PhD thesis
Classification and analysis of biological data

Dan Søndergaard

Supervisor
Christian Nørgaard Storm Pedersen

Bioinformatics Research Centre
Aarhus University
Denmark

February 2017
A dissertation presented to the
Faculty of Science and Technology
of Aarhus University
in partial fulfillment of the requirements for
the PhD degree
Abstract

As the amount of molecular data increases due to advances in sequencing technologies, the difficulty of analyzing the data has increased proportionally, posing challenges in diverse areas such as algorithm development, statistical modelling, data storage, and reproducibility. Meanwhile, machine learning has become the technology of the decade. Machine learning methods aim to automatically learn intricate patterns in data such that the patterns can be used for predictions on future data. Such methods are increasingly applied in bioinformatics, but not without new challenges. Many popular machine learning methods are difficult to understand and rely on thousands, millions and even billions of parameters. This makes the decisions made by the method almost impossible to understand.

In this thesis I first present two applications of the simple $k$-nearest neighbors machine learning method on protein sequence data. In both applications, the method performs surprisingly well and in many cases outperforms much more complicated methods. This serves as a reminder to always try the simpler method first.

However, in cases where the parameters learned by the method are interpretable, the parameters may provide interesting biological insights. In our work on non-ribosomal peptide synthetases, a modular protein which synthesizes peptides, our data analysis provides valuable, novel insights into the function of these proteins and demonstrates that we can recover knowledge of the protein structure previously reported in literature through sequence data only.

While many problems can be framed as classical classification problems such that existing methods can be applied effortlessly, some problems require a degree of modelling. One such problem concerns prediction of the location of the primary tumor in cancers of unknown primary (CUP). In CUP cases, the location of the metastasis is known, but the location of the primary tumor remains unknown. In this thesis I discuss the problem of primary tumor prediction and present a method based on gene expression data. The method improves the accuracy of the prediction by modelling impurities from the tumor-adjacent normal tissue.
Resumé


I denne afhandling vil jeg først præsentere to anvendelser af den simple $k$-nærmeste-naboer-metode med udgangspunkt i proteinsekvensdata. I begge anvendelser klarer metoden sig overraskende godt og udkonkurrerer i mange tilfælde langt mere komplicerede metoder. Dette er en påmindelse om altid at afprøve simple metoder først.


Acknowledgements

I wish to thank Christian N. S. Pedersen, my supervisor, for his continuous support throughout my PhD, for stopping by so often to talk about everything and nothing, and for his never-ending positivity.

My gratitude extends to my colleagues at the Bioinformatics Research Centre (BiRC). Thank you for your support, your trust and faith in me. Especially thank you to Ellen for always helping out with practicalities and the intricate ways of the Aarhus University administration.

I also wish to thank Chris Greening of Monash University, Melbourne, Australia, all of his lab, and the people I met at Monash during my three-month visit in Chris’ lab.

My time as a PhD student has also been made significantly more pleasant by my friends and family. I wish to thank Torben Muldvang Andersen, my study mate and friend through almost ten years, and without whom I would not be where I am today. I also wish to thank my friend Line Korsholm Lauridsen for her support, for always listening to my worries and for many, many joyful dinners.

Thank you to my family for always being there for me and for being the solid foundation that every human needs. Finally, I wish to thank my girlfriend, Anne Fabricius, for her support, love, and care, for always picking me up when I fall, and for the child, my son, who will soon meet the world.

This thesis is dedicated to my father, Martin Søndergaard, who was taken away much too early. Every day is a struggle to be as good a man, as pure hearted, and soon as loving a father as he. I will miss you always.

Humbly,
Dan Søndergaard
# Contents

Abstract v  
Resumé vii  
Acknowledgements ix  
Contents x  
Introduction 1

## I Overview  

1 Concepts in machine learning 7  
  1.1 Overview .................................................. 7  
  1.2 Evaluation .................................................. 8  
  1.3 Interpretability ............................................ 11  

2 Classification and analysis of protein sequences 13  
  2.1 Classification of protein sequences .................... 15  
  2.2 Learning important sites in protein sequences .......... 20  
  2.3 Conclusion and future work ............................ 27  
  2.4 Contributions ............................................... 29  

3 Prediction of the origin of cancers of unknown primary 31  
  3.1 Previous works .............................................. 33  
  3.2 Method ...................................................... 34  
  3.3 Results ...................................................... 35  
  3.4 Conclusion and future work ............................ 38  
  3.5 Contributions ............................................... 39
# Publications and manuscripts

## 4 PATBox: A Toolbox for Classification and Analysis of P-Type ATPases

4.1 Abstract

4.2 Introduction

4.3 Materials and Methods

4.4 Results

4.5 Discussion

4.6 Acknowledgments

4.7 Supporting Information

## 5 Computational Classification of P-Type ATPases

5.1 Summary

5.2 Introduction

5.3 Databases

5.4 Methods

5.5 Acknowledgments

## 6 Computational discovery of specificity-conferring sites in non-ribosomal peptide synthetases

6.1 Abstract

6.2 Introduction

6.3 Methods

6.4 Results and discussion

6.5 Web service

6.6 Conclusion

## 7 HydDB: A web tool for hydrogenase classification and analysis

7.1 Abstract

7.2 Introduction

7.3 Results and Discussion

7.4 Conclusions

7.5 Methods

7.6 Acknowledgements

7.7 Author Contributions

7.8 Competing financial interests

## 8 Prediction of primary tumors in cancers of unknown primary

8.1 Abstract

8.2 Introduction
Introduction

*It is a capital mistake to theorize before one has data.*
*Insensibly, one begins to twist the facts to suit theories,*
*instead of theories to suit facts.*

_Sherlock Holmes_

One of the most brilliant fictional characters ever envisioned, Sherlock Holmes, certainly understood the importance of data, and without question, he would have felt right at home in the increasingly popular field of data science. In the quote above, Holmes foresees the rise of data-driven, as opposed to hypothesis-driven, science. However, he fails to recognize the flaws and insecurities, as well as the many practical problems, that emerge when dealing with big data.

In recent years we have seen a deluge of data coming from all corners of society. In the field of bioinformatics the rise of next-generation sequencing has resulted in an exponential amount of sequence data. The scale of the data has challenged the ways we used to visualize, analyze, and store data, forcing us to innovate in all of these areas. More often than not, computers come in handy when trying to solve these challenges, and as the cost of sequencing has decreased, luckily so has the cost of compute power and storage.

Machine learning is one way of analyzing data computationally. The field of machine learning concerns itself with the theoretical and practical aspects of teaching computers to *learn* from data, and use what they learned to *predict* properties of new, but related, data. As a field, machine learning has experienced a surge in popularity in recent years and is being applied to everything from automatic speaker adjustment to autonomous cars, medical diagnosis, and stock market speculation.

During my PhD my primary focus has been the development and application of machine learning methods to answer biological questions.

The first half of my PhD was mainly concerned with protein sequence classification and motif detection, resulting in three papers [96, 56, 94] and one book chapter [95], reprinted in chapter 4, 6, 7 and 5, respectively. One paper [94] was the result of an on-going collaboration with Chris Greening of Monash University, Melbourne, Australia. During the second part of my PhD I visited Chris’ lab for three months working with data collection and protein sequence analysis. Working close to wet-lab scientists with very little bioinformatics knowledge was a rewarding and eye-opening experience.

During the second part of my PhD, I became interested in cancer research and therefore organized a bi-weekly journal club on cancer evolution and data analysis. I started a collaboration with Søren Besenbacher from the Department
of Molecular Medicine at Aarhus University Hospital and Svend Nielsen, PhD student at Bioinformatics Research Centre, working on a machine learning method for predicting the location of the primary tumor in cases of cancers of unknown primary. This resulted in a paper [97] and a manuscript [98], reprinted in chapter 8 and 9, respectively.

This thesis is structured as follows. For the uninitiated reader, chapter 1 contains a brief introduction to the parts of machine learning that are relevant for this thesis. Chapter 2 covers my work on protein classification and motif detection and chapter 3 my work on predicting primary tumors in cancers of unknown primary. In these chapters I describe the motivation for and contributions made to each project, walk the reader through the main results, and present related work, as well as my thoughts on future research. During my PhD I have been involved in several projects. Some of this work has been included in this thesis as manuscripts, while some work resulted in publications that I do not include because my contribution was minor or in the form of supervision:


I have also become the main contributor to gwf (https://github.com/gwforg/gwf), a locally developed workflow tool for orchestrating large, complex workflows.

The second part of the thesis, spanning chapters 4–9, contains reprints of the papers and manuscripts relevant to this thesis. The papers are included in chronological order and are listed here for reference:


• Chapter 8: D. Søndergaard et al. “Prediction of Primary Tumors in Cancers of Unknown Primary.” In: *Journal of integrative bioinformatics* 14.2 (July 2017)

Part I

Overview
Chapter 1

Concepts in machine learning

This chapter serves as an introduction to some concepts and terminology of machine learning with relevance to this thesis. Machine learning is a broad field spanning statistical modeling, mathematics, computer science and often knowledge specific to the domain of the problem that machine learning is being applied to. Since contributions to the field of machine learning have come from such distinct corners of science [12, 1, 52], the terminology used can be inconsistent and redundant. Thus, the terminology used in this chapter may not be completely consistent with the terminology of the included papers.

1.1 Overview

In broad terms we usually consider two broad classes of machine learning: supervised and unsupervised. Supervised machine learning is used when the data that must be analyzed is of the form \((x_i, y_i)\) where \(x_i\) is an observed data point and \(y_i\) is some property of \(x_i\), for example, \(x_i\) may be a tissue sample from a cancer patient and \(y_i\) the type of tissue. A data set is a set of data points \(D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}\). In the case of unsupervised learning only \(x_i\) is given. Some methods also support a mix of supervised and unsupervised learning. These methods are known as semi-supervised learners.

In general, \(x_i\) is called the observation, sample, feature vector, instance or independent variable, while \(y_i\) is known as label, class or dependent variable. When \(y_i\) is finite (and typically quite small) we say that we have a classification problem, while an infinite \(y_i\) means that we are dealing with a regression problem. The example above is a classification problem since \(y_i\) is a tissue type, of which there is a finite and small number. I will not cover issues related to regression problems further since the work presented in this thesis only deals with classification problems.
In machine learning, the assumption is that the observations and labels can be related by a target function \( y_i = f(x_i) \) where \( f : \mathcal{X} \rightarrow \mathcal{Y} \). \( \mathcal{X} \) is known as the input space and \( \mathcal{Y} \) as the output space. In short, \( f \) is a function that maps instances of the input space to instances of the output space (e.g. mapping cancer samples to tissue types). However, \( f \) is unknown and the above formulation does not restrict the complexity of \( f \). Thus, we must choose an appropriate \( f \).

In practice, choosing \( f \) consists of choosing a model and then training the model parameters, with the overall goal of finding an \( f^* \) that approximates \( f \) well. The model may also have parameters that are not to be learned, but must be specified by the user. These parameters are known as hyper-parameters. Machine learning models strive to be as general as possible, that is, ideally \( f \) should be able to estimate any function well. This is in contrast to statistical modeling where the model is often designed for the problem at hand.

The choice of a model is typically based on two types of criteria: objective and subjective. Objectively, the best model is the model which results in the best predictive performance on the given data. To quantify this we must find robust ways to evaluate models. However, we may also wish to consider a number of subjective criteria. How easy is the method to understand and explain? How well does it scale to the amount of data? Are the parameters of the model interpretable?

We will treat these subjects, model evaluation and interpretability, individually in the remaining sections of this chapter.

## 1.2 Evaluation

How well does our model perform on new data? This is the question we attempt to answer by model evaluation or model validation. One way of estimating the performance of a model on new data is to use \( k \)-fold cross-validation in which the data set is divided into \( k \) non-overlapping parts of which \( k-1 \) are used for training of the model and the remaining part is used as a test set. This process is repeated \( k \) times such that all parts have been used as a test set. This ensures that all of the data is used for evaluating the model and helps to avoid over-fitting.

Before dividing the data set, the samples are often shuffled such that samples belonging to the same label are distributed approximately evenly between the folds. Stratified \( k \)-fold cross-validation is a variant \( k \)-fold cross-validation which preserves the distribution of labels in the folds.

When \( k = N \) this method is known as leave-one-out (LOO) cross-validation. However, while LOO provides a better estimate of the predictive performance of the model since it maximizes the amount of training data in each fold, it is
**Figure 1.1:** Relationship between the entire data set (red box) and derivations used in the validation process. Initially, the data set is split into a cross-validation (blue box) and validation (dark gray box) set. The former is used for cross-validation and grid search. That is, the hyper-parameters of the model are chosen using only the cross-validation set. The predictor is then trained on the cross-validation set and predictions are made on the validation set. White boxes represent data used for training during cross-validation, while light gray boxes represent the test set.

<table>
<thead>
<tr>
<th>Cross-validation set</th>
<th>Validation set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fold 1</td>
<td></td>
</tr>
<tr>
<td>Fold 2</td>
<td></td>
</tr>
<tr>
<td>Fold 3</td>
<td></td>
</tr>
<tr>
<td>Fold 4</td>
<td></td>
</tr>
<tr>
<td>Fold 5</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.2:** The confusion matrix can be used as the basis of a wide range of measures. Each cell is filled out according to the true and predicted labels.

<table>
<thead>
<tr>
<th>Pred</th>
<th>True</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>TN</td>
<td>FP</td>
</tr>
<tr>
<td>+1</td>
<td>FN</td>
<td>TP</td>
</tr>
</tbody>
</table>

Often computationally unfeasible since $k$ models must be trained. In practice it is common to use 5 or 10 folds.

Additionally, a validation set is often isolated before any model validation is done. The validation set is excluded from any cross-validation and grid search performed to optimize the hyper-parameters of the model, and thus it serves as a completely independent data set that can be used to obtain an estimate of the accuracy of the model on new data. See figure 1.1 for an illustration of how the training, test, and validation sets are used and related.

When predictions have been obtained by the cross-validation the results are often summarized through one or more measures. Consider the simple case of binary classification, that is $y_i \in \{-1, +1\}$. Then the results can be summarized in a confusion matrix as shown in figure 1.2. This table lays the foundation for many of the measures used in machine learning literature, some of which will be described here.

The confusion matrix is a cross tabulation of the true and predicted labels of a set of samples. In figure 1.2, the rows and columns represent the predicted
and true labels, respectively. In the binary class case the table represents the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). We may derive a series of measures from these numbers. One such measure is the accuracy, which is defined as

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.
\]

That is, the number of correct predictions divided by the total number of predictions. The accuracy does not take unbalanced classes into account. This measure is commonly used due to its simplicity, despite the existence of more informative measures. However, using such measures makes comparison to other methods, which report accuracy, impossible.

Two measures that are often used together are the sensitivity and precision, defined as

\[
\text{Sensitivity} = \frac{TP}{TP + FN},
\]

\[
\text{Precision} = \frac{TP}{TP + FP}.
\]

The sensitivity measures the proportion of samples belonging to +1, which were also correctly predicted as +1. The precision measures the proportion of samples predicted as +1, which were truly +1. These measures are often combined into the F-score, defined as the harmonic mean of the sensitivity and precision,

\[
\text{F-score} = 2 \times \frac{\text{Sensitivity} \times \text{Precision}}{\text{Sensitivity} + \text{Precision}}.
\]

In cases where the positive class is larger than the negative class, the F-score may give slightly optimistic results. For example, if \(TP = 99, FN = 1, FP = 9, TN = 1\), the F-score is 0.95, even though only \(1/10\) of the samples in the negative class were predicted correctly. Additionally, the F-score is, in most cases, not symmetric. That is, switching the +1 and -1 labels will give a different F-score. While the accuracy is defined for any number of classes, the F-score can only be applied in the binary case.

Another commonly used measure is the area-under-curve receiver operating characteristic (AUC-ROC). To define what a ROC curve is, we will first assume that we are given a binary classifier which outputs a probability \(p\) for each prediction. If \(p \leq t\), where \(0 \leq t \leq 1\), the sample is classified as -1, and +1 otherwise. The constant \(t\) is known as the discrimination threshold.

The receiver operating characteristic is then a plot of the sensitivity (as defined above) as a function of the false positive rate (FPR), which is defined as \(\text{FPR} = \frac{FP}{FP + TN}\), for a range of discriminations thresholds, \(t\). The area under the ROC curve is maximized when the false-positive rate is close to zero.
and the true positive rate is close to one. If the predictions are random, the ROC curve will be a straight line on the diagonal of the plot, meaning that the area under curve will be 0.5.

While more reliable measures exist, such as the Matthews correlation coefficient and Informedness measure, these are not widely used. The work presented in this thesis primarily uses the accuracy and F-score, despite their fallacies since this is what is mostly used in machine learning literature in the bioinformatics field.

1.3 Interpretability

Interpretability is an ambiguous and rarely discussed property of machine learning models. However, as machine learning is increasingly applied to problems in clinical medicine and law, the need to understand a wider range of aspects regarding models and specific predictions is needed, and better definitions of interpretability are essential to be able to accurately discuss and evaluate interpretability [62].

Here we will consider one interpretation of interpretability and limit our interpretation to the problem of classification. Thus, our definition of interpretability is the degree to which a human is able to understand why a prediction was made. As argued in [62], even this definition is quite broad. Do we consider the process that lead to the prediction? Or are we interested in understanding the parameters of the model? Do we wish to understand how predictions are made in general, or do we want to understand a specific prediction? These are all valid questions that contribute to the overall understanding of the prediction.

One example of a reasonably interpretable machine learning model is the decision tree. Decision trees are binary trees in which a node represents a feature with a cut-off value, known as a decision rule. Each node splits the training set into two subsets, one with samples which satisfy the decision rule and one with samples that do not. The splits are chosen to separate samples belonging to different classes. Prediction is then performed by following the path given by the decision rules for the new sample until a leaf node is reached. The path represents a series of logical statements that resulted in the prediction.

While it is easy to understand how and why the prediction was made, it may be less obvious how the parameters of the model, the decision rules, were chosen and thus a decision tree is only interpretable to some degree. However, understanding the prediction may yield a better understanding of the data at hand, aid in outlier detection, and even give a better understanding of the specific problem domain.
In this thesis, we use model parameters to discover novel important sites in a protein family. Such discoveries would not have been possible if we had applied a model with less interpretable parameters, such as a neural network. This work is described in section 2.2.
Chapter 2

Classification and analysis of protein sequences

Proteins are the little machines of biology. Their responsibilities in the human body range from activation of nervous signals, destruction and decomposition of ‘dead’ proteins, and pumping of substrates through cell membranes. Proteins fold into complex three-dimensional structures and this structure defines the function performed by the protein. Since proteins are the active machinery of the body they are often targeted in drug development.

There are four categories describing protein structure. The primary structure describes the sequence of amino acids that makes up the protein. The secondary structure refers to the three-dimensional structure of local regions of the proteins, the most famous being the α-helix and β-sheet structures. The tertiary structure describes the overall three-dimensional structure of the protein, which is often a tightly bound spherical structure which minimizes the energy of the protein.

Proteins may also gather in protein clusters consisting of two or more individual proteins then known as subunits. When studying a cluster of proteins working collaboratively to exhibit a specific function, we study the quaternary structure of the proteins, that is, the interaction of the subunits to perform the function of the protein cluster.

In protein sequence analysis we are concerned with the primary structure of the protein. As sequencing has become cheaper and entire genomes have been made public, genomes can be scanned for coding regions and these regions can be translated into protein sequences. These sequences can be obtained from e.g. the UniProtKB [65] database, which has seen explosive growth since the introduction of next-generation sequencing. The latest release of the UniProtKB database (January 2018) contains 107,627,435 automatically annotated protein sequences and 556,568 manually curated sequences (see figure 2.1).
In contrast, obtaining the tertiary structure of a protein is a time-consuming and complex task often requiring years of work in the wet-lab, but while the tertiary structure is important for gaining a deep biological understanding of the mechanics of a protein, much can still be learned from computational methods on protein sequences. Traditionally, protein sequence classification has relied on the construction of a phylogenetic tree with the sequence of interest and a set of references of known type. The type of the sequence is then assigned based on its position in the tree. This process is labor intensive and requires the tree to be small enough to be meaningfully visualized.

The work presented in this chapter applies machine learning methods to protein sequence data to predict the specific subtype of a protein belonging to a protein family.

The next two sections describe my work on protein sequence classification and analysis. The work presented in section 2.1 demonstrates that simple methods such as $k$-NN can perform extremely well for protein sequence data and even outperform much more sophisticated methods, while the work presented in section 2.2 shows how we can obtain novel biological knowledge when using models with interpretable parameters. This work resulted in three papers and a book chapter:


The following sections each contain a brief history of the project, motivation, a brief overview of related work, and a short discussion of the method and results. My contributions to the work will be stated at the end of each section.

### 2.1 Classification of protein sequences

When developing a classifier for a data set the overall aim is to provide a quick and accurate prediction. However, in bioinformatics the classifier will likely be used by scientists that are not familiar with machine learning, or even programming. Thus, an additional requirement is that the classifier must be easy to use and understand.

Unfortunately, protein sequences do not lend themselves easily to analysis with common machine learning methods since protein sequences are variable-length strings with no inherent features and most machine learning methods are limited to working with fixed-size feature vectors [111]. Methods such as logistic regression, neural networks, decision trees and support vector machines all require fixed-length inputs \( x_i \in \mathbb{R}^D \) where \( D \) is the number of dimensions/features. Alternatively, generative models such as hidden Markov models can be used.

An obvious solution to this problem is to extract a fixed-size set of features from the sequences and then apply a common machine learning method to these feature vectors. Commonly, the number of occurrences of all possible \( k \)-mers (substrings of length \( k \)) in the sequence is used as a feature vector. However, this can lead to extremely high-dimensional and sparse vectors, and removes the possibility of learning patterns dictated by the location of the \( k \)-mers in the protein sequence.

Another solution is to define a (distance measure) between protein sequences and apply e.g. the \( k \)-nearest neighbors (\( k \)-NN) method [27]. We will now have
to pick a sensible distance measure $d : (X, X) \to \mathbb{R}$. The $k$-NN method omits the training phase and instead predicts the label of a new observation $x$ by computing the distance $d(x, x_i)$ for each $i = 1, \ldots, N$. A prediction is then made by performing a majority vote of the labels of the $k$ nearest points to $x$. This necessitates that $d$ is computationally cheap since it must be computed $N$ times for each prediction that is to be made. However, this again requires us to translate our variable-length sequences into fixed-length feature vectors, to avoid expensive alignment of sequences, and thus our method will suffer from the same problems as described above.

In the following sections the application of the $k$-nearest neighbors method will be demonstrated on two proteins sequence data sets. Instead of inventing a novel distance measure, we applied the BLAST+ search engine for protein sequences to rank sequences according to a query sequence. The top $k$ results were then subjected to a majority vote to obtain the final prediction. For both applications, we found the method to perform well, often outperforming more complicated methods.

**Application to P-Type ATPases**

The work presented in this section was initially prompted by a collaboration with Poul Nissen at the Centre for Membrane Pumps in Cells and Disease (PUMPkin), Aarhus University. The goal was to develop an accurate classifier for predicting the subtype of a P-Type ATPase sequence and make the classifier available as a web service. The work resulted in a book chapter [95] and a published paper [96].

The P-Type ATPases protein family consists of membrane proteins which pump ions through the cell membrane. Since the first P-Type ATPase, the $Na^+/K^+$ ATPase, was discovered by Jens-Christian Skou in 1957 [93] the understanding of the significance of the P-Type ATPases has only increased.

P-Type ATPases are an important part of the regulatory system in all kingdoms of life [14] and their malfunction has been related to a wide range of diseases such as Menkes and Wilson disease, heart failure and several neurodegenerative disorders [14]. This makes P-Type ATPases interesting drug targets, but their ubiquity in eukaryotic cells also make them interesting in studies of evolution.

Based on the analysis of 159 protein sequences, Axelsen and Palmgren found that the P-Type ATPases can through phylogenetic analysis be divided into five major groups (denoted P1, P2, P3, P4, P5) according to the substrate being pumped [8]. These groups can be divided further into subtypes (e.g. P1A and P3B) which have been shown to correspond to the specificity of the pumps.
Multiple classifiers have been developed for predicting the subtype of P-Type ATPases given only the protein sequence. One attempt was based on a Structured Logistic Regression (SLR) method described in [50]. SLR is a logistic regression based method which aims to find subsequences that separate the classes (subtypes) well by searching sequence-space using a bound. The method was applied to P-Type ATPases in [80], where it was used to build a classifier for predicting whether a sequence was a P-Type ATPase or not, and a classifier for predicting the subtype of the sequence. To improve the performance of the classifier, a re-training procedure was implemented on top of SLR. The classifier obtained an AUC of 99.61%. Additionally, the authors implemented a method based on profile hidden Markov models which provided a slightly higher AUC (99.99%), but was 8 times slower than SLR.

In [34] the author develops a classifier based on neural networks and demonstrates the complexity that must be dealt with when applying traditional machine learning methods to sequence data. The author conducted experiments with sparse vectors of \( n \)-gram counts for \( n = 1, 2, 3, 4 \). Additionally, the author applies the well-known TF-IDF scoring method usually used in text mining to reduce the dimensionality of the feature vectors by only including \( n \)-grams with a high TF-IDF score. The final classifier obtains an accuracy of 99.1% on 5/6 of the data set.

The classifier presented in [80] was made available as a web service, but the service is no longer available. The method developed in [34] was never made available.

We collected a data set of 515 sequences by combining data from [8, 73]. An effort was made to clean the data set by removing duplicates and sequences with invalid characters.

Inspired by the work in [34] we asked ourselves how well the simple \( k \)-NN method would perform. Since the P-Type ATPase subtypes can be easily isolated via a phylogeny, we expected this method to perform well since BLAST+ computes the distance between sequences via an approximate local alignment.

To evaluate the classifier, we performed repeated 5-fold cross-validations, shuffling the data set between each run. The best value of \( k \) was determined by evaluating the classifier for all \( k = 1, \ldots, 50 \) and picking the \( k \) which resulted in the highest accuracy. For \( k = 1 \) we obtained an accuracy of 100%.

Robustness of the classifier was evaluated by filtering the data with CD-HIT [49] at 30%, 50%, 75%. The results showed that the prediction accuracy is robust and remains high, even for highly reduced data sets.

This work presents a minor increase in accuracy on the problem of predicting the subtype of P-Type ATPases. However, we believe that it demonstrates that a simple method like \( k \)-NN can be surprisingly powerful and sometimes even beat much more complicated methods. As computers become more powerful,
the machine learning community has worked towards increasingly complex methods. This work demonstrates that this may not always be necessary.

Ultimately, the most significant contribution of this work is the PATBox web service which makes the classifier available, but also includes a browser for the P-Type ATPase data set. The web service was developed in Python using the Flask framework, and is packaged and deployed as a set of Docker containers, orchestrated by Docker Compose. The service is available as https://services.birc.au.dk/patbox. Usage statistics for the service are not available, but we have received multiple requests for support since the service was launched. A screenshot of the service is shown in figure 2.2.

Application to hydrogenases

Shortly after the work described in section 2.1 was published we were contacted by Chris Greening, Monash University, Australia, whose primary interest is the metabolism of bacteria, and specifically how the metabolism adapts to extreme environmental conditions and changes. P-Type ATPases are involved extensively
in the metabolism of most organisms, and thus Chris had acquired an interest in the paper [96] and the PATBox web service. This initiated an on-going collaboration and resulted in a three-month visit at the Greening Lab at Monash University.

One protein of particular interest to bacterial metabolism is the hydrogenase protein family. Hydrogenases are an important actor in the metabolism since they are able to produce H₂ from hydrogen ions, but also split H₂ back into hydrogen ions, which is an important part of energy conservation.

The hydrogenases can be divided into three groups known as [NiFe], [FeFe] and [Fe] based on their metal binding site, and have been subdivided further according to sequence similarity and a wide range of biochemical properties. The classification has refined continuously [47, 104, 16], and a major contribution of the work described in this chapter is the further refinement of this classification scheme through sequence-similarity networks.

The curation and classification resulted in a data set consisting of 3248 hydrogenase sequences and their associated subtype (e.g. [NiFe] Group 1b and [FeFe] Group C1). Using this data set, we built a classifier based on the $k$-NN with BLAST+ as a distance measure and evaluated it with 5-fold cross-validation. The classifier obtained an accuracy of 99.8%. Additionally, the classifier was tested against 24 newly obtained hydrogenase sequences from 12 genomes from bacteria, eukaryotes, and archaea. All sequences were classified correctly when compared with the classification obtained through a sequence-similarity network (SSN). The classifications were used to predict the physiological roles of each hydrogenase. We then compared the predictions with existing literature and metabolic models and found them to be in agreement.

Through the process of evaluating the classifier we found several sequences that were mislabeled in the data set, and thus building the classifier increased the quality of the data set itself. These experiments also made it clear that the four subgroups of [FeFe] Group A, denoted A1, A2, A3 and A4, were frequently misclassified. These subgroups can not be separated via sequence similarity since they were derived from the genetic organization, as noted in [47]. Instead, the downstream protein can be used to separate the four subgroups, and thus we implemented a post-processing step which allows the user to provide the sequence of the downstream protein. This secondary classification step uses the CDD [67] service to determine which conserved domains are present in the sequence and these are used to assign a subgroup to the sequence according to a simple set of rules.

Additionally, prior to classification, the pipeline optionally checks whether the sequences in question are hydrogenases. While this check is trivial, it is useful for researchers who often submit proteins from entire genomes to find hydrogenases.
Figure 2.3: Screenshot of the HydDB web service.

The classifier is made available through a web service, HydDB. The classifier is integrated with an interface for browsing and filtering the hydrogenase data set, along with detailed descriptions of each subgroup, references to relevant literature, and integration with the NCBI and PDB databases. Thus, HydDB is a unique and definitive resource for researchers seeking to understand hydrogenases.

The web service was implemented in Python with the Django framework, and is packaged and deployed as a set of Docker containers. Deployment of the service is orchestrated through Docker Compose. The service sees regular usage and has so far been used to classify approximately 760,000 sequences. The service is available at https://services.birc.au.dk/hyddb. A screenshot of the web service is shown in figure 2.3.

2.2 Learning important sites in protein sequences

In the previous section we were concerned with classification for the sake of classification. However, some machine learning methods can also provide
A PCP C
. . .
A PCP C
F

Figure 2.4: Each NRPS module consists of at least three domains, the A, PCP and C domains. In a chain-like fashion, each module binds a substrate, which is then passed on to the next module where another substrate is attached to the first substrate.

valuable insights into the important features of the data we are analyzing. One of the most popular methods in this respect is the decision tree, which was briefly described in chapter 1. Since a decision tree learns a set of logical rules that lead to a prediction, it is trivial to track why a prediction was made. The learned decision tree itself can also be used to learn about the most important features that divide two classes, since these will be near the root of the tree. Decision trees, however, are not suited for sequence data and thus one might be interested in other methods that can provide a similar insight into the data.

In this section our work applying such a method, Sequence Learner (SeqL) [51] is described. The method was applied to a set of proteins belonging to the family of non-ribosomal peptide synthetases (NRPSs) which are primarily found in bacteria and fungi. Their function is to synthesize non-ribosomal peptides (NRPs), a large and diverse group of natural products with many applications in antibiotics and drug development in general [38]. NRPSs are particularly relevant for discovering novel antibiotics due to an increasing number of resistant bacterial strains. Tools such as antiSMASH [71] assist this process by scanning entire genomes for gene clusters of e.g. NRPSs and predict the resulting NRP.

Non-ribosomal peptide synthetases are large, modular proteins composed of at least three domains denoted the A, PCP (also known as T) and C. The A domain selects an amino acid/substrate which is ‘activated’ and then transferred to the PCP domain. The amino acid is then carried to the C domain which is responsible for attaching the activated amino acid to the next one in the chain [38]. The structure of NRPS is illustrated in figure 2.4. The diversity of NRPs produced is due to the fact that NRPSs can synthesize products composed of non-proteinogenic amino acids and other substrates. Currently, approximately 500 such substrates are known [15]. In 1999, Stachelhaus, Mootz, and Marahiel discovered that 10 residues in the A domain are essential in choosing the amino acid that is bound by careful analysis of the three-dimensional structure of a specific A domain sequence, PheA, with phenylalanine bound to it [100]. Shortly after, very similar work was presented in [22].
Through an alignment of the PheA sequence with other A domains and subsequent study of the 10 residues in the alignment, the authors found that they were able to accurately predict the substrate bound by each A domain. The code obtained from the alignment has later become known as the specificity-conferring code.

However, aligning sequences and extracting the specificity-conferring code to perform a prediction is cumbersome, and thus several more-or-less alignment-free attempts have been made to predict the substrate specificity of an A domain sequence through machine learning. I will devote the remainder of this introduction to a brief, chronological overview of these works, before giving a brief introduction to the SeqL method and the results obtained from our data analysis.

For reference, the methods presented here, and many other related to the mining of secondary metabolites such as NRPs have recently been collected in [108].

In 2015, Rausch et al. developed a semi-supervised method based on support vector machines (SVM) which does not directly predict the substrate specificity of an A domain, but predicts certain physico-chemical properties of the substrate, which are then clustered with the properties of known substrates, which produces a substrate prediction. Instead of using raw sequence data the authors extracted residues near the bound substrate and encoded these with a number of physico-chemical properties of the bound substrate. The SVM method was then applied to this data set [83]. An improved version of the method was published in 2011 [86]. The classifier was previously available at http://www-ab.informatik.uni-tuebingen.de/software, but is now defunct (tested January 29th, 2018).

Two years later, a HMM based method was published by Minowa, Araki, and Kanehisa. The authors constructed an alignment of A domains binding a variety of substrates and extracted sites with a high degree of conservation within each group of substrates. They then built profile hidden Markov models for each group and used these for the prediction. The method was tested on 418 A domain sequences and achieved an accuracy of 78%. Interestingly, the authors found that misclassifications would often result in structurally analogous substrates, and thus they obtained an accuracy of 89% when predicting into clusters of substrates with similar physico-chemical properties (as defined in [83]), instead of directly predicting the substrate. To my knowledge, the classifier is not available for download or as a web service.

The amount of A domain data available increased significantly in 2012 when Prieto et al. published a HMM based method for substrate prediction [82]. A major contribution of this paper was the construction and curation of a substantially larger database of 1598 A domains and their associated substrate specificity. Unfortunately, as later noted in [54], the data set included sequences whose sub-
strate specificity had been inferred from sequence only and not experimentally verified, as well as a number of non-NRPS and wrongly annotated sequences. Unlike the HMM based method of Minowa, Araki, and Kanehisa, the HMM profiles by Prieto et al. were constructed using complete A domain sequences. The method obtained an accuracy of 86.4%, however, the results are difficult to compare due to different data sets and number of supported substrates. The classifier was made available as a web service at \url{http://www.nrpssp.com/}.

Shortly after, another approach based on profile hidden Markov models was described by Khayatt et al. in [54]. Due to the poor quality of the data set provided in [82] the authors collected a new data set which went through a thorough curation process which resulted in a data set of 537 sequences distributed across 37 substrates. The HMM approach presented was novel since the authors chose to employ an ensemble strategy where some substrates are divided into several subtypes according to how they cluster in a phylogenetic tree. Thus, 51 HMMs were built for 39 substrates. The authors evaluated the classifier with leave-one-out cross-validation and obtained an accuracy of 84%, or 66% when including low-scoring results. Again, these results are difficult to compare to previous works since the data sets, the number of classes and evaluation methods differ. The authors made their classifier available as a web service at \url{http://www.cmbi.ru.nl/NRPS-PKS-substrate-predictor/}.

In 2014, an approach using latent semantic indexing (LSI) was described by Baranašić et al. [10] using the data set from [83]. The LSI method is based on the matrix decomposition of a matrix of counts of a term in a document. The method has its origins in the implementation of search engines for text documents. However, in this work the documents are protein sequences and the terms are \(n\)-grams extracted from an alignment of A domain sequences, augmented with the position of the \(n\)-gram in the multiple alignment. By computing the singular value decomposition of this matrix, a set of ‘concepts’ ranked by importance is produced along with a matrix of sequence-concept assignments. Only the most important concepts are used to reduce noise. To perform a prediction, the sequence in question converted into a vector of concepts assignments and a distance is computed using the cosine of the angle between the query vector and the vectors in the training set. The subtype of the highest scoring sequence is then returned as the prediction. Their method obtained an accuracy of 84% using leave-one-out cross-validation. The accuracy was slightly improved (87%) by truncating the sequences to the region which includes the 10 sites of the specificity-conferring code. However, a substantial improvement was obtained by only including the ten sites of the specificity-conferring code.

Interestingly, Baranašić et al. also constructed a 1-nearest neighbor classifier with BLAST+ as a distance measure, which is equivalent to the classifiers described in section 2.1. Using full A domain sequences this method obtained an
accuracy of 85% and thus outperforms the LSI method with location-augmented \( n \)-grams. Unfortunately, this result is not discussed further in the paper, which seems to confirm the issue raised in section 2.1, where it is discussed that simple methods are often ignored by researchers in the field of machine learning.

Some of these methods allow some degree of interpretability of the learned parameters. The profile hidden Markov models presented in [82, 54, 72] can be seen as probabilistic profiles describing characteristics of the protein family that it was built from. However, profile hidden Markov models are difficult to visualize. The LSI method by Baranašić et al. can be used to project the sequences to a lower-dimensional space (as illustrated in figure 1 in [10]) and thus the relations between subtypes and in-group clusterings can be exposed. For the SVM method by Rausch et al. there is no direct way to explore the parameters.

In 2016, we published [56] where the SeqL method was applied to the A domain classification problem. Shortly after an ensemble method based on several of the existing classifiers was published by Chevrette et al. [24].

The work described in the coming sections was carried out in collaboration with the Centre for Natural Non-Ribosomal Peptide Synthesis (NANORIPES), Aarhus University. We initially intended to see whether we could improve the accuracy of A domain classification using the SeqL method [51], but soon found that interesting biology could be derived from the trained models. This project is especially interesting because it grew into a data science project, the results of which were the basis for discovering previously unknown sites of interest in the proteins in question.

The remainder of this section will introduce the SeqL method and how it was applied to the NRPS sequences, before diving into our graphical representation of the parameters learned by SeqL and the results of studying this representation.

**Method**

We applied the Sequence Learner (SeqL) method [51] which is an improved version of the Sequence Logistic Regression (SLR) method developed by the same authors and described in [50]. Sequence Learner is a binary classification method which searches the space of all subsequences of the given sequences to find the subsequences that best separates the sequences into their respective classes.

In general, SeqL is trained with a data set \( D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \) where \( x_i \) is a sequence and \( y_i \in \{-1, +1\} \) describes whether \( x_i \) is in the negative or positive class. To support multiple substrates we constructed classifiers according to the one-vs-rest (OvR) strategy. A classifier is trained for each substrate using all \( N \) sequences in the training set. Thus, \( y_i \) is transformed such that \( y_i = +1 \) if the A domain binds the substrate and \( y_i = -1 \) otherwise.
As briefly mentioned earlier, SeqL learns sequences in the space of all sub-sequences of sequences in D. Each sequence \( x_i \) in D can be represented as a \( d \)-dimensional vector \( x_i = (x_{ij}, \ldots, x_{id}) \) such that \( x_{ij} = 1 \) if subsequence \( j \) occurs in \( x_i \) and \( x_{ij} = 0 \) otherwise. SeqL then applies gradient descent to find the set of parameters \( \beta = (\beta_1, \ldots, \beta_d) \) which minimize the regularized squared-hinge loss function,

\[
L(\beta) = \sum_{i=1}^{N} \max(1 - y_i \beta^T x_i, 0)^2 + CR_\alpha(\beta),
\]

where \( C \) is a constant and \( R_\alpha \) is the elastic-net regularizer with balancing factor \( \alpha \). Since the search space is massive, SeqL employs a bound based on the number of occurrences of the prefixes of a subsequence. Intuitively, a subsequence can not occur more often than its prefix, and thus SeqL can prune parts of the search space which contain subsequences whose prefix does not occur more often than the least occurring subsequence so far. The method is described in more detail in [51, 56].

What is noteworthy is that the trained model consists of a set of subsequences ranked by \( \beta_j \) which acts as a weight describing how well the subsequences separate the two classes (in our case how well the subsequence separates the substrate in question from all other substrates). This makes the model highly interpretable.

**Results**

The results described in [56] are twofold. First, we evaluated the classifiers and compared it to existing methods. Second, we developed a graphical representation of the motifs learned by the SeqL method and used it to rediscover the specificity-conferring code, but also regions which were not previously believed to be involved in substrate specificity.

**Classification performance**

The method was applied to the data set of Khayatt et al. [54]. We performed leave-one-out cross-validation and obtained an accuracy of 71.3%. We attempted a thorough comparison with existing methods. However, several factors made this difficult. The classifier of [82] is only made available through a web service which does not allow training the model on new data, and thus it was not possible to obtain the performance of the classifier on the new data set. Instead, we carried out leave-one-out cross-validation on our method with the data set from [82] and obtained an accuracy of 83.5%. Finally, we attempted a comparison
with the classifier by Khayatt et al. However, we were not able to train this classifier either. Also, the classifier uses both labeled and unlabeled sequences and thus a direct comparison is difficult under any circumstances. We did not attempt a comparison with [10] since we were unaware of its existence at the time. In short, our method performed equally well or better compared with existing methods, and did not require an alignment to be produced or any manual design of feature vectors.

Khayatt et al. and Rottig et al. provided single substrate results. Our method performs well on most substrates with Phe (phenylalanine) substrate being the exception. However, this is consistent with the other methods. We suspected that the low number of Phe sequences (15) in the data set could explain the issue, but several other substrates (Asp, Ile, Dhb-Sal, Orn, Aad, Dab, Gln) had the same or fewer samples, but were predicted with high accuracy. The fact that Phe is predicted badly is especially odd since Stachelhaus used an A domain binding phenylalanine to identify the specificity-conferring code. One reviewer pointed out that the issue had also been raised in [83, ‘Clustering of sequences with similar specificities’] where it is suggested that the behavior may be due to the existence of A domains that bind both Phe and Trp. We did not investigate the cause of the issue further.

**Graphical representation of subsequences**

We then turned our attention to the subsequences (or motifs) learned by SeqL. The goal was to investigate whether we could identify the specificity-conferring code of Stachelhaus, Mootz, and Marahiel based on the subsequences learned by SeqL.

For each substrate $s$ we extracted the list of subsequences that represented the positive class of the model, that is, the subsequences which were overrepresented in $s$. We then constructed a multiple alignment of all sequences in the data set and the sequence of PheA used by Stachelhaus, Mootz, and Marahiel. The subsequences were mapped to the PheA sequence by finding all occurrences of each subsequence in the multiple alignment and then mapping these positions to positions in the ungapped PheA sequence. For each position in the PheA sequence we then counted how many times a subsequence would overlap with that position. The counts were normalized by the total number of overlapping positions.

The result of our analysis is shown in figure 2.5. The substrates were grouped based on the grouping used in [86]. The plot shows clear peaks around the residues included in the specificity-conferring code (marked by gray bars) and thus it seems that SeqL can be used to rediscover the specificity-conferring code without the use of the three-dimensional structure or any expert knowledge.
However, there are also notable peaks around residue 110, 380 and 440. A literature search revealed that these locations had previously been reported by Marahiel, Stachelhaus, and Mootz [66, table 3] as conserved core sequences named A2, A6, and A8.

Interestingly, it is noted in [66] that A2 is highly conserved, but located far from the active site and thus the authors expected the core sequence to be conserved for structural reasons only. However, our analysis shows that it may well be indirectly involved in substrate binding. Additionally, we speculate the A6 core sequence may be involved in substrate specificity by limiting binding to substrates of a certain size. The role of the A8 core sequence is more unclear, but it may be involved in positioning of the molecules involved in substrate binding [56].

**Web service**

The web service was developed in Python using the Flask framework, and is packaged and deployed as a set of Docker containers, orchestrated with Docker Compose. The service is available as https://services.birc.au.dk/seql-nrps. A screenshot of the service is shown in figure 2.6. The user uploads sequences in FASTA format to the service. Predictions are shown with the colored score which signals the significance of the prediction. Each sequence is augmented with the subsequences which discriminate for and against the predicted substrate.

2.3 Conclusion and future work

Computational classification of proteins is an extensive field which poses many interesting problems due to the dimensionality of the space of protein sequences. As demonstrated in [34], additional measures must often be taken to reduce the dimensionality of the feature vector.

The work described in section 2.1 is the result of a very simple idea that works extremely well. Researchers commonly use BLAST+ as a way of identifying the type and organism of a protein, so it is not surprising that it also performs well.
in the classification scenario. To our knowledge, BLAST+ has not previously been used much in a more automated classification setting or evaluated in the context of machine learning. In this work we have treated BLAST+ as a black box, but internally BLAST+ also extracts \( k \)-mers and uses these to find similar sequences, before performing a ‘lazy’ local alignment, to limit the search space. Thus, using BLAST+ is merely a shortcut, but a shortcut that makes the method easy to understand for users, which is a major advantage in bioinformatics since the tools we develop are often made for less technically inclined users.

The work detailed in section 2.2 is also motivated by a classification problem, but in this case the interesting result was not the classifier itself, but the biological insights that could be gained through an analysis of the learned parameters of the classifier.

While the machine learning is central in the work presented in this chapter, the development and maintenance of the web services developed for each classifier represents a significant amount of work. An effort has been made to package and deploy the web services in a reliable, automated way to minimize the burden of maintenance, but also to ensure that the services remain avail-

Figure 2.6: Screenshot of the SEQL-NRPS web service.
able in the distant future. This has been accomplished by running the services, including databases, job queues and workers in separate Docker containers, thus ensuring that the services are isolated and can be deployed easily. This is especially important since all of the services share a single virtual machine. To minimize friction for users, the HydDB service uses Rollbar, an error collection and alert service, such that errors are automatically collected with as much debugging information as possible. In multiple cases, this has meant that bugs would be fixed before the user reported the issue herself.

Usage statistics for the PATBox and SEQL-NRPS services have not been collected, but we have received inquiries regarding both services. The usage statistics of HydDB has proven that it is a valuable resource for the hydrogenase community. Since the paper was published approximately 760,000 sequences have been classified distributed among almost 2000 jobs.

I do not have plans to continue working on PATBox or SEQL-NRPS, but will continue to maintain and add features to HydDB, as needed, in collaboration with Chris Greening. After our collaboration on the HydDB paper I visited Chris for three months at Monash University in Melbourne, Australia, which resulted in a new project regarding the RubisCO protein. This work is on-going.

2.4 Contributions

Section 2.1 I contributed to this work by implementing and evaluating the classifier, implementing and maintaining the PATBox web service, and writing the paper [96]. Additionally, I co-authored relevant sections of [95].

Section 2.2 I developed and evaluated the $k$-NN classifier, developed the pipeline for hydrogenase checking and the HydDB web service. I also contributed several sections to the paper [94].

Section 2.2 I contributed to idea development, implementation and evaluation of the method, implementation and maintenance of the web service, comparison with existing methods and co-authored the paper [56].
Chapter 3

Prediction of the origin of cancers of unknown primary

The previous chapter described work that was mainly carried out during the first part of my time as a PhD student. This chapter describes work that was mainly carried out during the second part of my PhD in collaboration with Søren Besenbacher from the Department of Molecular Medicine (MOMA) at Aarhus University Hospital. The work described has so far resulted in the following manuscripts:


In the first manuscript, we describe our method and evaluate it on data from TCGA and a small set of metastatic samples. The second manuscript, which is in preparation, expands upon this work by including data from the Genotype-Tissue Expression (GTEx) project [63]. Additionally, the classifier is made available as a software package.

According to the World Health Organization (WHO), cancer is the second largest cause of death worldwide. This has devastating implications for families that need to cope with the loss of a loved one, but also has far-reaching implications for the world economy in general. The WHO estimates that the total cost of cancer in 2010 was approximately $1.16 trillion. Cancer is most commonly caused by external factors such as smoking, excessive exposure to sunlight, excessive consumption of alcohol and obesity [77].

Cancer is not one disease, but a group of diseases all characterized by uncontrolled cell division usually caused by mutations to either tumor suppressor or
oncogenes. The uncontrolled cell growth causes a tumor to form. As sequencing has become cheaper it has become possible to sequence individual tumors to gain insights into which somatic mutations may be causing the cancer. The idea of precision medicine is to employ such information clinically to design better treatments and reduce side effects by tailoring the treatment to the genetics of the patient.

Cancers are typically named and categorized by their location, also known as the histological type (for example, breast cancer is cancer located in breast tissue). Cancers may also metastasize to other tissue types and form satellite tumors known as metastases. In this case there are two locations, the histological type, which is the location of the metastasis, and the primary site, which is the location of the primary tumor from which the metastasis originated. Treatments are administered according to the type of the cancer and thus knowing the location of the primary tumor is essential to treatment. In cases where a metastasis is found, but the location of the primary tumor remains unknown after extensive testing, the patient is diagnosed with a cancer of unknown primary (CUP). This happens in 3–5% of cases and the prognosis for such patients is poor [106].

Evaluation of methods predicting the primary in CUP cases are inherently difficult to train and evaluate since, by definition, the primary is unknown. Thus, it is not possible to obtain a data set where $x_i$ represents tissue from the CUP and $y_i$ is the origin of the primary tumor, and thus most machine learning methods for primary prediction use primary tumor data for training. The works described below demonstrate the diversity of the methods used for evaluation and indeed, evaluation was one of the main challenges in our own work [97].

Additionally, methods must be able to deal with impurities that occur due to human error when extracting biopsies, which causes some fraction of the sample to consist of normal cells from the tissue surrounding the tumor, but also due to white blood cells invading the tumor known as tumor-infiltrating leukocytes [40]. Work by Aran, Sirota, and Butte has shown that the degree of tumor purity is variable across tissue types [5]. Methods such as ESTIMATE and DeMix [112, 3] can be used to estimate and remove noise generated by non-tumor cells in gene expression data, a process known as devoncolution.

We speculated that prediction of primary tumors in CUP cases could be improved through two measures. First, the contribution of normal cells in a sample acts as noise and may confuse the prediction. Thus, our method models the contribution of normal cells to isolate the tumor component of a sample for improved prediction accuracy. Second, the tissue type of the normal component to a sample is always known as it corresponds to the excision site and therefore our method allows this parameter to be fixed during prediction to improve prediction of the tumor component of the sample.
Our method, which works directly with gene expression data, was thoroughly evaluated with primary tumors, simulated metastasis data, as well as with eight metastases of known origin.

3.1 Previous works

In recent years, large initiatives such as the Cancer Genome Atlas (TCGA) project [102] and the International Cancer Genome Consortium (ICGC) [21] have worked towards making substantial amounts of molecular data available from cancerous tissues. A wide variety of such molecular data has been used for primary tumor prediction.

Using expression data, Vikeså et al. [106] found that CUPs have unique features compared to metastases of known origin (MOKOs). The authors extracted a set of gene expression profiles for different tumor tissue types and defined an outlier score for each tissue type and used the outlier score to compute the distance between CUP and MOKO samples and their predicted primary site. They conclude that the distance for CUPs is generally higher and thus CUPs are more distantly related to their primary than MOKOs [106]. The authors also constructed a classifier based on linear discriminant analysis (LDA) which obtained an accuracy of 87% on MOKOs. Additionally, a high degree of chromosomal instability was demonstrated in CUP samples.

In 2016, Moran et al. developed a novel DNA methylation array and a classifier based on data from the array. The authors applied the random forest method to extract important CpG sites and then constructed a random forest classifier for primary site prediction also based on these CpG sites. The classifier was evaluated on a validation set of 7691 tumor samples. The classifier was also used to predict the primary of 188 samples from patients diagnosed with cancer of unknown primary. The results were evaluated experimentally when possible. In a single case, an autopsy was made to confirm the prediction, which lead to the discovery of two previously undiscovered metastases. In 38 cases, the primary tumor was detected long after the first examination of the patient. 87% of these were correctly predicted by the classifier. The authors also performed light microscopy evaluation (174 samples) and immunohistochemistry (31 samples) and found that 96% and 100%, respectively, were predicted correctly. Since the classifier uses methylation patterns it does not suffer from the instability of mRNA-based methods [32], such as the method described by [37].

Marquard et al. [68] developed a random forest classifier using somatic mutation data from the COSMIC database [39] with $K = 10$ tissues. Feature vectors were extracted manually.
Finally, a classifier based on multinomial linear ridge regression (MLRR) was developed at MOMA. However, this work is unpublished. The MLRR classifier was trained on gene expression data obtained from TCGA. Like the other classifiers introduced in this section, the classifier does not distinguish between tumor and normal samples.

### 3.2 Method

The gene expression data is represented as a $M \times N$ matrix where $x_{ij}$ corresponds to the $j$'th gene of the $i$'th sample. Each sample belongs to a specific tissue $y_i = 1, \ldots, K$, where $K$ is the number of tissues. Additionally, the sample is either tumor (T) or normal (N) tissue, denoted by $z_i \in \{T, N\}$.

We assume that a sample $x_i$ is a mixture of a tumor and normal component, both of which are modeled by a normal distribution. The degree of mixing between the two distributions is modeled by the mixing coefficient $0 \leq \alpha_i \leq 1$. We assume that $\alpha_i \sim B(\beta_{1k}, \beta_{2k})$ where $k = 1, \ldots, K$. That is, $\alpha_i$ has a beta distributed prior tissue-specific shape parameters which are assumed to be known.

Thus, the likelihood for tissue $k$ is

$$L_k(\alpha_1, \ldots, \alpha_{M_k}, \mu_{T_k}, \mu_{N_k}, \sigma_k^2) = \prod_{i=1}^{M_k} N(x_i; \alpha_i \mu_{T_k} + (1 - \alpha_i) \mu_{N_k}, \sigma_k^2) B(\alpha_i; \beta_{1k}, \beta_{2k}),$$

where $\mu_{T_k}$ and $\mu_{N_k}$ are centroids representing tumor and normal samples, respectively, from tissue $k$. To reduce over-fitting we add a ridge regularization term on the centroids with regularization parameter $\lambda$,

$$R_\lambda(\mu_{T_k}, \mu_{N_k}) = \exp(\lambda/(2\sigma_k^2) \cdot (|\mu_{T_k}|^2 + |\mu_{N_k}|^2)).$$

The optimal value of $\lambda$ is found through a grid search.

Training the model consists of maximizing the above likelihood to obtain estimates of $\mu_{T_k}$ and $\mu_{N_k}$ for each tissue. Gradient ascent can be used to maximize the likelihood, but for efficiency reasons we apply a two-step process. First, we fix the $\alpha_i$'s and maximize the likelihood with respect to $\mu_{T_k}$ and $\mu_{N_k}$ since an expression for these can be derived analytically. We then maximize the likelihood with respect to the $\alpha_i$'s using the Newton-Raphson method implemented in SciPy [53]. We assume that samples representing normal tissue ($z_i = N$) are purely normal tissue and thus we set $\alpha_i = 0$ for $i \in \{i \mid z_i = N\}$ after each iteration. The two steps are repeated until the difference in likelihoods between iterations is smaller than a given convergence threshold.
Given a new sample \( \hat{x} \) and the adjacent normal tissue \( \hat{y} \), a prediction, \( y^* \), is obtained as

\[
y^* = \arg \min_{k=1,\ldots,K} \left[ \min_{0 \leq \alpha \leq 1} \left| \hat{x} - \alpha \mu_k + (1 - \alpha) \mu_N \right| \right].
\]

That is, we minimize to find the \( \alpha \) that results in the shortest distance between \( \hat{x} \) and the two centroids for each tumor tissue centroid, and pick the tumor tissue which gives the shortest distance.

In [97], the shape parameters \( \beta_{1k} \) and \( \beta_{2k} \) were obtained by fitting a beta distribution to consensus estimates (CPE) from Aran, Sirota, and Butte [5] for each tissue, and only tissues with CPE estimates were included. In [98] we apply a flat prior, \( \beta_{1k} = \beta_{2k} = 1 \), for tissues for which CPE data is not available.

The data is normalized to zero mean and unit variance. We then employ either principal component analysis (PCA) or linear discriminant analysis (LDA) to reduce the dimensionality of the data.

### 3.3 Results

We conducted a series of experiments on a range of data sets to assess the predictive performance of the classifier. First, we evaluated the classifier on the problem of primary tissue prediction. That is, during prediction it is assumed that the tumor and adjacent normal tissue is the same and thus the experiment shows how well the classifier is able to predict the tissue type of a sample. This experiment is denoted as primary (P). Second, we investigated the performance of the classifier with simulated metastasis samples, which we denote simulated (S). Finally, the classifier was evaluated on a set of eight real metastasis samples. The experiments were repeated with the MLRR classifier developed at MOMA (see section 3.1 for a short description of this classifier).

We obtained gene expression data from TCGA. The data set consists of \( m = 7065 \) samples and with expression for \( n = 18696 \) genes, and contains unpaired primary tumor (approx. 90%) and normal samples (approx. 10%) (D1). The samples are spread across \( K = 16 \) tissues. This data is split into a cross-validation (90%, D2) and validation set (10%, D3) in a stratified fashion such that the distribution of tissue types and tumor/normal samples is approximately the same in the two sets. A data set, D4, of metastatic samples was simulated from D3. To simulate \( n \) metastatic samples, \( n \) tumor and \( n \) normal samples are sampled with replacement. The tumor and normal samples are then paired such that the final sample \( x = \alpha x_T + (1 - \alpha) x_N \) where \( \alpha \) is drawn from the beta distribution of the tumor sample.

We conducted a grid search with 5-fold cross-validation to obtain the optimal hyper-parameters (dimensionality reduction method and regularization factor).
Table 3.1: Results for the primary (P) and simulated (S) experiments for both methods. The best parameters were found through a grid search for each experiment and method. On simulated metastatic data, our method clearly outperforms the MLRR method. Note that we in some cases obtain a higher accuracy on the validation data since more training data is available. Reprint of table 8.1 from chapter 8.

<table>
<thead>
<tr>
<th>Method</th>
<th>CV Accuracy (%)</th>
<th>Validation Accuracy (%)</th>
<th>Dimensionality Reduction</th>
<th>Number of Components</th>
<th>Regularization Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>P LoCUP</td>
<td>94.9</td>
<td>95.2</td>
<td>LDA</td>
<td>221</td>
<td>819.2</td>
</tr>
<tr>
<td>MLRR</td>
<td>96.4</td>
<td>97.2</td>
<td>LDA</td>
<td>15</td>
<td>0.05</td>
</tr>
<tr>
<td>S LoCUP</td>
<td>96.3</td>
<td>95.5</td>
<td>LDA</td>
<td>55</td>
<td>102.4</td>
</tr>
<tr>
<td>MLRR</td>
<td>91.1</td>
<td>90.8</td>
<td>LDA</td>
<td>105</td>
<td>0.1</td>
</tr>
</tbody>
</table>

For the classifier on the cross-validation set. For the S experiments, the cross-validation was modified such that the test set of each fold was used to simulate data according to the procedure explained above. Predictions were then made on the simulated data. The results are listed in the ‘CV’ column of table 8.1. In summary, the classifier obtained an accuracy of 94.9% in the primary experiment, which is slightly lower than the accuracy of 96.4% obtained with MLRR. In the simulated experiment, the classifier obtained an accuracy of 96.3% versus 91.1% with MLRR.

The classifiers were then trained on D2 using the optimal hyper-parameters and predictions were made on D3 (P) and D4 (S). Our classifier obtained an accuracy of 95.5% on the validation set, while MLRR obtained an accuracy of 90.8%. Since our method aims to model low-purity samples to improve the accuracy, these results are encouraging. See table 3.1 for an overview of the results.

To show that LoCUP indeed performs better on low-purity samples, we plotted the (true) mixing coefficients ($\alpha$) of the simulated samples against the prediction accuracy in figure 3.1. For samples with a low mixing coefficient, our method clearly outperforms the MLRR method, and performs as well as MLRR on high-purity samples (see figure 3.1).

As a final step of validation, we obtained eight metastases of known origin from MOMA. This data set consists of only eight samples and therefore does not allow us to make any definitive conclusions as to the performance of our method on real metastatic samples. However, predictions on this data set are encouraging and show that LoCUP performs as well as, or better than, the MLRR method. The results are listed in table 8.2.
The results presented so far were published in [97]. However, work on this project is on-going. To investigate whether the classification could be improved with additional data, we obtained data from the GTEX project. This data set consists of normal samples only from a variety of tissues. Since the number of normal samples in the TCGA data set is very low, we anticipated that the addition of normal samples from GTEX would improve the accuracy. An overview of the combined data set is shown in table 3.2. After inclusion of the GTEX data, the proportion of normal tissue samples is increased to approx. 35%. The number of tissues supported was increased due to two factors. First, tissues that were previously excluded due to a lack of normal samples in the TCGA data were now included. Second, tissues for which beta shape parameters could not be estimated from data in [5] had previously been excluded. However, we now apply a flat beta prior to such tissues. Thus, $K = 18$ tissues are now supported. Table 3.2 provides an overview of the new data set.

Running the experiments with the changes described above resulted in a significant increase in prediction accuracy. On simulated data, the accuracy on the validation set increased from 95.5% to 97.6%. These numbers can not be compared directly due to the difference in the number of classes and the merging of classes in the data set used in [97] and the extended data set. See table 3.3 for an overview of the results.

Finally, the LoCUP classifier is available as a Python package installable through the Python package manager pip. The package provides a set of commands for training, prediction, evaluation, and cross-validation on the LoCUP model, and provides a Python API for interacting directly with the

![Figure 3.1: Write caption!!! Reprint of figure 8.4 from chapter 8.](image)
Table 3.2: Overview of the new data set, consisting of tumor and normal samples from TCGA, as well as normal samples from GTEX. Initial experiments showed that none of the classifiers could distinguish between stomach and esophageal tissues and thus we decided to merge these classes.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>Normal GTEX</th>
<th>Tumor TCGA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adrenal Gland</td>
<td>135</td>
<td>0</td>
<td>258</td>
</tr>
<tr>
<td>Bladder</td>
<td>11</td>
<td>19</td>
<td>405</td>
</tr>
<tr>
<td>Brain</td>
<td>1091</td>
<td>0</td>
<td>672</td>
</tr>
<tr>
<td>Breast</td>
<td>170</td>
<td>114</td>
<td>1093</td>
</tr>
<tr>
<td>Cervix Uteri</td>
<td>11</td>
<td>0</td>
<td>305</td>
</tr>
<tr>
<td>Colon</td>
<td>285</td>
<td>51</td>
<td>380</td>
</tr>
<tr>
<td>Kidney</td>
<td>21</td>
<td>129</td>
<td>889</td>
</tr>
<tr>
<td>Liver</td>
<td>96</td>
<td>49</td>
<td>371</td>
</tr>
<tr>
<td>Lung</td>
<td>273</td>
<td>108</td>
<td>1016</td>
</tr>
<tr>
<td>Ovary</td>
<td>86</td>
<td>0</td>
<td>303</td>
</tr>
<tr>
<td>Pancreas</td>
<td>153</td>
<td>0</td>
<td>178</td>
</tr>
<tr>
<td>Prostate</td>
<td>80</td>
<td>52</td>
<td>497</td>
</tr>
<tr>
<td>Salivary Gland</td>
<td>57</td>
<td>44</td>
<td>520</td>
</tr>
<tr>
<td>Skin</td>
<td>824</td>
<td>0</td>
<td>103</td>
</tr>
<tr>
<td>Stomach+Esophageal</td>
<td>0</td>
<td>46</td>
<td>588</td>
</tr>
<tr>
<td>Testis</td>
<td>138</td>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td>Thyroid</td>
<td>268</td>
<td>59</td>
<td>505</td>
</tr>
<tr>
<td>Uterus</td>
<td>74</td>
<td>24</td>
<td>233</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3773</strong></td>
<td><strong>695</strong></td>
<td><strong>8466</strong></td>
</tr>
</tbody>
</table>

Table 3.3: Results with data set consisting of TCGA and GTEX samples. The addition of normal samples results in a higher accuracy.

<table>
<thead>
<tr>
<th></th>
<th>CV</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>99.3%</td>
<td>99.2%</td>
</tr>
<tr>
<td>Simulated</td>
<td>93.4%</td>
<td>97.6%</td>
</tr>
</tbody>
</table>

Table 3.3: Results with data set consisting of TCGA and GTEX samples. The addition of normal samples results in a higher accuracy.

LoCUP model. The source code has been made available at https://github.com/dansondergaard/locup.

3.4 Conclusion and future work

While the results presented above are promising, further validation is necessary to determine how well the classifier performs on CUP samples. However, the difficulty of obtaining true CUP and primary samples makes direct validation
infeasible and therefore alternative approaches are needed. In the work above we used simulation experiments and metastases of known origin to estimate the performance of our classifier. We plan to extend this work further by including a much larger data set of metastases in our validation process. We have recently obtained such a data set, known as the MET500 cohort [85].

The presence of normal tissue cells in a sample is part of the LoCUP model based on the assumption that \( x = \alpha x_T + (1 - \alpha) x_N \), that is, any sample is a mixture of some fraction of tumor and normal tissue. However, samples may also contain other kinds of contamination such as immune cells. An obvious choice would be to extend the likelihood with a third 'centroid', essentially assuming that \( x = \alpha_1 x_T + \alpha_2 x_N + \alpha_3 x_I \) where \( \alpha_1 + \alpha_2 + \alpha_3 = 1 \) and \( x_I \) is a centroid representing immune cells.

The method proposed here is a derivation of the nearest-centroid method which may be described as a “nearest pair of centroids” method. While this approach seems to work well, it may also be beneficial to look into alternative approaches such as non-negative matrix factorization.

While not presented in this chapter, experiments show that the estimated value of \( \alpha \) is not reliable as a measure of tumor purity. Work by Yoshihara et al. has shown that reliably estimating tumor purity from gene expression data is indeed possible [112] and thus we believe that there is room for improvement of this aspect of the LoCUP model. It is also possible that performance can be improved by taking as input the estimates of cell type proportions obtained with the method presented by Yoshihara et al.

We also plan to investigate whether the addition of other data types such as somatic mutation data may increase the performance of the classifier by building an ensemble of classifiers of different data types.

### 3.5 Contributions

I contributed to all aspects of this work, including idea development, experimental setup, execution of experiments, analysis of results, implementation, evaluation and packaging of the classifier, and writing the papers [97, 98].
Part II

Publications and manuscripts
Chapter 4

PATBox: A Toolbox for Classification and Analysis of P-Type ATPases

Dan Søndergaard\textsuperscript{1,2} Christian N. S. Pedersen\textsuperscript{1}

This chapter is a reprint of:


with only typographical changes.

4.1 Abstract

P-Type ATPases are part of the regulatory system of the cell where they are responsible for transporting ions and lipids through the cell membrane. These pumps are found in all eukaryotes and their malfunction has been found to cause several severe diseases. Knowing which substrate is pumped by a certain P-Type ATPase is therefore vital. The P-Type ATPases can be divided into 11 subtypes based on their specificity, that is, the substrate that they pump. Determining

\textsuperscript{1}Bioinformatics Research Centre, Aarhus University, Aarhus, Denmark

\textsuperscript{2}Centre for Membrane Pumps in Cells and Disease, Aarhus University, Aarhus, Denmark
the subtype experimentally is time-consuming. Thus it is of great interest to be able to accurately predict the subtype based on the amino acid sequence only. We present an approach to P-Type ATPase sequence classification based on the $k$-nearest neighbors, similar to a homology search, and show that this method provides performs very well and, to the best of our knowledge, better than any existing method despite its simplicity. The classifier is made available as a web service at http://services.birc.au.dk/patbox/ which also provides access to a database of potential P-Type ATPases and their predicted subtypes.

4.2 Introduction

P-Type ATPases are a large group of transmembrane transporters which pump ions and lipids as part of the regulatory system of the cell. It has been found that the malfunction of some P-Type ATPases cause several severe diseases in humans such as dystonia parkinsonism and Wilson disease [14].

The first P-Type ATPase, the sodium-potassium pump, was discovered in the 1950s and since then over 500 P-Type ATPases have been sequenced, their specificity experimentally verified, and several structures determined [14]. Phylogenetic analysis has shown that the P-Type ATPases can be divided into 5 major and 11 minor subtypes (1A, 1B, 2A, 2B, 2C, 2D, 3A, 3B, 4, 5A and 5B) based on the substrate transported by the pump [8].

Experimentally determining the subtype of a P-Type ATPase is a slow and expensive process, but computational methods for predicting the subtype can aid this analysis significantly. After evaluating several methods for subtype prediction we found that the method presented in this paper provides surprisingly good results and in fact, to the best of our knowledge, performs better than all existing methods despite its simplicity.

4.3 Materials and Methods

We present a machine learning approach for accurately predicting the subtype of a P-Type ATPase from the amino acid sequence by applying the $k$-nearest neighbors ($k$-NN) method [27] to a curated dataset of 515 P-Type ATPase sequences annotated with experimentally verified subtypes. The dataset (S1 Dataset) has been gathered from [8, 73]. The sequences were obtained by mapping the accession identifiers to UniProtKB [65]. Sequences with invalid characters and duplicates were removed. This resulted in 515 sequences with known subtype.

The classifier is made available as a web service. Sequences in FASTA format can be uploaded and the results are available as a web page or can
be downloaded in comma-separated values (CSV) format. The web service also provides access to an automatically constructed database of all sequences from UniProtKB containing PROSITE motif D-K-T-G-T-[LIVM]-[TI] (PS00154) characteristic for P-Type ATPases annotated with their classification obtained by our $k$-NN method and a classifier based on the Sequence Learner (SeqL) method [50, 51], which is also made available through the web service. The latter method has previously been applied to P-Type ATPase classification in [80]. This database is thus a valuable resource for exploring P-Type ATPases.

The PATBox web service is implemented in Python using Flask as a web framework, Celery as a job queue, and SQLite as the database. The service is packaged using Docker for reproducability and maintainability.

**Method**

The prediction method presented here is based on the $k$-NN method. Given a labeled dataset with data points $(x_1, y_1), \ldots, (x_n, y_n)$ and a query with unknown label $x$, the $k$-NN method looks at the $k$ nearest neighbors to $x$ by applying some distance function $d$ to each data point in the labeled dataset. The label $y$ of $x$ is then decided by majority vote. The distance function used in our approach is that of a BLAST [17] search. Thus, for some sequence $x$ a search is performed via BLAST and the top $k$ results are then used to perform a majority vote. For $k = 1$ this corresponds to a homology search on the curated dataset. Formulating the method in terms of nearest neighbor classification enables us to evaluate it using well-known machine learning evaluation methods.

Additionally we have implemented weighed majority vote such that the weight of a class is given by the sum of the E-values of results belonging to that class divided by the number of results belonging to that class. The class with the minimum weight is chosen as the predicted subtype.

**4.4 Results**

The overall performance of the $k$-NN classifier has been evaluated by non-stratified 5-fold cross-validation. The dataset is shuffled and split into five parts. A fold is then carried out by training on four parts and predicting on the remaining part. This is repeated five times. We denote this as a run. To obtain an estimate of the variance of the accuracy the run is repeated 20 times, shuffling the sequences every time, for a total of 100 parts per $k$ and the standard deviation and average accuracy is reported.

We evaluated both unweighed and weighed $k$-NN for $1 \leq k \leq 50$ to determine the best $k$ for each method.
Figure 4.1: The results of 20 runs of 5-fold cross-validation for $1 \leq k \leq 50$. The weighed and unweighed approaches both perform well for small $k$. For $k = 1$ we obtain an accuracy of 100%. Dots are outliers. Lines show accuracy for reduced datasets.

The results are summarized in Fig. 4.1 as a box plot. The average accuracy of the shuffled and repeated folds for each majority vote method is shown on the vertical axis with error bars showing the standard deviation. As we expect the accuracy of the two methods for $k = 1$ is the same. For both weighed and unweighed majority vote we see that as $k$ increases the accuracy decreases and the standard deviation increases. We obtain the best result when $k = 1$ for which the accuracy is 100%. Similar results are obtained for 2-fold cross-validation, where only half of the data is available for training, suggesting that the classifier is not prone to over-fitting (data not shown).

The high accuracy is not a surprise. The average area-under-curve (AUC) over all classes of the Structured Logistic Regression (SLR) classifier in [80] is 97.7%. An advanced prediction method presented in [34] based on neural networks also yields a very high accuracy of 99.1% based on a 10-fold cross-validation on 5/6 of the dataset.
The consistently good results obtained through a variety of independent methods also suggests that the methods are not over-fitting and should generalize well.

To further investigate the predictive power of the \( k \)-NN method we used the CD-HIT [49] web server to cluster the dataset at similarity thresholds of 30\%, 50\%, 75\% and 90\%, and extracted the representative sequences of each cluster. The cross-validation was repeated with weighed \( k \)-NN on the four reduced datasets and the results are shown as lines in Fig. 4.1 (error bars omitted to reduce complexity of the plot). We find that the method is very robust, obtaining 100\% accuracy for \( k = 1 \) for similarity thresholds as low as 50\%.

### 4.5 Discussion

We present a method for accurate classification of P-Type ATPases into 11 subtypes based on the \( k \)-NN method using BLAST as a distance measure. We show experimentally that the optimal \( k = 1 \) for which we obtain an accuracy of 100\%. More advanced methods have previously provided similar results which leads us to believe that the representative sequences for each subtype in the dataset cluster well based on sequence similarity. The results obtained by the \( k \)-NN method confirms this observation.

The contribution of this paper is twofold. Firstly, we show that \( k \)-NN performs extremely well on P-Type ATPases, despite the simplicity of the method, and that homology searches therefore can be used to determine the subtype of P-Type ATPase sequence. Secondly, the method presented here performs better than a multitude of more complicated methods, emphasising that simple methods should not be forgotten, even in the presence of more complicated methods.

The classifier is made available through a new web service for researchers in the field of P-Type ATPases, the P-Type ATPase Toolbox (PATBox), which also gives access to a database of predicted P-Type ATPases and their predicted subtype, based on UniProtKB [65].

### 4.6 Acknowledgments

We wish to thank Michael Knudsen and the reviewers for comments and corrections to the initial manuscript.
4.7 Supporting Information

S1 Dataset

Dataset of annotated P-Type ATPase sequences. The dataset used for cross-validation and final training of the classifier described in this manuscript.
Chapter 5

Computational Classification of P-Type ATPases

Dan Søndergaard\textsuperscript{1}  Michael Knudsen\textsuperscript{1}
Christian N. S. Pedersen\textsuperscript{1}

This chapter is a reprint of:


with only typographical changes.

5.1 Summary

Analysis of sequence data is inevitable in modern molecular biology, and important information about e.g. proteins can be inferred efficiently using computational methods. Here, we explain how to use the information in freely available

\textsuperscript{1}Bioinformatics Research Centre (BiRC), Aarhus University, Denmark
databases together with computational methods for classification and motif detection to assess whether a protein sequence corresponds to a P-type ATPase (and if so, which subtype) or not.

5.2 Introduction

With the increase in new data obtained from genomics projects, the need for automated methods assisting the determination of structural and functional properties of proteins from amino acid sequences alone is becoming apparent. In this chapter we give an overview of some freely available databases that may be used to assess whether a protein sequence corresponds to a P-type ATPase (and if so, which subtype) or not. We also explain two different methods for classification and discovery of P-type ATPases and present a new web service that makes these methods easily accessible, which we believe will be a useful tool for practitioners in the P-type ATPase field.

5.3 Databases

In this section we present three popular protein databases: UniProt, PROSITE, and TCDB. Even though neither of them are limited to P-type ATPases in particular (TCDB comes closest by focusing exclusively on transport proteins), they are all of great value for assessing whether a given protein sequence corresponds to a P-type ATPase or not. We also present PATBox, a new web service designed specifically with the classification and discovery of P-type ATPases in mind.

The Universal Protein Resource (UniProt)

The UniProt database [25] contains information about proteins obtained from the scientific literature. The information is split into four core databases – UniProtKB/Swiss-Prot, UniProtKB/TrEMBL, UniParc, and UniRef – which we review below.

The UniProtKB/Swiss-Prot database contains manually curated information. The data for a protein may include information about e.g. function, catalytic activity, pathways involving the protein, and protein-protein interactions. The user may enter a protein sequence on the UniProt web page and BLAST it against the UniProtKB/Swiss-Prot database.

Since manual curation cannot keep up with the rapid arrival of new data, the UniProtKB/TrEMBL database provides an alternative to UniProtKB/Swiss-Prot by also including automatically annotated entries. This gives a much higher
coverage but with a higher degree of uncertainty. As of June 11th 2014, the Swiss-Prot (resp. TrEMBL) database contains entries for 545,536 (resp. 69,014,937) protein sequences, and of these approximately 500 (resp. 68,000) correspond to P-type ATPases.

The two UniProtKB databases contain many redundant sequences, which may not only slow down computations but also make interpretation of results challenging. The UniRef database is created with this problem in mind. It contains clusters of sequences from UniProtKB based on sequence similarity and provides three sets – UniRef100, UniRef90, and UniRef50 – where sequences are clustered at the 100, 90, and 50 percent identity levels, respectively. Using e.g. UniRef50 results in a 70% reduction in database size compared to UniProtKB. The user may enter a protein sequence on the UniRef web page, which then finds the best matching clusters and provides details about the individual cluster members and links to their entries in UniProtKB.

The last database is UniParc, which gathers protein sequences from various sources, which besides UniProtKB includes the Protein Data Bank (PDB), the Reference Sequence Collection (RefSeq), and many others. The UniParc web page provides a convenient entry point for a search across multiple databases.

The UniProtKB database is accessible from: http://www.uniprot.org/

PROSITE

The PROSITE database [92] provides two web servers, ScanProsite and ProfileScan, which are capable of identifying protein families and domains. Each method has its own strengths and weaknesses and we discuss both below.

ScanProsite identifies protein families and domains based on short sequence patterns, also known as motifs. Functional or structural similarities between distantly related proteins are not easily picked up by pairwise alignments, but short subsequences (e.g. corresponding to binding sites) may still be very well conserved. Patterns are determined from multiple alignments of structurally and functionally related proteins and domains. Typical patterns are around 10–20 amino acids long. Some positions are fixed, whereas others may vary. For example, a pattern could be A-x-G-[IV]-P which translates to “alanine, any amino acid, glycine, isoleucine or valine, proline”.

It is computationally efficient to find all occurrences of a set of patterns in any given amino acid sequence, and the latest version of ScanProsite looks for 1,308 patterns. The web server provides links to detailed documentation about all found patterns cross-linked with other databases.

ProfileScan differs from ScanProsite by taking entire sequences into account using position-dependent scoring matrices, also known as profiles, calculated from multiple alignments of functionally and structurally related sequences.
When comparing an amino acid sequence to a profile, ProfileScan will always return a score reflecting how well the sequence matches the profile. If sequences are so distantly related that no patterns are conserved (and ScanProsite returns a negative answer), ProfileScan may still be able to detect an overall sequence similarity reflecting a structural, and hence possibly also functional, relationship.

ScanProsite and ProfileScan complement each other. ScanProsite identifies specific regions of sequences and the accompanying documentation provides a valuable starting point for further manual inspection and validation. Even if ScanProsite finds no patterns, distant evolutionary relationships may still be detected by ProfileScan, which may help narrow down the field of possible candidates for further analyses.

The PROSITE database is accessible from: http://prosite.expasy.org/

The Transport Classification Database (TCDB)

The Transporter Classification Database (TCDB) [87] is a manually curated database containing information about more than 10,000 transport proteins obtained from the scientific literature.

Proteins in TCDB are classified based on both functional and phylogenetic information, and each protein is assigned to a group identified by a name on the form N1.L1.N2.N3.N4, where N1 is the class, L1 the subclass, N2 the family, N3 the subfamily, and N4 the transport system.

As of July 3rd 2014, TCDB contains 214 P-type ATPase sequences. P-type ATPases are characterized by having N1.L1.N2 = 3.A.3 and are divided into 32 subfamilies (N3) which are further subdivided based on transport system (N4).

The TCDB web page allows users to enter protein sequences and BLAST them against the database to find the best matching classifications. The results page links to relevant literature and detailed information about the matching groups.

The TCDB database is accessible from: http://www.tcdb.org/

P-type ATPase Toolbox

The P-type ATPase Toolbox (PATBox) is a newly developed web service, which uses two computational tools (see Methods section) for discovery and classification of P-type ATPases. PATBox provides much of the functionality also available in a tool described in [80] and available at http://pumpkin.au.dk. However, it is based upon a more recent dataset and classification methods.

Additionally, the user can browse all sequences from the UniProtKB database containing the D-K-T-G-T-[LIVM]-[TI] motif which is conserved in all P-type
ATPases. Each sequence is also annotated with its true subtype, if known, and the subtype predicted by the computational methods.

The web service also provides download access to the manually curated dataset used to build the classifiers and the database. The dataset contains 524 P-type ATPase sequences and their associated subtype collected from [8, 73] and the previously described databases.

The P-type ATPase Toolbox is accessible from: http://services.birc.au.dk/patbox

5.4 Methods

In this section we provide step-by-step instructions for how to use PROSITE (described in the Databases section) to determine whether a protein sequence corresponds to a P-type ATPase or not.

We then present a method for predicting the subtype of a P-type ATPase based on BLAST searches. Finally, we describe a method for finding motifs, which may be of biological importance using a novel pattern discovery and classification technique introduced in [51].

ScanPROSITE – P-type ATPase or not?

The sequence motif D-K-T-G-T-[LIVM]-[TI] occurs in most P-type ATPases but only rarely in other proteins [36, 8], and hence it serves as a good indicator for whether a sequence corresponds to a P-type ATPase or not. Here are step-by-step instructions for how to scan sequences for this motif using ScanPROSITE:

1. Open a browser and navigate to http://prosite.expasy.org/scanprosite/
2. Make sure that Option 3 (Submit PROTEIN sequences and MOTIFS to scan them against each other) is checked.
3. In the box labeled “STEP 1”, enter the sequences you wish to identify as P-type ATPases or not.
4. In the box labeled “STEP 2”, enter the following PROSITE accession identifier: PS00154.
5. Click the button “START THE SCAN”.

When the results are ready, scroll down to see the graphical representation of the results (see Fig. 5.1).
Figure 5.1: The green bar shows the number of sequences, which contains the motif. Each match is then shown and the position in the sequence at which the motif was located is shown graphically. In this example, six sequences were searched for the D-K-T-G-T-[LIVM]-[TI] motif, and all six of them contained the pattern (referred to as hits).

Note that PROSITE reports nine sequences that, despite containing the D-K-T-G-T-[LIVM]-[TI] motif, are not P-type ATPases. Thus, the motif is a good indicator that a sequence is a P-type ATPase, but care should be taken when evaluating the results.

BLAST+ Method – What kind of P-type ATPase?

BLAST+ [17] is an efficient similarity search method for sequence data and a well-known method for searching large databases of sequences such as UniProt. Here we use a local version of BLAST+ with a custom database.

This method is available as a service at The P-type ATPase Toolbox, but for analysis on one's own machine or cluster, follow the instructions below.

This method requires some technical knowledge and experience with use of the command line. When classifying sequences with this approach your query sequence(s) must be saved to a FASTA formatted file.

To use the BLAST+ method you must download and install the BLAST+ software package, which is available for all major platforms from:
You must download the file, which is compatible with your platform. If in doubt whether your platform is 32- or 64-bit, use the 32-bit version.

2. Windows (64-bit): ncbi-blast-X.Y.Z+-win64.exe
3. Mac OS X: ncbi-blast-X.Y.Z+.dmg
4. Linux: Refer to your Linux distribution’s package manager or manual to find out which file to download.

When the download is complete, you must follow the installation instructions for your platform in the “Installation” section at:

http://www.ncbi.nlm.nih.gov/books/NBK1762/

When the BLAST+ approach is used, you search against a database of labeled sequences (in this particular case, the label is the subtype of the P-type ATPase). The query sequence is then believed to be of the same class as the labeled sequence with the best score returned by a search against the database.

A database built from our curated data set is available at The P-type ATPase Toolbox and consists of three files which must be placed in your working directory.

To query the database, one must open a command prompt or terminal and navigate to the directory containing the database files. This directory should also contain the FASTA file containing the sequences one wishes to classify. One must then run the command:

```bash
blastp -db ptype-atpase-blast.db -query input.fa -out ptype-atpase-results.txt
```

This creates a file called ptype-atpase-results.txt, which can be opened in a text editor (e.g. Word or Notepad on Windows or TextEdit on Mac). In the output file, locate the line:

```
Sequences producing significant alignments
```

This line signifies the beginning of a listing of all sequences in the database, which were found by BLAST+, sorted by their similarity with the query sequence. Example output is shown below.
Score E Sequences producing significant alignments: (Bits) Value

D2A4B7|5B|D2A4B7_TRICA Putative uncharacterized protein GLEAN_1... 2693 0.0
Q16XE5|5B|Q16XE5_AEAE AAE0008902-PA [Aedes aegypti ] 1239 0.0
Q7QH5N|5B|Q7QH5N_ANOGA AGAP011271-PA (Fragment) [Anopheles gam... 1216 0.0
Q7KQN3|5B|Q7KQN3_DROME CG32000, isoform G [Drosophila melanog... 1181 0.0
H9KBL9|5B|H9KBL9_APIME Uncharacterized protein [Apis mellifera ] 1082 0.0
Q9H7F8|5B|AT133_HUMAN Probable cation-transporting ATPase 13A3 ... 901 0.0
H2QP02|5B|H2QP02_PANTR ATPase type 13A3 [Pan troglodytes ] 901 0.0
Q5XF89|5B|AT133_MOUSE Probable cation-transporting ATPase 13A3 ... 879 0.0
Q9CTG6|5B|AT132_MOUSE Probable cation-transporting ATPase 13A2 ... 803 0.0

Each result is presented on a separate line containing the sequence identifier, class, and descriptive name of the sequence. Additionally, a bit-score and an E-value are given. To evaluate the significance of a hit, we will look at the E-value, which represents the number of sequences in the database, which can be expected to be given as a result with the same bit-score by chance. That is, a lower E-value is better since the probability that a similar hit is returned by chance is lower.

For example, consider the first result in the output shown above. An E-value of 0.0 means that, for the given query sequence, we expect to see 0.0 sequences in the database with the bit-score 2,693 or higher, meaning that the result is a very significant match. In the output above, the shown results are all very significant and belong to the class 5B. Hence we believe that the query sequence also belongs to the 5B class.

The accuracy of this method depends greatly on the number of results considered. We have experimentally found that considering the first three results provides the highest accuracy. This is a special case (k = 1) of the k-nearest neighbors method from the field of machine learning, where k is the number of “neighbors” considered.

**Sequence Learner (SeqL) Method – Motif Detection**

Sequence Learner (SeqL) [51] is a computational method for classification of sequences. Given two groups of sequences, e.g. sequences that correspond to a particular P-type ATPase subtype and sequences that do not, SeqL computes a set of sequence motifs that separate the two groups. A new sequence of unknown type is then scanned for occurrences of the motifs from each group, and the subtype is predicted to be that of the best matching group.

Note that SeqL only compares two groups. In the case of the 11 P-type ATPase subtypes, we build 11 models (1A versus not 1A, 1B versus not 1B, and so on), and use the model assigning the highest score to the sequence for prediction.
The most substantial benefit of this method is that the model contains the motifs that divide the groups, which may provide insight into the biology of the sequence. We will focus on the motif detection aspect of the SeqL method, but note that it can also be used for classification. For example, PATBox also provides classification of P-type ATPases using SeqL.

SeqL is a command line tool meaning that the user must be comfortable using the command line of the operating system.

Preparing the Input File
Recall that the SeqL method finds motifs, which divide two groups of sequences well. This means that SeqL must be given sequences for each group and for each sequence, one must know which group the sequence belongs to. The input file must be a plain text file where each line contains the class (negative class: -1 or positive class: +1) and the sequence. An example of a SeqL input file is shown below.

```
+1 ACCCGT
+1 CCCGTA
-1 ACACAC
-1 ACATAC
```

Of course, the input file can contain sequences of arbitrary length. Also, an arbitrary number of sequences may be specified for each group.

Installation and Usage
When the input is properly formatted, one must download and install the SeqL classifier. The most recent version of the SeqL source code is available at:

https://github.com/heerme/seql-sequence-learner/archive/master.zip

Since SeqL is distributed as source code, one must compile (translate the source code to machine code) the program before using it. We shall not detail this process as it depends on the operating system and which tools are available.

When SeqL has been compiled and the input file has been prepared and saved as e.g. example.txt, one must first build a preliminary model as follows:

```
./seql_learn -r 0 example.txt example.model
```

This creates a file containing the preliminary model called example.model. To be able to classify sequences using this model, one must first create an optimized model as follows:
./seql_mkmodel -i example.model -o example.model.bin -O example.model.predictors

This creates two additional files:

1. example.model.bin, which can be used for classification of new sequences,
2. example.model.predictors, which contains the weighted motifs which divide the two groups and may be interesting in a biological context.

In a motif detection context the predictors file is therefore the most interesting file to look at. The file can be opened in a text editor. A shortened version of the contents of the predictors file for this example is shown below.

```
-0.0372289452055364206151111
0.110800089590139996809093 CG
0.0875246051909023825565725 G
0.0857454245816138133085005 CCG
0.0856232047335227092199617 GT
0.071988986889090150412159 CGT
0.0719002330870201350476734 CCC
...

-0.0122787123594572476076303 ATAC
-0.0150784793067655267101435 ACAT
-0.0150784793067655267101435 CACAC
-0.0153275287278630562709525 AT
-0.0153275287278630562709525 CAC
-0.0250578076168992865124796 ACACA
...
```

The first line contains the bias of the model and is not relevant in this context. For each remaining line a motif is shown with its associated score. The score signifies how well the motif divides the two groups. In this example, CG is an important motif and indeed it only occurs in the positive class of the example data set. Equivalently, CACAC occurs only in the negative class and therefore has a good negative score.

### 5.5 Acknowledgments

We wish to thank Marco Palos Franco, Bioinformatics Research Centre (BiRC), Aarhus University, for collecting and curating the P-type ATPase dataset.
Chapter 6

Computational discovery of specificity-conferring sites in non-ribosomal peptide synthetases

Michael Knudsen\(^1\) Dan Søndergaard\(^1\) Claus Olesen\(^2\)
Ditlev Egeskov Brodersen\(^2\) Christian N. S. Pedersen\(^1\)

This chapter is a reprint of:


with only typographical changes.

6.1 Abstract

**Motivation:** By using a class of large modular enzymes known as Non-Ribosomal Peptide Synthetases (NRPS), bacteria and fungi are capable of synthesizing a

\(^1\)Bioinformatics Research Centre, Aarhus University, Denmark
\(^2\)Department of Molecular Biology and Genetics, Aarhus University, Aarhus, Denmark
large variety of secondary metabolites, many of which are bioactive and have potential, pharmaceutical applications as e.g. antibiotics. There is thus an interest in predicting the compound synthesized by an NRPS from its primary structure (amino acid sequence) alone, as this would enable an in silico search of whole genomes for NRPS enzymes capable of synthesizing potentially useful compounds.

**Results:** NRPS synthesis happens in a conveyor belt like fashion where each individual NRPS module is responsible for incorporating a specific substrate (typically an amino acid) into the final product. Here, we present a new method for predicting substrate specificities of individual NRPS modules based on occurrences of motifs in their primary structures. We compare our classifier to existing methods and discuss possible biological explanations of how the motifs might relate to substrate specificity.

**Availability:** SEQL-NRPS is available as a web service implemented in Python with Flask at [http://services.birc.au.dk/seql-nrps](http://services.birc.au.dk/seql-nrps) and source code available at [https://bitbucket.org/dansondergaard/seql-nrps/](https://bitbucket.org/dansondergaard/seql-nrps/).

**Contact:** micknudsen@gmail.com or cstorm@birc.au.dk

### 6.2 Introduction

Non-Ribosomal Peptide Synthetases (NRPS) are large proteins found mainly in bacteria and fungi where they are responsible for synthesis of a variety of peptides, many of which are bioactive and have significant pharmaceutical value [38]. While the most famous example is the penicillin precursor tripeptide (synthesized by fungi in the *Penicillium* genus) many other compounds are active as antibiotics or e.g. immunosuppressants are known. In a time where antibiotic resistance is becoming increasingly problematic, investigating the possibilities for synthesis of novel therapeutical compounds from NRPSs is therefore of great interest.

NRPSs are modular enzymes with each module comprising at least three domains: An adenylation (A) domain, a thiolation (T) domain, and a condensation (C) domain [38]. Of these, the A domain recruits a specific substrate and activates it by transfer of an adenylate moiety. The activated substrate is then picked up by a prosthetic phosphopantetheine arm coupled to the T domain and physically moved to the C domain where the peptide bond is formed to an activated substrate from the next module in line. This process continues up until the last C domain (which in bacteria this usually is a thioesterase domain) where
a hydrolysis reaction causes the chain of substrates to detach from the NRPS, often by circularization of the final product. One module is thus responsible for incorporating one substrate into the final peptide product.

More than 500 individual substrates have been observed in peptides synthesized by NRPSs, including both the 20 proteinogenic L-amino acids as well as their D-isomers, non-proteinogenic amino acids (such as ornithine), and fatty acids [15]). Determining the substrate specificity of a given A domain biochemically by isolating it and trying all possible candidates is a cumbersome process and often fruitless effort. Hence there is a desire to predict the specificity from the amino acid sequences of A domains alone, and state-of-the-art prediction methods such as NRPSsp [82] and NRPSpredictor2 ([86] are able to achieve that with relatively high accuracy using profile Hidden Markov Models (pHMM) and Support Vector Machines (SVM), respectively.

The first successful attempt to predict A domain substrate specificity was devised by examining the crystal structure of the A domain PheA from GrsA in Bacillus brevis in complex with its cognate substrate phenylalanine [100]. The analysis revealed ten residues critically involved in substrate binding and recognition. Due to high sequence similarity between A domains in general, it was speculated that sequence alignment of PheA with other A domains would reveal the residues important for substrate recognition. Using this approach, the authors successfully predicted the specificity of most A domains in their data set using only these ten residues, which were then aptly named the specificity-conferring code.

Here we present a new approach for predicting the specificity of NRPS A domains using automatic identification of sequence motifs discriminating sequences of A domains with different specificities. Evaluated on a manually curated, non-redundant data set, our method achieves an overall prediction accuracy of 71.3%, significantly higher than the 66.3% obtained by the current state-of-the-art method. The motifs discovered include the well-known specificity-conferring code but also other sites not previously associated with substrate specificity. We then discuss the significance of these sites from a biological and structural perspective.

### 6.3 Methods

#### Data

The data used for classification consists of amino acid sequences of A domains annotated with their corresponding substrate specificities. The largest data set available was collected for training the NRPSsp [82]) classifier and contains 1578
sequences. However, as recently pointed out by [54], this data set contains many (near) duplicate sequences, incorrectly annotated A domains, and even sequences not related to NRPSs. As a consequence, the authors manually curated a new data set from public databases and merged it with data from NRPSpredictor2 [86]. After removal of (near) duplicate sequences, this data set, which was also used for the work described here, comprises 537 sequences corresponding to a total of 37 different substrate specificities. The dataset contains both bacterial and fungal A-domain sequences, and we do not distinguish between these in training or testing of the method.

Specificity-conferring code

Using Clustal Omega [91], we generated a multiple alignment of the PheA sequence and all the sequences in the data set, and from this extracted the positions corresponding to the specificity-conferring code in PheA as described in [100].

Sequence Learner

We use Sequence Learner (SEQL) [51], an algorithm which, given two sets of sequences, identifies the sequence motifs that best discriminate between sequences from each set. In this case, we wanted to distinguish sequences of A domains binding a specific substrate $S$ from those that do not. The training set thus consists of pairs $\{x_i, y_i\}_{i=1}^N$, where $x_i$ is an A domain sequence and $y_i \in \{-1, +1\}$ indicates whether or not the corresponding A domain binds $S$.

The feature space (the variables on which classification is based) of SEQL is the set of all subsequences of sequences present in the training set. Let $d$ denote the size of the feature space. Each sequence $x_i$ can then be represented as a binary vector $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})^T$, where $x_{i,j} \in \{0, 1\}$ indicates whether or not the $j$-th feature is present in the sequence. Using a gradient descent algorithm, SEQL then finds parameters $\beta = (\beta_1, \beta_2, \ldots, \beta_d)$ that minimize a given loss function. Here we use a regularized squared-hinge loss function,

$$L(\beta) = \sum_{i=1}^N \max(1 - y_i\beta^T x_i, 0)^2 + CR_\alpha(\beta),$$

where $C$ is a constant, and where

$$R_\alpha(\beta) = \alpha \sum_{j=1}^d |\beta_j| + (1 - \alpha) \frac{1}{2} \sum_{j=1}^d \beta_j^2$$
is a regularizer penalizing large coordinates of $\beta$ to prevent over-fitting. In our analysis, we set $C = 0.7$ and $\alpha = 0.9$. In each step, only the coordinate corresponding to the maximal gradient magnitude is updated. Even though the feature space is enormous, the search can be done efficiently by pruning the search tree: This relies on the fact that the frequency of a subsequence is bounded by the frequency of its prefixes, and hence large groups of sequences can be discarded because their prefixes are less frequent than the most frequent subsequence found so far.

It was proved in [51] that the method is guaranteed to converge. In practice, however, the algorithm is terminated when the improvement in one step is below a certain threshold. The optimization is always started from $\beta = (0, 0, \ldots, 0)^T$, and when the process terminates, the non-zero elements in $\beta$ are interpreted as weights of significant features. A positive (resp. negative) value of $\beta_i$ corresponds to the $i$-th feature being overrepresented among the positive (resp. negative) examples in the training set. When applied to A domain sequences, the SEQL method typically finds fewer than 20 significant motifs.

### 6.4 Results and discussion

To locate sequence motifs with a significant influence on A domain specificity, we used the data set collected by [54], which had been manually curated and contains no duplicate sequences (see Methods. For each substrate specificity present in the set, we built a SEQL model which discriminates between that specificity and all other specificities, and given an unknown A domain, we then apply all these models and determine the specificity to be the one corresponding to the highest scoring model. This way of constructing a multi-class classifier from a binary classifier is known as the one-versus-rest schema.

To evaluate the performance of SEQL, we conducted a leave-one-out (LOO) cross-validation: For each sequence $S$ in the data set, we built models based on all sequences except $S$, and we then predict the specificity of $S$ based on this new set of models and ask whether the highest scoring model is the one corresponding to the substrate specificity of $S$. Note that since the data contains no duplicate sequences, the sequences for which we are trying to predict the specificity are never present in the corresponding training set, and hence we do not risk achieving an artificially high performance by predicting specificities already contained within the training set.
Comparison with existing methods

When evaluating SEQL on the data set from [54] using LOO an overall accuracy of 71.3% was obtained, which is higher than the 66.3% achieved by ([54] using an ensemble of pHMMs.

While the NRPSsp predictor [82] is available as a web service, there is no option to train it on a different data set. We are therefore unable to assess the performance of NRPSsp on the data set used here. However, we can perform a LOO analysis using SEQL on the NRPSsp data set, and we obtain an accuracy of 83.5%, which is comparable to the 86.4% reported by [82]. Similarly, we are not able to train NRPSpredictor2 on the data set from [54]. Note also that NRPSpredictor2 is a semi-supervised predictor: Besides training on labeled data (576 sequences of A domains with known specificities), it also takes unlabeled data into account (5096 sequences of A domains with unknown specificities). A comparison with the labeled sequences from NRPSpredictor2 is made available in Supplementary Section S1.

Single substrates

The accuracy of prediction for single substrates obtained using SEQL are shown in Table 6.1. Generally, the method performs well on single substrates, but the prediction accuracy for phenylalanine (PHE) is noticeably lower than for all other substrates. This is also the case using ensemble pHMMs (see the LOO column in Table 4 in [54]), and phenylalanine also ranks among the substrates most poorly classified by NRPSpredictor2 (see Table 1 in [86]). Accuracies for individual substrates are not reported for NRPSsp, so we are not able to assess whether phenylalanine also poses a problem for this classifier (further discussion in Supplementary Section S2).

Biological significance of motifs

To investigate whether the sequence motifs discovered by SEQL have any biological significance, we identified all occurrences of motifs in sequences with specificity S and mapped those, via the multiple alignment, to the corresponding positions in the PheA sequence. Thus, for each position in the PheA sequence, we obtain the frequency of how many times that site is part of a motif in a sequence of specificity S.

Figure 6.2 summarizes the results for hydrophobic aliphatic, hydrophobic aromatic, and hydrophilic substrates in the data set (this grouping of substrates was also used in the evaluation of NRPSpredictor2 in [86]). The SEQL method clearly ranks positions around the specificity-conferring code as important,
Table 6.1: Performance of the SEQL method on individual substrates in the data set from [54]. Only substrates occurring at least ten times in the data set are shown.

<table>
<thead>
<tr>
<th>Substrate</th>
<th>Count</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ala</td>
<td>46</td>
<td>0.48</td>
<td>0.65</td>
<td>0.56</td>
</tr>
<tr>
<td>Leu</td>
<td>41</td>
<td>0.60</td>
<td>0.78</td>
<td>0.68</td>
</tr>
<tr>
<td>Thr</td>
<td>34</td>
<td>0.91</td>
<td>0.94</td>
<td>0.93</td>
</tr>
<tr>
<td>Val</td>
<td>34</td>
<td>0.56</td>
<td>0.74</td>
<td>0.63</td>
</tr>
<tr>
<td>Ser</td>
<td>33</td>
<td>0.77</td>
<td>0.91</td>
<td>0.83</td>
</tr>
<tr>
<td>Gly</td>
<td>30</td>
<td>0.77</td>
<td>0.77</td>
<td>0.77</td>
</tr>
<tr>
<td>Cys</td>
<td>27</td>
<td>0.74</td>
<td>0.85</td>
<td>0.79</td>
</tr>
<tr>
<td>Hpg-HpgCl</td>
<td>21</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Asn</td>
<td>20</td>
<td>0.91</td>
<td>1.00</td>
<td>0.95</td>
</tr>
<tr>
<td>Pro</td>
<td>20</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>Tyr</td>
<td>18</td>
<td>0.55</td>
<td>0.67</td>
<td>0.60</td>
</tr>
<tr>
<td>Abu-Iva</td>
<td>17</td>
<td>0.79</td>
<td>0.65</td>
<td>0.71</td>
</tr>
<tr>
<td>Glu</td>
<td>16</td>
<td>0.67</td>
<td>0.50</td>
<td>0.57</td>
</tr>
<tr>
<td>Asp</td>
<td>15</td>
<td>1.00</td>
<td>0.73</td>
<td>0.85</td>
</tr>
<tr>
<td>Phe</td>
<td>15</td>
<td>0.31</td>
<td>0.27</td>
<td>0.29</td>
</tr>
<tr>
<td>Trp</td>
<td>14</td>
<td>0.64</td>
<td>0.50</td>
<td>0.56</td>
</tr>
<tr>
<td>Ile</td>
<td>13</td>
<td>0.92</td>
<td>0.85</td>
<td>0.88</td>
</tr>
<tr>
<td>Dhb-Sal</td>
<td>12</td>
<td>0.92</td>
<td>0.92</td>
<td>0.92</td>
</tr>
<tr>
<td>Orn</td>
<td>12</td>
<td>0.56</td>
<td>0.42</td>
<td>0.48</td>
</tr>
<tr>
<td>Aad</td>
<td>10</td>
<td>1.00</td>
<td>0.70</td>
<td>0.82</td>
</tr>
<tr>
<td>Dab</td>
<td>10</td>
<td>1.00</td>
<td>0.80</td>
<td>0.89</td>
</tr>
<tr>
<td>Gin</td>
<td>10</td>
<td>0.75</td>
<td>0.60</td>
<td>0.67</td>
</tr>
</tbody>
</table>

but there are also noticeable peaks around residues at positions 110, 380, and 440 (numbers correspond to the PheA reference sequence). These regions are not considered part of the specificity-conferring code, instead they correspond to three out of ten conserved core sequences as defined in [66], namely core sequences A2, A6, and A8, respectively (see Figure 6.1).

The A2 core sequence is located in a central, four-stranded parallel β-sheet in the N-terminal region of the A domain and constrains torsion angles on the conserved glycine in the sequence. The motif is believed to have purely structural significance, as it is found far away from the active site and is thus not directly involved in substrate recognition. However, as this study shows, the A2 motif may still influence the substrate specificity in an indirect way (see Figure 6.2).

The A6 core sequence is located close to the active site and forms part of a distorted β-bundle in the A domain, which is believed to be involved in the adenylation reaction of A domains [79]. Photochemical labeling of 2-azido-ATP has shown that the A6 motif is involved in binding adenine or the ribose of ATP. Within the A6 motif, a highly conserved tyrosine may stack with the adenine base of ATP and the role of A6 in specificity determination may therefore relate to
Figure 6.1: Cartoon overview of PheA with the active site cleft indicated by a surface representation. Side chains of the residues constituting the specificity-conferring code and the substrate Phe are shown as stick representations. The core sequences A2, A6 and A8 are shown in blue, orange, and pink, respectively.

Figure 6.2: Relative frequencies of how often sites in PheA correspond to sites that are part of motifs in sequences among hydrophobic aliphatic, hydrophobic aromatic, and hydrophilic substrates. The black curve is the average relative frequency. Locations of the core sequences A2, A6, and A8 are shown using wide bars. The specificity-conferring sites (235, 236, 239, 278, 299, 301, 322, 330, 331) are indicated using gray bars.
the size of the amino acid substrate indirectly affecting positioning the tyrosine in relation to ATP.

Finally, the A8 core sequence is located in the hinge loop connecting the N-terminal and the C-terminal subdomains of the A domain. The motif is involved in both adenylation and thioester reaction and includes a completely conserved arginine residue that interacts with the 2' and 3' ribose hydroxyl groups of the adenylate intermediate. Mutagenesis has shown that a single mutation of the conserved glycine residue results in the inability to activate the substrate [103]). The glycine residue is located in a turn between two antiparallel β-strands and this turn is of particular importance for the adenylation reaction. The A8 glycine is necessary for maintaining the turn structure between the two antiparallel β-strands. However, the exact role of the glycine residue remains unclear ([103, 79].

The A domain adopts two major conformations during the reaction cycle: One conformation catalyzes the adenylation half-reaction while a 140° rotation orients the C-terminal subdomain of the A domain in the correct position for the second half-reaction, thioester formation [31]. Thus, alternate faces of the C-terminal domain are exposed to the same active site for the two half-reactions, a phenomenon referred to as domain alternation [9]. The rotation of the C-terminal subdomain moves the A8 loop into the active site responsible for the adenylation half-reaction and the conserved glycine of A8 ~30Å. In this new position, A8 can interact with the adenylate intermediate formed by displacement of pyrophosphate. Furthermore, mutagenesis has shown that residue composition of the A8 core sequence is important for the thioester reaction ([84]. Mutating the conserved glycine to leucine disrupts only the second half-reaction, which leads to the conclusion that the bulky side chain of leucine causes steric interference [84, 31]. The role of A8 in determining substrate specificity is therefore most likely related to positioning of the ATP and phosphopantetheine in the adenylation reaction and thioester reactions. Since neither A2, A6, nor A8 interacts directly with the amino acid substrate, we propose that the overall residue composition of these conserved core sequences influence substrate specificity indirectly.

### 6.5 Web service

Our classifier is available as a web service where users can paste or upload A domains in FASTA format (http://services.birc.au.dk/seql-nrps). The sequences are then classified and the results page shows the sequence identifier, the predicted substrate specificity, and the probability given by SEQL that the sequence belongs to the predicted substrate. This probability is colored in
accordance with the confidence of the prediction, which is computed as the number of standard deviations that the highest probability is over the mean of the probabilities given by each substrate model.

The sequence is shown with motifs which discriminate for the predicted substrate colored green, and motifs discriminating against the predicted substrate colored red.

The web service, code for running the experiments and instructions for usage is MIT licensed and available at https://bitbucket.org/dansondergaard/seql-nrps/.

6.6 Conclusion

We have presented a new tool for predicting substrate specificities of NRPS adenylation domains. Besides achieving a high accuracy, the predictor differs from existing black box methods in that it identifies sequence motifs significantly distinguishing between different substrate specificities. These motifs may be used as starting points for further investigations of the biological nature of specificity determination. Indeed, we identify three regions of which two have previously been suggested to play a role in substrate specificity determination. The predictor is available online as a free-to-use web service (http://services.birc.au.dk/seql-nrps).

Acknowledgement

Funding: Aarhus University Research Foundation (the NANORIPES project) and Danish National Research Foundation (the PUMPkin center of excellence).
Chapter 7

HydDB: A web tool for hydrogenase classification and analysis

Dan Søndergaard¹ Christian N. S. Pedersen¹
Chris Greening² ³

This chapter is a reprint of:


with only typographical changes. Table 1 and 2 in the original paper have been left out due to space constraints.

¹Aarhus University, Bioinformatics Research Centre, C.F. Møllers Allé 8, Aarhus DK-8000, Denmark
²The Commonwealth Scientific and Industrial Research Organisation, Land and Water Flagship, Clunies Ross Street, Acton, ACT 2060, Australia
³Monash University, School of Biological Sciences, Clayton, VIC 2800, Australia

69
7.1 Abstract

$\text{H}_2$ metabolism is proposed to be the most ancient and diverse mechanism of energy-conservation. The metalloenzymes mediating this metabolism, hydrogenases, are encoded by over 60 microbial phyla and are present in all major ecosystems. We developed a classification system and web tool, HydDB, for the structural and functional analysis of these enzymes. We show that hydrogenase function can be predicted by primary sequence alone using an expanded classification scheme (comprising 29 [NiFe], 8 [FeFe], and 1 [Fe] hydrogenase classes) that defines 11 new classes with distinct biological functions. Using this scheme, we built a web tool that rapidly and reliably classifies hydrogenase primary sequences using a combination of k-nearest neighbors’ algorithms and CDD referencing. Demonstrating its capacity, the tool reliably predicted hydrogenase content and function in 12 newly-sequenced bacteria, archaea, and eukaryotes. HydDB provides the capacity to browse the amino acid sequences of 3248 annotated hydrogenase catalytic subunits and also contains a detailed repository of physiological, biochemical, and structural information about the 38 hydrogenase classes defined here. The database and classifier are freely and publicly available at http://services.birc.au.dk/hyddb/.

7.2 Introduction

Microorganisms conserve energy by metabolizing $\text{H}_2$. Oxidation of this high-energy fuel yields electrons that can be used for respiration and carbon-fixation. This diffusible gas is also produced in diverse fermentation and anaerobic respiratory processes [1]. $\text{H}_2$ metabolism contributes to the growth and survival of microorganisms across the three domains of life: chemotrophs and phototrophs, lithotrophs and heterotrophs, aerobes and anaerobes, mesophiles and extremophiles alike [88, 47]. On the ecosystem scale, $\text{H}_2$ supports microbial communities in most terrestrial, aquatic, and host-associated ecosystems [88, 26]. It is also proposed that $\text{H}_2$ was the primordial electron donor [58, 109]. In biological systems, metalloenzymes known as hydrogenases are responsible for oxidizing and evolving $\text{H}_2$ [88, 64]. Our recent survey showed there is a far greater number and diversity of hydrogenases than previously thought [47]. It is predicted that over 55 microbial phyla and over a third of all microorganisms harbor hydrogenases [47]. Better understanding $\text{H}_2$ metabolism and the enzymes that mediate it also has wider implications, particularly in relation to human health and disease [26, 18], biogeochemical cycling [45], and renewable energy [60, 28].
There are three types of hydrogenase, the [NiFe], [FeFe], and [Fe] hydrogenases, that are distinguished by their metal composition. Whereas the [Fe]-hydrogenases are a small methanogenic-specific family [90], the [NiFe] and [FeFe] classes are widely distributed and functionally diverse. They can be classified through a hierarchical system into different groups and subgroups/subtypes with distinct biochemical features (e.g. directionality, affinity, redox partners, and localization) and physiological roles (i.e. respiration, fermentation, bifurcation, sensing) [88, 64]. It is necessary to define the subgroup or subtype of the hydrogenase to predict hydrogenase function. For example, while Group 2a and 2b [NiFe]-hydrogenases share > 35% sequence identity, they have distinct roles as respiratory uptake hydrogenases and H₂ sensors respectively [59, 44]. Likewise, discrimination between Group A1 and Group A3 [FeFe]-hydrogenases is necessary to distinguish fermentative and bifurcating enzymes [47]. Building on previous work [104, 16], we recently created a comprehensive hydrogenase classification scheme predictive of biological function [47]. This scheme was primarily based on the topology of phylogenetic trees built from the amino acid sequences of hydrogenase catalytic subunits/domains. It also factored in genetic organization, metal-binding motifs, and functional information. This analysis identified 22 subgroups (within four groups) of [NiFe]-hydrogenases and six subtypes (within three groups) of [FeFe]-hydrogenases, each proposed to have unique physiological roles and contexts [47].

In this work, we build on these findings to develop the first web database for the classification and analysis of hydrogenases. We developed an expanded classification scheme that captures the full sequence diversity of hydrogenase enzymes and predicts their biological function. Using this information, we developed a classification tool based on the k-nearest neighbors’ (k-NN) method. HydDB is a user-friendly, high-throughput, and functionally-predictive tool for hydrogenase classification that operates with precision exceeding 99.8%.

7.3 Results and Discussion

A sequence-based classification scheme for hydrogenases

We initially developed a classification scheme to enable prediction of hydrogenase function by primary sequence alone. To do this, we visualized the relationships between all hydrogenases in sequence similarity networks (SSN) [6], in which nodes represent individual proteins and the distances between them reflect BLAST E-values. As reflected by our analysis of other protein superfamilies [2, 76], SSNs allow robust inference of sequence-structure-function relationships for large datasets without the problems associated with phyloge-
netic trees (e.g. long-branch attraction). Consistent with previous phylogenetic analyses [47, 104, 16], this analysis showed the hydrogenase sequences clustered into eight major groups (Groups 1 to 4 [NiFe]-hydrogenases, Groups A to C [FeFe]-hydrogenases, [Fe]-hydrogenases), six of which separate into multiple functionally-distinct subgroups or subtypes at narrower logE filters (Figure 7.1; Figure S1). The SSNs demonstrated that all [NiFe]-hydrogenase subgroups defined through phylogenetic trees in our previous work [47] separated into distinct clusters, which is consistent with our evolutionary model that such hydrogenases diverged from a common ancestor to adopt multiple distinct functions [47]. The only exception were the Group A [FeFe]-hydrogenases, which as previously-reported [47, 16], cannot be classified by sequence alone as they have principally diversified through changes in domain architecture and quaternary structure. It remains necessary to analyze the organization of the genes encoding these enzymes to determine their specific function, e.g. whether they serve fermentative or electron-bifurcating roles.

The SSN analysis revealed that several branches that clustered together on the phylogenetic tree analysis [47] in fact separate into several well-resolved subclades (Figure 7.1). We determined whether this was significant by analyzing the taxonomic distribution, genetic organization, metal-binding sites, and reported biochemical or functional characteristics of the differentiated subclades. On this basis, we concluded that 11 of the new subclades identified are likely to have unique physiological roles. We therefore refine and expand the hydrogenase classification to reflect the hydrogenases are more diverse in both primary sequence and predicted function than accounted for by even the latest classification scheme [47]. The new scheme comprises 38 hydrogenase classes, namely 29 [NiFe]-hydrogenase subclasses, 8 [FeFe]-hydrogenase subtypes, and the monophyletic [Fe]-hydrogenases (Table 1).

Table 7.1: Expanded classification scheme for hydrogenase enzymes. The majority of the classes were defined in previous work 2,16,17,46. The [NiFe] Group 1i, 1j, 1k, 2e, 4d, 4g, 4h, and 4i enzymes and [FeFe] Groups C1, C2, and C3 enzymes were defined in this work based on their separation into distinct clusters in the SSN analysis (Figure 1). HydDB contains detailed information on each of these classes, including their taxonomic distribution, genetic organization, biochemistry, and structures, as well as a list of primary references.

Three lineages originally classified as Group 1a [NiFe]-hydrogenases were reclassified as new subgroups, namely those affiliated with Coriobacteria (Group 1i), Archaeoglobi (Group 1j), and Methanosarcinales (Group 1k). Cellular and molecular studies show these enzymes all support anaerobic respiration of H₂,
but differ in the membrane carriers (methanophenazine, menaquinone) and terminal electron acceptors (heterodisulfide, sulfate, nitrate) that they couple to [101, 30]. The previously-proposed 4b and 4d subgroups [47] were dissolved, as the SSN analysis confirmed they were highly polyphyletic. These sequences are reclassified here into five new subgroups: the formate- and carbon monoxide-respiring Mrp-linked complexes (Group 4b) [55], the ferredoxin-coupled Mrp-linked complexes (Group 4d) [70], the well-described methanogenic Eha (Group 4h) and Ehb (Group 4i) supercomplexes [61], and a more loosely clustered class of unknown function (Group 4g). Enzymes within these subgroups, with the exception of the uncharacterized 4g enzymes, sustain well-described specialist functions in the energetics of various archaea [55, 70, 61]. Three crenarchaeotal hydrogenases were also classified as their own family (Group 2e); these enzymes enable certain crenarchaeotes to grow aerobically on O$_2$ [7, 41] and hence may represent a unique lineage of aerobic uptake hydrogenases currently underrepresented in genome databases. The Group C [FeFe]-hydrogenases were also separated into three main subtypes given they separate into distinct clusters even at relatively broad logE values (Figure 7.1); these subtypes are each transcribed with different regulatory elements and are likely to have distinct regulatory roles [47, 16, 81] (Table 1).

**HydDB reliably predicts hydrogenase class using the $k$-NN method and CDD referencing**

Using this information, we built a web tool to classify hydrogenases. Hydrogenase classification is determined through a three-step process following input of the catalytic subunit sequence. Two checks are initially performed to confirm if the inputted sequence is likely to encode a hydrogenase catalytic subunit-domain. The Conserved Domain Database (CDD) 29 is referenced to confirm that the inputted sequence has a hydrogenase catalytic domain, i.e. “Complex1_49kDa superfamily” (cl21493) (for NiFe-hydrogenases), “Fe_hyd_lg_C superfamily” (cl14953) (for FeFe-hydrogenases), and “HMD” (pfam03201) (for Fe-hydrogenases). A homology check is also performed that computes the BLAST E-value between the inputted sequence and its closest homolog in HydDB. HydDB classifies any inputted sequence that lacks hydrogenase conserved domains or has low homology scores (E-value > $10^{-5}$) as a non-hydrogenase (Table S1).

In the final step, the sequence is classified through the $k$-NN method that determines the most similar sequences listed in the HydDB reference database. To determine the optimal $k$ for the dataset, we performed a 5-fold cross-validation for $k = 1 \ldots 10$ and computed the precision for each $k$. The results are shown in Figure 7.2. The classifier predicted the classes of the 3248 hydrogenase
sequences with 99.8% precision and high robustness when performing a 5-fold cross-validation (as described in the Methods section) for \( k = 4 \). The six sequences where there were discrepancies between the SSN and \( k \)-NN predictions are shown in Table S2. The classifier has also been trained to detect and exclude protein families that are homologous to hydrogenases but do not metabolize \( \text{H}_2 \) (Nuo, Ehr, NARF, HmdII [88, 47]) using reference sequences of these proteins (Table S1).

Sequences of the [FeFe] Group A can be classified into functionally-distinct subtypes (A1, A2, A3, A4) based on genetic organization [47]. The classifier can classify such hydrogenases if the protein sequence immediately downstream from the catalytic subunit sequence is provided. The classifier references the CDD to search for conserved domains in the downstream protein sequence. A sequence is classified as [FeFe] Group A2 if one of the domains “GltA”, “GltD”, “glutamate synthase small subunit” or “putative oxidoreductase”, but not “NuoF”, is found in the sequence. Sequences are classified as [FeFe] Group A3 if the domain “NuoF” is found and [FeFe] Group A4 if the domain “HycB” is present. If none of the domains are found, the sequence is classified as A1. These classification rules were determined by collecting 69 downstream protein sequences. The sequences were then submitted to the CDD and the domains which most often occurred in each subtype were extracted.

In addition to its precision, the classifier is superior to other approaches due to its usability. It is accessible as a free web service at \text{http://services.birc.au.dk/hyddb/}. HydDB allows the users to paste or upload sequences of hydrogenase catalytic subunit sequences in FASTA format and run the classification (Figure S2). When analysis has completed, results are presented in a table that can be downloaded as a CSV file (Figure S3). This provides an efficient and user-friendly way to classify hydrogenases, in contrast to the previous standard which requires visualization of phylogenetic trees derived from multiple sequence alignments [11].

**HydDB infers the physiological roles of \( \text{H}_2 \) metabolism**

As summarized in Table 1, hydrogenase class is strongly correlated with physiological role. As a result, the classifier is capable of predicting both the class and function of a sequenced hydrogenase. To demonstrate this capacity, we used HydDB to analyze the hydrogenases present in 12 newly-sequenced bacteria, archaea, and eukaryotes of major ecological significance. The classifier correctly classified all 24 hydrogenases identified in the sequenced genomes, as validated with SSNs (Table 2). On the basis of these classifications, the physiological roles of \( \text{H}_2 \) metabolism were predicted (Table 2). For five of the organisms, these predictions are confirmed or supported by previously published data [41, 46,
Table left out due to space constraints.

**Table 7.2:** HydDB accurately determined hydrogenase content and predicted the physiological roles of H2 metabolism in 12 newly-sequenced archaeal and bacterial species.

23, 57, 19]. Other predictions are in line with metabolic models derived from metagenome surveying [48, 35, 13]. In some cases, the capacity for organisms to metabolize H2 was not tested or inferred in previous studies despite the presence of hydrogenases in the sequenced genomes [23, 99, 33, 110].

While HydDB serves as a reliable initial predictor of hydrogenase class and function, further analysis is recommended to verify predictions. Hydrogenase sequences only provide organisms with the genetic capacity to metabolise H2; their function is ultimately modulated by their expression and integration within the cell [88, 43]. In addition, some classifications are likely to be overgeneralized due to lack of functional and biochemical characterisation of certain lineages and sublineages. For example, it is not clear if two distant members of the Group 1h [NiFe]-hydrogenases (Robiginitalea bifomata, Sulfolobus islandicus) perform the same H2-scavenging functions as the core group 9. Likewise, it seems probable that the Group 3a [NiFe]-hydrogenases of Thermococci and Aquificae use a distinct electron donor to the main class [42]. Prominent cautions are included in the enzyme pages in cases such as these. HydDB will be updated when literature is published that influences functional assignments.

**HydDB contains interfaces for hydrogenase browsing and analyzing**

In addition to its classification function, HydDB is designed to be a definitive repository for hydrogenase retrieval and analysis. The database presently contains entries for 3248 hydrogenases, including their NCBI accession numbers, amino acid sequences, hydrogenase classes, taxonomic affiliations, and predicted behavior (Figure S4). To enable easy exploration of the data set, the database also provides access to an interface for searching, filtering, and sorting the data, as well as the capacity to download the results in CSV or FASTA format. There are individual pages for the 38 hydrogenase classes defined here (Table 1), including descriptions of their physiological role, genetic organization, taxonomic distribution, and biochemical features. This is supplemented with a compendium of structural information about the hydrogenases, which is integrated with the Protein Databank (PDB), as well as a library of over 500 literature references (Figure S5).
7.4 Conclusions

To summarize, HydDB is a definitive resource for hydrogenase classification and analysis. The classifier described here provides a reliable, efficient, and convenient tool for hydrogenase classification and functional prediction. HydDB also provides browsing tools for the rapid analysis and retrieval of hydrogenase sequences. Finally, the manually-curated repository of class descriptions, hydrogenase structures, and literature references provides a deep but accessible resource for understanding hydrogenases.

7.5 Methods

Sequence datasets

The database was constructed using the amino acid sequences of all curated non-redundant 3248 hydrogenase catalytic subunits represented in the NCBI RefSeq database in August 2014 [47] (Dataset S1). In order to test the classification tool, additional sequences from newly-sequenced archaeal and bacterial phyla were retrieved from the Joint Genome Institute’s Integrated Microbial Genomes database.

Sequence similarity networks

Sequence similarity networks (SSNs) [6] constructed using Cytoscape 4.1 [89] were used to visualize the distribution and diversity of the retrieved hydrogenase sequences. In this analysis, each node represents one of the 3248 hydrogenase sequences in the reference database (Dataset S1). Each edge represents the sequence similarity between them as determined by E-values from all-vs-all BLAST analysis, with all self and duplicate edges removed. Three networks were constructed, namely for the [NiFe]-hydrogenase large subunit sequences (Dataset S2), [FeFe]-hydrogenase catalytic domain sequences (Dataset S3), and [Fe]-hydrogenase sequences (Dataset S4). To control the degree of separation between nodes, logE cutoffs that were incrementally decreased from -5 to -200 until no major changes in clustering was observed. The logE cutoffs used for the final classifications are shown in Figure 7.1 and Figure S1.

Classification method

The k-NN method is a well-known machine learning method for classification [27]. Given a set of data points \( x_1, x_2, \ldots, x_N \) (e.g. sequences) with known labels \( y_1, y_2, \ldots, y_N \) (e.g. type annotations), the label of a point, \( x \), is predicted
by computing the distance from $x$ to $x_1, x_2, \ldots, x_N$ and extracting the $k$ labeled points closest to $x$, i.e. the neighbors. The predicted label is then determined by majority vote of the labels of the neighbors. The distance measure applied here is that of a BLAST search. Thus, the classifier corresponds to a homology search where the types of the top $k$ results are considered. However, formulating the classification method as a machine learning problem allows the use of common evaluation methods to estimate the precision of the method and perform model selection. The classifier was evaluated using $k$-fold cross-validation. The dataset is first split into $k$ parts of equal size. $k - 1$ parts (the training set) are then used for training the classifier and the labels of the data points in the remaining part (the test set) are then predicted. This process, called a fold, is repeated $k$ times. The predicted labels of each fold are then compared to the known labels and a precision can be computed.

7.6 Acknowledgements

We thank A/Prof Colin J. Jackson, Dr Hafna Ahmed, Dr Andrew Warden, Dr Stephen Pearce, and the two anonymous reviewers for their helpful advice and comments regarding this manuscript. This work was supported by a PUMPkin Centre of Excellence PhD Scholarship awarded to DS and a CSIRO Office of the Chief Executive Postdoctoral Fellowship awarded to CG.

7.7 Author Contributions

CG and DS designed experiments. DS and CG performed experiments. CG, DS, and CNSP analyzed data. CNSP supervised students. CG and DS wrote the paper.

7.8 Competing financial interests

The authors declare no competing financial interests.
Figure 7.1: Nodes represent individual proteins and the edges show the BLAST E-values between them at the logE filter defined at the bottom-left of each panel. The sequences are colored by class as defined in the legends. Figure S1 shows the further delineation of the encircled [NiFe] hydrogenase classes.
Figure 7.2: For each $k$, a 5-fold cross-validation was performed. The mean precision ± two standard deviations of the folds is shown in the figure (note the $y$-axis). $k = 1$ provides the most accurate classifier. However, $k = 4$ provides almost the same precision and is more robust to errors in the training set (reflected by the lower standard deviation). In general, the standard deviation is very small, indicating that the predictions are robust to changes in the training data.
Chapter 8

Prediction of primary tumors in cancers of unknown primary

Dan Søndergaard1,* Svend V. Nielsen1
Christian Nørgaard Storm Pedersen1 Søren Besenbacher2

This chapter is a reprint of:


with only typographical changes.

8.1 Abstract

A cancer of unknown primary (CUP) is a metastatic cancer for which standard diagnostic tests fail to identify the location of the primary tumor. CUPs account for 3-5% of cancer cases. Using molecular data to determine the location of the

1Bioinformatics Research Centre (BiRC), Aarhus University, Denmark
2Department of Molecular Medicine (MOMA), Aarhus University Hospital, Denmark
primary tumor in such cases can help doctors make the right treatment choice and thus improve the clinical outcome. In this paper, we present a new method for predicting the location of the primary tumor using gene expression data: LoCUP (Locating Cancers of Unknown Primary). The method models the data as a mixture of normal and tumor cells and thus allows correct classification even in impure samples, where the tumor biopsy is contaminated by a large fraction of normal cells. We find that our method provides a significant increase in classification accuracy (95.8% over 90.8%) on simulated low-purity metastatic samples and shows potential on a small dataset of real metastasis samples with known origin.

**Keywords:** cancer of unknown origin; classification; transcriptomics; precision medicine

### 8.2 Introduction

Cancers are named based on their primary location (the type of tissue where the cancer originated). A lung cancer that has metastasized (spread) to the liver will for example still be defined as a lung cancer and not a liver cancer. In 3-5% of cancer cases doctors only find a metastasis but fail to locate the original tumor; these are called cancers of unknown primary (CUP) [105]. The standard treatment is different for different types of cancer and CUP cases are thus generally harder to treat and consequently have significantly worse prognosis compared to the average cancer patient. Finding the source of malignancy, i.e. the location of the primary tumor, is crucial for improving the treatment of CUP patients. Pathologic evaluation of a biopsy usually includes immunohistochemical (IHC) testing that in some cases can help identify the tissue of origin, but often these tests cannot give a definitive answer. As a result there is a growing interest in using genomic or proteomic molecular data from the biopsy to identify the location of the primary tumor. Several kinds of molecular data from the metastasis such as gene expression [74], methylation [75], miRNA expression [37] or somatic mutations [69] have been shown to be informative about the tissue of origin. Additionally, a FDA-approved test for CUP classification also exists [75]. In this paper, we present a novel method for predicting the location of the primary tumor using gene expression data.

Inherent to the diagnosis, it is impossible to collect samples of the primary tumor from CUP patients and thus no dataset of metastasis and primary tumor samples from CUP patients exists. Instead CUP classification methods have to use data from patients with known primary tumors as training data. Optimally, the training data would come from metastases collected from patients with a known
primary tumor, but almost all publicly available data comes from biopsies of primary tumors. For this reason, we use publicly available primary tumor RNA-seq data generated by the TCGA Research Network (http://cancergenome.nih.gov/) as training data.

A problem that can lead to classification errors when predicting the tissue type of a tumor is that biopsies are not pure, but a mixture of tumor tissue and adjacent normal tissue. This is particularly a problem for biopsies from metastases since the surrounding tissue will be of a different tissue type. It is thus very relevant to develop a CUP classification method that is robust to sample impurity. Our method, LoCUP (Locating Cancers of Unknown Primary), takes sample purity into account by (1) modelling the mixture of normal and tumor cells directly, (2) employing an empirical prior on sample purity during training, and (3) exploiting that the normal tissue component of the metastatic sample is known when predicting on a metastatic sample. This provides substantial improvements in classification accuracy on impure samples. To our knowledge, LoCUP is the first method to model tumor purity during CUP classification. It is however not the first method to incorporate tumor purity in the analysis of gene expression data. Several methods [107, 3, 4] have been developed that address the closely related problem of deconvolution of a mixed sample into its constituent tumor and normal part. Such deconvolution methods can improve the accuracy of differential expression studies and lead to more accurate biomarkers but they are not directly applicable to the problem of CUP classification.

We compare our method to the classifier used clinically at Department of Molecular Medicine (MOMA), Aarhus University Hospital, Denmark, a multinomial logistic regression classifier with ridge penalty (MLRR) which does not take purity of the samples into account. To the best of our knowledge, this classifier is the only existing solution for CUP samples. Both classifiers are evaluated in three stages. First, classification accuracy is determined when predicting the tissue of origin of primary tumor samples. Second, classification accuracy is determined when predicting on simulated metastatic samples. Third, we evaluate the classifier on a small dataset of real metastatic samples with known primary.

### 8.3 Implementation

The input to our method is a $m \times n$ matrix where entry $i, j$ is the gene expression of gene $j$ in tissue sample $i$. A sample $x_i$ comes from either a primary tumor (T) or normal, healthy tissue (N) adjacent to a primary tumor denoted by $z_i \in \{T, N\}$. The tissue type of the each sample is denoted $y_i \in \{1, \ldots, K\}$. 
The gene expression levels of a tumor sample are assumed to be a mixture of two normal distributions. One distribution represents the normal tissue from which the sample is collected and the other distribution represents the tumor of origin. The degree of mixing of the two distributions is determined by a sample-specific mixing coefficient \( \alpha \in [0, 1] \), which has a tissue-dependent beta-distributed prior with shape parameters \( \beta_1k \) and \( \beta_2k \) which are assumed to be known (see section 8.3). We illustrate the idea behind the model in Figure 8.1. The likelihood splits up into tissue tissue-specific likelihoods, i.e.:

\[
L(\alpha_1, \ldots, \alpha_m, \mu_{Tk}, \mu_{Nk}, \sigma^2_k) = \prod_{i=1}^{m_k} N(x_i; \alpha_i \mu_{Tk} + (1 - \alpha_i) \mu_{Nk}, \sigma^2_k) \beta(\alpha_i; \beta_1k, \beta_2k),
\]

for \( k = 1, \ldots, K \) where \( \mu_{Nk} \) and \( \mu_{Tk} \) are the centroids of normal and tumor tissue respectively from tissue type \( k \). The variance of the normal distribution is fixed within each tissue type. We obtain a pseudo-likelihood by multiplying the likelihood with a ridge regularization term of \( \mu_{Tk} \) and \( \mu_{Nk} \):

\[
R_\lambda(\mu_{Tk}, \mu_{Nk}) = \exp\left(\frac{\lambda}{2\sigma^2_k} \cdot (|\mu_{Tk}|^2 + |\mu_{Nk}|^2)\right)
\]

where we determine \( \lambda \) through grid search. We maximize each tissue-specific pseudo-likelihood independently to get estimates of \( \mu_{Tk} \) and \( \mu_{Nk} \) for \( k = 1, \ldots, K \). That is the 'training' of the model.

To train the model standard gradient ascend can be applied. However, for efficiency reasons we maximize the pseudo-likelihood by alternating between analytically maximizing the likelihood for \( \mu_{Tk} \) and \( \mu_{Nk} \) with fixed \( \alpha_i \)'s, and numerically optimizing \( \alpha \) until convergence.

To predict the tumor tissue type of a given metastatic sample our classifier takes the metastatic sample, \( x \), and the tissue type of the metastatic-adjacent tissue, \( y \). We classify to the best explaining tumor centroid \( \mu_{Tk} \) using the Euclidean distance.

\[
y_p = \arg \min_{k=1,\ldots,K} \min_{\alpha \in (0,1)} |x - \alpha \mu_{Tk} - (1 - \alpha) \mu_{Nk}|
\]

**Estimation of parameters for beta priors**

Shape parameters and for the empirical beta priors were estimated from data published in [5] by fitting a beta distribution to the consensus measurement of purity estimations (abbreviated CPE in [5]) for each disease. Histograms of the estimated purities and the fitted distributions can be seen in Figure 8.2.
Figure 8.1: Example of the model in a 2-dimensional space using simulated data. The normal tissue samples belong to an ordinary normal distribution with "Normal Centroid" as mean. The tumor samples produce an elongated shape because impurity drags them towards the normal tissue centroid.

Preprocessing

Before training and prediction, the data is (1) scaled to zero mean and unit variance, and (2) the number of dimensions is reduced. We performed a grid search with both principal component analysis (PCA) and linear discriminant analysis (LDA) to determine the most suitable method for dimensionality reduction. The best method and number of components for each type of experiment (see section 8.4) is listed in table 8.1. Our experiments without scaling and dimensionality reduction resulted in a significantly lower accuracy for both of the tested methods.

8.4 Application

To investigate the performance of our method we performed a series of experiments on primary tumors, and simulated and true metastatic samples. In this section, we will give an overview of the datasets used and the experimental setup. Source code for the LoCUP classifier and data files used in the analysis are available by request to the corresponding author.
Gene expression data covering $K = 16$ diseases was collected from The Cancer Genome Atlas (TCGA) ($m = 7065, n = 18696$ after removing genes that were expressed in less than 75% of samples). Approximately 10% of the samples are normal tissue. Note that this dataset does not contain any metastatic or CUP samples. We will denote this dataset D1. The dataset was split into a cross-validation (CV) ($m = 6358$, denoted D2) and validation set ($m = 707$, denoted D3) in a stratified manner such that each set contains approximately the same number of tissue types and tumor/normal samples. A fourth dataset, D4, was simulated from the validation set by mixing tumor and normal samples.
(m = 707) from D3. To simulate a metastatic sample, a tumor xT and a normal xN sample is sampled with replacement. The mixed sample is then computed as a xT + (1 − a) xN where a is sampled from the prior distribution for the tissue type of xT. The simulated data maintains the distribution of tissue types of the original data.

Additionally, a dataset, denoted D5, consisting of eight metastatic samples (all with colon primary, 1/8 metastasized to liver, 7/8 metastasized to lung) was obtained from the Department of Molecular Medicine, Aarhus University Hospital, Denmark. This data was processed via a replica of the pipeline used by TCGA. Note that this dataset does not include any normal tissue samples and only contains metastatic samples with known prior. The relationship between these datasets is illustrated in Figure 8.3.

![Figure 8.3: Relationship between the datasets used for test (grid search and cross-validation) and validation. Simulated datasets are shown in blue. Datasets derived during cross-validation (only a single fold is shown) are shown in red.](image)

A grid search with 5-fold cross-validation on D2 was performed to optimize the hyper parameters of each method and the preprocessing pipeline. For each set of parameters, two experiments were performed for each fold of the cross-validation. First, to estimate the best parameters of each method on the task of predicting the tissue type of a tumor/normal sample, the method was trained on the training data and predictions were made on the test data. This experiment is denoted P (primary). Second, to estimate the best parameters of each method on the task of predicting the primary tumor component of a metastatic (mixed)
Table 8.1: Results for the primary (P) and simulated (S) experiments for both methods. The best parameters were found through a grid search for each experiment and method. On simulated metastatic data, our method clearly outperforms the MLRR method. Note that we in some cases obtain a higher accuracy on the validation data since more training data is available.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th>Best Parameters</th>
<th>Number of Components</th>
<th>Regularization Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>P LoCUP</td>
<td>94.9</td>
<td>95.2</td>
<td>LDA</td>
<td>221</td>
</tr>
<tr>
<td>MLRR</td>
<td>96.4</td>
<td>97.2</td>
<td>LDA</td>
<td>15</td>
</tr>
<tr>
<td>S LoCUP</td>
<td>96.3</td>
<td>95.5</td>
<td>LDA</td>
<td>55</td>
</tr>
<tr>
<td>MLRR</td>
<td>91.1</td>
<td>90.8</td>
<td>LDA</td>
<td>105</td>
</tr>
</tbody>
</table>

8.5 Discussion

We performed three experiments to validate the performance of the classifiers. Firstly, the performance of the classifiers on the problem of tissue prediction. That is, the samples that are predicted on may be either tumor or normal and we simply wish to predict the tissue type. The classifiers were trained on D2 and predictions were made on D3. Our method obtains an accuracy of 95.32% while MLRR obtains an accuracy of 97.2%. This is to be expected since we in this experiment do not take advantage of the ability of our method to handle mixtures of tumor and normal tissue. Secondly, we assessed the performance of the classifiers on simulated mixed samples by training on D2 and predicting on D4. Our method obtains an accuracy of 95.5%, compared to 90.8% for MLRR. Our method thus provides a substantial increase in prediction accuracy. The results are summarized in Table 8.1. Thirdly, we were able to collect a small dataset (D5) of metastatic samples with known primary. While this dataset is too small to conclude any improvement in prediction accuracy, it provides an extra layer of validation and suggests that our method predicts as well or better
Table 8.2: Our method correctly predicts five of eight samples while MLRR correctly predicts four of eight samples. Sample 2 is correctly predicted by LoCUP, while MLRR predicts LUAD. Note that the second-best scoring LoCUP prediction for sample 3 is also correct. However, MLRR also predicts correctly on sample 2 and 3 when considering the second-best prediction. Sample 1 may be a polluted or mislabeled sample.

<table>
<thead>
<tr>
<th>#</th>
<th>Prediction</th>
<th>True</th>
<th>MLRR</th>
<th>Est. α</th>
<th>Normal</th>
<th>Tumor</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
<td>1st</td>
<td>2nd</td>
<td>3rd</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>KIRC</td>
<td>KICH</td>
<td>KIRC</td>
<td>KIRC</td>
<td>CESC</td>
<td>LUSC</td>
</tr>
<tr>
<td>2</td>
<td>COAD</td>
<td>LIHC</td>
<td>CESC</td>
<td>LUAD</td>
<td>COAD</td>
<td>CESC</td>
</tr>
<tr>
<td>3</td>
<td>LIHC</td>
<td>COAD</td>
<td>UCEC</td>
<td>LIHC</td>
<td>COAD</td>
<td>CESC</td>
</tr>
<tr>
<td>4</td>
<td>COAD</td>
<td>CESC</td>
<td>BLCA</td>
<td>COAD</td>
<td>LIHC</td>
<td>CESC</td>
</tr>
<tr>
<td>5</td>
<td>LIHC</td>
<td>SKCM</td>
<td>BLCA</td>
<td>LIHC</td>
<td>CESC</td>
<td>BLCA</td>
</tr>
<tr>
<td>6</td>
<td>COAD</td>
<td>CESC</td>
<td>BRCA</td>
<td>COAD</td>
<td>CESC</td>
<td>BRCA</td>
</tr>
<tr>
<td>7</td>
<td>COAD</td>
<td>CESC</td>
<td>BLCA</td>
<td>COAD</td>
<td>LIHC</td>
<td>CESC</td>
</tr>
<tr>
<td>8</td>
<td>COAD</td>
<td>CESC</td>
<td>UCEC</td>
<td>COAD</td>
<td>KIRC</td>
<td>CESC</td>
</tr>
</tbody>
</table>

than the MLRR method on real samples. While the two classifiers agree in most cases, we observe a single sample where our method correctly predicts COAD while the MLRR method predicts LUAD (see Table 8.2). To further validate that our method improves classification on impure tumor samples we plotted the prediction accuracy on the simulated data (D4) binned by the true (simulated) value of $\alpha_i$. See Figure 8.4. The plot clearly illustrates that our method has a higher accuracy on samples of low to mid-range $\alpha_i$, i.e. a higher accuracy on impure samples. In conclusion, we have developed a method for prediction of the tumor of origin of metastatic samples by modelling a metastatic sample as a mixture of the tumor of origin and the adjacent normal tissue of the metastasis. We have shown that our method outperforms the classification method used at Department of Molecular Medicine (MOMA), Aarhus University Hospital, Denmark for clinical diagnostics (see Table 8.1) on simulated metastatic samples, with a clear improvement on very impure samples (see Figure 8.4). We have further validated our method on a small dataset of real metastatic samples (see Table 8.2) and obtained a small improvement. The method models metastatic samples as a mixture between normal and tumor cells, but in some cases tumor purity can also be affected by tumor-infiltrating leukocytes [4]. A possible future improvement of the method would thus be to investigate this phenomenon and possibly improve the classification by adding leukocytes as a third component to the mixture.
Figure 8.4: Accuracy for the LoCUP and MLRR methods binned by the true $\alpha$ of the simulated samples in D4. The number of samples in each bin is shown in bold. Our method outperforms MLRR on samples where $\alpha \in (0.2, 0.7)$, that is low-purity samples.

8.6 Acknowledgments

The results presented here are in part based upon data generated by the TCGA Research Network: http://cancergenome.nih.gov/. We wish to thank Jacob Malte Jensen and Palle Villesen for useful discussions.
Chapter 9

LoCUP: Locating cancers of unknown primary

Dan Søndergaard\textsuperscript{1,*} Svend V. Nielsen\textsuperscript{1} Christian Nørgaard Storm Pedersen\textsuperscript{1} Søren Besenbacher\textsuperscript{2}

This chapter presents an unpublished manuscript currently in preparation:


with only typographical changes.

9.1 Abstract

Summary: Yearly, millions of people are diagnosed with cancer. In many cases, the cancer can be located and treated. The cancer may have spread, or metastasised, to other parts of the body. In approximately 5\% of cases the origin

\textsuperscript{1}Bioinformatics Research Centre (BiRC), Aarhus University, Denmark
\textsuperscript{2}Department of Molecular Medicine (MOMA), Aarhus University Hospital, Denmark
of the metastasis can not be located through extensive diagnostic tests. These cases are denoted cancers of unknown primary (CUPs). Treatment of these cases is notoriously difficult since the primary tumour remains unknown and thus can not be treated directly.

As precision medicine is increasingly applied for more efficient treatments, molecular data has become abundant. We developed a machine learning method, LoCUP, which uses gene expression data to locate the primary tissue in cancers of unknown primary. The method takes the adjacent normal tissue into account to improve prediction accuracy. We have previously shown that the method performs well [97].

Here, we present an efficient implementation of the method, as well as an updated data set. By combining data from The Cancer Genome Atlas (TCGA) and the Genotype-Tissue Expression (GTEx) project we have increased the number of supported tissues and improved the accuracy of prediction from 95.5% to 97.6% on simulated samples.

**Availability and Implementation:** LoCUP was implemented in Python 3 and supports Python 3.4 and newer. The software can be installed through pip or the Conda package manager. Installation instructions are available at https://github.com/dansondergaard/locup.

**Contact:** das@birc.au.dk

**Supplementary information:** None.

### 9.2 Introduction

In [97] we presented a method for predicting the location of the primary tumour in cases of cancers of unknown primary based on gene expression data. That is, given a sample of tissue from the metastasis, we aim to predict the tissue type of the primary tumour. One factor which may cause wrong predictions is the contribution of normal cells to the sample. This may be due to impurities inside the tumour or accidental excision of small amounts of adjacent normal tissue.

Since the tissue type of the metastasis is known, so is the type of the adjacent normal tissue. Our method aims to improve prediction accuracy by modelling a sample as a mixture between tumour and normal tissue. For each tissue $k = 1, \ldots, K$, the method learns two “centroids”, $\mu_{T_k}$ and $\mu_{N_k}$, representing tumour and normal tissue, respectively. To predict the origin of a sample, $x$,
located in tissue \( I \), we compute

\[
y^* = \arg \min_{k=1,\ldots,K} \left[ \min_{\alpha} \left| x - \alpha \mu_T + (1 - \alpha) \mu_N \right| \right].
\]

Estimating the centroids well therefore has a large effect on the performance of the classifier. See [97] for details on the method.

The method was tested on simulated data and a small number of metastatic samples. In both cases, our method was superior to existing methods such as the multinomial logistic regression (MLRR) classifier described in [97]. However, only approximately 10% of the data set consisted of normal samples and thus we expected that increasing this number would result in better estimation of normal tissue centroids and improve classification accuracy.

Here, we obtained a large amount of normal samples from the GTEx project [63] and merged it with the existing data set from TCGA [102]. Additionally, an efficient implementation of the method is made available as a Python and Conda package and includes small improvements in the preprocessing steps. The software may be used from the command line, or imported as a library.

### 9.3 Data

See [97] for a description of the data obtain from TCGA. We obtained data from the GTEx project [63]. Only samples with a RNA Integrity Number (RIN) larger than or equal to 7 were used [20]. The data was processed with the same pipeline as the TCGA data, as described in [97]. This resulted in a data set of 12934 samples of which 4438 (approx. 34%) are normal samples. See table 9.1 for an overview of the data set.

Initial experiments with the LoCUP and MLRR classifiers showed that none of them could distinguish between stomach and esophageal tissues, and thus these tissues were merged. Note that some tissue names in table 9.1 map to multiple TCGA tissues. For example, the kidney class contains samples from the KICH, KIRC and KIRP classes in TCGA.

### 9.4 Experiments

The method was evaluated with 5-fold cross-validation. The evaluation setup was described in detail in [97]. In summary, 90% of the data was used for grid search and cross-validation to estimate the best hyper-parameters of the model. The remaining 10% was used for final validation of the model. Two experiments were performed. The primary (P) experiment aims to estimate the accuracy of
Table 9.1: Overview of the new data set, consisting of tumour and normal samples from TCGA, as well as normal samples from GTEX. Initial experiments showed that none of the classifiers could distinguish between stomach and esophageal tissues and thus we decided to merge these classes.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>GTEX</th>
<th>TCGA</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adrenal Gland</td>
<td>135</td>
<td>0</td>
<td>258</td>
</tr>
<tr>
<td>Bladder</td>
<td>11</td>
<td>19</td>
<td>405</td>
</tr>
<tr>
<td>Brain</td>
<td>1091</td>
<td>0</td>
<td>672</td>
</tr>
<tr>
<td>Breast</td>
<td>170</td>
<td>114</td>
<td>1093</td>
</tr>
<tr>
<td>Cervix Uteri</td>
<td>11</td>
<td>0</td>
<td>305</td>
</tr>
<tr>
<td>Colon</td>
<td>285</td>
<td>51</td>
<td>380</td>
</tr>
<tr>
<td>Kidney</td>
<td>21</td>
<td>129</td>
<td>889</td>
</tr>
<tr>
<td>Liver</td>
<td>96</td>
<td>49</td>
<td>371</td>
</tr>
<tr>
<td>Lung</td>
<td>273</td>
<td>108</td>
<td>1016</td>
</tr>
<tr>
<td>Ovary</td>
<td>86</td>
<td>0</td>
<td>303</td>
</tr>
<tr>
<td>Pancreas</td>
<td>153</td>
<td>0</td>
<td>178</td>
</tr>
<tr>
<td>Prostate</td>
<td>80</td>
<td>52</td>
<td>497</td>
</tr>
<tr>
<td>Salivary Gland</td>
<td>57</td>
<td>44</td>
<td>520</td>
</tr>
<tr>
<td>Skin</td>
<td>824</td>
<td>0</td>
<td>103</td>
</tr>
<tr>
<td>Stomach+Esophageal</td>
<td>0</td>
<td>46</td>
<td>588</td>
</tr>
<tr>
<td>Testis</td>
<td>138</td>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td>Thyroid</td>
<td>268</td>
<td>59</td>
<td>505</td>
</tr>
<tr>
<td>Uterus</td>
<td>74</td>
<td>24</td>
<td>233</td>
</tr>
<tr>
<td>Total</td>
<td>3773</td>
<td>695</td>
<td>8466</td>
</tr>
</tbody>
</table>

Table 9.2: Results with data set consisting of TCGA and GTEX samples. The addition of normal samples results in a higher accuracy.

<table>
<thead>
<tr>
<th></th>
<th>CV</th>
<th>Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>99.3%</td>
<td>99.2%</td>
</tr>
<tr>
<td>Simulated</td>
<td>93.4%</td>
<td>97.6%</td>
</tr>
</tbody>
</table>

the classifier when the tumour tissue is equal to the normal tissue. That is, we simply aim to predict the tissue of a sample. In the simulated (S) experiment, metastatic samples are simulated by mixing tumour and normal samples. The results of these experiments are shown in table 9.2.

A significant increase in prediction accuracy is obtained in both experiments. This increase is larger than the numbers indicate, since the number of supported tissues has increased from $K = 16$ to $K = 18$. 
9.5 Conclusion

We present an efficient implementation of the LoCUP method for predicting the origin of the primary tumor in cancers of unknown primary. It has been demonstrated that including data from the GTEx project increases the accuracy of the classification, but further validation of the results on real metastasis samples is required to obtain a better estimate of the performance of the classifier.
Bibliography


[47] C. Greening et al. “Genome and metagenome surveys of hydrogenase diversity indicate H₂ is a widely-utilised energy source for microbial growth and survival.” In: The ISME Journal (2016).


