Graph-Based Methods for Large-Scale Protein Classification and Orthology Inference

Thesis committee

Thesis supervisor
Prof.dr. Jack A.M. Leunissen
Professor of Bioinformatics
Wageningen University

Thesis co-supervisors
Dr. Roeland C.H.J. van Ham
Associate professor at the Laboratory of Bioinformatics
Wageningen University

Prof.dr. Sándor Pongor
Group leader at the International Centre for Genetic Engineering and Biotechnology
Trieste, Italy

Other members
Prof.dr. Ton A.H.J. Bisseling, Wageningen University
Dr. Freek T. Bakker, Wageningen University
Prof.dr. Wilfried W. de Jong, Radboud University Nijmegen
Prof.dr. Martijn Huynen, Radboud University Nijmegen

This research was conducted under the auspices of the Graduate School of Experimental Plant Sciences.

Graph-Based Methods for Large-Scale Protein Classification and Orthology Inference

Arnold Kuźniar

Thesis
submitted in partial fulfilment of the requirements for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. dr. M.J. Kropff,
in the presence of the
Thesis Committee appointed by the Doctorate Board
to be defended in public on Friday 6 November 2009
at 11 AM in the Aula.

Arnold Kuźniar
Graph-Based Methods for Large-Scale Protein Classification and Orthology Inference

Thesis, Wageningen University
With summaries in English and Dutch
Wageningen, the Netherlands (2009)

ISBN 978-90-8585-501-9

To my dear parents, mother Helena and father Jozef

Srdečne venujem mojim drahím rodičom

Sok szeretettel drága szüleimnek

CONTENTS

CHAPTER 1

INTRODUCTION

Proteins are complex biomolecules indispensable for life. They have evolved many different structures and functions to carry out collectively the biological processes within living cells. The diversity of proteins would not have arisen without the molecular changes that occur in genomes during the course of evolution. Some changes, however, are deleterious for protein function, and thereby cause, amongst others, genetic diseases in humans. The quest for understanding how proteins evolve and function has therefore been a prominent and costly human endeavor. With advances in genomics and use of bioinformatics tools, the diversity of proteins in present day genomes can now be studied more efficiently than ever before.

Evolutionary (phylogenetic) concepts have been instrumental in studying subjects as diverse as the diversity of genomes, cellular networks, protein structures and functions, and functional genome annotation. In particular, the detection of orthologous proteins (or genes) across genomes have provided reliable means to infer biological functions and processes from one organism to another, and hence have enabled, for example, human genetic diseases to be studied in model organisms such as yeast, fruit fly, worm or mouse. Despite this progress, fully automated phylogenomic pipelines scalable to the hundreds of genomes (or proteomes) currently available have been an elusive goal of comparative genomics. Such methods are however indispensable for reliable high-throughput functional genome annotation, phylogenetic inferences as well as for maintaining high quality protein and family databases.

First, we introduce the biological context of protein structure, function and evolution, review the state-of-the-art sequence-based protein classification (clustering) methods, and then describe computational methods used for cluster validation. Finally, we present the outline and objectives of this thesis.

1.1 From genes to proteomes

Nucleotide sequences of genes or entire genomes are technically much easier and faster to obtain than the amino acid sequences of gene products, the proteins (further on the terms such as gene and protein will be used interchangeably). Most protein sequences are deduced from the corresponding coding genes *in silico* rather than from direct chemical sequencing of proteins. New genomic sequences are deposited in public bioinformatics databases such as the EMBL/GenBank/DDBJ nucleic acid database (Benson et al., 2009; Cochrane et al., 2009; Sugawara et al., 2009), which is used for most (if not all) downstream genome analyses as well as for constructing other, derived databases. Examples of the latter include the non-curated section of the UniProt database (Consortium, 2009), namely UniProtKB/TrEMBL, which is constructed primarily by translating all the coding sequences of the EMBL/GenBank/DDBJ

database into protein sequences. Databases such as Entrez Gene (Maglott et al., 2005), International Protein Index (IPI) (Kersey et al., 2004), Integr8 (Kersey et al., 2005), Genome Reviews (Sterk et al., 2007) and Ensembl (Hubbard et al., 2009) provide comprehensive collections of genes and/or proteins for fully sequenced genomes. Most of these genome database also involve manual curation by experts; however, the experimental validation of protein sequences, namely those expressed at protein level, has been done for only 1% of the total number of predicted proteins present in UniProt. As the number of published genomes exceeds 1000 (see the Genomes On Line Database (GOLD) (Liolios et al., 2008), there is a need for reliable and scalable algorithms that make biologically sound predictions of protein function, 3D structure and evolution for many poorly annotated genomes. The development of such bioinformatics methods requires involvement of multiple disciplines such as molecular and evolutionary biology, mathematics, statistics, and computer science.

1.2 Protein structure and function

From the standard 20 letter amino acid (aa) alphabet, one can assemble an astronomically large number of protein sequences; for example, there are 20^{100} different protein sequences of 100aa possible. However, only a tiny fraction of all those combinations have been "tried and selected" in the course of evolution, and have evolved into proteins of biological importance. For example, the shortest biologically meaningful amino acid sequence is that of the cuttlefish neuropeptide (2aa), whereas the longest is that of the mouse titin (35,213aa) (taken from UniProtKB/SwissProt).

Proteins have complex structures, which are commonly described using the following structural hierarchy (IUBMB, 1992). The primary structure refers to the sequence of amino acid residues in the polypeptide chain while the secondary structure refers to the spatial arrangement of short segments of the polypeptide chain, which give rise to structures such as the alpha-helix, beta-sheet, beta-turns and loops. The tertiary or three-dimensional (3D) structure of a protein molecule (or a subunit thereof) is the arrangement of all its atoms in space but without regard to its relationships with neighboring molecules or subunits. Finally, the quaternary structure refers to the spatial organization of multiple protein molecules into a multi-subunit complex (such as the alpha and beta subunits of the hemolglobin molecule). The connection between amino acid sequence and the biologically active conformation of a protein was first described by Anfinsen, who postulated that the primary structure of a protein dictates how the protein folds into a specific 3D structure (Anfinsen, 1973). In the Protein Data Bank (PDB) (Berman et al., 2009) there are about 60,000 proteins whose 3D structures have been determined experimentally using X-ray crystallography and/or nuclear magnetic resonance experiments (NMR).

Additional terms such as motif, domain or fold are also used to describe protein structure and function. For example, a motif is a specific combination of secondary structures (e.g., helix-loop-helix motif), which might have a particular biological function (e.g., calcium binding), whereas a domain is a compact structural (Richardson, 1981; Wetlaufer, 1973), functional and evolutionary unit (Bork, 1991; Thornton et al.,

1999) that combines with other domains in multi-domain or chimeric proteins. Although each protein has a unique 3D structure, many proteins can share the same structural fold. The estimates of the total number of protein folds vary from 1000 to 10,000, depending on the methods used (Grant et al., 2004). With the increasing repertoire of known protein structures in the PDB database, many examples provide evidence that protein folds exhibit plasticity and robustness in the course of evolution (Kinch and Grishin, 2002). Some protein folds and superfamilies are associated with many functions and structurally diverse ligands, whereas others are conserved both in structure and function (Todd et al., 2001). Therefore the 'one protein – one function' paradigm has become challenged by a 'one protein – multiple functions' view (Nobeli et al., 2009). For example, some (chimeric) proteins such as the yeast's pentafunctional enzyme catalyzes multiple reactions of the shikimate pathway.

1.3 Molecular evolution of proteins

The study of molecular evolution is an inquiry into the processes that cause changes in genetic material such as DNA, RNA or proteins in the course of evolution. Several, sometimes conflicting theories have been proposed to explain genetic variability and biological consequences of mutations (Bernardi, 2007). Mutations are considered to be the driving force of evolution – wherein less favorable or deleterious mutations are removed from the gene pool by negative (purifying) natural selection while more favorable or beneficial ones are fixed by positive (adaptive) selection in the population.

Theories of molecular evolution

In the early 80's, Kimura's revolutionary proposal – the neutral theory of molecular evolution – started an intensive debate on the role of chance (genetic drift) in evolution. The theory stated that a substantial fraction of mutations are selectively neutral (or silent) without a significant biological effect on the organism's fitness. Therefore a much smaller role than previously proposed was attributed to natural selection in evolution, yet the role of purifying selection, which eliminates the majority of new mutations, was acknowledged (Kimura, 1983). Further modification to the theory was made by Ohta, who introduced nearly neutral mutations or intermediates between neutral and advantageous, and between neutral and deleterious changes, hence formulating the nearly neutral theory of molecular evolution (Ohta, 1992). Recently a new, neoselectionist theory was proposed to reconcile (nearly) neutral changes with natural selection (Bernardi, 2007).

Molecular clock hypothesis

In early days, only a few small globular proteins such as insulin, globins and cytochrome were available for comparative protein sequence and/or structure analyses. These have provided fundamental insight into protein structure and function, as well as into phylogenetic relatedness of species. Specifically, Emile Zuckerkandl and Linus Pauling conjectured that the number of amino acid or nucleotide substitutions in

hemoglobin genes of distinct species is roughly proportional to the time that passed since the species diverged from a common ancestor (Zuckerkandl and Pauling, 1962). This led to the concept of the 'molecular clock', which revolutionized the way the evolutionary dating of species was carried out using genetic material instead of fossil records (Sarich and Wilson, 1967) and contributed to the emergence of molecular systematics and phylogenetics.

As more molecular sequences have become available, however, several evolutionary studies have shown that this 'clock' does not always tick regularly: the rates of evolution vary across distinct evolutionary lineages, genes families and sites of the same gene (Ayala, 1997; Hasegawa and Kishino, 1989), and the clock may tick faster for structurally and/or functionally less constrained proteins (such as receptor kinases) and slower for highly constrained proteins (such as histones). To address this, molecular models that account for these variations have been developed (Yoder and Yang, 2000). Although the factors that affect the speed and constancy of protein evolution have been investigated intensively (Rocha, 2006), the results have remained controversial (Decottignies et al., 2003; Hirsh and Fraser, 2001). So far, the expression of genes has been found to explain the most significant proportion of the variation in the rates of protein evolution (Drummond et al., 2005).

Protein taxonomy

A framework, similar to the taxonomy (hierarchy) of living organisms, has been proposed for classifying the ever-expanding repertoire of proteins into 'families' and 'superfamilies' (Dayhoff, 1976). Concepts such as superfamily, family and subfamily are used to indicate common evolutionary descent (homology) of members of the same group, as well as to reflect the increasing degree of similarity in sequence, structure and function. In general, members of a family are close homologs sharing high sequence similarity, whereas members of a superfamily are remote homologs sharing low sequence similarity. Moreover, members of a family usually share similar molecular functions (e.g., dehydrogenase) but may vary in finer molecular details such as substrate specificity (e.g., lactate or malate dehydrogenase). As a result, the family can be subdivided into two or more subfamilies with more specific molecular functions.

However, structural and/or functional similarity between proteins can also arise by convergent or parallel evolution other than through common evolutionary descent (Fitch, 2000). Such unrelated (analogous) proteins can serve identical functions in different species; a phenomenon known as non-orthologous gene displacement (Koonin et al., 1996). While protein families and superfamilies are undoubtedly monophyletic (derived from a common ancestor), the monophyly of protein folds, however, remains an issue of debate (Koonin et al., 2002).

The knowledge of protein evolution has been instrumental for predicting the structure and function of uncharacterized proteins using homology-based modeling (Chothia and Lesk, 1986; Rodriguez et al., 1998; Sander and Schneider, 1991). Indeed, a sound phylogenetic classification of homologs is a prerequisite for virtually all types of inferences about protein structure, function and evolution, biochemical pathways, as well as the relationship between genetic change and morphological in-

novation (Thornton and DeSalle, 2000). Homology may result from three distinct processes, namely gene duplication, speciation and horizontal gene transfer (HGT), which yield paralogous, orthologous and xenologous relationships between proteins, respectively. Particularly the distinction between orthologs and paralogs across multiple genomes is central to comparative genomics because orthologs are more likely to retain the same function in different species than paralogs (Tatusov et al., 1997).

Protein family evolution

Gene duplication, gene loss and domain shuffling are important processes that contribute to the expansion and contraction of protein families as well as may lead to complex phylogenetic relationships between proteins. Several models that explain the diversity and evolution of protein families to various extents have been developed. The birth-and-death model (BDM) is the most plausible because it explains the evolution of most protein families (Hughes and Nei, 1989; Nei and Rooney, 2005). In this model new genes arise by gene duplication, of which some remain active in the genome for a long time while others become inactivated/deleted from the genome independently and at random. Moreover, new families arise by random splitting of existing ones. Furthermore, BDM can result in a highly skewed, power-law distribution of protein (or domain) family sizes – wherein there are only a few large families and many small families (Koonin et al., 2002).

Gene and genome duplication

Gene duplication is a key evolutionary process that enables new genes and functions to arise in the course of evolution (Ohno, 1999). Genomic sequences of diverse organisms have provided substantial evidence that gene duplication is prevalent in the evolutionary history of all organisms, and particularly rampant in multicellular eukaryotes (Lynch and Conery, 2003). Without gene duplication the ability of genomes (species) to adapt to changing environments would be severely constrained (Zhang, 2003).

Duplication events occur at two distinct scales: small-scale duplications involve one gene (or part thereof) or several genes, whereas large-scale duplications involve chromosome segments, entire chromosome or a whole genome. In particular, whole genome duplication (WGD) has contributed significantly to the evolution of plants, fungi, as well as to some animal lineages (Vision et al., 2000; Wolfe and Shields, 1997). The molecular mechanisms by which genes duplicate, persist and diverge in a genome are complex. Duplicate genes or paralogs can arise by means of unequal crossing-over, retroposition, WGD, or horizontal gene transfer (HGT), the outcome of which is quite different. The former mechanism results in a duplicated region that subsumes part of a gene, an entire gene, or several genes arranged in tandem. Therefore tandem duplication is an important mechanism to produce multi-domain proteins that consists of multiple copies of the same type of protein fold (e.g., cytoskeletal spectrins) (Sonnenberg et al., 2007). In retroposition a reverse-transcribed messenger RNA (mRNA) is inserted into the genome more or less at random site, hence the duplicate gene

does not link to the original gene. Genes that arise by retroposition usually become unexpressed or function-less genes (pseudogenes) because they often lack regulatory elements needed for transcription. Computational methods that can distinguish between the distinct mechanisms of duplication will therefore be useful for predicting protein functions as well as for reconstructing the evolutionary past of genomes (Durand and Hoberman, 2006). The role of HGT in acquiring new genes and functions by genomes has been reviewed extensively elsewhere (Eisen, 2000; Koonin et al., 2001).

The evolutionary fate of duplicate genes depends on whether the gene duplication is selectively advantageous, deleterious or neutral. The most common scenario is when one copy of the gene changes through mutations into a pseudogene, a process known as pseudogenization (Lynch et al., 2001). For example, over 60% of human olfactory receptor genes have been subject to pseudogenization since the origin of hominoids (Zhang and Firestein, 2002). So, is it then possible for two functionally redundant gene copies to remain active in a genome after a duplication event? Indeed, selection can prevent paralogs to diverge both in sequence and in function (Nei et al., 2000). Another scenario is when each of the duplicate genes adopts only part of the function of their parental gene (Hughes, 1994; Jensen, 1976). This is called a subfunctionalization whereby the functional divergence between paralogs can occur at different levels including gene expression patterns, protein functions or sub-cellular localization (Hittinger and Carroll, 2007; Marques et al., 2008). The most important outcome of gene duplication is when new genes acquire entirely novel functions (neofunctionalization). In this model one copy of the gene preserves the original function while the other evolves a novel one, adaptive function. Specifically, there is first a relaxation of selection on the redundant copy, which becomes free to evolve and by chance may acquire beneficial mutations. The examples include the human eosinophil cationic gene (a member of the RNase A superfamily), which acquired an antibacterial activity that is independent of the ribonuclease activity (Zhang et al., 1998). Most commonly, related rather than entirely novel functions emerge through this process; examples include the human opsin genes (Yokoyama and Yokoyama, 1989) and other G-protein coupled receptors (GPCRs) (Choi and Lahn, 2003), and immunoglobulins (Sumiyama et al., 2002).

Combinatorial game of protein evolution

Whilst small proteins usually consist of single domains, large proteins constitute of multiple domains (Doolittle, 1995; Orengo and Thornton, 2005). By combining or 'shuffling' existing coding genes, exons or conserved domains, new protein structures and functions can arise (Babushok et al., 2007; Bork, 1991). While gene duplication increases the abundance of domains, the combinatorial game of recombination increases the distinct contexts in which a domain can occur (Vogel et al., 2005). Therefore some protein domains are more promiscuous than others, where domain promiscuity refers to the propensity of protein domains to combine with other distinct domains in multi-domain and functional contexts (Marcotte et al., 1999). The individual domains of a multi-domain protein can act either in concert by increasing the protein's functional specificity, or in new contexts by evolving entirely novel functions

within the new structural framework (Bashton and Chothia, 2007).

Additionally, recombination can lead to gene fusions and gene fission, which form composite or chimeric proteins in one organism and two (or more) smaller split proteins in another organism, respectively. Although both events are widespread across all the kingdoms of life, gene fusions are about four times more common than gene fissions (Kummerfeld and Teichmann, 2005). Identifying these events in the genomes available is important particularly for inferring protein functions using the 'guilt by association' (or Rosetta Stone) hypothesis. Accordingly, the physical link between distinct domains of a chimeric protein implies a functional linkage (Enright et al., 1999; Marcotte et al., 1999; Snel et al., 2002).

Several studies have suggested that the increase in domain promiscuity as well as in the likelihood of forming multi-domain proteins relates to the increase in the phenotypical complexity of organisms (Apic et al., 2001; Tordai et al., 2005). A recent study suggested that only a limited repertoire of promiscuous domains contributed significantly to the diversity and plasticity of eukaryotic proteomes in general, and to the evolution of signaling networks in particular (Basu et al., 2008). Sometimes, lineage-specific mechanisms that form chimeric proteins, such as 'exon shuffling', emerge in the course of evolution. Interestingly, the mechanism of exon shuffling has been found exclusively in metazoan species; therefore it has been associated with metazoan radiation (Patthy, 1996).

In higher eukaryotes, individual genes can produce a combinatorial number of transcripts and protein isoforms through alternative splicing of pre-mRNAs. So far, comprehensive analysis of the human transcriptome suggests that up to 94% of human genes undergo alternative splicing (Wang et al., 2008). This genetic mechanism might therefore play a crucial role in providing structural and functional diversity in eukaryotic proteomes (Birzele et al., 2008).

1.4 Comparative proteome analysis

Proteins whose sequences, structures or functions have been characterized through rigorous yet tedious biochemical experiments (further on denoted as known proteins) constitute the biological knowledge of proteins. This knowledge has played an essential role in predicting the structure and function of many uncharacterized proteins by computational methods of comparative protein analysis. The prediction methods include algorithms for protein sequence and/or structure alignments, database searching, phylogenetic tree reconstruction and machine learning. The evolutionary view on alignments has contributed to the development of efficient database search algorithms such as BLAST (Altschul et al., 1990; 1997) that can infer homology reliably from significant sequences similarities, and as such, enabled the structure or function of uncharacterized proteins to be inferred based on a second, known protein. However, homology implies similarity in structure rather than similarity in function; therefore, the concept of orthology (see Glossary) has been used instead for function prediction. Whilst homology can be inferred from sequence similarity alone, orthology is best supported by phylogenetic analysis (Pearson and Sierk, 2005).

Protein alignments and evolution

Comparing protein sequences (or structures) with each other is a key operation in protein analysis, whereby one can learn about the similarities and/or differences in relation to structure, function and evolution. This is done by aligning two or more related protein sequences against each other in a way that one-to-one correspondence is set up between the amino acid residues of the proteins. The underlying assumption is that each column of the true alignment consists of residues which were derived from a common ancestral (homologous) amino acid residue and diverged through accumulation of mutations such as substitutions, insertions or deletions (indels) in the corresponding coding genes. Therefore only a reliable alignment can be used to infer evolutionary events correctly. However, the prediction of protein evolution from an alignment is only meaningful when homologous proteins are used in the alignment. So how does one know *a priori* whether the proteins involved are homologous or not? As homology is a hypothesis about common evolutionary descent, it can only be tested by statistical means, based on sequence and 3D structure alignments and known molecular functions. The homology hypothesis testing will be discussed later in the section *Reliable database searching and homology inferences*.

Constructing a reliable alignment is difficult particularly for distantly related proteins (remote homologs), as these often yield alignments with many gaps (due to indels or inversions), making the homology inference of individual residues problematic. Doolittle called this the 'twilight zone' for sequence alignment, which denotes an area of homologous protein pairs with low sequence identity (20–30%) (Doolittle, 1987). Further, Sander and Schneider introduced a length-dependent threshold curve for significant sequence identity (so called HSSP-curve) to improve upon remote homology detection (Sander and Schneider, 1991). As remote homologs have diverged more in sequence than structure, the structural information can be used to construct more accurate alignments than using sequences alone (Chothia and Lesk, 1986). Such structure-based alignments have been used successfully to infer homology for protein pairs with less than 10% pairwise sequence identity (Valencia et al., 1991). Therefore, the knowledge of protein evolution has been instrumental for developing computational tools that predict the 3D structure and function of proteins using homology-based modeling techniques (Rodriguez et al., 1998).

Alignment algorithms and scoring schemes

Many algorithms for aligning protein sequences and structures have been developed in the last few decades and have been extensively reviewed in the literature (Lassmann and Sonnhammer, 2005; Notredame, 2007; Pearson and Sierk, 2005; Wallace et al., 2005). In general, the algorithms differ in the alignment approach (global *versus* local), computational complexity, as well as in alignment accuracy. Most alignment algorithms rely on the dynamic programming (DP) optimization paradigm whereby the optimal solution to a problem, such as constructing an optimal alignment between two sequences, is divided into overlapping subproblems that are solved optimally in a recursive manner. In principle, there are two fundamentally distinct alignment

approaches: first, the global approach seeks to align the sequences over the entire length, hence it is suitable for closely related sequences with similar lengths. The second, local approach aligns the most similar part(s) or subsequences of the sequences only, and hence is suitable for distantly related sequences with different lengths. In biological terms, the local similarity regions between distantly related sequences are likely to coincide with known motifs or domains. Therefore a global alignment method might fail to identify a significant match between the sequences owing to the local similarity region(s) lost in the background of random residue matches.

Classical pairwise sequence alignment algorithms include the Needleman-Wunsch algorithm (NW) (Needleman and Wunsch, 1970) and the Smith-Waterman algorithm (SW) (Smith and Waterman, 1981), which guarantee to find optimal global and optimal local pairwise sequence alignments, respectively. An optimal alignment is the one that scores the highest of all possible alignments for a given scoring scheme, namely a scoring matrix and gap penalties. However, the highest-scoring alignment is not necessarily the true one; therefore, the choice of a scoring scheme is an important, yet difficult task. Generally, the default scoring (substitution) matrices such as the collection of BLOSUM (Henikoff and Henikoff, 1992) and PAM matrices (Dayhoff et al., 1978), and the gap penalties associated with the matrix used often yield satisfactory alignments. As the rates of protein evolution vary across protein families, alignments might be further improved, for example, by delicate selection of the scoring scheme according the evolutionary distances between proteins being compared.

Similarity searching in large protein databases

The algorithms described above are computationally intensive, in particular for searching large protein sequence databases, such as UniProt or Refseq, in an all-*versus*-all sequence comparison manner. Therefore heuristic algorithms that trade speed for sensitivity, yet can infer sequence homology reliably, have been implemented in programs such as BLAST (Basic Local Alignment Search Tool) (Altschul et al., 1997) and FASTA (Pearson and Lipman, 1988). The idea behind these is to use a fast searching method that can find approximately equal small segments in the two sequences and then extend the segment pairs into high-scoring ones (called HSPs in BLAST terminology) using the SW algorithm. For example, the BLAST algorithm calculates similarity scores only for sequences that are likely to share significant similarity, while FASTA does it for all sequences in the database; therefore, the latter program is slower than the former.

Further improvement in searching times can be achieved by parallelization techniques, which enable the algorithms to be executed simultaneously on multiple computers and/or processors using a local cluster environment or an (inter)national GRID environment such as the Dutch Life Science Grid. In either case, the entire input dataset needs to be split-up into multiple smaller “chunks” prior to parallel execution. Dedicated hardware such as graphics processing units (GPUs) can be used to perform the computations orders of magnitude faster than general purpose processing units (CPUs) owing to their highly parallel architecture. Several vectorized implementations of the SW algorithms have been developed to carry out optimal protein

similarity searching close to the speed of heuristic methods (Farrar, 2007; Rognes and Seeberg, 2000; Szalkowski et al., 2008; Wozniak, 1997).

Reliable database searching and homology inferences

The problem of searching a database for sequence similarities can be defined as follows: given a query protein sequence q and a database D, find those sequences in D which share the highest similarities with q, rank the similarities (matches) according to statistical significance, and then infer homology between proteins using only statistically significant matches. Local alignment algorithms for database searching have been preferred over global methods because the statistics of local sequence similarities are better understood compared to those of global similarity scores. Moreover, the local methods also take the modularity of proteins into account. Furthermore, the underlying assumption of a database search is that the distribution of local alignment scores between two random sequences follows the Gumbel extreme-value distribution whereby the statistical significance of an alignment can be assessed (Karlin and Altschul, 1990).

A reliable statistical estimate for similarity scores is one that can distinguish similarities due to common evolutionary descent (homology) from those that are due to convergent evolution (analogy) or chance. In other words, an alignment that is less likely to occur by chance is more likely to be biologically meaningful. Simple similarity measures such as percent identity are far less reliable than those based the substitution matrices and significance estimates (Altschul, 1991). The significance of pairwise sequence similarities is commonly assessed using expectancy value or E-value, P-value probability score, or Monte-Carlo-based Z-scores (Karlin and Altschul, 1990; Levitt and Gerstein, 1998; Pearson and Lipman, 1988). For example, the calculation of Z-scores involves random shuffling of a query and/or target sequences, followed by comparisons against a random sample of database sequences. Herein it is assumed that the similarity scores of real non-homologous proteins follow the scores of randomly generated sequences. Several studies have suggested the superior performance of the Z-score statistics over E-value when using simulated sequences; however, this has been questioned when using biologically real sequences (Hulsen et al., 2006b). Moreover, it takes much more time to compute Z-scores than E-values, in particular when the SW algorithm is used for sequence comparisons. In summary, efficient database searching relies not only on fast and accurate alignment algorithms but also on reliable statistical estimates for protein similarity scores (Pearson and Sierk, 2005).

Evaluating database search algorithms

Searching a sequence database D with a query sequence q is a binary classification exercise – wherein q is classified as being either homologous or non-homologous to one or more sequences of D. In this two-class prediction, one wants to know how an algorithm performs on a particular benchmark dataset relative to other methods. First, the numerical output of a database search must be converted into the categorical

	Homologous sequences	Non-homologous sequences
Significant score	TP	FP
Non-significant score	FN	TN

Table 1.1. Four possible outcomes of a database search shown in a contingency table.

output. This is done by choosing an appropriate cutoff threshold T, which divides the scores into statistically significant and insignificant ones. Moreover, the meaning of significance in relation to (dis)similarity scores must be clarified prior to such a division because it can depend on the scores used (Z-scores or E-values).

In binary classification the sequences with significant scores are positive instances while the sequences with insignificant scores are negative instances; however, both instances might be classified incorrectly. As a result, the classification has only four outcomes, which can be summarized using a two-by-two contingency table or confusion matrix (Table 1.1).

Specifically, a sequence classified as positive is called a true positive (TP) if it is known to be homologous to q; otherwise it is called a false positive (FP) sequence. Alternatively, a sequence classified as negative is called a true negative (TN) sequence if it is not homologous to q; otherwise it is called a false negative (FN) sequence. Therefore the classification results can be summarized by counting the instances for TP, FP, TN and TP. It is important to note that these statistics depend on the threshold T used, and hence there is generally a trade-off between the number of FP and FN.

Concepts such as sensitivity and specificity are widely used to evaluate programs for database searching. The sensitivity (true positive or recall rate) measures the proportion of actual homologs (positives) classified correctly [TP / (TP + FN)]. Specificity measures the proportion of actual non-homologs (negatives) classified correctly [TN / (TN + FP)]. For database searching, this estimate is, however, unreliable because of the bias towards hight values (close to 100%). The reason for this is that a database consists of much more unrelated than homologous sequences, resulting in a strong bias towards TN compared to FP. Therefore, it is recommended to use another estimate of specificity (precision) [TP / (TP + FP)], which measures the proportion of relevant cases returned (Baldi and Brunak, 2001). Among many other evaluation methods (Bajić, 2000), the Receiver Operating Characteristic (ROC) analysis has been particularly useful for evaluating sequence and structure comparison algorithms (Sonego et al., 2008) as it is both a visual and numeric method. Specifically, the ROC curve depicts the relationship between the true positive rate (sensitivity) and false positive rate (1 - specificity) at different thresholds T, while the area under the curve (AUC) statistics is the probability by which the predictor assigns a higher score to positive rather than to negative instance. Notably, any performance measure that reduces to a single number discards some information when compared to the four statistics, namely TP, FP, TN and FN.

In the past few years, machine learning techniques such as the k-nearest neighbor

classifiers, support vector machines and artificial neural networks have been used with database search algorithms to improve protein classification (Baldi and Brunak, 2001). As new protein sequences accumulate rapidly, it is important to design protein classification algorithms that will make reliable predictions as well as scale to large datasets (Sonego et al., 2007). Cross-validation methods such as holdout, leave-one-out or k-fold have been used to assess the performance, specifically the generalization ability, of a classifier in two distinct scenarios: in the unsupervised scenario, the training and test samples are selected randomly, while in the supervised scenario these are selected according to known classes. The latter approach has been shown to provide more reliable estimates in the protein classification domain (Kertész-Farkas et al., 2008).

1.5 Protein classification and machine learning

Classifying many proteins manually by human experts is a tedious and costly endeavor. One might therefore design an automated classifier that performs satisfactorily on a small but representative training set of known (labeled) classes, and use the classifier for unlabeled datasets without supervision. Alternatively, this can be done the other way around: first, automatically group a large unlabeled dataset based on (dis)similarities between data points without supervision, and then label/refine the resulting clusters using human expertise. In principle, the classifiers used for protein classification can be divided into two categories according the learning approach used: those which group proteins *de novo* into biologically sound clusters, such as families or orthologous groups, and those which assign proteins to already known groups (classes) defined by experts. Both learning approaches, however, make use of training samples in the design of a classifier. Specifically, the former approach is equivalent to unsupervised learning or clustering because the class membership (labels) of the training samples is not known (or hidden) *a priori*. Moreover, there is usually no explicit teacher who guides the clustering procedure. In contrast, the latter approach corresponds to supervised learning or classification where the labels of the training samples are known during the process of training. Nevertheless, there is also an intermediate case between the two forms of learning, known as reinforcement learning. Herein, a sample is first assigned to a tentative cluster and then the assignment is judged only as correct or incorrect by a critic who provides merely a nonspecific (binary) feedback (Duda et al., 2000). From here on we use the term 'protein classification' to refer to all the approaches above.

The mathematical theory of graphs has provided a conceptual framework in which the structure, function and evolution of complex biological systems can be modeled and better understood (Barabási and Oltvai, 2004). Although the use of (phylogenetic) trees has dominated in biological classification over the past decades, there has been recently a substantial shift in this paradigm towards using also networks for classifying proteins of fully sequenced genomes according to functional or phylogenetic criteria (Kuzniar et al., 2008; Sharan et al., 2007). In such graphs the relationships (edges) between proteins (nodes) might have, however, different meanings such as

functional equivalence (e.g., identical substrate specificity or reaction mechanism) or common evolutionary descent (e.g., homology, orthology or paralogy). Figure 1.1 illustrates the use of trees and networks in protein classification.

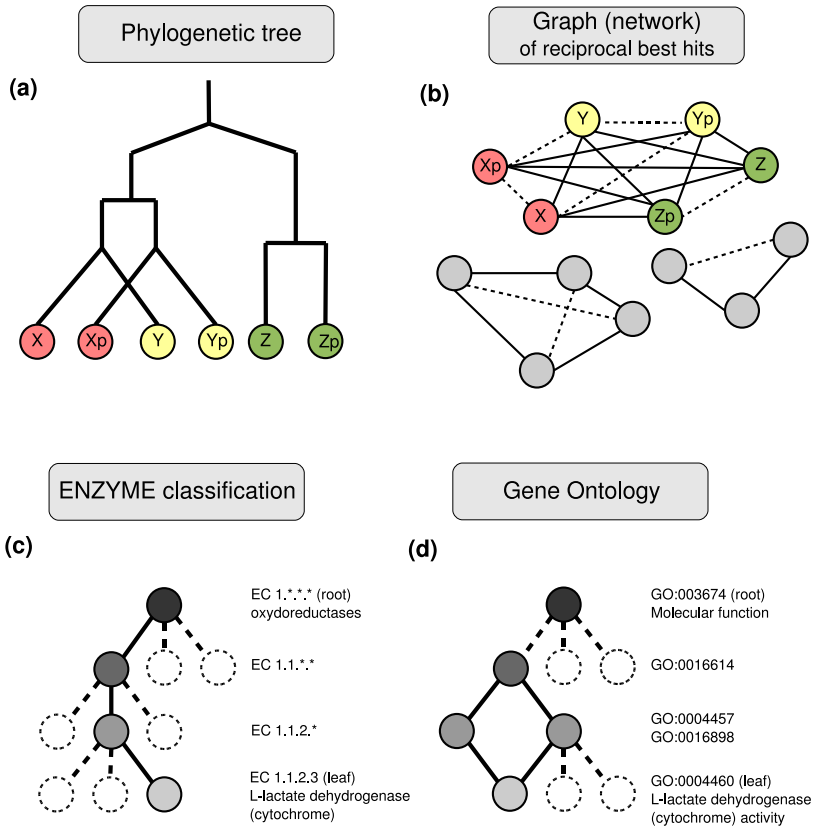


Figure 1.1. The use of trees and networks in protein classification. **(a)** A phylogenetic tree of six homologous genes (labeled x, x_p, y, y_p, z and z_p) from three genomes X, Y and Z is shown; speciation events are depicted as an inverted 'Y', otherwise branchings are gene duplications. **(b)** A protein similarity network is constructed of reciprocal best hits (e.g., defined on BLAST scores) for the genomes. Only tree disjoint sub-graphs are shown of which one corresponds to the phylogenetic tree. In the sub-graphs orthologous and paralogous relationships are depicted as solid and dashed lines (edges), respectively. **(c)** The ENZYME database (Bairoch, 2000) classifies enzymes according to the reaction they catalyze using the hierarchical Enzyme Commission (EC) numbers; an example classification of oxydoreductases is depicted herein. **(d)** The Gene Ontology (GO) database (Ashburner et al., 2000) describes biological processes, molecular functions and cellular components of gene products from all organisms using a consistent language that is both human- and computer-readable. Unlike the ENZYME's three-like structure, the GO structure is based on an directed acyclic graph (DAG) wherein a biological term such as 'L-lactate dehydrogenase (cytochrome) activity' can have multiple parents.

Protein family and cluster databases

Over the past decades, many protein family databases have been developed to improve the predictions of protein function, orthologs and distant family members over simple pairwise similarity search methods (Lee et al., 2007). For example, a database search using the best BLAST hit might yield incomplete or even incorrect function predictions when multi-domain or chimeric (hybrid) proteins are involved. Incomplete functional annotation occurs when single rather than multiple functions are inferred for a hybrid protein. Moreover, the function of a query protein is inferred incorrectly when the query returns a multi-domain protein as the most significant match because of a common, promiscuous domain, which occurs in many functional and domain contexts (Bork et al., 1998). It has been estimated that 5–40% of proteins of fully sequenced genomes are annotated erroneously (Brenner, 1999; Devos and Valencia, 2001). Therefore the function of an unknown protein can be predicted more reliably when compared against collections of domains and motifs with known functions.

The first collection of structural and functional motifs has been compiled manually more than 25 years ago (Bairoch, 1992). With the increasing amount of protein sequences, however, fully- and semi-automated methods have gained an increasing importance to construct protein and domain family databases. Historically, SBASE (Simon et al., 1992) and ProDom (Sonnhammer and Kahn, 1994) were the first available protein domain sequence databases, and the Clusters of Orthologous Groups of proteins (COG/KOG) database (Tatusov et al., 1997) was among the first to provide both examples and an algorithmic approach to infer orthologous proteins from fully sequenced genomes. Several large protein cluster databases such as PIRSF (Wu et al., 2004), SYSTERS (Meinel et al., 2005), CluSTr (Kriventseva et al., 2001a) and ProtoNet (Kaplan et al., 2005) have emerged to classify protein sequences into 'hierarchical' rather than 'flat' protein clusters. Such hierarchical classifications provide means, for example, to identify functional residues common to families (e.g., G-protein-coupled receptor family) but distinct between subfamilies (e.g., dopamine and histamine receptor subfamilies). Furthermore, a considerable effort has been invested to develop integrated databases such as InterPro (Hunter et al., 2009) and CDD (Marchler-Bauer et al., 2009) that link many diverse resources to increase the reliability of predictions by combined assignments.

Sequence-based protein clustering

An excellent general review of clustering techniques was given by Jain et al. (1999). Clustering protein or domain sequences into protein families or orthologous groups provides efficient means to study the structure, function and evolution of proteins for many genomes available (Kriventseva et al., 2001b). In particular, grouping proteins based on orthologous rather than homologous relationships (the latter might involve both orthologs and paralogs) provides more reliable means to infer the function of an unknown protein using the knowledge of functionally known protein ortholog(s) in other species (Koonin, 2005). For example, if an orthologous group consists of at least

one known protein then the function of other proteins of this group can be inferred reliably given that protein. The computational tools used for delineating orthologous groups have been reviewed elsewhere (Kuzniar et al., 2008).

Protein clustering is, however, a non-trivial task which requires several decisions to be made about a protein similarity (distance) measure, clustering algorithm, validation criterion and dataset used. The main caveats of large-scale protein clustering include multi-domain (chimeric) proteins, distant homology relationships, as well as the computational complexity (scalability) of algorithms used (Rahman et al., 2008). As there is no formal definition of a protein cluster, an approach to this problem is to use an internal criterion based on the notion that similar proteins should be within the same cluster whereas dissimilar ones should be in distinct clusters. Therefore a clustering algorithm can partition the input data into disjoint (non-overlapping) groups in such a way that the within-group similarities are maximized while the between-group similarities are minimized. One might also use an external criterion function that measures the similarity between the predicted and reference partitions; the optimal partition of all computed ones is the one that maximizes the value of such a criterion function (Duda et al., 2000).

In any clustering the choice of a similarity measure between (clusters of) objects is critical for the sensitivity and specificity of clustering. A common practice is to use a sequence similarity search algorithm, such as BLAST, FASTA or SSEARCH, to calculate all-*versus*-all similarities. Notably, several alignment-free sequence comparison methods have also been proposed to deal with 'hard-to-align' sequences, to substitute computationally intensive alignment algorithms, as well as to improve the quality of protein classification (Kelil et al., 2007; Kocsor et al., 2006; Vinga and Almeida, 2003). Furthermore, pre-processing steps such as score normalization and filtering are frequently included prior to clustering. For example, spurious sequence similarities are filtered out by choosing an appropriate similarity threshold T. However, such a filtering might change the clustering results significantly: if T is low then most proteins will be assigned to one large cluster, whereas if T is high then most proteins will remain on their own as singletons. There have been many studies aimed at establishing reliable similarity measures between proteins for making inferences of homology, 3D structure and function; however, these have been rarely evaluated in relation to protein clustering algorithms (chapter 5 of this thesis).

A similarity-based clustering involves two important processes namely the calculations of similarities between individual data points as well as between entire clusters. According to the latter the clustering algorithms are classified into three linkage categories namely single-, average- and complete-linkage (Defays, 1977; Sibson, 1973; Voorhees, 1986). Specifically, these linkage criteria are implemented into computations as mathematical functions that return the minimum (nearest neighbor or single link), average (average link) or maximum (complete link) value of all distances between individual data points. In graph-theoretical terms, the type of linkage relates to the concepts of closeness and connectivity between nodes in a graph (Murtagh, 1985). Specifically, methods based on single-linkage and complete-linkage correspond to finding maximal connected subgraphs (connected components) and maximal completely-connected subgraphs (cliques), respectively. Whilst the former

yields somewhat 'loose' clusters, the latter results in 'tight' clusters. The average-linkage is in fact an intermediate between the two methods.

The major computational bottleneck for similarity-based clustering is the aforementioned calculation of all-*versus*-all sequence similarities because this computation does not scale linearly but quadratically with respect to the number of sequences used. Hence, comparing twice as many proteins will take four times as long to run. Several databases of precomputed all-*versus*-all protein similarities such as CluSTr (Petryszak et al., 2005), SIMAP (Rattei et al., 2006) and PairsDB (Heger et al., 2008) have been made publicly available to facilitate efficient downstream analyses including sequence clustering. However, the quadratic time complexity of all-*versus*-all sequence comparisons cannot be solved by increasing the speed of database search algorithms. Recently, a similarity heuristic approach has been proposed to reduce the amount of sequence comparisons to a smaller subset, yet without a significant decrease in the quality of clusters (Kull and Vilo, 2008). A different approach relies on an alignment-free method that speeds-up the calculation of sequence identities significantly (Li and Godzik, 2006). This method is used for clustering millions of protein sequences into series of non-redundant protein identity groups, such as those provided by the UniProt Reference Clusters databases (UniRef50/90/100) (Suzek et al., 2007).

Protein clustering algorithms

Over the past decades hierarchical clustering techniques have been used extensively in biological sequence analysis, in particular for phylogenetic clustering of amino acid or nucleotide sequences (Felsenstein, 1989; Sankoff, 1975). Hierarchical methods such as the Unweighted Pair Group Method with Arithmetic mean (UPGMA or average-linkage) (Michener and Sokal, 1957) or Neighbor-Joining algorithm (NJ) (Saitou and Nei, 1987) construct nested series of partitions that can be viewed as a tree or dendogram. Such dendograms can be constructed in two different ways either using an agglomerative (bottom-up) approach such as UPGMA, or using a divisive (top-down) approach such as NJ. Whilst the UPGMA algorithm begins with singleton clusters at the bottom (leaves) and iteratively joins the nearest clusters until a single all-inclusive cluster is formed at the top (root) of the tree, the NJ algorithm proceeds in reverse order. The most important difference between the two algorithms is that UPGMA implicitly assumes the existence of a clock-like, or ultra-metric tree, in which the total branch length from the root to any leaf is equal, while NJ does not make such an assumption. Therefore UPGMA might yield incorrect results for data where the 'molecular clock' hypothesis does not hold.

Many agglomerative (hierarchical) clustering methods using single-linkage (Enright et al., 2002; Enright and Ouzounis, 2000; Krause and Vingron, 1998; Petryszak et al., 2005), average-linkage (Uchiyama, 2006; Yona et al., 1999) or complete-linkage (Roth et al., 2008) have been used successfully for predicting protein function, 3D structure, remote homology (Bolten et al., 2001) and orthologous groups for large sets of proteins (proteomes) (Kuzniar et al., 2008). Recently these methods have been further improved in speed and memory usage using efficient heuristics that can handle even large protein collections (Kull and Vilo, 2008; Loewenstein et al., 2008). In some

scenarios the amount of data to be clustered, however, is so large that it cannot fit into the computer's memory. In chapter 4 we propose a simple graph streaming algorithm that is suitable for clustering proteins in (nearly) linear time and space without storing the entire similarity matrix in memory. The graph-theoretic interpretation of clustering have motivated the development of clustering algorithms that can be used in hierarchical or 'flat' (non-hierarchical) clustering schemes. The examples include the minimum cut (Hartuv et al., 2000), normalized cut (Abascal and Valencia, 2002; Shi and Malik, 2000), locally minimal cut (Kawaji et al., 2004), SYSTERS (Krause et al., 2005), FORCE (Wittkop et al., 2007) and Markov Cluster algorithms (MCL) (van Dongen, 2000). In particular, the MCL algorithm has been incorporated in most protein clustering methods including TribeMCL (Enright et al., 2002), OrthoMCL (Li et al., 2003), hybrid (single-linkage combined with MCL) clustering (Harlow et al., 2004) and MACHOS (Wong and Ragan, 2008), which were designed for different, yet overlapping purposes such as the detection of protein (domain) families and/or orthologous groups from available genomes. An important feature of the MCL algorithm is that it can split large, 'loosely-connected' subgraphs (clusters) into smaller, 'tighter' clusters by increasing the value of its inflation parameter. Specifically, large assemblages of non-homologous proteins, which arose due to the presence of chimeric proteins, can be split into smaller groups of protein with similar domain architecture. In contrast, a single-linkage clustering does not address this problem explicitly; however, the use of adjusted scoring schemes or post-processing procedures can prevent illegitimate use of transitive homology effectively (Bolten et al., 2001; Park and Teichmann, 1998; Pipenbacher et al., 2002). The concept of transitive (indirect) homology implies that two proteins are homologous to each other if they have direct homologous relationships to a third non-chimeric protein. Somewhat exotic clustering algorithms such as the Super Paramagnetic Clustering (SPC) (Tetko et al., 2005) and Spectral clustering (Paccanaro et al., 2006) have shown higher sensitivity and specificity than the popular TribeMCL method. Interestingly, two independent studies showed that a simple single-linkage method might yield protein clusters of comparable or, sometimes, even higher quality than those produced by sophisticated methods such as TribeMCL and SPC (Kelil et al., 2007; Krause et al., 2005). These conflicting results indicate that there is a need for reliable benchmark protocols and datasets used to evaluate protein clustering methods with respect to quality *versus* scalability trade-off.

Evaluating protein clustering results

Any unsupervised method such as clustering requires results to be evaluated using some kind of validation method. The aim of cluster validation is to show the strengths and weaknesses of a clustering method on different benchmark datasets. For this, many validation techniques have been developed in the domain of data-mining in general (Halkidi et al., 2001); however, only a few of these have been applied to protein clustering (Handl et al., 2005; Miller et al., 2008). Moreover, there is an increasing demand for standardized benchmark datasets and reliable validation protocols with which the performance of protein classification methods can be compared

systematically (Sonego et al., 2007).

In practice, protein clusters are assessed for biological soundness using visual inspection of the corresponding multiple-sequence alignments, phylogenetic trees, and protein domain or motif predictions. For example, a web-based visualization tool such as TreeDomViewer (Alako et al., 2006) provides a combined view of such inferences into a single plot where different family predictions methods can be examined. Alternatively, one can construct an ensemble clustering in which different, sometimes competing protein classifications can be compared visually in set- and graph-theoretic terms (Kuzniar et al., 2009), as well as combined into a single consensus clustering using an election algorithm (Nikolski and Sherman, 2007). A different approach to validate protein clusters involves statistical methods for set-enrichment analysis whereby the clusters are assessed using manually curated annotations such as Gene Ontology (GO) terms (Ashburner et al., 2000), ENZYME commission (EC) numbers (Bairoch, 2000), structural/functional domains and motifs [e.g., SCOP (Andreeva et al., 2008), Pfam (Finn et al., 2008) or PROSITE (Hulo et al., 2008), and UniProtKB/SwissProt descriptions]. These methods rely upon a statistical test such as hyper-geometric, binomial, chi-square or Fischer's exact test, and are available as web-based and stand-alone tools (Huang et al., 2009).

In principle, there are three types of criteria used in cluster validation: (i) external, (ii) internal and (iii) relative criteria (Halkidi et al., 2001). First, in external validation two partitions, namely a new clustering *versus* benchmark set with known classes, are compared to each other in order to quantify the amount of (dis)agreement between them using an appropriate measure (index). Such an index can be either a similarity or distance function. The clustering literature references many validation indices designed for different scenarios. For example, some indices are better suited for 'hard' rather than 'soft' clustering where an object is assigned to one (disjoint) cluster rather than multiple clusters, respectively. Second, when a 'gold' standard is not available, clustering can be evaluated only using features intrinsic to a clustering and the underlying dataset. These include compactness (such as intra- or inter-cluster variance) and spatial separation (see linkage criteria). In both internal and external scenarios, Monte Carlo methods are used to assess the significance of *de novo* clusters. Finally, one can compare different partitions to each other constructed by the same algorithm used with different parameter settings. In a strict sense, this is an optimization technique rather than a true validation because the aim is to obtain the "best" clustering that maximizes the criterion function (index). In summary, the biological soundness of protein clusters cannot be truly assessed without an objective external criterion that relies on prior biological knowledge.

All validation indices for comparing clustering can be derived from a contingency table whose elements correspond to intersections between clusters of the partitions compared. In protein clustering a TP instance is when two similar proteins are assigned to the same cluster, while a TN instance is when two dissimilar proteins are assigned to different clusters. However, two types of errors might occur during such assignments, in particular when two dissimilar proteins belong to the same cluster (FP), or when two similar protein belong to different clusters (FN). According to Meilă the cluster validation indices are classified into three categories: i) indices

based on counting points on which two partitions agree and/or disagree [e.g., Jaccard (Jaccard, 1901; Rand, 1971; van Rijsbergen, 1979)], ii) indices based on set-matching between two partitions [e.g., van Dongen metric (van Dongen, 2000)], and entropy-based indices such as the Variation of Information (VI) metric (Meilă, 2007).

Cluster purity, Rand and Jaccard indices, and F-measure are amongst the most popular indices used for validating protein clustering algorithms (Enright and Ouzounis, 2000; Kaplan et al., 2005; Krause et al., 2005; Wittkop et al., 2007; Yang and Zhang, 2008). These indices are bounded between 0 and 1 and can therefore be interpreted as probabilities. To calculate cluster purity, each cluster is labeled with the most frequent class label in that cluster, and then the total count of correctly assigned proteins is divided by N. This measure is, however, overly optimistic particularly for clustering with large numbers of small (or singleton) clusters. In contrast, the Jaccard similarity coefficient (index) is a strict statistic measure used for comparing the similarity and diversity of sample sets because it rewards only TP instances. A somewhat more liberal, Rand index rewards both TP and FP decisions.

1.6 Thesis outline

Computational methods used in phylogenetics as well as in machine learning are important for many areas of genomic research such as comparative genomics, high-throughput genome annotation, genome (or proteome) evolution and reconstruction of cellular networks. The main objective of this thesis was to develop a reliable and scalable method suitable for predicting protein families and orthologous relationships from hundreds of fully sequenced genomes available in public databases.

In chapter 2 we review computational tools, such as algorithms and databases, used for inferring orthologous relationships between genes from fully sequenced genomes. We discuss the main caveats of large-scale orthology detection in general, as well as focus on the merits and pitfalls of each method in particular. This review aims at providing a set of guidelines to aid researchers in selecting the correct tool, as well as motivates further research carried out in this thesis.

Chapter 3 proposes a framework in which various protein knowledge-bases are combined into unique network of mappings (links), and hence allows comparisons to be made between expert curated and fully-automated protein classifications from a single entry point. These mappings are stored in a non-redundant protein database called *ProGMap* (Protein Group Mappings), which is meant to help not only users of high-throughput techniques (e.g., microarrays or proteomics) who deal with partially annotated genomic data, but also database curators who often need to test the coherence of proposed annotations and/or assignments.

Chapter 4 is divided into two sub-chapters namely 4.1 and 4.2. Chapter 4.1 presents a benchmark study of graph-based software used commonly for detecting similarity groups, such as protein families or orthologous groups, in protein similarity networks. In this study the computational complexity of the programs is tested using both simulated and biological datasets. We introduce a fast and memory-efficient implementation called *netclust* suitable for finding similarity groups in large protein

similarity networks, such as those of millions of proteins and pairwise sequence similarities, using a standard desktop computer. In chapter 4.2 we describe a network-based tool, *Multi-netclust*, that can find connected clusters in multi-parametric networks by combining different datasets using a simple kernel-fusion method.

Sequence-based clustering using the nearest neighbor approach is known to yield biologically meaningful groups such as protein families. Nevertheless, the quality of the resulting groups is generally believed to be inferior to other, more sophisticated, yet computationally demanding methods. In chapter 5 the best known protein clustering methods are evaluated systematically using distinct protein similarity networks and 'gold' standard or validation datasets. Here, we aim at improving our scalable method using an optimized scoring (such as protein similarity measure and threshold) and hierarchical classification scheme.

In chapter 6 we summarize the major contributions of this thesis, discuss the results in the view of current findings in the literature, and propose directions for future research.

CHAPTER 2

THE QUEST FOR ORTHOLOGS: FINDING THE CORRESPONDING GENE ACROSS GENOMES

Abstract

Orthology is a key evolutionary concept in many areas of genomic research. It provides a framework for subjects as diverse as the evolution of genomes, gene functions, cellular networks and functional genome annotation. Although orthologous proteins usually perform equivalent functions in different species, establishing true orthologous relationships requires a phylogenetic approach, which combines both trees and graphs (networks) using reliable species phylogeny and available genomic data from more than two species, and an insight into the processes of molecular evolution. Here, we evaluate the available bioinformatics tools and provide a set of guidelines to aid researchers in choosing the most appropriate tool for any situation.

2.1 The concept of orthology

In the early days of comparative biology, relationships between different species were studied using morphological characters. With the emergence of sequencing techniques and, in particular, the high-throughput techniques of the past decade, the amount of molecular characters in the form of fully sequenced genomes from a diverse range of organisms has increased enormously. A wide array of bioinformatics tools has been developed to interpret the sequence data from evolutionary and functional perspectives (Ouzounis et al., 2003). The knowledge of molecular phylogenies in general and orthology in particular has become an integral component of many genome-scale studies of gene content, conserved gene order and gene expression, regulatory networks, metabolic pathways and in functional genome annotation (Bandyopadhyay et al., 2006; Delsuc et al., 2005; Eisen, 1998a; Goodstadt and Ponting, 2006; Grigoryev et al., 2004; Hulsen et al., 2006a; Jeffroy et al., 2006; Mao et al., 2006; Mazurie et al., 2005; Tatusov et al., 1997).

The concept of homology (see Glossary) is fundamental to make inferences about evolutionary processes such as speciation, gene duplication or horizontal gene transfer (HGT). At the beginning of the 1970s, Walter Fitch divided homology into orthology and paralogy according to the distinct evolutionary processes, namely speciation and gene duplication, respectively (Fitch, 1970; 2000). Thus, orthologs are homologous genes that relate through speciation from a single ancestral gene present in their latest common ancestor, whereas paralogs are homologs that arose through gene duplication. Nonetheless, an understanding of homology, orthology and paralogy has been challenged by other important evolutionary processes such as HGT and gene fusion or fission events, which are thought to have enabled the formation of complex

phylogenetic networks (Doolittle and Bapteste, 2007; Rivera and Lake, 2004). Several terms (e.g., in-paralogs, out-paralogs, super-orthologs or ultra-paralogs) have been coined to further refine the various evolutionary origins of sequence similarities. The term 'orthology' is often misunderstood to refer to functionally equivalent genes in different species; but, it is strictly an evolutionary concept, rather than a functional one (Fitch, 2000). Orthologs have primarily been used as evolutionary markers for inferring species phylogenies because they follow species divergence (Blair and Hedges, 2005; Ciccarelli et al., 2006), but they can be used to link functionally equivalent genes across genomes and, as such, enable the function of an unknown protein to be inferred using known (i.e., functionally characterized) orthologs in other species (Koonin, 2005; Tatusov et al., 1997). However, the main caveats of using orthologs in function annotation are domain shuffling, presence or absence of a domain, lineage-specific gene duplication and gene loss (Sjölander, 2004). Controlled vocabularies (ontologies) have emerged to describe biological functions (e.g., gene functions, mode and site of action within a cell) in a standardized form and have intensively been used to link heterogeneous datasets of various molecular databases (Ashburner et al., 2000; IUBMB, 1992; Ruepp et al., 2004). For example, databases such as Kyoto Encyclopedia of Genes and Genomes (KEGG; http://www.genome.jp/kegg/), BioCyc (http://biocyc.org/) or IMG (http://img.jgi.doe.gov/) integrate molecular data on pathways, enzymes and substrates associated with orthologous genes (proteins) from diverse genomes (Kanehisa et al., 2006; Krummenacker et al., 2005; Markowitz et al., 2008).

Here, we review the computational tools (i.e., programs and databases) commonly used to infer orthologous relationships between genes and proteins (Boxes 1–3). We compare the orthology detection tools and demonstrate the advantages and/or limitations of these methods using real examples of gene families and evolutionary scenarios. Also proposed is a set of guidelines to aid researchers in selecting the correct tool in a given situation.

2.2 Orthology detection methodologies

For the purpose of this review, a classification scheme that recognizes both conceptual and practical differences among orthology detection tools available to date has been introduced (Figure 2.1). The different tools are grouped along methodological lines: those based on trees (tree-based methods), graphs (network or graph-based methods) or both (hybrid methods). From a practical point of view, this classification distinguishes between *ab initio* and post-processing tools. The former example infers orthologs in entire sets of genes (proteins) of two or more species and the latter two use precomputed homologs to infer orthologs and paralogs. Furthermore, a distinction is made between the methods that use exclusively primary sequence data and those that also use auxiliary information, such as conserved gene neighborhood (CGN). Although CGN might assist in finding additional orthologs when inference of homology is hampered by low sequence similarity (Simillion et al., 2004), or in distinguishing true orthologs from single-copy paralogs (out-paralogs) in the presence of reciprocal

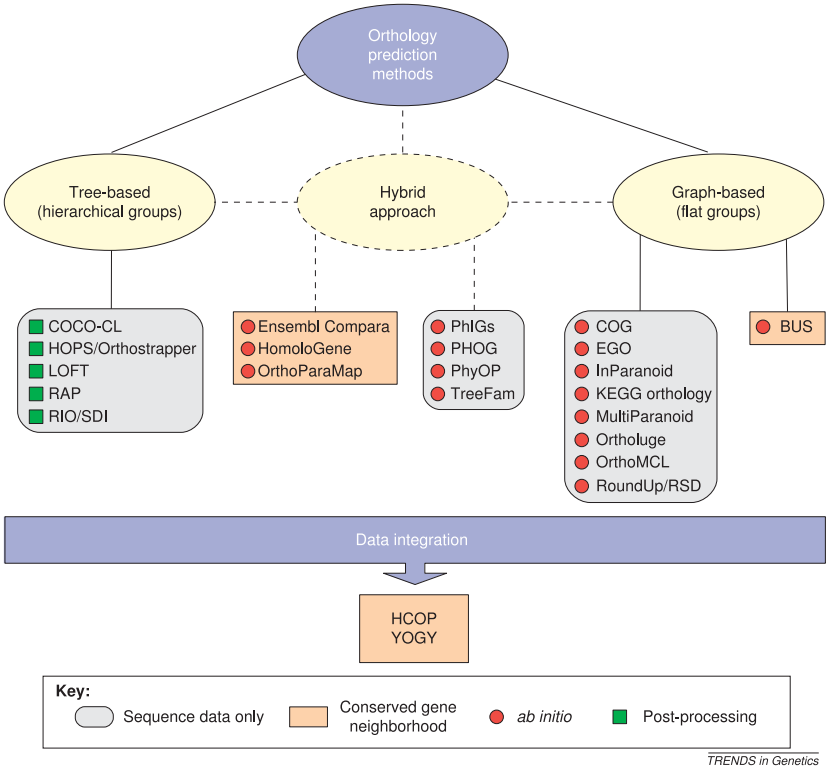


Figure 2.1. Classification of orthology detection methods. Three main categories are recognized according to the data representations they operate on, including tree-based, graph-based and hybrid methods (see main text for a full description). Further distinctions are based on conserved gene order (CGN) and *ab initio* or post-processing approaches. Data integration does not offer a new algorithmic approach *per se*, but is used to merge multiple datasets, which include both experimentally verified and automatically predicted orthologs, into a unified, consolidated collection. The examples of integrated databases include HUGO gene nomenclature committee (HGNC) Comparison of Orthology Predictions (HCOP; http://www.genenames.org/) and Eukaryotic Orthology (YOGY; http://www.bahlerlab.info/YOGY/) (Eyre et al., 2007; Penkett et al., 2006). A comparison of tree-based, graph-based and hybrid methods is given in Boxes 1–3, respectively.

gene losses (Scannell et al., 2006; 2007), it is applicable only to closely related species (Huynen and Bork, 1998). The merits and pitfalls of various orthology detection tools are summarized in Boxes 1–3 (Alexeyenko et al., 2006; Bandyopadhyay et al., 2006; Cannon and Young, 2003; Chen et al., 2006; Dehal and Boore, 2006; Deluca et al., 2006; Dufayard et al., 2005; Duret et al., 1994; Fulton et al., 2006; Goodstadt and Ponting, 2006; Hirsh and Fraser, 2001; Hubbard et al., 2009; Jothi et al., 2006; Koski and Golding, 2001; Lee et al., 2002; Li et al., 2006; 2003; Merkeev et al., 2006; O'Brien et al., 2005; 2004; Overbeek et al., 1999; Perrière et al., 2000; Remm et al., 2001; Storm and Sonnhammer, 2002; Tatusov et al., 1997; van der Heijden et al., 2007; Wall et al., 2003; Wheeler et al., 2008; Zmasek and Eddy, 2002).

Tree-based methods

Tree-based methods infer orthologous and paralogous relationships from phylogenetic trees. First, one must collect homologous sequences, construct a multiple-sequence alignment and phylogenetic tree(s) and then, the relationships can be analyzed either in the presence or absence of 'known' phylogenetic relations between species (e.g., mouse, rat and human). Because a gene tree does not necessarily have the same topology as the species tree, owing to evolutionary processes such as gene loss and HGT, tree-reconciliation techniques, which infer speciation (orthologs) and duplication (paralogs) events from reconciled trees, have been commonly used to account for these differences (Dufayard et al., 2005; Goodman et al., 1979; Page and Charleston, 1997; Zmasek and Eddy, 2001). However, this approach can only be used when the species tree is reliable. This poses the question of how one deals with those cases in which the phylogenetic relationships between species are not known. Recently, two methods, namely the Correlation Coefficient-based Clustering (COCO-CL) and the Levels of Orthology From Trees (LOFT), have been proposed to distinguish between orthologs and paralogs in a gene tree without using a corresponding species tree (Jothi et al., 2006; van der Heijden et al., 2007).

The current tree-based methods have several shortcomings. First, phylogenetic-tree reconstruction algorithms rarely produce completely reliable trees. Ambiguities in either a gene tree or a species tree result in a spurious inference of duplication and speciation events. However, one can use sampling methods, such as bootstrap (Felsenstein, 1988) or Markov Chain Monte Carlo (MCMC) (Larget and Simon, 1999) methods to assess the reliability of the tree. Second, the tree-based algorithms require properly rooted trees, which are commonly rooted by the midpoint in the tree or by the careful manual selection of an out-group species. Midpoint rooting approaches are often problematic for protein families in which members evolve at different rates, whereas the manual selection of out-groups might be impractical and difficult to automate, especially for large-scale genome analyses (Zmasek and Eddy, 2001). Alternatively, the trees can be rooted by an approach that minimizes dissimilarity between the gene and species trees (Page and Charleston, 1997). Third, a plausible phylogenetic gene tree depends on a biologically correct multiple-sequence alignment. Therefore, incorrect alignments draw false conclusions about evolution. Finally, algorithms for phylogenetic-tree construction and multiple-sequence alignment scale poorly with the increasing amount of sequence data available and are not suitable for complete genomes. Although the computational cost can be reduced with heuristic algorithms, or deploying parallel algorithms on distributed systems, it is challenging to construct reliable sequence alignments and trees for large gene families that have complex histories.

In summary, tree-based methods provide phylogenetic resolution at multiple levels of a gene tree and are suitable to infer orthologs and paralogs from any protein (domain) family database available. However, these approaches are computationally intensive for large datasets, not easily automated owing to the need to choose appropriate outgroup species and depend on the pre-defined protein families.

Box 1. Tree-based methods

Correlation Coefficient-based Clustering (COCO-CL)
The COCO-CL program takes the non-transitivity of phylogenetic relations within a set of homologous proteins into account using a hierarchical numbering scheme (Jothi et al., 2006). It uses a heuristics based upon Pearson's correlation matrix of sequence distances to decides upon speciation and duplication events without a species tree. Sets containing out-paralogs are recursively split into two smaller sub-sets until no additional out-paralogs are found, thus forming a hierarchy of sets. Each split is flagged as either speciation or duplication according to its reliability (bootstrap) score. **Pros:** COCO-CL infers orthologs and paralogs from precomputed homologs in a hierarchical framework without a species tree. The COCO-CL program and refined COG dataset are freely available (http://www.ncbi.nlm.nih.gov/CBBresearch/Przytycka/COCOCL/). **Cons:** COCO-CL does not implement a tree-reconciliation algorithm.

Orthostrapper and Hierarchical grouping of Orthologous and Paralogous Sequences (HOPS)
The Orthostrapper program uses a heuristic sequence similarity search to infer orthologs with confidence values from a set of bootstrapped gene trees (Storm and Sonnhammer, 2002). Orthostrapper does not use a species tree in a strict sense. Instead, sequences are assigned to a taxonomic group. The HOPS database provides orthology assignments for eukaryotic Pfam domains (Storm and Sonnhammer, 2002). **Pros:** HOPS provides domain-based orthologs. The Orthostrapper program is freely available (http://sonnhammer.sbc.su.se/download/software/orthostrapper/). **Cons:** HOPS dataset is not available for download and the web server does not work.

Levels of Orthology From Trees (LOFT)
The LOFT program addresses the non-transitivity of phylogenetic relations within phylogenetic trees (van der Heijden et al., 2007). It implements two algorithms to infer speciation or duplication events in a given gene tree. Besides the SDI tree-reconciliation algorithm, LOFT offers an alternative approach, the so-called 'species-overlap' rule, especially when the species tree is not known. This simple heuristics implies that a speciation event is only assigned to an internal node if its branches contain mutually exclusive sets of species. LOFT makes a use of a hierarchical numbering scheme for orthologous groups (similar to that found in COCO-CL). **Pros:** LOFT infers orthologs and paralogs from precomputed homologs in a hierarchical framework without a species tree. The LOFT program comes with a GUI. Both the program and the refined COG dataset are freely available (http://www.cmbi.ru.nl/LOFT/). **Cons:** LOFT cannot be executed without the GUI as a command line tool. The 'species-overlap' is not adjustable.

Réconciliateur d'Arbres Phylogénétiques (RAP)
Originally, the RAP tree-reconciliation program (Dufayard et al., 2005) was used to infer orthologs in HOVERGEN and HOBACGEN (Duret et al., 1994; Perrière et al., 2000) databases. **Pros:** The algorithm can handle unresolved trees and take both bootstrap values and branch lengths into account for the reliability of trees. The RAP program is freely available (http://pbil.univ-lyon1.fr/software/RAP/). **Cons:** RAP cannot be used as a command line tool.

Speciation Duplication Inference (SDI) and Resample Inference of Orthologs (RIO)
The SDI tree-reconciliation algorithm requires properly rooted and completely binary input trees to infer speciation and duplication events reliably. The orthology assignments in the RIO database (Zmasek and Eddy, 2002) were made by using the Pfam protein domains and SDI algorithm on bootstrap re-sampled gene trees (Zmasek and Eddy, 2002). A confidence (orthology bootstrap) score is given for each database hit. High scores indicate 'true' orthology, whereas low values indicate absence of orthologs. Three novel homology concepts were introduced to enhance function prediction of genes (see Glossary; super-orthologs, ultra-paralogs and subtree-neighbors). **Pros:** RIO provides phylogenetic resolution for domain-based orthologs with confidence scores. The SDI algorithm is freely available. **Cons:** RIO data are not available and the web server is not operational (http://rio.janelia.org). SDI cannot root the input trees and requires fully resolved trees.

Graph-based methods

Graph-based methods are suitable for orthology inferences from two or more complete genomes (proteomes). Unlike tree-based methods, they do not construct multiple-sequence alignments and phylogenetic trees, but rely on pairwise sequence similarities calculated between all sequences involved and an operational definition of orthology, for example, reciprocal best hits (RBHs) (Box 2). The choice of a sequence-similarity search algorithm [e.g., basic local alignment search tool (BLAST) or SmithWaterman] and a scoring scheme for pairwise alignments has a bearing on the sensitivity and specificity of orthology predictions (Hulsen et al., 2006b). Some graph-based methods use clustering techniques [e.g., single-linkage, complete-linkage or Markov Cluster algorithm (Enright et al., 2002)] to extend nearest neighbors to more than two species and construct multi-species orthologous groups (OGs) of particular granularity (Chen et al., 2007). These approaches use the definition of orthology liberally because orthologs and paralogs are often grouped together in an OG, in which all members arecollapsed down to the last common ancestor of all species in that OG. However, this is not a concern for graph-based methods that analyze two species at once (either in the presence or absence of an out-group) (Fulton et al., 2006; Remm et al., 2001; Wall et al., 2003).

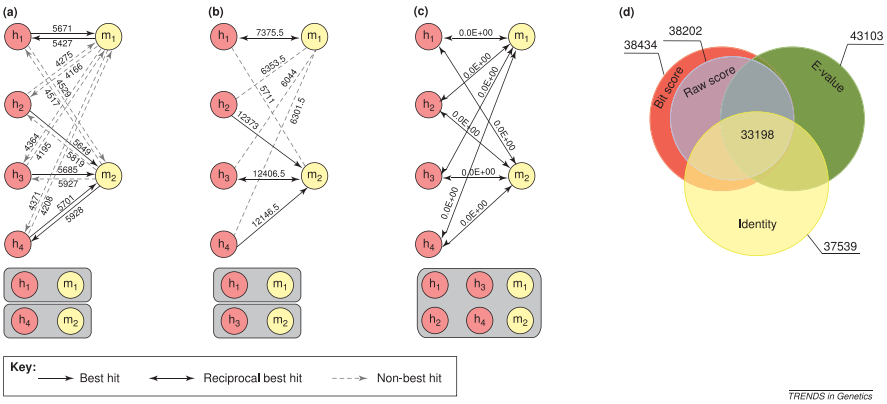


Figure 2.2. Different sets of putative orthologs defined as reciprocal best hits. Three graphs of human (*h1–4*) and mouse (*m1–2*) mucin-5 proteins are constructed using three different protein similarity measures: **(a)** asymmetric BLAST raw score; **(b)** symmetric Smith-Waterman score; and **(c)** symmetric BLAST E-value. The corresponding set of predicted orthologs is shown below each graph. Clearly, the reciprocal best hit approach using different similarity measures might result in different but largely overlapping sets of orthologs. **(d)** Venn diagram of four different sets of orthologs, using BLAST identity, E-value, raw and bit score, are inferred from complete human and mouse proteomes (Refseq version 29).The total number of orthologs is indicated for the sets and four-way intersection. Graph nodes correspond to RefSeq protein accessions: *h1*, XP_001717932; *h2*, NP_059981 (*Muc5ac*); *h3*, NP_002449 (*Muc5b*); *h4*, XP_001719401; *m1*, NP_034974 (*Muc5ac*); *m2*, NP_083077 (*Muc5b*).

Box 2. Graph-based method

Nearest neighbor
We use the term 'nearest neighbor' to collectively designate all approaches that apply an operational definition of orthology. Even though the approaches do not necessarily imply phylogenetic proximity (Koski and Golding, 2001), they are commonly used as first-pass approximations to find putative orthologs using some 'flavor' of the 'best' genome-wide matches between two species. These methods include best hit (BeT), reciprocal best hit (RBH), bi-directional best hit (BBH), symmetrical best hit (SymBeT) and reciprocal smallest distance (RSD) (Hirsh and Fraser, 2001; Lee et al., 2002; Overbeek et al., 1999; Remm et al., 2001; Tatusov et al., 2003; Wall et al., 2003). The nearest-neighbor methods might also address one-to-many and many-to-many orthologous relations depending on which definition is used and how it is implemented in the computation. The key concepts are best understood using graph theory (Figure 2.2). Clearly, the RBH approach using different similarity measures might result in distinct, but largely overlapping, sets of orthologs.

Clusters of Orthologous Groups (COGs) of proteins
The COG approach extends best BLAST hits (BeTs) to multiple proteomes by using congruent 'triangles' of BeTs from at least three different species (Tatusov et al., 2003). These minimal COGs are then merged by a single linkage into larger groups (protein families). The database consists of two sections for unicellular (mainly prokaryotes) and eukaryotic proteomes (euKaryotic Orthologous Groups or KOGs) from 66 fully sequenced genomes. **Pros:** The COG database is a widely used resource for functional annotation of genomes, mainly owing to availability and manual curation (http://www.ncbi.nlm.nih.gov/COG/). COGs are functionally annotated. The COG database stores orthologous groups from prokaryotic and eukaryotic genomes. **Cons:** The 'triangles' of the COG are disadvantageous in the presence of gene losses. The COG approach does not differentiate between in- and out-paralogs automatically; therefore, one needs to investigate the precomputed phylogenetic trees for duplication and speciation events. The automatic clustering procedure creates exclusive clusters, thus, multi-domain proteins must be handled manually. The database has not been updated since 2003.

Eukaryotic Gene Orthologs (EGO)
The EGO (previously known as TIGR Orthologous Gene Alignments or TOGA) database is constructed by an orthology detection procedure similar to that of the COG system (Lee et al., 2002), but instead of proteins, it uses virtual assemblies of transcripts, which provide evidence of a gene at the transcription level. **Pros:** The EGO database is freely available (http://compbio.dfci.harvard.edu/tgi/ego/) and contains more genomes (89) than COG. **Cons:** It has similar disadvantages as the COG approach and it does not have functional annotations.

InParanoid
The InParanoid program distinguishes between in-paralogs and out-paralogs for two proteomes without using phylogenetic trees (Remm et al., 2001). Instead, the method implements a set of heuristic rules to merge, delete and separate predicted orthologous groups. First, the main orthologs are identified as protein pairs having the highest symmetric BLAST score and are used as 'seeds' for finding all in-paralogs for each species. InParanoid and OrthoDisease (http://orthodisease.sbc.su.se) databases store orthology assignments mainly of eukaryotic species (35) (O'Brien et al., 2005; 2004). **Pros:** InParanoid addresses one-to-many and many-to-many orthologous relationships between two proteomes. It also enables an out-group species. Confidence values are assigned to individual in-paralogs and orthologous groups as a whole. The program and the database are freely available (http://inparanoid.sbc.su.se/). **Cons:** InParanoid is limited to pair-wise proteome comparisons and does not permit overlapping clusters in the presence of a hybrid protein.

MultiParanoid
The MultiParanoid program constructs multi-species orthologous groups of proteins from all possible pairwise species InParanoid comparisons. The clustering is less stringent (a single-linkage approach) than that of the approach of COG (Alexeyenko et al., 2006). **Pros:** MultiParanoid constructs

multi-species orthologous groups. The program and the dataset of four eukaryotic species is freely available (http://multiparanoid.sbc.su.se/index.html). **Cons:** MultiParanoid can be used for only a few species, which diverged at roughly the same time point from a common ancestor, otherwise the approach becomes inclusive for out-paralogs. It does not address the non-transitivity of phylogenetic relations. The web server is broken; a major update is planned (JL, personal communication).

Ortholuge
The Ortholuge program is designed to improve the specificity of RBH-based orthology predictions by handling gene-loss events for both bacterial and eukaryotic species (Fulton et al., 2006). The method is similar to InParanoid but it uses phylogenetic distance ratios instead of BLAST similarities. **Pros:** Ortholuge can use precomputed (tentative) orthologs or construct a dataset using an RBH-based BLAST approach. It is freely available (http://www.pathogenomics.ca/ortholuge). **Cons:** Ortholuge predictions of orthologs are incomplete in the presence of single gene loss. Ortholuge is limited to pair-wise proteome comparisons.

OrthoMCL and OrthoMCL–DB
The OrthoMCL pipeline integrates a Markov Cluster algorithm (MCL) (S. van Dongen, PhD thesis, University of Utrecht, 2000) for grouping proteins into multi-species orthologous groups (Li et al., 2003). First, 'seed' orthologs and in-paralogs are found using a similar approach to that of InParanoid and clustered using the MCL algorithm. Similarities between proteins are calculated as normalized BLAST P-values. The OrthoMCL-DB database stores orthologs of mainly eukaryotic genomes (87 species) (Chen et al., 2006). **Pros:** The OrthoMCL program constructs multi-species orthologous groups, which can be queried by phylogenetic patterns (presence and absence of species). The program and the database are freely available (http://orthomcl.cbil.upenn.edu/). **Cons:** OrthoMCL does address the non-transitivity of phylogenetic relations within orthologous groups. It might group out-paralogs and orthologs together in the presence of gene losses and does not handle hybrid proteins. The groups do not have function annotations.

Reciprocal Smallest Distance (RSD) and RoundUp
The RSD approach combines local and global sequence alignments and maximum likelihood estimation of evolutionary distances together to predict orthologous proteins (Overbeek et al., 1999). The RoundUp repository encompasses pairwise species orthologs from more than 250 genomes at various threshold levels of BLAST E-values and sequence divergence (Deluca et al., 2006). **Pros:** RSD uses explicit evolutionary model to calculate distances between proteins. The RoundUp database covers wide range of species (http://roundup.hms.harvard.edu/). **Cons:** RSD cannot compare more than two genomes simultaneously and does not permit the use of an out-group species.

Best Unambiguous Subset (BUS)
The BUS algorithm detects groups of orthologs between two genomes using a single linkage graph clustering (M. Kellis, PhD thesis, Massachusetts Institute of Technology, 2003). Graph edges are weighted by the amino acid sequence identity and the overall length of BLAST matches. An orthologous group consists only of genes that have 'best' matches within the group and no 'best' matches of any gene are outside that group. **Pros:** BUS makes a use of CGN to find additional putative orthologs, and can handle incomplete (draft) genomes. **Cons:** BUS is limited to pair-wise genome comparisons and is not available online.

Hybrid methods

Hybrid methods make use of both tree and graph representations at various stages of processing; for example, to refine OGs within a hierarchical framework of phylogenetic trees or to guide the clustering procedure using a species tree (Cannon and Young, 2003; Dehal and Boore, 2006; Goodstadt and Ponting, 2006; Hubbard et al., 2009; Li et al., 2006; Merkeev et al., 2006; Wheeler et al., 2008). Although all hybrid methods must incorporate phylogenies of some form, they are not required to use CGN (Figure 2.1). Because the hybrid approaches combine tree and graph-based methods by using the phylogenetic resolution of the former and the scalability of the latter, they are suitable for genome-wide analyses. Besides the advantages, one must be aware of which of these methods do not provide a phylogenetic resolution at multiple levels in *de novo* generated OGs (Dehal and Boore, 2006; Wheeler et al., 2008).

Box 3. Hybrid methods

Ensembl Compara
The database provides comparative genome and proteome data for more than 30 eukaryotic species, mainly mammals (Hubbard et al., 2009). The orthology prediction pipeline combines both BLAST-based RBHs and a phylogenetic tree reconciliation. **Pros:** The orthology uses a phylogenetic approach for handling gene losses. Orthologous relationships are labeled as one-to-one, one-to-many and many-to-many. Moreover, additional orthologs can be inferred in the genome context using whole-genome alignments. The Ensembl Compara database is regularly updated, freely available and accessible through several interfaces (http://www.ensembl.org/). **Cons:** The approach does not consider alternative transcripts for a gene, but assumes that a gene is best represented by the longest transcript or translation product.

HomoloGene
The HomoloGene database provides automatically predicted homologs of 19 completely sequenced eukaryotes (animals, plants and fungi) and includes cross-references to other resources on experimentally verified protein functions, conserved domains and phenotypic data (Wheeler et al., 2008). The clustering procedure uses precomputed BLAST protein similarities and CGN and is guided by a species phylogeny (starting from closely-related species). Aligned protein sequences are linked to their corresponding DNA sequences, from which non-synonymous-to-synonymous nucleotide substitution ratios are calculated to prevent inclusion of out-paralogs into groups. Paralogs are identified as sequences that are more similar within species than between species. **Pros:** HomoloGene groups are constructed using explicit species phylogeny and CGN and do not group unrelated proteins together in the presence of a hybrid protein. The database is regularly updated and freely available (http://www.ncbi.nlm.nih.gov/homologene/). **Cons:** HomoloGene groups are exclusive and lack plausible function annotations (only labeled by the last common ancestor of group members). The clustering procedure is not available.

OrthoParaMap (OPM)
The OPM package integrates comparative genomic positional databased on BLAST comparisons and gene phylogenies to infer evolutionary processes in gene families from two species (Cannon and Young, 2003). Unlike tree-reconciliation methods, OPM does not use a species tree but a conserved gene neighborhood (CGN) to decide upon speciation and duplication events. **Pros:** OPM incorporates CGN and distinguishes between segmental and tandem duplicates. The program is freely available (http://www.tc.umn.edu/ cann0010/Software.html). **Cons:** OPM cannot be used for more than two genomes simultaneously.

Phylogenetically inferred groups (PhIGs)
The PhIGs database provides protein clusters, protein family trees and synteny maps for 23 completely sequenced genomes of fungi and metazoans (Dehal and Boore, 2006). Protein clusters are constructed using all-*versus*-all BLAST comparisons, calculations of protein distances from refined alignments and a hierarchical clustering guided by a species tree. A maximum likelihood protein family tree is inferred for each protein cluster. **Pros:** The clustering procedure takes species phylogeny into account. The web server provides visualization of synteny maps (http://phigs.org). **Cons:** Trees must be examined manually to infer speciations and duplications. The database has not been updated since its first release and is not available for download.

Phylogenetic orthologous groups (PHOGs)
The PHOG database stores clusters of orthologous groups (PHOGs) at various levels of the species tree from mainly prokaryotic genomes (Merkeev et al., 2006). PHOGs are constructed by traversing the species tree from the leaves towards the root and finding BBH-based BLAST hits for each pair of species (proteomes). Only the highest-scoring protein pairs (seeds) within newly created PHOGs are aligned by Smith-Waterman algorithm and used in the next iteration. **Pros:** The PHOG approach constructs orthologous groups at various levels using species phylogeny. It incorporates automatic detection and handling of fusion events in multi-domain proteins. **Cons:** The database server is not available online.

Phylogenetic orthology and paralogy (PhyOP)
The PhyOP orthology prediction pipeline explicitly handles multiple transcripts per gene to reliably infer orthology and paralogy relationships between genes for recently diverged species (Goodstadt and Ponting, 2006). First, clusters of transcripts are constructed using single linkage clustering based on BLAST protein similarities, protein-to-transcript mappings and synonymous nucleotide substitutions. In the next step, clusters are used to infer phylogenies of transcripts using a modified least-square distance-based method. A set of heuristic rules is applied to the phylogenies to detect orphan genes and to distinguish between functional genes and pseudogenes. **Pros:** The PhyOP pipeline takes multiple-transcripts per gene into account to predict orthologs. It can distinguish between functionally active and inactive genes (pseudogenes). Moreover, PhyOP is particularly useful in predicting orthologous genes for incomplete (draft quality) genomes. The program is available upon request. **Cons:** The PhyOP can only be used for two closely related genomes.

TreeFam
TreeFam is a database of curated (TreeFam-A) and automatically constructed (TreeFam-B) animal gene families, phylogenetic trees, inferred orthologs and paralogs for fully sequenced animal genomes (Li et al., 2006). First, TreeFam clusters are created by hierarchical clustering of all-*versus*-all BLAST similarities and then gene family trees are constructed using several different approaches including maximum likelihood and neighbor-joining. Orthologs and paralogs are inferred using the Duplication/Loss Inference (DLI) tree-reconciliation algorithm, which uses the taxonomy tree of NCBI as a species tree. **Pros:** The orthology prediction uses a phylogenetic approach for handling gene losses. Speciation, duplication and gene-loss events are displayed in the phylogenetic trees. Experts manually correct errors in the automatically constructed trees. All data and software can be freely downloaded (http://www.treefam.org/). Besides a web interface, users can access the TreeFam database directly. **Cons:** A gene is represented by one transcript.

2.3 Caveats of orthology detection

Mosaics of proteins

The fusion, fission, shuffling, gain and loss of protein domains are common processes in protein evolution, which give rise to protein chimeras or hybrids (i.e., a protein

that consists of at least two distinct, non-homologous sequence regions, either in the form of a single domain or as a full-length protein). Hybrid proteins can complicate orthology assignments in a way illustrated by the bifunctional dihydrofolate reductase-thymidylate synthase gene (*DHFR-TS1*) from *Arabidopsis thaliana* (Figure 2.3). Importantly, OGs delineated without considering the possibility of hybrids run the risk of containing proteins that do not have a common evolutionary ancestry. Clearly, a hybrid protein can be legitimately similar to more than one OG. Therefore, grouping proteins into overlapping (non-exclusive) OGs is likely to provide more reliable and informative gene trees and a more complete representation of phylogenetic and functional relationships among the proteins than exclusive grouping schemes (wherein a protein sequence is assigned to its most similar neighbors based on partial homology), which are used by most orthology detection tools. For example, the Resample Inference of Orthologs (RIO) and the Hierarchical Grouping of Orthologous and Paralogous Sequences (HOPS) databases consider protein domains as the basic units for orthology (domain-centric view) (Storm and Sonnhammer, 2003; Zmasek and Eddy, 2001), whereas the Phylogenetic Ortholog Groups (PHOG) database organizes proteins into overlapping OGs (protein-centric view) in which hybrid proteins are automatically flagged (Merkeev et al., 2006). Moreover, alternative splicing, errors in gene structures and low-complexity regions create problems analogous to those of hybrid proteins. Interestingly, the Phylogenetic Orthology and Paralogy (PhyOP) program is the only approach that explicitly handles genes with multiple transcripts during orthology detection (Goodstadt and Ponting, 2006). Although attempts have been made to solve the problems described above, most tools currently in use were designed for single-domain proteins; therefore, all orthology data might need additional manual refinements on a case-by-case basis.

Horizontal gene transfer

HGT is an important phenomenon in the evolution of prokaryotes and eukaryotes (Koonin et al., 2001; Lerat et al., 2005; Loftus et al., 2005). Genes inherited through HGT are known as xenologs (Hillis, 1994). A phylogenetic inference without awareness of xenologs often leads to confounding outcomes and might indicate, for example, very close phylogenetic relationships between two distantly related organisms that have recently exchanged a gene. Moreover, HGT introduces an additional problem in classification (i.e., xenologs must be distinguished from other types of homologs). None of the methods that are compared in Boxes 1–3 explicitly detects xenologs, which usually requires a cereful phylogenetic analysis taking phylogenetic incongruence, mobile elements, insertion and deletion patterns, and atypical sequence composition into account (Gupta, 2001; Sundin, 2007). Most methods that can infer HGT are only capable of detecting examples of recently acquired genes. To detect early HGT events, using the phylogenetic distribution of protein families across all domains of life might prove effective (Kunin et al., 2005; Kunin and Ouzounis, 2003).

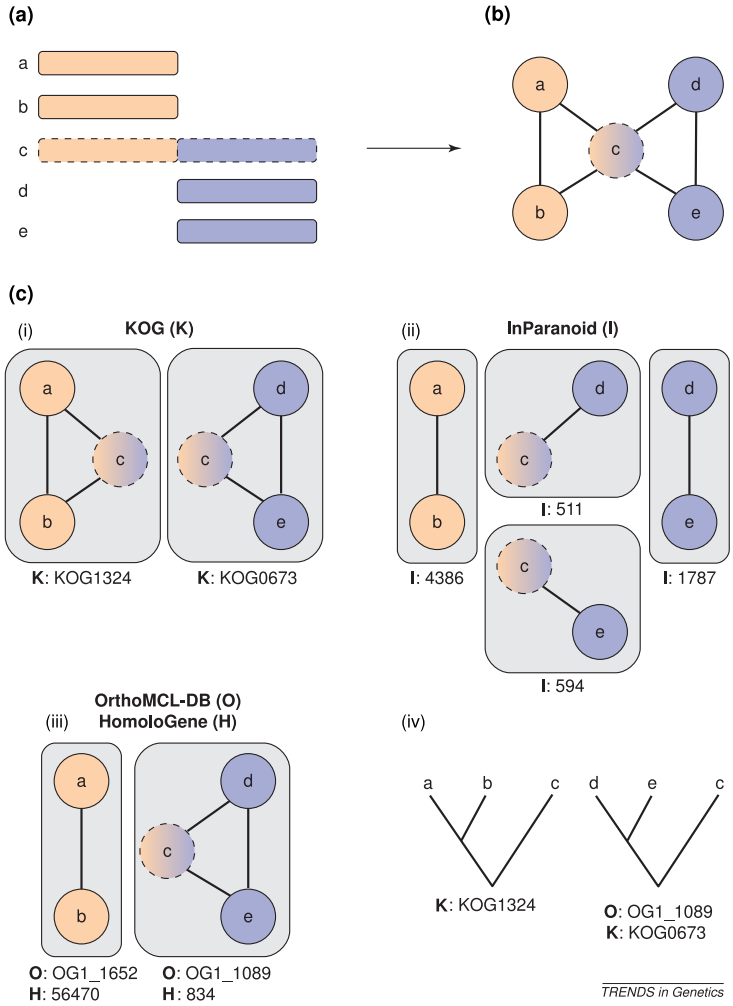
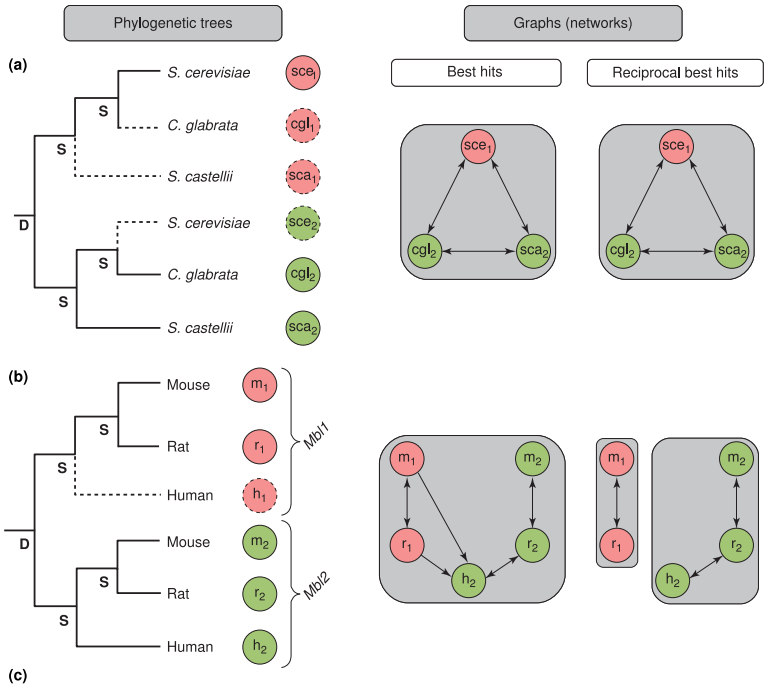


Figure 2.3. Partial homology to a hybrid (fusion) protein causes distinct orthologous groups to overlap. **(a)** The five proteins involved in overlapping [labeled (*a*–*e*)] are depicted as rectangles and grouped together into two overlapping groups (*a,b,c* and *c,d,e*), where protein *c* is the hybrid having partial homology to both groups. **(b)** The protein similarity graph of significant similarities between the proteins. Two phylogenetically unrelated protein groups are joined together. **(c)** Diagram illustrating how different databases handle the grouping of these proteins: (i) KOG (K); (ii) InParanoid (I); and (iii) HomoloGene (H), OrthoMCL-DB (O). In the current example, only the KOG database reflects the orthologous relationships between the proteins correctly, leading to a reliable inference of the protein phylogenies (iv). It should be emphasized that a phylogenetic gene tree cannot be constructed from the protein similarity graph in panel (b), because this group includes proteins that have no mutual sequence similarity at all [(*a,b*) *versus* (*d,e*)]. Graph nodes correspond to UniProt accessions: *a*, dihydrofolate reductase of *Drosophila melanogaster* (fruit fly), P17719; *b*, dihydrofolate reductase of *Homo sapiens* (human), P00374; *c*, bifunctional dihydrofolate reductase-thymidylate synthase 2 of *Arabidopsis thaliana* (thale cress), Q05762; *d*, thymidylate synthase of fruit fly, O76511; *e*, thymidylate synthase of human, P04818.



(c)

Tool	Method	Orthology prediction: Correct	Incomplete	Incorrect	Comment
Orthostrapper	Tree-based	✓			a
COCO-CL	Tree-based	✓			b
RAP	Tree-based	✓			b
Ensembl Compara	Hybrid	✓			
HomoloGene	Hybrid	✓			
Treefam	Hybrid	✓			
PhIG	Hybrid		✓		c
InParanoid	Graph-based	✓			
RoundUp	Graph-based	✓			
Reciprocal best hits	Graph-based	✓			
Ortholuge	Graph-based		✓		d
OrthoMCL-DB v2.0	Graph-based		✓	✓	e
EGO	Graph-based		✓	✓	f
Best hits	Graph-based			✓	g
OrthoMCL-DB v1.0	Graph-based			✓	h

TRENDS in Genetics

Figure 2.4. Comparison of orthology detection methods in the presence of gene losses. The relationships between genes are shown from a tree (left) and a graph (right) perspective. **(a)** A reconciled gene tree (midpoint rooted) of single-copy genes (general transcriptional co-repressors) from three yeast species (*Saccharomyces cerevisiae*, *S. castellii* and *Candida glabrata*) is inferred using known species phylogeny (for details, see Scannell et al. (2006)). Genes of *S. cerevisiae* and *S. castellii* are not orthologs but paralogs owing to the reciprocal gene loss in these species. The graph-based (nearest neighbor) approaches cannot distinguish between out-paralogs and orthologs (*sce1* is in one group with *cgl2* and *sca2*). **(b)** A reconciled gene tree (midpoint rooted) of mannose-binding lectin genes (experimentally verified) from mouse, rat and human. Both rodents have two ▶

▶ paralogous genes (*Mbl1* and *Mbl2*), whereas human has only one gene (*Mbl2*) owing to a single gene loss (Sastry et al., 1995). **(c)** The table summarizes the results of 15 different orthology prediction methods using the example of *Mbl1* and *Mbl2* genes. Orthology predictions are classified into three quality categories: (i) correct, the inference must be correct for all genes; (ii) incomplete, some orthologous relationships might be absent; and (iii) incorrect, out-paralogs and orthologs are grouped together (e.g., *Mbl1* gene in the *Mbl2* group). Meaning of the letters (a–h) present in the 'Comment' column: a, zebrafish is used as an out-group; b, default parameters are used; c, human *Mbl2* gene (protein) is apart from mouse and rat Mbl2 orthologs; d, mouse, rat and human Mbl2 orthologs (transcripts) are absent; e, human Mbl2 and mouse and rat *Mbl1* genes (proteins) are in one cluster (OG2_81338); f, human *Mbl2* and mouse *Mbl1* genes (transcripts) are in one cluster (#1119333); g, mouse and rat *Mbl1* genes link to paralogous human *Mbl2* gene; h, *Mbl1* and *Mbl2* genes (proteins) are in one cluster (OG1_4283). Graph nodes correspond to accessions: *sce1*, YBR112C (UniProt: P14922); *cgl2*, CAGL0D01364g (UniProt: Q6FWC0); *sca2*, 705.55; *m1*, UniProt: P39039, RefSeq: NM_010775; *r1*, UniProt: P19999, RefSeq: NM_012599; *m2*, UniProt: P41317, RefSeq: NM_010776; *r2*, UniProt: P08661, RefSeq: NM_022704; *h2*, UniProt: P11226, RefSeq: NM_000242.

Gene loss and 'incomplete' genomes

Gene losses in genomes are an important source of false-positive orthology predictions. An analysis of fungal genomes has indicated that, by incorporating the information of CGNs into orthology detection, approximately half of the predicted one-to-one orthologs are, in fact, out-paralogs owing to reciprocal gene losses (Scannell et al., 2007). Therefore, out-paralogs might erroneously be inferred as orthologs when true orthologs are physically absent. Given the two real examples of gene losses in Figure 2.4, it is demonstrated that, unlike tree-reconciliation, a graph-based approach cannot distinguish between orthologs and out-paralogs in the presence of multiple gene loss evens (Figure 2.4a). In another case of a single gene loss, however, some graph-based methods (e.g., InParanoid and RoundUp) and RBH can provide reliable orthology assignments, which are equivocal to those of all tree-based and most hybrid methods compared (Figure 2.4b,c). An out-group species is commonly used to identify false-positive orthologs. However, this has both advantages and disadvantages because the added sequence might provide extra resolution and specificity, but it might also decrease the sensitivity by removing authentic orthologs (Remm et al., 2001) (Figure 2.4). Similarly, using 'triangles' of best hits among three species is particularly disadvantageous for the Clusters of Orthologous Groups (COG) of proteins in which a gene of one species is lost because such COGs will, consequently, be discounted (Koonin, 2005). In principle, the tree-based methods are more robust in the presence of gene losses and varying rates of evolution than graph-based methods. This is as a result of the fact that the former group defines an orthologous relationship in the global context of all homologs and a well-established species phylogeny, whereas the latter considers pairwise nearest neighbor relations between genes from only two species. In other words, an orthology relationship must be defined in a given context, especially in terms of taxonomic sampling. However, even then, one cannot be completely certain that genes inferred as orthologs are in fact out-paralogs (Zmasek and Eddy, 2002). In two databases, namely Ensembl Compara and TreeFam, gene losses are addressed explicitly using reconciled trees (Hubbard et al., 2009; Li et al., 2006).

Semantics and limitations of phylogenetic concepts

How does the language used to describe the relationships between genes complicate matters? Orthologs and paralogs are defined with respect to one event of speciation and duplication, respectively, whereas terms such as co-orthologs, in-paralogs, out-paralogs, super-orthologs and ultra-paralogs reflect a particular sequence (pattern) of speciation and/or duplication events. In principle, new terms could be associated with some other patterns in a phylogenetic tree as well, but this would be impractical for large trees. Moreover, from a visual perspective, large trees are not suitable for retrieving a subset of genes with desired properties (e.g., a taxonomic coverage or a pattern). One way to approach this problem is to convert a gene tree into one that can facilitate these 'gene-centric' queries for large-scale genome studies; for example, by means of the hierarchical numbering of OGs (similar to the way enzymes are classified (IUBMB, 1992)) used by the COCO-CL and LOFT programs (Jothi et al., 2006; van der Heijden et al., 2007). Because the phylogenetic relationships are strictly non-transitive, an OG must always be hierarchical and defined with respect to the last common ancestor of the investigated genes (taxonomic position). In general, trees are sufficient for most evolutionary scenarios; however, the complex background of some sequences (e.g., mosaics of proteins or xenologs) requires another kind of representation, such as a graph (network), which, unlike a tree, accounts for many-to-many relations. Therefore, it seems reasonable to use both a tree and a graph (network) interchangeably in phylogenetic inferences, instead of using either one exclusively (Doolittle and Bapteste, 2007; Rivera and Lake, 2004).

'Gold' standards in benchmarks

Orthology methods can be judged using several criteria including phylogenetic congruence, functional conservation and computational complexity (e.g., scalability, runtimes or memory usage). These benchmarks are often hampered by several factors including lack of 'gold' standards, availability of results, heterogeneous datasets, taxonomic biases, differences in the underlying methodologies and sparse documentation of the methods (Sonego et al., 2007). Amidst the flood of raw data, reliable functional annotations have only been found for a few model organisms, making the extrapolation of the results to distant species difficult owing to the high level of sequence divergence. Some orthology detection tools perform better than others in predicting a particular kind of functional conservation (e.g., co-expression, pathways or proteinprotein interactions) using functional genomic data (Hulsen et al., 2006a). A common observation is that the tree-based orthology prediction methods generally exhibit low sensitivity and high specificity, whereas the graph-based methods show high sensitivity and low specificity (Alexeyenko et al., 2006; Chen et al., 2007; Storm and Sonnhammer, 2002). Of the graph-based tools, InParanoid and OrthoMCL perform best with respect to consistency of protein function and domain architecture (Chen et al., 2007; Hulsen et al., 2006a). In contrast to functional benchmarks, phylogenetic benchmark sets of true orthologous relationships between sequences are not available yet. Although several attempts have been made to provide manually cu-

rated and consolidated sets of orthologs, mainly of vertebrate species (Eyre et al., 2007; Li et al., 2006), the following issues, in our opinion, should be addressed systematically. First, orthology is a testable hypothesis about the evolutionary descent through speciation; therefore, the orthology detection tools should be evaluated using reliable species phylogenies in the context of known evolutionary processes. For instance, simulation studies of sequence (genome) evolution involving events of gene loss might be helpful in establishing reliable orthologous relationships (Fulton et al., 2006). Furthermore, CGN might be considered for another benchmark because most orthologs tend to be found in CGN, especially if the rate of genomic rearrangements is low (van der Heijden et al., 2007). Second, it is not clear how to construct alignments of distant homologs consisting of multiple domains in shuffled order and how to model sequence rearrangements such as domain fusions, fissions or losses in phylogenetic inferences (Sjölander, 2004). As a result, orthology is usually addressed using either a domain-centric or a protein-centric view. Third, orthology data cannot be exploited efficiently without thorough integration of sequence data from genomes to proteomes, distinguishing between *in silico* predicted from experimentally verified gene products and using standard and stable identifiers for database entries. Finally, standardized protocols, rules and definitions should be established and documented when manual curation is used to decide upon whether two sequences are orthologs or not.

Computation of orthologs

The large number of fully sequenced genomes raises several questions for further research, including the scalability of the orthology detection algorithms and the availability of reliable and up-to-date orthology databases (see pros and cons of the databases in Boxes 1–3). The scalability is only an issue if the number of genomes (proteomes) being compared at once is large, owing to high demands on computer resources. In fact, most graph-based methods are suitable only for pair-wise proteome comparisons (sometimes including an out-group). Clearly, these approaches do not consider all sequence data and phylogenetic information available, therefore, they are more error-prone than the tree-based methods. On the contrary, hybrid methods attempt to address the scalability and reliability by incorporating phylogenenies at various steps of the clustering process, and by using more species (genomes) to increase the reliability of orthology predictions. Therefore, fast and scalable sequence similarity search and clustering algorithms are essential for further inferences of orthologies in the hundreds of genomes available (Enright et al., 2002).

2.4 Recommendations and conclusions

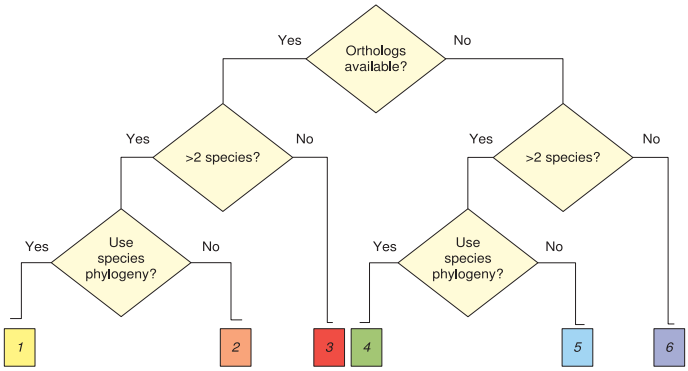
The basis for most current bioinformatics tools used to detect orthology relies on three major computational principles. The proposed classification aids researchers in recognizing the essential design principles and main attributes of newly developed orthology detection tools and in designing benchmarks by means of a careful analysis of the results. Although the different tools and approaches provide superior solutions for a variety of scenarios, the choice of methods depends on the purpose,

availability and phylogenetic background (e.g., number and diversity of species or known relationships between species) of OGs (Figure 2.5). When biologists are interested in identifying orthologs, they might want, for example, to find functionally equivalent genes (proteins) involved in a particular biological process (e.g., cell cycle) or metabolic pathway (e.g., lipid metabolism), to study fundamental processes and mechanisms of genome evolution (e.g., speciation, duplication or HGT), fate of genes and biological functions (e.g., gain and loss), or the genetic background of complex traits and inheritable diseases. Although this list is probably far from being complete, we propose the following guidelines to choose the appropriate tool. First, one should use publicly available databases of orthologs, query them with sequences (species) of interest and, upon the availability of orthologous sequences, decide whether to use the precomputed orthologs or to make the inferences partially (i.e., using post-processing programs) or entirely *de novo* (i.e., using *ab initio* programs). Several databases are available and updated regularly, including InParanoid, OrthoMCL-DB, Ensembl Compara, HomoloGene, TreeFam and HCOP (Chen et al., 2006; Eyre et al., 2007; Hubbard et al., 2009; Li et al., 2006; O'Brien et al., 2005; Wheeler et al., 2008).

In the next step, one should address whether the context of many species is important for the research or not, which also closely relates to the trade-off between sensitivity and specificity. If this is not a concern, then a graph-based (nearest neighbor) method is usually reliable for inferring orthologs between two closely related genomes, even in the presence of a single gene loss; otherwise, a tree-based method should be used for robust handling of multiple gene losses (Figure 2.4). Alternatively, multi-species OGs constructed by a graph-based approach can be used when the phylogenetic resolution is not required. Finally, if the phylogenetic relationships between species of interest are known, a choice should be made between a tree-based and a hybrid method, depending on the desired phylogenetic resolution of OGs.

Orthology detection methods seek to extend the limits of sequence comparisons by extracting information from sequence similarity networks and phylogenetic trees or by using auxiliary information of structural (conserved gene neighborhoods) and functional (ontologies) origins. Hybrid orthology detection methods, which have addressed several shortcomings of the tree-based and graph-based methods, are likely to provide enriched context of phylogenetic and functional relationships by using both a tree and a graph representation in the computation. The application of network propagation algorithms seems especially promising for detecting relevant functional relationships among proteins by incorporating various external sources of knowledge (Carroll and Pavlovic, 2006; Kuang et al., 2005; Noble et al., 2005).

At present, the number of published complete genomes approaches nearly 1000 (http://www.genomesonline.org) and hundreds more are being sequenced. The orthology detection tools reviewed here represent a valuable foundation and guide for further manual analyses. However, a scalable, fully automated procedure for inferring orthologs across genomes of all kingdoms of life still remains an elusive goal for current comparative genomics.



Database/Data files	1	2	3	4	5	6
COG & KOG		✓				
COGs refined by COCO-CL		✓				
COGs refined by LOFT	✓	✓				
EGO		✓				
Ensembl Compara	✓		✓			
HCOP	✓		✓			
HomoloGene	✓					
Inparanoid			✓			
KEGG orthology			✓			
Multiparanoid		✓				
OrthoMCL-DB		✓				
RoundUp			✓			
TreeFam	✓					
Programs	**1**	**2**	**3**	**4**	**5**	**6**
COCO-CL					✓	
Inparanoid						✓
LOFT				✓	✓	
Multiparanoid					✓	
OrthoParaMap						✓
Ortholuge						✓
OrthoMCL					✓	
Orthostrapper				✓		
PhyOP						✓
RAP				✓		
RSD						✓
SDI				✓		

TRENDS in Genetics

Figure 2.5. A decision tree for choosing the appropriate orthology detection tool. Databases and programs are listed in the table below the tree. Each tool is assigned (by a check mark) to a leaf in the tree, corresponding to a particular decision. Note: some tools are not listed here because of the limited availability or access.

Acknowledgments

The authors are grateful to Jack Franklin and Simon Fisher for their help in shaping the manuscript and to the anonymous reviewers for their valuable comments.

CHAPTER 3

PROGMAP: AN INTEGRATED ANNOTATION RESOURCE FOR PROTEIN ORTHOLOGY

Abstract

Current protein sequence databases employ different classification schemes that often provide conflicting annotations, especially for poorly characterized proteins. *ProGMap* (Protein Group Mappings, http://www.bioinformatics.nl/progmap) is a web-tool designed to help researchers and database annotators to assess the coherence of protein groups defined in various databases and thereby facilitate the annotation of newly sequenced proteins. ProGMap is based on a non-redundant dataset of over 6.6 million protein sequences which is mapped to 240,000 protein group descriptions collected from UniProt, RefSeq, Ensembl, COG, KOG, OrthoMCL-DB, HomoloGene, TRIBES and PIRSF. ProGMap combines the underlying classification schemes via a network of links constructed by a fast and fully automated mapping approach originally developed for document classification. The web interface enables queries to be made using sequence identifiers, gene symbols, protein functions or amino acid and nucleotide sequences. For the latter query type BLAST similarity search and QuickMatch identity search services have been incorporated, for finding sequences similar (or identical) to a query sequence. ProGMap is meant to help users of high throughput methodologies who deal with partially annotated genomic data.

3.1 Introduction

Functional annotation of new protein sequences is primarily a classification exercise that is based on searching several pre-classified protein or domain family databases (Finn et al., 2008; Hunter et al., 2009). Current databases use a variety of classification schemes and methods, and therefore the resulting protein groups (e.g., families or orthologous groups) and functional annotations provided may vary from database to database (Kuzniar et al., 2008; Liu and Rost, 2003). This problem is often encountered by users of high throughput methodologies especially when dealing with partially annotated genomes and poorly characterized proteins. Unifying and/or reclassifying the protein databases appears to be a plausible solution, however it also has major drawbacks. First, if properly done, this approach would require an effort equivalent to establishing and maintaining a new, curated protein database. Second, the individual classification schemes of the databases represent a very important added value which would go at least partly lost if we replace them with a new classification scheme. These problems led us to seek solutions that preserve all the information present in the underlying datasets and yet can be maintained in a largely automated fashion. ProGMap is a single-entry web-tool that unifies the classification information

of the current protein databases. Instead of creating a new classification scheme in which some of the expert knowledge used to construct the underlying databases would be inevitably lost, ProGMap combines the distinct classification schemes through constructing a network of links using a fast and fully automated hashing/mapping method originally developed for document classification (Rivest, 1992). Briefly, this algorithm converts sequences into unique 'message digests' or 'fingerprints' which can then be used for mapping sequences (identifiers) from various database rapidly. The purpose of ProGMap is 3-fold: (i) to provide a direct insight into the relationships among the various datasets through a single entry point, (ii) to refine and improve upon existing protein classification (clustering) methodologies, and ultimately, (iii) to gain better understanding of the concepts used for grouping proteins. ProGMap consists of a non-redundant dataset of over 6.6 million protein sequences which are mapped to 240,000 protein and group descriptions collected from UniProt (Consortium, 2009), RefSeq (Pruitt et al., 2007), Ensembl (Flicek et al., 2008), COG and KOG (Tatusov et al., 2003), HomoloGene (Wheeler et al., 2008), OrthoMCL-DB (Chen et al., 2006), TRIBES (Enright et al., 2003) and PIRSF(Wu et al., 2004). Looking up a query sequence or a group name in ProGMap provides information whether or not the underlying databases are in agreement on a certain term, and it also gives a plausible indication on how the conflicting annotations and/or group assignments could be improved. Therefore ProGMap is an annotation tool designed not only for database annotators, but also for users of high throughput methodologies such as microarrays or proteomics.

3.2 Methods

We used a centralized data warehouse approach implemented in a relational database (Oracle version $10.2g$) to store protein-to-protein, protein-to-group and group-to-group mappings as well as functional descriptions of proteins and groups. Specifically, these descriptions and mappings can be best pictured as nodes and edges in ProgMap's network, respectively. This network-based architecture enables queries to be made, for example, with distinct protein identifiers without explicitly specifying their type. For instance, queries such as HBA_HUMAN, P69905, NP_000549, ENSP00000251595 and 3039 used by UniProt, Refseq, Ensembl and EntreGene databases, respectively, yield identical results as they point to the same node within the network. The data used to build ProGMap (Table 3.1) were extracted from the source databases using our local Sequence Retrieval Server (SRS) (Etzold and Argos, 1993) as well as using modules written in Perl. Our goal is to keep the database up-to-date by following the regular updating schedule of the HomoloGene database (using only the odd-numbered releases). First, we constructed a non-redundant set of over 6.6 million protein sequences and cross-referenced them using a fast and reliable hashing/mapping method implementing the MD4 algorithm (Rivest, 1992). This algorithm was intended for digital signature applications such as for 'compressing' large files prior secure encryption. As the algorithm can take any string of characters and convert it into a unique 128-bit 'message digest' or 'fingerprint' in an efficient

Database	Supported identifiers	Data	URL
UniProt	• Protein ID (e.g., HBA_HUMAN) • Protein ACCESSION (e.g., P69905, P01922)	Proteins	ftp://ftp.ebi.ac.uk/pub/databases/uniprot/knowledgebase/
RefSeq	• Protein GI (e.g., 4504347) • Protein ACCESSION (e.g., NP_000549)	Proteins	ftp://ftp.ncbi.nih.gov/refseq/release/
Ensembl	• Translation ID (e.g., ENSP00000251595)	Proteins	ftp://ftp.ensembl.org/pub/
EnsemblCompara	• Family ID (e.g., ENSF00000005499)	Protein families	ftp://ftp.ensembl.org/pub/
HomoloGene	• RefSeq protein ACCESSION (e.g., NP_000549) • Protein GI (e.g., 4504347) • Entrez GeneID (e.g., 3039) • Official gene symbol (e.g., HBA1) • Group ID (e.g., 469)	Orthologous clusters of 20 eukaryotic proteomes	ftp://ftp.ncbi.nih.gov/pub/HomoloGene/
COG	• DB-specific protein ID (e.g., ampG) • Group ID (e.g., COG0477) (unicellular only) proteomes	Orthologous clusters of 66 prokaryotic and eukaryotic	ftp://ftp.ncbi.nih.gov/pub/COG/COG/
KOG	• DB-specific protein ID (e.g., Hs4504345) • Group ID (e.g., KOG3378)	Orthologous clusters of seven eukaryotic proteomes	ftp://ftp.ncbi.nih.gov/pub/COG/KOG/
OrthoMCL-DB	• DB-specific protein ID (e.g., hsa|ENSP00000322421) • Group ID (e.g., OG2_83619)	Orthologous clusters of 87 proteomes (both eukaryotes and prokaryotes)	http://www.orthomcl.org/common/downloads/
TRIBES	• DB-specific protein ID (e.g., MMUS-XXX-02-000372) • Group ID (e.g., TR-006821)	Protein families	http://cgg.ebi.ac.uk/services/tribes/ Website no longer supported.
PIRSF	• UniProt ACCESSION (e.g., P68871) • Group ID (e.g., PIRSF500045)	Protein families, subfamilies and superfamilies	ftp://ftp.pir.georgetown.edu/databases/pirsf/

Table 3.1. Database members and supported identifiers in the ProGMap database.

manner, we applied it for comparing protein sequences to each other as well as to group only sequences identical over the entire length into uniquely labeled 'Protein Identity Groups' (PIGs). Sequences which differ by a single (amino acid) residue give rise to different fingerprints (except for the N-terminal methionine which is disregarded) whereas sequences identical over the entire length share the same fingerprint. Each PIG corresponds to a unique protein sequence associated with various synonymous source databases' protein identifers (labels) and descriptions, therefore these are kept intact as present originally in the source databases. Importantly, the algorithm guarantees that no two distinct protein sequences produce identical message digests, and hence be members of the same PIG. Once the initial mapping was completed, group-to-group mappings were established through the process of translating the group members' identifiers into the unique keys and directly linking only those groups which shared at least one common member.

3.3 Results

Network of protein group mappings

The large databases underlying ProGMap were integrated using a centralized (data warehouse) approach to enable fast response to user queries. Once the datasets were downloaded and formatted according to ProGMap's database scheme, mapping these onto each other and building the Oracle database (16.4GB in total) took less than an hour on a database server with two Intel Xeon processors (4GB RAM). This fully automated mapping procedure resulted in a complex network of groups interlinked by one-to-one, one-to-many and many-to-many relationships. The resulting network of links on this centralized system enables functional as well as evolutionary information to be retrieved for many proteins being studied in high throughput experiments.

Web interface

The ProGMap database is equipped with a web interface that enables queries to be performed on the entire datasets (provided by the member databases) from a single entry point. The results are presented in both numerical and graphical forms. The web interface consists of six pages: (i) the 'About' page provides some background information about ProGMap; (ii) the 'Query' page is the main entry point for submitting queries; (iii) the 'BLAST' page enables protein or nucleotide sequences to be compared to the non-redundant ProGMap dataset using the BLAST algorithm (Altschul et al., 1997); (iv) the 'Quick Match' page is an interface to an exact protein sequence retrieval service which is much faster than a BLAST similarity search; (v) the 'Statistics' page summarizes the ProGMap's content in several tables and charts; and (vi) the 'Help' page. These pages were developed using Oracle's rapid application development environment (APEX version 3.1.1) which facilitates both easy maintenance and implementation of new features. The main 'Query' page offers eight predefined queries (Q0–7) using 'keywords', 'proteinID', 'groupID' or combinations of thereof. Importantly, valid gene symbols and database-specific identifiers (Table 3.1) can be used for querying ProGMap without the need to convert these into a specific type prior to searching the databases owing to its network-based architecture. Moreover, users can use batch mode to upload more than one query item in a space-delimited file. Once the results of a query have been retrieved, these can be saved in a text file or inspected visually using built-in graphical web tools. The ProGMap interface provides numeric and graphical tools for visualizing group-to-group relations. For example, the 'Group comparison matrix' (denoted as 'matrix' from here on) is available via the 'Compare Protein Groups' button (applicable to only some queries) in the upper left corner of the query results (Figure 3.1). Each cell in the matrix corresponds to a pairwise group comparison, and provides several measures shown in a bar chart and explained in the help message. This chart consists of three bars that indicate the extent of the overlap (coverage) of two groups A and B (denoted as CA and CB for groups A and B, respectively), as well as the similarity between them using the Jaccard index (denoted as J). This index equals to one for identical groups

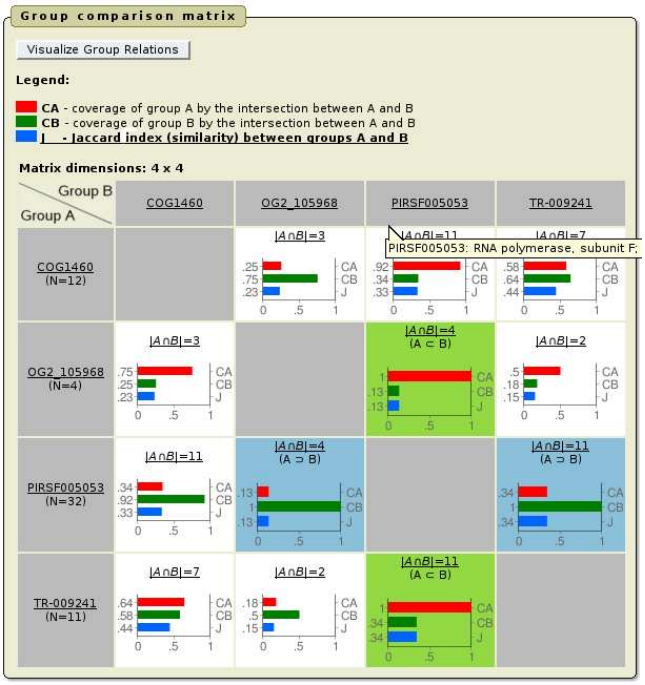


Figure 3.1. Comparing protein groups using the matrix comparison tool. Using an uncharacterized protein from *Methanococcus jannaschii* (RefSeq: NP_247002), ProGMap annotates this protein sequence as a 'RNA polymerase subunit F' on the basis of the manually curated PIRSF family (PIRSF005053). Although three other groups – wherein the protein is also found – do not provide plausible functional annotations (COG: COG1460; TRIBES: TR-009241; OrthoMCL-DB: OG2_105968), these, however, have more than one member in common as well as form either perfect (TR-009241 and OG2_105968) or nearly perfect subsets (COG1460) of the PIRSF family. The matrix comparison tool provides detailed information on set theoretic relations, per-group coverage (*CA* and *CB*, bars in red and green) and Jaccard index (*J*, bars in blue).

(that have all sequences in common) and equals to zero for non-overlapping groups (that do not share any common sequence). Additionally, the number of common members shared by two groups (intersection) and their set relations such as identity, superset and subset, are indicated in each non-empty cell. Another complementary visualization tool, which is available via the 'Visualize Group Relations' button in the upper left corner of the matrix (Figure 3.2), has been developed to gain a direct insight into the interlinked network of relations between protein groups. One can choose between three different network layouts, namely circle (default), spiral, or random, and adjust the representation of data to his/her own needs. The active nodes and edges (highlighted in red) are accompanied by hyperlinks to additional information about protein groups and relationships. The tools above have been developed using PL/SQL, scalable-vector graphics (SVG) and Javascript and have been extensively tested using the Firefox, Internet Explorer, and Opera web browsers.

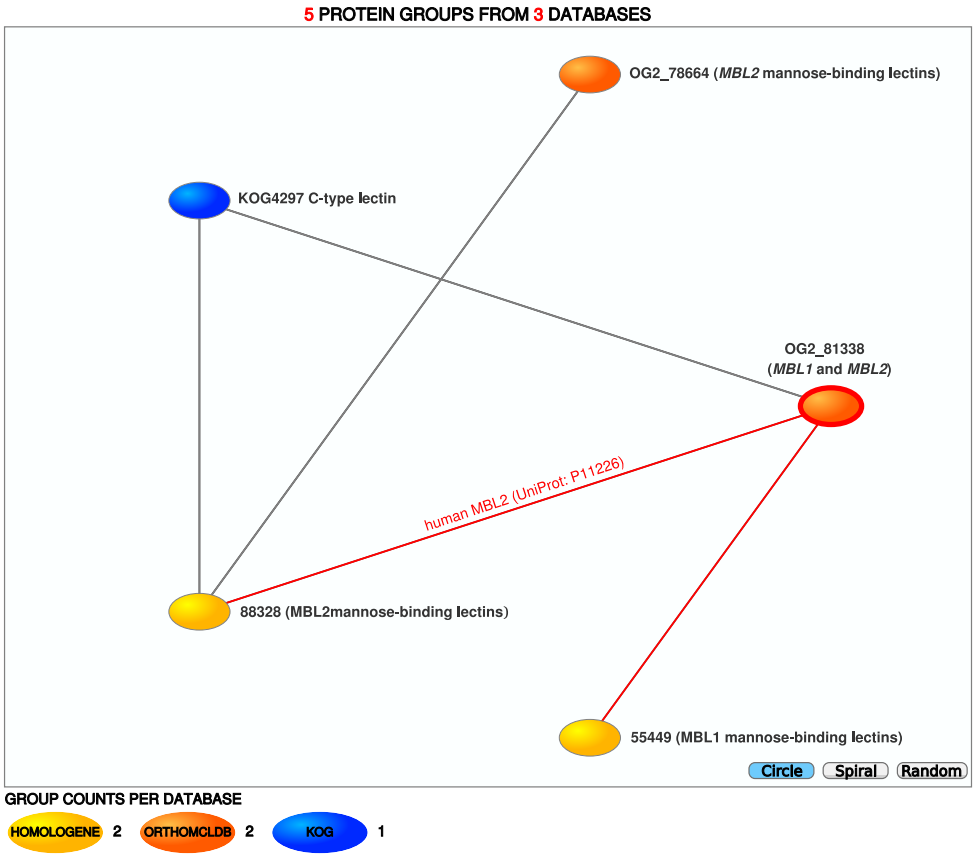
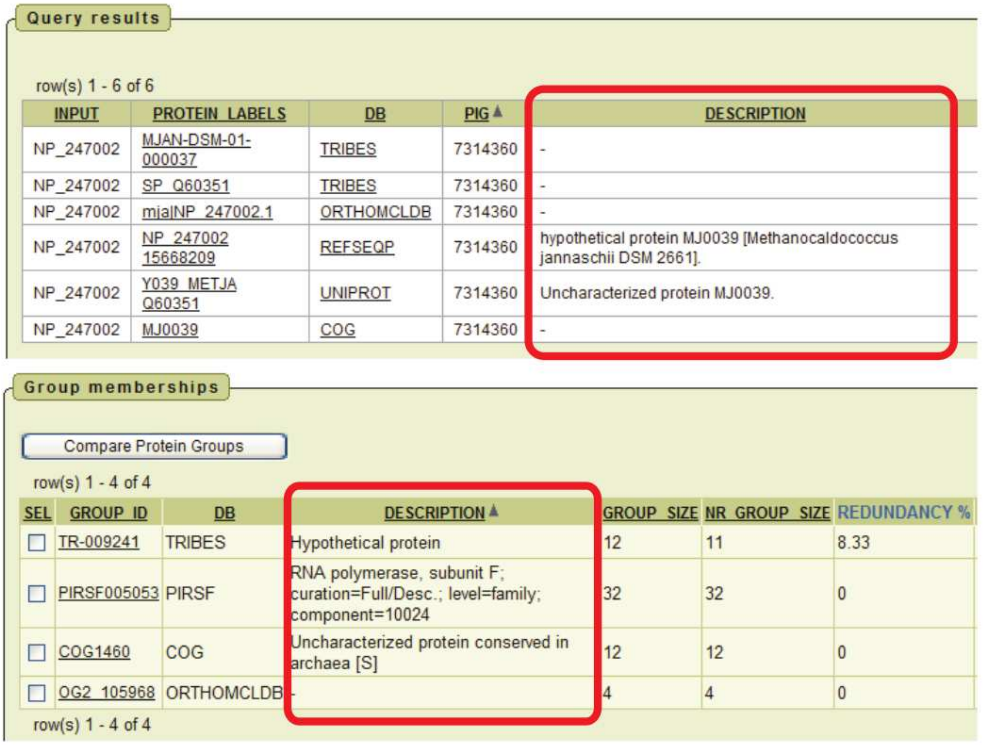


Figure 3.2. Comparing protein groups using the network visualization tool. The relationships among five orthologous groups of mannose-binding lectins (KOG: KOG4297; OrthoMCL-DB: OG2_78664, OG2_81338; HomoloGene: 55449, 88328). Groups sharing at least one protein are connected with an edge. In this particular example, the HomoloGene database (yellow) divides the lectins precisely into the two orthologous groups described in the literature (Phatsara et al., 2007; Sastry et al., 1995), whereas the other databases either combine them into one group (KOG, blue), or divide them differently (OrthoMCL-DB, orange).

Examples

If a protein sequence is found in the databases underlying ProGMap, submitting the sequence ID (using the 'Q6' option) will return all synonymous sequence IDs of the protein in ProGMap, along with the functional annotations. One can then view the groups into which this protein is classified in the various databases. Figure 3.3 shows an example of an ID-based query using an uncharacterized protein of *Methanococcus jannaschii* that is referenced in some of the databases; however a table of parallel annotations shows that only the PIRSF group was manually curated and provides a plausible biological function for the query protein. In practice, a list of protein



Query results

row(s) 1 - 6 of 6

INPUT	PROTEIN_LABELS	DB	PIG	DESCRIPTION
NP_247002	MJAN-DSM-01-000037	TRIBES	7314360	-
NP_247002	SP_Q60351	TRIBES	7314360	-
NP_247002	mjaINP_247002.1	ORTHOMCLDB	7314360	-
NP_247002	NP_247002 15668209	REFSEQP	7314360	hypothetical protein MJ0039 [Methanocaldococcus jannaschii DSM 2661].
NP_247002	Y039_METJA Q60351	UNIPROT	7314360	Uncharacterized protein MJ0039.
NP_247002	MJ0039	COG	7314360	-

Group memberships

Compare Protein Groups

row(s) 1 - 4 of 4

SEL	GROUP_ID	DB	DESCRIPTION	GROUP_SIZE	NR_GROUP_SIZE	REDUNDANCY %
☐	TR-009241	TRIBES	Hypothetical protein	12	11	8.33
☐	PIRSF005053	PIRSF	RNA polymerase, subunit F; curation=Full/Desc.; level=family; component=10024	32	32	0
☐	COG1460	COG	Uncharacterized protein conserved in archaea [S]	12	12	0
☐	OG2_105968	ORTHOMCLDB	-	4	4	0

row(s) 1 - 4 of 4

Figure 3.3. Finding functional annotations with ProGMap. A hypothetical protein query is submitted to the BLAST server that shows significant similarities with an uncharacterized protein from *M. jannaschii* (RefSeq: NP_247002) (output not shown). By submitting this entry to ProGMap, all the synonymous protein identifiers along with protein descriptions and links to protein groups are retrieved from the underlying databases. Only one of the databases, PIRSF assigns this protein to a curated family annotated as 'RNA polymerase subunit F'. The annotation of the PIRSF group indicates manual curation, which is an argument for accepting this tentative function. Although the group comparison view (Figure 3.1) shows that the databases are highly consistent with respect to this group (the groups are in nearly perfect agreement in all databases), the functional annotations are different for the groups compared.

(gene) IDs or names obtained from microarray or proteomics experiments can be submitted for ID-based search in batch mode to retrieve the corresponding proteins' annotations. If the protein ID is not found, there are two alternatives for submitting a query sequence: (i) searching for exact matches (no mismatches allowed) using the 'Quick Match' service, or (ii) use the BLAST algorithm to search for similar sequences in ProGMap.

In this case, one can simply select the desired entries by clicking on the top list and submit them to ProGMap for functional annotation. Some automatically inferred protein families contain conflicting annotations. For instance, the putative ENSF00000005499 family of Ensembl Compara is annotated as 'heat shock homolog

hsp20'. ProGMap allows one to compare this family to curated (reference) groups from other databases. The 'Q2' query returns four groups (without the query group), of which three are from PIRSF (PIRSF036514, PIRSF000228 and PIRSF002680) and one from TRIBES (TR-000776). The pairwise comparisons between these four groups and Ensembl's family show minimal overlaps, which include a small heat shock protein (hsp20), collagen and an NADH dehydrogenase subunit. In contrast, the TR-000776 family in TRIBES is functionally coherent when compared to manually curated PIRSF036514 (alpha-crystallin-related small heat-shock proteins) or KOG3591 (alpha crystallins), so the user has an option to choose. Reliable orthology detection is crucial, amongst others, for functional annotation of uncharacterized proteins (Kuzniar et al., 2008). ProGMap can also help in finding false positive orthology assignments (i.e., paralogs) in protein orthology databases. An example is the human mannose-binding lectin *MBL2* gene. Previous phylogenetic and functional studies showed that mannose-binding lectin proteins of vertebrates belong to two distinct orthologous groups (represented by *MBL1* and *MBL2* genes), which duplicated before the divergence of primates and rodents, as well as show tissue-specific gene expression (Phatsara et al., 2007; Sastry et al., 1995). Due to loss of the *MBL1* gene, humans retained only *MBL2*. Mannose-binding lectins can be retrieved by using the *MBL1* and *MBL2* gene symbols in the 'Q7' query of the ProGMap interface. This results in a list of 15 groups; orthology is explicitly shown only in five of the groups, OrthoMCL-DB (OG2_78664 and OG2_81338), HomoloGene (55449 and 88328) and KOG (KOG4297) so we compare these groups using the 'Q4' query. By examining these orthologous groups we find that only the HomoloGene database infers the orthologs of the mannose-binding proteins in agreement with the cited paper i.e., the out-paralogous families being separated into two distinct groups (Figure 3.2). OrthoMCL-DB's assigns human *MBL2* protein to the paralogous group OG2_81338 instead of the orthologous group OG2_78664. On the other hand, KOG4297 includes species at large phylogenetic distances.

Comparison with other tools

There are an increasing number of tools designed to interlink multiple databases and make the information available through single WWW entry points, among others MatchMiner (Bussey et al., 2003), SOURCE (Diehn et al., 2003), Harvester (Liebel et al., 2004), iHOP (Hoffmann and Valencia, 2004), IDConverter (Alibés et al., 2007), CARGO (Cases et al., 2007), YOGY (Penkett et al., 2006) and HCOP (Eyre et al., 2007). Some functionalities of ProGMap are also included in several other services. For example, some services including IDconverter enable queries to bemade using synonymous names or IDs for various genes (proteins); however, the relevant biological information can only be retrieved for a limited number of well-annotated eukaryotic genomes including human and mouse. In contrast, ProGMap includes all currently known proteins i.e., it covers all the kingdoms of life. At present only ProGMap includes a sequence similarity and an identity search service. Text searches using multiple keywords, gene symbols or protein IDs/accessions are supported by several other web portals including IDConverter, MatchMiner, SOURCE, CARGO and HCOP, but

in addition ProGMap allows full text queries to be combined using Boolean operators. Graphic presentation of query results is an integral part of ProGMap, CARGO, YOGY and iHOP. ProGMap is unique among these portals because it can directly compare protein groups in different databases, and thereby provide statistical support to annotation decisions.

3.4 Conclusions and perspectives

In this article we present ProGMap, a comprehensive mapping of the UniProt, RefSeq, Ensembl, COG, KOG, OrthoMCL-DB, HomoloGene, TRIBES and PIRSF databases that can be queried via a single interface (http://www.bioinformatics.nl/progmap). ProGMap is meant for users such as biologists and database annotators, who want to find the most probable functions for poorly characterized sequences, or want to assess the coherence between automatically inferred and expert curated protein families/orthologous groups. The ProGMap interface is freely accessible and presents the results both in numerical and graphical form. Future work includes the development of a web services-based interface suitable to link to high throughput pipelines.

Acknowledgements

The authors thank Hong Luo for expert help in the initial set-up of the Oracle database and Blaise Alako for stimulating discussions.

CHAPTER 4

GRAPH ALGORITHMS

4.1 Efficient search for similarity groups in large protein networks

Abstract

Graphs (networks) provide a powerful framework in which complex biological systems and processes can be modeled and better understood. With the deluge of high-throughput data, developing scalable and reliable algorithms for delineating meaningful similarity groups, such as protein families and orthologous groups, in large biological networks is one of the major interests in bioinformatics. However, freely available software cannot handle large networks, such as those encountered in large-scale proteome analyses owing to high demands on computer resources. We have implemented a straightforward, memory-efficient graph algorithm in the program called *netclust*, which can handle large biological networks constructed of hundreds of proteomes. This command-line tool is fast and scalable; a network of more than 10^6 nodes and 10^8 edges can be analyzed within a few minutes on a standard computer. The netclust program is written in the C language, and is freely available (under the GNU GPL license) from http://www.bioinformatics.nl/netclust/ for Unix (Linux, FreeBSD, OSX) and Windows platforms.

Introduction

In recent years, the graph (network) theory as well as the algorithms involved have opened up new avenues in contemporary biology towards understanding the structure, function and evolution of complex biological systems (Barabási and Oltvai, 2004; Sharan and Ideker, 2006). Networks, or graphs in formal mathematical language, are widely used objects to model the cell's internal organization, in which individual nodes represent, for example, genes, proteins or biochemical compounds, and in which edges between nodes correspond to interactions of a certain kind. Consequently, some types of graphs are more suitable for a particular biological data than other. For instance, protein-protein interactions (PPI) can be modeled conveniently by an unweighted undirected graph, whereas pairwise sequence similarities between proteins can be represented by a weighted undirected graph. On the other hand, gene regulatory networks or metabolic pathways require directed (hyper) graphs to capture the directionality of edges, and hence of biological processes.

Many bioinformatics tools using graph-based machine learning algorithms have emerged to aid analyses and interpretations of these cellular networks (Aittokallio and Schwikowski, 2006; Huber et al., 2007; Larrañaga et al., 2006). In particular,

unsupervised methods such as clustering have become instrumental for inferring biologically sound groups consisting of similar proteins (or genes), in which members are likely to share common biological function, 3D structure and/or evolutionary origin (Kriventseva et al., 2001b; Kuzniar et al., 2008; Lee et al., 2007). With the deluge of genomic sequence data the scalability of such algorithms has become an increasingly important issue. Specifically, searching for protein similarity groups in a large network constructed of hundreds of proteomes is a difficult task owing to high demands on computer's resources as well as long run-times. To address this, we implemented a fast and memory-efficient graph algorithm in the program, netclust, which can delineate biologically meaningful protein groups from large similarity networks of more than 10^6 proteins (nodes) and 10^8 sequence similarities (edges) within a few minutes using inexpensive computer hardware. Our implementation outperforms significantly available software used in bioinformatics not only in memory requirements but also in run-times. Here, we focused particularly on graph-based software that can use the nearest neighbor approach (Duda et al., 2000). This approach has been used in many areas of genomic research owing to its computational simplicity, scalability, and most importantly, biological relevance (Eisen et al., 1998b; Enright and Ouzounis, 2000; Koonin et al., 2004; Krause and Vingron, 1998).

Methods

Algorithms

A score-based algorithm for finding similarity groups (e.g., connected components or cliques) in a sparse (un)directed graph using the nearest neighbor approach requires precomputed similarities or distances between nodes, and a procedure to merge the 'nearest' (sets of) nodes above a certain similarity threshold in an iterative manner. Table 4.1 lists freely available packages and stand-alone programs used in bioinformatics for this purpose. In principle, these can be classified according to two distinct algorithmic approaches. The first approach requires an entire graph to be stored in computer's memory prior to finding similarity groups in the graph by using either depth-first search (DFS) or breadth-first search (BFS) algorithms. Therefore, this approach (denoted as 'in core' approach from here on) can be very memory expensive, especially for graphs consisting of large numbers of edges. The second approach, which is used by the netclust program, does not store the entire graph in the memory (denoted as 'external-memory' approach from here on). Instead, only clusters are gradually built while reading the graph from a hard disk; therefore, the memory requirements can be significantly reduced from quadratic $O(N^2)$ to linear $O(N)$, where N denotes the number of nodes. This improvement was achieved using a family of well-known algorithms called UNION-FIND algorithms (UFA) (Tarjan, 1975). In the netclust program, we implemented the asymptotically optimal variant of UFA with nearly-linear time complexity of $O(E * \alpha(E))$ in the worst-case scenario (E and α denote the number of edges and the slowly growing inverse Ackerman's function, respectively) (Tarjan and van Leeuwen, 1984).

Specifically, this greedy algorithm involves three abstract operations namely make

Code	Software	Language	Approach
NET	netclust[1] (1.0)	C	external-memory
BCL	blastclust[2], NCBI-BLAST package (2.2.18)	C	external-memory
CLM	clmclose[3], MCL package (1.006)	C	in core
GVZ	ccomps[4], Graphviz package (2.2.1)	C	in core
BIO	Bio::Graph::SimpleGraph module[5] (1.12.4.1)	Perl (5.8.7)	in core
GBF	Graph module (BFS)[6] (0.81)	Perl (5.8.7)	in core
GUF	Graph module (UFA)[6] (0.81)	Perl (5.8.7)	in core
PBG	Boost::Graph module[6] (1.2)	Perl (5.8.7)	in core
OBF	our implementation	Perl (5.8.7)	in core
NEX	networkx module[7] (0.33)	Python (2.4.1)	in core
RBG	RBGL module[8] (1.14.0)	R (2.6.2)	in core

Table 4.1. Graph-based software used in the benchmark analysis.
[1] http://www.bioinformatics.nl/netclust/
[2] ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html
[3] http://www.micans.org/mcl/
[4] http://www.graphviz.org/
[5] http://www.bioperl.org/
[6] http://www.cpan.org/
[7] http://networkx.lanl.gov/
[8] http://cran.r-project.org/

a group, find a group membership for a node, and unite groups sharing at least one common member. Further, the algorithm uses an array of non-negative integers, in which nodes correspond to array indices and each preliminary groups is represented by a rooted tree. Each node (array index) has exactly one parent node (array value) in the tree, except the root node that points to itself. Two post-processing steps are introduced to retrieve similarity groups from an input graph. First, each preliminary group is compressed in such a way that all members of a group point directly to the same (representative) member of that group. Second, the resulting similarity groups are sorted by their sizes in descending order and labeled by increasing integer values.

Inputs and outputs

The netclust program takes an (un)directed weighted graph in the form of an edge list, in which each row records an edge of two interacting nodes and the edge's weight that quantifies the similarity or distance between them. As biological networks are usually sparse, storing a graph in this form is space efficient compared to other matrix formats. Moreover, an unweighted graph, such as used for modeling PPI, can also be analyzed once all the edge weights are set to either one or zero depending on whether similarity (default) or distance threshold is used, respectively. Before using the netclust program, a graph must be indexed using the *netindex* program, which generates two binary index files from the input graph. These binary files provide fast access to graph data in a machine-readable form, hence reducing the overhead caused by parsing text files. Two input parameters can be adjusted namely the weight type between node pairs (similarity or distance) and the weight cutoff, which provide control over the sensitivity and specificity of the results. Moreover, once a graph is

indexed, netclust can be applied on the same input graph using different threshold values. Finally, the results can be written either into a text file or into a standard output in two distinct space-delimited formats.

Benchmark analysis

We compared the run-times and memory usages of freely available bioinformatics software (Table 4.1) using both artificial and real biological networks. For each program these performance values were obtained using the Perl's 'Benchmark' module and the 'pmap' Unix/Linux program, respectively. In total eleven networks (Table 4.2) were constructed using Perl scripts and the BLAST program (version 2.2.17) (Altschul et al., 1997), of which eight were artificial (denoted as RN, T1–4, C1–3) and three were real biological networks (denoted as B1–3). In particular, RN is a random Erdős-Rényi graph (Erdős and Rényi, 1959), whereas T1–4 and C1–3 are non-random 'thread-like' and 'cliquish' graphs, respectively. B1–3 are protein similarity networks constructed of BLAST sequence similarities (Figure 4.1a–c). To obtain reliable run-times, each program was executed ten times on a particular network, and the resulting run-times were averaged over all executions. Most importantly, the clustering results of different software were compared to check whether the underlying algorithms indeed produced identical results for the same input data; as expected, the results were indeed identical. Finally, this benchmark analysis was conducted on a single desktop computer (Intel Pentium 4 3GHz CPU, 32 bit, 1GB RAM, 160GB SATA hard disk, SUSE Linux 10.0 operating system).

Graph	NG	NNG	NEG	NN	NE	File size
T1*	1	10^4	9,999	10^4	9,999	116KB
T2*	10	10^4	9,999	10^5	9,999 x 10	1.4MB
T3*	10^2	10^4	9,999	10^6	9,999 x 10^2	16MB
T4*	10^3	10^4	9,999	10^7	9,999 x 10^3	170MB
C1*	10	10^2	4,950	10^3	495 x 10^2	474KB
C2*	10^2	10^2	4,950	10^4	495 x 10^3	5.6MB
C3*	10^3	10^2	4,950	10^5	495 x 10^4	66MB
RN*	1,441	*NA*	*NA*	79,083	10^5	1.4MB
B1+	35,397	*NA*	*NA*	178,228	2,745,123	61MB
B2+	33,953	*NA*	*NA*	826,554	166,445,591	4.6GB
B3+	41,072	*NA*	*NA*	2,713,908	781,328,458	21GB

Table 4.2. Simulated and biological graphs (networks) used. Note: NG - number of groups; NNG - number of nodes per group; NEG - number of edges per group; NN - total number of nodes; NE - total number of edges; *NA* - not applicable; *simulated network of arbitrary nodes and edges; +protein similarity network of UniProt or Refseq proteins and BLAST similarities.

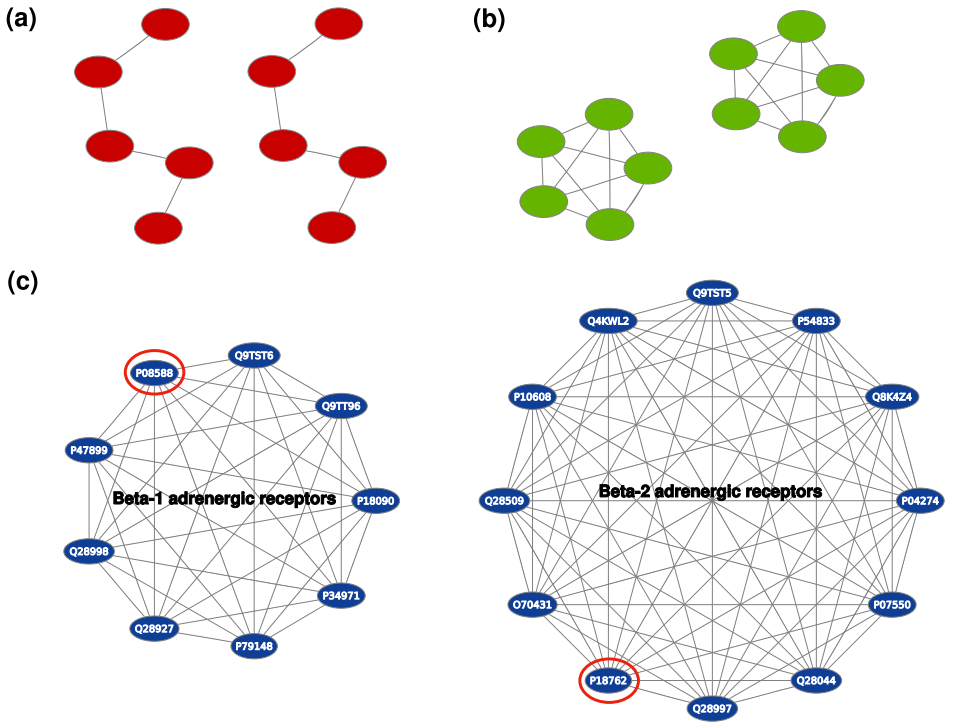


Figure 4.1. Examples of similarity groups in artificial and biological graphs (a–c). The 'thread-like' **(a)** and 'cliquish' graphs **(b)** consisting of arbitrary nodes and edges. The B1 protein similarity network includes two similarity groups **(c)** that correspond to beta-1 and beta-2 types (subfamilies) of adrenergic receptors (where nodes are protein accessions from UniProt, and edges represent significant BLAST similarities). Of the proteins only two were verified experimentally at protein level (in red circles).

Results

Benchmark analysis

Table 4.3 summarizes the run-times and memory requirements of the software used for finding groups in both artificial and real protein similarity networks. These benchmark results clearly showed that netclust outperforms significantly any other available software both in memory usage and in computation time. Generally, most programs were unable to process networks larger than 10^6 nodes and 10^6 edges due to overflown computer's memory. For example, the (Bio)Perl's 'Graph' modules performed poorly in run-times as it took about 3 hours to process a network as small as 10^5 nodes and 10^5 edges. Unexpectedly, the RBGL module written in R was the worst one in terms of memory usage amongst the software compared. On the other hand, the Python's 'networkx' package, MCL's 'clmclose' program and our own BFS-based implementation in Perl were amongst the fastest in-core programs, ranking directly

Code	T1	T2	T3	T4	C1	C2	C3	RN	B1	B2	B3
NET	0.3s 0.5MB	0.6s 4MB	4.7s 38.3MB	1m 12s 381.7MB	0.5s 0.3MB	1.5s 0.5MB	15.1s 4MB	0.8s 3.3MB	0.5s[a] 7.7MB	2m 13s[a] 32.3MB	4m 47s[a] 104.6MB
BCL	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	*NA*	1m 20s[a] 17.4MB	3m 42s[a] 39.7MB	17m 40s[a] 152MB
CLM	2.8s 3.8MB	23.2s 12.7MB	3m 47s 115.7MB	(32m 29s) (1.1GB)	0.6s 2.7MB	3.9s 9.2MB	39s 81.6MB	1m 31s 11.4MB	26.8s 40.5MB	[c]	[c]
OBF	0.4s 5.1MB	2.2s 39.5MB	20s 376.7MB	(3m 12s) (7.1GB)	0.7s 5MB	4.7s 36.9MB	47.7s 356.3MB	2.2s 35.1MB	22.5s 173.4MB	(24m 34s) (14.3GB)	[b] (>32GB)
NEX	1.3s 5.7MB	2.5s 28.3MB	23.3s 266.3MB	(13m 49s) (2.8GB)	1s 12.6MB	5s 97.2MB	50.3s 943.3MB	2.4s 28.3MB	27.7s 237.3MB	(36m 24s) (21.4GB)	[b] (>32GB)
PBG	0.9s 10.6MB	7.2s 90.3MB	1m 8s 873.6MB	(8m 25s) (16.4GB)	2.3s 13.9MB	23.5s 126.3MB	(2m 16s) (2.3GB)	6.7s 67.5MB	2m 4s 1GB	[b] (>32GB)	[b] (>32GB)
GUF	2s 4.1MB	21.5s 116.6MB	(1m 40s) (2GB)	(18m 28s) (21GB)	4.3s 22.8MB	42s 189.3MB	(4m 26s) (3.5GB)	14.3s 92.5MB	(2m 27s) (1.2GB)	[b] (>32GB)	[b] (>32GB)
GVZ	39.1s 3MB	9m 25s 18.1MB	1h 49m 169.1MB	(22h 46m) (3GB)	2.4s 6.9MB	24.6s 53.9MB	7m 10s 535.2MB	1h 45m 23.5MB	23m 21s 327.8MB	[b] (>32GB)	[b] (>32GB)
RBG	5.3s 38.1MB	42s 311.6MB	(8m 41s) (5.1GB)	[b] (>32GB)	10.6s 94.2MB	2m 6s 787.1MB	(40m 12s) (13.6GB)	45.6s 297.7MB	(5m) (2.4GB)	[b] (>32GB)	[b] (>32GB)
BIO	10m 13.6MB	1h 45m 72.9MB	17h 49m 1GB	(>24h) (>12.5GB)	5.4s 31.6MB	56s 294.5MB	(5m 33s) (5.3GB)	9h 44m 109.6MB	(6m 11s) (2.2GB)	[b] (>32GB)	[b] (>32GB)
GBF	1m 44s 23.3MB	2h 42m 134.3MB	(>24h) (>2GB)	(>24h) (>20GB)	4.5s 23.5MB	2m 15s 197.1MB	(3h 21m) (3.6GB)	2h 1m 152.4MB	(7h 50m) (1.5GB)	[b] (>32GB)	[b] (>32GB)

Table 4.3. Benchmark results of the available graph software show the superior performance of the *netclust* program. Run-times are indicated in seconds (s), minutes (m) or hours (h), and memory usages in megabytes (MB) or gigabytes (GB). Benchmark graphs are denoted as T1–4, C1–3, RN and B1–B3. *NA* - not applicable; () - the run-time or memory usage was obtained using a 64-bit computer with 32GB of memory; [a]the run-time was obtained from precomputed and indexed BLAST data; [b]the run-time was not determined due to the memory overflow; [c]the run-time was not determined because the matrix loading terminated prematurely at memory usage of 3GB.

below netclust (running times for a network of 10^6 nodes and 10^6 edges range between 20 seconds and 4 minutes). Nevertheless, the only programs suitable for the largest networks tested (more than 10^6 nodes and 10^8 edges) were netclust and BLASTClust. Moreover, netclust was more than 100 times faster, as well as required 56% less amount of memory than BLASTClust, in particular when the B1 network was used. The superior run-time performance of the netclust program, however, become less significant when larger networks (B2 and B3) were analyzed.

Biological relevance

Grouping similar protein sequences into protein families or orthologous groups provides efficient means to study the structure, function and evolution of proteins for many sequenced genomes available (Kriventseva et al., 2001b; Lee et al., 2007). For example, if an orthologous group consists of several uncharacterized proteins and at least one known protein, then the experimental knowledge available for this protein can be used to predict the function of the other proteins of that group reliably (Kuzniar et al., 2008). Therefore the biological soundness of tentative protein similarity groups can be truly assessed only with an objective external criterion that relies on a prior biological knowledge present in protein databases such as Uniprot. Figure 4.1c shows a biological example of two similarity groups in the form of 'cliques' that correspond to beta-1 and beta-2 types (subfamilies) of adrenergic receptors (belong to a large family of G-protein-coupled receptors or GPCRs). GPCRs include cell-surface receptors which are important in signal transduction processes, which are a major target for drugs (Alkhalfioui et al., 2009).

Discussion and conclusions

Delineating meaningful groups in large biological networks, such as those constructed of hundreds of proteomes, is a non-trivial task which requires a reliable similarity (or distance) measure, scalable algorithm, and validation method. Here, we describe a fast and memory-efficient tool, netclust, that can detect biologically sound protein groups in large protein similarity networks using the nearest neighbor linkage criterion. This software implements a straightforward and efficient graph algorithm that has been known in computer science for many years, but have not been used in bioinformatics to address biological problems, such as large-scale protein family detection. In contrast to netclust, memory-based (denoted as 'in core') methods, which store an entire network in the computer's memory, are not suitable for large-scale analysis.

Our empirical benchmark analyses on artificial and real biological networks confirmed the theoretical advantages of the algorithm used in the netclust program. As our program keeps most of the input data on a hard drive and hence saves a lot of memory space, the size of datasets is not a limiting factor (Chiang et al., 1995). Although the BLASTClust program is a reliable and efficient sequence clustering software, used successfully in several comparative genome studies (Horan et al., 2005; Koonin et al., 2004), however its use is limited to BLAST similarity networks, whereas netclust is generally applicable. Moreover, netclust improves upon the speed

and memory management in relation to BLASTClust. In principle, the netclust's performance could be further improved, for instance by using dedicated computer hardware such as larger RAM memory, faster hard disk or parallel access to data on disks, and thereby enabling interactive use (data not shown).

Besides the advantages, netclust, as any other nearest neighbor approach, is prone to the 'chaining' effect that may cause, for instance, non-homologous proteins sharing a common (promiscuous) domain or partial homology to be grouped together in one group. For this several graph-based solutions have been proposed such as using asymmetric similarity measures, transitive closure or graph-pruning (post-processing) procedures (Bolten et al., 2001; Jothi et al., 2006; Kawaji et al., 2004).

In summary, netclust is a fast and memory-efficient program that is suitable for exploratory analyses of large biological networks in real-time on an standard computer.

Acknowledgements

The authors would like to thank Pieter van Beek (SARA Computing and Networking Services, The Netherlands) for expert help in GRID computing for obtaining the BLAST data, Harm Nijveen and Pieter Neerincx for making the software running on the Windows and Mac OS X platforms, respectively, Anand Gavai for his assistance in R programming, Tao Tao for discussing some of the BLASTClust's parameters, Stijn van Dongen for the help with the MCL utilities, and the NBIC initiative for using the Dutch Life Science Grid platform.

4.2 Multi-netclust: A tool for finding connected clusters in multi-parametric data-networks

Abstract

Multi-netclust is a tool that can extract connected clusters of data represented by different network datasets given in the form of matrices. The tool uses user-defined threshold values to combine the matrices, and uses a straightforward, memory-efficient graph algorithm to find clusters that are connected in all or in either of the networks. The tool is programmed in C++ and is available either as a form-based or as a command-line based program running on Linux platforms. The algorithm is fast, processing a network of more than 10^6 nodes and 10^8 edges within a few minutes on a standard desktop computer.

Introduction

Finding tightly connected clusters in large datasets is a frequent task in many areas of bioinformatics such as the analysis of protein similarity networks, microarray data or protein-protein interaction data. Classical clustering algorithms have difficulties in handling large datasets used in bioinformatics. Fast heuristic algorithms have been developed for specific tasks; for example, BLASTClust from the NCBI-BLAST package (BLASTClust), CD-HIT (Li and Godzik, 2006) or the Tribe-MCL (Enright et al., 2002) can detect protein families in large networks of BLAST sequence similarities (Altschul et al., 1990). On the other hand there is apparently no bioinformatics tool that could efficiently handle large multiple networks, such as those necessary to group proteins according to distinct criteria Figure 4.2.

We developed a heuristic algorithm that takes the users' empirical knowledge of cutoff values into account below which interaction or similarity data can be neglected. As a result, multiple thresholded datasets can be combined together using an averaging or kernel fusion method (Kittler, et al., 1998). The resulting combined network can then be queried for connected components using an efficient implementation of the UNION-FIND algorithm (Tarjan, 1975), which correspond to groups of nodes that are connected either by any or by all of the constituent networks, depending on the form of the weighted averaging used (Figure 4.2). In order to adapt this method to large heterogeneous datasets, we combined the thresholding, aggregation as well as connected component search into a single, memory and time efficient tool, Multi-netclust that uses external-memory (Chiang et al., 1995) for matrix manipulations so that the size of the datasets is not a limiting factor.

Multi-netclust

The input to Multi-netclust are network data given in sparse matrix format, as well as the weight and threshold values associated with each matrix. The data can be entered either via a CGI interface, or from the command line. The output of Multi-netclust is a list of the connected clusters given in a structured text format. Multi-netclust is written in the C++ language, the CGI interface is a Perl script. The

code, sample datasets, explanations and performance data are available on the website http://www.bioinformatics.nl/netclust/. There is also a web-based application suitable to run smaller test-sets.

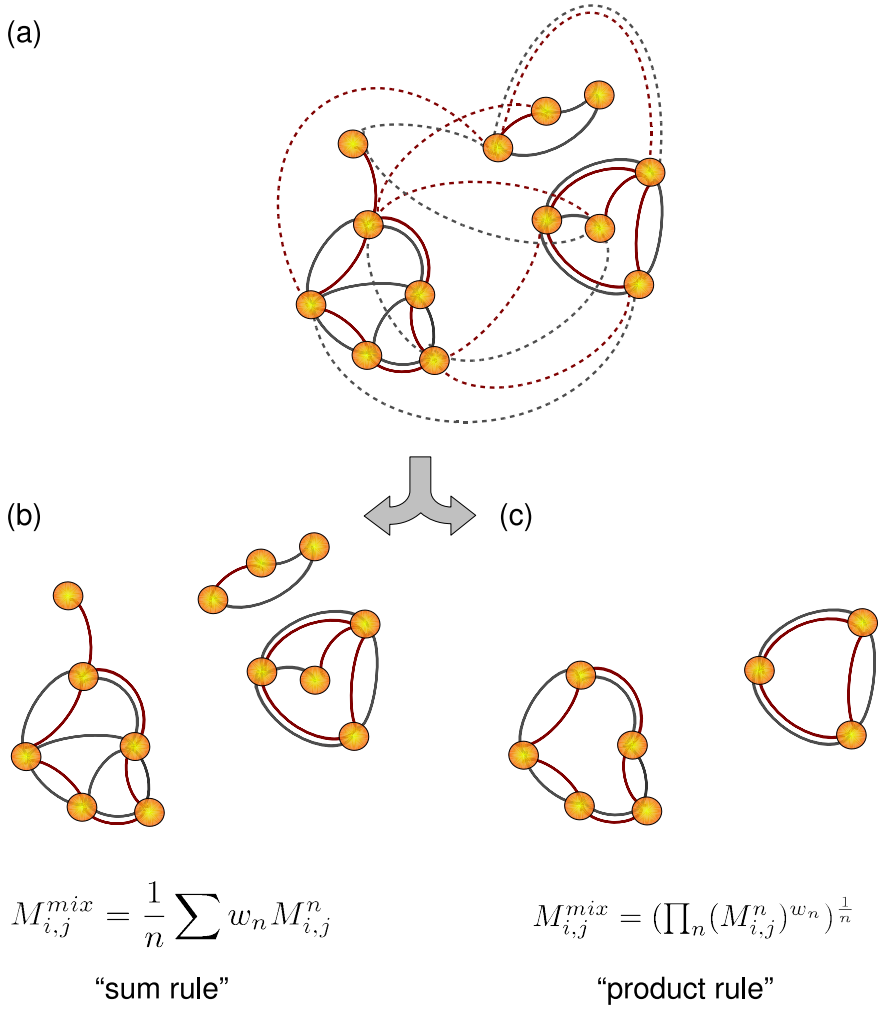


Figure 4.2. The principle of Multi-netclust is illustrated on a two-parameter network or hypergraph **(a)** consisting of red and gray edges. Dotted lines denote edges that are below the respective threshold and hence are omitted from the networks. **(b)** Aggregation by weighted arithmetic averaging (“sum rule”) gives connected components that are connected within either of the two networks. **(c)** Aggregation by weighted geometric averaging (“product rule”) gives connected components connected within both networks. $M_{i,j}$ denotes the value assigned to the edges, w is the weighting factor of the two matrices, and in the above example n=2.

Performance

The CPU-time of Multi-netclust subsumes the (i) preprocessing time needed for reading-in the data, thresholding and aggregation (more than 99%), and (ii) the time for finding the connected components and writing the results (less than 1%). A benchmark dataset of 1357 proteins, taken from the Protein Classification Benchmark database (Sonego et al., 2007) was used to combine BLAST sequence similarity and DALI 3D structure similarity data (Holm and Sander, 1995). The analysis took 4 seconds on a 2 GHz processor, the influence of thresholds on the purity of connected clusters is apparent from the data (Table 4.4). An interesting example is the immunoglobulin (Ig) superfamily (b.1.1), which has 125 members in the benchmark dataset. Using DALI alone as an input, clusters them with the E set domains (b.1.18), which is an "Early" Ig-like fold families possibly related to the immunoglobulin and/or fibronectin type III superfamilies. With BLAST, they are clustered with a number of other superfamilies whereas, the combination of the two [BLAST (0.1) * DALI (0.4)] made 94% of the group cluster correctly.

The external memory-based, connected component search algorithm is fast as compared to single-linkage based clustering methods and in-memory graph algorithms used for similar purposes within the bioinformatics community; several benchmark results are given at the website. The strength of Multi-netclust is more obvious when we deal with large data that can not be handled with other algorithms. For example, a dataset of 2,713,908 nodes and 781,328,458 edges took less than 5 minutes on a standard desktop processor. Of the other algorithms tested (see case studies on the website), only BLASTClust was able to handle a dataset of similar size, however its use is limited to BLAST similarity networks (and at greater expense of CPU time and memory required), whereas Multi-netclust is generally applicable. To conclude, Multi-netclust is an efficient preprocessing tool that can aid exploratory analyzes of large biological networks using an ordinary computer. Specifically, the potential applications include any task where network data of heterogeneous sources are to be combined, such as merging microarray and protein-protein interaction data, or combining gene ontology data with various similarity data.

Dataset	Correct	Incorrect	Singletons
BLAST (0.1)	66	1,101	190
BLAST (0.4)	36	0	1,321
DALI (0.4)	790	475	91
BLAST (0.4) + DALI (0.4)	803	469	85
BLAST (0.1) * DALI (0.4)	888	0	469

Table 4.4. Combining network data using the product rule at different threshold levels. Note: the numbers in parenthesis denote the applied threshold; Correct = proteins connected only to members of the same superfamily; Incorrect = proteins connected to members of other superfamilies.

Acknowledgements

The authors would like to thank Pieter van Beek (SARA Computing and Networking Services) for help in providing additional BLAST data, Anand Gavai for his assistance in R programming, Stijn van Dongen for the help with the MCL utilities, and the NBIC initiative for using the Dutch Life Science Grid platform.

CHAPTER 5

SYSTEMATIC EVALUATION OF PROTEIN CLUSTERING METHODS

Abstract

Automated protein classification (clustering) is an important task in genome annotation projects and evolutionary studies. During the past years, several protein clustering programs have been developed for detecting protein similarity groups such as families or orthologous groups from large datasets. However, most programs have not been benchmarked systematically, in particular with respect to the trade-off between computational complexity and biological soundness. We evaluated systematically three distinct algorithms to find out which one can scale to hundreds of proteomes and still delineate high quality similarity groups using minimum computational resources. A partition-based approach was used to assess the biological soundness of predicted groups using known protein functions, manually curated protein and/or domain families and orthologous groups of expert curated databases. Finally, we introduce an efficient graph-based method that can be used to delineate protein orthologs into hierarchical similarity groups. This protein hierarchy not only captures the information contained in the expert classifications but also provides an enriched framework in which the functional and evolutionary relationships between proteins can be studied at various levels of specificity. The validity of this method is demonstrated on data obtained from 347 prokaryotic proteomes.

5.1 Introduction

Classifying proteins (or genes) of diverse species based on sequence similarity is one of the fundamental tasks for many genome-wide functional and evolutionary studies which depend on reliable delineation of protein similarity groups such as families, subfamilies, superfamilies or orthologous groups. Proteins in such groups share some degree of functional and structural similarity, and common evolutionary descent via speciation (called orthologs) or duplication events (called paralogs). Since the first release of the manually compiled collection of protein families, the PROSITE database (Hulo et al., 2008), many (semi-)automated protein classification methods, which differ in purpose, classification scheme (namely 'flat' *versus* hierarchical) and quality of predictions, have emerged over the past years. For example, semi-automated classifications such as COG/KOG (Tatusov et al., 2003), PIRSF (Wu et al., 2004) and Pfam (Finn et al., 2008) involve expert curation of tentative groups on a case-by-case basis, while other databases such as CluSTr (Petryszak et al., 2005), OrthoMCL-DB (Chen et al., 2006) and HomoloGene (Wheeler et al., 2008) rely upon fully automated methods. Moreover, efforts have been made to construct integrated databases such as InterPro (Hunter et al., 2009), CDD (Marchler-Bauer et al., 2009) or ProGMap (Kuzniar et al., 2009) that combine various methods with the purpose of providing more

reliable predictions than those of individual methods. In particular, unsupervised learning techniques such as clustering have become increasingly important for functional and structural annotation of proteomes or (meta) genomic sequences (Enright et al., 2002; Yeats et al., 2008; Yooseph et al., 2007), phylogenomics reconstruction of species evolution (Dunn et al., 2008), prediction of orthologous genes (Kuzniar et al., 2008), and remote homology detection (Bolten et al., 2001; Li et al., 2002), as well as for constructing non-redundant sequence databases (Suzek et al., 2007). In contrast to supervised approaches, protein clustering methods do not require prior knowledge of pre-defined classes, such as experimentally verified protein functions, hand-crafted protein or domain families, for grouping proteins meaningfully *de novo*. These methods typically use a precomputed protein similarity network (further denoted as PSN) constructed by all-*versus*-all sequence comparisons.

Protein clustering, however, is a difficult task that involves several decisions to be made about (i) a sequence homology detection algorithm such as BLAST or Smith-Waterman, (ii) clustering algorithm, (iii) sequence similarity measure, (iv) classification scheme, and (v) validation method. Although many different database search algorithms and the use of various sequence similarity measures have been evaluated for different purposes such as protein function prediction or remote homology detection, the best known protein clustering methods have not been benchmarked systematically, in particular with respect to the trade-off between computational complexity and clustering quality (Rahman et al., 2008). Moreover, studies comparing the reliability of various scoring schemes with respect to different protein clustering algorithms and scenarios are scarce (Joseph and Durand, 2009; Yang and Zhang, 2008). As a result, researchers usually rely upon *ad hoc* settings. This paucity of systematic benchmarks can be largely attributed to the lack of a fully automated validation method that is both reliable and computationally feasible for entire protein classifications.

In this study we compare three distinct and most widely used protein clustering methods to investigate which one provides the 'best' trade-off between scalability *versus* biological soundness, and consequently is most suitable for reliable analysis of hundreds of proteomes. For this, the protein knowledge available in various expert-curated databases such as UniProtKB/Swiss-Prot, ENZYME, PIRSF, Pfam and COG/KOG served as the 'gold' standard. Specifically, known protein functions, manually curated protein/domain families and orthologous groups were used to evaluate the biological soundness of protein similarity groups. These reference sets were also used to optimize the methods for a particular biological inference using different parameter settings and scoring schemes. In addition to the biological aspect of this study, we compared the algorithms also in terms of run-time and memory usage.

Finally, we introduce a fast and memory-efficient graph-based method that by partitioning a PSN delineates protein orthologs into meaningful hierarchically nested similarity groups. The resulting protein hierarchy not only captures the information contained in expert classifications but also provides an enriched framework in which the functional and evolutionary relationships between proteins can be studied at various levels of specificity. We also demonstrate the biological plausibility of this method on some well-known biological examples such as the globin superfamily and bacterial RNA-polymerase sigma subunits, as well as on 347 prokaryotic proteomes

(about 1.1M proteins).

5.2 Materials and methods

Figure 5.1 provides a schematic overview of methods and datasets involved in the benchmark analysis. We describe these in detail in the following paragraphs.

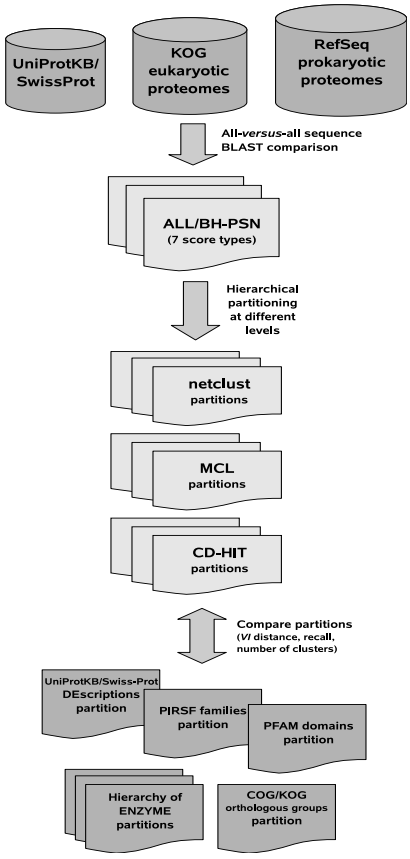


Figure 5.1. The schematic diagram illustrates the data flow involved in the benchmark analysis. Three protein clustering methods namely netclust, MCL and CD-HIT were evaluated on best known protein knowledge bases as well as optimized for reliable delineation of protein similarity groups (partitions). ALL- and BH-PSN refer to protein similarity networks constructed of all- and best-BLAST hits, respectively.

Collecting protein data

Protein sequences and entire proteomes were collected from three distinct databases namely UniProtKB/Swiss-Prot (Consortium, 2009), euKaryotic Orthologous Groups (KOG) database (Tatusov et al., 2003) and NCBI's Reference Sequence or Refseq database (Pruitt et al., 2007), which differ in size, quality of sequences and functional annotations, as well as in taxonomic sampling (Table 5.1). The first and smallest protein set encompasses expert curated (high quality) protein entries from both prokaryotes and eukaryotes wherein protein sequences and their functions were verified experimentally. The second and larger dataset consists of seven eukaryotic proteomes, including human, fruit fly or baker's yeast, in which proteins were classified into orthologous groups (KOGs). Although the KOG database has not been updated since 2003, it is still one of the most popular resources for functional annotation of new genomes because of its availability and expert curation. The third and largest protein collection has more than 1.1 million protein sequences (mostly predicted *in silico*) from 347 fully sequenced prokaryotic genomes.

Database	Release	Number of sequences	Number of proteomes	Phylogenetic coverage	Manual curation
UniProtKB/ Swiss-Prot	14.6/56.6	61,894	*NA*	Archea, Bacteria, Eukaryota, Viruses	Yes
KOG	year 2003	60,758	7	Eukaryota	Yes
Refseq	16	1,153,884	347	Archea, Bacteria	Partial

Table 5.1. Protein collections used for sequence comparisons differ in sizes, phylogenetic coverage and extent of manual curation. Note: *NA* - not applicable.

Constructing protein similarity networks

Similarities between protein sequences can be viewed as a weighted graph $G_w = (V, E_w)$ where nodes or vertices (V) correspond to proteins and weighted edges (E_w) denote sequence similarities between the proteins. Depending on the sequence similarity search algorithms used, the resulting pairwise sequence similarities (scores) can be either symmetric or asymmetric. For example, asymmetric scores are typically encountered in reciprocal BLAST comparisons. From here on we use the terms 'protein similarity network' (PSN) and 'graph' interchangeably.

For each dataset we constructed fourteen different PSNs (two graph types weighed with seven different similarity scores) by comparing the proteins using the BLAST algorithm (version 2.2.18; default settings) in the all-*versus*-all sequence manner. One of the PSN types is based on all BLAST hits (further denoted as ALL-PSN) whereas the other is based on best hits only (further denoted as BH-PSN). Comparing 347 prokarytic proteomes was a time intensive task that took about 30 days on a Linux computer farm (10 nodes with 2 CPUs each running in the Sun Grid Engine

environment). Once the BLAST comparisons were completed, seven BLAST-based similarity scores such as the percent of sequence identity, bit score or its threshold-filtered variant – bitcov60 score (at least 60% of the longer sequence must be aligned with the shorter one), raw score, log_{10}-transformed E-value (logE), self-normalized or norm score (S_{norm}) and common neighbor score ($S_{neighbor}$) were used to weight the edges of PSNs. Note, the latter two scores were calculated from the genuine BLAST raw score (S_{raw}) using (Equations 5.1 and 5.2). For example, the neighbor score was developed to take the shared network 'neighborhoods' of proteins into account by calculating the Jaccard similarity between protein pairs (Equation 5.3).

In principle, other similarity search methods such as FASTA (Pearson and Lipman, 1988), Smith-Waterman algorithm (Smith and Waterman, 1981), PSI-BLAST (Altschul et al., 1997) or HMM-based methods (Karplus et al., 1998) could be used instead; however, the BLAST algorithm was chosen due to its superior run-time performance particularly when large protein collections are being compared in all-*versus*-all sequence manner.

$$S_{norm}(A,B) = 100\ \frac{S_{raw}(A,B)}{S_{raw}(A,A)} \tag{5.1}$$

where A and B refer to sequences

$$S_{neighbor}(A,B) = S_{norm}(A,B)\ J(A,B) \tag{5.2}$$

where J denotes the Jaccard index

$$J(A,B) = \frac{N_A \bigcap N_B}{N_A \bigcup N_B} \tag{5.3}$$

where N_A and N_B correspond to sets of nodes in the immediate 'network neighborhood' of node A and B, respectively

Protein clustering algorithms

Automated protein clustering is an important unsupervised technique used to predict the biological functions and/or evolutionary relatedness of proteins, as well as to construct non-redundant sequence databases (Suzek et al., 2007). Typically, these methods require pre-computed all-*versus*-all sequence similarities in the form of a PSN prior to clustering. For our benchmark study, we selected three distinct algorithms used for automated protein classification. They are based on distinct clustering paradigms, and differ in computational complexity and clustering quality.

Graph-based Nearest Neighbor clustering algorithm

The first method is based on a well-known clustering paradigm of "nearest neighbor" or single-linkage (Bolten et al., 2001; Enright and Ouzounis, 2000; Krause and Vingron, 1998; Petryszak et al., 2005; Sibson, 1973) wherein the distance between two

groups is defined by the "nearest" (or most similar) data points. In other words, two groups are joined together only if at least one pair of data points, each of which belongs to a different group, passes through the distance (or similarity) threshold defined by a user. Despite this straightforward approach, most implementations available cannot handle large datasets due to high computational costs in terms of run-time and memory usage. To overcome these limitations, we have implemented a fast and memory-efficient graph algorithm in the netclust software, which can process very large PSNs, such as those constructed of hundreds of proteomes (see chapter 4 of this thesis). Although the underlying algorithm (Tarjan, 1975) has been known among computer scientists for long, it has not been applied for large-scale detection of protein similarity groups in PSNs. The netclust software (binaries and source files) is freely available under the terms of GNU Lesser General Public License at http://www.bioinformatics.nl/netclust/.

Graph-based Markov Cluster algorithm

The second algorithm, Markov Cluster algorithm (MCL) (van Dongen, 2000), is perhaps the most popular clustering method used in many areas of genomic research including functional genome annotation, phylogenomics (Dunn et al., 2008), and detection of protein (domain) families (Enright et al., 2002; Hubbard et al., 2009; Wall et al., 2008; Wong and Ragan, 2008) and orthologous groups (Kim et al., 2008; Li et al., 2003). The MCL algorithm detects clusters by simulating random (stochastic) walks within a graph while alternating two operators called expansion and inflation, hence the algorithm is very different from linkage-based methods. Moreover, this algorithm can handle multi-domain proteins explicitly by splitting up 'loosely-connected' proteins into smaller, 'tighter' groups – wherein proteins share similar domain architecture – by increasing the value of the inflation parameter. The MCL algorithm is reasonably fast, yet the size of a graph can be a limiting factor due to the quadratic space complexity of the algorithm. The MCL clustering software (release 06-058) is freely available at http://www.micans.org/mcl/.

Sequence-based CD-HIT clustering algorithm

A somewhat different approach is used by the CD-HIT algorithm, which was specifically developed for constructing non-redundant protein and nucleotide sequence collections by removing very similar (or identical) sequences from large sequence databases. The underlying heuristics uses a short word filtering rather than more time-consuming all-*versus*-all sequence BLAST comparisons to detect significant sequence similarities; such an alignment-free approach can then speed up the clustering process by a factor of two (100 times faster) than typical score-based clustering programs such as BLASTClust (Li and Godzik, 2006). Recently the CD-HIT algorithm was also used to detect protein families from millions of metagenomic sequences (Li et al., 2008). However, this algorithm cannot group together protein sequences with less than 40% identity due to the theoretical limit of the heuristic similarity search. Moreover, the algorithm cannot use a pre-computed PSN as input but re-

quires sequences instead. The CD-HIT clustering software can be downloaded from http://www.bioinformatics.org/cd-hit/.

Hierarchical partitioning the protein similarity networks through thresholding

The protein clustering methods used here typically yield a single or 'flat' partitioning of the similarity data; by using a series of different clustering thresholds on the same dataset we constructed a collection of partitions at different similarity levels (for further details see S1 of the Supplementary materials). Such partitions can be viewed collectively as a hierarchy of protein similarity groups wherein the proteins are grouped into families and superfamilies according to global and local (i.e., domain- or motif-based) similarities, respectively. In general, a higher (conservative) threshold will predict fewer homologous relationships between proteins while reducing the number of false positive relationships to a minimum (high precision and low recall). Conversely, a lower (permissive) threshold allows more homologous relationships to be detected but at the expense of more false positive relationships (high recall and low precision).

Computational complexity of the algorithms

Comparing algorithms in terms of both theoretical and empirical computational complexity provides important information that can guide a user in choosing the most appropriate software for processing a dataset with a particular size. We compared the three clustering methods in terms of run-time and memory usage on three datasets of different sizes using a standard computer (Intel Pentium 4 3 GHz processor, 32 bit, 1GB RAM, 160GB SATA hard disk, SUSE Linux 10.0 operating system). For this, we used the Perl's 'Benchmark' module and the 'pmap' Unix/Linux program.

Evaluating the biological soundness of protein similarity groups

An integral part of any clustering exercise is the actual validation of results using expert knowledge. Therefore the biological soundness of protein similarity groups constructed *de novo* cannot be truly assessed without an objective external criterion that relies on prior biological knowledge available in protein databases such as UniProt, ENZYME (Bairoch, 2000), COG/KOG, PIRSF or Pfam. These knowledge bases represent different biological aspects of proteins such as functional, structural or phylogenetic (Table 5.2). For example, the hierarchical Enzyme Commission (EC) numbering scheme, as used by the ENZYME database, classifies experimentally verified enzymatic reactions and enzymes into a four-level hierarchy. Specifically, the four digits of an EC number define the reaction specificity (class, sub-class, and sub-subclass numbers) and substrate specificity (serial number). We used three out of the four levels (except the class-level) for evaluating the clustering results. It is important to note that the EC classification is based on function rather than evolutionary relationships between the proteins (enzymes).

Reference partition	Class labels	Number of proteins	Number of classes	*VI* space
PIRSF-1	PIRSF protein families	12,620	1,985	9.44
ENZ-1	ENZYME (EC a.b.*.*) functions	14,558	60	9.59
ENZ-2	ENZYME (EC a.b.c.*) functions	14,558	205	9.59
ENZ-3	ENZYME (EC a.b.c.d) functions	14,558	2,166	9.59
SPROT*	UniProtKB/SwissProt functions	33,210	10,143	10.41
Pfam	Pfam domain families	35,224	3,211	10.47
KOG	KOG orthologous groups	60,758	4,852	11.01
COG+	COG orthologous groups	189,596	4,481	12.15
PIRSF-2+	PIRSF protein families*	221,222	11,154	12.31

Table 5.2. Reference partitions used for validating predicted protein similarity groups (target partitions). For each reference partition the numbers of proteins and classes, as well as the size of the *VI* metric space are given.
*The partition was derived from the DE line of protein entries.
+The partition was obtained by mapping its class members to 347 prokaryotic proteomes.

In practice, tentative protein groups are frequently evaluated manually by experts who use the knowledge of proteins for which molecular functions, functional/structural protein domains or motifs are known. Moreover, one can also inspect multiple sequence alignments and phylogenetic trees to find support for the proposed protein classification. Alternatively, the coherence between different, sometimes competing protein classifications can be inspected visually in an ensemble classification such as the ProGMap resource (Kuzniar et al., 2009). Validating entire protein classifications of thousands of similarity groups, however, requires a different, fully-automated approach that involves reliable measures and 'gold' standard sets with known class labels. In fact, such cluster validation techniques have been described in the data-mining literature, yet are rarely used in bioinformatics in general and protein classification in particular (Handl et al., 2005).

Comparing partitions

An automated protein classification is typically evaluated using external rather than internal criteria as the class labels of some proteins are usually known *a priori*. The idea of the external validation is to compare a new clustering (target partition) against a set of known class labels (reference partition), and then quantify the amount of (dis)agreement between them using a reliable distance or similarity measure (index).

Perhaps the most straightforward measures for cluster validation include the cluster purity and completeness; however, these provide only a limited amount of information about the relationships between two partitions, and hence render them less reliable than more comprehensive indices such as the F-measure (van Rijsbergen, 1979), Jaccard similarity or distance (Jaccard, 1901), Rand index Rand (1971), or the variation of information metric (*VI*) (Meilă, 2007). For example, perfect purity or compactness (100%) can be achieved trivially for a partition consisting of singleton clusters only (all-in-singletons, further denoted as AIS partition) or a partition in which all data points are in one large cluster (all-in-one, further denoted as AIO

partition), respectively. Therefore, a reliable index must provide a single estimate on how "close" two partitions are to each other in a clustering space delineated by the AIS and AIO partitions.

Selecting the 'best' method

We evaluated the clustering methods according to four distinct criteria: (i) the *VI* distance between target and reference partitions; (ii) the overall recall (*Rec*), which is calculated as the fraction of proteins in non-singleton groups; (iii) the number of similarity groups in a target partition (N_{SG}); and (iv) run-time and space complexity. An ideal method would simultaneously minimize the *VI* distance to zero, maximize the *Rec* value to 100%, approach the 'true' number of classes, as well as it would require a minimum amount of computational resources for data processing. In practice however, the methods show a trade-off between biological soundness and computational complexity.

The *VI* index is a true metric that measures the information exchange (in *nits* rather than *bits*; the former is based on natural logarithm or *ln*) namely loss and gain between a target and reference partitions (Equation 5.4). For example, the *VI* distance between two identical partitions equals to zero, and the distance between very distinct partitions namely AIO and AIS is no more than *ln(N)* (where N denotes the number of data points in the largest partition). Therefore, the *VI* distance space is always bound between the two extreme values (Table 5.2). We used the 'clmdist' program of the MCL package that provides an efficient implementation of the *VI* distance method for comparing partitions.

For each method the 'best' target partition was selected according to the *VI*, *Rec* and N_{SG} values. Consequently, we used the following scoring scheme to rank the methods according to biological soundness: a method was assigned (i) the maximum of three points if all three measures were the 'best', (ii) two points if *VI* and N_{SG} or *Rec* and N_{SG} were the 'best', (iii) one point if either *VI* or *Rec* was the 'best', and (iv) zero points otherwise; the higher the total sum of points the 'better' the ranking of the method.

$$VI(P_A, P_B) = H(P_A) + H(P_B) - 2I(P_A, P_B) \tag{5.4}$$

where H is the entropy of a partition and
I is the mutual information between partitions P_A and P_B

5.3 Results

Biological soundness of protein similarity groups

In this benchmark study, we evaluated the methods with respect to the biological soundness of protein similarity groups using the protein knowledge available in expert-curated databases (see Methods). This knowledge was stratified into nine reference partitions (namely SPROT, Pfam, PIRSF-1, PIRSF-2, KOG, COG, ENZ-1, ENZ-2, ENZ-3) that collectively account for the various biological aspects of proteins

such as molecular function, structure and common evolutionary descent. Surprisingly, our benchmarks suggest that the simple and computationally cheap nearest neighbor method, implemented in the netclust program (Kuzniar et al., submitted; chapter 4), performs nearly as good as, and in some instances, slightly better than the high complexity MCL algorithm (Table 5.3). For example, the netclust method approximated the manually curated PIRSF and Pfam families better than the MCL algorithm with respect to *VI* and N_{SG} values; however, the latter algorithm had slightly better *Rec* values. Overall, both algorithms performed most reliably on PIRSF and Pfam families, as well as on UniProtKB/Swiss-Prot descriptions (*VI* distances from 0.32 to 0.67 *nits*; *Rec* values from 95.53 to 99.25%). Additionally, the KOG and ENZYME (3^{rd} and 4^{th} levels of the EC hierarchy) classifications constructed *de novo* were also biologically meaningful but were of lower quality. On the contrary, none of the methods could predict the second-level ENZYME partition reliably (indicated by the *VI* distance larger than the AIO baseline of 3.24 *nits*). Furthermore, the CD-HIT algorithm performed significantly worse than the other methods over all validation sets tested. Nevertheless, the CD-HIT's clustering results were particularly of reasonable quality when validated against manually curated UniProtKB/Swiss-Prot descriptions and PIRSF families. Using the ranking scheme described above (see Methods section), the algorithms were ranked in the following order: MCL, netclust and CD-HIT scored 10, 7 and 0 points, respectively.

Knowledge base	**MCL**	**netclust**	**CD-HIT**	**'Best' method**
UniProtKB/Swiss-Prot	*Rec* (1)	*VI*; *Rec*; N_{SG} (3)	None (0)	netclust
PIRSF	*Rec*; N_{SG} (2)	*VI* (1)	None (0)	MCL
KOG	*VI*; *Rec*; N_{SG} (3)	*Rec* (1)	None (0)	MCL
Pfam	*Rec* (1)	*VI*; N_{SG} (2)	None (0)	netclust
ENZYME	*VI*; *Rec*; N_{SG} (3)	None (0)	None (0)	MCL
Total score	**10**	**7**	**0**	

Table 5.3. Benchmark results of clustering algorithms evaluated on different protein knowledgebases (reference partitions). The overall scores suggest a slightly superior performance of the MCL algorithm. In addition, for each reference the 'best' method with the maximum number of points (between parentheses) was selected. Note, there can be more than one method with one or more 'best' criteria such as the variation of information (*VI*), recall (*Rec*) or number of similarity groups (N_{SG}). The actual values of *VI*, *Rec* and N_{SG} can be found in the Table S2 of the Supplementary materials.

Computational complexity

The run-times and memory-usages of the three methods tested on different datasets are given in Table 5.4. The results show that netclust performs significantly better than the MCL or CD-HIT algorithms given that sequence similarities between proteins are already computed. Out of the methods compared, the netclust algorithm was computationally the most efficient using minimum amount of computer resources. Moreover, it was the only method that could handle the largest dataset

Algorithm	Complexity (time & space)	Run-time and memory usage		
		UniProtKB/ Swiss-Prot[1]	KOG[2]	347 proteomes[3]
netclust	$O(E\alpha(E))$	12s	40s	15m 22s
	$O(N)$	2MB	4MB	68MB
MCL	$O(N^3)$	3m 9s[a]; 5m[b]	27m 11s[a]; 32m 42s[b]	(>32GB)
	$O(N^2)$	86MB[a]; 107MB[b]	330MB[a,b]	*ND*
CD-HIT	non-linear	46s[c]; 3h 36m[d]	1m 35s[c]; 23h[d]	1h 43m[c]; *ND*[d]
	$O(N)$	219MB[c]; 88.5MB[d]	339MB[c]; 123MB[d]	2.2GB[c]; <1GB[d]

Table 5.4. Comparison of clustering algorithms with respect to time and space complexity. The netclust algorithm is more scalable than the other methods. Note: E - number of edges; α - inverse Ackerman's (slowly growing) function; N - number of nodes time is given in seconds (s), minutes (m), hours (h) or days (d); *ND* - the run-time was not determined due to memory overflow (>1GB) or the computation did not finish within one month.
[a] The value was obtained by clustering with inflation parameter set to 6.0.
[b] The value was obtained by clustering with inflation parameter set to 1.5.
[c] The value was obtained by clustering with identity threshold set to 100%.
[d] The value was obtained by clustering with identity threshold set to 40%.
[1] The dataset was used in the form of a PSN (consisting of 32,840 nodes and 2,262,455 edges; file size of 60MB) or protein sequences in FASTA (file size of 17MB).
[2] The dataset was used in the form of a PSN (consisting of 60,743 nodes and 9,976,221 edges; file size of 233MB) or protein sequences in FASTA (file size of 31MB).
[3] The dataset was used in the form of a PSN (consisting of 973,202 nodes and 136,808,511 edges; file size of 3.4GB) or protein sequences in FASTA (file size of 851MB.

of 347 prokaryotic proteomes on a computer with 1 CPU and 1GB RAM; processing this dataset took about 15 minutes and required only 68MB RAM. In contrast, the MCL algorithm was the least efficient (except when the smallest dataset was used) using about 80 times more memory at 40 times slower speed than netclust on the KOG set. In fact, this difference between the two algorithms is expected to increase in non-linear manner with the size of dataset used given the time and space complexities of the algorithms (as indicated by the big O notation). The MCL algorithm does not scale linearly but quadratically in terms of space, so its use is limited by the size of the dataset and available computer's memory; processing a PSN of more than 10^6 nodes and 10^8 edges would require a dedicated computer hardware with more than 32GB RAM.

In contrast to the graph-based methods, the CD-HIT algorithm uses sequences rather than pre-computed all-*versus*-all sequence similarities. In fact, this algorithm combines two processes, namely the sequence similarity search and delineation of sequence similarity groups, that cannot be performed independently; therefore the final benchmark values take into account both processes. Specifically, sequence similarities are calculated by an alignment-free heuristics rather than by more time intensive BLAST comparisons. As a result, this algorithm can be faster than typical graph-based clustering methods based on all-*versus*-all sequence BLAST comparisons, in particular when the goal is to construct a non-redundant sequence database by grouping very similar proteins sequences at a conservative identity threshold. This superior performance, however, becomes less obvious when delineating protein families at a

permissive threshold. In fact, the computational complexity of the CD-HIT algorithm depends on the sequence identity threshold used; for example, clustering the KOG's proteins collection at the lowest threshold (40% identity) took about 1 day to complete, which is more than 800 times slower than at the highest threshold (100% identity). Moreover, for this implementation the comparison of run-times across the datasets suggest that this algorithm does not scale linearly at the permissive threshold: doubling the size of data caused six fold increase in computation time, indicating cubic time complexity [$O(N^3)$]. Based on this premise (as it has not been proved analytically) we estimated by simple extrapolation how much time would it take to delineate protein similarity groups at the lowest threshold: if the largest dataset is 30 times bigger compared to the smallest one and the computation of the latter took about 3.5 hours then in $O(N^3)$ scenario it would take about 54 days (or 30 x 3.5^3 hours) to complete the clustering. This theoretical estimate is partially supported by our empirical observation that the computation was not finished within one month.

Optimizing the methods for functional, structural and phylogenetic inferences

Using different similarity scores, score thresholds, and PSN types (namely ALL-PSN or BH-PSN) provided means to optimize the methods for making reliable inferences of protein functions and (remote) evolutionary relationships between proteins. In case of the CD-HIT algorithm, we could only adjust a single parameter namely sequence identity threshold (between 40–100%) because the underlying heuristics cannot use scoring schemes (e.g., based on BLOSUM or PAM substitution matrices) other than percent identity. Interestingly, for the CD-HIT algorithm there was a single 'best' threshold setting (40% sequence identity) found over all validation instances. In contrast, the other methods had different 'best' settings for different datasets. In particular, different score types showed a trade-off between *VI* and *Rec*; for example, the netclust algorithm used with the neighbor scoring achieved the smallest *VI* distance and sub-optimal Rec value while the algorithm used with the logE scoring achieved the highest *Rec* value and slightly worse *VI* distance when validated against Pfam domain families (Figure 5.2a). In contrast to netclust, the MCL algorithm yielded the most optimal clustering solution with different scores namely percent identity and norm score for this set; the percent identity achieved better *VI* distance while the norm score achieved better *Rec* value (Figure 5.2b). Conversely, the least reliable predictions of the Pfam families were obtained when using netclust with percent identity and MCL with logE scores.

Furthermore, the results suggest that inferences based on an ALL-PSN are better suited for reliable protein and/or domain family detection (such as PIRSF or Pfam families), whereas inferences based on a BH-PSN are better suited for predicting protein functions and/or orthologous groups (as defined by the UniProtKB/Swiss-Prot and KOG databases). For the MCL algorithm, the 'best' scoring scheme to use is identity, norm or neighbor scores with an ALL-PSN, and raw or bit scores with a BH-PSN. In contrast, the 'worst' scheme for this algorithm to use is an ALL-PSN based on logE scores. For the netclust the best score types are bit score, bitcov60,

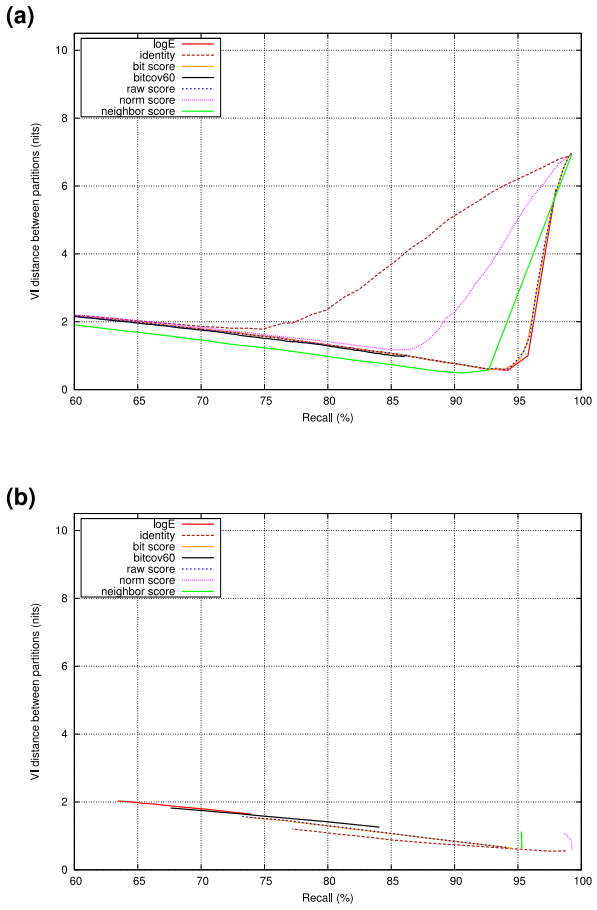


Figure 5.2. Comparison of the netclust and MCL algorithms on the Pfam reference set using seven different sequence similarity scores. The clustering results show that the algorithms differ with respect to 'best' and 'worst' scoring schemes. **(a)** For the netclust algorithm the neighbor score and logE score are the most appropriate scores to use. These, however, show a trade-off between the clustering distance (VI) and recall (Rec): the neighbor score achieved the smallest VI distance at sub-optimal Rec while the logE score achieved the highest Rec at sub-optimal VI distance. **(b)** For the MCL algorithm the percent identity and norm score are the 'best' score types, which show a similar trade-off. However, the percent identity and logE scores are not appropriate for netclust and MCL clustering, respectively.

logE or neighbor score, depending on the purpose. In particularly, the neighbor score is suitable for reliable detection of (remote) homology. In contrast, for this algorithm the least reliable results were obtained by partitioning an ALL-PSN based on percent identity. For further details see the Table S2 and Figures S3.1–3.14 of the Supplementary materials.

Remote homology detection: the globin superfamily use case

Reliable inference of protein homology and subsequent grouping of proteins, particularly those sharing weak sequence similarities (below 30% identity), is an important yet non-trivial task in comparative genome (proteomes) analysis. Here we use the textbook example of the globin superfamily to find out which of the three clustering methods can classify functionally distinct but related families such as hemoglobins, myoglobins, cytoglobins and neuroglobins correctly into a single similarity group (superfamily). For this, the UniProtKB/Swiss-Prot dataset was partitioned by the three distinct methods, and then the grouping of human globins (10 proteins in total) was evaluated further. Moreover, we used all rather than best BLAST hits to enable more sensitive homology detection by dwelling into the "twilight zone" using the concept of transitive homology (Bolten et al., 2001).

Out of the three methods, only netclust grouped all the human globins correctly into a single superfamily, as defined by expert-curated databases such as CATH (1.10.490.10) (Cuff et al., 2009), Pfam (PF00042) or KOG (KOG3378). However, not all similarity scores used with netclust performed equally well; the detection of this superfamily was (i) completely correct when using either the scores, neighbor or bitcov60, (ii) incomplete but correct (e.g., missing one or more proteins) when using the raw, bit, logE or norm scores, and (iii) incorrect or unreliable when using the percent of sequence identity. On the other hand, the MCL algorithm clustered the human globins into two or more groups (e.g., alpha- and beta- hemoglobins were not grouped together) rather than into one large group. As expected, the CD-HIT algorithm used with the lowest threshold failed to recover this superfamily because some human globins such as myoglobin and alpha-hemoglobin subunit share less than 30% of sequence identity.

Hierarchical grouping of protein orthologs

Over 1.1 million protein sequences encoded in 347 fully sequenced prokaryotic genomes were subjected to pairwise proteome BLAST comparisons, resulting in 4.2×10^9 threshold-filtered sequence similarities (defined by bitcov60 scores larger than 50 bits). The resulting ALL-PSN was further processed by the reciprocal best hit (RBH) method to detect putative protein orthologs, yielding a PSN of about 136.8×10^6 orthologous relationships (further denoted as ortho-PSN). Importantly, this RBH-based implementation takes into account one-to-one, one-to-many and many-to-many orthologous relationships between protein homologs, instances of single gene loss (but not reciprocal gene losses), as well asymmetric BLAST scores that might cause "true" orthologs being missed by some graph-based orthology detection methods (Kuzniar et al., 2008).

Next, we partitioned this large ortho-PSN in a hierarchical manner using our scalable netclust software with a series of different score cutoffs. Each of the resulting partitions was validated against manually curated COGs and PIRSF families to find the one that minimizes the *VI* distance and/or maximizes the *Rec* value. Consequently, the 'best' correspondence between the partitions was achieved by setting the cutoff

to 228 bits for COG (VI=1.13; Rec=97.5%; N_{SG}=55,363) and 244 bits for PIRSF (VI=0.6; Rec=97%; N_{SG}=57,429). In particular, these low VI distances suggest that our fully automated procedure can delineate protein families and/or orthologous groups of comparable quality to those of expert classifications. Figure 5.3 shows an example of grouping bacterial DNA-directed RNA polymerase sigma subunits (also called σ factors) by our method as well as manually curated COGs and PIRSF families. Sigma factors are important bacterial proteins that promote sequence-specific binding of RNA polymerase holoenzyme to promoters of various genes (regulons) and thereby initiate their transcription. Specifically, we looked at the classification of primary (house-keeping) and alternative (specific) σ factors that are involved in diverse functions such as sporulation, flagella biosynthesis and heat-shock response (Paget and Helmann, 2003). Clearly, the protein hierarchy constructed by our method provides a "richer" representation of functional and phylogenetic relationships between proteins than that of single-level or 'flat' classifications.

5.4 Discussion

In this study we benchmarked three principally distinct methods used to delineate protein similarity groups from large protein collections such as proteomes. Our aim was to find out which of the three methods provides the 'best' trade-off between scalability and biological soundness, so that proteins of hundreds of proteomes can classified quickly and meaningfully. Many thousands of partitions constructed *de novo* were compared with several reference partitions derived from the protein knowledge bases, and evaluated using a reliable cluster validation method based on information theoretic concepts.

The results of biological benchmarks suggest that the straightforward nearest neighbor method used by netclust or NCBI's BLASTClust can perform almost as good as and in some instances even slightly better than a more sophisticated method such as the MCL algorithm. The latter was the 'best' performer for the PIRSF, KOG and ENZYME benchmark sets. The difference in the quality of protein similarity groups (partitions) delineated by the two methods is, however, negligible; for example, the MCL algorithm seems to perform better in terms of recall (no more than 5% increase) but in the majority of cases slightly worse in VI distance in comparison to the netclust algorithm. Importantly, both algorithms used specifically with the optimized scoring scheme, yielded partitions that highly resembled those of the expert classifications. Both algorithms performed better on PIRSF and Pfam families, and UniProtKB/Swiss-Prot descriptions than on KOG orthologous groups and ENZYME function classification. However, the CD-HIT algorithm performed significantly worse than the graph-based methods; nevertheless, the results were of reasonably good quality for UniProtKB/Swiss-Prot descriptions and PIRSF families. None of the methods tested, however, could predict the first two levels (class and subclass numbers) of the EC hierarchy reliably. This is in agreement with the empirical limits of sequence-based function prediction proposed by Devos and Valencia (2000) who concluded that only two out of four EC digits (namely the sub-subclass and

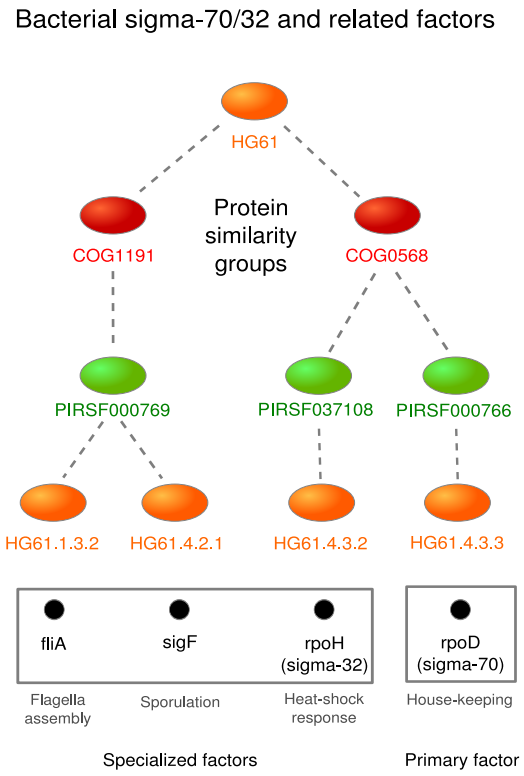


Figure 5.3. Comparing hierarchical *versus* single-level classifications using an example of bacterial DNA-directed RNA polymerase sigma subunits (σ factors). For this σ-70/32 and related factors were compared across COG, PIRSF and our HGPO (Hierarchical Grouping of Protein Orthologs) classifications. In HGPO functional and phylogenetic relationships between proteins are presented at various levels of specificity: one the one hand, the root-level HG61 group subsumes the two distantly related orthologous groups, COG1191 (annotated as 'DNA-directed RNA polymerase specialized sigma subunit') and COG0568 (annotated as 'DNA-directed RNA polymerase, sigma subunit (sigma70/sigma32)'; and the 4^{th}-level contains functionally very specific subgroups on the other hand. An intermediate between the two levels is the PIRSF classification, which classifiesthe proteins into groups of higher functional specificity than those of the COGs yet can be divided further into functionally more coherent subgroups of the HGOP classification. The PIRSF000769 (annotated as 'transcription sigma factor, G type; curation=Full; level=family') corresponds to two 4^{th}-level HGOP groups, whereas the two PIRSF037108 and PIRSF000766 groups (annotated as 'RNA polymerase sigma-32 factor; curation=Full; level=family' and 'transcription initiation factor sigma 70; curation=Preliminary; level=family', respectively) have one-to-one mapping to two HGOP groups. The four seed protein sequences, each member of distinct functional and/or phylogenetic family (share less than 30% sequence identity), were collected from the UniProt database: the *fliA* gene of *Salmonella typhimurium* encodes a sigma factor for flagellar operone (P0A2E8); the *sigF* gene of *Bacillus subtilis* codes for sporulation protein (P07860); the *rpoH* gene of *Escherichia coli* codes for sigma-32 factor or heat shock regulatory protein (P0AGB3); the *rpoD* gene of *E. coli* codes for primary sigma-70 factor (P00579).

serial number) can be predicted reliably for proteins that share at least 15% sequence identity.

As mentioned above, the choice of similarity scores and amount of similarity data (e.g., ALL-PSN *versus* BH-PSN) is crucial for reliable delineation of protein similarity groups used for a particular purpose. Such refinements, however, were not possible for the CD-HIT algorithm because the only adjustable cutoff parameter was the percentage of sequence identity. We found that by setting this threshold to the lowest possible value (40% identity) the algorithm yielded best results for all datasets tested herein. For the other two algorithms, however, the choice of a similarity score and PSN type seems to be algorithm- and task-specific. Nevertheless, the results suggest that a BH-PSN is better suited for predicting protein functions and/or orthologous groups, whereas an ALL-PSN is better suited for delineating protein and/or domain families. This is expected as the former predictions need only the closest members while the latter need also remote relatives. It seems that the amount of data needed to address a particular biological problem optimally is an important parameter of protein clustering. Interestingly, the use of E-values, although reliable for database searches using algorithms such as BLAST or Smith-Waterman (Hulsen et al., 2006b), is not necessarily the best choice for protein clustering; on the contrary, partitioning the logE-based ALL-PSNs by the MCL algorithm yielded partitions of the lowest quality (indicated by the lowest recall and largest *VI* distance). It has been known for long that sequence identity is not a reliable measure for inferring homologous proteins (Dayhoff et al., 1978). Interestingly, this does not necessarily hold when clustering proteins with the graph-based methods used herein; for instance, clustering an identity-based ALL-PSN by the MCL algorithm resulted in the highest agreement with the Pfam classification. For the MCL algorithm, the 'best' scoring scheme to use is identity, norm or neighbor scores with an ALL-PSN, and raw or bit scores with a BH-PSN. For the netclust algorithm the most appropriate scores to use are bit score, bitcov60, log_{10}-transformed E-value and neighbor score; the latter is particularly suitable for detecting reliable remote homology but its computation is more expensive compared to the genuine BLAST scores. Interestingly, the amount of similarity data, namely ALL-PSN *versus* BH-PSN, seems to have a more significant bearing on netclust than on MCL clustering. The reason for this may partially be due to the underlying clustering paradigms of these algorithms.

Further, we found that none of the methods except netclust grouped all human globins into a single globin superfamily correctly. This supports the view that linkage-based graph algorithms such as netclust can reliably delineate remote homologs by the legitimate use of transitive homology (Bolten et al., 2001). Similarly, the homology concept can be used to detect more distant family members through iterative database searches, an approach also known as "sequence-space-hopping" (Rost, 1999). However, the outcome of this approach might depend on the order in which "seed" sequence(s) are used i.e., input-order dependency, as well as on the number of iterations involved in the search; both input parameters need to be defined by a user. In contrast, the graph-based clustering algorithms such as netclust and MCL do not depend on these parameters, and as such can exploit the transitivity of homology either by linkage (such as netclust) or random-walk (such as MCL) through an entire

network in a fully automated manner.

The empirical run-times and memory usages of the algorithms (Table 5.4) support the theoretical complexities (as indicated by the big O notation), and most importantly, provide a practical guide for choosing one algorithm over another depending on the size of the data at hand. For large-scale applications, it is desirable that an algorithm scales linearly both in time and space with respect to the increasing size of data. However, this is often difficult to achieve, so that one always deals with a trade-off between the two characteristics. The netclust program implements a fast and memory-efficient algorithm with (nearly) linear time and space complexity, making it suitable for large-scale applications that, in case of other methods with non-linear characteristics, would require expensive computer hardware with significantly larger RAM memory and/or more computational time. Out of the methods tested, netclust required the least amount of computer resources, and was the only method that could easily handle the dataset of 347 prokaryotic proteomes on a standard computer with 1GB RAM. In fact, 99% of the processing time is spent on indexing the input PSN so that the actual search for groups can be done efficiently, in particular when a series of similarity (or distance) cutoffs is applied on the same (indexed) PSN. The indexing time could be reduced using faster harddisks with parallel access to data (data not shown), as the speed of the algorithm is bounded to speed of the disk. Alternatively, one could use solid-state disks to improve the clustering speed even more. In contrast to netclust, typical graph-based clustering algorithms such as MCL need to store an entire similarity matrix or PSN in computer's memory prior to delineating groups; this algorithm would require more than 32GB RAM to process a PSN of 10^6 nodes and 10^8 edges. Moreover, the algorithm does not scale in time as its complexity is cubic, requiring much more time to process the same amount of data compared to netclust.

The CD-HIT algorithm was specifically developed to cluster very similar (redundant) sequences in large sequence databases rapidly. For this, the algorithm is perfectly suited owing to its linear complexity particularly when conservative identity thresholds are used. As this method uses a heuristic (alignment-free) search rather than more time intensive all-*versus*-all sequence BLAST comparisons, and hence can be faster than typical graph-based methods such as netclust or MCL, provided that these comparisons have not be performed *a priori*. However, the CD-HIT's superior run-time performance is not so obvious when processing large datasets at permissive identity thresholds, commonly used to delineate protein families. We found that the algorithm was by orders of magnitude slower at permissive thresholds (e.g., 40–60% identity) than at conservative thresholds (e.g., 70–100% identity). Specifically, by comparing the running times across different datasets we concluded that the CD-HIT algorithm has a non-linear time complexity, and is likely to be cubic [$O(N^3)$].

As the size of the UniProt database doubles every 18 month (the current release 15.5 contains nearly 10 million entries), comparing sequences in the all-*versus*-all sequence manner is the major computational bottleneck for any graph-based clustering method; the overall complexity of the comparison is $O(N^2)$ regardless of the similarity search algorithm used. This problem can be alleviated by distributing the sequence comparisons over many computers (or CPUs) for parallel processing.

Although the problem of automate protein clustering consists of two tightly linked sub-problems, namely sequence similarity search and delineation of protein similarity groups, our main focus here has been the latter. Computing all-*versus*-all BLAST similarities and then using the graph-based netclust, theoretically speaking, can have better overall time complexity than the sequence-based CD-HIT method because the former is $O(N^2)$ while the latter is $O(N^3)$ particularly when distant rather than nearly identical sequences are being grouped. Both netclust and CD-HIT algorithms have same (linear) space complexity but they differ in the actual memory usage; the latter consumes significantly higher amount of RAM.

Using precomputed all-*versus*-all sequence similarity data available in databases such as SIMAP (Rattei et al., 2006) or CluSTr (Petryszak et al., 2005) can reduce the analysis time of large protein collections from months to minutes particularly when the netclust program is used. In contrast to graph-based methods, the CD-HIT algorithm cannot use precomputed sequence similarities as input and hence one cannot reduce the computational time needed to obtain protein similarity groups. Even if it would be possible the graph-based methods tested herein yield significantly better results.

Besides the advantages, the graph-based method as used by netclust has several limitations that need further attention. First, the underlying algorithm can lack robustness when there is little spatial separation between groups (Handl et al., 2005). A 'network-rewiring' approach seems to be a promising way to improve protein classification particularly in the presence of multi-domain proteins (Joseph and Durand, 2009). This approach increases the scores of related protein pairs while decreases the scores of unrelated pairs. Second, the netclust algorithm can be used to construct a large protein hierarchy of many levels in which any parent node (group) can have one or more children nodes (sub-groups), resulting in a n-ary rather than binary tree representation. As in any other hierarchical clustering, the number of levels to use is not known *a priori*. Recently, an automated method, which is independent of scoring scheme and clustering algorithm used, has been proposed to address this problem (Donald and Shakhnovich, 2005). And last but not least, a deep protein hierarchy may contain very similar (redundant) protein similarity groups at the nearest levels. This redundancy can be reduced by tree-pruning (or compression) techniques that may also remove some biologically relevant information from the tree (Kaplan et al., 2004; Petryszak et al., 2005).

5.5 Conclusions

We present a systematic evaluation of distinct protein clustering algorithms that addresses both computational complexity and biological validity of the algorithms. In this study, the algorithms were compared using a reliable cluster validation method that takes into account entire protein classifications (such as those curated by experts) and in addition, it involves different parameter settings, scoring schemes and input datasets.

Our results are in agreement with previous (small-scale) studies suggesting that a

simple nearest neighbor clustering method as implemented in the netclust or NCBI's BLASTClust programs can perform just as good or even better in the cases of protein function prediction and remote homology detection than the computationally more intensive TribeMCL algorithm (Kelil et al., 2007; Krause et al., 2005; Paccanaro et al., 2006; Yang et al., 2009). Yet the MCL algorithm is still the 'best' quality method for the majority of benchmark sets used herein. In contrast, the sequence-based CD-HIT algorithm performed significantly worse in terms of quality than the graph-based methods. It should be noted, however, that the CD-HIT algorithm is still the best choice for clustering nearly-identical sequences from large protein collections in the shortest time.

To conclude, the netclust algorithm is the most scalable and least compromising method in terms of quality that is suitable for large-scale analyzes of hundreds of proteomes on a standard computer. When used with a reliable scoring scheme this straightforward method can delineate protein similarity groups of quality similar to that of expert classifications. In the coming years, we expect much development of new protein clustering methods. Therefore, it is important to compare these methods systematically and demonstrate their merits and shortcomings in relation to simple methods such as nearest neighbor clustering.

Supplementary materials

S1. Clustering programs and parameters used

program: netclust
parameter: weight (similarity) cutoff
threshold values chosen according to the BLAST score type:
(i) *percent of sequence identity* from 0 to 100 in step of 1,
(ii) *raw score* from 0 to 1000 in step of 5,
(iii) *bit score* from 0 to 1000 in step of 2,
(iv) *logE score* from 0 to 200 in step of 2,
(v) *bitcov60 score* from 0 to 1000 in step of 2,
(vi) *norm score* from 0 to 100 in step of 1,
(vii) *neighbor score* from 0 to 100 in step of 1.

program: MCL
parameter: inflation
threshold values: from 1.5–6 in step of 0.5

program: CD-HIT
parameter: percent of sequence identity
threshold values: from 40–100% in step of 5

Reference partition	netclust	MCL	CD-HIT
SPROT	BH-PSN bit(26): 97.74, 0.65 identity(22): 97.77, 0.85	BH-PSN identity+inflation(1.5): 97.77, 0.87 raw/bit+inflation(1.5): 97.74, 0.67	identity(40): 87.36, 0.79
Pfam	ALL-PSN neighbor(2): 90.61, 0.49 logE(4): 94.21, 0.57	ALL-PSN norm+inflation(1.5): 99.25, 0.6 identity+inflation(2): 97.52, 0.55	identity(40): 72.99, 2.02
PIRSF-1	ALL-PSN bitcov60(36): 95.53, 0.32 bitcov60(26): 96.03, 0.35	ALL-PSN bitcov60+inflation(1.5): 95.01, 0.33 norm+inflation(2-3): 99.64, 0.37	identity(40): 81.00, 0.98
PIRSF-2	RBH-PSN bitcov60(244): 97.00, 0.6	*ND*	*ND*
KOG	BH-PSN logE(0): 99.96, 1.46	ALL-PSN norm+inflation(1.5): 99.97, 1.31 neighbor+inflation(2): 99.89, 1.11	identity (40): 57.25, 2.51
COG	RBH-PSN bitcov60(228): 97.5, 1.13	*ND*	*ND*
ENZ-1	ALL-PSN raw(50)/bit(24)/identity(18): 97.27, 3.36 identity(22): 97.23, 3.35	ALL-PSN identity+inflation(1.5): 96.24, 3.47 norm+inflation(2): 97.26, 3.61	identity(40): 74.41, 5.13
ENZ-2	ALL-PSN neighbor(3): 90.16, 2.58	ALL-PSN identity+inflation(1.5): 96.24, 2.52 norm+inflation(2): 97.26, 2.75	identity(40): 74.41, 3.98
ENZ-3	ALL-PSN neighbor(4): 88.85, 1.26	ALL-PSN bit+inflation(1.5): 93.61, 1.16 norm+inflation(2-2.5): 97.26, 1.25	identity(40): 74.41, 1.8

Table S2. Benchmark results of different clustering algorithms. For each reference partition the 'best' scoring scheme namely the graph type (ALL-, RBH- or BH-PSN), BLAST-based score (identity, raw, bit, E-value, norm or neighbor scores) and cutoff value, as well as the 'best' results [see recall (Rec) and variation of information (VI) after a colon] obtained using that scoring scheme are shown. Note: *ND* - the values were not determined due to computational limitations.

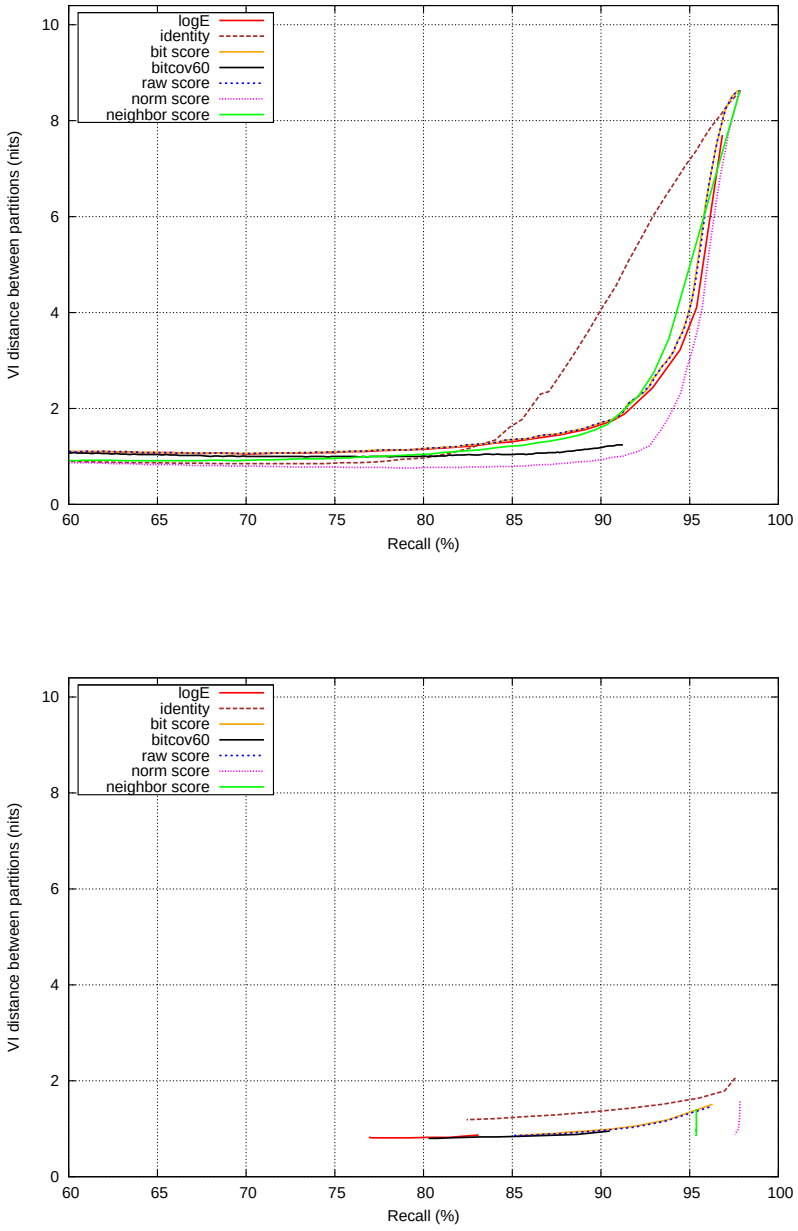


Figure S3.1. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (bottom) are compared the SPROT reference partition.

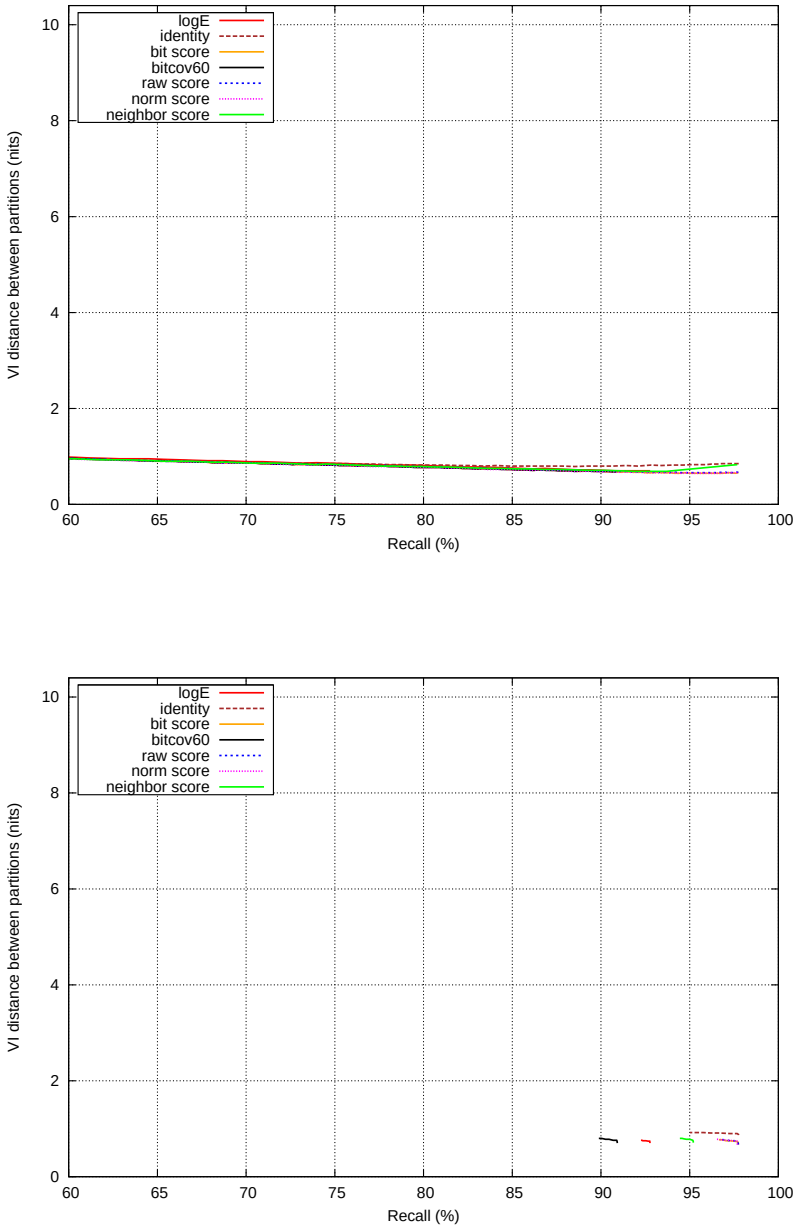


Figure S3.2. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the SPROT reference partition.

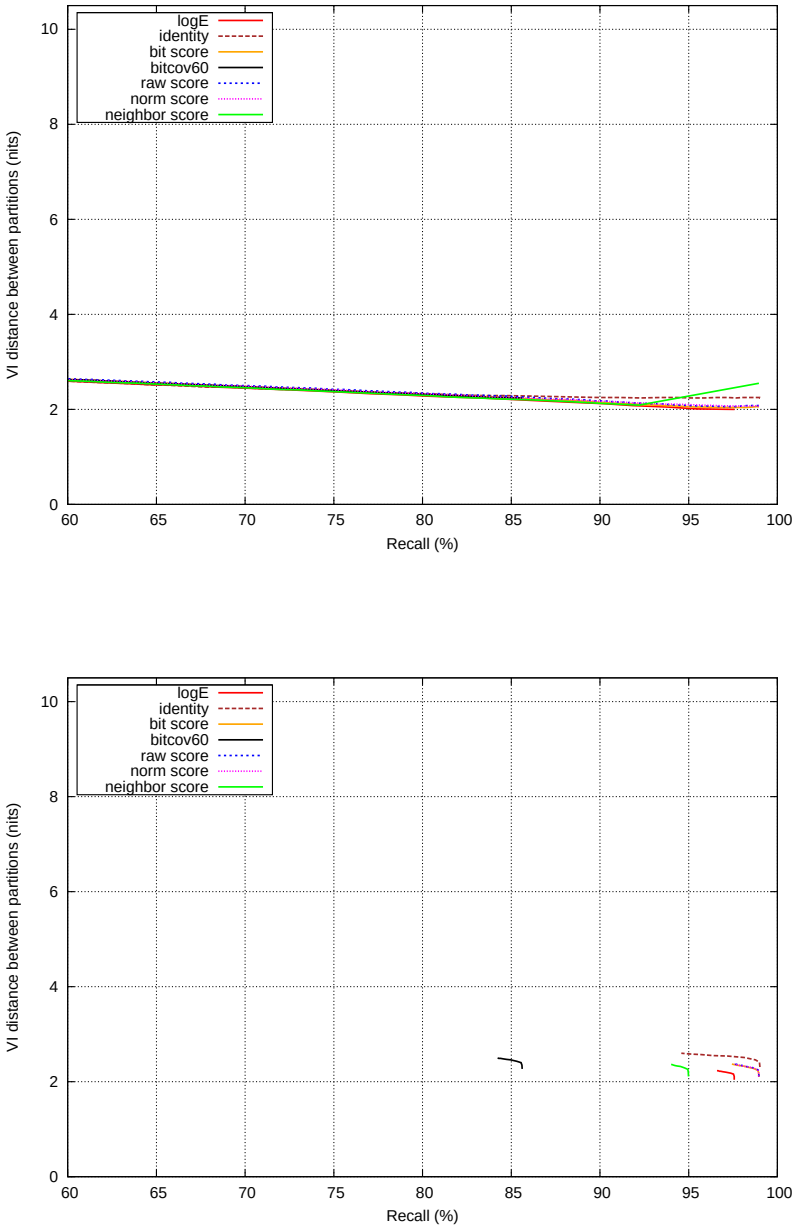


Figure S3.3. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the Pfam reference partition.

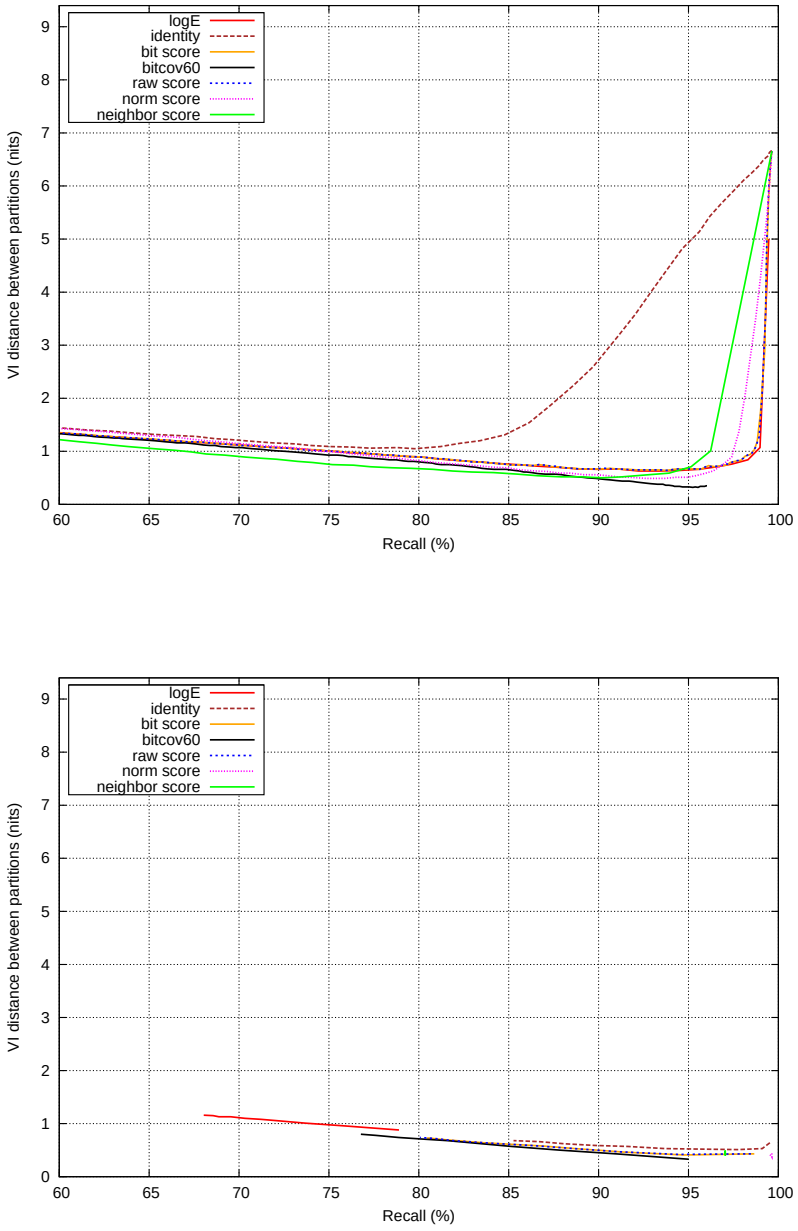


Figure S3.4. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the PIRSF-1 reference partition.

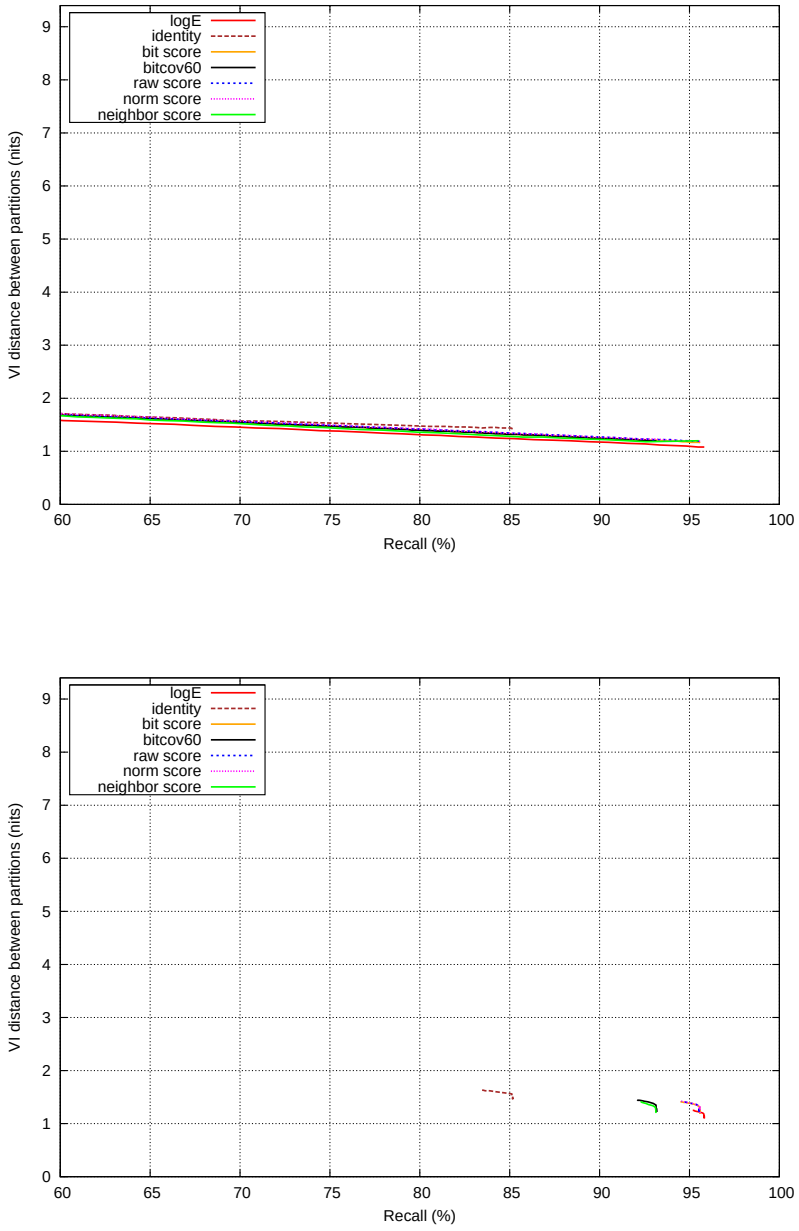


Figure S3.5. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the PIRSF-1 reference partition.

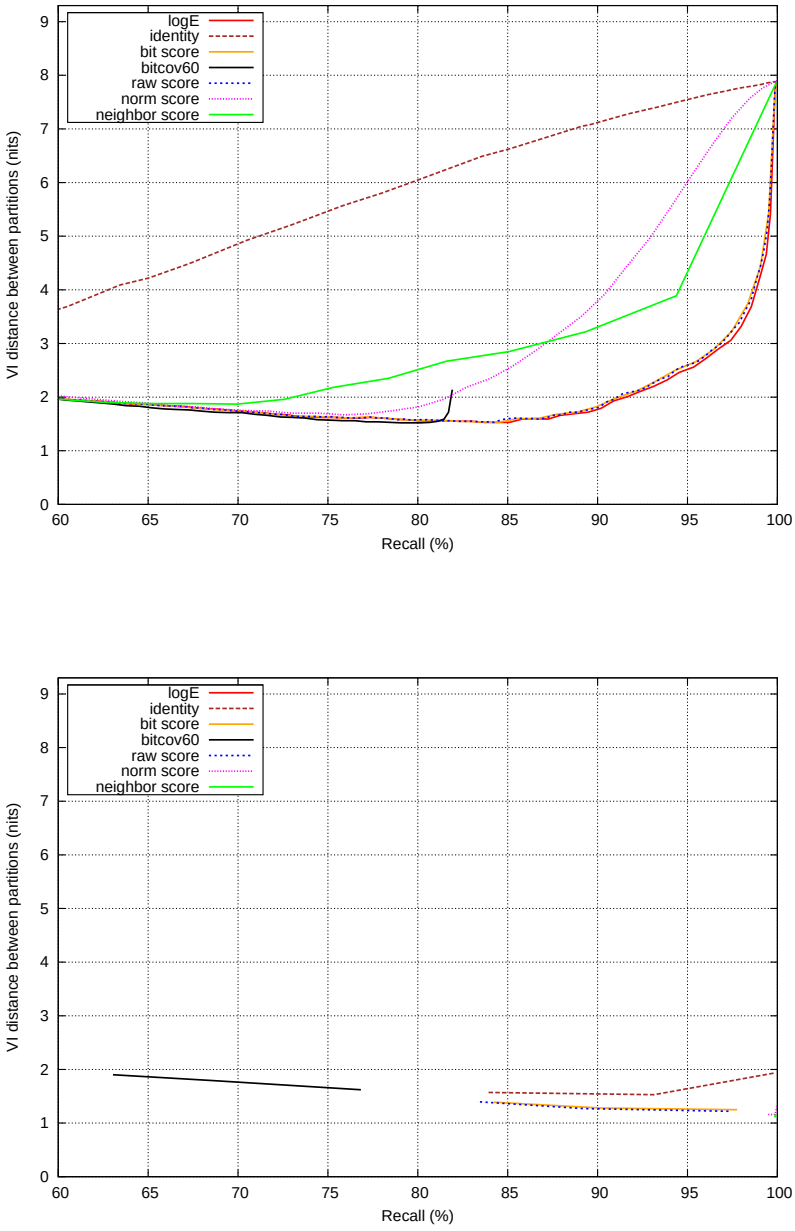


Figure S3.6. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the KOG reference partition.

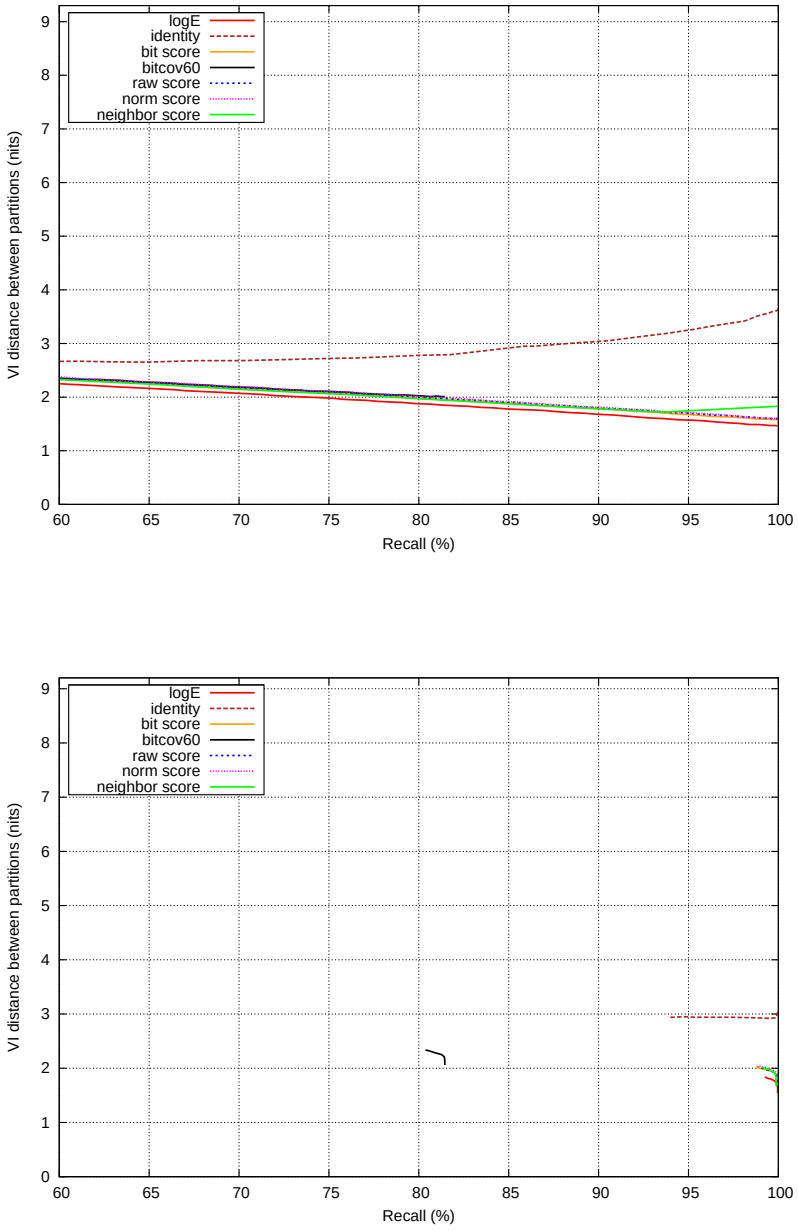


Figure S3.7. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the KOG reference partition.

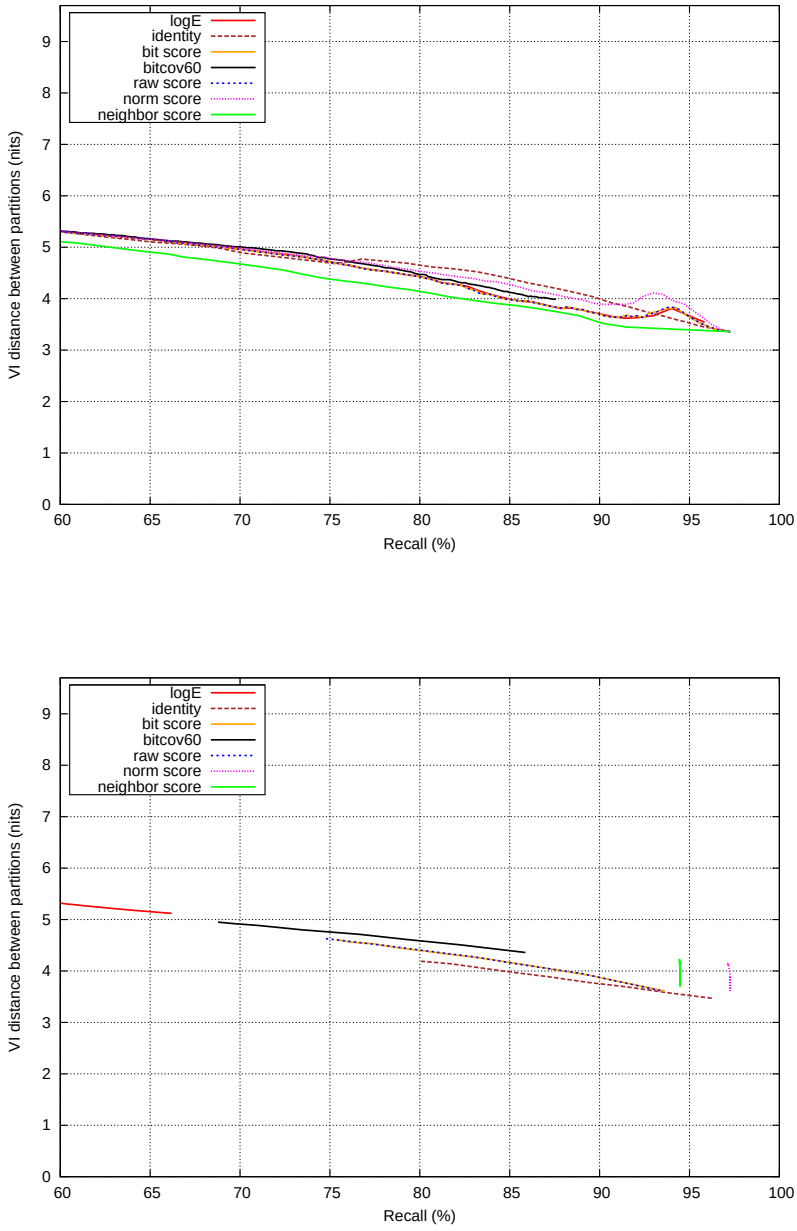


Figure S3.8. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (botttom) are compared to the ENZ-1 reference partition.

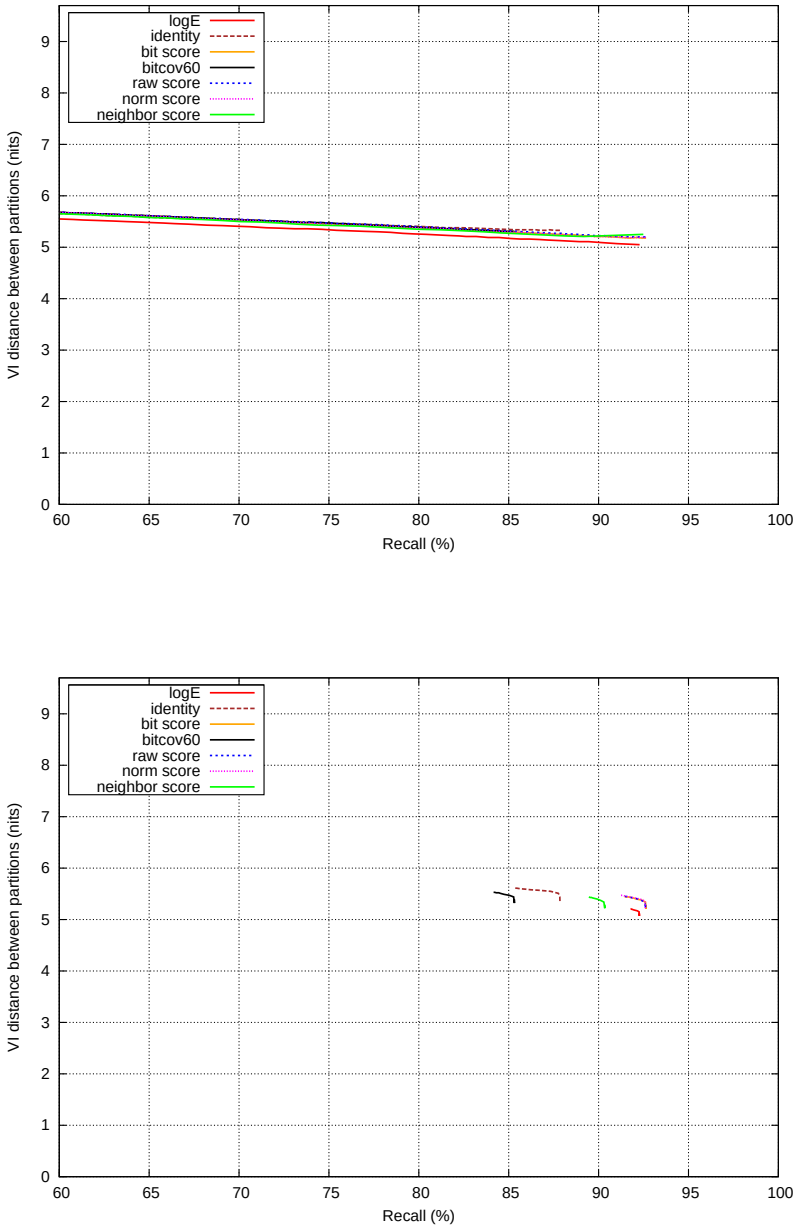


Figure S3.9. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the ENZ-1 reference partition.

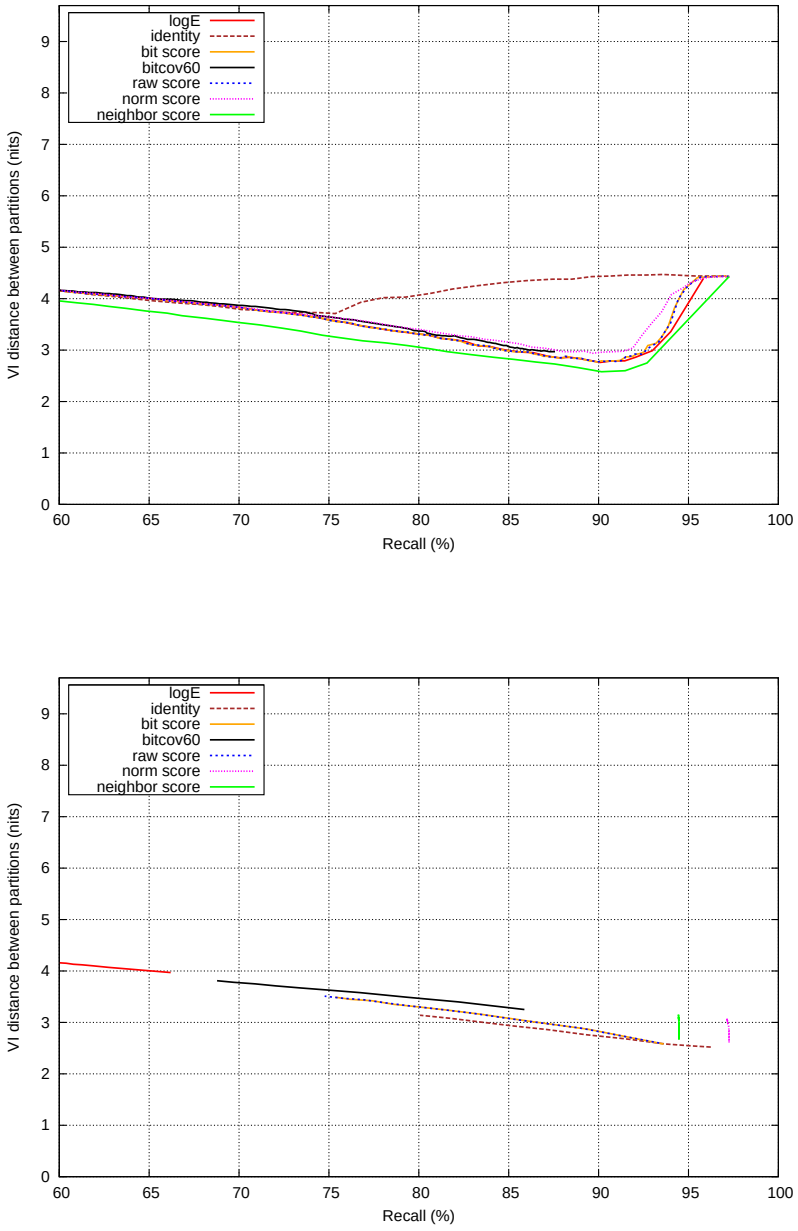


Figure S3.10. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the ENZ-2 reference partition.

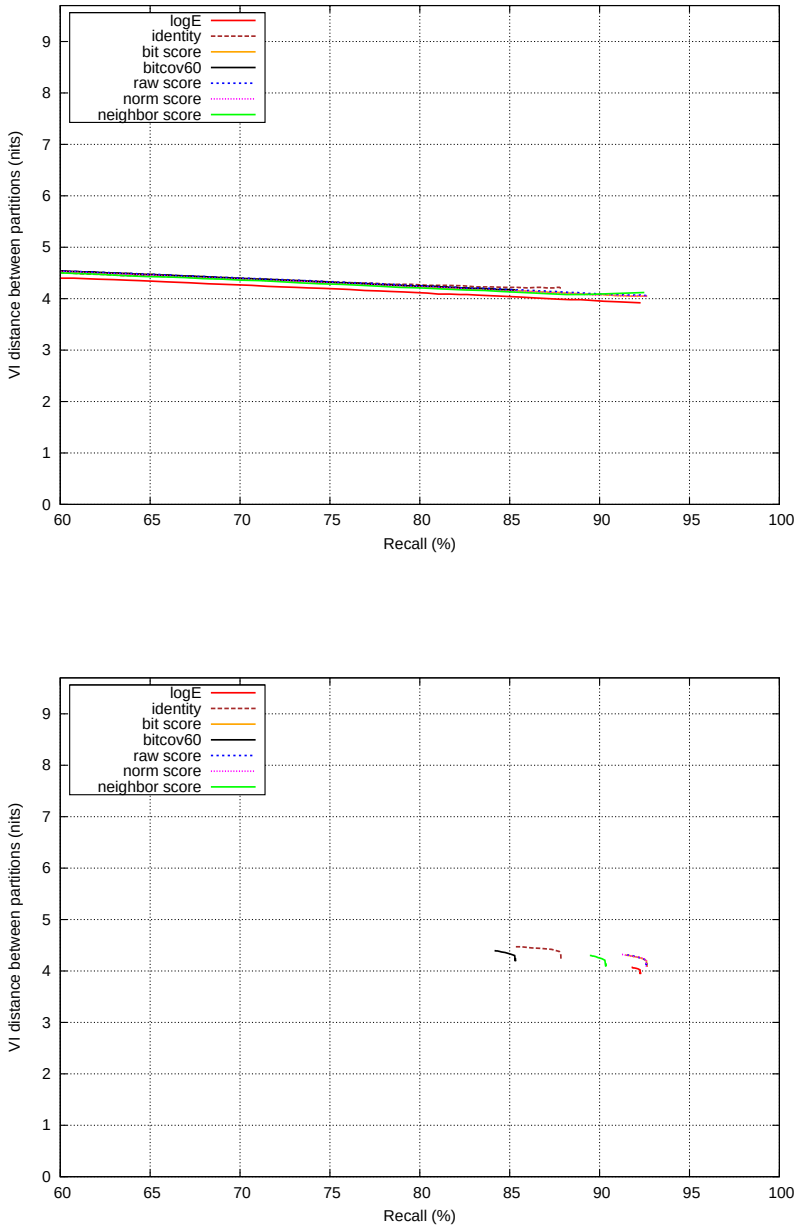


Figure S3.11. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the ENZ-2 reference partition.

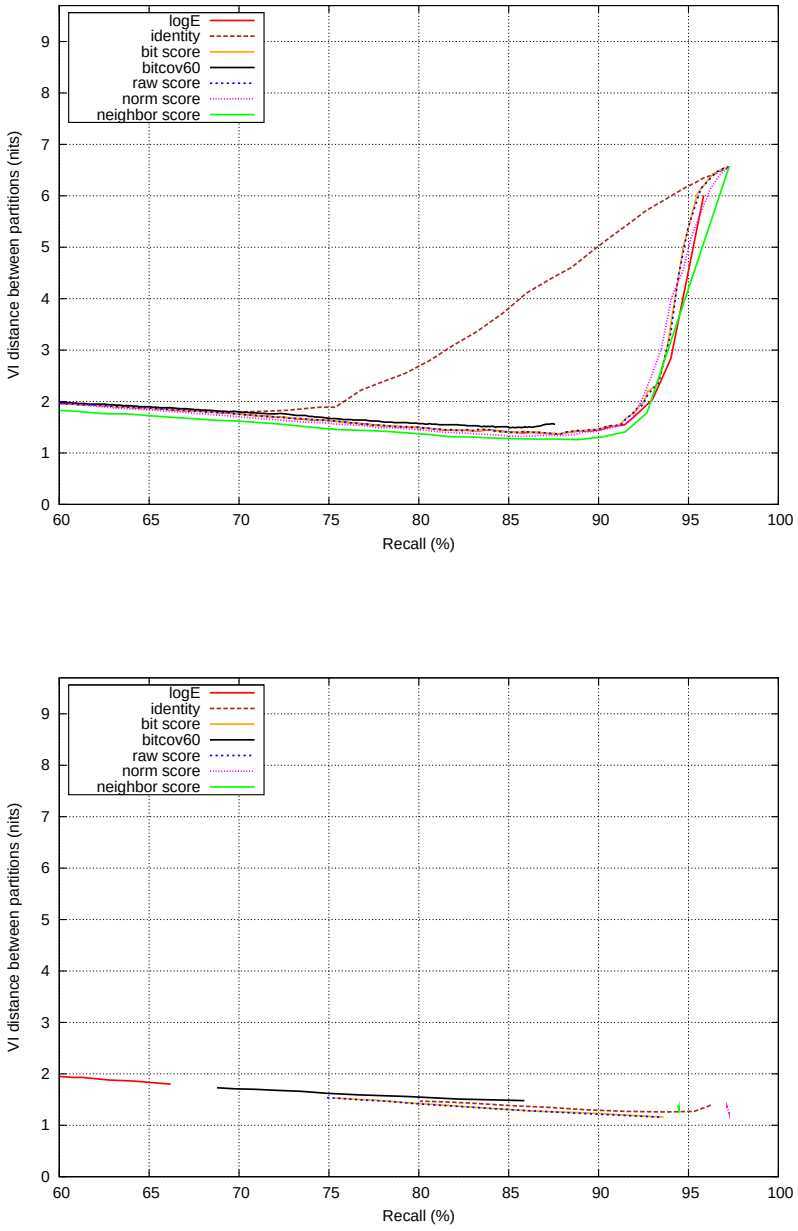


Figure S3.12. Recall *versus VI* distance plots. ALL-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the ENZ-3 reference partition.

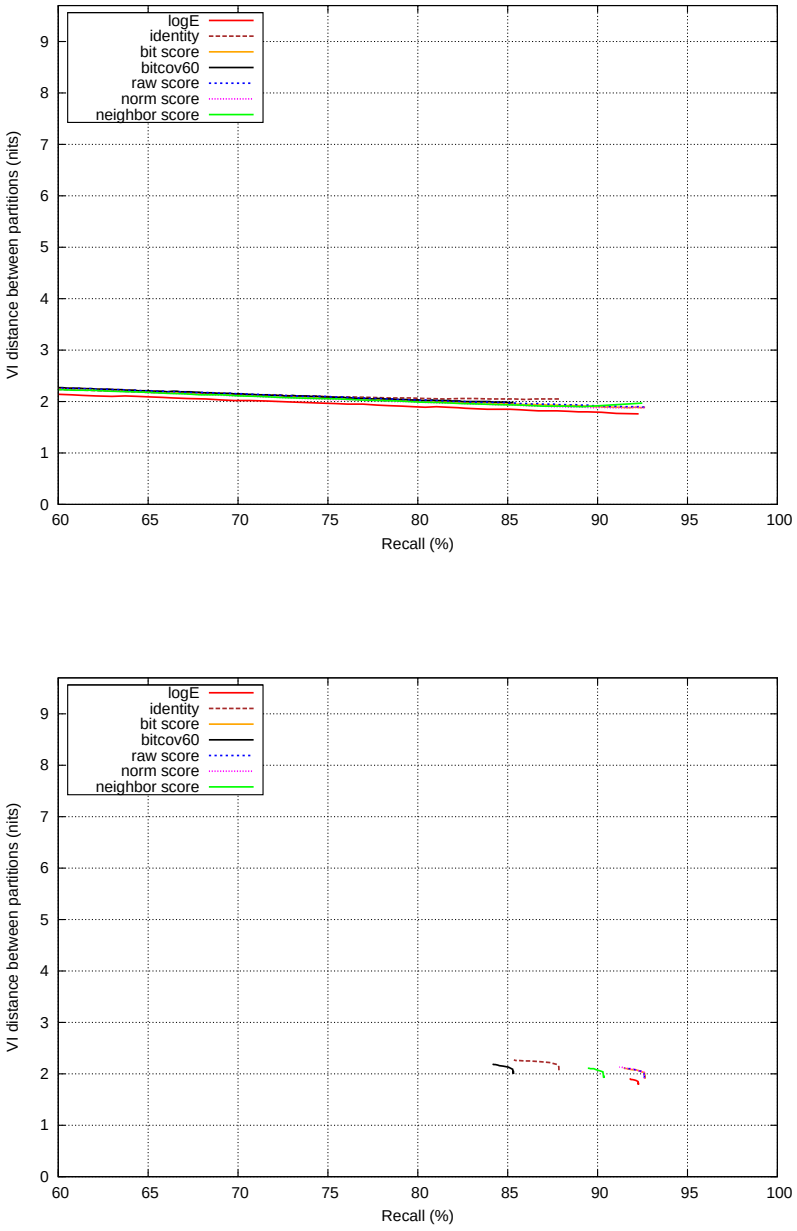


Figure S3.13. Recall *versus VI* distance plots. BH-PSN-based partitions of netclust (top) and MCL (bottom) are compared to the ENZ-3 reference partition.

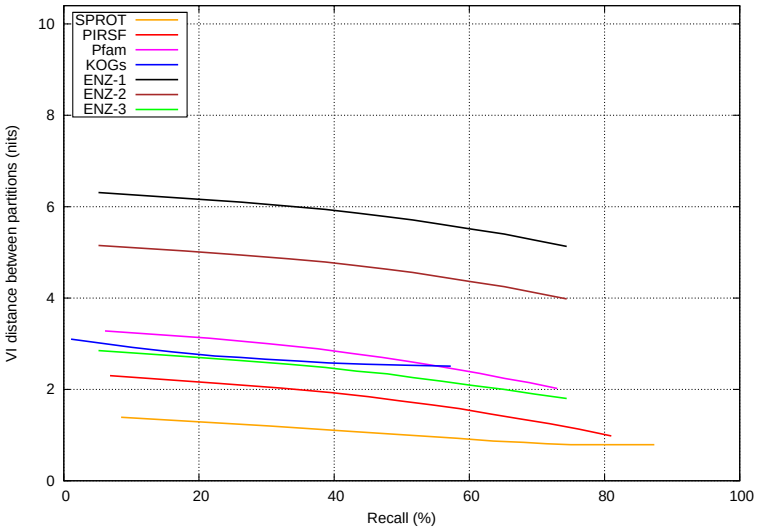


Figure S3.14. Recall *versus VI* distance plot of CD-HIT partitions when compared to all reference partitions.

CHAPTER 6

DISCUSSION

6.1 Scope and aims

The subject of this thesis is automated protein classification. We focused on computational methods that can delineate phylogenetically and functionally related proteins into similarity groups from large sequence collections, such as proteomes of diverse species. Specifically, these methods combine unsupervised learning (clustering) with the knowledge of molecular phylogenetics, particularly that of orthology. An emerging problem arises with the increasing sizes of sequence databases (for example, the UniProt database has nearly 10 million protein entries): most available bioinformatics software used for protein clustering, although maybe very accurate, cannot handle large data sets. Hence the trade-off between scalability and reliability of these methods has become an important issue. Our aim was to develop an efficient unsupervised method suitable for detection of protein families and orthologous relationships from hundreds of available proteomes. We also aimed at implementing an integrated database where the predictions of different methods can be stored and readily compared. In this chapter we first discuss the results in the view of current literature, then propose directions for future research, and finally summarize the main contributions of this thesis.

6.2 The quest for orthologs

Evolutionary (phylogenetic) concepts have been instrumental in many areas of genomic research. In particular orthology provides a framework for subjects as diverse as the evolution of genomes, gene functions, cellular networks and functional genome annotation. Genome-wide detection of orthologous genes across different species has enabled biological functions and cellular processes to be inferred reliably from one species to another. For example, this knowledge is used to search for cures of human genetic diseases using model organisms (e.g., mouse or fruit fly).

Orthology combines the knowledge of molecular and evolutionary biology, namely that of genes and species, in a unique web of biological knowledge that can now be explored using bioinformatics methods. This integration requires a common language that can be used by both biologists and computer scientists. Graph theory provides such a language because it can be used to model various biological objects (e.g., proteins or families thereof) and their relationships (e.g., orthology) as the nodes and edges of a tree or a network. In principle, orthology (and other phylogenetic relationships) can be modeled using both a phylogenetic tree and a network. Many bioinformatics tools, such as programs and databases to infer orthologous relationships between genes of fully sequenced genomes have been developed over the past

years. In chapter 2 we have compared this plethora of prediction methods in order to identify the main caveats of large-scale orthology inference in general as well as the merits and/or shortcomings of each method in particular (Kuzniar et al., 2008). Although several new orthology detection methods have emerged since then (Datta et al., 2009; Huerta-Cepas et al., 2008; Kim et al., 2008; Vilella et al., 2009), there are only a few that combine orthology predictions obtained by different methods (Eyre et al., 2007; Penkett et al., 2006). Most of these integrated databases, however, have limited functionality and comparative data on few eukaryotic genomes.

6.3 ProGMap's merits and shortcomings

To overcome the limitations of these databases, we developed an integrated annotation resource for protein orthology called *ProGMap* (Protein Group Mappings) (Kuzniar et al., 2009). This resource is designed to help biologists and database annotators who deal with partially annotated genomic data by searching for complete information about proteins across different databases, as well as who need to test the coherence of different group assignment methods from a single WWW portal. Moreover, ProGMap gives a plausible indication on how conflicting predictions could be improved. All the information present in the underlying databases are preserved and maintained in a largely automated manner. Specifically, the different classifications are mapped onto each other by constructing a unique network of links between (groups of) proteins using a fast and fully automated sequence-based mapping method. The ProGMap's network-based architecture enables queries to be made using synonymous sequence identifiers, gene symbols, protein functions, group identifiers and annotations, and amino acid/nucleotide sequences. For the latter one can use the ProGMap's services namely BLAST similarity and QuickMatch identity searches for finding sequences similar (or identical) to a query sequence. The user-friendly web interface makes the process of comparing different classifications accessible via graphical tools.

Although the ProGMap resource is unique of its kind, its functionality can still be improved. This resource is based on a centralized data warehouse rather than distributed approach (e.g., using web services) so most of the data need to be stored and maintained locally (redundantly) prior to processing and querying. To keep the amount of mirrored data to a minimum, ProGMap stores mainly pointers rather than entire database entries, which can be viewed directly at the original sites and the site provided by our local Sequence Retrieval System (SRS) server (Etzold and Argos, 1993). Moreover, a new data parser must be written each time a new member database is added to ProGMap. For this, we may use the libraries (modules) of the SRS to maintain both the data and parsers using a single interface. The reason for using the centralized approach was to ensure that the large amount of data can be queried as fast as possible. Furthermore, web interfaces based on HTML forms are not suitable for constructing automatic workflows (pipelines). For this XML-based web services are better suited to guarantee interoperability between processes (Neerincx and Leunissen, 2005). Our future work therefore includes the development of such a web services-based interface that can link ProGMap to high throughput pipelines.

Furthermore, ProGMap does not allow users to display phylogenetic trees and domain predictions for a protein group of interest (although this information is available at the WWW sites of some constituent databases). Therefore, the ProGMap's graphic functionality can be extended with phylogenetic tools such as TreedomViewer (Alako et al., 2006) that collectively provide biologically relevant information about the function and evolution of protein domains, as well as allow the putative family or orthology assignments to be validated. ProGMap is a manual tool that requires human expertise to resolve conflicting assignments or annotations; one could therefore develop an automated method that might improve the classification or annotation quality by exploring the ProGMap's protein network. Last but not least, the use of protein (gene) ontologies such as the PRO (Natale et al., 2007) within ProGMap may provide a framework to facilitate cross-species comparisons, pathway analyses or genotype-phenotype studies.

6.4 Netclust's merits and shortcomings

In chapter 4 we investigated the computational complexity of available bioinformatics software used to delineate protein families or orthologous groups from pairwise sequence similarities using the straightforward nearest neighbor (or single link) criterion. The premise here was that this simple unsupervised method will scale better than more sophisticated algorithms.

We implemented a fast and memory-efficient graph software called *netclust* that is suitable for delineating meaningful protein groups from large sequence similarity networks using the nearest neighbor criterion. The underlying algorithm, which belongs to the class of UNION-FIND algorithms (UFA), has been known in computer science for many years; historically, it was first used in FORTRAN compilers and later also in many graph and tree processing applications (Lao, 1981). However, UFAs have rarely been used in bioinformatics. We adapted an asymptotically optimal (scalable) variant of the UFA to our needs, namely large-scale protein classification, and implemented this algorithm in netclust. Our benchmark analysis showed that most software used for protein clustering can only scale to data sets of moderate sizes because it needs to store the entire similarity matrix in the computer's memory prior to processing (called a memory-based or 'in core' model). In contrast, there are only two freely available programs, namely netclust and NCBI's BLASTClust, that can scale to much larger data sets; by keeping most of the input data on the hard drive, both programs can save a lot of RAM memory space. Although BLASTClust is an efficient and reliable sequence clustering program, its use is limited to BLAST-based similarity networks, whereas netclust is generally applicable. Moreover, netclust is faster (up to 100 times) and it requires significantly less memory (about 50%). To conclude, netclust can handle very large networks, such as those constructed of hundreds of proteomes, on inexpensive computer hardware.

In chapter 5, three distinct protein clustering algorithms namely netclust, MCL and CD-HIT were evaluated systematically on available protein knowledge bases in order to find out which one provides the 'best' trade-off between scalability and bio-

logical soundness. It is generally asserted that the nearest neighbor criterion, as used by netclust, is not suitable for grouping functionally and/or phylogenetically related protein sequences owing to the poor quality of the resulting groups, and therefore more sophisticated clustering algorithms should be used instead. Our benchmark results, however, refute this hypothesis. The results support the view that a simple, computationally cheap method can perform similar to and in cases even better than sophisticated, yet much more costly methods; this implies that the former should be preferred over the latter when clustering large protein collections.

We used netclust to construct a large protein hierarchy (about 1 million proteins of 347 prokaryotic proteomes) that not only captures the information contained in expert protein classifications such as COG (Tatusov et al., 2003) and PIRSF (Wu et al., 2004), but also provides an enriched framework in which the functional and (remote) evolutionary relationships between proteins can be studied at various levels of specificity (chapter 5). The importance of using a hierarchical rather than single-level 'flat' scheme for grouping proteins of entire proteomes has also been recognized by others (Jensen et al., 2008; Klimke et al., 2009; Kriventseva et al., 2008). Although the idea of hierarchical protein classification is not novel, the concepts such as 'family' and 'superfamily', which were first introduced by Dayhoff (1976), are still valid for the ever-expanding protein networks. We showed that netclust can also be used for reliable detection of remote homologs (chapter 4.2 and 5).

The netclust program, however, is not perfect; there are several issues that need further attention. First, the underlying method is prone to the effect of 'chaining' that may cause, for instance, grouping non-homologous proteins due to a common (promiscuous) domain or partial homology. As any other clustering method of this kind, it is not robust particularly in cases where the spatial separation between groups is poor (Handl et al., 2005). To achieve better separation, one could use a scoring scheme that increases the scores of homologous pairs and decreases the scores of non-homologous pairs. For example, clustering based on asymmetric rather than symmetric scores may reduce the illegitimate use of transitive homology particularly in the presence of multi-domain proteins (Bolten et al., 2001). Second, the use of post-processing procedures such as 'network-rewiring' (Joseph and Durand, 2009), graph pruning (Kawaji et al., 2004; Zaslavsky and Singh, 2006), (pseudo)clique detection (Bron and Kerbosch, 1973), hierarchical partitioning (Jothi et al., 2006) or kernel-fusion methods (chapter 4.2) seem promising to achieve better classification. Third, by applying a series of similarity thresholds on the same (indexed) protein network, the netclust program can partition the network into a multi-level protein hierarchy efficiently. However, the cutoff values and number of levels to use are not known beforehand; this is in fact a problem for any hierarchical clustering algorithm. To address this issue, we used an external criterion to measure the distance between target and reference partitions using an entropy-based metric (chapter 5). Accordingly, the 'best' clustering solution is one that minimizes this criterion. However, the optimal similarity threshold may vary from one data set to another and therefore this threshold needs to be determined for each data set individually. For this, Donald and Shakhnovich (2005) have proposed a plausible method that can select such a threshold regardless the scoring scheme and clustering algorithm used. Finally, a protein

hierarchy of many levels may contain a lot of redundancy especially when groups at the nearest levels are very similar (or identical) to one another. Tree-pruning (or compression) techniques seem suitable for this task, although they may produce a 'pruned' tree that contains incomplete biological information (Kaplan et al., 2004; Petryszak et al., 2005). In principle, the netclust's performance could be improved further, for instance, by using dedicated computer hardware with larger RAM memory and faster hard disks with parallel access to data, thereby enabling the program to be used interactively. For such 'real-time' applications it may be necessary to identify potential input/output bottlenecks (namely between main memory and external-memory) that deteriorate the performance of such external-memory graph algorithms (Chiang et al., 1995). The issues above may serve as a guide in designing new algorithms for protein classification in general and improving the netclust program in particular.

Highlights

- A reliable orthology prediction method should use preferably both a graph (network) and a tree in computations rather than either one exclusively, and incorporate the available knowledge of species and gene evolution (chapter 2).
- Orthologous groups must be hierarchical and defined with respect to the last common ancestor of the investigated genes to guarantee the non-transitivity of orthologous (or paralogous) relationships between genes (chapter 2).
- Genome-wide orthology prediction based on reciprocal best hits (e.g., using BLAST) is usually a reliable method that can infer also co-orthologs (in-paralogs) in addition to single-copy orthologs depending on how it is implemented in the computation. Moreover, this method can distinguish between orthologs and out-paralogs when a single gene loss occurred in one of the lineages. However, it cannot distinguish between the two homologs when true orthologs are physically absent from the genome, for example, due to reciprocal gene loss (chapter 2).
- Converting protein (or nucleotide) sequences into fingerprints using the MD4 hashing algorithm is a fast heuristics to compare large collections of (nearly) identical sequences and hence can be used to interlink large databases efficiently (chapter 3).
- ProGMap is a unique integrated database of protein orthology that can be used to assess the coherence of group assignment (classification) methods from a single interface (chapter 3).
- The netclust program is an efficient graph-based bioinformatics tool that can be used to delineate meaningful similarity groups from a large protein network (e.g., more than 10^6 nodes and 10^8 edges) on a standard computer. Its

extended version, Multi-netclust, is suitable for finding connected clusters in multi-parametric networks of heterogeneous data sets (chapter 4).

- A simple and computationally cheap method such as netclust can delineate protein similarity groups of comparable and in some cases of better quality than those of sophisticated, yet much more costly methods such as MCL; therefore the former can process much larger data sets than the latter using the same computer hardware (chapter 5).
- The choice of sequence similarity scores, amount of similarity data (e.g., all BLAST hits *versus* best hits), and clustering algorithm has a bearing on the biological soundness of protein similarity groups (chapter 5).

SUMMARY

The quest for understanding how proteins evolve and function has been a prominent and costly human endeavor. With advances in genomics and use of bioinformatics tools, the diversity of proteins in present day genomes can now be studied more efficiently than ever before. This thesis describes computational methods suitable for large-scale protein classification of many proteomes of diverse species. Specifically, we focus on methods that combine unsupervised learning (clustering) techniques with the knowledge of molecular phylogenetics, particularly that of orthology.

In **chapter 1** we introduce the biological context of protein structure, function and evolution, review the state-of-the-art sequence-based protein classification methods, and then describe methods used to validate the predictions. Finally, we present the outline and objectives of this thesis.

Evolutionary (phylogenetic) concepts are instrumental in studying subjects as diverse as the diversity of genomes, cellular networks, protein structures and functions, and functional genome annotation. In particular, the detection of orthologous proteins (genes) across genomes provides reliable means to infer biological functions and processes from one organism to another. **Chapter 2** evaluates the available computational tools, such as algorithms and databases, used to infer orthologous relationships between genes from fully sequenced genomes. We discuss the main caveats of large-scale orthology detection in general as well as the merits and pitfalls of each method in particular. We argue that establishing true orthologous relationships requires a phylogenetic approach which combines both trees and graphs (networks), reliable species phylogeny, genomic data for more than two species, and an insight into the processes of molecular evolution. Also proposed is a set of guidelines to aid researchers in selecting the correct tool. Moreover, this review motivates further research in developing reliable and scalable methods for functional and phylogenetic classification of large protein collections.

Chapter 3 proposes a framework in which various protein knowledge-bases are combined into unique network of mappings (links), and hence allows comparisons to be made between expert curated and fully-automated protein classifications from a single entry point. We developed an integrated annotation resource for protein orthology, *ProGMap* (Protein Group Mappings, http://www.bioinformatics.nl/progmap), to help researchers and database annotators who often need to assess the coherence of proposed annotations and/or group assignments, as well as users of high throughput methodologies (e.g., microarrays or proteomics) who deal with partially annotated genomic data. ProGMap is based on a non-redundant dataset of over 6.6 million protein sequences which is mapped to 240,000 protein group descriptions collected from UniProt, RefSeq, Ensembl, COG, KOG, OrthoMCL-DB, HomoloGene, TRIBES and PIRSF using a fast and fully automated sequence-based mapping approach. The ProGMap database is equipped with a web interface that enables queries to be made using synonymous sequence identifiers, gene symbols, protein functions, and amino acid or nucleotide sequences. It incorporates also services, namely BLAST similarity search and QuickMatch identity search, for finding sequences similar (or identical) to a query sequence, and tools for presenting the results in graphic form.

Graphs (networks) have gained an increasing attention in contemporary biology because

they have enabled complex biological systems and processes to be modeled and better understood. For example, protein similarity networks constructed of all-*versus*-all sequence comparisons are frequently used to delineate similarity groups, such as protein families or orthologous groups in comparative genomics studies. **Chapter 4.1** presents a benchmark study of freely available graph software used for this purpose. Specifically, the computational complexity of the programs is investigated using both simulated and biological networks. We show that most available software is not suitable for large networks, such as those encountered in large-scale proteome analyzes, because of the high demands on computational resources. To address this, we developed a fast and memory-efficient graph software, *netclust* (http://www.bioinformatics.nl/netclust/), which can scale to large protein networks, such as those constructed of millions of proteins and sequence similarities, on a standard computer. An extended version of this program called *Multi-netclust* is presented in **chapter 4.2**. This tool that can find connected clusters of data presented by different network data sets. It uses user-defined threshold values to combine the data sets in such a way that clusters connected in all or in either of the networks can be retrieved efficiently.

Automated protein sequence clustering is an important task in genome annotation projects and phylogenomic studies. During the past years, several protein clustering programs have been developed for delineating protein families or orthologous groups from large sequence collections. However, most of these programs have not been benchmarked systematically, in particular with respect to the trade-off between computational complexity and biological soundness. In **chapter 5** we evaluate three best known algorithms on different protein similarity networks and validation (or 'gold' standard) data sets to find out which one can scale to hundreds of proteomes and still delineate high quality similarity groups at the minimum computational cost. For this, a reliable partition-based approach was used to assess the biological soundness of predicted groups using known protein functions, manually curated protein/domain families and orthologous groups available in expert-curated databases. Our benchmark results support the view that a simple and computationally cheap method such as netclust can perform similar to and in cases even better than more sophisticated, yet much more costly methods. Moreover, we introduce an efficient graph-based method that can delineate protein orthologs of hundreds of proteomes into hierarchical similarity groups *de novo*. The validity of this method is demonstrated on data obtained from 347 prokaryotic proteomes. The resulting hierarchical protein classification is not only in agreement with manually curated classifications but also provides an enriched framework in which the functional and evolutionary relationships between proteins can be studied at various levels of specificity.

Finally, in **chapter 6** we summarize the main findings and discuss the merits and shortcomings of the methods developed herein. We also propose directions for future research.

The ever increasing flood of new sequence data makes it clear that we need improved tools to be able to handle and extract relevant (orthological) information from these protein data. This thesis summarizes these needs and how they can be addressed by the available tools, or be improved by the new tools that were developed in the course of this research.

SAMENVATTING

Het onderzoek naar de ontwikkeling en het functioneren van eiwitten is een belangrijke en tevens kostbare inspanning. Met de nu beschikbare schat aan genomische informatie en geavanceerde bioinformatica software kan de verscheidenheid aan eiwitten efficiënter dan ooit te voren worden bestudeerd. Dit proefschrift belicht methoden voor grootschalige eiwitclassificatie en beschrijft hun bruikbaarheid bij de analyse van de in hoog tempo beschikbaar komende proteoom informatie. We richten ons in het bijzonder op methoden die clustertechnieken ("unsupervised learning") combineren met kennis van de moleculaire fylogenie, in het bijzonder de orthologie.

In **hoofdstuk 1** introduceren we de biologische context, met name de termen eiwitstructuur, -functie en -evolutie, bespreken we de geavanceerde eiwit classificatiemethodes gebaseerd op en beschrijven we methodes die gebruikt kunnen worden om voorspellingen te valideren. Tot slot zetten we de hoofdlijn en doelstellingen van dit proefschrift uiteen.

Evolutionaire (fylogenetische) concepten zijn essentieel bij het bestuderen van onderwerpen zo divers als cellulaire netwerken, eiwitstructuur en -functie, en functionele genoomannotatie. Zo kan de biologische functie van nieuw gevonden eiwitten worden bepaald door te kijken naar evolutionair nauw verwante (orthologe) eiwitten die goed gekarakteriseerd zijn. **Hoofdstuk 2** evalueert de beschikbare software, algoritmes en databanken die gebruikt worden voor het afleiden van orthologierelaties tussen genen van volledig opgehelderde genomen. We behandelen de belangrijkste problemen van grootschalige orthologiedetectie in het algemeen, als ook de voordelen en tekortkomingen van elke methode afzonderlijk. We beargumenteren dat het ophelderen van orthologierelaties een gecombineerde fylogenetische benadering vereist die gebruik maakt van bomen en grafen (netwerken), een betrouwbare fylogenie van de onderliggende soorten, genoominformatie voor meer dan twee soorten, en inzicht in de processen die een rol spelen in moleculaire evolutie. Daarnaast stellen we een aantal richtlijnen op om wetenschappers te helpen bij het selecteren van het juiste instrument. Tot slot motiveert dit overzicht verder onderzoek naar het ontwikkelen van betrouwbare en schaalbare methoden voor functionele en fylogenetische classificatie van grote verzamelingen eiwitvolgordes.

Hoofdstuk 3 introduceert een systeem waarin verschillende eiwitdatabanken gecombineerd zijn tot een uniek netwerk van verbanden (koppelingen). Hiermee is het mogelijk om bijvoorbeeld automatisch gegenereerde eiwitclassificaties te vergelijken met door experts gecureerde. Dit geïntegreerde systeem, *ProGMap* genaamd (Protein Group Mappings, http://www.bioinformatics.nl/progmap), helpt wetenschappers en databankannotators om de coherentie van annotaties en/of groepstoewijzingen te onderzoeken. Gebruikers van 'high throughput' technieken als microarrays en proteomics kunnen met ProGMap alle gedocumenteerde annotaties voor een bepaalde aminozuurvolgorde vinden. ProGMap is gebaseerd op een niet-redundante dataset van meer dan 6,6 miljoen eiwitketens, gekoppeld aan 240.000 eitwitgroepbeschrijvingen afkomstig uit een groot aantal eiwit- en clusterdatabanken, en maakt gebruik van een snelle, volledig geautomatiseerde en op aminozuurvolgorde gebaseerde identificatiemethode. De ProGMap databank is voorzien van een web-interface die het mogelijk maakt om te zoeken op eiwit-ID, accessienummer, gensymbool, eiwitfunctie en aminozuur - of nucleotidevolgorde. ProGMap bevat ook hulpmiddelen voor het vinden

van sequenties die identiek of verwant zijn aan een gegeven sequentie, namelijk BLAST (zoeken naar overeenkomsten) en QuickMatch (zoeken naar identiteit). De web-interface biedt tevens de mogelijkheid om een grafisch overzicht van de resultaten te presenteren.

Grafen (netwerken) genieten een toenemende belangstelling in de biologie, omdat zij het mogelijk maken ingewikkelde biologische systemen en processen te modeleren en beter te begrijpen. Zo worden bijvoorbeeld netwerken van overeenkomstige eiwitten ("similarity networks") opgebouwd uit 'all *versus* all' sequentievergelijkingen, veel gebruikt in 'comparative genomics studies. **Hoofdstuk 4.1** presenteert een vergelijkende studie van vrij beschikbare software die voor dit doel wordt gebruikt. Hierbij is vooral de rekenkundige complexiteit van de programma's onderzocht, gebruikmakend van zowel biologische als gesimuleerde netwerken. We laten zien dat de meeste software niet geschikt is voor de grote netwerken zoals men die tegenkomt bij grootschalige analyse van proteomen, door de hoge eisen die gesteld worden aan de beschikbare computerinfrastructuur. Om dit probleem op te lossen hebben we een snel en geheugenefficiënt computerprogramma ontwikkeld, *netclust* genaamd (http://www.bioinformatics.nl/netclust). Dit progamma kan grote eiwitnetwerken bestaande uit miljoenen eiwitten en sequentierelaties verwerken op een standaard PC. Een uitgebreidere versie van deze software, *Multi-netclust*, wordt gepresenteerd in **hoofdstuk 4.2**. Dit programma kan met elkaar verbonden clusters vinden door de gegevens uit verschillende netwerkdatasets met elkaar te combineren.

Geautomatiseerde clustering van sequenties is een belangrijke taak in genoomannotatieprojecten en fylogenomische studies. In de afgelopen jaren zijn verschillende eiwitclusterprogramma's ontwikkeld om eiwitfamilies of orthologe groepen in grote sequentie databanken te detecteren. Echter, de meeste van deze programma's zijn niet systematisch onderzocht, in het bijzonder op hun rekenkundige complexiteit en biologische diepgang. In **hoofdstuk 5** evalueren we drie veel gebruikte algoritmes met verschillende eiwitnetwerken en referentie datasets om te onderzoeken welk van deze methodes in staat is om grote aantallen (i.e. honderden) proteomen te hanteren en tegelijkertijd een zo hoog mogelijke kwaliteit tegen zo laag mogelijke rekenkosten te waarborgen. Hiervoor is een betrouwbare, op partitionering gebaseerde, benadering gebruikt om de biologische relevantie van de voorspelde groepen te bepalen, gebruik makend van bekende eiwitfuncties, handmatig gecureerde eiwit- en domeinfamilies en orthologe groepen zoals beschikbaar in gecureerde databanken. Onze resultaten tonen aan dat een eenvoudige en rekenkundig goedkope methode zoals netclust even goed en in sommige gevallen zelfs beter kan presteren dan meer geavanceerde, rekenintensieve methoden. Daarnaast introduceren we een efficiëte op grafen gebaseerde methode die eiwitten van honderden proteomen kan indelen in hiërarchische orthologe groepen. De kracht van deze methode wordt gedemonstreerd aan de hand van 347 prokaryotische proteomen; de resulterende hiërarchische eiwitclassificatie is niet alleen in overeenstemming met handmatig gecureerde classificaties, maar maakt het ook mogelijk om de functies van en evolutionaire relaties tussen eiwitten op verschillende niveaus te bestuderen.

Tot slot vatten we in **hoofdstuk 6** de voornaamste bevindingen samen en bespreken we de voordelen en tekortkomingen van de hierboven ontwikkelde methoden. We doen ook suggesties voor toekomstig onderzoek.

De nog steeds groeiende stroom aan nieuwe sequentiegegevens maakt het duidelijk dat we betere instrumenten nodig hebben om te kunnen omgaan met deze eiwitgegevens en er relevante (orthologe) informatie uit te extraheren. Dit proefschrift beschrijft deze behoeftes en hoe ze kunnen worden aangepakt met de beschikbare instrumenten of met de nieuwe methoden die in de loop van dit onderzoek zijn ontwikkeld.

REFERENCES

Abascal, F. and Valencia, A.: 2002, Clustering of proximal sequence space for the identification of protein families., *Bioinformatics* 18, 908–921.

Aittokallio, T. and Schwikowski, B.: 2006, Graph-based methods for analysing networks in cell biology., *Brief Bioinform* 7, 243–255.

Alako, B. T. F., Rainey, D., Nijveen, H. and Leunissen, J. A. M.: 2006, TreeDomViewer: a tool for the visualization of phylogeny and protein domain structure., *Nucleic Acids Res* 34, W104–W109.

Alexeyenko, A., Tamas, I., Liu, G. and Sonnhammer, E. L.: 2006, Automatic clustering of orthologs and inparalogs shared by multiple proteomes., *Bioinformatics* 22, e9–15.

Alibés, A., Yankilevich, P., Cañada, A. and Díaz-Uriarte, R.: 2007, IDconverter and IDClight: conversion and annotation of gene and protein IDs., *BMC Bioinformatics* 8, 9.

Alkhalfioui, F., Magnin, T. and Wagner, R.: 2009, From purified GPCRs to drug discovery: the promise of protein-based methodologies., *Curr Opin Pharmacol* .

Altschul, S. F.: 1991, Amino acid substitution matrices from an information theoretic perspective., *J Mol Biol* 219, 555–565.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J.: 1990, Basic local alignment search tool., *J Mol Biol* 215, 403–410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D. J.: 1997, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs., *Nucleic Acids Res* 25, 3389–3402.

Andreeva, A., Howorth, D., Chandonia, J. M., Brenner, S. E., Hubbard, T. J. P., Chothia, C. and Murzin, A. G.: 2008, Data growth and its impact on the SCOP database: new developments., *Nucleic Acids Res* 36, D419–D425.

Anfinsen, C. B.: 1973, Principles that govern the folding of protein chains., *Science* 181, 223–230.

Apic, G., Gough, J. and Teichmann, S. A.: 2001, Domain combinations in archaeal, eubacterial and eukaryotic proteomes., *J Mol Biol* 310, 311–325.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M. and Sherlock, G.: 2000, Gene ontology: tool for the unification of biology. The Gene Ontology Consortium., *Nat Genet* 25, 25–29.

Ayala, F. J.: 1997, Vagaries of the molecular clock., *Proc Natl Acad Sci U S A* 94, 7776–7783.

Babushok, D. V., Ostertag, E. M. and Kazazian, H. H.: 2007, Current topics in genome evolution: molecular mechanisms of new gene formation., *Cell Mol Life Sci* 64, 542–554.

Bairoch, A.: 1992, PROSITE: a dictionary of sites and patterns in proteins., *Nucleic Acids Res* 20 Suppl, 2013–2018.

Bairoch, A.: 2000, The ENZYME database in 2000., *Nucleic Acids Res* 28, 304–305.

Bajić, V. B.: 2000, Comparing the success of different prediction software in sequence analysis: a review., *Brief Bioinform* 1, 214–228.

Baldi, P. and Brunak, S.: 2001, *Bioinformatics: The Machine Learning Approach.*, 2 edn, The MIT press.

Bandyopadhyay, S., Sharan, R. and Ideker, T.: 2006, Systematic identification of functional orthologs based on protein network comparison., *Genome Res* 16, 428–435.

Barabási, A. L. and Oltvai, Z. N.: 2004, Network biology: understanding the cell's functional organization., *Nat Rev Genet* 5, 101–113.

Bashton, M. and Chothia, C.: 2007, The generation of new protein functions by the combination of domains., *Structure* 15, 85–99.

Basu, M. K., Carmel, L., Rogozin, I. B. and Koonin, E. V.: 2008, Evolution of protein domain promiscuity in eukaryotes., *Genome Res* 18, 449–461.

Benson, D. A., M., I. K., Lipman, D. J., Ostell, J. and Sayers, E. W.: 2009, GenBank., *Nucleic Acids Res* 37, D26–D31.

Berman, H. M., Westbrook, J. D., Gabanyi, M. J., Tao, W., Shah, R., Kouranov, A., Schwede, T., Arnold, K., Kiefer, F., Bordoli, L., Kopp, J., Podvinec, M., Adams, P. D., Carter, L. G., Minor, W., Nair, R. and Baer, J. L.: 2009, The protein structure initiative structural genomics knowledgebase., *Nucleic Acids Res* 37, D365–D368.

Bernardi, G.: 2007, The neoselectionist theory of genome evolution., *Proc Natl Acad Sci U S A* 104, 8385–8390.

Birzele, F., Csaba, G. and Zimmer, R.: 2008, Alternative splicing and protein structure evolution., *Nucleic Acids Res* 36, 550–558.

Blair, J. E. and Hedges, S. B.: 2005, Molecular phylogeny and divergence times of deuterostome animals., *Mol Biol Evol* 22, 2275–2284.

Bolten, E., Schliep, A., Schneckener, S., Schomburg, D. and Schrader, R.: 2001, Clustering protein sequences–structure prediction by transitive homology., *Bioinformatics* 17, 935–941.

Bork, P.: 1991, Shuffled domains in extracellular proteins., *FEBS Lett* 286, 47–54.

Bork, P., Dandekar, T., Diaz-Lazcoz, Y., Eisenhaber, F., Huynen, M. and Yuan, Y.: 1998, Predicting function: from genes to genomes and back., *J Mol Biol* 283, 707–725.

Brenner, S. E.: 1999, Errors in genome annotation., *Trends Genet* 15, 132–133.

Bron, C. and Kerbosch, J.: 1973, Algorithm 457: finding all cliques of an undirected graph, *Commun ACM* 16, 575–577.

Bussey, K. J., Kane, D., Sunshine, M., Narasimhan, S., Nishizuka, S., Reinhold, W. C., Zeeberg, B., Ajay, W. and Weinstein, J. N.: 2003, MatchMiner: a tool for batch navigation among gene and gene product identifiers., *Genome Biol* 4, R27.

Cannon, S. B. and Young, N. D.: 2003, OrthoParaMap: distinguishing orthologs from paralogs by integrating comparative genome data and gene phylogenies., *BMC Bioinformatics* 4, 35.

Carroll, S. and Pavlovic, V.: 2006, Protein classification using probabilistic chain graphs and the Gene Ontology structure., *Bioinformatics* 22, 1871–1878.

Cases, I., Pisano, D. G., Andres, E., Carro, A., Fernández, J. M., Gómez-López, G., Rodriguez, J. M., Vera, J. F., Valencia, A. and Rojas, A. M.: 2007, CARGO: a web portal to integrate customized biological information., *Nucleic Acids Res* 35, W16–W20.

Chen, F., Mackey, A. J., Stoeckert, C. J. and Roos, D. S.: 2006, OrthoMCL-DB: querying a comprehensive multi-species collection of ortholog groups., *Nucleic Acids Res* 34, D363–D368.

Chen, F., Mackey, A. J., Vermunt, J. K. and Roos, D. S.: 2007, Assessing performance of orthology detection strategies applied to eukaryotic genomes., *PLoS ONE* 2, e383.

Chiang, Y. J., Goodrich, M. T., Grove, E. F., Tamassia, R., Vengroff, D. E. and Vitter, J. S.: 1995, *External-Memory Graph Algorithms*, Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95), San Francisco, PA.

Choi, S. S. and Lahn, B. T.: 2003, Adaptive evolution of MRG, a neuron-specific gene family implicated in nociception., *Genome Res* 13, 2252–2259.

Chothia, C. and Lesk, A. M.: 1986, The relation between the divergence of sequence and structure in proteins., *EMBO J* 5, 823–826.

Ciccarelli, F. D., Doerks, T., von Mering, C., Creevey, C. J., Snel, B. and Bork, P.: 2006, Toward automatic reconstruction of a highly resolved tree of life., *Science* 311, 1283–1287.

Cochrane, G., Akhtar, R., Bonfield, J., Bower, L., Demiralp, F., Faruque, N., Gibson, R., Hoad, G., Hubbard, T., Hunter, C., Jang, M., Juhos, S., Leinonen, R., Leonard, S., Lin, Q., Lopez, R., Lorenc, D., McWilliam, H., Mukherjee, G., Plaister, S., Radhakrishnan, R., Robinson, S., Sobhany, S., Hoopen, P. T., Vaughan, R., Zalunin, V. and Birney, E.: 2009, Petabyte-scale innovations at the European Nucleotide Archive., *Nucleic Acids Res* 37, D19–D25.

Consortium, U.: 2009, The Universal Protein Resource (UniProt) 2009., *Nucleic Acids Res* 37, D169–D174.

Cuff, A. L., Sillitoe, I., L., T., Redfern, O. C., Garratt, R., Thornton, J. and Orengo, C. A.: 2009, The CATH classification revisited–architectures reviewed and new ways to characterize structural divergence in superfamilies., *Nucleic Acids Res* 37, D310–D314.

Datta, R. S., Meacham, C., Samad, B., Neyer, C. and Sjölander, K.: 2009, Berkeley PHOG: PhyloFacts orthology group prediction web server., *Nucleic Acids Res* 37, W84–W89.

Dayhoff, M. O.: 1976, The origin and evolution of protein superfamilies., *Fed Proc* 35, 2132–2138.

Dayhoff, M. O., Hunt, L. T., Barker, W. C., Schwartz, R. M., Orcutt, B. C. and Young, C. L.: 1978, *Atlas of Protein Sequence and Structure.*, Vol. 5, Nat. Biomed Res Found.

Decottignies, A., Sanchez-Perez, I. and Nurse, P.: 2003, Schizosaccharomyces pombe essential genes: a pilot study., *Genome Res* 13, 399–406.

Defays, D.: 1977, An efficient algorithm for a complete link method., *Comput J* 20, 364–366.

Dehal, P. S. and Boore, J. L.: 2006, A phylogenomic gene cluster resource: the Phylogenetically Inferred Groups (PhIGs) database., *BMC Bioinformatics* 7.

Delsuc, F., Brinkmann, H. and Philippe, H.: 2005, Phylogenomics and the reconstruction of the tree of life., *Nat Rev Genet* 6, 361–375.

Deluca, T. F., Wu, I., Pu, J., Monaghan, T., Peshkin, L., Singh, S. and Wall, D. P.: 2006, Roundup: a multi-genome repository of orthologs and evolutionary distances., *Bioinformatics* 22, 2044–2046.

Devos, D. and Valencia, A.: 2001, Intrinsic errors in genome annotation., *Trends Genet* 17, 429–431.

Diehn, M., Sherlock, G., Binkley, G., Jin, H., Matese, J. C., Hernandez-Boussard, T., Rees, C. A., Cherry, J. M., Botstein, D., Brown, P. O. and Alizadeh, A. A.: 2003, SOURCE: a unified genomic resource of functional annotations, ontologies, and gene expression data., *Nucleic Acids Res* 31, 219–223.

Donald, J. E. and Shakhnovich, E. I.: 2005, Determining functional specificity from protein sequences., *Bioinformatics* 21, 2629–2635.

Doolittle, R. F.: 1987, *Of URFs and ORFs: a primer on how to analyze derived amino acid sequences.*, University Science Books, California, USA.

Doolittle, R. F.: 1995, The multiplicity of domains in proteins., *Annu Rev Biochem* 64, 287–314.

Doolittle, W. F. and Bapteste, E.: 2007, Pattern pluralism and the Tree of Life hypothesis., *Proc Natl Acad Sci U S A* 104, 2043–2049.

Drummond, D. A., Bloom, J. D., Adami, C., Wilke, C. O. and Arnold, F. H.: 2005, Why highly expressed proteins evolve slowly., *Proc Natl Acad Sci U S A* 102, 14338–14343.

Duda, R. O., Hart, P. E. and Stork, D. G.: 2000, *Pattern Classification (2nd Edition)*, Wiley-Interscience, New York, USA.

Dufayard, J. F., Duret, L., Penel, S., Guy, M., Rechenmann, F. and Perrière, G.: 2005, Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases., *Bioinformatics* 21, 2596–2603.

Dunn, C. W., Hejnol, A., Matus, D. Q., Pang, K., Browne, W. E., Smith, S. A., Seaver, E., Rouse, G. W., Obst, M., Edgecombe, G. D., Srensen, M. V., Haddock, S. H. D., Schmidt-Rhaesa, A., Okusu, A., Kristensen, R. M., Wheeler, W. C., Martindale, M. Q. and Giribet, G.: 2008, Broad phylogenomic sampling improves resolution of the animal tree of life., *Nature* 452, 745–749.

Durand, D. and Hoberman, R.: 2006, Diagnosing duplications–can it be done?, *Trends Genet* 22, 156–164.

Duret, L., Mouchiroud, D. and Gouy, M.: 1994, HOVERGEN: a database of homologous vertebrate genes., *Nucleic Acids Res* 22, 2360–2365.

Eisen, J. A.: 1998a, Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis., *Genome Res* 8, 163–167.

Eisen, J. A.: 2000, Horizontal gene transfer among microbial genomes: new insights from complete genome analysis., *Curr Opin Genet Dev* 10, 606–611.

Eisen, M. B., Spellman, P. T., Brown, P. O. and Botstein, D.: 1998b, Cluster analysis and display of genome-wide expression patterns., *Proc Natl Acad Sci U S A* 95, 14863–14868.

Enright, A. J., Dongen, S. V. and Ouzounis, C. A.: 2002, An efficient algorithm for large-scale detection of protein families., *Nucleic Acids Res* 30, 1575–1584.

Enright, A. J., Iliopoulos, I., Kyrpides, N. C. and Ouzounis, C. A.: 1999, Protein interaction maps for complete genomes based on gene fusionevents., *Nature* 402, 86–90.

Enright, A. J., Kunin, V. and Ouzounis, C. A.: 2003, Protein families and TRIBES in genome sequence space., *Nucleic Acids Res* 31, 4632–4638.

Enright, A. J. and Ouzounis, C. A.: 2000, GeneRAGE: a robust algorithm for sequence clustering and domain detection., *Bioinformatics* 16, 451–457.

Erdős, P. and Rényi, A.: 1959, On random graphs., *Publ Math* 6, 290–297.

Etzold, T. and Argos, P.: 1993, SRS–an indexing and retrieval tool for flat file data libraries., *Comput Appl Biosci* 9, 49–57.

Eyre, T. A., Wright, M. W., Lush, M. J. and Bruford, E. A.: 2007, HCOP: a searchable database of human orthology predictions., *Brief Bioinform* 8, 2–5.

Farrar, M.: 2007, Striped Smith-Waterman speeds database searches six times over other SIMD implementations., *Bioinformatics* 23, 156–161.

Felsenstein, J.: 1988, Phylogenies from molecular sequences: inference and reliability., *Annu Rev Genet* 22, 521–565.

Felsenstein, J.: 1989, PHYLIP - Phylogeny Inference Package (Version 3.2)., *Cladistics* 5, 164–166.

Finn, R. D., Tate, J., Mistry, J., Coggill, P. C., Sammut, S. J., Hotz, H. R., Ceric, G., Forslund, K., Eddy, S. R., Sonnhammer, E. L. and Bateman, A.: 2008, The Pfam protein families database., *Nucleic Acids Res* 36, D281–D288.

Fitch, W. M.: 1970, Distinguishing homologous from analogous proteins., *Syst Zool* 19, 99–113.

Fitch, W. M.: 2000, Homology a personal view on some of the problems., *Trends Genet* 16, 227–231.

Flicek, P., Aken, B. L., Beal, K., Ballester, B., Caccamo, M., Chen, Y., Clarke, L., Coates, G., Cunningham, F., Cutts, T., Down, T., Dyer, S. C., Eyre, T., Fitzgerald, S., Fernandez-Banet, J., Gräf, S., Haider, S., Hammond, M., Holland, R., Howe, K. L., Howe, K., Johnson, N., Jenkinson, A., Kähäri, A., Keefe, D., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Megy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Prlic, A., Rice, S., Rios, D., Schuster, M., Sealy, I., Slater, G., Smedley, D., Spudich, G., Trevanion, S., Vilella, A. J., Vogel, J., White, S., Wood, M., Birney, E., Cox, T., Curwen, V., Durbin, R., Fernandez-Suarez, X. M., Herrero, J., Hubbard, T. J. P., Kasprzyk, A., , Proctor, G., Smith, J., Ureta-Vidal, A. and Searle, S.: 2008, Ensembl 2008., *Nucleic Acids Res* 36, D707–D714.

Fulton, D. L., Li, Y. Y., Laird, M. R., Horsman, B. G. S., Roche, F. M. and Brinkman, F. S. L.: 2006, Improving the specificity of high-throughput ortholog prediction., *BMC Bioinformatics* 7, 270.

Goodman, M., Czelusniak, J., Moore, G. W., Romero-Herrera, A. E. and Matsuda, G.: 1979, Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by cladograms constructed from globin sequences., *Syst Zool* 28, 132–163.

Goodstadt, L. and Ponting, C. P.: 2006, Phylogenetic reconstruction of orthology, paralogy, and conserved synteny for dog and human., *PLoS Comput Biol* 2, e133.

Grant, A., Lee, D. and Orengo, C.: 2004, Progress towards mapping the universe of protein folds., *Genome Biol* 5, 107.

Grigoryev, D. N., Ma, S. F., Irizarry, R. A., Ye, S. Q., Quackenbush, J. and Garcia, J. G. N.: 2004, Orthologous gene-expression profiling in multi-species models: search for candidate genes., *Genome Biol* 5, R34.

Gupta, R. S.: 2001, The branching order and phylogenetic placement of species from completed bacterial genomes, based on conserved indels found in various proteins., *Int Microbiol* 4, 187–202.

Halkidi, M., Batistakis, Y. and Vazirgiannis, M.: 2001, On Clustering Validation Techniques., *J Intell Inf Syst* 17, 107–145.

Handl, J., Knowles, J. and Kell, D. B.: 2005, Computational cluster validation in post-genomic data analysis., *Bioinformatics* 21, 3201–3212.

Harlow, T. J., Gogarten, J. P. and Ragan, M. A.: 2004, A hybrid clustering approach to recognition of protein families in 114 microbial genomes., *BMC Bioinformatics* 5, 45.

Hartuv, E., Schmitt, A. O., Lange, J., Meier-Ewert, S., Lehrach, H. and Shamir, R.: 2000, An algorithm for clustering cDNA fingerprints., *Genomics* 66, 249–256.

Hasegawa, M. and Kishino, H.: 1989, Heterogeneity of tempo and mode of mitochondrial DNA evolution among mammalian orders., *Jpn J Genet* 64, 243–258.

Heger, A., Korpelainen, E., Hupponen, T., Mattila, K., Ollikainen, V. and Holm, L.: 2008, PairsDB atlas of protein sequence space., *Nucleic Acids Res* 36, D276–D280.

Henikoff, S. and Henikoff, J. G.: 1992, Amino acid substitution matrices from protein blocks., *Proc Natl Acad Sci U S A* 89, 10915–10919.

Hillis, D. M.: 1994, *Homology in molecular biology.*, In Homology, the Hierarchical Basis of Comparative Biology, Academic press.

Hirsh, A. E. and Fraser, H. B.: 2001, Protein dispensability and rate of evolution., *Nature* 411, 1046–1049.

Hittinger, C. T. and Carroll, S. B.: 2007, Gene duplication and the adaptive evolution of a classic genetic switch., *Nature* 449, 677–681.

Hoffmann, R. and Valencia, A.: 2004, A gene network for navigating the literature., *Nat Genet* 36, 664.

Holm, L. and Sander, C.: 1995, Dali: a network tool for protein structure comparison, *Trends Biochem Sci* 20, 478–480.

Horan, K., Lauricha, J., Bailey-Serres, J., Raikhel, N. and Girke, T.: 2005, Genome cluster database. A sequence family analysis platform for Arabidopsis and rice., *Plant Physiol* 138, 47–54.

Huang, D. W., Sherman, B. T. and Lempicki, R. A.: 2009, Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists., *Nucleic Acids Res* 37, 1–13.

Hubbard, T. J. P., Aken, B. L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., Coates, G., Fairley, S., Fitzgerald, S., Fernandez-Banet, J., Gordon, L., Graf, S., Haider, S., Hammond, M., Holland, R., Howe, K., Jenkinson, A., Johnson, N., Kahari, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Megy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Rios, D., Schuster, M., Slater, G., Smedley, D., Spooner, W., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wilder, S., Zadissa, A., Birney, E., Cunningham, F., Curwen, V., Durbin, R., Fernandez-Suarez, X. M., Herrero, J., Kasprzyk, A., Proctor, G., Smith, J., Searle, S. and Flicek, P.: 2009, Ensembl 2009., *Nucleic Acids Res* 37, D690–D697.

Huber, W., Carey, V. J., Long, L., Falcon, S. and Gentleman, R.: 2007, Graphs in molecular biology., *BMC Bioinformatics* 8 Suppl 6, S8.

Huerta-Cepas, J., Bueno, A., Dopazo, J. and Gabaldón, T.: 2008, PhylomeDB: a database for genome-wide collections of gene phylogenies., *Nucleic Acids Res* 36, D491–D496.

Hughes, A. L.: 1994, The evolution of functionally novel proteins after gene duplication., *Proc Biol Sci* 256, 119–124.

Hughes, A. L. and Nei, M.: 1989, Nucleotide substitution at major histocompatibility complex class II loci: evidence for overdominant selection., *Proc Natl Acad Sci U S A* 86, 958–962.

Hulo, N., Bairoch, A., Bulliard, V., Cerutti, L., Cuche, B. A., de Castro, E., Lachaize, C., Langendijk-Genevaux, P. S. and Sigrist, C. J. A.: 2008, The 20 years of PROSITE., *Nucleic Acids Res* 36, D245–D249.

Hulsen, T., de Vlieg, J., Leunissen, J. A. M. and Groenen, P. M. A.: 2006b, Testing statistical significance scores of sequence comparison methods with structure similarity., *BMC Bioinformatics* 7, 444.

Hulsen, T., Huynen, M. A., de Vlieg, J. and Groenen, P. M. A.: 2006a, Benchmarking ortholog identification methods using functional genomics data., *Genome Biol* 7, R31.

Hunter, S., Apweiler, R., Attwood, T. K., Bairoch, A., Bateman, A., Binns, D., Bork, P., Das, U., Daugherty, L., Duquenne, L., Finn, R. D., Gough, J., Haft, D., Hulo, N., Kahn, D., Kelly, E., Laugraud, A., Letunic, I., Lonsdale, D., Lopez, R., Madera, M., Maslen, J., McAnulla, C., McDowall, J., Mistry, J., Mitchell, A., Mulder, N., Natale, D., Orengo, C., Quinn, A. F., Selengut, J. D., Sigrist, C. J. A., Thimma, M., Thomas, P. D., Valentin, F., Wilson, D., Wu, C. H. and Yeats, C.: 2009, InterPro: the integrative protein signature database., *Nucleic Acids Res* 37, D211–D215.

Huynen, M. A. and Bork, P.: 1998, Measuring genome evolution., *Proc Natl Acad Sci U S A* 95, 5849–5856.

IUBMB: 1992, *Enzyme Nomenclature (1992) Recommendations of the Nomenclature Committe of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*, Academic Press, San Diego, USA.

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, 547–579.

Jain, A. K., Murty, M. N. and Flynn, P. J.: 1999, Data clustering: A review., *ACM Comput Surv* 3, 264–323.

Jeffroy, O., Brinkmann, H., Delsuc, F. and Philippe, H.: 2006, Phylogenomics: the beginning of incongruence?, *Trends Genet* 22, 225–231.

Jensen, L. J., Julien, P., Kuhn, M., von Mering, C., Muller, J., Doerks, T. and Bork, P.: 2008, eggNOG: automated construction and annotation of orthologous groups of genes., *Nucleic Acids Res* 36, D250–D254.

Jensen, R. A.: 1976, Enzyme recruitment in evolution of new function., *Annu Rev Microbiol* 30, 409–425.

Joseph, J. M. and Durand, D.: 2009, Family classification without domain chaining., *Bioinformatics* 25, i45–i53.

Jothi, R., Zotenko, E., Tasneem, A. and Przytycka, T. M.: 2006, COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations., *Bioinformatics* 22, 779–788.

Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M. and Hirakawa, M.: 2006, From genomics to chemical genomics: new developments in KEGG., *Nucleic Acids Res* 34, D354–D357.

Kaplan, N., Friedlich, M., Fromer, M. and Linial, M.: 2004, A functional hierarchical organization of the protein sequence space., *BMC Bioinformatics* 5, 196.

Kaplan, N., Sasson, O., Inbar, U., Friedlich, M., Fromer, M., Fleischer, H., Portugaly, E., Linial, N. and Linial, M.: 2005, ProtoNet 4.0: a hierarchical classification of one million proteinsequences., *Nucleic Acids Res* 33, D216–D218.

Karlin, S. and Altschul, S. F.: 1990, Methods for assessing the statistical significance of molecular sequencefeatures by using general scoring schemes., *Proc Natl Acad Sci U S A* 87, 2264–2268.

Karplus, K., Barrett, C. and Hughey, R.: 1998, Hidden Markov models for detecting remote protein homologies., *Bioinformatics* 14, 846–856.

Kawaji, H., Takenaka, Y. and Matsuda, H.: 2004, Graph-based clustering for finding distant relationships in a large set of protein sequences., *Bioinformatics* 20, 243–252.

Kelil, A., Wang, S., Brzezinski, R. and Fleury, A.: 2007, CLUSS: clustering of protein sequences based on a new similarity measure., *BMC Bioinformatics* 8, 286.

Kersey, P., Bower, L., Morris, L., Horne, A., Petryszak, R., Kanz, C., Kanapin, A., Das, U., Michoud, K., Phan, I., Gattiker, A., Kulikova, T., Faruque, N., Duggan, K., Mclaren, P., Reimholz, B., Duret, L., Penel, S., Reuter, I. and Apweiler, R.: 2005, Integr8 and Genome Reviews: integrated views of complete genomesand proteomes., *Nucleic Acids Res* 33, D297–D302.

Kersey, P. J., Duarte, J., Williams, A., Karavidopoulou, Y., Birney, E. and Apweiler, R.: 2004, The International Protein Index: an integrated database for proteomics experiments., *Proteomics* 4, 1985–1988.

Kertész-Farkas, A., Dhir, S., Sonego, P., Pacurar, M., Netoteia, S., Nijveen, H., Kuzniar, A., Leunissen, J. A. M., Kocsor, A. and Pongor, S.: 2008, Benchmarking protein classification algorithms via supervised cross-validation., *J Biochem Biophys Methods* 70, 1215–1223.

Kim, S., Kang, J., Chung, Y. J., Li, J. and Ryu, K. H.: 2008, Clustering orthologous proteins across phylogenetically distant species., *Proteins* 71, 1113–1122.

Kimura, M.: 1983, Rare variant alleles in the light of the neutral theory., *Mol Biol Evol* 1, 84–93.

Kinch, L. N. and Grishin, N. V.: 2002, Evolution of protein structures and functions., *Curr Opin Struct Biol* 12, 400–408.

Klimke, W., Agarwala, R., Badretdin, A., Chetvernin, S., Ciufo, S., Fedorov, B., Kiryutin, B., O'Neill, K., Resch, W., Resenchuk, S., Schafer, S., Tolstoy, I. and Tatusova, T.: 2009, The National Center for Biotechnology Information's Protein Clusters Database., *Nucleic Acids Res* 37, D216–D223.

Kocsor, A., Kertész-Farkas, A., Kaján, L. and Pongor, S.: 2006, Application of compression-based distance measures to protein sequence classification: a methodological study., *Bioinformatics* 22, 407–412.

Koonin, E. V.: 2005, Orthologs, paralogs, and evolutionary genomics., *Annu Rev Genet* 39, 309–338.

Koonin, E. V., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Krylov, D. M., Makarova, K. S., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Rogozin, I. B., Smirnov, S., Sorokin, A. V., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J. and Natale, D. A.: 2004, A comprehensive evolutionary classification of proteins encoded in complete eukaryotic genomes., *Genome Biol* 5, R7.

Koonin, E. V., Makarova, K. S. and Aravind, L.: 2001, Horizontal gene transfer in prokaryotes: quantification and classification., *Annu Rev Microbiol* 55, 709–742.

Koonin, E. V., Mushegian, A. R. and Bork, P.: 1996, Non-orthologous gene displacement., *Trends Genet* 12, 334–336.

Koonin, E. V., Wolf, Y. I. and Karev, G. P.: 2002, The structure of the protein universe and genome evolution., *Nature* 420, 218–223.

Koski, L. B. and Golding, G. B.: 2001, The closest BLAST hit is often not the nearest neighbor., *J Mol Evol* 52, 540–542.

Krause, A., Stoye, J. and Vingron, M.: 2005, Large scale hierarchical clustering of protein sequences., *BMC Bioinformatics* 6, 15.

Krause, A. and Vingron, M.: 1998, A set-theoretic approach to database searching and clustering., *Bioinformatics* 14, 430–438.

Kriventseva, E. V., Biswas, M. and Apweiler, R.: 2001b, Clustering and analysis of protein families., *Curr Opin Struct Biol* 11, 334–339.

Kriventseva, E. V., Fleischmann, W., Zdobnov, E. M. and Apweiler, R.: 2001a, CluSTr: a database of clusters of SWISS-PROT+TrEMBL proteins., *Nucleic Acids Res* 29, 33–36.

Kriventseva, E. V., Rahman, N., Espinosa, O. and Zdobnov, E. M.: 2008, OrthoDB: the hierarchical catalog of eukaryotic orthologs., *Nucleic Acids Res* 36, D271–D275.

Krummenacker, M., Paley, S., Mueller, L., Yan, T. and Karp, P. D.: 2005, Querying and computing with BioCyc databases., *Bioinformatics* 21, 3454–3455.

Kuang, R., Weston, J., Noble, W. S. and Leslie, C.: 2005, Motif-based protein ranking by network propagation., *Bioinformatics* 21, 3711–3718.

Kull, M. and Vilo, J.: 2008, Fast approximate hierarchical clustering using similarity heuristics., *BioData Min* 1, 9.

Kummerfeld, S. K. and Teichmann, S. A.: 2005, Relative rates of gene fusion and fission in multi-domain proteins., *Trends Genet* 21, 25–30.

Kunin, V., Goldovsky, L., Darzentas, N. and Ouzounis, C. A.: 2005, The net of life: reconstructing the microbial phylogenetic network., *Genome Res* 15, 954–959.

Kunin, V. and Ouzounis, C. A.: 2003, GeneTRACE-reconstruction of gene content of ancestral species., *Bioinformatics* 19, 1412–1416.

Kuzniar, A., Lin, K., He, Y., Nijveen, H., Pongor, S. and Leunissen, J. A. M.: 2009, ProGMap: an integrated annotation resource for protein orthology., *Nucleic Acids Res* 37, W428–W434.

Kuzniar, A., van Ham, R. C. H. J., Pongor, S. and Leunissen, J. A. M.: 2008, The quest for orthologs: finding the corresponding gene across genomes., *Trends Genet* 24, 539–551.

Lao, M. J.: 1981, A Class of Tree-Like UNION-FIND Data Structures and the Non-linearity, *CAAP '81: Proceedings of the 6th Colloquium on Trees in Algebra and Programming*, Springer-Verlag, London, UK, pp. 255–267.

Larget, B. and Simon, D. L.: 1999, Markov chain Monte Carlo algorithms for Bayesian analysis of phylogenetic trees., *Mol Bio Evol* 16, 750–759.

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A. and Robles, V.: 2006, Machine learning in bioinformatics., *Brief Bioinform* 7, 86–112.

Lassmann, T. and Sonnhammer, E. L.: 2005, Automatic assessment of alignment quality., *Nucleic Acids Res* 33, 7120–7128.

Lee, D., Redfern, O. and Orengo, C.: 2007, Predicting protein function from sequence and structure., *Nat Rev Mol Cell Biol* 8, 995–1005.

Lee, Y., Sultana, R., Pertea, G., Cho, J., Karamycheva, S., Tsai, J., Parvizi, B., Cheung, F., Antonescu, V., White, J., Holt, I., Liang, F. and Quackenbush, J.: 2002, Cross-referencing eukaryotic genomes: TIGR Orthologous Gene Alignments (TOGA)., *Genome Res* 12, 493–502.

Lerat, E., Daubin, V., Ochman, H. and Moran, N. A.: 2005, Evolutionary origins of genomic repertoires in bacteria., *PLoS Biol* 3, e130.

Levitt, M. and Gerstein, M.: 1998, A unified statistical framework for sequence comparison and structure comparison., *Proc Natl Acad Sci U S A* 95, 5913–5920.

Li, H., Coghlan, A., Ruan, J., Coin, L. J., Hériché, J. K., Osmotherly, L., Li, R., Liu, T., Zhang, Z., Bolund, L., Wong, G. K. S., Zheng, W., Dehal, P., Wang, J. and Durbin, R.: 2006, TreeFam: a curated database of phylogenetic trees of animal gene families., *Nucleic Acids Res* 34, D572–D580.

Li, L., Stoeckert, C. J. and Roos, D. S.: 2003, OrthoMCL: identification of ortholog groups for eukaryotic genomes., *Genome Res* 13, 2178–2189.

Li, W. and Godzik, A.: 2006, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences., *Bioinformatics* 22, 1658–1659.

Li, W., Jaroszewski, L. and Godzik, A.: 2002, Sequence clustering strategies improve remote homology recognitions while reducing search times., *Protein Eng* 15, 643–649.

Li, W., Wooley, J. C. and Godzik, A.: 2008, Probing Metagenomics by Rapid Cluster Analysis of Very Large Datasets., *PLoS ONE* 3, e3375.

Liebel, U., Kindler, B. and Pepperkok, R.: 2004, 'Harvester': a fast meta search engine of human protein resources., *Bioinformatics* 20, 1962–1963.

Liolios, K., Mavromatis, K., Tavernarakis, N. and Kyrpides, N. C.: 2008, The Genomes On Line Database (GOLD) in 2007: status of genomic andmetagenomic projects and their associated metadata., *Nucleic Acids Res* 36, D475–D479.

Liu, J. and Rost, B.: 2003, Domains, motifs and clusters in the protein universe., *Curr Opin Chem Biol* 7, 5–11.

Loewenstein, Y., Portugaly, E., Fromer, M. and Linial, M.: 2008, Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space., *Bioinformatics* 24, i41–i49.

Loftus, B., Anderson, I., Davies, R., Alsmark, U. C., Samuelson, J., Amedeo, P., Roncaglia, P., Berriman, M., Hirt, R. P., Mann, B. J., Nozaki, T., Suh, B., Pop, M., Duchene, M., Ackers, J., Tannich, E., Leippe, M., Hofer, M., Bruchhaus, I., Willhoeft, U., Bhattacharya, A., Chillingworth, T., Churcher, C., Hance, Z., Harris, B., Harris, D., Jagels, K., Moule, S., Mungall, K., Ormond, D., Squares, R., Whitehead, S., Quail, M. A., Rabbinowitsch, E., Norbertczak, H., Price, C., Wang, Z., Guillén, N., Gilchrist, C., Stroup, S. E., Bhattacharya, S., Lohia, A., Foster, P. G., Sicheritz-Ponten, T., Weber, C., Singh, U., Mukherjee, C., El-Sayed, N. M., Petri, W. A., Clark, C. G., Embley, T. M., Barrell, B., Fraser, C. M. and Hall, N.: 2005, The genome of the protist parasite Entamoeba histolytica., *Nature* 433, 865–868.

Lynch, M. and Conery, J. S.: 2003, The origins of genome complexity., *Science* 302, 1401–1404.

Lynch, M., O'Hely, M., Walsh, B. and Force, A.: 2001, The probability of preservation of a newly arisen gene duplicate., *Genetics* 159, 1789–1804.

Maglott, D., Ostell, J., Pruitt, K. D. and Tatusova, T.: 2005, Entrez Gene: gene-centered information at NCBI., *Nucleic Acids Res* 33, D54–D58.

Mao, F., Su, Z., Olman, V., Dam, P., Liu, Z. and Xu, Y.: 2006, Mapping of orthologous genes in the context of biological pathways: An application of integer programming., *Proc Natl Acad Sci U S A* 103, 129–134.

Marchler-Bauer, A., Anderson, J. B., Chitsaz, F., Derbyshire, M. K., DeWeese-Scott, C., Fong, J. H., Geer, L. Y., Geer, R. C., Gonzales, N. R., Gwadz, M., He, S., Hurwitz, D. I., Jackson, J. D., Ke, Z., Lanczycki, C. J., Liebert, C. A., Liu, C., Lu, F., Lu, S., Marchler, G. H., Mullokandov, M., Song, J. S., Tasneem, A., Thanki, N., Yamashita, R. A., Zhang, D., Zhang, N. and Bryant, S. H.: 2009, CDD: specific functional annotation with the Conserved Domain Database., *Nucleic Acids Res* 37, D205–D210.

Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O. and Eisenberg, D.: 1999, Detecting protein function and protein-protein interactions from genome sequences., *Science* 285, 751–753.

Markowitz, V. M., Szeto, E., Palaniappan, K., Grechkin, Y., Chu, K., Chen, I. M., Dubchak, I., Anderson, I., Lykidis, A., Mavromatis, K., Ivanova, N. N. and Kyrpides, N. C.: 2008, The integrated microbial genomes (IMG) system in 2007: data content and analysis tool extensions., *Nucleic Acids Res* 36, D528–D533.

Marques, A. C., Vinckenbosch, N., Brawand, D. and Kaessmann, H.: 2008, Functional diversification of duplicate genes through subcellular adaptation of encoded proteins., *Genome Biol* 9, R54.

Mazurie, A., Bottani, S. and Vergassola, M.: 2005, An evolutionary and functional assessment of regulatory network motifs., *Genome Biol* 6, R35.

Meilă, M.: 2007, Comparing clusterings–an information based distance., *J Multivar Anal* 98, 873–895.

Meinel, T., Krause, A., Luz, H., Vingron, M. and Staub, E.: 2005, The SYSTERS Protein Family Database in 2005., *Nucleic Acids Res* 33, D226–D229.

Merkeev, I. V., Novichkov, P. S. and Mironov, A. A.: 2006, PHOG: a database of supergenomes built from proteome complements., *BMC Evol Biol* 6, 52.

Michener, C. D. and Sokal, R. R.: 1957, A quantitative approach to a problem in classification., *Evolution* 11, 130–162.

Miller, D. J., Wang, Y. and Kesidis, G.: 2008, Emergent unsupervised clustering paradigms with potential application to bioinformatics., *Front Biosci* 13, 677–690.

Murtagh, F.: 1985, *Multidimensional clustering algorithms.*, Vol. 4 of *COMPSTAT Lectures*, Physica-Verlag, Vienna.

Natale, D. A., Arighi, C. N., Barker, W. C., Blake, J., Chang, T. C., Hu, Z., Liu, H., Smith, B. and Wu, C. H.: 2007, Framework for a protein ontology., *BMC Bioinformatics* 8 Suppl 9, S1.

Needleman, S. B. and Wunsch, C. D.: 1970, A general method applicable to the search for similarities in the amino acid sequence of two proteins., *J Mol Biol* 48, 443–453.

Neerincx, P. B. T. and Leunissen, J. A. M.: 2005, Evolution of web services in bioinformatics., *Brief Bioinform* 6, 178–188.

Nei, M., Rogozin, I. B. and Piontkivska, H.: 2000, Purifying selection and birth-and-death evolution in the ubiquitin gene family., *Proc Natl Acad Sci U S A* 97, 10866–10871.

Nei, M. and Rooney, A. P.: 2005, Concerted and birth-and-death evolution of multigene families., *Annu Rev Genet* 39, 121–152.

Nikolski, M. and Sherman, D. J.: 2007, Family relationships: should consensus reign?–consensus clustering for protein families., *Bioinformatics* 23, e71–e76.

Nobeli, I., Favia, A. D. and Thornton, J. M.: 2009, Protein promiscuity and its implications for biotechnology., *Nat Biotechnol* 27, 157–167.

Noble, W. S., Kuang, R., Leslie, C. and Weston, J.: 2005, Identifying remote protein homologs by network propagation., *FEBS J* 272, 5119–5128.

Notredame, C.: 2007, Recent evolutions of multiple sequence alignment algorithms., *PLoS Comput Biol* 3, e123.

O'Brien, K. P., Remm, M. and Sonnhammer, E. L.: 2005, Inparanoid: a comprehensive database of eukaryotic orthologs., *Nucleic Acids Res* 33, D476–D480.

O'Brien, K. P., Westerlund, I. and Sonnhammer, E. L.: 2004, OrthoDisease: a database of human disease orthologs., *Hum Mutat* 24, 112–119.

Ohno, S.: 1999, Gene duplication and the uniqueness of vertebrate genomes circa 1970-1999., *Semin Cell Dev Biol* 10, 517–522.

Ohta, T.: 1992, Theoretical study of near neutrality. II. Effect of subdivided population structure with local extinction and recolonization., *Genetics* 130, 917–923.

Okuda, Y., Sasaki, D., Nogami, S., Kaneko, Y., Ohya, Y. and Anraku, Y.: 2003, Occurrence, horizontal transfer and degeneration of VDE intein family in Saccharomycete yeasts., *Yeast* 20, 563–573.

Orengo, C. A. and Thornton, J. M.: 2005, Protein families and their evolution-a structural perspective., *Annu Rev Biochem* 74, 867–900.

Ouzounis, C. A., Coulson, R. M., Enright, A. J., Kunin, V. and Pereira-Leal, J. B.: 2003, Classification schemes for protein structure and function., *Nat Rev Genet* 4, 508–519.

Overbeek, R., Fonstein, M., D'Souza, M., Pusch, G. D. and Maltsev, N.: 1999, The use of gene clusters to infer functional coupling., *Proc Natl Acad Sci U S A* 96, 2896–2901.

Paccanaro, A., Casbon, J. A. and Saqi, M. A. S.: 2006, Spectral clustering of protein sequences., *Nucleic Acids Res* 34, 1571–1580.

Page, R. D. and Charleston, M. A.: 1997, From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem., *Mol Phylogenet Evol* 7, 231–240.

Paget, M. S. B. and Helmann, J. D.: 2003, The sigma70 family of sigma factors., *Genome Biol* 4, 203.

Park, J. and Teichmann, S. A.: 1998, DIVCLUS: an automatic method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins., *Bioinformatics* 14, 144–150.

Passarge, E., Horsthemke, B. and Farber, R. A.: 1999, Incorrect use of the term synteny., *Nat Genet* 23, 387.

Patthy, L.: 1996, Exon shuffling and other ways of module exchange., *Matrix Biol* 15, 301–310.

Pearson, W. R. and Lipman, D. J.: 1988, Improved tools for biological sequence comparison., *Proc Natl Acad Sci U S A* 85, 2444–2448.

Pearson, W. R. and Sierk, M. L.: 2005, The limits of protein sequence comparison?, *Curr Opin Struct Biol* 15, 254–260.

Penkett, C. J., Morris, J. A., Wood, V. and Bähler, J.: 2006, YOGY: a web-based, integrated database to retrieve protein orthologs and associated Gene Ontology terms., *Nucleic Acids Res* 34, W330–W334.

Perrière, G., Duret, L. and Gouy, M.: 2000, HOBACGEN: database system for comparative genomics in bacteria., *Genome Res* 10, 379–385.

Petryszak, R., Kretschmann, E., Wieser, D. and Apweiler, R.: 2005, The predictive power of the CluSTr database., *Bioinformatics* 21, 3604–3609.

Phatsara, C., Jennen, D. G. J., Ponsuksili, S., Murani, E., Tesfaye, D., Schellander, K. and Wimmers, K.: 2007, Molecular genetic analysis of porcine mannose-binding lectin genes, MBL1 and MBL2, and their association with complement activity., *Int J Immunogenet* 34, 55–63.

Pipenbacher, P., Schliep, A., Schneckener, S., Schönhuth, A., Schomburg, D. and Schrader, R.: 2002, ProClust: improved clustering of protein sequences with an extended graph-based approach., *Bioinformatics* 18 Suppl 2, S182–S191.

Pruitt, K. D., Tatusova, T. and Maglott, D. R.: 2007, NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins., *Nucleic Acids Res* 35, D61–D65.

Rahman, S. A., Bakar, A. A. and Hussein, Z. A. M.: 2008, *A review on protein sequence clustering research.*, Vol. 21 of *IFMBE Proceedings*, Springer Berlin Heidelberg.

Rand, W. M.: 1971, Objective criteria for the evaluation of clustering methods., *J Am Stat Assoc* 66, 846–850.

Rattei, T., Arnold, R., Tischler, P., Lindner, D., Stümpflen, V. and Mewes, H. W.: 2006, SIMAP: the similarity matrix of proteins., *Nucleic Acids Res* 34, D252–D256.

Remm, M., Storm, C. E. and Sonnhammer, E. L.: 2001, Automatic clustering of orthologs and in-paralogs from pairwise species comparisons., *J Mol Biol* 314, 1041–1052.

Richardson, J. S.: 1981, The anatomy and taxonomy of protein structure., *Adv Protein Chem* 34, 167–339.

Rivera, M. C. and Lake, J. A.: 2004, The ring of life provides evidence for a genome fusion origin of eukaryotes., *Nature* 431, 152–155.

Rivest, R.: 1992, The MD4 Message-Digest Algorithm., *Technical report*, MIT, Cambridge (MA), United States.

Rocha, E. P. C.: 2006, The quest for the universals of protein evolution., *Trends Genet* 22, 412–416.

Rodriguez, R., Chinea, G., Lopez, N., Pons, T. and Vriend, G.: 1998, Homology modeling, model and software evaluation: three related resources., *Bioinformatics* 14, 523–528.

Rognes, T. and Seeberg, E.: 2000, Six-fold speed-up of Smith-Waterman sequence database searches using parallel processing on common microprocessors., *Bioinformatics* 16, 699–706.

Rost, B.: 1999, Twilight zone of protein sequence alignments., *Protein Eng* 12, 85–94.

Roth, A. C. J., Gonnet, G. H. and Dessimoz, C.: 2008, Algorithm of OMA for large-scale orthology inference., *BMC Bioinformatics* 9, 518.

Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M. and Mewes, H. W.: 2004, The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes., *Nucleic Acids Res* 32, 5539–5545.

Saitou, N. and Nei, M.: 1987, The neighbor-joining method: a new method for reconstructing phylogenetic trees., *Mol Biol Evol* 4, 406–425.

Sander, C. and Schneider, R.: 1991, Database of homology-derived protein structures and the structural meaning of sequence alignment., *Proteins* 9, 56–68.

Sankoff, D.: 1975, Minimal mutation trees of sequences., *SIAM J Appl Math* 28, 35–42.

Sarich, V. M. and Wilson, A. C.: 1967, Immunological time scale for hominid evolution., *Science* 158, 1200–1203.

Sastry, R., Wang, J. S., Brown, D. C., Ezekowitz, R. A., Tauber, A. I. and Sastry, K. N.: 1995, Characterization of murine mannose-binding protein genes Mbl1 and Mbl2 reveals features common to other collectin genes., *Mamm Genome* 6, 103–110.

Scannell, D. R., Byrne, K. P., Gordon, J. L., Wong, S. and Wolfe, K. H.: 2006, Multiple rounds of speciation associated with reciprocal gene loss in polyploid yeasts., *Nature* 440, 341–345.

Scannell, D. R., Frank, A. C., Conant, G. C., Byrne, K. P., Woolfit, M. and Wolfe, K. H.: 2007, Independent sorting-out of thousands of duplicated gene pairs in two yeast species descended from a whole-genome duplication., *Proc Natl Acad Sci U S A* 104, 8397–8402.

Sharan, R. and Ideker, T.: 2006, Modeling cellular machinery through biological network comparison., *Nat Biotechnol* 24, 427–433.

Sharan, R., Ulitsky, I. and Shamir, R.: 2007, Network-based prediction of protein function., *Mol Syst Biol* 3, 88.

Shi, J. and Malik, J.: 2000, Normalized cuts and image segmentation., *IEEE Trans Pattern Anal Mach Intell* 22, 888–905.

Sibson, R.: 1973, SLINK: An optimally efficient algorithm for for the single-link clustering method, *Comput J* 16, 30–34.

Simillion, C., Vandepoele, K. and de Peer, Y. V.: 2004, Recent developments in computational approaches for uncovering genomic homology., *Bioessays* 26, 1225–1235.

Simon, G., Paladini, R., Tisminetzky, S., Cserzö, M., Hátsági, Z., Tossi, A. and Pongor, S.: 1992, Improved detection of homology in distantly related proteins: similarityof adducin with actin-binding proteins., *Protein Seq Data Anal* 5, 39–42.

Sjölander, K.: 2004, Phylogenomic inference of protein molecular function: advances and challenges., *Bioinformatics* 20, 170–179.

Smith, T. F. and Waterman, M. S.: 1981, Identification of common molecular subsequences., *J Mol Biol* 147, 195–197.

Snel, B., Bork, P. and Huynen, M. A.: 2002, The identification of functional modules from the genomic association of genes., *Proc Natl Acad Sci U S A* 99, 5890–5895.

Sonego, P., Kocsor, A. and Pongor, S.: 2008, ROC analysis: applications to the classification of biological sequencesand 3D structures., *Brief Bioinform* 9, 198–209.

Sonego, P., Pacurar, M., Dhir, S., Kertész-Farkas, A., Kocsor, A., Gáspári, Z., Leunissen, J. A. M. and Pongor, S.: 2007, A Protein Classification Benchmark collection for machine learning., *Nucleic Acids Res* 35, D232–D236.

Sonnenberg, A., Rojas, A. M. and de Pereda, J. M.: 2007, The structure of a tandem pair of spectrin repeats of plectin reveals a modular organization of the plakin domain., *J Mol Biol* 368, 1379–1391.

Sonnhammer, E. L. and Kahn, D.: 1994, Modular arrangement of proteins as inferred from analysis of homology., *Protein Sci* 3, 482–492.

Sterk, P., Kulikova, T., Kersey, P. and Apweiler, R.: 2007, The EMBL Nucleotide Sequence and Genome Reviews Databases., *Methods Mol Biol* 406, 1–21.

Storm, C. E. and Sonnhammer, E. L.: 2002, Automated ortholog inference from phylogenetic trees and calculation of orthology reliability., *Bioinformatics* 18, 92–99.

Storm, C. E. and Sonnhammer, E. L.: 2003, Comprehensive analysis of orthologous protein domains using the HOPS database., *Genome Res* 13, 2353–2362.

Sugawara, H., Ikeo, K., Fukuchi, S., Gojobori, T. and Tateno, Y.: 2009, DDBJ dealing with mass data produced by the second generation sequencer., *Nucleic Acids Res* 37, D16–D18.

Sumiyama, K., Saitou, N. and Ueda, S.: 2002, Adaptive evolution of the IgA hinge region in primates., *Mol Biol Evol* 19, 1093–1099.

Sundin, G. W.: 2007, Genomic insights into the contribution of phytopathogenic bacterial plasmids to the evolutionary history of their hosts., *Annu Rev Phytopathol* 45, 129–151.

Suzek, B. E., Huang, H., McGarvey, P., Mazumderand, R. and Wu, C. H.: 2007, UniRef: comprehensive and non-redundant UniProt reference clusters., *Bioinformatics* 23, 1282–1288.

Szalkowski, A., Ledergerber, C., Krähenbühl, P. and Dessimoz, C.: 2008, SWPS3 - fast multi-threaded vectorized Smith-Waterman for IBM Cell/B.E. and x86/SSE2., *BMC Res Notes* 1, 107.

Tarjan, R. E.: 1975, Efficiency of a Good But Not Linear Set Union Algorithm, *J ACM* 22, 215–225.

Tarjan, R. E. and van Leeuwen, J.: 1984, Worst-case Analysis of Set Union Algorithms, *J ACM* 31, 245–281.

Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J. and Natale, D. A.: 2003, The COG database: an updated version includes eukaryotes., *BMC Bioinformatics* 4, 41.

Tatusov, R. L., Koonin, E. V. and Lipman, D. J.: 1997, A genomic perspective on protein families., *Science* 278, 631–637.

Tetko, I. V., Facius, A., Ruepp, A. and Mewes, H. W.: 2005, Super paramagnetic clustering of protein sequences., *BMC Bioinformatics* 6, 82.

Thornton, J. M., Orengo, C. A., Todd, A. E. and Pearl, F. M.: 1999, Protein folds, functions and evolution., *J Mol Biol* 293, 333–342.

Thornton, J. W. and DeSalle, R.: 2000, Gene family evolution and homology: genomics meets phylogenetics., *Annu Rev Genomics Hum Genet* 1, 41–73.

Todd, A. E., Orengo, C. A. and Thornton, J. M.: 2001, Evolution of function in protein superfamilies, from a structural perspective., *J Mol Biol* 307, 1113–1143.

Tordai, H., Nagy, A., Farkas, K., Bányai, L. and Patthy, L.: 2005, Modules, multidomain proteins and organismic complexity., *FEBS J* 272, 5064–5078.

Uchiyama, I.: 2006, Hierarchical clustering algorithm for comprehensive orthologous-domain classificationin multiple genomes., *Nucleic Acids Res* 34, 647–658.

Valencia, A., Kjeldgaard, M., Pai, E. F. and Sander, C.: 1991, GTPase domains of ras p21 oncogene protein and elongation factorTu: analysis of three-dimensional structures, sequence families, and functional sites., *Proc Natl Acad Sci U S A* 88, 5443–5447.

van der Heijden, R. T. J. M., Snel, B., van Noort, V. and Huynen, M. A.: 2007, Orthology prediction at scalable resolution by phylogenetic tree analysis., *BMC Bioinformatics* 8, 83.

van Dongen, S.: 2000, *Graph Clustering by Flow Simulation.*, PhD thesis, University of Utrecht.

van Rijsbergen, C. J.: 1979, *Information retrieval.*, Butterworths, London, UK.

Vilella, A. J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R. and Birney, E.: 2009, EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates., *Genome Res* 19, 327–335.

Vinga, S. and Almeida, J.: 2003, Alignment-free sequence comparison - a review., *Bioinformatics* 19, 513–523.

Vision, T. J., Brown, D. G. and Tanksley, S. D.: 2000, The origins of genomic duplications in Arabidopsis., *Science* 290, 2114–2117.

Vogel, C., Teichmann, S. A. and Pereira-Leal, J.: 2005, The relationship between domain duplication and recombination., *J Mol Biol* 346, 355–365.

Voorhees, E. M.: 1986, The efficiency of inverted index and cluster searches., *SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, New York, USA, pp. 164–174.

Wall, D. P., Fraser, H. B. and Hirsh, A. E.: 2003, Detecting putative orthologs., *Bioinformatics* 19, 1710–1711.

Wall, P. K., Leebens-Mack, J., Müller, K. F., Field, D., Altman, N. S. and dePamphilis, C. W.: 2008, PlantTribes: a gene and gene family resource for comparative genomics in plants., *Nucleic Acids Res* 36, D970–D976.

Wallace, I. M., O'Sullivan, O. and Higgins, D. G.: 2005, Evaluation of iterative alignment algorithms for multiple alignment., *Bioinformatics* 21, 1408–1414.

Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., Kingsmore, S. F., Schroth, G. P. and Burge, C. B.: 2008, Alternative isoform regulation in human tissue transcriptomes., *Nature* 456, 470–476.

Wetlaufer, D. B.: 1973, Nucleation, rapid folding, and globular intrachain regions in proteins., *Proc Natl Acad Sci U S A* 70, 697–701.

Wheeler, D. L., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., Dicuccio, M., Edgar, R., Federhen, S., Feolo, M., Geer, L. Y., Helmberg, W., Kapustin, Y., Khovayko, O., Landsman, D., Lipman, D. J., Madden, T. L., Maglott, D. R., Miller, V., Ostell, J., Pruitt, K. D., Schuler, G. D., Shumway, M., Sequeira, E., Sherry, S. T., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusov, R. L., Tatusova, T. A., Wagner, L. and Yaschenko, E.: 2008, Database resources of the National Center for Biotechnology Information., *Nucleic Acids Res* 36, D13–D21.

Wittkop, T., Baumbach, J., Lobo, F. P. and Rahmann, S.: 2007, Large scale clustering of protein sequences with FORCE -A layout based heuristic for weighted cluster editing., *BMC Bioinformatics* 8, 396.

Wolfe, K. H. and Shields, D. C.: 1997, Molecular evidence for an ancient duplication of the entire yeast genome., *Nature* 387, 708–713.

Wong, S. and Ragan, M. A.: 2008, MACHOS: Markov clusters of homologous subsequences., *Bioinformatics* 24, i77–i85.

Wozniak, A.: 1997, Using video-oriented instructions to speed up sequence comparison., *Comput Appl Biosci* 13, 145–150.

Wu, C. H., Nikolskaya, A., Huang, H., Yeh, L. S., Natale, D. A., Vinayaka, C. R., Hu, Z. Z., Mazumder, R., Kumar, S., Kourtesis, P., Ledley, R. S., Suzek, B. E., Arminski, L., Chen, Y., Zhang, J., Cardenas, J. L., Chung, S., Castro-Alvear, J., Dinkov, G. and Barker, W. C.: 2004, PIRSF: family classification system at the Protein Information Resource., *Nucleic Acids Res* 32, D112–D114.

Yang, F., Zhu, Q., Tang, D. and Zhao, M.: 2009, Using Affinity Propagation Combined Post-processing to Cluster Protein Sequences., *Protein Pept Lett* . Epub ahead of print.

Yang, K. and Zhang, L.: 2008, Performance comparison of gene family clustering methods with expert curated gene family data set in Arabidopsis thaliana., *Planta* 228, 439–447.

Yeats, C., Lees, J., Reid, A., Kellam, P., Martin, N., Liu, X. and Orengo, C.: 2008, Gene3D: comprehensive structural and functional annotation of genomes., *Nucleic Acids Res* 36, D414–D418.

Yoder, A. D. and Yang, Z.: 2000, Estimation of primate speciation dates using local molecular clocks., *Mol Biol Evol* 17, 1081–1090.

Yokoyama, S. and Yokoyama, R.: 1989, Molecular evolution of human visual pigment genes., *Mol Biol Evol* 6, 186–197.

Yona, G., Linial, N. and Linial, M.: 1999, ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space., *Proteins* 37, 360–378.

Yooseph, S., Sutton, G., Rusch, D. B., Halpern, A. L., Williamson, S. J., Remington, K., Eisen, J. A., Heidelberg, K. B., Manning, G., Li, W., Jaroszewski, L., Cieplak, P., Miller, C. S., Li, H., Mashiyama, S. T., Joachimiak, M. P., van Belle, C., Chandonia, J. M., Soergel, D. A., Zhai, Y., Natarajan, K., Lee, S., Raphael, B. J., Bafna, V., Friedman, R., Brenner, S. E., Godzik, A., Eisenberg, D., Dixon, J. E., Taylor, S. S., Strausberg, R. L., Frazier, M. and Venter, J. C.: 2007, The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families., *PLoS Biol* 5, e16.

Zaslavsky, E. and Singh, M.: 2006, A combinatorial optimization approach for diverse motif finding applications., *Algorithms Mol Biol* 1, 13.

Zhang, J.: 2003, Evolution by gene duplication: an update., *Trends Ecol Evol* 18, 292–298.

Zhang, J., Rosenberg, H. F. and Nei, M.: 1998, Positive Darwinian selection after gene duplication in primate ribonuclease genes., *Proc Natl Acad Sci U S A* 95, 3708–3713.

Zhang, X. and Firestein, S.: 2002, The olfactory receptor gene superfamily of the mouse., *Nat Neurosci* 5, 124–133.

Zmasek, C. M. and Eddy, S. R.: 2001, A simple algorithm to infer gene duplication and speciation events on a gene tree., *Bioinformatics* 17, 821–828.

Zmasek, C. M. and Eddy, S. R.: 2002, RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs., *BMC Bioinformatics* 3, 14.

Zuckerkandl, E. and Pauling, L. B.: 1962, *Molecular disease, evolution, and genetic heterogeneity*, Academic Press, New York, USA, pp. 189–225.

GLOSSARY

Conserved gene neighborhood (CGN): refers to conserved genomic segments containing orthologous genes in a similar collinear order between species. Sometimes, the term conserved synteny is used instead, which originally denoted gene loci on the same chromosome regardless of whether or not they are genetically linked. Respecting the original definition of 'synteny' and its etymology, we therefore use the term 'conserved gene neighborhood' (Passarge et al., 1999).

Co-orthologs: two or more sequences in one lineage that are collectively orthologous to one or more sequences in another lineage owing to a lineage-specic duplication(s).

Homology: refers to a testable hypothesis that characters in different species sharing significant sequence similarity (at least 30–35% as a rule of thumb for protein sequences) descend from a single common ancestral character. Sequences that are evolutionarily related to each other in this way are known as homologs. Note that homology is independent of the size and molecular nature of a biological sequence.

Horizontal gene transfer (HGT): an evolutionary process that involves transfer of genetic material between species but does not follow the vertical descent from a parental lineage to its offspring. HGT is an important phenomenon in the evolution of prokaryotes and eukaryotes (Koonin et al., 2001; Lerat et al., 2005; Loftus et al., 2005).

In-paralogs: paralogs that result from a lineage-specic duplication(s) subsequent to a given speciation event (sometimes termed 'recent' paralogs). They are likely to have retained similar functions within a species. Non-transitivity of phylogenetic relationships: orthology, paralogy and xenology are strictly pairwise and non-transitive relationships between (groups of) genes. This can best be understood using the following example: if two genes, a and b, are equally (co-)orthologous to gene c, it does not imply that a and b must also be orthologous to each other (Fitch, 2000). Therefore, an OG must always be hierarchical and dened with respect to the last common ancestor of the investigated genes (taxonomic position).

Orthologous group of genes (OG): a collection of homologous genes from at least two species. After a duplication event, an OG might group paralogs and orthologs together. Therefore, an OG must be dened within a phylogenetic tree in the context of speciation and duplication events to guarantee the non-transitivity of phylogenetic relationships. If an OG consists of single-copy

orthologous genes, then all of the genes can be grouped together because the phylogenetic relationships between all of them are equivalent.

Orthologs: homologous sequences derived by a speciation event from a single ancestral sequence in the last common ancestor of the species being compared. Orthologs typically perform equivalent functions in closely related species.

Out-paralogs: paralogs resulting from a duplication(s) preceding a given speciation event (sometimes termed 'ancient' paralogs). They are likely to have different functions.

Paralogs: homologous sequences derived by a duplication event from a single sequence. Paralogous relationships occur both within and between genomes. Paralogs can evolve novel functions and are likely to have mechanistically distinct but biologically related functions.

Subtree-neighbors: homologs in a rooted gene tree that are found at a particular level (parent node) of the tree (Zmasek and Eddy, 2002).

Super-orthologs: a subset of orthologs selected on a rooted gene tree such that only speciation events are assigned to each internal node on their connecting path (Zmasek and Eddy, 2002).

Ultra-paralogs: a subset of paralogs selected on a rooted tree such that its internal nodes connecting them represent only duplication events (in-paralogs) (Zmasek and Eddy, 2002).

Xenologs: homologous sequences, the history of which involves transfer of genetic information between species (see horizontal gene transfer or HGT). They often appear as true orthologs in genome comparisons and might exhibit variable functions (Okuda et al., 2003).

ACKNOWLEDGEMENTS

When I visited the Netherlands for the very first time, in the year 2002, I could never have imagined that Wageningen would become my second home in the coming years. I'm still wondering how all this have happened. One thing I know for sure: on this journey I have met many people who have helped me (directly or indirectly) to find my way and to whom I would like to express my gratitude.

First of all, I would like to thank to my daily supervisor and promotor, Prof. Jack Leunissen, for giving me the opportunity to pursue a PhD study in bioinformatics. During the years he provided me the freedom to explore and guidance I needed. From him I learned to appreciate simplicity before moving into complexity. I appreciate your hands-on assistance in my coding activities; for example, I still recall your genuine solutions – changing a zero to one or *vice versa* – that did the trick. Also thank you for (re)writing my *Samenvatting*. Second, I'm grateful to my co-promotor, Dr. Roeland van Ham, for teaching me to write succinctly and precisely, and to endure (by now should know the meaning of the Dutch word "polderen"). Thank you for being patient with me. Further, I feel very lucky to have a supervisor such as Prof. Sándor Pongor. You have been a very inspiring mentor for me through the years. Although I met you in person only once in Trieste (Italy), through Skype we could brainstorm from time to time, resulting in several publications. Your optimism, humor and enthusiasm helped me to stay on the track, thank you very much.

Then I would like to thank to my (ex-)colleagues and students at the Laboratory of Bioinformatics and Applied Bioinformatics Group (PRI), especially to Pieter Neerincx, for his technical assistance, ticket bookings, exhaustive explanations, and great company at AsterBIOS. Harm Nijveen, thank you for joining me for jogging (the Dutch weather is not always friendly in the early mornings so let's Zzz...). I appreciate very much your first-aid attitude in all kinds of matters in our lab. Anand Gavai and Jifeng Tang, thank you for the irregular yet very refreshing sport nights. It was fun to run, swim or play a table tennis with you. Hong Luo and Blaise Alako, I enjoyed your company and our brainstorm sessions. Judith Risse, thank you for helping me in a detective pursuit for my lost backpack, as well as for mobilizing us (guys) to participate in the We Day activities. Ernest van Ophuizen, your stories and quiz questions were always entertaining; now I know whom should I call first when sitting in the Millionaire TV show. Also thank to people from the Laboratory of Molecular biology for nice social events such as lab trips and dinners. Many thanks to our office managers Marie-José van Iersel and Maria Augustijn for their swift and pampering assistance, and to the EPS education coordinator, Dr. Douwe Zuidema, for his guidance and advices.

I would like to thank to Dr. Beata Hasiów-Jaroszewska for initiating a project on comparative genomics plant RNA viruses (our manuscript is now being peer-

reviewed), and to Pieter van Beek for expert help in GRID computing.

Further, I'm grateful to my friends from Wageningen (and surroundings) who accompanied me during this journey. Marco Schreuders, thank you for your friendship and support during the years, as well as being my paranymph at the ceremony. Through your openness and directness I got to know the Dutch culture as well as integrated into the new environment quickly. Many thanks to Robert Onzima, for his friendship and particularly for introducing me into the ecumenical life at Student Chaplaincy Wageningen (SCW). For me and my family it was an unforgettable experience when you visited us in Slovakia at Christmas. Ke Lin, thank you for giving me a hand in my stressful times, sharing your optimism, humor and expertise, and for being my second paranymph. Working with you has always involved a lot of fun and creativity. Sama Iziah, thank you bro for sharing cheerful times during dinners, movie nights, parties, BBQs and carnivals. I also thank to my friends Reza Aghnoum, Nie Haisheng, Ningwen Zhang and Xin-Ying Ren, Tomás Zubcsák, Katarína Belobradová, Luis Jarrin Pasmio, Marieke Bos, Alejandro Flores, Jan-Willem Klaas, Caucasella Diaz Trujillo, Randy Wilbrink for pleasant conversations, dinners and/or parties.

Furthermore, I would like to express my gratitude to people in the SCW community for their friendship and care. Thank you for many uplifting events and trips in which I could experience true unity. Many thanks to Josine van der Horst, Wiel Eggen, Ingeborg Brower (thank you very much also for proofreading some chapters of my thesis), Evert Jansen, Sulaiman Bangura, Piet Oosterom, Yansen Lauw, Mária Szaszkó, Robert Bara, Alexandre Villela, Daniela Barbosa, Susanne Bellmann and to musicians from all over the world, especially Wil Lyklema, Jolande Mol, Alejandro Flores, Kingsley Asare, Ged Guinto, Joebel Gurang and Simone Albrecht. I would also like to thank to Latino musicians from Rijsteeg and Bornsesteeg, especially to Walter Castro for great parties and beautiful Andean melodies (maybe it's already time to resume the "One day music band" project...).

Learning the Dutch languange next to science is not easy for a foreign PhD student, particularly when it comes to a self-study. I would like to thank to my teachers at 't Venster for being patient, my classmates namely Anand Gavai, Artak Ghandilyan, Greer Wilson and Ningwen Zang for the 'low-profile' and cheerful lessons in the late evenings, as well as to Dutch friends, 'biologen', for their openness and help in practicing the languange.

Dear Sabina, Michael and Mark Król, thank you so much for providing me the means to experience life beyond Slovak borders, as well as for wonderful times we have spent together.

Finally, the deepest gratitude to my parents, Jozef and Helena Kuźniar, for their endless love, support and encouragement, my brother, Marian Kuźniar, my godparents, Kamocsai László and Sarolta, for being there for me whenever I needed. Despite being far away from home you all have always been in my heart. Brenda van der Zee, my loving girlfriend, I'm a very lucky man to meet someone like you. I'm thankful to you for your care, patient and understanding; you gave me the wings to fly. Many thanks also to your parents, Cees and Janny van der Zee, for their openness and support.

LIST OF PUBLICATIONS

Kuzniar, A., van Ham, R. C. H. J., Pongor, S. and Leunissen, J. A. M.: 2008, The quest for orthologs: finding the corresponding gene across genomes., *Trends Genet* 24, 539–551.

Kuzniar, A., Lin, K., He, Y., Nijveen, H., Pongor, S. and Leunissen, J. A. M.: 2009, ProGMap: an integrated annotation resource for protein orthology., *Nucleic Acids Res* 37, W428–W434.

Kertész-Farkas, A., Dhir, S., Sonego, P., Pacurar, M., Netoteia, S., Nijveen, H., **Kuzniar**, A., Leunissen, J. A. M., Kocsor, A. and Pongor, S.: 2008, Benchmarking protein classification algorithms via supervised cross-validation., *J Biochem Biophys Methods* 70, 1215–1223.

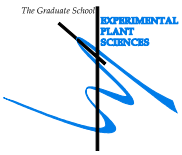
Kuzniar, A. Dhir, S. Nijveen, H., Pongor, S., Leunissen, J. A. M.: 2009, Multinetclust: A tool for finding connected clusters in multi-parametric data-networks. *Submitted*.

Hasiów-Jaroszewska, B., **Kuzniar**, A. , Leunissen, J. A. M., Pospieszny, H.: 2009, Evidence for RNA recombination between distinct isolates of Pepino mosaic virus. *Submitted*.

CURRICULUM VITAE

Arnold Kuźniar was born on the 9th of February 1981 in Komárno, Slovakia. He grew up in a small city, Kolárovo (Gúta), near the Slovak-Hungarian borders. After he graduated from Gymnasium Ľ. J. Šuleka Komárno he moved to the capital, Bratislava, to study biology at the Faculty of Natural Sciences of Comenius University (PriF UK). In the last years of the MSc programme he visited the Netherlands several times to participate in workshops on the safety of genetically modified plants. In 2003 he was granted a Socrates Erasmus scholarship to study abroad for a period of six months. He decided to go to the Netherlands and followed courses within the biotechnology study programme at Wageningen University. Besides the courses he worked on his master thesis at Laboratory of Virology and Applied Bioinformatics Group where he could combine his interest in molecular evolution and bioinformatics. After completing the exams in Wageningen he returned back home and graduated successfully from PriF UK with the master's degree in molecular biology. In the same year he was offered a position at the Laboratory of Bioinformatics of Wageningen University to pursue a PhD in comparative genomics. The results of this 5 years commitment are presented in this thesis.

Education Statement of the Graduate School
Experimental Plant Sciences



Issued to: Arnold Kuzniar
Date: 6 November 2009
Group: Laboratory of Bioinformatics, Wageningen University

1) Start-up phase	date
► **First presentation of your project**	
Comparative genomics of fungi	Oct 22, 2004
► **Writing or rewriting a project proposal**	
Genome-wide orthology prediction	Sep 15, 2004
► **Writing a review or book chapter**	
The quest for orthologs: finding the corresponding gene across genomes, Trend in Genetics 24, p 539-551 (2008)	Oct 2006-2008
► **MSc courses**	
► **Laboratory use of isotopes**	
Subtotal Start-up Phase	*13.5 credits**

2) Scientific Exposure	date
► **EPS PhD student days**	
EPS PhD Student Day, Radboud University Nijmegen	Jun 02, 2005
SVD PhD student day 2006, Paris, France	Jun 09, 2006
EPS PhD Student Day, Wageningen University	Sep 19, 2006
EPS PhD Student Day, Wageningen University	Sep 13, 2007
► **EPS theme symposia**	
EPS Theme 4 symposium 'Genome Plasticity', Wageningen University	Dec 09, 2005
EPS Theme 4 symposium 'Genome Plasticity', Radboud University Nijmegen	Dec 08, 2006
EPS Theme 4 symposium 'Genome Plasticity', Leiden University	Dec 07, 2007
► **NWO Lunteren days and other National Platforms**	
NBIC Netherlands Conference on Bioinformatics, Groningen	Oct 07-08, 2004
NBIC Netherlands BioInformatics Conference, Ede	Apr 24, 2006
NBIC: BioRange Consortium Meeting	Mar 05-06, 2008
► **Seminars (series), workshops and symposia**	
Geneyous Symposium, Oss, NL	Apr 27, 2005
EPS Flying seminars (3x)	2004-2005
Minisymposium CBS/WUR-Phtopathology, Utrecht	Jun 17, 2005
Seminars (series) PSG Joint Bioinformatics Meetings	Jan-Jul, 2006
SIM workshop on "Statistics for Biological Networks", Eindhoven	Jan 16-18, 2006
"Grid tutorial 2007", Utrecht	Sep 24, 2007
Prof. Jian-Kang Zhu "DNA methylation in Arabidopsis"	Nov 03, 2008
Dr. Pamela Hines "Science from an Editor's View"	Nov 06, 2008
► **Seminar plus**	
► **International symposia and congresses**	
Benelux Bioinformatics Conference (BBC2005), Ghent, Belgium	Apr 14-15, 2005
ESF-EMBO Symp. on Comparative Genomics of Eukaryotic Microorganisms, Girona, Spain	Nov 13-17, 2005
Benelux Bioinformatics Conference (BBC2006), Wageningen, The Netherlands	Oct 17-18, 2006
ISMB/ECCB 2007, Vienna, Austria	Jul 21-25, 2007
Benelux Bioinformatics Conference (BBC2007), Leuven, Belgium	Nov 12-13, 2007
► **Presentations**	
Poster: "Groups of Orthologous Genes (GOGs)", ESF-EMBO Symposium, Girona, Spain	Nov 13-17, 2005
Oral: "DNA polymerases of *M. graminicola*" Mycospaherella annotation jamboree at JGI, USA	Jun 07, 2006
Oral: "Large-scale comparison of prokaryotic proteomes", PSG Joint Bioinformatics Meeting	Oct 19, 2006
Oral: PSG Joint Bioinformatics Meeting	Oct 19, 2006
Poster: "ProGMap database", ISMB/ECCB 2007, Vienna, Austria	Jul 21-25, 2007
Poster & application showcase: "ProGMap database", NBIC BioRange Consortium Meeting	Mar 05-06, 2008
► **IAB interview**	Sep 14, 2007
► **Excursions**	
Subtotal Scientific Exposure	*14.4 credits**

3) In-Depth Studies	date
► **EPS courses or other PhD courses**	
PhD cource "Molecular Phylogenies: Reconstruction & Interpretation"	Oct 17-21, 2005
Practical Course "Bioinformatics: Computer Methods in Molecular Biology", ICGEB-Trieste, Italy	Jun 23-30, 2006
► **Journal club**	
Literature discussion together with Applied Bioinformatics Group (PRI)	2004-2008
► **Individual research training**	
Subtotal In-Depth Studies	*6.3 credits**

4) Personal development	date
► **Skill training courses**	
Time planning & project management	Jan 21, 2005
Scientific writing	Feb 25, 2008
Organisation of PhD students day, course or conference	
► **Membership of Board, Committee or PhD council**	
Member of EPS PhD Council	2006
Subtotal Personal Development	*4.0 credits**

TOTAL NUMBER OF CREDIT POINTS*	**38.2**

Herewith the Graduate School declares that the PhD candidate has complied with the educational requirements set by the Educational Committee of EPS which comprises of a minimum total of 30 credits

** A credit represents a normative study load of 28 hours of study*

Thesis layout by
Arnold Kuźniar using LaTeX and the `Memoir` class written by Peter Wilson

Cover design by
Brenda van der Zee
A photo taken from a grape (*Vitis vinifera*) leaf and further edited by the GNU Image Manipulation Program (GIMP 2.6.7)

Printed by
GVO printers & designers B.V., Ponsen & Looijen
Ede, the Netherlands