THE EFFECT OF METHODOLOGICAL FLEXIBILITY ON MEMBRANE PROTEIN CLASSIFICATION

Hamidreza Heidarzadeh

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Science (Computer Science)
Concordia University
Montréal, Québec, Canada

July 2021

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Hamidreza Heidarzadeh**

Entitled: **The effect of methodological flexibility on membrane protein classification**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Dr. Yiming Xiao _____ Chair

Dr. Gregory Butler _____ Examiner

_____ Examiner

_____ Examiner

Dr. Tristan Glatard _____ Supervisor

Approved Dr. Lata Narayanan _____
Chair of Department or Graduate Program Director

July, 16 20 21 Dr. Mourad Debbabi _____
Dean
Faculty of Engineering and Computer Science

Abstract

The effect of methodological flexibility on membrane protein classification

Hamidreza Heidarzadeh

Reproducibility, the ability to reproduce computational results using identical data and software, is a cornerstone of the scientific methodology. However, through the past decade, several studies revealed a widespread lack of results' reproducibility, to the point that the existence of a reproducibility crisis is now acknowledged in various fields. In Machine Learning, given the flexibility available in various phases of constructing a computational model, the experiments are not immune to reproducibility issues either. In case of imbalance learning for problems with multiple classes, the problem is even more severe since there are more parameters in play for constructing a model. The resulting reproducibility challenges have implications in various disciplines including bioinformatics, the primary focus of our study.

Researchers have already taken counter-measures proposing various recommendations for having results' reproducibility in this domain of study. Some conferences have even adopted new measures in that regard. Following those guidelines could ensure reproducibility to an agreeable degree in balanced problems. In this work we demonstrate that in an imbalanced scenario, even in its basic form, a study report with a fair amount of details, could reproduce a wide range of results if methodological flexibility is permitted.

Acknowledgments

First and foremost, I would like to thank my supervisor Dr Tristan Glatard, who supported me throughout my thesis process with his courageous words, patience and knowledge. I consider myself lucky to have such an exceptional and friendly supervisor. A great amount of appreciation to Dr Gregory Butler for his support and help with the initial bioinformatics-related concepts of my work. With special thanks to Greg Kiar a Phd student of Dr Tristan Glatard and Munira Alballa a Phd student of Dr Gregory Butler who also helped me out in the process. I would especially like to thank my wife Sophie who has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Reproducibility, the ability to reproduce computational results using identical data and software [73], is a cornerstone of scientific methodology. In the past decade, however, several studies revealed a widespread lack of results reproducibility, to the point that the existence of a reproducibility crisis is now acknowledged in various fields [5].

To improve the results' reproducibility, counter-measures were identified and the movement towards examining and enhancing the reliability of research was expanded [10]. Scientists addressed the issue by defining reproducibility-specific terms and terminologies (e.g. methodological reproducibility, replicability, robustness, etc.) and providing guidelines [18, 32] and best practices [63, 81] for having a reproducible research. It was then suggested that the scientific community needs to develop a "culture of reproducibility" for computational science and require it for the published claims [72]. Given the methodological flexibility associated with computational experiments, a reproducible study through this culture is required to share the analytical data sets (original raw or processed data), the relevant metadata, the analytical code(s) and the related software [92].

Given the available flexibility in data pre-processing, train/test set definitions, algorithm selection and parametrization, library implementations, and evaluation metrics, machine learning experiments are not immune to reproducibility issues either [78]. In case of imbalance learning for problems with multiple classes, the problem is even more severe since there are more parameters in play for constructing a computational model. The resulting reproducibility challenges have implications in various

disciplines including bioinformatics, the primary focus of our study.

Membrane proteins are vital molecules that act as gatekeepers to a cell. It is estimated that one in every three proteins found in a cell is a membrane protein [15]. In a living organism, they play several important roles such as: cell signaling, transportation of molecules and nutrients across the membrane of a cell, energy production and foreign bodies recognition [50]. Considering the contribution of these molecules to cell functionalities, defects in membrane proteins could lead to different diseases [34].

Today, almost half of the drugs target these proteins [17]. Due to the hydrophobic surfaces of these molecules and their lack of conformational stability, using conventional experimental methods for annotation of these proteins are time-consuming, costly and sometimes impossible. So, researchers have turned into computational intelligent techniques for annotation and prediction of the structure and functionalities of the membrane proteins [35, 36, 68, 82, 16]. Year after year, with advances in technology, researchers can use cheaper and faster sequencing methods (more data for their problems), new computational intelligent techniques and software tools. In search for more accurate and generalizable results, reproducible studies allows applying the same technique on new datasets and new techniques on the initial ones.

This work presents a reproducibility study of a classification problem with an imbalanced dataset involving multiple classes which is a common case when dealing with proteins in bioinformatics. We report our attempts to reproduce a membrane protein classification problem with an imbalanced dataset [62], showing the impact of methodological flexibility (the flexibility associated with implementing the original study experiment using the same data, analysis tools and through the same environment to obtain the same result) on classification performance, and deriving best practices to report Machine Learning results for similar problems. We explore methodological options related to data preparation, hyperparameter tuning, classifier implementation, aggregation of binary classifiers for multi-class classification, and prediction method for final labels. The resulting variations emulate the range of results that might be obtained by reasonably skilled experimenters aiming at reproducing the same model.

The work in [62] is a reference contribution that we selected given the availability of its input data, the quality of its writing and methods reporting, and its overall impact in the field.

Chapter 2

Background

2.1 Reproducible Experiment

Reproducing an experiment is a common practice by which scientists verify claims and apply new ideas to their own domain of study. Reproducible research saves a great amount of time and budget when another scientist picks a study and tries to recreate the same experiment. Overall, it avoids "reinventing the wheel."

In the scientific community, the reproducibility concept is often approached from two different perspectives. Some researchers view it as a tool for verifying a claim. This approach is required when for example new findings are planned to be applied to real life problems. There is also a complementary perspective through which the scientists view reproducibility as a tool for improving an existing method, adapting it to new requirements and needs, or applying the methodology to a new domain of research.

Through recent years, along with the increase in the amount of data available to the scientific community, more high performance computational resources have also become accessible to researchers. The combination of these two, has led to both new discoveries and research opportunities while introducing new challenges. The traditional scientific paper of an experiment (being designed to include all the necessary information of a study) does not appear to be able to contain all the required information of the data-intensive and computationally-intensive methods of the modern studies. While it's challenging to reproduce a study from scratch using the provided information on the study report, not having proper enough details, adds up to the

problem and endangers the reproducibility of a claim for verification, adjustment or application purposes.

In an attempt to reproduce a paper on protein classification on an imbalance dataset (which is a common case in this domain) from the second perspective mentioned above, we obtained a wide range of results with the available details on the study report. The authors had shared a fair amount of details on the paper which does not seem to be enough for problems with an imbalanced dataset. In such problems, for predicting the final labels, a researcher needs to take some extra steps which involves more parameters throughout the whole process. We believe in similar problems, the study report should also include details on these important parameters to enable reproducibility.

In our study, we approach the methodological reproducibility of the classification problems with an imbalance dataset from the perspective mentioned above. We address the potential underlying reproducibility-related issues of the similar studies and how those could lead to a wide range of results in reproduction. In this section, we will briefly provide our adopted definition of some common reproducibility terms to further clarify the upcoming discussions.

2.1.1 Taxonomy and Terms

Reproducibility

The term "reproducibility" came about in the early 90s in computational science by John Claerbout and through the context of transparency. He provided a set of procedures on the paper allowing the reader to see the entire process from the data to figures and tables [23]. The concept has been carried forward into other domains (e.g. bioinformatics, economics, etc) afterwards [47, 74, 83]. since then and has been transformed into the context we use today.

In its modern context, the U.S. National Science Foundation (NSF) [18] defines reproducibility as "the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator. That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results."

Reproducibility, Replicability, Robustness and Generalizability

For the Reproducible, Replicable, Robust and Generalizable studies, we adopt the following definitions from [75]. A study is **reproducible** when the same results could be obtained using the same data, analytical tools and through the same environment.

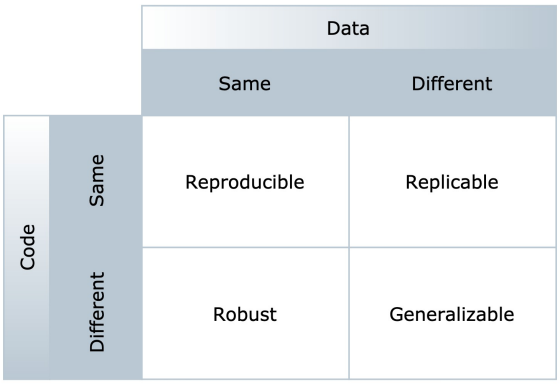

Figure 1: Reproducible, Replicable, Robust and Generalizable research reproduced from [90]

For achieving reproducibility, all the required information for re-doing the experiment should be available on the study report. For example, when collecting the initial data for the study, if the data is collected within a predetermined framework or it is being cured before being used, then all the related information needs to be included. For a machine learning problem, the same rule applies to data pre-processing, feature engineering, classification algorithm, result generation, performance metrics calculation and any other possible involved process.

A **replicable** study is the one that could re-generate the same results, if the same analytical tools are applied to a different set of data (relevant data being collected and cured through the same framework, distribution and method) and through the same environment.

A study is **robust**, if the same results could be achieved by applying different analysis (e.g. re-implementing the code in a different environment or using the same algorithm from a different well-recognized library) to the same set of data.

A **generalizable** study leads to the same results if a different analysis is applied to a different set of data (in such a way mentioned above).

Figure 1 illustrates reproducible, replicable, robust and generalizable research in

regards to data and analysis. If a study is not reproducible, then replicability, robustness and generalizability of that experiment could not be assessed. Through this study, our focus is on reproducible research as being addressed in this section.

Method reproducibility, Result reproducibility, Inferential reproducibility

Goodman et al [8] explain that the reproducibility and replicability definitions being provided by the U.S.National Science Foundation (NSF) [18], do not provide a clear operational criteria for making a distinction in between these two concepts. Based on the published definitions, one can not draw a clear line in between what constitutes a successful reproduction or replication. To address the underlying construct of a reproducibility study, they suggest using three following terms instead of reproducibility. They believe these terms can make a more meaningful distinction in between various interpretations of reproducibility.

Method reproducibility is the ability to implement the original study experiment using the same data, analysis tools and through the same environment to obtain the same result. In definition and practice, it is the same as the original reproducibility term being defined in the last section.

Results reproducibility (previously defined as replicability) is the ability to re-generate the same results in a new (independent) study by following the same experimental procedures being provided in the original study on a new set of data.

Inferential reproducibility is the ability to achieve qualitatively similar conclusions from reanalysis or replication of the original study.

In our study, we address the methodological reproducibility of the problems with an imbalance datasets as being described in this section.

Low, Medium and High Reproducibility

According to the current standards (being placed to ensure the reproducibility of a claim) for a paper to be reproducible (adopted by conferences like NeurIPS [75]), all the details of the study should be included in the submitted paper. Also, the data, programs and any involved software needs to be submitted along with the study report. But sometimes due to some restrictions (e.g. confidentiality [25]) submitting all the required materials are not possible. Thus, the "reproducible" term, can not describe how reproducible an experiment is. When it comes to older studies, this

problem is even more visible as some involved materials are not accessible anymore. To address this problem [86] proposes three following terms by which the amount of reproducibility could be expressed.

Low reproducibility: A low reproducible study is one that has only submitted the experiment report for the claim. The paper needs to contain all the correspondent details for an independent reproduction of the same experiment from scratch. The reproduced work needs to generate the same results.

Medium reproducibility: A study with medium level of reproducibility shares the codes and data along with the experiment report. The submitted data and code should permit an independent reproduction of the experiment leading to the same conclusion.

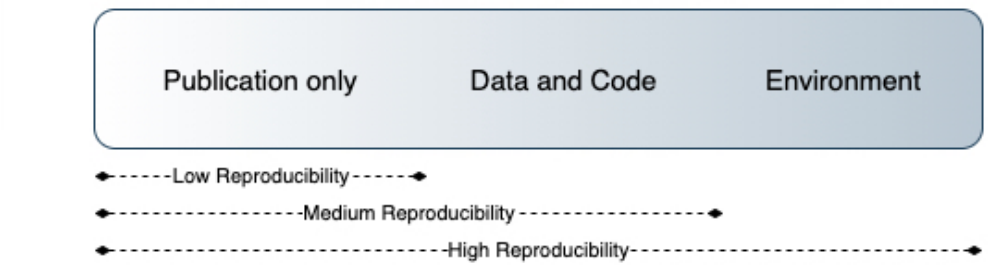

Figure 2: Studies with Low, Medium and High degree of Reproducibility

High reproducibility: A study is highly reproducible if the environment information through which the experiment has been conducted, is also shared along with the data, code and the submitted paper. By definition, the environment includes all the libraries and dependencies necessary for a program to be run on a new machine. This level of reproducibility has been referred to as "linked and executable code and data" in [72].

Figure 2, shows the reproducibility spectrum through which the above three terms could be addressed. In our study of classification problems with an imbalance dataset, we address the potential underlying issues of the studies with low degree of reproducibility and how those could lead to a wide range of results in reproduction.

2.1.2 Reproducibility Crisis

As being mentioned earlier in this section, the "reproducibility" term has been around for a while meaning that researchers were always concerned about the results of the studies in their domain of interest. We can track this back to the early 90s when John Claerbout in his book, "Earth Soundings Analysis" [22] claimed that few published results are reproducible in practice.

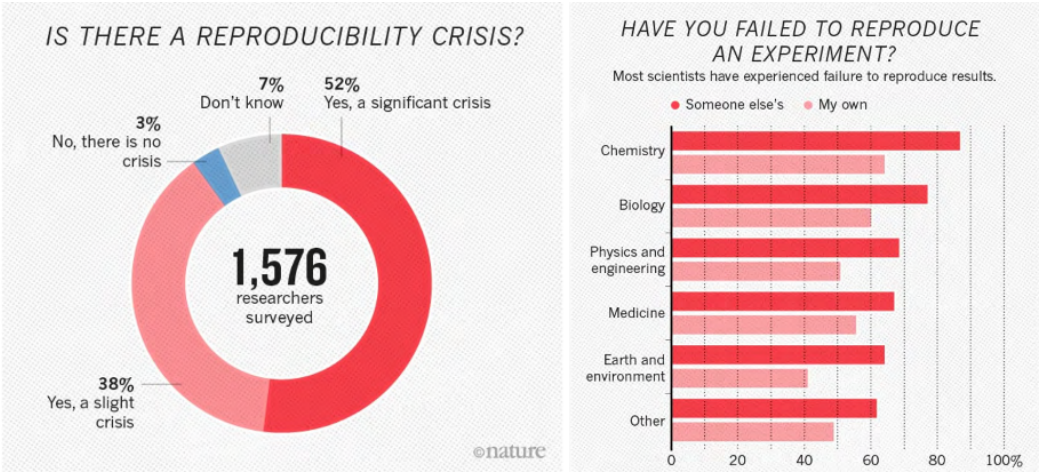

Figure 3: Reproducibility crisis related results of the survey conducted by Nature in 2016 extracted from [5]

With the emergence of disappointing results from large scale reproducibility projects in various domains, the term "Reproducibility crisis" gained currency in publications over the last decade (e.g. [69]). In a study being conducted by the Nature journal in 2016 [5], more than 70% of surveyed scientists reported failure in their attempt to reproduce another scientist's experiments, and more than half (52%) believed that science is facing a "reproducibility crisis". The cause of the problem covers a wide range of subjects from pressure to publish and selective reporting to poor analysis. Most of the researchers believed that failure to reproduce published results does not mean that the result is probably wrong, and most say that they still trust the published literature. Figure 3 shows reproducibility crisis related results of the survey conducted by Nature in 2016 .

On the same subject in machine learning, Nicolas Rougier, a computational neuroscientist says "I think people outside the field might assume that because we have

code, reproducibility is kind of guaranteed" but "far from it" he said at France's National Institute for Research in Computer Science and Automation in Bordeaux [42]. One of the common problems is that due to some restrictions (e.g. confidentiality), the dataset or source code used in the study is not open-sourced. For example in a reproducibility study on 400 presented papers at two top AI conferences conducted by Odd Erik Gundersen, 6% of the presenters had released their codes, third of them had shared their data, and 50% had shared their pseudocodes [38].

Through recent years, the results of similar studies across different domains, have raised great concerns in the scientific community leading to various works providing reproducibility-related lexicons, guidelines, and platforms for measuring and improving the reproducibility of the research results. In a move aimed at providing reproducible research, some conferences even put new mandates in places for submitting the publications [75, 32, 86]. In this work, we address the potential reproducibility-related issues of the classification problems with imbalanced datasets.

2.1.3 Reproducibility-related works

Throughout the past decade, reproducibility-related studies have received a notable amount of attention in the scientific community. Figure 4 shows the publications recorded in the Scopus that have, in the title or abstract, at least one of the reproducibility-related expressions [32]. The studies have been conducted in various domains from different perspectives to define the terms, address the cause of the problem, create frameworks, platforms, guidelines and mandated to encourage reproducible research in practice. In this section, we will mention some of the related works in the field.

Around the same time when Claerbout claimed that few published results are reproducible [22], a computer scientist, Donald Knuth, introduced the concept of Literate Programming [49]. In Literate Programming, computer code is embedded within the program's documentation making it more understandable for humans. The consolidated standards of reporting trials or consort [http://www.consort-statement.org], published a set of guidelines In 1996 to fix problems associated with inadequate reporting of randomized controlled trials.

In 2004, the International Committee of Medical Journal [http://www.icmje.org] announced that they would not publish a clinical trial without registration. The updated publication's criterias includes conditions like "Manuscripts submitted to

ICMJE journals that report the results of clinical trials must contain a data sharing statement". In a move towards reproducible research, the Journal of Biostatistics [https://academic.oup.com/biostatistics] began marking accepted papers based on the standards of reproducibility. also encouraged reproducible practices of authors submissions. For example, D means the study data is freely available, A C means the code is available and R means the paper is reproducible.

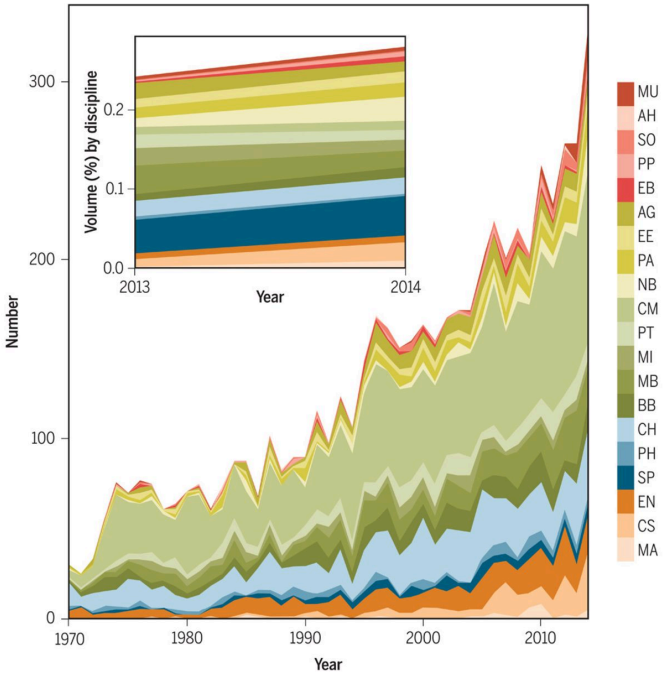

Figure 4: publications recorded in the Scopus that have, in the title or abstract, at least one of the reproducibility-related expressions extracted from [32]

In research on 500 research papers in 2011 [2], 30% of the submitted papers did not adopt any data sharing policy. Among the ones who adhered to the data sharing instructions, 91% deposited only the specific required data type and 9% made the full primary raw data accessible online. The Open Science Collaboration in 2015 [66] announced that only 30-50 percent of the results being taken from more than 100 studies were reproducible. U.S. National Science Foundation (NSF) [18] in 2015, addresses the reproducibility problem providing definitions for reproducibility, replicability, robustness and generalizability.

ASCB's survey on reproducibility in 2015 [4] showed that almost 70% of the questioned researchers were unable to replicate the results of studies they were interested

in. Nature journal in 2016 [5], addresses the "reproducibility crisis" by conducting a survey. The study showed that more than 70% of surveyed scientists reported failure in their attempt to reproduce another scientist's experiments. In 2106, Mark D. Wilkinson et al [91] wrote a paper on FAIR principles, providing guidelines for implementing Findable, Accessible, interoperable and Reusable research in practice.

Steven N. Goodman et al in 2016 [32] published a paper suggesting new terms for reproducibility. They believed the underlying construct of the reproducibility studies could be addressed by using these new terminologies. In 2017, Babatunde K. Olorisade et al [65], published a paper on practicing reproducibility in machine Learning studies by providing an example in the text mining domain. Through the same year Hans E. Plesser [76] also published a paper in "Frontiers in Neuroinformatics" clearing up on the various in-use definitions of the reproducibility-related term in the field.

In 2018, Joelle Pineau et al [75] provided guidelines for improving reproducibility in machine learning research. She also published a "The Machine Learning Reproducibility Checklist" designed to be used simultaneously with the ML code submission checklist for NeurIPS. In his paper, Rachael Tatman [86] provided a new taxonomy for reproducibility for machine learning research in 2018 by which the amount of reproducibility could be expressed.

In 2019, Andrew L. Beam et al [9] published a paper on the reproducibility of the machine learning models in health care discussing the unique challenges for the problems using machine learning models to predict the outcome. Since a machine learning model should be reproduced, and ideally replicated, before it is deployed in a clinical setting, they highly encourage adopting reproducible research practices for the studies in this field. Through the same year, Matthew B.A. McDermott et al [61] also published a paper on reproducibility in machine learning for health (ML4H) providing a comparison between the available amount of reproducibility in the field and other machine learning-based domains of studies. They also discuss the causes of the problem, the unique challenges associated with the problems in this field (e.g. confidentiality) providing guidelines over how to improve reproducibility considering the current challenges.

The scientific community has also initiated and developed various softwares and platforms (such as scikit-learn [71], R [19], etc.) for software development in general

and statistical computing in specific (in our case) through the reproducible research framework. All these cumulative efforts have been done in an aim to provide all the researchers with free, well-documented and publicly accessible softwares leading to creation of standard practices and reproducible studies.

2.2 Bioinformatics Context

In July 2000, the NIH Biomedical Information Science and Technology Initiative Consortium released a document [41] through which they defined bioinformatics as "Research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data."

Bioinformatics is a highly interdisciplinary field. Figure 5 shows the Interaction of various disciplines that have contributed to the formation of bioinformatics [8]. It conceptualizes biology in terms of macromolecules and aims to extract knowledge from the information associated with these molecules on a large-scale. It extracts the intended information by applying "informatics" techniques (derived from various disciplines such as statistics, computer science, maths, linguistics, etc.) to the biological data. Depending on the goal of the study, the biological data could be collected from sources such as information stored in the genetic code, experimental results, patient statistics, scientific literature, etc. [64, 58].

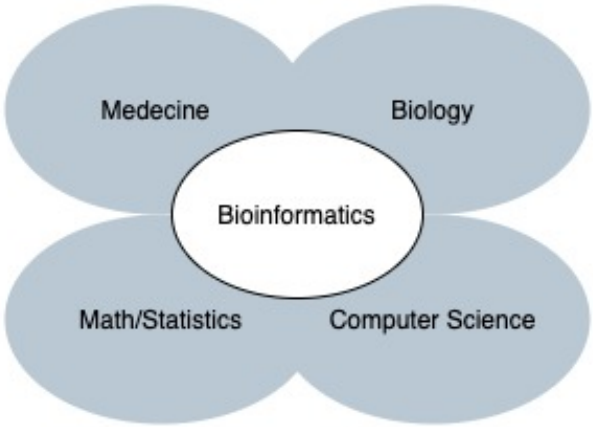

Figure 5: Disciplines contributed to the formation of bioinformatics reproduced from [8]

With the exponential growth in the amount of available biological data, research in bioinformatics should be able to address method development for both data management (e.g. storage, retrieval of data) and extraction of useful information from these data (data analysis). One of the main challenges in bioinformatics is development of the tools and methods capable of transforming data into biological knowledge which is the focus of the second area being mentioned above. These tools and methods should be able to extract knowledge in the form of testable models. By this simplifying abstraction that constitutes a model, we will be able to predict the behavior of the system. In modern biology and medicine, bioinformatics is essential and has many practical applications in different areas of those fields.

One of the popular data analysis tools by which researchers try to predict the behavior of a system is "Machine Learning". It is a direct descendant of statistical model fitting. Machine learning tries to extract useful information from a set of data by building good probabilistic models. However, according to [6],

> the particular twist behind machine learning, is to automate this process as much as possible, often by using very flexible models characterized by large numbers of parameters, and to let the machine take care of the rest

In a problem, the term "learning" refers to running a computer program to induce a model by using training data or past experience. Machine-learning approaches are best suited for areas where there is a large amount of data but little theory which is exactly the situation in computational molecular biology [20].

There are various biological domains where computational methods and techniques are applied for data analysis and knowledge extraction from the data. Pedro Larrañaga et al. [54] have classified those problems into following domains: genomics, proteomics, microarrays, systems biology, evolution and text mining. Figure 6 shows the main biological problems where machine learning and computational methods are being applied. The "other applications" category includes all the remaining problems besides the ones mentioned above.

In the proteomics domain, supervised machine learning techniques are used for protein structure prediction and protein function prediction [55, 62]. As a subcategory of machine learning, supervised learning is defined by its use of labeled datasets for training the algorithm that is supposed to classify data or predict the outcome accurately. Unlike supervised learning, unsupervised learning tries to discover patterns

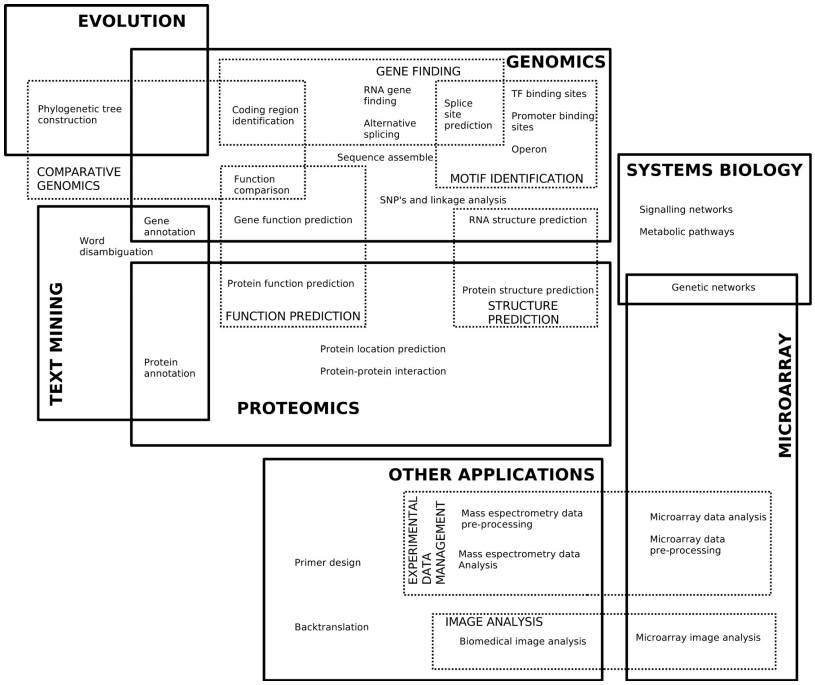

Figure 6: Main biological problems where computational methods are applied extracted from [54]

in an unlabeled set of data.

Since extraction and annotation of the protein sequences are a time consuming and expensive process, the researchers often have to build and train models on imbalanced datasets (imbalanced learning). As we mentioned earlier, bioinformatics is a highly interdisciplinary field. With the new achievements in the related fields (e.g. sequencing, machine learning), researchers should be able to apply new techniques to the fairly older problems to see if those new techniques could improve results or solve the problem. In the case of machine learning, new achievements can provide the researchers with more data which can improve learning and eventually the model performance.

Due to the reasons mentioned above, along with the fact that bioinformatics research in some areas directly deals with humans health (e.g. medicine), like some other fields of studies with similar characteristics (e.g. psychology), reproducible research has received a notable amount of attention through the past decade [86, 61]. Reproducibility plays an important role as a tool for both claim verification and research improvement and adjustment.

In the case of imbalanced learning (which is the focus of this study), a researcher needs to take some extra steps (compared to machine learning problems on a balanced

set of data) for building the model and predicting the final outcome. Since there are more parameters involved, we believe in similar problems, the study report should also include details on these important parameters to ensure high degree of reproducibility. Otherwise, older studies need to be remodeled and programmed from scratch which could be time consuming, expensive and sometimes may not be possible since some resources being used in the initial study may not be available anymore.

The following section will provide a brief introduction to the supervised classification of imbalance data (imbalanced learning). We will briefly review the corresponding concepts, how imbalanced data classes could affect the learning process, performance metrics and suggested solutions for similar problems through different domains of studies. The section intends to picture how balanced classification is different from the imbalanced one which is the focus of this study.

2.3 Imbalanced Learning

Exponential growth in generated raw data through various domains (e.g. security, bioinformatics, finance, etc.), has introduced new challenges to the research community for knowledge discovery and data analysis. The existing knowledge discovery and data engineering techniques have shown great success in many real-world applications, but the problem of learning from imbalanced data is a relatively new challenge. The problem is concerned with the performance of learning algorithms in the presence of underrepresented data and severe class distribution skews. Due to the characteristics of imbalanced data sets, learning from such data requires new principles, algorithms, and tools for knowledge discovery and information extraction. Though this section we will briefly go over the correspondent concepts and common approaches related to our study.

2.3.1 Problem Definition

Haibo He [39], defines imbalanced learning as

> the learning process for data representation and information extraction with severe data distribution skews to develop effective decision boundaries to support the decision making process. The learning process could

involve supervised, unsupervised, semi supervised learning or combination two or all of them.

In other words, it is learning from two (binary classification) or multiple classes (multi-class or multi-label classification) of data where the member classes have an unequal amount of examples.

Generally, any dataset with an unequal distribution of examples in between the member classes is technically imbalanced. But when a dataset is labeled as "imbalanced", it means that through that dataset, there is a significant (or in some cases extreme) disproportion in between the number of examples of member classes.

In a binary classification problem with an imbalanced dataset, the class with lower number of instances is called the minority (positive) class and the one with the higher number of examples is called the majority (negative) class. In such a problem, Imbalance Ratio (IR) refers to the degree of existing imbalance in between the two member classes of the dataset [67]. It is defined as the number of negative class examples divided by the number of positive class examples which is 10 for our example in Figure 7. In other words, IR 10 (or 1:10) refers to the fact that for every instance of the minority class, there exist 10 instances in the majority class. Figure 7 shows a sample imbalanced dataset for a binary classification problem with an unequal distribution ratio of 1:10.

In bioinformatics (as we mentioned earlier through the last section) protein research is one of the fields where researchers try to identify the protein structures or its functions [62, 55]. One of the popular approaches for solving these kinds of problems is protein classification. But the protein datasets are mostly imbalanced and therefore specific techniques are required. However, bioinformatics is not the only domain where researchers have to deal with imbalance datasets. Email classification [11], face recognition [94], anomaly detection [46] and medical decision making [60] are among other applications where scientists need to learn and model on imbalanced sets of data.

Most of the imbalanced classification literature has been devoted to binary classification problems. However, there are also multi-class problems where the dataset is imbalanced [62, 84]. The approach for solving these sorts of problems normally includes transforming the multi-class classification problem into multi-binary classification problems. Which is one of the reasons the literature is mostly focused on

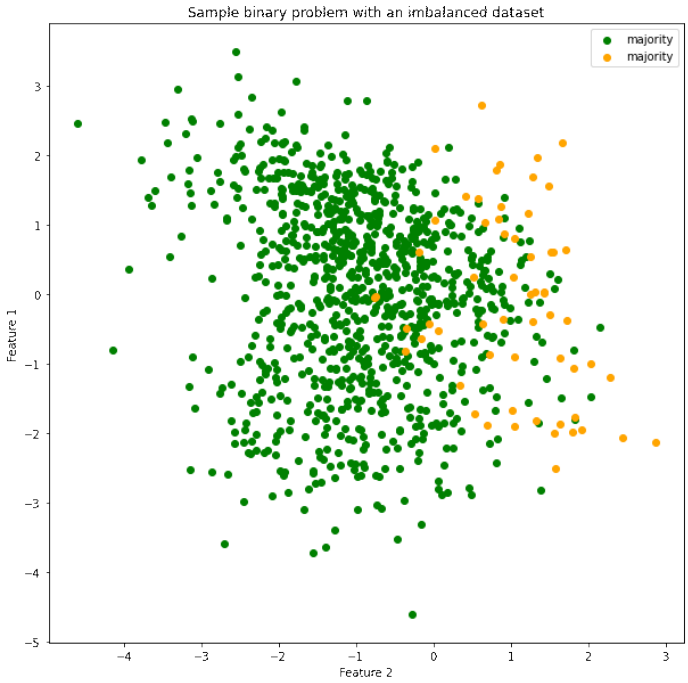

Figure 7: Sample binary imbalanced problem with ratio 1:10

binary classification problems. The multi-class and multi-label classification problem and approaches will be discussed in further details later through this section.

2.3.2 Challenges in Imbalanced learning

The main issue with the imbalanced problems is that normally, the underrepresented class (minority class instances) is the class of interest of the problem from the application point of view [21]. Usually, standard classifier learning algorithms are biased toward the majority class.

In a standard learning algorithm, rules for prediction of the instances are positively weighted in favour of the accuracy metric or the corresponding cost function. In such an algorithm, specific rules for prediction of the examples from the minority class can be ignored (it treats them as noise), because more general rules are preferred. As a consequence, compared to instances from the majority class, minority class instances are more often misclassified. The amount of misclassified instances is even greater for highly imbalanced datasets.

By analyzing 26 binary-class datasets in a study, N. Japkowicz [44] shows how class imbalance impacts minority class classification performance. Figure 8 (being

extracted from the study) shows that the ratio between the minority and the majority class error rates is the greatest when the dataset is highly imbalanced. It also shows that the above error rate decreases as the amount of class imbalance decreases. With an error rate ratio above 1.0, it shows that class imbalance leads to a poorer performance on classifying minority class elements.

So, in similar problems, accuracy is no longer a proper metric for measuring the model performance in an imbalance scenario. The accuracy only takes into account the total number of correctly classified instances. In an imbalance scenario, it often provides a high accuracy value with a very low true positive and a very high true negative value in the confusion matrix. For such a problem, We need to somehow construct classifiers that are biased toward the minority class without being harmful to the accuracy over the majority class.

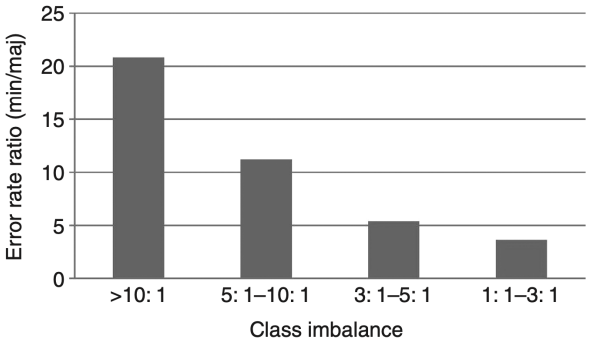

Figure 8: Impact of class imbalance on minority class performance reproduced from [44]

In an imbalanced scenario, the following distributions of the instances of the minority class could also add up to the difficulty of the classification task.

Small sample size Generally imbalanced datasets have a lack of minority class examples. The ratio in between the minority and majority class examples indicates the degree of imbalance in a problem. Datasets with the higher degree of imbalance produce greater error rates.

Overlapping (class separability) When the elements from both minority and majority classes are mixed in the feature space, the decision boundary cannot be clearly established. As a result, more general rules will be applied to the problem in

the classification phase, which will then lead to misclassifying some instances from the minority class [31].

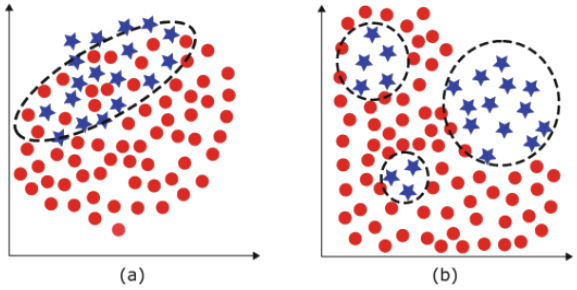

Figure 9: Imbalanced datasets difficulties (a) Class overlapping. (b) Small disjuncts extracted from [28]

Small disjuncts this problem occurs when the represented concept by the minority class is formed of subconcepts [89]. As an example, in case of protein classification, the transporter proteins are classified into 7 substrate specific classes (amino acid transporter, anion transporter, cation transporter, electron transporter, protein/mRNA transporter, sugar transporter and other transporter). In most of the problems, small disjuncts increase the complexity of the problem because the amount of instances among them is not usually balanced.

Imbalanced Classification Approaches

Various techniques have been developed to correctly classify the minority class examples. These techniques can be categorized into four main groups, depending on the way they deal with the problem.

Algorithm level approaches are the ones trying to bias the existing learning algorithms towards the minority class [56]. To achieve this goal, knowledge of both the corresponding classifier and the application domain is required to comprehend the reasons behind the classifier failure when the class distribution is uneven.

Data level approaches are the ones trying to rebalance the class distribution by resampling the data space [7, 27]. This approach does not need to modify the learning

algorithm since the effect caused by the imbalance will decrease after the rebalancing process.

Cost-sensitive learning approach falls between data and algorithm level approaches. In order to achieve the desired classification result on the minority class, It incorporates data level transformations and algorithm modifications [57, 21].

Ensemble-based methods are usually a combination of an ensemble learning algorithm and one of the approaches above [29]. In the data level ensemble learning approach, the data will be preprocessed before training each classifier. On the other hand, the cost-sensitive ensemble learning hybrid guides the cost minimization via the ensemble learning algorithm.

2.3.3 Performance Measurement

The quality of the trained model is generally evaluated by analyzing how well it performs on the test data [2]. To evaluate the model, the provided predictions of the trained classifier are compared to the true classes of test data and some performance measures will be then calculated. Depending on the information being provided by the classifier, we can evaluate the model using either of following approaches:

Nominal class predictions where we compare the predicted class labels with the actual true class values, create a confusion matrix and then calculate the performance measure(s) for evaluating the model.

Scoring Predictions where we use the associated scores (or the probability values) of the predictions to grade test examples according to the likelihood of pertaining to a class and then calculate the required measure for evaluating the model.

For the **nominal class predictions**, a convenient way for summarizing the performance of classifiers is to create a confusion matrix [10]. The columns of the confusion matrix represent the counts of instances in the predicted classes while the rows represent the counts of instances in the actual classes (or vice versa). In this matrix (for a binary class problem), TP and TN (for true positives /true negatives) indicate the correct classification of positive and negative instances, respectively, and FN and FP

(for false negatives /false positives) indicate positive/negative instances misclassified as negative/positive, respectively.

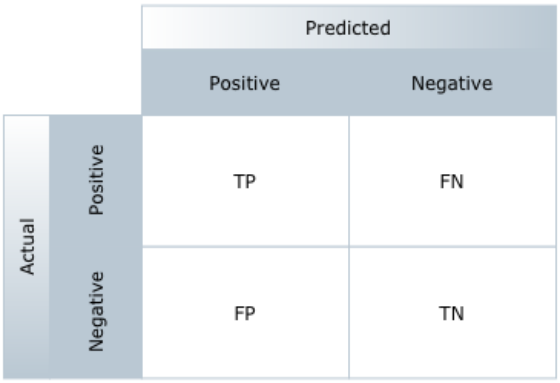

Figure 10: Confusion Matrix

Various performance measures could be calculated using the confusion matrix. These measures correspond to different views of what constitutes a good classifier. Using these different measures we can summarize the confusion matrix into performance metrics so that we can assess the strengths and weaknesses of a classifier from different perspectives.

The first and mostly used measure for evaluating the classification performance is accuracy. Accuracy [1] is the ratio of the correctly classified instances to the total instances of the test set. In the confusion matrix, it is the sum of the true positive and the true negative (which in the binary case is TP + TN) divided by the total number of instances. Error rate [2] is the percentage of incorrectly classified instances.

$$Accuracy = \frac{TP + TN}{N} \tag{1}$$

$$Error = 1 - Accuracy = \frac{FP + FN}{N} \tag{2}$$

The accuracy or error rate is widely used as a performance measure in various problems. But it is not a proper measure in the imbalance scenario [26]. In a highly imbalanced scenario, regardless of number of truly classified instances (tp), it is easy to obtain high accuracy. Figure 11 shows three possible scenarios for an imbalanced problem where all produce the same accuracy value. Accuracy assumes that errors

have an equal cost. But, in an imbalanced classification problem, when compared to the instance of the majority class, misclassifying instances of the minority class is much costlier.

$$M_1 = \begin{pmatrix} 0 & 10 \\ 0 & 990 \end{pmatrix} \qquad M_2 = \begin{pmatrix} 10 & 0 \\ 10 & 980 \end{pmatrix} \qquad M_3 = \begin{pmatrix} 10 & 10 \\ 0 & 980 \end{pmatrix}$$

Figure 11: 3 confusion matrices with the same accuracy

Due to the drawbacks of the accuracy for assessing the performance of the models in an imbalanced scenario, we need some other measures along with the accuracy through which we could obtain more insight on the performance of the model. There are various measures such as Kappa, G-mean, G-measure, Sensitivity, Specificity, MCC, Precision, Recall, F-Measure etc. that could be calculated from the confusion matrix, but the common ones for the problems with an imbalance datasets are as follow.

Sensitivity and Specificity:

The sensitivity of a classifier [3] corresponds to its true positive rate (TPR). It is the proportion of the positive examples being predicted as positive by the model. The complementary metric to the sensitivity is called the specificity of the classifier [4]. It corresponds to the proportion of negative examples that are being predicted correctly. These two metrics are typically used to assess the effectiveness of a clinical test in detecting a disease.

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

$$Specificity = \frac{TN}{TN + FP} \tag{4}$$

MCC

The MCC [59] is a measure that comes from the field of Bioinformatics, where class imbalance occurs very often [5]. It is a measure that takes into account all values of the confusion matrix, considering errors and correct classification in both classes.

MCC ranges from 1 (when the classification is always wrong) to 0 (when it is no better than random) to 1 (when it is always correct).

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

Precision and Recall

The precision of a classifier indicates how precise the model is when identifying the examples of a given class [6]. It assesses whether the proportion of the examples being predicted as positive are truly positive or not. In this pair, recall is the same as the Sensitivity measure being mentioned above [7]. These two measures are commonly used together where scientists are interested in the proportion of the identified relevant information along with the amount of actually relevant information.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

Geometric Mean

Introduced by Kubat et al. [52], The G-mean [8] was a response to the class imbalance problem in an effort to create a single metric by combining a pair. This measure takes into account the relative balance of the classifier's performance on both the positive and the negative classes. By defining a function which takes into account both the sensitivity and the specificity of the classifier.

$$G - Mean = \sqrt{(Sensitivity * Specificity)} \tag{8}$$

F-Measure

The F-measure is a combination metric whose purpose is to combine the values of the precision and recall of a classifier to a single scalar [9]. It does so in a different way than the G-mean, as it allows the user to weigh the contribution of each component

as desired.

$$F\alpha = \frac{(1+\alpha)[Precision * Recall]}{[\alpha * Precision] + Recall} \tag{9}$$

Scoring Predictions

Let's consider a classifier that gives a numeric score or a probability of an instance belonging to a class. Therefore, instead of a simple positive or negative prediction, we will have a score (probability value) for each predicted instance, instances with higher probabilities are more likely to have to be classified as positive.

Having a probability value (or a score) for an instance, we can determine our own threshold to interpret the result of the classifier. Different thresholds will result in different values for the confusion matrix elements (TP, TN, FP, FN) which leads to different values for the calculated measures (e.g sensitivity, specificity, etc.).

A higher threshold will reduce the false positive rate (FPR) and increase the false negative rate (FNR), because less instances will be classified as positive. On the other hand, a lower threshold will increase the FPR and reduce the FNR value. To evaluate these kinds of models, we use the ROC curve.

The ROC curve [26] is a graphical evaluation method that is not dependent on a specific threshold. A ROC graph is a plot of False Positive Rate (FPR) on the x-axis, and True Positive Rate (TPR) on the y-axis. The threshold starts with the one that produces the highest score, all the way to the lowest score. For each possible value of the threshold, there is a point in the ROC space based on the values of FPR and TPR for that threshold.

The AUC (or the Area Under Curve) of the ROC can be interpreted as the probability that the probabilities (or scores) given by a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. The AUC ROC of random guessing is 0.5, so it is expected that the AUC ROC for a useful classifier is higher than 0.5 and the ideal classifier would produce an AUC ROC value of 1.

2.3.4 Dealing with multiple classes

Traditionally, when we talk about imbalance classification, we refer to a binary classification problem with one class having more instances (majority) than the other

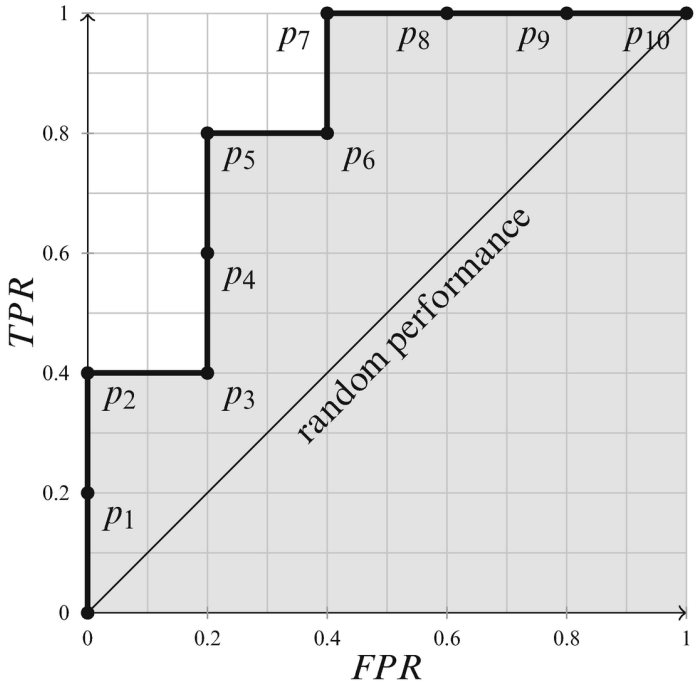

Figure 12: Sample ROC graph extracted from [28]

one (minority) [39, 14]. However, there are many cases in real life that we have to deal with more than two classes. Target detection [79], microarray research [95] and protein classification [96] are among those topics where we face multiple classes of data and the distribution of examples among the classes is not homogeneous.

In such cases, the problem that must be taken into account is the presence of multi-minority and multi-majority classes [88] which somehow implies that we can no longer just focus on a single class to reinforce the learning models towards it. Also, any further complication (e.g. overlapping classes) can affect the problem severely and must be analyzed in depth [80].

To address all the issues, a simple and effective way is to somehow decompose the multi-class imbalance problem into multiple binary-class problems with an imbalanced dataset. We can then assign a classifier to each decomposed problem and the outputs of all the classifiers for a given instance will be aggregated to make the final decision [53]. Therefore, the difficulty in addressing the multi-class problem will be shifted from the classifier itself to the combination stage.

The underlying idea is to undertake the multi-classification using binary classifiers with a divide and conquer strategy. Among decomposition strategies, the most

popular techniques are the One-vs-One (OVO) [40, 48] and One-vs-All (OVA) [24, 3].

The One-vs-All Scheme (OVA)

In OVA decomposition strategy, a problem with n classes of instances is divided into n binary problems. An independent classifier will be then assigned to each binary sub-problem which is responsible for distinguishing one of the classes from all other classes. The learning step of the classifiers is done using the whole training data, considering the patterns from the single class as positives and all other examples as negatives. Figure 13 shows the OVA binarization technique for a 3-class problem.

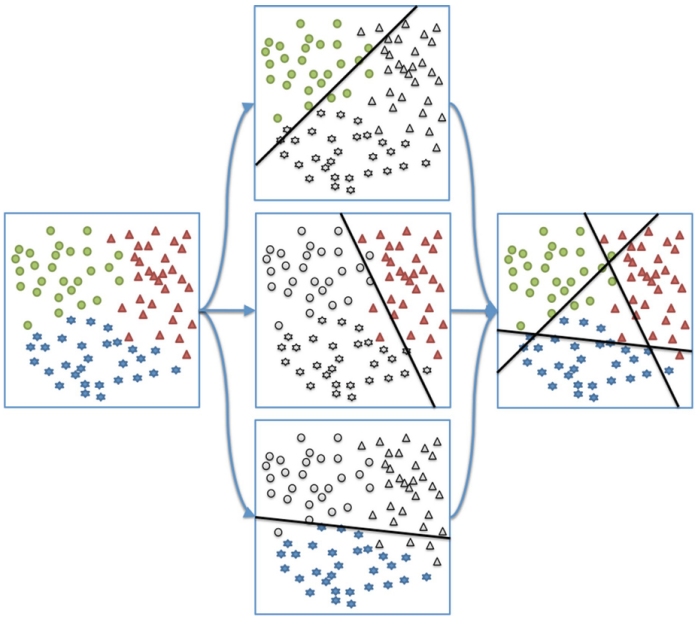

Figure 13: Sample OVA binarization technique for a 3-class problem extracted from [28]

The One-vs-One Scheme (OVO)

In this strategy, a problem with n classes is divided into $n * (n + 1)/2$ binary class problems (one for each possible pair of classes). An independent classifier will be then assigned to each binary sub-problem. For each binary subset, the learning phase is then carried out using a subset of the original training instances with only those that contain any of the two classes and the instances with different class labels are simply ignored. Figure 14 shows the OVO binarization technique for a 3-class problem.

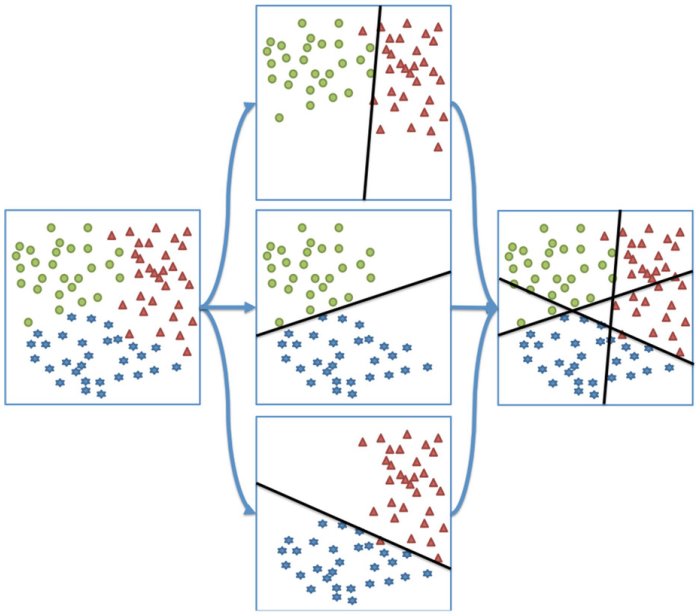

Figure 14: Sample OVO binarization technique for a 3-class problem extracted from [28]

Multi-class Classification vs Multi Label Classification

Multi-class classification refers to a classification task with more than two classes. The classification makes the assumption that one and only one label will be assigned to an instance through the prediction phase. As an example we can consider a fruit image classification problem with 3 classes of Orange, Apple and Tangerine. Using a multi-class classification approach, through the prediction phase, an image can be either an apple or an orange or a tangerine.

Multi-label (multi-output) classification on the other hand, is a generalization of multi-class classification where multiple labels may be assigned to each instance. As an example, we can consider a comment classification problem with 4 classes of Positive, Negative, Toxic, Insulting. Using a multi-label classification approach, an instance can be both Negative and Toxic while another one can be just Negative.

Aggregation

After decomposing the initial multi-class problem into multiple binary class problems, a classifier will be assigned to each problem. As an example, in an OVA case, a problem with 3 classes (labeled as Orange, Tangerine and Apple) will be decomposed into a 3 binary class problem with its own classifier. In order to classify a new

example, the instance will be submitted to all the classifiers. Then, the predictions of all the classifiers are considered in the combination phase, which is also known as classifier fusion or aggregation [93].

In case of Multi-class classification problems, since we are looking for one and only one prediction for each instance, in the aggregation phase, the result with the highest score/probability will be chosen as the final prediction for that instance. In the example mentioned above, if classifier 1 predicts Orange with 0.75, classifier 2 predicts not a Tangerine with 0.64 and classifier 3 predicts Apple with 0.82, then the final predicted label will be an Apple. The predicted results are then put together and compared to the actual labels to generate the confusion matrix which is then used to calculate the performance metrics.

In case of Multi-label classification problems, since all the labels are acceptable, then all will be considered. As an example, in a comment classification problem, a comment can be both Negative and an Insult.

Threshold-Moving

Many machine learning algorithms are capable of producing a probability or a score value for an instance of a dataset. This value needs to be interpreted before being mapped to a class label. The decision for converting a predicted probability or scoring into a class label is governed by a parameter referred to as the "decision threshold," "discrimination threshold," or simply the "threshold." The default value for the threshold is 0.5 for normalized predicted probabilities or scores in the range between 0 or 1.

When studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake" [77]. So, for those classification problems with an imbalance dataset, the default threshold can result in poor performance. A simple and straightforward approach for improving the performance of a classifier that predicts and produces probability values for instances of an imbalanced classification problem, is tuning the default threshold being used by the algorithm for mapping the probability values to class labels.

In a problem, if a specific threshold is considered, the threshold should be applied to each classifier before the aggregation phase. For example in the case of the classifier

for Oranges, Tangerines and Apples, if the considered threshold is 0.71 (which means the scores/probabilities above that threshold is considered as positive) then the results will be Oranges (classifier 1), not a Tangerine (classifier 2) and not an Apple (Classifier 3). Which then leads to the final prediction of the Orange as the label for the instance after aggregation. The threshold-moving effect could be observed and analyzed through the ROC/AUC or Precision/Recall curves.

Micro averaging vs Macro averaging

Considering a performance metric P being calculated based on the confusion matrix elements (true positives tp, true negatives tn, false positives fp, false negatives fn). The macro and micro averages of a specific measure can be calculated as follow:

$$P_{macro} = \frac{1}{q}\sum_{\lambda=1}^{q} P(tp_\lambda, tn_\lambda, fp_\lambda, fn_\lambda) \tag{10}$$

$$P_{micro} = P(\sum_{\lambda=1}^{q} tp_\lambda, \sum_{\lambda=1}^{q} tn_\lambda, \sum_{\lambda=1}^{q} fp_\lambda, \sum_{\lambda=1}^{q} tp_\lambda) \tag{11}$$

Where λ is a Label and $L = \lambda_j : j = 1...q$ is the set of all labels.

For a problem in machine learning, normally, we take different samples from a dataset and then we run the model on all of those samples independently to estimate the performance of a machine learning model on the unseen data. A popular technique in this area is k-fold cross validation [43]. In order to produce the final metrics for a model, we need to average between a performance metric (e.g. sensitivity) of different classifiers in a model or multiple models on different samples of data (which is the case in multi-class or multi-label classification problems). The averaging could be done by either macro or micro averaging approach which can produce different results.

Micro and macro-averages compute slightly different things. So, their generated result's interpretation is different from one another. Macro-average approach, computes the metric independently for each class and then takes the average (hence treating all classes equally), whereas the micro-average approach, aggregates the contributions of all classes to compute the average metric [85]. For imbalanced problems involving multiple classes, the micro-averaging approach is preferable.

Support Vector Machines (SVM)

Algorithm-level solutions (the ones trying to bias the existing learning algorithms towards the minority class [56]) concentrate on modifying existing learners methods for handling imbalanced datasets. Instead of focusing on modifying the training set in order to combat class skew, this approach aims at modifying the classifier learning procedure itself to alleviate their bias towards majority class instead on altering the supplied training set [51].

Due to their powerful generalization abilities, convergent properties and flexibility in adapting to various learning difficulties, Support Vector Machines (SVMs)[87] are among the most popular algorithms for pattern classification in problems with imbalanced datasets. The algorithm is effective in high dimensional spaces and it accepts different Kernel functions for the decision making process.

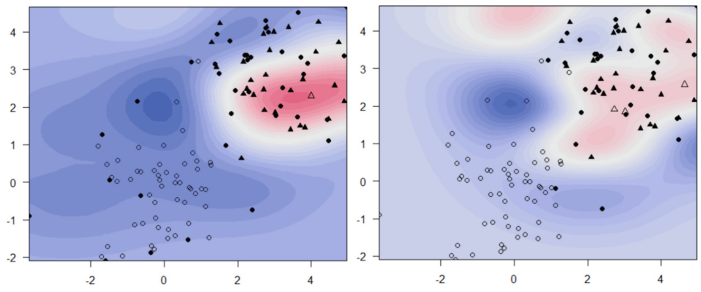

Figure 15: Support Vector Machine boundries for an imbalanced dataset: (left) standard approach; (right) instance-level weighting extracted from [28]

Fernandez et al [28] explains the support vector machine, involved parameters and the algorithm's related imbalanced classification concerns as:

> SVM algorithm aims at finding the optimal hyperplane which separates instances into two classes. Traditional linear classifiers offer many desirable properties, but they are not able to cope with complex data structures. SVM transforms the input instances into a higher dimensional artificial feature space(s). So, by using a non-linear mapping θ, it can achieve a linear separation between classes in the new space, which in turn translates to a non-linear decision boundary in the original feature space. The potential separating hyperplane can be represented as:

$$w.\theta(x) + b = 0 \tag{12}$$

Where w stands for a weight vector normal to this hyperplane. In case of considered data are linearly separable, the decision hyperplane characterized by a maximum margin can be obtained by optimization of margin as: $min(1/2w.w)$ subject to $\forall_{i=1...l}, y_i(w.\theta(x_i)+b) \geq 1$ where l stands for the number of training instances.

However, datasets are rarely linearly separable. So, we need to modify the the equation to include the possibilities of classifying some of training instances, to achieve greater generalization and reduce overfitting. This is done by using slack variable associated with $i-th$ instance $\epsilon_i \geq 0$. This allows to rewrite the margin optimization problem as soft margin:

$$min(\frac{1}{2}w.w + C\sum_{i=1}^{l}\epsilon_i) \tag{13}$$

Subject to $\forall_{i=1...l}, \forall_{\epsilon \geq 0}, y_i(w.\theta(x_i)+b) \geq 1-\epsilon_i$ Where C stands for the regularization parameter that controls the trade-off between maximizing the separation margin between classes and minimizing the number of misclassified instances. This is a quadratic optimization problem that can be solved by transforming it into Lagrangian optimization problem with the following dual form:

$$max_{\alpha_i}\left(\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \theta(x_i).\theta(x_j)\right) \tag{14}$$

Subject to $\forall_{i=1...l}, \forall_{0\leq\epsilon_i\leq C}, \sum_{i=1}^{l} y_i\alpha_i = 0$. As learning the mapping function $\theta(x)$ may be difficult or even impossible, SVMs use kernel functions $K(x_i, x_j) = \theta(x_i).\theta(x_j)$. So, would be able to write the dual optimization problem in its kernelized form as:

$$max_{\alpha_i}\left(\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j)\right) \tag{15}$$

Subject to $\forall_{i=1...l}, \forall_{0\leq\epsilon_i\leq C}, \sum_{i=1}^{l} y_i\alpha_i = 0$. Solving this kernelized dual optimization form and finding optimal values of α_i allows us to calculate

$w = \sum i = l\alpha_i y_i \theta(x_i)$ and determine value of parameter b from Karush-Kuhn-Tucker conditions. Training instances with associated non-zero values of α_i are known as support vectors and deemed as sufficient to represent the training set. Therefore, SVMs achieve instance reduction by relying only on support vectors. This will lead to the following equation:

$$f(x) = sign(w.\theta(x) + b) = sign\left(\sum_{i=1}^{l} \alpha_i yi K(x_i, x) + b\right) \qquad (16)$$

In its original form, SVMs are prone to imbalanced class distributions. The first problem is about the soft margin optimization task (formula 13).The regularization parameter C here is the misclassification cost for penalizing errors on the training set. But, it assumes that the cost assigned to both of classes are identical. Therefore, the learning algorithm will favor the majority class over the minority class.

Another potential drawback of SVMs is related to the support vectors derived from imbalanced data. Only instances with $\alpha_i \geq 0$ will be preserved and used as support vectors. As this process is also skew-insensitive, the larger imbalance ratio will lead to bigger disproportions in the number of support vectors associated with each class.

The following algorithm-level solutions has been proposed to overcome these shortcomings:

- Kernel Modifications
- Kernel Boundary and Margin Shift
- Kernel Target Alignment
- Kernel Scaling
- Weighted Approaches
- Instance Weighting
- Support Vector Weighting
- Fuzzy Approaches

SVM Implementation Libraries (SVMLight, Scikit-Learn)

Scikit-learn [70] is one of the main libraries for machine learning in python. It provides a wide range of functionalities for machine learning problems. As a part of it's library, it provides an implementation of the Support Vector Machine algorithm with more options. The support vector machines in scikit-learn support both dense ($numpy.ndarray$ and convertible to that by $numpy.asarray$) and sparse (any $scipy.sparse$) sample vectors as input. However, to use an SVM to make predictions for sparse data, it must have been fit on such data. For optimal performance, use C-ordered $numpy.ndarray$ (dense) or $scipy.sparse.csr_matrix$ (sparse) with $dtype = float64$.

SVMLight, is another implementation of Vapnik's Support Vector Machine [87] in C. According to the description being provided by it's creator Thorsten Joachims [45], It can be used in pattern recognition, regression and learning a ranking function. The software also provides methods for assessing the generalization performance.

Through this chapter, we explained how learning from the imbalanced sets of data is different from the balanced ones when dealing with multiple classes (which is a common case for protein classification). The issue was addressed by describing the reproducibility-related terms, reproducibility-related studies in machine learning, bioinformatics context and its subdomains, the characteristics of the protein classification problems, imbalanced learning approach in machine learning and the involved phases and parameters for calculation of the final results.

Compared to the machine learning problems with balanced sets of data, imbalanced learning involves more parameters. The model also needs to go through extra phases to predict the labels and eventually calculate the final performance metrics. Failing to report on these parameters creates methodological flexibility in the replication process which could then produce a wide range of results.

Chapter 3

Protein Classification Process

Through this chapter and the next one, we demonstrate the impact of methodological flexibility (the flexibility associated with implementing the original study experiment using the same data, analysis tools and through the same environment to obtain the same result) on the classification performance in an imbalanced scenario.

The materials and methods described in this section present a replication of the study performed by [62]. In cases where insufficient details were provided for replication, these decision points were noted and several sensible options selected and compared. All software developed to curate the dataset and perform the experiment, including a Jupyter notebook to reproduce our key results, can be found publicly available on our GitHub repository: https://github.com/big-data-lab-team/reproducibility-bioinfo/ .

3.1 Dataset

The SwissProt UniProt database with rich sequence and substrate annotations [12] was subsampled to include 900 membrane transporter proteins and 660 non-transporter proteins. For the training set, 780 transport proteins were divided into 7 substrate-specific classes (70 amino acid transporters, 60 anion transporters, 260 cation transporters, 60 electron transporters, 60 sugar transporters, 70 protein/mRNA transporters, and 200 other transporters). With the addition of 600 non-transporter proteins, the total dataset contained $1,380$ protein sequences.

The independent set contains 60 non-transporter proteins and the remaining 120

transporter proteins being divided into the same 7 substrate-specific classes (15 amino acid transporters, 12 anion transporters, 36 cation transporters, 10 electron transporters, 12 sugar transporters, 15 protein/mRNA transporters, and 20 other transporters).

Features were computed for each protein, including: Amino Acid Composition (AAC), Dipeptide Composition (DPC), Physico-Chemical Composition (PHC), Biochemical Composition (AAindex) and Position-specific scoring matrix (PSSM) profile. Each feature was computed identically to the methods described by [62] and are briefly summarized below:

- **Amino Acid Composition (AAC)**: a feature vector of 20 values ranging from 0 – 100 indicating the percentage of all standard amino acids present within a protein, as defined by [33]. Also known as Monopeptide Composition (MPC).
- **Dipeptide Composition (DPC)**: a feature vector of 400 values ranging from 0 – 100 indicating the percentage of all possible ordered amino acid pairs present within a protein, as defined by [33].
- **Physico-Chemical Composition (PHC)**: a feature vector of 11 values corresponding to percentage composition of physico-chemical residue classes, including: Aliphatic, Neutral, Aromatic, Hydrophobic, Charged, Positively charged, Negatively charged, Polar, Small, Large, and Tiny.
- **Biochemical Composition (AAindex)**: a feature vector of 49 physical, chemical, energetic, and conformational amino acid properties which have been averaged across all amino acids present within the protein.
- **Position-specific scoring matrix (PSSM) profile**: a feature vector of 400 sequence likelihoods aggregated across min-max scaled probabilities for each ordered amino acid pair within proteins in the SwissProt database.

While the computed AAC, DPC, and PHC features were verified against those previously generated in literature [62], the web reference on the initial study for AAindex and PSSM profiles were not available. So, to validate their similarity they were checked with Munira Alballa, a PHD student of Dr. Gregory Butler who was working on related subject [1] and was experienced with the matter.

3.2 Model Flexibility

Though the majority of dataset and model specifications were clearly specified by [62], there remains flexibility along various axes in the analysis, namely:

1. the number of involved classes in the classification task,
2. the sorting and balancing of samples within the dataset,
3. the selected SVM hyperparameters, gamma and cost,
4. the uniformity (or possible lack of) SVM hyperparameters across binary classifiers,
5. the technique applied to aggregation and evaluation of binary classifiers, and
6. the prediction method for the final labels.

Considering the available degrees of freedom and limited computational resources, the AAC feature was used initially to train and evaluate model parameters. The best performing model using AAC was then re-trained using the full feature set. In the following section, the experimental design is described in detail with reference to the axes of flexibility, above. Diagram 16 provides a graphic representation of the parameters explored in this section alongside the process through which the study was conducted.

3.3 Experimental Design

Despite the multi-class nature of this task, the models developed and evaluated below were constructed in a binary classification scenario. This was accomplished using the "one versus rest" strategy which was performed either prior to training or automatically depending on the classifier. Support Vector Machine (SVM) Classifiers were initially built using the SVMLight library, originally used by [62], which reported a probability of class membership in each binary setting which were then combined as a multi-class confusion matrix. These models were replicated using SciKit-learn (Scikit-Learn), a popular library for machine learning in Python, in both an identical

setting to SVMLight (termed: Scikit-Learn Probability) and an approach which automatically performs the class reconstruction and prediction described above (termed: Scikit-Learn Prediction).

3.3.1 Training

All models were fit for both 7 and 8 class scenarios (Addr: 1) – excluding and including non-transport proteins, respectively. The models were fit using 3 distinct training paradigms: i) balanced, ii) shuffled, and iii) downsampled (Addr: 2). In the balanced case, training- and testing-sets were created for each model through 5-fold cross validation (CV) that were randomly generated and stratified to balance class membership across folds. The shuffled case was performed similarly to the balanced case without stratification guaranteeing balanced class membership across folds. The downsampled case was performed in accordance to the balanced case following a reduction in samples to 60 observations per class. This resulted in 6 distinct training methods.

3.3.2 Model Hyperparameters

For all models, the Radial Basis Function (RBF) kernel was used and the gamma and cost parameters for the model ranged from $1e^{-5}$ – 10 (gamma) and 1 – 4 (cost), respectively, consistent with those presented by [62]. Specific values were determined through a grid search (Addr: 3). In the case of SVMLight and Scikit-Learn Probability scenarios, gamma and cost values were either uniform or varied across classes (Addr: 4), whereas the implementation of the Scikit-Learn Prediction model permitted only uniform pairs across all classes.

3.3.3 Performance Evaluation

Model performance was evaluated through standard measures of sensitivity, specificity, accuracy, true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), and Matthew's Correlation Coefficient (MCC) which is a measure of correlation suitable for imbalanced classification problems [13]. As micro- and macro-averaging approaches – evaluating binary classifiers before or after aggregation into a multi-class model, respectively – lead to different results in an imbalanced

classification setting, both approaches were used for scoring here (Addr: 5).

In the case of both the SVMLight and Scikit-Learn Probability models, the resulting classification and performance for each model was determined by the aggregation of independent binary classifiers according to three distinct methods: maximum probability, unweighted average, and balanced average (Addr: 6). The maximum probability method assigns each sample a label corresponding to the binary classifier with the highest certainty, resulting in a non-overlapping classification result which was evaluated. In the case of both averaging methods, each probability is converted into a binary classification through thresholding and is scored independently prior to aggregating the performance of all classifiers. The unweighted average thresholds probabilities at the median value, whereas the balanced case uses a threshold proportional to the true number of members belonging to a given class. As the Scikit-Learn Prediction model returned pre-determined group confusion matrix and class memberships, performance metrics were computed upon these directly.

3.4 Model Comparison

Models were compared to the reported reference classifier through the Euclidean distance between a 4-dimensional feature vector containing Sensitivity, Specificity, Accuracy, and MCC. The closest models were chosen as those which minimized this distance. Model settings, such as those defined to address points 1 – 6 above, were compared quantitatively through the pairwise application of two-sided Mann–Whitney U Tests on the distribution of distance values for all unique settings within a given category (i.e. the distribution of distances for all models using micro-averaging was compared to the equivalent distribution for all models using macro-averaging). To avoid overfitting, the closest 10% of models to the reference were selected as the models used for further investigation.

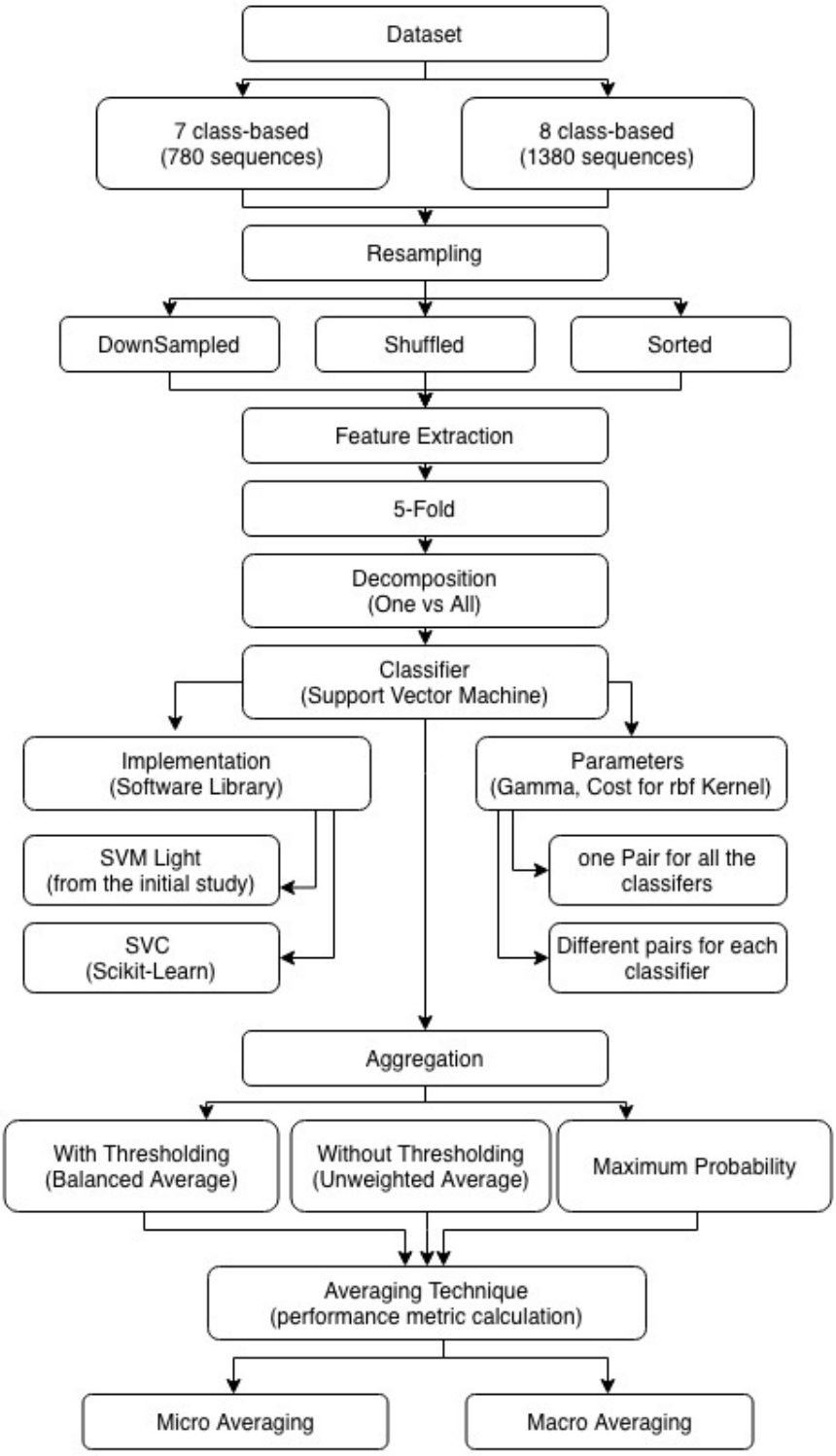

Figure 16: The study process

Chapter 4

Results

In this work, considering the available flexibility along the various axes in the analysis (Section 3.2), multiple models were built on the AAC (amino acid composition) feature set (Tables 1, 2, 3). For each model, the distance between the generated results and the reference values from the original study was then calculated (Figure 17). We then picked the settings from the 16 models showing the least distance from the reference values. The model settings from those 16 models were then used to build models for the 18 remaining feature sets (Tables 4, 5, 6, 7, 8, 9). The results from all the 19 feature sets were then compared and illustrated in Figure 18. Through this section, the context is also organized accordingly.

4.1 AAC Models

Tables 1 contains the results of the probability-based models with 7-classes of proteins in the dataset. All the models were evaluated using the performance metrics in [62]. Among all the models, the 8 highlighted ones had a distance value between 0.07 and 0.13 from the reference. These are the ones that produced the closest results to the initial ones.

Tables 2 shows the performance of the probability-based models with 8-classes of proteins in the dataset. Among all the models, the closest ones (8 highlighted models) reported distance values between 0.08 and 0.10 from the reference values.

Table 3 shows the results of the scikit-learn prediction-based models for the models with 7- and 8-classes of proteins in the dataset. This function from the scikit-learn

library, by default, aggregates the results using the maximum probability technique. Using this approach, all the models reported distance values above 0.32 from the reference values.

4.2 Closest Models

Figure 17 shows the sensitivity and specificity of each tested model alongside the performance of the originally published model (on AAC feature set), with 10% of models most closely matching performance to the reference highlighted.

The closest 10% of models (16) used a variety of configurations, and each reported a distance score of less than 0.13 from the reference. The breakdown of configurations for these models included: micro aggregation (all), balanced average prediction method (all), balanced (8) or shuffled (8) dataset, contained 7 (8) or 8 (8) classes in the dataset, were trained with uniform (8) or heterogeneous (8) hyperparameters, and were developed using SVMLight (8) or the Scikit-Learn Probability (8) model architectures. While the Scikit-Learn Prediction model and downsampled dataset configuration are notably absent from these models, all other settings were either dominated by a single value, such as in the case of micro aggregation and the balanced average prediction method, or the settings were equally represented. This uniformity in representation is consistent with the direct comparisons between settings described above.

4.3 Model Differences

This section will explore the differences in the model performances based on the defined axes of flexibility enumerated in Section 3.3.

Number of Classes While the 7-class models appear to be slightly closer to the reference, there was no significant difference between the number of classes and the distance from reference ($p > 0.1$). Models trained with 8 classes tended to achieve higher sensitivity and specificity values. It seems that the addition of the background class improved the performance.

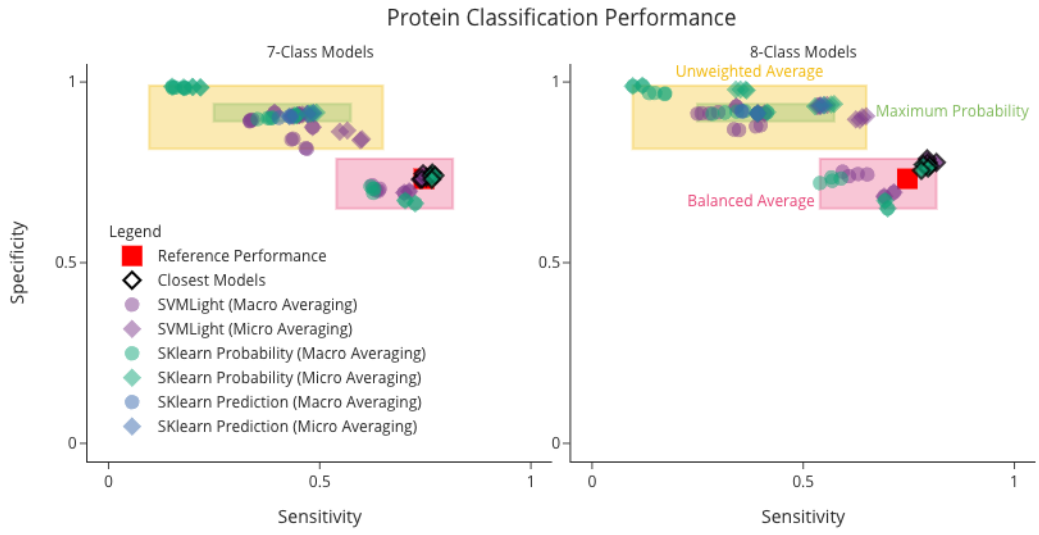

Figure 17: Sensitivity and Specificity of each tested model. Each panel contains models trained with a fixed number of categories (7: left; 8: right), and shows the published reference performance in red. The closest 10% of models to this reference have been outlined in black. The symbol colour and shape refer to the classifier type and aggregation strategy, respectively. Each shaded region illustrates the bounds of performance for a given binary classifier aggregation strategy.

Dataset Sampling The dataset composition had no significant impact on the closeness of the model to the reference ($p > 0.1$ for all comparisons). However, none of the closest 10% of models were trained using the downsampled dataset.

SVM Hyperparameters All uniformly parameterized models converged to same set of hyperparameters within the number of classes. For the models with 7-classes of proteins, the ones with the closest performance used Gamma and Cost values of 0.02 and 4.5 while for the models with 8-classes of proteins, the closest results were achieved using the values of 0.01 and 4 for Gamma and Cost respectively. There was no significant difference between these sets of parameters.

Hyperparamter Heterogeneity Similarly to the case of uniform parameters, models converged on Gamma values between 0.02 and 0.04 for all classes and models, and Cost values between 4 and 5, with no statistically significant difference between models or classes.

Aggregation Technique Models using the micro performance-aggregation technique (i.e. evaluating individual binary classifiers prior to aggregation into a multi-class classifier) obtained closer results to the reference than those using the macro technique ($p < 1 \times 10^{-4}$). All of the closest models used micro-aggregation.

Prediction Method The balanced averaging prediction method produced significantly closer results to the reference than both the unweighted average and maximum probability methods ($p < 1 \times 10^{-5}$ for both). The maximum probability method also produced significantly closer results than the unweighted average method ($p < 0.001$).

Tool The SVMLight classifiers produced closer results to the reference than both Scikit-Learn Probability and Prediction models ($p < 0.05$ for both). While the Scikit-Learn Prediction model architecture did not appear in the set of closest models, there was no statistically significant difference between its performance and that of the Scikit-Learn Probability models.

4.4 All Feature Sets Results

According to [62], among all the 19 feature sets, the hybrid dataset that includes the biochemical composition (AAindex) and the PSSM profile, provides the best results for the membrane protein classification with the highest MCC value.

Table 4 shows the results from running the closest-performing models (from the AAC experiment) on the main and the independent datasets for the AAindex+PSSM profile dataset. All 16 models reported distance values between 0.07 and 0.09 from the reference values.

Tables 5, 6, 7, 8, 9 contain the results from running the 10% closest-performing models (16) on all the other 18 features.

Table 5 shows the results from running those models on DPC, PHC, AAindex and PSSM feature sets. They all reported the distance values between 0.06 and 0.13 from the reference values. For the DPC feature set, the models seem to perform slightly better in the 7-class-based settings. The rest of the models (for the PHC, AAindex and PSSM feature sets) reported quite close performance values in both 7- and 8-class-based settings.

Tables 6, 7 show the results from running the closest models' settings on the hybrid feature sets being produced by combining 2 different features (8 features). All the models reported distance values between 0.06 and 0.13 from the reference values. The models for DPC+AAC, DPC+PSSM and DPC+AAINDEX feature sets seem to perform slightly better in 7-class-based settings while the results from all the other 5 feature sets show a close performance in both 7- and 8-class-based settings.

Tables 8, 9 show the results from running the closest models' setting on the hybrid datasets being produced by combining 3 feature sets(6 features). All the models reported distance values between 0.06 and 0.11 from the reference values. The hybrid AAC+DPC+AAINDEX model seems to perform slightly better in 7-class-based settings while all the models show a close performance in both 7- and 8-class-based settings.

Figure 18 compares the MCC values resulting from running the closest-performing models on all the feature sets (19 features). The hybrid dataset that includes the biochemical composition (AAindex) and the PSSM profile, outperforms others. Compared to all the other models, these models produce the highest MCC values.

7-class based model for AAC				
Original Results	Accuracy	Sensitivity	Specificity	MCC
	73.74	74.65	73.22	0.46

	Dist	Gamma-Cost different for each class						same Gamma-Cost for all classes				Dist
	ance	acc	sens	spec	mcc			acc	sens	spec	mcc	ance
SVM Light												
unweighted average	0.25	82.28	56.53	86.58	0.37	b	b	80.73	60.00	84.18	0.37	0.21
	0.43	84.07	39.28	91.55	0.32	d	Micro d	81.90	48.33	87.5	0.32	0.33
	0.26	81.74	54.74	86.24	0.36	s	s	80.42	59.61	83.88	0.36	0.21
	0.37	82.28	43.76	84.31	0.31	b	b	80.73	46.64	81.69	0.30	0.33
	0.43	84.08	39.28	91.54	0.32	d	Macro d	81.90	48.33	87.49	0.33	0.33
	0.38	81.73	43.27	84.06	0.28	s	s	80.42	46.96	81.49	0.29	0.33
balanced average	0.09	73.84	75.25	73.60	0.36	b	b	74.10	75.12	73.93	0.36	0.09
	0.18	69.45	70.00	69.36	0.28	d	Micro d	69.96	71.19	69.76	0.29	0.16
	0.11	72.06	76.15	71.39	0.34	s	s	72.91	75.38	72.50	0.35	0.13
	0.20	73.84	64.18	70.45	0.28	b	b	74.10	63.42	70.70	0.28	0.21
	0.18	69.45	70.00	69.36	0.28	d	Macro d	69.96	71.19	69.76	0.30	0.16
	0.20	72.06	66.89	68.16	0.27	s	s	72.91	64.46	69.24	0.26	0.21
maximum probability	0.35	84.87	47.05	91.17	0.38	b	b	84.54	45.89	90.98	0.36	0.36
	0.37	84.35	45.24	90.87	0.36	d	Micro d	83.87	43.57	90.59	0.34	0.38
	0.37	84.32	45.12	90.85	0.35	s	s	84.21	44.74	90.79	0.35	0.37
	0.48	84.87	34.05	89.58	0.28	b	b	84.54	33.59	89.48	0.27	0.49
	0.37	84.35	45.23	90.87	0.34	d	Macro d	83.87	43.57	90.59	0.33	0.39
	0.48	84.32	33.52	89.14	0.27	s	s	84.21	33.47	89.15	0.27	0.48
Scikit Learn												
unweighted average	0.61	87.51	21.47	98.44	0.33	b	b	87.45	19.39	98.69	0.32	0.63
	0.65	87.03	17.43	98.60	0.29	d	Micro d	86.68	15.28	98.56	0.27	0.68
	0.61	87.49	21.66	98.41	0.34	s	s	87.55	20.878	98.63	0.34	0.62
	0.66	87.49	18.29	98.20	0.24	b	b	87.44	14.87	98.46	0.20	0.71
	0.67	87.03	17.46	98.61	0.24	d	Macro d	86.68	15.32	98.56	0.22	0.70
	0.67	87.47	17.52	98.18	0.23	s	s	87.53	16.32	98.40	0.21	0.69
balanced average	0.07	75.33	76.42	75.13	0.38	b	b	75.33	76.41	75.15	0.38	0.07
	0.20	68.09	70.95	67.62	0.27	d	Micro d	67.31	70.47	66.78	0.26	0.21
	0.07	75.32	76.41	75.15	0.38	s	s	73.46	75.25	73.16	0.35	0.10
	0.20	75.32	61.79	71.51	0.29	b	b	75.32	61.79	71.51	0.30	0.20
	0.18	68.09	70.95	67.61	0.29	d	Macro d	67.31	70.47	66.78	0.27	0.20
	0.20	75.32	61.79	71.22	0.29	s	s	73.46	61.48	69.51	0.25	0.24
maximum probability	0.34	85.38	48.84	91.47	0.40	b	b	85.20	48.20	91.36	0.39	0.34
	0.37	84.21	44.76	90.79	0.35	d	Micro d	84.14	44.52	90.75	0.35	0.37
	0.34	85.12	47.95	91.32	0.39	s	s	85.12	47.94	91.32	0.39	0.34
	0.43	85.38	38.43	89.98	0.32	b	b	85.20	37.61	89.90	0.31	0.44
	0.38	84.21	44.76	90.79	0.32	d	Macro d	84.14	44.52	90.75	0.33	0.38
	0.44	85.12	38.33	89.94	0.31	s	s	85.12	36.43	89.85	0.30	0.46

B, D and S are balanced, down-sampled and shuffled instances of the main dataset.
Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient

Table 1: The average sensitivity, specificity, accuracy, and MCC for 7 class-based models.

8-class based model for AAC				
Original Results	Accuracy	Sensitivity	Specificity	MCC
	73.74	74.65	73.22	0.46

	Dist	Gamma-Cost different for each class						same Gamma-Cost for all classes				Dist
	ance	acc	sens	spec	mcc			acc	sens	spec	mcc	ance
SVM Light												
unweighted average	0.24	87.33	64.92	90.53	0.49	b	b	86.22	63.40	89.48	0.46	0.23
	0.49	85.86	34.16	93.24	0.30	d	Micro d	85.02	40.83	91.34	0.32	0.42
	0.24	87.02	63.98	90.31	0.48	s	s	86.24	62.46	89.63	0.46	0.23
	0.43	87.33	39.91	88.02	0.30	b	b	86.22	34.78	86.71	0.25	0.48
	0.49	85.85	34.16	93.24	0.28	d	Macro d	85.02	40.83	91.34	0.30	0.42
	0.44	87.02	38.81	87.71	0.28	s	s	86.24	33.57	86.89	0.23	0.50
balanced average	0.10	79.98	79.78	80.01	0.44	b	b	79.74	77.97	80.00	0.43	0.10
	0.19	68.93	71.87	68.51	0.27	d	Micro d	68.07	69.79	67.82	0.25	0.22
	0.09	77.71	79.93	77.40	0.41	s	s	78.31	78.33	78.31	0.41	0.09
	0.21	79.59	57.34	76.24	0.28	b	b	79.59	57.34	76.24	0.28	0.25
	0.19	68.93	71.87	68.51	0.27	d	Macro d	68.07	69.79	67.82	0.25	0.22
	0.21	77.71	53.55	74.14	0.27	s	s	78.31	58.42	74.72	0.26.	0.26
maximum probability	0.31	88.91	55.65	93.66	0.49	b	b	88.51	54.05	93.43	0.47	0.32
	0.43	84.84	39.37	91.33	0.30	d	Micro d	84.79	39.16	91.30	0.30	0.44
	0.32	88.44	53.76	93.39	0.47	s	s	88.33	53.33	93.33	0.46	0.32
	0.54	88.91	29.87	91.65	0.26	b	b	88.51	25.93	91.35	0.22	0.58
	0.44	84.84	39.37	91.34	0.29	d	Macro d	84.79	39.16	91.31	0.28	0.44
	0.57	88.44	27.85	91.28	0.23	s	s	88.33	24.93	91.22	0.20	0.60
Scikit Learn												
unweighted average	0.48	90.22	36.33	97.84	0.46	b	b	90.11	35.22	97.92	0.45	0.49
	0.72	88.10	12.18	98.86	0.23	d	Micro d	87.75	9.63	98.86	0.19	0.76
	0.48	90.10	36.52	97.68	0.45	s	s	89.89	33.28	97.96	0.43	0.50
	0.68	90.19	17.30	96.85	0.22	b	b	90.10	15.26	96.98	0.17	0.71
	0.74	88.10	12.21	98.86	0.18	d	Macro d	87.75	9.71	8.86	0.13	0.78
	0.68	90.07	17.17	96.68	0.20	s	s	89.88	12.90	97.02	0.15	0.74
balanced average	0.08	77.08	80.0	76.66	0.40	b	b	77.70	78.18	77.63	0.40	0.08
	0.22	67.91	69.16	67.73	0.25	d	Micro d	68.75	67.28	68.96	0.25	0.23
	0.08	76.66	78.47	76.40	0.39	s	s	74.36	79.41	73.64	0.37	0.09
	0.24	77.08	59.13	73.10	0.27	b	b	77.70	54.72	74.14	0.25	0.28
	0.21	67.91	69.16	67.73	0.26	d	Macro d	68.75	67.29	68.95	0.26	0.22
	0.27	76.66	56.55	72.78	0.25	s	s	74.36	56.27	0.07	0.22	0.30
maximum probability	0.31	89.33	57.31	93.90	0.51	b	b	88.73	54.92	93.56	0.48	0.32
	0.42	85.10	40.41	91.49	0.31	d	Micro d	85.31	41.25	91.60	0.32	0.41
	0.31	89.09	56.37	93.76	0.50	s	s	88.15	52.60	93.22	0.45	0.33
	0.47	89.32	36.12	91.94	0.34	b	b	88.73	31.39	91.51	0.29	0.52
	0.43	85.10	40.41	91.48	0.30	d	Macro d	85.31	41.25	91.60	0.31	0.42
	0.49	89.09	33.78	91.73	0.30	s	s	88.15	28.05	91.14	0.23	0.56

B, D and S are balanced, down-sampled and shuffled instances of the main dataset.
Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient

Table 2: The average sensitivity, specificity, accuracy, and MCC for 8 class-based models.

Scikit-learn prediction-based models													
Original Results	Accuracy		Sensitivity		Specificity			MCC					
	73.74		74.65		73.22			0.46					
	Dist	7 class-based models							8 class-based models				Dist
	ance	acc	sens	spec	mcc				acc	sens	spec	mcc	ance
Prediction-based	0.70	76.88	19.10	86.51	0.05	s		s	84.58	38.33	91.19	0.29	0.45
	0.39	83.66	42.86	90.47	0.33	d	Micro	d	84.79	39.16	91.31	0.30	0.44
	0.36	84.43	45.51	90.91	0.36	sh		sh	88.71	54.85	93.54	0.48	0.32
	0.81	76.88	6.69	81.70	0.01	s		s	84.58	76.82	86.79	0.03	0.81
	0.39	83.67	42.85	90.47	0.32	d	Macro	d	84.79	39.16	91.31	0.29	0.44
	0.39	85.27	43.31	90.33	0.35	sh		sh	88.71	35.32	91.94	0.29	0.48
S, D and SH are sorted, down-sampled and shuffled instances of the main dataset. Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient													

Table 3: The average sensitivity, specificity, accuracy, and MCC values for scikit-learn prediction-based models for amino acid composition (AAC).

PSSM + AAindex													
number of	gamma	Dist	main							independent				Dist
classes	cost	ance	acc	sens	spec	mcc				acc	sens	spec	mcc	ance
7	different	0.07	77.76	77.57	77.80	0.42	b		b	76.50	72.00	77.25	0.37	0.05
		0.07	77.32	77.69	77.26	0.42	sh	scikit	sh	75.78	73.83	76.11	0.37	0.05
	same	0.07	76.74	78.72	76.41	0.42	b	learn	b	75.71	73.34	76.11	0.37	0.05
		0.07	76.99	77.95	76.84	0.42	sh		sh	75.38	73.83	75.64	0.37	0.05
	different	0.09	76.10	76.41	76.05	0.40	b		b	74.24	70.67	74.84	0.34	0.08
		0.08	77.13	76.03	77.31	0.41	sh	svm	sh	74.60	69.67	75.42	0.34	0.08
	same	0.09	75.88	78.20	75.49	0.40	b	light	b	74.60	75.50	74.44	0.37	0.06
		0.09	76.12	77.69	75.85	0.40	sh		sh	74.67	75.33	74.55	0.37	0.06
8	different	0.08	80.89	80.80	80.90	0.46	b		b	79.57	77.33	79.89	0.43	0.02
		0.09	81.73	81.96	81.70	0.48	sh	scikit	sh	79.63	77.11	79.98	0.42	0.02
	same	0.08	80.80	80.94	80.78	0.46	b	learn	b	79.85	77.66	80.16	0.43	0.02
		0.09	81.58	81.81	81.54	0.48	sh		sh	79.89	77.44	80.24	0.43	0.02
	different	0.08	80.77	78.05	81.16	0.45	b		b	78.94	71.78	79.97	0.39	0.02
		0.08	81.42	78.41	81.85	0.46	sh	svm	sh	79.20	71.45	80.30	0.39	0.02
	same	0.08	81.11	80.14	81.25	0.46	b	light	b	79.24	78.55	79.33	0.43	0.02
		0.08	81.35	81.09	81.39	0.47	sh		sh	79.51	77.56	79.79	0.43	0.02
b and sh are balanced and shuffled instances of the main dataset. Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient														

Table 4: The results from running 10% best models on the hybrid feature set including AAindex and PSSM for both main and independent datasets. This feature set outperfms the other 18 combinations.

number of classes	gamma cost	Distance	DPC acc	DPC sens	DPC spec	DPC mcc				PHC acc	PHC sens	PHC spec	PHC mcc	Distance
7	different	0.07	75.074	75.256	75.044	0.38	b	scikit learn	b	72.034	72.180	72.006	0.33	0.06
		0.07	74.358	74.36	74.358	0.36	sh		sh	71.098	71.154	71.088	0.31	0.07
	same	0.07	74.78	74.998	74.744	0.37	b		b	71.008	71.154	70.984	0.31	0.07
		0.07	74.396	74.104	74.444	0.36	sh		sh	70.918	70.898	70.918	0.31	0.07
	different	0.07	74.962	74.486	75.044	0.37	b	svm	b	72.400	72.306	72.412	0.33	0.06
		0.06	74.376	74.614	74.336	0.37	sh		sh	71.888	71.538	71.944	0.32	0.06
	same	0.07	75.238	75.128	75.256	0.38	b	light	b	72.804	72.180	72.906	0.33	0.06
		0.07	74.818	74.232	74.916	0.37	sh		sh	71.942	71.666	71.988	0.32	0.06
8	different	0.12	78.254	78.55	78.21	0.41	b	scikit learn	b	74.056	74.276	74.026	0.34	0.07
		0.11	77.828	77.898	77.814	0.41	sh		sh	73.550	73.478	73.562	0.33	0.07
	same	0.13	78.95	78.116	79.068	0.42	b		b	74.184	74.132	74.190	0.34	0.07
		0.11	77.978	77.464	78.056	0.41	sh		sh	74.176	74.130	74.184	0.34	0.07
	different	0.12	78.024	78.476	77.96	0.41	b	svm	b	75.202	75.218	75.198	0.36	0.08
		0.12	78.242	78.188	78.25	0.41	sh		sh	75.644	75.940	75.600	0.37	0.09
	same	0.13	78.396	78.554	78.376	0.42	b	light	b	76.784	76.668	76.802	0.39	0.11
		0.13	78.38	78.696	78.334	0.42	sh		sh	75.644	75.940	75.600	0.37	0.09
number of classes	gamma cost	Distance	AAINDEX acc	AAINDEX sens	AAINDEX spec	AAINDEX mcc				PSSM acc	PSSM sens	PSSM spec	PSSM mcc	Distance
7	different	0.07	72.068	72.050	72.074	0.33	b	scikit learn	b	76.282	76.794	76.194	0.40	0.07
		0.08	71.684	71.282	71.750	0.32	sh		sh	76.010	77.950	75.682	0.40	0.08
	same	0.08	71.942	71.284	72.048	0.32	b		b	76.540	76.026	76.624	0.40	0.08
		0.08	71.686	71.408	71.732	0.32	sh		sh	76.130	77.820	75.860	0.40	0.08
	different	0.09	71.136	71.536	71.070	0.31	b	svm	b	75.934	75.514	76.006	0.39	0.08
		0.11	70.054	70.128	70.046	0.29	sh		sh	77.216	75.000	77.586	0.40	0.08
	same	0.09	71.264	71.280	71.260	0.31	b	light	b	75.840	76.920	75.660	0.40	0.07
		0.10	70.622	70.898	70.578	0.30	sh		sh	76.230	76.790	76.130	0.40	0.07
8	different	0.07	75.154	75.508	75.104	0.36	b	scikit learn	b	80.346	80.146	80.372	0.45	0.09
		0.07	74.910	74.492	74.968	0.35	sh		sh	80.136	81.958	79.876	0.46	0.10
	same	0.07	75.570	75.506	75.580	0.37	b		b	80.152	80.074	80.166	0.45	0.09
		0.07	74.294	75.000	74.192	0.35	sh		sh	80.884	81.086	80.860	0.46	0.10
	different	0.08	76.322	76.378	76.318	0.38	b	svm	b	80.308	80.216	80.318	0.45	0.09
		0.08	76.008	76.232	75.974	0.38	sh		sh	81.006	81.160	80.984	0.46	0.10
	same	0.08	76.142	76.014	76.160	0.38	b	light	b	80.354	80.508	80.332	0.45	0.09
		0.08	75.960	75.726	75.992	0.37	sh		sh	80.408	81.666	80.230	0.46	0.10

b and sh are balanced and shuffled instances of the main dataset.
Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient

Table 5: The results from running 10% best models for DPC, PHC, AAindex and PSSM feature sets on main dataset.

number of classes	gamma cost	Distance	AAC+DPC acc	sens	spec	mcc				AAC+PHC acc	sens	spec	mcc	Distance
7	different	0.07	76.830	76.412	76.900	0.40	b	scikit learn	b	74.854	74.486	74.914	0.37	0.07
		0.06	75.970	75.640	76.026	0.39	sh		sh	73.846	73.974	73.824	0.36	0.08
	same	0.06	75.898	75.642	75.940	0.39	b		b	74.798	74.488	74.852	0.37	0.07
		0.07	75.458	75.898	75.386	0.38	sh		sh	73.900	73.460	73.972	0.35	0.09
	different	0.08	74.526	74.360	74.552	0.36	b	svm	b	74.964	74.872	74.980	0.37	0.07
		0.08	74.212	74.490	74.166	0.36	sh		sh	74.196	74.488	74.146	0.36	0.08
	same	0.07	75.568	75.128	75.640	0.38	b	light	b	74.524	74.616	74.508	0.37	0.07
		0.08	74.212	74.486	74.166	0.36	sh		sh	73.168	73.974	73.036	0.35	0.09
8	different	0.11	80.084	80.362	80.040	0.45	b	scikit learn	b	77.998	77.896	78.012	0.41	0.08
		0.10	79.438	79.782	79.390	0.44	sh		sh	77.030	77.030	77.030	0.39	0.08
	same	0.12	80.434	80.072	80.484	0.45	b		b	77.928	77.682	77.962	0.41	0.08
		0.10	79.682	79.420	79.718	0.44	sh		sh	76.984	76.882	76.998	0.39	0.08
	different	0.10	79.374	79.494	79.358	0.43	b	svm	b	78.324	78.042	78.366	0.41	0.09
		0.09	78.986	78.840	79.008	0.42	sh		sh	77.870	77.680	77.898	0.41	0.08
	same	0.11	79.920	79.782	79.940	0.44	b	light	b	78.036	78.044	78.034	0.41	0.08
		0.10	79.394	79.276	79.408	0.43	sh		sh	77.628	77.606	77.628	0.40	0.08
number of classes	gamma cost	Distance	AAC+AAINDEX acc	sens	spec	mcc				AAC+PSSM acc	sens	spec	mcc	Distance
7	different	0.07	76.814	76.154	76.924	0.40	b	scikit learn	b	76.812	76.924	76.794	0.41	0.07
		0.06	75.532	75.898	75.470	0.39	sh		sh	75.934	75.386	76.028	0.39	0.07
	same	0.06	75.880	75.256	75.980	0.39	b		b	76.356	76.026	76.412	0.40	0.06
		0.07	74.248	74.358	74.230	0.37	sh		sh	75.332	75.130	75.364	0.38	0.07
	different	0.07	74.948	74.232	75.064	0.37	b	svm	b	76.042	76.410	75.980	0.39	0.07
		0.10	72.986	72.948	72.990	0.34	sh		sh	75.018	75.126	75.002	0.38	0.07
	same	0.07	74.396	74.998	74.296	0.37	b	light	b	75.144	75.128	75.150	0.38	0.07
		0.09	73.702	73.460	73.740	0.35	sh		sh	73.956	73.332	74.056	0.35	0.09
8	different	0.10	79.050	79.276	79.016	0.43	b	scikit learn	b	78.172	78.044	78.188	0.41	0.09
		0.09	78.206	78.044	78.228	0.41	sh		sh	78.278	78.118	78.304	0.41	0.09
	same	0.08	77.960	77.898	77.972	0.41	b		b	78.650	78.696	78.642	0.42	0.09
		0.08	77.082	77.028	77.092	0.39	sh		sh	78.396	78.550	78.374	0.42	0.09
	different	0.11	79.874	79.708	79.898	0.44	b	svm	b	78.198	78.694	78.126	0.41	0.09
		0.10	78.898	78.912	78.892	0.42	sh		sh	77.864	77.682	77.888	0.41	0.08
	same	0.10	79.012	79.204	78.986	0.43	b	light	b	77.832	77.826	77.838	0.41	0.08
		0.09	78.372	78.334	78.376	0.41	sh		sh	77.990	77.972	77.992	0.41	0.08

b and sh are balanced and shuffled instances of the main dataset.
Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient

Table 6: The results from running 10% best models for AAC+DPC, AAC+PHC, AAC+AAindex and AAC+PSSM hybrid feature sets on main dataset.

number of classes	gamma cost	Distance	DPC+PHC acc	sens	spec	mcc				DPC+PSSM acc	sens	spec	mcc	Distance
7	different	0.07	74.872	74.614	74.914	0.37	b	scikit learn	b	76.282	76.924	76.176	0.40	0.09
		0.07	74.782	74.230	74.870	0.37	sh		sh	76.354	76.156	76.390	0.40	0.09
	same	0.07	75.036	75.770	74.914	0.38	b		b	76.300	76.540	76.262	0.40	0.09
		0.07	74.854	74.616	74.892	0.37	sh		sh	76.006	76.538	75.918	0.39	0.09
	different	0.07	74.760	74.232	74.850	0.37	b	svm light	b	75.880	75.640	75.920	0.39	0.08
		0.07	73.956	73.848	73.974	0.36	sh		sh	75.550	75.386	75.578	0.38	0.08
	same	0.08	73.754	73.460	73.804	0.35	b		b	75.422	75.256	75.450	0.38	0.08
		0.09	73.004	73.078	72.990	0.34	sh		sh	75.458	75.386	75.470	0.38	0.08
8	different	0.10	78.178	78.622	78.116	0.41	b	scikit learn	b	78.106	78.550	78.042	0.41	0.12
		0.09	77.736	77.898	77.710	0.41	sh		sh	78.088	78.260	78.066	0.41	0.12
	same	0.09	77.952	77.970	77.950	0.41	b		b	77.444	77.968	77.370	0.40	0.11
		0.10	78.262	78.042	78.292	0.41	sh		sh	77.942	77.896	77.952	0.41	0.12
	different	0.09	77.934	77.898	77.942	0.41	b	svm light	b	77.946	77.754	77.970	0.41	0.12
		0.08	77.282	77.172	77.298	0.40	sh		sh	77.980	77.100	78.106	0.40	0.12
	same	0.08	76.994	76.594	77.052	0.39	b		b	77.446	77.900	77.382	0.40	0.11
		0.08	76.314	76.374	76.304	0.38	sh		sh	78.080	78.118	78.074	0.41	0.12
number of classes	gamma cost	Distance	DPC+AAINDEX acc	sens	spec	mcc				AAINDEX+PHC acc	sens	spec	mcc	Distance
7	different	0.07	75.330	75.000	75.384	0.38	b	scikit learn	b	71.978	71.668	72.030	0.32	0.07
		0.07	74.358	74.360	74.358	0.36	sh		sh	71.044	71.284	71.002	0.31	0.07
	same	0.07	75.002	75.000	75.002	0.37	b		b	71.264	71.026	71.302	0.31	0.07
		0.07	74.468	74.230	74.510	0.36	sh		sh	71.100	71.410	71.048	0.31	0.07
	different	0.07	74.596	74.616	74.596	0.37	b	svm light	b	72.436	72.306	72.456	0.33	0.06
		0.07	74.340	74.488	74.318	0.36	sh		sh	71.814	71.794	71.816	0.32	0.07
	same	0.07	74.854	74.870	74.850	0.37	b		b	72.490	72.692	72.458	0.33	0.06
		0.06	73.918	73.848	73.930	0.36	sh		sh	71.942	71.280	72.052	0.32	0.07
8	different	0.12	78.416	78.188	78.446	0.41	b	scikit learn	b	74.086	74.202	74.070	0.34	0.08
		0.11	77.854	77.970	77.836	0.41	sh		sh	73.560	73.478	73.572	0.33	0.07
	same	0.13	78.684	78.406	78.728	0.42	b		b	74.056	74.566	73.988	0.34	0.08
		0.11	77.854	77.464	77.908	0.40	sh		sh	74.088	74.130	74.080	0.34	0.08
	different	0.12	78.134	78.622	78.066	0.41	b	svm light	b	75.164	75.218	75.156	0.36	0.08
		0.12	78.160	78.044	78.180	0.41	sh		sh	75.202	75.000	75.226	0.36	0.08
	same	0.13	78.672	78.114	78.748	0.42	b		b	76.840	76.306	76.916	0.39	0.11
		0.13	78.922	78.044	79.048	0.42	sh		sh	75.924	75.942	75.922	0.37	0.10

b and sh are balanced and shuffled instances of the main dataset.
Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient

Table 7: The results from running 10% best models for DPC+PHC, DPC+AAindex, DPC+PSSM and AAindex+PHC hybrid feature sets on main dataset.

number of classes	gamma cost	Distance	AAC+DPC+PHC acc	sens	spec	mcc				AAC+DPC+AAINDEX acc	sens	spec	mcc	Distance
7	different	0.08	75.952	75.130	76.092	0.39	b	scikit learn	b	76.996	76.668	77.054	0.41	0.06
		0.09	74.578	75.128	74.486	0.37	sh		sh	76.006	76.794	75.878	0.40	0.06
	same	0.08	75.916	75.384	76.006	0.39	b		b	76.374	76.412	76.366	0.40	0.06
		0.08	75.366	75.258	75.384	0.38	sh		sh	75.988	75.640	76.046	0.39	0.06
	different	0.10	74.140	74.486	74.082	0.36	b	svm	b	74.506	74.744	74.466	0.37	0.07
		0.11	73.810	73.078	73.930	0.35	sh		sh	74.102	74.104	74.104	0.36	0.08
	same	0.10	74.138	74.872	74.018	0.36	b	light	b	75.074	75.768	74.956	0.38	0.07
		0.11	73.240	73.972	73.120	0.35	sh		sh	74.086	74.486	74.020	0.36	0.08
8	different	0.10	79.880	79.708	79.906	0.44	b	scikit learn	b	79.856	79.856	79.856	0.44	0.11
		0.09	78.948	78.766	78.976	0.42	sh		sh	79.014	79.202	78.984	0.43	0.09
	same	0.09	78.786	78.840	78.778	0.42	b		b	80.236	80.796	80.154	0.45	0.12
		0.09	78.896	78.404	78.964	0.42	sh		sh	79.648	79.926	79.606	0.44	0.10
	different	0.09	79.466	79.056	79.524	0.43	b	svm	b	78.676	78.842	78.654	0.42	0.09
		0.09	79.094	79.566	79.026	0.43	sh		sh	78.860	78.188	78.954	0.42	0.09
	same	0.08	78.396	78.622	78.364	0.42	b	light	b	79.810	79.854	79.804	0.44	0.11
		0.09	77.926	77.244	78.024	0.40	sh		sh	79.546	79.058	79.618	0.43	0.10
number of classes	gamma cost	Distance	AAINDEX+PSSM acc	sens	spec	mcc				AAC+DPC+PSSM acc	sens	spec	mcc	Distance
7	different	0.07	77.76	77.57	77.80	0.42	b	scikit learn	b	78.276	78.078	78.312	0.43	0.08
		0.07	77.32	77.69	77.26	0.42	sh		sh	77.232	77.438	77.202	0.41	0.07
	same	0.07	76.74	78.72	76.41	0.42	b		b	77.600	77.178	77.670	0.42	0.07
		0.07	76.99	77.95	76.84	0.42	sh		sh	77.090	77.054	77.094	0.41	0.07
	different	0.09	76.10	76.41	76.05	0.40	b	svm	b	75.988	75.898	76.006	0.39	0.06
		0.08	77.13	76.03	77.31	0.41	sh		sh	75.144	75.898	75.020	0.38	0.07
	same	0.09	75.88	78.20	75.49	0.40	b	light	b	75.972	75.896	75.984	0.39	0.06
		0.09	76.12	77.69	75.85	0.40	sh		sh	75.110	75.384	75.064	0.38	0.07
8	different	0.08	80.89	80.80	80.90	0.46	b	scikit learn	b	78.704	78.552	78.726	0.42	0.09
		0.09	81.73	81.96	81.70	0.48	sh		sh	78.614	78.622	78.610	0.42	0.09
	same	0.08	80.80	80.94	80.78	0.46	b		b	78.860	78.262	78.944	0.42	0.09
		0.09	81.58	81.81	81.54	0.48	sh		sh	78.486	78.768	78.446	0.42	0.09
	different	0.08	80.77	78.05	81.16	0.45	b	svm	b	77.318	77.100	77.352	0.40	0.08
		0.08	81.42	78.41	81.85	0.46	sh		sh	77.038	77.028	77.040	0.39	0.08
	same	0.08	81.11	80.14	81.25	0.46	b	light	b	78.566	78.042	78.642	0.42	0.09
		0.08	81.35	81.09	81.39	0.47	sh		sh	78.098	78.260	78.074	0.41	0.08
b and sh are balanced and shuffled instances of the main dataset. Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient														

Table 8: The results from running 10% best models for AAC+DPC+PHC, AAC+ DPC+AAindex, AAC+DPC+PSSM and AAindex+PSSM hybrid feature sets on main dataset.

number of classes	gamma cost	Distance	AAC+AAINDEX+PHC acc	sens	spec	mcc				AAC+AAINDEX+PSSM acc	sens	spec	mcc	Distance
7	different	0.07	74.854	74.486	74.914	0.37	b	scikit learn	b	77.068	77.694	76.966	0.41	0.07
		0.08	73.864	73.846	73.868	0.36	sh		sh	75.934	75.514	76.004	0.39	0.07
	same	0.07	74.414	74.744	74.360	0.37	b		b	76.208	76.282	76.196	0.40	0.06
		0.08	73.956	73.716	73.996	0.36	sh		sh	75.238	75.386	75.216	0.38	0.07
	different	0.07	75.000	75.128	74.978	0.38	b	svm light	b	76.082	76.796	75.962	0.40	0.06
		0.08	74.214	74.358	74.190	0.36	sh		sh	75.018	75.126	75.000	0.38	0.07
	same	0.07	74.468	74.616	74.444	0.37	b		b	75.018	75.000	75.020	0.37	0.08
		0.09	73.076	73.716	72.968	0.35	sh		sh	73.918	73.460	73.994	0.35	0.09
8	different	0.08	77.996	77.968	78.002	0.41	b	scikit learn	b	78.196	78.114	78.212	0.41	0.09
		0.08	77.038	77.030	77.040	0.39	sh		sh	78.010	78.406	77.950	0.41	0.08
	same	0.08	77.928	77.682	77.962	0.41	b		b	78.650	78.696	78.642	0.42	0.09
		0.08	76.984	76.956	76.988	0.39	sh		sh	78.532	78.190	78.582	0.42	0.09
	different	0.09	78.324	78.042	78.366	0.41	b	svm light	b	78.416	78.334	78.424	0.42	0.09
		0.08	77.852	77.680	77.878	0.41	sh		sh	77.798	77.608	77.826	0.40	0.08
	same	0.09	78.044	78.044	78.044	0.41	b		b	77.842	77.898	77.838	0.41	0.08
		0.08	77.628	77.606	77.628	0.40	sh		sh	77.990	77.972	77.992	0.41	0.08
b and sh are balanced and shuffled instances of the main dataset. Acc: Accuracy, Sens: Sensitivity, Spec: Specificity, Mcc: Matthews correlation coefficient														

Table 9: The results from running 10% best models for AAC+AAindex+PHC, AAC+AAindex+PSSM, hybrid feature sets on main dataset.

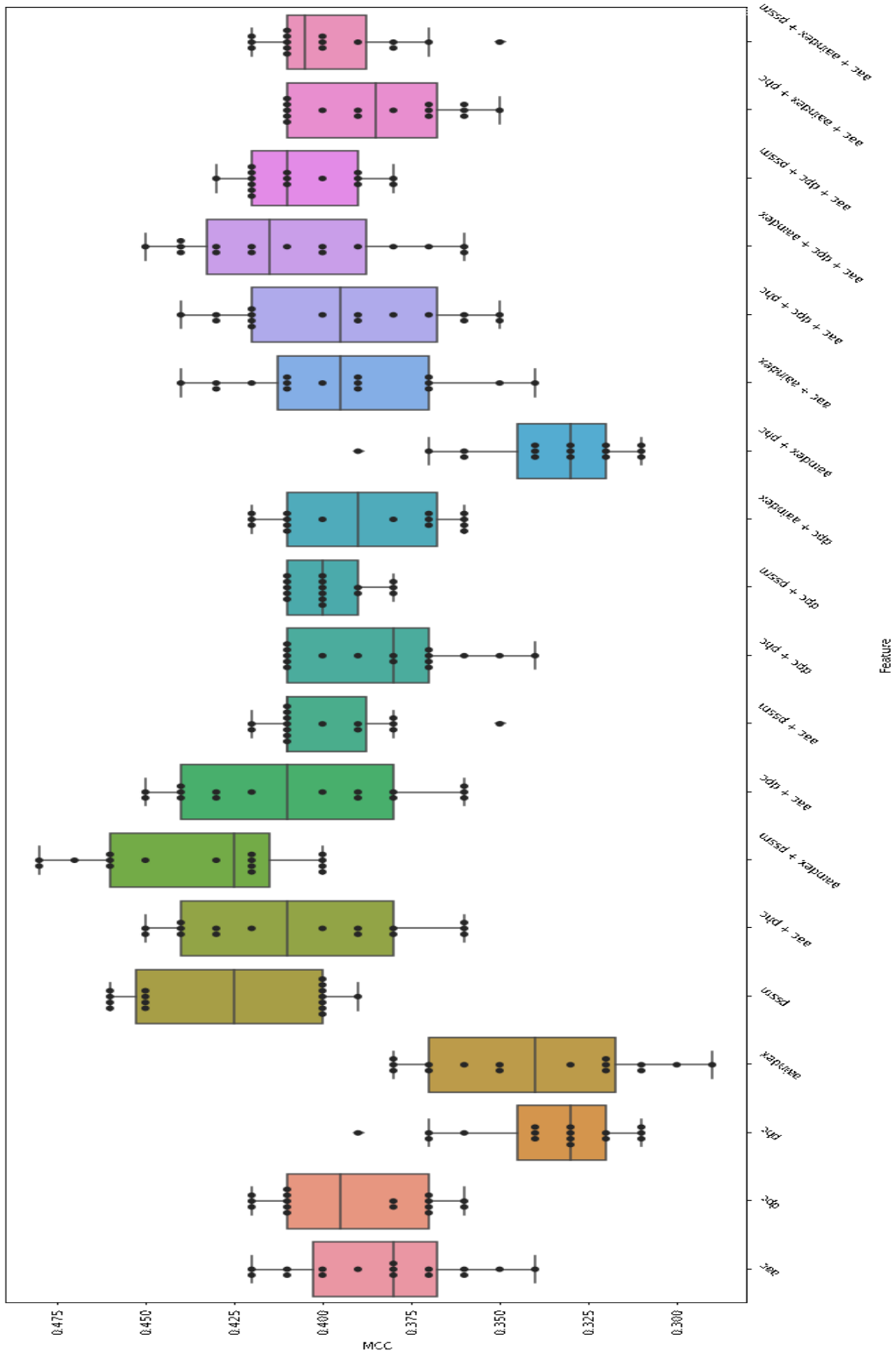

Figure 18: MCC results from applying the closest 10% models to all the features. The hybrid model that included the AAindex and the PSSM profile (7th box), outperforms others.

Chapter 5

Conclusion and future work

5.1 Conclusion

Reproducible research saves a great amount of time and budget as it enables other researchers to quickly run either the same experiment or a modified version of the same experiment for various purposes. However, through the past decade, concerns over the reproducibility of scientific results have been steadily rising with reports revealing a widespread lack of results reproducibility in various domains of science.

Since there are numerous parameters involved in building a model for a problem in machine learning, the experiments in this field of study are not immune to reproducibility-related issues either. When it comes to learning from multiple imbalanced sets of data, the process (from data sampling to calculating the performance metrics) requires even more analysis and considering more parameters.

In this work, we demonstrated that in an imbalanced learning problem with multiple classes in the dataset, a study report with a fair amount of details could generate a wide range of results on a reproducibility attempt if methodological flexibility is permitted.

For such a problem, flexibility allows researchers to make different assumptions for the parameters involved in constructing a model. As mentioned in 2.3, compared to balanced binary datasets, learning from imbalanced multiclass datasets involves more parameters. So, one should even make more assumptions for building a model. These different assumptions made by various experts could then lead to numerous results for the same problem. Some close to the initial one published in the study

and some far from it.

Another source of variation could be using different approaches (from the same library) for building a model and calculating the final performance metrics. The difference could occur due to the existing presumptions in the underlying layers of that specific approach for different phases of building a model on imbalanced data. Although insignificant, for the same classification algorithm (e.g. support vector machines), various libraries could also produce different results.

Among all the methodological flexibilities, we believe dataset sampling, aggregation and averaging techniques affect the final results the most.

Regarding the dataset, if the applied re-sampling technique balances out the sets (e.g. down-sampling, up-sampling, etc.), micro and macro averaging, produce close results. If the model is built on an imbalanced sets of data, since the ratio between the minority and the majority class error rates increase with the amount of available degree of imbalance in the dataset [44], the Imbalance Ratio should be kept the lowest. In such a scenario, stratified sampling with micro averaging technique produces the closest results.

With regards to the aggregation technique, According to Haibo He et al. [39]: "It has been stated that trying other methods, such as sampling, without trying by simply setting the threshold may be misleading". Because, usually, standard classifier learning algorithms are biased toward the majority class. "When studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake" [77]. So, for imbalanced learning problems, applying the threshold-moving technique is recommended.

For such problems, we believe the recommendations in appendix A could ensure an agreeable amount of reproducibility. We produced this recommendations as an extension to the "The Machine Learning Reproducibility Checklist". According to the authors in [75] the checklist was "designed to improve the standards across the community for how we conduct, communicate and evaluate machine learning research."

The recommendations are organized under data provenance, feature provenance and model provenance. According to W3C Incubator Group Report [37], provenance of a resource is a record that describes entities and processes involved in producing and delivering or otherwise influencing that resource. Provenance provides a critical

foundation for assessing authenticity, enabling trust, and allowing reproducibility.

5.2 Future Work

Considering the nature of imbalanced data, and characteristics of the machine learning algorithms (Section 2.3), learning from imbalanced sets of data requires more analysis. A researcher also needs to consider more parameters. When it comes to learning from multiple imbalanced classes of data, the process takes even more analysis and involves even more parameters.

So, there could be 2 directions to follow from here. The first path to follow would be to focus on exploring the approaches that could improve the results for such problems. It would be interesting to explore how combining several individual models on the same dataset could lead to a better generalization performance (ensemble learning for imbalanced data).

The second one would be to focus on the approaches through which reproducibility-related issues could be avoided for these types of problems. There are various applications for imbalanced data in various domains of science, reproducible experiments could save a big amount of time and budget.

Results Improvement In machine learning, combining several classifiers into a single one (ensemble learning) is known to improve performance. However, ensemble learning techniques by themselves are not able to solve the class imbalance problem.

To solve this problem, we need to adapt the ensemble learning algorithms. For this purpose, usually, we can combine an ensemble learning strategy with any of the methods that deal with the class imbalance (section 2.3). Different solutions mainly differ on how this hybridization is done and which ones are the methods considered for the construction of the new model.

On the other hand, there are several approaches for building a model in ensemble settings. The ensemble models could be broadly categorised into models like bagging, boosting and stacking, negative correlation based deep ensemble models, explicit/implicit ensembles, homogeneous/heterogeneous ensemble, decision fusion strategies, unsupervised, semi-supervised, reinforcement learning and online/incremental, multilabel based deep ensemble models [30].

An interesting field of study would be exploring hybridization techniques. One can research what type of hybridization would improve the performance in a specific domain. Combining and training the classifiers in this category also could increase the training cost. Hence, one can investigate the alternate ways of inducing diversity in the base models with lesser training costs.

Reproducibility-related Issues Through this path, one can conduct deeper analysis into various approaches for learning from imbalanced datasets through various domains of machine learning applications with a focus on the key points and parameters that could lead to an irreproducible experiment.

When it comes to learning from imbalanced sets of data, there are numerous approaches available for different applications in this field. Each model depending on the problem it solves has its own characteristics. These approaches could be explored and the key points could be addressed to improve reporting on the matter.

Appendix A

Reproducible Experiment Report

Data Provenance and Sharing
•Report on the source(s) of the raw data
•Explain the curation process
•Share the final dataset.
Feature Provenance and Sharing
•Explain the concept associated with the extracted feature
•Explain the process through which the feature is extracted
•For formulas, describe the associated parameters
•Share the extracted feature file (or reasonable amount of the final extracted feature file)
Model Provenance and Sharing:
Data Pre-processing
•Explain the pre-processing technique concept along with any involved process, formula and parameters
•Share the final transformed feature file (or reasonable amount of the final transformed feature file)
Model Structure
•Explain any applied sampling technique along with the process formula and parameters
•Describe the strategy for splitting the original data into train, validation and test
•For problems with multiple classes, describe the decomposition strategy
•Explain the deployed algorithm, considered range of hyper-parameters and the associated values for obtaining the published results. For multiple classes, this process should be done for all the decomposed models.
•If a specific optimal hyper-parameters search technique is used, provide the final deployed values resulted from the process, describe the method, any involved parameter(s), the process and how it has been applied to the model(s).
•For problems with multiple classes (or ensemble models), report on the structure, describe the underlying models and the aggregation strategy and how all those were applied to the model to generate the final results. If the results are generated using a different threshold rather than using the default one used by the classification algorithm, report on the threshold value for each decomposed model.
•Share reasonable amount of the generated results (where possible)
Model Evaluation
•Describe the choice of statistical method used for evaluation of the results, any involved formula and its parameter(s)
•If averaging through multiple results, describe the technique (micro vs macro)
•Define error bars (if any)

Bibliography

[1] Munira Alballa, Faizah Aplop, and Gregory Butler. Trancep: Predicting the substrate class of transmembrane transport proteins using compositional, evolutionary, and positional information. *PloS one*, 15(1):e0227683, 2020.

[2] Alawi A. Alsheikh-Ali, Waqas Qureshi, Mouaz H. Al-Mallah, and John P. A. Ioannidis. Public Availability of Published Research Data in High-Impact Journals. *PLoS ONE*, 6(9):e24357, September 2011.

[3] Rangachari Anand, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. Efficient classification for multiclass problems using modular neural networks. *IEEE Transactions on Neural Networks*, 6(1):117–124, 1995.

[4] ASCB. ASCB Member Survey on Reproducibility, 2015.

[5] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, May 2016.

[6] Pierre Baldi and Søren Brunak. *Bioinformatics: the machine learning approach*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2nd ed edition, 2001.

[7] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, June 2004.

[8] A. Bayat. Science, medicine, and the future: Bioinformatics. *BMJ*, 324(7344):1018–1022, April 2002.

[9] Andrew L. Beam, Arjun K. Manrai, and Marzyeh Ghassemi. Challenges to the Reproducibility of Machine Learning Models in Health Care. *JAMA*, 323(4):305, January 2020.

[10] C Glenn Begley and John PA Ioannidis. Reproducibility in science: improving the standard for basic and preclinical research. *Circulation research*, 116(1):116–126, 2015.

[11] Pablo Bermejo, Jose A. Gámez, and Jose M. Puerta. Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. *Expert Systems with Applications*, 38(3):2072–2080, March 2011.

[12] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[13] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PloS one*, 12(6), 2017.

[14] Paula Branco, Luis Torgo, and Rita Ribeiro. A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv:1505.01658*, 2015.

[15] Lukas Buehler. *Cell Membranes*. Garland Science, 2016.

[16] Ahmad Hassan Butt, Sher Afzal Khan, Hamza Jamil, Nouman Rasool, and Yaser Daanial Khan. A prediction model for membrane proteins using moments based features. *BioMed research international*, 2016, 2016.

[17] Ahmad Hassan Butt, Nouman Rasool, and Yaser Daanial Khan. A treatise to computational approaches towards prediction of membrane protein and its subtypes. *The Journal of membrane biology*, 250(1):55–76, 2017.

[18] John T Cacioppo, Robert M Kaplan, John A Krosnick, James L Olds, and Heather Dean. Social, Behavioral, and Economic Sciences Perspectives on Robust and Reliable Science. *stanford.edu*, 2015.

[19] John M. Chambers. *Software for data analysis: programming with R*. Statistics and computing. Springer, New York, 2008. OCLC: ocn191243189.

[20] Yves Chauvin and David E. Rumelhart, editors. *Backpropagation: theory, architectures, and applications*. Developments in connectionist theory. Lawrence Erlbaum Associates, Hillsdale, N.J, 1995.

[21] Nitesh V. Chawla, David A. Cieslak, Lawrence O. Hall, and Ajay Joshi. Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17(2):225–252, October 2008.

[22] John F Claerbout. *Earth soundings analysis: Processing versus inversion*, volume 6. Blackwell Scientific Publications London, 1992.

[23] Jon F. Claerbout and Martin Karrenbach. Electronic documents give reproducible research a new meaning. In *SEG Technical Program Expanded Abstracts 1992*, SEG Technical Program Expanded Abstracts, pages 601–604. Society of Exploration Geophysicists, January 1992.

[24] Peter Clark and Robin Boswell. Rule induction with cn2: Some recent improvements. In *European Working Session on Learning*, pages 151–163. Springer, 1991.

[25] Cynthia Dwork and Jonathan Ullman. The Fienberg Problem: How to Allow Human Interactive Data Analysis in the Age of Differential Privacy. *Journal of Privacy and Confidentiality*, 8(1), December 2018.

[26] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[27] Alberto Fernández, Salvador García, María José del Jesus, and Francisco Herrera. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. *Fuzzy Sets and Systems*, 159(18):2378–2398, September 2008.

[28] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C. Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from Imbalanced Data Sets*. Springer International Publishing, Cham, 2018.

[29] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(4):463–484, July 2012.

[30] MA Ganaie, Minghui Hu, et al. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*, 2021.

[31] V. García, R. A. Mollineda, and J. S. Sánchez. On the k-NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, 11(3-4):269–280, September 2008.

[32] Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12–341ps12, June 2016.

[33] M Michael Gromiha. *Protein bioinformatics: from sequence to function*. Academic Press, 2010.

[34] M Michael Gromiha and Yu-Yen Ou. Bioinformatics approaches for functional annotation of membrane proteins. *Briefings in bioinformatics*, 15(2):155–168, 2014.

[35] M Michael Gromiha and Makiko Suwa. Discrimination of outer membrane proteins using machine learning algorithms. *PROTEINS: Structure, Function, and Bioinformatics*, 63(4):1031–1037, 2006.

[36] M Michael Gromiha and Yukimitsu Yabuki. Functional discrimination of membrane proteins using machine learning techniques. *BMC bioinformatics*, 9(1):135, 2008.

[37] W3C Incubator Group. Provenance xg final report, 2010.

[38] Odd Erik Gundersen and Sigbjørn Kjensmo. State of the Art: Reproducibility in Artificial Intelligence. *aaai.org*, pages 1644–1651, 2018.

[39] Haibo He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

[40] Trevor Hastie, Robert Tibshirani, et al. Classification by pairwise coupling. *Annals of statistics*, 26(2):451–471, 1998.

[41] Michael Huerta, Florence Haseltine, Yuan Liu, Gregory Downing, and Belinda Seto. NIH working definition of bioinformatics and computational biology. *US National Institute of Health*, July 2000.

[42] Matthew Hutson. Artificial intelligence faces reproducibility crisis. *Science*, 359(6377):725–726, February 2018.

[43] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, editors. *An introduction to statistical learning: with applications in R*. Number 103 in Springer texts in statistics. Springer, New York, 2013. OCLC: ocn828488009.

[44] Nathalie Japkowicz. Concept-Learning in the Presence of Between-Class and Within-Class Imbalances. In Eleni Stroulia and Stan Matwin, editors, *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, pages 67–77, Berlin, Heidelberg, 2001. Springer.

[45] Thorsten Joachims. *Learning to classify text using support vector machines*, volume 668. Springer Science & Business Media, 2002.

[46] Wael Khreich, Eric Granger, Ali Miri, and Robert Sabourin. Iterative Boolean combination of classifiers in the ROC space: An application to anomaly detection with HMMs. *Pattern Recognition*, 43(8):2732–2752, August 2010.

[47] Gary King. Replication, Replication. *PS: Political Science and Politics*, 28(3):444, September 1995.

[48] Stefan Knerr, Léon Personnaz, and Gérard Dreyfus. Single-layer learning revisited: a stepwise procedure for building and training a neural network. In *Neurocomputing*, pages 41–50. Springer, 1990.

[49] D. E. Knuth. Literate Programming. *The Computer Journal*, 27(2):97–111, February 1984.

[50] Daniel Kozma, Istvan Simon, and Gabor E Tusnady. Pdbtm: Protein data bank of transmembrane proteins after 8 years. *Nucleic acids research*, 41(D1):D524–D529, 2012.

[51] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[52] Miroslav Kubat, Robert C Holte, and Stan Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, 30(2):195–215, 1998.

[53] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[54] Pedro Larrañaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, José A. Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, and Victor Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1):86–112, March 2006.

[55] Neal Lesh, Michael Mitzenmacher, and Sue Whitesides. A complete and effective move set for simplified protein folding. In *Proceedings of the seventh annual international conference on Computational molecular biology - RECOMB '03*, pages 188–195, Berlin, Germany, 2003. ACM Press.

[56] Yi Lin, Yoonkyung Lee, and Grace Wahba. Support Vector Machines for Classification in Nonstandard Situations. *Machine Learning*, 46(1/3):191–202, 2002.

[57] C.X. Ling, V.S. Sheng, and Q. Yang. Test strategies for cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1055–1067, August 2006.

[58] N. M. Luscombe, D. Greenbaum, and M. Gerstein. What is bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine*, 40(4):346–358, 2001.

[59] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[60] Maciej A. Mazurowski, Piotr A. Habas, Jacek M. Zurada, Joseph Y. Lo, Jay A. Baker, and Georgia D. Tourassi. Training neural network classifiers for medical

decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2-3):427–436, March 2008.

[61] Matthew B. A. McDermott, Shirly Wang, Nikki Marinsek, Rajesh Ranganath, Marzyeh Ghassemi, and Luca Foschini. Reproducibility in Machine Learning for Health. *arXiv:1907.01463 [cs, stat]*, July 2019. arXiv: 1907.01463.

[62] Nitish K. Mishra, Junil Chang, and Patrick X. Zhao. Prediction of Membrane Transport Proteins and Their Substrate Specificities Using Primary Sequence Information. *PLoS ONE*, 9(6):e100278, June 2014.

[63] Thomas E Nichols, Samir Das, Simon B Eickhoff, Alan C Evans, Tristan Glatard, Michael Hanke, Nikolaus Kriegeskorte, Michael P Milham, Russell A Poldrack, Jean-Baptiste Poline, et al. Best practices in data analysis and sharing in neuroimaging using mri. *Nature neuroscience*, 20(3):299–303, 2017.

[64] Michael Nilges and Jens P Linge. Bioinformatics-a definition. *Unité de Bio–Informatique Structurale, Institut Pasteur*, 2011.

[65] Babatunde K Olorisade, Pearl Brereton, and Peter Andras. Reproducibility in Machine Learning-Based Studies: An Example of Text Mining. *openreview*, 2017.

[66] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716–aac4716, August 2015.

[67] Albert Orriols-Puig and Ester Bernadó-Mansilla. Evolutionary rule-based systems for imbalanced data sets. *Soft Computing*, 13(3):213–225, February 2009.

[68] Yu-Yen Ou, Shu-An Chen, and M Michael Gromiha. Classification of transporters using efficient radial basis function networks with position-specific scoring matrices and biochemical properties. *Proteins: Structure, Function, and Bioinformatics*, 78(7):1789–1797, 2010.

[69] Harold Pashler and Eric–Jan Wagenmakers. Editors' Introduction to the Special Section on Replicability in Psychological Science: A Crisis of Confidence? *Perspectives on Psychological Science*, 7(6):528–530, November 2012.

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[71] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *arXiv:1201.0490 [cs]*, June 2018. arXiv: 1201.0490.

[72] R. D. Peng. Reproducible Research in Computational Science. *Science*, 334(6060):1226–1227, December 2011.

[73] Roger D Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011.

[74] Roger D. Peng, Francesca Dominici, and Scott L. Zeger. Reproducible Epidemiologic Research. *American Journal of Epidemiology*, 163(9):783–789, May 2006.

[75] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving Reproducibility in Machine Learning Research (A Report from the NeurIPS 2019 Reproducibility Program). *arXiv:2003.12206 [cs, stat]*, April 2020. arXiv: 2003.12206.

[76] Hans E. Plesser. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers in Neuroinformatics*, 11:76, January 2018.

[77] Foster Provost. Machine Learning from Imbalanced Data Sets 101 Extended. *Proceedings of the AAAI'2000 workshop on imbalanced data sets*, 68:1–3, 2000.

[78] Edward Raff. A step toward quantifying independently reproducible machine learning research. In *Advances in Neural Information Processing Systems*, pages 5486–5496, 2019.

[79] Sebastien Razakarivony and Frederic Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, 34:187–203, 2016.

[80] José A Sáez, Bartosz Krawczyk, and Michał Woźniak. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164–178, 2016.

[81] Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. Ten simple rules for reproducible computational research. *PLoS computational biology*, 9(10), 2013.

[82] NS Schaadt and V Helms. Functional classification of membrane transporters and channels based on filtered tm/non-tm amino acid composition. *Biopolymers*, 97(7):558–567, 2012.

[83] M. Schwab, N. Karrenbach, and J. Claerbout. Making scientific computations reproducible. *Computing in Science & Engineering*, 2(6):61–67, December 2000.

[84] Shuo Wang and Xin Yao. Multiclass Imbalance Problems: Analysis and Potential Solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1119–1130, August 2012.

[85] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, July 2009.

[86] Rachael Tatman, Jake VanderPlas, and Sohier Dane. A practical taxonomy of reproducibility for machine learning research. *Open Review*, 2018.

[87] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[88] Shuo Wang, Leandro L Minku, and Xin Yao. Dealing with multiple classes in online class imbalance learning. In *IJCAI*, pages 2118–2124, 2016.

[89] G. M. Weiss and F. Provost. Learning When Training Data are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research*, 19:315–354, October 2003.

[90] Kirstie Whitaker. Showing your working: a how to guide to reproducible research. page 5527634 Bytes, 2017. Artwork Size: 5527634 Bytes Publisher: figshare.

[91] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, 3(1):160018, December 2016.

[92] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

[93] Michał Woźniak, Manuel Grana, and Emilio Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17, 2014.

[94] Yi-Hung Liu and Yen-Ting Chen. Total Margin Based Adaptive Fuzzy Support Vector Machines for Multiview Face Recognition. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1704–1711, Waikoloa, HI, USA, 2005. IEEE.

[95] Hualong Yu, Shufang Hong, Xibei Yang, Jun Ni, Yuanyuan Dan, and Bin Qin. Recognition of multiple imbalanced cancer types based on dna microarray data using ensemble classifiers. *BioMed research international*, 2013, 2013.

[96] Xing-Ming Zhao, Xin Li, Luonan Chen, and Kazuyuki Aihara. Protein classification with imbalanced data. *Proteins: Structure, function, and bioinformatics*, 70(4):1125–1132, 2008.