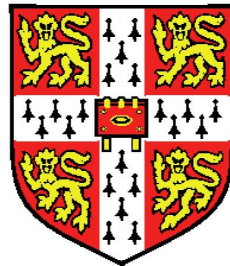


# Motif based computational identification of protein subcellular localisation



Mutlu Doğruel

Wellcome Trust Sanger Institute

University of Cambridge

A thesis submitted for the degree of

*Doctor of Philosophy*

January 2008

---

To my deeply missed father,

Sami Han Doḡruel.

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

## Acknowledgements

I would like to express my gratitude to Dr. Tim Hubbard for being a great supervisor. His support, invaluable suggestions and comments during our weekly meetings are much appreciated. I am also deeply indebted to my PhD committee members Dr. Ewan Birney and Dr. Ian Dunham for their time and advice. I would also like to thank my examiners Prof. Søren Brunak and Dr. Nick Goldman for reading the thesis thoroughly and for their useful feedback on the preliminary version of the thesis.

My sincere thanks go to Dr. Thomas Down for his useful suggestions and also patience during the discussions we made (usually accompanied with a cup of coffee) on Biojava, NestedMICA, Eponine and many other topics... Many thanks also to the other members of Tim's research group at the Sanger Institute: Dr. Andreas Prlić, and will-be Drs. Markus Brosch, Matias Piipari, and Jenny Mattison for making my PhD days more enjoyable, and being excellent friends. I think Markus did a great job by encouraging me to join the Cambridge University gliding club. Being in the air over the weekends in the summer was a good way to relax and get ready for the productive work days;)

Thanks to Tim, Matias, Andreas and Jenny again for critically reading the preliminary versions of the chapters. I also thank Antony Quinn from EBI for his suggestions about the thesis.

I also would like to thank the Sanger Institute's web, systems and library staff for their help on various occasions.

I am grateful to the Wellcome Trust for financing me during my 4 years in Cambridge.

Finally, I am deeply and forever indebted to my parents, sisters and brother for their love, support and encouragement throughout my entire life: Sizi çok seviyorum!

[Mutlu Doğruel](#), January 2008, Cambridge, UK.

## Abstract

Discovering overrepresented patterns in amino acid sequences is an important step in protein functional annotation which includes the identification of subcellular localisation. I adapted and extended NestedMICA, an *ab initio* protein motif finder originally developed for finding transcription binding site motifs, to find short protein signals, and compared its performance with another popular protein motif finder, MEME.

In order to assess NestedMICA as a protein motif finder, I have tested it on synthetic datasets produced by spiking instances of known motifs from protein databases into a randomly selected set of protein sequences. Apart from the artificially implanted motifs, NestedMICA also successfully recovered subcellular localisation signals from biologically-authentic test sets. NestedMICA found most of the short test protein motifs spiked into a test set of sequences at different frequencies. In all the assessment experiments, its overall motif discovery performance was better than that of MEME.

As a practical application of NestedMICA, I developed a novel Support Vector Machines based protein subcellular classification tool,

Lokum, for eukaryotic protein subcellular localisation prediction, covering all major localisation classes for animal, fungal and plant sequences. It uses targeting and retention signal motifs reported by NestedMICA, and other protein features including transmembrane topologies and amino acid composition. Additionally, in Lokum I use bipartite nuclear localisation signals obtained by adding protein support to Eponine, a tool originally developed for transcription start site modeling. Lokum does not use sequence similarity, or any other *a priori* knowledge such as known nuclear localisation signals by searching databases.

I compared proteins targeted into the nuclei and nucleoli in terms of the features used in Lokum, and also their predicted disorder regions. I demonstrate that it is possible to computationally distinguish these two sub-nuclear protein categories.

Finally, as an alternative to the transmembrane topology predictor TMHMM that is used in Lokum, I designed and tested a new prototype program that is based on hidden Markov models (HMM). The HMM has been trained by a novel, nested sampling based transition probability optimisation procedure.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General introduction . . . . .	1
1.1.1	Previous work on subcellular localisation prediction . . . . .	6
1.2	Sequence identity thresholds . . . . .	12
1.3	Computational methodologies . . . . .	15
1.3.1	HMMs . . . . .	15
1.3.2	The general idea behind motif finding . . . . .	18
1.3.3	Inference by Nested Sampling . . . . .	20
1.3.4	Support vector machines . . . . .	22
<b>2</b>	<b>NestedMICA as an <i>ab initio</i> protein motif finder</b>	<b>25</b>
2.1	Background . . . . .	25
2.2	Materials and methods . . . . .	28
2.2.1	NestedMICA . . . . .	28
2.2.1.1	The NestedMICA sequence model . . . . .	29
2.2.1.2	Implementation of NestedMICA . . . . .	31
2.2.2	Adding protein support to NestedMICA . . . . .	33
2.2.3	Program output and sequence logos . . . . .	34

2.2.4	Background model training . . . . .	35
2.2.5	Testing NestedMICA’s performance . . . . .	36
2.3	Results and discussions . . . . .	41
2.3.1	Protein sequence background model . . . . .	41
2.3.2	Performance vs. motif abundance . . . . .	42
2.3.3	Performance with multiple motifs . . . . .	48
2.3.4	Performance vs. protein length . . . . .	53
2.3.5	“Null test” and significance of motifs . . . . .	55
2.3.6	Testing non- <i>ab initio</i> motif finders . . . . .	58
2.4	Conclusions . . . . .	59
2.5	Availability and requirements of NestedMICA . . . . .	61
<b>3</b>	<b>Lokum: <i>ab initio</i> protein subcellular localisation prediction for eukaryotes by using mono and bipartite motifs, transmembrane protein topologies, and amino acid composition</b>	<b>63</b>
3.1	Introduction . . . . .	63
3.1.1	Features used in Lokum . . . . .	64
3.1.2	Predicted classes . . . . .	69
3.2	Materials and methods . . . . .	69
3.2.1	Localisation motif discovery with NestedMICA . . . . .	69
3.2.2	Motif selection . . . . .	71
3.2.3	Using Eponine with NestedMICA for multi-component motif discovery . . . . .	73
3.2.4	Using amino acid composition . . . . .	77

3.2.5	Using transmembrane topology predictions . . . . .	78
3.2.6	Training and testing of SVM . . . . .	79
3.2.7	Evaluation of Lokum predictions . . . . .	80
3.3	Results . . . . .	80
3.3.1	Discovered monopartite motifs . . . . .	82
3.3.1.1	Contribution of N-linked glycosylation signal . . . . .	85
3.3.1.2	Alternative ER retrieval . . . . .	86
3.3.1.3	Scanning motifs in certain positions . . . . .	89
3.3.1.4	Scoring multiple instances of motifs . . . . .	89
3.3.2	Bipartite motif models . . . . .	91
3.3.2.1	Bipartite NLS . . . . .	91
3.3.2.2	Bipartite PTS2 . . . . .	92
3.3.3	Golgi N-terminal transmembrane topology prediction statistics help in localisation prediction . . . . .	94
3.3.4	Effect of amino acid composition . . . . .	95
3.3.5	Lokum’s performance . . . . .	97
3.3.6	Contributions of different features . . . . .	98
3.3.7	Contribution of disordered region predictions . . . . .	102
3.4	Discussions . . . . .	103
3.5	Availability . . . . .	105
<b>4</b>	<b>Discriminating nucleolar proteins from nuclear proteins: is it possible?</b>	<b>107</b>
4.1	Introduction . . . . .	107

4.1.1	Disordered protein regions . . . . .	111
4.1.2	Protein disorder region prediction . . . . .	112
4.2	Materials and methods . . . . .	113
4.2.1	Datasets . . . . .	113
4.2.2	Training background models for nucleolar and nuclear datasets	116
4.2.3	Running RONN . . . . .	118
4.2.4	Training the SVM . . . . .	118
4.3	Results . . . . .	120
4.4	Discussions and conclusions . . . . .	131
<b>5</b>	<b>Predicting protein transmembrane topology and signal peptides: An HMM approach with a new parameter optimisation strategy</b>	<b>134</b>
5.1	Introduction . . . . .	134
5.1.1	The aim of this study . . . . .	134
5.1.2	Transmembrane topology and signal peptide prediction . .	136
5.2	Materials and methods . . . . .	139
5.2.1	Architecture of the HMM . . . . .	139
5.2.1.1	Representing helix caps in the HMM . . . . .	142
5.2.1.2	Duration HMM states . . . . .	147
5.2.2	Datasets and training of the model . . . . .	148
5.2.3	Transition probability optimisation: a new approach . . . .	150
5.3	Results . . . . .	159
5.3.1	Signal peptides at the DNA level . . . . .	161
5.4	Discussions . . . . .	163

<b>6</b>	<b>Conclusions</b>	<b>165</b>
	<b>A Motifs discovered by NestedMICA</b>	<b>170</b>
	<b>B Amino acid composition rates in different localisations</b>	<b>180</b>
	<b>C Sequence IDs of nuclear and nucleolar proteins filtered from the LOCATE database</b>	<b>191</b>
	C.1 Proteins in nucleoli . . . . .	191
	C.2 Proteins in nuclei . . . . .	194
	<b>D Kullback-Leibler divergence for transmembrane helix cap posi- tions in terms of amino acid composition</b>	<b>197</b>
	<b>References</b>	<b>233</b>

# List of Figures

1.1	The “thesis graph” . . . . .	4
1.2	Probability of staying in the same state in a minimum duration capable HMM state . . . . .	17
1.3	Profile HMMs . . . . .	18
2.1	Likelihood curve for different number of mosaic classes . . . . .	37
2.2	Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set1 . . . . .	43
2.3	Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set2 . . . . .	44
2.4	Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set3 . . . . .	45
2.5	Inserting more than one different motif into the sequences. . . . .	51
2.6	Motif recovery performance against sequence length . . . . .	55
2.7	NestedMICA’s “null motifs” . . . . .	57
2.8	A snapshot showing the regular expressions reported by the Dil- imot web service . . . . .	60
3.1	Eponine TSS model . . . . .	74

## LIST OF FIGURES

---

3.2	Eponine TTS model . . . . .	75
3.3	SVM kernel parameter optimisation . . . . .	81
3.4	Some of the protein localisation related signals as recovered by NestedMICA. . . . .	83
3.5	Unannotated or less known motifs found by NestedMICA. . . . .	84
3.6	Manually built motifs that are used in the ChloroP predictor. . . . .	85
3.7	Contribution of the N-linked glycosylation motif in localisation classification . . . . .	87
3.8	The two C-terminal ER retention motifs reported . . . . .	88
3.9	A ROC curve showing the effect of scanning sequences with PWMs in certain segments only . . . . .	90
3.10	Schematic representation of the Eponine bipartite NLS model . . . . .	92
3.11	The expected number of amino acids in the first 60 N-terminal residues to be part of a transmembrane region. . . . .	96
3.12	Individual contributions of features used in the SVM . . . . .	101
3.13	Lokum prediction service hosted by the Wellcome Trust Sanger Institute . . . . .	106
4.1	Nuclear pore . . . . .	109
4.2	NestedMICA motifs discovered from nuclear and nucleolar datasets	117
4.3	A protein disordered regions plot based on RONN predictions . . . . .	119
4.4	A selection of the protein motifs recovered by NestedMICA from a set of nuclear and a set of nucleolar proteins, using a cytoplasmic background . . . . .	121

## LIST OF FIGURES

---

4.5	Score histograms for N- and C-terminal nucleolar motifs . . . . .	122
4.6	Distributions of amino acids predicted to be within TM helices in nuclear and nucleolar proteins . . . . .	124
4.7	Differences between nucleolar and nuclear proteins in terms of their amino acid compositions . . . . .	125
4.8	Score distribution of a core nucleolar motif within nuclear and nucleolar proteins . . . . .	127
4.9	A larger number of nucleolar localisation signal hits can be found in disordered regions . . . . .	129
4.10	Generally a larger number of NLS signal hits can be found in disordered regions of nucleolar proteins compared to nuclear sequences	130
5.1	The architecture of the developed transmembrane predictor . . . . .	141
5.2	Total likelihood function monotonically increases in nested sampling	153
5.3	Transition probability set having the least likelihood is replaced by new better one . . . . .	154
5.4	It becomes more difficult over time to find “acceptable” states . . .	155
5.5	Signal peptide motif at the RNA level . . . . .	162
A.1	“Nuclear motifs” discovered by NestedMICA . . . . .	171
A.2	“Plasma membrane motifs” discovered by NestedMICA . . . . .	172
A.3	“Cytoplasmic motifs” discovered by NestedMICA . . . . .	173
A.4	“Mitochondrial motifs” discovered by NestedMICA . . . . .	174
A.5	“Endoplasmic reticulum motifs” discovered by NestedMICA . . . . .	175
A.6	“Golgi motifs” discovered by NestedMICA . . . . .	176



## LIST OF FIGURES

---

A.7	“Extracellular motifs” discovered by NestedMICA . . . . .	177
A.8	“Lysosome motifs” discovered by NestedMICA . . . . .	178
A.9	“Peroxisomal motifs” discovered by NestedMICA . . . . .	178
A.10	“Vacuolar motifs” discovered by NestedMICA . . . . .	179
B.1	Amino acid composition for Alanine (ALA / A) . . . . .	181
B.2	Amino acid composition for Arginine (ARG / R) . . . . .	181
B.3	Amino acid composition for Asparagine (ASN / N) . . . . .	182
B.4	Amino acid composition for Aspartic Acid (ASP / D) . . . . .	182
B.5	Amino acid composition for Cysteine (CYS / C) . . . . .	183
B.6	Amino acid composition for Glutamine (GLN / Q) . . . . .	183
B.7	Amino acid composition for Glutamic Acid (GLU / E) . . . . .	184
B.8	Amino acid composition for Glycine (GLY / G) . . . . .	184
B.9	Amino acid composition for Histidine (HIS / H) . . . . .	185
B.10	Amino acid composition for Isoleucine (ILE / I) . . . . .	185
B.11	Amino acid composition for Leucine (LEU / L) . . . . .	186
B.12	Amino acid composition for Lysine (LYS / K) . . . . .	186
B.13	Amino acid composition for Methionine (MET / M) . . . . .	187
B.14	Amino acid composition for Phenylalanine (PHE / F) . . . . .	187
B.15	Amino acid composition for Proline (PRO / P) . . . . .	188
B.16	Amino acid composition for Serine (SER / S) . . . . .	188
B.17	Amino acid composition for Thereonine (THR / T) . . . . .	189
B.18	Amino acid composition for Tryptophan (TRP / W) . . . . .	189
B.19	Amino acid composition for Tyrosine (TYR / Y) . . . . .	190

## LIST OF FIGURES

---

B.20 Amino acid composition for Valine (VAL / V) . . . . .	190
--	-----

# List of Tables

1.1	A list of some popular eukaryotic localisation predictors . . . . .	7
1.2	Maximum mutual sequence identity rates allowed in the different predictors . . . . .	13
1.3	Several allowed maximum pairwise sequence identity rates versus the number of vacuolar sequences . . . . .	14
2.1	Motif recovery performance for NestedMICA and MEME for individual test sets. . . . .	46
2.2	Total motif recovery performance summary for NestedMICA and MEME. . . . .	47
2.3	Sensitivity and specificity values for motifs of Set 1, reported by NestedMICA and MEME in the single-motif spiking tests . . . . .	48
2.4	Sensitivity and specificity values for motifs of Set 2, reported by NestedMICA and MEME in the single-motif spiking tests . . . . .	49
2.5	Sensitivity and specificity values for motifs of Set 3, reported by NestedMICA and MEME in the single-motif spiking tests . . . . .	50
2.6	Performance summary for NestedMICA in the multiple motif spiking tests. . . . .	52

## LIST OF TABLES

---

2.7	Performance summary for MEME in the multiple motif spiking tests.	53
2.8	Sensitivity and specificity values for motifs reported by Nested-MICA in the multiple-motif spiking tests . . . . .	53
2.9	Sensitivity and specificity values for motifs reported by MEME in the multiple-motif spiking tests . . . . .	54
3.1	Sequences used in the motif discovery phase . . . . .	71
3.2	Protein background parameters for datasets used in localisation motif discovery . . . . .	72
3.3	Alternative amino acid groupings used in composition calculation	78
3.4	Prediction performance summary for Lokum. . . . .	99
3.5	MCCs and correct prediction rates for Lokum, MultiLoc and PSORT	100
4.1	Performance measures calculated from the blind testing of nine disorder prediction methods against the main blind test set of 80 proteins of CASP 6 . . . . .	114
5.1	Amino acid emission probabilities in the transmembrane helix cytoplasmic side cap. . . . .	142
5.2	Amino acid emission probabilities in the transmembrane helix non-cytoplasmic side cap. . . . .	143
5.3	KL deviations of cytoplasmic side helix termini from the noncytoplasmic side helix termini in relative amino acid abundance rates.	144
5.4	Minimum allowed emission state occupancy numbers for the transmembrane topology predicting HMM . . . . .	149

## LIST OF TABLES

---

5.5	Sequences used in the training of Phobius and of the program developed . . . . .	149
5.6	Prediction performance summary for “TM-and-SP”, “TM-only” and “non-SP, non-TM” proteins. . . . .	160
5.7	Correct prediction rates for SP-only proteins. . . . .	160
C.1	Nucleolar proteins . . . . .	194
C.2	Nuclear proteins . . . . .	196
D.1	KL deviations of cytoplasmic side helix cap position 1 from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates. . . . .	198
D.2	KL deviations of cytoplasmic side helix cap position 2 from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates. . . . .	199
D.3	KL deviations of cytoplasmic side helix cap position 3 from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates. . . . .	200
D.4	KL deviations of cytoplasmic side helix cap position 4 from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates. . . . .	201
D.5	KL deviations of cytoplasmic side helix cap position 5 from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates. . . . .	202
D.6	KL deviations of cytoplasmic side helix cap positions. . . . .	203

## LIST OF TABLES

---

D.7	KL deviations of cytoplasmic side helix cap positions. . . . .	204
D.8	KL deviations of non-cytoplasmic side helix cap positions. . . . .	205
D.9	KL deviations of non-cytoplasmic side helix cap positions. . . . .	206

# Chapter 1

## Introduction

### 1.1 General introduction

Proteins perform vital functions in all living organisms. After being synthesised in the cytosol, most of the proteins are transferred to other places in the cell or sent out to the extracellular space where they carry out their specialised tasks. One of the important questions modern biology has been trying to address is how new born proteins can find their ways in reaching their destinations. Throughout their journey they come across many obstacles, including different organelle or cell membranes, pores that have to be passed, and pathways that must be followed. Studies on the structure of the secretory pathway by George Palade were awarded with a Nobel prize in Physiology or Medicine in 1974. This and other pioneering studies yielded the theory that proteins carry intrinsic signals that govern their localisation, which sounds simple and natural to us now. This important discovery brought Günter Blobel the 1999 Nobel prize in Physiology or Medicine, less than a decade ago.

The identification of protein subcellular localisation has been the subject of

numerous experimental and computational studies. However, despite the advances made in understanding the underlying mechanisms, this complicated process has still not been fully explained. Most of the protein targeting mechanisms have been identified and well studied, certain localisation-related protein signals have been discovered, but it is still not possible to determine every proteins' localisation by inspecting only their amino acid sequences. In automatic protein localisation annotation, computational methods that rely on sequence and structure homology could be advantageous only if some other similar protein localisations have already been fully characterised in experiments. Furthermore, these methods, while performing well, cannot help much in explaining the underlying biological processes and interactions involved in protein targeting.

In this study, in accordance with the general notion that “signals govern protein targeting”, my aim was to investigate whether an *ab initio*, signal-based computational prediction system can adequately help us to predict and classify sub-cellular localisation, without using any kind of sequence similarity, text-mining, or any kind of database searches to check for known localisation signal matches. Using known localisation signals, protein domain motifs etc., but no sequence similarity could still be anticipated as a valid *ab initio* methodology, however, in this work, in addition to predicting localisation, as a secondary goal I tried to directly discover potentially localisation-related amino acid sequence motifs as well, by extending a robust, *ab initio*, probabilistic DNA motif discovery tool program, NestedMICA (Down & Hubbard, 2005) to work on amino acid sequences.

As can be seen in the “thesis graph” (Figure 1.1), Chapter 2 is devoted to motif finding using NestedMICA, which was originally developed for transcription



factor binding site motif finding. This chapter is an extended version of our study, published under the title “NestedMICA as an *ab initio* protein motif discovery tool” (Doğruel *et al.*, 2008), where I added protein support to the program and fine tuned it for optimal protein motif discovery. A comparison of the protein-capable NestedMICA with another popular program, MEME (Bailey & Elkan, 1994), is also given in the same chapter.

In Chapter 3, I introduce Lokum, or localisation by using motifs, a novel eukaryotic protein subcellular localisation prediction program which mainly uses motifs discovered by the new NestedMICA. In addition to NestedMICA motifs found from datasets of proteins with experimentally determined localisations, I modified and used a hierarchical motif finder, Eponine (Down & Hubbard, 2002), for discovering and modeling multi-component localisation motifs such as the bipartite nuclear localisation signals. I changed Eponine, originally developed for finding transcription start site motif models, to work with amino acid sequence, too. Lokum incorporates both mono and bipartite motifs along with amino acid composition, and finally transmembrane topology statistics. In Lokum, predictions based on these features are made by a Support Vector Machine (SVM), a robust machine learning strategy.

The predicted eukaryotic localisation categories are:

1. Cytoplasmic
2. Nuclear
3. Plasma membrane

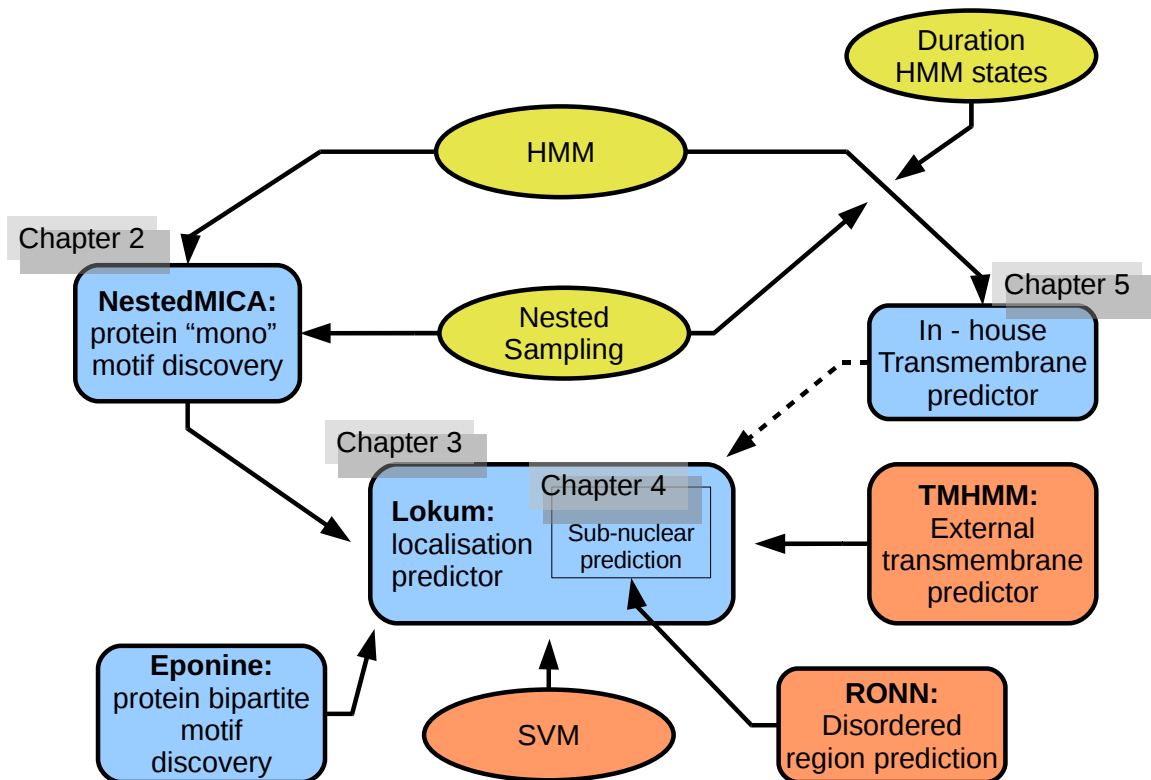


Figure 1.1: **The “thesis graph”**. Main relations between the thesis chapters are shown. Rectangular shapes indicate programs that can produce “deliverables” such as a motif or a prediction, whereas elliptical shapes indicate the used intermediate computational methodologies or algorithms. Orange shapes represent external tools used, while the others are developed, implemented or modified programs.

4. Endoplasmic reticulum (ER)
5. Golgi
6. Extracellular / secretory
7. Mitochondrial
8. Peroxisomal
9. Lysosomal
10. Vacuolar
11. Chloroplast

The first nine localisations above represent the protein localisation categories for animals. Another nine categories predicted for fungi are the first eight classes with the addition of vacuolar proteins (instead of lysosomes). Finally, in addition to the categories in the list of fungal localisations, proteins targeted into the chloroplast are predicted, too, to have a total of ten categories for plants.

Chapter 4, which can be considered as an application of what is learned in Chapters 2 and 3, discusses whether it is possible to fine tune predictions by classifying some nuclear proteins in terms of their sub-localisation categories. I chose nuclear proteins as an example, because it was possible to find a significant number of protein sequences from databases, annotated as “nuclear” or “nuclear”. As an addition to the features used in Lokum, here I also evaluate the use of protein disordered regions as predicted by the RONN (Yang *et al.*, 2005) disorder predictor (Figure 1.1).

Transmembrane topologies are predicted by an external program, TMHMM (Krogh *et al.*, 2001). As an alternative to this, I developed a hidden Markov model (HMM)-based, prototype predictor (see Chapter 5) which can be plugged into this system. The underlying HMM of this predictor was optimised for its transition probabilities by using a novel procedure developed that relies on nested sampling. This new approach is introduced in the same chapter along with the prototype transmembrane topology predictor. For this HMM approach to work more efficiently I also implemented “duration capable HMMs” which are defined in this chapter.

The main chapters in the thesis have their own introduction sections that will be useful while reading a particular chapter. In this general introduction, I summarise the most popular subcellular localisation prediction programs, briefly describe three main computational techniques I used in the developed tools, namely support vector machines, HMMs, and lastly motif finding by using sampling strategies.

Finally, in the Conclusions (Chapter 6, page 165) I summarise the developed computational tools and discuss their applications in biology, together with their pros and cons. In the rest of this introductory chapter, I briefly mention previous work done on automatic protein subcellular localisation prediction and the main computational tools used.

### 1.1.1 Previous work on subcellular localisation prediction

Dozens of software applications are available that deal with particular aspects of subcellular localisation prediction. Some of the popular ones are listed in Table

## 1.1 General introduction

1.1. I will first mention three widely used sets of prediction programs, before discussing others: those developed at the Danish Technical University (DTU), programs by the Rost group in Columbia University, and those developed by Kenta Nakai of the University of Tokyo and his colleagues.

Predictor	Architecture	Features	Original reference
TargetP	ANN	N-terminal sequence	<a href="#">Emanuelsson <i>et al.</i> (2000)</a>
SignalP	HMM and ANN	N-terminal sequence	<a href="#">Nielsen <i>et al.</i> (1999)</a> <a href="#">Bendtsen <i>et al.</i> (2004b)</a>
ChloroP	ANN	Presence of cTP	<a href="#">Emanuelsson <i>et al.</i> (1999)</a>
LipoP	HMM	N-terminal sequence	<a href="#">Juncker <i>et al.</i> (2003)</a>
PredictNLS	Template based	NLS look-up	<a href="#">Cokol <i>et al.</i> (2000)</a>
LOChom	Database	Sequence similarity	<a href="#">Nair &amp; Rost (2002b)</a>
LOckey	Lexical analysis	Sequence similarity	<a href="#">Nair &amp; Rost (2002a)</a>
PSORT	“If-then” rules	PSORT features	<a href="#">Nakai &amp; Kanehisa (1991)</a>
PSORT II	kNN	PSORT features	<a href="#">Horton &amp; Nakai (1997)</a> <a href="#">Nakai &amp; Horton (1999)</a>
iPSORT	Rule based	N-terminal patterns	<a href="#">Bannai <i>et al.</i> (2002)</a>
WolfPSORT	kNN	PSORT features, aa	<a href="#">Horton <i>et al.</i> (2007)</a>
PLOC	SVM	aa	<a href="#">Park &amp; Kanehisa (2003)</a>
SubLoc	SVM	aa features	<a href="#">Hua &amp; Sun (2001)</a>
CELLO	SVM	aa of k-words	<a href="#">Yu <i>et al.</i> (2004)</a>
ELSpred	SVM	aa, BLAST	<a href="#">Bhasin &amp; Raghava (2004)</a>
Proteom Analyst	Naive Bayes	SwissProt keywords	<a href="#">Lu <i>et al.</i> (2004)</a>
pTarget	SVM	PFAM domains	<a href="#">Guda &amp; Subramaniam (2005)</a>
MultiLoc	SVM	aa, motif DBs	<a href="#">Höglund <i>et al.</i> (2006)</a>
BaCelLo	SVM	aa, decision tree	<a href="#">Pierleoni <i>et al.</i> (2006)</a>

Table 1.1: **A list of some popular eukaryotic localisation predictors.** For each prediction tool the main computational methodology and features used are listed, along with related bibliographic reference(s). ANN stands for Artificial Neural Networks, kNN represents the “k-Nearest Neighbours” algorithm, “aa” indicates amino acid composition, SVM indicates support vector machines , cTP is Chloroplast targeting peptide, and finally DB means database.

One of the most popular protein subcellular localisation predictors that use N-terminal sorting signals is TargetP ([Emanuelsson \*et al.\*, 2000](#)), developed at the DTU. This program is limited to only three classes (signal peptides (SP),

mitochondrial, and “other”) for non-plants, and four classes (SP, mitochondrial, chloroplast, and “other”) for plants. The “other” class represents proteins that do not have N-terminal signals, and consists of only nuclear and cytosolic proteins. Until now, most of the novel programs that predict the presence of N-terminal targeting signals still use TargetP datasets as a benchmark set and compare their prediction performance with that of TargetP. Another popular tool, SignalP (Bendtsen *et al.*, 2004b; Nielsen *et al.*, 1997b) from the same group, predicts the presence and location of signal peptide cleavage sites, and can accept eukaryotic, Gram-positive and Gram-negative bacteria input. SignalP has two different architectures: one is based on artificial neural networks (ANN), while the other is an HMM predictor. ChloroP (Emanuelsson *et al.*, 1999) predicts chloroplast transit peptides (cTP) and the possible cleavage site position. Similar to TargetP, it is based on ANNs. LipoP (Juncker *et al.*, 2003) predicts lipoprotein signal peptides for Gram-negative bacteria, achieving a reported correct prediction rate of 96.8%. LipoP is an HMM based prediction system. Programs developed in this group are mainly specialised tools, and based on predicting certain localisation related features in proteins. Last year, the group published a Nature Protocols article (Emanuelsson *et al.*, 2007) describing the use of several localisation predictors that aim to detect N-terminal sorting signals, including TargetP, SignalP, and ChloroP which are all hosted at DTU’s Centre for Biological Sequence Analysis.

Predictors developed in Rost’s group can possibly be shortened by LOC\*, with the exception of PredictNLS. PredictNLS (Cokol *et al.*, 2000) uses NLSdb (Nair *et al.*, 2003), a database of nuclear localisation signals (NLS) containing both experimentally verified and “extrapolated” NLSs, to predict nuclear pro-

teins. LOChom (Nair & Rost, 2002b) is a sequence similarity based classifier, and it is based on the findings of a large-scale analysis of the relation between sequence similarity and identity in subcellular localisation. Another “LOC” program, LOCKey, (Nair & Rost, 2002a) classifies proteins according to their localisations by a lexical analysis of SWISS-PROT keywords that assigns sub-cellular localisation. LOCtarget and LOCtree (Nair & Rost, 2004) are two programs that combine and use the other LOC\* predictors, with the latter being based on SVM decision trees.

Predictors based on Prof. Nakai’s “localisation knowledge base” (Nakai & Kanehisa, 1991, 1992) constitute the PSORT family of programs. This knowledge base is a set of “if-then” rules that are either determined from experimental observations or derived empirically. The first PSORT predictor was announced together with the knowledge base publication by Nakai & Kanehisa in 1991. This is an expert system which is based on detection of the compiled rules. An improved version, PSORT II (Horton & Nakai, 1997; Nakai & Horton, 1999) works by detecting the same PSORT features using a “k-nearest neighbours” classifier. Bannai *et al.* extended the PSORT family by a new predictor, iPSORT (Bannai *et al.*, 2002), which is the “TargetP counterpart” of this group. It basically has additional rules to check for some physiochemical patterns in signal peptide sequences. This program did not perform as well as the neural network based TargetP, nevertheless it directly used signals and signal properties for N-terminal sorting sequence prediction. After the development of PSORT-b (Gardy *et al.*, 2003, 2005) to predict Gram-negative bacterial localisation, the newest predictor of the PSORT family, WoLF PSORT (Horton *et al.*, 2007) was released (more

than one year before its publication). WoLF PSORT, a eukaryotic localisation predictor, is an extension of PSORT II. It uses the PSORT “if-then” rules, but additionally incorporates some of the iPSORT features. This program uses amino acid composition as well as some functional motifs such as DNA-binding motifs obtained from public protein databases. As in the previous version PSORT II, it is based on the k-nearest neighbour algorithm with feature selection.

In addition to the above, there are many other, mostly support vector machine -based protein classification programs. Examples of SVM-based methods using amino acid composition as their main feature to predict eukaryotic protein localisation categories include: PLOC (Park & Kanehisa, 2003), SubLoc (Hua & Sun, 2001), CELLO (Yu *et al.*, 2004), and ELSpred (Bhasin & Raghava, 2004) (also PLSpred (Bhasin *et al.*, 2005) from the same authors for bacteria), and so on. BaCelLo (Pierleoni *et al.*, 2006) is another SVM-based prediction system that can predict 4 localisation categories for non-plant and 5 for plant protein sequences. BaCelLo does not distinguish between secretory pathway proteins. It uses N- and C-terminal sequence features such as the composition rates of amino acid chunks of different lengths from both termini.

In spite of the numerous available methods to predict protein localisation, there are only a few programs that can predict all major eukaryotic localisation categories. Apart from the WoLF-PSORT program mentioned above, Proteom Analyst (Lu *et al.*, 2004), pTarget (Guda & Subramaniam, 2005) and MultiLoc (Höglund *et al.*, 2006) are the notable multi-class predictors.

Proteom Analyst predicts protein localisations for animal, plant, fungi, Gram-negative and Gram-positive bacteria with reported correct prediction rates of



around 81% for fungi, and at rates ranging from 92 to 94% for the other four categories. These high prediction accuracies are not surprising because this method, combined with some sequence features, looks up textual subcellular localisation annotations of other homologous sequences in annotated databases to report localisation.

pTarget is a subcellular localisation predictor that searches for the presence of over 2100 PFAM (Bateman *et al.*, 2004) domains in sequences, and also uses N- and C-terminal amino acid composition. It classifies mammalian proteins in nine localisation classes. Sequences used in pTarget’s development and evaluation have been filtered to remove highly homologous sequences. However, only sequences having identity rates greater than 95% were eliminated in the localisation datasets used for training and testing of the program, which possesses the danger that sequences with too high identities will be ‘recognised’ by the program rather than ‘predicted’.

MultiLoc (Höglund *et al.*, 2006) is a new, SVM-based eukaryotic localisation predictor which combines features such as N-terminal signals, amino acid composition, and protein motifs from databases including Prosite (Hulo *et al.*, 2006) and the nuclear localisation signals database NLSdb (Nair *et al.*, 2003). It predicts nine animal, nine fungal and ten plant subcellular localisation categories with an accuracy of around 74%. It uses a total of 5959 non-homologous sequences having a maximum identity rate of 80%.

Sprenger *et al.* (2006) compared five mammalian protein subcellular localisation programs including the multi-class predictors CELLO, MultiLoc, Proteom Analyst, pTarget, and WoLF PSORT, although these are not equivalent pro-

## 1.2 Sequence identity thresholds

---

grams in terms of their methodology and training procedures in that some are not *ab initio*, and that they were originally trained from datasets having different sequence similarity rates. Nevertheless, all these prediction programs can predict the nine major subcellular localisation categories, and they are publicly available for download or use as a web service that can accept large number of input sequences. This comparative study showed that no individual method had a sufficient level of sensitivity for the datasets used in the evaluation that would enable reliable application to entirely new or different proteins. All methods showed lower performance than reported in the original publications. The benchmarking tests were performed with low-redundancy sequences from the LOCATE database (Fink *et al.*, 2006). However, the datasets were constructed such that two-thirds of them consist of only nuclear and extracellular proteins, while the remaining seven localisation categories make up the remaining portion.

Despite this, even when we judge from what these programs report as their accuracies, there is still a need for true *ab initio* automatic classifiers that can mimic the underlying biology and predict localisation with higher accuracies in the protein annotation field.

## 1.2 Sequence identity thresholds

Protein subcellular localisation predictors (see the previous section on page 6) use amino acid sequences from public protein databases such as SWISS-PROT (Bairoch & Apweiler, 1996, 2000) for program training and prediction accuracy assessment purposes. Highly homologous sequences present in datasets used in program training and testing phases could result in misleading reported prediction

## 1.2 Sequence identity thresholds

---

accuracies, and therefore must be removed from sequence datasets prior to training and testing. Different programs allow different maximum mutual sequence identity thresholds to reduce sequence redundancy. Specialised programs that predict only a certain number of protein localisation categories tend to use non-homologous sequences as determined by some homology reduction algorithms (Hobohm *et al.*, 1992), or empirically determined sequence identity thresholds that could be as low as 30%. However, those covering the majority of protein localisation categories (generally 9-11 classes) tend to use higher thresholds, as demonstrated in Table 1.2.

Program	Max allowed sequence identity
MultiLoc (Höglund <i>et al.</i> , 2006)	80%
PLOC (Park & Kanehisa, 2003)	80%
pTarget (Guda & Subramaniam, 2005)	95%
PSORTb 2.0 (Gardy <i>et al.</i> , 2005)	100%
Proteome Analyst (Szafron <i>et al.</i> , 2004)	100%

Table 1.2: **Maximum mutual sequence identity rates allowed in the different predictors.** PSORTb datasets are not filtered to eliminate sequence redundancy. Sequence homology-based programs such as Proteome Analyst tend to use the entire protein sequence sets.

On the other hand, using very low sequence identity thresholds may dramatically reduce the number of available sequences in training and testing datasets. For example, the vacuolar sequence dataset used in MultiLoc (Höglund *et al.*, 2006) (see Chapter 3) normally contains 164 sequences with no redundancy reduction applied. In MultiLoc, the allowed maximum mutual sequence percent identity was taken as 80%, which reduces the number of vacuolar protein sequences to 103 (Table 1.3).

Chothia & Lesk (1986) demonstrated the relation between the divergence of

## 1.2 Sequence identity thresholds

---

Max allowed sequence identity	Number of vacuolar proteins
100%	164
80%	103
40%	36
30%	26
25%	23

Table 1.3: **Several allowed maximum mutual sequence identity rates versus the number of vacuolar sequences.** The vacuolar sequences refer to the same dataset used in MultiLoc and in Chapter 3. Generally, as the percent identity decreases, sequence dataset size shrinks.

sequence and structure in proteins. [Sander & Schneider \(1991\)](#) later showed that sequence identity does not correlate linearly with sequence homology. Namely, to avoid homologous pairs in a protein sequence dataset, the maximum percent sequence identity for long amino acid sequences must be smaller than that of relatively shorter sequences. That is, even a pairwise alignment with only 30% sequence similarity over a length of 60 residues may imply homology, but it does not if the alignment length is around 40. Generally, 30% sequence identity is regarded as a good threshold. However, as the percent identity threshold is decreased, there is a danger that there won't be sufficient number of sequences required for healthy training and testing. Therefore, whenever possible, I used 30% (Chapter 4) and when the number of sequences was critically low, 40% sequence identities (for instance, for the training and testing of Lokum: see Chapter 3). Compared to the other sequence identity thresholds used by the other mentioned multi-class predictors, the 40% maximum threshold used in Lokum is significantly lower (Table 1.2).

I used the CD-HIT ([Li & Godzik, 2006](#); [Li et al., 2001, 2002](#)) clustering algorithm to eliminate the existence of homologous sequences in the various sequence

datasets that are employed in Chapters 2, 3 and 4. As explained in [Li \*et al.\* \(2002\)](#), CD-HIT, in principle, uses the same basic sequence clustering algorithm originally developed by [Hobohm \*et al.\* \(1992\)](#) that guarantees the elimination of homologous pairs, but also uses some alternative heuristic strategies instead of directly performing pairwise alignments that could normally be quite CPU intensive. In CD-HIT, the minimum number of identical short substrings, called ‘words’, such as dipeptides, tripeptides and so on, shared by two proteins is a function of their sequence similarity ([Li & Godzik, 2006](#)).

### 1.3 Computational methodologies

Below I summarise and describe briefly the main computational methods I used (See Figure 1.1). These are, primarily, hidden Markov Models (HMM), motif finding by Nested Sampling, and SVMs. Motif finding and inference by Nested Sampling is the topic of Chapter 2 and explained there in more detail in the context of NestedMICA. Another area where Nested Sampling is used is Chapter 5 which introduces a prototype HMM based transmembrane topology predictor. Nested Sampling in Chapter 5 is used for HMM parameter optimisation. SVMs form the crux of Lokum, combining all the features used.

#### 1.3.1 HMMs

HMMs are useful for characterising sequentially changing behaviour, including signals such as speech or a string of amino acids, in a mathematically tractable way. An HMM is a stochastic finite automaton consisting of finite states. Each state in a model is associated with a probability distribution which usually has

## 1.3 Computational methodologies

---

multiple dimensions. An outcome or observable is said to be emitted from a state based on the emission probability distribution of that particular state. Transition probabilities reflect the transition frequencies between the states of a given model. They must be explicitly set in the model.

The three main problems HMMs can address are:

1. Computing likelihood (given a set of observables, find the corresponding probability of having that sequence). This problem is solved by the forward algorithm.
2. Viterbi decoding (given a model, find the most probable sequence of states which might have yielded a certain set of observables)
3. Model learning (inferring the model parameters, mainly that of the transition probabilities, that would best describe a set of observables) Parameter optimisation or learning can be achieved with the Baum-Welch algorithm which is an expectation maximisation (EM) procedure.

HMMs have been widely used in bioinformatics ([Durbin \*et al.\*, 1999](#)) particularly for computational gene prediction, secondary structure prediction, and modeling of protein families and domains. For example, gene prediction using HMMs involves the second and possibly the third tasks of the above HMM objectives. In the case of motif finding with NestedMICA, we use all three canonical objectives in [Chapter 2](#), for tasks including sequence and background likelihood calculation, model learning and fitting.

Duration HMMs ([Rabiner, 1989](#)), which were originally developed for alleviating problems in speech recognition, have many benefits in most applications

of HMMs in bioinformatics. Probability distributions of state occupancy can be represented by continuous probability density functions. Duration HMMs may utilise functions like Gaussian or Gamma distribution functions, instead of a decaying exponential in the case of classic HMMs. Thus, in practical terms, it is possible to ‘set’ the minimum (and in certain circumstances the maximum) number of times a model has to emit from within a certain state, once it enters that state. Figure 1.2 shows an example state occupancy probability plot for a duration-enabled HMM state with a pre-determined minimum number of self-transitions.

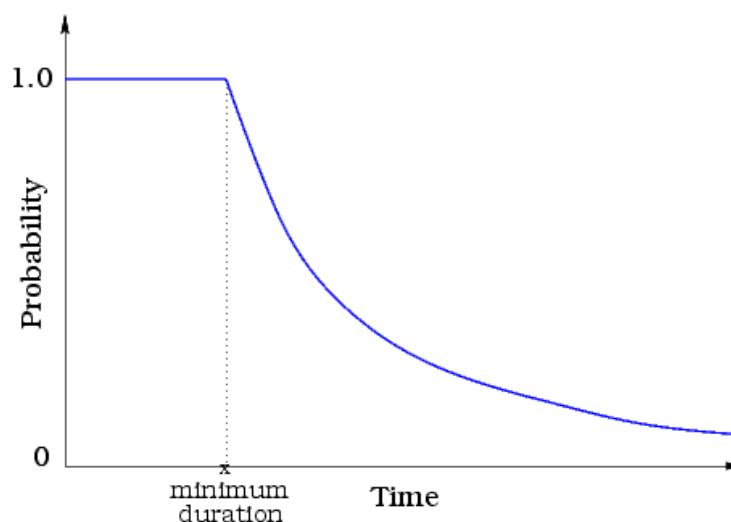


Figure 1.2: **Probability of staying in the same state in a minimum duration capable HMM state.** Normally the probability curve would be only a decaying function (of the form  $ae^{-x}$ ) from a maximum probability towards zero. However, in duration-enabled states, it has to spend at least a certain number of emission times in the same state before it starts to decay (corresponding to the horizontal part in the curve).

A profile HMM (Gribskov *et al.*, 1987) consists of multiple states connected in series, none of which has a self-transition but usually a single transition to the

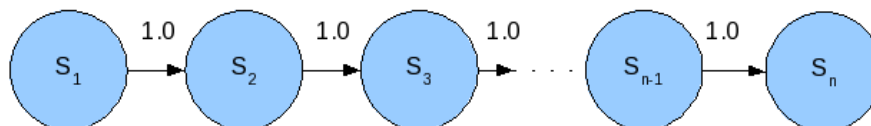


Figure 1.3: **Profile HMMs.** In this thesis, in what I call a “profile HMM” each state  $S_i$  has a single transition to the next state  $S_{i+1}$  with a duly set probability of 1.0, which makes them a way of representing position weight matrices (PWMs) in the context of HMMs.

next state. For instance, gapped multiple alignments can be represented as profile HMMs, in which case there is a need to add a “delete” and an “insert” state along with each “match” state (see [Durbin \*et al.\* \(1999\)](#) for use of HMMs in sequence alignment). However, throughout the thesis I will use the term “profile HMM” to indicate linearly constructed series of states, each of which has a transition probability of 1.0 to go to the next state, excluding the last state (Figure 1.3). In this regard, there is not much difference between such a construct and a sequence motif represented as a position weight matrix (PWM) where each fixed column has its own symbol distribution.

### 1.3.2 The general idea behind motif finding

Interesting motif regions and the remaining uninteresting parts of sequences can be represented as HMMs. These types of models can be referred to as sequence mixture models (SMM), as they contain states representing motifs as well as some prior models. Example of an SMM is the zero-or-one occurrences per sequence (ZOOPS) model which is the default strategy in most motif finders based on expectation maximisation ([Dempster \*et al.\*, 1977](#)), or Gibbs sampling ([Smith, 1987](#)), a typical example of which is the MEME ([Bailey & Elkan, 1995](#)) motif



discovery program.

NestedMICA differs from other ZOOPS models in that it does not perform a greedy search to discover the best single motif and then by masking it out focus on the next motif (if necessary). Instead, it considers different motifs at the same time and learns a model to best describe them based on independent component analysis (ICA) (Comon, 1994). In signal processing, ICA is a computational technique aiming to separate multivariate signals into independent subcomponents that constitute a given (generally noisy) signal. In linear, noiseless ICA:

$$x_i = a_{i,1}s_1 + \dots + a_{i,k}s_k + \dots + a_{i,n}s_n \quad (1.1)$$

where  $x$  represents the observed components vector, i.e:

$$x = (x_1, \dots, x_m)^T \quad (1.2)$$

with the constituent components, each having a weight  $a_{ik}$ , being:

$$s = (s_1, \dots, s_n)^T \quad (1.3)$$

The task is to be able to write  $s$  in terms of  $x$  :  $s = Wx$ , where  $W$  is some static transformation matrix. This situation is generally likened to the “cocktail party problem” which involves different people talking simultaneously in a room, and therefore one hears a constant random “noise”. If individual components of the observed “noise” are independent, then using ICA one can try to map the individuals in the room to what each person has said. In the case of motif ICA (MICA), motifs correspond to the individual voices in this example.

### 1.3.3 Inference by Nested Sampling

Inferring optimal parameters for probabilistic models is a difficult task, particularly when the number of model parameters becomes large. NestedMICA performs inference using Nested Sampling (Skilling, 2004), a robust Bayesian sampling method for model selection and parameter optimisation. Nested Sampling is a Monte Carlo inference strategy which can find globally good solutions to high-dimensional problems. Classical Monte Carlo methods work by moving a single state (*i.e.* set of parameters) around the problem’s parameter space, accepting or rejecting proposed moves depending on whether they increase or decrease the likelihood of the observed data. Nested Sampling is always applied to an ensemble of  $e$  different states, where the value of  $e$  is typically a few hundred. The process starts with an ensemble of states sampled uniformly from the prior.

Having sampled the states, they are then sorted in order of likelihood, and the least likely state is removed from the ensemble. To maintain the ensemble size, a new state is sampled, subject to the constraint that the new state must have a likelihood greater than that of the state it is replacing. Repeating this process many times means that nested samplers progressively move towards a small subset of the state space which contains high-likelihood states. This is somewhat analogous to simulated annealing methods where a temperature parameter is reduced to bring the model progressively closer to the posterior distribution, but nested sampling avoids the need to explicitly cool the model: progress towards high-likelihood states occurs automatically.

For each step of Nested Sampling, a certain fraction of state space is removed

### 1.3 Computational methodologies

---

from further consideration (since it contains states with likelihoods lower than the threshold). Over many steps, the fraction of prior mass that is removed from consideration at step  $t$  will tend towards <sup>1</sup>

$$W_t = \frac{1}{e} \left( \frac{e}{e+1} \right)^t \quad (1.4)$$

where  $e$  is the ensemble size. Since all the states which have been removed from consideration will have a likelihood of approximately  $L_t$ , the likelihood of the state which was removed at step  $t$ , the Bayesian evidence for the model,  $Z$ , can be estimated as:

$$Z = \sum_{t=1}^{\infty} W_t L_t \quad (1.5)$$

Clearly, it is possible to progressively accumulate an estimate of  $Z$  during the Nested Sampling process. The final estimate of  $Z$  can be used for model comparison purposes (for example, finding optimal parameters for the NestedMICA sequence model). NestedMICA also uses  $Z_t$ , the online  $Z$  estimate up to step  $t$ , to decide when to terminate the Nested Sampling process. Specifically, we terminate when:

$$\frac{1}{Z_t} L_t \left( \frac{e}{e+1} \right)^t < 0.01 \quad (1.6)$$

*i.e.* the likely increase of  $Z$  in future iterations is small compared to the current value. Formally, this may lead to premature termination if  $L$  increases dramati-

---

<sup>1</sup>Derivation of this formula is explained in the 4-page Nested Sampling illustrations by David MacKay at <http://www.inference.phy.cam.ac.uk/bayesys/box/nested.ps> (URL last visited in 2008)

cally late in the training process, but in practice we find that this simple criterion is effective for motif discovery.

### 1.3.4 Support vector machines

SVMs are one of the most popular classification and regression algorithms, and are applied in a variety of disciplines for tasks including signal processing, pattern and image recognition, and biological sequence analysis. The support vector (SV) algorithm is a generalised, non-linear form of the “generalised portraits” concept developed by Vapnik in the 1960s (Vapnik & Lerner, 1963).

SVMs can be thought of as classifiers that try to maximise the geometric margin separating data points from different classes. Points are actually multidimensional feature or attribute vectors which are mapped by some selected function into some other mathematical space, where it would be more convenient to perform the required tasks such as classification or regression. This new space usually has a larger number of dimensions than the actual feature space, which in turn increases the separability of data.

This separability is determined by the VC (Vapnik-Chervonenkis) dimension of the model, which can be considered an upper theoretical limit for the set of points a classifier can “shatter” in that space. To “shatter” some given points belonging to different classes, SVMs “draw” hyperplanes near the support vectors of each class. SVs are those that are near the class boundaries, and contribute more than the other points in shaping the hyperplanes. Then, an optimal separating hyperplane is selected such that it maximises the geometric distance between any two drawn hyperplanes that define class zones. Although defining hyperplanes

### 1.3 Computational methodologies

---

associated with each class can in principle solve the problem of correctly assigning new, unseen data that is similar to the training data into one of the classes, for more “difficult” test points that lie between any two zone-determining hyperplanes, assignments can be done according to their distances to the maximum margin hyperplane.

A dot-product function can be used in simple problems for data mapping, but using kernel functions allows non-linear hyperplanes to be created. Apart from linear kernels, the most commonly used kernels are polynomial (Eq 1.7), radial-basis function (RBF)(Eq 1.8), and the sigmoid (Eq 1.9):

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d \quad (1.7)$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) \quad (1.8)$$

$$k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa\mathbf{x} \cdot \mathbf{x}' + c) \quad (1.9)$$

More information regarding SVMs can be found in the excellent tutorials of [Burges \(1998\)](#) and [Smola & Scholkopf \(1998\)](#), and also from SVM-dedicated web sites such as:

- <http://www.support-vector.net/>
- <http://www.support-vector-machines.org/>.

Examples of popular SVM implementations that are free for use in academic studies are *libsvm* ([Chang & Lin, 2001](#)), *SVM<sup>light</sup>* ([Joachims, 1999](#)), and another

### 1.3 Computational methodologies

---

*libsvm* derivative, BSVM (Hsu & Lin, 2002). In Lokum we used both the C and Java implementations of *libsvm* (Chang & Lin, 2001), version 2.85 (see Sections 3.2.6 and 3.5).

Artificial Neural Networks (ANNs), similar to SVMs in terms of their goal and function, are widely used classifiers. ANNs differ substantially from SVMs in that their proposed solutions could correspond to some local maxima. As C. Burges put it in his SVM tutorial (Burges, 1998), “They (SVMs) differ radically from comparable approaches such as neural networks: SVM training always finds a global minimum, and their simple geometric interpretation provides fertile ground for further investigation”.

# Chapter 2

## NestedMICA as an *ab initio* protein motif finder

1

### 2.1 Background

Discovering linear sequence motifs common to a set of protein sequences has long been an important problem in biology. It is possible to check if a set of proteins contain a known sequence motif by searching protein motif or domain databases. Databases including Pfam (Bateman *et al.*, 2004), eukaryotic linear motif database (ELM) (Puntervoll *et al.*, 2003), Prosite (Hulo *et al.*, 2006) and ScanSite (Obenauer *et al.*, 2003) contain sequence motifs and domains in the form of regular expressions or profile HMMs. Obviously, one cannot use these resources to discover a novel or unannotated sequence motif that is suspected to be a common feature in a given protein set. While new protein domains such as

---

<sup>1</sup>This chapter was partly published in BMC Bioinformatics in January 2008 (Doğruel *et al.*, 2008), by the author of this PhD thesis (MD), Dr. Thomas Down (TD), and finally my thesis supervisor Dr. Tim Hubbard (TH). Authors' contributions are as follows: TH and MD conceived this work, MD and TD modified the NestedMICA code, MD performed the tests and wrote the manuscript.

## 2.1 Background

---

those contained in Pfam can be defined from alignments of evolutionarily related sequences, the identification of short sequence motifs, potentially shared between proteins that appear evolutionarily unrelated, is much harder.

To tackle this problem, several multiple alignment approaches (Hertz & Stormo, 1999; Zaslavsky & Singh, 2006) have been proposed. One such tool, Dilimot (Neduva & Russell, 2006), is a recent protein motif search tool aiming at finding relatively short overrepresented motifs by aligning only sequence regions that are likely to contain a linear motif. It filters out regions including globular domains and coiled-coil regions which are reported or predicted by some other algorithm, before searching for known motifs in several protein databases such as PFAM, and finally uses a pattern search program, TEIRESIAS (Rigoutsos & Floratos, 1998) to find overrepresented matches. TEIRESIAS, an *ab initio* program that is not based on database look-up, can list frequently repeating character-based patterns that include gaps, from a given sequence set. Patterns can include one or two events separated by wild-card characters, as in AT..G (Burgard *et al.*, 2001). Another similar and robust amino acid pattern search tool is SLIMFinder (Edwards *et al.*, 2007) in which short protein motifs are built by combining dimers into longer patterns, retaining only those motifs occurring in a sufficient number of unrelated proteins. Motifs with fixed amino acid positions are identified and then combined to incorporate amino acid ambiguity and variable-length wildcard spacers. Dilimot, TEIRESIAS and SLIMFinder report results as regular expressions. There are also other algorithms in the non *ab initio* motif finding category, using evolutionary or structural information, which are specifically designed to predict DNA-binding regions in protein sequences (Ahmad & Sarai, 2005; Hwang



## 2.1 Background

---

*et al.*, 2007; Kuznetsov *et al.*, 2006). However since the MEME tool was developed (Bailey & Elkan, 1995) and provided a way to carry out *ab initio* protein motif finding, returning a set of Position Weight Matrices (PWMs) rather than regular expressions, not many multi-purpose sequence-based probabilistic motif finders have been developed, despite there being numerous tools for finding motifs in DNA. Examples to other well known DNA motif discovery tools are SeSiMCMC (Favorov *et al.*, 2005), AlignACE (Hughes *et al.*, 2000), ANN-Spec Workman & Stormo (2000), Weeder (Pavesi *et al.*, 2004), and YMF (Sinha & Tompa, 2003).

NestedMICA (Down & Hubbard, 2005) is a probabilistic motif discovery algorithm which uses a new Monte Carlo inference strategy called Nested Sampling (Skilling, 2004). Written in the Java programming language as an open source application, NestedMICA uses Biojava libraries (BioJava, 2007). It has been successfully used for transcription binding site and large-scale promoter motif discovery (Down & Hubbard, 2002). In this manuscript, I extend the application of NestedMICA to finding motifs in protein sequences and compared it with the popular program MEME using both biologically-authentic and synthetic test data sets. I chose to compare NestedMICA with MEME, because the output of MEME is motifs in the form of PWMs, making comparison possible. MEME is also an *ab initio* method and uses probabilistic models like NestedMICA.

To evaluate the performance of the two methods I have performed various spiking tests in which some test motifs generated from protein domain alignments were spiked into a set of protein sequences, as described in the Methods. This assessment procedure is similar to the approach followed in a previous transcription binding site motif discovery programs comparison by Tompa *et al.* (2005).

NestedMICA has also been assessed by testing its ability to find a subcellular localisation motif in datasets known to contain a specific localisation signal.

## 2.2 Materials and methods

### 2.2.1 NestedMICA

NestedMICA is a probabilistic motif inference method based on a generative sequence model. The model has three sets of parameters: firstly, a background model which represents all the non-motif parts of the input sequences; second, a set of position-weight matrices which represent the motifs themselves; finally, a binary matrix (the occupancy matrix) whose elements specify whether a given motif should be considered when modeling a given input sequence. The background model is built in advance and held constant during motif inference, while the motifs and occupancy matrix are updated to fit the supplied data. NestedMICA uses the Nested Sampling strategy (Skilling, 2004) to update both of these sets of parameters.

The implementation of NestedMICA's `nminfer` program can be split into two major parts: code that calculates the likelihood of some sequences under the generative model, and code which implements the Nested Sampling process. The Nested Sampling code makes few assumptions about the internal structure of the model (and could potentially be used to perform inference of quite different models), so I consider these two components separately.

NestedMICA was designed completely in an object oriented and modular manner that allows one to plug in a very different model without touching the trainer code: Similarly, the likelihood calculators do not know anything about Nested

Sampling (and could potentially be used in another training framework). Below, sequence models, likelihood calculation, nested sampling, and finally the implementation are discussed.

### 2.2.1.1 The NestedMICA sequence model

NestedMICA relaxes the constraints of the ZOOPS model (see 1.3.2) slightly by allowing a given motif to appear multiple times in the same input sequence. To calculate the likelihood of a given sequence, NestedMICA first consults to appropriate row of the occupancy matrix to determine a (possibly empty) subset,  $M$ , of the complete motif set which applies to this sequence. In the case where  $M$  is empty, the likelihood of the sequence is simply its likelihood under the background model (see below). When  $M$  is non-empty, NestedMICA sums over all possible configurations of motif occurrences along the sequence, filling in any gaps using the background model. This is performed using a dynamic programming recursion which gives the likelihood,  $L_n$  of all paths up to a given point in the input sequence,  $n$  as:

$$L_n = (1 - t)B_{n-1}L_{n-1} + \frac{t}{|M|} \sum_{m \in M} m(S_{n-|m|+1}^{n-1})L_{n-|m|} \quad (2.1)$$

where  $|M|$  is the number of motifs selected by the occupancy matrix,  $|m|$  is the length of weight matrix  $m$ ,  $B_n$  is the probability that the sequence symbol at position  $n$  was emitted by the background model,  $m(S_i^j)$  is the probability that the sequence from  $i$  to  $j$  was emitted by the weight matrix  $m$ , and  $t$  is a transition probability specifying the estimated density of motifs in the sequence.

We initialise  $L_0 = 1$  then apply the above formula recursively along the length

of the input sequence until the final position is reached, giving a likelihood for the complete sequence.

In principle, any background model could be used with this formulation. In practise, I choose to use a mosaic background (Down & Hubbard, 2005) which admits the possibility of several different classes of background sequence, each of which is modeled using a low-order Markov chain (*i.e.* within a given class, the probability of observing a particular symbol at position  $n$  depends on the symbols observed at a fixed number of previous positions). The mosaic model is implemented as a fully connected HMM (transitions are allowed between any pair of classes).

To calculate  $B_n$ , NestedMICA first applies the standard posterior decoding algorithm (Durbin *et al.*, 1999) to find  $P_{hn}$ , the posterior probability that the symbol at position  $n$  in the input sequence was generated by state  $h$  of the background model  $H$ . We can then calculate  $B_n$  as:

$$B_n = \sum_{h \in H} P_{hn} h(S_n) \tag{2.2}$$

(*i.e.* summing over any remaining uncertainty in which background class is used at  $n$ ). Note that when the Markov chain order,  $o$  is greater than zero, the probability of observing a given symbol,  $h(S_n)$ , depends on  $o$  previous symbols in the sequence. This means that it is not possible to exactly calculate  $B_n$  where  $n \leq o$ . We choose to ignore the first  $o$  symbols in the input sequence (except for background calculation purposes) in order to avoid any edge effects.

### 2.2.1.2 Implementation of NestedMICA

The NestedMICA `nminfer` program is based around a fairly general implementation of the Nested Sampling strategy, which can be applied to any probabilistic model. This code takes three inputs: a data set (*i.e.* a set of sequences), some code to calculate the likelihood of the dataset given a model state (*i.e.* an implementation of the likelihood function given above), plus a set of “sampling” operations which perturb a state and can be used to move around state space.

Each state consists of two sets of parameters: a set of motif weight matrices, and an occupancy matrix specifying whether the motifs appear in the input sequence set. Most of NestedMICA’s sampling moves are applied to one randomly selected weight matrix (WM):

- making a small perturbation to one column of a weight matrix, by slightly increasing or decreasing one of the weights, then renormalizing so they still sum to 1.
- replacing a WM column with a new one, sampled from the prior.
- removing a column in one end of a WM while adding another one to the other end.
- adjusting motif length, by adding or removing a column from either end.

In addition, it is necessary to resample the occupancy matrix. In principle, a straightforward and valid sampling move would be to simply flip the state of one randomly-selected element in the occupancy matrix. In practise, NestedMICA

## 2.2 Materials and methods

---

tests multiple occupancy matrix moves at the same time, since this improves performance when running on multi-processor systems.

Finally, it is necessary to place a prior over the state space. NestedMICA uses a simple non-informative prior for the Weight Matrix motif models: a uniform prior over weight-matrix space with a constraint that extremely low weights are forbidden. The lower limit is specified by the `-minClip` parameter and is typically  $10^{-7}$  for amino acid, and of the order of  $10^{-3}$  for dna input. We also place a non-informative prior on the occupancy matrix, although if there is some prior knowledge about the frequency of the target motif in the dataset, this can be specified using the `-expectedUsageFraction` option.

The main challenge when implementing nested samplers is to sample uniformly from the prior while respecting the likelihood constraint. In practice, this is usually solved by duplicating a randomly-selected state from the ensemble then using classical (single-state) Monte Carlo strategies to move the duplicate state. NestedMICA uses a straightforward Metropolis-Hastings approach for prior sampling. Further information on the use of this strategy is available in the original publication of NestedMICA ([Down & Hubbard, 2005](#)).

Rather than storing the weight matrix in its traditional form as a list of probability distributions over an alphabet, in NestedMICA it is stored as a circular buffer of distributions that is slightly larger than the longest motif being modeled, with the addition of an offset parameter (where the motif starts in the buffer) and a length parameter. The nice thing about this representation of motifs is that it is possible to extend the motif in either direction when length is needed to be sampled, up to the size of the circular buffer.

### 2.2.2 Adding protein support to NestedMICA

I made several changes to NestedMICA in order to support protein motif discovery. Firstly, I added support for loading and analysing protein sequences (enabled with the “-alphabet PROTEIN” switch). The inference strategy remains identical to that previously described (Down & Hubbard, 2005). However, the dimensionality of the protein motif discovery problem is much higher than in nucleic acids: a DNA motif model has three free parameters per position, while a protein motif has 19. To compensate for this difference, I found that a rather larger ensemble of models in the Nested Sampling process was required than for DNA. Having found an optimal ensemble size by performing a systematic parameter sweep test, I altered this to be the default ensemble size when running the program in protein mode. Unless set otherwise by the user, it is automatically set to either 4000 divided by number of target motifs, or set to a minimum of 1000, in case the division would be less than 1000.

Another important difference between the protein-capable version and the previous version of NestedMICA is the way distribution probability initialisation is performed in setting up the amino acid probability distributions for each background mosaic class. Starting off with flat probability distributions in all the mosaic classes of a given background as in the DNA case was not ideal for protein sequences, as I observed a minimal learning rate with these equal initial states. Instead, a semi random, semi actual input-based initialisation was preferred: the distributions were initialised such that they directly reflect the amino acid distributions of the actual input data, except, these numbers were slightly changed

randomly by a certain margin for the training to learn and converge faster.

Since the initial publication of NestedMICA (Down & Hubbard, 2005), an important extra feature was added of automatically optimising a motif's length within a user-specified motif length range. NestedMICA treats the motif length as another free parameter of the motif model, and optimises it using the same Nested Sampling strategy as for all the other parameters. Another change in the new version is that, if no background model is provided by the user, NestedMICA uses a basic, zero-order background model which is trained on the fly from the user supplied input sequences.

Further information regarding the parameters used in motif finding can be found in the user manual at the NestedMICA web site:

<http://www.sanger.ac.uk/Software/analysis/NestedMICA/>

### 2.2.3 Program output and sequence logos

NestedMICA reports discovered motifs as PWMs which can be viewed as sequence logos by an accompanying motif-viewer tool. In a single NestedMICA protein motif logo, each column has a maximum information content of 4.32 bits ( $\log_2 20$ ), and amino acid letters are coloured according to their general physical and chemical properties, as depicted in Figure 2.2

As opposed to majority of motif finders, NestedMICA does not report any significance measures such as E-values, or entropy scores, as these values could be quite unreliable. All these scores are calculated based on the idea that a motif finder has picked up a real motif, which obviously cannot always be true. The recent publication by Ng *et al.* (2006), discusses in detail why using such scores



could lead to undesirable results.

### 2.2.4 Background model training

Probabilistic motif finding tools usually employ background models to represent sequence regions where ideally no motif of interest exists. In most cases, however, these programs use a homogenous background model, assuming that all non-motif portions of the sequence can be represented using a single amino acid frequency distribution. In reality, protein sequences are generally composed of different functional domains which can be chemically biased towards certain compositional forms. In addition, protein sequences are very likely to carry different sequence signals responsible for various molecule-recognition and binding related tasks. NestedMICA uses non-homogenous (“mosaic”) background models which subdivide the background sequences into several classes. Each class is modelled as a Markov chain. The order of the chain (i.e. the number of previous symbols on which the probability distribution for the next observed symbol is conditioned) can be set to an arbitrary value, but for protein sequence analysis I recommend only using zeroth or first-order background models, since higher order models will have an extremely high parameter count and will be hard, if not impossible, to parametrise effectively.

A built-in background likelihood estimation procedure in NestedMICA (called “nmevaluatebg”) allows an optimal background model architecture to be found for a given set of sequences. A NestedMICA background model can be of any order Markov chain and consist of an arbitrary number of mosaic classes. As a good representative sequence set, I used the pTarget protein subcellular locali-

sation dataset (Guda & Subramaniam, 2005) for background model parameter optimisation (Figure 2.1). This is mainly because it includes different types of proteins from different subcellular localisations, eliminating the chance of some strong domain and localisation signals to dominate the background model training and evaluation. Furthermore, I reduced the sequence identity of the set from 95% down to a maximum of 40% by using the CD-HIT (Li & Godzik, 2006) clustering software to have a total of 7437 eukaryotic proteins, which had an average sequence length of 522. For evaluation purposes, 6000 of these were used to train several different background models with different parameters, while the remaining sequences were used to test how well a certain background model represented them. As Figure 2.1 shows, using order-1 probabilities, where the compositional probability of a certain residue depends only on a single adjacent residue, performs better than a zero-order model. Moreover, likelihood for the test sequences increased monotonically with the number of mosaic classes. Training a multi-class higher-order background requires sufficient sequence data in order to prevent a possible over-fitting of the background. For example, using a first order, 6-classes model corresponds to having a total of 2400 different amino acid distributions.

### 2.2.5 Testing NestedMICA’s performance

In order to get a better understanding of NestedMICA’s protein motif finding capabilities and limits, a number of motif spiking experiments were performed using synthetic and biological motifs, similar to the approach previously used by Down & Hubbard (2005). In a motif spiking test, a number of short amino acid sequences are generated according to the weight matrix distribution probabilities

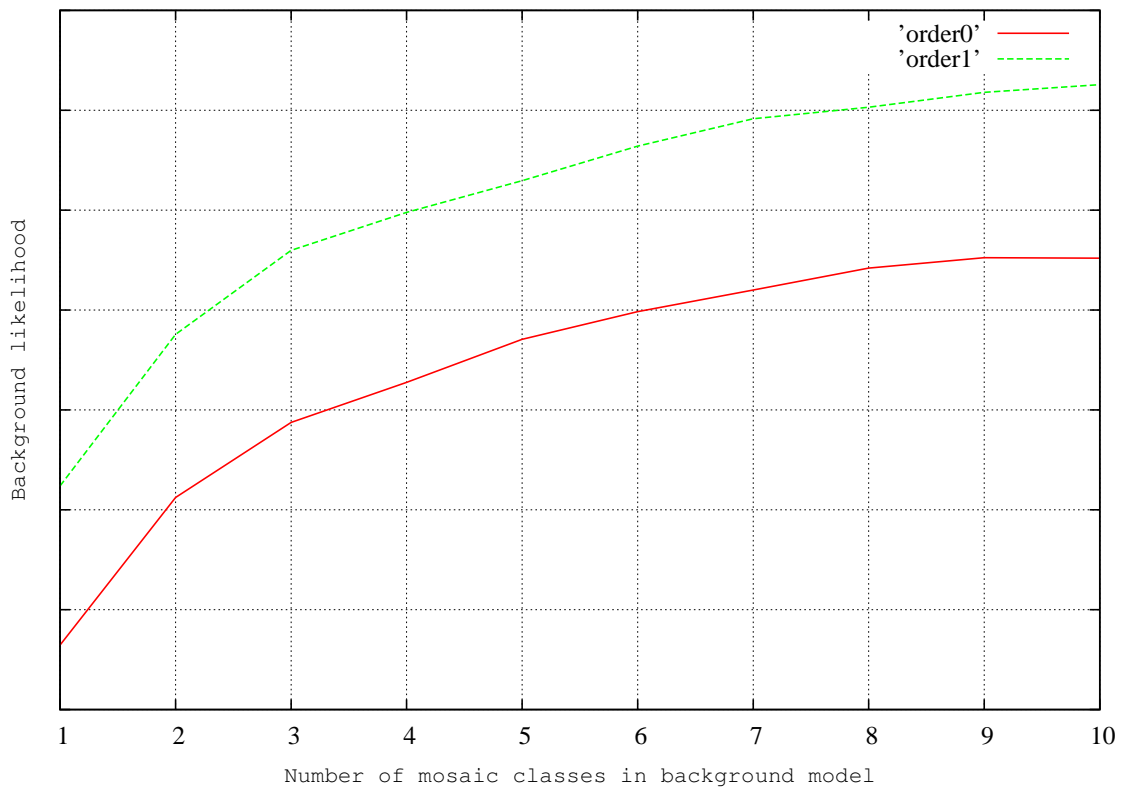


Figure 2.1: **Likelihood curve for different number of mosaic classes.** The x-axis represents the total number of mosaic classes in the tested background model architecture. The logarithmic y-axis corresponds to a likelihood measure that can take arbitrary values, of how well a background model represents the given sequence set. The red line represents a zero-order while the green one represents a first-order background model.

of a given motif. These motif-resembling short peptides are then inserted at random positions into a set of sequences. The program under test is then applied to the set of sequences to predict a set of motifs. Finally, the predicted candidate motif set is compared with the original test set to assess the performance of the program in recovering the spiked motifs. MEME PWMs were converted into NestedMICA sequence logos for easier comparison.

To evaluate how similar a reported motif is to the original one, I used Cartesian motif-motif distances. The Cartesian motif distance metric is the sum of individual Cartesian distances calculated for each motif position, between corresponding pairs of the 20 amino acid probabilities from both motifs. For a motif to be considered as recovered with a reasonable precision, I used an empirically set threshold for the maximum allowed Cartesian motif distance normalized for the original motif length. Motifs showing an average deviation per position of more than 0.3 of Cartesian motif distance were considered as false discoveries.

For each motif, in addition to reporting Cartesian motif distances, I calculated sensitivity (Equation 2.3) and specificity (Equation 2.4) values:

$$SN = \frac{TP}{TP + FN} \quad (2.3)$$

$$SP = \frac{TP}{TP + FP} \quad (2.4)$$

Matthew's Correlation Coefficient (MCC) (Matthews, 1975), shown in Equation 2.5, values were calculated, too, to show a PWM's scanning power as in Kiemer *et al.* (2005):

$$MCC = \frac{TP \quad TN - FN \quad FP}{\sqrt{(TN + FN)(TP + FN)(TN + FP)(TP + FP)}} \quad (2.5)$$

where TP, FP, FN, TN stand for true positives, false positives, false negatives and true negatives, respectively.

One advantage of using MCC in a PWM evaluation is that for random motif predictions MCC tends to be around zero, while for a perfect scanning performance it will have a maximum value of 1. On the other hand, depending on the choice of a score threshold, even for an irrelevant or weak motif one can get a sensitivity of 1, for instance, while the corresponding specificity value could be as low as 0.5, if the number of sequences in both datasets are equal. In such cases, MCC will tend to be very low, reflecting the random prediction.

To calculate these measures of motif scanning performance, first, I spiked every sequence in the test dataset with a particular motif, then I scanned a reported motif both in the spiked and original datasets to see how many motif instances would be correctly or falsely predicted in both datasets. For each individual test case, I picked a threshold score that maximises the corresponding MCC value, after trying a range of different score thresholds systematically incremented in each iteration to compute sensitivity, specificity and MCC values. I calculated these values not only for motifs reported by the programs I assessed, but also for the original test motifs. I did this because values measuring the scanning performances of recovered motifs should be considered relative to those of the original motif. A more objective and absolute metric of motif recovery is the Cartesian motif distance, which is the sum of probability differences in corre-

## 2.2 Materials and methods

---

sponding columns of any two compared motifs. For example, a test motif which contains only a small number of strongly conserved residues cannot be expected to have a good scanning performance in identifying all spiked motifs, because the motif tolerates too much sequence variation. Therefore judging the performance of a motif discovery tool based on only such sensitivity/specificity measures is inadequate, since a motif tool should find a weak motif from a set of spiked data, if the original motif is a weak one, too. The sensitivity/specificity of this type of less conserved motifs would be relatively low, and not reflect or reward a program's ability to have discovered such a difficult motif. Therefore, I report MCC of the original test motifs primarily as a measure indicating how difficult a motif is to recover by a motif discovery program, and I report Cartesian motif distances with the purpose of indicating how good the program is in that task. For instance, even an MCC value of 0.65 would still be good for a motif found by a program, if the corresponding real test motif did not have a much better MCC.

To generate test motifs for the program's assessment, I used conserved blocks of several ClustalW multiple alignments of sufficiently large number of Swiss-Prot (Bairoch & Apweiler, 1996) proteins. These proteins feature arbitrarily chosen Prosite (Hulo *et al.*, 2006), or PFAM domain entries. Segments from these domains' alignments were converted into PWMs to obtain 3 sets of 7 test motifs of varying lengths between 3 and 9. The 21 test motifs used in the evaluations are available for download at the NestedMICA home page.

As a dataset to carry out the spiking tests on, I used 438 whole-length cytoplasmic protein sequences obtained from the redundancy-reduced non-plants version of the TargetP (Emanuelsson *et al.*, 2000) subcellular localisation dataset.

Having an average sequence length of 582, this dataset does not include any homologous proteins, after a filtering process performed as suggested by [Hobohm \*et al.\* \(1992\)](#). Both NestedMICA and MEME were run with the default options. Note that, NestedMICA’s default parameters differ from those used in DNA motif finding. Both NestedMICA and MEME require a target motif length interval, and no matter what the actual spiked motif’s length was, for all of our spiking tests this was set to be between 3 and 15.

The background model used in the spiking tests was trained from the same cytoplasmic sequence dataset. The similar background likelihood analysis that was performed on another set ([Figure 2.1](#)) suggested that there would be no significant gain in likelihood when using a model with more than 4 mosaic classes for this particular small dataset. Therefore, a first order background model containing 4 mosaic classes was used in the tests.

Finally, for the evaluation of the program’s assessment in subcellular localisation motif recovery, which was performed using sequences of different lengths, I used the ER dataset of a multi-class protein subcellular localisation predictor, MultiLoc ([Höglund \*et al.\*, 2006](#)). This dataset contains 198 homology-reduced, eukaryotic ER proteins.

## 2.3 Results and discussions

### 2.3.1 Protein sequence background model

The first step in using NestedMICA is the generation of a background model to represent the uninteresting parts of sequences that do not contain motifs of interest (see methods). From a series of tests I concluded that different sets of protein

sequences vary in complexity and composition too much to develop a generic background model. Most of the time, training a dedicated background model for each protein dataset is the best way to maximise performance and sensitivity. Prior to motif finding, sequence likelihood analysis must be performed to test a variety of background models and select the optimal one. Figure 2.1 shows one such likelihood curve performed on a set of cytoplasmic proteins. Generally, if there is sufficient data to perform a proper training, using order-1 background models proved to be better than order-0 models for proteins. As far as the number of mosaic classes is concerned, a class number should be picked that falls on the corresponding likelihood curve before it starts to saturate or drop, regardless of whether it increases at a later stage.

### 2.3.2 Performance vs. motif abundance

I used 3 different motif sets each containing 7 motifs of lengths ranging from 3 to 9 amino acids. Instances of each of the motifs depicted in Figures 2.2, 2.3 and 2.4 (for motif sets 1, 2 and 3, respectively) were separately spiked into the cytoplasmic dataset (see Section 2.2.5). The 21 motifs were inserted into the sequences at different frequencies (10, 20 and 30%), allowing us to test motif discovery software under different conditions of motif abundance. Generally, performance for both NestedMICA and MEME increased with increasing abundance rate of the inserted motif.

Each of these three figures shows a set of tests performed at different motif abundance rates with the original test motifs, along with the corresponding motifs found by both NestedMICA and MEME. For each motif reported by NestedMICA



## 2.3 Results and discussions




































Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
	10	0.753		0.57 0.830		
	20			0.34 0.830		
	30			0.33 0.830		
	10	0.856		2.70 0.153		
	20			3.72 0.015		
	30			0.72 0.537		
	10	0.749		1.58 0.499		
	20			0.50 0.699		
	30			0.55 0.723		
	10	0.815		5.67 0.011		
	20			0.71 0.648		
	30			0.70 0.653		
	10	0.918		5.10 0.015		
	20			0.78 0.816		
	30			0.68 0.795		
	10	0.993		0.80 0.926		
	20			0.52 0.935		
	30			0.52 0.935		
	10	0.990		5.21 0.118		
	20			1.00 0.784		
	30			0.93 0.795		

Figure 2.2: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set1. Motifs in this set were obtained from several Pfam domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well. In NestedMICA protein sequence logos, hydrophobic residues are represented in orange, polar and hydrophilic ones in green, acidic ones in pink, and finally basic amino acids are depicted in blue.

## 2.3 Results and discussions

Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
GPF	10	0.850		0.84 0.726		
	20			0.35 0.726		
	30			0.26 0.850		
KYGV	10	0.822		4.50 0.018		
	20			0.69 0.693		
	30			0.32 0.693		
ATCP	10	0.921		0.76 0.931		
	20			0.50 0.936		
	30			0.29 0.884		
WYKQ	10	0.884		0.81 0.911		
	20			0.51 0.858		
	30			0.47 0.886		
QDRDK	10	0.943		1.03 0.939		
	20			0.64 0.939		
	30			0.45 0.948		
YIRLP	10	0.830		5.45 0.012		
	20			0.71 0.810		
	30			0.61 0.812		
REGLYRSG	10	0.941		6.85 0.016		
	20			0.99 0.927		
	30			0.67 0.932		

Figure 2.3: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set2. Motifs in this set were obtained from several Prosite domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well.

## 2.3 Results and discussions

Original motif	Abundance	MCC for original	NestedMICA	Distance & MCC for NestedMICA	MEME	Distance & MCC for MEME
	10	0.753		1.11 0.539		
	20			0.14 0.753		
	30			0.10 0.753		
	10	0.856		3.22 0.037		
	20			0.53 0.758		
	30			0.37 0.730		
	10	0.749		0.63 0.659		
	20			0.58 0.673		
	30			0.34 0.708		
	10	0.815		1.11 0.750		
	20			0.58 0.780		
	30			0.55 0.761		
	10	0.918		0.99 0.857		
	20			0.77 0.873		
	30			0.43 0.890		
	10	0.993		0.59 0.990		
	20			0.34 0.990		
	30			0.23 0.990		
	10	0.990		1.10 0.988		
	20			0.63 0.993		
	30			0.56 0.990		

Figure 2.4: Motifs recovered by NestedMICA and MEME in the single-motif spiking tests, for motif set3. Motifs in this set were obtained from several Pfam domain entries. For each original test motif used in the motif spiking tests, the 3 tested abundance rates are shown in the next column. For motifs recovered by NestedMICA (fourth column) and MEME (sixth column) the Cartesian distance to the original test motif and the MCC value obtained when the motif is used for sequence scanning are shown. For comparison purposes, the MCC values of the original test motifs are shown as well.

## 2.3 Results and discussions

---

and MEME, its Cartesian distance from the corresponding original motif is given. As Tables 2.1 and 2.2 summarise, low abundance motifs and short motifs were more difficult to recover for MEME, even if they had a high information content. For example, out of the maximum 4.32 bits per position, the average information content per position was 3.96 bits (91.5%) for motif of length 3 in set 2, while it was 3.68 bits (85.2%) for motif of length 4 in the same motif set (Figure 2.3). Both could not be recovered by MEME at the tested 10, 20 and 30% abundance rates. The motif of length 3, for example, could only be recovered correctly by MEME when it was present in at least 80% of the sequences (data not shown). In contrast, the same motif was recovered by NestedMICA when present in only 10% of the sequences. NestedMICA did not miss any of the 21 motifs when they were present at 30% abundance. It also correctly recovered 95.2% and 61.9% of them when the motif abundance rate was 20%, and 10%, respectively (Table 2.2).

Spiked in(%)	Set 1		Set 2		Set 3	
	NestedMICA	MEME	NestedMICA	MEME	NestedMICA	MEME
10	3	0	4	0	6	2
20	6	1	7	4	7	3
30	7	3	7	5	7	4

Table 2.1: **Motif recovery performance for NestedMICA and MEME for individual test sets.** Numbers shown correspond to the correctly recovered number of motifs for each test set, each of which contains 7 motifs, for the single-motif spiking tests. Motifs recovered for set 1, 2 and 3 can be seen on Figures 2.2, 2.3, and 2.4, respectively. A motif is considered as correctly recovered if the average Cartesian distance per residue position between the recovered motif and the original motif that was spiked is  $< 0.3$  (see Section 2.2.5).

In addition to Cartesian motif distances, measuring the similarity between the recovered motif and the original, the performance of the motifs in finding motif

## 2.3 Results and discussions

---

instances when scanning test sequences is indicated by Matthew’s Correlation Coefficient (MCC) (Matthews, 1975) values (Figures 2.2, 2.3 and 2.4). The MCC is a single measure that captures performance over a range of sensitivity and specificity values (see methods). Raw sensitivity and specificity values are given Tables 2.3, 2.4 and 2.5 for all three motifs sets. These measures have been used to evaluate the scanning performances of the original and reported motifs, by testing spiked datasets (independent of the spiked datasets used for training) where each sequence contains an instance of a particular motif. I provide the MCC values for the original test motifs, too, for better interpretation of the MCC values given with the motifs reported by both programs. Having relatively lower sensitivity / specificity values, and hence a lower MCC, does not necessarily mean that a program is not doing well in finding a certain motif, but in certain cases it can indicate that the target motif is a weak one and therefore more difficult to recover. MCC values for the original motifs were calculated in a similar way to the others, i.e., by spiking every sequence in the background test dataset with the generated instances of a particular motif, and then scanning the spiked dataset with the original motif to see how many motif hits would be found using a range of score

Motif abundance(%)	Total correct (%)	
	NestedMICA	MEME
10	61.9	9.5
20	95.2	38.0
30	100.0	57.1

Table 2.2: **Total motif recovery performance summary for NestedMICA and MEME.** Percentages of correctly recovered motifs are given for the 3 motif abundance rates tested, considering all 21 test motifs from three of the sets.

thresholds (see methods).

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.988	0.855	0.995	0.501
3	20	0.988	0.855	0.995	0.501
3	30	0.988	0.855	0.995	0.501
4	10	0.811	0.545	0.197	0.506
4	20	0.995	0.501	0.197	0.506
4	30	0.847	0.728	0.197	0.506
5	10	0.487	0.914	0.592	0.507
5	20	0.753	0.921	0.592	0.507
5	30	0.782	0.921	0.592	0.507
6	10	0.950	0.501	0.978	0.501
6	20	0.849	0.808	0.978	0.501
6	30	0.703	0.913	0.978	0.501
7	10	0.995	0.501	0.995	0.501
7	20	0.890	0.923	0.995	0.501
7	30	0.823	0.958	0.818	0.950
8	10	0.957	0.968	0.856	0.507
8	20	0.959	0.976	0.959	0.976
8	30	0.971	0.964	0.964	0.969
9	10	0.974	0.514	0.990	0.502
9	20	0.835	0.938	0.995	0.501
9	30	0.875	0.915	0.851	0.939

Table 2.3: **Sensitivity (SN) and specificity (SP) values for motifs of Set 1, reported by NestedMICA and MEME in the single-motif spiking tests.** Length refers to number of residue positions in motifs.

### 2.3.3 Performance with multiple motifs

Individual protein sequences may contain multiple different motif of interest. For example, proteins targeted into the endoplasmic reticulum (ER) by an N-terminal Signal Peptide (SP) sequence are maintained in the ER if they have also a [KH]DEL retention signal on their C-terminus. After determining the ability of

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.770	0.933	0.990	0.502
3	20	0.770	0.933	0.990	0.502
3	30	0.950	0.904	0.995	0.501
4	10	0.930	0.503	0.942	0.504
4	20	0.664	0.986	0.988	0.501
4	30	0.842	0.850	0.988	0.501
5	10	0.978	0.953	0.995	0.501
5	20	0.954	0.980	0.995	0.501
5	30	0.974	0.914	0.986	0.895
6	10	0.921	0.987	0.935	0.502
6	20	0.866	0.984	0.918	0.958
6	30	0.914	0.969	0.871	0.976
7	10	0.947	0.990	0.866	0.503
7	20	0.942	0.995	0.952	0.978
7	30	0.962	0.985	0.957	0.964
8	10	0.959	0.501	0.974	0.504
8	20	0.873	0.931	0.861	0.940
8	30	0.873	0.933	0.851	0.947
9	10	0.995	0.501	0.998	0.501
9	20	0.940	0.985	0.935	0.975
9	30	0.938	0.992	0.957	0.980

Table 2.4: Sensitivity (SN) and specificity (SP) values for motifs of Set 2, reported by NestedMICA and MEME in the single-motif spiking tests.

Length	Abundance	NestedMICA		MEME	
		SN	SP	SN	SP
3	10	0.559	0.903	0.197	0.506
3	20	0.875	0.877	0.197	0.506
3	30	0.875	0.877	0.197	0.506
4	10	0.921	0.506	0.993	0.501
4	20	0.839	0.909	0.993	0.501
4	30	0.775	0.934	0.993	0.501
5	10	0.731	0.897	0.854	0.506
5	20	0.782	0.874	0.854	0.506
5	30	0.837	0.866	0.854	0.506
6	10	0.839	0.902	0.995	0.501
6	20	0.863	0.911	0.995	0.501
6	30	0.794	0.948	0.856	0.932
7	10	0.906	0.947	0.139	0.532
7	20	0.882	0.984	0.926	0.977
7	30	0.902	0.984	0.928	0.968
8	10	0.995	0.995	0.995	0.995
8	20	0.993	0.998	0.995	0.998
8	30	0.993	0.998	0.990	1.000
9	10	0.993	0.995	0.986	1.000
9	20	0.995	0.998	0.995	0.998
9	30	0.995	0.995	0.995	0.995

Table 2.5: Sensitivity (SN) and specificity (SP) values for motifs of Set 3, reported by NestedMICA and MEME in the single-motif spiking tests.



Name	Motif	MCC
m4		0.82
m7		0.94
m10		0.97

Figure 2.5: **Inserting more than one different motif into the sequences.** Original motifs used in multiple motif test are shown. These were inserted into the test sequences, at 40 and 20% total motif abundance rates. Resulting spiked sequences contain either zero, one or multiple different instances of the shown motifs, while sequences were not allowed to contain multiple instances of the same motif. The MCC values of these original motifs are given for comparison with the recovered motifs' MCCs. Results for recovered motifs are presented in Tables 2.6 and 2.7.

both NestedMICA and MEME to find single motifs, I assessed the two programs' ability to recover multiple motifs from a single dataset.

I used 3 test motifs of length 4, 7 and 10 aa, in the multiple motif spiking tests (Figure 2.5). Multiple motifs were spiked in such a way as to ensure an unbiased distribution. For example, in the first multiple motif spiking test, corresponding to a 40% abundance rate for each motif, it was ensured that 24% of the sequences were spiked with only motif of length 7, 24% only with motif of length 10 and 16% with both motifs. This corresponds to the distribution of motifs that would be expected by chance. The test was repeated by halving the total abundance rate for each motif.

In a similar way, two other pair combinations of the motifs were tested, and

## 2.3 Results and discussions

---

finally, three motifs were spiked at the same time. When the abundance rate for each spiked motif in the triple motif test was 40%, it was ensured that three different groups of sequences, each corresponding to 14.4% of the total, contained either motif of length 4, or 7 or 10; three different groups, each corresponding to 9.6% of the total contained two motif instances simultaneously (i.e. one group had both motifs of length 4 and 7, another had both 7 and 10, and finally another had both 4 and 10) and one group corresponding 6.4% contained all three motifs.

Tables 2.6 and 2.7 summarise the performances of NestedMICA and MEME, respectively, for the multiple motif finding tasks performed under different conditions. It shows the Cartesian distances and MCC values of the reported motifs (The corresponding sensitivity and specificity values are given in Table 2.8 for NestedMICA and Table 2.9 for MEME). In general, both NestedMICA and MEME performed well, except MEME had a tendency not to recover shorter motifs and instead report PWMs of maximum allowed length which did not correspond to any of the spiked motifs.

Motifs	Abundance	Distances	MCCs
m4 + m7	40	0.23, 0.45	0.74, 0.93
	20	0.54, 0.62	0.71, 0.93
m4 + m10	40	0.44, 0.75	0.81, 0.95
	20	0.34, 0.73	0.75, 0.96
m7 + m10	40	0.47, 1.11	0.95, 0.96
	20	0.71, 0.75	0.93, 0.95
m4 + m7 + m10	40	0.42, 1.01, 1.00	0.75, 0.95, 0.97
	20	0.64, 0.54, 0.57	0.71, 0.95, 0.97

Table 2.6: **Performance summary for NestedMICA in the multiple motif spiking tests.** The “distances” columns refer to the Cartesian distances between the reported motifs and the original ones which are shown in Figure 2.5. Motif names indicate length. In addition to Cartesian distances, MCC values are given for motifs recovered by NestedMICA.

## 2.3 Results and discussions

Motifs	Abundance	Distances	MCCs
m4 + m7	40	11.73, 0.53	0.02, 0.92
	20	11.73, 0.56	0.02, 0.94
m4 + m10	40	11.73, 0.46	0.02, 0.96
	20	11.73, 0.75	0.02, 0.96
m7 + m10	40	0.38, 0.45	0.94, 0.95
	20	0.70, 0.62	0.92, 0.95
m4 + m7 + m10	40	11.73, 0.44, 0.42	0.02, 0.93, 0.96
	20	11.73, 0.76, 0.82	0.02, 0.93, 0.95

Table 2.7: **Performance summary for MEME in the multiple motif spiking tests.** The “distances” columns refer to the Cartesian distances between the reported motifs and the original ones which are shown in Figure 2.5. Motif names indicate length.

Motifs	Abundance (%)	NestedMICA	
		SN	SP
m4 + m7	40	0.892, 0.949	0.855, 0.980
	20	0.685, 0.947	0.986, 0.980
m4 + m10	40	0.973, 0.964	0.856, 0.985
	20	0.745, 0.978	0.974, 0.980
m7 + m10	40	0.968, 0.976	0.982, 0.987
	20	0.932, 0.971	0.994, 0.983
m4 + m7 + m10	40	0.978, 0.968, 0.976	0.798, 0.985, 0.990
	20	0.685, 0.964, 0.978	0.986, 0.980, 0.987

Table 2.8: **Sensitivity (SN) and specificity (SP) values for motifs reported by NestedMICA in the multiple-motif spiking tests.** Motif names (m4, m7 etc.) refer to length and are shown in Figure 2.5. SN and SP values are given for each of the motifs involved in a multiple motif spiking test, and are separated by commas.

### 2.3.4 Performance vs. protein length

Having performed the motif spiking tests, in order to evaluate the two programs in a more natural situation, I observed the effects of varying sequence length on motif finding in multiple protein sets expected to contain C-terminal motifs. To this

## 2.3 Results and discussions

---

end, I used 198 non-redundant ER proteins (see Methods), a high proportion of which would be expected to contain the C-terminal ER retention signal mentioned above. I created three datasets containing sequence chunks of 60, 80 and 100 amino acid letters, respectively, taken from the C-terminal regions of these ER proteins.

Figure 2.6 depicts the motifs recovered from these three datasets by both programs. While MEME could not find the [KH]DEL motif at the tested sequence lengths of 80 and 100 amino acids, NestedMICA performed well, even when 100 amino acid long chunks were used. Apart from not looking similar at all to the KDEL motif, there was no consistency between the motifs reported by MEME when using the 80 and 100aa long sequences. Both programs were run with default protein parameters with a target motif length set to between 3 and 15 amino acids.

To investigate whether NestedMICA would still find the motif when there are

Motifs	Abundance (%)	MEME	
		SN	SP
m4 + m7	40	0.942, 0.947	0.503, 0.975
	20	0.942, 0.959	0.503, 0.982
m4 + m10	40	0.942, 0.980	0.503, 0.985
	20	0.942, 0.988	0.503, 0.978
m7 + m10	40	0.954, 0.980	0.982, 0.976
	20	0.949, 0.978	0.975, 0.973
m4 + m7 + m10	40	0.942, 0.952, 0.978	0.503, 0.980, 0.985
	20	0.942, 0.954, 0.976	0.503, 0.975, 0.978

Table 2.9: **Sensitivity (SN) and specificity (SP) values for motifs reported by MEME in the multiple-motif spiking tests.** Motif names (m4, m7 etc.) refer to length and are shown in Figure 2.5. SN and SP values are given for each of the motifs involved in a multiple motif spiking test, and are separated by commas.



Figure 2.6: **Motif recovery performance against sequence length.** The figure shows recovered motifs using NestedMICA and MEME. “Length” refers to how many amino acid letters from the right-most (C-terminal) part of sequences were used in each dataset created. The 4 amino acid long ER retention signal was recovered successfully by NestedMICA while MEME reported motifs of the maximum allowed length (given by the user) when the sequences were longer than 80 residues.

more than 100 residues per sequence, I tested it using 120 residue long C-terminal regions. The ER retention motif was found only when NestedMICA was asked to find two motifs. Investigating the other reported motif, I found that it was a thioredoxin family active site motif (Prosite id: PDOC00172) that is usually found in ER proteins. MEME was also tested when forced to find two motifs from the dataset containing the 80 amino acid long sequences. However, in addition to the motifs shown in Figure 2.6, it reported a 15 residue long motif which I could not locate in domain databases. Scanning this motif against the sequences, I noticed that it exists in 8 of the 198 proteins in the dataset.

### 2.3.5 “Null test” and significance of motifs

For motif discovery assessment purposes, spiking motifs into a dataset of sequences that already contained strong motifs would be undesirable, as the method

in question might report some of these intrinsic motifs instead of the artificially implanted ones. On the other hand, evaluating a motif discovery tool using a dataset of randomly generated sequences would be unfair, too, as this would be relatively easy for the program to recover a test motif.

Given that even sequences having a low sequence identity can in theory share some common sequential features, it is important to ensure that an unbiased set of sequences is used in the tests. For this reason I used non-homologous cytoplasmic sequences from the TargetP subcellular localisation dataset for these tests. This dataset had been already filtered by the TargetP developers using a homology reduction algorithm (Hobohm *et al.*, 1992) that ensures no homologous sequences exist in the set (Emanuelsson *et al.*, 2000), before I filtered it again to further reduce the maximum sequence identity between any of the sequences.

I ran both NestedMICA and MEME on this dataset, before it was spiked by any test motifs, using different minimum target motif lengths for each program tested. This “null test” was performed to confirm that the dataset I used in performing motif spiking tests is a reasonably suitable one. This negative control test also gives an idea about how well the trained background model represented the sequences.

For this purpose, NestedMICA was run with the default parameters optimized for protein sequences (for more details on the parameters, please see the program manual). In this test, the minimum target length was initially set to 2, then 3, and finally 4, while the maximum length was always kept as 15, as in the motif spiking tests. Motifs generated by NestedMICA from these runs were weak (Figure 2.7), having average information bit scores per position not exceeding

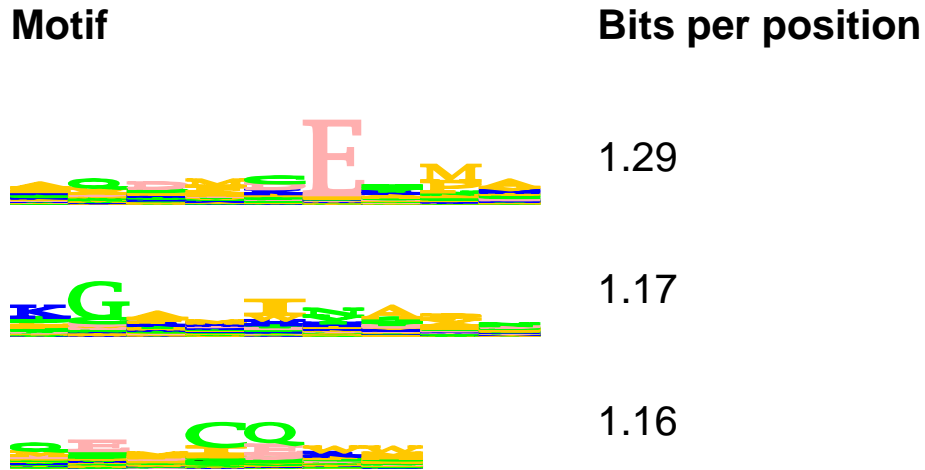


Figure 2.7: NestedMICA’s “null motifs”. When the minimum length parameter was set to 2, 3, and 4, NestedMICA generated almost flat motifs with few conserved positions, when no motif was inserted into the cytoplasmic test dataset. Bits per position is the averaged out value for the total information content of a motif, where it could be a maximum of 4.32 bits per position.

1.3 out of the possible 4.32 bits per position, which corresponds to roughly less than one third of the maximum height in a sequence logo. This indicates that NestedMICA does not generally report false positive motifs, and that the chosen background model parameters are good enough to represent the test set. As we have seen above, NestedMICA is sensitive enough to report even scarce motifs of length 3 when present in only 10% of the sequences, as the examples in Figures 2.2 - 2.4 indicate. Therefore, the fact that NestedMICA only reports weak “null test” motifs increases our confidence that the cytoplasmic sequence set that I use to assess motif discovery performance is not likely to contain significant motifs that a motif finder would prefer to report over any of our spiked motifs.

MEME, on the other hand, generally tended to report high-information containing motifs of the maximum allowed length, corresponding to about 46 bits in

total, and above 3 bits per residue position. To minimize any remaining common patterns in the sequence set, I further reduced the maximum sequence identity within the set to 30%. Furthermore, all sequence regions matching a Prosite pattern were removed, based on hits reported by an annotated motif search tool PPSearch (Quevillon *et al.*, 2005). However, even with this extra filtered dataset, MEME still reported strong and long motifs similar to the 15 amino acid long ones in Figures 2.2 - 2.4.

When the user-specified number of target motifs exceeds the number of actual motifs, NestedMICA has been observed to generate motifs that look like the null motif of that particular dataset (data not shown). Similarly, MEME produced the same type of long motifs it found in the null tests when it failed to find an inserted motif in the spiking tests.

### 2.3.6 Testing non-*ab initio* motif finders

As mentioned in the introduction section of this chapter, there are protein discovery tools which are not in the *ab initio* motif discovery category because they either might be using database look-ups, or homology search etc. One such program is Dilimot (Neduva & Russell, 2006). However, in addition to searching databases including PFAM (Bateman *et al.*, 2004) and SMART (Schultz *et al.*, 1998), it also utilises an *ab initio* tool, called TEIRESIAS (Rigoutsos & Floratos, 1998), which finds and lists frequently occurring patterns that could even contain gaps. Motifs are not reported as PWMs by this program. I normally compared NestedMICA with another probabilistic, *ab initio* method, MEME, which outputs motifs as PWMs, too. In this section, I provide an example to show whether



tools generating regular expressions for describing discovered patterns could be used successfully as the other probabilistic methods.

The Dilimot web server was provided with one dataset of protein sequences, 30% of which were spiked with motif of length 3 from the motif set 1 that I used to assess other programs (see Figure 2.2). The dataset contained 409 redundancy-reduced cytoplasmic sequences taken from the targetP (Emanuelsson *et al.*, 2000) subcellular localisation training set. After running a couple of days, the Dilimot program produced a table of discovered patterns in the form of regular expressions (Figure 2.8), however, none of the reported motifs were similar to the artificially spiked motif.

One disadvantageous aspect of such programs is that they are not based on probabilistic background models, which makes it very difficult for them to recover less abundant and short functional motifs, if not merely impossible. Because of this reason, they may report frequently repeating sequence regions instead, or regions that could be related to compositional features, unless they use motif databases having an entry for that particular motif.

## 2.4 Conclusions

I have added support for protein motif discovery in NestedMICA. It reports protein motifs in the form of PWMs. It has been optimized for better protein motif discovery under stringent conditions, and automatic motif length adjustment. In summary, our performance assessment tests show that NestedMICA performs very well when finding single and multiple motifs even at low motif abundance rates and different motif lengths, thus proving itself to be a robust and sensitive

## 2.4 Conclusions

**Input parameters:** Job id QLMD\_22970. Filters: GlobPlot No, Redundancy Yes, Pfam Yes, SMART Yes.  
 Motif parms: fixed pos. 3; maximum len 8; min no. proteins w. motif (ie col. 3) 3.  
 Species contributing to Scons: F.rubripes M.musculus R.norvegicus G.gallus H.sapiens D.melanogaster A.gambiae  
 C.elegans A.thaliana S.pombe K.lactis C.albicanis S.cereviciae D.hansenii K.waltii S.cereviciae C.glabrata

**Display criteria:** Max Scons 1.00e-05; Min. seq. w. motif (col. 3): 4; Max motifs shown 10

Motif	S <sub>cons</sub> ( <a href="#">help</a> )	Proteins with motif (in unfiltered regions) ( <a href="#">help</a> )	Proteins total (non-redundant) ( <a href="#">help</a> )	P (binomial test) ( <a href="#">help</a> )
<input type="checkbox"/> GxExF	0.00e+00	24	409	0.00e+00
<input type="checkbox"/> NxxxCxK	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> GGGGGGxG	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> LAxxxAxR	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> DxxSSSS	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> DxxxxEKQ	0.00e+00	4	409	0.00e+00
<input type="checkbox"/> HxALxxN	0.00e+00	5	409	0.00e+00
<input type="checkbox"/> FxxxxKxG	0.00e+00	27	409	0.00e+00
<input type="checkbox"/> PPPxPPxP	0.00e+00	6	409	0.00e+00
<input type="checkbox"/> SxVxxLxS	0.00e+00	9	409	0.00e+00
<input type="button" value="Apply new selection"/>	<a href="#">Display graphically all motifs in this table (help)</a>			

Figure 2.8: A snapshot showing the regular expressions reported by the Dilimot web service. Dilimot was run with the default options. It was allowed both to use the *ab initio* program, TEIRESIAS, and to consult other public protein pattern databases.

## 2.5 Availability and requirements of NestedMICA

---

protein motif finder. Judging from the calculated sensitivity, specificity and MCC values, there was no clear difference regarding the quality of motifs correctly recovered by NestedMICA or MEME. However, when it comes to the number of correctly recovered motifs, NestedMICA significantly outperformed MEME in our protein motif finding tasks including finding low abundant motifs, finding short motifs, and finally discovering motifs from amino acid sequences of different lengths.

In addition to assessing its ability in finding true positive motifs, as shown in the results section, by running it on a non-redundant dataset where no test motif was inserted, I have shown that NestedMICA does not tend to report high-information content motifs when there is no meaningful motif contained in the dataset, i.e. that it tends not to report strong false negatives.

Considering that some protein signals such as subcellular localisation motifs could be as short as 3 amino acids, this new protein motif finder is a promising tool in functional sequence annotation.

## 2.5 Availability and requirements of NestedMICA

- **Project Name:** NestedMICA
- **Project home page:** <http://www.sanger.ac.uk/Software/analysis/NestedMICA/>
- **Operating systems:** Platform independent
- **Programming language:** Java
- **Other requirements:** Biojava1.4, WoodStox, StAX-compliant XML parser

## 2.5 Availability and requirements of NestedMICA

(all included within the NestedMICA package), ANT 1.7.0 (<http://ant.apache.org>)  
to compile the project

- **License:** LGPL
- **Any restrictions to use by non-academics:** None

## Chapter 3

# Lokum: *ab initio* protein subcellular localisation prediction for eukaryotes by using mono and bipartite motifs, transmembrane protein topologies, and amino acid composition

### 3.1 Introduction

Protein sorting in eukaryotes is generally more complicated than in bacteria, simply because a typical eukaryotic cell contains a larger number of compartments. Presence of different compartments defined by various internal membranes within the cell mean different proteins must successfully pass through these internal envelopes, which naturally involves a larger number of molecules and different targeting and retention mechanisms. Identification of protein regions that are involved in protein transport across a certain membrane is a key step in all prediction efforts mimicking the underlying biological interactions. I try to address this issue by using a new, probabilistic, *ab initio* protein motif discovery tool,

NestedMICA (Down & Hubbard, 2005), which has been recently shown to work better than another popular program MEME (Bailey & Elkan, 1995), particularly for short proteins motifs that range in 3-9 amino acids (aa) (see Chapter 2 or Doğruel *et al.* (2008)). This makes NestedMICA suitable for use in localisation signal discovery, as targeting signals could be as short as 3 aa. NestedMICA, using a new Monte Carlo technique called Nested Sampling (Skilling, 2004), reports motifs in Position Weight Matrices (PWMs).

One of the basic forms protein localisation signals could be characterised by are multi-component probabilistic motifs, which most motif finders cannot deal with. I use a combinatorial strategy involving both NestedMICA and the Eponine tool (Down & Hubbard, 2002) that I have improved for protein sequence support.

### 3.1.1 Features used in Lokum

In this study, to predict protein localisation I used mono- and bipartite protein localisation signals discovered by NestedMICA and Eponine, other NestedMICA motifs that are not directly involved in localisation but that I show to be useful in the computational predictions, amino acid frequency distributions, and finally protein transmembrane topology statistics.

Apart from the difficulty of discovering genuine localisation signals, in signal-based *ab initio* protein subcellular localisation prediction another complication is the poor discriminative power of these motifs in the classification problem. Proteins can share the same type of localisation motifs, not necessarily because they are from the same cellular localisation, but because they could be involved in a similar translocation pathway. Partly because of such common localisation

signals, it is usually difficult to attain high prediction accuracies in automatic *ab initio* classification methodologies (for a list of some popular automatic prediction tools, see Section 1.1.1 in the introduction chapter). One possible way to reduce the weaknesses of individual features is to use as many relevant protein properties in combination as we can, where a pre-trained automatic prediction system will evaluate possible relations among the features to make a final decision. I used a popular classification method, Support Vector Machines (SVM), as they can provide very good generalisation performance by finding optimal hyper-surfaces that split data points of different classes in multi-dimensional spaces.

One general type of intrinsic signals proteins carry is targeting sequences. They are usually found in the N-terminal regions of proteins, and some of them are cleaved off from the nascent protein after the protein is translocated across a membrane. There could also be targeting signals located on the far C-terminus, like the Peroxisomal Targeting Signal 1 (PTS1) which is usually characterised by the tripeptide sequence SKL (Gould *et al.*, 1987, 1989). However, PTS1 is not found in all proteins that are post-translationally transported to the peroxisome. It is believed that peroxisomal proteins contain a weakly conserved N-terminal signal of the form [RK][LVI].....[HQ][LA], named PTS2 for “Peroxisomal Targeting Signal 2”, where the dots represent any amino acid (Osumi *et al.*, 1991; Swinkels *et al.*, 1991). Certain mitochondrial targeting peptides are located in the N-terminus, too, while these proteins can also have secondary signals which are thought to be present possibly anywhere along the entire pre-protein sequence (Endres *et al.*, 1999; Wiedemann *et al.*, 2001).

Not all secreted proteins have N-terminal targeting signals (Bendtsen *et al.*,

2004a; Nickel, 2003), however the major type of proteins that have an N-terminal targeting signal is the secretion pathway proteins, as they contain a conserved signal peptide (SP) (Milstein *et al.*, 1972) that can range in length between 20 and 30 amino acids in eukaryotes (Emanuelsson *et al.*, 1999; von Heijne, 1990). A usually cleavable N-terminal targeting peptide directs them into the ER by penetrating through the ER membrane, while the rest of the nascent polychain peptide is still being synthesised in ribosomes that are located near the ER. A smaller number of them are maintained and employed by the ER if they contain the tetrapeptide KDEL signal on their C-termini (Pelham, 1995). Most of these proteins that pass several “quality control tests” of the ER are then sent to the Golgi apparatus for further processing, but some of them, such as the mal-folded or unassembled ones that failed those tests, are delivered by the ER to the proteolytic system for degradation. This indicates there is some sort of back-and-forth traffic between the ER and Golgi, but that there are no reported retention or targeting motifs associated with the Golgi compartment. However, Yuan & Teasdale (2002) showed that up to a certain extent it is possible to distinguish Golgi Type II membrane proteins from the others, by using the hydrophobicity values and frequencies of different residues within their transmembrane domains. For most cargo molecules traversing through the “Golgi cisternae”, or multiple ordered stacks of the Golgi apparatus, Golgi acts only as an intermediate place. They eventually either end up in the plasma membrane, or are secreted out of the cell.

N-linked glycosylation is a common type of post-translational protein modification that takes place shortly after the nascent chain enters into the ER lumen



(Kaplan *et al.*, 1987; Machamer *et al.*, 1985). Starting as early as 1985, some previous studies have claimed glycosylation could have a role in cell transport (Guan *et al.*, 1985; Hannink & Donoghue, 1986; Kelley & Kinsella, 2003; Yan *et al.*, 2002) while others (Matsuda *et al.*, 2004; Mohrmann *et al.*, 2005) reported that it is not specifically required in cell surface transport for the tested protein molecules. A recent study demonstrated that N-linked glycosylation is required for structural stabilisation but not for membrane localisation of a tested particular protein (Gao & Mehta, 2007). The generally accepted notion seems to be that N-linked glycosylation is not directly involved in localisation. However, I show in this chapter that it is enriched in secretory pathway proteins over the other types, making it a potential secondary signal to aid in computational localisation prediction, just in a similar way to use the “secondary signal” coming from protein composition.

Amino acid residues can have similar physical and chemical characteristics. It is for this reason that protein signals such as the secretory pathway signal peptide (SP) are described often in terms of their general characteristics like hydrophobicity, net charge etc., rather than in terms of their individual amino acid letters which might not be conserved, as in the case of SP, for example. Individuals of different generations can have protein sequences that are still functionally similar yet different in terms of the actual amino acid line up due to the associated DNA-level mutations that take place in the process of evolution. Up to a certain extent, it is therefore possible to safely substitute certain amino acid residues with other similar ones without much harming the function and thus affecting the tertiary structure of a protein. The study of such functionally homologous blocks

showing sequence variation has resulted in amino acid substitution matrices like PAM (Eck & Dayhoff, 1966) and BLOSUM (Henikoff & Henikoff, 1992)

In Lokum, apart from the motifs discovered *ab initio*, I used the normalised amino acid abundance rates in sequences. Amino acid composition has been proven useful in localisation prediction (Klein *et al.*, 1984; Nakai & Kanehisa, 1991; Reinhardt & Hubbard, 1998). There have been many machine learning approaches incorporating amino acid frequency distributions alone or sometimes accompanied with other features. Reinhardt & Hubbard (1998) suggested that using amino acid composition would be advantageous over other signal-based methods as it makes a protein less susceptible to possible annotation errors, particularly in the 5' regions where most targeting signals reside. However, by using probabilistic representations such as Position Weight Matrices (PWMs) to characterise such signals it is possible to tolerate slight sequential variations. This argument becomes more valid especially for PWM positions having almost flat distributions of amino acid probabilities, where any amino acid can be expected to occupy those positions.

The third type of protein feature I used is predicted secondary transmembrane structures. Amongst the transmembrane topology predicting programs such as TopPred (Claros & von Heijne, 1994), SOSUI (Hirokawa *et al.*, 1998), TMHMM (Krogh *et al.*, 2001) and HMMTOP (Tusnady & Simon, 2001), studies on evaluation of these programs showed that TMHMM performed better than the rest of the predictors. It has been reported that, in general all the tested programs can easily misclassify the predominantly hydrophobic membrane spanning regions as N-terminal signal peptides which also contain a similar strong hydrophobic re-

gion (Lao *et al.*, 2002; Müller *et al.*, 2001). For the same reason, signal peptide predictors may often misjudge transmembrane regions as signal peptides, too. Chapter 5 summarises my efforts to develop a transmembrane topology predictor that can be used in subcellular localisation prediction, but this HMM-based tool didn't perform as well as TMHMM. Therefore, in the end, in Lokum I used transmembrane topology statistics based on TMHMM predictions.

### 3.1.2 Predicted classes

In this manuscript, I compare my *ab initio* method, Lokum (Localisation prediction using motifs) with both PSORT and MultiLoc. Similarly to these programs, Lokum predicts nine localisation categories for animal proteins: nucleus, cytoplasm, plasma membrane, extracellular space, mitochondrion, endoplasmic reticulum, Golgi apparatus, lysosome, and peroxisome. Next, substituting lysosomes with vacuolar proteins in the animal set, Lokum's predictions are extended to cover all major nine fungal protein localisations, and finally ten plant localisation classes with the addition of chloroplast to the list of fungal classes.

## 3.2 Materials and methods

### 3.2.1 Localisation motif discovery with NestedMICA

I used NestedMICA, an *ab initio* DNA and protein motif discovery tool, to search for localisation-specific motifs that can be used in classification. NestedMICA employs a new Monte Carlo inference technique called nested sampling developed by Skilling (see page 28). It was originally developed for finding DNA motifs, and has been recently extended to find protein motifs (see Chapter 2 or Doğruel *et al.*

## 3.2 Materials and methods

---

(2008)). It reports motifs in the form of Position Weight Matrices (PWMs) which allow more flexibility for having alternative residues at certain positions than, for example, motifs represented as regular expressions.

The target motif length interval parameter was given to be between 3 and 15 amino acids long in all the NestedMICA runs. Initially, the target motif number was specified as 2, but I experimented with this program parameter for each localisation class to cover as many potentially localisation-related motifs as possible. NestedMICA was run on the full-length sequences, as well as 50 N- and C-terminal amino acid chunks for each localisation dataset. The ER retention signals (Figure 3.4f-g) and PTS1 (Figure 3.4j) were recovered when NestedMICA was fed with the last (C-terminal) 50aa long regions.

For motif discovery purposes, I used nine datasets from pTarget (Guda & Subramaniam, 2005), a subcellular localisation predictor based on searching more than 2100 PFAM domains, after reducing the mutual sequence identities of the datasets from 95% to a maximum of 40% by the CD-HIT algorithm (Li & Godzik, 2006). Table 3.1 lists the number of sequences before and after applying redundancy reduction. For localisation categories that do not exist in pTarget, namely for chloroplasts and vacuolar classes, I used the redundancy-reduced datasets of MultiLoc (Höglund *et al.*, 2006), a recent subcellular localisation prediction program. The details for these two sequence sets can be seen in Table 3.4 where Lokum predictions are compared with those of MultiLoc. I further decreased the maximum mutual sequence identities of the MultiLoc datasets as well to 40% before running NestedMICA.

NestedMICA uses complex background models which could be composed of

## 3.2 Materials and methods

Localisation class	Number of sequences in the original set	Number of sequences after the filtering
Cytoplasmic	2062	946
ER	693	251
Extracellular	5688	1671
Golgi	221	141
Lysosome	174	66
Mitochondria	1698	711
Nuclear	3446	2014
Peroxisome	173	83
Plasma membrane	4162	1212

Table 3.1: **Sequences used in the motif discovery phase.** Each pTarget (Guda & Subramaniam, 2005) dataset, originally having a sequence identity of 95%, was filtered to have a maximum mutual identity of 40% by using the CD-HIT (Li & Godzik, 2006) clustering program. Vacuolar and chloroplast classes do not exist in pTarget, so the corresponding datasets of MultiLoc (Höglund *et al.*, 2006) were used for these two categories (Table 3.4).

multiple subgroups of different amino acid probability distributions to better represent different sequence regions statistically inclined to feature certain amino acid residues more frequently. As has been discussed in 2.3.1, training dedicated background models for each sequence dataset yields better performance than using a generic background model. Therefore, for each type of localisation a specialised NestedMICA background model was trained. The background model parameters used for each localisation dataset are summarised in Table 3.2. NestedMICA was run on each dataset with its default protein motif finding parameters.

### 3.2.2 Motif selection

NestedMICA does not report any significance measure. To decide if a reported motif is significantly contributing to localisation classification, I scanned it across

## 3.2 Materials and methods

---

some test sequences to plot Receiver Operating Characteristic (ROC) curves, as in Figures 3.7 and 3.9. A motif discovered from the plasma membrane set, for example, was tested for its usefulness to discriminate between plasma membrane sequences and every other class of sequences. By using equal number of sequences of both types, in each binary classification based on only raw bit scores of a motif, I classified sequences in two classes according to a range of motif score thresholds. ROC curves were plotted using the sensitivity and specificity pairs obtained for each threshold used. Motifs producing promising ROC curves in any possible binary classification were then selected to be used in the general multi-class SVM.

Additionally, I performed a brute-force principle component analysis to assess the contribution of each selected feature, or dimension of SVM vectors. I observed

Dataset	MC-order	Number of Mosaics
ER	0	5
Vacuolar	0	2
Lysosome	0	4
Golgi	0	5
Mitochondria	1	3
Chloroplast	0	5
Peroxisomal	0	6
Nuclear	1	4
Cytoplasmic	1	6
Extracellular	1	4
Plasma membrane	1	6

Table 3.2: **Protein background parameters for datasets used in localisation motif discovery** The table summarises the NestedMICA background properties of the datasets where localisation related motifs were searched, in terms of the used Markov-chain order and the number of mosaic classes in the background. These parameter values have been optimised after a systematic analysis of each dataset as described in Chapter 2, Section 2.2.4.

the effects of removing a single or multiple dimensions from the input vectors on the overall performance. Features increasing the prediction performance upon removal were not used in the final SVM. None of the amino acid frequency dimensions were necessary to remove. As an interesting example, PTS2 was among the motifs I decided not to use in the end (see results).

### 3.2.3 Using Eponine with NestedMICA for multi-component motif discovery

Some localisation signals can consist of multiple components separated by a distance. The best known such signal is the bipartite NLS (Dingwall & Laskey, 1991) which has been identified to have two core NLS parts that are separated by at least 10 (Robbins *et al.*, 1991) and around 12 (Schreiber *et al.*, 1992) “spacer” amino acids. NestedMICA currently does not deal with multi-component motifs. I modified and extended the Eponine (Down & Hubbard, 2002) tool to discover and represent such protein localisation signals.

Eponine was originally developed to find promoter models from mammalian genomic DNA to represent multi-component, hierarchical motifs. Eponine describes these multi-component motifs as Eponine Anchored Sequence (EAS) models, where motifs are modeled around a fixed, or “anchor” point. It generates a number of weight matrices corresponding to different sequence motifs which it believes to be collectively involved in signaling a certain sequence characteristics. Each motif within an Eponine motif set has a positional distribution relative to a point of interest, such as a transcription start site (TSS) point. When scoring sequences with an Eponine model, positional deviations of the best matching

sub-motifs with respect to the means of the corresponding Gaussian distributions are considered, too. Figure 3.1 shows an EAS which models mammalian TSS regions, as reported by [Down & Hubbard](#). It has been later shown in a PhD dissertation that it can actually be used as a multi-purpose motif finder, where it was specifically used in the detection of transcription termination sites (TTS) ([Ramadass, 2005](#)). Figure 3.2 shows the discovered EAS model for mammalian TTS regions.

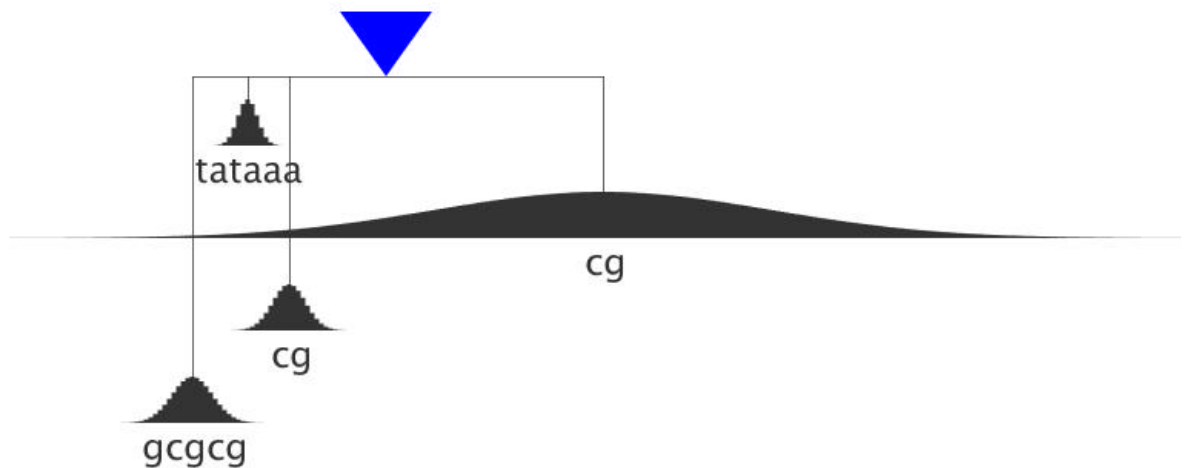


Figure 3.1: **Eponine TSS model.** Blue triangle in Eponine Anchored Sequence (EAS) models indicate the anchor point. Individual motif weight matrices are positioned with respect to the anchor. Gaussian distributions indicate the positional distributions of the corresponding motif. This TSS model has been reproduced from the original Eponine publication ([Down & Hubbard, 2002](#)).

Eponine was later extended for use in non-coding DNA region analysis with the purpose of discovering overrepresented multi-component motifs conserved in mouse and human intergenic regions [Down & Hubbard \(2004\)](#). This version of Eponine describes motifs as Eponine Windowed Sequence (EWS) models, in analogy to the previous model type. In EWS, unlike the first version, there





Figure 3.2: **Eponine TTS model.** As in Figure 3.1, this mammalian transcription termination site (TTS) model, too, is an example to EAS models, as reported in [Ramadass's](#) PhD thesis.

is no need to have a specific sequence position around which other sub motif components are placed. Instead, this model classifies sequence regions based on only their sequence contents within specific windows.

Eponine, which uses Biojava ([BioJava, 2007](#)) libraries, relies on a new machine learning strategy called Relevance Vector Machines (RVMs) ([Tipping, 2001](#)) taking a set of suggested basis functions and then iteratively choosing certain combinations that would presumably yield a better performance at each step. To this end, it optimises candidate PWMs and their parameters including width and positional Gaussian distributions. It requires both a positive and a negative training set to decide if the combination used at each step is better discriminating the two classes. Because Eponine actually works by trying to discriminate data points, it searches for motifs in the negative set, too. This can result in reported models to have some “negative” motifs which have negative weights in the models (they are drawn in blue colour in the graphical representations, as opposed to the black “positive” motifs). Generally speaking, not having any negative motifs in reported Eponine models trained using negative datasets that are obtained by

shuffling the used positive samples indicates a successful training. More information about how Eponine works can be found in the original Eponine publication by [Down & Hubbard \(2002\)](#).

With the idea of employing Eponine to discover multi-component protein motifs such as the bipartite nuclear motifs, I added protein sequence support to Eponine. Having modified it to accept protein sequence input, I tested its efficiency in the protein space. However, my tests generally indicated that the parameter space was too large for Eponine to be directly used efficiently in multi-component protein motif discovery (data not shown), which could be explained by the fact that amino acid alphabet is 5 fold larger than the DNA one, having high noise levels to be analysed with this tool. In most of these experiments, the system never converged automatically, and it contained “negative” motifs (data not shown).

In order to limit the problem size, I have come up with a hybrid, semi-guided, two-step procedure involving the probabilistic motif discovery tool NestedMICA, as well as the Eponine tool which can build multi-component hierarchical motif models to describe complicated sequence structures with its machine learning strategies. In the first step, I use NestedMICA to find some monopartite motifs, then by expanding those sequence regions by around 20 amino acids from both sides, where there is a significant match of a reported NestedMICA motif, I construct a new dataset composed of sequence chunks that have an instance of the used single-part motif. In the second phase, Eponine is run on this filtered dataset containing the positive samples, and also a negative dataset which has the same number of samples but not containing any motif hit.

To preserve the general sequence characteristics of the positive set across the negative set, sequence samples in the negative set are obtained from the same protein dataset so as to prevent Eponine from finding motifs that could possibly be reflecting potential compositional differences of the two sets.

Because I have sequence chunks with fixed lengths, each having a monopartite motif at the middle, Eponine was run in the EAS mode. After all, the aim is to find a multi-component motif model based on a reported NestedMICA motif whose position is known. The anchor point was specified as the maximum scoring point when scanned with the monopartite NestedMICA motif.

### 3.2.4 Using amino acid composition

It is possible to group amino acids according to their physical and chemical characteristics. If there are similar amino acids, one question to ask is whether grouping similar amino acid residues together, and then calculating the composition of the ‘labels’ of these groups rather than finding occurrence rates for each of the 20 amino acids could be a better approach or not. This brings two complications: determining the optimal number of such groups, and deciding which amino acid letters will be classified under which group. I used three amino acid groupings suggested by [Thomas & Dill \(1996\)](#), found by an iterative procedure involving “energy” scores calculated by iteration until they correctly discriminate a set of known protein folds from decoy conformations. [Table 3.3](#) shows two types of amino acid groupings from [Thomas & Dill \(1996\)](#) and one grouping I formed based on general amino acid characteristics.

In the SVM, I kept all other features, except that the composition values

	Group 1	Group 2	Group 3
1	VILMF	VILMFWYA	ILVM
2	HQN	GPSTHQ	TSNQ
3	C	C	EDKRH
4	ED	ED	WFYP
5	RK	RK	C
6	A		GA
7	G		
8	WY		
9	P		
10	ST		

Table 3.3: **Alternative amino acid groupings used in composition calculation.** Groups 1 & 2 are from [Thomas & Dill \(1996\)](#), while Group 3 was constructed based on general amino acid properties.

were calculated according to these amino acid groups rather than using the 20 amino acids directly. The performances of the SVMs in the experiments were evaluated by using 5-fold cross validation. All parameters of the kernel function were optimised for each type of amino acid grouping I used, as in the optimisation of the actual SVM I used (see the section below, [3.2.6](#)).

It turned out that grouping amino acids according to their physical and chemical properties is not particularly helpful (see page [95](#) in the Results section), so instead, 20 values have been computed to demonstrate amino acid composition statistics for each sequence.

### 3.2.5 Using transmembrane topology predictions

Apart from using amino acid composition and bit scores of motifs discovered by NestedMICA and Eponine, predicted transmembrane topology statistics were used as well to create Support Vector Machine feature vectors. Transmembrane

region predictions were reported by the 2c version of the TMHMM transmembrane topology prediction program (Krogh *et al.*, 2001). TMHMM was run in the “short statistics” mode. Amongst the reported TMHMM statistics, I included the following reported features:

- the number of predicted transmembrane helices
- the expected number of amino acids lying in transmembrane helices, considering the entire sequence
- the expected number of amino acids lying in transmembrane helices, considering only the first 60 N-terminal amino acids

Before using these reported numbers in the SVM, they were normalised with respect to the length of the input sequence considered.

### 3.2.6 Training and testing of SVM

I used a popular open source implementation of SVM, *libsvm* (Chang & Lin, 2001), in the multi-class predictor Lokum. In the parameter optimisations carried out to maximise the performance of each tried SVM application, *libsvm* performed slightly better than the other popular SVM applications I tried, namely, *SVM<sup>light</sup>* (Joachims, 1999) and BSVM (Hsu & Lin, 2002).

Eventually, a radial basis kernel function (RBF) was used in *libsvm* after a systematic evaluation of a selection of kernel functions. In a similar way, I performed a grid search to optimise the gamma ( $g$ ) and cost ( $C$ ) parameters of this kernel function (Figure 3.3). The training and performance assessment of the SVM involved a 5-fold cross validation procedure in which the data were

divided into 5 portions; 4/5 of which were used for training and 1/5 for testing, using a particular portion for testing at a time in each of the 5 cycles. All protein scores coming from different features have been normalised to have a minimum value of -1 and a maximum value of 1, before the SVM software was run. The individual SVMs constructed to give an idea about the contributions of motif scores, composition and structural information were trained with 4/5 of the data. Kernel parameters of each SVM using a particular type of feature has been optimised, too, before I tested the SVMs with the remaining 1/5 portion. During kernel parameter optimisation, 3-fold cross validation was used for faster analysis.

### 3.2.7 Evaluation of Lokum predictions

The reported overall accuracy is the arithmetic mean of the correctly classified sequence percentage in each cross validation iteration. Sensitivity (SN), specificity (SP) and Matthew’s Correlation Coefficient (MCC) ([Matthews, 1975](#)) values were calculated for each predicted class according to the formulae given in Equations [2.3](#), [2.4](#) and [2.5](#), respectively.

## 3.3 Results

By using NestedMICA, I found many motifs from different localisation datasets (see Appendix [A](#) for sequence logos of these motifs). Not all of these motifs ended up being used in Lokum, however: discovered motifs were assessed for their discriminative powers (see Section [3.2.2](#)), and those not contributing to localisation prediction were filtered out. Figure [3.4](#) shows some mono-partite localisation

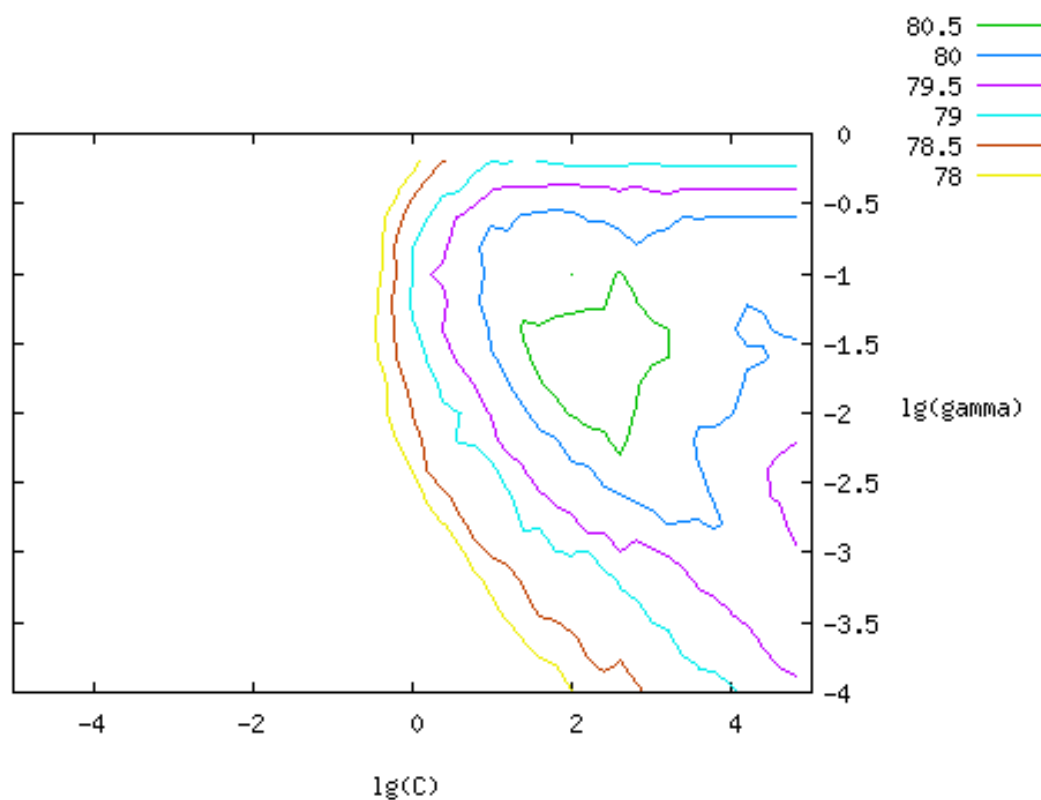


Figure 3.3: **SVM kernel parameter optimisation.** The plot shows an example set of percent accuracy contours formed by different values of the gamma ( $g$ ) and cost ( $C$ ) parameters (given in  $\log_2$ ) during the optimisation of a radial-based kernel function (RBF) used in the SVM. Different pairs of  $g$  and  $C$  may produce similar percent accuracy rates, hence the contours. The specific example shown is for the animal protein version of Lokum. Accuracies plotted have been rounded to the nearest lower half values, i.e., an accuracy of 80.78% was considered in the 80.5% group of accuracies for plotting. Increasing the number of cross-validation iterations can increase the perceived performance (see text for the actual percent accuracies attained for different organisms using cross validation).

signals I used in Lokum. These motifs, represented as sequence logos here, correspond to some known localisation signals which are mostly characterised as regular expressions in literature. Other longer, and probably mostly unannotated, part-of-domain motifs that were used in Lokum can be found in Figure 3.5. Lokum prediction server is available online for public use at:

<http://www.sanger.ac.uk/Software/analysis/lokum/>

The discovered localisation related motifs that were used by Lokum can be downloaded in NestedMICA's XML format (.xms) from the same web page. See Section 3.5 for more information on the Lokum web server.

### 3.3.1 Discovered monopartite motifs

As plasma membranes have a highly hydrophobic region within their transmembrane helices (Figure 3.4c), which is very similar to hydrophobic regions of signal peptide sequences (Figure 3.4b), only the latter was used in the predictor. The signal peptide (SP) that is found in most of the secretory pathway proteins can be thought of consisting three parts: an N-terminal part (n-region) which can vary in length and has a net positive charge, a central hydrophobic core (Figure 3.4b), and a c-region which features a “-3 -1” rule (von Heijne, 1986) indicating the conserved positions with respect to the cleavage site (Figure 3.4a).

Figure 3.4k shows a good example of how NestedMICA can be efficiently used in short functional protein site finding. The depicted 4-position PWM looks quite similar to the cleavage site of a previously reported long chloroplast transit peptide (cTP) sequence logo (Figure 3.6) which was obtained by aligning the N-terminal regions of 62 chloroplast sequences with known cleavage site positions



Name	Motif
a) SP cleavage site	
b) Hydrophobic part of SP	
c) Transmembrane helix hydrophobic core	
d) N-linked glycosylation (1)	
e) N-linked glycosylation (2)	
f) ER retention	
g) C-terminal signal for recycling into ER	
h) Nuclear signals	
i) Nuclear signals	
j) Peroxisomal targeting signal 1 (PTS1)	
k) Chloroplast transit peptide (cTP) cleavage site	

Figure 3.4: **Some of the protein localisation related signals as recovered by NestedMICA.** Each motif has a maximum information content of 4.3 bits per position. Amino acids are drawn in four colours: hydrophobic residues are depicted in orange, hydrophilic and polar ones in green, acidic ones in pink, and finally basic amino acids are in blue.

#	Dataset	Motif	Location scanned
1	mitochondrial	R	
2	plasma membrane	PA	
3	plasma membrane	C N C D	
4	lysosome	SY P	
5	lysosome	Y W I Y K N S W G W G G	
6	golgi	C A V V G N S G L S	
7	peroxisomal	X H A	
8	nuclear	K R I V N	
9	vacuolar	W E W M T S P R P H W Y	
10	vacuolar	E C C C F W Y G N T	
11	vacuolar	P E E	
12	vacuolar	N L D N	
13	vacuolar	I W E W M T Q M K H R H Y C C G	
14	vacuolar	E Y G C G	
15	vacuolar	E C F C G	
16	chloroplast	G V L I S V E Y P D Y I	
17	chloroplast	H T A Y E R A N Y C P S C	
18	chloroplast	K R A F H C H E C A	

Figure 3.5: Some of the unannotated signals, or part-of-domain motifs reported by NestedMICA. These motifs were discovered from localisation datasets given on the second column. Sometimes the motifs were scanned in certain positions on protein sequences, rather than using the whole sequence (last column).

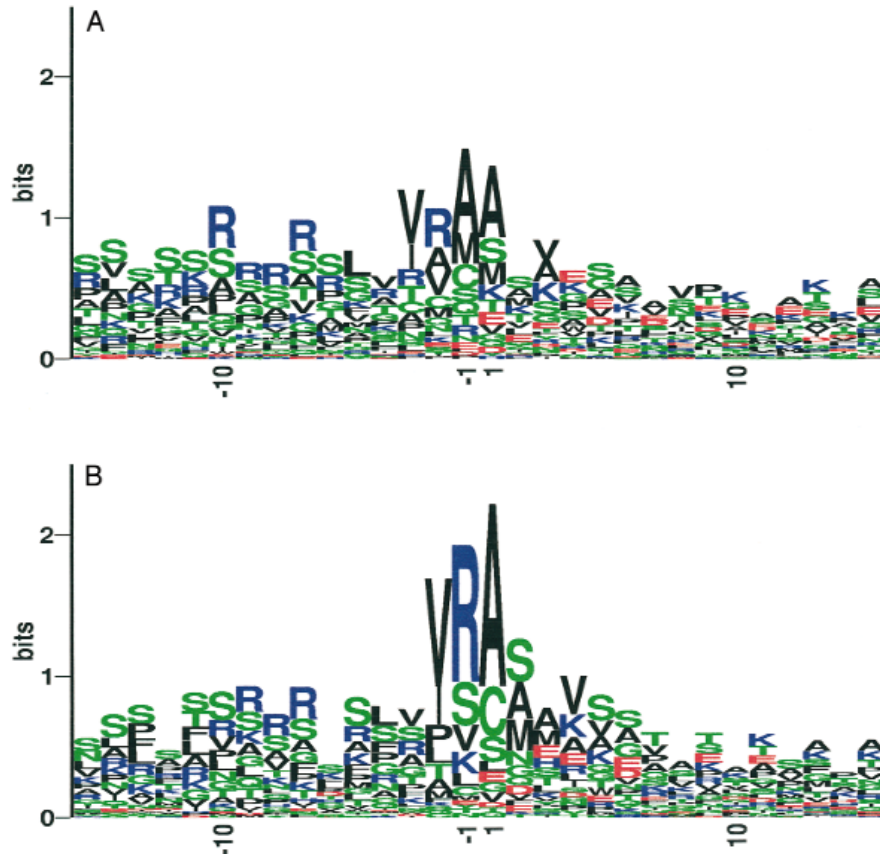


Figure 3.6: **Manually constructed motifs that are used in the ChloroP predictor.** This figure has been reproduced from the chloroP paper by [Emanuelsson \*et al.\* \(1999\)](#). The sequence logos were constructed from the 62 sequences used in the cleavage site predictor (chloroP) development. The sequences are aligned around their SWISS-PROT annotated cleavage site (top logo) and around the predicted cleavage site (bottom logo). Note the similarity between the motif shown in Figure 3.4k which is discovered automatically by NestedMICA and the conserved cleavage region of the manually aligned ChloroP logo in this figure (bottom).

that were kept fixed in the alignments ([Emanuelsson \*et al.\*, 1999](#)).

### 3.3.1.1 Contribution of N-linked glycosylation signal

Investigating the 3-letter motifs reported (Figure 3.4d-e), I found that these motifs correspond to the N-linked glycosylation signal which is found in two forms:

there is an Asparagine (N) residue in the first position followed by a non-conserved position, while the third position, determining the sub-variant, is occupied by either a Threonine (T) or a Serine (S) residue. Given that this is only a 3-letter motif, the chances are a fraction of its contribution to predictions could be due to some compositional effects. Namely, sequences having more number of the amino acid letters N, T or S, for example, could get higher scores when scanned with this motif, although, in reality they may not feature a glycosylation site. To investigate if there is a significant contribution coming from this motif apart from its compositional effects, I built artificial 3-letter motifs by inverting the positions of residues in this motif. Figure 3.7 shows the ROC curves measuring the classification power of the N-linked glycosylation motif, along with the shuffled motifs which of course retain the same composition as the original. The unshuffled original motif showed a better performance than all the other 5 possible variants, which indicates that using the N-linked glycosylation motifs is useful in computational protein localisation predictions, although it may not be directly involved in protein sorting processes as previous studies have demonstrated (see introduction).

### 3.3.1.2 Alternative ER retrieval

When NestedMICA was run on a dataset containing C-terminal ER sequences of length 20 aa, it reported the [KH]DEL motif shown in Figure 3.4f. When it was asked to find two motifs from the same region, instead of reporting a different or a weak motif (see the discussion on “null motifs” in Section 2.3.5), it reported another motif that looks like the first one, with the first residue being quite weak.

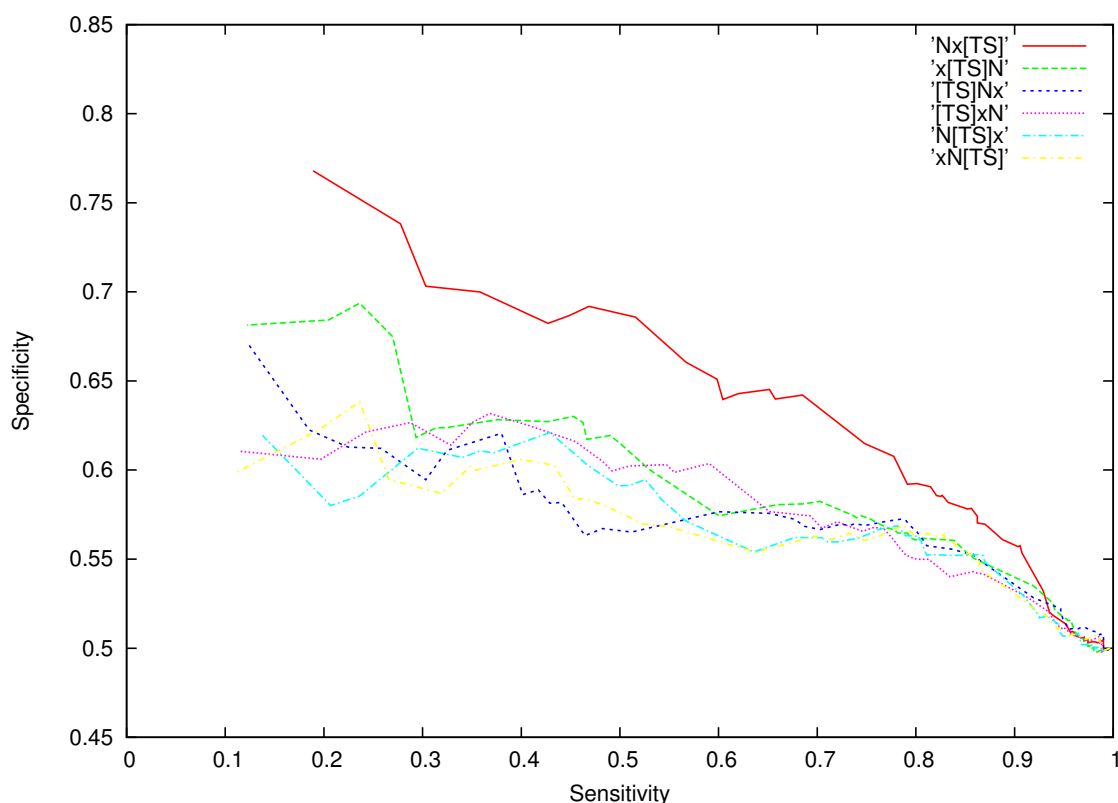


Figure 3.7: **ROC plots showing the contribution of the N-linked glycosylation motif in binary classification between nonredundant 509 plasma membrane and 509 mitochondrial protein sequences taken from the Multiloc datasets.** The curves correspond to the sensitivity (x-axis) and specificity (y-axis) values of multiple classifications performed by using a range of threshold scores. Each sequence was scored according to the best hit of the glycosylation motif and also the best hits of each of the derived PWMs obtained by shuffling the original motif's positions. This way I can evaluate a signal's performance with respect to the contribution of composition which is conserved in all the derived motifs. The red solid line shows the ROC for the original motif, while the dashed lines represent the shuffled PWMs' ROCs. Each motif's consensus sequence is shown in the legend, where the [TS] notation means there is either a T or S at that position, while 'x' indicates an unconserved position.



Figure 3.8: **The two C-terminal ER retention motifs reported.** The two motifs differed in their first residues, which may indicate that while some ER sequences have either K or H at position 1 of their C-terminal ER retention signals, some of them simply have not conserved the first position of this signal.

Figure 3.8 shows both motifs reported in this second run. This may suggest that while some sequences have either K or H at position 1 of this signal, in the others there is no preferred amino acid residue for this position, and that for them this signal is practically three amino acid longs.

NestedMICA has a useful feature which enables the user to find motifs other than a set of user-supplied motifs that are ignored during the program's motif search if they are found in the input sequence. When I run NestedMICA on a set of 20 amino acid long C-terminal ER amino acid sequence chunks by masking the [KH]DEL PWM found before, I came across an Arginine (R) and Lysine (K) rich motif that is shown in Figure 3.4g. While investigating some possible explanations to this motif in the literature, I found that [Pelham \(1995\)](#) had previously demonstrated that [KH]DEL is not the only C-terminus signal ER proteins might possess: a similar mechanism recycles escaped ER membrane proteins that have a loosely defined lysine (K)-rich, 4 amino acid long signal.

This reported NestedMICA PWM which does not have a clear consensus sequence could possibly be linked with this second ER retrieval mechanism. Including this motif in the SVM had a slight contribution ( $< 1\%$ ) in the overall prediction performance.

### 3.3.1.3 Scanning motifs in certain positions

Some localisation signals have specific positions in sequences. The ER retention signal (Figure 3.4f), for example, is located at the far C-terminal end. Therefore, while scanning and scoring sequences for the presence of such motifs, only specific regions have been considered. In the case of the ER retention signal, this was the last four residues on the C-terminus. The SP cleavage motif (Figure 3.4a) was scanned in a window of 50 N-terminal amino acid positions. Similarly, the hydrophobic-residue rich motif of Figure 3.4b has been scored only within the 20 N-terminal sequences. Scanning PWMs in specific sequence regions where they are more likely to be present has a significant advantage over scanning them in the entire sequences. Figure 3.9 demonstrates one such example of how well motif b of Figure 3.4 can discriminate between redundancy reduced 841 extracellular and 841 cytoplasmic proteins, where two ROC curves are plotted using scores obtained by scanning the motif in whole-length sequences, and only in the first (N-terminal) 40 amino acid region, respectively.

### 3.3.1.4 Scoring multiple instances of motifs

In constructing the SVM vectors, in addition to using the maximum motif score corresponding to the sequence position where the best match occurs, I used the second best scores for the core NLSs (Figure 3.4h-i), and also for the N-linked

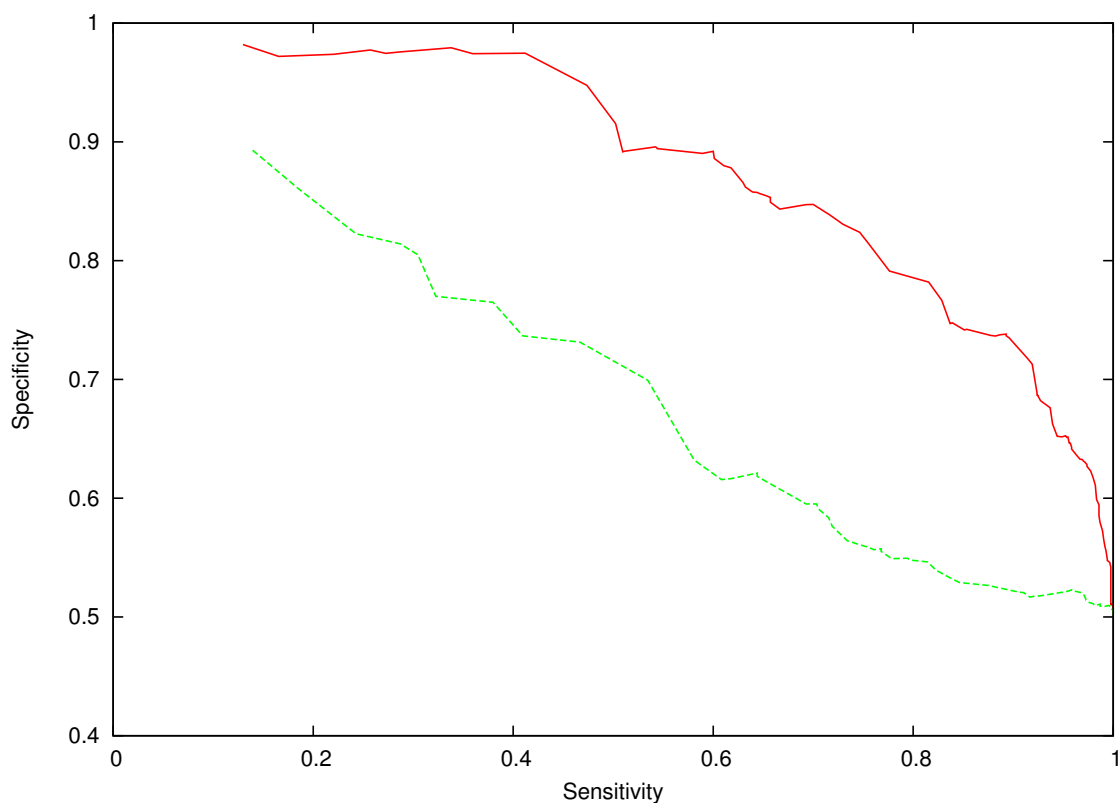


Figure 3.9: **A ROC curve showing the effect of scanning sequences with PWMs in certain segments only.** The plot shows different sensitivity (x-axis) and specificity (y-axis) values obtained for a range of score thresholds, indicating how well extracellular proteins can be discriminated from cytoplasmic proteins by using motif b of Figure 3.4. The red line is obtained when sequences were scored using only chunks of 40 N-terminal amino acids, while the green line represents the reduced performance attained when full-length sequences were scanned to obtain the maximum score.



glycosylation motifs (3.4 d-e). With this addition, I observed a significant increase in the overall classification accuracy, suggesting that some of the identified signals can possibly exist in more than a single region across a sequence. Other motifs did not even slightly increase the overall accuracy when I additionally used their second best scores.

### 3.3.2 Bipartite motif models

#### 3.3.2.1 Bipartite NLS

As described in the methods section, I used a semi-guided procedure where I used both Eponine and NestedMICA to characterise such motifs. Figure 3.10 shows one possible model to describe a bipartite NLS signal. Generally, individual motif components do not have to have fixed positions in Eponine models; instead in Eponine’s EAS models they are attributed with positional distributions with respect to an anchor point as described in Section 3.2.3. These Gaussian distributions reflect a motif’s occurrence frequency within an optimal sequence range. The variations in the distributions shown on Figure 3.10 are quite minimal, indicating that relative sub-motif positions in this particular NLS model usually vary at most by a couple of residues.

In “nuclear versus others” type binary predictions made to assess the contribution of individual nuclear motifs, this bipartite NLS motif by its own classified correctly 141 nuclear sequences that the other mono-partite motifs shown in Figure 3.4h-i could not predict alone. Raw motif score thresholds used in these two-way classifications were chosen such that they maximise the corresponding MCC values computed to measure correct classification rate. The “others” sequence

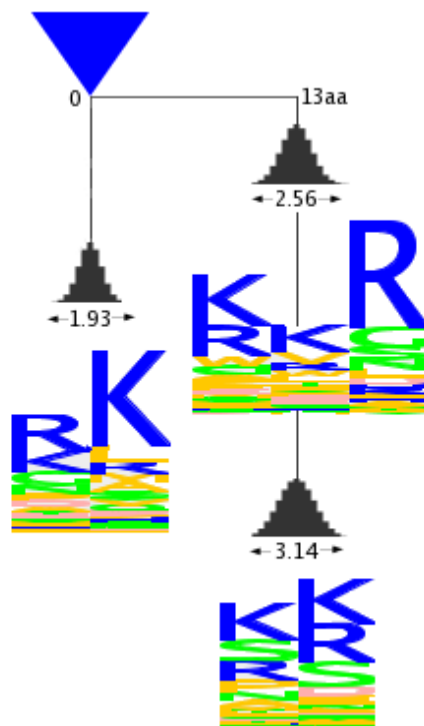


Figure 3.10: **Schematic representation of the Eponine bipartite NLS model.** The constraint distributions, the sequence logos and the relative positions of the individual components of the model are shown with respect to an anchor point (blue triangle). The central parts of the two main branches in the model are separated by 13aa's as shown. The model tolerates each sub-PWM to change position within the depicted probabilistic distribution width.

set in this particular experiment was compiled from the remaining 8 localisation datasets contributing in roughly equal numbers, and contained the same number of sequences in the tested nuclear set, 836.

### 3.3.2.2 Bipartite PTS2

To find a bipartite Eponine model for PTS2 (see Section 3.1.1), I followed the same procedure in modeling the bipartite NLS. However, it was more difficult for NestedMICA to discover the individual components of this weak bipartite motif,

each consisting of only a couple of adjacent residues, to enable me to perform the sequence filtering step in the multi-component model finding methodology (see methods) before running Eponine.

To investigate why NestedMICA failed to identify this motif or its components, I scanned the PTS2 regular expression “[RK][LVI].....[HQ][LA]” (Section 3.1.1) in 157 peroxisomal sequences I used. PTS2 is normally regarded as an N-terminal signal, but surprisingly I could locate only 4 hits within the first 50 amino acid N-terminal regions of these sequences. There were only a total of 31 matches of this regular expression when it was scanned in the whole-length sequences. This low abundance rate could explain why this weak motif, having two not well conserved amino acids on either side separated by 5 “spacers”, could not be found by NestedMICA.

As an alternative, I ran Eponine on a dataset consisting of amino acid chunks matching the regular expression [RK][LVI].....[HQ][LA] of this motif. However, neither plotting ROCs to assess the obtained model’s discriminative power from other types of proteins, nor the principle component analyses (see methods) I have performed suggested any performance gain from using this model. This indicates that this particular less conserved motif could be found in other classes of proteins by chance, and therefore it is not disjunctive enough in localisation prediction.

On the other hand, although the C-terminal PTS1 motif (having the short but conserved “SKL” form) that is shown in Figure 3.4j was not present in the majority of peroxisomal proteins, whenever a motif hit was found in the far C-terminal region, its selectivity was high, namely, it was most of the time capable

of discriminating a peroxisomal protein from another type.

### 3.3.3 Golgi N-terminal transmembrane topology prediction statistics help in localisation prediction

Knowing the transmembrane topology of a protein contributes to its localisation determination, since most of the cytoplasmic proteins will not contain as many membrane-spanning regions as plasma membrane proteins, for example. I found that even for different proteins of the secretory pathway where transmembrane regions are abundant, this could be used as a distinguishing feature.

Golgi does not have an apparent targeting or retention signal, but I observed that TMHMM, which may not distinguish between a signal peptide (SP) and a transmembrane (TM) helix, predicted at least one TM helix for 91% of the sequences in the Golgi dataset, 86% of which were predicted to be crossing the membrane once, while only approximately half of the ER sequences had at least one predicted TM helix. An overwhelming majority (97%) of plasma membrane sequences were predicted to possess at least one TM helix, too, but these were distributed across the sequence unlike in the Golgi sequences. Figure 3.11 shows the expected number of amino acids among the first 60 N-terminal residues that fall within a transmembrane region as reported by TMHMM for different protein classes. We know that these N-terminal transmembrane domain predictions are most likely signal peptide (SP) sequences responsible for targeting the majority of secretory pathway proteins into the ER after their synthesis. Unlike the other types of secretory pathway proteins, most of the Golgi proteins have their predicted membrane-spanning regions containing between 15 and 25 amino acids,

with a strong length preference of around 20 amino acid residues. This observation is justified by a previous study that showed that changing the length of the transmembrane domain of Golgi or plasma membrane proteins affected their protein localisation (Munro, 1995). In short, when incorporated into the SVM as described in Section 3.2.4 structural properties, such as the number and length of predicted TM structures in the N-termini and as well as in full-length sequences, clearly help Lokum in identifying protein localisation.

### 3.3.4 Effect of amino acid composition

In this work, in addition to using other protein features I use amino acid composition, too. However, this is not associated with the intention of by-passing possible annotation errors with this choice; instead, it is mostly to make advantage of the biological fact that proteins in a certain compartment can possess similar macroscopic properties such as composition, possibly for better interacting with their environment. As mentioned in the introduction section of this chapter, many previous studies have used amino acid composition as a strong sequence-level attribute that can be used as a distinguishing feature in subcellular localisation prediction. I used normalised amino acid frequencies to convey this macro-molecular characteristics that would presumably be similar in proteins sharing a common compartment. Proteins in different localisations can bear different predilections for certain amino acid residues, as the plots in Appendix B demonstrate.

Using the first type of amino acid grouping suggested by Thomas & Dill (1996) (see page 77 in the Methods for more detail), instead of the 20 amino acid letters

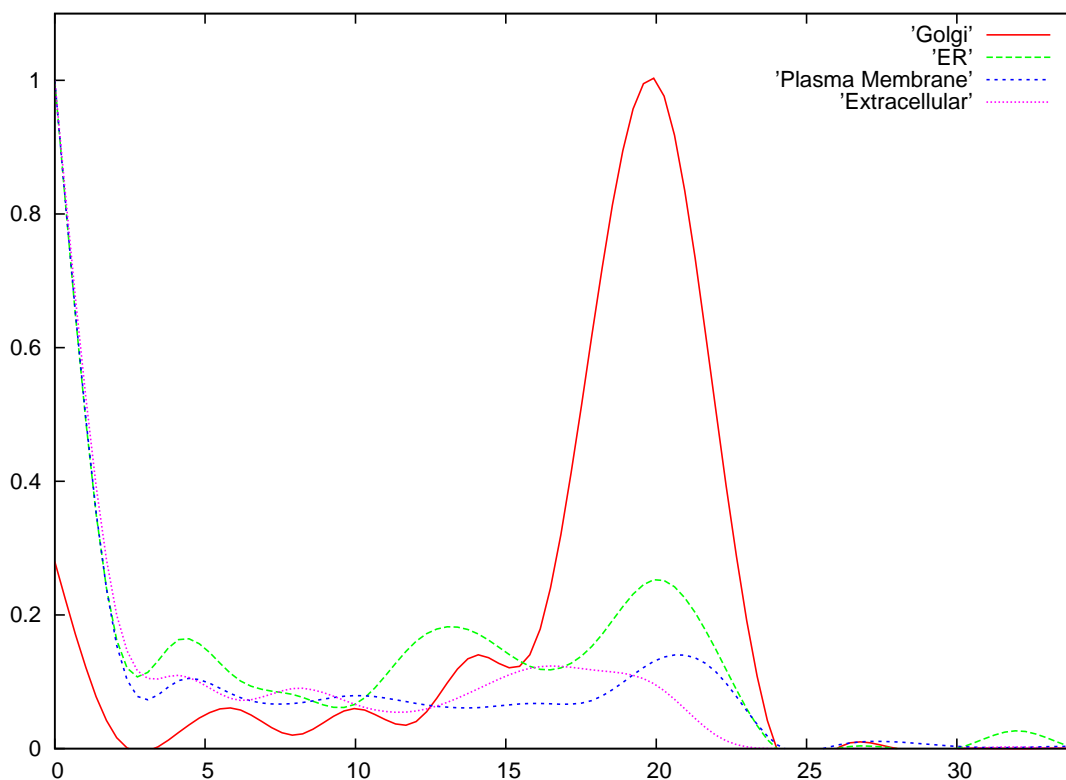


Figure 3.11: **The expected number of amino acids in the first 60 N-terminal residues to be part of a transmembrane region.** Plotted using a bin size of 5, the normalised histogram shows the predicted number of amino acids in the first 60 N-terminal residues lying within a transmembrane region as reported by TMHMM. Distributions for different types of secretory pathway proteins are shown. Plasma membrane proteins generally have a larger number of membrane-spanning regions spread across their entire amino acid sequence (see text), whereas Golgi sequences tended to have a single predicted TM helix in their N-termini, demonstrating a strong total length preference of around 20 amino acids.

in the calculation of composition, I was able to get a maximum correct prediction rate of 77.61% from the SVM using 5-fold cross validation, which is about 4% less than the result obtained from using all the amino acids in classifying the animal proteins. In addition to using composition, I also kept the other features like motif scores and transmembrane topology that I use in the general classifier. This group has 10 classes of amino acids (Table 3.3). Similarly, when I used the second type of amino acid grouping (3.3) from Thomas & Dill, where amino acids are categorised in 5 subgroups, the mean of the accuracy in the 5-fold cross validation tests was 76.54%. Finally, using the third type of grouping, also shown in Table 3.3, in which I mapped the 20 amino acids into 6 classes based on general physical and chemical properties of amino acids, I obtained an average correct prediction percentage of 75.81, in predicting the 9 animal protein localisation sets. These obtained figures are about 3-4% smaller than what I obtained by calculating the composition of each of the 20 amino acids without any grouping.

### 3.3.5 Lokum's performance

Table 3.4 summarises the performance of Lokum, in terms of the program's classification sensitivity (SN) and specificity (SP). Also, Matthew's Correlation Coefficient (MCC) values are given for Lokum, MultiLoc and PSORT in Table 3.5. SN, SP and MCC values were computed as explained in Section 3.2.7 on page 80. Individual cross validation sets used in the MultiLoc study were not available to enable me to perform a direct comparison. However, since Lokum is trained and evaluated using the same datasets of MultiLoc (Höglund *et al.*, 2006), for comparison I reproduced MCCs in the table for both MultiLoc and PSORT from

the MultiLoc paper where the latter programs are compared. For the SN and SP values of MultiLoc and PSORT please refer to the same article by Höglund *et al.* (unfortunately, this paper does not mention SN, SP and MCC values for all plant localisations).

#### 3.3.6 Contributions of different features

To better understand the individual contributions of using motifs, composition and structural information, I stratified the prediction system by using only a particular type of feature at a time. I counted the number of correctly predicted protein sequences by running 3 different SVM predictors that use only motifs, only amino acid composition, and finally only transmembrane structure information. Figure 3.12 shows the proteins that were independently classified correctly by a single predictor, by any two, or by three of them. The Venn diagram tells us that about a third of the correct predictions can be achieved by either using only motifs or by composition alone. This indicates that the amino acid composition can be thought of as partially representing some of the motif information and vice versa. 13.7% of the predicted proteins can be said to be the easiest to predict, because they could be classified by any of the SVMs. More than a quarter of the proteins were predicted successfully only by the SVM using motif scores. The SVM that was trained only with structural information had the least number of correct predictions (3.8%) that the other predictors could not correctly classify.



Version	Localisation	Total sequence	Lokum performance		
			SN	SP	MCC
Animal	plasma membrane	1238	0.85	0.95	0.86
	mitochondrial	510	0.76	0.73	0.71
	nuclear	837	0.79	0.74	0.72
	cytoplasmic	1411	0.75	0.82	0.70
	ER	198	0.79	0.68	0.72
	extracellular	843	0.87	0.90	0.85
	Golgi	150	0.86	0.71	0.77
	lysosome	103	0.88	0.57	0.71
	peroxisomal	157	0.77	0.30	0.46
Fungal	plasma membrane	1238	0.86	0.95	0.86
	mitochondrial	510	0.75	0.72	0.70
	nuclear	837	0.77	0.75	0.70
	cytoplasmic	1411	0.75	0.81	0.70
	ER	198	0.82	0.68	0.73
	extracellular	843	0.85	0.90	0.85
	Golgi	150	0.84	0.71	0.77
	vacuolar	63	0.86	0.24	0.45
	peroxisomal	157	0.75	0.30	0.46
Plants	chloroplast	449	0.76	0.56	0.62
	cytoplasmic	1411	0.59	0.79	0.56
	plasma membrane	1238	0.86	0.95	0.86
	mitochondrial	510	0.69	0.66	0.63
	nuclear	837	0.75	0.73	0.69
	ER	198	0.81	0.67	0.73
	extracellular	843	0.84	0.88	0.83
	Golgi	150	0.83	0.72	0.77
	vacuolar	63	0.86	0.24	0.45
peroxisomal	157	0.79	0.30	0.47	

Table 3.4: **Prediction performance summary for Lokum.** Sensitivity (SN) and specificity (SP) and Matthew’s Correlation Coefficient (MCC) values are given for Lokum. Lokum was trained and evaluated by 5-fold cross validation using the MultiLoc (Höglund *et al.*, 2006) datasets.

### 3.3 Results

Version	Localisation	Lokum		MultiLoc		PSORT	
		MCC	Correct%	MCC	Correct%	MCC	Correct%
Animal	p. membrane	0.86	81.73	0.76	74.6	0.73	59.9
	mitochondrial	0.71		0.83		0.58	
	nuclear	0.72		0.73		0.54	
	cytoplasmic	0.70		0.68		0.43	
	ER	0.72		0.60		0.11	
	extracellular	0.85		0.77		0.72	
	Golgi	0.77		0.53		0.04	
	lysosome	0.71		0.48		0.18	
Fungal	peroxisomal	0.46		0.44		0.25	
	p. membrane	0.86	81.67	0.86	74.9	0.78	53.9
	mitochondrial	0.70		0.88		0.58	
	nuclear	0.70		0.73		0.54	
	cytoplasmic	0.70		0.69		0.43	
	ER	0.73		0.60		0.13	
	extracellular	0.85		0.73		0.68	
	Golgi	0.77		0.60		0.04	
Plants	vacuolar	0.45		0.42		0.08	
	peroxisomal	0.46		0.43		0.25	
	chloroplast	0.62	78.92	0.85	74.6	0.50	57.5
	cytoplasmic	0.56		0.70		0.42	
	p. membrane	0.86					
	mitochondrial	0.63					
	nuclear	0.69					
	ER	0.73					
extracellular	0.83						
Golgi	0.77						
vacuolar	0.45						
peroxisomal	0.47						

Table 3.5: **MCCs and correct prediction rates for Lokum, MultiLoc and PSORT.** The shown MCCs for MultiLoc and PSORT were taken from Table 3 of the MultiLoc article (data not available for all plant classes).

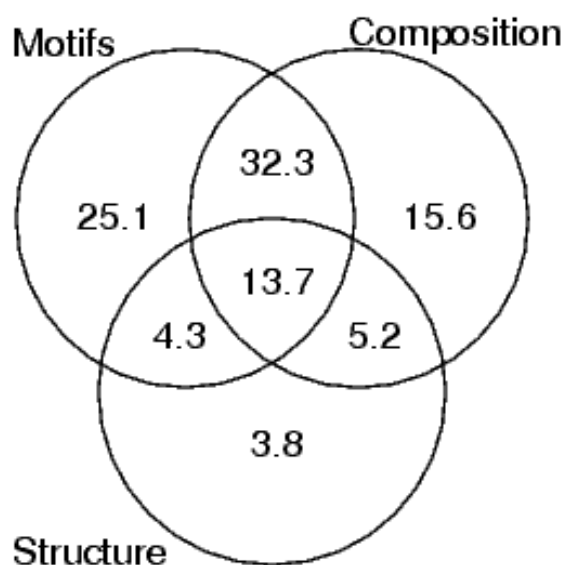


Figure 3.12: **Individual contributions of features used in the SVM.** This Venn diagram shows the percentage of proteins that could be correctly predicted by 3 individual SVM systems designed to use only motif scores, only amino acid composition, or only transmembrane statistics, respectively. The provided figures sum up to 100%, because only the distribution of proteins classified correctly at least by one predictor is given. The overlapping region between “Motifs” and “Composition” for example, indicates that amongst the proteins that could be predicted by at least one predictor, 32.3% of the “labeled” proteins could be successfully classified independently both by an SVM using only motif and another one using only composition information.

### 3.3.7 Contribution of disordered region predictions

Protein disorder regions are described and discussed in Chapter 4 (page 111) where I evaluated the use of disorder region statistics for use in sub-organelle localisation prediction. It has been suggested that inferring function improves when using patterns of native disorder in proteins (Lobley *et al.*, 2007). In order to assess the possible contribution of disorder prediction: i) I scanned the discovered localisation-related motifs in the predicted disorder regions to obtain a second set of motif scores, and ii) I considered the predicted disorder scores of sequence regions where a particular NestedMICA motif has a maximum score. Disorder region predictions were made using the RONN (Yang *et al.*, 2005) disorder prediction program (for the description of the software and methodology please see the dedicated Chapter 4).

However, adding these extra score sets (both at the same time or individually) to the SVM vectors resulted in no significant performance increase in the overall localisation prediction. After trying individual scores from both categories in different combinations, as performed by a systematic analysis, only a negligible maximum gain of around 0.01% could be achieved.

Using protein disorder predictions did not improve the overall prediction for the major localisation categories. This could be due to a number of reasons. Proteins can use different means to reach the same destination. Targeting into major cellular localisations can be achieved through general characteristics such as having a certain tendency in amino acid composition, which makes the disorder region statistics less effective for general localisation prediction. Nevertheless, as

shown in Chapter 4, knowing disorder regions could be useful for distinguishing proteins localised in different specific sub-organelle compartments.

## 3.4 Discussions

Computational prediction of protein localisation from amino acid sequence only is a challenging task not only because of some possible limitations in the methodologies, or even because of the lack of enough knowledge about the underlying biology. We know that proteins can migrate from a certain compartment to another, which does not permit a “one protein one localisation” correlation to always hold true. Besides, not all proteins have targeting signals, some are ‘piggy-backed’ and transported by other proteins which have the necessary signals (Wu *et al.*, 2000). Also, not all proteins from the same localisation categories show significant similarities in their general properties such as amino acid composition to enable one to make near-perfect predictions by only using these statistics. Therefore, one key factor in getting reasonable prediction accuracies lies in using as much relevant information as possible. When protein features such as localisation motifs, amino acid composition, or structural information etc. are used in combination, perhaps each bit would be characterising a certain number of protein classes better, but also their synergy would result in better overall prediction quality by possibly reducing some of the false positive predictions that individual feature components would otherwise produce.

Motifs like the N-linked glycosylation signal, one of the oldest known protein signals (Prosite id: PDOC00001), could be of great help in localisation prediction, even though they may not be directly involved in protein targeting. The N-

linked glycosylation process which normally takes place in the ER lumen aids us in predicting secretory pathway proteins when combined with the extra motifs found.

Representing motifs as PWMs rather than regular expressions is advantageous. As mentioned in the introduction, there are two major types of peroxisomal targeting signals (PTS). The first identified PTS is the C-terminal SKL-type signal. However, in some cases, it can take the form of a similar tripeptide, namely “KKL” (Takada *et al.*, 1990) and in some other eukaryotes it could be “SQL” (Purdue *et al.*, 1992), “NKL” (Lumb *et al.*, 1994; Oda *et al.*, 1987), or “SSL” (Motley *et al.*, 1995). Existence of many such possible variants clearly indicates that localisation motifs represented in regular expressions cannot be as efficient as probabilistic representations. PWMs, such as the PTS1 motif shown in Figure 3.4j, can potentially tolerate slightly differing forms by allowing a certain degree of sequence variation due to their probabilistic construction.

Although the *ab initio* Lokum does not use any database look ups to detect proteins matching a certain Prosite or NLSdb motif, its performance in assigning eukaryotic proteins into the correct localisation category was better for most of the localisation categories than the other multi-class predictors compared. I showed that by combining features including motifs represented as PWMs, amino acid composition and transmembrane topology statistics, one can get very reasonable (as high as 81%) prediction accuracies. As I demonstrated with the glycosylation motif example, protein motifs that are not directly involved in protein sorting could be used as secondary signals, too. In some cases, composition can substitute the information coming from a signal, but most of the time using direct biological

localisation signals along with composition and structure statistics proved to be more efficient.

By leaving out one sequence at a time and training a dedicated model by using the rest of the sequences to predict the localisation of that sequence, I was able to get an average correct prediction rate of 81.77% after repeating this procedure for each sequence in the entire dataset. This accuracy rate obtained by this “jack-knifing” methodology, however, only marginally differs from the reported correct prediction percentage of 81.73 (Table 3.5), which is obtained from the 5-cross validation tests done for the animals category. On the other hand, the overall performance was calculated to be 79.5%, 80.5%, and 81.1% when I used, 2, 3 and finally 4-fold cross validation, respectively. This indicates that using 5-cross validation was adequate and that there is no need to further increase the number of cross validation test sets.

## 3.5 Availability

The Lokum protein subcellular localisation predictor is available for public use through a web server which can be reached at:

<http://www.sanger.ac.uk/Software/analysis/lokum/>

It allows users to either paste some sequences into a text box or upload a file of protein sequences in fasta format. A screenshot of the server can be seen in Figure 3.13. Users can specify the Lokum prediction mode (animals, plants or fungi) that they want to use for their sequences.

I wrote the public Lokum predictor as a Java servlet. It runs on a “Resin” dynamic web server on a Linux cluster, but it has been also tested on different

## 3.5 Availability

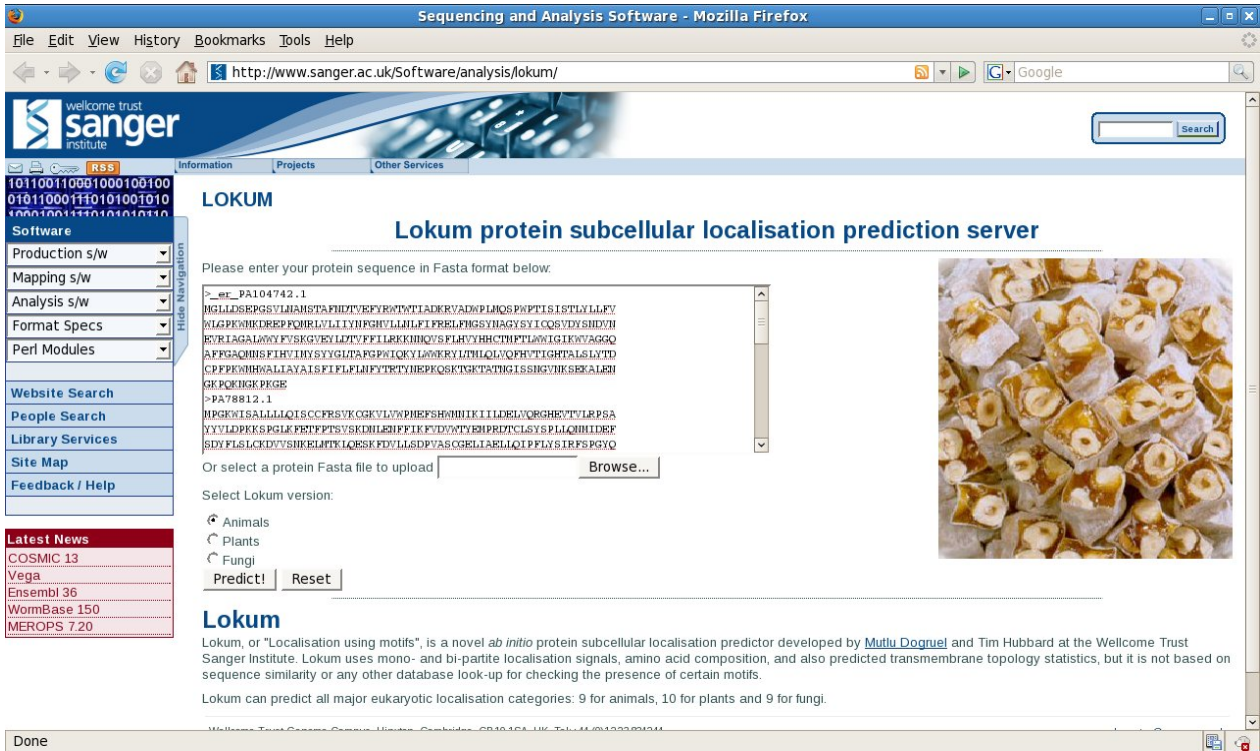


Figure 3.13: **Lokum prediction service hosted by the Wellcome Trust Sanger Institute.** Sequences must be uploaded either as a single fasta file, or entered into the text box in fasta format. Predictions are displayed in a separate page, following the submission of data.

platforms and using Tomcat, another popular web server. The servlet is based on the same command line version of Lokum, and also the same trained SVM classification model files. However, the prediction server works with a Java implementation of *libsvm* version 2.85, instead of the commonly used version written in the C programming language. No significant difference was observed between the predictions made by the two Lokum versions.

Interested users can download the protein motifs used in Lokum in Nested-MICA's XML format (XMS) from the Lokum home page.



# Chapter 4

## Discriminating nucleolar proteins from nuclear proteins: is it possible?

### 4.1 Introduction

In Lokum (Chapter 3), I tried to predict the conventional eukaryotic protein localisation categories which usually fall into one of the general localisation groups of cell organelles, cell membrane or extracellular space. Here, I investigate the possibility of fine tuning some of these predictions by trying to predict sub-organelle categories. As an example, I consider nuclear proteins, and try to classify proteins in this category under two labels: nuclear and nucleolar.

Proteins destined to the nucleus have to pass through the nuclear pores (Figure 4.1<sup>1</sup>). Nuclear pores could be imagined as holes piercing the impenetrable, hard nuclear envelope which, unlike the ER or plasma membrane, does not permit proteins to cross the membrane of the organelle directly regardless of whether

---

<sup>1</sup>The image, originally designed by Mike Jones (<http://en.wikipedia.org/wiki/User:Adenosine>) has been reproduced here under the “Attribution-Share Alike 2.5 Generic” license of Creative Commons.

they contain membrane spanning regions. This makes the translocation of nuclear proteins different from secretory pathway proteins, including that they do not contain any cleavable targeting signals. Nuclear localisation signals (NLS), which mediate the import of proteins into the nucleus, could be anywhere on the sequence, unlike the C-terminal ER retention signal (see Section 3.3.1.2 and Figure 3.8), for instance. They comprise short sequences of basic amino acids like Arginine (R) and Lysine (K) (see Figures 3.4h-i and A.1), and form short binding sites for recognition by other molecules. In 1986, Goldfarb *et al.* showed that mutations in the NLSs can impair nuclear localisation, but also, non-nuclear proteins can be targeted into the nucleus if artificial NLSs were added to them.

Previously, other subnuclear localisation compartments have been proposed for where RNA splicing related proteins (“nuclear speckles”) (Li & Bingham, 1991) accumulate, and also for small nuclear ribonucleoprotein (snRNP) components (“foci”) (Chang & Lin, 2001), but the major and most studied subnuclear compartment is the nucleolus. There is experimental evidence suggesting a sequence-dependent targeting into the nucleolus by means of Nucleolar Localisation Signals (NOSs) (Dang & Lee, 1989) which are similar in composition to NLSs. Because nucleolar proteins have to first pass through the nuclear pores just like any other nuclear proteins, it is quite reasonable to expect them to have similar sort of signals that mediate their passages. Furthermore, having no membrane around the nucleoli may suggest that localisation in nucleoli could actually be achieved through mainly molecular binding. In fact, in an experimental study some nucleolar proteins in mouse have been reported to carry only an NLS but no identifiable NOS (Maeda *et al.*, 1992). Therefore in addition to the presence

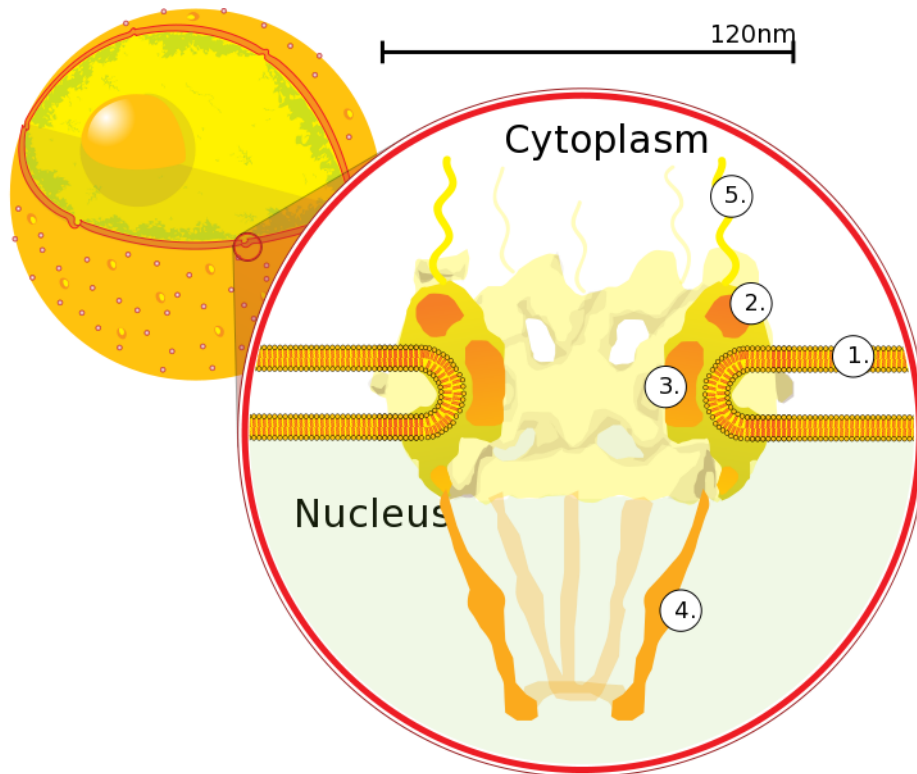


Figure 4.1: **Nuclear pore.** This schematic representation shows the nucleus, its nuclear envelope and a cross-section view of nuclear pores. Nuclear envelope is made of double membranes enclosing the genetic material in eukaryotic cells. Nuclear pores, crossing the nuclear envelope, allow water-soluble molecules to cross the nuclear envelope. Labels shown represent: 1 - Nuclear Envelope, 2 - Outer Ring, 3 - Spokes, 4 - Basket, and 5 - Filaments.

of NOSs, general protein properties such as amino acid composition could be important in nucleolar localisation.

The prediction of nuclear proteins is important because there are a lot of nuclear proteins in the cell, and difficult because the NLSs vary in sequence (Cokol *et al.*, 2000) and do not have specific positions. Prediction of nuclear proteins has probably begun with the multi-class localisation predictor PSORT (Nakai & Horton, 1999) which is based on many “if-then” type rules that comprise many biological features including discovered and known localisation signals (for a comparison of PSORT with Lokum see page 97). One of the more recent nuclear sequence prediction methods is PredictNLS (Cokol *et al.*, 2000). It predicts nuclear proteins by extrapolating from known NLSs which are listed in a specific database called NLSdb (Nair *et al.*, 2003). Initially, NLSdb had 114 experimentally determined NLSs that were obtained through an extensive literature search, but using ‘in silico mutagenesis’ this set was extended to 308 experimental and potential NLSs. PredictNLS is now part of a more general classifier, LOCTarget (Nair & Rost, 2004) that uses 4 specialised predictor programs: apart from NLSdb matches, it uses sequence homology (LOChom), SWISS-PROT keywords that are strongly correlated with localisation (LOCKey), and hierarchical support vector machines (LOCnet). Another dedicated nuclear sequence predictor, NucPred (Brameier *et al.*, 2007), has been recently developed to predict proteins that spend at least some time in the nucleus. NucPred is based on regular expression matching of NLSs and multiple program classifiers induced by genetic programming, and has similar overall prediction sensitivity and specificity with PSORT and PredictNLS. Predictors involving nuclear proteins also include NetNES (la Cour *et al.*, 2004)

that predicts nuclear export signal containing proteins.

While there are several dedicated tools that can directly predict or help identifying nuclear proteins, no particular prediction algorithm has been available that can predict proteins destined into the nucleolus or that can distinguish nucleolar proteins from nuclear proteins. Nevertheless, there has been studies to derive a knowledge-base that could be useful in predicting nucleolar proteins (Leung *et al.*, 2003), which generally suggested the use of amino acid and peptide composition and sequence homology information across different species.

### 4.1.1 Disordered protein regions

Natively unstructured regions are a common feature of eukaryotic proteins and many proteins have such regions with no well-defined 3-D structures in their native states (Dunker *et al.*, 2000). These natively unfolded protein regions could be involved in molecular recognition, and they can occasionally take regular forms when functioning. The first evidence came from a study carried out by Alber *et al.* in 1983, where it was concluded that the structure analysis of a complex, triose phosphate isomerase-substrate, had shown that a mobile region of 10 amino acids becomes ordered when an associated ligand binds. Disordered-to-ordered transition patterns can allow natively unstructured, related proteins to make formations (Weinreb *et al.*, 1996). However, these type of interactions involving disordered regions are not limited to only protein-protein interactions, and could be observed in protein-dna, enzyme-DNA, receptor-ligand interactions as well (Huber, 1979).

Dunker *et al.* (2000); Wright & Dyson (1999) showed that intrinsically unstructured protein regions are important regarding protein function. Lobley *et al.* (2007) directly used predicted disorder patterns successfully to improve protein function prediction. In Lokum, however, using disorder prediction did not improve the prediction of general localisation categories (see 3.3.7), so in this chapter, I try to address the potential contribution of protein disorder in distinguishing proteins localised in nucleoli from the rest of the other nuclear proteins, where I also used the features used in Lokum (Chapter 3).

### 4.1.2 Protein disorder region prediction

PONDR<sup>®</sup> is one of the best-known tools to predict disorder (Garner *et al.*, 1999; Li *et al.*, 1999; Radivojac *et al.*, 2003, 2004; Romero *et al.*, 2004). It uses pattern recognition techniques employing a set of attributes which are based on biological knowledge. Examples of other disorder software are FoldIndex (Prilusky *et al.*, 2005), DisEMBL (Linding *et al.*, 2003a), GlobPlot 2 (Linding *et al.*, 2003b), DISOPRED2 (Ward *et al.*, 2004), and Prelink (Coeytaux & Poupon, 2005).

The protein disorder prediction category has been introduced in the fifth “Critical assessment of methods of protein structure prediction” (CASP) competition (Cozzetto *et al.*, 2005, 2007; Soro & Tramontano, 2005; Valencia, 2005), with the participation of the mentioned programs and several others.

A program developed in 2005, RONN, has been recently compared with most of the notable CASP participants in the disorder category (Yang *et al.*, 2005) on an official CASP assessment dataset which contains 159 proteins sequences with experimentally determined disorder regions. Table 4.1 summarises the per-

formances of the 8 compared programs, with DisEMBL being compared using three different versions of the program. In addition to the traditional assessment measures sensitivity (Equation 2.3), specificity (Equation 2.4) and Matthew’s Correlation Coefficient (Equation 2.5), in CASP, a new weighted score (CASP-S) (Jin & Dunbrack, 2005) was used which was defined as:

$$CASP-S = \frac{100(w_{TP}TP + w_{FP}FP + w_{TN}TN + w_{FN}FN)}{TP + FP + TN + FN} \quad (4.1)$$

where  $w_{TP}$  stands for the number of disordered residues divided by the total number of residues, and so on. ( $w_{FN}$  was taken as  $-w_{TP}$  and similarly,  $w_{FP} = -w_{TN}$ ).

Also, the developers of RONN added yet another measure in their performance assessment, probability excess:

$$Prob. \ excess = \frac{TN \ TP - FN \ FP}{(FN + TP) + (TN + FP)} \quad (4.2)$$

Because of its reported reasonably good performance over the other predictors and availability as a stand-alone application I chose RONN for performing disordered protein region predictions.

## 4.2 Materials and methods

### 4.2.1 Datasets

The first proteins annotated as “nucleolar” came from mass-spectrometry studies (Andersen *et al.*, 2002, 2005; Scherl *et al.*, 2002). Recently, the list of nucleolar proteins, which have been previously identified mainly through mass-

spectrometry, has been extended by a protein-protein interactions approach (Hinsby *et al.*, 2006).

The nuclear and nucleolar protein sequences used in this study were downloaded from the LOCATE mouse protein sequence database (Fink *et al.*, 2006). LOCATE is a well curated, web-accessible database containing descriptions for the membrane organisation and subcellular localisation of FANTOM proteins. The FANTOM (Functional Annotation of the mouse) consortium (Carninci *et al.*, 2005; Maeda *et al.*, 2006) aims at providing the ultimate characterization of the mouse transcriptome. Only full length proteins from the FANTOM-3 project are present in LOCATE.

In LOCATE, I only considered protein annotations that are verified either by experiments or from literature. Among these, I picked nuclear (GO id:0005634<sup>1</sup>)

<sup>1</sup><http://www.ebi.ac.uk/ego/GSearch?query=0005634&mode=id&ontology=component>

Method	SN	SP	MCC	Casp-S	Prob excess
RONN	0.603	0.878	0.395	9.33	0.481
DISOPRED2	0.405	0.972	0.470	7.81	0.377
PONDR®	0.557	0.816	0.278	7.22	0.373
DisEMBL(hot)	0.492	0.840	0.260	6.43	0.332
DisEMBL(465)	0.334	0.981	0.437	6.10	0.315
FoldIndex	0.488	0.811	0.224	5.79	0.299
PreLink	0.237	0.947	0.219	3.55	0.183
GlobProt	0.372	0.811	0.140	3.54	0.183
DisEMBL(coils)	0.740	0.424	0.104	3.19	0.165

Table 4.1: **Performance measures calculated from the blind testing of nine disorder prediction methods against the main blind test set of 80 proteins of CASP 6.** The performance measures are sensitivity (SN), specificity (SP), Matthews correlation coefficient (MCC), CASP S-score and probability excess (Prob. excess). This table is re-produced from Yang *et al.* (2005).



and nucleolar (GO id:0005730<sup>1</sup>) protein sequences. I removed sequences annotated as both nuclear and cytoplasmic etc. to have two datasets at the end, one consisting of nucleolar proteins and another one consisting of only nuclear proteins. Some nucleolar proteins could also be annotated as nuclear, as they can spend some time in the nucleus, too. The final list of protein IDs used in this study can be found in Appendix C.

Using the CD-HIT (Li & Godzik, 2006) sequence clustering program to reduce the maximum sequence identity between any two sequences to 30%, the nuclear mouse protein dataset downloaded from LOCATE was reduced to 386 sequences from an initial number of 715. Similarly, the nucleolar set which initially had 815 sequences was filtered to allow a maximum identity of 30% between any two sequences at the end, which resulted in 397 sequences. One third of each dataset was reserved for testing purposes, while the remaining sequences were used in motif discovery.

Protein-capable NestedMICA (Doğruel *et al.*, 2008) was run on randomly chosen 257 nuclear and 265 nucleolar sequences, leaving the rest of the sequences in the datasets for test purposes. In order to detect possible motifs at both termini, N-terminal amino acid chunks of length 20 were compiled from the nucleolar and nuclear sequences. Similarly, two more datasets were produced which contained 20aa C-terminal sequences from both types. NestedMICA has been run on the nucleolar and nuclear training datasets containing whole-length sequences, 20 N-terminal amino acid chunks, and finally 20 C-terminal peptides.

---

<sup>1</sup><http://www.ebi.ac.uk/ego/GSearch?query=0005730&mode=id&ontology=component>

### 4.2.2 Training background models for nucleolar and nuclear datasets

Two NestedMICA background models were trained using a similar strategy described in the background model related sections on pages 35 and 41. However, particularly the nuclear motifs (shown in Figure 4.2) obtained using these background models were quite short and surprisingly not rich in residues like Lysine or Arginine which are expected to be abundant in the core parts of the NLSs. Nucleolar motifs were quite short, too, and they only possessed strong Arginine residues but no Lysines (Figure 4.2). This could have resulted because of using relatively simple, zero order background models which are trained on the relatively small number of sequences in these two datasets (in Chapter 2 I showed that using order-1 background models would be better than using an order-0 background, but if there is enough data to train it).

An alternative, third background model was trained using 438 redundancy reduced cytoplasmic sequences (see page 55). Nuclear proteins are transferred into the nucleus by the means of some molecules binding to their NLSs. Therefore, as previous studies have shown, for example by Goldfarb *et al.* (1986), if these signals are altered it is likely that a protein will remain in the cytoplasm and will not be able to be carried into the nucleus. Thus, the uninteresting, non-localisation segments of nuclear and other sub-nuclear proteins could best be represented by a cytoplasmic background. Indeed, when I ran NestedMICA with this first order cytoplasmic background model consisting of 4 mosaic classes on the individual nuclear and nucleolar protein datasets that have been created, the results were much more promising. Motifs obtained from each background model







Localisation	Sequence segment	Motif
a) Nucleolar	N-terminal	
b) Nuclear	N-terminal	
c) Nucleolar	Entire	
d) Nuclear	Entire	
e) Nucleolar	C-terminal	
f) Nuclear	C-terminal	

Figure 4.2: **NestedMICA motifs discovered from nuclear and nucleolar datasets.** NestedMICA was run on two sets: nuclear and nucleolar datasets. In each run, it used a dedicated background model trained with the corresponding dataset. Figure 4.4 shows a set of “better motifs” discovered using another (cytoplasmic) background model trained with more sequences.

were assessed in terms of their performances to separate nuclear and cytoplasmic proteins, and it was actually when I used this cytoplasmic background model that they better discriminated the two classes, rather than when I tried the two background models trained on nuclear and nucleolar sequences.

### 4.2.3 Running RONN

The RONN protein disordered prediction program (Yang *et al.*, 2005) was run with the “short output” command line options on a Linux server. BioJava scripts were written to parse the output of the program and perform the statistics. A score of greater than 0.5 was considered as a disordered prediction, as recommended in the RONN manual. Figure 4.3 shows an example plot drawn according to RONN predictions from a nuclear sequence, where RONN produces disorder scores for each amino acid position. RONN version 3 was obtained by personal communication with the program’s developers.

### 4.2.4 Training the SVM

As in previous chapters, a popular Support Vector Machine (SVM) implementation, *libsvm* (Chang & Lin, 2001), was used in the task of classifying nuclear and nucleolar proteins. 10-fold cross validation was applied, and I used a radial-basis kernel function whose gamma ( $\gamma$ ) and C penalty parameters were systematically optimised.

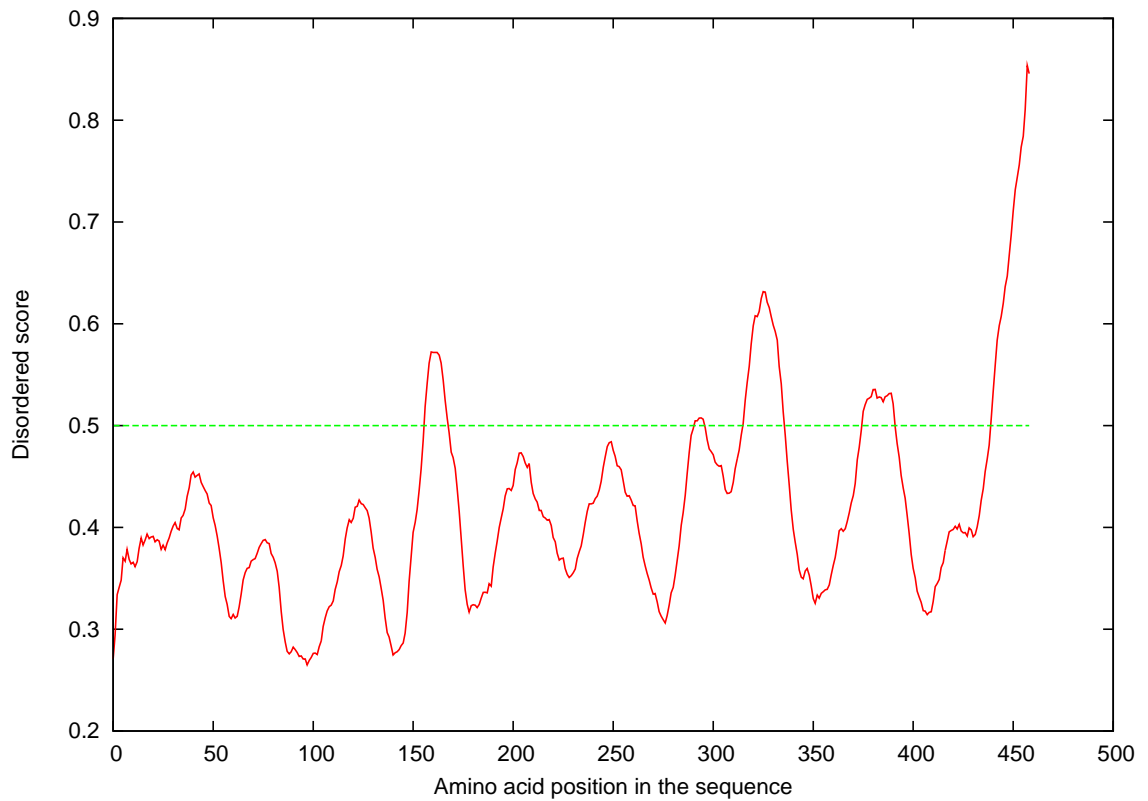


Figure 4.3: **A protein disordered region plot based on RONN predictions.** The plot shows the disorder score of a nuclear sequence of length 459, as an example. RONN produces a score between 0 and 1 for every single amino acid position across a sequence. A score above 0.5 indicates a disordered residue or a region. As the plot illustrates, a sequence can have multiple disordered regions (5 in this example, with a strong disordered sequence chunk at the C-terminal end).

## 4.3 Results

Nucleolar proteins possess NOSs (Dang & Lee, 1989) to enter into the nucleus from the cytoplasm. Figure 4.4 shows some of the nuclear and nucleolar protein motifs reported by NestedMICA. NestedMICA was run on 3 datasets for each localisation class: a dataset consisting of full-length sequences, and two datasets of 20aa N- and C-terminal sequence chunks, respectively. The most striking difference between the nuclear and nucleolar sequence motifs is how nucleolar motifs are enriched with Arginine (R) and Lysine (K) amino acid letters over the nuclear motifs discovered in the N- and C-terminal regions. NLSs have been known not to have specific positions and can be located across the entire primary structures of nuclear proteins; however, these results suggest the possibility that nucleolar proteins, unlike nuclear proteins, have stronger NLS-like motifs (NOSs) in their both N and C termini. We scanned and scored both the N- and C-terminal nucleolar motifs (Figure 4.4) in the corresponding 20 aa N or C terminal regions of both nucleolar and nuclear proteins to see if we can observe any difference in the score distributions. The highest scores obtained from these nucleolar motifs both in the nuclear and nucleolar sequences are plotted in Figure 4.5.

By using a simple SVM consisting of input vectors formed with only the scores of the N- and C-terminal motifs (4 in total) shown in Figure 4.4, it was possible to classify 65.4% of the proteins correctly into the two classes of nuclear and nucleolar localisations. Adding amino acid composition to the four motif scores, I was able to increase the performance up to 74.5%. Using only the 20-dimensional amino acid composition rates was sufficient to predict 73.5% proteins correctly.







Localisation	Sequence segment	Motif
a) Nucleolar	N-terminal	
b) Nuclear	N-terminal	
c) Nucleolar	Entire	
d) Nuclear	Entire	
e) Nucleolar	C-terminal	
f) Nuclear	C-terminal	

Figure 4.4: A selection of the protein motifs recovered by NestedMICA from a set of nuclear and a set of nucleolar proteins, using a cytoplasmic background. N-terminal motifs shown were reported from the first 20 N-terminal amino acid regions. Similarly, the C-terminal motifs were searched within the last 20 amino acid regions. Other motifs indicated by the “Entire” segment were discovered when full length sequences were used.

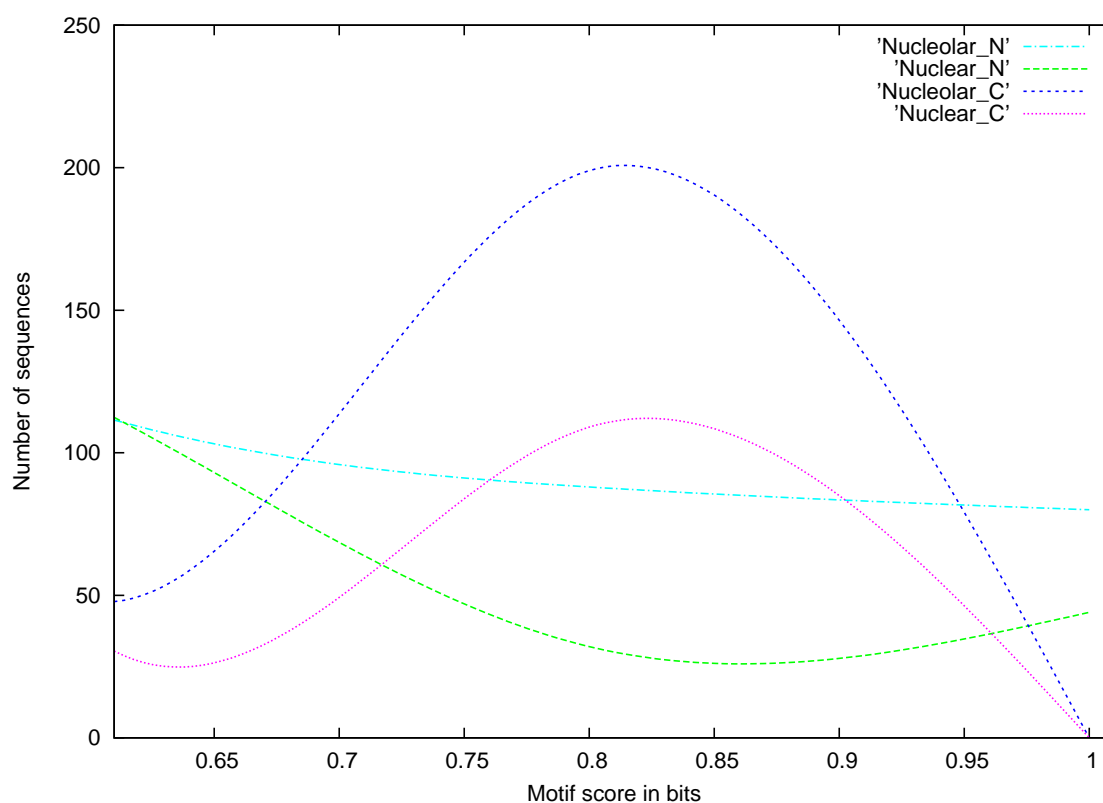


Figure 4.5: **Score histograms for N- and C-terminal nucleolar motifs.** Both nuclear and nucleolar sequences were scanned using the N-terminal nucleolar motif (Figure 4.4a). Similarly, the C-terminal nucleolar motif (Figure 4.4e) was scored in both types of datasets. Scores shown on the x-axis correspond to the best matches within the relevant 20 amino acid long N or C terminal chunks. The C-terminal motif generated Gaussian-like distributions when scored in the last 20aa C-terminal regions, however, this motif is clearly more abundant in the nucleolar C-termini. The other two curves indicate that the N-terminal regions are less abundant in terms of the N-terminal nucleolar motif, but still, this motif was less frequent in the N-termini of nuclear proteins than the N-termini of proteins localised in the nucleoli.



Using transmembrane (TM) statistics reported by TMHMM (Krogh *et al.*, 2001) (reported “features” are summarised in 3.2.5) improved the prediction accuracy in Lokum. Furthermore, nuclear proteins, in theory, should possess a larger number of TM helices, compared to the nucleolar sequences which are confined to the centre of the nucleus and less likely to have TM helices. In fact, running TMHMM on the entire sequences in both datasets to compare them in terms of their number of predicted residues that possibly lie in a TM helix (Figure 4.6) revealed that this feature can significantly improve predictions. With the addition of the two more types of predicted TM statistics mentioned in Section 3.2.5, the correct prediction rate increased to 77.14%. When I used the three TMHMM statistics alone, the correct prediction rate was 64.4%.

Finally, after adding the bipartite NLS motif (Figure 3.10, page 92) that we obtained using the combinatorial approach involving both NestedMICA and Eponine, the overall correct classification rate increased to as high as 78.42%. Sequences used in the motif discovery were not used in training and testing of the SVM. Due to the relatively low number of sequences in both datasets (783 in total), this particular SVM was trained and tested using 10-fold cross validation (see Methods).

That amino acid composition helped us in making more correct predictions implies there is a certain degree of bias in composition even between the similar classes of nuclear and nucleolar proteins, which could be associated with the possibility that nucleolar proteins have slightly different compositional preferences than the other proteins in the nucleus so as to allow them to be packed more tightly to form the nucleolus. Figure 4.7 shows the compositional differences

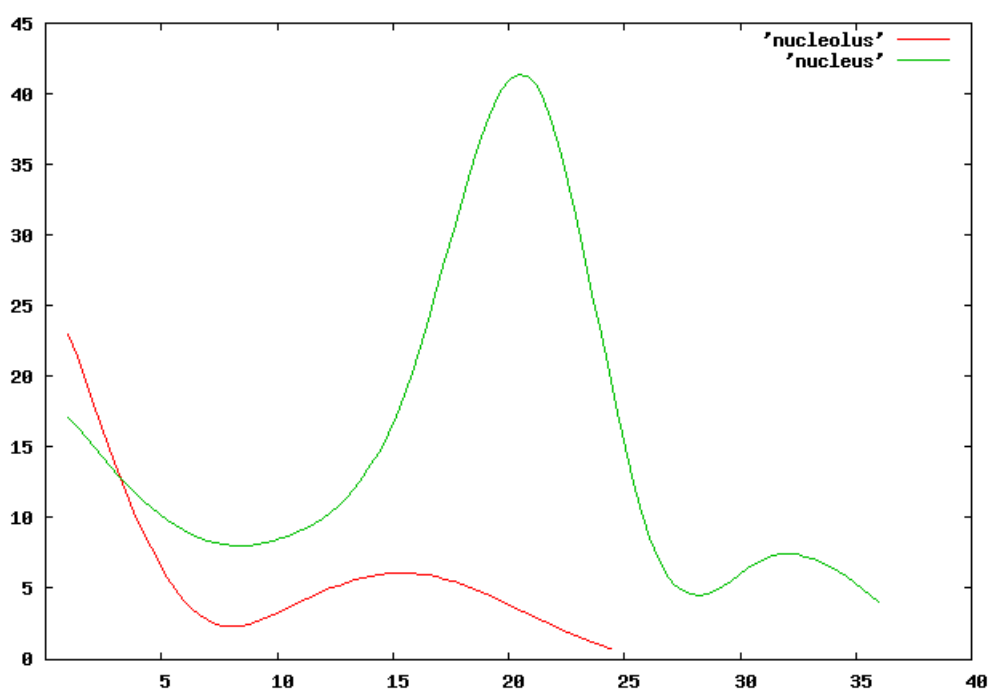


Figure 4.6: **Distributions of amino acids predicted to be within TM helices in nuclear and nucleolar proteins.** TMHMM (Krogh *et al.*, 2001) was run on the entire nucleolar and nuclear protein sequences. The curves show the total number of sequences (y-axis) having a certain, predicted total number of amino acid residues in their sequences that fall in a membrane-spanning region, for nucleolar (red), and nuclear (green) proteins. According to this plot, most nuclear proteins have around 20 amino acids within their TM helices all together. A bin size of 5 amino acids was used to plot the frequencies, and the curves were smoothed by the “cubic splines” algorithm.

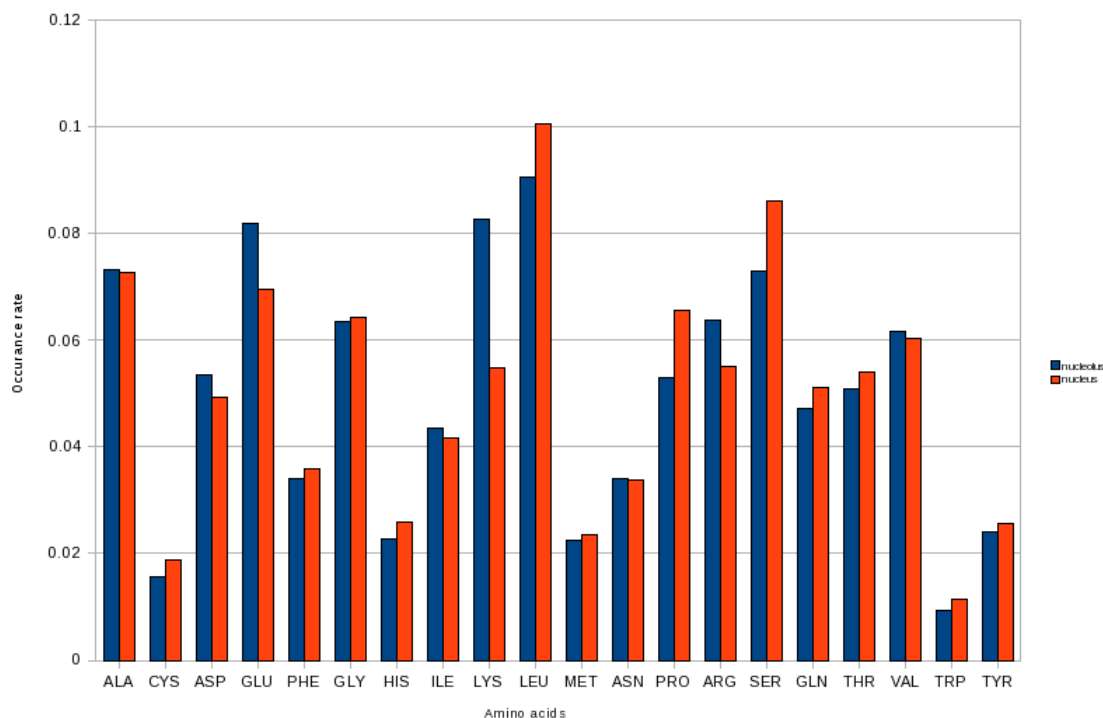


Figure 4.7: **Differences between nucleolar and nuclear proteins in terms of their amino acid compositions.** This statistics was obtained using the nuclear sequences datasets consisting of 386 sequences and the nucleolar sequence set having 397 sequences (see Materials and Methods). The most noticeable difference is how nucleolar proteins are enriched with Lysine (K) over nuclear proteins.

between the two types of proteins localised in nuclei and nucleoli. A similar figure showing the comparison of nucleolar and nuclear proteins in terms of their amino acid composition has been reported previously by [Leung \*et al.\* \(2003\)](#). As seen in Figure 4.7, the most notable difference is how nucleolar proteins are enriched with Lysine (K) over nuclear proteins. While most other amino acid composition rates were more or less identical, nuclear proteins had a larger number of the nonpolar amino acids Leucine (L) and Proline (P), and the polar Serine (S) than the nucleolar proteins.

The fact that our motif finder discovered some motifs (Figure 4.4) from the nucleolar protein set does not necessarily mean that these motifs can not be found in the nuclear proteins, and vice versa. As can be seen in Figure 4.8, which, as an example, shows the score distributions of motif c of Figure 4.4 for both types of protein sequences, some nuclear proteins may also contain this particular K- and R-rich motif despite that it was originally discovered in the nucleolar set. However, the histogram plot suggests that mostly high scoring instances of this motif are more abundant in nucleolar proteins compared to the best hits of the motif in sequences localised in the nucleus.

In addition to demonstrating that terminal regions of nucleolar proteins could be more biased towards positively charged residues, I investigated whether nuclear and nucleolar proteins differ in terms of their disordered region distribution. 41.99% of the amino acids in the nucleolar proteins set and 41.87% of the amino acids in the nuclear proteins set were predicted as disordered by the RONN software. This indicates that there is no significant difference in terms of the number of residues falling into a disordered region between both types of proteins. However, there is a difference about what constitutes these disordered regions: it turned out that these disordered regions are enriched more with charged residues in proteins localised in the nucleolus over proteins of the nucleus after some tests I performed with some of the motifs found.

Using motif c of Figure 4.4 to scan only both types of sequences, I observed that a larger number of disordered regions in the nucleolar sequences contained this motif than disordered regions in the nuclear proteins. Figure 4.9 shows the normalised frequency distribution of strong hits of this charged-residue rich

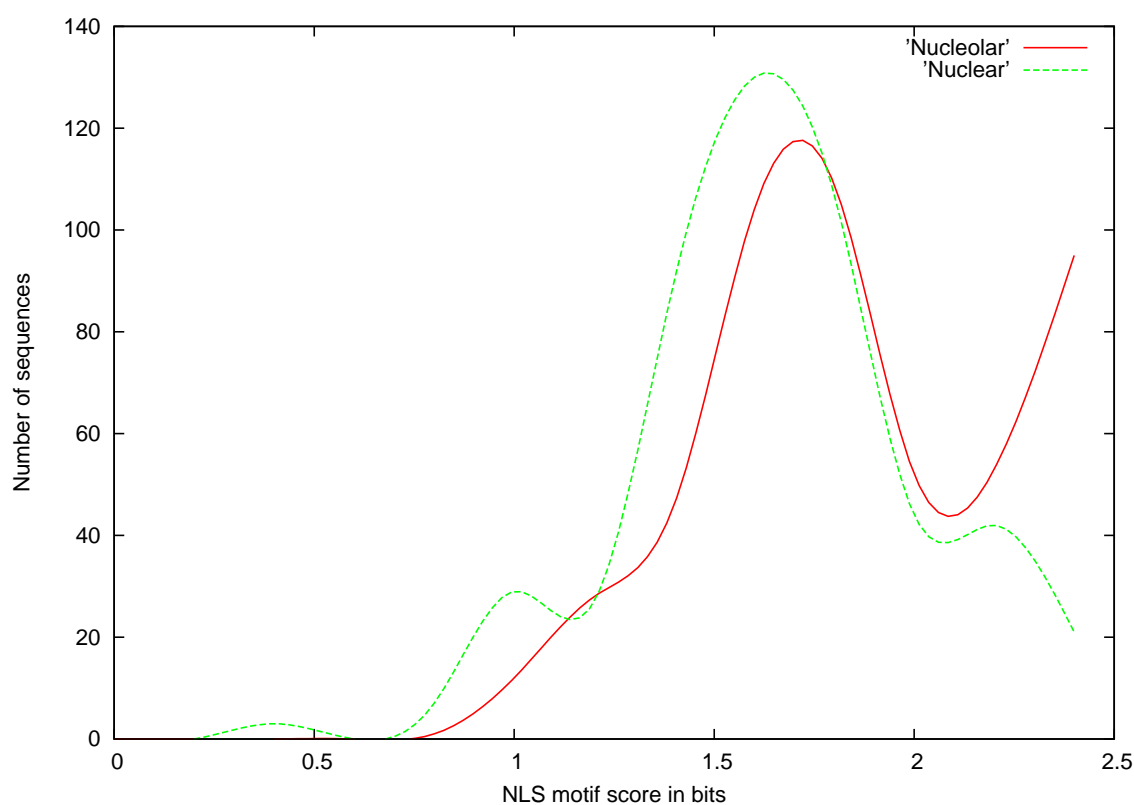


Figure 4.8: **Score distribution of a core nucleolar motif within nuclear and nucleolar proteins.** Motif scores shown on the x-axis are given in information bits, for the best match per sequence. The y-axis indicates the number of sequences for nucleolar (red line) and nuclear (green dashed line) featuring this motif with different scores. For plotting the histogram, 500 nucleolar and 500 nuclear sequences that were sampled randomly from the original datasets have been used.

motif within the predicted disordered regions for both types of sequence classes. Sequence regions scoring less than an empirically chosen value of 1.8 were not considered as true NLS matches (Figure 4.8). However, a second similar analysis performed by using another motif, which was discovered from a general nuclear localisation dataset in the previous chapter (Figure 3.4i), revealed that even when we consider the entire range of scores without using any threshold it is still possible to observe the same kind of tendency of finding more NLS motifs within disorder regions (Figure 4.10).

Given that there is a tendency in nucleolar proteins to possess “K & R”-rich motifs more abundantly within their disordered regions compared to nuclear proteins, I investigated whether this bias could be used in a prediction system. The SVM that was built initially to distinguish nuclear proteins from nucleolar proteins was modified so as to allow us to test this phenomenon. To this end, firstly, I added to the SVM the best scores of those core NLS signals (represented as a PWM in part c of Figure 4.4) that fall into a disordered region, excluding other potential motif hits in the rest of the sequence regions. Secondly, I added the predicted disordered scores of sequence regions featuring a core nucleolar motif, such as motif c of Figure 4.4. Unfortunately, both approaches, when used separately or together, failed to provide a substantial increase in the SVM’s performance, meaning that their potential contributions are somehow already achieved by the other used features including NLS motif scores and amino acid composition etc. Using general disordered statistics for each sequence, such as the number of disordered blocks per residue and the ratio of amino acid residues predicted as disordered to the total number of residues in a sequence, resulted

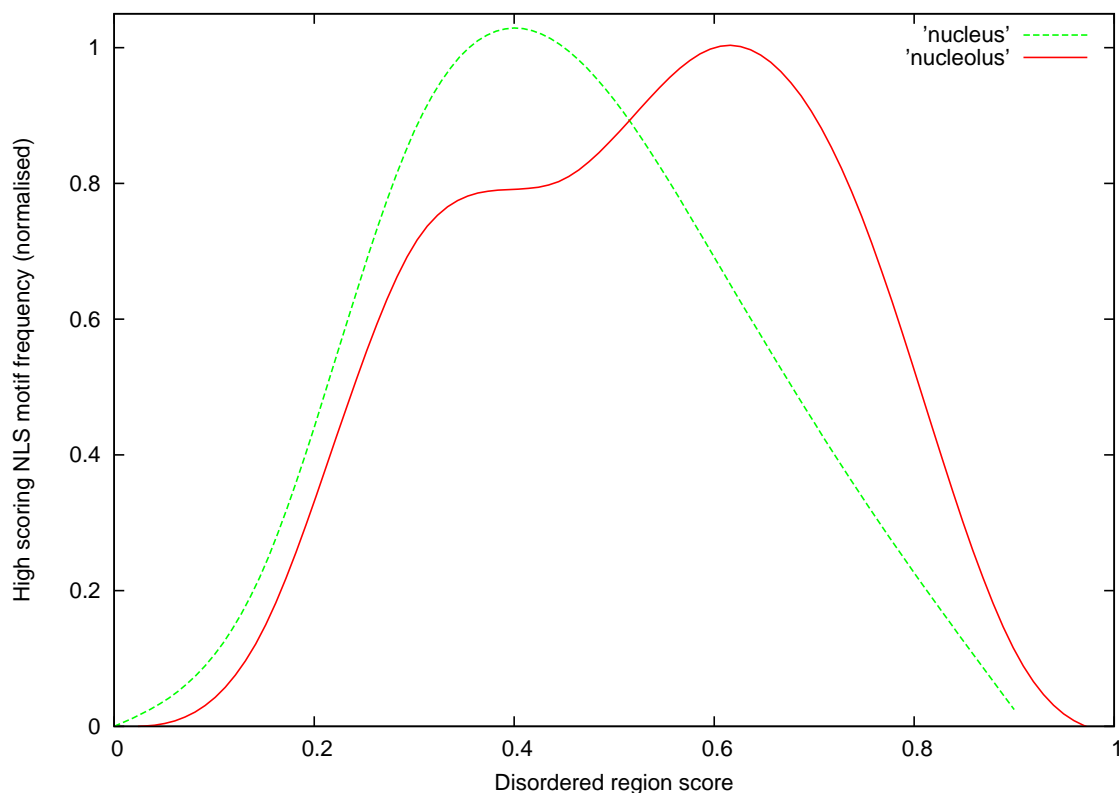


Figure 4.9: **A larger number of nucleolar localisation motif hits fall in disordered regions, compared to the NLS motifs in disordered regions of nuclear proteins.** The y-axis corresponds to the normalised frequencies, while the x-axis represents the disorder region scores as reported by RONN. A score of greater than 0.5 indicates a predicted disorder region. The dashed green curve represents nuclear proteins which show a normal distribution around a score of 0.4, while the solid red curve shows the histogram for nucleolar sequences having a tendency to contain more number of the NOS within their disordered regions.

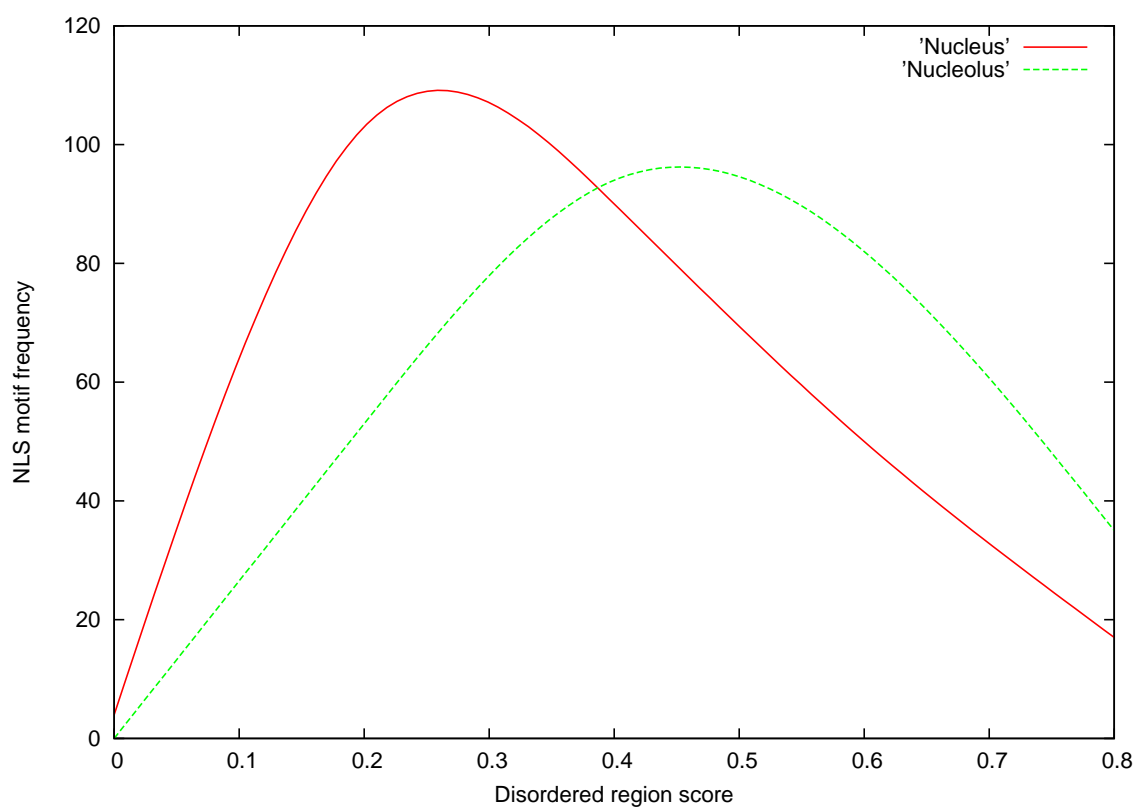


Figure 4.10: **Generally a larger number of NLS signal hits can be found in disordered regions of nucleolar proteins compared to nuclear sequences.** The y-axis corresponds to the frequencies of motif hits, while the x-axis represents the disordered score regions as reported by RONN. A score greater than 0.5 indicates a predicted disordered region.



in the same maximum correct prediction percentage (78.42%) that I obtained without using the disorder-related scores (see above). At the end, scores associated with disorder were not included in the SVM, as this did not improve the performance, although the nucleolar sequences showed a bias to possess a larger number of “K & R”-rich motifs in their disordered regions.

## 4.4 Discussions and conclusions

Using the observation that nucleolar proteins tend to contain a larger number of charged residues in their disordered regions was not particularly helpful in automatic classification of nuclear and nucleolar proteins. Instead, using these motifs directly without considering disordered regions to score proteins was more effective. In addition to using the reported motifs found in the terminal regions of nucleolar and nuclear sequences, incorporating amino acid composition in the SVM proved useful, as in predicting major localisation categories (see previous chapter). Thus, despite being confined by the nuclear membrane and sharing similar characteristics, there are significant differences in amino acid compositions between the members of these two types of proteins.

Loop regions and regions with no specific secondary structure in proteins do not have to be disordered necessarily. A disordered region means that that region has the capacity to change into an ordered state when needed, unlike, for example, some loop regions which can not become “ordered”, that is, have a certain structure and shape.

It is not very surprising to have observed that proteins forming the subnuclear compartment nucleolus are rich in charged amino acids like K and R, and that

they are more abundant in regions predicted to be disordered. It has been shown that aromatic amino acids like Tryptophan (W), Tyrosine (Y) and Phenylalanine (F) are less likely to be found in long disordered regions (Kissinger *et al.*, 1995), because these amino acids usually have a strong interaction capability to develop a structure, and thereby they inhibit disorder (Burley & Petsko, 1985). It has also been observed by the same groups that charge imbalance in protein sequences tends to favour disorder. But to find out that there are more of these charged residues in disordered regions of nucleolar proteins compared to nuclear proteins was surprising. This can be explained, to a certain extent, by the speculation that nucleolar proteins have to behave like any other nuclear proteins while traversing the nuclear pore to enter into the nucleus, but after that point, most probably their disordered regions which potentially convey the extra signals of nucleolar localisation signals (NOS) involved in their transport into their subnuclear destination, become more ordered and functional.

Unfortunately, good quality and reliable localisation annotation is too limited to satisfactorily study sub-localisation classes such as nucleolar or mitochondrial membrane proteins. Also, there can always be annotation errors in the datasets used. I tried to minimise these data related problems by choosing manually annotated and well curated datasets. To avoid a potential bias in predictions, sequence identity was lowered to a maximum of 30% by using a clustering algorithm (see methods). Another problem stemming from the underlying biology is that some proteins can be functioning in more than one compartment. However, even if the datasets contain such protein sequences, statistically it should still be possible to retrieve the general characteristics representing an individual group.

In the case of motif finding, for example, a few sequences coming from different types of protein localisations or those having multiple possible localisations should not prevent NestedMICA from finding the overexpressed, representative sequence motifs.

In spite of possible errors and data related limitations, I think the observation that disorder regions have more charged residues in nucleolar proteins, the compositional differences between the two classes, and finally the motifs found in the terminal sequence regions to distinguish nuclear and nucleolar proteins are promising results that can be used for discriminating nucleolar proteins from other nuclear proteins, as the tests indicated.

# Chapter 5

## Predicting protein transmembrane topology and signal peptides: An HMM approach with a new parameter optimisation strategy

### 5.1 Introduction

#### 5.1.1 The aim of this study

In Chapter 3, we have seen that using predicted protein transmembrane (TM) topology statistics, such as the fraction of N-terminal amino acids lying in a TM helix and the number of TM helices, improves subcellular localisation prediction. In this chapter, I investigated the possibility of developing an alternative to TMHMM (Krogh *et al.*, 2001) that was used in Lokum.

In this chapter, I also introduce a new strategy to optimise hidden Markov model (HMM) transition probabilities, based on nested sampling. This is an alternative to the classical approach of Baum-Welch optimisation procedure (see Section 1.3.1).

I tested this new methodology in optimising transition probabilities of an HMM that tries to automatically annotate a set of given sequences with their most probable TM topologies, and presence of SPs. This prediction is a mapping procedure of the most probable state path (“annotations”) to best describe a sequence, according to a pre-determined model (see the “second task of HMMs”, on page 15). Thus we require a good HMM model to make predictions from. Using our *a priori* knowledge about what sequence regions are preceded by what other sequential features etc., it is possible to construct a finite, deterministic state machine to describe this problem. We are also given a set of sequences from which it is possible to directly determine the emission probabilities of the symbols that a state can emit.

However, usually there is only a couple of available options to determine the relations, or transition probabilities, between these states: the use of the Baum-Welch algorithm to find a set of “optimal” probabilities by trying to find the optimal ordering of states which will maximise the series multiplication of emitted symbols’ probabilities, or to manually set them. If there are multiple states in the HMM having the same emission probability distributions (like a loop state and a globular-region state that are both trained with some cytoplasmic sequence), the transition optimisation will be harder by the Baum-Welch algorithm which already does not guarantee finding the best probability set.

With the method I will introduce I try to overcome these difficulties by:

- using nested sampling to search the whole parameter space and also partially to steer Baum-Welch, which reduces the chance of getting stuck in a

local maxima, and,

- following a fully supervised training that utilises known state labels of a given set of training data.

### 5.1.2 Transmembrane topology and signal peptide prediction

Membrane proteins span bilayer lipid phases. Membrane spanning regions of these proteins are usually made up of transmembrane  $\alpha$ -helices or antiparallel  $\beta$ -sheets. Most of the membrane proteins have  $\alpha$ -helices, although there are a number of proteins containing  $\beta$ -barrel structures in the outer membrane regions of bacteria, and in the organelles mitochondria and chloroplasts. Tight bundling of these  $\alpha$ -helical segments forms globular structures in membrane proteins. A typical transmembrane  $\alpha$ -helix contains around 20-25 predominantly hydrophobic amino acid residues. This property forms the basis of computational methods in identifying membrane proteins.

Like transmembrane  $\alpha$ -helices, SPs are also rich in hydrophobic residues. SPs typically range in length between 20 and 30 amino acids in eukaryotes ([Emanuelson \*et al.\*, 1999](#); [von Heijne, 1990](#)), however it is possible to have up to 70aa long SPs (for example, the SP of a protein, P1383, “Ring-infected erythrocyte surface antigen precursor” is 65aa long). They can be divided into three sections in terms of their amino acid content (see Section [3.3.1](#)), with the core hydrophobic region and the cleavage site being more conserved (see Section [5.3.1](#) and Figure [5.5](#) in Results, as an interesting note on how mRNAs of SPs look like). This tri-partite structure is quite useful to predict SPs. Also, their cleavage sites feature a “-3,

-1” rule (von Heijne, 1986), corresponding to the positions occupied by small, conserved amino acids like G or A relative to the actual cleavage position (Figure 3.4a).

Membrane topology describes which regions of the polypeptide chain span the membrane, and which portions lie on either of the watery sides of the lipid bilayer. Membrane topology prediction is important in many ways, as it can help biochemists design drugs or antibodies etc. which are bound to a membrane protein. Many researchers have studied automatic transmembrane topology prediction, and many predictors including TopPred (Claros & von Heijne, 1994), SOSUI (Hirokawa *et al.*, 1998), TMHMM (Krogh *et al.*, 2001) and HMMTOP (Tusnády & Simon, 2001) have been developed in recent years. In 2001 Müller *et al.* showed that all transmembrane prediction methods available at that time had a tendency to interpret hydrophobic parts of signal sequences and transit peptides as membrane-spanning regions. A year after this study, Lao *et al.* evaluated 12 transmembrane topology prediction methods, including the popular ones mentioned above, for their abilities to discriminate between signal peptides and transmembrane regions. These review studies showed that there is still room for improvement in the prediction performance of these programs. While it was shown that TMHMM performed better than the rest of the predictors, in general all the tested programs were badly affected by the presence of a signal peptide in tested sequences. Examples for other TM predictors developed after 2001 are ENSEMBLE (Martelli *et al.*, 2003), Phobius (Käll *et al.*, 2004), PONGO (Amico *et al.*, 2006), PRODIV-TMHMM (Viklund & Elofsson, 2004), and MEMSAT 3 (Jones, 2007).

[Käll \*et al.\* \(2004\)](#) developed a hidden Markov model (HMM) based system called Phobius, which combined the transmembrane protein topology predictor TMHMM ([Krogh \*et al.\*, 2001](#)) and the signal peptide (SP) predictor, SignalP ([Nielsen \*et al.\*, 1997b](#)) (SignalP is discussed in [1.1.1](#)). This combinatorial design of Phobius has been shown ([Käll \*et al.\*, 2007](#)) to improve the performance of TMHMM: By forcing the predictor to choose either of the two sub-models, they increased the discrimination rate between transmembrane regions and N-terminal signal peptides, which resulted in fewer false positives for transmembrane regions.

Unfortunately, the stand-alone version of the Phobius program, although it is downloadable from the program’s prediction service web page, does not come with the “model file” which contains the crucial program parameters. Academic users who want to use this application on their local servers or computers are required to sign a user license agreement. The “terms and conditions” of this license restricts full ownership of even other independent programs that somehow use Phobius or its modifications.<sup>1</sup> In the Lokum localisation prediction system I used TMHMM, because of the mentioned limitations in Phobius, and also because TMHMM is available as a stand-alone application. However, because Phobius has been shown to outperform TMHMM, I chose Phobius to be my sample model as a transmembrane predictor. Thus, the developed prototype predictor is an HMM system whose architecture is similar to that of Phobius.

---

<sup>1</sup>The LICENSOR retains ownership of the SOFTWARE delivered to the LICENSEE. Any modifications or derivative works based on the SOFTWARE are considered part of the SOFTWARE and ownership thereof is retained by the LICENSOR, and are to be made available to him upon request.



## 5.2 Materials and methods

### 5.2.1 Architecture of the HMM

It is no surprise that most of the major transmembrane protein prediction programs use Hidden Markov models (HMMs) to predict protein transmembrane topology. The prototype predictor introduced in this chapter is also based on HMMs (see Section 1.3.1 for a brief description of HMMs).

The architecture of the program introduced here (Figure 5.1) is similar to that of Phobius (Käll *et al.*, 2004). In this model, I used an SP cleavage site motif by directly attaching it into the HMM as a “profile HMM” (Section 1.3.1) where inner states have no self-transitions. This motif was discovered by NestedMICA from a set of secretory protein sequences for the developed localisation prediction program Lokum, and is shown in Figure 3.4.

There are two major possible routes a sequence can be “threaded” into the shown HMM architecture: it could start by traversing through the SP states if this is a more probable option as determined by the dynamic programming part of the algorithm, or it can choose to go directly to the hub state.

In the first route, the SP is modeled as consisting of a three parts: An n-part, a hydrophobic core part (h-part), and a c-part which includes the cleavage site and connects it to the rest of the mature protein region. From here on, it can either go into a short or a long non-cytoplasmic state. Note that, if a sequence has an actual SP, it is not possible for the adjacent part to be in a cytoplasmic region, as the SP will be pointing towards the ER and will drag the rest of the mature part which remains behind it. However, if the N-terminal

were a non-SP transmembrane region, it could penetrate into the membrane from either direction. So the described HMM was designed to reflect this biological phenomenon, by not allowing an SP signal to be followed by a cytoplasmic loop.

The second route that can be followed is to directly go to the hub state of the HMM, from where it is possible to go to either a cytoplasmic or a non-cytoplasmic region (the hub state serves as a symbolic state for better visualising state connections and does not emit any symbols). Non-cytoplasmic loops have been modeled as two states: one for modeling shorter ones, and one for the relatively longer loops. Both loops can be followed by a globular region, although this is not necessary. If a non-cytoplasmic globular state is visited from a non-cytoplasmic state, the system has to go back through the same type of loop, namely the same non-cytoplasmic state (the short or the long one). On the contrary, cytoplasmic loops were not modeled as two separate states, as it has been suggested that their length distributions show less variation (Krogh *et al.*, 2001). Similar to the loops on the other side, cytoplasmic loops can also be preceded by cytoplasmic globular regions.

Emission probabilities for the cytoplasmic and non-cytoplasmic globular states in the HMM have been trained by using cytoplasmic and non-cytoplasmic amino acid sequence chunks of a set of annotated proteins (see “Datasets” below, Section 5.2.2). Similarly, amino acid distributions of SP, transmembrane and loop states were trained from the corresponding sequence chunks. As stated above, the emission probability distributions for the cleavage site positions of SP states were determined directly using the weights of the discovered cleavage site motif, instead of using a single distribution to represent the entire signal. Unlike cytoplasmic

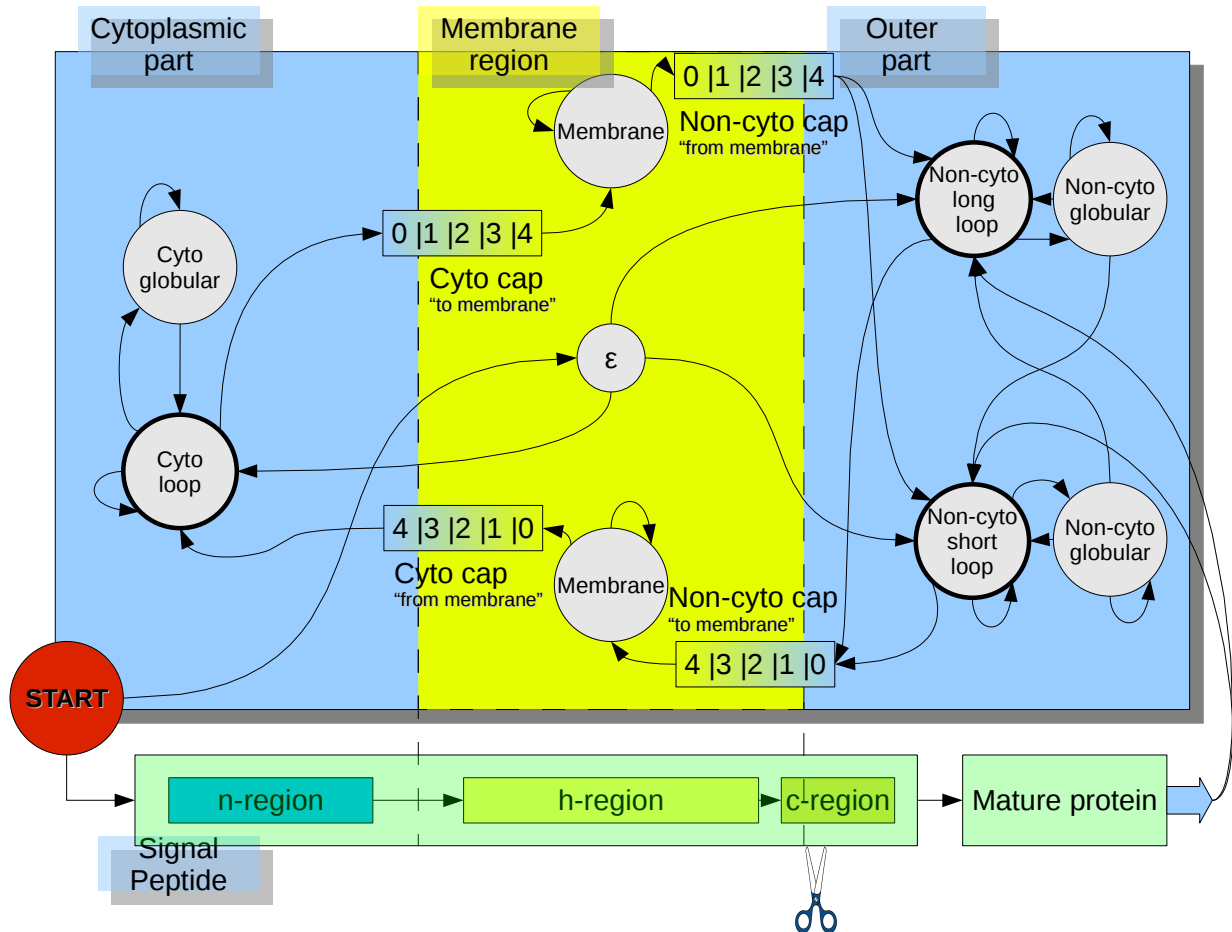


Figure 5.1: **The architecture of the developed transmembrane predictor**  
 The “cyto” HMM states indicate those representing globular or loop protein regions on the cytoplasmic side, while the “non-cyto” ones correspond to regions lying outside of the cell. Globular and loop regions are represented in different states, to better represent these structures by allowing different length distributions and amino acid distribution for each. The model can either follow the path of a Signal Peptide (SP) or the alternative route where it may pass through transmembrane regions.  $\epsilon$  represents the “hub state” which does not emit any symbol but connects certain states. Cap columns are shown in the direction the HMM moves (0 to 4, or 4 to 0), and “to membrane” caps correspond to the positions shown in Tables 5.1 and 5.2, where “0” lies outside of the membrane. The system can terminate while being in any of the states depicted in thicker circles.

regions, for instance, symbol emission distributions at different positions of the cleavage site vary significantly (see Section 3.3.1 for the “-3 -1” rule).

### 5.2.1.1 Representing helix caps in the HMM

Amino acid	POS 0	POS 1	POS 2	POS 3	POS 4
TRP	0.012	0.034	0.040	0.026	0.030
SER	0.059	0.053	0.045	0.049	0.068
ASN	0.056	0.020	0.018	0.018	0.019
ALA	0.049	0.101	0.121	0.101	0.094
HIS	0.029	0.009	0.014	0.015	0.009
TYR	0.034	0.056	0.037	0.039	0.050
PHE	0.038	0.074	0.078	0.082	0.094
ARG	0.148	0.021	0.020	0.020	0.016
LEU	0.053	0.187	0.181	0.163	0.147
PRO	0.035	0.032	0.029	0.037	0.025
MET	0.015	0.052	0.044	0.046	0.040
GLU	0.041	0.011	0.016	0.011	0.011
ILE	0.025	0.098	0.099	0.113	0.114
VAL	0.041	0.104	0.097	0.097	0.104
GLY	0.070	0.059	0.065	0.074	0.077
LYS	0.136	0.016	0.017	0.018	0.014
GLN	0.031	0.010	0.016	0.016	0.019
THR	0.051	0.045	0.044	0.047	0.049
ASP	0.059	0.010	0.010	0.011	0.002
CYS	0.018	0.008	0.010	0.017	0.018

Table 5.1: **Amino acid emission probabilities in the transmembrane helix cytoplasmic side cap.** POS 0 refers to the residue which falls in the cytoplasmic part, while positions 1 to 4 are within the helix. POS0 also corresponds to position-0 of the top and position-4 of the bottom “cyto” caps shown in Figure 5.1.

Transmembrane helices show different amino acid propensities in different positions, too, although these are not so obvious and consistent to be represented as constant motifs. It has been suggested (Jones *et al.*, 1994; Richardson & Richardson, 1988) that middle regions, parts closer to the cytoplasmic loops, and parts

Amino acid	POS 0	POS 1	POS 2	POS 3	POS 4
TRP	0.022	0.039	0.045	0.040	0.038
SER	0.079	0.040	0.059	0.062	0.054
ASN	0.053	0.026	0.024	0.023	0.015
ALA	0.053	0.103	0.099	0.087	0.110
HIS	0.031	0.016	0.015	0.018	0.015
TYR	0.039	0.051	0.048	0.048	0.050
PHE	0.033	0.096	0.090	0.093	0.077
ARG	0.069	0.014	0.011	0.010	0.012
LEU	0.063	0.162	0.156	0.145	0.164
PRO	0.051	0.046	0.041	0.034	0.028
MET	0.025	0.035	0.039	0.043	0.035
GLU	0.073	0.015	0.015	0.011	0.005
ILE	0.028	0.096	0.084	0.100	0.105
VAL	0.043	0.099	0.086	0.106	0.109
GLY	0.082	0.068	0.096	0.083	0.093
LYS	0.062	0.009	0.005	0.008	0.006
GLN	0.047	0.018	0.019	0.015	0.023
THR	0.062	0.046	0.046	0.051	0.041
ASP	0.073	0.017	0.013	0.011	0.008
CYS	0.013	0.005	0.009	0.010	0.011

Table 5.2: **Amino acid emission probabilities in the transmembrane helix non-cytoplasmic side cap.** POS 0 refers to the non-cytoplasmic residue position (outer cell), while positions 1 to 4 are within the helix. POS0 also corresponds to position-4 of the top and position-0 of the bottom “non-cyto” caps shown in Figure 5.1.

near the non-cytoplasmic regions of membrane spanning helices feature significantly different amino acid composition. Consistent with this idea, I calculated amino acid emission probabilities for helix middle regions and helix “cap” regions independently. I extended both helix cap regions to overhang outside of the helix by one residue, where a total of 5 amino acid positions are considered.

If the symbol “i” represents the “inner” cytoplasmic part, “M” the membrane, and “o” the outer, non-cytoplasmic region residues, then one can write the possible two configurations that alpha helices can be in, in terms of these letters, as follows:

1. ...iiiMMMMM...MMMMooo...
2. ...oooMMMMM...MMMMiii...

Residue	KL	Residue	KL
TYR	-0.0064	ASP	-0.01727
MET	-0.0110	ASP	-0.01727
LEU	-0.0134	PRO	-0.01922
ILE	-0.0043	TRP	-0.01052
GLN	-0.0188	CYS	0.00751
PHE	0.0080	SER	-0.02420
THR	-0.0142	HIS	-0.00233
ALA	-0.0056	GLU	-0.03438
GLY	-0.0157	<b>ARG</b>	<b>0.16262</b>
VAL	-0.0024	<b>LYS</b>	<b>0.15494</b>

Table 5.3: **KL deviations of cytoplasmic side helix termini from the noncytoplasmic side helix termini in relative amino acid abundance rates.** Amongst the other amino acids, Arginine (ARG) and Lysine (LYS) differ the most in terms of their relative KL deviations between the two first non-helical positions on both sides (see text). That is, the first non-helix position of a helix cap on the cytoplasmic side is more enriched in ARG and LYS than the first non-helix position on the other side of the membrane.

The so-called “cyto cap” of Figure 5.1 corresponds to the left-hand side of the first helix topology, and the right-hand side of the second. Similarly, the “non-cyto cap” regions overlap where M is either preceded or followed by “o”. Each position in the caps was represented as an independent state as in a profile HMM, and therefore, for each position a separate probability distribution was calculated, rather than using a single, general distribution for the entire cap lengths. This was done after reversing the order of residues in the second type topology. Tables 5.1 and 5.2 summarise the amino acid emission probabilities in the helix caps near the cytoplasmic side, and non cytoplasmic side, respectively. Probability distributions for amino acid position 0 in the tables indicate the first amino acid residue outside of the helix on either side. A good statistical measure to evaluate the difference between two distributions is to use Kullback Leibler (KL) deviation which measures the relative entropy between two distributions:

$$D_{\text{KL}}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (5.1)$$

where P is the “true” distribution from which we measure how another distribution “Q” deviates in terms of relative entropy. KL deviation is not a proper distance measure as it is not symmetric, hence it is called a deviation, although sometimes it may be referred to as “KL distance”. Its value is always positive, but some of the individual terms summed up in the above equation could be negative, due to the nature of the logarithmic part.

Amino acid probability distribution of the cytoplasmic cap at position 0 (Table 5.1) deviates from the non-cytoplasmic cap’s position zero distribution (5.2) by a

KL value of 0.137 (as computed in logarithm base 2). This number does not make much sense alone, but from the individual values contributing to the sum, which are given in Table 5.3, one can easily spot the differences from the relative values computed for the same pair of amino acid symbols. From Table 5.3 it can be easily noticed that the primary difference between the two distributions is due to the differing abundance rates of the Arginine (ARG or R) and Lysine (LYS or K) residues. That is, we can safely conclude that the first amino acid position outside of the helix that points to the cytoplasmic region is more enriched in ARG and LYS than the first non-cytoplasmic position at the other end of the membrane, protruding into the extracellular space. This is also evident by directly comparing the raw probabilities of both distributions.

The other cap positions (1 to 4) obviously differ from the rest of the sequence positions in that they tend to be rich in hydrophobic residues. However, comparing the corresponding positions in the cytoplasmic and non-cytoplasmic cap regions in terms of KL deviation, we see that they are more similar to each other. While position zero distributions of the caps deviate by a KL of 0.137 as mentioned above, the KL deviations (specifically, of the cyto-cap distributions from the non-cyto cap ones) for the other positions are:

1. 0.037
2. 0.044
3. 0.030
4. 0.037



Appendix D lists the individual KL distances between all corresponding pairs of amino acid distributions within the cytoplasmic side, within the non-cytoplasmic side, and finally amongst cytoplasmic and non-cytoplasmic helix cap positions. The computed KL deviations implied that within the same helix cap, there is not much difference among the different cap positions, except for when compared with the ones in the edges. For example, the KL deviation for the distributions at non-cytoplasmic side helix cap positions 3 and 4 is 0.013, which is pretty small. For this reason, it may not be necessary to use separate distributions for such similar two positions; however, this does not introduce any extra complication to the algorithm in optimising their transition probabilities in the HMM, as they have a fixed probability of moving to the adjacent state of 1. Furthermore, any small distribution difference across the same position of different caps can be more valuable than the possible differences within a particular cap's positions in figuring out the correct overall state path. That is why all cap positions were modeled independently, while this does not increase the burden of the state path optimiser.

### 5.2.1.2 Duration HMM states

Even though it will not be biologically possible to have a transmembrane helix of, say length of 3aa, HMMs which are probabilistic methods, can generate predicted state labels corresponding to biologically unfeasible lengths no matter how high the self-transition probabilities of the associated states are. This can be prevented by setting a “minimum number of self-visits” to the problematic states. As mentioned earlier in the Introduction chapter (page 15), such HMMs are referred to

as “duration HMMs”. Unfortunately, the current version of Biojava ([BioJava, 2007](#)), the Java libraries collection which was used to implement this prediction system, does not have such a duration concept. However, I implemented a duration HMM package that allows users to create Markov models having states with a certain number of minimum self-transitions. The probability of staying in a certain “duration” state remains constant while the number of transitions are less than the user-set minimum number, and then goes into an exponential decay in accordance with its classical self-transition probability (see [Figure 1.2](#)).

The states representing the helical regions (see [Figure 5.1](#)) were not allowed to be any shorter than 6 amino acids. This makes the minimum allowed length of a transmembrane alpha helix 14 amino acids, when both cap regions are considered (in Phobius this was taken as 15). Note that, although the cap regions were of length 5aa each, only 4 positions are spanning helices in the designed model. Minimum durations for all the states of the HMM are given in [Table 5.4](#).

### 5.2.2 Datasets and training of the model

For training the emission probabilities of the HMM states, I used the same sequence sets used in Phobius. These sequences are provided in labelled fasta format ([Krogh, 2002](#)) to indicate what type of region (i.e.  $\alpha$ -helix, n-region of signal peptide, loop protruding towards the extracellular space etc.) they fall into. [Table 5.5](#) lists the number of sequences from each category of sequences. As the table shows, the HMM was trained using sequences that include both transmembrane regions and signal peptides, sequences having only one type of these features, and finally sequences not having any of these structures. HMM state

HMM State	Minimum duration
n-region of SPs	6
h-region of SPs	6
c-region of SPs	5 (including cleavage site)
Non-cytoplasmic long loop	15
Non-cytoplasmic short loop	1
Cytoplasmic loop	1
Globular (cytoplasmic)	15
Globular (non-cytoplasmic)	15
Transmembrane alpha helices	6 (14, with both caps)

Table 5.4: **Minimum allowed emission state occupancy numbers for the transmembrane topology predicting HMM.** The n, h, and c regions refer to the sub-regions in N-terminal Signal Peptides (SP). The c-region includes the cleavage site which is modeled as a profile HMM, based on a NestedMICA motif for this region. Transmembrane alpha helices can hardly be shorter than 14, so the corresponding helical states in the HMM allow a minimum of stay of 6 emissions in these states, which is then added with durations of the N- and C-terminal helix caps regions to yield a total length of 14 amino acids.

emission probabilities were determined from the amino acid frequency distributions of only the corresponding sequence segments. That is, the “membrane” states shown in Figure 5.1, for example, have amino acid emission probabilities calculated only from the transmembrane segments of sequences featuring those regions.

Type of sequence	Number of sequences
Transmembrane proteins with signal peptides	45
Transmembrane proteins without signal peptides	247
Non-transmembrane proteins with signal peptides	1773
Non-transmembrane proteins without signal peptides	1520

Table 5.5: **Sequences used in the training of Phobius and of the program developed.**

The HMM was trained using 10-fold cross validation, with the new HMM tran-

sition probability optimisation procedure that is introduced below. The dataset (Table 5.5) was divided into 10 equal portions having more or less the same number of sequences from each type, 9 of which were used in the training while the singled out one was used for testing performance.

### 5.2.3 Transition probability optimisation: a new approach

The optimisation of state transition probabilities was first performed using the standard Biojava (BioJava, 2007) implementation of the Baum-Welch algorithm. This semi-supervised learning method tries to find the best model parameters by maximising the likelihood of the state path, given a set of “emittable” observables and their emission probabilities for each state. It can be considered as semi-supervised, because it tries to optimise the transition probabilities without using the true state labels of a given training data. If the number of parameters to optimise is too large, this method may not produce a good set of parameters at all, particularly for problems where different emission states have similar emission probability distributions. Also, it is often the case during any automatic learning that overfitting of the model can occur, which makes determining the number of iterations in the Baum-Welch algorithm somewhat tricky. Of course, another laborious approach would be to set the model parameters empirically, judging according to what set of parameters maximises performance at the end, where performance would be the number of observables correctly assigned into the associated state label. However, this trial-and-error approach would not be practical for problems involving many emission states.

Here I introduce a new approach to optimise HMM transition probabilities.

It is based on finding a set of state transition probabilities learnt from a given training set with known state labels, by using a probabilistic, generative sampling strategy. “Labels” correspond to sequence annotations such as transmembrane helix, cytoplasmic loop, globular region etc. for each amino acid position. At each iteration of the method, a different set of transition probabilities is tried, and a likelihood score corresponding to the fraction of correctly labelled amino acids in the training dataset is calculated.

This procedure is analogous to other Monte Carlo techniques in that we either accept or reject a proposed set of probabilities, but in order to eliminate the possibility of getting stuck in some local maxima, I use Nested Sampling (see Section 2.2.1 on page 28) that keeps an ensemble of fixed number of proposals, or “probability vectors”. Because we sample typically hundreds of different vectors, this ensures finding a globally optimal solution given enough time and a good likelihood function. In the initialisation step, a likelihood score for each vector in the ensemble is calculated. Transition probabilities of HMM states having a single transition are automatically set to a value of 1.0, and these states are omitted in the rest of the parameter optimisation.

Each step in this algorithm starts with the search for the vector generating the least likelihood score. This “worst” vector is removed from the ensemble, and replaced with a newly sampled one according to the following two rules:

1. The new probability vector is generated by modifying the probabilities of the removed vector.
2. The new vector has to have a likelihood score that is greater than that of

the removed. Another random vector is sampled until this condition is met, or the algorithm is terminated under certain termination criteria.

Once an appropriate move is found the ensemble is updated. This ensures that after each step we move into a better set of solutions, and that the total likelihood increases after each accepted step as illustrated in the cumulative likelihood plot in Figure 5.2. Figure 5.3 shows the likelihood curves of the worst and the replacement states after each move, in an example optimisation problem. The overall likelihood continues to get better monotonically until the system converges or termination occurs as determined by some stopping criteria. Convergence slows down when it becomes harder to find “better” moves than those in the current ensemble (Figure 5.4).

In the sampling process, it is necessary to choose a good, relevant likelihood function that will reflect the fact that we are optimising for the number of correct labels in an HMM state path. To this end, I used a simple likelihood function calculated in the log-space:

$$L(m) = \sum_{i=1}^N \log \left( \frac{1}{|S_i|} \sum_{j=1}^{|S_i|} k_j^i \right) \quad (5.2)$$

where  $|S_i|$  is the length of sequence  $i$ ,  $N$  is the total number of sequences, and  $k_j^i$  is a unit function that is non-zero if the  $j^{th}$  amino acid position of the  $i^{th}$  sequence is correctly identified by the evaluated Markov model  $m$ . The total likelihood function is the sum of accepted model likelihoods that constitute the ensemble. It is this cumulative likelihood function that is ensured to increase at each step before accepting a proposed Monte Carlo move. Similarly, the least

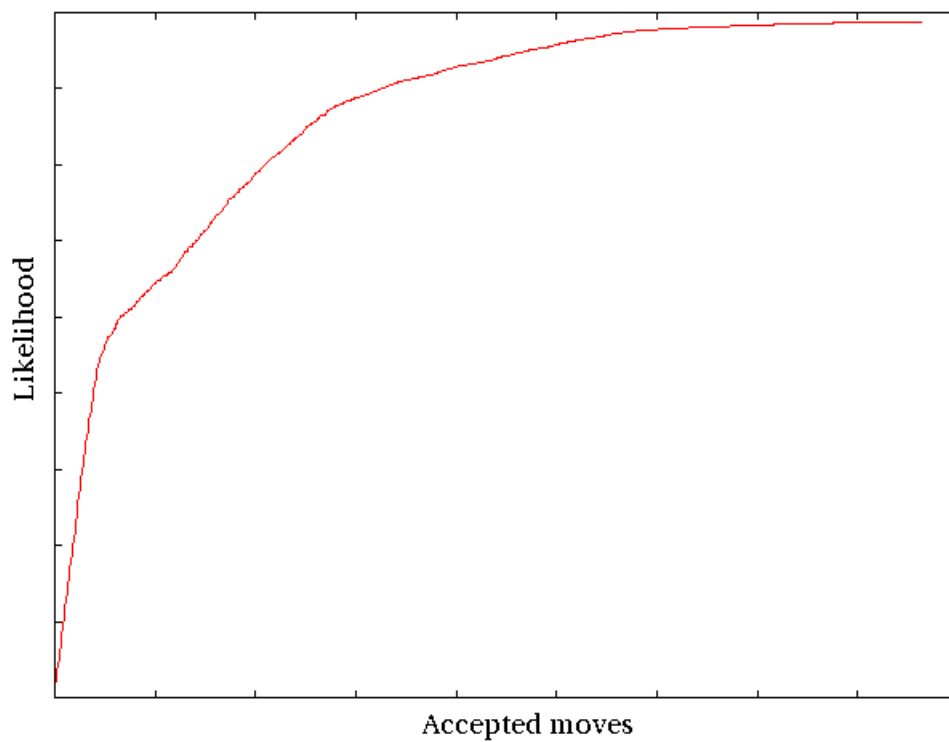


Figure 5.2: **Total likelihood function monotonically increases in nested sampling.** This plot illustrates how the likelihood function (y-axis) used in the developed HMM transition probabilities optimisation technique varies over accepted steps (x-axis).

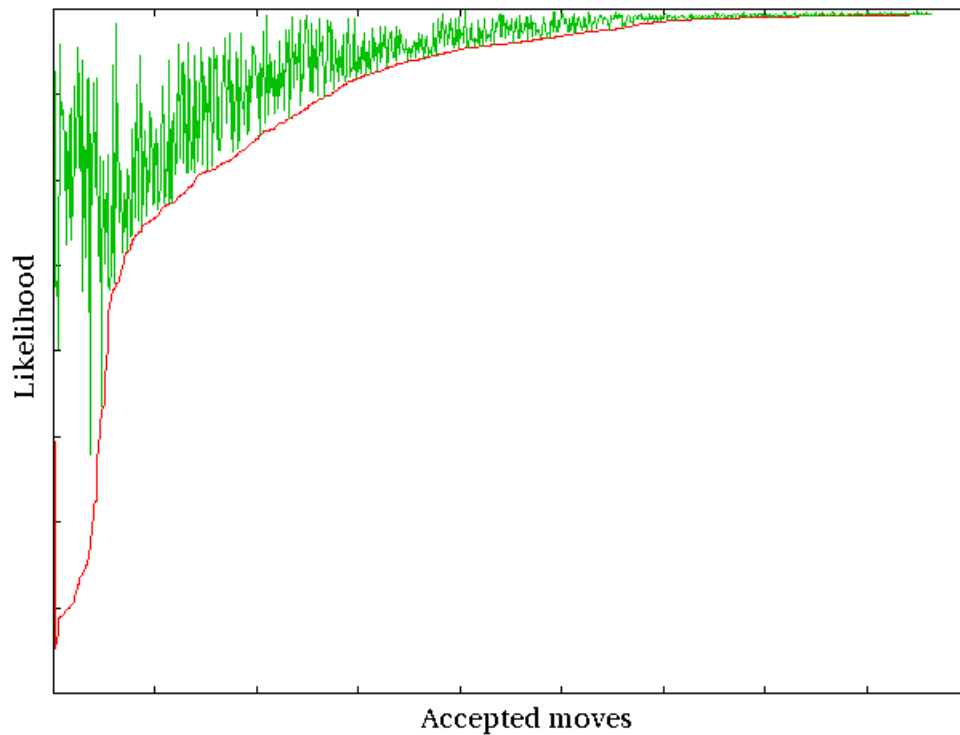


Figure 5.3: **Transition probability set having the least likelihood is replaced by new better one.** A suggested move has to have a likelihood that is larger than the “worst” vector’s likelihood to be accepted. The red line represents the least likely probability vector of the ensemble at a particular step. It is replaced by a “better” vector, shown in green colour.





Figure 5.4: **It becomes more difficult over time to find “acceptable” states.** The curve shows the ratio of accepted moves to the total number of proposed moves, which is updated after each accepted move. This accepted/(accepted+rejected) ratio helps to determine whether an optimisation run is close to convergence or not.

likely state's score  $L_w$  at each step will increase in parallel with the improving overall goodness of the ensemble.

Increasing the ensemble size may result in better parameter sets. However, the number of sampled vectors in the ensemble hugely affects the program speed and convergence rate. At the end, I observed that setting the ensemble size to around 10 times the number of HMM states being optimised in a particular problem proved to be sufficient for typical HMM transition probability optimisation problems.

The individual probabilities forming the vectors are sampled according to Gaussian functions associated with each of the probabilities. The sampling is carried out using a Gaussian with a mean that is equal to the previous probability value. The main sampled entity is, of course, probability distributions of HMM states. This is achieved through multiple ways:

- All state probabilities at the same time, using a large Gaussian variance
- All state probabilities at the same time, using a small Gaussian variance
- A randomly picked state's probabilities, using a large Gaussian variance
- A randomly picked state's probabilities, using a small Gaussian variance
- All state probabilities at the same time, using a Gaussian variance dynamically updated according to the number of rejected moves
- A randomly picked state's probabilities, using a Gaussian variance dynamically updated according to the number of rejected moves

- All state probabilities at the same time, using a random Gaussian variance
- A Baum-Welch iteration of a model created based on the probabilities of a randomly picked state

That is, the distributions are perturbed by using uniform Gaussian distributions with either pre-determined or dynamically updated standard deviations (see below for how this is achieved). The mean of each Gaussian is equalised to the previous selected probability value of a particular state transition. Two standard deviations, one small and one large, are used. The one that will be utilised in a particular step is determined by a random selection process. These variance values can be later dynamically updated based on the number of accepted and rejected proposals. This reduces the number of rejected proposals, and also allows the system to explore different sets of solutions as much as possible. As the above list shows, sampling could be applied on all states of an HMM simultaneously, or by working on a single distribution in each proposed step. Another possibility is to select a state randomly and then change its transition weights according to a randomly picked variance value around the previous values of the probabilities.

Finally, a Baum-Welch move can be proposed, based on a certain probability (for instance, 15% of the proposed steps are based on moves proposed by the Baum Welch algorithm in the current implementation). In a Baum-Welch move, the actual Markov model is updated with the transition weights of a randomly chosen item from the ensemble. A certain number of Baum-Welch iterations are run on the model characterised by the selected transition weights set, as in a classical transition probability optimisation procedure. If this new model

increases our likelihood function that measures the number of correctly assigned labels, then this move is accepted, and the ensemble is updated with the transition set that the Baum-Welch algorithm fine-tuned.

In the case of rejected Baum-Welch proposals for the same worst state, depending on the number of recent rejections, the iteration number of the Baum-Welch training is increased. This iteration number is randomly chosen from numbers up to a maximum value equal to the number of local rejections, provided that it does not exceed an empirically set 20 iterations per move.

Whenever needed, the standard deviations that are allowed to change are dynamically altered, as stated above (some remain fixed during the entire optimisation). This is performed by multiplying or dividing the previous standard deviation value by  $e^{1.0/rejected}$ , making sure it will be in the interval (0,1). This way, when there is room for large gains in the total likelihood, this is achieved faster by using a larger standard deviation, and when the system begins to reject more and more proposals, the probability distributions are less perturbed when sampling from the ensemble.

Individual transition probabilities for each state are sampled from their Gaussians by using the “online” standard deviation divided by the number of total transitions a state possesses. If a negative value is obtained from a particular sampling of a state, all other transition probabilities in that state are shifted to the positive side by an amount equal to the minimum probability obtained (plus some very small number, to avoid absolute zero probabilities). The values sampled from the Gaussians in each step are then re-normalised to obtain sensible probabilities that will sum up to 1 for each state.

## 5.3 Results

Tables 5.6 and 5.7 compare transmembrane (TM) predictors Phobius and the developed prototype program, in terms of their performance in predicting TM topology for i) proteins having both TM and signal peptides (SP), ii) proteins having only TM helices, iii) proteins having only SPs, and finally, iv) proteins having neither a TM nor an SP. For a prediction to be counted as correct, all annotated individual TM helices and loop regions must be predicted correctly. An overlap of at least 5 residues was considered a “correct” prediction for each helix, as done in the CASP competitions (Cozzetto *et al.*, 2005, 2007; Soro & Tramontano, 2005; Valencia, 2005), or as in other studies reporting TM accuracies (Jones, 2007; Käll *et al.*, 2004). In the presence of a signal peptide (SP), whether a program predicted the SP or not in a particular protein was not taken into account in determining the overall correct TM topology.

SP prediction performance is evaluated separately, and the results for predictors that are capable of detecting SPs are summarised in Table 5.7. The developed predictor was compared with two versions of SignalP, in addition to Phobius which is both a TM and SP predictor. TMHMM is not designed to predict SPs directly.

The results generally suggest that the developed HMM program is better in predicting SPs than the other compared programs, although its architecture is quite similar to that of Phobius. On the other hand, the prototype program performs relatively badly in correct transmembrane (TM) topology prediction, although it performs reasonably well in predicting individual TM helices.

	This predictor	Phobius	TMHMM2.0
TM and SP proteins			
Correct topology	66.7%	91.1%	71.1%
Correct TMs	76.1%		
Correct SPs	88.9%		
False positive TMs	18.1%		
TM-only proteins			
Correct topology	36.0%	63.6%	65.2%
Correct TMs	76.1%		
False positives	3.6%	7.7%	
Non-SP and non-TM proteins			
Correct topology	99.87%	98.2%	98.7%

Table 5.6: **Prediction performance summary for “TM-and-SP”, “TM-only” and “non-SP, non-TM” proteins.** A prediction was taken as correct when all the predicted Transmembrane (TM) helices overlap all the annotated TM helices of the protein over at least 5 amino acids, and when the loops were correct. In the evaluation of predictions for proteins having no TM helices, the reported “correct topology” corresponds to not having any predicted TM helices for that protein. Incorrect signal peptides (SPs) were not considered in determining correct topology prediction rates. The available prediction rates for Phobius and TMHMM2.0 were taken as reported in Käll *et al.* (2004), where TMHMM results were not cross validated.

	This predictor	Phobius	SignalP-NN	SignalP-HMM
Correct SPs	89.8%	96.5%	97.7%	98.6%

Table 5.7: **Correct prediction rates for SP-only proteins.** Correct SP prediction rates are shown for the program developed, Phobius, Neural network version of SignalP (Nielsen *et al.*, 1997a) and the HMM version of SignalP (Nielsen & Krogh, 1998). The available prediction rates for Phobius and the two version of SignalP were taken as reported in Käll *et al.* (2004), where SignalP results were not cross validated.

Using a duration-enabled HMM which was trained by the new optimisation procedure improved the results dramatically. When no minimum-durations were set in the HMM model, the correctly predicted TM helices rate, for example, decreased to 12.5% from 76.1% for the TM and SP containing protein dataset, while it decreased to 19.6% from the same initial value of 71.6% for the TM-only proteins. On the other hand, optimising transition probabilities by Baum-Welch without using any initial “clever guesses” resulted in very poor TM helix prediction accuracies (<1%), after letting it to run for about a dozen, 50 and finally a few hundreds of cycles, even when it was trained using the entire protein set.

### 5.3.1 Signal peptides at the DNA level

As mentioned in Section 3.1.1, most of the signal peptides are at least 20 aa long. Inspecting the available annotated protein sequences having signal peptides, we observed that this signal could stretch out to as long as 50 aa. One question that arises for such long N-terminal sequence patterns is how their untranslated form might look at the DNA level, given that their corresponding mRNAs would be three times longer than the amino acid signal. Because NestedMICA is a DNA discovery tool at the same time, using the motif finder in the DNA mode, I had a chance to investigate how conserved the hydrophobic regions are at the genomic level. Interestingly, as Figure 5.5 shows, only certain residues in the codons that translate into this signal are conserved. The hydrophobic part of the signal is generated from codons having a conserved second position which is usually a “T”, while the cleavage site, having small residues like Alanine (A), Glycine (G)

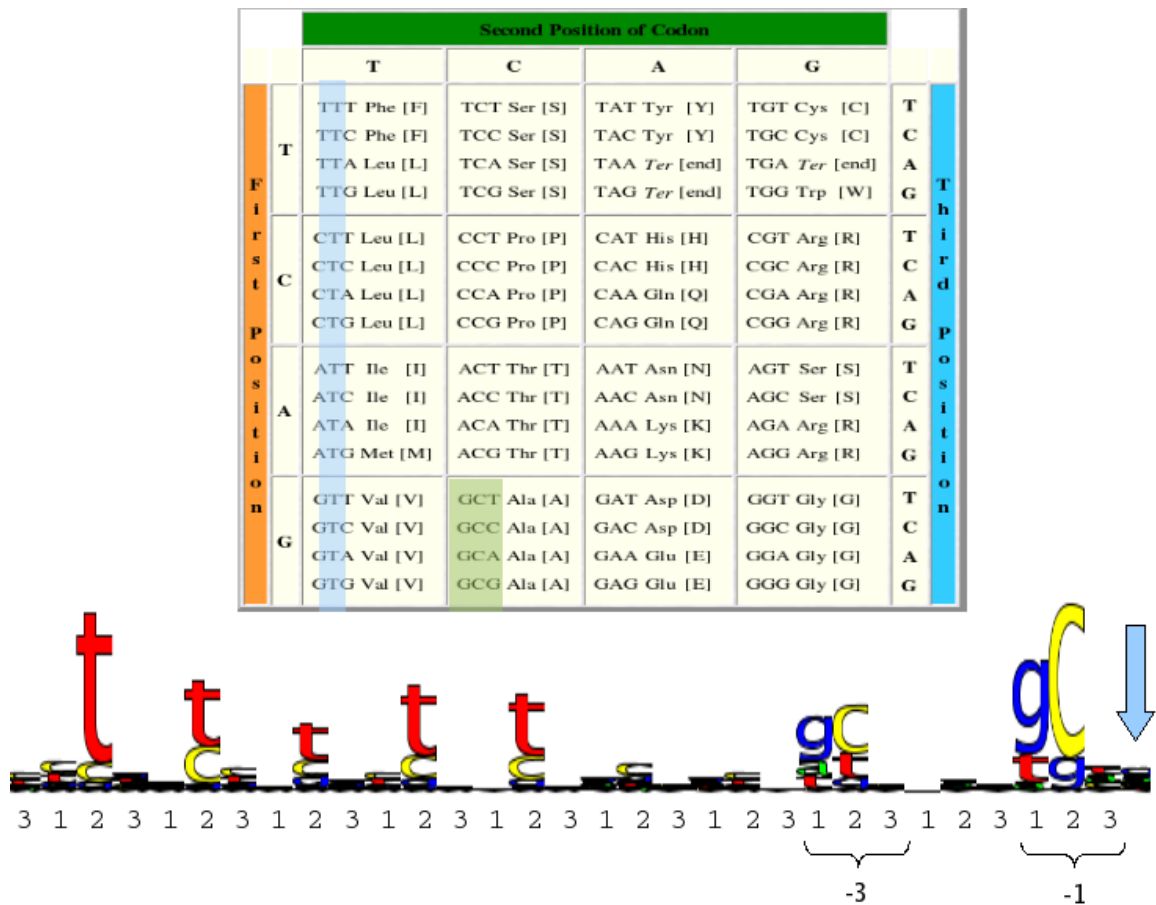


Figure 5.5: **Signal peptide motif at the RNA level.** Conservation of only certain positions in the codons corresponding to the SP signal suffices to generate hydrophobic amino acid chains, as the genetic code table above illustrates having a 'T' at the middle of a codon usually generates hydrophobic amino acid residues. The sequence logo shows 10 residues before the actual cleavage residue position which is indicated by an arrow. Codons beginning with the "GC" dinucleotides correspond to the small amino acids of the "-3 and -1 positions" rule (see text), with respect to the cleave site. Numbers right below the sequence logo correspond to the actual positions of amino acids within the codons in the associated correct reading frame.

etc in the protein level, is dominated by Guanine and Cytosine at positions 3 and 1, respectively, at the DNA level.



## 5.4 Discussions

In this chapter, I introduced a new, fully supervised methodology for training HMM transition probabilities and applied it on TM topology prediction as an example. While this method is open to further improvements and also to some more testing, the essential idea behind it, the use of Monte Carlo strategies to fine tune transitions, seems to be a promising approach. These kind of optimisation problems normally involve huge parameter landscapes where each optimised parameter can take any probability value. Another hurdle in this type of search heuristics is that, in principle it is not uncommon for a sampled property to move towards some local maxima and get stuck there. However, with the use of nested sampling, a fruitful strategy that has proven itself in biological sequence motif discovery before, such possibilities can be avoided. With this approach multiple possible solutions representing different maxima from the entire probability landscape are considered at the same time, instead of trying to make a single sample better during the entire process. This is simply to eliminate the greediness of sampling at each step that could possibly miss the real solution set at the end, had it not moved to the locally “best” condition in previous steps. Removal of the “weakest” state having the least likelihood probability from a large population, and re-sampling from the “fitter” entities to replace the worst, is conceptually nothing but a genetic algorithm way of optimising, with the exception that the space is continuous here – entities are not simply of type that either exist or not.

Transmembrane topology prediction is a well established field for many years now, and there are many good TM topology prediction programs. However, due

to the similarities between N-terminal TM helices and SPs (see Sections 3.1.1 and 3.3.1), they tend to misclassify SPs as TMs and vice versa. Programs such as Phobius have recently reduced this by using a combinatorial approach where SPs and TMs are predicted in the same model. This resulted in a reduced number of false positive predictions and cross-misclassifications (Käll *et al.*, 2007).

As mentioned in the methods, this prototype program differs from Phobius in that in the HMM I directly used a cleavage site motif that was discovered by NestedMICA, and I use a new optimal transition probability estimation method. A relatively low correct topology prediction with respect to the correctly identified TM helices indicates that the program tends to invert the orientations of the topologies it predicts. Apart from the difficulties inherent in the biology of the problem, this could possibly be due to immature termination of the transition probability optimisation procedure, which has not been fully optimised for termination criteria yet.

Both the parameter optimisation approach and the prototype TM topology predictor that was developed to demonstrate that this approach can actually be used successfully are promising. The optimisation method is open to further development, which, in turn, can significantly improve the TM predictor's performance.

# Chapter 6

## Conclusions

Motif discovery is an important step in protein functional annotation as it can help to identify different protein properties in curation of protein annotation. I adapted and extended NestedMICA for finding short protein signals, and compared its performance with the MEME tool. NestedMICA was tested on synthetic and biologically-authentic datasets produced by spiking instances of known motifs into a some random protein sequences. NestedMICA was also assessed at various conditions including using different input sequence lengths, target motif length, target motif number, and finally different motif abundance rates.

Generally NestedMICA recovered most of the short (3-9 amino acid long) test protein motifs spiked into a test set of sequences at different frequencies. All assessments experiments I performed showed that NestedMICA's motif discovery performance was better than MEME in terms of the number of correctly recovered motifs, although generally there was no significant difference in terms of the quality of recovered motifs by both of the compared programs. NestedMICA performed clearly well even in the discovery of relatively short motifs that exist in only a small fraction of sequences.

---

Protein subcellular localisation identification is another concrete key step in functional annotation. Most of the biologically inspired *ab initio* methods that have been developed to tackle this problem had either a limited number of localisation categories, or low prediction accuracies, particularly for eukaryotic sequences. Similarity-based prediction methods could be more reliable than *ab initio* predictors for sequences having annotated highly homologous counterparts in databases. However, predicting localisation for unseen, different proteins becomes a more challenging task for this type of prediction program. Furthermore, signal-based *ab initio* prediction efforts can give us more insight and clues about the underlying biology in protein targeting.

I developed a novel computational *ab initio* classification tool, Lokum, for protein subcellular localisation prediction, covering 9 major localisation classes for animal, 9 for fungal and 10 for plant sequences. It uses targeting and retention signal motifs reported by the probabilistic motif discovery tool NestedMICA, and other protein features including transmembrane topologies and amino acid composition. Lokum does not use sequence similarity, or any other *a priori* knowledge such as known nuclear localisation signals by searching databases. Additionally, we propose a multi-component, probabilistic model tolerating positional shifts for the bipartite nuclear localisation signals (NLS). To find the bipartite NLS, we added protein support to Eponine, a tool originally written for mammalian transcription start site modeling. We also show that using the N-linked glycosylation motif, which was amongst the motifs detected by NestedMICA, can contribute to localisation prediction.

Combining all these features in a Support Vector Machine (SVM), we get an

---

average correct prediction rate of more than 80% for nine animal, nine fungal and ten plant protein localisation classes in 5-fold cross-validated tests performed on an eukaryotic dataset. Finally, a web service has been implemented for public use.

In Chapter 3, I showed that including reported statistics from transmembrane prediction programs can increase prediction accuracy in automatic *ab initio* classification of protein subcellular localisation. A large number of transmembrane proteins follow the secretory pathway and end up in localisations such as ER, Golgi, plasma membrane or extracellular space. Plasma membrane proteins have a larger number of membrane-spanning regions than the other classes of proteins, as shown in the same chapter. Therefore, it is actually not surprising that transmembrane topology prediction can improve localisation.

Motifs reported by NestedMICA and Eponine have been more useful than any other component in the prediction system. In addition to the reported motifs that I could associate with known localisation signals, a couple of three-letter PWMs were discovered from a set of plasma membrane sequences, which turned out to be the two variants of the N-linked glycosylation site motifs. Some of the discovered motifs, such as these glycosylation motifs that are known not to be directly involved in localisation, also increased the prediction performance, because of their differing abundance rates in different types of proteins.

In Chapter 4, I showed that it is reasonably possible to predict more specific, sub-compartmental localisation categories, by showing that proteins that spend at least some time in nucleoli can be distinguished reasonably well from the remaining nuclear proteins. In addition to the features used in Lokum, I used

---

protein disorder region predictions. As summarised in Section 3.3.7, using disorder prediction did not contribute significantly to the prediction of the general localisation categories. But I demonstrated that disorder prediction can be a useful feature in discriminating between proteins targeted into different sub-nuclear compartments. In fact, sub-dividing the main localisation categories to further fine tune localisation prediction can be said to overlap with the field of *ab initio* protein function identification, where disorder prediction has been shown to work (Dunker *et al.*, 2000; Lobley *et al.*, 2007; Wright & Dyson, 1999). Interestingly, the results obtained in Chapter 4 suggested that a larger number of nuclear localisation signals exist in the disordered regions of nucleolar proteins as compared to the disordered regions of other proteins in the nucleus. It should be possible to further exploit this phenomenon in the prediction of proteins localised in other sub-compartments.

An interesting observation we can make from Chapters 3, 4, and 5 is that there is a general tendency in protein amino acid composition to contain Lysine (K) and Arginine (R) residues at larger proportions as we move from the extracellular space towards the cytoplasm, and finally into the nucleus and other subnuclear compartments. If we consider the amino acid contents of extracellular, cytoplasmic and nuclear proteins, amino acids K (Figure B.12) and R (Figure B.2) are least abundant in extracellular, followed by cytoplasmic and then nuclear proteins, in this order. In Section 5.2.1.1 of Chapter 5, we saw that membrane spanning regions towards the cytoplasm become richer in K and R content compared to their parts on the opposite side, towards the extracellular space (Tables 5.1, 5.2 and 5.3). Finally, in Chapter 4, where I analysed the differences between

---

nuclear and nucleolar protein sequences, it turned out that nuclear proteins, which are confined in the sub-nuclear compartment of nucleolus, tend to contain a larger number of K and R amino acid residues (Figure 4.7).

Finally, as demonstrated by its application on transmembrane topology prediction, the introduced alternative transition probability optimisation method that I developed (Chapter 5) is a promising approach for use in any prediction program that utilises HMMs, including the classical problems of gene finding, secondary structure prediction, and so on.

# References

- AHMAD, S. & SARAI, A. (2005). PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinformatics*, **6**, 33. [26](#)
- ALBER, T., GILBERT, W.A., PONZI, D.R. & PETSKO, G.A. (1983). The role of mobility in the substrate binding and catalytic machinery of enzymes. *Ciba Found Symp*, **93**, 4–24. [111](#)
- AMICO, M., FINELLI, M., ROSSI, I., ZAULI, A., ELOFSSON, A., VIKLUND, H., VON HEIJNE, G., JONES, D., KROGH, A., FARISELLI, P., MARTELLI, P.L. & CASADIO, R. (2006). PONGO: a web server for multiple predictions of all-alpha transmembrane proteins. *Nucleic Acids Res*, **34**, W169–W172. [137](#)
- ANDERSEN, J.S., LYON, C.E., FOX, A.H., LEUNG, A.K.L., LAM, Y.W., STEEN, H., MANN, M. & LAMOND, A.I. (2002). Directed proteomic analysis of the human nucleolus. *Curr Biol*, **12**, 1–11. [113](#)
- ANDERSEN, J.S., LAM, Y.W., LEUNG, A.K.L., ONG, S.E., LYON, C.E., LAMOND, A.I. & MANN, M. (2005). Nucleolar proteome dynamics. *Nature*, **433**, 77–83. [113](#)



## REFERENCES

---

- BAILEY, T.L. & ELKAN, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol*, **2**, 28–36. [3](#)
- BAILEY, T.L. & ELKAN, C. (1995). The value of prior knowledge in discovering motifs with meme. *Proc Int Conf Intell Syst Mol Biol*, **3**, 21–29. [18](#), [27](#), [64](#)
- BAIROCH, A. & APWEILER, R. (1996). The SWISS-PROT protein sequence data bank and its new supplement TREMBL. *Nucleic Acids Res*, **24**, 21–25. [12](#), [40](#)
- BAIROCH, A. & APWEILER, R. (2000). The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res*, **28**, 45–48. [12](#)
- BANNAI, H., TAMADA, Y., MARUYAMA, O., NAKAI, K. & MIYANO, S. (2002). Extensive feature detection of N-terminal protein sorting signals. *Bioinformatics*, **18**, 298–305. [7](#), [9](#)
- BATEMAN, A., COIN, L., DURBIN, R., FINN, R.D., HOLLICH, V., GRIFFITHS-JONES, S., KHANNA, A., MARSHALL, M., MOXON, S., SONNHAMMER, E.L.L., STUDHOLME, D.J., YEATS, C. & EDDY, S.R. (2004). The Pfam protein families database. *Nucleic Acids Res*, **32**, D138–D141. [11](#), [25](#), [58](#)
- BENDTSEN, J.D., JENSEN, L.J., BLOM, N., HEIJNE, G.V. & BRUNAK, S. (2004a). Feature-based prediction of non-classical and leaderless protein secretion. *Protein Eng Des Sel*, **17**, 349–356. [65](#)

## REFERENCES

---

- BENDTSEN, J.D., NIELSEN, H., VON HEIJNE, G. & BRUNAK, S. (2004b). Improved prediction of signal peptides: SignalP 3.0. *J Mol Biol*, **340**, 783–795. [7](#), [8](#)
- BHASIN, M. & RAGHAVA, G.P.S. (2004). ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST. *Nucleic Acids Res*, **32**, W414–W419. [7](#), [10](#)
- BHASIN, M., GARG, A. & RAGHAVA, G.P.S. (2005). PSLpred: prediction of subcellular localization of bacterial proteins. *Bioinformatics*, **21**, 2522–2524. [10](#)
- BIOJAVA (2007). <http://www.biojava.org>. [27](#), [75](#), [148](#), [150](#)
- BRAMEIER, M., KRINGS, A. & MACCALLUM, R.M. (2007). NucPred—predicting nuclear localization of proteins. *Bioinformatics*, **23**, 1159–1160. [110](#)
- BURGARD, A.P., MOORE, G.L. & MARANAS, C.D. (2001). Review of the TEIRESIAS-based tools of the IBM Bioinformatics and Pattern Discovery Group. *Metab Eng*, **3**, 285–288. [26](#)
- BURGES, C.J.C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, **2**, 121–167. [23](#), [24](#)
- BURLEY, S.K. & PETSKO, G.A. (1985). Aromatic-aromatic interaction: a mechanism of protein structure stabilization. *Science*, **229**, 23–28. [132](#)
- CARNINCI, P., KASUKAWA, T., KATAYAMA, S., GOUGH, J., FRITH, M.C., MAEDA, N., OYAMA, R., RAVASI, T., LENHARD, B., WELLS, C., KODZ-

## REFERENCES

---

IUS, R., SHIMOKAWA, K., BAJIC, V.B., BRENNER, S.E., BATALOV, S., FORREST, A.R.R., ZAVOLAN, M., DAVIS, M.J., WILMING, L.G., AIDINIS, V., ALLEN, J.E., AMBESI-IMPIOMBATO, A., APWEILER, R., ATURALIYA, R.N., BAILEY, T.L., BANSAL, M., BAXTER, L., BEISEL, K.W., BERSANO, T., BONO, H., CHALK, A.M., CHIU, K.P., CHOUDHARY, V., CHRISTOFFELS, A., CLUTTERBUCK, D.R., CROWE, M.L., DALLA, E., DALRYMPLE, B.P., DE BONO, B., GATTA, G.D., DI BERNARDO, D., DOWN, T., ENGSTROM, P., FAGIOLINI, M., FAULKNER, G., FLETCHER, C.F., FUKUSHIMA, T., FURUNO, M., FUTAKI, S., GARIBOLDI, M., GEORGIIHEMMING, P., GINGERAS, T.R., GOJOBORI, T., GREEN, R.E., GUSTINCICH, S., HARBERS, M., HAYASHI, Y., HENSCH, T.K., HIROKAWA, N., HILL, D., HUMINIECKI, L., IACONO, M., IKEO, K., IWAMA, A., ISHIKAWA, T., JAKT, M., KANAPIN, A., KATOH, M., KAWASAWA, Y., KELSO, J., KITAMURA, H., KITANO, H., KOLLIAS, G., KRISHNAN, S.P.T., KRUGER, A., KUMMERFELD, S.K., KUROCHKIN, I.V., LAREAU, L.F., LAZAREVIC, D., LIPOVICH, L., LIU, J., LIUNI, S., MCWILLIAM, S., BABU, M.M., MADERA, M., MARCHIONNI, L., MATSUDA, H., MATSUZAWA, S., MIKI, H., MIGNONE, F., MIYAKE, S., MORRIS, K., MOTTAGUI-TABAR, S., MULDER, N., NAKANO, N., NAKAUCHI, H., NG, P., NILSSON, R., NISHIGUCHI, S., NISHIKAWA, S., NORI, F., OHARA, O., OKAZAKI, Y., ORLANDO, V., PANG, K.C., PAVAN, W.J., PAVESI, G., PESOLE, G., PETROVSKY, N., PIAZZA, S., REED, J., REID, J.F., RING, B.Z., RINGWALD, M., ROST, B., RUAN, Y., SALZBERG, S.L., SANDELIN, A., SCHNEIDER, C., SCHNBACH, C., SEKIGUCHI, K., SEMPLE, C.A.M., SENO, S., SESSA, L., SHENG, Y.,

## REFERENCES

---

- SHIBATA, Y., SHIMADA, H., SHIMADA, K., SILVA, D., SINCLAIR, B., SPERLING, S., STUPKA, E., SUGIURA, K., SULTANA, R., TAKENAKA, Y., TAKI, K., TAMMOJA, K., TAN, S.L., TANG, S., TAYLOR, M.S., TEGNER, J., TEICHMANN, S.A., UEDA, H.R., VAN NIMWEGEN, E., VERARDO, R., WEI, C.L., YAGI, K., YAMANISHI, H., ZABAROVSKY, E., ZHU, S., ZIMMER, A., HIDE, W., BULT, C., GRIMMOND, S.M., TEASDALE, R.D., LIU, E.T., BRUSIC, V., QUACKENBUSH, J., WAHLESTEDT, C., MATTICK, J.S., HUME, D.A., KAI, C., SASAKI, D., TOMARU, Y., FUKUDA, S., KANAMORI-KATAYAMA, M., SUZUKI, M., AOKI, J., ARAKAWA, T., IIDA, J., IMAMURA, K., ITOH, M., KATO, T., KAWAJI, H., KAWAGASHIRA, N., KAWASHIMA, T., KOJIMA, M., KONDO, S., KONNO, H., NAKANO, K., NINOMIYA, N., NISHIO, T., OKADA, M., PLESSY, C., SHIBATA, K., SHIRAKI, T., SUZUKI, S., TAGAMI, M., WAKI, K., WATAHIKI, A., OKAMURA-OHO, Y., SUZUKI, H., KAWAI, J., HAYASHIZAKI, Y., CONSORTIUM, F.A.N.T.O.M., GROUP, R.I.K.E.N.G.E.R. & GROUP), G.S.G.G.N.P.C. (2005). The transcriptional landscape of the mammalian genome. *Science*, **309**, 1559–1563. [114](#)
- CHANG, C.C. & LIN, C.J. (2001). LIBSVM: a library for support vector machines. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. [23](#), [24](#), [79](#), [108](#), [118](#)
- CHOTHIA, C. & LESK, A.M. (1986). The relation between the divergence of sequence and structure in proteins. *EMBO J*, **5**, 823–826. [13](#)
- CLAROS, M.G. & VON HEIJNE, G. (1994). TopPred II: an improved software for membrane protein structure predictions. *Comput Appl Biosci*, **10**, 685–686.

68, 137

- COEYTAUX, K. & POUPON, A. (2005). Prediction of unfolded segments in a protein sequence based on amino acid composition. *Bioinformatics*, **21**, 1891–1900. [112](#)
- COKOL, M., NAIR, R. & ROST, B. (2000). Finding nuclear localization signals. *EMBO Rep*, **1**, 411–415. [7](#), [8](#), [110](#)
- COMON, P. (1994). Independent component analysis, a new concept? *Signal Process.*, **36**, 287–314. [19](#)
- COZZETTO, D., MATTEO, A.D. & TRAMONTANO, A. (2005). Ten years of predictions ... and counting. *FEBS J*, **272**, 881–882. [112](#), [159](#)
- COZZETTO, D., GIORGETTI, A., RAIMONDO, D. & TRAMONTANO, A. (2007). The evaluation of protein structure prediction results. *Mol Biotechnol.* [112](#), [159](#)
- DANG, C.V. & LEE, W.M. (1989). Nuclear and nucleolar targeting sequences of c-erb-a, c-myb, N-myc, p53, HSP70, and HIV tat proteins. *J Biol Chem*, **264**, 18019–18023. [108](#), [120](#)
- DEMPSTER, A., LAIRD, N. & RUBIN, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J Royal Statistical Society, Series B*, **39**. [18](#)
- DINGWALL, C. & LASKEY, R.A. (1991). Nuclear targeting sequences— a consensus? *Trends Biochem Sci*, **16**, 478–481. [73](#)

## REFERENCES

---

- DOGRUEL, M., DOWN, T. & HUBBARD, T. (2008). NestedMICA as an ab initio protein motif discovery tool. *BMC Bioinformatics*, **9**, 19. [3](#), [25](#), [64](#), [69](#), [115](#)
- DOWN, T.A. & HUBBARD, T.J.P. (2002). Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res*, **12**, 458–461. [3](#), [27](#), [64](#), [73](#), [74](#), [76](#)
- DOWN, T.A. & HUBBARD, T.J.P. (2004). What can we learn from noncoding regions of similarity between genomes? *BMC Bioinformatics*, **5**, 131. [74](#)
- DOWN, T.A. & HUBBARD, T.J.P. (2005). NestedMICA: sensitive inference of over-represented motifs in nucleic acid sequence. *Nucleic Acids Res*, **33**, 1445–1453. [2](#), [27](#), [30](#), [32](#), [33](#), [34](#), [36](#), [64](#)
- DUNKER, A.K., OBRADOVIC, Z., ROMERO, P., GARNER, E.C. & BROWN, C.J. (2000). Intrinsic protein disorder in complete genomes. *Genome Inform Ser Workshop Genome Inform*, **11**, 161–171. [111](#), [168](#)
- DURBIN, R., EDDY, S.R., KROGH, A. & MITCHISON, G. (1999). *Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press. [16](#), [18](#), [30](#)
- ECK, R.V. & DAYHOFF, M.O. (1966). *Atlas of Protein Sequence and Structure*, vol. 3. National Biomedical Research Foundation, Silver Spring, Maryland. [68](#)
- EDWARDS, R.J., DAVEY, N.E. & SHIELDS, D.C. (2007). SLiMFinder: a probabilistic method for identifying over-represented, convergently evolved, short linear motifs in proteins. *PLoS ONE*, **2**, e967. [26](#)

## REFERENCES

---

- EMANUELSSON, O., NIELSEN, H. & VON HEIJNE, G. (1999). ChloroP, a neural network-based method for predicting chloroplast transit peptides and their cleavage sites. *Protein Sci*, **8**, 978–984. [7](#), [8](#), [66](#), [85](#), [136](#)
- EMANUELSSON, O., NIELSEN, H., BRUNAK, S. & VON HEIJNE, G. (2000). Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. *J Mol Biol*, **300**, 1005–1016. [7](#), [40](#), [56](#), [59](#)
- EMANUELSSON, O., BRUNAK, S., VON HEIJNE, G. & NIELSEN, H. (2007). Locating proteins in the cell using TargetP, SignalP and related tools. *Nat Protoc*, **2**, 953–971. [8](#)
- ENDRES, M., NEUPERT, W. & BRUNNER, M. (1999). Transport of the ADP/ATP carrier of mitochondria from the TOM complex to the TIM22.54 complex. *EMBO J*, **18**, 3214–3221. [65](#)
- FAVOROV, A.V., GELFAND, M.S., GERASIMOVA, A.V., RAVCHEEV, D.A., MIRONOV, A.A. & MAKEEV, V.J. (2005). A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. *Bioinformatics*, **21**, 2240–2245. [27](#)
- FINK, J.L., ATURALIYA, R.N., DAVIS, M.J., ZHANG, F., HANSON, K., TEASDALE, M.S., KAI, C., KAWAI, J., CARNINCI, P., HAYASHIZAKI, Y. & TEASDALE, R.D. (2006). LOCATE: a mouse protein subcellular localization database. *Nucleic Acids Res*, **34**, D213–D217. [12](#), [114](#)

## REFERENCES

---

- GAO, Y. & MEHTA, K. (2007). N-linked glycosylation of CD38 is required for its structure stabilization but not for membrane localization. *Mol Cell Biochem*, **295**, 1–7. [67](#)
- GARDY, J.L., SPENCER, C., WANG, K., ESTER, M., TUSNÁDY, G.E., SIMON, I., HUA, S., DEFAYS, K., LAMBERT, C., NAKAI, K. & BRINKMAN, F.S.L. (2003). PSORT-B: Improving protein subcellular localization prediction for Gram-negative bacteria. *Nucleic Acids Res*, **31**, 3613–3617. [9](#)
- GARDY, J.L., LAIRD, M.R., CHEN, F., REY, S., WALSH, C.J., ESTER, M. & BRINKMAN, F.S.L. (2005). PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis. *Bioinformatics*, **21**, 617–623. [9](#), [13](#)
- GARNER, ROMERO, DUNKER, BROWN & OBRADOVIC (1999). Predicting binding regions within disordered proteins. *Genome Informatics*, **10**, 41–50. [112](#)
- GOLDFARB, D.S., GARIPY, J., SCHOOLNIK, G. & KORNBERG, R.D. (1986). Synthetic peptides as nuclear localization signals. *Nature*, **322**, 641–644. [108](#), [116](#)
- GOULD, S.G., KELLER, G.A. & SUBRAMANI, S. (1987). Identification of a peroxisomal targeting signal at the carboxy terminus of firefly luciferase. *J Cell Biol*, **105**, 2923–2931. [65](#)
- GOULD, S.J., KELLER, G.A., HOSKEN, N., WILKINSON, J. & SUBRAMANI, S. (1989). A conserved tripeptide sorts proteins to peroxisomes. *J Cell Biol*, **108**, 1657–1664. [65](#)



## REFERENCES

---

- GRIBSKOV, M., MCLACHLAN, A.D. & EISENBERG, D. (1987). Profile analysis: detection of distantly related proteins. [17](#)
- GUAN, J.L., MACHAMER, C.E. & ROSE, J.K. (1985). Glycosylation allows cell-surface transport of an anchored secretory protein. *Cell*, **42**, 489–496. [67](#)
- GUDA, C. & SUBRAMANIAM, S. (2005). pTARGET [CORRECTED] a new method for predicting protein subcellular localization in eukaryotes. *Bioinformatics*, **21**, 3963–3969. [7](#), [10](#), [13](#), [36](#), [70](#), [71](#)
- HANNINK, M. & DONOGHUE, D.J. (1986). Cell surface expression of membrane-anchored v-sis gene products: glycosylation is not required for cell surface transport. *J Cell Biol*, **103**, 2311–2322. [67](#)
- HENIKOFF, S. & HENIKOFF, J.G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, **89**, 10915–10919. [68](#)
- HERTZ, G.Z. & STORMO, G.D. (1999). Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577. [26](#)
- HINSBY, A.M., KIEMER, L., KARLBERG, E.O., LAGE, K., FAUSBLL, A., JUNCKER, A.S., ANDERSEN, J.S., MANN, M. & BRUNAK, S. (2006). A wiring of the human nucleolus. *Mol Cell*, **22**, 285–295. [114](#)
- HIROKAWA, T., BOON-CHIENG, S. & MITAKU, S. (1998). SOSUI: classification and secondary structure prediction system for membrane proteins. *Bioinformatics*, **14**, 378–379. [68](#), [137](#)

## REFERENCES

---

- HOBOHM, U., SCHARF, M., SCHNEIDER, R. & SANDER, C. (1992). Selection of representative protein data sets. *Protein Sci*, **1**, 409–417. [13](#), [15](#), [41](#), [56](#)
- HÖGLUND, A., DÖNNES, P., BLUM, T., ADOLPH, H.W. & KOHLBACHER, O. (2006). MultiLoc: prediction of protein subcellular localization using N-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*, **22**, 1158–1165. [7](#), [10](#), [11](#), [13](#), [41](#), [70](#), [71](#), [97](#), [98](#), [99](#), [180](#)
- HORTON, P. & NAKAI, K. (1997). Better prediction of protein cellular localization sites with the k nearest neighbors classifier. *Proc Int Conf Intell Syst Mol Biol*, **5**, 147–152. [7](#), [9](#)
- HORTON, P., PARK, K.J., OBAYASHI, T., FUJITA, N., HARADA, H., ADAMS-COLLIER, C.J. & NAKAI, K. (2007). WoLF PSORT: protein localization predictor. *Nucleic Acids Res*, **35**, W585–W587. [7](#), [9](#)
- HSU, C.W. & LIN, C.J. (2002). A simple decomposition method for support vector machines. *Machine Learning*, **46**, 291–314. [24](#), [79](#)
- HUA, S. & SUN, Z. (2001). Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, **17**, 721–728. [7](#), [10](#)
- HUBER, R. (1979). Conformational flexibility in protein molecules. *Nature*, **280**, 538–539. [111](#)
- HUGHES, J.D., ESTEP, P.W., TAVAZOIE, S. & CHURCH, G.M. (2000). Computational identification of cis-regulatory elements associated with groups of

## REFERENCES

---

- functionally related genes in *Saccharomyces cerevisiae*. *J Mol Biol*, **296**, 1205–1214. [27](#)
- HULO, N., BAIROCH, A., BULLIARD, V., CERUTTI, L., CASTRO, E.D., LANGENDIJK-GENEVAUX, P.S., PAGNI, M. & SIGRIST, C.J.A. (2006). The PROSITE database. *Nucleic Acids Res*, **34**, D227–D230. [11](#), [25](#), [40](#)
- HWANG, S., GOU, Z. & KUZNETSOV, I.B. (2007). DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics*, **23**, 634–636. [26](#)
- JIN, Y. & DUNBRACK, R.L. (2005). Assessment of disorder predictions in CASP6. *Proteins*, **61 Suppl 7**, 167–175. [113](#)
- JOACHIMS, T. (1999). Making large-scale support vector machine learning practical. 169–184, MIT Press. [23](#), [79](#)
- JONES, D.T. (2007). Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics*, **23**, 538–544. [137](#), [159](#)
- JONES, D.T., TAYLOR, W.R. & THORNTON, J.M. (1994). A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry*, **33**, 3038–3049. [142](#)
- JUNCKER, A.S., WILLENBROCK, H., HEIJNE, G.V., BRUNAK, S., NIELSEN, H. & KROGH, A. (2003). Prediction of lipoprotein signal peptides in Gram-negative bacteria. *Protein Sci*, **12**, 1652–1662. [7](#), [8](#)

## REFERENCES

---

- KÄLL, L., KROGH, A. & SONNHAMMER, E.L.L. (2004). A combined transmembrane topology and signal peptide prediction method. *J Mol Biol*, **338**, 1027–1036. [137](#), [139](#), [159](#), [160](#)
- KÄLL, L., KROGH, A. & SONNHAMMER, E.L.L. (2007). Advantages of combined transmembrane topology and signal peptide prediction—the Phobius web server. *Nucleic Acids Res*, **35**, W429–W432. [138](#), [164](#)
- KAPLAN, H.A., WELPLY, J.K. & LENNARZ, W.J. (1987). Oligosaccharyl transferase: the central enzyme in the pathway of glycoprotein assembly. *Biochim Biophys Acta*, **906**, 161–173. [67](#)
- KELLEY, L.P. & KINSELLA, B.T. (2003). The role of N-linked glycosylation in determining the surface expression, G protein interaction and effector coupling of the alpha isoform of the human thromboxane A(2) receptor. *Biochim Biophys Acta*, **1621**, 192–203. [67](#)
- KIEMER, L., BENDTSEN, J.D. & BLOM, N. (2005). NetAcet: prediction of N-terminal acetylation sites. *Bioinformatics*, **21**, 1269–1270. [38](#)
- KISSINGER, C.R., PARGE, H.E., KNIGHTON, D.R., LEWIS, C.T., PELLETIER, L.A., TEMPCZYK, A., KALISH, V.J., TUCKER, K.D., SHOWALTER, R.E. & MOOMAW, E.W. (1995). Crystal structures of human calcineurin and the human FKBP12-FK506-calcineurin complex. *Nature*, **378**, 641–644. [132](#)
- KLEIN, P., KANEHISA, M. & DELISI, C. (1984). Prediction of protein function from sequence properties. Discriminant analysis of a data base. *Biochim Biophys Acta*, **787**, 221–226. [68](#)

- KROGH, A. (2002). *www.binf.ku.dk/~krogh/docs/labeled\_fasta\_format.html*.  
[148](#)
- KROGH, A., LARSSON, B., VON HEIJNE, G. & SONNHAMMER, E.L. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J Mol Biol*, **305**, 567–580. [6](#), [68](#), [79](#), [123](#), [124](#), [134](#), [137](#), [138](#), [140](#)
- KUZNETSOV, I.B., GOU, Z., LI, R. & HWANG, S. (2006). Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins*, **64**, 19–27. [27](#)
- LA COUR, T., KIEMER, L., MØLGAARD, A., GUPTA, R., SKRIVER, K. & BRUNAK, S. (2004). Analysis and prediction of leucine-rich nuclear export signals. *Protein Eng Des Sel*, **17**, 527–536. [110](#)
- LAO, D.M., OKUNO, T. & SHIMIZU, T. (2002). Evaluating transmembrane topology prediction methods for the effect of signal peptide in topology prediction. *In Silico Biol*, **2**, 485–494. [69](#), [137](#)
- LEUNG, A.K.L., ANDERSEN, J.S., MANN, M. & LAMOND, A.I. (2003). Bioinformatic analysis of the nucleolus. *Biochem J*, **376**, 553–569. [111](#), [125](#)
- LI, ROMERO, RANI, DUNKER & OBRADOVIC (1999). Predicting protein disorder for N-, C-, and internal regions. *Genome Informatics*, **10**, 30–40. [112](#)

## REFERENCES

---

- LI, H. & BINGHAM, P.M. (1991). Arginine/serine-rich domains of the su(wa) and tra RNA processing regulators target proteins to a subnuclear compartment implicated in splicing. *Cell*, **67**, 335–342. [108](#)
- LI, W. & GODZIK, A. (2006). CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659. [14](#), [15](#), [36](#), [70](#), [71](#), [115](#)
- LI, W., JAROSZEWSKI, L. & GODZIK, A. (2001). Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**, 282–283. [14](#)
- LI, W., JAROSZEWSKI, L. & GODZIK, A. (2002). Tolerating some redundancy significantly speeds up clustering of large protein databases. *Bioinformatics*, **18**, 77–82. [14](#), [15](#)
- LINDING, R., JENSEN, L.J., DIELLA, F., BORK, P., GIBSON, T.J. & RUSSELL, R.B. (2003a). Protein disorder prediction: implications for structural proteomics. *Structure*, **11**, 1453–1459. [112](#)
- LINDING, R., RUSSELL, R.B., NEDUVA, V. & GIBSON, T.J. (2003b). GlobPlot: Exploring protein sequences for globularity and disorder. *Nucleic Acids Res*, **31**, 3701–3708. [112](#)
- LOBLEY, A., SWINDELLS, M.B., ORENGO, C.A. & JONES, D.T. (2007). Inferring function using patterns of native disorder in proteins. *PLoS Comput Biol*, **3**, e162. [102](#), [112](#), [168](#)

## REFERENCES

---

- LU, Z., SZAFRON, D., GREINER, R., LU, P., WISHART, D.S., POULIN, B., ANVIK, J., MACDONELL, C. & EISNER, R. (2004). Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, **20**, 547–556. [7](#), [10](#)
- LUMB, M.J., PURDUE, P.E. & DANPURE, C.J. (1994). Molecular evolution of alanine/glyoxylate aminotransferase 1 intracellular targeting. Analysis of the feline gene. *Eur J Biochem*, **221**, 53–62. [104](#)
- MACHAMER, C.E., FLORKIEWICZ, R.Z. & ROSE, J.K. (1985). A single N-linked oligosaccharide at either of the two normal sites is sufficient for transport of vesicular stomatitis virus G protein to the cell surface. *Mol Cell Biol*, **5**, 3074–3083. [67](#)
- MAEDA, N., KASUKAWA, T., OYAMA, R., GOUGH, J., FRITH, M., ENGSTRM, P.G., LENHARD, B., ATURALIYA, R.N., BATALOV, S., BEISEL, K.W., BULT, C.J., FLETCHER, C.F., FORREST, A.R.R., FURUNO, M., HILL, D., ITOH, M., KANAMORI-KATAYAMA, M., KATAYAMA, S., KATOH, M., KAWASHIMA, T., QUACKENBUSH, J., RAVASI, T., RING, B.Z., SHIBATA, K., SUGIURA, K., TAKENAKA, Y., TEASDALE, R.D., WELLS, C.A., ZHU, Y., KAI, C., KAWAI, J., HUME, D.A., CARNINCI, P. & HAYASHIZAKI, Y. (2006). Transcript annotation in FANTOM3: mouse gene catalog based on physical cDNAs. *PLoS Genet*, **2**, e62. [114](#)
- MAEDA, Y., HISATAKE, K., KONDO, T., HANADA, K., SONG, C.Z., NISHIMURA, T. & MURAMATSU, M. (1992). Mouse rRNA gene transcription

## REFERENCES

---

- factor mUBF requires both HMG-box1 and an acidic tail for nucleolar accumulation: molecular analysis of the nucleolar targeting mechanism. *EMBO J*, **11**, 3695–3704. [108](#)
- MARTELLI, P.L., FARISELLI, P. & CASADIO, R. (2003). An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins. *Bioinformatics*, **19 Suppl 1**, i205–i211. [137](#)
- MATSUDA, K., ZHENG, J., DU, G.G., KLCKER, N., MADISON, L.D. & DALLOS, P. (2004). N-linked glycosylation sites of the motor protein prestin: effects on membrane targeting and electrophysiological function. *J Neurochem*, **89**, 928–938. [67](#)
- MATTHEWS, B.W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta*, **405**, 442–451. [38](#), [47](#), [80](#)
- MILSTEIN, C., BROWNLEE, G.G., HARRISON, T.M. & MATHEWS, M.B. (1972). A possible precursor of immunoglobulin light chains. *Nat New Biol*, **239**, 117–120. [66](#)
- MOHRMANN, K., VAN EIJNDHOVEN, M.A.J., SCHINKEL, A.H. & SCHELLENS, J.H.M. (2005). Absence of N-linked glycosylation does not affect plasma membrane localization of breast cancer resistance protein (BCRP/ABCG2). *Cancer Chemother Pharmacol*, **56**, 344–350. [67](#)
- MOTLEY, A., LUMB, M.J., OATEY, P.B., JENNINGS, P.R., ZOYSA, P.A.D., WANDERS, R.J., TABAK, H.F. & DANPURE, C.J. (1995). Mammalian ala-



## REFERENCES

---

- nine/glyoxylate aminotransferase 1 is imported into peroxisomes via the PTS1 translocation pathway. Increased degeneracy and context specificity of the mammalian PTS1 motif and implications for the peroxisome-to-mitochondrion mistargeting of AGT in primary hyperoxaluria type 1. *J Cell Biol*, **131**, 95–109. [104](#)
- MÜLLER, S., CRONING, M.D. & APWEILER, R. (2001). Evaluation of methods for the prediction of membrane spanning regions. *Bioinformatics*, **17**, 646–653. [69](#), [137](#)
- MUNRO, S. (1995). An investigation of the role of transmembrane domains in Golgi protein retention. *EMBO J*, **14**, 4695–4704. [95](#)
- NAIR, R. & ROST, B. (2002a). Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, **18 Suppl 1**, S78–S86. [7](#), [9](#)
- NAIR, R. & ROST, B. (2002b). Sequence conserved for subcellular localization. *Protein Sci*, **11**, 2836–2847. [7](#), [9](#)
- NAIR, R. & ROST, B. (2004). LOCnet and LOCtarget: sub-cellular localization for structural genomics targets. *Nucleic Acids Res*, **32**, W517–W521. [9](#), [110](#)
- NAIR, R., CARTER, P. & ROST, B. (2003). NLSdb: database of nuclear localization signals. *Nucleic Acids Res*, **31**, 397–399. [8](#), [11](#), [110](#)
- NAKAI, K. & HORTON, P. (1999). PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization. *Trends Biochem Sci*, **24**, 34–36. [7](#), [9](#), [110](#)

## REFERENCES

---

- NAKAI, K. & KANEHISA, M. (1991). Expert system for predicting protein localization sites in gram-negative bacteria. *Proteins*, **11**, 95–110. [7](#), [9](#), [68](#)
- NAKAI, K. & KANEHISA, M. (1992). A knowledge base for predicting protein localization sites in eukaryotic cells. *Genomics*, **14**, 897–911. [9](#)
- NEDUVA, V. & RUSSELL, R.B. (2006). DILIMOT: discovery of linear motifs in proteins. *Nucleic Acids Res*, **34**, W350–W355. [26](#), [58](#)
- NG, P., NAGARAJAN, N., JONES, N. & KEICH, U. (2006). Apples to apples: improving the performance of motif finders and their significance analysis in the twilight zone. *Bioinformatics*, **22**, e393–e401. [34](#)
- NICKEL, W. (2003). The mystery of nonclassical protein secretion. A current view on cargo proteins and potential export routes. *Eur J Biochem*, **270**, 2109–2119. [66](#)
- NIELSEN, H. & KROGH, A. (1998). Prediction of signal peptides and signal anchors by a hidden Markov model. *Proc Int Conf Intell Syst Mol Biol*, **6**, 122–130. [160](#)
- NIELSEN, H., ENGELBRECHT, J., BRUNAK, S. & VON HEIJNE, G. (1997a). Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. *Protein Eng*, **10**, 1–6. [160](#)
- NIELSEN, H., ENGELBRECHT, J., BRUNAK, S. & VON HEIJNE, G. (1997b). A neural network method for identification of prokaryotic and eukaryotic signal

## REFERENCES

---

- peptides and prediction of their cleavage sites. *Int J Neural Syst*, **8**, 581–599. [8](#), [138](#)
- NIELSEN, H., BRUNAK, S. & VON HEIJNE, G. (1999). Machine learning approaches for the prediction of signal peptides and other protein sorting signals. *Protein Eng*, **12**, 3–9. [7](#)
- OBENAUER, J.C., CANTLEY, L.C. & YAFFE, M.B. (2003). Scansite 2.0: Proteome-wide prediction of cell signaling interactions using short sequence motifs. *Nucleic Acids Res*, **31**, 3635–3641. [25](#)
- ODA, T., MIYAJIMA, H., SUZUKI, Y. & ICHIYAMA, A. (1987). Nucleotide sequence of the cDNA encoding the precursor for mitochondrial serine:pyruvate aminotransferase of rat liver. *Eur J Biochem*, **168**, 537–542. [104](#)
- OSUMI, T., TSUKAMOTO, T., HATA, S., YOKOTA, S., MIURA, S., FUJIKI, Y., HIJIKATA, M., MIYAZAWA, S. & HASHIMOTO, T. (1991). Amino-terminal presequence of the precursor of peroxisomal 3-ketoacyl-coa thiolase is a cleavable signal peptide for peroxisomal targeting. *Biochem Biophys Res Commun*, **181**, 947–954. [65](#)
- PARK, K.J. & KANEHISA, M. (2003). Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, **19**, 1656–1663. [7](#), [10](#), [13](#)
- PAVESI, G., MEREGHETTI, P., MAURI, G. & PESOLE, G. (2004). Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Res*, **32**, W199–W203. [27](#)

## REFERENCES

---

- PELHAM, H.R. (1995). Sorting and retrieval between the endoplasmic reticulum and Golgi apparatus. *Curr Opin Cell Biol*, **7**, 530–535. [66](#), [88](#)
- PIERLEONI, A., MARTELLI, P.L., FARISELLI, P. & CASADIO, R. (2006). Ba-CellLo: a balanced subcellular localization predictor. *Bioinformatics*, **22**, e408–e416. [7](#), [10](#)
- PRILUSKY, J., FELDER, C.E., ZEEV-BEN-MORDEHAI, T., RYDBERG, E.H., MAN, O., BECKMANN, J.S., SILMAN, I. & SUSSMAN, J.L. (2005). FoldIndex: a simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics*, **21**, 3435–3438. [112](#)
- PUNTERVOLL, P., LINDING, R., GEMND, C., CHABANIS-DAVIDSON, S., MATTINGSDAL, M., CAMERON, S., MARTIN, D.M.A., AUSIELLO, G., BRANNETTI, B., COSTANTINI, A., FERR, F., MASELLI, V., VIA, A., CESARENI, G., DIELLA, F., SUPERTI-FURGA, G., WYRWICZ, L., RAMU, C., MCGUIGAN, C., GUDAVALI, R., LETUNIC, I., BORK, P., RYCHLEWSKI, L., KSTER, B., HELMER-CITTERICH, M., HUNTER, W.N., AASLAND, R. & GIBSON, T.J. (2003). ELM server: A new resource for investigating short functional sites in modular eukaryotic proteins. *Nucleic Acids Res*, **31**, 3625–3630. [25](#)
- PURDUE, P.E., LUMB, M.J. & DANPURE, C.J. (1992). Molecular evolution of alanine/glyoxylate aminotransferase 1 intracellular targeting. Analysis of the marmoset and rabbit genes. *Eur J Biochem*, **207**, 757–766. [104](#)

## REFERENCES

---

- QUEVILLON, E., SILVENTOINEN, V., PILLAI, S., HARTE, N., MULDER, N., APWEILER, R. & LOPEZ, R. (2005). InterProScan: protein domains identifier. *Nucleic Acids Res*, **33**, W116–W120. [58](#)
- RABINER, L.R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257–286. [16](#)
- RADIVOJAC, P., OBRADOVIC, Z., BROWN, C.J. & DUNKER, A.K. (2003). Prediction of boundaries between intrinsically ordered and disordered protein regions. *Pac Symp Biocomput*, 216–227. [112](#)
- RADIVOJAC, P., OBRADOVIC, Z., SMITH, D.K., ZHU, G., VUCETIC, S., BROWN, C.J., LAWSON, J.D. & DUNKER, A.K. (2004). Protein flexibility and intrinsic disorder. *Protein Sci*, **13**, 71–80. [112](#)
- RAMADASS, A.S. (2005). *Computational detection of regulatory signals in human genome sequence*. Ph.D. thesis, University of Cambridge. [74](#), [75](#)
- REINHARDT, A. & HUBBARD, T. (1998). Using neural networks for prediction of the subcellular location of proteins. *Nucleic Acids Res*, **26**, 2230–2236. [68](#)
- RICHARDSON, J.S. & RICHARDSON, D.C. (1988). Amino acid preferences for specific locations at the ends of alpha helices. *Science*, **240**, 1648–1652. [142](#)
- RIGOUTSOS, I. & FLORATOS, A. (1998). Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm. *Bioinformatics*, **14**, 55–67. [26](#), [58](#)

## REFERENCES

---

- ROBBINS, J., DILWORTH, S.M., LASKEY, R.A. & DINGWALL, C. (1991). Two interdependent basic domains in nucleoplasmin nuclear targeting sequence: identification of a class of bipartite nuclear targeting sequence. *Cell*, **64**, 615–623. [73](#)
- ROMERO, P., OBRADOVIC, Z. & DUNKER, A.K. (2004). Natively disordered proteins: functions and predictions. *Appl Bioinformatics*, **3**, 105–113. [112](#)
- SANDER, C. & SCHNEIDER, R. (1991). Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*, **9**, 56–68. [14](#)
- SCHERL, A., COUT, Y., DON, C., CALL, A., KINDBEITER, K., SANCHEZ, J.C., GRECO, A., HOCHSTRASSER, D. & DIAZ, J.J. (2002). Functional proteomic analysis of human nucleolus. *Mol Biol Cell*, **13**, 4100–4109. [113](#)
- SCHREIBER, V., MOLINETE, M., BOEUF, H., DE MURCIA, G. & DE MURCIA, J.M. (1992). The human poly(ADP-ribose) polymerase nuclear localization signal is a bipartite element functionally separate from DNA binding and catalytic activity. *EMBO J*, **11**, 3263–3269. [73](#)
- SCHULTZ, J., MILPETZ, F., BORK, P. & PONTING, C.P. (1998). SMART, a simple modular architecture research tool: identification of signaling domains. *Proc Natl Acad Sci U S A*, **95**, 5857–5864. [58](#)
- SINHA, S. & TOMPA, M. (2003). YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res*, **31**, 3586–3588. [27](#)

## REFERENCES

---

- SKILLING, J. (2004). Nested Sampling. In R. Fischer, R. Preuss & U.V. Toussaint, eds., *American Institute of Physics Conference Series*, 395–405. [20](#), [27](#), [28](#), [64](#), [69](#)
- SMITH, G.B. (1987). Preface to S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”. 562–563, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [18](#)
- SMOLA, A. & SCHOLKOPF, B. (1998). A tutorial on support vector regression. Tech. rep. [23](#)
- SORO, S. & TRAMONTANO, A. (2005). The prediction of protein function at CASP6. *Proteins*, **61 Suppl 7**, 201–213. [112](#), [159](#)
- SPRENGER, J., FINK, J.L. & TEASDALE, R.D. (2006). Evaluation and comparison of mammalian subcellular localization prediction methods. *BMC Bioinformatics*, **7 Suppl 5**, S3. [11](#)
- SWINKELS, B.W., GOULD, S.J., BODNAR, A.G., RACHUBINSKI, R.A. & SUBRAMANI, S. (1991). A novel, cleavable peroxisomal targeting signal at the amino-terminus of the rat 3-ketoacyl-coa thiolase. *EMBO J*, **10**, 3255–3262. [65](#)
- SZAFRON, D., LU, P., GREINER, R., WISHART, D.S., POULIN, B., EISNER, R., LU, Z., ANVIK, J., MACDONELL, C., FYSHE, A. & MEEUWIS, D. (2004). Proteome Analyst: custom predictions with explanations in a web-based tool for high-throughput proteome annotations. *Nucleic Acids Res*, **32**, W365–W371. [13](#)

## REFERENCES

---

- TAKADA, Y., KANEKO, N., ESUMI, H., PURDUE, P.E. & DANPURE, C.J. (1990). Human peroxisomal L-alanine: glyoxylate aminotransferase. Evolutionary loss of a mitochondrial targeting signal by point mutation of the initiation codon. *Biochem J*, **268**, 517–520. [104](#)
- THOMAS, P.D. & DILL, K.A. (1996). An iterative method for extracting energy-like quantities from protein structures. *Proc Natl Acad Sci U S A*, **93**, 11628–11633. [77](#), [78](#), [95](#), [97](#)
- TIPPING, M.E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, **1**, 211–244. [75](#)
- TOMPA, M., LI, N., BAILEY, T.L., CHURCH, G.M., MOOR, B.D., ESKIN, E., FAVOROV, A.V., FRITH, M.C., FU, Y., KENT, W.J., MAKEEV, V.J., MIRONOV, A.A., NOBLE, W.S., PAVESI, G., PESOLE, G., RGNIER, M., SIMONIS, N., SINHA, S., THIJS, G., VAN HELDEN, J., VANDENBOGAERT, M., WENG, Z., WORKMAN, C., YE, C. & ZHU, Z. (2005). Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, **23**, 137–144. [27](#)
- TUSNÁDY, G.E. & SIMON, I. (2001). The HMMTOP transmembrane topology prediction server. *Bioinformatics*, **17**, 849–850. [68](#), [137](#)
- VALENCIA, A. (2005). Protein refinement: a new challenge for CASP in its 10th anniversary. *Bioinformatics*, **21**, 277. [112](#), [159](#)
- VAPNIK, V. & LERNER, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, **24**, 774–780, 1963. [22](#)



## REFERENCES

---

- VIKLUND, H. & ELOFSSON, A. (2004). Best alpha-helical transmembrane protein topology predictions are achieved using hidden Markov models and evolutionary information. *Protein Sci*, **13**, 1908–1917. [137](#)
- VON HEIJNE, G. (1986). A new method for predicting signal sequence cleavage sites. *Nucleic Acids Res*, **14**, 4683–4690. [82](#), [137](#)
- VON HEIJNE, G. (1990). The signal peptide. *J Membr Biol*, **115**, 195–201. [66](#), [136](#)
- WARD, J.J., SODHI, J.S., MCGUFFIN, L.J., BUXTON, B.F. & JONES, D.T. (2004). Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J Mol Biol*, **337**, 635–645. [112](#)
- WEINREB, P.H., ZHEN, W., POON, A.W., CONWAY, K.A. & LANSBURY, P.T. (1996). NACP, a protein implicated in Alzheimer’s disease and learning, is natively unfolded. *Biochemistry*, **35**, 13709–13715. [111](#)
- WIEDEMANN, N., PFANNER, N. & RYAN, M.T. (2001). The three modules of ADP/ATP carrier cooperate in receptor recruitment and translocation into mitochondria. *EMBO J*, **20**, 951–960. [65](#)
- WORKMAN, C.T. & STORMO, G.D. (2000). ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. *Pac Symp Biocomput*, 467–478. [27](#)

## REFERENCES

---

- WRIGHT, P.E. & DYSON, H.J. (1999). Intrinsically unstructured proteins: re-assessing the protein structure-function paradigm. *J Mol Biol*, **293**, 321–331. [112](#), [168](#)
- WU, L.F., CHANAL, A. & RODRIGUE, A. (2000). Membrane targeting and translocation of bacterial hydrogenases. *Arch Microbiol*, **173**, 319–324. [103](#)
- YAN, K., KHOSHNOODI, J., RUOTSALAINEN, V. & TRYGGVASON, K. (2002). N-linked glycosylation is critical for the plasma membrane localization of nephrin. *J Am Soc Nephrol*, **13**, 1385–1389. [67](#)
- YANG, Z.R., THOMSON, R., MCNEIL, P. & ESNOUF, R.M. (2005). RONN: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics*, **21**, 3369–3376. [5](#), [102](#), [112](#), [114](#), [118](#)
- YU, C.S., LIN, C.J. & HWANG, J.K. (2004). Predicting subcellular localization of proteins for gram-negative bacteria by support vector machines based on n-peptide compositions. *Protein Sci*, **13**, 1402–1406. [7](#), [10](#)
- YUAN, Z. & TEASDALE, R.D. (2002). Prediction of Golgi Type II membrane proteins based on their transmembrane domains. *Bioinformatics*, **18**, 1109–1115. [66](#)
- ZASLAVSKY, E. & SINGH, M. (2006). A combinatorial optimization approach for diverse motif finding applications. *Algorithms Mol Biol*, **1**, 13. [26](#)

# Appendix A

## Motifs discovered by NestedMICA

Figures in this appendix shows the motifs discovered by NestedMICA (Chapter 2) in multiple runs on different eukaryotic localisation datasets (see 3.2.1). NestedMICA was run using the entire sequences as well as chunks of certain length from the N or C-terminal regions. Not all of these motifs, shown as sequence logos here, have been used in the development of Lokum.

The “Notes” columns in the below figures imply the sequence region (N-terminus, whole sequence, or C-terminus) a particular shown motif was discovered from. “First 20aa”, for example, indicates that the corresponding motif has been discovered within the first 20 N-terminal amino acid chunks of a particular localisation dataset.












#	Dataset	Motif	Notes
1	Nuclear		
2	Nuclear		
3	Nuclear		
4	Nuclear		
5	Nuclear		
6	Nuclear		
7	Nuclear		
8	Nuclear		
9	Nuclear		
10	Nuclear		
11	Nuclear		
12	Nuclear		
13	Nuclear		

Figure A.1: "Nuclear motifs" discovered by NestedMICA

#	Dataset	Motif	Notes
1	Plasma membrane		
2	Plasma membrane		
3	Plasma membrane		
4	Plasma membrane		
5	Plasma membrane		
6	Plasma membrane		
7	Plasma membrane		
8	Plasma membrane		
9	Plasma membrane		
10	Plasma membrane		
11	Plasma membrane		

Figure A.2: “Plasma membrane motifs” discovered by NestedMICA

---










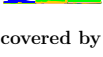
#	Dataset	Motif	Notes
1	Cytoplasmic		
2	Cytoplasmic		
3	Cytoplasmic		
4	Cytoplasmic		
5	Cytoplasmic		
6	Cytoplasmic		
7	Cytoplasmic		
8	Cytoplasmic		
9	Cytoplasmic		
10	Cytoplasmic		

Figure A.3: “Cytoplasmic motifs” discovered by NestedMICA






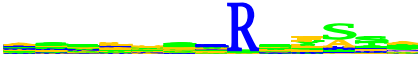
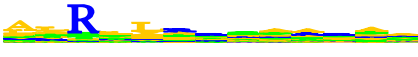














#	Dataset	Motif	Notes
1	Mitochondrial		
2	Mitochondrial		
3	Mitochondrial		
4	Mitochondrial		
5	Mitochondrial		
6	Mitochondrial		
7	Mitochondrial		
8	Mitochondrial		
9	Mitochondrial		
10	Mitochondrial		
11	Mitochondrial		
12	Mitochondrial		
13	Mitochondrial		
14	Mitochondrial		
15	Mitochondrial		
16	Mitochondrial		
17	Mitochondrial		
18	Mitochondrial		
19	Mitochondrial		
20	Mitochondrial		
21	Mitochondrial		

Figure A.4: "Mitochondrial motifs" discovered by NestedMICA

#	Dataset	Motif	Notes
1	ER		
2	ER		
3	ER		
4	ER		
5	ER		
6	ER		
7	ER		
8	ER		
9	ER		
10	ER		

Figure A.5: “Endoplasmic reticulum motifs” discovered by NestedMICA



#	Dataset	Motif	Notes
1	Golgi		
2	Golgi		
3	Golgi		
4	Golgi		
5	Golgi		
6	Golgi		
7	Golgi		
8	Golgi		
9	Golgi		
10	Golgi		
11	Golgi		
12	Golgi		
13	Golgi		
14	Golgi		
15	Golgi		

Figure A.6: “Golgi motifs” discovered by NestedMICA

---












#	Dataset	Motif	Notes
1	Extracellular		
2	Extracellular		
3	Extracellular		
4	Extracellular		
5	Extracellular		
6	Extracellular		
7	Extracellular		
8	Extracellular		
9	Extracellular		
10	Extracellular		
11	Extracellular		

Figure A.7: “Extracellular motifs” discovered by NestedMICA

#	Dataset	Motif	Notes
1	Lysosome		
2	Lysosome		
4	Lysosome		
4	Lysosome		
5	Lysosome		
6	Lysosome		
7	Lysosome		
8	Lysosome		
9	Lysosome		

Figure A.8: "Lysosome motifs" discovered by NestedMICA

#	Dataset	Motif	Notes
1	Peroxisomal		
2	Peroxisomal		
4	Peroxisomal		
4	Peroxisomal		
5	Peroxisomal		
6	Peroxisomal		
7	Peroxisomal		

Figure A.9: "Peroxisomal motifs" discovered by NestedMICA

#	Dataset	Motif	Notes
1	Vacuolar		
2	Vacuolar		
4	Vacuolar		
4	Vacuolar		
5	Vacuolar		
6	Vacuolar		
7	Vacuolar		
8	Vacuolar		
9	Vacuolar		
10	Vacuolar		
11	Vacuolar		

Figure A.10: “Vacuolar motifs” discovered by NestedMICA

# Appendix B

## Amino acid composition rates in different localisations

Figures in this appendix show frequency distributions for each of the 20 amino acids in 11 eukaryotic subcellular localisation classes. The provided amino acid composition rates were obtained from the redundancy-reduced protein sequence datasets used in the training and testing of Multiloc (Höglund *et al.*, 2006), a eukaryotic localisation predictor. All the given composition values sum up to 1.0 for each localisation class.

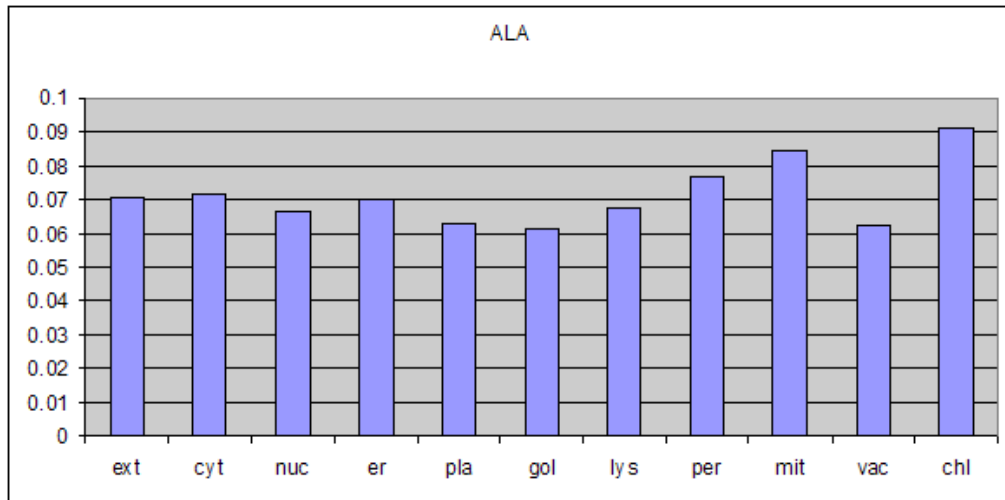


Figure B.1: Amino acid composition for Alanine (ALA / A)

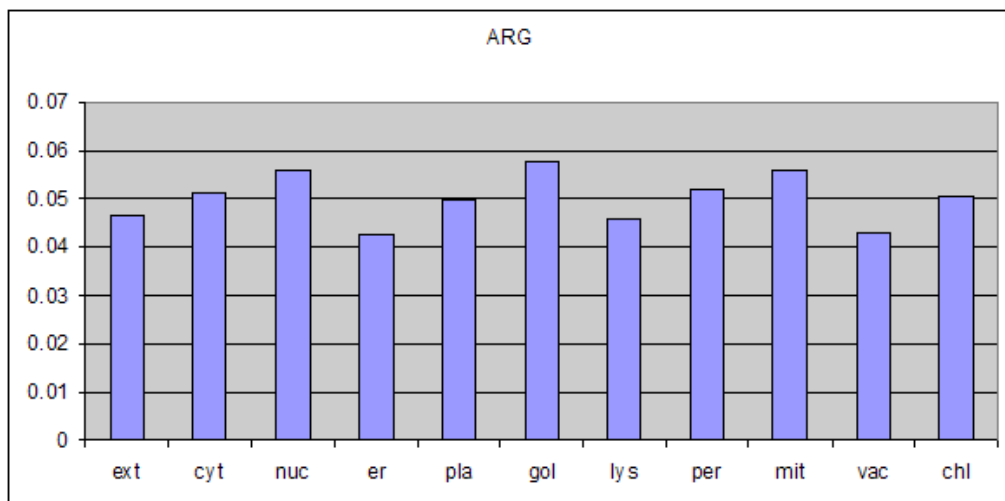


Figure B.2: Amino acid composition for Arginine (ARG / R)

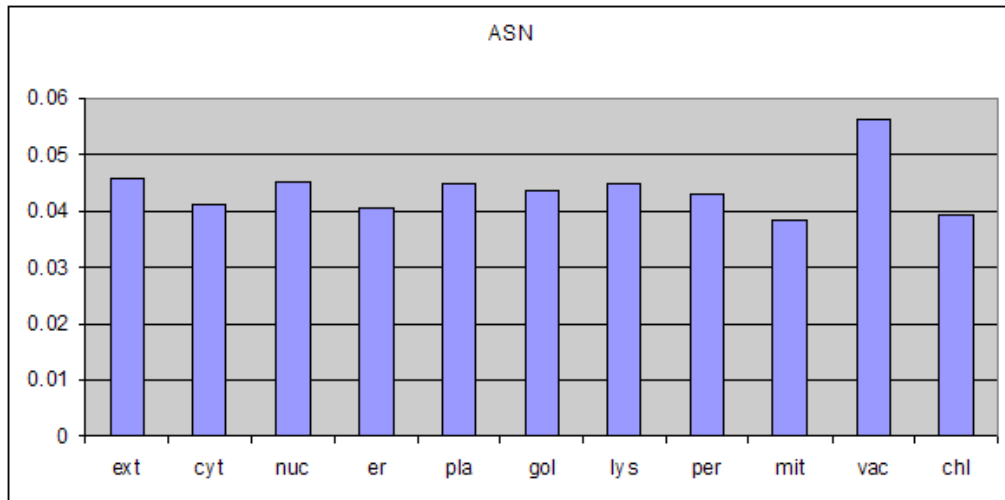


Figure B.3: Amino acid composition for Asparagine (ASN / N)

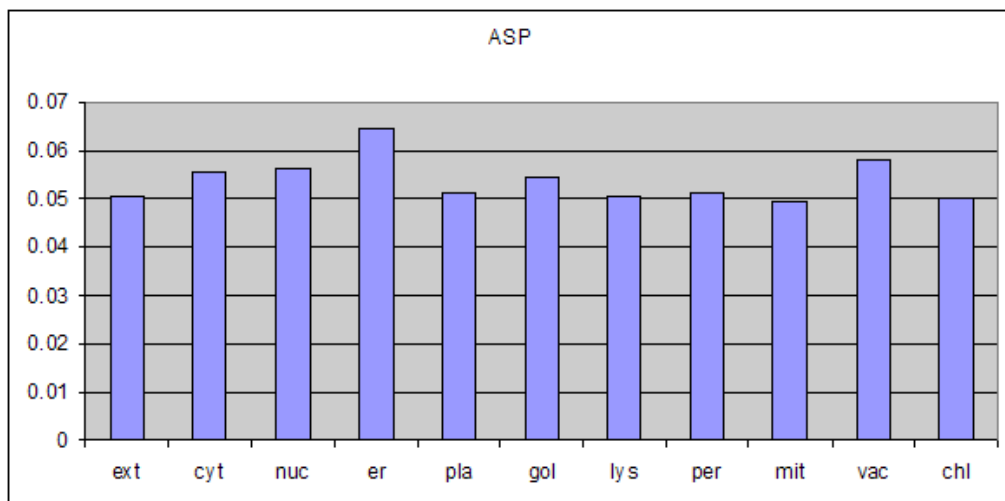


Figure B.4: Amino acid composition for Aspartic Acid (ASP / D)

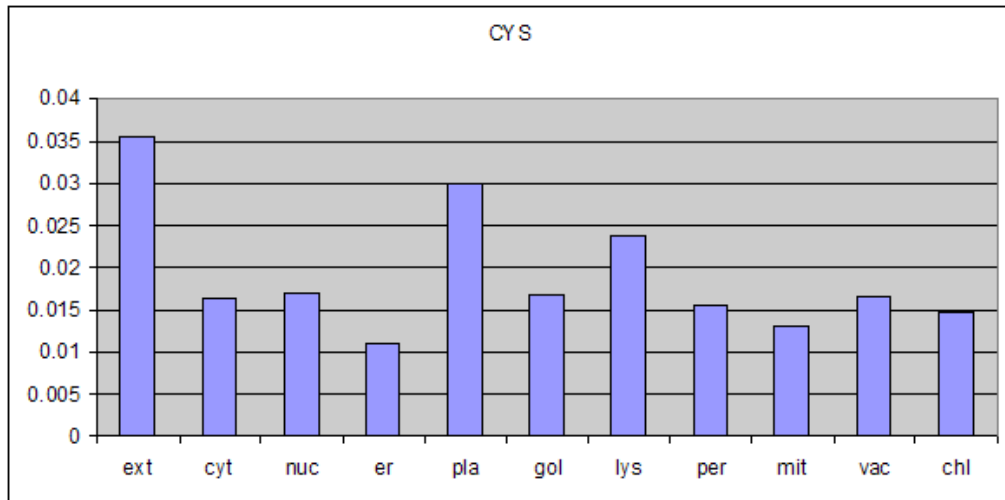


Figure B.5: Amino acid composition for Cysteine (CYS / C)

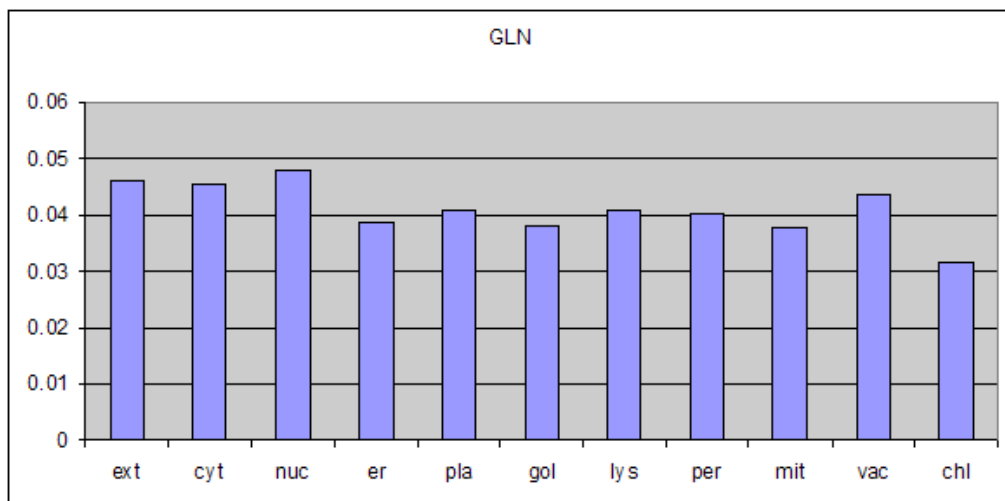


Figure B.6: Amino acid composition for Glutamine (GLN / Q)



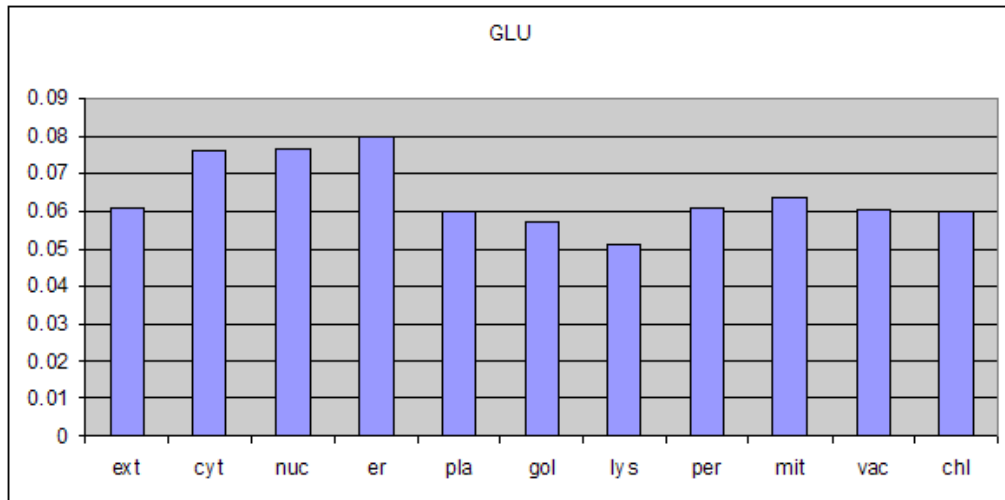


Figure B.7: Amino acid composition for Glutamic Acid (GLU / E)

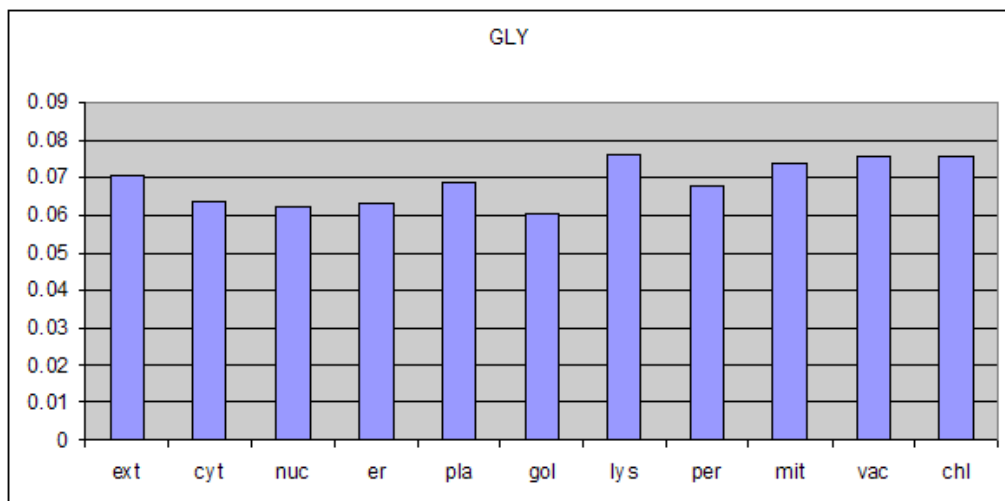


Figure B.8: Amino acid composition for Alanine (GLY / G)

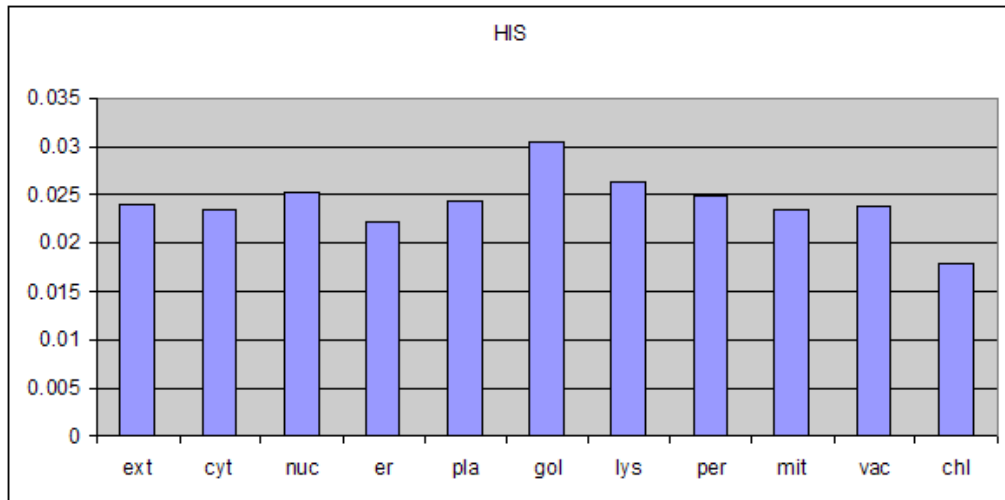


Figure B.9: Amino acid composition for Histidine (HIS / H)

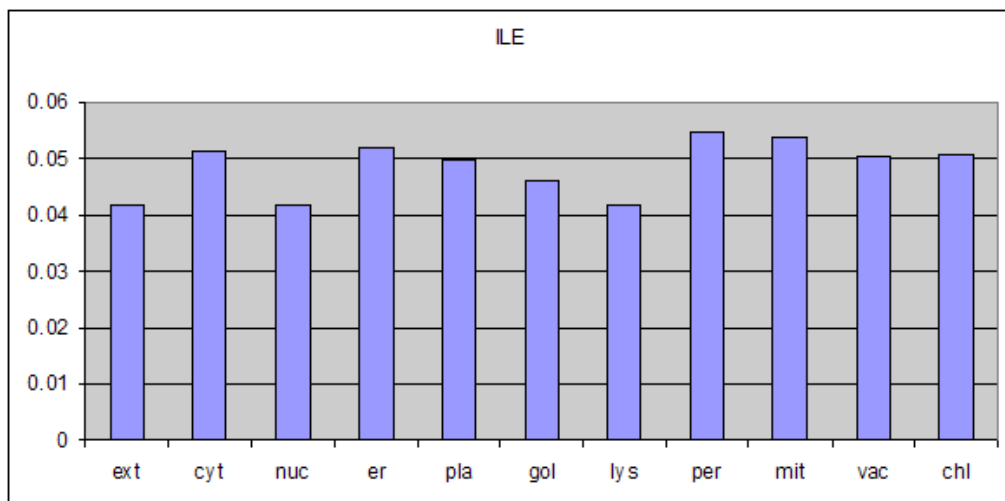


Figure B.10: Amino acid composition for Isoleucine (ILE / I)

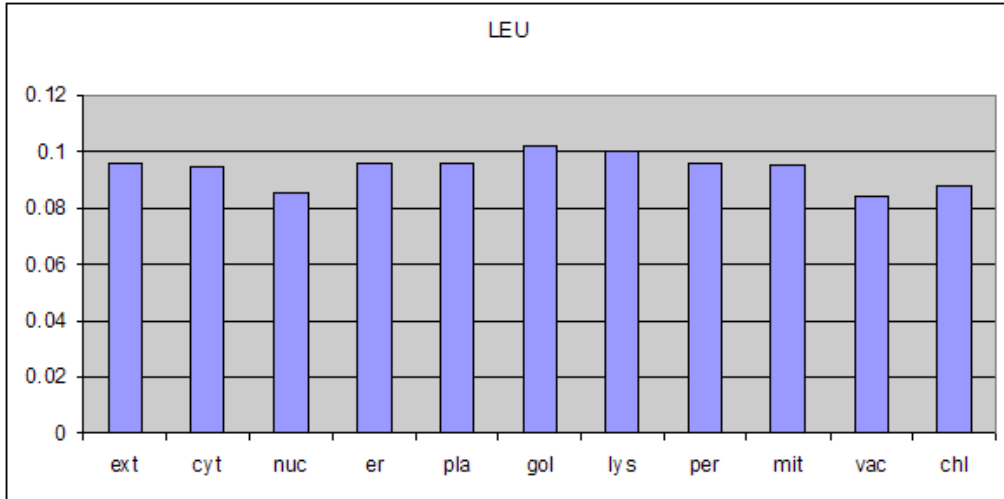


Figure B.11: Amino acid composition for Leucine (LEU / L)

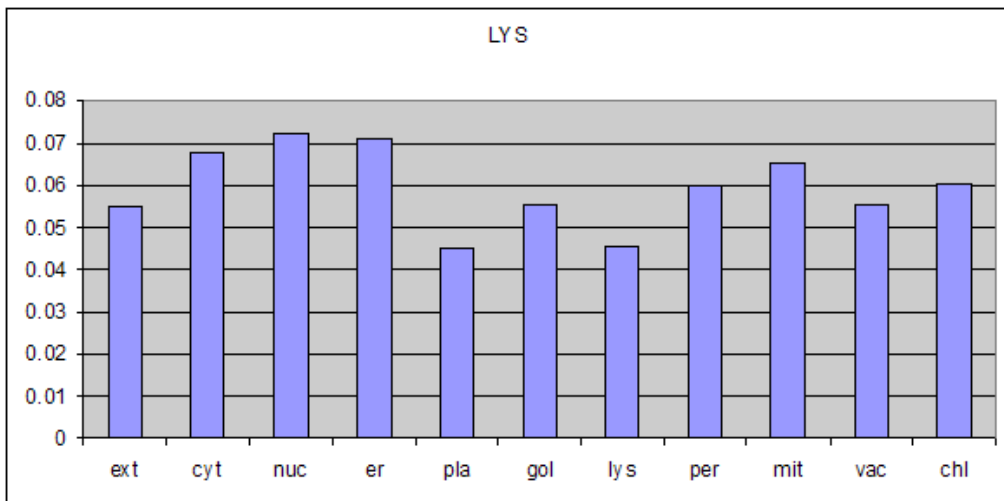


Figure B.12: Amino acid composition for Lysine (LYS / K)

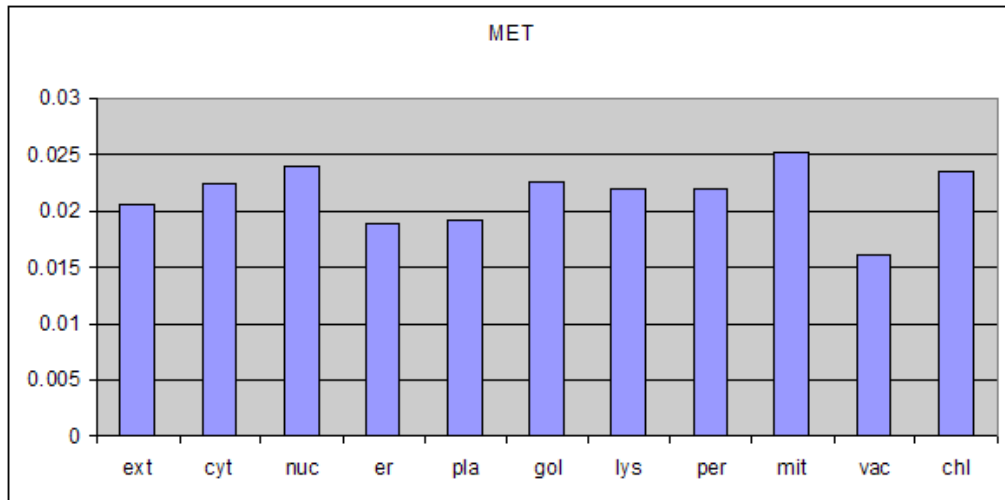


Figure B.13: Amino acid composition for Methionine (MET / M)

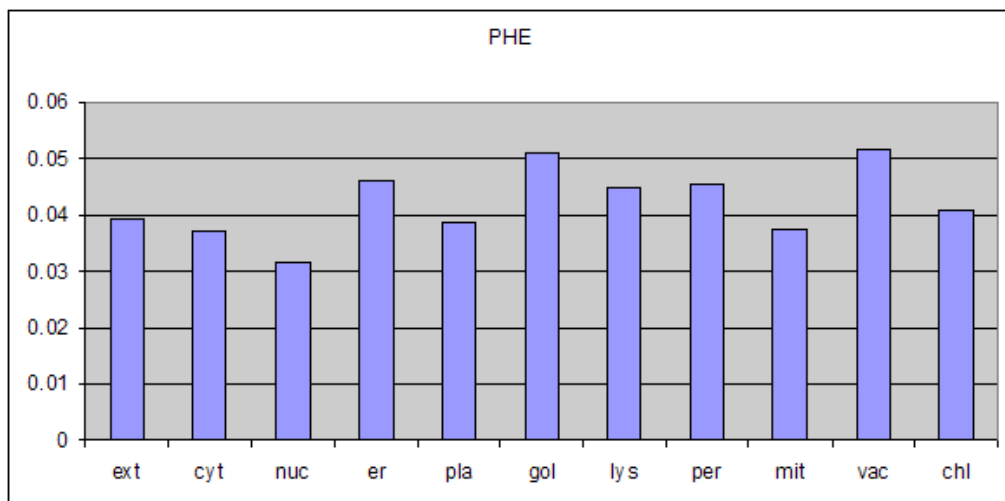


Figure B.14: Amino acid composition for Phenylalanine (PHE / F)

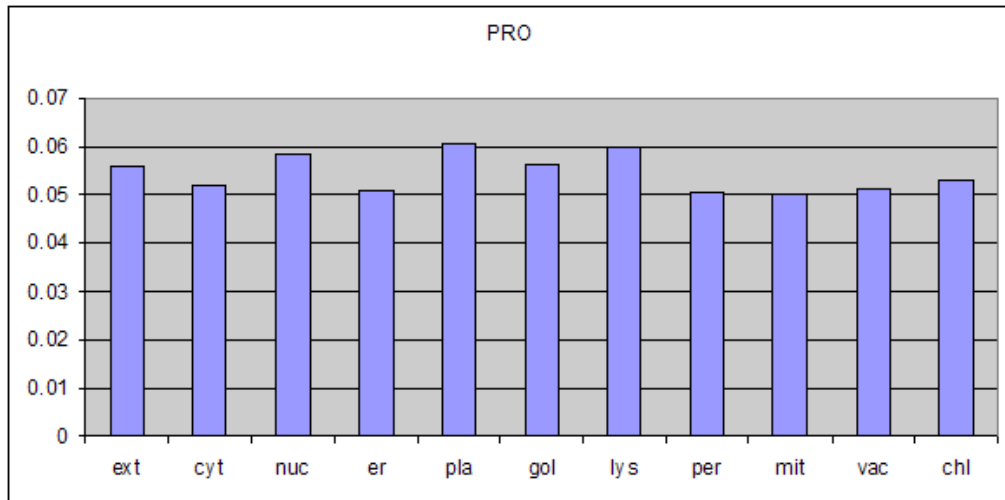


Figure B.15: Amino acid composition for Proline (PRO / P)

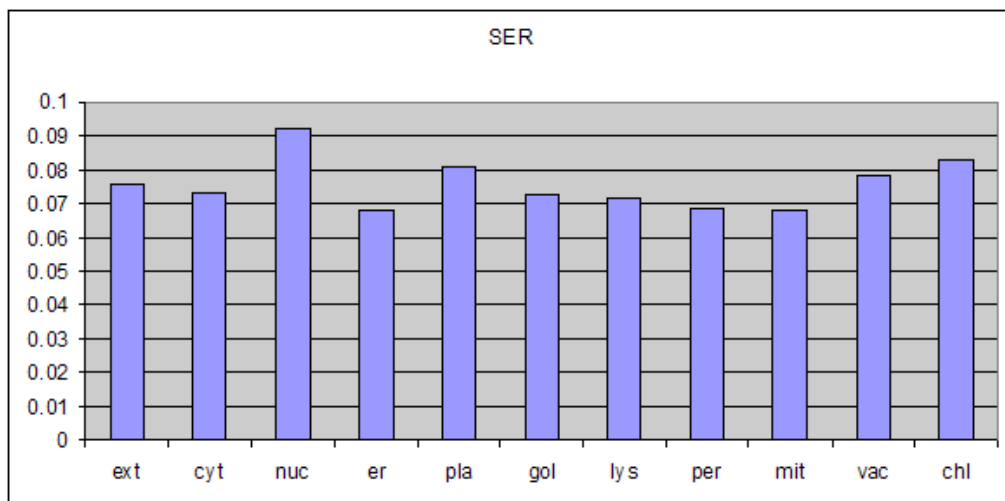


Figure B.16: Amino acid composition for Serine (SER / S)

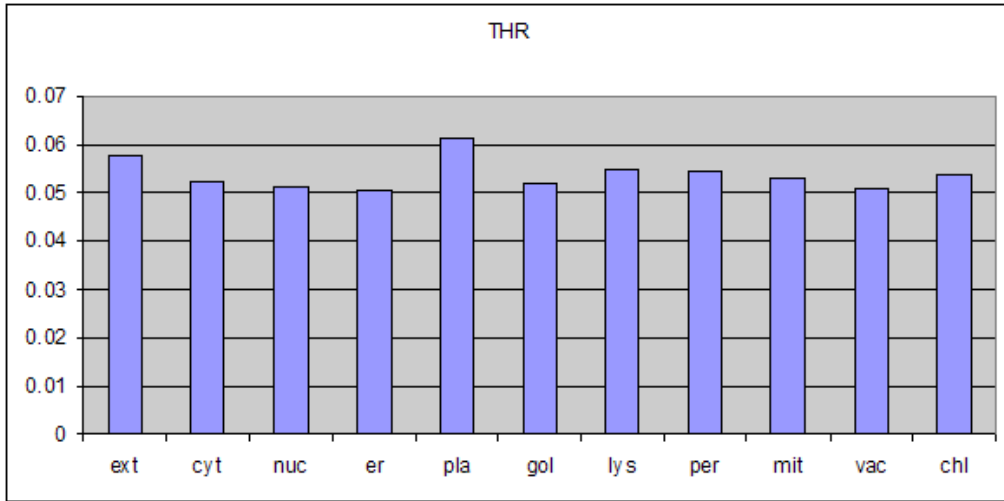


Figure B.17: Amino acid composition for Threonine (THR / T)

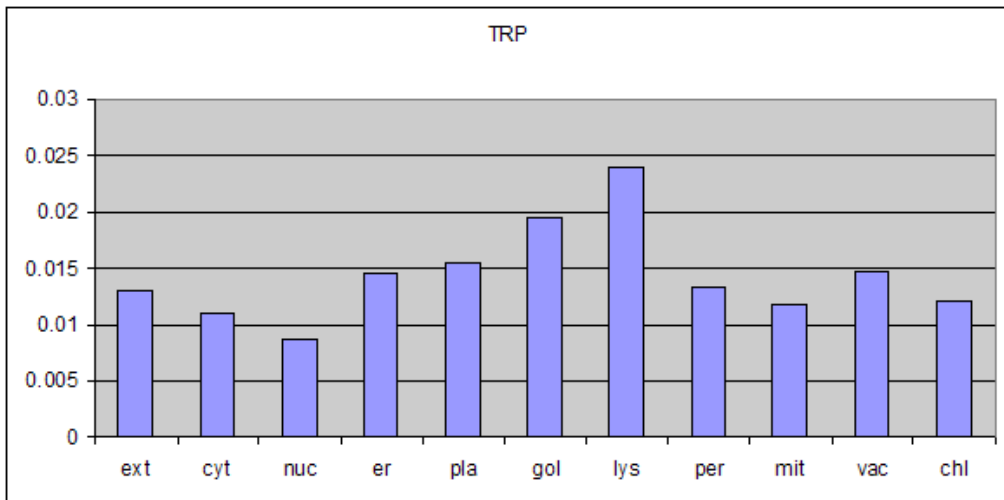


Figure B.18: Amino acid composition for Tryptophan (TRP / W)

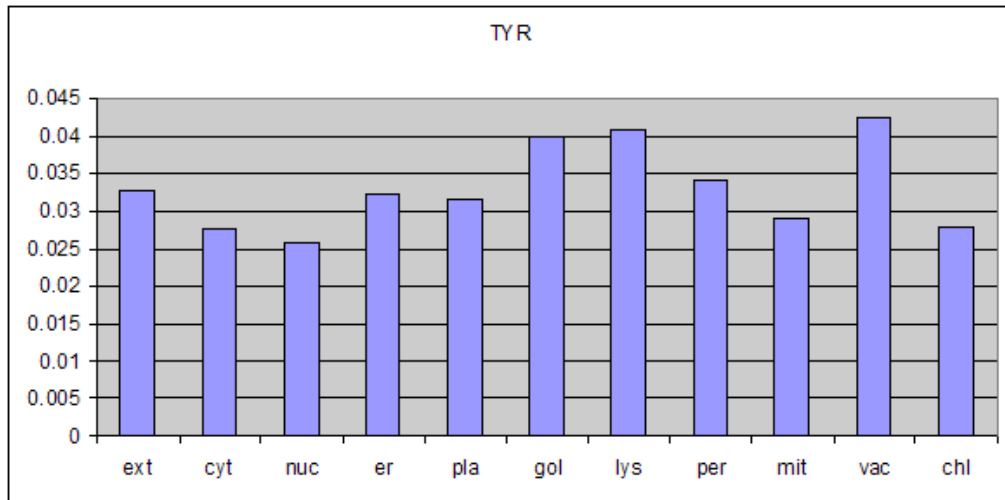


Figure B.19: Amino acid composition for Tyrosine (TYR / Y)

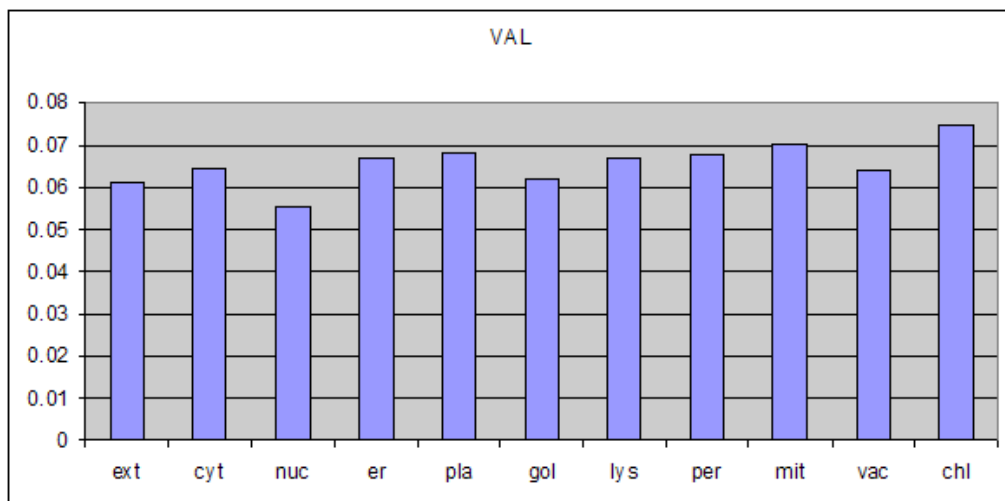


Figure B.20: Amino acid composition for Valine (VAL / V)

# Appendix C

## Sequence IDs of nuclear and nucleolar proteins filtered from the LOCATE database

### C.1 Proteins in nucleoli

5832447M01	8030477B02	AAH20037	B130024I17
0610007L03	9030008E11	AAH21402	B230113I11
0610010G24	9030015I21	AAH21438	B230341P13
0610010K23	9030404K10	AAH21497	B230345A13
0610010L07	9430023O10	AAH21646	B230384I08
0610012B16	9430042M18	AAH21922	B430205I06
0610037N12	9430068A02	AAH22656	B430304I01
0610041G09	9430070M15	AAH23108	BAA13139
0610043M01	9630032J03	AAH23495	BAA19479
0710005E17	9630058I18	AAH23755	BAA88301
1110007P10	9830141H16	AAH24049	BAA95050
1110017C15	9930013P05	AAH24718	BAB68541
1110017O22	9930036K22	AAH24730	C130053F21
1190002L16	A130086A08	AAH24881	C130060J12
1190005P17	A230078D05	AAH25074	C130087M08



## C.1 Proteins in nucleoli

---

1200003I18	A330071M14	AAH26492	C230037L02
1500005E20	A530027J07	AAH27220	C230071K24
1600021G09	A530056M01	AAH27223	C430045D17
1700010I21	A630008G24	AAH27357	C730016M05
1700020D05	A730016J17	AAH27399	C920029A19
1700026C17	A830025P17	AAH28246	CAA31278
1810029B16	AAA40067	AAH28305	CAA32372
1810063O22	AAA64248	AAH28640	CAA40012
1810073C22	AAB01504	AAH28860	CAA43091
2010206B19	AAB03664	AAH29834	CAA50196
2010300E13	AAB08894	AAH29892	CAA59260
2200007C21	AAB22970	AAH30169	CAB09797
2210401D21	AAB48630	AAH30493	CAD59182
2310002H12	AAB50013	AAH31127	CAE11688
2310039I18	AAB63526	AAH31531	D030042B21
2310040C05	AAB63915	AAH32932	D130027G07
2310057C03	AAB91426	AAH34506	D130070F09
2310057K05	AAB94491	AAH34516	D130072G11
2310061O04	AAB96870	AAH37634	D330049F08
2400004F19	AAC08435	AAH37681	D430026L04
2400011D10	AAC32982	AAH39185	D430043E23
2410041L12	AAC37664	AAH39648	D630003B12
2410089D17	AAC40061	AAH42502	D830050A13
2410115I17	AAC53171	AAH42708	E130104C03
2410130M07	AAC62511	AAH42940	E230013K19
2510038A11	AAC79683	AAH43014	E230019A18
2510039P04	AAD02877	AAH43017	E330001M23
2610204M17	AAD08676	AAH46977	E330016H10
2610507A14	AAD15718	AAH48190	E330019F09
2700027I18	AAD26855	AAH48412	E330028F04
2700052B17	AAD32094	AAH48685	E430003J02
2700066J21	AAF25951	AAH48709	E430005G16
2700067M10	AAF80246	AAH49118	E430008H02
2810004E23	AAH02004	AAH49166	E430012M21
2810012N22	AAH02014	AAH49245	E430014C08
2810017J07	AAH02025	AAH49565	E430014G22
2810026E11	AAH02027	AAH49928	E430014N21
2810037I08	AAH02044	AAH51673	E430019K12
2810453C09	AAH02079	AAH52386	E430020A18
2810473M21	AAH02108	AAH52401	E430031K14
2810486E17	AAH02306	AAH52482	E430031M22

## C.1 Proteins in nucleoli

---

2900001K19	AAH03244	AAH52790	E860029H08
3010025E17	AAH03261	AAH53333	F630017O19
3100001N19	AAH03709	AAH53404	F630021E13
3200001N24	AAH03775	AAH53453	F630048K01
4432409G09	AAH03885	AAH54085	F630105J12
4732414G15	AAH04028	AAH54541	F630222J08
4831429D18	AAH05547	AAH54723	F630223G06
4832420E07	AAH05734	AAH54778	F730043N02
4833436C12	AAH05776	AAH55393	F830044L17
4833442I16	AAH06631	AAH55484	F830213J22
4930408P03	AAH06684	AAH55787	G270004D20
4930417F03	AAH06805	AAH55860	G270124L20
4930429N24	AAH07174	AAH56232	G430020C23
4930512K19	AAH07487	AAH56383	G430046N16
4930528I04	AAH08161	AAH56650	G430074J02
4930558P17	AAH08270	AAH56992	G430138A13
4930563C04	AAH09100	AAH57033	G430146M18
4931421E07	AAH09142	AAH57054	G630007K23
4932409F19	AAH10987	AAH57156	G830049J11
4932434G09	AAH11213	AAH57342	G930019G02
4933403E10	AAH11248	AAH57645	G930027I02
4933431P07	AAH11484	AAH59089	I0C0003A04
5330437I08	AAH12276	AAH59822	I0C0030N23
5430425F10	AAH12281	AAH60072	I0C0040N18
5730405D16	AAH12433	AAH60147	I1C0027A21
5730406H19	AAH12641	AAH60375	I1C0031M12
5730419M09	AAH13165	AAH60959	I420019J01
5730436C18	AAH13618	AAH62146	I420024J09
5730470K22	AAH14688	AAH63100	I530003K08
5730563P06	AAH14703	AAH63748	I730026M06
5730589J07	AAH16194	AAH63755	I730039C23
5830405E04	AAH16489	AAH64712	I730045C06
5830465M17	AAH16569	AAK01204	I730051L04
6030446B09	AAH16676	AAK49787	I830034K17
6030461M12	AAH17637	AAK70403	I830055B02
6330401E03	AAH18321	AAL27006	I920011D05
6430407C24	AAH18373	AAL62331	I920020L14
6430528J02	AAH18399	AAL74402	I920030N07
6430603C09	AAH18545	AAO15605	I920037I11
6430628D06	AAH19218	AAO18683	I920056H18
6720463L11	AAH19418	AAR87796	I920065D03

6720473C09	AAH19535	B020030J01	I920089B02
7330416M24	AAH19693	B130012H23	I920089E19
			K530012F21

Table C.1: **Nucleolar proteins.** FANTOM3 IDs of nucleolar proteins which are filtered from LOCATE database (Chapter 4).

## C.2 Proteins in nuclei

0610010G04	6430529J03	AAH19520	C330049H01
0710005K22	6430549H08	AAH20099	C430003H13
1110003H09	6430598F23	AAH20990	C730024K17
1110021J02	6820408J04	AAH21306	C920026J05
1110067L22	7120441D04	AAH21750	CAA31138
1110069I04	7120476M05	AAH21839	CAA31808
1200009L24	7420438E06	AAH22600	CAA31957
1300002I11	8430431N14	AAH22628	CAA32372
1500010M05	9130009B16	AAH22681	CAA33096
1500017I02	9130019G03	AAH22733	CAA33373
1600032G08	9130211K13	AAH23110	CAA43091
1700003P16	9130217P20	AAH23324	CAA43723
1700019E19	9330101O11	AAH23775	CAA55350
1700028K03	9330177B18	AAH23815	CAA56450
1700030B17	9430072B20	AAH23915	CAA63733
1700030G05	9530046I22	AAH23961	CAA70213
1700067K01	9630025P05	AAH24341	CAA72404
1810037C20	9630027H07	AAH24521	CAA76637
1810046K07	9630029K22	AAH25073	CAB60732
2010001O09	9630045G21	AAH25602	CAB86873
2010300P09	9630050P12	AAH26841	CAC83967
2210002J07	9630054L03	AAH28871	D030011P09
2310008J22	9830147C21	AAH29834	D030020D13
2310014B11	9830160I16	AAH30915	D030053M22
2310042K15	9830169A11	AAH31168	D030054H07
2310043K02	A030005M07	AAH31463	D130019F14
2310047L21	A230011H19	AAH31769	D130064J02
2400002C23	A230039L04	AAH34855	D130084E01
2400011D10	A230054A08	AAH35298	D230016D14

## C.2 Proteins in nuclei

---

2410003J06	A230057M07	AAH36287	D230040E22
2410012M07	A230078I01	AAH37187	D330006L06
2410046L22	A230084K08	AAH37695	D430003F23
2410089D17	A230106F01	AAH40370	D530036A19
2410141M05	A330080J22	AAH43086	D630039M16
2510005J23	A430043P11	AAH46286	D830005I23
2510049I19	A530086E13	AAH47152	D930007K17
2610021E10	A730013C09	AAH48503	D930030M14
2700099C19	A730020L03	AAH48779	D930033J18
2810004A21	A730063C17	AAH50803	E030029E20
2810021G24	A730096N15	AAH51049	E030029N02
2810039M17	A830038H15	AAH51261	E130012E07
2810417H13	A830097I09	AAH51631	E130013F06
2810457D07	A930007L12	AAH51967	E130118K14
2900074D10	A930041F19	AAH52030	E130303A03
3000002C10	A930104E21	AAH52173	E230001H19
3100002L24	AAA20039	AAH52468	E330032L15
3110007F17	AAA37184	AAH52672	E330034H06
3110030B08	AAA37291	AAH52856	E430001M22
3200002N09	AAA97500	AAH53409	E430007C11
3732413B21	AAA98977	AAH54456	E430007F09
4432404N24	AAB24330	AAH54768	E430014D12
4631408J24	AAB40892	AAH56922	E970008A17
4632401G08	AAB41327	AAH57096	F420011N06
4632404G05	AAB65839	AAH57165	F530014L05
4632415C14	AAB70094	AAH57205	F630011I01
4632417G13	AAB81245	AAH57453	F630205L24
4732403I07	AAC02226	AAH58103	F730014I05
4732424P06	AAC36358	AAH60072	F730216I21
4732458H05	AAC40148	AAH60234	F830002J06
4732467A04	AAC52994	AAH60613	F830007J16
4833406K08	AAD00238	AAH61493	F830017M05
4833413D08	AAD13139	AAH64018	F830022C10
4921510H08	AAD39396	AAH64757	F830108M10
4921531G14	AAF27311	AAH65165	F830211J08
4922501K05	AAF27551	AAK07621	G430032E03
4930433I11	AAF63757	AAK35053	G430037K05
4930538K15	AAF72874	AAK39099	G430090A08
4931400M17	AAF86375	AAK39438	G530118I12
4931400O07	AAG01633	AAK60496	G630048M14
4931408L03	AAG29950	AAL09305	G730014O11

## C.2 Proteins in nuclei

---

4931409I21	AAG34081	AAL40860	G730050K01
4932416N14	AAG34793	AAL47577	G830045E23
4932441K08	AAG40809	AAL67834	I0C0040N18
4933400A06	AAG50171	AAL69526	I0C0048H21
4933415E13	AAH03259	AAL71902	I0C0048J01
4933439J20	AAH03266	AAM33069	I0C0048L09
5330404L13	AAH03292	AAM64199	I1C0020M01
5330418E10	AAH03330	AAM77216	I1C0033H16
5430425F10	AAH04738	B020012J09	I420001D20
5430434J22	AAH05426	B130019L12	I420006B09
5530401L07	AAH05516	B230111C05	I420014I19
5730407F12	AAH05620	B230120H23	I420025C06
5730438N18	AAH05694	B230213G02	I420033H08
5730548J20	AAH05744	B230309O17	I530008I17
5730592N24	AAH06016	B230375D17	I530014J18
5930431H10	AAH06939	BAA05885	I530027I10
6330405E07	AAH09004	BAA21725	I530028C19
6330414C15	AAH10496	BAA23648	I830025I02
6330417C18	AAH10841	BAA95075	I830031G17
6330437A14	AAH11091	BAA96361	I830037L15
6330503C03	AAH11131	BAB79232	I830043E20
6330513G01	AAH12715	BAC53845	I830128O08
6330541F16	AAH12953	BAC75669	I920021I20
6330562H21	AAH13718	C130032B15	I920062F22
6430402E12	AAH14828	C130083B17	I920087J04
6430519P13	AAH19168	C230004D03	K230011N15
K230305H19	K230320F07		

Table C.2: **Nuclear proteins.** FANTOM3 IDs of nuclear proteins which are filtered from LOCATE database (Chapter 4).

## Appendix D

Kullback-Leibler divergence for  
transmembrane helix cap  
positions in terms of amino acid  
composition

---

Amino acid	1 - 1	1 - 2	1 - 3	1 - 4	1 - 5
GLY	-0.016	0.003	-0.031	-0.017	-0.028
PHE	0.008	-0.051	-0.047	-0.049	-0.039
SER	-0.024	0.034	0.001	-0.004	0.009
PRO	-0.019	-0.014	-0.008	0.001	0.012
TYR	-0.006	-0.020	-0.017	-0.017	-0.019
ARG	0.163	0.505	0.544	0.575	0.531
TRP	-0.011	-0.020	-0.023	-0.021	-0.020
ALA	-0.006	-0.053	-0.050	-0.041	-0.057
LYS	0.155	0.530	0.636	0.547	0.610
ASN	0.004	0.061	0.068	0.071	0.103
GLN	-0.019	0.024	0.022	0.032	0.014
CYS	0.008	0.034	0.016	0.014	0.012
MET	-0.011	-0.018	-0.021	-0.023	-0.018
THR	-0.014	0.008	0.008	0.000	0.017
VAL	-0.002	-0.052	-0.044	-0.056	-0.058
ILE	-0.004	-0.049	-0.044	-0.050	-0.052
ASP	-0.017	0.108	0.130	0.141	0.167
LEU	-0.013	-0.085	-0.082	-0.077	-0.086
HIS	-0.002	0.025	0.029	0.019	0.027
GLU	-0.034	0.062	0.062	0.076	0.121
TOTAL KL	0.137	1.032	1.148	1.122	1.245

Table D.1: **KL deviations of cytoplasmic side helix positions from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates.** The “1-1” column lists the KL distances for the first position of the cytoplasmic side cap and the first position of the non-cytoplasmic side, and so on.

---

Amino acid	2 - 1	2 - 2	2 - 3	2 - 4	2 - 5
GLY	-0.027	-0.012	-0.041	-0.028	-0.038
PHE	0.086	-0.027	-0.021	-0.025	-0.005
SER	-0.031	0.021	-0.009	-0.012	-0.001
PRO	-0.022	-0.017	-0.012	-0.003	0.007
TYR	0.030	0.008	0.013	0.013	0.009
ARG	-0.036	0.013	0.019	0.023	0.017
TRP	0.021	-0.006	-0.014	-0.007	-0.005
ALA	0.095	-0.003	0.004	0.022	-0.012
LYS	-0.031	0.013	0.025	0.015	0.022
ASN	-0.028	-0.008	-0.005	-0.004	0.007
GLN	-0.022	-0.009	-0.009	-0.006	-0.012
CYS	-0.006	0.006	-0.002	-0.003	-0.004
MET	0.054	0.029	0.021	0.014	0.029
THR	-0.021	-0.001	-0.001	-0.009	0.007
VAL	0.134	0.008	0.028	-0.002	-0.007
ILE	0.176	0.003	0.022	-0.003	-0.009
ASP	-0.028	-0.008	-0.004	-0.002	0.002
LEU	0.295	0.039	0.049	0.069	0.035
HIS	-0.016	-0.007	-0.006	-0.009	-0.007
GLU	-0.030	-0.005	-0.005	-0.001	0.011
TOTAL KL	0.594	0.037	0.052	0.040	0.045

Table D.2: **KL deviations of cytoplasmic side helix positions from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates.** The “2-1” column lists the KL distances for the second position of the cytoplasmic side cap and the first position of the non-cytoplasmic side, and so on.



---

Amino acid	3 - 1	3 - 2	3 - 3	3 - 4	3 - 5
GLY	-0.021	-0.004	-0.036	-0.022	-0.033
PHE	0.098	-0.022	-0.016	-0.020	0.002
SER	-0.036	0.008	-0.018	-0.021	-0.011
PRO	-0.024	-0.019	-0.015	-0.007	0.002
TYR	-0.003	-0.018	-0.015	-0.015	-0.017
ARG	-0.036	0.010	0.016	0.020	0.014
TRP	0.033	0.001	-0.007	0.000	0.002
ALA	0.145	0.028	0.036	0.057	0.016
LYS	-0.032	0.015	0.028	0.017	0.024
ASN	-0.028	-0.010	-0.008	-0.007	0.003
GLN	-0.025	-0.003	-0.004	0.001	-0.008
CYS	-0.004	0.011	0.001	0.000	-0.001
MET	0.036	0.014	0.008	0.002	0.014
THR	-0.022	-0.002	-0.002	-0.009	0.005
VAL	0.114	-0.003	0.016	-0.012	-0.017
ILE	0.179	0.004	0.023	-0.002	-0.008
ASP	-0.028	-0.008	-0.004	-0.002	0.002
LEU	0.277	0.029	0.039	0.058	0.025
HIS	-0.016	-0.002	0.000	-0.005	-0.001
GLU	-0.035	0.002	0.002	0.008	0.025
TOTAL KL	0.572	0.031	0.044	0.040	0.040

Table D.3: **KL deviations of cytoplasmic side helix positions from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates.** The “3-1” column lists the KL distances for the third position of the cytoplasmic side cap and the first position of the non-cytoplasmic side, and so on.

---

Amino acid	4 - 1	4 - 2	4 - 3	4 - 4	4 - 5
GLY	-0.011	0.009	-0.027	-0.012	-0.024
PHE	0.109	-0.018	-0.011	-0.015	0.007
SER	-0.034	0.014	-0.013	-0.017	-0.007
PRO	-0.017	-0.011	-0.005	0.004	0.016
TYR	0.000	-0.016	-0.012	-0.012	-0.015
ARG	-0.036	0.010	0.016	0.020	0.014
TRP	0.006	-0.015	-0.021	-0.016	-0.015
ALA	0.095	-0.003	0.004	0.022	-0.012
LYS	-0.032	0.018	0.032	0.020	0.029
ASN	-0.028	-0.010	-0.008	-0.007	0.003
GLN	-0.025	-0.003	-0.004	0.001	-0.008
CYS	0.006	0.031	0.015	0.013	0.011
MET	0.039	0.017	0.010	0.004	0.017
THR	-0.019	0.002	0.002	-0.006	0.010
VAL	0.116	-0.002	0.017	-0.011	-0.016
ILE	0.227	0.027	0.049	0.020	0.013
ASP	-0.030	-0.006	-0.002	0.000	0.005
LEU	0.224	0.001	0.010	0.027	-0.002
HIS	-0.015	-0.001	0.001	-0.004	0.000
GLU	-0.030	-0.005	-0.005	-0.001	0.011
TOTAL KL	0.545	0.040	0.047	0.030	0.038

Table D.4: **KL deviations of cytoplasmic side helix positions from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates.** The “4-1” column lists the KL distances for the fourth position of the cytoplasmic side cap and the first position of the non-cytoplasmic side, and so on.

---

Amino acid	5 - 1	5 - 2	5 - 3	5 - 4	5 - 5
GLY	-0.007	0.014	-0.024	-0.008	-0.021
PHE	0.141	-0.003	0.005	0.000	0.026
SER	-0.014	0.054	0.015	0.010	0.024
PRO	-0.026	-0.022	-0.018	-0.011	-0.003
TYR	0.018	-0.001	0.003	0.003	0.000
ARG	-0.034	0.003	0.008	0.011	0.006
TRP	0.012	-0.012	-0.018	-0.013	-0.011
ALA	0.079	-0.012	-0.006	0.011	-0.021
LYS	-0.030	0.009	0.021	0.011	0.018
ASN	-0.028	-0.009	-0.006	-0.005	0.006
GLN	-0.025	0.001	0.000	0.006	-0.005
CYS	0.009	0.036	0.018	0.016	0.014
MET	0.027	0.008	0.002	-0.003	0.008
THR	-0.017	0.004	0.004	-0.004	0.013
VAL	0.132	0.007	0.027	-0.003	-0.008
ILE	0.230	0.028	0.050	0.022	0.014
ASP	-0.008	-0.005	-0.005	-0.004	-0.004
LEU	0.180	-0.021	-0.013	0.002	-0.024
HIS	-0.016	-0.007	-0.006	-0.009	-0.007
GLU	-0.030	-0.005	-0.005	-0.001	0.011
TOTAL KL	0.595	0.068	0.052	0.031	0.037

Table D.5: **KL deviations of cytoplasmic side helix positions from the non-cytoplasmic side helix cap positions in relative amino acid abundance rates.** The “5-1” column lists the KL distances for the fifth position of the cytoplasmic side cap and the first position of the non-cytoplasmic side, and so on.

---

Amino acid	1 - 2	1 - 3	1 - 4	1 - 5
GLY	0.017	0.007	-0.005	-0.009
PHE	-0.036	-0.040	-0.042	-0.049
SER	0.010	0.024	0.017	-0.012
PRO	0.005	0.010	-0.003	0.017
TYR	-0.025	-0.003	-0.006	-0.019
SEC	0.000	0.000	0.000	0.000
ARG	0.412	0.428	0.428	0.474
TRP	-0.018	-0.021	-0.013	-0.016
ALA	-0.051	-0.064	-0.051	-0.046
LYS	0.421	0.412	0.395	0.441
ASN	0.083	0.093	0.093	0.086
GLN	0.052	0.030	0.030	0.022
CYS	0.021	0.014	0.001	-0.001
MET	-0.027	-0.023	-0.024	-0.021
THR	0.009	0.011	0.006	0.003
VAL	-0.055	-0.051	-0.051	-0.055
ILE	-0.049	-0.050	-0.055	-0.055
ASP	0.153	0.153	0.141	0.314
LEU	-0.096	-0.094	-0.086	-0.078
HIS	0.048	0.029	0.027	0.048
GLU	0.080	0.056	0.080	0.080
TOTAL KL	0.953	0.921	0.880	1.122

Table D.6: **KL deviations of cytoplasmic side helix cap positions.** The “1-2” column lists the KL distances for the first position of the cytoplasmic side cap and the second position of the cytoplasmic side, and so on.

---

Amino acid	2 - 3	2 - 4	2 - 5	3 - 4	3 - 5	4 - 5
GLY	-0.008	-0.019	-0.022	-0.011	-0.015	-0.004
PHE	-0.006	-0.011	-0.025	-0.005	-0.020	-0.015
SER	0.012	0.006	-0.020	-0.005	-0.027	-0.024
PRO	0.005	-0.007	0.011	-0.011	0.006	0.021
TYR	0.035	0.030	0.009	-0.003	-0.017	-0.014
SEC	0.000	0.000	0.000	0.000	0.000	0.000
ARG	0.002	0.002	0.009	0.000	0.006	0.006
TRP	-0.007	0.014	0.007	0.024	0.016	-0.005
ALA	-0.026	0.000	0.010	0.031	0.043	0.010
LYS	-0.001	-0.003	0.002	-0.002	0.004	0.006
ASN	0.003	0.003	0.001	0.000	-0.002	-0.002
GLN	-0.007	-0.007	-0.009	0.000	-0.004	-0.004
CYS	-0.003	-0.009	-0.010	-0.008	-0.009	-0.002
MET	0.012	0.009	0.019	-0.002	0.006	0.008
THR	0.001	-0.003	-0.005	-0.004	-0.006	-0.002
VAL	0.011	0.010	0.001	-0.001	-0.010	-0.009
ILE	-0.001	-0.020	-0.021	-0.019	-0.020	-0.001
ASP	0.000	-0.002	0.027	-0.002	0.027	0.033
LEU	0.009	0.038	0.066	0.028	0.055	0.024
HIS	-0.006	-0.007	0.000	-0.001	0.010	0.011
GLU	-0.006	0.000	0.000	0.009	0.009	0.000
TOTAL KL	0.019	0.025	0.049	0.017	0.051	0.037

Table D.7: **KL deviations of cytoplasmic side helix cap positions.** The “2-3” column lists the KL distances for the second position of the cytoplasmic side cap and the third position of the cytoplasmic side, and so on.

---

Amino acid	1 - 2	1 - 3	1 - 4	1 - 5
GLY	0.022	-0.018	-0.001	-0.015
PHE	-0.051	-0.048	-0.049	-0.040
SER	0.078	0.033	0.027	0.044
PRO	0.008	0.016	0.029	0.046
TYR	-0.015	-0.012	-0.012	-0.015
SEC	0.000	0.000	0.000	0.000
ARG	0.160	0.178	0.192	0.171
TRP	-0.018	-0.023	-0.019	-0.017
ALA	-0.051	-0.048	-0.038	-0.056
LYS	0.171	0.219	0.178	0.207
ASN	0.054	0.061	0.063	0.094
GLN	0.065	0.062	0.077	0.050
CYS	0.020	0.007	0.005	0.004
MET	-0.012	-0.016	-0.019	-0.012
THR	0.027	0.027	0.017	0.038
VAL	-0.052	-0.043	-0.056	-0.058
ILE	-0.050	-0.044	-0.052	-0.053
ASP	0.153	0.180	0.193	0.226
LEU	-0.086	-0.082	-0.076	-0.087
HIS	0.028	0.033	0.023	0.031
GLU	0.172	0.172	0.197	0.277
TOTAL KL	0.622	0.652	0.681	0.834

Table D.8: **KL deviations of non-cytoplasmic side helix cap positions.** The “1-2” column lists the KL distances for the first position of the non-cytoplasmic side cap and the second position of the non-cytoplasmic side, and so on.

---

Amino acid	2 - 3	2 - 4	2 - 5	3 - 4	3 - 5	4 - 5
GLY	-0.033	-0.019	-0.030	0.020	0.004	-0.014
PHE	0.008	0.003	0.029	-0.004	0.020	0.025
SER	-0.023	-0.025	-0.017	-0.004	0.008	0.013
PRO	0.007	0.019	0.034	0.011	0.024	0.011
TYR	0.005	0.005	0.001	0.000	-0.003	-0.003
SEC	0.000	0.000	0.000	0.000	0.000	0.000
ARG	0.004	0.006	0.002	0.002	-0.001	-0.003
TRP	-0.008	-0.001	0.001	0.008	0.011	0.002
ALA	0.007	0.025	-0.010	0.018	-0.016	-0.029
LYS	0.007	0.001	0.005	-0.003	-0.001	0.004
ASN	0.003	0.005	0.020	0.001	0.015	0.013
GLN	-0.001	0.005	-0.006	0.006	-0.005	-0.009
CYS	-0.005	-0.005	-0.006	-0.001	-0.002	-0.001
MET	-0.005	-0.010	0.000	-0.005	0.006	0.012
THR	0.000	-0.007	0.008	-0.007	0.008	0.017
VAL	0.019	-0.010	-0.015	-0.025	-0.029	-0.005
ILE	0.019	-0.005	-0.012	-0.021	-0.027	-0.006
ASP	0.006	0.009	0.017	0.002	0.008	0.005
LEU	0.009	0.026	-0.003	0.016	-0.012	-0.026
HIS	0.002	-0.003	0.001	-0.005	-0.001	0.005
GLU	0.000	0.005	0.021	0.005	0.021	0.013
TOTAL KL	0.021	0.023	0.042	0.013	0.029	0.024

Table D.9: **KL deviations of non-cytoplasmic side helix cap positions.** The “2-3” column lists the KL distances for the second position of the non-cytoplasmic side cap and the third position of the non-cytoplasmic side, and so on.