Comparison of homologous protein sequences using direct coupling information by pairwise Potts model alignments

Hugo Talibart

DOCTORAT BRETAGNE LOIRE MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Hugo TALIBART

Comparison of homologous protein sequences using direct coupling information by pairwise Potts model alignments

Thèse présentée et soutenue à Inria Rennes, le 24 février 2021
Unité de recherche : IRISA
Thèse N° :

Rapporteurs avant soutenance :

Sean EDDY Professeur à Harvard University, Cambridge, USA
Martin WEIGT Professeur à Sorbonne Université, Paris

Composition du Jury :

Attention, en cas d'absence d'un des membres du jury le jour de la soutenance, la composition du jury doit être revue pour s'assurer qu'elle est conforme et devra être répercutée sur la couverture de thèse

Président :	Guillaume GRAVIER	Directeur de recherche CNRS, Rennes
Examinateurs :	Sean EDDY	Professeur à Harvard University, Cambridge, USA
	Guillaume GRAVIER	Directeur de recherche CNRS, Rennes
	Juliette MARTIN	Chargée de recherche CNRS, Lyon
	Thomas SCHIEX	Directeur de recherche INRAE, Toulouse
	Martin WEIGT	Professeur à Sorbonne Université, Paris
Dir. de thèse :	Jacques NICOLAS	Directeur de recherche Inria, Rennes
Encadr. de thèse :	François COSTE	Chargé de recherche Inria, Rennes

Nothing whets the intelligence more than a passionate suspicion, nothing develops all the faculties of an immature mind more than a trail running away into the dark.

STEFAN ZWEIG

Acknowledgements

First of all, I want to sincerely thank Sean Eddy and Martin Weigt for reviewing this manuscript, and Guillaume Gravier, Juliette Martin and Thomas Schiex for agreeing to be part of the jury, it is a real honour and pleasure.

Formally but gratefully, I would like to acknowledge the support of Université de Rennes 1 and IRISA lab, who made it possible for me to carry out this PhD.

Unsurprisingly, my deepest gratitude goes to my main supervisor François Coste. François, thank you for having agreed to go on this three-year adventure with me even though at the time I was a complete stranger interested in your work. Thank you for your trust, your patience, thank you for all this time you dedicated to me, for everything you taught me, for reviewing my work with such attention and care to important details, for all these brief discussions that turned into hours-long brainstorming sessions. Thank you for your support and your sense of humor, I was very lucky to carry out this PhD under your supervision.

I also wish to warmly thank Jacques Nicolas for agreeing to be my official thesis supervisor, and for his valuable guidance and his kindness.

Incidentally, I want to extend my sincere thanks to members of my PhD follow-up committee, Annie Foret and Jean-François Gibrat, for their helpful advice.

I would also like to express my appreciation to Rumen Andonov and Inken Wohlers for their help and for providing me with the source code of their solver, Mathilde Carpentier particularly for her advice on protein structure alignment databases and the helpful scripts she provided, and Susann Bader for helpful discussion on CCMpredPy.

Many thanks to the GenOuest Bioinformatics platform as well: virtually all computations in this thesis were performed using the computing resources they provide, efficient and easy to use, with a very responsive and helpful support

team.

Then I want to express how pleased I was to be a part of Dyliss team and Symbiose superteam, and I would like to thank each and every member for such a pleasant atmosphere of mutual assistance and of course traditional coffee breaks. More specifically, I would like to thank the former and the current leaders of Dyliss team, Anne Siegel and Olivier Dameron, for their dedication to the well-being and intellectual growth of their team members, especially when it comes to PhD students. I also owe an important debt to Marie Le Roïc, the Snow White to her Seven Dwarves, who takes care of so much complicated administrative work we don't understand, and brightens our days with her contagious smile. Special thanks to my fellow past and current precarious co-workers, for these lunches where crazy ideas such as writing a paper on how much Kiri cheese is needed to uniformly cover a giraffe naturally arise, for the pub experiences, the raclette evenings, for the laughs and the mutual support: Anne, Arnaud, Célia, Céline, Cervin, Clara, Clémence, Grégoire, Kévin, Lolita, Lucas, Maël, Marie, Marine, Maxime, Méline, Méziane, Nicolas, Olivier, Téo, Xavier, and a special mention to Chloé and Wesley, who had to bear with me and my aversion to heat on a daily basis. Many thanks to Guillaume as well for a very fun introductory course to Batucada.

Many thanks also to the organizing team of the *Sciences en Cour[t]s* festival, in 2018 since it allowed me to sit back and reflect upon my PhD subject with this video I had much fun making, but also to those who kept the festival alive in subsequent years.

On a more personal level, I would like to thank my family for supporting me, as well as my friends. Among them, I will more specifically mention Céline, Clara, Élise, Jim, Manon, Mélanie, Naomi and Olivier, because they probably played the most important role in the completion of this PhD, for various reasons.

I also want to thank coaches from Élancia Rennes Maginot, working out in this gym was the best way to start the day and enter a calm mental state before addressing my research challenges, and I hope you will be able to reopen very soon.

Thanks also to Alexandra Elbakyan for having created Sci-Hub, and to all these other things that were essential to me during these three years: coffee, Python, LaTeX and peanut butter.

And finally, I want to thank you, unknown reader randomly stumbling across

this acknowledgement page though we do not know each other. If you are a PhD student reading this for your background study, I hope this will be clear to you, and above all I hope you will enjoy your PhD time as much as I enjoyed mine.

Contents

List of Figures

List of Tables

Résumé de la thèse – Summary in french

Introduction

Grâce aux technologies de séquençage, de plus en plus de séquences de protéines sont disponibles, mais leur annotation demeure un important goulet d'étranglement. Prédire expérimentalement la fonction et la structure d'une protéine étant long et coûteux, les méthodes d'annotation *in-vivo* et *in-vitro* ne peuvent pas suivre le rythme auquel les séquences remplissent les bases de données. Cette situation appelle au développement de méthodes *in-silico*. L'approche la plus courante pour annoter une séquence protéique est de transférer des annotations depuis des protéines dites *homologues* : des protéines avec un ancêtre commun, partageant probablement des structures et des fonctions similaires. La détection d'homologie est rendue possible par le fait que la nécessité de maintenir la fonction et la structure contraint l'évolution des séquences de protéines, impliquant que des séquences homologues partagent des caractéristiques communes. Quand deux protéines ne sont pas trop distantes dans l'arbre de l'évolution, leurs séquences d'acides aminés sont assez similaires pour pouvoir inférer leur homologie en considérant simplement un score de similarité dérivé de l'alignement de leurs séquences. Mais plus deux homologues sont lointains, plus le nombre de mutations qui les séparent est important, diminuant leur identité de séquence et rendant plus difficile la prédiction de leur homologie à partir des séquences seules. Ainsi, pour détecter des homologies plus lointaines, plutôt que de considérer les séquences seules, une approche plus efficace est de modéliser les propriétés conservées et la

variabilité admissible d'ensembles de séquences homologues. Un score de similarité peut être obtenu en alignant une séquence à un modèle représentant un ensemble de protéines homologues, ou en alignant deux modèles. Aujourd'hui, l'état de l'art est représenté par des modèles statistiques appelés *profile Hidden Markov Models* (pHMMs) qui modélisent des ensembles de séquences par une succession d'états liés par des probabilités de transition, reflétant les probabilités de trouver les acides aminés aux différentes positions et les probabilités d'insertions et de délétions. Ces modèles sont aujourd'hui largement utilisés pour l'annotation et la classification de séquences. L'outil le plus populaire pour l'annotation fonctionnelle est probablement HMMER [FCE11], qui permet d'aligner une séquence à un pHMM. Il est au coeur de plusieurs bases de données classifiant des familles de protéines et domaines protéiques, telles que Pfam [SED97; ElG+19] et TIGRFAMs [HSW03]. Pour détecter une homologie plus lointaine, HHsuite [Söd05; Ste+19], basé sur les alignements pHMM-pHMM de son outil HHalign, permet une recherche plus sensible. Ces méthodes ont permis d'annoter un grand nombre de séquences alors que les modèles sont construits sur l'hypothèse simplificatrice selon laquelle les positions dans les séquences de protéines évoluent indépendamment les unes des autres. Pourtant, on sait que les résidus co-évoluent pour respecter des contraintes structurelles et fonctionnelles, où des *mutations compensatoires* peuvent compenser des mutations qui, seules, auraient été délétères, et cela n'est pas capturé par les pHMMs de part leur nature positionnelle.

Cette thèse constitue une première étape dans l'étude de la contribution des dépendances distantes entre positions dans l'alignement de séquences de protéines et la détection d'homologie. Nous proposons pour cela d'exploiter le modèle de Potts, un champ aléatoire de Markov dont l'intérêt a déjà été prouvé dans un autre contexte. Ce modèle, issu de la physique statistique, a été appliqué avec succès à la prédiction de co-évolution directe de résidus dans une méthode appelée Analyse en Couplage Directs (*Direct Coupling Analysis*), qui a permis une percée dans le domaine de la prédiction de contacts. Inférés sur un alignement multiple de séquences, ses paramètres reflètent à la fois la conservation positionnelle et les couplages directs entre les positions. Ce modèle, dérivé du principe de maximum d'entropie, génère de façon consistante les fréquences observées avec le moins de biais possible. Motivés par ces propriétés, dans cette thèse nous nous posons la

question suivante : *le modèle de Potts peut-il également améliorer la détection d'homologie par alignement de séquences ?*

Inspirés par le succès des méthodes d'alignement pHMM-pHMM pour la détection d'homologues lointains, nous nous intéressons ici au problème d'alignement de deux modèles de Potts, où les modèles représentent une séquence de protéine enrichie avec ses homologues proches. Notre approche est résumée dans la figure 1.

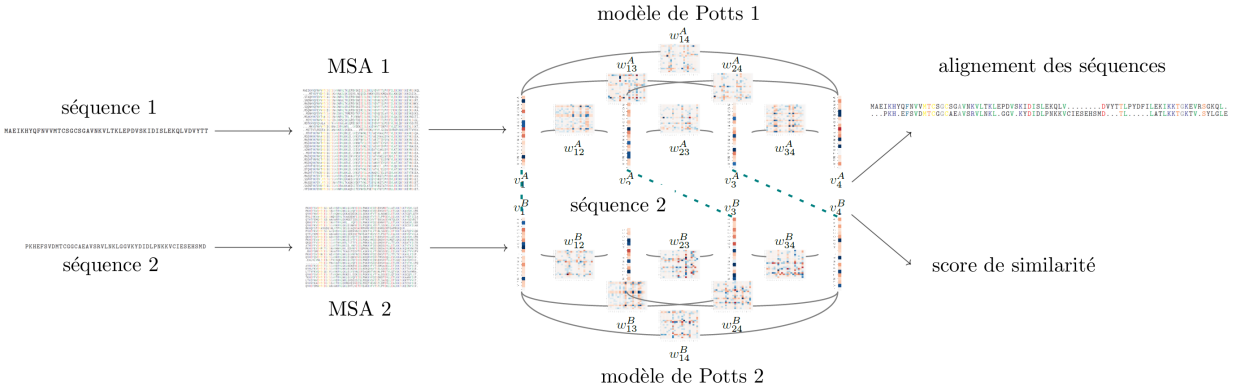


Figure 1 – Résumé de notre approche. Les séquences à aligner sont d'abord enrichies avec leurs homologues proches pour capturer la variabilité admissible autour d'elles et obtenir un signal de covariation. Des modèles de Potts sont ensuite inférés sur les alignements multiples (MSA) de ces homologues proches. Ces modèles sont ensuite alignés avec notre méthode pour obtenir un score de similarité et un alignement des séquences.

Notre contribution porte à la fois sur la construction et l'alignement de modèles de Potts, deux problèmes étroitement liés dont l'étude a nécessité de nombreux allers-retours entre les deux.

Vers la construction de modèles de Potts canoniques

Nous avons identifié le modèle de Potts introduit par la Direct Coupling Analysis comme étant un bon candidat pour la modélisation d'ensembles de séquences de protéines et la recherche d'homologues. En effet, il permet de modéliser

un alignement multiple de séquences avec des paramètres reflétant à la fois la conservation positionnelle et les couplages directs entre les positions(voir figure 2) : pour chaque position i, un vecteur v_i donne un poids réel pour chaque acide aminé, qui tend à être positif si l'acide aminé est particulièrement présent à la position et négatif s'il est particulièrement absent, et pour chaque paire de positions (i, j) une matrice w_{ij} donne un poids réel pour chaque paire d'acides aminés, reflétant les corrélations et les anti-corrélations. Étant donné qu'il dérive du principe de maximum d'entropie, ce modèle génère les fréquences observées avec le moins de biais possible.

Figure 2 – Les paramètres d'un modèle de Potts sont inférés sur un alignement multiple de séquences et reflètent la conservation positionnelle à chaque position (les vecteurs v_i, donnant un poids réel pour chaque lettre de l'alphabet, tendant à être positif lorsque la lettre apparaît fréquemment à la position et négatif lorsqu'elle est rarement vue à la position) et les couplages directs entre les positions (les matrices w_{ij}, donnant un poids réel pour chaque paire de lettres de l'alphabet, positif pour une corrélation entre les lettres et négatif pour une anti-corrélation.)

Cependant, les méthodes et *workflows* actuels pour l'inférence de modèles de Potts ont été optimisés dans un but de prédiction de contact et ne sont pas forcément adaptés à la comparaison de modèles de Potts dans un but de

détection d'homologie. Dans le chapitre 4, nous nous sommes attachés à définir ces nouveaux besoins, identifier les leviers d'action sur lesquels nous pouvons jouer pour rendre les modèles plus comparables, et nous avons proposé un premier *workflow* opérationnel se basant sur les méthodes existantes.

Pour construire un modèle de Potts représentant une séquence de protéine, nous enrichissons cette séquence en allant chercher ses homologues proches pour obtenir du signal de covariation. Nous utilisons pour cela HHblits [Rem+12] en nous basant sur des recommandations pour la prédiction de contact.

En théorie, l'unique modèle de Potts représentant un alignement multiple est obtenu en maximisant la vraisemblance des données, mais l'existence d'une constante de normalisation rend cette maximisation impraticable. Plusieurs méthodes ont donc été proposées pour obtenir des approximations en temps raisonnable. Nous basons ici notre *workflow* sur CCMpredPy [VSS18], une méthode basée sur la maximisation de la pseudo-vraisemblance – une approche considérée comme état de l'art pour la prédiction de contacts et présentant une complexité raisonnable – et offrant une option inédite que nous proposons comme standard pour la construction de modèles de Potts canoniques. Cette fonctionnalité est la possibilité d'initialiser l'inférence et centrer la régularisation autour d'un modèle de Potts sans couplage. Ce choix permet de placer le plus de poids possible sur les paramètres positionnels et de n'ajouter que les couplages nécessaires, ce qui rend le modèle plus interprétable et diminue a priori le temps de calcul pour l'alignement de deux modèles.

Nous avons identifié que des anti-corrélations fallacieuses pouvaient être engendrées par un simple manque de données, et que la comparabilité de deux modèles de Potts était compromise par la sensibilité de l'inférence aux variations d'échantillonnage. Nous estimons que la solution idéale serait l'utilisation de pseudo-comptes, comme largement utilisés par les pHMMs, mais aucune méthode permettant à la fois d'intégrer des pseudo-comptes et l'initialisation à un modèle de Potts sans couplage n'a été publiée à ce jour. Nous avons donc proposé des solutions opérationnelles pour rendre malgré tout les modèles plus comparables par un lissage des paramètres du modèle après inférence afin de se concentrer sur les valeurs positives les plus importantes.

Alignement optimal de modèles de Potts

Nous avons introduit une méthode d'alignement de modèles de Potts, que nous avons nommée PPalign (voir figure 3), détaillée dans le chapitre 5.

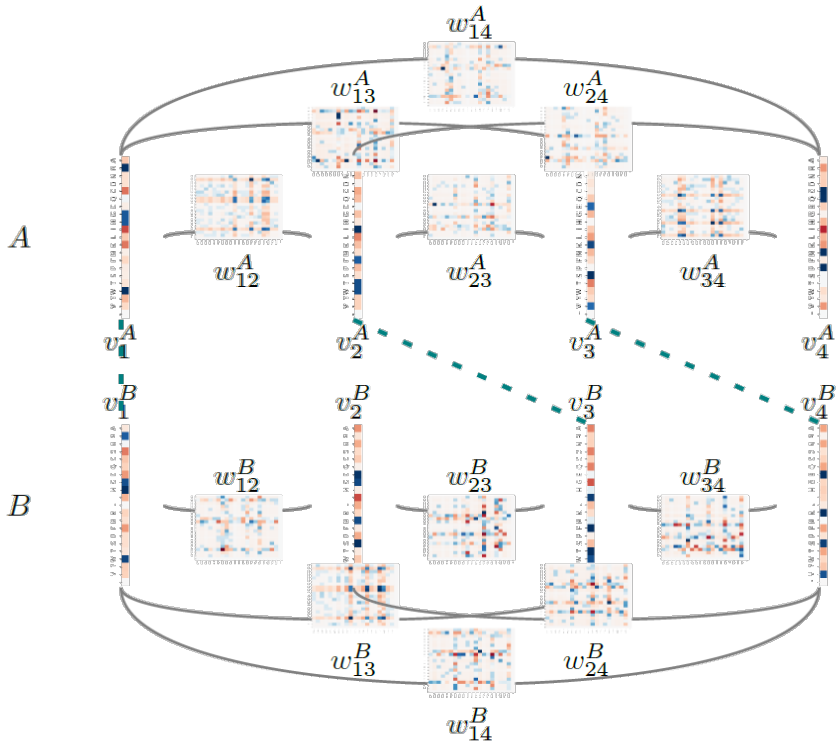


Figure 3 – Illustration de l'alignement de deux modèles de Potts A et B.

À cause des dépendances distantes, ce problème est NP-difficile et ne peut pas être résolu par programmation dynamique. Pour limiter les biais pouvant être causés par l'utilisation d'heuristiques, nous proposons une formulation comme problème de programmation linéaire en nombres entiers (Integer Linear Programming (ILP)) qui étend la formulation proposée par [WAK12; Woh12] pour le problème d'alignement de structures de protéines utilisant des matrices de distances inter-résidus en introduisant un score de similarité pour les positions en plus des paires de positions. Ce problème peut être résolu de façon optimale en utilisant leur solveur particulièrement efficace.

Cette formulation ILP est construite sur la maximisation d'une fonction de similarité entre deux modèles de Potts. Nous avons proposé une fonction basée sur le produit scalaire. Ce choix nous permet de rendre compte à la fois de la similarité des paramètres et de leur importance dans leurs modèles respectifs, et étend naturellement le problème de l'alignement d'une séquence à un modèle. Nous proposons d'ajouter à cette fonction de similarité la possibilité d'effectuer la comparaison par rapport à un modèle nul.

Nous avons validé notre méthode en examinant la qualité de ses alignements par rapport à des alignements de référence à faible identité de séquence extraits de la base de données d'alignements structuraux SISYPHUS [And+07], après un entraînement des hyperparamètres. Sur ces données, des solutions à un ϵ choisi près de la solution représentant l'alignement optimal ont été trouvées en un temps tractable avec 1 min 37 en moyenne, et la qualité de ces alignements était en moyenne meilleure que celle de la méthode HHalign basée sur l'alignement de pHMMs entraînés sur les mêmes données. Nous avons de plus pu montrer que les couplages pouvaient considérablement améliorer la qualité de certains alignements avec une faible identité de séquence.

Premières expériences de recherche d'homologues

Notre méthode d'alignement validée, nous avons réalisé en complément des expériences préliminaires appliquant notre méthode à la détection d'homologues, reportées dans le chapitre 6.

De premières expériences sur la détection d'homologie au niveau de familles de protéines réalisées plus tôt durant cette thèse montrent que, sur trois jeux de données pour des familles conservées, notre score est capable de discriminer parfaitement les membres de la famille des exemples négatifs. Ces résultats sont aussi atteints sans prendre en compte le score de couplage, arrivant ainsi à égalité avec HHalign. La contribution des couplages n'a pas pu être montrée pour cette expérience, traduisant la difficulté de construire un jeu de données difficile au niveau de la famille avec des exemples négatifs annotés.

À la fin de cette thèse, nous avons réalisé une expérience rapide au niveau du fold dans la classe des protéines *"all-beta"*, afin d'avoir une idée des performances de

notre méthode avec notre workflow actuel pour la détection d'homologues lointains, et avec la contrainte supplémentaire d'une limite de temps de 1 minute pour chaque alignement. Cette expérience a montré des résultats encourageants : notre méthode obtient en moyenne de meilleurs résultats qu'une autre méthode basée sur les champs aléatoires de Markov, MRFalign [Ma+14], et que HHalign entraîné sur les mêmes données. Nous constatons que nos plus mauvais résultats sont dûs à la contrainte de temps de 1 minute impliquant que la solution optimale n'était pas atteinte, ce qui suggère que ces résultats pourraient encore être améliorés avec un temps plus long.

Conclusion

Avec cette thèse, nous avons établi des bases pour le développement futur d'une méthode de recherche d'homologues basée sur l'alignement de modèles de Potts et pour de futures études sur les forces et faiblesses du modèle de Potts pour la détection d'homologie. Nous avons identifié de nouveaux besoins pour la construction de modèles de Potts comparables dans un but de détection d'homologie et nous avons proposé un premier *workflow* opérationnel implémentant des premiers choix vers un idéal de canonicité et une stratégie de lissage des paramètres rendant les modèles plus comparables. Nous avons développé une méthode d'alignement de modèles de Potts capable de donner la solution exacte du problème d'alignement à un epsilon près en un temps raisonnable. Cette méthode, ainsi que les différents outils que nous avons développés pour construire des modèles de Potts à partir d'une séquence et les visualiser, ont été mis à disposition dans un dépôt GitHub : `https://github.com/htalibart/ppsuite`. Nous avons construit un benchmark d'alignements de référence à faible identité de séquence (mis à disposition ici : `https://www-dyliss.irisa.fr/PPalign`) sur lequel nous avons validé notre méthode. Ces premiers résultats indiquent que les couplages directs peuvent considérablement améliorer la qualité de certains alignements avec une faible identité de séquence par rapport à la méthode HHalign d'alignement pHMM-pHMM et suggèrent que ces modèles pourraient améliorer la recherche d'homologues plus lointains, ce que semblent confirmer d'encourageants résultats préliminaires de détection d'homologie. Nous avons identifié des pistes

d'amélioration pour la construction de modèles de Potts plus comparables, un travail que nous jugeons prioritaire pour mieux représenter les protéines et effectuer des comparaisons plus sensibles. Notre méthode, dont l'optimalité est garantie, pourrait être un atout précieux pour des études non biaisées dans cette direction.

Introduction

Thanks to sequencing technologies, the number of protein sequences available is constantly increasing but their annotation remains a bottleneck. Experimentally predicting a protein's functions and shapes being costly and time-consuming, *in-vivo* and *in-vitro* approaches cannot keep up with the exponential pace at which protein sequences are filling data banks. This situation raises a need for *in-silico* methods. The most widely used approach to annotate a protein sequence is to transfer annotations from identified *homologs*: proteins with a common ancestor, likely to share similar structures and functions. Detecting homology is made possible by the fact that maintaining function and structure constrains the evolution of protein sequences, implying that homologous sequences share common features. When proteins are not too distant in the evolutionary tree, their amino acid sequences are similar enough to infer their homology by examining a similarity score derived from a simple pairwise sequence alignment. However, the more remote two homologs are, the more mutations separate them, lowering their pairwise sequence identity and making it difficult to assert their homology based on the two sequences only. To detect more remote homologs, rather than considering only single sequences, a successful approach is to model conserved features and allowed variability of whole sets of homologous sequences. A similarity score can be derived by aligning a query sequence to a model representing a set of homologous proteins, or aligning two models, enriching the query sequence as well as the target with close homologs to gain further sensitivity. State-of-the-art approaches rely on statistical models termed *profile Hidden Markov Models* which model probabilities of finding amino acids at conserved positions in multiple sequence alignments of considered homologous sequences. These methods made it possible to computationally annotate a large number of sequences whose

homologs could not be retrieved by pairwise sequence similarity only. These successes were achieved despite the fact that these models are built on the strong simplifying assumption that positions in protein sequences evolve independently. Yet, in practice, it is known that residues co-evolve to comply with structural and functional constraints, where so-called *compensatory mutations* make up for mutations that would have been deleterious on their own, and these features cannot be captured by profile Hidden Markov Models due to their inherently positional nature.

In this thesis we provide ground work for investigations on the contribution of pairwise dependencies in protein sequence alignment and homology detection by making use of a Markov Random Field which already proved its relevance in a different context: the Potts model. This model, originating from statistical physics, was successfully applied to the prediction of directly co-evolving residues in a method termed Direct Coupling Analysis, leading to a breakthrough in the field of contact and 3D structure prediction. Inferred on a multiple sequence alignment, its parameters reflect both positional conservation and direct couplings between positions. This model derives from the maximum entropy principle, guaranteeing that it consistently generates observed statistics with as little bias as possible. Driven by its compelling properties, in this thesis we raise the following question: in addition to its successful application to the prediction of contacts, protein-protein interactions and mutational effects, could the Potts model improve sequence alignment and homology detection as well?

Here, inspired by the success of pairwise profile Hidden Markov Model alignment methods in remote homology detection, we focus on the pairwise Potts model alignment problem, where models represent a protein sequence and its close homologs.

Before presenting our contributions, in the first part of this manuscript we provide background on the modeling of protein sequences. We recall first the relation between proteins and amino acid sequences and provide principles motivating alignment-based homology search before putting forth strengths of state-of-the-art approaches and identifying some of their limitations, which motivated us to represent proteins with Potts models and design a pairwise Potts model alignment method. The second part outlines our contributions.

During the development of this pairwise Potts model alignment method, we faced three main challenges. The first challenge relates to the question of representing proteins with Potts models with the aim of comparing them. Existing inference methods and workflows were designed for direct interaction prediction and do not necessarily comply with such requirements, and since our goal is the pairwise comparison of Potts models, our results highly depend on their ability to properly model proteins in a comparable way. We address these questions in chapter 4, identifying choices to be made towards canonical Potts models and describing our implemented solutions and open propositions for improvement. The following chapter covers the two other major challenges we faced regarding the design of a pairwise Potts model alignment method. The first one is the definition of a similarity score for the alignment of two Potts models: unlike profile Hidden Markov Models, Potts model parameters are not probabilities but real weights, making the design of an appropriate scoring scheme less trivial. We proposed a similarity score which naturally extends the case of sequence to model alignment. The other major challenge is to compute the best alignment: due to non-local dependencies, this NP-hard problem cannot be efficiently solved with dynamic programming. To avoid biases caused by the use of heuristics, we propose an Integer Linear Programming formulation for the pairwise Potts model alignment problem which can be optimally solved by an efficient solver. Our alignment method's performances were assessed on a selected set of reference alignments with low sequence identity. We showed that alignments can be solved to optimality up to a chosen epsilon in tractable time and that direct couplings could substantially improve the quality of some alignments with lowest sequence identity with respect to pairwise profile Hidden Markov Models alignment method, and should thus improve the detection of remote homologs, as suggested by encouraging preliminary results reported in the last chapter.

Part I

Background

This part provides background on the modeling of protein sequences. The first chapter provides a general introduction to proteins, outlining their diversity and the different ways to look at them, from a three-dimensional structure to a primary sequence provided as a text string thanks to DNA sequencing. The second chapter introduces the sequence annotation problem and reviews main homology search methods addressing it, based on alignments of positional models capturing residue conservation. The third chapter raises the question of the relevance of taking distant dependencies into account when performing homology search. Biology-based arguments suggesting their significance are provided and early methods modeling them with Markov Random Fields are reviewed. Finally, a specific Markov Random Field originating from the contact prediction field, known as the *Potts model*, is brought forward as a promising alternative to improve homology search by taking into account distant dependencies in proteins.

Chapter 1

Proteins: functions encoded by sequences

This chapter provides a general understanding of the objects we are interested in: proteins, the building blocks of life. We start by giving an overview of the multiple functions a protein can endorse before focusing on the protein object and the different ways to describe it.

1.1 Introduction to the world of proteins

A stereotypical sentence reflecting an amazing truth is that "proteins are the building blocks of life". In essence, proteins are chains composed of subunits called amino acids that fold into space, yet the diversity of the resulting folds and functions is astonishing. Proteins are key operators in every organism, holding multiple positions, from molecule transport to immune system, from DNA repair to muscle contraction. Their roles are chemical and mechanical, sometimes in motion, sometimes parts of static structural components, binding with specific molecules. In this section, we give a glimpse of the dazzling diversity of biological functions proteins can endorse before delving into a more detailed picture of the protein object itself and the different levels to which it can be conceptualized, from a primary amino acid sequence to a fully functional three-dimensional structure, including local substructures. For a deeper dive into the protein world, the reader

can refer to a reference book such as [Alb18].

1.1.1 A diversity of functions

To give a sense of the vertiginous protein universe, it is estimated that the human genome contains around 21000 protein-coding genes [Lan11]. Mechanisms known as *alternative splicing* allow one single gene to yield several proteins, implying that cells can produce even more proteins: estimations suggest around 100000 [NG10]. And these numbers only cover the human species. Proteins are ubiquitous in every life form as we know it: bacteria, elephants, mushrooms, spiders, trees, seaweed, viruses... Each species, with its own specific features, has its own set of proteins.

An important feature of proteins is their capacity to bind to other molecules. This binding can be very tight, or it can be brief and weak, in any case it is always highly specific. One of the first images that may come to mind is that of an antibody (see figure 1.1) binding to an antigen to neutralize an external pathogen.

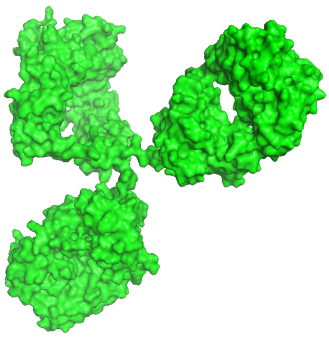


Figure 1.1 – 3D representation[1] of mouse immunoglobulin (antibody) (PDB 1IGT)

One can also think of messenger proteins, such as insulin, which binds to a transmembrane receptor (insulin receptor, another protein) to regulate glucose homeostasis (see figure 1.2).

[1] This figure, like all 3D representations of proteins in this manuscript, was rendered by PyMOL [Sch15].

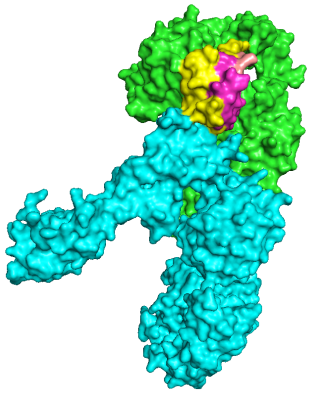


Figure 1.2 – 3D representation of human insulin (magenta and yellow) binding to an insulin receptor (blue and green). (PDB 6CE7)

Another classical example is enzymes, which form complexes to speed up reactions. For instance amylase (figure 1.3) catalyses the hydrolysis of starch into glucose, and luciferase catalyses a reaction responsible for light emission in fireflies. These complexes can involve other proteins that will maintain enzymes in close proximity and allow substrates to go from one active site to another.

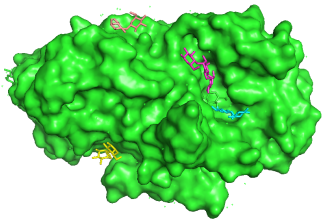


Figure 1.3 – 3D representation of pig pancreatic alpha-amylase (in green) in complex with oligosaccharides, which are simple sugar polymers (PDB 1PIG)

Besides, some proteins are structural components, literally building blocks for organisms. There are ordered fibrous proteins which can span a large distance such as α-keratin which is the primary component of hair, nails, claws and feathers (see figure 1.4), there are disordered proteins forming a loose, elastic material such as elastin which allows our tissues to stretch without tearing, there are proteins with a gel-like consistency such as nucleoporin, a constituent of nuclear pores, regulating

the transport of molecules across the nuclear envelope.

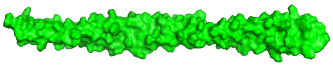


Figure 1.4 – 3D representation of α-keratin (PDB 6EC0)

Some proteins arrange into patterns and form virus capsids, coats of tubes and spheres binding DNA and RNA molecules.

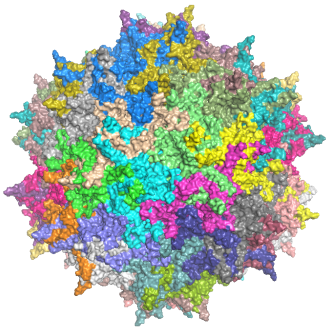


Figure 1.5 – Coat proteins (represented with different colors) assemble into a virus capsid, here of the human adeno-associated virus capsid isolat (PDB 6U3Q)

Proteins are a marvel of engineering, implied in an efficient (and silent) biological machinery: motors, switches, pumps... Myosin walks along actin filaments to allow muscles to contract. Kinesin drives chromosomes apart during mitosis. ATP-binding cassette transporters pump toxic molecules across membranes. Also worth mentioning are bacteria flagella, moving thanks to a full-fledged rotary engine entirely made of proteins.

Moreover, proteins can perform multiple functions: this is a phenomenon known as *protein moonlighting* [Jef99]. Some crystallin proteins, which fill lenses in our eyes and increase light refracting index, act as chaperones to prevent damaged proteins from aggregating into opaque complexes and forming cataracts and some of them show active enzyme activity in other tissues, like aldehyde dehydrogenase [Bat+03].

The diversity of protein functions and structures is dazzling. And yet, much is still to be discovered.

1.1.2 Ways of looking at the protein object

In this section, we take a closer look at the protein object, describe its chemical nature and lay out different levels to look at a protein, introducing the notion of primary, secondary and tertiary structures, and the notions of modularity and domains.

1.1.2.1 Macromolecules made of amino acids.

Proteins are macromolecules, that is to say that they are large molecules composed of smaller subunits. In the case of proteins, these subunits are *amino acids*.

Amino acids are organic molecules with a carboxyl and an amine functional group connected to a central carbon atom termed *alpha carbon* (C_α). They have the same overall structure, and differ from each other with a specific *side chain* which characterizes the amino acid. The first carbon atom of the side chain is termed *beta carbon* (C_β).

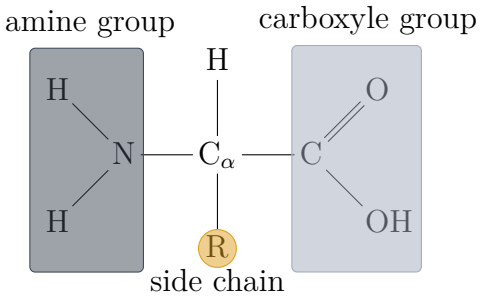


Figure 1.6 – Chemical formula of an amino acid. Every amino acid has an amine groupe, a carboxyle group, and a side chain (denoted R) which varies from one amino acid to another and make it unique.

In total, there are 20 different amino acids in the standard genetic code, which we recap in table 1.1 with their 3-letter and 1-letter codes.

name	**3-letter code**	**1-letter code**
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamic acid	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y

Table 1.1 – Recap of the 20 different amino acids with their 3-letter and 1-letter codes.

Side chains grant unique properties to amino acids, including:

- *Size*: amino acids can have side chains of variable sizes. Tryptophan for example is the bulkiest amino acid with its large aromatic side chain, while Glycine is the tiniest with a side chain consisting of a single hydrogen atom, providing substantial structural flexibility.

- *Affinity to water*: *hydrophobic* amino acids such as Valine tend to avoid water while *polar* amino acids such as Glutamine tend to interact with water.

In aqueous environments – where most proteins are found – hydrophobic amino acids engage in van der Waals interactions in the protein core, minimizing their contact with water and stabilizing the structure. In specific environments such as the lipid portion of a membrane, hydrophobic amino acids are rather found on the surface, interacting with lipid molecules, while hydrophilic amino acids engage in hydrogen bonds and create hydrophilic channels.

- *Electric charge*: while most amino acids are electrically neutral, Aspartic acid and Glutamic acid are negatively charged while Arginine, Histidine and Lysine are positively charged. Oppositely charged residues tend to form salt bridges contributing to the protein conformational stability.

- *Reactivity*: *aromatic* amino acids like Phenylalanine tend to participate in stacking reactions and often bind to other molecules while *aliphatic* amino acids like Alanine are highly non-reactive.

- *Ability to form disulfide bridges*: Cysteines can form disulfide bonds, which contribute to the stability of a protein structure or complex.

- *Geometry*: side chains have different orientation angles and space conformations. Proline in particular has a distinctive cyclic structure which favor tight turns in protein structures.

Though each amino acid is unique, they can be classified into overlapping sets sharing common physical, chemical and structural properties. A common classification is the Taylor classification [Tay86], displayed as a Venn diagram in figure 1.7.

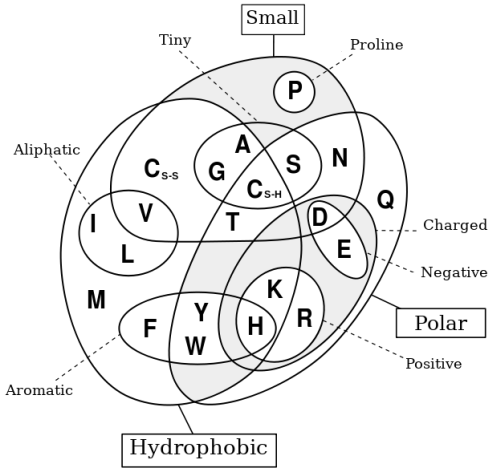


Figure 1.7 – Venn diagram of the 20 amino acids according to Taylor classification [Tay86], translated from [Wik12]

1.1.2.2 Primary structure: an amino acid chain.

A protein consists in a chain of these amino acids, linked together by covalent *peptide bonds* in a *polypeptide chain*, typically between 50 and 2000 long (see figure 1.8).

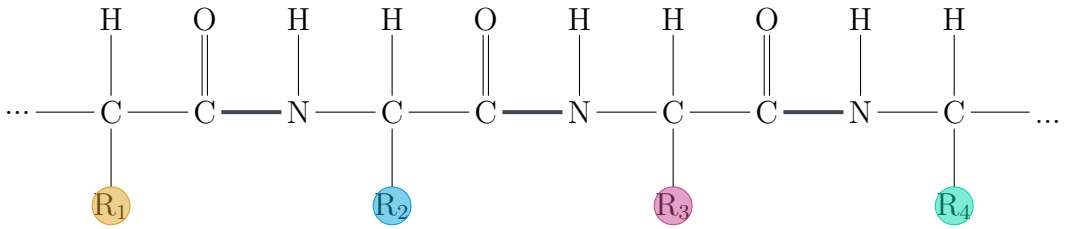


Figure 1.8 – Chemical formula of a polypeptide chain. Peptide bonds link amine and carboxyl groups, forming the so-called *backbone*, and side chains (R_1, R_2,...) are attached to it.

The linkage of the amine and carboxyl groups is referred to as the *backbone*,

to which side chains are attached. An amino acid embedded at a given position in a polypeptide chain is referred to as *residue*.

Having assigned a letter to each amino acid (see table 1.1), this primary chain is usually represented using a sequence of letters on the amino acid alphabet $\mathcal{A} = \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$. See for example the sequence of the Atx1 metallochaperone of Saccharomyces cerevisiae:

```
MAEIKHYQFNVVMTCSGCSGAVNKVLTKLEPDVSKIDISLEKQLVDVYTTLPYDFILEKIKKTGKEVRSGKQL
```

Figure 1.9 – Sequence of the Atx1 metallochaperone of Saccharomyces cerevisiae (PDB 1CC8)

1.1.2.3 Tertiary structure: a folded chain.

This primary chain folds into a three-dimensional conformation referred to as the *tertiary structure*, which one can define as the 3D structure defined by its atomic coordinates [Cle11].

A representation is provided figure 1.10 for the Atx1 metallochaperone protein of Saccharomyces cerevisiae, rendered by PyMOL [Sch15] from an atomic coordinate file provided by the Protein Data Bank [Ber+00] which lists the locations in space of atoms in the protein, derived from X-ray diffraction or NMR experiments.

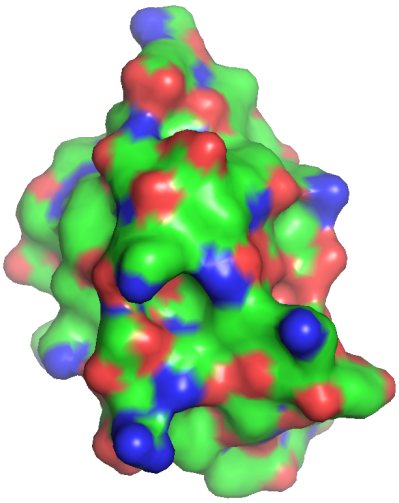


Figure 1.10 – Representation of the three-dimensional structure of the Atx1 metallochaperone protein of Saccharomyces cerevisiae (PDB identifier 1CC8)

1.1.2.4 Secondary structure: common local folding patterns.

When looking at the structures of most proteins, one will notice that they often locally fold into two intermediate regular patterns: *α-helices* and *β-sheets*, often represented by schematic "cartoon" representations (see figures 1.11 and 1.12). This prevalence can be explained by the fact that these folding patterns result from bonds between the amine and the carboxyle group in the backbone that do not involve the intrinsically heterogeneous side chains.

In α-helices, the backbone is bonded to itself, forming a rigid cylinder (see figure 1.11).

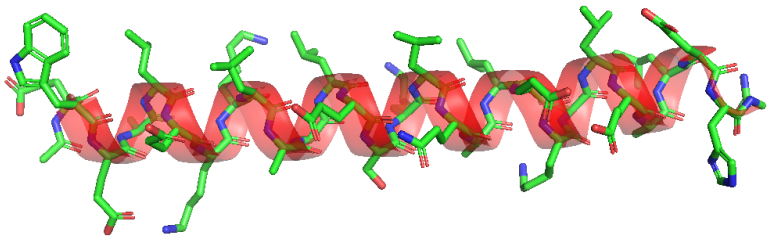


Figure 1.11 – An alpha helix. ("cartoon" representation in red)

Such structures are commonly found in cell membranes, since the hydrophile

backbone folded on itself is shielded against hydrophobic lipid environments. Wrapped around each other, they also form stable structures known as *coiled-coil structures*, leading to elongated proteins such as the previously introduced α-keratin.

β-sheets are structures formed by bonds between different segments of the chain (see figure 1.12).

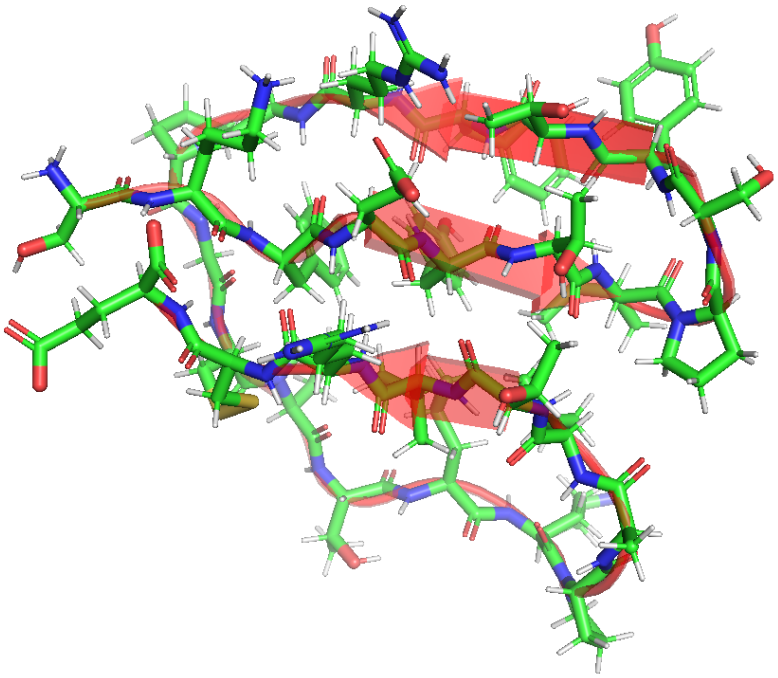
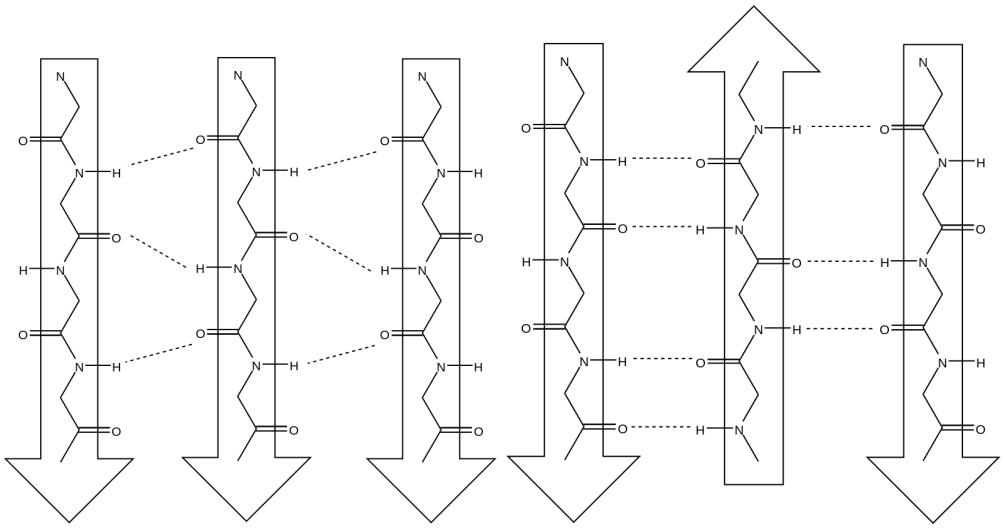


Figure 1.12 – A beta sheet. ("cartoon" representation in red)

Depending on the orientation, the arrangement can be *parallel* or *antiparallel* (see figure 1.13).



(a) Parallel beta sheet (b) Antiparallel beta sheet

Figure 1.13 – Representations of parallel (figure 1.13a) and antiparallel (figure 1.13b) beta sheets, extracted from [Com13]. Arrows indicate the direction of the polypeptide chain, from N-terminus to C-terminus.

Such rigid structures are often found in protein cores.

The sequences of α-helices and β-sheets, interspersed with regions termed *coil*, is referred to as the *secondary structure* of the protein. Secondary structures are typically represented by so-called *cartoon* (or *ribbon*) representations as in figure 1.14), probably one of the most common ways to visualize a protein structure nowadays.

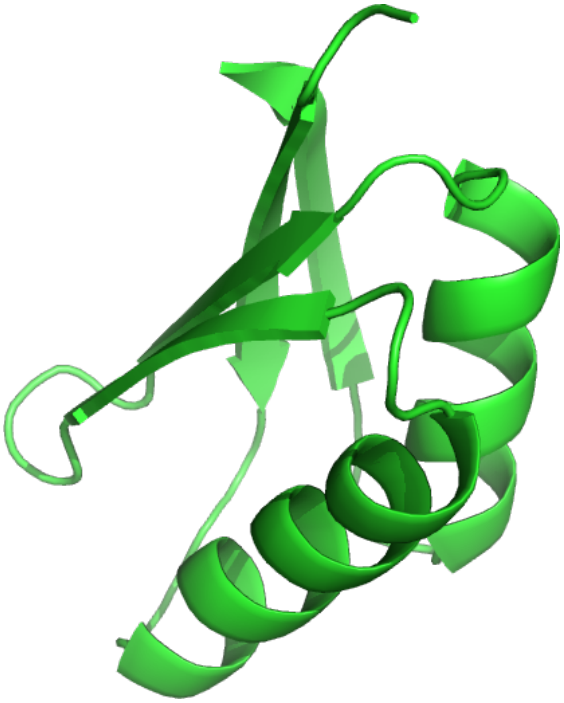


Figure 1.14 – Cartoon representation of the Atx1 metallochaperone protein of Saccharomyces cerevisiae (PDB identifier 1CC8)

1.1.2.5 Proteins can be divided into modular units.

In addition to secondary structures, a protein can be divided into substructures called *domains* which can fold more or less independently from each other. One domain is between 40 and 350 residues long, and one protein is typically made of one to several dozens of domains, which are often connected by short unstructured flexible parts. Some domains are found in many different proteins, potentially originating from the accidental joining of independent gene sequences.

A well-studied example of small protein domain is the *Src Homology 3 domain* (SH3 domain), a 60-85 residues long domain present in a large number of proteins involved in cell polarization, subcellular localization signal transduction and regulation of tyrosine kinase activity [MWS94].

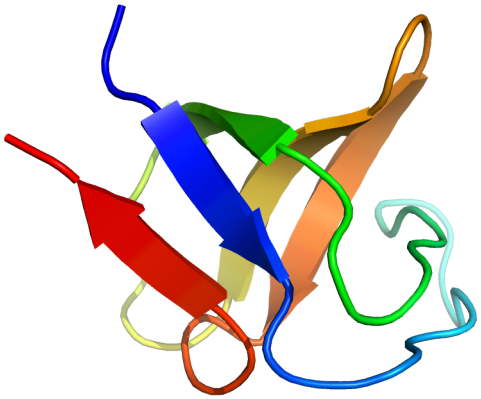


Figure 1.15 – Cartoon representation of a SH3 domain (PDB 1SHG)

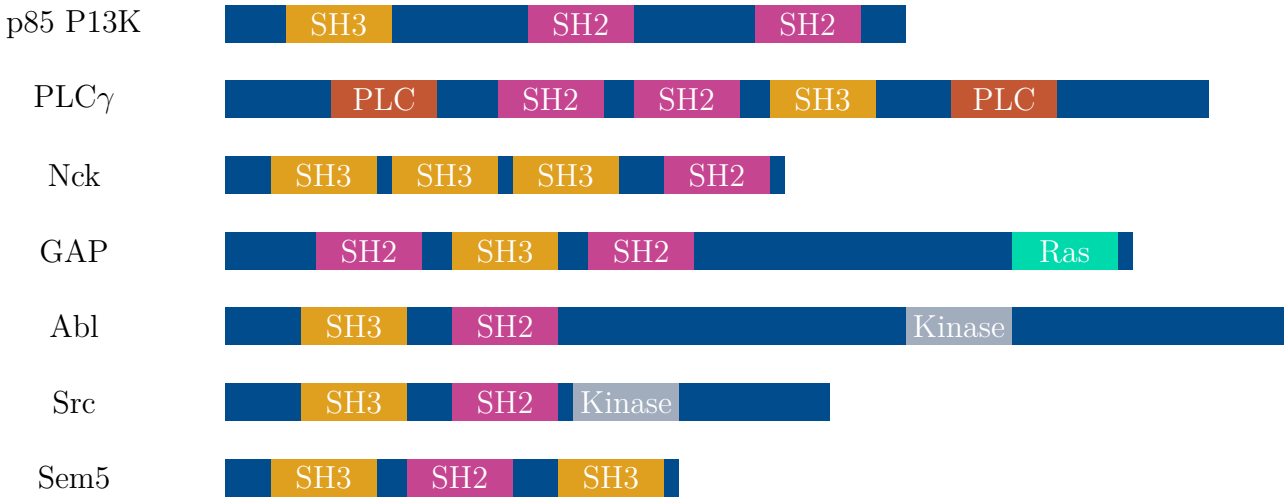


Figure 1.16 – Illustration of locations of different domains in SH3-containing sequences. Redrawn from [Mus+92]

1.1.2.6 Protein subunits arrange into quaternary structures

Some proteins are actually assemblies of several folded protein chains forming complexes held together by noncovalent interactions. Examples of such proteins include the previously described immunoglobulin (figure 1.1) and insulin (figure 1.2). When subunits are identical or similar, these proteins are termed *oligomers*

(more specifically *dimers* if they have two subunits, *trimers* if they have three subunits, etc.).

Hemoglobin is a famous example of tetramer (see figure 1.17).

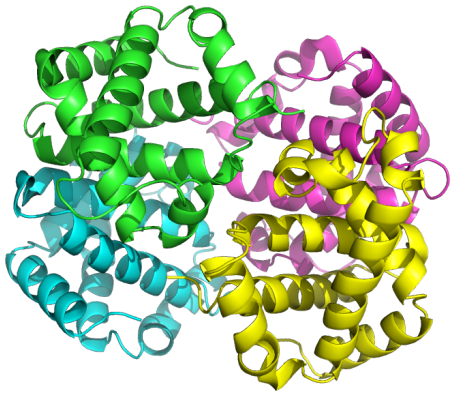


Figure 1.17 – Cartoon representation of deoxy human hemoglobin (PDB 1A3N). The four chains of this tetramer are represented with different colors.

Arrangements of these subunits are referred to as *quaternary structures*.

1.2 A functional 3D molecule determined by a 1D sequence

A central principle in bioinformatics is that a protein's amino acid chain encodes its shapes and functions. This principle is known as *Anfinsen's dogma* [Anf73]. Biological evidence of this was provided for ribonuclease: in the presence of certain solvents, a protein loses its natural shape, and spontaneously restores it when the solvent is removed, relying only on its primary chain. To make it simple, the primary chain can be seen as a necklace with beads that have different colors, sizes, properties, and these properties constrain the necklace to fold a certain unique way corresponding to a minimum free energy, strained by possible bond angles and physico-chemical interactions between residues. Depending on the side chains, these interactions can be electrostatic interactions, hydrogen bonds, van der Waals attractions, disulfide bonds, or arise from the hydrophobic clustering force which prevents water from accessing the binding site so it does not compete

with ligands. This 3D structure makes it possible for the protein to fulfill its functions.

Chapter 2

Using residue conservation to annotate a protein sequence

In this chapter, we outline the underlying challenge motivating this thesis: the number of unannotated protein sequences is exponentially increasing and calls for *in-silico* annotation methods. We explain how sequence conservation throughout evolution make it possible to annotate sequences, by searching for *homologs* – proteins with a common ancestor – and introduce the reader to *homology search* with an overview of the standard *alignment-based homology search* approaches, based on positional residue conservation throughout evolution. We start with the most straightforward approach which is pairwise sequence alignment, then we point out how sensitivity can be improved by modeling residue conservation and variability in a whole set of homologous sequences using sequence profiles and profile Hidden Markov Models, starting with sequence-to-model alignments, and we conclude with the further sensitivity brought by model-to-model alignments in remote homology detection.

2.1 The sequence annotation problem

Since the 1970s, sequencing technologies have become fully automated, faster, smaller and cheaper (see figure 2.1), making it easier to sequence lots of genomes.

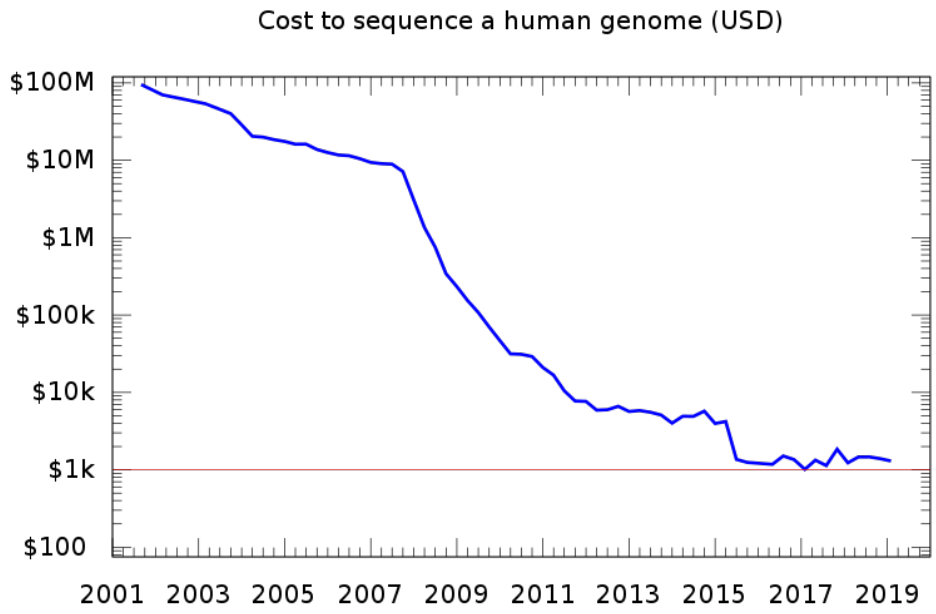


Figure 2.1 – Total cost of sequencing a human genome over time on a logarithmic scale, taken from [Moo18]

As a consequence, databases started to increasingly fill with protein sequences lacking annotation.

What "annotation" means is open to interpretation. Generally speaking, it means tagging the sequence (or parts of the sequence) with relevant information about structure (the fold of the protein) or function. However, the concept of "protein function" is vague and can basically be defined by "everything that happens to or through a protein" [Ros+03]. In an effort to standardize annotations, the Gene Ontology Consortium [Ash+00] distinguishes three levels: molecular function (e.g. transmitting a signal, catalyzing a reaction), biological process (the protein takes part in broader biological goals such as mitosis) and cellular component (the localisation of the protein, its involvment in macro-molecular complexes...). For the specific case of enzymes, functionally annotating the protein can mean identifying its class in the Enzyme classification [Web+92]. One may also provide position-specific annotations such as post-translational modifications, binding sites and enzyme active sites locations and local secondary structures, as provided by the UniProt database [Uni19]. As for structural annotation, besides

three-dimensional structural data itself (i.e. 3D coordinates of atoms in the folded protein), annotation can include the identification of secondary structure elements and infomation on symmetry and biological assembly [GB15].

Experimental methods to predict protein shapes (X-Ray Crystallography [Ken+58], Nuclear Magnetic Resonance [Wut89]) and functions (microarray analysis [Sch+95], RNA interference [Fir+98], yeast two-hybrid system [Chi+91], Deep Mutational Scanning [FF14], affine purification and mass spectrometry [Gin+07], ...) can be costly and take years of experimentation. As a consequence, while the number of unannotated sequences grows exponentially, the number of annotated sequences is growing at a much slower rate (see figure 2.2).

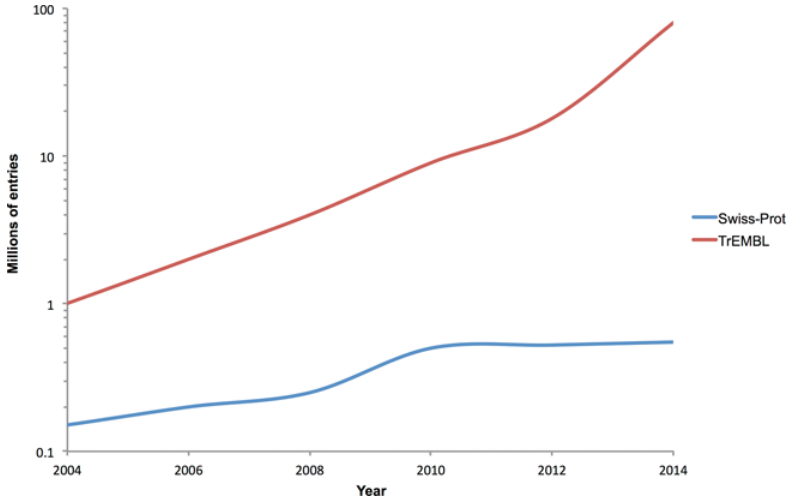


Figure 2.2 – Number of entries in the manually annotated part (in blue) SwissProt of the protein sequence database UniProt [Con19] and the uncurated part TrEMBL (in red) on a logarithmic scale, taken from [Mit+15].

It becomes obvious that *in-vitro* and *in-vivo* methods cannot keep up with the ever-increasing number of available protein sequences: this challenge raises a need for *in-silico* annotation methods.

2.2 Sequence is conserved throughout evolution

Thankfully, protein sequences do not arise spontaneously in the genome but rather constitute stable entities that were fine-tuned for thousands of years during the process of evolution and speciation.

During this process, a protein P_0 randomly undergoes *mutations* (which can be *substitutions*, *insertions* or *deletions* of residues), producing several slightly different proteins $P_1, \cdots, P_n$. P_0 is referred to as the *common ancestor* of the *homologous proteins* $P_1, \cdots, P_n$. Over generations, sequences evolve, and are naturally selected to fit their environment: evolutionary pressure will favor mutations preserving the protein structure and function or enhancing it. As described in section 1.1.2.1, some subsets of amino acids have common characteristics, and it is sometimes possible to substitute a residue at a given position with another one sharing a common property – for instance hydrophobicity or electric charge – without loss of function. Mutations in some regions of a given protein may be more constrained than others: in specific regions such as binding sites, a given residue may have to be conserved to maintain the protein functions, in some regions conservation may involve only a given property such as hydrophobicity, while other regions less involved in the function are less constrained. As a consequence, homologous protein sequences share more or less conserved regions and, provided the evolutionary period is not too long, identifying already annotated homologous sequences on the basis of sequence conservation can make it possible to annotate a protein sequence.

2.3 Pairwise sequence comparison

The most straightforward way to assess whether two sequences are homologous is to *align* them and consider the underlying similarity score.

2.3.1 Overview

A sequence alignment can be defined as an assignment of residue-residue matchings that preserves the order of the residues in the sequences.

Sequences are often visualized as aligned with an additional gap character ("." or "−"), as illustrated figure 2.3.

```
1CC8  KHYQFNVVMTCSGCSGAVNKVLTKLEPDVSKIDISLEKQLVDVYTTLPYDFILEKIKKTGKEV
4YDX  KH.EFSVDMTCGGCAEAVSRVLNKL..GGVKYDIDLPNKKVCIESEHSMDTLLATLKKTGKTV
```

Figure 2.3 – Example of pairwise sequence alignment of sequences of proteins of PDB identifiers 1CC8 and 4YDX, computed using Smith-Waterman algorithm (see section 2.3.4) with a gap open of 10 and a gap extend of 0.5

Given two sequences, several alignments are eligible, and we need to define what the best alignments are. In DNA sequences, we usually count the number of exact matches (an A in the first sequence is aligned to an A in the second sequence, etc.) while for the annotation of proteins it can be more interesting to take into account the physico-chemical similarity of the amino acids.

The key is to establish criteria for the best alignments. In practice, this translates into a scoring function yielding a score for each possible alignment.

For proteins, this score generally takes the form of a sum of terms for each aligned residue pair along with gap penalties, which can be interpreted as the logarithm of the relative likelihood that the two sequences are related compared to being unrelated, assuming that mutations have occurred independently [Dur+98].

Formally, given two sequences $x = x_1 \cdots x_n$ and $y = y_1 \cdots y_m$ on the amino acid alphabet $\mathcal{A}$, we define a *background* model B assuming that sequences are unrelated, i.e. the probability of the two sequences is simply the product of the background probability p_0 of each amino acid:

$$\mathbb{P}(x, y|B) = \prod_{i=1}^{n} p_0(x_i) \prod_{j=1}^{m} p_0(y_i)$$

and we define a *match* model M assuming that each aligned pair (x_k, y_k) occurs with a joint probability $p(x_k, y_k)$:

$$\mathbb{P}(x, y|M) = \prod_{k} p(x_k, y_k)$$

then the scoring function relies on the *log-odds ratio* of these two models along with an overall gap cost γ (discussed in section 2.3.3):

$$S(x, y) = \log \frac{\mathbb{P}(x, y|M)}{\mathbb{P}(x, y|B)} + \gamma$$

The former term can be rewritten as such [Dur+98]

$$\log \frac{\mathbb{P}(x, y|M)}{\mathbb{P}(x, y|B)} = \log \frac{\prod_k p(x_k, y_k)}{\prod_{i=1}^{n} p_0(x_i) \prod_{j=1}^{m} p_0(y_i)} = \log \prod_k \frac{p(x_k, y_k)}{p_0(x_k)p_0(y_k)} = \sum_k s(x_k, y_k)$$

where $s(a, b)$ is a *log-odds score* for residue pair (a, b):

$$s(a, b) = \log \frac{p(a, b)}{p_0(a)p_0(b)}$$

yielding the following general function:

$$S(x, y) = \sum_k s(x_k, y_k) + \gamma$$

To fully define this similarity function, we need to assign a score $s(a, b)$ for each residue pair (a, b) – this is done using *substitution matrices*, as we will see in the next section – and we need to select a gap penalty strategy, which will be discussed in section 2.3.3.

2.3.2 Substitution matrices

A *substitution matrix* or *score matrix* is a 20×20 matrix s where $s(a, b)$ is a log-odds score for amino acids a and b:

$$s(a, b) = \log \frac{p(a, b)}{p_0(a)p_0(b)}$$

where $p_0(a)$ and $p_0(b)$ are the background probabilities of amino acids a and b and $p(a, b)$ is the expected probability of observing a and b aligned in reference alignments of homologous sequences. If $s(a, b)$ is greater than 0, this substitution is termed *conservative substitution* and we expect a and b to be aligned more often than by chance.

Dividing this pairwise probability by single background frequencies is essential, since some amino acids are rarer than others. For instance, alanine A and leucine L are often found aligned in homologous alignments, but they both are very common amino acids, hence $p(A, L)$ alone is not informative. Furthermore, identity score $s(a, a)$ is different for each amino acid. Tryptophan (W), for instance, is a very rare amino acid, hence finding two W aligned is more informative than finding two A aligned for example.

Expected probabilities $p(a, b)$ are derived from sets of known trusted homologous alignments, ideally between sequences as distant in the evolution that the sequences to be aligned. In the literature, there are two main substitution matrix approaches: PAM and BLOSUM, which differ by the set of trusted alignments used and their strategy for the computation of $p(a, b)$.

2.3.2.1 PAM.

In PAM matrices [DSO78] (PAM stands for *Point Accepted Mutation*), $p(a, b)$ is the expected probability of a being replaced by b through a series of so-called *point accepted mutations* (i.e. amino acid substitutions) during a specified length of time in the evolution of the protein sequence, which is estimated using time-reversible Markov models on hypothetical phylogenetic trees. The base unit is the 1PAM matrix which corresponds to 1% accepted mutations, computed on pairwise alignments of very close homologous sequences ($\geq 85\%$ pairwise sequence identity) to make sure that there has been no more than one mutation per position. Matrices for more divergent sequences are estimated by powers of this 1PAM matrix. The most commonly used is PAM250, which corresponds to approximately 20% sequence identity.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2.0	−2.0	0.0	0.0	−2.0	0.0	0.0	1.0	−1.0	−1.0	−2.0	−1.0	−1.0	−3.0	1.0	1.0	1.0	−6.0	−3.0	0.0
R	−2.0	6.0	0.0	−1.0	−4.0	1.0	−1.0	−3.0	2.0	−2.0	−3.0	3.0	0.0	−4.0	0.0	0.0	−1.0	2.0	−4.0	−2.0
N	0.0	0.0	2.0	2.0	−4.0	1.0	1.0	0.0	2.0	−2.0	−3.0	1.0	−2.0	−3.0	0.0	1.0	0.0	−4.0	−2.0	−2.0
D	0.0	−1.0	2.0	4.0	−5.0	2.0	3.0	1.0	1.0	−2.0	−4.0	0.0	−3.0	−6.0	−1.0	0.0	0.0	−7.0	−4.0	−2.0
C	−2.0	−4.0	−4.0	−5.0	12.0	−5.0	−5.0	−3.0	−3.0	−2.0	−6.0	−5.0	−5.0	−4.0	−3.0	0.0	−2.0	−8.0	0.0	−2.0
Q	0.0	1.0	1.0	2.0	−5.0	4.0	2.0	−1.0	3.0	−2.0	−2.0	1.0	−1.0	−5.0	0.0	−1.0	−1.0	−5.0	−4.0	−2.0
E	0.0	−1.0	1.0	3.0	−5.0	2.0	4.0	0.0	1.0	−2.0	−3.0	0.0	−2.0	−5.0	−1.0	0.0	0.0	−7.0	−4.0	−2.0
G	1.0	−3.0	0.0	1.0	−3.0	−1.0	0.0	5.0	−2.0	−3.0	−4.0	−2.0	−3.0	−5.0	0.0	1.0	0.0	−7.0	−5.0	−1.0
H	−1.0	2.0	2.0	1.0	−3.0	3.0	1.0	−2.0	6.0	−2.0	−2.0	0.0	−2.0	−2.0	0.0	−1.0	−1.0	−3.0	0.0	−2.0
I	−1.0	−2.0	−2.0	−2.0	−2.0	−2.0	−2.0	−3.0	−2.0	5.0	2.0	−2.0	2.0	1.0	−2.0	−1.0	0.0	−5.0	−1.0	4.0
L	−2.0	−3.0	−3.0	−4.0	−6.0	−2.0	−3.0	−4.0	−2.0	2.0	6.0	−3.0	4.0	2.0	−3.0	−3.0	−2.0	−2.0	−1.0	2.0
K	−1.0	3.0	1.0	0.0	−5.0	1.0	0.0	−2.0	0.0	−2.0	−3.0	5.0	0.0	−5.0	−1.0	0.0	0.0	−3.0	−4.0	−2.0
M	−1.0	0.0	−2.0	−3.0	−5.0	−1.0	−2.0	−3.0	−2.0	2.0	4.0	0.0	6.0	0.0	−2.0	−2.0	−1.0	−4.0	−2.0	2.0
F	−3.0	−4.0	−3.0	−6.0	−4.0	−5.0	−5.0	−5.0	−2.0	1.0	2.0	−5.0	0.0	9.0	−5.0	−3.0	−3.0	0.0	7.0	−1.0
P	1.0	0.0	0.0	−1.0	−3.0	0.0	−1.0	0.0	0.0	−2.0	−3.0	−1.0	−2.0	−5.0	6.0	1.0	0.0	−6.0	−5.0	−1.0
S	1.0	0.0	1.0	0.0	0.0	−1.0	0.0	1.0	−1.0	−1.0	−3.0	0.0	−2.0	−3.0	1.0	2.0	1.0	−2.0	−3.0	−1.0
T	1.0	−1.0	0.0	0.0	−2.0	−1.0	0.0	0.0	−1.0	0.0	−2.0	0.0	−1.0	−3.0	0.0	1.0	3.0	−5.0	−3.0	0.0
W	−6.0	2.0	−4.0	−7.0	−8.0	−5.0	−7.0	−7.0	−3.0	−5.0	−2.0	−3.0	−4.0	0.0	−6.0	−2.0	−5.0	17.0	0.0	−6.0
Y	−3.0	−4.0	−2.0	−4.0	0.0	−4.0	−4.0	−5.0	0.0	−1.0	−1.0	−4.0	−2.0	7.0	−5.0	−3.0	−3.0	0.0	10.0	−2.0
V	0.0	−2.0	−2.0	−2.0	−2.0	−2.0	−2.0	−1.0	−2.0	4.0	2.0	−2.0	2.0	−1.0	−1.0	−1.0	0.0	−6.0	−2.0	4.0

Table 2.1 – PAM250 matrix

While effective for close homologs, PAM matrices are less reliable for more divergent sequences, outperformed by BLOSUM matrices [TN20].

2.3.2.2 BLOSUM.

BLOSUM matrices [HH92] (BLOSUM stands for *BLOcks SUbstitution Matrix*) are more straightforward. Rather than extrapolated from comparisons of closely related proteins, substitution probabilities are directly counted in sequence alignments from the *BLOCKS* database [PHH96], a database of local ungapped multiple sequence alignments of conserved regions. Just as PAM, there are different BLOSUM matrices based on the sequence identity threshold wanted, e.g. BLOSUM62, the most used BLOSUM matrix, is built from conserved blocks with less than 62% sequence identity.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4.0	−1.0	−2.0	−2.0	0.0	−1.0	−1.0	0.0	−2.0	−1.0	−1.0	−1.0	−1.0	−2.0	−1.0	1.0	0.0	−3.0	−2.0	0.0
R	−1.0	5.0	0.0	−2.0	−3.0	1.0	0.0	−2.0	0.0	−3.0	−2.0	2.0	−1.0	−3.0	−2.0	−1.0	−1.0	−3.0	−2.0	−3.0
N	−2.0	0.0	6.0	1.0	−3.0	0.0	0.0	0.0	1.0	−3.0	−3.0	0.0	−2.0	−3.0	−2.0	1.0	0.0	−4.0	−2.0	−3.0
D	−2.0	−2.0	1.0	6.0	−3.0	0.0	2.0	−1.0	−1.0	−3.0	−4.0	−1.0	−3.0	−3.0	−1.0	0.0	−1.0	−4.0	−3.0	−3.0
C	0.0	−3.0	−3.0	−3.0	9.0	−3.0	−4.0	−3.0	−3.0	−1.0	−1.0	−3.0	−1.0	−2.0	−3.0	−1.0	−1.0	−2.0	−2.0	−1.0
Q	−1.0	1.0	0.0	0.0	−3.0	5.0	2.0	−2.0	0.0	−3.0	−2.0	1.0	0.0	−3.0	−1.0	0.0	−1.0	−2.0	−1.0	−2.0
E	−1.0	0.0	0.0	2.0	−4.0	2.0	5.0	−2.0	0.0	−3.0	−3.0	1.0	−2.0	−3.0	−1.0	0.0	−1.0	−3.0	−2.0	−2.0
G	0.0	−2.0	0.0	−1.0	−3.0	−2.0	−2.0	6.0	−2.0	−4.0	−4.0	−2.0	−3.0	−3.0	−2.0	0.0	−2.0	−2.0	−3.0	−3.0
H	−2.0	0.0	1.0	−1.0	−3.0	0.0	0.0	−2.0	8.0	−3.0	−3.0	−1.0	−2.0	−1.0	−2.0	−1.0	−2.0	−2.0	2.0	−3.0
I	−1.0	−3.0	−3.0	−3.0	−1.0	−3.0	−3.0	−4.0	−3.0	4.0	2.0	−3.0	1.0	0.0	−3.0	−2.0	−1.0	−3.0	−1.0	3.0
L	−1.0	−2.0	−3.0	−4.0	−1.0	−2.0	−3.0	−4.0	−3.0	2.0	4.0	−2.0	2.0	0.0	−3.0	−2.0	−1.0	−2.0	−1.0	1.0
K	−1.0	2.0	0.0	−1.0	−3.0	1.0	1.0	−2.0	−1.0	−3.0	−2.0	5.0	−1.0	−3.0	−1.0	0.0	−1.0	−3.0	−2.0	−2.0
M	−1.0	−1.0	−2.0	−3.0	−1.0	0.0	−2.0	−3.0	−2.0	1.0	2.0	−1.0	5.0	0.0	−2.0	−1.0	−1.0	−1.0	−1.0	1.0
F	−2.0	−3.0	−3.0	−3.0	−2.0	−3.0	−3.0	−3.0	−1.0	0.0	0.0	−3.0	0.0	6.0	−4.0	−2.0	−2.0	1.0	3.0	−1.0
P	−1.0	−2.0	−2.0	−1.0	−3.0	−1.0	−1.0	−2.0	−2.0	−3.0	−3.0	−1.0	−2.0	−4.0	7.0	−1.0	−1.0	−4.0	−3.0	−2.0
S	1.0	−1.0	1.0	0.0	−1.0	0.0	0.0	0.0	−1.0	−2.0	−2.0	0.0	−1.0	−2.0	−1.0	4.0	1.0	−3.0	−2.0	−2.0
T	0.0	−1.0	0.0	−1.0	−1.0	−1.0	−1.0	−2.0	−2.0	−1.0	−1.0	−1.0	−1.0	−2.0	−1.0	1.0	5.0	−2.0	−2.0	0.0
W	−3.0	−3.0	−4.0	−4.0	−2.0	−2.0	−3.0	−2.0	−2.0	−3.0	−2.0	−3.0	−1.0	1.0	−4.0	−3.0	−2.0	11.0	2.0	−3.0
Y	−2.0	−2.0	−2.0	−3.0	−2.0	−1.0	−2.0	−3.0	2.0	−1.0	−1.0	−2.0	−1.0	3.0	−3.0	−2.0	−2.0	2.0	7.0	−1.0
V	0.0	−3.0	−3.0	−3.0	−1.0	−2.0	−2.0	−3.0	−3.0	3.0	1.0	−2.0	1.0	−1.0	−2.0	−2.0	0.0	−3.0	−1.0	4.0

Table 2.2 – BLOSUM62 matrix

2.3.3 Gap costs

So far, we have seen how we can score each aligned residue pair using substitution scores. We still need to handle the two remaining mutation types: insertions and deletions, which, in sequence alignments, materialize as *gaps*. Recalling that the closer the sequences, the fewer mutations should separate them, it seems justified to penalize gaps with a negative score.

The standard gap cost for a gap of length g is either linear:

$$\gamma(g) = -\alpha g$$

or affine:

$$\gamma(g) = -\beta - \alpha g$$

where β is referred to as the *gap open penalty* and α as the *gap extend penalty*, usually smaller than β. This is justified by the observation that multiple residues can be deleted or inserted in a single mutational event, making stretches of gaps common.

Unfortunately, unlike substitutions, there is no standard statistically grounded

approach to score a priori insertions and deletions. In practice, gap costs of pairwise sequence alignments are chosen empirically.

2.3.4 Alignment algorithms

Given a score function, there are different algorithms to find an optimal alignment, depending on the context. The two seminal algorithms are Needleman-Wunsch[Spr70] and Smith-Waterman[SW+81a] algorithms, both yielding an exact solution based on dynamic programming. The former performs *global* alignments that is to say that it attempts to align every residue in each sequence, while the latter performs *local* alignments by not penalizing gaps at both ends of the sequences, making it more suited for the alignment of a small region to a larger sequence.

While dynamic programming can be effective, performing several pairwise alignments in a row remains costly. Heuristics were designed to speed up the process, such as FASTA[PL88] and BLAST[Alt+90], one of the most widely used homology search methods, which relies on local short and exact matches (*seed and extend*). Such fast tools allow the user to search a target sequence against a whole database efficiently and collect potential homologs, usually based on the Expect value (*E-value*) of the sequences aligned with the target, which reflects the significance of a match with the number of hits one might expect to get by chance with a greater score in a database of the same size.

2.4 Embody residue conservation and variability in homologous sequences

We saw that a sequence could be annotated by aligning it to an annotated sequence. While effective for fairly close homologs, this approach is not suited for more remote homologs. Indeed, as stated in section 2.2, some homologous sequences can share 20% identity or less, making pairwise alignments less reliable. A solution is to model whole sets of homologous sequences rather than considering one single annotated sequence. This way, one can make use of identified conserved

regions and overall variability within the considered sequences to decide whether a target sequence is related to them or not and annotate it accordingly. We'll start by describing *multiple sequence alignments* before reviewing positional models representing them, from straightforward ungapped matrices to gap-handling *profiles* and *profile Hidden Markov Models*.

2.4.1 Multiple sequence alignments

Multiple sequence alignments are the extension of pairwise sequence alignments to more than two sequences. Formally, given $S = \{s^n\}_{n=1,\cdots,N}$ a set of N protein sequences of lengths $l_1, \cdots, l_N$, a multiple sequence alignment (MSA) of these sequences can be defined as a set of N sequences $X = \{x^n\}_{n=1,\cdots,N}$ on the alphabet of S extended with a new gap character '$-$', which all have the same length L and such that removing all gaps from a sequence x^n gives s^n. By extension, L is called the length of the MSA.

Aligning several homologous sequences provides an overview of amino acid conservation patterns within the considered set, and adequate color schemes reflecting residue types (such as polar, positively/negatively charged, hydrophobic, small nonpolar) bring to light conserved regions representative of a family and exhibit properties retained by evolutionary pressure at each position (see figure 2.4).

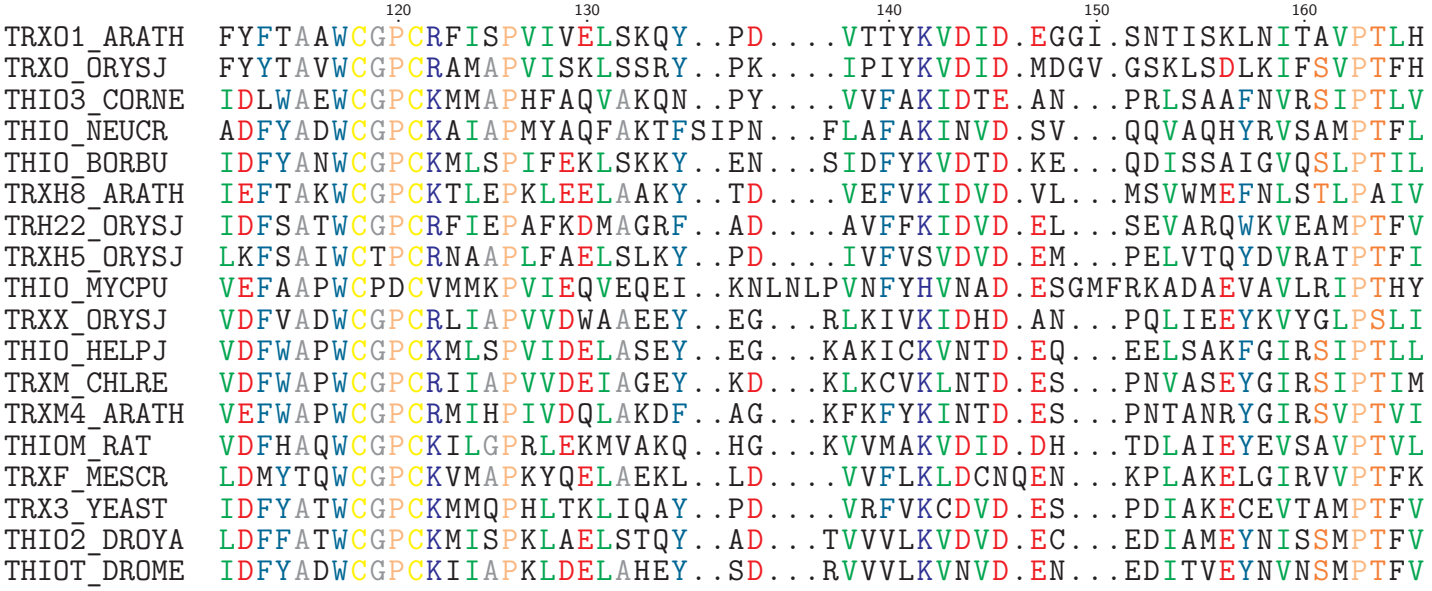


Figure 2.4 – Sample of an MSA of sequences from the thioredoxin family made by MUSCLE [Edg04]. Residues are colored here according to the consensus chemical property identified in the column, highlighting conserved regions characteristic of the thioredoxin family, notably the double cysteine bond of the redox active site.

Multiple sequence alignments can be made by hand using expert knowledge on conserved residues, buried residues, secondary and tertiary structures, insertions and deletions etc., but obviously this task is time-consuming and complex, and computational approaches are preferred. Since finding the optimal MSA is an NP-complete problem [SW+81b], multiple sequence alignment algorithms used in practice are based on heuristics. The most commonly used approach is progressive alignment construction, which builds the MSA by successively combining pairwise alignments following a binary guide tree. This approach is implemented for instance in algorithms of the Clustal family [HS88], MAFFT [Kat+02], T-Coffee [NHH00] and Kalign2 [LFS09]. Other main approaches include iterative methods such as MUSCLE [Edg04], which work in a similar way but iteratively realign the sequences and append new ones, and consensus methods such as M-Coffee[Wal+06] and MergeAlign[CK12], which find consensus among MSAs outputted by different methods.

Sequence logos can provide a visual representation of the most conserved parts of a given MSA. Each position i in the MSA is assigned a stack of letters whose height is the column information in bits, measured by Shannon entropy: $H_i = -\sum_a p_i(a) \log p_i(a)$, and each letter is scaled according to its frequency in the column. An example of sequence logo for the SH3 domain is given figure 2.5.

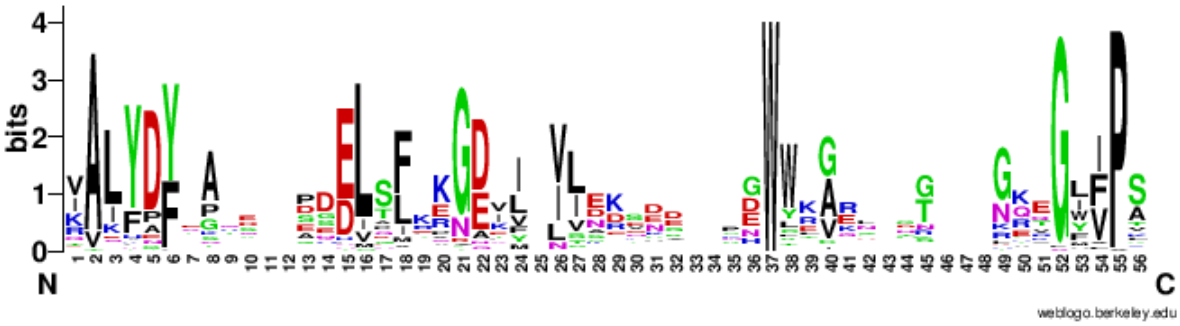


Figure 2.5 – Example of sequence logo built using WebLogo [Cro+04] from the "seed" alignment of the Pfam entry PF00018 [SED97] for the SH3 domain.

2.4.2 Positional models for multiple sequence alignments

Measuring the similarity between a target sequence and a given set of homologous proteins is typically done by aligning the sequence to a model built from their multiple sequence alignment. We review in this section the commonly used models that are based on positional residue conservation.

2.4.2.1 Ungapped matrices model conserved regions of multiple sequence alignments

Position-Specific Scoring Matrices (PSSMs), also known as *Position-Specific Weight Matrices* (PSWMs) or *Position Weight Matrices* (PWMs), are matrices modelling conserved regions of multiple sequence alignments, introduced by [Sch+86]. A PSSM matrix turns a MSA into a position-specific scoring system. Each column i in a PSSM M corresponds to a column in the corresponding MSA and has 20 rows, one for each amino acid a, and $M_i(a)$ is a log-odds score derived from the probability $p_i(a)$ of finding amino acid a at position i in the MSA and

its background probability $p_0(a)$:

$$M_i(a) = \log \frac{p_i(a)}{p_0(a)}$$

One can decide whether a sequence is similar to the sequences represented by the PSSM by applying the matrix as a sliding window along the sequence and considering the position with the highest sum of the log-odds score. This implies the assumption that positions in the MSA are independent.

2.4.2.2 Adding pseudocounts to compensate for a lack of data

A prevalent problem in learning a model's parameters is that, most of the time, there are not enough sequences in the training set compared to the number of free parameters that need to be estimated, and these sequences are not uniformly sampled. As a consequence of sampling variation, amino acids at some positions might be misrepresented in the sequence set with respect to the actual population.

To compensate for this lack of data, smooth sampling variations, and avoid null probabilities, artificial additional counts termed *pseudocounts* are added to observed amino acid counts.

A first simple technique is to add a constant to the observed counts:

$$\hat{p}_i(a) = \frac{o_i(a) + \alpha}{\sum_b (o_i(b) + \alpha)}$$

where $o_i(a)$ is the observed count of letter a at position i and α is an arbitrary constant which tunes the importance of pseudo-counts with respect to observed counts: α prevails when observed counts in a are insufficient. This type of pseudocounts is sometimes called *additive smoothing*, or *Laplace smoothing* when $\alpha = 1$.

More elaborate pseudocount strategies take into account prior knowledge on amino acids physico-chemical properties. A first strategy is to incorporate amino acid background probabilities p_0:

$$\hat{p}_i(a) = \frac{o_i(a) + \alpha p_0(a)}{\sum_b (o_i(b) + \alpha p_0(b))}$$

A more evolved approach introduced by [HH96a] relies on substitution matrices such as BLOSUM or PAM (see section 2.3.2), using the probability $\tilde{p}_i(a)$ of having a at position i by mutation of residues, i.e. $\tilde{p}_i(a) = \sum_b p_i(b)p(a|b)$ where $p(a|b)$ is the probability for b to mutate into a.

$$\hat{p}_i(a) = \frac{o_i(a) + \alpha\tilde{p}_i(a)}{\sum_b (o_i(b) + \alpha\tilde{p}_i(b))}$$

The choice of α enables to tune the number of observations simulated according to prior knowledge and thus the number of actual observations needed to dominate this prior.

A more advanced pseudocount strategy has been proposed to incorporate knowledge on the whole column composition using Dirichlet mixtures [Bro+93]. The idea is that, for example, if a MSA column is biased towards small hydrophobic amino acids, probabilities of other small hydrophobic amino acids should be increased. Dirichlet mixtures consist of vectors representing Dirichlet distributions corresponding to typical distributions of amino acids, associated with *mixture parameters* weighing the distributions. Dirichlet mixtures are inferred from large datasets of trusted multiple sequence alignments, usually using maximum likelihood heuristics.

2.4.2.3 Reweighting sequences to compensate for selection and phylogenetic bias

As explained in [Dur+98], a common issue when working with multiple sequence alignments is that the assumption that sequences represent independent samples of a given protein family is inappropriate. In a typical MSA, a selection bias (some species are more sequenced than others, typically human pathogens) and a phylogenetic bias (there is a dependency structure in sequences due to their evolutionary relationship) lead to sequences closely related to each other. To compensate for this effect, a different weight is assigned to each sequence, a simple strategy being to assign the inverse of the number of similar sequences given an identity threshold. Probabilities of amino acids are then computed factoring in each sequence weight.

2.4.2.4 Profiles implement gap treatment.

To search for longer regions, one needs to take insertions and deletions into account. For this purpose, PSSMs have been extended to *profiles* [GME87] by allowing additional position-specific gap costs and making it possible to align a sequence with affine gap penalty using a Smith-Waterman-like algorithm.

One of the most widely used homology search approaches based on profiles is PSI-BLAST [Alt+97]. It iteratively builds a profile from a target sequence by aligning it to sequences in a non-redundant sequence database using a BLAST-like algorithm, expanding it each time with newly found matching sequences.

Profiles for known families of protein sequences are catalogued in databases such as PROSITE [Sig+12] whose tool ProfileScan allows users to scan a given sequence for the occurrence of matching profiles in the database.

2.4.2.5 Profile Hidden Markov Models introduce transition probabilities.

Though gap costs are allowed in profiles, they are not derived from a statistical justification but rather empirically calculated in an *ad hoc* fashion. *Profile Hidden Markov Models* (pHMMs) went further by adding insertion and deletion states and transition probabilities to enter those states. Unlike simple profiles where gap penalties are determined empirically, gap parameters in a pHMM are parameters of a probabilistic model, optimized on a training set together with amino acid emission probabilities.

pHMMs are a subclass of hidden Markov Models (HMMs) with a specific linear left-to-right architecture particularly suited for the representation of sequences, introduced by Krogh *et al.* in 1994 for proteins [Kro+94] and widely popularized with the release of the software package HMMER by Eddy *et al.* [Edd96; Edd98; FCE11]. A pHMM can be described as a finite probabilistic generative model representing a given set of sequences and defining a probability distribution over an infinite number of possible sequences. It is typically represented by a state diagram (see figure 2.6) consisting of states and transitions between these states. A sequence is generated starting from the initial "start" state and transitioning from one state to another according to transition probabilities until the "end"

state, where some states emit a symbol according to the corresponding emission probabilities.

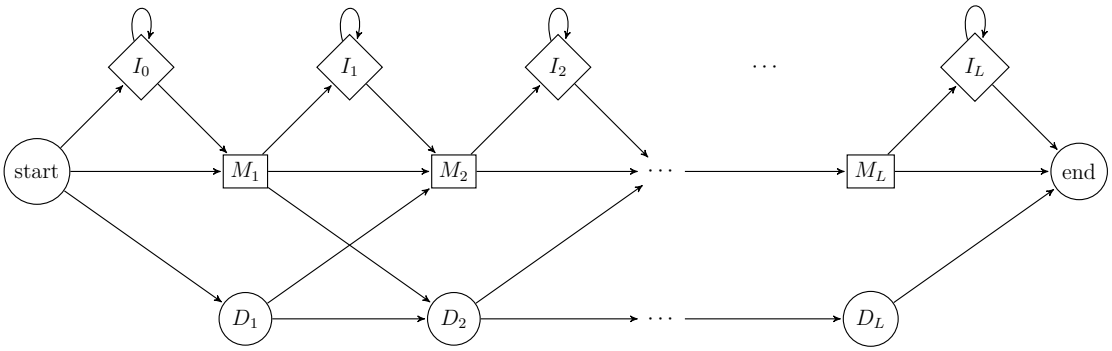


Figure 2.6 – Architecture of an L-state profile Hidden Markov Model.

Besides the *start* and *end* states, there are three kinds of states in a pHMM:

- *Match* states, conventionally labeled M and represented by squares, correspond to conserved columns in the underlying multiple sequence alignment (typically columns with less than 50% gaps in HMMER). A match state emits a letter with a probability derived from the frequency of the letter at the position in the MSA and its background probability. It can be seen as a column in a profile matrix.

- *Insertion* states, conventionally labeled I and represented by diamonds, handle insertions. They emit letters with a probability usually set to the background probability.

- *Deletion* states, conventionally labeled D and represented by circles, handle deletions. They are silent states: they do not emit anything but allow "jumps" between match states.

Gap treatment is analogous to pairwise affine gap costs [Dur+98]: an insertion of length x will have a score for transitioning from the match state to the insert state and leaving the insert state for another match state (gap open cost) and $(x-1)$ times the score of the loop transition from the insertion state to itself (gap extend cost).

One can align a target sequence to a query pHMM and get a probability for the pHMM to generate the sequence using dynamic programming algorithms. As for pairwise sequence alignments and profile alignments, probabilities are turned into log-odds scores using a background model, the key difference is the existence of transition probabilities defining different paths: the probability of a sequence and a path is then simply the product of the transition probabilities and the emission probabilities on the path. The Viterbi algorithm [Vit67] is a dynamic programming algorithm which finds the optimal alignment of a sequence to a pHMM, that is to say the most probable path generating the sequence. The Forward algorithm, on the other hand, gives the probability that a pHMM generates a given sequence. It is similar to the Viterbi algorithm but yields the full probability summed over all possible paths.

A pHMM can be built from a set of unaligned sequences using the Baum-Welch algorithm, a special case of the Expectation-Maximization algorithm. Most of the time though, a pHMM is built from an existing multiple sequence alignment. Training a pHMM can be broken down into two subproblems: the choice of architecture and the assignment of the probability parameters (emission and transition probabilities). Usually, the architecture is simply designed by assigning columns of the MSA with less than 50% gap characters to match states and adding the corresponding insertion and deletion states in between. Then, probabilities are assigned by counting the number of times each transition or emission is used in the alignment.

Profile Hidden Markov Models are extensively used for the purposes of sequence annotation and classification. The idea is to build a pHMM for each known family of homologous sequences and to annotate unknown sequences by identifying the pHMM which yields the best alignment score above a given threshold. Moreover, besides their use in automated sequence annotation, profile Hidden Markov Models also provide a visual understanding of protein families (see figure 2.7).

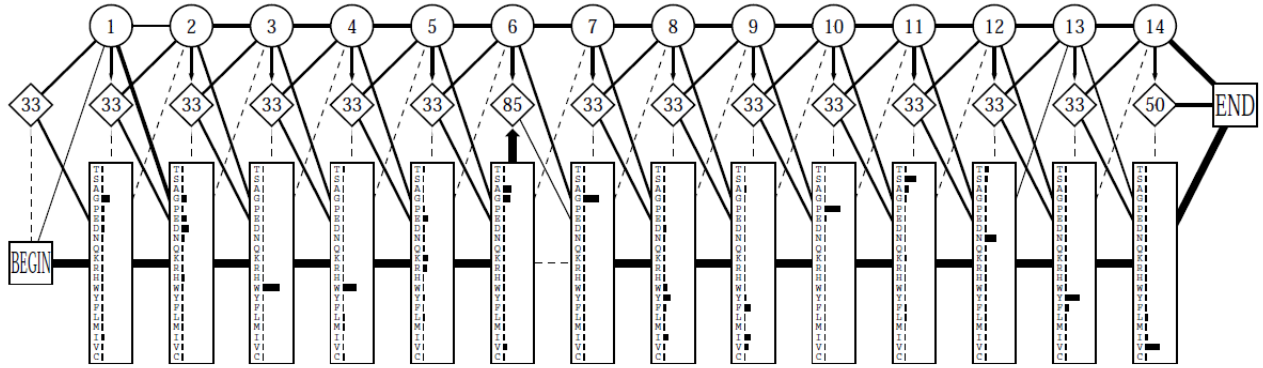


Figure 2.7 – Example of pHMM for the SH3 domain, taken from [Kro98]. The model provides a visual representation of the domain: by looking at the model, one can identify positions that are more conserved within the family.

HMMER [FCE11] is probably the most popular homology search package based on pHMMs. It was shown to outperform BLAST while having an overall comparable speed [Edd09], and is used in several databases including Pfam [SED97; ElG+19] which gathers protein domain families, TIGRFAMs [HSW03] which focuses more on full-length proteins and CATH-Gene3D [Cuf+09] which extends the CATH protein structure database with domains without experimentally determined structures. SAM [HK96] is another popular software package, used in the SUPERFAMILY [Gou+01; Pan+19] database which gathers pHMMs representing protein domains at the superfamily level in the structural domains database SCOP [Mur+95]. Also mentionable is the PANTHER [Tho+03] database which focuses more on the functional classification of proteins.

These packages are based on the alignment of a sequence to a pHMM, well suited for refined functional characterization of homologous proteins. More recent tools such as HHsuite [Ste+19] enable a more sensitive remote homology search by performing pHMM-pHMM alignment. These approaches will be detailed in the next section.

2.5 Improve sensitivity by aligning models to models

In the last section, we saw that homology search sensitivity could be improved by embodying residue conservation and variability within a whole set of homologous sequences. We will see in this section that this could be further enhanced by modeling the variability on the target side. We will detail this approach for profiles and pHMMs.

2.5.1 Profile-profile alignment

As explained in section 2.4.2.1, aligning a sequence to a profile improves homology search sensitivity compared to aligning it to a single sequence. Indeed, modeling a whole set of homologous protein sequences includes the notion of conserved positions representative of the considered family and reflects allowed residue variations. Sensitivity was further improved with the application of this idea to both objects being aligned using profile-profile alignment methods, introduced by Pietrokovski et al. with LAMA [Pie96], which performs alignments of conserved ungapped regions termed *blocks* classified in the BLOCKS database [PHH96]. Two other popular methods are PROF_SIM [YL02] and COMPASS [SG03] which allow gaps in the alignments.

Central to the problem of aligning two profiles is the question of how to score the similarity of two columns. As opposed to sequence-profile alignment where a score for each letter in the sequence can be derived from a profile column, in profile-profile alignment two vectors have to be compared. Different strategies have been proposed, comparing either two probability vectors p_i and q_k or two log-odds score vectors A_i and B_k, including:

- *Pearson correlation*, implemented in [Pie96]:

$$s(A_i, B_k) = \frac{\sum_a (A_i(a) - \bar{A}_i)(B_k(a) - \hat{B}_k)}{\sqrt{\sum_a (A_i(a) - \bar{A}_i)^2}\sqrt{\sum_a (B_k(a) - \hat{B}_k)^2}}$$

- *Jensen-Shannon score*, implemented in [YL02]:

$$s(p_i, q_k) = \frac{1}{2}\left(1 - D^{JS}(p_i, q_k)\right)\left(1 + D^{JS}\left(\frac{p_i + q_k}{2}, p_0\right)\right)$$

where p_0 is the amino acids background probability distribution and D^{JS} is the Jensen-Shannon divergence, defined as:

$$D^{JS}(p, q) = \frac{1}{2}D^{KL}(p, \frac{p+q}{2}) + \frac{1}{2}D^{KL}(q, \frac{p+q}{2})$$

where D^{KL} is the Kullback-Leibler divergence:

$$D^{KL}(p, q) = \sum_a p(a) \log \frac{p(a)}{q(a)}$$

Unlike Pearson correlation, this scoring function assigns the highest scores to column pairs that have both similar distributions and a higher significance, which is defined here by how distant the average distribution $\frac{p_i+q_k}{2}$ is from the background distribution.

- *Log-average*, implemented in [OZ01]:

$$s(p_i, q_k) = \log \sum_a \sum_b p_i(a) q_k(b) \frac{p_0(a, b)}{p_0(a) p_0(b)}$$

where $p_0(a)$ is the background probability for a and $p_0(a, b)$ is the background probability for a and b to be aligned in homologous sequence alignments, usually derived from the BLOSUM62 matrix.

This scoring function assigns higher scores to conserved positions with a similar probability distribution, while random distributions are assigned intermediate scores.

- *PICASSO score*, implemented in [HH03]:

$$s(p_i, q_k) = \sum_a p_i(a) \log \frac{q_k(a)}{p_0(a)}$$

and made symmetric in [MSG03]:

$$s(p_i, q_k) = \sum_a p_i(a) \log \frac{q_k(a)}{p_0(a)} + \sum_b q_k(b) \log \frac{p_i(b)}{p_0(b)}$$

This score behaves quite similarly to Log-average [OWE04].

- *Dot-product scoring* (scalar product), implemented in FFAS [Ryc+00]:

  on the profile columns:

  $$s(A_i, B_k) = \langle A_i, B_k \rangle = \sum_a A_i(a) B_k(a)$$

  or on the probabilities:

  $$s(p_i, q_k) = \langle p_i, q_k \rangle = \sum_a p_i(a) q_k(a)$$

  This scoring function assigns high scores to highly conserved positions. A theoretically perfect match (two identical columns) will not necessarily have the highest score. A normalized alternative was assessed in [OWE04]. Interestingly, it did not perform well in fold recognition and alignment benchmarks compared with the unnormalized version. Authors explain this difference in performance with the fact that normalizing leads to higher scores for columns that are similar to the amino acid background distribution.

The wideness of this list of scoring functions illustrates how largely ill-defined the problem of comparing two columns is and how deciding on a scoring function is not trivial. Overall, performances of all of these scoring functions are rather similar when their parameters are properly optimized, but probabilistic scoring functions Log-average and PICASSO are more robust to changes in gap penalties, as opposed to dot product in particular whose gap parameters have to be optimized for each task [OWE04].

Profile-profile alignment based homology search is essentially performed in a similar way to sequence-profile alignment based homology search, with the difference that the template representing the sequence to be annotated is a profile.

Usually, this profile is built using PSI-BLAST, by searching the initial sequence against a selected database. Enriching the target sequence with similar homologous sequences allows for a modeling of residue conservation and allowed variability on both sides, enabling a more remote homology search – profile-profile methods have been shown to outperform PSI-BLAST alone in fold recognition [OWE04] –, the identification of new relations between protein families and protein structure predictions [SBG03].

2.5.2 pHMM-pHMM alignment

Since profile Hidden Markov Models can be seen as extensions of sequence profiles with an appropriate gap treatment, and since profile-profile alignments improve sensitivity over sequence-profile alignments, one can easily see the potential of performing pHMM-pHMM alignment. In 2004, Söding released HHsearch [Söd05], an homology search tool based on a pHMM-pHMM alignment algorithm. This algorithm is a natural extension of the sequence-pHMM alignment algorithm based on the Viterbi algorithm and generalizing the original log-odds score to a log *sum-of-odds* score. An alignment of two pHMMs is viewed as a path through both pHMMs and probabilities are summed over all sequences that can be co-emitted along this alignment path compared with a background model. Following [Söd05], the simple log-odds score of a given sequence x on a path:

$$S_{LO} = \log \frac{\mathbb{P}(x_1, \cdots, x_L | \textit{emission on path})}{\mathbb{P}(x_1, \cdots, x_L | \textit{Null})}$$

is generalized into:

$$S_{LSO} = \log \sum_{x_1, \cdots, x_L} \frac{\mathbb{P}(x_1, \cdots, x_L | \textit{co-emission on path})}{\mathbb{P}(x_1, \cdots, x_L | \textit{Null})}$$

and the sum runs over all sequences $x = x_1, \cdots, x_L$ that can be emitted along the alignment path. It can be rewritten as:

$$S_{LSO} = \sum_{k: X_k Y_k = MM} S_{aa}(q_{i(k)}, p_{j(k)}) + \log \mathcal{P}_{tr}$$

where k runs over all paired match states, $\mathcal{P}_{tr}$ aggregates all transition

probabilities, $q_{i(k)}$ and $p_{j(k)}$ are the emission probability vectors of the two pHMMs for match state pair k and S_{aa} is the column score:

$$S_{aa}(q_i, p_j) = \log \sum_{a=1}^{20} \frac{q_i(a) p_j(a)}{p_0(a)}$$

This column score can be seen as an extension of the simple log-odds score. It assigns highest scores to columns that are similar and conserved: as we can see, when a column follows the background probability distribution, the score vanishes.

On top of the log-odds score, two additional scores taking into account prior knowledge on proteins are added to the final scoring function to help distinguish true homologs from chance hits.

The first one is an *autocorrelation* score, defined as:

$$S_{corr} = \sum_{d=1}^{4} \sum_{l=1}^{L-d} S_l S_{l+d}$$

where S_l is the column score associated with the l^{th} pair state: $S_l = S_{aa}(q_{i(l)}, p_{j(l)})$. This score reflects the expectation that column pairs with the highest scores should occur in clusters along the alignment, following the expected distribution of conserved columns along sequences [PG01].

The second one reflects prior knowledge on secondary structures:

$$S_{SS}(q_i, p_j) = M_{SS}(\rho_i^q, c_i^q, \rho_j^p, c_j^p)$$

where ρ_i^q is the secondary structure state predicted by PSIPRED [MBJ00] with confidence c_i^q and M_{SS} is a substitution matrix derived from states predicted by PSIPRED and DSSP [KS83] on SCOP domains.

In a subsequent version, authors also integrated their own method to compute pseudo-counts in a context-specific way, which was shown to improve sensitivity at the fold level [ABS12a].

HHsearch was shown to retrieve more than 2.7 times more homologs than PSI-BLAST, HMMER, COMPASS and PROF_SIM on SCOP20 domains (structural domains from SCOP database with maximum 20% pairwise sequence identity), showing its relevance in the detection of remote homologs. However, a downside

to this approach is that, unlike approaches based on sequence-to-model alignments such as HMMER where target pHMMs can be built using expert knowledge, here query pHMMs are iteratively built with the alignment of sequences that are not reliable.

Despite remarkable homology search performances, HHsearch is too slow to perform iterative search through large databases and the target pHMM has to be built using another method, originally using a multiple sequence alignment outputted by PSI-BLAST – until the advent of HHblits, published in 2012 [Rem+12; Ste+19]. HHblits ("HMM-HMM-based lightning-fast iterative sequence search") is a derivative of HHsearch able to iteratively search a database to build a pHMM (or, equivalently, a multiple sequence alignment) on the target side. Starting from a single sequence, HHblits searches against a pre-built pHMM database (usually built on the UniClust30 database [Mir+17] which clusters the UniProtKB database [Con19] into groups with a maximum pairwise sequence identity of 30% and a minimum sequence length overlap of 80%), collects all the sequences modeled by each matching pHMM (for HHblits this means pHMMs with an E-value lower than a given threshold), builds a new pHMM incorporating these sequences, and repeats the process. It is based on the same scoring scheme as HHsearch, but is much faster thanks to a vectorization of the Viterbi algorithm, an early termination strategy, and more importantly a smart prefiltering strategy (accounting for most of the runtime) which reduces the number of pHMM pairs to be aligned "from many millions to a few thousands" [Rem+12] by discretizing the probability vectors into sequences on a 219 letters alphabet.

HHsearch and HHblits are the main programs of the widely used HH-suite software package [Ste+19], both based on the pairwise pHMM alignment tool HHalign, along with other useful scripts such as HHfilter which can filter MSAs by criteria like maximum sequence identity. These tools are also the baseline of the online server HHpred [SBL05] dedicated to homology detection and template-based structure prediction using structural models computed by MODELLER [ŠB93].

Evidence of HH-suite's ability to retrieve remote homologs with a higher sensitivity than its contemporary sequence alignment-based homology search methods was provided in template-based protein structure prediction experiments as parts of the worldwide protein structure prediction experiment CASP ("Critical

Assessment of protein Structure Prediction") which delivers an independent assessment of the current state-of-the-art in protein structure prediction every two years. This template-based modeling competition consists in predicting the structure of a protein, whose experimental 3D structure was not released yet, given its sequence using detected homologues as structural templates. CASP8, CASP9 and CASP10 in resp. 2008, 2010 and 2012 ranked HHpred among the best methods [Coz+09; Mar+11a; Hua+14]. Its most serious rivals in this competition are mainly meta-predictor approaches and, as of CASP13 in 2018, methods incorporating long distance information such as predicted contacts and inter-residue distances [Cro+19]. By their very nature, pHMMs cannot model such information, since they are strongly positional models. In the next section, we provide biology-based arguments to support the hypothesis that taking these long-distant dependencies into account when searching for homologs is relevant, and describe the model we propose to use as an alternative to profile Hidden Markov Models.

Chapter 3

Towards homology search with models capturing distant dependencies

In the last chapter, we described homology search approaches based on residue conservation: pairwise sequence, profile, and pHMM alignment methods, where assumption is made that positions are independent. In this section, we question the relevance of this assumption and argue that, though positional models can be effective approximations, valuable information on a protein shape and functions is lost when disregarding coevolutionary relations. We describe existing alignment-based homology search methods taking these distant dependencies into account, all based on alignments of Markov Random Fields. Finally, we detail the properties and background story of the Potts model, the Markov Random Field that we propose to use in our own alignment-based homology search method.

3.1 Biological arguments for taking co-evolution into account

As reminded in the first section, during the course of evolution a protein undergoes random mutations. So far, we only described mutations as position-specific events leading to variability in homologous proteins. In reality, the function is maintained

collectively by all residues. Sometimes, a single mutation which in itself would result in a loss of function occurs and can be rescued by *compensatory mutations* making up for the change, allowing the cell to survive and the mutations to be transmitted to future generations [Neh94]. In this manner, if two residues are particularly dependent on each other and subject to specific local constraints such as proximity in the folded structure, they will *co-evolve* to keep meeting these constraints. For instance, if a positively charged residue mutates into a negatively charged one, the mutation of its negatively charged neighbor will maintain the salt bridge.

Most of these co-evolving positions correspond to *direct contacts* in the folded structure [Ani+17]. Conventionally, two residues are said to be in contact if the distance between the first carbon atoms of its side-chains – their beta carbons C_β (or alpha carbons C_α for glycine which doesn't have a side chain) – is smaller than a given threshold, which may vary between 6 and 12 Angstrom, the commonly used threshold being 8 Angstrom. The fact that contacts and co-evolving positions are closely related is not surprising considering that two contacting residues are subject to significant compatibility constraints such as electrical charge, size or polarity (see figure 3.2 for an example of co-evolving positions associated with residues in contact). Constraints can also arise from ligand-mediated interactions [Mor+11], where two residues interact via a third molecule and thus co-evolve to maintain the binding. Furthermore, as demonstrated in [Ani+17], a frequent reason for a strong co-evolution between residues besides contact is the existence of *homo-oligomeric interfaces*: several copies of a domain are assembled into an oligomer (see example figure 3.1) and residues on different copies interact to maintain the structure.

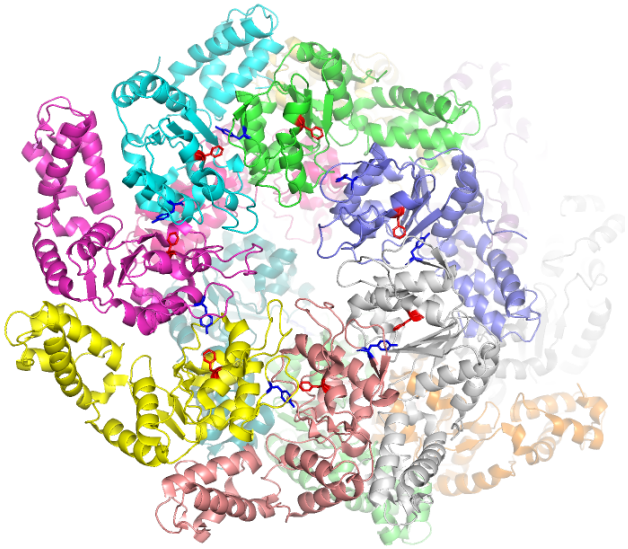
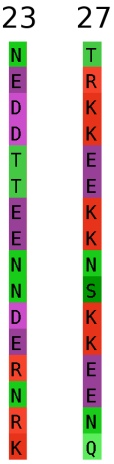


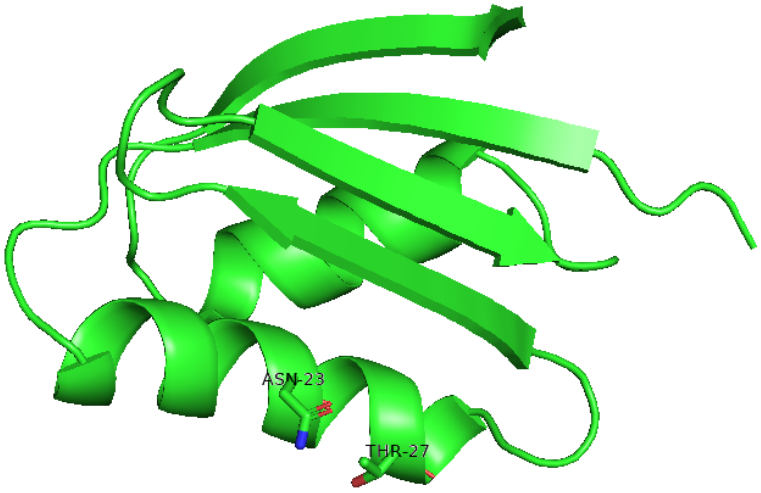
Figure 3.1 – Example of interdomain co-evolving positions in the oligomer Sigma54 interaction domain of protein NtrC1 of A. aeolicus (PDB 1NY6) as predicted in [Mor+11]. Residues 226 (in red) and 261 (in blue) co-evolve to maintain an interdomain contact which maintains the oligomer structure.

At the sequence level, co-evolution translates into correlated positions in multiple alignments of homologous sequences. An example is given figure 3.2. The direct contact between positions 23 and 27 constrains the probabilities of two letters to be seen together at positions 23 and 27 in the MSA: residues cannot have the same electrical charge.

Figure 3.2 – Example of co-evolving residues in the Atx1 metallochaperone protein of Saccharomyces cerevisiae (PDB 1CC8) corresponding to direct contacts.



(a) Columns extracted from a multiple sequence alignment of close homologous sequences of 1CC8, corresponding to positions 23 and 27 in 1CC8



(b) 3D view of 1CC8 and the positions 23 and 27, in contact in the folded structure

From these observations, it becomes clear that co-evolution provides key information on a protein function and shape. Instead of only modeling residue composition in columns of a multiple sequence alignment, one should also look into compatibility between residues at all positions. Unfortunately, by nature, positional models like profile Hidden Markov Models cannot reflect such dependencies. To take them into account in the homology search, one needs to look towards more expressive models, such as Markov Random Fields.

3.2 First attempts to model distant dependencies and perform homology search using pairwise Markov Random Fields

As explained in the previous section, distant dependencies reflect valuable information on proteins. Since profile Hidden Markov Models and sequence profiles cannot capture them, models of higher order should be used and, as a first approximation, models of order 2 have been proposed. These models, embedding both positional features and pairwise dependencies, go under the term of *pairwise Markov Random Fields*. After a brief general introduction to these models, we will describe three initial methods based on them. The first two approaches are mainly described for historical reasons and are less relevant to our study since they are sequence-to-model alignment methods based on structural information, while the last approach is, to our knowledge, the only existing method performing MRF-MRF alignment based on sequence information.

3.2.1 Introduction to pairwise Markov Random Fields

Markov Random Fields (*MRFs*) generalize Markov models in multiple dimensions. Where, in a Markov chain or Hidden Markov Model, each state depends only on the previous states, in Markov Random Fields each state depends on all of its neighbors.

More formally, as explained in [Kin80; WKP13; Wit+17], a Markov Random Field is built from an undirected graph $G = (V, E)$ where each node i in V corresponds to a random variable $X_i \in X$ and edges E represent dependencies between the variables, and satisfies the *local Markov property*, stating that each node is conditionally independent of any other node given its neighbors:

$$\forall i \in V, X_i \perp\!\!\!\perp X_{V \setminus \{i\}} | X_{N(i)}$$

where $N(i) = \{j | (i, j) \in E\}$ is the neighborhood of i.

According to the Hammersley-Clifford theorem, the probability of a field configuration x is a Gibbs distribution (also known as Boltzmann distribution)

which can be factorized over all cliques of the graph:

$$\mathbb{P}(X = x) = \frac{1}{Z} \prod_{c=1}^{C} \phi_c(x_c)$$

where $\phi_c(x_c)$ is the *potential function* of clique c and Z is the partition function ensuring $\sum_x \mathbb{P}(X = x) = 1$, i.e. $Z = \sum_{x \in X} \prod_{c=1}^{C} \phi_c(x_c)$.

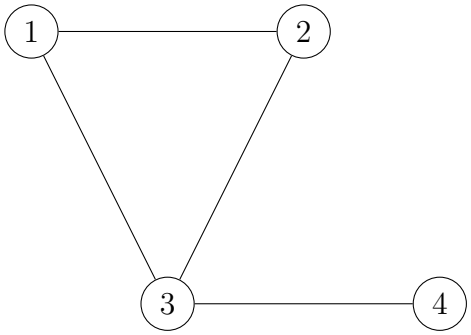


Figure 3.3 – Example of Markov Random Field built on a simple graph with 4 nodes, associating a random variable X_i to each node i. The graph has 9 cliques: $\{1,2,3\}$, $\{1,2\}$, $\{2,3\}$, $\{1,3\}$, $\{3,4\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, hence the underlying probability for a given configuration x has the following form: $\mathbb{P}(X = x) = \frac{1}{Z}\phi_{123}(x_1, x_2, x_3)\phi_{12}(x_1, x_2)\phi_{23}(x_2, x_3)\phi_{13}(x_1, x_3)\phi_{34}(x_3, x_4)\phi_1(x_1)\phi_2(x_2)\phi_3(x_3)\phi_4(x_4)$ where each ϕ_c is the potential for clique c

In this thesis, we focus on discrete pairwise Markov Random Fields, a subclass of Markov Random Fields where random variables are discrete – in our case they are amino acids – and only single and pairwise potentials are considered, disregarding higher order cliques. In that particular case, the probability function can be written as:

$$\mathbb{P}(X = x) = \frac{1}{Z} \prod_{i=1}^{L} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{i,j}(x_i, x_j)$$

where L is the length of the protein or protein alignment considered, E is the set of edges considered, and ϕ_i and $\psi_{i,j}$ are single and double potential functions.

.

3.2.2 Early work: Markov Random Fields as protein threading templates

Early efforts to use Markov Random Fields to model proteins originate from the *protein threading* field. Protein threading aims at recognizing the fold of a given protein by "threading" a target sequence into different structural models called "templates" from a template library usually built from PDB structures and compare their relative fitness scores. This problem is akin to alignment-based homology search but is oriented towards structures: models are built using structural features and the scoring function relies on knowledge of the relation between sequence and structure rather than similarity between sequences. Protein threading is a vast field of study and many different approaches have been proposed [PX11; SBL05; XX00; WZ08; Zhe+19; Yan+11; Xu+14; Du+20].

In 1994, White et al. [WMS94] proposed to use Markov Random Fields as protein core templates for non-homologous protein domains with a common core contact topology. Residues in the core are considered as observations of a pairwise MRF built from a graph (V, E) whose nodes are positions in the template and edges are specified by the contact graph.

Its probability distribution can be written as:

$$\mathbb{P}(X = x) = \prod_{i \in V} p_{s(i)}(x_i) \prod_{(i,j) \in E} \frac{p_{s(i,j)}(x_i, x_j)}{p_{s(i)}(x_i) p_{s(j)}(x_j)}$$

where s is a "state function" associating each node i and edge (i, j) to a discrete state describing its environment in terms of secondary structure and potential exposure to solvent, and to each state is associated a probability distribution built from single and pairwise amino acid frequencies in reference cores from PDB for each state.

The scoring function for an entire primary sequence is a log-likelihood including the core MRF probability distribution and out-of-core segments modeled as independent and identically distributed random variables. In this paper, authors do not provide an alignment algorithm but propose to rely on generic optimal threading algorithms. The threading problem was proven to be NP-hard [Lat94], but a first Branch-and-Bound approach tractable for small proteins was already

available at that time [LS94], and more efficient algorithms have been proposed since, including a Lagrangian-based approach [Yan+08] – a predecessor to the solver we will use in our own approach.

3.2.3 Augment profile Hidden Markov Models with dependencies between beta strands: SMURF

In 2010, Menke et al. proposed to model structural information with Markov Random Fields, this time not for protein threading but for homology detection purposes, with their method SMURF ("Structural Motifs Using Random Fields") [MBC10]. SMURF generalizes Hidden Markov Models by allowing dependencies between beta strands, turning them into pairwise Markov Random Fields particularly suited for β-structural motifs.

Models are built on multiple structure alignments made with Matt [MBC08]. So-called *β-strand match states* are decided by looking at the proportion of residues that are part of a β sheet at each position. Edges are drawn between paired β-strands and assigned with pairwise probabilities derived from frequency tables [Bra+01] according to a "buried" or "exposed" predicted states of the positions. The rest is modeled by a simple pHMM trained on the multiple sequence alignment derived from the multiple structure alignment, while no insertions or deletions are allowed in β-strand match states.

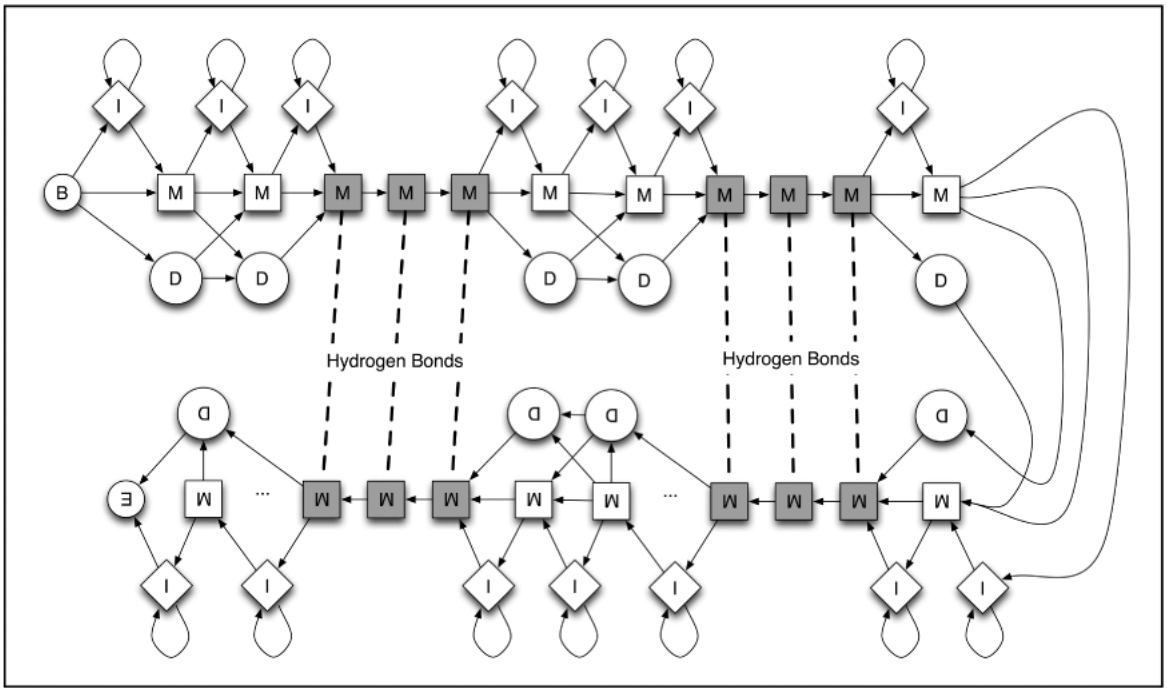


Figure 3.4 – Architecture of a SMURF MRF, taken from [Dan+14]

The score of a given target sequence is a combination of a pHMM score and a pairwise score. The probability of assigning residue r_k to the next match state m_j is given by:

$$\mathbb{P}(m_j|r_k, u_{j-1}) = \mathit{HMM}(m_j, r_k) \times \mathit{transition}(u_{j-1}, m_j) \times \beta\text{-}\mathit{strand}(r_j, r_k, m_j, m_k)$$

where u_{j-1} is the previous state, $\mathit{HMM}(m_j, r_k)$ is the pHMM probability of emitting amino acid r_k at state m_j, and $\beta\text{-}\mathit{strand}(r_j, r_k, m_j, m_k)$ is the pairwise component, calculated if m_j is a β-strand state paired with a match state m_k earlier in the template. Pairwise components are only computed for the second pair of each β-strand once the first residue is fixed. This implies to keep track of the possible subproblem solutions, which cannot be done with simple dynamic programming but rather with *multidimensional dynamic programming*.

SMURF was successfully applied to recognize β-propeller shapes. It significantly outperformed HMMER on leave-family-out cross-validation prediction experiments on the six, seven and eight-bladed β-propeller

superfamilies, and identified a potential class of hybrid two-component sensor YYY proteins by chaining propeller templates.

Two methods were proposed to enhance SMURF: SMURFLite [Dan+12] makes use of a simulated evolution method [KC10] to augment limited training data with artifical data using prior information about pairwise dependencies in beta sheets and simplifies MRF by setting a maximum number of unrelated beta strands between two paired beta strands to reduce computational complexity, and MRFy [Dan+14] proposes to reduce this complexity as well but with a stochastic search approach.

SMURF's superiority over HMMER in propeller-fold prediction suggests that taking into account pairwise dependencies improves homology search, at least for the case of β-structural motifs.

3.2.4 MRF-MRF alignment with all pairwise dependencies: MRFalign

In 2014, Ma et al. introduced MRFalign [Ma+14], a sequence-based homology search method performing MRF-MRF alignments with nearly full pairwise MRFs.

Unlike previous methods, MRFalign does not rely on structures and builds Markov Random Fields from multiple sequence alignments, in practice obtained using PSI-BLAST [Alt+97] from a primary sequence. Our interpretation of this approach is that each node in the MRF represents a column in the MSA, and its associated node potential function is the probability distribution of amino acids in the column. Edges are drawn between every residue pair (i, j) where $|i - j| \geq 6$, and the associated edge potential function is the *mutual information* between the two columns X_i and X_j:

$$MI(X_i, X_j) = \sum_{x_i \in X_i} \sum_{x_j \in X_j} \mathbb{P}(x_i, x_j) \log \frac{\mathbb{P}(x_i, x_j)}{\mathbb{P}(x_i)\mathbb{P}(x_j)}$$

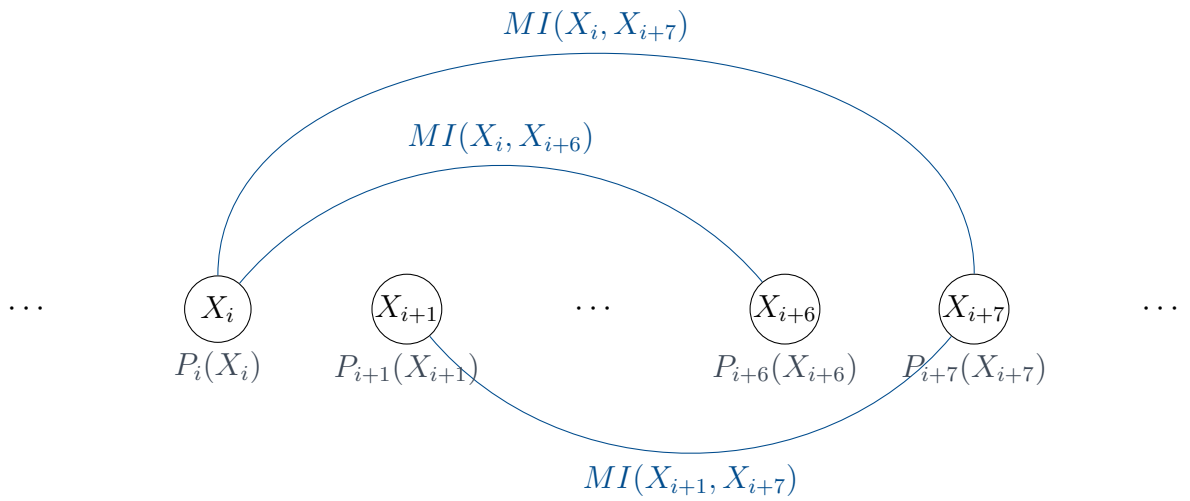


Figure 3.5 – Illustration of a Markov Random Field in MRFalign. Each node represents a random variable X_i associated with position i in the multiple sequence alignment. The node potential function is the probability distribution of the column. Edges are drawn between nodes separated by 6 positions or more, where the edge potential function is the mutual information between two columns in the MSA.

MRFalign performs MRF-MRF alignment, supposedly improving sensitivity over sequence-MRF alignment. The alignment of two MRFs is scored as a sum of a node alignment score and an edge alignment score for a given alignment path. The node alignment score is summed over all node pairs in the alignment path considered. Each term is computed using a neural network which inputs the "sequence profile context" of each node i, which is a matrix of marginal probabilities at positions in a window centered at i. Neural networks are trained on a set of structural alignments generated by DeepAlign [Wan+13]. The edge alignment score is derived from predicted inter-residue Euclidean distances for each pair of aligned nodes. Distances are predicted using a probabilistic neural network [ZX12] which inputs mutual information and its power series and sequence profile contexts, also trained on reference alignments generated by DeepAlign. The workflow seems to rely on several different tools such as HHmake [Ste+19], HHpred [SBL05], PSIPRED [MBJ00] and RaptorX-SS8 [Wan+10], though their use is not referenced in the publications. Given the node and edge alignment

score functions, the alignment is computed using ADMM ("Alternating Direction Method of Multipliers") which iteratively finds the optimal alignment alternatively without the edge alignment score or without the node alignment score.

Alignment accuracy was tested on custom datasets (not available anymore at this time) extracted from PDB with respect to reference structural alignments made by TM-align [ZS05], Matt and DeepAlign. MRFalign outperforms HMMER and HHalign in alignment precision and recall on all of these sets, especially at the superfamily and fold levels. Homology detection experiments were conducted at the superfamily and fold level on Söding's SCOP20, SCOP40 and SCOP80 sets [ABS12b] which are obtained by filtering the SCOP database [Mur+95] with a maximum sequence identity of respectively 20%, 40% and 80%. A MRF was built for each sequence, aligned to all other MRFs in the benchmark and the top 1, 5, and 10 ranked proteins are considered. In these experiments, MRFalign had a better success rate than HMMER's hmmscan, HHsearch, HHblits and FFAS.

MRFalign's results suggest that introducing pairwise dependencies may improve alignment quality and homology search at a more remote level for a larger number of proteins than those with a β-structural motif.

3.3 Capture direct couplings with the Potts model

In this section, we describe the Potts model, a subtype of pairwise Markov Random Field which we propose to use for homology search purposes. We briefly summarize the context in which it emerged and describe its properties.

3.3.1 Context: a need for a global statistical model to improve contact prediction accuracy

When no structural template are available, predicting a protein three-dimensional structure given its sequence is challenging. Using the finding that a protein fold can be solved with sufficient accurate information on its residue-residue contacts

[GS04; Göb+94; OV97], one approach is to perform *contact-guided protein folding*. As suggested by [LZS04], for 200-residue long single-domain proteins, knowing only one true contact every 8 residues is enough to guide the simulations towards correct folds. The challenge lies in the accuracy of these predicted contacts.

Among sequence-based contact prediction methods, typical approaches are statistical methods based on coevolution. Indeed, as explained in section 3.1, direct contacts account for a significant part of co-evolving residues, thus the idea of such methods is to detect correlations between columns in a multiple sequence alignment built with the target sequence and its close homologs, hoping to identify co-evolving positions and, by extension, contacts (see figure 3.6).

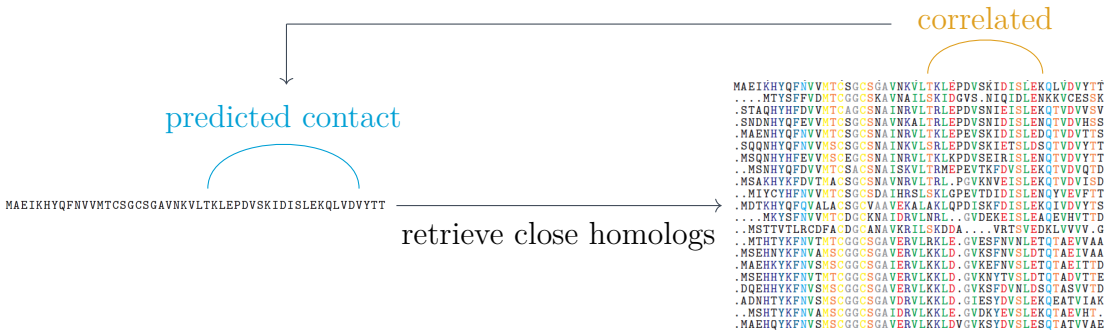


Figure 3.6 – Coevolution-based approach to predict contacts. The target sequence is enriched with close homologs into a multiple sequence alignments and positions whose columns that are identified as "correlated" are predicted as structural contacts.

Early attempts proposed *local* measures of covariation, considering each pair of columns independently. Such methods make use of χ^2 statistics [FA04], prior knowledge on amino acids mutational propensities with the McLachlan substitution matrix [Göb+94; ORV99], and the most commonly used methods [Jon+12] are based on mutual information (MI). Several MI-based methods have been proposed [Atc+00; TL03; GP07] including corrections to reduce background biases caused by phylogenetic relations and entropic effects, the most widely used being Average Product Correction (APC) [DWG08], which substracts a so-called APC term: $MI_{ij}^{APC} = MI_{ij} - \frac{\bar{M}_i \bar{M}_j}{\bar{M}_{ij}}$ where $\bar{M}_i = \frac{1}{N} \sum_{k \neq i}^{N} M_{ik}$ and $\bar{M}_{ij} = \frac{1}{N(N-1)} \sum_{i \neq j} M_{ij}$.

However, these methods yield high false positive rates [AP15]. This can be explained by their inability to deal with *transitive correlations*: as illustrated in figure 3.7, if two residues i and j both co-evolve with a third residue k, an indirect correlation signal will arise between positions i and j. This phenomenon is also known as the *chaining effect*.

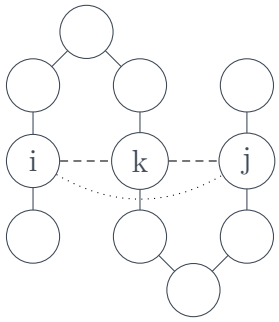


Figure 3.7 – Illustration of a transitive correlation. Residue k is in contact with residue i and residue j, which generates an indirect correlation signal between i and j.

The fact that correlations are chained make it difficult to predict direct coevolutionary relations using only local measures. Accurately predicting co-evolving position requires methods that take into account the entire interaction graph to disentangle direct couplings from indirect ones.

3.3.2 Emergence of Direct Coupling Analysis

A way to disentangle direct from indirect couplings was proposed by Lapedes et al. in 1999 [Lap+99]. They noticed that the chaining effect was analogous to the problem of "correlation at a distance" in spin systems and proposed to adapt the corresponding statistical physics model representing spins with discrete variables (states reflecting spins) to proteins (states reflecting amino acids). This model is called *Ising model* or *Potts model*, a pairwise Markov random field derived from the maximum entropy principle which will be fully described in the next section. The approach consists in inferring a Potts model on a multiple sequence alignment of the target sequence and close homologs and using its inferred edge potential parameters to predict contact likeliness. As opposed to previous methods based on

local measures, the Potts model is a global statistical model of the entire MSA, and the maximum entropy constraint theoretically implies that it consistently generates observed statistics with as little bias as possible and that its single and pairwise potentials parsimoniously explain the observed correlations – in other words, it yields a set of direct couplings between positions, disentangling the chaining effect.

Though Lapedes et al. were the first to suggest the use of the Potts model to predict direct co-evolutionary relations, their work remained theoretical, involving prohibitively time-consuming calculations – inferring a Potts model is a computationally expensive task, as we will see – and at that time multiple sequence alignments used to train the model were not deep enough since there were significantly less sequences available in the databases. Consequently, implications of their contribution went unnoticed.

A decade later, the Potts model was brought to the forefront by Weigt et al. [Wei+09] in their direct interaction prediction method termed *Direct Coupling Analysis* (DCA). Inference was made more tractable using a message-passing approximation. A further impulse was given in 2011 with a mean-field approximation of the model [Mor+11] significantly reducing the complexity and broadening its reach to practically all Pfam families. Since then, plenty of inference methods have been proposed [Eke+13; KOB13; SGS14; Bal+14; Bar+16; FBW18; Sut+15] and will be discussed in more details in section 4.2.

Direct Coupling Analysis had a remarkable impact on the field of contact prediction. Spectacular performances were achieved in 2016 during the CASP11 contact prediction contest [Mon+16]. Hundreds of protein structures have been predicted thanks to contacts predicted by DCA [Mar+11b; Dag+12; NJ12; Ovc+17; Hop+15; Hay+15], including membrane proteins [Hop+12] whose structures are hard to determine via crystallography methods.

DCA was also successfully applied to protein-protein interaction prediction [Sch+09; Gue+16; Hop+14; OKB14; Bit+16], prediction of mutational effects [Hop+17; Che+16; Fig+16; But+16; Man+14; Fly+17], and was also extended to RNA structure prediction [De +15; CTB20].

3.3.3 Potts model on proteins: definition and properties

In the context of proteins, Potts models are pairwise Markov Random Fields whose random variables are residues corresponding to positions in the multiple sequence alignments they model. In contrast with other MRFs on proteins described in the previous sections, whose node and edge potential functions are defined separately with local measures such as mutual information, the Potts model is globally defined as the probability distribution whose marginals respect empirical single and double frequencies in the MSA and complies to the maximum entropy principle [Jay57]. In other words, the underlying distribution reflects observed frequencies in the data while being as simple as possible. Let us give a more formal definition before deriving a practical interpretation that will be extensively used throughout this work.

3.3.3.1 Formal introduction

Let X be a MSA of length L with N sequences $\{x^1, \cdots, x^N\}$ on an alphabet $\mathcal{A}$ extended with a new gap character '$-$'. We denote by q the size of the alphabet.

A Potts model with q states for MSA X can be defined as a statistical model whose probability distribution P over all sequences of length L maximizes the Shannon entropy $H(P) = -\sum_{y \in \{1,\cdots,q\}^L} P(y) \log P(y)$ and generates the empirical single and double frequencies of the MSA as marginals:

$$\forall i = 1, \cdots, L, \forall a = 1, \cdots, q, \sum_{\substack{y \in \{1,\cdots,q\}^L \\ y_i = a}} P(y) = f_i(a) = \frac{1}{N} \sum_{n=1}^{N} \delta(x_i^n, a) \tag{3.1}$$

$$\forall i, j = 1, \cdots, L, \forall a, b = 1, \cdots, q, \sum_{\substack{y \in \{1,\cdots,q\}^L \\ y_i = a, y_j = b}} P(y) = f_{ij}(a, b) = \frac{1}{N} \sum_{n=1}^{N} \delta(x_i^n, a)\delta(x_j^n, b) \tag{3.2}$$

This corresponds to the distribution P that maximizes the following functional:

$$\begin{aligned} F(P,\lambda,\Omega) = H(P) \\ &+ \sum_i \sum_a \lambda_i(a)\left(P_i(a) - f_i(a)\right) \\ &+ \sum_{i,j} \sum_{a,b} \lambda_{ij}(a,b)\left(P_{ij}(a,b) - f_{ij}(a,b)\right) \\ &+ \Omega\left(1 - \sum_x P(x)\right) \end{aligned} \tag{3.3}$$

where λ_i and λ_{ij} are Lagrange multipliers associated with constraints on the single and double frequencies and Ω is the Lagrange multiplier ensuring that P is a probability distribution. This functional has a global maximum which is a full pairwise Markov Random Field whose probability distribution has the following form:

$$\mathbb{P}(X = x) = \frac{1}{Z} e^{-\mathcal{H}(x)} \tag{3.4}$$

where $Z = \sum_y e^{-\mathcal{H}(y)}$ is the normalization constant and $\mathcal{H}$ is the energy or Hamiltonian defined by:

$$\mathcal{H}(x) = -\left(\sum_{i=1}^{L} v_i(x_i) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} w_{ij}(x_i, x_j)\right) \tag{3.5}$$

where v_i are vectors of length q termed *fields* (or *biases*) and corresponding to the Lagrange multipliers associated with constraints on single frequencies, and w_{ij} are $q \times q$ matrices termed *couplings* corresponding to Lagrange multipliers associated with constraints on double frequencies.

This global maximum is unique [Lap+99], up to a *gauge invariance*. Indeed, constraints 3.1 and 3.2 are not independent since they sum to 1: the model is over-parameterized. Hence, for example, adding an arbitrary constant to all $v_i(a)$ in the same column i will yield the same probabilities. To fix this indeterminacy, a so-called *gauge* is chosen. Gauge choices will be discussed in section 4.5.1.

Given the functional form, the parameters v and w of the Potts model are the ones that maximize the likelihood of the sequences in the multiple sequence

alignment:

$$\mathcal{L}(v, w|X) = \prod_{n=1}^{N} \mathbb{P}(x_1^n, \cdots, x_L^n) = \prod_{n=1}^{N} \frac{1}{Z} \exp\left(\sum_{i=1}^{L} v_i(x_i^n) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} w_{ij}(x_i^n, x_j^n)\right) \tag{3.6}$$

Likelihood maximization is intractable, since it would imply the computation of the normalization constant Z at each iteration. For that reason, several approximations have been proposed and will be reviewed in section 4.2.

A representation of a small Potts model of length 4 is given figure 3.8.

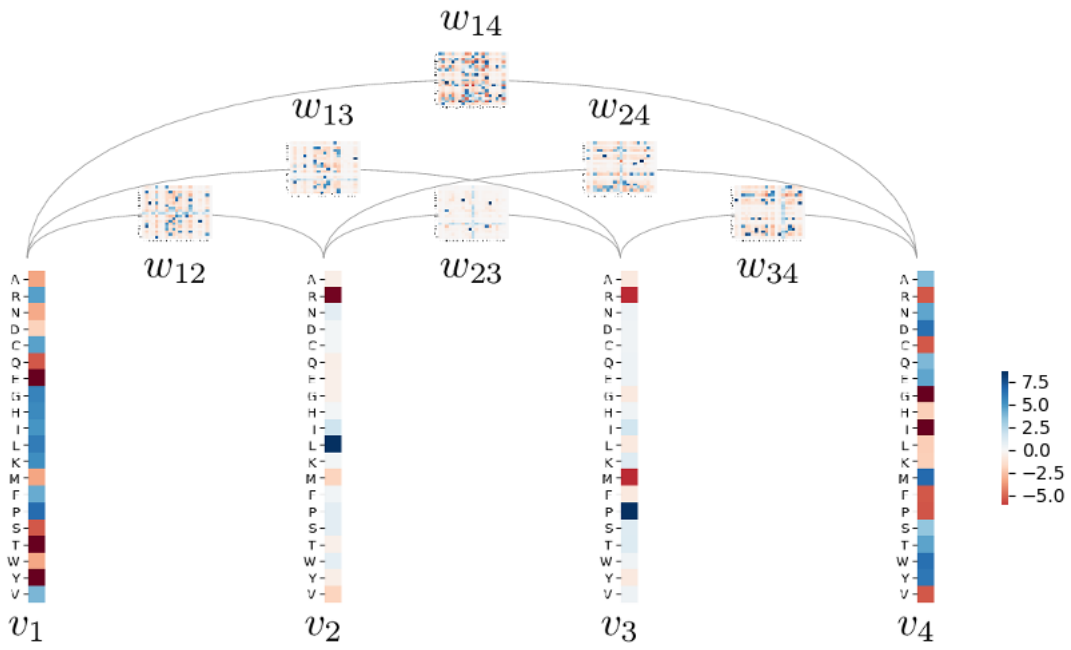


Figure 3.8 – Example of a Potts Model of length 4. For each position $i = 1, .., 4$ the node potential is a vector v_i of length q. All pairs of positions (i, j) are connected by an edge, associated with an edge potential which is a $q \times q$ coupling matrix $i \times j$

Note that, without loss of generality, a Potts model can also be defined for RNA sequences using the RNA alphabet $\{A, C, G, U\}$.

3.3.3.2 Practical interpretation

A key point is that, since the MRF potentials v and w are Lagrange multipliers ensuring the constraints, their values inferred on a multiple sequence alignment have a practical interpretation. For a position i and a letter a, $v_i(a)$ can be seen as the rate of change in the likelihood of the MSA as a function of the single frequency $f_i(a)$ of amino acid i at position a, and a similar interpretation can be given to $w_{ij}(a, b)$. As a consequence, the Potts model parameters can be described as follows:

- *Fields* v_i are weight vectors reflecting positional conservation, where $v_i(a)$ tends to be positive if a is particularly conserved in the column and negative is a is particularly deficient at position i.
- *Couplings* w_{ij} are pairwise interaction matrices between two positions i and j where $w_{ij}(a, b)$ is a weight quantifying the compatibility of having a and b together at positions i and j: it tends to be positive if it is likely to find a and b together at positions i and j, negative if a and b are found at positions i and j but not together, and null in other cases.

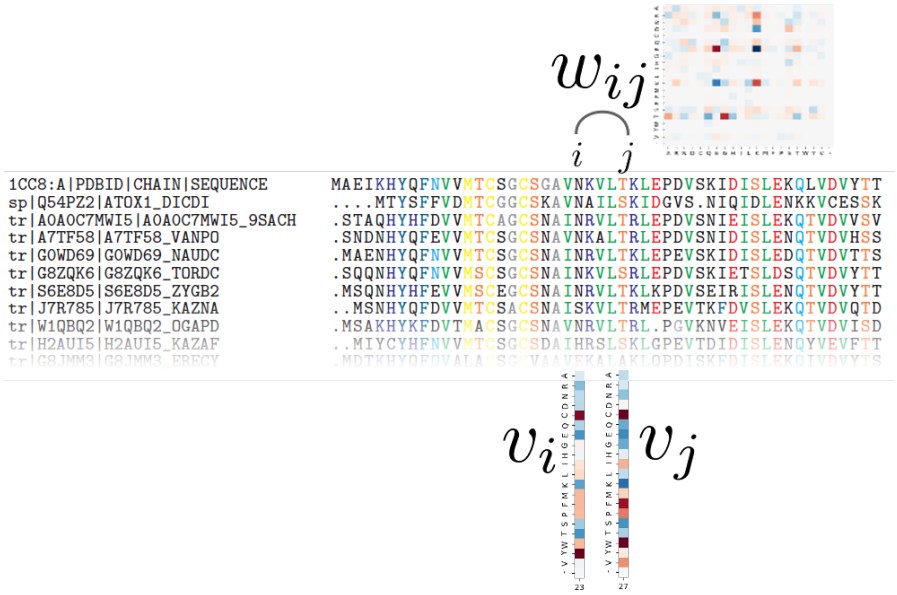


Figure 3.9 – A Potts model's parameters v and w are inferred on a multiple sequence alignment and reflect positional conservation at each position (v_i) and direct couplings between positions (w_{ij}).

Following this interpretation, couplings inferred on a multiple sequence alignment can be regarded as indicators for a direct co-evolutionary relation. Contact prediction methods compute the Frobenius norms of the coupling matrices:

$$\|w_{ij}\| = \sqrt{\sum_{a=1}^{q}\sum_{b=1}^{q} w_{ij}^2(a,b)}$$

considering that, the higher the norm – after having applied a correction such as APC described in 3.3.1 – the likelier it is that positions i and j are co-evolving, and thus probably in contact.

3.4 On the relevance of Potts models for homology search

Markov Random Field based approaches depicted in section 3.2 suggested that taking distant dependencies into account may improve homology search. We

argue that, among other MRFs, DCA's Potts model in particular emerges as an interesting alternative.

Though it was designed to predict direct contacts and not for the purposes of modeling homologs, its parameters can describe both positional conservation and distant dependencies. Furthermore, as opposed to MRFalign whose edge potentials are local measures of mutual information, the Potts model's coupling parameters were inferred jointly with the fields parameters in the process of finding the distribution that best represents the data. This gives a valuable interpretation to its edge potentials: they reflect direct interactions between positions.

Interestingly, as demonstrated in [Cho+20], nowadays these direct couplings may not be used for contact prediction as accurately as more recent methods based on neural networks, but the strongest couplings have more *evolutionary significance* than contacts predicted with other methods: they have on average more bonds, are more widely distributed in the structure, and involve more supersecondary interactions. Their findings suggest that direct couplings contain valuable biological information on the protein, beyond structural proximity.

Its parameters interpretation in terms of residue conservation and direct couplings between positions, backed by above-mentioned studies on their biological relevance, suggest that the Potts model, beyond its application in contact prediction, might be a suitable model to represent sets of homologous sequences.

3.5 Sequence to Potts model alignment methods

Reasons explained above led us to propose to use the Potts model for homology search purposes. This idea was proposed at the same workshop in 2019 simultaneously by us [TC19] and by Muntoni et al. [Mun+19]. Their heuristic for sequence-Potts model alignment was later described in a pre-print published in 2020, which we review in the first section. Another pre-print was published the same year by Wilburn and Eddy who propose a heuristic to align sequences to hybrid models termed *hidden Potts models* combining Potts models and profile Hidden Markov Models, which we describe in the last section.

3.5.1 DCAlign: statistical physics inspired models to represent alignments

Muntoni et al. [Mun+20] propose to align a sequence to a Potts model by representing each possible alignment with a statistical physics inspired model and finding the best one using an approximate message-passing strategy.

The alignment of a sequence $A = (A_1, \cdots, A_N)$ to a Potts model for a protein domain $S = (S_1, \cdots, S_L)$ is described by a discrete model $(\boldsymbol{x}, \boldsymbol{n})$ where:

- $\boldsymbol{x} = x_1, \cdots, x_L$ is a sequence of *spins* x_i indicating if S_i is a match or a gap
- $\boldsymbol{n} = n_1, \cdots, n_L$ is a sequence of *pointers* where n_i points to the position in A corresponding to S_i

A *Boltzmann weight* is associated to each possible alignment $(\boldsymbol{x}, \boldsymbol{n})$:

$$W(\boldsymbol{x}, \boldsymbol{n}) = \frac{1}{Z} e^{-\mathcal{H}_{DCA}(\boldsymbol{x},\boldsymbol{n})} \chi_{in}(x_1, n_1) \chi_{end}(x_L, n_L) \prod_{i=2}^{L} \chi_{sr}(x_{i-1}, n_{i-1}, x_i, n_i) \prod_{i=1}^{L} \chi_{gap}(x_i, n_i)$$

where $\mathcal{H}_{DCA}$ is the Potts model Hamiltonian for the subsequence $(A_{x_1 \cdot n_1}, \cdots A_{x_L \cdot n_L})$ corresponding to the alignment, and χ_{in}, χ_{end}, χ_{sr}, χ_{gap} are energetic contributions associated with gap treatment. More specifically:

- χ_{gap} handles the cost of gaps in the target sequence, defined in a position-specific way:

  $$\chi_{gap}(x_i, n_i) = e^{-1(1-x_i)\mu(n_i)}$$

  where $\mu(n_i)$ can take two different values μ^{ext} or μ^{int} according to whether the gap is "internal" or "external", values are learned in a supervised way

- χ_{sr} handles the cost of insertions in the target sequence, where an insertion is associated with a position-specific affine gap penalty energy:

  $$\chi_{sr}(1, n_{i-1}, 1, n_i) = e^{-(1-\delta_{n_i,0})\left(\lambda^i_o + \lambda^i_e(\Delta n_i - 1)\right)}$$

  where λ parameters are learned on the seed alignment by maximizing

likelihood of the data with a probability distribution:

$$P_i(\Delta n) = \begin{cases} \frac{1}{z}, & \text{if } \Delta n = 0 \\ \frac{e^{-\lambda_o^i - \lambda_e^i(\Delta_n - 1)}}{z}, & \text{otherwise} \end{cases}$$

where $z = 1 + \sum_{\Delta_n > 0} e^{-\lambda_o^i - \lambda_e^i(\Delta_n - 1)}$

Finding the best alignment ultimately means finding the variables $(\boldsymbol{x}^*, \boldsymbol{n}^*)$ that maximize the Boltzmann distribution, i.e.

$$(\boldsymbol{x}^*, \boldsymbol{n}^*) = \underset{(\boldsymbol{x}, \boldsymbol{n})}{\operatorname{argmax}} W(\boldsymbol{x}, \boldsymbol{n})$$

To solve this problem in a tractable way, this paper introduces a heuristic approach based on mean-field approximations.

Experimentations focused on DCAlign's application to building coevolutionary consistent multiple sequence alignments. 4 Pfam families were selected for the short length of the seed alignment (< 100 residues) and the large number of effective sequences (> 1000), and for each family a model (including a Potts model and gap parameters) was inferred on the seed alignment and additional sequences were aligned to this seed alignment. MSAs built this way were compared with MSAs built with HMMER in a similar way. In most cases, the resulting MSAs were similar. For one Pfam family though (PF00677), a large group of sequences were aligned differently by DCAlign and HMMER. Based on a further PCA analysis, authors suggested that HMMER probably miscategorized these sequences. With respect to RNA alignments from Rfam [Kal+18], DCAlign displayed performances similar to Infernal [NE13], a method based on Covariance Models which are probabilistic models with a Stochastic Context-Free Grammar architecture allowing pairwise emissions, enabling them to model both primary sequence and secondary structure.

Thanks to their efficient heuristics, the running time of DCAlign is roughly quadratic in the length of the sequence to be aligned and the length of the Potts model, with median computation times ranging from 7 to 48 seconds using a laptop computer. Interestingly, alignments seem to be faster with models trained on a

larger number of sequences, suggesting that more accurate Potts models allow for the target domain to be more easily detected.

3.5.2 Combining pHMMs and Potts models into hidden Potts models

Wilburn and Eddy [WE20] propose to merge Potts models and profile Hidden Markov Models into hybrid models termed *hidden Potts models* (HPM). HPMs benefit both from Potts model's pairwise dependencies and pHMMs' probabilistic treatment of insertions.

The architecture of the model is the same as that of a pHMM but without deletion states: the model is a chaining of match states corresponding to conserved columns in a MSA, interspersed with insertion states (see figure 3.10). The main difference with pHMMs is that letter emission probabilities in match states are not independent from each other but follow a distribution associated with a Potts model inferred on the MSA. Moreover, deletions are handled as gap characters in match states.

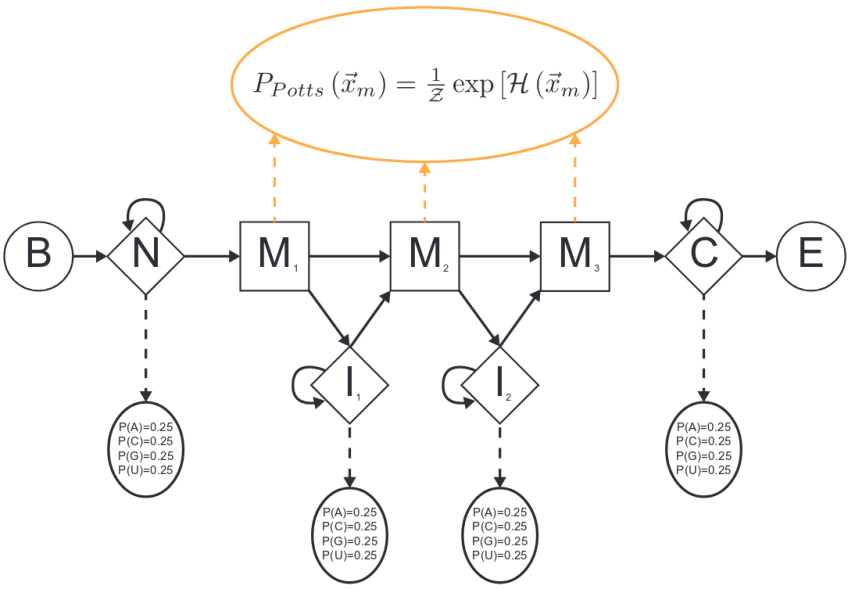


Figure 3.10 – Architecture of the Hidden Potts model, figure from the paper [WE20]

Since, due to distant dependencies, the alignment cannot be efficiently computed with dynamic programming, an approximated algorithm based on importance sampling is used: possible alignments are sampled using the pHMM part and re-scored and re-ranked under the HPM. With this approach, a sequence is typically aligned in roughly 1 minute.

The pHMM-like part and the Potts model-like part are trained separately: the pHMM part is trained using the HMMER package and the Potts models are inferred by GREMLIN [KOB13]. Since Potts model inference methods such as GREMLIN provide the field and coupling parameters but not the probability distribution – since this would imply the intractable computation of the normalization constant Z – probability distributions in HPMs are unnormalized, up to a constant Z.

Though the HPM formalization authors introduced in this paper is general and can be applied to proteins and RNA, it was only tested on RNA here. Their benchmark consists of 3 curated alignments based on known 3D structures, each randomly split into a training set and a positive test set, and augmented with

a negative test set of randomly generated sequences drawn from a distribution defined by the nucleotide composition of the positive test set. HPMs outperform HMMER in terms of alignment accuracy and homology search sensitivity, but are outperformed by Infernal. Authors explain the latter by a decade worth of parameter optimization on Infernal's side while Potts models' parameters are inferred with an approximate method. They conclude by saying that *future work is needed to optimize Potts model training for remote homology search rather than structure prediction alone.*

Summary

The content of this first background part can be summarized in a few key points. Proteins are complex macro-molecules implied in a variety of biological processes. Their structures and functions are determined by their primary sequences of amino acids, translated from DNA during the process of protein synthesis. Thanks to increasingly fast and cheap sequencing technologies, such sequences are extracted and fill protein sequence databases at an exponential rate which *in-vivo* and *in-vitro* protein sequence annotation methods cannot follow, raising a need for *in-silico* annotation methods. The most common way to annotate a sequence *in-silico* is to transfer annotations from its *homologs*, proteins with a common ancestor. Using the fact that sequence is conserved throughout evolution, when sequences did not diverge too much, homology can be detected by performing pairwise sequence alignment. However, the more remote the homologs, the more mutations separate them, lowering pairwise sequence identity and making it harder to provide reliant alignments and similarity scores. To detect more remote homologs, it becomes necessary to factor in mutation constraints resulting from evolutionary pressure to maintain function and structure. To this end, profiles, and later profile Hidden Markov Models, have been designed to capture amino acid conservation at positions in multiple sequence alignments, making it possible to compute a similarity score between a target sequence and a set of homologous proteins by sequence-pHMM alignment, and for even more sensitivity by pHMM-pHMM alignment to factor in mutation constraints on the query side as well. These methods can detect a wide range of homologs while relying on the simplifying assumption that mutations occur at each position independently.

In this thesis, we raise the question of whether taking pairwise co-evolution into account could improve remote homology detection. We provided biological

arguments indicating that valuable information on protein shape and function is lost when disregarding these pairwise dependencies, and described early methods based on Markov Random Fields. Among them, a pairwise Markov Random Field alignment method termed MRFalign showed particularly encouraging results suggesting that pairwise dependencies might improve the detection of remote homologs. While remarkably efficient, this approach, relying on neural networks and on a complex workflow computing single and double potentials of the MRFs independently, lacks interpretability. We argue that the Potts model, a global statistical model grounded on the maximum entropy principle, would be a suitable candidate to perform remote homology search with an ideal of interpretability. Introduced by Direct Coupling Analysis in the context of contact prediction, this model was proven to capture direct co-evolving positions with an unprecedented accuracy, and here we raise the question of whether this model could improve homology search as well. In parallel to our work, pre-prints for two methods performing sequence to Potts model alignments based on heuristics were posted on bioRxiv, showing encouraging preliminary results. Yet, when it comes to remote homology detection, model-model alignment methods were shown to be more sensitive than sequence-model alignment methods. Inspired by the performances in remote homology detection of HHsuite, whose core component is the pairwise pHMM alignment method HHalign, we developed PPalign, a method for pairwise Potts model alignment presented in the following chapters.

Part II

Contributions

In the previous chapter, we described the Potts model as introduced by Direct Coupling Analysis and proposed to use it for homology search purposes. In this thesis, inspired by HHsuite which relies on its core pHMM-pHMM alignment method HHalign to perform remote homology detection, we introduce PPalign, our Potts model to Potts model alignment method, intended to be the core component of a future PPsuite package for homology search using direct coupling information.

Here, we propose to build Potts models to represent protein sequences by enriching them with their close homologs to capture allowed variability around them and acquire covariation signal, and to compute their optimal alignment and similarity with our method PPalign. Our approach is summarized in the workflow figure 3.11.

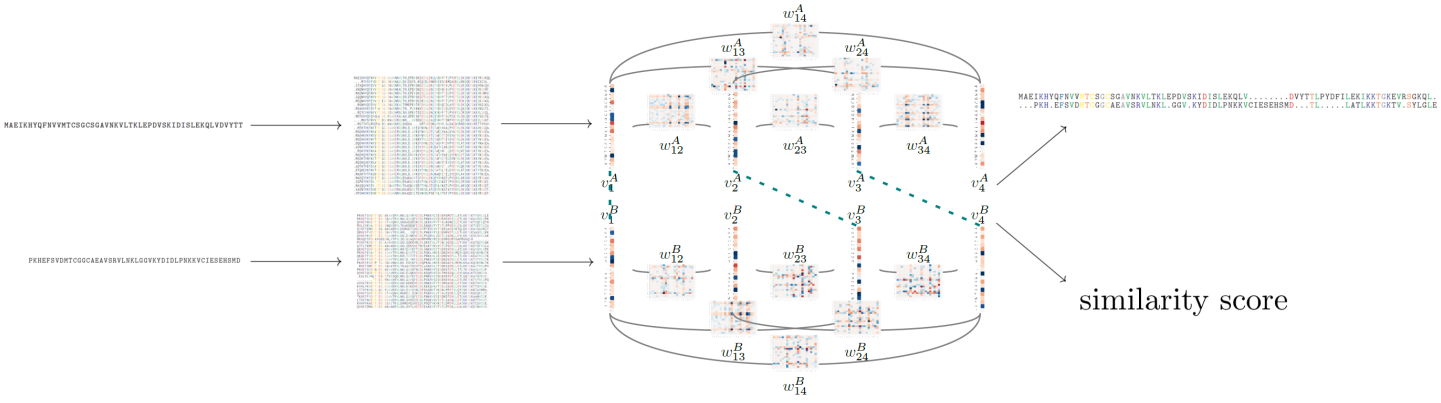


Figure 3.11 – Workflow diagram of our approach. First, protein sequences are enriched with close homologous sequences to capture allowed variability around them and acquire covariation signal. Then, Potts models are inferred on multiple sequence alignments of these close homologs. The two Potts models are then aligned with our method PPalign, providing a similarity score and a pairwise sequence alignment.

The construction and the alignment of Potts models are two strongly intertwined problems, and their study required several back-and-forths between them. We discuss them separately here, following the order the workflow.

This part is organized as follows. Chapter 4 addresses the challenge of building Potts models able to properly represent proteins in a comparable way.

Indeed, existing inference methods and workflows to build Potts models were originally designed for direct interaction prediction purposes and do not necessarily comply with these new requirements, which are critical to pairwise Potts model comparison. In this chapter, we attempt to identify directions and factors one can act upon towards more canonical Potts models, propose an operational workflow and discuss areas for improvement. Chapter 5 describes our optimal Potts model alignment method and its performances. Two major challenges are addressed in this chapter: the definition of an appropriate score for the alignment of two Potts models, and its efficient optimization with respect to constraints for a proper alignment despite the NP-hardness nature of the problem. Our alignment method's performances were assessed on a selected set of reference alignments with low sequence identity, on which optimal alignments where found in tractable time and direct couplings were shown to substantially improve the quality of some alignments with lowest sequence identity. Finally, encouraging preliminary results on homology detection are reported in chapter 6.

Chapter 4

Towards canonical Potts models

This chapter marks a first step towards homology search through pairwise Potts model comparison by addressing the question of properly representing a protein with a Potts model, in an effort towards the need for canonicity raised by our Potts model to Potts model pairwise alignment approach. The first section attempts to identify expected characteristics of a canonical Potts model, and the following sections investigate various factors one can act upon to pursue these directions. Five key topics are addressed: choice of an inference method, input data to train a Potts model representing a protein, gauge and priors on the model to be inferred, and finally specific concerns regarding the comparability of positional parameters and of coupling parameters are examined in more details.

4.1 From contact prediction to homology search: new requirements for canonical Potts models

As stated before, inference of protein Potts models was originally intended for the prediction of co-evolving positions, with applications including contact prediction and protein-protein interactions. To our knowledge, no previous work addresses the question of how to infer Potts models that will properly represent given sets of homologous proteins and be comparable to each other. Yet, such different goals conceivably imply different choices in design and training – especially since, due to computational complexity, the inferred Potts model will necessarily be

an approximation. Therefore, it seems sensible to guide inference towards the approximation that will best suit one's needs. Existing work mainly focuses on maximizing direct coupling prediction accuracy: emphasis is mainly placed on coupling norms, while pertinence of positional parameters v_i and individual coupling matrix values $w_{ij}(a, b)$ is not guaranteed. In contrast, using Potts models for homology search purposes requires all parameter values to be appropriately set and balanced.

First of all, it seems important to make sure that the parameters of the resulting Potts model can actually be interpreted as described before:

- $v_i(a)$ should be significantly positive if residue a is significantly conserved at position i, significantly negative if a is significantly deficient at position i, and close to 0 if whether a is at position i or not does not give much information on the sequence's affiliation to the modeled protein set

- $w_{ij}(a, b)$ should be significantly positive if residues a and b are significantly often found together at position i and j, significantly negative if a and b should not be together at positions i and j, and null otherwise. In other words, $w_{ij}(a, b)$ should reflect strong direct correlations or anti-correlations indicating that positions i and j are co-evolving and how residues are concerned by this co-evolution, and should be close to 0 if no such direct co-evolution is detected.

Furthermore, just as higher norms for the coupling parameters should indicate important pairs of positions for the set of protein sequences being modeled – hopefully co-evolving positions – one should expect a position with a higher positional parameter norm to be particularly important for the characterization of the protein set. In other words, a v_i should have a high norm if v_i is informative in terms of amino acid conservation or deficiency and a low norm if it is a random column that would equally match any letter.

Finally, we propose that data should be explained using positional conservation as much as possible, increasing the coupling values only when the underlying signal cannot be explained otherwise. This should provide a more consistent representation without spurious couplings and favor more canonical models. Moreover, we expect this to speed up the alignment process.

4.2 Choice of an inference method

Theoretically, parameters of a Potts model representing a multiple sequence alignment $X = \{x^1, \cdots, x^L\}$ of length L are obtained by maximizing the likelihood function (equation 3.6). However, computing the normalization constant Z is intractable, and approximation methods have to be used.

Let alone the historically first tractable inference approach based on a message-passing approximation (mpDCA [Wei+09]) which is still computationally expensive and less accurate than subsequent methods, recent approximate inference methods can be roughly classified into 3 categories in terms of performances: less accurate but very fast methods relying on a matrix inversion, more accurate methods with reasonable computational cost based on pseudo-likelihood maximization, and methods that accurately reproduce empirical frequencies at a substantially higher computational cost.

The first category regroups mean-field approximations (mfDCA [Mor+11]) based on a small coupling expansion of the Potts model and Gaussian inference (gaussDCA [Bal+14]) where the Potts model is approximated by a Gaussian model in which variables representing amino acids can be continuous. These two approaches ultimately amount to inverting a $Lq \times Lq$ matrix, providing an interesting intuition on the coupling parameters:

$$w_{ij}(a, b) \simeq -(C^{-1})_{ij}(a, b) \tag{4.1}$$

where C is the empirical covariance matrix:

$$C_{ij}(a, b) = f_{ij}(a, b) - f_i(a) f_j(b) \tag{4.2}$$

Though these methods are remarkably fast, with a complexity of $O(L^3)$ - mfDCA sped up computation time by a factor of 10^3 to 10^4 with respect to mpDCA with even more accurate contact predictions, historically making it possible to apply DCA to nearly all Pfam families - they do not converge to the exact solution in the limit of infinite data, and are outperformed by subsequent methods in contact prediction.

Pseudo-likelihood methods, on the other hand, are statistically consistent

[KF09; Bes75], which means that the true Potts model is inferred in the limit of infinite data. Rather than maximizing the full likelihood in equation 3.6, this approach consists in maximizing a *pseudo-likelihood*, defined by

$$pll(v, w|X) = \prod_{n=1}^{N} \prod_{i=1}^{L} \mathbb{P}(X_i = x_i^n | x_{\setminus i}^n, v, w) \tag{4.3}$$

where its logarithm $\log pll(v, w|X)$ can be rewritten as:

$$\log pll(v, w|X) = \sum_{n=1}^{N} \sum_{i=1}^{L} \left(v_i(x_i^n) + \sum_{j \neq i} w_{ij}(x_i^n, x_j^n) - \log Z_i^n \right) \tag{4.4}$$

where $Z_i^n = \sum_a \exp\left(v_i(a) + \sum_{j \neq i} w_{ij}(a, x_j^n)\right)$ is a normalization constant computed over only L terms, making it easier to compute than the full normalization constant Z. This implies that the Potts model is not directly optimized on the empirical frequencies but on the full sequences in the train MSA, implying a complexity growing linearly with the number of sequences. The two primary pseudo-likelihood methods are plmDCA [Eke+13] and GREMLIN [KOB13], and an efficient implementation has been proposed with CCMpred [SGS14] which yields the same precision while being significantly faster.

The pseudo-likelihood approach has been shown to provide the highest contact prediction accuracy, substantially outperforming earlier methods[KOB13; Eke+13; Coc+18], and is thereby considered as the state-of-the-art DCA method in contact prediction. [MB19].

Finally, higher accuracy inference methods have been introduced, using two main approaches: Boltzmann machine learning, which iteratively computes the gradient of the likelihood function by Monte-Carlo simulations (implemented in bmDCA [FBW18] and ELSS [Sut+15]), and Adaptive Cluster Expansion (ACE) [Bar+16], which iteratively builds an improved approximation of the model using a cluster expansion. These approaches yield models that accurately reproduce empirical frequencies, making them appropriate when a faithfully generative model is wanted, though their performances in direct coupling prediction are only comparable to pseudo-likelihood approaches [MB19]. Unfortunately, these methods suffer from a considerable computational cost, making MSAs of length

$L \geq 200$ out of reach [Coc+18].

In this work, we focused on the pseudo-likelihood approach, since it provides statistically consistent models with state-of-the-art direct coupling prediction within a reasonable complexity. Among the three main pseudo-likelihood inference methods, we looked into CCMpred for its computational efficiency, and ultimately opted for its most recent Python version, CCMpredPy [VSS18], for the compelling additional features it provides, as will be explained in section 4.5.2.

4.3 From a target protein sequence to the input data of an inference method

This section covers preparatory steps leading to an input data set on which a Potts model is to be inferred. Since a single sequence does not contain covariation signal, representing a target protein with a Potts model requires an appropriate multiple sequence alignment to be built, implying further choices: how to retrieve additional sequences, how many should be included in the final set, and how insertions and deletions in this resulting MSA should be handled.

4.3.1 Key idea: model a target sequence and its close homologs

The success of profile-profile methods over sequence-profile methods in homology search, acknowledged in the previous chapter, suggests that the most sensible way to represent a target protein is to factor in its close homologs. In positional models, this provides substantial information on residue conservation and variability. In the case of Potts model, this addition is mandatory, since a covariation signal is needed to infer distant dependencies. Essentially, this can be seen as modeling our target sequence along with a small portion of the sequence space surrounding it.

In our work, homologs are retrieved using HHblits [Rem+12], more specifically using the workflow recommended for the Potts model inference CCMpred [SGS14] on their FAQ page [See], that is using HHblits with the following options:

```
-maxfilt 100000 -realign_max 100000 -all -B 100000 -Z 100000 -n 3 -e 0.001
```

on UniClust30 database [Mir+17], filtering the result at 80% identity using HHsuite's tool HHfilter. This yields a 80% non-redundant multiple sequence alignment of the sequence's close homologs which will be fed to a Potts model inference method after additional pre-processing steps detailed in the next sections.

It is worth noting that the reliance of Potts model inference on a pre-built multiple sequence alignment constitutes a first issue, since existing methods to retrieve homologs and build multiple sequence alignments such as HHblits only make use of positional conservation, thus distant dependencies are not taken into account when building the input MSA. Besides, since HHblits iteratively adds retrieved homologs to the target to retrieve more homologs, biases towards positional conservation are reinforced. To address this latter issue, we considered using BLAST to retrieve the sequences and align them using a multiple sequence alignment method such as MAFFT. Some preliminary tests on alignment quality with respect to reference structure alignments yielded significantly worst results with this workflow rather than the HHblits workflow, hence we provisionally put this idea aside.

4.3.2 The adequate number of effective homologs

HHblits can often retrieve a large number of putative homologs for a given target sequence. One should then find a balance between picking enough sequences so that the covariation signal is strong enough and at the same time make sure that sequences in the training set are not too distant from the target sequence – in other words, balance between specificity and sensitivity.

It has been empirically observed that using Potts models inferred on 1000 effective sequences – i.e. the number of sequences after having applied a 80% sequence identity reweighting (see section 2.4.2.3) – [Mor+11] achieves similar results in contact prediction to using Potts models inferred on a full alignment, and this number is broadly considered as a threshold below which contact prediction accuracy is significantly impacted [Ugu+17; AS15]. Following this empirical observation, we set the depth of our input multiple sequence alignments to 1000 effective sequences, disregarding all sequences retrieved after them. This allows us to have theoretically enough signal with a lesser risk of including unrelated

sequences in the set (see experiments section 6.1) and to reduce computation time of the inference method.

Naturally, this initial choice, practical and rather arbitrary, would need to be refined, typically with an E-value threshold, which would probably depend on the task. A complementary approach would be to reweight sequences in the MSA (cf section 2.4.2.3) so that closest homologs have more weight than further ones, decreasing weight as the risk of introducing noise increases.

4.3.3 Handling insertions and deletions

The main weakness of Potts models compared with profile Hidden Markov Models is probably their handling of insertions and deletions: while multiple sequence alignments typically contain stretches of gaps, Potts models only contain match states. Two models extending the Potts model with a gap handling strategy can be found in the literature: the most recent hidden Potts models introduced by Sean Eddy and described in the previous chapter, and an extended Potts models with gap parameters by Feinauer, Skwark et al. [Fei+14a]. They provide interesting perspectives, but the already challenging complexity of comparing two models with a quadratic number of parameters impelled us to focus on simpler models as a first step. Since the models contain only match states, it is particularly important to reflect on how gap symbols should be considered and what columns will be retained as match states. We discuss these questions in this section and describe how they are handled in our own workflow, which was mainly constrained by our choice of inferring the models with CCMpredPy.

4.3.3.1 The specific case of the gap symbol

Most inference methods treat the gap symbol as the 21^{st} amino acid, simply inferring a Potts model with an alphabet of $q = 21$ letters. In her PhD thesis [Vor17], Vorberg questioned this usage and suggested that gaps should be treated as missing information instead, arguing that treating gaps as a symbol leads to spurious artificial couplings. Indeed, as illustrated in figure 4.1, in the situation where the target protein to be modeled contains two domains and the multiple sequence alignment covers both domains separately, at two positions i and j in

different domains some double frequencies $f_{ij}(a,-)$ and $f_{ij}(-,b)$ between amino acid letters and gap letters will be high while double frequencies $f_{ij}(a,b)$ between amino acids will be null, despite relatively high respective single frequencies $f_i(a)$ and $f_j(b)$. This will lead to highly positive values of $w_{ij}(a,-)$ and $w_{ij}(-,b)$ and highly negative values of $w_{ij}(a,b)$, and overall a high $\|w_{ij}\|$ despite positions i and j not co-evolving in the target protein.

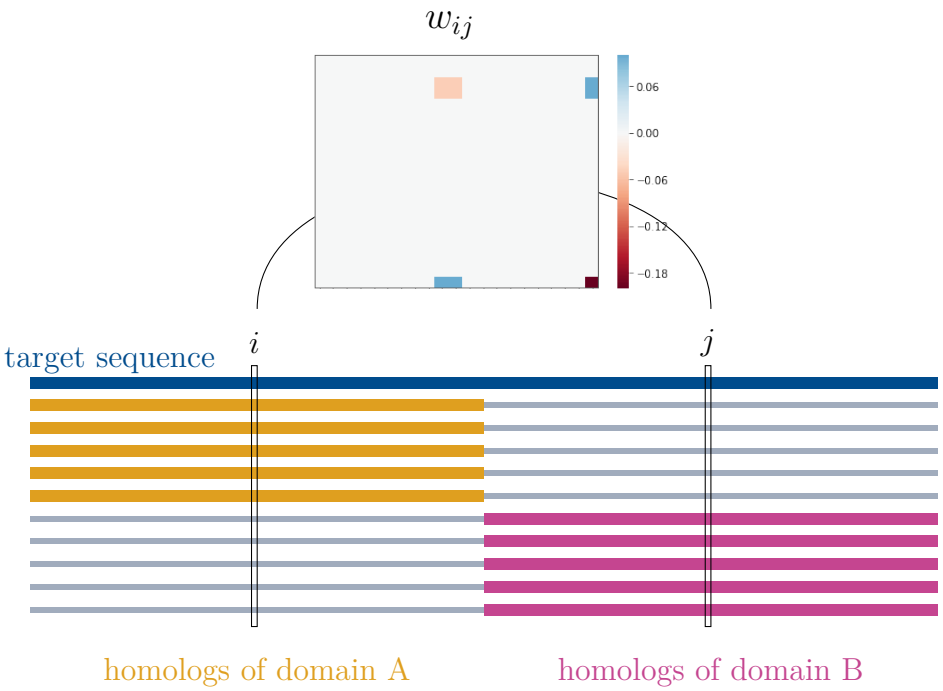


Figure 4.1 – Hypothetical MSA for a target sequence containing two distinct domains A and B and homologs covering only their corresponding domains, on two different parts of the MSA. Positions i in domain A and j in domain B do not co-evolve at all, yet if the gap symbol is considered as the 21^{st} letter their coupling matrix w_{ij} is significantly not null: for any a in column i and any b in column j, $w_{ij}(a,-)$ and $w_{ij}(-,b)$ are positive, while $w_{ij}(a,b)$ and $w_{ij}(-,-)$ are negative. (figure inspired from [Vor17])

Treating gaps as missing information overcomes this problem. However, it is clear that valuable information on the protein will also be lost in the process. In this thesis, since we will rely on Vorberg's inference method CCMpredPy, gaps are discarded and Potts models have $q = 20$ states. Whether gap information actually brings more noise than useful information is still an open question. The answer is

probably MSA-dependent, as a function of the protein's modularity and coverage of the retrieved homologs.

4.3.3.2 Handling gap-containing columns

Inferring a Potts model on all columns in the original multiple sequence alignment could lead to a rather noisy model if it contains many columns with many gaps, especially if gap symbols are treated as missing information.

A solution is to infer a Potts model on columns whose gap rate does not exceed a given threshold by trimming the input alignment, for instance using trimal [CSG09]. The lower this threshold, the more columns are removed.

To maintain consistency with positions in the original sequence, trimmed positions i can be re-inserted in the model with positional parameters at position i set to background fields defined using frequencies f_0 given by [Gil+01]

$$v_0(a) = \log f_0(a) - \frac{1}{q}\sum_{b=1}^{q} \log f_0(b) \tag{4.5}$$

and pairwise coupling parameters with position i set to:

$$\forall j, a, b, w_{ij}(a, b) = 0 \tag{4.6}$$

This becomes necessary in the context of pairwise Potts model alignment with a gap cost strategy.

4.4 Visualization of parameter choices effects

Determining objectively the hyperparameters to best represent proteins with Potts models in the most comparable way is not a trivial task. Ideally, all these hyperparameters would have to be jointly trained with respect to a comprehensive benchmark to optimize alignment quality and/or homology detection performances of our method. However, without prior analysis the search space would be overwhelming. In this thesis, we attempted to gain insight on the effect of these parameters to propose a first operational workflow for the construction of

comparable Potts models based on these intuitions, leaving their refinement to future work.

To gain intuition on each choice to be made, we developed visualization tools to see their effect on inferred parameters and on predicted couplings, available as part of PPsuite in our repository https://github.com/htalibart/ppsuite, and we examined the energy landscapes of the inferred Potts models for sequences at different homology levels. In this manuscript, these insights will be shown on a chosen toy protein: Atx1 metallochaperone of *Saccharomyces Cerevisiae* (PDB identifier 1CC8). This protein was selected for its small length (73 positions), the presence of strongly conserved positions (notably two cysteines at positions 15 and 18), a (metal) binding site and both alpha and beta secondary structures. A cartoon representation of 1CC8 is displayed figure 4.2.

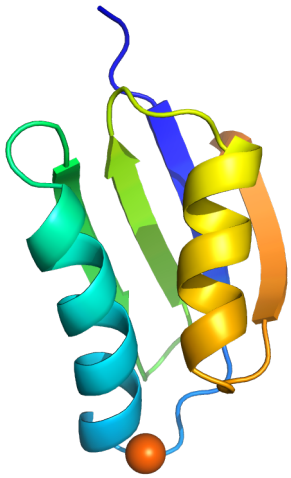


Figure 4.2 – Cartoon representation of 1CC8, rendered with PyMOL

For these purposes, following the workflow described in 4.3.1, we started by building a multiple sequence alignment of effective close homologs of 1CC8 using HHblits and filtering the output at 80% sequence identity. To avoid potential issues due to gaps, we substantially trimmed the resulting MSA with a maximal fraction of gaps allowed of 20%.

This initial set of homologs was used to build three separate sequence sets with different levels of similarity with 1CC8 in order to evaluate the ability of the

inferred Potts models to score sequences at different homology levels and visualize energy distributions:

- The first 1100 effective sequences were split into a S_{train} sequence (and its associated MSA M_{train} set which will be used to train Potts models and a $S_{left\text{-}out}$ set, where one sequence every 100 is assigned to the $S_{left\text{-}out}$ set. A sequence logo for the resulting S_{train} MSA is given figure 4.3.
- The following 1000 sequences constitute a sequence set denoted S_{close} which, as its name suggests, contains sequences close to sequences in the training set in terms of similarity, with the associated MSA M_{close}.

An additional random set termed S_{pdb30} was also built by randomly picking 1000 sequences out of 30% non-redundant PDB representatives.

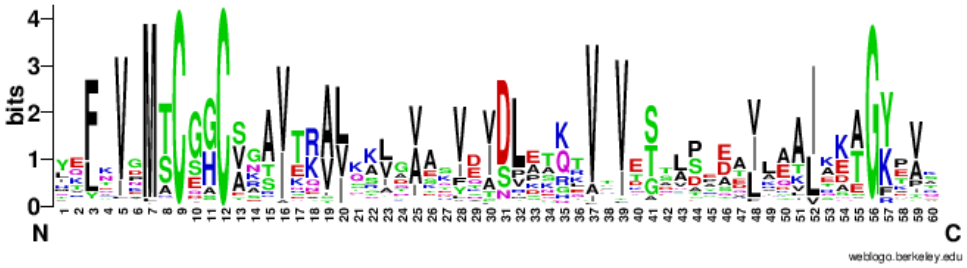


Figure 4.3 – Sequence logo built using WebLogo [Cro+04] from the *train* MSA M_{train}. As reflected by letters' heights, several positions are conserved in the MSA such as the two cysteines at positions 9 and 12 and a guanine at position 56.

Parameters of the Potts model inferred on the train MSA with CCMpredPy using default parameters is displayed figure 4.4.

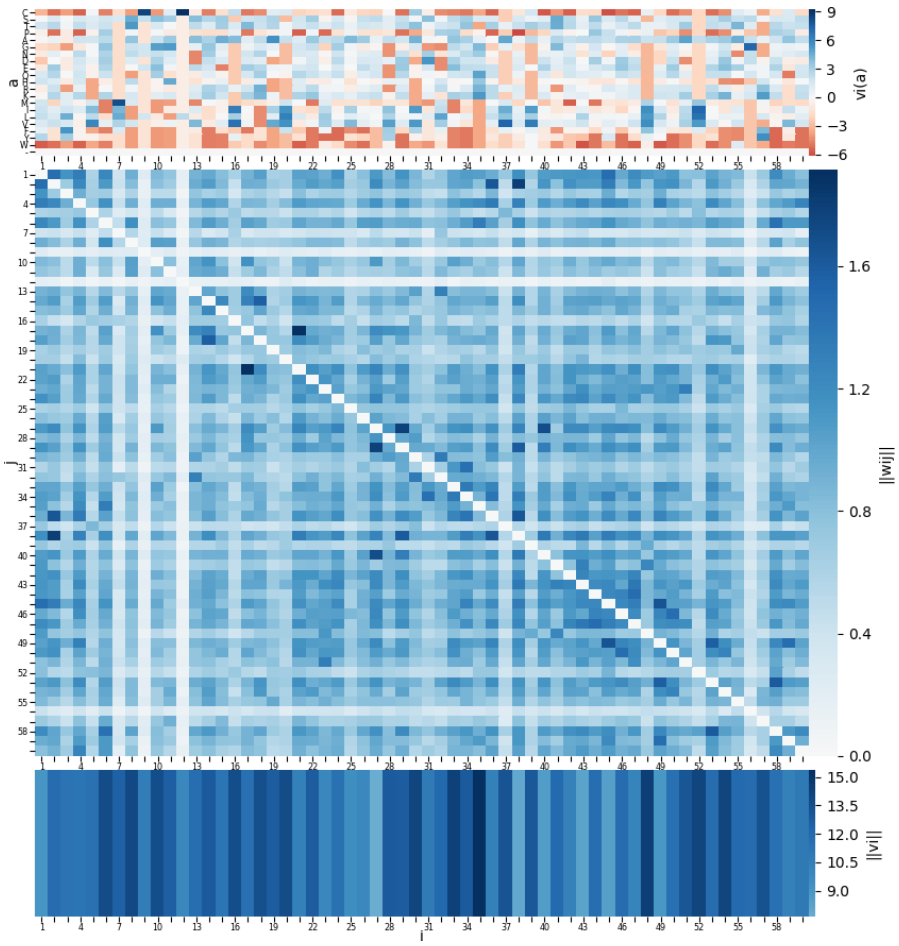


Figure 4.4 – Inferred parameters of a Potts model for 1CC8 on M_{train} . The first subplot shows the values of $v_i(a)$ for each amino acid a (following the order CSTPAGNDEQHRKMILVFYW) at each position i in the multiple sequence alignment. Blue indicates that $v_i(a)$ is positive (interpreted as "a is conserved at position i") and red indicates that $v_i(a)$ is negative (interpreted as "a should not be at position i" or "a is deficient at position i"). The two following subplots display parameter norms, which can be seen as their importance in the model: the second subplot is a heatmap of coupling norms (which can be interpreted as a predicted contact map) and the last subplot shows the value of $\|v_i\|$ at each position i. As we can see, conserved letters revealed in figure 4.3 stand out in v_i vectors, and lead to significantly low coupling norms, as expected.

To gain insight on the inferred Potts model's energy landscape, we computed the energy $\mathcal{H}(x) = -\left(\sum_i v_i(x_i) + \sum_{i,j} w_{ij}(x_i, x_j)\right)$ of each sequence x in each of

the four sets described above. Sequences of the three first sets were pre-aligned by HHblits while sequences in the S_{pdb30} set were aligned to the train MSA M_{train} using MUSCLE v3.8.31 [Edg04] with the `-profile` option. Distributions for the Potts model inferred with default options are plotted figure 4.5. Note that, since the probability of a sequence x is defined by $P(x) = \frac{1}{Z}\exp(-\mathcal{H}(x))$, such graphs thoroughly reflect the distributions of the sequence probabilities.

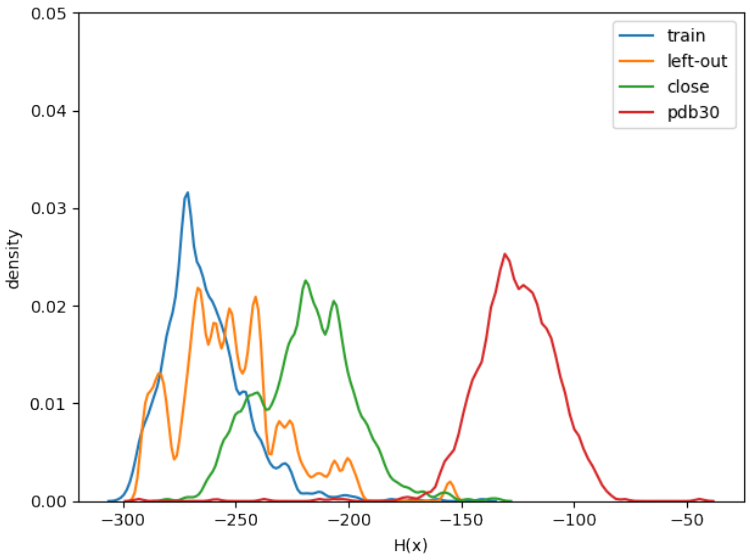


Figure 4.5 – Distribution of the energies $\mathcal{H}(x)$ for each sequence x in the different data sets introduced earlier with respect to a Potts model inferred on M_{train} with CCMpredPy using default options. As expected, since the Potts model was inferred on the S_{train} set, sequences in this set have the lowest energies. Sequences in the $S_{left\text{-}out}$ set seem to have slightly higher energies, suggesting that the Potts model might be slightly overfitting the data. Energies of sequences in the S_{close} set, that are close to sequences in the S_{train} set in terms of similarity, are higher but stay close to energies of sequences in the train set, while the difference with the random set S_{pdb30} is rather clear.

In addition to inferred parameter visualization and energy landscapes, top couplings – position pairs whose corresponding coupling matrix has the highest Frobenius norm – can be visualized using our visualization tool VizPymol, released as part of our PPsuite software suite, which generates a PyMOL session to visualize them on the protein's 3D structure and examine whether they correspond to

positions in contact. The top 20 couplings for 1CC8 using default options on M_{train} are displayed figure 4.6.

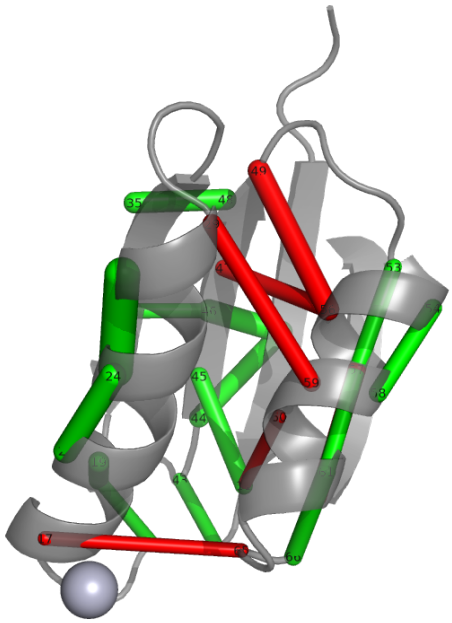


Figure 4.6 – Top 20 position pairs (i, j) with the highest $\|w_{ij}\|$ and $|i-j| > 3$ for the Potts model inferred on the train MSA M_{train} by CCMpredPy with default options projected on 1CC8 PDB structure. Couplings whose endpoints are separated by less than 8 Angstrom in the 3D structure are displayed in green while others are displayed in red.

4.5 Prior choices on the model towards canonicity

4.5.1 Gauge choice for more interpretable parameters

As explained in section 3.3.3, the likelihood function has a unique global maximum up to a gauge choice. Two traditional gauge choices are the *zero-sum gauge* (used by [Wei+09]), fixing:

$$\forall i, \sum_a v_i(a) = 0 \tag{4.7}$$

$$\forall i, j, a, \sum_b w_{ij}(a, b) = \sum_b w_{ij}(b, a) = 0 \tag{4.8}$$

and the *lattice-gas gauge* (used by [Mor+11]):

$$\forall i, v_i(c) = 0 \tag{4.9}$$

$$\forall i, j, a, w_{ij}(a, c) = w_{ij}(c, a) = 0 \tag{4.10}$$

where c is an arbitrary letter acting as a reference state: all potentials are measured with respect to state c.

Inferred parameters can easily be translated from one gauge to another. A gauge transformation adds a constant to all energies, and yields the same probabilities since this shift is compensated in the normalization constant Z. However, this choice has an impact on the interpretation of each parameter. An illustration of the effect of gauge choice on positional parameters and their norms is given figure 4.7.

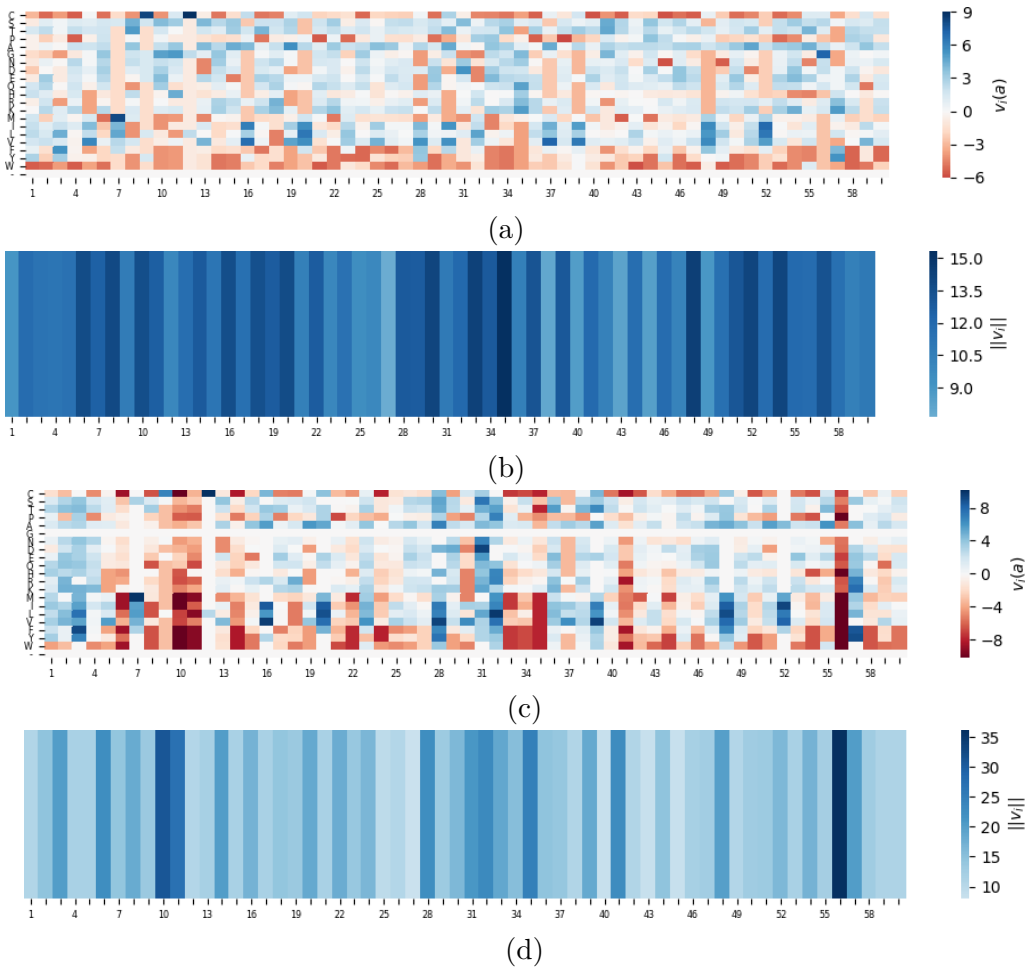


Figure 4.7 – Inferred $v_i(a)$ and $\|v_i\|$ for the same Potts model inferred for 1CC8 with a zero-sum gauge (subfigures 4.7a and 4.7b) and a lattice-gas gauge arbitrarily centered at G (subfigures 4.7c and 4.7d). Higher fields norms in the lattice-gas gauge model correspond to positions where G is highly conserved.

When it comes to interpretability, a zero-sum gauge seems preferable, since there is no apparent reason to consider a letter as reference more than another. Furthermore, a zero-sum gauge implies that parameters are centered: the mean for each column v_i (or each matrix w_{ij}) is 0, hence each value $v_i(a)$ (or $w_{ij}(a, b)$) can directly be interpreted as a bias relative to the column (or matrix) mean. This also means that $\|v_i\|$ is the standard deviation of v_i times constant $\sqrt{q}$, which is consistent with our intention of attributing higher norms to positions with more information. In the same way, $\|w_{ij}\|$ is the standard deviation of w_{ij} times q, thus

a high norm informs us on existence and intensity of pairwise relations between residues.

As we will see in the next section, gauge can also be determined by a regularization choice.

4.5.2 Guiding inference towards more canonical parameters

Because of the intractable computational complexity of exact inference, in practice, inferred Potts models will be different from the "true" Potts models maximizing entropy. In fact, down to some accuracy threshold in reproducing input data statistics, Potts models with highly divergent parameters can define highly similar probability distributions [Bar18]. With this in mind, it seems appropriate to guide inference towards the approximation that is best suited to our needs by constraining parameter inference towards a desired prior, with the use of regularization, and starting inference at an appropriate initialization point, knowing that parameters will tend to stay close to it. Here, we argue that this prior and initial configuration should be an independent-site Potts model based on theoretical arguments from [Vor17] and early experimentation on artificial and real data, and discuss the influence of underlying hyperparameters on the inferred Potts model.

4.5.2.1 Choice of prior at an independent-site Potts model

This section motivates our choice of setting a prior at a Potts model without couplings (*independent-site model*). The idea of choosing this prior rather than a null model was introduced by [Vor17] for CCMpredPy. Though initialization at this configuration was implemented as well, their initial motivation was to rectify biases introduced by traditional regularization centered at 0.

Regularization is a widely used tool to prevent a model from overfitting its training data. The idea is to reduce model complexity by introducing a regularization term to the objective function in order to constrain inference of its parameters, usually to favor sparser parameters and small values.

In case of Potts models, this means that the objective function to be maximized during inference becomes:

$$\mathcal{L}(v, w|X) + R(v, w) \tag{4.11}$$

where $\mathcal{L}(v, w|X)$ is the likelihood of parameters v and w given MSA X and $R(v, w)$ is a regularization term constraining v and w inference. The most commonly used regularization term in Potts model inference is zero-centered L_2 regularization:

$$R(v, w) = -\lambda_v \|v\|_2 - \lambda_w \|w\|_2 \tag{4.12}$$

which, as we can see by identifying mean $\boldsymbol{\mu} = 0$ and covariance matrix $\boldsymbol{\Sigma} = \frac{1}{2\lambda} I$ in the multivariate normal distribution density function of a N random vector:

$$f(x) = \frac{1}{(2\pi)^{\frac{N}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(x - \boldsymbol{\mu})\right) \tag{4.13}$$

amounts to adding a zero-centered Gaussian prior on the parameters.

Since at optimum $\frac{\partial \mathcal{L}(v,w|X)}{\partial v} = \frac{\partial R(v,w)}{\partial v}$ and $\frac{\partial \mathcal{L}(v,w|X)}{\partial w} = \frac{\partial R(v,w)}{\partial w}$, this regularization term implies the following gauge (a detailed proof can be found in [EHA14]):

$$\sum_a v_i(a) = 0 \tag{4.14}$$

$$\sum_b w_{ij}(a, b) = \frac{\lambda_v}{\lambda_w} v_i(a) \tag{4.15}$$

While equation 4.14 is equivalent to zero-sum gauge on v, equation 4.15 introduces a bias which binds fields and couplings. Indeed, for instance in the case where a is highly conserved at position i, since $v_i(a)$ is large, (4.15) forces $\sum_b w_{ij}(a, b)$ to be large as well despite the absence of direct coupling.

This undesired behavior was fixed in CCMpredPy by centering the Gaussian prior on the positional parameters and initializing them at a different mean v^* which corresponds to the parameters of an independent-site model:

$$f_i(a) = \frac{\exp(v_i^*(a))}{\sum_b \exp(v_i^*(b))} \tag{4.16}$$

which yields

$$v_i^*(a) = \log f_i(a) - \frac{1}{q}\sum_b f_i(b) \tag{4.17}$$

if we fix the remaining indeterminacy with a zero-sum gauge $\forall i, \sum_a v_i^*(a) = 0$.

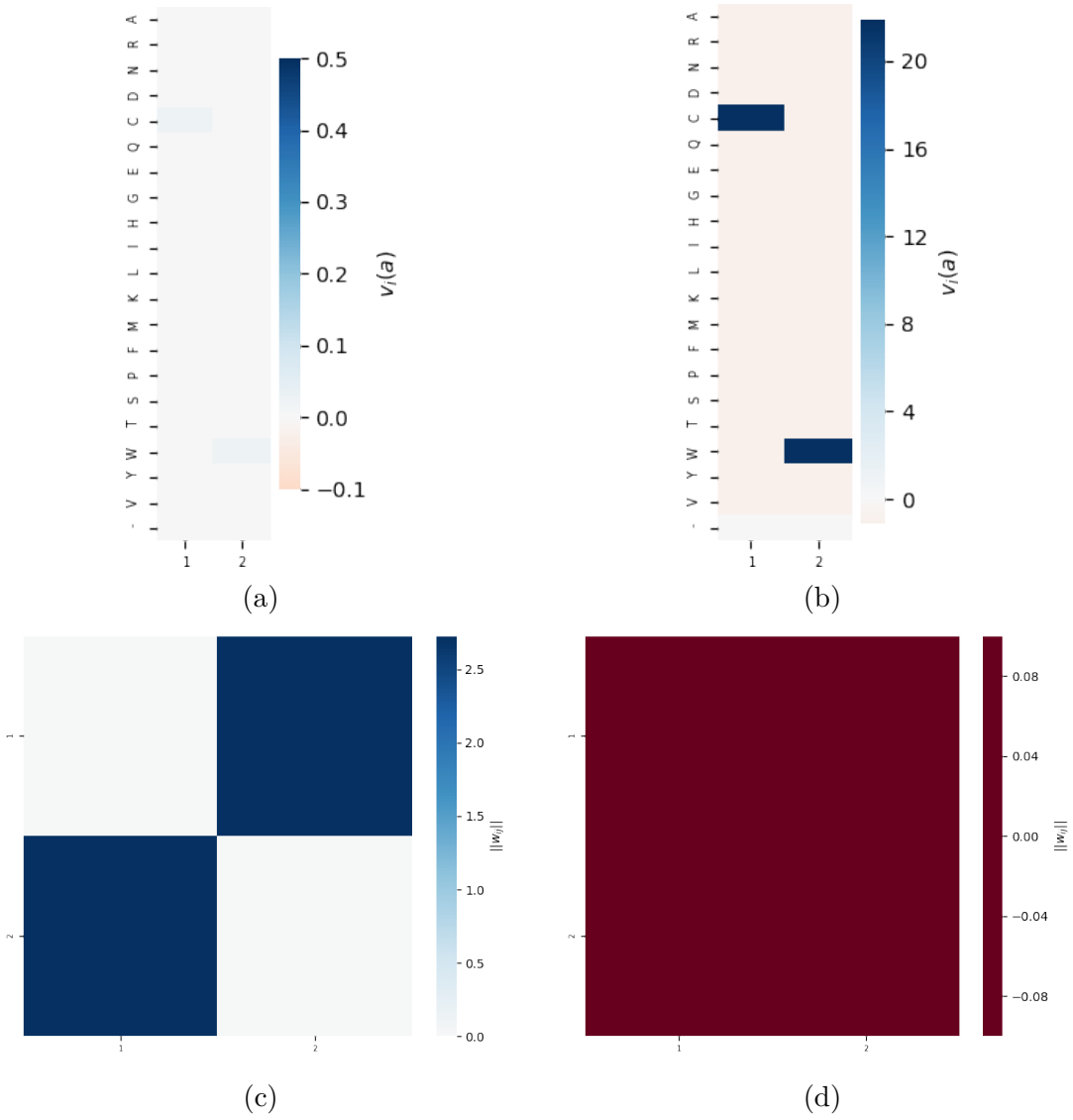
The L_2 regularization term becomes:

$$R(v, w) = -\lambda_v \left\| v - v^* \right\|_2 - \lambda_w \left\| w \right\|_2 \tag{4.18}$$

At optimum, equation 4.15 becomes:

$$v_i(a) - v_i^*(a) = \frac{\lambda_w}{\lambda_v}\sum_b w_{ij}(a, b) \tag{4.19}$$

which means that $\sum_b w_{ij}(a, b)$ will deviate from 0 only when v_i deviates from the independent-site vector, which complies with the intention of explaining data with residue conservation as much as possible and only add couplings that are necessary. An illustration of the effect of centering and initializing positional parameters at v^* on the inferred w is given figures 4.8 on artificial data and 4.9 for our toy protein 1CC8.



(a) (b)

(c) (d)

Figure 4.8 – Parameters of Potts models inferred with two different priors on the positional parameters from an artificial conserved MSA made of two columns, the first one with only cysteines and the second one with only tryptophans. Figures 4.8a and 4.8c show the $v_i(a)$ and $\|w_{ij}\|$ of a Potts model inferred with a prior centered at 0 while figures 4.8b and 4.8d correspond to a Potts model inferred with a prior centered at the independent-site positional parameters v^*. Both were inferred without pseudo-counts. As we can see, the Potts model initialized at the independent-site model did not add any unwanted coupling between the two conserved positions while the Potts model initialized at 0 explained the MSA using both fields and couplings parameters.

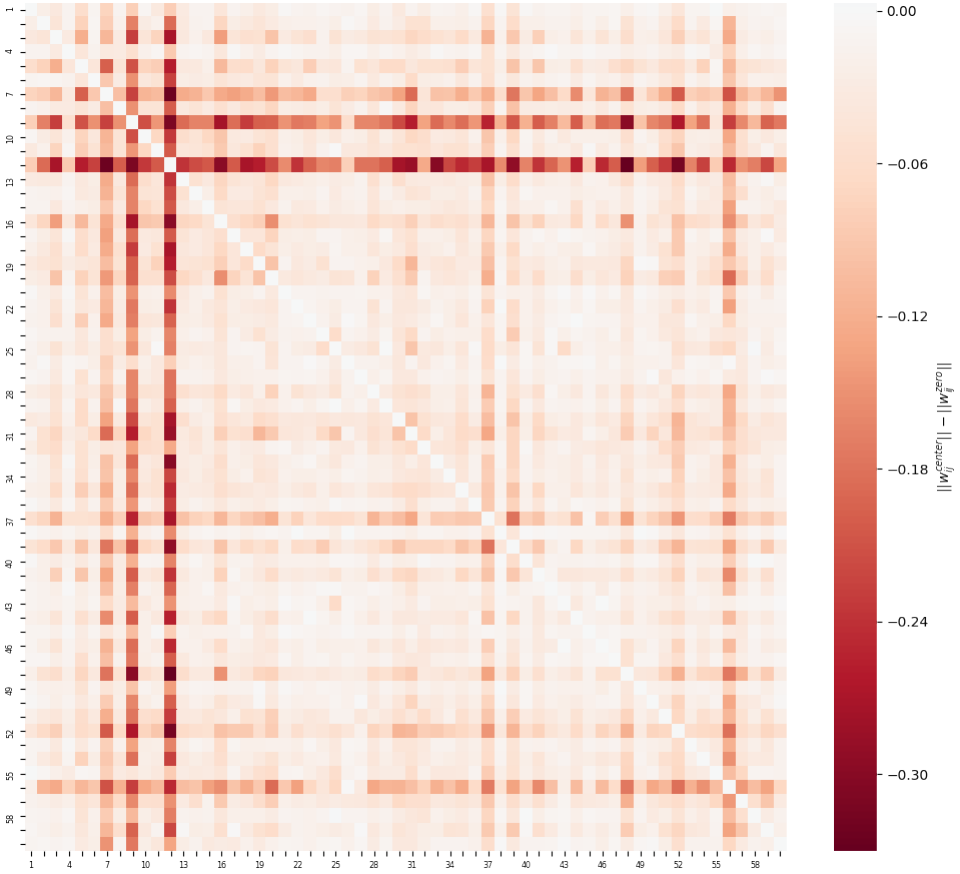


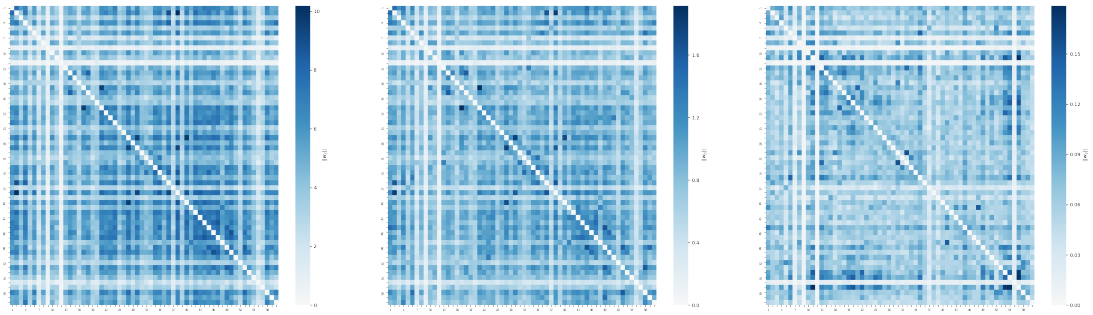
Figure 4.9 – Heatmap of coupling norm differences between a model inferred with v centered at v^* and at 0 for 1CC8. The biggest differences correspond to most conserved columns in the MSA, notably the two conserved cysteines at positions 9 and 12. This shows that a Potts model inferred with a zero-centered prior has a greater tendency to explain some positional conservation signal with pairwise coupling parameters.

4.5.2.2 Influence of regularization coefficients

Regularization strength can be tuned with coefficients λ_w and λ_v : the higher λ_w, the more the couplings are pushed towards 0, and the higher λ_v, the more the fields are pushed towards v^*.

Influence of pairwise regularization coefficient λ_w. Since the number of coupling matrices increases quadratically with the length of the input MSA L, in CCMpred λ_w is set as a linear function of $L-1$, by default $\lambda_w = 0.2 \times (L-1)$.

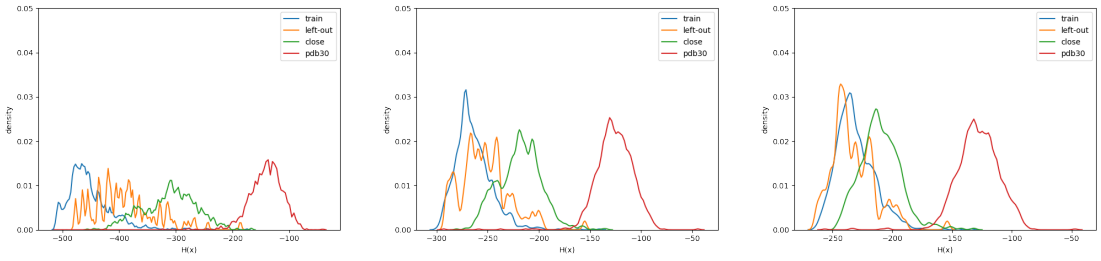
Inferred couplings for 1CC8 for different λ_w are shown figure 4.10.



(a) $\lambda_w = 0.001 \times (L-1)$ (b) $\lambda_w = 0.2 \times (L-1)$ (c) $\lambda_w = 10 \times (L-1)$

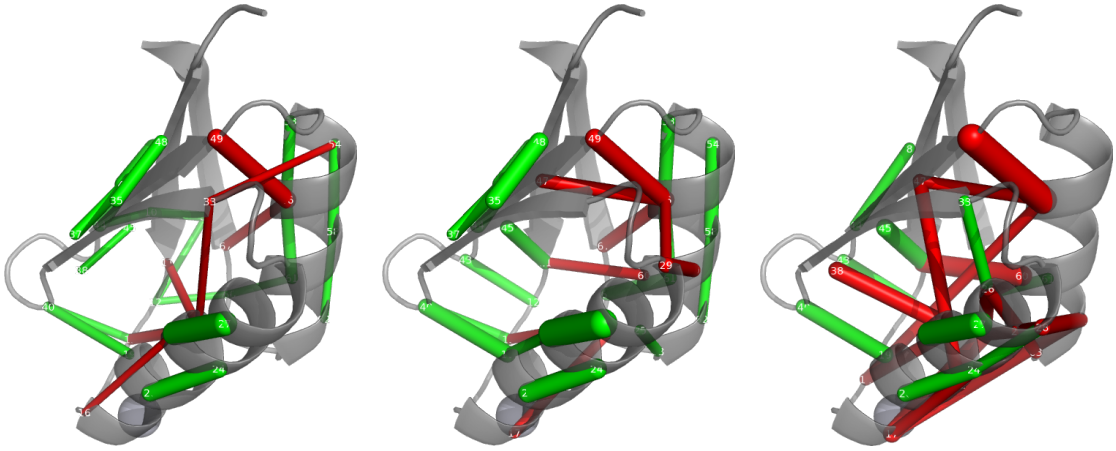
Figure 4.10 – $\|w_{ij}\|$ of a Potts model inferred for 1CC8 with different values of λ_w. A lower λ_w favors higher values of $\|w_{ij}\|$ while a higher λ_w tends to favor sparsity.

Observing the energy landscape (figure 4.11), one can see that λ_w plays a role in preventing the model from overfitting the training set.



(a) $\lambda_w = 0.001 \times (L-1)$ (b) $\lambda_w = 0.2 \times (L-1)$ (c) $\lambda_w = 10 \times (L-1)$

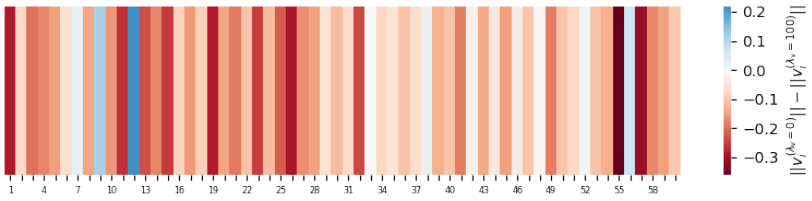
Figure 4.11 – Energies of sequences in the different sets presented in section 4.4 for Potts model inferred with different values of λ_w. The lower λ_w, the more energies of sequences in the $S_{left\text{-}out}$ and S_{close} sets are different from energies of sequences in the S_{train} set. In other words, the lower λ_w, the more the model fits the input data.



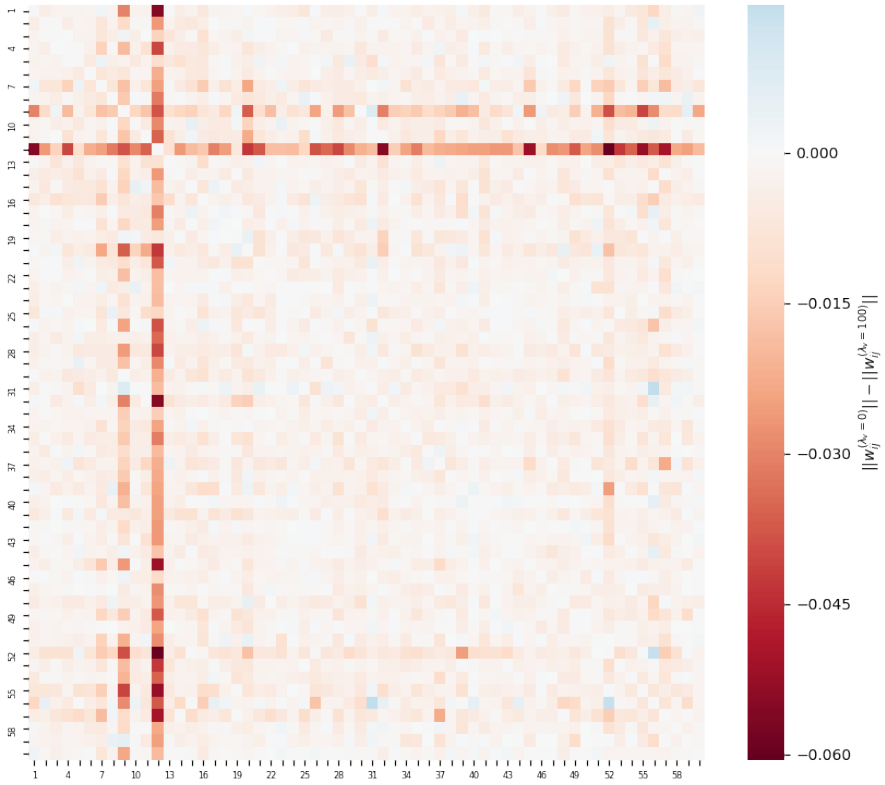
(a) $\lambda_w = 0.001 \times (L-1)$ (b) $\lambda_w = 0.2 \times (L-1)$ (c) $\lambda_w = 10 \times (L-1)$

Figure 4.12 – Top 25 position pairs (i, j) with the highest $\|w_{ij}\|$ and $|i-j| > 3$ projected on 1CC8 PDB structure for different values of λ_w. Green indicates that positions are in contact in the structure while red indicates otherwise. As we can see, λ_w changes the value but also the rank of the coupling norms.

Influence of single regularization coefficient λ_v. While it is clear that initializing v to v^* avoids spurious couplings at conserved positions (see figures 4.8 and 4.9), pushing v too much towards v^* seems to have the opposite effect (figure 4.13 illustrates this behavior). This might be explained by the fact that, in practice, even at conserved positions, $f_{ij}(a, b)$ is different from $f_i(a)f_j(b)$ and constraining v_i and v_j towards independent-site parameter values compels $w_{ij}(a, b)$ to deviate from 0 to compensate for this difference.



(a)



(b)

Figure 4.13 – Difference in $\|v_i\|$ (subfigure 4.13a) and $\|w_{ij}\|$ (subfigure 4.13b) between two Potts models inferred with $\lambda_v = 0$ and $\lambda_v = 100$. $\lambda_v = 0$ results in v_i with higher norms at more conserved positions such as the cysteines at positions 9 and 11 while $\lambda_v = 100$ tends to put more weight on the coupling matrices at these positions.

4.6 Gearing parameters towards more comparable models

While the previous section focused mainly on how weights are distributed between fields and couplings of a given model, this section examines field vectors and coupling matrices independently, with a view to making each considered parameter comparable with another model's parameters. As we will see, we found that the difficulty of comparing two fields or two couplings is closely related to the problem of lessening the effect of small sampling variations as introduced in section 2.4.2.2.

4.6.1 Towards more comparable field parameters

4.6.1.1 How small sampling variations affect field parameters in the presence of low probabilites

By nature, Potts model's parameters are linked to empirical frequencies through a logarithmic transformation. Parameters v^* of an independent-site Potts model, as stated in 4.5.2, are defined by:

$$v_i^*(a) = \log f_i(a) - \frac{1}{q}\sum_b \log f_i(b) \tag{4.20}$$

Parameters $v_i(a)$ of a pairwise Potts model have a less straightforward relation to their associated single frequencies $f_i(a)$ due to the additional contribution of all couplings but are still subject to this logarithmic transformation, and since they are expected to stay close to the independent-site parameters $v_i^*(a)$ during inference due to the chosen L_2 regularization, one can reasonably transfer understanding on the behavior of v^* to v without loss of generality.

By nature of log itself, applying logarithm to small probabilities substantially spreads their values, while higher probabilities are less affected (see figure 4.14).

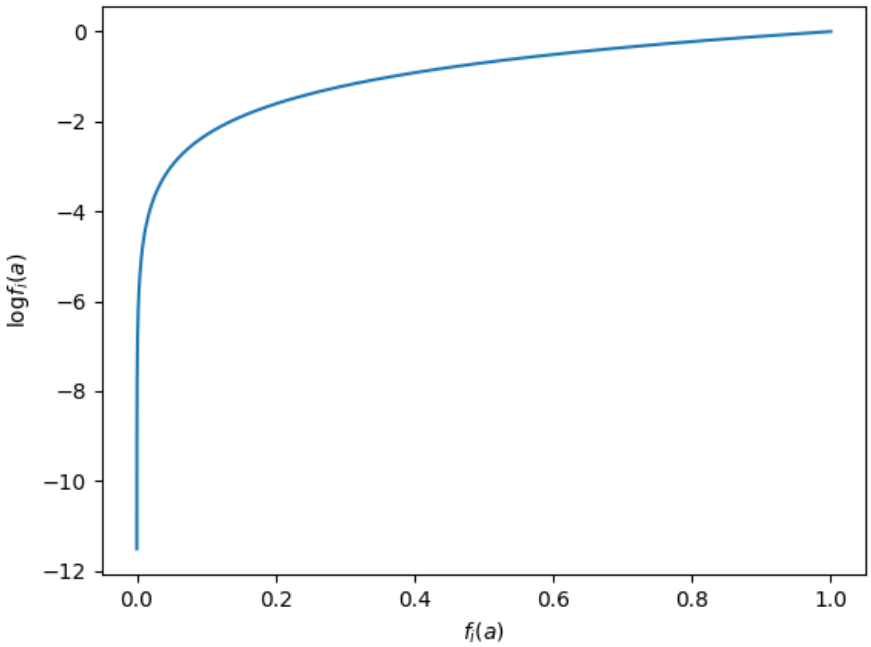
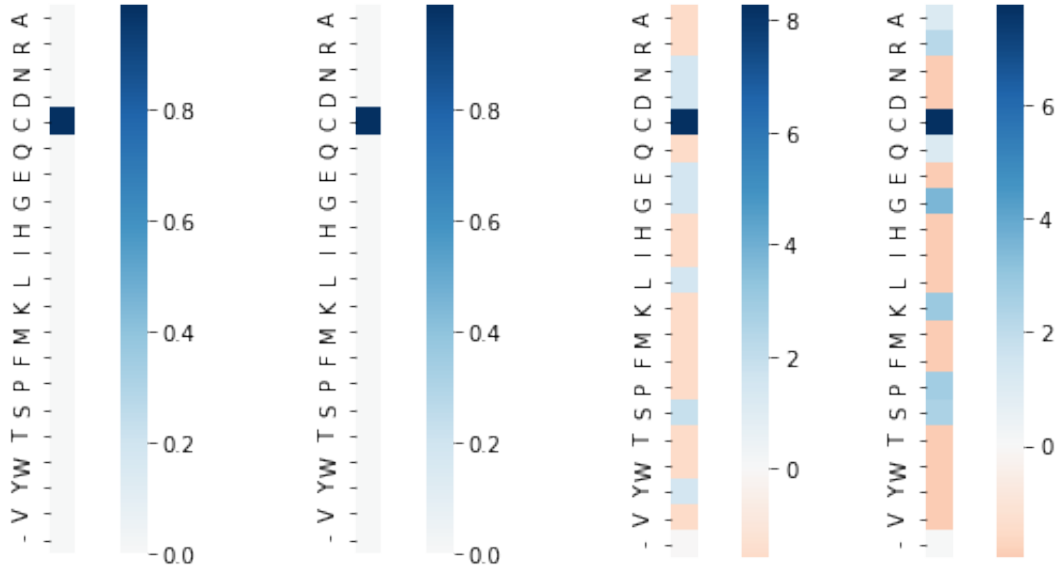


Figure 4.14 – $\log f_i(a)$ versus $f_i(a)$ for $f_i(a)$ between 10^{-5} and 1.

In the case of sufficient and noiseless data, this behavior is desirable: it allows to distinguish between amino acids that must not be at position i (very low probabilities) and amino acids that can (probabilities from moderately low to high). However, in practice, this makes fields parameters highly sensitive to small sampling variations and outliers.

Let us illustrate this problem by looking at a single position $i = 9$ in our toy protein 1CC8: the location of the first highly conserved cystein. Looking at our M_{train} MSA, i.e. basically the first 1000 effective sequences, only 8 of them do not have a C at position i – it is possible that these sequences retrieved by HHblits are actually outliers and are not actually homologous to 1CC8 – hence the probability mass is almost exclusively on letter C (figures 4.15a and 4.15b) but computing v_i^* with equation 4.20 yields substantially dissimilar values for parameters of unconserved letters (figure 4.15c). Furthermore, computing $v_i'^*$ on the same column using the next 1000 sequences – same column i but in our M_{close}

MSA – yields a rather different vector (figure 4.15d) while only 32 sequences out of the 1000 do not have a C at position i. These few letters significantly affect the resulting field vector.



(a) (b) (c) (d)

Figure 4.15 – Probability distribution in amino acids in column $i = 9$ of the first (figure 4.15a) and second (figure 4.15b) parts of our toy MSA for 1CC8 with the respective v_i^* computed for each part (resp. figures 4.15c and 4.15d). While the probability distributions look identical in the two parts with a highly conserved C and other probabilities seemingly equally close to 0, the field parameters are quite different: the conserved C is still preponderant but values $v_i^*(a)$ for other amino acids differ, some of them switch sign between the two parts.

As illustrated by this example, small changes in small frequencies have a dramatic effect on the model's parameters, making it harder to compare them. In particular, this has a dramatic effect on our similarity function which will be introduced in the next chapter, since it is based on the scalar product. While the scalar product has valuable properties, it is highly dependent on the signs of the parameters. Defining a single position normalized similarity score between two

columns as:

$$s_{normalized}(v_i, v_k) = \frac{2\langle v_i, v_k\rangle}{\langle v_i, v_i\rangle + \langle v_k, v_k\rangle} \tag{4.21}$$

it appears that the normalized similarity score between the two v_i that are supposed to represent the same position in 1CC8 is only 0.495, while we would like it to be close to 1.

4.6.1.2 Initializing the prior on positional parameters with additional pseudo-counts

As explained in section 2.4.2.2, a common way to smooth these small sampling variations is to add pseudo-counts to empirical frequencies, that should prevail when signal is insufficient. However, in pseudo-likelihood, models are not inferred on the frequencies but on the full sequences, making it impossible to add pseudo-counts directly. Nonetheless, in CCMpredPy additional information can be taken into account when initializing and centering the single parameters v at those of an independent-site model v^* since these are computed using empirical single frequencies, making it possible to introduce pseudo-counts via the prior:

$$f_i(a) = (1 - \tau_v) f_{0i}(a) + \tau_v \tilde{f}_i(a) \tag{4.22}$$

where $f_{0i}(a)$ is the initial observed frequency of a at position i in the MSA, $\tilde{f}_i(a)$ is the pseudo-count frequency, and τ_v is the single pseudo-count rate.

Two pseudo-count schemes are implemented: uniform pseudo-counts and pseudo-counts based on the BLOSUM62 substitution matrix.

Smoothing parameters without adding prior information using uniform pseudo-counts. Uniform pseudo-counts add the same amount to each residue count regardless of the column distribution:

$$f_i(a) = (1 - \tau_v) f_{0i}(a) + \frac{\tau_v}{q} \tag{4.23}$$

In other words, as a function of the pseudo-count rate τ_v, probabilities become closer to a plain uniform distribution, and corresponding parameters become closer

to 0. But, due to the nature of log, low probabilities (below uniform) are much more affected by this redistribution. Figure 4.16 shows the logarithm of $f_i(a)$ with respect to $f_{0i}(a)$ for different values of τ_v.

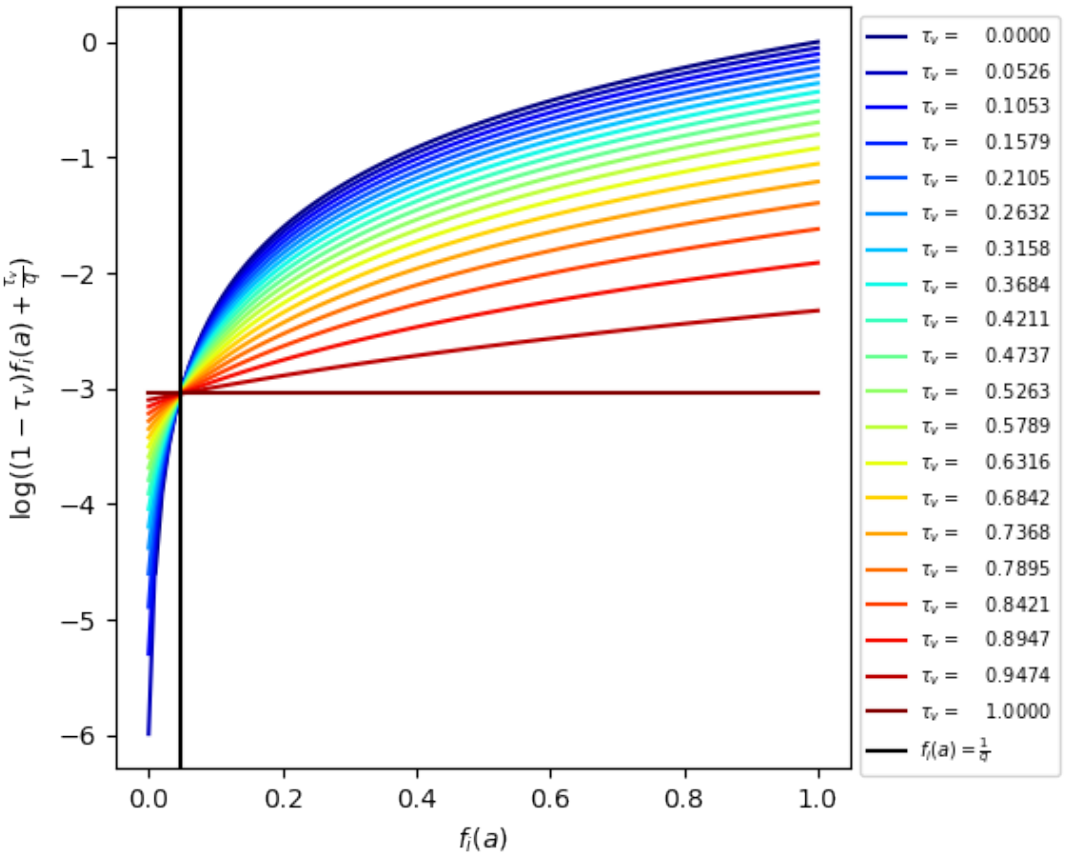
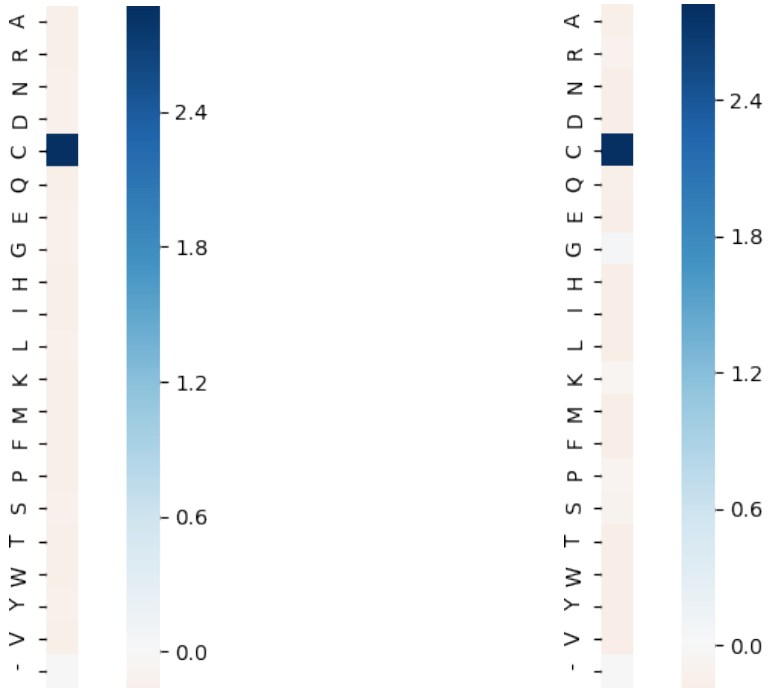


Figure 4.16 – Influence of uniform pseudo-count rate τ_v on the logarithm of the rescaled frequencies. The black vertical line indicates the normal distribution $f_i(a) = \frac{1}{q}$. The darkest blue line ($\tau_v = 0$) is the simple logarithm function, without any pseudo-counts. As we have seen in the previous section, small probabilities are widely spread. The darkest red line ($\tau_v = 1$) corresponds to a completely uniform distribution: all $f_i(a)$ are assigned the same value. Between the two extremes, logarithms of probabilities are plotted for different pseudo-count rates τ_v. As we can see, probabilities below the uniform probability threshold are substantially pulled towards uniform distribution (more or less depending on τ_v) while higher probabilities are also pushed down to the uniform distribution, but on a lesser scale.

Consequently, adding uniform pseudo-counts on the single frequencies can be used to smooth parameters and avoid numerical problems due to an insufficient number of observations. These small probabilities are pushed towards $\frac{1}{q}$ while bigger probabilities stand out.

Effects of uniform pseudo-counts are shown figure 4.17 for our conserved column from 1CC8 and figure 4.18 for all positions in 1CC8. As shown in the figures, small variations were smoothed while the conserved cystein still stands out. The two columns can be properly compared, and their normalized similarity score as defined in equation (4.21) changes from 0.495 to 0.995, i.e. close to 1, as it should be.



(a) (b)

Figure 4.17 – v_i^* parameters for the first (figure 4.17a) and second (figure 4.17b) part of the same column i in our MSA for 1CC8 after applying uniform pseudo-counts with $\tau_v = 0.5$. The two columns look rather similar: thanks to pseudo-counts, small variations were smoothed, while the conserved C still stands out.

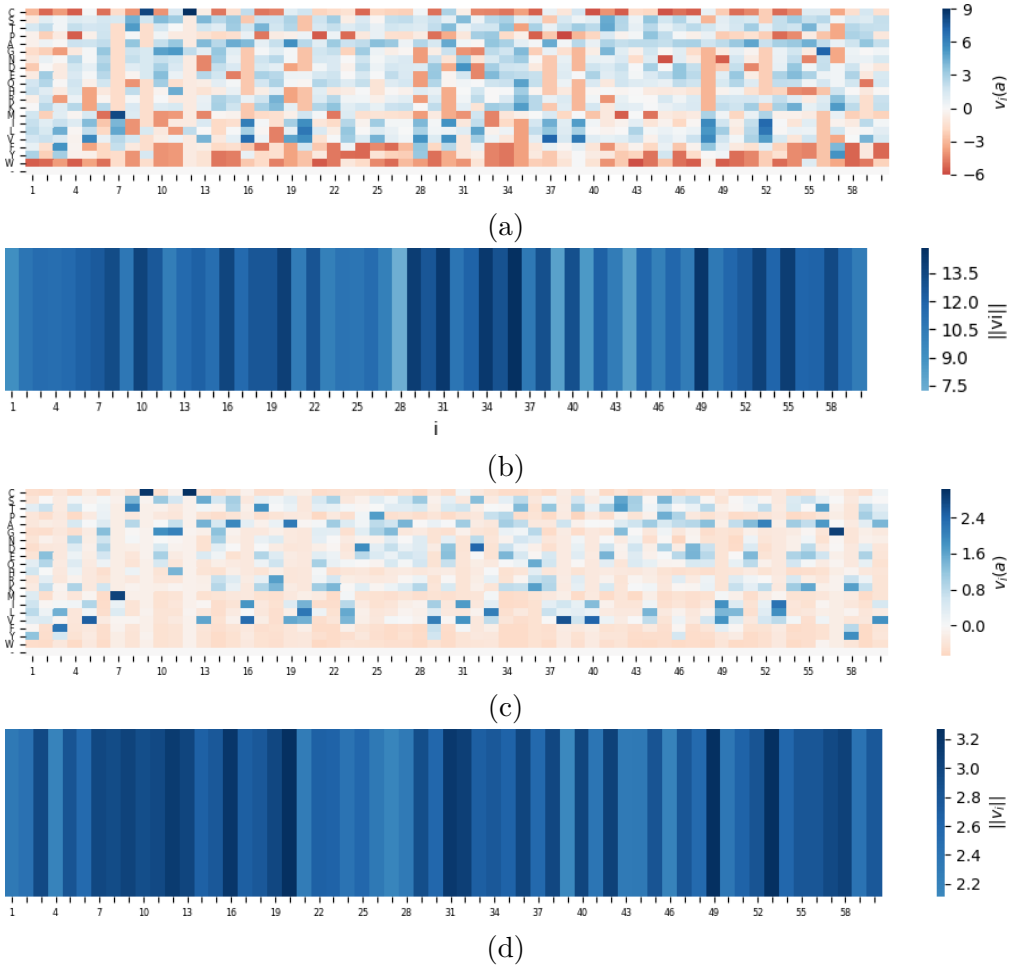


Figure 4.18 – v_i parameters and corresponding norms before (figures 4.18a and 4.18b) and after (figures 4.18c and 4.18d) applying uniform pseudo-counts with $\tau_v = 0.5$. Uniform pseudo-counts made field parameters shrink towards 0, especially negative ones, making particularly conserved amino acids stand out.

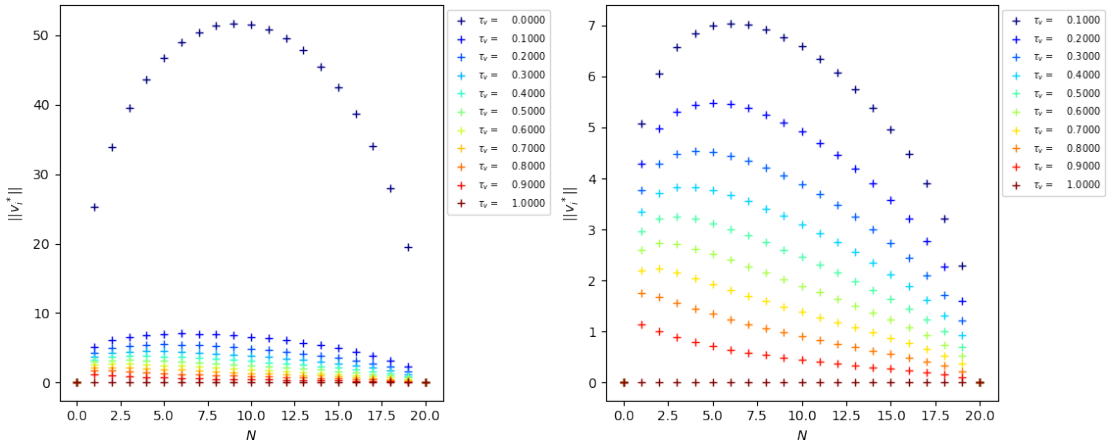
Note that, by definition, this pseudo-count scheme does not introduce a bias for the distribution in a given column to deviate towards any particular letters. Provided that $\tau_v < 1$, vector v_i^* is still null if and only if the distribution of the letters in the original column is uniform.

Interestingly, the choice of a uniform pseudo-count rate τ_v also has consequences on the norms of the positional parameters of the resulting Potts model. Let us illustrate this on artificial columns f_{0i} where weight is equally distributed between

a given number N of letters:

$$f_{0i}(a) = \begin{cases} \frac{1-\epsilon}{N} & \text{if } a \leq N \\ \frac{\epsilon}{q-N} & \text{otherwise} \end{cases} \tag{4.24}$$

where ϵ is set to 10^{-10} since $\log(0)$ is $-\infty$. Norms of field vectors v_i^* computed as in (4.20) based on f_{0i} for different numbers of conserved letters N and different pseudo-count rates τ_v are displayed figure 4.19.



(a) (b)

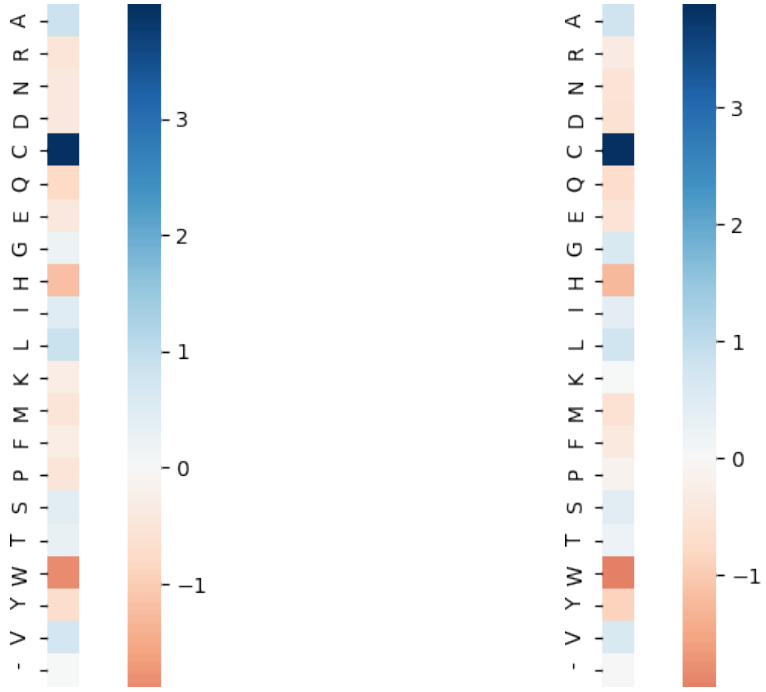
Figure 4.19 – Norms of v_i^* for different numbers of conserved letters N and different pseudo-count rates τ_v, from 0 to 1 in figure 4.19a and from 0.1 to 1 in figure 4.19b. As we can see, when no pseudo-count are added, the norm of the field vector is maximal when weight is equally distributed between half the letters, and adding uniform pseudo-counts affects this distribution: as τ_v increases, norms of field vectors of positions with fewer conserved letters increases.

Hence, τ_v can be chosen so as to put more weight on positions with the desired conservation pattern. A high τ_v will give higher norms as the number of conserved letters decreases in a virtually linear fashion, while some intermediate τ_v can assign higher norms to positions with 1 to 3 conserved letters for instance, making it possible to put nearly as much weight on a position with a conserved cystein as a position with a conserved property involving more letters such as hydrophobicity.

Introducing prior information using substitution matrix pseudocounts. Another pseudo-count scheme implemented in CCMpredPy is the popular one based on a substitution matrix:

$$f_i(a) = (1 - \tau_v) f_{0i}(a) + \tau_v \sum_b p(a|b) f_{0i}(b) \tag{4.25}$$

where $p(a|b)$ is the probability of mutation from b to a extracted from the BLOSUM62 substitution matrix. Unlike in the uniform pseudo-counts scheme where the influence of τ_v can directly be assessed, each $f_i(a)$ depends on the whole column i, thus it will have a different effect depending on the composition of the column. Effects of substitution matrix pseudo-counts on our conserved column from 1CC8 are shown figure 4.20 for our conserved column from 1CC8 and figure 4.21 for all positions in 1CC8.



(a) (b)

Figure 4.20 – v_i^* parameters for the first (figure 4.20a) and second (figure 4.20b) part of the same column i in our MSA for 1CC8 after applying BLOSUM62 substitution matrix pseudo-counts with $\tau_v = 0.5$. The two columns look similar, the C still stands out while the field values of other amino acids reflect the background probability distribution.

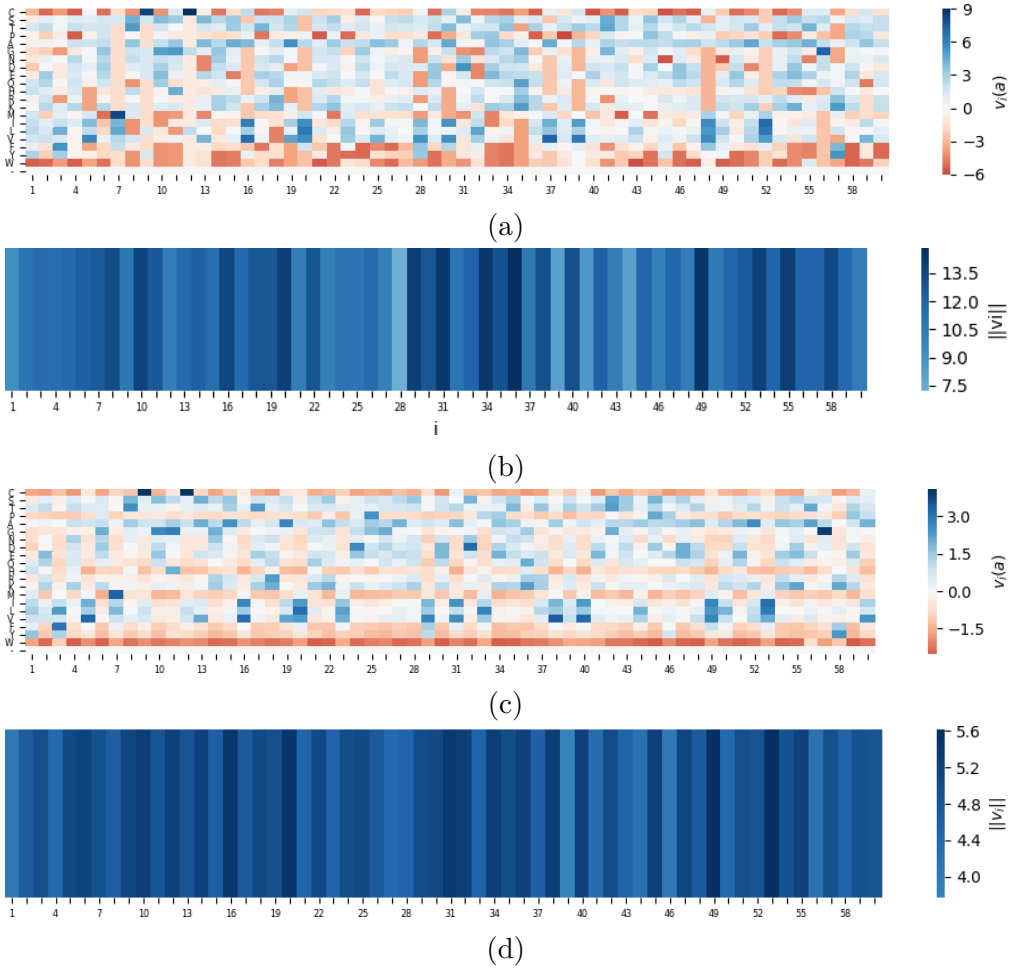


Figure 4.21 – v_i parameters and corresponding norms before (figures 4.21a and 4.21b) and after (figures 4.21c and 4.21d) having applied substitution matrix pseudo-counts with $\tau_v = 0.5$.

Overall, one can see that substitution matrix pseudo-counts introduce a bias in the column distribution towards amino acid background distribution. For instance, since tryptophan (W) is a rare amino acid, $v_i(W)$ will almost always be significantly negative, even for small τ_v values. As a function of τ_v, this bias will inevitably make dissimilar v_i more similar to each other and each v_i less characteristic of the modeled protein.

Though this pseudo-count scheme provides interesting perspectives, as will be explained in the next section due to the inconsistencies that can occur when adding pseudo-counts on the prior in pseudo-likelihood inference we opted for another

strategy.

4.6.1.3 Post-inference smoothing strategy

The problem with adding pseudo-counts on the prior v^* before inference is that it generates inconsistencies: on the one hand, with regularization, v parameters are constrained to stay close to the prior v^* with the hope that only non-zero covariations $C_{ij} = f_{ij} - f_i f_j$ can deviate coupling parameters from 0, but on the other hand, single frequencies in the MSA do not match the independent-site model anymore, which may again lead to spurious couplings. Since our main goal when adding pseudo-counts on single frequencies is to make two field vectors more comparable, we implemented a post-processing smoothing strategy inspired from the uniform pseudo-count scheme after Potts model inference.

Our strategy consists in extracting the v_i probability distribution using a softmax as in 4.16:

$$p_i(a) = \frac{\exp(v_i(a))}{\sum_b \exp(v_i(b))} \tag{4.26}$$

adding pseudo-counts to these extracted probabilities:

$$\tilde{p_i}(a) = (1 - \tau_v)p_i(a) + \frac{\tau_v}{q} \tag{4.27}$$

and reverting back to v_i:

$$\tilde{v_i}(a) = \log \tilde{p_i}(a) - \frac{1}{q}\sum_b \log \tilde{p_i(b)} \tag{4.28}$$

This also allows for more flexibility, since smoothing can be applied at any rate τ_v without having to re-infer the models. This is the strategy we implemented in our worfklow.

4.6.2 A need for more comparable coupling matrices to capture more remote homologs

4.6.2.1 How lack of data causes misleading anticorrelations

In theory, coupling values inside a w_{ij} matrix are supposed to deviate positively or negatively from 0 to reflect a (direct) correlation or anti-correlation. For instance, if i and j are in an electrostatic interaction, then one should expect couplings between two positively charged letters or two negatively charged letters to be negative, reflecting the compatibility constraints dictated by their spatial proximity. In practice, $w_{ij}(a, b)$ is negative when the three following conditions are met:

- a is frequently found at position i
- b is frequently found at position j
- a and b are not found together at positions i and j

on the multiple sequence alignment used to train the Potts model. While this will properly capture above-mentioned anti-correlations between identically charged residues in contact, this means that additional spurious anti-correlations can also arise from a lack of data, more specifically from the absence of sequences where a and b appear together at positions i and j in the training set despite their existence. Indeed, while input data can be sufficient to assert that two letters a and b are likely to be found together at positions i and j, deducing that they should not be found together at positions i and j requires more examples to have sufficient countings on all pairs of a and b. Considering that our data set is limited, a large number of spurious anti-correlations can arise from a mere lack of data. This problem is illustrated figure 4.22 on artificial data.

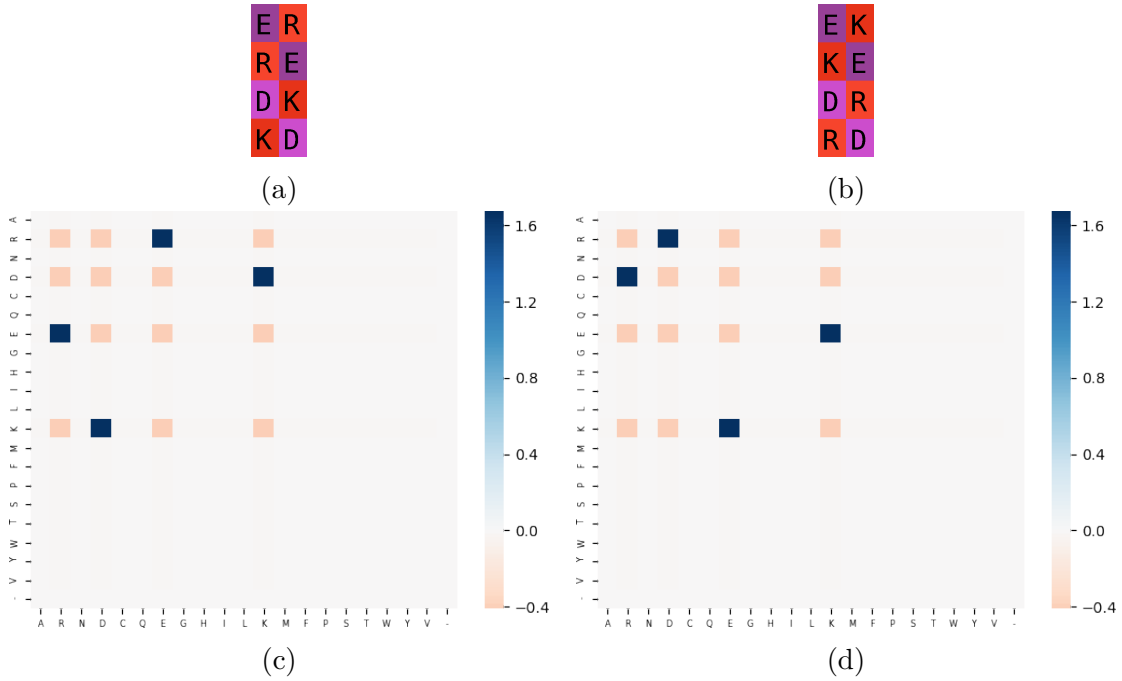


Figure 4.22 – Illustration of unwanted negative couplings on artificial data. Two pairs of columns in an electrostatic interaction (figure 4.22a and 4.22b) lead to different coupling matrices (figures 4.22c and 4.22d). While negative coupling values are correctly assigned to pairs of identically charged amino acids, some pairs of amino acids with opposite charges are also assigned the same negative coupling value since they were not in the input data. Consequently, the normalized similarity between these two coupling matrices which both reflect an electrostatic interaction is negative: $\frac{2\langle w_{ij},w_{kl}\rangle}{\langle w_{ij},w_{ij}\rangle+\langle w_{kl},w_{kl}\rangle} = -0.311$

4.6.2.2 Ideal solution: pseudo-counts on the double frequencies

The ideal solution would be to add pseudo-counts on the double frequencies before inferring the Potts model, similarly to 4.25:

$$f_{ij}(a,b) = (1-\tau_w) f_{0ij}(a,b) + \tau_w \sum_{c,d} p(ab|cd) f_{0ij}(c,d) \tag{4.29}$$

where $f_{0ij}(a,b)$ is the observed frequency of a and b at positions i and j in the initial MSA and $p(ab|cd)$ is a background probability for (c,d) to mutate into (a,b), for instance extracted from double frequencies of contacting pairs in the BLOCKS database, as proposed by the pair-to-pair double substitution matrix [Eya+07].

However, as mentioned before, this cannot be implemented in inference methods based on pseudo-likelihood since they rely on the full sequences.

4.6.2.3 Provisional patch-up for pseudo-likelihood inference: diminishing contributions of anti-correlations

Though not being able to add pseudo-counts on the double frequencies is a major drawback of CCMpredPy, its performances, along with the unprecedented possibility of choosing a prior centered at an independent-site model, still make it a good option to infer Potts models with a view to homology search until proper inference methods compiling all desirable features are developed. For this reason, we propose a provisional patch-up for the lack of pseudo-counts on double frequencies which consists in rescaling each coupling matrix after inference in order to limit the impact of negative couplings.

Unlike positional parameters, whose values are closely related to single frequencies, pseudo-counts cannot be added a posteriori on the coupling parameters, since they were inferred to account for direct couplings factoring in the whole network and thus cannot be directly linked with frequencies or covariations taken independently. Hence, our proposed solution is to limit the impact of spurious anti-correlations by limiting the impact of all anti-correlations. More specifically, since positive correlations are more likely to be supported by available training sample than negative ones, our approach here is to skew the coupling value distribution inside each w_{ij} matrix to favor higher, positive values.

One way to do this is, for one pair of positions (i, j), to retrieve the distribution of $w_{ij}(a, b)$ with a biased softmax applying more weight to positive values, apply a transformation to smooth small variations and avoid the above-mentioned log problems, and revert the softmax to obtain a different w_{ij} matrix.

More formally, we propose to retrieve the distribution using:

$$p_{ij}(a, b) = \frac{\exp(\beta_w w_{ij}(a, b))}{\sum_{c,d} \exp(\beta_w w_{ij}(c, d))} \tag{4.30}$$

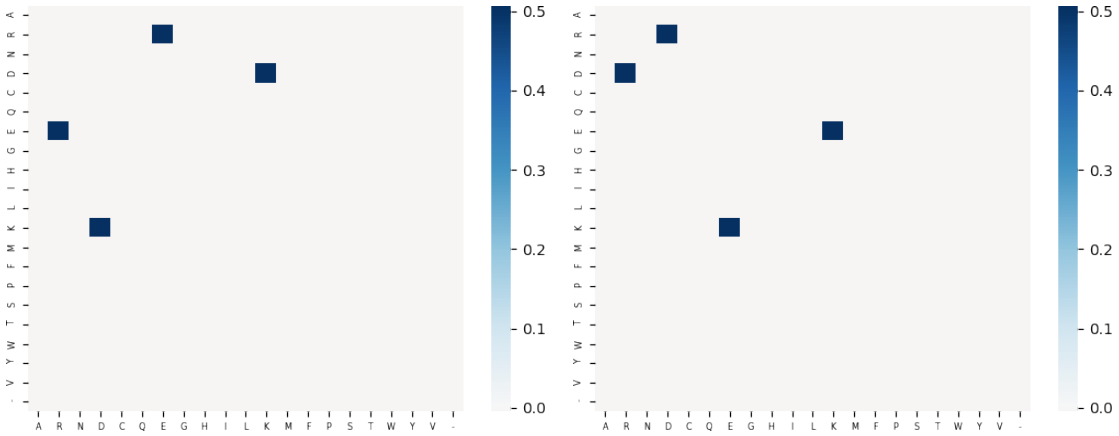
where β_w is the softmax base: the greater β_w, the more the distribution is skewed towards higher probabilities.

We smooth the distribution towards a uniform distribution:

$$\tilde{p_{ij}}(a,b) = (1-\tau_w)p_{ij}(a,b) + \frac{\tau_w}{q^2} \tag{4.31}$$

and we revert back to couplings by:

$$\tilde{w_{ij}}(a,b) = \frac{1}{\beta_w}\left(\log \tilde{p_{ij}}(a,b) - \frac{1}{q^2}\sum_{c,d}\log \tilde{p_{ij}}(c,d)\right) \tag{4.32}$$



(a) (b)

Figure 4.23 – The two coupling matrices introduced in 4.22 after having applied this smoothing scheme with $\tau_v = 0.5$. Their normalized similarity score rose up to -0.009.

This strategy amounts to focusing on positive information in the coupling matrices – which are more likely to be informative – and diminishing the importance of negative information – which might be misleading. However, information on actually relevant anti-correlations is also diminished in the process.

4.7 Summary: current recommended workflow

Several aspects of the construction of Potts models representing proteins were discussed in this chapter, and our workflow to build a Potts model from a protein sequence somewhat evolved in the course of this thesis throughout our experiments.

In this section, we recap our current recommended workflow to build a Potts model from a protein sequence, based on HHblits and CCMpredPy.

4.7.1 From sequence to train MSA

1. Retrieve close homologs by running HHblits on UniClust30 with the following options:

   ```
   -maxfilt 100000 -realign_max 100000 -all -B 100000 -Z 100000 -n 3 -e 0.001
   ```

2. Filter the resulting MSA at 80% sequence identity using HHfilter
3. Take the first 1000 effective sequences
4. Remove columns with $\geq 50\%$ gaps with `trimal -gt 0.5`

4.7.2 Potts model inference

Run CCMpredPy on the train MSA with default options.

4.7.3 Potts model post-processing

1. Re-insert trimmed positions in the Potts model as explained in section 4.3.3.2
2. Rescale inferred parameters as in sections 4.6.1.3 and 4.6.2.3 with $\tau_v = \tau_w = 0.4$ following preliminary observations, and $\beta_w = 8$ as trained in the experiment described in section 5.6.2.

4.8 Conclusion

This chapter addressed the question of building Potts models to represent proteins with a view to homology search. We established that, unlike contact prediction which simply requires coupling norms to reflect co-evolution between positions, inferring Potts models to properly represent protein properties and make it possible to compare them raises further challenges. First, we discussed the data on which

to infer a Potts model to represent a protein. Just as state-of-the-art profile and profile Hidden Markov Model based methods, we model a protein by factoring in its sequence and sequences of its close homologs, setting the number of close homologs to be considered according to previous observations from the field of contact prediction. Unlike profile Hidden Markov Models, though, Potts models do not have insertion or deletion states, hence we discussed the question of whether to include the gap symbol as the 21^{st} letter. To model protein sequences, we examined existing Potts model inference methods and opted for a state-of-the-art inference method based on pseudo-likelihood, CCMpredPy. We argued that the judicious choice of prior on the model that this method provides, which consists in initializing the model and regularizing it towards an independent-site model, is a significant step towards more canonical Potts models, complying with the intention of explaining data with positional conservation as much as possible and adding only necessary and meaningful couplings. Finally, we identified an important limit to this inference approach, which is the impossibility to implement an adequate pseudo-count scheme. We provided solutions to make parameters more comparable and improve sensitivity despite this shortcoming. However, we suspect that results described in the next chapters can be greatly improved with models inferred with a method providing an appropriate pseudo-count strategy. To our knowledge, such a method, embedding single and double pseudo-counts and an independent-site model prior, is not available yet to this day.

In summary, the two main contributions of this chapter are both the design of a workflow to build a canonical Potts model to represent a target protein sequence and the identification of areas for potential improvements in inference methods for more comparable Potts models. Our efforts are driven by the intuition that proper representations of proteins by Potts models, thanks to their parameters embedding positional conservation and direct coupling information, will allow for more sensitive pairwise comparisons and improve homology search.

Chapter 5

An exact method to align Potts models representing proteins

In the last chapter, we addressed the question of properly representing proteins with Potts models to reflect single and pairwise evolutionary constraints with a view to homology detection. Taking another step towards homology search with Potts models, in this chapter we introduce an optimal pairwise Potts model alignment method, named PPalign. We start by outlining the problem in general terms, pinpointing its underlying challenges. Then, having identified common constraints with the protein distance matrix alignment problem, we build on an existing exact method for this problem to propose an Integer Linear Programming (ILP) formulation for the pairwise Potts model problem which can be solved using the same efficient solver. This ILP involves the definition of an adequate similarity score between two Potts models, which we introduce in the next sections. We specify details of PPalign's implementation to improve computation time and its embedding in a software package suite and, finally, we lay out its first results: preliminary sequence-to-model alignments of close homologs of 1CC8, and further experimentations on alignment quality with respect to reference structural alignments.

5.1 Introduction to the pairwise Potts model alignment problem

Similarly to pairwise sequence alignment, a pairwise Potts model alignment can be defined as an assignment of one-to-one mappings of positions in the two Potts models that preserves the order of the positions. The best alignment can be found by maximizing a scoring function yielding a score for each possible alignment, with respect to constraints ensuring that the alignment is proper: aligned positions are not crossing, and a position in a Potts model cannot be aligned with more than one position in the other. An example of such an alignment is given figure 5.1.

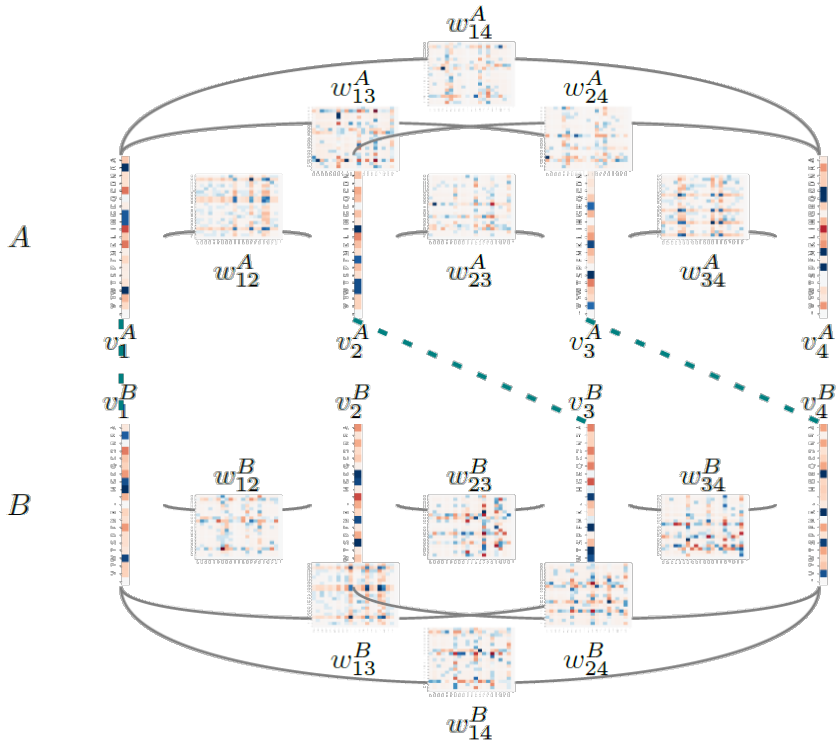


Figure 5.1 – Illustration of the alignment of two Potts models A and B.

Unlike sequence alignment or profile alignment though, the Potts model alignment problem raises a substantial computational complexity challenge: in addition to vectors v_i representing amino acid conservation at each position,

Potts models also embed a quadratic number of pairwise coupling matrices w_{ij}. Because of these non-local dependencies between positions, this problem cannot be efficiently solved using dynamic programming. Furthermore, it is assumed to be NP-hard since, as we will see in the next section, this problem is related to the distance matrix alignment problem, which can be seen as a generalization of the contact map overlap (CMO) problem which was proven to be NP-hard [GIP99]. While heuristics can provide approximations in little time, their inability to provide bounds with respect to the exact solution make it difficult to assess the relevance of a scoring scheme. For this reason, we aim at providing an exact solution or a sub-optimal solution whose score is within a chosen epsilon of the exact one. To tackle this complexity challenge, an efficient implementation is needed, which is why we resorted to existing work in the field of protein structure alignment.

5.2 Building on an exact method for the protein distance matrix alignment problem

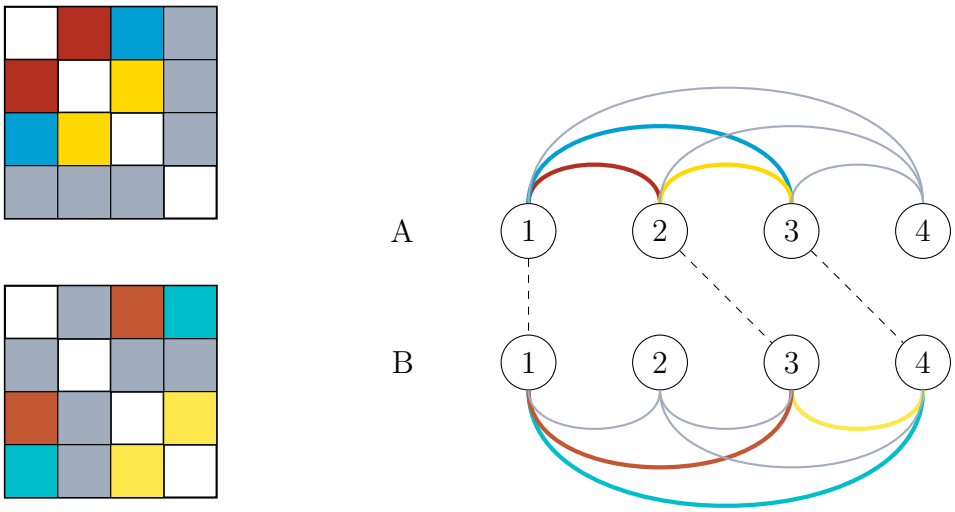
In this section, we relate the pairwise Potts model alignment problem to the distance matrix alignment problem, which enables us to build on an existing exact method to provide an Integer Linear Programming formulation for the pairwise Potts model alignment problem and use their efficient solver to solve it in tractable time.

We start by presenting the distance matrix alignment problem in section 5.2.1 before describing in section 5.2.2 the ILP formulation introduced in [WAK12] to solve this problem exactly, we briefly describe in section 5.2.3 their efficient solver and finally derive an ILP formulation for the pairwise Potts model alignment problem in section 5.2.4.

5.2.1 The protein distance matrix alignment problem

When sequence identity is low and structures are available, an alternative way of determining the similarity of two proteins is to perform structure alignment, where residues are matched according to structural information rather than sequence information. One of the most widely used pairwise structure alignment method

is to rely on the *DALI* (*Distance matrix ALIgnment*) [HS93] program. In this approach, each pair of residues (i, j) in a protein is assigned the euclidean distance between the alpha carbons of i and j, laid out in a matrix termed *inter-residue distance matrix* (see figure 5.2).



(a) (b)

Figure 5.2 – Illustration of the distance matrix alignment problem for two proteins A and B of length 4. Residue pairs are assigned their distances in the 3D structure, summarized in distance matrices (figure 5.2a). Residues are aligned so as to maximize the similarity between inter-residue distances of matching edges endpoints (figure 5.2b).

In this framework, finding the alignment of two proteins amounts to maximizing – under constraints ensuring that the alignment is proper – the DALI score between their distance matrices A and B, defined as $\sum_{i,k \ aligned} \sum_{j,l \ aligned} s_{DALI}(A_{ij}, B_{kl})$ where pairs of inter-residue distances A_{ij} and B_{kl} are scored with the following elastic score:

$$s_{DALI}(A_{ij}, B_{kl}) = \left(\theta - \frac{|A_{ij} - B_{kl}|}{\frac{1}{2}(A_{ij} + B_{kl})}\right) e^{-\left(\frac{\frac{1}{2}(A_{ij}+B_{kl})}{20}\right)^2} \tag{5.1}$$

when $i \neq j$ and $k \neq l$ and $s(A_{ii}, B_{kk})$ is set to θ for the diagonal entries, where θ is a similarity threshold in practice set to 0.2.

5.2.2 An Integer Linear Programming formulation for the distance matrix alignment problem

In [WAK12], Wohlers, Andonov and Klau proposed a mathematical model and efficient algorithm to provide an exact solution to the DALI problem. Their method, termed DALIX, extends the existing APURVA solver designed by Andonov, Malod-Dognin and Yanev for the maximum contact map overlap problem (where proteins are aligned by maximizing overlap of their contact maps) by allowing real negative score values [AMY11]. The approach is based on an Integer Linear Programming formulation, efficiently solved using a Lagrangian relaxation method.

Following [WAK12], we outline the mathematical framework leading to the ILP formulation of the distance matrix alignment problem.

Let A and B be two distance matrices for two proteins of lengths L_A and L_B. Their alignment is represented using an *alignment graph*, defined as an $L_A \times L_B$ grid graph where rows (from bottom to top) represent positions in A and columns (from left to right) represent positions in B and a node $i.k$ in the alignment graph represents the alignment of position i in the first protein and position k in the second protein. Directed edges $(i.k, j.l)$ are drawn for $i < j$ and $k < l$, representing the matching of inter-residue distances A_{ij} and B_{kl}. In this framework, an alignment of n positions in the two proteins is represented by a set of nodes $\{i_1.k_1, \cdots, i_n.k_n\}$ where $i_1 < \cdots < i_n$ and $k_1 < \cdots < k_n$, termed *increasing path*. An example of alignment graph is given figure 5.3.

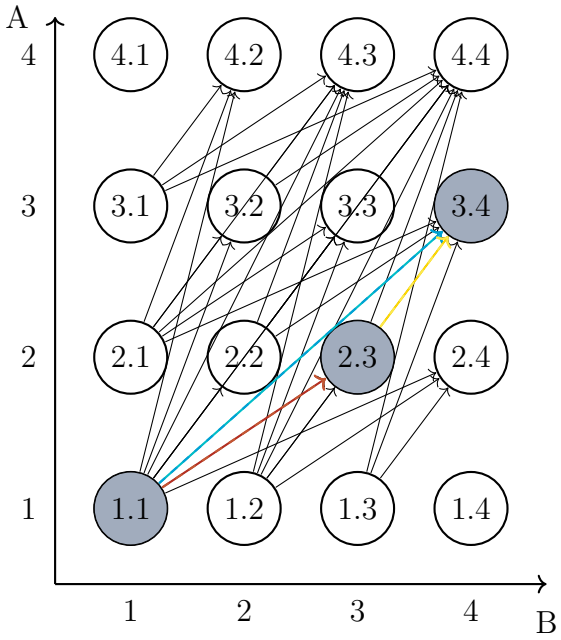


Figure 5.3 – Example of alignment graph for the two proteins in figure 5.2. Activated nodes and matched edges are colored.

To cast the alignment problem into an ILP framework, binary variables x_{ik} are assigned to each node $i.k$ in the alignment graph, with $x_{ik} = 1$ if node $i.k$ is activated, i.e. iff position i in the first protein and positions k in second protein are aligned, and similarly binary variables y_{ikjl} are assigned to each edge $(i.k, j.l)$ in the alignment graph where $y_{ikjl} = 1$ if edge $(i.k, j.l)$ is activated, i.e. iff pairs of positions (i, j) in the first protein and (k, l) in the second protein are matched.

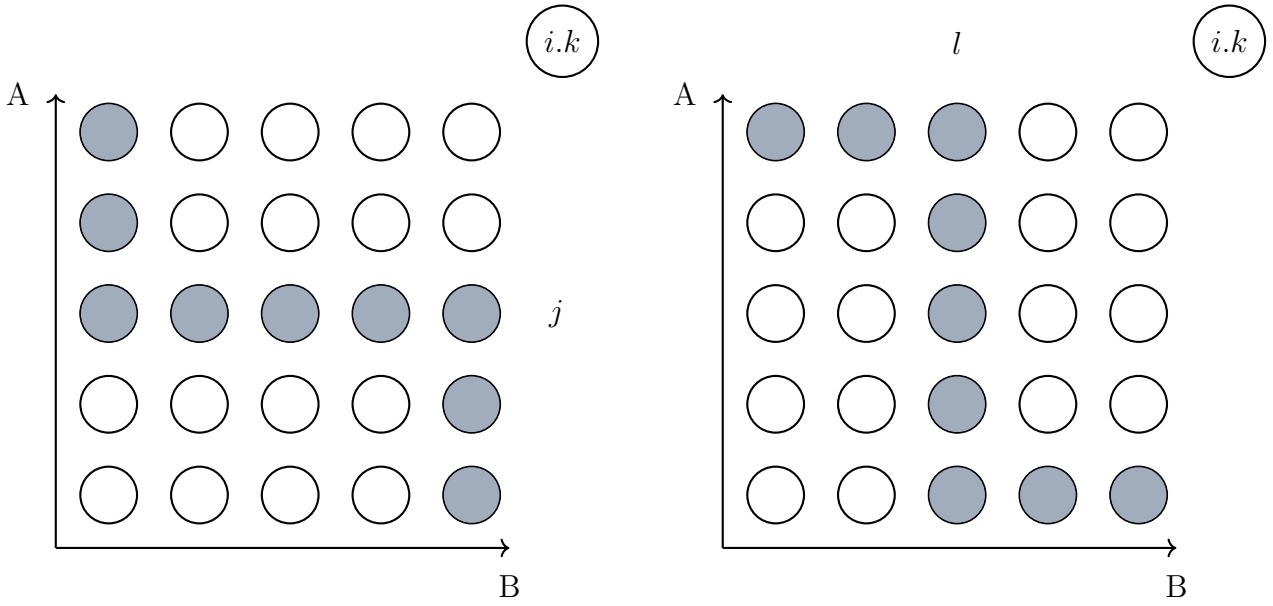
Finally, in order to properly set constraints on the alignment, for i, j in the first protein and k, l in the second protein, two additional node sets $\text{row}_{ik}(j)$ and $\text{col}_{ik}(l)$ are defined as maximal sets of nodes with endpoint at $(i.k)$ that mutually contradict (i.e. no two of them lie on an increasing path).

Formally:

$$\text{row}_{ik}(j) = \begin{cases} \{(j.1), (j.2), \cdots, (j.k-1)\} & \\ \cup\{(j+1.1), (j+2.1), \cdots, (i-1.1)\} & \\ \cup\{(1.k-1), (2.k-1), \ldots, (j-1.k-1)\} & \textit{if } j < i \\ \{(j.k+1), (j.k+2), \cdots, (j.L_B)\} & \\ \cup\{(j+1.k+1, (j+2.k+1), \cdots, (L_A.k+1) & \\ \cup\{(i+1.L_B), (i+2.L_B), \cdots, (j-1.L_B)\} & \textit{otherwise} \end{cases} \tag{5.2}$$

$$\text{col}_{ik}(l) = \begin{cases} \{(1.l), (2.l), \cdots, (i-1.l)\} & \\ \cup\{(i-1.1), (i-1.2), \cdots, (i-1.l-1)\} & \\ \cup\{(1.l+1), (1.l+2), \cdots, (1.k-1)\} & \textit{if } l < k \\ \{(i+1.l), (i+2.l), \cdots, (L_A.l)\} & \\ \cup\{(L_A.k+1), (L_A.k+2), \cdots, (L_A.l-1)\} & \\ \cup\{(i+1, l+1), (i+1, l+2), \cdots, (i+1, L_B)\} & \textit{otherwise} \end{cases} \tag{5.3}$$

Illustrations for $\text{row}_{ik}(j)$ and $\text{col}_{ik}(l)$ are given figure 5.4.



(a) $\mathrm{row}_{ik}(j)$ (b) $\mathrm{col}_{ik}(l)$

Figure 5.4 – Illustration of sets $\mathrm{row}_{ik}(j)$ (figure 5.4a) and $\mathrm{col}_{ik}(l)$ (figure 5.4b) in a situation where $j < i$ and $l < k$ (redrawn from [Woh12]). Colored nodes are mutually contradicting (no two of them lie on an increasing path) tails of edges with a common head $i.k$.

Given these definitions, the alignment of two distance matrices A and B for two proteins of lengths L_A and L_B can be formulated as:

$$\max \quad \sum_{i=1}^{L_A-1} \sum_{j=i+1}^{L_A} \sum_{k=1}^{L_B-1} \sum_{l=k+1}^{L_B} 2s(A_{ij}, B_{kl}) y_{ikjl} + \sum_{i=1}^{L_A} \sum_{k=1}^{L_B} s(A_{ii}, B_{kk}) x_{ik} \tag{5.4}$$

$$\text{s.t.} \quad x_{ik} \geq \sum_{r.s \in row_{ik}(j)} y_{ikrs} \quad j \in [i+1, L_A], i \in [1, L_A - 1], k \in [1, L_B - 1] \tag{5.5}$$

$$x_{ik} \geq \sum_{r.s \in col_{ik}(l)} y_{ikrs} \quad l \in [k+1, L_B], i \in [1, L_A - 1], k \in [1, L_B - 1] \tag{5.6}$$

$$x_{ik} \geq \sum_{r.s \in row_{ik}(j)} y_{rsik} \quad j \in [1, i-1], i \in [2, L_A], k \in [2, L_B] \tag{5.7}$$

$$x_{ik} \geq \sum_{r.s \in col_{ik}(l)} y_{rsik} \quad l \in [1, k-1], i \in [2, L_A], k \in [2, L_B] \tag{5.8}$$

$$x_{ik} \leq \sum_{\substack{r.s \in row_{ik}(j) \\ s(A_{ri}, B_{sk}) \leq 0}} (y_{rsik} - x_{rs}) + 1 \quad j \in [1, i-1], i \in [2, L_A], k \in [2, L_B] \tag{5.9}$$

$$\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1 \quad i \in [1, L_A], k \in [1, L_B] \tag{5.10}$$

$$x, y \text{ binary} \tag{5.11}$$

Constraints (5.5) and (5.6) prevent edges from activating if their tails are not activated and ensure that heads of edges with a common tail do not contradict, and constraints (5.7) and (5.8) denote the reverse situation. Constraint (5.9) ensures that edges are activated if their heads and tails are activated (this constraint is necessary since similarity scores can be negative). Finally, constraint (5.10) ensures that the nodes lie on an increasing path.

Though their work was mainly driven by the distance matrix alignment problem, as explained in Wohler's thesis [Woh12] their framework is actually applicable to any two-dimensional scoring scheme for the protein structure alignment problem, i.e. any other scoring function where structural information is assigned to pairs of residues. Their exact algorithm made it possible to properly compare different scoring schemes without potential biases caused by implementations based on heuristics [Woh+12].

5.2.3 An efficient solver

To solve this problem, authors of [WAK12] proposed an efficient solver which extends the A_PURVA solver [AMY11] for Maximum Contact Map Overlap with real-valued scores, and which is able to yield the exact solution in tractable time.

Their approach is based on a Lagrangian relaxation: constraints (5.7), (5.8) and (5.9) are relaxed, allowing edges to be activated ($y_{ikjl} = 1$) when their heads are not ($x_{kl} = 0$), and moved to the objective function with Lagrangian multipliers to penalize their violation. An optimal solution for the resulting relaxed problem, which constitutes an upper bound for the original problem, can be found in $O(L_A^2 L_B^2)$ using double dynamic programming, along with an induced feasible solution, which constitutes a lower bound. Lagrangian multipliers, initially set to 0, are iteratively updated using a subgradient descent method, leading to a new relaxed problem, until the difference between the upper bound and the lower bound is smaller than a chosen ϵ.

This Lagrangian approach is embedded into a Branch-and-Bound framework, recursively splitting the solution space into smaller spaces and pruning candidates that cannot be part of the optimal solution. This algorithm has an exponential worst case runtime, but in practice the solver performs remarkably well on structure alignment [Mav+10; Mal+11].

5.2.4 Deriving a general Integer Linear Programming formulation for the Potts model alignment problem

Just as the pairwise Potts model alignment problem, the distance matrix alignment problem aims at aligning two proteins with pairwise dependencies. An Integer Linear Programming formulation for the pairwise Potts model alignment problem can be directly derived from the Integer Linear Programming formulation for the protein distance matrix alignment problem given in 5.2.2, using the same constraints.

The framework is general enough to be extended to any two-dimensional scoring scheme, and though the scoring function is initially based on similarities between pairs of positions only, it is straightforward to introduce a similarity score between

positions as well, along with a coefficient to balance the two scores.

This way, the problem of aligning two Potts models A and B of parameters $(\boldsymbol{v^A}, \boldsymbol{w^A})$ and $(\boldsymbol{v^B}, \boldsymbol{w^B})$ can be formulated with the same ILP formulation except for the objective function which we can define as a sum of similarity scores between field parameters $s_v(v_i^A, v_k^B)$ and similarity scores between coupling parameters $s_w(w_{ij}^A, w_{kl}^B)$ with a coefficient α_w to balance the two:

$$\max \quad \sum_{i=1}^{L_A}\sum_{k=1}^{L_B} s_v(v_i^A, v_k^B)x_{ik} + \alpha_w \sum_{i=1}^{L_A-1}\sum_{j=i+1}^{L_A}\sum_{k=1}^{L_B-1}\sum_{l=k+1}^{L_B} s_w(w_{ij}^A, w_{kl}^B)y_{ikjl} \tag{5.12}$$

$$\text{s.t.} \quad x_{ik} \geq \sum_{r.s \in row_{ik}(j)} y_{ikrs} \quad j \in [i+1, L_A], i \in [1, L_A - 1], k \in [1, L_B - 1] \tag{5.13}$$

$$x_{ik} \geq \sum_{r.s \in col_{ik}(l)} y_{ikrs} \quad l \in [k+1, L_B], i \in [1, L_A - 1], k \in [1, L_B - 1] \tag{5.14}$$

$$x_{ik} \geq \sum_{r.s \in row_{ik}(j)} y_{rsik} \quad j \in [1, i-1], i \in [2, L_A], k \in [2, L_B] \tag{5.15}$$

$$x_{ik} \geq \sum_{r.s \in col_{ik}(l)} y_{rsik} \quad l \in [1, k-1], i \in [2, L_A], k \in [2, L_B] \tag{5.16}$$

$$x_{ik} \leq \sum_{\substack{r.s \in row_{ik}(j) \\ s(A_{ri}, B_{sk}) \leq 0}} (y_{rsik} - x_{rs}) + 1 \quad j \in [1, i-1], i \in [2, L_A], k \in [2, L_B] \tag{5.17}$$

$$\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1 \quad i \in [1, L_A], k \in [1, L_B] \tag{5.18}$$

$$x, y \ \textit{binary} \tag{5.19}$$

This general formulation for the pairwise Potts model alignment problem can be solved exactly using DALIX's efficient solver.

Now, we need to decide on appropriate functions s_v and s_w to score the similarity of Potts models parameters.

5.3 Introducing a natural similarity score for two Potts models

5.3.1 The scalar product as a natural candidate

Our search for scoring functions s_v and s_w for the similarity of two field parameters and two coupling parameters was guided by the intuition that the score between two parameters should be maximal when:

- parameters are similar
- parameters reflect features that are important for the characterization of the modeled set of proteins

In other words, two fields (resp. couplings) should be given a high score if the amino acid distributions captured by the vectors (resp. the correlations or anti-correlations captured by the matrices) are similar, and if they were assigned a higher weight during Potts model inference. Indeed, as covered in the previous chapter, unlike profile Hidden Markov Models whose match states consist of probability vectors (thus intrinsically normalized) weights of Potts models are globally distributed on the fields and couplings so as to maximize the likelihood (or pseudo-likelihood) of the training set, potentially assigning more weight to some positions or pairs of positions and hopefully reflecting important conserved properties.

Following this line of thought, we chose to score the alignment of two Potts models using the scalar product:

$$\langle v_i^A, v_k^B \rangle = \sum_a v_i^A(a) v_k^B(a) \tag{5.20}$$

$$\langle w_{ij}^A, w_{kl}^B \rangle = \sum_{a,b} w_{ij}^A(a,b) w_{kl}^B(a,b) \tag{5.21}$$

This scoring scheme factors in both similarity and importance of parameters, since the scalar product between two vectors X and Y can be rewritten as:

$$\langle X, Y \rangle = \|X\| \, \|Y\| \cos\theta \tag{5.22}$$

where θ is the angle between the two vectors. $\cos\theta$ reflects how aligned the two vectors are, and $\|X\|$ and $\|Y\|$ reflect their respective importance (in our case hopefully positional residue conservation or interaction strength between residues).

Moreover, this scoring function appears as a natural choice since it extends the score of a sequence for a given Potts model. Indeed, by modeling a sequence $x = x_1, \cdots, x_L$ in a one-hot encoding fashion:

- $\forall i, a, e_i(a) = \delta(a, x_i)$
- $\forall i, j, a, b, e_{ij}(a, b) = \delta(a, x_i)\delta(b, x_j)$

where δ is the Kronecker symbol ($\delta(x, y) = 1$ iff $x = y$), setting $\alpha_w = 1$, its similarity score with a Potts model A of length L and parameters (v, w) is given by:

$$s(A, x) = \sum_{i=1}^{L} \langle v_i, e_{x_i} \rangle + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \langle w_{ij}, e_{x_i x_j} \rangle \tag{5.23}$$

which can be rewritten as:

$$s(A, x) = \sum_{i=1}^{L} v_i(x_i) + \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} w_{ij}(x_i, x_j) \tag{5.24}$$

which is exactly the energy (Hamiltonian) of sequence x with respect to Potts model A (see section 3.3.3) with the opposite sign.

5.3.2 Comparison with respect to background

Inspired by sequence alignment methods which use log-odds ratios to compute their scores with respect to a background model, we remove the background field v_0 defined in equation (4.5) to each field vector before computing the scalar product. The actual similarity score between two positional parameters v_i^A and v_k^B used in this paper is thus:

$$s_v(v_i^A, v_k^B) = \langle v_i^A - v_0, v_k^B - v_0 \rangle \tag{5.25}$$

This can be interpreted as comparing the distance between v_i^A and background to the distance between v_k^B and background. In the case where Potts models are

independent-site, this can also be seen as computing the similarity of fields built with log-odds rather than frequencies:

$$\begin{aligned} v_i^A - v_0 &= \left(\log f_i(a) - \sum_b \log f_i(b)\right) - \left(\log f_0(a) - \sum_b \log f_0(b)\right) \\ &= \log \frac{f_i(a)}{f_0(a)} - \sum_b \log \frac{f_i(b)}{f_0(b)} \end{aligned} \tag{5.26}$$

Considering a null model without couplings, the background coupling w_0 is set to 0, hence the similarity score between two coupling parameters w_{ij}^A and w_{kl}^B remains:

$$s_w(w_{ij}^A, w_{kl}^B) = \langle w_{ij}^A, w_{kl}^B \rangle \tag{5.27}$$

5.4 Gap cost and offset

To account for insertions and deletions, we make use of the gap cost strategy implemented in the DALIX solver, which consists in an affine gap penalty:

$$\gamma(g) = -\gamma_o - \gamma_e g \tag{5.28}$$

Picking the gap open penalty coefficient γ_o and the gap extend penalty coefficient γ_e is particularly challenging in the case of Potts model alignment. Indeed, unlike profile columns, parameters to be compared are unnormalized and can theoretically take any real value. While this provides the unprecedented possibility to give more weight to conserved properties important for the protein family, it also makes it difficult to decide on default hyperparameters such as a gap open and a gap extend penalty. In practice, we use parameters trained on reference structure alignments as described in section 5.6.2: $\gamma_o = 13$ and $\gamma_e = 0$.

As currently implemented, this gap cost scheme penalizes internal gaps as much as external gaps. Future work should probably implement different costs for external gaps to better handle the alignment of smaller sequences with longer sequences.

Furthermore, as in most profile-profile methods [WD04], in order to prevent our method from greedily aligning every position, we penalize each aligned pair

with a fixed negative offset hyperparameter. As for gap cost parameters, due to the unbounded nature of Potts model parameters, defining the best offset is a complex task. As a first step, we remove the same value to each column. In practice we use an offset of 1, as yielded by the hyperparameter optimization described in 5.6.2.

Finally, the function to be optimized becomes:

$$S(A, B) = s(A, B) + s_{gap} + s_{offset} \tag{5.29}$$

where s_{gap} and s_{offset} are the scores associated with gap cost and offset and s is the similarity function:

$$s(A, B) = \sum_{i=1}^{L_A} \sum_{k=1}^{L_B} s_v(v_i^A, v_k^B) x_{ik} + \alpha_w \sum_{i=1}^{L_A-1} \sum_{j=i+1}^{L_A} \sum_{k=1}^{L_B-1} \sum_{l=k+1}^{L_B} s_w(w_{ij}^A, w_{kl}^B) y_{ikjl} \tag{5.12}$$

5.5 Implementation

5.5.1 Practical choices to speed up computations

5.5.1.1 Stopping computations when precision is high enough

As mentioned before, the ILP solver can yield a solution as close to the exact solution as wanted by stopping the computations when the difference between the upper bound and the lower bound is small enough: $UB - LB \leq \epsilon$. In our experience, the solver quickly finds the optimal solution, but spends a significant amount of time checking that there is no better solution. An appropriate choice of $\epsilon > 0$ allows for a significant speed up in computation time with a precision guarantee on the solution – in practice often the exact solution.

Since our scoring function is not normalized and can take a wide range of values, we define a normalized alignment scoring function S_{norm} between Potts models A and B by dividing it with the mean of alignment scores of each Potts model with itself:

$$S_{norm}(A, B) = \frac{2S(A, B)}{S(A, A) + S(B, B)} \tag{5.30}$$

and we stop computations when the normalized scores of the upper bound and

the lower bound solutions are smaller than a chosen ϵ:

$$\frac{2UB}{S(A,A)+S(B,B)} - \frac{2LB}{S(A,A)+S(B,B)} \leq \epsilon \tag{5.31}$$

In practice, we found $\epsilon = 0.005$ to yield almost always the exact solution in significantly less time, and $\epsilon = 0.02$ to further reduce the computation costs by yielding reliable approximations.

5.5.1.2 Stopping computations if the models are not similar enough

In the specific case of homology detection, one is often not interested in the alignment itself but simply in deciding whether two proteins are homologous or not. In this case, experiments can be substantially sped up by stopping computations when the normalized score of the upper bound as defined in equation (5.30) is lower than a given threshold. In practice we set this threshold to 0.

5.5.2 PPalign implementation as part of PPsuite

Our method was implemented in a software tool termed PPalign. We slightly adjusted the C++ code of the DALIX solver kindly provided by Wohlers et al. to implement our own objective function and embedded it into a Python package thanks to the *ctypes* library, in which we implemented functions to perform all pre-processing steps such as parameter rescaling and to provide sequence alignments in FASTA format from the solver's output.

Along with the alignment tool itself, we included tools to build Potts models starting from a sequence or a sequence set using different workflows introduced in the previous chapter and different visualization tools which were used to generate illustrations in this thesis (inferred Potts model parameters, predicted couplings, PPalign alignment scores) as a whole software package termed PPsuite, available at https://github.com/htalibart/ppsuite.

5.6 Alignments with PPalign

In this section, we focus on PPalign's results as an alignment method. As preliminary experiments, we looked at sequence-Potts model alignments by re-aligning single sequences previously aligned by HHblits to a Potts model using a one-hot encoding and examine the differences between the two alignments and the computation time. Then, we validate our method on Potts model-Potts model alignments on a set of reference structural alignments with low sequence identity, optimizing our hyperparameters towards accurate remote homolog alignments in the process.

5.6.1 Preliminary experiments on sequence-model alignments

Since our similarity score between two Potts models is the extension of the 1D score for a sequence, we ran preliminary experiments on sequence to model alignments with PPalign to check its computational tractability on this problem and estimate how its alignments differ from the alignments made by HHblits.

To this end, we inferred a Potts model for 1CC8 on the M_{train} MSA described in section 4.4 and used PPalign to re-align 1000 close homologs in the S_{close} set by building a Potts model for each homolog sequence $x = x_1, \cdots, x_L$ with a *one-hot encoding* approach, where parameters were set as described in section 5.3.1, i.e.:

- $v_i(a) = \delta(a, x_i)$
- $w_{ij}(a, b) = \delta(a, x_i)\delta(b, x_j)$

We rescaled parameters of the Potts model for 1CC8 as described in sections 4.6.2.3 and 1.1.2.1 using $\tau_v = \tau_w = 0.4$ in order to smooth parameters without flattening signal too much and $\beta_w = 8$ as trained in our experiments on reference structural alignments (see next section). Other hyperparameters of PPalign were set so that its objective function corresponds to the sequence's 1D score, i.e. without offset nor comparison with respect to background and setting the w score coefficient α_w to 1, except for the gap extend penalty which we set to 13 according to the next section's training. We set ϵ to 0.005 which, in our experience, yields the

same alignments as the optimal ones in significantly less time. We also ran PPalign without coupling ($\alpha_w = 0$), termed PPalign-1D, to evaluate the contribution of pairwise couplings.

We compared PPalign's alignments with the original alignments made by HHblits in the M_{close} MSA by computing the Matthews correlation coefficient:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5.32}$$

where:

- TP is the number of pairs aligned by HHblits and PPalign
- FP is the number of pairs aligned by PPalign and not by HHblits
- FN is the number of pairs aligned by HHblits and not by PPalign
- TN is the number of pairs that were not aligned by HHblits or PPalign, in practice $TN = (L_A + 1) \times (L_B + 1) - TP - FP - FN$

Overall, PPalign's alignments of 1CC8's Potts model with sequences' one-hot encodings are similar to HHblits' alignments, with an average MCC of 0.9775. MCCs of PPalign without couplings are slightly lower (0.9773 on average). Lowest correlations (down to 0.6080) are mainly achieved for sequences that do not feature the two strongly conserved cysteins, which is not surprising since during Potts model inference (and rescaling) this strongly conserved property resulted in two high norm field vectors where all letters were assigned negative values except for cystein.

Alignments with and without coupling score are similar, with an average Matthews coefficient of 0.9900, but a few alignments are significantly different (down to an MCC of 0.4198 for sequence *A0A1I4FB16*). Most of the time, these lower correlations coincide with a significant difference in total gap cost between PPalign and PPalign-1D – suggesting that an excessive gap penalty prevented PPalign-1D from opening a gap to align some positions, which was compensated by a high coupling score in standard PPalign – except for one sequence (*A0A074Z5C0*), which does not feature the two strongly conserved cysteins.

In terms of computation time, it took on average 24 seconds to align each sequence on a Fedora desktop computer with 15G RAM and 4 CPUs, with uneven computation times ranging from 2 seconds to 1h12 in the worst case. Computation times with respect to sequence lengths are plotted figure 5.5. Computations without coupling scores were much faster, with an average of 0.04 seconds.

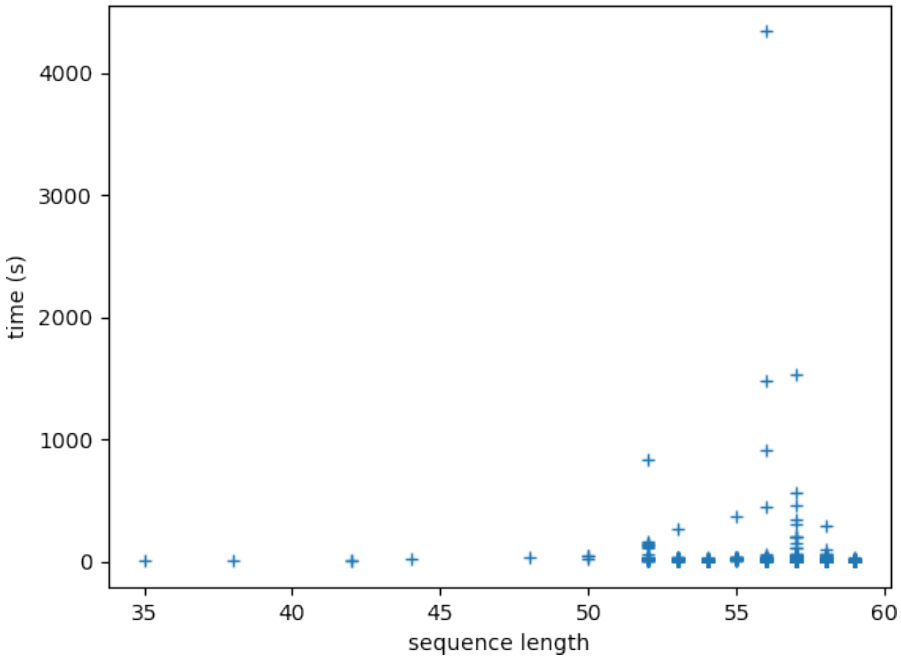


Figure 5.5 – PPalign computation time for the alignment of each sequence in the S_{close} set to the Potts model inferred on the M_{train} MSA as a function of sequence lengths. Computation time seem to be partly related to the length of the sequence to be aligned.

These preliminary experiments on close homologs show that, although taking couplings into account significantly slows down computations, on average PPalign can perform sequence to model alignments in tractable time. This could be used to build train MSAs starting from a seed MSA to rely less on HHblits' alignments which are based on positional conservation. However, in this case there is no ground truth we can rely on to know which alignment is best.

5.6.2 Validation experiments on model-model alignments

5.6.2.1 Data

To evaluate PPalign and the contribution of distant dependencies, we focused on reference alignments based on structures with low sequence identity. We opted for SISYPHUS database [And+07] since it provides manually curated structural alignments for proteins with non-trivial relationships. Our data set was built as follows:

- From each multiple sequence alignment in SISYPHUS, every possible pairwise sequence alignment with a sequence identity lower than 20% was extracted (we set a low sequence identity threshold to focus on harder targets)
- For each sequence in each of these extracted pairwise reference alignments, we attempted to build a Potts model with the workflow previously described. Sequences that had less than 1000 effective homologs were discarded to focus on sequences with sufficient co-evolution signal. Due to CCMpredPy memory consumption, trimmed MSAs whose length was longer than 200 also had to be discarded.
- Finally, for each reference multiple sequence alignment in SISYPHUS with more than two of such eligible sequences, a reference sequence pair was randomly selected. This last steps discards many alignment pairs but ensures that no multiple sequence alignment biases the results.

This resulted in a set of 33 non-redundant reference pairwise alignments which was randomly split into a train set of 11 alignments on which our hyperparameters were trained (see table 5.1) and a test set of 22 target alignments (see table 5.2).

Potts models were built using the workflow described in section 4.7.

Table 5.1 – Training set.

MSA	sequences	sequence identity (%)
AL10050464	1r5bA_559_659, 1r5bA_470_549	3.85
AL00053697	1vimA_36_164, 1iatA_334_500	4.04
AL00063412	1bccA_34_201, 1ezvB_236_357	5.59
AL00051306	1ay9A_51_137, 1b12A_81_302	6.28
AL00052113	1kzyC_1731_1838, 1in1A_853_916	8.60
AL10069117	1kncA_13_172, 2gmyA_14_141	9.09
AL00050815	1i4uA_33_167, 1np1A_21_166	10.00
AL00054790	1vig_10_72, 1k1gA_136_223	11.36
AL00054403	4monA_6_47, 1roaA_23_119	13.33
AL00048098	1cmzA_90_199, 1omwA_54_168	13.91
AL00089800	1p6oA_10_147, 1wkqA_2_150	17.88

Table 5.2 – Test set.

MSA	sequences	sequence identity (%)
AL00050475	1ci0A_43_200, 1uscA_12_145	3.61
AL00050692	1uheA_11_87, 1q16A_1084_1225	4.14
AL10050815	1exsA_17_124, 1qftA_27_139	5.04
AL10050875	1rbp_19_140, 1hms_3_131	5.19
AL00050715	1dfuP_2_94, 1qtqA_340_541	5.22
AL00055723	1tu1A_1_140, 1v2bB_18_186	5.81
AL00050799	1pklA_88_180, 1o65A_12_173	6.02
AL00074653	1tolA_151_213, 1ihrA_172_230	6.15
AL10063410	1qf6A_68_223, 1hr6B_48_215	6.29
AL00053335	1ri5A_51_291, 1nv8A_106_279	7.43
AL10050155	1k32A_764_851, 1lcyA_228_321	9.62
AL10050335	1h9mA_5_141, 1v43A_247_366	10.22
AL10074933	1k32A_763_852, 1te0A_257_349	10.68
AL00052141	1mwiA_9_163, 1oe4A_87_277	11.48
AL20089447	1z0rA_8_48, 1n0gA_33_142	12.93
AL00047241	1tjoA_29_171, 1lb3A_15_153	13.01
AL00054814	1egaB_197_282, 1hh2P_199_275	13.40
AL00050021	1jm1A_57_211, 1nykA_54_191	14.61
AL00047861	1m12A_3_74, 1n69B_2_73	15.38
AL00052441	1c30A_7_127, 1w93A_59_184	15.38
AL00054407	1eqkA_11_95, 2ch9A_38_144	15.74
AL00052787	5pnt_5_155, 1jl3A_3_137	17.72

5.6.2.2 PPalign hyperparameter optimization

PPalign's hyperparameters were optimized on the 11 alignments from the training set using Hyperopt library [**hyperopt**] to maximize the F_1 score, where:

$$P = \frac{\#\ correctly\ aligned\ pairs}{\#\ aligned\ pairs\ in\ computed\ alignment} \tag{5.33}$$

and recall:

$$R = \frac{\#\ correctly\ aligned\ pairs}{\#\ aligned\ pairs\ in\ reference\ alignment} \tag{5.34}$$

using Edgar's qscore program [Edg] v2.1, and F_1 score:

$$F_1 = \frac{2PR}{P+R} \tag{5.35}$$

This process showed to be excessively time-consuming, Hyperopt being unable to show a convergence on the choice of the parameters after one month. In order to reduce the hyperparameter search space and speed up the convergence of this process, we had to arbitrarily set some parameters after some trials on the training set: precision ϵ was set to 0.02, τ_v and τ_w from sections 4.6.1.3 and 4.6.2.3 were both set to 0.4 and the gap extend penalty was set to 0. In accordance with the expected NP-hardness of the problem, time needed to find optimal alignment could be very long for some sets of parameters and even exceed the 6 hours time-out we set. We observed yet that good alignments were usually already found in less than 1 minute and decided to set the time-out by alignment to this value to speed-up more the optimisation of the remaining parameters by Hyperopt, which yielded the following values:

- Gap open penalty: 13
- Coupling contribution coefficient α_w: 6
- Softmax base β_w: 8.0
- Offset γ: 1.0

5.6.2.3 Other methods to be compared

In this experiment, we compared the results of PPalign with:

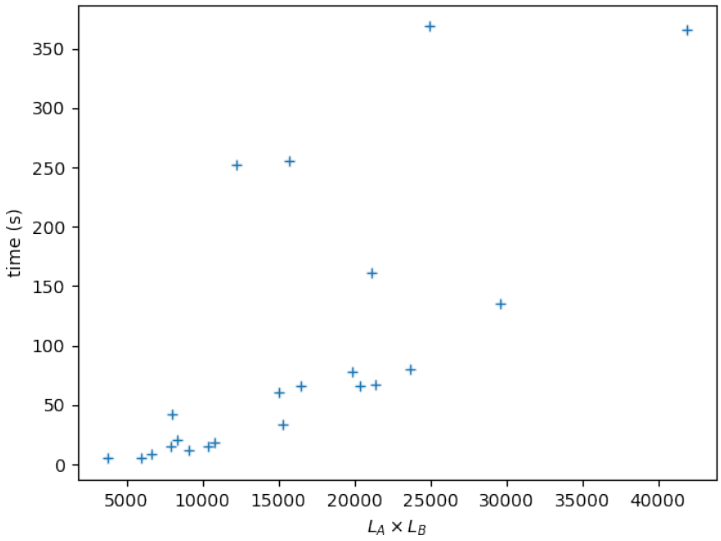
- PPalign without coupling score, i.e. $\alpha_w = 0$ (termed PPalign-1D)
- HHalign v3.0.3, run with default options to align pHMMs built with HHmake with default options from the MSAs used to infer Potts models (except for the trimming of the positions with $> 50\%$ gaps since pHMMs handle well insertions and deletions)
- BLASTp v2.9.0+ without E-value cutoff, run on the sequences truncated as in our training MSAs, to provide an indication on the sequences' similarity

5.6.2.4 Results

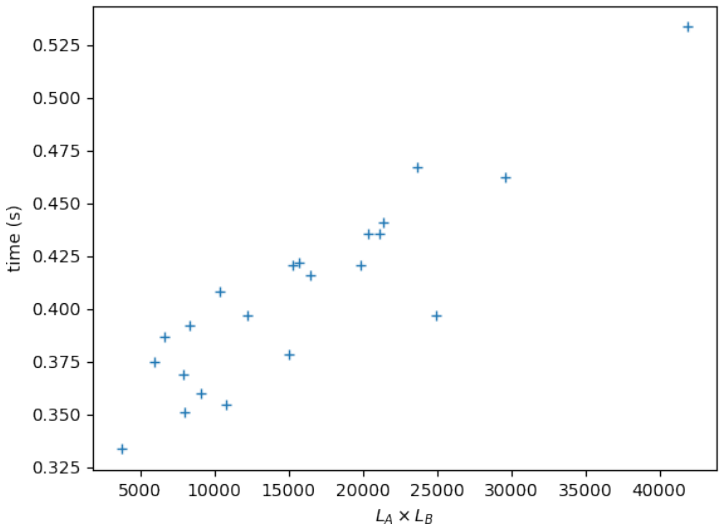
Tractable computation time. We examined the computation times of PPalign, PPalign-1D and HHalign, considering the time they took to align the models (and not the steps to build them, that can be done offline) of the sequence pairs from the test set. Experiments were run on a Debian9 virtual machine with 4 VCPUs (2.3 GHz) and 8 GB RAM. The timeout for each alignment was set to 6 hours. The first result is that all the alignments could be computed by PPalign in running times ranging from 5 seconds to 6 minutes, with an average of 1 min 36. Figure 5.6a plots the running times with respect to the lengths of the models to align. It shows that most problems (17/22) are easily solved and that running time for these problems increases gently with the lengths of the models, while a few (5/22) other problems stand out from this majority trend but are still solved in a few minutes.

When couplings are not considered, the problem is fundamentally easier and running times of HHalign and PPalign-1D are significantly faster than PPalign: both programs were able to compute each optimal positional alignment in less than 1 second. The running times of HHalign and PPalign-1D are plotted in Figure 5.6b and Figure 5.6c. The two plots are not completely comparable since time needed to load the models is here included for HHalign and not for PPalign-1D, but they illustrate the difference between the dynamic programming approach of HHalign, with a steady running time increment with the length of the models, and the

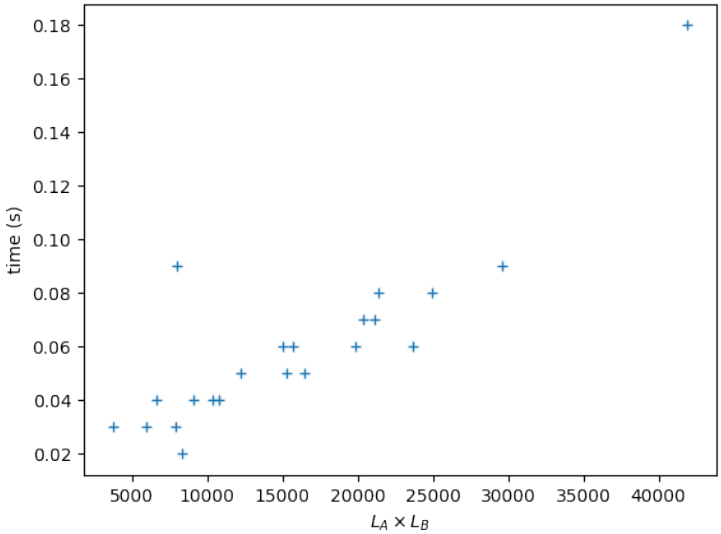
Integer Linear Programming optimization approach of PPalign-1D, showing here 2 outliers with respect to the general tendency.



(a) PPalign.



(b) HHalign.



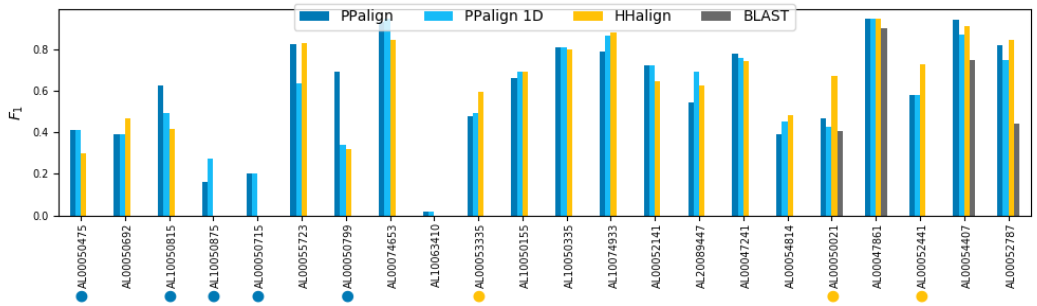
(c) PPalign-1D.

Figure 5.6 – Time for aligning models of lengths L_A and L_B for sequence pairs from test set.

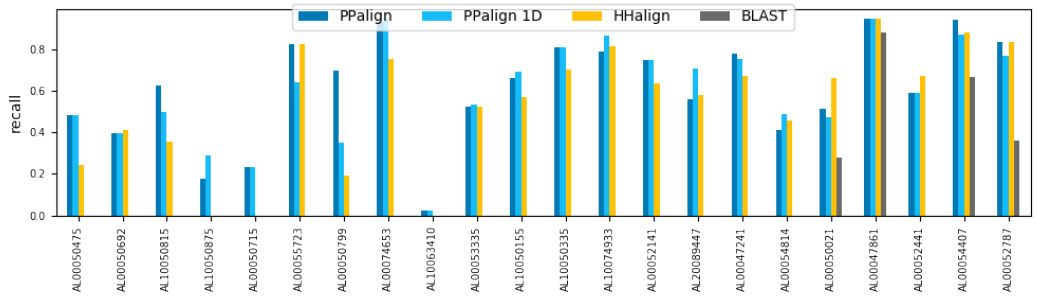
Alignment quality Alignment quality was assessed by comparing the alignment obtained by the different methods for the 22 sequences pairs in the test set to their reference alignment.

Overall, PPalign achieves a better F_1 score than HHalign (0.600 versus 0.578) with a better recall (0.613 vs 0.533) but a lower precision (0.587 vs 0.661), outperforming it in 12 out of the 22 alignments. BLAST only aligned 4 out of the 22 pairs, yielding an average F_1 score of 0.113.

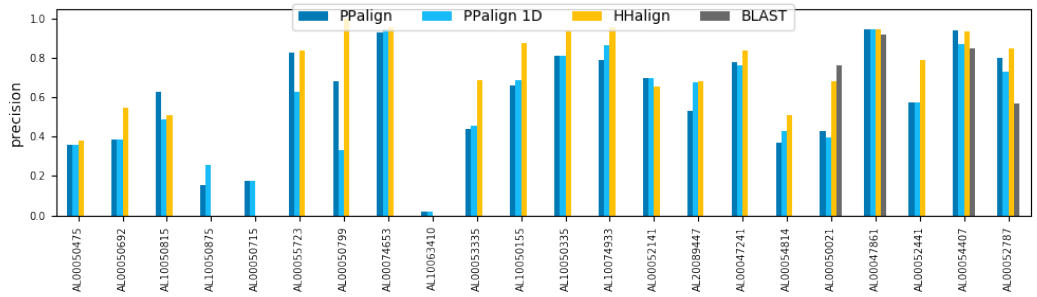
Results for each sequence pair of the test set are displayed in Figure 5.7.



(a) F_1 measure



(b) Recall



(c) Precision

Figure 5.7 – Quality of the alignments computed by PPalign, PPalign-1D, HHalign and BLAST with respect to target reference alignments in test set (ordered by increasing percentage of sequence identity).

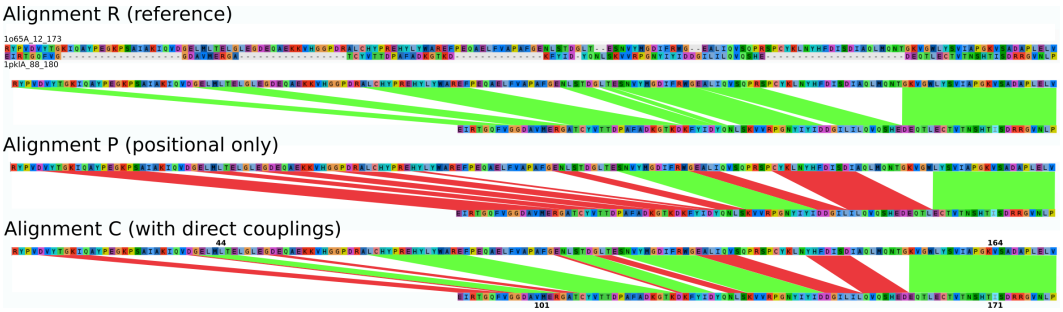
In most cases, PPalign and HHalign yield similar F_1 scores (with less than 0.1 difference), except for 8 sequence pairs. 5 of them, marked by blue dots in the

Figure 5.7a, are significantly better aligned by PPalign: AL00050475, AL00050692, AL10050875, AL00050715 and AL00050799 which are among the 7 alignments with the smallest percentage of sequence identity with respectively 3.61%, 5.04%, 5.19%, 5.22% and 6.02%. AL10050875 and AL00050715 are part with AL10063410 of the three sequence pairs that HHalign fails completely to align, yielding small and incorrect alignments with an F_1 score of 0. On AL10063410, PPalign also failed, but on AL10050875 and AL00050715 it was able to do a bit better than HHalign by correctly aligning in each case roughly a fifth of the target alignment while still being wrong on the four other fifths. On AL00050475 and AL00050692, PPalign successfully retrieves about half of the target alignments when HHalign was retrieving only respectively a fifth and a third of it. The contribution of the coupling parameters is particularly noticeable for AL00050799, PPalign correctly retrieving almost 70% of the alignment while HHalign retrieves only 20% of it (see detailed analysis in Figure 5.8).

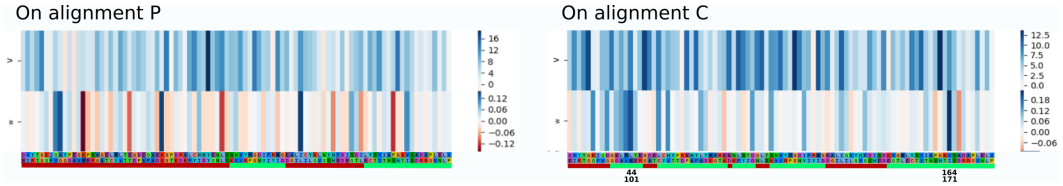
PPalign is significantly outperformed by HHalign on 3 pairs, marked by yellow dots in Figure 5.7a. On AL00053335 (7.43% sequence identity), PPalign suffers from its tendency to align too many positions: like HHalign it correctly aligns half of the target alignment, but it proposes a longer alignment than HHalign, making its precision drop to around 40% when HHalign stays around 60%. The two other pairs are AL00050021 and AL00052441 with respectively 14.61% and 15.38% sequence identity allowing HHalign to correctly align 60% of the target alignment. On AL00052441, PPalign correctly aligns more than 50% of the target alignment but the main difference comes here again from the precision (0.58 vs 0.81). Results on AL00050021 are clearly in favour of HHalign with an F_1 score of 0.6 compared to 0.4 for PPalign and can be explained by the extremely gappy MSAs used to build the models (more than $\frac{1}{3}$ positions in the reference alignment were trimmed).

Interestingly, PPalign without coupling score (PPalign-1D) achieves an F_1 score comparable to HHalign (0.580 vs 0.578) despite a poor handling of gaps by Potts models as opposed to pHMMs. Besides, while PPalign's alignment is most of the time better with the coupling score, 2 sequence pairs were yet significantly better aligned by PPalign-1D than by PPalign with couplings: on already discussed AL10050875, where it improves a bit the poor quality of the alignment by PPalign,

but also on AL00089447 (12.93% sequence identity) where it improves over the improvement of HHalign on PPalign.

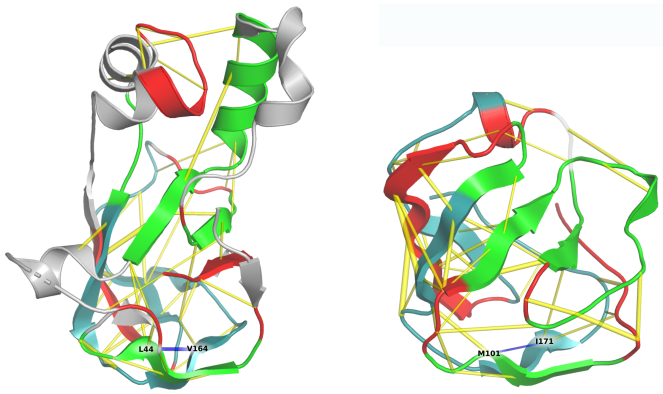


(a) Alignments. Alignment R is the reference alignment from SISYPHUS. Alignment P, obtained by PPalign-1D, and Alignment C, obtained by PPalign (with positional and coupling scores), are shown using green color for properly aligned positions and red color for misaligned positions with respect to Alignment R. It can be seen that alignment C improves over alignement P by aligning properly 31 new positions in addition to the 30 positions properly aligned in P. Since it still misaligns 28 positions with respect to the 89 positions to be aligned in R, precision and a recall are then both equal to 0.69. Alignment by HHalign, not shown here, aligns only 17 positions (the segments `V152-A168` and `Q159-R175` near the right-ends of the sequences) which are all correct, resulting on a precision of 1, but with a recall of 0.19.



(b) Positional and coupling scores of aligned positions for P and C. At each aligned position (i,k), the `v` row shows $s_v(v_i^A, v_k^B)$ while `w` row shows the sum of coupling similarities $s_w(w_{ij}^A, w_{kl}^B)$ between (i,k) and the other aligned positions (j,l), A and B denoting the Potts models inferred for sequences 1o56A_12_173 and 1pklA_88_180. Coupling scores were not used to find alignment P, but if we compute them on this alignment we can see many negative coupling scores. Introducing coupling scores in the optimization enables to find a better alignment C with lower positional similarities compensated by higher coupling similarities. The maximum positive contribution of couplings is on aligned positions 164 and 171, mainly due to a high similarity of $w_{44,164}^A$ with $w_{101,171}^B$ that makes positions 44 and 101 be the second highest coupling score contribution among aligned positions and helps aligning them properly in C.

Figure 5.8 – Illustration of the contribution of couplings for the alignment of 1o65A_12_173 and 1pklA_88_180 sequences.



(c) Visualisation a posteriori on pdb structures. Positions correctly aligned by PPalign-1D and PPalign are in deepteal, new positions correctly aligned by PPalign are in green, misaligned positions by PPalign are in red and correctly unaligned positions are in grey. The top 50 position pairs (i, j) with highest $\|w_{ij}\|$ are linked by yellow sticks, except $(44, 164)^A$ and $(101, 171)^B$ colored in blue. Although these pairs do not have not the strongest norm, they are those with the highest similarity helping to anchor correctly the alignment on L44 and M101 beta strands.

Figure 5.8 – Illustration of the contribution of couplings for the alignment of 1o65A_12_173 and 1pklA_88_180 sequences.

5.6.2.5 Discussion

Although the problem is assumed to be NP-hard, these experiments demonstrate that PPalign yields optimal Potts to Potts alignments up to a precision ϵ in tractable time. These results have to be confirmed on bigger instances. For now, experimentation is limited by memory handling in CCMpredPy, which is currently the only inference method offering the features we require to infer comparable Potts models, but the current implementation of CCMpred [SGS14] shows that this type of inference can be optimized to handle significantly larger models. This should enable us to test larger alignments in the future. Based on our experimentation, we expect these alignments to be also tractable. This is surprising with respect to the NP-complete nature of the problem, but it seems that alignments of Potts models are not the hardest instances when they properly represent homologous proteins. We think that this depends yet on the choice of the parameters shaping the inference of Potts models and the similarity of the models to align: these questions deserve further studies to better understand the application scope of

this method.

Regarding alignment quality, our results for the alignment of Potts models inferred using a pseudo-likelihood method designed for co-evolution prediction purposes are overall better than for the alignment of pHMMs by HHalign, with significant examples demonstrating how taking couplings into account can improve the alignment of remote homologous proteins, especially for lowest similarity alignments. There is still room for improvement in our method. We have noticed a tendency to align too many positions that can be corrected and our worst score with respect to HHalign is associated with very gappy train MSAs, indicating that augmenting Potts models with an appropriate gap handling strategy would undoubtedly improve our results.

Above all, it is worth noting that PPalign-1D finds sometimes a better alignment than PPalign, coupling matrices bringing more noise than assistance in these cases.

5.7 Conclusion

In this chapter, we introduced PPalign, a Potts model to Potts model alignment method. This method is based on an Integer Linear Programming formulation which simply extends a formulation for structure alignment with two-dimensional scoring schemes designed by Wohlers et al. by introducing a similarity score for both positions and pairs of positions. We based this similarity function on the scalar product to score pairwise matches of fields and couplings by taking into account both their similarity and their importance within their respective models. The optimal solution of this ILP formulation can be found efficiently using Wohlers et al.'s solver, yielding a solution within a chosen small epsilon range of the exact solution in tractable time.

We carried out preliminary experiments on sequence to model alignments which indicated that PPalign can align a sequence to a Potts model inferred on its close homologs in tractable time. This could be used to build multiple sequence alignments on which Potts models are inferred and thereby reduce biases caused by homology search methods based on positional conservation such as HHblits.

Besides these first experiments on sequence-model alignments, our main

contribution is the unprecedented possibility of aligning two Potts models representing sets of protein sequences. Before using it for homology detection purposes, we focused on its performances in terms of alignment quality. To this end, we extracted low sequence identity reference pairwise alignments from the manually curated structural alignments database SISYPHUS and compared PPalign's alignments of Potts models built from the sequences with the reference alignments. A training phase allowed us to select suitable hyperparameters – which we'll reuse in our homology detection experiments in the next chapter – and PPalign's performances on the test set were compared with its positional only version, and with its pHMM counterpart HHalign. On this selection of highly divergent sequence pairs, PPalign's alignments were better on average, PPalign's alignments without coupling score were comparable to HHalign's, and we showed that direct couplings could substantially improve some alignments with the lowest sequence identity. We argue that these results could be greatly improved with a suitable handling of insertions and deletions and, considering that taking the coupling score into account occasionally deteriorates alignment quality, we suspect that the additional sensitivity provided by an appropriate pseudo-count scheme on the double frequencies would further improve the alignment of such divergent proteins. Nevertheless, results with current Potts models built using our workflow indicate that our alignment method can yield better alignments of remote homologs than existing methods based on positional conservation only, and therefore should improve homology detection as well.

Chapter 6

First experiments on homology detection

In this thesis, we proposed to use Potts models for homology search purposes. To this end, we addressed the question of building canonical Potts models to represent proteins and designed a pairwise Potts model alignment method, PPalign, which was compared to the state-of-the-art pairwise pHMM alignment method HHalign, showing its potential to find better alignments on low identity remote homologs. We acknowledged that our method is inherently limited by the comparability of the Potts models to be aligned and their ability to represent proteins in a comparable way, and we argue that effort should be placed towards more comparable Potts models before performing extensive homology detection experiments and develop our homology search suite further. Nonetheless, in this final chapter we report results of our method on preliminary experiments to give some idea of PPalign's current performances in homology detection. The first section reports early experiments at the family level and the second section reports latest experiments at the fold level.

6.1 Early experiments at the family level

This section describes results of three homology detection experiments at the family level that were carried out early in this thesis, before the design of a

coupling parameter smoothing strategy, before the introduction of a comparison with respect to background and offset in the score function, and before PPalign's hyperparameters training on reference alignments. Since these results were already good, we did not run these experiments again with these improvements.

The first section describes our method for these experiments, then we describe each data set and report the corresponding results, and finally discuss these results.

6.1.1 Methods

In these experiments, each data set consists of a set of positive examples, which are sequences annotated as members of the family considered, and a set of negative examples – in the case of thioredoxin families, negative examples for each family are the members of the other families.

For each example, we built a Potts model by retrieving close homologs using HHblits as described in section 4.3 on UniClust30 (08/2018), filtering at 80% identity, removing columns with more than 20% gaps, taking the first 1000 sequences, and then we inferred Potts models with CCMpredPy using default parameters except for the number of pseudo-counts which we set to 1000 (which corresponds to a single pseudo-count rate $\tau_v = 0.5$). Parameters were not smoothed after inference.

Each positive example was aligned with every other example in the dataset by running PPalign using the pure scalar product similarity function (without comparison to background), with $\alpha_w = 1$, without offset and with gap open and extend penalties arbitrarily set to respectively 8 and 0.

To score the similarities between sequence pairs, we considered the normalized alignment score:

$$S_{norm}(A, B) = \frac{2S(A, B)}{S(A, A) + S(B, B)} \tag{5.30}$$

We also ran PPalign without couplings (PPalign-1D) to examine couplings contribution.

To compare our results, we also considered E-values yielded by:

- HHalign v3.0.3 aligning pHMMs trained on the train MSAs
- BLASTp v2.9.0+ aligning the original sequences

both with default parameters.

6.1.2 Kunitz family

The Kunitz (or pancreatic trypsin inhibitor) family is a family of serine proteinase inhibitors with a relatively small active domain of about 50 amino acids (see logo figure 6.1).

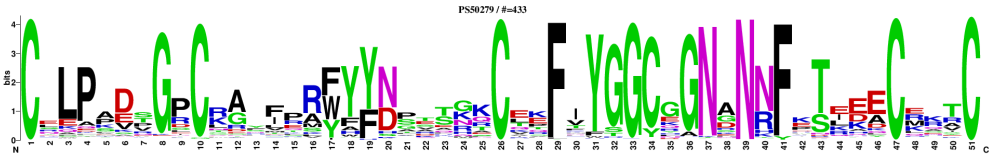


Figure 6.1 – Sequence logo for the Pancreatic trypsin inhibitor (Kunitz) family profile (PS50279) built by PROSITE [PROb]

PROSITE database [Sig+12] provides a pattern signature (*PS00280*) and a profile matrix (*PS50279*) for this family, along with corresponding manually curated true positive, false positive and false negative sequences, which we used to generate a dataset of positive and negative examples to assess whether our similarity score could discriminate members from non-members of the Kunitz family.

Our dataset was build with the following workflow:

1. An initial set of positive examples (resp. negative examples) was generated with sequences annotated as true positive and false negative (resp. false positive) in PROSITE (release prosite2019_08) entries *PS50279* and *PS00280*

2. Entries whose descriptions contained "fragment", "uncharacterized", "probable", "putative", "like" or "inactive", or whose sequences contained X characters were removed

3. The set of negative examples was augmented by calling BLASTp v2.9.0+ on SwissProt (08/2019 release) on each positive example and filtering out sequences whose descriptions contained typical keywords describing the

Kunitz family ("kunitz", "inhibitor", "amyloid", "collagen", "toxin", "allergen", "boophilin", "lactation", "stephenin", "tigerin", "blackelin", "scutellin", "Mambaquaretin", "Papilin", "Ornithodorin", "anticoagulant")

4. Sequences with lengths > 200 were removed from the sets, for memory considerations with respect to CCMpredPy

5. Each set was made 30% non-redundant using MMseqs2 [SS17]

In the end, our dataset consisted in a total of 28 positive examples and 18 negative examples – each representing a 30% non-redundant cluster, which we recap in table 6.1.

	UniProt identifiers
positive examples	P0C1X2, P0DJ46, P25660, A5X2X1, P84875, P81547, P11424, W4VSH9, Q8WPI3, P56409, Q11101, A7X3V7, P0DJ76, P86959, Q8T3S7, Q29100, Q6UDR6, P00993, Q8T0W4, D3GGZ8, H2A0P0, P07481, Q9D263, P82968, O62845, Q589G4, D2Y488, P81162
negative examples	P40958, O67526, P08938, A0A075B6J1, B0SH16, P35578, P11589, P02755, P06910, B9L6N1, Q9NPH6, Q21D07, Q01584, Q1WUC5, B6JM17, Q46036, P84811, P06911

Table 6.1 – Data set for our experiment on the Kunitz family

Later, we realized that our workflow failed to model one sequence (*P82968*) properly. Indeed, though this protein contains four domains – three Kazal-like domains and one Kunitz domain – homologs retrieved by HHblits only cover the first three domains, consequently the region containing the Kunitz domain was trimmed out. To focus on PPalign's ability to compare Potts models given proper input data, we removed this sequence from the dataset.

For each positive example, we examined whether its normalized scores with other positive examples were higher than with negative examples by computing the *Area Under the Curve* (AUC), the area under the *ROC* (*Receiver Operating*

Characteristic) curve which plots the true positive rate against the false positive rate for several classification thresholds. Results are reported figure 6.2.

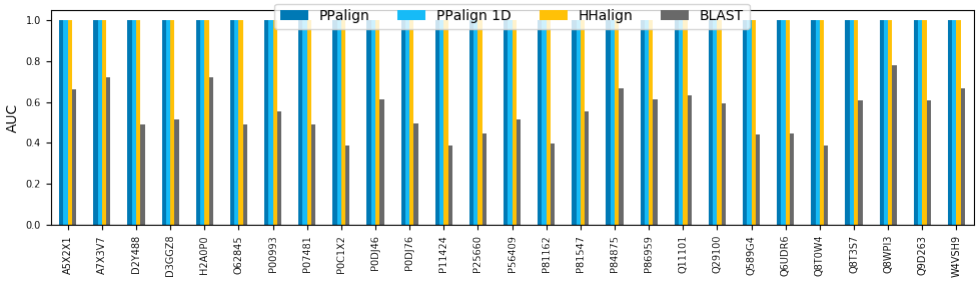


Figure 6.2 – AUC for each Kunitz family member in the data set

Except for BLAST, all AUCs are 1, indicating that, for each positive example, one can define a score threshold to fully discriminate positive examples from negative ones.

6.1.3 RR domain

The Response Regulatory domain (RR domain) is the conserved N-terminal domain of the response regulator proteins, involved in phosphotransfer schemes with histidine kinase [SRG00]. This domain is about 125 residues (see logo figure 6.3).

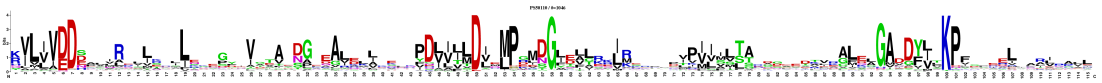


Figure 6.3 – Sequence logo for the RR domain profile (PS50110) built by PROSITE [PROa]

We performed the same experiment as for the Kunitz domain, starting with PROSITE entry *PS50110*, on a total of 13 positive and 5 negative 30% non-redundant examples (see table 6.2).

	UniProt identifiers
positive examples	A6UEL7, B0R4K1, C4Y489, F4JZT3, O69280, P33394, P37740, P46384, Q2YIF7, Q55169, Q6H468, Q7XN30, Q9ZWS6
negative examples	A0AK95, P37470, Q31GG2, Q5NRC4, Q9F7A2

Table 6.2 – Data set for our experiment on the RR domain

As for the previous experiments, AUCs of PPalign, PPalign-1D and HHalign were all 1.

6.1.4 Families of the thioredoxin fold

In this experiment, we focused on the protein families forming the thioredoxin fold, as described in [Mar95]: thioredoxin, glutaredoxin, glutathione S-transferase, DsbA and glutathione peroxidase, to examine whether our similarity score could discriminate members of one family from members of other families in the same fold.

For each family, we retrieved sequences from UniProt (2019_09 release) using the *family* annotation and `reviewed:yes` and then followed the same steps as in section 6.1.2 to build Potts models for representatives of 30% non-redundant clusters of sequences. In the end, our dataset consisted of 18 members of the thioredoxin family, 25 members of the glutaredoxin family, 4 members of the glutathione S-transferase family and 1 member of the glutathione peroxidase family (see table 6.3). There was no sequence of length ≤ 200 in the DsbA family.

	UniProt identifiers
thioredoxin	O51088, O64764, O81332, P23400, P25372, P42115, P52228, P66929, P97615, Q655X0, Q6XHI1, Q75GM1, Q7XKD0, Q851R5, Q8IFW4, Q98PL5, Q9CAS1, Q9SEU
glutaredoxin	P00276, P0AC64, P0AC72, P0C291, P10575, P17695, Q05926, Q0J3L4, Q0JQ97, Q5SMY5, Q6YFE4, Q76QK5, Q7XIZ1, Q84TF4, Q84Y95, Q851Y7, Q86SX6, Q8L9S3, Q8LBS4, Q923X4, Q96305, Q9FVX1, Q9LIF1, Q9SGP6, Q9Y7N3
glutathione S-transferase	O16115, O77462, P30102, P35661
glutathione peroxidase	O23970

Table 6.3 – Data set for our experiment on the thioredoxin families

AUC results for each method on each family are reported figure 6.4.

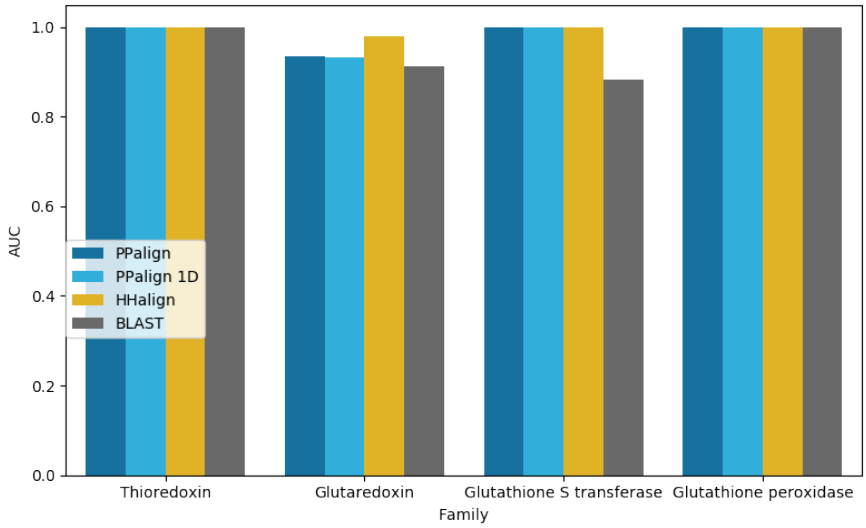


Figure 6.4 – AUCs for the four families of the thioredoxin fold considered

We identified that only two proteins prevented all methods from achieving AUCs of 1: *P00276* and *Q76QK5*.

P00276 is a special case: it was assigned to the glutaredoxin family, but since it can be reduced by thioredoxin reductase, it is also a member of the thioredoxin family [NGE93]. All methods yield high similarity scores (or low E-values) for the alignment of *P00276* with both thioredoxins and glutaredoxins.

As for *Q76QK5*, though it does belong to the glutaredoxin family, it appears that the training set of 1000 effective sequences used to train its Potts model and HHalign's pHMM contains very few sequences of the glutaredoxin family: of all annotated sequences in this training set, only 86 are annotated as members of the glutaredoxin family and 766 are annotated as members of the thioredoxin family.

These two special cases aside, all methods yield AUCs of 1 for each family in the data set.

6.1.5 Conclusion

In these experiments, when HHblits provided suitable multiple sequence alignments of close homologs, our method was able to fully discriminate positive examples from negative examples. A few examples demonstrated that our results are heavily dependent on the multiple sequence alignments used to train our Potts models, especially on HHblits' output. These results were also achieved by our method with the positional score alone, matching the performance of HHalign. This suggests that our similarity score is appropriate to detect members of considered families. However, the contribution of couplings could not be shown in these experiments. Indeed, at the family level it is difficult to find sets of sequences hard to characterize with annotated negative examples. Families are often defined using PROSITE motifs or profile Hidden Markov Models, thus their definition is biased towards positional conservation.

6.2 Homology detection at the fold level with time constraint

At the end of this thesis, we performed a quick experiment to examine the current performances of our method with optimized hyperparameters and the current workflow for Potts model construction on the detection of more remote homologs,

at the fold level. We focused on folds of the all-beta class, since long-range interactions were shown to predominate in this class [GS97]. Since the time we had left to carry out these experiments was limited, we took it as an opportunity to assess our method's performances with a stringent time constraint.

6.2.1 Data

Our dataset was extracted from the structural classification database SCOPe [Mur+95; FBC14] (version 2.07-stable), focusing on the class of "All beta proteins" (sunid 48724). First, we selected each fold with at least two superfamilies, for a total of 228 superfamilies in 36 different folds. Then, for each superfamily, we selected the domain with the smallest length and with at least 1000 effective (80% sequence identity) close homologs retrieved by HHblits, limiting the maximum length to 200 amino acids. Since such a domain did not exist in all superfamilies and since we require at least two domains per fold, in the end our dataset consisted of 116 domains in 23 different folds, which we recap in table 6.4. A Potts model was inferred for each domain following the workflow described in section 4.7.

fold	**superfamilies**	**domains**
Immunoglobulin-like beta-sandwich (48725)	Immunoglobulin (48726)	d3tvme2
	Fibronectin type III (49265)	d1q38a1
	PKD domain (49299)	d1b4ra_
	beta-Galactosidase/glucuronidase domain (49303)	d2vl4b2
	Cadherin-like (49313)	d1zvna1
	Cu,Zn superoxide dismutase-like (49329)	d1do5c_
	CBD9-like (49344)	d1i8ua_
	PapD-like (49354)	d1m1sa1
	Purple acid phosphatase, N-terminal domain (49363)	d4dsyb1
	Superoxide reductase-like (49367)	d1vzib1
	Invasin/intimin cell-adhesion fragments (49373)	d1f02i2
	Integrin domains (69179)	d5neua2

	Lamin A/C globular tail domain (74853)	d3hn9c_
	Thiol:disulfide interchange protein DsbD, N-terminal domain (DsbD-alpha) (74863)	d1se1a3
	E set domains (81296)	d2d7na1
	ApaG-like (110069)	d2f1ea_
	LEA14-like (117070)	d1xo8a_
	ICP-like (141066)	d2nnrb_
	CalX-like (141072)	d3eadc_
	Macroglobulin (254121)	d3cu7a6
	Zn aminopeptidase insert domain (254133)	d1z5hb3
	Fibronectin III-like (254143)	d2x42a3
Common fold of diphtheria toxin/transcription factors/cytochrome f (49379)	Carbohydrate-binding domain (49384)	d2xbda_
	Bacterial adhesins (49401)	d1klfh2
	beta-sandwich domain of Sec23/24 (81995)	d1m2va2
	DR1885-like metal-binding protein (110087)	d2jqaa1
Prealbumin-like (49451)	Starch-binding domain-like (49452)	d2xhna2
	Carboxypeptidase regulatory domain-like (49464)	d1h8la1
	Transthyretin (synonym: prealbumin) (49472)	d3qvac_
	Cna protein B-type domain (49478)	d1vlft1
	Aromatic compound dioxygenase (49482)	d3pcdd_
	Hypothetical protein PA1324 (117074)	d1xpna1
C2 domain-like (49561)	C2 domain (Calcium/lipid-binding domain, CaLB) (49562)	d2fjub2
	Periplasmic chaperone C-domain (49584)	d4djma2
	PHL pollen allergen (49590)	d1bmwa_
SH3-like barrel (50036)	C-terminal domain of transcriptional repressors (50037)	d2e64b1
	SH3-domain (50044)	d2a28d1

	Myosin S1 fragment, N-terminal domain (50084)	d2mysa1
	Electron transport accessory proteins (50090)	d4zgjn_
	Translation proteins SH3-like domain (50104)	d1nppa2
	Cell growth inhibitor/plasmid maintenance toxic component (50118)	d1vubd_
	Fumarylacetoacetate hydrolase, FAH, N-terminal domain (63433)	d1qqjb1
	Tudor/PWWP/MBT (63748)	d2lvma1
	Cap-Gly domain (74924)	d2e3ha1
	Prokaryotic SH3-related domain (82057)	d2hbwa1
	BAH domain (82061)	d3ptaa3
	Chromo domain-like (54160)	d3deoa1
	PAZ domain (101690)	d1si2a1
	YccV-like (141255)	d1vbva1
	CarD-like (141259)	d2eyqa1
GroES-like (50128)	GroES-like (50129)	d3nx6a_
	SacY-like RNA-binding domain (50151)	d1l1ca_
Sm-like fold (50181)	Sm-like ribonucleoproteins (50182)	d4emhe_
	YhbC-like, C-terminal domain (74942)	d1ib8a1
OB-fold (50198)	Staphylococcal nuclease (50199)	d1rkna_
	TIMP-like (50242)	d3ckib_
	Nucleic acid-binding proteins (50249)	d1uebb3
	Inorganic pyrophosphatase (50324)	d1wcfa1
	MOP-like (50331)	d1v43a1
	CheW-like (50341)	d2ch4b1
	gp5 N-terminal domain-like (69255)	d2p5zx1
	Heme chaperone CcmE (82093)	d1j6qa_
	Hypothetical protein YgiW (101756)	d1nnxa_
	NfeD domain-like (141322)	d2exda1
	HupF/HypC-like (159127)	d3vyua_
	EutN/CcmL-like (159133)	d5l37b_

	Proteasome regulatory subunits PAN/Rpt, OB-fold domain (345919)	d3h43a2
beta-Trefoil (50352)	Ricin B-like lectins (50370)	d1tfmb2
	Actin-crosslinking proteins (50405)	d3llpa4
	MIR domain (82109)	d3qr5a_
	AbfB domain (110221)	d1wd3a2
Reductase / isomerase / elongation factor common domain (50412)	FucI/AraA C-terminal domain-like (50443)	d4r1qc2
	Translation proteins (50447)	d3mqkc_
	Riboflavin synthase domain-like (63380)	d1kzla1
	Riboflavin kinase-like (82114)	d3op1c2
Elongation factor / aminomethyltransferase common domain (50464)	EF-Tu/eEF-1alpha/eIF2-gamma C-terminal domain (50465)	d1r5ba2
	Aminomethyltransferase beta-barrel domain (101790)	d1yx2a2
Split barrel-like (50474)	FMN-binding split barrel (50475)	d3a20b_
	PilZ domain-like (141371)	d3kygb2
Domain of alpha and beta subunits of F1 ATP synthase-like (50614)	N-terminal domain of alpha and beta (or A/B) subunits of rotary ATPases (50615)	d3tgwb1
	Alanine racemase C-terminal domain-like (50621)	d1rcqa1
	Aminopeptidase/glucanase lid domain (101821)	d1ylob1
Double psi beta-barrel (50684)	Barwin-like endoglucanases (50685)	d4jp6a_
	ADC-like (50692)	d4rv0e1
Streptavidin-like (50875)	D-aminopeptidase, middle and C-terminal domains (50886)	d1ei5a2
	YceI-like (101874)	d3hpeb_
	YdhA-like (141488)	d2f09a1
WW domain-like (51044)	WW domain (51045)	d1zr7a1
	Carbohydrate binding domain (51055)	d1ed7a_

Single-stranded right-handed beta-helix (51125)	Cell-division inhibitor MinC, C-terminal domain (63848)	d1hf2c1
	beta-Roll (51120)	d1o0ta1
	Pentapeptide repeat-like (141571)	d2j8ia1
Single-stranded left-handed beta-helix (51160)	Trimeric LpxA-like enzymes (51161)	d1fxja1
	Adhesin YadA, collagen-binding domain (101967)	d1p9ha_
	Guanosine diphospho-D-mannose pyrophosphorylase/mannose-6-phosphate isomerase linker domain (159283)	d2cu2a1
Double-stranded beta-helix (51181)	RmlC-like cupins (51182)	d2k9za_
	Clavaminate synthase-like (51197)	d1s4cb_
	cAMP-binding domain-like (51206)	d2zcwa2
	Regulatory protein AraC (51215)	d1xjac_
	TRAP-like (51219)	d1pg6a_
	Thiamin pyrophosphokinase, substrate-binding domain (63862)	d1ig3b1
	Metal cation-transporting ATPase, actuator domain A (81653)	d5avva2
Barrel-sandwich hybrid (51229)	Single hybrid motif (51230)	d2l5ta_
	Rudiment single hybrid motif (51246)	d1vf5c2
	Duplicated hybrid motif (51261)	d2gpra_
	Ribosomal L27 protein-like (110324)	d2nn6h2
	V1 ATP synthase A subunit, bulge domain-like (310577)	d3gqba3
beta-clip (51268)	AFP III-like domain (51269)	d4ur6b_
	Urease, beta-subunit (51278)	d1ejxb_
	dUTPase-like (51283)	d2d4la_
	MoeA C-terminal domain-like (63867)	d1wu2a1
	SET domain (82199)	d3kmja_

Nucleoplasmin-like/VP (viral coat and capsid proteins) (88632)	PHM/PNGase F (49742)	d3miba1
	Positive stranded ssRNA viruses (88633)	d5xs4b_
Double-split beta-barrel (89446)	AbrB/MazE/MraZ-like (89447)	d2mrua1
	AF2212/PG0164-like (141694)	d2d9ra1

Table 6.4 – Domains in the all-beta class considered for the homology detection experiment at the fold level

6.2.2 Experiment

We aligned every domain pair using PPalign with hyperparameters previously trained in the alignment quality experiment described in chapter 5, i.e. with $\alpha_w = 6$, a gap open penalty of 13, a gap extend penalty of 0, an offset of 1.0, and using the similarity score with comparison to background. To assess how reliable our results are with a stringent time constraint, we set a 1 minute time out for each of the 6670 domain pairs to be aligned.

In this experiment, to compute the similarity between two domains we use PPalign's normalized similarity score:

$$s_{norm}(A, B) = \frac{2s(A, B)}{s(A, A) + s(B, B)} \tag{6.1}$$

where $s(A, B) = s_v(A, B) + \alpha_w s_w(A, B)$. In other words, once Potts models are aligned, we focus on the similarity of parameters at aligned positions, disregarding other components of the scoring function that were used to provide the best alignment (i.e. gap cost and offset).

As before, we ran PPalign-1D, HHalign v3.0.3, BLAST, and also MRFalign v0.90, with default hyperparameters. Models of the latter were not built on the same MSAs as PPalign or HHalign since the software only allow us to input a single sequence, from which it builds a Markov Random Field in a rather opaque fashion (see section 3.2.4).

6.2.3 Results

We computed AUCs for each method and each fold. On average, PPalign yields the best AUC (0.809), closely followed by PPalign-1D (0.800). Both PPalign methods achieve better AUCs on average than all other considered methods in this experiment (0.765 for MRFalign, 0.732 for HHalign and 0.683 for BLAST).

Results for each individual fold are reported figure 6.5. As we can see in this figure, performances differ widely from fold to fold.

Though all methods outperform BLAST in most cases, folds 51125 (Single-stranded right-handed beta-helix) and 51160 (Single-stranded left-handed beta-helix) are significantly better recognized by BLAST.

PPalign significantly outperforms (with a difference in AUCs of at least 0.1) MRFalign in 6 folds – while being significantly outperformed by the latter in 2 folds – and HHalign in 6 folds – while being significantly outperformed by the latter in 2 folds as well.

Both PPalign methods remarkably outperform other methods on the recognition of fold 51044 (WW domain-like) with an AUC of 0.956 for PPalign and 0.998 for PPalign-1D, while MRFalign and HHalign only achieve AUCs of respectively 0.706 and 0.560. This may be related to the fact that considered domains in this fold are exceptionally small (resp. 28 and 45 amino acids) and since PPalign's computation time depends on the lengths of the sequences to be aligned, all alignments with members of this fold were solved to optimality within the 1 minute time out, as opposed to other folds, as we will see below.

One of the two folds where PPalign is significantly outperformed by HHalign is fold 50474 (Split barrel-like) where PPalign achieves an AUC of 0.645 versus 0.770 for HHalign, a performance matched by PPalign-1D. This is the only fold where PPalign-1D significantly outperforms PPalign. Conversely, PPalign achieves a significantly better AUC than PPalign-1D in one fold: 50614 (Domain of alpha and beta subunits of F1 ATP synthase-like) with an AUC of 0.892 versus 0.767.

These results were obtained with a 1 minute timeout only and some alignments were actually far from convergence. As shown figure 6.6, our AUCs are highly related to the precision of the solution yielded by the solver after one minute, indicating that in cases where our performances were the poorest, the solution

was actually probably far from the actual maximum of our scoring function. This precision is in turn, unsurprisingly, related to the lengths of the considered domains (see figure 6.7).

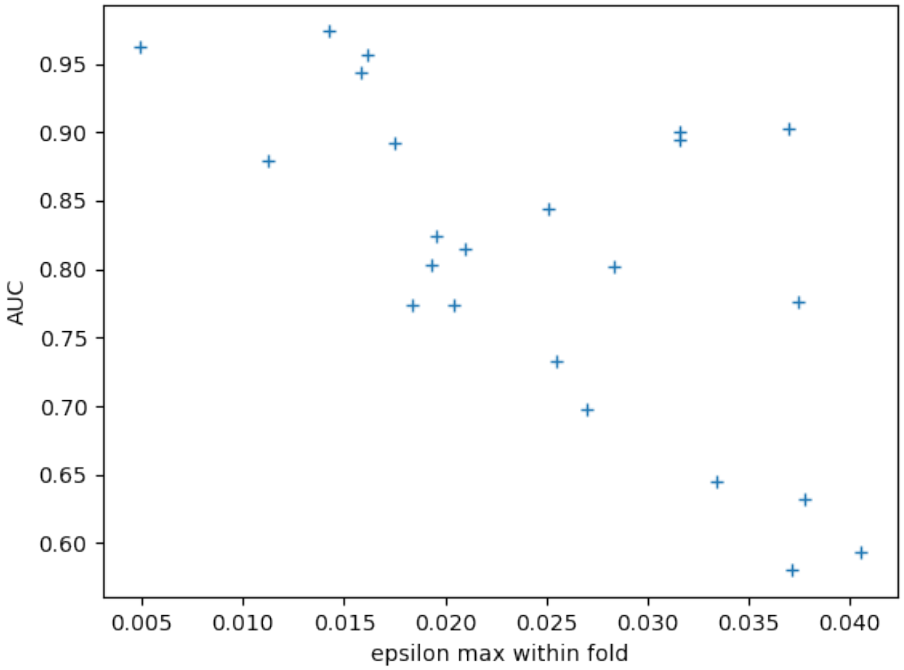


Figure 6.6 – AUC yielded by PPalign for each fold against the maximum precision epsilon of the solution yielded by the solver after a 1 minute time out for the alignments of positive examples in the fold. Except for a few folds with rather good AUCs despite a large epsilon, our AUCs are clearly linked with the precision of the yielded solution with respect to the optimal solution.

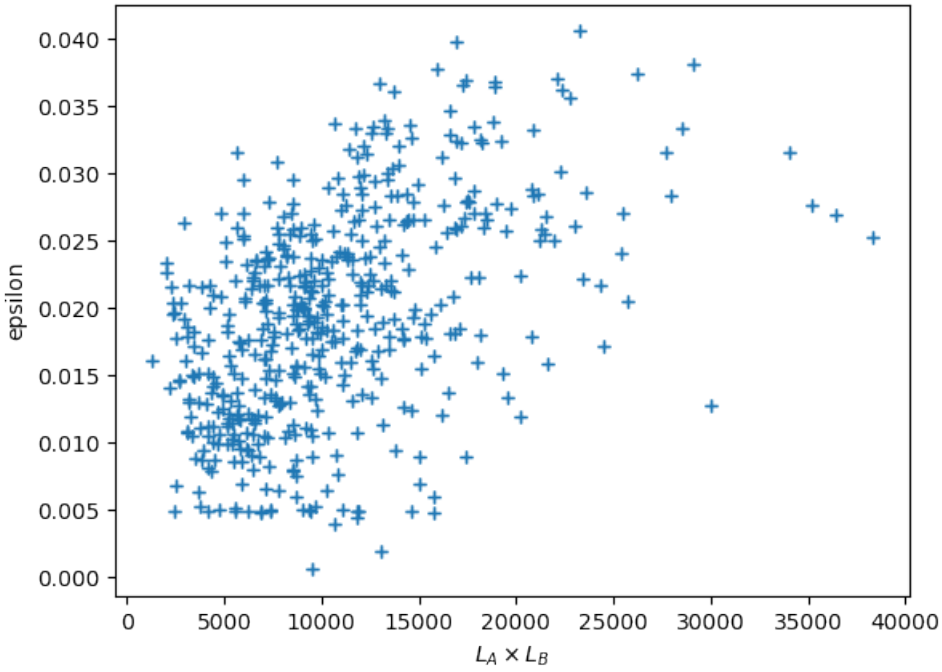


Figure 6.7 – Precision epsilon after one minute time out for alignments of domains within the same fold with respect to the product of their lengths. As shown in this figure, convergence seems to be slower as the size of the models increases.

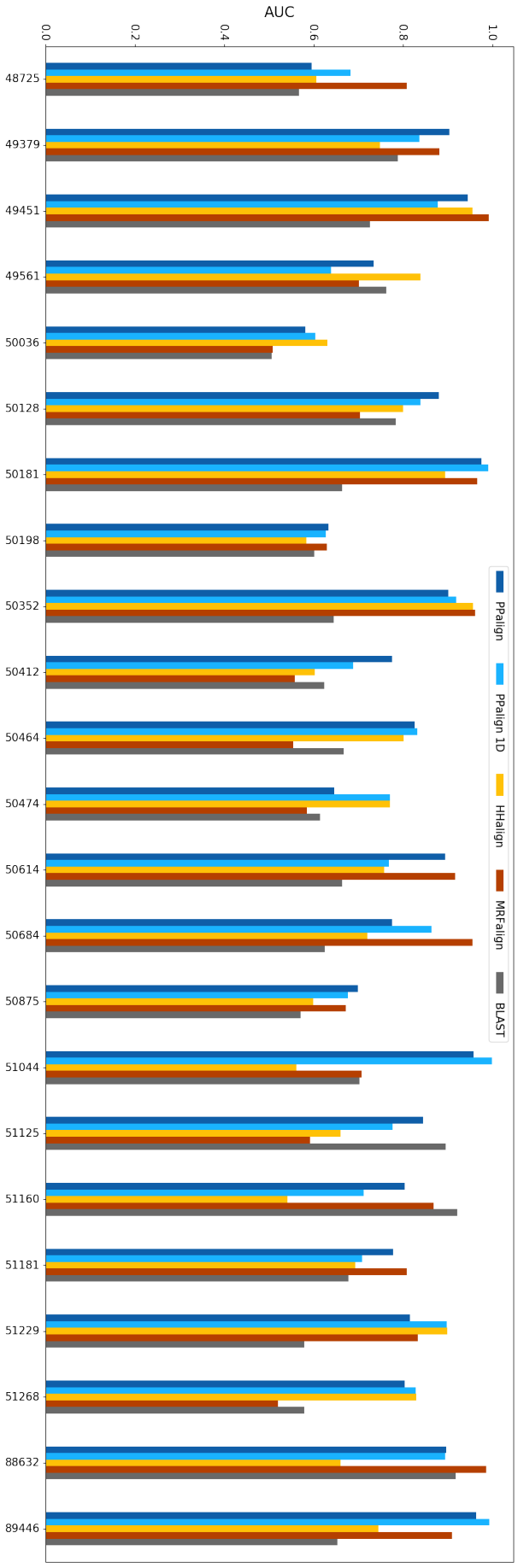


Figure 6.5 – AUCs for each considered fold.

6.2.4 Conclusion

In this section, we reported results of a first experiment carried out at the end of this PhD thesis to assess the ability of our method – with the current workflow – to distinguish members of a fold from non-members with a stringent 1 minute time constraint on the smallest domains of the all-beta class.

In this experiment, our method yielded very encouraging results, with on average better AUCs than our main competitor MRFalign. Our AUCs were also better on average than HHalign though, unlike with MRFalign where MRFs were built using its own workflow, pHMMs were built on the same input data as PPalign, which is not the data used by HHsearch in practice, and a different experiment should be carried out to properly compare our results with HHsearch by looking at its retrieved sequences and their E-values.

While these results are already very encouraging, we also showed that our performances were highly linked with the precision of the solution yielded by the solver after the one minute time out, demonstrating that our poorest results were associated with solutions that were actually probably very different from the actual optimal solution of our alignment scoring function. This suggests that better results would likely be achieved with a higher computation time, especially for folds with longer domains. This would also allow us to properly assess the contribution of couplings.

6.3 Conclusion

In this chapter, we performed first experiments to assess the performances of our pairwise Potts model comparison method in homology detection with our current Potts model construction workflow.

Early experiments at the family level suggested that our similarity score is appropriate to detect members of families defined by a strong positional conservation. In these experiments, the positional score was sufficient, and the contribution of couplings could not be shown at this level, since we did not manage to find an appropriate benchmark for a family hard to characterize with annotated negative examples.

At the end of this thesis, we carried out an initial experiment at the fold level on SCOPe's all-beta class' smallest domains with a very limited time constraint. This experiment yielded very encouraging results: on average we outperformed our main competitor, MRFalign, and we also outperformed HHalign though we used it on pHMMs trained on the same data as us, which is different from HHsearch's approach. This outperformance was achieved both by PPalign and PPalign-1D, confirming the relevance of our scalar product based similarity function. Couplings contribution remained unclear at this point, only significantly improving the recognition of one fold, and sometimes leading to poorer results though it led to slightly better AUCs on average, but we highlighted a clear relation between our AUCs and the precision of the solution yielded after one minute, indicating that our poorest results correspond to alignments that were probably very different from the actual optimal of our scoring function. To improve further these results and properly assess the contribution of direct couplings, we will need to perform experiments allowing for a longer computation time in the future, especially for longer domains. Beforehand, we need to improve the quality of our Potts models. While experiments on alignment quality reported in the previous chapter suggested that better results would be obtained with a better gap handling and more sensitive pairwise coupling parameters, experiments reported in this chapter brought to light our dependence on the multiple sequence alignment of close homologs on which models are inferred. Improving the quality of this input data will undoubtedly improve the quality of the Potts models and thereby improve our homology detection performances. Then, we will still have to provide normalized statistics such as E-values so that our method can be used for homology search purposes in practice.

Conclusion and perspectives

Conclusion

In this thesis, we showed that Potts models introduced by Direct Coupling Analysis are relevant candidates to model proteins, align them and compute their similarity in the context of homology search. We based this proposition on their ability to model both positional conservation – as profile Hidden Markov Models – and pairwise dependencies, enabling to reflect structural constraints, and overall on their very definition grounded on the maximum entropy principle which guarantees that they consistently generate observed statistics with as little bias as possible. This principle makes their pairwise potentials reliable predictors of directly co-evolving positions and provides interpretability, as opposed to the Markov Random Field previously proposed by MRFalign. While the Potts model's relevance for homology search purposes was also simultaneously identified by colleagues who designed heuristics to align sequences to Potts models or hybrid models combining profile Hidden Markov Models and Potts models, we were the first to propose a Potts model to Potts model alignment method, to guarantee optimality of the alignment and to show experimental evidence of the relevance of couplings. Due to the presence of non-local dependencies, the alignment problem is assumed to be NP-hard and cannot be efficiently computed with dynamic programming. Here, we made the choice not to rely on heuristics to limit biases in our investigations. We introduced an Integer Linear Programming formulation for the alignment problem, whose objective is to maximize a similarity score based on the scalar product to naturally extend the sequence-to-model score. This formulation extends simply the ILP formulation for alignments with two-dimensional scoring schemes in [Woh12]

with positional similarity scores and can be solved efficiently with the solver they developed. We implemented our method, named PPalign, in a software package embedding tools to build Potts models with our proposed workflow and visualize their parameters. Our experiments on the benchmark extracted from SISYPHUS showed that, with our parameters, a solution within a chosen ϵ of the optimal solution can be found in less than two minutes on average for Potts models representing homologous proteins. Besides the alignment method itself, a reflection on how to best represent proteins with Potts models so that they can be properly compared was central to this thesis. One of our contributions is the introduction of an initial workflow to build Potts models for pairwise comparison purposes. We based this workflow on CCMpredPy, a pseudo-likelihood inference method derived from CCMpred, which is considered as state-of-the-art for contact prediction, with additional features such as prior centered at an independent-site model, which we established as a first principle towards canonicity. We identified several obstacles to a sensitive Potts model comparison and implemented operational solutions. We validated PPalign as an alignment method on the benchmark of reference alignments with low sequence identity, on which the quality of our alignments was on average better than HHalign with models built on the same data and we showed that direct couplings were able to significantly improve the quality of some of the lowest sequence identity alignments. This suggests that Potts models might improve the detection of remote homologs, and encouraging preliminary results reported in the last chapter appear to concur with this statement.

To sum up, with this thesis we provided ground work for the development of an homology search framework based on pairwise Potts model alignment and for further investigations on the strengths and weaknesses of Potts models in homology detection tasks. Our first experiments suggest that Potts models might improve the detection of more remote homologs, but that further work should be conducted on the construction of more comparable models to better represent proteins and perform more sensitive comparisons. Hopefully, our method's guaranteed optimality will be a powerful asset to perform unbiased investigations in this direction.

Perspectives

Potts model construction

From sequence to MSA

As argued throughout this manuscript, our results depend as much on the method to align Potts models as on the way they are built.

The first step to build a Potts model from a sequence is to retrieve its close homologs. In this thesis, following initial recommendations for CCMpred, we used HHblits on UniClust. Alternatives specially designed for contact prediction and fold recognition purposes have been proposed since [BJ18; Ovc+17; Wan+19], the most recent one being DeepMSA [Zha+20], which proposes a hybrid pipeline based on HHblits and Jackhmmer/HMMsearch collecting sequences from both whole genome and metagenome sequence databases, and was shown to improve accuracy in long-range contact prediction with CCMpred by up to 24.4% with respect to their recommended worfklow, as well as structure prediction and protein threading. Since these approaches can build deeper MSAs, this would also enable us to model proteins that cannot be modeled with our current workflow due to a too small number of effective homologs retrieved by HHblits.

Not all retrieved homologs are equally relevant in the modeling of the target protein. As pointed out before, if too distant sequences are included in the input MSA, the protein is misrepresented, and thus can be misclassified. So far, we decided to include only the first 1000 retrieved effective sequences in order to take as little sequences as possible to avoid such bias while allowing for a minimum coupling prediction accuracy. This rather rough approximation would have to be refined to improve our performances on homology detection. A first step would be to decide on an appropriate E-value threshold, which would probably depend on the task – fold detection would probably require deeper alignments than family detection for instance. Furthermore, rather than defining a cut-off threshold under which all sequences will equally contribute to the model, a different reweighting scheme could be applied so that closest homologs of the protein to be modeled have more weight so that information from more remote homologs can still be used. A related idea was proposed in [MB19], where subfamilies of interest were

modeled along with other subfamilies with a lesser weight.

Though good results can be achieved with multiple sequence alignments built using positional conservation only, it should be noted that this constitutes a bias, and taking pairwise dependencies into account while building the MSAs might improve protein representation. A workaround would be to build a first Potts model on a seed alignment and re-align subsequent sequences with PPalign.

From MSA to Potts model

Results reported in this thesis were achieved with models built with CCMpredPy. Despite its valuable features, we identified two major shortcomings in using CCMpredPy for our goals.

The first issue is the gap handling strategy in CCMpredPy. While treating gaps as missing information is probably appropriate when it comes to modeling multi-domain proteins to avoid spurious couplings, when focusing on domains it might be more relevant not to disregard this information and instead treat the gap symbol as the 21^{st} letter as in most DCA methods. This could be easily implemented in CCMpredPy.

More critical issues raised from the pseudo-likelihood approach itself. While this approach is considered state-of-the-art in contact prediction by providing accurate results in limited computational complexity, their reliance on the full sequences rather than on MSA frequencies make it impossible to implement pseudo-count schemes, while the relevance of pseudo-counts in homology detection is long-established [HH96b]. Using an approach based on frequencies such as Boltzmann Machine Learning [FBW18] would allow us to use advanced pseudo-count strategies such as pseudo-counts based on Dirichlet mixtures used in HMMER or context-specific pseudo-counts used in HHsuite, and would enable us to introduce pseudo-counts on the double frequencies as well. This would significantly improve the comparability of inferred pairwise coupling parameters by reducing observed spurious anti-correlations arising from a lack of data. One major drawback, however, is that more accurate methods based on frequencies are also significantly slower than methods based on pseudo-likelihood.

Other options to be tested include regularization strategies other than the L_2

regularization. To start with, we could simply try with standard L_1 regularization or block L_1 regularization as implemented in [KOB13] to infer sparser models – or instead rely on more recent methods to iteratively decimate less significant couplings [Bar+20]. This would reduce parameter overfitting while favoring more interpretable couplings, and might speed up PPalign alignments since our computation times seem to depend on the number of couplings considered. However, though we think this is worth a try, we suspect that sparsity might not be relevant in our case. Indeed, it is possible that couplings with the highest norms for a given set of close homologs do not reflect most important couplings conserved at the family (or fold) level (see figures 6.10 and 6.9 for examples), and making models sparse would remove valuable information which would have enabled us to still compare the models by considering the nature of the interactions, reflected by the composition of the coupling matrices, besides their strength.

Regarding inference options in general, in this work we used CCMpredPy with default hyperparameters, optimized for contact prediction purposes. To improve our results, inference hyperparameters such as regularization coefficients should be trained as well for alignment and homology detection purposes.

Finally, in this thesis we focused on the alignment of standard Potts models, where gaps are modeled at most with an additional letter. However, as evidenced by superior performances of profile Hidden Markov Models over simple profiles, an appropriate gap handling strategy would be relevant. We could use position-specific gap costs associated with an additional energy term as in [Fei+14b] and DCAlign [Mun+20], or consider the Hidden Potts Model architecture proposed in [WE20] which includes insertion states in between Potts model match states.

Improving alignment quality

In this thesis we proposed a general scoring function for the alignment of two Potts models depending on several hyperparameters whose values have to be set. We realized that our hyperparameter training was very time-consuming and it led us to arbitrarily set some of the hyperparameters following previous empirical observations. To improve our results, a longer hyperparameter training could be performed to train these hyperparameters as well, along with some of the model

construction hyperparameters. Furthermore, in our hyperparameter training we set a time out to 1 minute to speed up the training and hopefully optimize hyperparameters towards faster computations, but our results suggested that this time was probably not sufficient and that it should be set according to the lengths of the models being aligned.

Our alignment scoring function itself could be improved as well. First, though we proposed to remove a fixed offset hyperparameter for each aligned column as an initial measure against overly greedy alignments, our intuition is that, considering the fact that Potts model parameters are not simply probabilities, this offset hyperparameter should depend on the model to be aligned, possibly even on the positions.

Then, so far we stuck with the gap scoring scheme already implemented in the solver, which corresponds to a global alignment with affine gap costs as in Gotoh's algorithm. Besides improvements suggested in the previous section including position-specific gap costs, a simple change we could try out would be to switch from global to local alignments by only applying gap penalties within the aligned region.

Finally, as outlined in section 2.5.1 for profile-profile alignment methods, there are multiple ways to compute the similarity between two columns, and though the similarity score we proposed appears as a natural candidate, this choice is still open to discussion. One way to improve it would be to take into account the context of each position, which MRFalign does by inputting profile sequence context windows centered at each node to its neural networks. In our framework, this could be done by extracting a feature vector for each position which describes its context, and computing the similarity of feature vectors using the scalar product just like other parameters.

Towards PPsearch

In this thesis, we mainly focused on the design of our pairwise Potts model alignment method, PPalign, and on the best way to represent proteins with Potts models so that they can be properly compared. This method is actually intended to be the base component of a future homology search suite, in the same way as the

pairwise pHMM alignment method HHalign is central to HHsuite. In addition to points already raised in the previous sections, several challenges must be addressed before releasing our own PPsuite.

First and foremost, further reflection should be conducted on improving the similarity score. While our scoring function might be appropriate for alignment purposes and yielded encouraging results in preliminary homology detection experiments, some adjustments might be necessary for the specific task of homology detection. For instance, the question of whether or not to include gap penalties in this pairwise comparison score is still open. Besides, even though these scores are normalized with respect to models to be compared, in our experience these values can substantially fluctuate depending notably on chosen hyperparameters, and this can even affect the relative similarity rankings with respect to a given protein. After having elected a relevant similarity score, extensive statistical studies should be carried out to provide more significant scores such as E-values.

Once the quality of our predictions is established, we could start looking into the compromises that can be made to speed up the process while ensuring reliable predictions. Though we believe our alignment method's guaranteed optimality to be an important asset mainly for investigation purposes, in practice it cannot be used for an efficient search against a database. For such use, search can be further sped up with the use of pre-filtering steps including positional only alignments, since the latter are much faster, or by using solutions of the relaxed problem instead.

Building a Potts model for a whole protein family

Originally, the subject of this thesis was to model whole protein families or superfamilies with distant dependencies. With this goal in mind, we came across challenges: how to best represent proteins with Potts models and how to align two Potts models, which became the actual subject of this thesis. Nonetheless, in this section we present the workflow we had in mind to build a Potts model for a whole protein family using successive Potts model alignments.

Starting from a given set of sequences belonging to the family to be modeled,

we propose to build its Potts model following these steps:

1. Cluster available sequences for the family at 30% sequence identity to reduce sampling bias
2. Perform hierarchical clustering on the representative sequences using sequence identity as a metric to obtain a tree for progressive alignment construction (see figure 6.8)
3. Retrieve close homologs of each leaf sequence and infer Potts models to represent the corresponding sets
4. Align Potts models representing leaves following the topology of the tree
5. For each node, infer a Potts model on the MSA built by concatenating MSAs of its children at positions aligned by PPalign
6. Repeat the process until arriving at root

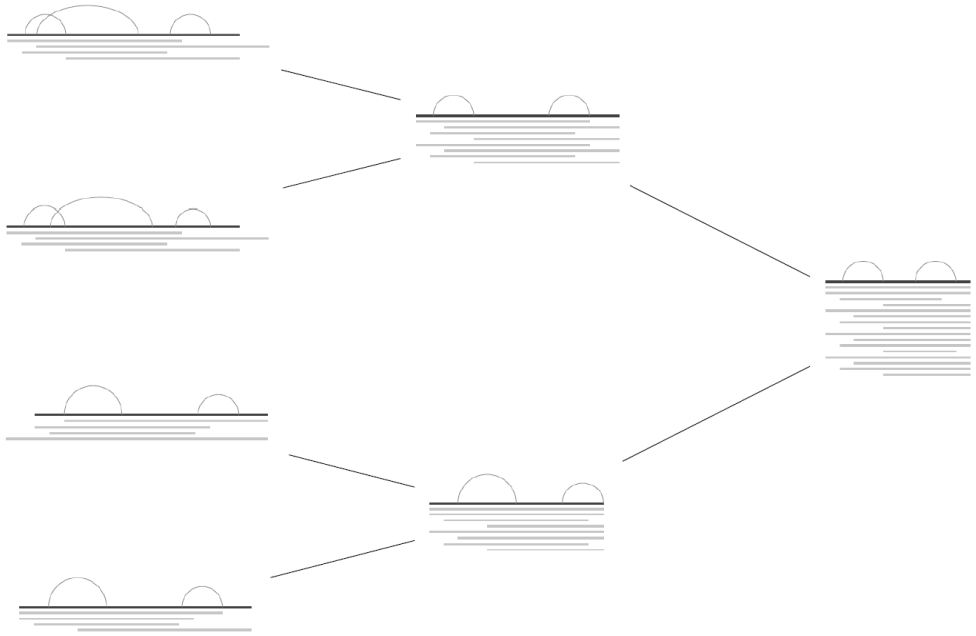
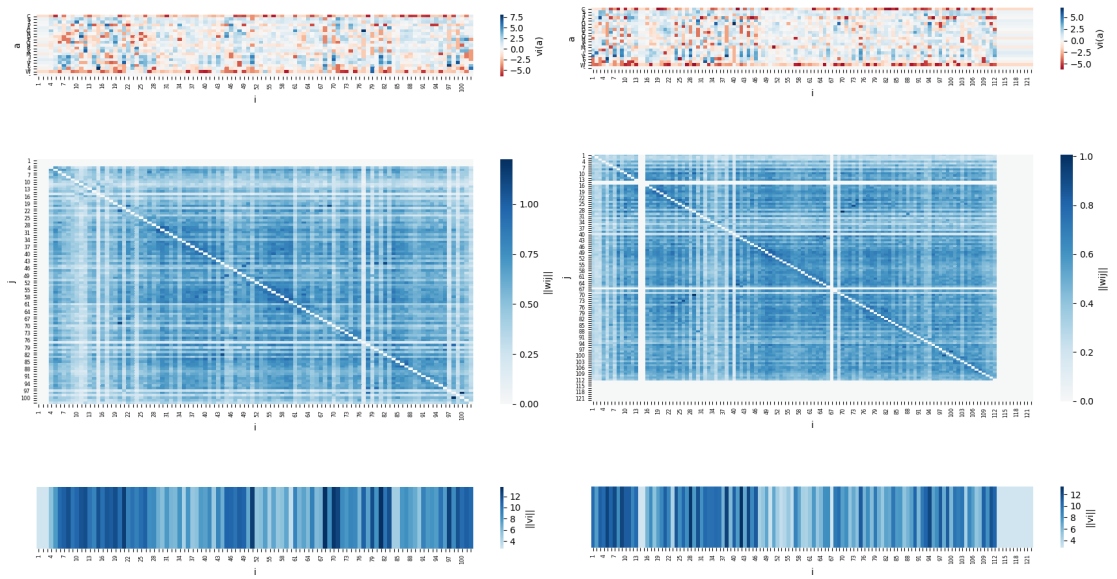


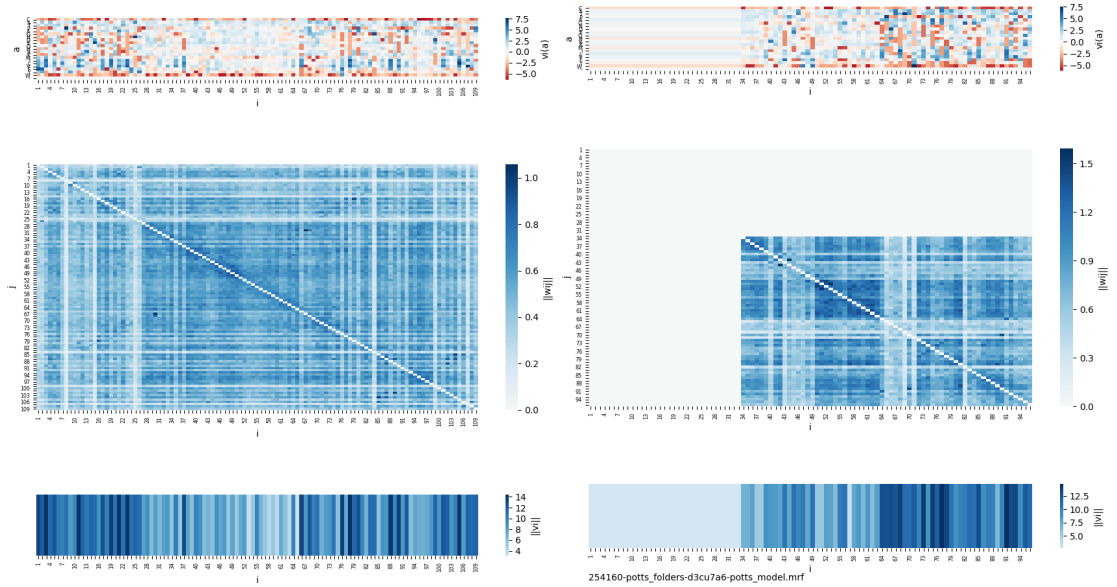
Figure 6.8 – Simplified diagram illustrating our progressive alignment method for the construction of a Potts model for a protein family. Potts models representing subsets of the protein family are aligned following a hierarchical clustering based on sequence identity, their alignment induce a new MSA obtained by merging their train MSAs at aligned positions, on which a new Potts model is inferred and aligned with the Potts model representing the nearest cluster, until a Potts model is built for the whole family.

We applied this workflow to build a Potts model for the Macroglobulin superfamily (SCOPe sunid 254121). Potts models for some domains used to build it are displayed figure 6.9, and the model is displayed figure 6.10.



(a) d3cu7a2 from family 254156

(b) d2a73a3 from family 254157



(c) d3cu7b7 from family 254161

(d) d3cu7a6 from family 254160

Figure 6.9 – Parameters of Potts models built for some domains in different families in the Macroglobulin superfamily

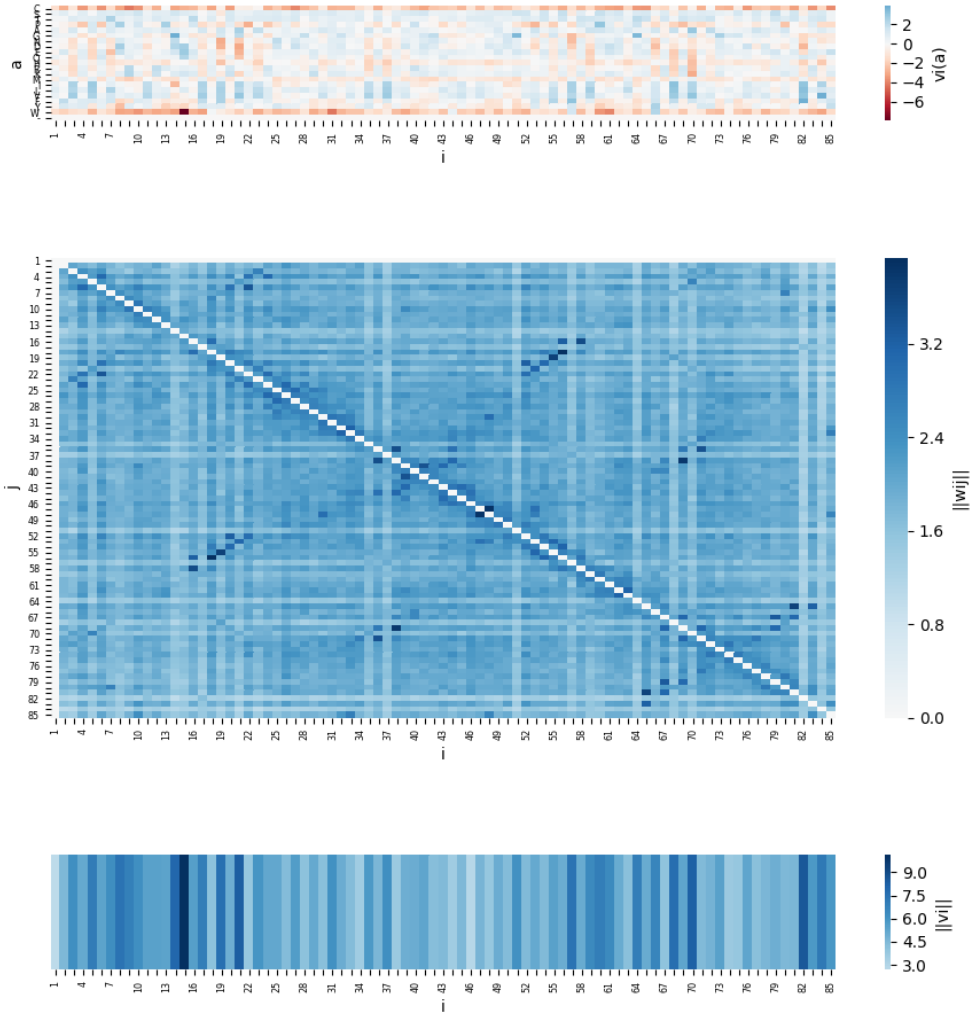


Figure 6.10 – Parameters of the Potts model built for the Macroglobulin superfamily. The central heatmap reveals couplings conserved within the superfamily, which actually correspond to contacts between beta strands. Interestingly, not all these couplings were visible in the individual Potts models (see figure 6.9).

This approach enables to build a multiple sequence alignments for whole protein families by making use of direct coupling information. This could be used to improve the identification of direct couplings conserved within the family

by limiting biases due to multiple sequence alignments built using positional conservation only, and to predict whether a protein belongs to a family by aligning a Potts model built from its sequence to the Potts model built for the whole family. This raises the additional question of whether the two models can be sensibly compared. Ultimately, this offers the prospect of creating databases for the classification of proteins using direct coupling information, where query sequences could be searched against the database using PPalign. A first option would be to provide alternatives to already existing databases currently based on pHMMs such as SUPERFAMILY [Gou+01], with the hope that the use of Potts models will provide more sensitivity in the identification of remote homologs. But the idea of creating a database based on Potts models would probably be even more relevant for specific applications such as the classification of viral proteins: considering the high mutation rates viruses are subject to, co-evolution information provided by Potts models may prove to be a key asset.

Bibliography

[ABS12a] C. Angermüller, A. Biegert, and J. Söding. "Discriminative modelling of context-specific amino acid substitution probabilities". In: *Bioinformatics* 28.24 (2012), pp. 3240–3247 (cit. on p. 44).

[ABS12b] C. Angermüller, A. Biegert, and J. Söding. "Discriminative modelling of context-specific amino acid substitution probabilities". In: *Bioinformatics* 28.24 (2012), pp. 3240–3247 (cit. on p. 58).

[Alb18] B. Alberts. "Molecular biology of the cell". In: (2018) (cit. on p. 6).

[Alt+90] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. "Basic local alignment search tool". In: *Journal of molecular biology* 215.3 (1990), pp. 403–410 (cit. on p. 30).

[Alt+97] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs". In: *Nucleic acids research* 25.17 (1997), pp. 3389–3402 (cit. on pp. 36, 56).

[AMY11] R. Andonov, N. Malod-Dognin, and N. Yanev. "Maximum contact map overlap revisited". In: *Journal of Computational Biology* 18.1 (2011), pp. 27–41 (cit. on pp. 125, 130).

[And+07] A. Andreeva, A. Prlić, T. J. Hubbard, and A. G. Murzin. "SISYPHUS—structural alignments for proteins with non-trivial relationships". In: *Nucleic acids research* 35.suppl_1 (2007), pp. D253–D259 (cit. on pp. xxix, 140).

[Anf73] C. B. Anfinsen. "Principles that govern the folding of protein chains". In: *Science* 181.4096 (1973), pp. 223–230 (cit. on p. 19).

[Ani+17] I. Anishchenko, S. Ovchinnikov, H. Kamisetty, and D. Baker. "Origins of coevolution between residues distant in protein 3D structures". In: *Proceedings of the National Academy of Sciences* 114.34 (2017), pp. 9122–9127 (cit. on p. 48).

[AP15] A. Avila-Herrera and K. S. Pollard. “Coevolutionary analyses require phylogenetically deep alignments and better null models to accurately detect inter-protein contacts within and between species”. In: *BMC bioinformatics* 16.1 (2015), p. 268 (cit. on p. 60).

[AS15] J. Andreani and J. Söding. “bbcontacts: prediction of β-strand pairing from direct coupling patterns”. In: *Bioinformatics* 31.11 (2015), pp. 1729–1737 (cit. on p. 84).

[Ash+00] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. “Gene ontology: tool for the unification of biology”. In: *Nature genetics* 25.1 (2000), pp. 25–29 (cit. on p. 22).

[Atc+00] W. R. Atchley, K. R. Wollenberg, W. M. Fitch, W. Terhalle, and A. W. Dress. “Correlations among amino acid sites in bHLH protein domains: an information theoretic analysis”. In: *Molecular biology and evolution* 17.1 (2000), pp. 164–178 (cit. on p. 59).

[Bal+14] C. Baldassi, M. Zamparo, C. Feinauer, A. Procaccini, R. Zecchina, M. Weigt, and A. Pagnani. “Fast and accurate multivariate Gaussian modeling of protein families: predicting residue contacts and protein-interaction partners”. In: *PloS one* 9.3 (2014), e92721 (cit. on pp. 61, 81).

[Bar+16] J. P. Barton, E. De Leonardis, A. Coucke, and S. Cocco. “ACE: adaptive cluster expansion for maximum entropy graphical model inference”. In: *Bioinformatics* 32.20 (2016), pp. 3089–3097 (cit. on pp. 61, 82).

[Bar+20] P. Barrat-Charlaix, A. P. Muntoni, K. Shimagaki, M. Weigt, and F. Zamponi. “Sparse generative modeling of protein-sequence families”. In: *arXiv preprint arXiv:2011.11259* (2020) (cit. on p. 179).

[Bar18] P. Barrat-Charlaix. “Understanding and improving statistical models of protein sequences”. PhD thesis. Sorbonne Université, 2018 (cit. on p. 95).

[Bat+03] O. Bateman, A. Purkiss, R. Van Montfort, C. Slingsby, C. Graham, and G. Wistow. “Crystal structure of η-crystallin: adaptation of a class 1 aldehyde dehydrogenase for a new role in the eye lens”. In: *Biochemistry* 42.15 (2003), pp. 4349–4356 (cit. on p. 8).

[BB99] J. Besemer and M. Borodovsky. “Heuristic approach to deriving models for gene finding”. In: *Nucleic acids research* 27.19 (1999), pp. 3911–3920.

[Ber+00] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. "The protein data bank". In: *Nucleic acids research* 28.1 (2000), pp. 235–242 (cit. on p. 13).

[Bes75] J. Besag. "Statistical analysis of non-lattice data". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 24.3 (1975), pp. 179–195 (cit. on p. 82).

[Bit+16] A.-F. Bitbol, R. S. Dwyer, L. J. Colwell, and N. S. Wingreen. "Inferring interaction partners from protein sequences". In: *Proceedings of the National Academy of Sciences* 113.43 (2016), pp. 12180–12185 (cit. on p. 61).

[BJ18] D. W. Buchan and D. T. Jones. "Improved protein contact predictions with the MetaPSICOV2 server in CASP12". In: *Proteins: Structure, Function, and Bioinformatics* 86 (2018), pp. 78–83 (cit. on p. 177).

[BK97] C. Burge and S. Karlin. "Prediction of complete gene structures in human genomic DNA". In: *Journal of molecular biology* 268.1 (1997), pp. 78–94.

[BR03] M. J. Betts and R. B. Russell. "Amino acid properties and consequences of substitutions". In: *Bioinformatics for geneticists* 317 (2003), p. 289.

[Bra+01] P. Bradley, L. Cowen, M. Menke, J. King, and B. Berger. "BETAWRAP: successful prediction of parallel β-helices from primary sequence reveals an association with many microbial pathogens". In: *Proceedings of the National Academy of Sciences* 98.26 (2001), pp. 14819–14824 (cit. on p. 54).

[Bro+93] M. Brown, R. Hughey, A. Krogh, I. S. Mian, K. Sjölander, and D. Haussler. "Using Dirichlet mixture priors to derive hidden Markov models for protein families." In: *Ismb*. Vol. 1. 1993, pp. 47–55 (cit. on p. 35).

[But+16] T. C. Butler, J. P. Barton, M. Kardar, and A. K. Chakraborty. "Identification of drug resistance mutations in HIV from constraints on natural evolution". In: *Physical Review E* 93.2 (2016), p. 022412 (cit. on p. 61).

[Che+16] R. R. Cheng, O. Nordesjö, R. L. Hayes, H. Levine, S. C. Flores, J. N. Onuchic, and F. Morcos. "Connecting the sequence-space of bacterial signaling proteins to phenotypes using coevolutionary landscapes". In: *Molecular biology and evolution* 33.12 (2016), pp. 3054–3064 (cit. on p. 61).

[Chi+91] C.-T. Chien, P. L. Bartel, R. Sternglanz, and S. Fields. "The two-hybrid system: a method to identify and clone genes for proteins that interact with a protein of interest." In: *Proceedings of the National Academy of Sciences* 88.21 (1991), pp. 9578–9582 (cit. on p. 23).

[Cho+20] M. Chonofsky, S. H. de Oliveira, K. Krawczyk, and C. M. Deane. "The evolution of contact prediction: Evidence that contact selection in statistical contact prediction is changing". In: *Bioinformatics* 36.6 (2020), pp. 1750–1756 (cit. on p. 67).

[Cho92] C. Chothia. "One thousand families for the molecular biologist". In: *Nature* 357.6379 (1992), pp. 543–544.

[CK12] P. W. Collingridge and S. Kelly. "MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments". In: *BMC bioinformatics* 13.1 (2012), p. 117 (cit. on p. 32).

[Cle11] H. J. Cleaves. "Tertiary Structure (Protein)". In: *Encyclopedia of Astrobiology*. Ed. by M. Gargaud, R. Amils, J. C. Quintanilla, H. J. (Cleaves, W. M. Irvine, D. L. Pinti, and M. Viso. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1659–1659. URL: https://doi.org/10.1007/978-3-642-11274-4_1573 (cit. on p. 13).

[Coc+18] S. Cocco, C. Feinauer, M. Figliuzzi, R. Monasson, and M. Weigt. "Inverse statistical physics of protein sequences: a key issues review". In: *Reports on Progress in Physics* 81.3 (2018), p. 032601 (cit. on pp. 82, 83).

[Com13] W. Commons. *Beta sheets*. 2013. URL: https://commons.wikimedia.org/wiki/File:Beta_sheets.svg (cit. on p. 16).

[Con19] U. Consortium. "UniProt: a worldwide hub of protein knowledge". In: *Nucleic acids research* 47.D1 (2019), pp. D506–D515 (cit. on pp. 23, 45).

[Coz+09] D. Cozzetto, A. Kryshtafovych, K. Fidelis, J. Moult, B. Rost, and A. Tramontano. "Evaluation of template-based models in CASP8 with standard measures". In: *Proteins: Structure, Function, and Bioinformatics* 77.S9 (2009), pp. 18–28 (cit. on p. 46).

[Cro+04] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. "WebLogo: a sequence logo generator". In: *Genome research* 14.6 (2004), pp. 1188–1190 (cit. on pp. 33, 89).

[Cro+19] T. I. Croll, M. D. Sammito, A. Kryshtafovych, and R. J. Read. "Evaluation of template-based modeling in CASP13". In: *Proteins: Structure, Function, and Bioinformatics* 87.12 (2019), pp. 1113–1127 (cit. on p. 46).

[CS96] S. Y. Chung and S. Subbiah. "A structural explanation for the twilight zone of protein sequence homology". In: *Structure* 4.10 (1996), pp. 1123–1127.

[CSG09] S. Capella-Gutierrez, J. M. Silla-Martinez, and T. Gabaldon. "trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses". In: *Bioinformatics* 25.15 (2009), pp. 1972–1973 (cit. on p. 87).

[CTB20] F. Cuturello, G. Tiana, and G. Bussi. "Assessing the accuracy of direct-coupling analysis for RNA contact prediction". In: *RNA* 26.5 (2020), pp. 637–647 (cit. on p. 61).

[Cuf+09] A. L. Cuff, I. Sillitoe, T. Lewis, O. C. Redfern, R. Garratt, J. Thornton, and C. A. Orengo. "The CATH classification revisited—architectures reviewed and new ways to characterize structural divergence in superfamilies". In: *Nucleic acids research* 37.suppl_1 (2009), pp. D310–D314 (cit. on p. 39).

[Dag+12] A. E. Dago, A. Schug, A. Procaccini, J. A. Hoch, M. Weigt, and H. Szurmant. "Structural basis of histidine kinase autophosphorylation deduced by integrating genomics, molecular dynamics, and mutagenesis". In: *Proceedings of the National Academy of Sciences* 109.26 (2012), E1733–E1742 (cit. on p. 61).

[Dan+12] N. M. Daniels, R. Hosur, B. Berger, and L. J. Cowen. "SMURFLite: combining simplified Markov random fields with simulated evolution improves remote homology detection for beta-structural proteins into the twilight zone". In: *Bioinformatics* 28.9 (2012), pp. 1216–1222 (cit. on p. 56).

[Dan+14] N. M. Daniels, A. Gallant, N. Ramsey, and L. J. Cowen. "MRFy: remote homology detection for beta-structural proteins using Markov random fields and stochastic search". In: *IEEE/ACM transactions on computational biology and bioinformatics* 12.1 (2014), pp. 4–16 (cit. on pp. 55, 56).

[De +15] E. De Leonardis, B. Lutz, S. Ratz, S. Cocco, R. Monasson, A. Schug, and M. Weigt. "Direct-Coupling Analysis of nucleotide coevolution facilitates RNA secondary and tertiary structure prediction". In: *Nucleic acids research* 43.21 (2015), pp. 10444–10455 (cit. on p. 61).

[DSO78] M. Dayhoff, R. Schwartz, and B. Orcutt. "22 a model of evolutionary change in proteins". In: *Atlas of protein sequence and structure* 5 (1978), pp. 345–352 (cit. on p. 27).

[Du+20] Z. Du, S. Pan, Q. Wu, Z. Peng, and J. Yang. "CATHER: a novel threading algorithm with predicted contacts". In: *Bioinformatics* 36.7 (2020), pp. 2119–2125 (cit. on p. 53).

[Dur+98] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998 (cit. on pp. 25, 26, 35, 37).

[DWG08] S. D. Dunn, L. M. Wahl, and G. B. Gloor. "Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction". In: *Bioinformatics* 24.3 (2008), pp. 333–340 (cit. on p. 59).

[Edd04] S. R. Eddy. "Where did the BLOSUM62 alignment score matrix come from?" In: *Nature biotechnology* 22.8 (2004), pp. 1035–1036.

[Edd09] S. R. Eddy. "A new generation of homology search tools based on probabilistic inference". In: *Genome Informatics 2009: Genome Informatics Series Vol. 23*. World Scientific, 2009, pp. 205–211 (cit. on p. 39).

[Edd96] S. R. Eddy. "Hidden markov models". In: *Current opinion in structural biology* 6.3 (1996), pp. 361–365 (cit. on p. 36).

[Edd98] S. R. Eddy. "Profile hidden Markov models." In: *Bioinformatics (Oxford, England)* 14.9 (1998), pp. 755–763 (cit. on p. 36).

[Edg] R. C. Edgar. *Qscore*. http://www.drive5.com/qscore/ (cit. on p. 142).

[Edg04] R. C. Edgar. "MUSCLE: multiple sequence alignment with high accuracy and high throughput". In: *Nucleic acids research* 32.5 (2004), pp. 1792–1797 (cit. on pp. 32, 91).

[EHA14] M. Ekeberg, T. Hartonen, and E. Aurell. "Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences". In: *Journal of Computational Physics* 276 (2014), pp. 341–356 (cit. on p. 96).

[Eke+13] M. Ekeberg, C. Lövkvist, Y. Lan, M. Weigt, and E. Aurell. "Improved contact prediction in proteins: using pseudolikelihoods to infer Potts models". In: *Physical Review E* 87.1 (2013), p. 012707 (cit. on pp. 61, 82).

[ElG+19] S. El-Gebali, J. Mistry, A. Bateman, S. R. Eddy, A. Luciani, S. C. Potter, M. Qureshi, L. J. Richardson, G. A. Salazar, A. Smart, et al. "The Pfam protein families database in 2019". In: *Nucleic acids research* 47.D1 (2019), pp. D427–D432 (cit. on pp. xxiv, 39).

[Esq+13] R. O. Esquivel, M. Molina-Espiritu, F. Salas, C. Soriano, C. Barrientos, J. S. Dehesa, and J. A. Dobado. "Decoding the building blocks of life from the perspective of quantum information". In: *Advances in Quantum Mechanics*. IntechOpen, 2013.

[Eya+07] E. Eyal, M. Frenkel-Morgenstern, V. Sobolev, and S. Pietrokovski. "A pair-to-pair amino acids substitution matrix and its applications for protein structure prediction". In: *PROTEINS: Structure, Function, and Bioinformatics* 67.1 (2007), pp. 142–153 (cit. on p. 116).

[FA04] A. A. Fodor and R. W. Aldrich. "Influence of conservation on calculations of amino acid covariance in multiple sequence alignments". In: *Proteins: Structure, Function, and Bioinformatics* 56.2 (2004), pp. 211–221 (cit. on p. 59).

[FBC14] N. K. Fox, S. E. Brenner, and J.-M. Chandonia. "SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures". In: *Nucleic acids research* 42.D1 (2014), pp. D304–D309 (cit. on p. 161).

[FBW18] M. Figliuzzi, P. Barrat-Charlaix, and M. Weigt. "How pairwise coevolutionary models capture the collective residue variability in proteins?" In: *Molecular biology and evolution* 35.4 (2018), pp. 1018–1027 (cit. on pp. 61, 82, 178).

[FCE11] R. D. Finn, J. Clements, and S. R. Eddy. "HMMER web server: interactive sequence similarity searching". In: *Nucleic acids research* 39.suppl_2 (2011), W29–W37 (cit. on pp. xxiv, 36, 39).

[Fei+14a] C. Feinauer, M. J. Skwark, A. Pagnani, and E. Aurell. "Improving contact prediction along three dimensions". In: *PLoS Comput Biol* 10.10 (2014), e1003847 (cit. on p. 85).

[Fei+14b] C. Feinauer, M. J. Skwark, A. Pagnani, and E. Aurell. "Improving contact prediction along three dimensions". In: *PLoS Comput Biol* 10.10 (2014), e1003847 (cit. on p. 179).

[FF14] D. M. Fowler and S. Fields. "Deep mutational scanning: a new style of protein science". In: *Nature methods* 11.8 (2014), pp. 801–807 (cit. on p. 23).

[Fig+16] M. Figliuzzi, H. Jacquier, A. Schug, O. Tenaillon, and M. Weigt. "Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1". In: *Molecular biology and evolution* 33.1 (2016), pp. 268–280 (cit. on p. 61).

[Fir+98] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. "Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans". In: *nature* 391.6669 (1998), pp. 806–811 (cit. on p. 23).

[Fly+17] W. F. Flynn, A. Haldane, B. E. Torbett, and R. M. Levy. "Inference of epistatic effects leading to entrenchment and drug resistance in HIV-1 protease". In: *Molecular biology and evolution* 34.6 (2017), pp. 1291–1306 (cit. on p. 61).

[FMK09] P. Filippakopoulos, S. Müller, and S. Knapp. "SH2 domains: modulators of nonreceptor tyrosine kinase activity". In: *Current opinion in structural biology* 19.6 (2009), pp. 643–649.

[GB15] M. J. Gabanyi and H. M. Berman. "Protein structure annotation resources". In: *Structural Proteomics*. Springer, 2015, pp. 3–20 (cit. on p. 23).

[Gil+01] D. Gilis, S. Massar, N. J. Cerf, and M. Rooman. "Optimality of the genetic code with respect to protein stability and amino-acid frequencies". In: *Genome biology* 2.11 (2001), research0049–1 (cit. on p. 87).

[Gin+07] A.-C. Gingras, M. Gstaiger, B. Raught, and R. Aebersold. "Analysis of protein complexes using mass spectrometry". In: *Nature reviews Molecular cell biology* 8.8 (2007), pp. 645–654 (cit. on p. 23).

[GIP99] D. Goldman, S. Istrail, and C. H. Papadimitriou. "Algorithmic aspects of protein structure similarity". In: *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*. IEEE. 1999, pp. 512–521 (cit. on p. 123).

[GME87] M. Gribskov, A. D. McLachlan, and D. Eisenberg. "Profile analysis: detection of distantly related proteins". In: *Proceedings of the National Academy of Sciences* 84.13 (1987), pp. 4355–4358 (cit. on p. 36).

[Göb+94] U. Göbel, C. Sander, R. Schneider, and A. Valencia. "Correlated mutations and residue contacts in proteins". In: *Proteins: Structure, Function, and Bioinformatics* 18.4 (1994), pp. 309–317 (cit. on p. 59).

[Gou+01] J. Gough, K. Karplus, R. Hughey, and C. Chothia. "Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure". In: *Journal of molecular biology* 313.4 (2001), pp. 903–919 (cit. on pp. 39, 186).

[GP07] R. Gouveia-Oliveira and A. G. Pedersen. "Finding coevolving amino acid residues using row and column weighting of mutual information and multi-dimensional amino acid representation". In: *Algorithms for Molecular Biology* 2.1 (2007), p. 12 (cit. on p. 59).

[Gre90] J. Greer. "Comparative modeling methods: application to the family of the mammalian serine proteases". In: *Proteins: Structure, Function, and Bioinformatics* 7.4 (1990), pp. 317–334.

[GS04] M. M. Gromiha and S. Selvaraj. "Inter-residue interactions in protein folding and stability". In: *Progress in biophysics and molecular biology* 86.2 (2004), pp. 235–277 (cit. on p. 59).

[GS97] M. M. Gromiha and S. Selvaraj. "Influence of medium and long range interactions in different structural classes of globular proteins". In: *Journal of Biological Physics* 23.3 (1997), pp. 151–162 (cit. on p. 161).

[Gue+16] T. Gueudré, C. Baldassi, M. Zamparo, M. Weigt, and A. Pagnani. "Simultaneous identification of specifically interacting paralogs and interprotein contacts by direct coupling analysis". In: *Proceedings of the National Academy of Sciences* 113.43 (2016), pp. 12186–12191 (cit. on p. 61).

[Hay+15] S. Hayat, C. Sander, D. S. Marks, and A. Elofsson. "All-atom 3D structure prediction of transmembrane β-barrel proteins from sequences". In: *Proceedings of the National Academy of Sciences* 112.17 (2015), pp. 5413–5418 (cit. on p. 61).

[HC16] J. M. Heather and B. Chain. "The sequence of sequencers: The history of sequencing DNA". In: *Genomics* 107.1 (2016), pp. 1–8.

[HH03] A. Heger and L. Holm. "Exhaustive enumeration of protein domain families". In: *Journal of molecular biology* 328.3 (2003), pp. 749–767 (cit. on p. 41).

[HH92] S. Henikoff and J. G. Henikoff. "Amino acid substitution matrices from protein blocks". In: *Proceedings of the National Academy of Sciences* 89.22 (1992), pp. 10915–10919 (cit. on p. 28).

[HH96a] J. G. Henikoff and S. Henikoff. "Using substitution probabilities to improve position-specific scoring matrices". In: *Bioinformatics* 12.2 (1996), pp. 135–143 (cit. on p. 35).

[HH96b] J. G. Henikoff and S. Henikoff. “Using substitution probabilities to improve position-specific scoring matrices”. In: *Bioinformatics* 12.2 (1996), pp. 135–143 (cit. on p. 178).

[HK96] R. Hughey and A. Krogh. “Hidden Markov models for sequence analysis: extension and analysis of the basic method”. In: *Bioinformatics* 12.2 (1996), pp. 95–107 (cit. on p. 39).

[HMS13] M. Hauser, C. E. Mayer, and J. Söding. “kClust: fast and sensitive clustering of large protein sequence databases”. In: *BMC bioinformatics* 14.1 (2013), p. 248.

[Hop+12] T. A. Hopf, L. J. Colwell, R. Sheridan, B. Rost, C. Sander, and D. S. Marks. “Three-dimensional structures of membrane proteins from genomic sequencing”. In: *Cell* 149.7 (2012), pp. 1607–1621 (cit. on p. 61).

[Hop+14] T. A. Hopf, C. P. Schärfe, J. P. Rodrigues, A. G. Green, O. Kohlbacher, C. Sander, A. M. Bonvin, and D. S. Marks. “Sequence co-evolution gives 3D contacts and structures of protein complexes”. In: *Elife* 3 (2014), e03430 (cit. on p. 61).

[Hop+15] T. A. Hopf, S. Morinaga, S. Ihara, K. Touhara, D. S. Marks, and R. Benton. “Amino acid coevolution reveals three-dimensional structure and functional domains of insect odorant receptors”. In: *Nature communications* 6.1 (2015), pp. 1–7 (cit. on p. 61).

[Hop+17] T. A. Hopf, J. B. Ingraham, F. J. Poelwijk, C. P. Schärfe, M. Springer, C. Sander, and D. S. Marks. “Mutation effects predicted from sequence co-variation”. In: *Nature biotechnology* 35.2 (2017), pp. 128–135 (cit. on p. 61).

[HS88] D. G. Higgins and P. M. Sharp. “CLUSTAL: a package for performing multiple sequence alignment on a microcomputer”. In: *Gene* 73.1 (1988), pp. 237–244 (cit. on p. 32).

[HS93] L. Holm and C. Sander. “Protein structure comparison by alignment of distance matrices”. In: *Journal of molecular biology* 233.1 (1993), pp. 123–138 (cit. on p. 124).

[HSW03] D. H. Haft, J. D. Selengut, and O. White. “The TIGRFAMs database of protein families”. In: *Nucleic acids research* 31.1 (2003), pp. 371–373 (cit. on pp. xxiv, 39).

[Hua+14] Y. J. Huang, B. Mao, J. M. Aramini, and G. T. Montelione. “Assessment of template-based protein structure predictions in CASP10”. In: *Proteins: Structure, Function, and Bioinformatics* 82 (2014), pp. 43–56 (cit. on p. 46).

[HW19] A. J. Hockenberry and C. O. Wilke. "Evolutionary couplings detect side-chain interactions". In: *PeerJ* 7 (2019), e7280.

[IAE09] K. Illergård, D. H. Ardell, and A. Elofsson. "Structure is three to ten times more conserved than sequence—a study of structural response in protein cores". In: *Proteins: Structure, Function, and Bioinformatics* 77.3 (2009), pp. 499–508.

[Jay57] E. T. Jaynes. "Information theory and statistical mechanics". In: *Physical review* 106.4 (1957), p. 620 (cit. on p. 62).

[Jef99] C. J. Jeffery. "Moonlighting proteins". In: *Trends in biochemical sciences* 24.1 (1999), pp. 8–11 (cit. on p. 8).

[Jon+12] D. T. Jones, D. W. Buchan, D. Cozzetto, and M. Pontil. "PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments". In: *Bioinformatics* 28.2 (2012), pp. 184–190 (cit. on p. 59).

[Jon+15] D. T. Jones, T. Singh, T. Kosciolek, and S. Tetchner. "MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins". In: *Bioinformatics* 31.7 (2015), pp. 999–1006.

[Kal+18] I. Kalvari, J. Argasinska, N. Quinones-Olvera, E. P. Nawrocki, E. Rivas, S. R. Eddy, A. Bateman, R. D. Finn, and A. I. Petrov. "Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families". In: *Nucleic acids research* 46.D1 (2018), pp. D335–D342 (cit. on p. 69).

[Kat+02] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata. "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform". In: *Nucleic acids research* 30.14 (2002), pp. 3059–3066 (cit. on p. 32).

[KC10] A. Kumar and L. Cowen. "Recognition of beta-structural motifs using hidden Markov models trained with simulated evolution". In: *Bioinformatics* 26.12 (2010), pp. i287–i293 (cit. on p. 56).

[Ken+58] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. Parrish, H. Wyckoff, and D. C. Phillips. "A three-dimensional model of the myoglobin molecule obtained by x-ray analysis". In: *Nature* 181.4610 (1958), pp. 662–666 (cit. on p. 23).

[KF09] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009 (cit. on p. 82).

[Kim+03] D. Kim, D. Xu, J.-t. Guo, K. Ellrott, and Y. Xu. "PROSPECT II: protein structure prediction program for genome-scale applications". In: *Protein engineering* 16.9 (2003), pp. 641–650.

[Kin80] R. Kindermann. "Markov random fields and their applications". In: *American mathematical society* (1980) (cit. on p. 51).

[KOB13] H. Kamisetty, S. Ovchinnikov, and D. Baker. "Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era". In: *Proceedings of the National Academy of Sciences* 110.39 (2013), pp. 15674–15679 (cit. on pp. 61, 71, 82, 179).

[Koc+91] C. A. Koch, D. Anderson, M. F. Moran, C. Ellis, and T. Pawson. "SH2 and SH3 domains: elements that control interactions of cytoplasmic signaling proteins". In: *Science* 252.5006 (1991), pp. 668–674.

[Kro+94] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler. "Hidden Markov models in computational biology. Applications to protein modeling". In: *Journal of molecular biology* 235.5 (1994), pp. 1501–1531 (cit. on p. 36).

[Kro97] A. Krogh. "Two methods for improving performance of an HMM and their application for gene finding". In: *Center for Biological Sequence Analysis. Phone* 45 (1997), p. 4525.

[Kro98] A. Krogh. *In SL Salzberg et al., An Introduction to Hidden Markov Models for Biological Sequences, Computational Methods in Molecular Biology.* 1998 (cit. on p. 39).

[KS83] W. Kabsch and C. Sander. "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features". In: *Biopolymers: Original Research on Biomolecules* 22.12 (1983), pp. 2577–2637 (cit. on p. 44).

[Lan11] E. S. Lander. "Initial impact of the sequencing of the human genome". In: *Nature* 470.7333 (2011), pp. 187–197 (cit. on p. 6).

[Lap+99] A. S. Lapedes, B. G. Giraud, L. Liu, and G. D. Stormo. "Correlated mutations in models of protein sequences: phylogenetic and structural effects". In: *Lecture Notes-Monograph Series* (1999), pp. 236–256 (cit. on pp. 60, 63).

[Lat94] R. H. Lathrop. "The protein threading problem with sequence amino acid interaction preferences is NP-complete". In: *Protein Engineering, Design and Selection* 7.9 (1994), pp. 1059–1068 (cit. on p. 53).

[Les05] A. Lesk. *Introduction to bioinformatics.* Oxford university press, 2005.

[LFS09] T. Lassmann, O. Frings, and E. L. Sonnhammer. "Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features". In: *Nucleic acids research* 37.3 (2009), pp. 858–865 (cit. on p. 32).

[LME91] R. Lüthy, A. D. McLachlan, and D. Eisenberg. "Secondary structure-based profiles: Use of structure-conserving scoring tables in searching protein sequence databases for structural similarities". In: *Proteins: Structure, Function, and Bioinformatics* 10.3 (1991), pp. 229–239.

[LS94] R. H. Lathrop and T. F. Smith. "A branch-and-bound algorithm for optimal protein threading with pairwise (contact potential) amino acid interactions". In: *HICSS (5)*. 1994, pp. 365–374 (cit. on p. 54).

[LS96] R. H. Lathrop and T. F. Smith. "Global optimum protein threading with gapped alignment and empirical pair score functions". In: *Journal of molecular biology* 255.4 (1996), pp. 641–665.

[LZS04] W. Li, Y. Zhang, and J. Skolnick. "Application of sparse NMR restraints to large-scale protein structure prediction". In: *Biophysical journal* 87.2 (2004), pp. 1241–1248 (cit. on p. 59).

[Ma+14] J. Ma, S. Wang, Z. Wang, and J. Xu. "MRFalign: protein homology detection through alignment of Markov random fields". In: *PLoS Comput Biol* 10.3 (2014), e1003500 (cit. on pp. xxx, 56).

[Mal+11] N. Malod-Dognin, M. Le Boudic-Jamin, P. Kamath, and R. Andonov. "Using dominances for solving the protein family identification problem". In: *International Workshop on Algorithms in Bioinformatics*. Springer. 2011, pp. 201–212 (cit. on p. 130).

[Man+14] J. K. Mann, J. P. Barton, A. L. Ferguson, S. Omarjee, B. D. Walker, A. Chakraborty, and T. Ndung'u. "The fitness landscape of HIV-1 gag: advanced modeling approaches and validation of model predictions by in vitro testing". In: *PLoS Comput Biol* 10.8 (2014), e1003776 (cit. on p. 61).

[Mar+11a] V. Mariani, F. Kiefer, T. Schmidt, J. Haas, and T. Schwede. "Assessment of template based protein structure predictions in CASP9". In: *Proteins: Structure, Function, and Bioinformatics* 79.S10 (2011), pp. 37–58 (cit. on p. 46).

[Mar+11b] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander. "Protein 3D structure computed from evolutionary sequence variation". In: *PloS one* 6.12 (2011), e28766 (cit. on p. 61).

[Mar95] J. L. Martin. “Thioredoxin—a fold for all reasons”. In: *Structure* 3.3 (1995), pp. 245–250 (cit. on p. 158).

[Mav+10] L. Mavridis, V. Venkatraman, D. Ritchie, N. Morikawa, R. Andonov, A. Cornu, N. Malod-Dognin, J. Nicolas, M. Temerinac-Ott, M. Reisert, et al. “Shrec-10 track: Protein models”. In: *3DOR: Eurographics Workshop on 3D Object Retrieval*. 2010, pp. 117–124 (cit. on p. 130).

[MB19] D. Malinverni and A. Barducci. “Coevolutionary analysis of protein sequences for molecular modeling”. In: *Biomolecular Simulations*. Springer, 2019, pp. 379–397 (cit. on pp. 82, 177).

[MBC08] M. Menke, B. Berger, and L. Cowen. “Matt: local flexibility aids protein multiple structure alignment”. In: *PLoS Comput Biol* 4.1 (2008), e10 (cit. on p. 54).

[MBC10] M. Menke, B. Berger, and L. Cowen. “Markov random fields reveal an N-terminal double beta-propeller motif as part of a bacterial hybrid two-component sensor system”. In: *Proceedings of the National Academy of Sciences* 107.9 (2010), pp. 4069–4074 (cit. on p. 54).

[MBJ00] L. J. McGuffin, K. Bryson, and D. T. Jones. “The PSIPRED protein structure prediction server”. In: *Bioinformatics* 16.4 (2000), pp. 404–405 (cit. on pp. 44, 57).

[Mir+17] M. Mirdita, L. von den Driesch, C. Galiez, M. J. Martin, J. Söding, and M. Steinegger. “Uniclust databases of clustered and deeply annotated protein sequences and alignments”. In: *Nucleic acids research* 45.D1 (2017), pp. D170–D176 (cit. on pp. 45, 84).

[Mit+15] A. Mitchell, H.-Y. Chang, L. Daugherty, M. Fraser, S. Hunter, R. Lopez, C. McAnulla, C. McMenamin, G. Nuka, S. Pesseat, et al. “The InterPro protein families database: the classification resource after 15 years”. In: *Nucleic acids research* 43.D1 (2015), pp. D213–D221 (cit. on p. 23).

[Mon+16] B. Monastyrskyy, D. D’Andrea, K. Fidelis, A. Tramontano, and A. Kryshtafovych. “New encouraging developments in contact prediction: Assessment of the CASP 11 results”. In: *Proteins: Structure, Function, and Bioinformatics* 84 (2016), pp. 131–144 (cit. on p. 61).

[Moo18] B. Moore. *Historic cost of sequencing a human genome*. 2018. URL: https://commons.wikimedia.org/wiki/File:Historic_cost_of_sequencing_a_human_genome.svg (cit. on p. 22).

[Mor+11] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt. "Direct-coupling analysis of residue coevolution captures native contacts across many protein families". In: *Proceedings of the National Academy of Sciences* 108.49 (2011), E1293–E1301 (cit. on pp. 48, 49, 61, 81, 84, 93).

[MSG03] D. Mittelman, R. Sadreyev, and N. Grishin. "Probabilistic scoring measures for profile–profile comparison yield more accurate short seed alignments". In: *Bioinformatics* 19.12 (2003), pp. 1531–1539 (cit. on p. 42).

[Mun+19] A. P. Muntoni, A. Pagnani, M. Weigt, and F. Zamponi. "Using Direct Coupling Analysis for the protein sequences alignment problem". In: *CECAM 2019 - workshop on Co-evolutionary methods for the prediction and design of protein structure and interactions*. 2019 (cit. on p. 67).

[Mun+20] A. P. Muntoni, A. Pagnani, M. Weigt, and F. Zamponi. "Aligning biological sequences by exploiting residue conservation and coevolution". In: *arXiv preprint arXiv:2005.08500* (2020) (cit. on pp. 68, 179).

[Mur+95] A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia, et al. "SCOP: a structural classification of proteins database for the investigation of sequences and structures". In: *Journal of molecular biology* 247.4 (1995), pp. 536–540 (cit. on pp. 39, 58, 161).

[Mus+92] A. Musacchio, T. Gibson, V.-P. Lehto, and M. Saraste. "SH3—an abundant protein domain in search of a function". In: *FEBS letters* 307.1 (1992), pp. 55–61 (cit. on p. 18).

[MWS94] A. Musacchio, M. Wilmanns, and M. Saraster. "Structure and function of the SH3 domain". In: *Progress in biophysics and molecular biology* 61.3 (1994), pp. 283–297 (cit. on p. 17).

[NE13] E. P. Nawrocki and S. R. Eddy. "Infernal 1.1: 100-fold faster RNA homology searches". In: *Bioinformatics* 29.22 (2013), pp. 2933–2935 (cit. on p. 69).

[Neh94] E. Neher. "How frequent are correlated changes in families of protein sequences?" In: *Proceedings of the National Academy of Sciences* 91.1 (1994), pp. 98–102 (cit. on p. 48).

[NG10] T. W. Nilsen and B. R. Graveley. "Expansion of the eukaryotic proteome by alternative splicing". In: *Nature* 463.7280 (2010), pp. 457–463 (cit. on p. 6).

[NGE93] M. Nikkola, F. Gleason, and H. Eklund. “Reduction of mutant phage T4 glutaredoxins by Escherichia coli thioredoxin reductase.” In: *Journal of Biological Chemistry* 268.6 (1993), pp. 3845–3849 (cit. on p. 160).

[NHH00] C. Notredame, D. G. Higgins, and J. Heringa. “T-Coffee: A novel method for fast and accurate multiple sequence alignment”. In: *Journal of molecular biology* 302.1 (2000), pp. 205–217 (cit. on p. 32).

[NJ12] T. Nugent and D. T. Jones. “Accurate de novo structure prediction of large transmembrane protein domains using fragment-assembly and correlated mutation analysis”. In: *Proceedings of the National Academy of Sciences* 109.24 (2012), E1540–E1547 (cit. on p. 61).

[OKB14] S. Ovchinnikov, H. Kamisetty, and D. Baker. “Robust and accurate prediction of residue–residue interactions across protein interfaces using evolutionary information”. In: *Elife* 3 (2014), e02030 (cit. on p. 61).

[ORV99] O. Olmea, B. Rost, and A. Valencia. “Effective use of sequence correlation and conservation in fold recognition”. In: *Journal of molecular biology* 293.5 (1999), pp. 1221–1239 (cit. on p. 59).

[OV97] O. Olmea and A. Valencia. “Improving contact predictions by the combination of correlated mutations and other sources of sequence information”. In: *Folding and Design* 2 (1997), S25–S32 (cit. on p. 59).

[Ovc+17] S. Ovchinnikov, H. Park, N. Varghese, P.-S. Huang, G. A. Pavlopoulos, D. E. Kim, H. Kamisetty, N. C. Kyrpides, and D. Baker. “Protein structure determination using metagenome sequence data”. In: *Science* 355.6322 (2017), pp. 294–298 (cit. on pp. 61, 177).

[OWE04] T. Ohlson, B. Wallner, and A. Elofsson. “Profile–profile methods provide improved fold-recognition: A study of different profile–profile alignment methods”. In: *Proteins: Structure, Function, and Bioinformatics* 57.1 (2004), pp. 188–197 (cit. on pp. 42, 43).

[OZ01] N. von Ohsen and R. Zimmer. “Improving profile-profile alignments via log average scoring”. In: *International Workshop on Algorithms in Bioinformatics*. Springer. 2001, pp. 11–26 (cit. on p. 41).

[Pan+19] A. P. Pandurangan, J. Stahlhacke, M. E. Oates, B. Smithers, and J. Gough. “The SUPERFAMILY 2.0 database: a significant proteome update and a new webserver”. In: *Nucleic acids research* 47.D1 (2019), pp. D490–D494 (cit. on p. 39).

[Pea13] W. R. Pearson. "An introduction to sequence similarity ("homology") searching". In: *Current protocols in bioinformatics* 42.1 (2013), pp. 3–1.

[PG01] J. Pei and N. V. Grishin. "AL2CO: calculation of positional conservation in a protein sequence alignment". In: *Bioinformatics* 17.8 (2001), pp. 700–712 (cit. on p. 44).

[PHH96] S. Pietrokovski, J. G. Henikoff, and S. Henikoff. "The Blocks database—a system for protein classification". In: *Nucleic acids research* 24.1 (1996), pp. 197–200 (cit. on pp. 28, 40).

[Pie96] S. Pietrokovski. "Searching databases of conserved sequence regions by aligning protein multiple-alignments". In: *Nucleic acids research* 24.19 (1996), pp. 3836–3845 (cit. on p. 40).

[PL88] W. R. Pearson and D. J. Lipman. "Improved tools for biological sequence comparison". In: *Proceedings of the National Academy of Sciences* 85.8 (1988), pp. 2444–2448 (cit. on p. 30).

[PROa] PROSITE. *Sequence logo for PS50110*. https://prosite.expasy.org/cgi-bin/prosite/sequence_logo.cgi?ac=PS50110. (Visited on) (cit. on p. 157).

[PROb] PROSITE. *Sequence logo for PS50279*. https://prosite.expasy.org/cgi-bin/prosite/sequence_logo.cgi?ac=PS50279. (Visited on) (cit. on p. 155).

[PX11] J. Peng and J. Xu. "RaptorX: exploiting structure information for protein alignment by statistical inference". In: *Proteins: Structure, Function, and Bioinformatics* 79.S10 (2011), pp. 161–171 (cit. on p. 53).

[Ree+00] M. G. Reese, D. Kulp, H. Tammana, and D. Haussler. "Genie—gene finding in Drosophila melanogaster". In: *Genome Research* 10.4 (2000), pp. 529–538.

[Rem+12] M. Remmert, A. Biegert, A. Hauser, and J. Söding. "HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment". In: *Nature methods* 9.2 (2012), pp. 173–175 (cit. on pp. xxvii, 45, 83).

[Ros+03] B. Rost, J. Liu, R. Nair, K. O. Wrzeszczynski, and Y. Ofran. "Automatic prediction of protein function". In: *Cellular and Molecular Life Sciences CMLS* 60.12 (2003), pp. 2637–2650 (cit. on p. 22).

[Ryc+00] L. Rychlewski, W. Li, L. Jaroszewski, and A. Godzik. “Comparison of sequence profiles. Strategies for structural predictions using sequence information”. In: *Protein Science* 9.2 (2000), pp. 232–241 (cit. on p. 42).

[SAD00] J. M. Sauder, J. W. Arthur, and R. L. Dunbrack Jr. “Large-scale comparison of protein sequence alignment algorithms with structure alignments”. In: *Proteins: Structure, Function, and Bioinformatics* 40.1 (2000), pp. 6–22.

[ŠB93] A. Šali and T. L. Blundell. “Comparative protein modelling by satisfaction of spatial restraints”. In: *Journal of molecular biology* 234.3 (1993), pp. 779–815 (cit. on p. 45).

[SBG03] R. I. Sadreyev, D. Baker, and N. V. Grishin. “Profile–profile comparisons by COMPASS predict intricate homologies between protein families”. In: *Protein Science* 12.10 (2003), pp. 2262–2272 (cit. on p. 43).

[SBL05] J. Söding, A. Biegert, and A. N. Lupas. “The HHpred interactive server for protein homology detection and structure prediction”. In: *Nucleic acids research* 33.suppl_2 (2005), W244–W248 (cit. on pp. 45, 53, 57).

[SC75] F. Sanger and A. R. Coulson. “A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase”. In: *Journal of molecular biology* 94.3 (1975), pp. 441–448.

[Sch+09] A. Schug, M. Weigt, J. N. Onuchic, T. Hwa, and H. Szurmant. “High-resolution protein complexes from integrating genomic information with molecular simulation”. In: *Proceedings of the National Academy of Sciences* 106.52 (2009), pp. 22124–22129 (cit. on p. 61).

[Sch+86] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. “Information content of binding sites on nucleotide sequences”. In: *Journal of molecular biology* 188.3 (1986), pp. 415–431 (cit. on p. 33).

[Sch+95] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. “Quantitative monitoring of gene expression patterns with a complementary DNA microarray”. In: *Science* 270.5235 (1995), pp. 467–470 (cit. on p. 23).

[Sch15] Schrödinger, LLC. “The PyMOL Molecular Graphics System, Version 1.8”. Nov. 2015 (cit. on pp. 6, 13).

[SED97] E. L. Sonnhammer, S. R. Eddy, and R. Durbin. “Pfam: a comprehensive database of protein domain families based on seed alignments”. In: *Proteins: Structure, Function, and Bioinformatics* 28.3 (1997), pp. 405–420 (cit. on pp. xxiv, 33, 39).

[See] S. Seemayer. *GitHub CCMpred - Frequently Asked Questions (FAQ)*. https://github.com/soedinglab/CCMpred/wiki/FAQ (cit. on p. 83).

[SG03] R. Sadreyev and N. Grishin. "COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance". In: *Journal of molecular biology* 326.1 (2003), pp. 317–336 (cit. on p. 40).

[SGS14] S. Seemayer, M. Gruber, and J. Söding. "CCMpred—fast and precise prediction of protein residue–residue contacts from correlated mutations". In: *Bioinformatics* 30.21 (2014), pp. 3128–3130 (cit. on pp. 61, 82, 83, 150).

[Sig+12] C. J. Sigrist, E. De Castro, L. Cerutti, B. A. Cuche, N. Hulo, A. Bridge, L. Bougueleret, and I. Xenarios. "New and continuing developments at PROSITE". In: *Nucleic acids research* 41.D1 (2012), pp. D344–D347 (cit. on pp. 36, 155).

[Söd05] J. Söding. "Protein homology detection by HMM–HMM comparison". In: *Bioinformatics* 21.7 (2005), pp. 951–960 (cit. on pp. xxiv, 43).

[Spr70] C. Spr. "A general method applicable to the search for similarities in the amino acid sequence of two proteins". In: *Mol. Biol* 48 (1970), pp. 443–153 (cit. on p. 30).

[SRG00] A. M. Stock, V. L. Robinson, and P. N. Goudreau. "Two-component signal transduction". In: *Annual review of biochemistry* 69.1 (2000), pp. 183–215 (cit. on p. 157).

[SS17] M. Steinegger and J. Söding. "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets". In: *Nature biotechnology* 35.11 (2017), pp. 1026–1028 (cit. on p. 156).

[SS18] M. Steinegger and J. Söding. "Clustering huge protein sequence sets in linear time". In: *Nature communications* 9.1 (2018), pp. 1–8.

[SSL94] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. "Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames". In: *Nucleic Acids Research* 22.24 (1994), pp. 5156–5163.

[SSL95] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. "Identification of human gene structure using linear discriminant functions and dynamic programming." In: *Ismb*. Vol. 3. 1995, pp. 367–375.

[Ste+19] M. Steinegger, M. Meier, M. Mirdita, H. Vöhringer, S. J. Haunsberger, and J. Söding. "HH-suite3 for fast remote homology detection and deep protein annotation". In: *BMC bioinformatics* 20.1 (2019), pp. 1–15 (cit. on pp. xxiv, 39, 45, 57).

[Ste01] L. Stein. "Genome annotation: from sequence to biology". In: *Nature reviews genetics* 2.7 (2001), pp. 493–503.

[Sut+15] L. Sutto, S. Marsili, A. Valencia, and F. L. Gervasio. "From residue coevolution to protein conformational ensembles and functional dynamics". In: *Proceedings of the National Academy of Sciences* 112.44 (2015), pp. 13567–13572 (cit. on pp. 61, 82).

[SW+81a] T. F. Smith, M. S. Waterman, et al. "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1 (1981), pp. 195–197 (cit. on p. 30).

[SW+81b] T. F. Smith, M. S. Waterman, et al. "Identification of common molecular subsequences". In: *Journal of molecular biology* 147.1 (1981), pp. 195–197 (cit. on p. 32).

[SW18] H. Szurmant and M. Weigt. "Inter-residue, inter-protein and inter-family coevolution: bridging the scales". In: *Current opinion in structural biology* 50 (2018), pp. 26–32.

[Tay86] W. R. Taylor. "The classification of amino acid conservation". In: *Journal of theoretical Biology* 119.2 (1986), pp. 205–218 (cit. on pp. 11, 12).

[TC19] H. Talibart and F. Coste. "Using residues coevolution to search for protein homologs through alignment of Potts models". In: *CECAM 2019 - workshop on Co-evolutionary methods for the prediction and design of protein structure and interactions*. 2019 (cit. on p. 67).

[Tho+03] P. D. Thomas, M. J. Campbell, A. Kejariwal, H. Mi, B. Karlak, R. Daverman, K. Diemer, A. Muruganujan, and A. Narechania. "PANTHER: a library of protein families and subfamilies indexed by function". In: *Genome research* 13.9 (2003), pp. 2129–2141 (cit. on p. 39).

[TL03] E. R. Tillier and T. W. Lui. "Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments". In: *Bioinformatics* 19.6 (2003), pp. 750–755 (cit. on p. 59).

[TN20] R. Trivedi and H. A. Nagarajaram. "Substitution scoring matrices for proteins-An overview". In: *Protein Science* 29.11 (2020), pp. 2150–2163 (cit. on p. 28).

[TRB08] J. Thomas, N. Ramakrishnan, and C. Bailey-Kellogg. "Graphical models of residue coupling in protein families". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 5.2 (2008), pp. 183–197.

[Ugu+17] G. Uguzzoni, S. J. Lovis, F. Oteri, A. Schug, H. Szurmant, and M. Weigt. "Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis". In: *Proceedings of the National Academy of Sciences* 114.13 (2017), E2662–E2671 (cit. on p. 84).

[UM91] E. C. Uberbacher and R. J. Mural. "Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach". In: *Proceedings of the National Academy of Sciences* 88.24 (1991), pp. 11261–11265.

[Uni19] UniProt. *Sequence annotation (Features)*. 2019. URL: https://www.uniprot.org/help/sequence_annotation (cit. on p. 22).

[Ven+01] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al. "The sequence of the human genome". In: *science* 291.5507 (2001), pp. 1304–1351.

[Vit67] A. Viterbi. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE transactions on Information Theory* 13.2 (1967), pp. 260–269 (cit. on p. 38).

[Vor17] S. Vorberg. "Bayesian Statistical Approach for Protein Residue-Residue Contact Prediction". PhD thesis. Ludwig-Maximilians-Universität, 2017 (cit. on pp. 85, 86, 95).

[VSS18] S. Vorberg, S. Seemayer, and J. Söding. "Synthetic protein alignments by CCMgen quantify noise in residue-residue contact prediction". In: *PLoS computational biology* 14.11 (2018), e1006526 (cit. on pp. xxvii, 83).

[WAK12] I. Wohlers, R. Andonov, and G. W. Klau. "DALIX: optimal DALI protein structure alignment". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 10.1 (2012), pp. 26–36 (cit. on pp. xxviii, 123, 125, 130).

[Wal+06] I. M. Wallace, O. O'sullivan, D. G. Higgins, and C. Notredame. "M-Coffee: combining multiple sequence alignment methods with T-Coffee". In: *Nucleic acids research* 34.6 (2006), pp. 1692–1699 (cit. on p. 32).

[Wan+10] Z. Wang, F. Zhao, J. Peng, and J. Xu. "Protein 8-class secondary structure prediction using conditional neural fields". In: *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2010, pp. 109–114 (cit. on p. 57).

[Wan+13] S. Wang, J. Ma, J. Peng, and J. Xu. "Protein structure alignment beyond spatial proximity". In: *Scientific reports* 3 (2013), p. 1448 (cit. on p. 57).

[Wan+19] Y. Wang, Q. Shi, P. Yang, C. Zhang, S. Mortuza, Z. Xue, K. Ning, and Y. Zhang. "Fueling ab initio folding with marine metagenomics enables structure and function predictions of new protein families". In: *Genome biology* 20.1 (2019), pp. 1–14 (cit. on p. 177).

[WD04] G. Wang and R. L. Dunbrack Jr. "Scoring profile-to-profile sequence alignments". In: *Protein Science* 13.6 (2004), pp. 1612–1626 (cit. on p. 134).

[WE20] G. W. Wilburn and S. R. Eddy. "Remote homology search with hidden Potts models". In: *BioRxiv* (2020) (cit. on pp. 70, 71, 179).

[Web+92] E. C. Webb et al. "Recommendations of the Nomenclature Committee of the International Union of Biochemistry and molecular Biology on the Nomenclature and Classification of Enzymes". In: *Enzyme Nomenclature 1992*. Academic Press, Inc., 1992, pp. 346–365 (cit. on p. 22).

[Wei+09] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa. "Identification of direct residue contacts in protein–protein interaction by message passing". In: *Proceedings of the National Academy of Sciences* 106.1 (2009), pp. 67–72 (cit. on pp. 61, 81, 92).

[Wik12] P. sur Wikipédia français / Public domain. *Acides aminés propriétés diagramme Venn*. 2012. URL: https://commons.wikimedia.org/wiki/File:Acides_amin%C3%A9s_propri%C3%A9t%C3%A9s_diagramme_Venn.svg (cit. on p. 12).

[Wit+17] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. "Chapter 9 - Probabilistic methods". In: *Data Mining (Fourth Edition)*. Ed. by I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. Fourth Edition. Morgan Kaufmann, 2017, pp. 335–416 (cit. on p. 51).

[WKP13] C. Wang, N. Komodakis, and N. Paragios. "Markov random field modeling, inference & learning in computer vision & image understanding: A survey". In: *Computer Vision and Image Understanding* 117.11 (2013), pp. 1610–1627 (cit. on p. 51).

[WMS94] J. V. White, I. Muchnik, and T. F. Smith. "Modeling protein cores with Markov random fields". In: *Mathematical biosciences* 124.2 (1994), pp. 149–179 (cit. on p. 53).

[Woh+12] I. Wohlers, N. Malod-Dognin, R. Andonov, and G. W. Klau. "CSA: comprehensive comparison of pairwise protein structure alignments". In: *Nucleic acids research* 40.W1 (2012), W303–W309 (cit. on p. 129).

[Woh12] I. Wohlers. "Exact Algorithms For Pairwise Protein Structure Alignment". PhD thesis. Vrije Universiteit, Jan. 2012, pp. 1–147 (cit. on pp. xxviii, 128, 129, 175).

[Wut89] K. Wuthrich. "Protein structure determination in solution by nuclear magnetic resonance spectroscopy". In: *Science* 243.4887 (1989), pp. 45–50 (cit. on p. 23).

[WZ08] S. Wu and Y. Zhang. "MUSTER: improving protein sequence profile–profile alignments by using multiple sources of structure information". In: *Proteins: Structure, Function, and Bioinformatics* 72.2 (2008), pp. 547–556 (cit. on p. 53).

[Xu+14] D. Xu, L. Jaroszewski, Z. Li, and A. Godzik. "FFAS-3D: improving fold recognition by including optimized structural features and template re-ranking". In: *Bioinformatics* 30.5 (2014), pp. 660–667 (cit. on p. 53).

[XX00] Y. Xu and D. Xu. "Protein threading using PROSPECT: design and evaluation". In: *Proteins: Structure, Function, and Bioinformatics* 40.3 (2000), pp. 343–354 (cit. on p. 53).

[Yan+08] N. Yanev, R. Andonov, P. Veber, and S. Balev. "Lagrangian approaches for a class of matching problems in computational biology". In: *Computers & Mathematics with Applications* 55.5 (2008), pp. 1054–1067 (cit. on p. 54).

[Yan+11] Y. Yang, E. Faraggi, H. Zhao, and Y. Zhou. "Improving protein fold recognition and template-based modeling by employing probabilistic-based matching between predicted one-dimensional structural properties of query and corresponding native properties of templates". In: *Bioinformatics* 27.15 (2011), pp. 2076–2082 (cit. on p. 53).

[YL02] G. Yona and M. Levitt. "Within the twilight zone: a sensitive profile-profile comparison tool based on information theory". In: *Journal of molecular biology* 315.5 (2002), pp. 1257–1275 (cit. on pp. 40, 41).

[YWA03] Y.-K. Yu, J. C. Wootton, and S. F. Altschul. "The compositional adjustment of amino acid substitution matrices". In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15688–15693.

[Zha+20] C. Zhang, W. Zheng, S. Mortuza, Y. Li, and Y. Zhang. "DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins". In: *Bioinformatics* 36.7 (2020), pp. 2105–2112 (cit. on p. 177).

[Zha97] M. Q. Zhang. "Identification of protein coding regions in the human genome by quadratic discriminant analysis". In: *Proceedings of the National Academy of Sciences* 94.2 (1997), pp. 565–568.

[Zhe+19] W. Zheng, Q. Wuyun, Y. Li, S. Mortuza, C. Zhang, R. Pearce, J. Ruan, and Y. Zhang. "Detecting distant-homology protein structures by aligning deep neural-network based contact maps". In: *PLoS computational biology* 15.10 (2019), e1007411 (cit. on p. 53).

[ZS05] Y. Zhang and J. Skolnick. "TM-align: a protein structure alignment algorithm based on the TM-score". In: *Nucleic acids research* 33.7 (2005), pp. 2302–2309 (cit. on p. 58).

[ZX12] F. Zhao and J. Xu. "A position-specific distance-dependent statistical potential for protein structure and functional study". In: *Structure* 20.6 (2012), pp. 1118–1126 (cit. on p. 57).

List of published contributions

[Dyr+19] W. Dyrka, M. Pyzik, F. Coste, and H. Talibart. "Estimating probabilistic context-free grammars for proteins using contact map constraints". In: *PeerJ* 7 (2019), e6559.

[TC20] H. Talibart and F. Coste. "ComPotts: Optimal alignment of coevolutionary models for protein sequences". In: *JOBIM 2020-Journées Ouvertes Biologie, Informatique et Mathématiques*. 2020.

[TC21] H. Talibart and F. Coste. "PPalign: Optimal alignment of Potts models representing proteins with direct coupling information". In: *BMC bioinformatics* (2021).

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Comparaison de protéines homologues avec dépendances entre positions par alignement de modèles de Potts

Mots-clés : protéines, homologie, modèle de Potts, Direct Coupling Analysis, alignement de séquences, coévolution

Résumé : Pour attribuer des annotations de structure et de fonction au nombre toujours croissant de protéines séquencées, la principale approche consiste à utiliser des méthodes de recherche d'homologues basées sur des alignements significatifs de séquences à des protéines ou familles de protéines déjà annotées. Bien que les méthodes existantes soient performantes, elles ne prennent pas en compte la co-évolution entre les résidus. Dans cette thèse, nous proposons de tirer parti d'avancées récentes dans le domaine de la prédiction de contact en représentant les protéines par des modèles de Potts, qui modélisent les couplages directs entre les positions en plus de la composition positionnelle, et de comparer les protéines en alignant ces modèles. Cette nouvelle utilisation des modèles de Potts nous a amenés à identifier de nouveaux critères pour leur construction dans un idéal de canonicité. Dû aux dépendances distantes, le problème d'alignement de deux modèles de Potts est NP-difficile. Nous avons introduit ici une méthode basée sur la formulation de l'alignement comme un problème de programmation linéaire en nombres entiers, dont la solution exacte peut être trouvée en temps raisonnable. Nos résultats suggèrent que prendre en compte les couplages directs permet d'améliorer la qualité de l'alignement d'homologues plus lointains et pourrait ainsi améliorer la détection d'homologie lointaine.

Title: Comparison of homologous protein sequences using direct coupling information by pairwise Potts model alignments

Keywords: proteins, homology, Potts model, Direct Coupling Analysis, sequence alignment, coevolution

Abstract: To assign structural and functional annotations to the ever increasing amount of sequenced proteins, the main approach relies on sequence-based homology search methods based on significant alignments of query sequences to annotated proteins or protein families. While powerful, existing approaches do not take coevolution between residues into account. Taking advantage of recent advances in the field of contact prediction, in this thesis we propose to represent proteins by Potts models, which model direct couplings between positions in addition to positional composition, and to compare proteins by aligning these models. This novel application of Potts models raised further requirements for their construction, and we identified several key points towards building more comparable Potts models, towards an ideal of canonicity. Due to non-local dependencies, the problem of aligning Potts models is NP-hard. Here, we introduced a method based on an Integer Linear Programming formulation of the problem which can be optimally solved in tractable time. Our first results suggest that taking pairwise couplings into account can improve the alignment of remote homologs and could thus improve remote homology detection.