



Research Department of Structural  
and Molecular Biology

Protocols to Capture the  
Functional Plasticity of Protein  
Domain Superfamilies

Robert Rentzsch

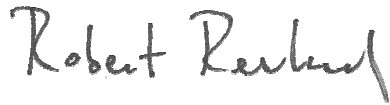
October 2011

SUBMITTED IN SUPPORT OF THE DEGREE OF

Doctor of Philosophy

# Declaration

I, Robert Rentzsch, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

A handwritten signature in black ink that reads "Robert Rentzsch". The script is cursive and somewhat stylized, with the first letters of the first and last names being capitalized and prominent.

Robert Rentzsch

October 2011

# Abstract

Most proteins comprise several domains, segments that are clearly discernable in protein structure and sequence. Over the last two decades, it has become increasingly clear that domains are often also functional modules that can be duplicated and recombined in the course of evolution. This gives rise to novel protein functions. Traditionally, protein domains are grouped into homologous domain superfamilies in resources such as SCOP and CATH. This is done primarily on the basis of similarities in their three-dimensional structures. A biologically sound subdivision of the domain superfamilies into families of sequences with conserved function has so far been missing. Such families form the ideal framework to study the evolutionary and functional plasticity of individual superfamilies. In the few existing resources that aim to classify domain families, a considerable amount of manual curation is involved. Whilst immensely valuable, the latter is inherently slow and expensive. It can thus impede large-scale application.

This work describes the development and application of a fully-automatic pipeline for identifying functional families within superfamilies of protein domains. This pipeline is built around a method for clustering large-scale sequence datasets in distributed computing environments. In addition, it implements two different protocols for identifying families on the basis of the clustering results: a supervised and an unsupervised protocol. These are used depending on whether or not high-quality protein function annotation data are associated with a given superfamily. The results attained for more than 1,500 domain superfamilies are discussed in both a qualitative and quantitative manner. The use of domain sequence data in conjunction with Gene Ontology protein function annotations and a set of rules and concepts to derive families is a novel approach to large-scale domain sequence classification. Importantly, the focus lies on domain, not whole-protein function.

## Acknowledgements

Thanks are due to several people that helped making my time in the Orengo lab and in London in general a valuable experience. From very early on until the now nearing end, my fellow PhD student and friend Jim Perkins played a big role in this – from quite practical issues, such as moving houses, sleeping in other people’s houses and playing pool in semi-accordance with the rules, to long and inspiring conversations about basically everything imaginable, including science. I apologise for having destroyed his belief in German punctuality, though. Benoit Dessailly was a great postdoctoral colleague and will hopefully remain a friend; his special sense of humour is certainly beaten by no other Belgian. Adam Reid and Ollie Redfern, who both left after my first year, were inspiring people to talk to, each in their own very special way. Thanks to Ian Sillitoe, Corin Yeats and Jon Lees for helping me to squeeze the relevant bits and bytes out of ‘their’ databases. Further, thanks to everyone else in the Orengo and Martin groups for the many random acts of kindness, most of which involved food.

I would like to particularly thank my supervisor, Christine Orengo, for her help along the way, reading manuscripts on the weekend or whilst (technically) being on holiday, and for giving me the opportunity to work in such a friendly and prestigious group. Thanks are also due to Kate Bowers, chair of my thesis committee, and David Jones, member of my thesis committee for guidance. Tom Knight, Jesse Oldershaw and Jahid Ahmed, known in the Darwin building as ‘the IT guys’, were of great help on several occasions. If I was Fox Mulder, these would be my personal Lone Gunmen. And I may well hire Tristan Clark from UCL Computer Science too, as he was incredibly helpful on short notice when it came to cluster troubles.

Back to the lower ranks, it should be noted here that I never met a fellow student in the Darwin building who would not have been friendly, helpful and

easy to get along with. Actually, I think this holds for everyone I have ever met there – apart from maybe this one person, but to be honest, who would not get a bit grumpy maintaining a building that poses unprecedented challenges to climate and pest control. Concerning the failure of the latter, maybe I am even glad that the mice have never really left. Their jumping around like mad in the bins was a comforting noise in the long, otherwise solitary nights that were spent at the office during ‘peak’ times.

It would be difficult to mention all the people and things outside work that made my time in London worthwhile. Yet, I will try to name a few important ones. I enjoyed every single fruit I ever bought on my way to and from work from the guy in front of the ‘Euston Supermarket’ – ten bananas for a pound! More importantly, he was always friendly and...well, there. Rain or shine. The same accounts for the Hare Krishna people who give out free food to students, homeless people and to everyone else who wants it. Again, value for (no) money paired with exceptional friendliness. Finally, there is the Poetry Library on the Southbank. This is probably the most serene and friendly place in London.

Last but not least, I want to thank the European Union and all its tax payers, for funding my research. As acknowledgements tend to be written at a point where the word ‘deadline’ becomes perceptible in all its nasty metaphorical power, and memory weakens, there will inevitably be people that should have been mentioned here but are not. To those I apologise. Please do not think you were less important or helpful than the banana man. One person I will certainly not forget to thank is Anna. It is never easy to live with someone who is constantly working towards a self-imposed goal, but, at the same time, it is great to know there will be someone to share the comfort with when the goal is achieved. To doing this with her, my close friends and my family, I look forward.

# Abbreviations

aaRS	Aminoacyl-tRNA-synthetase
AMP	Adenosine monophosphate
ATP	Adenosine triphosphate
BP	Biological Process (in GO)
CC	Cellular Component (in GO)
DAG	Diacyclic graph
DNA	Deoxyribonucleic acid
EM	Electron microscopy
GMP	Guanosine monophosphate
GO	Gene Ontology
GTP	Guanosine triphosphate
HIV	Human Immunodeficiency Virus
HMM	Hidden Markov model
HPC	High-performance computing
I/O	Input/Output
LUCA	Last Universal Common Ancestor (organism)
MF	Molecular Function (in GO)
NAD	Nicotinamide adenine dinucleotide
NADP	Nicotinamide adenine dinucleotide phosphate
NCBI	National Centre for Biotechnology Information
NMR	Nuclear magnetic resonance (spectroscopy)
ORF	Open Reading Frame
RAM	Random Access Memory
RMSD	Root Mean Square Deviation
RNA	Ribonucleic acid
UNIX	UNiplexed Information and Computing System

# Contents

<b>Chapter 1. Introduction .....</b>	<b>17</b>
1.1 Protein domains and superfamilies.....	18
1.1.1 Origin and definition of concepts .....	19
1.1.1.1 Superfamilies .....	20
1.1.1.2 Protein domains.....	21
1.1.2 The evolution of multi-domain proteins.....	22
1.1.3 Existing superfamily studies .....	23
1.2 Relationships between protein sequences.....	24
1.2.1 Pairwise relationships.....	25
1.2.1.1 Homology.....	25
1.2.1.2 Orthology and Paralogy.....	26
1.2.2 Group concepts.....	27
1.2.2.1 Superfamily.....	27
1.2.2.2 Orthologue cluster.....	28
1.2.2.3 Family.....	28
1.2.3 Application to proteins and domains .....	30
1.3 Protein function annotation .....	31
1.3.1 The Gene Ontology.....	32
1.3.2 Other systems .....	32
1.4 Bioinformatics methods.....	33
1.4.1 Sequence alignment.....	33
1.4.1.1 Pairwise sequence alignment.....	34
1.4.1.2 Multiple sequence alignment.....	37
1.4.2 Alignment profiles.....	38
1.4.2.1 Construction.....	38
1.4.2.2 Comparison.....	40
1.5 Bioinformatics resources.....	41
1.5.1 Primary sequence and structure databases.....	41
1.5.2 Protein classification resources.....	42

1.5.2.1	Classification based on structure .....	43
1.5.2.2	Classification based on sequence.....	45
1.6	Summary of work and overview .....	47
1.6.1	Summary of work.....	47
1.6.2	Overview of chapters.....	48

**Chapter 2. GeMMA: profile-based clustering of protein sequences in distributed computing environments..... 50**

2.1	Background and aims .....	50
2.1.1	Clustering biological sequences .....	51
2.1.2	Clustering algorithms.....	52
2.1.2.1	Hierarchical clustering.....	53
2.1.2.2	Partitional clustering.....	54
2.1.2.3	Graph-based clustering.....	56
2.1.2.4	Greedy incremental methods.....	58
2.1.3	Clustering evaluation measures .....	60
2.1.3.1	Unsupervised measures .....	61
2.1.3.2	Supervised measures .....	62
2.1.4	Existing tools and resources .....	63
2.2	Implementation.....	65
2.2.1	The GeMMA clustering protocol .....	65
2.2.2	Modular use of existing tools.....	66
2.2.3	The GeMMA heuristics.....	68
2.2.3.1	Greedy merging .....	71
2.2.3.2	Comparison sampling .....	72
2.3	Discussion.....	77
2.3.1	Notes on performance and its measurement.....	77
2.3.2	Future work .....	78
2.3.2.1	Changes in the use of third-party tools.....	78
2.3.2.2	Further technical integration.....	79
2.3.2.3	Changes to the GeMMA heuristics.....	81



2.3.2.4	Changes to the protocol as a whole .....	84
2.3.2.5	Potential use with other types of data.....	85

### **Chapter 3. The DFX pipeline: identification of functional families**

#### **within protein domain superfamilies ..... 86**

3.1	Background.....	88
3.1.1	Existing family resources.....	88
3.1.2	Automatic methods and protocols .....	93
3.2	Concepts .....	94
3.2.1	The domain to function relationship.....	95
3.3	Implementation.....	97
3.3.1	The DFX pipeline .....	97
3.3.1.1	Technical implementation .....	100
3.3.2	Input data preparation.....	101
3.3.3	Sequence clustering.....	103
3.3.3.1	Pre-clustering .....	103
3.3.3.2	Hierarchical clustering.....	104
3.3.4	The two family identification protocols.....	105
3.3.5	Family naming and taxonomic characterisation.....	109
3.3.6	Model library generation and family assignment.....	113
3.3.7	Function assignment to whole-protein sequences.....	115
3.4	Discussion and future work.....	120
3.4.1	The use of sequence data .....	120
3.4.2	The two family identification protocols.....	121
3.4.3	The family naming protocol .....	122
3.4.4	The overall architecture of DFX.....	123

### **Chapter 4. Unsupervised protein domain family identification in DFX**

#### **..... 124**

4.1	Background.....	125
4.1.1	Preliminary remarks .....	125
4.1.2	Existing unsupervised family identification methods.....	126

4.1.2.1	Integrated methods .....	126
4.1.2.2	Combined protocols .....	128
4.2	Implementation.....	130
4.2.1	The gold standard and derived datasets .....	131
4.2.1.1	The gold standard dataset.....	131
4.2.1.2	Two derived datasets.....	133
4.2.2	Performance measures.....	134
4.2.3	Derivation of generic clustering granularity settings .....	137
4.2.4	Benchmarking.....	138
4.2.4.1	High quality benchmark .....	138
4.2.4.2	Large-scale benchmark .....	139
4.3	Results and Discussion.....	140
4.3.1	Derivation of generic clustering granularity settings .....	140
4.3.2	Analysis of entire Gene3D domain superfamilies.....	145
4.3.3	Benchmarking.....	148
4.3.3.1	High-quality benchmark .....	148
4.3.3.2	Large-scale benchmark .....	151
4.4	Conclusions and future work .....	154
4.4.1	Recent use cases .....	155
4.4.2	Future work .....	156
<b>Chapter 5. Supervised protein domain family identification in DFX ..</b>		<b>158</b>
5.1	Background.....	159
5.1.1	Existing supervised family identification methods .....	159
5.1.2	Existing methods to derive domain-specific annotations.....	160
5.1.3	Protein domain function and the Gene Ontology.....	165
5.1.4	Protein families with functionally conserved domains.....	170
5.1.4.1	The P-loop type ATPase family.....	170
5.1.4.2	The class I aaRS family .....	175
5.2	Concepts .....	177
5.2.1	Sequence and cluster annotation using sets .....	178

5.2.2	The core annotation of domain sequence clusters .....	179
5.2.3	Chaining in the clustering of annotated sequences.....	181
5.3	Implementation.....	185
5.3.1	Overview of the protocol.....	185
5.3.2	Identification of the cluster core annotation .....	188
5.3.3	Detection and removal of non-core annotations.....	191
5.3.4	Assessment of cluster functional coherence.....	193
5.3.5	Detection of cluster chaining.....	195
5.4	Results and Discussion.....	200
5.4.1	Domain function captured in selected domain families.....	200
5.4.1.1	P-ATPase catalytic domains in the HAD superfamily.....	201
5.4.1.2	Class I aaRS catalytic domains in the HUP superfamily....	206
5.4.2	The sequence footprint of domain function conservation .....	215
5.4.3	Examples of annotation complexity and inconsistency .....	221
5.5	Conclusions and future work .....	227
5.5.1	Uniqueness and aim of the developed protocol.....	228
5.5.2	The limits of rule-based heuristics .....	229
5.5.3	Potential improvements to data usage.....	232
5.5.4	The future of the chaining concept .....	236
5.5.5	Proposed further analyses .....	238

**Chapter 6. Quantitative analysis of the DFX results and comparison of the two family identification protocols.....242**

6.1	Results .....	242
6.1.1	Statistics on the produced families.....	243
6.1.2	The scale-free size distribution of families and superfamilies ..	247
6.1.3	The largest families in the largest superfamilies .....	249
6.1.4	Comparison with the unsupervised protocol .....	252
6.1.5	Fairness of the comparison.....	258
6.2	Discussion.....	265
6.2.1	Notes on performance assessment .....	265

6.2.2	The significance of family size distributions.....	267
<b>Chapter 7. The DFX pipeline: summary and future work.....</b>		<b>269</b>
7.1	Summary of work .....	269
7.2	Current usage and data availability.....	271
7.2.1	The Gene3D family level .....	272
7.2.2	DFX in protein function prediction .....	272
7.2.3	DFX in the detailed study of superfamilies .....	276
7.3	Recent improvements and future work .....	278
7.3.1	The chaining concept and a potential two-layer system.....	278
7.3.2	Potential replacement of GeMMA.....	280
7.3.3	Potential replacement of the unsupervised protocol.....	281
7.3.4	Further potential improvements .....	282
7.4	DFX in the context of other novel methods .....	282
7.5	Final remarks .....	284
<b>Appendix A – HPC implementation of GeMMA.....</b>		<b>308</b>
A.1	Challenges.....	309
A.2	The pairs matrix.....	312
A.3	The results matrix .....	313
A.4	Resource utilisation .....	315
A.5	Job monitoring and rescue .....	316
<b>Appendix B – Superfamily studies 1990-2010 .....</b>		<b>318</b>

# Figures

Figure 2.1. Hierarchical agglomerative clustering.....	55
Figure 2.2. The GeMMA workflow .....	66
Figure 2.3. The GeMMA heuristics .....	69
Figure 2.4. Potential effects of the GeMMA heuristics.....	70
Figure 3.1. The workflow of the DFX pipeline. ....	98
Figure 3.2. Comparison of the unsupervised and supervised family identification protocols by example.....	107
Figure 3.3. The DFX family naming protocol.....	112
Figure 3.4. Generation and use of the family model library .....	116
Figure 3.5. Probabilistic GO term assignment based on a single query domain .....	119
Figure 4.1. Agreement of the partitionings produced by GeMMA clustering with known functional families in the SFLD and SFLD-Gene3D superfamilies .....	144
Figure 4.2. Agreement of the best partitionings produced by GeMMA clustering with known functional families in the three SFLD-derived datasets .....	147
Figure 4.3. Agreement of the family partitionings produced by $DFX_{\text{unsuper}}$ and SCI-PHY with known functional families in cross-validation benchmarking on the SFLD dataset.....	151
Figure 4.4. Performance of $DFX_{\text{unsuper}}$ and SCI-PHY in the Pfam benchmark .....	153
Figure 4.5. Functional conservation in the Pfam benchmark families and the produced SCI-PHY and $DFX_{\text{unsuper}}$ families .....	154
Figure 4.6. Transfer of functional annotations within the Pfam benchmark families.....	154
Figure 5.1. The hydrolytic deamination reaction catalysed by Glucosamine-6- phosphate deaminase.....	167

Figure 5.2. The four structural domains of P-type ATPase transport proteins .....	173
Figure 5.3. The two structural domains of class I aaRS proteins.....	176
Figure 5.4. The concept of cluster chaining.....	182
Figure 5.5. Annotation editing in the supervised family identification protocol .....	187
Figure 5.6. Two example domain sequence clusters and the associated protein function annotationsmilies according to the DFX family concept.....	190
Figure 5.7. The rules followed by the chain elongation algorithm.....	198
Figure 5.8. The clustering dendrogram of the P-ATPase P domain family...203	
Figure 5.9. Families of aaRS catalytic domains in the family dendrogram of the HUP superfamily.....	208
Figure 5.10. The clustering dendrograms of two aaRS catalytic domain families.....	214
Figure 5.11. The Gene3D domain architectures of nine homologous P-ATPase proteins .....	218
Figure 5.12. High sequence conservation in functionally equivalent domains from functionally divergent multi-domain proteins.....	219
Figure 5.13. Annotation complexity in the P-ATPase P domain family as identified by $DFX_{super}$ .....	224
Figure 5.14. The three transferase functions corresponding to the three domains of Trifunctional protein ribF/mnmA.....	227
Figure 5.15. Inconsistently linked NAD <sup>+</sup> synthase activities in the GO MF DAG .....	241
Figure 6.1. The DFX families identified in 1,793 Gene3D domain superfamilies of varying size and sequence diversity .....	245
Figure 6.2. The scale-free size distribution of domain superfamilies and their DFX families.....	248
Figure 6.3. The family size distribution of the ‘Winged Helix DNA-binding’ domain superfamily.....	249

Figure 6.4. The relative performance of DFX <sub>super</sub> and DFX <sub>unsuper</sub> as measured by enzyme function conservation .....	254
Figure 6.5. The relative performance of DFX <sub>super</sub> and DFX <sub>unsuper</sub> with a generic granularity setting of 10 <sup>-40</sup> .....	255
Figure 6.6. The impact of superfamily size, sequence diversity and functional diversity on the performance of the DFX family identification protocols....	264
Figure 7.1. The coverage of the HUP domain superfamily with EC functional family assignments before and after scanning with family-specific models...	275

# Tables

Table 2.1. The number of comparisons made in the first iteration of GeMMA clustering depending on the size of the input dataset.....	74
Table 4.1. The SFLD protein dataset and its mapping to Gene3D.....	133
Table 4.2. Peak family partitioning performance when clustering the superfamilies in the three SFLD-derived datasets with GeMMA.....	146
Table 4.3. Performance of $DFX_{\text{unsuper}}$ and SCI-PHY in cross-validation benchmarking on the SFLD dataset .....	149
Table 4.4. Size of the family partitionings produced by $DFX_{\text{unsuper}}$ and SCI-PHY in cross-validation benchmarking on the SFLD dataset .....	150
Table 5.1. The three subclasses of class I aaRS proteins.....	177
Table 6.1. The ten largest Gene3D superfamilies and their diversity in sequence, structure and function.....	246
Table 6.2. The families identified in the ten largest Gene3D superfamilies ..	251
Table 6.3. The average performance of $DFX_{\text{super}}$ and $DFX_{\text{unsuper}}$ .....	256
Table 6.4. The $DFX_{\text{super}}$ and $DFX_{\text{unsuper}}$ partitionings of five functionally diverse enzyme domain superfamilies .....	257
Table 6.5. The impact of edit distance normalisation on the calculation of overall family identification performance .....	260
Table 6.6. The effect of a modified normalisation procedure on the calculation of overall family identification performance.....	262
Table 6.7. Correspondence of alternative performance measures with the performance scores attained when using the modified normalisation procedure.....	263



# Chapter 1. Introduction

The superfamily and the family are the most commonly used frameworks to study the evolution of protein and protein domain function. This is demonstrated by the non-exhaustive list of 80 studies and reviews from the past two decades that use these concepts, in Appendix B. Further, in the words of Monica Riley, who created the first protein function ontology (Riley 1993), ‘...a useful step would be to expand databases to provide explicit information on domain function’ (Riley 2007). This remark was made with regards to genome annotation, and certainly holds for the study of the evolution of protein function. In fact, Riley’s article was to introduce one of the seminal studies on the domain-based evolution of proteins (Bashton and Chothia 2007).

The overarching aim of the presented work was the development of a software pipeline (and the underlying algorithms) to identify the functionally conserved families within protein domain superfamilies. Such families have many potential uses; most importantly, they can help study the evolution of protein function on the domain level. As a first important part of this, an efficient yet sensitive sequence clustering method for use in HPC environments was to be developed, with potential applications in other large-scale clustering tasks. The second challenge was to integrate, first, the clustered domain sequence data and, subsequently, the available high-quality protein annotation data to establish a family level below the domain superfamily. Further, to do this on a large scale, all the necessary steps had to be implemented in a high-throughput pipeline for processing thousands of superfamilies of highly varying size and sequence diversity. Finally, the results of this endeavour had to be analysed both qualitatively, through the detailed analysis and discussion of examples, and quantitatively, using statistics and benchmarking.

Specific attention is paid in this work to define all theoretical concepts introduced and used as clearly as possible. The lack of such definitions is a frequent complication for understanding related studies, and sometimes even a hindrance of progress. The latter is not always easy to trace objectively in bioinformatics. This makes it even more important to establish the most important precondition for progress: a ‘common ground’ on which research is conducted, that is, clearly defined terms and concepts. A striking example of how difficult this seems to be is the notion that even articles in high-profile journals still routinely talk about ‘high sequence homology’ (see, for example: Mair, Braks et al. 2006; Hang, Yang et al. 2010; Salmena, Poliseno et al. 2011), after decades of urging researchers, sometimes in the same journals, not to do so (Lewin 1987; Reeck, de Haen et al. 1987; Marabotti and Facchiano 2009; Marabotti and Facchiano 2010).

Several theoretical concepts for grouping protein and protein domain sequences are of crucial importance to the present work, just as the notions of protein and protein domain function. As these concepts are so widely used but, at the same time, so seldom defined or discussed, they are specifically addressed in the following sections (and chapters). The second part of this general introduction then describes several key bioinformatics concepts, methods and resources that are relevant to this thesis as a whole. This chapter concludes with a summary of the work conducted and provides an overview of all the following chapters.

## **1.1 Protein domains and superfamilies**

Two concepts build the ‘theoretical backbone’ of the developments and studies described in this thesis: the protein domain and the protein (domain) superfamily. Both are widely used in the fields of sequence and structural biology, and, in conjunction with these, especially in the bioinformatics area.

The following sections trace the roots of both concepts and underline their specific importance to the present work.

### 1.1.1 Origin and definition of concepts

The most important, shared aim of research in the above-mentioned areas is to capture the intricate movements of evolution on the molecular level. This observation supports Dobzhansky's famous essay title, 'Nothing makes sense except in the light of evolution' (Dobzhansky 1973), which has become a catch-phrase of evolutionary research: as of September 2011, it yields more than 35,000 Google hits. Further, related aims are better to understand the functional machinery of individual cells and, eventually, the resulting macroscopic phenotypes. Therefore, it is no coincidence that both the domain and superfamily concepts were first explicitly introduced in the early 1970s, at the origin of modern-day, computer-aided evolutionary biology (which paraphrases 'bioinformatics').

It was in the 1970s when X-ray crystallography became a widely-used technology, the first phylogenetic studies on sets of evolutionarily related sequences appeared, and the first steps towards developing efficient structure and sequence comparison algorithms were taken (as reviewed in Ouzounis and Valencia [2003]). Within the course of the same year, Walter Wetlaufer and Margaret Dayhoff introduced the protein 'domain' (Wetlaufer 1973) and 'superfamily' (Dayhoff 1974) terms, respectively. Since then it has become increasingly clear that the synthesis of the two concepts, the protein domain superfamily, is the most appropriate framework for conducting studies on the (long-term) evolution of protein sequence and structure and the resulting (long-distance) homology relationships between proteins. This framework is immediately understood when the ancestral concepts are clearly defined, as follows.

### 1.1.1.1 Superfamilies

The superfamily concept was introduced by the Dayhoff group based on their efforts to classify evolutionarily related proteins from the 1960s onwards. These eventually resulted in the Protein Information Resource (PIR) (Barker, George et al. 1993; Nikolskaya, Arighi et al. 2006), which exists to the present day. Defined in a both evolutionary and pragmatic manner, the term referred to a group of monophyletic protein families that can only be established using methods for remote homology detection, in contrast to the families themselves; in principle, this definition is still (implicitly) used today.

Dayhoff's classification of protein sequences did not yet take into account the existence of domains, and was still largely based on establishing similarities in sequence. Only by the mid 1990s, when the domain concept had been widely established and structure-based domain classification resources such as SCOP (Murzin, Brenner et al. 1995) and CATH (Orengo, Michie et al. 1997) emerged, PIR and other protein (super)family resources started to implicitly incorporate the domain concept. For example, since that time protein superfamilies are required to be 'homeomorphous' in PIR, that is, they must share the same domain architecture (Barker, George et al. 1993).

Following the original (protein) superfamily concept, both SCOP and CATH establish homologous domain superfamilies based on similarities in sequence, structure and function. In this, at least two of the latter types of similarity must be discernable to group two domain sequences into the same superfamily (see Section 1.5.2.1). This can be reformulated as the concept of exhaustively grouping sequences by homology relationships, including cases of very weakly detectable (i.e., remote) homology (see Section 1.2.1.1); this superfamily concept is followed in the present work.

### 1.1.1.2 Protein domains

According to Wetlaufer's original definition, protein domains are compact structural units that can fold independently, and therefore 'nucleate' the folding of whole-protein chains. Notably, based on observations in only 18 protein structures, he also already included the possibility of such domains being discontinuous in sequence (Wetlaufer 1973). Apart from the strict constraint of independent folding, which may apply in many cases but can (so far) hardly be assessed on a large scale, this is the principal definition that is still followed by extant domain (super)family resources such as Pfam (Finn, Mistry et al. 2010), SCOP and CATH. It should be noted that other researchers, most famously Michael Rossmann, shared Wetlaufer's discovery of the protein domain (Rossmann, Moras et al. 1974).

Even given that the tertiary structures of related proteins and, therefore, protein domains are usually much more similar (evolutionarily conserved) than the underlying sequences (Chothia and Lesk 1986; Illergard, Ardell et al. 2009), structural domains can normally be detected on the sequence level (Doolittle 1995; Koonin, Wolf et al. 2002); this is also illustrated by the existence and *modus operandi* of resources such as SUPERFAMILY (Gough, Karplus et al. 2001) and Gene3D (Buchan, Shepherd et al. 2002) (see Section 1.5.2.1). Different and often overlapping definitions of the domain concept have been proposed since Wetlaufer's times, for example, those of the folding unit, the structural unit, the evolutionary unit, or the functional unit (Yeats and Orengo 2001). A 'dual' definition of the protein domain appears to represent the broad consensus and is, therefore, used in the present work: a continuous or discontinuous region that is conserved in both structure and sequence among related proteins, where the sequence signal may be weak.

### 1.1.2 The evolution of multi-domain proteins

Most proteins contain more than one domain. It has been estimated that this accounts for more than half of all prokaryotic and eighty percent of all eukaryotic proteins (Apic, Gough et al. 2001). More conservative estimates lie in the range of forty percent for prokaryotes and sixty to seventy percent for eukaryotes (Ekman, Bjorklund et al. 2005). While such estimates are highly method-dependent, an increase in the abundance and complexity of domain architectures in the eukaryotic lineage is obvious. Protein domains usually fall into a size range of 100 to 250 residues (Islam, Luo et al. 1995; Chothia, Gough et al. 2003; Ekman, Bjorklund et al. 2005; Wang, Kurland et al. 2011), with the average number of residues varying depending on the methodology used (for example, whether looking at only domains with known structure or at the much higher number of domains assigned on a sequence basis). Considerably smaller and larger outliers exist. The number of domains per protein has been shown to assume a power law distribution (Koonin, Wolf et al. 2002), and only few proteins have more than three domains.

Some domain types have been shown to be particularly ‘promiscuous’ (Marcotte, Pellegrini et al. 1999), with regards to their occurrence in many different domain architectures. These usually correspond to evolutionarily ancient and widespread domain superfamilies that fulfil basic, partial protein functions. Examples are domains that bind ubiquitous cofactors such as ATP and NAD(P) or such that serve as a general ‘linker’ between proteins, thereby enabling protein interactions. Promiscuous domains further play a major role in the extensive signalling networks of metazoan species; examples are the SH2, SH3 and PDZ domain types (Pawson and Nash 2003).

The evolutionary events that give rise to the different domain architectures observed in proteins in general, and the frequent reuse of (promiscuous) domains in particular, are still not entirely understood. Since the pioneering

works of Walter Gilbert (Gilbert 1978) on the mobility of exons, increasing evidence has accumulated for the hypothesis that, domains can be ‘shuffled’ between eukaryotic genes (proteins) (Patthy 1999; Chothia, Gough et al. 2003). The concept of ‘exon shuffling’, a practically proven phenomenon (Doolittle 1995; Patthy 1999; Liu and Grigoriev 2004; King, Westbrook et al. 2008; Basu, Poliakov et al. 2009) is thought to play a major role in this, among other factors. This refers to the insertion of an exonic region from one gene into an intronic region of another gene (Patthy 1999), probably mediated by unequal crossing-over during meiosis (exon duplication) in conjunction with transposable elements (exon shuffling). The recipient gene (protein) in this manner gains one or more additional domains.

The different extents to which exon shuffling, gene fusion and fission and further types of non-homologous genetic recombination occur, and their relative impacts on the recombination of domains, is still subject to considerable study and debate. A comprehensive review of both is provided in Nagy and Patthy (2011), where it is also claimed that internal domain insertion and deletion events (such not occurring at the termini of proteins) are a frequent phenomenon in metazoan evolution, as opposed to earlier studies.

### 1.1.3 Existing superfamily studies

Notwithstanding recent speculations on the continuity of fold space (Shindyalov and Bourne 2000; Grishin 2001; Kolodny, Petrey et al. 2006; Taylor 2007; Cuff, Redfern et al. 2009), and on whether protein classification is necessary at all (Petrey and Honig 2009), the superfamily concept has had, and continues to have, a tremendous impact on how both computational and experimental biologists study the evolution of protein sequence, structure and function. A sense of this impact is conveyed by the selection of publications

on protein and protein domain superfamilies compiled in Appendix B, coming from both wet-lab experimental and bioinformatics groups.

Studies on individual superfamilies usually fall into one of three categories: (i) general reviews and/or classification efforts, (ii) those that report the identification of one or several novel subgroups and (iii) those that characterise one or more novel member sequences. While physical experiments usually play a major role in the last case, a core set of bioinformatics concepts and methods is shared by almost all of these publications. In particular, these are the generation of multiple sequence alignments, the construction of phylogenetic trees from the former and the modelling of protein structure, where template structures are available. Sequence similarity networks are increasingly used too (Song, Joseph et al. 2008; Atkinson, Morris et al. 2009), as a fourth, powerful visualisation method. Most importantly in the context of the present work, any already established knowledge on (or classification of) functional families within the studied superfamily is mapped onto these alignments, trees, structures and networks.

Only when all that is already known about the functions and functional group relationships within a superfamily is put into context, the unexplored sequence space and the gaps in the established knowledge become apparent. This is illustrated by the way in which different studies and reviews on individual superfamilies build on their predecessors. A ‘functional skeleton’ of the superfamily is almost always available, and it is the unknown parts that need to be fleshed out. In the course of this, classification systems for the identified subgroups are frequently developed, extended and sometimes abolished and replaced.

## **1.2 Relationships between protein sequences**

At the beginning of any effort to group protein sequences stands the identification of pairwise relationships. This information can then be used to



establish wider groups of sequences, following different grouping concepts. The pair- and group-wise relationships between protein sequences that are most widely used for classification are introduced in the following sections; all are based on evolutionary considerations.

### 1.2.1 Pairwise relationships

There exist three basic concepts that are used to describe pairwise evolutionary relationships between proteins. These are the notions of homology, orthology and paralogy. Both orthology and paralogy imply homology, which is therefore discussed and defined first in the following.

#### 1.2.1.1 Homology

Both Richard Owen's original homology concept that is used to compare common anatomical traits of related species (Rupke 1993) and the homology concept that is used in molecular biology today share the core of their definition, namely that 'homology' (from ancient Greek *ομολογεω*, 'to agree') refers to 'possessing a common evolutionary origin' (Reeck, de Haen et al. 1987). Note that homology therefore must not necessarily imply readily observable similarity. In this strict sense, and when assuming a single last common ancestor sequence at the origin of the DNA world, for the sake of the argument, all extant DNA (protein) sequences would be homologous.

To be a useful concept, sequence homology must be defined with additional constraints. These can be derived in different ways, but usually include the notion of observable similarity. The probably most straightforward and logical way, with the use of fossils in evolutionary biology in mind, is via the (probabilistic) reconstruction of ancestral sequences (Fitch 1971). This particularly accounts for distinguishing homologous from analogous sequences, which have independently evolved to a similar fold and/or function. In brief, if two sequences are compared and their ancestral

sequences are more similar than the sequences themselves, this is a sign of homology; if the opposite is true, analogy is suggested. This useful definition of (or test for) sequence homology implies a certain amount of similarity between the compared sequences, as a reconstruction of ancestral character states with reasonable confidence is otherwise impossible.

William Pearson, co-developer of the FASTA algorithm for sequence alignment (Lipman and Pearson 1985), advocates a pragmatic definition of sequence homology in his talks and publications (Pearson and Sierk 2005; Lavelle and Pearson 2009). In brief, this says that establishing homology between two proteins requires statistically significant (non-random) sequence and/or structural similarity, that is, ‘excess similarity’ (Doolittle 1981; Pearson and Sierk 2005). Tools such as BLAST (Altschul, Gish et al. 1990) (see Section 1.4.1.1) and FASTA assess this criterion for protein sequences, and structural comparison tools (Hasegawa and Holm 2009) do the same for structures. Further, according to Pearson, statistically significant sequence similarity always implies structural similarity, whereas the opposite is not true. As this definition of homology is the most commonly (if not always explicitly) used and in line with the definition of homologous domain superfamilies in resources such as SCOP and CATH (see Section 1.5.2.1), it will be followed in the present work.

### 1.2.1.2 Orthology and Paralogy

The definitions of orthology and paralogy that are used in sequence biology today were given by Walter Fitch (Fitch 1970; Fitch 2000). According to this, two homologous genes (proteins) should be called orthologous if their last common ancestor sequence was duplicated in a speciation event (leading to two copies in different genomes) and paralogous if it was duplicated in a gene duplication event (leading to two copies within the same genome).

By definition, paralogues can occur in both the same genome and different genomes, in contrast to orthologues. Better to distinguish between the two cases, the terms ‘inparalogue’ (duplication after the last speciation event) and ‘outparalogue’ (duplication prior to the last speciation event) were introduced later on (Sonnhammer and Koonin 2002). Further, the term ‘xenology’ is sometimes used to account for events of horizontal gene transfer (Fitch 1970), the lateral exchange of genes (proteins) between species in the taxonomic tree.

Based on the assumption that one of the two copies of a duplicated gene is subject to reduced selection pressure (Ohno 1970), as the other copy retains the original (protein) function, it is commonly assumed that orthologous and inparalogous proteins are, on average, functionally more conserved than outparalogous proteins.

### 1.2.2 Group concepts

Two out of three commonly used concepts to partition protein sequence space into groups of sequences are defined relatively clearly, with these definitions being commonly accepted among researchers in the field and not varying considerably between different resources. This is the sequence superfamily on the one hand and the orthologue cluster on the other hand. The following sections first briefly outline these two concepts and then put them into context with the less clearly defined family concept, which is discussed in more detail. For simplicity, it is assumed that only proteins that share exactly the same domain architecture are grouped. Section 1.2.3 then discusses in how far each concept can be used on the protein and domain levels, respectively.

#### 1.2.2.1 Superfamily

At the superfamily level, which lies below the fold level, sequences are grouped based on common ancestry: all members must be homologous. This

can be thought of as the most inclusive, or ‘loose’, criterion that can be applied and reliably tested for (see Section 1.2.1.1). Similarities of sequences beyond this level are (and should) be very difficult to detect with significant reliability. The latter can be expected when different superfamilies share the same fold. Overall, the superfamily classification level is most useful in studying the evolution of protein structure. As protein domains are believed to fold relatively independently from the rest of the protein chain, and can be independently rearranged and reused throughout evolution (see Sections 1.1.1.2 and 1.1.2), superfamily resources often classify domain sequences, not whole proteins.

#### 1.2.2.2 Orthologue cluster

At the other end of the partition granularity scale stands the concept of orthologue clusters. Here, the goal is to group only very closely related sequences, linked by either orthology or inparalogy. The latter types of relationships often imply equivalence in function. They can be established by pair- or group-wise sequence comparisons in combination with either subsequent experiments that reveal functional identity or algorithms that evaluate the comparison results; of course, both can also be combined. By definition, orthologue<sup>1</sup> resources cluster whole proteins, not protein domains. They are mostly used to infer species trees in molecular phylogeny, to study how specific protein functions are encoded and conserved, and in protein function prediction (as reviewed in Li, Stoeckert et al. [2003]).

#### 1.2.2.3 Family

Many different monophyletic groups become apparent when studying sequence-based phylogenetic trees of large superfamilies (Iyer, Anantharaman et al. 2003; Zelensky and Greedy 2005; Yang and Bourne 2009). Above the

---

<sup>1</sup> For ease of reading, both orthology and inparalogy will be implied when orthologue clusters are referred to in the following.

level of orthologue clusters, there usually exist a wide range of non-random partitions in superfamily sequence space, as this is shaped by evolutionary processes. However, these partitions are hard to delineate, both manually and algorithmically. This is because, they can differ with each superfamily, depending on evolutionary speed, superfamily size, age, and so forth. In principal, the different sequence groups observed at all levels of the tree, that is, between the root node and the individual orthologue clusters, are all candidates for the family (or ‘subfamily’<sup>2</sup>) level. This is why the family concept is particularly problematic: there is no clear definition *per se* what the clustering criterion to establish such families would be.

By definition, all sequences in a superfamily, and therefore in any group of sequences it subsumes, are structurally highly similar. This is not always true for function, as relatively small modifications in sequence and structure can be sufficient to alter it (Seffernick, de Souza et al. 2001; Almonacid, Yera et al. 2011). While superfamilies can thus be functionally diverse, several studies have illustrated that the basic reaction mechanism is usually conserved (Babbitt, Hasson et al. 1996; Aravind, Leipe et al. 1998; Burroughs, Allen et al. 2006). On the other end of the scale, close to the leafs of the superfamily tree, it can often be observed that different orthologous clusters exhibit considerable functional similarity. Such groups mix orthologues and paralogues, prominent examples being the large metazoan multi-gene families. Interestingly, two recent studies challenge the established view after which orthologues are generally more conserved in function than paralogues altogether (Studer and Robinson-Rechavi 2009; Nehrt, Clark et al. 2011).

It follows from the above that functional change below the superfamily and (closely) above the orthologue cluster level is usually gradual. This can refer, for example, to different substrate specificities and/or reaction rates in enzymes, different ligand binding characteristics in receptors, different solute

---

<sup>2</sup> The terms family and subfamily are often used interchangeably to describe a grouping of sequences below the superfamily level.

affinities and/or flux rates in channel proteins, and so forth. When protein families are, therefore, defined in a way that leaves room for a certain degree of variability in function, this makes them a suitable framework to study (the evolutionary processes that govern) functional change. For example, multiple sequence and structure alignments of such families can highlight changes in key residues and in the orientation of catalytic side-chains, respectively, and both can help to explain changes in protein function.

### 1.2.3 Application to proteins and domains

The concepts described in the above two sections can be applied to whole-protein and protein domain sequences to different extents, since evolutionary events can occur asynchronously on the two levels. Specifically, this refers to gene duplication and speciation events on the one hand and to domain gain, loss and shuffling events on the other hand. It is particularly obvious for promiscuous domains (see Section 1.1.2), which appear in different proteins with widely varying domain architectures, functions and evolutionary backgrounds. These domains can encode conserved (partial) protein functions, even in cases where the corresponding parent proteins are not homologues.

Homology, based on its definition in Section 1.2.1.1 (statistically significant similarity), can be established or rejected for evolving sequences in general, not only proteins (Koonin 2005). It is therefore straightforward to use the superfamily concept on the domain level; in fact, it usually applies to individual domains only. This is why studies on specific ‘protein superfamilies’ often actually deal with a single core domain (set of domains) that is shared by all member sequences, and then analyse the combination of this protein core with additional domains throughout evolution. The family concept can equally well be used for domains and (multi-domain) proteins, in the latter case implying conserved domain architecture. The concepts of orthology and paralogy, however, are inherently tied to species phylogeny (specifically, gene

duplication and speciation events) and can, therefore, not be applied in a consistent manner to protein domains.

It could be argued that two domain sequences of the same type that are found in different proteins can be called orthologous if the proteins themselves are orthologues. However, this concept would be of limited use in studying the evolution of sequence, structure and function on the domain level. Orthologous domain clusters in this sense would be (artificially) confined to the size of the corresponding protein orthologue clusters. These considerations equally apply for the concept of paralogy. However, there is an ongoing conceptual debate as to whether both concepts could be consistently used (or even ‘recoined’) for the protein domain level (Koonin 2005; Song, Sedgewick et al. 2007; Song, Joseph et al. 2008). In this case, different parts of a multi-domain protein that has acquired at least one of its domains by means of domain shuffling (see Section 1.1.2) would have to be described with the respective terms independently.

### **1.3 Protein function annotation**

The notion of ‘protein function’ is multi-faceted (Rentzsch and Orengo 2009), but three general aspects can be distinguished. Traditionally, protein function refers to the molecular function of a sequence, such as the catalytic activity of enzymes, the scaffolding activity of structural proteins, the transport and signalling activities of transmembrane proteins, and so forth. This ‘narrow’ aspect of function is solely determined by sequence and structure. In contrast, protein function in a ‘broad’, contextual sense describes the activities of proteins in the context of cellular pathways and processes. A third type of information that is ancillary to functional information is location, that is, *where* a certain molecular function (a certain process) is carried out (takes place) in- or outside a cell.

### 1.3.1 The Gene Ontology

In the Gene Ontology (GO) annotation system (Ashburner, Ball et al. 2000), each of the three general aspects of protein function is represented by a tree-like structure, formally a directed acyclic graph (DAG). In these trees, terms that describe specific activities are found close to the leaf nodes, whereas the root nodes are the most unspecific: ‘molecular function’ (MF), ‘biological process’ (BP) and ‘cellular component’ (CC). The terms in each DAG are connected in a bottom-up manner, by child-parent (‘is a’) relationships, and each term can have multiple parent terms.

The different terms in each of the three GO DAGs are arranged hierarchically, by the degree of specificity to which they describe protein function. This leads to the so-called ‘true path rule’: a sequence annotated with a given GO term is inherently associated with all its parent terms. Each GO annotation (annotated term) is further associated with an evidence code. These codes are primarily used to distinguish experimentally derived and computationally predicted annotations, with different codes in each class to describe more specifically *how* an annotation was derived or predicted (for details, see the GO documentation<sup>3</sup>).

### 1.3.2 Other systems

Alternative schemes for annotating proteins with molecular functions are the Enzyme Commission (EC) (Webb 1992) and Transporter Classification (TC) (Busch and Saier 2002) systems; these are applicable to enzyme and transport proteins only, respectively. The EC system is still as widely used as the more recently introduced GO system. In contrast to the EC and TC systems, the Riley scheme (Riley 1993) (the first functional ontology that was devised), the MIPS Functional Catalogue (FUNCAT) (Ruepp, Zollner et al. 2004) and the

---

<sup>3</sup> <http://www.geneontology.org/>



Kyoto Encyclopaedia of Genes and Genomes (KEGG) (Kanehisa and Goto 2000) classify proteins according to their positions in cellular pathways and processes (corresponding to the GO biological process DAG).

The EC system uses four-digit numbers to reflect a four-level hierarchy of enzyme functions. The first position refers to one of six general enzyme classes (e.g., ligases), the second to a certain sub-class and the third and fourth positions (usually) distinguish between specific substrates and cofactors; for example, EC 1.1.1.1 captures an alcohol:NAD<sup>+</sup> oxidoreductase activity. The original Riley scheme and its descendants (Rison, Hodgman et al. 2000) assign prokaryotic proteins to cellular processes using a hierarchical numbering system, similar to the EC system. FUNCAT extends this system to all kingdoms of life, and to more specific processes. The KEGG Orthology (KO) assigns KO terms, each referring to a certain family of supposedly orthologous proteins that perform the same function in an evolutionary conserved metabolic pathway.

## **1.4 Bioinformatics methods**

The most important bioinformatics concept in the context of the present work is that of the sequence alignment profile. A necessary precondition for the construction of such profiles is the alignment of multiple sequences. In turn, the corresponding multiple alignment methods build on algorithms for the pairwise alignment of sequences. Such algorithms, optimal or heuristic, form the foundation of bioinformatics research. Starting from those, the basic algorithms that underlie the above concepts are described in the following.

### 1.4.1 Sequence alignment

The similarity of protein sequences can be measured by the use of sequence alignment methods, which try to align evolutionarily equivalent residues. In

this, point mutations are accounted for by using a matrix that captures the (observed) probabilities with which amino acid residues are replaced by other residues in the course of evolution (substitution matrix; for example, PAM (Dayhoff, Schwartz et al. 1978) and BLOSUM (Henikoff and Henikoff 1992)). In addition, gaps can be introduced at different positions in the aligned sequences, to account for insertions and deletions.

Alignment methods usually first generate a (normalised) match score that expresses how good an alignment is in comparison with any other alignment. This depends on the degree of residue conservation in each column of the alignment and a predefined gap penalty. A second score is then calculated that indicates how likely it is to attain the observed match score by chance, that is, how statistically significant the match is. This is based on a distribution of hypothetical scores for random sequences. Common methods and algorithms for the pair- and group-wise alignment of sequences are described in the following.

#### 1.4.1.1 Pairwise sequence alignment

Pairwise sequence alignment can be done in either an optimal or a heuristic manner, and either in a global (whole-sequence) or local manner. The classic algorithm for optimal global pairwise sequence alignment is the Needleman-Wunsch algorithm (Needleman and Wunsch 1970); its local pendant is the Smith-Waterman algorithm (Smith and Waterman 1981). Both are based on the dynamic programming approach (Bellman 1952). As these algorithms have quadratic time complexity, heuristic methods were developed later on. The most widely used heuristic tools for local pairwise sequence alignment are FASTA ('Fast All') (Lipman and Pearson 1985) and BLAST (Basic Local Alignment Search Tool) (Altschul, Gish et al. 1990). These tools are commonly used to search entire databases of (target) sequences with a given query sequence. As both algorithms are derived from the Smith-Waterman

algorithm and are very similar in principle, only BLAST is described in the following.

As a heuristic method, BLAST breaks the problem of finding a good local alignment between two sequences down into finding several very similar, short residue stretches ('words') first, and connecting these subsequently. The basic workflow is as follows.

- i) All words of length  $k$  (the default setting for protein sequences is 3) that are found in the query sequence are stored in a table  $W$ .
- ii) Using a substitution matrix, all possible words of length  $k$  that yield a score higher than a given threshold  $t$  (the default setting for protein sequences is 13) when compared with one of the words in  $W$  are added to  $W$ ; the insertion of gaps is not allowed in the word comparisons.
- iii) The target sequence is queried with all words in  $W$ , which is referred to as 'seeding'. Matches are subsequently extended to so-called high-scoring segment pairs (HSPs) in both directions, allowing for gaps. This continues until the total, cumulative alignment score sinks below a given threshold or the end of either sequence is reached.
- iv) The local alignments derived in this way are connected, given that they show a sufficiently high score and small distance to each other, respectively. The individual alignments scores are summed up and the connected alignments are reported as the BLAST result.

The two central measures for the evaluation of BLAST alignments are the overall alignment score  $S$  and the so-called expectation value  $E$  (E-value). These values are interdependent:  $E$  is the statistical measure of the significance of a score  $S$ . Thus, given a random sequence composition of

query and target sequence (of lengths  $m$  and  $n$ ), there are  $E$  alignments with a score of at least  $S$  expected to occur by chance alone.  $E$  is defined as follows:

$$E = K \cdot m \cdot n \cdot e^{-\lambda S}$$

The lengths of the query and target sequences are parameters in the calculation of  $E$ . When either  $m$  or  $n$  are doubled, for example, the probability of seeing a score  $S$  double as well. Further,  $E$  decreases exponentially with increasing alignment score. This seems reasonable based on the consideration that to double the alignment score  $S$ , a HSP must attain the given score ‘twice in a row’. The parameters  $K$  and  $\lambda$  are statistical parameters depending on the size of the search space ( $m \cdot n$ ) and the applied scoring system. The latter refers to the substitution matrix used, which assigns a ‘cost’ to the alignment of each pair of different residues. For protein alignments this is based on the biochemical and biophysical (dis)similarities between amino acids and the respectively expected replacement frequencies. The gapped BLAST algorithm further penalises the insertion and extension of gaps by different costs. By default, the protein BLAST program BLASTP uses the BLOSUM62 matrix (Henikoff and Henikoff 1992). To account for different scoring schemes each alignment further gets assigned a so-called ‘bit score’ value  $S'$ . This is calculated based on the raw alignment score  $S$ :

$$S' = \frac{\lambda \cdot S - \ln(K)}{\ln(2)}$$

Bit scores are directly comparable given that the size of the search space remains unchanged. In contrast to  $E$ ,  $S'$  grows linearly with the length of alignment. Again, the E-value is a measure for the probability of seeing a given bit score depending on the size of the search space. It is connected to  $S'$  by the following equation:

$$E = m \cdot n \cdot 2^{-S'}$$

The E-values and bit scores returned by other tools that compare sequences or groups of sequences are calculated in manners very similar to those outlined above for BLAST.

#### 1.4.1.2 Multiple sequence alignment

Following from the time complexity of optimal pairwise sequence alignment, a straightforward extension of the respective dynamic programming algorithms to the alignment of multiple sequences results in exponentially increasing runtimes. Therefore, common methods for multiple sequence alignment (MSA) use different heuristics, similar to those used in BLAST. The most important of those is the general strategy of ‘progressive’ MSA, that is, the construction of the MSA from individual pairwise alignments. This requires two steps: first, the construction of a ‘guide tree’ using an efficient sequence clustering method, and second, the iterative (progressive) addition of sequences to a growing MSA in the order suggested by the tree (starting from the most similar pair of nodes, i.e., sequences). The second step requires the growing alignment to be expressed as a residue profile in each iteration (see Section 1.4.2.1), thus ‘simulating’ a pairwise sequence alignment.

So-called iterative alignment methods use the progressive alignment paradigm but refine the growing alignment in each round, by partial realignment of pairs of sequences. The most popular progressive alignment tool is CLUSTAL (Thompson, Higgins et al. 1994). The MAFFT alignment method (Katoh, Kuma et al. 2005; Katoh and Toh 2008) combines both the progressive and iterative approaches, and is both faster and more accurate than CLUSTAL and most other (more sophisticated) methods (Thompson, Linard et al. 2011). It was therefore used for all alignment tasks in the present work.

## 1.4.2 Alignment profiles

The residue distributions in multiple sequence alignments can be captured in (alignment) profiles (Gribskov, McLachlan et al. 1987). These can then be used to assess whether an arbitrary sequence is similar to the sequences in the alignment. Further, pairs of profiles can be compared to measure how similar the sequences in two alignments are. There exist two commonly used types of profiles: Position-Specific Scoring Matrices (PSSMs) and profile Hidden Markov Models (profile HMMs). The algorithms used to construct and compare such profiles are outlined in the following.

### 1.4.2.1 Construction

For each residue position in an MSA, the corresponding alignment profile captures the probability for each residue type to occur. In the case of protein MSAs, the residue types are the different amino acids. As the construction of both PSSMs and profile HMMs requires the same set of initial steps, these are outlined first below. A description of the additional steps necessary to create profile HMMs follows thereafter.

The simplest approach to create an alignment profile is the following. For each alignment position (column), the occurrence counts of all residue types (amino acids) are divided by the number of rows (sequences) in the alignment. The result is an alignment profile, with an observed frequency (probability) value between zero and one for each residue and alignment position. Several additional steps are typically used to produce profiles of higher quality, which primarily refers to their sensitivity in recognising related sequences. For example, weighting schemes are usually employed to account for redundancy in (some of) the aligned sequences (Henikoff and Henikoff 1994).

Many MSAs do not contain a sufficient amount of information (sequences) to construct a profile that is representative of, for example, a certain sequence

family. If no further related sequences are at hand (known) and can be added to the MSA, so-called pseudo-counts (for residue types that are not observed at all) can be used to add some ‘leeway’ to a profile (Dodd and Egan 1987; Tatusov, Altschul et al. 1994). This can serve to make the profile more sensitive in detecting remote family members. Instead of pseudo-counts, expected residue frequencies can be used to convert a (simple) profile into a PSSM.

PSSMs usually contain log-likelihood ratios instead of frequency (probability) values. These are calculated for each residue in a given position from its observed frequency and its expected frequency. To determine the latter, an empirical background distribution is required that describes how frequently a given residue type occurs in sequences in general. This can be derived from a (large) collection of manually curated MSAs, as used in the construction of substitution matrices, or simply a large collection of sequences. Arbitrary query sequences can be ‘scanned’ with a PSSM by summing the position-specific log-odds values (as found in the PSSM) for all residues they contain.

Profile HMMs capture the content of alignment profiles in a yet more sophisticated way. A given profile can be modelled as a (Markov) chain of states that each can ‘emit’ a range of symbols from an alphabet of size  $N$ . For a protein MSA, the alphabet contains 21 symbols: the 20 standard amino acid letters (‘match’ states) and a letter indicating a gap (‘indel’ state). Each of the states is associated with an  $N$ -dimensional probability vector that describes the probability for the state to emit a particular symbol (‘emission probability’), respectively; the individual probability values in each vector sum to one. The states in such a chain (model) can be traversed from left to right (start to finish), thereby emitting a sequence of symbols. In doing so, it is possible to remain in an indel state for more than one step, that is, to emit several gaps in a row. The probabilities of either doing so or not doing so, that is, moving forward to the next node, are stored in a second probability vector of size two

(‘transition probabilities’). Each state in the chain is associated with such a vector, however, for the match states the probability of moving forward to the next state is always one.

A Hidden Markov Model is ‘hidden’ since only the emitted symbol sequence(s) can usually be observed (for example, the rows in an MSA), while the underlying emission and transition probabilities cannot directly be inferred. In ‘training’ on an MSA, algorithms such as the Viterbi algorithm (Viterbi 1967), which is similar to the dynamic programming algorithms used in pairwise sequence alignment, can be used to infer these probabilities, and so generate a profile HMM. Scoring an arbitrary sequence against such a model means to assess how likely it is for the model to generate this specific chain of residues. Therefore, the chain is ‘fed through’ the model, multiplying the corresponding, subsequent emission and transition probabilities to attain an overall score.

#### 1.4.2.2 Comparison

To compare a sequence to an alignment (profile), the sequence and the profile have to be aligned. This is implemented for PSSMs, for example, in the PSI-BLAST (Position-Specific Iterated BLAST) method (Altschul, Madden et al. 1997). The latter first scans a target sequence database with a query sequence, like BLAST (see Section 1.4.1.1). However, it then constructs a PSSM based on the sequences hit and uses this profile to scan the database again. This can be done in several iterations, leading to an enhanced ability over BLAST to detect remote homologues.

To compare two alignments via their profiles, the profiles have to be aligned. The COMPASS (COMparison of Multiple Protein Alignments with Assessment of Statistical Significance) set of tools (Sadreyev and Grishin 2003), which was used in the work presented here, generates and aligns ‘generalised’ PSSMs that incorporate position-specific gap penalties (in contrast to the fixed gap penalties in the BLAST suite of methods). The



algorithm used to align two PSSM profiles is a straightforward extension of the sequence-sequence and sequence-profile alignment algorithms used in BLAST and PSI-BLAST. Accordingly, an E-value score is calculated to indicate the statistical significance of detected similarities. Methods that compare alignments in the form of profile HMMs have also been published (Soding 2005; Madera 2008).

## 1.5 Bioinformatics resources

The domain sequence data used in the present work was provided by Gene3D (Buchan, Shepherd et al. 2002), which itself relies on the CATH (Orengo, Michie et al. 1997) resource and the major protein sequence databases, UniProtKB (Magrane and Consortium 2011), RefSeq (Pruitt, Tatusova et al. 2009) and Ensembl (Flicek, Amode et al. 2011). In turn, CATH classifies structures from the PDB (Berman, Westbrook et al. 2000), and the protein databases are ultimately sourced from the primary nucleotide sequence databases. In addition, several secondary classification resources for proteins and protein domains are relevant to this work and/or are used for comparative purposes. The most important of these resources are introduced in the following.

### 1.5.1 Primary sequence and structure databases

The largest existing repository for nucleotide sequences (genes) are the mutually mirrored INSDC (International Nucleotide Sequence Database Collaboration) databases (Cochrane, Karsch-Mizrachi et al. 2011), most prominently the GenBank resource (Benson, Karsch-Mizrachi et al. 2011) that is hosted at the National Center for Biotechnology Information (NCBI) in the United States. The largest existing repository for protein sequences is UniProtKB (Universal Protein Resource Knowledge Base) database, which is also a collaborative effort, hosted at the European Bioinformatics Institute (EBI) in the UK. UniProtKB is further subdivided into the SwissProt and

TrEMBL databases, which store curated and non-curated sequence data (and corresponding information), respectively. The Protein Data Bank (PDB) is the primary resource collecting protein three-dimensional structures, as solved by X-ray crystallography, NMR, EM and other methods.

Due to the rapid progress in sequencing technologies over the last three decades, both GenBank and UniProtKB have been growing and continue to grow with exponential pace (Cochrane, Karsch-Mizrachi et al. 2011; Magrane and Consortium 2011). The PDB has also been growing at near-exponential rates in the past and continues to grow (Berman, Westbrook et al. 2000; Rose, Beran et al. 2011), while a decreasing number of novel folds are being discovered (Chandonia and Brenner 2006; Jaroszewski, Li et al. 2009). As of August 2011, GenBank contains about 140 million sequences, UniProtKB over twelve million sequences and the PDB over 70,000 sequences. SwissProt contains about 500,000 sequences, that is,  $\sim 5\%$  of the sequences in UniProtKB.

### 1.5.2 Protein classification resources

A multitude of resources exist that classify protein sequences according to sequence, structure and function (Henikoff and Henikoff 2001; Mulder 2001; Redfern, Grant et al. 2005). These follow different grouping concepts as discussed in Section 1.2.2, but will be collectively referred to as ‘family resources’ in the following. Family resources share the following key characters. First, manual curation is involved, to varying extents. Second, a model library concept is followed, as opposed to, for example, a full clustering of all available sequences. In brief, this concept entails a workflow that (i) starts with a set of (curated) seed sequence groups for different families (classification), (ii) continues with the extension of these groups to families (extension) and (iii) finishes with building one or more models to recognise each family, respectively (library generation).

### 1.5.2.1 Classification based on structure

Both SCOP/SUPERFAMILY and CATH/Gene3D, respectively, are ‘sister’ resources for the structural classification of protein domains. Formally, both distribute the above-described workflow across two separate resources, respectively: SCOP and CATH correspond to the classification stage (with PDB structures as the primary input), whereas SUPERFAMILY and Gene3D incorporate the extension and library generation stages (with sequence data as the primary input). Manual curation is used in both SCOP and CATH. While the former is a largely manual effort, CATH uses curation only in particularly difficult stages of the classification process (Greene, Lewis et al. 2007). SUPERFAMILY and Gene3D are both entirely automated resources. SCOP and CATH both follow a hierarchical arrangement. Several superfamilies can share the same fold (SCOP) or topology (CATH), with both terms referring to a relationship of ‘structural similarity without a significant signal of homology’. The superfamily level is the most relevant in the context of this work and therefore focussed on in the following.

Below the fold level, the Structural Classification of Proteins (SCOP) resource defines both a superfamily and a family layer. Superfamilies of homologous domains are identified manually, which includes the assignment of domain boundaries to the incoming protein structures from the PDB. Each domain identified in this way is either assigned to an existing superfamily or nucleates a novel superfamily. The sequences in each superfamily are further assigned to families, which are identified in a semi-automatic manner: a clustering at 30% sequence identity is followed by a manual merging of individual (singleton) clusters in cases where the clustering threshold is not met but structural and/or functional properties indicate shared family membership.

The CATH database defines four hierarchically organised levels of domain classification: Class, Architecture, Topology, and Homologous Superfamily;

hence the name. Superfamilies are identified using a battery of tools for sequence comparison, structure comparison, clustering and domain boundary assignment (Greene, Lewis et al. 2007). In addition, manual curation is employed in the key steps of domain boundary assignment and the assignment of domains on the topology (fold) level. A pair of domain sequences (structures) is assigned to the same superfamily if it meets at least one of the following criteria:

- i) A sequence identity of at least 35% in conjunction with at least 60% of the longer sequence covering the shorter (overlap).
- ii) A SSAP (Taylor and Orengo 1989; Orengo and Taylor 1996) (structure alignment tool) score of at least 80 in conjunction with a sequence identity of at least 20% and a minimum overlap of 60%.
- iii) A SSAP score of at least 70 in conjunction with a minimum overlap of 60% and a clear similarity in function, as inferred from the literature and the Pfam domain family database.
- iv) Significant (if potentially very low) sequence profile similarity in profile-profile comparison (see Section 1.4.2.2) with SAM (Hughey and Krogh 1996), HMMER (Eddy 1998; Eddy 2009) and PRC (Madera 2008).

To generate SUPERFAMILY and Gene3D, one or more models (HMMs) are built to represent each (seed) superfamily defined in SCOP or CATH, respectively (Gough, Karplus et al. 2001; Lee, Grant et al. 2005). By scanning the major protein sequence databases with these models, the seed superfamilies (containing only structurally characterised sequences) are extended by homologous sequences from all fully or partially sequenced genomes (proteomes). As in SCOP and CATH, this involves the crucial step of domain boundary assignment, but without the help of structural

information. While the respective models hit a given protein target sequence in specific positions, the hits from different models frequently overlap. SCOP and CATH employ different algorithms to resolve such cases, as described in Gough, Karplus et al. (2001) and Yeats, Redfern et al. (2010), respectively.

#### 1.5.2.2 Classification based on sequence

An overview of the most important extant family resources is most easily achieved when looking at those that contribute to InterPro (Hunter, Apweiler et al. 2009), a meta-resource for protein classification that is described in detail in Section 5.1.2. Among the InterPro members are the most widely used resources for protein and domain family classification: SUPERFAMILY and Gene3D detect putative structural domains in protein sequences (see Section 1.5.2.1), Pfam classifies whole-protein and domain sequences, PRODOM (Servant, Bru et al. 2002) and SMART (Letunic, Doerks et al. 2009) classify domain sequences, and PANTHER (Thomas, Campbell et al. 2003), PIRSF (Nikolskaya, Arighi et al. 2006) and TIGRFAMs (Haft, Loftus et al. 2001) classify whole-protein sequences.

PRODOM automatically clusters evolutionary conserved sequence segments (putative domains) based on recursive PSI-BLAST searches of UniProtKB. The other family resources mentioned above all use libraries of HMMs to represent families, that is, they follow the model library concept (see above). For the most relevant of those resources (in the context of the present work) an overview is provided in the following, which is augmented by a discussion of the individual family layers in Chapter 5. There also exist many resources that aim to establish clusters of orthologous proteins (see Section 1.2.2.2). These are not immediately relevant to this work and are reviewed in Fang, Bhardwaj et al. (2010).

Pfam, as the most widely used sequence-based family resource, classifies protein and domain sequences into families of functionally related sequences

below the superfamily level, with a focus on domain function (a family concept has not explicitly been formulated; see also Section 1.2.2.3). Among the member databases of InterPro, the manually curated Pfam-A and the automatically generated Pfam-B parts of the resource together provide the highest coverage of the known protein sequence space. As of October 2011, more than 75% of all protein sequences are assigned at least one Pfam domain, from more than 12,000 families. Pfam families were and are created on an ad-hoc basis, with a bias towards large families (Sonnhammer, Eddy et al. 1997). The sequence groups underlying the corresponding, curated seed alignments are compiled using a variety of sources and tools, such as PROSITE (Sigrist, Cerutti et al. 2002), PRODOM, SwissProt and BLAST; published high-quality alignments of individual protein (domain) families are also used. Following a similar goal to Pfam, SMART consists of a considerably smaller but entirely manually curated set of domain families.

PANTHER aims to delineate functional divergence within homologous protein families found in metazoan species. By expert curation, the families are split into functionally conserved ‘subfamilies’, each annotated with GO molecular function and biological process terms. TIGRFAMs focuses on functional conservation as well, with half of its protein families containing so-called ‘equivalogs’. These are defined as sequences of conserved molecular function, and the families can therefore contain a mixture of orthologues, paralogues and xenologues; this definition may also (implicitly) apply to many Pfam protein families. PIRSF compiles ‘homeomorphic’ families of homologues, where all members show full-length sequence similarity and share the same domain architecture; conserved function is not a necessary requirement.

## 1.6 Summary of work and overview

A summary of the work presented here, in the order it was conducted, is first provided in the following. The subsequent section then closes the introduction, giving an overview of the following chapters.

### 1.6.1 Summary of work

The overarching aim of the presented work was the development of a software pipeline to identify the functionally conserved families within protein domain superfamilies. The individual steps it took to reach this aim are outlined in the following. After preliminary studies had shown the potential of using domain sequence and protein annotation data to study the functional plasticity of protein domain superfamilies (Addou, Rentzsch et al. 2009), the development of a clustering method for domain family identification (Lee, Rentzsch et al. 2010) stood at the beginning of this work. As the potential of using annotation data directly, not only in training the method, had been recognised, it was first extended to use EC annotations. These should serve both to select a relevant subset of the input data for clustering (i.e., reduce the computational overhead) and to guide the clustering process. The following switch towards using GO annotation data brought increased coverage but also new challenges. Further, the abandonment of exhaustive clustering in favour of a ‘leaner’ protocol, following the model library concept, required the design of a pipeline.

The integration of the developed methods for data preparation, clustering, family identification and assignment into the DFX (Domain Function Exploration) pipeline was necessary to make the large-scale processing of superfamilies possible whilst maintaining usability and flexibility. With regards to the latter, the sequence clustering and family identification steps were entirely disentangled at this point. This made it possible to embed the GemMA clustering protocol, the original family identification protocol (not

directly using annotation data) and the novel family identification protocol (using GO data) as independent modules in DFX. In the following, this was augmented with the development of several more specific modules, for example, for naming the identified domain families and for their use in whole-protein function annotation. Finally, a detailed analysis of the results (families) obtained with each of the developed methods was conducted, where this had not already happened. While a first version of DFX is now stably integrated with the Gene3D resource, many challenges remain to be addressed in future work. Their identification and detailed discussion formed the last part of the work presented here. An outline of the subsequent chapters follows.

### 1.6.2 Overview of chapters

The chapters of this thesis are arranged in the order the work was conducted, with the exception of Chapter 3, the DFX ‘overview’ chapter. This provides an overview of the pipeline and its individual models. In detail, the order of chapters is as follows.

Chapter 2 describes the development of a high-throughput HPC sequence clustering method based on alignment profile-profile comparisons, GeMMA.

Chapter 3 discusses the integration of GeMMA and further developed methods into a pipeline for the identification of functional families within all Gene3D domain superfamilies, DFX. The concept of domain function and the corresponding family concept followed in DFX are also defined in this chapter. The pipeline embeds two alternative protocols for family identification, which are discussed and evaluated in the two following chapters.

Chapter 4 introduces the unsupervised family identification protocol of the DFX pipeline, which is used in cases where individual domain superfamilies are not associated with any high-quality protein function annotation data. This protocol uses the results of domain sequence clustering with GeMMA in



conjunction with a generic granularity setting that is determined in an initial training step.

Chapter 5 describes the supervised family identification protocol of the DFX pipeline, which is used to process the majority of domain superfamilies. This protocol processes the clustering results in conjunction with high-quality protein function annotation data to derive domain families with conserved (domain) function.

Chapter 6 provides an overview of the results obtained with DFX in a quantitative manner. In particular, this involves different statistics on the domain families produced in the first large-scale run of the pipeline and an overall comparison of the results obtainable with each of the two family identification protocols.

Chapter 7 closes this thesis, with an overall summary of the work conducted, the current usage of the already generated family data, recent and further possible changes to the pipeline and how these changes are expected further to improve its performance. DFX is also put into context with a newly introduced method for domain family annotation in SUPERFAMILY, the recent protein function prediction challenge CAFA 2011, and with several recent studies on particular domain superfamilies. In the context of the latter, a generic protocol for such studies is proposed. The chapter closes with some final remarks on the state and direction of domain-centric research, the possibilities it provides and the requirements for its further advancement.

## Chapter 2. GeMMA: profile-based clustering of protein sequences in distributed computing environments

*This work has been published in Lee, Rentzsch et al. (2010) and is partly based on ideas of my colleague and co-author David A. Lee. Specifically, this refers to the two heuristics described in Section 2.2.3. Their theoretical foundation, implementation and all remaining parts of the chapter represent my own work.*

This chapter discusses the development of a novel, distributed method to cluster biological sequence data. GeMMA can be run in various HPC environments and is applicable to large input datasets with hundreds of thousands of data points. The background section primarily discusses the different types of generic algorithms that underlie individual existing sequence clustering methods. The implementation section then describes the GeMMA protocol in detail, focussing on the different heuristics and HPC strategies it uses to leverage the handling of large datasets with high throughput. The chapter closes with an outline of potential caveats and possible future improvements of the algorithm.

### 2.1 Background and aims

In the following, the importance of sequence clustering in general is first discussed, together with its main applications. Emphasis is put on the necessity to develop novel, flexible methods that can be used in a high-throughput setting, in the light of the ever increasing amounts of sequence data. The related features of the GeMMA (Genome Modelling and Model Annotation<sup>4</sup>) method and its advantages over existing methods are outlined. The rest of this section contains a detailed classification and review of existing clustering algorithms, followed by a brief summary of clustering

---

<sup>4</sup> The method was initially developed for structural genomics target selection.

evaluation strategies and an overview of the most widely used clustering tools and resources.

### 2.1.1 Clustering biological sequences

A comprehensive review of protein and protein domain sequence clustering by Liu and Rost (2003) concludes: ‘One point is clear: we urgently need better tools to dissect proteins into domains and to cluster these domains’. More generally, the clustering of different types of datasets, based on different similarity measures, is one of the most common requirements in bioinformatics analyses. Examples of data types to be clustered are: sequence data, expression profiles and scientific articles. Depending on the size of the dataset and the complexity of the clustering criterion, computational cost can quickly become a limiting factor.

The two general applications of sequence clustering are (i) redundancy removal and (ii) the automatic identification of different types of groups in sequence datasets. One example for the latter is the use of GeMMA in the DFX pipeline, as discussed in Chapter 3. Another is structural genomics target selection (Liu and Rost 2003), where protein sequence clusters are used to choose target sequences with high ‘impact’. This refers to the number of other proteins for which a homology model could be built if the respective protein structures were solved and used as templates. For both applications, large sequence datasets have to be clustered with high sensitivity.

In 1965, George Moore correctly predicted an exponential growth rate for the processing power of CPUs (Moore 1965). A similar observation has later been made with respect to the size (and cost) of storage media (Walter 2005). However, as of 2011, the amount of available sequence data increases even faster (Kahn 2011), and the genome of Moore himself was recently added to this data (Rothberg, Hinz et al. 2011). In the words of the review article

quoted above: ‘...the growth-rate for bio-sequences continues to grow’ (Liu and Rost 2003). Serving as a striking example, the field of metagenomics has already produced more sequence data than all whole-genome sequencing projects taken together (Wooley, Godzik et al. 2010).

While some existing sequence clustering methods can handle large datasets, sometimes with high speed, they suffer from different limitations. In particular, these are limited sensitivity in the detection of weak relationships between sequences, the (systematic) introduction of clustering errors by certain heuristics and/or impractically high memory requirements. In general, fast methods often lack sensitivity and introduce errors, while well-performing methods are commonly slow. Most clustering methods further require an all-by-all similarity matrix as input, which can often not efficiently be produced on a single standard desktop PC for large datasets (as of 2011). This is due to the processing power required to calculate all pair-wise similarities (speed bottleneck) and the amount of system memory required to hold the resulting matrix (memory bottleneck).

Based on the above considerations, the development of novel methods for clustering biological sequences is an important area of bioinformatics research. The GeMMA clustering protocol uses distributed computing, novel heuristics and a profile comparison strategy to balance speed with sensitivity. Further, it can process large sequence datasets in a memory-efficient manner.

### 2.1.2 Clustering algorithms

Clustering refers to a type of unsupervised learning process in which a dataset is partitioned into a number of (usually) disjoint subsets, according to the similarity relationships between all data points in the initial set. The generated subsets are called clusters. The produced clusters together form what is called a partitioning or clustering of the initial dataset. Clustering methods require

the choice of a (dis)similarity measure, based on which individual data points and/or clusters are compared. Further, the granularity of the obtained partitioning can usually be controlled for, in a method- and algorithm-specific way. The following sections describe the clustering algorithms most frequently applied to datasets of biological sequences. These broadly fall into hierarchical, partitional, graph-based and heuristic approaches, as explained in the following.

#### 2.1.2.1 Hierarchical clustering

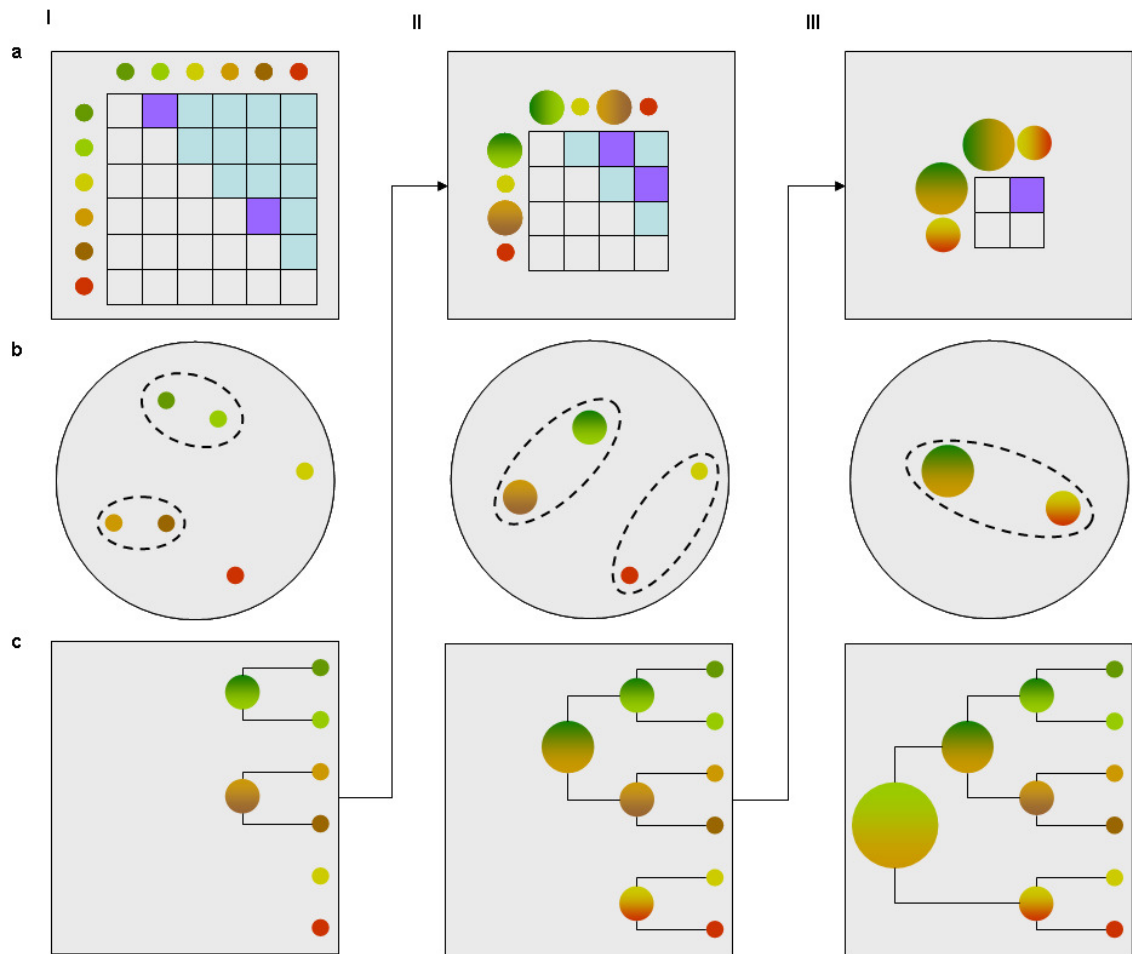
Hierarchical methods start with an all-by-all comparison of a set of initially defined clusters, usually containing individual data points. Subsequently, they merge or split clusters in a recursive manner. This process is best envisioned as the growing of a tree of clusters (clustering dendrogram), either from the leaf nodes to the root (agglomerative hierarchical clustering) or vice versa (divisive hierarchical clustering) (Johnson 1967). The process is illustrated in Figure 2.1. While the root cluster contains all data points, the leaf clusters each contain a single one. In standard agglomerative hierarchical clustering, the most similar pair of clusters is merged in each iteration. In divisive hierarchical clustering, partitional methods (see below) are used to split the most diverse existing cluster into two more homogenous clusters in each iteration.

The granularity of hierarchical clustering methods, that is, at which point (and whether or not) the iterative clustering process is stopped, is controlled using specific stopping criteria. As such, simple global threshold parameters are frequently used; for example, a similarity value that no pair of clusters must exceed, or a certain number of clusters not to be under-run. More complex stopping criteria can involve cost functions, as discussed in Section 2.1.3.1.

Hierarchical clustering methods commonly use one of three ways to measure cluster dissimilarity: single linkage, complete linkage or average linkage. In single linkage (or nearest neighbour) clustering, the distance  $d$  between two clusters  $A$  and  $B$  is defined as the distance between the two closest data points in  $A$  and  $B$ . In complete or multiple linkage (farthest neighbour) clustering,  $d$  is defined as the maximum distance of two points in  $A$  and  $B$ . In average linkage clustering  $d$  is calculated by averaging over the distances between any two points in  $A$  and  $B$ ; in the construction of phylogenetic trees this corresponds to the UPGMA (Unweighted Pair Group Method with Arithmetic mean) method (Michener and Sokal 1957). In addition, numerous derivatives of these three basic paradigms exist (Berkhin 2002). Based on the necessary initial all-by-all comparison, and depending on the dissimilarity measure used, naïve agglomerative clustering approaches have a non-linear time complexity of maximally  $O(n^2 \cdot \log(n))$ .

#### 2.1.2.2 Partitional clustering

Partitional methods directly cluster datasets at a single level of granularity, not producing a clustering dendrogram like hierarchical methods. The most widely used partitional clustering algorithm is the  $k$ -means or, more generally, the  $k$ -centres approach (Macqueen 1967). This algorithm starts with randomly assigning a chosen number ( $k$ ) of data points to be cluster centres and then iterates between two stages until convergence: (i) cluster formation, where each data point is assigned to the closest centre, and (ii) reassignment, where the centres of all clusters are recalculated. The number of centres  $k$  equals the produced number of clusters, and therefore represents the granularity parameter of the  $k$ -centres approach. Different strategies exist to calculate the cluster centres. Most commonly, the cluster centroids are used ( $k$ -means approach). If means can not be calculated for a given data type, medoids can be used instead ( $k$ -medoids approach). The medoid is the data point that is, on average, most representative for all data points in a given cluster.



**Figure 2.1. Hierarchical agglomerative clustering.** An example dataset with data points of different similarity (colours) is clustered in three iterations (I-III). (a) shows the shrinking similarity matrix; (b) shows the distance between the individual data points; (c) shows the growing clustering dendrogram. In each iteration, the most similar pair of data points is merged, as indicated by the purple matrix entries in (a); if several pairs are equally similar, all these pairs are merged.

K-centres clustering does not require an initial all-by-all comparison of the data points but only an all-by- $n$  comparison, where  $n$  equals the chosen number of clusters  $k$ . With a linear time complexity  $O(n)$  it is faster than hierarchical clustering, and its implementation requires less memory (Fayech, Essoussi et al. 2009). Still, the k-centres approach is rarely used to cluster biological sequence data. This is despite the fact that k-means centroids could be calculated in the form of alignment profiles. One reason for this lack of popularity is the requirement to specify a fixed number of clusters initially. This usually implies prior manual and/or algorithmic analysis of the input dataset. In addition, partitional algorithms can converge to locally optimal

solutions and, when initialised randomly, do not yield identical results in repeated runs on the same dataset.

### 2.1.2.3 Graph-based clustering

Like hierarchical clustering methods, graph-based clustering methods require as input an all-by-all matrix of pair-wise similarities between the data points. This matrix is transformed into a graph (network), where nodes (vertices) represent data points and edges (connections) represent relationships between them; each edge can additionally be associated with a weight value. The ‘global’ character of graph-based approaches is thought to make them more powerful than hierarchical and partitional methods on some datasets (Jaromczyk and Toussaint 1992; Schaeffer 2007; Wang, Li et al. 2010). Based on iterative updates of the initial matrix, they can take into account the similarity relationships between all data points at any point in clustering.

When clustering biological sequences, the graph to be clustered is a sequence similarity network, in which nodes represent sequences and edges represent similarity relationships between them. Another example is biomolecular interaction networks, in which nodes represent different types of molecules and edges indicate the (probabilities of) interactions between them. The algorithms that are most widely used to cluster such networks are Markov clustering (MCL) (van Dongen 2000) and affinity propagation clustering (APC) (Frey and Dueck 2007). Both are outlined in the following. Other potentially powerful algorithms are spectral clustering (Shi and Malik 2000), superparamagnetic clustering (Blatt, Wiseman et al. 1996) and transitivity clustering (Wittkop, Emig et al. 2010). However, so far these have been seldom applied and existing implementations are relatively slow.

MCL was conceived by S. van Dongen in 2000. It requires as input a symmetric similarity matrix, from which it generates a stochastic Markov



matrix of ‘transition probabilities’ between all data points. These probabilities are the edge weights in the corresponding sequence similarity network. The algorithm then iteratively simulates random walks between the nodes in this network, guided by the edge weights, and adjusts these weights. In each iteration, ‘flow’ is promoted where it is already strong (high transition probability) and lowered or entirely removed where it is weak (low transition probability). In this manner, MCL converges on a set of disjoint clusters. The process is technically implemented in the form of matrix multiplication operations on the underlying stochastic matrix, where flow promotion corresponds to matrix ‘expansion’ and lowering flow corresponds to matrix ‘inflation’. Expansion and inflation are iterated over until convergence, that is, until no net change in the matrix is observed anymore (van Dongen 2000). The clustering granularity of MCL is determined by setting an inflation parameter. An improved version of the algorithm, based on dynamically decreasing the value of the inflation parameter during clustering, was published by Medves and colleagues in 2008 (Medves, Szilagyi et al. 2008).

APC was initially published by (Frey and Dueck 2007) and has later been reformulated in a simpler manner by (Givoni and Frey 2009). Since both formulations of the algorithm yield equal results, the simpler version is summarised in the following. Somewhat similar to flow simulation in MCL, APC is an algorithm that passes availability and responsibility messages between nodes (data points) connected by edges in a similarity network. The aim is to converge on a set of so called ‘exemplars’, data points that best represent the partitions inherent in the dataset to be clustered. Exemplars correspond to the centres in k-centres partitional clustering methods (see above). Accordingly, APC shares with these methods the strategy of minimising an overall cost function in each round: the sum of distances between all data points and their respective exemplars. The final clusters can directly be derived from the exemplars, since all data points are assigned to

one and only one exemplar in all steps of APC. The latter is the first of two important constraints in the clustering process. The second constraint is that a data point can only be assigned another data point as its exemplar given that this other data point also is its own exemplar. The clustering granularity of APC is controlled by a so-called preference parameter, which corresponds to the inflation parameter in MCL.

Graph-based sequence clustering is frequently used with the aim of partitioning arbitrary sequence datasets according to one of the concepts described in Chapter 1 (protein superfamily, family or orthologue cluster). However, as for any other type of clustering algorithm, this can only be done successfully in conjunction with an unsupervised or supervised clustering evaluation strategy (see Section 2.1.3), that is, a strategy to estimate optimised, case-dependent settings for the respective clustering granularity parameter. Chapters Chapter 4 and Chapter 5 discuss such combined protocols in the context of protein family identification.

#### 2.1.2.4 Greedy incremental methods

Greedy incremental clustering methods are frequently used to partition sequential data (for example, strings and vectors) in a fast but heuristic manner. The granularity parameter of these methods is a global redundancy threshold that defines the level of pair-wise similarity above which two data points should share the same cluster. The basic idea was formulated for the clustering of biological sequences by Hobohm and colleagues in 1992 (Hobohm, Scharf et al. 1992).

The generic workflow of the greedy incremental algorithm (or: list removal algorithm) is as follows. First, all sequences in the target dataset are sorted in order of decreasing length. The generated list is then traversed from top to bottom and each sequence is either (i) added to the (initially empty) set of

cluster representatives or (ii) assigned to an existing representative. This decision depends on whether or not the respective sequence is sufficiently similar to (at least) one of the existing representative sequences. The result of this process is a set of representative sequences, each representing a cluster that contains the representative itself and any sequences assigned to it. In other words, the result is a partitioning of the input dataset.

Greedy incremental clustering corresponds to a  $k$ -centres approach (see above) *without* iterative cluster refinement. It only deviates from this definition in that it automatically delineates the number of clusters  $k$ , based on the redundancy threshold value set. Generally, the algorithm operates in  $O(n \cdot k)$  linear time. Existing implementations of greedy incremental sequence clustering use different sequence similarity measures and follow different strategies to select the representative sequence to which a given sequence is assigned to.

The most influential strain of greedy incremental clustering methods uses a BLAST-like short word filtering approach (see Section 1.4.1.1) to compare sequences with high efficiency (Grillo, Attimonelli et al. 1996; Holm and Sander 1998; Li, Jaroszewski et al. 2001; Edgar 2010). Short word filtering is based on the notion that a certain degree of overall sequence similarity between two sequences necessarily implies that the sequences also match in a number  $r$  of short residue stretches with length  $k$ ; these are commonly referred to as ‘words’ or  $k$ -tuples. For example, a sequence identity value of 90% requires two sequences with length 100 to share at least one continuous stretch of 10 identical residues, a 10-tuple or decamer.

CD-HIT (Li and Godzik 2006) is the so-far most widely used heuristic clustering tool and relies on the above strategy. It uses small word sizes in combination with in-memory lookup tables for fast sequence comparisons. According to (Edgar 2010) CD-HIT is outperformed by the UCLUST method, in terms of speed, memory requirement and accuracy. UCLUST

follows a very similar algorithmic workflow but uses the USEARCH algorithm (Edgar 2010) to compare sequences. In general, both CD-HIT and UCLUST provide (profoundly) increased speed at the cost of diminished accuracy when compared with methods that implement non-heuristic clustering algorithms such as those discussed above.

### 2.1.3 Clustering evaluation measures

Clustering methods can be used to partition input datasets at arbitrary levels of granularity. This is done by adjusting the respective method-specific stopping criteria and granularity parameters, as discussed above. These options are sufficient for some important applications of clustering, such as redundancy removal in biological sequence datasets or a uniform, hierarchical sampling of sequence space. However, it is often necessary not only to cluster a given dataset but also to be able to select from a set of different possible partitionings the *best* one.

Clustering evaluation measures are used to estimate the degree to which an obtained partitioning of a given dataset corresponds to the (assumed or known) ‘true’ underlying structure of the dataset. There exist two types of measures. Unsupervised measures require no information apart from that used and/or obtained in the clustering process. They are, by definition, measures of *relative* goodness. Supervised measures require additional, external information. They are frequently used in benchmarking; in particular, to benchmark the performance of unsupervised measures.

Examples for the combined use of clustering algorithms and unsupervised evaluation measures are the automatic *ab-initio* methods for protein family identification discussed in Chapter 4. In contrast, the benchmarking of GeMMA (used in isolation for the same purpose) in the same chapter is an

example of supervised evaluation. Both supervised and unsupervised measures are outlined in the following.

### 2.1.3.1 Unsupervised measures

Most unsupervised measures for evaluating specific partitionings of a given dataset rely on the assumption that a good partitioning maximises both cluster cohesion and cluster separation. Cohesion or ‘compactness’ refers to the average similarity between all data points in a cluster. Separation or ‘isolation’ refers to the average similarity between all data points in a cluster and the data points in other clusters. It indicates how well-separated a cluster is from all other clusters. If a cost function is designed that takes the cohesion and separation values of all clusters in a partitioning into account, this function can be used as an unsupervised evaluation measure (see below).

A general unsupervised strategy to measure the quality of a given partitioning is sampling; that is, assessing it in the light of (many) other partitionings generated for the same dataset. Consensus approaches assume that a clustering solution is good given that the partitions it proposes are robust towards changes in the parameter settings of the clustering method in use. The same rule can be applied to the results of repeated runs (with unchanged parameters) when a non-deterministic method such as k-centres is used. Cost function approaches assess the behaviour of a specifically designed function over the range of all sampled partitionings. Good solutions can then be identified at global optima or at points of sharp transition and/or specific stability in slope of this function. Commonly used cost functions are based on cluster cohesion and separation, the number of produced clusters, or the size of the ‘giant component’. The latter refers to the largest cluster in a given partitioning and can be used as another (rather coarse) measure of its quality. According to this, a balanced partitioning of a given dataset can generally be expected close to the point at which a giant component can be clearly

detected. For example, in Dokholyan, Shakhnovich et al. (2002) the authors estimate that a good partitioning can be found at the point at which the giant component contains about half of all data points.

Unsupervised clustering evaluation measures, such as those described above, are an integral part of *ab-initio* methods and protocols for protein family identification, as described in Chapter 4.

### 2.1.3.2 Supervised measures

Supervised measures for clustering evaluation generally rely on external information in the form of ‘gold standard’ datasets. The ideal (correct) partitioning of such datasets is known; that is, the individual data points are assigned to one of several known classes within the dataset, respectively. The quality of any given partitioning of the same dataset can therefore be measured by accessing how well it matches the gold standard classification. In general, this is calculated using the notions of sensitivity (are all data points that belong to the same class found in the same partition?) and specificity (do the data points in each partition belong to a single class, respectively?). Scores that measure sensitivity and specificity are usually integrated to yield an overall performance score, as only the combination of both provides a good estimate of how well the proposed and known partitionings match.

There exist different ways for deriving values of sensitivity and specificity in the context of clustering, and different strategies to combine these values into a single value (for a comprehensive review see (Tan, Steinbach et al. 2005)). Traditionally widely used are measures that count pairs of data points, such as the Rand (Rand 1971) and Jaccard (Jaccard 1901) indices. In brief, these are based on measuring how many pairs of data points that are found in the same (in different) class(es) in the reference partitioning, respectively, show the same relationship in the proposed partitioning. More recently, several

information theoretic measures have been introduced (Vinh, Epps et al. 2010), with the V (Rosenberg and Hirschberg 2007) and VI (Variation of Information; Meila [2007]) measures being the most influential. These take into account both the purity and completeness of each individual cluster in the proposed partitioning, with regards to the class membership of its data points in the gold standard classification, respectively. The VI measure is used as one of three measures in evaluating protein (domain) family partitionings in Chapter 4 and Chapter 6.

#### 2.1.4 Existing tools and resources

Hierarchical and heuristic clustering algorithms underlie the generic tools that are most widely used to cluster sequences. These are BLASTClust (Altschul, Gish et al. 1990) and CD-HIT (Li and Godzik 2006), respectively. The former is part of the NCBI BLAST package and implements standard single linkage agglomerative hierarchical clustering; it uses BLAST raw scores as the similarity measure and optional constraints on sequence overlap. CD-HIT is based on a greedy incremental clustering algorithm and uses short word matches to measure sequence identity, as described in Section 2.1.2.4. The only input required for either method is protein or DNA sequence data.

Heuristic methods such as CD-HIT are very fast but not very accurate. Therefore, they are primarily (and very frequently) used to generate non-redundant sequence sets at arbitrary levels of maximum pair-wise sequence identity; for example, in the UniRef (Suzek, Huang et al. 2007) and SwissProt parts of UniProtKB and in the CAMERA repository for metagenomic sequence data (Seshadri, Kravitz et al. 2007). The NCBI still uses nr90db (Holm and Sander 1998) to make its DNA databases non-redundant, a Perl script that is a remote ancestor of CD-HIT.

Hierarchical clustering algorithms can be used to obtain multi-layer libraries of sequence clusters. Two resources use this approach to cluster the known sequence space as a whole, ProtoNet (Sasson, Vaaknin et al. 2003) and CluSTr (Kriventseva, Fleischmann et al. 2001). ProtoNet is based on a memory-constrained implementation of UPGMA (Loewenstein, Portugaly et al. 2008) and uses BLAST similarity scores. In contrast, CluSTr follows the single linkage paradigm but uses Smith-Waterman alignment Z-scores. Both resources use heuristics and redundancy removal schemes to cope with the immense sequence load. ProtoNet builds a ‘skeleton’ cluster tree from the sequences in SwissProt only, and subsequently assigns all TrEMBL sequences to their best matching clusters in this tree using BLAST (Sasson, Vaaknin et al. 2003). CluSTr excludes any clusters from the output set that contain more than 90% of the sequences in their direct parent cluster.

Interestingly, CluSTr is no longer being maintained (as of 2011), a sign of the immense computational overhead involved. A promising basis for related large-scale clustering projects in the future is the Similarity Matrix of Proteins (SIMAP) resource (Rattei, Tischler et al. 2008). This provides a regularly updated all-by-all similarity matrix of all known protein sequences, using a FASTA-based comparison algorithm in a distributed volunteer computing framework (Anderson 2003). SIMAP itself further implements a basic clustering scheme, using MCL.

Many studies and resources that apply sequence clustering in a more specific manner, for example, to derive families, use graph-based algorithms such as MCL, APC or spectral clustering. The respective tools can usually be obtained from the authors, in the form of standalone executables or packages for the R programming language. MCL was first used to cluster protein sequences in 2002, as part of the TRIBE-MCL (Enright, Van Dongen et al. 2002) workflow; one of the co-authors was the MCL inventor S. van Dongen. TRIBE-MCL formed the basis for the later abandoned TRIBES protein



family resource (Enright, Kunin et al. 2003) and has become part of the Ensembl pipeline (Flicek, Amode et al. 2011). In addition, several more recent, lineage- and organism-specific resources use MCL to establish sequence clusters, for example, PlantTribes (Wall, Leebens-Mack et al. 2008) and YeastWeb (Chu, Yuan et al. 2010). Li and colleagues developed the OrthoMCL protocol (Li, Stoeckert et al. 2003) and created the corresponding OrthoMCL-DB resource (Chen, Mackey et al. 2006), which aims to establish orthologue clusters through the combined use of pair-wise sequence comparisons and MCL. Other graph-based clustering methods have been used in a similar way, for example, APC in Frey and Dueck (2007) and spectral clustering in Paccanaro, Casbon et al. (2006).

## 2.2 Implementation

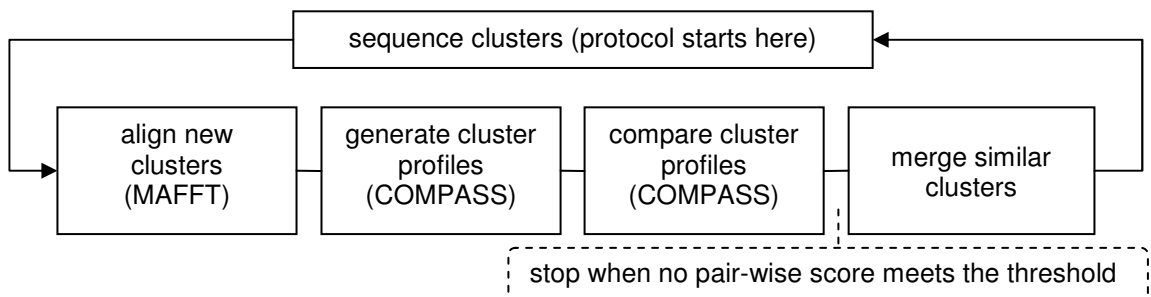
The following sections describe the implementation of the GeMMA clustering protocol. The protocol as a whole is first outlined. Subsequently, its modularity, high-throughput heuristics and HPC implementation are discussed in detail.

### 2.2.1 The GeMMA clustering protocol

The GeMMA clustering protocol is based on the common agglomerative hierarchical clustering paradigm. It takes as input a set of starting clusters, each containing one or more sequences, and then iteratively performs pair-wise cluster comparison and cluster merging operations. This process proceeds until a stopping criterion is met or only a single cluster is left. The stopping criterion is based on a global cluster similarity threshold: the clustering is stopped when no pair of clusters compared in a given iteration is more similar than the specified threshold value. This overall workflow is outlined in Figure 2.2. In addition to the final partitioning, GeMMA produces a full trace of the clustering process (in particular, the cluster merging order).

Further, all clusters and cluster alignments produced in the course of clustering can be stored.

GeMMA deviates from other hierarchical clustering methods that are used to cluster sequences in three key points. First, cluster dissimilarity is measured using a ‘profile linkage’ approach. This is similar in principle to the widely used average linkage paradigm. However, instead of carrying out all-by-all sequence comparisons between pairs of clusters, GeMMA builds and compares cluster profiles. Second, a ‘comparison sampling’ heuristic is used to speed up the clustering of large datasets. In brief, not the full set of all possible pair-wise cluster comparisons is carried out at any one point in clustering but rather a randomly drawn subset. Third, a ‘greedy merging’ heuristic is used as a further speed-up strategy. Based on this, not only the most similar pair of clusters is merged in each iteration but all pairs that meet the global similarity threshold value.



**Figure 2.2. The GeMMA workflow.** GeMMA is a protocol to cluster protein sequences based on the agglomerative hierarchical clustering paradigm. It iteratively aligns sequence clusters, generates cluster profiles, compares clusters based on their profiles and merges pairs of clusters based on the comparison results. The protocol makes use of third-party tools in all steps apart from the merging step.

### 2.2.2 Modular use of existing tools

Third-party tools are used at three points in the GeMMA workflow (see Figure 2.2). Specifically, these are MAFFT (Kato, Kuma et al. 2005) for multiple sequence alignment and COMPASS (Sadreyev and Grishin 2003) for alignment profile generation and profile comparison. This modular design is

thought to make the protocol highly transparent and flexible, unlike the ‘black box’ approaches seen in tightly integrated methods. It further ensures that GeMMA can profit from ongoing external development efforts. Not only can all tools be used in their latest versions at any given point but they can also be replaced by other tools with equivalent functions. The implementation of GeMMA as a distributed HPC protocol (see below) means that this modularity does not create substantial additional overhead. This is because the HPC implementation itself requires the splitting of the protocol into independent modules, to be executed by individual HPC jobs.

MAFFT is a suite of methods that implement different algorithms for progressive multiple sequence alignment (see Section 1.4.1.2). As of 2011, it is one of the fastest and best-performing extant alignment tools (Thompson, Linard et al. 2011). Depending on the size and other features of the sequence set to be aligned, different algorithmic refinements can be switched on and off. This allows for a flexible balancing between speed and performance. MAFFT is further regularly updated. For these reasons, it was selected for use in the GeMMA protocol.

COMPASS is a suite of tools for alignment profile generation and comparison (see Section 1.4.2). It takes two MSAs as input, from those generates two MSA profiles in the form of PSSMs, and computes their similarity. For performance reasons, the profile generation and comparison steps are separated in the GeMMA workflow (see Figure 2.2). This makes sure that a profile is generated for each individual cluster only once, even if the cluster is subsequently compared with many other clusters. The E-value scores reported by COMPASS are used to monitor the progress of GeMMA. Accordingly, the global cluster similarity threshold used in the protocol is an E-value threshold<sup>5</sup>.

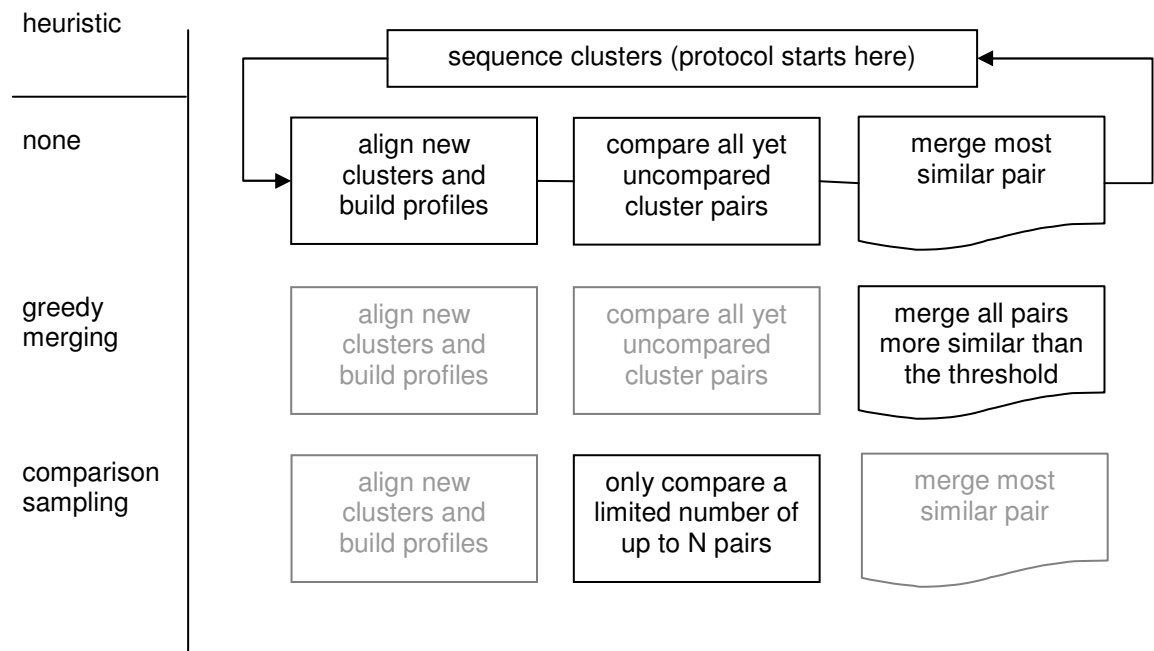
---

<sup>5</sup> While the E-value denotes the significance of a given similarity score rather than the score itself, this difference will be ignored at times in the following, for ease of reading.

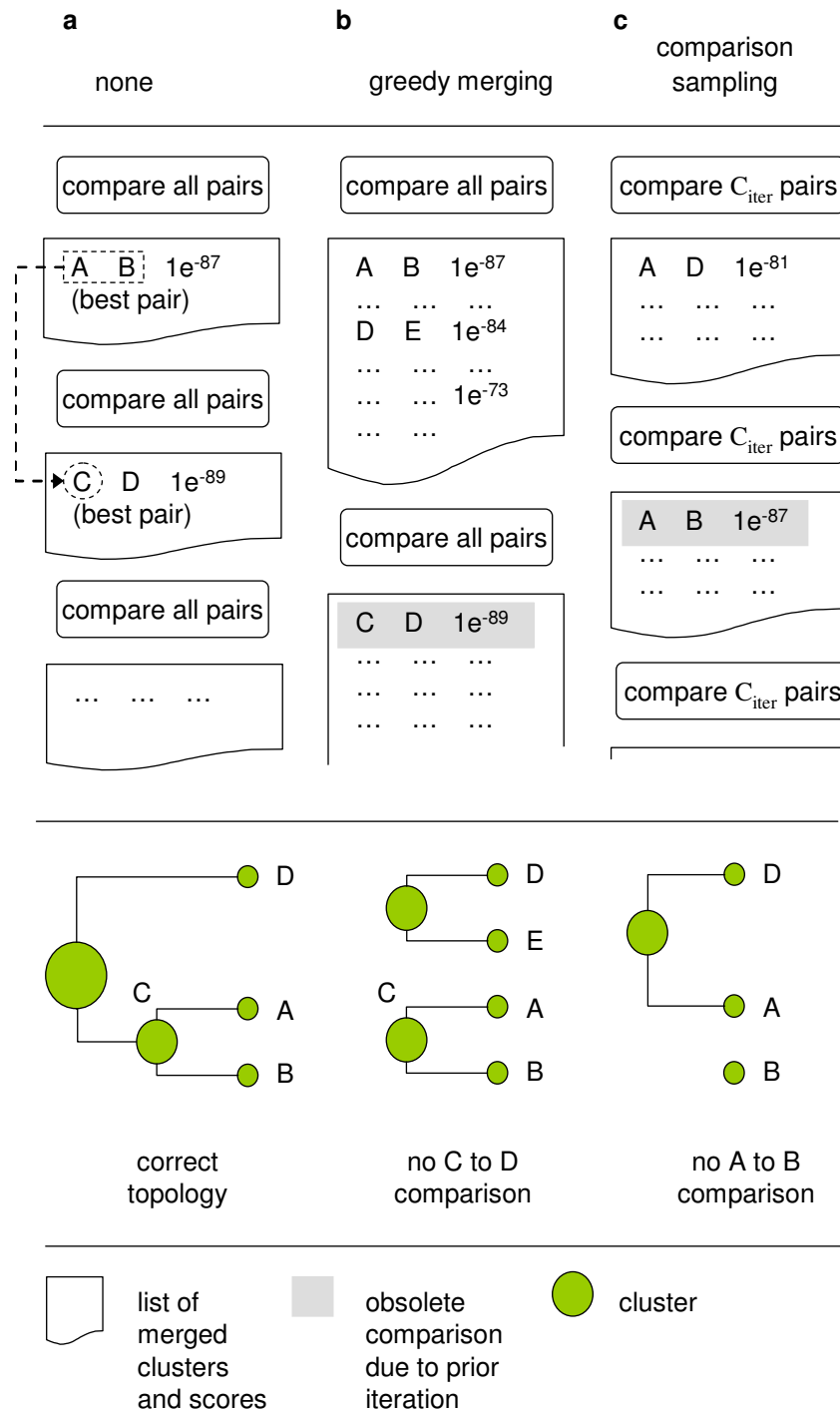
### 2.2.3 The GeMMA heuristics

The GeMMA protocol implements two heuristics to speed up the clustering of large sequence datasets: greedy merging and comparison sampling. The speed gain is achieved by increasing the number of merges and decreasing the number of comparisons per iteration, respectively. Both heuristics are based on a series of observations made for the type of dataset at GeMMA is primarily targeted. In brief, these are sequence superfamilies, each containing one or more families (see Section 1.2.2.1). First, superfamilies follow a scale-free size distribution. Second, the few large superfamilies usually contain many different families. Third, these families show relatively high degrees of internal sequence conservation whilst their average degree of similarity to each other (average level of sequence similarity) is often very low. This means that the families in large superfamilies are usually relatively well-separated. Similar properties can be found in other large yet structured datasets, in- and outside the biological realm.

The GeMMA heuristics are explained in detail in the following two sections. Figure 2.3 provides an overview of where and how they affect the clustering process. Figure 2.4 illustrates by example the effects they can have on the cluster merging order, and therefore on the resultant clustering as a whole.



**Figure 2.3. The GeMMA heuristics.** This shows where and how the introduction of the two heuristics modifies the basic agglomerative hierarchical clustering approach on which GeMMA is based. Note that the overall workflow remains unchanged.



**Figure 2.4. Potential effects of the GeMMA heuristics.** (a) In traditional agglomerative hierarchical clustering, an all-by-all cluster comparison is made and only the best matching pair merged in each iteration. In the example shown this produces the correct order of merges. (b) Using the greedy merging heuristic, all cluster pairs that meet the current cluster similarity threshold value are merged in each iteration. While cluster C is still created in the first iteration, it is never compared with its ideal match D in the following iteration, since D has already been merged with E. (c) Using the comparison sampling heuristic, no more than a randomly drawn subset of  $C_{iter}$  comparisons is carried out in each iteration. Here, A and B are not compared in the first iteration because the pair is not drawn. A is instead merged with D; hence it is no longer available for comparison to B in the following iteration.

### 2.2.3.1 Greedy merging

A cluster merging heuristic was implemented in GeMMA to reduce its time complexity. ‘Greedy’ merging means that not only the best-matching cluster pair is merged in each iteration but all pairs that match better than the global cluster similarity threshold value; cluster merging is done in order of decreasing similarity. This is to lower the number of existing clusters and therefore the number of necessary pair-wise comparisons in subsequent iterations. Based on common structural characteristics of the processed datasets (as described above), the heuristic relies on the assumption that the exact order of cluster merges – especially in early stages of clustering – should not have a great effect on the composition of the eventually derived partitioning of the input dataset. Put differently, the initially small, abundant and highly similar sequence clusters are expected to later be subsumed by the same set (or very similar sets) of larger clusters, regardless of the exact order of merges.

In the worst case example in Figure 2.4b, an existing cluster D that shows high similarity (higher than the similarity threshold value set) to a newly created cluster C is merged with another, less similar cluster E, because D and C are never compared. The latter is a direct consequence of merging multiple cluster pairs per iteration: A and B are merged to form C in the same iteration the merger of D and E is created. Hence, D does not exist anymore when it could be compared with C in the next iteration. However, based on the assumption above, this does not necessarily have to affect the clustering result: the sequences in the example clusters (A, B, D, and E) often end up in one and the same larger cluster.

Generally speaking, the larger and more diverse the clusters get in the course of clustering, the greater is the potential negative performance impact of the greedy merging heuristic. The individual impact depends on how much the

cluster merging order deviates from the ideal order, that is, the order of merges observed when only the best pair of clusters would be merged in each iteration. The detailed effects of such deviations are difficult to illustrate and discuss in theory. However, a significant performance decrease when individual partitionings of gold standard datasets (obtained with and without the heuristic) are benchmarked would clearly hint at such effects. This has so far not been observed (see Chapter 4).

### 2.2.3.2 Comparison sampling

The greatest speed bottleneck when clustering (sequence) data is the initial computation of the similarity matrix. Usually, all data points have to be compared with all others. The profile linkage strategy to compare pairs of sequence clusters, as employed in GeMMA, is even more computationally demanding. Traditional hierarchical clustering approaches can rely on the initially calculated similarity matrix over the whole course of clustering. Whenever two clusters are compared, the comparison results for the underlying data points (sequences) are reused. This is not possible when following a profile linkage approach. In this case a profile has to be generated from scratch for each newly created cluster, and the cluster profiles are compared.

Based on the above considerations, a second heuristic was added to GeMMA further to reduce its time complexity when clustering large sequence datasets. This is that only a fraction  $C_{\text{frac}}$  (a number  $C_{\text{iter}}$ ) of all yet uncomparing cluster pairs ( $C_{\text{left}}$ ) are compared in each iteration, while the remaining comparisons are ‘postponed’ (to be considered in following iterations). In addition,  $C_{\text{iter}}$  is kept within the lower and upper boundary values  $C_{\text{min}}$  and  $C_{\text{max}}$ . The default settings for  $C_{\text{frac}}$ ,  $C_{\text{min}}$  and  $C_{\text{max}}$  are 1%, 2,000,000 and 10,000,000, respectively.



Table 2.1 illustrates how the boundary rule stated above affects the number of comparisons made, by the example of the very first iteration of GeMMA for a given sequence dataset (or a set of starting clusters). In this case, the total number of non-redundant pairwise comparisons to be made  $C_{left}$ , depending on the total number of starting clusters  $N$ , is given by:

$$C_{left} = \frac{N \cdot (N - 1)}{2}$$

Note, however, that the stated rule does not apply in all further stages of GeMMA clustering, from the second iteration onwards. This is because the merging heuristic (see Section 2.2.3.1) allows for multiple cluster merges per iteration, and the comparison sampling heuristic described here continuously postpones (often the great majority of) comparisons. The combination of both heuristics make the speed of convergence increase in a manner that is dataset-dependent, and thus no simple rule to estimate  $C_{left}$  for a given iteration exists.

When using the default settings of GeMMA, the boundary values  $C_{min}$  and  $C_{max}$  only become relevant when the number of clusters to be compared all-by-all exceeds  $\sim 20,000$  (see Table 2.1). Below this value, a fixed number of  $C_{min}$  (two million) pairs are compared per iteration. Further, the number of comparisons made per iteration  $C_{iter}$  can only drop below a fraction  $C_{frac}$  (1%) of all remaining comparisons ( $C_{left}$ ) if it exceeds  $C_{max}$  (ten million); it is then set to  $C_{max}$ . This is the case when more than  $\sim 43,000$  clusters have to be compared. When the number of clusters lies between the two boundary values, a fraction  $C_{frac}$  of all remaining comparisons is made in each iteration.

**Table 2.1. The number of comparisons made in the first iteration of GeMMA clustering depending on the size of the input dataset.** For each input dataset size, the total number of non-redundant pairwise (all-by-all) comparisons is given by the formula stated in the main text. As also described there, the number of comparisons made per iteration is bounded by  $C_{min}$  and  $C_{max}$ , as set to 2,000,000 and 10,000,000 by default, respectively.

Initial number of clusters (individual sequences)	Total number of non-redundant comparisons	Comparisons made in the first iteration
1,000	499,500	499,500
2,000	1,999,000	1,999,000
5,000	12,497,500	2,000,000
10,000	49,995,000	2,000,000
20,000	199,990,000	2,000,000
30,000	449,985,000	4,499,850
40,000	799,980,000	7,999,800
45,000	1,012,477,500	10,000,000
70,000	2,449,965,000	10,000,000
100,000	4,999,950,000	10,000,000

The distinct structural characteristics of large sequence superfamilies (as described above) led to the two primary rules underlying the comparison sampling heuristic. First, all pair-wise comparisons should ideally be made before merging clusters when small to medium-sized superfamilies are processed. Such superfamilies are equally likely to be diverse or conserved in sequence and therefore should be treated ‘neutrally’. This is possible, since all-by-all profile comparisons are, in these cases, computationally feasible. Second, the heuristic should be used when large superfamilies are clustered, in early stages of clustering. In these cases, a large fraction of all comparisons will yield low similarity values. These results do not have to be readily available initially, and only a few of them become relevant later on, depending on whether or not the corresponding clusters still exist.

In general, whenever there exist pairs of clusters that are more similar than the cluster similarity threshold value set at any point in clustering, a sufficiently large, random sample of all remaining cluster comparisons is expected to reveal at least one such pair. This is sufficient to keep the iterative merging and (asynchronous all-by-all) comparison process going. In summary, the rationale behind the comparison sampling heuristic is that most pair-wise cluster comparisons can safely be postponed in early stages of clustering.

Despite the considerations above, the comparison sampling heuristic adds a further scenario potentially detrimental to the performance of GeMMA. Apart from merging suboptimal pairs of clusters owing to the greedy merging heuristic (see Figure 2.4b), the former can now also happen when the comparison of a given cluster with all others is split over several iterations. In Figure 2.4c, the best-matching cluster pair (A and B) is not compared in the first iteration; instead, A is compared with and subsequently merged with D, and thus never compared with B. Theoretically, there could be cases where the difference in the similarity values for the A:D and A:B pairs is large. To prevent a severe impact of such cases on the overall clustering outcome, GeMMA is used to cluster datasets in multiple, consecutive ‘rounds’.

Each round consists of one or more iterations and is defined by a specific setting of the cluster similarity threshold (the stopping criterion), which is decreased in a step-wise manner between rounds. Any comparison results produced that do not meet the current threshold value are stored (see Appendix A.3), to be considered in following rounds. In this manner, the initial fraction of cluster pairs with available results, out of all possible pairs of existing clusters, increases between rounds. This is true even if the comparison sampling heuristic is active (i.e., not all possible pairs are compared in each round).

The round system is thought to remediate the potentially considerable negative effects of the greedy merging heuristic in late stages of clustering, as outlined above. ‘Difficult’ merging decisions, at medium and low levels of average cluster similarity, are made on the basis of more comprehensive information than ‘easier’ decisions, made when most clusters are still small and highly similar to many other clusters. The number of GeMMA rounds and the associated range of decreasing similarity threshold settings are both decisive factors in balancing the speed gains achieved through the two GeMMA heuristics with their negative performance impacts. These settings have to be made depending on the specific purpose of clustering. For example, in the use of GeMMA for protein domain family identification, as discussed in Chapter 3, a relatively evenly distributed range of threshold settings is sampled over 10 rounds (see Section 3.3.3.2).

For the comparison sampling heuristic to work, it is of crucial importance that the fraction  $C_{\text{frac}}$  of all remaining comparisons  $C_{\text{left}}$  is a randomly drawn (and thus representative) subset. Appendix A.2 explains how this was implemented algorithmically. The heuristic could otherwise lead to situations where many comparison results were produced for some clusters in a given iteration and no results at all for others. This in turn would lead to biased merging of only clusters with available results, leading to an erroneous clustering result. There further exists a potential problem associated with setting  $C_{\text{frac}}$  too low, relative to  $C_{\text{left}}$ . If there still exist pairs of clusters in any given GeMMA iteration that match better than the cluster similarity threshold value set, but none of these pairs is compared, the respective GeMMA round is terminated prematurely. Then, the protocol continues with the next round, using a lower threshold value. The higher  $C_{\text{frac}}$  is set, the less likely this situation becomes, and the lower should be the negative performance impact of the comparison sampling heuristic in general.

## 2.3 Discussion

The problematic issue of assessing the performance of clustering methods in general, and GeMMA in particular, is addressed in the following section. Several possible improvements to the protocol are discussed subsequently.

### 2.3.1 Notes on performance and its measurement

The performance of clustering methods can, apart from an assessment of their resource usage, only be assessed in the light of a specific clustering goal. For example, the accuracy with which GeMMA clusters protein (domain) sequences is assessed indirectly in the two family identification modules of the DFX pipeline (see Chapter 4 and Chapter 5). In general, it can be expected that the use of sequence profiles in GeMMA provides advantages over traditional similarity measures that are based on pairwise similarities (e.g., average linkage) when clustering sequences with a focus on function; this is the case whenever the premise is used that sequence similarity (usually) reflects functional similarity, as done in DFX.

The above-mentioned advantages may show in two scenarios in particular. First, in early stages of clustering, two distinct functional groups of sequences may differ in only a few key residues. A profile-based method can be expected to pick up on the residue signal and therefore be able to distinguish between the two groups correctly. This may not be the case for one of the traditional ‘sequence linkage’ methods: the overall high pairwise similarities between sequences from both groups may disguise the residue signal. A second situation in which the high sensitivity of sequence profiles could be advantageous is in late stages of clustering, where clusters represent coarser functional and/or structural groups. The pairwise relationships of such coarser groups in a sequence dataset (e.g., a superfamily) may again be resolved more clearly by profile-based methods, so that the order of cluster

merges may still correctly reflect the true (wider) functional and evolutionary relationships between individual groups.

One way of testing whether these assumed advantages do exist in practice would be to cluster a range of sequence datasets with both GeMMA and the three traditional sequence linkage approaches. Using corresponding annotation data (supervised clustering evaluation) it could then be assessed for each method at which point in clustering the highest family partitioning performance is achieved. Finally, the maximum performance values for each method could be compared to assess whether one of them (GeMMA) provides a statistically significant (or even consistent) advantage.

### 2.3.2 Future work

It should be possible to improve or enhance GeMMA in different aspects. Specifically, these are the use of specific third-party tools, the removal of redundancy on the technical level, changes to GeMMA heuristics, changes to the overall protocol and the potential application of GeMMA to other data types. These possibilities are discussed in the following sections.

#### 2.3.2.1 Changes in the use of third-party tools

While the MAFFT alignment method is still one of the best and fastest in its field, it may be beneficial to replace the profile-profile comparison method COMPASS with either the HMM-HMM comparison method HHSearch (Soding 2005), which has been shown to have increased sensitivity (Soding and Remmert 2011), or with its direct successor, PROCAINE (Wang, Sadreyev et al. 2009). The latter makes use of (horizontal) residue patterns and predicted secondary structure elements in the sequences that constitute the input alignment to create profiles. It was reported that PROCAINE outperforms HHSearch with respect to remote homology detection in that

manner (Wang, Sadreyev et al. 2009). There further exist plans to add an HMM-HMM comparison tool to the widely used HMMER suite of tools. All these alternative tools would integrate seamlessly with GeMMA, as they take the same input data as COMPASS (alignments) and, just as the latter, make it possible to separate the profile (HMM) generation and comparison steps. The decision between HHSearch and PROCAINE could be based on speed. Note that, as the E-value calculation in these algorithms may differ from that in COMPASS, any workflow that uses GeMMA with a specific clustering granularity setting (E-value threshold) would have to be reassessed, that is, a novel generic setting be derived. For example, this would be the case for the unsupervised family identification method discussed in Chapter 4.

### 2.3.2.2 Further technical integration

More generally, it must be noted that there is a certain redundancy in using progressive alignment methods such as MAFFT in conjunction with profile comparison methods such as those mentioned above. In brief, this is because (i) these alignment methods construct initial ‘guide trees’ based on clustering the input sequences and (ii) an essential step in the subsequent progressive alignment process is the pairwise comparison of (sub-)alignments via intermediately constructed profiles. For speed reasons, the guide trees are constructed using traditional sequence linkage clustering approaches (see above), commonly UPGMA (see Section 2.1.2.1), and the profiles are built and compared in a heuristic manner, for example, by Fast Fourier Transformation (FFT) in MAFFT. On the other hand, profile comparison methods have to align the profiles in order to compare them.

The above considerations show that the strategies and algorithms behind progressive alignment methods, which aim to construct an accurate multiple sequence alignment, are very similar to those used in clustering methods like GeMMA, which aim to construct an accurate dendrogram (tree) reflecting the

similarity relationships in a sequence dataset. For these reasons, primarily to speed up the clustering process as a whole, it would be tempting to integrate the alignment, profiling, profile comparison and tree construction steps (aspects) of GeMMA more tightly. One such attempt was made in the hybrid multiple alignment and tree building method SATCHMO (Edgar and Sjolander 2003), which itself tries to overcome the speed bottleneck of HMM comparison by using MAFFT alignments initially, for easy-to-align sequences (Hagopian, Davidson et al. 2010). However, a fundamentally modular approach like that followed in GeMMA can be regarded as being more future-proof, for example, in the sense of the above-considered (straightforward) changes in the third-party tools used.

Other (‘softer’) integration strategies that uphold the modular structure of GeMMA (and do not require the development or integration of additional algorithms), while yielding a speed increase, could therefore be considered. For example, the partial GeMMA dendrogram that exists for each cluster (subtree) at any point in the course of clustering could be fed as a (more accurate) guide tree into MAFFT when aligning the clusters. Further, instead of aligning the sequences in each cluster created during clustering from scratch (as done so far), the recently added group-to-group (alignment-to-alignment) alignment option of MAFFT (Kato and Toh 2008) could be used. This takes two alignments as input, converts one of them to a profile and then aligns the profile to the other alignment. To make use of this option, the implications of assuming monophyly and paraphyly for the sequences in both clusters, respectively (see MAFFT website<sup>6</sup>), would have to be studied. If these are found to be in agreement with the design of the GeMMA clustering process (which is probably the case), the only remaining decision would be which of two clusters is treated as (the existing) alignment and which as (the

---

<sup>6</sup> <http://mafft.cbrc.jp/alignment/software/>



added alignment) profile, respectively. An intuitive solution would be treating the larger of two clusters as the existing alignment in all cases.

### 2.3.2.3 Changes to the GeMMA heuristics

The GeMMA heuristics could be made more (and explicitly) flexible with regards to the stage of clustering, which may lead to increased performance in some cases. This could be achieved, for example, by setting the respective parameters (more) dynamically, depending on the average similarity values observed. A similar thing is already done implicitly, based on the number of comparisons yet to be made at any one point; this is in conjunction with the comparison sampling heuristic (see Section 2.2.3.2). A brief worked example illustrates that the impact of this heuristic becomes smaller (and accuracy can be expected to be higher) in late stages of clustering, where more difficult merging decisions have to be made. This is achieved by using a bounded (and thus non-linear) function for the number of comparisons made per iteration. Whenever the total number of remaining cluster comparisons to be made exceeds a certain threshold (lower boundary; 2,000,000), at any point in clustering, only a fraction of all comparisons is made. This fraction is initially large (100%) and slowly decreases with the number total comparisons increasing. This is because the number of comparisons made is set to a constant number: the stated lower boundary value. However, if the number of total comparisons exceeds 200,000,000 (100 times the lower boundary), the number of comparisons made is set dynamically, to 1% of all comparisons. Finally, if a level of one billion comparisons is exceeded, the number of comparisons made is capped, at a fixed level (upper boundary; 10,000,000).

Additional heuristics may also be added to GeMMA. Especially interesting is the idea of using so-called ‘pivot points’ when hierarchically clustering inherently structured datasets (Kull and Vilo 2008), such as sequence superfamilies. This strategy shares with GeMMA the general heuristic of not

generating a complete all-by-all similarity matrix prior to clustering. In contrast to GeMMA, however, the initial matrix is also not extended in the course of clustering: the whole dataset is clustered based on an initially generated partial similarity matrix. The distance between any two clusters is then measured based on only the known distances between the data points they contain. As no such individual distances are calculated when using a profile linkage approach, the strategy followed by Kull and Villo could not be applied in the case of GeMMA. However, the selection procedure that is used to identify an appropriate subset of pairwise comparisons (to populate the partial similarity matrix), via so-called pivot points, could inspire a similar strategy in GeMMA, where a subset of all comparisons is currently sampled randomly in each round.

In brief, the pair selection strategy in Kull and Vilo (2008) works as follows. A limited range of pivot data points (e.g., sequences)  $N$  is initially chosen randomly from the dataset, and these are compared with all remaining data points. This corresponds to an ‘ $N$ -by-all’ approach. The distance between any two non-pivot data points can then be estimated from the similarity of their  $N$ -dimensional pivot distance vectors, their ‘pseudo-distance’. If two data points show a similar pattern of pivot point distances and their pseudo-distance is thus small, their true distance can also be expected to be small. For each pair of data points with a pseudo-distance smaller than a certain threshold level, the true distance is therefore measured. In addition, the distances for a certain number of randomly selected pairs are measured as well. The calculated distances are used to populate the (partial) similarity matrix. The reasoning behind biasing the matrix towards shorter distances (similar data points) is that these are important in the early stages of the following hierarchical clustering step and impact the whole clustering dendrogram, while only a few long-distance relationships have to be known to make correct merging decisions in later stages (Kull and Vilo 2008).

In the case of GeMMA, the above-described pivot point paradigm could potentially be used in a similar way, but iteratively. The basic strategy, which can most easily be understood when picturing the dataset as a similarity network (with the nodes being the clusters and the edges their pairwise distances), would be the following. For a given iteration of the protocol, a set of  $N$  pivot clusters is first selected. In the first iteration, this is done randomly. The pivot clusters are compared with all remaining clusters, respectively. A certain number of cluster pairs are then selected for comparison, based on their calculated pseudo-distances (see above). As these pairs can be expected to be among the most similar pairs in the dataset, they are likely to meet the cluster similarity threshold set and, therefore, to be merged in the same iteration. All steps from this point onwards would be novel with respect to the method described above. After merging some cluster pairs, all newly created clusters are compared with the pivot clusters. In the following iteration, a new set of pivot clusters is selected. This is done in a manner that aims successively and evenly to explore the structure of the dataset, as follows. Let  $PD_{\min}$  be the distance a cluster exhibits to its closest pivot cluster. All clusters that still exist and were not pivot clusters in the last iteration (including newly created ones) can then be sorted by their  $PD_{\min}$  values, from highest to lowest. The first data point in the list, for example, is then the point with the strongest ‘outlier character’ relative to the original set of pivot points. Therefore, the first  $N$  data points in the list are chosen as the new pivot points, which are subsequently compared with all other data points, and so forth.

Regardless of whether or not the outlined strategy could be implemented in full, and whether or not it would work well, the general shift from selecting subsets of comparisons randomly to selecting them in a more directed manner (biased for even coverage) warrants further investigation.

#### 2.3.2.4 Changes to the protocol as a whole

The GeMMA protocol could also be changed in a more fundamental way, by implementing a medium-sensitivity, non-profile approach (for example, one of the traditional sequence linkage methods) to be used in early stages of clustering, that is, when many clusters are still highly similar. The high-sensitivity, profile-based method could then be used later on. This would require the initial generation of a pairwise sequence similarity, for example, using BLAST. Specifically, each sequence in the dataset would have to be compared with all other sequences that do not share the same starting cluster. Particularly when the starting clusters are small (they may contain single sequences) and the processed dataset is large, calculating such a full matrix can take up considerable (HPC) resources (and thus be slow; as seen in common hierarchical clustering methods). However, the subsequent speed advantage over using the profile-based method could make this a good investment.

Up to a certain point, the implemented sequence linkage method could cluster the dataset without creating any new entities (such as profiles) or performing any further comparisons. This is the great speed advantage of sequence linkage methods over the profile-based method. For this advantage, the GeMMA heuristics could potentially be completely deactivated at this stage, which may effectively compensate for the loss in profile-based sensitivity. Both the feasibility of generating very large initial matrices (in the case of large input datasets) and the factual loss in performance by not using the profile-based approach initially would have to be studied in detail before making such changes to the protocol, to avoid circularity: the use of profiles and the use of heuristics are parts of GeMMA due to the (assumed) lower sensitivity of pairwise comparisons and the resource challenge posed by calculating (very) large similarity matrices.

### 2.3.2.5 Potential use with other types of data

A further interesting point about the above-cited work by Kull and Villo (see Section 2.3.2.3) is that it underlines how flexibly a clustering algorithm, once established, can be used. Just like their method, GeMMA could relatively simply be adapted for clustering other biological and non-biological data types, for example, expression profiles or (online) documents. In such scenarios, it is primarily the similarity measure that changes, while the implemented heuristics are still valid and the technical implementation remains unchanged.

## **Chapter 3. The DFX pipeline: identification of functional families within protein domain superfamilies**

It has been established in Chapter 1 that the protein domain superfamily is the most appropriate framework to study protein sequence evolution on a large scale. Often, the emphasis in such endeavours lies on the evolution of overall protein structure, both on the levels of domain architecture and tertiary structure (fold). While concepts do exist to classify and study protein sequence space in a more fine-grained way, with a focus on function, these have not yet been consistently applied on the domain level. This claim is based on three observations. First, there exist both (structure-based) domain superfamily and (function-based) domain family resources. However, the consistent integration of both levels into a single resource is rarely seen. Second, in the few cases where this has been done, the respective resources are either meta-resources, integrating foreign data, or make heavy use of manual curation in the family identification process; the latter means that different families will inevitably be ‘treated’ differently. Third, despite the fact that the notion of (conserved) domain function is an observed and well-known biological phenomenon, which is also implicitly presupposed by these resources, whole-protein function frequently governs the domain family identification process to a (too) large extent.

Adding to the above, there still exists a tendency (in the literature and in protein research as a whole) to focus on protein domains when structure is analysed and on the sequences as a whole when function is the main interest. This may not be surprising, given that important evolutionary concepts that are frequently used in studying protein function on the whole-sequence level, such as orthology and paralogy, cannot be readily applied on the domain level. While the opposite has repeatedly been argued (Fitch 2000; Song, Sedgewick

et al. 2007; Song, Joseph et al. 2008; Nagy and Patthy 2011), namely that these concepts are, in fact, *more* appropriate to use on the domain level, doing this may cause considerable confusion and follow-up problems. Moreover, it is not necessary.

A distinct set of terms that describe domain evolution by means of duplication and shuffling is at hand and widely agreed on (see Section 1.1.2). Further, it has long been noted that many types of protein domains are conserved functional units and have ‘promiscuous’ character, in the sense that these sequences appear in proteins with variable domain architecture and overall protein function (see Section 1.1.2). Within the latter, they fulfil a certain partial function. The overall function of multi-domain proteins can therefore often be discerned in a logically straightforward manner from the combination of domains it contains (Bashton and Chothia 2007; Forslund and Sonnhammer 2008). However, this may not always be the most interesting question: the evolution of *domain* function, in the context of different types of parent proteins, is hitherto studied much less.

To study the function of proteins and protein domains on the domain level, the Domain Family eXploration (DFX) pipeline was developed. This integrates large-scale sequence clustering with GeMMA, as discussed in the above chapter, with both unsupervised and supervised post-processing protocols to identify families of protein domains. This chapter describes the family concept followed by DFX, the overall architecture of the pipeline, and the implementation of all common steps in the workflow. The two core family identification protocols developed for DFX represent alternative routes, depending on the availability of function annotation data. Therefore, they are first introduced and contrasted in the present chapter, and subsequently described in detail in the two below chapters, respectively. Similarly, a discussion of those components of DFX that are common to both protocols,

and their observed performance, is found at the end of this chapter, while the results that can be achieved with either strategy are described in a qualitative way in the two following chapters. Chapter 6 contains a quantitative, large-scale comparison of the two DFX family identification protocols. Chapter 7 contains a concluding discussion of the DFX pipeline as a whole, in the context of protein domain research.

### 3.1 Background

Many studies make use of existing, manually curated family resources to define ‘functional families’ of proteins or protein domains. That is, they (explicitly or implicitly) treat the families defined by these resources as families with conserved function. Alternatively, there exist different automatic methods and protocols that directly split arbitrary sequence datasets into families. In this case, the respective sequences are usually known to belong to the same superfamily. Studies using such family information have varying aims. For example, to annotate proteins, to measure functional enrichment in proteins from specific (meta)genomes, or to study the evolution of protein (domain) function in the context of specific superfamilies. An overview of existing resources and methods for identifying protein and protein domain families is provided in the following.

#### 3.1.1 Existing family resources

All commonly used protein and protein domain family resources are based on the model library concept (see Section 1.5.2). In addition, they all use manual curation, if in different steps and to varying extents. There are usually two distinct layers of models defined, following the superfamily and family concepts (see Section 1.2.2). However, the naming of the respective layers is highly variable among the resources; for example, a ‘subfamily’ layer in one resource may correspond to a ‘family’ layer in another. Depending on the



resource, the layers also capture different degrees of conservation in protein sequence, structure and function, respectively; for example, the ‘family’ layer in one resource may capture a different level of functional similarity compared with a layer with the same name in another.

Examples of entirely sequence-based two-layer model libraries are the PANTHER subfamilies and families (Thomas, Campbell et al. 2003), the TIGRFAMs subfamilies and superfamilies (Haft, Loftus et al. 2001) and the PhyloFacts subfamilies (or ‘books’) and families (Krishnamurthy, Brown et al. 2006). Pfam has also recently introduced a second layer above the family level, dubbed Pfam ‘clans’ (Finn, Mistry et al. 2006). This superfamily-like layer is not itself represented by a library of models, however, and is established based on the existing family models and the underlying sequences. In particular, remote homology relationships between two or more Pfam families are established by different types of evidence and manual curation. The primary sources of evidence are the pairwise comparison of family models, the detection of structural similarities between proteins in different families, and the analysis of cases in which individual sequences match the models of different families similarly well. The clan concept as a whole is similar to an earlier effort to combine Pfam families into superfamilies in the SUPFAM resource (Pandit, Gosar et al. 2002).

SCOP defines protein domain superfamilies, like CATH, but has also established a family level. SCOP families are defined as clusters of (structurally known) sequences within SCOP superfamilies that fulfil stricter criteria for sequence and function conservation (at least 30% sequence identity or significant functional and/or structural conservation) than those applied on the superfamily level (Murzin, Brenner et al. 1995). While the SUPERFAMILY resource assigns sequences without known structure to both

SCOP superfamilies and families, it only contains a single layer of models: those for the superfamily level.

The assignment of sequences to SCOP families in SUPERFAMILY is based on a hybrid approach (Gough 2006). In this, the model(s) for the SCOP superfamily to which the query sequence belongs serve(s) as a ‘bridge’. For all seed sequences that underlie a specific SCOP superfamily model there exist pre-calculated scores and alignments with that model, respectively. Such can also be produced for the query sequence and each of the superfamily models. By combining this information, a query sequence can effectively be aligned and compared with all SCOP seed sequences of its superfamily. Since each of the seed sequences is assigned to a manually curated family in SCOP (see above), the query sequence can then inherit the family assignment of the seed sequence it matches best.

Like Pfam and SCOP, both PANTHER and TIGRFAMs rely on manual curation, specifically when splitting superfamilies into families (Haft, Selengut et al. 2003; Thomas, Campbell et al. 2003). Further, Pfam and TIGRFAMs manually refine the sequence composition of their seed alignments and the alignments themselves before building models, and curate the model-specific detection thresholds (Haft, Selengut et al. 2003; Finn, Tate et al. 2008). The PhyloFacts resource splits its protein and protein domain superfamilies into families using the unsupervised SCI-PHY algorithm (Krishnamurthy, Brown et al. 2006); the latter is discussed in detail in Section 4.1.2.1. It then builds subfamily models according to the protocol described in Brown, Krishnamurthy et al. (2005). A limited amount of manual curation is involved in assessing the global homology of multi-domain proteins.

While the Pfam and SUPERFAMILY resources can in principal reach the same taxonomic and proteomic coverage as the underlying primary sequence and structure databases, the other resources mentioned can not. This is

because they build models based on limited sequence datasets, in an interest-driven way; for example, to cover the human proteome, eukaryotic proteomes or families of high medical interest.

Both InterPro (Hunter, Apweiler et al. 2009), provided by the European Bioinformatics Institute, and the NCBI Conserved Domain Database (CDD) (Marchler-Bauer, Lu et al. 2011) are meta-resources that try to approach the family assignment (granularity) problem by integrating the information from other databases into multi-layer model libraries. InterPro contains both protein and domain families, while the CDD concentrates on the domain level only. In both resources, the number of defined layers varies with every protein and/or domain family identified. From a technological point of view, InterPro mainly integrates resource-specific HMMs and uses HMMER (Eddy 2009) for library scans, while the CDD converts all models it integrates into PSSMs and uses RPS-BLAST (Altschul, Madden et al. 1997).

Among the currently eleven InterPro member databases are the most important resources for protein and domain (super)family classification (see above and Section 1.5.2.2). These databases define family models at varying levels of granularity (see above) and contribute them to InterPro. The sequence coverage of all models is then matched by searches against UniProtKB, based on which they are manually integrated into two types of meta-families (InterPro entries), using the following naming conventions: ‘family’ (protein level) and ‘domain’ (domain level). All models subsumed in a family entry are required to cover all domains in the underlying sequences and span at least 80% of their length. Entries of the type domain are required to have adjacent (or surrounding) other entries (see below), for example, a family entry. The grouping of different domain models into a single entry is an entirely manual process, with no further formal constraints.

InterPro also defines two types of relationships between individual entries. These are ‘parent/child’ and ‘contains/found in’ relationships. The former implicitly define hierarchical layers of increasing granularity for both (the top-most) family and domain entries. Specifically, 75% of the sequences covered by a child entry must also be covered by the parent entry, and each sequence covered by a parent entry must be covered by only one of its children. In contrast to these ‘vertical’ relationships between InterPro entries, ‘contains/found in’ relationships constitute a ‘horizontal’ hierarchy of sequence elements. For example, domain entries can (and should always) be ‘found in’ family entries, that is, designate specific parts of proteins.

The CDD integrates domain family models mainly coming from five resources: Pfam, SMART, COGs/KOGs (Tatusov, Fedorova et al. 2003), TIGRFAMs, and NCBI ProtClusDB (Klimke, Agarwala et al. 2009). In addition, the NCBI manually curates specific, structurally defined domain families (CDs or Conserved Domains), which are added to the CDD data pool. Similarly to InterPro, the CDD combines different domain models into individual entries, which are dubbed ‘superfamilies’. Note that this is slightly misleading, as these entries primarily represent collections of models with overlapping scope (just as the InterPro entries), not a specific layer of (sequence conservation) granularity. The CDD model grouping procedure also largely matches that of InterPro. In brief, the NCBI Entrez Protein database (instead of UniProtKB) is scanned with all models and the overlap in the hit sequences is assessed.

Apart from creating combined, non-redundant entries for domain family models from different source databases, the CDD curators also create a ‘domain family hierarchy’ for each of the NCBI-curated domain models (CDs), respectively. This means that the sequences in the respective domain families (which can be superfamilies in terms of evolutionary scope) are

manually subdivided, in each case using a tree-like hierarchy with a variable, family-specific number of granularity levels. The nodes at each level of the tree correspond to mutually exclusive sequence groups ('subfamilies'), for which models are created ('child models' of the CD 'parent' model). The rationale followed in creating these hierarchies is the following. Underlying each CD is as a set of domain sequences that share a common ancestor domain (in an ancestral parent protein), a core set of shared residues, and a shared overall function. The different subfamily layers are created primarily based on putative domain (or parent protein) duplication events identified in phylogenetic trees, which are created based on curated multiple (domain) sequence alignments. Therefore, the corresponding domain subfamily models are expected to represent evolutionary subgroups, with distinct phylogenetic distribution, functional specificity and (additional) conserved residues. Notably, to constrain the granularity range of the different domain family hierarchies created to some extent, the CDD curators aim only to create layers based on domain duplication events that occurred ~0.5 billion years in the past or earlier (Marchler-Bauer, Anderson et al. 2005).

### 3.1.2 Automatic methods and protocols

When a set of manually curated seed groups is not available to establish families in a 'bottom-up' manner, as, for example, in Pfam, automatic protocols to split sequence datasets into families ('top-down') can be used instead. Such protocols generally involve a clustering step, for which different clustering algorithms are used. More importantly, they follow different unsupervised and supervised clustering evaluation strategies to estimate at which level of clustering granularity the obtained clusters best correspond to

functional families<sup>7</sup>. Both clustering algorithms and evaluation strategies are discussed in detail in Chapter 2.

In principle, any type of sequence clustering method can be combined with any clustering evaluation strategy to constitute a protocol for automatic family identification. If an unsupervised evaluation strategy is used, the protocol as a whole has *ab-initio* character. If a supervised strategy is used, the protocol uses external (annotation) data either directly, in the family identification process, or indirectly, in a training step. Family identification approaches can further be classified into integrated methods (individual pieces of software that seamlessly integrate the clustering and clustering evaluation steps) and multi-step protocols or workflows, which keep the two steps separated. Note that a sequence clustering method alone, without a corresponding evaluation strategy, cannot be regarded a protocol for family identification.

Different existing unsupervised methods and protocols for automatic family identification are discussed along with the DFX unsupervised family identification protocol in Chapter 4. Accordingly, Chapter 5 reviews existing supervised protocols.

## 3.2 Concepts

In the following two sections, the relationship between individual protein domains and whole-protein function is discussed first. Based on this, the domain family concept followed by the DFX pipeline is subsequently defined.

---

<sup>7</sup> Note that none of the automatic family identification methods discussed here clearly defines the sequence family concept it follows (an example for such a definition can be seen in Section 0).

### 3.2.1 The domain to function relationship

The traditional concepts of orthology and paralogy (see Section 1.2.1.2) that often form the basis of grouping whole-protein sequences with a focus on function cannot be readily applied on the domain level (see Section 1.2.3). The overall function of a protein is the result of its domain architecture and the mutual structural arrangement of the respective domains. Evolution principally acts on the whole protein (gene), not the domain level. Therefore, the domains found in multi-domain proteins cannot always be expected to represent entirely autonomous functional units. Rather, different domains can contribute to a protein's overall function in an orchestrated way. Conserved functions can still be derived for many types of domains, both manually (Vogel, Teichmann et al. 2005; Bashton and Chothia 2007) and through the use of specific algorithms (see Section 5.1.2). The conservation of (basic) domain function is especially obvious in the case of promiscuous domains (see Section 1.1.2) that often stem from large, evolutionarily old domain superfamilies.

It follows from the above considerations that changes in overall protein function will usually be reflected in all parts of a protein's sequence and structure. However, the extent of this signal can be highly variable over the length of the sequence, owing to the presence of structurally and functionally conserved domains. In conjunction with the different degrees of functional autonomy that are observed for different types of protein domains, any system to establish functional domain families must come with a clear definition of what 'functional' means in the context of the families it produces. Specifically, such a system must focus on either the conservation of whole-protein function or the conservation of domain function. An operative domain family concept

There exist three established concepts to partition protein (domain) sequence space, as discussed in detail in Section 1.2.2. In brief, classified by the expected level of function conservation within the resulting partitions, these are (i) the ‘broad’ superfamily concept, (ii) the ‘narrow’ orthologue cluster concept and (iii) the ‘intermediate’ family concept. The family concept is the only logical choice when studying protein function on the domain level. This is due to the great functional diversity that is expected and observed within protein superfamilies on the one hand and the incompatibility of the orthology concept with a protein-domain centric view on the other (see Section 1.2.3).

In its aim to identify functionally conserved domain families, the DFX pipeline focuses on domain function, not whole-protein function (see Section 3.2.1). With respect to conservation, it principally follows the protein family concept introduced in Section 1.2.2.3. According to this, families allow for a limited degree of functional variability in their member sequences. For example, this can refer to substrate specificity. Based on these considerations, an individual domain family would ideally only include sequences that are functionally identical or highly similar, that is, responsible for identical or highly similar partial functions in the respective parent proteins. Importantly, according to this definition, domains can be grouped into the same family even if the respective parent proteins are not homologous (over their entire range) and only share a certain partial function, whilst differing in overall function.

Following from the DFX family concept, as introduced above, the following general rules should apply in domain family identification. First, subtle variation among closely related proteins in overall protein sequence and function, owing to whole-protein evolutionary events (speciation and gene duplication), should not lead to a separation of corresponding domains in these proteins into different families (within their superfamilies). This is in



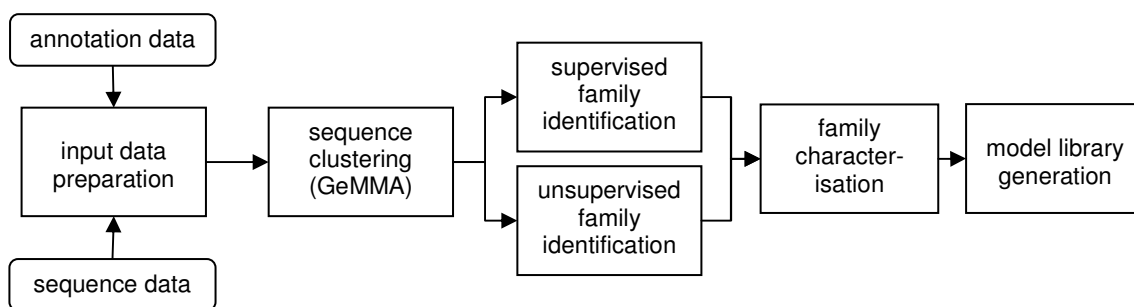
agreement with the protein family concept, as outlined above. Second, if the overall function of closely related proteins is altered by events of domain evolution (e.g., domain gain or loss) but the domain under analysis *D* remains shared ('stable'), the respective *D* domain sequences should still share the same family. Third, if the overall function of related proteins changes owing to changes in the sequence and structure of a specific domain *D*, and if this change is not just subtle (see above), the different domain *D* sequences should be grouped into different families. The assumption that ties these rules to those that apply for protein families is that, normally, gradual functional change is a result of whole-protein evolution, whereas radical functional change or added functionality is a result of changes in protein domain architecture.

### **3.3 Implementation**

The following sections describe the implementation of the DFX domain family identification pipeline. The architecture of the pipeline as a whole is first outlined. This is followed by a brief outline of its implementation on the technical and conceptual levels. The rest of this section discusses the consecutive modules of DFX in detail.

#### **3.3.1 The DFX pipeline**

DFX is a pipeline for the identification of functional families within protein domain superfamilies. Its design follows the generic model library concept as outlined in Section 1.5.2. In brief, the functional families in a given superfamily are identified using one of two developed protocols, based on which a family model library for the superfamily is established. This library can then be used to discern the family membership of arbitrary sequences in the superfamily. The core steps of the pipeline are shown in the workflow diagram in Figure 3.1.



**Figure 3.1. The workflow of the DFX pipeline.** DFX is a pipeline to identify families within protein domain superfamilies based on the model library concept. It starts with collecting and preparing the sequence data and further, optional data. The sequence data is pre-clustered to obtain a set of starting clusters. This set is then hierarchically clustered using GeMMA. Depending on the availability of annotation data, families are then identified in either supervised or unsupervised manner. This is followed by family naming and taxonomic characterisation. Finally, the model library is generated, along with model-specific thresholds.

DFX uses different types of input data. While only the domain sequence data itself is essential to process a superfamily, much better results can be obtained when additional, associated data is provided with the sequences. In particular, this refers to high-quality function annotation data, naming and taxonomic data. All three are only available for the whole protein level<sup>8</sup>. As part of the data preparation process (see Figure 3.1), all sequences in the processed superfamily are initially clustered at a high level of similarity, in a fast but low-sensitivity manner. This results in a set of ‘starting clusters’. If function annotation data are available for the processed superfamily, the pipeline commences in supervised mode. In this case, all unannotated starting clusters (those that do not contain at least one annotated member sequence) are excluded from further processing at this point. If annotation data are not available, DFX runs in unsupervised mode.

In the clustering step of DFX, the high-sensitivity sequence clustering method GeMMA (see Chapter 2) is used further to cluster the set of starting clusters, until only a single cluster remains. Based on the produced clustering dendrogram, a set of functional domain families is then identified using one

<sup>8</sup> For ease of reading, these terms will be used as if they would apply to protein domain sequences in the following, unless otherwise stated.

of two protocols, depending on whether DFX is running in supervised or unsupervised mode (see above); this is the key step in the pipeline.

In the subsequent step, the identified families are characterised. This includes family naming and taxonomic characterisation, given the respective types of input data that are available. Finally, multiple sequence alignments, models and corresponding detection thresholds are derived for all identified families. In addition, a family dendrogram that depicts the relationships between the families is generated. This can be enriched with further, family-associated data, if this is required in the context of specific studies.

Two alternative protocols to identify protein domain families (see Figure 3.1) form the core of the DFX pipeline. These are covered in Chapter 4 and Chapter 5, respectively. For either protocol, the GeMMA clustering results form the main input. In the supervised protocol (DFX<sub>super</sub>), families are identified in the clustering dendrogram based on the initially compiled annotation data. The unsupervised protocol (DFX<sub>unsuper</sub>) uses a generic clustering granularity setting that is derived by training on a gold standard dataset, to identify families using the dendrogram alone<sup>9</sup>.

The use of DFX for a given domain superfamily can be summarised as follows. The main input is the domain sequence data. To process the superfamily in supervised mode, associated protein annotation data are required as well. Additionally, protein naming and taxonomy data are necessary to characterise the produced families. The main output of DFX comprises a library of domain families, each with a name, a full alignment and a model that should recognise its known and unknown member sequences. The following sections describe the common steps of the DFX workflow,

---

<sup>9</sup> Both family identification protocols therefore include a supervised component: post-processing and training, respectively. They will still be referred to as supervised and unsupervised protocols below, for ease of reading.

while the two alternative protocols used for family identification are discussed in detail in Chapter 4 and Chapter 5, respectively.

### 3.3.1.1 Technical implementation

On the technical level, DFX is currently implemented as a complex but flexible pipeline, consisting of more than 20 interrelated and hierarchically interacting Perl scripts, modules and third-party tools. The DFX clustering module GeMMA requires an HPC system. In both the local and HPC stages batches of superfamilies are processed in a parallel manner, using a hierarchical system of UNIX jobs, job identifiers and job control. The clustering module is highly configurable and has already been used in different HPC environments, controlled with Sun (now Oracle) Grid Engine and the Portable Batch System (PBS) (Wang, Korambath et al.), respectively; among those was the UCL Legion facility (Lee, Rentzsch et al. 2010). Attempts have also been made to cluster in the Amazon EC2 compute cloud<sup>10</sup> (Ostermann, Iosup et al. 2010). To this end, the StarCluster<sup>11</sup> package was used to build virtual SGE clusters of up to 100 work nodes in EC2. However, primarily owing to a persistent bottleneck in inter-node communication and relatively frequent node instabilities, the cloud could so far not be efficiently used on a large scale.

In addition to the third-party tools used by GeMMA, MAFFT and COMPASS (see Section 2.2.2), DFX itself uses further existing software. In particular, these are the heuristic sequence clustering tool CD-HIT (see Section 2.1.4) and the HMMER suite of tools (Eddy 2009) for building and handling sequence profile HMMs (see Section 1.4.2). CD-HIT is used as a fast means to pre-cluster the sequences found in a given superfamily, at a similarity level where high sensitivity is not required (see Section 3.3.3.1). HMMER is used to

---

<sup>10</sup> <http://aws.amazon.com>

<sup>11</sup> <http://web.mit.edu/stardev/cluster/>

generate the family models and model-specific thresholds, and to perform scans against the model library (see Section 3.3.6).

On the conceptual level, DFX uses the project paradigm. This is implemented using hierarchical directory structures in conjunction with both default and project-specific configuration files. The data generated in each project are kept separately, leveraged by project-specific data and working directories. DFX further differentiates between ‘projects’ and ‘mappings’: the former are used when the whole pipeline is run and the family model libraries are created, the latter are used when (novel) domain sequences are scanned and assigned to the existing families. This makes it straightforward to organise and maintain the data generated for different versions and/or types of domain (super)family databases, such as Gene3D or Pfam, and in assigning different collections of target sequences to families (for example, domains detected in newly sequenced genomes and metagenomes).

### 3.3.2 Input data preparation

The most important types of input data for the DFX pipeline are protein domain sequence data and protein function annotation data. Large-scale domain sequence data are provided by resources such as Gene3D, SUPERFAMILY and Pfam (see Section 1.5.2). The most comprehensive resource that stores and curates high-quality protein annotation data is UniProtKB; specifically, the UniProtKB Gene Ontology Annotation (UniProtKB-GOA) database. DFX is currently used to identify functional families within Gene3D domain superfamilies, with the help of GO annotation data from UniProtKB-GOA. Further, DFX uses protein naming and taxonomic information from UniProt to characterise the produced domain families.

The GO protein annotation data used in the DFX pipeline are retrieved and pre-filtered for high-quality annotations. This is done once for each large-scale family identification task (for example, different releases of Gene3D), before running the pipeline. The unfiltered UniProtKB-GOA gene association file, which contains all available GO annotations for proteins in the SwissProt and TrEMBL parts of UniProtKB, is retrieved from the GO FTP website<sup>12</sup>. A filtered version of this file is then produced that retains (i) all non-IEA (manually derived or curated) GO annotations to proteins in SwissProt and TrEMBL as well as (ii) all IEA annotations to proteins in SwissProt that were made using either the SwissProt Keyword2GO (Camon, Barrell et al. 2005) or the EC2GO (Hill, Davis et al. 2001) mapping methods. Both the latter IEA annotation transfer methods were shown to exhibit between 70% and 100% accuracy in benchmarking (Camon, Barrell et al. 2005) and, owing to the restriction to SwissProt proteins, primarily represent a ‘translation’ of manually curated SwissProt keyword and EC annotations to GO annotations, respectively.

For a given superfamily, all domain sequences are retrieved from the Gene3D database and stored in a single FASTA file. The FASTA sequence headers in this file contain the protein sequence identifier and the domain coordinates for each domain sequence, respectively. An annotation file is then written that maps all proteins which have one or more domains in the superfamily and are associated with at least one high-quality GO annotation in the filtered UniProtKB-GOA gene association file (see above) to their annotations. This requires a mapping from the whole-protein sequence identifiers of Gene3D (sequence MD5s) to UniProtKB accession numbers, which is done using the Gene3D database. In addition, the species taxon identifiers (taxon IDs; from UniProt Taxonomy) and the names of all UniProt proteins with domains in the superfamily are written to protein species and name files, respectively.

---

<sup>12</sup> <ftp://ftp.geneontology.org/pub/go/>

### 3.3.3 Sequence clustering

DFX makes use of sequence clustering at two points. First, it uses a fast but low-sensitivity clustering method to pre-cluster the input sequence dataset in the data preparation step. Subsequently, the produced starting clusters are further clustered using a high-sensitivity method. Both stages are described in the following.

#### 3.3.3.1 Pre-clustering

As the most important step of data preparation, all sequences in the processed superfamily are pre-clustered at a maximum pair-wise sequence identity level of 60% using CD-HIT (Li and Godzik 2006). This reduces the number of initial data points (starting clusters) for the high-sensitivity sequence clustering step, which would otherwise comprise the individual sequences. The consequence is a reduced running time of the pipeline as a whole. At the same time, previous work (Addou, Rentzsch et al. 2009) has shown that a threshold of 60% sequence identity on the domain level is sufficiently conservative to ensure the functional purity of the great majority of starting clusters. Since domains with different function (or partial function, in the context of their parent proteins) are not usually mixed at such high levels of sequence similarity, using a fast but low-sensitivity clustering tool like CD-HIT for pre-clustering seems justified.

When DFX runs in supervised mode (annotation data for the processed superfamily is available), the starting clusters produced by pre-clustering are processed further. First, the set of starting clusters is filtered for unannotated clusters, that is, clusters without at least one annotated member sequence. Note that this filtering usually reduces the number of starting clusters in the sequence clustering step by at least 75% in the case of large superfamilies; the reduction rate generally depends on the degree of sequence diversity within

the processed superfamily and the annotation status of the proteins harbouring a domain in that superfamily. The therefore greatly reduced number of necessary pair-wise cluster comparisons is the reason for the considerable speed increase achieved when large superfamilies are processed in supervised (as opposed to unsupervised) mode.

The rationale behind the above-outlined filtering process is that unannotated (starting) clusters cannot be assessed in the supervised family identification procedure (see Chapter 5). It is much more efficient to attempt to assign the unannotated sequences in these clusters to the identified functional families later on, through scanning with the family model library; this matches the procedure followed to assign *novel* superfamily member sequences to families. Any sequence that cannot be assigned to a family in this manner would then either belong to a yet unknown family (with no single functionally characterised member) or represent a yet uncharacterised outlier member of a known family in the superfamily.

### 3.3.3.2 Hierarchical clustering

The high-sensitivity HPC sequence clustering method GeMMA (see Chapter 2) is used further to cluster the starting clusters generated in the pre-clustering step, until only a single cluster remains. In particular, the clustering process is split into ten consecutive rounds (executions of GeMMA), corresponding to ten decreasing settings of the clustering granularity threshold. The latter controls the maximum similarity of any two clusters in the produced partitioning, and is expressed as an E-value. The ten threshold settings used in DFX are  $10^{-80}$ ,  $10^{-70}$ ,  $10^{-60}$ ,  $10^{-50}$ ,  $10^{-40}$ ,  $10^{-30}$ ,  $10^{-20}$ ,  $10^{-10}$ ,  $10^{-05}$  and 100. The output set of clusters produced in each round forms the input set of the subsequent round, respectively. This multi-round strategy is necessary for the GeMMA heuristics to work (see Section 2.2.3).



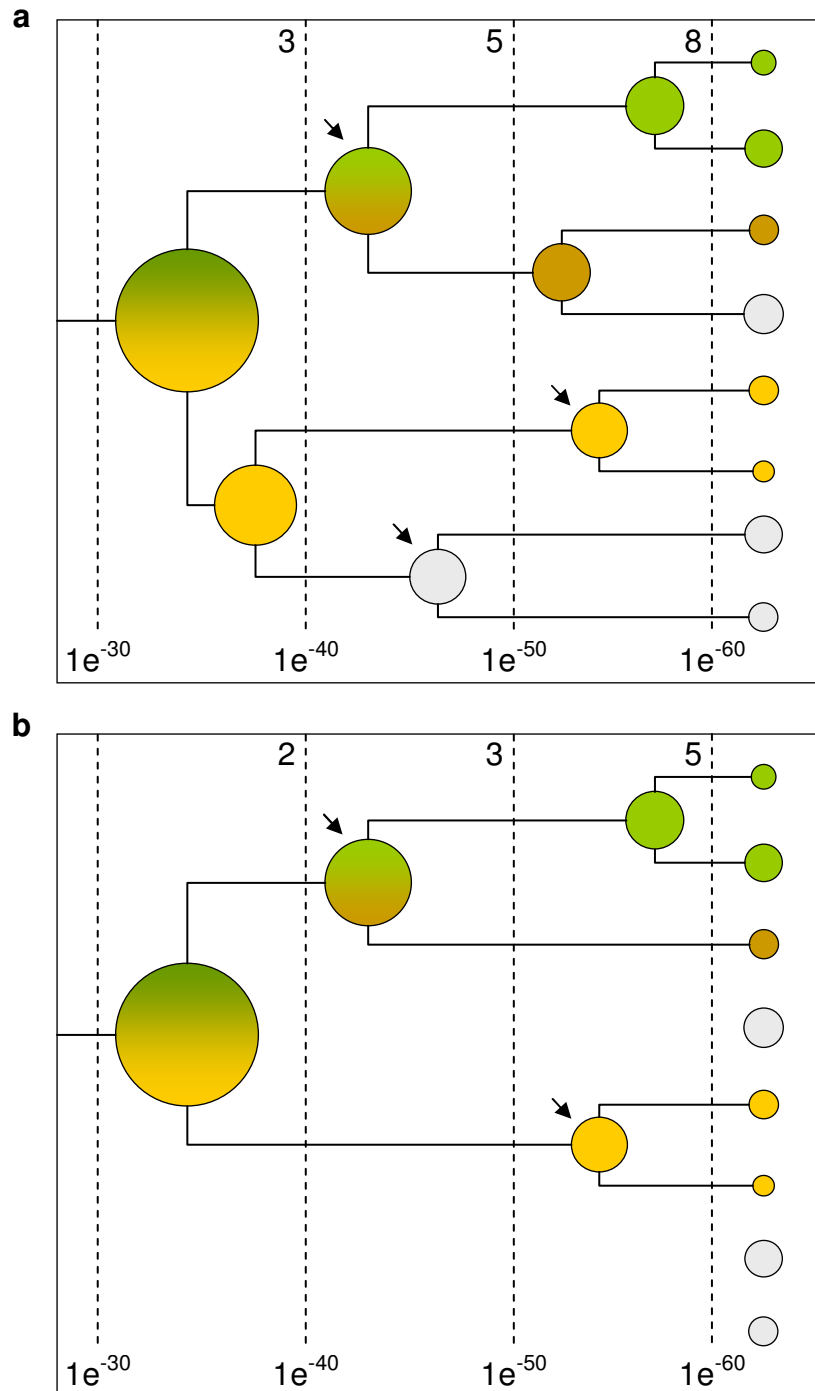
The primary output of this process is a full clustering dendrogram, where the starting clusters are the leaf nodes and the root cluster subsumes all other clusters. Based on the information in this dendrogram, in conjunction with the initial set of starting cluster sequence files, all further cluster-specific files can be regenerated in consecutive steps of the DFX pipeline. While the FASTA sequence files, the MAFFT multiple alignments and the COMPASS alignment profiles that are generated by GeMMA for each cluster could also be stored and used directly, there are good reasons not to do this. First, it is not known at the end of the clustering step which clusters will become family clusters. The necessary storage and transfer (from the HPC to the local system) of the above-mentioned files for *all* clusters created during clustering would result in considerable additional overhead. Second, the realignment of all family clusters for model generation (see Section 3.3.6) should be performed with high-quality parameters set at all times, in contrast to the use of either high- or low-quality settings depending on the number of sequences to align, the current strategy in GeMMA. Third, the COMPASS method, as currently used by GeMMA to create and compare cluster profiles, is considerably slower than the HMMER suite of tools that is used to create and scan against the DFX family model libraries.

### 3.3.4 The two family identification protocols

The DFX supervised family identification protocol (see Chapter 5) uses a supervised clustering evaluation strategy (see Section 2.1.3.2). In particular, it makes use of protein function annotation data in deriving the family partitioning. The unsupervised protocol (see Chapter 4) also uses this strategy in the training step, to derive a generic clustering granularity setting. However, the relationship between sequence and function conservation in the training superfamilies may differ substantially from that in an arbitrary superfamily. Therefore, the use of annotation data, where it is available, can generally be

expected to produce partitionings of higher quality than those produced by the unsupervised protocol.

The supervised protocol considers all sequence clusters in the produced clustering dendrogram that existed at any point in the clustering process in family identification. In contrast, the use of a generic granularity setting in the unsupervised protocol only allows for family partitionings that correspond to ‘straight vertical cuts’ of the clustering dendrogram. The second major difference between the two family identification protocols lies in the preceding sequence clustering step. When DFX runs in unsupervised mode (and the unsupervised protocol is used), all sequences in the superfamily take part in clustering. In contrast, when DFX runs in supervised mode (and the supervised protocol is used), only those sequences with high-quality function annotations take part in clustering (see Section 3.3.3.1). These fundamental differences between the two protocols, and the (theoretical) benefits of using the supervised protocol, are illustrated in Figure 3.2.



**Figure 3.2. Comparison of the unsupervised and supervised family identification protocols by example.** (a) DFX<sub>unsuper</sub> clustering dendrogram of an example sequence superfamily. The colours correspond to the different protein functions associated with the clusters; grey indicates a lack of function annotation. Clusters without annotation are coloured grey. (b) The corresponding part of the DFX<sub>super</sub> clustering dendrogram. Note that unannotated starting clusters (grey) are here filtered out prior to clustering. The numbers at the bottom of both subfigures indicate the respective GeMMA round (threshold setting); the numbers at the top state how many clusters exist at a given point in clustering, respectively. Black arrows indicate which clusters exist after the  $10^{-40}$  round of GeMMA clustering, respectively.

Figure 3.2 depicts the partial clustering dendrogram of a given domain superfamily, processed with DFX running in either unsupervised mode (Figure 3.2a) or supervised mode (Figure 3.2b). The clusters are coloured by the functions (annotations) that are associated with some or all of the sequences they contain, respectively; grey indicates a lack of annotations. The COMPASS E-values at the bottom of the subfigures indicate how far the clustering process has progressed. They correspond to four out of ten threshold levels that are consecutively used for clustering in DFX (see Section 3.3.3.2). The number of existing clusters at each E-value level is stated at the top.

An immediately obvious difference between the two subfigures of Figure 3.2 is that the grey clusters are not part of the dendrogram in Figure 3.2b. This is because these clusters are either leaf clusters or parents of leaf clusters that do not contain any high-quality annotated sequences. As such, the respective starting clusters are removed prior to clustering when DFX runs in supervised mode. The grey clusters in Figure 3.2b therefore indicate non-existing clusters. Later on, after family identification, the sequences from these clusters are assigned to families using the generated model libraries (see Section 3.3.6).

The DFX unsupervised family identification protocol currently uses a generic granularity threshold setting of  $10^{-40}$ . In the example in Figure 3.2a this would produce three families for the shown part of the superfamily dendrogram, namely the three clusters that still exist after the  $10^{-40}$  round of GeMMA (arrows). In contrast, the supervised protocol would identify three other clusters as putative families (arrows in Figure 3.2b), based on tracing the dendrogram as a whole and identifying the individual points at which clusters (sequences) with different associated functions get merged (mixed).

The families produced by the two protocols in the example in Figure 3.2 show different characteristics in terms of functional purity and size, respectively.

While the three families identified by the supervised protocol (see Figure 3.2b) are relatively small but functionally fully conserved, and there only exists a single family per function, the picture is different for the unsupervised protocol (see Figure 3.2a). Here, a large but impure family is derived (green/brown cluster), along with two other, smaller families (yellow and grey clusters). If it is assumed that the sequences in the grey cluster, which all lack high-quality annotations, in fact have the ‘yellow’ function, this means that the unsupervised protocol not only mixes different functions in this example (decreased specificity or purity) but also produces two families for the same function (decreased sensitivity or increased overdivision). The supervised protocol, however, would yield 100% purity and 0% overdivision.

In addition to the above considerations, the supervised protocol would not suffer from a coverage decrease in the (idealised) example discussed above (Figure 3.2). Despite the smaller size of the produced families on the whole (number of seed sequences), and the therefore comparatively small multiple alignments used in family model generation (for example, in the case of the ‘brown’ function in Figure 3.2b), it can be expected that all unannotated sequences (from the grey clusters) are assigned to the right families in the assignment step of DFX, provided that appropriate model thresholds are used (see Section 3.3.6). In summary, in the outlined example the supervised protocol would yield considerable better family identification performance than the unsupervised protocol, whilst maintaining the same coverage rate.

### 3.3.5 Family naming and taxonomic characterisation

To label each domain family identified in a superfamily uniquely, DFX implements a naming protocol that is based on the UniProtKB names of the respective parent proteins. The generated family names are augmented by domain order information (numbers) if it is indicated that domains from two or more families in the superfamily (consistently) co-occur in proteins. Further,

the species information that is initially compiled along with the names for all proteins (see Section 3.3.2) is used to identify the last common ancestor taxon (or the domains of life covered) for each family. In the case of otherwise identical family names, domain order and/or taxonomic information can be used to distinguish the families in a biologically ‘meaningful’ manner (not only by their family ID, which is a unique number by definition). The additional information can hint at, for example, domain duplication or horizontal gene transfer events. The naming protocol is described in detail in the following.

For a given family, the DFX naming protocol performs four steps. First, all words that appear in any of the parent protein names are compiled, splitting the latter where whitespace characters occur. Second, a score for each occurring word is derived. Currently, this is given (simply) by the word’s frequency (occurrence count) in the set of protein names, respectively, where each individual instance counts, including several instances in the same protein name. Third, a score for each protein name is derived. This is given by the sum of the scores of all words it contains divided by the number of words (normalisation), respectively. The normalisation procedure is to prevent protein names from achieving high scores merely on the basis of length. Rather, the commonness of the words in a given name in the overall set of names is the decisive factor. Fourth, a sorted list of all protein names is compiled. Specifically, the names are sorted twice, first by decreasing length and then by decreasing score. As a result, the longest of all protein names that share the highest score is found at the top of the list.

After sorting all protein names in the above-described way, two final tests are made to identify the most appropriate family name. First, the list is traversed from top to bottom to identify the first name that does not include any of the following (currently defined) ‘blacklisted’ terms: ‘bifunctional’, ‘chromosome’, ‘clone’, ‘confirmed’, ‘containing’, ‘domain’, ‘expressed’, ‘fragment’, ‘genomic’,

‘homolog’, ‘isoform’, ‘-like’, ‘partial’, ‘possible’, ‘probable’, ‘protein’, ‘putative’, ‘similar’, ‘trifunctional’, ‘uncharacterised’, ‘unknown’, and the double minus (dash) pattern ‘--’. If all protein names in the list contain blacklisted terms, the top-most name is selected. In the second step, the identified name is tested for being longer than a maximum length of  $C_{\max}$  characters. If this is the case, and the name contains more than one word, it is ‘reconstructed’ in the following way to meet the length constraint: the individual words are added (in the order they occur and separated by spaces) to an initially empty string up to the point at which the resulting string would be longer than  $C_{\max}$ . Effectively that means truncating the selected protein name in a ‘soft’ manner, not splitting words at the end (but rather omitting them). The resulting string is the family name, optionally with the added suffix ‘-like domain’.

An example of the naming process is given in Figure 3.3a, for a family of domains with (phospho)adenosine phosphosulfate reductase activity that is identified by  $DFX_{\text{super}}$  in the HUP superfamily (see Section 5.4.1.2). The family name, ‘Phosphoadenosine phosphosulfate reductase -like domain’, is a direct result of the naming process described below. As explained above, the name with the second-highest score in Figure 3.3a would not be chosen as the family name in the first place, because of the occurrence of ‘probable’. Notably, however, even if it contained another non-informative word instead of the latter, it would not attain the same score as the top-scoring naming; this is due to the normalisation of all protein name scores by word count (see above). Figure 3.3b illustrates that the ‘uncharacterised’ protein names associated with the family do not reflect current knowledge (and should be changed).

**a**

Name of domain sequence cluster parent protein	Score
Phosphoadenosine phosphosulfate reductase	112.33
Probable <b>phosphoadenosine phosphosulfate reductase</b>	84.75
Likely phosphoadenylylsulfate <b>reductase</b>	39.00
5'-adenylylsulfate <b>reductase</b> 3, chloroplastic	30.50
5'-adenylylsulfate <b>reductase</b> 1, chloroplastic	30.50
5'-adenylylsulfate <b>reductase</b> 2, chloroplastic	30.50
Uncharacterized protein MJ0973	1.66
Uncharacterized protein MJ0066	1.66

**b**

Alignments	Accession	Entry name	Status	Protein names	Length	Identity	Score	E-Value
	Q58383	Y973_METJA	★	Uncharacterized protein MJ0973	411	100.0%	2,177	0.0
	D3S4E0	D3S4E0_METSF	★	Phosphoadenosine phosphosulfate reductase	410	93.0%	2,056	0.0
	C7P7Y8	C7P7Y8_METFA	★	Phosphoadenosine phosphosulfate reductase	409	89.0%	1,999	0.0
	C9RIA6	C9RIA6_METVM	★	Phosphoadenosine phosphosulfate reductase	409	78.0%	1,784	0.0
	D5VS16	D5VS16_METIM	★	Phosphoadenosine phosphosulfate reductase	395	68.0%	1,509	1.0×10 <sup>-165</sup>

**Figure 3.3. The DFX family naming protocol.** (a) This example shows a non-redundant list of the protein names that are associated with the domain family (cluster) 48077, as identified by  $DFX_{super}$  in the HUP superfamily (see Section 5.4.1.2). As described in the main text, each protein name is associated with a score that is based on the frequency of the words it contains in the overall set of names. Words occurring in the top-scoring name are highlighted in red in the remaining names. The cluster as a whole contains 254 sequences. (b) The results of UniProtKB BLAST searches demonstrate that the poor quality of some protein names associated with the cluster in (a) does not reflect the available similarity information. One of the proteins in (a), stemming from SwissProt (golden star), was used as the query (dashed arrow).

In the course of compiling all candidate (parent protein) names for the domain family naming procedure, as outlined above, a list for each protein is compiled that contains the identifiers of all families in the superfamily in which it has at least one domain. For a minority of proteins, this list contains more than a single family ID, and only those are further analysed. In particular, for each family that appears in any of the protein family lists, the protein that has domains in the highest number of further families in the superfamily is identified, respectively, and denoted as the ‘representative parent’ sequence. Finally, it is checked for each representative parent protein whether all the families in which it has domains received the same family name in the naming



procedure described above. If that is the case, each of the family names is made unique by extending it with a suffix that indicates the relative position (domain number) in the representative parent protein. The domain numbers are derived by sorting the starting coordinates of the domains in the parent protein from lowest to highest value.

Having named all families and added domain order information, the last step in the protocol is to characterise the families taxonomically. To this end, the last (most specific) common ancestor taxon of all parent protein taxa (source genomes) is determined for each family using the data from the UniProt Taxonomy, respectively. This information is stored and optionally added to the family name.

### 3.3.6 Model library generation and family assignment

Once the families in a given superfamily have been identified, the DFX workflow commences with the generation of a library of models, one for each family, together with model-specific thresholds. First, a profile-HMM (see Section 1.4.2) is built for each family sequence cluster and the cluster it is most closely related to, respectively. The latter is the ‘sibling’ cluster of the family cluster, that is, the one it was merged with during sequence clustering. Second, two different model-specific detection thresholds (as described below) are determined for each family model. These are derived from the results of scanning (i) the sequences in the family cluster and (ii) the sequences in the family’s sibling cluster with the respective model. Third, any sequences known to belong to the processed superfamily (but potentially not functionally characterised) can be scanned with the model library to assign them to one of the identified families, respectively.

The DFX pipeline currently uses the HMMER suite of tools for handling profile HMMs. In particular, the following three commands are used. First, to

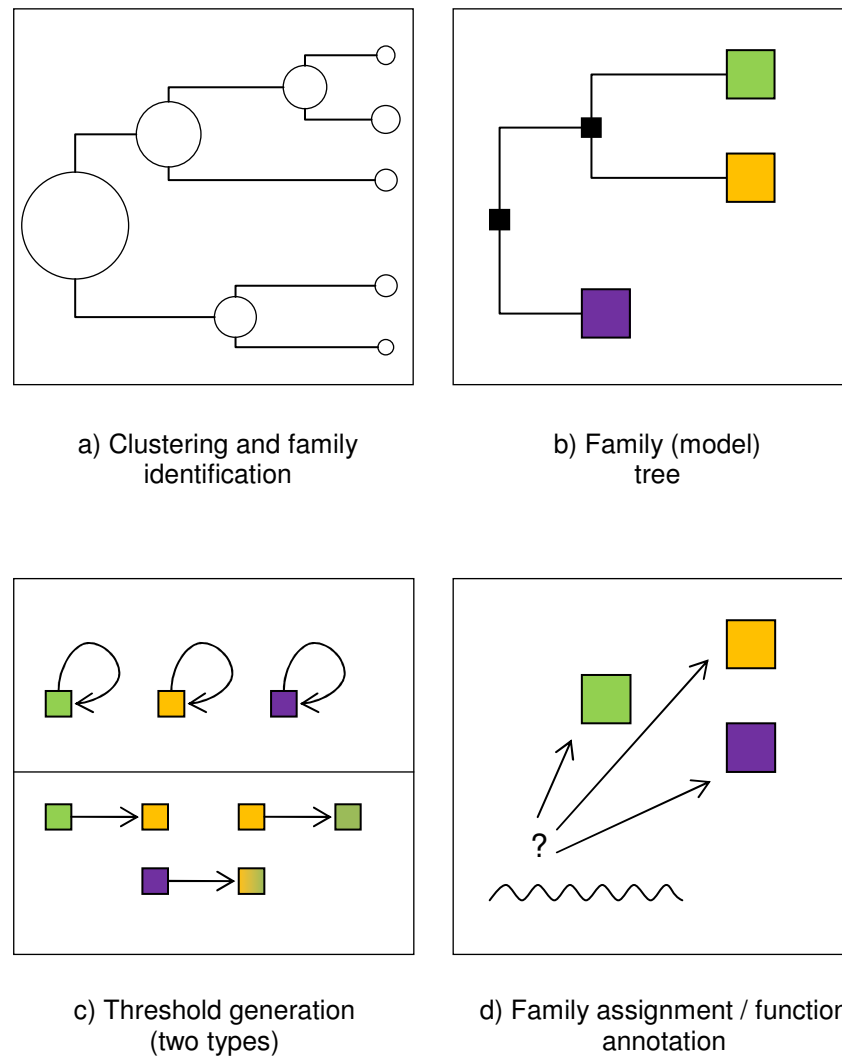
build a model for a given sequence cluster, the *hmm\_build* command is used with default parameters; the cluster is aligned beforehand, using MAFFT with high-quality settings ('--amino --localpair --maxiterate 1000'). Second, the *hmm\_search* command is used to scan sets of sequences with individual models. Third, the *hmm\_scan* command is used to scan sets of sequences with all models in a model library. The latter two HMMER commands return bit scores, to indicate how well a given sequence matches a certain model.

Stringent model-specific detection thresholds can be derived by scanning all the sequences from which a model was built against the model itself, and then use the lowest score attained as the threshold (Podell and Gaasterland 2007; Fong and Marchler-Bauer 2008). This is the score any query sequence will have to meet or exceed to be classified as belonging to the sequence family represented by a given model. Apart from this 'inclusion' threshold, representing the upper boundary of the range of possible threshold values for a given model, the DFX pipeline also generates a more liberal 'exclusion' threshold for each model. This time it is not the sequences in the family cluster underlying the respective model itself that are scanned against the model, but those found in its sibling cluster (see above). Accordingly, it is the highest score observed that serves as the model-specific exclusion threshold. This is the score any query sequence will have to meet or exceed to be classified as being (most) closely related to the sequence family represented by a given model. Such relatedness can either indicate that the query sequence represents a new member of the family in question, or a member of an uncharacterised family occupying a part of superfamily sequence space that lies 'between' the family hit and the closest neighbouring family. The process of generating the model library along with the corresponding thresholds is summarised in Figure 3.4. It is indicated that the model-based family assignment step also forms the basis of (protein) function assignments, as described in the below section.

### 3.3.7 Function assignment to whole-protein sequences

An important application of protein family libraries is the (probabilistic) assignment of functions to uncharacterised sequences (protein function prediction). DFX generates such libraries on the level of individual protein domains, and these can be used to annotate proteins accordingly. To exploit the potential of this approach fully, it was implemented in a manner that allows it to combine the similarity signals between individual domains of a query (multi-domain) sequence and different domain families (that were identified in different superfamilies). The function assignment protocol is described in detail in the following.

Each domain family identified by DFX can be associated with Gene Ontology protein annotations in a probabilistic manner. Specifically, for a given family, this is (currently) done for all most-specific GO terms from the total set of terms that are associated with the domain sequences underlying the family model (seed sequences); that is, the annotations of the respective parent proteins. Note that this simple approach implicitly takes into account the likelihood of other domains (with other, specific functions) to co-occur with domains from the processed family in known proteins. The probability of each most-specific GO term being associated with the family (model) is calculated as its annotation frequency among the seed sequences, respectively. Further, the generated probability values are propagated up the GO DAG, where the probability of each parent term is given as the average probability of its direct child terms.



**Figure 3.4. Generation and use of the family model library.** (a) A hypothetical sequence superfamily is processed with DFX, using either  $DFX_{super}$  or  $DFX_{unsuper}$ . (b) Subsequently, a model is generated for each identified family (cluster). (c) Two types of model-specific thresholds are then generated, an inclusion threshold (determined by the worst self-hit of a model seed sequence to its model) and an exclusion threshold (determined by the best hit to a model from a seed sequence of its sibling model in the tree). (d) Domain sequences that have been assigned to the superfamily in question can be scanned against the model library and assigned to the family whose model they hit best (optionally using one of the two thresholds).

Note that, while high-quality GO annotations are only available for those superfamilies (and their families) that are processed by DFX in supervised mode, some (low-quality or recently added high-quality) annotations for the remaining superfamilies (and their families) are often found too, and can optionally be used. Further, the protocol outlined below is meant to exemplify the more general framework of domain-based protein function prediction, which could exploit any kind of whole-protein annotation data. Finally, the

protocol relies heavily on the generated model libraries and the associated thresholds. For these reasons, it is described here rather than in Chapter 5.

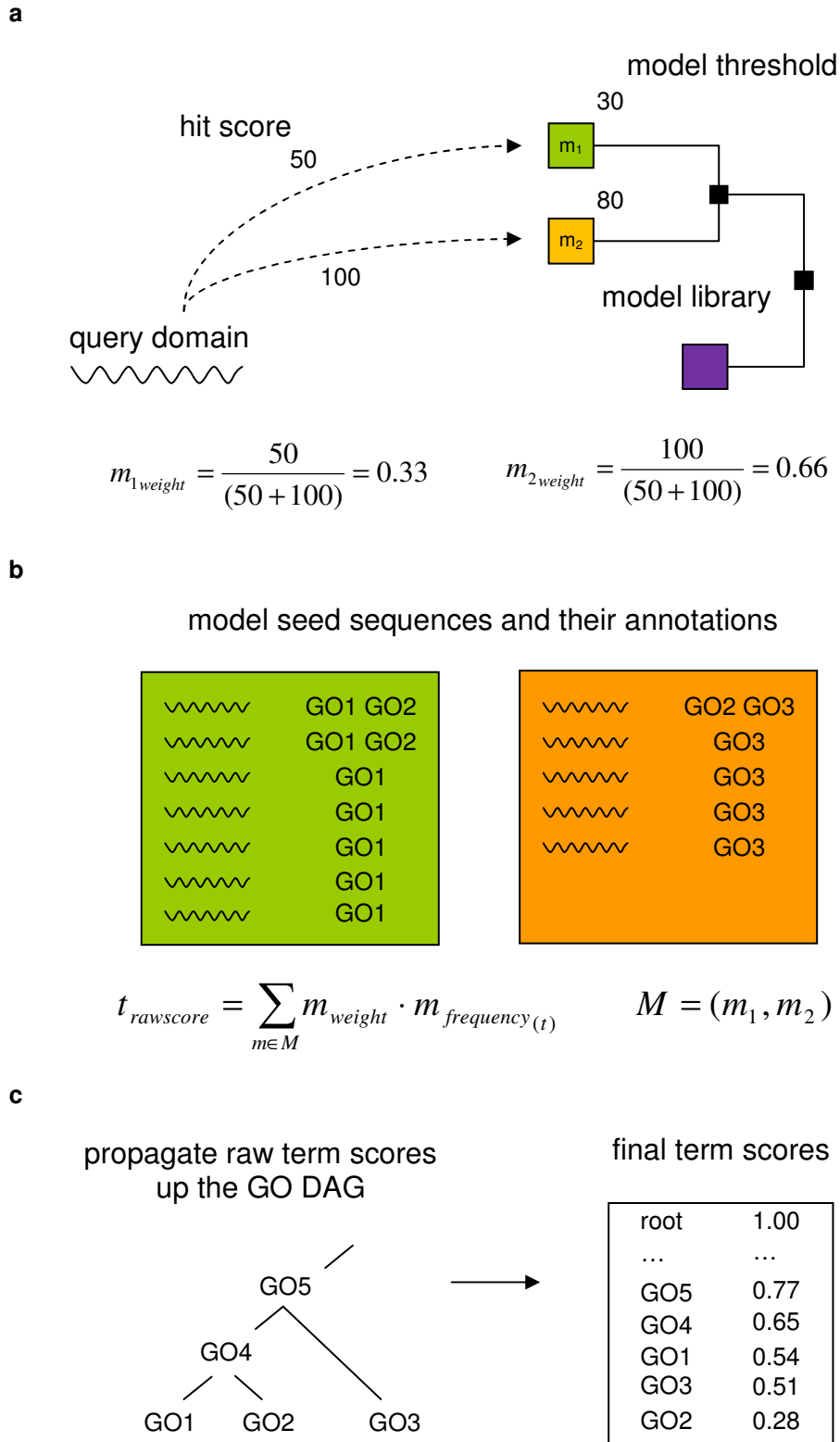
Figure 3.5 summarises the process of probabilistic function assignment for an individual domain in a protein query sequence (as identified and assigned to a superfamily by Gene3D beforehand). First, the domain sequence is scanned against the DFX family model library of its superfamily (Figure 3.5a). The numbers stated next to the models represent the model-specific thresholds used. The dashed lines indicate hits to individual models, with the respective hit scores given as well. For both hits shown the scores exceed the respective target family's threshold.

Which type of model-specific thresholds are used when scanning the model library, inclusion or exclusion (see Section 3.3.6), depends on the specific prediction task; it is a user choice. The same is true for the decision of whether all hits to models are considered (as assumed in the further description of the protocol below) or only the top hit. In fact, for the CAFA 2011 function prediction challenge (see Section 7.2.2), the best function prediction performance was attained using the top hits only, and without using *any* model-specific thresholds.

After scanning the query domain sequence against the family model library of its superfamily (Figure 3.5a), all models that have been hit with a score that meets the respective model-specific threshold (target models) are considered in probabilistic function assignment (Figure 3.5b). This is done in a weighted manner, where the weight of each target model ( $m_{\text{weight}}$ ) is given by its relative hit score. The relative hit score of a given model is its hit score expressed as a fraction of the sum of all target model hit scores. For each GO term  $t$  that is associated with at least one target model  $m$ , a 'raw' total probability score is calculated using the equation stated in Figure 3.5b.

The calculation of the raw probability scores for each GO term as shown in Figure 3.5b is based on the relative weights of the target models associated with  $t$  and its annotation frequency among the seed sequences of these models, respectively. The frequency values used in this calculation are stored with each model, including the frequencies of all parent terms in the GO DAG (see above). The up-propagation procedure shown at the bottom of Figure 3.5c is therefore implicit in the outlined calculations (all parent terms are considered). At the end of the workflow as shown in Figure 3.5 stands a list of GO terms that are predicted to be associated with the query domain sequence, each with a certain probability.

When multi-domain protein sequences are to be characterised functionally using the above protocol, the protocol is run for each individual domain and a simple integration procedure is devised subsequently. In brief, the whole-protein probability score of each GO term is set to the maximum domain probability score obtained for the term, respectively. This accounts for cases in which a specific term is assigned based on more than one domain in (more than one run of the protocol for) the same protein. Possible enhancements of this basic integration procedure are discussed in Section 7.2.2.



**Figure 3.5. Probabilistic GO term assignment based on a single query domain.** This example shows the three-step workflow that is followed for each domain detected in a given protein. (a) The domain sequence is scanned against the full family model library of its superfamily and produces two hits (dashed lines). The model-specific thresholds and attained scores are shown. In this case, both models are hit above their thresholds. Each is assigned a relative weight. (b) A raw score is derived for each term  $t$ , by integrating the different model weights and associated term frequencies (number of model seed sequences with that

term). (c) Coarse annotations are corroborated by more specific annotations through up-propagation of term scores in the GO DAG. This yields the final list of probabilistic function assignments.

### 3.4 Discussion and future work

The concepts, overall architecture and individual modules of the DFX pipeline for protein domain family identification have been discussed above. DFX embeds the GeMMA clustering method, as discussed in Chapter 2, and two different family identification protocols, as described in Chapter 4 and Chapter 5. The two protocols are further compared in a quantitative manner in Chapter 6, based on the results of the first large-scale run of DFX. Finally, Chapter 7 summarises the overall work conducted for DFX, that is, the work presented in this thesis, and gives an overview of its current and future usage, as well as the plans for its further development. The specific characteristics and challenges of the algorithms used in each of the DFX modules are discussed in the respective chapters, as well as the results obtained and the assessment strategies used. Accordingly, in the following the focus is on potential improvements to the common modules of DFX, that is, those that are used in both supervised and unsupervised mode.

#### 3.4.1 The use of sequence data

Concerning the DFX input data, it has become clear in the process of analysing the results of the DFX supervised family identification protocol (DFX<sub>super</sub>) that a scheme for filtering out fragmentary sequences, and possibly also low-complexity sequences, should be implemented. Gene3D, as a hitherto purely superfamily-based resource, assigns domains to protein sequences of all sizes. This means that fragmentary proteins from UniProtKB currently also contribute domains to the DFX input sequence data. It has been noted that truncated domain sequences can misguide the sequence clustering process and in this way give rise to artefactual (often single-sequence) families, as shown by different examples in Chapter 5.



Erroneously assigned domains can cause the same problems as domains from fragmentary protein sequences, as described above. Such sequences are often substantially shorter or longer than their relatives. Filtering them out, however, is a much more challenging task than the removal of fragmentary sequences. It is conceivable that available information about protein domain architecture could be used to identify putative domain assignment errors in multi-domain proteins, either within DFX or earlier, in the domain assignment process (Yeats, Redfern et al. 2010). Further, superfamily-specific minimum sequence length thresholds could be used. However, it must be kept in mind that such approaches bear the risk of filtering out valid domains that merely represent outlier cases, possibly of high (evolutionary) interest.

#### 3.4.2 The two family identification protocols

For both family identification protocols used in DFX it is conceivable to exploit taxonomy information (apart from further annotation data in DFX<sub>super</sub>; see Section 5.5.3). For example, there could be constraints as to which higher level (sequence source) taxa can be merged in individual families. An example would be a rule that prevents the merging of sequences from different domains of life (DoLs) to establish DoL-specific families in large and diverse domain superfamilies; such families could aid the study of domain evolution. A similar but merely ‘monitoring’ rule (test) could also be used to detect putative instances of horizontal gene transfer in the currently established families.

Another example of using taxonomic information would be a rule that prevents similar (in sequence and/or annotation) sequences that stem from the same source genome from being merged, or a test that keeps track of such events. A similar strategy is used in phylogenomic protein function prediction approaches (Eisen 1998; Engelhardt, Jordan et al. 2005; Engelhardt, Jordan et al. 2009; Thomas 2010), to differentiate between orthologous and

paralogous sequences. While the latter two concepts can only be partially transferred to the domain level (see Section 1.2.3), particularly cases where the sequences in two merged clusters have the same or a very similar taxonomic distribution could hint at gene duplication events and, therefore, functional divergence (on the whole-protein level). Especially in conjunction with annotation information, as in  $DFX_{\text{super}}$ , this could be a promising approach. Further, and particularly interesting in the context of domain evolution, ancestral *domain* duplication events may be detected in a similar manner. The above rule would then analyse the pattern of parent proteins, not parent taxa, of the sequences in two merged clusters. This, of course, would have to take into account the less likely divergence in function in the case of duplicated domains (within in the same gene) as compared with whole-protein paralogues.

### 3.4.3 The family naming protocol

Domain architecture information could be used to improve the naming of domain families in DFX, in addition to its potential uses in the identification of putatively erroneous domain assignments (see above) and in disentangling the function annotations arising from different domains in the same protein (see Section 5.5.3). In particular, information on whether or not sequences from a given domain family appear with other domains at all, and, if so, at which relative position in proteins, could be added to the family name. Currently, this is only done in cases where a protein has domains from different identified families in the same superfamily (see Section 3.3.5).

More generally, the current family naming protocol could be enhanced using advanced semantic methods. Particularly inspiring in this context could be two existing tools: the Protein Naming Utility (Goll, Montgomery et al. 2010) and

Gene Pidgin<sup>13</sup>. More specifically, instead of scoring all terms that appear in the protein names associated with a domain family and subsequently basing the family name on the protein name with the highest total score (as currently done; see Section 3.3.5), a more sophisticated protocol could take into account the frequency of word combinations and the order and type of words in the protein names. Potentially, in some cases, domain family names could then be constructed that arise from a combination of highly informative terms that is not seen in any of the associated protein names, and omit less informative terms from these names. For superfamilies with associated GO annotations, these could be of additional value in the naming process; for example, if a family's last common ancestor GO term (as readily identified by DFX) is specific enough to convey some information about the (proteins associated with the) domains in the family.

#### 3.4.4 The overall architecture of DFX

On the technical level, it is conceivable that all processing steps in the DFX pipeline (not from data preparation and storage) could be implemented to run in HPC environments. While this would result in large amounts of data having to be transferred to a local storage system after running the pipeline, it would make the protocol more integrated, from a user's point of view. Since DFX uses parallel batch processing in all local parts of the pipeline already (see Section 3.3.1.1) this moving to HPC entirely would be relatively straightforward. So far, however, the need for an HPC system in the clustering stage was considered a 'necessary evil' (based on the amounts of sequence data to process compared with the processing power of a single machine) rather than an asset of DFX. With distributed (and cloud) computing becoming more and more common, however, this may change. A more integrated, purely HPC-oriented protocol could then be favoured by users.

---

<sup>13</sup> <http://genepidgin.sourceforge.net/>

## Chapter 4. Unsupervised protein domain family identification in DFX

The DFX unsupervised family identification protocol (DFX<sub>unsuper</sub>) uses the results of the DFX sequence clustering step in conjunction with a generic granularity setting to identify families in protein domain superfamilies. The respective setting is derived in a training step, which is based on a gold standard family dataset. Therefore, DFX<sub>unsuper</sub> is not, in the strict sense of the word, an unsupervised protocol. Just as the supervised protocol (DFX<sub>super</sub>; see Chapter 5) it uses ‘annotation’ data, in the form of family assignments in the training step. However, the extent of these data is much smaller, and it is *external* data, with respect to the large majority of potential target domain superfamilies (all apart from those used in training). The name of the DFX<sub>unsuper</sub> protocol is primarily to distinguish the two DFX family identification protocols.

The main motivation behind developing DFX<sub>unsuper</sub> was that established methods for unsupervised sequence family identification are not able to cope with large input datasets, that is, large protein domain superfamilies. This refers to both resource usage (processing power and/or memory) and technical difficulties, such as the requirement for an initial (large) multiple sequence alignment. Resource constraints, as the main problem, can be overcome by the implementation of algorithms to run in HPC environments. In the case of family identification methods, the bottleneck is the sequence clustering step. Therefore, the DFX<sub>unsuper</sub> protocol uses the GeMMA clustering method (see Chapter 2).

The background section first reviews existing methods and protocols for unsupervised sequence family identification. The datasets, measures and procedures used to train and benchmark the DFX<sub>unsuper</sub> protocol are then described in the implementation section. The corresponding results are

presented and discussed thereafter. The qualitative assessment of  $\text{DFX}_{\text{unsuper}}$  in benchmarking is augmented by a larger, quantitative assessment together with  $\text{DFX}_{\text{super}}$  in Chapter 6. The current chapter closes with a brief outline of recent use cases of the protocol and suggested next steps in its development. Additional points that may affect both DFX family identification protocols are discussed in Chapter 7.

## 4.1 Background

Subsequent to some general marks on the origin of the protocol discussed in this chapter, this section reviews existing methods for unsupervised family identification.

### 4.1.1 Preliminary remarks

Notably,  $\text{DFX}_{\text{unsuper}}$  was initially published in Lee, Rentzsch et al. (2010) under the name GeMMA, which was at that point referring to a combined workflow that included sequence clustering *and* family identification. As DFX as a pipeline did not yet exist, it had not been necessary or beneficial conceptually to disentangle the two steps. GeMMA, as an independent and entirely HPC-based sequence clustering method, is now used in the clustering step of DFX, regardless of which of the two DFX family identification protocols is used subsequently.

A second important point is that, apart from the choice of appropriate training data, a functionally ‘blind’ protocol such as  $\text{DFX}_{\text{unsuper}}$  can only *aim* to produce families that adhere to the DFX domain family concept introduced in Chapter 3. Unlike  $\text{DFX}_{\text{super}}$ , it cannot ‘force’ a certain level of functional granularity with the help of annotation data.

## 4.1.2 Existing unsupervised family identification methods

In the following, two standalone, *ab-initio* methods for family identification are discussed in detail. These integrate hierarchical sequence clustering and unsupervised clustering evaluation (see Section 2.1.3.1). Thereafter, several non-integrated protocols that combine different clustering algorithms with different unsupervised evaluation strategies are outlined briefly.

### 4.1.2.1 Integrated methods

Both the SCI-PHY (Brown, Krishnamurthy et al. 2007) and CLUSS (Kelil, Wang et al. 2007) methods combine agglomerative hierarchical clustering with an unsupervised clustering evaluation strategy, respectively, to identify protein families in an *ab-initio* manner. The clustering step is to sample from the space of all possible partitionings of the input dataset. The evaluation step is to select the best partitioning from those sampled, according to some global measure of cluster cohesion and separation (see Section 2.1.3.1). SCI-PHY requires as input a multiple alignment of the sequences in the dataset to be processed, whereas CLUSS is an entirely alignment-free method.

SCI-PHY (Brown, Krishnamurthy et al. 2007) (formerly called BETE for ‘Bayesian Evolutionary Tree Estimation’ (Sjolander 1998)) first clusters the sequences (rows) in the input alignment using a profile linkage approach to hierarchical clustering. Such an approach is also used by GeMMA (see Chapter 2), which, however, creates alignments in a ‘bottom-up’ manner during clustering. In a second step, SCI-PHY then identifies the best partitioning according to an encoding cost function that incorporates both the number of clusters as well as profile-based measures for cluster cohesion and separation (see Section 2.1.3.1). This partitioning is reported as the family decomposition of the input alignment.

During the hierarchical clustering process, SCI-PHY generates residue distribution profiles to represent the clusters (initially consisting of single sequences) and uses the relative entropy between these profiles as the cluster dissimilarity measure. The profiles are derived from the observed residue counts in each column of the respective cluster alignment (the ‘posterior’) and an assumed generic background distribution of the different residue types (the ‘prior’). The combination of prior and posterior allows for the calculation of estimated residue frequency values for each alignment position and residue type, even for residues that are not observed in a given alignment column.

The SCI-PHY method uses a residue distribution prior at two points: in the construction of cluster profiles and when measuring the entropy of individual alignment columns in the encoding cost function. In mathematical terms, the prior is a residue probability density function in the form of a Dirichlet mixture density (Sjolander, Karplus et al. 1996). The SCI-PHY standard prior is derived from the residue distributions observed in sets of high-quality alignments in the BLOCKS database (Henikoff and Henikoff 1992). The use of priors has been shown to help create more specific and selective profiles than those based on common substitution matrices (Brown, Hughey et al. 1993). It corresponds to the use of ‘pseudo-counts’ in PSSMs (see Section 1.4.2.1) and is thought to increase profile sensitivity in the case of sparse and/or unevenly sampled sequence data (Sjolander, Karplus et al. 1996).

The CLUSS method first constructs an all-by-all similarity matrix of the sequences in the input dataset. In this, it uses an alignment-free sequence similarity measure, based on short exact subsequence (‘word’) matches. This Substitution Matching Similarity (SMS) measure weighs each word match according to its length and the ‘inertia’ of the residues it comprises, based on their self-substitution scores in a standard amino acid substitution matrix. The individual word weights are then summed up and the total score is normalised to produce a total similarity score for a pair of sequences. The second step in

CLUSS is the construction of a clustering dendrogram based on the calculated similarity matrix, following the standard average linkage hierarchical clustering approach (see Section 2.1.2.1). Finally, the best partitioning is identified in three sub-steps. First, a ‘co-similarity’ value for each node (cluster) in the dendrogram is calculated. This takes into account both cluster cohesion and separation. Second, each cluster is assigned to one of two groups, high co-similarity or low co-similarity. This division is made using a maximum interclass inertia method. Finally, the disjoint set of all largest high co-similarity clusters is reported as the family partitioning of the dataset.

#### 4.1.2.2 Combined protocols

Donald and Shakhnovich (2005) first used single linkage hierarchical clustering in conjunction with a giant component (see Section 2.1.3.1) approach to identify protein domain families *ab-initio*, at an ‘...intermediate level of functional detail...’ above the orthologue cluster level. This basic definition in principle corresponds to the more detailed definition of the family concept found in Section 0. The protocol was benchmarked on three datasets of eukaryotic transcription factor DNA binding domains, trying to divide each dataset into families of domains with matching binding specificity automatically. Its performance is shown to be superior to both the use of a fixed global sequence identity threshold to stop the clustering process and the use of the graph-based method TRIBE-MCL (see Section 2.1.4) with a range of different settings for its granularity parameter (‘inflation’ value).

It must be stressed that graph-based clustering methods, like any other, are not suitable to establish sequence families when used in isolation, that is, without a strategy to optimise the respective granularity settings. In contrast to what is sometimes suggested (Enright, Van Dongen et al. 2002), the structure of the graph alone cannot usually be expected to reveal biologically ‘meaningful’ partitionings such as families. For example, the granularity setting



and the size and type of the input dataset have significant effects on the family partitionings obtained from the graph-based clustering method MCL (Donald and Shakhnovich 2005; Wall, Leebens-Mack et al. 2008) (see Section 2.1.2.3). As for any other clustering method, this becomes especially obvious when such methods are used to infer sequence families across different genomes (Frech and Chen 2010). The observed, highly variable performance when clustering methods alone are used to identify families can be expected from the ‘intermediate’ character of the family concept (see Section 1.2.2.3). In brief, sequence families (and the boundaries between them) are particularly difficult to establish, as they lie between the superfamily level and the level of tight (orthologue) clusters.

For the above reasons, different unsupervised evaluation strategies have been proposed for use with graph-based clustering methods to identify sequence families. One general strategy is to sample a (wide) range of settings for the respective clustering granularity parameter and subsequently select the (relatively) best partitioning, based on some unsupervised evaluation measure (see Section 2.1.3.1). Yang and colleagues sampled 100 evenly distributed settings of the APC ‘preference’ parameter for a given sequence dataset and then used a ‘stable number’ criterion to identify the best family partitioning (Yang, Zhu et al. 2010). In brief, this assesses at which point in the sampled range of settings the longest range of corresponding partitionings with the same (stable) number of clusters is found. The mean value of the parameter settings in this range is then used to derive the final partitioning. The authors benchmark the ability of this approach to correctly separate sequences from different protein superfamilies and families, and claim that their method yields much better performance than BLASTClust, TRIBE-MCL, CLUSS and spectral clustering (all used with a range of settings, but without the stable number optimisation procedure). Strikingly, a single linkage hierarchical sequence clustering method is reported to perform second-best, BLASTClust (see Section 2.1.4).

Apeltsin and co-workers recently proposed a simple heuristic to derive appropriate granularity settings for protein family identification with a wide range of graph-based clustering algorithms (Apeltsin, Morris et al. 2011), including MCL and APC. In brief, this is based on pre-filtering the edges in a given sequence similarity network (SSN) based on their weights (e.g., BLAST E-values) prior to clustering the network. First, the authors sample 100 E-values with an exponent range of 0 to -100 to threshold the network of a given protein superfamily. Each of the 100 networks is then fed into the respective clustering algorithm, using the default setting for the respective granularity parameter. From the results of this, the distribution of average network node degree depending on the initial threshold setting is generated (node degree distribution). From manually inspecting the edge weight distributions of the SSNs of four different superfamilies, the authors infer the following heuristic: a relatively good family partitioning of the input dataset is achieved with any of the clustering algorithms when using the pre-filtering threshold setting that corresponds to the point at which the slope (first derivative) of the node degree distribution reaches its first local maximum (the maximum corresponding to the lowest E-value exponent). The overall methodology is shown to increase the family partitioning performance of all clustering algorithms tested, as compared with using them on unthresholded superfamily SSNs. Strikingly, MCL outperformed all other tested algorithms when using thresholded networks, whilst also being the fastest. Both APC and spectral clustering could not produce meaningful family partitionings for the used datasets at all.

## 4.2 Implementation

$DFX_{\text{unsuper}}$  uses a generic setting for the level of GeMMA clustering granularity to identify putative families in protein domain superfamilies. This E-value threshold (see Section 3.3.3.2) is derived in a one-off training step, using a set of gold standard superfamilies and corresponding family

assignments. By default, DFX clusters all input superfamilies in full, that is, until only a single, large cluster remains. Apart from training, the  $DFX_{\text{unsuper}}$  protocol therefore entails only a single step for each processed superfamily: tracing the individual merges that constitute the full GeMMA clustering dendrogram from the first merge (of two leaf clusters) up to the point at which the first pair of sibling clusters is less similar than the generic threshold (setting) used.

The performance of  $DFX_{\text{unsuper}}$  was assessed using both a small, high quality superfamily dataset (cross-validation on the gold standard superfamilies) and a large, medium quality dataset consisting of functionally diverse Pfam families. In both cases, the performance of the protocol was compared with that of the SCI-PHY method, the putatively best-performing method in the field at the time this work was conducted (Brown, Krishnamurthy et al. 2007). The following sections describe the gold standard dataset, the training and benchmarking procedures and the performance measures used in both training and benchmarking.

#### 4.2.1 The gold standard and derived datasets

A manually curated gold standard dataset of enzyme superfamilies partitioned into families and two derived datasets were used in training (see Section 4.2.3) and, partially, in benchmarking (see Section 4.2.4) the  $DFX_{\text{unsuper}}$  protocol. These datasets are described in the following.

##### 4.2.1.1 The gold standard dataset

The Structure-Function Linkage Database (SFLD) (Pegg, Brown et al. 2005) provides manually curated partitionings of several mechanistically diverse enzyme superfamilies into families, with a focus on function. Where individual domains from multi-domain proteins can catalyze a given reaction by themselves, only the respective ‘core’ domain sequences form the superfamily.

All sequences in a given SFLD superfamily are required to share the same fold and the same principle reaction mechanism (for example, a certain type of catalytic triad). The SFLD curators further divide each superfamily using two hierarchical levels, a coarse (subjective, superfamily-specific) ‘subgroup’ level and a fine (functional) ‘family’ level. In the latter case, all sequences are required to fulfil exactly the same function.

The use of the SFLD as a challenging benchmark for family identification methods has been described in several studies (Brown, Krishnamurthy et al. 2007; Brown 2008; Albayrak, Otu et al. 2010; Moll, Bryant et al. 2010). As of 2009, the SFLD contained six superfamilies, divided into a total of 140 functional families. The full (parent) protein sequences for each superfamily and the respective (domain-based) family annotations were retrieved from the SFLD website<sup>14</sup> on 08/01/2009, as listed in Table 4.1.

---

<sup>14</sup> <http://sfld.rbvi.ucsf.edu/>

**Table 4.1. The SFLD protein dataset and its mapping to Gene3D.** The superfamily sizes for each of the three datasets described in the main text are given in the second (SFLD and SFLD-Gene3D) and last (Gene3D) columns, respectively. The shown figures are for the SFLD database as of January 2009 and Gene3D 7.0. The Terpene cyclases could not be mapped to CATH (see main text).

SFLD superfamily	Total sequences	Annotated sequences (~ % total)	SFLD families	CATH superfamily	Gene3D sequences
Amidohydrolase	1,693	802 (47)	35	3.20.20.140	15,932
Crotonase	1,330	931 (70)	14	3.90.226.10	19,323
Enolase	1,556	1,152 (74)	17	3.20.20.120	4,114
Haloacid dehalogenase	1,285	936 (73)	17	3.40.50.1000	20,614
Terpene cyclase	228	228 (100)	40	n/a	n/a
Vicinal oxygen chelate	683	291 (43)	17	3.10.180.10	11,592

#### 4.2.1.2 Two derived datasets

Two Gene3D domain datasets were derived from the SFLD protein dataset: the SFLD-Gene3D and Gene3D datasets. First, the specific core domains that give rise to the different SFLD superfamilies were (re-)identified in the SFLD whole-protein sequences, through Gene3D domain assignment. The resulting SFLD-Gene3D dataset contains the Gene3D domain sequences that correspond to the (original) SFLD domain sequences for each SFLD superfamily, respectively. The Gene3D dataset extends the SFLD-Gene3D dataset by adding to each superfamily the full set of member domain sequences from Gene3D 7.0. This corresponds to an expansion of the SFLD-Gene3D dataset to related proteins that are either not yet classified in the SFLD or not functionally characterised at all. Consequently, the Gene3D

dataset is considerably larger than the SFLD-Gene3D dataset (see Table 4.1), and subsumes the latter.

The CATH superfamilies that were found to correspond to each of the SFLD superfamilies are listed in Table 4.1. For example, a CATH 3.20.20.140 domain is found in all SFLD protein sequences from the Amidohydrolase superfamily. The Vicinal oxygen chelate proteins are composed of a single CATH domain, while the proteins with domains in the other five superfamilies are multi-domain proteins. In these cases, a variety of different domains accompany the respective SFLD core domain. The Terpene cyclase superfamily was not found fully classified in CATH and therefore had to be excluded from all domain-based analyses.

#### 4.2.2 Performance measures

In Brown, Krishnamurthy et al. (2007) the authors demonstrate the superior performance of the SCI-PHY method compared with a number of other approaches for *ab-initio* family identification. In particular, they use three distinct measures to evaluate a given family partitioning in a supervised manner, that is, based on a gold standard set of protein family (function) assignments: purity, edit distance and VI (Variation of Information) distance (see Section 4.2.2). Purity is a measure of family functional coherence (homogeneity), while edit and VI distance are alternative measures of how well the different functional classes are separated across different families. The same three measures were used in the present work. In detail, they are defined as follows.

##### i) Purity

Purity is measured as the percentage of families within which all annotated member sequences are annotated with the same function. 100% purity can be attained trivially by having each sequence in a separate family.

## ii) Edit distance

Edit distance measures the number of family split or merge operations that are required to transform the proposed family partitioning into the true family partitioning of the dataset. The edit distance between a reference partitioning and a proposed partitioning with clusters  $k$  and  $k'$ , respectively, is calculated as

$$Edit = 2 \cdot \left( \sum_{k,k'} r_{k,k'} \right) - K - K'$$

where  $r_{k,k'}$  equals 1 if clusters  $k$  and  $k'$  have items in common, and 0 otherwise, and  $K$  and  $K'$  are the number of clusters in each partitioning.

## iii) VI distance

VI distance measures the amount of information that is not shared between two family partitionings of the same dataset. It is calculated as

$$VI = H(S) + H(S') - 2I(S, S')$$

where  $H$  is the entropy of a partition and  $I$  is the mutual information between two partitionings:

$$H(S) = \sum_{k=1}^K \frac{n_k}{N} \log \frac{n_k}{N}$$

and

$$I(S, S') = \sum_{k=1}^K \sum_{k'=1}^K \frac{n_{k,k'}}{N} \log \frac{n_{k,k'}}{N}$$

Here,  $n_k$  is the number of items in cluster  $k$  of partitioning  $S$ ,  $n_{k,k'}$  is the number of overlapping items between cluster  $k$  in partitioning  $S$  and cluster  $k'$  in partitioning  $S'$ ,  $K$  and  $K'$  are the total number of clusters in the partitionings

$S$  and  $S'$ , respectively, and  $N$  is the total number of items in the set. Identical partitionings will have both an edit and VI distance of zero.

Both edit distance and VI distance penalize overdivision as well as mixing of subtypes. These two measures are analogous to sensitivity (recall) while purity is analogous to specificity (precision). The edit distance measure penalizes overdivision of subtypes (different families) proportionately more than joining a few subtypes into large clusters. The VI distance measure takes cluster size into account, and errors in large clusters (affecting many sequences) contribute more to the distance than errors in small clusters.

#### iv) Performance

It is further useful to have a single performance measure that captures the commonly desired balance between high sensitivity and high specificity. Edit and VI distances are expressed as a percentage of their initial values for the given dataset by multiplying by the scaling factors  $c_e$  and  $c_v$ , respectively, where

$$c_e = 100/e_0 \text{ and } c_v = 100/v_0$$

Here,  $e$  is edit distance,  $v$  is VI distance, and  $e_0$  and  $v_0$  are the initial values of edit and VI distance, respectively. The former are calculated by putting each sequence in the dataset into a separate cluster. Then,

$$performance = \frac{2p + (100 - c_e \cdot e) + (100 - c_v \cdot v)}{4}$$

where  $p$  is the purity value expressed as a percentage. Since both edit and VI distance are measures of sensitivity but only purity is a measure of specificity, purity is here multiplied by a factor of 2.



### 4.2.3 Derivation of generic clustering granularity settings

In a preliminary analysis, the SFLD gold standard superfamilies (see Section 4.2.1) were used to assess (i) to what extent the partitionings produced by GeMMA at different levels of clustering granularity reflect known functional families and (ii) the variability in the sequence-to-function relationship among these superfamilies. Assessing the latter was necessary to confirm that the SFLD superfamilies would form a sufficiently diverse training dataset to derive generic granularity thresholds for family identification.

The sequences in each of the SFLD protein superfamilies were clustered in 20 consecutive rounds of GeMMA, respectively. Starting from individual sequences, the clustering granularity setting was decreased in a regular manner with each round. Subsequently, the partitionings obtained for each superfamily, at each level of granularity, were assessed using the four evaluation measures described in Section 4.2.1.1. Based on the results of this analysis, generic clustering granularity settings for protein family identification with GeMMA (and therefore  $DFX_{\text{unsupcr}}$ ) were derived. These settings correspond to those granularity levels at which the best family partitioning performance was observed, as averaged over all SFLD superfamilies (the training set), respectively.

In a second step, the analyses described above were extended to the SFLD-Gene3D dataset (see Section 4.2.1.2). This dataset contains superfamilies of protein domains, not whole proteins. It was therefore assessed whether the generic granularity setting derived for the whole-protein level would also apply for the domain level. Finally, the same protocol was used to process the Gene3D dataset, to measure how and if greater superfamily size leads to partitionings of lower quality. This was expected, as the GeMMA clustering method implements different heuristics (see Section 2.2.3) to speed up the clustering of large sequence datasets. The negative impact of these heuristics

on the accuracy of clustering is expected to increase with the size and diversity of the processed datasets.

#### 4.2.4 Benchmarking

The  $DFX_{\text{unsup}}^{\text{unsup}}$  protocol was benchmarked internally and against SCI-PHY using two distinct test sets of superfamilies. One was the small but high-quality SFLD gold standard dataset described in Section 4.2.1.1, corresponding to a high quality benchmark. The other was a larger set of functionally diverse Pfam families, corresponding to a benchmark showing the broad applicability of the method(s). This dual strategy was followed since, as of 2011, there exists no family dataset that is larger than the SFLD one and, at the same time, comparable in scope and equally well curated. Both benchmarking setups are described in the following.

##### 4.2.4.1 High quality benchmark

$DFX_{\text{unsup}}^{\text{unsup}}$  was first benchmarked and compared against SCI-PHY based on the SFLD gold standard dataset and two derived datasets (see Section 4.2.1). As the protocol involves a one-off training step to derive a generic clustering granularity setting (see Section 4.2.3), variation in the training dataset had to be taken into account in benchmarking. Further, a mixing of the training and benchmarking datasets had to be avoided. For these reasons, the performance of  $DFX_{\text{unsup}}^{\text{unsup}}$  was measured for each test superfamily (from the gold standard dataset) with the respective superfamily excluded in the training stage. In each case, the training set then comprised the remaining superfamilies, respectively. The overall performance of the protocol is measured as the average performance over all test superfamilies. This benchmarking strategy corresponds to a five-fold cross validation ('leave-one-out') approach.  $DFX_{\text{unsup}}^{\text{unsup}}$  was benchmarked against the SCI-PHY method only, as this was shown to be superior to several other unsupervised methods (Brown, Krishnamurthy et al. 2007).

To determine how much the family partitioning performance is increased when using a basic, entirely supervised approach instead of the above-described training procedure, a superfamily-specific clustering granularity setting for each SFLD superfamily was derived as well. This was based on comparing the different family partitionings derived with GeMMA clustering for each SFLD superfamily with the respective gold standard partitioning.

#### 4.2.4.2 Large-scale benchmark

DFX<sub>unsuper</sub> was further benchmarked on a larger (but lower-quality) dataset of protein domain families from Pfam, to test its broad applicability. Again, the performance of DFX<sub>unsuper</sub> was compared with that of SCI-PHY. The protein domain families in the Pfam database are known to contain different functional (sub)families (see below). They were therefore treated as superfamilies in the context of this benchmark (but will not be referred to as such below). EC numbers were used to identify known functional families, as these are similar in type and specificity (if not quality) to the family assignments in the SFLD benchmark.

1,741 families from Pfam 23.0 that contained at least two enzyme types annotated with EC numbers in UniProtKB were obtained from the Pfam website. These families comprised between 5 and 71,535 sequences each. The largest variety of EC numbers was found in PF00106, the short chain dehydrogenase family. This contains 87 different four-level EC numbers. The largest Pfam family for which SCI-PHY successfully produced a result contained 29,970 members; 15 larger families were therefore removed from the benchmark dataset. This appears to be a problem with memory allocation for SCI-PHY. Furthermore, due to the computational expense of this analysis, a representative set of 571 families was selected to constitute the final benchmark dataset. This representative set had approximately the same distribution of family size and diversity as had been found in the original

1,741 families. The mean number of different four-level EC numbers per family in this dataset was 3.6.

An average of 20.1% of the sequences in the 571 Pfam (super)families had an annotation, compared with an average of 64.1% of the sequences in the SFLD superfamilies that were used in the high-quality benchmark described above. Note further that the EC functional annotations for Pfam sequences are not expected to be as accurate as the SFLD annotations. Performance in the large-scale Pfam benchmark was assessed using the same measures (see Section 4.2.2) as in the high-quality benchmark. Further, the use of Pfam families meant that the input alignments for SCI-PHY were available.

### 4.3 Results and Discussion

The following sections present the results obtained in each of the analyses described in Sections 4.2.3 and 4.2.4. At the beginning stands the derivation of generic clustering granularity thresholds for both the whole-protein and domain levels using the SFLD gold standard dataset. The domain level thresholds are then confirmed by an extension of the analysis to whole Gene3D domain superfamilies. In the last part of this section, the results of two different benchmarks are presented: one small-scale but high-quality (in terms of the dataset used) and one large-scale but medium-quality benchmark.

#### 4.3.1 Derivation of generic clustering granularity settings

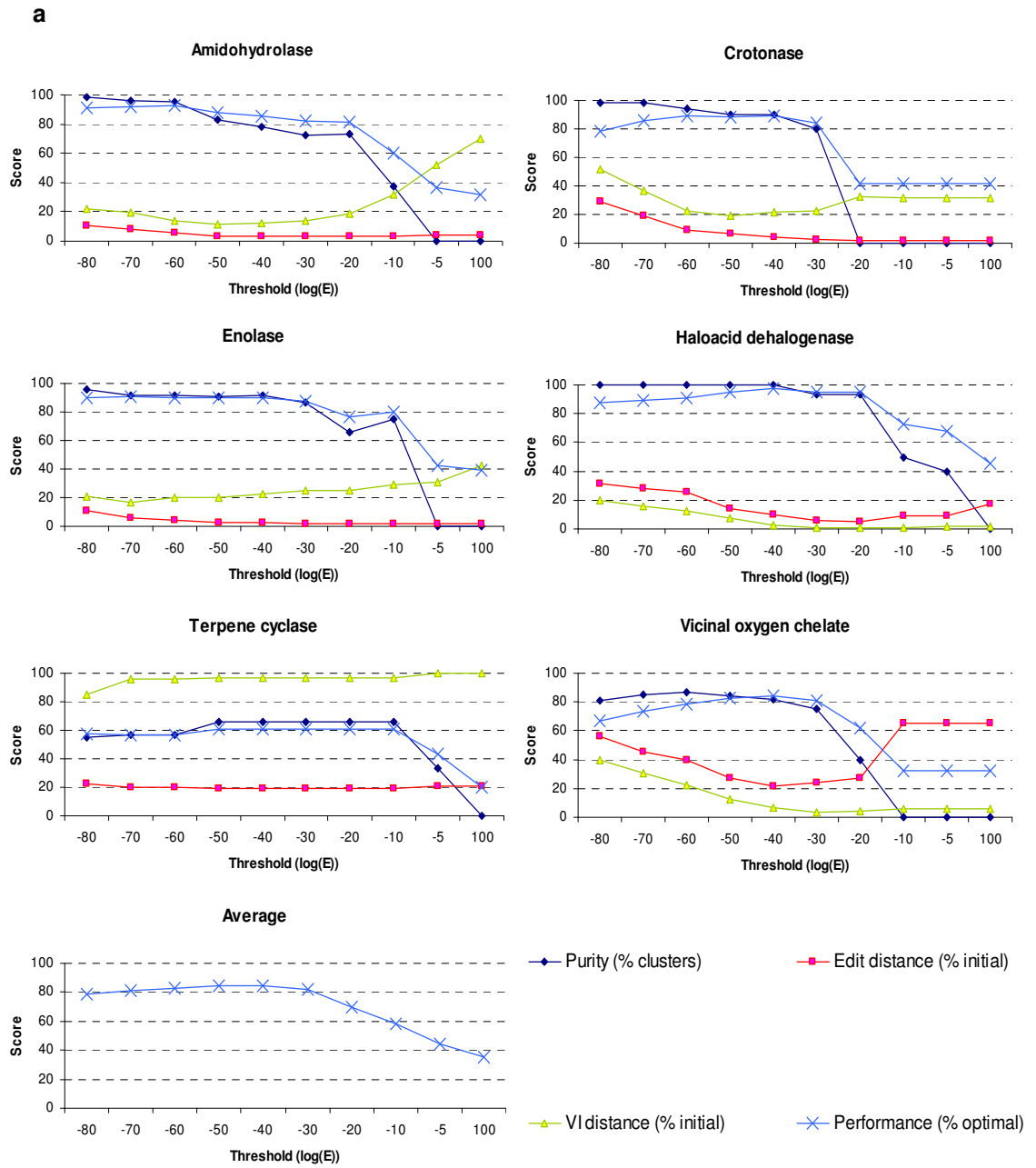
In a preliminary analysis, the sequences of each of the gold standard sequence superfamilies found in the Structure-Function Linkage Database (SFLD) were clustered in 20 consecutive rounds of GeMMA, respectively. The whole process was repeated for the domain superfamilies in the SFLD-Gene3D dataset, each containing the Gene3D domains of the protein sequences in the corresponding SFLD superfamily. Figure 4.1 shows the results for each superfamily and each level of clustering granularity, respectively. These

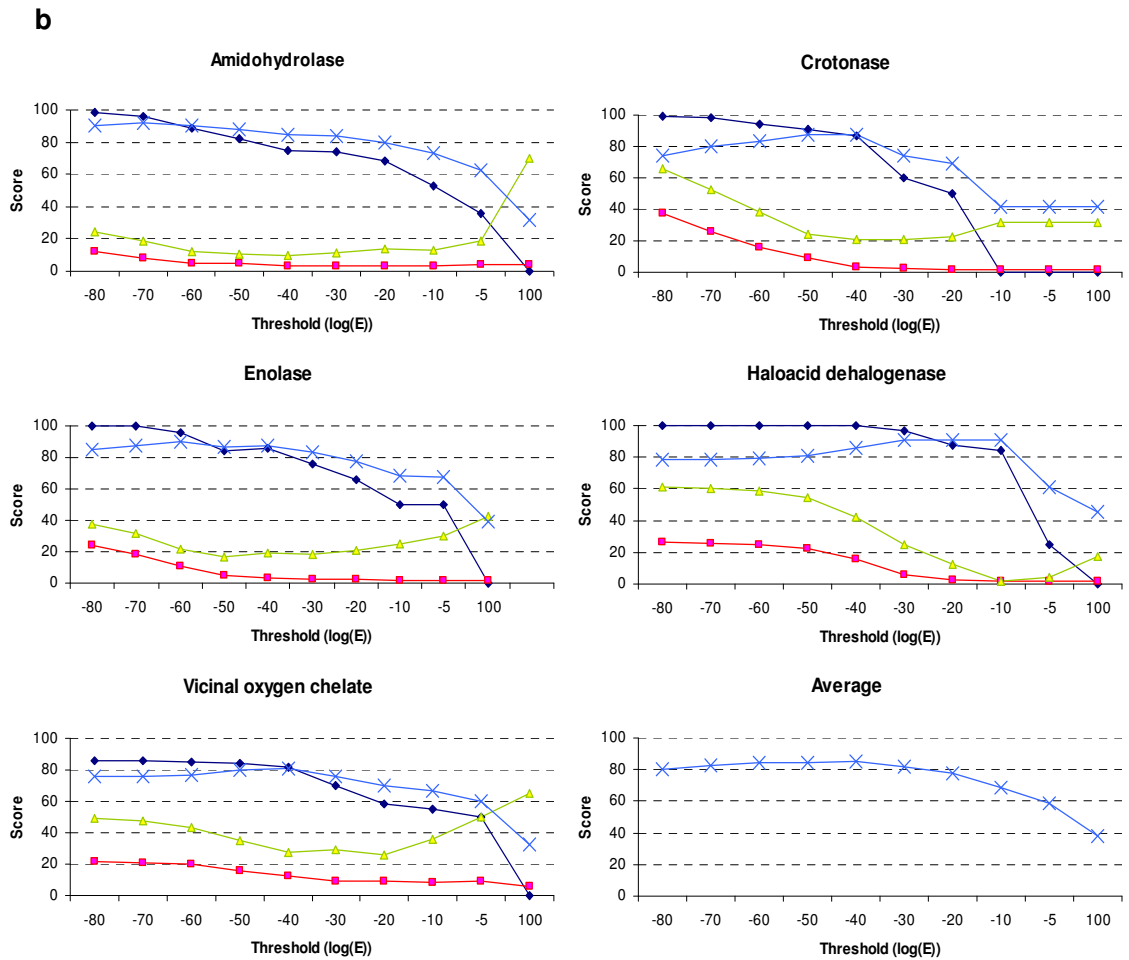
confirm that the SFLD superfamilies form a sufficiently diverse training dataset. This is illustrated by (i) the observed range of different peak performance levels and (ii) the highly variable behaviour of all values between individual superfamilies, depending on the level of clustering granularity, respectively. This translates to a high variability in the levels of sequence and function conservation between these superfamilies.

As a general trend, the purity of the produced clusters (specificity) decreases as the GeMMA E-value threshold is increased above a certain level, that is, as the level of clustering granularity decreases. At the same time, edit distance decreases (sensitivity increases) and VI distance decreases to a minimum and then increases again (sensitivity increases to a maximum and then decreases again). Purity is sometimes seen to decrease and then increase again, for example for the Terpene cyclase superfamily in Figure 4.1a. This can arise in two different ways. First, two impure clusters can be merged together so that the total proportion of impure clusters decreases. Second, a new pure cluster can be created that contains two annotated sequences that were previously found in separate clusters, each without further annotated member sequences (and therefore without an influence on the purity value); this leads to the overall proportion of pure clusters increasing.

As can be expected, the highest performance scores were obtained at different levels of clustering granularity for different superfamilies. For example, the peak for the Amidohydrolase SFLD superfamily in Figure 4.1a is at  $10^{-60}$ , while for the Haloacid dehalogenase family it is at  $10^{-40}$ . Average performance scores were thus calculated for the six SFLD protein superfamilies in Figure 4.1a and the five SFLD-Gene3D protein domain superfamilies in Figure 4.1b. The average peak performance is in both cases observed at an E-value of  $10^{-40}$ . The latter therefore serves as the generic clustering granularity setting for  $DFX_{\text{unsuper}}$ .

In Figure 4.1, the peak in the performance score for each superfamily is generally quite ‘blunt’. These observations support the use of a generic granularity setting to approximate functional domain families when high-quality annotations are lacking.





**Figure 4.1. Agreement of the partitionings produced by GeMMA clustering with known functional families in the SFLD and SFLD-Gen3D superfamilies.** This shows purity, edit distance, VI distance and overall performance for partitionings obtained at different levels of clustering granularity, when clustering (a) the protein superfamilies in the SFLD dataset and (b) the corresponding domain superfamilies in the SFLD-Gen3D dataset. Clustering granularity is indicated by the different E-value thresholds that define the individual GeMMA rounds (see Section 2.2.3.2).



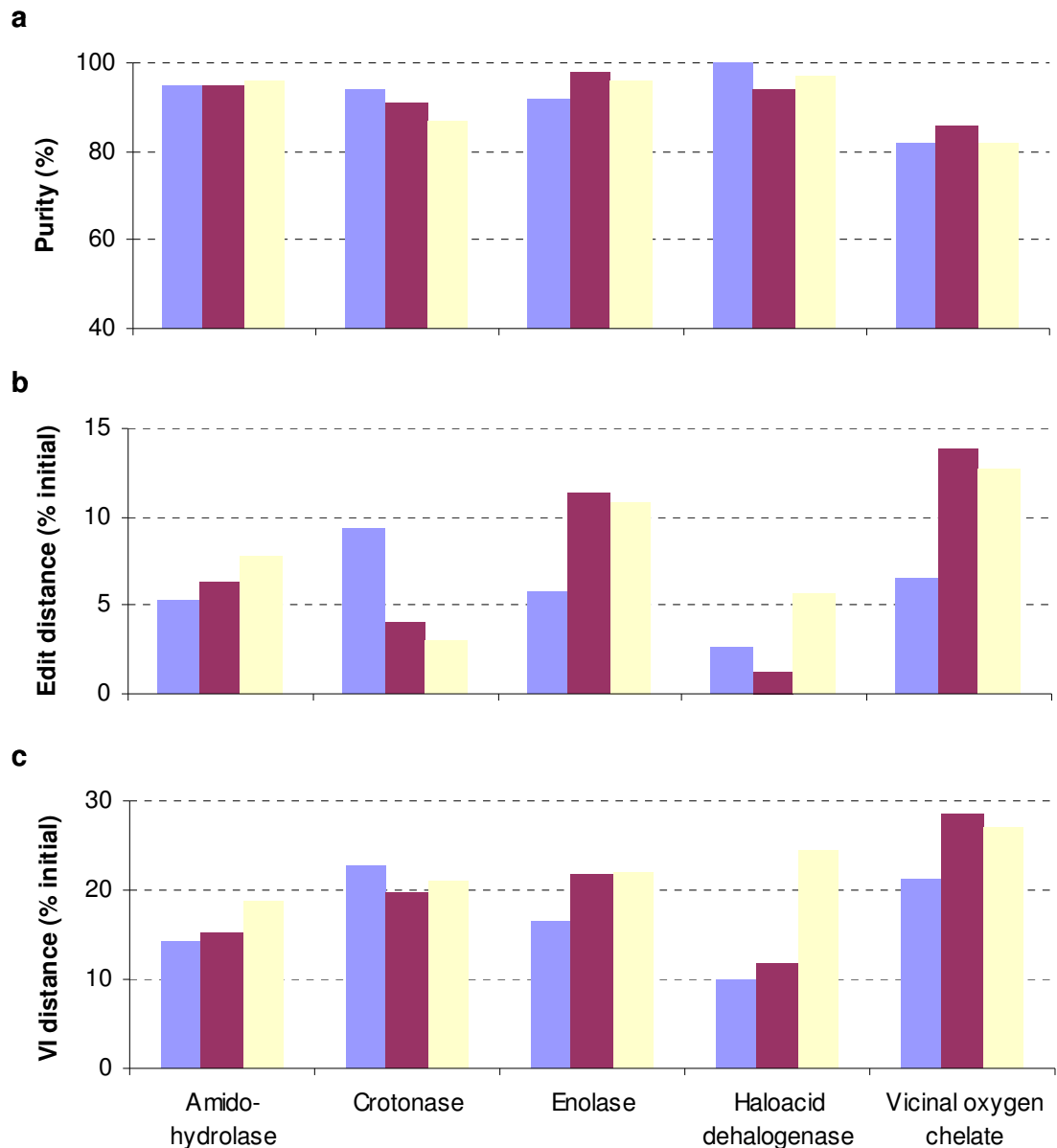
### 4.3.2 Analysis of entire Gene3D domain superfamilies

The GeMMA clustering method implements different heuristics to speed up the clustering of large sequence datasets (see Section 2.2.3). The negative impact of these heuristics on the overall hierarchical clustering result is expected to increase with the size and the diversity of the processed datasets. Therefore, the analyses described above were extended to the Gene3D dataset. Note that the whole Gene3D superfamilies in this set are considerably larger than the superfamilies in the two SFLD-only datasets (see Table 4.1). Reassuringly, however, the results obtained are comparable for all three datasets, as summarised in Table 4.2. Neither do the absolute peak performance values (which are found at different levels of clustering granularity for each superfamily) deteriorate significantly nor does the average clustering granularity level at which peak performance is obtained change between the small and the large domain datasets. Note that only the original SFLD annotations were used in all cases.

Figure 4.2 shows the behaviour of the three evaluation measures used when progressing from the SFLD protein to the SFLD-Gene3D and Gene3D domain datasets. These measures underlie the combined performance values in Table 4.2. Again, the results are very similar for all three measures, with no overall trend upwards or downwards exhibited. Overall, there is a small decrease observed in the peak performance scores when  $DFX_{\text{unsuper}}$  is applied to the much larger Gene3D superfamilies, with purity generally being a little lower and edit and VI distances being a little higher (see Figure 4.2).

**Table 4.2. Peak family partitioning performance when clustering the superfamilies in the three SFLD-derived datasets with GeMMA.** This shows, for the superfamilies in each dataset, the highest performance values observed when clustering the sequences in the respective superfamily with GeMMA. Each superfamily was clustered at 20 levels of clustering granularity, and the obtained partitionings were assessed for how well they match the known functional families in the superfamily. A perfect match corresponds to a performance score of 100.

Family	Dataset	Performance score	Granularity setting (log(E))
Amidohydrolase	SFLD	92.75	-60
	SFLD-Gene3D	92.25	-60
	Gene3D	91.75	-70
Crotonase	SFLD	89.25	-40
	SFLD-Gene3D	89.75	-40
	Gene3D	87.50	-40
Enolase	SFLD	90.75	-60
	SFLD-Gene3D	91.00	-60
	Gene3D	90.25	-60
Haloacid dehalogenase	SFLD	97.25	-20
	SFLD-Gene3D	94.00	-20
	Gene3D	91.25	-30
Vicinal oxygen chelate	SFLD	84.25	-40
	SFLD-Gene3D	82.75	-40
	Gene3D	81.25	-40



**Figure 4.2. Agreement of the best partitionings produced by GeMMA clustering with known functional families in the three SFLD-derived datasets.** The shown values for the SFLD protein (cyan), SFLD-Gene3D (magenta) and Gene3D (yellow) datasets correspond to the level of clustering granularity at which peak performance is reached (see Table 4.2). A good partitioning has high purity (maximum = 100%) and low edit and VI distances (maxima = the initial values).

The observed performance decrease when clustering whole Gene3D superfamilies is not large, with performance scores falling by no more than 6% in the worst case (see Table 4.3). This is true especially given that other methods such as SCI-PHY are not applicable to such large and diverse datasets. A possible explanation for the small decrease, apart from the effects of the GeMMA clustering heuristics (see above), is that the SFLD

superfamilies only contain carefully manually filtered sequences, while the Gene3D superfamilies include a certain amount of protein fragments and less rigorously validated sequences. Altogether, it can be concluded that there is sufficient sequence information in the functional core domains alone to reproduce the results that are obtained when analysing the whole-protein SFLD sequences.

### 4.3.3 Benchmarking

The  $DFX_{\text{unsuper}}$  protocol was benchmarked internally and against SCI-PHY using two distinct test sets of superfamilies. One was the small but high-quality SFLD gold standard dataset described in Section 4.2.1.1, corresponding to a high quality benchmark. The other was a subset of functionally diverse Pfam families, corresponding to a benchmark showing the broad applicability of the method(s). The results of both benchmarks are described in the following.

#### 4.3.3.1 High-quality benchmark

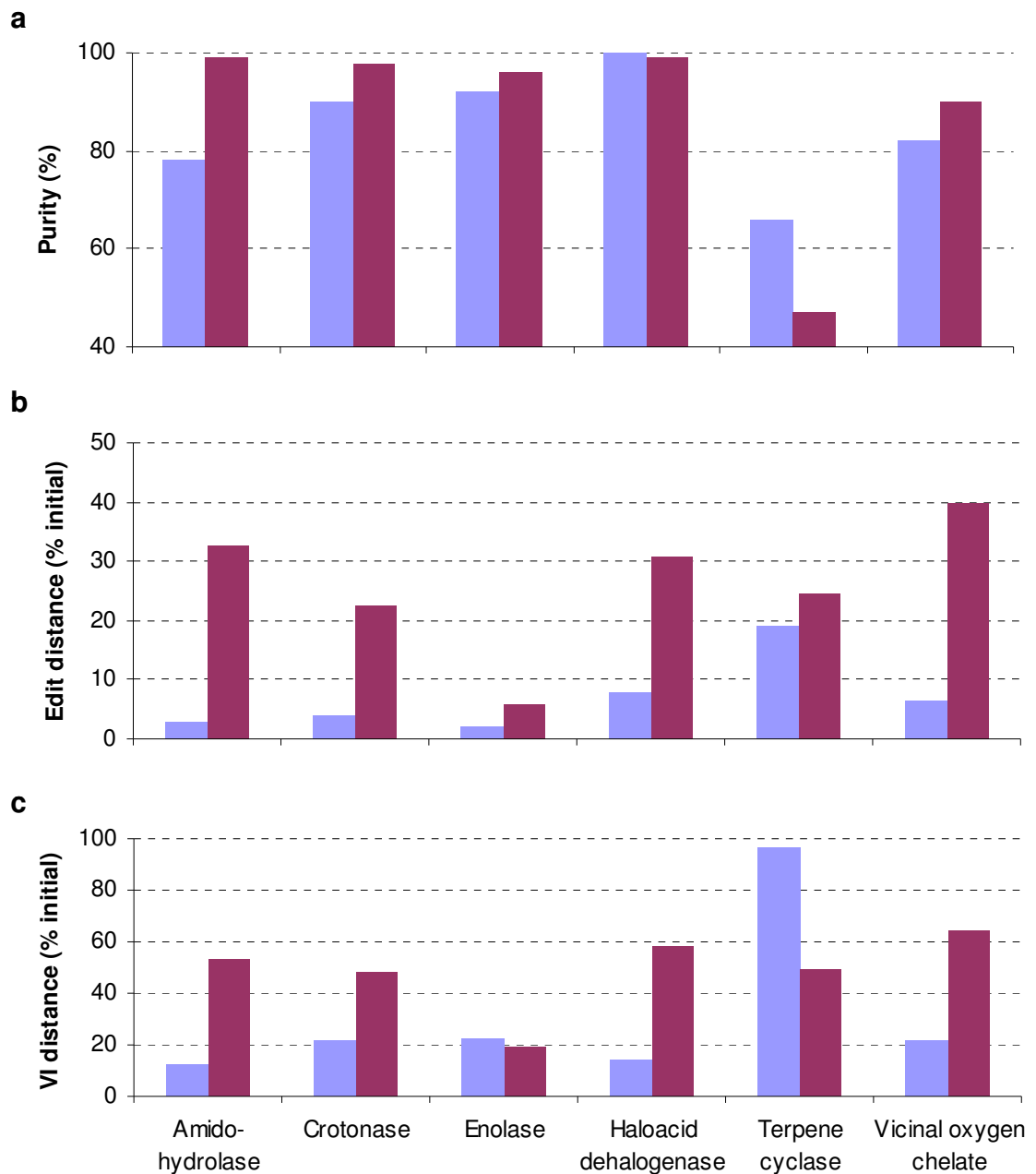
The performance scores achieved in the SFLD benchmark by SCI-PHY and  $DFX_{\text{unsuper}}$  are listed in Table 4.3. These results indicate that  $DFX_{\text{unsuper}}$  usually achieves a good balance between sensitivity and specificity, outperforming SCI-PHY in that respect. Only in a single case, the Enolase superfamily, the two methods are on a par. The main reason for this seems to be that SCI-PHY is optimised for high specificity (high purity) at the expense of rather low sensitivity (high edit and VI distances) compared with  $DFX_{\text{unsuper}}$ , as can be seen in Figure 4.3. The accordingly lower number of identified families for  $DFX_{\text{unsuper}}$  as compared with SCI-PHY in this benchmark is shown in Table 4.4.

**Table 4.3. Performance of DFX<sub>unsuper</sub> and SCI-PHY in cross-validation benchmarking on the SFLD dataset.** The GeMMA clustering granularity setting used for each superfamily was derived from training on the remaining superfamilies, respectively (see Section 4.2.3).

Family	Method	Performance score	Granularity setting (log(E))
Amidohydrolase	SCI-PHY	77.99	
	DFX <sub>unsuper</sub>	85.50	-40
Crotonase	SCI-PHY	81.29	
	DFX <sub>unsuper</sub>	89.00	-40
Enolase	SCI-PHY	91.70	
	DFX <sub>unsuper</sub>	90.00	-40
Haloacid dehalogenase	SCI-PHY	77.18	
	DFX <sub>unsuper</sub>	94.75	-50
Terpene cyclase	SCI-PHY	54.99	
	DFX <sub>unsuper</sub>	61.25	-40
Vicinal oxygen chelate	SCI-PHY	69.02	
	DFX <sub>unsuper</sub>	84.25	-40
Average	SCI-PHY	75.36	
	DFX <sub>unsuper</sub>	84.13	-41.66

**Table 4.4. Size of the family partitionings produced by DFX<sub>unsuper</sub> and SCI-PHY in cross-validation benchmarking on the SFLD dataset.** The values shown for each superfamily and method correspond to the partitionings assessed in Table 4.3. Singletons are clusters with only a single member.

Family	Method	Clusters	Singletons
Amidohydrolase	SCI-PHY	638	364
	DFX <sub>unsuper</sub>	100	47
Crotonase	SCI-PHY	320	149
	DFX <sub>unsuper</sub>	141	75
Enolase	SCI-PHY	201	75
	DFX <sub>unsuper</sub>	56	31
Haloacid dehalogenase	SCI-PHY	332	181
	DFX <sub>unsuper</sub>	161	110
Terpene cyclase	SCI-PHY	22	1
	DFX <sub>unsuper</sub>	6	0
Vicinal oxygen chelate	SCI-PHY	302	163
	DFX <sub>unsuper</sub>	138	82



**Figure 4.3. Agreement of the family partitionings produced by DFX<sub>unsuper</sub> and SCI-PHY with known functional families in cross-validation benchmarking on the SFLD dataset.** The shown values for DFX<sub>unsuper</sub> (cyan) and SCI-PHY (magenta) correspond to the generic clustering granularity setting used (see Table 4.3). A good partitioning has high purity (maximum = 100%) and low edit and VI distances (maxima = the initial values).

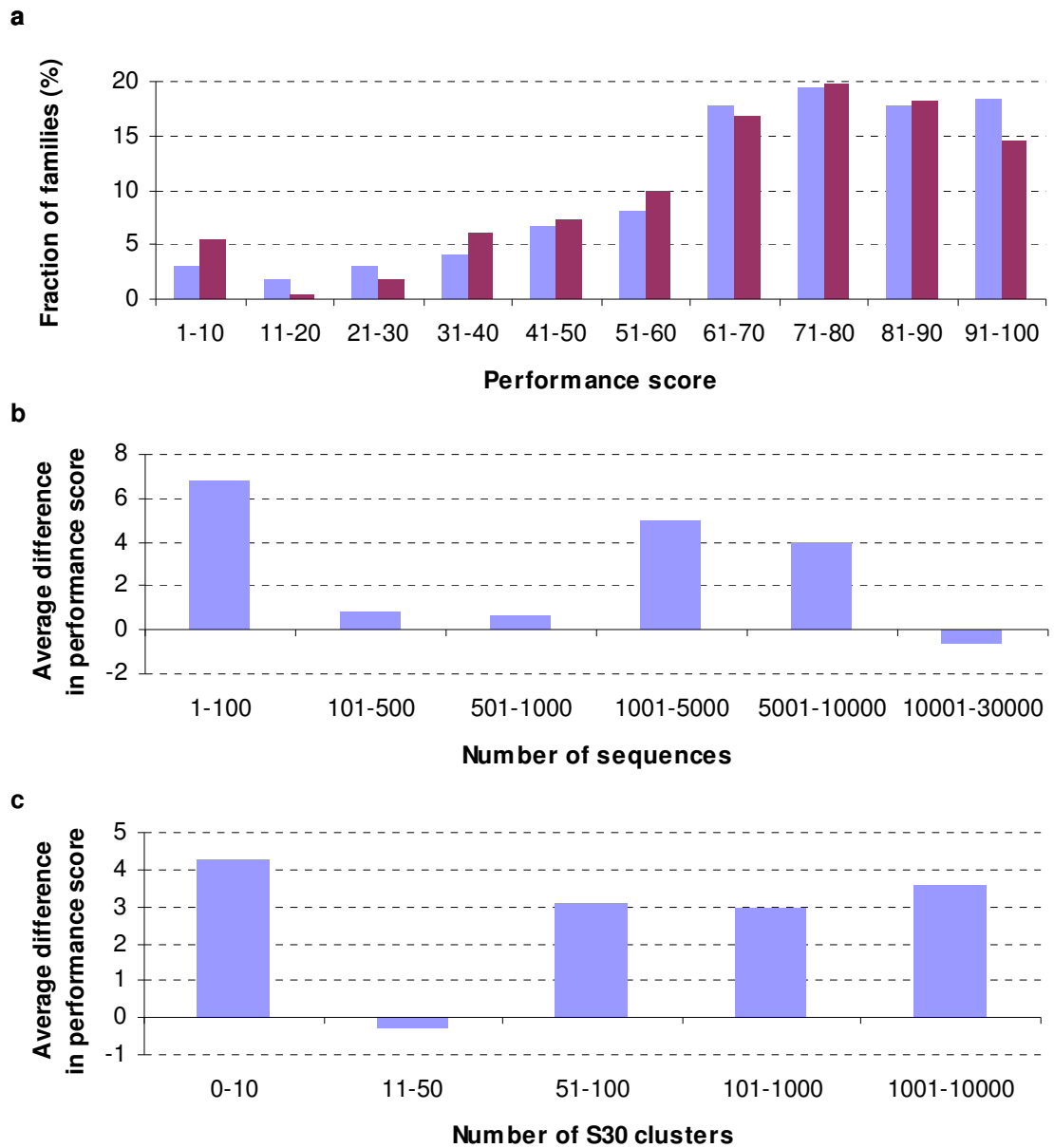
#### 4.3.3.2 Large-scale benchmark

For both DFX<sub>unsuper</sub> and SCI-PHY the observed performance is similar to that seen in the SFLD benchmark when benchmarking on this much larger and more diverse set of domain (super)families. The majority of performance scores in the Pfam benchmark are found in the top three bins in Figure 4.4a.

Since the total sums of the performance scores for each method are very similar to each other, neither method is clearly superior to the other (the total for  $DFX_{\text{unsuper}}$  is 2.8% higher than that for SCI-PHY). Further, the difference in the performance score of  $DFX_{\text{unsuper}}$  and SCI-PHY was plotted against Pfam family size (see Figure 4.4b) and diversity (Figure 4.4c), to test whether either has a differential effect on the relative performance of the methods. It can be seen that this is not the case.

The Pfam families in this benchmark often contain sequences with different annotated functions, in the form of different EC numbers. Both  $DFX_{\text{unsuper}}$  and SCI-PHY are effective in subdividing these families into functionally pure (sub)families (see Figure 4.5), with SCI-PHY achieving a slightly higher proportion of approximately 3% overall. Further, the transfer of functional annotations within the produced functional families can significantly increase the annotation coverage of the parental Pfam families (see Figure 4.6). In terms of sensitivity, both methods show the advantage of using a profile linkage approach (see Section 2.2.1) when clustering the sequences, as opposed to complete linkage clustering based on pair-wise sequence comparisons (at a ‘safe’ pair-wise sequence identity threshold of 60%). The latter is a common target selection strategy in structural genomics. Further, the greater sensitivity of  $DFX_{\text{unsuper}}$  compared with SCI-PHY results in greater post-transfer annotation coverage, albeit risking a small decrease in specificity (see above). That is, a minor fraction of the families in which annotations have been transferred may comprise more than one function.





**Figure 4.4. Performance of DFX<sub>unsuper</sub> and SCI-PHY in the Pfam benchmark.** (a) Distribution of performance scores for DFX<sub>unsuper</sub> (cyan) and SCI-PHY (magenta). Also shown is the average difference in the performance score between DFX<sub>unsuper</sub> and SCI-PHY (DFX<sub>unsuper</sub> score minus SCI-PHY score) depending on (b) family size and (c) family diversity (estimated as the number of Gene3D 7.0 S30 clusters in the family).

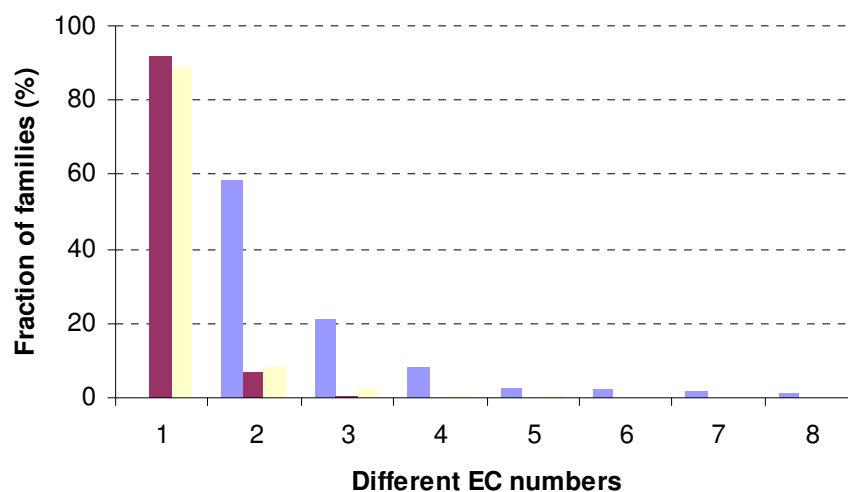


Figure 4.5. Functional conservation in the Pfam benchmark families and the produced SCI-PHY and DFX<sub>unsuper</sub> families. Shown is the respective proportion of families that contain the indicated number of different EC annotations (plotted up to a number of eight different ECs) for the initial Pfam (cyan) and the produced SCI-PHY (magenta) and DFX<sub>unsuper</sub> (yellow) families.

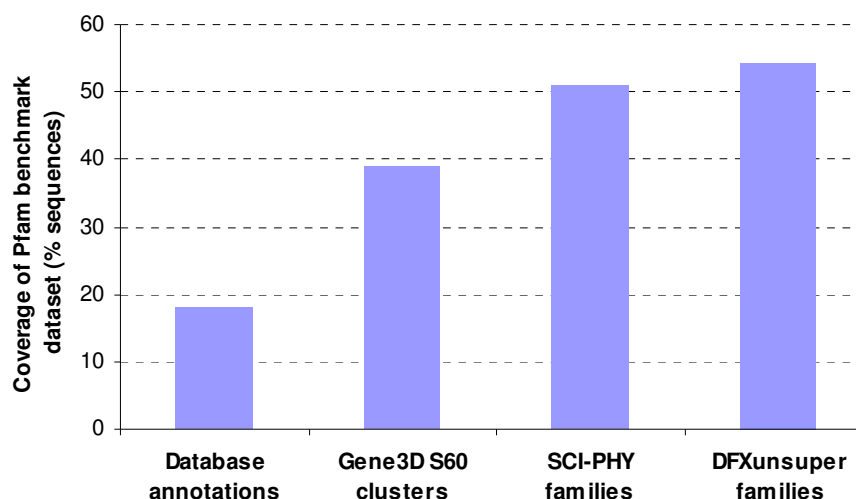


Figure 4.6. Transfer of functional annotations within the Pfam benchmark families. This illustrates the initial and post-transfer annotation coverage of the Pfam benchmark families when using Gene3D S60 clusters and SCI-PHY or DFX<sub>unsuper</sub> functional families.

## 4.4 Conclusions and future work

The future of the unsupervised protocol in DFX is discussed in conjunction with other aspects that concern both DFX family identification protocols in Sections 3.4 and 7.3. Two recent use cases of DFX<sub>unsuper</sub> and two obvious next steps in its development are discussed in the following.

#### 4.4.1 Recent use cases

Families identified by  $DFX_{\text{unsuper}}$  were used in a recent study (Dessailly, Redfern et al. 2010) on the evolution of structure and function in the large and diverse HUP domain superfamily (see Section 5.1.4). Nine Functional Sub-Groups (FSGs) for the superfamily were defined manually in this work, based on extensive literature and database review, and the 85 non-redundant CATH structural domains available for the HUP superfamily were each assigned to one of the those FSGs. Three of the nine FSGs comprised non-enzymatic domains. Notably, the definition of the manually defined FSGs was based on a domain family concept similar to the DFX one (see Section 0). This primarily refers to the amount of functional ‘leeway’ in these groups, which, for example, allows for different enzyme substrate specificities within a group as long as the overall reaction mechanism is conserved.

The above study reports that most of the 85 HUP superfamily domains under analysis were separated into families of perfectly conserved (parent protein) function by  $DFX_{\text{unsuper}}$ . Only in four cases were sequences with matching function found in different families; in a single case, the opposite was observed. However, even for these special cases, potential functional reasons are put forward (Dessailly, Redfern et al. 2010). On the somewhat coarser FSG level, a tremendous overdivision of individual FSGs by  $DFX_{\text{unsuper}}$  is reported. This can be expected, based on the training of the method on functionally perfectly conserved families from the SFLD (see Section 4.2.3) and, generally, the use of a generic clustering granularity setting. The study further finds that the alignments of the  $DFX_{\text{unsuper}}$  families exhibit strongly conserved residue patterns that either correspond to catalytic or ligand-binding residues. In particular, in 2 out of 11 families for which reliable residue information was available, catalytic residues were found more conserved than ligand-binding residues, with the latter in turn less conserved than all further (non-characterised) residues. The study concludes with

suggesting that  $DFX_{\text{unsuper}}$  families could - together with additional structural or functional information - serve to generate a family classification that would match the manually defined FSGs perfectly. This is essentially the strategy followed in the later developed  $DFX_{\text{super}}$  protocol (see Chapter 5), apart from the use of structural information (so far).

The  $DFX_{\text{unsuper}}$  protocol has also recently been applied to Structural Genomics target selection for the Midwest Center for Structural Genomics (MCSG), in order to improve the coverage of structurally underrepresented superfamilies in the second phase of the Protein Structure Initiative (PSI-2) (Dessailly, Nair et al. 2009). It was demonstrated in Lee, Rentzsch et al. (2010), by the example of eleven domain superfamilies, that large structurally unrepresented clusters of sequences, as identified with  $DFX_{\text{unsuper}}$ , can be exploited for this aim. In this context, it was also shown that such clusters identify many more targets for homology modelling that produce acceptable models than are found by using a traditional approach (sequence comparison and selection of targets that share at least 30% sequence identity with the available template structures).

#### 4.4.2 Future work

As an obvious next step,  $DFX_{\text{unsuper}}$  will be used to identify putative functional families in all Gene3D superfamilies that cannot be processed in supervised mode in the next run of the DFX pipeline. These ~25% of all superfamilies that are not associated with high-quality protein annotations at all are mostly small, and were not included in the first run of DFX. However, in the quantitative assessment in Chapter 6, a ‘light’ version of the  $DFX_{\text{unsuper}}$  protocol (not based on exhaustive clustering) is compared with  $DFX_{\text{super}}$  in terms of the method’s theoretic ability to identify families in more than 400 Gene3D enzyme superfamilies. Together with the analyses presented above,

the results of this assessment provide an estimate of the performance that can be expected, on average, for  $DFX_{\text{unsuper}}$ .

The  $DFX_{\text{unsuper}}$  method should be benchmarked against further (recently published) unsupervised methods for family identification. Importantly, this must take into account both performance and applicability to large datasets, in conjunction with runtime. In a first step, the alignment-free CLUSS method in its second incarnation (Kelil, Wang et al. 2008) should be tested on the SFLD superfamilies. Preliminary tests suggest that it may be on a par with, or outperform, SCI-PHY and/or  $DFX_{\text{unsuper}}$  in some cases. However, less encouragingly, a recent study reports poor performance for CLUSS and runtimes of up to 55 hours (Frech and Chen 2010). Other candidates against which to benchmark are more recently published protocols that are based on graph-based clustering, such as those reviewed in Section 2.1.4. A generic clustering granularity threshold that is optimised for family identification may be derived for these methods, as it is currently done for  $DFX_{\text{unsuper}}$  (see Section 4.2.3).

## Chapter 5. Supervised protein domain family identification in DFX

After the unsupervised family identification protocol for the DFX pipeline had been developed, a second, more sophisticated protocol was implemented. This takes into account available knowledge on whole-protein function and uses this information to guide the domain family identification process. It is therefore a supervised protocol. In this manner, it produces functional domain families that adhere to the domain family concept introduced in Section 0 with increased precision and control compared with the unsupervised protocol. Equally importantly, the supervised protocol makes possible the processing of the largest and most diverse domain superfamilies with 100,000s of sequences. As these superfamilies contain the most promiscuous domains that appear in a large number of different domain architectures and functional contexts, the supervised family identification protocol (DFX<sub>super</sub>) is the most important part of the DFX pipeline.

The background section first reviews existing methods for supervised protein family identification and approaches to derive domain-specific annotation data. After a detailed discussion of Gene Ontology annotations in the context of protein domains, it concludes with the introduction of two ancient protein families. The superfamilies that contain the catalytic domains of these proteins serve as examples in the discussion section, respectively. The concepts section introduces several concepts that are used by DFX<sub>super</sub> to capture the functional annotation of domain sequences and sequence clusters, when identifying domain families. The identification process itself is then described in the implementation section. Finally, a detailed qualitative analysis of the families produced by DFX<sub>super</sub> is performed based on the two example domain superfamilies introduced earlier. A quantitative assessment, in conjunction with the unsupervised protocol, follows in Chapter 6. The present chapter closes with a summary of the benefits and potential caveats

of the developed protocol and a discussion of suggested further work. Additional points that may affect both DFX family identification protocols are discussed in Chapter 7.

## 5.1 Background

In the following, existing supervised (protein) family identification methods are reviewed first, complementary to the review of unsupervised methods in the previous chapter. The same is then done for existing approaches to derive domain-specific function annotation data. Since  $DFX_{\text{super}}$  uses the Gene Ontology protein annotation system (see Section 1.3.1), this system is specifically discussed in the context of protein domain function thereafter. Finally, two multi-domain protein families whose members contain domains from two evolutionarily ancient, functionally diverse superfamilies are introduced. These serve as examples when characterising the families produced by  $DFX_{\text{super}}$  in Section 5.4.1.

### 5.1.1 Existing supervised family identification methods

Supervised family identification protocols combine sequence clustering with supervised clustering evaluation techniques (see Section 2.1.3.2). Currently existing methods come in two flavours: those based on hierarchical and those based on graph-based clustering approaches. The known assignments of all or part of the sequences to functional classes that are used to select appropriate settings for the respective clustering granularity parameters (i.e., to stop the clustering) are in all cases simple annotation types, such as EC numbers or manually assigned family numbers. Complex annotations, such as sets of GO terms assigned to individual sequences, cannot be used. As the very task of translating from such complex assignments into simple class (family) assignments is the core functionality of  $DFX_{\text{super}}$ , and as it works on protein domains instead of whole-protein sequences, the existing methods are only reviewed briefly in the following.

In principle, (semi-)supervised family identification methods share the goal of profile-based function prediction methods: to group sequences with known function in a homogenous manner, to be able to assign uncharacterised sequences to the respective groups (and functions) thereafter. Existing profile-based methods for the prediction of enzyme function follow a two-step approach. They first generate highly specific enzyme family profiles to which unknown sequences are subsequently assigned. While the CatFam method (Yu, Zavaljevski et al. 2009) was shown to outperform its predecessors EFICAz (Arakaki, Huang et al. 2009) and PRIAM (Claudel-Renard, Chevalet et al. 2003) in terms of assignment accuracy, it lacks a publicly available server. The latter is also the case for the recently published ModEnzA method (Desai, Nandi et al. 2011), which was claimed to perform better than all the above methods. Another recent member of this strain of methods is BrEPS (Bannert, Welfle et al. 2010), which was only benchmarked against PRIAM and shown to be, on average, on a par with it. All these methods use hierarchical clustering approaches to group protein sequences by EC number (i.e., family) and, based on this, create one or several profile HMMs to represent each family. The breadth of these profiles (determined by the size and number of the underlying sequence groups) is in each case optimised by testing how well a given model can differentiate between class members and non-members in a test set of (more or less high-quality annotated) enzyme sequences.

### 5.1.2 Existing methods to derive domain-specific annotations

There currently exists only a single regularly updated mapping of function annotation terms to protein domain families, the InterPro2GO mapping (Camon, Barrell et al. 2005), which is also the only mapping that is integrated with a family resource, InterPro. As a meta-resource, InterPro integrates both whole-protein and protein domain classification (see Section 5.1.2), and does not itself aim to generate domain families at any particular level of granularity. Rather, for classification on the domain level it relies on six of its currently



eleven member databases: Pfam, ProDom, SMART, TIGRFAMs, SUPERFAMILY and Gene3D (see Section 1.5.2.2).

For creating the InterPro2GO mappings on the domain level, curators review the annotations of the SwissProt protein sequences that are assigned to a given InterPro domain entry. They then associate the most specific functions that are deemed to be shared by all sequences with the entry (family) as a whole<sup>15</sup>. The same process is followed for InterPro protein family entries. Since the GO annotation system is used, identifying the functions shared by all sequences in a family is (theoretically) trivial: the ‘last common ancestor’ parent terms can be readily identified in each of the three GO branches. Resource-specific subsets of InterPro2GO are available too; for example, Pfam2GO. All mappings are currently updated on a monthly basis.

The most important responsibility of the InterPro2GO curators is not the identification of common ancestor GO terms (which can be largely automated), but rather the decision as to which of those functions are related to the domain (family) in question and which are mediated by other domains in the proteins harbouring this domain. The inherent uncertainty in making this decision (see Section 3.2.1) and the often coarse level of functional granularity in Pfam families (which nucleate many InterPro domain entries) together lead to relatively sparse, coarse and sometimes inconsistent InterPro2GO annotations. For example, as of September 2011, only a fraction of all InterPro protein and domain family entries that are associated with the HAD superfamily of hydrolase domains (see Section 5.1.4) are assigned the ‘hydrolase’ term (GO:0016787). Most importantly, the ‘Haloacid dehalogenase -like hydrolase’ domain family entry (InterPro IPR005834) that specifically represents the catalytic hydrolase domain does not have this annotation. Annotations can further be entirely missing for entries where no common ancestor term below the respective GO DAG root term can be

---

<sup>15</sup> <http://www.ebi.ac.uk/GOA/InterPro2GO.html>

identified, owing to inconsistent, incomplete or erroneous annotation of the SwissProt sequences assessed.

Several other attempts at deriving domain-specific annotations have been made that have not yet been integrated with a domain family resource. A common characteristic of these methods is that they essentially evaluate (explicitly or implicitly) a matrix that captures co-occurrences of GO terms and protein domains. For example, if a given term is found with all proteins that contain a certain pair of domains but never with proteins that contain only one of those domains, it would be assumed that the combination of the two domains is both necessary and sufficient to give rise to the respective functionality.

Schug and colleagues used a rule-based approach to predict domain-specific GO annotations for individual domain families (Schug, Diskin et al. 2002) defined in the ProDom and CDD resources (see Sections 1.5.2.2 and 3.1.1, respectively). Their protocol works as follows. Initially, all sequences in a training dataset of GO-annotated sequences are scanned against all ProDom and CDD domain families, using BLAST and RPS-BLAST, respectively. To derive domain-specific annotations for a given domain entry, its BLAST hit list is first sorted by the associated P-values, from low to high. Different types of ‘rules’ are then created and associated with certain P-value thresholds. For example, a ‘single function’ rule is generated when the N first hits in the list have only a single, shared most specific GO annotation. The rule is associated with a P-value threshold that corresponds to the P value observed for the hit at position N. Since the sequences hit are usually associated with several different GO terms, at different levels of specificity (depending on domain architecture and level of experimental characterisation), the other rule types are more sophisticated and try to take into account the possibility of missing annotations and varying annotation granularity (‘consensus’ rules). For example, a ‘consensus ancestor’ rule associates a given domain family with the

GO DAG ancestor term that is shared by the first  $N$  sequences in the BLAST hit list, as long as this is reasonably specific. Rules were created in this manner for all ProDom and CDD domain families that had a non-empty BLAST hit list, using a training set of GO-annotated yeast, fly and mouse proteins. It was subsequently assessed how many domain families can be associated with GO terms in this manner, and this coverage was compared with that of the InterPro2GO mappings for ProDom and Pfam domain families. Interestingly, the results were found to be complementary in both cases. The absolute coverage of the developed method was slightly lower than that of InterPro2GO in the case of ProDom and slightly higher for Pfam families.

One limitation of the above-discussed approach is that the (joint) functions of consistently co-occurring domains cannot be resolved. A more sophisticated framework was therefore employed in the GOTrees method (Hayete and Bienkowska 2005). This first models the domain content of all proteins in the training set in the form of a binary presence/absence vector, where the number of dimensions is the total number of defined Pfam domains. Protein annotation is thus translated into a classification problem, where individual domain vectors are mapped to GO term labels. For this classification, a decision tree is generated for each individual GO term that best separates those proteins (i.e., single domains or domain combinations) that are assigned the term from those that are not. Using the decision trees derived from a training set of annotated human, mouse and yeast proteins, the authors annotated all proteins in a test set of fly and worm proteins (both sets were subsets of SwissProt). In this benchmark, the method achieved a considerable increase in coverage over InterPro2GO, with a slight decrease in specificity. GOTrees was not compared with the simpler method described above.

Two alternative protocols that mimic and extend the InterPro2GO approach (but without any manual curation), respectively, were later presented by

Forslund and Sonnhammer (Forslund and Sonnhammer 2008). The simpler of the two, MultiPfam2GO, is a straightforward generalisation of the principle behind InterPro2GO to multi-domain sets: the sparsest possible set (Pfam-A domain combination) that consistently occurs in UniProt UniRef50 proteins associated with a given GO term (set) is associated with that term (set). Notably, while putatively beneficial for protein function prediction, this association does not necessarily imply that the respective domain set gives rise to the function(s) in question; single-domain proteins (and their GO annotations) are generally required to construct such unequivocal, ‘strong’ domain to function relationships. To account better for sparse and missing protein annotations, as well as missing domain assignments, the authors introduce a second, probabilistic method. This implements a naïve Bayesian network classifier (Friedman, Geiger et al. 1997) to associate domain sets with GO terms (or term sets), assuming that all possible domain combinations appear independently of each other; the latter is not the case but serves to simplify the algorithm, as the authors state. In that manner, a mapping was created between more than 400 distinct Pfam-A domain combinations and 186 different GO terms, with associated P-values. In ten-fold cross-validation on the protein dataset used, the method showed an overall (if small) performance gain when compared with BLAST, given that only remote homologues with at least one Pfam-A domain were considered. While the authors state that a direct benchmark of the above-discussed GOTrees method against theirs is difficult to construct, the ‘raw’ sensitivity and specificity values obtained in the individual benchmarks (in both studies) indicate superior performance of the Pfam2GO-derived method, which is also less demanding in terms of computational resources (more scalable).

The SCOP2GO method (Lopez and Pazos 2009) focuses on the (SCOP) domain fold level, and tries to associate individual structural domains (folds) occurring in whole-protein PDB structures with specific GO terms. To this end, the fold composition of each PDB chain with a given GO term is first

collected in a matrix. The following iterative protocol is then applied. First, the fold  $F$  with the highest occurrence count in the matrix is associated with (deemed to be responsible for) the respective function (term). Second, all domains of this fold type in all chains are labelled as associated with that function. If another fold co-occurs with  $F$  in almost all (a heuristic fraction of 97%) chains that contain  $F$ , the respective domains are labelled accordingly. All other domains in these chains are labelled as non-associated with the term. For the second iteration,  $F$  is determined as the second-most frequently occurring fold in all chains associated with the GO term in question. The protocol iterates until no domain in any of the chains is left unlabelled, that is, until all occurring folds have been assessed. For each of the folds the method then calculates a P-value (based on a hypergeometric distribution) that states the likelihood of the fold being responsible for the function (GO term) in question. In that manner, multiple terms can be probabilistically associated with the same fold, and vice versa. The authors annotated almost 40,000 SCOP domains with one or several GO MF terms, from a set of 948 distinct terms that are found at least two steps below the root node in the GO MF DAG. The method was compared with the InterPro2GO mapping for SCOP domain families. Importantly, this comparison showed that the InterPro2GO annotations often did not refer to individual domains but to whole proteins; this is somewhat surprising given the manual curation effort behind these annotations.

### 5.1.3 Protein domain function and the Gene Ontology

At the core of the DFX supervised family identification protocol stands the assessment of protein domain sequence clusters for functional coherency, based on whole-protein GO annotations. The GO annotation system is described in detail in Section 1.3.1. Generally, it is more complex than older systems, particularly the EC system, but it can also capture much richer information on protein function, including non-enzymatic activities. The

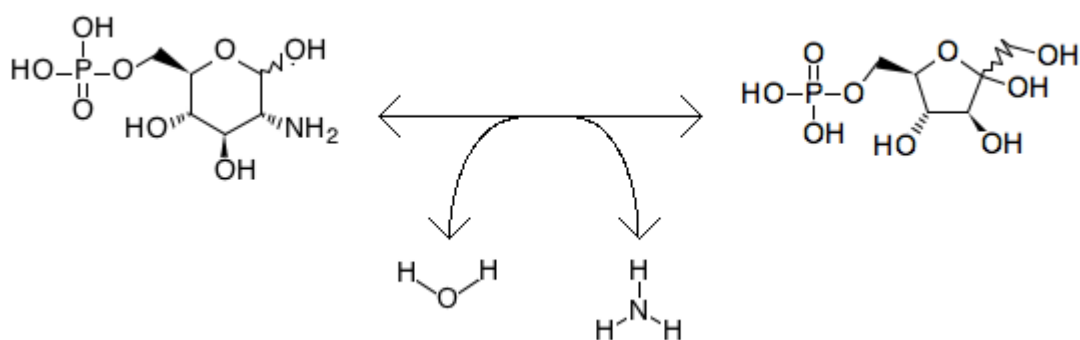
different rules implemented in  $DFX_{\text{super}}$  are based on a set of observations on how GO molecular function terms are used in protein function annotation.

Within the GO annotation system, only the MF term type is used to describe the specific activity or reaction chemistry of proteins (see Section 1.3.1). Importantly, MF terms can be used in isolation to describe fully this most important aspect of protein function; no additional terms from the BP or CC branches of GO are required. To translate between whole-protein and (putative) domain annotation in a heuristic manner (see Section 5.1.3), the supervised protocol distinguishes ‘essential’ MF terms from ‘non-essential’ MF terms, and ‘related’ pairs of MF terms from ‘unrelated’ pairs. The following paragraphs explain these dichotomies, with a particular focus on the function(s) and corresponding MF annotations of multi-domain proteins and their individual domains.

As outlined above, GO MF terms are very similar to EC annotations with respect to the type of protein functions they describe, namely specific biochemical activities. Unlike EC numbers, however, they can be used to describe non-enzymatic functions too. Further, GO terms are defined and used in a more ‘atomistic’ manner than EC numbers. For example, the overall molecular function of an enzyme as described by an EC number can often be split into its substrate binding, cofactor binding and chemical reaction aspects using three different GO MF terms.

Many proteins are currently annotated with GO MF terms according to a mixture of the ‘holistic’ (EC) and the atomistic (GO) paradigms. Particularly enzyme sequences are often assigned a single essential GO MF term describing their overall function, in conjunction with one or more additional, non-essential MF terms that focus on specific aspects of this function. Non-essential MF terms can be defined as neither necessary nor sufficient to describe the function of a protein as a whole. An example is given in Figure

5.1. As almost all other deaminases (enzymes that catalyse the removal of an amine group from a molecule), Glucosamine-6-phosphate (GlcN6P) deaminase uses water as a co-substrate, that is, it catalyses a hydrolytic deamination reaction. This means that the ‘hydrolase activity’ term (GO:0016787) that is associated with many of these proteins describes a certain mechanistic aspect of the overall reaction, whereas the ‘deaminase activity’ term (GO:0019239) and its more specific child term ‘glucosamine-6-phosphate deaminase activity’ (GO:0004323) describe the (net result of the) overall reaction. In that sense, the former term is non-essential whilst the latter terms are essential. Unsurprisingly, a part of the GlcN6P deaminase proteins is, as of October 2011, lacking the GO:0016787 annotation; this includes three manually reviewed SwissProt entries (e.g., UniProt Q8AB53).



**Figure 5.1.** The hydrolytic deamination reaction catalysed by Glucosamine-6-phosphate deaminase. The corresponding enzyme annotation is EC 3.5.99.6; the corresponding GO MF annotations are GO:0004323 and GO:0016787 (see main text). The reaction diagram was taken from KEGG.

Based on the current GO MF term definitions and the use of these terms in protein annotation, as outlined above, a single MF term is usually sufficient to describe the overall function of a single-domain protein, and therefore to judge whether two such proteins, annotated with MF terms, are functionally identical or similar. An example is a single-domain enzyme that combines a substrate binding site and an active site in one and the same domain. If the protein has two GO MF terms annotated, one can be expected to describe the overall function of the protein (the enzymatic activity) and the other an individual aspect of this function (the binding of the substrate). The two

terms are therefore related, but one is essential and the other non-essential (see above). Whenever a specific domain or protein can carry out more than a single function, the number of expected essential GO MF terms would then equal the number of observed functions. Such a functional ‘moonlighting’ of proteins has been observed in several cases (Jeffery 1999; Huberts and van der Klei 2010).

The relationship between protein function (annotation) and protein sequence is often more complex in multi-domain proteins. Here, each domain can encode a distinct partial protein function, as discussed in detail in Section 1.1.2. On average, it can be expected that each individual domain in a given multi-domain protein gives rise to at least one GO MF term annotated for the protein. The domain functions, and therefore the terms, can either be related or unrelated. An example for two domains with related functions is the combination of a transporter domain with an active site domain that harvests (transforms) the energy required for the transport (e.g., by hydrolysing ATP), in an active transmembrane transport protein. This corresponds to the distribution of a single overall function (active transport requiring ATP) across two domains. An example for two domains with unrelated functions is the combination of an active site domain with another active site domain with different function. This is the case, for example, in multi-functional enzymatic fusion proteins such as the human ‘Bifunctional ATP-dependent dihydroxyacetone kinase/FAD-AMP lyase (cyclizing)’ protein (UniProt Q4KLZ6). The latter exhibits both ‘glycerone kinase activity’ (GO:0004371) and ‘FAD-AMP lyase (cyclizing) activity’ (GO:0034012), with each function encoded by a distinct domain and the functions differing in the first digit of the corresponding EC numbers. Fusion proteins with three or four different functions are rare but do exist, most prominently in evolutionarily old pathways in eukaryotes. There, they are sometimes found to encode a range of consecutive steps that require a set of individual proteins earlier in



evolution. An example is the human ‘CAD protein’ (UniProt P27708), which encodes four enzymatic activities in the pyrimidine pathway.

The different GO MF terms associated with multi-domain proteins can be related or unrelated to each other based on the function(s) of the individual domains, as illustrated by the transporter and bifunctional fusion protein examples above, respectively. Given that two terms are annotated in either case, the following assumptions about the character of these annotations should hold. In the first case, one MF term describes the overall function of the protein, the active transport of a specific substrate using ATP, while the other refers to a single aspect of this function, the binding (and consumption) of ATP. The two annotations (and functions) are therefore related, in the same way as essential and non-essential annotations are related in single-domain proteins. In contrast, in the second example above, the two annotated GO MF terms describe two entirely different enzymatic functions, carried out independently (with no shared, overall ‘aim’) by two different domains. The two annotations are therefore unrelated.

As a challenge for any classification algorithm, more complicated cases that mix the two scenarios outlined above exist in the databases. For example, one domain of a two-domain protein can give rise to both an essential and a non-essential term while the other is described by a single, essential term only. In addition, subsets of domains in multi-domain proteins may be collectively responsible for a given function (for example, mediated by an interface region) whilst other domains in the same proteins function autonomously.

It is important to note that, even if a heuristic algorithm could be devised that correctly differentiates between all the above-described scenarios for the relationship of protein domains and whole-protein function (annotation), any such algorithm has to work on the background of many preceding steps. The most important ones are gene prediction, protein domain decomposition and

the annotation process itself. If errors are made in any of these steps, this can deteriorate the performance of the algorithm. For example, fragments of real genes and pseudo-genes can lead to wrong or missing results in domain identification, wrongly identified domain boundaries can lead to (apparent) outlier sequences when clustering domain superfamilies, and wrong or missing protein function annotations can directly ‘misguide’ annotation-based algorithms such as those discussed in the present chapter.

#### 5.1.4 Protein families with functionally conserved domains

To illustrate the difference between protein domain and whole-protein function, two families of multi-domain proteins can serve as examples. These are the P-type ATPase (P-ATPase) family of ion transmembrane transporters and the class I aminoacyl-tRNA synthetase (aaRS) family. The catalytic domains of the proteins in these families come from two evolutionarily ancient domain superfamilies: the HAD (Haloacid dehalogenase) and HUP (‘HIGH-signature proteins, UspA, and PP-ATPase’) superfamilies, as characterised in Koonin and Tatusov (1994) and Aravind, Anantharaman et al. (2002), respectively. Both the HAD superfamily (CATH 3.40.50.1000) and the HUP superfamily (CATH 3.40.50.620) belong to the Rossmannoid fold class (CATH 3.40.50). Further, for both superfamilies, the respective parent proteins are assumed to have diverged into distinct functional families prior to the emergence of (an assumed) Last Universal Common Ancestor (LUCA) organism (Aravind, Anantharaman et al. 2002; Burroughs, Allen et al. 2006). Consequently, these superfamilies have member domain sequences in all three domains of life, and most of the proteins containing these sequences are essential for cell survival.

##### 5.1.4.1 The P-loop type ATPase family

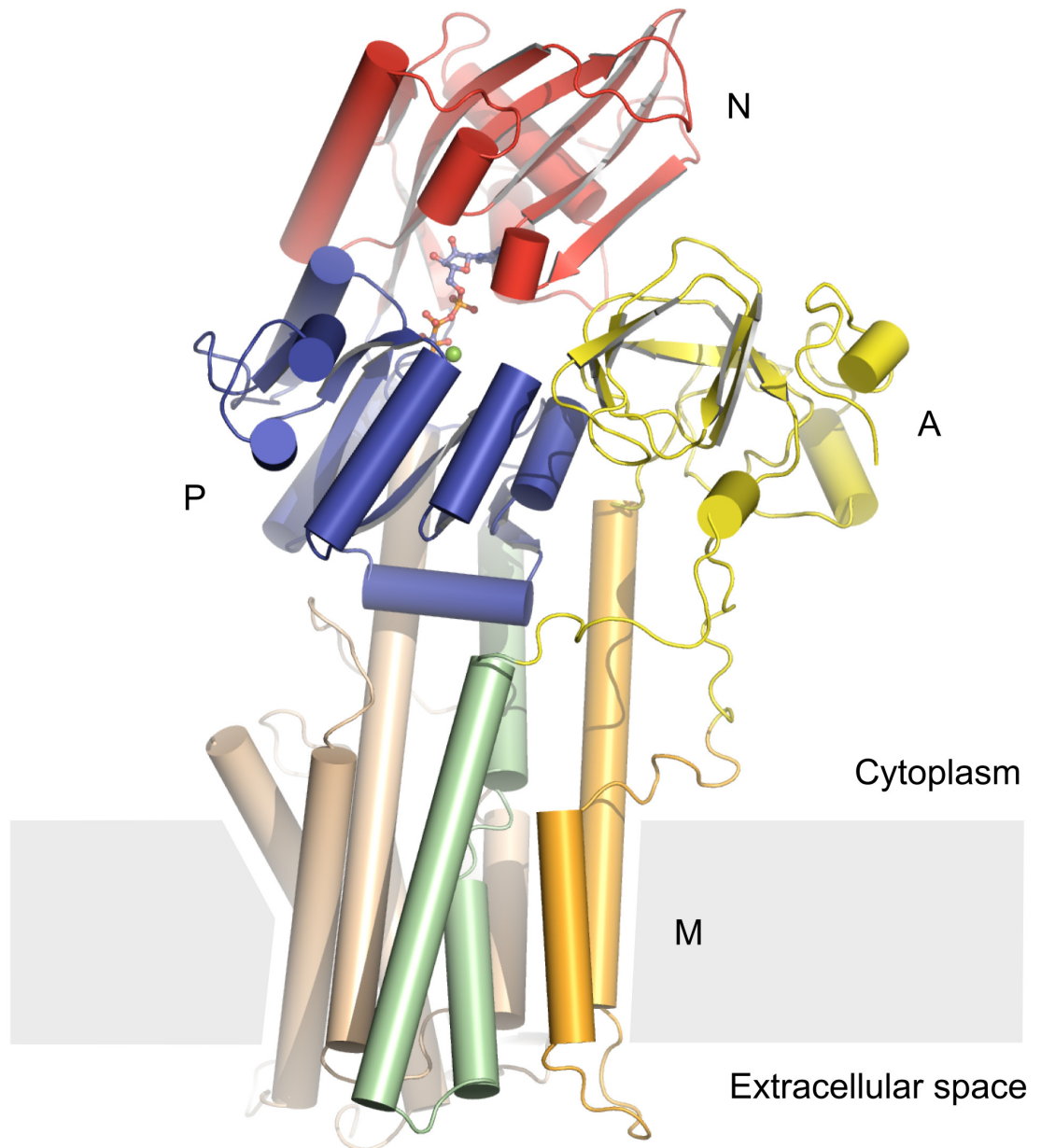
The P-loop type ATPases (P-ATPases) are an evolutionarily ancient and ubiquitous family of transmembrane transport proteins (Axelsen and

Palmgren 1998). These proteins actively transport different metal ion species across the cellular membrane and those of certain organelles; some are also known as ‘flippases’ that can transport phospholipids between the two membrane layers. The most prominent members of this family are the  $\text{Na}^+, \text{K}^+$ -ATPase found in the plasma membrane of animal cells, which was the first P-ATPase identified (Skou 1957), and the  $\text{Ca}^{2+}$ -ATPase found in the sarcoplasmic reticulum membrane of muscle cells, which was the first P-ATPase protein with a solved structure (Toyoshima, Nakasako et al. 2000). Ion transport through P-ATPases is mediated by conformational changes, induced by ATP hydrolysis and following reversible autophosphorylation of the protein.

Due to their early evolutionary divergence, the P-ATPase proteins usually exhibit low overall pairwise sequence identity, down to  $\sim 20\%$  (Geisler, Richter et al. 1993). Yet, they show relatively high structural conservation in their non-membrane parts (Palmgren and Nissen 2011). Both phylogeny and ion specificity are in some cases still unclear (Axelsen and Palmgren 1998; Thever and Saier 2009). According to the widely-used classification of Axelsen and Palmgren (Axelsen and Palmgren 1998), the P-ATPases fall into five classes and several subclasses, of which class Ib includes the transition and heavy metal ion transporters relevant to the examples below. This corresponds to the 3.A.3.5 and 3.A.3.6 families of the TC classification (Thever and Saier 2009).

All P-ATPase proteins share a common (core) domain architecture, as shown in Figure 5.2. A transmembrane (M) domain binds and transports specific ions, while a three-domain subunit that protrudes from the inside of the membrane drives this transport, through binding and hydrolysis of ATP and following conformational change. This subunit comprises the actuator (A), phosphorylation (P) and nucleotide-binding (N) domains. All four domains are clearly discernable as compact units in protein tertiary structure (see

Figure 5.2a), but only the A and N domains are also continuous in sequence. In detail, A is inserted N-terminally into M, while P is also inserted into M, close to its centre, and itself contains N as an insert. The P domain is found in the HAD domain superfamily. It was speculated that the common ancestor of all extant P-ATPases was the product of sequence fusion between a membrane transport protein (M domain) and a soluble ATPase enzyme (P domain) (Ogawa, Haga et al. 2000; Bramkamp, Gassel et al. 2003); the A and N domains would then have been acquired later on.



**Figure 5.2. The four structural domains of P-type ATPase transport proteins.** These proteins transform chemical energy stored in the form of ATP into mechanical energy for the active transmembrane transport of different metal ion species. The cytoplasmic actuator (A) and phosphorylation (P) domains are both inserts of the transmembrane domain (M). The cytoplasmic subunit is completed by the nucleotide-binding (N) domain, which is inserted into the P domain. A coupled ATP hydrolysis and protein autophosphorylation reaction takes place at the interface of the P and N domains. Following major conformational changes in the A domain lead to corresponding movements in several of the transmembrane helices of the M domain. This reaction cycle drives ion transport. The image was taken from Creative Commons and altered; it shows a cartoon representation of the structure of *Arabidopsis thaliana* proton ATPase AHA2 (PDB 3b8c).

The overall function of P-ATPases is the transport of ions across a cellular membrane, catalysed by the hydrolysis of ATP. While the three cytoplasmic domains are jointly responsible for hydrolysis, the transmembrane domain transports ions by going through a cycle of conformational changes (Palmgren and Nissen 2011). This corresponds to a transformation of chemical energy into mechanical energy. In brief, the N domain binds and positions an ATP molecule so that its  $\gamma$ -phosphate moiety points towards a conserved, reactive aspartate residue in the P domain. In a nucleophilic attack reaction, the phosphate is then transferred to the aspartate side chain to create an unstable aspartyl-phosphoanhydride intermediate. This corresponds to an autophosphorylation of the P-ATPase protein in the P domain. ADP is released and the N-domain reverts to its initial state. The A domain now undergoes conformational change and fills the ‘gap’ left by the N-domain. Mediated by a set of conserved residues in the A domain that bind and activate a water molecule for nucleophilic attack, by abstraction of a proton, the P domain is subsequently dephosphorylated. The release of  $P_i$  is stimulated by a newly bound ATP molecule (N domain), the A domain reverts to its initial state, and the cycle completes. The substantial conformational changes in the A domain during this catalytic cycle are translated, via linker regions, into movements of several transmembrane helices in the M domain. In turn, these movements facilitate the transmembrane transport of ions. It is important to note that the reactive aspartate residue in the P domain is conserved throughout the HAD superfamily.

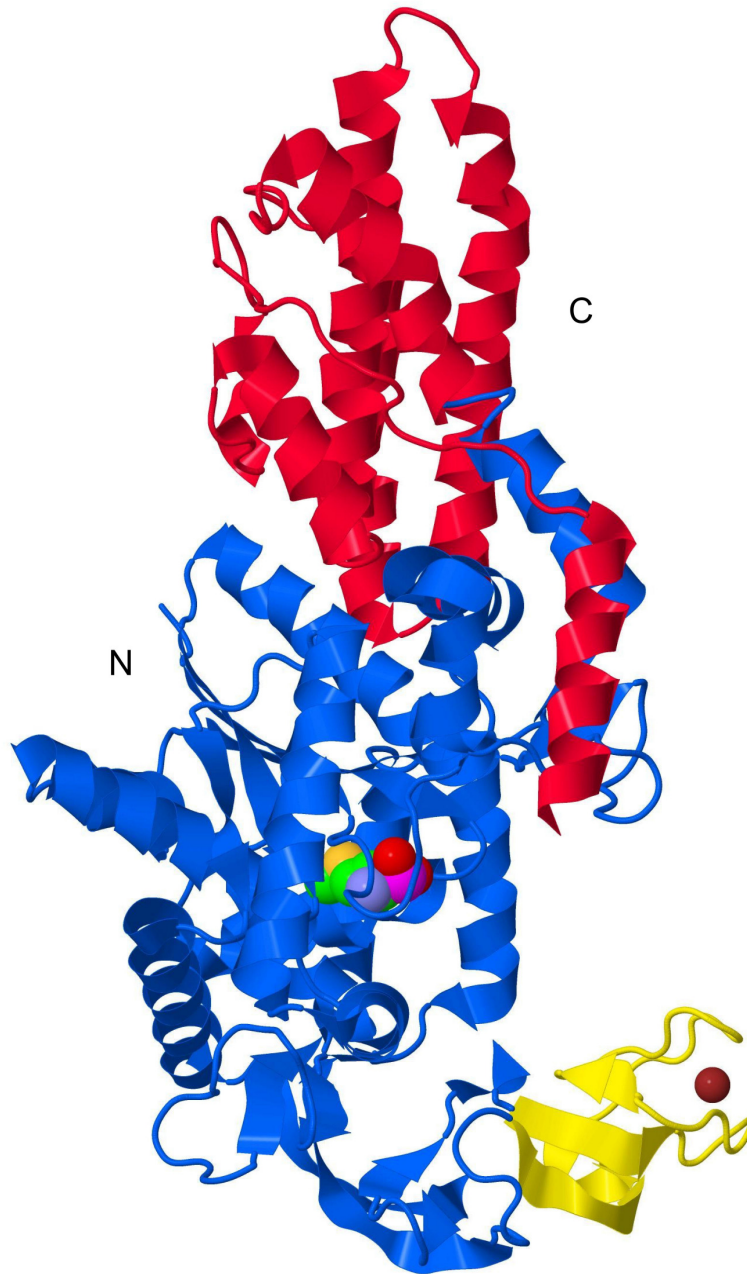
The four domains found in P-ATPase proteins show different degrees of conservation in sequence, structure and function. Substrate specificity, that is, which ion species can be transported, is largely determined by structural variation in the transmembrane domain (M) (Palmgren and Nissen 2011). This refers to relatively subtle differences in structure, based on sometimes extensive changes in sequence, to accommodate for the binding of specific ion species (Palmgren and Nissen 2011). Among the remaining domains (P, A

and N), only the P domain shows considerable conservation in sequence and structure throughout the P-ATPase protein family. Importantly, it is also the only of the three domains that belongs to a domain superfamily (HAD) whose members appear in different functional contexts, that is, in proteins other than P-ATPases.

#### 5.1.4.2 The class I aaRS family

Aminoacyl-tRNA synthetase (aaRS) proteins are responsible for charging the different transfer-RNAs (tRNAs) with their respective amino acids (Woese, Olsen et al. 2000) and thus fulfil a crucial role in one of the oldest cellular pathways: protein biosynthesis. They fall into two distinct classes, class I and class II (Eriani, Delarue et al. 1990). The HUP domain superfamily contains domains that are found in class I aaRSs, while the structurally unrelated class II aaRSs contain domains from other superfamilies and folds.

Class I aaRS proteins comprise two distinct domains, with the N-terminal HUP superfamily domain being the catalytic one (see Figure 5.3). This domain is responsible for a chain of reactions. In brief, these are the recognition and binding of both ATP and the respective amino acid, the splitting of the ATP molecule into AMP and inorganic pyrophosphate ( $PP_i$ ), with the latter being released from the complex, the subsequent formation of an activated aminoacyl-AMP (aminoacyl-adenylate) intermediate, and the final transfer (esterification) of the respective amino acid to its tRNA counterpart. The second, C-terminal domain of class I aaRSs is mainly responsible for recognising and binding the correct tRNA(s), through its highly specific anticodon region. Additional binding or editing domains are sometimes found but are not relevant in the context of this work (for example, the yellow zinc-binding domain in the *E. coli* MetRS structure in Figure 5.3).



**Figure 5.3. The two structural domains of class I aaRS proteins.** These proteins use chemical energy stored in the form of ATP to charge tRNAs with their cognate amino acids, for eleven of the 20 residue types. All steps that are necessary to complete the esterification of tRNA and amino acid are performed by the catalytic N-terminal domain (N). The C-terminal domain (C) is primarily responsible for anticodon recognition and, therefore, specific tRNA- binding. The image was created with Jmol and shows a ribbon diagram of the structure of *E. coli* Methionyl-tRNA synthetase (PDB 1pfy).

The class I aaRSs comprise those specific for arginine (ArgRS), cysteine (CysRS), glutamic acid (GluRS), glutamine (GlnRS), isoleucine (IleRS), leucine (LeuRS), methionine (MetRS), tyrosine (TyrRS), tryptophan (TrpRS) and valine (ValRS); it also contains lysine aaRS type 1 (LysRS), which has an unrelated counterpart in class II. Class I aaRSs can further be classified into



three different subclasses, as shown in Table 5.1. These were defined mainly based on protein structure comparisons (Cusack 1995).

The aaRS enzymes in each of the three class I subclasses tend to recognise chemically similar amino acid types. Members of class Ia recognise hydrophobic amino acids, such as the branched aliphatic (Ile, Leu and Val) and sulphur-containing (Met and Cys) types. Class Ib proteins recognise charged amino acids (Glu and Lys) and the uncharged polar Gln, a derivative of Glu. Class Ic enzymes recognise the aromatic amino acids Tyr and Trp (Ribas de Pouplana and Schimmel 2001). Note that especially the ValRS, LeuRS and IleRS proteins are known to be functionally closely related (and even overlapping) (Nureki, Vassylyev et al. 1998), which is also expressed by their shared proofreading mechanism (Nordin and Schimmel 2003). MetRS is also associated with this subgroup of class Ia aaRSs: ValRS, LeuRS, IleRS, and MetRS recognize A35 of tRNA with their tRNA-binding domains, whereas ArgRS and CysRS recognize C35 (Fukai, Nureki et al. 2003).

**Table 5.1. The three subclasses of class I aaRS proteins.** This classification was first provided in Cusack (1995) and is based on protein structure comparisons. Note that two subgroups can be distinguished in class Ia (vertical line; see main text).

Aminoacyl-tRNA synthetase class	Class member proteins
Ia	MetRS, ValRS, LeuRS, IleRS   CysRS, ArgRS
Ib	GluRS, GlnRS, LysRS
Ic	TyrRS, TrpRS

## 5.2 Concepts

The following sections introduce three important concepts that form the theoretical basis of the algorithms explained in the implementation section. While the concept of annotation term sets is a general one, used throughout

the DFX<sub>super</sub> workflow, the core set and chaining concepts are specific and characteristic of the idea behind the protocol.

### 5.2.1 Sequence and cluster annotation using sets

To capture the GO function annotations associated with individual sequences and sequence clusters, the ‘term set’ concept is consistently used in all stages of DFX<sub>super</sub>. Each of the sequences in a given domain sequence cluster is linked to a set of GO terms, the annotation of the respective parent protein. These sequence term sets contain only the most specific terms annotated for a protein, from all three GO branches. Further, for each unique sequence term set (sequence annotation) observed in the cluster, a single sequence is arbitrarily chosen as the representative sequence. This is to use all necessary, but no redundant, information in assessing the functional coherence of clusters (see Section 5.3.4). The cluster term set (cluster annotation) of a cluster is defined as the union of all terms that are found in any of its representative (this qualifier will henceforth be omitted) sequence term sets, with GO DAG parent terms removed. Consequently, the cluster term set contains only the most specific terms associated with sequences in the cluster. All sequence term sets and the cluster term set can be split into MF, BP and CC term sets, respectively.

DFX<sub>super</sub> further distinguishes between informative and problematic GO MF terms. The informative MF term set is the set of all terms defined in the GO MF DAG except terms in the problematic set. The problematic MF term set contains terms that are generally thought to convey less information about the overall molecular function of a protein than informative MF terms. Currently, it includes the ‘binding’ (GO:0005488) term and all its child terms. The binding term is currently annotated for proteins in a highly redundant way, in the sense that the assigned informative MF terms already imply the binding activity by definition; for example, the substrate binding of enzymes. Similarly

applies to many of its child terms, such as, for example, ‘protein-binding’ (GO:0005515) and ‘ATP-binding’ (GO:0005524). It could be argued that either the respective annotation guidelines (or habits) should change or, more profoundly, the binding term should be made the root of a separate, fourth branch of the Gene Ontology.

### 5.2.2 The core annotation of domain sequence clusters

The supervised family identification protocol derives a domain family partitioning from, first, the sequence clustering dendrogram of a given domain superfamily and, second, the (whole-protein) GO annotations that are associated with the clustered domain sequences via their parent proteins. GO molecular function (MF) annotations are the most relevant in this process, as established in Section 5.1.3.

Domain-specific annotation data would ideally be required to assess the functional coherency of protein domain sequence clusters. However, so far such data are not readily available (see also Section 5.1.2). Therefore, the supervised protocol includes a heuristic algorithm first to identify those MF terms that are most specific to the function(s) of the domain sequences in a cluster, out of all MF terms that are associated with the respective parent proteins. This is the cluster core MF term set (core set). The intended ideal composition of the core set is outlined below; the algorithm that has been implemented to compile it is described in Section 5.3.2.

Based on the domain family concept introduced in Section 0, and the definition of essential and non-essential annotations in Section 5.1.3, the core set of a given domain sequence cluster would ideally comprise all (and only those) MF terms that are essential to measure the degree of functional diversity among its member sequences. For domains from single-domain proteins, any non-essential terms that merely describe individual aspects of the overall protein function should be excluded from the core set. For

domains from multi-domain proteins, any terms that describe ‘foreign’ domain functions (those that are mediated by other domains in the respective protein) should additionally be excluded. This is true regardless of whether the function of a foreign domain is related or unrelated (see Section 5.1.3) to that of the domain under analysis, that is, whether the domains serve different partial functions (of a common overall function) or entirely independent functions.

The exclusion of non-essential MF terms from the core set is based on the observation that such terms are more frequently missing from protein annotations than essential MF terms. This can happen when (partial) functions are deemed not important or are simply ‘forgotten’ in the manual annotation process. The latter becomes even more likely when automatically assigned annotations are merely curated (manually checked), since the missing annotations may not be proposed by the automatic protocol used in the first place. In other words, the less important a term is to describe the overall function of a given protein, the more likely it is that the term is missing from the protein’s annotation. If non-essential MF terms were taken into account in measuring the functional coherence of protein (domain) clusters, some clusters could be erroneously judged functionally incoherent only due to such annotation incoherencies.

The exclusion of foreign domain MF terms from the core set directly follows from the domain family concept on which the DFX pipeline is based (Section 0). Further, annotations that refer to domains other than that under inspection can compromise the assessment of cluster functional coherence in a manner similar to inconsistently annotated non-essential MF terms (see above). An example would be a catalytic domain A that, in one exceptional case, is found together with a second catalytic domain B in a fusion protein. Clearly, the MF term describing the catalytic function of B should not be considered when

judging the functional coherence of a sequence cluster populated by domains of type A.

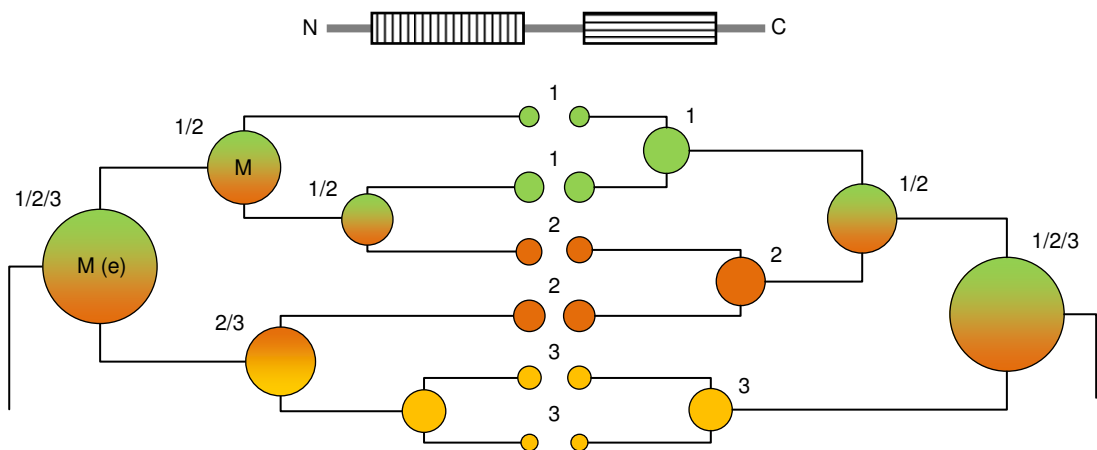
### 5.2.3 Chaining in the clustering of annotated sequences

The concept of cluster chaining can help to establish domain families with a focus on domain (not whole-protein) function, an aim that is discussed in detail in Section 3.2 above. It relies on both the clustering dendrogram, as obtained in the sequence clustering step of DFX (see Section 3.3.3.2), and the protein annotations associated with the domain clusters in this dendrogram. In brief, the concept is used to detect cases of incongruence between domain sequence conservation and protein function conservation. Such incongruence can, in turn, be a signal of domain function conservation, and can therefore be used to establish domain families of the above-mentioned type.

A cluster chain is a sequence of clusters in the dendrogram that are connected by child-parent relationships (edges; Figure 5.4). A cluster-function chain is a cluster chain in which each pair of sibling clusters shares at least one associated protein function (annotation). An end-of-chain cluster is the last parent cluster in a cluster-function chain, that is, the chain node that is closest to the root of the clustering dendrogram. A more exact definition of the concept of ‘chaining’ than given above is that of an observed deviation in the pattern of domain sequence clustering from the pattern of deviating function in the respective parent proteins; this definition will be used henceforth.

Cluster-function chains are expected and usually observed for cluster merges close to the leaf nodes of the sequence clustering dendrogram, that is, in the initial stages of agglomerative clustering. At this point, many functionally pure clusters exist and are expected to be merged with their closest relatives, that is, with clusters that are also functionally pure and represent the same function. However, in later stages of clustering (closer towards the dendrogram root node), sibling clusters are not generally expected to share any identical

functions. This is because most of the leaf clusters that represent the same single function should already have been merged, respectively, assuming that there exists a positive correlation between protein domain sequence and overall protein function similarity in the superfamily (which is expected for single-domain proteins). What *is* normally expected in later stages is the progressive merging of the ‘fully grown’ functionally pure clusters into larger, impure clusters, until only a single, maximally impure cluster (the root node) remains. In summary, the usual expectation is that all sequences with matching functions are first joined in a cluster before they join sequences with different functions.



**Figure 5.4. The concept of cluster chaining.** The top part shows the domain architecture of the sequences in a hypothetical two-domain protein family F. The N-terminal domain (N) fulfils the same partial protein function throughout the family, whereas the function of the C-terminal domain (C) varies. For both domains, a part of the GeMMA clustering dendrogram of the respective domain superfamily is shown. The colouring (numbering) of the clusters indicates the different annotations (functions) of the parent proteins. While the C superfamily clusters exactly according to the annotation pattern, the N superfamily does not. This becomes apparent in merges where at least one of the two sibling clusters is functionally impure and the cluster annotations overlap (here indicated by an M in the respective parent clusters). In a chain of such merges, the cluster closest to the dendrogram root node is the end-of-chain cluster (e).

Cluster-function chains become relevant for domain family identification whenever the above-outlined idealised merging order is violated in the clustering dendrogram of a given superfamily. In these cases, at least once in the chain an already functionally impure cluster is merged with another (either pure or impure) cluster that matches the former in at least one annotation. As

a result, the respective end-of-chain cluster is also functionally impure, and would hence not normally be judged functionally coherent (see Section 5.3.4). Importantly, when using end-of-chain clusters to derive domain families, this may result in different family partitionings for the individual superfamilies containing different domains from the same set of parent proteins. This is illustrated in the following example.

Figure 5.4 shows, at the top, a protein from a hypothetical two-domain protein family F. The N-terminal domain fulfils the same partial protein function throughout the family (e.g., ATP hydrolysis), whereas the function of the C-terminal domain varies (e.g., phosphorylation of various different substrates). The two domains come from different domain superfamilies, whose sequence clustering dendrograms are shown in part. The sequence clusters (nodes) in both dendrograms are coloured and numbered according to the union of the annotations of the respective parent proteins. For simplicity, it is assumed that only proteins from the F family have domains in these clusters. As can be seen, the domain sequences in the two superfamilies exhibit different clustering patterns, putatively due to the different functional constraints on (local) sequence conservation. While the clustering dendrogram of the C domain superfamily is in perfect agreement with the functional pattern (and the expected sequence clustering pattern) of the parent proteins, the N domain superfamily deviates from this pattern, that is, it exhibits chaining. The individual points of deviation are those merges where at least one of the two sibling clusters is functionally impure and the cluster function (annotation) sets overlap. As in any cluster-function chain (see above), the parent cluster closest to the dendrogram root node is the end-of-chain cluster ('e' in Figure 5.4).

The higher conservation of individual domains in sequence and function relative to their parent proteins is the only 'valid' (i.e., biological) reason for chaining. Otherwise, especially when using a highly sensitive profile-profile

sequence clustering method like GeMMA (see Chapter 2), it is highly unlikely that a domain that directly mediates the functional (e.g., substrate) specificity of its parent protein (i.e., that changes in function with its parent protein) would not cluster with its relative domains according to this specificity. This is because even changes in only a few functional key residues, within an otherwise highly conserved domain, should normally be sufficient to guide the (profile) clustering process. The same would be true, in fact, when clustering the parent proteins as a whole.

There also exist methodological, or artefactual, reasons for chaining. In the simplest case, one or more domain sequences in a cluster are erroneously annotated (via their parent proteins), that is, their annotation does not correspond to their true function (that of the parent proteins). This can lead to the (false) impression that sequences with different functions are joined in a cluster before joining other sequences with identical function, respectively. Such annotation errors are often a consequence of the biological reason for chaining mentioned above, the existence of highly conserved domains among the members of a protein family. Depending on the size of these domains, the respective proteins can exhibit substantial overall similarity in sequence and structure, especially when this is measured automatically (and not by eye). This makes a correct assignment of protein function difficult, both when using purely automatic function assignment pipelines (function prediction methods) as well as (if to a lesser extent) manual curation. In fact, in analogy to the chaining concept explained for domain sequence clustering above, this very uncertainty in protein function annotation often indicates the joint membership of proteins in a protein family. This is especially true if the family concept followed allows for a certain degree of function variation within families, as it is the case for the DFX domain family concept (see Section 0).



## 5.3 Implementation

If high-quality function annotation data are available for a superfamily, it is compiled in the data preparation step of the DFX pipeline (see Section 3.3.2). The pipeline then runs in supervised mode. This means, first, that all unannotated starting clusters are filtered out after pre-clustering (see Section 3.3.3.1), and second, that a supervised protocol is used to identify functional families after the main sequence clustering step. This protocol combines the clustering results with supervised clustering evaluation based on the annotation data, in the same way in which *ab-initio* methods like SCI-PHY (see Section 4.1.2.1) combine sequence clustering with unsupervised clustering evaluation (see Section 2.1.3.1).

For a given superfamily, the DFX supervised family identification protocol performs the following three steps. First, the initially compiled function annotation data are used to assess the functional coherence of all clusters (nodes) in the generated clustering dendrogram. Second, all nodes that are not sufficiently coherent are removed from the dendrogram. This splits the dendrogram into sub-trees, since the level of cluster functional coherence generally decreases between the leaf nodes and the root node. Third, only the root clusters of all derived sub-trees are retained, to form the set of identified functional families in the superfamily. The key step in the protocol (and the only non-trivial one) is the first, which also involves an extensive annotation editing procedure prior to the assessment of each cluster. The editing and assessment procedures are discussed in detail in the following.

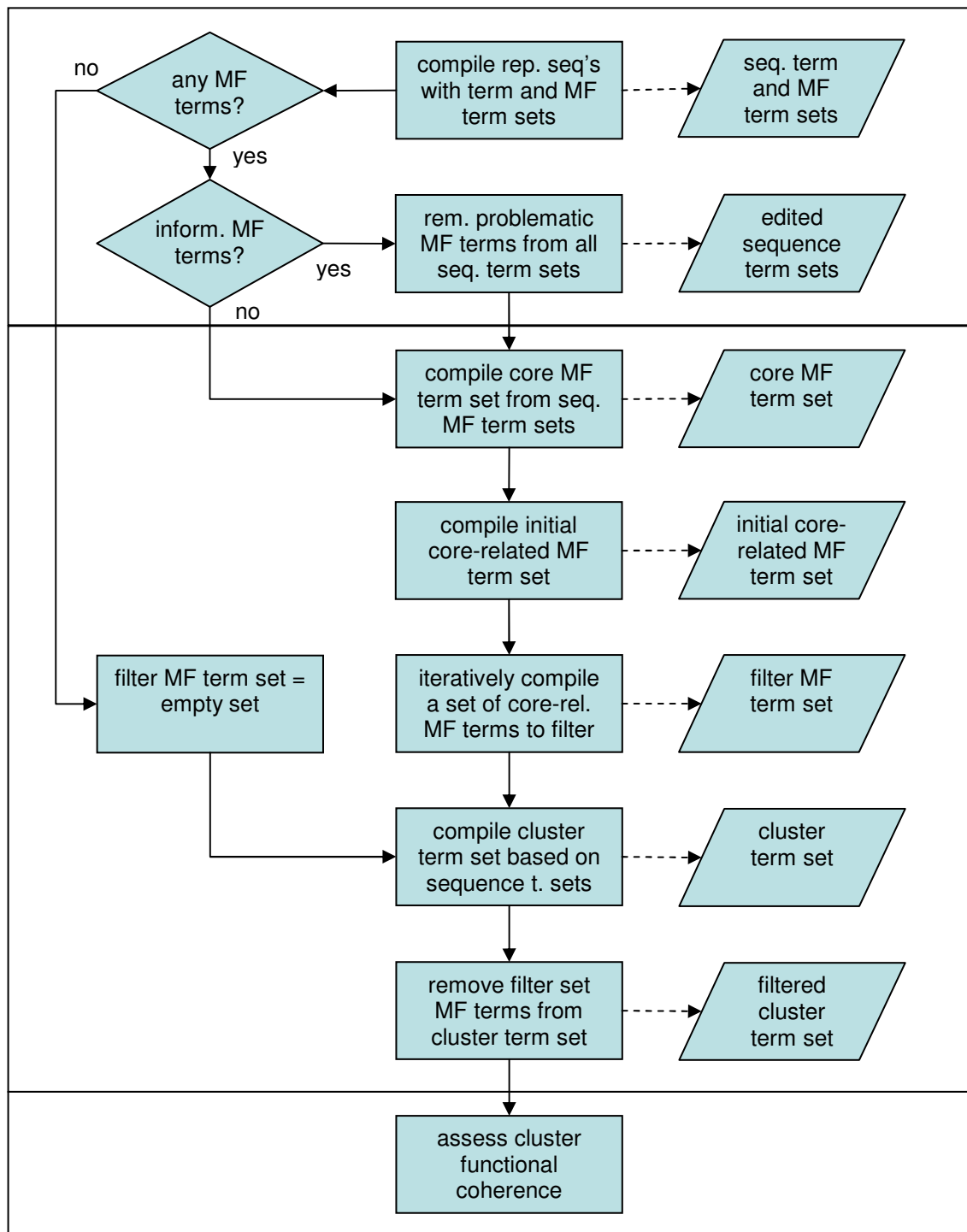
### 5.3.1 Overview of the protocol

Protein function annotations play a crucial role in the DFX supervised family identification protocol. Therefore, the functional coherence of each domain sequence cluster is assessed only after the annotations associated with the sequences in the cluster have been pooled, analysed and (potentially) edited.

This section provides an overview of the protocol as a whole; the remaining sections then focus on each individual step.

The diagram in Figure 5.5 provides an overview of the workflow followed for each individual domain sequence cluster. Initially, after identifying the set of cluster representative sequences and compiling the respective sequence term sets (sequence annotations; see Section 5.2.1), the latter are analysed to determine whether the cluster under analysis contains MF terms at all (see Figure 5.5, top). If this is the case, a second condition is tested: does the cluster contain at least one informative MF term? If so, any problematic MF terms are removed from both the (full) sequence term sets and the MF term sets. The distinction between informative and problematic MF terms is explained in Section 5.2.1. In brief, the reasoning here is to avoid the use of problematic MF terms if possible, since they can be detrimental (and are usually irrelevant) to correct cluster assessment.

In the next step, the cluster core term set (core set; see Section 5.2.2) is compiled (see Figure 5.5, middle). The core set is used to edit the full GO term set of the cluster (cluster annotation), prior to assessing its functional coherence based on the latter and the individual sequence term sets. This annotation editing can be important for a correct assessment, which concludes the process (see, Figure 5.5, bottom).



**Figure 5.5. Annotation editing in the supervised family identification protocol.** This shows the workflow followed for an individual domain sequence cluster. The right-most column shows the most relevant datasets that are generated in each step of the workflow, and how they relate. After pooling the non-redundant annotations of all sequences in the cluster, and compiling the core term set (top), the latter is used in an iterative process (middle) to compile the filter term set. This contains MF terms that are to be removed from the cluster annotation as a whole (cluster term set), before the latter is used, in conjunction with the individual sequence term sets, to assess the functional coherence of the cluster (bottom).

For the majority of protein domain clusters this (complete) workflow is followed, that is, the majority of clusters *do* contain sequences with MF terms. If this is not the case, however, several steps in the workflow are skipped (see Figure 5.5, left), the cluster term set is compiled and not edited, and either BP or CC terms (in this order of preference) are used to assess the functional coherence of the cluster. The characteristics of GO MF annotations that make pre-processing necessary, especially when dealing with domain sequences (see Section 5.1.3), do not apply to BP and CC annotations. In brief, this is because the cellular process(es) in which a protein takes part, and its corresponding location(s) in the cell, are the same for the protein as a whole and for each of its constituent domains.

### 5.3.2 Identification of the cluster core annotation

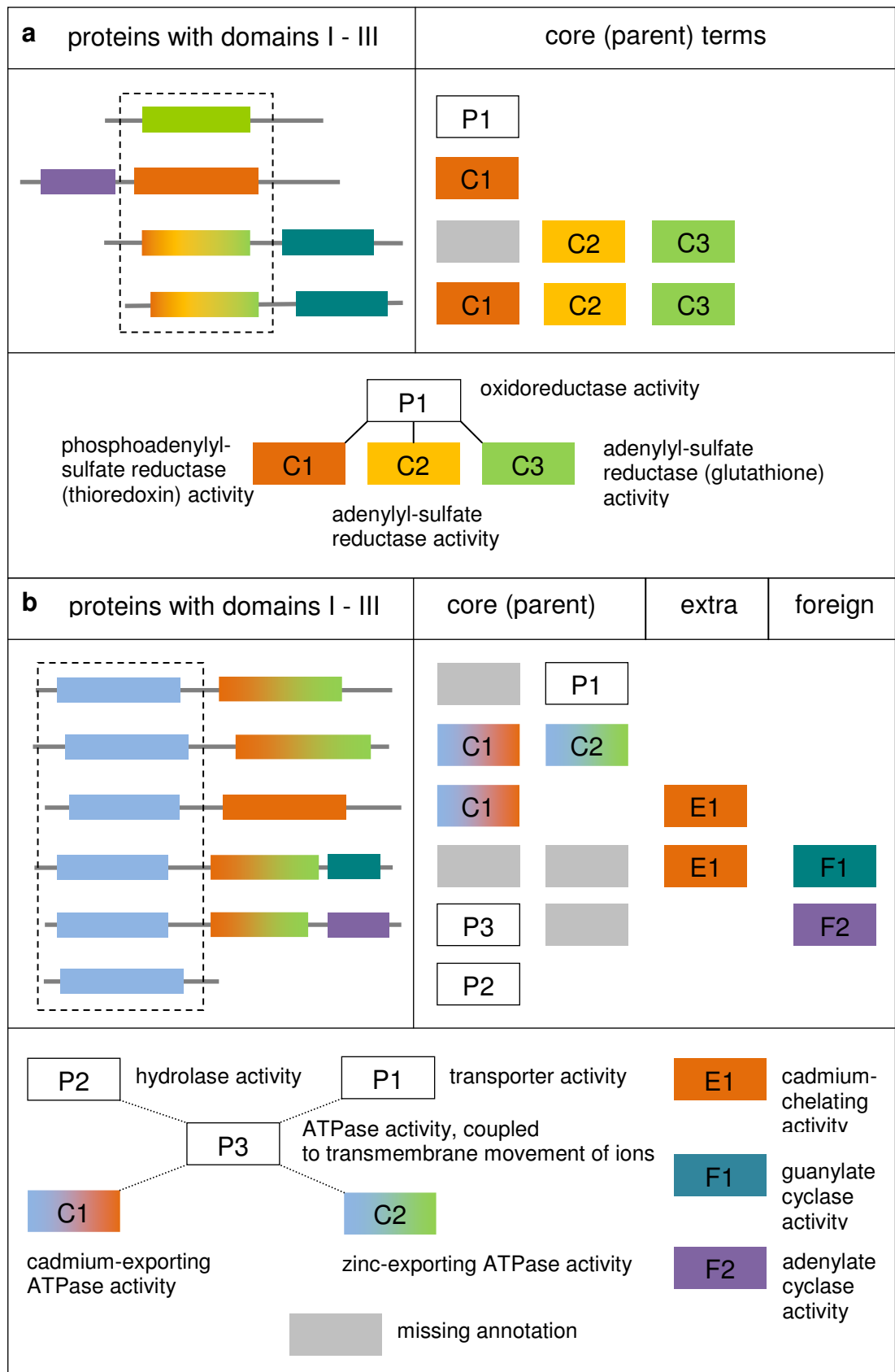
Section 5.2.2 outlines how the function(s) of the sequences in a given domain sequence cluster can theoretically be captured as a subset of the GO MF annotations associated with the corresponding parent proteins (the cluster MF term set), in a heuristic manner. In brief, this core set ideally only includes those MF terms that are relevant (essential; see Section 5.1.3) to describe the functions of the domain sequences in the cluster. However, it is not obvious *per se* from the MF annotations of the parent proteins which of them are essential and which are non-essential annotations. More importantly, it is not obvious in the first place which protein annotations should be considered in the context of the domain under analysis (the one in the cluster) and which refer to other (foreign) domains in these proteins. For these reasons, the supervised family identification protocol uses the following workflow to approximate the ideal core set composition.

The initial core set is compiled as the union of all terms found in those of the sequence MF term sets that have the minimum (but greater than zero) size observed. This strategy is an attempt to exclude both foreign domain and

non-essential annotations from the core set, an aim explained in the above section. In brief, it is based on the assumption that essential terms are less likely to be missing from the MF annotations of individual sequences than non-essential ones. Consequently, smaller sequence MF term sets tend to contain all (or a high proportion) of terms that are essential to describe the function(s) of the respective sequences, while larger MF term sets tend to contain additional, non-essential annotations.

Figure 5.6a shows a simple example annotation scenario, where a cluster contains four domain sequences with conserved reductase activity; these are the centred domains in the parent protein chains shown on the left, respectively. This domain is multi-functional in the sense that it can perform the same reaction on a range of highly similar (co-)substrates (Figure 5.6, bottom). For simplicity, the other domains in these proteins are assumed to have scaffold function only. The (partially incomplete) annotations of the parent proteins are shown on the right. In this example the initial core set is (C1, P1), based on the two sequences associated with a single MF term.

After compiling the initial core set as described above, this is processed further, in two steps. First, for any term occurring in those sequence MF term sets that have a size greater than the minimum size observed (those that were not considered when compiling the initial core set; see above), the presence of GO DAG parent terms in the initial core set is assessed. All terms for which parent core terms are found, and which are not already part of the initial core set, are added to this set. In the example in Figure 5.6a, these are C2 and C3, both children of P1. Second, any parent terms in the (now extended) initial core set are removed (P1). Taken together, these two steps ensure that the resulting core set contains the most specific out of all putatively essential annotations (functions) that occur in a cluster, and only those.



**Figure 5.6.** Two example domain sequence clusters and the associated protein function annotations. All domains are coloured and labelled according to their true functions; the high-quality GO annotations of the parent proteins are shown on the right, respectively. The terms are coloured according to the specific functions they describe, and their hierarchical relationships in the GO DAG are shown at the bottom, respectively (dashed lines represent omitted intermediate terms). Both clusters (dashed boxes) represent

conserved functional domain families according to the DFX family concept (see Section 0). The three reductase functions in (a) are closely related, as indicated by the three-functional cluster member sequences. In (b), the hydrolase function is perfectly conserved among all member domain sequences. Note that both the true domain functions and the different annotation types (core, extra and foreign; see main text) are ‘invisible’ to the core set identification protocol.

In addition to the simple example scenario in Figure 5.6a, which has been discussed above, Figure 5.6b shows a more complex situation. Here, a range of essential (core), non-essential (extra) and foreign domain annotations is associated with the domain sequences in the inspected cluster (domain I in the parent protein chains on the left, respectively), via their parent proteins. The core set is established according to the above-described steps. Therefore, the initial core set is (P1, P2), and the final core set, derived from the former, is (C1, C2).

The term P3 in Figure 5.6b exemplifies how specific GO terms can refer to the combined function of different domains. Note that it is impossible in this case to establish a core set that reflects the actual function of domain I (ATP hydrolysis), for two reasons. First, P2, despite its name ‘hydrolase activity’, which is a function of domain I only, is also a parent of P3. The child terms of P3, C1 and C2, therefore enter the core set. Second, even if that were not the case, these terms would still enter the core set via P1. This could only be avoided if the first protein was associated with more terms than just P1; for example, the missing ‘hydrolase activity’ for domain I. In this case, P1 would not be found in a sequence term set of minimum size (see above), unlike P2, and would therefore not play a role in identifying the core set.

### 5.3.3 Detection and removal of non-core annotations

Non-essential and foreign domain annotations should not be considered when assessing the functional coherence of domain sequence clusters, as established in Section 5.2.2. To this end, the cluster term set (cluster annotation), which plays a decisive role in the assessment procedure (Section 5.3.4), is edited prior

to assessment. This is only relevant for clusters that (i) contain sequences with MF terms and (ii) contain at least one sequence with an MF term set greater than the minimum sequence MF term set size observed, that is, one that was not used in establishing the initial core set (see Section 5.3.2). Such greater MF term sets tend to be greater because, apart from core terms, they also contain non-essential and/or foreign domain annotations (see Section 5.1.3). These are the types of annotations that the editing procedure (see Figure 5.5, middle) is supposed to remove from the cluster annotation. The detailed workflow is as follows.

Initially, the core parent set is generated. This contains the union of all GO DAG parent terms of the terms in the core set. The core and core parent sets together form the initial set of ‘core-related’ terms. A further, empty set is created at this point: the filter set. This is to hold all (putatively) non-essential and foreign domain MF terms that are detected in the iterative procedure described in the following; all terms compiled in the filter set are later removed from the cluster annotation, prior to cluster assessment. The process works as described in the following (for the first iteration).

All sequence MF term sets with greater than minimum size (see above) are analysed in the following way. First of all, all terms in the set are checked for whether or not they are core-related; a term is core-related if it is found in the core-related term set. If at least one core-related term is found, all terms in the set that are not yet registered as core-related then get added to a set of novel core-related terms. After assessing all sequence MF term sets, it is checked whether any novel core-related terms have been identified. If this is the case, these terms are added to the core-related term set, together with the union of all their GO DAG child and parent terms. Importantly, all these terms are also added to the filter set. Subsequently, all sequence MF term sets are assessed afresh. The iterative term set assessment procedure continues until no further core-related terms are identified. As a result, the filter set



contains all core-related terms that could be identified (including transitive identification), except the core terms themselves.

The core-related terms that are added to the filter set in the iterative process described above are expected to represent either non-essential or foreign domain annotations, with respect to the function(s) of the domain sequences in the processed cluster. If a sequence MF term set does not contain any core-related terms at all in the described process, none of its terms are added to the filter set. This situation can arise, for example, when a cluster mixes domain sequences from proteins with a single annotated MF term and such from proteins with multiple MF terms, and the (overall) functions of the single- and multiple-term proteins are not related.

The example scenario in Figure 5.6b, which has already been introduced in Section 5.3.2, illustrates how the filter set is progressively populated with non-essential and foreign domain terms. The core set is (C1, C2). In the first iteration of assessing the sequence MF term sets, E1, a non-essential (extra) term, is identified as core-related, based on its co-annotation with C1, a core term. At the same time, F2, a foreign domain term, is identified as core-related too, based on its co-annotation with the core parent term P3. F1, however, is only identified as core-related in the second iteration, based on its co-annotation with E1, a core-related term. In the simpler scenario in Figure 5.6a, no extra or foreign domain terms are annotated. Therefore, no core-related terms are identified in the single iteration that is carried out.

#### 5.3.4 Assessment of cluster functional coherence

The DFX supervised family identification protocol assesses the functional coherence of sequence clusters based on the GO annotations associated with their individual member sequences. In particular, it uses the following central rule. A cluster is deemed functionally coherent if it contains at least one sequence that, based on the associated annotations, covers all the functions

ascribed to any of the sequences in the cluster. As opposed to an assumed simpler protocol, requiring exactly matching annotations for all cluster sequences, this strategy is much more tolerant towards missing annotations. In particular, in combination with the annotation pre-processing step described in the previous section, it prevents domain sequences with matching functions but inconsistent annotations from being separated into different families. The assessment procedure is discussed in detail in the following.

The functional coherence of individual sequence clusters is assessed considering only a single type of GO term at a time. The order of term type preference is: informative MF, problematic MF, BP and CC. This corresponds to the importance of each term type when trying to identify functionally coherent sequence families based on GO annotations (see Section 5.1.3). Only if the term set of a given cluster does not contain terms of a specific type at all, the next type in the above list is used. Informative MF annotations are preferred over problematic MF annotations, and are solely used for assessment if at least a single informative MF term is found in the cluster term set. In turn, if only problematic MF annotations are available, these are still preferred over BP and CC annotations. This is because only MF annotations directly describe the function(s) (in a narrow sense) of individual proteins and domains (see Section 5.1.3). Note also that the MF annotations in the individual sequence term sets and in the cluster term set have at this point already been pre-processed in the manner described in Section 5.3.1.

After determining which GO term type is used to assess the functional coherence of the cluster, the assessment term type  $T_a$ , the protocol proceeds with compiling the necessary data. First, all  $T_a$ -type terms are collected from the cluster term set, forming the cluster type term set. Second, all sequences with at least one  $T_a$ -type term are compiled, and the corresponding sequence type term sets determined. The following step marks the core of the assessment protocol. All non-redundant sequence type term sets are

compared with the cluster type term set. In particular, it is tested how many (if any) terms of the given term type are part of the cluster annotation whilst not being part of the sequence annotation.

There are two possible outcomes of the comparison between the sequence annotations and the cluster annotation as described above. First, if at least one of the sequences covers all the annotations (functions) in the cluster annotation, the cluster is judged functionally coherent. Second, if that is not the case, one final test is performed, given that the cluster is assessed based on the MF term type. The test rule states that any end-of-chain cluster (see Section 5.2.3) is judged functionally coherent. This rule is based on the inherently increased probability for such clusters to represent functionally coherent sequence families, even in cases where this is not indicated by their (potentially diverse) annotation. The detection of cluster chaining (see Section 5.3.5) can be optionally disabled. Therefore, while it takes place prior to the assessment of all clusters in the  $DFX_{\text{super}}$  workflow, it is discussed last, in the below section.

### 5.3.5 Detection of cluster chaining

To identify cases of cluster chaining (see Section 5.2.3), chains of annotation-similar ('sticky') sibling clusters in the clustering dendrogram (cluster-function chains; see Figure 5.4) are established first. Stickiness is defined as a partial or full match of the GO term sets of two clusters. Whenever two sticky clusters are merged in the clustering dendrogram, this leads to either the start of a new chain or to the elongation of an already existing (growing) chain. Specific rules apply for different 'degrees of stickiness', as described in detail below. In each chain elongation step, the new (parent) node is connected to the child cluster that subsumes the other child cluster's term set. If the term sets of both child clusters match perfectly, the cluster that is itself the head of the longer chain (a chain length of zero means there is no chain) is connected to

the parent. A growing chain is terminated once a merge of two non-sticky clusters occurs.

Different cluster annotation properties have to be distinguished when establishing cluster chains. In particular, this refers to both the availability and the specificity of GO MF annotations for the clusters that are merged in each chain elongation step, respectively. The exact rules followed by the chain extension algorithm, for each merge in a growing chain, are shown in the workflow diagram in Figure 5.7. The first condition tested is whether at least one of the term sets of the two sibling clusters (sets T1 and T2) includes MF terms. Depending on whether or not this is the case, the left or right main branch of the workflow is followed. Importantly, if the term set sizes differ, it is made sure that T2 is the larger of the two term sets compared before branching.

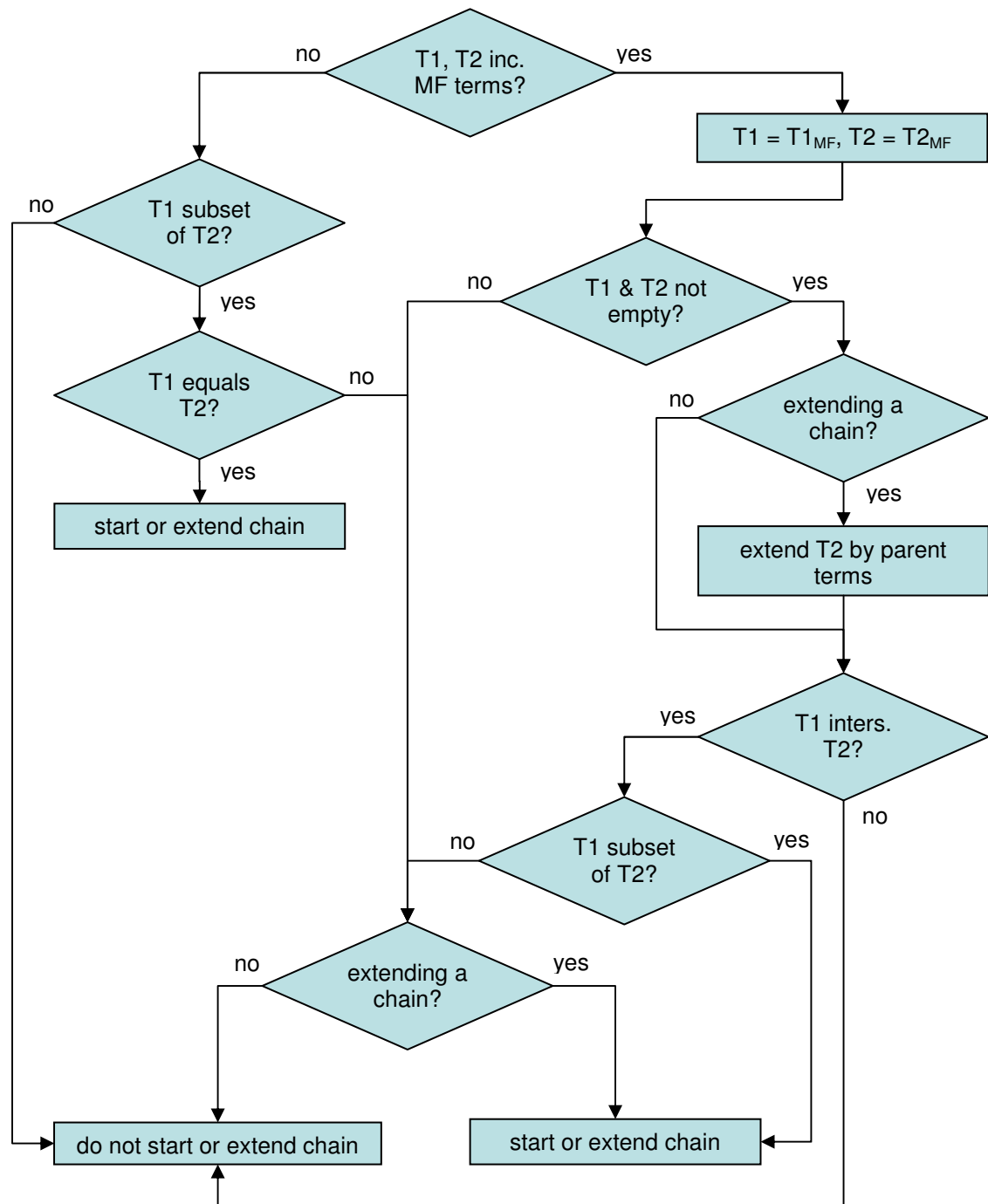
In the right main branch (MF branch) of the chain extension workflow shown in Figure 5.7, T1 and T2 are first replaced by their MF term subsets, respectively. Again, it is made sure that T2 is of greater or equal size when compared with T1. It is then tested whether or not both sets (still) contain terms. If this is not the case, that is, if one of the sibling clusters is not annotated with MF terms at all, several tests can be skipped; the point where the workflow continues in this case is pointed out in the text below. If both T1 and T2 contain MF terms, however, the workflow progresses with the next test.

At this point, the (possibly larger) T2 MF term set is extended by the union of all GO DAG parent terms of its member terms, given that the current step is a chain extension and not a chain nucleation step. This exception allows, as will become clear below, for chain extension in cases where the sequences in one of two sibling clusters are assigned only to coarser (parent) terms than (of) those in the other cluster. It was added as a heuristic to account for lacking

annotation specificity (depth). This heuristic is, of course, problematic in cases where the latter reflects a lack of knowledge of the respective sequence's function. When being cautious and excluding chain nucleation events from this exception, the rule appears to be beneficial to the family identification process.

First, if T1 is a perfect subset of T2, the chain is extended (or started). In this case, one of the sibling clusters covers all the functions of the other cluster (with more specific annotations, at least partly, if the above-described exception was made). If T1 is not a perfect subset of T2, starting a chain is ruled out at this point. The same is true for cases where one of the sibling clusters is not associated with MF terms at all, in which case several of the just described steps were skipped (see above) and the workflow continues at this point. In both cases the reasoning behind not starting (but possibly extending) the chain is that chain nucleation should require higher confidence in the functional equivalence (or high similarity; see Section 0) of the domain sequences in the merged clusters than chain elongation. This is particularly important when a low minimum chain length setting is used in end-of-chain cluster detection (see below), as it currently is the case.

The non-MF branch of the chain extension workflow in Figure 5.7 (left) is followed when both sibling clusters under analysis are not associated with any GO MF terms. In this case, the assessment of cluster functional similarity (stickiness; see above) is only possible in a crude, heuristic manner, based on GO BP and/or CC terms. Therefore, a chain is only extended or started without further tests if the term sets T1 and T2 match perfectly. If they do not match, and if T1 is not (at least) a perfect subset of T2 (which never contains fewer terms than T1 at this point; see above), the chain is not extended (or started), and the workflow terminates. If T1 *is* a perfect subset of T2, however, a chain can still be started. The workflow therefore continues and joins the right main branch (see



**Figure 5.7. The rules followed by the chain elongation algorithm.** In each chain elongation step the right or left main branch of this workflow is followed, depending on whether or not at least one of the sibling clusters being merged is associated with GO MF terms. All steps of the workflow are discussed in detail in the main text.

Figure 5.7). Chain extension (not starting) is granted here, provided that the above-described test for jumps in cluster size is passed. Otherwise, the chain is terminated. The key step in the (MF branch of the) chain extension algorithm follows. This is the test for whether or not the term sets  $T1$  and  $T2$  intersect.

If not, the chain is terminated (or not started) at this point. Otherwise, the tests continue as follows.

By detecting all nodes in the clustering dendrogram that qualify as chain elements, following the rules outlined above, chains of varying length can be established. Only the top nodes of each chain become end-of-chain clusters, hence the name. Since all end-of-chain clusters are judged functionally coherent in the assessment stage of the supervised protocol (see Section 5.3.4), any child clusters they subsume (in the chain) need themselves not be marked as such. Further, a minimum chain length is set using the parameter  $L_{\min}$ . This is the minimum number of consecutive merges of sticky clusters (merges that pass the above-described tests) that is required to constitute a chain.

As further discussed in Section 5.5.4, the setting of  $L_{\min}$  has a considerable influence on the family partitioning eventually derived, and can be used to adjust the degree of functional coherence of the produced families; it is currently set to a value of two. This setting is based on the assumption that a single merge of two sticky clusters may often represent an insignificant ‘outlier case’. Such can arise, for example, through errors in protein annotation or domain assignment (see Section 3.4.1). A single merge is therefore not seen as a strong enough indicator to assume a close functional relationship between the sequences in the merged clusters, whereas consecutive merges are. This is partly based on the manual inspection of family partitionings derived with different minimum chain length ( $L_{\min}$ ) settings.

As a by-product of tracing the clustering dendrogram for end-of-chain nodes, all sibling clusters with perfectly matching GO term sets are marked to be ignored in the family identification process. This is because they would always yield the same result in the assessment of functional coherence as their parent

cluster. In the case of a positive result, the parent would always supersede them in the final step of family identification (removing all child nodes of family clusters in the dendrogram). In the case of a negative result, neither the parent nor any of the two sibling clusters would be made a family. Such considerations, based on the hierarchical structure of the clustering dendrogram, are important to speed up the protocol.

## 5.4 Results and Discussion

The characteristics of the domain families produced by  $DFX_{\text{super}}$  are discussed by example in the following. In particular, the families that were identified for two types of catalytic domains, one with conserved and one with variable domain function, are analysed in detail. In this, specific examples serve to underline the importance of the key concepts used in  $DFX_{\text{super}}$ . Subsequently, the sequence footprint that domain function conservation can leave is discussed in a detailed example. The last section illustrates the challenges posed to the family identification protocol by frequently complex and sometimes inconsistent patterns of GO function annotations. Importantly, a quantitative assessment of the families produced in the first large-scale run of the DFX pipeline is found in Chapter 6, in conjunction with a comparison of the two DFX family identification protocols.

### 5.4.1 Domain function captured in selected domain families

The families that were identified by  $DFX_{\text{super}}$  in two evolutionarily ancient domain superfamilies are discussed in the following two sections. The focus lies on the catalytic domains of the two multi-domain protein families introduced in Section 5.1.4. More specifically, the use of the chaining concept (see Section 5.2.3) to identify ‘true’ domain families is demonstrated, that is, such that adhere to the domain family concept introduced in Section 0. The latter is based on the observation that some types of domains can act as (relatively) independent functional modules, responsible for the functions in



different whole-protein contexts, that is, in different domain architectures. The goal of the chaining concept is to group such sequences into families that represent their shared domain function, independent of whole-protein function similarity and overall homology relationships.

#### 5.4.1.1 P-ATPase catalytic domains in the HAD superfamily

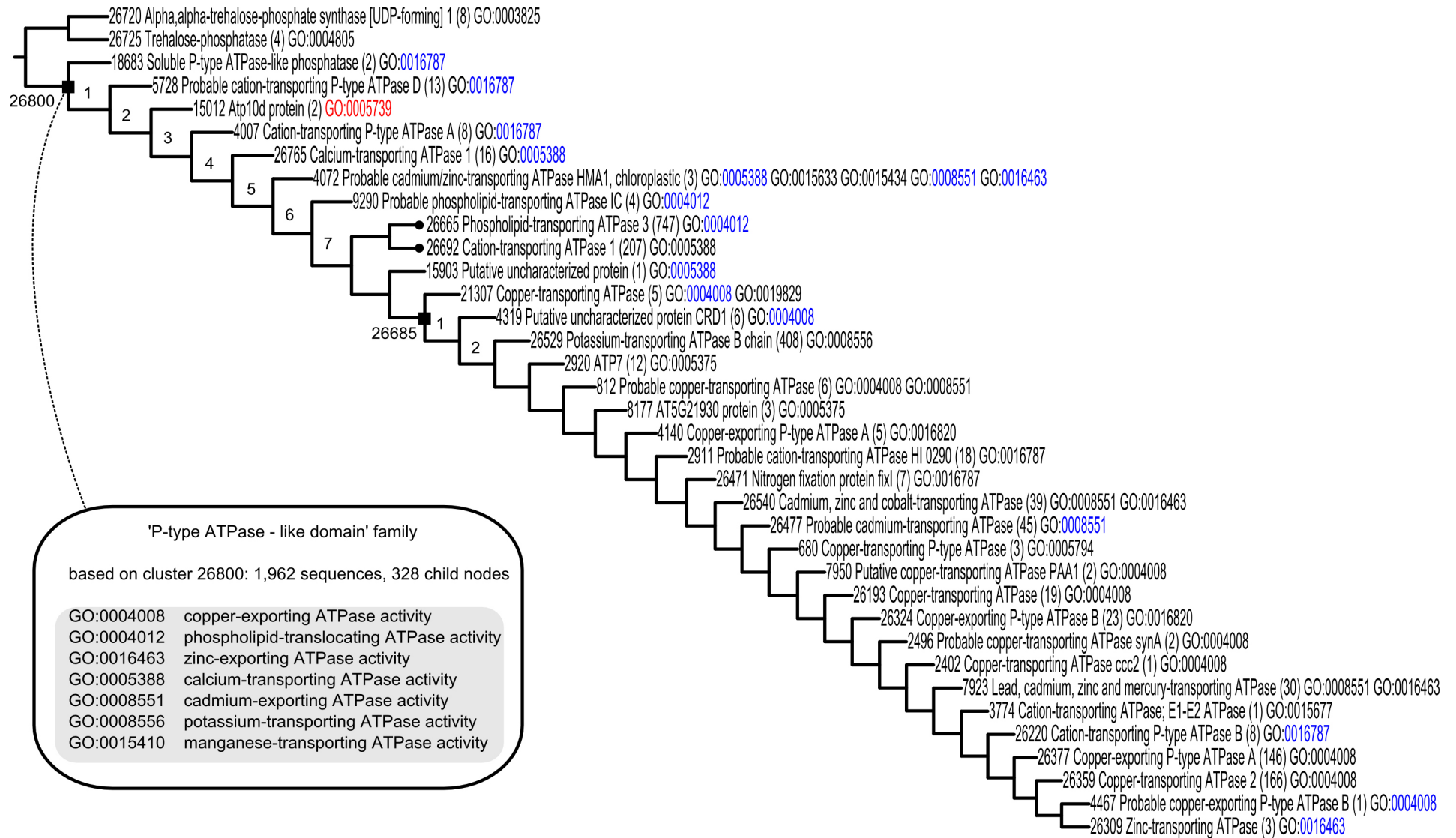
The ‘P-type ATPase -like domain’ family is not only the largest family identified in the HAD superfamily using the supervised family identification protocol but also the only family that represents the catalytic P domain of P-ATPase proteins. As explained in Section 5.1.4.1, this domain fulfils the exact same partial protein function in all P-ATPases. Putting all these domains into a single domain family seems therefore justified, despite the different annotations (functions) of their parent proteins (GO terms, EC numbers and protein names). Notably, this family is also a prime example of the DFX family naming protocol (see Section 3.3.5) working well. Out of all protein names associated with the respective parent proteins, it identifies the most suitable one for naming the family; the P-ATPase function is shared by all these proteins, while their ion specificity and corresponding naming varies.

The identification of the P-ATPase P domain family in the HAD superfamily, as described above, is only possible when the underlying end-of-chain cluster (cluster 26800) is recognised as such. Figure 5.8 shows the relevant part of the HAD superfamily clustering dendrogram. The cluster has a cluster-function chain (see Section 5.2.3) of length seven, as indicated by the numbers next to the internal nodes. The GO MF terms that are responsible for chain elongation, respectively, are highlighted in blue. All other blue-marked GO terms indicate the first appearance of a given function (annotation) in the clustering process; the red-marked term is discussed below. Note that some of the larger leaf clusters subsume a range of prior, ‘unproblematic’ cluster

merges, that is, merges of clusters with matching annotation; these are not shown.

Only six of the seven leaf clusters in the cluster-function chain of cluster **26800** (see Figure 5.8) show one or more blue-marked (chain elongation) GO terms. In the case of cluster **15012**, the only high-quality GO annotation available (for one of the two parent protein sequences) is the cellular component term ‘mitochondrion’ (GO:0005739), as highlighted in red. However, since the respective cluster merge is a chain-elongating merge, not a chain-nucleating one, the lack of MF terms in the merged-in node does not lead to chain termination (see Section 5.3.5). In other words, while the substrate specificity of the ATP10D parent proteins of the domains in this cluster (e.g., UniProt **Q6PEW3**) is yet unknown and they lack any high-confidence MF annotations, the DFX<sub>super</sub> protocol rightly ‘assumes’ that these domains still belong to the same family. This is achieved through the detection of the chaining context and the corresponding end-of-chain cluster, that is, by taking into account the surrounding sequence and annotation space.

**Figure 5.8. The clustering dendrogram of the P-ATPase P domain family.** This family of domains with conserved function was identified by  $DFX_{\text{super}}$  in the HAD superfamily. The figure shows the cluster merging events that underlie the identified end-of-chain cluster 26800, which gives rise to the family (box). The names, sizes and core term sets of all clusters are shown; the ‘-like domain’ suffix is omitted in the cluster names. End-of-chain clusters are indicated by black square nodes; the nodes of the corresponding cluster-function chains are numbered. The GO terms responsible for chain elongation are highlighted in blue, respectively; blue-marked terms outside chains indicate the first appearance of an annotation; non-MF terms are highlighted in red. Two other clusters that are mentioned in the main text are indicated by black circle nodes. Note that some leaf clusters subsume a range of prior merges that are not shown. The dendrogram was generated using iTOL; relative branch lengths were derived from  $-\log(E)$  values that indicate the similarity of two clusters, as reported by COMPASS.



When deactivating the detection of cluster chaining for *only* cluster 26800, that is, masking its end-of-chain status, it first breaks apart into eleven smaller clusters. The three largest of those are, from top to bottom in Figure 5.8, clusters 26665, 26692 (small black squares) and 26685. Note that the latter cluster (‘Phospholipid-transporting ATPase 3’<sup>16</sup>), the largest of the three, is itself an end-of-chain cluster; this is indicated by a black square. Its cluster-function chain originates two merges prior to its creation, hence the cluster has a chain length of two (the current minimum chain length, see Section 5.3.5).

When deactivating the detection of cluster chaining entirely, the large end-of-chain cluster 26800 in Figure 5.8 (the family representing the P-ATPase P domain; see above) breaks up into a total of 34 clusters, that is, into all the clusters in the shown dendrogram. In brief, this is because none of the internal nodes qualifies as coherent when applying the normal protocol for assessing cluster functional coherence (see Section 5.3.4). This can be understood from looking at the functions (core term sets; see Section 5.3.2) that are associated with each of the leaf clusters in Figure 5.8. Some of these clusters contain sequences from multi-functional P-ATPase, for example, cluster 4072 (‘Probable cadmium/zinc-transporting ATPase HMA1, chloroplastic’) and cluster 26540 (‘Cadmium, zinc and cobalt-transporting ATPase’). However, at no point in the sequence of cluster merges depicted in this dendrogram is an individual domain associated with *all* the functions of cluster 26800. Therefore, this cluster is only identified as a domain family when using the end-of-chain cluster exception.

Interestingly, the last leaf cluster that is added to the growing P-ATPase P domain family cluster in the dendrogram of the HAD superfamily (see Figure 5.8) contains domains from two archaeal proteins that each have only a single domain, the HAD domain. It has been noted before that these soluble, single-

---

<sup>16</sup> The ‘-like domain suffix is henceforth omitted when stating family names.

domain phosphatases are probably the closest extant relatives of the precursor of all P-ATPases (Ogawa, Haga et al. 2000; Burroughs, Allen et al. 2006). Importantly, while the proteins' domain architecture has changed dramatically here (from a single domain to at least four different domains; see Section 5.1.4.1), the function of the domain itself has either not changed at all or only subtly, from a phosphatase activity of yet unknown specificity (involving ATP-binding) to an ATPase activity (involving a phosphointermediate) (Bramkamp, Gassel et al. 2003). The domains in cluster **18683** in Figure 5.8 are therefore correctly identified by  $DFX_{\text{super}}$  as (remote) members of the P-ATPase P domain family. As the dendrogram further indicates, the P domain family seems to be most closely related to the phosphatase domain of Trehalose-phosphatases.

#### 5.4.1.2 Class I aaRS catalytic domains in the HUP superfamily

The HUP domain superfamily shows a complicated, 'patchy' GeMMA clustering pattern, and, following from this, a complex family dendrogram (see Figure 5.9). The families representing the catalytic domains of the different class I aaRS types are focussed on in the following. Eight of the eleven aaRSs are represented by more than one domain family. In two cases, these families represent more than a single aaRS type. The black circle nodes in Figure 5.9 highlight the largest identified (main) domain families for each aaRS type, respectively. The family sizes and core term sets of all families are shown. In addition, all families are taxonomically characterised by the last common ancestor taxon (or the domains of life) of the species harbouring their member sequences, respectively.

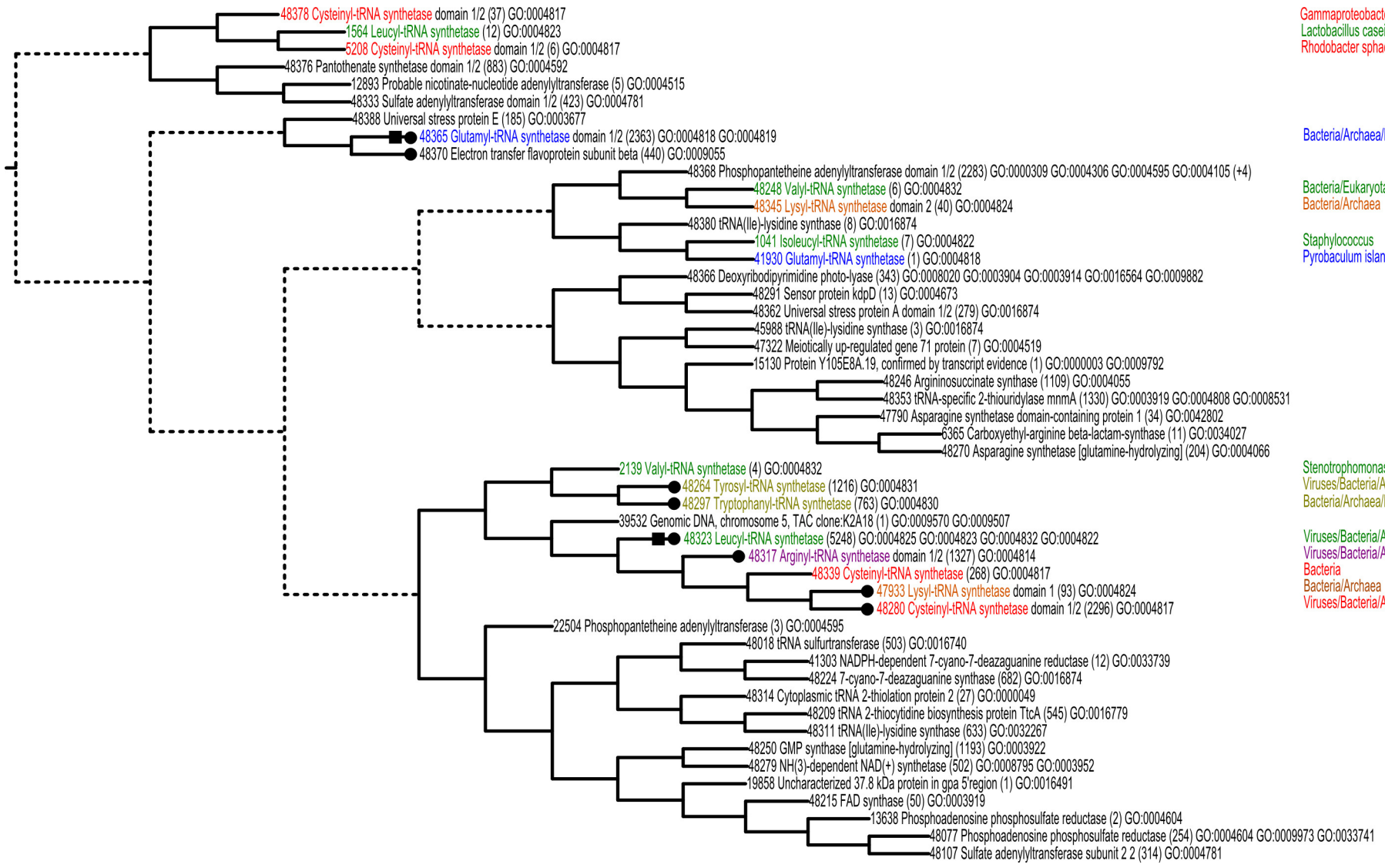
Outlier domain sequences with unusual composition, from certain taxa, may explain the complicated clustering pattern of the different aaRS types (see Figure 5.9). In particular, the well-known patterns of horizontal gene transfer among these ancient proteins and the occurrence of distinct organellar

isoforms in eukaryotes (Wolf, Aravind et al. 1999; Woese, Olsen et al. 2000) may contribute to the observed complexity. However, all this is not the focus of the following sections. Rather, the known and predicted relationships between the different aaRS (domain) types that can be discerned from the family dendrogram are discussed, and the impact of cluster chaining (detection) on the derived family partitioning is illustrated by examples.

In general, the pair- and group-wise proximities of the different types of aaRS domains in the family dendrogram in Figure 5.9 corresponds to the known relationships between the three subclasses of class I aaRS proteins (see Section 5.1.4.2). Families that represent domains with closely related aaRS functions are here shown in the same colour. These will be addressed together in the text below. In agreement with class Ia proteins being the most abundant group in the known class I sequence space (Ribas de Pouplana and Schimmel 2001), the largest aaRS domain family identified in the HUP superfamily is the ‘Leucyl-tRNA synthetase’ family (cluster **48323**). Despite the name assigned by the current family naming protocol (which splits the protein names by whitespace only, and not, e.g., hyphens; see Section 3.3.5), this family includes domains from four types of class Ia aaRSs: LeuRS, IleRS, ValRS and MetRS (indicated by green colouring in Figure 5.9). The underlying cluster contains high-quality annotated sequences with these functionalities in relatively balanced proportions (between 300 and 600 of each type). Three other families in the dendrogram contain sequences with the above-mentioned functions (apart from MetRS). These can be regarded as outlier cases, with only ~10 sequences per family on average. The co-occurrence of the four mentioned aaRS types in the large main cluster indicates the close relationships in this class Ia subgroup (see Section 5.1.4.2).

**Figure 5.9. Families of aaRS catalytic domains in the family dendrogram of the HUP superfamily.** This shows the families identified by DFX<sub>super</sub> and their proximity in superfamily sequence space. The coloured families designate different aaRS types, as mentioned in the main text. Black circle nodes highlight the largest family identified for each type, respectively; black square nodes indicate families that were identified on the basis of end-of-chain clusters and are therefore particularly addressed in the main text. The identifiers, names, sizes, core term sets and last common ancestor taxa of all families are shown; the ‘-like domain’ suffix is omitted in the family names. The dendrogram was generated using iTOL; relative branch lengths were derived from  $-\log(E)$  values that indicate pairwise cluster similarity, as reported by COMPASS. The dashed branches correspond to artificially introduced merges with an arbitrarily chosen (large) branch length. They indicate points at which at least one of two merged domain sequence clusters could not be aligned with MAFFT and the superfamily clustering process was therefore terminated prematurely.





Gamma proteobacteria  
 Lactobacillus casei  
 Rhodobacter sphaeroides

Bacteria/Archaea/Eukaryota

Bacteria/Eukaryota  
 Bacteria/Archaea

Staphylococcus  
 Pyrobaculum islandicum

Stenotrophomonas maltophilia  
 Viruses/Bacteria/Archaea/Eukaryota  
 Bacteria/Archaea/Eukaryota

Viruses/Bacteria/Archaea/Eukaryota  
 Viruses/Bacteria/Archaea/Eukaryota  
 Bacteria  
 Bacteria/Archaea  
 Viruses/Bacteria/Archaea/Eukaryota

The second-largest class Ia family is the ‘CysteinyI-tRNA synthetase domain 1/2’ family, closely neighbouring the above-described LeuRS family in the dendrogram in Figure 5.9, and shown in red. Compared with the latter, this family contains about half the number of sequences. Between this and another, smaller CysRS family (~200 sequences) lies a family that exclusively represents the first of two homologous LysRS domains (orange), whereas the family representing the second domain is found to cluster closely with one of the small ValRS outlier families. The occurrence of additional, fragmentary copies of the catalytic domain (or the artefactual assignment of such by Gene3D) is also observed in several of the other aaRS subclasses. Interestingly, this clustering pattern suggests a membership of LysRS in class Ia, instead of class Ib as predicted earlier (Ribas de Pouplana and Schimmel 2001). However, these earlier predictions may be more reliable as they were made on the basis of structural comparisons. The observed confinement of the aaRS class I LysRS domains to the archaeal and bacterial lineages confirms established knowledge (Ambrogelly, Korencic et al. 2002). Similar to the LeuRS-and-relatives family discussed above, relatively small outlier families with considerable distance from the main family in the dendrogram also exist for the CysRS family (red). The single family of ArginyI-tRNA synthetase domains (purple) that is found between the larger families discussed above completes the picture of the class Ia aaRSs.

All GlutamyI-tRNA synthetase domains are found in a single, large family (cluster 48365) in Figure 5.9 (blue), apart from a single GluRS domain sequence from the archaeon *Pyrobaculum islandicum* that was wrongly assigned (truncated) by Gene3D, in cluster 41930. The ‘GlutamyI-tRNA synthetase domain 1/2’ family also contains the closely related GlnRS domains (see Section 5.1.4.2). In summary, and if LysRS is assigned to class Ia as suggested above, the discussed Glu/GlnRS domain family exclusively represents the class Ib aaRS catalytic domain in the partitioning of the HUP superfamily produced by DFX<sub>super</sub>.

Finally, the two class Ic aaRSs, TyrRS and TrpRS, are both represented by a single identified domain family, respectively, as indicated in olive in Figure 5.9. The two families arise from two sibling clusters, clusters **48264** and **48297**, which underlines their close relatedness. Overall, the DFX<sub>super</sub> family dendrogram confirms the view of class Ic aaRSs being more closely related to the class Ia proteins than to those in class Ib (Nureki, Fukai et al. 2001). In summary, apart from several small outlier families, the domain sequences of the three class I aaRS subclasses are grouped into families in a biologically reasonable manner by DFX<sub>super</sub>. Importantly, this partitioning into different families directly reflects variation in *domain* function, that is, the function of the aaRS catalytic, N-terminal domain. It stands in contrast with the grouping of all P-ATPase P domains into the same family within the HAD superfamily (see Section 5.1.4.1) based on their conserved domain function.

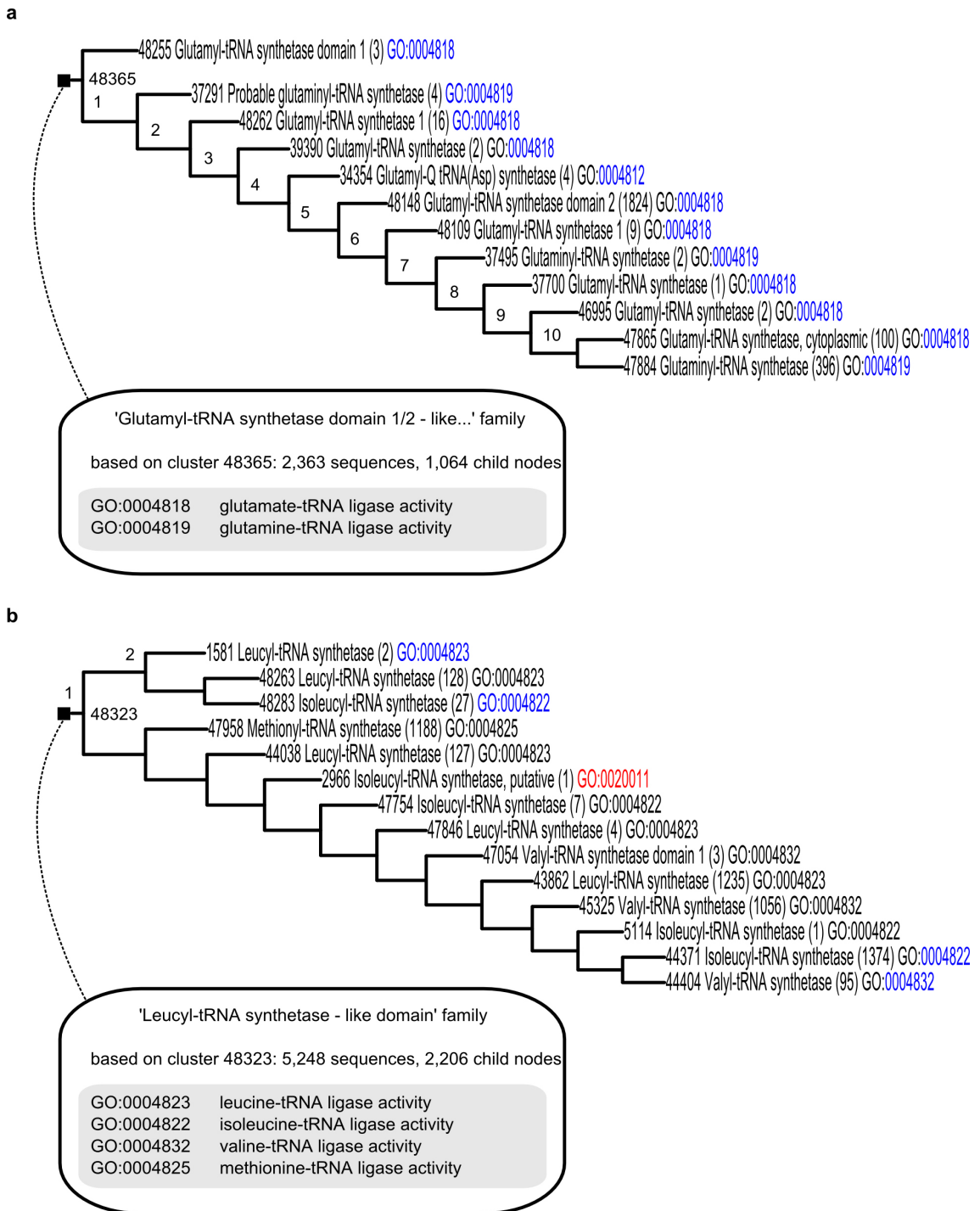
The single domain sequence in cluster **39532**, which lies between the aaRS class Ia and Ic families in Figure 5.9, has a size of ~100 aa and represents a group of functionally uncharacterised proteins of about 200 aa. These proteins are found in the chloroplast stroma (GO:0009570) of *Arabidopsis* (UniProt **Q9FKX3**) and other plants, as identified by BLAST searches on the UniProt website. No molecular function for these sequences can be predicted using InterProScan, and an attempt to model the domain's 3D structure using SWISS-MODEL (Schwede, Kopp et al. 2003) yielded no models of good quality. Based on the position of the family in the dendrogram (see Figure 5.9), it could represent a yet-to-be studied outlier group of either the class Ia or class Ic aaRSs. To further corroborate this, it would have to be verified that these uncharacterised sequences are not a result of systematic (e.g., gene prediction) errors or represent pseudo-genes. The dendrogram further suggests that the remote BLAST similarity to the Universal Stress Proteins that is exhibited by the discussed group of proteins (for example, E-value 0.0008 and sequence identity 30% for **Q9FKX3** vs. **A5GW74**) merely signals the (known) membership in the same superfamily, not functional similarity.

Closing this analysis of the  $\text{DFX}_{\text{super}}$  family partitioning of the HUP superfamily, two aaRS domain families that could only be identified using the concept of cluster chaining (see Figure 5.9; black square nodes) are particularly addressed in the following; a similar case in the HAD superfamily is described in detail in Section 5.4.1.1. Figure 5.10 shows the child cluster dendrograms and cluster core term sets of the two example end-of-chain clusters; the colour code used is the same as in Figure 5.8. In brief, cluster-function chain nodes are numbered, GO terms responsible for chain elongation are highlighted in blue, blue-marked terms outside chains indicate the first appearance of an annotation, and non-MF terms are highlighted in red. Again, some leaf clusters subsume a range of prior merges that are not shown.

Figure 5.10a dissects cluster **48365**, which gives rise to the Glu/GlnRS (class Ib aaRS) family described above. This cluster has a chain length of 10 (see Section 5.2.3). First, that illustrates the high sequence diversity among the Glu/GlnRS catalytic domains (all leaf clusters are, or are parents of, individual DFX pre-clusters; see Section 3.3.3.1). Second, it lends high confidence to the end-of-chain exception made, that is, to postulating a family-like character for the cluster. The high sequence and structure similarity of the GluRS and GlnRS catalytic domains (Woese, Olsen et al. 2000) leads to a clustering pattern that deviates from the pattern of whole-protein function (for a theoretical discussion of this phenomenon see Section 5.4.2). For example, the **GO:0004819** annotation is found in all parts of the dendrogram, interspersed with occurrences of **GO:0004818**. This is the signal that is picked up by the chaining detection algorithm (see Section 5.3.5). As none of the sequences in cluster **48365** is associated with both GluRS and GlnRS activity, the corresponding family could not have been identified based on the normal cluster assessment procedure (see Section 5.3.4).

Interestingly, the  $DFX_{\text{super}}$  protocol correctly identifies and removes a ProRS annotation that is associated with a sequence in cluster **48365**. No class I aaRS (domain) exhibits this activity (see Section 5.1.4.2). It stems from an aaRS fusion protein found in eukaryotic species (e.g., UniProt **P07814**), with an N-terminal class I (GluRS) and a C-terminal class II (ProRS) domain (Berthonneau and Mirande 2000). Therefore, the ‘proline-tRNA ligase activity’ term (GO:0004827) occurs together with a core set term (GO:0004818), and is filtered from the annotations based on its putative ‘foreign’ domain status (see Section 5.2.2).

Figure 5.10b shows a more complicated example of family identification via an end-of-chain cluster (cluster **48323**). Here, the catalytic domains of four closely related species of class Ia aaRSs (see Section 5.1.4.2) together form a domain family. The end-of-chain cluster has a chain length of two, the current minimum chain length (see Section 5.3.5). The confidence put in making the end-of-chain exception is therefore considerably lower than in the above, first example. However, making it picks up on a biologically valid chaining signal in this case. While the domains from the three closely related ValRS, LeuRS and IleRS proteins (see Section 5.1.4.2) could already be grouped into families based on end-of-chain clusters that appear earlier in the merging process (for example, cluster **5114** for ValRS and IleRS, or cluster **47054** for all three), the MetRS function is only merged in with cluster **47958**.



**Figure 5.10. The clustering dendrograms of two aaRS catalytic domain families.** These families of aaRS domains with closely related functions (boxes) were identified by chaining detection in the HUP superfamily. Shown are the underlying end-of-chain clusters (black squares) and the underlying cluster merging events; the nodes of the corresponding cluster-function chains are numbered. The names, sizes and core term sets of all clusters are shown; the ‘-like domain’ suffix is omitted in the cluster names. GO terms that are responsible for chain elongation are highlighted in blue, respectively; blue-marked terms outside chains indicate the first appearance of an annotation; non-MF terms are highlighted in red. Note that some leaf clusters subsume a range of prior merges that are not shown. The dendrograms were generated using iTOL; relative branch lengths were derived from  $-\log(E)$  values that indicate pairwise cluster similarity, as reported by COMPASS.

The MetRS sequences can only join the established domain family by detection of the chaining incident marked by cluster **48323**, that is, the fact that LeuRS (and IleRS) join the growing family cluster *again* at this point. Further, the putative IleRS sequence from *Plasmodium falciparum* (UniProt Q8I5G6\_PLAF7) in cluster 2966, which carries only a single high-quality annotation of the cellular component type ('apicoplast'; red in Figure 5.10b), is successfully 'bridged' by the chain detection algorithm (see Section 5.3.5). This case is analogous to that of the GO CC term in Figure 5.8. In summary, based on the individual MF annotations that are associated with the sequences in cluster **48323**, the corresponding domain family of four class Ia aaRSs could not have been detected with chaining detection disabled.

#### 5.4.2 The sequence footprint of domain function conservation

The Gene3D 10.2 domain architectures of nine P-ATPase transporter proteins (orthologues and paralogues) are shown in Figure 5.11, with the primary substrate ion species listed next to each MDA. The discontinuous P domain sequences in these proteins (shown in black) are all members of the same domain family in the HAD superfamily, as identified by DFX<sub>super</sub> and discussed in Section 5.1.4.1. Apart from some variability in the number of detected heavy metal binding domains (or motifs; cyan), which are not part of the four-domain core architecture of P-ATPases (see Section 5.1.4.1), all the shown proteins share this core architecture. The transmembrane domain, into which all other domains are inserted, is not yet recognised by Gene3D. The A domain is shown in magenta. The N domain, which is inserted into the P domain, is shown in grey.

The nine P domains from the proteins in Figure 5.11 can serve as an example of how domain sequences with conserved domain function that occur in evolutionarily related multi-domain proteins with (overall) differing function can show a pattern of pairwise similarities that deviates from that of their

parent proteins. This is demonstrated in Figure 5.12, and discussed below. In particular, it illustrates how high local (domain) sequence conservation can ‘disguise’ an overall functional divergence of the respective parent proteins, one of the central assumptions behind the domain family concept introduced in Section 0.

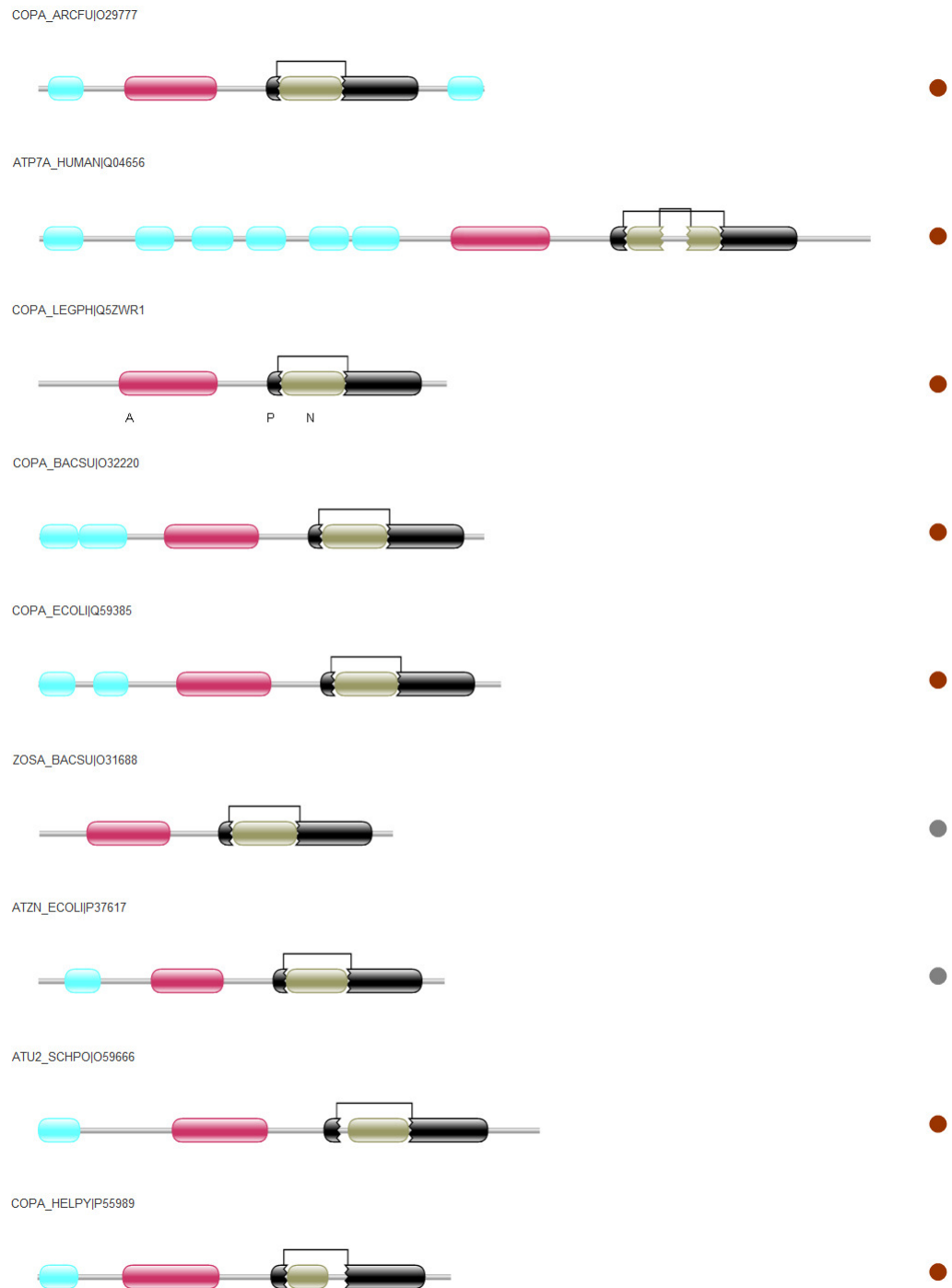
Figure 5.12a shows a full alignment of the nine P domain sequences, with the domain positions shown as part of the protein identifiers on the left. Residue conservation is indicated in the alignment by different shades of blue. The two orange residue ranges indicate the conserved phosphorylation and ATP-binding motifs (Rensing, Fan et al. 2000), as found in the P and N domains, respectively. Interestingly, Gene3D seems to extend the second half of the P domain into the N domain, which (by definition) harbours the ATP-binding motif. The two sequences marked by the grey box are the zinc transporters **ZOSA** from *B. subtilis* and **ATZN** from *E. coli*. The remaining sequences all transport copper. Figure 5.12b shows the phylogenetic tree that was derived from this alignment. Again, the above two proteins are marked by a grey box. The tree was rooted using a **COPA** orthologue from the archaeon *Archaeoglobus fulgidus*.

A first important observation made in Figure 5.12b is that the two copper transporter P domains from **ATU2\_SCHPO** (fission yeast) and **COPA\_HELPY** (*Helicobacter pylori*) are found closer in the tree to the zinc transporter P domains from **ZOSA\_BACSU** and **ATZN\_ECOLI** than to all other P domains. This would not normally be expected, since the remaining P domains all stem from copper-transporting P-ATPases too. However, it could be explained by the fact that the substrate (ion) specificity of P-ATPases is not determined (or even influenced) by the structure of the P domain but, most likely, by that of the transmembrane domain, domain M (see Section 5.1.4.1).

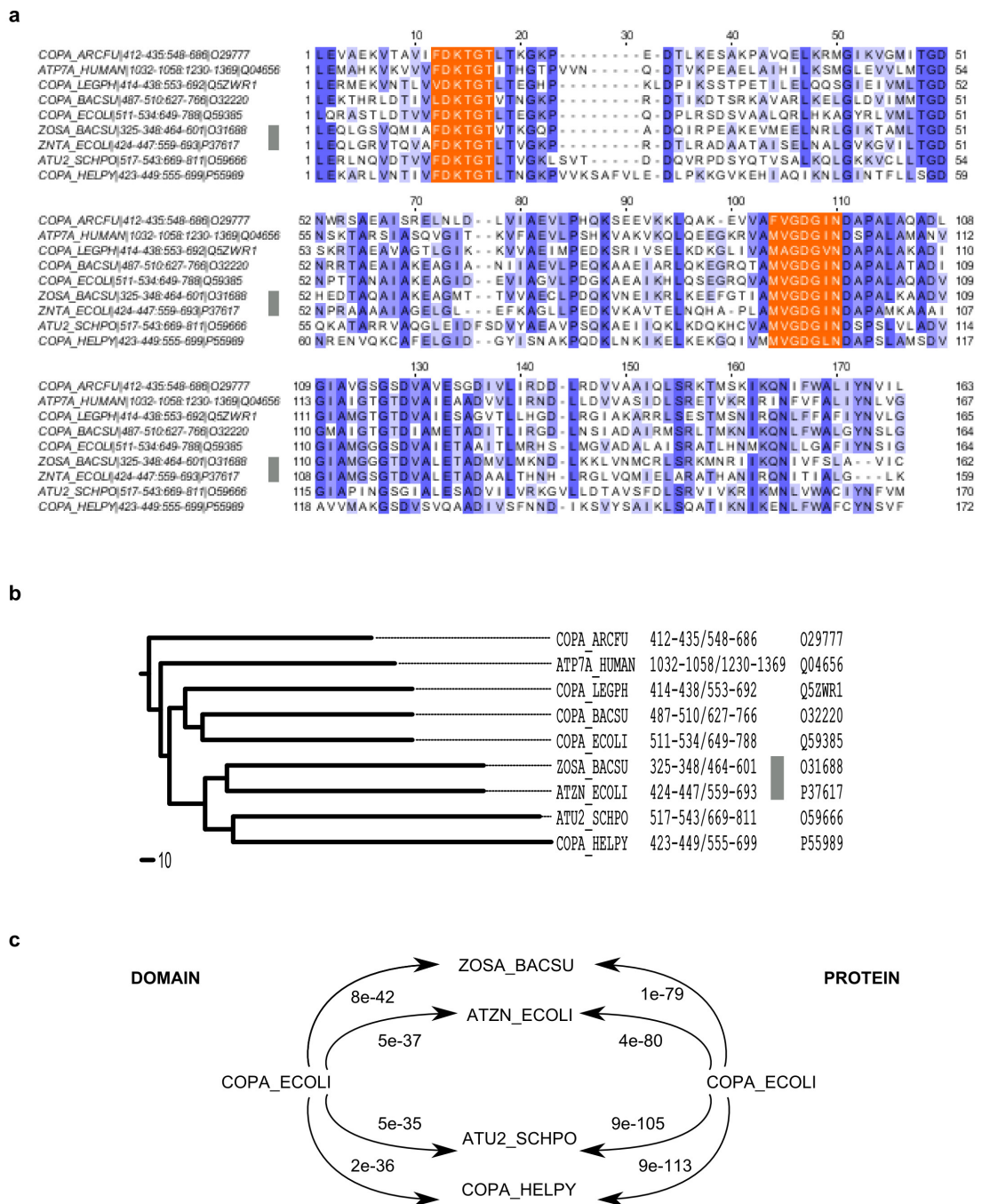


The above observations suggest that the sequence evolution of the P domain may primarily be constrained by its domain function (ATPase activity mediated by autophosphorylation), and only very little by the ion specificity of the respective P-ATPase parent proteins. From a taxonomic point of view, it is further interesting that the fission yeast domain does not cluster with the domain from its homologous (and potentially co-orthologous) counterpart in man, **ATP7A\_HUMAN**; mutations in the latter are the cause for Menkes disease (Tumer, Moller et al. 1999).

Based on the above-described, unexpected ‘disorder’ in the domain-based phylogenetic tree shown in Figure 5.12b, the pairwise sequence similarity relationships underlying this tree were investigated further. Figure 5.12c shows the E-values reported by NCBI Blast2 (pairwise protein BLAST) with default settings when scanning the P domain of *E. coli* **COPA** and the whole-protein sequence against the P domains and whole-protein sequences of four of the other homologues, respectively. The four targets comprised the two zinc transporters **ZOSA\_BACSU** and **ATZN\_ECOLI** on the one hand and the two copper transporters **ATU2\_SCHPO** and **COPA\_HELPY** on the other hand (the corresponding source species are stated above).



**Figure 5.11. The Gene3D domain architectures of nine homologous P-ATPase proteins.** Domains from four different Gene3D superfamilies are identified in these proteins. Among those are the three cytoplasmic P-ATPase domains (see Figure 5.2), the actuator (A, magenta), phosphorylation (P, black) and nucleotide-binding (N, silver) domains. Note that most P-ATPases are associated with one or several copies of a Heavy-Metal-Associated (HMA) motif (cyan) that plays a role in binding the respective ion species for subsequent transport. The different ion species transported are shown next to each protein, with copper in brown and zinc in grey; the primary substrates are shown in the case of multi-functional transporters. The CATH domain codes are, in N- to C-terminal order, 3.30.70.100 (HMA), 2.70.150.10 (A), 3.40.50.1000 (P) and 3.40.1110.10 (N). Note that the discontinuous P domain sequences of the shown proteins are aligned, in the same order, in Figure 5.12a.



**Figure 5.12. High sequence conservation in functionally equivalent domains from functionally divergent multi-domain proteins.** (a) shows an MSA of the nine P-ATPase P domains from the proteins in Figure 5.11. The phosphorylation and ATP-binding motifs are highlighted in orange; residue conservation is otherwise indicated in shades of blue (darker = stronger). (b) the corresponding phylogenetic tree. (c) BLAST E-values for comparisons on the domain and protein levels. Zinc-transporting sequences are marked with grey boxes. The alignment was generated with MAFFT and visualised with JalView; the tree was produced with the ‘average distance using BLOSUM62’ option of JalView.

The difference in the similarity signals (BLAST E-values) obtained on the whole-protein and domain levels for the same proteins, as stated in Figure 5.12c, is striking. On the protein level, the results are as would be expected for a protein family that has diverged very early in evolution: those target proteins that share the substrate specificity of the query protein (copper) are clearly more similar to the query than those that have a different specificity (zinc). This corresponds to the different expected evolutionary relationships between these proteins and the query, respectively: orthology and paralogy. In fact, the sequence conservation signal for the homologues with shared function is strong enough to ‘bridge’ about two billion years of evolution in the case of *E. coli* and fission yeast (Gu, Zhang et al. 2005). On the domain level, however, the picture is entirely different (see Figure 5.12c). Here, the different target proteins and their substrate specificities are not readily distinguishable by E-value. In fact, the similarity signal is even inverted. That is, the P domain of the query sequence, COPA\_ECOLI, shows higher similarity to the two P domains from zinc-transporting proteins than to those that stem from the other, copper-transporting ATPases. The difference in the whole-protein and domain signals is especially prominent when comparing the similarity relationships between the query and ZOSA\_BACSU and between the query and COPA\_HELPY (the four outer arrows in Figure 5.12c).

On the protein level, COPA\_ECOLI exhibits the highest similarity with COPA\_HELPY; this is expected, since the two proteins are close orthologues with matching substrate specificity. On the domain level, however, it shows the highest similarity with the zinc-transporting ZOSA\_BACSU protein, while its similarity to COPA\_HELPY is the second-lowest. Further, the large evolutionary distance between the two copper-transporting target proteins from fission yeast and *H. pylori* and their almost equal amount of similarity to the query protein in the P domain underline how little the domain has changed over this distance. In fact, it may even have evolved ‘back and forth’ on the sequence level, while retaining a stable structure and function. This is

indicated by the fact that the *Helicobacter* domain is less similar to the *E. coli* query domain than that from *B. subtilis*, whilst being evolutionarily closer. However, such specific hypotheses would have to be corroborated by further phylogenetic and structural studies (see also Section 5.5.5).

Notably, all the proteins used in the above-described studies are reviewed SwissProt entries. This means their sequences and functions have been manually curated. Further, while COPA\_ECOLI, for example, can transport different ion species, there is no indication that it can also transport zinc, or that the substrate specificities of the copper and zinc transporters that were compared overlap otherwise. The two Gene3D superfamilies that contain the A and N domains of P-ATPase proteins are not split by DFX<sub>super</sub> (unlike the HAD superfamily, containing the P domain family) but rather form a single, large family each. This is in agreement with the fact that both types of domains are exclusively found in P-ATPase proteins, that is, fulfil only a single domain function. Finally, the similar lengths of the P domain sequences used (see Figure 5.12a) and the apparent completeness of protein domain decomposition in all cases (see Figure 5.11) make it highly unlikely that errors in Gene3D domain assignment play a role in the observations made. Taken together, the above-described studies on the P domain and the apparently similarly ‘unspecific’ clustering behaviour of the A and N domains (which has yet to be studied in more detail) support the current view of P-ATPase substrate (ion) specificity and sequence divergence being largely mediated by the transmembrane domain M (see Section 5.1.4.1).

### 5.4.3 Examples of annotation complexity and inconsistency

The patterns of Gene Ontology function annotations that are associated with groups of related proteins are frequently complex, and sometimes incomplete or inconsistent. Similar attributes hold for the GO molecular function DAG.

A selected set of example cases that were observed during the development of  $\text{DFX}_{\text{super}}$  is discussed in the following.

A striking example of how annotation diversity in sets of closely related proteins can pose a serious challenge to supervised family identification is the P-ATPase P domain family that  $\text{DFX}_{\text{super}}$  identifies within the HAD superfamily (see Section 5.1.4.1 and Figure 5.8). For the underlying cluster, Figure 5.13 shows all GO MF terms that are assigned, as most-specific terms, to at least one of the sequences it contains (red and yellow boxes). Note that the resulting, relevant part of the GO DAG is large, and therefore had to be broken up into two parts for visualisation. These are separated by the curved, dashed line in the middle, and connected by the numbered connection points.

The yellow boxes in Figure 5.13 highlight the terms that belong to the core annotation of the P domain cluster, as derived by  $\text{DFX}_{\text{super}}$ . This comprises all terms deemed to describe most specifically and uniquely the functions of the domain sequences in the cluster (see Section 5.3.2). The 273 (out of 1,962 total) domain sequences in the cluster that are associated with high-quality GO MF protein annotations exhibit considerable variability in the specificity of these annotations, as shown by the ‘patchy’ distribution of most-specific GO MF annotations. This reflects the varying amount of current empirical knowledge concerning their functions.

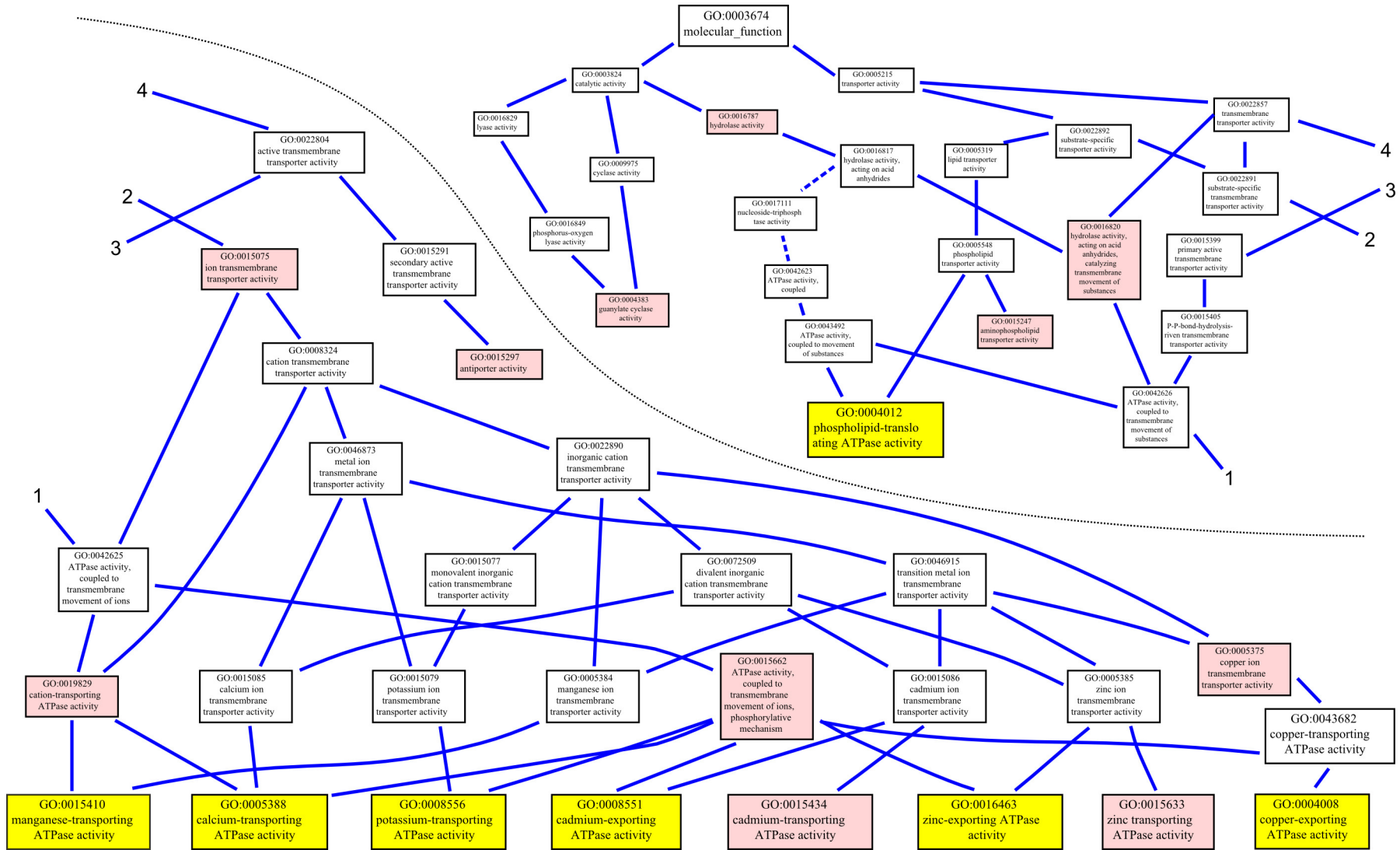
The annotation ‘breadth’ of the sequences in the P domain cluster varies as well. This is captured by the distribution of the number of annotated, most-specific MF terms assigned per sequence: 251 sequences have one term, 16 have two, two have three, and one sequence has five terms assigned; three sequences only have problematic MF terms assigned (which are ignored; see Section 5.3.1). Further, when all most-specific sequence annotations are propagated up the GO MF DAG, 266 sequences share the coarse ‘hydrolase’ term (GO:0016787), whereas only 150 share the much more specific term

‘ATPase activity, coupled to transmembrane movement of ions, phosphorylative mechanism’ (GO:0015662).

Apart from the high variability in the ‘richness’ of the annotations available for individual sequences, as discussed above, a second challenge for the supervised family identification protocol are inconsistencies in the used annotation system, the Gene Ontology, itself. In particular, these are logical inconsistencies in both patterns of existing (or non-existing) terms and in the interconnectivity of terms in the GO molecular function DAG. Figure 5.13 provides several examples of such cases. First, given the existing parent-child relationships between the ‘cation-transporting ATPase activity’ and both the ‘manganese-transporting ATPase activity’ and the ‘calcium-transporting ATPase activity’ terms, all further terms shown that refer to the ATP-driven transport of a specific cation species (the bottom row of terms in Figure 5.13) should be linked to the parent term accordingly. Second, all the terms in the bottom row (and a few others) should be consistently linked with the ‘ATPase activity, coupled to transmembrane movement of ions, phosphorylative mechanism’ term in the row above. Third, there is no apparent reason for having both a ‘-transporting’ and an ‘-exporting’ term defined in the ontology for some ion species (e.g., copper) but only the less specific ‘-transporting’ term for others (e.g., manganese). Even if there were biologically grounded reasons for these differences in the term definitions, the ‘-exporting’ terms should always be children (and not siblings) of the ‘-transporting’ terms, to preserve the logic of the GO DAG; currently this is only the case for the copper terms.

**Figure 5.13. Annotation complexity in the P-ATPase P domain family as identified by DFX<sub>super</sub>.** This GO DAG shows all MF terms that are associated, as most-specific terms, with at least one of the sequences in the end-of-chain cluster discussed in Section 5.4.1.1 (red and yellow boxes). Core set terms are highlighted in yellow. Note that the relevant part of the GO MF DAG had to be broken up into two parts for visualisation (dotted curve); these are connected by numbered connection points. Dashed edges correspond to omitted intermediate terms. The initial DAG was generated with AmiGO.

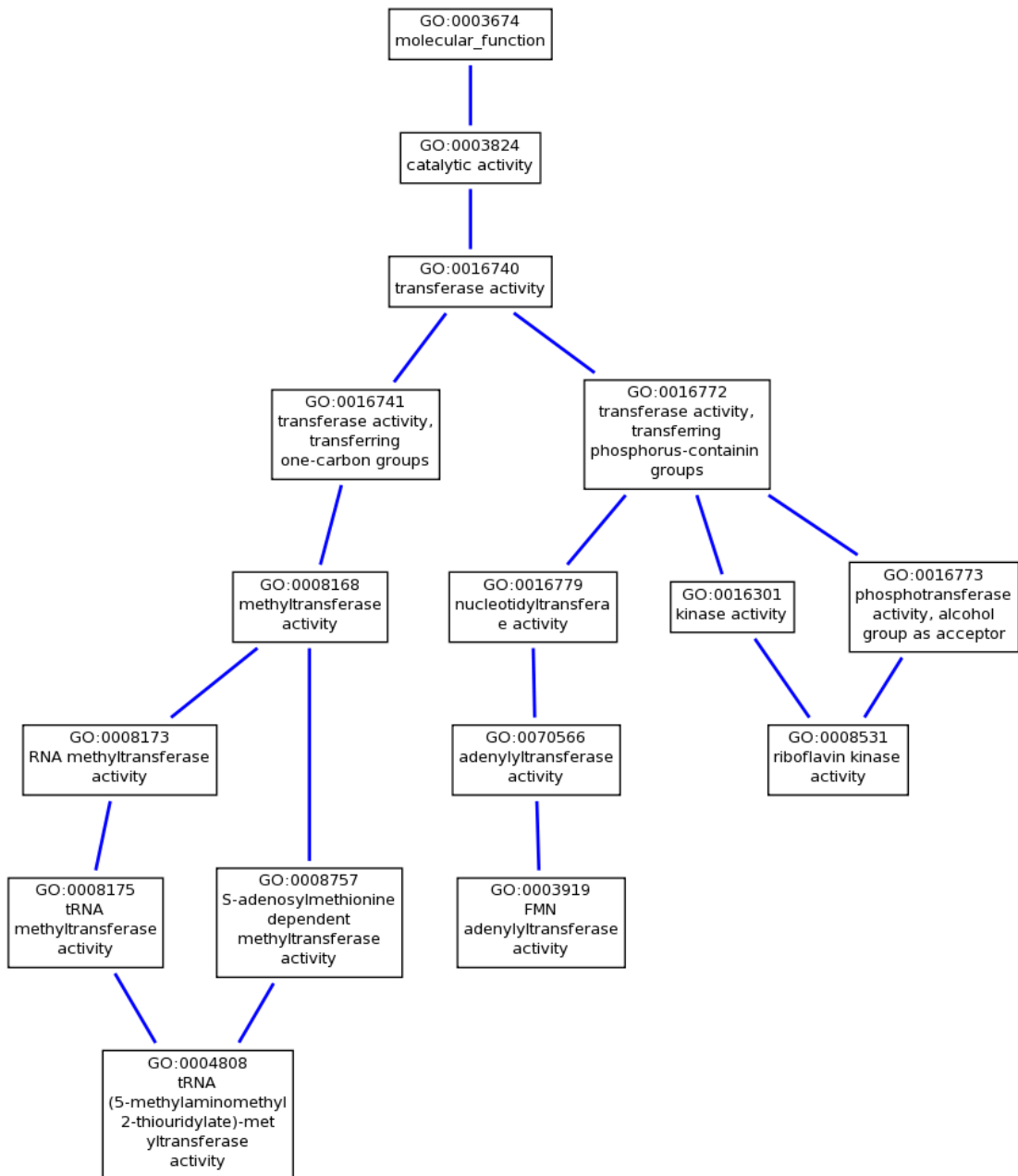




Another example of functional complexity, accompanied by annotation incompleteness, is the case of a SwissProt-reviewed but so far (apparently) not specifically studied trifunctional protein from *Mycoplasma gallisepticum*. This ‘Trifunctional protein ribF/mnmA’ (UniProt **Q7NBZ0**) exerts its three functions via three distinct domains. Interestingly, all three activities are relatively closely related transferase functions (see Figure 5.14), therefore sharing the first EC digit: ‘FMN adenylyltransferase activity’, ‘riboflavin kinase activity’ and a tRNA-specific methyltransferase activity. While the combination of the former two functions (domains) is observed in many bacteria, the additional tRNA methyltransferase (Rmt) domain is only found in the *Mycoplasma* protein. Otherwise, this domain occurs either alone or with further domains in bifunctional proteins.

The Rmt domain belongs to the HUP superfamily (see Section 5.1.4), and gives rise to the ‘tRNA-specific 2-thiouridylase mnmA -like’ domain family identified by DFX<sub>super</sub> (see Figure 5.9, close to the vertical centre). The sequences in this family stem from a mixture of single- and multi-domain proteins, as mentioned above, and can only be identified when the underlying end-of-chain cluster (cluster **48353**) is identified as such.

The normal assessment protocol (see Section 5.3.4) does not judge this cluster functionally coherent. This is because the trifunctional *Mycoplasma* (parent) protein is associated with only two most-specific GO MF terms (describing two of the three domain functions), which form the cluster’s core term set (see Section 5.3.2), but lacks the annotation for the function of the Rmt domain, unlike the other parent proteins. In consequence, the core set comprises all three GO MF terms, whilst none of the sequences in the cluster is associated with all of them.



**Figure 5.14.** The three transferase functions corresponding to the three domains of Trifunctional protein *ribF/mmmA*. A trifunctional protein from *Mycoplasma gallisepticum* (UniProt Q7NBZ0) can carry out the three functions shown (leaf GO terms). The GO DAG was generated with AmiGO.

## 5.5 Conclusions and future work

Several conclusions are first drawn in the following two sections, concerning both the benefits and potential caveats of the developed  $DFX_{super}$  protocol for domain family identification. Thereafter, potential changes in data usage and in the use of the chaining concept are discussed. The chapter then closes with several suggestions for specific follow-up analyses of the presented results.

### 5.5.1 Uniqueness and aim of the developed protocol

The development of the supervised protocol discussed in this chapter was necessary to be able (i) to make use of the wealth of available curated protein annotation data in domain family identification and (ii) to process the largest existing superfamilies at all. Both can not be achieved using the unsupervised, exhaustive-clustering based protocol that was developed earlier and is presented in the above chapter. Supervised protocols for automated protein family identification are relatively sparse (see Section 5.1.1). Established resources that are based on such protocols are even rarer. Instead, manual curation and/or the use of unsupervised protocols are the norm. On the domain family level, fully-automatic supervised methods are practically non-existent at this point. These observations are somewhat surprising, as it could be expected that supervised methods are generally easier to implement than unsupervised ones, owing to their ‘information advantage’.

The current lack of supervised protocols for protein domain family identification can be attributed to four factors. First, the scarcity of high-quality protein function annotations in the past. Such data is now increasingly available, in an increasingly organised form. International consortia such as the Gene Ontology are the major driving force behind this. Second, the non-trivial problem of mapping from whole-protein annotations to the domain function level. This is probably the most complicated step in any such protocol, and the most difficult to automate (see Section 5.1.2). Third, the ‘success’ of the *protein* family concept in the past. Most studies on protein families are essentially studies on protein function. For this reason, they focus specifically on orthology and related concepts of close homology (see Section 1.2.1.2). These are concepts whose meaning in the context of protein domains (and multi-domain proteins) is not yet clearly defined (see Section 1.2.3). Further, when individual studies discuss the functional subgroups of specific *protein* superfamilies, it is not always made explicit that such

superfamilies are usually defined on the basis of a certain core domain or a set of such (see Section 1.2.2.1). The fourth and last factor is the concentration on the manual or semi-manual generation of family libraries in the past. Established family resources such as Pfam were created at a time when it was still possible (for any resource) to concentrate the efforts of manual curators on individual families, even on manually ‘stitching together’ the corresponding alignments. This was primarily due to the much lower amount of available sequence data, which also meant that only a fraction of the protein and domain families known today were characterised (Sonnhammer, Eddy et al. 1997). While such manually curated databases are of enormous importance for the bioinformatics community, only very few will continuously be able to afford the resources to manually curate the equally enormous (and growing) amount of available sequence data.

The supervised domain family identification protocol presented in this chapter is the most important building block of the DFX pipeline, a means to establish a sustainable and curation-free family level below the domain superfamily level. This specific importance, compared with the unsupervised protocol, results from two factors. First, the established family level will be most accurate and most useful for those superfamilies that are associated with at least some high-quality annotations, that is, those processed with  $DFX_{\text{super}}$ . Second, by abandoning the exhaustive clustering strategy followed in the unsupervised protocol,  $DFX_{\text{super}}$  makes it possible to process even the largest superfamilies in Gene3D, with currently up to 1,000,000 member sequences.

### 5.5.2 The limits of rule-based heuristics

An early decision in the development of the supervised protocol ( $DFX_{\text{super}}$ ) was that rule-based approach should be followed, instead of an approach based on scores and thresholds. Specifically, this refers to the way the protocol deals with the GO function annotations of (sets of) proteins. The decision in

favour of a rule-based algorithm was based on two assumptions. First, researchers that use the established families as the basis of their studies (e.g., a metagenomic function enrichment analysis) or publications (e.g., a review of a specific domain superfamily) would need to know exactly *how* these families were derived. A ‘black box’ approach, as, for example, using GO semantic similarity (GOSS) (Pesquita, Faria et al. 2009) thresholds to identify functional families, would complicate the explanation of how the protocol works in *biological* (function) terms. When assessing the specific relationships between individual annotations and sequences (see Sections 5.3.2 and 5.3.4), however, this is easily possible. It mimics the way a human curator would decide whether or not to group sequences into the same family. This consideration is analogous to the decision between using relatively basic concepts, such as sequence clustering, sets of annotations and so forth, and using highly specific machine learning concepts such as SVMs or Neural Networks. The second reason for following a rule-based approach in  $DFX_{\text{super}}$  was that the development process itself (which is still ongoing) is made easier and more transparent by defining rules first and implementing them thereafter, instead of using a threshold-based ‘trial-and-error’ approach. All this does not mean, however, that a (purely) rule-based protocol is necessarily the best way to tackle the supervised family identification problem. To give an example, a single, empirically derived threshold (e.g., a minimum average pairwise GOSS score) that is used to support or reject a family relationship may be preferable over a set of ten different rules when both yield comparable overall performance.

Certain limitations of the rule-based approach have emerged during the development of  $DFX_{\text{super}}$ . In general, there is an obvious clash between striving to detect and account for all observed (and imaginable) exceptions, to produce biologically sound families, and the aim of simplicity or, at least, transparency. More complex rule sets than those described here were considered during the development of the protocol. While some of those

yielded results similar in quality to the results presented here, it was not clear *why* and in which circumstances they would (or would not) work.  $DFX_{\text{super}}$  deals with both sequence and function (annotation) similarity, and with thousands of superfamilies of highly variable size and diversity.

For the above considerations, it was one of the goals in developing  $DFX_{\text{super}}$  to implement a set of rules that would be intuitively understood. Further, these rules were to be as ‘future-proof’ as possible. For example, the currently limited quality of GO annotations may lead to rules which are over-fitted and break in the future. A rule that may work in many cases at this point, for example, because it exploits a certain annotation ‘habit’, or a certain level of annotation specificity, must not necessarily work well when these conditions change. For example, GO annotations can show genome- and annotator-specific characteristics, such as the average number of terms assigned per protein (Buza, McCarthy et al. 2008). Further, individual proteins are frequently annotated with different experimental and/or electronic methods (evidence codes), at different levels of specificity (Park, Kim et al. 2011). As soon as these things improve, however, any rule that takes into account these ‘teething troubles’ of GO could become detrimental to the performance of the protocol implementing it.

The problem of grouping proteins or protein domains by function, based on (often sparse) GO annotation data, has so far not been widely addressed (apart from the developed measures for GOSS). This stands in contrast to, for example, the field of sequence-based protein function annotation. Further, the problem is still a ‘transient’ one, in the sense that certain heuristics (rules) that work, as of now, cannot automatically be assumed still to work in the future. The reason for this is the expected further evolution of the GO annotation system; for example, logical changes to the individual DAGs, interconnections between them, and so forth. It is even conceivable that a more radical development could alter the challenges posed fundamentally. For

example, the Protein Ontology (Natale, Arighi et al. 2007; Natale, Arighi et al. 2011) could at one point become integrated with GO, in an effort to provide function annotations to protein sequence segments below the whole-chain level, such as domains or residue motifs.

### 5.5.3 Potential improvements to data usage

The  $DFX_{\text{super}}$  protocol depends on a variety of preceding steps and decisions made in the data preparation stage. These include the use and filtering of annotation data. In addition, the Gene3D domain assignments for different proteins (inherently) provide information on domain architecture; this data is so far not being used in  $DFX_{\text{super}}$ . Potential improvements to the protocol in these aspects are discussed in the following.

So far, the GO annotation data for  $DFX_{\text{super}}$  that is retrieved from UniProt GOA is filtered in a relatively strict manner (see Section 3.3.2), to ensure that the annotations used in the protocol are of high quality. However, this can complicate the family identification process. In a minority of cases sequences can ‘lose’ essential annotations in the filtering process. This can happen, for example, when a non-curated sequence in UniProtKB-TrEMBL has a high-quality GO annotation for one of its partial functions, mediated by domain A, but only a low-quality (electronically transferred) annotation for another partial function, mediated by domain B. When low-quality annotations are filtered out, the B domain sequence (like the whole protein) then remains associated with only the annotation for domain A, which may be an entirely different type of function. When processing the B domain’s superfamily, the regular cluster assessment protocol (see Section 5.3.4) therefore cannot correctly group the domain into a functional family with its sequence relatives (assuming these are all single-domain proteins with only domain B and the corresponding B function annotation). In contrast, if the low-quality B function annotation of the TrEMBL protein is not filtered out (and its B



domain sequence remains annotated with both the A and B functions), the core set concept would make  $DFX_{\text{super}}$  (correctly) disregard the A function entirely in family identification (see Section 0).

Similar problems with ‘uneven’ GO annotation coverage owing to annotation filtering were reported before, in the context of protein function prediction (Forslund and Sonnhammer 2008; Wass and Sternberg 2008). In conjunction with this, it has been observed in some cases that including non-curated data in the GO annotation datasets used was not detrimental (or even beneficial) to the overall performance of annotation-based methods. For example, in Schug, Diskin et al. (2002) the authors state that ‘...including IEA annotation yields significantly greater coverage (67%) with essentially the same reliability.’ In summary, while the current annotation filtering scheme of DFX is thought to be well-balanced and does not exclude electronic annotations entirely, the optimal filtering strategy may have yet to be found. It can further be expected to change over time, as the reliability of different GO evidence types is periodically reassessed (Camon, Barrell et al. 2005; Jones, Brown et al. 2007; Buza, McCarthy et al. 2008; van den Berg, McCarthy et al. 2010).

It is generally conceivable to use additional types of protein annotation data in  $DFX_{\text{super}}$  in the future, on top of the Gene Ontology data. Obvious choices would be enzyme molecular function annotations from the KEGG Orthology (Kanehisa, Goto et al. 2004) and EC systems; the latter is already used in the analysis of the produced families (see Chapter 6). In conjunction with the current protocol for family naming (see Section 3.3.5), it has further become clear that protein names (from UniProtKB) could potentially be exploited when any other annotation data are missing. It has been shown repeatedly that the annotation data provided by different resources is complementary (Koski, Gray et al. 2005; Schmid and Blaxter 2008; Sun, Kim et al. 2009). More importantly, corresponding (but independently generated) data from more than a single annotation system could convey increased (or decreased)

confidence in certain family partitioning decisions. An example scenario is outlined below.

In the case of erroneous GO annotations, chaining detection (see Section 5.3.5) can currently lead to an overly ‘liberal’ mixing of domain sequences with different functions. This could be partially avoided by using a threshold heuristic that takes into account the sizes of the merged clusters, where increased caution is suggested when both sibling clusters are large and the annotation signal is weak. Additional annotation evidence for or against specific merges (for example, two different 3<sup>rd</sup> level EC numbers) would be a more straightforward means of avoiding it.

Despite the potential benefits of using further types of annotation data, as outlined above, this could also introduce new problems. To name the most important ones, the protocol would be further complicated, the annotation ‘habits’ and sources of each annotation system with regards to protein domains would have to be analysed separately first (see Section 5.1.3), the mentioned systems do not use evidence codes like GO does (the reason why high-confidence EC annotations had to be derived by mapping from the respective GO annotations for the analyses presented in Chapter 6), and all systems work with different levels of functional granularity.

Another type of data that could potentially be used to enhance the performance of  $DFX_{\text{super}}$  for individual superfamilies is domain architecture information. This is inherently available in Gene3D, which assigns domains to proteins. Currently, the supervised protocol does not use information on the MDA context(s) in which the members of a given domain sequence cluster appear. In the simplest case, this information could be used in a binary way, to answer the question ‘are there other domains in the parent proteins for this cluster?’. The core set concept was primarily designed to identify (and thus be able to ignore) protein annotations that arise from domains other than that

under analysis (foreign annotations). Knowing whether or not a given protein has more than one domain would certainly simplify this process to some extent. However, MDA information cannot solve the more general problem of automatically mapping between the function annotations of proteins and their individual domains (see Section 5.1.2). Whenever a protein has more than one domain, which accounts for the majority of proteins, it can only be estimated but not verified with certainty which domains give rise to which functions (annotations). The most comprehensive and therefore probably most powerful approach of incorporating MDA information in the family identification process would share characteristics with the approach currently used by DFX in domain-based protein function annotation (see Section 3.3.7); this is because the two processes both have to perform the above-mentioned mapping task, in opposite ‘directions’. It can be outlined as follows.

A protocol that takes into account MDA information in the family identification process can be outlined as follows. First, for a given domain superfamily  $F$ , a list of all co-occurring superfamilies (COFs) is generated. Second, the annotation data for  $F$  and all COFs are compiled. Third,  $F$  is processed in the context of these additional data to derive families. In particular, whenever annotations from multi-domain proteins are associated with a domain sequence cluster whose core annotation set is to be compiled (see Section 5.2.2), it is assessed for each annotation whether it also appears with at least one protein that contains a COF domain (and possibly further domains) but not an  $F$  domain. If that is the case, it increases the probability of the respective term (function) not being associated with the  $F$  domain cluster under analysis. If this basic protocol is translated into a probabilistic one, based on term-superfamily association frequencies (as done in the protocols described in Section 3.3.7), it could boost the accuracy of family identification, primarily in promiscuous domain superfamilies.

#### 5.5.4 The future of the chaining concept

The concept of chaining was introduced relatively late in the development of  $DFX_{\text{super}}$ . First analyses such as those presented in the present chapter show that it has a tremendous impact on the level at which common domain function can be captured, and that the derived families of domain sequences ‘make sense’ biologically. The phenomenon that domains with identical or highly similar functions are conserved to a (much) higher extent than the respective parent proteins, in both sequence and function, therefore seems to be relatively frequent. That domains with a specific, conserved function should exhibit this behaviour may not be surprising, given that it has long been established for conserved residue patterns that are functionally or structurally important (overall) in protein families that contain sequences with (otherwise) different functional specificities (Pazos and Sternberg 2004; Sankararaman and Sjolander 2008; Kalinina, Gelfand et al. 2009).

It is important to note that the use of the chaining concept only makes sense when the sequence unit clustered is (i) assumed to be a distinct functional unit and (ii) short enough, so that it is unlikely that functionally neutral mutations (over time) lead to clearly discernable patterns (clusters) among functionally equivalent sequences. For these reasons, chaining is a phenomenon that is observed when clustering domain sequences, not whole-protein sequences. In other words, it is highly unlikely that two full-length proteins with different but similar functions would show higher overall sequence similarity to each other than to other proteins with the exact same function, respectively. On the domain level, however, this seems to be possible (see Section 5.4.2).

Until the chaining phenomenon has been verified to be a biological phenomenon in the above-described sense, the concept must be used with caution. It is clear that, in a certain number of cases, detected instances of chaining will have entirely artefactual causes (see Section 5.2.3). For example, a

single erroneous protein annotation (or two, when a minimum chain length setting of two is used) can mislead the detection algorithm into ‘assuming’ that a case of chaining based on domain sequence and function conservation is observed. The manual inspection of different chaining events (see, for example, Section 5.4.1.2) can reveal annotation errors, especially when these events are based on the annotations of individual (or a few) sequences amongst many very similar ones. Another potential source of chaining are errors made in the course of (heuristic) sequence clustering. To identify such errors, a non-heuristic clustering method could be used with a set of test superfamilies, and the family identification process repeated based on the produced, alternative dendrogram, respectively. No matter what the main causes for chaining are, it is a phenomenon that has so far been observed with considerable frequency and often leads to ‘biologically meaningful’ domain families based on the concept of domain function. It should therefore be investigated further.

There exists a specific rule in the chaining detection protocol that should be addressed first in its further development. This is the minimum chain length criterion, as described in Section 5.3.5. It is expressed in the parameter  $L_{\min}$ , which dictates how many consecutive cluster merges in the clustering dendrogram have to exhibit chaining characteristics to consider the respective end-of-chain cluster in family identification. The setting of  $L_{\min}$ , currently a value of two, greatly influences the family partitioning derived in large and diverse superfamilies. It is primarily a means to account for artefactual signs of chaining, for example, owing to erroneous function annotations. One route of investigation would be to assess whether chains that start with leaf clusters of the overall clustering dendrogram should be considered more relevant than such that only start later. The reasoning behind this would be that ‘late chains’ are more likely to arise from sequence clustering errors and, at the same time, often have a greater impact on the resulting family partitioning. In other words, in a worst-case scenario, a few wrong sequence annotations and/or

truncations can currently collapse the superfamily into a single family with high functional variability.

### 5.5.5 Proposed further analyses

Based on the presented qualitative analyses of results, the use of several established bioinformatics tools and concepts for comparative analyses is strongly suggested. First of all, the full sequences of the proteins that contain the domains from the two analysed superfamilies should be clustered using the same algorithm as that used to cluster the domains (GeMMA; see Chapter 2), respectively, and then processed using a simplified  $DFX_{\text{super}}$  protocol. A comparison of both the raw clustering results and the final family partitionings derived on the protein level with those obtained on the domain level could then be made (see, for example, Figure 5.12). The points at which the whole-chain sequences cluster differently from the domain-only sequences, and/or lead to different family partitionings, could help further to corroborate (or reject) different assumptions made in the development and analysis of the  $DFX_{\text{super}}$  protocol (see above). This refers primarily to the chaining concept (see Section 5.2.3), but also to different potential sources of error, such as Gene3D domain assignment.

A comparison of the similarity relationships between whole proteins and between their individual domains, at the level of clustering and derived family dendrograms, would essentially repeat on the large scale what was done using an example set of homologous P-type ATPase proteins in Section 5.4.2. The increasingly used framework of the sequence similarity network (Song, Joseph et al. 2008; Atkinson, Morris et al. 2009) could help to visualise the results of such an endeavour. In addition, the GeMMA clustering results for different superfamilies could be compared – where it is technically possible – with those obtained from traditional clustering approaches (e.g., single linkage; see

Section 2.1.2.1) and those that can be inferred from phylogenetic trees, at both the whole-protein and domain levels.

As has been demonstrated in Section 5.4.1, especially the investigation of ‘outlier’ families in the  $DFX_{\text{super}}$  family dendrograms can be fruitful to unravel the intricacies of evolution within domain superfamilies. For example, in the context of phylogenetic trees and by in-depth (e.g., taxonomic, functional, structural) characterisation of the members of small and/or apparently misplaced families, lineage-specific sequence specifics and phenomena such as horizontal gene transfer, rapid evolution and long-branch attraction (Boc and Makarenkov 2011) could be identified. Further, when working with domain instead of whole-protein sequences, it is particularly tempting to try tracing the most important factors of sequence evolution at this level: domain duplication and domain shuffling (see Section 1.1.2).

To close this section, a few examples of (biologically) very specific observations and suggested further analyses are given, as a direct result of the detailed analyses conducted for this chapter. First, several annotation errors have been identified and reported to the GO curators. For example, the fly Cryptochrome 1 protein (UniProt **O77059**) was wrongly associated with GPCR activity, as (allegedly) inferred from a mutant phenotype (IMP evidence code). The underlying publication in *Cell* (Stanewsky, Kaneko et al. 1998) does not mention this activity at all, and the respective annotation has already been removed. Second, inconsistencies in the GO DAG itself have been reported. For example, the logical inconsistency that ‘NAD<sup>+</sup> synthase (glutamine hydrolyzing) activity’ (GO:0008795) is not a child term of ‘NAD<sup>+</sup> synthase activity’ (GO:0003952), as shown in Figure 5.15.

While the two GO leaf terms in Figure 5.15 were derived from two different EC numbers (6.3.5.1 and 6.3.1.5), the described reactions merely differ in substrate (L-glutamine and NH<sub>3</sub>), and the corresponding proteins have

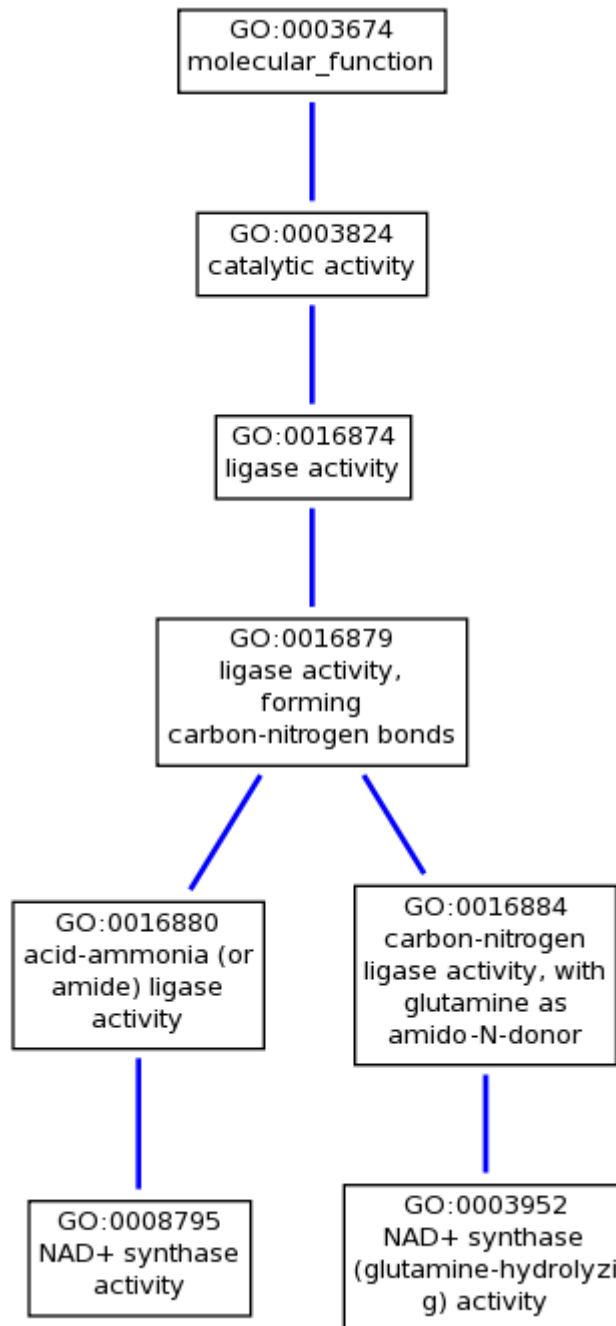
overlapping substrate specificities. It would therefore be advisable to revise the GO DAG (and potentially the EC hierarchy) in the relevant parts. At the end of this process, a novel ‘NAD<sup>+</sup> synthase’ term could be the parent of two novel terms ‘NAD<sup>+</sup> synthase, with NH<sub>3</sub> as amido-N-donor’ and ‘NAD<sup>+</sup> synthase, with glutamine as amido-N-donor’. Further GO DAG inconsistencies are discussed in Section 5.4.3.

An example of where the DFX<sub>super</sub> results could contribute to yet unsolved biological questions is the case of an intriguing fusion protein that is found in several protist species, including the Malaria parasite *Plasmodium* (Linder, Engel et al. 1999). While the N-terminal domain of this protein (e.g., UniProt Q8IHY1) closely resembles the P domain of P-ATPase transporters – specifically, Ca<sup>2+</sup> and phospholipid-transporting ones (Baker 2004) – its C-terminal domain has guanylyl cyclase activity. More than a decade after this was reported, and following partial structural characterisation of the protein, it is still uncertain (as of October 2011) whether or not the N-terminal domain is also active and, if so, what this activity is (Moon, Taylor et al. 2009; Baker 2011).

One possibility is that these proteins couple cellular Ca<sup>2+</sup> influx with cyclic GMP messaging. Their N-terminal domain gets assigned to the P-ATPase P domain family that is identified by DFX<sub>super</sub> in the HAD superfamily (see Section 5.4.1.1). This is hinted at by the ‘guanylate cyclase activity’ term (GO:0004383) close to the middle of Figure 5.13. In fact, this is the only high-quality annotation associated with the proteins of this type, as a P-ATPase activity could not yet be demonstrated. The exact position of the domain sequences from these protozoan fusion proteins in the P domain family, and in the corresponding clustering dendrogram, may provide valuable hints for further bioinformatics analyses (e.g., homology modelling). It may even be possible to devise a hypothesis as to where in early protozoan evolution the corresponding domain fusion took place. In general, the DFX<sub>super</sub> results



support (at least) an ATP binding activity for the HAD domain of these proteins and suggest further biochemical experiments in that direction.



**Figure 5.15. Inconsistently linked NAD<sup>+</sup> synthase activities in the GO MF DAG.** Based on their names and, potentially, the underlying biology, the two leaf GO terms would normally be expected to have a parent-child relationship. The GO DAG was generated with AmiGO.

## Chapter 6. Quantitative analysis of the DFX results and comparison of the two family identification protocols

A comparative analysis of the results attained when using the two different family identification protocols of the DFX pipeline, the unsupervised ( $\text{DFX}_{\text{unsuper}}$ ) and supervised ( $\text{DFX}_{\text{super}}$ ) protocols, is presented in the following sections. This extends on the qualitative assessments made in Chapter 4 and Chapter 5. The present analysis is based on family data that were produced in the first large-scale run of the DFX pipeline. This included all Gene3D domain superfamilies that were found to be associated with high-quality GO annotation data. First, the generated results are presented and interpreted. A discussion of the comparisons made follows, which is augmented by the discussion of the DFX pipeline as a whole in Chapter 7.

### 6.1 Results

In the first run of the DFX pipeline in December 2010, 1,793 (~75%) of the 2,382 protein domain superfamilies in Gene3D 9.2 could be processed using the DFX supervised family identification protocol. No high-quality GO protein function annotation data were available for the remaining 25%, which makes these (mostly small) superfamilies targets for the unsupervised protocol. Since the size of many of the superfamilies that were processed using  $\text{DFX}_{\text{super}}$  makes them intractable for  $\text{DFX}_{\text{unsuper}}$ , only the supervised family identification protocol was used.

To make a comparison of the two protocols based on function annotation data possible, despite the above, results for  $\text{DFX}_{\text{unsuper}}$  were approximated by feeding it with the (non-exhaustive) sequence clustering results that were also fed to  $\text{DFX}_{\text{super}}$ . For each superfamily, this corresponds to a clustering of all

sequences with high-quality function annotation and close relatives (see Section 3.3.3.1), instead of all sequences. The  $DFX_{\text{unsuper}}$  families produced in this way, using a generic GeMMA granularity threshold of  $10^{-40}$ , are in the following sections referred to as GeMMA40 (G40) clusters. The number of G40 clusters can also serve as a proxy for sequence diversity within superfamilies.

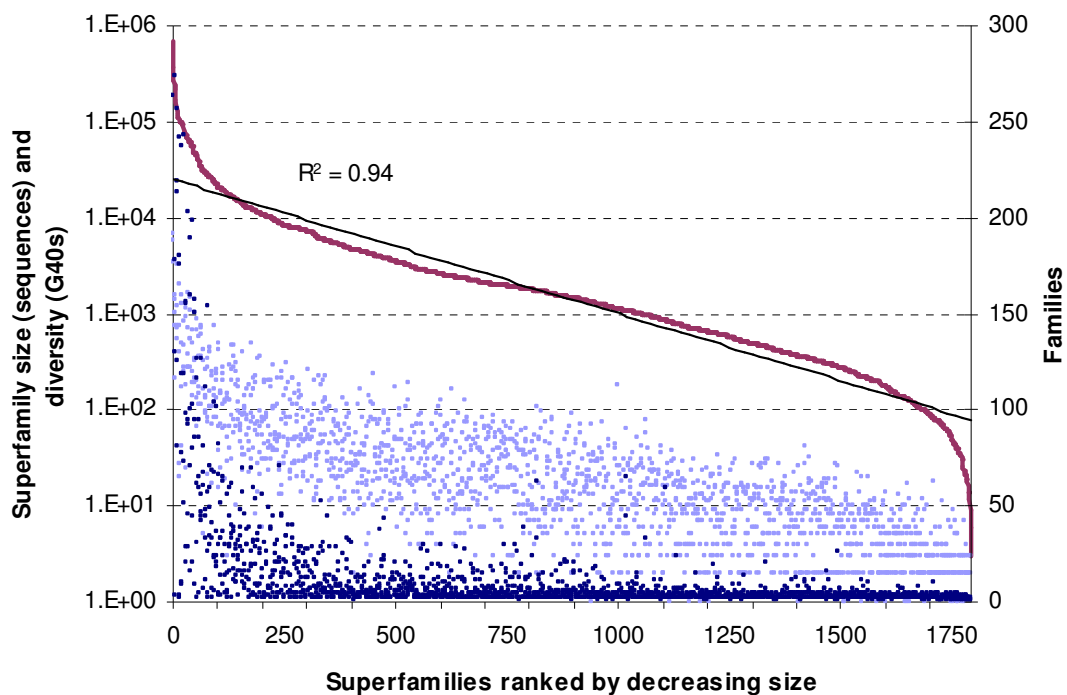
### 6.1.1 Statistics on the produced families

Figure 6.1 shows the size (magenta) and the number of families (dark blue) identified for all processed superfamilies, respectively. In addition, the number of G40 clusters obtained (light blue) is shown for each superfamily, as a proxy for sequence diversity. It can be seen that the size distribution of the functionally characterised Gene3D superfamilies under analysis shows scale-free behaviour. It approximately follows an exponential law over a wide, central range, with extreme ‘tails’ on either end. To illustrate this further: the 100 largest superfamilies together contain about 60% of the domain sequences in Gene3D.

Scale-free size distributions are a well-known phenomenon in databases that classify proteins and domains by fold, superfamily or family (Qian, Luscombe et al. 2001; Dokholyan, Shakhnovich et al. 2002; Koonin, Wolf et al. 2002; Goldstein 2008; Koonin 2011). In most cases, power-law distributions are reported. These observations are thought to indicate the tremendous evolvability and following evolutionary ‘success’ of a small fraction of all (super)folds, and the homologous sequence superfamilies and families that exhibit these folds. While a certain amount of bias in the sequence databases and an incomplete picture of sequence space will also play a role, the general ‘signal’ is too strong to be entirely artefactual.

The correlation between superfamily size and the number of identified  $DFX_{\text{super}}$  families, respectively, as shown in Figure 6.1, is weaker than the (known) correlation between superfamily size and sequence diversity ( $R = 0.77$  vs.  $0.85$ ). This can be expected, as the supervised family identification protocol uses function annotation data to identify families, on top of sequence similarity information (the clustering results). As will become clear in the following and is supported by this plot, many large families within the processed superfamilies are functionally conserved whilst being highly diverse in sequence. A protocol based on sequence similarity alone, like  $DFX_{\text{unsuper}}$ , will (over)divide such families into several smaller families.

Table 6.1 shows statistics on the ten largest superfamilies that were processed using the supervised protocol, which are also the ten largest superfamilies in Gene3D. These superfamilies are highly diverse in sequence (as seen by the G40 and CATH S35 cluster counts). Sometimes, the variation in the domain sequences detected for a superfamily by scanning the genomes (Gene3D) is considerably higher than the variation in the CATH sequences with known structure that gave rise to the respective superfamily model(s); for example, in the cases of 3.30.160.60 (classic Zinc Finger containing) and 1.20.1250.20 (Major Facilitator transporter like).



**Figure 6.1.** The DFX families identified in 1,793 Gene3D domain superfamilies of varying size and sequence diversity. Superfamily size (magenta) and sequence diversity (light blue) is plotted using the logarithmic scale Y-axis (left). The number of identified DFX<sub>super</sub> families in each superfamily is plotted using the linear scale Y-axis (right). The superfamily size data points were fitted with an exponential distribution (black line); the shown  $R^2$  value approximates the goodness of fit. Note that the three largest superfamilies have more than 300 families (603, 613, and 891, respectively). Sequence diversity is measured as the number of G40 clusters.

**Table 6.1. The ten largest Gene3D superfamilies and their diversity in sequence, structure and function.** <sup>1</sup>Sequence clusters obtained at a GeMMA clustering granularity of  $10^{-40}$ , as described in main text; <sup>2</sup>non-redundant sequence clusters at 35% sequence identity; <sup>3</sup>close structural subgroups with a maximum normalised RMSD of 5Å; <sup>4</sup>EC annotations with corresponding high-quality GO annotations only; note that this can include functions mediated by other domains in the parent proteins; \*manually named for this study.

CATH code	CATH name	Gene3D sequences	GeMMA 40s <sup>1</sup>	CATH S35s <sup>2</sup>	CATH SSG5s <sup>3</sup>	Gene3D EC3s <sup>4</sup>
3.40.50.300	P-loop NTP hydrolase	676,037	3,405	208	54	33
3.30.160.60	Classic Zinc Finger containing	369,184	6,860	23	2	1
2.60.40.10	Immunoglobulin	266,818	5,960	278	44	11
3.40.50.720	NAD(P)-binding Rossmann	242,209	1,419	203	38	52
1.10.10.10	Winged Helix DNA-binding	233,908	1,553	174	52	9
3.30.70.270	Reverse Transcriptase related*	223,046	210	19	6	4
1.20.1250.20	Major Facilitator transporter like*	165,937	1,017	2	1	1
3.40.190.10	Periplasmatic small ligand binding*	160,541	768	103	25	11
3.40.50.2300	Natriuretic peptide receptor*	126,816	553	103	15	12
1.25.40.10	Tetratricopeptide repeat*	122,066	1,177	24	5	11

### 6.1.2 The scale-free size distribution of families and superfamilies

The scale-free size distribution that is observed for protein domain superfamilies is also commonly observed at the family level. This is supported both by manual inspection of the obtained family distributions in the large superfamilies discussed above and the power law fits shown in Figure 6.2. In Figure 6.2c, the average distribution of relative family sizes within three sets of superfamilies is shown in a double-logarithmic plot. The same is done for the size of the corresponding superfamilies in Figure 6.2a, and the number of G40 clusters within these superfamilies in Figure 6.2b. The three superfamily sets analysed are subsets of the total set of Gene3D 9.2 superfamilies that were processed with  $DFX_{\text{super}}$ . Out of this total set, they comprise all superfamilies with at least 10,000 sequences (171 superfamilies; green), 20,000 sequences (94 superfamilies; pink) and 100,000 sequences (14 superfamilies; blue), respectively. For the family size distributions in Figure 6.2c, only superfamilies with at least ten identified families were included in the three sets, respectively.

The power law fits and corresponding  $R^2$  values shown in Figure 6.2 can only give a rough indication of how well the superfamilies' G40 cluster and family size distributions are described by a power law function. However, these data confirm the general trend that can be observed when individual superfamilies and families are manually inspected. This scale-free behaviour, with few very large and many small (super)families, and a wide range of (super)family sizes in general, is a recurring observation (see above). Note that Figure 6.2a is merely a 'reformulation' and extension of Figure 6.1.

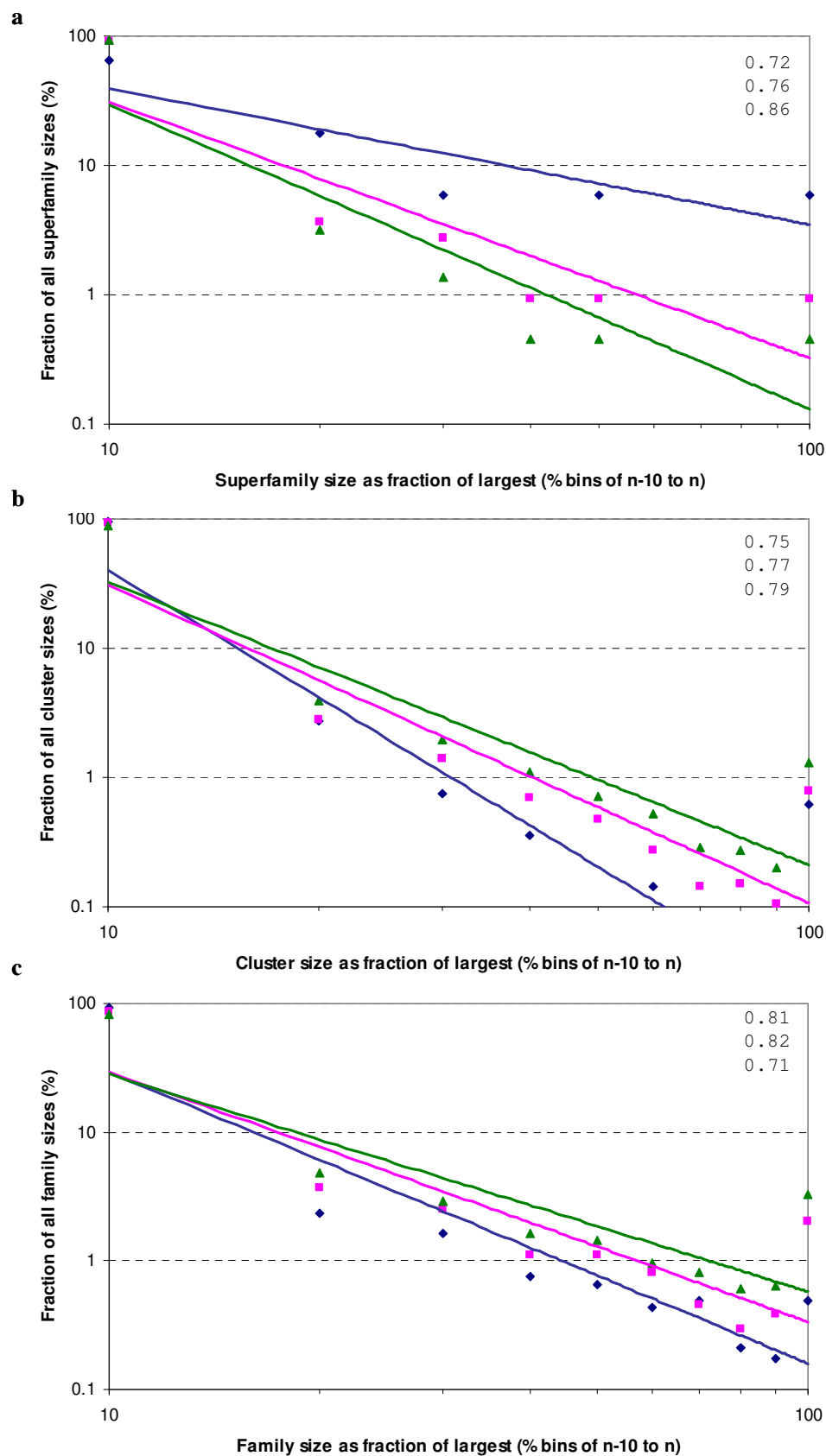
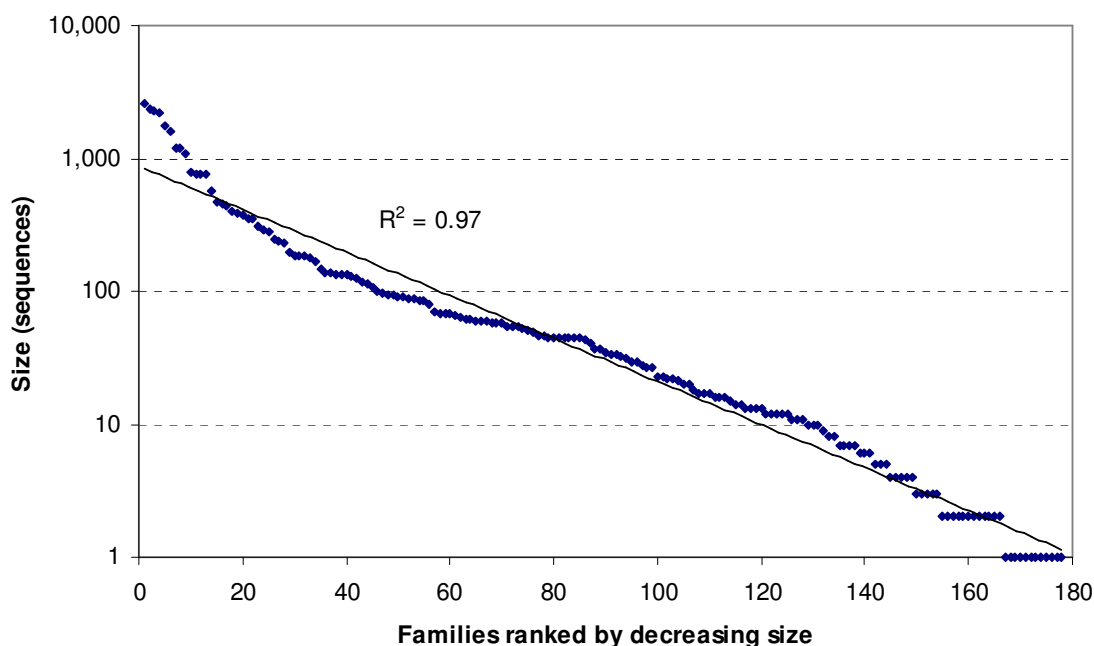


Figure 6.2. The scale-free size distribution of domain superfamilies and their DFX families. (a) The total set of superfamilies processed; (b) G40 clusters in these superfamilies; (c) DFX<sub>super</sub> families in these superfamilies. Each double-logarithmic plot is based on a histogram with ten bins of range ten percent on the



X-axis (the upper boundary values are shown as tick marks). For the plot in (a), these bins collect all superfamilies that fall into the respective size range, relative to the size of the largest superfamily. The Y-axis states what fraction of all superfamilies fall into each bin (size range). This procedure is applied to all superfamilies with at least 10,000 (green triangles), 20,000 (pink squares) and 100,000 (blue diamonds) sequences, respectively. For the plots in (b) and (c), the overall procedure is the same, but the plots are generated by averaging over the cluster and family histograms of all the superfamilies in (a), respectively. Each set of data points was fitted with a power-law distribution (lines). The respective  $R^2$  values on the upper right corner of each plot approximate the goodness of fit (for each line, top to bottom), respectively.



**Figure 6.3.** The family size distribution of the ‘Winged Helix DNA-binding’ domain superfamily.  $DFX_{super}$  identifies 178 families in this superfamily. The family size data points were fitted with an exponential distribution (black line); the shown  $R^2$  value approximates the goodness of fit.

The scale-free size distribution that is found for the DFX functional families identified in the Gene3D domain superfamilies, on average, is further illustrated by the example of the ‘Winged Helix DNA-binding’ (CATH 1.10.10.10) domain superfamily in Figure 6.3. As this semi-logarithmic plot shows, the family size distribution in this superfamily is relatively accurately modelled using an exponential fit. The largest family in this superfamily is characterised in Table 6.2, in the following section.

### 6.1.3 The largest families in the largest superfamilies

Table 6.2 provides information on the  $DFX_{\text{super}}$  families identified in the ten largest Gene3D superfamilies (see Table 6.1), specifically focusing on the largest family in each case. The shown family names were generated using the DFX naming protocol (see Section 3.3.5) and are thus based on the names of all parent proteins, respectively. In cases where at least one parent protein contains more than one domain in a given family (and, potentially, further domains from other families) in a superfamily, this is indicated by the domain numbers (in order of N- to C-terminal appearance) in the family names.

It can be seen in Table 6.2 that the largest identified family seems to be exclusive to the eukaryotic domain (based on the seed sequences underlying the family model) in six of the ten largest superfamilies in Gene3D, respectively. Further, only in a single case is the largest family found to be phylogenetically ubiquitous, also including viral sequences. This family represents a highly promiscuous type of NAD(P)-binding domain that appears in different protein domain architectures and functional contexts; therefore, the family name is misleading. This is true for most of the domain families in this table, particularly for the largest CATH 3.40.50.2300 family, which functions in two-component systems and is associated with 12 different third-level EC numbers. This highly abundant domain family occurs together with the largest CATH 3.40.190.10 family, which is also found in Table 6.2, in glutamate receptor proteins.

**Table 6.2. The families identified in the ten largest Gene3D superfamilies.** The family with the most sequences is shown, respectively, along with the LCA taxa (or domains of life) of these sequences; <sup>1</sup>EC annotations with corresponding high-quality GO annotations only; note that this can include functions mediated by other domains in the parent proteins; \*includes viruses

CATH code	Families	Largest family	Sequences	LCA taxon/taxa	EC3s <sup>1</sup>
3.40.50.300	891	ATP-dependent RNA helicase domain 2 -like	16,964	Eukaryota	4
3.30.160.60	263	Zinc finger protein domain 1, 2 -like	36,903	Eukaryota	0
2.60.40.10	613	Titin domain 1, 2 -like	18,597	Bilateria	4
3.40.50.720	603	Siroheme synthase domain 1, 2 -like	2,039	ubiquitous*	4
1.10.10.10	178	Forkhead box protein G1 domain -like	2,588	Bilateria	1
3.30.70.270	3	Gag-Pol polyprotein domain 1 -like	46,101	Caulimoviri- dae	4
1.20.1250.20	274	Solute carrier family 2, facilitated glucose -like d.	1,809	Bacteria/ Eukaryota	0
3.40.190.10	130	Glutamate receptor domain 2 -like	1,450	Bilateria	0
3.40.50.2300	81	Transcriptional regulatory protein phoP dom. -like	1,528	Bacteria/ Eukaryota	12
1.25.40.10	213	Kinesin light chain domain 1, 2 -like	487	Eukaryota	11

An interesting ‘outlier’ case observed in Table 6.1 and Table 6.2 is the Gag-Pol polyprotein domain family and its superfamily (‘Reverse Transcriptase related’). Gag and Pol are two of the three major proteins encoded by retroviral genomes and have been studied primarily in the context of HIV (Frankel and Young 1998). Gag (Group Antigens) is a polypeptide that is post-translationally cleaved into a range (at least three but up to ten) individual proteins with varying structural and functional roles. One of these cleavage products is a multi-functional enzyme with both reverse transcriptase (RT;

RNA-dependent DNA polymerase) and ribonuclease H activity (the core of the retroviral reproduction machinery). The domains in this family (and superfamily) are therefore associated with up to three distinct EC numbers via their parent proteins. Whilst being among the largest ten superfamilies in Gene3D, CATH 3.30.70.270 exhibits comparatively low sequence and structural diversity (see Table 6.1). Accordingly, the great majority of its member sequences are found in the single, large family shown. The non-viral sequences in the superfamily (mostly from DNA polymerase IV proteins) contain motifs similar to those found in the RT domains. DNA polymerase IV is an error-prone, weakly processive polymerase that is involved in untargeted mutation in bacteria; the latter is deemed to convey increased drug resistance (Goodman 2002).

#### 6.1.4 Comparison with the unsupervised protocol

The main motivation behind developing the supervised family identification protocol was the observation that the degree of correlation between sequence and function conservation differs considerably amongst the known protein domain superfamilies (Addou, Rentzsch et al. 2009). Based on this, it was assumed that taking function annotation data into consideration, in addition to the results of sequence clustering, would lead to significantly better family partitionings for most superfamilies. This section examines to what extent this expectation was met in the first run of the DFX pipeline, by comparing the results of supervised family identification with those that could have been achieved for the same superfamilies using the unsupervised protocol. Note that, owing to the non-exhaustive clustering performed (see Section 6.1), the  $DFX_{\text{unsuper}}$  results can only be approximated.

Since the most comprehensive source of function annotation data to date, the Gene Ontology, is already used in the supervised family identification protocol itself, and since it is inherently difficult to assess the coherence and

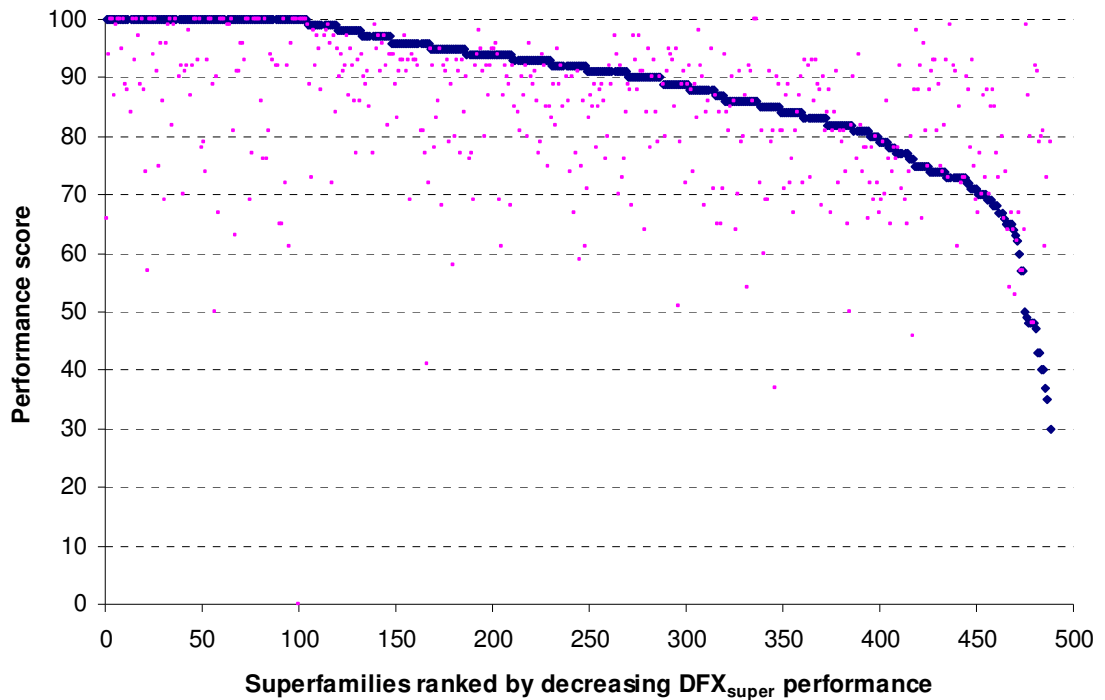
separation of different protein (domain) functions using GO annotations (illustrated by the complexity of cluster assessment as described in Section 5.3.4), the EC system for enzyme annotation was used to compare the results of supervised and unsupervised family identification. For this comparison, the performance measures that were also used to benchmark the unsupervised protocol (see Section 4.2.2) were devised. High-quality EC annotations were compiled based on the evidence codes of the corresponding GO annotations (see Section 3.3.2).

A total of 488 superfamilies containing sequences with at least two distinct, high-quality EC4 annotations were identified in the set of all processed superfamilies. Figure 6.4 shows the percentage of these superfamilies for which the use of the supervised family identification protocol or the unsupervised protocol (in conjunction with ten different settings of the clustering granularity threshold) produces the best observed family partitioning (the highest observed performance score), respectively. Note that this score may be shared by more than one method/setting per superfamily.



**Figure 6.4.** The relative performance of  $DFX_{\text{super}}$  and  $DFX_{\text{unsuper}}$  as measured by enzyme function conservation. 488 Gene3D superfamilies with at least two different high-quality EC4 annotations were processed with both protocols, using ten different clustering granularity settings for  $DFX_{\text{unsuper}}$ . The Y-axis shows the fraction of superfamilies for which each method produces the best observed performance score, respectively. The performance scores were derived using the EC4 annotations for each superfamily in conjunction with the combined performance measure introduced in Section 4.2.2. This measures both specificity (purity) and sensitivity (overdivision, or the lack of it).

Figure 6.5 contrasts the performance scores attained for all 488 superfamilies when using either the supervised protocol or the unsupervised protocol with a generic threshold setting of  $10^{-40}$ , as derived based on the SFLD gold standard dataset introduced in Chapter 4. A very similar picture can be expected when using a threshold of  $10^{-30}$ , the best setting identified for this particular dataset (see Figure 6.4).



**Figure 6.5.** The relative performance of  $DFX_{super}$  and  $DFX_{unsuper}$  with a generic granularity setting of  $10^{-40}$ . This shows the individual performance scores attained by both methods for the dataset described in conjunction with Figure 6.4. The generic granularity setting was determined earlier, as the putatively optimal setting (see Section 4.3.1). The performance scores were derived using the EC4 annotations for each superfamily in conjunction with the combined performance measure introduced in Section 4.2.2). This measures both specificity (purity) and sensitivity (overdivision, or the lack of it).

While the supervised family identification protocol generally outperforms the unsupervised protocol (see Figure 6.4), independent of the generic clustering granularity setting that is used (see Figure 6.5), the average performance and corresponding standard deviation values as listed in Table 6.3 indicate that the margin between the best-performing and all lower-ranking methods or settings is usually narrow. For the different granularity settings explored in conjunction with the unsupervised protocol this is true within the E-value range  $10^{-80}$  to  $10^{-20}$ .

For the unsupervised family identification protocol, Table 6.3 further supports the use of a generic granularity setting in the same range as suggested from training on the SFLD dataset (see Section 4.3.1). The on average best setting is  $10^{-30}$ , closely followed by the currently used, SFLD-derived setting of  $10^{-40}$ .

**Table 6.3. The average performance of DFX<sub>super</sub> and DFX<sub>unsuper</sub>.** 488 Gene3D superfamilies with at least two different high-quality EC4 annotations were processed with both protocols, using ten different clustering granularity settings for DFX<sub>unsuper</sub>. The performance scores were derived using the EC4 annotations for each superfamily in conjunction with the combined performance measure introduced in Section 4.2.2. This measures both specificity (purity) and sensitivity (overdivision, or the lack of it); \*standard deviation.

Protocol / setting	DFX <sub>super</sub>	10 <sup>-80</sup>	10 <sup>-70</sup>	10 <sup>-60</sup>	10 <sup>-50</sup>	10 <sup>-40</sup>	10 <sup>-30</sup>	10 <sup>-20</sup>	10 <sup>-10</sup>	10 <sup>-05</sup>	100
Performance (avg.)	<b>88.44</b>	80.61	81.34	82.24	83.14	84.15	<b>84.67</b>	82.35	73.76	61.89	37.72
Performance (SD*)	12.38	12.39	12.34	12.21	12.15	12.50	13.75	16.86	21.73	24.63	11.86

The results shown in Table 6.4 suggest that the unsupervised protocol is practically on a par with the supervised protocol in the case of the five Gene3D 9.2 superfamilies that correspond to the five SFLD enzyme superfamilies introduced in Section 4.2.1.1. However, it is important to put these values into context with the total number of families identified, respectively. The superfamilies under analysis here all exclusively contain domains from enzymatically active proteins. Even if a certain amount of non-enzymatic parent proteins (only annotated with GO annotations) were assumed, or a number of enzyme functions (EC numbers) that are not assigned to any sequence in the respective superfamily with high confidence (have no corresponding high-quality GO annotation), the family numbers produced by the unsupervised protocol clearly indicate an overdivision of the superfamily in all cases but the Enolase one. In other words, these families cannot represent functional families.



**Table 6.4. The  $DFX_{super}$  and  $DFX_{unsuper}$  partitionings of five functionally diverse enzyme domain superfamilies.** The five Gene3D superfamilies processed correspond to the respective SFLD superfamilies introduced in Section 4.2.1.1. The number of produced families for both protocols and the number of different EC3/4 annotations associated with each superfamily are shown for comparative purposes; only EC annotations with corresponding high-quality GO annotations are counted.

CATH code	SFLD name	$DFX_{super}$ performance	$DFX_{unsuper}$ performance	$DFX_{super}$ families	$DFX_{unsuper}$ families	EC4(3)s
3.20.20.140	Amidohydrolase	94	94	44	114	29 (14)
3.90.226.10	Crotonase	93	92	65	115	18 (11)
3.20.20.120	Enolase	94	96	20	29	09 (05)
3.40.50.1000	Haloacid dehalogenase	90	88	85	326	46 (15)
3.10.180.10	Vicinal oxygen chelate	85	81	23	83	08 (06)

Based on the considerations in the preceding paragraph, two important questions arise. First, why are the domain family numbers in Table 6.4 so high compared with the numbers of different EC numbers associated with each superfamily? This general question refers to both protocols (in fact, to all automatic family identification methods discussed in this thesis), if to different extents. Second, why are the performance values produced by the unsupervised protocol so high, despite the fact that the integrated performance measure used (see Section 4.2.2) penalises the division of functional classes (here: EC4s) across more than one family, respectively (in the same way that it penalises the mixing of classes). This second question is addressed in the following.

### 6.1.5 Fairness of the comparison

In order to understand why the unsupervised protocol achieves high performance scores despite its apparent overdivision of functional classes (lack of sensitivity) one has to investigate the individual performance measures used. These have been introduced in Section 4.2.2: purity, edit distance and VI distance. While purity is a specificity measure, the other two measure sensitivity. Purity is a simple concept, has a fixed value range, and is not involved in penalising overdivision. Therefore, the two sensitivity measures have to be analysed further. Both edit and VI distance have a minimum (optimal) value of zero but, unlike purity, no fixed upper bound. Instead, the maximum (worst) value depends on the size of the clustered dataset (number of sequences) and the number of different functional classes (e.g., EC4s) that are defined, respectively. In principle, this lack of normalisation is not a problem when comparing the edit and VI distance values for different clustering protocols (here: family identification protocols) on the same dataset (here: a given superfamily).

The (combined) performance values as stated in Chapter 4 and in the present chapter are derived by averaging over the three base measures (the purity value is doubled in this, to keep the balance of specificity and sensitivity), respectively. However, before this averaging can take place, both the edit and VI distance values have to be normalised to the same range as the purity value (0-100%). This normalisation is done based on the *initial* value, respectively, that is, the value that is observed when all sequences are put in a single cluster (see Section 4.2.2). Since both these initial values are dataset-dependent (see above), it follows that, through normalisation, the relative differences in the *observed* values for edit and VI distance between different algorithms (here:  $DFX_{\text{super}}$  and  $DFX_{\text{unsuper}}$ ) become less prominent (influential on the combined performance score) with increasing superfamily complexity (size and number of functional classes). In other words, purity (specificity) gains a higher weight

in the combined performance score than edit and VI distance (sensitivity). Therefore, the overdivision of functional classes is not penalised as much as mixing them anymore.

The example calculations for the Amidohydrolase domain superfamily (CATH 3.20.20.140) in Table 6.5 illustrate the above-described effect. As can be seen when comparing the top and bottom parts of the table, a change in the initial edit distance value by one order of magnitude (corresponding to, for example, the difference between a large, diverse and a small, less diverse superfamily) has a considerable impact on the overall performance score attained. The difference in performance between the two algorithms not only changes in sign but is also made 20 times more prominent, based on comparing the percentage difference in the performance scores in both cases (see Table 6.5, top vs. bottom). The example would work equally well for an analogous change in the initial VI distance value.

**Table 6.5. The impact of edit distance normalisation on the calculation of overall family identification performance.** This uses the performance scores attained for the Amidohydrolase domain superfamily to illustrate the impact of a change in the absolute value of initial edit (or VI) distance, which is dataset-dependent, on these scores. All measures (column headers) are described in Section 4.2.2.

Partitioning	Purity	Edit distance	E. d. % initial	VI distance	VI d. % initial	Performance
initial	100	<b>1,324</b>	100	5.05	100	50
DFX <sub>super</sub>	91	18	<b>1.36</b>	0.29	5.74	<b>93.72</b>
DFX <sub>unsuper</sub>	97	57	<b>4.31</b>	0.68	13.47	<b>94.06</b>
initial	100	<b>132.40</b>	100	5.05	100	50
DFX <sub>super</sub>	91	18	<b>13.60</b>	0.29	5.74	<b>90.67</b>
DFX <sub>unsuper</sub>	97	57	<b>43.05</b>	0.68	13.47	<b>84.37</b>

In order to assess the overall impact of the above-outlined bias in the performance measures that were used to compare different family partitionings, modified measures can be devised. In particular, the normalisation procedure for the edit and VI distance values (to derive percentages) can be changed to follow a ‘best-value’ oriented strategy, instead of the original ‘worst-value’ oriented one. In other words, instead of using the initial value as the 100% mark and deriving all other values as percentages accordingly (see Table 6.5), the *best* (i.e., lowest) value that was attained by any of the compared algorithms can be used as the 0% mark and all other values derived accordingly. The advantage of such modified, *relative* edit and VI distance measures would be their independence from the initial values, that is, superfamily size and functional diversity.

Table 6.6 shows how using a modified normalisation procedure as described above for both edit and VI distance would impact the performance calculation and comparison in the example introduced in Table 6.5. Arguably, the performance values attained when using the modified procedure (see Table 6.6, bottom) reflect much more the good balance between specificity and sensitivity generally achieved by  $DFX_{super}$  (and the frequent overdivision by  $DFX_{unsuper}$ ) that is further demonstrated in Table 6.7.

In Table 6.7, two simple alternative measures for specificity and sensitivity were used to reassess the partitionings derived for the same five SFLD-related Gene3D superfamilies as listed in Table 6.4: the average number of different EC4s per identified family (specificity) and the average number of identified families per EC4 (sensitivity). As was already indicated by the number of families produced for each superfamily in Table 6.4, respectively, it can be seen that the  $DFX_{unsuper}$  families exhibit considerably higher overdivision of the functional classes than those produced by  $DFX_{super}$ . While this lower sensitivity is attended by a slightly higher specificity, as can be expected, the altered performance scores attained by using the modified normalisation procedure reflect well the overall better balance in the results of the supervised method.

**Table 6.6. The effect of a modified normalisation procedure on the calculation of overall family identification performance.** This comparison extends on that in Table 6.5. It illustrates how the difference in the two methods' performance scores for the Amidohydrolase domain superfamily changes sign and becomes much more prominent when the respective edit (and VI) distance values are normalised by the best observed values instead of the worst (initial) ones, respectively. All measures (column headers) are described in Section 4.2.2.

Partitioning	Purity	Edit distance	E. d. % initial	VI distance	VI d. % initial	Performance
initial	100	<b>1,324</b>	100	5.05	100	50
DFX <sub>super</sub>	91	18	<b>1.36</b>	0.29	<b>5.74</b>	<b>93.72</b>
DFX <sub>unsuper</sub>	97	57	<b>4.31</b>	0.68	<b>13.47</b>	<b>94.06</b>
Partitioning	Purity	Edit distance	E. d. % best	VI distance	VI d. % best	Performance
initial	100	<b>1,324</b>	1.36	5.05	5.74	51.78
DFX <sub>super</sub>	91	18	<b>100</b>	0.29	<b>100</b>	<b>95.50</b>
DFX <sub>unsuper</sub>	97	57	<b>31.58</b>	0.68	<b>42.65</b>	<b>67.06</b>

**Table 6.7. Correspondence of alternative performance measures with the performance scores attained when using the modified normalisation procedure.** The five enzyme domain superfamilies analysed in Table 6.4 were reassessed using two simple measures for specificity and sensitivity (see main text). The relative performance scores attained when using the original measure in conjunction with the modified normalisation procedure are shown for comparative purposes.

CATH code	DFX <sub>super</sub> EC4s/fam	DFX <sub>super</sub> fam's/EC4	DFX <sub>unsuper</sub> EC4s/fam.	DFX <sub>unsuper</sub> fam's/EC4	DFX <sub>super</sub> rel. perf.	DFX <sub>unsuper</sub> rel. perf.
3.20.20.140	1.15	1.52	1.05	3.12	94	67
3.90.226.10	1.15	2.07	1.14	2.73	74	67
3.20.20.120	1.11	1.11	1.08	1.44	95	64
3.40.50.1000	1.18	1.41	1.01	3.85	95	68
3.10.180.10	1.15	1.88	1.00	5.25	87	66

Figure 6.6 underlines that there is no strong correlation of either method's performance with superfamily size (a), sequence diversity (b), or functional diversity (c); the values for DFX<sub>unsuper</sub> and DFX<sub>super</sub> are shown in pink and blue, respectively. Note that the reason for the pattern apparent in the performance scores in Figure 6.6c is that the superfamilies were ranked by DFX<sub>super</sub> performance score prior to ranking them by functional diversity. Since several superfamilies are associated with the same number of distinct EC3s (functional diversity measure), the performance scores for each of these 'plateaus' in the functional diversity curve in Figure 6.6c are sorted. The measures in Figure 6.6a and Figure 6.6c have wider ranges (right Y-axis, respectively) and therefore do not show plateaus; no similar pattern in the performance scores is thus observed.

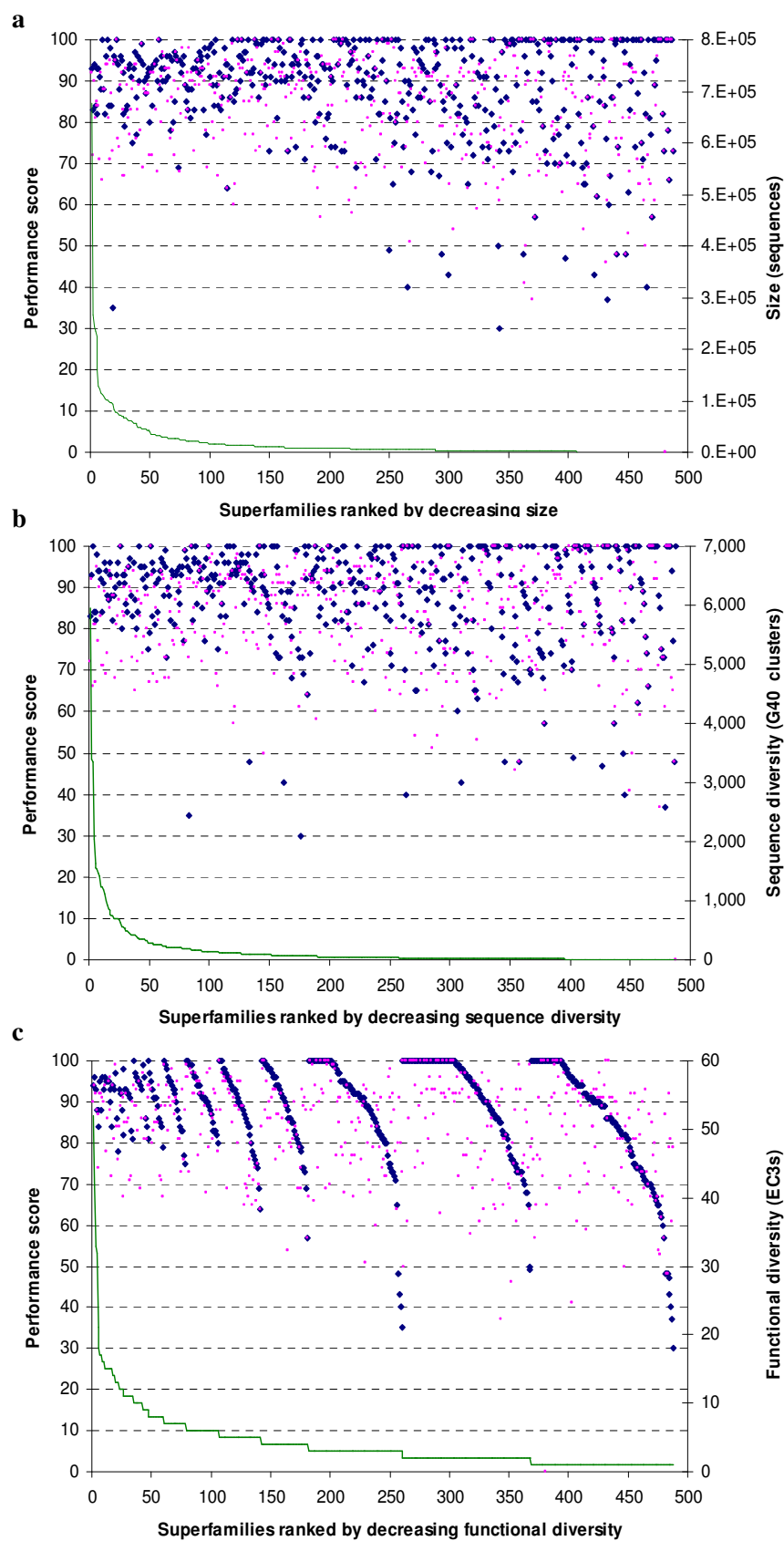


Figure 6.6. The impact of superfamily size, sequence diversity and functional diversity on the performance of the DFX family identification protocols. The values for  $DFX_{unsuper}$  and  $DFX_{super}$  are shown in pink and blue, respectively.



For the two measures assessed in Figure 6.6a and Figure 6.6b, superfamily size and sequence diversity,  $DFX_{\text{super}}$  yields the highest (close to 100%) performance scores (blue diamonds) at low and medium levels, that is, for those superfamilies ranking among the lower  $\sim 75\%$ . This can be expected given the exceptional character of the upper  $\sim 25\%$  of large and  $\sim 5\%$  of very large superfamilies. As discussed above, these exhibit considerable variability in sequence, function and annotation quality, and therefore represent (more) challenging targets for family identification.

When assessing the dependence of family identification performance on superfamily functional diversity (see Figure 6.6c), a clearer trend becomes apparent. As can be expected for the above-stated reasons, it becomes much more difficult to attain high performance scores above a level of about five different EC3s that are associated with a superfamily (left from the left-most 100% plateau in the performance scores of both methods). Further, neither method yields a perfect performance score (100%) for a superfamily with more than ten different EC3s. Note that especially the analysis of Figure 6.6c must be performed with the non-ideal measure (EC annotations) in mind (see also Section 6.2.1).

## 6.2 Discussion

The overall methodology behind the comparisons presented above is first discussed in the following, before the significance of the observed superfamily and family size distributions is addressed in particular.

### 6.2.1 Notes on performance assessment

Notably, the comparisons made above are not benchmarks between the two family identification protocols; they cannot be, since one uses annotation data (an inherent advantage) and the other does not. Rather, they serve to

demonstrate the benefit of using available high-quality function annotation data when identifying functional protein (domain) families, instead of ignoring it. Specifically, how well the supervised protocol can ‘translate’ between the GO annotations of individual proteins (protein domains) and their actual level of functional similarity was measured. This necessary translation, in the case of GO, is the downside of using the annotation data, and makes supervised family identification a non-trivial task.

Note that the comparison strategy followed in this chapter, namely the use of four-level EC annotations, is not ideal. The DFX pipeline follows a domain family concept that focuses on domain function, not whole-protein function (see Section 0). This allows for putting domain sequences with conserved function into the same family even in cases where the respective function (EC annotation) of the parent proteins differs substantially. In particular,  $DFX_{\text{super}}$  addresses this aim using several heuristics. This is not taken into account when using the whole-protein EC annotations for assessing the family partitioning performance of the two protocols. In brief, multi-domain proteins with multiple enzyme functions can impact the functional purity measurements, and a mixing of different whole-protein functions (at the highly specific EC4 level) is penalised. However, these issues are not thought to render the presented results less relevant overall, since the results of both  $DFX_{\text{super}}$  and  $DFX_{\text{unsuper}}$  are assessed in exactly the same manner. The *relative* performance signal is therefore not disturbed, only the absolute one. Yet, it has to be kept in mind that there is a fundamental difference between domain-based function annotations (e.g., from the SFLD) and protein-based EC annotations. This difference is highly relevant to both, family identification and benchmarking.

In contrast to the DFX unsupervised family identification protocol (see Chapter 4), a true benchmarking of the supervised protocol, against competing methods, does not seem feasible at this point. This is because no

existing method the author knows of is similar enough in scope and aim, that is, tries to perform the above-outlined ‘translation’ task. Should more comprehensive domain family gold standard datasets become available in the future, such as the SFLD dataset that was used to benchmark the unsupervised protocol (see Section 4.2.1.1), and should these follow a domain family concept that is similar to the DFX one, as well as include non-enzyme sequences, further comparisons *between* the DFX protocols (and against any novel, competing methods) will become possible. It is further conceivable that a coarse benchmark of the core term set strategy (the derivation of domain-specific GO annotations) of DFX<sub>super</sub>, as described in Section 5.3.2, could be devised based on the InterPro2GO mapping (see Section 5.1.2), as this has been done before.

### 6.2.2 The significance of family size distributions

With some caution, especially keeping in mind sequencing bias, the observed scale-free size distributions for families and superfamilies can be regarded as evolutionary fact. As a general rule, the most evolvable, most ancient and biologically most important folds, superfamilies and (based on the present results) families can be expected at the upper end of a scale-free size distribution in each category. While structural stability and evolvability can be expected to be the most important factors for abundance (evolutionary success) on the fold level (Mirny and Shakhnovich 1999; Bloom, Labthavikul et al. 2006; Allen and Dunaway-Mariano 2009; Rorick and Wagner 2011), functional importance most likely becomes a decisive factor when ‘zooming in’ to the superfamily and family levels. Examples are the many evolutionary ancient, primary cellular processes that involve the binding of nucleic acids: five of the ten largest superfamilies in Gene3D (see Table 6.1) comprise domain sequences that are involved in such processes. In the future, it will be possible to study this relationship between functional importance and

abundance of individual domain types in much more detail, on the family level (see, for example, Table 6.2).

## Chapter 7. The DFX pipeline: summary and future work

DFX is a pipeline for the identification, storage and assignment of families within protein domain superfamilies. Its overall design, concept and implementation are discussed in Chapter 3. DFX embeds the large-scale sequence clustering method GeMMA, which is presented in Chapter 2, and two alternative protocols for family identification based on the clustering results, as discussed in Chapter 4 and Chapter 5, respectively. There, the performance of either protocol is analysed primarily qualitatively. A corresponding quantitative analysis is presented in Chapter 6, where both protocols are compared based on the results of the first large-scale run of DFX.

### 7.1 Summary of work

The foundation for the DFX pipeline was laid with the development of the GeMMA sequence clustering method; this had initially been used in isolation to derive protein domain families (Lee, Rentzsch et al. 2010). GeMMA is a highly modular and thus flexible implementation of agglomerative hierarchical sequence clustering that uses profile-profile comparisons for high sensitivity and several heuristics for speed-up. For the latter goal, it was specifically designed to run in HPC environments. These characteristics are crucial for making DFX cope with the growing amount of protein sequence data, which is an even more pressing problem when classifying protein domains: the large number of multi-domain proteins implies that there will always be more domain than protein sequences.

DFX uses a hybrid approach to establish families. Where high confidence information in the form of protein function annotation data is available, this information is not available, the pipeline falls back to an unsupervised method.

It can be argued that this decision reflects the way in which a human curator would approach each individual superfamily much more than, for example, the universal use of a ‘functionally blind’, unsupervised protocol. As more and more reliable biological information on individual proteins accumulates over time, it will be possible to identify families in a way that takes into account this knowledge in more and more superfamilies.

The unsupervised family identification protocol ( $DFX_{\text{unsuper}}$ ) serves to identify families based on the GeMMA clustering results alone, in cases where a domain superfamily is not associated with high-quality protein annotation data. It is simpler and scales better than related established protocols, whilst reaching comparable or better performance. Unlike some of the former, however, it so far depends on a one-off training step (and is therefore not an *ab-initio* protocol in the strict sense). The implementation of  $DFX_{\text{unsuper}}$ , the training procedure and the application of the protocol on both a small scale (a gold standard set of superfamilies) and a large scale (a subset of Pfam) are discussed in detail in Chapter 4.

The supervised family identification protocol ( $DFX_{\text{super}}$ ) was developed to be able to exploit the growing body of high-quality protein annotation data in domain family identification. About 75% of all domain superfamilies in Gene3D are associated with such data and can therefore be processed using  $DFX_{\text{super}}$ . The protocol uses Gene Ontology protein annotations to analyse the domain sequence clusters produced by GeMMA and subsequently selects a subset of clusters with putative family character.  $DFX_{\text{super}}$  therefore essentially groups domain sequences based on similarity in both sequence and function. As an important part of this, the protocol derives domain-specific annotations in a heuristic manner. While a few methods exist that group whole-protein enzyme sequences by using the results of sequence clustering and annotation data (see Section 5.1.1), the fact that these do not work on the domain level and use the EC annotation system makes them very different in

scope.  $DFX_{\text{super}}$  therefore addresses a very specific, somewhat novel problem. The latter is discussed in detail in Chapter 5, which includes an in-depth qualitative analysis on the basis of established biological knowledge.

Using the DFX pipeline, a detailed picture of the known and uncharacterised parts of sequence and function space within all Gene3D domain superfamilies (currently more than 2,500) can be generated, in a fully-automatic and consistent manner. The generated libraries of domain family models (together more than 25,000) can then be used to assign the majority of sequences in these superfamilies to one of the identified families (see Section 6.1). Further, the DFX model libraries allow for whole-protein function prediction, using a newly introduced prediction framework that integrates the family information for all domains in the respective target proteins (see Section 3.3.7).

## 7.2 Current usage and data availability

The DFX pipeline was first run in a large-scale, fully-automatic manner in 2010. This has produced more than 25,000 domain families for  $\sim 1,900$  superfamilies in Gene3D that were amenable to processing with  $DFX_{\text{super}}$  (associated with functional information). The results of this are analysed in a quantitative manner in Chapter 6.

As of October 2011, several in-house projects have used and are using the generated family data. These projects aim (i) to study the evolution of domain function through the identification of function-determining residues, (ii) to select putatively promising target proteins for structural genomics, (iii) to analyse the distribution of protein functions in metagenomes and (iv) to study the structural and functional evolution of ancient protein domain superfamilies (Dessailly, Redfern et al. 2010).

### 7.2.1 The Gene3D family level

Based on the data generated in the first DFX run, a functional family level in Gene3D has recently been introduced and made available to the research community via the Gene3D website<sup>17</sup>. Web-services for on-demand domain family assignment and (potentially) domain-based protein function prediction will be added soon. For future maintenance, the pipeline was implemented in a highly flexible, configurable and user-oriented manner. Full reruns are planned on an annual basis, to take advantage of newly added protein sequence and function data. Intermittent, incremental updates of the families with each release of Gene3D are planned additionally. This hybrid updating strategy is made possible by the implementation of DFX as a model-based system, mimicking the strategy of established family resources such as Pfam, SUPERFAMILY and Gene3D itself.

### 7.2.2 DFX in protein function prediction

A preliminary module for domain-based protein function prediction (see Section 3.3.7) was recently added to the DFX pipeline. Each DFX domain family is associated with a set of GO function annotations. When all domains in a given (multi-domain) protein are assigned to their Gene3D superfamilies and, subsequently, to their DFX family, this module combines the functional information associated with each family and returns a range of probabilistic GO term assignments. It is anticipated that this generic framework, that is, to elucidate the composite function(s) of whole proteins by combining functional information on each of their domains, could become a powerful tool. To this end, the DFX function prediction module should be further improved.

---

<sup>17</sup> <http://gene3d.biochem.ucl.ac.uk/Gene3D/>



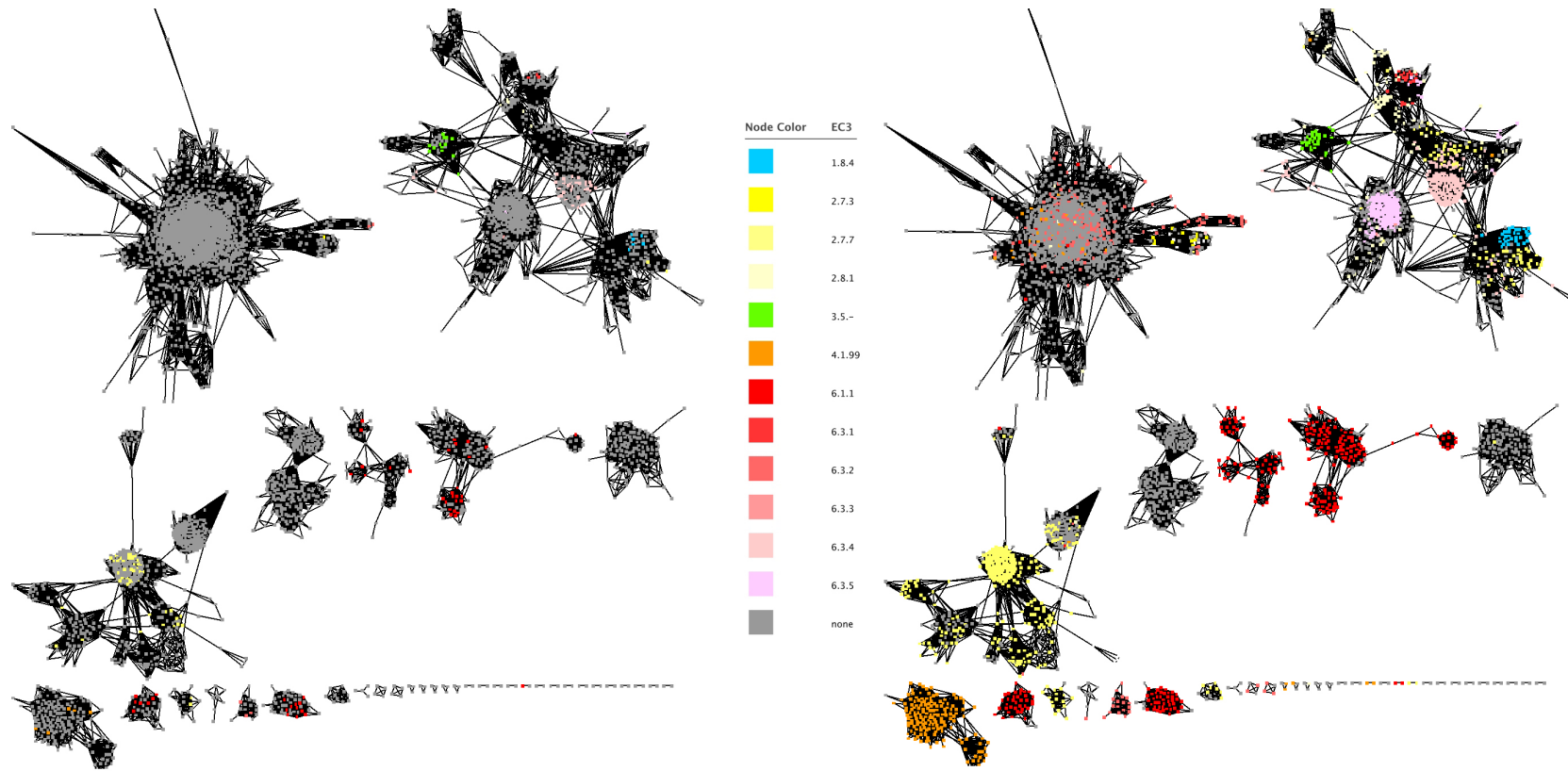
One important point of possible improvement of the function prediction module is the integration of the information coming from individual domains. For example, additional information on domain order and size (the fraction of a protein that is covered by a given domain) could be used in up- or down-weighting the impact of each domain in the function prediction procedure. Moreover, the occurrence count of individual domain types within the same protein should be considered.

In the long-term, it is conceivable that the occurrence (or non-occurrence) of highly conserved residue patterns is used to characterise the DFX domain families as either (primarily) catalytic or (primarily) binding domains. The same applies for the occurrence of repetitive and low-complexity regions in transmembrane domains. This information could then also be used in protein function prediction, where catalytic domains may play a more important role than binding and transmembrane domains (this corresponds to the relationship between the GO ‘catalytic activity’ and ‘binding’ branches, for example, as described in Section 5.2.1).

As a first large-scale assessment of the potential that may lie in a domain-based approach to protein function prediction, predictions were made and submitted for the about 50,000 target protein sequences of the CAFA (Critical Assessment of Function Annotations) 2011 function prediction challenge. While the detailed results of this challenge have not yet been published in a manner that compares between the competing methods, a preliminary analysis of the DFX results shows that there is much room for improvement. In particular, DFX appears to be in the medium performance range, among other methods that did not beat the performance of the best baseline method used in CAFA, GOTcha (Martin, Berriman et al. 2004). It must be stressed in this context that the primary aim in developing DFX was not function prediction but establishing functional groups. Particularly the use

of GO annotations makes these two distinctly different problems (see also Section 5.1.2).

It will further be interesting to see the exact rates of family assignment coverage (how many domain sequences can be safely assigned to one of the established families) before and after scanning the whole Gene3D superfamilies with their respective DFX model libraries. Preliminary studies on the HUP superfamily, prior to the switch from using EC numbers to GO annotations in the development of DFX, have shown promising results. An approximately three-fold increase in enzyme (domain) annotation coverage was reached, going from an initial annotation level of  $\sim 20\%$  to  $\sim 75\%$  of all sequences in the superfamily. Figure 7.1 illustrates this increase based on three-digit EC number functional families, using a domain sequence similarity network. The network nodes represent HUP domain sequence clusters with 40% maximum inter-cluster sequence identity. The edges are based on a pairwise sequence similarity matrix of the cluster representative sequences (one sequence per cluster) that includes all BLAST E-values of  $10^{-5}$  or lower. Based on this, the network was laid automatically using the Organic layout option in Cytoscape.



**Figure 7.1.** The coverage of the HUP domain superfamily with EC functional family assignments before and after scanning with family-specific models. Each node in the domain sequence similarity networks shown corresponds to a CD-HIT 40% sequence identity cluster representative sequence. All nodes are coloured according to the functional (EC3) family assignment of the respective representative sequence. After establishing families with a DFX<sub>super</sub>-like protocol, only the sequences in the seed family clusters are annotated (left), reflecting the available high-quality EC annotations. After scanning all sequences with the model library, the annotation coverage increases (right). Edges represent pairwise BLAST E-values of 10<sup>-5</sup> or lower; Cytoscape with Organic layout was used.

The sequence similarity networks for the novel, GO-based DFX<sub>super</sub> domain families, which specifically focus on *domain* function, can be expected to differ remarkably from those obtained for families based on EC numbers and whole-protein function. In many cases, two domains whose parent proteins have different EC numbers (functions) will be assigned to the same domain family by DFX<sub>super</sub>, based on a putatively conserved domain function. This will translate into sequence similarity networks with fewer families (colours) but higher information content with regards to domain function instead of whole-protein function. Further, for superfamilies that contain domains from proteins with non-enzymatic functions, an increased coverage can be expected when using GO-based functional families. Figure 7.1 exemplifies the great potential of the sequence similarity network paradigm when trying to shed light on the unexplored sequence space (families) within large domain superfamilies.

### 7.2.3 DFX in the detailed study of superfamilies

While quantitative assessments such as those made in Chapter 6 can reveal interesting general trends about domain superfamilies and families, using the family data in detailed studies on specific superfamilies can lead to more immediate, intuitive and thorough insights into the evolution of proteins. Prime examples of such endeavours are the studies presented in Koonin and Tatusov (1994); Babbitt, Hasson et al. (1996); Aravind, Leipe et al. (1998); Aravind, Anantharaman et al. (2002); Vogel, Teichmann et al. (2003); Burroughs, Allen et al. (2006); Garza-Garcia, Harris et al. (2009); Dessailly, Redfern et al. (2010). A non-exhaustive list of protein and domain superfamily studies from the last two decades is found in Appendix B. Importantly, in the context of the present work, such studies can use the comprehensive information on known functional families provided by the developed pipeline as a starting and orientation point.

From the already conducted studies mentioned above, a generic framework to assess, organise and analyse the existing knowledge (gaps) concerning the sequence, structure and function plasticity of individual domain superfamilies can be drafted. Importantly, this is not restricted to the pre-defined superfamilies in Gene3D. Such a protocol could work as follows.

- i) For a given superfamily, retrieve the latest sequence data from Gene3D and CATH. Alternatively, use a manually curated or automatically generated sequence signature(s) of the superfamily for exhaustive searches against extant protein sequence databases, and subsequently assign the respective domains using the Gene3D web services (Yeats, Lees et al. 2011). Corresponding signature resources and tools are, for example, PROSITE, PRINTS (Attwood, Bradley et al. 2003) and MEME (Bailey, Boden et al. 2009).
- ii) Retrieve the latest protein function annotation data from UniProt-GOA (and, potentially, further sources). Filter and map the sequence data to the annotation data. This can be done by feeding the two types of data into the DFX pipeline's data preparation module.
- iii) Run the DFX pipeline to retrieve a set of functional families for the superfamily, based on current knowledge. Potentially novel families can be automatically identified by scanning all sequences in the superfamily against the DFX family model library and pooling those that hit a certain model best but not with a score that meets its exclusion threshold (see Section 3.3.6). Shared patterns of this to specific models can be an even stronger indicator of 'betweenness', that is, a novel family found between two known families in sequence space.

- iv) Analyse any putatively novel, functionally uncharacterised families by their alignments (e.g., identify conserved residues) and any solved or modelled associated protein structures. Modules that help to identify the most ‘promising’ candidates for novel families quickly may be implemented for DFX in the future.
  
- v) Generate a sequence, or family, similarity network of the superfamily, based on a matrix of pairwise sequence or profile similarities generated with tools such as PSI-BLAST or HHSearch. In both cases, this matrix can be filtered for very close relationships beforehand, to make the network less ‘cluttered’ and more tractable with tools such as Cytoscape (Cline, Smoot et al. 2007) or VisAnt (Hu, Hung et al. 2009). In conjunction with the network, the family tree generated by GeMMA and optionally constructed phylogenetic trees can be analysed using tools such as iTOL (Letunic and Bork 2011) or Archaeopteryx (Han and Zmasek 2009).
  
- vi) Put all the compiled information into context with information from the literature and public databases, using data mining tools such as iHOP (Hoffmann and Valencia 2004) or BioGraph (Liekens, De Knijf et al. 2011).

### **7.3 Recent improvements and future work**

The recently introduced chaining concept and its future use in DFX, as well as potential major and minor changes to the pipeline are discussed in the following sections.

#### **7.3.1 The chaining concept and a potential two-layer system**

The most recent addition to DFX is the chaining concept and detection algorithm as implemented in DFX<sub>super</sub> (see Sections 5.2.3 and 5.3.5). With this

having been implemented, the DFX pipeline will soon be rerun in full and an improvement in performance is expected. Specifically, this refers to the compliance of the identified families with the domain family concept introduced in Section 0, putting the focus on domain function instead of protein function. This rerun will also include the processing of all superfamilies that are not associated with functional information using the  $DFX_{\text{unsuper}}$  protocol. In the long-term, it is anticipated that the  $DFX_{\text{super}}$  chaining concept will be used to produce two layers of domain families, corresponding to two different levels of sequence and function conservation. This can already be achieved, by simply turning chaining detection on and off, but would require further modifications to the pipeline as a whole; for example, the introduction of a second layer in  $DFX_{\text{unsuper}}$ , for consistency reasons.

A multi-layer approach can be beneficial and is followed by related resources such as the Conserved Domain Database (see Section 3.1.1). This is owing to the widely varying patterns of sequence and function conservation in domain superfamilies, and the different usage scenarios for domain families. For example, while a very fine-grained family layer would be preferred in domain-based protein function prediction, a coarser layer is more suitable when the functional and structural plasticity of domain families and superfamilies is studied.

In the context of protein function prediction, a certain amount of family overdivision (having several families that represent the same or very closely related functions; decreased sensitivity) is not problematic and can even be beneficial. Using several different, smaller models for the same function may lead to increased annotation coverage over using a single, large model; a similar strategy is followed in superfamily assignment by Gene3D and SUPERFAMILY.

When studying the evolution of domain function in the context of families and superfamilies, the focus in producing the families lies on sensitivity (coverage). Here, a certain degree of functional impurity (decreased specificity) is not problematic and can even be beneficial. For example, shifts in functional specificity based on individual, specificity-determining residues can only be identified (with sufficient confidence) in family sequence alignments when these are large enough.

### 7.3.2 Potential replacement of GeMMA

A more radical change to the DFX pipeline would be the total replacement of the GeMMA method by a better-performing clustering approach. In principle, such a replacement is straightforward, given the shared two-component architecture of all family identification methods (see Section 2.1.3). In brief, this refers to the combination of any type of clustering method with either a supervised or an unsupervised clustering evaluation strategy. GeMMA was especially designed for large-scale clustering tasks in HPC environments. To speed up the clustering process, it also implements different heuristics. A method suitable to replace GeMMA would have to be able to cluster the same amounts of data with the same or higher speed, but with higher accuracy. Such a method could so far not be identified. However, there exist promising candidates among the graph-based clustering methods (see Section 2.1.2.3). A general advantage of graph-based methods when compared with GeMMA could be the combination of speed and sensitivity. With regards to speed, they are similar to traditional hierarchical clustering methods (see Section 2.1.2.1) in the sense that an all-by-all similarity matrix of data points must only be calculated once, initially. In terms of sensitivity, however, graph-based methods can be expected to be superior to these traditional methods. As they work on a network of pairwise relationships, they inherently take groupwise relationships between the data points (sequences) into account. Further, advances have been made in the implementation of graph-based and other



clustering algorithms for use in HPC environments (Olman, Mao et al. 2009; Changjun 2010; Bustamam, Burrage et al. 2011; Miele, Penel et al. 2011; Yang, Zola et al. 2011). As the performance of individual clustering methods can only be assessed in the context of a specific aim, that is, a specific usage of the obtained partitionings, other methods would have to be compared with GeMMA based on the respective family partitionings produced by DFX before they could be considered to replace it.

### 7.3.3 Potential replacement of the unsupervised protocol

A second DFX module that may be replaced entirely in the long-term is the unsupervised family identification protocol,  $DFX_{\text{unsuper}}$  (see Chapter 5). This protocol is the embedded successor, or reformulation, of an earlier approach based on the use of generic thresholds with GeMMA in isolation (Lee, Rentzsch et al. 2010). For use in DFX, the sequence clustering and family identification steps have been entirely disentangled. This reflects the composite nature of family identification protocols in general, as outlined above. Based on this, it would be possible to implement *any* type of truly unsupervised, training-free clustering evaluation strategy to replace  $DFX_{\text{unsuper}}$  in the long term.

Different unsupervised clustering evaluation strategies have been successfully implemented in existing *ab-initio* protocols for family identification (Kelil, Wang et al. 2007; Brown 2008; Yang, Zhu et al. 2010). However, these methods are restricted to datasets of small to medium size, owing to the ‘conservative’, poorly-scaling hierarchical clustering strategies they employ. For example, none of these methods runs in HPC environments. Strikingly, if the clustering and clustering evaluation steps in these protocols at this point were uncoupled, as is the case in DFX, they could already be used in DFX, by feeding the GeMMA clustering results into the respective evaluation method. Even without any changes, *ab-initio* methods like CLUSS or SCI-PHY could

readily be used in DFX instead of  $DFX_{\text{unsuper}}$ , for superfamilies of small to medium size. For large superfamilies, however, this is hindered by speed and memory constraints (see also Section 4.2.4.2).

### 7.3.4 Further potential improvements

Further modifications to the individual modules used by the DFX pipeline may be considered in the future. Possible changes with regards to the usage of domain architecture information in family identification are discussed in Section 5.5.3. The same information may be used to improve the naming of domain families (see Section 3.4.3). In addition, potential modifications in the handling and filtering of both domain sequence and protein function annotation data, the two main types of input data for DFX, are considered in Sections 3.4.1 and 5.5.3, respectively.

## 7.4 DFX in the context of other novel methods

Following and now accompanying the sequence data ‘explosion’ (Cochrane, Karsch-Mizrachi et al. 2011; Magrane and Consortium 2011), a substantial and continuous increase in available protein structure data has been observed over the last decade (Berman, Westbrook et al. 2000; Rose, Beran et al. 2011). Driven to a large extent by structural genomics initiatives (Dessailly, Nair et al. 2009), this has remarkably increased the sequence coverage of the two major structure-based domain superfamily resources, SCOP/SUPERFAMILY and CATH/Gene3D (see Section 1.5.2.1). At the same time, sequence-based protein family resources have become more and more enriched with data on protein function, and thus more valuable to researchers (Jaroszewski, Li et al. 2009; Bateman, Coghill et al. 2010; Roberts, Chang et al. 2011). This was made possible by worldwide biochemical research into protein function, and the improved formalisation, curation and distribution of its results, primarily by the Gene Ontology project.

Based on the above observations, it is no coincidence that the domain superfamily resources, SUPERFAMILY and Gene3D, increasingly aim to incorporate functional data, following their protein family relatives. Especially in the case of promiscuous domains that appear in proteins with many different functions, this cannot be done in a specific manner on the superfamily level. A sub-classification of superfamilies into families is therefore necessary. It further poses the challenge of mapping between whole-protein function assignments and the (putative) functions of individual domains. Both the family identification and functional characterisation (mapping) tasks were therefore first approached manually.

SCOP included a family level below the domain superfamily from the beginning (Murzin, Brenner et al. 1995), combining a clustering approach with manual curation (see Section 1.5.2.1). SUPERFAMILY adopted this second layer a decade later (Gough 2006; Wilson, Madera et al. 2007) and, at the same time, started to make function assignments at the superfamily level, using a specifically designed ontology (Vogel, Teichmann et al. 2005). Only very recently, and therefore not discussed in the present work, SUPERFAMILY then started to assign GO terms to both its domain superfamilies and families. This is done in a probabilistic manner, using a newly developed protocol (de Lima Morais, Fang et al. 2011). In conjunction with this, the resource has introduced a trimmed-down version of GO for domain function assignment, dubbed the ‘Structural Domain Functional Ontology’.

CATH and Gene3D have long been incorporating external protein function annotation data (Lee, Grant et al. 2005; Pearl, Todd et al. 2005); however, until very recently both had neither a family level nor a means of associating their domain superfamilies with functional information. The development of the DFX pipeline has now made it possible to solve both problems at once. The conceptual and methodological differences and similarities between the novel SUPERFAMILY and Gene3D approaches to domain function will yet have to

be studied, and the results of such study may well be mutually inspiring. One important difference is, however, immediately obvious. In the case of SCOP/SUPERFAMILY, the identification of domain families has so far been a largely manual effort. In the case of Gene3D, families have been established in a fully-automatic manner, using DFX.

## 7.5 Final remarks

Based on the very recent addition of functional domain families to the SUPERFAMILY and Gene3D databases, as outlined in the above section, it is clear that the concepts of the protein domain family and protein domain function will be a necessary and active area of research for the foreseeable future. Related, highly curated resources that are under active development and will help to sharpen these concepts are the Structure-Function Linkage Database (SFLD; see Section 4.2.1.1), the Conserved Domain Database (CDD) and, of course, Pfam. First and foremost, however, the notion that there is such a thing as ‘conserved domain function’ (see Section 5.1.3) seems to be more and more acknowledged by resources and researchers alike.

The further development of the Gene Ontology annotation system may also impact protein domain research. There is still much room for improvement of the GO (see, for example, Section 5.2.1), and it is conceivable in the long-term that it will incorporate a concept of local (e.g., domain) functionality within proteins; a starting point for this could be the Protein Ontology project (Natale, Arighi et al. 2011). The experimental community is also increasingly striving for large-scale, coordinated and collaborative efforts such as the Enzyme Function Initiative (Gerlt, Allen et al. 2011) and the COMBREX project (Roberts, Chang et al. 2011). This means that the elucidation of protein function, as the last and – from a biologist’s point of view – most important part of the sequence-structure-function triad, may soon enter an era of ‘high throughput’.

On a more specific note, it can be expected that the focus of bioinformatics research in the area of protein function will shift slightly, from single-protein function prediction (which has already reached a high level of sophistication) towards the accurate prediction of groupwise functional relationships between proteins. This refers to both similarities in molecular function and process function, and is particularly relevant to family identification and functional enrichment analyses in the -omics fields (Chagoyen, Carazo et al. 2008; Chagoyen and Pazos 2011; Chitale, Palakodety et al. 2011). The network paradigm will play a more and more important role in capturing and visualising such group-wise relationships (Frickey and Lupas 2004; Cline, Smoot et al. 2007; Hu, Hung et al. 2009); examples for this are sequence similarity networks (Song, Joseph et al. 2008; Atkinson, Morris et al. 2009) and functional linkage networks (Marcotte, Pellegrini et al. 1999; Hu, Hung et al. 2009; Rentzsch and Orengo 2009; Szklarczyk, Franceschini et al. 2011).

The task of translation between the available knowledge on the functions of individual proteins (or domains) and the functional families they form has become a challenge of its own. This is because this knowledge is frequently incomplete (or incompletely captured) and sometimes erroneous (or erroneously captured) (Jones, Brown et al. 2007; Schnoes, Brown et al. 2009). In this context, the success of the Gene Ontology represents both ‘a blessing and a curse’, where the former refers to annotation coverage and the latter to annotation diversity, that is, the highly varying quality of individual annotations and their inconsistent usage by different groups and annotators (Costanzo, Park et al. 2011).

It has repeatedly been argued that the protein domain is the key unit of protein function evolution (Tatusov, Altschul et al. 1994; Storm and Sonnhammer 2003; Koonin 2005; Marchler-Bauer, Anderson et al. 2005; Bashton and Chothia 2007; Song, Sedgewick et al. 2007; Song, Joseph et al. 2008). Therefore, they should be increasingly focussed on studying this

evolution, instead of whole genes or proteins. In the long term, the accumulating knowledge about domains and a more thorough understanding of their specific means of evolution (see Section 1.1.2) may lead to substantial redefinitions and changes in the use of concepts like orthology and paralogy (Song, Joseph et al. 2008).

Already at this point, some researchers differentiate between whole-protein orthology and ‘domain orthology’ (Storm and Sonnhammer 2003; Dessimoz, Cannarozzi et al. 2005). In principal, such neologisms only express what has long been observed in (multi-domain) proteins. The importance of conserved domain functions in catalysis and binding is more and more acknowledged in both bioinformatics analyses (Ibrahim, Eldeeb et al. 2011; Itzhaki 2011; Luo, Pagel et al. 2011; Xie, Jin et al. 2011) and experimental studies (Carducci, Perfetto et al. 2011; Spitzweck, Brankatschk et al. 2011; Tricker, Arvand et al. 2011). Importantly, different drugs frequently target different domains of one and the same protein, for example, in the case of the Epidermal Growth Factor Receptor (EGFR), a key player in different types of cancer (Overington, Al-Lazikani et al. 2006).

The study of the functional plasticity of protein domain superfamilies, to which this work can hopefully contribute, is an endeavour that benefits directly from the concerted efforts mentioned above. Only because more and more protein functions are experimentally validated and carefully annotated can domain-specific functions be identified and the respective sequences grouped, to study their evolution. Only because more and more protein structures are solved can the respective structural domains be classified in resources such as SCOP and CATH and subsequently detected in proteins on a large scale. And, finally, only when methods are developed that allow for a biologically sound grouping of proteins and domains by their (annotated) functions can the evolution of function in protein and protein domain superfamilies be studied efficiently. Apart from providing intriguing insights,

as exemplified by many of the works listed in Appendix B, there is good reason to hope that such studies may also have an impact on medical research and drug discovery in the long term. This is because they can put findings about individual domains into context with the ‘wider’ picture that is provided by their superfamilies.

# Bibliography

- Addou, S., R. Rentzsch, et al. (2009). "Domain-based and family-specific sequence identity thresholds increase the levels of reliable protein function transfer." J Mol Biol **387**(2): 416-30.
- Albayrak, A., H. H. Otu, et al. (2010). "Clustering of protein families into functional subtypes using Relative Complexity Measure with reduced amino acid alphabets." BMC Bioinformatics **11**: 428.
- Allen, K. N. and D. Dunaway-Mariano (2009). "Markers of fitness in a successful enzyme superfamily." Curr Opin Struct Biol **19**(6): 658-65.
- Almonacid, D. E., E. R. Yera, et al. (2011). "Quantitative comparison of catalytic mechanisms and overall reactions in convergently evolved enzymes: implications for classification of enzyme function." PLoS Comput Biol **6**(3): e1000700.
- Altschul, S. F., W. Gish, et al. (1990). "Basic local alignment search tool." J Mol Biol **215**(3): 403-10.
- Altschul, S. F., T. L. Madden, et al. (1997). "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic Acids Res **25**(17): 3389-402.
- Ambrogelly, A., D. Korencic, et al. (2002). "Functional annotation of class I lysyl-tRNA synthetase phylogeny indicates a limited role for gene transfer." J Bacteriol **184**(16): 4594-600.
- Anderson, D. P. (2003). Public Computing: Reconnecting People to Science. Proceedings of the 2003 Conference on Shared Knowledge and the Web, Residencia de Estudiantes, Madrid, Spain.
- Apeltsin, L., J. H. Morris, et al. (2011). "Improving the quality of protein similarity network clustering algorithms using the network edge weight distribution." Bioinformatics **27**(3): 326-33.
- Apic, G., J. Gough, et al. (2001). "An insight into domain combinations." Bioinformatics **17 Suppl 1**: S83-9.
- Arakaki, A. K., Y. Huang, et al. (2009). "EFICAz2: enzyme function inference by a combined approach enhanced by machine learning." BMC Bioinformatics **10**: 107.
- Aravind, L., V. Anantharaman, et al. (2002). "Monophyly of class I aminoacyl tRNA synthetase, USPA, ETFP, photolyase, and PP-ATPase nucleotide-binding domains: implications for protein evolution in the RNA." Proteins **48**(1): 1-14.
- Aravind, L., D. D. Leipe, et al. (1998). "Toprim--a conserved catalytic domain in type IA and II topoisomerases, DnaG-type primases, OLD family nucleases and RecR proteins." Nucleic Acids Res **26**(18): 4205-13.
- Ashburner, M., C. A. Ball, et al. (2000). "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium." Nat Genet **25**(1): 25-9.



- Atkinson, H. J., J. H. Morris, et al. (2009). "Using sequence similarity networks for visualization of relationships across diverse protein superfamilies." PLoS One **4**(2): e4345.
- Attwood, T. K., P. Bradley, et al. (2003). "PRINTS and its automatic supplement, prePRINTS." Nucleic Acids Res **31**(1): 400-2.
- Axelsen, K. B. and M. G. Palmgren (1998). "Evolution of substrate specificities in the P-type ATPase superfamily." J Mol Evol **46**(1): 84-101.
- Babbitt, P. C., M. S. Hasson, et al. (1996). "The enolase superfamily: a general strategy for enzyme-catalyzed abstraction of the alpha-protons of carboxylic acids." Biochemistry **35**(51): 16489-501.
- Bailey, T. L., M. Boden, et al. (2009). "MEME SUITE: tools for motif discovery and searching." Nucleic Acids Res **37**(Web Server issue): W202-8.
- Baker, D. A. (2004). "Adenylyl and guanylyl cyclases from the malaria parasite *Plasmodium falciparum*." IUBMB Life **56**(9): 535-40.
- Baker, D. A. (2011). "Cyclic nucleotide signalling in malaria parasites." Cell Microbiol **13**(3): 331-9.
- Bannert, C., A. Welfle, et al. (2010). "BrEPS: a flexible and automatic protocol to compute enzyme-specific sequence profiles for functional annotation." BMC Bioinformatics **11**: 589.
- Barker, W. C., D. G. George, et al. (1993). "The PIR-International databases." Nucleic Acids Res **21**(13): 3089-92.
- Bashton, M. and C. Chothia (2007). "The generation of new protein functions by the combination of domains." Structure **15**(1): 85-99.
- Basu, M. K., E. Poliakov, et al. (2009). "Domain mobility in proteins: functional and evolutionary implications." Brief Bioinform **10**(3): 205-16.
- Bateman, A., P. Coggill, et al. (2010). "DUFs: families in search of function." Acta Crystallogr Sect F Struct Biol Cryst Commun **66**(Pt 10): 1148-52.
- Bellman, R. (1952). "On the Theory of Dynamic Programming." Proc Natl Acad Sci U S A **38**(8): 716-9.
- Benson, D. A., I. Karsch-Mizrachi, et al. (2011). "GenBank." Nucleic Acids Research **39**(suppl 1): D32-D37.
- Berkhin, P. (2002). *Survey Of Clustering Data Mining Techniques*.
- Berman, H. M., J. Westbrook, et al. (2000). "The Protein Data Bank." Nucleic Acids Res **28**(1): 235-42.
- Berthonneau, E. and M. Mirande (2000). "A gene fusion event in the evolution of aminoacyl-tRNA synthetases." FEBS Lett **470**(3): 300-4.

- Blatt, M., S. Wiseman, et al. (1996). "Superparamagnetic Clustering of Data." Physical Review Letters **76**(18): 3251-3254.
- Bloom, J. D., S. T. Labthavikul, et al. (2006). "Protein stability promotes evolvability." Proc Natl Acad Sci U S A **103**(15): 5869-74.
- Boc, A. and V. Makarenkov (2011). "Towards an accurate identification of mosaic genes and partial horizontal gene transfers." Nucleic Acids Res.
- Bramkamp, M., M. Gassel, et al. (2003). "The Methanocaldococcus jannaschii protein Mj0968 is not a P-type ATPase." FEBS Lett **543**(1-3): 31-6.
- Brown, D., N. Krishnamurthy, et al. (2005). "Subfamily HMMs in functional genomics." Pac Symp Biocomput: 322-33.
- Brown, D. P. (2008). "Efficient functional clustering of protein sequences using the Dirichlet process." Bioinformatics **24**(16): 1765-71.
- Brown, D. P., N. Krishnamurthy, et al. (2007). "Automated protein subfamily identification and classification." PLoS Comput Biol **3**(8): e160.
- Brown, M., R. Hughey, et al. (1993). "Using Dirichlet mixture priors to derive hidden Markov models for protein families." Proc Int Conf Intell Syst Mol Biol **1**: 47-55.
- Buchan, D. W., A. J. Shepherd, et al. (2002). "Gene3D: structural assignment for whole genes and genomes using the CATH domain structure database." Genome Res **12**(3): 503-14.
- Burroughs, A. M., K. N. Allen, et al. (2006). "Evolutionary genomics of the HAD superfamily: understanding the structural adaptations and catalytic diversity in a superfamily of phosphoesterases and allied enzymes." J Mol Biol **361**(5): 1003-34.
- Busch, W. and M. H. Saier, Jr. (2002). "The transporter classification (TC) system, 2002." Crit Rev Biochem Mol Biol **37**(5): 287-337.
- Bustamam, A., K. Burrage, et al. (2011). "Fast Parallel Markov Clustering in Bioinformatics using Massively Parallel Computing on GPU with CUDA and ELLPACK-R Sparse Format." IEEE/ACM Trans Comput Biol Bioinform.
- Buza, T. J., F. M. McCarthy, et al. (2008). "Gene Ontology annotation quality analysis in model eukaryotes." Nucleic Acids Res **36**(2): e12.
- Camon, E. B., D. G. Barrell, et al. (2005). "An evaluation of GO annotation retrieval for BioCreAtIvE and GOA." BMC Bioinformatics **6 Suppl 1**: S17.
- Carducci, M., L. Perfetto, et al. (2011). "The protein interaction network mediated by human SH3 domains." Biotechnol Adv.
- Chagoyen, M., J. M. Carazo, et al. (2008). "Assessment of protein set coherence using functional annotations." BMC Bioinformatics **9**: 444.
- Chagoyen, M. and F. Pazos (2011). "Quantifying the biological significance of gene ontology biological processes--implications for the analysis of systems-wide data." Bioinformatics **26**(3): 378-84.

- Chandonia, J. M. and S. E. Brenner (2006). "The impact of structural genomics: expectations and outcomes." *Science* **311**(5759): 347-51.
- Changjun, W. (2010). *A Scalable Parallel Algorithm for Large-Scale Protein Sequence Homology Detection*. 39th International Conference on Parallel Processing, San Diego, IEEE.
- Chen, F., A. J. Mackey, et al. (2006). "OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups." *Nucleic Acids Res* **34**(Database issue): D363-8.
- Chitale, M., S. Palakodety, et al. (2011). "Quantification of protein group coherence and pathway assignment using functional association." *BMC Bioinformatics* **12**: 373.
- Chothia, C., J. Gough, et al. (2003). "Evolution of the protein repertoire." *Science* **300**(5626): 1701-3.
- Chothia, C. and A. M. Lesk (1986). "The relation between the divergence of sequence and structure in proteins." *Embo J* **5**(4): 823-6.
- Chu, Y., X. Yuan, et al. (2010). "YeastWeb: a workset-centric web resource for gene family analysis in yeast." *BMC Genomics* **11**: 429.
- Claudiel-Renard, C., C. Chevalet, et al. (2003). "Enzyme-specific profiles for genome annotation: PRIAM." *Nucleic Acids Res* **31**(22): 6633-9.
- Cline, M. S., M. Smoot, et al. (2007). "Integration of biological networks and gene expression data using Cytoscape." *Nat Protoc* **2**(10): 2366-82.
- Cochrane, G., I. Karsch-Mizrachi, et al. (2011). "The International Nucleotide Sequence Database Collaboration." *Nucleic Acids Research* **39**(suppl 1): D15-D18.
- Costanzo, M. C., J. Park, et al. (2011). "Using computational predictions to improve literature-based Gene Ontology annotations: a feasibility study." *Database* **2011**.
- Cuff, A., O. C. Redfern, et al. (2009). "The CATH hierarchy revisited-structural divergence in domain superfamilies and the continuity of fold space." *Structure* **17**(8): 1051-62.
- Cusack, S. (1995). "Eleven down and nine to go." *Nat Struct Biol* **2**(10): 824-31.
- Dayhoff, M. O. (1974). "Computer analysis of protein sequences." *Fed Proc* **33**(12): 2314-6.
- Dayhoff, M. O., R. M. Schwartz, et al. (1978). A model of evolutionary change in proteins. *Atlas of protein sequence and structure*. **5**: 345-351.
- de Lima Morais, D. A., H. Fang, et al. (2011). "SUPERFAMILY 1.75 including a domain-centric gene ontology method." *Nucleic Acids Res* **39**(Database issue): D427-34.
- Desai, D. K., S. Nandi, et al. (2011). "ModEnzA: Accurate Identification of Metabolic Enzymes Using Function Specific Profile HMMs with Optimised Discrimination Threshold and Modified Emission Probabilities." *Adv Bioinformatics* **2011**: 743782.
- Dessailly, B. H., R. Nair, et al. (2009). "PSI-2: structural genomics to cover protein domain family space." *Structure* **17**(6): 869-81.

- Dessailly, B. H., O. C. Redfern, et al. (2010). "Detailed analysis of function divergence in a large and diverse domain superfamily: toward a refined protocol of function classification." Structure **18**(11): 1522-35.
- Dessimoz, C., G. Cannarozzi, et al. (2005). OMA, A Comprehensive, Automated Project for the Identification of Orthologs from Complete Genome Data: Introduction and First Achievements  
Comparative Genomics, Springer Berlin / Heidelberg. **3678**: 61-72.
- Dobzhansky, T. (1973). "Nothing in biology makes sense except in the light of evolution." American Biology Teacher **35**(3): 125-129.
- Dodd, I. B. and J. B. Egan (1987). "Systematic method for the detection of potential lambda Cro-like DNA-binding regions in proteins." J Mol Biol **194**(3): 557-64.
- Dokholyan, N. V., B. Shakhnovich, et al. (2002). "Expanding protein universe and its origin from the biological Big Bang." Proc Natl Acad Sci U S A **99**(22): 14132-6.
- Donald, J. E. and E. I. Shakhnovich (2005). "Determining functional specificity from protein sequences." Bioinformatics **21**(11): 2629-35.
- Doolittle, R. F. (1981). "Similar amino acid sequences: chance or common ancestry?" Science **214**(4517): 149-59.
- Doolittle, R. F. (1995). "The multiplicity of domains in proteins." Annu Rev Biochem **64**: 287-314.
- Eddy, S. R. (1998). "Profile hidden Markov models." Bioinformatics **14**(9): 755-63.
- Eddy, S. R. (2009). "A new generation of homology search tools based on probabilistic inference." Genome Inform **23**(1): 205-11.
- Edgar, R. C. (2010). "Search and clustering orders of magnitude faster than BLAST." Bioinformatics **26**(19): 2460-1.
- Edgar, R. C. and K. Sjolander (2003). "SATCHMO: sequence alignment and tree construction using hidden Markov models." Bioinformatics **19**(11): 1404-11.
- Eisen, J. A. (1998). "Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis." Genome Res **8**(3): 163-7.
- Ekman, D., A. K. Bjorklund, et al. (2005). "Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions." J Mol Biol **348**(1): 231-43.
- Engelhardt, B. E., M. I. Jordan, et al. (2005). "Protein molecular function prediction by Bayesian phylogenomics." PLoS Comput Biol **1**(5): e45.
- Engelhardt, B. E., M. I. Jordan, et al. (2009). "Phylogenetic molecular function annotation." J Phys **180**(1): 12024.
- Enright, A. J., V. Kunin, et al. (2003). "Protein families and TRIBES in genome sequence space." Nucleic Acids Res **31**(15): 4632-8.

- Enright, A. J., S. Van Dongen, et al. (2002). "An efficient algorithm for large-scale detection of protein families." Nucleic Acids Res **30**(7): 1575-84.
- Eriani, G., M. Delarue, et al. (1990). "Partition of tRNA synthetases into two classes based on mutually exclusive sets of sequence motifs." Nature **347**(6289): 203-6.
- Fang, G., N. Bhardwaj, et al. (2010). "Getting started in gene orthology and functional analysis." PLoS Comput Biol **6**(3): e1000703.
- Fayech, S., N. Essoussi, et al. (2009). "Partitioning clustering algorithms for protein sequence data sets." BioData Min **2**(1): 3.
- Finn, R. D., J. Mistry, et al. (2006). "Pfam: clans, web tools and services." Nucleic Acids Res **34**(Database issue): D247-51.
- Finn, R. D., J. Mistry, et al. (2010). "The Pfam protein families database." Nucleic Acids Res **38**(Database issue): D211-22.
- Finn, R. D., J. Tate, et al. (2008). "The Pfam protein families database." Nucleic Acids Res **36**(Database issue): D281-8.
- Fitch, W. M. (1970). "Distinguishing homologous from analogous proteins." Syst Zool **19**(2): 99-113.
- Fitch, W. M. (1971). "Toward defining the course of evolution: Minimum change for a specified tree topology." Syst Zool **20**: 406-416.
- Fitch, W. M. (2000). "Homology a personal view on some of the problems." Trends Genet **16**(5): 227-31.
- Flicek, P., M. R. Amode, et al. (2011). "Ensembl 2011." Nucleic Acids Res **39**(Database issue): D800-6.
- Fong, J. H. and A. Marchler-Bauer (2008). "Protein subfamily assignment using the Conserved Domain Database." BMC Res Notes **1**: 114.
- Forslund, K. and E. L. Sonnhammer (2008). "Predicting protein function from domain content." Bioinformatics **24**(15): 1681-7.
- Frankel, A. D. and J. A. Young (1998). "HIV-1: fifteen proteins and an RNA." Annu Rev Biochem **67**: 1-25.
- Frech, C. and N. Chen (2010). "Genome-wide comparative gene family classification." PLoS One **5**(10): e13409.
- Frey, B. J. and D. Dueck (2007). "Clustering by passing messages between data points." Science **315**(5814): 972-6.
- Frickey, T. and A. Lupas (2004). "CLANS: a Java application for visualizing protein families based on pairwise similarity." Bioinformatics **20**(18): 3702-4.
- Friedman, N., D. Geiger, et al. (1997). "Bayesian Network Classifiers." Machine Learning **29**(2): 131-163.

- Fukai, S., O. Nureki, et al. (2003). "Mechanism of molecular interactions for tRNA(Val) recognition by valyl-tRNA synthetase." *Rna* **9**(1): 100-11.
- Garza-Garcia, A., R. Harris, et al. (2009). "Solution structure and phylogenetics of Prod1, a member of the three-finger protein superfamily implicated in salamander limb regeneration." *PLoS One* **4**(9): e7123.
- Geisler, M., J. Richter, et al. (1993). "Molecular cloning of a P-type ATPase gene from the cyanobacterium *Synechocystis* sp. PCC 6803. Homology to eukaryotic Ca(2+)-ATPases." *J Mol Biol* **234**(4): 1284-9.
- Gerlt, J. A., K. N. Allen, et al. (2011). "The Enzyme Function Initiative." *Biochemistry*.
- Gilbert, W. (1978). "Why genes in pieces?" *Nature* **271**(5645): 501.
- Givoni, I. E. and B. J. Frey (2009). "A binary variable model for affinity propagation." *Neural Comput* **21**(6): 1589-600.
- Goldstein, R. A. (2008). "The structure of protein evolution and the evolution of protein structure." *Curr Opin Struct Biol* **18**(2): 170-7.
- Goll, J., R. Montgomery, et al. (2010). "The Protein Naming Utility: a rules database for protein nomenclature." *Nucleic Acids Res* **38**(Database issue): D336-9.
- Goodman, M. F. (2002). "Error-prone repair DNA polymerases in prokaryotes and eukaryotes." *Annu Rev Biochem* **71**: 17-50.
- Gough, J. (2006). "Genomic scale sub-family assignment of protein domains." *Nucleic Acids Res* **34**(13): 3625-33.
- Gough, J., K. Karplus, et al. (2001). "Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure." *J Mol Biol* **313**(4): 903-19.
- Greene, L. H., T. E. Lewis, et al. (2007). "The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution." *Nucleic Acids Res* **35**(Database issue): D291-7.
- Gribskov, M., A. D. McLachlan, et al. (1987). "Profile analysis: detection of distantly related proteins." *Proc Natl Acad Sci U S A* **84**(13): 4355-8.
- Grillo, G., M. Attimonelli, et al. (1996). "CLEANUP: a fast computer program for removing redundancies from nucleotide sequence databases." *Comput Appl Biosci* **12**(1): 1-8.
- Grishin, N. V. (2001). "Fold change in evolution of protein structures." *J Struct Biol* **134**(2-3): 167-85.
- Gu, X., Z. Zhang, et al. (2005). "Rapid evolution of expression and regulatory divergences after yeast gene duplication." *Proc Natl Acad Sci U S A* **102**(3): 707-12.
- Haft, D. H., B. J. Loftus, et al. (2001). "TIGRFAMs: a protein family resource for the functional identification of proteins." *Nucleic Acids Res* **29**(1): 41-3.

- Haft, D. H., J. D. Selengut, et al. (2003). "The TIGRFAMs database of protein families." Nucleic Acids Res **31**(1): 371-3.
- Hagopian, R., J. R. Davidson, et al. (2010). "SATCIMO-JS: a webserver for simultaneous protein multiple sequence alignment and phylogenetic tree construction." Nucleic Acids Res **38**(Web Server issue): W29-34.
- Han, M. V. and C. M. Zmasek (2009). "phyloXML: XML for evolutionary biology and comparative genomics." BMC Bioinformatics **10**: 356.
- Hang, C. T., J. Yang, et al. (2010). "Chromatin regulation by Brg1 underlies heart muscle development and disease." Nature **466**(7302): 62-7.
- Hasegawa, H. and L. Holm (2009). "Advances and pitfalls of protein structural alignment." Curr Opin Struct Biol **19**(3): 341-8.
- Hayete, B. and J. R. Bienkowska (2005). "Gotrees: predicting go associations from protein domain composition using decision trees." Pac Symp Biocomput: 127-38.
- Henikoff, S. and J. G. Henikoff (1992). "Amino acid substitution matrices from protein blocks." Proc Natl Acad Sci U S A **89**(22): 10915-9.
- Henikoff, S. and J. G. Henikoff (1994). "Position-based sequence weights." J Mol Biol **243**(4): 574-8.
- Henikoff, S. and J. G. Henikoff (2001). Protein Family Databases. eLS, John Wiley & Sons, Ltd.
- Hill, D. P., A. P. Davis, et al. (2001). "Program description: Strategies for biological annotation of mammalian systems: implementing gene ontologies in mouse genome informatics." Genomics **74**(1): 121-8.
- Hobohm, U., M. Scharf, et al. (1992). "Selection of representative protein data sets." Protein Sci **1**(3): 409-17.
- Hoffmann, R. and A. Valencia (2004). "A gene network for navigating the literature." Nat Genet **36**(7): 664.
- Holm, L. and C. Sander (1998). "Removing near-neighbour redundancy from large protein sequence collections." Bioinformatics **14**(5): 423-9.
- Hu, Z., J. H. Hung, et al. (2009). "VisANT 3.5: multi-scale network visualization, analysis and inference based on the gene ontology." Nucleic Acids Res **37**(Web Server issue): W115-21.
- Huberts, D. H. and I. J. van der Klei (2010). "Moonlighting proteins: an intriguing mode of multitasking." Biochim Biophys Acta **1803**(4): 520-5.
- Hughey, R. and A. Krogh (1996). "Hidden Markov models for sequence analysis: extension and analysis of the basic method." Comput Appl Biosci **12**(2): 95-107.
- Hunter, S., R. Apweiler, et al. (2009). "InterPro: the integrative protein signature database." Nucleic Acids Res **37**(Database issue): D211-5.

- Ibrahim, S. S., M. A. R. Eldeeb, et al. (2011). "The role of protein interaction domains in the human cancer network." Network Biology **1**(1): 59-71.
- Illergard, K., D. H. Ardell, et al. (2009). "Structure is three to ten times more conserved than sequence--a study of structural response in protein cores." Proteins **77**(3): 499-508.
- Islam, S. A., J. Luo, et al. (1995). "Identification and analysis of domains in proteins." Protein Eng **8**(6): 513-25.
- Itzhaki, Z. (2011). "Domain-domain interactions underlying herpesvirus-human protein-protein interaction networks." PLoS One **6**(7): e21724.
- Iyer, L. M., V. Anantharaman, et al. (2003). "Ancient conserved domains shared by animal soluble guanylyl cyclases and bacterial signaling proteins." BMC Genomics **4**(1): 5.
- Jaccard, P. (1901). "Étude comparative de la distribution florale dans une portion des Alpes et des Jura." Bulletin del la Société Vaudoise des Sciences Naturelles **37**: 241-272.
- Jaromczyk, J. W. and G. T. Toussaint (1992). Relative Neighborhood Graphs And Their Relatives. Proceedings of the IEEE.
- Jaroszewski, L., Z. Li, et al. (2009). "Exploration of uncharted regions of the protein universe." PLoS Biol **7**(9): e1000205.
- Jeffery, C. J. (1999). "Moonlighting proteins." Trends Biochem Sci **24**(1): 8-11.
- Johnson, S. C. (1967). "Hierarchical clustering schemes." Psychometrika **32**(3): 241-254.
- Jones, C. E., A. L. Brown, et al. (2007). "Estimating the annotation error rate of curated GO database sequence annotations." BMC Bioinformatics **8**: 170.
- Kahn, S. D. (2011). "On the future of genomic data." Science **331**(6018): 728-9.
- Kalinina, O. V., M. S. Gelfand, et al. (2009). "Combining specificity determining and conserved residues improves functional site prediction." BMC Bioinformatics **10**: 174.
- Kanehisa, M. and S. Goto (2000). "KEGG: kyoto encyclopedia of genes and genomes." Nucleic Acids Res **28**(1): 27-30.
- Kanehisa, M., S. Goto, et al. (2004). "The KEGG resource for deciphering the genome." Nucleic Acids Res **32**(Database issue): D277-80.
- Katoh, K., K. Kuma, et al. (2005). "MAFFT version 5: improvement in accuracy of multiple sequence alignment." Nucleic Acids Res **33**(2): 511-8.
- Katoh, K. and H. Toh (2008). "Recent developments in the MAFFT multiple sequence alignment program." Brief Bioinform **9**(4): 286-98.
- Kelil, A., S. Wang, et al. (2008). "CLUSS2: an alignment-independent algorithm for clustering protein families with multiple biological functions." Int J Comput Biol Drug Des **1**(2): 122-40.



- Kelil, A., S. Wang, et al. (2007). "CLUSS: clustering of protein sequences based on a new similarity measure." *BMC Bioinformatics* **8**: 286.
- King, N., M. J. Westbrook, et al. (2008). "The genome of the choanoflagellate *Monosiga brevicollis* and the origin of metazoans." *Nature* **451**(7180): 783-8.
- Klimke, W., R. Agarwala, et al. (2009). "The National Center for Biotechnology Information's Protein Clusters Database." *Nucleic Acids Res* **37**(Database issue): D216-23.
- Kolodny, R., D. Petrey, et al. (2006). "Protein structure comparison: implications for the nature of 'fold space', and structure and function prediction." *Curr Opin Struct Biol* **16**(3): 393-8.
- Koonin, E. V. (2005). "Orthologs, paralogs, and evolutionary genomics." *Annu Rev Genet* **39**: 309-38.
- Koonin, E. V. (2011). "Are There Laws of Genome Evolution?" *PLoS Comput Biol* **7**(8): e1002173.
- Koonin, E. V. and R. L. Tatusov (1994). "Computer analysis of bacterial haloacid dehalogenases defines a large superfamily of hydrolases with diverse specificity. Application of an iterative approach to database search." *J Mol Biol* **244**(1): 125-32.
- Koonin, E. V., Y. I. Wolf, et al. (2002). "The structure of the protein universe and genome evolution." *Nature* **420**(6912): 218-23.
- Koski, L. B., M. W. Gray, et al. (2005). "AutoFACT: an automatic functional annotation and classification tool." *BMC Bioinformatics* **6**: 151.
- Krishnamurthy, N., D. P. Brown, et al. (2006). "PhyloFacts: an online structural phylogenomic encyclopedia for protein functional and structural classification." *Genome Biol* **7**(9): R83.
- Kriventseva, E. V., W. Fleischmann, et al. (2001). "CluSTr: a database of clusters of SWISS-PROT+TrEMBL proteins." *Nucleic Acids Res* **29**(1): 33-6.
- Kull, M. and J. Vilo (2008). "Fast approximate hierarchical clustering using similarity heuristics." *BioData Min* **1**(1): 9.
- Lavelle, D. T. and W. R. Pearson (2009). "Globally, unrelated protein sequences appear random." *Bioinformatics* **26**(3): 310-8.
- Lee, D., A. Grant, et al. (2005). "Identification and distribution of protein families in 120 completed genomes using Gene3D." *Proteins* **59**(3): 603-15.
- Lee, D. A., R. Rentzsch, et al. (2010). "GeMMA: functional subfamily classification within superfamilies of predicted protein structural domains." *Nucleic Acids Res* **38**(3): 720-37.
- Letunic, I. and P. Bork (2011). "Interactive Tree Of Life v2: online annotation and display of phylogenetic trees made easy." *Nucleic Acids Res* **39**(Web Server issue): W475-8.
- Letunic, I., T. Doerks, et al. (2009). "SMART 6: recent updates and new developments." *Nucleic Acids Res* **37**(Database issue): D229-32.

- Lewin, R. (1987). "When does homology mean something else?" *Science* **237**(4822): 1570.
- Li, L., C. J. Stoeckert, Jr., et al. (2003). "OrthoMCL: identification of ortholog groups for eukaryotic genomes." *Genome Res* **13**(9): 2178-89.
- Li, W. and A. Godzik (2006). "Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences." *Bioinformatics* **22**(13): 1658-9.
- Li, W., L. Jaroszewski, et al. (2001). "Clustering of highly homologous sequences to reduce the size of large protein databases." *Bioinformatics* **17**(3): 282-3.
- Liekens, A. M., J. De Knijf, et al. (2011). "BioGraph: unsupervised biomedical knowledge discovery via automated hypothesis generation." *Genome Biol* **12**(6): R57.
- Linder, J. U., P. Engel, et al. (1999). "Guanylyl cyclases with the topology of mammalian adenylyl cyclases and an N-terminal P-type ATPase-like domain in Paramecium, Tetrahymena and Plasmodium." *Embo J* **18**(15): 4222-32.
- Lipman, D. J. and W. R. Pearson (1985). "Rapid and sensitive protein similarity searches." *Science* **227**(4693): 1435-41.
- Liu, J. and B. Rost (2003). "Domains, motifs and clusters in the protein universe." *Curr Opin Chem Biol* **7**(1): 5-11.
- Liu, M. and A. Grigoriev (2004). "Protein domains correlate strongly with exons in multiple eukaryotic genomes--evidence of exon shuffling?" *Trends Genet* **20**(9): 399-403.
- Loewenstein, Y., E. Portugaly, et al. (2008). "Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space." *Bioinformatics* **24**(13): i41-9.
- Lopez, D. and F. Pazos (2009). "Gene ontology functional annotations at the structural domain level." *Proteins* **76**(3): 598-607.
- Luo, Q., P. Pagel, et al. (2011). "DIMA 3.0: Domain Interaction Map." *Nucleic Acids Res* **39**(Database issue): D724-9.
- Macqueen, J. B. (1967). *Some Methods of Classification and Analysis of Multivariate Observations*. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.
- Madera, M. (2008). "Profile Comparer: a program for scoring and aligning profile hidden Markov models." *Bioinformatics* **24**(22): 2630-1.
- Magrane, M. and U. Consortium (2011). "UniProt Knowledgebase: a hub of integrated protein data." *Database (Oxford)* **2011**: bar009.
- Mair, G. R., J. A. Braks, et al. (2006). "Regulation of sexual development of Plasmodium by translational repression." *Science* **313**(5787): 667-9.
- Marabotti, A. and A. Facchiano (2009). "When it comes to homology, bad habits die hard." *Trends Biochem Sci* **34**(3): 98-9.
- Marabotti, A. and A. Facchiano (2010). "The misuse of terms in scientific literature." *Bioinformatics* **26**(19): 2498.

- Marchler-Bauer, A., J. B. Anderson, et al. (2005). "CDD: a Conserved Domain Database for protein classification." Nucleic Acids Res **33**(Database issue): D192-6.
- Marchler-Bauer, A., S. Lu, et al. (2011). "CDD: a Conserved Domain Database for the functional annotation of proteins." Nucleic Acids Res **39**(Database issue): D225-9.
- Marcotte, E. M., M. Pellegrini, et al. (1999). "A combined algorithm for genome-wide prediction of protein function." Nature **402**(6757): 83-6.
- Martin, D. M., M. Berriman, et al. (2004). "GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes." BMC Bioinformatics **5**: 178.
- Medvés, L., L. Szilágyi, et al. (2008). A Modified Markov Clustering Approach for Protein Sequence Clustering
- Pattern Recognition in Bioinformatics, Springer Berlin / Heidelberg. **5265**: 110-120.
- Meila, M. (2007). "Comparing clusterings--an information based distance." Journal of Multivariate Analysis **98**(5): 873-895.
- Michener, C. D. and R. R. Sokal (1957). "A Quantitative Approach to a Problem in Classification." Evolution **11**(2): 130-162.
- Miele, V., S. Penel, et al. (2011). "Ultra-fast sequence clustering from similarity networks with SiLiX." BMC Bioinformatics **12**: 116.
- Mirny, L. A. and E. I. Shakhnovich (1999). "Universally conserved positions in protein folds: reading evolutionary signals about stability, folding kinetics and function." J Mol Biol **291**(1): 177-96.
- Moll, M., D. H. Bryant, et al. (2010). "The LabelHash algorithm for substructure matching." BMC Bioinformatics **11**: 555.
- Moon, R. W., C. J. Taylor, et al. (2009). "A cyclic GMP signalling module that regulates gliding motility in a malaria parasite." PLoS Pathog **5**(9): e1000599.
- Moore, G. (1965). "Cramming more components onto integrated circuits." Electronics **38**(8).
- Mulder, N. J. (2001). Protein Family Databases. eLS, John Wiley & Sons, Ltd.
- Murzin, A. G., S. E. Brenner, et al. (1995). "SCOP: a structural classification of proteins database for the investigation of sequences and structures." J Mol Biol **247**(4): 536-40.
- Nagy, A. and L. Patthy (2011). "Reassessing Domain Architecture Evolution of Metazoan Proteins: The Contribution of Different Evolutionary Mechanisms." Genes **2**(3): 578-598.
- Natale, D. A., C. N. Arighi, et al. (2007). "Framework for a protein ontology." BMC Bioinformatics **8 Suppl 9**: S1.

- Natale, D. A., C. N. Arighi, et al. (2011). "The Protein Ontology: a structured representation of protein forms and complexes." Nucleic Acids Res **39**(Database issue): D539-45.
- Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." J Mol Biol **48**(3): 443-53.
- Nehrt, N. L., W. T. Clark, et al. (2011). "Testing the ortholog conjecture with comparative functional genomic data from mammals." PLoS Comput Biol **7**(6): e1002073.
- Nikolskaya, A. N., C. N. Arighi, et al. (2006). "PIRSF family classification system for protein functional and evolutionary analysis." Evol Bioinform Online **2**: 197-209.
- Nordin, B. E. and P. Schimmel (2003). "Transiently misacylated tRNA is a primer for editing of misactivated adenylates by class I aminoacyl-tRNA synthetases." Biochemistry **42**(44): 12989-97.
- Nureki, O., S. Fukai, et al. (2001). "Structural basis for amino acid and tRNA recognition by class I aminoacyl-tRNA synthetases." Cold Spring Harb Symp Quant Biol **66**: 167-73.
- Nureki, O., D. G. Vassylyev, et al. (1998). "Enzyme structure with two catalytic sites for double-sieve selection of substrate." Science **280**(5363): 578-82.
- Ogawa, H., T. Haga, et al. (2000). "Soluble P-type ATPase from an archaeon, *Methanococcus jannaschii*." FEBS Lett **471**(1): 99-102.
- Ohno, S. (1970). Evolution by Gene Duplication, Springer-Verlag.
- Olman, V., F. Mao, et al. (2009). "Parallel clustering algorithm for large data sets with applications in bioinformatics." IEEE/ACM Trans Comput Biol Bioinform **6**(2): 344-52.
- Orengo, C. A., A. D. Michie, et al. (1997). "CATH--a hierarchic classification of protein domain structures." Structure **5**(8): 1093-108.
- Orengo, C. A. and W. R. Taylor (1996). "SSAP: sequential structure alignment program for protein structure comparison." Methods Enzymol **266**: 617-35.
- Ostermann, S., A. Iosup, et al. (2010). A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing. Cloud Computing. O. Akan, P. Bellavista, J. Cao et al, Springer Berlin Heidelberg. **34**: 115-131.
- Ouzounis, C. A. and A. Valencia (2003). "Early bioinformatics: the birth of a discipline--a personal view." Bioinformatics **19**(17): 2176-90.
- Overington, J. P., B. Al-Lazikani, et al. (2006). "How many drug targets are there?" Nat Rev Drug Discov **5**(12): 993-996.
- Paccanaro, A., J. A. Casbon, et al. (2006). "Spectral clustering of protein sequences." Nucleic Acids Res **34**(5): 1571-80.
- Palmgren, M. G. and P. Nissen (2011). "P-type ATPases." Annu Rev Biophys **40**: 243-66.

- Pandit, S. B., D. Gosar, et al. (2002). "SUPFAM--a database of potential protein superfamily relationships derived by comparing sequence-based and structure-based families: implications for structural genomics and function annotation in genomes." Nucleic Acids Res **30**(1): 289-93.
- Park, Y. R., J. Kim, et al. (2011). "GOChase-II: correcting semantic inconsistencies from Gene Ontology-based annotations for gene products." BMC Bioinformatics **12 Suppl 1**: S40.
- Patthy, L. (1999). "Genome evolution and the evolution of exon-shuffling--a review." Gene **238**(1): 103-14.
- Pawson, T. and P. Nash (2003). "Assembly of cell regulatory systems through protein interaction domains." Science **300**(5618): 445-52.
- Pazos, F. and M. J. Sternberg (2004). "Automated prediction of protein function and detection of functional sites from structure." Proc Natl Acad Sci U S A **101**(41): 14754-9.
- Pearl, F., A. Todd, et al. (2005). "The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis." Nucleic Acids Res **33**(Database issue): D247-51.
- Pearson, W. R. and M. L. Sierk (2005). "The limits of protein sequence comparison?" Curr Opin Struct Biol **15**(3): 254-60.
- Pegg, S. C., S. Brown, et al. (2005). "Representing structure-function relationships in mechanistically diverse enzyme superfamilies." Pac Symp Biocomput: 358-69.
- Pesquita, C., D. Faria, et al. (2009). "Semantic similarity in biomedical ontologies." PLoS Comput Biol **5**(7): e1000443.
- Petrey, D. and B. Honig (2009). "Is protein classification necessary? Toward alternative approaches to function annotation." Current Opinion in Structural Biology **19**(3): 363-368.
- Podell, S. and T. Gaasterland (2007). "DarkHorse: a method for genome-wide prediction of horizontal gene transfer." Genome Biol **8**(2): R16.
- Pruitt, K. D., T. Tatusova, et al. (2009). "NCBI Reference Sequences: current status, policy and new initiatives." Nucleic Acids Res **37**(Database issue): D32-6.
- Qian, J., N. M. Luscombe, et al. (2001). "Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model." J Mol Biol **313**(4): 673-81.
- Rand, W. M. (1971). "Objective criteria for the evaluation of clustering methods." Journal of the American Statistical association: 846-850.
- Rattei, T., P. Tischler, et al. (2008). "SIMAP--structuring the network of protein similarities." Nucleic Acids Res **36**(Database issue): D289-92.
- Redfern, O., A. Grant, et al. (2005). "Survey of current protein family databases and their application in comparative, structural and functional genomics." J Chromatogr B Analyt Technol Biomed Life Sci **815**(1-2): 97-107.

- Reeck, G. R., C. de Haen, et al. (1987). "'Homology' in proteins and nucleic acids: a terminology muddle and a way out of it." *Cell* **50**(5): 667.
- Rensing, C., B. Fan, et al. (2000). "CopA: An Escherichia coli Cu(I)-translocating P-type ATPase." *Proc Natl Acad Sci U S A* **97**(2): 652-6.
- Rentzsch, R. and C. A. Orengo (2009). "Protein function prediction--the power of multiplicity." *Trends Biotechnol* **27**(4): 210-9.
- Ribas de Pouplana, L. and P. Schimmel (2001). "Aminoacyl-tRNA synthetases: potential markers of genetic code development." *Trends Biochem Sci* **26**(10): 591-6.
- Riley, M. (1993). "Functions of the gene products of Escherichia coli." *Microbiol Rev* **57**(4): 862-952.
- Riley, M. (2007). "Searchlight on domains." *Structure* **15**(1): 1-2.
- Rison, S. C., T. C. Hodgman, et al. (2000). "Comparison of functional annotation schemes for genomes." *Funct Integr Genomics* **1**(1): 56-69.
- Roberts, R. J., Y. C. Chang, et al. (2011). "COMBREX: a project to accelerate the functional annotation of prokaryotic genomes." *Nucleic Acids Res* **39**(Database issue): D11-4.
- Rorick, M. M. and G. P. Wagner (2011). "Protein structural modularity and robustness are associated with evolvability." *Genome Biol Evol* **3**: 456-75.
- Rose, P. W., B. Beran, et al. (2011). "The RCSB Protein Data Bank: redesigned web site and web services." *Nucleic Acids Res* **39**(Database issue): D392-401.
- Rosenberg, A. and J. Hirschberg (2007). V-measure: A conditional entropy-based external cluster evaluation measure. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).
- Rossmann, M. G., D. Moras, et al. (1974). "Chemical and biological evolution of nucleotide-binding protein." *Nature* **250**(463): 194-9.
- Rothberg, J. M., W. Hinz, et al. (2011). "An integrated semiconductor device enabling non-optical genome sequencing." *Nature* **475**(7356): 348-52.
- Ruepp, A., A. Zollner, et al. (2004). "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes." *Nucleic Acids Res* **32**(18): 5539-45.
- Rupke, N. A. (1993). "Richard Owen's vertebrate archetype." *Isis* **84**(2): 231-51.
- Sadreyev, R. and N. Grishin (2003). "COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance." *J Mol Biol* **326**(1): 317-36.
- Salmena, L., L. Poliseno, et al. (2011). "A ceRNA hypothesis: the Rosetta Stone of a hidden RNA language?" *Cell* **146**(3): 353-8.
- Sankararaman, S. and K. Sjolander (2008). "INTREPID--INformation-theoretic TREE traversal for Protein functional site IDentification." *Bioinformatics* **24**(21): 2445-52.

- Sasson, O., A. Vaaknin, et al. (2003). "ProtoNet: hierarchical classification of the protein space." Nucleic Acids Res **31**(1): 348-52.
- Schaeffer, S. (2007). "Graph clustering." Computer Science Review **1**(1): 27-64.
- Schmid, R. and M. L. Blaxter (2008). "annot8r: GO, EC and KEGG annotation of EST datasets." BMC Bioinformatics **9**: 180.
- Schnoes, A. M., S. D. Brown, et al. (2009). "Annotation error in public databases: misannotation of molecular function in enzyme superfamilies." PLoS Comput Biol **5**(12): e1000605.
- Schug, J., S. Diskin, et al. (2002). "Predicting gene ontology functions from ProDom and CDD protein domains." Genome Res **12**(4): 648-55.
- Schwede, T., J. Kopp, et al. (2003). "SWISS-MODEL: An automated protein homology-modeling server." Nucleic Acids Res **31**(13): 3381-5.
- Seffernick, J. L., M. L. de Souza, et al. (2001). "Melamine deaminase and atrazine chlorohydrolase: 98 percent identical but functionally different." J Bacteriol **183**(8): 2405-10.
- Servant, F., C. Bru, et al. (2002). "ProDom: automated clustering of homologous domains." Brief Bioinform **3**(3): 246-51.
- Seshadri, R., S. A. Kravitz, et al. (2007). "CAMERA: a community resource for metagenomics." PLoS Biol **5**(3): e75.
- Shi, J. and J. Malik (2000). "Normalized cuts and image segmentation." Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(8): 888-905.
- Shindyalov, I. N. and P. E. Bourne (2000). "An alternative view of protein fold space." Proteins **38**(3): 247-60.
- Sigrist, C. J., L. Cerutti, et al. (2002). "PROSITE: a documented database using patterns and profiles as motif descriptors." Brief Bioinform **3**(3): 265-74.
- Sjolander, K. (1998). "Phylogenetic inference in protein superfamilies: analysis of SH2 domains." Proc Int Conf Intell Syst Mol Biol **6**: 165-74.
- Sjolander, K., K. Karplus, et al. (1996). "Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology." Comput Appl Biosci **12**(4): 327-45.
- Skou, J. C. (1957). "The influence of some cations on an adenosine triphosphatase from peripheral nerves." Biochimica et biophysica acta **23**: 394-401.
- Smith, T. F. and M. S. Waterman (1981). "Identification of common molecular subsequences." J Mol Biol **147**(1): 195-7.
- Soding, J. (2005). "Protein homology detection by HMM-HMM comparison." Bioinformatics **21**(7): 951-60.
- Soding, J. and M. Remmert (2011). "Protein sequence comparison and fold recognition: progress and good-practice benchmarking." Curr Opin Struct Biol **21**(3): 404-11.

- Song, N., J. M. Joseph, et al. (2008). "Sequence similarity network reveals common ancestry of multidomain proteins." *PLoS Comput Biol* **4**(4): e1000063.
- Song, N., R. D. Sedgewick, et al. (2007). "Domain architecture comparison for multidomain homology identification." *J Comput Biol* **14**(4): 496-516.
- Sonnhammer, E. L., S. R. Eddy, et al. (1997). "Pfam: a comprehensive database of protein domain families based on seed alignments." *Proteins* **28**(3): 405-20.
- Sonnhammer, E. L. and E. V. Koonin (2002). "Orthology, paralogy and proposed classification for paralog subtypes." *Trends Genet* **18**(12): 619-20.
- Spitzweck, B., M. Brankatschk, et al. (2011). "Distinct protein domains and expression patterns confer divergent axon guidance functions for *Drosophila* Robo receptors." *Cell* **140**(3): 409-20.
- Stanewsky, R., M. Kaneko, et al. (1998). "The cryb mutation identifies cryptochrome as a circadian photoreceptor in *Drosophila*." *Cell* **95**(5): 681-92.
- Storm, C. E. and E. L. Sonnhammer (2003). "Comprehensive analysis of orthologous protein domains using the HOPS database." *Genome Res* **13**(10): 2353-62.
- Studer, R. A. and M. Robinson-Rechavi (2009). "How confident can we be that orthologs are similar, but paralogs differ?" *Trends in genetics : TIG* **25**(5): 210-216.
- Sun, C. H., M. S. Kim, et al. (2009). "COFECO: composite function annotation enriched by protein complex data." *Nucleic Acids Res* **37**(Web Server issue): W350-5.
- Suzek, B. E., H. Huang, et al. (2007). "UniRef: comprehensive and non-redundant UniProt reference clusters." *Bioinformatics* **23**(10): 1282-8.
- Szklarczyk, D., A. Franceschini, et al. (2011). "The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored." *Nucleic Acids Res* **39**(Database issue): D561-8.
- Tan, P. N., M. Steinbach, et al. (2005). "Cluster Analysis: basic concepts and algorithms." *Introduction to data mining*: 487-568.
- Tatusov, R. L., S. F. Altschul, et al. (1994). "Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks." *Proc Natl Acad Sci U S A* **91**(25): 12091-5.
- Tatusov, R. L., N. D. Fedorova, et al. (2003). "The COG database: an updated version includes eukaryotes." *BMC Bioinformatics* **4**: 41.
- Taylor, W. R. (2007). "Evolutionary transitions in protein fold space." *Curr Opin Struct Biol* **17**(3): 354-61.
- Taylor, W. R. and C. A. Orengo (1989). "Protein structure alignment." *J Mol Biol* **208**(1): 1-22.
- Thever, M. and M. Saier (2009). "Bioinformatic Characterization of P-Type ATPases Encoded Within the Fully Sequenced Genomes of 26 Eukaryotes." *Journal of Membrane Biology* **229**(3): 115-130.



- Thomas, P. D. (2010). "GIGA: a simple, efficient algorithm for gene tree inference in the genomic age." *BMC Bioinformatics* **11**: 312.
- Thomas, P. D., M. J. Campbell, et al. (2003). "PANTHER: a library of protein families and subfamilies indexed by function." *Genome Res* **13**(9): 2129-41.
- Thompson, J. D., D. G. Higgins, et al. (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." *Nucleic Acids Res* **22**(22): 4673-80.
- Thompson, J. D., B. Linard, et al. (2011). "A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives." *PLoS One* **6**(3): e18093.
- Toyoshima, C., M. Nakasako, et al. (2000). "Crystal structure of the calcium pump of sarcoplasmic reticulum at 2.6 Å resolution." *Nature* **405**(6787): 647-55.
- Tricker, E., A. Arvand, et al. (2011). "Apoptosis induced by cytoskeletal disruption requires distinct domains of MEKK1." *PLoS One* **6**(2): e17310.
- Tumer, Z., L. B. Moller, et al. (1999). "Mutation spectrum of ATP7A, the gene defective in Menkes disease." *Adv Exp Med Biol* **448**: 83-95.
- van den Berg, B. H., F. M. McCarthy, et al. (2010). "Re-annotation is an essential step in systems biology modeling of functional genomics data." *PLoS One* **5**(5): e10642.
- van Dongen, S. (2000). A cluster algorithm for graphs, CWI (Centre for Mathematics and Computer Science).
- Vinh, N., J. Epps, et al. (2010). "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance." *Journal of Machine Learning Research*.
- Viterbi, A. (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." *Information Theory, IEEE Transactions on* **13**(2): 260-269.
- Vogel, C., S. A. Teichmann, et al. (2003). "The immunoglobulin superfamily in *Drosophila melanogaster* and *Caenorhabditis elegans* and the evolution of complexity." *Development* **130**(25): 6317-28.
- Vogel, C., S. A. Teichmann, et al. (2005). "The relationship between domain duplication and recombination." *J Mol Biol* **346**(1): 355-65.
- Wall, P. K., J. Leebens-Mack, et al. (2008). "PlantTribes: a gene and gene family resource for comparative genomics in plants." *Nucleic Acids Res* **36**(Database issue): D970-6.
- Walter, C. (2005). "Kryder's law." *Sci Am* **293**(2): 32-3.
- Wang, J., P. Korambath, et al. (2011). Facilitating e-Science Discovery Using Scientific Workflows on the Grid
- Guide to e-Science, Springer London: 353-382.
- Wang, J., M. Li, et al. (2010). "Recent advances in clustering methods for protein interaction networks." *BMC Genomics* **11 Suppl 3**: S10.

- Wang, M., C. G. Kurland, et al. (2011). "Reductive evolution of proteomes and protein structures." Proc Natl Acad Sci U S A **108**(29): 11954-8.
- Wang, Y., R. I. Sadreyev, et al. (2009). "PROCAIN: protein profile comparison with assisting information." Nucleic Acids Res **37**(11): 3522-30.
- Wass, M. N. and M. J. Sternberg (2008). "ConFunc--functional annotation in the twilight zone." Bioinformatics **24**(6): 798-806.
- Webb, E. C. (1992). Enzyme nomenclature 1992 : recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes, Academic Press.
- Wetlaufer, D. B. (1973). "Nucleation, rapid folding, and globular intrachain regions in proteins." Proc Natl Acad Sci U S A **70**(3): 697-701.
- Wilson, D., M. Madera, et al. (2007). "The SUPERFAMILY database in 2007: families and functions." Nucleic Acids Res **35**(Database issue): D308-13.
- Wittkop, T., D. Emig, et al. (2010). "Partitioning biological data with transitivity clustering." Nat Methods **7**(6): 419-20.
- Woese, C. R., G. J. Olsen, et al. (2000). "Aminoacyl-tRNA synthetases, the genetic code, and the evolutionary process." Microbiol Mol Biol Rev **64**(1): 202-36.
- Wolf, Y. I., L. Aravind, et al. (1999). "Evolution of aminoacyl-tRNA synthetases--analysis of unique domain architectures and phylogenetic trees reveals a complex history of horizontal gene transfer events." Genome Res **9**(8): 689-710.
- Wooley, J. C., A. Godzik, et al. (2010). "A primer on metagenomics." PLoS Comput Biol **6**(2): e1000667.
- Xie, X., J. Jin, et al. (2011). "Evolutionary versatility of eukaryotic protein domains revealed by their bigram networks." BMC Evol Biol **11**: 242.
- Yang, F., Q. Zhu, et al. (2010). "Using affinity propagation combined post-processing to cluster protein sequences." Protein Pept Lett **17**(6): 681-9.
- Yang, S. and P. E. Bourne (2009). "The evolutionary history of protein domains viewed by species phylogeny." PLoS One **4**(12): e8378.
- Yang, X., J. Zola, et al. (2011). Parallel Metagenomic Sequence Clustering via Sketching and Maximal Quasi-clique Enumeration on Map-reduce Clouds. Parallel & Distributed Processing Symposium (IPDPS), Anchorage, IEEE.
- Yeats, C., J. Lees, et al. (2011). "The Gene3D Web Services: a platform for identifying, annotating and comparing structural domains in protein sequences." Nucleic Acids Res **39**(Web Server issue): W546-50.
- Yeats, C., O. C. Redfern, et al. (2010). "A fast and automated solution for accurately resolving protein domain architectures." Bioinformatics **26**(6): 745-51.
- Yeats, C. A. and C. A. Orengo (2001). Evolution of Protein Domains. eLS, John Wiley & Sons, Ltd.

- Yu, C., N. Zavaljevski, et al. (2009). "Genome-wide enzyme annotation with precision control: catalytic families (CatFam) databases." Proteins **74**(2): 449-60.
- Zelensky, A. N. and J. E. Gready (2005). "The C-type lectin-like domain superfamily." Febs J **272**(24): 6179-217.

# Appendix A – HPC implementation of GeMMA

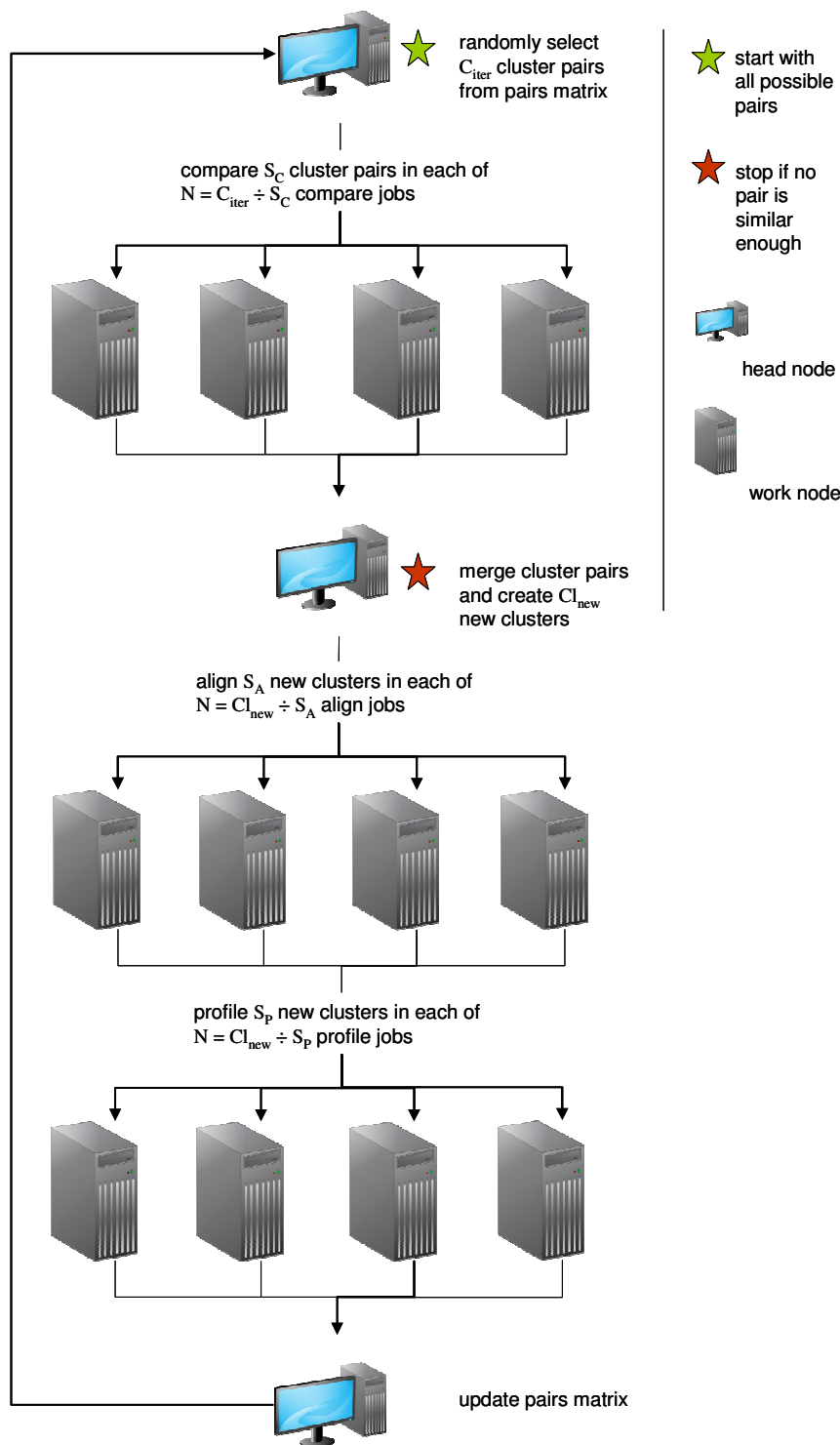
Even with the heuristics described in Section 2.2.3, the first execution of GeMMA on a large sequence dataset can still involve hundreds of millions of cluster comparisons. On a single standard desktop PC, the clustering process takes hours to days for datasets up to  $\sim 10,000$  sequences and weeks or months for larger sets. Therefore, in addition to implementing the heuristics described above, GeMMA was designed as a distributed HPC protocol. More specifically, the sequence alignment, profile generation and profile comparison steps (as the major speed bottlenecks) were made distributed tasks. In each step, the overall workload is distributed evenly among a number of work nodes (see Figure A.1).

## **A.1 Challenges**

The HPC implementation of GeMMA posed different challenges on the technical level, mainly due to the iterative nature of the protocol. In particular, all cluster comparisons carried out in a given iteration have to finish before any merging can take place, and vice versa. The GeMMA master script therefore has to run on the head node (the node the user can login to and submit jobs) for the time of (at least) an individual GeMMA round; this can mean minutes up to weeks. This is a problem since HPC systems in the scientific field are usually shared resources. On such systems, the execution of user tasks on the head node is normally deprecated. The head node has to run a multitude of persistent tasks related to job scheduling and user account control. Consequently, both CPU and memory usage by user tasks have to be kept to a minimum. For purely serial HPC workflows, this is normally not a hindrance: the head node is used for job submission and collection of results only, often on a one-off basis. Examples would be comparing a large set of sequences or carrying out a large number of independent mathematical calculations.

Based on the constraints of shared HPC systems outlined above, the CPU and/or memory requirements of the GeMMA master script had to be kept to a minimum. In contrast, the respective systems usually impose very liberal (or no) limits on the usage of disk space. Further, the prices for storage media continue to fall (Walter 2005), much more rapidly than RAM prices. Whenever a program requests more than the available amount of physical memory, modern operating systems automatically cache data structures in files, that is, they provide ‘virtual’ memory. However, this swapping process can severely slow down memory-intensive tasks. For this reason, it was important to follow an approach that avoids the holding or sorting of large data structures in memory from the outset.

The two largest persistent data structures in the case of GeMMA are the pairs matrix (capturing which pairs of clusters have and which have not yet been compared) and the results matrix (storing all so-far produced cluster similarity values). While the pairs matrix shrinks as the clustering process proceeds, the results matrix grows. Whenever clusters are merged, any entries in these matrices that relate to one or both of the merged (no longer existent) partners are removed. GeMMA implements memory-efficient storage and updating strategies for both data structures. This is explained in detail in the following two sections.



**Figure A.1. The HPC implementation of GeMMA.** A GeMMA round starts with the first iteration, at the point indicated by the green star. The number of cluster pairs to compare  $C_{iter}$  (see Section 2.2.3.2) and the job sizes  $S_C$ ,  $S_A$  and  $S_P$  are dynamically calculated in each iteration, respectively. The number of newly created clusters  $C_{new}$  depends on the number of merges made. Note that after round termination (red star), GeMMA can be executed on the set of remaining clusters, using a lower similarity threshold value. Lowering the threshold gradually over several rounds of GeMMA is important for the comparison sampling heuristic described in Section 2.2.3.2 to work. The graphics are taken from Creative Commons.

## A.2 The pairs matrix

When GeMMA is first executed on a given set of starting clusters it assigns a unique cluster number to each cluster, starting from one. The program then generates a sparse (symmetrical) matrix of all possible cluster pairs. It holds only a single bit of information for each cluster pair, indicating whether the pair has already been compared or not. It can therefore be kept in relatively little memory. For each cluster (matrix row) a bit vector is generated that holds the information about all possible comparisons of this cluster with any other cluster (matrix columns). The ‘raw’ memory requirement of this data structure in bytes  $B$ , depending on the number of initial clusters  $N$ , and disregarding any additional overhead produced by the interpreter, is therefore given by:

$$B = \frac{N \cdot (N - 1)}{2 \cdot 8}$$

Initially, all matrix fields are set to ‘false’. At the start of each GeMMA iteration, following the workflow in Figure A.1,  $C_{\text{iter}}$  cluster pairs are randomly selected from all pairs that have not been compared yet ( $C_{\text{left}}$ ). This translates to randomly indexing the pairs matrix until  $C_{\text{iter}}$  pairs with a ‘false’ value have been found. The lower of the two cluster numbers indexes the list of bit strings (matrix rows). The higher number indexes the bit number (matrix column) in the respective string. If  $C_{\text{left}}$  does not exceed  $C_{\text{iter}}$ , randomisation is not necessary and all ‘false’ entries of the pairs matrix are selected.

According to the calculated settings for the compare job size  $S_C$  and the number of jobs  $N$  (see Section A.4), the subset of  $C_{\text{iter}}$  comparisons is then split into equally sized parts, to be processed in a distributed manner. Subsequently, a number of cluster pairs are merged based on their calculated similarities. The pairs matrix is then updated as follows. First, for all pairs that have been compared, the respective matrix fields are set to ‘true’. Second, whenever two clusters are merged their matrix rows are deleted (the two bit



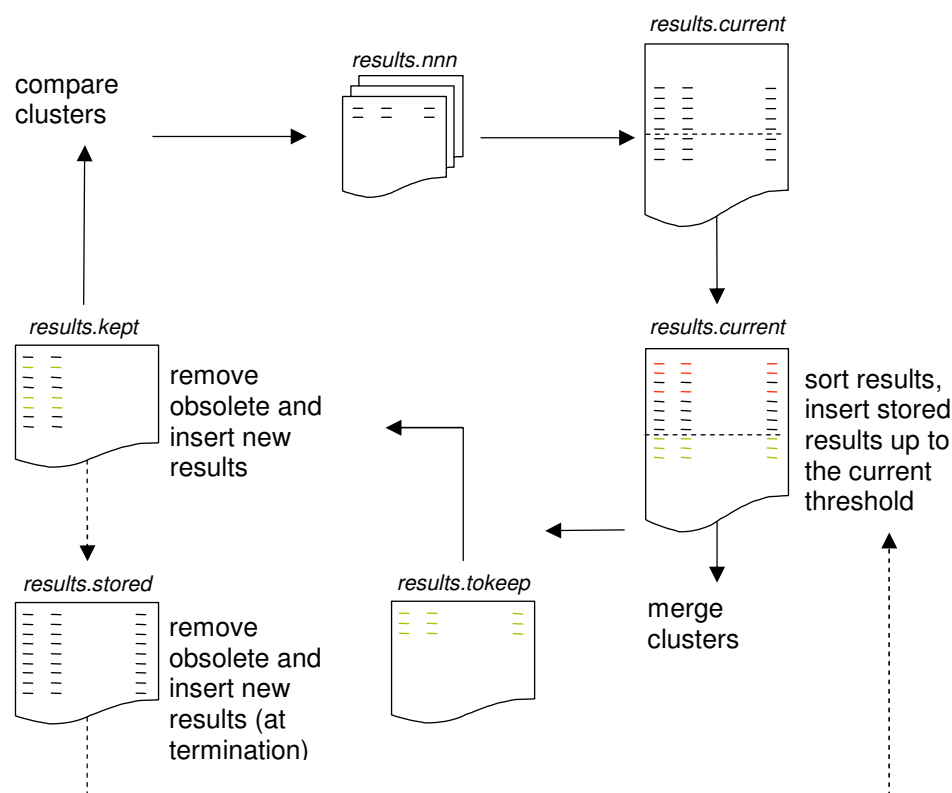
strings are set to be empty) and their columns (the two respective positions in all other bit strings) are set to ‘true’. Third, each new cluster that is produced by merging two old clusters is assigned a unique cluster number (the highest existing cluster number incremented by one). For each new cluster, a new row (bit string) and a new column (position in all other bit strings) are added to the matrix, with all fields set to ‘false’, respectively.

Further to reduce the memory footprint of the bit matrix used, an internal offset is subtracted from all cluster numbers when indexing the matrix. This is the number of the cluster with the lowest cluster number that exists at any point in the clustering process, respectively. This offsetting is made possible by the fact that cluster numbers are never reused, that is, the numbers of newly created clusters are always higher than those of clusters created earlier in the process.

### **A.3 The results matrix**

In traditional hierarchical clustering approaches, for example using average linkage, the pair-wise similarities between all data points are calculated prior to clustering and kept in memory throughout the whole process. Clusters are compared based on the pre-calculated similarities between the data points they contain. In contrast, the GeMMA protocol continuously produces new cluster profiles to be compared, while old ones become obsolete. Further, sequence datasets are clustered with GeMMA in several consecutive rounds, decreasing the cluster similarity threshold value after each round (see Section 2.2.3.2).

The strategy used to store cluster comparison results in GeMMA is based on two rules. First, any comparison results that do not meet the threshold value set in a given round should be stored to avoid re-calculation in following rounds, where a lower threshold value is set. Second, it would be inefficient and should therefore be avoided to keep comparison results for clusters that have already undergone merging (and thus no longer exist).



**Figure A.2 The life-cycle of the results file(s).** This flowchart illustrates how HT-GeMMA stores and updates the cluster similarity matrix. Once all cluster comparisons for a given iteration (top, left) have been completed, the results are collected from the individual results files, to be stored and sorted in *results.current* in order of decreasing similarity. If any stored results from prior HT-GeMMA rounds (with a higher cluster similarity threshold value) are found, those which meet the current threshold value are merged into *results.current*. This list is then traversed top-down and all similar enough cluster pairs are merged into new clusters. Any results not meeting the current threshold value (as indicated by the dashed line) are intermediately stored in *results.tokeep*, which is then merged with *results.kept*. The latter is initially empty and grows with each iteration; obsolete results are constantly removed while the order of decreasing similarity is upheld. At program termination *results.kept* is merged with *results.stored* (bottom, left).

For memory efficiency, the GeMMA results matrix was implemented in a file-based manner (see Figure A.2). Accordingly, the strategies for updating the matrix had to be optimised for speed, that is, for minimising disk I/O. Most importantly, whenever clusters are merged (insertion of new, and deletion of obsolete, results), the top-down sorted order of results is maintained in the respective files. In this manner, the merging process can be implemented as a simple traversal through the list of current results: any pair more similar than the cluster similarity threshold value set is merged, until either an insufficiently similar pair or the end of the list is reached. Note that constantly maintaining the sorted order of results is more time-efficient than repeatedly sorting the

respective (large) files from scratch. Figure A.2 illustrates in detail how GeMMA stores results within individual iterations (*results.current*), within individual rounds (*results.kept*) and between subsequent rounds (*results.stored*).

#### A.4 Resource utilisation

Apart from making the GeMMA HPC implementation memory-efficient, another aim was to optimise HPC resource utilisation. In a typical large-scale, shared HPC system the job queuing systems often have to handle tens of thousands of jobs simultaneously, assigned to hundreds of different users. For this scheduling to work efficiently, users have to provide a maximum wall time setting for each job, that is, the maximum time a job is estimated to take until completion. Correctly setting this parameter is important for two reasons. First, shorter wall time settings often make jobs start earlier. Second, and more importantly, the scheduler terminates any job that exceeds its wall time limit. In general, submitting very small (quickly finishing) or very large (time-intensive) jobs is not considered ‘good practice’ on HPC systems. Small jobs can create considerable overhead, because the scheduling process can take more time than the job itself takes to finish. Large jobs tend to block the shared HPC resources for too long and are thus ‘penalised’ by the scheduler. This means that it commonly takes a long time before such jobs are submitted.

For GeMMA jobs, a relatively stable and predictable wall time is desired both for performance and monitoring reasons. In the interests of maximising utilisation and avoiding job loss, the job size  $S$  for all job types is dynamically calculated in each GeMMA iteration, while the wall time setting is kept constant. The formula determining  $S$  is:

$$S = \frac{S_{\max}}{L}$$

$S_{\max}$  is the maximum job size and  $L$  is the number of sequences found in the largest existing cluster. With increasing  $L$ ,  $S$  decreases.  $S$  is further kept within fixed upper and lower boundaries, currently set to 10,000 ( $S_{\max}$ ) and 10, respectively. These values are chosen according to the processing power (speed) of the individual work nodes and the desired range of job runtimes. To ensure a good utilisation and fair sharing in the case of shared HPC resources, GeMMA is set up by default to generate jobs that do not take less than five or more than 120 minutes. The dynamic calculation of  $S$  is to ensure that all jobs fall in this range.

To balance the workload evenly among all jobs within a given GeMMA iteration it is not sufficient for each job to have the same size. The latter refers to the number of individual instances of the same task type in a single job, for example, the number of pair-wise cluster comparisons. Rather, the size of the input data for each instance has to be taken into account as well. When processing sequence superfamilies with GeMMA (see Chapter 3), the clusters are empirically found to show a scale-free size distribution in late stages of clustering. Larger clusters take longer to align, and larger alignments lead to increased profile generation times. Further, longer profiles take longer to compare than shorter ones. It is therefore not only to provide a representative sampling of comparisons when the pairs matrix is assessed randomly (see Section 2.2.3.2) but also to distribute the workload evenly among individual HPC jobs.

## **A.5 Job monitoring and rescue**

There are two main sources of potential errors or inconsistencies during a GeMMA execution. First, the somewhat ‘fragile’ character of HPC systems in general, primarily on the hardware end. In most situations, problems emerging from this can be detected and rectified through constant job monitoring. Second, the complexity that is generated when concurrent instances of

GeMMA that cluster different sequence datasets are run concurrently on the same HPC system. This parallel strategy is generally advisable, since it maximises HPC utilisation and therefore leads to an overall speed gain. The problems it could create are avoided by a rigorous job naming scheme.

The most frequent causes of job loss (premature termination) and failure (the production of erroneous output) on HPC systems are hardware related. In particular, this refers to (i) work nodes being shut down or rebooted and (ii) problems with the shared or local storage systems. The iterative workflow of GeMMA could potentially come to an indefinite halt in the case of such events. The GeMMA master script that runs on the head node therefore periodically checks the numbers and identifiers of any already finished jobs. At the same time, it checks how many and which jobs are still running. Whenever jobs that have not finished are also not running, this indicates job loss. In this case, the ‘missing’ jobs get resubmitted.

The introduction of unique instance and job identifiers for all jobs, in the form of a composite job name, was necessary to be able to run multiple GeMMA instances in parallel on the same HPC system. In this way, each instance can unambiguously identify its daughter jobs and monitor their progress. The number of sequence datasets that can efficiently be processed in parallel depends on their size and the overall utilisation of the HPC system in use. However, the parallel strategy is generally advisable. This is due to the hybrid (partly serial and partly parallel) nature of the GeMMA workflow: while one GeMMA instance is busy with sorting comparison results and merging clusters on the head node, another instance can run jobs on any available work nodes.

## Appendix B – Superfamily studies 1990-2010

Author(s)	Year	Title	Journal
H. R. Bourne, D. A. Sanders and F. McCormick	1990	<i>The GTPase superfamily</i>	Nature
J. Downward	1990	<i>The ras superfamily of small GTP-binding proteins</i>	Trends Biochem Sci
W. A. Hide, L. Chan and W. H. Li	1992	<i>Structure and evolution of the lipase superfamily</i>	J Lipid Res
V. Laudet, D. Stehelin and H. Clevers	1993	<i>Ancestry and diversity of the HMG box superfamily</i>	Nucleic Acids Res
E. V. Koonin, A. R. Mushegian, R. L. Tatusov, S. F. Altschul, S. H. Bryant, P. Bork and A. Valencia	1994	<i>Eukaryotic translation elongation factor 1 gamma contains a glutathione transferase domain-study of a diverse, ancient protein superfamily using motif search and structural modelling</i>	Protein Sci
B. Henrissat and A. Romeu	1995	<i>Families, superfamilies and subfamilies of glycosyl hydrolases</i>	Biochem J
H. Mellor and P. J. Parker	1998	<i>The extended protein kinase C superfamily</i>	Biochem J
K. Sjolander	1998	<i>Phylogenetic inference in protein superfamilies</i>	In Silico Biol
T. W. Grebe and J. B. Stock	1999	<i>The histidine protein kinase superfamily</i>	Adv Microb Physiol
V. Hwa, Y. Oh and R. G. Rosenfeld	1999	<i>The insulin-like growth factor-binding protein (IGFBP) superfamily</i>	Endocr Rev
L. Aravind	1999	<i>An evolutionary classification of the metallo-beta-lactamase fold proteins</i>	In Silico Biol
D. A. Six and E. A. Dennis	2000	<i>The expanding superfamily of phospholipase A(2) enzymes</i>	Biochim Biophys Acta
J. Adams, R. Kelso and L. Cooley	2000	<i>The kelch repeat superfamily of proteins</i>	Biol Direct
J. A. Irving, R. N. Pike, A. M. Lesk and J. C. Whisstock	2000	<i>Phylogeny of the serpin superfamily</i>	Genome Res
F. Nollet, P. Kools and F. van Roy	2000	<i>Phylogenetic analysis of the cadherin superfamily allows identification of six major</i>	J Mol Biol

Author(s)	Year	Title	Journal
		<i>subfamilies besides several solitary members</i>	
H. T. Idriss and J. H. Naismith	2000	<i>TNF alpha and the TNF receptor superfamily</i>	Microsc Res Tech
Y. Z. Gu, J. B. Hogenesch and C. A. Bradfield	2000	<i>The PAS superfamily</i>	Proteins
J. A. Gerlt and P. C. Babbitt	2001	<i>Divergent evolution of enzymatic function</i>	Annu Rev Biochem
T. Ogura and A. J. Wilkinson	2001	<i>AAA+ superfamily ATPases</i>	Genes Cells
H. C. Pace and C. Brenner	2001	<i>The nitrilase superfamily</i>	Genome Biol
S. Khuri, F. T. Bakker and J. M. Dunwell	2001	<i>Phylogeny, function, and evolution of the cupins, a structurally conserved, functionally diverse superfamily of proteins</i>	Mol Biol Evol
F. Horn, G. Vriend and F. E. Cohen	2001	<i>Collecting and harvesting biological data</i>	Nucleic Acids Res
H. J. Sofia, G. Chen, B. G. Hetzler, J. F. Reyes-Spindola and N. E. Miller	2001	<i>Radical SAM, a novel protein superfamily linking unresolved steps in familiar biosynthetic pathways with radical mechanisms</i>	Nucleic Acids Res
R. F. Thompson and G. M. Langford	2002	<i>Myosin superfamily evolutionary history</i>	Anat Rec
K. Truong and M. Ikura	2002	<i>Identification and characterization of subfamily-specific signatures in a large protein superfamily by a hidden Markov model approach</i>	BMC Bioinformatics
S. Cheek, H. Zhang and N. V. Grishin	2002	<i>Sequence and structure classification of kinases</i>	J Mol Biol
M. A. Nieto	2002	<i>The snail superfamily of zinc-finger transcription factors</i>	Nat Rev Mol Cell Biol
H. Riveros-Rosas, A. Julian-Sanchez, R. Villalobos-Molina, J. P. Pardo and	2003	<i>Diversity, taxonomy and evolution of medium-chain dehydrogenase/reductase superfamily</i>	Eur J Biochem



Author(s)	Year	Title	Journal
E. Pina			
E. M. Hrabak, C. W. Chan, M. Gribskov, J. F. Harper, J. H. Choi, N. Halford, J. Kudla, S. Luan, H. G. Nimmo, M. R. Sussman, M. Thomas, K. Walker-Simmons, J. K. Zhu and A. C. Harmon	2003	<i>The Arabidopsis CDPK-SnRK superfamily of protein kinases</i>	Plant Physiol
J. C. Ame, C. Spenlehauer and G. de Murcia	2004	<i>The PARP superfamily</i>	Bioessays
A. D. Santos, J. M. McIntosh, D. R. Hillyard, L. J. Cruz and B. M. Olivera	2004	<i>The A-superfamily of conotoxins</i>	J Biol Chem
L. M. Iyer, D. D. Leipe, E. V. Koonin and L. Aravind	2004	<i>Evolutionary history and higher order classification of AAA+ ATPases</i>	J Struct Biol
D. A. Shagin, E. V. Barsova, Y. G. Yanushevich, A. F. Fradkov, K. A. Lukyanov, Y. A. Labas, T. N. Semenova, J. A. Ugalde, A. Meyers, J. M. Nunez, E. A. Widder, S. A. Lukyanov and M. V. Matz	2004	<i>GFP-like proteins as ubiquitous metazoan superfamily</i>	Mol Biol Evol
G. J. Praefcke and H. T. McMahon	2004	<i>The dynamin superfamily</i>	Nat Rev Mol Cell Biol

Author(s)	Year	Title	Journal
J. Colicelli	2004	<i>Human RAS superfamily proteins and related GTPases</i>	Sci STKE
M. W. Vetting, S. d. C. LP, M. Yu, S. S. Hegde, S. Magnet, S. L. Roderick and J. S. Blanchard	2005	<i>Structure and functions of the GNAT superfamily of acetyltransferases</i>	Arch Biochem Biophys
Y. Zhang and L. Wang	2005	<i>The WRKY transcription factor superfamily</i>	BMC Evol Biol
A. N. Zelensky and J. E. Gready	2005	<i>The C-type lectin-like domain superfamily</i>	Febs J
S. C. Dillon, X. Zhang, R. C. Trievel and X. Cheng	2005	<i>The SET-domain protein superfamily</i>	Genome Biol
R. Bhadra, N. Srinivasan and S. B. Pandit	2005	<i>A new domain family in the superfamily of alkaline phosphatases</i>	In Silico Biol
K. Wennerberg, K. L. Rossman and C. J. Der	2005	<i>The Ras superfamily at a glance</i>	J Cell Sci
L. M. Iyer, E. V. Koonin, D. D. Leipe and L. Aravind	2005	<i>Origin and evolution of the archaeo-eukaryotic primase superfamily and related palm-domain proteins</i>	Nucleic Acids Res
H. Miki, Y. Okada and N. Hirokawa	2005	<i>Analysis of the kinesin superfamily</i>	Trends Cell Biol
L. M. Iyer, A. M. Burroughs and L. Aravind	2006	<i>The ASCH superfamily</i>	Bioinformatics
S. Dunin-Horkawicz, M. Feder and J. M. Bujnicki	2006	<i>Phylogenomic analysis of the GIY-YIG nuclease superfamily</i>	BMC Genomics

Author(s)	Year	Title	Journal
A. G. McLennan	2006	<i>The Nudix hydrolase superfamily</i>	Cell Mol Life Sci
J. Fan, J. Lefebvre and P. Manjunath	2006	<i>Bovine seminal plasma proteins and their relatives</i>	Gene
A. M. Burroughs, K. N. Allen, D. Dunaway-Mariano and L. Aravind	2006	<i>Evolutionary genomics of the HAD superfamily</i>	J Mol Biol
M. Ammelburg, T. Frickey and A. N. Lupas	2006	<i>Classification of AAA+ proteins</i>	J Struct Biol
M. Novinec, D. Kordis, V. Turk and B. Lenarcic	2006	<i>Diversity and evolution of the thyroglobulin type-1 domain superfamily</i>	Mol Biol Evol
H. H. Park, Y. C. Lo, S. C. Lin, L. Wang, J. K. Yang and H. Wu	2007	<i>The death domain superfamily in intracellular signaling of apoptosis and inflammation</i>	Annu Rev Immunol
A. M. Burroughs, S. Balaji, L. M. Iyer and L. Aravind	2007	<i>A novel superfamily containing the beta-grasp fold involved in binding diverse soluble ligands</i>	Biol Direct
S. Ojha, E. C. Meng and P. C. Babbitt	2007	<i>Evolution of function in the "two dinucleotide binding domains" flavoproteins</i>	PLoS Comput Biol
D. J. Rigden	2008	<i>The histidine phosphatase superfamily</i>	Biochem J
T. T. Nguyen, S. Brown, A. A. Fedorov, E. V. Fedorov, P. C. Babbitt, S. C. Almo and F. M. Raushel	2008	<i>At the periphery of the amidohydrolase superfamily</i>	Biochemistry
O. Hadjebi, E. Casas-Terradellas, F. R. Garcia-Gonzalo and J. L. Rosa	2008	<i>The RCC1 superfamily</i>	Biochim Biophys Acta

Author(s)	Year	Title	Journal
G. M. Gibbs, K. Roelants and M. K. O'Bryan	2008	<i>The CAP superfamily</i>	Endocr Rev
B. H. Dessailly, A. J. Reid, C. Yeats, J. G. Lees, A. Cuff and C. A. Orengo	2009	<i>The evolution of protein functions and networks</i>	Biochem Soc Trans
J. F. Rakus, C. Kalyanaraman, A. A. Fedorov, E. V. Fedorov, F. P. Mills-Groninger, R. Toro, J. Bonanno, K. Bain, J. M. Sauder, S. K. Burley, S. C. Almo, M. P. Jacobson and J. A. Gerlt	2009	<i>Computation-facilitated assignment of the function in the enolase superfamily</i>	Biochemistry
D. Kordis and V. Turk	2009	<i>Phylogenomic analysis of the cystatin superfamily in eukaryotes and prokaryotes</i>	BMC Evol Biol
Y. Yin, J. Huang and Y. Xu	2009	<i>The cellulose synthase superfamily in fully sequenced plants and algae</i>	BMC Plant Biol
J. Hedlund, J. Johansson and B. Persson	2009	<i>BRICHOS - a superfamily of multidomain proteins with diverse functions</i>	BMC Res Notes
S. D. Copley	2009	<i>Prediction of function in protein superfamilies</i>	F1000 Biol Rep
C. Gao and B. Han	2009	<i>Evolutionary and expression study of the aldehyde dehydrogenase (ALDH) gene superfamily in rice (<i>Oryza sativa</i>)</i>	Gene
H. S. Fernando, H. G. Kynaston and W. G. Jiang	2009	<i>WASP and WAVE proteins</i>	Int J Mol Med
J. Bains, R. Leon and M. J. Boulanger	2009	<i>Structural and biophysical characterization of BoxC from <i>Burkholderia xenovorans</i> LB400</i>	J Biol Chem

Author(s)	Year	Title	Journal
D. Gangwar, M. K. Kalita, D. Gupta, V. S. Chauhan and A. Mohmmed	2009	<i>A systematic classification of Plasmodium falciparum P-loop NTPases</i>	Malar J
H. J. Atkinson and P. C. Babbitt	2009	<i>An atlas of the thioredoxin fold class reveals the complexity of function-enabling adaptations</i>	PLoS Comput Biol
A. Garza-Garcia, R. Harris, D. Esposito, P. B. Gates and P. C. Driscoll	2009	<i>Solution structure and phylogenetics of Prod1, a member of the three-finger protein superfamily implicated in salamander limb regeneration</i>	PLoS One
E. A. Romanel, C. G. Schrago, R. M. Counago, C. A. Russo and M. Alves-Ferreira	2009	<i>Evolution of the B3 DNA binding superfamily</i>	PLoS One
A. M. Burroughs, L. M. Iyer and L. Aravind	2009	<i>Natural history of the E1-like superfamily</i>	Proteins
S. C. Andrews	2010	<i>The Ferritin-like superfamily</i>	Biochim Biophys Acta
R. C. de Melo-Minardi, K. Bastard and F. Artiguenave	2010	<i>Identification of subfamily-specific sites based on active sites modeling and clustering</i>	Bioinformatics
J. Engelken, H. Brinkmann and I. Adamska	2010	<i>Taxonomic distribution and origins of the extended LHC (light-harvesting complex) antenna protein superfamily</i>	BMC Evol Biol
J. C. Jimenez-Lopez, E. W. Gachomo, M. J. Seufferheld and S. O. Kotchoni	2010	<i>The maize ALDH protein superfamily</i>	BMC Struct Biol
Y. Martinez de la Torre, M. Fabbri, S. Jaillon, A. Bastone, M. Nebuloni, A.	2010	<i>Evolution of the pentraxin family</i>	J Immunol

Author(s)	Year	Title	Journal
Vecchi, A. Mantovani and C. Garlanda			
J. J. Liu, R. Sturrock and A. K. Ekramoddoullah	2010	<i>The superfamily of thaumatin-like proteins</i>	Plant Cell Rep
J. T. Bridgham, G. N. Eick, C. Larroux, K. Deshpande, M. J. Harms, M. E. Gauthier, E. A. Ortlund, B. M. Degnan and J. W. Thornton	2010	<i>Protein evolution by molecular tinkering</i>	PLoS Biol
S. O. Kotchoni, J. C. Jimenez-Lopez, D. Gao, V. Edwards, E. W. Gachomo, V. M. Margam and M. J. Seufferheld	2010	<i>Modeling-dependent protein characterization of the rice aldehyde dehydrogenase (ALDH) superfamily reveals distinct functional and structural features</i>	PLoS One
B. H. Dessailly, O. C. Redfern, A. L. Cuff and C. A. Orengo	2010	<i>Detailed analysis of function divergence in a large and diverse domain superfamily</i>	Structure