Faculty of Biological and Environmental Sciences
University of Helsinki

and

Integrative Life Science Doctoral Program (ILS)

COMPARATIVE EVALUATION OF METHODS FOR SEQUENCE ALIGNMENT AND ANNOTATION

Ilya Plyusnin

DOCTORAL DISSERTATION

To be presented, with the permission of the Faculty of Biological and Environmental Sciences of the University of Helsinki, for public examination in Raisio-hall, Metsätieteiden talo, on the 9th of October, 2020 at 12 o'clock.

Helsinki 2020

Supervisor

Professor Liisa Holm
Department of Biosciences & Institute of Biotechnology
University of Helsinki
Finland

Reviewed by

Adjunct Professor Jari Björne
Department of Future Technologies
University of Turku
Finland

and

Associate Professor Leo Lahti
Department of Future Technologies
University of Turku
Finland

Opponent

Professor Christophe Dessimoz
University of Lausanne
Switzerland

Custos

Professor Liisa Holm
Department of Biosciences & Institute of Biotechnology
University of Helsinki
Finland

ISBN 978-951-51-6213-7 (pbk.)
ISBN 978-951-51-6214-4 (PDF)

Unigrafia
Helsinki 2020

“Believe nothing, O monks, merely because you have been told it … or because it is traditional, or because you yourselves have imagined it. Do not believe what your teacher tells you merely out of respect for the teacher. But whatsoever, after due examination and analysis, you find to be conductive to the good … that doctrine believe and cling to and take it as your guide.”

– Siddhārtha Gautama

TABLE OF CONTENTS

ABSTRACT

The speed of DNA and RNA sequencing has long ago surpassed the capacity of laboratories to assign function to these sequences by direct experiment. Fortunately, function and other information can be effectively transferred to novel data from previously accumulated knowledge by sequence homology. This has resulted in the development of hundreds of novel homology-based methods. However, the tendency of method developers to be overoptimistic about their own results, biases in the evaluation metrics used to rank methods, inconsistency between different rankings and evaluation metrics, misplaced popularity of methods relative to their performance all indicate that, in many cases, clear knowledge of the comparative performance of different methods is lacking. This has two main consequences. First, researchers use suboptimal tools. Second, method development may go astray because the merits used for guiding method optimization are biased or unclear. To avoid these difficulties, further research is needed into methodology of evaluation and comparative studies.

One core approach for transferring function by sequence homology is to create a multiple sequence alignment (MSA) that represents a given group of similar sequences. The resulting alignment can be applied to annotate novel sequences using profile hidden Markov models (HMMs), to create phylogenetic trees or to compare structural features. The application of MSAs and profile HMMs for genome annotation was explored in publication (I). Creating MSA has been addressed by a vast field of research, however there is a lack of independent comparative studies and no comparative studies for alignment strategies. In publication (II) a novel modular MSA aligner was implemented to aid in comparative evaluation of different MSA strategies. Different MSA strategies were then compared to each other and to the state-of-the-art MSA software on three benchmark databases.

Another core approach has been to combine homology searches with assignment of annotation terms from a controlled vocabulary such as the Gene Ontology (GO). Hundreds of methods that assign GO terms to novel sequences have been introduced. The research community has also invested into the objective evaluation of these methods via third party competitions. However, the evaluation metrics and merits used in these competitions are still under active debate and need further research and development. In publication (III) a novel framework was introduced for the development of unbiased high-quality evaluation metrics. By testing 37 variations of popular metrics, our approach revealed strong differences between metrics, a list of clearly biased metrics, and a list of high-quality metrics that are well suited for the evaluation of GO annotations.

In summary, this thesis presents novel frameworks and implementation platforms for comparative evaluation of two important classes of homology-based methods: MSA aligners and GO sequence classifiers. These results will be instrumental for developing more accurate MSA aligners, for eliminating many forms of bias inherent in contemporary evaluation protocols, for producing informative method rankings for non-specialist users and for guiding method development towards merits that truly reflect the utility of the designed tools.

LIST OF ORIGINAL PUBLICATIONS

This thesis is based on the following articles, which are referred in the text by their Roman numerals:

I Plyusnin I, Holm L, Kankainen M. LOCP--locating pilus operons in gram-positive bacteria. Bioinformatics. 2009;25:1187–1188.

II Plyusnin I, Holm L. Comprehensive comparison of graph based multiple protein sequence alignment strategies. BMC Bioinformatics. 2012;13:64.

III Plyusnin I, Holm L, Törönen P. Novel comparison of evaluation metrics for gene ontology classifiers reveals drastic performance differences. PLoS Comput. Biol. 2019;15:e1007419.

Author's contributions:

I. Plyusnin Ilja (IP) contributed to conceptualization, data curation and method development. IP's unique contribution was the following: designed and implemented the software, performed formal data analysis, performed comparative evaluation of the results. IP also wrote the first and revised versions of the manuscript.
This paper was previously included in Matti Kankainen's (MK) dissertation. MK's unique contribution was the following: MK proposed the original concept for the study, curated dataset selection and method development, edited the manuscript draft.

II. IP was responsible for conceptualization, data curation, method development, software implementation, formal analysis, validation, visualization and writing of the original manuscript.

III. IP has contributed to conceptualization, data curation and method development. IP implemented the software and performed most of the formal analysis and validation. IP contributed to visualization and writing of the original manuscript.

ABBREVIATIONS

ADS	Artificial Dilution Series
AFP	Automated Function Predictor
APS	Artificial Prediction Set
AUC	Area Under Curve
BAliBASE	Benchmark Alignment database
BLAST	Basic Local Alignment Search Tool
BLOSUM	BLOcks SUbstitution Matrix
EM	Evaluation Metric
FP	False Positives
FPS	False Positive Set
FN	False Negatives
GC	Gene-Centric
GO	Gene Ontology
GTG	Global Trace Graph
HMM	Hidden Markov Model
HOMSTRAD	HOMologous STRucture Alignment Database
MICA	Most Informative Common Ancestor
MMSA	Modular Multiple Sequence Aligner
MSA	Multiple Sequence Alignment
NJ	Neighbor Joining
PAM	Point Accepted Mutation
PFAM	Protein FAMily database
PLDR	Pilus-Like Dense Region
PSI-BLAST	Position Specific Iterative Basic Local Alignment Search Tool
ROC	Receiver Operating Characteristic
SABmark	Sequence Alignment Benchmark
SAM	Sequence Alignment and Modeling system
SCOP	Structural Classification of Proteins
SIM	Similarity Matrix
SP	Sum of Pairs
TC	Term-Centric
TN	True Negatives
TP	True Positives
UPGMA	Unweighted Pair Group Methods with Arithmetic mean
US	UnStructured
WPGMA	Weighted Pair Group Methods with Arithmetic mean

1. INTRODUCTION

High throughput DNA and RNA sequencing has created an information gap between the available sequence data and biological function that is attributed to these sequences. Fortunately, function can be transferred by homology to other sequences that are already described. This has led to the development of hundreds of methods that are rooted in sequence homology [1,2]. Such rapid development has created a challenge for an efficient information transfer within the community of researchers developing and applying these methodologies (see Fig 1).

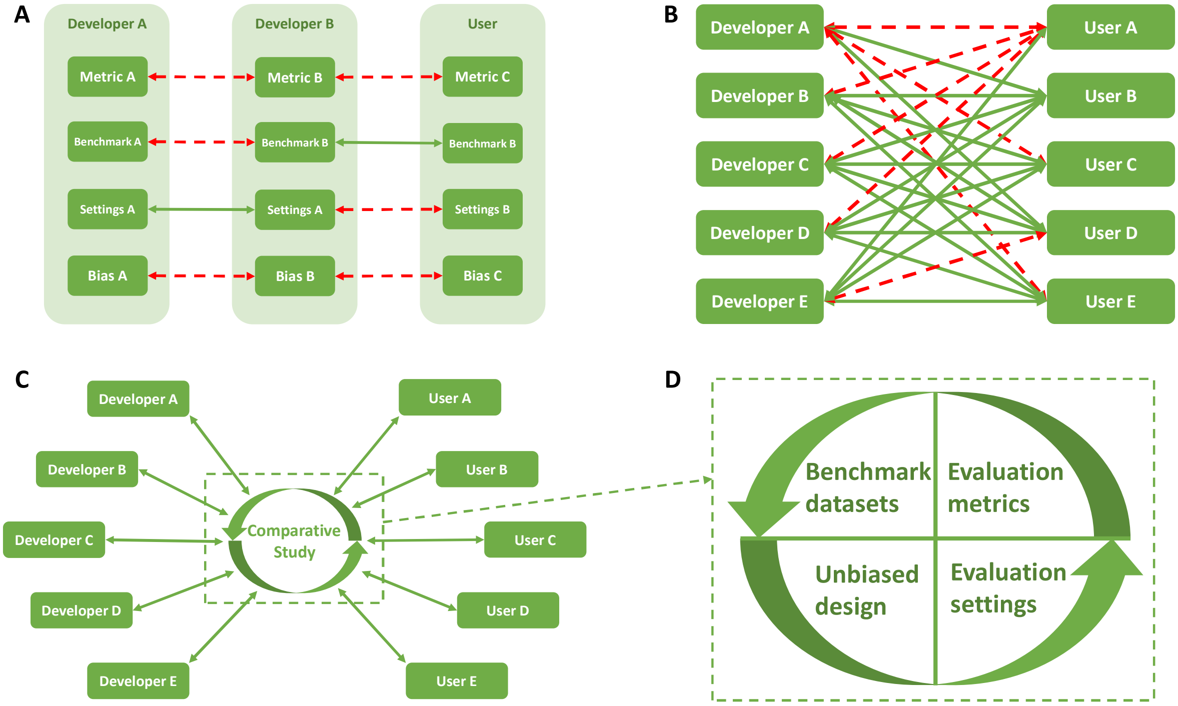


Figure 1 The role of comparative studies in researcher networks.
A) Researcher networks that are limited to method developers and users have inefficient information flow (red dashed arrows) due to inconsistencies between benchmarks, evaluation metrics and evaluation settings applied by different researchers. In the illustrated toy example, Developer A reports performance in metric A on benchmark A, while Developers B reports performance in metric B on benchmark B. This complicates assessment of competitive methods for both developers as well as for the User who is accustomed to metric C and is not familiar with benchmark A. Furthermore, all three researchers are biased in their evaluation: both developers are inclined to promote their own methods while the User is biased towards a popular method C that has a long history of usage. B) The lack of centralized source of information creates an unnecessary load on communication. To keep updated, each user is required to read publications by all developers. Furthermore, in many cases the information transfer from the developer to the user will be inefficient due to the reasons outlined in A. C) Comparative study acts like a network hub processing information received from developers and redistributing it to other developers and users. D) Effective information flow is implemented by defining benchmark datasets, evaluation metrics, evaluation settings and by an unbiased design of the study. Included benchmarks and evaluation metrics should cover most evaluation criteria and user cases that are of importance to developers and users.

The method users may not have the resources and expertise to compare and evaluate available and emerging methodologies [1]. For users it is also difficult to compare different methods due to each author employing different evaluation metrics, benchmark datasets, evaluation settings and formats in their reports [1–3] (Fig 1A). For example, in the field of multiple sequence alignments (MSAs) this has led to the steady popularity of aligners that have long ago been surpassed in terms of accuracy by newer methods [4]. Similar developer-researcher gaps are common in several fields of computational biology [1].

The method developers are also challenged by similar issues. Different authors apply different evaluation metrics on different datasets with different settings resulting in noisy information transfer between developers (Fig 1A). It is also well established that authors experience pressure to report positive findings, which renders self-assessment prone to a number of biases [5–8].

Comparative studies can address most of these challenges. Comparative studies act like network hubs processing information received from developers and redistributing it to other developers and users (Fig 1C). Comparative studies can be divided into four main components (Fig 1D): defining benchmark datasets, evaluation metrics, evaluation settings and choosing a study design to address biases. Common benchmarks, evaluation metrics and evaluation settings define the common reference framework against which the performance of different methods can be mapped. Comparative studies are generally performed by a third-party research group, which can effectively address biases related to self-assessment. Bias can be further reduced by adopting prospective evaluation design, also known as challenges [7], in which evaluation datasets are hidden from all participants, or by applying blinded settings similar to biomedical research [9]. Centralizing evaluation to a single research group that publishes a single report also reduces the amount of communication required between community members (for illustration compare Fig 1B and 1C).

The most credited advantage of comparative studies is their ability to address biases related to self-assessment [2,7]. Thus, in the next section these biases are discussed more in detail.

2. SELF-ASSESSMENT IS PRONE TO BIAS

Publications presenting novel methods are prone to many biases as a result of common practices in method development and policies of scientific journals [5–7,10].

Selective reporting refers to authors' conscious or unconscious decisions to report method variants, parameters, data sets or other evaluation settings that show improvements while leaving negative findings unreported [7]. Selective reporting is similar to *fishing-for-significance* in biomedical and *fishing-for-improvement* in bioinformatics research. Selective reporting can be aggravated by journal policies, when these favor papers demonstrating clear improvements over competing methods [5,7,10]. This phenomenon, known as *publication bias*, is well documented both in biomedical [5] and bioinformatics research [7,10]. For example, in a survey of 57 papers related to computational biology, all novel methods were introduced as the best in most of the assessed metrics and data sets [7]. In this study there was not a single exception to this rule. In another survey conducted on 55 articles (published during 2010-2012 in bioinformatics, computational statistics and machine learning), the method introduced by the author was almost always promoted as a general winner or as a method with important advantages [10].

Systematic bias is introduced when the model is evaluated on a dataset that is not independent of the data set used for estimating model parameters [7]. In this case evaluation will give overoptimistic results, because the testing data is more similar to the training data than datasets in real-life applications. *Model overfitting* refers to a similar scenario when models are well fitted to the training data, but perform poorly on other datasets. Systematic bias and model overfitting can occur in bioinformatics when the available datasets are scarce or undiversified.

2.1 Advantages of comparative studies

Third-party comparative studies can address many problems related to self-reporting and self-evaluation. Comparative studies are likely to be less prone to publication bias, because the authors do not experience pressure to promote any particular method or to publish positive findings. In a literature review by Boulesteix et al., third-party evaluations were shown to identify clear winners less frequently and, in general, to report less drastic differences between the compared methods than self-conducted evaluations [10].

Comparative studies can also address systematic bias, model overfitting, selective reporting, fishing-for-significance and fishing-for-improvements. Recently, several authors have published guidelines for designing and conducting comparative studies aimed at uncovering biases related to self-evaluation and producing consistent method rankings [1–3,8,11]. According to these guidelines comparative studies should be performed on independent benchmark datasets, that are ideally unrelated to the training datasets employed in method development [2,3,8,11]. These guidelines also recommend employing several different datasets and several different evaluation metrics [2,8,11]. Using several independent datasets with several evaluation metrics increases the likelihood of exposing any overfitting present in compared methods and also counteracts selective reporting and fishing-for-improvement [2,8].

In general, independent comparative studies have more resources and motivation to focus on healthy evaluation practices, which results in more consistent, objective and informative rankings of the compared methods [1,2,8]. The importance of these studies is increasingly recognized by the bioinformatics research community. In the last decade a growing number of independent comparative evaluations have been published for homology-based methods [1]. These include genome assembling and characterization, read alignment, protein function prediction, RNA-Seq analysis, variant calling and multiple sequence alignments [1]. Additionally, several authors have focused on implementing software for evaluation [12,13]. Such software can generate key metrics and graphical reports that contribute to defining unbiased, efficient and reproducible evaluation frameworks. As several authors have pointed out, developing comprehensive evaluations requires a significant effort [2,14]. Thus, it is important to develop robust evaluation platforms that can serve as the starting point for novel comparative studies.

Comparative studies can also help to identify similarities between different methods, which creates opportunities for code sharing and the emergence of standardized code libraries (e.g. SeqAn [15]). Standardized code libraries can markedly accelerate further method development, reduce resource drainage due to reimplementation of the same algorithms and are also likely to improve code reliability and maintenance.

Scientific journals are also starting to recognize the importance of comparative studies. For example, PLOS Computational Biology, one of the leading journals in bioinformatics, launched in November 2018 a new article category aimed exclusively for benchmarking. The aim is, as the editorial put it, to elevate comparative benchmarking where it belongs: “the heart of computational biology” [14].

This thesis continues the trend of research in the domain of comparative studies. The scope of the thesis is focused on comparative studies for multiple sequence aligners (MSAs) and automated function predictors (AFPs). Both MSAs and AFPs are based on sequence homology, which is the primal source of information in many, if not the majority, of methods in bioinformatics. Thus, the next sections will focus on introducing the concept of homology and the underlying biological principles.

2.2 Biological basis of sequence homology

2.2.1 Biopolymers and the central dogma of molecular biology

Figure 2 illustrates the central dogma as this was formulated by Francis Crick [16]. Crick postulated that in biological systems information can be transferred (i.e. copied between molecules) from deoxyribonucleic acid (DNA) to ribonucleic acid (RNA) to polypeptides (syn. proteins) [17]. All of these molecules are chains of monomers linked into long polymeric sequences. Monomers composing DNA are called nucleotides, of which there are four different types. Nucleotides are composed of three subunits: a phosphate group, a five carbon sugar and a base molecule, from which only the bases differ between different nucleotides. In DNA there are four different bases: adenine (A), guanine (G), thymine (T) and cytosine (C). The monomers that compose RNA are also nucleotides, however thymine (T) is replaced with uracil (U). Monomers composing the polypeptides are the amino acids, of which there are

20 types in most species. Due to their linear polymeric structures, DNA, RNA and polypeptides can be represented by sequences of symbols which are collectively known as biological sequences. The significance of DNA, RNA and proteins is that these molecules direct the flow of information in biological systems implementing information storage, replication, transmission and translation into function [17].

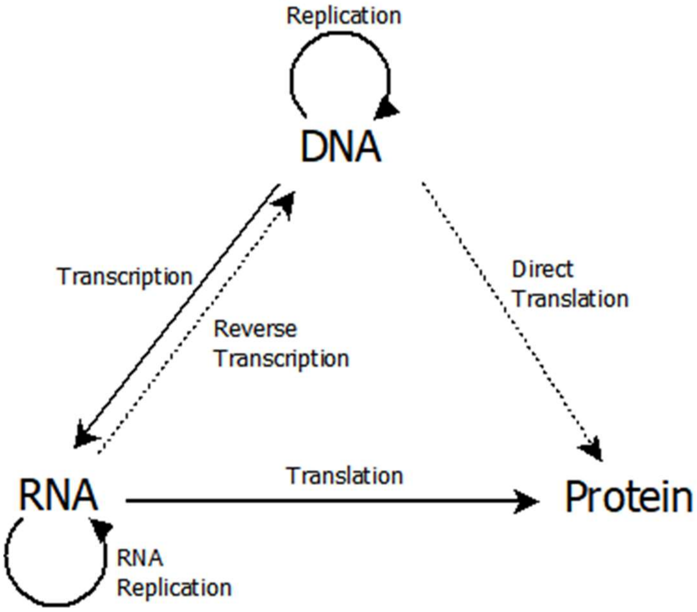


Figure 2 Central dogma of molecular biology. Reproduction of Fig 2 from the famous publication by Francis Crick in 1970 [16]. The arrows indicate the flow of information. The central dogma states that between the classes of primal biopolymers information flows mainly from DNA to RNA to proteins. Crick further formulated that there are three groups of information transfer in biological systems [16]: (I) those for which evidence exists (solid arrows), (II) those for which no evidence existed but that might occur (dashed arrows) and (III) those that are very unlikely to occur (protein to DNA and protein to protein transfers). 50 years later this dogma still holds with evidence that category (I) transfers represent the main flow of information, while reverse translation (e.g. in retroviruses) and direct translations [18] are special cases.

DNA is a double helix structure consisting of two polynucleotide chains running in opposite directions that are held together by hydrogen bonds between opposing base pairs: A pairs to T and G pairs to C. The primary role of DNA is to store and replicate biological information. The double-stranded structure of DNA allows it to replicate precisely by separation of the two strands, followed by synthesis of two new strands, in accordance with the sequence of the original molecule [17] (Fig 3).

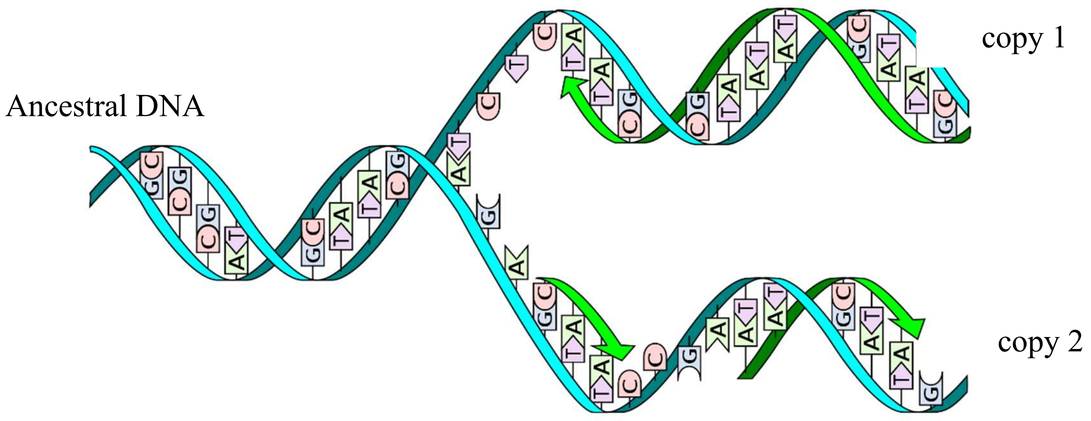


Figure 3 DNA replication generates two copies of the original sequence. Alternations to the original DNA sequence can be introduced by different mutation events. These include substitutions, insertions and deletions. Most of these mutations are introduced during DNA replication. (source: Wikimedia commons, “DNA replication split” by Madprime under CC0 1.0).

RNA contains only a single polynucleotide chain that can be folded onto itself, to another RNA molecule or a single DNA-strand to form base-pairs similar to a DNA double helix. In RNA, thymine is replaced by uracil that can form a base pair to adenine [17]. RNA comes in many forms and has multiple functions. Here we are mainly interested in messenger RNA (mRNA) and its role in DNA transcription and translation.

Transcription is the process of transferring information from DNA to mRNA (Fig 2). During transcription the DNA strands separate and serve as templates for synthesis of complementary strands of mRNAs [17].

During translation mRNA binds to ribosomes and is processed by these structures one triplet of nucleotides (known as *codons*) at a time. Thus, the sequence of amino acids (AAs) in the forming polypeptide is determined by triplet sequences in the mRNA and the triplet to AA correspondence. The correspondence between codons and AAs is known as the *genetic code*. The genetic code is largely conserved across different branches of the Tree of Life (see the next section) allowing a relatively precise automated conversion between DNA, RNA and polypeptide sequences.

Proteins are chains of amino acids connected by peptide bonds and folded on themselves to form various secondary (syn. 2D) and tertiary (syn. 3D) structures [17]. 2D structures refer to local symmetric folding of the polypeptide chain (e.g. α-helixes and β-sheets) while the 3D structure refers to the overall 3D shape of the folded chain. The 3D structure is also known as the *fold* of the protein. Protein *domains* refer to conserved parts of the polypeptide chain and the corresponding 3D folds that can form, function and evolve relatively independently. Complex proteins are formed by several domains often interconnected by linear motifs that have low folding complexity [17]. The significance of proteins is their ability to carry out a multitude of functions specified by the information encoded in DNA and required of the organism by its environment [17].

2.2.2 Evolution and the Tree of Life

The basis of biology lies in the theory of evolution and the assumption that all lifeforms can be traced to a common ancestor [19]. Evolutionary mechanisms operating over thousands or millions of generations create gradual changes in genetic composition of populations which eventually leads to emergence of novel genetic, structural and functional forms. Still, all extant life-forms remain connected by their evolutionary history. This idea was captured by the *Tree of Life* concept first proposed by Charles Darwin in his famous “On the origin of species” [19] and in the 1990’s reinvented by Woese et al [20] (see Fig 4). Based on sequence data, Woese et al. suggested that all extant taxa can be grouped into three domains: *Bacteria*, *Archaea* and *Eukarya* [20]. Although, the placement of the root and relationships between Achaea and Eukarya are still subject to debate [21], the main concept of relatedness of all organisms is now well established [21,22]. More recent and more detailed Trees of Life than that presented in Fig 4 have also been proposed [22,23].

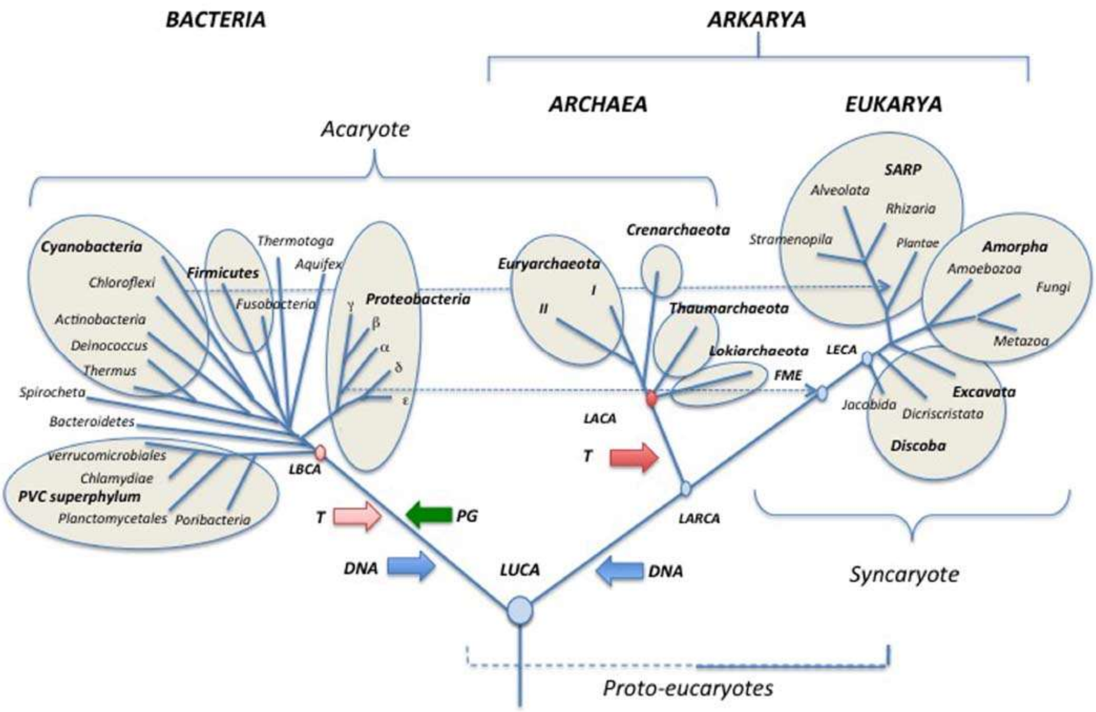


Figure 4 The Tree of Life. The concept of a unifying phylogenetic tree that connects all living organisms was proposed by Charles Darwin in his famous work “On the origin of species” [19]. Contemporary scientists have created several versions of this tree. The consensus tree presented here is the update by Forterre [21] for the three-domain model by Woese et al. [20]. The main domains are *Bacteria*, *Archae* and *Eukarya*, with the latter two grouped by some authors into a monophyletic group *Arkarya*. Common ancestors are traceable for key taxonomic groups: *LECA*, Last Eukaryotic Common Ancestor; *FME*, First Metochondriate Eukarya; *LACA* and *LBCA*, Last Archael and Bacterial Common Ancestor; *LARCA,* Last Arkarya Common Ancestor; All taxonomic groups trace back to the *LUCA* node representing the Last Universal Common Ancestor. Source: Forterre [21].

2.2.3 Sequence homology

Sequence homology refers to the common evolutionary origin of the compared DNA or amino acid sequences [24]. Biological sequences evolve and change through time by mutations, that are mainly introduced during DNA replication events (see Fig 3). On the scale of a single gene (or protein) the most common mutations are point substitutions of one residue for another and insertions or deletions of single residues or short strips of residues. These events produce diversity on the population level that is then pruned down to functional variants by natural selection. Overall, biological sequences seem to evolve through networks of functional variants, were each variant has a biological function and is related to other functional variants by duplication and mutation events [25].

2.3 Comparative studies for multiple sequence aligners

2.3.1 Multiple sequence alignments

The goal of the multiple sequence alignment (MSA) is to find one-to-one correspondences (an alignment) of amino acids (or nucleotides) that descend from a common ancestor [26]. MSAs have a very broad range of applications. These include studies of evolutionary history and phylogenetics, protein structure and function, drug design, epidemiology, virulence, human genetics, cancer and biodiversity [26,27].

In phylogenetics each column of the MSA is treated as a character and residue values as the character states. The phylogeny program then searches for the evolutionary tree that by some criteria (e.g. parsimony) and a substitution model is optimal for explaining evolutionary relationships of the compared sequences [24]. To date, thousands of phylogenetic trees have been constructed [23], many of them based on MSAs. Coverage of sequences for extant taxa is continuously improving leading to efforts for global phylogenies that would connect all existing organisms. For example, Hug et al. used MSAs of ribosomal protein sequences from all sequenced genera to construct a comprehensive Tree-of-Life [22].

In molecular biology, MSAs are often used as the initial source of information on structural and functional properties of novel biological sequences. For example, a newly sequenced gene can be aligned against homologs in a database to gain fast insight on structural and functional motifs that are likely to be present in the sequence.

In epidemiological and virulence studies, MSAs are often used to visualize sequence variation responsible for virulence or drug resistance in pathological strains. For example, the NCBI Influenza Virus Resource [28] includes online MSA alignments and phylogenetic trees for human influenza strains [28]. NCBI offers similar resources for a number of zoonotic viruses [29] including the currently relevant SARS-Coronavirus-2 (referring here to the COVID19 outbreak in Wuhan, China).

In human genetics, MSAs can be of value by highlighting regions of cross-taxonomic conservation in human genes linked to diseases. Locating mutations to conserved regions can help to elucidate the causative mechanisms of the disease [30]. In cancer research, missense substitution analysis has been developed to identify mutations linked to elevated cancer risk [31]. The central idea here is that mutations in conserved regions and mutations falling outside the range of variation among related species elevate the risk of cancer. Both conserved regions and variation are detected in these analysis using high quality MSAs [32].

MSAs are also a prerequisite for position-specific scoring matrices (PSSMs) and profile hidden Markov models (profile HMMs), which are both important tools in computational biology. PSSMs and HMMs can be constructed from MSAs to represent families of related sequences [33]. Novel sequences can be compared against these models to identify homologs or homologous regions. PSSMs were introduced by Stormo et al. in 1982 [33] and have been applied in MSA visualizations, motif finding and increasing the sensitivity of database searches, such as PSI-BLAST [33–35]. HMMs were introduced later and represented a more advanced and accurate way to model sequence homology [36]. Large collections of biological sequence families are now available in HMM format in public databases such as Pfam [37] and TIGRFAM [38]. These databases can be searched for homologs using popular search engines such as SAM [39], HHsearch [40] and HMMER [41]. PSSMs and HMMs are discussed in more detail in sections 3.1 and 3.2, respectively.

To summarize, there is as broad spectrum of applications that start with an MSA and depend on MSA quality. Evaluating, benchmarking and improving MSA quality is thus of considerable importance for many fields of biological and biomedical research.

2.3.2 Motivation and goals

Several authors [3,4,26] have discussed the motivation, aims and implementation options for comparative studies in the MSA field. The particular aims for MSA evaluation include 1) informing method developers on the pros and cons of different alignment algorithms [3,4,26], 2) providing updates on novel tools and their quality [3], 3) providing non-specialists with method rankings and recommendations [3] and 4) avoiding biases related to self-assessment [3,4]. These aims are in agreement with the general aims of comparative studies [1,2].

2.3.3 Defining benchmarks for multiple sequence alignments

Computational models for sequence homology (discussed in 3.1.3) are imperfect and may not yield biologically meaningful alignments. To keep the field on track, several curated references, referred as *MSA benchmarks*, have been created. Most MSA benchmarks are based on expert-curated reference alignments derived from structural correspondence between aligned sequences [3,4].

The first large-scale benchmark designed specifically for the evaluation of MSA software was the Benchmark Alignment database (BAliBASE) published in 1999 [42]. Reference alignments in BAliBASE are mainly based on superposition of 3D and 2D protein structures followed by manual validation and refinement steps [26,42,43]. BAliBASE is to date considered by many authors to be the most useful benchmark for MSA [3,4,44] and is included in almost all of the comparative MSA studies. The utility of BAliBASE stems from its comprehensive coverage of different sequence types, that can potentially affect the alignment accuracy. The first release of BAliBASE (version 1.0) included five main datasets (references 1-5) [42]. Later releases expanded BAliBASE with refences 6-8 (release 2.0), reference 1-5 update (release 3.0), reference 9 and reference 10 (release 4.0).

Following BAliBASE release in 1999 several other MSA benchmarks emerged. OXBench was published in 2003 [45], PREFAB in 2004 [46] and SABmark in 2005 [47]. Similar to BAliBASE, these benchmarks are based on superposition of 3D protein structures. Advantages of OXBench, PREFAB and SABmark are the large database size, extensive coverage of the protein fold space and fully automated compilation and updating procedures [45–47]. The main shortcoming of these benchmarks compared to BAliBASE is the lack of diversity in the reference sets [44]. These benchmarks do not cover the multitude of input sequence sets encountered in real-life situations [26,43]. Unlike BAliBASE, these benchmarks also lack manual refinement or other steps incorporating expert knowledge [4].

Historical development after the release of MSA benchmarks indicates that these had an important role in fostering MSA method development [3,26]. Benchmarks can be compared to network hubs, which greatly improved information flow within the MSA research community (see Fig 1 C). The release of BAliBASE highlighted the pros and cons of alignment algorithms [42]. Specifically, it was shown that no existing MSA algorithm was able to perform well on all reference sets. Global alignments were shown to perform well on conserved and local alignment on diverged sequences. Also, all aligners struggled with sequence sets with below 20% pairwise identity [26,42,44]. This low range of identity, also known as the “twilight zone”, became one of the focal points of improvement for many novel MSA software [27]. Combining local and global alignments, consistency transformations and iterative refinement techniques, statistical algorithms and other innovations led to the publication of several novel software packages with notable improvement in performance. These included publication of T-Coffee in 2000 [48], MAFFT in 2002 [49], Muscle in 2004 [46], ProbCons [50] and Kalign in 2005 [51], MSAProbs in 2010 [52] and ClustalO in 2011 [53]. These software packages are discussed in more detail in section 3.1.4.

2.3.3.1 Performance metrics for MSA benchmarks

Any discussion on benchmarks for MSA evaluation would be incomplete without a description of performance metrics. These define a function or a procedure that measures how accurately the MSA output by a given method matches the reference MSA. The most widely adopted performance metrics for MSAs are the sum-of-pairs score (SPS) and the true-column score (CS) [3]. These are the default metrics proposed for BAliBASE [26,42,43]. SPS is the recall of correctly aligned residue pairs averaged across all sequence pairs and CS is the recall of correctly aligned MSA columns. SPS is better suited for sets of distantly related sequences, since these may lack correctly aligned columns [3]. CS score is less sensitive to small differences in accuracy, but can be illustrative for sets of closely related sequences [3]. For PREFAB the proposed metric is the Q score, which correspond to the recall of correctly aligned residue pairs in the structurally aligned pair [46]. For SABmark the proposed metrics are FD and FM scores, which correspond to the recall and precision of correctly aligned residue pairs [47]. The SPS, Q and FD scores for these benchmarks are thus very similar. Also other metrics have been proposed, e.g. the structure-oriented metrics in the OXBench benchmark [45].

2.3.4 Alternatives to MSA benchmarks

Simulated datasets provide an appealing alternative to real benchmarks. These are sets of sequences and reference alignments generated under a known evolutionary model for substitutions, deletions and insertions. For this purpose many software packages have been developed including SIMPROT [54], Indel-Seq-Gen [55] and INDELible [56]. Generally, these implementations allow to tune the underlying evolutionary model with a number of input parameters. There are several clear advantages of simulated data relative to real benchmarks [4,57,58]. First, datasets can be generated with great efficiency. Second, the correct homology in these datasets is known. Third, a large variety of datasets with different mutation frequency, size and location of indels, length, domain composition and other properties can be generated in a short time [4,57,58]. Finally, simulations can generate alignments for non-coding regions which generally lack structure-based references [59].

Although applying simulated datasets can greatly simplify the evaluation process this approach has serious disadvantages. The main concern is that simulations are based on models of evolution that cannot account for all evolutionary forces operating in reality [4,60]. When comparative evaluation is centered on simulations, there is a serious risk of optimizing methods for scenarios that have little or no relevance to real biological data [4]. Another risk is that simulations can favor MSA methods with similar underlying evolutionary models [4]. For these reasons simulated data is often used in combination with real benchmarks [57,58].

In addition to simulations, there are still other alternatives. For example, Dessimoz and Gil [60] proposed to use tests based on phylogenetic trees to evaluate MSA quality. Phylogeny based methods have an advantage of being independent of reference alignments [4]. These also provide a way to evaluate gap-rich and highly divergent regions [4]. The disadvantage of these methods is that they ignore any similarity between the aligned sequences that is not rooted in a common ancestor. It is known that structural or functional similarity may occur between unrelated residues by convergent evolution [61]. And though evolutionary applications of MSAs are mainly interested in homology, biologists are often interested as much in functional and structural similarities [61].

2.3.5 MSA comparative studies

Table 1 presents a selection of comparative studies for MSA methods. These were published during 2006-2014 evaluating the state of the art MSA aligners on benchmark datasets described in section 1.4.3. Performance of the compared aligners is summarized by splitting methods into categories denoted “high” and “low accuracy” as well as “fast” and “slow”. Note that these categories are nominal and do not indicated absolute accuracy or speed. Table 1 can be used to draw general conclusions about MSA method performance. Results indicate that ProbCons and MAFFT are among the few programs that show consistently high accuracy across all evaluations and all datasets. MAFFT, Kalign and Muscle are consistently among the fastest MSA aligners. MAFFT seems to be the only program that consistently appears in both “fast” and “accurate” categories.

Table 1 also illustrates typical benchmarks, evaluation metrics and evaluated methods in MSA comparative studies. Tests on different benchmarks resulted in different method rankings, which illustrates the importance of testing on many datasets [1,2]. Program suites included in different studies were surprisingly similar (program versions are not considered here). All six studies included ClustalW [62] (published in 1994) or ClustalO [53] (2011), T-Coffee [48] (2000), MAFFT [49] (2002), Muscle [46] (2004) and ProbCons [50] (2005). Four out of six studies also included Kalign [51] (2005) and three out of six studies included POA [63] (2002). The number of compared MSA methods varied between eight to ten. The most common performance metrics were SPS and SC, although there was a certain discrepancy between metrics applied in different studies.

Table 1. ***MSA comparative studies.*** *This table presents a selection of comparative studies for multiple sequence aligners (MSAs). High accuracy and Low accuracy columns divide evaluated methods into two roughly equal-sized groups based on the performance metric used in the study. Fast and Slow columns divide methods into two groups based on their execution time. The absolute time threshold for fast-slow division is stated at the top of the method list whenever available. Exact method rankings are included as numbers whenever these are clearly stated in the original publication. References: Blackshields 2006* [44], Nuin 2006 [58], Perrodou 2008 [64], Thompson 2011 [26], Pais 2014 [65] and Pervez 2014 [57].*
**Method ranking reported only for OXBench. For MAFFT and Dialign only the top performing variants are listed.*

Study	Benchmark	High accuracy	Low accuracy	Metric	Fast	Slow
Blackshields 2006*	OXBench Master	1. ProbCons 2. Muscle 3. MAFFT-ginsi 4. PCMA 5. ClustalW	6. T-Coffee 7. Dialign-t 8. Align-m 9. POA	Shift score	NA	NA
Nuin 2006	Simprot simulated data	1. ProbCons 2. MAFFT-linsi 3. MAFFT-fftns2 4. T-Coffee 5. Muscle	6. Kalign 7. ClustalW 8. Dialign-T 9. Dialign2.2 10. POA	SPS	1. MAFFT-fftns2 2. Kalign 3. Muscle 4. POA 5. MAFFT-linsi	6. ClustalW 7. Dialign-T 8. Dialign2.2 9. ProbCons 10. T-Coffee
Nuin 2006	BAliBASE 3.0, Ref1-5	1. ProbCons 2. MAFFT-linsi 3. T-Coffee 4. Muscle 5. Kalign	6. ClustalW 7. MAFFT-fftns2 8. Dialign-T 9. Dialign2.2 10. POA	SPS	NA	NA
Perrodou 2008	BAliBASE 3.0, Ref9, V11	1. ProbCons 2. MAFFT-linsi 3. Mummals 4. Muscle 5. Muscle-fast	6. T-Coffee 7. MAFFT-fftn2 8. Kalign 9. ClustalW 10. Dialign	Friedman test on SPS	<1000 sec 1. Kalign 2. MAFFT-fftns2 3. Muscle-fast 4. ClustalW 5. MAFFT-linsi	>1000 sec 6. Muscle 7. Dialign 8. ProbCons 9. T-Coffee 10. Mummals
Thompson 2011	BAliBASE 4.0, Ref10	1. MAFFT-linsi 2. T-Coffee 3. ProbCons 4. MAFFT-fftns2	5. Kalign 6. Dialign 7. Muscle 8. ClustalW	CS	1. Kalign 2. MAFFT-fftns2 3. MAFFT-linsi 4. Muscle	5. ClustalW 6. Dialign 7. T-Coffee 8. ProbCons
Pais 2014	BAliBASE 3.0, Ref1-9	ProbCons T-Coffee ProbAlign MAFFT	Muscle ClustalO ClustalW Dialign-TX POA	SPS and CS	ClustalW Muscle MAFFT ClustalO	ProbCons ProbAlign T-Coffee
Pervez 2014	iSG simulated data	1. ProbCons 2. SATe 3. MAFFT-linsi 4. Kalign 5. Muscle	6. MAFFT-fttns2 7. T-Coffee 8. ClustalO 9. Dialign-TX 10. Multalin	SPS and CS	<1 h 1. MUSCLE 2. MAFFT-fttns2 3. Multalin 4. Kalign	>11 h 5. ClustalO 6. SATe 7. Dialign-TX 8. MAFFT-linsi 9. T-Coffee 10. ProbCons
Pervez 2014	BAliBASE 3.0, Ref1-5	SATe ProbCons MAFFT-linsi	T-Coffee Muscle ClustalO Kalign Dialign-TX Multalin	SPS and CS	<1 h 1. Kalign 2. MUSCLE 3. MAFFT-fttns2 4. ClustalO 5. MAFFT-linsi	>1 h 6. Dialign-TX 7. Multalin 8. SATe 9. T-Coffee 10. ProbCons

2.3.6 Limitations of MSA comparative studies

It seems that many authors are still underestimating the importance of testing on different benchmarks [1,2]. Only three out of six studies in Table 1 included more than one benchmark in their evaluation. From these, two studies complemented BAliBASE with a simulated dataset [57,58] and only one study included more than one real benchmark. This was the study published by Blackshields et al. which included an exhaustive set of six real benchmarks [44]. The remaining three studies were based on one to several reference sets from BAliBASE.

There was a certain amount of discrepancy between evaluation metrics applied in the examined studies (Table 1). Most studies reported sum-of-pairs and column scores which are the default performance metrics for BAliBASE. Still, some studies adopted unique evaluation metrics [44] or reported only method ranks based on statistical tests [64]. Differences in evaluation metrics creates difficulties for interpreting results (see Fig 1). For instance, it is not clear which differences in rankings are due to different benchmarks and which are due to different metrics. Furthermore, in many of these studies, clear rankings of the compared methods were not published. However, most of these studies did provide recommendations on pros and cons of different methods in the form of a discussion.

Some authors did not report execution time [44], chose to report execution time in a confusing graphical format [65] or to report only relative execution times [58] (Table 1).

Only one of the examined studies evaluated the effect of input parameters on the MSA accuracy (Table 1). Blackshields et al. [44] performed an extensive optimization of GOP and GEP penalties for three different methods (Muscle, ClustalW and MAFFT).

Only one of the examined studies evaluated the effect of different alignment strategies and these were limited to iterative refinement in MAFFT (iterative refinement is discussed in section 3.1). MAFFT offers several iterative refinements that can be specified by input parameters [66]. These include a single iterative refinement (fft-ns-2), multiple iterative refinements (fft-nn-i), iterative refinement with consistency scores from local pairwise alignments (fft-linsi) and iterative refinement with consistency scores from global alignments (fft-ginsi). These options were included in the comparative study by Blackshields et al. [44] although there was no synthesis on the observed differences. Anyhow, the degree of control offered by MAFFT is rather an exception and even for MAFFT it is limited. Most MSA aligners offer even less control over the alignment algorithm. Thus, in most cases it is not possible to compare alignment strategies within the framework of comparative studies. Exploration and comparison of alignment strategies is thus left to method developers, which, however, can be biased and selective in their reporting (see section 1.1).

2.4 Automated function predictors

Genes are sequenced at a much higher rate than their function or structure can be determined experimentally. This has created a demand for automated function predictors (AFPs), i.e. methods that can automatically predict function for novel sequences. Most of AFPs are based on homology and assign function to query sequences by comparison with sequences in annotated databases. The majority of sequences in these reference databases are also annotated with AFPs. Even in UniProtKB/SwissProt, which is one of the largest manually curated protein databases, 69.0% of the 561k sequences are annotated by homology [67]. In a much larger automatically curated UniProtKB/TrEMBL (172M sequences in 09_2019 release), all sequences are annotated by homology [68]. In many cases de novo experimental annotations can be considered redundant because homology to genes (or other sequences) that already have experimental annotation is a very strong evidence for similar function.

The main goal of AFPs is to assign sequences with biologically meaningful function. Although function can be described in various ways, adopting a controlled vocabulary has many advantages. A controlled vocabulary improves communication between research groups, simplifies integration of computational tools into workflows and aids objective evaluation and comparison of different methods. The majority of AFP methods have adopted the Gene Ontology (GO) [69] as a controlled vocabulary for annotating genes [70].

2.4.1 Gene ontology

Gene ontology (GO) is a predefined and curated set of functional terms or classes that are arranged into a hierarchical graph by their semantic relationships. In terms of graph theory, GO is a directed acyclic graph with vertices representing GO terms and edges representing semantic relationships between terms. GO comprises three orthogonal ontologies, which describe distinct aspects of gene products: molecular function (MF), cellular component (CC) and biological process (BP) [71,72]. MF ontology describes the function of gene products at the molecular level. CC ontology assigns gene products to cellular locations such as cytoplasm and cell membrane. BP ontology assigns gene products to pathways and to larger processes to which they contribute.

GO release 2019-10 has 44.7K GO terms including 29.5K terms in BP, 11.1K terms in MF and 4.2K terms in CC subontologies [73]. This release has 7330K annotations for 1405K genes split almost equally between MC, CC and BP subontologies. Only about 11% (780K) of all annotations are labeled with evidence codes indicating direct experiment (evidence codes EXP and HTP). The vast majority of annotations are based on curated or fully automated AFPs: 46% are annotated by phylogeny-based AFP (code PHYLO referring to annotations with PAINT [74]), 27% by fully automated AFPs (code IEA), 13% by various curated AFPs (codes ISS, ISO, ISA, ISM, IGC and RCA) and the remaining 4% are based on author or curator statements (codes TAS, NAS, IC and ND).

2.4.2 Comparative studies for automated function predictors

Before turning to AFP comparative studies, it is appropriate to estimate the scale of the evaluated field. The latest large scale AFP challenges (see the next section) have covered over 100 AFP methods for each challenge [75,76]. Most likely these competitions did not cover all of the published methods. To make a better estimate, we searched PubMed with keywords "(novel OR new) function annotation method". According to the search results an average of 600 studies introducing novel methods are published annually (Fig 5). Taking together the PubMed search and statistics from AFP challenges, we estimate that the number of novel AFPs published each year ranges in the hundreds.

With this plethora of methodologies, it is difficult and time-consuming to keep updated on the best tools available. The situation is further complicated by the tendency of different method developers to use different evaluation metrics on different benchmarks (Fig 1). Furthermore, it is well established that authors are prone to a number of self-assessment biases (see section 1.1). These difficulties inhibit information flow within researcher communities involved in developing and applying AFP methodologies (Fig 1). As discussed in section 1.2, most of these difficulties can be addressed by comparative studies, which act as information hubs that process information from method developers and redistribute it to other developers and users. In the field of AFP, comparative studies have adopted the challenge-based format discussed here.

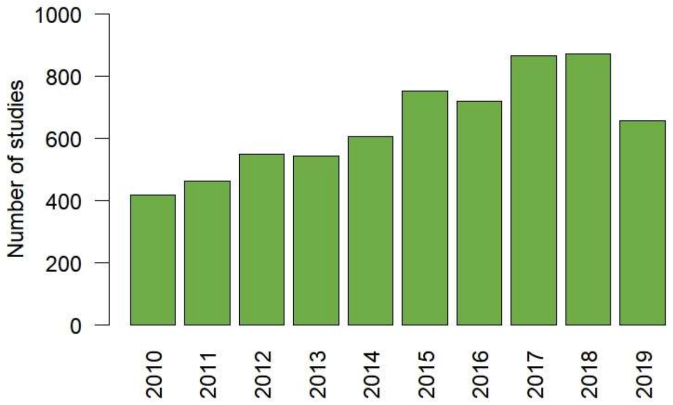


Figure 5 Annual publications related to novel AFP methods. Statistics were collected by querying PubMed with keywords "(novel OR new) function annotation method". Note that the number of annual publications ranges in hundreds.

2.4.3 CAFA challenges

The Critical Assessment of protein Function Annotation algorithms (CAFA) is a large-scale, third-party, prospective evaluation challenge for automated function predictors (AFPs) that assign gene or protein sequences with Gene Ontology [69] and Human Phenotype Ontology terms [77]. To date three CAFA competitions have been completed (years 2010-2011 [70], 2013-2014 [75] and 2016-2017 [76]) and the fourth competition is ongoing (years 2019-2020). The number of methods evaluated in CAFA I-III add up to 332 methods submitted by 154 research teams making these challenges truly large-scale [76].

CAFA is based on prospective evaluation that can be subdivided into prediction, annotation growth and assessment phases. At the start of the competition a set of target protein-coding genes is selected. During the prediction phase, method developers can submit functional predictions for the target genes made by their methods. During the annotation growth phase, new experimental annotations are accumulated. In the assessment phase, submitted predictions are evaluated against the reference annotations accumulated during the annotation growth phase. This arrangement ensures independency between training and testing datasets.

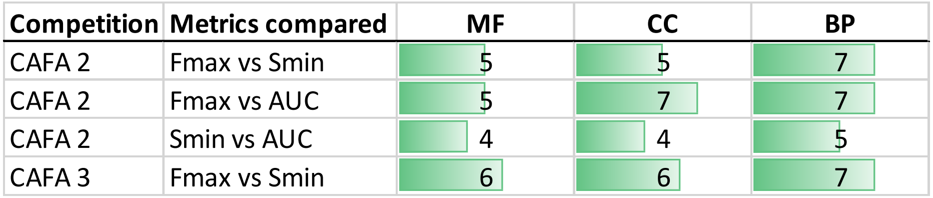
In CAFA I-III assessment was mainly based on four evaluation metrics: graphical precision-recall curves, area under the receiver operating characteristic curve (ROC AUC), F_{max} and S_{min} (for details, see section 3.3). In the CAFA II article [75], the top 10 method rankings were published for F_{max}, S_{min} and ROC AUC, while in the CAFA III preprint article [76], rankings were only published for F_{max} and S_{min}. AFPs are compared on the full set of target genes and also on various subsets, such as eukaryotic and prokaryotic targets, as well as targets from selected key species like *Escherichia coli* and *Arabidopsis thaliana.* In this thesis, I only discuss results for evaluations on the full target set. Methods are compared against each other and against two baseline methods: a Naïve method (referred also as the null model), which assigns all genes the same set of GO terms with confidence scores derived from term frequencies, and BLAST, which assigns terms from top BLAST hits.

2.4.4 Limitations of CAFA challenges

Although CAFA is a step forward towards a more objective evaluation of AFP methods, it has several shortcomings. Several authors have expressed their concern about bias and instability of the produced AFP rankings. Gillis and Pavlidis [78] observed that the F_{max} metric for CAFA is "unsatisfactory. ... by this measure, a null ‘prediction method’ outperforms most methods.” Furthermore, Kahanda et al. [79] pointed out that the ranking of methods may vary considerably between different CAFA metrics.

In Table 2, the top 10 performing methods by F_{max}, S_{min} and ROC AUC rankings are compared. We see that for all three subontologies there is considerable disagreement between different evaluation metrics. For example, in F_{max} and S_{min} rankings for MF predictions in CAFA II only 5 out of 10 of the top performing methods are the same. The remaining 10 methods in these two lists are thus different. In CAFA III the situation is similar: 6 methods are the same and the remaining 8 methods are different. Furthermore, F_{max} and S_{min} give different ranks even to the common methods in the top 10.

Table 2. *Comparing top 10 AFPs by different evaluation metrics in CAFA II-III. The number of common AFPs in the top 10 rankings by the compared metrics is listed in the last three columns. The green bar represents numerical values graphically. Here MF, Molecular Function, CC, Cellular Component, BP, Biological Process subontologies. AFP rankings were collected from [75,76].*



Competition	Metrics compared	MF	CC	BP
CAFA 2	Fmax vs Smin	5	5	7
CAFA 2	Fmax vs AUC	5	7	7
CAFA 2	Smin vs AUC	4	4	5
CAFA 3	Fmax vs Smin	6	6	7

This discrepancy in method rankings creates confusion in the interpretation of CAFA results. There are several AFPs that appear top ranked in both F_{max} and S_{min} lists, however there are also AFPs that appear with a good rank in only one or the other list. Furthermore, if the rankings differ this much, can they be considered an objective and unbiased guideline for AFP selection and future method development? Clearly, AFP evaluation requires further research in order to avoid such confusion in future competitions. In my opinion, the main source of confusion is the lack of research into the properties of different evaluation metrics. Different metrics are designed to monitor different types of errors and may have different biases, and thus can produce different rankings. For example, F_{max} mainly monitors the number of false positive and false negative predictions while S_{min} monitors the information content of these erroneous predictions (for details, see section 3.3). In this thesis I argue that it may be necessary to redefine and adopt evaluation metrics that are currently used for AFP evaluation in order to exclude possible biases and to improve consistency of the resulting method rankings.

3. AIMS OF THE PRESENT STUDY

The aim of this thesis is to investigate and develop frameworks for unbiased evaluation of homology-based methods. This aim is approached from three different perspectives, each addressed by one publication.

I Publication (I) introduces a novel homology-based method for annotation of gene-clusters. The questions that arise in this problem include: Given a clearly defined type of a gene-cluster, such as a pilus operon, how to detect genes that are related to it? How to select gene-clusters that are non-random? How to evaluate performance of a new annotation tool?

II Publication (II) is focused on comparative evaluation of MSA alignment strategies. The questions addressed here: Which alignment tools are available and what is their performance? What differences and similarities can be found in alignment strategies implemented by different tools and how do these impact alignment accuracy? Is it possible to achieve high performance by integrating the best parts from different MSA implementations?

III The last publication (III) is focused on metrics for comparative evaluation of AFPs. This part attempts to answer the following questions: What evaluation metrics are commonly used in comparative studies of AFPs? Are these metrics unbiased? Are these metrics consistent? Are certain metrics better than others and how this can be quantified? Is it possible to design optimal metrics?

4. MATERIALS AND METHODS

4.1 Multiple sequence alignment methods

This section discusses more in detail algorithms and implementations for creating and benchmarking multiple sequence alignments (MSAs). The discussion progresses from classical algorithms for pairwise alignments, to multiple alignments, and further to the state-of-the-art implementations and benchmarking. The last part discusses a modular framework for comparative evaluation of MSA strategies introduced in publications (II).

4.1.1 Pairwise sequence alignment

Two sequences can be aligned with a simple scoring function and an optimization algorithm [24]. The minimal scoring function for aligning two sequences can be constructed from a set of substitution scores and gap penalties (equation 1). Substitution scores, *S(xi,yi),* define scores for all possible pairwise substitutions of aligned residues. For protein sequences this is a 20 x 20 substitution matrix. For global alignments, the substitution matrix may include any positive numbers with larger numbers assigned to the more likely substitutions. For local alignments, the expected score of two random sequences must be less than zero imposing constrains on the selected scores. When substitution scores are defined as log-likelihoods of target and background probabilities (discussed in 3.1.3) the resulting alignment will be a more accurate reconstruction of the evolutionary events connecting the two sequences. In the affine gap model, a gap opening penalty (*GOP*) is added once for each gap opening event and a gap extension penalty (*GEP*) is added for events extending the gap [24].

(1) $$S(x,y) = \sum_{i\ in\ aligned\ positions} S(xi,yi) - \ d * GOP - e * GEP$$

Using this scoring function, an optimal global alignment can be constructed using the dynamic programming algorithm introduced by Needleman and Wunsch [80] and later refined by Gotoh [81]. Needleman-Wunsch is based on a simple rule, which states that the best score aligning prefixes x_i and y_j, $S(x_i,y_j)$ depends only on the best scores for sequence prefixes that are one residue shorter. This allows optimal alignments to be constructed in a step-by-step fashion, at each step updating the matrix of the best scores. At the end of this procedure, the best score for the entire global alignment will be calculated as the last entry of the matrix. When calculating scores, we also keep pointers to the shorter alignments that we extend at each step. This allows for a trace back from the last pair of aligned residues to the first, yielding a global alignment.

Many proteins tend to have a mosaic structure composed of functional domains and this structure can evolve over time by various recombinational events [82,83]. To find homology between multidomain proteins, we need to retrieve local alignments. Local alignments can be collected by modifying Needleman-Wunsch so that new alignments can be initiated at each alignment step. One of the first algorithms of this type was introduced by Smith and Waterman [84]. Smith-Waterman finds a single local alignment that gets the highest score. If we want to retrieve all local alignments that score above a certain threshold, we can use Waterman-Eggert [85], which is a straightforward modification of Smith-Waterman.

4.1.2 Multiple sequence alignment

To align multiple sequences, we also need a scoring function and an optimization algorithm. A simple and commonly applied scoring function is the sum-of-pairs score (equation 2). Here *k* and *l* are sequences and *i* is the alignment position. This scoring is an extension of equation (1), that simply sums scores for all pairwise sequence comparisons. Gap penalties are moved to the substitution matrix that now has scores for aligning each residue against a gap.

$$(2) \qquad SP(m) = \sum_i \sum_{k<l\leq N} S(m_i^k, m_i^l)$$

Adopting SP score leads to two core problems in constructing MSAs. First, optimal pairwise scores and alignments are often in conflict between pairs of aligned sequences. In other words, residues of all sequences cannot be arranged in columns such that these would correspond to optimal pairwise alignments. Second, optimization of the SP score is challenged by exponential time and memory complexity. Optimizing the SP score with dynamic programming requires $O(L^N)$ memory and $O(2^N L^N)$ time, which is clearly intractable for more than a few sequences [24]. The main approach has been to relax the combinatorial search for the optimal alignment and to search instead for suboptimal solutions using heuristic algorithms. The most commonly used heuristic for building MSAs is progressive alignment [26]. This was introduced by Feng and Doolittle [86] and later extended by many authors. The general idea is to build a guide tree that relates individual sequences and then to progressively align sequences or groups of sequences using that tree.

4.1.3 Alignment scoring models

To understand different scoring models, it is important to recall that aligned sequences are a sample from a larger sequence space, where all sequences are interconnected by evolutionary events [25]. Because natural selection will only allow functional variants, the observed sequences in the sequence space are only the functional sequences. It also follows that mutations connecting observed sequences are restricted to mutations that preserve function [25]. Thus, observed mutations depend on the structure, position and function of the residues that are changed in evolutionary events. This allows us to model these events with statistical models of varying complexity. Substitution matrices are limited to the modeling of dependencies between mutations and the structural properties of different residues (e.g. hydrophobicity and size). Position specific scoring matrices (PSSM) add to the substitution model dependencies on sequence position. Hidden Markov Models (HMM) cover position dependent substitution events, but also insertions, deletions, duplications and, possibly, other events.

4.1.3.1 Log-odds scores and substitution matrices

Substitution matrices define scores for substituting any residue to any other residue. For nucleotide residues these are 4 by 4 matrices, and for amino acids 20 by 20 matrices. A substitution matrix can be interpreted as a collection of log-odds ratios [87]. Each log-odds is then the logarithm of target and background probabilities (equation 3). Here the nominator (target probabilities) is the probabilistic model for observed substitution events and denominator (background probabilities) is the probabilistic model for observing random alignments. Target and background probabilities can be estimated empirically from a reference collection of aligned sequences.

$$(3) \qquad S(i,j) = \log\left(\frac{p(x_i,y_j)}{q(x_i)q(y_j)}\right)$$

PAM matrices define log-odds for point accepted mutations (PAMs) at different average mutation rates [88]. These are the observed mutations, i.e. point mutations introduced by evolutionary events and accepted by natural selection. PAM scores were estimated by Dayhoff from a corpus of aligned sequences in 71 families of closely related proteins. From these data, Dayhoff estimated target and background probabilities that were used to calculated log-odds scores for PAM matrices. For example, PAM_{60} assumes an average of 60 PAMs for each 100 amino acids and PAM_{120} an average of 120 PAMs. In practice, the lower the homology of the aligned sequences, the higher PAM index is appropriate for scoring the alignment [88].

BLOSUM scores were estimated by Henikoff and Henikoff from local alignments from the BLOCKS database [89]. In BLOSUM matrices, target probabilities are estimated directly from subsets of sequences of varying evolutionary distance. For example, BLOSUM62 is estimated based on sequences with a mean pairwise identity of 62%. Commonly used BLOSUM matrices are BLOSUM45, BLOSUM62 and BLOSUM80. In practice, the lower the homology of aligned sequences, the lower BLOSUM index is appropriate for scoring the alignment.

Scoring based on log-odds is convenient for detecting homology from the background of random similarity. Still, non-significant similarity does not exclude homology, which has been shown by many cases of structural comparison [90–92]. This has motivated researchers to develop more powerful scoring systems that are based on position-specific target frequencies derived from MSA-columns.

4.1.3.2 MSA-based scoring: position-specific scoring matrices

Position-specific scoring matrices (PSSMs) are estimated from MSAs. From each column in the MSA, target frequencies are estimated, normalized by background frequencies and log-odds are calculated. PSSMs can also include log-odds scores for gap insertions at each position in the MSA. PSSMs were introduced by Stormo et al. in 1982 and have been applied in MSA visualization, motif finding and increasing sensitivity of database searches, such as PSI-BLAST [33–35].

Compared to substitution matrices, PSSMs have advantages and disadvantages. The advantage is that PSSMs estimate substitution scores for each position. This is clearly a more realistic model for protein sequences, which can have multiple and complex interactions in their 3D structure. PSSM scores are also estimated from a relevant subsample which can have target frequencies deviating from PAM or BLOSUM estimates. The disadvantage is that PSSMs require a collection of related sequences and these must be arranged in an MSA. Also, PSSMs do not have a clear model for insertion and deletion events.

4.1.3.3 MSA-based scoring: profile-profile comparison

The most distant protein homologs can be detected by aligning HMM profiles against other HMM profiles. This approach has been implemented in HHsearch [40]. HHsearch defines a novel scoring function that relates the probability of two HMMs emitting the same amino acid at a given position to a null model distribution. Scores for all 20 amino acids at all positions are summed and a logarithm is taken, hence this scoring function is referred as the log-sum-of-odds score [40]. This scoring function in effect compares the distribution of emitted amino acids at each position in the aligned profiles, adding a positive score when these distributions match and a negative score when they mismatch. The two HMMs are aligned using dynamic programming similar to sequence to HMM alignments. Benchmarking on SCOP data showed that HHsearch was more sensitive in recovering distant families that belong to the same superfamily than both PSI-BLAST and HMMER.

4.1.3.4 Added information from intermediate sequences

For protein MSAs, pairwise identity below 20% is referred to as the twilight zone. For these cases, reconstructing homology can be difficult even for state-of-the-art software [27]. Since the aligned sequences are a sample from an interconnected sequence space, common origin of homologous residues is easier to reconstruct when intermediate sequences are added. For example, MAFFT-5 showed ~10% improvement in the MSA accuracy by first adding close homologs to the input and then removing them from the output [93].

Including intermediate sequences can also guide MSA optimization. Due to exponential scaling (see 3.1.2) most MSA implementations have adopted progressive alignments, which is challenged by inconsistencies between subalignments. By transforming scores in the subalignments to consider intermediate sequences, these conflicts can be relaxed. These techniques can be collectively referred as consistency transformations [27]. To understand relationships between various consistency transformations, it is illustrative to redefine the alignment scoring function in terms of an alignment graph.

In our definition of the alignment graph residues are represented by vertices and homology by edges. Other definitions are possible: Rauch et al. used an alignment graph where sequence segments from synteny blocks were represented by vertices and alignments between these segments by edges [94]. In both definitions, edges representing the homology links can be made more consistent by integrating information from pathways that pass through one or more intermediate vertices. In T-Coffee, this was implemented as triplet library extension and considered all paths through a single intermediate vertex [48]. In MaxFlow, consistency was introduced by assigning edge weights equal to the ratio of common versus all neighbors (i.e. Jaccard index) of the paired vertices [95]. MaxFlow also introduced transitivity to the scoring function [95]. This was implemented by considering all possible paths between a pair of vertices and by assigning the edge weight to the minimal Jaccard index among these paths [95]. In publication (II) we introduced a simpler version of MaxFlow referred as the *clique transformation*. Like MaxFlow, clique transformation connects distant vertices by the minimal edge score among all possible paths between the vertices. Unlike the MaxFlow clique transformation does not reweight the edges with Jaccard index.

In ProbCons, consistency transformation was defined in probabilistic terms. The program starts by estimating posterior probabilities assigned by the underlying HMM model to all pairs of residues in all pairs of sequences. These are then used to estimate transformed scores by multiplying and summing probabilities for all possible paths via a single intermediate residue in a third sequence [50]. In essence this redefined triplet library extension (introduced in T-Coffee) in probabilistic terms. In MSAProbs, a weighted probability consistency transformation was introduced. Similar to ProbCons, scores from all paths via a single intermediate residue were summed, but each path was weighted using weights assigned to sequences in that path [52].

4.1.4 State-of-the-art software

Most popular MSA implementations are based on progressive alignments [26,86] and have a similar architecture. First, pairwise similarities or distances are estimated between sequences and a guide tree is built. Second, several factors are combined to define a scoring function: sequence weights, substitution scores, gap penalties and, for some aligners, also a consistency transformation. Then sequences are aligned by the progressive procedure. Finally, the alignment is refined to reduce errors introduced by inconsistencies between pairwise alignments. In Table 3, I have summarized eight popular progressive aligners in terms of these basic steps. Table 3 also lists citation index by Google Scholar (retrieved 2019 November 10), execution time and SP performance on BALIBASE 3.0. These were the aligners compared in publication (II).

Generally, an aligner starts by estimating pairwise homology, which is then used to build a guide tree. A good estimate of homology is the fractional identity which can be estimated from global alignments. The drawback here is the overall $O(N^2 L^2)$ time complexity for building global alignments for all sequence pairs [24] (sometimes referred as the quadratic tree problem). K-mer counting and k-mer pattern scoring [24] are faster methods. K-mer counting estimates the number or the proportion of k-mers shared between pairs of sequences. Variants of k-mer counting implemented in ClustalW [62], Muscle [46] and MAFFT [49] have $O(N^2 L)$ time complexity. Even better scaling can be achieved. MAFFT version 6 implements PartTree algorithm for constructing guide trees with $O(N^2 \log(L))$ time complexity [66]. In Kalign pairwise distances are estimated using fast string matching: Muth and Manner algorithm in Kalign 2 and Gene and Myers algorithm in Kalign 3 [96].

Pairwise similarities or the corresponding distances are used as input to a hierarchical clustering algorithm. For progressive alignments, Neighbor Joining [97] and UPGMA [98] have been popular, although it is possible to use any other clustering method. For example, ProbCons achieves high accuracy with a custom algorithm similar to UPGMA [50]. In publication (II), we have shown that single-linkage clustering performs very well.

Most aligners use the basic SP scoring scheme. The main differences are in the substitution scores and gap penalties. In ClustalW, sequences are aligned with different BLOSUM matrices depending on pairwise identity: sequences with high identity (80-100%) are aligned with BLOSUM80 and those with low identity (0-30%) with BLOSUM30 [62]. In MAFFT, all alignments are done with normalized PAM200 [49] and in Kalign2 with GONNET250 [99]. Most aligners allow users to change the default substitution matrix.

MUSCLE does not use SP scoring for progressive alignment. Profiles are aligned with a log-expectation score, that is based on target probabilities, background probabilities and position-specific residue frequencies estimated from the aligned profiles. Target and background probabilities are derived from the 240 PAM VTML substitution matrix [46].

T-Coffee introduced a novel scoring function similar to our definition of the alignment graph. This combines any number of residue-to-residue homology links from local and global alignments as well as any other information sources. In the original T-Coffee program, global alignments were generated and scored with ClustalW and local alignments with Lalign [48]. This procedure defines a matrix of substitution scores for each pair of residues in each pair of sequences that is transformed for consistency and used for SP scoring during progressive alignment.

In ProbCons, HMM emission probabilities are based on BLOSUM62 log-odd scores [50]. In MSAProbs there are two HMM models: emission probabilities of the first are based on BLOSUM62 and of the those of the second on Gonnet160 [100]. Substitution scores in these programs are defined as posterior match probability matrices. These specify posterior probabilities for matching any pair of residues given the HMM model. The posterior match probability matrices for each pair of sequences are calculated using variations of the Forward and Backward algorithms.

Different aligners implement different gap models. ClustalW uses a position specific gap model that is further modified by several factors and a set of hierarchically applied rules [62]. MUSCLE, MAFFT and Kalign implement variations of the affine gap model. In ProbCons and MSAProb, gaps are modeled explicitly by the insertion states and fitted to the data. In T-Coffee, gap penalties are only used to construct pairwise alignments. ProbCons, MSAProbs and T-Coffee do not use gap penalties during progressive alignment.

Most MSA aligners implement sequence weighting, which is incorporated in the scoring function. Popular algorithms for sequence weighting are the position-based weighting [101], ClustalW method based on a guide tree [62] and the three-way method introduced by Gotoh [102]. Sequence weighting is thought to counteract bias introduced by the uneven sampling of the sequence space by the set of aligned sequences [62].

Once the guide tree and scoring function are defined, sequences can be aligned. At each fork of the guide tree two sequences or profiles are aligned using the SP-score and dynamic programming. Commonly used alignment algorithms are adaptations of Needleman-Wunsch [80], Gotoh [81] and Myers and Millers [103] algorithms (see Table 3).

MSA output by progressive alignment can be iteratively refined to reduce inconsistencies between subalignments. In this process MSA is repeatedly divided into two parts and the resulting subalignments are realigned. The process of dividing MSA during refinement can be based on several strategies. These include the leave-one-out partitioning [104], random partitioning [105] and tree-dependent partitioning [106]. In random iterative refinement, the alignment is “cut” at a random row and realigned. In tree-dependent iterative refinement, an edge is selected from the guide tree and the alignment is divided into subalignments according to the two subtrees. In the leave-one-out refinement, single sequences are realigned to the rest of the alignment.

Table 3. *Basic Architecture of eight popular MSA aligners: ClustalW [62], MUSCLE [46], MAFFT [49], ClustalO [53], T-Coffee [48], ProbCons [50], Kalign [99] and MSAProbs [52]. Column abbreviation: SP, sum-of-pairs performance on BaliBase 3.0, Time, time performance on BaliBase 3.0, Sim met, similarity metric, Dist met, distance metric, Refinement, consistency transformation and iterative refinement algorithm(s), Aligment, alignment algorithm(s), Weights, method for sequence weighting.Other abbreviations: fide, fractional identity, kmer, fraction of conserved k-mers or similar, 6mer, fraction of conserved 6-mers, G-score, posterior probability of the optimal global alignment as defined in [52], SP, sum-of-pairs score, WSP, weighted sum-of-pairs score, AGS, affinity gap score, M&M, Muth and Manner string matching, MM, Myers and Millers algorithm [103], NM, Needleman-Wunsch algorithm [80], Gotoh, Gotoh algorithm [81].*

Method	Citations	SP	Time	Sim met	Dist met	Guide-tree	Subst score	Gap score	Refinement	Alignment	Weights
ClustalW	62 261	0.75	9.48	fide kmer	1 - fide 1 - kmer	NJ	BLOSUM/PAM WSP	position specific AGS based on hierarchical rules	none	profile-profile MM	ClustalW
MUSCLE	26 500	0.82	5.18	fide kmer	1 - fide 1 - kmer -log(1-pide-pide2/5)	NJ UPGMA	BLOSUM/PAM WSP	AGS	tree-dependent partitioning	profile-profile NW	Position-based ClustalW Three-way
MAFFT	7 288	0.87	14.70	6mer FASTA	1 - 6mer	UPGMA PartTree	Normalised PAM200 WSP	gap model based on start and end gaps	tree-dependent partitioning + consistency transformation	profile-profile NW with constrains emposed by FFT homology blocks	ClustalW
ClustalO	7 171	0.84	3.37		mBed	K-means UPGMA		HMM formulation		profile-profile HMM	
T-Coffee	6 505	0.86	226.27	fide	1 - fide	NJ	SP based on pairwise alignments	ClustalW and Lalign defaults for pairwise 0 for progressive	consistency transformation	profile-profile Gotoh	none
ProbCons	1 194	0.86	113.70	Expected accuracy	none	UPGMA derivative	SP from posterior probability matrix	0	probabilistic consistency transformation, random interative refinement	profile-profile NW	none
Kalign	226	0.82	0.35	M&M	none	UPGMA	GONNET250 SP	gap model based on gap insertion/extention and terminal gaps	additional info from feature annotations	profile-profile MM	none
MSAProbs	190	0.88	105.68	Gscore	1 - Gscore(x,y)/ min{|x|,|y|}	UPGMA	WSP from posterior probability matrix	0	weighted pobabilistic consistency transformation, random iterative refinement	profile-profile NW	ClustalW

4.1.5 Multiple sequence alignment benchmarks

BAliBASE benchmark includes a large collection of reference sets that cover a spectrum of variation and challenges encountered in alignment of protein sequences. To evaluate performance related to these factors, BAliBASE 1.0 included five different reference sets [42]. Reference 1 contained alignments for close relatives and alignments for distant relatives. Reference 2 contained alignments of orphan sequences with a group of close relatives. Reference 3 contained alignments of sequences in groups with high identity within groups and below 25% pairwise identity between groups. Reference 4 and 5 contained large terminal and internal insertions [42]. References 1-5 from BAliBASE 3.0 were used for MSA evaluation in publication (II).

Protein REFerence Alignment Benchmark (PREFAB) is a large database generated automatically by supplementing structural pairs from the FSSP database with homologs found through PSI-BLAST queries [46]. Each alignment set is filtered to have at most 80% identity and is limited to a set of 50 PSI-BLAST homologs. There are 1682 alignments in the main set and 100 alignments in the weighted set (PREFAB 4.0). Notably, MSAs are evaluated against a single pair of structurally aligned sequences for each of the reference alignments.

Sequence Alignment Benchmark (SABmark) contains pairwise structural alignments from SOFI and CE databases, that are organized according to SCOP classification [47]. SABmark 1.65 contains two main references: the “Twilight Zone” and “Superfamilies”. Twilight Zone reference is a collection of 1740 single domain protein sequences grouped into 209 SCOP folds. Most sequences in this reference have pairwise identity well below 25%. Superfamilies reference contains 3280 single domain sequences grouped into 425 SCOP superfamilies.

4.1.6 Modular Multiple Sequence Aligner (II)

In publication (II) we analyzed contemporary MSA alignment strategies in terms of their finite components that can be rearranged for evaluation and optimization. We implemented a novel Modular Multiple Sequence Aligner (MMSA) in C++. Our implementation was based on SeqAn, an open source C++ library for sequence alignments created and updated by the scientific community [15,107]. By selecting SeqAn we supported implementation transparency, which, in our view, is important for efficient method development. Our implementation had a modular structure, which allowed us to swap different components of the alignment process and, thereby, to investigate their contribution to the alignment quality and computational efficiency. To compare alignment strategies, we systematically varied information sources, guiding trees, consistency transformations and iterative refinement strategies, evaluating the resulting alignments on BAliBASE and SABmark (II).

To place this research into the context of existing MSA software, the best MMSA strategies were compared to a selection of MSA aligners (listed in Table 3) that have been previously included in similar comparative studies (see Table 1). Evaluation was done on three benchmark databases: BAliBASE 3.0 [43], SABmark 1.65 [47] and PREFAB 4.0 [46].

For more details on this work please refer to the attached publication (II).

4.2 Annotating gene clusters

In publication (I) we presented a novel homology-based method for LOcating Pilus operons (LOCP) in bacterial genomes. The key techniques employed in LOCP were: (i) homology searches with profile hidden Markov models (profile HMMs) (ii) enrichment statistics and (ii) multiple hypothesis testing. Techniques employed in publication (I) are interlinked with publications (II) and (III). Namely, HMMs can be applied in sequence alignments, while homology searches and enrichment statistics are common strategies for AFPs.

Annotation of genes by homology is a highly successful strategy that has been employed by many AFP methods. For example, in CAFA2 (see 1.5.3) homology was the most popular source of information among the compared methods [75]. The general strategy for AFPs is to search for homologs in an annotated database and then to use various enrichment statistics to transfer GO terms (or other annotations) from the k-nearest homologs to the target sequence [108–114]. Reference databases used by AFPs include UniProtKB, RefSeq and Pfam, and search engines employed include BLAST [115], PSI-BLAST [116], HMMER [117] and SANSParallel [118]. Here we discuss profile HMMs and HMMER search engine, which were the methods used in publication (I).

4.2.1 Hidden Markov models and HMMER

Profile hidden Markov models (HMMs) model all of the key evolutionary events that operate on single genes and proteins: substitutions, insertions and deletions. Constructing a profile HMM requires an MSA of the input sequences. The architecture of a profile HMM is constrained to a series of emission, insertion and deletion states [36]. Emission states are hidden states that assign probabilities for observing different residues at different sequence positions. Insertion states model the observed insertion events and deletion states the deletion events. Transition through a given path of hidden states can emit a number of observed sequences with different probabilities. In this way, a given profile HMM infers the probability distribution to the family of sequences that are modelled by that HMM.

In a typical HMM architecture there is one emission or match state for each MSA column that is considered a homology match. Further, there are insertion states between each pair of emission states, which model insertions between matching columns, and two flanking insertions states to model a mismatching head and/or tail of the sequence. Finally, transitions to the deletion states are allowed from the begin state and all match states to model deletions. Once the model architecture is defined, transition and emission probabilities can be estimated from the MSA using maximum likelihood or other methods [24,36].

The scoring used with HMM profiles are the scores returned by the Viterbi, Forward or Backward algorithms [119]. Viterbi returns the probability of best path, the Viterbi path, through the model states to emit the observed sequence. Forward and backward algorithms return the overall posterior probability of emitting the observed sequence [24]. Alternately, these algorithms can return the log-odds scores corresponding to these probabilities.

The most popular implementation for profile HMMs is the HMMER package [41]. Also, databases of profile HMMs representing protein families have emerged. The two major databases of this type are

Pfam [37] and TIGRFAMs [38]. Profile HMM formulation and database collections allow to search for distant homologs of a protein (and nucleotide) sequences.

4.2.2 Enrichment statistics

Gene-enrichment analysis uses enrichment statistics to transfer GO terms from the list of reference genes (e.g. hits from a database search) to the target gene [120]. Here, we want to find the probability of observing at least *k* annotations by term A in a set of *n* reference genes relative to a database of *N* genes with *K* annotations by A [120–122]. This defines the null model: the gene-set and annotation term are independent. In gene-set enrichment analysis the null model is usually specified as either binomial or hypergeometric distribution (equations 4 and 5). These p-values can be assigned as the confidence scores for the predicted GO terms or combined with other information. Statistical test based on hypergeometric distribution is also known as the one-tailed Fisher's exact test [123] (equation 5).

$$\mathrm{P}_{Binomial}(i \geq k) = \sum_{i=k}^{n} \binom{n}{i} \left(\frac{K}{N}\right)^{i} * \left(1 - \frac{K}{N}\right)^{n-i} \tag{4}$$

$$P_{Hyperg}(i \geq k) = \sum_{i=k}^{n} \frac{\binom{K}{i}\binom{N-K}{n-i}}{\binom{N}{n}} \tag{5}$$

4.2.3 Enrichment statistics for gene-clusters

Settings similar to the annotation of individual genes are encountered in the annotation of gene-clusters. Homologs to the reference genes or the profile HMMs are identified in the target genome and various enrichment statistics are calculated for the observed clusters of these genes. This problem is well illustrated by the computational methods for detection of prophages and phage-related sequences. Here we review two articles that are directly relevant to LOCP project (I): Phage_Finder [124] and ProFinder [125].

In both Phage_Finder and ProFinder, phage-related sequences are located by scanning target genomes against annotated databases. These scans retrieve matches to phage sequences or profile HMMs that are referred to as *hits*. ProFinder also implements clustering of hits into phage-like dense regions (PLDRs). PLDRs are located by inspecting every window that starts with a hit and covers between 5 to 300 genes. ProFinder assigns PLDRs p-values from a binomial distribution (equation 4).

Theoretically, a better model for the "hits in a gene-cluster" problem would be a hypergeometric distribution (equation 5). The argument here is that hypergeometric distribution models sampling without replacement while binomial models sampling with replacement. Sampling with replacement does not hold for a bacterial genome that can include only a limited number of genes for each function.

4.2.4 Significance tests for multiple gene-clusters

When a series of gene-clusters is tested for significance, the probability of obtaining at least one false positive, referred as the *family-wise error rate*, exceeds the significance level, *α,* set for the individual tests [123]. This phenomenon is known as the problem of multiple hypothesis testing (see Fig 6).

Several methods have been proposed for controlling family-wise error rate [123]. One of the most statistically powerful of these methods is the Monte Carlo simulation [126].

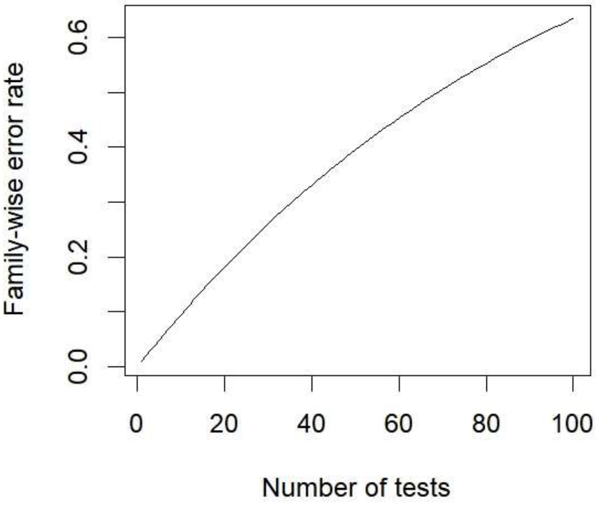


Figure 6 Family wise error rate in x tests. As the number of tests increases the family-wise error rate largely exceeds the significance level, α, set for the individual tests. In this example α = 0.01.

The Monte Carlo method estimates the distribution of p-values directly from a set of simulated datasets for which the null hypothesis is true [126]. In this technique, *S* datasets are generated from a population of datasets that conform to the null hypothesis. Each dataset is assigned a p-value. The distribution of p-values from this null model is then used to estimate the probability of obtaining at least one false positive for any given significance level [126].

4.2.5 Locating Pilus Operons (I)

Publication (I) is an example of how a clearly defined class of genes and gene-clusters can be efficiently and accurately annotated in a bacterial genome. In this project, we focused on pilus operons in gram-positive prokaryotes. However, the same general strategy can be applied to annotate other types of gene clusters. In publication (I) we followed the general ideas outlined in sections 3.2.1-4: we started by locating *hits*, i.e. genes related to pili, then we looked for clusters of hits, assigned p-values to located clusters and adjusted those p-values for multiple hypothesis testing.

To detect genes with pilus-related features we collected HMM profiles from Pfam and TIGRPFAM databases that represent families of sortase enzymes and their recognition sites. In addition, using the HMMER2.0 package we constructed one HMM representing E-box and 4 HMMs representing stabilizing motifs that are characteristic of pilus genes. This collection of HMMs were then searched against the target genome and the matching genes were labeled as hits (I).

In the next step, we detected pilus-related gene-clusters that we named Pilus Like Dense Regions (PLDR). We iterated all genes in the genome clustering hits with at most *gapmax* non-hit intermediate genes. To separate random PLDRs from nonrandom, we selected the hypergeometric distribution as the null model. PLDRs were then assigned p-values using the one-tailed Fisher's exact test.

In the next step, we used Monte Carlo simulations to control family-wise error rate. In more detail, we generated random gene-clusters by sampling random genomes with the same number of genes, *N*, and hits, *K*, as in the target genome (sampling the $\binom{N}{K}$ $space$). These were sampled by generating 1000 random permutations of hit position vector *X*, where X_i is the position of the *i:th* hit. For each simulation, PLDRs were detected and assigned p-values the same way as for the target genome. For each simulation in the 1000 simulation runs we collected the minimum p-value, $p_{i,min}$. These values were then used to assign target PLDRs the family-wise error rates (referred in publication (I) as the adjusted p-values, p_{Adj}).

For more details on the LOCP method please refer to the attached publication (I).

4.3 Evaluation of automated function predictors

4.3.1 Metrics commonly used in comparative studies

Table 4 lists evaluation metrics used in AFP competitions and selected papers with focus on AFP evaluation. We see that many different metrics have been applied to AFP evaluation with no clear consensus on which particular metrics to use. Even in CAFA, evaluation metrics have changed between consecutive challenges: CAFA I started with precision-recall curves and F_{max}, then was re-evaluated with ROC AUC, Lin and Resnik semantics; CAFA II abolished the use of Lin and Resnik semantics and introduced S_{min}; CAFA III introduced term-centric F_{max}. This constant perturbation of metrics indicates an ongoing development of the field, and I believe that a fair consensus on which metrics to use is yet to be reached. In the next section definitions to the commonly used evaluation metrics (EvMs) are given, and their properties elucidated in more detail.

Table 4. ***Overview of some Evaluation Metrics used in AFP literature****. The table presents selected AFP method papers, AFP competition papers and re-evaluations of competitions (re-eval). Semantics refers to semantic similarity measure. Other abbreviations as defined in section 3.3.2. Notice the variety of metrics used.*

Article	Evaluated AFP	Evaluation metric(s)
Martin *et al*., 2004 [127]	GOtcha	Selectivity vs. p-value Coverage vs. p-value
Engelhardt *et al*., 2005 [128]	Sifter	ROC curve Accuracy
Götz *et al.,*2008 [114]	BLAST2GO	Accuracy
Friedberg, 2006 [129]	AFP 2005	Resnik semantics
Hawking *et al*., 2008 [130]	PFP	Schlicker semantics
Wass *et al*., 2008 [131]	ConFunc	Precision-Recall curve
Chitale *et al*., 2009 [132]	ESG	Schlicker semantics Precision and Recall value
Falda *et al*., 2012 [110]	ARGOT2	Prec-Recall curve
Engelhardt *et al*., 2011 [133]	Sifter v2	ROC curve Precision-Recall curve
Minneci *et al*., 2013 [134]	FFPred 2.0	F_{max} Precision-Recall curve SimGIC
Radivojac *et al*., 2013 [70]	CAFA I	F_{max} Precision-Recall curve Weighted Precision-Recall curve
Gillis and Pavlidis, 2013 [78]	CAFA I, re-eval	Precision-Recall curve TC ROC AUC Resnik semantics Lin semantics
Koskinen *et al*., [108]	PANNZER	Weighted Precision-Recall curve Lin semantics
Jiang *et al*., 2016 [75]	CAFA II	F_{max} Precision-Recall curve S_{min} TC ROC AUC
Zhou *et al*., 2019 [76]	CAFA III	F_{max} and TC F_{max} Precision-Recall curve S_{min} TC ROC AUC

4.3.2 Metric properties and definitions

4.3.2.1 Metrics for binary classifiers

AFP methods assign sequences to functional classes. In most cases this can be reduced to a binary classification, that either assigns a sequence to a class (i.e. makes a positive classification) or not (makes a negative classification). In both cases, the assignment can be either true or false relative to a given reference classification that is considered to be the "ground truth". Performance of a binary AFP that predicts class A for a set of gene sequences can be summarized by a 2 X 2 confusion matrix:

Table 5. *Confusion matrix for an AFP method. Here ref is the set of sequences that are known to have function A and pred is the set of sequences predicted to have function A by the evaluated AFP. The confusion matrix divides sequences into four cells. The true and false positives are the correct and false positive predictions, respectively. The true and false negatives are the correct and false negative predictions. For further details see the text.*

AFP\Truth	**x ∈ ref**	**x ∉ ref**
x ∈ pred	true positives	false positives
x ∉ pred	false negatives	true negatives

In this table, the correct positive classifications are referred as the *true positives,* $TP = pred \cap ref$, and the correct negative classifications as the *true negatives,* $TN = \{x \notin pred\} \cap \{x \notin ref\}$. The erroneous positive classifications are referred as the *false positives,* $FP = pred \backslash ref$ and erroneous negative classifications as the *false negatives,* $FN = ref \backslash pred$. Probabilistic AFP methods will assign predictions with a score such as e-value or p-value. For these methods *pred, TP, TN, FP* and *FN* are all a function of a confidence threshold, *th*, at which the predictions are reported. In addition, let *rank(x)* be the rank of prediction *x* assigned by the AFP. Using these definitions, we can give closed-form expressions for many common Evaluation Metrics (EvMs) (see Table 6).

Table 6. ***Defining popular evaluation metrics***. *Abbreviations: pred, annotations predicted by the AFP, ref, annotations that are known to be true (i.e. reference or the ground truth annotations), TP, true positives, TN, true negatives, FP, false positives, FN, false negatives, th, classifier confidence threshold.*

(6)	$\boldsymbol{Accuracy(pred, ref, th) = \frac{TP + TN}{TP + FP + FN + TN}}$
(7)	$\boldsymbol{Precision(pred, ref, th) = \frac{TP}{TP + FP}}$
(8)	$\boldsymbol{Recall(pred, ref, th) = Sensitivity(pred, ref, th) = \frac{TP}{TP + FN}}$
(9)	$\boldsymbol{TPR(pred, ref, th) = \frac{TP}{TP + FN}}$
(10)	$\boldsymbol{FPR(pred, ref, th) = \frac{FP}{FP + TN}}$
(11)	$\boldsymbol{ROC\ AUC(rank, ref, refneg) = \frac{1}{\lvert ref\rvert \lvert refneg\rvert}\left(\sum_{x\in ref} rank(x) - \frac{\lvert ref\rvert(\lvert ref\rvert + 1)}{2}\right)}$
(12)	$\boldsymbol{Specificity(pred, ref, th) = 1 - FPR(pred, ref, th) = \frac{TN}{TN + FP}}$
(13)	$\boldsymbol{F(pred, ref, th) = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{2 \times TP + FP + FN}}$
(14)	$\boldsymbol{F_{max}(pred, ref) = \max_{th} \frac{2 \times TP}{2 \times TP + FP + FN}}$
(15)	$\boldsymbol{Jacc(pred, ref, th) = \frac{\lvert pred \cap ref\rvert}{\lvert pred \cup ref\rvert} = \frac{\lvert TP\rvert}{\lvert TP + FP + FN\rvert}}$
(16)	$\boldsymbol{Jacc_{max}(pred, ref) = \max_{th} \frac{\lvert TP\rvert}{\lvert TP + FP + FN\rvert}}$

EvMs defined in equations 6-16 are discussed below more in detail.

Accuracy is defined as the proportion of correct classifications *(TP* + *TN)* to all classifications *(TP* + *TN* + *FP* + *FN).* Accuracy is a somewhat rough evaluation metric, because it does not distinguish between false positive and false negative errors. Another shortcoming of this metric is that it requires a clear definition of the true negatives. Functional annotations of genes often lack any meaningful estimates for the number of true negative annotations (see supplementary text in (III) for further discussion).

Precision (also known as the *positive predictive value*) is defined as the proportion of true positives *(TP)* to all positive classifications *(TP* + *FP)*. *Recall* (also known as *sensitivity*) is defined as the proportion of true positives *(TP)* to all ground truth positives *(TP* + *FN)*.

For a probabilistic AFP, precision can be plotted against recall at different thresholds, *th*, as a precision-recall curve. A limited number of precision-recall curves can be plotted on the same figure to compare different AFPs, but this is an inexact method. To compare several AFPs we generally need a single scalar evaluation metric. In the MouseFunc project, that evaluated AFPs for mouse genes, precision was assessed at several fixed recall values [135]. A more general scalar metric is the *area* *under the precision-recall curve* [136], which we refer to as *AUC-PR*. One attractive property of precision-recall analysis is that it does not depend on true negatives (III).

True positive rate (TPR) is equivalent to recall and sensitivity: it is the number of true positives *(TP)* relative to all ground truth positives *(TP + FN). False positive rate (FPR)* is the number of false positives *(FP)* relative to the number of ground truth negatives *(FP + TN).*

In Receiver Operating Characteristic analysis (*ROC analysis*), *TPR* is plotted against *FPR* at different thresholds, *th*. Area under the ROC curve, *ROC AUC*, is a scalar metric derived from ROC analysis. AUC is closely related to the Mann-Whitney U-statistic and is an estimate of the probability that a binary classifier will rank positive prediction higher than a negative prediction [137]. Let *ref* represent our positive class for term A, namely all genes that are annotated with A according to our reference. Let $refneg = \{x \notin ref\}$ represent our negative class and let *rank(x)* define the rank of sequence *x* as returned by the AFP. Then ROC AUC is defined by equation 11.

Note that ROC AUC metric requires ranking of both ground truth positives and negatives. This can lead to ambiguity, because, in the case of assigning genes to functional terms, there is no clear definition for ground truth negatives. As we discussed in publication (III) true negatives can be defined as all genes in the reference that do not have function A. Or these can be all genes in the reference database that do not have function A. Or these can be all genes for the annotated organism that do not have this function, etc.

Specificity or the *true negative rate* is the number of true negatives *(TN)* relative to the number of all ground truth negatives *(TN + FP)*. Again, this metric depends on the number of true negatives which are often undefined.

F-score is the harmonic mean of precision and recall. For probabilistic AFPs, a scalar metric called F_{max}*-score* can be defined as the maximum of *F-score* values across all thresholds, *th*.

Jaccard index or *Jaccard coefficient* is the number of annotations common to predicted and reference sets divided by the number of annotations in the joined set. For probabilistic AFPs it is convenient to use maximum Jaccard index over thresholds, *th*.

4.3.2.2 Semantic similarity metrics

Terms near the root of GO are general terms that convey little information and terms near the leaves of GO are more specific terms that convey more information. At the same time more general terms have higher prior probabilities (i.e. larger class size) than more specific terms. This causes the naïve prediction of the more general non-informative GO terms to perform well in AFP comparisons [70,75]. This problem can be counteracted by metrics that consider the *information content* of the predicted GO terms. Information content was defined by Resnik as the negative log-likelihood of a given term *y* in a given annotation corpus [138] (equation 17 in Table 7). Thus, the more specific terms have higher information content and the more general terms have lower information content.

Furthermore, terms in gene ontology are related semantically. This implies that information conveyed by the neighboring terms is lost in evaluation that rigidly divides terms into correct and erroneous. Therefore, metrics that consider the size of the assigned GO classes and/or semantics are likely to be more powerful in identifying truly informative AFPs [139,140].

Before introducing semantic metrics, we need to extend our notation. Let *i*-index refers to all annotations for gene *i*, and *j*-index to all annotations with term *j*. For example, $pred_i$ are predictions for gene *i*, ref_i are reference annotations for gene *i*.

4.3.2.3 Semantic similarity for pairs of GO terms

The *Most Informative Common Ancestor (MICA)* for nodes *x* and *y* is the common ancestor of nodes *x* and *y* that has the highest information content (equation 20 in Table 7). The ancestral nodes of x, *A(x)*, denote the set of all ancestors in GO for node *x*. Using ancestral nodes and MICA we can define ancestral Jaccard index or AJacc (**III**), Resnik [138] and Lin [141] (equations 21-23 in Table 7). Note that these semantic similarity metrics define pairwise similarities and do not work as such for comparing sets of GO terms.

The set of predicted GO terms can be scored against the correct set by considering all pairwise comparisons. This results in a similarity matrix, SIM, of pairwise scores (equation 24). In this notation, rows stand for predicted and columns for correct GO terms, *sem* can be any semantic similarity metric (e.g. *Resnik*).

When semantic similarity metrics are used to evaluate AFPs, the SIM matrix needs to be converted to a scalar metric. As there are no clear recommendations for this, in publication (**III**) we have considered six alternatives for the function *S*, that converts SIM to a single score:

A. Mean of matrix. This is the overall similarity between all classes in $pred_i$ and ref_i.
B. Mean of column maxima. This is the average of best hits in $pred_i$ for classes in ref_i.
C. Mean of row maxima. This is the average of best hits in ref_i for classes in $pred_i$.
D. Mean of B and C.
E. Minimum of B and C.
F. Mean of concatenated row and column maxima of SIM.

Method A is the all-pair arithmetic average proposed by Lord [142]. Methods B, C and D have been proposed previously by several authors [143–146]. Method B is inherently weak at monitoring false positives, because for each term in the reference set only the best match in the predicted set is used. Method C is weak at monitoring false negatives, because terms in the reference set that have no match in the predicted set are not penalized in any way. D aims to correct B and C but is still sensitive to outliers. Therefore, we proposed novel methods E and F as improvements to method D (**III**). E monitors the weaker of B and C and is therefore monitoring both false positives and false negatives. F combines two vectors used for B and C.

Finally, the score returned by methods A to F is a function of the gene and threshold value. These can be converted to a scalar value by using the mean value across genes, *i*, and the max value across thresholds, *th*. The overall definition for an evaluation metric based on a pairwise semantic similarity is given in equation 25 in Table 7. Here, *sem* can be any semantic similarity function, such as Resnik, Lin or AJacc, and *S* is one of the six summation functions outlined above.

Table 7. *Semantic similarity metrics for pairwise comparison of GO terms. Notations are explained in the text.*

(17)	$\boldsymbol{ic_{Node}(x) = log\frac{1}{p(x)}}$
(18)	$\boldsymbol{ic1(G) = \sum_{y\in G} log(\frac{1}{p(y)})}$
(19)	$\boldsymbol{ic2(G) = log(\frac{1}{\prod_{y\in G} p(y\mid R(y))}) = \sum_{y\in G} log(\frac{1}{p(y\mid R(y))})}$
(20)	$\boldsymbol{MICA(x,y) = \underset{z\in A(x)\cap A(y)}{argmax}\ ic_{Node}(z)}$
(21)	$\boldsymbol{AJacc(x,y) = \frac{\lvert A(x) \cap A(y)\rvert}{\lvert A(x) \cup A(y)\rvert}}$
(22)	$\boldsymbol{Resnik(x,y) = ic_{Node}(MICA(x,y))}$
(23)	$\boldsymbol{Lin(x,y) = \frac{2 \times ic_{Node}(MICA(x,y))}{ic_{Node}(x) + ic_{Node}(y)}}$
(24)	$\boldsymbol{SIM(pred_i, ref_i, sem) = \begin{bmatrix} sem(pred_i(1), ref_i(1)) & \dots & sem(pred_i(1), ref_i(n)) \\ \dots & \dots & \dots \\ sem(pred_i(m), ref_i(1)) & \dots & sem(pred_i(m), ref_i(n)) \end{bmatrix}}$
(25)	$\boldsymbol{EvM(pred, ref, sem, S) = \max_{th} \frac{1}{genes(th)} \sum_i^{genes(th)} S(\ SIM(pred_i, ref_i, sem)\ )}$

4.3.2.4 Semantic similarity for sets of GO terms

Pesquita et al. introduced Jaccard index weighted by the information content, which they referred to as the SimSIG metric [143] (equation 26 in Table 8).

Clark and Radivojac extended the definition of information content for individual GO terms to information content for GO subgraphs [18]. Let *R(y)* denote the set of immediate ancestors of term *y*. Then the likelihood of subgraph *G* can be factorized as a product of conditional probabilities *p(y|R(y))*. Information content of the subgraph *G* is then a sum of negative log-likelihoods of its component nodes (equation 19 in Table 7).

Clark and Radivojac then defined *remaining uncertainty*, *ru*, as the mean gene-wise information content of the *FN* set and *misinformation*, *mi*, as the mean gene-wise information content of *FP* set. Based on these, they defined a scalar metric S_{min} [140] (equations 27-29 in Table 8).

Table 8. *Semantic similarity metrics for sets of GO terms. Notations are explained in the text.*

(26)	$\boldsymbol{SimSIG(pred, ref, th) = \max_{th} \frac{1}{n} \sum_{i}^{n} \frac{ic1(TP_i)}{ic1(TP_i)+ic1(FP_i)+ic1(FN_i)}}$
(27)	$\boldsymbol{ru(pred, ref, th) = \frac{1}{n} \sum_{i}^{n} ic2(FN_i)}$
(28)	$\boldsymbol{mi(pred, ref, th) = \frac{1}{n} \sum_{i}^{n} ic2(FP_i)}$
(29)	$\boldsymbol{S_{min}(pred, ref) = \min_{th} \sqrt{ru(pred, ref, th)^2 + mi(pred, ref, th)^2}}$

4.3.3 Artificial Dilution Series (III)

The CAFA challenges have demonstrated discrepancies in AFP rankings and possible biases in AFP evaluation (see section 1.5.4). Many of these issues can be addressed by selecting appropriate evaluation metrics or by adopting the existing metrics to the challenges of AFP evaluation. To aid metric selection and development we presented a novel method called Artificial Dilution Series (ADS, publication III). Our approach uses existing GO annotation data to generate a series of GO annotation datasets with different levels of correctness (referred as signal). These datasets are then applied to evaluate candidate evaluation metrics (EvMs) in two separate tests.

In our first test, scores for the tested EvM are calculated for datasets at different signal levels and these scores are compared to the signal values revealing discriminative properties of the metric. We refer to this test as the Rank Correlation test (*RC test*). Results from the RC test can be examined visually by plotting EvM scores as a series of boxplots, one boxplot for each signal level (for an example see Fig 4 in III). A metric that is discriminative will have compact and monotonically declining boxplots. Results are also summarized numerically by the *RC score*: the rank correlation between EvM scores and the signal level.

In our second test, scores for the tested EvM are calculated for several false positive datasets and these are contrasted with scores for the ADS series. This second test is designed to reveal systemic biases in the tested EvM. We refer to this test as the False Positive test (*FP test*). Results can be visualized by plotting EvM scores for the tested false positive datasets on the boxplots from the first test. High quality EvMs are expected to assign false positive datasets scores that are close to scores from dilution series sets with zero signal. This second test also assigns a numerical *FP score* to each EvM: the approximated signal level of the artificial datasets with similar EvM scores (we select the maximum from all evaluated false positive datasets).

These two tests, the RC and the FP tests, define an orthogonal scoring system against which EvMs can be evaluated and compared. In publication (III) we argued that both tests are required in order to screen for discriminative and unbiased EvMs. For more details on the ADS method please refer to the attached publication (III).

4.3.3.1 Evaluated metrics and datasets

We compared representatives from various types of EvMs commonly used in comparative studies for AFPs (see section 3.3.1). The tested EvMs were grouped into three families: rank-based, semantic similarity-based and group-based metrics. We evaluated two rank-based metrics: area under ROC curve (AUC-ROC) and area under precision-recall curve (AUC-PR). From metrics based on semantic similarity, we evaluated Lin, Resnik and a novel AJacc metric. From group-based methods, we evaluated SimGIC and S_{min}. We also evaluated F_{max}, one of the most popular evaluation metrics in machine learning.

Most EvMs defined in section 3.3.2 can be further modified by using the same core function with different data structuring. By data structuring we refer to the gene centric, term centric or unstructured evaluations (III). In Gene Centric evaluation (GC) we evaluate the predicted set of GO terms against the true set separately for each gene and then summarize these values with the mean over all genes. In Term Centric evaluation (TC) we compare the set of genes assigned to a given GO term against the true set separately for each GO term and take the mean over all GO terms. In UnStructured evaluation (US) we compare predictions as a single set of gene–GO tuples disregarding any grouping by shared genes or GO terms. In total, we tested 37 different evaluation metrics (III).

All metrics were tested on three annotation datasets: the evaluation set used in CAFA I [70], the evaluation set from the MouseFunc competition [135] and a random sample of 1000 well-annotated genes from the UniProt database (ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UniProt/, 01.2019). Annotations in CAFA I and MouseFunc datasets were updated with contemporary GO term annotations from UniProt (ftp://ftp.ebi.ac.uk/pub/databases/GO/goa/UniProt/, 01.2019). The selected datasets varied in the annotation depth and density which allowed us to test metric performance in different settings of AFP evaluation.

4.3.3.2 Comparison with previous research

To my knowledge, there has been little research on evaluation metrics with Gene Ontology (GO). Clark and Radivojac compare seven methods by looking at thresholds that optimize each evaluation metric, and explore if the selected thresholds are rational for classification purposes [140]. GO Semantic similarities [143] have been evaluated using correlation against other datasets, e.g. sequence similarity [147], but their performance as the classifier evaluation metrics has not been thoroughly tested. These reports lack quantification of metric performance and do not provide common reference points that are needed for the comparison of various evaluation metrics. Contrary to these, ADS framework defines two quantitative tests that serve as a reference against which any metric can be evaluated and compared (III).

In the machine learning field, a large and thorough comparison by Ferri et al. [148] tested a large number of evaluation metrics under different challenging situations with artificial datasets. Sokolova and Lapalme discussed how measures are invariant towards the changes in the classifier results [149]. Seliya et al. and Ferri et al. compared similarities between evaluation measures [148,150]. These comparisons, however, used mainly artificial datasets and focused on a few challenging features at a time. Real data, on the other hand, tends to include combinations of challenging features.

Furthermore, the classification structures, used in previous machine learning articles, differs significantly from Gene Ontology (GO). These articles discuss almost solely either binary or multinomial classification tasks, where each item cannot be correctly classified to many classes simultaneously. Assignment of GO terms to biological sequences is a multiple binary classification task, where each item can correctly belong to many classes or might not belong to any of the available classes. In addition, GO prediction is further complicated by the correlations, created by the hierarchical structure of GO. All these specific properties of GO classification create evaluation challenges that are uncommon in machine learning. The ADS framework addresses these challenges by working with GO data, and by conserving complex correlations present in real-life GO annotations.

5. RESULTS AND DISCUSSION

5.1 LOCP results (I)

To evaluate the performance of LOCP, we manually compiled a benchmark of 20 complete gram-positive bacterial genomes with gene-clusters conforming to one of the 10 pilus operon types that had been described at the time of publication. Our benchmark contained 28 pilus operons, 20 stains and 4 species. Notably, all four bacterial species in our benchmark are pathogenic and have clinical importance: *Corynebacterium diphtheria*, *Streptococcus agalactiae*, *Streptococcus pneumoniae* and *Streptococcus pyogenes*.

We ran LOCP on the benchmark assigning all genes an adjusted p-value. Genes labeled as hits were assigned the p-value of the corresponding PLDR and all other genes a p-value of 1. We then performed a standard ROC analysis (see 3.3.2). The ROC AUC was above 99% indicating almost perfect recognition of the pilus operons in our benchmark.

We also used LOCP to identify pilus-related gene-clusters in all complete gram-positive genomes available at the time of publication (August 2008). From 181 genomes analyzed, 67 (37%) were predicted to have at least one putative pilus operon. Altogether, 98 PLDR were located including 20 PLDRs corresponding to known pilus operons listed in the benchmark set and 78 PLDRs that, to our knowledge, had not yet been described. Notably, *Corynebacterium diphtheriae* had 3 PLDRs and *Clostridium perfringens* had 4 (I).

LOCP analysis was repeated in October 2017 for complete genomes of gram-positive phyla: *Actinobacteria, Firmicutes*, *Tenericutes* and *Chloroflexi*. Here we found pilus-related gene-clusters in 5306 (18%) out of 29507 gene assemblies. Table 9 lists a selection of bacterial species with identified PLDRs that are known pathogens or are otherwise significant to human health. From Table 9 we see that PLDRs are common in gram-positive pathogens, commensals and probiotic species. PLDRs were found in several significant pathogens including *Corynebacterium diphtheriae*, *Listeria monocytogenes*, *Streptococcus pneumoniae* and *S. pyogenes*. For *Listeria monocytogenes* we identified over 800 assemblies with PLDRs and individual strains with up to 4 PLDRs. PLDRs were identified in several nosocomial pathogens associated with hospital infections and multi-drug resistance: these included *Enterococcus faecalis*, *Enterococcus faecium* and *Staphylococcus aureus*. For *E. faecium* and *S. aureus,* we identified hundreds of assemblies with PLDRs. Finally, PLDRs were identified in several members of *Bifidobacterium* and *Lactobacillus* genera. These are important commensals in human GI tract and are used to produce fermented foods and/or as probiotics.

To summarize, PLDRs seem to be associated with significant interactions with the host and these interactions can take both pathological and mutualistic form.

All PLDR predictions are available on LOCP website: http://ekhidna2.biocenter.helsinki.fi/LOCP/.

Table 9. *Gram-positive bacteria with PLDRs predicted by LOCP analysis (2017). Assem, number of genome assemblies with at least one PLDR, PLDR, number of PLDRs per genome, GI, gastrointestinal, OC, oral cavity, UG, urogenital. The table includes a selection of significant pathogens, zoonotic species, common commensals and probiotic species.*

Species	Assem	PLDR	Significance	Ref
Actinomyces naeslundii (oris)	17	1-2	Human OC commensal Periodontal disease and tooth decay	[151]
Bacillus cereus	14	2-3	Food contamination leading to GI symptoms e.g. diarrhea	[152]
Bifidobacterium adolescentis	18	1-4	Human GI commensal Fermentation in food industry; Probiotic	[153]
Bifidobacterium bifidum	14	1-2	Human GI commensal; Probiotic	[153]
Bifidobacterium breve	12	1-2	Human GI commensal; Probiotic	[153]
Bifidobacterium dentium	2	3	Human pathogen causing tooth decay	[154]
Clostridioides difficile	21	1	Human GI commensal Toxic strains cause colitis and diarrhea	[155]
Clostridium botulinum	16	1-2	Foodborne and wound botulism in humans and livestock	[156]
Clostridium perfringens	38	1-3	One of the leading causes of food poisoning in humans and domestic animals	[157]
Corynebacterium diphtheriae	83	1-3	Human pathogen causing diphtheria	[158]
Corynebacterium jeikeium	15	1-2	Opportunistic multiresistant nosocomial pathogen	[159]
Corynebacterium striatum	9	1-3	Opportunistic nosocomial pathogen	[160]
Entercoccus faecalis	81	1-2	Human GI commensal Multiresistant nosocomial infections Endocarditis, meningitis, sepsis	[161]
Enterococcus faecium	443	1-4	Human GI commensal Multiresistant nosocomial infections Meningitis and Endocarditis in neonates	[161]
Eubacterium sp.	15	1-4	Associated with bacterial vaginosis	[162]
Lactobacillus sp.	233	1-3	Human GI and UG commensals Fermentation in food industry; Probiotic	[163, 164]
Listeria monocytogenes	826	1-4	Over 2000 listeriosis cases yearly in EU, high fatality rates GI symptoms, septicaemia, CNS sequelae	[165, 166]
Propionibacterium freudenreichii	10	1-2	Probiotic; Cheese production	[167]
Staphylococcus argenteus	102	1-3	Human pathogen; Alpha-haemolytic	[168]

Staphylococcus aureus	515	1-3	Opportunistic nosocomial pathogen Methicillin-resistant strains (MRSA) Skin lesions, sinusitis, meningitis, pneumonia, endocarditis, sepsis	[169]
Staphylococcus hyicus	1	2	Animal pathogen; Zoonotic Skin diseases in livestock	[170]
Streptococcus canis	1	2	Opportunistic pathogen in dogs and cats; Zoonotic Skin infections, sepsis, abortions	[171]
Streptococcus oralis	18	1-3	Human OC commensal and opportunistic pathogen	[172]
Streptococcus parasanguinis	10	1-2	Human OC commensal Participates in plaque formation	[173]
Streptococcus pneumoniae	592	1-2	Community acquired pneumonia, meningitis Penicillin-resistant strains (PRSP)	[174, 175]
Streptococcus pyogenes	90	1-3	Group A Streptococci (GAS) pathogen Pharyngitis, skin lesions, cellulitis, necrotizing fasciitis, rheumatic fever, glomerulonephritis, other	[176]
Streptococcus suis	74	1-3	Infections in domestic pig; Zoonotic	[177]
Trueperella pyogenes	3	1-4	Opportunistic pathogen in livestock Common cause of mastitis in cattle and pyometra in dogs	[178]

5.2 Modular MSA results (II)

5.2.1 Comparing pairwise alignments and guide trees

In our first test, we evaluated the contribution of different pairwise information sources to the MSA quality. Our results showed that both local and global pairwise alignments are required to construct high quality MSA (II). Similar results were previously reported using the T-Coffee aligner on BAliBASE [48]. Adding GTG motif information had a minor effect on the MSA quality increasing quality scores by an average of 1% for BAliBASE and SABmark.

In the second test, we compared methods for the construction of guide trees. We found that single linkage clustering was the best option for all reference sets in BAliBASE and SABmark. Contrary to our expectations, this method produced more accurate alignments than commonly used neighbor joining or UPGMA (II). However, we note that these results were obtained for alignments without consistency transformation or iterative refinement. It is likely that the aforementioned techniques would render the impact of the guide tree to be less significant.

5.2.2 Comparing consistency and clique transformations

In our third test, we compared consistency transformations. The best option was either Triplet$_{\text{SeqAn}}$ or Triplet$_{\text{SeqAn}}$ repeated twice for all five references in BAliBASE and in the overall evaluation. The second best option was either Triplet$_{\text{T-Coffee}}$ or the MaxFlow method (II). Clique transformation decreased the MSA quality. Applying clique transformation after Triplet$_{\text{SeqAn}}$ did not improve MSA quality (II).

The concept of consistency transformation was introduced by Notredame et al. [48], who also demonstrated superior performance of this strategy on BAliBASE. The utility of this technique was then recognized by many authors and applied in many MSA implementations with reported superior performance [50,52,65,66]. Also, in independent comparative studies, ProbCons, MAFFT-linsi, ProbAlign and Mummals, which all implement consistency transformation, appear as top performers (see Table 1).

Better results with Triplet$_{\text{SeqAn}}$ compared to Triplet$_{\text{T-Coffee}}$ demonstrated that introducing new edges necessary for consistency does improve MSA quality. This is in agreement with previous reports [94]. Improvement with MaxFlow demonstrated the utility of transitive homology links for MSA alignments. Previously, MaxFlow scoring was applied to recover distant homologs [179], but was never before tested on MSA benchmarks. MaxFlow implements consistency transformation by weighting edges in the alignment graph (i.e. homology links) by the number of common neighbors divided by the number of all neighbors. Links between distant homologs are then introduced by considering all possible pathways through the alignment graph and by weighting these links by the weakest pairwise link in the path (II). Clique transformation also considers all possible pathways in the alignment graph, but keeps pairwise weights from the original graph, i.e. no consistency measure is introduced. Thus, our results demonstrate that transitive homology links improve alignments, but only if these links incorporate consistency.

5.2.3 Comparing strategies for iterative refinement

The tree-dependent iterative refinement was the best iterative refinement method for improving MSA quality, although, all tested iterative refinement strategies improved MSA quality. Random tree-based partitioning (Tree$_{\text{Random}}$) and breadth-first partitioning (Tree$_{\text{BF}}$) yielded similar alignment quality (Tables 5 and 6 in (II)). Furthermore, performance of these iterative refinement strategies was comparable to the best consistency transformation, Triplet$_{\text{SeqAn}}$, in both alignment quality and execution time (Tables 5 and 6 in (II)). The random partitioning, although yielding some improvement, was clearly inferior to both tree-based strategies and the consistency transformation (II).

The notion that tree-dependent partitioning and consistency transformations can be equally accurate and equally fast is a novel finding. Previous MSA implementations have either included only the iterative refinement or combined refinement with consistency transformation without directly contrasting these two strategies. For example, in Muscle [46] tree-dependent partitioning is the main strategy for enhancing alignment accuracy. In comparative studies, Muscle had almost invariably lower accuracy than aligners implementing consistency transformation (Table 1). However, these differences do not necessary stem from superiority of the consistency transformation over iterative refinement (i.e. it may be due to different gap models, substitution scores etc.). ProbCons [50] and MSAprobs [52] also implement iterative refinement, however, here it is an optional step, and the main strategy for improving alignment quality is the consistency transformation. MAFFT (version 6 and later) implements tree-dependent iterative refinement, but again, it is inseparably combined with consistency scores and thus cannot be contrasted to the consistency strategy [66].

In summary, previous implementations do not support comparison of the iterative refinement to the consistency transformation while comparisons of programs that implement one or the other strategy is ambiguous. In this respect, publication (II) was the first study where these two strategies were directly compared. Additionally, we found that random iterative refinement is clearly a weaker option than the tree-dependent partitioning. This might be of significance since certain high quality aligners, such as ProbCons and MSAprobs, implement random partitioning. Changing partitioning strategies for these aligners might further improve their accuracy. Based on our results we can recommend the tree-dependent iterative refinement as a simpler but worthy option for the consistency transformation.

5.2.4 The best strategy and the best MSA software

The overall best strategy for our modular aligner was to use global and local pairwise alignments complemented with GTG motifs as input information; to apply Triplet$_{\text{SeqAn}}$ consistency transformation; and to align the sequences using a single linkage guiding tree (II).

Comparison of the popular MSA aligners and the best MMSA strategy showed the importance of consistency transformation and the tree-dependent iterative refinement. The top four most accurate aligners were consistently the same for all three benchmarks: the most accurate aligner was MSAProbs, followed closely by ProbCons, T-Coffee and MAFFT. This ranking is in good agreement with previous comparative studies (see Table 1). The accuracy margin for these top four aligners was quite narrow (see Tables 7 and 8 in publication II). Our own best strategy fell into the same accuracy ranges: level with T-Coffee for BAliBASE and just below the T-Coffee for SABmark (II).

These results demonstrate that the best contemporary multiple sequence aligners operate within a narrow accuracy margin. Furthermore, all top scoring aligners perform consistency transformation and the very best also the iterative refinement. Our own modular implementations of the consistency transformation and the iterative refinement also produced comparable MSA quality.

Our further attempts to improve alignments by introducing transitive homology links did not produce the anticipated results. We tested repeated consistency transformation, MaxFlow and clique transformations and adding GTG motifs to the MSA scoring function. Although the GTG motifs, MaxFlow and clique transformation did produce some improvements for subsets in BAliBASE, these were minor and not consistent.

Also, by varying different components of the general progressive framework, we were not able to induce improvements in MSA quality over the top-ranking methods. However, we note that this study does not cover all known components and alternatives for the progressive alignment strategy. For example, we did not cover different options for sequence weighting, substitution matrices, gap models and conserved homology blocks. Alignments based on HMM formulation and statistical consistency were also out of scope. Thus, there are still plenty of options and combinations that might prove to be fruitful in improving alignment accuracy.

5.2.5 Method rankings and recommendations

According to our results, the most accurate aligner (on all three benchmarks) was MSAProbs, followed closely by ProbCons and MAFFT (II). The three fastest aligners were Kalign, ClustalW and ClustalO, although we do not recommend using Kalign or ClustalW due to the overall low accuracy of these aligners. The fastest among the most accurate aligners was MAFFT, which was up to six times faster than MSAProbs on the BAliBASE benchmark. Based on these results, we recommend MSAProbs for user cases that require the best possible MSA quality and are not limited in execution time. For producing fast alignments and for aligning large sequence sets we recommend using ClustalO. Finally, for the best compromise between speed and accuracy, we recommend using MAFFT.

5.3 ADS results (III)

We compared 37 evaluation metrics for GO classifiers using ADS on three different datasets. The results showed that many of these metrics are extremely biased and must not be used for evaluation of GO AFP methods. The results also identified a single metric that performed well on all datasets and a set of high-quality metrics that showed dataset-dependent performance.

Our first observation was that metric performance varied drastically in both ADS and FPS tests. This is illustrated in Figure 4 in publication (III), which compares *US AUC-ROC*, F_{max}, S_{min}, *Resnik A*, *Resnik D* and *Lin D*. We see a clear separation across ADS signal levels in the boxplots for F_{max} (RC = 0.982) and S_{min} (RC = 0.985), and the next best separation for *US AUC-ROC*. All three semantic similarity measures (*Resnik A, Resnik D* and *Lin D*) performed poorly in the ADS test: *Resnik* and *Lin* boxplots have wide interquartile ranges and low RC scores. The *US AUC-ROC* failed in the FP test: it ranks the FP sets as equally good as ADS sets at *signal = 1*. Note, that FP sets do not convey any real information, but are designed to reveal biases in EvMs. Also, F_{max} had low performance in the FP test. From these metrics only S_{min} had good performance in both tests.

5.3.1 Comparing area under the curve metrics

In Figure 5 in publication (III) we compared the performance of different AUC metrics. These results demonstrate that unstructured and gene-centric ROC-AUC metrics have high correlation with signal level, but fail in FP tests (top row in Fig 5 in (III)). This is somewhat expected, because AUC has been criticized for being a noisy metric with sensitivity to sample size and class imbalance [150,182]. Still, variations of AUC metric have been used extensively in AFP evaluation [70,75,78,128,133]. Our results clearly demonstrate that the only unbiased version of ROC-AUC is the term-centric ROC-AUC and that any other versions should not be used for AFPs predicting GO terms (i.e. GO classifiers). We also tested precision-recall AUC (PR-AUC), which, to our knowledge, has never previously been applied to GO classifiers. Our results showed that PR-AUC (bottom row in Fig 5 in (III)) consistently outperformed ROC-AUC in ADS correlation scores. Notably, unstructured and gene-centric versions of PR-AUC showed less profound but similar bias to the ROC-AUC metrics and should not be used for GO classifiers.

5.3.2 Improving semantic similarity metrics

We compared the following GO semantic similarities: Resnik, Lin and Ancestor Jaccard (AJacc), each coupled with six different semantic summation methods, A-F (III). Our results revealed that summations methods have a much stronger influence on metric quality than the core semantic similarity function (see Fig 6 in (III)). Differences between summation methods A to F were drastic. Summation methods A (matrix mean) and C (mean of row maxima) failed in the RC test and methods B (mean of column maxima) and D (mean of B and C) failed in the FP test. The only methods that performed well in both tests were the novel methods introduced in this study: methods E (minimum of B and C) and F (mean of concatenated row and column maxima). Differences between core semantic functions were less prominent, however, metrics based on information content (Lin and Resnik) seemed to have better performance.

We also tested the unstructured and gene-centric variations of SimGIC, S_{min} and Jacc (III). All of these demonstrated very high correlation to the ADS signal level (high RC score) and reasonable performance in the FP tests (low FP score). Unstructured versions outperformed the gene-centric versions in RC test, FP test or both (III).

5.3.3 Making clear recommendations

Based on our results we were able to recommend which metrics are well suited for AFP evaluation and which are clearly not (see Table 1 in (III)).

First, we discuss metrics that showed poor performance. Notably, all of these metrics have been used in AFP research, and some of these metrics have been widely used (see Table 4). Based on the very high FP scores of US and GC versions of both ROC-AUC and PR-AUC, these metrics should never be used in AFP evaluation. Although F_{max} is preferred for its simplicity, our results demonstrate that F_{max} is also biased for naïve predictions (high FP score). Also, in CAFA I F_{max} ranked the naïve method above 7 out of 10 (for "easy targets") and above 9 out of 10 (for "difficult targets") top performing AFPs in the task of assigning molecular function-related GO terms [70]. Additionally, CAFA rankings based on F_{max} differ significantly from rankings based on S_{min} (see Table 2), a metric that we found to be reliable. Thus, all the mounting evidence indicates that F_{max} is biased and should not be used in AFP evaluation. We also recommend avoiding summation methods A (matrix mean), B (mean of column maxima), C (mean of row maxima) and D (mean of B and C) in EvMs based on pairwise semantic similarity.

The metrics that showed good performance were the weighted Jaccard (*SimGIC*), S_{min} and term-centric (TC) versions of AUC. The most consistent metric with high RC and FP values across all datasets was *US SimGIC*, which we recommend as the most stable high-quality metric. We also give a list of potentially recommended metrics, which showed high performance on one or two datasets: *TC AUC-PR* and *TC AUC-ROC*, *Lin E*, *Resnik E* and S_{min}. We note that our results indicate options for further improvements for many of these metrics.

5.3.4 Developing metrics with ADS

Results for *ROC-AUC* and F_{max} demonstrate that ADS can quickly and efficiently locate shortcomings in GO evaluation metrics. The core ADS library is written in C++ and is quite fast, allowing us to run conclusive tests within a single day. This creates a platform for fast development of metrics based on their performance. Using ADS, we were able to experiment with different variations of metrics that, based on our theoretical understanding, could potentially eliminate bias for false positive predictions or improve rank correlation. Comparing variations that showed good and bad performance in rank correlation and false positive tests often pinpointed theoretical explanations for these differences.

The numerous positive findings reported in publication (III) demonstrate that the tandem application of practical and theoretical metric development approaches is very efficient. Using ADS, we designed the novel *PR-AUC* metric and showed that its term-centric version has very good discriminative properties as well as stability across different datasets and no bias (Fig 5 in (III)). Additionally, we were able to design novel summation methods E and F for the family of semantic similarity metrics and to demonstrate that these were superior to previous standards A and D (Fig 6 in (III)). Finally, we demonstrated that for metrics based on information content, the unstructured versions had better discriminative performance (III). Thus, this work also provides a general framework for developing and improving evaluation metrics.

5.3.5 Signal and noise models

ADS implements a combination of error and signal models that might be the focus of future studies. The current signal model is based on a set of correct GO annotations. From these annotations, a random fraction of GO terms is rotated randomly in the small space of 2 to 4 ancestral nodes. The current error model represents hard-to-distinguish and misinformative errors: GO terms that have no semantic relationship to the correct terms and that are assigned scores from the same distribution as the correct GO terms. A legitimate question arises: Does this signal and error models represent real life AFPs? This question can be approached by collecting AFP predictions from the literature or large scale competitions such as CAFA. This data could be analyzed to elucidate statistical and other properties of common AFPs. This analysis may discover novel information about biases and challenges involved in the comparative evaluation of real life AFPs.

Another question that arises is whether it is necessary to model signal and noise in real life AFPs in order to perform informative EvM evaluation? The key idea here is that modelling the “simplest case” of all possible AFPs might be sufficient to spot the high quality and low quality metrics. This is the approach taken in the ADS project. The current ADS implementation checks metric performance on a signal model that is very close to the ground truth reference and an error model that is very far from the reference. We argue that if a metric does not work for this simple case, then it is also likely to fail on AFPs that produce more subtle errors or signal that is more distant to the ground truth. Refining and justifying this “simplest case” approach is another possible way of developing the ADS project. In general, the ADS approach can encompass various models for signal and error.

6. CONCLUSIONS AND PERSPECTIVES

This thesis has focused on comparative studies for homology-based methods. These were addressed in three articles, one introducing a novel annotation method and two others introducing frameworks for comparative evaluation.

Publication (I) presented a novel method for detecting pilus operons in bacterial genomes. This illustrated the application of homology-based methods for the annotation of biological sequences. Using the developed LOCP method, we were able to retrieve 5306 putative pilus operons from the available whole genomes of monoderm bacteria. The vast majority of these operons were novel findings, and many were located in bacteria that are known to be significant human pathogens, commensals or probiotics. The enrichment of species that have significant human interactions among those predicted with pilus operons indicates that LOCP has notable potential for contributing valuable information for ongoing and future bacteriological and medical studies.

Publication (II) addressed the topic of comparative evaluation of MSA methods. Particularly, we were interested in revealing which alignment strategies are the top performers and how these relate to the state-of-the-art MSA implementations. We had several interesting findings that may prove informative for future method development in the MSA field.

As our minor findings in (II) we concluded that single linkage clustering is as good as the more popular UPGMA and neighbor joining for constructing the guide trees. Also, in support of the existing notion, we found that pairwise local and global alignments provide sufficient information for constructing high quality MSAs. Contrary to our expectations, conserved GTG motifs had minor effect on MSA quality. We also found that transitive homology links, as implemented by the clique transformation, did not improve accuracy.

As our main finding in (II) we were able to demonstrate the importance of consistency transformations and iterative refinement techniques. We concluded that MSA aligners implementing statistical consistency were the most accurate, and that triplet library extension that introduced new edges “on demand” was more accurate than the more conservative triplet library extension. We also found that MaxFlow is less accurate than both triplet library extensions. Additionally, we found that the performance of iterative refinement depends critically on the type of alignment partitioning: the tree-dependent restricted partition was found to be clearly superior to random partitioning. We also found that tree-dependent iterative refinement is very similar in accuracy and execution time to the best consistency transformations. This later finding can be applied to future MSA implementations that can benefit from the simplicity of the iterative refinement and from the straightforward tradeoff this technique offers between computational time and accuracy. Another implication is that the most accurate aligners (MSAProbs and ProbCons), which currently implement random iterative refinements, may benefit from adopting the tree-dependent strategy.

In publication (II) we also fulfilled the more traditional function of a comparative study by providing clear method rankings and recommendations for method users. Based on our results we recommend MSAProbs or ProbCons for the best accuracy, ClustalO for maximal speed and MAFFT for the best combination of both accuracy and speed. We also note that the latest MAFFT implementation has many options for making tradeoffs between accuracy and speed, and is generally more flexible than ClustalO. Although, these recommendations appear simple and to some extent repeat previous studies, it is still of considerable importance to communicate these results to a wider research audience. Currently, only a minority of researchers seems to be aware of the most accurate and fast MSA tools, while the majority seems to favor MSA tools that are drastically outdated.

In publication (III) we presented the Artificial Dilution Series (ADS), which is the first framework for selecting and developing evaluation metrics (EvMs) for GO classifiers. We were motivated by the discrepancies and confusion related to method rankings in CAFA challenges. Using ADS, we demonstrated that several EvMs used in CAFA and other comparative studies are either biased or indiscriminative. We were able to improve existing EvMs and to make clear recommendations.

Using the ADS framework, we demonstrated that most gene-centric (GC) and unstructured (US) EvMs that do not consider GO class size are biased. This consideration has been partly acknowledged in CAFA challenges which rightly avoided all AUC EvMs except the term-centric (TC) ROC-AUC, which is insensitive to class-imbalance. Still, the CAFA challenges did use F_{max}, which is also gene-centric and does not have any mechanism for addressing the class-imbalance problem. Using ADS, we demonstrated that F_{max} is indeed clearly biased for naïve and other false positive predictions that simply assign the same set of large uninformative GO classes to all genes. Furthermore, we demonstrated that the term-centric EvMs and EvMs that include information content in their definition are immune to the class-imbalance problem. Particularly, we showed that TC AUC, S_{min}, SimGIC, Lin E/F and Resnik E/F are immune to bias induced by class-imbalance.

The practical implications of these findings apply to method rankings in completed and future AFP comparative studies. For example, CAFA rankings based on TC ROC-AUC and S_{min} seem valid, while those based on F_{max} are questionable. In future CAFA challenges, it might be advisable to include AFP rankings based on SimGIC, which demonstrated consistently high performance across all datasets. Rankings based on high performing semantic similarity metrics, such as Lin E/F and Resnik E/F, would also elucidate method performance from a novel perspective. These EvMs consider semantically similar predictions and are not limited to the binary true or false evaluation inherent in other metrics.

More generally, ADS provides the first universal framework for testing and developing EvMs for AFP evaluation. EvMs are the core components of AFP comparative studies and, thus, applying the more discriminative and objective EvMs will certainly improve information flow within the researcher community concerned with AFP development and application.

ADS was designed for EvMs for GO annotations, however, similar problems are encountered with other biological ontologies. Generally, ontologies in biosciences tend to contain hierarchical and graph-like dependencies that can give rise to various forms of bias. This calls for extending the ADS framework to other biological ontologies.

To conclude, I would like to recap the importance of comparative evaluation studies for bioinformatics and biosciences at large. Identifying limitations and strong sides of different solutions, reducing bias related to self-assessment, providing clear rankings and recommendations and communicating this information in clear and recognizable format both to users and method developers is likely to drastically improve information flow within the scientific community.

ACKNOWLEDGEMENTS

This work was carried out at the Institute of Biotechnology at the University of Helsinki during 2008-2011 and 2016-2020. The funding for this study has been provided by the Academy of Finland, Emil Aaltonen Foundation, Integrative Life Science Doctoral Program and the University of Helsinki.

I would like to express my gratitude to many people that I've worked with at the Institute of Biotechnology. First of I am most grateful to my supervisor Professor Liisa Holm. Everything I know about sequence analysis I owe to her. I thank Dr. Matti Kankainen for introducing me to the field of sequence annotations, for fruitful discussions and collaboration. I thank Petri Törönen for inspiring discussions, for the brainstorming sessions, for his humor and an ever-positive attitude. I also thank all my co-authors, collaborators and colleagues at the University of Helsinki.

I would like to thank Professor Jari Björne, Professor Leo Lahti and Dr. Ari Löytynoja for their valuable comments that helped to shape this thesis into its final form. I also thank Dr. Alan Medlar for his generously provided language revisions.

Finally, I am most grateful to my friends and family. Thank you all for your love and support.

Helsinki, May 2020

REFERENCES

[1] Mangul S, Martin LS, Hill BL, et al. Systematic benchmarking of omics computational tools. Nat. Commun. 2019;10:1–11.

[2] Weber LM, Saelens W, Cannoodt R, et al. Essential guidelines for computational method benchmarking. Genome Biol. 2019;20:125.

[3] Aniba MR, Poch O, Thompson JD. Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. Nucleic Acids Res. 2010;38:7353–7363.

[4] Iantorno S, Gori K, Goldman N, et al. Who watches the watchmen? An appraisal of benchmarks for multiple sequence alignment. Mult. Seq. Alignment Methods. Springer; 2014. p. 59–73.

[5] Boulesteix A-L. Over-optimism in bioinformatics research. Bioinformatics. 2010;26:437–439.

[6] Jelizarow M, Guillemot V, Tenenhaus A, et al. Over-optimism in bioinformatics: an illustration. Bioinformatics. 2010;26:1990–1998.

[7] Norel R, Rice JJ, Stolovitzky G. The self-assessment trap: can we all be better than average? Mol. Syst. Biol. 2011;7:537.

[8] Boulesteix A-L. Ten Simple Rules for Reducing Overoptimistic Reporting in Methodological Computational Research. PLoS Comput. Biol. 2015;11:e1004191.

[9] Boulesteix A-L, Wilson R, Hapfelmeier A. Towards evidence-based computational statistics: lessons from clinical research on the role and design of real-data benchmark studies. BMC Med. Res. Methodol. 2017;17:138.

[10] Boulesteix A-L, Lauer S, Eugster MJA. A plea for neutral comparison studies in computational sciences. PloS One. 2013;8:e61562.

[11] Boulesteix A-L, Binder H, Abrahamowicz M, et al. On the necessity and design of studies comparing statistical methods. Biom. J. 2018;60:216–218.

[12] Meyer F, Hofmann P, Belmann P, et al. AMBER: Assessment of Metagenome BinnERs. GigaScience. 2018;7:giy069.

[13] Meyer F, Bremges A, Belmann P, et al. Assessing taxonomic metagenome profilers with OPAL. Genome Biol. 2019;20:51.

[14] Peters B, Brenner SE, Wang E, et al. Putting benchmarks in their rightful place: The heart of computational biology. PLoS Comput. Biol. 2018;14:e1006494.

[15] Döring A, Weese D, Rausch T, et al. SeqAn an efficient, generic C++ library for sequence analysis. BMC Bioinformatics. 2008;9:11.

[16] Crick F. Central dogma of molecular biology. Nature. 1970;227:561–563.

[17] Alberts B, Bray D, Hopkin K, et al. Essential cell biology. 4th ed. Garland Science; 2013.

[18] Uzawa T, Yamagishi A, Oshima T. Polypeptide Synthesis Directed by DNA as a Messenger in Cell-Free Polypetide Synthesis by Extreme Thermophiles, Thermus thermophilus HB27 and Sulfolobus tokodaii Strain 7. J. Biochem. (Tokyo). 2002;131:849–853.

[19] Darwin C. On the origin of species. Routledge; 1859.

[20] Woese CR, Kandler O, Wheelis ML. Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya. Proc. Natl. Acad. Sci. 1990;87:4576–4579.

[21] Forterre P. The universal tree of life: an update. Front. Microbiol. 2015;6:717.

[22] Hug LA, Baker BJ, Anantharaman K, et al. A new view of the tree of life. Nat. Microbiol. 2016;1:1–6.

[23] Hinchliff CE, Smith SA, Allman JF, et al. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. Proc. Natl. Acad. Sci. U. S. A. 2015;112:12764–12769.

[24] Durbin R, Eddy SR, Krogh A, et al. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge university press; 1998.

[25] Smith JM. Natural selection and the concept of a protein space. Nature. 1970;225:563–564.

[26] Thompson JD, Linard B, Lecompte O, et al. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. PloS One. 2011;6:e18093.

[27] Pei J. Multiple protein sequence alignment. Curr. Opin. Struct. Biol. 2008;18:382–386.

[28] Bao Y, Bolotov P, Dernovoy D, et al. The influenza virus resource at the National Center for Biotechnology Information. J. Virol. 2008;82:596–601.

[29] Hatcher EL, Zhdanov SA, Bao Y, et al. Virus Variation Resource - improved response to emergent viral outbreaks. Nucleic Acids Res. 2017;45:D482–D490.

[30] Singh S, Tokhunts R, Baubet V, et al. Sonic hedgehog mutations identified in holoprosencephaly patients can act in a dominant negative manner. Hum. Genet. 2009;125:95–103.

[31] Tavtigian SV, Greenblatt MS, Lesueur F, et al. In silico analysis of missense substitutions using sequence-alignment based methods. Hum. Mutat. 2008;29:1327–1336.

[32] Chan PA, Duraisamy S, Miller PJ, et al. Interpreting missense variants: comparing computational methods in human disease genes CDKN2A, MLH1, MSH2, MECP2, and tyrosinase (TYR). Hum. Mutat. 2007;28:683–693.

[33] Stormo GD, Schneider TD, Gold L, et al. Use of the “Perceptron” algorithm to distinguish translational initiation sites in E. coli. Nucleic Acids Res. 1982;10:2997–3011.

[34] Altschul SF, Koonin EV. Iterated profile searches with PSI-BLAST—a tool for discovery in protein databases. Trends Biochem. Sci. 1998;23:444–447.

[35] Xia X. Position weight matrix, gibbs sampler, and the associated significance tests in motif characterization and prediction. Scientifica. 2012;2012:917540.

[36] Eddy SR. Profile hidden Markov models. Bioinformatics. 1998;14:755–763.

[37] Finn RD, Coggill P, Eberhardt RY, et al. The Pfam protein families database: towards a more sustainable future. Nucleic Acids Res. 2016;44:D279–D285.

[38] Haft DH, Selengut JD, Richter RA, et al. TIGRFAMs and genome properties in 2013. Nucleic Acids Res. 2013;41:D387–D395.

[39] Hughey R, Krogh A. Hidden Markov models for sequence analysis: extension and analysis of the basic method. Comput. Appl. Biosci. CABIOS. 1996;12:95–107.

[40] Söding J. Protein homology detection by HMM–HMM comparison. Bioinformatics. 2005;21:951–960.

[41] Eddy SR. Accelerated Profile HMM Searches. PLOS Comput. Biol. 2011;7:e1002195.

[42] Thompson JD, Plewniak F, Poch O. BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics. 1999;15:87–88.

[43] Thompson JD, Koehl P, Ripp R, et al. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins Struct. Funct. Bioinforma. 2005;61:127–136.

[44] Blackshields G, Wallace IM, Larkin M, et al. Analysis and Comparison of Benchmarks for Multiple Sequence Alignment. In Silico Biol. 2006;6:321–339.

[45] Raghava G, Searle SM, Audley PC, et al. OXBench: A benchmark for evaluation of protein multiple sequence alignment accuracy. BMC Bioinformatics. 2003;4:47.

[46] Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32:1792–1797.

[47] Van Walle I, Lasters I, Wyns L. SABmark—a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics. 2005;21:1267–1268.

[48] Notredame C, Higgins DG, Heringa J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. 2000;302:205–217.

[49] Katoh K, Misawa K, Kuma K, et al. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic Acids Res. 2002;30:3059–3066.

[50] Do CB, Mahabhashyam MS, Brudno M, et al. ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res. 2005;15:330–340.

[51] Lassmann T, Sonnhammer ELL. Kalign--an accurate and fast multiple sequence alignment algorithm. BMC Bioinformatics. 2005;6:298.

[52] Liu Y, Schmidt B, Maskell DL. MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities. Bioinformatics. 2010;26:1958–1964.

[53] Sievers F, Wilm A, Dineen D, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Mol. Syst. Biol. 2011;7:539.

[54] Pang A, Smith AD, Nuin PA, et al. SIMPROT: using an empirically determined indel distribution in simulations of protein evolution. BMC Bioinformatics. 2005;6:236.

[55] Strope CL, Abel K, Scott SD, et al. Biological sequence simulation for testing complex evolutionary hypotheses: indel-Seq-Gen version 2.0. Mol. Biol. Evol. 2009;26:2581–2593.

[56] Fletcher W, Yang Z. INDELible: a flexible simulator of biological sequence evolution. Mol. Biol. Evol. 2009;26:1879–1888.

[57] Pervez MT, Babar ME, Nadeem A, et al. Evaluating the accuracy and efficiency of multiple sequence alignment methods. Evol Bioinform Online. 2014;10:205–217.

[58] Nuin PAS, Wang Z, Tillier ERM. The accuracy of several multiple sequence alignment programs for proteins. BMC Bioinformatics. 2006;7:471.

[59] Kim J, Sinha S. Towards realistic benchmarks for multiple alignments of non-coding sequences. BMC Bioinformatics. 2010;11:54.

[60] Dessimoz C, Gil M. Phylogenetic assessment of alignments reveals neglected tree signal in gaps. Genome Biol. 2010;11:R37.

[61] Pearson WR, Sierk ML. The limits of protein sequence comparison? Curr. Opin. Struct. Biol. 2005;15:254–260.

[62] Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 1994;22:4673–4680.

[63] Lee C, Grasso C, Sharlow MF. Multiple sequence alignment using partial order graphs. Bioinformatics. 2002;18:452–464.

[64] Perrodou E, Chica C, Poch O, et al. A new protein linear motif benchmark for multiple sequence alignment software. BMC Bioinformatics. 2008;9:213.

[65] Pais FS-M, Ruy P de C, Oliveira G, et al. Assessing the efficiency of multiple sequence alignment programs. Algorithms Mol. Biol. AMB. 2014;9:4.

[66] Katoh K, Toh H. Recent developments in the MAFFT multiple sequence alignment program. Brief. Bioinform. 2008;9:286–298.

[67] UniProtKB/Swiss-Prot 2019_08 [Internet]. [cited 2019 Sep 20]. Available from: https://www.uniprot.org/statistics/Swiss-Prot.

[68] UniProt [Internet]. [cited 2020 Jan 21]. Available from: https://www.uniprot.org/.

[69] Ashburner M, Ball CA, Blake JA, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. 2000;25:25–29.

[70] Radivojac P, Clark WT, Oron TR, et al. A large-scale evaluation of computational protein function prediction. Nat. Methods. 2013;10:221–227.

[71] Gene Ontology Consortium. Gene Ontology Consortium: going forward. Nucleic Acids Res. 2015;43:D1049-1056.

[72] Consortium GO. Expansion of the Gene Ontology knowledgebase and resources. Nucleic Acids Res. 2016;45:D331–D338.

[73] Gene Ontology Resource [Internet]. Gene Ontol. Resour. [cited 2019 Nov 11]. Available from: http://geneontology.org/stats.html.

[74] Gaudet P, Livstone MS, Lewis SE, et al. Phylogenetic-based propagation of functional annotations within the Gene Ontology consortium. Brief. Bioinform. 2011;12:449–462.

[75] Jiang Y, Oron TR, Clark WT, et al. An expanded evaluation of protein function prediction methods shows an improvement in accuracy. Genome Biol. 2016;17:184.

[76] Zhou N, Jiang Y, Bergquist TR, et al. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biol. 2019;20:244.

[77] CAFA | Bio Function Prediction [Internet]. [cited 2019 Nov 12]. Available from: https://www.biofunctionprediction.org/cafa/.

[78] Gillis J, Pavlidis P. Characterizing the state of the art in the computational assignment of gene function: lessons from the first critical assessment of functional annotation (CAFA). BMC Bioinformatics. BioMed Central; 2013. p. S15.

[79] Kahanda I, Funk CS, Ullah F, et al. A close look at protein function prediction evaluation protocols. GigaScience. 2015;4:41.

[80] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 1970;48:443–453.

[81] Gotoh O. An improved algorithm for matching biological sequences. J. Mol. Biol. 1982;162:705–708.

[82] Buljan M, Bateman A. The evolution of protein domain families. Biochem. Soc. Trans. 2009;37:751–755.

[83] Holm L, Sander C. Parser for protein folding units. Proteins. 1994;19:256–268.

[84] Smith TF, Waterman MS. Identification of common molecular subsequences. J. Mol. Biol. 1981;147:195–197.

[85] Waterman MS, Eggert M. A new algorithm for best subsequence alignments with application to tRNA-rRNA comparisons. J. Mol. Biol. 1987;197:723–728.

[86] Feng D-F, Doolittle RF. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. J. Mol. Evol. 1987;25:351–360.

[87] Altschul SF. Amino acid substitution matrices from an information theoretic perspective. J. Mol. Biol. 1991;219:555–565.

[88] Dayhoff M, Schwartz R, Orcutt B. 22 a model of evolutionary change in proteins. Atlas Protein Seq. Struct. National Biomedical Research Foundation Silver Spring MD; 1978. p. 345–352.

[89] Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci. 1992;89:10915–10919.

[90] Holm L, Sander C. Dali/FSSP classification of three-dimensional protein folds. Nucleic Acids Res. 1997;25:231–234.

[91] Holm L, Laakso LM. Dali server update. Nucleic Acids Res. 2016;44:W351–W355.

[92] Lo Conte L, Ailey B, Hubbard TJ, et al. SCOP: a structural classification of proteins database. Nucleic Acids Res. 2000;28:257–259.

[93] Katoh K, Kuma K, Toh H, et al. MAFFT version 5: improvement in accuracy of multiple sequence alignment. Nucleic Acids Res. 2005;33:511–518.

[94] Rausch T, Emde A-K, Weese D, et al. Segment-based multiple sequence alignment. Bioinformatics. 2008;24:i187–i192.

[95] Heger A, Holm L. More for less in structural genomics. J. Struct. Funct. Genomics. 2003;4:57–66.

[96] Lassmann T. Kalign 3: multiple sequence alignment of large data sets. Bioinformatics. 2019;btz795 (corrected proof).

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

[98] Wheeler TJ, Kececioglu JD. Multiple alignment by aligning alignments. Bioinformatics. 2007;23:i559–i568.

[99] Lassmann T, Frings O, Sonnhammer EL. Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features. Nucleic Acids Res. 2008;37:858–865.

[100] Gonnet GH, Cohen MA, Benner SA. Exhaustive matching of the entire protein sequence database. Science. 1992;256:1443–1445.

[101] Henikoff S, Henikoff JG. Position-based sequence weights. J. Mol. Biol. 1994;243:574–578.

[102] Gotoh O. A weighting system and algorithm for aligning many phylogenetically related sequences. Comput. Appl. Biosci. CABIOS. 1995;11:543–551.

[103] Myers EW, Miller W. Optimal alignments in linear space. Comput. Appl. Biosci. CABIOS. 1988;4:11–17.

[104] Barton GJ, Sternberg MJ. A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons. J. Mol. Biol. 1987;198:327–337.

[105] Berger MP, Munson PJ. A novel randomized iterative strategy for aligning multiple protein sequences. Bioinformatics. 1991;7:479–484.

[106] Hirosawa M, Totoki Y, Hoshida M, et al. Comprehensive study on iterative algorithms of multiple sequence alignment. Bioinformatics. 1995;11:13–18.

[107] Reinert K, Dadi TH, Ehrhardt M, et al. The SeqAn C++ template library for efficient sequence analysis: A resource for programmers. J. Biotechnol. 2017;261:157–168.

[108] Koskinen P, Törönen P, Nokso-Koivisto J, et al. PANNZER: high-throughput functional annotation of uncharacterized proteins in an error-prone environment. Bioinformatics. 2015;31:1544–1552.

[109] Törönen P, Medlar A, Holm L. PANNZER2: a rapid functional annotation web server. Nucleic Acids Res. 2018;46:W84–W88.

[110] Falda M, Toppo S, Pescarolo A, et al. Argot2: a large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. BMC Bioinformatics. 2012;13:S14.

[111] Gong Q, Ning W, Tian W. GoFDR: A sequence alignment based method for predicting protein functions. Methods. 2016;93:3–14.

[112] Lan L, Djuric N, Guo Y, et al. MS-k NN: protein function prediction by integrating multiple data sources. BMC Bioinformatics. BioMed Central; 2013. p. S8.

[113] Cozzetto D, Buchan DW, Bryson K, et al. Protein function prediction by massive integration of evolutionary analyses and multiple data sources. BMC Bioinformatics. BioMed Central; 2013. p. S1.

[114] Götz S, García-Gómez JM, Terol J, et al. High-throughput functional annotation and data mining with the Blast2GO suite. Nucleic Acids Res. 2008;36:3420–3435.

[115] Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool. J. Mol. Biol. 1990;215:403–410.

[116] Altschul SF, Madden TL, Schäffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25:3389–3402.

[117] Potter SC, Luciani A, Eddy SR, et al. HMMER web server: 2018 update. Nucleic Acids Res. 2018;46:W200–W204.

[118] Somervuo P, Holm L. SANSparallel: interactive homology search against Uniprot. Nucleic Acids Res. 2015;43:W24–W29.

[119] Durbin R, Eddy SR, Krogh A, et al. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press; 1998.

[120] Subramanian A, Tamayo P, Mootha VK, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. 2005;102:15545–15550.

[121] Berriz GF, King OD, Bryant B, et al. Characterizing gene sets with FuncAssociate. Bioinformatics. 2003;19:2502–2504.

[122] Martin D, Brun C, Remy E, et al. GOToolBox: functional analysis of gene datasets based on Gene Ontology. Genome Biol. 2004;5:1.

[123] Ewens WJ, Grant GR. Statistical methods in bioinformatics: an introduction. Springer Science & Business Media; 2006.

[124] Fouts DE. Phage_Finder: automated identification and classification of prophage regions in complete bacterial genome sequences. Nucleic Acids Res. 2006;34:5839–5851.

[125] Lima-Mendez G, Van Helden J, Toussaint A, et al. Prophinder: a computational tool for prophage prediction in prokaryotic genomes. Bioinformatics. 2008;24:863–865.

[126] Kroese DP, Brereton T, Taimre T, et al. Why the Monte Carlo method is so important today. Wiley Interdiscip. Rev. Comput. Stat. 2014;6:386–392.

[127] Martin DM, Berriman M, Barton GJ. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. BMC Bioinformatics. 2004;5:178.

[128] Engelhardt BE, Jordan MI, Muratore KE, et al. Protein molecular function prediction by Bayesian phylogenomics. PLoS Comput. Biol. 2005;1:e45.

[129] Friedberg I. Automated protein function prediction—the genomic challenge. Brief. Bioinform. 2006;7:225–242.

[130] Hawkins T, Chitale M, Luban S, et al. PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. Proteins. 2009;74:566–582.

[131] Wass MN, Sternberg MJ. ConFunc—functional annotation in the twilight zone. Bioinformatics. 2008;24:798–806.

[132] Chitale M, Hawkins T, Park C, et al. ESG: extended similarity group method for automated protein function prediction. Bioinformatics. 2009;25:1739–1745.

[133] Engelhardt BE, Jordan MI, Srouji JR, et al. Genome-scale phylogenetic function annotation of large and diverse protein families. Genome Res. 2011;21:1969–1980.

[134] Minneci F, Piovesan D, Cozzetto D, et al. FFPred 2.0: improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. PLoS One. 2013;8:e63754.

[135] Peña-Castillo L, Tasan M, Myers CL, et al. A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. Genome Biol. 2008;9:1.

[136] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves. Proc. 23rd Int. Conf. Mach. Learn. ACM; 2006. p. 233–240.

[137] Hand DJ, Till RJ. A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach. Learn. 2001;45:171–186.

[138] Resnik P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. J Artif Intell ResJAIR. 1999;11:95–130.

[139] Pesquita C, Faria D, Falcao AO, et al. Semantic similarity in biomedical ontologies. PLoS Comput Biol. 2009;5:e1000443.

[140] Clark WT, Radivojac P. Information-theoretic evaluation of predicted ontological annotations. Bioinformatics. 2013;29:i53–i61.

[141] Lin D. An information-theoretic definition of similarity. Icml. Citeseer; 1998. p. 296–304.

[142] Lord PW, Stevens RD, Brass A, et al. Semantic similarity measures as tools for exploring the gene ontology. Biocomput. 2003. World Scientific; 2002. p. 601–612.

[143] Pesquita C, Faria D, Bastos H, et al. Evaluating GO-based semantic similarity measures. Proc 10th Annu. Bio-Ontol. Meet. 2007. p. 38.

[144] Couto FM, Silva MJ, Coutinho PM. Measuring semantic similarity between Gene Ontology terms. Data Knowl. Eng. 2007;61:137–152.

[145] Schlicker A, Domingues FS, Rahnenführer J, et al. A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics. 2006;7:302.

[146] Azuaje F, Wang H, Bodenreider O. Ontology-driven similarity approaches to supporting gene functional assessment. Proc. ISMB2005 SIG Meet. Bio-Ontol. 2005. p. 9–10.

[147] Pesquita C, Faria D, Bastos H, et al. Metrics for GO based protein semantic similarity: a systematic evaluation. BMC Bioinformatics. 2008;9:S4.

[148] Ferri C, Hernández-Orallo J, Modroiu R. An experimental comparison of performance measures for classification. Pattern Recognit. Lett. 2009;30:27–38.

[149] Sokolova M, Lapalme G. A systematic analysis of performance measures for classification tasks. Inf. Process. Manag. 2009;45:427–437.

[150] Seliya N, Khoshgoftaar TM, Van Hulse J. A study on the relationships of classifier performance metrics. IEEE; 2009. p. 59–66.

[151] Jenkinson HF. Beyond the oral microbiome. Environ. Microbiol. 2011;13:3077–3087.

[152] Kotiranta A, Lounatmaa K, Haapasalo M. Epidemiology and pathogenesis of Bacillus cereus infections. Microbes Infect. 2000;2:189–198.

[153] Turroni F, Van Sinderen D, Ventura M. Genomics and ecological overview of the genus Bifidobacterium. Int. J. Food Microbiol. 2011;149:37–44.

[154] Ventura M, Turroni F, Zomer A, et al. The Bifidobacterium dentium Bd1 genome sequence reflects its genetic adaptation to the human oral cavity. PLoS Genet. 2009;5:e1000785.

[155] Di Bella S, Ascenzi P, Siarakas S, et al. Clostridium difficile Toxins A and B: Insights into Pathogenic Properties and Extraintestinal Effects. Toxins. 2016;8:134.

[156] Peck MW. Biology and genomic analysis of Clostridium botulinum. Adv. Microb. Physiol. 2009;55:183–265, 320.

[157] Labbe RG, Juneja VK. Chapter 10 - Clostridium perfringens. In: Dodd CER, Aldsworth T, Stein RA, et al., editors. Foodborne Dis. Third Ed. Academic Press; 2017. p. 235–242.

[158] Hoskisson PA. Microbe Profile: Corynebacterium diphtheriae - an old foe always ready to seize opportunity. Microbiol. Read. Engl. 2018;164:865–867.

[159] Tauch A, Kaiser O, Hain T, et al. Complete genome sequence and analysis of the multiresistant nosocomial pathogen Corynebacterium jeikeium K411, a lipid-requiring bacterium of the human skin flora. J. Bacteriol. 2005;187:4671–4682.

[160] Martínez-Martínez L, Suárez AI, Rodríguez-Baño J, et al. Clinical significance of Corynebacterium striatum isolated from human samples. Clin. Microbiol. Infect. 1997;3:634–639.

[161] Guzman Prieto AM, van Schaik W, Rogers MRC, et al. Global Emergence and Dissemination of Enterococci as Nosocomial Pathogens: Attack of the Clones? Front. Microbiol. 2016;7:788.

[162] Africa CWJ, Nel J, Stemmet M. Anaerobes and bacterial vaginosis in pregnancy: virulence factors contributing to vaginal colonisation. Int. J. Environ. Res. Public. Health. 2014;11:6979–7000.

[163] Sun Z, Harris HMB, McCann A, et al. Expanding the biotechnology potential of lactobacilli through comparative genomics of 213 strains and associated genera. Nat. Commun. 2015;6:1–13.

[164] Martín R, Miquel S, Ulmer J, et al. Role of commensal and probiotic bacteria in human health: a focus on inflammatory bowel disease. Microb. Cell Factories. 2013;12:71.

[165] de Noordhout CM, Devleesschauwer B, Angulo FJ, et al. The global burden of listeriosis: a systematic review and meta-analysis. Lancet Infect. Dis. 2014;14:1073–1082.

[166] Authority EFS, Prevention EC for D, Control (EFSA, et al. The European Union summary report on trends and sources of zoonoses, zoonotic agents and food-borne outbreaks in 2017. EFSa J. 2018;16:e05500.

[167] Koskinen P, Deptula P, Smolander O-P, et al. Complete genome sequence of Propionibacterium freudenreichii DSM 20271(T). Stand. Genomic Sci. 2015;10:83.

[168] Johansson C, Rautelin H, Kaden R. Staphylococcus argenteus and Staphylococcus schweitzeri are cytotoxic to human cells in vitro due to high expression of alpha-hemolysin Hla. Virulence. 2019;10:502–510.

[169] Tong SY, Davis JS, Eichenberger E, et al. Staphylococcus aureus infections: epidemiology, pathophysiology, clinical manifestations, and management. Clin. Microbiol. Rev. 2015;28:603–661.

[170] Foster AP. Staphylococcal skin disease in livestock. Vet. Dermatol. 2012;23:342–351, e63.

[171] Lysková P, Vydržalová M, Královcová D, et al. Prevalence and Characteristics of *Streptococcus canis* Strains Isolated from Dogs and Cats. Acta Vet. Brno. 2007;76:619–625.

[172] Reichmann P, Nuhn M, Denapaite D, et al. Genome of Streptococcus oralis Strain Uo5. J. Bacteriol. 2011;193:2888–2889.

[173] Kolenbrander PE, Palmer RJ, Rickard AH, et al. Bacterial interactions and successions during plaque development. Periodontol. 2000. 2006;42:47–79.

[174] van de Beek D, de Gans J, Tunkel AR, et al. Community-acquired bacterial meningitis in adults. N. Engl. J. Med. 2006;354:44–53.

[175] Nilsson P, Laurell MH. Carriage of penicillin-resistant Streptococcus pneumoniae by children in day-care centers during an intervention program in Malmo, Sweden. Pediatr. Infect. Dis. J. 2001;20:1144–1149.

[176] Lynskey NN, Lawrenson RA, Sriskandan S. New understandings in Streptococcus pyogenes. Curr. Opin. Infect. Dis. 2011;24:196–202.

[177] Sriskandan S, Slater JD. Invasive disease and toxic shock due to zoonotic Streptococcus suis: an emerging infection in the East? PLoS Med. 2006;3:e187.

[178] Machado VS, Bicalho RC. Complete Genome Sequence of Trueperella pyogenes, an Important Opportunistic Pathogen of Livestock. Genome Announc. 2014;2:e00400-14.

[179] Heger A, Mallick S, Wilton C, et al. The global trace graph, a novel paradigm for searching protein sequence databases. Bioinformatics. 2007;23:2361–2367.

[180] Hanczar B, Hua J, Sima C, et al. Small-sample precision of ROC-related estimates. Bioinformatics. 2010;26:822–830.