	0.68 ↑
200	0.86	0.34	0.49 ↓	0.83	0.43	0.56 ↓	0.80	0.56	0.66 ↓	0.76	0.72	0.74	0.66	0.89	0.75 ↑
250	0.88	0.22	0.36 ↓	0.86	0.31	0.45 ↓	0.85	0.44	0.58 ↓	0.79	0.65	0.72 ↓	0.71	0.88	0.79 ↑
300	0.91	0.18	0.30 ↓	0.88	0.24	0.37 ↓	0.86	0.35	0.50 ↓	0.81	0.56	0.66 ↓	0.75	0.86	0.80 ↑
350	0.94	0.10	0.17 ↓	0.90	0.15	0.25 ↓	0.88	0.27	0.41 ↓	0.84	0.47	0.60 ↓	0.78	0.86	0.82 ↑
400	0.93	0.06	0.11 ↓	0.93	0.09	0.17 ↓	0.91	0.20	0.32 ↓	0.86	0.38	0.53 ↓	0.80	0.85	0.82
450	0.95	0.05	0.10 ↓	0.94	0.08	0.14 ↓	0.93	0.12	0.22 ↓	0.86	0.27	0.41 ↓	0.81	0.85	<b>0.83</b> ↑
500	0.95	0.03	0.06 ↓	0.94	0.05	0.09 ↓	0.93	0.08	0.15 ↓	0.88	0.17	0.28 ↓	0.82	0.85	0.83
550	0.95	0.03	0.06 ↓	0.95	0.03	0.06 ↓	0.94	0.04	0.08 ↓	0.89	0.09	0.17 ↓	0.82	0.84	0.83
600	<b>1.0</b>	0.02	0.05 ↓	<b>0.95</b>	0.02	0.05 ↓	<b>0.96</b>	0.03	0.06 ↓	<b>0.94</b>	0.04	0.07 ↓	<b>0.83</b>	0.84	0.83

## 9.5.2 Experimental Results

### SciBERT-based Semantification.

Given the results in Table 9.4, we examine the *RQ*: *is the proposed transformer-based neural method effective at semantifying bioassays?* A score of 0.54 F1 tells us, surprisingly, that our attempted neural-based method is not an effective solution to the problem which is a surprising result since it is the state-of-the-art in classification tasks over scientific data [44]. Further, it proves practically inefficient, since, given the full corpus of statements, each test assay is semantified at a rate of 1 hour on the CPU (see Table 9.5). On smaller subsets of the statement labels, the time is indeed faster and the scores are better (see Table 9.6), however, time performance rate of 28 minutes on the smallest subset is still impractical.



---

### K-means Clustering-based Semantification.

Detailed results with their performance rise and fall trends are shown in Table 9.7 for different cluster sizes and labels frequency thresholds within the clusters. E.g., the ‘Labels freq  $\geq 5$ ’ column evaluates only the statements that appeared 5 or more times within the cluster groups when the semantic statements from the various bioassays were aggregated. As the labels frequency threshold is lowered, the semantification score rises. The best scores are obtained when all the statements are considered (the ‘Labels freq  $\geq 1$ ’ column). This method obtains a high semantification score of 0.83 F1. This result when compared with the SciBERT-based neural model frustrates common expectations. Furthermore, this method is effective even w.r.t. the rate of semantification, since bioassays can be semantified in microseconds.

## 9.6 Digital Library Bioassay Semantification Workflows

We now describe the bioassay semantifier as an AI service application powering the structuring of scholarly knowledge in a real-world digital library (DL). The semantifier is importable within any DL that aim to establish knowledge-based information flows as the standard format for reporting and publishing research findings, aka contributions. The high-level workflow is a distributed, decentralized, and collaborative creation and development model comprising information templates, vocabularies, and ontologies (e.g., OBO foundry, Medline, MESH taxonomies, BAO in the Biomedical/Life Sciences domains). We discuss the service as implemented in TIB’s Open Research Knowledge Graph (ORKG) platform (<https://www.orkg.org/>) [1]. The online semantification workflow will be a synergistic combination of automated and manual processes involving the extraction of new ontologized entity types from literature (e.g., target, assay type, experimental conditions in bioassays publications), open access data generation in accord with the FAIR principles thus easily reusable by anyone, and curation support tools for semantified data curation. Figures 2, 3, and 4 depict the workflow. It is pragmatically designed as a hybrid of automatic semantification linked to the BAO (<http://bioassayontology.org/>) and a simplified user interface to help scientists curate their data with minimum effort. This offers a highly accurate semantification model without placing unrealistic expectations on scientists to semantify their assays from scratch. In general, by thus drastically reducing the time required for scientists to annotate their contributions, we can realistically advocate for semantified contributions to become a standard part of the publication process. On such digitalized data, the ORKG additionally supports advanced data interlinking, integration, visualization, and search.

## 9.7 Conclusion

In this work, we have presented an end-to-end model to semantify bioassays descriptions in the context of knowledge-based digital libraries as the ORKG. As a result, we have implemented a highly accurate semantification machine learning method based on clustering. Our code is open source <https://gitlab.com/TIBHannover/orkg/>

1 General 2 Research field 3 Contributions

Specify research contributions ?

+ Add Bioassay + Abstract annotator

Contribution 1 +

Use template ?

+ Test

Help

No data yet  
Start by adding a property from below

+ Add property

Previous step Finish

Figure 9.2: (1) General - add publication metadata; (2) Research field - select a research field from a taxonomy <https://gitlab.com/TIBHannover/orkg/orkg-backend/-/blob/master/scripts/ResearchFields.json>; and (3) Contributions - either structure an articles' contribution as *method*, *material* and *results*, or add a bioassay text description by clicking 'Add Bioassay.' Note the 'Add Bioassay' button is activated only for some research fields in the Life Sciences.

ORKG

Add paper | fdsfa

Specify research

Semantification of Bioassays

Click to upload bioassay .txt file Browse

Enter the Bioassays

copy a text into this form or use the upload button

Submit

Contribution 1 Contribution 2 +

+ Add new View graph

+ Abstract annotator

Figure 9.3: A popup pane to either upload or copy-paste a bioassay text description

orkg-bioassays-semantification. Finally, we report an unconventional finding that resource-light clustering problem formulation can better support bioassay semantification than a state-of-the-art neural approach.

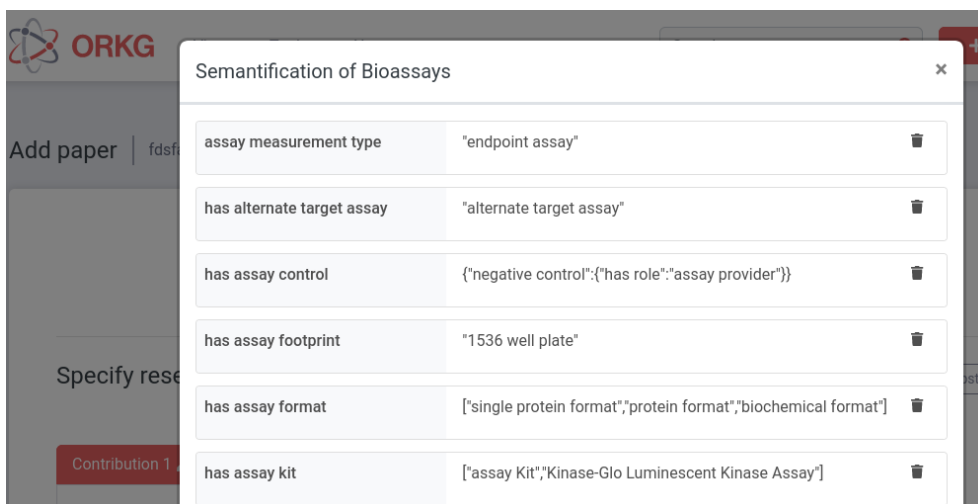


Figure 9.4: An automatically semantified bioassay based on submitted text with an interaction button to delete statements that the domain expert judges invalid

## Aknowledgement

Supported by TIB Leibniz Information Centre for Science and Technology, the EU H2020 ERC project ScienceGraph (GA ID: 819536) and the ITN PERICO (GA ID: 812968). This project was developed in the context of the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Auer, S. *Towards an Open Research Knowledge Graph*. Version 1. 2018. doi: 10.5281/zenodo.1157185.
- [2] Berners-Lee, T., Hendler, J., and Lassila, O. "The semantic web". In: *Scientific american* 284.5 (2001), 34–43.
- [3] Prud'hommeaux, E. and Seaborne, A. *SPARQL query language for RDF, W3C Recommendation*. 2008.
- [4] Katayama, T. et al. "BioHackathon series in 2011 and 2012: penetration of ontology and linked data in life science domains". In: *Journal of biomedical semantics* 5.1 (2014), 1–13.
- [5] Consortium, T. U. "UniProt: the universal protein knowledgebase in 2021". In: *Nucleic Acids Research* 49.D1 (2020), D480–D489. doi: 10.1093/nar/gkaa1100. URL: <https://doi.org/10.1093/nar/gkaa1100>.
- [6] Kanehisa, M. and Goto, S. "KEGG: Kyoto Encyclopedia of Genes and Genomes". In: *Nucleic Acids Research* 28.1 (2000), 27–30. doi: 10.1093/nar/28.1.27. URL: <https://doi.org/10.1093/nar/28.1.27>.
- [7] Jassal, B. et al. "The reactome pathway knowledgebase". In: *Nucleic Acids Research* (2019). doi: 10.1093/nar/gkz1031. URL: <https://doi.org/10.1093/nar/gkz1031>.
- [8] Richa Agarwala, and et al. "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Research* 46.D1 (2017), D8–D13. doi: 10.1093/nar/gkx1095. URL: <https://doi.org/10.1093/nar/gkx1095>.
- [9] Kononova, O. et al. "Text-mined dataset of inorganic materials synthesis recipes". In: *Scientific data* 6.1 (2019), 1–11.
- [10] Kulkarni, C. et al. "An Annotated Corpus for Machine Reading of Instructions in Wet Lab Protocols". In: *NAACL: HLT, Volume 2 (Short Papers)*. New Orleans, Louisiana, 2018, 97–106. doi: 10.18653/v1/N18-2016.
- [11] Mysore, S. et al. "The Materials Science Procedural Text Corpus: Annotating Materials Synthesis Procedures with Shallow Semantic Structures". In: *Proceedings of the 13th Linguistic Annotation Workshop*. 2019, 56–64.
- [12] Kuniyoshi, F. et al. "Annotating and Extracting Synthesis Process of All-Solid-State Batteries from Scientific Literature". In: *LREC*. 2020, 1941–1950.
- [13] Visser, U. et al. "BioAssay Ontology (BAO): a semantic description of bioassays and high-throughput screening results". In: *BMC Bioinformatics* 12.1 (2011), 257.
- [14] Schürer, S. C. et al. "BioAssay ontology annotations facilitate cross-analysis of diverse high-throughput screening data sets". In: *Journal of biomolecular screening* 16.4 (2011), 415–426.
- [15] Vempati, U. D. et al. "Formalization, annotation and analysis of diverse drug and probe screening assay datasets using the BioAssay Ontology (BAO)". In: *PloS one* 7.11 (2012), e49198.

- 
- [16] Clark, A. M. et al. “Fast and accurate semantic annotation of bioassays exploiting a hybrid of machine learning and user confirmation”. In: *PeerJ* 2 (2014), e524.
  - [17] Hoskins, W. M. and Craig, R. “Uses of bioassay in entomology”. In: *Annual review of entomology* 7.1 (1962), 437–464.
  - [18] Irwin, J. “Statistical Method in Biological Assay”. In: *Nature* 172.4386 (1953), 925–926.
  - [19] Thomas, A. L. “Essentials in Bioassay Development”. In: *BioPharm International* 32.11 (2019), 42–45.
  - [20] Abeyruwan, S. et al. “Evolving BioAssay Ontology (BAO): modularization, integration and applications”. In: *Journal of Biomedical Semantics* 5.Suppl 1 (2014), S5.
  - [21] Anteghini, M. et al. “Representing Semantified Biological Assays in the Open Research Knowledge Graph”. In: *Digital Libraries at Times of Massive Societal Transition*. Ed. by E. Ishita, N. L. S. Pang, and L. Zhou. Cham: Springer International Publishing, 2020, 89–98. ISBN: 978-3-030-64452-9.
  - [22] Anteghini, M. et al. “SciBERT-based semantification of bioassays in the open research knowledge graph”. In: *EKAU-PD 2020*. 2020, 22–30.
  - [23] Wang, Y. et al. “PubChem’s BioAssay Database”. In: *Nucleic Acids Research* 40.D1 (2011), D400–D412.
  - [24] Wang, Y. et al. “PubChem BioAssay: 2017 update”. In: *Nucleic Acids Research* 45.D1 (2016), D955–D963.
  - [25] Teufel, S., Carletta, J., and Moens, M. “An annotation scheme for discourse-level argumentation in research articles”. In: *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. Bergen, Norway: Association for Computational Linguistics, 1999, 110–117. URL: <https://aclanthology.org/E99-1015>.
  - [26] Soldatova, L. N. and King, R. D. “An ontology of scientific experiments”. In: *Journal of The Royal Society Interface* 3.11 (2006), 795–803. DOI: 10.1098/rsif.2006.0134. URL: <https://doi.org/10.1098/rsif.2006.0134>.
  - [27] Constantin, A. et al. “The Document Components Ontology (DoCO)”. In: *Semantic Web* 7.2 (2016). Ed. by O. Corcho, 167–181. DOI: 10.3233/SW-150177. URL: <https://doi.org/10.3233/SW-150177>.
  - [28] Pertsas, V. and Constantopoulos, P. “Scholarly Ontology: modelling scholarly practices”. In: *International Journal on Digital Libraries* 18.3 (2017), 173–190.
  - [29] Liakata, M. et al. “Corpora for the Conceptualisation and Zoning of Scientific Papers”. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. Valletta, Malta: European Language Resources Association (ELRA), 2010.
  - [30] Fisas, B., Ronzano, F., and Saggion, H. “A Multi-Layered Annotated Corpus of Scientific Papers”. In: *LREC*. 2016.

- [31] Liakata, M. et al. “Automatic recognition of conceptualization zones in scientific articles and two life science applications”. In: *Bioinformatics* 28.7 (2012), 991–1000. doi: 10.1093/bioinformatics/bts071.
- [32] D’Souza, J. et al. “The STEM-ECR Dataset: Grounding Scientific Entity References in STEM Scholarly Content to Authoritative Encyclopedic and Lexicographic Sources”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, 2020, 2192–2203. isbn: 979-10-95546-34-4. url: <https://aclanthology.org/2020.lrec-1.268>.
- [33] Q. Zadeh, B. and Handschuh, S. “The ACL RD-TEC: A Dataset for Benchmarking Terminology Extraction and Classification in Computational Linguistics”. In: *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University, 2014, 52–63. doi: 10.3115/v1/W14-4807. url: <https://aclanthology.org/W14-4807>.
- [34] Luan, Y. et al. “Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, 2018, 3219–3232. doi: 10.18653/v1/D18-1360. url: <https://aclanthology.org/D18-1360>.
- [35] Brack, A. et al. “Domain-Independent Extraction of Scientific Concepts from Research Articles”. In: *Advances in Information Retrieval*. Ed. by J. M. Jose et al. Cham: Springer International Publishing, 2020, 251–266.
- [36] Ammar, W. et al. “The AI2 system at SemEval-2017 Task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, 592–596. doi: 10.18653/v1/S17-2097. url: <https://aclanthology.org/S17-2097>.
- [37] Dessì, D. et al. “AI-KG: An Automatically Generated Knowledge Graph of Artificial Intelligence”. In: *The Semantic Web – ISWC 2020*. Ed. by J. Z. Pan et al. Cham: Springer International Publishing, 2020, 127–143.
- [38] Augenstein, I. et al. “SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, 546–555. doi: 10.18653/v1/S17-2091. url: <https://aclanthology.org/S17-2091>.
- [39] Gábor, K. et al. “SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, 2018, 679–688. doi: 10.18653/v1/S18-1111. url: <https://aclanthology.org/S18-1111>.

- 
- [40] D'Souza, J., Auer, S., and Pedersen, T. "SemEval-2021 Task 11: NLPContributionGraph - Structuring Scholarly NLP Contributions for a Research Knowledge Graph". In: *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Online: Association for Computational Linguistics, 2021, 364–376. DOI: 10.18653/v1/2021.semeval-1.44. URL: <https://aclanthology.org/2021.semeval-1.44>.
  - [41] Zhou, P. et al. "Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, 207–212. DOI: 10.18653/v1/P16-2034. URL: <https://aclanthology.org/P16-2034>.
  - [42] Wadden, D. et al. "Entity, Relation, and Event Extraction with Contextualized Span Representations". In: *ArXiv abs/1909.03546* (2019).
  - [43] Liu, H., Sarol, M. J., and Kilicoglu, H. "UIUC\_BioNLP at SemEval-2021 Task 11: A Cascade of Neural Models for Structuring Scholarly NLP Contributions". In: *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*. Online: Association for Computational Linguistics, 2021, 377–386. DOI: 10.18653/v1/2021.semeval-1.45. URL: <https://aclanthology.org/2021.semeval-1.45>.
  - [44] Beltagy, I., Lo, K., and Cohan, A. "SciBERT: Pretrained Language Model for Scientific Text". In: *EMNLP*. 2019.
  - [45] "TF-IDF". In: *Encyclopedia of Machine Learning*. Ed. by C. Sammut and G. I. Webb. Boston, MA: Springer US, 2010, 986–987. ISBN: 978-0-387-30164-8.
  - [46] Jin, X. and Han, J. "K-Means Clustering". In: *Encyclopedia of Machine Learning*. Ed. by C. Sammut and G. I. Webb. Boston, MA: Springer US, 2010, 563–564. ISBN: 978-0-387-30164-8.
  - [47] Syakur, M. et al. "Integration k-means clustering method and elbow method for identification of the best customer profile cluster". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 336. 1. IOP Publishing. 2018, 012017.





---

# Bioinforming: training the youth

This chapter is based on:

Marco Anteghini, Katarina Elez, Selle Bandstra, Rijuta Lamba, and Lisanna Paladin.  
Bioinforming.

*Preprint: BioHackrXiv*

## Abstract

Optimal formats to inform and engage young students in novel biology-related fields are short courses. Training schools, such as those lasting for five days, can provide enough content to introduce students to an extensive overview of bioinformatics and scientific career opportunities. In this work, we define a five-day training school format tailored to three target groups of young students: high school students, undergraduate students in biology-related fields and undergraduate students in computational fields. We structure the content and sessions around learning areas consisting of learning topics, detailing the dependencies between them. For each learning topic, we define learning outcomes and learning activities. Moreover, we conceptualize a teaching platform to manage FAIRyfyed (Findable, Accessible, Interoperable, Reusable) training materials that anyone will be able to use to design a new training school in bioinformatics.

## 10.1 Introduction

### 10.1.1 Background

Science promotes innovation and the growth of technology, which in turn stimulates the economic development of countries. Especially in underdeveloped countries, innovation and scientific drive could contribute to becoming competitive in the global economy [1]. Therefore, educating the next generation of scientists is essential for achieving progress. In particular, innovative fast-growing fields such as bioinformatics should receive more attention [2].

Science promotion and engagement of students at a young age can have a decisive influence on their decision to pursue a scientific career [3]. To raise interest in science among the youth, a significant impact can be made at a pre-university level i.e. in the later years of high school education [4]. Free educational content is essential for stimulating young students and increasing their interest in bioinformatics, facilitating their future choice of higher education studies regardless of their social background [5].

Non-governmental, non-profit organization Bioinformatika (Bioinform) aims to achieve this goal by offering various free-of-charge training events. One of the main event formats that Bioinform organizes is a one-week training school designed to introduce the participants to the most relevant topics in bioinformatics and stimulate their curiosity for science learning. This NGO originated from the “BioINForming - Pilot” project, supported by the European Commission through the 2nd call of the Western Balkans Alumni Association (WBAA) in 2021. During this pilot project, the founders of Bioinform organized a five-day training school in bioinformatics for high school students in Montenegro in January 2022. Moreover, a survey was conducted among young students (105) in Montenegro to get insights into their background knowledge and interest in bioinformatics.

In this paper, we present our improved method for designing a five-day training school in bioinformatics tailored to three different target groups: high school students, undergraduate students in biology-related fields and undergraduate students in computational fields. We also show specific use cases. During ELIXIR Bio-

---

Hackathon Europe 2022, we worked with other ELIXIR members and specialists to better define our training school content concerning learning areas and learning topics with their inter-dependencies. It allowed us to associate realistic learning outcomes and the related learning activities to perform. Moreover, we defined a database structure to implement a teaching platform that enables interested trainers to create tailored training schools in bioinformatics.

## 10.1.2 Learning principles

### Definitions

- **Learning area.** A learning area is defined as the “Grouping of traditionally discrete but related subjects with the explicit aim of integrating students’ learning” [6].
- **Learning topic.** In this work, we define learning topics as teaching units that include a presentation of the theory (corresponding to a learning outcome) and one or several learning activities (e.g. an exercise corresponding to a learning outcome). Our topics usually cover two or fewer hours of teaching.
- **Learning path.** Selection and interconnection of learning topics tied together for learners to progress through, while mastering a particular subject or program [7].
- **Learning outcome.** “Totality of information, knowledge, understanding, attitudes, values, skills, competencies or behaviors an individual is expected to master upon successful completion of an educational programme” [8].
- **Learning activity.** The specific activity that will support learners to achieve the learning outcomes. Learning activities are coupled with specific learning outcomes at each level of the Bloom’s taxonomy [9]. They can refer both to theoretical lectures and to practical exercises. The Bloom’s taxonomy is summarised subsequently.

### Bloom’s taxonomy

Bloom’s taxonomy is a set of six levels reflecting the cognitive domain [9]. These six levels represent complexity and specificity that classify the educational learning outcomes. Table 10.1 shows more details on each of Bloom’s levels.

## 10.2 Results

### 10.2.1 Defining and classifying learning areas

Specific areas might have a different role in a specific training context. To structure the training materials, we defined three labels to assign to each learning area: basic (B), core (C), and specialized (S). For example, when considering a training course, the B-labeled areas must cover information required to understand the other covered areas. The C-labeled areas are the ones that are essential to have a general overview

Table 10.1: Description of Bloom's levels.

Level	Level description
1	Remember (recall or reiterate information)
2	Understand (demonstrate understanding of facts)
3	Apply (apply knowledge to real/new situations)
4	Analyse (resolve ideas into simple parts, identify patterns)
5	Synthesize (pull ideas into a coherent whole, create new ideas)
6	Evaluate (make and defend judgments, assess theories and outcomes)

Biology	Computational methods	Interdisciplinary integration
Molecular sequence analysis	Omics	Molecular structure analysis
Molecular phylogenetics	Drug discovery	Machine learning
Bioimage analysis	Public health	Systems biology

Figure 10.1: Classification of learning areas. The first row shows the basic (B) areas. The second row shows the core (C) areas. The third and forth rows show six examples of the specialized (S) areas.

of the main field of the course (e.g. bioinformatics). The S-labeled areas are optional and tailored to the audience. The S areas fully rely upon previous knowledge obtained from the B and C areas. Figure 10.1 shows the different areas we define.

### 10.2.2 Defining and classifying learning topics

In order to provide a structured and comprehensive approach to scientific education, we have defined 20 learning topics. Each learning topic is grouped under a specific learning area, as visible in Table 10.2.

Each of these learning topics has been carefully selected to provide students with a deep understanding of the scientific disciplines they cover. By grouping these topics into specific learning areas, we aim to provide a structured approach to learning that allows students to build on their knowledge and develop a comprehensive understanding of the subject matter.

Table 10.2: Learning topics and their respective learning areas.

Learning Topic	Learning Area
Cell biology	Biology
Molecular biology and genetics	Biology
Biochemistry and biophysics	Biology
Shell scripting	Computational methods
Programming	Computational methods
Bioinformatics definition	Interdisciplinary integration
NGS concepts	Interdisciplinary integration
Biomedical databases concepts	Interdisciplinary integration
Structure determination concepts	Interdisciplinary integration
Sequence alignment	Molecular sequence analysis
Sequence data format	Molecular sequence analysis
Sequence database search	Molecular sequence analysis
Genomics/Epigenomics	Omics
Transcriptomics	Omics
Proteomics/Interactomics	Omics
Metabolomics	Omics
Protein structure	Molecular structure analysis
Structure data format	Molecular structure analysis
Structure prediction	Molecular structure analysis
Structure visualization, comparison and classification	Molecular structure analysis

### 10.2.3 Dependencies among learning topics

To design a training path for our training school we followed the guidelines of "The Learning Path step-by-step protocol". The protocol refers to a project started in the ELIXIR Biohackathon Europe 2021 and further developed in the ELIXIR Biohackathon Europe 2022 [10]. The protocol is field-agnostic and accompanied by guidelines for curriculum developers and trainers.

For the training on a specific topic to be effective, it may or may not require some previous knowledge from the learner. Consequently, it is essential to teach some modules before others, and eventually to assess that the learning of some topic happened before moving on to the dependent topic. After defining 20 topics which we consider essential to our training school, we captured the dependencies among topics creating a coherent learning path, as shown in Figure 10.2.

### 10.2.4 Defining learning outcomes

To identify how the session content can be informative to the students and correctly target their level of knowledge, it is good practice to define learning outcomes by framing them as a continuation of the sentence "By the end of the lesson, the learner will be able to..." and using the verbs in the Bloom's taxonomy (or their synonyms) [9]. As an example, the topic "Protein structure" can have different learning outcomes according to each of the six Bloom's levels (we only show 4 levels since 5 and 6 are meant for highly specialized individuals):

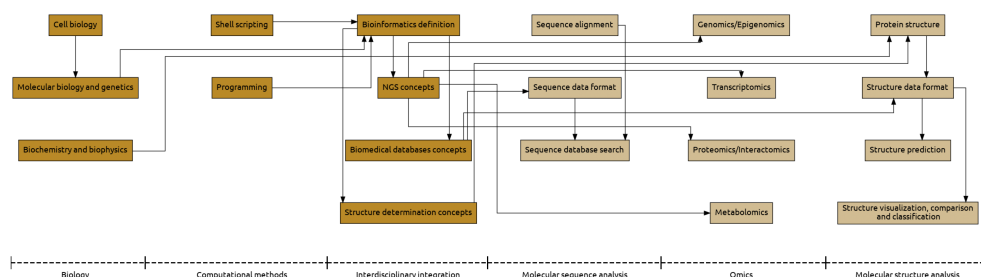


Figure 10.2: Dependencies between learning topics.

### Protein structure analysis

1. Name the most important structure determination methods
2. Describe levels of protein structure
3. Use structure visualization tools
4. Compare different structure prediction methods

## 10.2.5 Defining learning activities

Once the Bloom's levels and the learning outcomes are decided, it is possible to define one or more activities to reach these outcomes for each of the topics.

We list three examples of learning activities associated with the topic, level, and outcome:

### Example level 1

- Topic: Sequence data format
- Level: 1
- Outcome: Classify the different molecular sequences in fasta format (e.g., protein or nucleic acids)
- Activity: Lecture and examples of different molecular sequences

### Example level 2

- Topic: Molecular biology and genetics
- Level: 2
- Outcome: Explain the central dogma of molecular biology
- Activity: Lecture on replication, transcription, and translation

### Example level 3

- Topic: Protein structure analysis

BOLD SYSTEMS

DATABASES

IDENTIFICATION

TAXONOMY

WORKBENCH

RESOURCES

LOGIN

Q

species with interim taxonomy

Public Record Barcode Database (2,546,540 Sequences/154,296 Species/66,530 Interim Species)

All published COI records from BOLD and GenBank with a minimum sequence length of 500bp. This library is a collection of records from the published projects section of BOLD.

Full Length Record Barcode Database (3,154,637 Sequences/216,755 Species/95,477 Interim Species)

Subset of the Species library with a minimum sequence length of 640bp and containing both public and private records. This library is intended for short sequence identification as it provides maximum overlap with short reads from the barcode region of COI.

Enter fasta formatted sequences in the forward orientation:

>UNKNOWN

ATGTTTCGTCGAATCGTTGACTTTTCTCTACCAACCAAGACATCGGCACCTCTTATCTC

CTATTTGGAGCTTGAGCTGGGATGGTGGGAACAGCCCTCAGCTGCTGAATTCGAGCAGAA

TTAGGTGAGCCAGGGACTCTACTCGGGGATGATCAATCTATAATGTAATCGTCACCGCA

CATGCTTTGTAAATACTCTTCTATAGTATGATGCTTATTAATTGGAGCTTCGGGAAC

TGGCTTGTCCCCGTGATAATTGGGGCTCTGACATAGCTTCCCCCGAATAAATAATG

AGCTCTGACTCTCCCCCTTCATCTCTCTTCTACTAGCTCTCTCAATGATGAAGCT

GGGGCGGGGACTGGCTGAACGTTTATCCACCTCTAGCCGGTAATCTTGACATGCTGGA

GCCTCAGTGGATCTACTATTTTCCCCCTCACTAGCTGGAGTATCATCTATTTAGGG

GCTATTAACCTTATTACAATATTATTAATGAAAGCCCTGCAATCTCAATATCAA

ACCCCTTATTCTGATGATCTGTTCTAATCAGACCGGCTCTTCTCTCTCTTTACCA

GTCTAGCTGCTGGGATTACAATGCTTTTAACAGCCGAAACTTAAATACAACCTTCTTT

GATCCTGCAGGAGGAGGAGACCTCTCTCTACCAACCTATTCTGATTTTCGGGCGAC

CCGGAAGTATATTCTTATTCTTCCAGGATTGGATAATTTCGCACATTGTGACATAC

TATTCGGGAAAAAGAGCCATTGGCTATATAGGAATAGTATGAGCTATAATATCAATT

SUBMIT

Figure 10.3: The first step of the exercise is for species identification. An unknown FASTA sequence is pasted in the text box on the BOLD Systems website.

- Level: 3
- Outcome: Use structure visualization tools
- Activity: Visualize a protein structure using Biopython

## 10.2.6 Use cases

### Use case 1 - Identify the species from DNA sequences

**Area:** Omics

**Level:** 3

**Exercise:** Use the BOLD system for DNA Barcoding to identify the species from the DNA sequences provided in FASTA format. The DNA sequences are from the C oxidase subunit 1 mitochondrial gene (COX1 or COI) which is highly efficient for species identification [11].

#### Steps:

1. Open BOLD systems at [https://www.boldsystems.org/index.php/IDS\\_IdentificationRequest](https://www.boldsystems.org/index.php/IDS_IdentificationRequest), select the option "Species Level Barcode Records" and paste the FASTA sequence as shown in Figure 10.3.
2. Visualize the results and take note of the species found, as shown in Figure 10.4. The phylogenetic tree can also be visualized and interpreted (optional).
3. Iterate the process until each sequence is associated with the corresponding species. The final results are visible in Table 10.3.

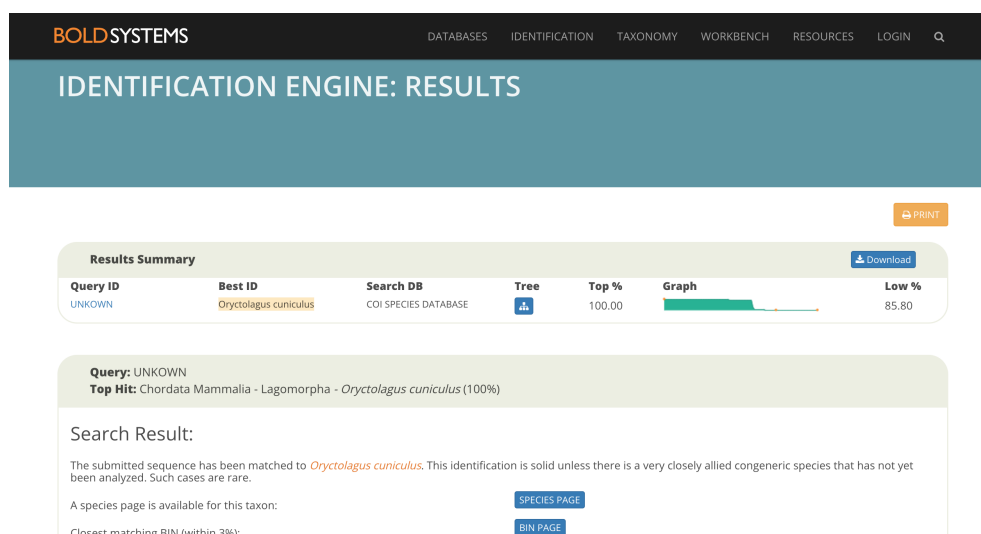


Figure 10.4: The second step of the exercise for species identification. Visualize and take note of the found species.

4. (Optional) Try to associate a protein sequence to each DNA sequence and report its UniProt ID.

## Use case 2 - Prepare the structure of a receptor protein for successive inhibitor detection.

**Area:** Drug discovery

**Level:** 4

**Exercise:** Use the provided scripts and software to access and modify the structure of a receptor protein to prepare it for docking. The receptor protein is PDB:2NYY available from Protein Data Bank (PDB) [12].

### Steps:

1. Move into a specific working directory:

```
cd /home/ubuntu/Desktop/<your_name>/practical5
```

Create necessary subdirectories:

```
mkdir receptor ligands scripts dockings
```

Activate virtual screening environment:

```
conda activate virtual_screening
```



Table 10.3: The final results from the omics exercise. The first column contains a list of identifiers (IDs) while the second column represents the species associated with each sequence.

ID	Organism
Unknown 1	Pan troglodytes (Chimpanzee)
Unknown 2	Pan paniscus (Pygmy chimpanzee) (Bonobo)
Unknown 3	Homo sapiens (Human)
Unknown 4	Equus caballus (Horse)
Unknown 5	Oryctolagus cuniculus (Rabbit)
Unknown 6	Bos taurus (Bovine)
Unknown 7	Ovis aries (Sheep)
Unknown 8	Homo sapiens (Human)
Unknown 9	Canis lupus (Gray wolf)
Unknown 10	Capra hircus (Goat)

2. Move into the 'receptor' directory. Open the PDB structure 2NYY in PyMOL. Considering the information available on UniProt (P0DPI0), export the light chain of botulinum neurotoxin type A into a PDB file called 'botA\_LC.pdb'. Note that the final structure should contain a zinc (Zn) atom, but no calcium (Ca) atoms.
3. Open 'botA\_LC.pdb' file in a text editor and find the x, y, and z coordinates of the C $\alpha$  atom of glutamate (GLU) 262. Write down these coordinates for future reference.
4. Prepare the receptor structure:

```
prepare_receptor4.py -r botA_LC.pdb -A hydrogens
```

5. Open 'botA\_LC.pdbqt' file in a text editor and manually set the charge of the zinc atom to 2.000.

### Use case 3 - Plot trends of publications on Artificial Intelligence in bioinformatics. Highlight challenges and ethical implications.

**Area:** Interdisciplinary integration

**Level:** 5

**Exercise:** Go to PubMed (<https://pubmed.ncbi.nlm.nih.gov/>) and look for papers about Artificial Intelligence in specific time frames. Use the provided scripts (or create your own) to plot the trends, compare them, and write your conclusions. The desired output should be similar to the one in Figure 10.5.

#### Steps:

1. Open the advanced search in PubMed and look for papers containing the words: "Artificial Intelligence", "Natural Language Processing", "Machine Learning", and "Deep Learning" in a specific time frame (e.g., from 1980 to 1985):

```
(((((Natural language processing[Title/Abstract]) OR  
(Artificial intelligence[Title/Abstract])) OR  
(Machine learning[Title/Abstract])) OR  
(Deep Learning[Title/Abstract])) AND  
("1980"[Date - Publication] : "1985"[Date - Publication]))
```

2. Repeat the search and annotate the results for time frames of 5 years until March 31st, 2023. Fill the lists with the results from the queries.

```
time=['1985', '1990', '1995', '2000', '2005', '2010', '2015', '2020', '2023']  
n_of_p1=[90, 272, 386, 506, 1093, 2897, 7069, 49398, 103962]
```

3. Repeat the search for the same time frames. This time add the words 'Ethic', 'Ethics', or 'Challenges' in the advanced search. Fill the list.

```
(((((Artificial Intelligence[Title/Abstract]) OR  
(Machine Learning[Title/Abstract]) OR  
(Deep Learning[Title/Abstract]) OR  
(Natural Language Processing[Title/Abstract])) AND  
((Ethic[Title/Abstract]) OR  
(Challenges[Title/Abstract]) OR  
(Ethics[Title/Abstract])) AND  
("2015"[Date - Publication] : "2023-03"[Date - Publication]))
```

```
n_of_p2=[1, 4, 8, 14, 36, 117, 355, 3546, 9862]
```

4. Plot and compare the trends.

```
import matplotlib.pyplot as plt  
  
# Create the plot  
plt.plot(time, n_of_p1, color='b', label='Paper on AI')  
plt.plot(time, n_of_p2, color='#77dd77', \  
label='Paper mentioning Ethics and Challenges of AI')  
  
# Set x and y axis labels  
plt.xlabel('Time steps')  
plt.ylabel('Number of publications')  
  
# Add legend  
plt.legend()  
  
# Adjust x-ticks spacing
```

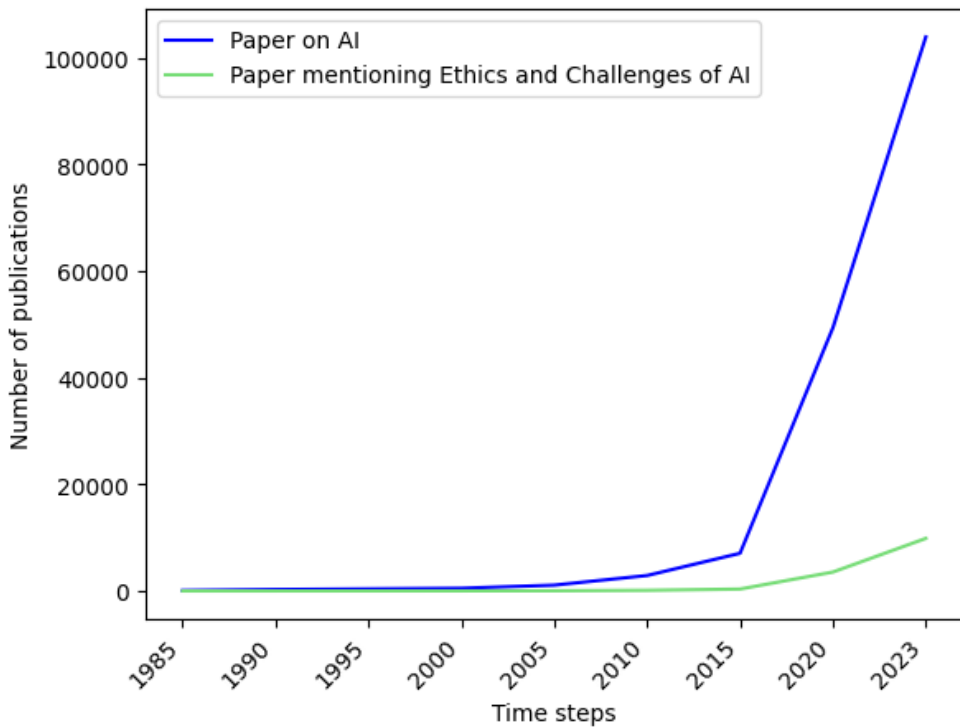


Figure 10.5: Trend in the number of papers published in a specific time range mentioning one of the following words: "Artificial Intelligence", "Natural Language Processing", "Machine Learning" or "Deep Learning". The blue line shows the general trend in the number of papers mentioning these words, the green line shows those that are also associated with the words: "Ethics", "Ethic" or "Challenges".

```
plt.xticks(rotation=45, ha='right')

# Show the plot
plt.show()
```

5. Write your conclusion. Are the trends identical? Do you think Challenges and Ethics of AI are well-represented in scientific publications? Motivate your answer.

### 10.2.7 Teaching platform

To create and share FAIR training materials, we envisaged the construction of an open-access teaching platform. The platform will store training materials and provide suggestions for designing a new training event. The materials will be labeled with information such as area, topic, bloom's level, outcomes, material type, authorship and time to perform the activity. Topic dependencies will be fundamental in

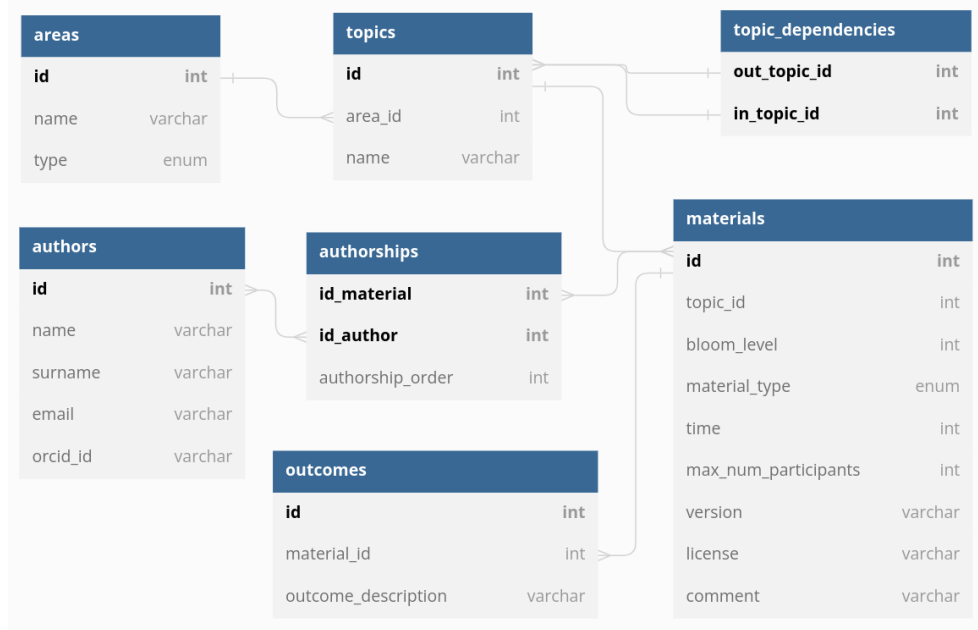


Figure 10.6: Database structure for the teaching platform

providing suggestions for designing a learning path. An overview of the database structure underlying the teaching platform is visible in Figure 10.6.

### 10.2.8 Additional topics and tailored content examples

Tailored content is essential for a successful outcome in every training event. In particular, a 5-day training school risks being too condensed and overwhelming if not properly planned. For these reasons, we believe it is important to add certain flexibility to our training events, monitoring the students' during and after the training school implementation. Our format aims to provide a space for general discussion and soft skill improvement in parallel to our highly structured training materials. In the regular call for applications to a training event, we ask questions like "How would you rate your interest in knowing more about the following topics?" and "What additional topics would you be interested in knowing more about?". Additionally, we dedicate sessions to complementary discussions for scientific training schools such as: "Career in bioinformatics", "Ethics and challenges of AI", "Environmental protection", "Gender imbalance in science" and "Erasmus exchange opportunities". As an example, the session "Ethics and challenges of AI" would consist of a discussion on the challenges connected to the use of AI in healthcare. These include: ensuring patient privacy and data security, addressing issues of bias and fairness in algorithmic decision-making, and establishing accountability for the actions of AI systems. Also, practical challenges can be clustered in the following groups: data integration problems, training set bias and black-box characteristics.

---

## 10.3 Conclusion

Preparation of training materials for a training school in bioinformatics tailored to different audiences requires a general structure that should both be flexible and well-planned. In this work, we show how to prepare a structure which can be adapted to different target groups. In particular, we define learning areas, learning topics and the dependencies between them. These dependencies define a learning path which can be a valuable resource while organizing training schools. It can work as a backbone for defining learning outcomes, that satisfy both the student and the trainer, through specific learning activities.

We developed a five-day training school format that caters to three target groups of young students: high school students, undergraduate students in biology-related fields, and undergraduate students in computational fields.

Our focus on interdisciplinary integration is reflected in the fact that four of the learning topics fall under the "Interdisciplinary integration" learning area. This reflects the growing importance of interdisciplinary approaches in scientific research, and our desire to prepare students to tackle complex scientific problems using a multidisciplinary approach. Our ultimate goal is to inspire young students to pursue careers in bioinformatics and related fields and contribute to scientific advancements.

One of the main advantages of our format is its ability to provide students with a broad overview of bioinformatics and scientific career opportunities. The tailored content ensures that each group receives information appropriate for their level of knowledge and experience. We also incorporate general discussion and soft skill development to create a more engaging and interactive learning environment. Another important feature of our approach is the use of assessment forms during and after training to monitor student progress and evaluate the effectiveness of the training school. By soliciting feedback from students, we can gather information that can be used to improve future iterations of the training school.

We believe the materials produced in our training initiatives should be FAIR and open to everyone interested in using them. Moreover, it should be possible for other trainers to easily contribute their training materials. For these reasons, we defined a structure for an open-source teaching platform. In the future, we envisage the implementation of such a training platform and the addition of new materials specific to different target groups that will participate in our training events.

## Acknowledgment

This work was funded by ELIXIR, the research infrastructure for life-science data. We are also grateful to the Investment-development fund (IDF) of Montenegro for their support. We thank Anna Spackova, Alexia Cardona, Renato Alves, Denise Slenter, Bérénice Batut, Linelle Abueg, Suchitra Thapa and Gültekin Ünal for their valuable suggestions. The first author of this paper project was funded through the PerICo International Training Network and has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No. 812968.

## References

- [1] Lema, R., Kraemer-Mbula, E., and Rakas, M. “Innovation in developing countries: examining two decades of research”. In: *Innovation and Development* 11.2-3 (2021), 189–210. doi: 10.1080/2157930x.2021.1989647. URL: <https://doi.org/10.1080/2157930x.2021.1989647>.
- [2] Bishop, O. T. et al. “Bioinformatics Education–Perspectives and Challenges out of Africa”. In: *Briefings in Bioinformatics* 16.2 (2014), 355–364. doi: 10.1093/bib/bbu022. URL: <https://doi.org/10.1093/bib/bbu022>.
- [3] Osborne, J. and Dillon, J. “Science Education in Europe: Critical Reflections”. In: (2008).
- [4] García, F. A. M. et al. “Digital Technologies at the Pre-University and University Levels”. In: *Sustainability* 12.24 (2020), 10426. doi: 10.3390/su122410426. URL: <https://doi.org/10.3390/su122410426>.
- [5] Mönkediek, B. and Diewald, M. “Do academic ability and social background influence each other in shaping educational attainment? The case of the transition to secondary education in Germany”. In: *Social Science Research* 101 (2022), 102625. doi: 10.1016/j.ssresearch.2021.102625. URL: <https://doi.org/10.1016/j.ssresearch.2021.102625>.
- [6] Education, U. I. B. of. *Glossary of Curriculum Terminology*. UNESCO-IBE, 2013.
- [7] Nabizadeh, A. H. et al. “Learning path personalization and recommendation methods: A survey of the state-of-the-art”. In: *Expert Systems with Applications* 159 (2020), 113596. doi: 10.1016/j.eswa.2020.113596. URL: <https://doi.org/10.1016/j.eswa.2020.113596>.
- [8] Statistics, U. I. for. *International Standard Classification of Education*. UNESCO-UIS, 2011.
- [9] Bloom, B. S. *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. David McKay Co Inc., 1956.
- [10] Cardona, A. et al. “BioHackathon project #32: Training booster. Developing Learning Paths”. In: (2022).
- [11] Rodrigues, M. S., Morelli, K. A., and Jansen, A. M. “Cytochrome c oxidase subunit 1 gene as a DNA barcode for discriminating *Trypanosoma cruzi* DTUs and closely related species”. In: *Parasites & Vectors* 10.1 (2017). doi: 10.1186/s13071-017-2457-1. URL: <https://doi.org/10.1186/s13071-017-2457-1>.
- [12] Berman, H. M. et al. “The Protein Data Bank”. In: *Nucleic Acids Research* 28.1 (2000), 235–242.







---

# Discussion

## 11.1 Introduction

In the era of unprecedented technological advancements, we are currently witnessing a radical change in the way we approach scientific research. The emergence and proliferation of Artificial Intelligence (AI) have revolutionized various domains, including bioinformatics and computational biology, enabling us to delve deeper into complex biological phenomena. This thesis aims to shed light on the pivotal role of AI, particularly machine learning (ML) and deep learning DL, in the context of predicting peroxisomal protein function, localization and interactions, and its implications for understanding cellular processes. By harnessing the power of computational methods and integrating them with AI algorithms, it is possible to unravel intricate relationships between protein sequences and their subcellular and sub-organelle localization, the way they interact between them, and the function they perform, paving the way for novel discoveries and advancing our understanding of cellular organization.

This thesis explores the application of natural language processing (NLP) technologies applied to text files (i.e. biological assays) and protein sequences. It shows the adaptability of these technologies to various contexts. The thesis chapters follow a linear structure, but they reveal two distinct lines of parallel research. The first line explores the applicability of NLP-based technologies to protein sequences, treating them as textual sentences (**Chapter 2, Chapter 3, Chapter 4, Chapter 5, Chapter 6, Chapter 7**). The second line focuses on studying NLP technologies themselves and their application to text files derived from biological assays (**Chapter 8 and Chapter 9**).

Both lines of research are ultimately useful to the peroxisomal-related research. Researchers can adopt the semi-automatic semantification system generated from the studies on the NLP and bioassays semantification to upload and share their bioassays. Also, they can take advantage of the tools herein developed for performing several protein prediction tasks.

Finally, the thesis looks into how to share bioinformatics knowledge and pass it to the next generation. It is highlighted how the efforts of non-profit organizations like Bioinform (<https://bioinform-org.github.io/>), through their training events and free educational content, play a crucial role in stimulating young students' interest in bioinformatics and encouraging their future involvement in higher education studies. This work is presented in (**Chapter 10**).

The overarching goal of this thesis was to develop and deploy DL-based embeddings to gain insights into peroxisomal biology.

The specific objectives of the thesis were:

1. To understand, explore and compare the practicality and effectiveness of computational approaches in various settings around peroxisomal related research;
2. To adapt DL-based embedding to specific prediction tasks, focused on peroxisomal research;
3. To define strategies focused on promoting bioinformatics knowledge to the next generations, incorporating critical thinking skills.

The objectives and the approach used to accomplish them are shown in Table 11.1.

Table 11.1: Approaches adopted to reach the thesis objectives.

Approach	Objectives
Defining pipelines to perform peroxisomal and sub-peroxisomal protein localization ( <b>Chapter 2</b> )	1
Comparing and concatenating the DL-based protein sequence embeddings among them and against classical protein encoding approaches, while predicting the sub-peroxisomal or the sub-mitochondrial localization of a protein ( <b>Chapter 3</b> )	1,2
Expanding and validating the methodology for PTS1 signal detection and develop a web server to easily perform sub-organelle localization prediction tasks ( <b>Chapter 4</b> )	1,2
Expanding and validating the methodology for the mPTS signal ( <b>Chapter 5</b> )	1,2
Exploring the capabilities of such DL-based protein embeddings in predicting the peroxisomal proteins interactions ( <b>Chapter 6</b> )	1,2
Scaling the DL-based approach for predicting transporter proteins and specifically test a peroxisome proteins subset ( <b>Chapter 7</b> )	1,2
Developing a semi-automatic system for bioassay semantification ( <b>Chapter 8</b> and <b>Chapter 9</b> )	1
Developing an improved method for designing training schools in bioinformatics ( <b>Chapter 10</b> )	3

Supplemented by additional data analysis, this chapter discusses the results and main conclusions presented in the previous chapters in relation to the stated objectives. It concludes by addressing the implications of these results, identifies further gaps in knowledge and makes recommendations for future studies.

## 11.2 Contextualizing the peroxisomes in the Eukaryotic cell

Eukaryotes possess several subcellular localizations, including intra-organellar compartments, which fulfill specific cellular tasks. Transporting proteins accurately to their intended destinations within organelles is vital for their proper functioning. That means, localization is strictly connected to protein interactions and function.

Our current knowledge of the subcellular organelles and their protein functions in eukaryotic cells is still incomplete, despite the ongoing discoveries being made. As a proof of this, Xu *et al.* (2023) have recently identified a new type of organelle inside animal cells that acts as a phosphate reservoir, regulating nutrient levels and triggering processes for tissue maintenance [1]. This discovery, described as one of the first to find phosphate storage in animal cells, has significant implications for understanding cell physiology.

Despite their involvement in numerous metabolic and non-metabolic processes, peroxisomes and their proteins are less well understood and require further in-depth investigation in comparison to other organelles. On 15/05/2023 the UniProtKB database <https://www.uniprot.org/> contains 1558 reviewed entries (manually curated) and 138406 automatically generated entries associated to the peroxisomal subcellular localization. A comparison of the peroxisomal proteome with respect to other organelles, is visible in Table 11.2.

Table 11.2: Number of proteins associated with a specific organelle available on UniProtKB, along with their dimensions. The column **SwissProt** contains the manually curated entries. The column **TrEMBL** contains the automatically generated entries

Organelle	SwissProt	TrEMBL	Dimension
Lipid droplet	767	117,789	0.5-3.7 $\mu\text{m}$
Peroxisome	1,558	138,406	0.1-1 $\mu\text{m}$
Lysosome	2,672	143,413	0.1-1.2 $\mu\text{m}$
Vacuole	6,089	282,580	-
Golgi apparatus	8,961	722,350	-
Endoplasmic reticulum	9,938	847,525	0.2 $\mu\text{m}$ , each layer
Mitochondria	20,596	3,800,632	0.5-3 $\mu\text{m}$
Cytosol	35,370	798,659	-
Nucleus	51,732	5,814,474	5-20 $\mu\text{m}$

As a species specific example, from the Human Protein Atlas (<https://www.proteinatlas.org/>), it is possible to retrieve the amount of proteins per subcellular compartment. Figure 11.1 shows the compartment also presented in Table 11.2 except from the vacuole which is not present in humans. The bar plot shows the distribution of the known protein coding genes in humans and we can notice that despite of the high importance of peroxisomal protein in the cell physiology we still do not have an high amount of entries in UniProt (e.g. the lysosomal protein entries are greater in UniProt).

This gives an idea on how we still can improve the knowledge about peroxisomal proteins, their functions and role in cellular physiology, particularly in humans.

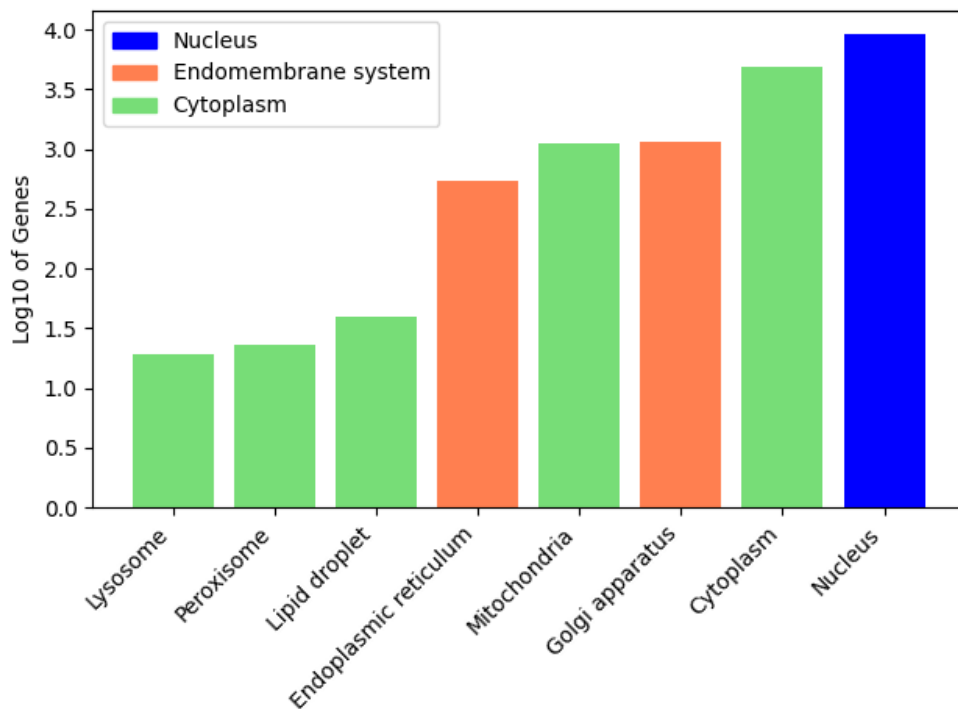


Figure 11.1: Bar plot showing the distribution of classifications of proteins (represented as Log10 of the coding genes) in organelles and subcellular structures in humans. Note that one protein can localize to more than one compartment. The bars are colored according to meta compartment.

---

## 11.3 The publication network

Providing proper context for the research findings presented in this thesis is important in advancing scientific knowledge, particularly in the field of peroxisomal research. By connecting the research to other relevant publications and works, the scientific community can achieve a deeper understanding of peroxisomes, enhance knowledge acquisition.

Computational approaches presented in this thesis play a crucial role in this process by providing tools and methodologies to analyze and integrate diverse datasets, enabling researchers to draw meaningful insights from large-scale data. By leveraging experimentally validated computational techniques, researchers can identify novel peroxisomal proteins, decipher their functions and interactions, and uncover underlying regulatory mechanisms. Furthermore, the integration of computational findings with other published studies facilitates the identification of commonalities, discrepancies, and knowledge gaps in the field. This collaborative approach, combining computational approaches with other experimental and theoretical investigations, promotes a holistic understanding of peroxisomes and paves the way for further advancements in peroxisomal research. By disseminating these findings to the scientific community, researchers can contribute to the collective knowledge base, foster collaborations, and drive future research in peroxisome biology and related fields.

Figure 11.2 shows a graph that represents the direct connections of the publications related to this thesis. The connections are from both the two lines of research mentioned in section 11.1. In the graph the node are publication and their size is proportional to their outgoing edges which are their related citations. In Figure 11.2B can be seen that the two bigger nodes have this size because are often cited together. More precisely, Kamoshita *et al.* (2023) [2] presents how bioinformatics pipelines can be adopted to further improve experimental studies, while Anteghini *et al.* (2021) [3] shows a practical example of the precursor computational method that was also used in Kamoshita *et al.* (2023). This highlights the importance of contextualizing the research on a larger scale, always considering the correlations and connections between similar works. An overview of each publication network can be generally visualized on <https://www.connectedpapers.com/>. A specific network for Anteghini *et al.* (2021) can be accessed at

<https://www.connectedpapers.com/main/0ac7cf70614d054651384b34bc16f4051d95af1f/In%20Pero%3A-Exploiting-Deep-Learning-Embeddings-of-Protein-Sequences-to-Predict-the-Localisation-of-Peroxisomal-Proteins/graph> or using the qr-code in Figure 11.3.

## 11.4 The role of computational approaches in peroxisomal proteins research

In the peroxisomal research community, well-known computational resources, such as the PeroxisomeDB, are not updated with recent discoveries and the specific servers are often not functional [5]. Also, algorithms for PTS1, PTS2 and especially mPTS have not been updated with the recent discoveries in consensus motifs. For these reasons, it is necessary to develop new easily accessible tools that rely on

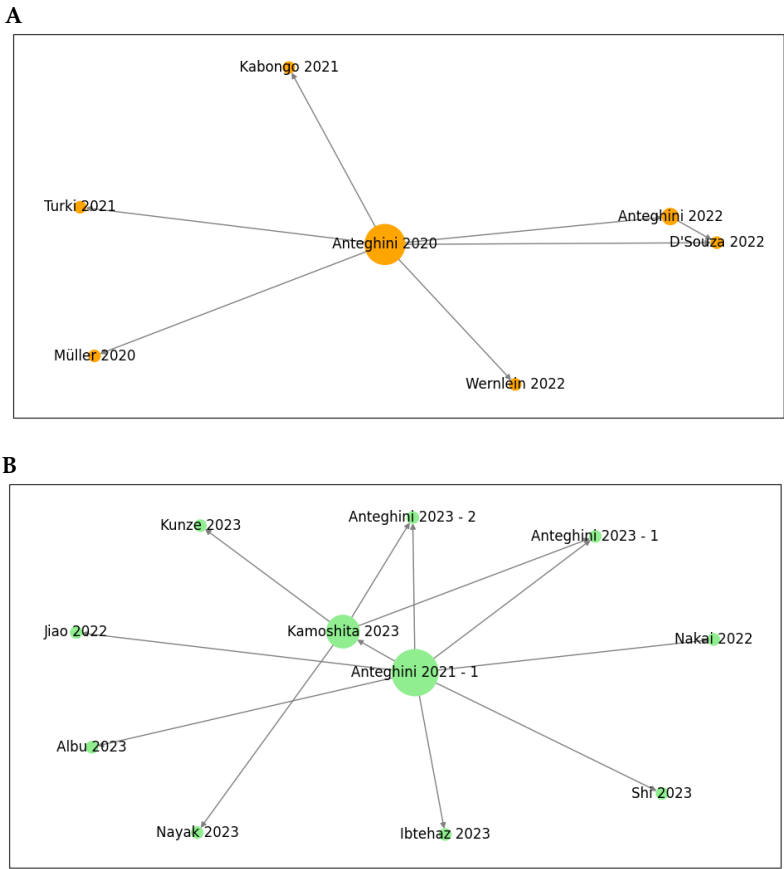


Figure 11.2: Networks showing connected papers of the two research lines. The nodes represent the publication. The node size is proportional to the number of outgoing arrows. **A.** Bioassay semantification papers networks. This graph shows the paper connected to Anteghini *et al.* 2020 which was the first publication showing a novel approach for biological assays semi-automatic semantification [4]. **B.** Peroxisomal research papers network. This graph shows the papers connected to Anteghini *et al.* (2021) which was the first computational method based on deep-learning approaches in peroxisomal research investigation [3].



Figure 11.3: Qr-code of the publications network

---

novel approaches and recent discoveries thus facilitating the peroxisomal community. **Chapter 2** presents guidelines to retrieve peroxisomal proteins candidates for further experimental validation. As an example, these guidelines were followed in Kamoshita & Kumar *et al.* (2022) [2]. In particular, the bioinformatics analysis was performed with the following steps:

1. The *Danio rerio* (zebrafish) proteome available on UniProt was screened for proteins carrying a PTS1 at the very C-terminus using all possible combinations of residues found in PTS1 motifs;
2. The corresponding fasta sequences were analyzed using the software TMHMM2 [6] to remove proteins presenting transmembrane helices
3. WoLF PSORT was executed to obtain remove proteins putatively localized in the Endoplasmic Reticulum (ER) [7];
4. The identified proteins were further analyzed by PTS1 and PTS2 predictor algorithms [5, 8];
5. Selected sequences were screened for conservation of the potential PTS1 using BLAST2.0. [9, 10];
6. Mitochondrial targeting and potential targeting to the secretory pathway were screened;

The above example shows the efficacy of the work presented in this thesis, which for the zebrafish allowed to improve the known peroxisomal proteome.

Moreover, the potential of DL-based algorithms in peroxisome-related research were not yet properly exploited before the works presented in this thesis, in particular when dealing with protein sequences.

As support to this, PubMed shows only 14 papers mentioning the word 'peroxisome' or 'peroxisomal' in their title combined with one or more of the following: 'machine learning', 'artificial intelligence', 'deep learning' or 'web server' in their abstract. The search query is: ((peroxisome[Title]) OR (peroxisomal[Title])) AND ((Deep Learning[Title/Abstract]) OR (Machine learning[Title/Abstract]) OR (Web server[Title/Abstract]) OR (Artificial intelligence[Title/Abstract])). The first is paper dated 2003 and the last March 2023 with half of them condensed in the last 5 years. Is is relevant to mention that among the 6 publications form 2021 to 2023, 4 of them are contribution discussed in this thesis. If the same search is performed using the words 'mitochondria' or 'mitochondrial' instead of 'peroxisome' or 'peroxisomal', the results would be 45 papers. The date of both searches is 15 May 2023.

From these statistics it is possible to deduce that: i) DL-based method in peroxisomal research were mostly discussed in this thesis and not fully exploited by the scientific community; ii) computational methods are more studied for other organelles such as mitochondria.

With the contributions presented in this thesis **Chapter 3, Chapter 4, Chapter 5, Chapter 6** and **Chapter 7** we contributed to set new standards for computational approaches in peroxisomal research. In addition, this approaches can be scaled to any eukaryotes and to far more organelles.

## 11.5 The applicability of DL-based protein sequence embeddings

The applicability of deep learning-based protein sequence embeddings has recently emerged in protein research and in particular, in peroxisomal proteins-related research [3, 8, 11]. These embeddings are capable of capturing complex patterns and relationships within protein sequences, enabling researchers to gain valuable insights into peroxisomal protein function and structure [12–16].

By leveraging large-scale protein sequence databases and applying deep learning algorithms, these embeddings can effectively encode the inherent features and representations of peroxisomal proteins. They provide a compact yet comprehensive numerical representation of protein sequences, facilitating downstream analyses such as sub-organelle localization, functional annotation, and protein-protein interaction prediction. The adaptability of deep learning-based protein sequence embeddings offers tremendous potential for accelerating discoveries in peroxisomal biology and enhancing our understanding of these vital organelles.

By harnessing their power, researchers can unlock new avenues for investigating peroxisomal functions and ultimately contribute to advancements in therapeutic interventions and personalized medicine targeting peroxisomal disorders.

### 11.5.1 DL-based protein embeddings for subcellular and sub-organelle localization

In **Chapter 2** it is shown how DL-based embeddings can be used in localizing peroxisomal proteins and moreover how to validate sub-organelle compartments in both peroxisome and mitochondria. **Chapter 3** highlights the accuracy of coupling embeddings representation and ML classifiers when predicting sub-peroxisomal and sub-mitochondrial protein localization. The chapter demonstrates how the scientific community can take advantage of pre-trained embeddings obtained by other research groups instead of just train new models thus saving computational resources.

In **Chapter 4** it is shown how these new technology can be re-adapted to PTS signal detection and integrated in user-friendly web servers.

A similar PTS signal validation approach is presented in **Chapter 5** with the mPTS signal validation algorithm thus proving the adaptability of DL-based protein embeddings to another very specific prediction task.

State-of-the-art performances were reached for all the above mentioned applications.

### 11.5.2 DL-based protein embeddings for transporter protein prediction

Considering the limited availability of tools and data sets for transporter protein prediction, **Chapter 7** focuses on generating a new data set and compare the DL-based embeddings algorithm with two previously used data sets in other studies [17, 18]. This approach ensured unbiased results and provided support for the conclusions.



---

The development of PortPred demonstrated the adaptability of deep learning-based sequence embeddings in predicting transporter proteins, opening up new possibilities for further analysis. Among the tested embeddings, ESM-1b showed the best performance. Its inclusion in the embedding concatenation, along with UniRep, SeqVec, and ProtBert, resulted in noticeable improvements across all evaluation metrics. However, it is important to note that ESM-1b embeddings cannot be generated for protein sequences longer than 1024 residues. Nonetheless, the average length of transporter protein sequences found in SwissProt (as of 23.12.2022) is 447 residues, indicating that transporter proteins are typically shorter than 1024 residues.

The other tested embeddings exhibited comparable performances and are generally more suitable for longer sequences. Therefore, it is important to make informed choices based on specific needs when utilizing these deep learning-based pre-trained representations.

The main objective of this chapter was to demonstrate the re-usability of pre-trained embeddings for transporter protein prediction and highlight how existing methods can be combined and tailored to enhance performance. Instead of solely producing more models, the chapter emphasizes the importance of studies focused on adapting, validating, and reporting the limitations of these embeddings, which were developed using extensive computational resources. This critical evaluation is essential to avoid being overwhelmed by a multitude of invalidated models that do not provide useful information.

### 11.5.3 DL-based protein embeddings for protein-protein interactions prediction

The evaluation and comparison of DL-based embeddings for PPI prediction, are often lacking, particularly when compared to state-of-the-art tools that incorporate different types of information or more elaborate algorithms. Therefore, it is crucial to thoroughly analyze the performance of DL-based embeddings for PPI prediction tasks and determine their applicability for fast and accurate predictions.

**Chapter 6**, focuses on benchmarking and comparing the most commonly used DL-based embeddings for predicting PPIs based solely on protein sequence information. Additionally, an example of integration with AlphaFold2 is presented [19].

The obtained results indicate promising preliminary outcomes for PPI prediction using DL-based protein sequence embeddings. This suggests that pre-trained deep learning models for protein sequence embedding can be effectively utilized for PPI prediction, opening up possibilities for further analysis.

To validate the effectiveness of DL-based embeddings for PPI prediction, several important steps remain to be taken:

1. **Comparative Analysis:** It is crucial to compare the performance of the PPI predictor developed using DL-based embeddings against state-of-the-art methods. This analysis will provide insights into the strengths and weaknesses of the DL-based approach and its competitiveness in relation to existing tools.
2. **Expansion to Other Subcellular Localizations:** To enhance the predictor's applicability, training it on PPIs from different subcellular localizations is neces-

sary. By incorporating a diverse range of PPI data, the predictor can capture variations in protein interactions across different cellular environments, further refining its predictive capabilities.

3. Integration into a unified tool: Integrating the DL-based algorithms, along with other relevant functionalities, into a single comprehensive tool can streamline PPI prediction workflows. This integrated tool would provide researchers with a user-friendly interface and a range of features for efficient and accurate PPI analysis.
4. Experimental validation and candidate PPIs suggestions: To validate the predictions made by the DL-based predictor, it is essential to experimentally verify the suggested PPIs. This experimental validation will provide a rigorous assessment of the predictor's reliability and enable the identification of true positive PPIs.

By addressing these steps, it is possible to gain a comprehensive understanding of the DL-based embeddings' effectiveness for PPI prediction. This will enable researchers to make informed decisions regarding the application of DL-based approaches in PPI studies and facilitate advancements in our understanding of protein interactions.

#### **11.5.4 Integrating the information of different embeddings**

Different embeddings can contain different types of information and combining multiple embeddings can lead to improved performances [3, 11]. To investigate the correlation between embeddings, Pearson's correlation coefficient can be used. Combining four different encodings and/or embeddings has been shown to lead to better predictions of peroxisomal sub-localization [3]. Specifically, concatenating UniRep, SeqVec, Protbert, and ESM-1b has been found to significantly improve performance [11]. This suggests that different embeddings contain unique and complementary information about the properties of the protein sequence. Figure 11.4 illustrates the lack of correlation between four embeddings in an example data set, explained in Chapter (Unirep,Seqvec,PROTBERT and ESM-1b).

#### **11.5.5 Approaches to reduce the dimensionality of concatenated embeddings**

The thesis addresses the challenge of reducing the dimensionality of DL-based concatenated embeddings while preserving their high informativeness. It builds upon the understanding that different embeddings contain distinct and complementary information regarding protein sequence properties. By combining different embeddings, namely UniRep, SeqVec, in **Chapter 3** and UniRep, SeqVec, Protbert, and ESM-1b, in **Chapter 7** the projects demonstrate improved predictions of peroxisomal sub-localization.

To further enhance the efficiency of these concatenated embeddings, **Chapter 7** presents a comprehensive framework for maximizing the predictive power of DL-based concatenated embeddings, reducing the concatenate embedding vector di-

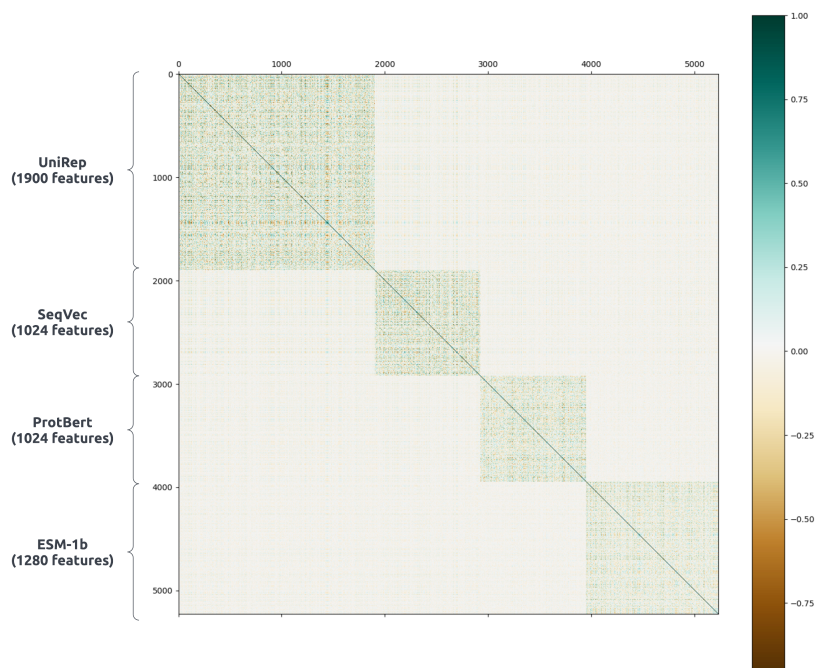


Figure 11.4: Figure from Anteghini *et al.* (2023) [11]. Correlation among UniRep (1900 features), SeqVec (1024 features), PortPred (1024 features) and ESM-1b (1280 features) protein sequence embeddings. Pearson's linear correlation is used and are calculated over 1580 transporter protein sequences of the PorPred training set. The four embeddings are uncorrelated

mension and while keeping them highly informative. The framework consists in a Recursive Feature Elimination (RFE) approach.

It is a technique that identifies the optimal subset of informative features for a given task. In the case of this study, RFE is applied to the four concatenated embeddings. It starts by considering all features in the training dataset and iteratively removes one or more features until either the performance deteriorates or a predetermined number of features remains. The performance evaluation is carried out using a 10-fold cross-validation classification approach.

The resulting model utilizes a hybrid input consisting of the relevant features from the concatenated embeddings, retaining only the specific values that contribute to the classification task. For example, out of the initial 5228 features, RFE selects and retains 2328 relevant features.

## **11.6 Limits in the usage of DL-based protein embeddings**

The use of DL-based protein embeddings has shown great promise in various protein-related tasks, such as subcellular classification, protein function prediction, and PPI prediction. However, there are certain limitations that need to be considered.

Firstly, DL-based embeddings heavily rely on the availability and quality of training data. Insufficient or biased training data may lead to suboptimal embeddings and reduced predictive performance. The protein embeddings adopted in this thesis (UniRep, SeqVec, ProteinBERT, ESM-1b, ESM2) [12, 13, 15, 16, 20] were built either using the protein sequences available on UniRef90 or UniRef50, which are non-redundant clusters with large number of sequences, thus ideal for DL methods [21]. Moreover, when adapting these embeddings for predicting specific tasks, the data set of protein sequences also needs to be carefully built. In the works presented in this thesis, the starting training set was always built following criteria like relevance (only representative samples of the data that the model will encounter in real-world scenarios were used), quality (only manually annotated protein sequences were used), sufficient size (the maximum size of available non-redundant protein sequences was always considered), diversity (the protein sequences were always clustered for at least 40% of sequence identity), and balanced distribution (the classes were always balanced or weighted accordingly).

Additionally, DL-based models can be computationally expensive and require substantial computational resources, limiting their practicality for certain applications. The works presented in this thesis, take advantage of pre-trained models so the ML algorithm can only run the already embedded proteins, in a fast and efficient way that can be performed on a laptop.

Another limitation is the lack of interpretability of DL-based embeddings. While they provide powerful representations of protein sequences, understanding the underlying features and patterns learned by the models can be challenging. Studies on this can be done hiding part of the protein sequences, so to notice which amino acid are more influential for the prediction tasks. In this work we focused more on validating the built methods with unbiased benchmark strategies, which is a quicker

---

and efficient approach that would replace the lack of interpretability.

Moreover, DL-based embeddings are limited to the information contained within the protein sequence and may not fully capture other crucial aspects, such as protein structure and post-translational modifications, which are important for comprehensive protein analysis. On this purpose we suggested to couple the AI-based predictors with other experiment, either computational or in the wet lab.

Despite these limitations, DL-based protein embeddings offer valuable insights and opportunities in protein research, but careful consideration of these limitations is essential for their effective and reliable application.

## 11.7 Limits of the transformer-based algorithm in bioassays semantification

In the realm of biological data and knowledge bases, the adoption of Semantic Web technologies and knowledge graphs has become increasingly prevalent for tasks such as data integration, retrieval, and federated queries. In **Chapter 8** and **Chapter 9**, it is presented a solution for the automated semantification of biological assays, an important step towards enhancing the organization and understanding of biological data.

In particular, **Chapter 9** distinguishes between two contrasting methods: labeling and clustering, which represent opposite ends of the method complexity spectrum. Surprisingly, the results demonstrate that the clustering solution outperforms a state-of-the-art labeling approach based on deep neural networks. This finding is significant for two reasons. Firstly, it suggests that a learning objective closely aligned with the inherent characteristics of the data can yield superior performance compared to a more complex semantic modeling approach. Secondly, our automated semantification of biological assays achieves an impressive  $F_1$  score of approximately 83%, which, to the best of our knowledge, represents the first reported standardized evaluation of this task and establishes a robust benchmark model.

The success of the clustering approach highlights the potential of leveraging pattern identification techniques in ML for bioassay semantification. By grouping bioassays based on similarities in their text descriptions, we observed that clusters exhibited similar semantic representations, indicating the feasibility of semantifying an entire cluster using a standard set of labels. This finding has important implications for reducing computational complexity and accelerating the semantification process.

In conclusion, the works presented in this thesis demonstrate the diverse applications of Semantic Web technologies and machine learning in the realm of biological data analysis. While deep learning-based approaches have shown remarkable performance in various tasks, such as image recognition and natural language processing, the findings in **Chapter 9** emphasize that DL-based approaches are not always the best choice. The success of the clustering solution highlights the importance of considering the inherent characteristics of the data and leveraging pattern identification techniques. This highlights the need for a balanced approach that combines the strengths of different methodologies, including both deep learning and traditional techniques, to address the complexity and heterogeneity of biological data

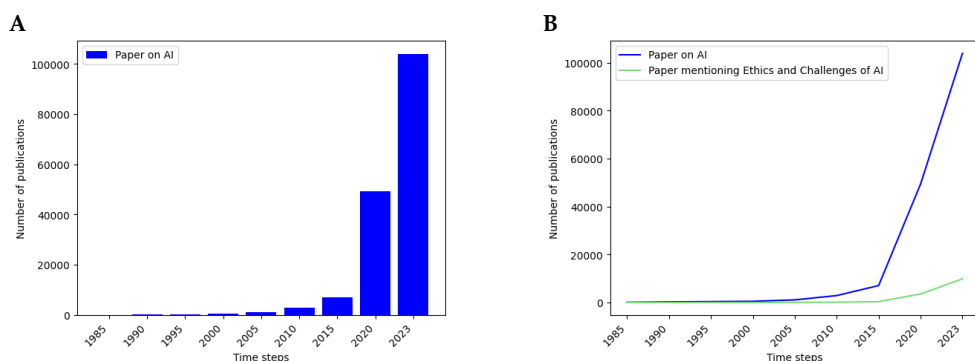


Figure 11.5: **A.** The trend of the number of papers published in a specific time range mentioning one of the following words: 'Artificial Intelligence', 'Natural Language Processing', 'Machine Learning' or 'Deep Learning'. **B.** Trend in the number of papers published in a specific time range mentioning one of the following words: "Artificial Intelligence", "Natural Language Processing", "Machine Learning" or "Deep Learning". The blue line shows the general trend in the number of papers mentioning these words, the green line shows those that are also associated with the words: "Ethics", "Ethic" or "Challenges".

effectively. By exploring and embracing a variety of approaches, we can pave the way for further advancements and insights in biological data semantification and analysis.

## 11.8 The lack of guidelines in AI-related research

In the **Introduction (Chapter 1)**, it is shown (Figure 1.4) the rapid rise of publications concerning AI and related disciplines. In **Chapter 10**, Figure 10.5 compares the trend of the rise of publications in AI with the increase in publications that consider the related ethical implications and challenges. A comparison is shown in Figure 11.5.

While AI continues to evolve rapidly, the ethical implications and potential risks associated with its deployment remain underexplored. The absence of well-defined guidelines and scholarly literature on this subject is concerning, as it leaves researchers, developers, and policymakers navigating through uncharted territories without a clear framework for addressing ethical dilemmas and potential societal consequences. The interdisciplinary nature of AI necessitates a comprehensive understanding of the ethical, social, and legal dimensions, highlighting the importance of robust guidelines and in-depth research that systematically explore the ethical challenges posed by AI systems. By establishing comprehensive guidelines and fostering research on AI ethics, we can ensure responsible and accountable development, deployment, and use of AI technologies, mitigating potential risks and promoting a more inclusive and beneficial AI landscape for society as a whole.

One effective approach to tackling the lack of guidelines in AI-related research is by starting with training the young generations in bioinformatics and scientific edu-

---

cation. By providing comprehensive education on the ethical challenges posed by AI systems to students at an early stage, we can cultivate a generation of scientists who are well-prepared to address these challenges in their future research and development. This can be achieved through dedicated training programs, such as the ones discussed in Chapter 10, that not only introduce students to emerging fields like bioinformatics but also emphasize the societal aspects of science and the importance of responsible AI practices. By instilling ethical considerations and critical thinking skills in the minds of young scientists, we can ensure that they are equipped with the knowledge and tools to navigate the ethical complexities of AI and contribute to the development of robust guidelines and ethical frameworks. Moreover, fostering interdisciplinary collaborations between bioinformatics, AI researchers, ethicists, policymakers, and other relevant stakeholders can facilitate the exchange of ideas, promote ethical discussions, and lead to the development of comprehensive guidelines for AI-related research and deployment. Through these combined efforts, we can address the lack of guidelines and promote responsible and ethical AI practices for the benefit of society as a whole.

## 11.9 Promoting bioinformatics for the next generations

By introducing students to emerging biology-related fields, such as bioinformatics, and emphasizing the societal aspects of science, we can foster a generation of scientists who are not only technically skilled but also well-versed in the ethical considerations surrounding AI. The training school format presented in Chapter 10 serves as a scalable and reusable model that not only promotes high-quality education in bioinformatics but also encourages critical thinking, gender equity in science, environmental awareness, and responsible AI practices. By integrating these efforts, we can ensure that the next generation of scientists is equipped with the knowledge and tools needed to navigate the ethical challenges and societal impact of AI technologies.

**Chapter 10** focuses on defining the optimal format for informing and engaging young students in emerging biology-related fields. The format, typically spanning five days, offers an opportunity to introduce students to a comprehensive overview of bioinformatics and the various scientific career pathways available to them.

The project builds upon the result of a survey conducted in 2021 by the non-profit organization Bioinformatika - Bioinform (<https://bioinform-org.github.io/>). In the sample of 105 students, 31 (29.5%) respondents declared a preference in studying natural sciences or mathematics/computer science. Despite the interest in these fields, only 23 (21.9%) of the respondents indicated that they are significantly promoted at their school or university, and 57 (54.3%) declared that they do not have an opportunity to hear about the practical uses of these fields at least once every 6 months. A positive outcome of the survey was that 57 (54.3%) respondents expressed interest in attending bioinformatics training schools.

These numbers demonstrate the need to promote science and innovation and this can also be applied to other areas where we face a lack of scientific promotion. To cover this, the training school format presented in **Chapter 10**, is aimed to be

scalable and re-usable in other areas by any interested trainer. The aim of the project is therefore to promote high quality free-education in bioinformatics.

## 11.10 Conclusion

In conclusion, this thesis aimed to investigate, develop, explore, and compare the practicality and effectiveness of computational approaches in the field of peroxisomal-related research. The primary objectives were to adapt deep learning-based embedding techniques to specific prediction tasks focused on peroxisomal research, as well as to define strategies that promote bioinformatics knowledge among future generations while incorporating critical thinking skills.

Throughout this study, I have successfully developed valuable tools that enable accurate prediction of peroxisomal (and mitochondrial) proteins. These tools leverage advanced computational methods and deep learning algorithms to enhance our understanding of peroxisomal (and mitochondrial) biology and facilitate the identification and characterization of these vital cellular components.

Also, the predictor tools have been expanded to protein-protein interaction predictions and transporter proteins predictions, thus showing the effectiveness of DL-based protein embedding representation for various purposes.

By employing state-of-the-art techniques and leveraging large-scale data analysis, we have made significant strides in the prediction of peroxisomal proteins. Our findings provide important insights into the computational approaches that can be employed in peroxisomal research paving the way for further advancements in the field.

The improvements generated from this research have far-reaching implications for peroxisomal research. The development of accurate prediction tools enables researchers to identify potential peroxisomal proteins with greater precision, facilitating the characterization of peroxisomal functions and pathways. This knowledge can shed light on the roles of peroxisomes in cellular processes and contribute to a deeper understanding of their involvement in various biological phenomena. Additionally, the utilization of deep learning-based approaches and computational methods enhances the efficiency and scalability of peroxisomal research, allowing for the analysis of large datasets and the exploration of complex biological systems. These advancements pave the way for further discoveries and advancements in peroxisomal research, ultimately leading to new insights, therapeutic targets, and potential treatments for peroxisomal-related disorders.

Moreover, the methodology used in this research can be scaled to various other fields involving protein sequence representation through DL-based embeddings. This scalability allows for the application of similar computational approaches to different subcellular compartments and organelles, expanding our ability to predict protein localization and function across diverse cellular contexts. By integrating multi-omics data and leveraging the power of computational methods, researchers can gain comprehensive insights into cellular dynamics and regulatory networks, leading to a deeper understanding of biological processes at a systems level.

Furthermore, this thesis recognizes the importance of nurturing the next generation of bioinformaticians and scientists. We have proposed strategies that focus on



---

promoting bioinformatics knowledge and fostering critical thinking skills among aspiring researchers. By empowering young minds with the necessary tools and knowledge, we can ensure the continued progress and success of peroxisomal research in the years to come.



---

# Summary

The objectives of the thesis were as follows: 1. To explore and compare the practicality and effectiveness of computational approaches in various settings related to peroxisomal research; 2. To adapt deep learning (DL)-based embedding techniques for specific prediction tasks in peroxisomal research; 3. To develop strategies focused on promoting bioinformatics knowledge and critical thinking skills among the next generation of researchers.

The primary objective has been achieved through several key accomplishments. Firstly, pipelines were defined to perform peroxisomal and sub-peroxisomal protein localization (**Chapter 2**). Secondly, a semi-automatic system for bioassay semantification was developed, which is not only useful for peroxisomal research but also scalable to all bioassays (**Chapter 8** and **Chapter 9**). Additionally, the chapters in between (3-7) overlapped with the achievement of objective 2. These chapters involved comparing and concatenating DL-based protein sequence embeddings, as well as classical protein encoding approaches, to predict the sub-peroxisomal or sub-mitochondrial localization of proteins (**Chapter 3**). The methodology for PTS1 signal detection was expanded and validated, and a web server was developed to facilitate sub-organelle localization prediction tasks (**Chapter 4**). The methodology for the mPTS signal was also expanded and validated (**Chapter 5**). Furthermore, the capabilities of DL-based protein embeddings were explored in predicting peroxisomal protein interactions (**Chapter 6**). Lastly, the DL-based approach was scaled for predicting transporter proteins, with a specific focus on a subset of peroxisomal proteins (**Chapter 7**).

Finally, the third objective was accomplished by developing an improved method for designing training schools in bioinformatics (**Chapter 10**).

The thesis starts with the **Introduction (Chapter 1)** which contextualizes the research carried out in this work. First, it shows some descriptive statistics that highlight the relevance of Artificial Intelligence in Bioinformatics and other Biomedical disciplines. The research explains the parallel between AI applied to text files and biological sequences. This parallel is then discussed in the scope of the thesis, starting with the explanation of DL-based sequence embeddings in the **Introduction (Chapter 1)**, showing their potential applications in the following chapters, and highlighting how similar DL algorithms can be applied to text file semantification.

From a molecular biology perspective, the main subject of this thesis is the peroxisome. This ubiquitous organelle is described in the **Introduction 1**, but more details about current research on peroxisomes are available in chapters 3,4,5,6,7. In this context, the peroxisome is a particular use case that shows how DL-based embeddings can be adopted in highly specific frameworks, ultimately improving the computational approaches in peroxisomal-related research.

**Chapter 2** provides an overview of the computational methods and tools that can be used to investigate peroxisomal proteins and their functions. It begins by discussing the importance of computational approaches in peroxisomal research, particularly in the detection and localization of novel peroxisomal proteins. It also provides a detailed description of various computational methods, including sequence-based methods, structure-based methods, and machine learning-based methods. The chapter then discusses the limitations of these methods and highlights the need for more accurate and efficient approaches. To address this need, a hybrid approach is proposed that combines multiple computational methods to improve the accuracy of peroxisomal protein detection and localization. Overall, this chapter emphasizes the importance of computational approaches in peroxisomal research and provides a comprehensive overview of existing methods. It also highlights the potential for future research to develop more advanced computational tools that can provide new insights into peroxisomal biology.

**Chapter 3** presents a detailed analysis of the In-Pero algorithm, which is a deep learning-based approach for predicting the localization of sub-peroxisomal proteins, specifically proteins that can be located inside the peroxisomal matrix or in the peroxisomal membrane. The problem of predicting the sub-localisation of peroxisomal proteins is addressed in this chapter using a computational strategy that combines protein-sequence embeddings with classical machine learning. Additionally, the approach is tested for sub-mitochondrial localization, resulting in a predictor (In-Mito) that outperforms most of the existing classifiers. The chapter begins with an overview of the dataset used to train and test the In-Pero algorithm. Overall, the effectiveness of deep learning-based approaches for predicting sub-peroxisomal proteins' localization is demonstrated in this chapter, along with its potential applicability to other organelles such as mitochondria.

**Chapter 4** presents the OrganelX e-Science Web Server, which is a user-friendly implementation of several deep learning-based algorithms for predicting sub-peroxisoma. The novel Is-PTS1 algorithm, used to detect potential peroxisomal proteins carrying a PTS1 signal sequence, is also introduced. Access to these algorithms and analysis of protein sequences can be done through the OrganelX e-Science Web Server, which provides a user-friendly interface. A detailed description of how to use the web server and interpret its results is also provided. In this chapter it is demonstrated the usefulness of the OrganelX e-Science Web Server in predicting the subcellular localization of proteins. Additionally, it is highlighted the potential for future research to develop more advanced computational tools that can provide new insights into complex biological processes.

**Chapter 5** focuses on the detection of the membrane peroxisomal targeting signal (mPTS), which plays a crucial role in the targeting of proteins to peroxisomes. Accurately identifying mPTSs in protein sequences is critical for understanding peroxisomal protein targeting and dysfunction, which can lead to various diseases. This chapter, presents a novel algorithm called Is-mPTS, which accurately detects mPTSs in protein sequences. The algorithm is based on a combination of an experimentally validated score and a novel method that relies on deep learning-based protein sequence embeddings to predict whether an mPTS is true or false. This approach provides a significant improvement over existing methods that rely on traditional sequence-based analysis, which can be less accurate and less efficient. Is-mPTS was

---

evaluated on a large dataset of protein sequences and demonstrated its superior performance compared to existing methods. Is-mPTS can be a valuable tool for identifying mPTSs in protein sequences, facilitating the study of peroxisomal protein targeting and dysfunction, and providing a valuable resource for researchers studying peroxisomal biology.

**Chapter 6** proposes a novel approach (in line with the ones presented in the other chapters) for protein-protein interaction prediction. Here, it is presented an algorithm called P-PPI, which is fine-tuned for peroxisomal protein-protein interaction prediction (thus the acronym). The chapter describes the usage of this algorithm to present potential candidates to be further checked either experimentally or with complementary computational approaches. In particular, it is reported a use case that highlights the synergy between P-PPI and the multimer predictor implemented within AlphaFold.

An additional application of DL-based sequence embeddings is presented in **Chapter 7**, where it is shown the adaptation of the embeddings methods to predict transporter proteins and classify them according to their specific substrate. Here, the peroxisomal protein dataset is only used as proof of concept since the algorithm is perfectly scalable for every protein sequence. This novel method takes advantage of a recursive feature elimination step that reduces the dimension of four different concatenated embeddings. Overall, this chapter demonstrates the effectiveness of deep learning-based embeddings and their adaptability for predicting transporter proteins and their substrate category, even for very specific use cases such as with peroxisomal proteins.

**Chapter 8** and **Chapter 9** are examples of how similar technologies can be applied and tested for different inputs and even topics. The change of topic here is strong but useful since it shows how a similar technology can be applied to different cases. The knowledge acquired from this application was then used to further improve the predictive algorithms herein developed.

**Chapter 8** focuses on the application of a SciBERT-based model (similar to DL-based sequence embeddings algorithms) for semantifying biological assays. The SciBERT model is a pre-trained language model based on the Bidirectional Encoder Representations from Transformers (BERT) architecture, which has been fine-tuned on scientific text to improve its performance on scientific tasks. The use of SciBERT in the bioassay semantification procedure helped to improve the accuracy and efficiency of the process by enabling the algorithm to better understand and interpret scientific text. The chapter begins by discussing the importance of representing biological assays in a standardized and machine-readable format to facilitate data integration and knowledge discovery. The Open Research Knowledge Graph (ORKG) is also described. The ORKG is an open platform for sharing and discovering scientific knowledge, and the reported novel method is used as a prototype version for representing biological assays in the ORKG.

**Chapter 9** examines the computational aspects of the automated semantification of biological assays (bioassays) by considering two different approaches and evaluating their effectiveness. The first approach is based on a transformer-based supervised classifier (SciBERT), as discussed in Chapter 8, which formulates a labeling objective. The second approach focuses on clustering bioassays for semantification. While both techniques share similarities, labeling involves assigning predefined la-

bels to objects, whereas clustering identifies similarities between objects and groups them based on common characteristics. In the chapter it is shown that bioassays with similar text descriptions also exhibited similar semantic representations. Therefore, fine-grained clustering of assays could enable the semantification of entire bioassay clusters using a standardized set of labels. While a classifier requires multiple passes to fully label an assay, a clustering model can semantify clusters in just one pass over the data. Through the reported experiments, it is shown that labeling and clustering yield contrasting scores and time requirements. Surprisingly, the powerful transformer-based labeling method exhibited lower accuracy compared to the clustering solution (54% F1 vs. 83% F1), and labeling with a large set of labels took significantly more time due to per-label classifications.

**Chapter 10** highlights the importance of training schools in bioinformatics as a means to inform and engage young students in biology-related fields. The five-day format of these training schools offers an extensive overview of bioinformatics and scientific career opportunities, catering to three distinct target groups: high school students, undergraduate students in biology-related fields, and undergraduate students in computational fields. The content and sessions are structured around specific learning areas, with defined learning outcomes and activities. Furthermore, the concept of a teaching platform is introduced, aiming to manage FAIRyfyed (Findable, Accessible, Interoperable, Reusable) training materials, enabling the design of new training schools in bioinformatics. The importance of promoting science and engaging students at a pre-university level is emphasized, as it can significantly influence their decision to pursue a scientific career.

The thesis ends with a **Discussion** section (**Chapter 11**) about the improvements that this set of scientific contributions brought to the scientific community. It is also presented an in-depth discussion about challenges and limitations of the current approaches adopted in artificial intelligence applied to biomedical sciences in general, and specifically in the protein sequences representation and peroxisomal research. These limitations are also discussed within the training sessions explained in Chapter 10, where the trainers aims to raise awareness about these topics for the next generations of scientists from an early stage of their careers.







---

# Acknowledgements

You always used to say things like, "I hope I live long enough to see you start school". Of course you knew you would see me. You uttered this phrase at every one of my milestones: first high school, then my bachelor's degree, and later my master's. You witnessed them all and provided me with strength through each one. Unconditional love, where it didn't matter what I did as long as I was happy doing it. This most important achievement of mine, I dedicate to you. Goodbye, Grandma.

I thank my family for supporting and enduring me during these years of study and work abroad. You've built highways and dug tunnels through high mountains for me.

I extend my gratitude to scientists I had the honor to collaborate with during these years, from whom I learned a great deal. Starting with my supervisors, Edo and Vitor, and continuing with my colleagues from PerICo, LifeGlimmer, and Wageningen University.

I thank my colleagues from the latter part of my doctoral journey and Tim, who, during my short stay at the Zuse Institute in Berlin, helped me grow.

I am especially grateful to my colleagues with whom I founded the NGO Bioinformatika-Bioinform. Selle, who brought the energy and skills necessary to realize our ambitions. Katarina, with whom this idea was born, and with whom I shared all my doubts and concerns, and who helped me overcome enormous obstacles during most of these doctoral years.

I extend my heartfelt thanks to my paranympths. Sonja, my colleague and friend with whom I shared the most experiences and opinions, and Francesco, the person with whom I've shared the most in life, starting with our education.

I thank Francesca for reassuring me as I neared the end of this journey and for lending her expertise to create infographics and the cover.



---

# Ringraziamenti

Scherzando, dicevi sempre cose del tipo "Spero di campare fino a che non inizi le scuole". Hai pronunciato questa frase ad ogni mio traguardo: prima il liceo, poi la laurea triennale, poi la magistrale. Li hai visti tutti e mi hai dato forza in ognuno. Un amore incondizionato, per cui non importava cosa facessi, purché fossi felice di farlo. Questo mio traguardo, il più importante, lo dedico a te. Ciao nonna.

Ringrazio la mia famiglia per avermi supportato e sopportato in questi anni di studio e lavoro all'estero. Mi avete aperto un'autostrada e scavato gallerie in alta montagna.

Ringrazio li scienziati con cui ho avuto l'onore di collaborare in questi anni, e da cui ho imparato molto. A partire dai miei supervisori, Edo e Vitor, per proseguire con i miei colleghi di PerICo, di LifeGlimmer e dell'Università di Wageningen.

Ringrazio i colleghi della seconda parte di questo dottorato e Tim, che allo Zuse Institut Berlin mi hanno fatto crescere anche se per poco tempo.

Ringrazio le mie brillanti colleghe con cui abbiamo fondato la NGO Bioinformatika-Bioinform. Selle, che ha portato energia e competenze necessarie per realizzare le nostre ambizioni. Katarina, con la quale questa idea è nata, e con la quale ho condiviso ogni mio dubbio e preoccupazione, e che mi ha aiutato a superare ostacoli enormi nella maggior parte di questi anni di dottorato.

Ringrazio particolarmente i miei paraninfi. Sonja, la mia collega e amica con cui ho condiviso più esperienze e opinioni, e Francesco, la persona con cui ho condiviso più cose nella vita, a partire dalla formazione.

Ringrazio Francesca per avermi tranquillizzato nella chiusura di questo percorso, nonché per avermi prestato le sue competenze per realizzare infografiche e la copertina.



---

# List of Publications

*This thesis:*

**Marco Anteghini**, Jennifer D'Souza, Vitor AP Martins dos Santos, Sören Auer (2020). "Representing semantified biological assays in the open research knowledge graph". *Digital Libraries at Times of Massive Societal Transition: 22nd International Conference on Asia-Pacific Digital Libraries, ICADL 2020, Kyoto, Japan, November 30–December 1, 2020, Proceedings* 89-98.

**Marco Anteghini**, Vitor AP Martins dos Santos, Edoardo Saccenti (2021). "In-Pero: Exploiting deep learning embeddings of protein sequences to predict the localisation of peroxisomal proteins". *International Journal of Molecular Sciences*, 22(12), 6409.

**Marco Anteghini**, Jennifer D'Souza, Vitor AP Martins dos Santos, Sören Auer (2021). "Easy semantification of bioassays". *AIxIA 2021–Advances in Artificial Intelligence: 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1–3, 2021, Revised Selected Papers*, 198-212.

**Marco Anteghini**, Asmaa Haja, Vitor AP Martins dos Santos, Lambert Schomaker, Edoardo Saccenti (2023). "OrganelX web server for sub-peroxisomal and sub-mitochondrial protein localization and peroxisomal target signal detection." *Computational and Structural Biotechnology Journal* 21,128-133, Elsevier.

**Marco Anteghini**, Vitor AP Martins dos Santos, Edoardo Saccenti (2023). "PortPred: exploiting deep learning embeddings of amino acid sequences for the identification of transporter proteins and their substrates." *bioRxiv*.

**Marco Anteghini**, Vitor AP Martins dos Santos (2023). "Computational Approaches for Peroxisomal Protein Localization". *Peroxisomes: Methods and Protocols*, 405-411, Springer US.

*Others:*

**Marco Anteghini**, Jennifer D'Souza, Vitor AP Martins dos Santos Vitor AP; Sören Auer (2020). "SciBERT-based semantification of bioassays in the open research knowledge graph." *arXiv preprint arXiv:2009.08801*.

Arghadwip Paul, Suman Samantray, **Marco Anteghini**, Mohammed Khaled, Birgit Strodel (2021). "Thermodynamics and kinetics of the amyloid- $\beta$  peptide revealed by Markov state models based on MD data in agreement with experiment." *Chemical science*, 12(19), 6652-6669.

Castrense Savojardo, Pier Luigi Martelli, Giulia Babbi, **Marco Anteghini**, Matteo Manfredi, Giovanni Madeo, Emidio Capriotti, Jumamurat R Bayjanov, Margherita Mutarelli, Rita Casadio (2021). "SB4ER: An elixir service bundle for epidemic response." *BioHackrXiv*.

Maki Kamoshita, Rechal Kumar; **Marco Anteghini**, Markus Kunze, Markus Islinger, Vitor AP Martins dos Santos, Michael Schrader (2022). "Insights into the peroxisomal protein inventory of zebrafish." *Frontiers in Physiology* 322.

Jennifer D'Souza, Anita Monteverdi, Muhammad Haris, Muhammad, **Marco Anteghini**, Kheir Eddine Farfar, Markus Stocker, Vitor AP Martins dos Santos, Sören Auer (2022). "The Digitalization of Bioassays in the Open Research Knowledge Graph." *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I* 63-68.

Cláudio Figueiredo Costa, Celien Lismont, **Marco Anteghini**, Iulia Revenco, Hongli Li, Marc Fransen, Iria Medraño-Fernández, Roberto Sitia, Patrick Bienert (2022). "The Mystery behind Hydrogen Peroxide Permeation across the Peroxisomal Membrane". *CRC Press/Taylor & Francis*.







---

# Overview of the completed training activities

## Discipline-specific activities

PERICO Kick-off Event	RUG	2019
AIRBUS GEO CHALLENGE (Final)	AIRBUS DS GEO	2019
FUTURE OF MEDICINE	Berlin Institute of Health	2019
PERMED/INFECT (project meeting)	Karolinska Institute	2019
PERICO MIDTERM MEETING (network meeting)	KU Leuven	2020
PERICO MEETING (network meeting)	RUG	2020
CROSS4HEALTH Final event	Norway Health Tech	2020
EKAU conference	University of Bozen	2020
BBCC conference	University Federico II Naples	2020
ICADL conference	ICADL	2020
FUTURE OF MEDICINE	Berlin Institute of Health	2020
Integrated Modeling and Optimization	BioSB research school	2020
Innovation for health conference	WTC Rotterdam	2021
BioSB	BioSB research school	2021
RECOMB 2022	UCSD	2022
BioSB	BioSB research school	2022

## General courses

TRAINING SCHOOL A	UMCG	2019
TRAINING SCHOOL B	Weizman institute	2019
Scientific writing and peer review	Max Plank institute	2020
VLAG online lecture series	WUR	2020-2021
TRAINING SCHOOL E	BioFaction	2020
TRAINING SCHOOL C	AMC Amsterdam	2020
Career Perspective	WGS	2020
Elixir BioHackathon Europe	Elixir Europe	2020
Popular science writing	WGS	2021
BioBusiness winter school	BCF Courses BV	2021
BioEthics	FU Berlin	2021
Elixir BioHackathon Europe	Elixir Europe	2021
Elixir BioHackathon Europe	Elixir Europe	2022

## Other activities

SSB Workgroup meeting	SSB	2019-2023
Writing Research Proposal	SSB	2019
PhD Trip	SSB/MIB	2022
Bioinforming training school (trainer)	NVO – Bioinformatika	2023



