Protein Classification from Primary Structures in the Context of Database Biocuration

Ilmira Terpugova

Advisors: Alfredo Vellido Alcacena, Enrique Romero Merino

Department of Computer Science

MASTER IN ARTIFICIAL INTELLIGENCE
FACULTAT D'INFORMÀTICA DE BARCELONA (FIB)
FACULTAT DE MATEMÀTIQUES (UB)
ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA (URV)
UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) - BarcelonaTech
UNIVERSITAT DE BARCELONA (UB)
UNIVERSITAT ROVIRA i VIRGILI (URV)

April, 2017

Acknowledgements

I would like to thank my advisors Alfredo Vellido Alcacena and Enrique Romero Merino for their kind and continued help, support and guidance without which this work would not have been possible.

Abstract

The problem of automatic protein classification using only their primary structures plays an important role in modern bioinformatics research, especially for proteins whose 3-D structures are yet unknown. One of these types of proteins, at the center of this thesis, is class C of the G-Protein Coupled Receptors super-family. This class is of a great interest in pharmaco-proteomics, from the point of view of drug design, because of their involvement in signaling pathways in cells of the central nervous system. The automatic classification of protein sequences may improve the understanding of their function and be a basis for the prediction of their 3-D structure, which is an information of interest for drug research.

This thesis compares classification results for different versions of the same database, including the most recent ones. This exploration of the evolution of classification provides relevant information about its capabilities and limitations. Furthermore, and given that several data transformations are investigated, it also provides strong evidence concerning the robustness of these transformations.

The other important contribution of the thesis is the investigation oriented towards the definition of approaches for semi-automatized database curation by using the automatic evaluation of the database changes between versions with advanced machine learning techniques. The thesis shows the consistency in improvements of the quality of the data between three versions of the database across different classification techniques and different primary structure transformations. It also validates the recently introduced continuous distributed representation for protein sequences, originally developed for natural text processing. This new representation is shown to be adequate and robust for the task of primary structure classification.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One of the most challenging problems in the field of bioinformatics in recent years and probably one that will only get more complex in the near future is the tremendous growth of experimental data availability. The reduction rates in the costs of the sequencing processes, compounded with their increasing technological sophistication, are significantly beating the rates of computation power growth, which follows Moore's law [National Human Genome Research Institute, 2016].

The number of genomes and other biomolecules being sequenced also follows an exponential trend of increase. In proteomics in particular, the context of the current thesis, data availability is not growing at a comparable rate but, likewise, the improvement in speed and the increasing accessibility of the required technologies have also led to a remarkable growth in this area [Richards et al., 2015].

This leads to all types of bio-data management challenges, including data storage and transmission, database public access, or data curation, to name just a few. And this is even before we consider the challenges related to the standardization of procedures for at least semi-automated data analysis oriented towards meaningful biological and medical knowledge extraction and interpretation.

This thesis is meant to contribute some tools and approaches to deal (or at least assist) with the problem of data curation in protein databases. "Manual" human-based (and therefore

not fully automated) data curation is an essential task in proteomics (as made evident by the many efforts devoted to the design and maintenance of large and publicly accessible protein databases), but the assumption according to which this thesis is built is that such task can be assisted by advances in semi-automated data analysis, based on statistical and machine learning methodologies. This assistance holds the promise of delivering gains in curation efficiency and reliability.

Many biological databases are curated and supported by research communities, networks and public-private partnerships, at a considerable cost. The work reported in this thesis, even if based on a specific publicly available and research network-managed database, was carried out independently and mainly from a Machine Learning research perspective. The results of this work and the knowledge herein generated, though, are meant to be of use for the database curation teams.

1.1 Motivation and Objectives

There are two main objectives for this thesis. The first one is to investigate the adequacy and potential of a newly proposed alignment-free transformation of protein primary structures (that is, a quantitatively analyzable transformation of the protein symbolic sequences). This can be accomplished by comparing the results of protein subtype classification obtained using this transformation with those obtained by previously successfully applied transformations, using similar classification procedures. The basic idea behind this recently defined transformation is treating the protein symbolic amino acid sequence as a kind of text in which "words" would correspond to protein motifs, thus taking advantage of methods originally developed for text processing and, in particular, for distributed word representation.

The second main objective of this thesis is to investigate the evolution of the main publicly accessible GPCR (G-Protein Coupled Receptors) proteins database. Previous research clearly showed evidence of subtype labeling problems in one of the classes of GPCR, namely Class C. Clearly and consistently misclassified proteins were found in earlier versions of the database,

along with well-defined maximum performance limits, which were shown to be independent of the choice of data transformation and classification method. This previous research was carried out using the 2011 version of the database, while new and greatly modified versions have recently been made available. The current thesis considers the 2011, May 2016 and September 2016 versions of the database. The aim is to perform an analysis of the changes of the consistency and quality of the data, of use to database curators, using Machine Learning techniques.

1.2 Structure

This document consist of six chapters, including the current introductory one.

In the second chapter, the context of the investigated problem is explained, covering the general terminology in proteomics, principles of GPCR structure and functionality and the description of the analyzed database.

The third chapter includes a self-contained overview of previous work in similar problems and for the particular problem of classification of class C proteins considered in the thesis.

Then, in the fourth chapter, a technical description of methods used in the thesis is presented.

The last two chapters cover the experimental settings description, main results and their discussion, as well as the conclusion and possible future ways of improvement and development of the current work.

Chapter 2

Background

This chapter is split into four main sections covering different aspects of the biological background of the problems at hand. They are by no means written with the aim of providing a complete overview of each of the topics, something that is well beyond the goals of the current thesis. Instead, the reader is provided with sufficient contextual background as to be able to appreciate the application interest of the reported research.

The first section is aimed at introducing the general biological background behind the problem, so as to shed some light on a few of the basic concepts used in proteomics. The second section explains the basic principles of GPCR's activation and describes the specific features of the class C of this type of proteins. In the third section, the specific database analyzed in the following chapters is summarily described, focusing on the important changes to its contents that can be observed through the three investigated versions. The last section introduces the problem of data curation in biological databases, also known as bio-curation. The fact that the thesis deals with database changes and adaptations over time should make its results useful from the viewpoint of data quality management.

2.1 Protein primary structure classification

The protein molecule is a polymer consisting of amino acid molecules. There are 20 different amino acids in the *protein alphabet*, usually encoded in symbolic form using the letters of the Latin alphabet. The number of amino acids in protein molecules varies widely, from a few dozens to over a thousand.

Four structural levels of the protein are commonly distinguished. The ***primary structure*** is simply a description of the linear sequence of amino acids in the protein and, as such, it conveys little explicit information about the 3-D crystal structure that the protein occupies in space. The ***secondary*** and ***tertiary structures*** determine such 3-D shape of the molecule and, consequently, its function. The ***quaternary structure*** also includes the bonds potentially established between several protein molecules.

The primary structure is the easiest one to obtain and, even if it conveys little explicit information about the 3-D crystal structure of the protein, it can still be used to predict it in the form of the 3-D molecule folding, in order to investigate protein functionality.

The primary structure is also used in such tasks as remote homology detection, i.e. finding genetic relations between the proteins of low sequence similarity and protein classification.

The current thesis covers only research that uses the available primary structures of the GPCR proteins, given that the 3-D crystal structure of most of the investigated receptors is currently still unknown.

2.2 G-protein coupled receptors

G-Protein Coupled Receptors (GPCR) are eukaryotic cell membrane proteins which are a part of cell signal system and serve to connect external stimuli to the intracellular signaling pathways.

GPCRs are also called 7TM receptors because they share a common structure of seven transmembrane helices (TM1-TM7), along with the extracellular N-Terminus and the intracellular

C-Terminus. The seven helices domain of the protein is linked by three intracellular and three extracellular loops named ICL1 to ICL3 and ECL1 to ECL3, respectively, counting from the N-terminus (See Fig. 2.1 for reference).

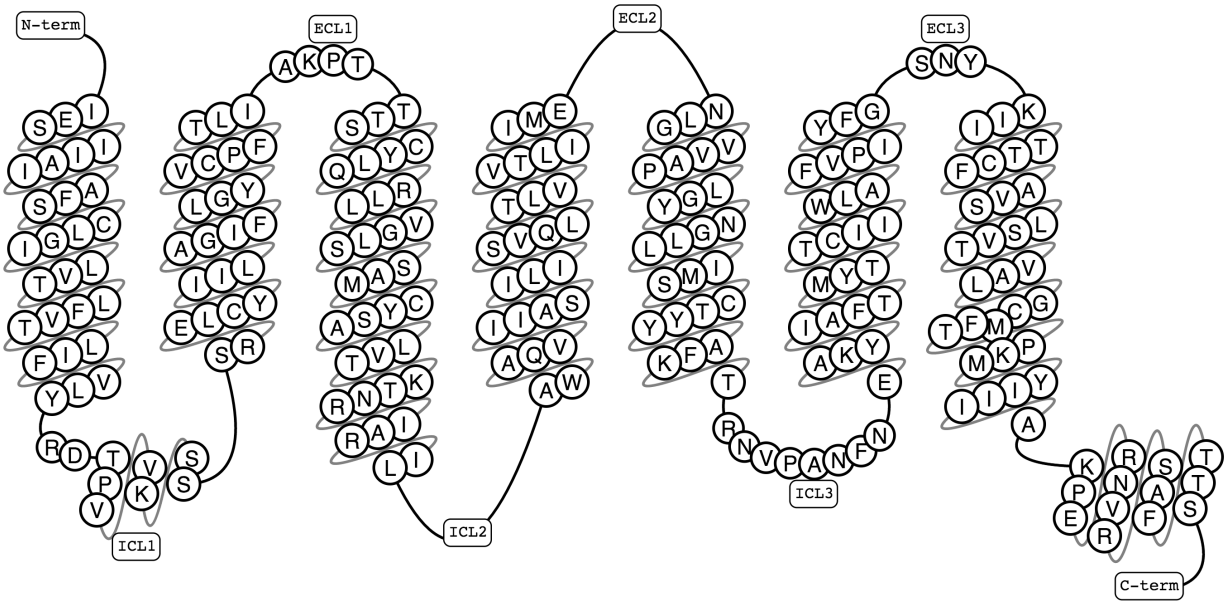

Figure 2.1: Basic graphical representation of the structure of a characteristic GPCR, namely the human metabotropic glutamate receptor 1.

The intracellular part of the receptor connects to a G protein inside the cell membrane.

The receptor activation works as follows:

- First, the external molecule, called ligand, binds to some part of the receptor (either in the N-Terminus or the 7TM domain), i.e. two molecules, receptor and ligand, create chemical bonds that allow conveying activation through the membrane.
- The binding process activates the receptor, i.e. changes its formation.
- Activation of the receptor causes the dissociation of the G protein, which in turns starts some signaling path inside the cell.

There are two types of ligand-binding: orthosteric and allosteric. The orthosteric binding causes the activation of the receptor, while the allosteric one can modify (modulate) the activation in different ways, while still allowing the orthosteric binding.

The allosteric processes of modulation are widely used in pharmacological drug design to reduce, increase or stop the activation of the protein by its natural ligand. Some of the drugs also use the orthosteric binding site and their activity is in this case based on substitution of the natural ligand [Lindsley et al., 2016].

Class C GPCRs

Class C of GPCRs gathers receptors that control action of the neurotransmitters glutamate and γ-aminobutyric acid, sweet and "umami" taste and calcium homeostasis [Wu et al., 2014].

For class C receptors, the orthosteric ligand binds in the usually large extracellular N-Terminus, which includes a structure known as Venus Flytrap (VFT) domain and a cystein-rich domain (CDR) that connects the VFT to the 7TM domain. The allosteric modulators in class C bind to the 7TM domain (See Figure 2.2).

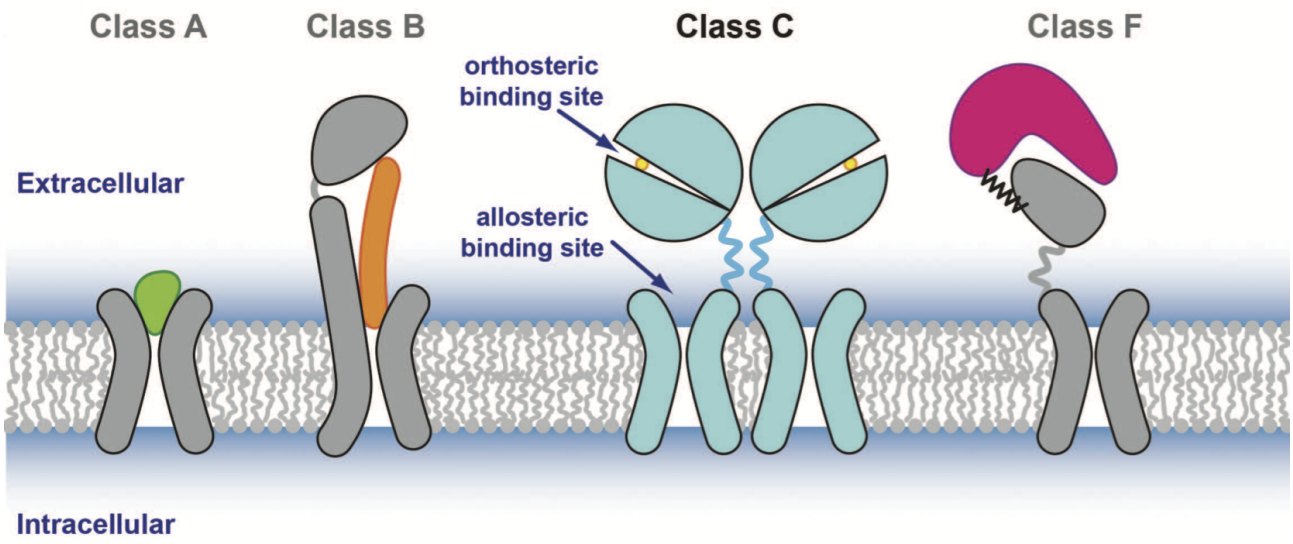

Figure 2.2: Graphic simplified representation of the differences in structure and binding sites of the GPCRs from different classes [Wu et al., 2014].

Considering their role in synaptic transmission in the central nervous system, class C receptors are important for drug discovery due to their role in the interaction with drugs for the treatment of psychiatric and neurological diseases such as schizoprenia, Parkinson's disease and depression,

amongst others [Gregory et al., 2012].

2.3 Data description

The GPCRdb [Munk et al., 2016, Isberg et al., 2016] is a publicly accessible repository of GPCR databases and web tools for the analysis of membrane proteins including about 400 human specimens.

Overall, the GPCRdb dataset contains 14,951 proteins from 3,184 species. At the time of writing, this resource has been available for over 20 years and its management was transferred in 2013 to Prof. David Gloriam's research group at the University of Copenhagen in Denmark.

Its classification of the available proteins follows the international IUPHAR system recommendations.

The whole database originally consisted of 7 classes: A (Rhodopsin), B1 (Secretin), B2 (Adhesion), C (Glutamate), F (Frizzled), Taste 2 and "other" GPCRs. This classification followed the system suggested in [Kolakowski Jr, 1993].

2.3.1 Class C GPCR data

At the highest level of grouping, class C discriminates the proteins according to the type of ligand as ion, amino acid, and sensory receptors.

At the second level of classification of the current database version, four subclasses are distinguished: metabotropic glutamate receptors (amino acid), $GABA_B$ (amino acid), calcium sensing (ion) and taste 1 receptors (sensory), covering sweet and umami tastes. The earlier version of the database (2011) included three extra sensory-related subclasses of the second level, namely vomeronasal, pheromone and odorant receptors.

As previously mentioned, class C proteins are characterized by the existence of a large extracellular domain, which behaves as a binding site for natural ligands, and by the transmembrane

domain (7TM) where allosteric modulators bind. This makes them promising drug targets and justifies the current extensive research they are subject to.

2.3.2 Database changes

This thesis covers three versions of the database: the first one from 2011, and two recent drastically changed versions: those of May 2016 and September 2016.

Over those five years, GPCRdb has undergone major changes in the total numbers of proteins belonging to class C, in the proportion between the different subclasses and even in the sequences contained in each of those subclasses (see Table 2.1 and Figure 2.3). Note that the database changed the whole management and development team in 2013.

Family	2011	May 2016	Sept 2016	2011 ∩ May 2016	May 2016 ∩ Sept 2016
MGluR	351	467	516	93 (26%)	357 (76%)
Calsium-sensing	48	125	103	10 (21%)	91 (73%)
$GABA_B$	208	60	89	10 (5%)	50 (83%)
Taste 1	65	193	228	42 (65%)	129 (67%)
Orphans	147	193	18	0	18 (9%)
Total				155	647

Table 2.1: Number of proteins in each subclass in the different datasets.

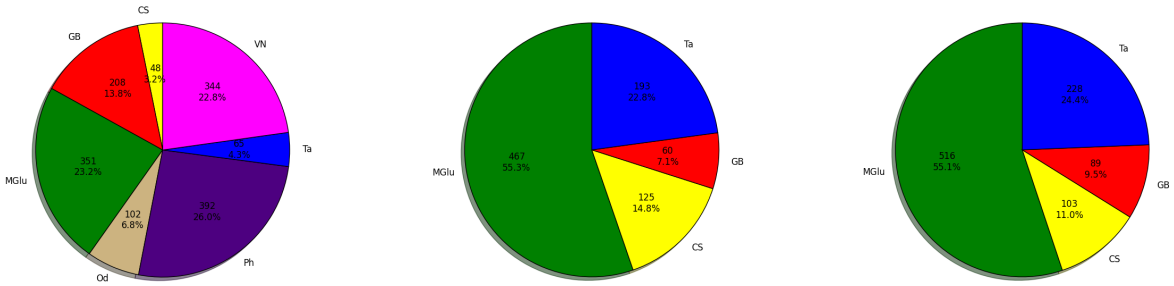

Figure 2.3: Class distribution for databases
left - March 2011, middle - May 2016, right - September 2016

The main changes happened switching from the 2011 version to the one of May 2016, with only 155 protein sequences remaining unchanged. Not only the receptors of three subclasses (vomeronasal, odorant and pheromone) were removed in full from class C, but the number of proteins in the other remaining subclasses also changed significantly.

The Metabotrobic Glutamate (MGluR) Receptors subclass grew by 33% and only 26% of sequences were kept unchanged ("2011 ∩ May 2016" column in table 2.1). The Calcium-Sensing Receptors subclass increased more than twice, keeping only 10 sequences unchanged. Finally, the Taste 1 subclass grew threefold, while the $GABA_B$ receptors subclass, on the contrary, decreased more than threefold.

The changes between the two 2016 versions are not that substantial, but still significant for a mere four-month period. In this case, the number of sequences kept completely unchanged varies from 65% to 85% for the four subclasses. The MGluR subclass kept growing in the September 2016 version by 10%; the $GABA_B$ and Taste 1 also increased in number by 50% and 18% respectively. Instead, Calcium-sensing decreased by 18%. The largest of differences, though, is in the number of orphan receptors (those not assigned to a subclass). Less than 10% of the original orphans were kept in the last versions.

2.4 Data curation

In many of the databases used for data analysis tasks with Machine Learning and statistical tools, data are provided 'as is' i.e. without explicit and/or complete guarantees of their validity from different viewpoints: consistency, artificats and errors, data missingness, or label reliability, to name a few.

However, biological information in its current form, and mainly in the -omics sciences, is usually curated by specially assigned professional scientists. This is a key issue, given that expert-curated knowledge bases that are accessible using web-tools are becoming one of the main driving forces in contemporary research in biology in general and bioinformatics in particular.

The responsibilities of curators include data collection, consistency and accuracy control, annotation using widely accepted nomenclatures, collection and standardization of results published in scientific literature, evaluation of computational analysis, etc.

Data curation in biological research (sometimes known as biocuration) requires broad expertise

in the domain because of the vast amount of information available from literature, which is usually not without its controversy and errors, and often lacking a unified and standardized approach for the representation and analysis of data.

Biocuration has been described as "the activity of organizing, representing and making biological information accessible" [Howe et al., 2008] to biologists. This activity, as a result, concerns data engineering tasks. Note though that it is far from being an established and well-defined activity and failure to establish and standardize biocuration procedures and to fund these efforts properly may hamper the complete process of scientific knowledge extraction.

One of the challenges of curation is the unambiguous identification of biological entities (proteins in the case of this thesis) from existing studies and literature. In the end, data trustworthiness can only be ensured through costly data management [Lord et al., 2004]. This task, when understood as "manual" expert curation, is uncertain and error-prone, so that the development of computational procedures to assist human experts in it is worth pursuing.

Note that GPCRs have been categorized into different classes and subclasses on the basis of sequence homology procedures [Isberg et al., 2016]. That is, receptor labeling is itself homology model-based and, therefore, uncertain to a degree and at least debatable.

According to these concerns, this thesis provides an analysis of class annotation in the GPCR protein database performed using Machine Learning techniques. It is based on three different version of the GPCRdb from different time periods and then can be considered from viewpoint of a cross-check of the quality and consistency of the database.

Chapter 3

Previous Work

Existing approaches to the problem of protein classification are commonly based on the concept of similarity between sequences. This chapter provides an overview of the most efficient and widely spread protein similarity measures used in this problem, along with the corresponding classification techniques that make use of these measures.

The first section gives an overview of the similarity techniques successfully applied in previous research for classification and homology detection for a wide range of proteins.

In the second section commonly used classification techniques for these tasks are considered.

The last section covers the classification of the G-protein coupled receptors of class C in particular, providing also a summary of the results achieved in previous research in this specific domain.

3.1 Protein sequence similarity measures

Two main groups of protein similarity measures can be distinguished in biological research. The first group includes those based on sequence alignment methods, which were designed to find regions with functional and structural relationships between two or more sequences. Protein sequence alignment methods have long history and are more conventional in biology than the

second group of protein similarity measures, which use different mechanisms and are usually referred to as "alignment-free" similarities because they do not have alignment of the sequence as a pre-requisite.

Alignment methods go back to Needleman-Wunsch algorithm (1970), its later variation Smith-Waterman (1981), and their faster, less accurate but extremely popular counterpart BLAST (1990)[Altschul et al., 1990]. The Needleman-Wunsch is a global alignment method for the whole sequences. The latter two are local alignment algorithms which compare subsequences of all possible lengths. The BLAST algorithm is extremely popular and ubiquitous and its original publication has been cited over 60,000 times at the time of writing this thesis.

However, and despite the popularity of alignment techniques, there is also intense research in the investigation of alignment-free sequence similarities. The four groups of similarities are considered below: family-based generative models, motif-based similarities, similarities based on amino acid composition transformation and a heterogeneous group of techniques which use physico-chemical properties of amino acids and spatial sequence information.

The first group of models to mention is family-based generative models. What these models have in common is that they are trained on the set of evolutionary related proteins - families - and try to capture common information in this set. The two main models in this group are profile-based and Hidden Markov Model (HMM)-based similarities.

The profile-based model unites information from a group of sequences in a "profile", i.e. the probability matrix that reflects the statistics of occurrences of amino acids in a certain position in a sequence [Gribskov et al., 1987]. Multiple sequence alignment techniques are commonly used to build this matrix. This method is not entirely an alignment-free technique.

The other quite popular and effective similarity model is based on HMM [Krogh et al., 1994]. This well-known statistical model is trained on protein sequences to assign high probabilities to the sequences from a particular set. Thus, it captures the most conserved primary structure from a given family. HMM similarity exploits the same idea as the profile-based modeling but in a more general way.

The second group of similarity methods is based on protein *motifs*, i.e. amino acid sequence patterns of likely biological significance. Instead of focusing on the whole sequence comparison, the vector space of motif presence can be used to build a similarity measure. The authors of this approach [Ben-Hur and Brutlag, 2003] showed that it can outperform BLAST and Smith-Waterman similarities when combined with Support Vector Machine classifiers for the task of remote homology detection.

A key element of alignment-free sequence analysis is the type of sequence transformation used. The third group of similarities is based on amino acid composition (AAC) transformation [Chou and Zhang, 1993]. The AAC is a simple transformation of the protein sequence to the vector of frequencies of the 20 amino acids *alphabet*. The AAC can be considered from the viewpoint of n-gram concept. The idea of n-gram is widely used in computational linguistics and refers to a contiguous subsequence of size n from a sequence of items. The n-grams of protein sequences can be interpreted as short motif. Considering this definition, the AAC transformation can be seen as a 1-gram (or unigram) transformation into frequencies and can thus be generalized to n-gram frequencies transformation. The main problem with n-grams is the exponential growth of the computational complexity (number of possible different n-gram combinations) as a function of n. Some of the works use this extension of AAC where the maximum value of n is set to 3 [König et al., 2015b, Shkurin and Vellido, 2016].

The forth group of similarity measures make use of transformation which incorporate spacial information and physico-chemical properties of amino acids: pseudo-AAC, autocross-covariance transformation, physico-chemical distance transformation.

The pseudo-AAC was introduced by Chou in [Chou, 2001] as an extension of AAC that uses autocorrelation of the sequence computed between original sequence and its shifted version with some lag. Each amino acid is substituted by the vector composed of its hydrophobicity value, hydrophilicity value, and side-chain mass. Extending AAC with physico-chemical properties of amino acids. is successfully applied in GPCR classification using two different amino acids grouping schemes developed by Sandberg and Davies [Shkurin and Vellido, 2016].

Somehow similar to Chou's idea of using autocorrelation is used in the model of autocross-

covariance transformation (ACC) [Wold et al., 1993] which considers protein sequence as a time sequence with several spacial dimensions. These dimensions can be computed from physical-chemical properties of amino acids or using position-specific score matrix [Liu et al., 2011]. For more detailed explanation of this transformation and AAC please refer to Section 4.1.

Another transformation which is based on the combination of physico-chemical properties and incorporation of spatial sequence information is physico-chemical distance transformation (PDT) [Liu et al., 2012]. Similarly to pseudo AAC, it computes the difference between amino-acids physical chemical properties using the original sequence and its shifted version. This method has a second variant which is based on a profile sequence instead of the original one. PDT was used for GPCR protein classification of class C [König et al., 2015b, Cruz-Barbosa et al., 2015] and gave results similar to other alignment-free transformations.

3.2 Protein sequence classification models

This section covers classifiers which are widely used for the task of protein sequence classification in combination with the similarity techniques discussed in the previous section.

One of the most popular models used for classification of protein sequences is the Support Vector Machine (SVM). One of the possible explanations of its ubiquity apart from its great performance is the possibility to use the arbitrary similarity measure between two sequences by the exploitation of the kernel trick. Originally, SVMs build a linear decision boundary between classes. Introduction of the kernel allows to consider the high-dimensional feature spaces without need of heavy computations. Thus, SVMs allow to avoid exact vector representation of the sequence and can be applied to the analysis of symbolic sequences by just providing an adequate kernel function.

The variety of approaches successfully used in combination with SVM is impressive, covering local alignment [Shah et al., 2008], motif-based [Ben-Hur and Brutlag, 2003, Håndstad et al., 2007], ACC [Dong et al., 2009], PDT [Liu et al., 2012] and many others.

A completely different method from SVM is the Random Forest (RF) classifier, which has also successfully been applied in many bioinformatics tasks including protein classification [Kandaswamy et al., 2011], protein function prediction [Lin et al., 2011], an 3-D structure prediction [Dehzangi et al., 2010], amongst others, using only primary structure protein sequence information. The effectiveness of RF and its main advantage is based on its ensemble learning approach, i.e., the combination of several weak classifiers (decision trees) [Breiman, 2001]. The base classifiers for RF are decision trees. Each of the tree is trained on a sampled version of the dataset with replacement and on a random subset of features. As a tree-based classifier, RF can naturally deal with multi-class classification, which is the case of protein classification. As an ensemble technique it can be used for providing a closer look at the phenomenon of misclassification and misclassification consistency, i.e the base classifiers can assign different labels for the same sample, and the final decision is made by voting system. These votes can be used to more closely inspect the misclassified sequence [Shkurin and Vellido, 2016]. For more detailed explanation of RF and SVM refer to Section 4.2.

The next group of classifiers to consider is artificial neural network (ANN)-based classifiers. The idea to use ANNs for protein classification and secondary structure prediction appeared back in late 80-s - early 90-s [Qian and Sejnowski, 1988, Demeler and Zhou, 1991, Wu et al., 1992]. Most of the studies of these years used Multi-Layer Perceptron (MLP) architectures and were trained using the back-propagation algorithm. The MLP is an ANN organized in several layers, where each of the layers is fully connected with neighboring layers. The representation of the protein sequence required to turn it into fixed-size vectors, then used as an input for the ANN.

More recent research uses ANN architectures that work with sequences of arbitrary lengths, such as Recurrent Neural Networks (RNN) [Pollastri et al., 2002, Mishra and Pandey, 2012]. This family of network architectures is characterized by the directed cycle in the connections, i.e. the outputs from the previous sample are used in the processing of the new sample. The input then can be of arbitrary length and no transformation is needed for the whole sequence, but only for each of the 20 amino acids. Thus, the trained network performs two tasks: the transformation of the sequence and its classification. A particular instantiation of RNNs, the Long Short-Term Memory (LSTM) model was shown to be a fast and effective alternative to SVM and MLP

classifiers for the task of protein remote homology detection in [Hochreiter et al., 2007].

3.3 Previous research on GPCR class C

Note that the research compiled in this thesis is part of a wider effort concerning the analysis of Class C GPCRs. This effort was substantiated in a Spanish R+D program research project called "KAPPA AIM: Knowledge Acquisition in Pharmacoproteomics using Advanced Artificial Intelligence Methods" and in the ongoing project "COIN-GPCR: Computational Intelligence for Knowledge Discovery from G Protein-Coupled Receptors".

In this section, I briefly describe the main previous published work resulting from these projects that led to formulate the open questions that the thesis aims to answer.

Early work on the 2011 version of the database provided clear indications that there existed boundaries to the separability of the different Class C subclasses, using both supervised [König et al., 2014, König et al., 2015a] and semi-supervised [Cruz-Barbosa et al., 2015] learning approaches. Furthermore, some of the classes, precisely those which were removed in the 2016 versions of the databases (vomeronasal, odorant and pheromone), were the most responsible for such lack of separability.

These results were also confirmed from the viewpoint of visualization-oriented unsupervised learning. In this case, the subclasses that were shown to be worse discriminated by supervised classifiers were also shown to heavily overlap in unsupervised visualization models [Cárdenas Domínguez et al., 2015]. Close and detailed inspection of the sequence misclassification behaviour, though, revealed an interesting pattern. Some of these misclassifications were far too consistent, in the sense that the sequence was almost always misclassified (by difference classifiers and different implementations of the same classifier) as belonging to a specific *wrong* subclass.

This behaviour suggested the possibility of an instance of the *label noise* problem occurring. This is, the possibility that the sequence subclass labels were actually wrong as the result

of the uncertainty of the own database sequence labeling procedure, very often model-based itself. This would explain both the presence of consistently misclassified proteins and (at least partially) the limits of sub-class discrimination accuracy which exist independently of the choice of data transformation and classification technique.

The problem was first analyzed in [König et al., 2015b], were individual sequences were short-listed as potential cases of label noise to be further analyzed by database curators. Unsurprisingly perhaps, most of them belonged to the same three subclasses identified as the most difficult to discriminate. The data transformations used in this work were alignment-free: AAC, digram frequencies, ACC and PDT, and the classifier of choice was SVM.

Further work in [Shkurin and Vellido, 2016] again employed alignment-free data transformations, in this case with a Random Forest classifier to gauge the consistency of misclassification. As was aforementioned this ensemble technique with an internal classification *voting* system is naturally adequate for classification consistency analyses. The transformations were based on n-gram frequencies using amino acids directly of two of physico-chemical groupings. Thus, the accuracies achieved were similar across transformations and most consistent misclassifications were again detected mainly in the same three sub-classes: vomeronasal, odorant and pheromone.

All these previous studies were based on the earlier 2011 version of the database, a fact that automaticaly raised the following question: if the 2011 database, which included vomeronasal, odorant and pheromone as Class C GPCR subclasses, suffered from these classification problems, would the new 2016 versions of the database, which do not include those subclasses, suffer from the same problems? This thesis is a step on the direction of answering such question.

Chapter 4

Methods

The experiments reported in this thesis were carried out taking into account the previous published research on class C described in section 3.3. Therefore, the methods described in this chapter were selected so as to compare with the ones shown to be most successful in those studies, with the new recently introduced data transformation techniques.

The transformations that delivered good results in class C GPCR classification in previous work include AAC, Digram composition and ACC. Being briefly introduced in previous work review they are described in details in the following section, devoted to sequence alignment-free transformations.

The other newly introduced transformation techniques were developed for the statistical language modeling problem and were only recently applied to analyze protein sequences for the first time [Asgari and Mofrad, 2015]. They are described in more detail in the same following section.

The classification techniques of choice are SVM, Random Forest and Naive Bayes, which are all summarily explained in the second section of this chapter.

4.1 Alignment-free sequence transformations

4.1.1 Amino-acid composition and digram composition

Amino acid composition (AAC) [Chou and Zhang, 1993] is a very simple sequence transformation that includes summarized information about the composition of the sequences (a 20-dimensional vector, where each element is relative rate of occurrence of an amino acid in the sequence) and eliminates all the structural and sequential information. It also has no information about physico-chemical properties of the amino-acids though many of the amino-acids are close and can be easily substituted by one another without changing the physico-chemical properties of the whole sequence.

The digram transformation is a slightly more advanced one, but still very simple. The digram is the specific case of an *n-gram* transformation, where n=2. The transformation thus consists of the frequencies of occurrence of all possible amino acid pairings (resulting in a 20x20=400-dimensional vector).

4.1.2 ACC transformation

The auto-cross-covariance transformation (ACC) was introduced in [Wold et al., 1993] and is more complex than the previous ones because it explicitly takes into account the order of amino acids in the sequence and their physico-chemical properties.

Each of the amino acids is represented by 5 *z-scales* [Sandberg et al., 1998], which are derived from 26 possible physico-chemical properties. Then, for each of the 5 dimensions ($d = 1, \ldots, 5$) the vector of auto-covariance is computed using a fixed lag l.

$$AC_d(l) = \sum_{i=1}^{n-l} \frac{(v_{d,i} - \bar{v}_d)(v_{d,i+l} - \bar{v}_d)}{(n-l)^p}.$$

For each of the pairs of the dimensions the cross-covariance with lag is computed:

$$CC_d d'(l) = \sum_{i=1}^{n-l} \frac{(v_{d,i} - \bar{v}_d)(v_{d',i+l} - \bar{v}'_d)}{(n-l)^p}.$$

The $v_{i,d}$ is the value of d-th descriptor of the i-th amino acid in the sequence.

The lag takes values from 1 to L (maximum lag, fixed) and p is a normalization constant. The best values of p and L are selected by cross-validation.

The length of the resulting vector depends on the maximum lag and is built as follows:

$$[AC(1)\ CC(1)\ AC(2)\ CC(2)\ ..\ AC(L)\ CC(L)].$$

4.1.3 Distributed representations

As long as the primary structure of a protein is represented as a sequence of letters of arbitrary length in the Latin alphabet (that is, as a symbolic sequence), the idea of using methods of natural language processing (NLP) comes about naturally. This field has well-established methods of statistical language modeling whose goal is to learn the joint probability distribution of sequences of words in natural language.

One of the fundamental problems in statistical language modeling lies in the fact that learning the joint distribution of sequences of words requires a huge number of parameters estimations which depends on the length l of the sequence exponentially as $|V|^l$, where $|V|$ is vocabulary size [Bengio et al., 2003].

One of the NLP models recently introduced to face this problem resorts to a distributed representation of the words in $\mathbb{R}^m$. The joint probability function is learned in terms of these distributed representations and is expressed as a product of the conditional probabilities of a word given *its context*.

The distributed representations are built to capture the statistical properties of the dataset. The idea behind the name "distributed" can be expressed by the following properties: one concept

in the domain is represented in several dimensions and one dimension unites information about several real-world concepts.

As long as the distributed representation is in practice learned using a feed-forward ANN model, this "distributed" property of representation is implemented by using shared weights for all of the input elements.

One very popular ANN architecture for learning statistical language models with distributed word representations was introduced in [Bengio et al., 2003]. It is composed of three layers: a projection layer (that computes distributed representation), a non-linear hidden layer and an output softmax layer to predict the probability of each word in the vocabulary. This architecture has two main computational advantages: the number of free parameters scales linearly with the growth of the vocabulary size and it also scales linearly with the growth of sequence length.

Mikolov and colleagues, in [Mikolov et al., 2013b], introduced a further computational improvement for distributed representations by separating learning word representations from learning a language model. Two main simplifications for the architecture of the ANN are made: the nonlinear hidden unit is completely removed and the word vectors computed from each word in the context are summed (Figure 4.1, on the left). The model is called Continuous Bag-of-Words (CBOW) because of its similarity to the standard NLP Bag-of-Words model, which also disregards information about word order while keeping information about multiplicity. Authors go even further and introduce the Continuous Skip-gram architecture which instead of predicting the word out of its context is trained to predict the context given one word. This group of models and their implementation in particular are well-known as "word2vec".

These representations are shown to capture the syntactic and semantic relationship between words, i.e., linear operations in the obtained vector space capture some semantic or syntactic transformations over words in natural language, such as the comparative form of an adjective (big-bigger), or a 'capital-country' relationship. Thus, for example, the subtraction of the vector for the word '*France*' from the vector for '*Paris*' and addition of the vector for '*Italy*' gives a vector that is very close to the vector of '*Rome*' i.e. *'Paris' - 'France' + 'Italy' = 'Rome'*.

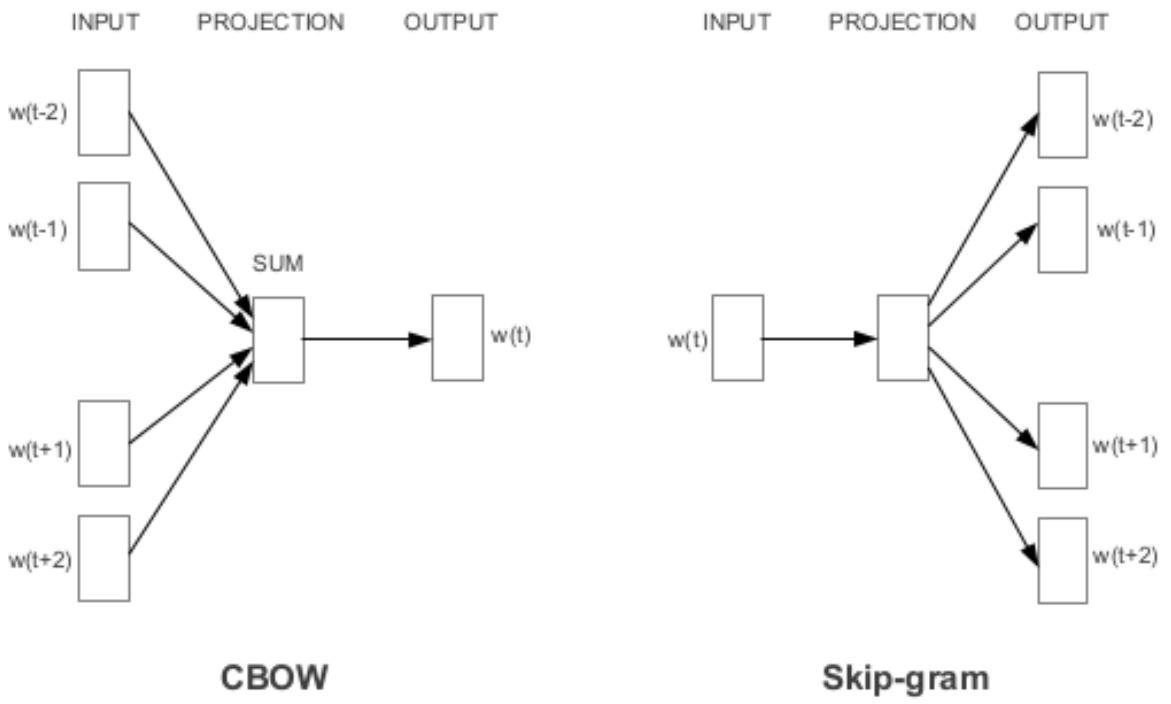

Figure 4.1: The architecture of two models for learning word embeddings [Mikolov et al., 2013a].

CBOW

The CBOW model is an ANN with one hidden layer. The input layer takes a subset from C words (where C is a parameter of the model) before and after the given word and sums their projections from a shared projection matrix. The output layer acts as a *log*-linear classifier which predicts the probability of the given word using the input. As long as the projections are summed up the order of the words in the context has no influence to the output which explains the name 'bag-of-words' of the method.

Continuous Skip-gram

A skip-gram is a generalization of an n-gram. While an n-gram is a contiguous sequence of n items from a given sequence, the skip-gram allows the sequence to be discontinuous, i.e. have gaps of size not more than k in total. More formally, the k-skip-n-gram for a sentence $w_1...w_m$

of arbitrary length m is:

$$\{w_{i_1}, w_{i_2}, ..., w_{i_n} | \sum_{j=2}^{n} i_j - i_{j-1} < k\}.$$

This idea of a skip-gram was used to build a second model which can also be represented as an ANN. The input is the middle word, which is projected to some vector space and the output is the probability of the words to appear within some range around the input word. More distant words have smaller weights when sampling the training set.

Protein structure modeling

This part is devoted to the explanation of the application of distributed word representations to the problem of protein classification using primary structures. The classification problem is split into two steps: getting a vector representation of a sequence and using this vector to train a classifier.

The concept of "word" in protein sequence is expressed as an n-gram of amino acids i.e. a contiguous subsequence of n amino acids. The whole sequence is considered as a "sentence" of the non-overlapping "words".

The distributed representation model is then trained to represent every n-gram as a fixed size vector. Getting a representation of a sentence from single word representations can be done by simply summing up all the word vectors. In this way a fixed size vector representation of a protein sequence can be obtained.

The idea of this representation of protein sequences was introduced in [Asgari and Mofrad, 2015]. It was also shown that this representation can capture meaningful physical and chemical properties of the proteins.

The training set is constructed using protein sequences by transforming them into n-gram (words) sequences. As long as words are chosen to be non-overlapping to consider all the possible n-grams in a sequence it should be split using offsets from 0 to $n - 1$. Thus, one sequence is transformed into n sequences of n-grams (Figure 4.2).

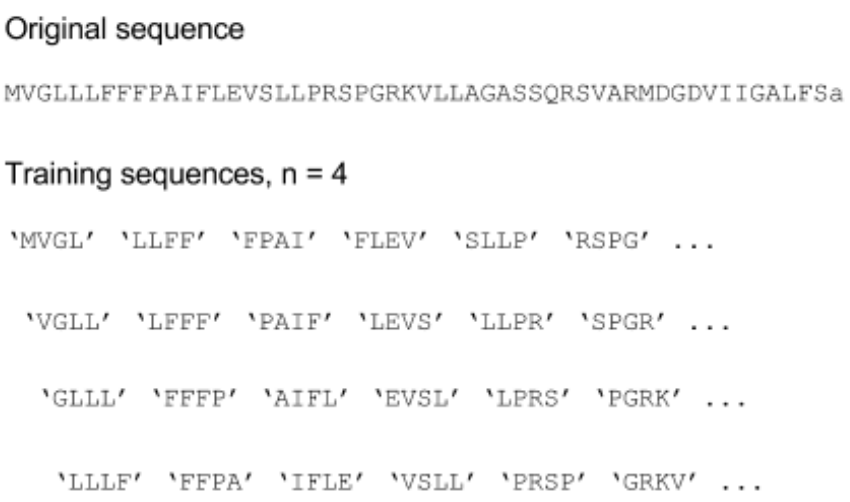

Figure 4.2: An example: obtaining training sequences from original sequence of human metabotropic glutamate receptor 1.

Each of the shifted sequences is then used to form training samples. Each training sample consists of a *context* and a *middle word*. The context is chosen using a context window size parameter - C i.e. maximum distance between the current and predicted word within a sequence. For a given C, a random value R from 1 to C is sampled. Then, R words from the history and R words from future are considered as a context for a middle word.

Then, training is performed depending on the type of the model. For the CBOW model, the input is constructed from the subset of all context words and the model is trained to maximize the probability for the middle word. For the Skip-gram model, the input is the middle word and the model is trained to maximize the average probability of all context words (Figure 4.3).

Summing up, the hyper-parameters for the distributed representation model are: the type of the model - CBOW or Continuous Skip-gram; the n - size of n-gram; the context window size C; and the size of the output vectors.

From here on, the distributed representations obtained for the protein sequences are referred to as "prot2vec".

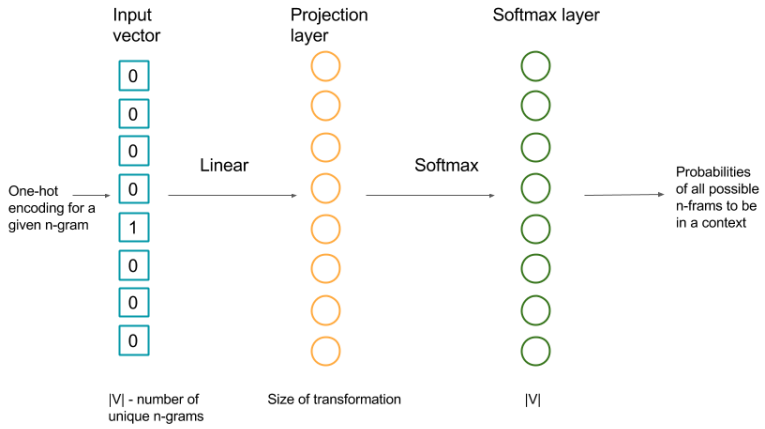

Figure 4.3: The architecture of Continuous Skip-gram model for the protein sequences

4.2 Classifiers

4.2.1 Support Vector Machines

Support Vector Machine (SVM) classifiers have *de facto* become the most popular method for protein classification. Their popularity can be explained by two main reasons: their simplicity combined with good performance, and the advantages of the *kernel trick*.

The SVM, introduced in [Boser et al., 1992], is trained to represent a function which is considered as a border between classes. The function is a linear combination of supporting vectors, i.e. of data samples that are close to the discriminative border. The optimal number of these vectors is estimated during a training process. The good generalization performance of the model can be explained by the nature of optimization function. The algorithm is trained to maximize the margin between the boundary and training samples.

In turn, the *kernel trick* allows to consider the task of class separation in a new dual space where the decision function can be expressed as a linear combination of basis functions, i.e. kernels.

The kernel trick allows to use convex optimization techniques due to linearity of decision function, even though the model itself is a non-linear function of input. The second great benefit of

a kernel trick is computational tractability. The kernel is a tractable function of input samples, while the direct computations in dual space could be intractable or resource-consuming, as the dual space can even have infinite dimensionality.

The most widely used kernel functions are Radial Basis Functions (RBF) with their centers corresponding to support vectors.

4.2.2 Random Forest

The Random Forest (RF) classifier is an ensemble decision tree-based classifier. The decision tree classifiers are trained to split an input space into the regions with associated class labels. The splits are typically axis-aligned and are selected to maximize the gain in specified metrics which are usually the Gini impurity or the information gain.

The main improvement of RF over simple decision tree classifier is the usage of bagging techniques and the random subspace method.

Bagging (or bootstrapped aggregating) is a technique of model averaging which uses models trained on subsamples of the original training set. The sub-sampling is performed independently with replacement. The technique allows to avoid over-fitting. That is, the generalization error is shown to converge to a limit with the growth of the number of trees [Breiman, 2001].

The random subspace method also allows to avoid over-fitting. It decreases the generalization error by reducing the correlation between trees. The computation times for training such forests are reasonably low.

The most popular implementation for the RF considers axis-aligned splits. This can cause worse performance for RF than for linear classifiers for problems with significant linear dependencies between variables (for example, when one of two classes appear for samples where $x_1 > x_2$) but performs well for non-linear problems.

4.2.3 Naive Bayes

Naive Bayes is a simple model that will provide us with a baseline for performance comparison. It is based on simply applying the Bayes theorem with the assumption of independence of variables, which is considered somehow 'naive'.

Under this assumption, which in practice often works well, the probability of a class C_i ($i = 1..N$) given an input X could be then computed as:

$$P(C_i|X) = P(C_i) \prod_{n=1}^{N} P(X_n|C_i).$$

This probability could be used for class prediction using Maximum A Posteriori (MAP) estimation:

$$y = arg\max_i P(C_i) \prod_{n=1}^{N} P(X_n|C_i).$$

The classifiers differ depending on the assumption about the probability distribution for $P(X_n|C_i)$.

For continuous variables the typical assumption is a Gaussian distribution:

$$P(X_n|C_i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp^{-\frac{X_n-\mu_n}{2\sigma_n^2}}.$$

The parameters for the Gaussian i.e. μ_n and σ_n are estimated using Maximum Likelihood.

4.3 Visualization techniques

Visualization of high dimensional data often becomes a key to their understanding, leading in turn to insights about the more efficient way to process them and making preliminary hypothesis about their structure and separability.

In order to visualize the data, dimensionality reduction techniques can be used. Unsupervised dimensionality reduction techniques (those which do not use labeled data) are of particular

interest because they allow to get an unbiased understanding of data structure and class separability.

One of the most popular dimensionality reduction techniques is Principal Component Analysis (PCA), summarily described next. This is followed by the description of a recently developed technique that has attracted much attention of late, called Stochastic Neighbourhood Embeddings (SNE), and a description of a variant called t-distributed SNE (t-SNE) [Maaten and Hinton, 2008].

4.3.1 PCA

PCA is a linear dimensionality reduction technique which is aimed at preserving the variance of the data. The lower space variables are linearly uncorrelated and are sorted in terms of percentage of variance of original data. As a consequence the dissimilar points are mapped as far as possible in lower dimensions. The dimensionality of the new space can thus be varied depending on the need of the research. For visualization techniques it is obviously convenient to use first 2 or 3 components. By its nature, PCA is sensitive to feature scales.

4.3.2 T-SNE

T-SNE performs a mapping from a high-dimensional input space in which the analyzed data reside to a lower dimensional space of latent variables in such a way that the local structure of the data is preserved as much as possible [Hinton and Roweis, 2002]. This allows to understand the structure of the data in cases when high dimensional data is lying in several low-dimensional manifolds. For example, for proteins distributed representations the different manifolds in the output space may represent different protein classes.

The non-linear mapping between spaces is learned by minimizing the Kullback-Leibler divergence between the conditional probabilities which represent similarities in both spaces. The conditional probability of x_j to be a neighbor of x_i is equal to the proportion of its density under a Gaussian distribution centered in x_i:

$$P(x_j|x_i) = \frac{\exp \frac{-(||x_j - x_i||)^2}{2\sigma_i^2}}{\sum_{k \neq j} \exp \frac{-(||x_k - x_i||)^2}{2\sigma_i^2}}.$$

The name of the method "stochastic neighbourhood embedding" is explained by the fact that the position of the point in the new space is determined via its neighbours. T-SNE is an improvement over the SNE method which uses a symmetric cost function and the Student t-distribution instead of a Gaussian distribution for low dimensional similarities. This allows simpler optimization computations and avoids the "crowding problem" [Maaten and Hinton, 2008] (as a t-Student distribution is less sensible to the presence of data outliers).

4.4 Performance metrics

Several performance metrics are used to select the best classification model in cross-validation. It is important to understand the difference between these figures of merit because they have much influence on the interpretation of the achieved results.

The accuracy metric is typical of classification tasks, and is calculated as the percentage of the correctly classified samples. It suits only reasonably well class-balanced data because it treats all the classes equally. One alternative to this is the use of a weighted metric, which is an average of binary metric weighted by the prevalence of the classes. The binary weighted metrics used in the reported experiments are precision, recall, and the f-score.

Precision and recall reflect the relevance of the result of a given class. Precision specifies the fraction of correctly classified samples out of all samples classified to be in a class. Recall defines the fraction of correctly classified out of all true samples of a class. Precision is then covers type I errors (false positives) and recall type II errors (false negatives).

The F-score is a harmonic mean of precision and recall. It unites information from both metrics and thus is more demonstrative if both types of errors are considered as equally undesirable.

The other way of treating multiclass classification with unbalanced classes is using metrics

which are adjusted to this type of task. Matthews correlation coefficient (MCC), for instance, is used for binary classification with unbalanced data and is generalized for the case of multiclass classification.

In summary, for the task of multiclass classification with k classes and N samples, the used metrics will be the following:

$$Accuracy = \frac{\sum_{i=1}^{k} tp_i}{N},$$

$$Precision = \frac{\sum_{i=1}^{k} w_i(\frac{tp_i}{tp_i+fp_i})}{N},$$

$$Recall = \frac{\sum_{i=1}^{k} w_i(\frac{tp_i}{tp_i+fn_i})}{N},$$

$$F\text{-}score = \frac{\sum_{i=1}^{k} 2w_i \frac{precision_i * recall_i}{precision_i + recall_i}}{N},$$

$$MCC = \frac{\sum_i \sum_j \sum_l C_{ii}C_{jl} - C_{ij}C_{li}}{\sqrt[2]{\sum_i(\sum_j C_{ij})(\sum_{i'|i'\neq i} \sum_{j'} C_{i'j'})} \sqrt[2]{\sum_i(\sum_j C_{ji})(\sum_{i'|i'\neq i} \sum_{j'} C_{j'i'})}},$$

where

w_i is the "weight" of a given class, equal to the number of samples in this class;

tp_i is the number of samples of the given class which were classified as belonging to the class;

fp_i is the number of samples of the other classes which were classified as belonging to the given class;

fn_i is the number of samples of the given class which were classified as belonging to the other classes; and

C_{ij} is the ij-th element of $C_{k\times k}$ confusion matrix, which contains all classes.

Chapter 5

Experiments and results

This chapter covers the description of the performed experiments and the most important results along with their discussion. Further detailed results can be found in Appendices A-C.

The first section is devoted to the comparison of sequence data transformations, including AAC, Digram composition and ACC. Then, the next two sections are devoted to the comparison of *prot2vec* trained on different datasets and *prot2vec* with different hyper-parameters. After that, the best models of *prot2vec* are compared to best models of ACC. The fifth section considers different classification techniques and their performances. The discussion also covers the comparison of the results for the three versions of the database, based on all the presented results.

The last part of the chapter provides further empirical evidence to support the previously presented results from the point of view of unsupervised learning visualization techniques.

All the experiments in this chapter were set and performed using the same algorithm, as follows:

1. Randomly permute data.

2. Perform stratified split into 5 folds.

3. Take first fold as test set, the rest folds as training set .

4. Perform grid search with 5-fold cross-validation on the training set to choose best hyper-parameters, maximizing f-score.

5. Evaluate performance for test set .

6. Repeat steps 3-5 for the next 4 folds.

7. Report an average performance for the 5 test sets.

The models used for the experiments were: *prot2vec* trained on Swiss-Prot, *prot2vec* trained on GPCRdb, ACC, AAC and Digram composition. The classifiers of choice were: Random Forest, SVM and Gaussian Naive Bayes. The hyper-parameters for the classifiers are summarily listed in Tables 5.1 and 5.2 for SVM and RF respectively.

Hyper-parameter	Values
Kernel parameter γ	0.1, 0.01, 0.001, 0.0001, 0.00001, 1e-06, 1e-07
Penalty parameter C	1,2,3,4,5, 50,100,1000, 10000
Kernel	RBF

Table 5.1: Hyper-parameters for the SVM classifier

Hyper-parameter	Values
Number of trees	100, 500, 1000
Criterion	Gini impurity
Fraction of features to consider for each split	0.1, 0.4, 0.7, 1.0
The minimum number of samples to make split	1, 2, 1%, 5%

Table 5.2: Hyper-parameters for RF classifier

5.1 Classification with different transformations

First, the different transformations used with SVM classifier are compared to each other for each of the database versions (tables 5.4-5.6). The SVM classifier was used because of its better performance for all the transformations. This is discussed in detail in the next section.

As previously mentioned, the transformations under consideration are: AAC, Digram composition and ACC. They are comparable for each of the version of the database because they either

Hyper-parameter	Values
Normalization constant p	0.5, 1, 2
Maximum lag L	2..30 (with step 2)

Table 5.3: Hyper-parameters for ACC transformation.

Model	Accuracy	MCC	F-measure
AAC	0.7682	0.7091	0.7677
Digram	0.9172	0.8965	0.9159
ACC (p=0.5, L=13)	**0.9278**	**0.9095**	**0.9271**

Table 5.4: Classification results for the 2011 database using the SVM classifier.

Model	Accuracy	MCC	F-measure
AAC	0.9846	0.9749	0.9844
Digram	0.9941	0.9904	0.9941
ACC (p=0.5, L=21)	**0.9964**	**0.9944**	**0.9964**

Table 5.5: Classification results for the May 2016 database using the SVM classifier

Model	Accuracy	MCC	F-measure
AAC	0.9743	0.9584	0.9743
Digram	0.9925	0.9875	0.9923
ACC (p=0.5, L=21)	**0.9957**	**0.9931**	**0.9957**

Table 5.6: Classification results for the September 2016 database using the SVM classifier.

have no hyper-parameters (AAC, Digram) or their hyper-parameters are fine-tuned for each version separately (ACC, the hyper-parameters checked are presented in Table 5.3).

The AAC transformation is the simplest and has the worst performance which can be explained by its simplicity. The digram transformation performance is very close to ACC. This is also an unexpected result because ACC incorporates the sequential information and physico-chemical properties of amino acids while digram takes only into account the number of subsequences composed by two amino acids.

Comparison of the three tables for the different versions of the dataset show a clear picture of improvement of its quality during time. The most significant step was made switching from the 2011 to the May 2016 versions. Taking into account that the most problematic classes in the 2011 database were vomeronasal, pheromone and odorant receptors and that they were removed in the 2016 versions, the overall improvement in classification makes sense. This is further confirmed by the confusion matrices (Tables 5.7-5.9).

	MGlu	CaS	$GABA_B$	VN	Ph	Od	Taste 1
MGlu	345	0	0	0	2	4	0
CaS	0	45	0	0	2	1	0
$GABA_B$	2	0	215	0	1	0	0
VN	0	1	0	311	28	4	0
Ph	8	0	1	23	356	4	0
Od	2	1	0	11	13	75	0
Taste 1	1	0	0	0	0	0	44

Table 5.7: Confusion matrix for the ACC transformation of the 2011 database, using the SVM classifier.

	$GABA_B$	Taste 1	MgluR	CaS
$GABA_B$	58	1	1	0
Taste 1	0	193	0	1
MgluR	0	0	467	0
CaS	0	0	1	124

Table 5.8: Confusion matrix for the ACC transformation of the May 2016 database, using the SVM classifier.

	$GABA_B$	Taste 1	MgluR	CaS
$GABA_B$	87	1	0	0
Taste 1	0	226	0	1
MgluR	0	0	516	0
CaS	0	1	1	101

Table 5.9: Confusion matrix for the ACC transformation of the September 2016 database, using the SVM classifier

5.2 Comparison of prot2vec models with fixed hyper-parameters

In the original proposal, [Asgari and Mofrad, 2015] all the hyper-parameters for *prot2vec* transformation were fixed:

- $n = 3$ (size of n-gram);
- $C = 25$ (size of context window);
- size of output vectors = 100;
- model type - Skip-gram.

As a training set, the Swiss-Prot database of October 2014 version was used. Swiss-Prot is a part of the Universal Protein Resource (UniProt), which contains more than 500,000 manually annotated protein sequences [Consortium et al., 2017]. The authors made their 3-gram representations publicly available from Harvard Dataverse: http://dx.doi.org/10.7910/DVN/JMFHTN.

For comparison of the *prot2vec* models trained on the different datasets the hyper-parameters were fixed to the values described in the original proposal. With these hyper-parameters, two models are obtained: the first one, trained on Swiss-Prot of October 2014 (from publicly available 3-gram representations), and the second one trained on the whole GPCRdb (not only class C) of one of the two versions of 2016 (Table 5.10).

Database	Model	Accuracy	MCC	F-measure
2011	prot2vec - Swiss-Prot 2014_10	0.8901	0.8624	0.8894
	prot2vec - GPCRdb May2016	0.8636	0.8294	0.8637
May 2016	prot2vec - Swiss-Prot 2014_10	0.9882	0.9809	0.9881
	prot2vec - GPCRdb Sept2016	0.9858	0.9775	0.9858
Sept 2016	prot2vec - Swiss-Prot 2014_10	0.9914	0.9860	0.9914
	prot2vec - GPCRdb Sept2016	0.9925	0.9875	0.9925

Table 5.10: Classification results for all databases using SVM classifier and prot2vec transformation

The performances of the two versions of the *prot2vec* trained using Swiss-Prot and GPCRdb are very close for both version of 2016, a result that was not expected because Swiss-Prot contains proteins with different functions and structures and a model trained using all these proteins was not expected to be relevant for classification of the very specific type of proteins (class C GPCRs) under investigation in this thesis.

On the other hand, the training set size for models trained on GPCRdb is about 35 times smaller than that of Swiss-Prot (≈14,000 and ≈500,000 respectively). Despite this size difference, the models are comparable, which is either explained by the higher similarity of the proteins inside GPCRdb (which compensates the small size of the training set), or by the sufficiency of this relatively small number of proteins for good classification.

However, the difference between the two models is larger for the 2011 database version. The possible explanation for this is the mismatch between the training version of GPCRdb and the

version used for classification. This is because the full database of 2011 is no longer available and thus the closest in time version was used (May 2016).

5.3 Hyper-parameter selection for prot2vec and its comparison with ACC

The parameters presented in Table 5.11) were used for the comparison of the different sets of hyper-parameters for training distributed representations .

Parameter	Values
n-gram size	2, 3, 4, 5 (n=5 only for GPCRdb Sept 2016)
C - context window size	5, 10, 20, 25, 30, 40
output vector size	100, 200, 300
model type	CBOW, Skip-gram

Table 5.11: Hyper-parameters used in training the continuous vector representations of n-grams of amino acids.

First, the search of the optimal hyper-parameters was performed using the latest version (September 2016) of the GPCRdb as training set and class C sequences of the same version for classification purposes (see results in Appendix B). We discovered that with this settings for classification the resulting accuracy approaches 100%, meaning that almost any combination of hyper-parameters works extremely well.

One conclusion which can be made for this experiment is that the *prot2vec* transformation with different parameters shows consistency of classification across different model types (Skip-gram, CBOW), different sizes of vectors, n-grams lengths and context windows. There are no linear dependencies, which can be interpreted as robustness of this type of transformations for this dataset.

The other conclusion is that the older version of the dataset, which provides far worse results, will be probably more illustrative in terms of showing differences across the hyper-parameters.

The second experiment was performed using the old 2011 GPCRdb version trained on the

Swiss-Prot dataset of April 2017 (see results in Appendix C). There is an increasing trend in accuracy as the size of the output vectors increases.

The best results for classification for all the versions of the dataset, among the three transformations from section 5.1, were achieved by ACC. For a completely fair comparison of ACC and prot2vec, the prot2vec hyper-parameters should be fine-tuned as it was done for ACC.

The results in Tables 5.12 and 5.13 compare prot2vec trained on Swiss-Prot with ACC for the version of 2011 database and prot2vec trained on GPCRdb with ACC for the version of September 2016 respectively.

Model	Best hyper-parameters	Accuracy	MCC	F-measure
prot2vec - Swiss-Prot 2017_04	size=300, n=5, C=20, CBOW	0.9232	0.9038	0.9231
ACC	p=0.5, L=13	0.9278	0.9095	0.9271

Table 5.12: Classification results for the 2011 database using the SVM classifier with distributed representations compared to ACC.

Model	Best hyper-parameters	Accuracy	MCC	F-measure
prot2vec - GPCRdb Sept 2016	size=200, n=2, C=25, CBOW	0.9968	0.9948	0.9968
ACC	p=0.5, L=21	0.9957	0.9931	0.9957

Table 5.13: Classification results for the September 2016 database, using the SVM classifier with distributed representations compared to ACC.

For both versions, the differences of results for prot2vec and ACC are below 0.5%, that is, rather small. The version of 2011 is even more demonstrative, because of the larger difference in results between ACC and other models. The comparable performance of ACC and prot2vec can lead to two interesting conclusions: 1) the models are equally applicable for the problem of GPCR class C classification and 2) the models are achieving the optimal performance limits as described in [König et al., 2015b].

5.4 Comparison of different classifiers

The second set of experiments compares the results for the given transformations with different classifiers (tables 5.14-5.16). The transformations presented are the two prot2vec transforma-

tions with fixed hyper-parameters : n-gram size = 3, output vector size = 100, context window = 25, model - Skip-gram; the results for others transformations can be found in Appendix A.

Model	Classifier	Accuracy	MCC	F-measure
prot2vec (Swiss-Prot 2014_10)	NB	0.6000	0.5153	0.6070
	RF	0.8583	0.8220	0.8566
	SVM	**0.8901**	**0.8624**	**0.8894**
prot2vec (GPCRdb May2016)	NB	0.5854	0.4931	0.5889
	RF	0.7947	0.7446	0.7971
	SVM	**0.8636**	**0.8294**	**0.8637**

Table 5.14: Classification results for the 2011 database using distributed representation.

Model	Classifier	Accuracy	MCC	F-measure
prot2vec (Swiss-Prot 2014_10)	NB	0.8118	0.7229	0.8207
	RF	0.9716	0.9535	0.9713
	SVM	**0.9882**	**0.9809**	**0.9881**
prot2vec (GPCRdb May2016)	NB	0.8615	0.7972	0.8688
	RF	0.9775	0.9642	0.9775
	SVM	**0.9858**	**0.9775**	**0.9858**

Table 5.15: Classification results for the May 2016 database, using distributed representation.

Model	Classifier	Accuracy	MCC	F-measure
prot2vec (Swiss-Prot 2014_10)	NB	0.9113	0.8619	0.9132
	RF	0.9850	0.9758	0.9849
	SVM	**0.9925**	**0.9875**	**0.9925**
prot2vec (GPCRdb Sept2016)	NB	0.9808	0.9692	0.9809
	RF	0.9893	0.9826	0.9893
	SVM	**0.9925**	**0.9875**	**0.9925**

Table 5.16: Classification results for the September 2016 database, using distributed representation.

The Naive Bayes classifier was used for baseline purposes and it fulfills such expectations. The difference between NB results and results for SVM and RF are larger for the 2011 version and become smaller for the most recent versions. This shows a different perspective on database version comparison, because, using only SVM, the difference between the two last versions is not clear. With NB, results show the significant improvement of the quality in the last version (September 2016) over the previous one (May 2016) - over 10% accuracy. The NB classifier is considered as more demonstrative in a sense that its good or bad performance more directly depends on the underlying data and not on the fine-tuning of the hyper-parameters.

The RF model performs worse than SVM for all the transformations which can be explained by the nature of decision-tree classifiers, which can show bad performance with linear dependencies, or perhaps even by the lack of hyper-parameter fine-tuning.

Note that, in any case, the classification using the 2016 databases yields excellent results, with very few misclassifications for all transformations and all classifiers. From the point of view of database curation, this is very good news, as misclassifications are now scarce and can be individually tackled in far more detail. Note that the amount of misclassifications in the 2011 database was significant and that, out of the misclassified cases, many showed very consistent misclassification behaviours that hinted the existence of label noise problems. Note that shortlisting those cases required detailed investigation [Shkurin and Vellido, 2016, König et al., 2015b].

5.5 Visualization

Two techniques were used for sequence data visualization: PCA and t-SNE. The new visualization open-source tool provided by Google Inc. - TensorBoard (https://www.tensorflow.org/get_started/summaries_and_tensorboard) allows to interactively visualize high-dimensional data in 3-D and 2-D using both techniques. The transformation used for all the cases was Swiss-Prot-based prot2vec (version October 2014).

5.5.1 PCA

The PCA technique, despite its simplicity and linear nature, reflects the structure of the datasets discovered by classification (Figure 5.1). As can be seen from the 3-D plots of first 3 principal components, the 2011 version of the dataset contains very mixed data and the borders between classes cannot be clearly seen.

For the May 2016 version, the classes can be visually distinguished and some are actually quite separated from the others. Almost all of the Calcium-Sensing receptors are far from

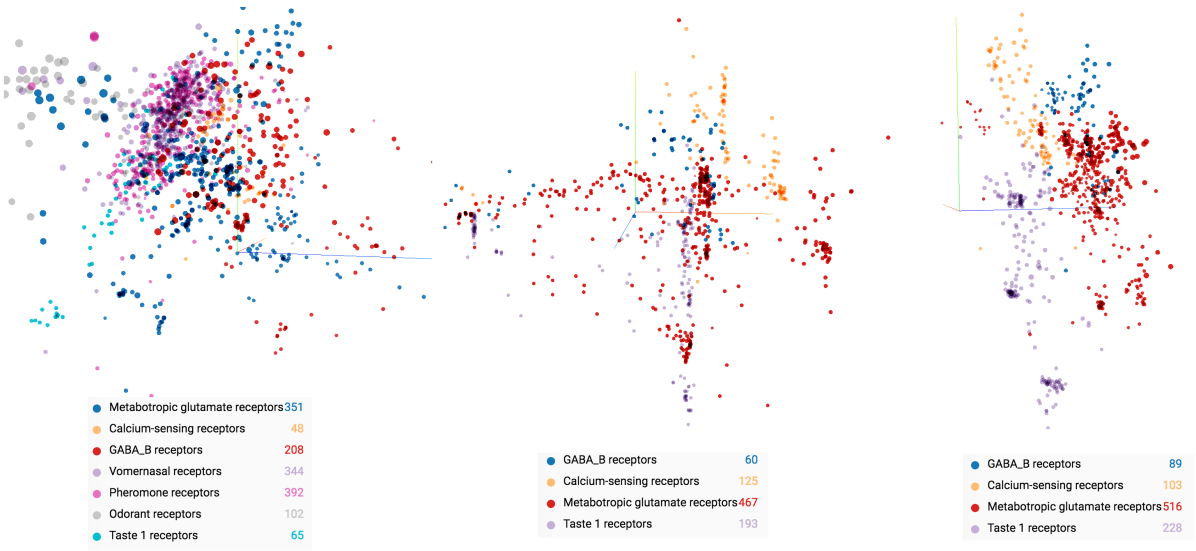

Figure 5.1: PCA with 3 components: left - 2011, middle - May 2016, right - September 2016

other points, while Taste 1 are also grouped together. $GABA_B$ are still seen to be mixed with Metabotropic glutamate receptors, though.

The picture for the September 2016 version is even better. All of the classes can be seen as separable groups with the exception of just a few points, which are likely to be the cause of the minor misclassifications.

5.5.2 T-SNE

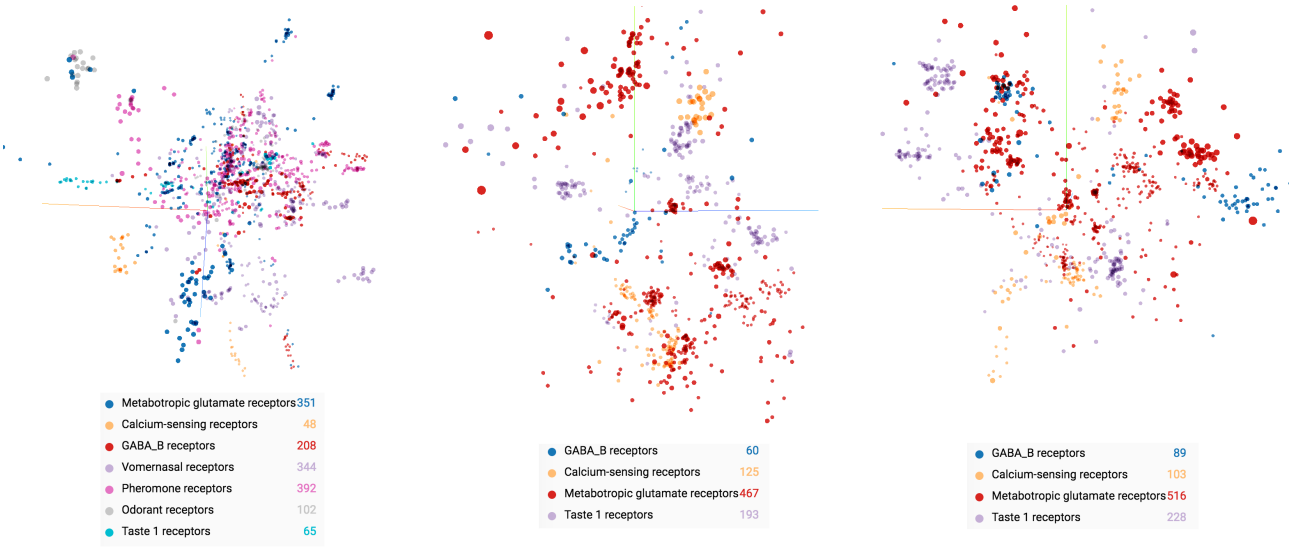

Figure 5.2: t-SNE: left - 2011, middle - May 2016, right - September 2016

Database	Perplexity	Learning rate	Number of iterations
2011	24	1	69
May 2016	56	0.1	141
September 2016	56	0.1	145

Table 5.17: Parameters for t-SNE visualization technique.

T-SNE works differently from PCA, which leads to a different resulting structure (see Figure 5.2). The points group in blocks the size of which is indirectly controlled by the perplexity parameter of the model.

The hyper-parameters for t-SNE were chosen empirically by using perplexities in a range from 2 to 100, with step 1, and learning rates from 0.001 to 100, with step 1 in logarithmic scale (Table 5.17).

The blocks for the two 2016 versions consist of the same classes and as a result classes become separable with minor exceptions. For the version of 2011, however, classes are quite mixed though not as much as in the PCA transformation.

Chapter 6

Conclusion

6.1 Summary of Thesis Achievements

The thesis has two perspectives: 1) application of machine learning techniques to investigate the changes of the quality of data in three different versions of the database, 2) application of distributed transformation model developed for natural text processing to the task of protein classification.

The first objective was obtained using five transformations, three classifiers and two visualization techniques. All of the experiments show that the most significant improvement in data quality were made between the version of 2011 and May 2016. These improvements can be explained mostly by the elimination of three problematic classes: vomeronasal, odorant and pheromone receptors. The best accuracy of classification was improved by about 7%.

The differences between the versions of May and September 2016 are not so large and are more controversial because some of the transformations perform better with earlier version while others on the contrary - with later version. Thus the quality of the data is almost the same, while the changes in classes distribution is significant.

The second objective of the exploration of prot2vec transformation was achieved by the following components in experimental part:

- Comparison with three other alignment-free transformations. Prot2vec performance difference with ACC is less than 0.5%.

- Training on two different training databases - Swiss-prot and GPCRdb. The difference between two databases as training sets is not significant though Swiss-prot is more universal, because it worked well for all three version of test database, while GPCRdb has the best performance when the training and test version of the database match.

- Training with different hyper-parameters: context window size, size of output vector, length of n-gram, two types of models - Skip-gram and CBOW. The prot2vec provided consistent results with all the hyper-parameters checked with a small increasing trend for the size of the output vector.

The experimental results reported in the thesis provide evidence that the prot2vec transformation is adequate and robust enough for the classification of protein primary structures, even if it does not provide clear advantages with respect to other existing and investigated sequence transformations.

6.2 Future Work

This thesis has opened a few avenues of research that could easily be followed in future work. Such work might include the processing of new versions of the GPCRdb database as they are released in the public domain, exploring their evolution as an aid to data curation. Such aid would be in the form of identification of potential cases of sub-class mislabeling. This analysis could also be extended to include more detailed levels of sub-class characterization (for instance, metabotropic glutamate receptors and $GABA_B$ receptors are known to be structured as a collection of subtypes at different levels; the first are known to be divided into eight types, but also in three different groups in this case based on their pharmacology and G protein-coupling profile).

All these results could be provided to the public through web-based access. This could include

the scores and confusion matrices for classification and consistent misclassifications found, including detailed information of individual proteins consistently mislabeled.

The investigation of the prot2vec transformation technique could be advanced by taking into account the whole GPCR database, instead of only its class C. In this case, some of the other classes might include receptors that are well characterized from the point of view of their 3-D structure, an information that could be used to supplement the analyses.

Bibliography

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

[Asgari and Mofrad, 2015] Asgari, E. and Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS One*, 10(11):e0141287.

[Ben-Hur and Brutlag, 2003] Ben-Hur, A. and Brutlag, D. (2003). Remote homology detection: a motif based approach. *Bioinformatics*, 19(suppl 1):i26–i33.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Cárdenas Domínguez et al., 2015] Cárdenas Domínguez, M. I., Vellido Alcacena, A., König, C., Alquézar Mancho, R., and Giraldo Arjonilla, J. (2015). Visual characterization of misclassified class C GPCRs through manifold-based machine learning methods. *Genomics and Computational Biology*, 1(1):e19.

[Chou, 2001] Chou, K.-C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function, and Bioinformatics*, 43(3):246–255.

[Chou and Zhang, 1993] Chou, K.-C. and Zhang, C.-T. (1993). A new approach to predicting protein folding types. *Journal of Protein Chemistry*, 12(2):169–178.

[Consortium et al., 2017] Consortium, U. et al. (2017). UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169.

[Cruz-Barbosa et al., 2015] Cruz-Barbosa, R., Vellido, A., and Giraldo, J. (2015). The influence of alignment-free sequence representations on the semi-supervised classification of class C G protein-coupled receptors. *Medical & Biological Engineering & Computing*, 53(2):137–149.

[Dehzangi et al., 2010] Dehzangi, A., Phon-Amnuaisuk, S., and Dehzangi, O. (2010). Using random forest for protein fold prediction problem: An empirical study. *J. Inf. Sci. Eng.*, 26(6):1941–1956.

[Demeler and Zhou, 1991] Demeler, B. and Zhou, G. (1991). Neural network optimization for E. coli promoter prediction. *Nucleic Acids Research*, 19(7):1593–1599.

[Dong et al., 2009] Dong, Q., Zhou, S., and Guan, J. (2009). A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20):2655–2662.

[Gregory et al., 2012] Gregory, K. J., Noetzel, M. J., and Niswender, C. M. (2012). Pharmacology of metabotropic glutamate receptor allosteric modulators: structural basis and therapeutic potential for CNS disorders. *Oligomerization and Allosteric Modulation in G-Protein Coupled Receptors*, 115:61.

[Gribskov et al., 1987] Gribskov, M., McLachlan, A. D., and Eisenberg, D. (1987). Profile analysis: detection of distantly related proteins. *Proceedings of the National Academy of Sciences*, 84(13):4355–4358.

[Håndstad et al., 2007] Håndstad, T., Hestnes, A. J., and Sætrom, P. (2007). Motif kernel generated by genetic programming improves remote homology and fold detection. *BMC Bioinformatics*, 8(1):23.

[Hinton and Roweis, 2002] Hinton, G. and Roweis, S. (2002). Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840.

[Hochreiter et al., 2007] Hochreiter, S., Heusel, M., and Obermayer, K. (2007). Fast model-based protein homology detection without alignment. *Bioinformatics*, 23(14):1728–1736.

[Howe et al., 2008] Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D., Kania, R., Schaeffer, M., St Pierre, S., and Twigger, S. (2008). Big data: The future of biocuration. *Nature*, 455(7209):47–50.

[Isberg et al., 2016] Isberg, V., Mordalski, S., Munk, C., Rataj, K., Harpsøe, K., Hauser, A. S., Vroling, B., Bojarski, A. J., Vriend, G., and Gloriam, D. E. (2016). GPCRdb: an information system for G protein-coupled receptors. *Nucleic Acids Research*, 44(D1):D356–D364.

[Kandaswamy et al., 2011] Kandaswamy, K. K., Chou, K.-C., Martinetz, T., Möller, S., Suganthan, P., Sridharan, S., and Pugalenthi, G. (2011). AFP-pred: A random forest approach for predicting antifreeze proteins from sequence-derived properties. *Journal of Theoretical Biology*, 270(1):56–62.

[Kolakowski Jr, 1993] Kolakowski Jr, L. F. (1993). GCRDb: a g-protein-coupled receptor database. *Receptors & Channels*, 2(1):1–7.

[König et al., 2014] König, C., Alquézar, R., Vellido, A., and Giraldo, J. (2014). Finding class C GPCR subtype-discriminating n-grams through feature selection. In *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*, pages 89–96. Springer.

[König et al., 2015a] König, C., Alquézar, R., Vellido, A., and Giraldo, J. (2015a). The extracellular N-terminal domain suffices to discriminate class C G protein-coupled receptor subtypes from n-grams of their sequences. In *Neural Networks (IJCNN), 2015 International Joint Conference on*, pages 1–7. IEEE.

[König et al., 2015b] König, C., Cárdenas, M. I., Giraldo, J., Alquézar, R., and Vellido, A. (2015b). Label noise in subtype discrimination of class C G protein-coupled receptors: A systematic approach to the analysis of classification errors. *BMC Bioinformatics*, 16(1):314.

[Krogh et al., 1994] Krogh, A., Brown, M., Mian, I. S., Sjölander, K., and Haussler, D. (1994). Hidden Markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235(5):1501–1531.

[Lin et al., 2011] Lin, W.-Z., Fang, J.-A., Xiao, X., and Chou, K.-C. (2011). iDNA-Prot: identification of DNA binding proteins using random forest with grey model. *PloS One*, 6(9):e24756.

[Lindsley et al., 2016] Lindsley, C. W., Emmitte, K. A., Hopkins, C. R., Bridges, T. M., Gregory, K. J., Niswender, C. M., and Conn, P. J. (2016). Practical strategies and concepts in GPCR allosteric modulator discovery: Recent advances with metabotropic glutamate receptors. *Chemical Reviews*, 116(11):6707–6741.

[Liu et al., 2012] Liu, B., Wang, X., Chen, Q., Dong, Q., and Lan, X. (2012). Using amino acid physicochemical distance transformation for fast protein remote homology detection. *PLoS One*, 7(9):e46633.

[Liu et al., 2011] Liu, X., Zhao, L., and Dong, Q. (2011). Protein remote homology detection based on auto-cross covariance transformation. *Computers in Biology and Medicine*, 41(8):640–647.

[Lord et al., 2004] Lord, P., Macdonald, A., Lyon, L., and D, G. (2004). From data deluge to data curation. In *Proceedings of the UK e-Science All Hands Meeting*, pages 371–375.

[Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Mishra and Pandey, 2012] Mishra, P. and Pandey, P. N. (2012). Elman RNN based classification of proteins sequences on account of their mutual information. *Journal of Theoretical Biology*, 311:40–45.

[Munk et al., 2016] Munk, C., Isberg, V., Mordalski, S., Harpsøe, K., Rataj, K., Hauser, A., Kolb, P., Bojarski, A., Vriend, G., and Gloriam, D. (2016). GPCRdb: the G protein-coupled receptor database - an introduction. *British Journal of Pharmacology*, 173(14):2195–2207.

[National Human Genome Research Institute, 2016] National Human Genome Research Institute, N. (2016). Dna sequencing costs: Data. [Online; accessed 14-April-2017].

[Pollastri et al., 2002] Pollastri, G., Przybylski, D., Rost, B., and Baldi, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235.

[Qian and Sejnowski, 1988] Qian, N. and Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *Journal of Molecular Biology*, 202(4):865–884.

[Richards et al., 2015] Richards, A. L., Merrill, A. E., and Coon, J. J. (2015). Proteome sequencing goes deep. *Current Opinion in Chemical Biology*, 24:11–17.

[Sandberg et al., 1998] Sandberg, M., Eriksson, L., Jonsson, J., Sjöström, M., and Wold, S. (1998). New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *Journal of Medicinal Chemistry*, 41(14):2481–2491.

[Shah et al., 2008] Shah, A. R., Oehmen, C. S., and Webb-Robertson, B.-J. (2008). SVM-HUSTLE - an iterative semi-supervised machine learning approach for pairwise protein remote homology detection. *Bioinformatics*, 24(6):783–790.

[Shkurin and Vellido, 2016] Shkurin, A. and Vellido, A. (2016). Random forests for quality control in G-protein coupled receptor databases. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 707–718. Springer.

[Wold et al., 1993] Wold, S., Jonsson, J., Sjörström, M., Sandberg, M., and Rännar, S. (1993). DNA and peptide sequences and chemical processes multivariately modelled by principal component analysis and partial least-squares projections to latent structures. *Analytica Chimica Acta*, 277(2):239–253.

[Wu et al., 1992] Wu, C., Whitson, G., McLarty, J., Ermongkonchai, A., and Chang, T.-C. (1992). Protein classification artificial neural system. *Protein Science*, 1(5):667–677.

[Wu et al., 2014] Wu, H., Wang, C., Gregory, K. J., Han, G. W., Cho, H. P., Xia, Y., Niswender, C. M., Katritch, V., Meiler, J., Cherezov, V., et al. (2014). Structure of a class C GPCR metabotropic glutamate receptor 1 bound to an allosteric modulator. *Science*, 344(6179):58–64.

Appendices

Appendix A

Results of classification for different transformations

Model	Classifier	Accuracy	MCC	F-measure
AAC	NB	0.7033	0.6307	0.7046
	RF	**0.8563**	**0.8194**	**0.8536**
	SVM	0.7682	0.7091	0.7677
prot2vec (Swiss-Prot 2014_10)	NB	0.6000	0.5153	0.6070
	RF	0.8583	0.8220	0.8566
	SVM	**0.8901**	**0.8624**	**0.8894**
prot2vec (GPCRdb May2016)	NB	0.5854	0.4931	0.5889
	RF	0.7947	0.7446	0.7971
	SVM	**0.8636**	**0.8294**	**0.8637**
Digram	NB	0.8358	0.7949	0.8375
	RF	0.9106	0.8881	0.9080
	SVM	**0.9172**	**0.8965**	**0.9159**
ACC	NB	0.8430	0.8064	0.8455
	RF	0.8848	0.8562	0.8775
	SVM	**0.9278**	**0.9095**	**0.9271**

Table A.1: Classification results for 2011 database using different transformations

A

Model	Classifier	Accuracy	MCC	F-measure
AAC	NB	0.9550	0.9271	0.9551
	RF	0.9799	0.9675	0.9796
	SVM	**0.9846**	**0.9749**	**0.9844**
prot2vec (Swiss-Prot 2014_10)	NB	0.8118	0.7229	0.8207
	RF	0.9716	0.9535	0.9713
	SVM	**0.9882**	**0.9809**	**0.9881**
prot2vec (GPCRdb May2016)	NB	0.8615	0.7972	0.8688
	RF	0.9775	0.9642	0.9775
	SVM	**0.9858**	**0.9775**	**0.9858**
Digram	NB	0.9811	0.9688	0.9808
	RF	0.9893	0.9821	0.9893
	SVM	**0.9941**	**0.9904**	**0.9941**
ACC	NB	0.9799	0.9673	0.9798
	RF	0.9870	0.9793	0.9866
	SVM	**0.9964**	**0.9944**	**0.9964**

Table A.2: Classification results for May 2016 database using different transformations

Model	Classifier	Accuracy	MCC	F-measure
AAC	NB	0.9594	0.9368	0.9598
	RF	0.9829	0.9724	0.9829
	SVM	**0.9957**	**0.9931**	**0.9957**
SWISSPROT	NB	0.9113	0.8619	0.9132
	RF	0.9850	0.9758	0.9849
	SVM	**0.9925**	**0.9875**	**0.9925**
prot2vec (GPCRdb Sept2016)	NB	0.9808	0.9692	0.9809
	RF	0.9893	0.9826	0.9893
	SVM	**0.9925**	**0.9875**	**0.9925**
Digram	NB	0.9893	0.9826	0.9893
	RF	**0.9947**	**0.9913**	**0.9948**
	SVM	0.9925	0.9875	0.9923
ACC	NB	0.9338	0.8933	0.9323
	RF	0.9712	0.9526	0.9708
	SVM	**0.9743**	**0.9584**	**0.9743**

Table A.3: Classification results for September 2016 database using different transformations

Appendix B

Hyper-parameters comparison for prot2vec trained on GPCRdb

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	3	100	5	0.9925	0.9879	0.9925
			10	0.9915	0.9863	0.9914
			20	0.9925	0.9879	0.9925
			25	0.9915	0.9861	0.9915
			30	0.9936	0.9898	0.9936
			40	0.9947	0.9913	0.9946
CBOW	3	100	5	0.9925	0.9880	0.9925
			10	0.9936	0.9898	0.9935
			20	0.9915	0.9862	0.9914
			25	0.9947	0.9912	0.9947
			30	0.9947	0.9913	0.9947
			40	0.9914	0.9862	0.9914
Skip-gram	3	200	5	0.9925	0.9876	0.9925
			10	0.9947	0.9911	0.9946
			20	0.9936	0.9895	0.9936
			25	0.9936	0.9896	0.9936
			30	0.9925	0.9879	0.9925
			40	0.9936	0.9898	0.9935
CBOW	3	200	5	0.9957	0.9931	0.9957
			10	0.9957	0.9929	0.9957
			20	0.9936	0.9895	0.9936
			25	0.9925	0.9880	0.9924
			30	0.9936	0.9897	0.9936
			40	0.9947	0.9912	0.9946
Skip-gram	3	300	5	0.9957	0.9930	0.9957
			10	0.9957	0.9930	0.9957
			20	0.9947	0.9913	0.9946
			25	0.9936	0.9898	0.9936
			30	0.9947	0.9914	0.9946
			40	0.9947	0.9912	0.9946
CBOW	3	300	5	0.9957	0.9929	0.9957
			10	0.9936	0.9895	0.9935
			20	0.9936	0.9896	0.9936
			25	0.9936	0.9898	0.9936
			30	0.9947	0.9913	0.9946
			40	0.9936	0.9897	0.9936

Table B.1: Comparison of distributed representation hyper-parameters for September 2016 database using SVM classifier, trained on GPCRdb same version, n-gram size = 3

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	4	100	5	0.9882	0.9804	0.9882
			10	0.9904	0.9845	0.9904
			20	0.9914	0.9864	0.9915
			25	0.9936	0.9899	0.9936
			30	0.9925	0.9880	0.9926
			40	0.9957	0.9931	0.9957
CBOW	4	100	5	0.9851	0.9764	0.9850
			10	0.9883	0.9809	0.9882
			20	0.9872	0.9796	0.9872
			25	0.9925	0.9875	0.9925
			30	0.9925	0.9883	0.9925
			40	0.9914	0.9860	0.9915
Skip-gram	4	200	5	0.9883	0.9806	0.9883
			10	0.9925	0.9878	0.9925
			20	0.9904	0.9842	0.9903
			25	0.9936	0.9896	0.9936
			30	0.9947	0.9914	0.9947
			40	0.9947	0.9915	0.9946
CBOW	4	200	5	0.9861	0.9773	0.9860
			10	0.9882	0.9809	0.9883
			20	0.9872	0.9796	0.9871
			25	0.9925	0.9880	0.9924
			30	0.9883	0.9817	0.9883
			40	0.9904	0.9846	0.9904
Skip-gram	4	300	5	0.9893	0.9828	0.9892
			10	0.9914	0.9862	0.9915
			20	0.9915	0.9861	0.9914
			25	0.9914	0.9861	0.9914
			30	0.9936	0.9894	0.9936
			40	0.9957	0.9931	0.9957
CBOW	4	300	5	0.9850	0.9757	0.9850
			10	0.9893	0.9827	0.9893
			20	0.9904	0.9846	0.9905
			25	0.9936	0.9898	0.9936
			30	0.9904	0.9843	0.9904
			40	0.9925	0.9878	0.9926

Table B.2: Comparison of distributed representation hyper-parameters for September 2016 database using SVM classifier, trained on GPCRdb same version, n-gram size = 4

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	2	100	5	0.9957	0.9932	0.9957
			10	0.9947	0.9914	0.9947
			20	0.9925	0.9878	0.9925
			25	0.9957	0.9930	0.9957
			30	0.9925	0.9881	0.9926
			40	0.9936	0.9894	0.9936
CBOW	2	100	5	0.9957	0.9931	0.9957
			10	0.9947	0.9911	0.9946
			20	0.9925	0.9880	0.9925
			25	0.9936	0.9899	0.9936
			30	0.9925	0.9876	0.9925
			40	0.9957	0.9931	0.9957
Skip-gram	2	200	5	0.9915	0.9863	0.9914
			10	0.9925	0.9876	0.9925
			20	0.9947	0.9910	0.9947
			25	0.9947	0.9917	0.9947
			30	0.9936	0.9890	0.9936
			40	0.9947	0.9913	0.9947
CBOW	2	200	5	0.9925	0.9880	0.9925
			10	0.9936	0.9894	0.9936
			20	0.9957	0.9934	0.9958
			25	0.9968	0.9948	0.9968
			30	0.9936	0.9895	0.9936
			40	0.9947	0.9913	0.9947
Skip-gram	2	300	5	0.9947	0.9916	0.9946
			10	0.9936	0.9896	0.9936
			20	0.9947	0.9912	0.9947
			25	0.9947	0.9913	0.9947
			30	0.9947	0.9911	0.9946
			40	0.9947	0.9913	0.9946
CBOW	2	300	5	0.9936	0.9896	0.9936
			10	0.9947	0.9911	0.9947
			20	0.9957	0.9931	0.9958
			25	0.9947	0.9914	0.9946
			30	0.9957	0.9934	0.9958
			40	0.9957	0.9933	0.9957

Table B.3: Comparison of distributed representation hyper-parameters for September 2016 database using SVM classifier, trained on GPCRdb same version, n-gram size = 2

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	5	100	5	0.9936	0.9896	0.9936
			10	0.9915	0.9864	0.9913
			20	0.9915	0.9862	0.9914
			25	0.9915	0.9862	0.9915
			30	0.9957	0.9932	0.9957
			40	0.9957	0.9933	0.9958
CBOW	5	100	5	0.9904	0.9845	0.9904
			10	0.9957	0.9929	0.9957
			20	0.9925	0.9880	0.9926
			25	0.9925	0.9879	0.9925
			30	0.9947	0.9913	0.9947
			40	0.9904	0.9845	0.9903
Skip-gram	5	200	5	0.9936	0.9895	0.9936
			10	0.9957	0.9929	0.9957
			20	0.9936	0.9898	0.9936
			25	0.9968	0.9947	0.9968
			30	0.9936	0.9897	0.9936
			40	0.9947	0.9915	0.9947
CBOW	5	200	5	0.9883	0.9811	0.9883
			10	0.9893	0.9830	0.9894
			20	0.9883	0.9812	0.9882
			25	0.9893	0.9829	0.9895
			30	0.9936	0.9897	0.9936
			40	0.9893	0.9829	0.9893
Skip-gram	5	300	5	0.9915	0.9863	0.9914
			10	0.9925	0.9880	0.9925
			20	0.9947	0.9915	0.9947
			25	0.9947	0.9915	0.9947
			30	0.9947	0.9915	0.9947
			40	0.9947	0.9915	0.9947
CBOW	5	300	5	0.9861	0.9777	0.9861
			10	0.9915	0.9864	0.9915
			20	0.9850	0.9760	0.9850
			25	0.9904	0.9846	0.9904
			30	0.9883	0.9812	0.9884
			40	0.9883	0.9811	0.9881

Table B.4: Comparison of distributed representation hyper-parameters for September 2016 database using SVM classifier, trained on GPCRdb same version, n-gram size = 5

Appendix C

Hyper-parameters comparison for prot2vec trained on Swiss-Prot

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	3	100	5	0.8828	0.8533	0.8827
			10	0.8901	0.8626	0.8895
			20	0.8841	0.8553	0.8844
			25	0.8682	0.8350	0.8675
			30	0.8742	0.8431	0.8739
			40	0.8815	0.8516	0.8810
CBOW	3	100	5	0.8980	0.8725	0.8981
			10	0.8907	0.8638	0.8901
			20	0.8947	0.8690	0.8942
			25	0.8828	0.8538	0.8816
			30	0.8841	0.8555	0.8832
			40	0.8980	0.8731	0.8979
Skip-gram	3	200	5	0.9000	0.8750	0.9001
			10	0.8993	0.8742	0.8990
			20	0.8980	0.8726	0.8982
			25	0.8940	0.8677	0.8937
			30	0.9066	0.8831	0.9067
			40	0.8907	0.8632	0.8906
CBOW	3	200	5	0.9046	0.8815	0.9037
			10	0.9053	0.8819	0.9052
			20	0.9139	0.8923	0.9139
			25	0.9073	0.8839	0.9073
			30	0.9086	0.8858	0.9081
			40	0.9079	0.8850	0.9073
Skip-gram	3	300	5	0.9113	0.8889	0.9103
			10	0.9099	0.8873	0.9094
			20	0.9166	0.8955	0.9164
			25	0.9093	0.8864	0.9094
			30	0.9040	0.8797	0.9035
			40	0.8947	0.8684	0.8952
CBOW	3	300	5	0.9106	0.8880	0.9106
			10	0.8987	0.8733	0.8990
			20	0.9007	0.8753	0.9007
			25	0.9119	0.8897	0.9114
			30	0.9013	0.8763	0.9010
			40	0.9079	0.8847	0.9079

Table C.1: Comparison of distributed representation hyper-parameters for 2011 database using SVM classifier, trained on Swiss-Prot 2017_04, n-gram size = 3

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	4	100	5	0.8795	0.8493	0.8788
			10	0.8781	0.8474	0.8776
			20	0.8735	0.8422	0.8737
			25	0.8589	0.8239	0.858
			30	0.8675	0.8340	0.8675
			40	0.8642	0.8298	0.8637
CBOW	4	100	5	0.8675	0.8339	0.8674
			10	0.8689	0.8354	0.8681
			20	0.8828	0.8530	0.8824
			25	0.8715	0.8387	0.8707
			30	0.8854	0.8564	0.8850
			40	0.8927	0.8653	0.8923
Skip-gram	4	200	5	0.8954	0.8692	0.8948
			10	0.8762	0.8455	0.8760
			20	0.9040	0.8801	0.9045
			25	0.8947	0.8684	0.8952
			30	0.8927	0.8658	0.8931
			40	0.8934	0.8667	0.8933
CBOW	4	200	5	0.8980	0.8722	0.8981
			10	0.8940	0.8671	0.8927
			20	0.8914	0.8640	0.8910
			25	0.8927	0.8655	0.8923
			30	0.8947	0.8678	0.8938
			40	0.9013	0.8761	0.9006
Skip-gram	4	300	5	0.9046	0.8806	0.9046
			10	0.9132	0.8913	0.9128
			20	0.8887	0.8612	0.8881
			25	0.9073	0.8838	0.9062
			30	0.8954	0.8689	0.8942
			40	0.8960	0.8699	0.8957
CBOW	4	300	5	0.9060	0.8821	0.9054
			10	0.9079	0.8850	0.9078
			20	0.9172	0.8963	0.9166
			25	0.9205	0.9006	0.9199
			30	0.9113	0.8886	0.9099
			40	0.9093	0.8862	0.9088

Table C.2: Comparison of distributed representation hyper-parameters for 2011 database using SVM classifier, trained on Swiss-Prot 2017_04, n-gram size = 4

Model	N-gram	Size	Window	Accuracy	MCC	F-measure
Skip-gram	2	100	5	0.8815	0.8515	0.8804
			10	0.8887	0.8604	0.8876
			20	0.8834	0.8542	0.8822
			25	0.8709	0.8379	0.8698
			30	0.8834	0.8543	0.8827
			40	0.8828	0.8527	0.8812
CBOW	2	100	5	0.9060	0.8822	0.9054
			10	0.9086	0.8856	0.9087
			20	0.8993	0.8740	0.8989
			25	0.8993	0.8740	0.8997
			30	0.9060	0.8821	0.9055
			40	0.9013	0.8761	0.9011
Skip-gram	2	200	5	0.9066	0.8829	0.9058
			10	0.9073	0.8837	0.9067
			20	0.8987	0.8728	0.8976
			25	0.8901	0.8625	0.8893
			30	0.9033	0.8786	0.9022
			40	0.8993	0.8736	0.8980
CBOW	2	200	5	0.9132	0.8913	0.9124
			10	0.9126	0.8905	0.9124
			20	0.9132	0.8915	0.9130
			25	0.9192	0.8988	0.9187
			30	0.9073	0.8837	0.9065
			40	0.9079	0.8845	0.9070
Skip-gram	2	300	5	0.9093	0.8864	0.9088
			10	0.9192	0.8989	0.9186
			20	0.9073	0.8837	0.9068
			25	0.9086	0.8857	0.9083
			30	0.9066	0.8836	0.9066
			40	0.9033	0.8791	0.9028
CBOW	2	300	5	0.9060	0.8821	0.9051
			10	0.9119	0.8900	0.9114
			20	0.9232	0.9038	0.9231
			25	0.9132	0.8912	0.9124
			30	0.9166	0.8954	0.9161
			40	0.9146	0.8930	0.9143

Table C.3: Comparison of distributed representation hyper-parameters for 2011 database using SVM classifier, trained on Swiss-Prot 2017_04, n-gram size = 2