

**Statistical methods for biological sequence
analysis for DNA binding motifs and
protein contacts**

Dissertation

for the award of the degree

“Doctor rerum naturalium”

Division of Mathematics and Natural Sciences
of the Georg-August-Universität Göttingen

within the doctoral program

International Max Planck Research School for Genome Science
of the Georg-August University School of Science (GAUSS)

submitted by

Christian Roth

from Deggendorf, Germany

Göttingen, June 2021

Thesis Committee

- Dr. Johannes Söding, Research Group Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry
- Prof. Dr. Tim Beißbarth, Department of Medical Bioinformatics, University Medical Center Göttingen
- Dr. Nico Posnien, Department of Developmental Biology, Göttingen Center for Molecular Biosciences

Members of the Examination Board

First Reviewer: Dr. Johannes Soeding, Research Group Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry

Second Reviewer: Prof. Dr. Tim Beißbarth, Department of Medical Bioinformatics, University Medical Center Göttingen

Further members of the Examination Board

- Prof. Dr. Burkhard Morgenstern, Institute for Microbiology and Genetics, Department of Bioinformatics, University of Göttingen
- Dr. Marieke Oudelaar, Research Group Genome Organization and Regulation, Max Planck Institute for Biophysical Chemistry
- Dr. Nico Posnien, Department of Developmental Biology, Göttingen Center for Molecular Biosciences
- Prof. Dr. Stephan Waack, Institute for Computer Science, University of Göttingen

Date of oral examination: 6th September 2021

Summary

Over the last decades a revolution in novel measurement techniques has permeated the biological sciences filling the databases with unprecedented amounts of data ranging from genomics, transcriptomics, proteomics and metabolomics to structural and ecological data. In order to extract insights from the vast quantity of data, computational and statistical methods are nowadays crucial tools in the toolbox of every biological researcher. In this thesis I summarize my contributions in two data-rich fields in biological sciences: transcription factor binding to DNA and protein structure prediction from protein sequences with shared evolutionary ancestry.

In the first part of my thesis I introduce our work towards a web server for analysing transcription factor binding data with Bayesian Markov Models. In contrast to classical PWM or di-nucleotide models, Bayesian Markov models can capture complex inter-nucleotide dependencies that can arise from shape-readout and alternative binding modes. In addition to giving access to our methods in an easy-to-use, intuitive web-interface, we provide our users with novel tools and visualizations to better evaluate the biological relevance of the inferred binding motifs. We hope that our tools will prove useful for investigating weak and complex transcription factor binding motifs which cannot be predicted accurately with existing tools.

The second part discusses a statistical attempt to correct out the phylogenetic bias arising in co-evolution methods applied to the contact prediction problem. Co-evolution methods have revolutionized the protein-structure prediction field more than 10 years ago, and, until very recently, have retained their importance as crucial input features to deep neural networks. As the co-evolution information is extracted from evolutionarily related sequences, we investigated whether the phylogenetic bias to the signal can be corrected out in a principled way using a variation of the Felsenstein's tree-pruning algorithm applied in combination with an independent-pair assumption to derive pairwise amino counts that are corrected for the evolutionary history. Unfortunately, the contact prediction derived from our corrected pairwise amino acid counts did not yield a competitive performance.

Acknowledgements

First and foremost, I would like to thank Dr. Johannes Söding for giving me the opportunity to work with so many kind and talented people and always being interested in discussing theory, society and politics. Furthermore, I would like to thank Prof. Tim Beissbarth and Dr. Nico Posnien for accompanying me on my journey as members of by TAC and Prof. Stephan Waack, Dr. Marieke Oudelaar and Prof. Burkhard Morgenstern for volunteering their time as examiners.

I will be eternally grateful to those who offered help when I needed it the most, especially Dr. Salma Sohrabi-Jahromi, Dr. Christel Winkelbach, Prof. Argyris Papantonis, Dr. Jessica Andreani and Dr. Nico Posnien. It is unlikely that I will ever be able to pay back what you did for me, but at the very least I will try my best to pay it forward.

I would like to thank Dr. Anja Kiesel and Dr. Wanwan Ge for their good work in the web server collaboration. I would like to thank Dr. Anna Sawicka for our exciting collaboration on the enhancer-promoter interaction project — unfortunately destiny had other plans. I would like to express my gratitude to Dr. Stefan Seemayer and Dr. Susann Bader. Even without scientific overlap, the high code quality in the scripts and tools you developed helped me getting started in the protein field.

Special thanks goes to Milot Mirdita who knowingly and willingly exposed himself to nerd-sniping and offered solutions to even the most obscure technical problems. I learnt a lot from you.

I would like to thank all present and former lab members that helped me acknowledge and appreciate diversity by sharing their views and opinions. Especially Dr. Eli Levy Karin for introducing me to the world of vegan cuisine, Ruoshi Zhang for being a patient cooking and dumpling teacher, Dr. Saikat Banerjee for reviving the board gaming tradition, countless interesting discussions and two amazing hiking trips.

I thank my family and Salma Sohrabi-Jahromi for helping me shape my value system and giving me advice while supporting me in all my decisions.

Last but not least, I would like to thank Dr. Henriette Irmer and Frauke Bergmann for always being helpful in case of problems and questions regarding the PhD, furthermore Janine Blümel and Almuth Burgdorf for assistance in all administrative tasks.

Contents

Board members	II
Summary	III
Acknowledgements	IV
Contents	V
List of Commonly used Abbreviations	VIII
I. BaMM Web Server	1
1. Introduction	2
1.1. Molecular basis of transcription and its regulation	2
1.1.1. Transcription and RNA polymerase II	2
Initiation	3
Elongation	4
Termination	4
1.1.2. Promoters and enhancers	5
1.1.3. Impact of chromatin architecture on gene regulation	6
1.1.4. Phase-separation in transcription	7
1.2. Transcription factors and the regulatory code	8
1.2.1. Molecular basis of DNA binding	9
1.2.2. Experimental methods for studying transcription factor binding	10
1.2.3. Mathematical basis of motif models	11
1.2.4. Computational approaches for uncovering the cis-regulatory code	13
1.3. Motivation aims and goals	13
2. Methods	14
2.1. Seeding stage PEnG	14
2.1.1. The PEnG algorithm	14
Counting stage	14
Scoring stage	14
Local optimization stage	15
PWM conversion stage	15
PWM sharpening stage	16

Merging stage	17
Derivation of local optimization scores	18
3. Results	22
3.1. PEnGmotif	22
3.1.1. PEnGmotif on artificial sequences	22
3.1.2. PEnGmotif on real sequences	23
3.1.3. Time benchmark	23
3.2. Five tools for motif analysis	27
3.2.1. De-novo motif discovery	27
3.2.2. Motif evaluation	28
3.2.3. Motif scanning	29
3.2.4. Motif-motif comparison	29
3.2.5. Browser for motif database	29
3.3. Job submission	30
3.4. Highly configurable, easily deployable open source server	31
3.5. Designed and setup for low maintenance	31
3.6. Comprehensive documentation	31
4. Manuscripts	32
4.1. BaMM web server	32
4.1.1. Author contributions	32
4.1.2. Code and data availability	32
4.1.3. Web server manuscript	33
4.2. BaMMmotif2	39
4.2.1. Publication abstract	39
4.2.2. Author contributions	39
4.2.3. Code and data availability	39
5. Discussion	40
5.1. Challenges when training complex models.	40
5.2. Limitations	42
5.3. Outlook	43
II. MRF coupling parameter correction	44
6. Introduction	45
6.1. Protein structure	46
6.1.1. Primary structure	46
6.1.2. Secondary structure	46
6.1.3. Tertiary structure	47
6.1.4. Quaternary structure	47

6.2.	Protein evolution	47
6.3.	Protein structure determination	48
6.3.1.	Experimental approaches to structure determination	49
	X-ray crystallography	49
	NMR spectroscopy	49
	Cryogenic electron microscopy	49
6.3.2.	Computational approaches to structure prediction	50
	The subdisciplines of structure prediction	50
6.3.3.	Contact prediction	51
	Transformation due to deep learning	53
	Evaluation of contact predictions	55
6.4.	Aims and scope	55
7.	Methods	57
7.1.	Felsenstein’s pruning algorithm for independent pairs	57
7.1.1.	A family-specific pairwise evolutionary model	57
7.1.2.	Calculating the likelihood	58
7.1.3.	Derivatives of the likelihood w.r.t to the model parameters	59
7.1.4.	Optimizing the likelihood	61
7.1.5.	Improvements to the core algorithm	61
	Transformation to logspace	61
	Reducing alphabet size	66
	Polynomial approximations to log2 and exp2	67
	Parallelization	69
7.1.6.	Calculating phylogenetically corrected pair counts	70
7.1.7.	Deriving the tree with a family-specific model	71
8.	Results	73
8.1.	Validation on simulated data	73
8.1.1.	FS-PCD on independent sequences	73
8.1.2.	FS-PCD on dependent sequences	78
8.2.	Pairwise couplings for contact prediction	82
8.2.1.	Simulations with artificial phylogenies	82
8.2.2.	Simulations with learnt phylogenies	85
8.2.3.	Applying pair-wise methods to real sequences	87
9.	Discussion	90
9.1.	Shortcomings and limitations	90
9.2.	End-to-end revolution	92
9.3.	Outlook	92
	References	95

List of Abbreviations

Pol II	RNA Polymerase II
mRNA	messenger RNA
TF	Transcription Factor
tRNA	transfer RNA
PIC	Pre-Initiation Complex
TSS	Transcription Start Site
TAD	Topologically Associated Domain
DBD	DNA Binding Domain
ZF	Zink Finger
PCR	Polymerase Chain Reaction
PWM	Position Weight Matrix
DNN	Deep Neural Network
BaMM	Bayesian Markov Model
EM	Expection Maximization
bp	base pairs
MSA	Multiple Sequence Alignment
DCA	Direct Coupling Analysis
MRF	Markov Random Field
PCD	Persistent Contrastive Divergence
APC	Average Product Correction

Part I.

BaMM Web Server

1. Introduction

In the 19th century Charles Darwin and Alfred Russel Wallace independently made probably the most profound discovery in modern biology: variation and selection are the two key drivers shaping all forms of life on earth. The beauty of their evolutionary theory lies in its simplicity: genetic information is susceptible to random changes (variation). The fraction of individuals carrying a specific bit of genetic information depends on the efficiency of carrier organisms in multiplying their own genetic information relative to non-carrier organisms (selection). When run over a long time, this process brings forth individuals that multiply their own genetic material efficiently under the existing constraints (well-adapted) (Darwin et al., 1858; Darwin, 1859).

100 years after Darwin's and Wallace's discovery, the DNA was discovered as the carrier molecule of genetic information. DNA contains the blueprints for the building blocks of the organism and encodes programs for regulating their production. This allowed a molecular interpretation of evolution: a well-adapted organism outperforms other organisms in spreading their own DNA by bringing the right molecules to the right place at the right time. The *Central Dogma of Molecular Biology* describes the realization process of genetic information. The DNA regions containing blueprints (genes) are transcribed to messenger RNAs (mRNAs) which in turn are translated into proteins. This assembly process runs constantly in living cells and all steps are under tight control by the encoded programs.

This chapter is concerned with the first step of the building block assembly: the transcription of genes into mRNA, especially in the regulation of this process.

1.1. Molecular basis of transcription and its regulation

1.1.1. Transcription and RNA polymerase II

Transcription is the process of transferring pieces of the genetic information (genes) stored in the DNA sequence into RNA molecules. On a molecular level transcription requires opening and unwinding the DNA double helix, synthesizing a new RNA molecule based on the DNA template strand and finally processing and releasing the RNA.

The central machineries in transcription are RNA polymerases which not only act as enzymes catalyzing RNA synthesis, but also provide the platform for recruiting and interacting with processing and regulatory factors. Eukaryotic cells encode several versions of the RNA polymerase, each specialised in the transcription of certain gene classes. While RNA Polymerase I

(Pol I) transcribes the precursor of the large ribosomal RNA, RNA Polymerase II Pol II produces mRNAs and a variety of non-coding RNAs, and RNA Polymerase III (Pol III) specializes in producing transfer RNAs (tRNAs) and the small ribosomal RNAs (Cramer, 2019). Pol II has an unstructured yet highly conserved stretch of tandem repeats with the consensus sequence $Y_1S_2P_3T_4S_5P_6S_7$ at the C-terminus of its largest subunit (Corden, 1990), referred to as the C-terminal domain (CTD). The CTD is present in all eukaryotes albeit with varying copy numbers of the tandem repeats and serves as an interaction platform of the polymerase with factors responsible for the RNA maturation process (Jeronimo et al., 2013).

A defining challenge in the evolution of complex multicellular organisms was the need to express subsets of genes during development and in all specialized cell types. As Pol II is responsible for transcribing mRNAs, the templates to all proteins, higher eukaryotes evolved a complex regulatory toolbox for controlling Pol II transcription in space and time. In the following the transcription cycle (Figure 1.1) and its regulation is discussed in more depth.

Initiation

The first phase in the transcription cycle is initiation. In initiation Pol II is recruited and positioned on the DNA upstream of the gene. The positions of Pol II in initiation are marked by regulatory signals encoded in the sequence of the DNA, called core-promoters (Smale and Kadonaga, 2003). Pol II reads out the core promoter indirectly by interacting with general transcription factors, a class of DNA binding proteins assembled on the core-promoter Orphanides et al. (1996); Juven-Gershon et al. (2008). The hereby formed complex has been termed pre-initiation complex (PIC). The PIC prepares for transcription initiation by opening up the DNA downstream of Pol II, a function contributed by a subunit of the general transcription factor TFIID (Kim et al., 2000). Co-activators binding to the PIC can be required to trigger transcription initiation in-vivo (Thomas and Chiang, 2006). The Mediator complex is a prominent co-activator and serves as a transient component of the PIC (Malik and Roeder, 2010; Wong et al., 2014). Just like the polymerase is recruited by general transcription factors binding to the core-promoter elements, the Mediator complex is recruited by transcription factors binding to regulatory signals in promoter-distal enhancer sequences (Kuras et al., 2003; Björklund and Gustafsson, 2005; Bhaumik et al., 2004). Loop structures in chromatin allow close spatial contact between the promoter bound Pol II and the enhancer bound Mediator (Carter et al., 2002; Petrenko et al., 2016). The phosphorylation of the CTD by the kinase module of TFIID is linked to Mediator dissociating from the PIC and Pol II promoter escape (Wong et al., 2014).

In order to achieve fine-grain control of gene expression, transcription initiation is tightly regulated. The recruitment of Pol II e.g. requires accessibility of the core-promoter to the general transcription factors. In its most compact form, DNA is tightly wrapped around nucleosomes and is thus not accessible to all but very few so-called pioneer factors (Zaret and Carroll, 2011; Magnani et al., 2011; Iwafuchi-Doi and Zaret, 2014). Displacing nucleosomes disrupts the tight chromatin packing and makes the promoter sequences accessible for transcription factor binding, a hallmark of transcriptionally active genes (Reeves, 1984; Lee et al., 2004). In addition

to chromatin accessibility at enhancer elements, gene activation is controlled by dynamic loop formation between promoter and enhancer elements (Kadauke and Blobel, 2009; De Laat and Duboule, 2013; Deng et al., 2012). These regulation strategies are underlying principles of the distinct expression patterns observed in different cell types of the same organism. It is however important to keep in mind that transcription is a highly stochastic process (Sanchez and Golding, 2013; Fukaya et al., 2016) and genome structure and loop formation in particular are dynamic processes (Hansen et al., 2017; Fudenberg et al., 2017; Banigan and Mirny, 2020). These dynamics are essential for the fine-grained transcription control individual cells exhibit in the face of internal and external stimuli.

Elongation

Having escaped the promoter, Pol II catalyzes the formation of the phosphodiester bonds between the nascent RNA and the ribonucleotides dictated by the DNA template strand, and thereby continuously elongates the RNA. In the elongation phase the CTD of Pol II accumulates phosphorylation, a hallmark of the processive Pol II (Christmann and Dahmus, 1981; Sims et al., 2004; Cramer, 2019). In order to gain its functional mature form, RNA produced in the elongation phase undergoes refinement steps such as capping, splicing, polyadenylation. The phosphorylated CTD of the moving polymerase acts as a binding platform and consequentially enables co-transcriptional RNA maturation (Phatnani and Greenleaf, 2006; Perales and Bentley, 2009).

As with transcription initiation, the need for fine-grain control of gene expression has given rise to regulation strategies of transcription elongation, of which proximal promoter pausing is probably the best understood mechanism. Typically 25 to 50 nucleotides after the transcription start site (TSS) the polymerase stalls while still tightly binding DNA and RNA (Core and Adelman, 2019). The unprocessive polymerase is then further stabilized through binding NELF and Spt5, while awaiting a pause-release signal (Core and Adelman, 2019; Yamaguchi et al., 1999). Pause release is directly linked to the phosphorylation of Spt5 by P-TEFb, which in turn releases NELF and thereby prevents it from stalling the Pol II (Cheng and Price, 2007; Vos et al., 2018). To release Pol II P-TEFb is recruited to the promoter by transcription factors and co-activators, hence a process that is under transcriptional control (Li et al., 2018). Promoter-proximal pausing has been suggested to be a mechanism for quantitatively fine-tuning expression, not just a binary on-off switch (Gressel et al., 2019; Core and Adelman, 2019).

Termination

In transcription termination, the final step of the transcription cycle, the RNA is cleaved, further processed towards maturity and Pol II is recycled for starting a new transcription cycle (reviewed in Kuehner et al. (2011); Porrua and Libri (2015)). There are multiple termination pathways, depending on the recruited co-factors. Protein coding transcripts undergo cleavage and polyadenylation, a requirement for their stability, export and translational efficiency in order

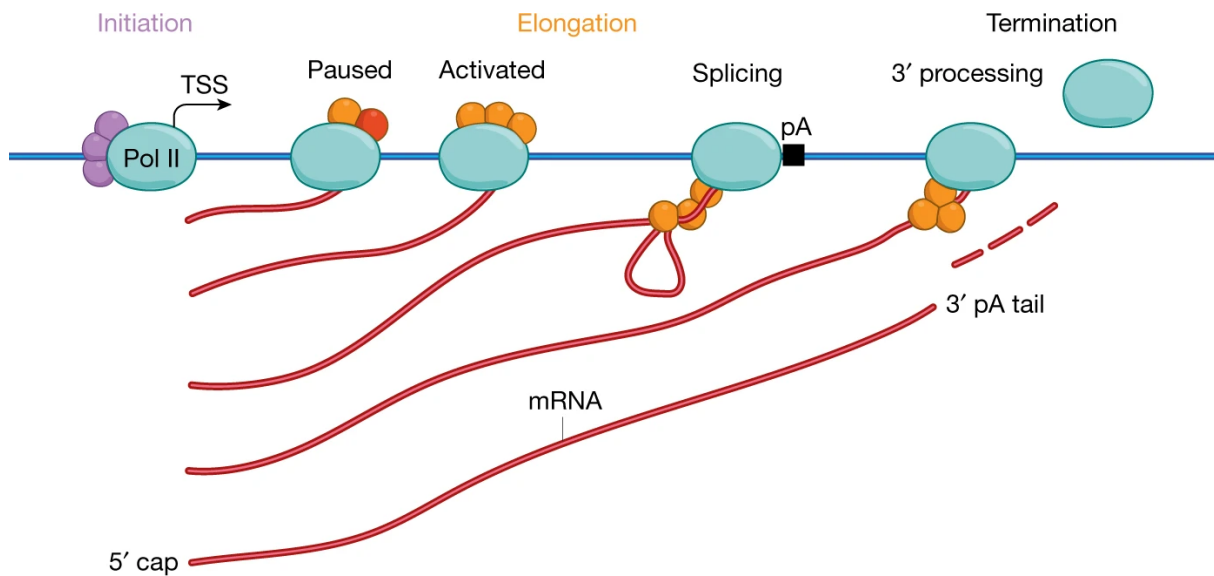


Figure 1.1.: **The transcriptional cycle.** mRNA transcription is a multi-stage process and the RNA polymerase II produces transcripts by iteratively cycling through initiation, elongation and termination phases. All stages are tightly regulated and the CTD of the moving polymerase serves as a platform for co-transcriptional modification processes, such as capping, splicing and 3' processing. Figure taken from Cramer (2019).

to fulfil their role as templates to proteins (Colgan and Manley, 1997).

Just like initiation and elongation, termination is a controlled, yet dynamic process. Premature termination, also referred to as transcription attenuation, is a strategy to discard unwanted transcripts, involved in taming pervasive transcription (Porrúa and Libri, 2015). Transcription attenuation has also been implicated to have a gene-specific regulatory role (Kim and Levin, 2011; Wagschal et al., 2012; Porrúa and Libri, 2015).

1.1.2. Promoters and enhancers

The roughly 20,000 protein coding genes encoded in the human DNA make up between 1–2% of the 3 billion base pairs of genetic information. This perplexingly low fraction has hit the scientific community by surprise (Claverie, 2001) and started a search for alternative explanations for bridging the wide complexity gap between the 1mm long nematode *C. elegans* with roughly the same number of genes and finding reasons for the vast amount of apparently unused genetic information. One possible answer was provided by mapping regulatory regions genome-wide with next-generation sequencing technologies. Around 3 million potentially regulatory regions were identified based on the accessibility of chromatin, allowing them to attract transcription factor binding (Thurman et al., 2012). Based on their overlap with annotated transcription start sites, regulatory regions are classified into promoters (core promoter and activating sequences upstream of a TSS) and enhancers (far from a TSS). As the general transcription factors bound to Pol II at the core promoter can drive transcription in-vitro (Roeder, 1996, 1998; Hahn, 2004), it is tempting to assign promoters and enhancers the distinct roles of basal transcription and expression level fine-tuning, respectively. Characteristic chromatin modifications and enrichment

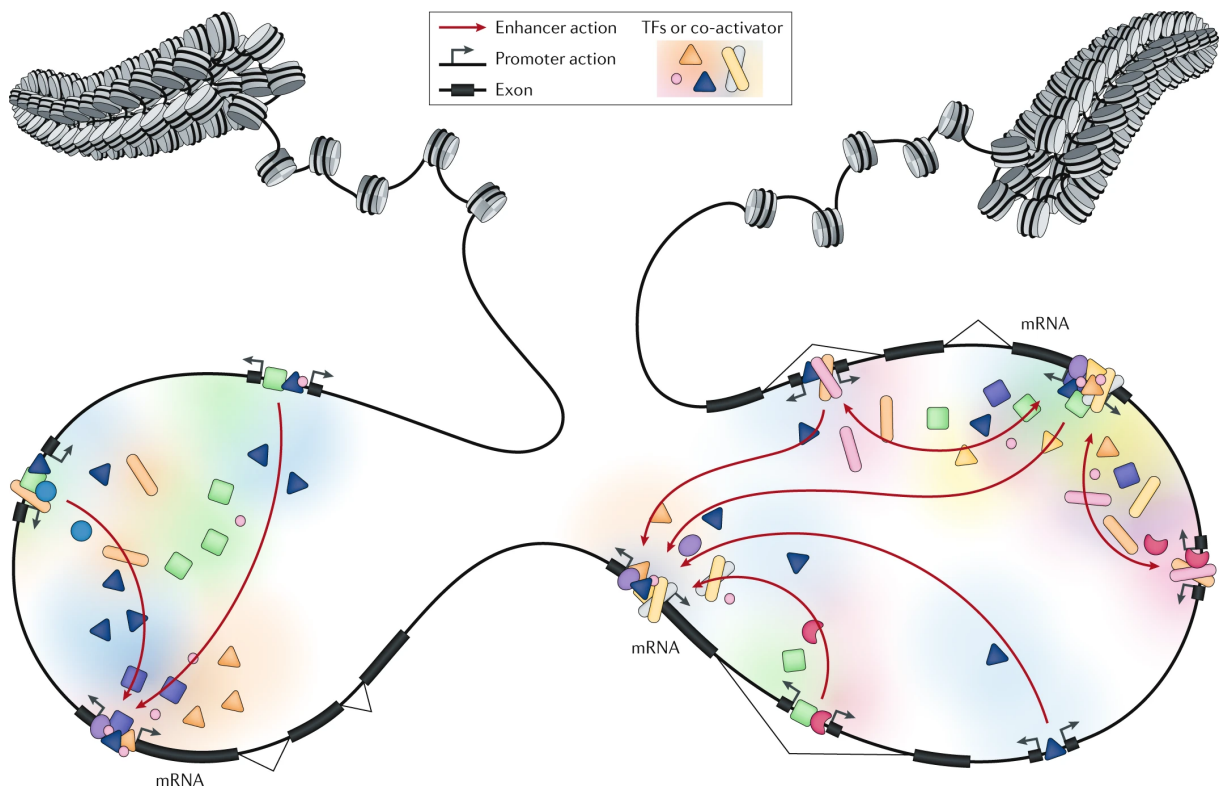


Figure 1.2.: **Interplay of regulatory elements regulate transcription.** According to our current understanding of gene regulation, the genome is structured into larger chromatin interaction domains (TADs), delimited by strong boundaries. Regulatory elements inside TADs share recruited transcription factors and co-factors within the same TAD by transient chromatin interactions, thereby accumulating the signals necessary to switch from low to bursts of high transcription levels. Figure taken from Andersson and Sandelin (2020).

of co-activator p300 specifically at enhancer regions supported the plausibility of a fundamental functional difference between promoters and enhancers (Heintzman et al., 2007, 2009). More recent findings such as transcriptional activity of enhancers (De Santa et al., 2010; Kim et al., 2010; Andersson et al., 2014), similarities in chromatin and sequence architecture (Koch et al., 2011; Core et al., 2014; Scruggs et al., 2015) and the ability of promoters to modulate the expression of distal genes (Rajagopal et al., 2016; Engreitz et al., 2016; Diao et al., 2017; Dao et al., 2017) challenge the notion of fundamental differences between promoters and enhancers (reviewed in Andersson and Sandelin (2020)). Ultimately, it is the context-specific interplay between a specific set of promoters and enhancers that determines the expression patterns of genes. While the underlying regulatory signals are encoded the genome sequence itself, deciphering this cis-regulatory code is still an ongoing challenge (Zeitlinger, 2020). I will discuss approaches to uncovering the cis-regulatory code in more detail in section 1.2.4.

1.1.3. Impact of chromatin architecture on gene regulation

If one would straighten and append the human DNA to one long single strand, its length would be approximately 2 meters long. The diameter of a typical human cell is 4 to 5 orders of magnitude smaller, and efficient packaging is thus crucial. While it has long been appreciated that

chromatin is organized into stable higher-order structures (Zink et al., 1998; Cremer and Cremer, 2001), methods based on next-generation proved powerful tools for advancing our understanding of the global genome architecture (reviewed in Kempfer and Pombo (2020)).

Chromatin architecture can be studied at different levels. At the finest level, DNA is packed into nucleosomes, 146 nucleotides of DNA wrapped around a histone octamer (Kornberg, 1974; Luger et al., 1997). When the DNA is tightly wrapped into nucleosomes, it is inaccessible to most transcription factors and is thus transcriptionally inactive (Magnani et al., 2011; Zaret and Carroll, 2011; Iwafuchi-Doi and Zaret, 2014). The three dimensional structure of chromatin is shaped by dynamic chromatin loops established by loop extrusion (Splinter et al., 2006; Fudenberg et al., 2017; Banigan and Mirny, 2020). Convergent CTCF binding sites on the DNA serve as road blocks for cohesin, a key component of the loop extrusion machinery and thereby encode the loop structure in the DNA (de Wit et al., 2015; Merkschlager and Nora, 2016; Pugacheva et al., 2020). Zooming further out, chromatin loops give rise to chromatin interaction domains, so called topologically associated domains (TADs) (Dixon et al., 2012). TADs are characterized by chromatin interacting more frequently inside two TAD boundaries than with other regions outside the TAD and their sizes range from tens of kilobases to few megabases (Dixon et al., 2012; Rao et al., 2014). TAD boundaries delimitate regulatory chromatin units and their disruption gives rise to ectopic regulatory interactions, thereby perturbing native gene regulation (Lupiáñez et al., 2015; Franke et al., 2016).

Studied at an even higher level, chromatin can be classified into active and inactive blocks, termed A and B compartments respectively. Chromatin interacts preferentially within blocks of the same compartment type, suggesting a function-based chromatin arrangement inside the nucleus (Lieberman-Aiden et al., 2009). The compartmentalization with respect to transcriptional activity at the highest level highlights the intimate relationship between chromatin architecture and transcription regulation. While the information of the chromatin structure is ultimately encoded in the sequence, the important influence of chromatin architecture further complicates the rules underlying transcription regulation. A visualization of our current understanding of the influence of chromatin structure and regulatory elements on gene-regulation is depicted in Figure 1.2.

1.1.4. Phase-separation in transcription

Many biological processes require the co-localization of a set of functionally related gene products at high concentration. To ensure high efficiency, some biological processes are performed in parallel in specialized reaction chambers, so called organelles, which are separated from the cytoplasm by membranes. Recently, liquid-liquid phase separation has been discovered as a general organisation principle of cells. Just like oil aggregates in water by de-mixing, the cytoplasm can de-mix and form biomolecular condensates that behave like membrane-less organelles (Hyman et al., 2014).

Weak multivalent interactions between unstructured, intrinsically disordered regions of proteins

play an important role in establishing biomolecular condensates (Lin et al., 2017). Incidentally, key players of the transcriptional cycle such as the activation domains of transcription factors, the CTD, and histone tails are unstructured and thus suggest involvement in biomolecular condensation.

In recent years evidence has accumulated that biomolecular condensates are a common phenomenon in transcription-related processes (reviewed in Sabari et al. (2020)). Among the processes covered in this thesis, Pol II clustering (Boehning et al., 2018), promoter-proximal pause release (Rawat et al., 2021), co-transcriptional mRNA processing (Guo et al., 2019; Chen and Belmont, 2019; Spector and Lamond, 2011; Kim et al., 2019), chromatin organisation (Gibson et al., 2019; Larson et al., 2017; Larson and Narlikar, 2018; Strom et al., 2017; Sanulli et al., 2019; Gibson et al., 2019; Wang et al., 2019; Li et al., 2020; Plys et al., 2019) and super-enhancers (Sabari et al., 2018; Boija et al., 2018; Cho et al., 2018) have been reported to have liquid-liquid phase separation as an underlying principle.

1.2. Transcription factors and the regulatory code

Just like the gene products themselves, their regulatory information is stored in the DNA sequence and needs to be decoded. Transcription factors (TFs) are specialized proteins that read-out regulatory information by binding to regulatory sequences in order to influence transcription (Fulton et al., 2009; Vaquerizas et al., 2009).

Transcription regulation is a complex process, characterized by the synergistic interplay between a large number of transcription factors which makes deciphering the mechanisms and logic of transcription factors and their underlying gene-regulatory networks a complex task. Nevertheless, some have been assigned distinct functions, which I will summarize in the following. (i) enabling and repressing transcription. As discussed in sections 1.1.2 and 1.1.3, transcriptionally inactive chromatin is generally inaccessible to most transcription factors. Pioneer TFs open chromatin up to facilitate recruitment of other factors (Magnani et al., 2011; Zaret and Carroll, 2011), whereas transcriptional repressors can prevent transcription factors from binding. Due to the default state of transcriptionally inactive chromatin, the dominant role of TFs in mammals has been attributed to transcription enhancement rather than repression with few exceptions (Thiel et al., 2004; Frum et al., 2019). Recently, a systematic screening suggested a prominent role of repressive silencer elements in mammalian genomes (Pang and Snyder, 2020) (ii) Chromatin architecture and promoter-enhancer pairing. As described in section 1.1.3, CTCF binding sites encode the chromatin loop positions in the genome. The most prominent TF in this category is CTCF, a TF known for its roles in transcription activation and repression by its ability to block promoter-enhancer interactions as a so called *insulator* (Bell et al., 1999; Kim et al., 2015). (iii) Recruiting and stabilising the transcription the machinery. The best studied TFs in this category are the general TFs and co-activators responsible for recruiting the polymerase and initiating transcription (see also section 1.1.1).

TFs gain their DNA binding ability by having one or more DNA binding domains. Many contain

additional effector domains in order to fulfil their role in modulating transcription. In-vivo, both DNA binding and the effect on transcription of TFs are synergistic (Reiter et al., 2017). By combinatorics alone, the estimated 1600 transcription factors encoded by the human genome offer an enormous regulatory toolkit for fine-tuning gene expression (Lambert et al., 2018).

A further layer of complexity is added by quantitative binding strength modulation by imperfect motifs. It is more and more appreciated that especially stochastic binding to weak, degenerate binding motifs play an important role in the fine-regulation of transcription (Crocker et al., 2016).

The context-specific interactions between chromatin architecture and accessibility, the synergistic, combinatorial recruiting and binding behavior of TFs and the importance of weak binding makes deciphering the logic behind the cis-regulatory code a defining challenge (Wasserman and Sandelin, 2004), yet to be cracked.

1.2.1. Molecular basis of DNA binding

DNA binding domains bind DNA via side-chain interactions between TFs and the DNA. Binding motifs of TFs are typically 6–12 nt in length in order to achieve sufficient specificity while maintaining flexibility (Lambert et al., 2018). The binding affinity to sequence motifs is achieved by a combination of sequence specific base readout and sequence unspecific shape readout (Rohs et al., 2010). Nucleotide specific hydrogen bonds between TF and nucleotides in the major groove of the DNA is the basis of the most efficient base-readout, but hydrogen bond formation in the minor groove and hydrophobic interactions are also used for base readout. Shape readout detects deviations of the DNA structure from the ideal B-DNA helix. DNA structure can be read out in the form of minor groove and major groove shape, local kinks, and global DNA bend, among other signatures (Rohs et al., 2010).

The DNA binding activity of TFs is provided by one or more evolutionary conserved DNA binding domains (DBD). Different structural elements for DNA binding have evolved (reviewed in Rohs et al. (2010)) and the 1700 human TFs obtain their specificity by mixing and matching these elements (Lambert et al., 2018). In the following I will briefly introduce three common structural motifs for DNA binding. (i) Helix-Turn-Helix (HTH) motif. One Helix serves as a "recognition helix" for specific base readout, whereas the other helix further stabilizes the binding. (ii) Basic Helix-Loop-Helix motif (bHLH). bHLH transcription factors using the bHLH strategy rely on dimerization. One alpha helix from each dimerization partner is involved in the base readout, enabling increased motif diversity by hetero-dimerization. (iii) Zinc finger (ZF). ZF domains bind DNA by coordinating a Zn^{2+} ion and have the size of approximately 30 amino acids. With 700 putative TFs, C2H2 ZFs play an important role in human transcription regulation (Weirauch and Hughes, 2011; Vaquerizas et al., 2009). Individual C2H2 ZF domains have specificity for around 3 nucleotides, with the specificity depending on the amino acid choice in variable positions in the ZFs domains (Najafabadi et al., 2015). In order to achieve sufficient specificity, ZF domains typically occur in arrays with human ZFs containing on average 10 ZF

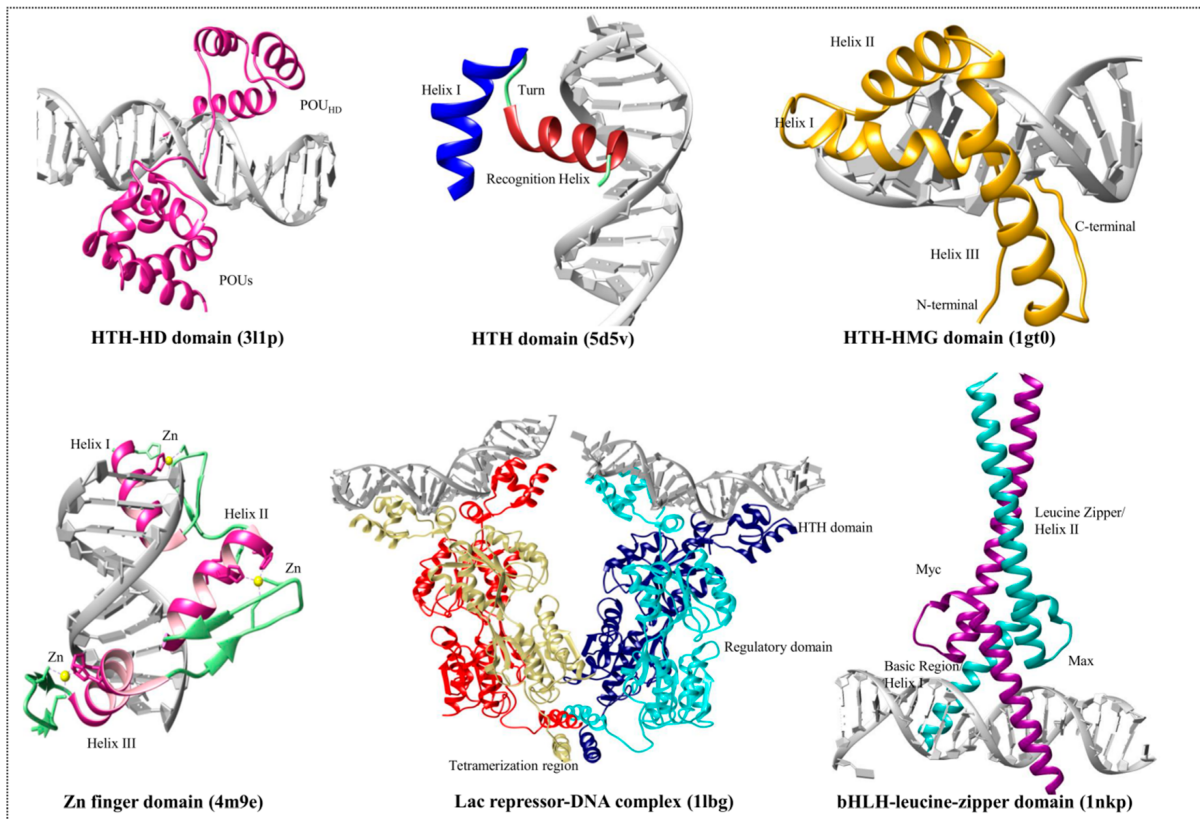


Figure 1.3.: **Proteins bind DNA by structurally conserved motifs.** The graphic visualizes the most common DNA binding strategies: the Helix-Turn-Helix (HTH) architecture, (first row and second row middle), basic Helix-Loop-Helix motif (bHLH), (second row, right) and Zinc fingers (second row, left). Figure taken from Yesudhas et al. (2017).

domains with only a subset of the domains binding at a time. The variable binding specificities of each domain and the combinatorial binding of individual ZF domains gives ZF proteins a high binding flexibility (Najafabadi et al., 2015). Examples of bound DNA-binding domains are shown in Figure 1.3.

Obtaining accurate, quantitative models of TF binding affinity is a key objective in bioinformatic studies of transcription factor binding. I will next discuss experimental methods used for identifying TF binding sites and quantifying their affinity and then introduce computational models for describing TF binding *in silico*.

1.2.2. Experimental methods for studying transcription factor binding

Transcription factor binding is typically studied in high-throughput *in-vivo* and *in-vitro* assays based on next-generation sequencing. Here I introduce chromatin immunoprecipitation DNA-sequencing (ChIP-seq) and high throughput systematic evolution of ligands by exponential enrichment (HT-SELEX) as widely-used representatives for studying TF binding *in-vitro* and *in-vivo*.

ChIP-seq is a widely-use *in-vivo* assay for studying the genomic binding sites of a protein of

interest (Johnson et al., 2007). Using a chemical crosslinking agent, transient interactions between proteins and DNA are stabilized in a large quantity of cells. A subsequent sonication step breaks the chromatin into small protein-bound fragments. Specific antibodies are used to enrich for fragments bound by the protein of interest. Finally all proteins are degraded and the DNA fragments are sequenced as a paired-end library. The genomic fragments bound by the transcription factor of interest can be identified by mapping the sequencing reads back to the genome.

As discussed earlier, the occupancy of potential transcription factor binding sites depends on the cellular context. By capturing real binding events, ChIP-seq is a powerful tool to unravel context-dependent transcription factor binding. Protocol inherent biases such as sonication biases, background binding, variable antibody quality, sequence-dependent PCR amplification, and mappability biases however make the quantitative interpretation of ChIP-seq signal challenging (Diaz et al., 2012; Park et al., 2013).

TF binding can also be studied in-vitro. Efficient high-throughput SELEX protocols allowed to characterize the binding preference of hundreds of transcription factors (Jolma et al., 2010, 2013). SELEX identifies preferential binding by iteratively amplifying and selecting preferentially bound sequences. The starting point for SELEX is a diverse pool of short random oligonucleotide sequences and a protein of interest, tagged for efficient pull-down. The DNA pool is then amplified via PCR and combined with the tagged proteins. Bound fragments are recovered by protein pulldown and amplified for a subsequent round. The sequence enrichment observed by comparing the DNA pool before and after each round allows to describe binding affinity by thermodynamic modelling (Ruan et al., 2017; Sakamoto et al., 2018).

HT-SELEX allows to quantify the affinity of TFs outside of their native chromatin context. As discussed earlier, TFs influence transcription in a highly synergistic manner, limiting the applicability of HT-SELEX methods in representing TF dynamics in the cell. This inherent neglect of context has been ameliorated in SELEX variants studying co-binding and binding nucleosomal DNA (Jolma et al., 2015; Zhu et al., 2018). Despite these advancements, it is important to keep in mind that deciphering the cis-regulatory code, i.e. dynamics of all involved TFs, is more complicated than the combined dynamics of all involved TFs.

In-vivo and in-vitro methods are thus complementary approaches studying different aspects of transcription factor binding. Due to their orthogonal approach, cross-platform validation is a powerful method for validating computational binding models (Weirauch et al., 2013).

1.2.3. Mathematical basis of motif models

Starting from a large number of preferentially bound sequences obtained from a TF-binding experiment, the computational challenge is to identify the DNA stretches bound by the TF and quantify their respective binding affinities. The Boltzmann distribution $p(\mathbf{x}) \propto \exp(-\frac{E(\mathbf{x})}{k_B T})$ provides the theoretical link between the statistically sampled bound sequences and their underlying binding affinities, where $p(\mathbf{x})$ denotes the probability of binding sequence \mathbf{x} , $E(\mathbf{x})$ is the

energy state of the TF bound to \mathbf{x} and k_B and T are Boltzmann’s constant and temperature, respectively. In the light of the exponential growth of possible binding motifs \mathbf{x} with the motif length, simplifying assumptions allow robust parameter estimation for modelling $E(\mathbf{x})$ and thus the binding affinities. As experimental TF-DNA binding measurements typically do not have the resolution to report individual binding sites, but longer bound fragments, the binding positions have learnt alongside the binding affinities. Expectation-maximization and Gibbs sampling are statistical frameworks that are frequently employed to jointly learn affinities and binding locations (Das and Dai, 2007).

As mentioned before, the DNA binding affinities are typically described by parametric statistical models. The most common assumption is that all positions in the motif contribute independently and additively to the total binding affinity. For a binding motif of length L , this assumption reduces the number of independent parameters to $3 \times L$. In additive models, the binding affinity is typically represented as a position weight matrix (PWM) (Stormo et al., 1982; Stormo, 2000). Using the PWM as a lookup table, affinities of sequences can be calculated by summing the weights w_{ia} of each the nucleotide a at each sequence position i . Despite the gross simplification in the independence assumption, PWM models have proven fairly accurate for describing the binding energy for most transcription factors (Benos et al., 2002; Zhao and Stormo, 2011).

It has long been known that the dependency assumption does not represent a biological truth. Correlations between neighboring nucleotides can arise due to single amino acids in TFs binding to more than one nucleotide and the influence of the DNA sequence on the local DNA structure (Luscombe et al., 2001; O’Flanagan et al., 2005). As PWMs arise as the special case of memoryless Markov models, higher-order Markov models have been proposed to model the binding affinities more accurately at the expense of a higher model complexity (Siddharthan, 2010; Zhao et al., 2012; Kulakovskiy et al., 2013; Siebert and Söding, 2016). In higher-order Markov models the contribution a nucleotide makes to the total binding affinity depends on the preceding nucleotides. Whereas a PWM as zeroth-order Markov model models $4^1 - 1 = 3$ independent parameters per position, a first-order Markov model considers dinucleotides and thus requires $4^2 - 1 = 15$ parameters per position. More generally, a higher-order model of order k has $L \times 4^{k+1} - 1$ independent parameters.

As the number of parameters increases, higher-order models are prone to overfitting and capturing complex biases in the datasets. In the past our group developed Bayesian Markov models that implicitly adapt the model complexity to the available data and therefore allow training higher-order models, effectively mitigating the risk of overfitting (Siebert and Söding, 2016). By training on in-vitro and testing on in-vivo data or vice versa, it has been shown that complex models can outperform simple models (Alipanahi et al., 2015; Siebert and Söding, 2016; Ge et al., 2021).

1.2.4. Computational approaches for uncovering the cis-regulatory code

Traditional approaches to TF binding use motif models such as PWMs or higher-order Markov models to describe the binding preference of individual transcription factors based on the statistical overrepresentation of their short binding motif sequences. As discussed previously, cooperativity is a key feature of transcription factor binding and activation, and understanding the interplay between motifs by order, orientation, spacing and individual affinities is crucial for understanding the cis-regulatory code (Farley et al., 2015, 2016). With their ability to derive generalizable models on large amounts of complex data, end-to-end differentiable deep neural networks (DNNs) have proven powerful models for describing transcription factor binding (Alipanahi et al., 2015; Kelley et al., 2016; Avsec et al., 2021b; Eraslan et al., 2019). Convolutional neural networks, originally developed in the computer-vision field, have proven especially suitable, due to the similarity of individual convolutional kernels with PWMs. Instead of learning just one PWM at a time, deep neural networks predict experimental read counts based on a complex combination of hundreds of convolutional kernels and can thereby capture the complex interplay between TFs binding sites. Instead of overrepresentation, motifs from DNNs can be derived by quantifying the contribution each base makes to the final prediction (Shrikumar et al., 2017; Avsec et al., 2021b).

A novel convolutional DNN with base-pair resolution recently extracted the soft syntax rules underlying the binding preferences of pluripotency TFs, highlighting the depth of biological insights that can be drawn from supervised training of DNN models (Avsec et al., 2021b).

1.3. Motivation aims and goals

Our group previously introduced Bayesian Markov Models (BaMMs) as a class of higher-order motif models that are not susceptible to statistical overfitting and showed that BaMMs outperform PWM models in detecting bound sequences and quantifying binding affinities (Siebert and Söding, 2016). While the original paper provided a proof-of-principle of BaMMs and made all code and data publicly available, the target audience for the tool were bioinformaticians and computationally versed biologists, well acquainted with the command line and de-novo discovery of TF motifs. The aim of this project was to make BaMMs widely accessible to the scientific community by not only developing an intuitive web interface, but also offering easy-to-use common workflows that simplify searching, annotating, evaluating and comparing motifs.

2. Methods

2.1. Seeding stage PEnG

2.1.1. The PEnG algorithm

We developed the PEnGmotif for finding enriched motifs as initialization for our higher-order refinement to Bayesian Markov Models (BaMMs). The PEnG algorithm is motif-centered and with its 6 consecutive steps aims to identify enriched motif patterns, reduce redundancy, sharpen the motifs' information content and merge shifted and overlapping motifs. An important feature of PEnG is that the information encoded in the input sequences only enters in form of k -mer counts. Therefore the runtime scales linearly in the total number of sequences and exponentially in the maximum pattern length K . As k is a user-defined constant (usually $K = 8$ or $K = 10$), PEnGmotif is especially suited for very large datasets.

Counting stage

The counting stage is the first stage of PEnGmotif. A sliding window of length K is shifted over the sequences and the total number of occurrences of each K -mer is counted and stored in a 4^K element array. This is done separately for input sequences and background sequences. When very large sets of background data exists, empirical background probabilities can be estimated by dividing the 4^K K -mer counts on the background set by the total number of counts. As background sequences are not always available even less so in large amounts, we model the background probabilities with a 2nd order Markov model by default. on either the background sequences or the input sequences. The output of the counting stage are two 4^K -element long vectors, one with the K -mer counts of the input sequences and one with the probabilities of observing each K -mer in non-specific binding events.

Scoring stage

The scoring stage assigns each K -mer X a z -score $Z(X)$ according to its binding potential. The z -score measures a scaled difference of the observed K -mer counts with the expected counts if there were only unspecific binding events. A K -mer with a z -score below 0 has been observed less often than expected by the unspecific binding, whereas highly enriched K -mers have large positive z -scores. The z -score is derived by modelling the K -mer counts by a Poisson distribution

with mean $\mu(X) = p_{bg}(X) \times L$ (Equation 2.1). All patterns that surpass a user-defined threshold (by default $Z_{thresh} = 10$) are selected as a set of K -mers bound with high confidence.

$$Z(X) = \frac{n(X) - L \times p_{bg}(X)}{\sqrt{L \times p_{bg}(X)}} \quad (2.1)$$

Local optimization stage

DNA binding is a statistical process and the protein occupancy of a stretch of DNA depends on its binding affinity. Depending on the statistical power, K -mers sufficiently similar to the optimal K -mer will also show enriched counts, albeit at a lower significance level. A single binding domain therefore creates not one, but a local neighborhood of enriched K -mers. We use the local optimization stage to reduce this redundancy and go from enriched K -mers to motif patterns. We do this by finding local optimal patterns that cannot achieve a higher score by replacing any of its positions with a letter from the degenerate IUPAC alphabet $\mathcal{D} = \{A, C, G, T, R, Y, S, W, K, N\}$, where $R = A$ or G , $Y = C$ or T , $S = G$ or C , $W = A$ or T , $K = G$ or T and $N = A$ or C or G or T . In addition to reducing redundancy, the degenerate alphabet allows to remove uninformative positions by introducing N and describing positions with weaker preference (e.g. R and Y for purine and pyrimidine bases respectively). We implemented three different scoring functions for the local optimization: maximization of pattern enrichment, minimization of log p-value and maximization of mutual information between the pattern X being an input sequence and an average input sequence \bar{X} containing at least one pattern match. By their construction, the three different scoring methods encourage different levels of degeneracy: pattern degeneracy is strongly discouraged for enrichment scoring and highly encouraged by the mutual information score. With the p-value scoring lying somewhere in between.

PWM conversion stage

In the PWM conversion stage locally optimal degenerate patterns are converted to PWMs. We offer two conversion schemes: in the simple scheme the probability of nucleotide a at position j is chosen as the relative frequency of the counts of all k -mers matching the IUPAC pattern.

$$p_{ja} = \frac{n_{ja}}{\sum_{b \in \{A, C, G, T\}} n_{jb}} \quad (2.2)$$

In order to prevent over-specific PWMs, we add uniform pseudocounts to the aggregated pattern counts n_{ja} (by default 10 pseudocounts).

With the assumption that optimally bound patterns with a single mutation still have a high binding affinity to the protein of interest, we can derive a more advanced PWM conversion scheme:

$$p_{ja} = \frac{n(y_{0:j-1} a y_{j+1:W-1})}{n(y_{0:j-1} N y_{j+1:W-1})} \quad (2.3)$$

Here K -mers are denoted in element-wise vector notation $\mathbf{y} = y_0 y_1 \dots y_{K-1}$. Note that this conversion scheme does not require pseudocounts and works especially well for long informative motifs typical for many transcription factors. By default we use the advanced PWM conversion scheme.

PWM sharpening stage

PWMs derived in the previous stage consider all K -mers that match the pattern with equal weight. By weighting the contribution of each K -mer by their binding affinity, we can further improve the PWM model. Starting the pattern-derived PWM, we use the expectation-maximization framework to iteratively refine the seed PWMs.

We derive the likelihood ratio of observing a K -mer under motif and background PWM model with parameters \mathbf{p} and \mathbf{p}_{bg} , respectively, as follows:

$$\frac{p(\mathbf{x}|\mathbf{p})}{p(\mathbf{x}|\mathbf{p}_{\text{bg}})} = \prod_{j=0}^{K-1} \frac{p_j(x_j)}{p_{\text{bg}}(x_j)} \quad (2.4)$$

In the expectation step we calculate for each K -mer the probability of being bound by the factor.

$$r(\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{p})/p(\mathbf{x}|\mathbf{p}_{\text{bg}})}{\sum_{\mathbf{x}' \in \{A,C,G,T\}^K} n(\mathbf{x}') p(\mathbf{x}'|\mathbf{p})/p(\mathbf{x}'|\mathbf{p}_{\text{bg}})} \quad (2.5)$$

In the maximization step we use the previously calculated expectations to update the model parameters \mathbf{p} .

$$p_j(a) = \sum_{\mathbf{x} \in \{A,C,G,T\}^W} I(x_j = a) n(\mathbf{x}) r(\mathbf{x}) \quad (2.6)$$

Plugging 2.5 in 2.6 gives the expectation-maximization iteration equation up to a normalization constant.

$$p_j^{(t)}(a) \propto \sum_{\mathbf{x} \in \{A,C,G,T\}^K} I(x_j = a) n(\mathbf{x}) \frac{p(\mathbf{x}|\mathbf{p}^{(t-1)})}{p(\mathbf{x}|\mathbf{p}_{\text{bg}})} \quad (2.7)$$

To model further model saturation at sites with very high affinity, we limit the likelihood ratio to a maximum of $A = 1000$ with a smooth decay and obtain the final iteration equation.

$$p_j^{(t)}(a) \propto \sum_{\mathbf{x} \in \{A,C,G,T\}^W} I(x_j = a) n(\mathbf{x}) \left(A^{-1} + \frac{p(\mathbf{x}|\mathbf{p}_{\text{bg}})}{p(\mathbf{x}|\mathbf{p}^{(t-1)})} \right)^{-1}. \quad (2.8)$$

Merging stage

In the merging stage, PWMs representing shifted instances of the same motif are combined. For the two PWMs $p^{(m)}$ and $p^{(m')}$ of length l and l' , respectively, we derive a similarity score $S(p, p')$ as the maximum overlap score when shifting the motifs by offset $d = -2, -1, \dots, l' - l + 2$ with respect to each other.

$$S(p^{(m)}, p^{(m')}) = \max_{-2 \leq d \leq l' - l + 2} \left\{ s(p_{j_1:j_2}^{(m)}, p_{j'_1:j'_2}^{(m')}) \right\} \quad (2.9)$$

Where $j_1 = \max\{0, d\}$, $j_2 = \min\{l - 1, l' - 1 + d\}$ and $j'_1 = \max\{0, -d\}$, $j'_2 = \min\{l' - 1, l - 1 - d\}$ denote the boundaries of the aligned overlap segment for p and p' , respectively. We measure the overlap similarity by the background-aware similarity function $s(p, p')$:

$$s(p, p') = \frac{1}{2} \left(d(p, p^{(\text{bg})}) + d(p', p^{(\text{bg})}) \right) - d(p, p'), \quad (2.10)$$

with the background nucleotide distribution $p^{(\text{bg})}$ and the distance function $d(p, p')$ defined as the sum of the KullbackLeibler distances $H(p||\bar{p})$ and $H(p'||\bar{p})$, with $\bar{p} := (p + p')/2$ defined as the average distribution of p and p' .

$$d(p, p') = \sum_{j=0}^{l-1} (H(p||\bar{p}) + H(p'||\bar{p})) = \sum_{j=0}^{l-1} \sum_{a \in \{A, C, G, T\}} (p_{ja} \log_2 p_{ja} + p'_{ja} \log_2 p'_{ja} - 2\bar{p}_{ja} \log_2 \bar{p}_{ja}). \quad (2.11)$$

$s(p, p')$ reaches its maximum for highly similar overlaps ($d(p, p') \approx 0$) where both overlap sequences are dissimilar to the background sequences ($d(p, p^{(\text{bg})}) + d(p', p^{(\text{bg})}) \gg 0$).

With a similarity score for two PWMs in hand, the iterative merging routine is defined as follows: (1) calculate pairwise similarity scores for all PWMs. (2) As long as there exists a pair (p, q) that surpasses a defined similarity threshold (e.g. $0.75 \times K$ bits), merge and remove p and q . (3) Update the distances of the newly merged PWM to all remaining PWMs. Go back to (2) until none of the pairs surpasses the similarity threshold.

PWM pairs $(p^{(m)}, p^{(m')})$ are merged position-wise aligned with the shift d_{max} that achieved the highest similarity score $S(p^{(m)}, p^{(m')})$ with following strategy: Non-overlapping positions take the nucleotide distribution from the PWM that has the overhang. For overlapping positions the joint distribution is calculated as the re-normalized sum of the probabilities in the aligned columns weighted by the total occurrences of the pattern that gave rise to each PWM.

At the end of the merging routine, all PWM with sufficient similarity have been combined and thus the output is a set of non-redundant enriched motifs as PWM models.

Derivation of local optimization scores

We developed three objective functions for the local optimization stage: enrichment optimization, mutual information optimization and p-value optimization. The objective functions score the discrepancies in observed and expected K -mer counts. The counts for degenerate patterns are the the sum of counts of all K -mers matching the degenerate patterns.

p-value score. Just like the z-score definition, the p-value score is based on a Poisson model of how many counts one would expect in absence of specific binding events. Having observed a pattern n times in the input set with an estimated mean of $\mu = p_{bg} \times L$ counts in a size-matched unspecific binding set, we derive a p-value for enriched patterns as follows:

$$\begin{aligned}
 \text{P-value} &= \sum_{k=n}^{\infty} \frac{\mu^k}{k!} e^{-\mu} \\
 &= \frac{\mu^n}{n!} e^{-\mu} \sum_{k=0}^{\infty} \frac{\mu^k}{(n+1) \cdots (n+k)} \\
 &\lesssim \frac{\mu^n}{n!} e^{-\mu} \sum_{k=0}^{\infty} \frac{\mu^k}{(n+1)^k} \\
 &\approx \frac{\mu^n}{n!} e^{-\mu} \frac{1}{1 - \mu/(n+1)} \\
 \log \text{P-value} &\approx n \log \frac{\mu}{n} + n - \mu - \frac{1}{2} \log(2\pi n) - \log \left(1 - \frac{\mu}{n+1} \right). \tag{2.12}
 \end{aligned}$$

As we only have to calculate p-values for enriched motifs ($\mu \ll n+1$), we can use the closed form solution of the geometric series in the third line. We transform the p-value into log space for numerical stability and apply Stirling's approximation to enable efficient computation.

In the local optimization stage we minimize the the log p-value, resulting in the statistically most significant locally optimal pattern.

Mutual information score. Instead of finding the statistically most significant motif, with the mutual information score we seek to find a motif that best distinguishes input from background sequences. Given a set of N input sequences of average length L and a pattern of length K that matches M_{obs} times out of $M := N(L - K + 1)$ possible positions in the input set and M_{exp} times in a size-matched background set, the empirical probability of the pattern matching any position in the input and background set are denoted as \tilde{p}_{obs} and \tilde{p}_{exp} :

$$\begin{aligned}\tilde{p}_{obs} &= \frac{M_{obs}}{M} \\ \tilde{p}_{exp} &= \frac{M_{exp}}{M}\end{aligned}$$

We can now approximate the probability of observing at least one pattern match in a sequence of average length in the input (p_{obs}) and background (p_{exp}) set:

$$\begin{aligned}p_{obs} &= 1 - (1 - \tilde{p}_{obs})^{L-K+1} \\ &\approx 1 - e^{-\tilde{p}_{obs}(L-K+1)} \\ &= 1 - e^{-M_{obs}/N} \\ p_{exp} &= 1 - (1 - \tilde{p}_{exp})^{L-K+1} \\ &\approx 1 - e^{-\tilde{p}_{exp}(L-K+1)} \\ &= 1 - e^{-M_{exp}/N}\end{aligned}$$

We further define the two random variables X and Z as follows.

$$\begin{aligned}X &= \begin{cases} 0 & \text{iff sequence contains no match to pattern} \\ 1 & \text{iff sequence contains at least one match to pattern} \end{cases} \\ Z &= \begin{cases} 0 & \text{iff sequence is a background sequence} \\ 1 & \text{iff sequence is an input sequence} \end{cases}\end{aligned}$$

With the mutual information score we seek the pattern that has the highest mutual information between X and Z . Intuitively speaking we seek the pattern for which the binding probability on an average sequence carries the most information about whether that sequence is an input or background sequence. Thus in short the pattern that best distinguishes input from background sequences.

The mutual information between X and Z is defined as

$$\begin{aligned}\text{MI} &= \sum_{Z \in \{0,1\}} \sum_{X \in \{0,1\}} p(X, Z) \log \frac{P(X, Z)}{P(X)P(Z)} \\ &= \sum_{Z \in \{0,1\}} \sum_{X \in \{0,1\}} p(X|Z) \log \frac{P(X|Z)}{P(X)} P(Z)\end{aligned}\tag{2.13}$$

With $q := p(Z = 1)$ we have by definition of X and Z :

$$\begin{aligned} p(X = 1|Z = 1) &= p_{obs} \\ p(X = 1|Z = 0) &= p_{exp} \\ p(X = 1) &= p(X = 1|Z = 1)p(Z = 1) + p(X = 1|Z = 0)p(Z = 0) \\ &= p_{obs}q + p_{exp}(1 - q) \end{aligned}$$

Defining further $p := p_{obs}q + p_{exp}(1 - q)$, and plugging into equation 2.13, we obtain:

$$\begin{aligned} \text{MI}(q) &= q \left[p_{obs} \log \frac{p_{obs}}{p} + p_{obs} \log \frac{1 - p_{obs}}{1 - p} \right] \\ &\quad + (1 - q) \left[p_{exp} \log \frac{p_{exp}}{p} + p_{exp} \log \frac{1 - p_{exp}}{1 - p} \right] \\ &= -qH(p_{obs}) - (1 - q)H(p_{exp}) + H(p) \end{aligned}$$

Where $H(X) := -X \log X - (1 - X) \log (1 - X)$ is defined as the entropy.

As we do not know the fraction of bound sequences $p(Z = 1)$, we build a heuristic score by summing over three values $q \in \{0.5, 0.1, 0.01\}$ for three broad regimes of binding. Furthermore we normalize the mutual information $\text{MI}(q)$ by the entropy $H(q)$ to derive at the final optimization score S_{MI} which we maximize in the local optimization.

$$S_{MI} := \sum_{q \in \{0.5, 0.1, 0.01\}} \frac{\text{MI}(q)}{H(q)}$$

Enrichment score. The enrichment score maximizes the ratio of the number of observed divided by expected pattern counts. Especially for expectation values much smaller than 1, the pure enrichment is susceptible to noise due to the discrete nature of counts. To prevent infrequent patterns obtaining high enrichment scores, we add pseudocounts to the expectation. The number of pseudocounts scale with the number of sequences and are controlled by a strength parameter f_{psdc} (by default $f_{psdc} = 0.005$).

$$S_{Enrich} := \frac{n}{(p_{bg} \times L + f_{psdc} \times N)}$$

The enrichment score is maximized in the local optimization stage. By its definition the enrichment score discourages pattern degeneracy. With $K = 4$ or $K = 6$, it is especially useful for detecting very short enriched K -mers which can be found in high-throughput DNA/RNA binding sets such as PAR-CLIP or eCLIP.

3. Results

3.1. PEnGmotif

By design the expectation-maximization (EM) framework offers motif refinement. In each iteration the algorithm proposes an update to the model parameters that increases the likelihood, thereby guaranteeing convergence to a local maximum of the likelihood function, but not necessarily the global optimum. In general the landscape of the likelihood can be rugged with multiple local maxima representing different enriched motifs, thereby making the strategic choice of initial seed parameters an important consideration. We developed the PEnGmotif algorithm as a fast tool for iterating promising seed PWMs for the higher-order refinement. Users can then select the most promising seeds for higher-order optimization. As the EM will only output locally optimal motifs, biological relevance ultimately has to be judged by comparative motif analysis and motif performance evaluations.

3.1.1. PEnGmotif on artificial sequences

In order to understand the individual stages of PEnGmotif better and provide a proof of principle, we use PEnGmotif to recover implanted motifs in simulated sequences. To this end, we simulate 10000 150nt long nucleotide sequences with a second order Markov model, thereby determining the background 3mer sequences. With a probability of 50% we implant a human CTCF motif according to the PWM model MA0139.1 from the JASPAR database (Fornes et al., 2020) by selecting a motif position with uniform probability. We then apply PEnGmotif to recover the CTCF motif as a potential seed for higher-order refinement.

A visualization of PEnGmotif's processes and outputs for pattern sizes 8, 10 and 12 are presented in the following. In the first step, kmers are counted on the input data and the most enriched kmers – also referred to as base patterns – are selected for further IUPAC optimization (Figure 3.1A). Depending on dataset size, pattern size and motif lengths, hundreds to thousands of base patterns can be found. For ease of visualization, only the fate of the five base patterns with highest z-score are visualized. Note also that unlike the EM framework, PEnGmotif's kmer centered approach is oblivious to motif positioning. The visual pattern alignment in the first two stages are for the ease of the reader, but unknown to the algorithm.

In the local optimization stage, base patterns are optimized to IUPAC patterns using a degenerate nucleotide alphabet (Figure 3.1B). Here we use mutual information, PEnGmotif's default optimization score. Patterns with same alignment with respect to the motif tend to be optimized

to the same IUPAC patterns, thereby drastically reducing the number of patterns that have to be considered for the downstream stages. The three different objective functions available during the local optimization routine are visualized in Figure 3.2.

In the PWM conversion stage, distinct IUPAC patterns are converted to PWMs, by taking into account the counts of all base patterns matching each IUPAC pattern (Figure 3.1C). Compared to the implanted motif, the information content of the inferred PWMs is lower due to the noise arising from unbound k-mers matching degenerate IUPAC patterns by chance.

In order to overcome the information gap, we use the previously PWMs in a EM algorithm that quantifies how strongly the base patterns are bound and thereby sharpens the motif by increasing the motif content (Figure 3.1D). In this example the information content of the sharpened PWMs slightly surpasses the information content of the implanted motif.

Lastly, all PWMs with significant overlaps are merged to a single motif that resembles the implanted motif, albeit with higher information content, thereby over-specifying bound sequences (Figure 3.1E). Seeded with the merged PWM, the position-aware motif optimization of the refinement EM can quickly converge to a realistic motif model.

3.1.2. PEnGmotif on real sequences

In order to validate the reliability of PEnGmotif in detecting seeds in in-vivo data, we applied PEnGmotif with pattern length 10 to CTCF peaks called from ENCODE (Wang et al., 2012; Consortium et al., 2012) ChIP-seq data in the five well-studied cell lines Hct116, Huvec, K562, Mcf7, Wi38 provided by the GTRD project (Yevshin et al., 2019) (Figure 3.3). In all data sets the 4–5 discovered motifs are consistent with a merged, 14 bp long main motif and 2–3 shorter submotifs that did not have sufficient similarity to the main motif and thus have not been merged. Moreover, a motif representing long A-T rich stretches that due to their length have not been corrected out by the second-order background model.

3.1.3. Time benchmark

In contrast to EM based motif discovery tools, PEnGmotif’s k-mer based approach allows high speed, a strength that makes it especially suitable for seed discovery (Figure 3.4). For typical ChIP-seq peak data sets with between 10,000 and 100,000 sequences, PEnGmotif requires less than a second for pattern length $k=8$, around 17 seconds for $k=10$ and around 5 minutes for $k=12$. At this data set size regime, the total runtime is dominated by the pattern length rather than number of sequences and sequence length. For very large sequence sets, processing and counting the sequences becomes the bottleneck. The largest test set with 150m sequences, contains about an order of magnitude more nucleotides than the human genome. PEnGmotif requires around 20 minutes on this dataset.

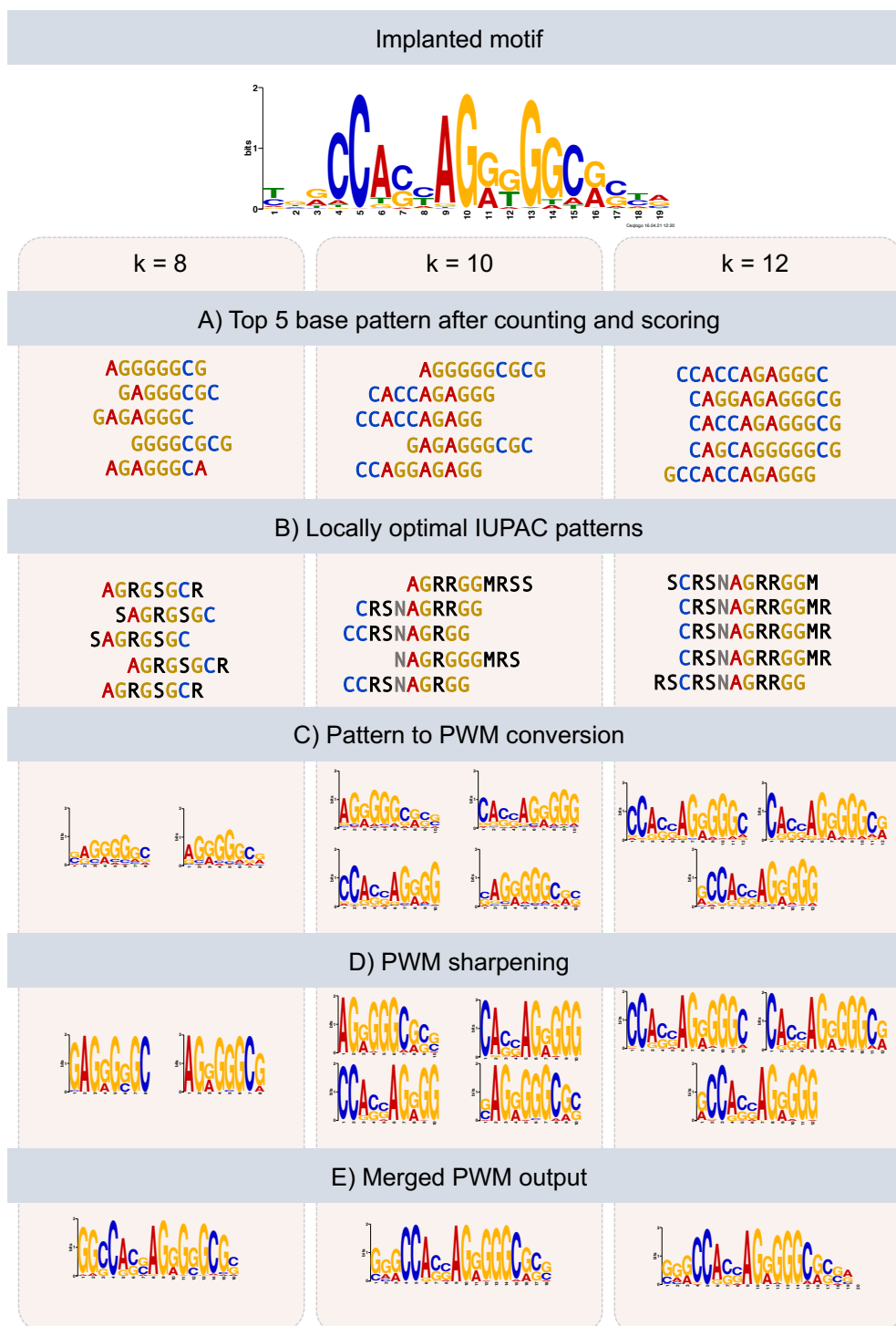


Figure 3.1.: **PEnGmotif recovers implanted motifs on simulated sequences.** We use PEnGmotif with k-mer length 8, 10, 12 to recover implanted CTCF motifs. Here we show the results of the PEnG algorithm on the five highest enriched base pattern k-mers and the PWM output **A)** By counting and ranking the k-mers by their z-score under a Poisson model, the highest ranking base patterns are selected for local optimization. Here the top 5 base patterns are depicted aligned to their respective motif position. **B)** Local optimization combines base patterns to enriched IUPAC patterns by detecting variable sites. **C)** IUPAC patterns are converted to PWM by combining the counts from matching base patterns. **D)** The information content of PWMs is increased by iterating the EM algorithm. **E)** The optimized PWMs are combined based on their overlap, resulting in one long PWM, resembling the implanted motif.

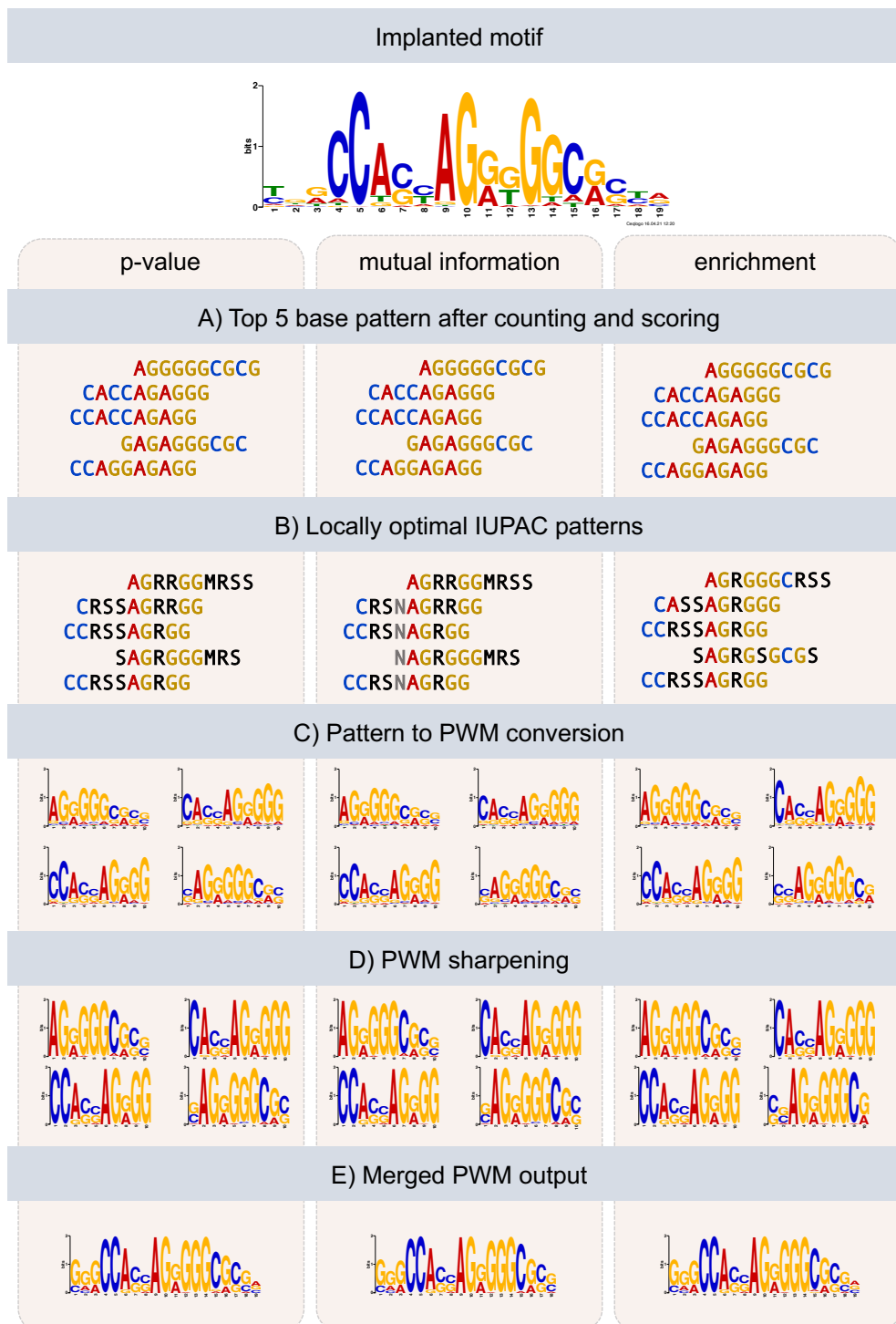


Figure 3.2.: **Local optimization strategies produce slightly different EM seeds.** We use PEnGmotif with k-mer length 10 to recover implanted CTCF motifs with our three different IUPAC pattern optimization goals: p-value, mutual information and pattern enrichment. Panels A–D as described in Figure 3.1. The three different objective functions lead to different levels of IUPAC pattern degeneracy with enrichment optimization. The degeneracy is lowest for enrichment optimization and highest for mutual information optimization. In this case the thereby introduced differences in the PWMs generated in the PWM conversion stage are small and the EM in the PWM sharpening step converges to the same output motif.

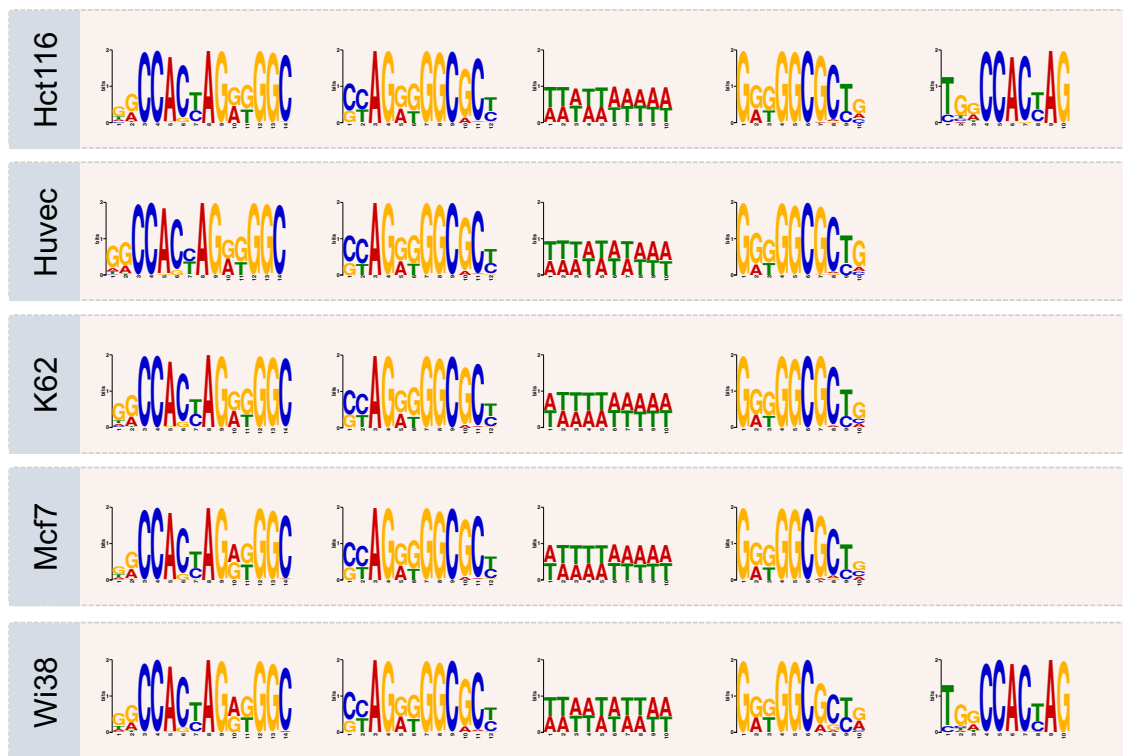


Figure 3.3.: **PEnGmotif's CTCF motifs are consistent across ENCODE ChIP-seq motifs.** When applied to real ChIPseq data from well-studied ENCODE cell lines, PEnGmotif discovers consistent motifs, with one 14bp long main motif, 2–3 unmerged submotifs and a motif representing overrepresented long A-T rich nucleotide stretches.

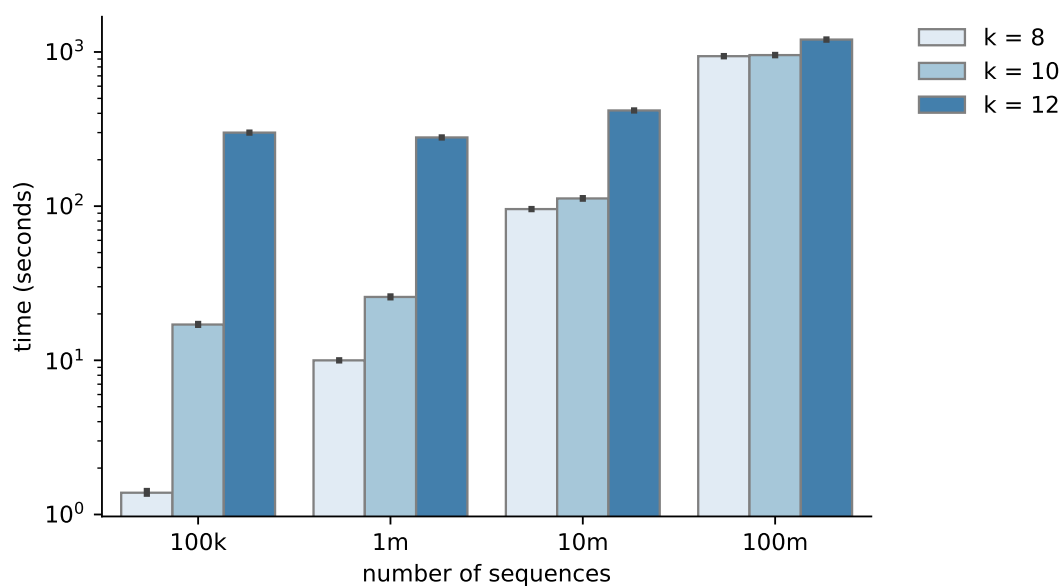


Figure 3.4.: **PEnGmotif runtime benchmark** PEnGmotif processes data sets of size of typical ChIP-seq peaks in less than a second for pattern length $k=8$, around 17 seconds for $k=10$ and around 5 minutes for $k=12$. At these data set sizes, the choice of the pattern length dominates the the total runtime. PEnGmotif can process data sets orders of magnitudes larger than typical data sets at acceptable speed.

3.2. Five tools for motif analysis

3.2.1. De-novo motif discovery

De-novo motif discovery is the core workflow of our web server. We implemented motif discovery as a process consisting of two consecutive stages: a seeding stage followed by higher order refinement.

In the seeding stage users the PEnGmotif algorithm is applied to a file with short nucleotide sequences in fasta format uploaded by the user. The PEnGmotif algorithm is highly configurable and many of the options are exposed to the user in our web interface (Figure 3.5). The default parametrization is well suited for processing peaks from most DNA ChIP-seq experiments. Users can optimize the motif detection to RNA binding experiments (e.g. PAR-CLIP/iCLIP/eCLIP) by switching to single-stranded motif detection and if necessary increase sensitivity for shorter motifs by reducing the pattern length, lowering the z-score threshold for required k-mer enrichments and experimenting with lower background orders. Users with in-vitro sequencing data (e.g. HT-SELEX) can provide sequences from the input library as external background set. The remaining options are situational and can be experimented with for fine-tuning results: users can remove spurious significant patterns by increasing the minimum number of pattern occurrences, can choose the metric for local optimal IUPAC pattern generation and deactivate EM optimization of IUPAC patterns.

After job processing the user is redirected to a result page that lists for each discovered PWM seed, (1) its consensus IUPAC string, (2) its sequence logo with reverse complement, (3) its motif AvRec score and (4) an estimation of the fraction of input sequences containing the seed motif (Kiesel et al. Figure 2A). Additionally, users can download the motifs in MEME format for further processing with third party tools. Based on the users' prior knowledge about the expected motif or the given occurrence and motif performance scores, they can select promising motifs as seeds for higher-order refinement. Users without prior knowledge can also choose non-interactive seeding, which automatically chooses the best performing motifs for higher-order refinement.

In the refinement stage, the BaMMmotif algorithm is applied iteratively to the user-provided sequences using each selected seed as a motif initialisation to train powerful higher-order models. By default 2nd order Bayesian Markov Models are trained, but users can choose to reduce the order. Users can also choose to extend the refined BaMM model by adding flanking nucleotides to the core seed.

Just like the seeding procedure, higher-order refinement is submitted as a job and a result page is shown after completion. In addition to the IUPAC string, zeroth-order motif logos and performance statistics, an additional section shows up to two higher-order logo plots for 1st and 2nd order contributions. For each position a higher-order logo visualize the additional information contributed by the respective context kmers (Kiesel et al. Figure 2B).

General settings

1 Search on both strands:

1 Background Sequences:

No file chosen

1 Background Model Order:

Seeding stage

1 Pattern Length:

1 Z-Score Threshold:

1 Count Threshold:

1 IUPAC Optimization Score:

1 Skip EM:

Figure 3.5.: **PEnGmotif is highly parameterizable in our web server UI.** Our web server allows users to tailor the seeding stage to their needs by exposing a many of PEnGmotif’s commandline options to the user via the UI. The default settings work well for most ChIP-seq and DNA-SELEX datasets. For RNA data, better results can be achieved by choosing the single stranded mode, reducing the pattern length to 6, and using enrichment and the IUPAC optimization score.

3.2.2. Motif evaluation

Motifs are over-represented oligonucleotide sequence stretches in the input data. Over-representation is a statistical property and is not always a good proxy for biological relevance: motifs with very small enrichments can become significant in sufficiently large data sets, while in biological context such motifs would be binding the transcription factor only marginally better than random stretches of DNA. To give users a meaningful visualization for judging the biological relevance of motifs, we provide p-value distribution and a recall over TP/FP-ratio plots.

For the p-value enrichment plot, p-values are estimated from motif binding scores for each sequence under the null hypothesis that the sequence does not contain the motif. If none of the input sequences would contain the motif, the p-value distribution would be uniform between 0 and 1. Sequences containing the motif will enrich for larger motif binding scores and thus skew the p-value distribution towards lower p-values. We visualize model quality by plotting the recall over the TP/FP enrichment. We use the AvRec score, the area under this curve, for ranking motifs.

Based on two different definitions on bound and unbound sequences, the server provides in total four motif evaluation plots (Kiesel et al. Figure 2C). The two plots corresponding to the dataset-centered definition have the underlying assumption that all uploaded sequences are bound by the factor of interest. The two plots corresponding to the motif-centered definition estimate

the fraction of bound sequences. When nearly all sequences contain the motif of interest, the dataset-centered and motif-centered analyses give similar results. For rarely occurring motifs, the motif-centered AvRec score captures the motif-specific binding and is thus independent of the frequency of the motif in the dataset. With both analyses side by side the user can estimate the presence of each motif in the dataset while also getting a good proxy for the biological binding capabilities for rarely observed motifs.

3.2.3. Motif scanning

Given a set of sequences and a motif model, the task of motif scanning is to estimate the binding strength of the motif at each possible binding position of the sequence. Our server offers a workflow that after uploading the sequences and a model in BaMM or MEME format scans sequences and returns a file with all motif positions with a p-value lower than a user-defined cutoff. Additionally the server provides a plot of the distribution of motif positions relative to the sequences (Kiesel et al. Figure 2C). Especially for sequences extracted symmetrically around peaks of binding signal, as typical for ChIP-seq, a prominent enrichment of center can help distinguishing primary motifs from secondary motifs from co-binding factors.

3.2.4. Motif-motif comparison

Motif-motif comparison is the process of annotating motifs by searching against a database of known motifs. Our server offers users the possibility to upload motifs in BaMM or MEME format and choose one out of nine databases to search against. The databases contain higher-order motif databases learnt from curated ChIP-seq data, such as GTRD (human, mouse, rat, yeast, zebrafish) (Yevshin et al., 2019) or modERN (fly) (Consortium et al., 2012) and manually curated databases such as HOCOMOCO (human, mouse) (Kulakovskiy et al., 2018) and JASPAR (cross-species) (Fornes et al., 2020). By decreasing the e-value cutoff users can limit the search to high-confidence hits. The result of motif-motif comparison is a table of up to five database motifs with the highest e-value. For each database motif the factor name, e-value, logos of query and database hit and a link to further information are listed in a table (Kiesel et al. Figure 2D).

3.2.5. Browser for motif database

Our motif databases are also accessible via the web interface. We offer users the ability to search databases by protein target name or browse all motifs of a specific database (Figure 3.6). For each motif in the database we list the name of the target protein, the zeroth-order sequence logo and its reverse complement, and the species. We also provide cell type and experiment type if the information is available. In case of external motifs (JASPAR: Fornes et al., HOCOMOCO Kulakovskiy et al.) we provide the link to the third party website of the motif. In case of the self-trained BaMM motif databases (GTRD: Yevshin et al. and modERN Consortium et al.), we also provide higher-order sequence logos, motif evaluation plots, motif distribution plots and

Results for: CTCF

BaMM models in this database are automatically generated and sometimes may include motifs from co-binding factors. If you rely on accurate motif annotation, please choose a manually curated motif database such as JASPAR.

2 entries found:


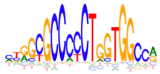
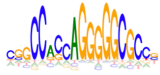
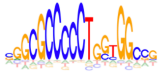
Entry #	Target name	Sequence logo	reverse Comp.	Cell type	Experiment	Species	Details
1	CTCF			HEK293 (embryonic kidney);epidermal keratinocytes;hepatocytes;neutrophils;retinoblastoma xenograft	ChIPseq	Homo sapiens	→
2	CTCFL			K562 (myelogenous leukemia)	ChIPseq	Homo sapiens	→

Figure 3.6.: **The database explorer allows to browse motif databases.** Users can search our databases by factor name and can obtain not only logo motifs, but a link to detailed information of our trained BaMM model, or an external link to the database entry in case of an external database such as JASPAR and HOCOMOCO.

offer users the possibility to download the models or use them as an input for a motif-scanning task.

3.3. Job submission

Depending on the workflow type and the dataset size, our analysis pipelines require a few minutes to an hour of processing time. In order to control resource utilization, the pipelines are submitted as jobs to a task queue. A scheduler monitors and executes jobs in chronological order when enough resources are available. Each job is assigned a randomly generated 128-bit Identifier in form of a universally unique identifier (UUID). Users can use the identifier to obtain information about the progress of their jobs and obtain the results once the job has been executed successfully. The UUIDs allow users to share their results with colleagues via url, without having to worry about third party access. Collisions are practically impossible due to the large range of possible UUIDs ($2^{128} \approx 3.4 \times 10^{38}$). To help users monitor their jobs, we display a table with time of submission, job name, job type, job status and a link to the result page of recently submitted jobs. We track recently submitted jobs by a session id stored in an HTTP cookie. This implementation assigns jobs to the browser session not the individual but cannot assign it perfectly to a person. Our implementation does not require user login and thus makes the server more accessible at the price that our job list cannot track across different browsers and computers and deleting the HTTP cookie will also stop tracking previously submitted jobs.

3.4. Highly configurable, easily deployable open source server

Especially for researchers without a deep computational background, servers can provide an easy access to software without having to worry about installation and computational resources. The source code of servers is however often not open nor is the server software designed for usage apart from the original web instance. The BaMM web server is not only open software, but also easily installable therefore adheres to high standards of open software in science. Instead of using the server instance that we provide at bammmotif.soedinglab.org, users can also set up the server on their own machines with little effort. We achieve this by a strong focus on configurability in the server design and employing of modern container technology: all functionality of our server is provided as interconnected docker containers and are freely available on hub.docker.com. The web server, the task queue and the databases (MySQL and Redis) are running in separate containers. The interplay of the containers is coordinated by *Docker Compose*, making starting a fully functional server as easy as typing `docker-compose up` in a terminal.

Focus on reuse poses high demands on configurability. We allow users to set a wide range of configuration options via key-value pairs in a flat text format. Users can set modalities for database access, the paths for data storage and extensive logging configurations, restrictions for file uploads, adapt the job processing to the available resources and set many more options without having to modify the code of the server. Moreover, we implemented a plug-in system for motif databases that allows dynamically adding and removing motif databases on server startup. This allows users to create their own motif databases and use them for motif-motif comparison on internal or locally deployed instances of the server.

3.5. Designed and setup for low maintenance

For each job BaMM web server stores uploaded input files and all generated files in order to offer users the opportunity to download the complete analysis. In order to prevent excessive resource usage, we give jobs a fixed life-time and implemented a daily cleanup routine that removes expired jobs and thus frees resources. The time for the cleanup routine and the maximum life-time of jobs can be controlled via configuration options. To further reduce the required monitoring, we implemented automatic email notifications when user inputs lead to unexpected crashes and automatically restart jobs in case of a server failure.

3.6. Comprehensive documentation

In order to help users understand the usage, capabilities and output of the server, we provide a comprehensive documentation rendered by *Sphinx* and hosted at bammserver.readthedocs.io. The documentation gives a succinct explanation of the four different job workflows users can submit and their parametrization, an extensive explanation of all visualizations and file formats and covers commonly asked questions.

4. Manuscripts

4.1. BaMM web server

"The BaMM web server for de-novo motif discovery and regulatory sequence analysis"

Anja Kiesel*, **Christian Roth***, Wanwan Ge, Maximilian Wess, Markus Meier
and Johannes Söding[†]

(* equal contribution, (†) corresponding author

Nucleic Acids Research (2018), Vol. 46, Web Server issue W215–W220, doi: 10.1093/nar/gky431.

4.1.1. Author contributions

A.K. designed and developed a first prototype of the BaMM web server. M.W. developed a prototype of the two-step de-novo motif discovery and refinement workflow under supervision of **C.R.**. **C.R.** developed the final submitted server. M.M. and J.S. developed the initial version of the seeding stage software (PEnGmotif), **C.R.** and J.S. further improved PEnGmotif by developing new approaches to local pattern optimization. W.G. developed BaMMmotif2, and generated the motif databases. **C.R.** and W.G. developed tools and scripts for server visualizations. J.S. conceived the project and supervised A.K, M.M, C.R. and W.G. A.K., W.G., **C.R.** and J.S. wrote the manuscript.

4.1.2. Code and data availability

The web server is based on the Django web framework and is written in Python, uses html/css for visualization, mysql and redis as backend databases and docker for easy deployment. It is hosted free of charge at bammotif.soedinglab.org and can be used without prior registration. The web server code is available on github ([soedinglab/BaMM_webserver](https://github.com/soedinglab/BaMM_webserver)) licensed under the AGPL-3.0 license. Both BaMMmotif2 and PEnGmotif are licensed under the GPLv3 license with the source code available without registration at github ([soedinglab/PEnG-motif](https://github.com/soedinglab/PEnG-motif) and [soedinglab/BaMMmotif2](https://github.com/soedinglab/BaMMmotif2)). The BaMM databases used on the BaMM web server are freely available from our data server wwwuser.gwdg.de/~compbiol/bamm.

The BaMM web server for *de-novo* motif discovery and regulatory sequence analysis

Anja Kiesel[†], Christian Roth[†], Wanwan Ge, Maximilian Wess, Markus Meier and Johannes Söding^{*}

Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany

Received February 14, 2018; Revised May 05, 2018; Editorial Decision May 06, 2018; Accepted May 09, 2018

ABSTRACT

The BaMM web server offers four tools: (i) *de-novo* discovery of enriched motifs in a set of nucleotide sequences, (ii) scanning a set of nucleotide sequences with motifs to find motif occurrences, (iii) searching with an input motif for similar motifs in our BaMM database with motifs for >1000 transcription factors, trained from the GTRD ChIP-seq database and (iv) browsing and keyword searching the motif database. In contrast to most other servers, we represent sequence motifs not by position weight matrices (PWMs) but by Bayesian Markov Models (BaMMs) of order 4, which we showed previously to perform substantially better in ROC analyses than PWMs or first order models. To address the inadequacy of P- and E-values as measures of motif quality, we introduce the AvRec score, the average recall over the TP-to-FP ratio between 1 and 100. The BaMM server is freely accessible without registration at <https://bammmotif.mpibpc.mpg.de>.

INTRODUCTION

Many methods such as ChIP-seq or high-throughput SELEX (1) produce a set of nucleotide sequences that are preferentially bound by a protein of interest *in vitro* or *in vivo*. From such data, a motif model for the sequence dependence of the binding affinity of the protein to the DNA or RNA can be derived. This model can then be used to predict binding sites and their strengths in other sequences.

Position weight matrices (PWMs) are the standard model to describe binding motifs. In the PWM every motif position contributes additively and independently from other positions to the total binding energy. Even though the approximation of independence of positions works well for many transcription factors, dependencies do occur (2,3), for example due to bendability or shape constraints during binding (4), to multiple binding configurations of the pro-

tein (5), or to cooperative interactions between closely binding factors that can modulate each others' binding affinities (6).

PWMs can be generalized to Markov models of order k that account for nucleotide dependencies by conditioning the probability for the four nucleotides at each motif position on the previous k nucleotides. First-order Markov models have been added to the popular motif databases JASPAR and HOCOMOCO (7,8). Models of order 2 and higher have not yet been adopted in the major databases, probably due to the difficulties to robustly train the many parameters of these models on limited data.

We recently developed Bayesian Markov Models (BaMMs) (9), which efficiently prevent overfitting by automatically learning conditional probabilities only up to an order k at which they can still be estimated reliably. The key idea is that the conditional probabilities of order $k - 1$ are used as prior probabilities for the conditional probabilities of order k . We have shown that BaMMs of order 4 and 5 systematically outperform PWMs and first-order models in distinguishing bound sequences from negative sequences generated by a second-order Markov model (9).

A very popular web server for regulatory sequence analysis based on PWMs offering a wide choice of tools is the MEME server (10). The RSAT web server (11) provides a general toolbox for the analysis of regulatory sequences including motif-based analyses. Furthermore, other web resources and databases are available for training first-order models (12,13).

The BaMMmotif server brings the improved quality of BaMM motif models within reach of users unfamiliar with command-line tools, in a largely self-explanatory web interface designed for ease of use. The user can discover BaMM models enriched in a set of input sequences, scan sequence sets with BaMM models for motif occurrences, and compare discovered or uploaded motifs with a database of BaMM models learned from ChIP-seq datasets.

^{*}To whom correspondence should be addressed. Tel: +49 551 201 2890; Fax: +49 551 201 2803; Email: soeding@mpibpc.mpg.de

[†]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

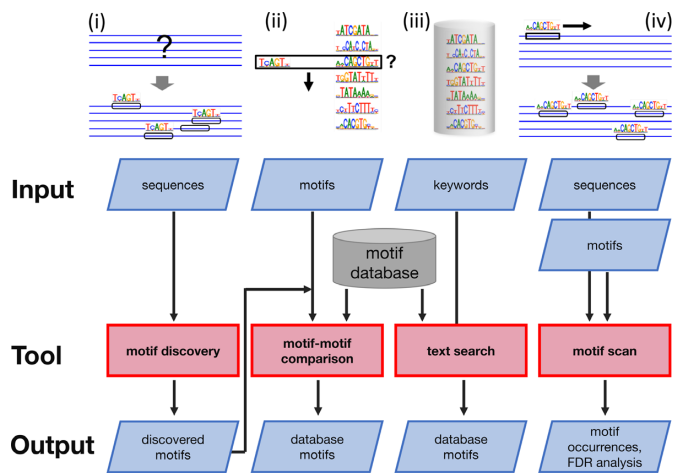


Figure 1. Tools offered by the BaMM server: (i) de-novo discovery of motifs enriched in a nucleotide sequence set. Motifs are represented by higher order BaMMs, which capture correlations between nucleotides. (ii) Searching with an input BaMM or PWM motif for similar motifs in our database of over 1000 fourth-order BaMM motifs. (iii) Browsing and keyword searching in our motif database. (iv) Scanning a set of nucleotide sequences with BaMM or PWM motifs to find motif occurrences.

BAMM TOOLS

In the following we describe the four tools offered by the BaMM server (Figure 1).

De-novo motif discovery using higher-order BaMMs

This tool discovers the motifs enriched in an input set of nucleotide sequences in comparison to the expectation from a background model. For example in sequences obtained from a ChIP-seq or HT-SELEX experiment, the BaMM motif models will approximately describe the sequence dependence of the binding energy of the protein to DNA (see page 2 of supplementary material in (9)). The motif model can be used to scan other sequences for motif occurrences (see next subsection).

Method. The motif discovery proceeds in two stages, seed pattern discovery and motif refinement. For the pattern discovery we developed a fast and sensitive algorithm (PENG-motif) that will be described in detail elsewhere. Briefly, it finds all locally optimal W -mers (default $W = 8$) over an alphabet of 11 IUPAC letters (A, C, G, T, R = A or G, Y = C or T, W = A or T, S = C or G, M = A or C, K = G or T, N = A, C, G or T), where locally optimal patterns are those for which changing any single one of its letters would result in a decreased enrichment relative to the random expectation from the background model. (Alternatively, the P -value or the mutual information between presence/absence of motifs and input versus background sequence can be optimized.) With each locally optimal pattern, a PWM of length W is initialized and optimized using an expectation maximization (EM) algorithm. PWMs that have very similar overlapping regions are merged and ranked by our new AvRec score (next section).

The seed motifs are then refined using BaMM!motif (9). It learns the parameters of the BaMMs with an

EM algorithm that maximizes the log likelihood of the motif model under a zero-or-one-occurrence-per-sequence (ZOOPS) model (14). The BaMM server offers to train motifs of up to fourth order.

By default, BaMM learns a second order Markov model from the input sequences as a background model. The background model is needed first in the motif discovery to model the sequence stretches not modeled by the motif model and second in the motif quality assessment step to generate negative sequences to estimate motif occurrence P -values. A second order model is generally preferable to first or zeroth order as it can better describe sequence biases observed in open versus closed chromatin, ChIPped versus unChIPped sequences etc. (15). A model of order 1 or 0 is recommended for the discovery of very short motifs (e.g. four to five nucleotides) such as to RNA-binding sites, as such short motifs could be learned to some extent even by a second order background model, severely reducing the sensitivity to discover them.

Usage of de-novo motif discovery. After uploading a FASTA file of up to 50 MB with the input sequences, the motif discovery can be started. A drop-down menu offers advanced options in four categories: general settings, seeding stage, model refinement stage and settings for plots and analyses.

In the general settings category the user can choose whether the motif can be present on both strands, set the order of the background model (default 2) and upload an optional sequence set to train the background model on. Settings of the seeding stage include the initial pattern length W , the z -score significance threshold for refining a motif, and the objective function to optimize in the search for locally optimal patterns. For the refinement stage the user can choose the motif model order (default 2) and the number of flanking positions on the left and right of the core model found in the seed stage. Finally, the user can choose to skip motif scanning, motif performance evaluation or motif annotation, and change the significance thresholds for scanning and annotation.

By default up to four best-performing seed patterns are refined to higher-order models. Seed patterns are ranked by their average recall (AvRec) score (see below). Alternatively, the user can choose to select seed patterns manually for refinement after the seeding stage.

The results page (Figure 2A) lists in a summary table the discovered enriched motifs with their IUPAC patterns, the sequence logos of the 0th-order model (forward and reverse complement), the AvRec motif quality score and the fraction of sequences with motifs ('frac. occurrence'), estimated using the fdrtool (16) (explained in subsection 'Dataset AvRec and motif AvRec'). By clicking on the motifs or scrolling down, detailed results for the motifs are shown: 0th-order (forward and reverse complement), first- and second-order sequence logos (Figure 2B); four motif quality assessment plots and a plot of the positional distribution of the motif occurrences relative to the center of the sequences (Figure 2C). (Sequences do not have to be of the same length.) Clicking on the download button in the summary table above saves a zip file containing motif files in BaMM format with the extension ihbcp and all analysis

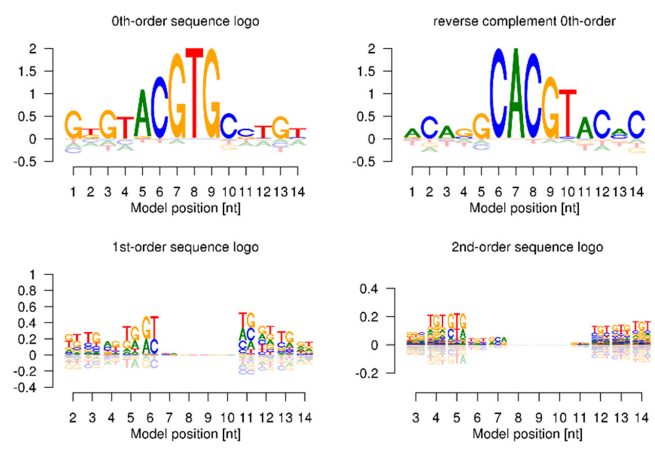
A Refined Motifs

DOWNLOAD ALL

#	IUPAC	PWM	reverse Comp.	AvRec	frac. occurrence	Download
1	GTACGTGCCY			0.554	0.494	
2	GGGCGGGG			0.855	0.159	
3	RCACGTMCA			0.862	0.111	

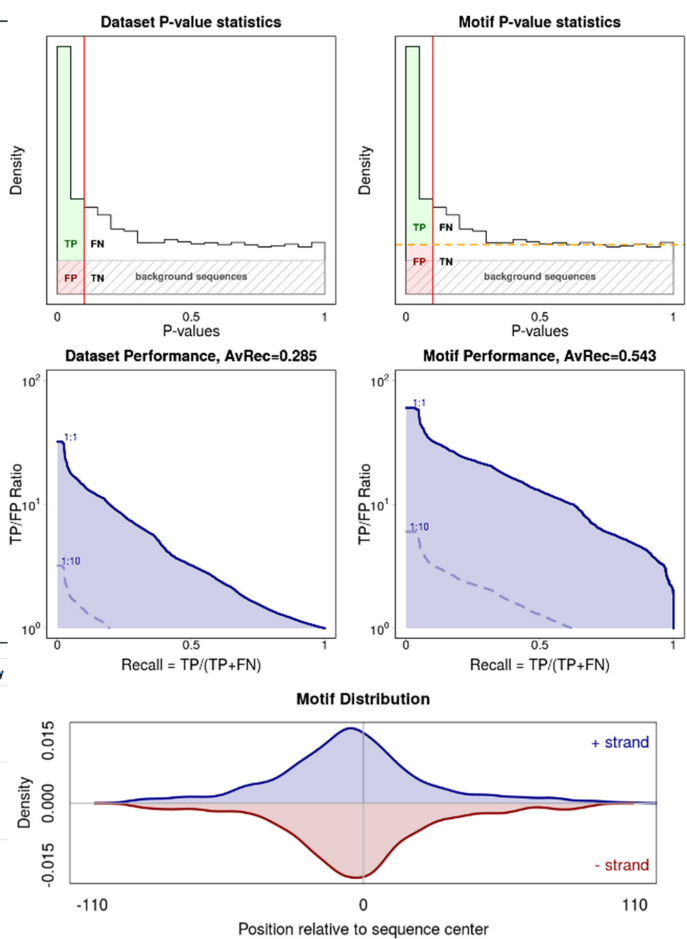
B

Motif # 1 DOWNLOAD MODEL



C

Motif Performance and Motif Distribution on Sequences



D

Best matches with our motif database

name	e-value	query motif	database PWM	reverse Comp.	DB Entry
HIF-1-alpha	5.6E-05				
HIF-3-alpha	1.2E-04				
DEC1	6.4E-02				

Figure 2. Selected results from a de-novo motif discovery run. (A) Summary table of discovered motifs. (B) Sequence logos of order 0, 1 and 2 for one discovered motif. (C) Motif quality analysis and positional distribution. In the dataset-centered analysis (left) all input sequences are defined as positives. In the motif-centered analysis (right), only input sequences carrying a motif occurrence are positives. Their fraction is estimated using fdrtool (orange broken line on the upper right). The quality of motifs is quantified by average recall (AvRec), the blue area under the TP-to-FP-versus-recall curves. The curves for positive-to-negative ratios in the dataset of 1:1, 1:10 and 1:100 are plotted. Recall = TP/(TP + FN), where TP = true positives, FP = false positives, FN = false negatives. Positional distribution of the motif occurrences relative to the center of the sequences is shown on the bottom. (D) List of database motifs similar to discovered motif.

plots for the motif. Last, the database motifs found similar to the discovered motif are listed (see ‘motif-motif comparison’ below) with links to the database entry (‘Best matches with our motif database’, Figure 2D). The results page can later be retrieved by giving the job ID on the ‘Find my job’ page. Results are stored for up to 3 months.

SCAN SEQUENCES FOR MOTIF OCCURRENCES

A set of input sequences can be scanned with a motif or a set of motifs for motif occurrences. The input motifs can be in MEME (version 4 and above) or BaMM format and could have been discovered de-novo by BaMM or they could come from the BaMM database or some other database.

We developed a motif scanning tool that evaluates the log odds score for BaMMs (and PWMs) of any order. A table with the motif occurrences can be downloaded in a zip file, together with the motif analysis on the supplied sequences. The table of motif occurrences contains in each line the sequence length, motif position, binding sites, P -value, and E -value of the occurrence. The P -values are computed by maximum-likelihood fitting of the high-scoring tail of the log-odds score distribution on sequences generated with the background model with an exponential function, which gave good fits (see PhD thesis at <https://edoc.uni-muenchen.de/21504/>). Each motif is also evaluated using the dataset and motif-based average recall (AvRec, see below) and the positional distribution of the motif occurrences around the center of the sequences (Figure 2C).

BAMM MOTIF DATABASE

Our database contains 1021 fourth-order BaMMs trained on ChIP-seq datasets of 620 human transcription factors (TFs), 345 mouse TFs, 19 rat TFs, 16 zebrafish TFs and 21 yeast TFs from the GTRD database (17). For each motif, a meta table, details with higher-order sequence logos, positional enrichment around the centers of training sequences, and motif quality assessment plots, evaluated on the ChIP-seq training sequences, are presented. The user can browse the database or perform a text search through the list of names of the transcription factor.

SEARCH WITH QUERY MOTIFS THROUGH THE MOTIF DATABASE

This tool searches for motifs in our BaMM motif database that are similar to the query motifs (in MEME or BaMM format). This motif-motif search is automatically run after de-novo motif discovery using each of discovered motifs as query. The query motifs can also be provided by the user. The output of this tool is shown in Figure 2D.

Motif-motif similarities are computed between the zeroth order contribution of the motifs. The distance between two motifs is the minimum distance for any gapless alignment of their columns that leaves at least four columns aligned. The similarity between aligned motifs M_1 and M_2 is defined as

$$\sum_j (-d^{\text{JS}}(M_{1j}, M_{2j}) + d^{\text{JS}}(M_{1j}, M_{\text{bg}}) + d^{\text{JS}}(M_{2j}, M_{\text{bg}})).$$

Here, the sum runs over all aligned columns j . $d^{\text{JS}}(M_{1j}, M_{2j})$ is the Jentsen-Shannon divergence between the four nucleotide probabilities of model 1 and of model 2 at aligned column j , and M_{bg} is the zeroth order background distribution in the set on which the query model was learned.

The E -values for the motif-motif matches are computed from these similarity scores by fitting the density of scores computed between 100 randomized query motifs and the databases motifs and fitting the high-scoring tail with an exponential distribution (see PhD thesis of Anja Kiesel at <https://edoc.uni-muenchen.de/21504/>). The randomization of the query motif is achieved by exchanging A with T probabilities of each position with probability 0.5, and analogously for C and G. In addition columns within 2 positions of each other were randomly swapped. This motif randomization keeps the local GC vs. AT content conserved. In our benchmarks, this score performed as well as the best of the TOMTOM scores (Pearson correlation) (18). An example of results of the motif search is shown in Figure 2D.

MOTIF QUALITY ASSESSMENT AND RANKING

P -values do not assess biological relevance of motifs

P -values and E -values have a severe drawback for ranking motif models: They can be very significant and yet the motifs have no biological relevance at all. For a fixed x -fold enrichment of motif occurrences on the input set in comparison to the background model, the P -value decreases exponentially with the number of sequences in the zero-or-one-occurrence-per-sequence (ZOOPS) model. For that reason, even biologically irrelevant motifs with very slight enrichment factors (e.g. 1.1) can obtain an extremely significant E -value if the input set is large enough. Small enrichment factors can occur frequently in practice simply due to an imperfect background model that slightly underestimates the expected frequency of occurrence.

Precision, recall and false discovery rate

To get a more relevant measure of how well the motif model can separate sequences with a motif (positives) from the background sequences (negatives), we first generate for each input sequence one random sequence of the same length sampled with the second-order Markov background model learned from the input sequences. The score for an input or background sequence is the maximum of the log odds scores of the BaMM over all possible motif positions (ZOOPS model). Every sequence with a score above a cut-off is predicted to carry a motif. We rank all sequences by their score and, for each cut-off score, we count the number of correct predictions above that score, called true positives (TP), and the number of incorrect predictions above the cut-off score, called false positives (FP). The precision is the fraction of predictions that are correct, $\text{TP}/(\text{TP} + \text{FP})$, and the recall (=sensitivity) is the fraction of positive sequences that are actually predicted, $\text{TP}/(\text{TP} + \text{FN})$. The false discovery rate is $\text{FDR} = 1 - \text{precision} = \text{FP}/(\text{TP} + \text{FP})$.

If we did this analysis on the same sequences from which we had trained the model, we could easily overestimate the motif model performance by overtraining. We therefore use

four-fold cross-validation to assess the motif model performance: We split the input and background sequences into four equal-sized parts, retrain the model on three. The results from the four hold-out sets are then combined.

The AUPRC assesses models partly in irrelevant regimes

The area under the recall-precision curve (AUPRC) (see Supplementary Figure S2B) can be interpreted as mean model recall (=sensitivity) averaged over the entire range of precision from 0 to 1. Consider two models: one achieves a maximum precision of 0.99 and the other achieves at any recall a 1% higher precision, with a maximum at 0.9999. Even though the two models have AUPRCs that only differ by 1%, their minimum false discovery rates differ by two orders of magnitude (0.01 and 0.0001), which can make a huge difference in practice.

Consider two application cases. In the first, the expected ratio of sequences with and without true binding sites is $\sim 1:1$, e.g. for a ChIP-seq experiment, and in the second case it is $1:100$, e.g. when scanning 10^4 promoter regions in the human genome for motif occurrences, of which 100 are expected to carry the motif. In the first case, an FDR of 0.1, determined at ratio 1:1 between positive and negative (background) sequences, is quite satisfactory to identify sequences with true binding sites. In the second case, an FDR of 0.1 would result in $0.1 \times 10^4 = 1000$ false predictions, which would swamp the expected 100 true binding occurrences. A model with an FDR of 0.001 determined at ratio 1:1 between positive and negative sequences would give us $0.001 \times 10^4 = 10$ false predictions, which would result in an acceptable FDR of $10/110$.

So the FDR (estimated for a ratio 1:1 of positives to negatives) that is relevant to assess the quality of motif models depends on the application, more precisely, on the expected ratio of positives to negatives in the sequence data. In contrast, the AUPRC puts much weight on very high FDRs, e.g. the range between 0.9 and 1 has as much weight as the range between 0 and 0.1. Another popular measure, the area under the receiver operator curve (AUROC), can be shown to be even less relevant and difficult to interpret for motif model assessment.

Average recall (AvRec)

We sought a motif quality analysis plot and associated quality measure (i) that covers the range of FDRs most relevant in practical applications and (ii) that allows the user to easily estimate the performance of the motif in her particular application, that is, given the ratio between positive and negative sequences expected for her application.

We replace the precision in the precision-recall plot by \log_{10} of the ratio $R = TP/FP$ between true and false positives, $\log_{10} TP/FP$ (Figure 2C, middle). From the ratio R one can immediately obtain the false discovery rate, $FDR = 1/(1 + R)$, and vice versa, $R = (1 - FDR)/FDR$. $R = 100$ corresponds to $FDR = 1/101$, $R = 1$ corresponds to $FDR = 0.5$. We define the AvRec quality measure as the average recall computed over a range of $\log_{10} R$ -values from 0 to 2, which corresponds to an FDR-range from $1/101$ to 0.5. We argue

that this range of FDRs is most relevant in practice, as illustrated by the two previous examples.

The new quality measure also satisfies the second requirement. The user can simply pick the curve in the AvRec plot that corresponds to the ratio of positive to negative sequences that she expects in her application. Nicely, the curve at ratio 1:10 is the curve at ratio 1:1 shifted down by one unit ($\log_{10} 10$), because R is proportional to the ratio of positive to negative sequences in the dataset: When the number of negative sequences is amplified by 10, the number of false positive predictions will also be increased by a factor of 10. On the web server, we show the curves with ratios of 1:1, 1:10 and 1:100 (if visible on the y -scale).

Dataset AvRec and motif AvRec

We used two definitions of positive and negative sequences. In the *dataset-centered analysis* (Figure 2C, left), the true positive sequences are all sequences from the input set above the cut-off score and the false positive sequences are all background sequences above the cut-off score. The upper left plot in Figure 2C shows the distribution of the motif occurrence P -values computed from their scores. The curve below shows the $\log_{10} TP/FP$ values over the recall for this definition of true and false positives.

In the *motif-centered analysis* (Figure 2C, right), we consider only those sequences as true positives that actually contain a motif instance. In order to estimate the number of TPs for a given score cut-off, we first estimate the fraction of input sequences that contain motif instances using the *fdrtool* (16). This tool assumes that the negative sequences in the positive set are uniformly distributed over all P -values between 0 and 1 and fits a horizontal line giving the fraction of negatives in the input set to the distribution (orange broken line in Figure 2C, top right). The definition of TPs and FPs illustrated in the top right graph of Figure 2C results in the motif-based AvRec analysis plot below.

When the fraction of motifs in the input sequences is near 100%, both approaches yield very similar results. But when this fraction is small, the motif model may still be very accurate. The motif-centered analysis takes account of that, while the dataset-centered analysis severely underestimates the model performance in these cases.

DOCUMENTATION, USABILITY AND SPEED

Each input parameter is briefly explained in a mouse-over text. A detailed documentation is accessible via the 'Documentation' tab on the top of each page. A motif discovery run with 10k (100k) sequences of length 200nt takes around 3.0 (12.5) min. Scanning 100k sequences of length 200nt on both strands for motif matches takes about 6 min per three motifs. A motif-motif search through the largest subcollection of motifs in our database (620 models) takes around 3.5 min per three motifs.

IMPLEMENTATION

The BaMM web server is built on the Django Web framework using Nginx as reverse proxy. Jobs are scheduled via Celery's asynchronous task queuing system, with the help of

Redis as a message broker, and executed on a Linux computer with 28 physical cores using 4 cores per job. MySQL is used as back end database to store results and job parameters. The web front end, back end and the database run in separate Docker containers, enabling easy deployment (Supplementary Figure S1).

CONCLUSION

We hope the BaMM web server will enable many users to exploit the greater descriptive power of BaMMs for motif discovery and regulatory sequence analysis. In the future we will work on extending the database of motifs, especially by training on HT-SELEX datasets.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We thank our beta users for testing and feedback and Fëdor Kolpakov of BioUML (<http://gtrd.biouml.org>) for support with their GTRD database.

FUNDING

German Federal Ministry of Education and Research (BMBF) within the frameworks of e:Bio [SysCore, project 0316176A]; SPP 1935 (project CR 227/6-1) of the German Research Foundation (DFG); International Max Planck Research School for Genome Science (IMPRS-GS). Funding for open access charge: Institutional.

Conflict of interest statement. None declared.

REFERENCES

- Lambert,S.A., Jolma,A., Campitelli,L.F., Das,P.K., Yin,Y., Albu,M., Chen,X., Taipale,J., Hughes,T.R. and Weirauch,M.T. (2018) The human transcription factors. *Cell*, **172**, 650–665.
- Bulyk,M.L., Johnson,P.L. and Church,G.M. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.*, **30**, 1255–1261.
- Jolma,A. and Taipale,J. (2011) Methods for analysis of transcription factor DNA-binding specificity in vitro. In *A Handbook of Transcription Factors*, Springer pp. 155–173.
- Rohs,R., West,S.M., Sosinsky,A., Liu,P., Mann,R.S. and Honig,B. (2009) The role of DNA shape in protein–DNA recognition. *Nature*, **461**, 1248
- Badis,G., Berger,M.F., Philippakis,A.A., Talukder,S., Gehrke,A.R., Jaeger,S.A., Chan,E.T., Metzler,G., Vedenko,A., Chen,X. *et al.* (2009) Diversity and complexity in DNA recognition by transcription factors. *Science*, **324**, 1720–1723.
- Jolma,A., Yin,Y., Nitta,K.R., Dave,K., Popov,A., Taipale,M., Enge,M., Kivioja,T., Morgunova,E. and Taipale,J. (2015) DNA-dependent formation of transcription factor pairs alters their binding specificity. *Nature*, **527**, 384.
- Khan,A., Fornes,O., Stigliani,A., Gheorghe,M., Castro-Mondragon,J.A., van der Lee,R., Bessy,A., Chèneby,J., Kulkarni,S.R., Tan,G. *et al.* (2017) JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Res.*, **46**, D260–D266.
- Kulakovskiy,I.V., Vorontsov,I.E., Yevshin,I.S., Sharipov,R.N., Fedorova,A.D., Rumynskiy,E.I., Medvedeva,Y.A., Magana-Mora,A., Bajic,V.B., Papatsenko,D.A. *et al.* (2018) HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Res.*, **46**, D252–D259.
- Siebert,M. and Söding,J. (2016) Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences. *Nucleic Acids Res.*, **44**, 6055–6069.
- Bailey,T.L., Boden,M., Buske,F.A., Frith,M., Grant,C.E., Clementi,L., Ren,J., Li,W.W. and Noble,W.S. (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res.*, **37**, W202–W208.
- Medina-Rivera,A., Defrance,M., Sand,O., Herrmann,C., Castro-Mondragon,J.A., Delerce,J., Jaeger,S., Blanchet,C., Vincens,P., Caron,C. *et al.* (2015) RSAT 2015: regulatory sequence analysis tools. *Nucleic Acids Res.*, **43**, W50–W56.
- Mathelier,A. and Wasserman,W.W. (2013) The next generation of transcription factor binding site prediction. *PLOS Comput. Biol.*, **9**, e1003214
- Kulakovskiy,I., Levitsky,V., Oshchepkov,D., Bryzgalov,L., Vorontsov,I. and Makeev,V. (2013) From binding motifs in ChIP-Seq data to improved models of transcription factor binding sites. *J. Bioinf. Comput. Biol.*, **11**, 1340004.
- Bailey,T.L. and Elkan,C. *et al.* (1994) Fitting a mixture model by expectation maximization to discover motifs in bipolymers. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **2**, 28–36.
- Hartmann,H., Guthöhrlein,E.W., Siebert,M., Luehr,S. and Söding,J. (2013) P-value-based regulatory motif discovery using positional weight matrices. *Genome Res.*, **23**, 181–194.
- Strimmer,K. (2008) fdrtool: a versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics*, **24**, 1461–1462.
- Yevshin,I., Sharipov,R., Valeev,T., Kel,A. and Kolpakov,F. (2017) GTRD: a database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Res.*, **45**, D61–D67.
- Gupta,S., Stamatoiyannopoulos,J.A., Bailey,T.L. and Noble,W.S. (2007) Quantifying similarity between motifs. *Genome Biol.*, **8**, R24

4.2. BaMMmotif2

“Bayesian Markov models improve the prediction of binding motifs beyond first order”

Wanwan Ge, Markus Meier, **Christian Roth**, and Johannes Söding[†]

([†]) corresponding author

NAR Genomics and Bioinformatics, 2021, Vol. 3, No. 2 1, doi: 10.1093/nargab/lqab026.

4.2.1. Publication abstract

Transcription factors (TFs) regulate gene expression by binding to specific DNA motifs. Accurate models for predicting binding affinities are crucial for quantitatively understanding transcriptional regulation. Motifs are commonly described by position weight matrices, which assume that each position contributes independently to the binding energy. Models that can learn dependencies between positions, for instance, induced by DNA structure preferences, have yielded markedly improved predictions for most TFs on in-vivo data. However, they are more prone to overfit the data and to learn patterns merely correlated with rather than directly involved in TF binding. We present an improved, faster version of our Bayesian Markov model software, BaMMmotif2. We tested it with state-of-the-art motif discovery tools on a large collection of ChIP-seq and HT-SELEX datasets. BaMMmotif2 models of fifth-order achieved a median false-discovery-rate-averaged recall 13.6% and 12.2% higher than the next best tool on 427 ChIP-seq datasets and 164 HT-SELEX datasets, respectively, while being 8 to 1000 times faster. BaMMmotif2 models showed no signs of overtraining in cross-cell line and cross-platform tests, with similar improvements on the next-best tool. These results demonstrate that dependencies beyond first order clearly improve binding models for most TFs.

4.2.2. Author contributions

W.G. developed the BaMMmotif2 software. M.M. and J.S. developed the initial version of the seeding stage software (PEnG), **C.R.** and J.S. further improved the software by developing new approaches to local pattern optimization. W.G. implemented the statistical approach and conducted all the benchmarks. W.G. and J.S. wrote the manuscript. J.S. supervised the research.

4.2.3. Code and data availability

Both BaMMmotif2 and PEnGmotif are implemented in C++ and Python. The code is licensed under GPLv3 and freely accessible without registration at github: [soedinglab/PEnG-motif](https://github.com/soedinglab/PEnG-motif), and [soedinglab/BaMMmotif2](https://github.com/soedinglab/BaMMmotif2), and supports Linux and macOS. Both BaMMmotif and PEnGmotif are also integrated in our BaMM web server. The analysis scripts are available in Jupyter Notebook format on github: [soedinglab/bamm-benchmark](https://github.com/soedinglab/bamm-benchmark).

5. Discussion

In this work we developed BaMM web server, a platform for learning higher-order TF binding motifs from large amounts of next-generation sequencing data and analysing their potential biological impact. To this end we developed four workflows. (i) Our de-novo motif discovery workflow is designed as a two stage process: a fast seeding stage presents users a list of enriched PWM models which can be selected for further higher-order refinement with our BaMMmotif software. (ii) We developed the AvRec score as a means to quantify the biological relevance of motifs. In contrast to the traditional evaluation metrics such as statistical significance, area under the receiver operating characteristic curve (AUROC) and area under the precision recall curve (AUPRC), the AvRec score puts higher emphasis on biologically relevant regimes of binding affinities. Our motif AvRec score quantifies motif quality independent of the abundance in the dataset, a property that can help identifying biologically meaningful motifs that are present only in a small subset of input sequences, such as motifs arising from TF co-binding. (iii) Our motif scanning identifies positions in the input sequences that are preferentially bound by the motif and quantifies the binding affinity. (iv) motif-motif comparison with databases of known TF motifs allows annotating motifs from co-binding factors. (v) Large databases of higher-order models learnt on public datasets allow browsing known motifs and scanning sequences with our collection of higher-order models. We hope that BaMM web server provides a toolbox that makes analyzing TF binding data with higher-order models more attractive and contributes to deriving better biological insight, especially for factors with binding affinities that are not well modelled by the additivity assumption of individual motif positions.

In order to achieve the wide range of functionalities of our web server, several standalone tools had to be developed or extended and improved, such as the fast seed search PEnGmotif, the higher-order refinement suite BaMMmotif, and various scripts for evaluation and fast motif-motif comparison. PEnGmotif is especially noteworthy, because by using a k-mer approach instead of the classical EM framework, it requires the input sequences only for the very fast k-mer counting step and thereby allows generating non-redundant, enriched PWM motifs at high speed that is in practise nearly independent of the data set size. This makes PEnGmotif especially suitable for a quick pre-screening for motif seeds that can serve as initializations to the higher-order refinement procedure with BaMMmotif.

5.1. Challenges when training complex models.

Due to the advancements in next-generation sequencing, the amount of available data is growing, making it possible to train more complex models. The high explanatory power of more complex models comes however with the risk of learning *too much*. I will discuss the arising problems in the context of TF binding models in the next paragraphs.

Statistical overfitting. Statistical overfitting is a common problem in the machine learning field that arises out of the bias-variance tradeoff. In the face of limited data, complex models have a high variance

in their high number of learnt parameters. The variance originates from describing the training data so well that even random noise is captured by the model parameters, thereby severely limiting the ability of the models to generalize. A model with fewer parameters cannot capture the noise and will have less variance in the parameter estimations, albeit at the risk of not having enough complexity to describe the underlying signal, thereby introducing bias. The success of PWMs as predominant motif models is due to their low variance (only $3 \times L$ free parameters). Since binding affinity of most TFs adheres to the additivity approximation, PWMs also have low bias, explaining their success as motif models. Higher-order Markov models are highly parameterized models and thus also risk overfitting to the training data. By construction, BaMMs are self-regularizing models: by adding pseudocounts based on the lower orders, in absence of sufficient evidence in form of k-mer counts, BaMMs fall back to lower orders, thereby reducing the model complexity. This design makes BaMMs resistant to statistical overfitting, allowing to train even 5th-order models on small data sets.

Learning experimental biases. Experimental biases are a hallmark of next-generation sequencing data. Sequence biases are inherent in essential steps such as library amplification and adapter ligation, making them an unavoidable challenge in the analysis of biological data (Aird et al., 2011; Diaz et al., 2012). By simultaneously modelling bound and unbound sequences, statistical motif models can correct out experimental biases that manifest themselves in skewed k-mer composition biases. Ideally, this bias correction however requires a set of unbound sequences exhibiting the same experimental biases as the sequences of interest, which is often difficult to obtain. In absence of background binding data, background binding preferences can be approximated on the sequencing data, by assuming that all but a tiny fraction of the sequenced DNA fragments bind the protein of interest. By default we train BaMM models using a 2nd-order homogeneous Markov model as background model, thereby correcting experimental biases that can be described as a skew the trimer composition. As background binding experiments are rare, it is important to bear in mind that PEnGmotif and BaMMmotif typically perform an approximative bias correction.

Learning motif mixtures. When training statistical motif models with the EM framework, a typical assumption is that sequences contain either zero or one motifs. All motif positions are described by the motif model, all other positions are described by the background binding model. Especially for in-vivo data, this assumption is violated due to TF cooperation and thus local clustering of TF binding sites: ChIP-seq enriched fragments of 100 to 250 bp length typically contain arrays of motifs. Due to their simplicity, PWM models cannot describe motif mixtures. Depending on the data and the initial seeding, the EM framework will converge to a strongly enriched motif, often the motif of the TF of interest. Highly parameterized models such as BaMMs, or DNNs are not constrained to learning only one motif, but can learn motif mixtures. For DNNs this capability is explicit by the user-defined number of convolution kernels (Alipanahi et al., 2015; Zeng et al., 2016). Explicit motif models that take interdependencies into account, such as BaMMs can also learn motif mixtures with co-occurring or secondary motifs, complicating the interpretation of the motif scores as TF binding affinities (Keilwagen and Grau, 2015; Eggeling et al., 2015; Eggeling, 2018).

When evaluating the performance of de-novo motif discovery tools, all these biases have to be taken into account. Cross-validation benchmarks by iterative train-test splits only account for biases arising from statistical overfitting. Cross-platform benchmarks that train on in-vitro data and test on in-vivo data can additionally take protocol-specific biases and undesirable predictive advantages due to learning motif mixtures of correlated motifs into account. Our group has shown that BaMMs outperform competitors when trained on in-vitro and evaluated on in-vivo data and vice versa, indicating that higher-order order

models learn biological signal beyond first-order dependencies (Ge et al., 2021).

Transcription factor binding is ultimately studied as a means to gain a deeper understanding in transcription regulation. The progress towards this goal can be measured by how well we can predict context-specific gene expression from the sequence alone.

5.2. Limitations

Obliviousness of genomic context. The short motif length compared to the 3 billion bp in the human genome gives rise to millions of potential binding sites with high binding affinity, of which only a small fraction are bound in individual cells and cell types (Wasserman and Sandelin, 2004). Understanding the regulatory code underlying gene expression therefore requires understanding the interplay between a wide variety of regulatory signals. By taking the interdependencies of neighboring positions in transcription factor binding motifs into account, higher-order Markov models describe the sequence-dependent binding affinity of individual transcription factors more accurately, albeit in absence of the biological context. This higher fidelity in modelling binding affinity can help finding weaker binding sites, an important mechanism of in transcription regulation (Crocker et al., 2016; Farley et al., 2016; Kribelbauer et al., 2019). However, due to their inherent ignorance of biological context, providing context falls ultimately to the user. A potential weak affinity binding site identified in a transcriptionally active enhancer in the cell type of interest is much more likely to be of biological relevance than sites with similar affinity in repressed regions of the genome.

Reliance on base-readout in consecutive short nucleotide stretches. The underlying assumption of motif discovery with simple PWM models is that the studied TF relies on high-affinity base-readout to find its target locations in the genome. Higher-order models increase the predictive power by their ability to capture TF shape readout, short variable spacers and variable dimerization partners (Siebert and Söding, 2016). With their aim of modelling the binding affinity of single transcription factors out of context, these models intentionally ignore some forms of cooperativity, such as co-recruitment. The boundary between modelled and unmodelled cooperativity is however fluid: Longer motifs arising from oligomerization are typically captured as single binding single motif. While complex higher-order models such as BaMMs can capture variable oligomerization partners to some extent, BaMMs do not model these cooperative interactions in a principled way and their capabilities in capturing these binding modes has not yet been studied in detail. Multiple DNA binding domains can lead to cooperative binding within the same TF that can result in multiple distinct motifs for a single transcription factor, depending on the combination of bound domains (Siggers and Gordan, 2014). Zinc-finger proteins are especially complicated due to their high number of DNA binding domains, compared to their observed motifs (Najafabadi et al., 2015). This hints towards the existence of complex cooperative binding motifs that are not well captured by current motif models.

Limited biological interpretability. By capturing complicated inter-motif correlation signatures, higher-order BaMMs learn predictive information beyond first order in a cross-platform benchmark (Ge et al., 2021). In the meantime, it has been indicated that DNA shape can be well described by first-order models (Rube et al., 2018), raising the question of what biological signatures BaMMs actually learn. While higher-order motif logos can visualize the information a specific order adds on top of the previous orders, it is often difficult to assign meaning to the informative k-mers. Ultimately, it would be desirable to be able to quantify the amount of information provided by variable spacers, learning alternate

binding motifs, and maybe other, still unknown biological signatures. This would not only advance our understanding of the binding behaviour of the studied TF, but also inspire the development of a new generation of models that could improve by modeling this information explicitly.

5.3. Outlook

End-to-end learning as a paradigm shift. It is my understanding that until very recently, a widely held belief in biology was that modelling the whole, requires a deep understanding of its parts. Recent breakthroughs in AI challenge this philosophy. Flexible, highly parameterized end-to-end differentiable DNNs have been employed to solve challenging problems in life sciences without relying on previously derived models and knowledge as feature input. In order to achieve high performance at their trained task, well-crafted DNNs tend to rediscover biological principles, making it the researchers task to extract this implicit knowledge from the parameters of the network.

In the field of transcription regulation it has for example been shown that predicting experimental transcription factor binding profiles can elucidate the joint binding syntax of the pluripotency TFs (Avsec et al., 2021b) and that predicting gene expression can reveal gene-regulatory features such as long-range enhancer-promoter interactions (Avsec et al., 2021a; Karbalayghareh et al., 2021).

Artificial intelligence and the big picture. For now DNNs are probably our best bet for gaining a holistic understanding of transcript regulation. The model that ultimately solves this defining challenge will have to be able to predict gene expression up to measurement errors from sequence alone in any cell type, possibly even across organisms. Recently the first papers have been published that among other features predict gene expression from sequence in a multitask learning objective (Avsec et al., 2021a; Karbalayghareh et al., 2021). Until now, these proposed methods predict gene expression only for specific cell types. The transformer-based architecture with a visual field of 100 kb is moreover limited to close and mid-range enhancer-promoter pairs (Avsec et al., 2021a), whereas the graph-convolutional network approach can detect enhancers in 2 mb distance, and is thereby able to integrate virtually all known cis-regulatory promoter-enhancer interactions (Karbalayghareh et al., 2021), while improving over state-of-the-art gene prediction methods. The currently proposed architectures only integrate cis-regulatory effects, ignoring trans-effects and therefore will likely not be yet be able to solve the challenge altogether. Nevertheless they will undoubtedly pave the way to a new era for big data in computational biology. As these DNNs are end-to-end differentiable, the role of current transcription factor binding models towards this ultimate goal is unclear.

Should we ever be able to obtain a holistic AI model of transcription regulation, human curiosity will no doubt demand us to attempt to make sense of it by extracting biological principles that can be described by human comprehensible biophysical models. Whether this is strictly necessary in order to truly be able to declare the gene regulation problem solved will ultimately remain a philosophical question.

Part II.

MRF coupling parameter correction

6. Introduction

Arguably the most impressive feature of life as we know it is that even the most complex organisms arise by subsequent divisions starting from one single cell. The necessary genetic information is stored in long sequences of the four nucleobases Adenine, Cytosine, Guanine and Thymine in double-stranded DNA molecules, which are redundantly copied in every single cell of the organism. The process of materializing this information in the cell has been formulated by the *central dogma of molecular biology*: DNA is transcribed into mRNA which in turn is translated into proteins. Just like DNA, RNA and proteins are macromolecules consisting of a small set of repeated building blocks. Due to the differences in the properties of their building blocks, DNA, RNA and proteins have very different properties and fulfil distinct, highly specialized purposes in the cells. DNA is a macromolecule specialized in storing, correcting and copying genetic information and thus contains the blueprints of all cellular structures and machines together with the logic of their regulation in space and time. Proteins spontaneously fold into complex three-dimensional structures and make up functional elements in the cell in the form of molecular machines, structural elements, and regulatory and messaging agents among others. The central dogma of molecular biology highlights the role of mRNA in information transmission from DNA to proteins. Due to its high versatility, information transmission is only one of many roles of RNA and other classes of RNA are involved in a high number of cellular processes. Its similarity with DNA allows RNA to act as information storage and regulation molecules while their ability to fold into complex three-dimensional structures allows them to fulfill roles in central processes such as translation and splicing.

Proteins are the protagonists of this chapter, especially their individual three-dimensional structure, which determines their cellular function. In contrast to DNA and RNA, polymers of 4 different building blocks, proteins are assembled from 20 different amino acids. At a central carbon atom C_α all amino acids share a hydrogen atom, and an amino and carboxyl group, but differ in a variable side chain. In contrast to nucleobases which mainly differ in their hydrogen-bond base-pairing behavior, the side chains of amino acids are diverse in their chemical properties such as charge, polarity and hydrophobicity and support a wider range of inter and intra-molecule interactions.

Proteins are synthesized by iteratively joining the carboxyl group of a growing protein chain with the amino group of an amino acid. This condensation reaction is catalyzed by the ribosomes and a stable peptide bond forms under the expulsion of a water molecule. The poly-peptide chain spontaneously folds into a three-dimensional structure with minimal free energy, determined by the chemical properties of the amino acid side chains (Anson and Mirsky, 1930; Lumry and Eyring, 1954; Anfinsen et al., 1961). From an evolutionary point of view, protein structure is ultimately a consequence of function and as many functions demand locally or even globally stable folds, a unique or nearly unique sequence to structure mapping is often possible. This close relationship between sequence, structure and function makes studying the structure of proteins a worthwhile endeavour. Both predicting structure from sequence and predicting function from structure have proven formidable challenges and have been subject to scientific inquiry for more than a century.

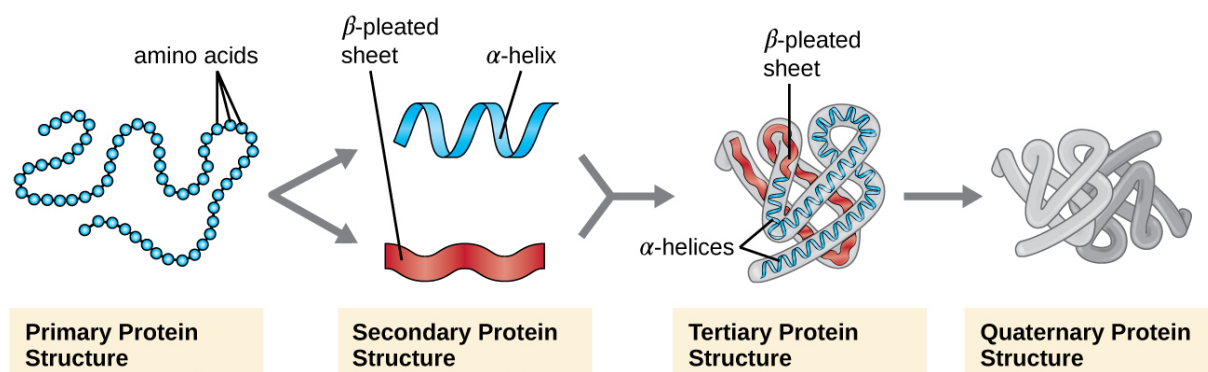


Figure 6.1.: **Protein structure is studied at four levels:** The sequence of amino acid residues (primary structure), locally favorable stable structures, especially α -helix and β -sheets (secondary structure), global fold of a single chain (tertiary structure) and joint fold of several protein chains (quaternary structure) (Figure taken from OpenStax CNX).

6.1. Protein structure

Protein structure is studied at four different levels: the amino acid sequence (primary structure), localized structure elements (secondary structure), global structure (tertiary structure) and multi-chain protein complexes (quaternary structure) (Figure 6.1), which I now discuss in more depth.

6.1.1. Primary structure

The primary structure of a protein is the sequence of amino acids that have been fused to obtain the protein chain. The two ends of the polypeptide chain carry an amino group and a carboxyl group, respectively, and thus impose a directionality. By convention the directionality of a polypeptide sequence is defined from amino end to carboxyl end, coinciding with the direction of the biological synthesis process. The nearly universal genetic code with its translational start and stop signals makes it possible to identify protein sequences by their blueprints in genomic and transcriptomic sequencing data. Nowadays, billions of protein sequences are readily available in databases, making primary structure the most well-known aspect of protein structure (The UniProt Consortium, 2021; Steinegger et al., 2019).

6.1.2. Secondary structure

The secondary structure of a protein describes the local conformations of a polypeptide chain as a consequence of locally energetically favorable interactions. The structure of the protein backbone is determined by triplets of torsion angles between the covalent bonds inside the amino acid units: The nitrogen atom from the amino group to C_α atom ($N - C_\alpha$: ϕ -angle), C_α atom to the carbon atom from the carboxyl group ($C_\alpha - C$: ψ -angle) and the carbon atom from the carboxyl group to the nitrogen atom of the next residue ($C - N$: ω -angle). Due to the double bond character of the involved covalent bonds, the peptide bond is nearly planar and thus $\omega \approx 180^\circ$.

The rotational freedom of the ϕ and ψ angles allows polypeptide chains to adopt an enormous variety of conformations. Hydrogen bonds between the nitrogen atoms of the amino groups as hydrogen bond donors and oxygen atoms of nearby carboxyl groups as hydrogen bond acceptors are however highly energetically favorable, leading to reoccurring stable local folding patterns. The most common of these secondary structure elements are α -helices and β -sheets. In an α -helix, hydrogen bonds form between

the carboxyl group and the amino group four residues further downstream in the sequence. This leads to ϕ/ψ angles of around $-60^\circ/-60^\circ$ and a repetitive right-handed helix-like structure in which 3.6 amino acids complete one helical turn. In contrast to bonding with succeeding residues, the hydrogen bonds in β -sheets are formed between two linear poly-peptide stretches folded onto each other. Depending on the relative directionality of the stretches, β -Sheets are further subdivided into parallel and anti-parallel β -Sheets. Completely straight β -sheets are flat with ϕ/ψ angles of $-120^\circ/+120^\circ$. α -Helix and β -Sheets are energetically preferred by involving all possible hydrogen bond donors and acceptors in electrostatic interactions and reducing steric clashes of side-chains. Apart from the more stable secondary structure elements, less favorable structure elements such as left-handed helices, turns and coils are also possible but less frequent. Thus despite the enormous degrees of freedom in the ϕ/ψ torsion angles of poly-peptide chains, the dihedral angles of stably folded proteins are concentrated in regions that support secondary structure formation (Ramachandran et al., 1963).

6.1.3. Tertiary structure

Tertiary structure describes the global 3D structure of a protein chain in its target medium. There are four broad classes of protein structures: globular proteins, membrane proteins, fibrous proteins, and disordered proteins. As with secondary structure, the tertiary structure is determined by a spontaneous energy minimization process in which the biochemical properties of the amino acid side chains play an important role. Globular proteins are compact in shape and water soluble with hydrophobic side chains clustered in a compact core and polar side-chains exposed at the surface. Membrane proteins are either integrated or associated with a biological membrane. Their native structure often depends on the membrane context. In their native form, fibrous proteins have elongated shapes and fulfil mainly structural roles. Disordered proteins are classified by their absence of stable structure and have recently stepped into the spotlight for their involvement in the formation of membraneless organelles.

The three-dimensional structure of proteins is constrained by physical contacts of residues that while being far from each other in the linear amino sequence, are spatially close in the native structure. The contacts are stabilized by strong covalent bonds in case of disulfide bridges, and weaker, non-covalent electrostatic salt bridges or hydrophobic van der Waals forces.

6.1.4. Quaternary structure

When multiple protein chains fold cooperatively to form a stable multi-chain complex, the resulting structure is referred to as the *quaternary structure* of the involved proteins. In order to fulfil their complex functions in the cells, especially molecular machines are known to be comprised of multiple protein units. The RNA polymerase II, the central player discussed in the previous part for example, consists of 12 subunits (Cramer et al., 2001). As predicting the binding interfaces connecting the individual protein chains can facilitate solving structures of protein complexes, the prediction of protein-protein interfaces naturally extends protein structure prediction problem to the structure prediction of protein complexes.

6.2. Protein evolution

The number of possible polypeptide chains grows exponentially with their length. For lengths of typical proteins, this space of possible sequences is unimaginably large, exceeding by far the number of particles

in the known universe. However only a small fraction of the possible polypeptide chains will fold into stable structures. Stable folds can thus be likened to small islands in a huge sea of unfoldable sequences (Lupas and Koretke, 2008). How many of these islands have been explored in the process of few billion years of evolution is not clear, especially because searching for islands far from those already explored by nature proves difficult (Woolfson et al., 2015).

Inspecting the first available protein structures has revealed that proteins contain independently folding structural building blocks (Wetlaufer, 1973). These protein domains are also evolutionary building blocks that are mixed and matched to bring forth a wide variety of proteins with different functions (Riley and Labedan, 1997; Apic et al., 2001; Ekman et al., 2005). Ontologies have been developed to classify domains according to structure and function (Andreeva et al., 2014, 2020; Sillitoe et al., 2019; Fox et al., 2014). SCOP classifies domains by their secondary structure content (class), secondary structure arrangement (fold), weak and strong sequence and functional homology (super-family, family). CATH uses secondary structure content (class), secondary structure arrangement (architecture), architecture in the context of chain connectivity (topology) and sequence homology (super-family) (Schaeffer and Daggett, 2011; Andreeva et al., 2014; Sillitoe et al., 2019). Currently roughly 1500 distinct folds are annotated with few new folds being detected over the last years despite ever growing sequence databases (Andreeva et al., 2020; Sillitoe et al., 2019; The UniProt Consortium, 2021).

Compared to the huge number of known protein sequences, the total number of domain folds they use is small. The number of naturally evolved folds has been estimated to lie between 1000 and 10000 (Woolfson et al., 2015; Chothia, 1992; Govindarajan et al., 1999; Kolodny et al., 2013; Orengo et al., 1994). This surprisingly small number of folds however does not imply that the number of islands of stable folds is small. More likely, evolution explores folds by assembling existing structure fragments, suggesting a large number of yet unexplored islands (Woolfson et al., 2015; Remmert et al., 2010; Kopec and Lupas, 2013; Cossio et al., 2010).

Proteins fulfil their structural and functional roles in their cellular contexts typically under strong evolutionary pressure. This pressure ultimately affects all levels of protein structure, despite levels closer to the sequence have more flexibility fulfilling the constraints (Zuckerandl, 1976). Whereas function and structure of some core domains is so conserved that it can be traced back to the last universal common ancestor of all life, the primary sequences sharing the same structure and function can be so diverged that inferring shared origin on the sequence level alone becomes challenging (Weiss et al., 2016; Rost, 1999; Remmert et al., 2012).

Protein sequences shaped by evolutionary forces thus share a very limited structural diversity. This many-to-one relationship between sequence and structure makes it possible to transfer structural knowledge to new sequences, a very important concept in protein structure determination (Bowie et al., 1991; Jones et al., 1992b; Chothia and Lesk, 1986).

6.3. Protein structure determination

The difficulty of both experimental and computational approaches to structure determination increases with the level. Due to the decreasing sequencing cost and large-scale metagenomic sequencing efforts, currently available sequence databases contain billions of protein sequences (Steinegger et al., 2019). The central database for protein structures (PDB) in contrast contains 170.000 experimentally derived protein structures at the time of writing, emphasising the desire for computational methods for determining structure from sequence. In the following sections I will briefly introduce experimental and computational

approaches to tertiary and quaternary structure prediction.

6.3.1. Experimental approaches to structure determination

There are currently three core methods available for experimental protein structure determination: X-ray crystallography, NMR spectroscopy and cryogenic electron microscopy. All three methods require protein-specific, non-automatic sample preparation relying heavily on serendipity and trial and error. I will briefly discuss the methods in more details.

X-ray crystallography

X-ray crystallography begins with crystallizing a homogeneous sample of a molecule of interest. Using the diffraction patterns of a powerful X-ray source, the electron density of molecule can be reconstructed, often at atomic resolution. Finding crystallization conditions however is often not straightforward and thus despite concerted effort, a non-negligible amount of domain families still do not have a member with a solved crystal structure. (Ovchinnikov et al., 2017; Montelione, 2012). Close to 90% of all currently resolved structures have been obtained by X-ray crystallography, making it by far the most widely used method for experimental structure determination (The Protein Data Bank, 2021; Berman et al., 2000).

NMR spectroscopy

NMR spectroscopy allows to study protein structure and interactions by measuring structural properties from the magnetic fields induced by spinning nuclei. Having inferred dihedral torsion angles and inter residue distances from NMR data, the protein structure can be obtained by modeling combined with energy minimization (Markwick et al., 2008). In contrast to X-ray crystallography, NMR spectroscopy allows to study soluble molecules in their native environment while additionally resolving protein dynamics. NMR does not require crystallization but is limited in the maximum size of proteins that can be studied. Currently 7.5% of structures deposited in the PDB have been solved with NMR (The Protein Data Bank, 2021; Berman et al., 2000). It has been suggested that X-ray crystallography and NMR spectroscopy are complementary tools for the structure determination of small proteins (Yee et al., 2005).

Cryogenic electron microscopy

Cryogenic electron microscopy, short cryo-EM, detects sample interactions with electron beams to obtain two dimensional projections of biological molecules. To protect the sample from immediate disintegration under the high-energy beam, it is frozen in near-native form in noncrystalline amorphous ice. While cryo-EM imaging requires less sample amount, no sample crystallisation and is more forgiving in case of sample heterogeneity, its limited resolution typically does not yet allow resolving individual side chains (Bai et al., 2015). Recent advances in detectors and software produced the first cryo-EM models at atomic resolution (Nakane et al., 2020; Yip et al., 2020) and thus hints towards cryo-EM becoming a worthy competitor of X-ray crystallography for high-resolution protein structure determination.

6.3.2. Computational approaches to structure prediction

The cost of determining the sequence of a protein has always been lower than determining the structure. Since it was well understood from very early on that the amino acid sequence carries the information of the structure (Anfinsen et al., 1961), computational biologists have long been invested in predicting the three dimensional structure from the sequence (Anfinsen, 1973; Levitt and Warshel, 1975). Purely energy-landscape based optimizations however have proven computationally intractable so far. By shifting from pure theory to methods that can learn from data, protein structure prediction performance has improved continuously, diligently monitored by the biennial CASP competition (Moult et al., 1995). Due to the high complexity, the field of predicting tertiary structure of proteins has specialized in sub-disciplines, among them template-based modeling, template-free modeling, refinement and contact/distance prediction which are briefly introduced below.

The subdisciplines of structure prediction

Template-based modeling describes the task of predicting protein structure by transferring information from evolutionarily related sequences with known structure. As structure is more conserved than sequence (Illergård et al., 2009), diverging sequences preserve structural similarity over long evolutionary time. However, identifying structural homologs quickly becomes challenging as sequence identity drops below 30% (Rost, 1999). Highly sensitive homology detection tools based on profile-profile search have been developed to facilitate template-based modelling on homologous yet strongly diverged sequences (Altschul et al., 1997; Söding, 2005; Söding et al., 2005). The query-target alignment produced by homology prediction software is the input to homology modeling software that build a structural model of the query sequence based on the query-target alignment and the structure of the target sequence (Krivov et al., 2009; Webb and Sali, 2016; Waterhouse et al., 2018).

Template-free modeling is the much harder task of predicting protein structure in the absence of known structure homologs and is thus closer to the original goal of folding based on the sequence alone. This however proves difficult and successful template-free modeling have been data-driven approaches to find structures good enough to serve as seeds for successive energy-based structure refinement. Current approaches are mainly based on the prediction of short-range local protein structure and long-range pairwise residue interaction data calculated on large alignments of homologous sequences (Kuhlman and Bradley, 2019). *Fragment assembly* uses structural knowledge in form of short structure fragments in combination with statistical sampling methods in order to solve local structure (Simons et al., 1997, 2001; Bonneau et al., 2001; Jones and McGuffin, 2003). In an iterative approach, a structure fragment is proposed for a short stretch of amino acids by sampling from a fragment library generated from known structures. The fragment is accepted probabilistically with a high acceptance rate if the proposed change decreases the energy and is likely rejected otherwise (Kuhlman and Bradley, 2019; Simons et al., 1997). By allowing changes that increase the energy with low probability, with enough time sampling procedures can escape local minima in the energy landscape while exploring regions with low overall energy.

Knowledge of amino acids that are far apart in the linear amino acid chain yet close in the three-dimensional structure simplifies the protein folding problem by restricting the search space. It has been shown that knowing one contact for every 12 residues suffices to fold proteins (Kim et al., 2014). Predicted long-range contacts are thus an important source of information in template-free modeling approaches. With ever advancing deep-learning techniques, the information density in long-range interactions has been increased by moving from binary contact or no-contact classification to predicting the distance

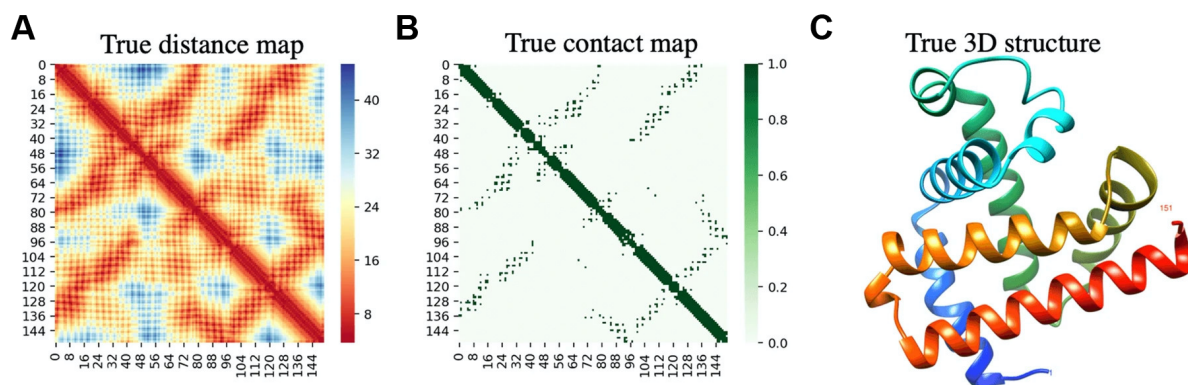


Figure 6.2.: **Protein structure, amino-acid pair distances and contacts.** The stable structure of a protein (C) can be viewed as a symmetric distance map, color coding the distances between all pairs of amino acids in the structure (A). Contact maps are binarized versions of distance maps in which amino acid pairs with a $C_{\beta} - C_{\beta}$ distance below 8 Å are assigned a value of one, all other pairs are assigned a value of zero. The goal of contact prediction is to predict contact maps for proteins with unknown structure. Accurately predicted contact maps can contribute to solving protein structures by constraining the search space. Figure adapted from Adhikari.

distribution between amino acid pairs (Yang et al., 2020; Xu, 2019; Senior et al., 2019). Due to its importance for the objectives of this thesis, I have dedicated section 6.3.3 to discuss contact prediction in more depth.

Model refinement describes the process of improving protein models further by resolving physically impossible atom configurations with fine-grained energy models. To this end, two strategies have been successfully applied. Molecular dynamics simulations simulate the protein in solution and calculate the Newtonian forces acting on the atoms. The motion of the atoms can be simulated for a short time step followed by recalculation of the forces. Iterative motion and force reevaluation allows to simulate the trajectories of the protein model towards an energetically more favorable state (Kuhlman and Bradley, 2019; Heo and Feig, 2018). An alternative strategy to model refinement are side-chain rotamer sampling Monte-Carlo simulations with energy force fields at atomic resolution (Kuhlman and Bradley, 2019; Raman et al., 2009; Leaver-Fay et al., 2011).

6.3.3. Contact prediction

Contact prediction is the task of predicting structural amino acid interactions (typically $C_{\beta} - C_{\beta}$ distance smaller than 8 Å) in the absence of structural information and thereby gaining information about protein structure ab-initio (Figure 6.2). Coevolution by compensatory mutations is a central source of information in modern contact prediction algorithms (Schaarschmidt et al., 2018). It has long been observed that amino acids that vary together in the sequence are connected by a shared structural or functional relationship (Wyckoff, 1968; Fitch and Markowitz, 1970; Altschuh et al., 1987). In order to be able to withstand the forces of evolutionary pressure, a destabilizing mutation in one amino acid often needs to be compensated by a re-stabilizing mutation in its interaction partner (Figure 6.3). Over the years, three classes of methods have been developed to extract coevolutionary signal from multiple sequence alignments which I introduce in more detail.

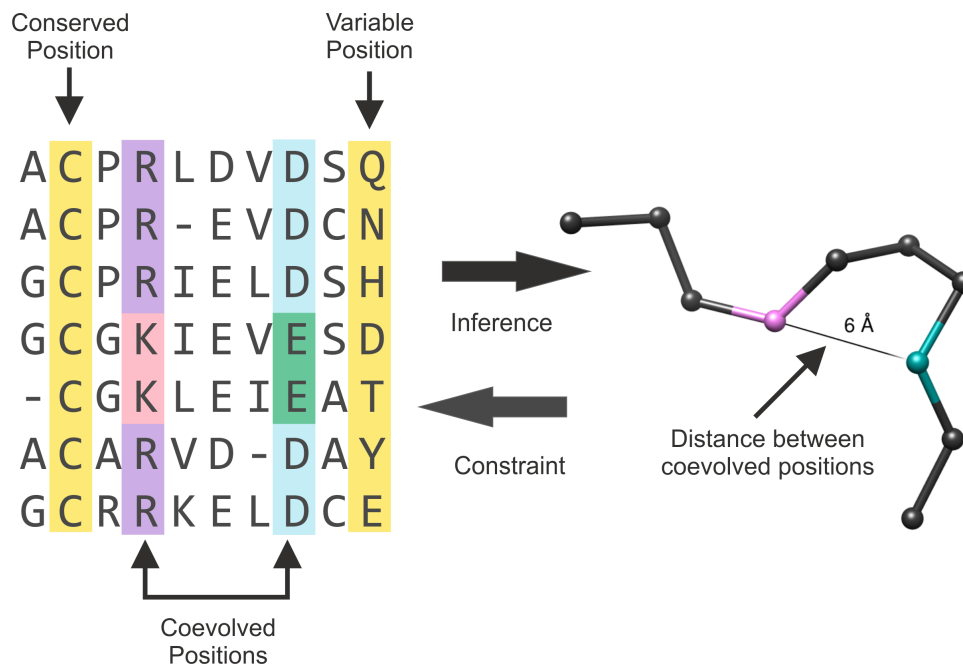


Figure 6.3.: **Coevolution signal encodes structural information.** Structure and function critical protein contacts impose evolutionary constraints on the interacting amino acids. When destabilizing mutations are overcome by compensatory mutations in the interaction partner, the corresponding columns in the multiple sequence alignment show correlated amino acid exchange patterns (Figure taken from MISTIC web server).

Pairwise methods use the amino acid tuples in a pair of alignment columns to assign each possible column pair a contact score. The output of pairwise methods are two-dimensional, symmetric score matrices reminiscent of the contact maps derived from known protein structures. To this end different classes of methods have been proposed: *Observed minus expected squared* (OMES) methods statistically test the compatibility of the observed number of amino acid pair combinations with the expected pair counts derived from the individual column frequencies under the independence assumption (Larson et al., 2000; Kass and Horovitz, 2002). Methods based on information theory use mutual information to quantify the information content shared between two columns (Chiu and Kolodziejczak, 1991; Korber et al., 1993; Wollenberg and Atchley, 2000). Yet another approach calculates for each alignment column a vector of pairwise similarity scores of all amino acids pairs observed in the alignment column. The contact score for a column pair is then calculated as the Pearson correlation of its similarity vectors (Göbel et al., 1994; Olmea et al., 1999). More methods have been developed to the same end but are not discussed in further detail (Neher, 1994; Shindyalov et al., 1994; Taylor and Hatrick, 1994; Lockless and Ranganathan, 1999). With the Average Product Correction (APC), developed to remove entropic, phylogenetic and sampling biases, the mutual information approach became the most sensitive method in 2008 (Dunn et al., 2008) shortly before global methods displaced pairwise methods in coevolution detection.

Global methods In 2009, direct coupling analysis (DCA), a novel approach based on statistical physics has revolutionized the field of contact prediction. Instead of calculating interaction scores independently for pairs of alignment columns, a global approach to contact prediction was proposed by learning a protein-family specific joint probability distribution $p(\mathbf{x})$ of proteins in a protein alignment. The model is chosen such that it is able to correctly capture the empirically observed single ($Np(x_i = a) = n_{ia}$) and pairwise ($Np(x_i = a, x_j = b) = n_{ijab}$) amino acid counts while requiring the least number of parameters (Weigt et al., 2009). The model specified by these constraints takes the form $p(\mathbf{x}|\mathbf{v}, \mathbf{w}) =$

$\frac{1}{Z} \exp\left(\sum_i v_i(x_i) + \sum_{i < j} w_{ij}(x_i, x_j)\right)$ with $L \times A$ singleton parameters \mathbf{v} and $\binom{L}{2} \times A \times A$ coupling parameters \mathbf{w} and a normalization constant Z that ensures that $\sum_{\mathbf{x}} p(\mathbf{x}|\mathbf{v}, \mathbf{w}) = 1$. It can be interpreted as an undirected graphical model, with the alignment positions as nodes and couplings as edges. A competing directed graphical model proposed shortly afterwards has not been able to compete with improved learning procedures for undirected graphical models (Burger and Van Nimwegen, 2010; Ekeberg et al., 2014).

The key advantage of the global approach is its ability to dissect the correlations of pairwise approaches into direct and indirect contributions. When alignment column i is coupled to alignment column j and alignment column j in turn is coupled to alignment column k , there will be correlation signatures between all three columns. The correlation signature between i and k is however a non-causal, indirect effect leading to a misleading correlation that pairwise methods cannot account for (Weigt et al., 2009). The authors of the original publication derive a direct information score which is reminiscent of the mutual information score but quantifies only direct contributions (Weigt et al., 2009).

The model formulation of $p(\mathbf{x}|\mathbf{v}, \mathbf{w})$, while theoretically appealing, is computationally challenging due to its normalization constant Z summing over the full sequence space (20^L), a number of summands so large that it cannot be computed for realistic alignment lengths L . Several methods have been proposed to learn the model parameters. A message-passing algorithm, used in the original paper, was limited to small alignment depths (Weigt et al., 2009). Regularized pseudolikelihood approaches approximate the likelihood such that the normalization Z becomes tractable (Morcos et al., 2011; Balakrishnan et al., 2011; Jones et al., 2012; Kamisetty et al., 2013; Ekeberg et al., 2014). In the past, our lab has contributed an ultra-fast pseudolikelihood implementation for CPU/GPU (Seemayer et al., 2014). A method to learn the true likelihood based on *persistent contrastive divergence* has recently been developed in our lab (Tieleman, 2008; Vorberg et al., 2018). Best performing pseudo-likelihood approaches reduce phylogenetic effects by reweighting the sequences according to their similarities, use a L_2 penalty on the coupling parameters during learning and calculate raw contact scores c_{ij} by taking the Frobenius norm of the 400 coupling parameters for each column pair (i,j) : $c_{ij} = \|\mathbf{w}_{ij}\|_2$ (Ekeberg et al., 2013, 2014). Curiously, normalizing the raw contact scores c_{ij} by applying the APC correction developed for mutual information in pairwise methods proved to increase the quality of predicted contacts and has been the state-of-the-art correction ever since (Dunn et al., 2008; Ekeberg et al., 2014).

Contact predictions based on DCA became state-of-the-art after outperforming competitors by a large margin in CASP11 (Monastyrskyy et al., 2016; Kosciolok and Jones, 2016; Jones et al., 2015) and the derived contact maps have since proven to be an invaluable information source for the next generation of contact predictors based on deep computer vision models (Schaarschmidt et al., 2018). Their future value is currently being challenged by Transformer-based architectures of the latest generation of deep language models which seem to be able to learn the information provided by DCA methods directly from multiple sequence alignments (Bhattacharya et al., 2020; Rao et al., 2021). Transformer-based architectures hence paved the way to the current state-of-the-art end-to-end structure prediction method AlphaFold2 (Jumper et al., 2020).

Transformation due to deep learning

Protein folding is a field with an enormous wealth in available data and it is thus a primary candidate for disruption by deep learning technologies. Until very recently, state-of-the-art protein structure predictors have been complex pipelines of independent sub tasks. When in 2016 deep learning methods entered competitive protein structure prediction, the first landslide improvements were achieved by applying

breakthroughs in the field of computer vision to the sub task of contact map predictions (Schaarschmidt et al., 2018; Wang et al., 2017). Subsequent refinement in the deep learning models allowed to go from contact maps to distance maps (Senior et al., 2019; Xu, 2019; Yang et al., 2020) and the prediction of more elaborate structural constraints such as pairwise local geometry (Senior et al., 2019; Yang et al., 2020). In the meantime, the status-quo of protein structure prediction as a pipeline of independent tasks has been challenged by visionary attempts to learn protein structure from MSA to structure end-to-end in a single deep learning model (AlQuraishi, 2019; Ingraham et al., 2019). End-to-end learning describes learning objectives that are differentiable from input to output and hence prediction errors at the output can propagate all the way back to the input, avoiding the inevitable information bottlenecks arising from manually crafted features. While in the end the method did not achieve competitive performance, it surely inspired the out-of-the-box thinking that would enable scientists at DeepMind to rise to the challenge by leaving the previous methodology behind. Their model AlphaFold2, presented only few months ago from the time I write this thesis, achieved unprecedented performance with an end-to-end Transformer model (Protein Structure Prediction Center, 2020; Jumper et al., 2020).

Deep computer vision models. With more and more sequences, structures and contact prediction methods available, data-integrating machine learning pipelines improved contact prediction by generalizing in supervised learning objectives (Skwark et al., 2013; Jones et al., 2015). When deep learning models entered the contact prediction field, they excelled by applying architectural innovations from the computer vision field such as ConvNets and ResNets (Skwark et al., 2014; Wang et al., 2017; Schaarschmidt et al., 2018). The preceding global methods however retained their importance by providing coevolution features crucial for competitive performance (Schaarschmidt et al., 2018). The deep neural networks were not only able to capture local dependencies in protein contact maps arising from local secondary structure and secondary structure element interactions, but also efficiently integrate large amounts of supplementary data. Further improvements have been made by refining the network architectures and integrating large amounts of data and predicting quantitative distances instead of binary contacts (Senior et al., 2019; Li et al., 2019; Xu, 2019; Yang et al., 2020).

Deep language models. Very recently, Transformer-based language models have proven effective for extracting structural information from protein sequences (Rives et al., 2019). In contrast to deep convolutional networks that harnessed the power of deep learning by likening contact maps to images, the deep Transformer-based models liken amino acids to words in long sentences of proteins with the aim of learning the "grammar of proteins". As with graphical models, but unlike the deep computer vision models, deep language models can be trained in an unsupervised manner, meaning that the methods capture the information without generalizing from annotated contacts. The unsupervised training procedure is also referred to as self-supervised and can be performed by predicting a small subset of intentionally hidden amino acids from training sequences and propagating the prediction errors back through the network. Despite having no explicit knowledge of protein evolution and structure, self-supervised training of deep language models captures deep biological concepts such as sequence homology, alignment within protein families and secondary structure and long-range contacts when trained on very large number of sequences (Rives et al., 2019; Elnaggar et al., 2020). Moreover it has been shown that self-attention, a key feature of the Transformer architecture, can be used to predict contacts as accurately as undirected graphical models while sharing amino-acid pair specific parameters across protein families (Bhattacharya et al., 2020) and that Transformers trained directly on a large number of multiple sequence alignments outperform computer-vision models (Rao et al., 2021). AlphaFold2, by a large margin the best protein structure predictor in CASP14, uses an end-to-end Transformer-based architecture to extract information directly

from the alignments Jumper et al. (2020).

Noise and bias in coevolution signal. Three sources of noise have been identified to dilute coevolution signal: entropic noise, phylogenetic noise and finite sample size effects (Atchley et al., 2000; Martin et al., 2005; Dunn et al., 2008; Vorberg et al., 2018). *Entropic noise* describes the effect that the distribution of amino acids in alignment columns influences coevolution scores. It has been shown that dividing mutual information by the column entropies improved the signal to noise ratio (Martin et al., 2005). *Phylogenetic noise* arises from the common history that sequences in a multiple sequence alignment share as a consequence of evolution. The theoretical frameworks used for detecting coevolution such as mutual information and DCA assume sequence independence, leading to misleading signal from clusters of related sequences. Three strategies have been used to correct for phylogenetic noise: (i) down-weighting closely related sequences with the help of sequence weights (Morcos et al., 2011; Ekeberg et al., 2013), (ii) using variants of the parametric bootstrap to assign significance (Wollenberg and Atchley, 2000; Colavin et al., 2020) and (iii) normalizing coevolution matrices (Dunn et al., 2008). *Noise due to finite sample sizes* is strongest in multiple sequence alignments with few sequences and is addressed by bootstrapping approaches and statistical testing, but not by the APC normalization, a current gold-standard (Dunn et al., 2008). The regularization techniques employed for global models can attenuate the sampling bias by exerting higher shrinkage on parameters associated with fewer empirical counts. Based on simulations in previous work from this group, entropy contributes twice as much to the overall noise as phylogeny.

Recently, an attempt to a theoretically principled phylogeny correction has been made by removing the assumption of sample independence for DCA by assuming a independent column-pair evolutionary model. Given a phylogenetic tree, new singleton and pair frequencies can be calculated that do not contain the influences of the underlying phylogeny. By constructing synthetic alignments that obey the corrected singleton and pair frequencies, standard contact prediction methods such as plmDCA can be applied. Despite improved parameter estimations on synthetic data, the method did not prove superior on real proteins (Rodriguez Horta et al., 2019).

Evaluation of contact predictions

As the number of contacts scales almost linearly with the length of the protein (Vendruscolo et al., 1997; Skwark et al., 2014) contact prediction performance is often evaluated in precision at fixed cutoffs as a function of alignment length L , such as $L/2$ and $L/5$ (Schaarschmidt et al., 2018). It is important to bear in mind that contact prediction is a means to structure prediction and thus the precision metric for correctly predicted contacts is merely an easily computable proxy of what we are truly interested in (Schaarschmidt et al., 2018). Especially a large number of correct yet highly clustered contact predictions will boost the precision score while contributing mostly redundant information to the task of protein folding under distance constraints (Jones et al., 2015).

6.4. Aims and scope

The aim of this project was to derive FS-PCD, a novel method for deriving coevolution features adjusted for the shared evolutionary history of the input sequences. Using the independent-pair model previously used by Rodriguez Horta et al. (2019), we derive an efficient pair-column variation of Felsenstein’s tree-pruning algorithm. The resulting phylogenetically corrected sufficient statistics can be used to train a Markov Random Field with the Persistent Contrastive Divergence algorithm, previously implemented by

our group (Vorberg et al., 2018). We furthermore compare the performance of FS-PCD to the phylogeny-unaware counterpart.

7. Methods

7.1. Felsenstein's pruning algorithm for independent pairs

7.1.1. A family-specific pairwise evolutionary model

In order to correct the phylogeny, we introduce a pairwise model of evolution which expresses our belief that a pair of amino acids (a, b) mutates into the pair (c, d) after δ_t units of evolutionary time have passed. We denote our evolutionary model $p(c, d|a, b; \delta_t)$.

Our first evolutionary model assumes that at each mutation event the new amino acids are drawn from an equilibrium model independent of the preceding amino acid. After δ_t time units, there are three possible outcomes for the amino acids in the pair (a, b) : (i) none mutated, (ii) one mutated at least once or (iii) both mutated at least once. For a single amino acid we can obtain the probability $r = e^{-\delta_t}$ that no mutations occurred from the Poisson distribution $\mathcal{P}(x = 0|\lambda = \delta_t)$. We assume further that mutation events occur independently from each other at all sites with the same rate and thus the probabilities of observing a mutation for the three cases become (i) r^2 , (ii) $r \times (1 - r)$ and $(1 - r)^2$.

The final part is a model that captures our believe for amino acid replacement in case of mutations. Here we assume that the replacement process is memoryless and time-invariant. We model the protein-family specific equilibrium model $p(x_i = c, x_j = d)$ with a Markov Random Field over the two pair columns i and j .

$$p(c, d) := p(x_i = c, x_j = d) = \frac{\exp(v_i(c) + v_j(d) + w_{ij}(c, d))}{\sum_{c', d'} \exp(v_i(c') + v_j(d') + w_{ij}(c', d'))} \quad (7.1)$$

$$p(c, d|\cdot, d) := p(x_i = c|x_j = d) = \frac{\exp(v_i(c) + w_{ij}(c, d))}{\sum_{c'} \exp(v_i(c') + w_{ij}(c', d))} \quad (7.2)$$

, and

$$p(c, d|c, \cdot) := p(x_j = d|x_i = c) = \frac{\exp(v_j(d) + w_{ij}(c, d))}{\sum_{d'} \exp(v_j(d') + w_{ij}(c, d'))}. \quad (7.3)$$

With the amino acid exchange model in hand, we can now express our believe in amino acid exchanges as a function of the evolutionary time passed:

$$p(c, d|a, b; \delta_t) = \begin{cases} r^2 \delta_{ac} \delta_{bd} & \text{none mutated (i)} \\ r(1 - r) (p(c, d|\cdot, d) \delta_{bd} + p(c, d|c, \cdot) \delta_{ac}) & \text{one mutated at least once (ii)} \\ (1 - r)^2 p(c, d) & \text{both mutated at least once (iii)} \end{cases} \quad (7.4)$$

The Kronecker delta δ_{xy} – a function that evaluates to 1 if x equals y and 0 otherwise – is used to assign zero probability to impossible configurations. E.g. case (i) is only possible if we observe the the same pair of amino acids before and after δ_t time units have passed. Note that the reverse is not true: If we observe the same pair, all three cases (i-iii) are possible explanations.

Since the three cases (i-iii) contain all possible outcomes and are pairwise mutually exclusive, we can obtain the final form of the evolutionary model by summing the individual probabilities.

$$p(c, d|a, b; t_{lm}) = r^2 \delta_{ac} \delta_{bd} + r(1-r) (p(c, d|\cdot, d) \delta_{bd} + p(c, d|c, \cdot) \delta_{ac}) + (1-r)^2 p(c, d) \quad (7.5)$$

This model is parameterized by the column-pair specific parameters $\mathbf{v}_i, \mathbf{v}_j$ (40) and interaction parameters (400) \mathbf{w}_{ij} , in total $\binom{L}{2} \times 440$ parameters which are to be estimated by a Maximum-Likelihood approach.

7.1.2. Calculating the likelihood

In this section we derive for a pair of columns (i, j) the likelihood $p(\mathbf{X}_0|\mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij})$ of observing the amino acid configuration \mathbf{X}_0 in contemporary species, given the evolutionary tree \mathcal{T} and the 440 column-pair specific protein family parameters $\mathbf{v}_i, \mathbf{v}_j$ and \mathbf{w}_{ij} with the goal to optimize $\mathbf{v}_i, \mathbf{v}_j$ and \mathbf{w}_{ij} numerically. Here we assume that the underlying phylogenetic tree \mathcal{T} is known. I will discuss a method for deriving the tree in section 7.1.7.

Starting from the formulation of the likelihood, we use the law of total probability and condition $p(\mathbf{X}_0|\mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij})$ on the 400 possible pairwise amino acid configurations at the root node to obtain:

$$p(\mathbf{X}_0|\mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}) = \sum_{a,b=1}^{20} p(\mathbf{X}_0|a, b, \mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}) p(a, b|\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}) \quad (7.6)$$

In order to facilitate readability I will keep the dependencies on the tree \mathcal{T} and the model parameters $\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}$ implicit in the following and thus equation 7.6 becomes:

$$p(\mathbf{X}_0) = \sum_{a,b=1}^{20} p(\mathbf{X}_0|a, b) p(a, b) \quad (7.7)$$

The phylogenetic relationship is modelled by a rooted binary tree and thus at each internal node the flow of information splits into two independent branches. We can therefore express the conditional likelihood $p(\mathbf{X}_l|a, b)$ of an arbitrary parent node l recursively as the product of the likelihoods of their two child nodes n and m . As the internal nodes represent ancestral species with unknown amino acid configurations, we sum over all 400 possible amino acid pair combinations.

$$p(\mathbf{X}_l|a, b) = \left(\sum_{c,d=1}^{20} p(\mathbf{X}_m|c, d) p(c, d|a, b; t_{lm}) \right) \left(\sum_{e,f=1}^{20} p(\mathbf{X}_n|e, f) p(e, f|a, b; t_{ln}) \right) \quad (7.8)$$

Plugging in the evolutionary model from equation 7.5, we can rewrite the two sums as:

$$\begin{aligned} p(\mathbf{X}_m|a, b; r) &:= \sum_{c,d=1}^{20} p(\mathbf{X}_m|c, d) p(c, d|a, b; r) \\ &= (1-r)^2 p(\mathbf{X}_m) + r(1-r) (p(\mathbf{X}_m|\cdot, b) + p(\mathbf{X}_m|a, \cdot)) + r^2 p(\mathbf{X}_m|a, b) \end{aligned} \quad (7.9)$$

where we defined

$$p(\mathbf{X}_m) := \sum_{c,d=1}^{20} p(\mathbf{X}_m|c, d) p(c, d) \quad (7.10)$$

$$p(\mathbf{X}_m|\cdot, b) := \sum_{c=1}^{20} p(\mathbf{X}_m|c, b) p(c, b|\cdot, b) \quad (7.11)$$

$$p(\mathbf{X}_m|a, \cdot) := \sum_{d=1}^{20} p(\mathbf{X}_m|a, d) p(a, d|a, \cdot) \quad (7.12)$$

With all this in hand, we can derive a recursive evaluation schema that evaluates the likelihood $p(\mathbf{X}_0|\mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij})$ bottom-up from the leafs to root. For given $i, j, \mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}$:

1. Calculate $p(c, d)$, $p(c, d|\cdot, d)$ and $p(c, d|c, \cdot)$ from $\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}$ according to equations 7.1–7.3.
2. Initialize the N leaf nodes with $p(\mathbf{X}_n|a, b) = \delta_{a, x_{ni}} \delta_{b, x_{nj}}$ and calculate $p(\mathbf{X}_n)$, $p(\mathbf{X}_n|\cdot, b)$ and $p(\mathbf{X}_n|a, \cdot)$ according to equations 7.10–7.12.
3. From bottom-up calculate $p(\mathbf{X}_l|a, b)$ for each internal node by evaluating equation 7.8 and $p(\mathbf{X}_l)$, $p(\mathbf{X}_l|\cdot, b)$ and $p(\mathbf{X}_l|a, \cdot)$ according to equations 7.10–7.12.
4. Having arrived at the root, the final likelihood can be calculated by equation 7.7.

For one pair (i, j) , a sequence alignment of N sequences and the amino acid alphabet \mathcal{A} , this algorithm requires $\mathcal{O}(|\mathcal{A}|^2)$ computations for each node, and thus has a time complexity of $\mathcal{O}((2N - 1)|\mathcal{A}|^2)$.

7.1.3. Derivatives of the likelihood w.r.t to the model parameters

Our goal is to determine the model parameters that maximise the likelihood using a gradient-based numerical optimization scheme. As we have seen in section 7.1.2, the likelihood of a column pair (i, j) only depends on the 440 model parameters $\mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}$. With the partial derivatives $\frac{\partial p(\mathbf{X}_0)}{\partial w_{ijab}}, \frac{\partial p(\mathbf{X}_0)}{\partial v_{ia}}, \frac{\partial p(\mathbf{X}_0)}{\partial v_{jb}}$ in hand, we can obtain the $\binom{L}{2} \times 440$ model parameters by solving $\binom{L}{2}$ independent optimization tasks. In this section we derive the partial derivatives of the likelihood with respect to the model parameters:

Taking the partial derivatives with respect to θ we obtain

$$\frac{\partial p(\mathbf{X}_0)}{\partial \theta} = \sum_{a,b=1}^{20} \frac{\partial p(\mathbf{X}_0|a, b)}{\partial \theta} p(a, b) + p(\mathbf{X}_0|a, b) \frac{\partial p(a, b)}{\partial \theta} \quad (7.13)$$

$$\frac{\partial p(\mathbf{X}_m|a, b; r)}{\partial \theta} = (1-r)^2 \frac{\partial p(\mathbf{X}_m)}{\partial \theta} + r(1-r) \left(\frac{\partial p(\mathbf{X}_m|\cdot, b)}{\partial \theta} + \frac{\partial p(\mathbf{X}_m|a, \cdot)}{\partial \theta} \right) + r^2 \frac{\partial p(\mathbf{X}_m|a, b)}{\partial \theta} \quad (7.14)$$

$$\frac{\partial p(\mathbf{X}_i|a, b)}{\partial \theta} = \frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial \theta} p(\mathbf{X}_n|a, b; r_{ln}) + p(\mathbf{X}_m|a, b; r_{lm}) \frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial \theta}. \quad (7.15)$$

for Equations 7.8, 7.7 and 7.9.

$$\frac{\partial p(\mathbf{X}_m)}{\partial \theta} = \sum_{c', d'=1}^{20} \frac{\partial p(\mathbf{X}_m|c', d')}{\partial \theta} p(c', d') + \sum_{c', d'=1}^{20} p(\mathbf{X}_m|c', d') \frac{\partial p(c', d')}{\partial \theta} \quad (7.16)$$

$$\frac{\partial p(\mathbf{X}_m|\cdot, b)}{\partial \theta} = \sum_{c'=1}^{20} \frac{\partial p(\mathbf{X}_m|c', b)}{\partial \theta} p(c', b|\cdot, b) + \sum_{c'=1}^{20} p(\mathbf{X}_m|c', b) \frac{\partial p(c', b|\cdot, b)}{\partial \theta} \quad (7.17)$$

$$\frac{\partial p(\mathbf{X}_m|a, \cdot)}{\partial \theta} = \sum_{d'=1}^{20} \frac{\partial p(\mathbf{X}_m|a, d')}{\partial \theta} p(a, d'|a, \cdot) + \sum_{d'=1}^{20} p(\mathbf{X}_m|a, d') \frac{\partial p(a, d'|a, \cdot)}{\partial \theta} \quad (7.18)$$

for equations 7.10–7.12.

Finally the partial derivatives of the amino acid replacement model (equations 7.1–7.3) by the the model parameters:

$$\frac{\partial p(a, b)}{\partial w_{ijcd}} = (\delta_{ac}\delta_{bd} - p(c, d)) p(a, b) \quad (7.19)$$

$$\frac{\partial p(a, b|\cdot, b)}{\partial w_{ijcd}} = \delta_{bd} (\delta_{ac} - p(c, d|\cdot, d)) p(a, b|\cdot, b) \quad (7.20)$$

$$\frac{\partial p(a, b|a, \cdot)}{\partial w_{ijcd}} = \delta_{ac} (\delta_{bd} - p(c, d|c, \cdot)) p(a, b|a, \cdot) \quad (7.21)$$

$$\frac{\partial p(a, b)}{\partial v_{ic}} = (\delta_{ac} - p(c, \cdot)) p(a, b) \quad (7.22)$$

$$\frac{\partial p(a, b)}{\partial v_{jd}} = (\delta_{bd} - p(\cdot, d)) p(a, b) \quad (7.23)$$

$$\frac{\partial p(a, b|\cdot, b)}{\partial v_{ic}} = (\delta_{ac} - p(c, b|\cdot, b)) p(a, b|\cdot, b) \quad (7.24)$$

$$\frac{\partial p(a, b|\cdot, b)}{\partial v_{jd}} = 0 \quad (7.25)$$

$$\frac{\partial p(a, b|a, \cdot)}{\partial v_{ic}} = 0 \quad (7.26)$$

$$\frac{\partial p(a, b|a, \cdot)}{\partial v_{jd}} = (\delta_{bd} - p(a, d|a, \cdot)) p(a, b|a, \cdot) \quad (7.27)$$

With derivatives 7.13–7.27 in hand, the 440 partial derivatives $\frac{\partial p(\mathbf{X}_0)}{\partial w_{ijab}}$, $\frac{\partial p(\mathbf{X}_0)}{\partial v_{ia}}$ $\frac{\partial p(\mathbf{X}_0)}{\partial v_{jb}}$ can be calculated with a recursive scheme analogous to that in section 7.1.2:

1. Calculate the partial derivatives of $p(c, d)$, $p(c, d|\cdot, d)$ and $p(c, d|c, \cdot)$ with respect to all model parameters v_{ia} , v_{jb} and w_{ijab} according to equations 7.19–7.27.
2. Initialize the partial derivatives at the N leaf nodes with 0 and calculate the partial derivatives of

- $p(\mathbf{X}_n)$, $p(\mathbf{X}_n|\cdot, b)$ and $p(\mathbf{X}_n|a, \cdot)$ with respect to the model parameters according to equations 7.16–7.18.
3. From bottom-up, calculate the partial derivatives of $p(\mathbf{X}_l|a, b)$ with respect to the model parameters for each internal node using evaluating equation 7.15 and the partial derivatives of $p(\mathbf{X}_l)$, $p(\mathbf{X}_l|\cdot, b)$ and $p(\mathbf{X}_l|a, \cdot)$ according to equations 7.16–7.18.
 4. Having arrived at the root, $\frac{\partial p(\mathbf{X}_0)}{\partial w_{ijab}}$, $\frac{\partial p(\mathbf{X}_0)}{\partial v_{ia}}$ and $\frac{\partial p(\mathbf{X}_0)}{\partial v_{jb}}$ can be calculated by equation 7.13.

For one pair (i, j) , a sequence alignment of N sequences and the amino acid alphabet \mathcal{A} , this algorithm requires $\mathcal{O}(|\mathcal{A}|^4)$ computations for each node, and thus has a time complexity of $\mathcal{O}((2N - 1)|\mathcal{A}|^4)$.

7.1.4. Optimizing the likelihood

With algorithms to calculate the likelihood and its partial derivatives with respect to the model parameters in hand, we can now derive a gradient-based optimization scheme for the model parameters. The input is a multiple sequence alignment with N sequences and L columns and a rooted phylogenetic tree with nodes representing species and the branch lengths quantifying the units of evolutionary time passed between two species. The output are the $\binom{L}{2} \times 440$ protein-family specific model parameters as introduced in section 7.1.1.

For each column pair (i, j) , do the following:

1. Initialize v_{ia} , v_{jb} with the logarithm of the relative amino acid frequency in the column, i.e. $\log[\hat{p}(x_{ni}=a)]$ and $\log[\hat{p}(x_{nj}=b)]$, respectively. Initialize all w_{ijab} with 0.
2. Until convergence calculate updates Δv_{ia} , Δv_{jb} and Δw_{ijab} that increase the likelihood with L-BFGS Virtanen et al. (2020); Byrd et al. (1995). Calculate $p(\mathbf{X}_0)$, $\frac{\partial p(\mathbf{X}_0)}{\partial w_{ijab}}$, $\frac{\partial p(\mathbf{X}_0)}{\partial v_{ia}}$ and $\frac{\partial p(\mathbf{X}_0)}{\partial v_{jb}}$ according to the algorithms presented in sections 7.1.2 and 7.1.3 as required for L-BFGS's line-search.
3. Upon convergence, store the 440 parameters \mathbf{v}_i^* , \mathbf{v}_j^* , \mathbf{w}_{ij}^* as part of the output.

After processing all $\binom{L}{2}$ column pairs, the $\binom{L}{2} \times 440$ parameters are returned as the output.

7.1.5. Improvements to the core algorithm

Transformation to logspace

The likelihood is a probability and as such a non-negative real value between 0 and 1. Real numbers by representing them as the product $s \times m \times 2^e$, commonly referred to as *floating-point representation*. The mantissa $1 \leq m < 2$ carries the precision, the exponent e determines the magnitude of the number, finally s is the sign (-1 for negative numbers and $+1$ else). Modern computer architectures offer high-performance arithmetic on signed single precision (1-bit sign + 23 bit mantissa + 8 bit exponent) and signed double precision (1 bit sign + 52 bit mantissa + 11 bit mantissa). Signed double precision can capture probabilities of orders of magnitudes as small as 10^{-308} . While this number is incomprehensible small to human minds (the visible universe is estimated to consist of 10^{80} particles!), the likelihoods of medium-large alignments can be magnitudes smaller than 10^{-308} and thus become numerically not representable by double precision floating point numbers. This is a result by the exponential growth of possible number of leaf configurations as the number of leaves increases. A phylogenetic tree with 128

leaf nodes has already $400^{128} \approx 10^{333}$ possible leaf configurations. As the law of total probability demands that the sum of likelihood over all leaf-configurations $\sum_{\mathbf{x}_0} p(\mathbf{X}_0 | \mathcal{T}, \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}) = 1$, independent of the initial choice of \mathbf{v}_i^0 , \mathbf{v}_j^0 and \mathbf{w}_{ij}^0 there are already at least one leaf configuration that is not representable with signed double precision floating-point numbers. While in practise the initial parameters \mathbf{v}_i^0 , \mathbf{v}_j^0 and \mathbf{w}_{ij}^0 are chosen such that likelihoods for $N = 128$ are still representable with double precision, with growing N this is not possible and such the aforementioned algorithm is limited to small phylogenetic alignments.

A common way to represent numbers outside the range of floating-point numbers is to represent them in logarithmic scale, i. e. as their logarithm in floating-point representation. Apart from expanding the range of representable numbers, the logarithm is also a monotonous function and therefore parameters that maximize the likelihood also maximise the log likelihood. We can thus obtain optimal parameters of the likelihood without having to switch from the logarithmic back to the linear scale and thus not running into the risk to have non-representable numbers (floating-point overflow). When performing calculations in logarithmic scale it is important that all operations that require bridging logarithmic and linear scale are performed such that overflows are not possible.

The logsumexp trick The most important operation bridging the linear and logarithmic scale is calculating the logarithm of the sum of exponentials, short logsumexp.

$$\text{logsumexp}(\mathbf{x}) := \log \sum_i \exp x_i \quad (7.28)$$

This operation arises when calculating the logarithm of sum of numbers y_i in linear scale $\log \sum_i y_i$, where y_i is represented in logarithmic scale ($y_i := \exp(x_i)$) and requires leaving the logarithmic scale ($\exp(x_i)$) to calculate the sum and thereby risking floating-point overflows.

In order to avoid overflows, the largest exponent, often a good approximation of total sum, is pulled out of the sum:

$$\log \sum_i \exp x_i = x_{max} + \log \sum_i \exp(x_i - x_{max}) \quad (7.29)$$

While this formulation still bridges logarithmic and linear scales, this calculation is not affected by numerical overflows as $\exp(x_j - x_{max})$ never surpasses 1. When $x_j - x_{max}$ is a large negative number, the exponentiation will yield 0 in linear scale. This is however not a problem as the number $y_j := \exp(x_j)$ is so small compared to $y_{max} := \exp(x_{max})$ that its contribution to the total sum $\sum_i y_i$ is negligibly small.

While logarithms are only defined for positive numbers, the logsumexp trick can be used to represent both positive and negative numbers by storing the sign separately from the logarithm of the absolute value $y_i := \text{sgn } y_i \times \exp(x_i)$, where $x_i := \log(|y_i|)$. We can then calculate z and $\text{sgn } z$ so that $\text{sgn } z \times \exp(z) = \sum_i \text{sgn } y_i \times \exp(x_i)$ as follows:

$$z = x_{max} + \log \left| \sum_i \text{sgn } y_i \exp(x_i - x_{max}) \right| \quad (7.30)$$

$$\text{sgn } z = \text{sgn} \left(\sum_i \text{sgn } y_i \exp(x_i - x_{max}) \right) \quad (7.31)$$

We will make use of the signed logsumexp trick when calculating derivatives in log scale which unlike probabilities can take any value in the real number space.

The likelihood in logarithmic scale With the logsumexp trick in hand, we can write the equations 7.7–7.12 in logarithmic scale:

$$\log p(\mathbf{X}_0) = \log \sum_{a,b=1}^{20} \exp [\log p(\mathbf{X}_0|a, b) + \log p(a, b)] \quad (7.32)$$

$$\log p(\mathbf{X}_l|a, b) = \log p(\mathbf{X}_m|a, b; r_{lm}) + \log p(\mathbf{X}_n|a, b; r_{ln}) \quad (7.33)$$

$$\begin{aligned} \log p(\mathbf{X}_m|a, b; r) := & \\ & \exp [2 \log(1-r) + \log p(\mathbf{X}_m)] \\ & + \exp [\log r + \log(1-r) + \log (\exp (\log p(\mathbf{X}_m|\cdot, b)) + \exp (\log p(\mathbf{X}_m|a, \cdot)))] \\ & + \exp [2 \log r + \log p(\mathbf{X}_m|a, b)] \end{aligned} \quad (7.34)$$

$$\log p(\mathbf{X}_m) := \log \sum_{c,d=1}^{20} \exp [\log p(\mathbf{X}_m|c, d) + \log p(c, d)] \quad (7.35)$$

$$\log p(\mathbf{X}_m|\cdot, b) := \log \sum_{c=1}^{20} \exp [\log p(\mathbf{X}_m|c, b) + \log p(c, b|\cdot, b)] \quad (7.36)$$

$$\log p(\mathbf{X}_m|a, \cdot) := \log \sum_{d=1}^{20} \exp [\log p(\mathbf{X}_m|a, d) + \log p(a, d|a, \cdot)] \quad (7.37)$$

and finally the equations of the substitution model 7.1–7.3:

$$\log p(c, d) = v_i(c) + v_j(d) + w_{ij}(c, d) - \log \sum_{c',d'} \exp (v_i(c') + v_j(d') + w_{ij}(c', d')) \quad (7.38)$$

$$\log p(c, d|\cdot, d) = v_i(c) + w_{ij}(c, d) - \log \sum_{c'} \exp (v_i(c') + w_{ij}(c', d)) \quad (7.39)$$

$$\log p(c, d|c, \cdot) = v_j(d) + w_{ij}(c, d) - \log \sum_{d'} \exp (v_j(d') + w_{ij}(c, d')) \quad (7.40)$$

The logarithm of the likelihood can be computed with the algorithm introduced in Section 7.1.2 by computing and storing the logarithmic equivalents calculated in Equations 7.32–7.40. In order ensure numerical stability, sums of exponentials are computed with the logsumexp trick. For illustration, the pseudo-code for the numerically stable calculation of 7.32 and 7.34 are presented below.

Input: $\log p(\mathbf{X}_0|a, b)$, $\log p(a, b)$, \mathcal{A}

Result: $\log p(\mathbf{X}_0)$

$A \leftarrow |\mathcal{A}|$

$q \leftarrow$ new array of size A^2

for $a \in \mathcal{A}$ **do**

for $b \in \mathcal{A}$ **do**

$q[a \times A + b] \leftarrow p(\mathbf{X}_0|a, b) + \log p(a, b)$

end

end

$\log p(\mathbf{X}_0) \leftarrow \text{logsumexp}(q)$

Algorithm 1: Numerically stable implementation of Equation 7.32

Input: $\log p(\mathbf{X}_m)$, $\log p(\mathbf{X}_m|\cdot, b)$, $\log p(\mathbf{X}_m|a, \cdot)$, $\log p(\mathbf{X}_m|a, b)$, $\log r$, $\log(1 - r)$, \mathcal{A}

Result: $\log p(\mathbf{X}_m|a, b; r)$

for $a \in \mathcal{A}$ **do**

for $b \in \mathcal{A}$ **do**

$q_1 \leftarrow 2 \log(1 - r) + \log p(\mathbf{X}_m)$

$q_2 \leftarrow \text{logsumexp}(\log p(\mathbf{X}_m|\cdot, b), \log p(\mathbf{X}_m|a, \cdot))$

$q_3 \leftarrow 2 \log(r) + \log p(\mathbf{X}_m|a, b)$

$\log p(\mathbf{X}_m|a, b; r) \leftarrow \text{logsumexp}(q_1, q_2, q_3)$

end

end

Algorithm 2: Numerically stable implementation of Equation 7.34

The derivative in logarithmic scale In the previous section I showed how to calculate the likelihood $\log p(\mathbf{X}_0)$ in logarithmic scale. In order to apply the numerical optimization scheme introduced in Section 7.1.4 we additionally have to obtain the partial derivatives of the log likelihood $\frac{\partial \log p(\mathbf{X}_0)}{\partial \theta}$ with respect to the model parameters \mathbf{v}_i , \mathbf{v}_j and \mathbf{w}_{ij} . The chain rule of calculus can be used to express the derivative of the log likelihood as the quotient of derivative of the likelihood and its derivative: $\frac{\partial \log p(\mathbf{X}_0)}{\partial \theta} = \frac{\partial p(\mathbf{X}_0)}{\partial \theta} / p(\mathbf{X}_0)$, or in logarithmic scale:

$$\frac{\partial \log p(\mathbf{X}_0)}{\partial \theta} = \text{sgn} \left(\frac{\partial p(\mathbf{X}_0)}{\partial \theta} \right) \exp \left(\log \left| \frac{\partial p(\mathbf{X}_0)}{\partial \theta} \right| - \log p(\mathbf{X}_0) \right) \quad (7.41)$$

Note that in contrast to probabilities, derivatives can take negative values and thus the sign of the derivative needs to be taken into account.

In the previous section we derived a numerically stable computation of the log likelihood $\log p(\mathbf{X}_0)$. The derivatives 7.13-7.27 can similarly be transformed into logarithmic scale by storing the log of the absolute and the sign separately.

The log absolute of Equation 7.13 becomes

$$\begin{aligned} \log \left| \frac{\partial p(\mathbf{X}_0)}{\partial \theta} \right| &= \log \sum_{a,b=1}^{20} \left| \text{sgn} \left(\frac{\partial p(\mathbf{X}_0|a, b)}{\partial \theta} \right) \exp \left[\log \left| \frac{\partial p(\mathbf{X}_0|a, b)}{\partial \theta} \right| + \log p(a, b) \right] \right. \\ &\quad \left. + \text{sgn} \left(\frac{\partial p(a, b)}{\partial \theta} \right) \exp \left[\log p(\mathbf{X}_0|a, b) + \log \left| \frac{\partial p(a, b)}{\partial \theta} \right| \right] \right| \end{aligned} \quad (7.42)$$

and the log absolute of Equation 7.15 is

$$\begin{aligned} \log \left| \frac{\partial p(\mathbf{X}_l|a, b)}{\partial \theta} \right| &= \left| \operatorname{sgn} \left(\frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial \theta} \right) \exp \left[\log \left| \frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial \theta} \right| + \log p(\mathbf{X}_n|a, b; r_{ln}) \right] \right. \\ &\quad \left. + \operatorname{sgn} \left(\frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial \theta} \right) \exp \left[\log p(\mathbf{X}_m|a, b; r_{lm}) + \log \left| \frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial \theta} \right| \right] \right| \end{aligned} \quad (7.43)$$

Input: $\log p(\mathbf{X}_0|a, b)$, $\log \left| \frac{\partial p(\mathbf{X}_0|a, b)}{\partial w_{cd}} \right|$, $\operatorname{sgn} \left(\frac{\partial p(\mathbf{X}_0|a, b)}{\partial w_{cd}} \right)$,
 $\log p(a, b)$, $\log \left| \frac{\partial p(a, b)}{\partial w_{cd}} \right|$, $\operatorname{sgn} \left(\frac{\partial p(a, b)}{\partial w_{cd}} \right)$,

Result: $\log \left| \frac{\partial p(\mathbf{X}_0)}{\partial w_{cd}} \right|$, $\operatorname{sgn} \left(\frac{\partial p(\mathbf{X}_0)}{\partial w_{cd}} \right)$

$A \leftarrow |\mathcal{A}|$

$q \leftarrow$ new array of size $2 \times A^2$

$s \leftarrow$ new array of size $2 \times A^2$

for $c \in \mathcal{A}$ **do**

for $d \in \mathcal{A}$ **do**

for $a \in \mathcal{A}$ **do**

for $b \in \mathcal{A}$ **do**

$i \leftarrow 2(a \times A + b)$

$q[i] \leftarrow \log \left| \frac{\partial p(\mathbf{X}_0|a, b)}{\partial w_{cd}} \right| + \log p(a, b)$

$s[i] \leftarrow \operatorname{sgn} \frac{\partial p(\mathbf{X}_0|a, b)}{\partial w_{cd}}$

$q[i + 1] \leftarrow \log p(\mathbf{X}_0|a, b) + \log \left| \frac{\partial p(a, b)}{\partial w_{cd}} \right|$

$s[i + 1] \leftarrow \operatorname{sgn} \frac{\partial p(a, b)}{\partial w_{cd}}$

end

end

$\log \left| \frac{\partial p(\mathbf{X}_0)}{\partial w_{cd}} \right|, \operatorname{sgn} \left(\frac{\partial p(\mathbf{X}_0)}{\partial w_{cd}} \right) \leftarrow \text{signed_logsumexp}(q, s)$

end

end

Algorithm 3: Example of a numerically stable implementation of the 400 derivatives $\frac{\partial p(\mathbf{X}_0)}{\partial w_{cd}}$ in logarithmic scale (Equation 7.42).

The remaining derivatives are transformed similarly by separating sign from log absolute and then applying the logarithm to Equations 7.16–7.27. We define the logarithm of the logarithm of the Kronecker Delta as

$$\log \delta_{ab} = \begin{cases} 0 & \text{if } a = b \\ \log_0 & \text{if } a \neq b \end{cases} \quad (7.44)$$

where we define \log_0 as a large negative number.

As with the log likelihood we can use the algorithm for calculating the derivatives in the linear space for calculating the derivatives of the log likelihood by storing all intermediate results in logarithmic representation. Unlike the likelihood that is always positive, for the derivatives we have to store the signs separately. With numerically stable calculation for the log likelihood $\log p(\mathbf{X}_0)$ and its derivatives

Input: $\log p(\mathbf{X}_m|a, b; r_{lm}), \log \left| \frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial w_{cd}} \right|, \text{sgn} \left(\frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial w_{cd}} \right),$
 $\log p(\mathbf{X}_n|a, b; r_{ln}), \log \left| \frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial w_{cd}} \right|, \text{sgn} \left(\frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial w_{cd}} \right),$
 \mathcal{A}
Result: $\log \left| \frac{\partial p(\mathbf{X}_l|a, b)}{\partial w_{cd}} \right|, \text{sgn} \left(\frac{\partial p(\mathbf{X}_l|a, b)}{\partial w_{cd}} \right)$
for $c \in \mathcal{A}$ **do**
 for $d \in \mathcal{A}$ **do**
 for $a \in \mathcal{A}$ **do**
 for $b \in \mathcal{A}$ **do**
 $q_m \leftarrow \log \left| \frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial w_{cd}} \right| + \log p(\mathbf{X}_n|a, b; r_{ln})$
 $s_m \leftarrow \text{sgn} \left(\frac{\partial p(\mathbf{X}_m|a, b; r_{lm})}{\partial w_{cd}} \right)$
 $q_n \leftarrow \log p(\mathbf{X}_m|a, b; r_{lm}) + \log \left| \frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial w_{cd}} \right|$
 $s_n \leftarrow \text{sgn} \left(\frac{\partial p(\mathbf{X}_n|a, b; r_{ln})}{\partial w_{cd}} \right)$
 $\log \left| \frac{\partial p(\mathbf{X}_l|a, b)}{\partial w_{cd}} \right|, \text{sgn} \left(\frac{\partial p(\mathbf{X}_l|a, b)}{\partial w_{cd}} \right) \leftarrow \text{signed_logsumexp}(q_m, s_m, q_n, s_n)$
 end
 end
 end
end

end

Algorithm 4: Example of a numerically stable implementation of the 400×400 derivatives $\frac{\partial p(\mathbf{X}_l|a, b)}{\partial w_{cd}}$ in logarithmic scale (Equation 7.43).

$\frac{\partial \log p(\mathbf{X}_0)}{\partial v_{ia}}, \frac{\partial \log p(\mathbf{X}_0)}{\partial v_{jb}}, \frac{\partial \log p(\mathbf{X}_0)}{\partial w_{ijab}}$ in hand, we can use gradient-based optimization routines to obtain $\mathbf{v}_i^*, \mathbf{v}_j^*, \mathbf{w}_{ij}^*$ that optimize the likelihood.

Reducing alphabet size

The alphabet size of canonical amino acids is $|\mathcal{A}| = 20$. The number of gradient computations at each node is in the order of $\mathcal{O}(|\mathcal{A}|^4)$ and thus is the bottleneck for realistic alignment sizes. However not all 20 amino acids are typically present in every alignment column. This can be due to limited evolutionary time passed so that not all viable amino acids have been observed and evolutionary constraints imposed by selection pressures that strongly favor a limited set of possible combinations. If a column i does not contain the amino acid a , the corresponding v_{ia} will tend towards $-\infty$, and thus $p(x_i = a, x_j = d) = 0$ for all d (see Equation 7.1). As the likelihood is a sum weighted by the pairwise probability (see Equation 7.7), all pairs with amino acids that do not occur in either of the columns do not contribute to the final sum.

This observation allows a drastic speedup for columns that have not observed all 20 possible amino acids: if only A_i and A_j amino acids have been observed at least once in columns i and j , respectively, only $A_i \times A_j$ terms in the summation of Equation 7.7 are non-zero and thus have to be computed. One can thus introduce reduced alphabets $\mathcal{A}_i = \{a \in \mathcal{A} : n_{ia} > 0\}$ and $\mathcal{A}_j = \{b \in \mathcal{A} : n_{jb} > 0\}$, where n_{ia} is defined as the total number of amino acid a is observed in column i . This reduces the order of required computations per node from $\mathcal{O}(|\mathcal{A}|^4)$ to $\mathcal{O}(|\mathcal{A}_i|^2 |\mathcal{A}_j|^2)$, a reduction by factor 16 if in each column only 10 out of the 20 possible amino acids are observed.

Polynomial approximations to log2 and exp2

The likelihood optimization in logarithmic scale requires a large number of logarithm and exponentiation operations when bridging logarithmic and linear scales in the logsumexp trick. Unlike addition and multiplication, which is readily available by specialized hardware units on the CPU, logarithms and exponents are complex operations that are composed of a large number of simple instructions. By performing the logsumexp trick with base 2 operations $\log_2(x)$ and 2^x and using polynomial approximations $p_{\log}(x; \boldsymbol{\theta}_{\log}) \approx \log_2(x)$ and $p_{\exp}(x; \boldsymbol{\theta}_{\exp}) \approx 2^x$ can increase the speed by trading-off accuracy. The polynomial approximation can be expressed by series of simple CPU operations, a benefit that will allow us to achieve further speed-up by using SIMD instructions in section 7.1.5.

Approximating 2^x Floating point numbers x are internally stored as mantissa and exponent in following form: $x = 1.\text{mmm} \dots \text{m} \times 2^e$, where $\text{mmm} \dots \text{m}$ and e are integers storing the precision and the magnitude, respectively. We can represent $y := 2^x$ as $y = 1.\text{aaa} \dots \text{a} \times 2^b$, by splitting x into the two parts $r := \lfloor x \rfloor$ and $q := x - r$:

$$\begin{aligned}
 y &:= 2^x \\
 &= 2^{q+r} \\
 &= 2^q \times 2^r \\
 &= 1.\text{aaa} \dots \text{a} \times 2^r
 \end{aligned}
 \tag{7.45}$$

As by construction $0 \leq q < 1$, it follows that $1 \leq 2^q < 2$ and therefore the equation yields y in its floating number representation. We approximate the calculation 2^q , required by the mantissa with a polynomial: $2^q \approx p_{\exp}(x, \boldsymbol{\theta})$. The parameters $\boldsymbol{\theta}$ are chosen such that

$$p_{\exp}(0, \boldsymbol{\theta}) = 1 \tag{7.46}$$

$$p_{\exp}(1, \boldsymbol{\theta}) = 2 \tag{7.47}$$

$$\|p_{\exp}(x, \boldsymbol{\theta}) - 2^x\|^2 \rightarrow \min, \text{ for } x \in [0, 1[\tag{7.48}$$

The accuracy of the approximation is determined by the order of the polynomial $p_{\exp}(x)$. The coefficients of a 6th order polynomial $\boldsymbol{\theta}$ that satisfy the three conditions 7.46 to 7.48 with a maximum deviation of $\|p_{\exp}(x, \boldsymbol{\theta}) - 2^x\|_{\infty} < 4.2 \times 10^{-9}$ up to double precision accuracy:

$$\begin{aligned}
\theta_1 &= 0.0002187767014305746279 \\
\theta_2 &= 0.001238881395488288005 \\
\theta_3 &= 0.009684327747431309072 \\
\theta_4 &= 0.05548068064239377456 \\
\theta_5 &= 0.2402303737183841825 \\
\theta_6 &= 0.693146959794871842 \\
\theta_7 &= 1.0
\end{aligned} \tag{7.49}$$

The polynomial is evaluated by successive multiplications and additions and thus requires only fast, efficient hardware-provided instructions:

$$p_{\text{exp}}(x, \boldsymbol{\theta}) = (((((\theta_1 x + \theta_2) \times x + \theta_3) \times x + \theta_4) \times x + \theta_5) \times x + \theta_6) \times x + \theta_7 \tag{7.50}$$

Combining the calculation of the mantissa 2^q , the exponent r with detection of numerical overflow (setting $y := \infty$) and underflow (setting $y := 0$) yields the efficient approximation of 2^x .

Approximating $\log_2(x)$ Again starting from x in floating point representation $x = 1.\text{mmm} \dots \text{m} \times 2^e$, we obtain:

$$\begin{aligned}
y &:= \log_2(x) \\
&= \log_2(1.\text{mmm} \dots \text{m}) + e
\end{aligned} \tag{7.51}$$

Here we approximate the mantissa $\log_2(1+x) \approx p_{\text{log}}(x, \boldsymbol{\theta})$ that satisfies following conditions:

$$p_{\text{log}}(0, \boldsymbol{\theta}) = 0 \tag{7.52}$$

$$p_{\text{log}}(1, \boldsymbol{\theta}) = 1 \tag{7.53}$$

$$\|p_{\text{log}}(x, \boldsymbol{\theta}) - \log_2(x+1)\|^2 \rightarrow \min, \text{ for } x \in [0, 1[\tag{7.54}$$

The 9th-order polynomial approximation that satisfied the criteria 7.52–7.54 has a maximum deviation $\|p_{\text{log}}(x, \boldsymbol{\theta}) - \log_2(x+1)\|_{\infty} < 1.3 \times 10^{-8}$.

$$\begin{aligned}
\theta_1 &= 0.00539574483271335 \\
\theta_2 &= -0.033134075405641866 \\
\theta_3 &= 0.09571929135783046 \\
\theta_4 &= -0.18043327446159182 \\
\theta_5 &= 0.26625227022774905 \\
\theta_6 &= -0.3553426744739997 \\
\theta_7 &= 0.4801415033950581 \\
\theta_8 &= -0.7212923532638644 \\
\theta_9 &= 1.4426935677917467 \\
\theta_{10} &= 0
\end{aligned}
\tag{7.55}$$

For valid inputs, the logarithm operation never overflows nor underflows and thus adding the exponent to the polynomial evaluation on the mantissa subtracted by one yields the approximation of $\log_2(x)$.

Parallelization

Modern CPU architectures are designed for efficient parallel processing. This is achieved on the one hand by using multiple widely independent processing units (cores), but on the other hand by each core having access to independent, specialized computation units that can perform their computations in parallel in one time unit (clock cycle). With more and more computation units available to each core, one of the key challenges of modern CPUs is to break the code into enough independent instructions to maintain high computation unit utilisation in each clock cycle. Code that is designed to use the CPU's capabilities of parallel computing efficiently thus can run orders of magnitudes faster than unoptimized code. In the following we

Core level parallelization Our algorithm consists of $\binom{L}{2}$ independent optimization tasks. For realistic $L > 100$, the number of tasks is much larger than the number of cores of modern CPUs, a fortunate situation termed *embarrassingly parallel*. By distributing the optimization of the column pairs across the available processors we can achieve optimal core with no further optimizations required.

Unit level parallelization In the computation heavy parts, our algorithm manipulates floating point numbers. For parallel floating point operations CPUs offer specialized long SIMD registers that fit multiple floating point numbers and a limited set of instructions applied in parallel to all floating point numbers in the register. The size of the registers are currently between 128bit and 512bit in size, therefore fitting 4 to 16 single precision or 2 to 8 double precision floating point numbers. The size and the available instruction set of the registers are continuously growing with newer CPU generations.

As an example consider the task of performing the operation $z[i] = x[i] + y[i]$ for all $i \in \{0, \dots, N\}$.

While a simple implementation in the C language may look like this:

```
void add_array(double* z, double* x, double* y, size_t N) {
```

```

for (int n = 0; n < N; n++) {
    z[n] = x[n] + y[n];
}

```

an explicitly vectorized implementation will take following form:

```

void add_array(double* z, double* x, double* y, size_t N) {
    for (int n = 0; n < N; n+=VECSIZE_DOUBLE) {
        simd64 x_chunk = simd64_load(x + n);
        simd64 y_chunk = simd64_load(y + n);
        simd64_store(z + n, simd64_add(x_chunk, y_chunk));
    }
}

```

Instead of processing each number at a time as the simple version, the vectorized version loads blocks of `VECSIZE_DOUBLE` double numbers into the SIMD registers and performs the `VECSIZE_DOUBLE` sum operations for the whole block in parallel. To harness the full potential of the vectorized version two conditions have to be met: (1) the number of computed blocks `N` has to be sufficiently large and (2) `x`, `y` and `z` have to be stored contiguously and linearly in memory, so that the load and store operations can make optimal use of the CPU's internal caching optimizations.

While modern compilers will try to perform vectorization automatically, for performance critical code it is often necessary to transform the code and vectorize explicitly, as it may take conscious reorganising of the memory layout in order to achieve highly efficient code. While the effort can be significant so are the potential gains: On top of the potential speedup by cache alignment, perfectly vectorized code can gain up to a factor of 16 times and 8 in speed on modern CPU architectures for single and double precision respectively.

Our fastest implementation relies on vectorized implementations of the logarithmic scale versions of Equations 7.14–7.18.

7.1.6. Calculating phylogenetically corrected pair counts

For each column pair (i, j) the Felsenstein-like algorithm developed in the previous sections calculates the single column parameters \tilde{v}_{ia} and pair parameters \tilde{w}_{ijab} under the assumption of the independent-pair model. Our goal is to use the pairwise parameters to calculate phylogenetically corrected pair counts n_{ijab} . As sufficient statistics of the MRF model, the corrected pair counts can be used to calculate phylogenetically corrected MRF parameters with the PCD algorithm.

By deriving the objective function of the MRF optimization and setting the derivative to zero, we can obtain a closed form expression of the pair counts n_{ijab} .

$$\begin{aligned}
 n_{ijab} - N_{ij} p(x_i = a, x_j = b | \mathbf{v}_i, \mathbf{v}_j, \mathbf{w}_{ij}) - \lambda \tilde{w}_{ijab} &= 0 \\
 n_{ijab} = N_{ij} \frac{\exp(\tilde{v}_{ia} + \tilde{v}_{jb} + \tilde{w}_{ijab})}{\sum_{a', b'=1}^{20} \exp(\tilde{v}_{ia'} + \tilde{v}_{jb'} + \tilde{w}_{ija'b'})} + \lambda \tilde{w}_{ijab} & \quad (7.56)
 \end{aligned}$$

In Equation 7.56 λ is a user-defined parameter and \tilde{v}_{ia} and \tilde{w}_{ijab} are known from the Felsenstein-like

optimization. In order to calculate the pair counts n_{ijab} , we need to know N_{ij} , the total number of independent pair counts for the column pair (i, j) . For independent sequences N_{ij} is the total number of sequences with no gaps in both column i and j and can be counted from the alignment. For dependent sequences N_{ij} has to be estimated.

One approach to estimate the N_{ij} is by quantifying how strongly the coupling parameters \mathbf{w} are shrunk by the prior in the pairwise optimization of the Felsenstein-like algorithm: having determined optimal pairwise parameters $(\tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_{ij})$ and $(\tilde{\mathbf{v}}'_i, \tilde{\mathbf{v}}'_j, \tilde{\mathbf{w}}'_{ij})$ for two different regularization strengths λ and λ' , following equations hold due to the vanishing gradient in the optimum:

$$\begin{aligned} n_{ijab} - N_{ij} p(x_i = a, x_j = b | \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_{ij}, \lambda) - \lambda \tilde{w}_{ijab} &= 0 \\ n_{ijab} - N_{ij} p(x_i = a, x_j = b | \tilde{\mathbf{v}}'_i, \tilde{\mathbf{v}}'_j, \tilde{\mathbf{w}}'_{ij}, \lambda') - \lambda' \tilde{w}'_{ijab} &= 0 \end{aligned} \quad (7.57)$$

The two equations can be solved for N_{ij}^2 .

$$\begin{aligned} N_{ij} (p(x_i = a, x_j = b | \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_{ij}, \lambda) - p(x_i = a, x_j = b | \tilde{\mathbf{v}}'_i, \tilde{\mathbf{v}}'_j, \tilde{\mathbf{w}}'_{ij}, \lambda')) &= -\lambda \tilde{w}_{ijab} + \lambda' \tilde{w}'_{ijab} \\ N_{ij}^2 \sum_{a,b=1}^{20} (p(x_i = a, x_j = b | \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_{ij}, \lambda) - p(x_i = a, x_j = b | \tilde{\mathbf{v}}'_i, \tilde{\mathbf{v}}'_j, \tilde{\mathbf{w}}'_{ij}, \lambda'))^2 &= \sum_{a,b=1}^{20} (\lambda \tilde{w}_{ijab} - \lambda' \tilde{w}'_{ijab})^2 \\ N_{ij}^2 &= \frac{\sum_{a,b=1}^{20} (\lambda \tilde{w}_{ijab} - \lambda' \tilde{w}'_{ijab})^2}{\sum_{a,b=1}^{20} (p(x_i = a, x_j = b | \tilde{\mathbf{v}}_i, \tilde{\mathbf{v}}_j, \tilde{\mathbf{w}}_{ij}, \lambda) - p(x_i = a, x_j = b | \tilde{\mathbf{v}}'_i, \tilde{\mathbf{v}}'_j, \tilde{\mathbf{w}}'_{ij}, \lambda'))^2} \end{aligned} \quad (7.58)$$

Together with λ , \tilde{v}_{ia} and \tilde{w}_{ijab} the n_{ijab} can be calculated using Equation 7.56.

7.1.7. Deriving the tree with a family-specific model

In the course of evolution the substitution rates of amino acids depend on their biochemical properties. Typically, phylogenetic tree reconstruction is using independent-site models with empirically derived universal substitution matrices.

By describing each protein family with an MRF, we use family-specific models that can take pairwise site-interactions into account. Neglecting the site interactions, we can approximate t , the number of mutations that occurred between two sequences \mathbf{x} and \mathbf{y} , and thereby derive protein family-specific evolutionary distances:

$$p(\mathbf{y}|\mathbf{x}, t, \mathbf{v}) = \prod_{i=1}^L p(y_i|x_i, t, \mathbf{v}) \quad (7.59)$$

Assuming a Poisson model, a site is unchanged with a probability of $r = e^{-t}$, and therefore:

$$p(y_i|x_i, t, \mathbf{v}) = r \delta_{x_i, y_i} + (1 - r) p(y_i|\mathbf{v}_i) \quad (7.60)$$

Using the notation $p_i := p(Y_i = y_i|\mathbf{v}_i) = e^{v_i(y_i)} / \sum_a e^{v_i(a)}$ and the exponential prior $p(t|\tau) = e^{-t/\tau} / \tau$, the log posterior can be written as:

$$\log p(t|\mathbf{x}, \mathbf{y}, \mathbf{v}) = \sum_{i=1}^L \log [r \delta_{x_i, y_i} + (1-r) p_i] - \frac{t}{\tau} + \text{const} \quad (7.61)$$

The scalar t can be efficiently optimized with by finding the root of the first derivative with Newton-Raphson iterations: $t \leftarrow t - \alpha \frac{f'(t)}{f''(t)}$ (with $\alpha < 1$), where

$$\begin{aligned} f'(t) &= \frac{d}{dt} \log p(t|\mathbf{x}, \mathbf{y}, \mathbf{v}) \\ &= \frac{dr}{dt} \sum_{i=1}^L \frac{\delta_{x_i, y_i} - p(y_i|\mathbf{v}_i)}{r \delta_{x_i, y_i} + (1-r) p_i} - \frac{1}{\tau} \\ &= -r \left(\sum_{i:x_i=y_i}^L \frac{1-p_i}{r+(1-r)p_i} - \sum_{i:x_i \neq y_i}^L \frac{p_i}{(1-r)p_i} \right) - \frac{1}{\tau} \\ &= -\frac{r}{1-r} \left(\sum_{i:x_i=y_i}^L \frac{(1-p_i)(1-r)}{r+(1-r)p_i} - L + \sum_{i:x_i=y_i}^L 1 \right) - \frac{1}{\tau} \\ &= -\frac{r}{1-r} \left(\sum_{i:x_i=y_i}^L \frac{1}{p_i+r(1-p_i)} - L \right) - \frac{1}{\tau} \end{aligned} \quad (7.62)$$

and

$$\begin{aligned} f''(t) &= \frac{d^2}{dt^2} \log p(t|\mathbf{x}, \mathbf{y}, \mathbf{v}) \\ &= \frac{r}{(1-r)^2} \sum_{i:x_i=y_i}^L \frac{1}{r+p_i-rp_i} - \frac{r}{1-r} \sum_{i:x_i=y_i}^L \frac{r(1-p_i)}{(r+p_i-rp_i)^2} - \frac{r}{(1-r)^2} L \\ &= \frac{r}{(1-r)^2} \left(\sum_{i:x_i=y_i}^L \frac{r+p_i-rp_i-r(1-r)(1-p_i)}{(r+p_i-rp_i)^2} - L \right) \\ &= \frac{r}{(1-r)^2} \left(\sum_{i:x_i=y_i}^L \frac{p_i+r^2(1-p_i)}{(p_i+r(1-p_i))^2} - L \right) \end{aligned} \quad (7.64)$$

We estimate the column parameters \mathbf{v} empirically from the MSA by assuming independent sequences. From the calculated pairwise evolutionary distances t a phylogenetic tree can be calculated with a standard neighbor-joining algorithm.

8. Results

8.1. Validation on simulated data

Our first objective is to validate the FS-PCD’s ability to correct out phylogenetic dependencies in well-controlled simulations in three steps: (i) simulate \mathbf{v}^* , \mathbf{w}^* and the evolutionary tree \mathcal{T} ; (ii) use \mathbf{v}^* , \mathbf{w}^* , and \mathcal{T} with CCMgen (Vorberg et al., 2018) to simulate sequence alignments; (iii) learn back the parameters $\hat{\mathbf{v}}$, $\hat{\mathbf{w}}$ by applying PCD and our FS-PCD method informed by the underlying evolutionary history.

8.1.1. FS-PCD on independent sequences

The independence assumption of standard DCA methods emerges naturally as a special case of FS-PCD, when a high number of mutations occur between every sequence along the phylogeny. This corresponds to a phylogenetic tree \mathcal{T} with long edge lengths. As sufficient statistics of the MRF, the amino acid pair counts suffice to learn the parameters of the MRF with PCD. For independent sequences the pair counts can be directly counted on the MSA, therefore allowing PCD to train the MRF on MSAs.

In order to validate FS-PCD’s properties on independent sequences, I simulated $N = 2056$ sequences of length $L = 5$ using an artificial phylogeny with long branches. The short alignment size allows to calculate the sequence probabilities $p(\mathbf{x}|\mathbf{v}^*, \mathbf{w}^*)$ analytically. On these sequences, our Felsenstein-like algorithm accurately recovers the pair counts \mathbf{n} from the \mathbf{v} and \mathbf{w} parameters obtained by the $\binom{L}{2}$ Felsenstein optimizations (Figure 8.1). Thereby, FS-PCD falls back to PCD and the learnt parameters $\hat{\mathbf{v}}$, $\hat{\mathbf{w}}$ are close to those obtained from the PCD algorithm (Figure 8.2A and Figure 8.2B).

When applied to protein families, the number of parameters of MRFs outnumber the number of sequences by several orders of magnitude, requiring regularization to prevent poor generalization due to overfitting. The commonly applied L2 regularization on the coupling parameters \mathbf{w} acts as a Gaussian prior with mean 0, thereby shrinking the learnt $\hat{\mathbf{w}}$ towards zero. The shrinkage strength increases with the regularization parameter λ and decreases with increasing evidence provided by the number of observed pair counts. FS-PCD can recapture the implanted coupling parameters \mathbf{w} , but the accuracy is limited in the regime of realistic MSA sizes (Figure 8.3A). Due to the evidence-dependent parameter shrinkage, coupling parameters corresponding to few observed pair counts are strongly pushed towards zero. By weighting the points according to their expected pair probability, the Pearson correlation between implanted and recovered coupling parameters measures the performance of FS-PCD more faithfully by de-emphasizing the coupling parameters for which no or only little information is available in the data. Naturally, this impreciseness in the $\hat{\mathbf{w}}$ also affects the accuracy of the estimated sequence probabilities $p(\mathbf{x}|\hat{\mathbf{v}}, \hat{\mathbf{w}})$, with higher deviation for less likely sequences (Figure 8.3B).

As the MRF is trained in a limited data regime, it is instructional to study the interplay of the spread of the implanted \mathbf{w}^* and the regularization strength λ .

For weak contact strengths $w_{ijab}^* \approx 0$ the deviations of the observed pair frequencies from the frequencies

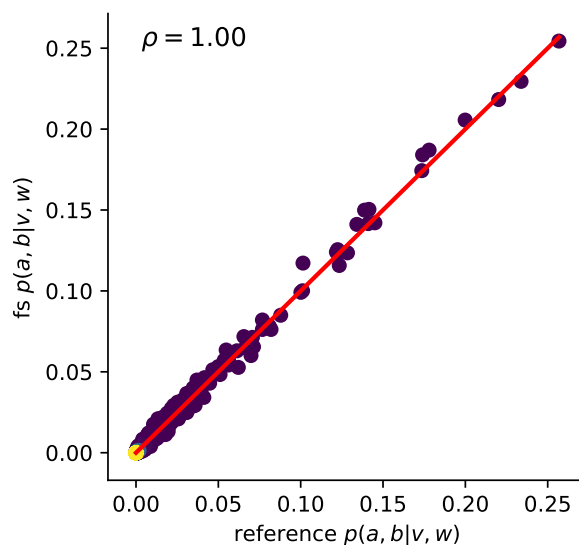


Figure 8.1.: **FS accurately recovers pair counts on independent sequences.** For independent sequences, the pair probabilities of the MRF model can be estimated by counting the amino acid pair frequencies of all column pairs from the MSA. Our Felsenstein-like method can recover the pair counts accurately from the estimated \hat{v} and \hat{w} .

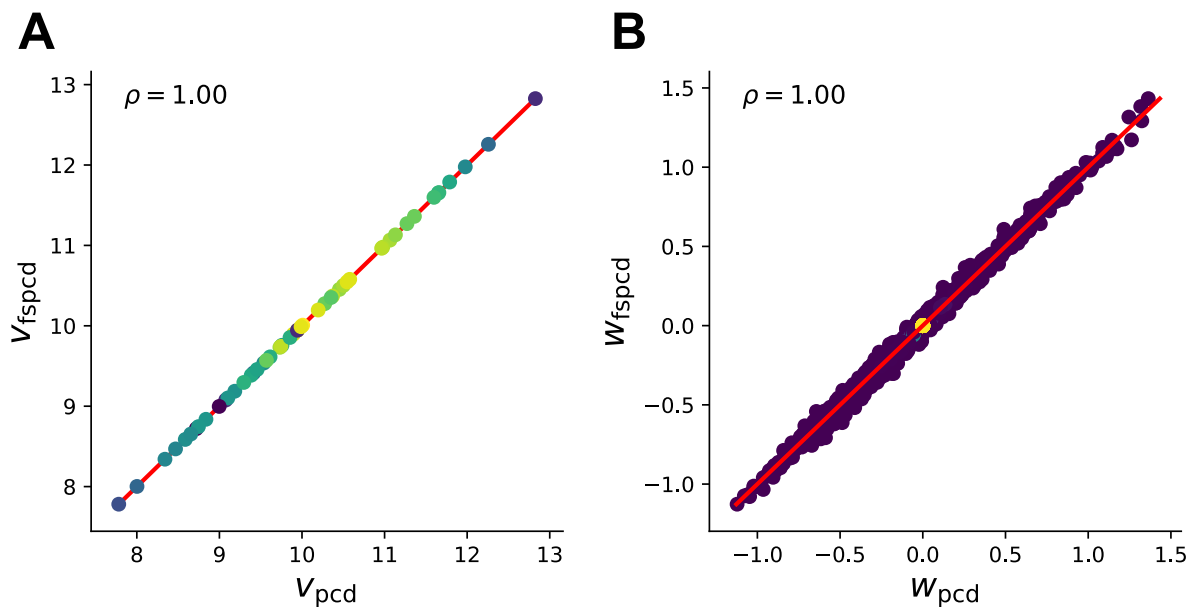


Figure 8.2.: **FS-PCD falls back to PCD for independent sequences.** As the pair counts are the sufficient statistics of the PCD algorithm, FS-PCD falls back to PCD for independent sequences, leading to high correlations between the learnt \hat{v} , \hat{w} .

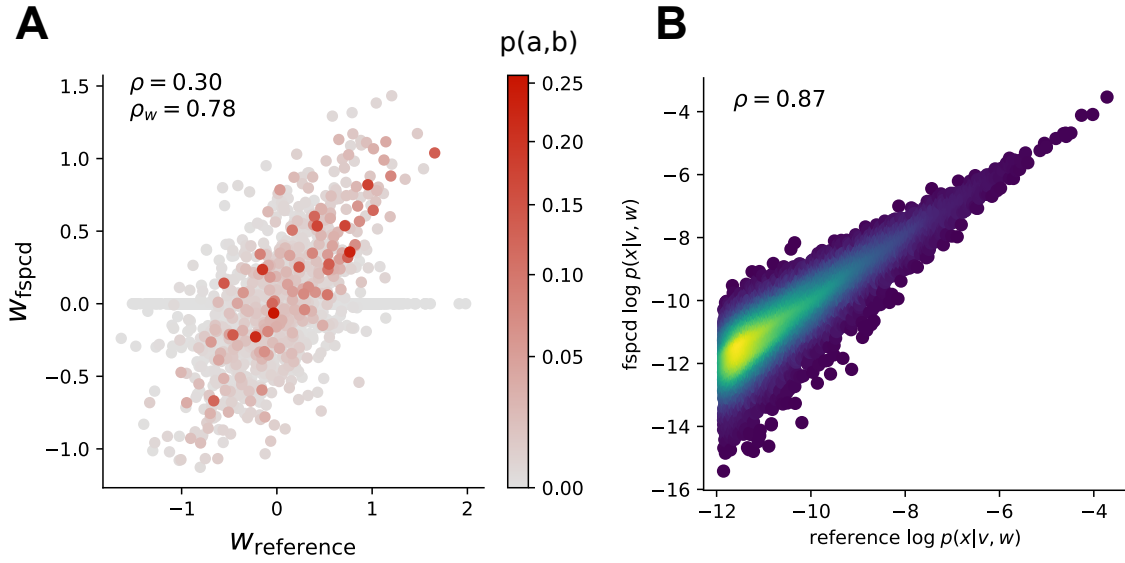


Figure 8.3.: **FS-PCD's parameter estimation is limited by number of sequences.** **A)** Correlation of implanted \mathbf{w}^* and recovered \mathbf{v} colored by their corresponding pair probability. \mathbf{w} associated with higher pair probabilities can be learnt more accurately, leading to higher Pearson correlation when observations are weighted with the pair probability ($\rho_w = 0.78$ vs. $\rho = 0.30$). **B)** Pearson correlation of probability assigned to the 10000 most probable sequences by the implanted parameterization of the MRF with the probabilities assigned by the recovered MRF parameterization.

expected by the amino acid frequencies of the involved columns are small and thus the \mathbf{w}^* cannot be reconstructed accurately (Figure 8.4A). Despite the poor approximation of \mathbf{w}^* , the Pearson correlation between implanted and recaptured sequence probabilities $p(\mathbf{x}|\mathbf{v}, \mathbf{w})$ are highest for small \mathbf{w}^* , especially in combination with strong regularization (Figure 8.4B). In this case the coupling parameters are negligible compared to columns frequency parameters \mathbf{v}^* and setting a high regularization strength λ effectively simplifies the model to $L \times A$ column frequency parameters which can be estimated accurately. With increasing size of \mathbf{w}^* and thus increased impact of the coupling parameters on the sequence probabilities, the optimal regularization shifts towards smaller values. The lowest sequence probability correlations correspond to the largest \mathbf{w}^* for which the limited accuracy of the coupling parameter estimation has a strong influence.

For independent sequences, the reconstruction of the pair counts from the estimated pairwise parameters $\hat{\mathbf{v}}$ and $\hat{\mathbf{w}}$ is accurate irrespective of the underlying coupling strength, the number of sequences and of the choice of the regularization parameter λ (Figure 8.6).

Using the independent-pair approximation, our Felsenstein-like optimization yields $\binom{L}{2} \times A \times A$ pairwise coupling parameters $\hat{\mathbf{w}}_{fs}$ which play similar roles as the coupling parameters in the MRF, except for their obliviousness of the influences of all but the two inspected columns. Just like the correlations of the MRF coupling parameters $\hat{\mathbf{w}}$ inferred by our FS-PCD method, we can compare the pairwise coupling parameters $\hat{\mathbf{w}}_{fs}$ to the simulated ground truth coupling parameters \mathbf{w}^* . The pendants to Figure 8.4A and Figure 8.5A) but correlating \mathbf{w}^* with $\hat{\mathbf{w}}_{fs}$ instead of $\hat{\mathbf{w}}$ show similar overall correlation patterns, but slightly lower correlations ρ_w expected from the inaccuracies due to the violation of the independent-pair assumption: 0.56 vs. 0.58 and 0.60 vs. 0.63 for Figure 8.7A and Figure 8.7B, respectively.

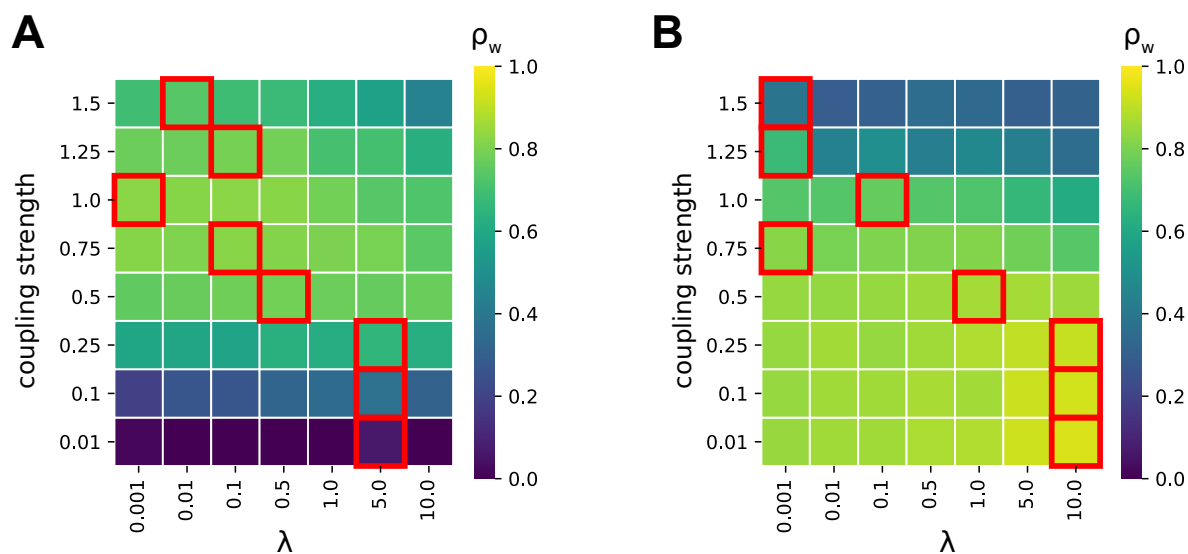


Figure 8.4.: **FS-PCD's parameter estimation at varying coupling and regularization strengths.** **A)** Weighted Pearson correlation between implanted \mathbf{w}^* and recovered $\hat{\mathbf{w}}$. **B)** Pearson correlation of probability assigned to the 10000 most probable sequences by the implanted parameterization of the MRF with the probabilities assigned by the recovered MRF parameterization. The highest correlation of each row is highlighted by a red box.

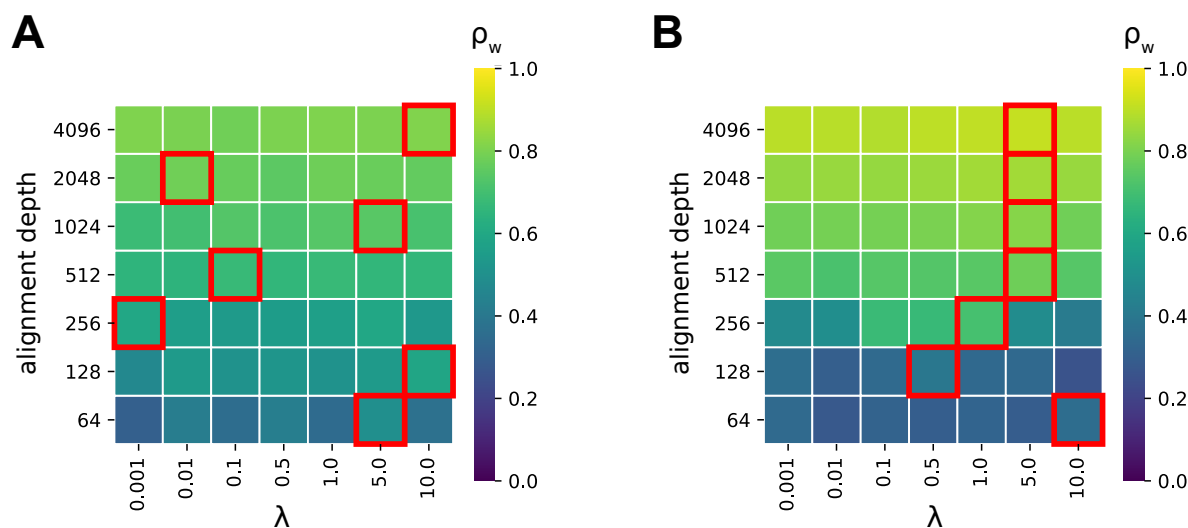


Figure 8.5.: **FS-PCD's parameter estimation for varying number of sequences and regularization strengths.** **A)** Weighted Pearson correlation between implanted \mathbf{w}^* and recovered $\hat{\mathbf{w}}$. **B)** Pearson correlation of probability assigned to the 10000 most probable sequences by the implanted parameterization of the MRF with the probabilities assigned by the recovered MRF parameterization.

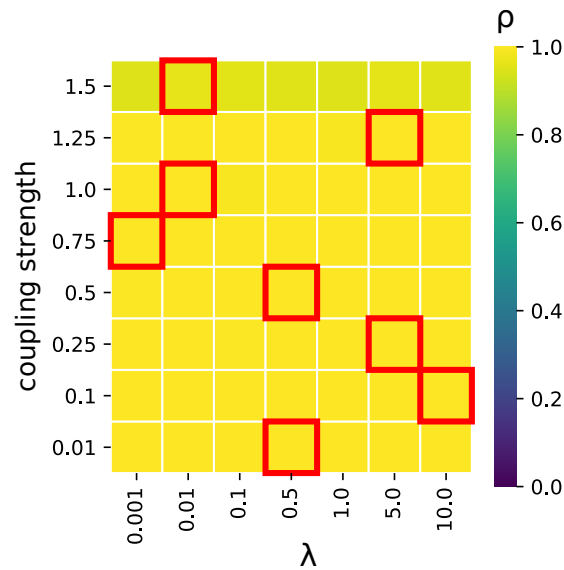


Figure 8.6.: **FS accurately recovers pair counts on independent sequences irrespective of coupling and regularization strength.** The Pearson correlation between the implanted pairwise probabilities $p(a, b|v^*, w^*)$ with the estimation from the Felsenstein-like algorithm. The highest correlation in each row is highlighted by a red box.

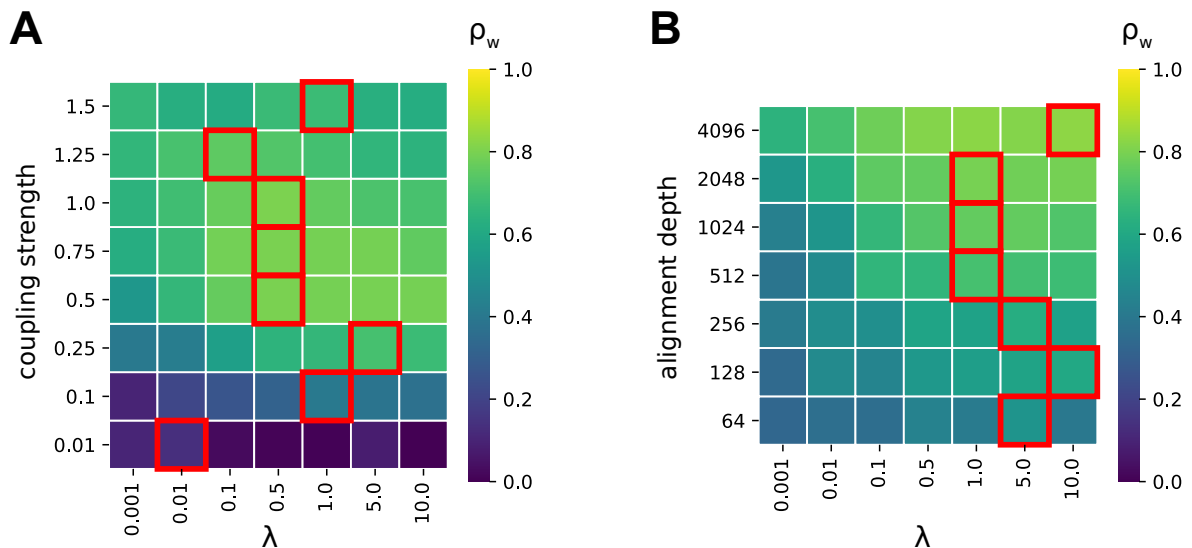


Figure 8.7.: **Column-pair derived coupling parameters approximate MRF parameters.** Weighted Pearson correlation between implanted w^* and recovered \hat{w}_{fs} by our Felsenstein-like algorithm. **A)** Weighted correlations with varying contact strength and regularization strength. **B)** Weighted correlations with varying alignment depth and regularization strength.

The coupling parameters derived under the independent-pair approximation have a slightly lower average correlation than the parameters derived under the MRF model (0.56 vs. 0.58 in case of (A) vs. Figure 8.4A and 0.60 vs. 0.63 in case of (B) vs. Figure 8.5A). The highest correlation in each row is highlighted by a red box.

8.1.2. FS-PCD on dependent sequences

In the previous section I showed that the parameters obtained with FS-PCD are close to the parameters learnt by PCD for independent sequences, thus validating the FS-PCD approach in this special case. In this section I generalize the simulations to phylogenetically dependent sequences. By reducing the number of mutations that occur along the phylogeny, information is shared between leaf nodes through their ancestors, thereby invalidating the independence assumption.

Just like in section 8.1.1, I choose alignments with $L = 5$ columns to allow calculating the sequence and pair probabilities from the simulated ground truth parameters analytically. To partially compensate for the the loss of information due to sequence dependence, I increased the number of simulated sequences fourfold from from $N = 2^{11} = 2048$ to $N = 2^{13} = 8192$. The underlying phylogeny is described by a binary tree averaging m mutations per position from root to leaf. The average number of mutations per position between leaf sequences thus range from $\frac{2m}{13}$ for sequences sharing the same direct ancestor to $2m$ for sequences sharing the root as the lowest common ancestor. In the following I will present and analyze the FS-PCD results averaging one ($m = 1$) and two ($m = 2$) mutations from root to leaf.

In the case of independent sequences the empirical pair counts as sufficient statistics suffice to infer the parameters of the underlying MRF. For dependent sequences, the pair counts are distorted by the dependency structure imposed by the underlying phylogenetic tree. As the corrected pair counts are input to PCD in order to obtain phylogeny-corrected MRF parameters in our FS-DCA approach, the accuracy of the pair count correction is studied in the following:

The corrected pair counts derived by our Felsenstein-like pair-column optimization correlate better with the pair counts expected from independent sequences (32% and 10% increase in mean correlation for $m = 1$ and $m = 2$, respectively, Figure 8.8). This is in line with our expectation that the benefits with increasing sequence dependencies. For small couplings the underlying MRF model can be well approximated by a independent-column model, reducing the impact of errors in the 400 coupling parameters on the pair counts. Our Felsenstein-like algorithm accurately corrects the pair counts in this case. It is noteworthy that for higher coupling strengths we see a high variance in the pair count correlations which can be ameliorated but not entirely corrected by our approach.

Contacts are predicted from the coupling parameters, therefore as a next step I analyse how well the inferred coupling parameters correlate with the simulated coupling strengths. I compare three different methods: (1) PCD with the assumption of independent sequences, (2) Felsenstein-like algorithm with independent pair-column assumption and (3) FS-PCD based on corrected pair counts. Again phylogeny with stronger ($m=1$) and weaker ($m=2$) dependencies are compared.

For stronger sequence dependencies (Figure 8.9 left column), the phylogeny correction with the independent-pair assumption yields a 10% increase in coupling parameter correlations over the independence assumption (A to C). The average weighted correlation of PCD with corrected pair counts drops to 34% percent (A to E). For weaker sequence dependencies (Figure 8.9 right column), the average weighted correlation of coupling parameters under the independent-pair assumption increases by 5% (B to D) and drops to 26% (B to F). Similar drops in average correlation is also observed across varying alignment depths (Figure 8.10) with FS-PCD reaching 64% (A to E) and 56% (B to F) of the average weighted correlation of PCD.

The low performance of FS-PCD compared to standard PCD (Figure 8.9 and Figure 8.10) is curious, given the higher accuracy of the corrected pair counts (Figure 8.8). Despite the higher correlation, the independent-pair model has an underlying consistency issue that I will illustrate in the following. For

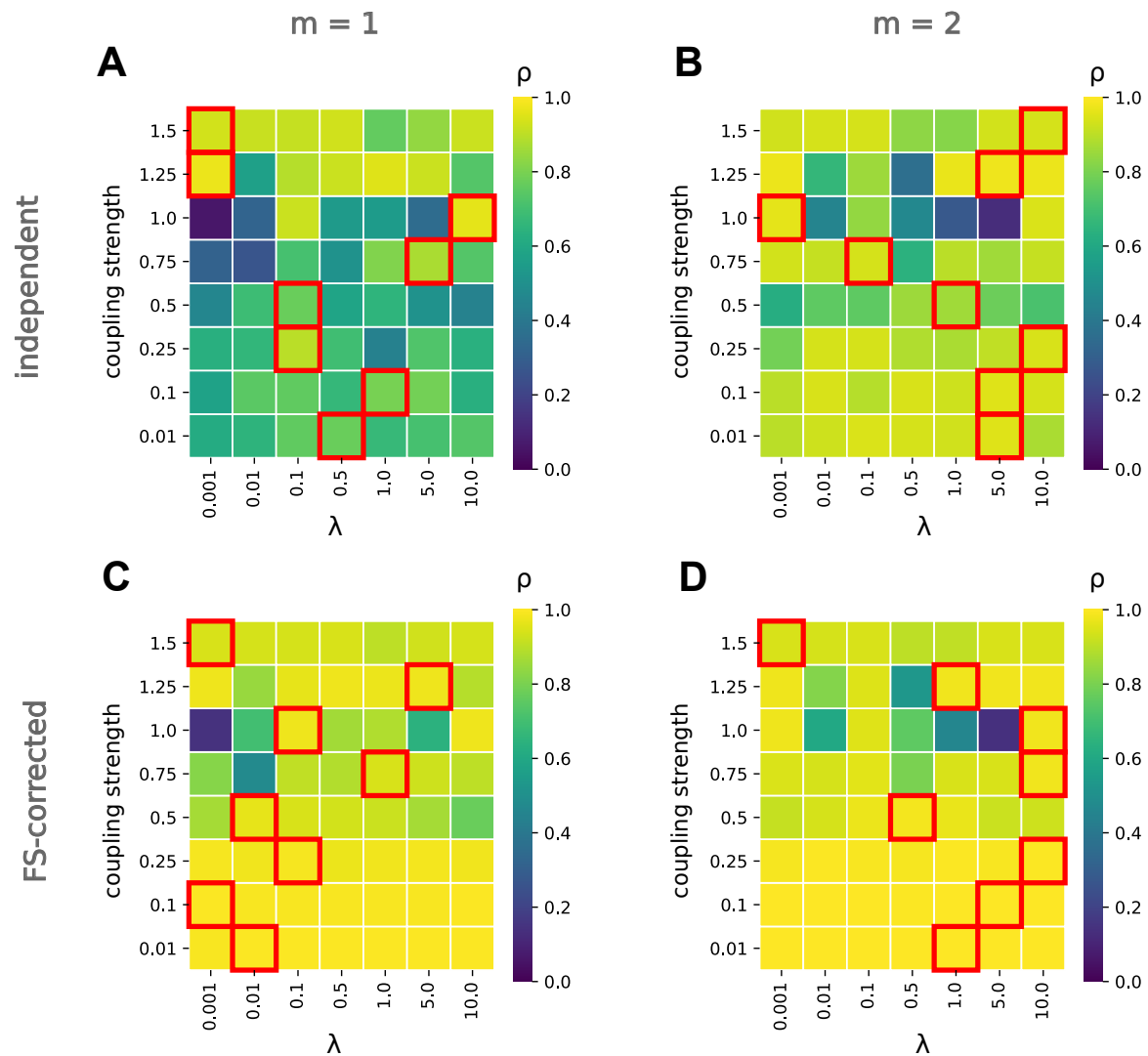


Figure 8.8.: **Phylogeny correction improves pair-count accuracy.** Correlations between simulated and estimated pair-counts. Rows: pair-counts by assuming sequence independence (A, B) and Felsenstein correction (C, D). Columns: average number of mutations in phylogeny from root to leaf: $m = 1$ (A, C) and $m = 2$ (B, D). The phylogeny correction by our Felsenstein-like algorithm increases the correlations by 32% and 10% (A to C and B to D, respectively).

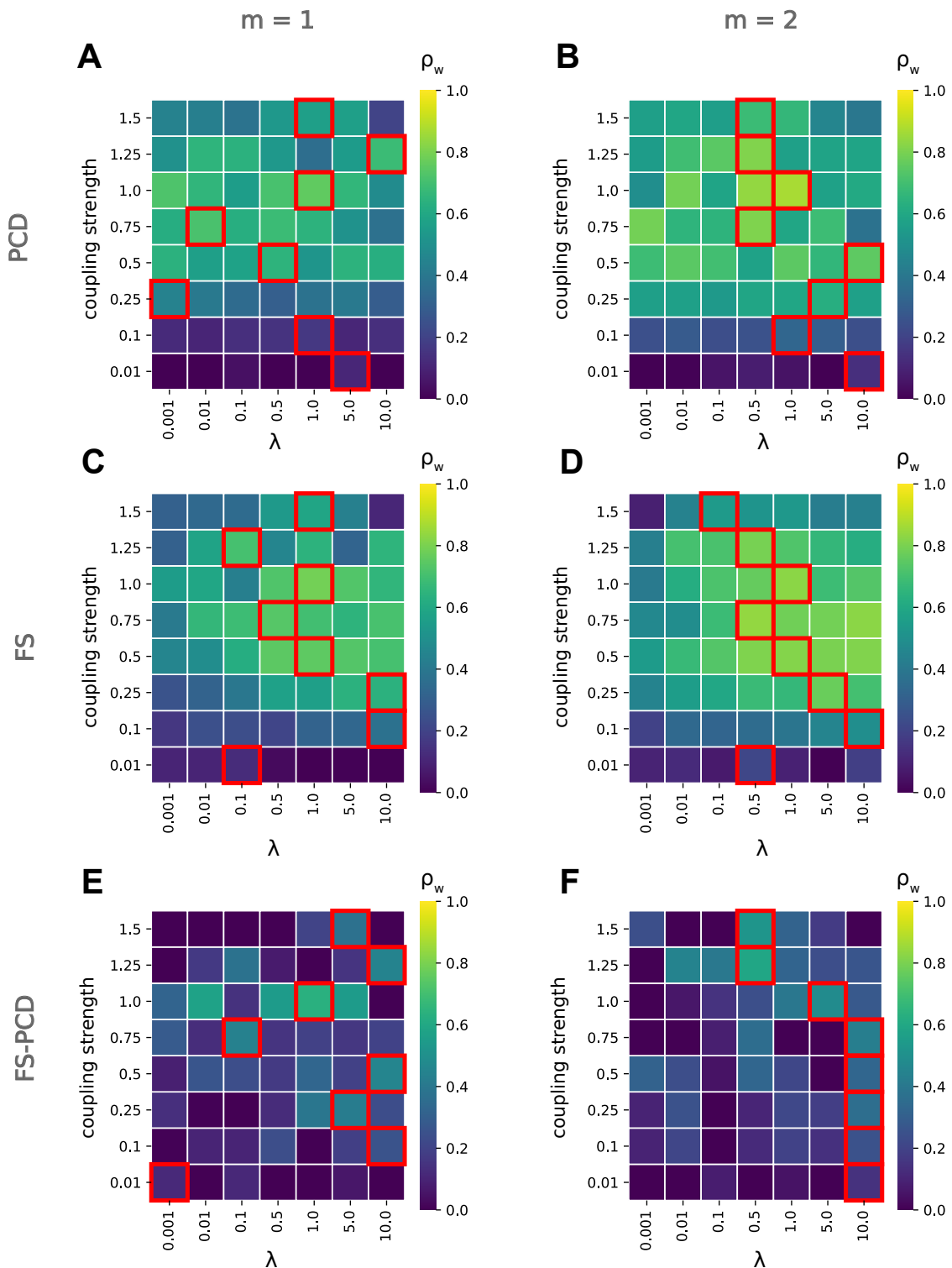


Figure 8.9.: **PCD based on corrected pair counts does not improve pair coupling estimation**
 Weighted correlations between simulated and estimated coupling parameters. Rows: coupling parameters inferred by PCD by assuming sequence independence (A, B), phylogeny-corrected independent-pair couplings (C, D), and PCD based on corrected pair counts (E, F). Columns: average number of mutations in phylogeny from root to leaf: $m = 1$ (A, C, E) and $m = 2$ (B, D, F). The average correlation of pair couplings derived by FS-PCD drops to 34% and 26% compared to PCD for $m=1$ and $m=2$, respectively.

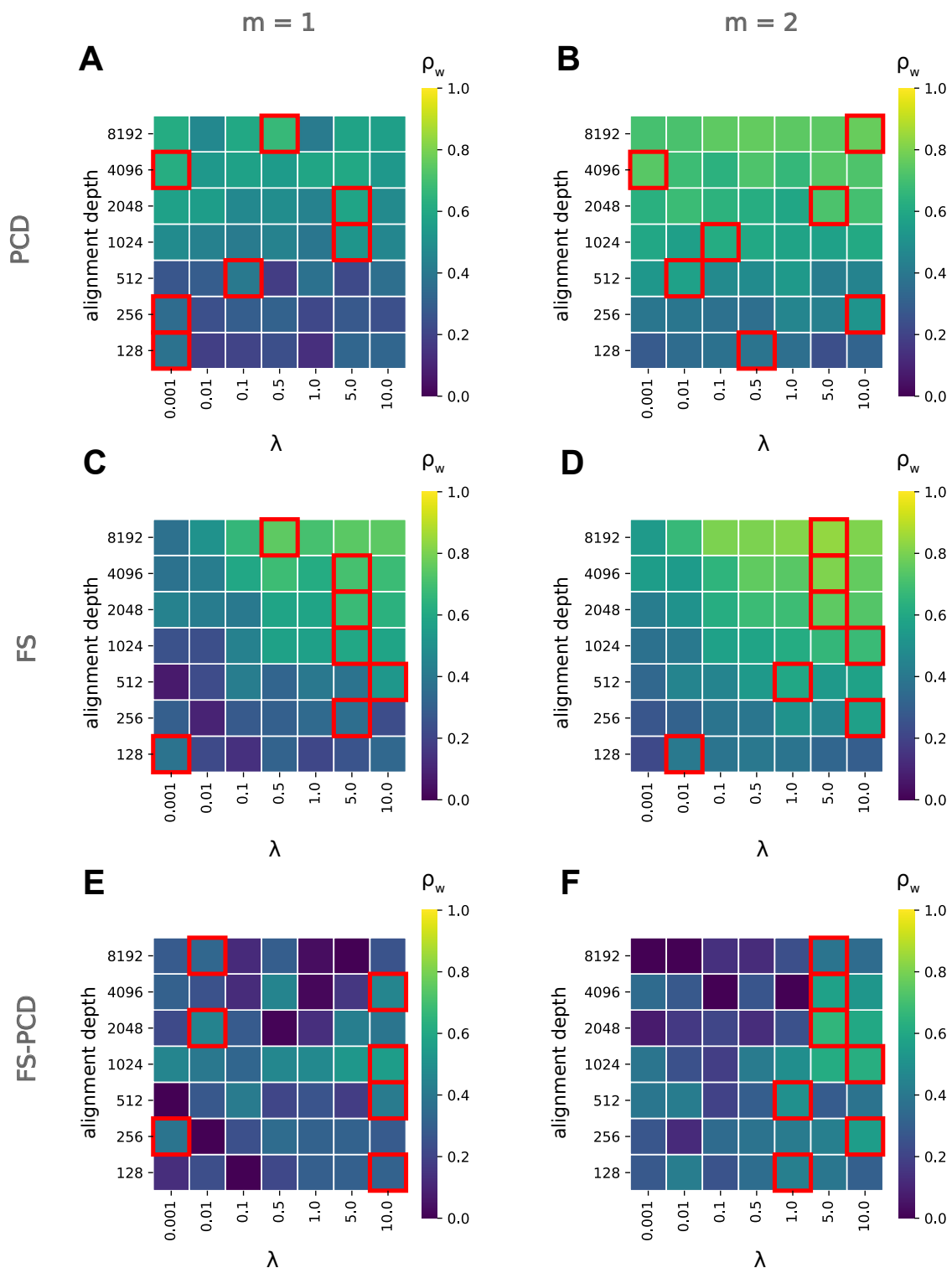


Figure 8.10.: **PCD based on corrected pair counts does not improve pair coupling estimation.** Figure layout reassembles Figure 8.9, but with alignment depth instead of coupling strength on the y-axis. The correlation of pair couplings derived by FS-PCD drop compare to standard PCD to 64% and 56% for $m=1$ and $m=2$, respectively.

independent sequences the observed counts for an amino acid a in a given column i can be calculated consistently, irrespective of the chosen partner column ($n_{ia} = \sum_b n_{ijab}$). Due to the independent-pair approximation this is however not true for the corrected counts, and slightly different n_{ia} are obtained based on the choice of the partner column (Figure 8.11). It is plausible that choosing the average n_{ia} as input to PCD is not a robust strategy to obtain the corresponding pair couplings.

8.2. Pairwise couplings for contact prediction

In the previous section I showed that our FS-PCD method does not recover more accurate MRF coupling parameters for phylogenetically dependent sequences. However our phylogenetically corrected pairwise coupling parameter correlate better with the MRF parameters than the parameters estimated under the sequence independence assumption with PCD (Figure 8.9 and Figure 8.10). In the following I investigate whether the phylogenetically-corrected pairwise couplings can be used in protein contact-prediction for sequence with strong phylogenetic dependencies.

Our contact prediction simulation is based on protein contact maps and MRF parameters chosen such that the coupling parameters are non-zero only when the corresponding positions are in contact according to the contact map. The goal is to predict the contact positions i.e. the positions with non-zero coupling parameters from the simulated alignments. In order to make the simulations more realistic, we use contact maps from solved structures and corresponding MRF parameters derived by a constrained PCD from a previous study in our lab (Vorberg et al., 2018). For faster computation, we randomly select 25 protein-families with between 75 and 125 alignment columns as our benchmark set (Table 8.1).

8.2.1. Simulations with artificial phylogenies

Using the pre-trained family-specific models, I simulate 256 sequences, learn the pairwise pair-couplings with our Felsenstein-like algorithm with $\lambda = 5$. The underlying phylogenies have on average m mutations from root to leaf just like the simulations in section 8.1.2. We compare the performance of standard PCD with the true underlying phylogeny (simulated tree), three phylogenies inferred from the alignment (RAxML FastTree2, family-specific tree) and a phylogeny describing independent sequences (indep sequences) (Figure 8.13 and Figure 8.12). For the tree inference we use RAxML (Stamatakis, 2014) with the Dayhoff substitution matrix (Dayhoff et al., 1978), FastTree2 (Price et al., 2010) with the JTT model (Jones et al., 1992a) and our family-specific model introduced in section 7.1.7.

Even without the entropy correction of the average product correction, the pairwise coupling without phylogeny-correction (indep sequences) shows systematically lower precision when predicting contacts than the global coupling parameters learnt from PCD. While the phylogenetically corrected methods (simulated tree, RAxML, FastTree2, family-specific tree) outperform the pairwise predictions with the independent sequence assumption, the high noise makes it difficult to compare the contact prediction performance with PCD. (Figure 8.12).

The entropy-corrected predictions support the trend that PCD outperforms pairwise contact predictions without phylogeny correction (PCD vs. indep sequences). With reduced entropy, the pairwise phylogeny corrected methods (simulated tree, RAxML, FastTree2, family-specific tree) systematically outperform PCD in predicting contacts. As expected, phylogeny correction based on trees inferred from the simulated alignments (RAxML, FastTree2, family-specific tree) overall perform slightly worse than the ground truth (simulated tree). Taken together the simulations suggest that phylogenetically corrected pairwise coupling

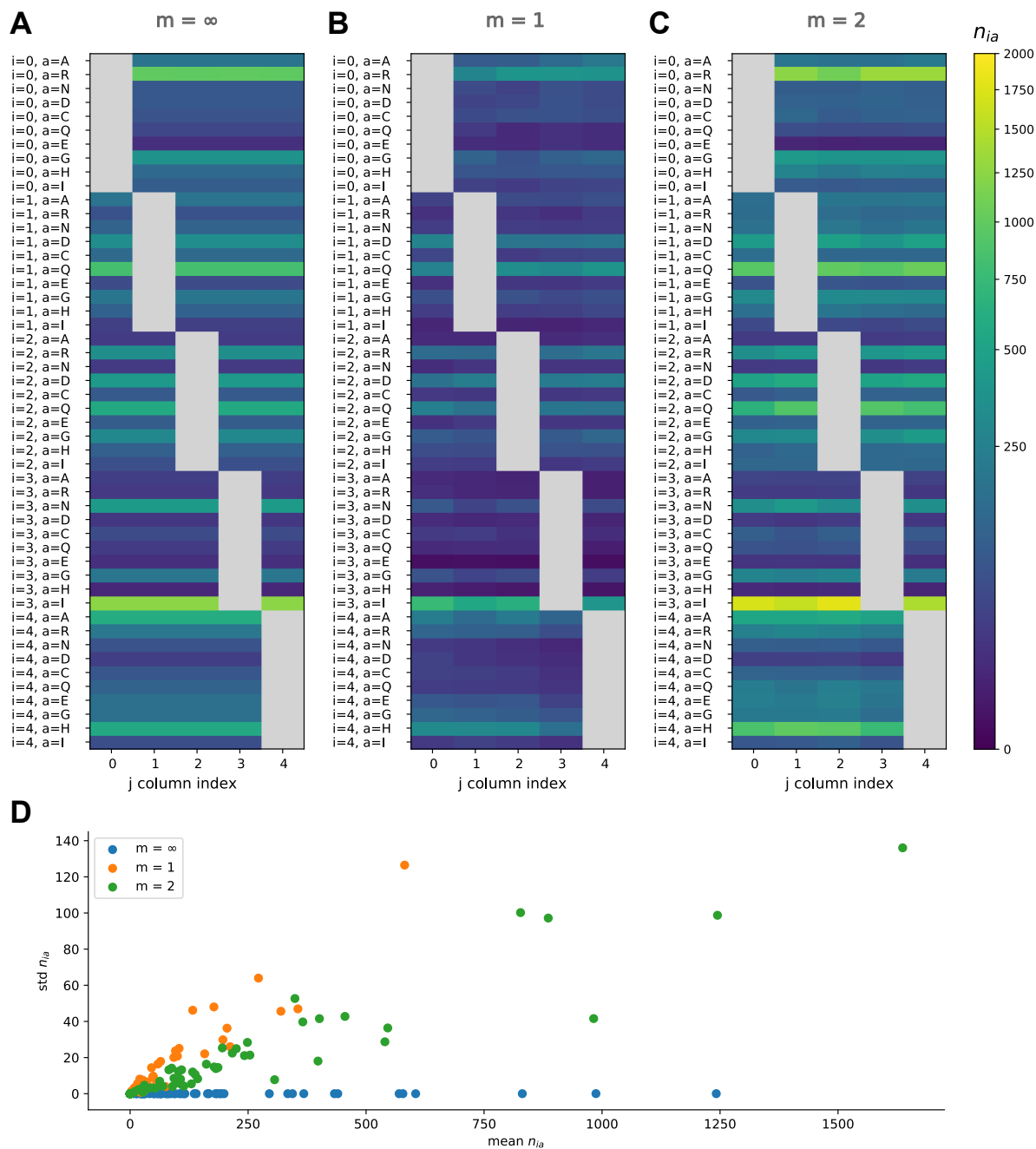


Figure 8.11.: **The independent-pair approximation causes inconsistencies in amino acid counts.** Estimations of n_{ia} counts based on the four possible partner columns for independent sequences ($m = \infty$), strong phylogeny ($m = 1$) and weak phylogeny ($m = 2$). While the striping pattern reveals that the estimated n_{ia} tend to be similar across all four partner column choices, the variance increases as the sequences become less independent (D).

PDB id	# sequences	# columns
1avsA	11693	81
1bdoA	8179	80
1cxyA	2453	81
1d0qA	2513	102
1dlwA	1217	116
1fk5A	882	93
1fnaA	20176	91
1g2rA	933	94
1g9oA	10561	91
1gmxA	11304	107
1h98A	12004	77
1i1jA	1092	106
1iibA	1167	103
1josA	1742	100
1nrvA	3414	100
1p90A	1258	123
1rw1A	2736	114
1smxA	1445	87
1tifA	1590	76
1tqgA	2197	105
1vmbA	1788	107
1whiA	2163	122
1wjxA	1755	112
2hs1A	72784	99
2mhrA	950	118

Table 8.1.: **PDBs used for the benchmark set.** 25 alignments with known structure and between 75 and 125 alignment columns from the PSICOV dataset have randomly been assigned to the benchmark data set.

scores have a potential to outperform PCD in contact prediction with suitable underlying phylogenies. In the following we make the simulations more realistic by using phylogenetic trees trained from real multiple sequence alignments instead of the artificially simulated sequence dependencies.

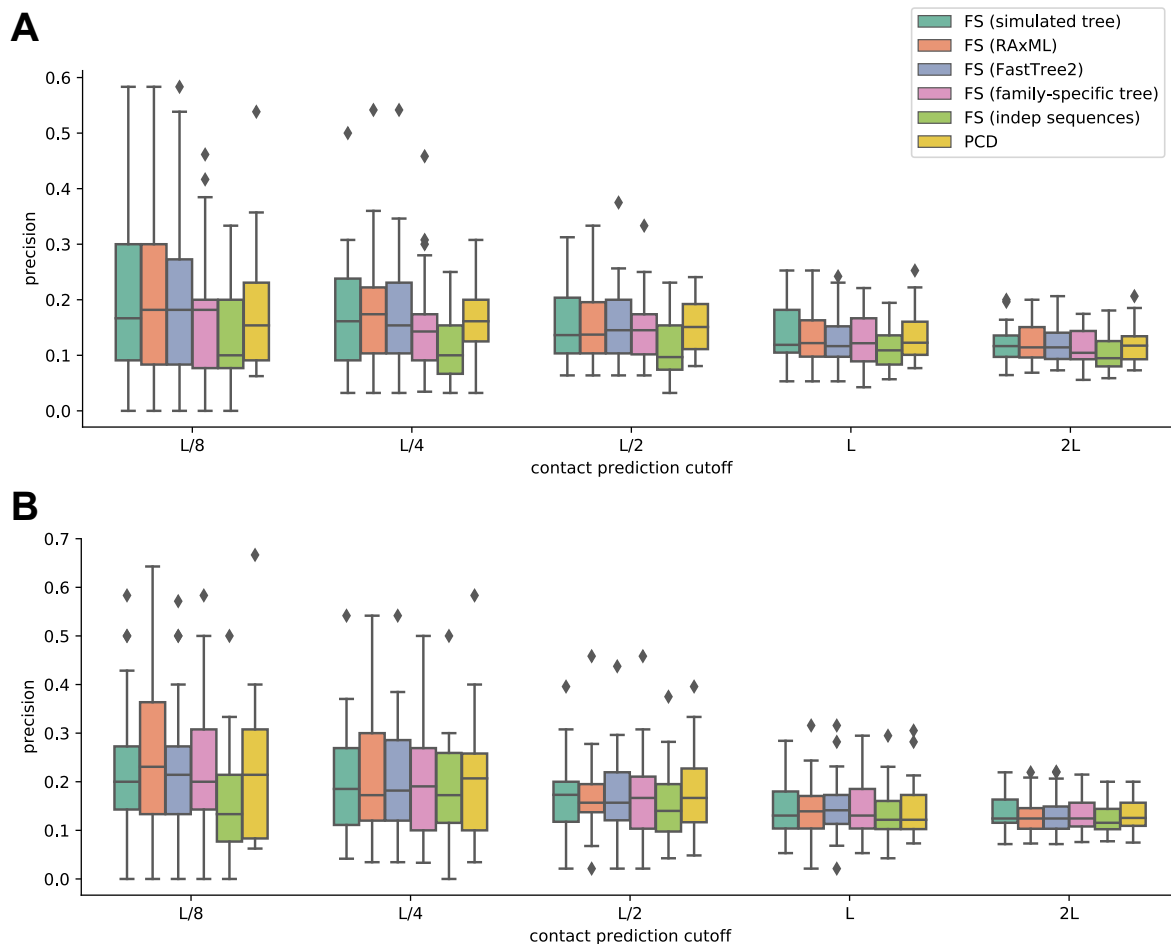


Figure 8.12.: **Phylogeny correction on simulated phylogenies (without APC)**. Due to high noise the performance of contact prediction with phylogeny-corrected pairwise coupling parameters compared to PCD is not clear. **(A)** one mutation per position on average from root to leaf ($m = 1$), **(B)** two mutation per position on average from root to leaf ($m = 2$).

8.2.2. Simulations with learnt phylogenies

In the previous section, I showed that our phylogeny-corrected pairwise coupling parameters predict contacts simulated under a MRF model with higher precision than the traditional PCD method. With equal branch lengths throughout the phylogeny, the simulated trees however do not resemble real-world evolutionary phylogenies. In order to perform more realistic simulations, I use FastTree2 to infer phylogenetic trees from the alignments in our benchmark set (Table 8.1).

As a high number of diverse sequences are available for each protein family in the benchmark set, contacts can be accurately predicted with standard methods such as PCD (Figure 8.14). In this case phylogenetic correction is not required for distinguishing the coevolution signal from the phylogenetic noise. Our goal is to predict contacts and thereby structures of protein families with few sequences and low sequence diversity. In order to obtain phylogenies that more closely resemble the trees in our target objective, I

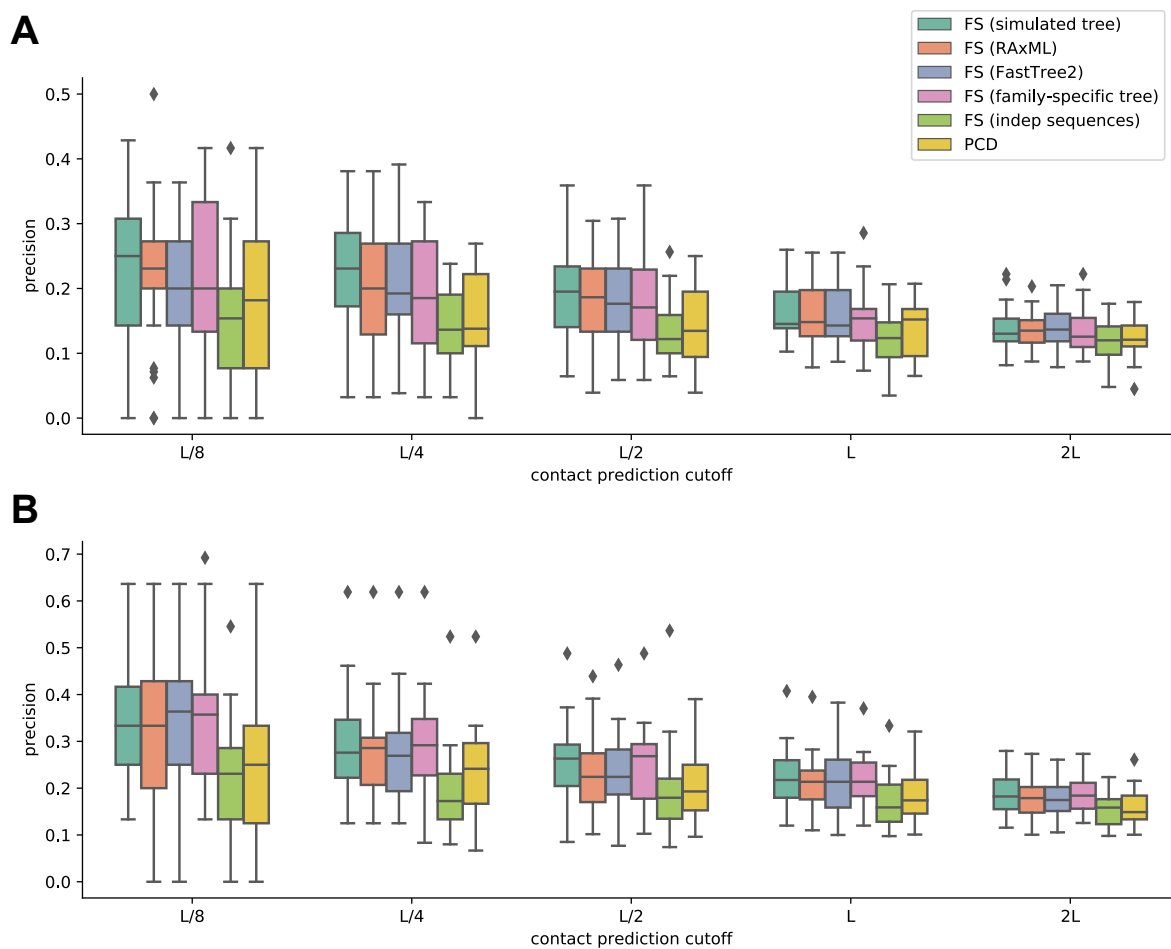


Figure 8.13.: **Phylogeny correction on simulated phylogenies (with APC)**. After the Average Produce Correction (APC), phylogeny corrected pairwise coupling parameters predict contacts better than PCD. **(A)** one mutation per position on average from root to leaf ($m = 1$), **(B)** two mutations per position on average from root to leaf ($m = 2$).

subsample the phylogenetic trees. The sampling strategy consists of two steps: (1) sampling a subtree whose number of leaf nodes falls between an accepted range, weighted by the total number of mutations (2) further randomly subsampling a fixed number of leaf nodes. This strategy is chosen such that the chosen sequences are taken from a real phylogenetic tree, and the alignments are small, with enough information content to derive contact predictions from co-evolution signal. For the first subsampling step I chose two different ranges: subtrees with 200–600 leaf nodes and 200–1000 leaf nodes, leading to less and more diverse sequences, respectively. With the phylogenies in hand, I simulate alignments with CCMgen using the protein-family models generated by constrained PCD and perform contact prediction, as described in section 8.2.1 (Figure 8.15).

Compared to the artificial phylogenies, the contact prediction performance on the realistic phylogenies is generally lower. While the trend that phylogeny-corrected pair-wise predictions (FS RAxML/simulated tree) outperform the uncorrected pair-wise predictions (FS indep sequences) persists, the improvement over PCD disappears, especially for the more diverse sequences.

8.2.3. Applying pair-wise methods to real sequences

As the phylogenetic trees from the previous section have been inferred from sequence alignments with known protein structures, we can apply our methods to real protein sequences instead of sequences simulated under a MRF model (Figure 8.16). Interestingly, the overall contact prediction performance is much higher than expected from the simulations, with a wide margin between PCD and the pairwise methods. While for the more diverse sequences phylogeny correction generally improves pairwise predictions, no improvements are observed for the less diverse sequences.

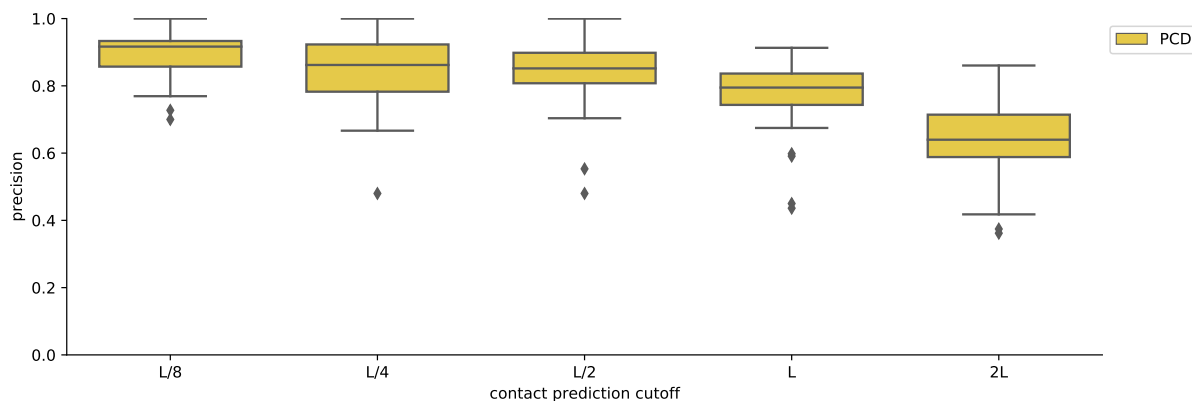


Figure 8.14.: **Contact prediction with PCD on all benchmark sequences.** The protein families in the benchmark set (Table 8.1) contain a high number of sequences. For such large alignments existing DCA methods such as PCD in combination with APC correction can predict contacts with high precision.

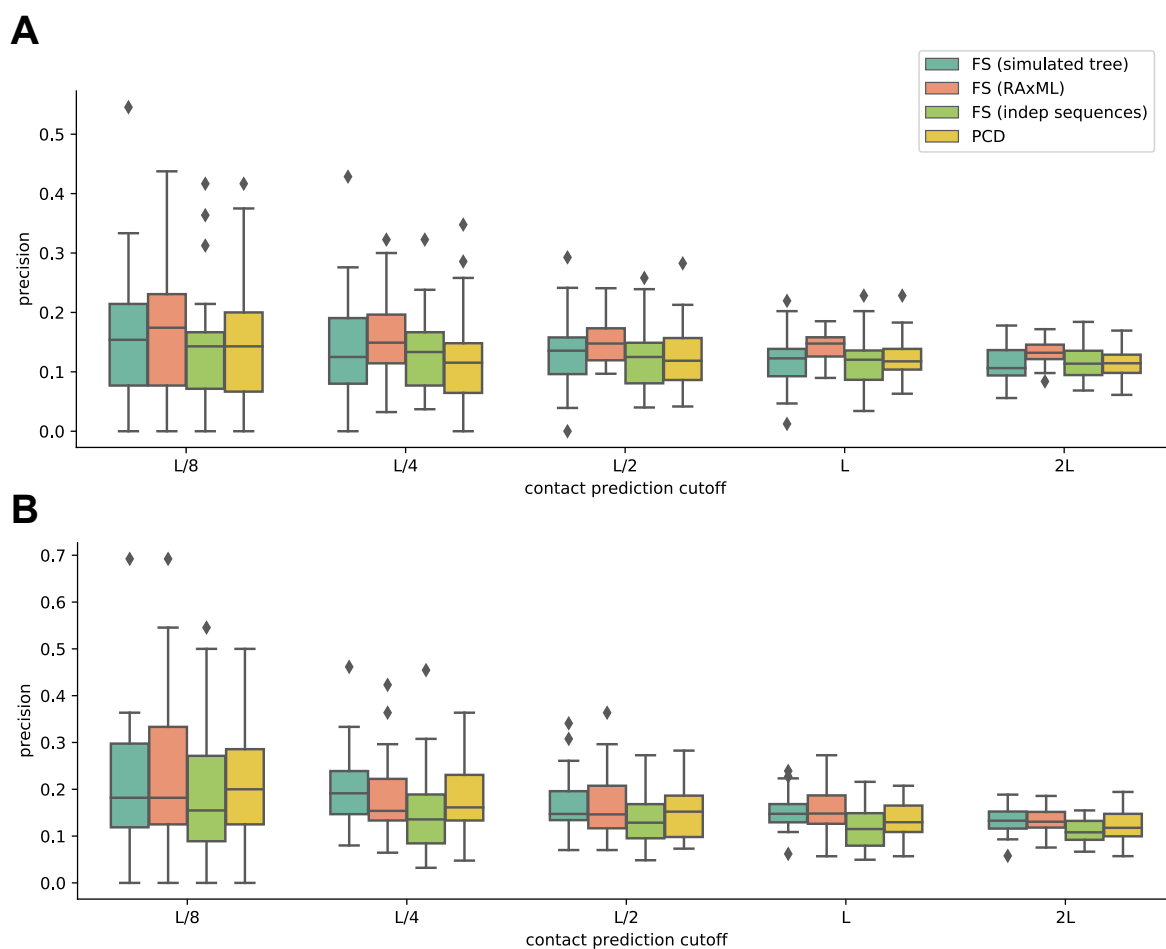


Figure 8.15.: **Contact prediction simulation with real phylogeny.** After the Average Produce Correction (APC), phylogeny corrected pairwise coupling parameters learnt on simulated sequences with real underlying phylogeny do not predict contacts better than PCD. **(A)** subsample with lower sequence diversity **(B)** subsample with higher sequence diversity.

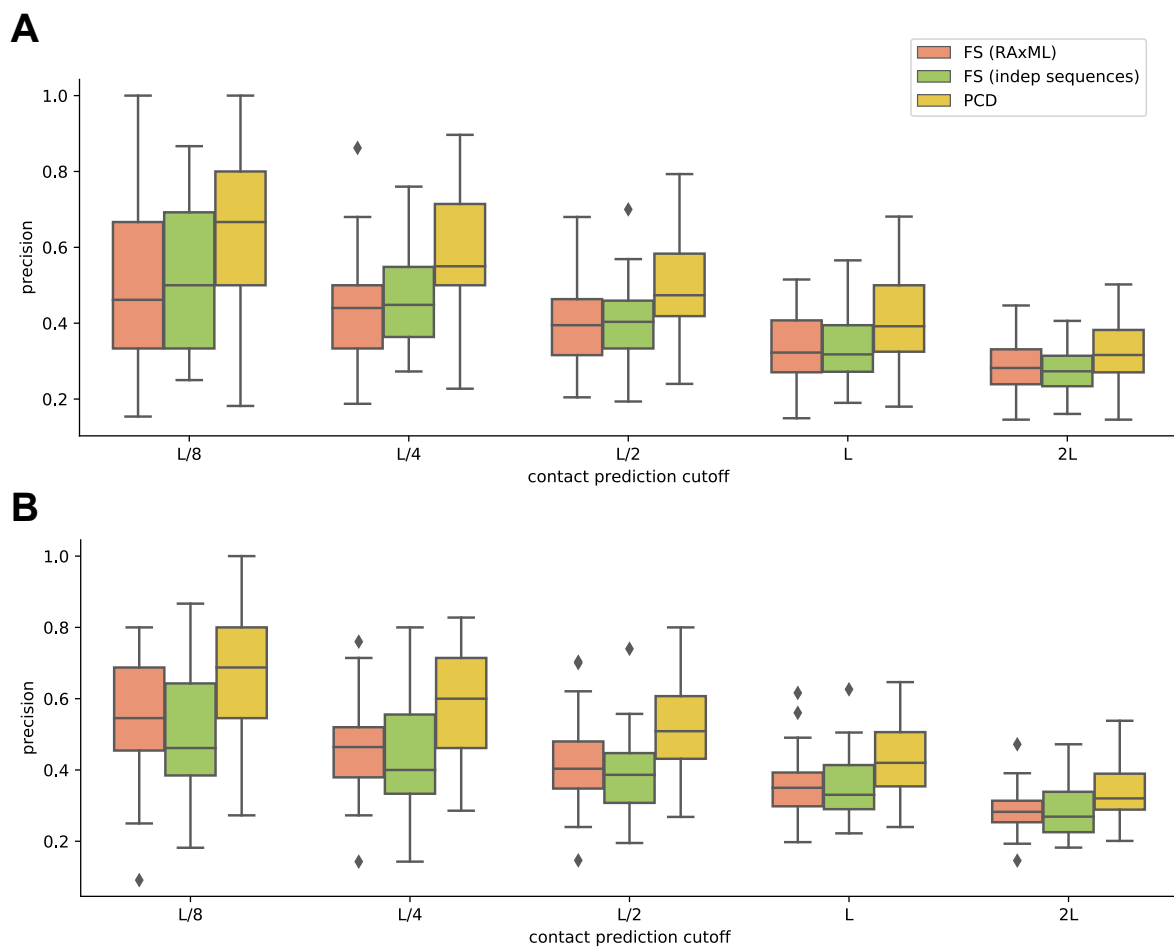


Figure 8.16.: **Contact prediction simulation on real sequences.** A wide performance gap between pairwise and global methods emerges when predicting contacts from real sequences after Average Product Correction.

9. Discussion

The aim of this work was to improve the state-of-the-art of protein-contact prediction by contributing a principled method for correcting out phylogenetic noise in the coevolution features provided by direct coupling analysis (DCA) methods. At the time of the conception of the project, the best contact prediction methods were based on deep neural network architectures repurposed from the computer vision field and all competitive contact-prediction methods relied on coevolution features derived by DCA methods as an input. Denoised coevolution features promised a path towards computationally solving the structures of protein families which have been out of reach for current methods due to having only few members that are evolutionarily highly related.

Traditionally, phylogenetic noise in DCA methods is reduced by clustering and downweighting similar input sequences (Ekeberg et al., 2013), but more recently also bootstrapping (Colavin et al., 2020) and principled methods (Rodriguez Horta et al., 2019) have been suggested as alternative strategies. Sequence reweighting and bootstrapping are robust strategies with few assumptions and work beyond simulated data. It has however been suggested that the improvements due to reweighting are limited (Hockenberry and Wilke, 2019).

Using a phylogenetic tree we were able to derive more accurate pair counts based on a novel pairwise algorithm based on Felsenstein’s pruning algorithm. With FS-DCA we introduced a method with conceptual similarities to a method previously developed by Rodriguez Horta et al. (2019). While methods share the same independent-pair evolutionary model and propose a phylogenetic correction of the pair counts, our method does not rely on inaccurate sampling techniques at the cost of model inconsistencies. Ultimately, the inaccuracies originating from the independent-pair approximation introduced inconsistencies that prevented their use in PCD algorithm.

We further showed that for certain dependency structures, phylogeny-corrected pairwise couplings correlate better with MRF coupling parameters than coupling parameters derived with PCD under an independent sequence model. When comparing our pairwise methods on simulated protein families based on real phylogenies and the corresponding real sequences, we observe a strong mismatch of overall contact prediction performance and a wide gap between our pairwise methods and PCD. This result is consistent with the observation that global methods based on MRF displace pairwise methods and is in line with the speculation that overcoming phylogenetic noise may not be the main bottleneck in MRF-based contact prediction (Rodriguez Horta et al., 2019).

9.1. Shortcomings and limitations

FS-DCA suffers from inconsistencies in the profile probabilities. The independent-pair model estimates a higher number of profile parameters $\binom{L}{2} \times 2A$ compared to $L \times A$ of the MRF model in DCA. Unlike in the MRF model, Rodriguez Horta et al. noticed that in the independent pair-model the profile frequencies $p(x_i = a) = \sum_b p(x_i = a, x_j = b)$ depend on the choice of the partner column j , giving rise to inconsistencies in the model parameters of the desired MRF model. The authors used a constrained

stochastic optimization scheme to ensure the consistency of the profile probabilities. Our approach does not address this inconsistency in favor of a more accurate optimization with a gradient-descent optimization, with the assumption that the inconsistencies are negligible for small coupling strengths. Based on simulations, I show that this assumption does not hold and that the introduced inconsistencies induce large errors in the couple parameter estimations. The observation that the constrained stochastic optimization did not outperform DCA on real world data (Rodriguez Horta et al., 2019) indicates at the very least that solving the inconsistencies may not be the only problem with phylogeny-aware DCA methods.

Pairwise methods cannot account for indirect correlation. Global methods such as DCA and Bayesian networks have been introduced in the contact prediction field to distinguish direct from indirect contributions to the observed coevolution (Weigt et al., 2009; Burger and Van Nimwegen, 2010). Our FS-DCA approach is theoretically appealing because it is a phylogeny-aware generalization of PCD, and thus has all advantages of the global models. I could show that phylogenetically corrected pairwise couplings can produce more accurate pair counts and depending on the underlying sequence dependencies can outperform PCD in predicting contacts. The pairwise methods will however outperform PCD only if the benefit of the corrected phylogentic noise surpasses the cost of being oblivious to indirect effects, making the pairwise method less robust for practical application.

Inadequacy of the evolutionary model. Based on the observation that in the course of evolution the frequency of amino acid exchanges varies with the biochemical properties of the involved amino acids, traditional methods to phylogeny rely on amino acid substitution matrices that quantify the substitution likelihoods based on a given evolutionary time frame. In addition, position specific mutation rates are used to account for function-critical positions being disproportionately highly conserved and thus seemingly mutate at a slower rate.

In contrast to the traditional models, the evolutionary model used in this work is protein-family specific. A protein family is described by a sequence-generating model, parameterized by time-invariant profile and pair-coupling parameters. In case of a mutation, the new amino acid is chosen according to the sequence-generating model. The model is oblivious to the nature of the replaced amino acid present at the time of the mutation event. In reality, evolution is based on both mutation and selection and thus variants created by mutation have to compete with the existing variants. Our independent-pair evolutionary model may very well be too simple to reflect realistic evolutionary amino acid exchanges.

Highly parameterized domain-unspecific model. MRFs capture amino-acid pair interactions with 400 parameters for each possible column pair, yielding highly complex models that are difficult to train accurately on the hundreds to thousands of sequences in each sequence family. In agreement with this, contact prediction based on DCA tends to only work well for large diverse sequence families (Ovchinnikov et al., 2017). Typical strategies in machine learning to overcome this bottleneck are increasing the amount of data and capturing more domain knowledge in the learning objective. MRF models are, however, neither flexible enough to generalize across protein families, nor can they encapsulate biological information such as amino-acid properties by sharing parameters across columns.

9.2. End-to-end revolution

DCA revolutionized the contact prediction field when it was conceived more than 10 years ago. The advent of pseudolikelihood methods allowed to quickly and accurately obtain approximate solutions for MRFs on large data sets, whereas DNNs based on computer vision were introduced to extract high-confidence contacts, effectively recognizing patterns of common local structural motifs from the DCA coevolution features.

More recently, self-attention has been applied to the field as a more flexible model for capturing the interaction graph. MRFs can be understood as statistical graphical models in which nodes represent alignment positions and the edges represent couplings. It has been shown that a model based on factored-attention, a simplified version of the *Scaled Dot-Product Attention* used in the Transformer architecture, achieves comparable performance to MRFs (Bhattacharya et al., 2020) with a reduced model complexity by sharing parameters across protein families. Moreover, jointly learning protein families with a Transformer-based architecture outperforms the traditional MRF models by a large margin (Rao et al., 2021). Unlike the previously employed computer vision models that integrated features from different sources, the new generation of Transformer learns all necessary information directly from sequence alignments. Together with the drastic increase in protein-structure prediction accuracy by AlphaFold2 which is also based on the Transformer architecture, it is plausible that end-to-end Transformer networks will displace coevolution-based methods such as MRFs in the near future.

The switch to end-to-end differentiable protein-structure prediction models is arguably among the most transformative innovations in the computational structural biology field. Not only do Transformer models predict the structure of protein domains at an unparalleled accuracy, they also reunify formerly independent subtasks to one single learning objective that, given enough computational resources, can be solved and innovated upon with general-purpose software. By formulating the protein-folding problem as an end-to-end objective, the latest generation of machine learning models does not rely on externally engineered features. This will likely reduce the impact of methods that produce or correct coevolution features in the near future.

9.3. Outlook

With AlphaFold2, DeepMind solved a defining challenge in the computational biology field - the accurate prediction of protein domain structure from evolutionarily related sequences. With billions of sequences, hundred thousands of structures and well-developed evaluation strategies available, the folding problem solved by AlphaFold2 is however among the most well-defined learning tasks in the field and it remains an open question whether other crucial tasks such as unravelling folding paths, predicting the structure of dynamic and multi-domain proteins, as well as protein complexes and protein interactions with DNA, RNA or small molecules can be solved by similar principles.

The protein folding problem, as it was originally formulated, demanded an algorithm that assigns a stable structure to any foldable amino acid sequence, and thus encapsulates the biophysical mechanisms of folding. The current machine learning models learn their folding logic by generalizing from sequences of protein families together with their known structures. It is however not yet clear how well state-of-the-art models generalize beyond sequences that evolved in the natural course of evolution, a property that could be a milestone in the field of protein engineering for exploring fundamentally new stable folds.

The current generation of machine learning models excel at solving problems with large amounts of

clean data and clear learning objectives. Problems that do not meet these criteria will likely stay fields to be explored by structural biologists. As with so many innovations, machine learning models such as AlphaFold2 are unlikely to end the structural biology field nor replace experimental scientists, but become yet another powerful tool to venture into exploring the complexities in nature.

References

- Adhikari, B. (2020). A fully open-source framework for deep learning protein real-valued distances. *Scientific reports*, 10(1):1–10.
- Aird, D., Ross, M. G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., Jaffe, D. B., Nusbaum, C., and Gnirke, A. (2011). Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome biology*, 12(2):1–14.
- Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, 33(8):831–838.
- AlQuraishi, M. (2019). End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301.
- Altschuh, D., Lesk, A., Bloomer, A., and Klug, A. (1987). Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of molecular biology*, 193(4):693–707.
- Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402.
- Andersson, R., Gebhard, C., Miguel-Escalada, I., Hoof, I., Bornholdt, J., Boyd, M., Chen, Y., Zhao, X., Schmidl, C., Suzuki, T., et al. (2014). An atlas of active enhancers across human cell types and tissues. *Nature*, 507(7493):455–461.
- Andersson, R. and Sandelin, A. (2020). Determinants of enhancer and promoter activities of regulatory elements. *Nature Reviews Genetics*, 21(2):71–87.
- Andreeva, A., Howorth, D., Chothia, C., Kulesha, E., and Murzin, A. G. (2014). SCOP2 prototype: a new approach to protein structure mining. *Nucleic acids research*, 42(D1):D310–D314.
- Andreeva, A., Kulesha, E., Gough, J., and Murzin, A. G. (2020). The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic acids research*, 48(D1):D376–D382.
- Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science*, 181(4096):223–230.
- Anfinsen, C. B., Haber, E., Sela, M., and White Jr, F. (1961). The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):1309.
- Anson, M. and Mirsky, A. (1930). Protein coagulation and its reversal: the preparation of insoluble globin, soluble globin and heme. *The Journal of general physiology*, 13(4):469.
- Apic, G., Gough, J., and Teichmann, S. A. (2001). Domain combinations in archaeal, eubacterial and eukaryotic proteomes. *Journal of molecular biology*, 310(2):311–325.

- Atchley, W. R., Wollenberg, K. R., Fitch, W. M., Terhalle, W., and Dress, A. W. (2000). Correlations among amino acid sites in bHLH protein domains: an information theoretic analysis. *Molecular biology and evolution*, 17(1):164–178.
- Avsec, Z., Agarwal, V., Visentin, D., Ledsam, J. R., Grabska-Barwinska, A., Taylor, K. R., Assael, Y., Jumper, J., Kohli, P., and Kelley, D. R. (2021a). Effective gene expression prediction from sequence by integrating long-range interactions. *bioRxiv*.
- Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., et al. (2021b). Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, pages 1–13.
- Bai, X.-C., McMullan, G., and Scheres, S. H. (2015). How cryo-EM is revolutionizing structural biology. *Trends in biochemical sciences*, 40(1):49–57.
- Balakrishnan, S., Kamisetty, H., Carbonell, J. G., Lee, S.-I., and Langmead, C. J. (2011). Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078.
- Banigan, E. J. and Mirny, L. A. (2020). Loop extrusion: theory meets single-molecule experiments. *Current opinion in cell biology*, 64:124–138.
- Bell, A. C., West, A. G., and Felsenfeld, G. (1999). The protein CTCF is required for the enhancer blocking activity of vertebrate insulators. *Cell*, 98(3):387–396.
- Benos, P. V., Bulyk, M. L., and Stormo, G. D. (2002). Additivity in protein–DNA interactions: how good an approximation is it? *Nucleic acids research*, 30(20):4442–4451.
- Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242.
- Bhattacharya, N., Thomas, N., Rao, R., Daupras, J., Koo, P., Baker, D., Song, Y. S., and Ovchinnikov, S. (2020). Single Layers of Attention Suffice to Predict Protein Contacts. *bioRxiv*.
- Bhaumik, S. R., Raha, T., Aiello, D. P., and Green, M. R. (2004). In vivo target of a transcriptional activator revealed by fluorescence resonance energy transfer. *Genes & development*, 18(3):333–343.
- Björklund, S. and Gustafsson, C. M. (2005). The yeast Mediator complex and its regulation. *Trends in biochemical sciences*, 30(5):240–244.
- Boehning, M., Dugast-Darzacq, C., Rankovic, M., Hansen, A. S., Yu, T., Marie-Nelly, H., McSwiggen, D. T., Kokic, G., Dailey, G. M., Cramer, P., et al. (2018). RNA polymerase II clustering through carboxy-terminal domain phase separation. *Nature structural & molecular biology*, 25(9):833–840.
- Boija, A., Klein, I. A., Sabari, B. R., Dall’Agnese, A., Coffey, E. L., Zamudio, A. V., Li, C. H., Shrinivas, K., Manteiga, J. C., Hannett, N. M., et al. (2018). Transcription factors activate genes through the phase-separation capacity of their activation domains. *Cell*, 175(7):1842–1855.
- Bonneau, R., Tsai, J., Ruczinski, I., Chivian, D., Rohl, C., Strauss, C. E., and Baker, D. (2001). Rosetta in CASP4: progress in ab initio protein structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 45(S5):119–126.
- Bowie, J. U., Luthy, R., and Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure. *Science*, 253(5016):164–170.

- Burger, L. and Van Nimwegen, E. (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, 6(1):e1000633.
- Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208.
- Carter, D., Chakalova, L., Osborne, C. S., Dai, Y.-f., and Fraser, P. (2002). Long-range chromatin regulatory interactions in vivo. *Nature genetics*, 32(4):623–626.
- Chen, Y. and Belmont, A. S. (2019). Genome organization around nuclear speckles. *Current opinion in genetics & development*, 55:91–99.
- Cheng, B. and Price, D. H. (2007). Properties of RNA polymerase II elongation complexes before and after the P-TEFb-mediated transition into productive elongation. *Journal of Biological Chemistry*, 282(30):21901–21912.
- Chiu, D. K. and Kolodziejczak, T. (1991). Inferring consensus structure from nucleic acid sequences. *Bioinformatics*, 7(3):347–352.
- Cho, W.-K., Spille, J.-H., Hecht, M., Lee, C., Li, C., Grube, V., and Cisse, I. I. (2018). Mediator and RNA polymerase II clusters associate in transcription-dependent condensates. *Science*, 361(6400):412–415.
- Chothia, C. (1992). One thousand families for the molecular biologist. *Nature*, 357(6379):543–544.
- Chothia, C. and Lesk, A. M. (1986). The relation between the divergence of sequence and structure in proteins. *The EMBO journal*, 5(4):823–826.
- Christmann, J. L. and Dahmus, M. (1981). Monoclonal antibody specific for calf thymus RNA polymerases IIO and IIA. *Journal of Biological Chemistry*, 256(22):11798–11803.
- Claverie, J.-M. (2001). What if there are only 30,000 human genes? *Science*, 291(5507):1255–1257.
- Colavin, A., Atolia, E., Bitbol, A.-F., and Huang, K. C. (2020). Extracting the phylogenetic dimension of coevolution reveals hidden functional signal. *bioRxiv*.
- Colgan, D. F. and Manley, J. L. (1997). Mechanism and regulation of mRNA polyadenylation. *Genes & development*, 11(21):2755–2766.
- Consortium, E. P. et al. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57.
- Corden, J. L. (1990). Tails of RNA polymerase II. *Trends in biochemical sciences*, 15(10):383–387.
- Core, L. and Adelman, K. (2019). Promoter-proximal pausing of RNA polymerase II: a nexus of gene regulation. *Genes & development*, 33(15-16):960–982.
- Core, L. J., Martins, A. L., Danko, C. G., Waters, C. T., Siepel, A., and Lis, J. T. (2014). Analysis of nascent RNA identifies a unified architecture of initiation regions at mammalian promoters and enhancers. *Nature genetics*, 46(12):1311.
- Cossio, P., Trovato, A., Pietrucci, F., Seno, F., Maritan, A., and Laio, A. (2010). Exploring the universe of protein structures beyond the Protein Data Bank. *PLoS Comput Biol*, 6(11):e1000957.
- Cramer, P. (2019). Organization and regulation of gene transcription. *Nature*, 573(7772):45–54.

- Cramer, P., Bushnell, D. A., and Kornberg, R. D. (2001). Structural basis of transcription: RNA polymerase II at 2.8 Ångstrom resolution. *Science*, 292(5523):1863–1876.
- Cremer, T. and Cremer, C. (2001). Chromosome territories, nuclear architecture and gene regulation in mammalian cells. *Nature reviews genetics*, 2(4):292–301.
- Crocker, J., Noon, E. P.-B., and Stern, D. L. (2016). The soft touch: low-affinity transcription factor binding sites in development and evolution. *Current topics in developmental biology*, 117:455–469.
- Dao, L. T., Galindo-Albarrán, A. O., Castro-Mondragon, J. A., Andrieu-Soler, C., Medina-Rivera, A., Souaid, C., Charbonnier, G., Griffon, A., Vanhille, L., Stephen, T., et al. (2017). Genome-wide characterization of mammalian promoters with distal enhancer functions. *Nature genetics*, 49(7):1073.
- Darwin, C. (1859). *The origin of species by means of natural selection*. John Murray, London.
- Darwin, C., Wallace, A. R., Lyell, S. C., and Hooker, J. D. (1858). On the tendency of species to form varieties: and on the perpetuation of varieties and species by natural means of selection. Linnean Society of London.
- Das, M. K. and Dai, H.-K. (2007). A survey of DNA motif finding algorithms. *BMC bioinformatics*, 8(7):1–13.
- Dayhoff, M., Schwartz, R., and Orcutt, B. (1978). 22 a model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5:345–352.
- De Laat, W. and Duboule, D. (2013). Topology of mammalian developmental enhancers and their regulatory landscapes. *Nature*, 502(7472):499–506.
- De Santa, F., Barozzi, I., Mietton, F., Ghisletti, S., Polletti, S., Tusi, B. K., Muller, H., Ragoussis, J., Wei, C.-L., and Natoli, G. (2010). A large fraction of extragenic RNA pol II transcription sites overlap enhancers. *PLoS Biol*, 8(5):e1000384.
- de Wit, E., Vos, E. S., Holwerda, S. J., Valdes-Quezada, C., Verstegen, M. J., Teunissen, H., Splinter, E., Wijchers, P. J., Krijger, P. H., and de Laat, W. (2015). CTCF binding polarity determines chromatin looping. *Molecular cell*, 60(4):676–684.
- Deng, W., Lee, J., Wang, H., Miller, J., Reik, A., Gregory, P. D., Dean, A., and Blobel, G. A. (2012). Controlling long-range genomic interactions at a native locus by targeted tethering of a looping factor. *Cell*, 149(6):1233–1244.
- Diao, Y., Fang, R., Li, B., Meng, Z., Yu, J., Qiu, Y., Lin, K. C., Huang, H., Liu, T., Marina, R. J., et al. (2017). A tiling-deletion-based genetic screen for cis-regulatory element identification in mammalian cells. *Nature methods*, 14(6):629–635.
- Diaz, A., Park, K., Lim, D. A., and Song, J. S. (2012). Normalization, bias correction, and peak calling for ChIP-seq. *Statistical applications in genetics and molecular biology*, 11(3).
- Dixon, J. R., Selvaraj, S., Yue, F., Kim, A., Li, Y., Shen, Y., Hu, M., Liu, J. S., and Ren, B. (2012). Topological domains in mammalian genomes identified by analysis of chromatin interactions. *Nature*, 485(7398):376–380.
- Dunn, S. D., Wahl, L. M., and Gloor, G. B. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340.

- Eggeling, R. (2018). Disentangling transcription factor binding site complexity. *Nucleic acids research*, 46(20):e121–e121.
- Eggeling, R., Roos, T., Myllymäki, P., and Grosse, I. (2015). Inferring intra-motif dependencies of DNA binding sites from ChIP-seq data. *BMC bioinformatics*, 16(1):1–15.
- Ekeberg, M., Hartonen, T., and Aurell, E. (2014). Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356.
- Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1):012707.
- Ekman, D., Björklund, Å. K., Frey-Skött, J., and Elofsson, A. (2005). Multi-domain proteins in the three kingdoms of life: orphan domains and other unassigned regions. *Journal of molecular biology*, 348(1):231–243.
- Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Bhowmik, D., et al. (2020). ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv preprint arXiv:2007.06225*.
- Engreitz, J. M., Haines, J. E., Perez, E. M., Munson, G., Chen, J., Kane, M., McDonel, P. E., Guttman, M., and Lander, E. S. (2016). Local regulation of gene expression by lncRNA promoters, transcription and splicing. *Nature*, 539(7629):452–455.
- Eraslan, G., Avsec, Ž., Gagneur, J., and Theis, F. J. (2019). Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403.
- Farley, E. K., Olson, K. M., Zhang, W., Brandt, A. J., Rokhsar, D. S., and Levine, M. S. (2015). Suboptimization of developmental enhancers. *Science*, 350(6258):325–328.
- Farley, E. K., Olson, K. M., Zhang, W., Rokhsar, D. S., and Levine, M. S. (2016). Syntax compensates for poor binding sites to encode tissue specificity of developmental enhancers. *Proceedings of the National Academy of Sciences*, 113(23):6508–6513.
- Fitch, W. M. and Markowitz, E. (1970). An improved method for determining codon variability in a gene and its application to the rate of fixation of mutations in evolution. *Biochemical genetics*, 4(5):579–593.
- Fornes, O., Castro-Mondragon, J. A., Khan, A., Van der Lee, R., Zhang, X., Richmond, P. A., Modi, B. P., Correard, S., Gheorghe, M., Baranašić, D., et al. (2020). JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic acids research*, 48(D1):D87–D92.
- Fox, N. K., Brenner, S. E., and Chandonia, J.-M. (2014). SCOPe: Structural Classification of Protein-sextended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309.
- Franke, M., Ibrahim, D. M., Andrey, G., Schwarzer, W., Heinrich, V., Schöpflin, R., Kraft, K., Kempfer, R., Jerković, I., Chan, W.-L., et al. (2016). Formation of new chromatin domains determines pathogenicity of genomic duplications. *Nature*, 538(7624):265–269.
- Frum, T., Watts, J. L., and Ralston, A. (2019). TEAD4, YAP1 and WWTR1 prevent the premature onset of pluripotency prior to the 16-cell stage. *Development*, 146(17).

- Fudenberg, G., Abdennur, N., Imakaev, M., Goloborodko, A., and Mirny, L. A. (2017). Emerging evidence of chromosome folding by loop extrusion. In *Cold Spring Harbor symposia on quantitative biology*, volume 82, pages 45–55. Cold Spring Harbor Laboratory Press.
- Fukaya, T., Lim, B., and Levine, M. (2016). Enhancer control of transcriptional bursting. *Cell*, 166(2):358–368.
- Fulton, D. L., Sundararajan, S., Badis, G., Hughes, T. R., Wasserman, W. W., Roach, J. C., and Sladek, R. (2009). TFCat: the curated catalog of mouse and human transcription factors. *Genome biology*, 10(3):1–14.
- Ge, W., Meier, M., Roth, C., and Söding, J. (2021). Bayesian Markov models improve the prediction of binding motifs beyond first order. *NAR Genomics and Bioinformatics*, 3(2):lqab026.
- Gibson, B. A., Doolittle, L. K., Schneider, M. W., Jensen, L. E., Gamarra, N., Henry, L., Gerlich, D. W., Redding, S., and Rosen, M. K. (2019). Organization of chromatin by intrinsic and regulated phase separation. *Cell*, 179(2):470–484.
- Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317.
- Govindarajan, S., Recabarren, R., and Goldstein, R. A. (1999). Estimating the total number of protein folds. *Proteins: Structure, Function, and Bioinformatics*, 35(4):408–414.
- Gressel, S., Schwalb, B., and Cramer, P. (2019). The pause-initiation limit restricts transcription activation in human cells. *Nature communications*, 10(1):1–12.
- Guo, Y. E., Manteiga, J. C., Henninger, J. E., Sabari, B. R., Dall’Agnese, A., Hannett, N. M., Spille, J.-H., Afeyan, L. K., Zamudio, A. V., Shrinivas, K., et al. (2019). Pol II phosphorylation regulates a switch between transcriptional and splicing condensates. *Nature*, 572(7770):543–548.
- Hahn, S. (2004). Structure and mechanism of the RNA polymerase II transcription machinery. *Nature structural & molecular biology*, 11(5):394–403.
- Hansen, A. S., Pustova, I., Cattoglio, C., Tjian, R., and Darzacq, X. (2017). CTCF and cohesin regulate chromatin loop stability with distinct dynamics. *Elife*, 6:e25776.
- Heintzman, N. D., Hon, G. C., Hawkins, R. D., Kheradpour, P., Stark, A., Harp, L. F., Ye, Z., Lee, L. K., Stuart, R. K., Ching, C. W., et al. (2009). Histone modifications at human enhancers reflect global cell-type-specific gene expression. *Nature*, 459(7243):108–112.
- Heintzman, N. D., Stuart, R. K., Hon, G., Fu, Y., Ching, C. W., Hawkins, R. D., Barrera, L. O., Van Calcar, S., Qu, C., Ching, K. A., et al. (2007). Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nature genetics*, 39(3):311–318.
- Heo, L. and Feig, M. (2018). Experimental accuracy in protein structure refinement via molecular dynamics simulations. *Proceedings of the National Academy of Sciences*, 115(52):13276–13281.
- Hockenberry, A. J. and Wilke, C. O. (2019). Phylogenetic weighting does little to improve the accuracy of evolutionary coupling analyses. *Entropy*, 21(10):1000.
- Hyman, A. A., Weber, C. A., and Jülicher, F. (2014). Liquid-liquid phase separation in biology. *Annual review of cell and developmental biology*, 30:39–58.

- Illergård, K., Ardell, D. H., and Elofsson, A. (2009). Structure is three to ten times more conserved than sequence: a study of structural response in protein cores. *Proteins: Structure, Function, and Bioinformatics*, 77(3):499–508.
- Ingraham, J., Riesselman, A. J., Sander, C., and Marks, D. S. (2019). Learning Protein Structure with a Differentiable Simulator. In *ICLR*.
- Iwafuchi-Doi, M. and Zaret, K. S. (2014). Pioneer transcription factors in cell reprogramming. *Genes & development*, 28(24):2679–2692.
- Jeronimo, C., Bataille, A. R., and Robert, F. (2013). The writers, readers, and functions of the RNA polymerase II C-terminal domain code. *Chemical reviews*, 113(11):8491–8522.
- Johnson, D. S., Mortazavi, A., Myers, R. M., and Wold, B. (2007). Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502.
- Jolma, A., Kivioja, T., Toivonen, J., Cheng, L., Wei, G., Enge, M., Taipale, M., Vaquerizas, J. M., Yan, J., Sillanpää, M. J., et al. (2010). Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome research*, 20(6):861–873.
- Jolma, A., Yan, J., Whittington, T., Toivonen, J., Nitta, K. R., Rastas, P., Morgunova, E., Enge, M., Taipale, M., Wei, G., et al. (2013). DNA-binding specificities of human transcription factors. *Cell*, 152(1-2):327–339.
- Jolma, A., Yin, Y., Nitta, K. R., Dave, K., Popov, A., Taipale, M., Enge, M., Kivioja, T., Morgunova, E., and Taipale, J. (2015). DNA-dependent formation of transcription factor pairs alters their binding specificity. *Nature*, 527(7578):384–388.
- Jones, D. T., Buchan, D. W., Cozzetto, D., and Pontil, M. (2012). PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190.
- Jones, D. T. and McGuffin, L. J. (2003). Assembling novel protein folds from super-secondary structural fragments. *Proteins: Structure, Function, and Bioinformatics*, 53(S6):480–485.
- Jones, D. T., Singh, T., Kosciolk, T., and Tetchner, S. (2015). MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics*, 31(7):999–1006.
- Jones, D. T., Taylor, W. R., and Thornton, J. M. (1992a). The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282.
- Jones, D. T., Taylor, W., and Thornton, J. M. (1992b). A new approach to protein fold recognition. *Nature*, 358(6381):86–89.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Tunyasuvunakool, K., Ronneberger, O., Bates, R., ídek, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Potapenko, A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Steinegger, M., Pacholska, M., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2020). In Fourteenth Critical Assessment of Techniques for Protein Structure Prediction.
- Juven-Gershon, T., Hsu, J.-Y., Theisen, J. W., and Kadonaga, J. T. (2008). The RNA polymerase II core promoter: the gateway to transcription. *Current opinion in cell biology*, 20(3):253–259.

- Kadauke, S. and Blobel, G. A. (2009). Chromatin loops in gene regulation. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1789(1):17–25.
- Kamisetty, H., Ovchinnikov, S., and Baker, D. (2013). Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679.
- Karbalayghareh, A., Sahin, M., and Leslie, C. S. (2021). Chromatin interaction aware gene regulatory modeling with graph attention networks. *bioRxiv*.
- Kass, I. and Horovitz, A. (2002). Mapping pathways of allosteric communication in GroEL by analysis of correlated mutations. *Proteins: Structure, Function, and Bioinformatics*, 48(4):611–617.
- Keilwagen, J. and Grau, J. (2015). Varying levels of complexity in transcription factor binding motifs. *Nucleic acids research*, 43(18):e119–e119.
- Kelley, D. R., Snoek, J., and Rinn, J. L. (2016). Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999.
- Kempfer, R. and Pombo, A. (2020). Methods for mapping 3D chromosome architecture. *Nature Reviews Genetics*, 21(4):207–226.
- Kim, D. E., DiMaio, F., Yu-Ruei Wang, R., Song, Y., and Baker, D. (2014). One contact for every twelve residues allows robust and accurate topology-level protein structure modeling. *Proteins: Structure, Function, and Bioinformatics*, 82:208–218.
- Kim, J., Han, K. Y., Khanna, N., Ha, T., and Belmont, A. S. (2019). Nuclear speckle fusion via long-range directional motion regulates speckle morphology after transcriptional inhibition. *Journal of cell science*, 132(8).
- Kim, K.-Y. and Levin, D. E. (2011). Mpk1 MAPK association with the Paf1 complex blocks Sen1-mediated premature transcription termination. *Cell*, 144(5):745–756.
- Kim, S., Yu, N.-K., and Kaang, B.-K. (2015). CTCF as a multifunctional protein in genome regulation and gene expression. *Experimental & molecular medicine*, 47(6):e166–e166.
- Kim, T.-K., Ebright, R. H., and Reinberg, D. (2000). Mechanism of ATP-dependent promoter melting by transcription factor IIIH. *Science*, 288(5470):1418–1421.
- Kim, T.-K., Hemberg, M., Gray, J. M., Costa, A. M., Bear, D. M., Wu, J., Harmin, D. A., Laptewicz, M., Barbara-Haley, K., Kuersten, S., et al. (2010). Widespread transcription at neuronal activity-regulated enhancers. *Nature*, 465(7295):182–187.
- Koch, F., Fenouil, R., Gut, M., Cauchy, P., Albert, T. K., Zacarias-Cabeza, J., Spicuglia, S., De La Chapelle, A. L., Heidemann, M., Hintermair, C., et al. (2011). Transcription initiation platforms and GTF recruitment at tissue-specific enhancers and promoters. *Nature structural & molecular biology*, 18(8):956.
- Kolodny, R., Pereyaslavets, L., Samson, A. O., and Levitt, M. (2013). On the universe of protein folds. *Annual review of biophysics*, 42:559–582.
- Kopec, K. O. and Lupas, A. N. (2013). β -Propeller blades as ancestral peptides in protein evolution. *PLoS One*, 8(10):e77074.

- Korber, B. T., Farber, R. M., Wolpert, D. H., and Lapedes, A. S. (1993). Covariation of mutations in the V3 loop of human immunodeficiency virus type 1 envelope protein: an information theoretic analysis. *Proceedings of the National Academy of Sciences*, 90(15):7176–7180.
- Kornberg, R. D. (1974). Chromatin structure: a repeating unit of histones and DNA. *Science*, 184(4139):868–871.
- Kosciolek, T. and Jones, D. T. (2016). Accurate contact predictions using covariation techniques and machine learning. *Proteins: Structure, Function, and Bioinformatics*, 84:145–151.
- Kribelbauer, J. F., Rastogi, C., Bussemaker, H. J., and Mann, R. S. (2019). Low-affinity binding sites and the transcription factor specificity paradox in eukaryotes. *Annual review of cell and developmental biology*, 35:357–379.
- Krivov, G. G., Shapovalov, M. V., and Dunbrack Jr, R. L. (2009). Improved prediction of protein side-chain conformations with SCWRL4. *Proteins: Structure, Function, and Bioinformatics*, 77(4):778–795.
- Kuehner, J. N., Pearson, E. L., and Moore, C. (2011). Unravelling the means to an end: RNA polymerase II transcription termination. *Nature reviews Molecular cell biology*, 12(5):283–294.
- Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697.
- Kulakovskiy, I., Levitsky, V., Oshchepkov, D., Bryzgalov, L., Vorontsov, I., and Makeev, V. (2013). From binding motifs in ChIP-Seq data to improved models of transcription factor binding sites. *Journal of bioinformatics and computational biology*, 11(01):1340004.
- Kulakovskiy, I. V., Vorontsov, I. E., Yevshin, I. S., Sharipov, R. N., Fedorova, A. D., Rumynskiy, E. I., Medvedeva, Y. A., Magana-Mora, A., Bajic, V. B., Papatsenko, D. A., et al. (2018). HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic acids research*, 46(D1):D252–D259.
- Kuras, L., Borggrefe, T., and Kornberg, R. D. (2003). Association of the Mediator complex with enhancers of active genes. *Proceedings of the National Academy of Sciences*, 100(24):13887–13891.
- Lambert, S. A., Jolma, A., Campitelli, L. F., Das, P. K., Yin, Y., Albu, M., Chen, X., Taipale, J., Hughes, T. R., and Weirauch, M. T. (2018). The human transcription factors. *Cell*, 172(4):650–665.
- Larson, A. G., Elnatan, D., Keenen, M. M., Trnka, M. J., Johnston, J. B., Burlingame, A. L., Agard, D. A., Redding, S., and Narlikar, G. J. (2017). Liquid droplet formation by HP1 α suggests a role for phase separation in heterochromatin. *Nature*, 547(7662):236–240.
- Larson, A. G. and Narlikar, G. J. (2018). The role of phase separation in heterochromatin formation, function, and regulation. *Biochemistry*, 57(17):2540–2548.
- Larson, S. M., Di Nardo, A. A., and Davidson, A. R. (2000). Analysis of covariation in an SH3 domain sequence alignment: applications in tertiary contact prediction and the design of compensating hydrophobic core substitutions. *Journal of molecular biology*, 303(3):433–446.
- Leaver-Fay, A., Tyka, M., Lewis, S. M., Lange, O. F., Thompson, J., Jacak, R., Kaufman, K. W., Renfrew, P. D., Smith, C. A., Sheffler, W., et al. (2011). ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. *Methods in enzymology*, 487:545–574.

- Lee, C.-K., Shibata, Y., Rao, B., Strahl, B. D., and Lieb, J. D. (2004). Evidence for nucleosome depletion at active regulatory regions genome-wide. *Nature genetics*, 36(8):900–905.
- Levitt, M. and Warshel, A. (1975). Computer simulation of protein folding. *Nature*, 253(5494):694–698.
- Li, C. H., Coffey, E. L., Dall’Agnese, A., Hannett, N. M., Tang, X., Henninger, J. E., Platt, J. M., Oksuz, O., Zamudio, A. V., Afeyan, L. K., et al. (2020). MeCP2 links heterochromatin condensates and neurodevelopmental disease. *Nature*, 586(7829):440–444.
- Li, Y., Liu, M., Chen, L.-F., and Chen, R. (2018). P-TEFb: Finding its ways to release promoter-proximally paused RNA polymerase II. *Transcription*, 9(2):88–94.
- Li, Y., Zhang, C., Bell, E. W., Yu, D.-J., and Zhang, Y. (2019). Ensembling multiple raw coevolutionary features with deep residual neural networks for contact-map prediction in CASP13. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1082–1091.
- Lieberman-Aiden, E., Van Berkum, N. L., Williams, L., Imakaev, M., Ragooczy, T., Telling, A., Amit, I., Lajoie, B. R., Sabo, P. J., Dorschner, M. O., et al. (2009). Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *science*, 326(5950):289–293.
- Lin, Y., Currie, S. L., and Rosen, M. K. (2017). Intrinsically disordered sequences enable modulation of protein phase separation through distributed tyrosine motifs. *Journal of Biological Chemistry*, 292(46):19110–19120.
- Lockless, S. W. and Ranganathan, R. (1999). Evolutionarily conserved pathways of energetic connectivity in protein families. *Science*, 286(5438):295–299.
- Luger, K., Mäder, A. W., Richmond, R. K., Sargent, D. F., and Richmond, T. J. (1997). Crystal structure of the nucleosome core particle at 2.8 Å resolution. *Nature*, 389(6648):251–260.
- Lumry, R. and Eyring, H. (1954). Conformation changes of proteins. *The Journal of physical chemistry*, 58(2):110–120.
- Lupas, A. and Koretke, K. (2008). Evolution of protein folds. In *Computational structural biology: methods and applications*, pages 131–152. World Scientific Hackensack, NJ.
- Lupiáñez, D. G., Kraft, K., Heinrich, V., Krawitz, P., Brancati, F., Klopocki, E., Horn, D., Kayserili, H., Opitz, J. M., Laxova, R., et al. (2015). Disruptions of topological chromatin domains cause pathogenic rewiring of gene-enhancer interactions. *Cell*, 161(5):1012–1025.
- Luscombe, N. M., Laskowski, R. A., and Thornton, J. M. (2001). Amino acid–base interactions: a three-dimensional analysis of protein–DNA interactions at an atomic level. *Nucleic acids research*, 29(13):2860–2874.
- Magnani, L., Eeckhoutte, J., and Lupien, M. (2011). Pioneer factors: directing transcriptional regulators within the chromatin environment. *Trends in Genetics*, 27(11):465–474.
- Malik, S. and Roeder, R. G. (2010). The metazoan Mediator co-activator complex as an integrative hub for transcriptional regulation. *Nature Reviews Genetics*, 11(11):761–772.
- Markwick, P. R., Malliavin, T., and Nilges, M. (2008). Structural biology by NMR: structure, dynamics, and interactions. *PLoS computational biology*, 4(9):e1000168.

- Martin, L., Gloor, G. B., Dunn, S., and Wahl, L. M. (2005). Using information theory to search for co-evolving residues in proteins. *Bioinformatics*, 21(22):4116–4124.
- Merkenschlager, M. and Nora, E. P. (2016). CTCF and cohesin in genome folding and transcriptional gene regulation. *Annual review of genomics and human genetics*, 17:17–43.
- MISTIC web server (2021). <http://mistic.leloir.org.ar/>. [Online; accessed 17-June-2021].
- Monastyrskyy, B., D’Andrea, D., Fidelis, K., Tramontano, A., and Kryshchak, A. (2016). New encouraging developments in contact prediction: Assessment of the CASP 11 results. *Proteins: Structure, Function, and Bioinformatics*, 84:131–144.
- Montelione, G. T. (2012). The Protein Structure Initiative: achievements and visions for the future. *F1000 biology reports*, 4.
- Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301.
- Moult, J., Pedersen, J. T., Judson, R., and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods.
- Najafabadi, H. S., Mnaimneh, S., Schmitges, F. W., Garton, M., Lam, K. N., Yang, A., Albu, M., Weirauch, M. T., Radovani, E., Kim, P. M., et al. (2015). C2H2 zinc finger proteins greatly expand the human regulatory lexicon. *Nature biotechnology*, 33(5):555–562.
- Nakane, T., Kotecha, A., Sente, A., McMullan, G., Masiulis, S., Brown, P. M., Grigoras, I. T., Malinauskaitė, L., Malinauskas, T., Miehling, J., et al. (2020). Single-particle cryo-EM at atomic resolution. *Nature*, 587(7832):152–156.
- Neher, E. (1994). How frequent are correlated changes in families of protein sequences? *Proceedings of the National Academy of Sciences*, 91(1):98–102.
- O’Flanagan, R. A., Paillard, G., Lavery, R., and Sengupta, A. M. (2005). Non-additivity in protein–DNA binding. *Bioinformatics*, 21(10):2254–2263.
- Olmea, O., Rost, B., and Valencia, A. (1999). Effective use of sequence correlation and conservation in fold recognition. *Journal of molecular biology*, 293(5):1221–1239.
- OpenStax CNX (2019). <https://bio.libretexts.org/@go/page/23733>. [Online; accessed 17-June-2021].
- Orengo, C. A., Jones, D. T., and Thornton, J. M. (1994). Protein superfamilies and domain superfolds. *Nature*, 372(6507):631–634.
- Orphanides, G., Lagrange, T., and Reinberg, D. (1996). The general transcription factors of RNA polymerase II. *Genes & development*, 10(21):2657–2683.
- Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G. A., Kim, D. E., Kamisetty, H., Kyriakidis, N. C., and Baker, D. (2017). Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298.
- Pang, B. and Snyder, M. P. (2020). Systematic identification of silencers in human cells. *Nature genetics*, 52(3):254–263.

- Park, D., Lee, Y., Bhupindersingh, G., and Iyer, V. R. (2013). Widespread misinterpretable ChIP-seq bias in yeast. *PloS one*, 8(12):e83506.
- Perales, R. and Bentley, D. (2009). Cotranscriptionality: the transcription elongation complex as a nexus for nuclear transactions. *Molecular cell*, 36(2):178–191.
- Petrenko, N., Jin, Y., Wong, K. H., and Struhl, K. (2016). Mediator undergoes a compositional change during transcriptional activation. *Molecular cell*, 64(3):443–454.
- Phatnani, H. P. and Greenleaf, A. L. (2006). Phosphorylation and functions of the RNA polymerase II CTD. *Genes & development*, 20(21):2922–2936.
- Plys, A. J., Davis, C. P., Kim, J., Rizki, G., Keenen, M. M., Marr, S. K., and Kingston, R. E. (2019). Phase separation of Polycomb-repressive complex 1 is governed by a charged disordered region of CBX2. *Genes & development*, 33(13-14):799–813.
- Porrua, O. and Libri, D. (2015). Transcription termination and the control of the transcriptome: why, where and how to stop. *Nature reviews Molecular cell biology*, 16(3):190–202.
- Price, M. N., Dehal, P. S., and Arkin, A. P. (2010). FastTree 2—approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490.
- Protein Structure Prediction Center (2020). CASP14 results.
- Pugacheva, E. M., Kubo, N., Loukinov, D., Tajmul, M., Kang, S., Kovalchuk, A. L., Strunnikov, A. V., Zentner, G. E., Ren, B., and Lobanenko, V. V. (2020). CTCF mediates chromatin looping via N-terminal domain-dependent cohesin retention. *Proceedings of the National Academy of Sciences*, 117(4).
- Rajagopal, N., Srinivasan, S., Kooshesh, K., Guo, Y., Edwards, M. D., Banerjee, B., Syed, T., Emons, B. J., Gifford, D. K., and Sherwood, R. I. (2016). High-throughput mapping of regulatory DNA. *Nature biotechnology*, 34(2):167–174.
- Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7(1):95–99.
- Raman, S., Vernon, R., Thompson, J., Tyka, M., Sadreyev, R., Pei, J., Kim, D., Kellogg, E., DiMaio, F., Lange, O., et al. (2009). Structure prediction for CASP8 with all-atom refinement using Rosetta. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):89–99.
- Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T., and Rives, A. (2021). MSA Transformer. *bioRxiv*.
- Rao, S. S., Huntley, M. H., Durand, N. C., Stamenova, E. K., Bochkov, I. D., Robinson, J. T., Sanborn, A. L., Machol, I., Omer, A. D., Lander, E. S., et al. (2014). A 3D map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680.
- Rawat, P., Boehning, M., Hummel, B., Aprile-Garcia, F., Pandit, A. S., Eisenhardt, N., Khavaran, A., Niskanen, E., Vos, S. M., Palvimo, J. J., et al. (2021). Stress-induced nuclear condensation of NELF drives transcriptional downregulation. *Molecular cell*, 81(5):1013–1026.
- Reeves, R. (1984). Transcriptionally active chromatin. *Biochimica et Biophysica Acta (BBA)-Gene Structure and Expression*, 782(4):343–393.

- Reiter, F., Wienerroither, S., and Stark, A. (2017). Combinatorial function of transcription factors and cofactors. *Current opinion in genetics & development*, 43:73–81.
- Remmert, M., Biegert, A., Hauser, A., and Söding, J. (2012). HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods*, 9(2):173–175.
- Remmert, M., Biegert, A., Linke, D., Lupas, A. N., and Söding, J. (2010). Evolution of outer membrane β -barrels from an ancestral $\beta\beta$ hairpin. *Molecular biology and evolution*, 27(6):1348–1358.
- Riley, M. and Labedan, B. (1997). Protein evolution viewed through Escherichia coli protein sequences: introducing the notion of a structural segment of homology, the module. *Journal of molecular biology*, 268(5):857–868.
- Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, page 622803.
- Rodriguez Horta, E., Barrat-Charlaix, P., and Weigt, M. (2019). Toward inferring Potts models for phylogenetically correlated sequence data. *Entropy*, 21(11):1090.
- Roeder, R. (1998). Role of general and gene-specific cofactors in the regulation of eukaryotic transcription. In *Cold Spring Harbor symposia on quantitative biology*, volume 63, pages 201–218. Cold Spring Harbor Laboratory Press.
- Roeder, R. G. (1996). The role of general initiation factors in transcription by RNA polymerase II. *Trends in biochemical sciences*, 21(9):327–335.
- Rohs, R., Jin, X., West, S. M., Joshi, R., Honig, B., and Mann, R. S. (2010). Origins of specificity in protein-DNA recognition. *Annual review of biochemistry*, 79:233–269.
- Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94.
- Ruan, S., Swamidass, S. J., and Stormo, G. D. (2017). BEESEM: estimation of binding energy models using HT-SELEX data. *Bioinformatics*, 33(15):2288–2295.
- Rube, H. T., Rastogi, C., Kribelbauer, J. F., and Bussemaker, H. J. (2018). A unified approach for quantifying and interpreting DNA shape readout by transcription factors. *Molecular systems biology*, 14(2):e7902.
- Sabari, B. R., Dall’Agnese, A., Boija, A., Klein, I. A., Coffey, E. L., Shrinivas, K., Abraham, B. J., Hannett, N. M., Zamudio, A. V., Manteiga, J. C., et al. (2018). Coactivator condensation at super-enhancers links phase separation and gene control. *Science*, 361(6400).
- Sabari, B. R., Dall’Agnese, A., and Young, R. A. (2020). Biomolecular condensates in the nucleus. *Trends in biochemical sciences*.
- Sakamoto, T., Ennifar, E., and Nakamura, Y. (2018). Thermodynamic study of aptamers binding to their target proteins. *Biochimie*, 145:91–97.
- Sanchez, A. and Golding, I. (2013). Genetic determinants and cellular constraints in noisy gene expression. *Science*, 342(6163):1188–1193.

- Sanulli, S., Trnka, M., Dharmarajan, V., Tibble, R., Pascal, B., Burlingame, A., Griffin, P., Gross, J., and Narlikar, G. (2019). HP1 reshapes nucleosome core to promote phase separation of heterochromatin. *Nature*, 575(7782):390–394.
- Schaarschmidt, J., Monastyrskyy, B., Kryshchak, A., and Bonvin, A. M. (2018). Assessment of contact predictions in CASP12: co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*, 86:51–66.
- Schaeffer, R. D. and Daggett, V. (2011). Protein folds and protein folding. *Protein Engineering, Design & Selection*, 24(1-2):11–19.
- Scruggs, B. S., Gilchrist, D. A., Nechaev, S., Muse, G. W., Burkholder, A., Fargo, D. C., and Adelman, K. (2015). Bidirectional transcription arises from two distinct hubs of transcription factor binding and active chromatin. *Molecular cell*, 58(6):1101–1112.
- Seemayer, S., Gruber, M., and Söding, J. (2014). CCMpredfast and precise prediction of protein residue–residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130.
- Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Židek, A., Nelson, A. W., Bridgland, A., et al. (2019). Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins: Structure, Function, and Bioinformatics*, 87(12):1141–1148.
- Shindyalov, I., Kolchanov, N., and Sander, C. (1994). Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering, Design and Selection*, 7(3):349–358.
- Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.
- Siddharthan, R. (2010). Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PloS one*, 5(3):e9722.
- Siebert, M. and Söding, J. (2016). Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences. *Nucleic acids research*, 44(13):6055–6069.
- Siggers, T. and Gordan, R. (2014). Protein–DNA binding: complexities and multi-protein codes. *Nucleic acids research*, 42(4):2099–2111.
- Sillitoe, I., Dawson, N., Lewis, T. E., Das, S., Lees, J. G., Ashford, P., Tolulope, A., Scholes, H. M., Senatorov, I., Bujan, A., et al. (2019). CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic acids research*, 47(D1):D280–D284.
- Simons, K. T., Kooperberg, C., Huang, E., and Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of molecular biology*, 268(1):209–225.
- Simons, K. T., Strauss, C., and Baker, D. (2001). Prospects for ab initio protein structural genomics. *Journal of molecular biology*, 306(5):1191–1199.
- Sims, R. J., Belotserkovskaya, R., and Reinberg, D. (2004). Elongation by RNA polymerase II: the short and long of it. *Genes & development*, 18(20):2437–2468.

- Skwark, M. J., Abdel-Rehim, A., and Elofsson, A. (2013). PconsC: combination of direct information methods and alignments improves contact prediction. *Bioinformatics*, 29(14):1815–1816.
- Skwark, M. J., Raimondi, D., Michel, M., and Elofsson, A. (2014). Improved contact predictions using the recognition of protein like contact patterns. *PLoS Comput Biol*, 10(11):e1003889.
- Smale, S. T. and Kadonaga, J. T. (2003). The RNA polymerase II core promoter. *Annual review of biochemistry*, 72(1):449–479.
- Söding, J. (2005). Protein homology detection by HMM–HMM comparison. *Bioinformatics*, 21(7):951–960.
- Söding, J., Biegert, A., and Lupas, A. N. (2005). The HHpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, 33(suppl_2):W244–W248.
- Spector, D. L. and Lamond, A. I. (2011). Nuclear speckles. *Cold Spring Harbor perspectives in biology*, 3(2):a000646.
- Splinter, E., Heath, H., Kooren, J., Palstra, R.-J., Klous, P., Grosveld, F., Galjart, N., and de Laat, W. (2006). CTCF mediates long-range chromatin looping and local histone modification in the β -globin locus. *Genes & development*, 20(17):2349–2354.
- Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313.
- Steinegger, M., Mirdita, M., and Söding, J. (2019). Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature methods*, 16(7):603–606.
- Stormo, G. D. (2000). DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23.
- Stormo, G. D., Schneider, T. D., Gold, L., and Ehrenfeucht, A. (1982). Use of the Perceptron algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic acids research*, 10(9):2997–3011.
- Strom, A. R., Emelyanov, A. V., Mir, M., Fyodorov, D. V., Darzacq, X., and Karpen, G. H. (2017). Phase separation drives heterochromatin domain formation. *Nature*, 547(7662):241–245.
- Taylor, W. R. and Hatrick, K. (1994). Compensating changes in protein multiple sequence alignments. *Protein Engineering, Design and Selection*, 7(3):341–348.
- The Protein Data Bank (2021). PDB Statistics.
- The UniProt Consortium (2021). UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489.
- Thiel, G., Lietz, M., and Hohl, M. (2004). How mammalian transcriptional repressors work. *European Journal of Biochemistry*, 271(14):2855–2862.
- Thomas, M. C. and Chiang, C.-M. (2006). The general transcription machinery and general cofactors. *Critical reviews in biochemistry and molecular biology*, 41(3):105–178.
- Thurman, R. E., Rynes, E., Humbert, R., Vierstra, J., Maurano, M. T., Haugen, E., Sheffield, N. C., Stergachis, A. B., Wang, H., Vernot, B., et al. (2012). The accessible chromatin landscape of the human genome. *Nature*, 489(7414):75–82.

- Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071.
- Vaquerizas, J. M., Kummerfeld, S. K., Teichmann, S. A., and Luscombe, N. M. (2009). A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, 10(4):252–263.
- Vendruscolo, M., Kussell, E., and Domany, E. (1997). Recovery of protein structure from contact maps. *Folding and Design*, 2(5):295–306.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- Vorberg, S., Seemayer, S., and Söding, J. (2018). Synthetic protein alignments by CCMgen quantify noise in residue-residue contact prediction. *PLoS computational biology*, 14(11):e1006526.
- Vos, S. M., Farnung, L., Urlaub, H., and Cramer, P. (2018). Structure of paused transcription complex Pol II–DSIF–NELF. *Nature*, 560(7720):601–606.
- Wagschal, A., Rousset, E., Basavarajaiah, P., Contreras, X., Harwig, A., Laurent-Chabalier, S., Nakamura, M., Chen, X., Zhang, K., Meziane, O., et al. (2012). Microprocessor, Setx, Xrn2, and Rrp6 co-operate to induce premature termination of transcription by RNAPII. *Cell*, 150(6):1147–1157.
- Wang, H., Maurano, M. T., Qu, H., Varley, K. E., Gertz, J., Pauli, F., Lee, K., Canfield, T., Weaver, M., Sandstrom, R., et al. (2012). Widespread plasticity in CTCF occupancy linked to DNA methylation. *Genome research*, 22(9):1680–1688.
- Wang, L., Gao, Y., Zheng, X., Liu, C., Dong, S., Li, R., Zhang, G., Wei, Y., Qu, H., Li, Y., et al. (2019). Histone modifications regulate chromatin compartmentalization by contributing to a phase separation mechanism. *Molecular cell*, 76(4):646–659.
- Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1):e1005324.
- Wasserman, W. W. and Sandelin, A. (2004). Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, 5(4):276–287.
- Waterhouse, A., Bertoni, M., Bienert, S., Studer, G., Tauriello, G., Gumienny, R., Heer, F. T., de Beer, T. A. P., Rempfer, C., Bordoli, L., et al. (2018). SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic acids research*, 46(W1):W296–W303.
- Webb, B. and Sali, A. (2016). Comparative protein structure modeling using MODELLER. *Current protocols in bioinformatics*, 54(1):5–6.
- Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72.

- Weirauch, M. T., Cote, A., Norel, R., Annala, M., Zhao, Y., Riley, T. R., Saez-Rodriguez, J., Cokelaer, T., Vedenko, A., Talukder, S., et al. (2013). Evaluation of methods for modeling transcription factor sequence specificity. *Nature biotechnology*, 31(2):126–134.
- Weirauch, M. T. and Hughes, T. (2011). A catalogue of eukaryotic transcription factor types, their evolutionary origin, and species distribution. In *A handbook of transcription factors*, pages 25–73. Springer.
- Weiss, M. C., Sousa, F. L., Mrnjavac, N., Neukirchen, S., Roettger, M., Nelson-Sathi, S., and Martin, W. F. (2016). The physiology and habitat of the last universal common ancestor. *Nature microbiology*, 1(9):1–8.
- Wetlaufer, D. B. (1973). Nucleation, rapid folding, and globular intrachain regions in proteins. *Proceedings of the National Academy of Sciences*, 70(3):697–701.
- Wollenberg, K. R. and Atchley, W. R. (2000). Separation of phylogenetic and functional associations in biological sequences by using the parametric bootstrap. *Proceedings of the National Academy of Sciences*, 97(7):3288–3291.
- Wong, K. H., Jin, Y., and Struhl, K. (2014). TFIIF phosphorylation of the Pol II CTD stimulates mediator dissociation from the preinitiation complex and promoter escape. *Molecular cell*, 54(4):601–612.
- Woolfson, D. N., Bartlett, G. J., Burton, A. J., Heal, J. W., Niitsu, A., Thomson, A. R., and Wood, C. W. (2015). De novo protein design: how do we expand into the universe of possible protein structures? *Current opinion in structural biology*, 33:16–26.
- Wyckoff, H. (1968). The compensating nature of substitutions in pancreatic ribonucleases. In *Brookhaven Symp Biol*, volume 21, pages 252–258.
- Xu, J. (2019). Distance-based protein folding powered by deep learning. *Proceedings of the National Academy of Sciences*, 116(34):16856–16865.
- Yamaguchi, Y., Takagi, T., Wada, T., Yano, K., Furuya, A., Sugimoto, S., Hasegawa, J., and Handa, H. (1999). NELF, a multisubunit complex containing RD, cooperates with DSIF to repress RNA polymerase II elongation. *Cell*, 97(1):41–51.
- Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., and Baker, D. (2020). Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503.
- Yee, A. A., Savchenko, A., Ignachenko, A., Lukin, J., Xu, X., Skarina, T., Evdokimova, E., Liu, C. S., Semesi, A., Guido, V., et al. (2005). NMR and X-ray crystallography, complementary tools in structural proteomics of small proteins. *Journal of the American Chemical Society*, 127(47):16512–16517.
- Yesudhas, D., Batool, M., Anwar, M. A., Panneerselvam, S., and Choi, S. (2017). Proteins recognizing DNA: Structural uniqueness and versatility of DNA-binding domains in stem cell transcription factors. *Genes*, 8(8):192.
- Yevshin, I., Sharipov, R., Kolmykov, S., Kondrakhin, Y., and Kolpakov, F. (2019). GTRD: a database on gene transcription regulation 2019 update. *Nucleic acids research*, 47(D1):D100–D105.

- Yip, K. M., Fischer, N., Paknia, E., Chari, A., and Stark, H. (2020). Atomic-resolution protein structure determination by cryo-EM. *Nature*, 587(7832):157–161.
- Zaret, K. S. and Carroll, J. S. (2011). Pioneer transcription factors: establishing competence for gene expression. *Genes & development*, 25(21):2227–2241.
- Zeitlinger, J. (2020). Seven myths of how transcription factors read the cis-regulatory code. *Current Opinion in Systems Biology*.
- Zeng, H., Edwards, M. D., Liu, G., and Gifford, D. K. (2016). Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, 32(12):i121–i127.
- Zhao, Y., Ruan, S., Pandey, M., and Stormo, G. D. (2012). Improved models for transcription factor binding site identification using nonindependent interactions. *Genetics*, 191(3):781–790.
- Zhao, Y. and Stormo, G. D. (2011). Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature biotechnology*, 29(6):480–483.
- Zhu, F., Farnung, L., Kaasinen, E., Sahu, B., Yin, Y., Wei, B., Dodonova, S. O., Nitta, K. R., Morgunova, E., Taipale, M., et al. (2018). The interaction landscape between transcription factors and the nucleosome. *Nature*, 562(7725):76–81.
- Zink, D., Cremer, T., Saffrich, R., Fischer, R., Trendelenburg, M. F., Ansorge, W., and Stelzer, E. H. (1998). Structure and dynamics of human interphase chromosome territories in vivo. *Human genetics*, 102(2):241–251.
- Zuckerandl, E. (1976). Evolutionary processes and evolutionary noise at the molecular level. *Journal of molecular evolution*, 7(4):269–311.