PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École normale supérieure

Reconstructing evolution with data-driven models: From epistasis inference to mutational paths

Soutenue par
Eugenio Mauri
Le 10 Novembre 2023

École doctorale n°564
Physique en Île-de-France

Spécialité
Physique

Préparée au
Laboratoire de Physique de l'École normale supérieure

Composition du jury :

Erik VAN NIMWEGEN Universität Basel	*Président du jury* *Rapporteur*
Alessia ANNIBALE King's College London	*Rapporteuse*
Valentina ROS LPTMS Orsay, Université Paris-Saclay	*Examinatrice*
Beatriz SEOANE BARTOLOMÉ LISN, Université Paris-Saclay	*Examinatrice*
Simona COCCO LPENS	*Directrice de thèse*
Rémi MONASSON LPENS	*Directeur de thèse*

ENS | PSL | LPENS LABORATOIRE DE PHYSIQUE DE L'ÉCOLE NORMALE SUPÉRIEURE

To Yash K. Bhati

Acknowledgments

I would like to thank my supervisors, Prof. Simona Cocco and Prof. Rémi Monasson, for their guidance and support during my PhD. I am grateful to them for giving me the opportunity to work on this project, for their patience and for the many discussions we had.

I would like also to thank our collaborators, Prof. Erik Aurell, Prof. Hong-Li Zeng, Prof. Marco Ribezzi-Crivellari, Prof. Clement Nizak, Mr. Ahmed Rehan, Mr. Amaury Paveyranne and (soon to be) Dr. Vito Dichio. I am grateful to them for the many discussions we had and for their help in the development of the project.

I am also greatful to the LabEx ENS-ICFP for the financial support during my PhD.

Finally, I would like to thank the team of OpenAI for their tool ChatGPT, which proved to be very useful for the grammatical revision of this manuscript.

Per la parte più informale di questi ringraziamenti, mi rivolgerò in italiano.

Innanzitutto, vorrei ringraziare la mia famiglia per il loro supporto in tutti questi anni. Questa tesi rappresenta per me una pietra miliare; l'inizio definitivo della mia vita adulta. Spero di avervi reso orgogliosi.

Ai miei colleghi di ufficio – Andrea ed Emanuele – un ringraziamento speciale. Il dottorato (come ben sapete!) è un percorso lungo, difficile, spesso noiso ed estremamente stressante. Non minimizzo se vi dico che non sarei arrivato in fondo senza di voi. Per le lunghe discussioni e il preziosissimo cazzeggio, vi sono eternamente grato. In bocca al lupo in particolare ad Emanuele per il suo ultimo anno di dottorato (tieni duro!).

Ringrazio anche tutti i miei altri (ex)colleghi: Masha, Hugo, Jorge, Max, Mauro, Francesco, Maria Francesca e il resto del gruppo di QBio. Avete creato un ambiente di lavoro estremamente piacevole, stimolante e sicuro. Grazie mille per i bei momenti passati davanti alla macchina del caffe' e per le lunghe chiacchierate in pausa pranzo.

A i miei amici parigini: Gabriele, Angela, Ephraim, Olya, Davide, Armando, Asmaa e (di nuovo) Vito; grazie per la rete di affetto e supporto con cui mi avete circondato e protetto in questi lunghi (e al tempo stesso brevissimi) 5 anni a Parigi. Spero di essere riuscito a restiturivi almeno una frazione di tutto quello che mi avete donato.

Infine, a Margherita, va forse il ringraziamento più importante. Sei stata con me per quasi tutti questi 3 anni di dottorato. Mi hai sempre sostenuto e incoraggiato nei momenti più difficili; mi spingi a migliorare ogni volta che rimango bloccato e rinchiuso in me stesso; mi hai amato con una forza che non credevo di poter meritare. Grazie per essere la mia compagna di vita e per la felicità che mi regali. Ti amo immensamente.

Introduction

Data is one of the keywords that dominated research in the last decade. Biology and Physics are not immune to this benign 'pandemic'. The exponential growth of experimental and available data has made it possible to tackle problems and achieve results that were unthinkable only a few years ago. One of the most famous achievements is probably the development of AlphaFold [Jumper et al. (2021)], a deep learning algorithm that can predict the 3D structure of a protein from its sequence and was trained on a dataset of around 100,000 unique protein structures from Protein Data Bank (`https://www.rcsb.org/`) and all amino-acid sequences from Uniprot (`https://www.uniprot.org/`).

Models based on deep learning aim to maximize performance at the cost of interpretability (the famous 'black box' paradigm) and at a great computational cost. On the other hand, many other machine learning tools inspired by statistical physics, such as Boltzmann machines, are more interpretable and computationally cheaper. Despite the fact that they are, in principle, less performant, they can still be of great help in many contexts.

The goal of this thesis is exactly to investigate some of these potential applications. In particular, we are interested in understanding the intricate relationship between protein sequence data and evolution (and more generally for any biopolymer such as DNA and RNA). Briefly stated, our main question is: *how can we use sequence data to understand how proteins evolve?*

More precisely, we will consider two main directions of research:

- The first direction is to understand how sequence data is shaped by evolution. Evolution is usually driven by three main factors: Selection, pushing the fittest individuals (according to a given fitness function) to survive; reshuffling terms such as mutations and recombinations, introducing new genetic material in the population; and genetic drift, accounting for random fluctuations due to the finite size of the population. In this thesis, we ask ourselves in which conditions important information regarding evolution (in particular the fitness landscape) can be recovered by looking at the data resulting from this process.

- The second direction focuses on how such simple sequence-based models introduced above can be used to understand how distant proteins sharing a common evolutionary history evolved apart from each other. In other words, we are interested in finding mutational paths interpolating between different protein sequences. Despite the fact that this problem is interesting from an evolutionary point of view, it is also of great practical importance in protein engineering. Indeed, if we are able to find a path between two proteins with different functions (in such a way that also the intermediate sequences are functional), we can use it to design new proteins with intermediate functions.

Outline of the thesis

The thesis is divided into four main parts.

The first part aims to introduce all the relevant concepts and tools necessary to understand the rest of the thesis.

- Chapter 1 contains an introduction to proteins and explains the main challenges that are emerging from the statistical analysis of sequence data.
- Chapter 2 introduces the concept of energy-based models and explains how they can be used to model the genotype distribution of a protein family. In particular, we will focus on two specific models (which are based on Boltzmann machines): Direct Coupling Analysis (DCA) and Restricted Boltzmann Machines (RBM). As said above, we are interested in these models for their simple and interpretable architecture, which is discussed together with their drawbacks throughout the chapter.
- Chapter 3 presents other state-of-the-art models that can be used to model protein structure and variability. Among those, we will briefly introduce AlphaFold and ProteinMPNN [Jumper et al. (2021); Dauparas et al. (2022)], which are going to be used to benchmark some of our results in the following chapters.

The second part approaches the first main direction of research of this thesis, namely the study of the relationship between sequence data and evolution.

- Chapter 4 introduces the regime of *Quasi-linkage Equilibrium* in an evolving population, mostly by reviewing the work of Kimura (1965), Neher and Shraiman (2011a), and Zeng and Aurell (2020). In particular, we report their discussion on the emergence of QLE in the limit where recombination is much faster than selection, and mutations can be neglected. In QLE, the genotype distribution takes an asymptotic form and can be described as a Boltzmann distribution with weakly interacting loci along the genotype.
- Chapter 5 aims to study in more detail the dynamics of a large population in QLE under a Gaussian ansatz on the genotype distribution. In this approximation scheme, the evolution is completely described by the first and second-order moments of the distribution and can be applied for generic values of the rates of mutation or recombination and fitness functions. We will illustrate this approximation scheme on a short-range fitness landscape with two far-away and competing maxima. It unveils the existence of a phase transition from a broad to a polarized distribution of genomes as the strength of epistatic couplings is increased, characterized by slow coarsening dynamics of competing allele domains. Moreover, we corroborate the theoretical results through numerical simulations. This chapter is based on our following publication:

  [1] Mauri, E., Cocco, S., & Monasson, R. (2021). Gaussian closure scheme in the quasi-linkage equilibrium regime of evolving genome populations. Europhysics Letters, 132(5), 56001.
- Chapter 6 extends the results of Chapter 4 to the case where mutations and recombinations are comparable (and stronger than evolution), resulting in an updated formula to infer epistatic interactions between pairs of sites from the data distribution, which can be studied using DCA or Gaussian approximation. The performance of this procedure is tested on a numerical model of evolution, showing good results for a variety of parameters. This chapter is based on the following publication (which is the result of our collaboration with Erik Aurell's group from Stockholm University):

[2] Zeng, H. L., Mauri, E., Dichio, V., Cocco, S., Monasson, R., & Aurell, E. (2021). Inferring epistasis from genomic data with comparable mutation and outcrossing rate. Journal of Statistical Mechanics: Theory and Experiment, 2021(8), 083501.

The second main direction of research is discussed in the third part of this thesis. In particular, we will study how to obtain and characterize good mutational paths using sequence-based models (and in particular RBMs).

- Chapter 7 introduces a Monte-Carlo algorithm that is able to sample mutational paths between two homologous proteins. The goal of this procedure is to maximize the probability of all the intermediate sequences along the path, given a certain model probability distribution inferred from the data. Using RBMs trained on Lattice Proteins and WW domain families, we have shown that this algorithm is able to find good mutational paths. Possible biological interpretations of the sampled paths are discussed, and results are benchmarked using AlphaFold and ProteinMPNN. This chapter is based on our following publication:

  [3] Mauri, E., Cocco, S., & Monasson, R. (2023). Mutational paths with sequence-based models of proteins: from sampling to mean-field characterization. Physical Review Letters, 130(15), 158402.

- Chapter 8 uses mean-field theory to characterize paths for different mutational dynamics of interest. In this framework, many statistics of interest can be computed for paths that are fixed at both extremities and those that are only fixed at one end. In particular, we propose a way to estimate evolutionary distances using sequence-based epistatic models of selection. This chapter is based on [3] and also on the following publication (accepted in *Physical Review E*):

  [4] Mauri, E., Cocco, S., & Monasson, R. (2023). Transition paths in Potts-like energy landscapes: general properties and application to protein sequence models.

- Chapter 9 studies a tractable Hopfield-Potts model to unveil the existence of a phase transition separating a regime in which paths are stretched in between their anchors from another regime where paths can explore the energy landscape more globally to minimize the energy. This chapter is also based on [4].

The last part of the thesis is dedicated to the experimental validation of mutational paths. To this aim, we are currently collaborating with Marco Ribezzi's group at ESPCI and Clement Nizak's group at Sorbonne University, which are respectively developing the experimental set-ups.

- In Chapter 10, we describe the experimental set-up to measure the binding properties of designed WW domains, developed by Mr. Ahmed Rehan and Prof. Marco Ribezzi-Crivellari at ESPCI. Furthermore, we show promising preliminary results regarding the experimental validation of a mutational path between two WW domains with different binding specificities.

- In Chapter 11, we aim to validate *de novo* designed serine proteases, and in particular, to find good mutational paths between two different classes of proteases with different catalytic specificities: Trypsin-like and chymotrypsin-like proteases. After introducing the model and the numerical techniques to sample sequences and paths, we present the experimental set-up (based on a microfluidics device) currently under development

by Prof. Clement Nizak and Amaury Paveyranne at Sorbonne University. Finally, we show and discuss some preliminary results obtained by testing a few of the designed sequences using a simpler biochemical assay.

Contents

List of Figures

Résumé substantiel en français

L'objectif de cette thèse de doctorat est d'étudier la puissance des modèles basés sur les séquences entraînés sur des alignements de séquences multiples (en particulier la DCA et les RBMs) pour récupérer des informations pertinentes sur l'évolution des protéines homologues, ainsi que leur application à des problèmes de bioingénierie pertinents.

Introduction à la DCA et aux RBMs

Au début de notre thèse, nous avons discuté des principales caractéristiques de deux modèles basés sur les séquences : l'Analyse des Couplages Directs (DCA) et les Machines de Boltzmann Restreintes (RBM) [Cocco et al. (2018); Tubiana et al. (2019)]. Dans cette section, nous résumons brièvement les points principaux de ces modèles.

En général, nous partons d'un ensemble de données de séquences de protéines homologues, présentées sous forme d'alignement de séquences multiples, $\{\mathbf{v}_s\}_{s=1}^{B}$, où $\mathbf{v}_s = (v_{s,1}, v_{s,2}, \ldots, v_{s,N})$. Ensuite, notre objectif est d'inférer la distribution de probabilité $P(\mathbf{v})$ de ce modèle. Pour ce faire, nous supposons une distribution de probabilité du modèle P_{model} de la forme

$$P_{\text{model}}(\mathbf{v}) = \frac{1}{Z} e^{-E(\mathbf{v}|\boldsymbol{\theta})}, \tag{1}$$

où $E(\mathbf{v}|\boldsymbol{\theta})$ est une fonction d'énergie dépendant d'un ensemble de paramètres $\boldsymbol{\theta}$, et Z est la fonction de partition.

Maintenant, l'objectif est d'inférer l'ensemble de paramètres $\boldsymbol{\theta}$ qui décrit au mieux les données, dans le sens où ils maximisent la vraisemblance des données $\propto \prod_{s=1}^{B} P_{\text{model}}(\mathbf{v}_s|\boldsymbol{\theta}) \times P_{\text{prior}}(\boldsymbol{\theta})$. Pour maximiser cette vraisemblance, nous utilisons des techniques d'optimisation standard, discutées dans le Chapitre 2.

Analyse des Couplages Directs

Dans l'analyse des couplages directs, nous modélisons l'énergie E comme une somme d'interactions bilatérales entre les sites, plus une contribution de site unique :

$$E_{\text{DCA}}(\mathbf{v}|\boldsymbol{\theta}) = \sum_{i=1}^{N} h_i(v_i) + \sum_{i<j}^{N} J_{ij}(v_i, v_j). \tag{2}$$

Les couplages $J_{ij}(v_i, v_j)$ inférés à partir de données réelles de protéines peuvent contenir des informations pertinentes sur les contraintes évolutives agissant sur la famille de protéines. En particulier, les couplages peuvent être utilisés pour déduire les résidus en contact dans la structure de la protéine, ainsi que les interactions épistatiques entre paires de sites [Cocco et al. (2018)].

Machines de Boltzmann Restreintes

Les machines de Boltzmann restreintes sont basées sur un réseau neuronal à deux couches avec une couche d'unités visibles, $\{v_i\}_{i=1}^{N}$, en interaction avec une couche d'unités

cachées, $\{h_\mu\}_{\mu=1}^{M}$. La fonction d'énergie de la RBM est donnée par

$$E_{\mathrm{RBM}}(\mathbf{v},\mathbf{h}|\boldsymbol{\theta}) = -\sum_{i=1}^{N} g_i(v_i) - \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu) - \sum_{i=1}^{N}\sum_{\mu=1}^{M} w_{i\mu}(v_i)h_\mu, \tag{3}$$

où $g_i(v_i)$ et $\mathcal{U}_\mu(h_\mu)$ sont les champs locaux sur les unités visibles et cachées, respectivement, et $w_{i\mu}(v_i)$ sont les poids des interactions entre les unités visibles et cachées. De plus, nous définissons l'input sur l'unité cachée comme $m^\mu(\mathbf{v}) = \frac{1}{N}\sum_{i=1}^{N} w_{i\mu}(v_i)$.

Une fois que la RBM est entraînée sur des données avec une régularisation appropriée, la matrice de poids peut contenir des informations pertinentes sur d'importants groupes de sites en interaction, tant du point de vue de la structure que de la fonction de la protéine [Tubiana et al. (2019)].

Résultats sur le régime de Quasi Équilibre de Liaison

Dans la Partie II de la thèse, nous considérons une population de séquences binaires (où chaque site est un spin, $v_i = \pm 1$) évoluant sous l'effet de la mutation, de la recombinaison, de la dérive génétique (si la population est petite) et de la sélection, qui est defini par une fonction de fitness de la forme

$$F(\mathbf{v}) = \sum_{i=1}^{N} f_i(v_i) + \sum_{i<j}^{N} f_{ij}(v_i, v_j), \tag{4}$$

où f_{ij} représente l'interaction épistatique entre les sites i et j.

Nous nous concentrons sur le régime de Quasi Équilibre de Liaison (QLE), où la distribution des génotypes ressemble à une distribution de Boltzmann avec des sites faiblement interactifs (évoluant dans le temps) de la forme [Neher and Shraiman (2011b)]

$$\log P(\mathbf{v},t) \propto \sum_{i=1}^{N} h(t)_i v_i + \sum_{i<j}^{N} J(t)_{ij} v_i v_j, \tag{5}$$

Schéma de fermeture gaussienne

Dans ce régime, nous introduisons une approximation gaussienne de la distribution de probabilité sous la forme [Mauri et al. (2021)]

$$P(\mathbf{v},t) \propto e^{-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(v_i-\chi_i)\Sigma_{ij}(t)(v_j-\chi_j)}, \tag{6}$$

où χ_i représente la valeur moyenne du i-ème spin et $\Sigma_{ij}(t)$ est la matrice inverse de covariance. Nous montrons que la dynamique de la population peut être décrite par un ensemble fermé d'équations différentielles pour les moments de première et deuxième ordre de la distribution, notamment χ_i et χ_{ij}. Nous utilisons cette approximation pour étudier l'évolution d'une population dans un paysage de fitness épistatique à courte portée, révélant l'existence d'une transition de phase entre une phase non localisée et une phase où la population a tendance à s'aligner le long de deux génotypes éloignés compétitifs et également adaptés (voir le Chapitre 5).

Inférence des interactions épistatiques

Nous étendons les travaux précédents [Neher and Shraiman (2011b); Zeng and Aurell (2020)] en caractérisant l'équilibre de Quasi Équilibre de Liaison (QLE) dans une large gamme de taux de mutation et de recombinaison. En particulier, nous pouvons utiliser

la DCA ou le schéma de fermeture gaussienne décrit ci-dessus. En particulier, lorsque la mutation ou la recombinaison sont plus fortes que les pressions sélectives, les interactions épistatiques peuvent être inférées comme

$$f_{ij} = J_{ij}^{\mathrm{DCA}}(4\mu + r c_{ij}) \tag{7}$$

en utilisant les couplages appris par la DCA, ou

$$f_{ij} = \frac{\chi_{ij}}{(1-\chi_i^2)(1-\chi_j^2)}(4\mu + r c_{ij}) \tag{8}$$

dans le cadre de l'approximation gaussienne [Zeng et al. (2021)]. Ici, μ et r représentent respectivement les taux de mutation et de recombinaison, tandis que c_{ij} est la probabilité que lors de la recombinaison, les sites i et j proviennent du même parent. Nous avons corroboré ces résultats par des simulations numériques approfondies (voir le Chapitre 6).

Caractérisation des chemins mutationnels

Le deuxième objectif principal de cette thèse (voir la Partie III) est d'étudier les chemins mutationnels entre des homologues distants. En particulier, nous cherchons à définir une procédure pour identifier la meilleure succession de mutations, interpolant entre deux séquences, qui maximise la fitness le long du chemin.

Échantillonnage de chemins

Dans l'hypothèse que la fitness peut être approximée par la distribution de probabilité des séquences, inférée avec des modèles basés sur l'énergie, P_{model}, nous définissons un modèle statistique pour les chemins mutationnels dans le Chapitre 7. Dans ce modèle, la probabilité d'un chemin de T étapes, $\mathcal{V}$, allant d'une séquence $\mathbf{v}_{\mathrm{départ}}$ à une séquence $\mathbf{v}_{\mathrm{arrivée}}$ est donnée par [Mauri et al. (2023a)]

$$\begin{aligned}\mathcal{P}[\mathcal{V}|\mathbf{v}_{\mathrm{départ}}, \mathbf{v}_{\mathrm{arrivée}}] \propto \prod_{t=1}^{T-1} P_{\mathrm{model}}(\mathbf{v}_t) \times \\ \pi(\mathbf{v}_{\mathrm{départ}}, \mathbf{v}_1) \times \prod_{t=1}^{T-2} \pi(\mathbf{v}_t, \mathbf{v}_{t+1}) \times \pi(\mathbf{v}_{T-1}, \mathbf{v}_{\mathrm{arrivée}}),\end{aligned} \tag{9}$$

où le facteur de 'transition' $\pi(\mathbf{v}, \mathbf{v}')$ augmente avec la similarité entre les séquences $\mathbf{v}, \mathbf{v}'$.

Nous introduisons un algorithme de Monte-Carlo pour échantillonner des chemins mutationnels à partir de la distribution de probabilité $\propto \mathcal{P}^\beta$, pour une certaine température inverse β (voir le Chapitre 7). Ensuite, nous appliquons cette méthode pour étudier les chemins mutationnels dans des modèles de Protéines sur réseau [Shakhnovich and Gutin (1990); Jacquin et al. (2016)] et de domaines WW [Zarrinpar and Lim (2000)] entraînés avec des RBM. Notre méthode est capable de trouver de très bons chemins mutationnels et les résultats sont comparés avec des outils numériques de pointe tels que ProteinMPNN et AlphaFold [Dauparas et al. (2022); Jumper et al. (2021)].

Théorie du champ moyen

Pour mieux caractériser les chemins mutationnels, nous introduisons une théorie du champ moyen pour les chemins dans les RBM. Dans ce contexte, les seuls paramètres pertinents sont les inputs de chaque séquence le long du chemin, $\boldsymbol{m} = \{m_t^\mu\}$, et les overlaps entre séquences adjacentes le long du chemin, $\boldsymbol{q} = \{q_t^\mu\}$ (ici, $q_t = \frac{1}{N}\sum_{i=1}^N \delta_{v_{i,t}, v_{i,t+1}}$).

En écrivant $\pi(\mathbf{v},\mathbf{v}') = \exp(-N\Phi(q(\mathbf{v},\mathbf{v}')))$, la distribution de probabilité $\mathcal{P}$ est dominée par les chemins qui minimisent l'énergie libre suivante :

$$f_{\text{chemin}}(\beta,\mathbf{m},\mathbf{q}) = -\frac{1}{N}\sum_{\mu,t}\Gamma_\mu(Nm_t^\mu) + \sum_t \Phi(q_t) - \frac{1}{\beta}\mathcal{S}(\mathbf{m},\mathbf{q}) \ , \tag{10}$$

où $\Gamma_\mu(I) = \log\int \mathrm{d}h \exp\left(-\mathcal{U}_\mu(h) + Ih\right)$ et $\mathcal{S}(\mathbf{m},\mathbf{q})$ est un terme entropique (voir le Chapitre 8 pour des éclaircissements). Ici, Φ peut représenter différentes dynamiques mutationnelles. En particulier, nous considéront le cas d'un potentiel à barrière rigide (Φ_{Cont}) et un potentiel linéaire simulant l'effet de mutations avec un taux de mutation donné μ (Φ_{Evo}).

Dans ce cadre, la théorie du champ moyen peut être utilisée pour calculer de nombreuses statistiques d'intérêt, tant pour les chemins attachés aux deux extrémités que pour ceux qui n'ont qu'une seule extrémité fixée. En particulier, nous pouvons calculer la probabilité de transition entre deux séquences, en fonction du temps T, comme

$$P(\mathbf{v}\to\mathbf{v}'|T) = \frac{\sum_{\mathcal{V}:\mathbf{v}_{\text{fin}}=\mathbf{v}'} e^{-N\mathcal{E}(\mathcal{V})}}{\sum_{\mathcal{V}} e^{-N\mathcal{E}(\mathcal{V})}} \underset{N\gg 1}{\sim} \frac{e^{-Nf^{\text{const.}}_{\text{chemin}}(\mathbf{v},\mathbf{v}'|T)}}{e^{-Nf^{\text{non const.}}_{\text{chemin}}(\mathbf{v}|T)}} \ , \tag{11}$$

où $f^{\text{const.}}_{\text{chemin}}$ est l'énergie libre de l'ensemble des chemins avec la contrainte que le chemin se termine en $\mathbf{v}'$ et $f^{\text{non const.}}_{\text{chemin}}$ est l'énergie libre de l'ensemble non contraint [Mauri et al. (2023a)]. Ces résultats peuvent être utiles dans des travaux futurs pour calculer les distances évolutives entre des séquences homologues afin de construire des arbres phylogénétiques basés sur des modèles épistatiques complexes d'évolution.

Enfin, nous étudions analytiquement les propriétés des chemins dans un modèle de Hopfield-Potts Minimal (MHP) (voir le Chapitre 9). Plus précisément, nous révélons l'existence d'une transition de phase séparant un régime dans lequel les chemins sont étirés entre leurs points d'ancrage, d'un autre régime où les chemins peuvent explorer le paysage énergétique de manière plus globale pour minimiser l'énergie. Cette transition dépend notamment de la rigidité du potentiel Φ_{Cont}, de la longueur des chemins T et de la structure interne du paysage énergétique [Mauri et al. (2023b)].

Validation expérimentale des chemins

La dernière partie de cette thèse présente des résultats préliminaires visant à valider expérimentalement les chemins mutationnels dans les domaines WW et les sérine protéases.

Pour les domaines WW, nous collaborons avec le groupe de Marco Ribezzi-Crivellari (à l'ESPCI) qui travaille actuellement sur une configuration expérimentale pour tester *in vitro* l'affinité de liaison des domaines WW conçus. Les résultats (présentés dans le Chapitre 10) sont prometteurs et montrent des signes de promiscuité pour certaines des séquences intermédiaires conçues.

Pour les sérine protéases, notre objectif est de caractériser les chemins d'interpolation entre les protéines de type trypsine et les protéines de type chymotrypsine. Nous décrivons le dispositif expérimental basé sur les microfluides que le groupe de Clement Nizak (à l'Université Sorbonne) est en train de développer (les résultats obtenus avec ce dispositif seront disponibles dans les prochains mois). Dans le Chapitre 11, nous décrivons également certains résultats préliminaires obtenus en testant manuellement certaines des séquences conçues à l'aide d'un test biochimique.

An Introduction to sequence-based models of proteins

Summary

The first part of this thesis aims to lay a solid foundation on the problems and state-of-the-art tools related to the field of Machine Learning in computational biology, particularly in modeling sequence data variation. Through these initial three chapters, the reader will obtain all the necessary information to follow and understand the original results presented in the subsequent parts of this thesis.

Chapter 1 delves into open problems in protein science, highlighting the need for continued research and innovation. It begins with a quick review of proteins, providing readers with a foundational understanding of their significance and structure. Subsequently, Section 1.2 explores the main challenges faced in the era of big data, addressing potential applications of Machine Learning to solve problems like protein design for drug discovery and evolutionary analysis of co-evolving proteins.

Chapter 2 introduces energy-based models of proteins, which form the basis for understanding protein structure and function. It delves into the techniques of multi-sequence alignments of homologous proteins, Direct Coupling Analysis (DCA), and Restricted Boltzmann Machines (RBM), each contributing to our understanding of protein sequences and their associated energy landscapes.

Chapter 3 expands on the topic by exploring alternative models of proteins. It discusses notable approaches such as AlphaFold2, ProteinMPNN, Variational Autoencoders, and alignment-free models. These models provide unique perspectives and insights into protein sequence data, enabling more accurate predictions and improved understanding of protein behavior.

1

Open problems in protein science

Computational biology is one of the fastest-growing and most successful fields of theoretical research, with critical applications in medicine. Since the early 2000s, this field has seen a surge in interest among researchers in biology and theoretical science. They have come to appreciate the synergies between biological data and established computational and mathematical methods from physics and chemistry. This approach has led to the development of a quantitative biology capable of handling the enormous degree of complexity that living systems possess [Noble (2002)]. One example of this is gene expression, which is characterized by the dynamic interplay between various actors such as transcription factors, RNA polymerases, microRNAs, chromatin remodelers, histones, DNA methylation enzymes, and post-transcriptional modifiers.

The exponential growth of accessible biological data and available computational power has further accelerated the growth of computational biology in the last decade. In particular, the development of novel machine learning tools has made it possible to tackle key questions that were previously difficult or impossible to answer using traditional statistical methods. These tools have enabled researchers to analyze complex biological data, identify patterns, make predictions, and generate new hypotheses. Ultimately, this has advanced our understanding of biological systems and diseases. As a result, computational biology has become an increasingly important field with a wide range of applications in basic research, drug discovery, and personalized medicine.

This thesis will almost exclusively focus on the application of computational biology to protein science. The goal of the upcoming chapter is to provide a quick overview of proteins, their basic properties, relevant vocabulary, and their role in living systems. Secondly, we will discuss some of the main open questions in protein science that can be addressed through data-based approaches. In the following chapters of this first part of the thesis, we will review some of the most important machine learning models of proteins developed in the past decade, from direct coupling analysis [Morcos et al. (2011)] to AlphaFold [Jumper et al. (2021)] and others.

1.1. What are proteins? A quick review

Proteins can be generally defined as polymers of organic compounds called amino acids. In a simplified way, they can be compared to necklaces made of pearls, where each pearl represents a specific amino acid with a particular color or shape. Amino acids are the building blocks of protein polymers and each one consists of a central carbon atom that forms 4 covalent bonds: one with a hydrogen atom, one with a carboxyl group (-COOH), one with an amine group (-NH2) and the fourth with a generic organic group (or side chain) represented by "R". Each amino acid is characterized by a different side chain "R"

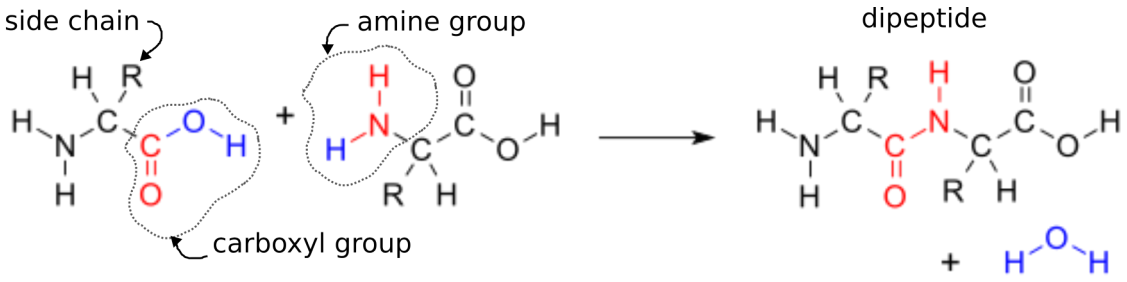


Figure 1.1: Skecth of the chemical structure of an amino acid and of the condensation reaction resulting into the formation of a dipeptide and a molecule of water.

which defines its specific chemical properties that are important to determine the function of the entire protein, as we will see later [Figure 1.1].

Every pair of amino acids binds together to form a protein polymer through a process called a peptide bond, which involves a condensation reaction between the carboxyl and the amine group of the first and second amino acids, respectively. This results in the formation of a molecule of water and a compound of two bound amino acids, also known as a dipeptide [Figure 1.1]. This process can be repeated to form a long chain that defines a specific protein.

If amino acids are the building blocks of proteins, proteins are the building blocks of life. More than 20 000 different proteins are expressed in the human body, with sometimes billions of copies per cell. Almost everything that happens inside a living cell (from structural support to biochemical reactions) involves proteins in one way or another. Proteins are involved in processes such as signal transduction, cell division and metabolism. They are also responsible for the physical properties of cells and tissues, such as elasticity, adhesion, and movement.

Furthermore, proteins are crucial for the immune system, as they can act as antigens that trigger the production of antibodies, and they can also serve as enzymes that catalyze chemical reactions in the body. Some proteins are involved in transport, such as hemoglobin which carries oxygen in the blood, and some proteins are hormones that regulate physiological processes in the body.

1.1.1 Classification of Amino Acids

There are over 500 types of amino acids found in nature, but only 22 of them (21 in eukaryotes) are used for protein biosynthesis within cells. The chemical properties of the amino-acid side chain, which is not involved in peptide bonding, determine the unique characteristics of each amino acid. These properties lead amino acids to interact with each other in different ways beyond the peptide bond that defines the amino acid chain, also known as the primary structure.

Using the analogy of a protein as a necklace made up of different pearls (amino acids), these interactions cause the chain to fold into a specific three-dimensional structure. Within this conformation, each pair of amino acids that do not share a peptide bond, but rather interact through their specific side chains, are placed close together. This 3D structure also determines how the protein interacts with other proteins or molecules, creating chemical bonds between the amino acids exposed on the protein surface and the target molecule. In the following sections, we will investigate how proteins fold into these structures in greater detail.

To classify the 21 proteinogenic amino acids found in eukaryotes based on their chemical

properties, we present the following scheme:

1. **Hydrophobic amino acids:** These amino acids have non-polar side chains that do not interact well with water. They tend to be buried inside the protein, away from the surrounding aqueous environment. The hydrophobic amino acids are alanine (Ala, A), isoleucine (Ile, I), leucine (Leu, L), methionine (Met, M), phenylalanine (Phe, F), proline (Pro, P), tryptophan (Trp, W), and valine (Val, V).

2. **Hydrophilic amino acids:** These amino acids have polar side chains that interact well with water. They tend to be exposed on the surface of the protein, interacting with the surrounding aqueous environment. The hydrophilic amino acids are asparagine (Asn, N), glutamine (Gln, Q), serine (Ser, S), and threonine (Thr, T).

3. **Acidic amino acids:** These amino acids have acidic side chains that are negatively charged at physiological pH. They tend to be negatively charged at the pH of the cell and can form ionic bonds with positively charged amino acids. The acidic amino acids are aspartic acid (Asp, D) and glutamic acid (Glu, E).

4. **Basic amino acids:** These amino acids have basic side chains that are positively charged at physiological pH. They tend to be positively charged at the pH of the cell and can form ionic bonds with negatively charged amino acids. The basic amino acids are arginine (Arg, R), histidine (His, H), and lysine (Lys, K).

5. **Special amino acids:** These amino acids have unique properties that make them important in protein structure and function. Cysteine (Cys, C) has a thiol group (-SH) that can form disulfide bonds, which help stabilize protein structure. Glycine (Gly) is the smallest amino acid and can occupy tight spaces in protein structures. Finally, the amino acid proline (Pro) has a unique cyclic structure that limits its conformational flexibility, making it important for protein structure and stability. Selenocysteine (Sec, U) is similar to cysteine, but replaces the sulfur atom in the thiol group with an atom of selenium (Se). Although this amino acid can have important roles in some proteins, it is rarely used in most proteins. For this reason, in the rest of the thesis, we will only consider the first 20 amino acids that we described above. Selenocysteine is found in some proteins, called selenoproteins, which are involved in a variety of cellular processes, including redox signaling, antioxidant defense, and thyroid hormone metabolism. However, the incorporation of selenocysteine into proteins is a complex process that requires specific machinery and can only occur in organisms that have evolved this capability.

Amino acids can interact with each other in a variety of ways, including hydrogen bonding, ionic bonding, and hydrophobic interactions. As said above, these interactions play a critical role in determining the structure and function of proteins. For example, hydrophobic amino acids tend to cluster together in the interior of the protein, while hydrophilic amino acids tend to be exposed on the surface. Amino acids with opposite charges can form ionic bonds, and amino acids with polar side chains can form hydrogen bonds with each other.

1.1.2 Protein biosynthesis in the cell

How are proteins synthesized inside the cell? This section is intended to give an overview of the fundamental processes that lead to the production of functional proteins.

Everything starts from Deoxyribonucleic acid (DNA), which is present inside each cell and contains all the information necessary for the cell to function. DNA is a polymer

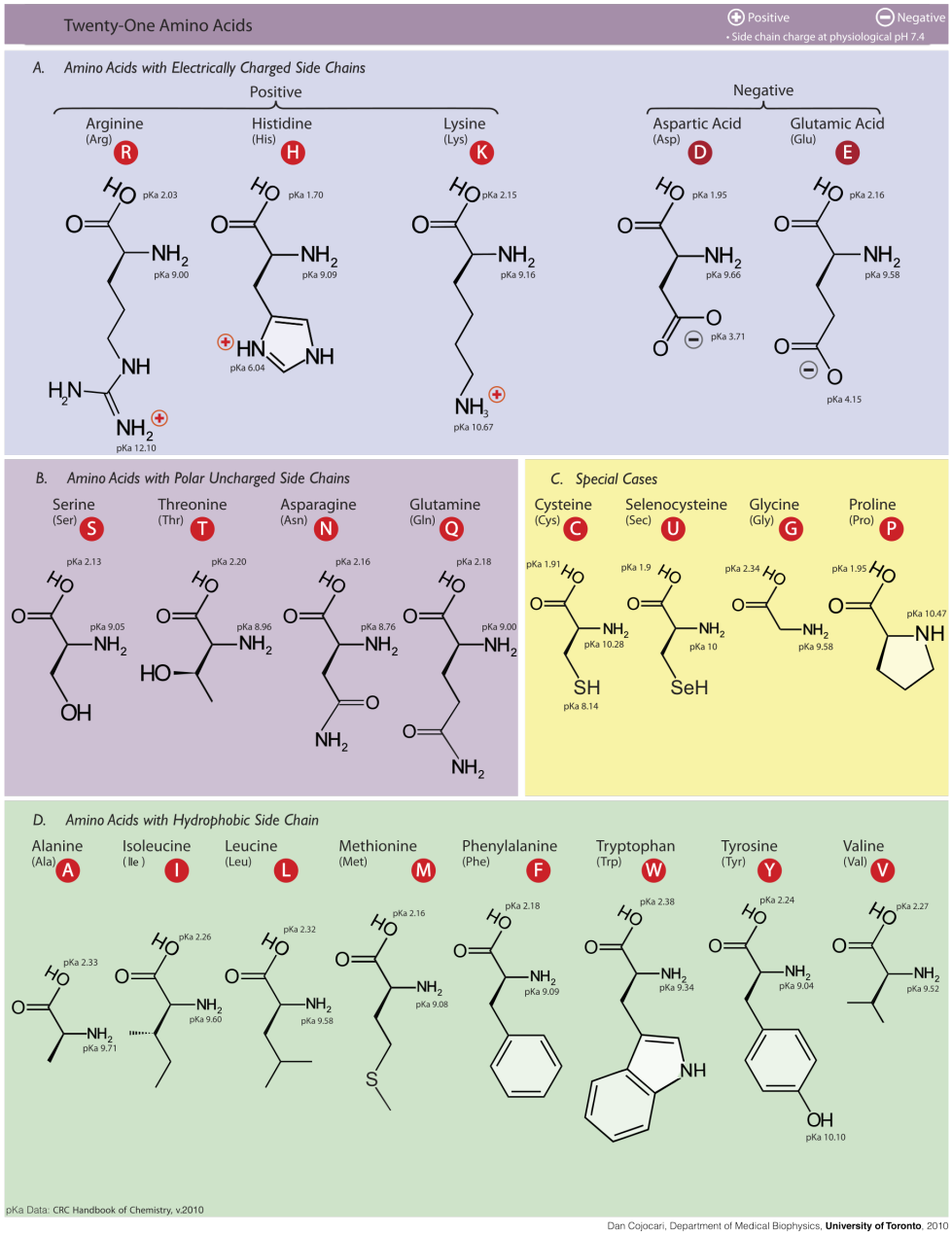


Figure 1.2: List of all 21 proteinogenomic amino acids divided accordingly to their chemical properties. Image credit: "Amino Acids" by Dan Cojocari, University of Toronto (CC BY 3.0).

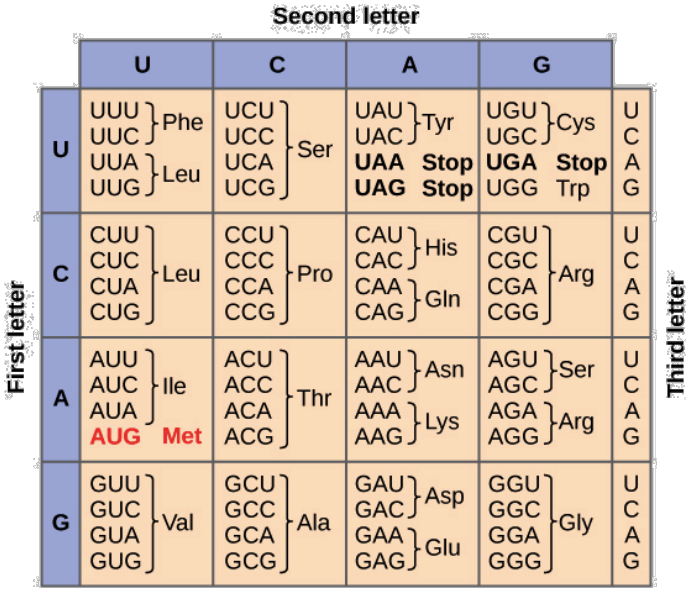


Figure 1.3: Table representing the standard genetic code, mapping each codon to a different amino acid (apart from the "stop" codon which terminates translation). The "AUG" codon encodes for both Methionine (Met) and the start codon. Image credit: "The genetic code" by OpenStax College, Biology (CC BY 3.0).

composed of nucleic acids. There are four different nucleic acids in DNA: adenine (A), thymine (T), cytosine (C), and guanine (G). DNA is assembled in two long strands of nucleic acids that bond to each other in a famous double helix shape. The two strands are complementary to each other, with the nucleic acids of one chain bonding to the nucleic acids of the other chain, forming a base pair (A with T and C with G). Among much information, DNA carries the instructions to build the proteins which are contained in a different region called *genes*.

Every protein is encoded by a specific gene present in the DNA. One of the DNA strands of this gene is first transcribed into RNA by an RNA polymerase (another protein). RNA is a polymer composed of the same nucleic acids as DNA, except that thymine is replaced by uracil (U). Synthesized RNA can undergo a function of their own or be an intermediate product for the synthesis of proteins called messenger RNA (mRNA). The mRNA is then brought into the cytoplasm after removing the non-coding part (a process called *splicing*). Once the mRNA molecule is in the cytoplasm, it binds to ribosomes, which are large complexes of proteins and RNA molecules that carry out the process of protein synthesis. The ribosome reads the sequence of nucleotides in the mRNA in groups of three, called codons, and matches each codon with a corresponding amino acid [Figure 1.3]. The map between codons and amino acids is called the genetic code, and it is highly conserved in almost all living systems, although some variant codes exist, such as the one used by mitochondria.

The ribosome then brings in the appropriate amino acid by recruiting a molecule called a transfer RNA (tRNA), which carries the matching amino acid and has a corresponding anticodon that binds to the codon on the mRNA. The ribosome then catalyzes the formation of a peptide bond between the amino acid carried by the tRNA and the growing protein chain. This process continues until the ribosome reaches a stop codon, which signals the end of protein synthesis. The completed protein chain is then released from the ribosome and undergoes further modifications, such as folding and post-translational modifications,

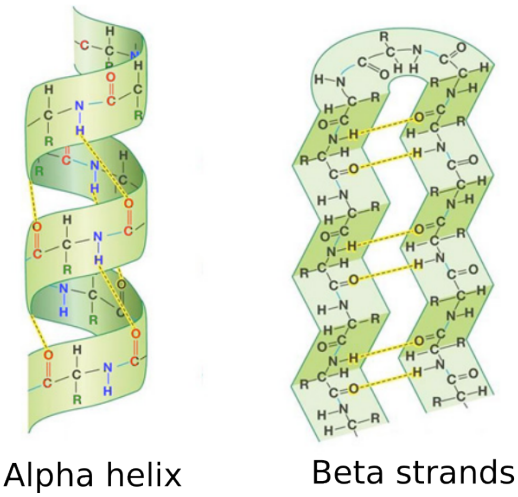


Figure 1.4: Sketch of α-helices and a β-sheets. Backbone-backbone hydrogen bonds that stabilize the structures are highlighted. (Source: cbm.msoe.edu)

to become a functional protein.

It's worth noting that the process of protein synthesis is highly regulated and complex, involving multiple steps and factors that ensure accurate and efficient translation of the genetic code. Any errors or mutations in this process can have severe consequences, such as the production of non-functional or harmful proteins, which can lead to diseases.

1.1.3 The 3-Dimensional Structure of Proteins

As mentioned in Subsection 1.1.1, proteins fold into specific 3-dimensional structures based on the interactions between their amino acids. Understanding how a protein will fold into a specific structure has been a central problem in biology since the 50's/60's, with the development of X-ray crystallography to predict the structure of myglobin [Kendrew et al. (1958)] and Ramachandran's work in 1963 [Ramachandran et al. (1963)]. The structure of a protein is essential for its functionality, and misfolded configurations can have deleterious consequences for the functioning of the entire system [Dobson (2003)].

Protein folding can be viewed from a physicist's perspective as a free energy minimization problem, as described by Anfinsen in 1973 [Anfinsen (1973)]. The free energy can be minimized by introducing more stable hydrogen bonds or by placing hydrophobic amino acids in the interior of the protein. This process leads to thermodynamically favorable structures that are unique (not challenged by a structure with similar energy), stable (resistant to interactions with the environment), and accessible (there is a minimization path from the unfolded state to the folded state).

Efforts over the past 50 years have led to the identification of four hierarchical levels that define the global structure of a protein. These levels are reviewed below:

1. **Primary structure**. This level refers to the linear sequence of amino acids that make up the protein, which is determined by the genetic code present in the DNA. The sequence is read starting from the amino acid with the free amine group (N-terminus) to the amino acid with a free carboxyl group (C-terminus). This reading direction is consistent with the way ribosomes translate mRNA during protein synthesis. At the early stage of translation, the protein sequence is unfolded, and as more amino acids are added to the growing chain, it begins to fold into its native conformation.

2. **Secondary structure**. This is the first actual structural level and it consists of a local rearrangement of close amino acids. Secondary structure is the first kind of arrangements appearing kinetically during protein synthesis by ribosomes. The two most common type of secondary structures are α-helices and β-strands, which were already predicted by Linus Pauling in 1951 [Pauling and Corey (1951)] (see Figure 1.4 for a scheme of these two configurations).

   α-helices are formed by a regular arrangement of amino acids, which creates a right-handed helix with a characteristic pitch of 3.6 amino acid residues per turn. The backbone of the helix is stabilized by hydrogen bonds between the carbonyl oxygen of one amino acid and the amide hydrogen of the fourth amino acid downstream. This gives rise to a repeating pattern of hydrogen bonds, known as the α-helix hydrogen bonding pattern. The side chains of the amino acids project outward from the helix and can participate in other interactions.

   β-strands, on the other hand, are formed by extended, stretched-out segments of the polypeptide chain that are held together by hydrogen bonds between the carbonyl oxygen of one amino acid and the amide hydrogen of an adjacent amino acid. Beta strands can be parallel, with the amino-terminal end of one strand adjacent to the carboxyl-terminal end of another, or anti-parallel, with the two strands running in opposite directions. The directionality of the strands is indicated by arrows, with the arrowhead pointing towards the carboxyl-terminal end.

   In addition to α-helices and β-strands, there are other types of secondary structures that can occur in proteins, such as loops and turns. Loops connect two secondary structures and are generally less ordered than helices or strands. Turns are short, hairpin-like segments that connect two strands or a strand and a helix. They are often stabilized by hydrogen bonds and can play important roles in protein structure and function.

3. **Tertiary structure**. It refers to the folded conformation of a single polypeptide chain. Several factors contribute to the convergence of the protein towards its native fold. Secondary structure motifs, such as α-helices and β-strands, create hydrophobic and hydrophilic regions in the protein that interact with water, guiding and accelerating the folding process [Kauzmann (1959); Šali et al. (1994); Shakhnovich (1997)]. Chaperone proteins are another factor that facilitate proper folding of the protein by assisting in the formation of correct disulfide bonds and preventing aggregation [Beissinger and Buchner (1998)]. However, environmental perturbations such as changes in temperature, pH, or exposure to certain chemicals can cause misfolding, unfolding, or denaturation of the protein.

   While the Anfinsen's dogma suggests that a protein's native structure is uniquely determined by its amino acid sequence, some proteins can adopt multiple native structures [Sinclair et al. (1994)]. These proteins can switch from one structure to another depending on the environment they are exposed to [Porter and Looger (2018)]. Furthermore, some proteins may fold to a local minimum instead of a global minimum, meaning they do not adopt the most thermodynamically stable conformation. Such proteins often require an external perturbation to undergo a change in their fold, as exemplified by the KaiB protein involved in the wake-sleep cycle [Kageyama et al. (2003)].

   Determining the three-dimensional structure of a protein is in general a challenging task. Different methods have been developed to resolve protein structures, each

with its own advantages and limitations, such as accuracy or lack of information on time-dependent fluctuations.

X-ray crystallography [Woolfson and Woolfson (1997)] for example is a widely used technique that involves crystallizing the protein and analyzing the diffraction patterns of X-rays passing through the crystal. This method can provide high-resolution structures, but it requires the protein to form a high density crystal, which may be challenging for certain proteins. Additionally, the crystallization process may introduce artifacts or alter the protein's conformation which will be different from its *in vivo* state.

Another powerlful method is Nuclear Magnetic Resonance (NMR) [Wüthrich (1989)] spectroscopy. It involves analyzing the interactions between atomic nuclei and magnetic fields, providing information about distances and angles between atoms in the protein. NMR can provide valuable insights into the dynamics of proteins in solution, but it is limited by the size of the protein that can be studied.

Finally, Cryo-Electron Microscopy (Cryo-EM) [Shi et al. (2013)] is another rapidly advancing technique. It involves freezing protein samples and imaging them using an electron microscope. Cryo-EM can provide high-resolution structures, even for large protein complexes, and is particularly useful for studying flexible or heterogeneous samples. However, it requires sophisticated and extremely sensitive equipment and still requires the protein to be in a freezed state.

4. **Quaternary structure**. It is the highest level of protein structure, and it pertains to multi-chain proteins. These chains can be identical and symmetric, such as in the case of TIM-barrel [Wierenga (2001)], or they can be different chains, as seen in hemoglobin. The quaternary structure is important for protein function because it can allow for greater stability and specificity of interactions, and it can enable proteins to perform functions that would not be possible with just a single chain.

   In addition to structure, the interaction of proteins with ligands is central to their functionality. Ligands can include other proteins in the case of antibody-antigen interactions, nucleic acids such as DNA or RNA strands as seen in the CRISPR-Cas9 enzyme or ribosomes, and a variety of other molecules such as lipids and organic compounds. The thermodynamic entropy minimization spectrum is a common way to study these interactions [Brandsdal et al. (2003)]. It can determine whether the thermodynamic free energy of a protein-ligand complex is lower than the sum of the thermodynamic free energy of the components alone, and can indicate the likelihood of the formation of the complex.

1.2. Main challenges in the Big Data era

As discussed in the previous section, protein functionality arises from a complex network of interactions between the amino acids of one or multiple peptide chains. Although the fundamental chemical interactions between pairs of amino acids are simple and well understood, it has remained unclear for many years how these interactions give rise to the intricate three-dimensional structures and biochemical functions exhibited by proteins.

The protein folding problem serves as a remarkable example of this issue and highlights the immense efforts made by researchers to unravel it. Prior to the development of sufficiently powerful computers for numerical simulations, the only way to access the tertiary structure of a protein was through experimental techniques such as X-ray crystallography, nuclear magnetic resonance spectroscopy, and cryo-electron microscopy. While these

methods are highly valuable, they are expensive and time-consuming. In particular, they all rely on the crystallization of the protein at low temperatures, which proves challenging for certain proteins (such as large proteins, membrane proteins, and disordered proteins) and for large complexes.

Advancements in computational methods have made it possible to estimate the three-dimensional folding of proteins using molecular dynamics simulations aimed at minimizing the system's free energy. Two notable examples of these methods are Rosetta [Du et al. (2021a)] and FoldX [Delgado et al. (2019)], which are based on approximate force fields that drive the dynamics of the molecule. However, these approximations rely on heuristics, and the computational cost remains high due to inefficient Monte Carlo simulations. This hinders the efficient recovery of the true structure of proteins.

A significant breakthrough in this field came with the emergence of Machine Learning and Deep Learning. Methods such as RaptorX, AlphaFold, and trRosetta [Wang et al. (2017); Senior et al. (2020); Du et al. (2021b); Jumper et al. (2021)] can predict the structures of input sequences based on models trained on multiple sequence alignments and protein structures already available in the Protein Data Bank (`https://www.rcsb.org/`).

Machine learning now dominates the world of bioinformatics and computational biology. In the case of protein science, aside from predicting the tertiary structure, models trained on ever-increasing biological datasets can address various questions. In the following section, we will review some of these questions that are particularly relevant for contextualizing the original results presented in this thesis.

1.2.1 Protein data mining

When new molecules are added to the Protein Data Bank, their structures are typically determined through time-consuming crystallography experiments. This process has traditionally posed a significant bottleneck in the generation of new entries in the database. However, the introduction of AlphaFold has revolutionized the field by enabling the easy estimation of three-dimensional protein structures from primary structure information obtained through sequencing experiments. This breakthrough has facilitated the rapid expansion of the Protein Structure Database in recent years [Varadi et al. (2021)], with a substantial increase in the number of new entries. The computationally estimated protein structures provided by AlphaFold have opened up numerous possibilities for various applications, ranging from functional analysis to drug discovery.

While the estimation of tertiary structure is a crucial aspect, there are numerous other bio-chemical features that play vital roles in characterizing a protein. Experimentally measuring properties such as secondary structure, contact prediction, solubility, ligand binding, and protein-protein interactions would require an immense effort and is practically impossible for every sequenced protein in the Protein Data Bank. However, computational methods can estimate these properties, offering an invaluable resource for various purposes. For instance, these methods can be employed to pre-screen proteins in the database based on specific target features, allowing researchers to select subsets for further analysis. Alternatively, computational estimations can serve as benchmarks for assessing the properties of de novo designed proteins before conducting costly and time-consuming experimental tests.

An example of the latter point will be presented in the next part of the thesis, where the author introduces original work [Mauri et al. (2023a)]. In this study, de novo designed proteins are evaluated using ProteinMPNN [Dauparas et al. (2022)], a deep learning model based on message passing neural networks. The goal is to assess the binding capabilities of these designed proteins with natural peptides. A more detailed description of this method will be provided in the upcoming chapters, shedding light on the potential of computational

approaches in protein engineering and design.

1.2.2 Protein design and drug discovery

Another major application of Machine Learning methods is in the design of novel proteins with improved functionality for medical applications. Traditionally, protein design has relied on laborious and time-consuming experimental techniques, such as direct evolution experiments [Kuchner and Arnold (1997)], which involve iterative cycles of mutagenesis and screening to optimize protein properties. However, Machine Learning techniques have emerged as powerful tools to streamline and enhance the protein design process [Dauparas et al. (2022); Verkuil et al. (2022); Russ et al. (2020)].

Machine Learning models can learn from vast amounts of existing protein sequence and structure data to extract patterns and correlations between sequence and function. By leveraging this knowledge, these models can predict the impact of specific mutations or combinations of mutations on protein properties, such as stability, binding affinity, or enzymatic activity. This allows researchers to computationally explore and generate libraries of protein variants with desired characteristics, accelerating the design of proteins with improved functionality.

One notable area where machine learning-driven protein design has shown great promise is in drug discovery. By employing predictive models trained on large-scale datasets, researchers can screen vast libraries of potential drug candidates and prioritize those with a high likelihood of success. Machine Learning models can predict drug-target interactions, assess binding affinities, and even estimate off-target effects, aiding in the identification of potential drug leads with desired properties.

Machine Learning-based protein design approaches are particularly valuable when combined with experimental validation. The iterative cycle of computational predictions and experimental testing allows for a rapid feedback loop, refining the models and improving the accuracy of predictions. This synergy between computational and experimental methods accelerates the discovery and optimization of proteins with specific functions, such as novel enzymes, therapeutic antibodies, or protein-based therapeutics [Mariz et al. (2021)].

Moreover, Machine Learning tools have the potential to tackle the challenges associated with protein engineering and drug design. For example, they can aid in de novo protein design, where completely novel proteins are designed from scratch, by predicting stable structures and optimizing desired properties. Machine Learning models can also assist in the rational design of protein-protein interactions, enabling the engineering of protein binders or inhibitors for specific targets.

The application of Machine Learning in protein design and drug discovery has the potential to revolutionize the field by greatly reducing the time and cost associated with traditional experimental approaches. It enables the exploration of vast protein sequence and structural spaces, guiding researchers toward more efficient and targeted experimentation. However, it is important to note that experimental validation remains crucial to confirm the predicted properties and functionalities of designed proteins and drug candidates.

1.2.3 Phylogeny reconstruction with complex models of evolution

Phylogenetic analysis aims to reconstruct the evolutionary relationships among species or proteins based on their genetic sequences. These relationships are usually depicted on a tree (see Figure 1.5), where each leaf represents a single species characterized by its own genetic sequence, and the branches indicate the evolutionary distance between species. Understanding how homologous proteins (proteins with a common evolutionary origin) evolved from ancestral species under the influence of selection and mutations, along with

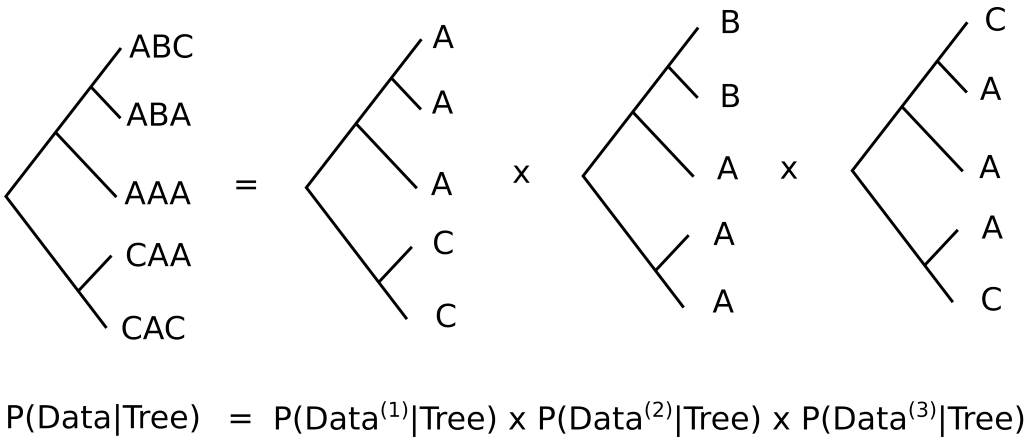


Figure 1.5: Example of a phylogenetic tree. Each species is characterised by a sequence of three letters (similar to protein or DNA sequences). Most methods computing the likelihood of the data given a specific tree assume that such probability can be decomposed as the product of the probabilities of the data at each site.

other possible effects such as recombinations, is a crucial aspect of evolutionary biology. Computational efforts to build evolutionary trees started in the 1950s and 1960s [Edwards (1963); Michener and Sokal (1957)] with the goal of understanding the target function that the reconstructed evolutionary tree needs to optimize and which algorithms are more efficient in exploring the space of possible evolutionary histories.

Around the many works done in the past 60 years, one can summarize three different classes of methods introduced to reconstruct evolutionary trees from data: parsimony methods, distance methods, and maximum likelihood methods [Felsenstein (2004)].

1. **Parsimony methods** seek to identify the evolutionary tree that requires the fewest evolutionary changes. They assume that the most likely tree is the one that minimizes the overall number of genetic changes, such as substitutions, insertions, and deletions, needed to explain the observed genetic data. This approach is based on the principle of Occam's razor, which favors the simplest explanation [Camin and Sokal (1965)].

2. **Distance methods** calculate pairwise distances between genetic sequences and use these distances to construct an evolutionary tree. These methods rely on the assumption that the more similar two sequences are, the more recently they diverged from a common ancestor. Distance methods can utilize various metrics to quantify the dissimilarity between sequences, such as Hamming distance or Jaccard coefficient [Fitch and Margoliash (1967)].

3. **Maximum likelihood methods** estimate the evolutionary tree that maximizes the probability of observing the given genetic sequences. These methods utilize probabilistic models of sequence evolution and search for the tree that best fits the observed data according to these models. Maximum likelihood methods take into account various factors, including substitution rates, nucleotide or amino acid frequencies, and branch lengths.

In particular, maximum likelihood methods assume a specific model for the evolution of the different genetic sequences. To make the computation of the likelihood of a tree feasible and for parsimony arguments, most methods rely on two fundamental assumptions (Figure 1.5) [Felsenstein (2004), Chapter 16]:

1. Evolution in different sites (on the given tree) is independent
2. Evolution in different lineages is independent

Although extremely useful for achieving computational efficiency, these assumptions are not generally true, as the effect of most mutations is strongly context-dependent, meaning they depend on the rest of the sequence, a phenomenon known as "epistasis" [Wolf et al. (2000)]. While some efforts have been made towards building context-dependent models of DNA evolution [Jensen and Pedersen (2000); Schöniger and Von Haeseler (1994)], it is still unclear how to efficiently infer phylogeny that accounts for the epistatic model of selection in general evolutionary processes. Complex models of evolution pose challenges as they often require larger amounts of training data and computational resources compared to simpler models. Additionally, interpreting the learned parameters and gaining biological insights from complex models can be more challenging.

In this context, machine learning techniques can help address the problem of efficiently dealing with complexity in phylogenetic reconstruction. Deep learning architectures, in particular, can capture complex patterns in genetic sequences and extract features relevant to phylogenetic relationships. Azer et al. (2020) proposed a two-layer fully connected neural network to infer the branching topology in the evolution of tumor cells. Alternatively, Rodríguez-Horta et al. (2022) introduced a model for ancestral sequence reconstruction that accounts for pairwise interactions between sites. Finally, Decelle et al. (2023) demonstrated that simple unsupervised models called Restricted Boltzmann Machines, which will be discussed in further detail in the following chapters, can build relational data trees by exploiting their learning dynamics.

In summary, the application of complex models of evolution in phylogenetic reconstruction, empowered by machine learning techniques, offers a promising avenue for understanding the evolutionary relationships among species or proteins. These models provide a more nuanced and accurate representation of sequence evolution, leading to improved phylogenetic reconstructions and deeper insights into the mechanisms driving evolutionary processes.

2

Energy-based models of proteins

The previous chapter provided a brief overview of the main open issues in computational protein science and the increasing significance of machine learning in addressing these challenges. The upcoming chapters will introduce significant data-based models developed over the past 15 years. We will describe the architecture and training algorithms of these models concisely. Additionally, we will explore their noteworthy applications in computational biology, emphasizing the key subjects covered by the original results presented in this thesis.

In this particular chapter, we will focus on energy-based models, a tool that has captured the interest of many physicists in recent years and led them into the field of computational biology. The main assumption of this class of methods is to treat data as independent samples from a Boltzmann distribution defined by an unknown energy landscape, which is defined as a function over the possible primary structure (*i.e.* amino acids sequence). The goal is to develop Bayesian inference algorithms that can reconstruct the most probable energy landscape and utilize this information to address relevant issues concerning the specific protein family under consideration.

We will focus in particular on two types of machine learning methods: Direct Coupling Analysis (DCA), based on Boltzmann Machines (BM), and Restricted Boltzmann Machines (RBM). We will review the main assumptions behind these models, the primary training algorithms developed in recent years, and their significant applications in the field of computational biology. Specifically, we will demonstrate how these methods have been utilized to address problems such as predicting coupling interactions, extracting patterns, and designing novel proteins [Morcos et al. (2011); Russ et al. (2020); Tubiana et al. (2019)].

Next chapters will focus on different models of proteins which are also based on tertiary structure information, rather than only on the amino acids sequence.

2.1. Multi sequence alignments of homologous proteins

Naively stated, the first step in learning a data-based model of a protein family is to collect data from that specific protein family. However, this task is not trivial. Firstly, one must extract sequences from the proteomes available in the data-set online, ensuring that only sequences corresponding to the specific family of interest are included. Secondly, the extracted homologous sequences need to be aligned to create a consistent data-set that can be further processed using Machine Learning techniques.

2.1.1 Sequence extraction and alignment

To extract the sequences corresponding to a specific protein family from the vast amount of data available online, several approaches can be employed. One common strategy is to

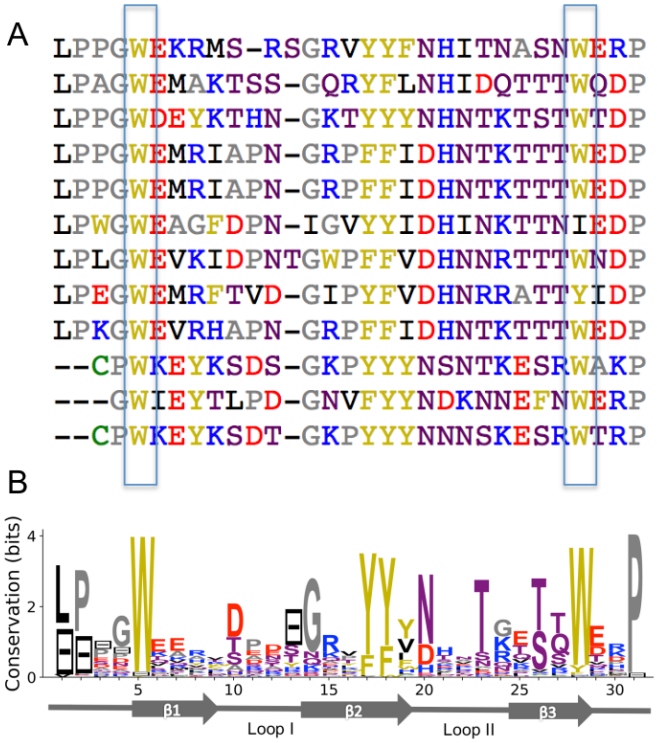


Figure 2.1: A. Multiple Sequence Alignment of the WW Domain. Rectangles highlight conserved sites. B. Corresponding Sequence Logo visualization. Image credit: Tubiana (2018b)

utilize protein sequence similarity search algorithms, such as BLAST or HMMER [Korf et al. (2003); Finn et al. (2011)], which compare the input sequence against a database and identify homologous sequences. These algorithms consider various factors, such as sequence conservation and statistical significance, to identify relevant sequences associated with the desired protein family.

Once the relevant sequences are extracted, the next challenge is to align them to create a consistent data-set. Sequence alignment involves arranging the sequences in a way that aligns the corresponding positions of similar amino acids. This process helps in identifying conserved regions and understanding the evolutionary relationships within the protein family. Multiple sequence alignment tools, such as ClustalW, MAFFT, or MUSCLE, can be utilized to perform this task [Thompson et al. (2003); Katoh et al. (2002); Edgar (2004)]. These tools employ algorithms that consider factors like pairwise sequence similarity and positional conservation to generate an alignment that optimally represents the similarities and differences among the sequences.

The final output of multi sequence alignment (MSA) is a $B \times N$ matrix where each row represents a single protein sequence of the family, $\mathbf{v}_a$ $(a = 1, \ldots, B)$, and each one of its entries, $v_{a,i}$ $(i = 1, \ldots, N)$, can have one of $A = 21$ possible states, corresponding to 20 amino acids plus a gap state (written "-") that can be introduced by the alignment algorithm to account for deletions or insertions occurred during evolution.

Figure 2.1 shows an example of a multi sequence alignment for the WW domain protein family (PFAM ID: PF00397). In particular, two particular site have been highlighted corresponding to highly conserved tryptophan (W) along the primary structure of this protein (from which the name "WW domain" takes inspiration). Another way to look at this is by looking at the site conservation which is related to the Shannon entropy based

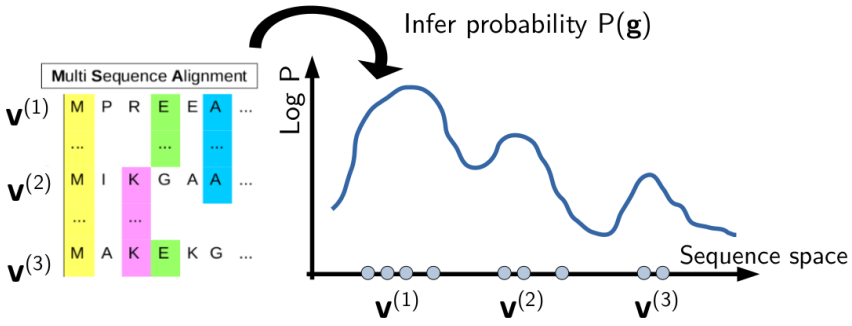


Figure 2.2: Sketch of the energy landscape reconstruction from multi sequence alignments. Regions of the sequence space with high density of data points will results with higher probability scores.

on the observed amino acids frequency (measured in "bits" of information):

$$C_i = \log 21 + \sum_{a=1}^{A} f_i(a) \log f_i(a)\,, \tag{2.1}$$

where $f_i(a)$ is the frequency of amino acid/gap a at site i. By construction, a completely not conserved site will have a conservation score $C_i = 0$ bits, while a completely conserved site will have $C_i \sim 4.4$ bits. The values of these conservation values can be graphically shown in a sequence logo as the one shown in Figure 2.1(B), where each column represents a site with total height corresponding to C_i, while the height of the letters are proportional to $f_i(a)$.

2.1.2 Sequence variability and correlations

Apart from the two highly conserved tryptophans, most of the sites have a very low conservation score, and in general, the entire MSA shows a high degree of variability (each couple of amino acids has an average identity of 30-40%). Despite this, it is also clear from the multi-sequence alignment that many sites are closely correlated. For example, the two sites after the first conserved tryptophan shown in Figure 2.1(A) always appear as a couple of amino acids with opposite charges. An important goal would be to train a model over this dataset of aligned sequences that is able to recognize and account for this kind of interaction. Stated more precisely, we can think of an MSA as a set of random samples taken from an unknown distribution of the sequence space, $P_{\text{real}}(\mathbf{v})$. Hence, the final goal would be to train a model over the MSA to infer a distribution $P_{\text{model}}(\mathbf{v})$ that best approximates the original probability landscape. To do so, we resort to Bayesian inference and Likelihood Maximization, where we propose a class of models $P_{\text{model}}(\mathbf{v}|\boldsymbol{\Theta})$ parameterized by a given set of hyperparameters $\boldsymbol{\Theta}$, and we look for the model (parameterized by a specific set $\hat{\boldsymbol{\Theta}}$) that maximizes the probability of sampling the original dataset (also called *likelihood*, L):

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \; L(\boldsymbol{\Theta}; \{\mathbf{v}_a\}_{a=1}^{B}) = \arg\max_{\boldsymbol{\Theta}} \; \frac{\prod_{a=1}^{B} P_{\text{model}}(\mathbf{v}_a|\boldsymbol{\Theta})}{P_{\text{data}}(\{\mathbf{v}_a\}_{a=1}^{B})} P_{\text{prior}}(\boldsymbol{\Theta})\,, \tag{2.2}$$

where $P_{\text{prior}}(\boldsymbol{\Theta})$ is a prior distribution over the hyperparameter space (accounting for the normalization term), and P_{data} is the total probability of observing the data (not relevant for likelihood maximization). A sketch of the procedure is given in Figure 2.2.

Using a point of view from Statistical Mechanics, we can rewrite the model probability as a Boltzmann distribution: $P_{\text{model}}(\mathbf{v}|\boldsymbol{\Theta}) = \exp(-E_{\text{model}}(\mathbf{v}|\boldsymbol{\Theta}))/Z_{\text{model}}(\boldsymbol{\Theta})$, where we

define the energy landscape E_{model} depending on the hyperparameters $\boldsymbol{\Theta}$ and the partition function Z_{model} that normalizes the probability distribution. In the next sections we will present two possible choices of energy landscapes and summarize their properties and their application to computational biology.

2.2. Direct Coupling Analysis

The first model we are going to describe to analyse protein sequence variation is called Direct Coupling Analysis (DCA) [Weigt et al. (2009); Morcos et al. (2011); Marks et al. (2011)]. The main goal behind the development of this method was to infer the pairs of sites that are in contact in the 3-dimensional structure of the protein. The idea is that if two sites are in contact, they will co-evolve together, and hence they will show a high degree of correlation in the MSA (see Figure 2.3). From a statistical point of view, this means that the two sites will have a high probability of being observed together in the dataset, and hence they will have a high probability score. In classical Statistical Physics, this problem is an extension of the inverse Ising model that aims to reconstruct the couplings between spins in a spin glass model from the knowledge of the spin configurations in the dataset [Opper and Saad (2001)]. Note that the method presented here is not the only way to infer contacting sites from multiple sequence alignments. Another possibility was presented by Burger and Van Nimwegen [Burger and Van Nimwegen (2008); Burger and van Nimwegen (2010)]. In particular, they have developed a bayesian method to infer the most likely graph of contacts between sites, where the posterior distribution can be computed usinga generalization of Kirchoff's matrix theorem [Meilă and Jaakkola (2006)].

2.2.1 The Potts Model

In Direct Coupling Analysis, the target energy landscape we want to reconstruct is the one of a Potts model [Wu (1982)]. The Potts model is a generalization of the Ising model to A-state variables, where each variable can take A different values. In the context of protein sequence analysis, the A-state variables are the amino acids, and the energy landscape is defined as:

$$E_{\text{DCA}}(\mathbf{v}|\mathbf{J},\mathbf{h}) = -\sum_{i=1}^{N} h_i(v_i) - \sum_{i<j}^{N} J_{ij}(v_i,v_j)\,, \tag{2.3}$$

where N is the number of sites in the protein, v_i is the amino acid at site i, and $\mathbf{J}$ and $\mathbf{h}$ are the parameters of the model. The first term in the energy landscape is the *local field*, and it accounts for the contribution of each site to the total energy. The second term is the *coupling term*, and it accounts for the interaction between pairs of sites. The couplings are defined as $J_{ij}(v_i,v_j) = J_{ij}(v_j,v_i)$, and they are symmetric in the two sites.

2.2.1.1 Gauge Invariance

The Potts model is invariant under a global transformation of the energy landscape, called *gauge transformation*, defined as:

$$\begin{aligned} h_i(a) &\leftarrow h_i(a) + c_i - \sum_{j\neq i}(K_{ij}(a) + K_{ji}(a)),, \\ J_{ij}(a,b) &\leftarrow J_{ij}(a,b) + K_{ij}(a) + K_{ji}(b),, \end{aligned} \tag{2.4}$$

where c_i and $K_{ij}(a)$ are arbitrary constants. Note that $K_{ij}(a)$ and $K_{ji}(b)$ are not independent, since any transformation of the form $K_{ij}(a) \leftarrow K_{ij}(a) + g$ and $K_{ji}(a) \leftarrow K_{ji}(a) - g$

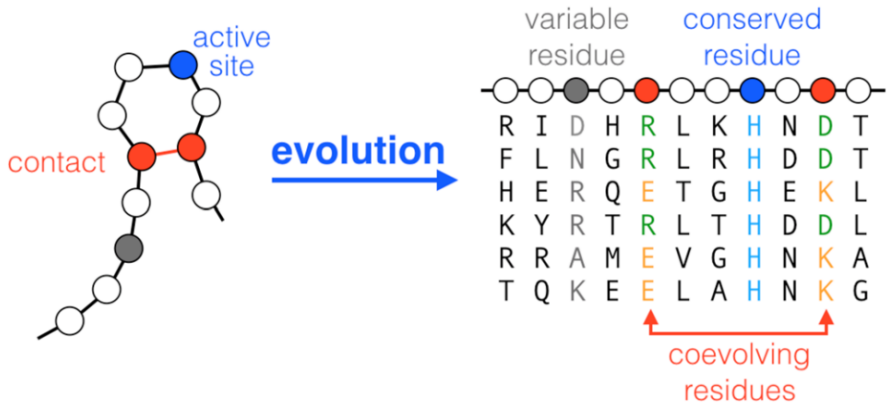


Figure 2.3: Sketch of possible evolutionary constraints shaping the variability of the protein family. While there exists constraining acting on single sites (e.g. active site relevant for the protein function), there are also constraints acting on pairs of sites (e.g. sites in contact in the 3-dimensional structure of the protein) leading to correlation between columns of the MSA. Image credit: Cocco et al. (2018).

leaves the energy landscape unchanged for any g. This leaves us with a total number of independent parameters defining the gauge transformation, N_A, equal to $N_A = N + \frac{N(N-1)}{2} \times (2A-1)$ [Cocco et al. (2018)]. This is a problem for the inference procedure since it means that the energy landscape is not uniquely defined. To solve this problem, we need to fix a gauge and impose a constraint on the parameters of the model. Two common choices are the following:

1. **Lattice Gauge**. In this gauge, for a given state q out of the possible A states per site, we fix
$$h_i(q) = J_{ij}(a,q) = J_{ij}(q,a) = 0 \text{ for all } i,j,a,. \tag{2.5}$$
This choice measures all the configurations with respect to the *empty* configuration $(q,\dots,q)$

2. **Zero-sum Gauge**. In this gauge, we fix
$$\sum_{a=1}^{A} h_i(a) = \sum_{a=1}^{A} J_{ij}(a,b) = \sum_{a=1}^{A} J_{ij}(b,a) = 0 \text{ for all } i,j,b. \tag{2.6}$$
This gauge choice is the most natural extension of the Ising model where $J_{ij}(s_i,s_j) = J_{ij}s_is_j$, $h_i(s_i) = h_is_i$, $s_{i,j} = \pm 1$. Moreover, this gauge does not break the symmetry between the A states of each site and is the most useful for the inference of contacting pairs as it will be shown in the following sections.

2.2.2 Likelihood Maximization

We now want to find the parameters of the model that maximize the likelihood of observing the data. To do so, we consider the logarithm of the likelihood function defined in Eq. (2.2), $LL = \log L$, and take its derivatives with respect to the $J_{ij}(a,b)$ and $h_i(a)$ parameters. This will require in particular to compute the derivative of the partition function of the model. We remind the reader that the partition function is defined as:

$$Z_{\text{model}}(\mathbf{J},\mathbf{h}) = \sum_{\mathbf{v}} \exp(-E_{\text{DCA}}(\mathbf{v}|\mathbf{J},\mathbf{h})) = \sum_{\mathbf{v}} \exp\left(\sum_{i=1}^{N} h_i(v_i) + \sum_{i<j}^{N} J_{ij}(v_i,v_j)\right), \tag{2.7}$$

where the sum is taken over all possible A^N configurations of the system. The derivative of the log-partition function with respect to the model's parameters can be computed using the following identities:

$$\frac{\partial \log Z_{\text{model}}(\mathbf{J},\mathbf{h})}{\partial J_{ij}(a,b)} = \sum_{\mathbf{v}} \delta_{v_i,a}\delta_{v_j,b} P_{\text{model}}(\mathbf{v}|\mathbf{J},\mathbf{h}) = \langle \delta_{v_i,a}\delta_{v_j,b}\rangle_{\text{model}} \tag{2.8}$$

$$\frac{\partial \log Z_{\text{model}}(\mathbf{J},\mathbf{h})}{\partial h_i(a)} = \sum_{\mathbf{v}} \delta_{v_i,a} P_{\text{model}}(\mathbf{v}|\mathbf{J},\mathbf{h}) = \langle \delta_{v_i,a}\rangle_{\text{model}}\,, \tag{2.9}$$

where $\langle \dots \rangle_{\text{model}}$ denotes the average over the model distribution. In particular, we observe that the derivative of the partition function with respect to the couplings $J_{ij}(a,b)$ is equal to the joint frequency of the symbols (i.e., amino acids) a and b at sites i and j in the model distribution, while the derivative with respect to the local fields $h_i(a)$ is equal to the average of the symbol a at site i in the model distribution.

We now compute the derivatives of the log-likelihood function with respect to the parameters of the model. For now, we exclude the prior term, P_{prior}, from the likelihood's definition. We will discuss the effects of regularization on the inference procedure later. This leads to the following set of equations:

$$\frac{\partial LL}{\partial J_{ij}(a,b)} = \sum_{s=1}^{B} -\frac{\partial E_{\text{DCA}}}{\partial J_{ij}(a,b)} - \frac{\partial \log Z_{\text{model}}}{\partial J_{ij}(a,b)} = \langle \delta_{v_i,a}\delta_{v_j,b}\rangle_{\text{data}} - \langle \delta_{v_i,a}\delta_{v_j,b}\rangle_{\text{model}} \tag{2.10}$$

$$\frac{\partial LL}{\partial h_i(a)} = \sum_{s=1}^{B} -\frac{\partial E_{\text{DCA}}}{\partial h_i(a)} - \frac{\partial \log Z_{\text{model}}}{\partial h_i(a)} = \langle \delta_{v_i,a}\rangle_{\text{data}} - \langle \delta_{v_i,a}\rangle_{\text{model}}\,, \tag{2.11}$$

where $\langle \dots \rangle_{\text{data}}$ denotes the average over the training dataset. This implies that the problem of maximizing the likelihood is equivalent to a matching problem between the statistics of the data and the model. We note that in general computing analytically the average over the model ditribution is hard. This comes from the necessity of computing the partition function Z_{model} which is given by a sum over all possible A^N configurations of the system. Below, we will discuss different approximation schemes to overcome this issue.

2.2.2.1 Not independence of the training samples

When computing the likelihood function in Eq. (2.2), we assumed that the samples in the training dataset are independent. However, this is not the case for the sequences in the MSA, since they are not independent samples from the same distribution. In fact, the homologous sequences in the MSA shared a common ancestry, which can be relatively recent for same entries of the training set. If the sequences in the data-set did not have enough time to evolve independently, this will result in many of them being atypically similar to each other. Even though this correlation is useful in the context of phylogeny reconstruction, it introduces a bias which does not depend on the function of the protein. Another source of bias is the fact that the sequences in the MSA are not sampled uniformly from the distribution of the protein family. In fact, model species or pathogens are often over-represented in the data-set, while other species without direct scientific interest are under-represented. To account for this bias, we modify the statistics of the data by re-weighting each samples as follows:

$$\langle \delta_{v_i,a} \rangle_{\text{data}} \leftarrow \frac{1}{B_{\text{eff}}} \sum_{s=1}^{B} w_s \delta_{v_i^s,a} \,, \tag{2.12}$$

$$\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} \leftarrow \frac{1}{B_{\text{eff}}} \sum_{s=1}^{B} w_s \delta_{v_i^s,a} \delta_{v_j^s,b} \,, \tag{2.13}$$

where w_s is the weight of the s-th sequence in the MSA, and $B_{\text{eff}} = \sum_{s=1}^{B} w_s$ is the effective number of sequences in the data-set. The weights are usually chosen to reduce the contribution of similar sequences in the MSA. A possible choice is to set w_s equal to the inverse of the number of sequences with more than 90% amino-acid identity (including itself).

2.2.2.2 Potts model as a maximum entropy distribution

Alternatively, the Potts model can be viewed as a maximum entropy distribution subject to constraints on the values of the first and second moments. In the context of statistical physics, maximum entropy models provide a framework for characterizing probability distributions that incorporate all available information while avoiding the introduction of additional assumptions.

The maximum entropy principle states that among all probability distributions consistent with the given constraints, the one with maximum entropy should be selected. The entropy S of a probability distribution measures its uncertainty or lack of information and is defined as:

$$S = -\sum_{\mathbf{v}} P(\mathbf{v}) \log P(\mathbf{v}),, \tag{2.14}$$

where $\mathbf{v}$ represents the configurations of the system, and $P(\mathbf{v})$ is the probability of observing configuration $\mathbf{v}$.

To specify the Potts model as a maximum entropy distribution, we need to impose constraints on the first and second moments. The first moment constraint ensures that the average values of the local fields and pairwise couplings match the observed data:

$$\langle \delta_{v_i,a} \rangle_{\text{model}} = \langle \delta_{v_i,a} \rangle_{\text{data}} \quad (\text{for all } i \text{ and } a) \,, \tag{2.15}$$

$$\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{model}} = \langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} \quad (\text{for all } i,j \text{ and } a,b),, \tag{2.16}$$

where $\langle \dots \rangle_{\text{model}}$ denotes the average with respect to the model distribution, and $\langle \dots \rangle_{\text{data}}$ represents the corresponding average with respect to the training data.

These constraints ensure that the our target model reproduces the observed frequencies of individual symbols at each site ($\langle \delta_{v_i,a} \rangle_{\text{model}}$) as well as the co-occurrence frequencies of pairs of symbols at different sites ($\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{model}}$).

In order to maximise the entropy subject to these constraints, we introduce Lagrange multipliers $h_i(a)$ and $J_{ij}(a,b)$ for the first and second moment constraints, respectively. Moreover, we add another Lagrange multiplier λ to account for the fact that the probability distribution has to be normalized to 1. Hence, the Lagrangian $\mathcal{L}$ of our maximisation problem is then defined as:

$$\mathcal{L} = S + \lambda \left(\sum_{\mathbf{v}} P_{\text{model}}(\mathbf{v}) - 1 \right) - \sum_{i,a} h_i(a) \left(\langle \delta_{v_i,a} \rangle_{\text{data}} - \langle \delta_{v_i,a} \rangle_{\text{model}} \right) + \\ - \sum_{i<j,a,b} J_{ij}(a,b) \left(\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} - \langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{model}} \right). \quad (2.17)$$

The maximum entropy distribution is then obtained by maximizing the Lagrangian with respect to the model probability distribution $P_{\text{model}}(\mathbf{v})$. In particular, deriving $\mathcal{L}$ with respect to $P_{\text{model}}(\mathbf{v})$, we obtain:

$$\frac{\partial \mathcal{L}}{\partial P_{\text{model}}(\mathbf{v})} = -\log P_{\text{model}}(\mathbf{v}) - 1 - \lambda - \sum_{i,a} h_i(a) \delta_{v_i,a} - \sum_{i<j,a,b} J_{ij}(a,b) \delta_{v_i,a} \delta_{v_j,b}. \quad (2.18)$$

Hence, by setting the above derivative to zero, we obtain the following expression for the maximum entropy distribution:

$$\log P_{\text{model}}(\mathbf{v}) = -1 - \lambda + \sum_i h_i(v_i) + \sum_{i<j} J_{ij}(v_i, v_j), \quad (2.19)$$

which is equivalent to the definition of the Potts model given in Eq. (2.3).

2.2.3 Regularization of the parameters

During the inference procedure, it is common to introduce regularization techniques to prevent overfitting and improve the generalization ability of the model. Following the definiton of the likelihood given in Eq. (2.2), the regularization term accounts for prior knoledge of the inferred parameter coming from the distribution $P_{\text{prior}}(\mathbf{J}, \mathbf{h})$. Two popular choices for regularization in the Potts model are L2 and L1 regularization, which impose different constraints on the parameter values.

- **L2 regularization**, also known as ridge regularization, adds a penalty term to the likelihood function based on the squared magnitudes of the parameters. This regularization term discourages large parameter values and promotes a smoother solution. The L2 regularized log-likelihood function becomes:

$$LL_{\text{L2}} = LL - \frac{1}{2} \left(\lambda_1 \sum_{i,a} h_i(a)^2 + \lambda_2 \sum_{i<j,a,b} J_{ij}(a,b)^2 \right), \quad (2.20)$$

  where λ_1 and λ_2 are the regularization parameters that control the strength of the penalty term.

  The effect of L2 regularization is to shrink the parameter values towards zero. In the case of the Potts model, this leads to smaller magnitudes of h_i and J_{ij} compared to the unregularized case. L2 regularization encourages the parameters to not be too large.

- **L1 regularization**, also known as Lasso regularization, introduces a penalty term based on the absolute values of the parameters. Similar to L2 regularization, it discourages large parameter values, but it has a more pronounced effect on driving

parameters to exactly zero, enforcing the sparsity of the inferred matrices $\mathbf{J}$ and $\mathbf{h}$. The L1 regularized log-likelihood function becomes:

$$LL_{\text{L1}} = LL - \left(\lambda_1 \sum_{i,a} |h_i(a)| + \lambda_2 \sum_{i<j,a,b} |J_{ij}(a,b)| \right). \tag{2.21}$$

As a result of L1 regularization, many of the parameter values become exactly zero, leading to a more sparse representation compared to L2 regularization. This sparsity allows for automatic feature selection, as the zero-valued parameters correspond to interactions or biases that are deemed irrelevant by the model.

- **Pseudo-count regularization**. In addition to L2 and L1 regularization, it is also common to add a pseudo-count term to the empirical frequencies and correlations of the data:

$$\langle \delta_{v_i,a} \rangle_{\text{data}} \leftarrow (1-\alpha) \langle \delta_{v_i,a} \rangle_{\text{data}} + \frac{\alpha}{A}, \tag{2.22}$$

$$\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} \leftarrow (1-\alpha) \langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} + \frac{\alpha}{A^2}. \tag{2.23}$$

This procedure is equivalent to extending the dataset by adding $\frac{B\alpha}{1-\alpha}$ sequences with amino acids sampled uniformly at all sites.

2.2.3.1 L2 vs L1 regularization

Looking at the log-likelihoods modified with L1 and L2 regularization terms in Eqs. (2.20) and (2.21), they resemble Lagrangians defined in constrained optimization problems. Specifically, we seek the optimal value of the log-likelihood LL subject to constraints on the L_2 and L_1 norms of the parameters. A surface of constant L_2 norm corresponds to a hyper-sphere of a given radius around the origin, while a surface of constant L_1 norm corresponds to a high-dimensional polyhedron with vertices placed on the orthogonal axes that define the parameter space.

In order to depict the effect of L2 and L1 regularization on the inferred parameters, we consider a toy example of a function $\ell(h_1, h_2)$ over two variables h_1 and h_2 that we want to maximize with constraints on the L_2 and L_1 norms of the vector (h_1, h_2). To solve this constrained optimization problem, we can define two Lagrangians $\mathcal{L}_{\text{L2}}$ and $\mathcal{L}_{\text{L1}}$ that correspond to the L2 and L1 regularized log-likelihoods, respectively:

$$\mathcal{L}_{\text{L2}} = \ell(h_1, h_2) - \frac{\lambda}{2}(h_1^2 + h_2^2), \tag{2.24}$$

$$\mathcal{L}_{\text{L1}} = \ell(h_1, h_2) - \lambda(|h_1| + |h_2|), \tag{2.25}$$

where λ is the regularization parameter that controls the strength of the penalty term.

An example is shown in Figure 2.4. While the L_2 regularization applies a mild penalty that smoothly pushes the optimal parameters (black dot in Figure 2.4(left)) towards the origin, the L_1 regularization applies a stronger penalty since the optimal parameters might be placed on the vertices of the L_1 polyhedron (black dot in Figure 2.4(right)) which are placed on orthogonal axes where one of the two parameters is exactly zero.

This simplified image clarifies how L_1 regularization enforces sparsity in the inferred parameters of a model. In the next section, we will discuss the practical aspects of parameter inference in the Potts model, including the optimization algorithms used to maximize the regularized log-likelihood and select appropriate regularization parameters.

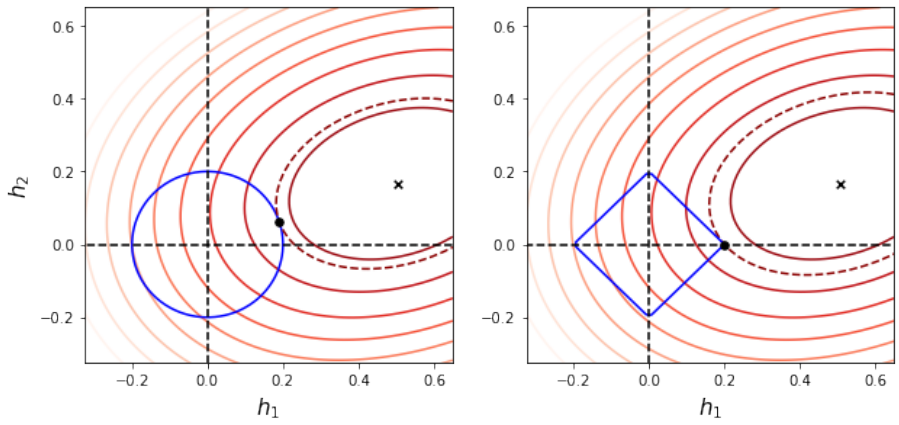


Figure 2.4: Example of L2 and L1 regularization on a toy function $\ell(h_1, h_2)$. The red lines represent the level curves of the function $\ell(h_1, h_2)$, while the blue lines represent the level curves of the L2 (*left*) and L1 (*right*) norms, respectively, which depends on the chosen value of regularization term λ. The black dots represent the optimal values of the parameters that maximize the regularized log-likelihoods. The dashed red line corresponds to the contour line of $\ell(h_1, h_2)$ that passes through the optimal parameters.

2.2.4 Training algorithms

We will now review some of the most common techniques to train the Potts model on a given MSA [Cocco et al. (2018)].

- **Boltzmann Machine learning**, is the most straightforward approach to train the Potts model by matching the first and second order statistics of the data with the ones of the model. Starting from a given set of initial parameters $\mathbf{J}$ and $\mathbf{h}$, we first sample a set of sequences from the model distribution $P_{\text{model}}(\mathbf{v}|\mathbf{J}, \mathbf{h})$ using a Monte Carlo Markov Chain (MCMC) algorithm. Then, we use these samples to estimate the first and second order statistics of the model distribution, and we update the parameters of the model using the following equations:

$$h_i(a) \leftarrow h_i(a) + \eta \left(\langle \delta_{v_i,a} \rangle_{\text{data}} - \langle \delta_{v_i,a} \rangle_{\text{model}} \right) , \tag{2.26}$$

$$J_{ij}(a,b) \leftarrow J_{ij}(a,b) + \eta \left(\langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{data}} - \langle \delta_{v_i,a} \delta_{v_j,b} \rangle_{\text{model}} \right) , \tag{2.27}$$

  where η is the learning rate that controls the size of the parameter updates. This procedure is repeated until the parameters converge to a stationary point of the log-likelihood function. The main drawback of this approach is that it is computationally expensive since it requires sampling from the model distribution at each iteration. For this reason, this method is not usually used directly to train the Potts model, but it is used to fine tune the parameters after the initial training with other, more time efficient, methods.

- **Gaussian Approximation**, aims to approximate the model distribution to a multivariate Gaussian distribution [Baldassi et al. (2014)]. In the lattice gauge, we can represent a sequence $\mathbf{v}$ using a one-hot encoding, where each amino acid is represented by a vector of length $A-1$ with all zero entries except for the one corresponding to the amino acid type (while the empty configuration of the gauge will correspond to the null vector). In this approximation, we forget that the amino acids are discrete variables and we treat space of sequences $\mathbf{v}$ as a continuous space. This allows us

to approximate the model distribution with a multivariate Gaussian distribution where the parameters $J_{ij}(a,b)$ defines the covariance matrix $\mathbf{\Sigma}$ of the distribution as $\Sigma_{ij}(a,b) = (\mathbf{J}^{-1})_{ij}(a,b)$. Defining $\boldsymbol{\mu}$ the average vector of the Gaussian distribution, we can write the log-likelihood function as:

$$LL_{\text{Gauss}} \propto -\frac{B}{2}\log\det\left(\mathbf{J}^{-1}\right) + \frac{1}{2}\sum_{s=1}^{B}\left(\mathbf{v}_s - \boldsymbol{\mu}\right)^T \mathbf{J}\left(\mathbf{v}_s - \boldsymbol{\mu}\right). \tag{2.28}$$

By defining with μ_{data} and $\mathbf{\Sigma}_{\text{data}}$ the empirical average and covariance matrix of the data, we obtain that the optimal parameters of the model are given by:

$$\boldsymbol{\mu} = \boldsymbol{\mu}_{\text{data}}, \tag{2.29}$$

$$\mathbf{J} = -\left(\mathbf{\Sigma}_{\text{data}}\right)^{-1}. \tag{2.30}$$

The main advantage of this approach is that it allows to compute the parameters of the model in a single step, without the need of iterative procedures. However, it will not converge to the correct solution if the model distribution even with infinite amount of data.

- **Pseudo-likelihood approximation**, is a method that aims to approximate the likelihood function by factorizing it into a product of conditional probabilities. In particular, we can write the log-likelihood function as:

$$\sum_{s=1}^{B} w_s \log P_{\text{model}}(\mathbf{v}_s) \to \sum_{i=1}^{N}\sum_{s=1}^{B} w_s \log P_{\text{model}}(v_i^s|\mathbf{v}_{-i}^s), \tag{2.31}$$

where w_s are the weights of the sequences in the MSA introduced in Section 2.2.2.1, while $\mathbf{v}_{-i}^s$ is the sequence $\mathbf{v}_s$ with the i-th site removed. $P_{\text{model}}(v_i^s|\mathbf{v}_{-i}^s)$ is the conditional probability at site i given the rest of the sequence and it has to be normalized by summing over all possible A states at site i. Doing so, we obtain the following expression for the conditional log-probability:

$$\log P_{\text{model}}(v_i^s|\mathbf{v}_{-i}^s) = h_i(v_i^s) + \sum_{j\neq i} J_{ij}(v_i^s, v_j^s) + \\ - \log\sum_{a=1}^{A}\exp\left(h_i(a) + \sum_{j\neq i} J_{ij}(a, v_j^s)\right). \tag{2.32}$$

Using the definition of the conditioned model defined in Eq. (2.32), we can write the average frequency of amino acid a at site i conditioned on the rest of the sequence as:

$$\mathbb{E}_i(a|\mathbf{v}_{-i}) = \frac{\exp\left(h_i(a) + \sum_{j\neq i} J_{ij}(a, v_j)\right)}{\sum_{a'=1}^{A}\exp\left(h_i(a') + \sum_{j\neq i} J_{ij}(a', v_j)\right)}. \tag{2.33}$$

We note that this average is quite easy to compute since its partition function is given by the sum over only the total number of states A. With this definition in mind, we can compute the derivatives of the log-likelihood under this approximation and optain the following set of equation for the optimal parameters of the model:

$$\langle \delta_{v_i,a} \rangle_{\text{data}} = \langle \mathbb{E}_i(a|\mathbf{v}_{-i}) \rangle_{\text{data}} \,, \tag{2.34}$$

$$2\langle \delta_{v_i,a}\delta_{v_j,b} \rangle_{\text{data}} = \langle \mathbb{E}_i(a|\mathbf{v}_{-i})\delta_{v_j,b} + \delta_{v_i,a}\mathbb{E}_j(b|\mathbf{v}_{-j}) \rangle_{\text{data}} \,. \tag{2.35}$$

Compared to the maximisation conditions given in Eqs. (2.10) and (2.11), where we have to compute the average over the model distribution (requiring at most to sum over all the A^N configurations of the system), using pseudo-likelihood approximation we only need to compute the average over the sampled data, with no need for an exponential-time alculation of the partition function. This allows parallel and efficient maximisation schemes. Although it is only an approximation, this method has been shown to be *statistically consistent*, *i.e.* it converges to the correct solution in the limit of infinite data $B \to \infty$ [Ravikumar et al. (2010)].

- **Adaptive Cluster Expansion** (ACE) [Cocco and Monasson (2012); Barton et al. (2016)], aims to expand the optimal log-likelihood of the inverse Potts model as a sum over the clusters of correlated sites in the protein. In particular, we can write:

$$LL(\{\mathbf{v}\}_{s=1}^B) = \max_{\mathbf{J},\mathbf{h}} LL(\mathbf{J},\mathbf{h}|\{\mathbf{v}\}_{s=1}^B) = \sum_{\Gamma} \Delta LL_\Gamma \,, \tag{2.36}$$

where Γ is a cluster a of sites, and ΔLL_Γ is the contribution to the log-likelihood function coming from the cluster Γ. The idea behind this procedure is that in most cases the log-likelihood function is dominated by the contribution of small clusters from which the partition function can be computed easily. The most relevant clusters can be constructed iteratively by merging smaller clusters with high overlap, and computing the contribution to the log-likelihood function at each step. The optimal parameters of the model can then be computed by maximizing the log-likelihood function over the selected clusters using iterative procedures. This methods converges to the correct solution by construction, with an increasing computational cost given by the desired precision of the solution.

2.2.5 Application to protein families

In this section, we will review two of the most important applications of the inverse Potts model to analyse families of homologous proteins. In particular, we will discuss the inference of the direct couplings between pairs of sites in the protein, in the context of protein structure prediction, and the design of new functional proteins.

- **Pairwise contact prediction**. As said before, the original goal of DCA is to infer the pairs of residues that are in contact in the tertiary structure of the protein. Inverse Potts model can be useful in this sense because it tries to infer a network of couplings that appear directly interacting (and not just correlated) and that are able to reproduce the statistics of the data. Moreover, L_1 regularization can enforce sparsity of the inferred couplings, which can be interpreted as a way to select the most relevant contacts in the protein, while filtering out the ones related to sampling noise of the training data. A common choice to score the couplings between pairs of sites is the Frobenius norm of the inferred coupling matrix $\mathbf{J}$, defined as:

$$F_{ij} = \sqrt{\sum_{a,b=1}^{A} J_{ij}(a,b)^2} \,. \tag{2.37}$$

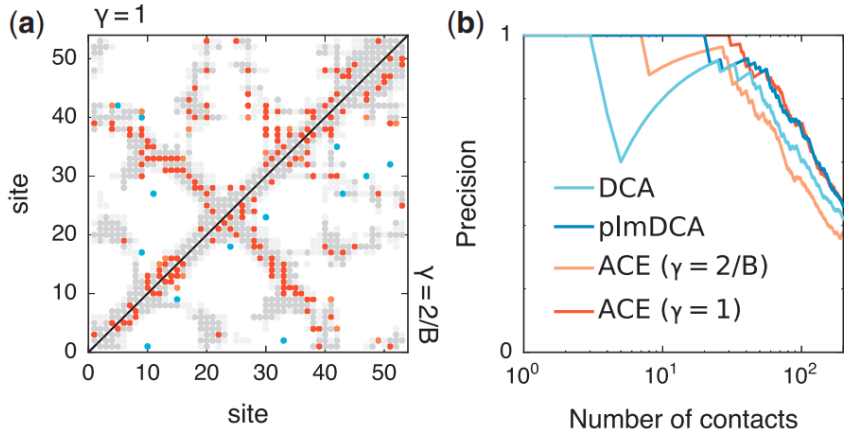


Figure 2.5: (a) Contact map for PF00014 inferred by ACE. Here, top 100 predicted contacts are shown (true predictions are indicated in orange and false predictions in blue). Other contact residues in the crystal structure are shown in dark gray for close contacts (< 6Å) and light gray for further contacts (< 8Å). Predictions for two possible L_1 regularization penalties γ are shown: $\gamma = 1$ (upper triangular part), $\gamma = 2/B$ (lower triangular part). (b) Precision (equal to the PPV shown in Eq. (2.39)) as a function of the number of contact predictions, n, of contact prediction for residues that are widely separated on the protein backbone ($|i-j| > 4$). Results using ACE are better compared with those obtained using Gaussian approximation (labeled "DCA" in the plot) [Morcos et al. (2011)] and are competitive with those from pseudo-likelihood maximisation (labeled "plmDCA") [Ekeberg et al. (2014)]. Image taken from [Barton et al. (2016)].

The Frobenius norm is gauge dependent. So, it is necessary to specify one. A sensible choice is the zero-sum gaude discussed in Section 2.2.1.1. A way to improve contact prediction is by introducing *average product correction* (APC) [Dunn et al. (2008)], which corrects the couplings between pairs of sites by subtracting the average product of the couplings of the two sites with the rest of the protein. The APC corrected Frobenius norm is defined as:

$$F_{ij}^{\mathrm{APC}} = F_{ij} - \frac{\sum_k F_{ik} \sum_k Fkj}{\sum_{k,l} F_{kl}} . \tag{2.38}$$

Once the couplings have been inferred, it is possible to predict the contacts in the protein by ranking the pairs of sites according to their corrected Frobenius norm. In particular, the top-ranked pairs of sites are predicted to be in contact in the protein. To evaluate the performance on the first n predicted contacts, it is common use the positive predictive value (PPV) defined as:

$$\mathrm{PPV}(n) = \frac{\#\text{ of true contacts in the top } n \text{ predictions}}{n} . \tag{2.39}$$

An example of the performance of DCA for contact prediction is shown in Figure 2.5, taken from [Barton et al. (2016)]. The plot compares the top-ranked contacts predicted by DCA with the true contacts in the protein, *i.e.* pairs of sites that are closer than 8Å in the tertiary structure, for the Kunitz domain (PFAM ID: PF00014).

- **Mutational effects prediction**. Another important application of the inverse Potts model is to predict the effect of mutations in the protein. Figliuzzi et al. (2016) trained a Potts model on a MSA of *Escherichia coli* TEM-1 β-lactamase and showed

that differences in the inferred energy between the wildtype and a mutant sequence correlates with the change in protein stability measures experimentally. More recent effort show the same correlation for the receptor binding domain of SARS-CoV-2 spike protein [Rodriguez-Rivas et al. (2022)].

From a theoretical point of view, other works showed that the inverse Potts model can be used as a proxy of the fitness landscape of the protein, mapping each sequence to its functionality. In particular, Jacquin et al. (2016) showed that the the energy inferred from Lattice Proteins (LP) [Shakhnovich and Gutin (1993)] is correlated with their folding probability. Moreover, Shekhar et al. (2013) managed to faithfully describe the evolution of HIV in hosts using models trained on protein data.

- ***de novo* protein design**. In the hypothesis that the Potts model can capture the essential features of the protein family [Jacquin et al. (2016)], a natural question arises: Can a Potts model trained on the data generate functional sequences that differ from those in the training set? This question has fundamental applications, as described in Section 1.2.2, as it has the potential to significantly enhance the efficiency of protein design for drug discovery. To achieve this objective, we can generate new sequences from the inferred energy landscape by utilizing Monte Carlo sampling at a specified inverse temperature β. Recent work by Russ et al. [Russ et al. (2020)] demonstrated the feasibility of this approach through experimental validation. They tested novel proteins generated from a Potts model trained on the AroQ family of Chorismate Mutases (CM), which are enzymes involved in the biosynthesis of aromatic amino acids like tyrosine (Tyr) and phenylalanine (Phe) and are found in bacteria, archaea, fungi, and plants.

  The authors trained a Potts model on a multiple sequence alignment (MSA) comprising 1259 sequences (using Boltzmann Machine learning, as described in Section 2.2.4), and subsequently generated new sequences at different temperatures. These generated sequences were then expressed in *Escherichia coli*. To evaluate the functionality of each novel variant generated by the Potts model, the authors deep sequenced the population of bacteria both before and after selection, enabling the computation of variant enrichment relative to a wildtype. Remarkably, many of the generated sequences exhibited functional activity, with some variants even surpassing the wildtype in terms of activity level. Furthermore, the average quality of the generated proteins improved when sampled at lower temperatures.

  In addition, the authors examined variants generated from an independent site model, $P_{\text{model}}(\mathbf{v}) \propto \exp(\sum_i h_i(v_i))$, trained on the same MSA. These sequences exhibited lower activity compared to those generated by the Potts model, underscoring the importance of pairwise correlations between sites in generating functional sequences.

  These findings align with the results presented in [Jacquin et al. (2016)], where the authors trained a Potts Model on sequences generated from a toy model for protein folding called Lattice Protein (LP) [Shakhnovich and Gutin (1993)]. Specifically, they demonstrated a correlation between the trained energy landscape and the native folding probability.

2.3. Restricted Boltzmann Machines

In this section, we introduce the second example of an energy-based model, known as the *Restricted Boltzmann Machine* (RBM). Unlike the Potts model used for Direct Coupling Analysis, which considers an energy landscape based on pairwise interactions

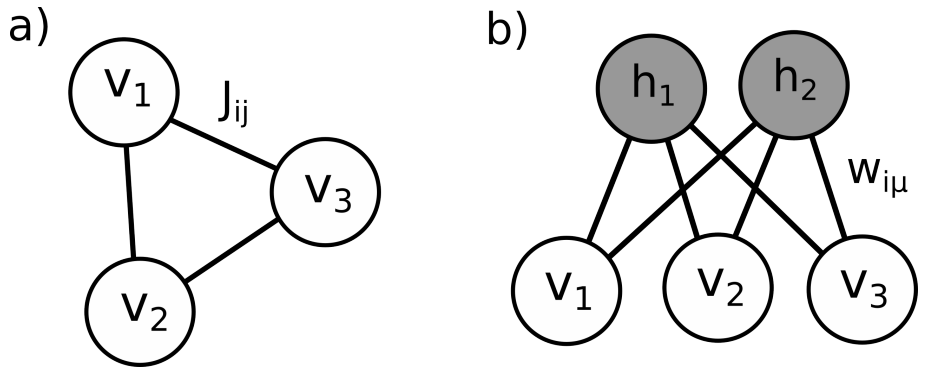


Figure 2.6: Sketch of a Potts model (a) and a Restricted Boltzmann Machine (b). In the Potts model, each couple of visible units interacts through an interaction matrix J_{ij}, while in the RBM, each visible unit interacts with all hidden units through a weight vector $w_{i\mu}$.

(along with local fields, see Figure 2.6(a)), our aim here is to define a model capable of capturing higher-order interactions. Specifically, the goal is to create an energy landscape that accounts for the activation of specific *patterns* of amino acids along the protein sequence. This bears resemblance to Hopfield networks [Hopfield (1982)], where different patterns are encoded in the weights governing the interactions between neurons [Storkey (1997)].

The significant advantage of this approach is its ability to extract patterns of amino acids, not just pairs of residues, providing insights not only into the protein's structure but also its function [Tubiana et al. (2019)]. In the following sections, we will precisely define the RBM model, discuss relevant training issues, and demonstrate its application in studying families of homologous proteins.

2.3.1 Definition of the model

Restricted Boltzmann Machines are defined as bipartite graphs with two layers of nodes. The first layer is called the *visible layer* and is composed of N nodes, each representing a different amino acid in the protein sequence. The second layer is called the *hidden layer* and is composed of M nodes, each representing the activation of a different pattern of amino acids. The joint probability distribution of the visible and hidden layers, $P_{\mathrm{RBM}}(\mathbf{v},\mathbf{h})$, is defined as

$$P_{\mathrm{RBM}}(\mathbf{v},\mathbf{h}) = \frac{1}{Z_{\mathrm{RBM}}} \exp\left(-E_{\mathrm{RBM}}(\mathbf{v},\mathbf{h})\right) \tag{2.40}$$

where Z_{RBM} is the partition function and $E_{\mathrm{RBM}}(\mathbf{v},\mathbf{h})$ is the energy function. The energy function is defined as

$$E_{\mathrm{RBM}}(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{N} g_i(v_i) - \sum_{\mu=1}^{M}\sum_{i=1}^{N} w_{i\mu}(v_i)h_\mu + \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu)\,, \tag{2.41}$$

where $g_i(v_i)$ is the local field acting on the i-th visible node, $w_{i\mu}$ is the weight of the interaction between the i-th visible node and the μ-th hidden node, and $\mathcal{U}_\mu(h_\mu)$ is the local field acting on the μ-th hidden node. See Figure 2.6(b) for a sketch of the RBM architecture.

At this level, one can define the hidden variables either as binary variables, $h_\mu \in \{0,1\}$, (with a local potential of the form $\mathcal{U}_\mu = -b_\mu h_\mu$, for a certian local field b_μ) or as continuous variables, $h_\mu \in \mathbb{R}$ (in this case $\mathcal{U}_\mu$ is an energy potential defined on the real axis). All this different schemes are discussed in Tubiana's PhD thesis [Tubiana (2018b)]. In the following

discussion, we will focus on real-valued hidden variables and *double Rectified Linear Units* (dReLU) as activation functions, defined as

$$\mathcal{U}_\mu(h) = \frac{\gamma_\mu^+}{2} h^{+2} + \theta_\mu^+ h^+ + \frac{\gamma_\mu^-}{2} h^{-2} + \theta_\mu^- h^- , \tag{2.42}$$

where $h^+ = \max(h,0)$ and $h^- = \min(h,0)$, and γ_μ^+, θ_μ^+, γ_μ^-, and θ_μ^- are the parameters of the dReLU activation function. This class of activation function is quite general and allows for a rich variety of behaviors for the hidden units.

To obtain the probability distribution over the visible units, $P_{\text{RBM}}(\mathbf{v})$, one needs to marginalize over the hidden units, *i.e.*

$$P_{\text{RBM}}(\mathbf{v}) = \int \mathrm{d}\mathbf{h}\, P_{\text{RBM}}(\mathbf{v},\mathbf{h}) = \frac{1}{Z_{\text{RBM}}} \exp\left(\sum_{i=1}^N g_i(v_i) + \sum_{\mu=1}^M \Gamma_\mu(I_\mu(\mathbf{v})) \right) , \tag{2.43}$$

where $I_\mu(\mathbf{v}) = \sum_{i=1}^N w_{i\mu}(v_i)$ is the input to the μ-th hidden unit and $\Gamma_\mu(x)$ is the cumulative generative function associated to the potential $\mathcal{U}_\mu(x)$, defined as

$$\Gamma_\mu(I) = \log \int \mathrm{d}h \exp\left(-\mathcal{U}_\mu(h) + Ih\right) . \tag{2.44}$$

In the case of dReLU activation functions, the cumulative generative function is given by

$$\Gamma_\mu(I) = \log \left[\frac{1}{\sqrt{\gamma_\mu^+}} \Phi\left(\frac{-I+\theta_\mu^+}{\sqrt{\gamma_\mu^+}} \right) + \frac{1}{\sqrt{\gamma_\mu^-}} \Phi\left(\frac{I-\theta_\mu^-}{\sqrt{\gamma_\mu^-}} \right) \right] . \tag{2.45}$$

where $\Phi(x) = \exp(x^2/2)\left[1 - \mathrm{erf}(x/\sqrt{2})\right]\sqrt{\pi/2}$.

This particular definition of the RBM is an direct extension of the well known Hopfield network model [Hopfield (1982)]. In fact, by setting the local fields g_i to zero and choosing the dReLU activation function to be a simple quadratic function with $\gamma_\mu^+ = \gamma_\mu^- = 1$ and $\theta_\mu^+ = \theta_\mu^- = 0$, one obtains that $\Gamma_\mu(I) = I^2/2 + \frac{1}{2}\log 2\pi$ and the effective energy over the visible configurations, up to an additive constant, is given by

$$E_{\text{eff}}(\mathbf{v}) = -\frac{1}{2} \sum_{\mu=1}^M \sum_{i,j=1}^N w_{i\mu}(v_i) w_{j\mu}(v_j) , \tag{2.46}$$

which is equivalent to an Hopfield model with M patterns for categorical data ($A > 2$). In this context, the weights $\mathbf{w}_\mu$ are patterns stored in the network and can be recovered by looking at the minima of the effective energy $E_{\text{eff}}(\mathbf{v})$.

2.3.1.1 RBMs as universal approximators

One of the most interesting properties of RBMs is that they are universal approximators of discrete distribution for infinitely large hidden layers. More precisely, Le Roux and Bengio (2008) derived this theorem for the case of Bernoulli-Bernoulli RBMs, *i.e.* where both hidden and visible neurons take binary values 0 or 1. In that case, the RBM energy can be written as

$$\begin{aligned} E(\mathbf{v}) = -\sum_{i=1}^N g_i v_i - \sum_{\mu=1}^M \log\left(1 + e^{c_\mu + \sum_{i=1}^N w_{i\mu} v_i}\right) \approx \\ \approx -\sum_{i=1}^N g_i v_i - \sum_{\mu=1}^M \max(0, I_\mu(\mathbf{v}) + c_\mu) , \end{aligned} \tag{2.47}$$

where $I_\mu(\mathbf{v}) = \sum_{i=1}^{N} w_{i\mu} v_i$ and the last approximation is valid when $|c_\mu + I_\mu(\mathbf{v})| \gg 1$. Suppose we want to approximate simple distribution whose support is given by a set of K configurations $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(K)}\}$ with the same probability, *i.e.* $P(\mathbf{v}) = \frac{1}{K} \sum_{k=1}^{K} \delta(\mathbf{v} - \mathbf{v}^{(k)})$. If $M = K$, we just have to choose each weight $\mathbf{w}_{i\mu}$ and biases c_μ such that the term $\max 0, I_\mu(\mathbf{v}) + c_\mu$ is non-zero if and only if $\mathbf{v} = \boldsymbol{\xi}^\mu$ (note that we are setting all visible fields to zero, $g_i = 0$). Equivalently stated, we are looking for the hyperplanes that separate the K configurations in the visible space. This procedure can be extended to the case of more complex distributions.

2.3.2 Sampling from the model

Similar to the Potts model, it is not possible to directly sample sequences from the probability distribution $P_{\text{RBM}}(\mathbf{v})$ defined in Eq. (2.43). Instead, one must employ Monte Carlo sampling techniques to generate samples from the model. In this case, the most common approach is Gibbs sampling. Gibbs sampling involves updating each unit x_l in a random order by sampling from its conditional distribution $P(x_l|\mathbf{x}_{-l})$. Gibbs sampling satisfies detailed balance, aperiodicity, and, in most cases, irreducibility, ensuring that given sufficient time, the Markov chain distribution converges to the Boltzmann distribution.

In the case of the RBM, the Gibbs sampling can be easily achieved by alternating between sampling the hidden units and the visible units. The procedure can be summarized as follows:

1. Starting from a initial configuration of the visible units, $\mathbf{v}^{(0)}$, compute the inputs to the hidden units, $I_\mu(\mathbf{v}^{(0)}) = \sum_{i=1}^{N} w_{i\mu}(v_i^{(0)})$.

2. Sample the hidden units, $\mathbf{h}^{(1)}$, from the conditional distribution $P(\mathbf{h}^{(1)}|\mathbf{v}^{(0)}) \propto \exp\left(-\sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu^{(1)}) + I_\mu h_\mu^{(1)}\right)$.

3. Compute the inputs to the visible units, $I_i(\mathbf{h}^{(1)}) = \sum_{\mu=1}^{M} h_\mu^{(1)} w_{i\mu}$. Note that I_i is a vector of length A over the alphabet of amino acids.

4. Sample the visible units, $\mathbf{v}^{(1)}$, from the conditional distribution $P(\mathbf{v}^{(1)}|\mathbf{h}^{(1)}) \propto \exp\left(\sum_{i=1}^{N} g_i(v_i^{(1)}) + I_i(v_i^{(1)})\right)$.

The first two steps can be viewed as a stochastic feature extraction from the configuration $\mathbf{v}$, while the last two steps involve a stochastic reconstruction of $\mathbf{v}$ from the features $\mathbf{h}$. In particular, one can define a data representation as the most likely hidden layer configuration given a visible layer configuration. This can be expressed as:

$$h_\mu^*(\mathbf{v}) = \arg\max_{h_\mu} P(\mathbf{h}|\mathbf{v}) = H_\mu(I_\mu(\mathbf{v})), \tag{2.48}$$

where $H_\mu(x) = (\mathcal{U}'_\mu)^{-1}(x)$ represents the inverse of the derivative of the activation function $\mathcal{U}_\mu(x)$. Another possibility is to use the average hidden layer activity given the visible layer:

$$h_\mu^*(\mathbf{v}) = \langle h_\mu|\mathbf{v}\rangle = \frac{\partial \Gamma_\mu}{\partial I}(I_\mu(\mathbf{v})), \tag{2.49}$$

where the last equality arises from the definition of the cumulant generative function. Similarly, we have $\text{Var}(h_\mu|\mathbf{v}) \equiv \partial_I^2 \Gamma_\mu(I_\mu(\mathbf{v}))$.

2.3.2.1 Annealed Importance Sampling for partition function estimation

As in the case of the Potts model, the partition function Z_{RBM} is intractable to compute exactly. However, it is possible to estimate it using Annealed Importance Sampling (AIS) [Neal (2001)]. The idea behind AIS is to define a sequence of intermediate distributions $P^{(0)}(\mathbf{v})$, $P^{(1)}(\mathbf{v})$, …, $P^{(K)}(\mathbf{v})$, where $P^{(0)}(\mathbf{v})$ is a distribution that can be sampled from easily (for example a uniform distribution) and $P^{(K)}(\mathbf{v}) = P_{\mathrm{RBM}}(\mathbf{v})$. Then the goal is to estimate ratios between partition functions of consecutive distributions. Consider two probability distributions $P^{(0)}(\mathbf{v}) = \hat{P}^{(0)}(\mathbf{v})/Z^{(0)}$ and $P^{(1)}(\mathbf{v}) = \hat{P}^{(1)}(\mathbf{v})/Z^{(1)}$, where $Z^{(0)}$ and $Z^{(1)}$ are respective partition functions. Then, we can write

$$\frac{Z^{(1)}}{Z^{(0)}} = \frac{1}{Z^{(0)}} \sum_{\mathbf{v}} \hat{P}^{(1)}(\mathbf{v}) = \frac{1}{Z^{(0)}} \sum_{\mathbf{v}} \frac{\hat{P}^{(1)}(\mathbf{v})}{\hat{P}^{(0)}(\mathbf{v})} \hat{P}^{(0)}(\mathbf{v}) = \left\langle \frac{\hat{P}^{(1)}(\mathbf{v})}{\hat{P}^{(0)}(\mathbf{v})} \right\rangle_{P^{(0)}}, \tag{2.50}$$

Hence, the ratio of partition functions can be estimated by sampling from the initial distribution $P^{(0)}(\mathbf{v})$ and computing the average ratio of the unnormalized probabilities $\hat{P}^{(1)}(\mathbf{v})/\hat{P}^{(0)}(\mathbf{v})$. Since the first partition function can be computed exactly, we can estimate iteratively all the partition function up to the target distribution $P^{(K)}(\mathbf{v}) = P_{\mathrm{RBM}}(\mathbf{v})$ using Eq. (2.50). The only remaining question is how to define the intermediate distributions $P^{(k)}(\mathbf{v})$. In fact, this method is proven to work only if each couple of subsequent distributions are close enough to each other (otherwise the method is not numerically stable). To address this issue, a standard choice is to construct a continuous path of interpolating distribution $P_\beta = (P^{(0)})^{1-\beta} P^{\beta}_{\mathrm{RBM}}$ (where $\beta \in [0,1]$) and define the intermediate distributions as $P^{(k)} = P_{\beta_k}$, where $\beta_k = k/K$. In this case, the target partition function Z_K can be estimated as a product of ratios of partition functions:

$$Z_K = Z^{(0)} \prod_{k=1}^{K} \frac{Z^{(k)}}{Z^{(k-1)}}. \tag{2.51}$$

In practice, $P^{(0)}$ is chosen to be the closest site-independent distribution to the data points. Moreover the average is computed in the logarithmic scale for numerical stability reasons. In fact, we can write $\log \frac{Z_1}{Z_0} \approx \left\langle \log \frac{\hat{P}^{(1)}}{\hat{P}^{(0)}} \right\rangle_{P^{(0)}}$ when the two distributions are close [Salakhutdinov and Murray (2008)].

2.3.3 Training the model

RBM are proven to be universal approximator for inifinitely large hidden layers [Le Roux and Bengio (2008)], $M \to \infty$. However, training RBMs with a finite number of hidden units is a challenging task. Here, we present two of the principal methods used to train RBMs and maximise the log-likelihood of the data: Contrastive Divergence (CD) [Hinton (2002)] and Persistent Contrastive Divergence (PCD) [Tieleman (2008)]. Both methods aims to compute the derivatives of the log-likelihood of the data with respect to the set of parameters $\mathbf{\Theta} = \{\mathbf{w}_\mu, \mathbf{g}, \gamma^{\pm}, \theta^{\pm}\}$ of the model. In particular this derivative takes in general the following form:

$$\frac{1}{B}\frac{\partial LL}{\partial \mathbf{\Theta}} = \left\langle \frac{\partial E_{\mathrm{RBM}}}{\partial \mathbf{\Theta}} \right\rangle_{P_{\mathrm{RBM}}} - \left\langle \frac{\partial E_{\mathrm{RBM}}}{\partial \mathbf{\Theta}} \right\rangle_{P_{\mathrm{data}}}, \tag{2.52}$$

where the first average is over the model distribution P_{RBM} and the second is over the sampled data distribution. In particular, by setting this derivative to zero, we find the

following equations for the parameters of the model (here we define a generic parameter of the potential $\mathcal{U}_\mu$ as ξ_μ):

$$\frac{1}{B}\frac{\partial LL}{\partial g_i} = \langle v_i \rangle_{P_{\text{data}}} - \langle v_i \rangle_{P_{\text{RBM}}}\,, \tag{2.53}$$

$$\frac{1}{B}\frac{\partial LL}{\partial \xi_\mu} = \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial \xi_\mu} \right\rangle_{P_{\text{data}}} - \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial \xi_\mu} \right\rangle_{P_{\text{RBM}}}\,, \tag{2.54}$$

$$\frac{1}{B}\frac{\partial LL}{\partial w_{i\mu}} = \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial I_\mu} v_i \right\rangle_{P_{\text{data}}} - \left\langle \frac{\partial \Gamma_\mu(I_\mu(\mathbf{v}))}{\partial I_\mu} v_i \right\rangle_{P_{\text{RBM}}}\,. \tag{2.55}$$

Similar to the case of the inverse Potts model, the most hard part of the training is to compute the averages over the model distribution P_{RBM}, Since it requires to have access to the partition function Z_{RBM}. In both CD and PCD, this average is approximated by sampling from the model distribution P_{RBM} as we will see below.

- **Contrastive Divergence (CD)**: In CD, we divide the training set in different batches of size N_{batch} and perform the following steps for each:

  1. Sample a batch of N_{batch} data points $\{\mathbf{v}^{(0)}\}$ from the data distribution P_{data}.
  2. Perform N_{MC} steps of Gibbs sampling starting from $\mathbf{v}^{(0)}$ to obtain a set of N_{batch} configurations $\{\mathbf{v}^{(N_{\text{MC}})}\}$.
  3. Use the sampled configurations $\{\mathbf{v}^{(N_{\text{MC}})}\}$ to compute derivatives of in Eq. (2.52) and update the parameters of the model accordingly.

  The name comes from the fact that we are computing the gradient in Eq. 2.52 as a contrast between the training data and the sampled data diverged from them. Bengio and Delalleau proved that this method is equivalent to do gradient descent over a truncated log-likelihood expansion [Bengio and Delalleau (2009)].

- **Persistent Contrastive Divergence (PCD)**: with respect to CD, in PCD we do not restart the Markov chains from the training data points at each epoch. Contrarily, we keep the configurations obtained at the end of the Gibbs sampling from the previous epoch and use them as starting points of the new Gibbs sampling. In this way, the Markov chains are initialized with configurations that are already close to the model distribution P_{RBM} and the convergence is faster. To guarantee that the method converges reasonably well, it is necessary to perform a sufficient number of Monte Carlo updates and to introduce a geometric decay on the learning rate.

Compared to classic Monte Carlo training algorithm (like BM learning presenteed in Section 2.2.4), CD and PCD are much faster and more efficient. However, they introduced biases in the training procedure. In particular, it has been recently shown how RBMs trained with CD do not replicate the statistics of the data in Eqs. (2.53)-(2.54)-(2.55) at the level of the equilibrium measure, but through a precise dynamical process [Agoritsas et al. (2023)]. However, they still provide a good approximation of the data statistics and are widely used in practice.

2.3.3.1 Regularization in RBMs

As in the case of the Potts model, it is necessary to introduce some regularization in the training procedure to avoid overfitting. In practice, we introduce L_1 regularization in the weights of the model, $w_{i\mu}$. This ensures that the weights are sparse and can capture only relevant patterns of amino acids that are present in the data.

Tubiana *et al.* have shown that L_1 regularizations stronger than the optimal one (in terms of log-probability of test set) has a low impact on the generative performance of the RBM, but it makes the weights much sparser. This is a desirable property since it allows to identify invariant patterns of amino acids across across sequences diverged during evolution [Tubiana et al. (2019)]. For enough sparse weights, the RBM is in a compositional phase in which each hidden unit encodes a limited portion of a sequence and a limited number of hidden units with strong inputs defines the representation of a sequence [Tubiana and Monasson (2017)]

On the contrary, we will add a L_2 regularization for the local fields parameters. In this case we want to avoid that the local fields are too large and the model is dominated by the local fields. In fact, in this case the model is not able to capture higher-order interactions between the amino acids.

2.3.3.2 Gauge invariance in RBMs

As in the case of the Potts model, the RBM is invariant under a gauge transformation of the weights and the local fields. In particular, if we perform the following transformation

$$w_{i\mu}(z) \to w_{i\mu}(z) + K_i\,, \qquad g_i(z) \to g_i(z) + G_i\,, \tag{2.56}$$

where K_i and G_i are arbitrary constants, the energy function $E_{\mathrm{RBM}}(\mathbf{v},\mathbf{h})$ is simply shifted by an irrelevant constant. To solve this problem, we choose a zero-sum gauge over the weights and the local fields, *i.e.* $\sum_a g_i(a) = 0$ and $\sum_a w_{i\mu}(a) = 0$. During Stochastic Gradient Descent upgrades defined above, the zero-sum gauge is preserved for the fields, but not for the weights which have to be re-centered at each epoch. In practice, this is done by subtracting the average of the weights at each epoch.

2.3.4 Feature extraction and generative power of RBMs

As stated at the beginning of the Section 2.3, the main advantage of RBMs is their ability to extract patterns of amino acids from the data that have meaningful interpretabilities in terms of protein structure and function. Tubiana *et al.* have trained RBMs on various families of homologous proteins and have shown that this class of unsupervised models are able to learn meanigful patters of the data and store them in their weights [Tubiana et al. (2019)].

For example they have trained an RBM on the family of WW domains [Zarrinpar and Lim (2000)], showing that the model is able to discriminate each sequence in the family according to its binding affinity just by looking at the inputs of a couple of hidden units. In particular, the weights associated to these hidden units have their support (*i.e.* the set of sites along the sequence with non-zero weights) localized in the two possible binding pockets of the WW domain [Sudol and Hunter (2000)], see Figure 2.7. More details about the application of RBM to study WW domains can be found in Part IV.

RBMs have been also shown to be good generative models, even though experimental validation of novel proteins is still needed. Many examples of the generative power of RBMs are given for synthetic data. For example, Tubiana *et al.* showed that RBM can generate high quality Lattice Proteins [Tubiana et al. (2019)]. Moreover, Fernandez-de-Cossio-Diaz *et al.* trained an RBM on a data set of face images using a modified training algorithm

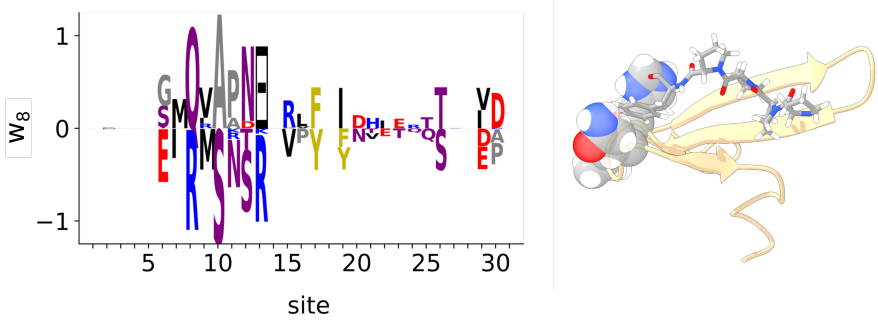


Figure 2.7: (*left*) Logo plot of one weight vector of the RBM trained on the WW domain family. the height of each letter is proportional to the value of the weight associated to that amino acid, $w_{i\mu}(a)$ (note that letters in the negative half of the graph corresponds to negative weigths). The sites with the strongest weights correspond to a possible binding pocket of the WW domain. (*right*) 3-dimensional structure of one WW domain (PDB code: 2LTW) (in yellow) binding to a peptide shown using stick representation. The molecule composing the binding pocket (corresponding to the strongly activated sites in the plotted weight) are highlighted. All the other weights of the RBM and the training details can be found in Appendix C.

that forces to store all the information regarding a feature of the image (*e.g.* "smiling/not smiling", "glasses/ no glasses") in a single hidden unit [Fernandez-de Cossio-Diaz et al. (2023)]. In this way, they were able to generate new images with the desired features.

3

Other models of proteins

In Chapter 2, we demonstrated how multiple sequence alignment can be utilized to learn a Boltzmann probability distribution, defined by an appropriate energy landscape, that maximizes the likelihood of the training data. However, in this chapter, our objective is to provide a review of state-of-the-art protein models that do not rely on the inference of energy landscapes.

Some of the models discussed here incorporate structural information into their training to address the protein folding problem. Examples of such models include AlphaFold and ProteinMPNN [Jumper et al. (2021); Dauparas et al. (2022)], which deserve an introduction since they have been used to benchmark the performance of the original methods presented in Part III and IV of this thesis.

Additionally, this chapter explores other methods based on variational autoencoders or alignment-free models. These discussions aim to enrich the overview of available tools for analyzing protein sequence data and provide a broader perspective on the field of protein modeling.

3.1. AlphaFold2

AlphaFold is probably the most remarkable tool in recent computational biology and in the field of deep learning in general. This method can predict the 3D structure of a protein from its amino acid sequence with high accuracy, managing to win the CASP14 competition in 2020 [Jumper et al. (2021)]. The CASP (Critical Assessment of protein Structure Prediction) is a biennial competition that evaluates the performance of computational methods for protein structure prediction. The CASP14 competition was the first time a method based on deep learning outperformed all other methods by a large margin. More precisely, in that competition AlphaFold structures had a median backbone accuracy of 0.96 Å r.m.s.d.$_{95}$ (Cα root-mean-square deviation at 95% residue coverage) (95% confidence interval = 0.85–1.16 Å) whereas the next best performing method had a median backbone accuracy of 2.8 Å r.m.s.d.$_{95}$ (95% confidence interval = 2.7–4.0 Å) [Jumper et al. (2021)].

The AlphaFold2 (the latest version available) model is based on a deep learning procedure that make extensive use of transformers [Vaswani et al. (2017)] and residual networks [He et al. (2016)]. The model is trained on a large dataset of protein sequences and their corresponding structures, which are obtained from the Protein Data Bank (PDB) [Berman et al. (2000)].

The inference of the 3D structure of a protein is performed in two steps. First, the model creates a MSA from the input sequence and look for available structures of similar sequences in the PDB. Then the model processes the data iteratively into a block of transformers and residual networks in order to encode the data into an abstract representation of the

aligned sequences and the residue-residue interactions. The second step uses the encoded data to predict the distances between residues and the angles of the protein backbone. Finally, the predicted distances and angles are used to reconstruct the 3D structure of the protein using a greedy algorithm.

In practice, in the following parts of this thesis, we will use AlphaFold to predict the structure of *de novo* designed proteins and compare it to the experimentally determined structure of close wildtypes. Each run of AlphaFold takes about 5-10 minutes on a single GPU and the model is available as a webserver at `https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb`. At the end of the run, the model returns a PDB file containing the predicted structure of the protein as well as a confidence score of the folding (called pLDDT). Although pLDDT gives a good indication of the quality of the predicted structure, it is not good enough to be used for phenotype estimation [McBride et al. (2022)].

3.2. ProteinMPNN

ProteinMPNN [Dauparas et al. (2022)] is a recently developped deep-learning tool whose goal is to design novel proteins with a given backbone structure. The architecture of this model is based on Messsage Passing Neural Networks [Gilmer et al. (2017)], which are designed to deal with structured data such as graphs and molecules.

In particular, ProteinMPNN looks at a protein backbone as a graph defined by the distances between the relevant atoms of each amino acids along the protein. This information is processed through a series of message passing and aggregation steps that allow information to be exchanged between connected elements in the structured data.

The key components of an MPNN typically include:

1. Node/Vertex Update: Each element (node or vertex) in the structured data initially receives an embedding or representation. In this step, the node updates its representation based on its own features and the features of its neighboring nodes.

2. Message Passing: The updated representations of the nodes are then used to compute messages that are sent from one node to its neighbors. These messages encode information about the sender node and are typically learned through trainable functions.

3. Aggregation/Pooling: The received messages are aggregated to obtain a combined representation of the neighborhood. Various aggregation functions can be used, such as summation, mean, or attention mechanisms.

4. Global Update: The aggregated neighborhood representation is then used to update the node's representation again, incorporating information from the entire neighborhood.

The output of this process is a global representation of the bakbone structure that its later decoded in order to build a probabilistic model of the protein sequence. In particular, the final output is a probability distribution, $P_{\text{MPNN}}(\mathbf{v})$, over the space of amino acid sequences, $\mathbf{v}$, measuring the likelihood of a given sequence to fold into the target structure.

The model has been trained on a large dataset of protein structures and their corresponding sequences by maximising the recovery of the target sequence.

In this thesis, we will use ProteinMPNN to evaluate the quality of designe protein sequences to bind specific peptides. More precisely, we start from a known structure (taken

from PDB) of a protein-peptide complex and use it as the input to run ProteinMPNN. The model will then return a probability distribution over the sequence space, $P_{\text{MPNN}}(\mathbf{v})$. We will then use this probability as a score for the designed sequences, *i.e.* the higher the probability, the better the sequence is expected to bind the target peptide. Note that this model can only work with sequences of a fixed length and does not accept deletions or insertions. This is a limitation of the model that will be addressed in Part III of this thesis.

3.3. Variational Autoencoder

Variational autoencoders (VAEs) [Kingma and Welling (2013)] are a class of generative models that aim to learn the underlying distribution of a dataset. In particular, VAEs are designed to learn a latent representation of the data that can be used to generate new samples from the same distribution.

VAEs are composed of two neural networks: an encoder and a decoder. The encoder takes an input sample, $\mathbf{x}$, and maps it to a latent probability distribution, $q_\phi(\mathbf{z}|\mathbf{x})$, parametrized by the weights of the encoder, ϕ. The decoder then takes a sample from the latent distribution, $\mathbf{z}$, and maps it to a reconstructed sample, $\mathbf{x}'$, parametrized by the weights of the decoder, θ. The goal of the VAE is to learn the parameters of the encoder and decoder such that the reconstructed sample, $\mathbf{x}'$, is as close as possible to the original sample, $\mathbf{x}$.

Compared to classical autoencoders, that are solely based on the reconstruction of the input data and can only be used for dimensionality reduction, VAEs are generative models that can be used to generate new samples from the same distribution as the training data.

The training of a VAE is performed by maximizing the evidence lower bound (ELBO) of the data, which is defined as:

$$\mathcal{L}(\mathbf{x};\theta,\phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\,, \tag{3.1}$$

where $p(\mathbf{z})$ is the prior distribution of the latent space and is usually set to be a standard Gaussian distribution, while $p_\theta(\mathbf{x}|\mathbf{z})$ is the likelihood of reobtaining the original data from the latent representation $\mathbf{z}$ through the decoder. The first term in Eq. 3.1 is the reconstruction loss, which measures how well the decoder can reconstruct the original data from the latent representation. The second term is the Kullback-Leibler divergence between the latent distribution and the prior distribution, which is used to regularize the latent space and prevent the model from learning a irregular representation of the data and ensure continuity in the latent space, *i.e.* similar samples in the original space should be mapped to similar samples in the latent space.

VAEs have been proven to be reliable generative models for a wide range of applications. In the context of protein design, many efforts showed the utility of VAEs and VAE-based methods to generate novel proteins. Riesselman and collaborators used deep generative model based on VAEs to predict effect of mutations on protein stability [Riesselman et al. (2018)], while Greener and collaborators used VAEs to generate novel protein folds [Greener et al. (2018)]. Another interesting work by Dean and Walper used VAEs to generate novel antimiocrobial peptides [Dean and Walper (2020)]. In particular, they linearly interpolated between two latent representations of two different peptides and showed that the generated peptides were able to maintain the antimicrobial activity of the original peptides. A final remarkable effort was made by Hawkins-Hooker and collaborators to show that a VAE is able to generate novel *luxA* proteins with measurable bioluminescence [Hawkins-Hooker et al. (2021)].

3.4. Alignment-free models

Energy-based models introduced in the last Chapter and that are going to be extensively used in the next parts of this thesis strongly rely on a multiple sequence alignment of the input data to infer an energy landscape. This alignment step introduces two sources of bias:

1. Multi sequence alignment algorithms depends on a set of hyperparameters that control the penalty over the gaps and mismatches in the alignment. These hyperparameters are usually set by the user and can introduce a bias in the alignment that will result in different energy landscapes. In order to have a good probabilitic model of the data, it is important first to find the optimal alignment of the data.

2. Energy-based models treat gaps in the same way as other amino acids. This introduces a bias in the model since in nature insertions and deletions are not as frequent as mutations. This will result in a model where insertions and deletions are not penalized enough and will be more likely to occur in the generated sequences. One can try to overcome this problem by introducing a penalty for gaps in the energy landscape, but this will require a careful tuning of the hyperparameters.

In order to overcome these problems, some efforts have been made to develop alignment-free models that do not rely on the inference of an energy landscape. A first example is DeepFam, an alignment-free deep learning method introduced by Seo and collaborators that is able to predict functional properties of proteins [Seo et al. (2018)]. The model is based on a convolutional neural network that takes as input a one-hot encoded sequence and is trained to predict the functional properties of the protein.

Another effort made by Weißenow and collaborators used develpped a free-alignemnt tool for predicting inter-residue distances [Weißenow et al. (2022)]. They do so by taking the embeddings of a pre-trained protein language model, called ProT5, from single sequences and inputting it into a convolutional neural network. Their method, called EMBER2, reaches good results (although not as good as AlphaFold2) in predicting inter-residue distances at a low computational cost.

Having reliable free-alignemnt models of protein that accurately accounts for insertion and deletions would be a great step forward in the field of protein modeling. In particular it would help for the reconstruction of evolutionary trajectories of proteins, which is one of the objective of this thesis.

II

Why DCA works? How evolution shapes the data

Summary

In Part I, we have shown how energy-based models, such as Potts models and Restricted Boltzmann Machines trained on multiple sequence alignments of homologous proteins, can be utilized to extract valuable information about the structure and function of proteins. Specifically, we have demonstrated how the energy landscape of a protein can serve as a proxy for the underlying fitness landscape, mapping each genotype (i.e., the amino-acid sequence) to its phenotype (e.g., stability, binding affinity, etc.). The primary objective of this second part of the thesis is to investigate the conditions under which the reconstruction of fitness landscapes from co-evolving sequence data is feasible.

For the sake of simplicity, we will focus on the case of populations composed of binary sequences, where each position can take one of two possible states. Such populations evolve under the combined effects of selection, which favors genotypes with higher fitness, recombination, which reshuffles genetic material, and mutations, which introduce new genetic variants.

Chapter 4 introduces the regime of Quasi-Linkage Equilibrium (QLE), wherein the population exists in a quasi-stationary monoclonal phase, and the distribution of genomes is described by an Ising-like model with weak pairwise interactions between sites. Building upon the foundational works of Kimura Kimura (1965) and Neher and Shraiman Neher and Shraiman (2011a), we demonstrate how this phase was initially characterized in the limit where recombination is the dominant evolutionary process and mutations can be neglected.

Chapter 5 extends the findings of Chapter 4 to a broader range of parameters, where mutations are no longer negligible. This extension is based on our original work Mauri et al. (2021), wherein we employ a Gaussian ansatz to describe the evolving distribution of genotypes. This approximation enables the study of evolution through a closed set of differential equations for the first and second moments of the genotype distribution. We illustrate this approximation scheme using a short-range fitness landscape featuring two competing and distantly located maxima, revealing the existence of a phase transition from a broad to a polarized distribution of genomes as the strength of epistatic couplings increases. The results obtained from the closure scheme are corroborated by numerical simulations.

Finally, in Chapter 6, we propose a formula for reconstructing pairwise epistatic effects from population data. This formula, derived from our original work in collaboration with Zeng et al. (2021), is validated by simulating the evolution of a large population of genomes across different recombination and mutation rates, as well as varying strengths of epistatic couplings. We demonstrate that the formula can accurately recover the correct epistatic coefficients in the limit of weak epistasis for a wide range of parameters.

4

Quasi-linkage equilibrium regime (QLE)

In Part I, we introduced energy-based models for analyzing families of homologous co-evolving proteins. Specifically, we discussed Direct Coupling Analysis (DCA), a tool that infers a max-entropy model from a multiple sequence alignment of homologous proteins to recover the direct couplings between amino acids. These direct couplings correspond to pairs of residues that are in contact in the 3-dimensional structure of the protein and co-evolve as a result.

The main assumption underlying this inference method is that the training set of sequences has been sampled at equilibrium from an unknown Boltzmann distribution associated with an energy landscape. If this assumption holds, we can guarantee that the inferred couplings recover the true direct couplings, and more generally, the inferred energy landscape correlates with the original one. In fact, we previously discussed the work by Jacquin and collaborators, which numerically demonstrated this result for the case of Lattice Protein [Jacquin et al. (2016)]. They used the probability of a sequence to fold into a specific structure to sample sequences at equilibrium using MCMC algorithms. Subsequently, they trained an inverse Potts model on these sequences to recover the couplings and the energy landscape. They showed that the inferred couplings correlates with the true ones, and the inferred energy landscape was close to the original one.

However, this assumption does not generally hold for real protein families. Proteins, and biological sequences in general, are not the result of Monte Carlo dynamics at equilibrium. Rather, they are the outcome of a Darwinian evolutionary process driven by stochastic factors such as mutations, recombination, genetic drift, and selective pressures based on a certain fitness function. Evolution is known to be a non-equilibrium process [Kussell and Vucelja (2014)], and in principle, the energy-based approach is not guaranteed to work.

Nevertheless, as we discussed in the previous chapters, DCA is capable of recovering true direct couplings in the case of real protein families. Furthermore, there is evidence that the inferred energy landscape can be correlated with the original one. Shekhar and collaborators introduced a numerical model for the evolution of HIV viruses using a fitness landscape trained on patient-derived viral sequence data [Shekhar et al. (2013)]. In their work, they demonstrated that the DCA inference method accurately describes the fitness landscape of the virus. Other studies have also compared the inferred energy landscape with mutational experiments, showing good correlations in some cases between differences in experimental activity among mutants and their respective differences in energy [Hopf et al. (2017); Rodriguez-Rivas et al. (2022)].

In light of these results, the natural questions that arise are: What is the relationship between the inferred energy landscape and the "true" fitness landscapes driving selection during evolution? And under what conditions do the two correlate? The scope of this

part of the thesis is to delve into these questions. Specifically, we will extensively study a particular regime of the evolutionary dynamics of biological sequences known as Quasi-Linkage Equilibrium (QLE) regime. This regime was first introduced by Kimura in the 1960s [Kimura (1965)] and has recently been studied in detail by Neher, Shraiman, and others [Neher and Shraiman (2011a); Neher et al. (2013)]. The QLE regime can be seen as a perturbation theory of the Wright-Fisher model of evolution [Wright (1931)] when the recombination rate between individuals is high compared to selective pressures. It is characterized by the fact that the genotype population converges to an equilibrium distribution where couplings between residues (also called *loci*) are linearly correlated with the strength of the epistatic effect between them as determined by the fitness landscape. Therefore, in this regime, an inferred energy landscape is expected to correlate with the true fitness landscape.

This chapter will primarily review the previous works on the QLE regime (in particular the work done by Neher and Shraiman (2011a) and Zeng and Aurell (2020)) when recombination is the dominant driving force of evolution. In the subsequent chapters, we will present our own work conducted in collaboration with researchers from Stockholm University, where we extend the QLE regime to include cases where mutations and recombinations have comparable strength.

4.1. Dynamics of the genotype distribution

In this section of the chapter we will introduce the theoretical framework for studying the dynamics of biological sequences under Wright-Fisher evolution, which will be necessary to introduce the QLE regime and the main results obtained by Neher and collaborators in [Neher and Shraiman (2011a); Neher et al. (2013)].

4.1.1 Genotypes and quantitative traits

Contrarily to the first part of this thesis, where we considered sequences of amino acids, in this section we will consider the case of haploid genomes of L loci with two alleles each. Practically, we write each possible genome as a sequence $\mathbf{g} = \{s_1, \dots, s_L\}$ where each s_i can take one of two possible values. For simplicity in the algebra, we choose the values to be $s_i \in \{-1, +1\}$ for each locus $i = 1, ..., L$. With this choice, we define a genotype space as a L-dimensional hypercube where each one of the possible 2^L vertices corresponds to a possible genome.

With this definition in mind we can define any quantitative trait as a function of the genotype. One of the most important trait of such sort is the fitness landscape, which we will denote by $F(\mathbf{g})$. This function assigns a fitness value to each genotype, measuring the expected reproductive success. This function can be parameterized in different ways as proposed in past literature [Barton and Turelli (1991); Weinberger (1991); Hansen and Wagner (2001); Hansen (2006)].

Generally speaking, any function defined on a L-dimensional hypercube can be expanded in a Fourier series as follows:

$$F(\mathbf{g}) = \bar{F} + \sum_i f_i s_i + \sum_{i<j} f_{ij} s_i s_j + \cdots + \sum_{i_1<\cdots<i_k} f_{i_1 \dots i_k} s_{i_1} \dots s_{i_k} + \dots , \tag{4.1}$$

where the first sum represents the independent contributions of single loci, the second sum accounts for the interactions between pairs of loci, and the higher-order terms capture the effects of subgroups of loci [Hordijk and Stadler (1998); Stadler and Wagner (1997)]. The coefficient f_i represents the additive effect of locus i that is independent of other loci. The higher-order terms involving locus i describe the genetic background dependence of the

effect of the allele s_i. These higher-order terms, beyond the first order, capture genetic interactions or "epistasis." The contribution of each locus or subgroup of loci is determined through unbiased averaging over the remainder of the genome, with each genotype entering with a weight of 2^L:

$$\bar{F} = \frac{1}{2^L}\sum_{\mathbf{g}} F(\mathbf{g})\,,\ f_i = \frac{1}{2^L}\sum_{\mathbf{g}} F(\mathbf{g})s_i\,,\ f_{ij} = \frac{1}{2^L}\sum_{\mathbf{g}} F(\mathbf{g})s_i s_j\,,\ \ldots$$
$$f_{i_1\ldots i_k} = \frac{1}{2^L}\sum_{\mathbf{g}} F(\mathbf{g})s_{i_1}\ldots s_{i_k}\,. \quad (4.2)$$

There are in total 2^L coefficients $f^{(k)}_{i_1\ldots i_k}$ that give an exact representation of the fitness landscape.

It is useful to define the variance of the fitness over the entire genome space as:

$$\bar{\sigma}^2 = \frac{1}{2^L}\sum_{\mathbf{g}}\left(F(\mathbf{g})-\bar{F}\right)^2 = \sum_i f_i^2 + \sum_{i<j} f_{ij}^2 + \cdots + \sum_{i_1<\cdots<i_k} f^2_{i_1\ldots i_k} + \ldots\,, \quad (4.3)$$

where the last decomposition is equivalent to the Parseval's theorem for Fourier series. The variance of the fitness landscape is a measure of the ruggedness of the fitness landscape. In fact, if the variance is controlled by the first order terms, the fitness landscape is smooth, whereas an increase in the contribution of higher-order terms leads to a more rugged landscape. [Mézard et al. (1987)].

Hereafter we will only consider fitness landscapes truncated at the second order, i.e. we will only consider the first two terms in the expansion (4.1)

$$F(\mathbf{g}) = \bar{F} + \sum_i f_i s_i + \sum_{i<j} f_{ij} s_i s_j\,. \quad (4.4)$$

This simplifies the following discussion and still allows for a rich phenomenology. Moreover, we will refer to the local fields f_i as the *additive effects* and to the couplings f_{ij} as the *epistatic effects*.

4.1.2 Evolutionary dynamics

In this section we are going to describe the dynamics of the population under evolution as presented in Neher and Shraiman (2011a). The key ingredients of evolution in classical population genetics are selection, mutation, recombination and genetic drift. Selection is the process by which individuals with higher fitness have a higher probability of reproducing in the next generation. Mutation is the process by which the genome of an individual changes from one generation to the next. Recombination is the process by which the genome of an individual is combined with the genome of another individual to produce new offspring. Genetic drift is a stochastic process that describes the random fluctuations in the frequency of a genotype in a population due to finite population size [Fisher (1930); Blythe and McKane (2007)].

Fluctuations due to genetic drift are inversely proportional with the population size. In the following discussion, we disregard genetic drift by assuming the population size to be infinite. In this context we can describe the state of a population by the probability distribution $P(\mathbf{g},t)$ of the different genotypes $\mathbf{g}$ at time t. Below we give a description of the dynamics of this probability distribution under the action of selection, mutation and recombination (see Figure 4.1).

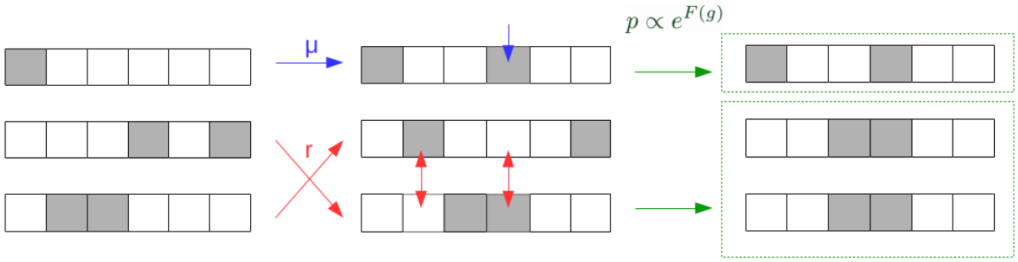


Figure 4.1: Schematic representation of the evolution of spin-like genomes. Each individual has a genome of L loci (white and grey colors represents the two possible alleles $s_i = \pm 1$). At each generation, spins randomly flips with a certain mutation rate μ. Random individuals mate and exchange their genomes with a recombination rate r. The offspring genome are then selected with probability proportional to the exponential of the fitness function.

- **Selection:** it acts on the population by increasing the frequency of genotypes with higher fitness. Here the fitness function $F(\mathbf{g})$ defines its replication rate in the population. Considering a small interval of time Δt, the probability distribution of the population changes as follows:

$$P(\mathbf{g}, t+\Delta t) = \frac{e^{F(\mathbf{g})\Delta t} P(\mathbf{g}, t)}{\langle e^{F\Delta t} \rangle}, \tag{4.5}$$

  where $\langle \cdot \rangle = \sum_{\mathbf{g}} (\cdot)\, P(\mathbf{g}, t)$ is the average over the population at time t. In the continuity limit $\Delta t \to 0$, the probability distribution $P(\mathbf{g}, t)$ satisfies the following differential equation:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{\mathrm{sel}} P(\mathbf{g}, t) = [F(\mathbf{g}) - \langle F \rangle]\, P(\mathbf{g}, t)\,. \tag{4.6}$$

- **Mutation:** it acts on the population by changing the genome of an individual. In the case of spin-like genomes, mutations are flips of the spin at a given locus. The probability of a mutation to occur at a given locus is $\mu \Delta t$, where μ is the mutation rate. The probability distribution of the population changes as follows:

$$P(\mathbf{g}, t+\Delta t) = P(\mathbf{g}, t) + \mu \Delta t \sum_{i=1}^{L} [P(M_i \mathbf{g}, t) - P(\mathbf{g}, t)]\,, \tag{4.7}$$

  where $M_i \mathbf{g}$ is the genome obtained by flipping the spin at locus i in the genome $\mathbf{g}$. The term $P(M_i \mathbf{g}, t)$ in the sum accounts for all the individuals that mutate into the genome $\mathbf{g}$, while the second term $-P(\mathbf{g}, t)$ accounts for all the individuals with genome $\mathbf{g}$ that mutate into anything else.

  In the continuity limit $\Delta t \to 0$, the probability distribution $P(\mathbf{g}, t)$ satisfies the following differential equation:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{\mathrm{mut}} P(\mathbf{g}, t) = \mu \sum_{i=1}^{L} [P(M_i \mathbf{g}, t) - P(\mathbf{g}, t)]\,. \tag{4.8}$$

- **Recombination:** it acts on the population by combining the genomes of two individuals with at a certain recombination rate r to produce a new offspring. In the case of spin-like genomes, recombination is the process by which two individuals exchange their genomes at different loci. For a short interval of time Δt, we can

generally write the evolution of the probability distribution as follows [Zeng et al. (2021)]:

$$P(\mathbf{g},t+\Delta t) = (1-r\Delta t)P(\mathbf{g},t) + r\Delta t \sum_{\mathbf{g}',\mathbf{g}''} P_2(\mathbf{g}',\mathbf{g}'',t)C(\mathbf{g}',\mathbf{g}''\to\mathbf{g})\,, \tag{4.9}$$

where $P_2(\mathbf{g}',\mathbf{g}'',t)$ is the joint probability distribution that two genomes meet at time t to recombine together and $C(\mathbf{g}',\mathbf{g}''\to\mathbf{g})$ is the probability that the recombination of the genomes $\mathbf{g}'$ and $\mathbf{g}''$ produces the genome $\mathbf{g}$. The first term in the equation, $(1-\Delta tr)P(\mathbf{g},t)$, accounts for all the individuals with genome $\mathbf{g}$ that do not recombine, while the second term accounts for all the new individuals with genome $\mathbf{g}$ that are produced by recombination.

In general, the joint probability distribution $P_2(\mathbf{g}',\mathbf{g}'',t)$ makes the whole model intractable, since in general the dynamics does not have a closed form (*i.e.* the evolution of P_2 will depend on three-genotype distribution, P_3, and so on). A common way to deal with this problem is to assume that the population is well-mixed, so that the joint probability distribution factorizes as follows:

$$P_2(\mathbf{g}',\mathbf{g}'',t) = P(\mathbf{g}',t)P(\mathbf{g}'',t)\,. \tag{4.10}$$

To simplify the algebra, we describe the cross-over as a vector $\boldsymbol{\xi}$, with $\xi_i \in \{0,1\}$. For each couple of mother and father genomes, respectively $\mathbf{g}^{(m)}$ and $\mathbf{g}^{(f)}$, if $\xi_i = 1$ the i-th locus of the new offspring is inherited from the mother, otherwise from the father. A bit of algebra leads to $s_i^{(m)} = s_i\xi_i + s_i'(1-\xi_i)$ and $s_i^{(f)} = s_i'\xi_i + s_i(1-\xi_i)$ where s_i is the allele of the child at locus i, and s_i' is the discarded allele. The probability of each realization of $\boldsymbol{\xi}$ is given by $C(\boldsymbol{\xi})$. Hence, we can rewrite Eq. (4.9) as follows:

$$P(\mathbf{g},t+\Delta t) = (1-r\Delta t)P(\mathbf{g},t) + r\Delta t \sum_{\mathbf{g}',\boldsymbol{\xi}} C(\boldsymbol{\xi})P(\mathbf{g}^{(m)},t)P(\mathbf{g}^{(f)},t)\,, \tag{4.11}$$

where the sum is done over the possible realizations of $\boldsymbol{\xi}$ and the possible genomes not passed to the offspring $\mathbf{g}'$. In the continuity limit, with a bit of algebra, this becomes:

$$\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{\mathrm{rec}} P(\mathbf{g},t) = r\sum_{\mathbf{g}',\boldsymbol{\xi}} C(\boldsymbol{\xi})\left[P(\mathbf{g}^{(m)},t)P(\mathbf{g}^{(f)},t) - P(\mathbf{g},t)P(\mathbf{g}',t)\right]. \tag{4.12}$$

We can now combine the three processes described above to obtain the full dynamics of the probability distribution $P(\mathbf{g},t)$. In the continuity limit, the dynamics is described by the following differential equation [Neher and Shraiman (2011a)]:

$$\begin{aligned}\left.\frac{\mathrm{d}}{\mathrm{d}t}\right|_{\mathrm{full}} P(\mathbf{g},t) = {} & \left[F(\mathbf{g}) - \langle F\rangle\right]P(\mathbf{g},t) + \mu\sum_{i=1}^{L}\left[P(M_i\mathbf{g},t) - P(\mathbf{g},t)\right] \\ & + r\sum_{\mathbf{g}',\boldsymbol{\xi}} C(\boldsymbol{\xi})\left[P(\mathbf{g}^{(m)},t)P(\mathbf{g}^{(f)},t) - P(\mathbf{g},t)P(\mathbf{g}',t)\right].\end{aligned} \tag{4.13}$$

which holds in the limit of infinite population size and provided that selection is weak ($\Delta t F(\mathbf{g}) \ll 1$).

Eq. (4.13) is a non-linear differential equation which is in general hard to solve. We will describe in the next section how this equation can be solved in the case of weak selection and high recombination rate, which is the regime of interest of QLE (we will then extend the discussion to the regime oh high mutation rate in the Chapter 5 and 6).

4.1.3 Dynamics of 1st and 2nd order cumulants

It would be useful for the following discussion to introduce first and second order cumulants of the probability distribution $P(\mathbf{g},t)$, which are defined as follows:

$$\begin{aligned}\chi_i &= \langle s_i \rangle \\ \chi_{ij} &= \langle s_i s_j \rangle - \chi_i \chi_j \,,\end{aligned} \tag{4.14}$$

where we note that by definition $\chi_{ii} = 1 - \chi_i^2$. The higher order cumulants are defined in a similar way and can be computed via the cumulant generating function [McQuarrie (2000)]. The second order cumulants χ_{ij} is useful to define the regime of *Linkage Equilibrium* (LE), where all pairs of site are uncorrelated, $\chi_{ij} = 0$. We will show in the next section how the QLE regime is defined as a perturbation of the LE regime.

Injecting the definition of the cumulants in Eq. (4.13), we obtain the equation for the evolution of the cumulants. For the first order cumulants, we obtain [Neher and Shraiman (2011a)]:

$$\begin{aligned}\dot{\chi}_i &= \sum_{\mathbf{g}} s_i \dot{P}(\mathbf{g},t) = \sum_{\mathbf{g}} s_i \left[F(\mathbf{g}) - \langle F \rangle\right] P(\mathbf{g},t) + \mu \sum_{j=1}^{L} \left[\sum_{\mathbf{g}} s_i P(M_j \mathbf{g},t) - \chi_i \right] , \\ \dot{\chi}_i &= \langle s_i \left[F(\mathbf{g}) - \langle F \rangle\right] \rangle - 2\mu \chi_i \,,\end{aligned} \tag{4.15}$$

where the sum on the RHS of the first line is either zero (when $i \neq j$) or $-2\chi_i$ (when $i = j$). It's easy to understand that recombinations do not play a role in the evolution of the first order cumulants, since they do not change the average value of the genome, which are only exchanged between individuals of the populations.

For the second order cumulant we can write $\dot{\chi}_{ij} = \langle \dot{s_i s_j} \rangle - \dot{\chi}_i \chi_j - \chi_i \dot{\chi}_j$. The last two terms of this equation can be easily obtained from Eq. (4.15). The derivative of $\langle s_i s_j \rangle$ due to only selection is also easy to obtain. Hence, we only have to compute the contribution of mutations and recombinations. For mutations, we have:

$$\langle \dot{s_i s_j} \rangle_{\text{mut}} = \mu \sum_{k=1}^{L} \left[\sum_{\mathbf{g}} s_i s_j P(M_k \mathbf{g},t) - \langle s_i s_j \rangle \right] = -4\mu \langle s_i s_j \rangle \,, \tag{4.16}$$

similarly to the case of χ_i's. Evaluating the recombination term, we find:

$$\begin{aligned}\langle \dot{s_i s_j} \rangle_{\text{rec}} &= r \sum_{\boldsymbol{\xi}} C(\boldsymbol{\xi}) \sum_{\mathbf{g},\mathbf{g}'} s_i s_j \left[P(\mathbf{g}^{(m)},t) P(\mathbf{g}^{(f)},t) - P(\mathbf{g},t) P(\mathbf{g}',t) \right] \\ &= r \sum_{\boldsymbol{\xi}} C(\boldsymbol{\xi}) \left[\xi_i (1-\xi_j) + \xi_j (1-\xi_i) \right] \left(\langle s_i s_j \rangle - \chi_i \chi_j \right) \\ &= -r c_{ij} \chi_{ij} \,,\end{aligned} \tag{4.17}$$

where the last equation can be obtained by taking advantage of the identities $s_i^{(m)} = s_i \xi_i + s_i' (1-\xi_i)$ and $s_i^{(f)} = s_i' \xi_i + s_i (1-\xi_i)$ described above. Here, we have defined $c_{ij} = \sum_{\boldsymbol{\xi}} C(\boldsymbol{\xi}) \left[\xi_i (1-\xi_j) + \xi_j (1-\xi_i) \right]$, which corresponds to the probability that loci i and j derive from different parents. For high-recombination organisms, c_{ij} depends on the cross-over rate ρ and the genomic distance between loci i and j [Zeng and Aurell (2020)], except when loci i and j are very closely spaced on the genome.

$$c_{ij} \approx \frac{1}{2} \left(1 - e^{-2\rho |i-j|} \right) \tag{4.18}$$

Merging all the terms together, we obtain the following equation for the second order cumulants [Neher and Shraiman (2011a)]:

$$\dot{\chi}_{ij} = \langle (s_i - \chi_i)(s_j - \chi_j)\,[F(\mathbf{g}) - \langle F\rangle]\rangle - 4\mu\chi_{ij} - rc_{ij}\chi_{ij}\,. \tag{4.19}$$

It is easy to notice that while mutations and recombinations tend to relax the cumulants to zero. It also can be noticed that in absence of epistatic terms on the fitness landscape, the populations evolves into LE. As a final remark we notice how in general the dynamics of the cumulants have a non-closed form (*i.e.* $\dot{\chi}_{ij}$ depends on third order cumulants, and so on). In Chapter 5 we will discuss an approximation scheme to close the dynamics of the cumulants under the assumption that the probability distribution is Gaussian.

4.1.4 Linear model of evolution without recombinations

As a final observation, we can linearize the equation in the absence of recombinations ($r = 0$) by considering the evolution of the unormalized probability distribution $Y(\mathbf{g},t) = P(\mathbf{g},t)e^{\int_0^t \langle F\rangle_{t'}\mathrm{d}t'}$. In this case, the dynamics is described by the following linear differential equation:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}Y(\mathbf{g},t) &= \left(\frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{\mathrm{sel}} + \frac{\mathrm{d}}{\mathrm{d}t}\bigg|_{\mathrm{mut}}\right) Y(\mathbf{g},t) \\ &= F(\mathbf{g})Y(\mathbf{g},t) + \mu\sum_{i=1}^{L}\left[Y(M_i\mathbf{g},t) - Y(\mathbf{g},t)\right]. \end{aligned}\tag{4.20}$$

Looking for stationary solution of this evolutionary model is equivalent to find the kernel of the linear operator in the RHS of Eq. (4.20). This shows how the complexity of solving the dynamics of the population is mostly related to the presence of recombination.

4.2. QLE as a perturbation theory

We now present the arguments of Neher and Shraiman (2011a) based on the initial work of Kimura (1965) to derive the QLE when selection is weak in the time scale of recombination, *i.e.* $\bar{\sigma} \ll r$ (we will refer to it as the Kimura-Neher-Shraiman theory, KNS). Later in this section, we will describe how in the QLE epistatic effect can be recovered from energy models inferred from sequence data of evolving genomes and present the numerical results obtained by Zeng and Aurell (2020).

4.2.1 Kimura-Neher-Shraiman theory

In this high-recombination regime the induced correlation will be also weak. Hence, we approximate probability distribution $P(\mathbf{g},t)$ as a Gibbs-Boltzmann distribution of the Ising/Potts type:

$$\log P(\mathbf{g},t) = -Z(t) + \sum_{i=1}^{L}\phi_i(t)s_i + \sum_{i<j}J_{ij}(t)s_is_j\,, \tag{4.21}$$

where $Z(t)$ is a normalization constant playing the role of the free energy in statistical mechanics and can be used as the generator function of the cumulants via

$$\chi_i = \frac{\partial Z}{\partial\phi_i}\,,\ \chi_{ij} = \frac{\partial^2 Z}{\partial\phi_i\partial\phi_j}\,. \tag{4.22}$$

The second order term J_{ij} capture information about correlations induced by selection and in this limit can be thought to be small.

Following Neher and Shraiman (2011a), we can inject this ansatz in the master equation for the evolution of $\log P(\mathbf{g},t)$ in presence of mutations and recombination and obtain

$$\begin{aligned}\frac{\mathrm{d}\log P(\mathbf{g},t)}{\mathrm{d}t} &= -\frac{\mathrm{d}}{\mathrm{d}t}\log Z(t) + \sum_i \dot{\phi}_i(t)s_i + \sum_{i<j}\dot{J}_{ij}(t)s_i s_j \\ &= F(\mathbf{g}) - \langle F\rangle + \mu \underbrace{\sum_i \left[\frac{P(M_i\mathbf{g},t)}{P(\mathbf{g},t)} - 1\right]}_{M(\mathbf{g},t)} + \\ &+ r\underbrace{\sum_{\boldsymbol{\xi},\mathbf{g}'} C(\boldsymbol{\xi})P(\mathbf{g}',t)\left[\frac{P(\mathbf{g}^{(m)},t)P(\mathbf{g}^{(f)},t)}{P(\mathbf{g},t)P(\mathbf{g}',t)} - 1\right]}_{R(\mathbf{g},t)}. \end{aligned} \tag{4.23}$$

where we have defined the contribution due to recombinations and mutations respectively $R(\mathbf{g},t)$ and $M(\mathbf{g},t)$. In the following discussion we also assume that the mutation are a much weaker process than selection and recombination. Hence, we neglect the term $M(\mathbf{g},t)$ in the following discussion. See Chapter 6 to see how this assumption can be relaxed.

The recombination term can be rewritten as follows:

$$R(\mathbf{g},t) = \sum_{\boldsymbol{\xi},\mathbf{g}'} C(\boldsymbol{\xi})P(\mathbf{g}',t)\left[e^{\sum_{i<j}J_{ij}\left[(\xi_i\xi_j+\overline{\xi}_i\overline{\xi}_j-1)(s_is_j+s'_is'_j)+(\xi_i\overline{\xi}_j+\overline{\xi}_i\xi_j)(s_is'_j+s_is'_j)\right]} - 1\right], \tag{4.24}$$

where $\overline{\xi}_i = (1-\xi_i)$. Now, as said before, in the high recombination limit we suppose that the interactions J_{ij} are small and can be expanded from the exponential. Hence, we can write:

$$\begin{aligned} R(\mathbf{g},t) &\sim \sum_{\boldsymbol{\xi},\mathbf{g}'} C(\boldsymbol{\xi})P(\mathbf{g}',t)\left[\sum_{i<j}J_{ij}\left[(\xi_i\xi_j+\overline{\xi}_i\overline{\xi}_j-1)(s_is_j+s'_is'_j)+(\xi_i\overline{\xi}_j+\overline{\xi}_i\xi_j)(s_is'_j+s_is'_j)\right]\right] \\ &= \sum_{i<j} c_{ij}J_{ij}\left[(s_i\langle s_j\rangle + s_j\langle s_i\rangle) - (s_is_j + \langle s_is_j\rangle)\right], \end{aligned} \tag{4.25}$$

where $c_{ij} \equiv \sum_\xi C(\{\xi\})\left[\xi_i\overline{\xi}_j + \overline{\xi}_i\xi_j\right]$ is the same defined in the above section.

Injecting these results for $R(g,t)$ into eq. (4.23) and separating the dependencies on s_i and s_is_j, we can obtain equations for $\dot{\phi}_i$ and $\dot{J}_{ij}$ similarly to what has been done in [Neher and Shraiman (2011a)]. In particular, we find:

$$\begin{aligned}\dot{\phi}_i &= f_i + r\sum_{j\neq i} c_{ij}J_{ij}\langle s_j\rangle \\ \dot{J}_{ij} &= f_{ij} - rc_{ij}J_{ij}. \end{aligned} \tag{4.26}$$

Hence, the interactions J_{ij} will quickly evolve through the stationary solution [Neher and Shraiman (2011a)]

$$J_{ij}^{st.} = f_{ij}/rc_{ij}. \tag{4.27}$$

The importance of the last equation is that it shows how the evolutionary couplings, f_{ij}, and the couplings inferred from sequence data, J_{ij}, only differ by a multiplicative prefactor that depends on the dynamics of recombination and mutation. This justifies the use of DCA to reconstruct fitness landscapes, at least in certain regimes of the evolutionary dynamics. We will see how this result can be improved in Chapter 6, where we analyse QLE in the case of comparable mutation and recombination rates.

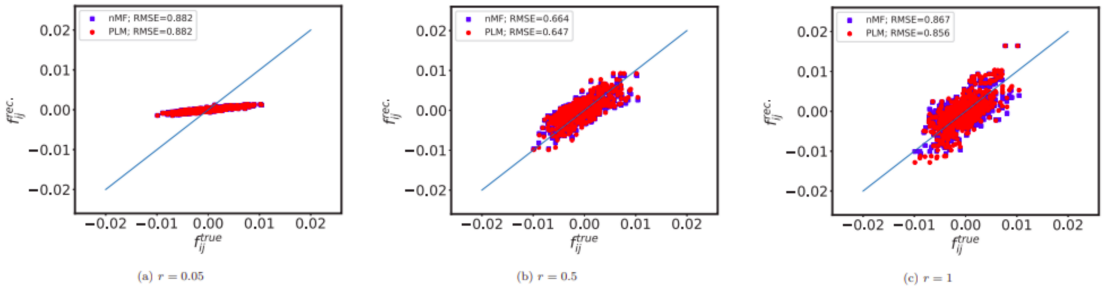


Figure 4.2: Comparison between real and reconstructed fitness interactions obtained with Eq. (4.28) at different recombination rates r (from left to right: $r = 0.05, 0.5, 1$). Mutation rate $\mu = 0.05$, crossover rate $\rho = 0.5$, fitness variation $\sigma = 0.004$. J_{ij} are inferred by the nMF and PLM procedures. Image taken from Zeng and Aurell (2020).

4.2.2 Comparison with inferred energy models

As we just described, in QLE the population quickly evolves to a quasi-factorized distribution where each couple of loci weakly interacts. This results in the probability distribution to reach a stationary state similar to a Boltzmann distribution define by an Ising model.

Recovering the definition of Direct Coupling Analysis done in Section 2.2, we realize that in QLE the correspondence between the real fitness function and the inferred energy holds by construction. Moreover, the inferred couplings J_{ij} are linearly related to the epistatic effects f_{ij} via Eq. (4.27).

To test this prediction, Zeng and Aurell (2020) performed numerical simulations of the evolution of a population of haploid sequences under the action of selection, mutation and recombination (plus genetic drift due to finite size in the population) using an algorithm created by Neher and Zanini called FFPopSim [Zanini and Neher (2012)].

In particular, they initialize a fitness function $F(\mathbf{g})$ with epistatic interactions f_{ij} drawn from a Gaussian distribution with mean zero and standard deviation σ. Then, they initialize a population of random sequences and evolve it for a certain number of generations using FFPopSim, with given values of mutation rate μ, out-crossing rate r and with a given out-crossing probability matrix c_{ij}. Then, when stationarity is reached, they take snapshots of the population at different generations and use this data to do DCA using both Gaussian approximation (also called *naive* mean-field, nMF) and pseudo-likelihood maximization (PLM). Finally, they recover the reconstructed epistatic interactions by inverting Eq. (4.27) as

$$f_{ij}^{*} = J_{ij}^{st.} \cdot r c_{ij} \,, \tag{4.28}$$

and compare the results with the original fitness function by computing the normalized L_2 distance given by:

$$\epsilon = \sqrt{\frac{\sum_{i<j}(f_{ij} - f_{ij}^{*})^2}{\sum_{i<j} f_{ij}^2}} \,. \tag{4.29}$$

An example is given in Fig. 4.2, where we can see how the inferred fitness function is closer to the original one when the recombination rate increases [see Fig. 4.2(mid panel)]. We also notice that for too high recombination rates [see Fig. 4.2(right panel)] the signal we are trying to recover is too small and mixes with the noise intrinsic in the simulation of finite populations, consequently reducing the reconstruction performance.

Importantly, they have tested the performance of this inference method, which disregards effects due to mutations, at different mutations rate μ (see Figure 4.3). Interestingly, they

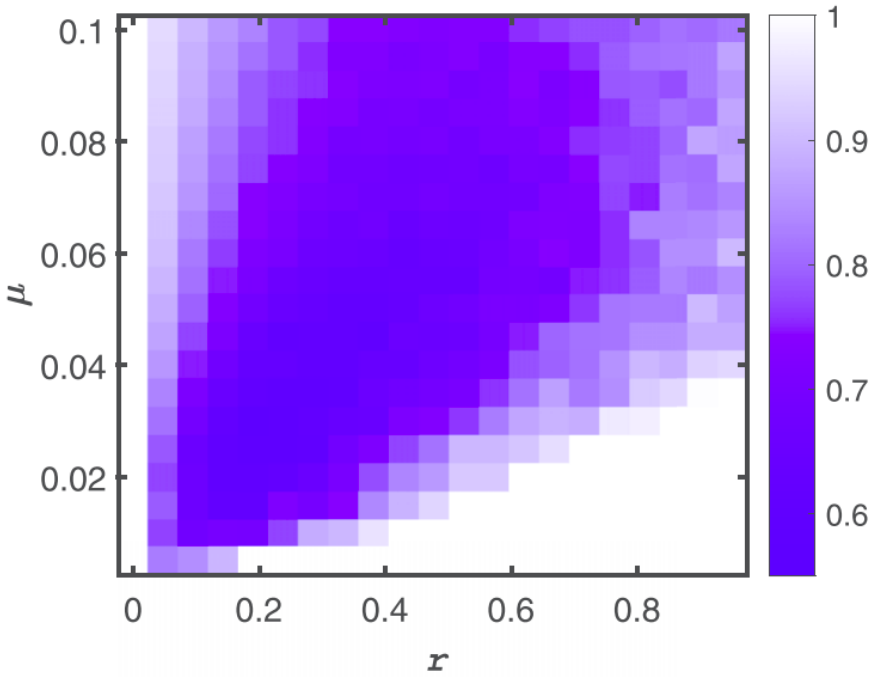


Figure 4.3: Color map of the reconstruction error ϵ in Eq. (4.29) as a function of mutation rate μ and recombination rate r (and $\rho = 0.5$). The fitness variation is $\sigma = 0.004$. Image taken from Zeng and Aurell (2020).

found that the reconstruction error ϵ is low for intermediate values of μ and increases for very low and very high values of μ. In Chapter 6, we show that by addressing also the effect of mutations, the inference method can be improved and the reconstruction error can be reduced for a wider range of mutation rates.

4.3. Breakdown of QLE and the Clonal Condensation phase (CC)

QLE is a very useful approximation scheme to describe the dynamics of genotype distributions. However, it is natural to ask when this perturbation expansion does not hold anymore. Neher et al. (2013) pointed out that in the case of lower recombination rate and/or stronger epistatic effects in the fitness function, the system encounters a phase transition similar to the glass transition in spin glasses [Mézard and Montanari (2009)]. In fact, while in the QLE phase the population is broadly distributed in the sequence space, with almost independent loci (similar to the paramagnetic phase of spin glasses), in the related transition the population condenses into a small number of genotypes (similar to the spin glass phase). This new phase is called *Clonal Condensation* (CC).

An analytical description of the QLE/CC transition is given in [Neher et al. (2013)] for a coarse-grained model of the population dynamics (similar to mean-field theory) in the limit of infinite population size and genome length. A sketch of the phase diagram as presented in [Neher et al. (2013)] is given in Figure 4.4. In particular, they separate the additive and epistatic contribution to the fitness function, called respectively A and E (such that $F = A + E$), and define the heritability of the model as the ratio between the variability (*i.e.* variance) of the additive fitness and the total fitness variance, $h^2 \equiv \sigma_A^2/\sigma_F^2$. For higher values of heritability and/or recombination rate (the QLE phase), all genotypes are short lived and quickly destroyed by recombination. The mean value of the additive fitness, $\langle A \rangle$ grows in time at constant pace, while the mean epistatic fitness $\langle E \rangle$ quickly

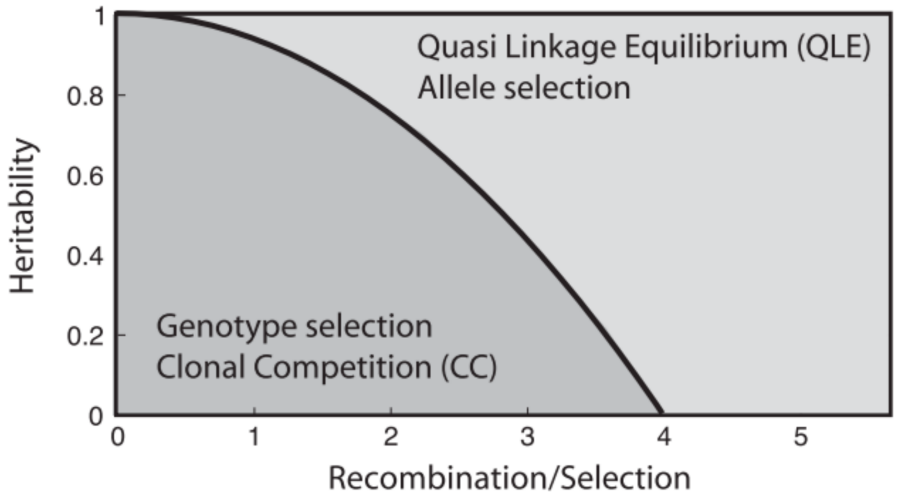


Figure 4.4: The range of validity of QLE as a function of $\bar{\sigma}/r$ and the heritability, *i.e.*, the ratio of additive variance to the total fitness variance. Below the transition line, strong linkage equilibrium is expected and selection operates on genotypes rather than alleles. Image and caption taken from Neher and Shraiman (2011a).

grows to a small constant value. In the CC phase, instead, recombinations are not able to quickly destroy new genotypes with strong epistatic fitness, which are able to survive and grow exponentially in time. Now the genotype distribution is dominated by a finite number of genotypes (called *clones*) which are populated by many individuals.

In this direction, numerical analysis has been conducted by Dichio et al. (2023). Using the same set-up of Zeng and Aurell (2020), they simulated the evolution of a finite population with random epistatic fitness and small additive fitness. They empirically observed for intermediate values of epistasis an intermittent regime fluctuating between QLE and Non-Random Coexistence (NRC) regime. In NRC the distribution of the fitness in the population is bimodal, with a group of individuals having high fitness, and DCA does not work anymore.

In general, approaches like Direct Coupling Analysis are no longer able to retrieve meaningful information on the *global* fitness landscape driving the evolution of a population in the CC phase. However, if we only focuses on understanding the shape of the fitness landscape *locally* around a given clone of the population, such methods can still be useful. As an example, Posani and collaborators in [Posani et al. (2022)] showed that training simpler models (such as independent site models) only on data close to a given wild-type sequence can give better performances at predicting effects of single mutations on that wild-type. This may lead to a piecewise approach for the inference of the global fitness landscape, where we first identify the high-fitness clones of the population and then we merge together local models trained around them.

5

Gaussian closure in the QLE regime

In Chapter 4, we presented the Kimura-Neher-Shraiman theory (KNS) to recover the Quasi-linkage equilibrium phase in the genotype distribution when recombination is dominant compared to selection and mutations are neglected [Kimura (1965); Neher and Shraiman (2011a)]. Their approach consists of perturbing the factorised distribution over the loci in presence of small epistatic terms in the fitness function, allowing the loci to weakly interact. This results in the genotype distribution (in the infinite population size limit) to quickly converge to an Ising-like Boltzmann distribution which can be reconstructed from the data using inverse statistical mechanics techniques such as Direct Coupling Analysis (DCA).

In this chapter, we try to extend the KNS theory for QLE to the case where mutations are not neglected. To this aim, we will introduce a different approach to study the dynamics of genotype distribution in the QLE regime, based on the hypothesis that the genotype distribution can be approximated by a Gaussian distribution. This approach is based on our original work published in [Mauri et al. (2021)].

The main advantage of such procedure is that the population dynamics can be described only by looking at the evolution of the first and second order cumulants (respectively χ_i and χ_{ij}). In the Gaussian closure (GC) scheme this dynamics takes the form of a closed set of coupled differential equations for the cumulants, which can be solved numerically.

In the first part of this chapter, we introduce the model for simulating the stochastic evolution of genome populations under the effects of selection, recombinations and mutations used in our paper [Mauri et al. (2021)].

The second part of this chapter focuses on introducing the Gaussian Closure (GC) scheme, deriving the equations for the cumulants, and discussing its regime of validity. Unlike previous works such as [Kimura (1956); Neher and Shraiman (2011a)], our Gaussian scheme allows us to study the existence of Quasi-Linkage Equilibrium (QLE) in a broader range of recombination and mutation rates without relying on large expansions.

In the last part, we apply the GC scheme to the short-range epistatic model proposed in [Mauri et al. (2021)]. This fitness landscape is characterised by two distant genomes having equal and maximal fitness. Using our approximation scheme and solving the model analytically, we explore the effects of mutation and recombination rates. Within the QLE regime, we observe a phase transition that separates a disordered (paramagnetic) regime, where alleles are not polarized along either genome, from a polarized (ferromagnetic) phase. In the polarized phase, alternating allele domains coarsen, and one of the two genomes dominates the population.

5.1. Model for the stochastic evolution of a genome population

Similarly to the previous chapter, we consider the evolution of N individuals with spin-like genomes of L loci. The genome of each individual is represented by a binary vector $\mathbf{g} = \{s_1, \ldots, s_L\}$, where $s_i = \pm 1$ is the allele at locus i. This population stochastically evolve over time due to selection, recombinations and mutations.

5.1.1 Selection

The individuals carrying the genome $\mathbf{g}$ grow in the population with a certain rate defined by the fitness function, $F(\mathbf{g})$. Recovering Eq. (4.4), we write the fitness function as

$$F(\mathbf{g}) = F_0 + \sum_{i=1}^{L} f_i s_i + \sum_{i<j}^{L} f_{ij} s_i s_j \tag{5.1}$$

where we consider only additive (f_i) and epistatic (f_{ij}) terms. We enforce selection in simulations as follows. First, we discretize time into very small time steps Δt. Then, at each time step selection results in a re-sampling of the available sequences according to a Boltzmann probability distribution with the weights defined by the fitness of each sequence. More precisely, each individual $\mathbf{g}_k$ has a probability to survive of $p(\mathbf{g}_k) \propto e^{F(\mathbf{g}_k)}$ with $k = 1, \ldots, N$. Such probability is normalised over all N individuals. In this way, we make sure that the population size, N, is kept fixed. Note that the constant F_0 plays no role and can be set to zero.

5.1.2 Mutation

The next step is to implement mutations in the population. For any allele/spin s_i of a specific genome in the population, we decide whether to mutate it or not with a rate (probability per unit of time) μ. In that case, we flip the sign of the spin, $s_i \to -s_i$.

5.1.3 Recombination

For each infinitesimal time interval Δt, a pair of individuals is randomly selected with probability $r\Delta t / N_{pairs}$ from the $N_{pairs} = \frac{1}{2}N(N-1)$ possible pairs, where r is the recombination rate. These individuals, referred to as the mother (m) and father (f), undergo outcrossing. Two new genotypes $\mathbf{g}$ and $\mathbf{g}'$ are formed by inheriting certain loci from the mother's genome, $\mathbf{g}^{(m)}$, and the complement from the father's genome, $\mathbf{g}^{(f)}$. This inheritance process is described by the stochastic inheritance vector $\boldsymbol{\xi} = \{\xi_i\}$, where $\xi_i \in \{0,1\}$. If $\xi_i = 1$, the i-th allele of $\mathbf{g}$ is inherited from the mother, and if $\xi_i = 0$, it is inherited from the father. The opposite is true for $\mathbf{g}'$. Therefore, $s_i = s_i^{(m)}\xi_i + s_i^{(f)}(1-\xi_i)$. The probability distribution of the inheritance vector $\boldsymbol{\xi}$ is denoted by $C(\boldsymbol{\xi})$. Specifically, we will consider the recombination correlation in Eq. (4.18), $c_{ij} \equiv \sum_{\boldsymbol{\xi}} C(\boldsymbol{\xi})\,[\xi_i(1-\xi_j) + (1-\xi_i)\xi_j]$, which represents the probability that loci i and j obtain their alleles s_i and s_j from different parents.

5.2. Gaussian ansatz and closure scheme

The aforementioned dynamical processes completely characterize the stochastic evolution of the genome population. In the following analysis, we will monitor the first two cumulants of the allele distribution over time:

$$\chi_i(t) = \langle s_i \rangle(t), \quad \chi_{ij}(t) = \langle s_i s_j \rangle(t) - \langle s_i \rangle(t)\langle s_j \rangle(t)\,, \tag{5.2}$$

where the averages are computed over the genomes in the population at time t, *i.e.* $\langle s_i \rangle = \frac{1}{N}\sum_{k=1}^{N} s_i^{(k)}$ and similarly for $\langle s_i s_j \rangle$. It should be noted that $\chi_{ii}(t) = 1 - \chi_i(t)^2$.

In the limit of infinite population size ($N \to \infty$), these moments satisfy deterministic first-order differential equations derived by Neher and Shraimann [Neher and Shraiman (2011a)] and presented in Eq. (4.15) and (4.19). As stated in Chapter 4, these equations are not closed as they involve higher-order moments in the allele variables (such as χ_{ijk}, χ_{ijkl}, ...). Below, we show how to overcome this problem by assuming that the allele distribution is Gaussian.

5.2.1 Gaussian measure over allele configurations

The key aspect of the present work is to impose a Gaussian constraint on the probability distribution of sequences $P(\mathbf{g},t)$. Specifically, we express it as follows[1]:

$$P(\mathbf{g},t) = \frac{1}{Z}\exp\left[-\frac{1}{2}\sum_{i,j}(s_i-\chi_i)(\boldsymbol{\chi}^{-1})_{ij}(s_j-\chi_j)\right], \tag{5.3}$$

where Z is the normalization factor and the matrix $\boldsymbol{\chi}$ represents the correlation matrix whose entries are $\chi_{ij} = \langle s_i s_j\rangle - \langle s_i\rangle\langle s_j\rangle$. This ansatz assumes that the population is distributed around an average sequence denoted by $\{\chi_0,\ldots,\chi_{L-1}\}$. One can visualize it as a cloud of sequences that are more or less similar to each other (see Figure 5.1).

In this Gaussian ansatz, the population is inherently monoclonal. However, this is not always the case, as there are scenarios where the population evolves with different dominant clones. Neher et al. also studied the emergence of clones in relation to the fitness landscape and recombination rate [Neher et al. (2013)]. Towards the end of this chapter, we will discuss the potential extension of this work to incorporate the evolution of multiple clones.

The crucial aspect of this ansatz is the ability to neglect cumulants of order higher than two. This information will allow us to derive closed versions of equations (4.15) and (4.19), which depend only on the χ_i's and χ_{ij}'s. To achieve this, it is necessary to consider the s_i's as real-valued variables. Otherwise, the three-point moments would not be zero, and the subsequent reasoning would not hold true. This additional condition still holds even when r and μ are sufficiently high, and the population can still be treated as monoclonal.

5.2.2 Closed equations for 1- and 2-point cumulants

As said above, the Gaussian approximation allows us to simplify the dynamical equations for χ_i and χ_{ij} by neglecting higher-order cumulants. In practice, we inject the Gaussian ansatz into Eqs. (4.15) and (4.19) and rewrite all higher order moments using Wick's theorem [Wick (1950)] as follows:

$$\langle s_i s_j s_k\rangle_{\neq} = \chi_i\chi_{jk} + \chi_j\chi_{ik} + \chi_k\chi_{ij} + \chi_i\chi_j\chi_k \tag{5.4}$$

$$\begin{aligned}\langle s_i s_j s_k s_l\rangle_{\neq} = {}& \chi_{ij}\chi_{kl} + \chi_{ik}\chi_{jl} + \chi_{jk}\chi_{il} + \chi_{ij}\chi_k\chi_l + \chi_{ik}\chi_j\chi_l + \chi_{il}\chi_j\chi_k + \\ & + \chi_{jk}\chi_i\chi_l + \chi_{jl}\chi_i\chi_k + \chi_{kl}\chi_i\chi_j + \chi_i\chi_j\chi_k\chi_l\ ,\end{aligned} \tag{5.5}$$

where the subscript $\neq$ indicates that all the indexes are different, otherwise they will result in lower order moments due to the binary nature of spins, $s_i^2 = 1$.

Within this Gaussian closure (GC) scheme we obtain the following set of $\frac{1}{2}L(L+1)$ coupled equations (with the convention $f_{ij} = f_{ji}$ if $i \neq j$ and $f_{ii} = 0$), see Appendix A for

[1] Eq. 5.3 assumes that the variables s_i are real numbers (intead of spins). To avoid this further assumption, one could alternatively define this approximation scheme by neglecting higher order cumulants of the probability distribution.

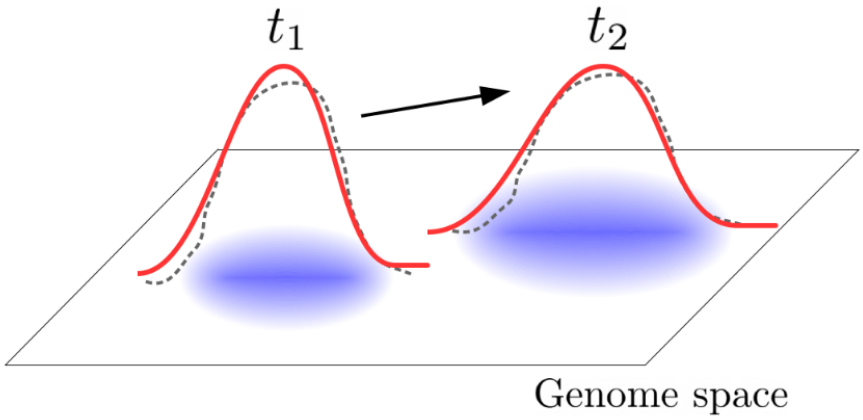


Figure 5.1: Sketch of the Gaussian closure scheme. The population is thought as a broad monoclonal cloud of sequences distributed around an average sequence $\{\chi_0, \ldots, \chi_{L-1}\}$ and which can be described as a Gaussian distribution. This cloud evolve over time due to a closed set of differential equations for the first and second order cumulants.

derivation,

$$\dot{\chi}_i = \sum_j \chi_{ij} \left[f_j + \sum_k f_{jk}\chi_k - 2f_{ij}\chi_i \right] - 2\mu\chi_i \,, \tag{5.6}$$

and, for $i \neq j$,

$$\begin{aligned} \dot{\chi}_{ij} &= -2\chi_{ij}\left[f_j\chi_j + f_i\chi_i\right] + \sum_{k\neq l} \chi_{ik} f_{kl} \chi_{jl} \\ &- 2\chi_{ij} \sum_k \left[f_{ki}(\chi_{ik} + \chi_i\chi_k) + f_{kj}(\chi_{jk} + \chi_j\chi_k) \right] \\ &+ 2f_{ij}\chi_{ij}\left[\chi_{ij} + 2\chi_i\chi_j\right] - (4\mu + rc_{ij})\chi_{ij} \,. \end{aligned} \tag{5.7}$$

As you can see, as far as this approximation holds, the evolution equations take a closed form and the dynamics is completely determined by the variables χ_i and χ_{ij}.

5.2.3 Case of additive fitness

In the absence of any epistatic contribution to the fitness ($f_{ij} = 0$ for all $i \neq j$), equation (5.7) for the second cumulant simplifies to

$$\dot{\chi}_{ij} = -(2f_j\chi_j + 2f_i\chi_i + 4\mu + rc_{ij})\chi_{ij} \,. \tag{5.8}$$

This equation has a single fixed point at $\chi_{ij} = 0$, meaning that recombination and mutations contribute to erase correlation between alleles, leading to linkage equilibrium (LE).

In this LE scenario, we also simplify Eq. (5.6) to obtain the following decoupled differential equations for the averages χ_i

$$\dot{\chi}_i = f_i(1 - \chi_i^2) - 2\mu\chi_i \,, \tag{5.9}$$

which can be solved analytically. A comparison between the theoretical prediction and a numerical simulation is shown in Figure 5.2.

To find the value of the first moment at equilibrium we simply have to solve $f_i(1-\chi_i^2) - 2\mu\chi_i = 0$. This solution will smoothly interpolates between $\chi_i = 2f_i/\mu$ at large mutation rate and $\chi_i = \text{sign}(f_i)$ at vanishingly small μ, which corresponds to monoclonality.

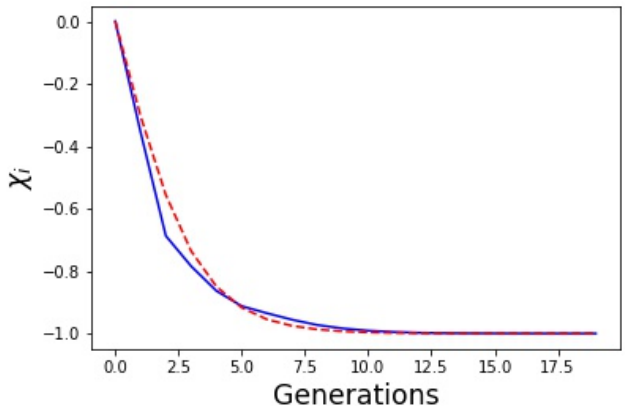


Figure 5.2: Allele dynamics in LE when epistasis is absent. The plot compares the simulated average of a spin χ_i (blue line) with its predicted evolution given by Eq. (5.9) (red dashed line). The simulation parameters were: $N = 10^5$, $L = 50$, $r = 0.3$, $\mu = 10^{-5}$.

5.2.4 Case of random epistatic contributions: comparison to numerical simulations

We now aim to test the accuracy of the Gaussian closure scheme at predicting the evolution of a population for more general fitness landscapes as done in [Mauri et al. (2021)]. To do this, we have first simulated the stochastic evolution of a population of $N = 10,000$ individuals. The fitness is defined by Eq. (4.4), with random quenched local biases f_i and epistatic couplings f_{ij} drawn from a normal distribution with zero mean and variances equal to, respectively, Δh^2 and Δf^2. After a transient time, the population reaches a stationary state in which we can compute the average values of the correlations χ_{ij}^{sim} over time. We then compare these values with the solution of the dynamical system (5.6) and (5.7) obtained by solving the equations numerically via Euler's method.

We also compare the results of the Gaussian closure scheme with the predictions of the large-r quasi-linkage equilibrium (QLE) theory from Neher and Shraiman (2011a). In particular, they estimate the evolution of the first and second cumulants as follows (corresponding to Eq. (27) in Neher and Shraiman (2011a)):

$$\begin{aligned} \dot{\chi}_i &= \sum_j \chi_{ij} \left[\hat{f}_j - \chi_i f_{ij} \right] \\ \dot{\chi}_{ij} &= \frac{(1-\chi_i^2)(1-\chi_j^2) f_{ij}}{2\hat{f}_i \chi_i + 2\hat{f}_j \chi_j + r c_{ij}}, \end{aligned} \tag{5.10}$$

where $\hat{f}_i \equiv f_i + \sum_j f_{ij}\chi_j$.

In Fig. 5.3 we show the scatter plot of the correlations obtained from simulations once stationarity is reached, χ_{ij}^{sim}, vs. the solutions of Eq. (5.7), χ_{ij}, for one generic sample. The Gaussian Ansatz correctly estimates small correlations, but is less accurate for large ones (in absolute values). It performs substantially better that the QLE approximation in the high-r regime from Neher and Shraiman (2011a), see red and blue dots and caption of Fig. 5.3 for details. Studying in more detail the scatter plot in Fig. 5.3, it seems to be a systematic bias (i.e. a tilt in the scatter plot) between the theretical and simulated correlations. This might be realated to the fact that our approximation scheme does not take into account effects of higher order moments that could in principle improve the prediction.

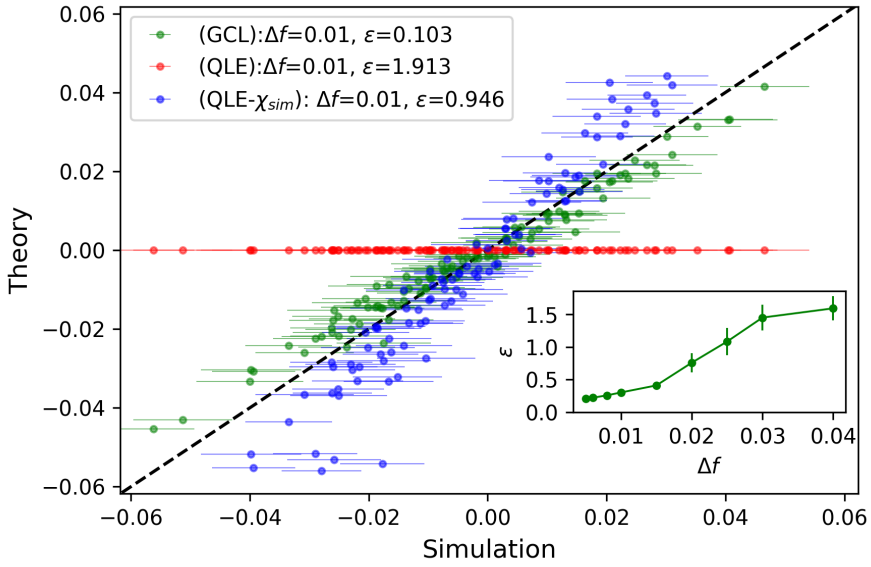


Figure 5.3: Scatter plot of the correlations χ_{ij} obtained from the Gaussian closure scheme (Eqs. (5.6) and (5.7)) (green) or their estimates from large–r QLE theory (red: eqs. (27) of Neher and Shraiman (2011a) for χ_i and χ_{ij}; blue: same equations for χ_{ij} only, replacing the χ_i's with their actual values χ_i^{sim} found in simulation to account for mutations) vs. simulations of finite populations (averaged over 200 generations once stationarity is reached) in the case of fully epistatic fitness landscape, see text and parameters in figure. Theoretical estimates of χ_{ij} are obtained by solving the dynamical system (5.6) and (5.7) within Euler approximation. Inset: rms z-score ϵ (5.11) as a function of the standard deviation Δf of the epistatic couplings. Parameters: $L = 15$, $\mu = 0.08$, $r = 0.85$, $\Delta h = 0.01$ and $c_{ij} = \frac{1}{2}(1-\rho^{|i-j|})$ with $\rho = 0.6$. Image and caption taken from Mauri et al. (2021).

To quantitatively measure the accuracy of each approximation scheme, we report in the inset of Fig. 5.3 the root mean square (rms) z-score,

$$\epsilon = \sqrt{\frac{2}{L(L-1)}\sum_{i<j}\left(\frac{\chi_{ij}-\chi_{ij}^{sim}}{\Delta\chi_{ij}^{sim}}\right)^2}, \tag{5.11}$$

as a function of Δf in the Gaussian closure. Here $\Delta\chi_{ij}^{sim}$ stands for the statistical error (standard deviation) of the correlation computed from simulation after stationarity is reached. For these values of μ and r, we obtain decent estimates of the correlations as long as Δf is not too large. In the next section, we will give a more precise estimate of the range of validity of the Gaussian closure scheme.

5.2.5 Validity of the Gaussian closure scheme

Here, we propose a self-consistent method to evaluate the regime of validity of the Gaussian closure scheme. The idea is to compute the moments of higher order $k \geq 3$ at stationarity whithin the Gaussian approximation and evaluate at which condition they are negligible compared to the second order moments.

The approximation scheme described in Section 5.2.2 can be used to compute moments of any order, such as $\chi_{i_1,i_2,\ldots,i_k} = \langle\prod_{\alpha=1}^{k}(s_{i_\alpha}-\chi_{i_\alpha})\rangle$ with $k \geq 3$. For simplicity, we define the variable a_k as $a_k = s_k - \chi_k$, $k = 1,\ldots,L$. In this way, we can write the k-th order moment in the stationary regime as

$$\chi_{i_1,i_2,\ldots,i_k} = \frac{1}{G_k}\left\langle \left\{ \prod_{\alpha=1}^{k} a_{i_\alpha} - \sum_{\alpha=1}^{k} (a_{i_\alpha} + \chi_{i_\alpha}) \langle \prod_{\beta(\neq\alpha)} a_{i_\beta} \rangle \right\} \times \left(\sum_i \hat{f}_i a_i + \sum_{i<j} f_{ij}(a_i a_j - \chi_{ij}) \right) \right\rangle \tag{5.12}$$

where $\hat{f}_i + \sum_{j(\neq i)} f_{ji}\chi_j$, $G_k = 2k\mu + r c_{i_1,i_2,\ldots,i_k}$, and $c_{i_1,i_2,\ldots,i_k}$ is the probability (computed with $C(\boldsymbol{\xi})$) that all k alleles do not come from the same parent (father or mother).

Upon careful analysis of the polynomial in the variables a_k, incorporating the substitutions $a_i^2 \to 1 - \chi_i^2 - 2\chi_i a_i$ to enforce $s_i^2 = 1$, and considering the effects of Wick's contraction, it becomes evident that the terms present in the average $\langle\cdot\rangle$ from Eq. (5.12) can be expressed as $M \times C \times (\chi_{ij})^E \times I$. Here, χ_{ij} represents a generic 2-point moment, and the following observations can be made:

1. the exponent E is given by $\frac{\ell}{2}$, where ℓ is an integer ranging from $k-2$ to $k+2$, based on the parity of k and the specific term being considered;
2. the combinatorial factor C equals $(E-1)!!$;
3. the multiplicity M is bounded by L^3;
4. the interaction term I is a polynomial of degree ≤ 2 involving two of the χ_{i_α}'s $(\alpha = 1...k)$ and is proportional to either a local fitness term $\hat{f}_i$ or an epistatic coupling f_{ij}.

The above observations allow us to conclude that, for large $k(\leq L)$, the k-th order moment is bounded from above by $\sim \exp(\log k!! + \frac{k}{2}\log|\chi_{ij}| + O(\log L))$, where $|\chi_{ij}|$ denotes the order of magnitude of the 2-point correlations. We now substitute the asymptotic scaling of the double factorial $\log k!! \sim \frac{k}{2}\log k - \frac{k}{2}$ and conclude that the k-th moment decays exponentially with k if

$$L \times |\chi_{ij}| < 1 \ . \tag{5.13}$$

In other words, the total amount of correlations that a site, say, i 'receives' from the $(L-1)$ other sites, say, j should be smaller than unity. This condition is sufficient and not necessary, as possible cancellation between terms of opposite signs have not been considered.

5.3. Application to short-range epistatic model

To further study the approximation scheme introduced in the previous section, we apply it to the short-range epistatic model proposed in [Mauri et al. (2021)]. This fitness landscape is fundamentally inspired by the one dimensional Ising model and is able to provide a non-trivial phenomenology despite being simple and analytically tractable. In practice, we write such fitness landscape as

$$F(\mathbf{g}) = f \sum_{i=1}^{L+1} s_i s_{i+1} \,, \tag{5.14}$$

where we have imposed periodic boundary conditions, $s_{L+1} = s_1$ (corresponding to haveing circular chromosome). A quick analysis shows that F is maximised when all the spins are

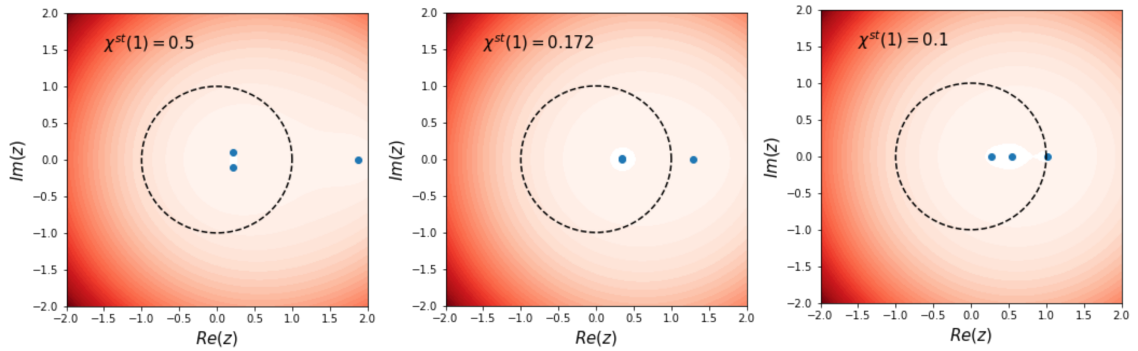


Figure 5.4: Roots of $\Delta(z, u = 0)$ in the complex plane for different values of the initial term $\chi^{st}(1)$ and for $a = 3$. For a specific intermediate value of $\chi^{st}(1)$ (see central panel), the two complex conjugated roots merges together on the real axis and the singularity can be removed from the square root of Eq. (5.20). Note that the fourth root is real and negative and is not shown in the figure. When $a < a_c \sim 2.542$ the third real root shown in the right panel is always inside the unit circle and the series in Eq. (5.21) does not converge.

aligned, corresponding to the two genomes $(+,+,+,...,+)$ and $(-,-,-,...,-)$, and minimised when they are anti-aligned, corresponding to $(+,-,+,...,+,-)$ and $(-,+,-,...,-,+)$. In general, this kind of model does not have a clear biological interpretation, but it was chosen mostly as a simple, but non trivial, case where the precision of our GC scheme can be studied in more detail.

In the following discussion we will focus on recombination correlations of the form $c_{ij} = \frac{1}{2}(1 - \rho^{|i-j|})$, where ρ is the inverse correlation length (also call crossover rate, see Eq. (4.18)). For the analytical treatment, we will consider the limit where $c_{ij} = 1/2$. Despite being less realistic, this choice allows us to obtain analytical results more easily in the large population limit ($N \to \infty$) and does not affect the qualitative behaviour of the model. Under this choice, we hereafter define the parameters $x = \mu/f$ and $y = r/(2f)$ and study the the probability distribution of the population as a function of those.

First of all, we inject this choice of F in the equations for the evolution of χ_i and χ_{ij} in the Gaussian closure scheme, Eqs. (5.6) and (5.7), which simplify to:

$$\dot{\chi}_i = f \sum_j \chi_{ij} \left[\chi_{j-1} + \chi_{j+1}\right] - 2f\chi_i \left[\chi_{ii-1} + \chi_{ii+1}\right] - 2\mu\chi_i \,, \tag{5.15}$$

$$\begin{aligned}\dot{\chi}_{ij} = {} & f \sum_k \left[\chi_{ki}\chi_{jk+1} + \chi_{k+1i}\chi_{jk}\right] + 2f\chi^2_{ii+1}\delta_{|i-j|,1} \\ & - 2f\chi_{ij}\left[\chi_{i-1i} + \chi_{ii+1} + \chi_{j-1j} + \chi_{jj+1}\right] - (4\mu + rc_{ij})\chi_{ij} \,.\end{aligned} \tag{5.16}$$

Starting from this system, we will consider the case where $\chi_i = 0$ (referred to as the *paramagnetic phase*) and study analytically the behaviour of χ_{ij} as a function of the distance between loci, $|i-j|$. Later, we study the stability of this solution and the conditions for the emergence of a *ferromagnetic phase* where $\chi_i \neq 0$.

5.3.1 Paramagnetic phase

In the absence of additive fields to the fitness, this model is invariant under a global spin flip, $s_i \to -s_i$. This implies that, in absence of spontaneous symmetry breaking, the average of the spins must be zero, $\chi_i = 0$. In this paramagnetic (PM) regime the correlations χ_{ij} only depend on the distance between loci, $d \equiv |i-j|$, due to the translational invariance

of the model. Hence, we can further simplify Eq. (5.16) by also taking advantage of the periodic boundary conditions defined in Eq. (5.14). This leads to the following set of equations for the stationary correlations $\chi_{ij} = \chi^{st}(d)$ (with $d \geq 1$):

$$\begin{aligned} \chi^{st}(d+1) &= [3\chi^{st}(1)+a]\chi^{st}(d) - \chi^{st}(1)^2\,\delta_{d,1} \\ &- \frac{1}{2}\sum_{k=0}^{d-1}\sum_{s=0,1}\chi^{st}(k+s)\chi^{st}(d-k-1+s) \\ &- \sum_{k=1}^{\infty}\sum_{s=0,1}\chi^{st}(k+s)\chi^{st}(d+k+1-s)\,, \end{aligned} \tag{5.17}$$

where $a = 2x + y/2$. Under the assumption that $\chi^{st}(d)$ decays quickly with the distance between loci, we can neglect the last term in the RHS of Eq. (5.18) (containing all the terms with $k \geq d+1$). This leads to a set of equations for the stationary correlations where each $\chi^{st}(d)$ only depends on the previous ones, $\chi^{st}(d-1)$, $\chi^{st}(d-2)$, etc. This set of equations can be solved iteratively, starting from $\chi^{st}(1)$, which is the only unknown.

At this point, we define the generating function of the correlations as follows:

$$\tilde{\chi}^{st}(z) \equiv \sum_{d\geq 0}\chi^{st}(d)z^d\,, \tag{5.18}$$

where z is a complex variable, and $\chi^{st}(d) = \frac{1}{d!}\partial_z^d\tilde{\chi}^{st}(z)|_{z=0}$. Within the approximation scheme described above, we can rewrite the generating function as follows:

$$\tilde{\chi}^{st}(z) = G(z, u=0) \tag{5.19}$$

$$\text{where } G(z,u) = \frac{z(3\chi^{st}(1)+a+4u) \pm \sqrt{\Delta(z,u)}}{1+z^2}\,, \tag{5.20}$$

while the term on the square root is given by $\Delta(z,u) = z^2(3\chi^{st}(1)+a+4u)^2 + (1+z^2)[(1-u)^2 - 2z(1-u)(2\chi^{st}(1)+4u+a) - 2z^2\chi^{st}(1)(\chi^{st}(1)+2u)]$.

The asymptotic behaviour of the correlations are controlled by the singularities of their generative function [Flajolet and Sedgewick (2009)]. In particular, if $\tilde{\chi}^{st}(z)$ behaves as $(z_c - z)^\beta$ close to its singularity, then

$$\chi^{st}(d) \sim z_c^{-d}\,d^{-(\beta+1)} \quad \text{for large } d\,. \tag{5.21}$$

It is important to note that the correlations will decay exponentially with the distance d only if $|z_c| > 1$. The singularities of $G(z, u=0)$ are given by the term $\sqrt{\Delta(z,u=0)}$ in Eq. (5.20), which fixes the exponent $\beta = 1/2$. Specifically, $\Delta(z, u=0)$ is a polynomial of the fourth order in z with two real roots, lying outside the unit circle (*i.e.* $|z| > 1$), and two complex conjugated roots lying in general inside the unit circle. Hence, the series in Eq. (5.21) does not converge in general. However, this can be corrected by carefully fixing the value $\chi^{st}(1)$ in such a way that the two conjugated roots merge together, resulting in a unique root with double multiplicity lying on the real axis. By also changing the sign in front of $\sqrt{\Delta}$ in Eq. (5.20), we can remove the corresponding singularity and ensure that the radius of convergence z_c of the generating function G is larger that unity. This procedure is illustrated in Figure 5.4.

The corresponding value of $\chi^{st}(1)$ and z_c as function of a are given in Figure 5.5. In particular, we observe that the radius of convergence z_c is larger that unity only when $a > a_c \sim 2.542$. For smaller values of a, there is no way for an exponential decay of $\chi^{st}(d)$,

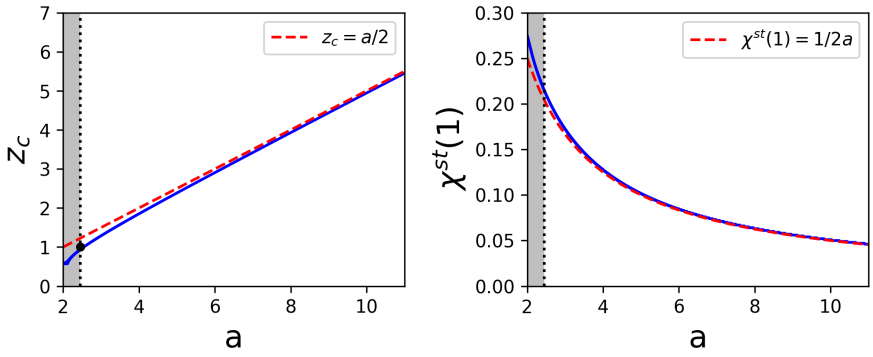


Figure 5.5: Radius of convergence z_c (left) and corresponding value of $\chi^{st}(1)$ (right) vs. $a = 2x + \frac{y}{2}$ in the PM regime. Red dotted lines show the large–a asymptotic behaviours. Gray region corresponds to $a < a_c \sim 2.542$, where the series in Eq. (5.21) does not converge in the PM regime. Image and caption taken from Mauri et al. (2021).

suggesting the existence of a different phase. As a final note on this point, we can also easily compute the asymptotic behaviour of z_c and $\chi^{st}(1)$ for large a. The results are shown in the legends of Figure 5.5 and are in agreement with the numerical results.

Similarly to what we have done in Figure 5.3, we can compare the expected correlations $\chi^{st}(d)$ in paramagnetic phase with the ones obtained from numerical simulations. The results are shown in Figure 5.6. Overall, we observe a good agreement between the two, although at large d it becomes more difficult to assess due to finite-size noise in the simulations (*i.e.* genetic drift).

5.3.2 Critical line and spontaneous symmetry breaking

As we shown above, the condition $a > a_c$ is necessary for the existence of a paramagnetic phase since it ensures the exponential decay of the correlations with distance, but in general is not sufficient. In fact, we have to check the stability of the PM solution under small fluctuations of the frequencies χ_i. To do this, we consider the linearised version of Eqs. (5.15) around the PM solution at fixed correlations $\chi_{ij} = \chi^{st}(|i-j|)$ which is given by

$$\dot{\chi}_i = -2f \sum_j \mathcal{M}_{ij} \chi_j \,, \tag{5.22}$$

$$\text{where } \mathcal{M}_{ij} = \left[2\chi^{st}(1) + x\right] \delta_{ij} - \frac{1}{2}\left[\chi^{st}(|j-i+1|) + \chi^{st}(|j-i-1|)\right] .$$

It is easy to note that $\mathcal{M}$ is translational invariant. Hence, we can digonalize it by Fourier plane waves. The corresponding eigenvalues λ_n are given by, for $n = 0, 1, ..., L-1$,

$$\lambda_n = 2\chi^{st}(1) + x - \left[2\mathrm{Re}\tilde{\chi}^{st}(e^{-i\frac{2\pi n}{L}}) - 1\right] \cos\frac{2\pi n}{L} \,, \tag{5.23}$$

where $\tilde{\chi}^{st}(z)$ is the generating function defined in Eq. (5.18). The PM solution is stable if all the eigenvalues have negative real part. In practice, we only need to check when the largest eigenvalue, λ_0, vanishes to define the boundary of the PM phase. This condition is equivalent to the following equation for x:

$$x = 2\tilde{\chi}^{st}(1) - 2\chi^{st}(1) - 1 \,. \tag{5.24}$$

This critical line is shown in Figure 5.7 together with the phase diagram of the model in the (x, y) plane. In Figure 5.7 we also compared the results evaluated in the GC scheme,

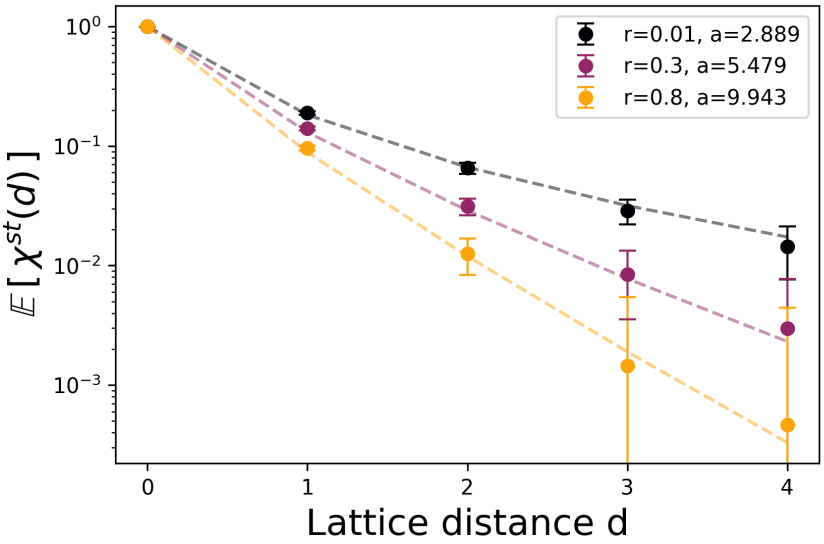


Figure 5.6: Average correlations in the PM phase for a finite population (dots) against the value computed from Gaussian closure eqs. (5.6) and (5.7) (dashed lines) through Euler approximation, for three recombination rates r. Parameters: $f = 0.028, L = 50, \mu = 0.03, N = 10^4$, $c_{ij} = \frac{1}{2}(1-\rho^{|i-j|})$ with $\rho = 0.6$; results are obtained by averaging over 5000 generations. Image and caption taken from Mauri et al. (2021).

valid at $N \to \infty$ with simulation of finite (but large) populations. In particular, for each value of x and y, we have computed the Edwards–Anderson overlap $q = \frac{1}{L}\sum_i \chi_i^2$ from the simulations and report it in the color map. As expected, q takes very low values to the right of the critical line, corresponding to the PM phase, and increases as we move to the left, corresponding to the emergence of a ferromagnetic (FM) phase. Here the mutation rate is to weak to ensure the average values of the spins to be zero and symmetry will be spontaneusly broken by genetic drift.

5.3.3 Ferromagnetic phase

In the new ferromagnetic phase (FM), where the simmetry is broken, the average values of the spins are not expected to be zero anymore. In this case, we suppose $\chi_i = \chi_{FM}$, up to a global reversal symmetry. With this new assumption, we can repeat the same steps as in Section 5.3.1 to obtain the stationary correlations $\chi^{st}(d)$, for $d \geq 1$, and the corresponding generating function $\tilde{\chi}^{st}(z)$. The correct value of χ_{FM} can be obtained by imposing the stationarity condition $\dot{\chi}_i = 0$ in Eq. (5.15) in the FM regime. This leads to the following equation for χ_{FM}:

$$\begin{aligned} \chi_{FM}^2 &= 1 - x + 2\sum_{d\geq 2} \chi^{st}(d) \\ &= x - \left[2\tilde{\chi}^{st}(1) - 2\chi^{st}(1) - 1\right] , \end{aligned} \tag{5.25}$$

where the generating function of the correlations is now given by $\tilde{\chi}^{st}(z) = G(z, \chi_{FM}^2)$, see Eq. (5.20).

As in the PM phase, we chose carefully the value of $\chi^{st}(1)$ in order to remove the spurious singularities inside the unit circle and garantee the exponential decay of the correlations with the distance. In this way, we can solve self-consistently Eqs. (5.20) and (5.25) to obtain the value of χ_{FM} as a function of x and y. The results are shown in

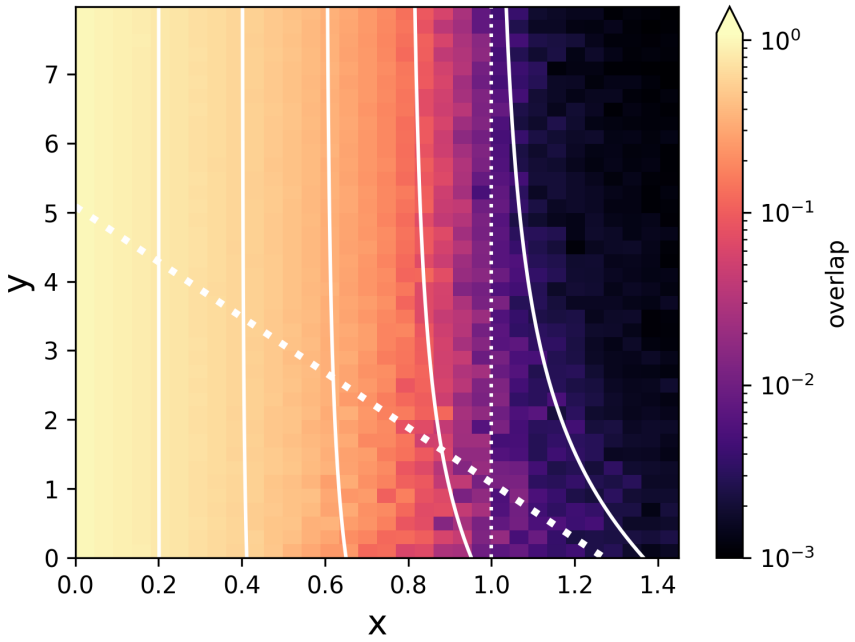


Figure 5.7: Phase diagram for genomes evolving under fitness (5.14) in the $(x=\mu/f, y=r/(2f))$ plane. The heatmap shows the Edwards–Anderson overlap q, see text and color bar, for a simulated population of $N=10^4$ individuals in the case of non-homogeneous recombination matrix $c_{ij}=\frac{1}{2}(1-\rho^{|i-j|})$ with $\rho=0.6$. The theoretical white lines were derived for long-range recombination correlation $c_{ij}=\frac{1}{2}$, see main text. The critical line $x=2\tilde{\chi}^{st}(1)-2\chi^{st}(1)-1$ (rightmost solid curve) separates PM (right) and FM (left) phases, and approaches the $x=1$ line (thin dashed) when $y\gg 1$. Theoretical contour lines corresponding to $\chi^2_{FM}=0,.2,.4,.6,.8$ (from right to left) are shown in the FM phase (full lines), in fair agreement with numerical findings for the short-range model, see color code. The thick dashed line stands for $a=a_c\sim 2.542$, showing that $a>a_c$ throughout the PM phase. Image and caption taken from Mauri et al. (2021).

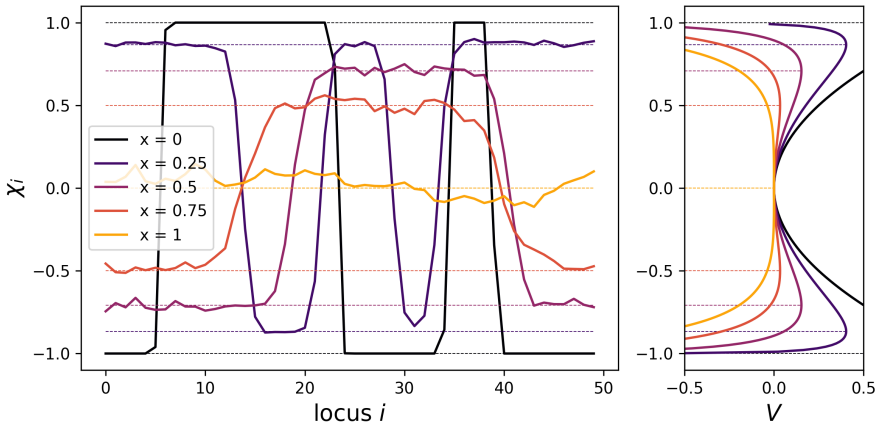


Figure 5.8: Left: Allele frequencies χ_i for one population evolved over 500 generations in the FM phase and for different values of the ratio x. Parameters: $f = 0.028, N = 10^4, L = 50, r = 0.85$ and $c_{ij} = \frac{1}{2}(1-\rho^{|i-j|})$ with $\rho = 0.6$. Dashed horizontal lines show $\pm\sqrt{1-x}$. Right: effective potential $V(\chi)$ for respective value of x shown in the legend and horizontal lines locating the maxima of the potential in $\chi_{FM} = \pm\sqrt{1-x}$. Image and caption taken from Mauri et al. (2021).

Figure 5.7. In particular, we observe that χ_{FM} is zero in the PM phase and increases as we move to the left, corresponding to the FM phase. The critical line separating the two phases is given by the condition $\chi_{FM} = 0$, which is equivalent to Eq. (5.24). We also note that for a large range of values of x and y, the correlations decay very quickly in agreement with QLE, and we have

$$\chi_{FM} \sim \sqrt{1-x}. \tag{5.26}$$

according to Eq. (5.25). Finally, we have numerically checked that the correlations in the FM and PM phase respect the condition of Eq. (5.13) for the validity of the Gaussian closure scheme, *i.e.* $2\sum_{d\geq 1}\chi^{st}(d) < 1$.

5.3.3.1 Short term dynamics: allele domains

Running simulations of the model in the FM phase, we observe that, after a transient, the χ_i's break the simmetry of the model and reach apparently stationary values. In particular, we observe the emergence of up and down domains of height $\pm\chi_{FM}$ as you can see in Figure 5.8. These domains are separated by characteristic domain walls and have variable lengths.

To get a better understanding of this phenomenon, we can characterise the shape of the domain walls in the high recombination limit, $r \to \infty$. If x is close to one and the length of the genome is high ($L \gg 1$), we can define a smooth, continuum limit for the index i which can be substituted with a continuous variable z. In this approximation, we have $\chi_{i+1} + \chi_{i-1} - 2\chi_i \approx \chi''(z)$ in the stationary regime, where $\chi(z)$ is the continuous version of χ_i. In addition, according to Eq. (5.6), we have

$$\chi''(z) = \frac{2x\chi(z)}{1-\chi^2(z)} - 2\chi(z)\ , \tag{5.27}$$

which is the equation of motion of a particle in a potential $V(\chi) = x\log(1-\chi^2) + \chi^2$. More precisely, we can integrate Eq. (5.27) to obtain $\frac{1}{2}\chi'(z)^2 + V(\chi(z)) = E$, where E is a constant of motion. Hence, finding the stationary profile of the allele frequency is equivalent to solve a mechanical problem of a unit mass particle moving in a potential $V(\chi)$ in the fictitious time z.

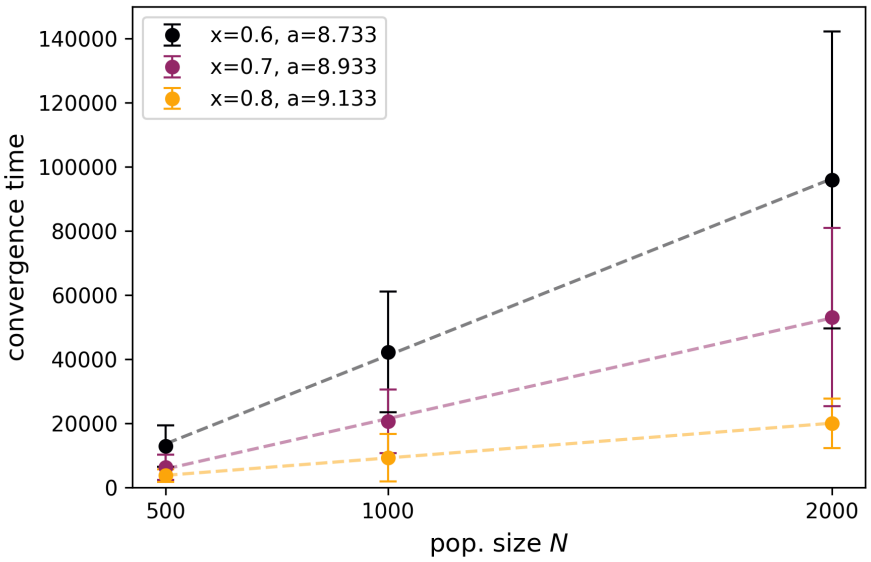


Figure 5.9: FM phase: Nb. of generations required for convergence towards the uniform frequency state $\chi = \chi_{FM}$ vs. population size N for different values of x. The dashed lines stand for the linear regression of the data. Each point is the average over 10 simulations. Parameters: $r = 0.85$, $f = 0.028$, $L = 50$, $c_{i\neq j} = \frac{1}{2}$. Image and caption taken from Mauri et al. (2021).

The potential V (which is shown in Figure 5.8(*right*)) is symmetric in 0, has two maxima in $\chi = \pm\chi_{FM}(x)$ and a local minimum in $\chi = 0$. Domain profiles are then defined by trajectories $\chi(z)$ with conserved energy $E = V(\chi_{FM})$ and depend on x. The results within this approximation are in good agreement with the numerical simulations, as shown in Figure 5.8(*left*).

5.3.3.2 Long term dynamics: domain coarsening

In the previous section, we have shown that at intermediate times the χ_i's evolve to form domains of up and down spins. However, in the long term, the domains are observed to encounter a diffusive process and merge when they encounter. The system reaches eventually a final stationary state with uniform frequency $\pm\chi_{FM}$. In Figure 5.9, we show the chracterisitc time needed by the system to converge into this uniformly magnetized state.

The coarsening process can be understood by the following argument. The typical size of domains grows with time as $\sim t^{1/2}$ according to Allen-Cahn theory of coarsening for non-conserved fields [Bray (2002)]. Hence, a single domain is expected to occupy the whole genome in a time $\sim L^2/D$, where D is the diffusion coefficient which is expected to be inversely proportional to the population size N. As a consequence, we expect the characteristic coarsening time to scale as

$$\tau \sim \tau_0 \times L^2 \times N \ , \tag{5.28}$$

where τ_0 is a constant that depends on the intensive parameters such as x. This scaling is corroborated by the numerical results reported in Fig. 5.9.

5.4. Final remarks on the Gaussian closure approach

In this section, we aim to provide a summary of the main results obtained in this chapter and discuss potential directions for future research. We have introduced a Gaussian

closure scheme to model the dynamics of multi-allele correlations in a population of evolving genomes. Our approach describes the distribution of genomes at time t as a cloud of covariance matrix $\chi_{ij}(t)$ around an 'average' genome characterized by allele frequencies $\chi_i(t)$. The resulting 1- and 2-point correlations are governed by a set of $\frac{1}{2}L(L+1)$ coupled deterministic, non-linear first-order differential equations, where L represents the genome length.

For simplicity, we initially assumed a binary state for alleles, but our framework can be readily extended to accommodate multi-categorical (Potts) variables, enabling the study of more complex real-world systems [Gao et al. (2019)]. Our scheme can be viewed as a practical approach to explore the quasi-linkage equilibrium (QLE) regime beyond the traditional large-r perturbative approach initiated by Kimura [Kimura (1965)]. By considering contributions to all orders in powers of $1/r$, our equations, including the expression for χ_{ij} presented in Eq. (5.7), encompass the QLE expressions at high recombination [Zeng et al. (2021)]. Moreover, they provide insights into novel behaviors that emerge at intermediate values of recombination and mutation rates, as well as diverse fitness functions.

We have extensively analyzed a specific model within this framework, shedding light on intriguing phenomena. Notably, our model consists of two competing genomes with distinct allele compositions, yet they possess equal and maximal fitness. At sufficiently low mutation rates, we observed a phase transition from a paramagnetic (PM) phase, where the distribution encompasses both fittest genomes, to a ferromagnetic (FM) regime, where one genome eventually dominates. Interestingly, in both the PM and FM phases, correlations decay exponentially with the distance between alleles along the genome, which aligns with their distance on the epistatic interaction graph. Remarkably, these phases formally fall within the QLE regime, even for moderate values of recombination (r) and mutation (μ) rates. However, unlike the conventional informal definition of QLE, our scheme accounts not only for allele frequencies χ_i but also for correlations χ_{ij}, which arise due to the finite values of r and μ.

While this chapter presents significant contributions, several aspects warrant further investigation. Firstly, the range of validity for closure schemes is typically a complex matter, as exemplified in small chemical systems [Schnoerr et al. (2015)]. Nonetheless, we have derived a sufficient condition, as shown in Eq. (5.13), which establishes the validity of our approach. However, this condition may break down in scenarios where the genome distribution exhibits multiple modes and correlations are not negligible, such as in the Clonal Competition (CC) regime [Neher et al. (2013)]. Nevertheless, even in the presence of CC, it is conceivable that our scheme remains suitable for describing the local distribution of genomes associated with individual clones. Hence, exploring mixtures of Gaussian Ansätze, each representing a clone, could be a fruitful approximation, particularly for situations involving a small number of clones and alleles.

Secondly, this framework offers a practical avenue for inferring the fitness determinants (f_i, f_{ij}) from statistics obtained from sequence data. Epistasis, notoriously challenging to infer solely from mean fitness observations [McCandlish et al. (2015); Otwinowski et al. (2018)], necessitates approaches that incorporate correlations. Notably, we will discuss in the next chapter how the Gaussian closure scheme introduced here has demonstrated successful reconstruction of epistatic contributions from synthetic data [Zeng et al. (2021)]. We envision that this approach will provide a solid population genetics foundation to advance inverse statistical methods for protein sequence data [Cocco et al. (2018)].

A final remark concerns the analytical results obtained within the Gaussian approximation. In the absence of recombination ($y = 0$) our fitness model is equivalent to a quantum

Ising chain in transverse field and the transition takes place at $x = \mu/f = 1$ [Baake et al. (1997)].

6

Inferring epistasis in the QLE regime

In Chapter 4, we introduced the quasi-linkage equilibrium (QLE) regime, also known as the KNS theory, following the foundational work of Kimura (1965) and Neher and Shraiman (2011a). The KNS theory proposes that in scenarios where recombination dominates over selection and mutations can be neglected, the population evolves into a mono-clonal phase characterized by a broad distribution of alleles at different loci, with weak interactions governed by the epistatic effects between them. Energy-based models, as presented by Zeng and Aurell (2020), provide a means to reconstruct epistasis from population data within this framework.

In contrast, Chapter 5 explores an alternative approach to address the QLE regime beyond the high-recombination scenario. Building upon our original work [Mauri et al. (2021)], we demonstrate the utility of a simple Gaussian ansatz in describing the dynamics of the population through a closed set of differential equations that capture the evolution of allele frequencies and correlations. This approximation exhibits good agreement with numerical results across a wide range of parameters within the QLE regime, and its analytical treatment is feasible for simple fitness landscapes.

In the context of extending the applicability of QLE, this chapter aims to investigate the recovery of epistatic effects using Direct Coupling Analysis in a broader regime where mutations cannot be neglected and are comparable to recombination. We present two approaches for improving the formulas used in epistasis reconstruction. The first approach extends the KNS perturbative theory to account for high mutation rates, while the second approach utilizes the Gaussian closure approximation introduced in the previous chapter. We then compare the accuracy of these different methods using data obtained from numerical simulations of evolution, employing the FFPopSim library [Zanini and Neher (2012)], and varying the parameter choices. This comparison allows us to explore how these approaches extend the range of parameters for which epistasis can be inferred from population data. Additionally, we investigate the impact of population size and directional selection on the accuracy of the inference. Finally, we discuss the potential applications of this approach in inferring epistasis from real genomic data.

This chapter is primarily based on our work in collaboration with Prof. Erik Aurell, Doct. Hong-Li Zeng, and Vito Dichio. The results presented here have been published in the paper *Inferring epistasis from genomic data with comparable mutation and outcrossing rate* [Zeng et al. (2021)].

6.1. QLE outside high-recombination

We now present the two possible ways to extend the applicability of QLE outside the high-recombination regime as presented in [Zeng et al. (2021)]. To help the reader, we recall from

Chapter 4 that we consider genomes of binary-valued alleles $s_i = \pm$. In QLE we suppose the probability distribution of the genomes to take the form of a Gibbs-Boltzmann distribution of an Ising model (or Potts model in categorical data), $\log P(\mathbf{g},t) \sim \sum_i \phi_i(t) s_i + \sum_{ij} J_{ij}(t) s_i s_j$, up to a normalization constant. Under this assumption, the dynamics of the population is given by $\mathrm{d}\log P(\mathbf{g},t)/\mathrm{d}t = F(\mathbf{g}) - \langle F\rangle + \mu M(\mathbf{g},t) + rR(\mathbf{g},t)$, where $F(\mathbf{g})$ is the fitness of the genome $\mathbf{g}$, μ and r are respectively the mutation and recombination rates and M and R account for the effects of mutations and recombination according to Eq. (4.23) and are given by

$$M(\mathbf{g},t) = \sum_i \left[\frac{P(M_i\mathbf{g},t)}{P(\mathbf{g},t)} - 1\right], \tag{6.1}$$

$$R(\mathbf{g},t) = \sum_{\boldsymbol{\xi},\mathbf{g}'} C(\boldsymbol{\xi}) P(\mathbf{g}',t) \left[\frac{P(\mathbf{g}^{(m)},t)P(\mathbf{g}^{(f)},t)}{P(\mathbf{g},t)P(\mathbf{g}',t)} - 1\right]. \tag{6.2}$$

6.1.1 Extension of KNS perturbative theory

In KNS theory, we saw that the couplings J_{ij} can be supposed to be small, leading to the expansion of $R(\mathbf{g},t) \sim \sum_{i<j} c_{ij} J_{ij} [(s_i\langle s_j\rangle + s_j\langle s_i\rangle) - (s_i s_j + \langle s_i s_j\rangle)]$, where c_{ij} is the crossover probability defined in Eq. (4.18). This assumption is valid when the recombination rate r is high, but it should also hold for non-negligible mutation rates μ. In addition, for high mutation rate μ, we can suppose that also the fields ϕ_i's are small. By rewriting $M(\mathbf{g},t)$ as

$$M(\mathbf{g},t) = \sum_i \left[e^{-2\phi_i s_i - 2\sum_j J_{ij} s_i s_j} - 1\right], \tag{6.3}$$

we can expand both ϕ_i and J_{ij} from the exponential to obtain

$$M(\mathbf{g},t) \sim -2\sum_i \phi_i s_i - 4\sum_{i<j} J_{ij} s_i s_j\,. \tag{6.4}$$

Following the same steps as in Section 4.2.1, we inject the expansions of M and R into the master equation (4.23) for the probability distribution $P(\mathbf{g},t)$ and separate the dependencies on s_i and $s_i s_j$. In this way, we obtain the following set of equations for the evolution of the fields ϕ_i and couplings J_{ij}:

$$\dot{\phi}_i = f_i + r\sum_{j\neq i} c_{ij} J_{ij}\langle s_j\rangle - 2\mu\phi_i\,, \tag{6.5}$$

$$\dot{J}_{ij} = f_{ij} - (4\mu + rc_{ij}) J_{ij}\,, \tag{6.6}$$

where f_{ij} is the epistatic interaction between loci i and j in the fitness function F. Hence, the interactions J_{ij} will quickly evolve through the stationary solution $J_{ij}^{st.} = f_{ij}/(4\mu + rc_{ij})$. Inverting the latter equation, we obtain the extended inference formula of the epistatic interactions

$$f_{ij}^* = J_{ij}^*(4\mu + rc_{ij})\,, \tag{6.7}$$

where the star denotes the inferred values from the data. J_{ij}^* can be obtained from Direct Coupling Analysis using methods such as *pseudo-likelihood maximization* or *naive mean-field inference* (corresponding to the Gaussian approximation described in Section 2.3.3) which amounts to matrix inversion of the empirical correlation matrix.

6.1.2 The argument by Gaussian closure

The second approximation scheme is based on the Gaussian closure scheme presented in Chapter 5. We recall that the Gaussian closure is based on the assumption that the probability distribution of the genome population can be approximated by a multivariate Gaussian distribution of the alleles. This leads to a close set of equations for the average allele frequencies and correlations, respectively $\chi_i = \langle s_i \rangle$ and $\chi_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$. The dynamics of the population is then given by the set of equations (5.6) and (5.7) that we recall here to be

$$\dot{\chi}_i = \sum_j \chi_{ij} \Big(f_j + \sum_k f_{jk} \chi_k - 2 f_{ij} \chi_i \Big) - 2\mu \chi_i \tag{6.8}$$

$$\begin{aligned} \dot{\chi}_{ij} = & -2\chi_{ij} \sum_k \Big[f_{ik}(\chi_{ik} + \chi_i \chi_k) + f_{jk}(\chi_{jk} + \chi_j \chi_k) \Big] + 2 f_{ij} \chi_{ij} (\chi_{ij} + 2\chi_i \chi_j) + \sum_{k,l} f_{kl} \chi_{ik} \chi_{jl} + \\ & - (4\mu + r c_{ij}) \chi_{ij} - 2\chi_{ij} (f_i \chi_i + f_j \chi_j) \, . \end{aligned} \tag{6.9}$$

In principle, Eqs. (6.8)-(6.9) could be simultaneously solved in order to determine the stationary state, which is of our interest, and this in turn would allow to determine the $L(L+1)/2$ quantities $\{f_i\}, \{f_{ij}\}$ as a function of the $\{\chi_i\}, \{\chi_{ij}\}$. Unfortunately, considering the size of the system, this is analytically not feasible.

Nevertheless, eq. (6.9) suggests another route to infer f_{ij} according to the following argument: when studying the stationary state, we can assume self-consistently that all the off-diagonal χ_{ij} are small, so that we can expand χ_{ij}, with $i \neq j$, as a power series of $\varepsilon \equiv 1/(4\mu + r c_{ij})$:

$$\chi_{ij} = \varepsilon \chi_{ij}^{(1)} + \varepsilon^2 \chi_{ij}^{(2)} + \varepsilon^3 \chi_{ij}^{(3)} + \mathcal{O}(\varepsilon^4) \, , \tag{6.10}$$

Injecting this ansatz in Eq. (6.9) at stationarity (*i.e.* setting $\dot{\chi}_{ij} = 0$) and evaluating separately each order of ε, we obtain all the corrections $\chi_{ij}^{(n)}$. In particular, in the absence of additive fitness ($f_i = 0$ for all i), we obtain [Zeng et al. (2021)]:

$$\chi_{ij}^{(1)} = f_{ij} \tag{6.11}$$

$$\chi_{ij}^{(2)} = 2 \sum_k f_{ik} f_{jk} \tag{6.12}$$

$$\chi_{ij}^{(3)} = \sum_{k<l} f_{kl}(f_{ik} f_{jl} + f_{jk} f_{il}) - f_{ij}^3 + \sum_k \Big[f_{ik} (2 \sum_l f_{kl} f_{jl} - f_{ij} f_{ik}) + f_{jk} (2 \sum_l f_{kl} f_{il} - f_{ij} f_{jk}) \Big] \tag{6.13}$$

In the more general case where $f_i \neq 0$ for all i, we find the first two orders in Eq. (6.10) to be [Zeng et al. (2021)]:

$$\chi_{ij}^{(1)} = f_{ij} (1 - \chi_i^2)(1 - \chi_j^2) \tag{6.14}$$

$$\begin{aligned} \chi_{ij}^{(2)} = & \sum_k f_{ik} (\chi_{jk}^{(1)} + \chi_{ik}^{(1)} \chi_i \chi_j - \chi_i \chi_k \chi_{ij}^{(1)}) - \sum_{k,l} f_{kl} \chi_i \chi_l \chi_{jk}^{(1)} - \sum_l f_{il} \chi_l \chi_i \chi_{ij}^{(1)} + \\ & + \sum_{k<l} f_{kl} \Big(\chi_{ik}^{(1)} \chi_j \chi_l + \chi_{il}^{(1)} \chi_j \chi_k - 2 f_i \chi_i \chi_{ij}^{(1)} + f_{ij} \chi_i \chi_j \chi_{ij}^{(1)} \Big) + \{ i \longleftrightarrow j \} \end{aligned} \tag{6.15}$$

where, for the sake of clarity, in the last equation we have left implicit the terms like $\chi_{ij}^{(1)}$ as specified in Eq. (6.14), while the term $\{ i \longleftrightarrow j \}$ corresponds to taking the RHS of the equation and exchanging the loci i and j.

Defining the standard deviation of the distribution of the epistatic interactions as $\sigma(\{f_{ij}\})$, we observe that each correction is of the order $L\times\sigma(\{f_{ij}\})$ with respect of the lower one, where L is the length of the genome. This suggests that the expansion is well behaved as long as $L\times\sigma(\{f_{ij}\})\ll 1$.

If we only consider the first order correction in the general case of non-zero additive fitness, we obtain that, to the first order,

$$\chi_{ij}=\frac{f_{ij}}{4\mu+rc_{ij}}(1-\chi_i^2)(1-\chi_j^2)\ . \tag{6.16}$$

Turning around this into an inference formula for fitness we arrive at [Zeng et al. (2021)]

$$f_{ij}^{*,SIE}=J^{*,SIE}\cdot(4\mu+rc_{ij})=\frac{\chi_{ij}}{\left((1-\chi_i^2)\left(1-\chi_j^2\right)\right)}\cdot(4\mu+rc_{ij})\ , \tag{6.17}$$

where we have defined $J^{*,SIE}\equiv\chi_{ij}/((1-\chi_i^2)(1-\chi_j^2))$, which corresponds to the *small-interaction expansion* (SIE) method for DCA introduced by Neher and Shraiman (2011a). In general, the SIE method is not very accurate as a general DCA method [Neher and Shraiman (2011a); Gao et al. (2019)],but has the advantage of being easy to compute.

6.1.3 Remarks on the validity of the inference procedures

In KNS theory, they study QLE in the regime of high-recombination limit and negligible mutation rate. This allows to avoid treating the term $M(\mathbf{g},t)$ in the master equation (4.23) and requires only to expand the recombination term $R(\mathbf{g},t)$ which only depends on the couplings J_{ij}. For finite populations, QLE is obtained for mutations much weaker than recombinations, while still being non-zero. This is necessary to avoid the fittest genotype to take over the population and make the QLE phase only a long-lived transient [Gao et al. (2019); Zeng and Aurell (2020)]. In this regime, the first order moments χ_i are typically different than zero. Differently stated, in the KNS theory we are not enforcing any condition on the local fields ϕ_i.

In the approximation scheme presented in Section 6.1.1, we enforce μ to be large in order to expand the fields ϕ_i from the mutation term $M(\mathbf{g},t)$ in the master equation (4.23). This in principle might be a limitation of that procedure since is enforcing the fields (and consequently the frequencies χ_i) to be small.

The Gaussian approximation instead allows for non-zero averages χ_i. We have already discussed the validity condition of Gaussian closure in the previous chapter (Section 5.2.5). In particular, the approximation is valid whenever the correlations stays low (*i.e.* $L\times|\chi_{ij}|<1$). As we described in the previous section, this point can be enforced by requiring that the standard deviation of the epistatic interactions $\sigma(\{f_{ij}\})$ is small, qualitatively $L\sigma(\{f_{ij}\})<1$. We will see below how the inference formula (6.17) does not work for $L\sigma(\{f_{ij}\})\approx 1$. Moreover, the additive fitness should be sufficiently weak as well to make sure the population is strictly mono-clonal which is one of the assumptions of the Gaussian closure [Mauri et al. (2021)].

6.2. Simulation strategies and results

We now compare the accuracy of the inference formula $f_{ij}^*=J_{ij}^*\cdot rc_{ij}$ from KNS theory with the extension presented in the past sections, $f_{ij}^*=J_{ij}^*\cdot(4\mu+rc_{ij})$, as done in [Zeng et al. (2021)]. The basic idea is to simulate the states of a population with N individuals (genome sequences) evolving under mutation, selection and recombination and genetic drift. As previously done by Zeng and Aurell (2020), we have used the FFPopSim package

developed by Zanini and Neher for this purpose [Zanini and Neher (2012)]. Simulation and parameter settings are given in Appendix B.

In practice, we let a population evolve using FFPopSim for a certain choice of the parameters μ, r and c_{ij} and for a given realization of the fitness landscape F. After a suitable relaxation period, we take the population data and use it to infer couplings J_{ij} using DCA. In particular, we might use naive mean-field ($J_{ij}^{*,nMF} = -(\chi^{-1})_{ij}$) or small-interaction expansion ($J_{ij}^{*,SIE} = \frac{\chi_{ij}}{(1-\chi_i^2)(1-\chi_j^2)}$) to do DCA. Later, we use the inference formulas from KNS and our theory with the inferred couplings to reconstruct the epistatic interactions and compare them with the original ones.

Here, the testing epistatic fitness is given by a Sherrington-Kirkpatrick model [Sherrington and Kirkpatrick (1975)], corresponding to interactions f_{ij} drawn from a normal distribution with different variances, $\sigma^2(\{f_{ij}\})$. The additive fitness f_i follows Gaussian distribution with zero means and the standard deviation $\sigma(\{f_i\}) = 0.05$ in our simulations. We note that (4.28) is proposed to be hold for weak selection and high recombination, and has already been tested in [Zeng and Aurell (2020)]. Data availability is an issue. As in Zeng and Aurell (2020), we have used *all-time* versions of the algorithms, where samples $\mathbf{g}^{(s)}(t)$ at different t are pooled. This is primarily to mitigate the effect that in a real-world population the number of individuals N is very large, but in the simulations it is only moderately large. All DCA methods as well as empirical correlations can be more accurately estimated with more samples.

As a final remark, below we will always compare the KNS inference formula with couplings inferred with nMF ($f_{ij}^* = J_{ij}^{*,nMF} \cdot rc_{ij}$) with the SIE expansion derived from Gaussian closure ($f_{ij}^* = J_{ij}^{*,SIE} \cdot (4\mu + rc_{ij})$). Using $J^{*,nMF}$ instead of $J^{*,SIE}$ in the latter formula would lead to similar results, as shown in Appendix B.

6.2.1 Mutation vs recombination rate

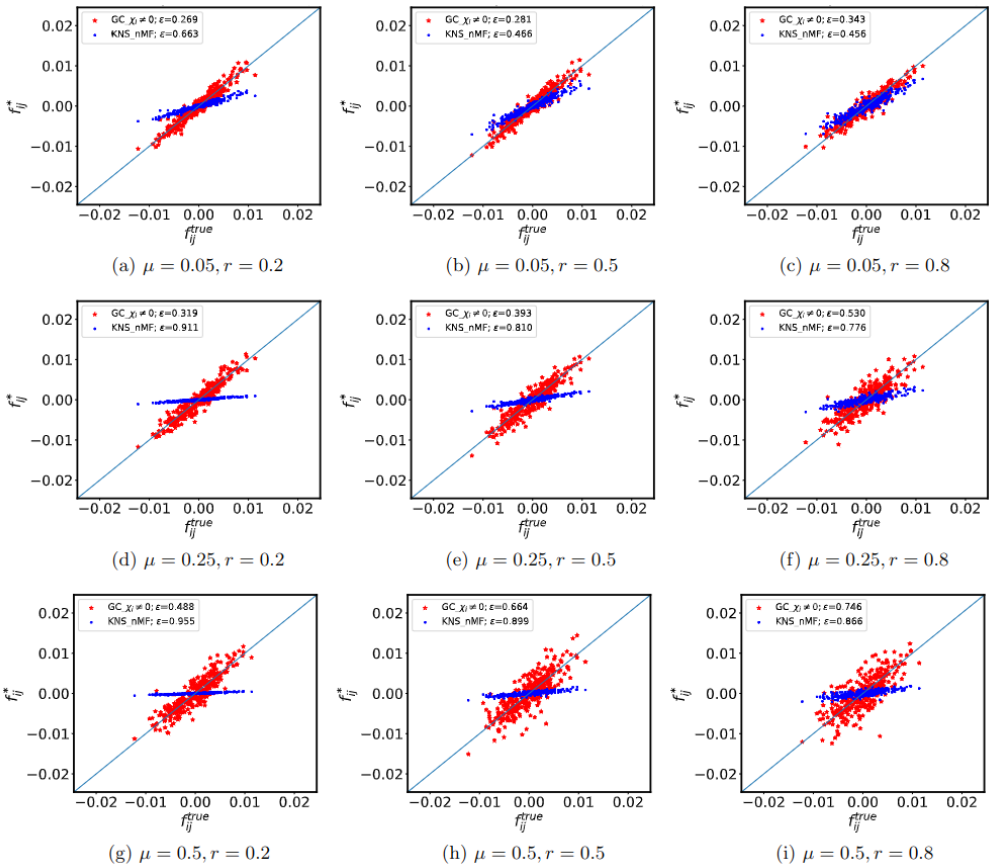
We start by taking a fixed fitness landscape (same f_{ij}) and systematically vary mutation and recombination rates (μ and r). Each sub-figure in large Fig. 6.1 shows scatter plots for the KNS fitness inference formula (4.28), $f_{ij}^* = J_{ij}^* \cdot rc_{ij}$, and the formula (6.17), $f_{ij}^* = \frac{\chi_{ij}}{(1-\chi_i^2)(1-\chi_j^2)} \cdot (4\mu + rc_{ij})$, based on Gaussian closure vs the model parameter f_{ij} used to generate the data. These model parameters were independent Gaussian random variables specified by their standard deviation $\sigma(\{f_i\})$ and $\sigma(\{f_{ij}\})$ as hyper-parameters. The parameters J_{ij}^* which enter (4.28) are inferred by naive mean-field (nMF).

The variations in Fig. 6.1 are such that each column has the same recombination rate in the order low-medium-high from left to right, and each row has the same mutation rate in the order low-medium-high from top to bottom. In the top row both inference formulae work well, particularly for high recombination rate at the top right. In the middle and bottom rows the KNS formula does not work while the formula based on Gaussian closure still performs well, and in particular does not have systematic errors.

For comparison in more extensive parameter ranges we can quantify inference performance by normalized root of mean square error

$$\epsilon = \sqrt{\frac{\sum_{ij}\left(f_{ij}^* - f_{ij}\right)^2}{\sum_{ij} f_{ij}^2}} \tag{6.18}$$

This reduces all the information in the scatter plots in Fig. 6.1 to one single number. Although we have not observed such behavior, it is conceivable that inference could be very accurate for most pairs (i, j) such that ϵ is small, but still have large errors for some



(a) $\mu = 0.05, r = 0.2$ (b) $\mu = 0.05, r = 0.5$ (c) $\mu = 0.05, r = 0.8$

(d) $\mu = 0.25, r = 0.2$ (e) $\mu = 0.25, r = 0.5$ (f) $\mu = 0.25, r = 0.8$

(g) $\mu = 0.5, r = 0.2$ (h) $\mu = 0.5, r = 0.5$ (i) $\mu = 0.5, r = 0.8$

Figure 6.1: Scatter plots for testing and recovered f_{ij}s with mutation rate μ and recombination rate r. r increases from left to right columns (0.2, 0.5 and 0.8 respectively) while μ enlarge from top to bottom (0.05, 0.25 and 0.5 respectively). The red stars for the Gaussian closed theory $f^*_{ij} = \chi_{ij} \cdot (4\mu + rc_{ij}) / \Big((1-\chi_i^2)(1-\chi_j^2)\Big)$; blue dots for KNS $f^*_{ij} = rc_{ij} \cdot J^{*,nMF}_{ij}$. Other parameters: $\sigma(\{f_i\}) = 0.05$, $\sigma(\{f_{ij}\}) = 0.004$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, number of generations $T = 10,000$. Inference by Gaussian closed theory works in much wider parameter range than KNS. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

few pairs. An overall value ϵ much less than one hence does not guarantee that fitness inference is accurate for all pairs. On the other hand, a large mean square error could correspond to either systematic or random errors in the scatter plots. Both behaviors that can be observed.

To improve the discussion, we plot the reconstruction error ϵ against the mutation rate times the average coalescence time $\langle T_2 \rangle$ in Figure 6.2. The pair coalescence time T_2 between two sequences is the distance in time from their common ancestor and it can be computed in the latest version of the FFPopSim algorithm. The meaning of $\mu\langle T_2 \rangle$ is roughly the average distance between diverged, coevolving sequences. We see from Figure 6.2 that the reconstruction error is minimized when $\mu\langle T_2 \rangle \sim L$, meaning that the inference is most accurate when sequences are maximally diverging from each other and phylogenetic biases are lost. For higher values of μ the reconstruction error increases, as expected, since the signal of epistasis is lost in the noise of genetic drift.

Coming back to comparison between KNS theory and its extension for high mutation rates, we report the reconstruction error ϵ vs. different values of μ and r as a heat map in

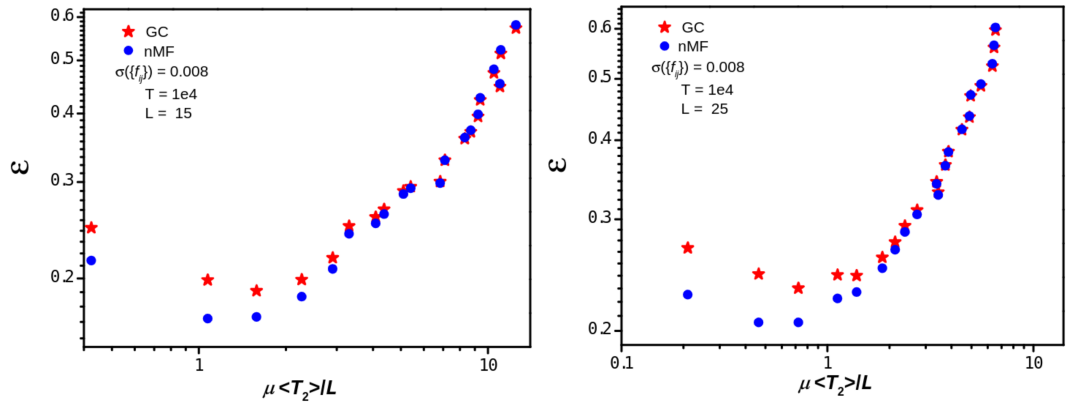


Figure 6.2: Epistasis reconstruction error ϵ versus $\mu\langle T_2\rangle/L$. Red stars for $f^*_{ij} = \chi_{ij}\cdot(4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$ while blue dots for $f^*_{ij} = J^{*,nMF}_{ij}\cdot(4\mu + rc_{ij})$. Epistasis f_{ij} are inferred best with $\mu = 0.05$. The other parameter values: $\sigma(f_i) = 0.05$, $\sigma(f_{ij}) = 0.008$, carrying capacity $N = 200$, out-crossing rate $r = 0.5$, cross-over rate $\rho = 0.5$, number of loci $L = 15$, generations $T = 10,000$. 10 realizations of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

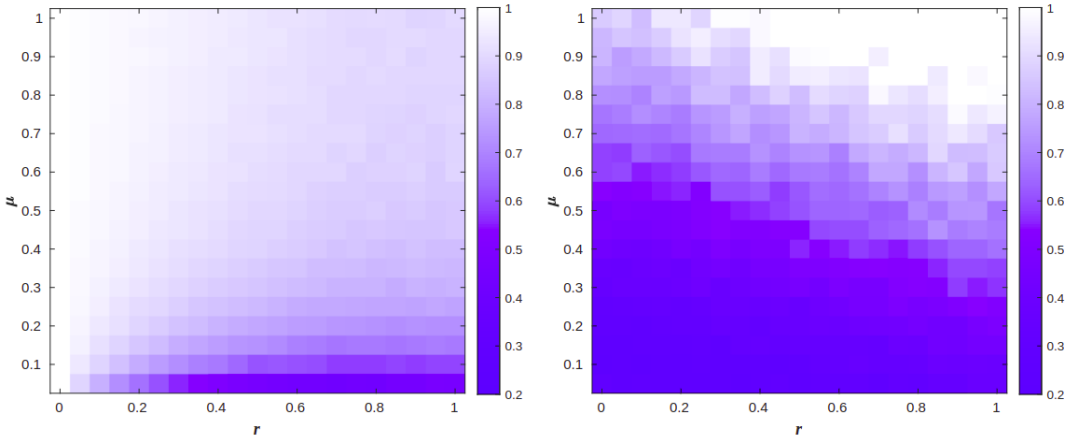
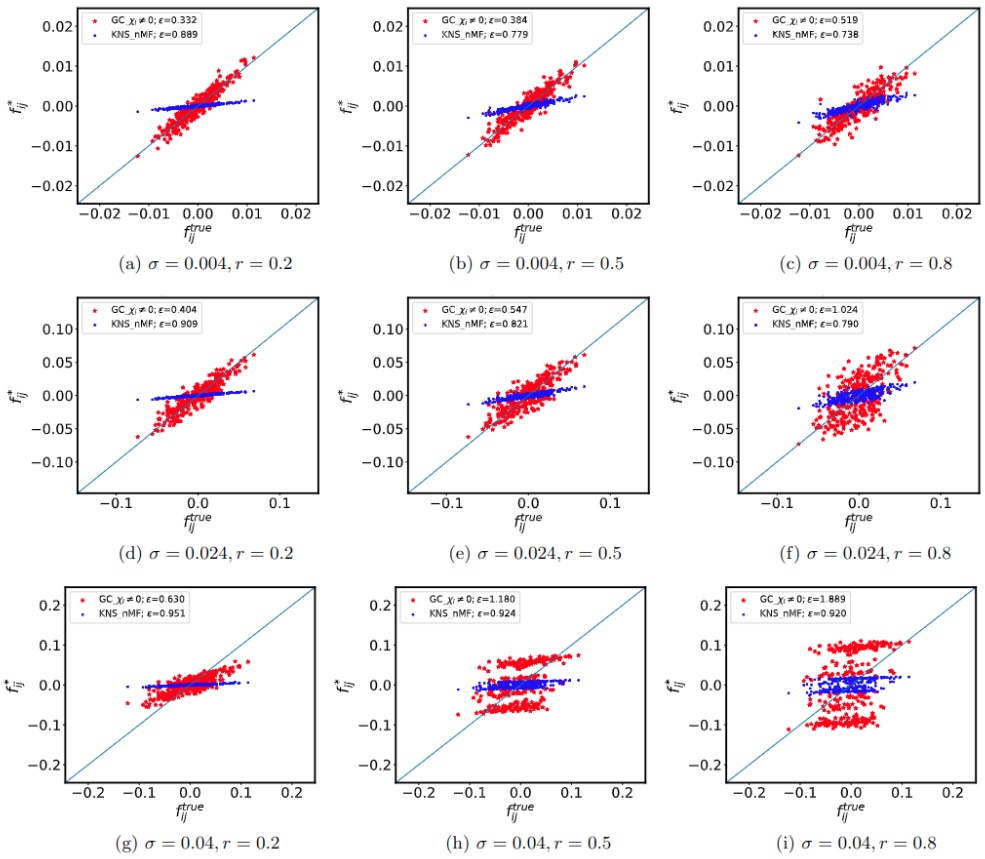


Figure 6.3: Phase diagram for mutation rate μ versus recombination rate r. The color is encoded by the reconstruction error ϵ given in eq. (6.18). Left: KNS theory $f_{ij} = J^{*,nMF}_{ij}\cdot rc_{ij}$. Right: Gaussian closed theory $f_{ij} = \chi_{ij}\cdot(4\mu + rc_{ij})/\Big((1-\chi_i^2)(1-\chi_j^2)\Big)$. Parameters:$\sigma(\{f_i\}) = 0.05$, $\sigma(\{f_{ij}\}) = 0.004$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, generations $T = 10,000$. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).



(a) $\sigma = 0.004, r = 0.2$ (b) $\sigma = 0.004, r = 0.5$ (c) $\sigma = 0.004, r = 0.8$

(d) $\sigma = 0.024, r = 0.2$ (e) $\sigma = 0.024, r = 0.5$ (f) $\sigma = 0.024, r = 0.8$

(g) $\sigma = 0.04, r = 0.2$ (h) $\sigma = 0.04, r = 0.5$ (i) $\sigma = 0.04, r = 0.8$

Figure 6.4: Scatter plots for testing and reconstructed f_{ij}s. The standard deviation $\sigma(\{f_{ij}\}^{true})$ increases from top to bottom rows (0.004, 0.024 and 0.04 respectively) and recombination rate r enlarges in columns from left to right (0.2, 0.5 and 0.8 respectively). Red stars for $f_{ij}^* = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$ and blue dots for $f_{ij}^* = J_{ij}^{*,nMF} \cdot rc_{ij}$. Both inference formulae do not work for large σ and high r, where strong correlations emerge between loci that drive the system out of the QLE phase Dichio (2020); Dichio et al. (2021), as shown in (g), (h) and (i). The other parameter values: standard deviation $\sigma(\{f_i\}) = 0.05$, mutation rate $\mu = 0.2$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, generations $T = 10,000$. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

Figure 6.3. Again, we use respectively KNS formula with couplings obtained from nMF and the formula from Gaussian closure. We kept the number of generations for each simulation constant at $T = 10,000$. Similarly to Figure 6.1, we see large difference between the two inference procedures. In particular, the KNS formula works only for low mutation rate and high recombination rate (Fig. 6.1(c)), while the new formula from Gaussian closure works for a much larger region with weak fitness. For stronger mutation rate and larger recombination rate, the root mean square error (ϵ) of inference based on the Gaussian closure formula increases, while KNS theory present severe systematic error. A possible explanation of this phenomenon is that random noise from genetic drift is too strong for any reconstruction to be accurate.

6.2.2 Fitness variations vs recombination rate

We continue by varying recombination r and the dispersion in the fitness landscape (f_{ij} drawn from Gaussian distributions with different hyper-parameters $\sigma(\{f_{ij}\})$). Each

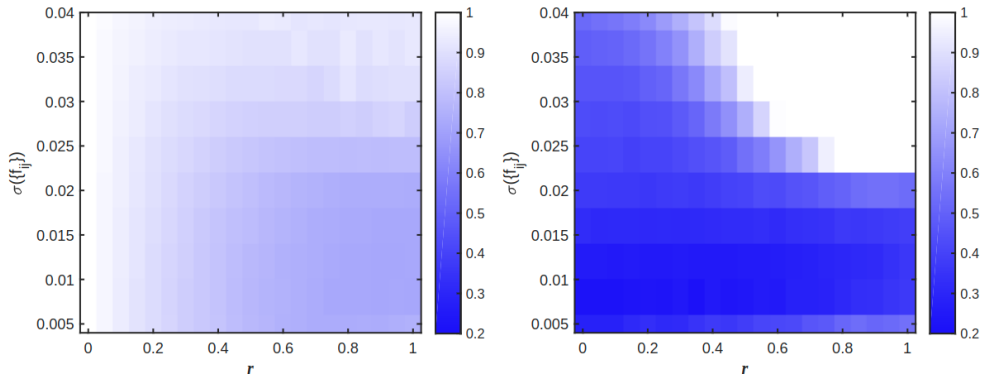


Figure 6.5: Phase diagram for the standard deviation $\sigma(\{f_{ij}\})$ versus recombination rate r. Left: KNS theory $f^*_{ij} = J^{nMF}_{ij} \cdot rc_{ij}$. Right: Gaussian closed theory $f^*_{ij} = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$. Parameters: mutation rate $\mu = 0.2$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, generations $T = 10,000$. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

sub-figure in Fig. 6.4 shows scatter plots for the two epistatic fitness inference formulae for the model parameter $\sigma(\{f_{ij}\})$ vs recombination rate r. The order in the Fig. 6.4 is increasing recombination rate r in columns from left to right, and increasing $\sigma(\{f_{ij}\})$ in rows from top to bottom. Here, the mutation rate $\mu = 0.2$ and the other parameters are the same with those tested in Fig. 6.1.

Overall, the KNS formula (4.28) does not work for any of the parameter values shown in Fig. 6.4 with mutation rate $\mu = 0.2$.

This either because of systematic errors as in top row (low fitness dispersion) and left column (low recombination), or due to the emergence of strong correlation between loci that drive the evolution out of the QLE regime [Dichio (2020); Dichio et al. (2021)], as in bottom right corner Fig. 6.4(h) and Fig. 6.4(i). The Gaussian closure formula (6.17) in contrast works well for low recombination or low fitness dispersion, or both but fails as well for sufficiently high recombination and fitness strength.

As above we have quantified inference performance in larger parameter ranges by the root of mean square error ϵ. The phase diagrams in Fig. 6.5 show again that the Gaussian closure formula works except when r and $\sigma(\{f_{ij}\})$ are both large, while the KNS formula does not work in any range with mutation rate $\mu = 0.2$.

6.2.3 The effect of population size

The effects of genetic drift on epistasis effects are studied through the inference error ϵ with different population sizes N. It is presented in a semi-log plot as shown in the main panel of Fig. 6.6. The red stars are for the epistasis inference error given by eq. (4.28) $f^*_{ij} = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$ while blue dots for eq. (6.7) $f^*_{ij} = J^{*,nMF}_{ij} \cdot (4\mu + rc_{ij})$. There is a clear trend that both methods work better with increasing population sizes. However, eq. (4.28) works slightly better when the population size is less than 400 while eq. (6.7) recovers the epistasis better when $N > 400$. The inserts (a) and (b) of Fig. 6.6 show the scatter plots for the recovered and testing epistasis f_{ij}s with $N = 25$ (equal number with that of locus in an individual sequence) and $N = 6400$ respectively. Clearly both Eqs. recover the epistasis better with large population size.

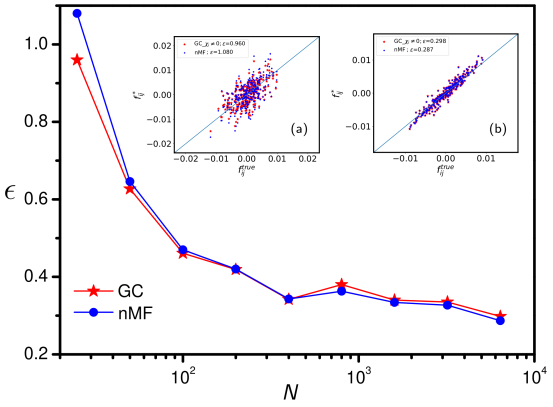


Figure 6.6: Semi-log plot for epistasis reconstruction error ϵ versus the average size of population N. Insert (a): scatter plot for testing and reconstructed f_{ij}s with $N = L = 25$; Insert (b): scatter plot with $N = 6400$. Red stars for $f^*_{ij} = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$ and blue dots for $f^*_{ij} = J^{*,nMF}_{ij} \cdot (4\mu + rc_{ij})$. Epistasis f_{ij} are recovered roughly better with increasing N. The other parameter values: $\sigma(f_i) = 0.05$, $\sigma(f_{ij}) = 0.004$, mutation rate $\mu = 0.25$, out-crossing rate $r = 0.5$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, generations $T = 10,000$. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

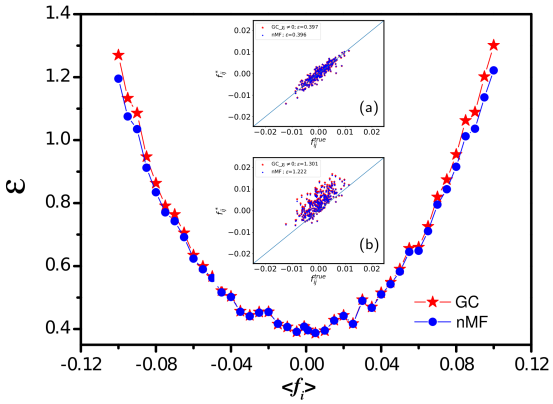


Figure 6.7: Epistasis reconstruction error ϵ versus the means of Gaussian distributed additive fitness $\langle f_i \rangle$. Insert (a): scatter plot for testing and reconstructed f_{ij}s with $\langle f_i \rangle = 0.001$; Insert (b): scatter plot with $\langle f_i \rangle = 0.01$. Red stars for $f^*_{ij} = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_j^2))$ and blue dots for $f^*_{ij} = J^{*,nMF}_{ij} \cdot (4\mu + rc_{ij})$. The epistasis reconstructions are getting worse with stronger directional fields. The other parameter values: standard deviation $\sigma(\{f_{ij}\}) = 0.004$, mutation rate $\mu = 0.25$, out-crossing rate $r = 0.5$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, number of generations $T = 10,000$. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image and caption taken from Zeng et al. (2021).

6.2.4 Epistasis inference with directional selection

In this section, we report the effect of additive fitness to epistasis reconstruction. Here the additive effects f_is are Gaussian distributed with non-zero means and the standard deviations are fixed as $\sigma(\{f_i\}) = 0.05$. The red stars for the epistasis inference with Gaussian closure eq. (6.17) while the blue dots for the revised KNS method by eq. (4.28). The inserts of Fig. 6.7 show the scatter plots for the recovered and testing epistasis effects with (a): $\langle f_i \rangle = 0.001$ and (b): $\langle f_i \rangle = 0.01$ respectively. The other parameters for each points in the main panel are as follows: standard deviation of the pairwise epistasis fitness $\sigma(\{f_{ij}\}) = 0.004$ and that of the single-locus additive fitness $\sigma(\{f_i\}) = 0.05$, mutation rate $\mu = 0.25$, out-crossing rate $r = 0.5$, cross-over rate $\rho = 0.5$, number of loci $L = 25$, carrying capacity $N = 200$, generations $T = 10,000$.

Both methods recover the tested epistasis better with weaker means of additive fitness compared with that following stronger directional selections. It is notable that the reconstructed epistasis have a roughly corrected trends with large additive fitness, as shown in the inner panel (b) of Fig. 6.7 for $\langle f_i \rangle = 0.01$. This may indicates the revision of the epistasis inference formulae in our work for stronger directional selections.

Reconstruct plausible evolutionary paths from data

Summary

Part II, in summary, focuses on understanding how data are shaped by evolution. In this third part, we explore how data, especially probabilistic models trained on them, can be leveraged to reconstruct plausible evolutionary processes that led to the emergence of this data. The primary emphasis will be on finding mutational paths between two homologous proteins with different functions. The problem of building transition path is not only useful from the evolutionary perspective but also from the perspective of protein engineering. Indeed, it is desirable to find a consistent way to modify the biological activity of a protein by applying a series of mutations that do not disrupt the protein's structure and function.

Chapter 7 introduces a novel method to sample mutational paths between two homologous proteins using probabilistic models such as Restricted Boltzmann Machines. Our method can effectively sample paths with predicted high fitness and is benchmarked against state-of-the-art numerical methods like AlphaFold and ProteinMPNN [Jumper et al. (2021); Dauparas et al. (2022)].

In Chapter 8, we provide a more detailed characterization of mutational paths in the context of mean-field theory. We demonstrate how mean-field approximations can compute relevant statistics of the paths, such as the total number of relevant paths (i.e., the entropy of the system), the frequency of each amino acid along the path, and transition probabilities under plausible evolutionary processes with given mutation rates. Additionally, we apply this scheme to different protein models based on RBMs.

Finally, Chapter 9 focuses on analyzing a simple toy-model of paths that can be analytically solved using mean-field theory. Particularly, we uncover the existence of a phase transition separating a regime in which paths are stretched between their anchors from another regime where paths can explore the energy landscape more globally to minimize the energy.

7

Sampling paths with sequence-based models of proteins

In the third part of this thesis, we will focus on the second main topic of this research project. While in Part II we discussed extensively the effect of evolution in shaping the data and in which condition energy-based models, such as DCA, can be used to reconstruct the shape of the fitness landscape, in this part we will focus on the (somehow opposite) question: How can we use the information contained in the data to reconstruct the evolutionary paths that led to the observed sequences? Stated differently, for each couple of sequences in the data set we aim to reconstruct the most likely succession of mutations that connects them with their most recent common ancestor.

This is a rather difficult problem, since there is no consistent way to verify the correctness of the reconstructed paths in real data. A slightly different question we can ask is whether it is possible to construct transition paths (*i.e.* succession of mutations) that connect two homologous proteins by keeping the functionality of all the intermediates high. This is a more tractable problem, since we are assuming that sequence-based models of proteins (as previously discussed in Part II) can be used as proxy of the fitness landscape driving the evolution of the data.

We notice that this is harder than the problem of designing *de novo* functional proteins which we discussed in Chapter 2, since many sites have a fundamental role in keeping the structure and/or the functionality of the protein and mutating them may result in the complete death of the molecule (see Figure 7.1). Designing paths of proteins is a problem that received little attention in the past years, see however Tian and Best (2020). Yet solving this problem would shed light on the navigability of the sequence landscape [Greenbury et al. (2021)] and on how functional specificity, such as binding to distinct substrates could have emerged from ancestral, promiscuous proteins in the course of evolution [Khersonsky and Tawfik (2010)]. In turn, it could help design new proteins interpolating between functional classes.

Due to the huge number of possible paths mutagenesis experiments generally restrict to *direct* paths going through the 2^D mutants containing the amino acids appearing in the two edge sequences (differing on D sites) [Poelwijk et al. (2019)]. However, constraining paths to be direct may preclude the discovery of much better *global* paths, involving mutations and their reversions and reaching more favorable regions in the sequence space. Mutational models have demonstrated that exploring the fitness landscape beyond the direct space can enhance adaptation Wu et al. (2016). Additionally, the existence of such beneficial 'global' mutations could provide valuable insights into the properties of the fitness landscape, *e.g.* the presence of high fitness regions responsible for the deviation of the paths from the direct subspace.

While various methods exist for building transition paths between the minima of

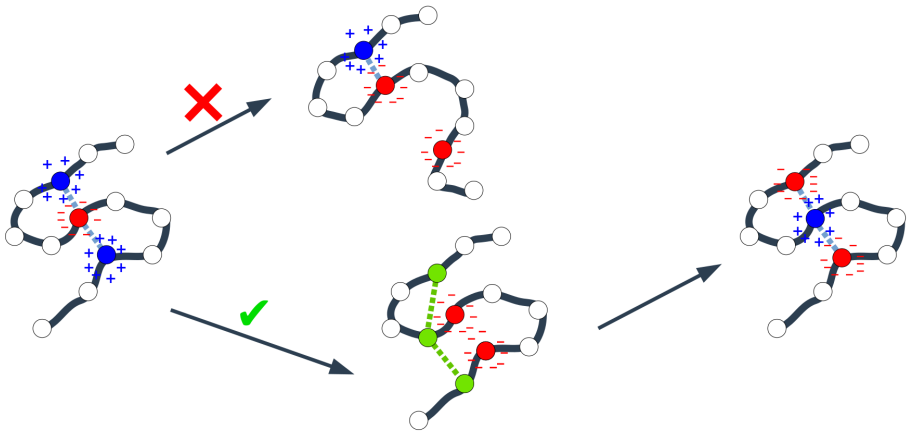


Figure 7.1: Sketch of transition path between two proteins with the same folded structure. Fundamental contact are realised by electrostatic bonds between oppositely charged amino acids. Mutating these sites may result in the complete death of the molecule. In contrast, we can circumnavigate this issue by mutating other sites in such a way that the structure is preserved (for example by introducing novel di-sulfide bonds between cysteines, here colored in green).

a multi-dimensional continuous landscape [Vanden-Eijnden et al. (2010); Bolhuis et al. (2002)] dealing with discrete configurations requires the development of specific procedures [Mora et al. (2012)]. The goal of this chapter is to introduce a Monte Carlo algorithm to sample mutational paths in protein landscapes, *e.g.* obtained by Restricted Boltzmann Machines trained on sequence data, but which can be easily generilized to any probabilistic model of sequences. We first benchmark our sampling procedure on an exactly solvable model of lattice proteins [Lau and Dill (1989); Shakhnovich and Gutin (1990)], and demonstrate its capability to find high-quality paths between two proteins belonging to different subfamilies. We then apply our algorithm to the WW domain, a binding module involved in the regulation of protein complexes [Sudol (1996); Socolich et al. (2005)]. The functionality of the sequences along the paths is validated with structure (ligand+protein)-informed software ProteinMPNN [Dauparas et al. (2022)], which we already presented in Chapter 3. In the next chapters, we will show how we can deeper characterize transition paths using mean-field theory and discuss the conditions under which global paths can be actually preferred to direct ones.

The results presented in this chapter have been published in the paper titled *Mutational paths with sequence-based models of proteins: from sampling to mean-field characterization* [Mauri et al. (2023a)].

7.1. Definition and sampling of mutational paths

We assume the sequence landscape is modeled through a probability distribution $P_{\text{model}}(\mathbf{v})$ over amino-acid sequences $\mathbf{v}$ of length N. Informally speaking, P_{model} quantifies the probability that $\mathbf{v}$ is a member of the protein family of interest, *i.e.* share its common structural and functional properties. As introduced in Chapter 2, exact expressions for $P_{\text{model}}(\mathbf{v})$ are not avilable, but approximate distributions can be inferred from multi-sequence alignments (MSA) using unsupervised learning techniques.

Hereafter, we use Restricted Boltzmann Machines (RBM) [Fischer and Igel (2012); Tubiana et al. (2019)] to model the sequence landscape. We recall from Section 2.3 that RBMs define a joint probability distribution of the protein sequence $\mathbf{v}$ (carried by the visible layer) and of its M-dimensional latent representation $\mathbf{h}$ (present on the hidden

layer) as

$$P_{\mathrm{RBM}}(\mathbf{v},\mathbf{h}) \propto \exp\left(\sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} h_\mu I_\mu(\mathbf{v}) - \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu)\right) , \tag{7.1}$$

where $I_\mu(\mathbf{v}) = \sum_i w_{i,\mu}(v_i)$ is the input to hidden unit μ. The g_i's and $\mathcal{U}_\mu$'s are local potentials acting on, respectively, visible and hidden units, and the $w_{i\mu}$'s are the interactions between the two layers. These hyperparameters are learned by maximising the marginal probabilities $P_{\mathrm{model}}(\mathbf{v}) = \int d\mathbf{h}\, P_{\mathrm{RBM}}(\mathbf{v},\mathbf{h})$ over the sequences $\mathbf{v}$ in a multi-sequence alignment of the family. In practice, we will use Persistent Contrastive Divergence (PCD) [Tieleman (2008)] with a L_1^2 regularization term over the weights $\propto \sum_\mu \left(\sum_{i,v} |w_{i,\mu}(v)|\right)^2$ and a L^2 regularization over the fields $\propto g_i(v)^2$ [Tubiana et al. (2019)] to train RBMs. The reason we chose this class of models is that it offers a convenient way to monitor the changes in sequences along mutational paths, as we will see in the next sections.

7.1.1 Probability of mutational paths

We consider mutational paths of T steps, $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{T-1}\}$, anchored at their extremities defined by the sequences $\mathbf{v}_{\mathrm{start}}$ and $\mathbf{v}_{\mathrm{end}}$. In this chapter, we aim to sample paths that are as smooth as possible, *i.e.* paths that do not contain too many mutations at each step. Moreover, we want to sample paths that are likely to be functional, meaning that the predicted probability given by our model, $P_{\mathrm{model}}(\mathbf{v})$, is high along the enntire path. To this end, we define the probability of a path as follows:

$$\begin{aligned}\mathcal{P}[\mathcal{V}|\mathbf{v}_{start}, \mathbf{v}_{end}] \propto \prod_{t=1}^{T-1} P_{\mathrm{model}}(\mathbf{v}_t) \times \\ \pi(\mathbf{v}_{start}, \mathbf{v}_1) \times \prod_{t=1}^{T-2} \pi(\mathbf{v}_t, \mathbf{v}_{t+1}) \times \pi(\mathbf{v}_{T-1}, \mathbf{v}_{end}) ,\end{aligned} \tag{7.2}$$

where the 'transition' factor $\pi(\mathbf{v},\mathbf{v}')$ increases with the similarity between the sequences $\mathbf{v},\mathbf{v}'$. In practice we choose $\pi = 1$ if the two sequences are identical, $e^{-\Lambda}$ if they differ by one mutation (with $\Lambda > 0$), and 0 if they are two or more mutations apart. This choice generates 'continuous' paths, along which successive sequences differ by one mutation at most. Other choices for π, more plausible from an evolutionary point of view will be introduced in Chapter 8.

In the above choice of π, Λ is a penalty term we introduce in the sampling algorithm to control the number of mutations accumulated along the path. Very high values of this parameter minimizes the total number of mutations along the path, which will be equal to the Hamming distance D between the target sequences. Hence, reverse mutations become statistically unlikely, limiting the exploration of the sequence space and resulting in intermediate sequences with generally lower scores according to P_{model}. Hence, we chose $\Lambda \sim 0.1 - 1$ In order to guarantee a deeper exploration of the sequence space, while disfavoring reverse transitions and obtaining smoother paths.

As a final remark on the definition of the path probability, we notice that even though in the following discussion we are going to use RBMs as models of proteins trained on data, Eq. (7.2) is weel defined for any possible probability distribution $P_{\mathrm{model}}(\mathbf{v})$, such as inverse Potts model.

7.1.2 Sampling algorithm and proof of convergence

We now introduce the details of the sampling procedure we use to obtain paths connecting two fixed sequences in a landscape described by the probability distribution $\mathcal{P}^\beta[\mathcal{V}]$ at a certain inverse temperature β. Starting from a path $\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{T-1}\}$, we look at intermediate sequences (starting from $t = 1$) and propose a mutation with the constraint that the Hamming distance between $\mathbf{v}_{t-1}$ and $\mathbf{v}_{t+1}$ is not greater than 1. We accept this move with a probability fixed to ensure detailed balance. Different cases have to be considered, depending on the Hamming distance D_H between the new attempted sequence and existing ones:

- $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 0$. In this case the new sequence $\mathbf{v}'_t$ can have a single mutation at any site, compared with the two adjacent sequences along the path. Hence, we first draw a random site i, then we propose a new sequence $\hat{\mathbf{v}}_t$ which is equal to $\mathbf{v}_{t-1}$ at all sites but i. The new amino-acid for that site $\hat{v}_t^i$ is drawn from the distribution $\propto P^\beta_{\text{model}}(\cdot|\mathbf{v}^{\setminus i}_{t-1})$ (here the amino acids are fixed on all the sites different from i). Then if the old sequence $\mathbf{v}_t$ already had a mutation with respect to $\mathbf{v}_{t-1}$ at given site j, we accept the new mutated sequence $\hat{\mathbf{v}}_t$ (which is equal to $\mathbf{v}_{t-1}$ apart from the amino acid at site i) with a probability

$$p_{\text{acc}}(\mathbf{v}_t \to \hat{\mathbf{v}}_t) = \\ = \min\left(1, \frac{\pi(\mathbf{v}_{t-1}, \hat{\mathbf{v}}_t)^{2\beta} \sum_z P^\beta_{\text{model}}(M_i^z \mathbf{v}_{t-1})}{\pi(\mathbf{v}_{t-1}, \mathbf{v}_t)^{2\beta} \sum_z P^\beta_{\text{model}}(M_j^z \mathbf{v}_{t-1})}\right), \tag{7.3}$$

  where M_i^z indicates the mutation z at site i. If $\mathbf{v}_t$ or $\hat{\mathbf{v}}_t$ are equal to $\mathbf{v}_{t-1}$, then the acceptance probability is $p_{\text{acc}}(\mathbf{v}_t \to \hat{\mathbf{v}}_t) = \min(1, \pi(\mathbf{v}_{t-1}, \hat{\mathbf{v}}_t)^{2\beta}/\pi(\mathbf{v}_{t-1}, \mathbf{v}_t)^{2\beta})$.

- $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 1$. In this case the new sequence $\hat{\mathbf{v}}_t$ can have a single mutation only at the site i where $\mathbf{v}_{t-1}$ and $\mathbf{v}_{t+1}$ are different. At that site, we propose a new mutation from the distribution $\propto P^\beta_{\text{model}}(\cdot|\mathbf{v}^{\setminus i}_{t-1})$ and accept it with probability $p_{\text{acc}} = \exp[-\Lambda\beta(D_H(\hat{\mathbf{v}}_t, \mathbf{v}_{t-1}) + D_H(\hat{\mathbf{v}}_t, \mathbf{v}_{t+1}) - D_H(\mathbf{v}_t, \mathbf{v}_{t-1}) - D_H(\mathbf{v}_t, \mathbf{v}_{t+1}))]$.

- $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 2$. In this case the previous and subsequent sequence present two mutations at site i and j. The new sequence $\hat{\mathbf{v}}_t$ can be of two forms: it can have the same mutation of $\mathbf{v}_{t+1}$ (with respect to $\mathbf{v}_{t-1}$) at site i or at site j. Hence, we extract one of the two possibilities with a probability weighted accordingly with $P^\beta_{\text{model}}(\hat{\mathbf{v}}_t)$.

In practice, to improve the quality of the sampled mutational paths we resort to simulated annealing. We then sample paths from $\mathcal{P}[\mathcal{V}]^\beta$, where β is initially very small and is progressively ramped up to some target value.

To prove that the MCMC algorithm described above converges to the correct distribution $\mathcal{P}[\mathcal{V}]^\beta$, we have to show that the detailed balance condition is satisfied, *i.e.* that each Markov step is reversible. We consider the transition from a path $\mathcal{V} = \{\mathbf{v}_t\}$ to a new path that differ only by one sequence $\mathbf{v}'_t$ at time t. we write the detailed balance condition as

$$\mathcal{P}^\beta(\{\mathbf{v}_t\}) p_{\text{trans}}(\mathbf{v}_t \to \mathbf{v}'_t) = \mathcal{P}^\beta(\{\mathbf{v}'_t\}) p_{\text{trans}}(\mathbf{v}'_t \to \mathbf{v}_t), \tag{7.4}$$

which corresponds to

$$\pi^\beta(\mathbf{v}_{t-1}, \mathbf{v}_t)\pi^\beta(\mathbf{v}_t, \mathbf{v}_{t+1}) P^\beta_{\text{model}}(\mathbf{v}_t) p_{\text{trans}}(\mathbf{v}_t \to \mathbf{v}'_t) = \\ = \pi^\beta(\mathbf{v}_{t-1}, \mathbf{v}'_t)\pi^\beta(\mathbf{v}'_t, \mathbf{v}_{t+1}) P^\beta_{\text{model}}(\mathbf{v}'_t) p_{\text{trans}}(\mathbf{v}'_t \to \mathbf{v}_t). \tag{7.5}$$

If $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 0$, the new sequence can have a mutation at any site i compared to its neighbour $\mathbf{v}_{t-1}$, while $\mathbf{v}_t$ will have the mutation at another site j (note that i and j can be equal). Hence the transition probability in this case will be

$$p_{\text{trans}}(\mathbf{v}_t \to \mathbf{v}'_t) = \frac{1}{N} \frac{P^{\beta}_{\text{model}}(\mathbf{v}'_t)}{\sum_{z=1}^{Q} P^{\beta}_{\text{model}}(M_i^z \mathbf{v}_{t-1})} p_{\text{acc}}(\mathbf{v}_t \to \mathbf{v}'_t), \tag{7.6}$$

$$p_{\text{trans}}(\mathbf{v}'_t \to \mathbf{v}_t) = \frac{1}{N} \frac{P^{\beta}_{\text{model}}(\mathbf{v}_t)}{\sum_{z=1}^{Q} P^{\beta}_{\text{model}}(M_j^z \mathbf{v}_{t-1})} p_{\text{acc}}(\mathbf{v}'_t \to \mathbf{v}_t). \tag{7.7}$$

By substituting everything in the detailed balance condition we obtain that condition (7.4) is respected. This will hold similarly when $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 1$ (with $i = j$). For $D_H(\mathbf{v}_{t-1}, \mathbf{v}_{t+1}) = 2$, the sequences $\mathbf{v}_t$ and $\mathbf{v}'_t$ can either be equal to $\mathbf{v}_{t+1}$ or $\mathbf{v}_{t-1}$, from which the detailed balance condition descends.

7.1.2.1 Sampling direct paths

As a final remark of the sampling procedure, we notice that in the case we are only interested in sampling paths in the direct space, meaning the space of the 2^D mutants containing the amino acids appearing in the two edge sequences (where D is the Hamming distance between the two sequences), we just need to start the sampling procedure form a path with $T = D$ steps. In this way, the sampling algorithm will be forbidden to propose mutations that are not direct, since it will require the path to be longer than D steps. This is a very useful feature, since it allows us to sample direct paths in a very efficient way, without the need to introduce any additional constraint in the sampling procedure.

7.2. Benchmarking path sampling on Lattice Proteins

At this point our goal is to test the performance of the sampling algorithm at finding good paths using RBMs. To this end, we first benchmark our algorithm on a simple model of lattice proteins (LP) [Lau and Dill (1989); Shakhnovich and Gutin (1993)]. The advantage of LP is that we exactly know what is the actual probability distribution from which the training data is sampled from. We will then use the trained RBM to sample paths from the distribution defined in Eq. (7.2) and then measure the quality of the intermediate sequences using the original LP distribution.

Below, we will first introduce the model and explain the training procedure of the RBM. Then, we will show the performance of the sampling algorithm at finding good paths in the LP landscape.

7.2.1 Definition of Lattice Proteins

We consider a protein sequence of 27 amino acids folding into a 3D structure specified as a self-avoiding path over a $3 \times 3 \times 3$ lattice where each amino acid occupies one node. The probability of a sequence $\mathbf{v}$ to fold into a specific structure $\mathbf{S}$ is given by the interaction energies between amino acids in contact in the structure (i.e. those who occupy neighbouring nodes of the lattice, but are not adjacent in the protein sequence). In particular, the total energy of a sequence with respect to a given structure is given by

$$\mathcal{E}_{\text{LP}}(\mathbf{v}|\mathbf{S}) = \sum_{i<j} c_{ij}^{\mathbf{S}} E_{\text{MJ}}(v_i, v_j) \tag{7.8}$$

where $c^{\mathbf{S}}$ is the contact map ($c_{ij}^{\mathbf{S}} = 1$ if sites are in contact and 0 otherwise), while the pairwise energy $E_{\text{MJ}}(v_i, v_j)$ represents the amino-acid physico-chemical interactions given

by the the Miyazawa-Jernigan knowledge-based potential [Miyazawa and Jernigan (1996)]. The probability to fold into a specific structure is written as

$$p_{\rm nat}(\mathbf{S}|\mathbf{v}) = \frac{e^{-\mathcal{E}_{\rm LP}(\mathbf{v}|\mathbf{S})}}{\sum_{\mathbf{S}'} e^{-\mathcal{E}_{\rm LP}(\mathbf{v}|\mathbf{S}')}}\,, \tag{7.9}$$

where the sum is over the entire set of self-avoiding path in the cubic lattice.

The function $p_{\rm nat}$ represents a suitable landscape that maps each sequence to a score measuring the quality of its folding. To study this landscape in more detail, we will sample $\sim 10^4$ sequences associated with a given structure $\mathbf{S}$ (as shown in Figure 7.2) by calculating $-\beta \log p_{\rm nat}(\cdot|\mathbf{S})$ (with $\beta = 10^3$) as the effective energy [Jacquin et al. (2016)].A first analysis of the structure of this data can be obtained by computing its Principal Components (PC). In practice, we use one-hot encoding to project the data in a real-valued space and compute its correlation matrix. The eigenvectors associated with the strongest eigenvalues are the PCs that explain most of the variability of the data. In Figure 7.2(d,e) we plot the logo corresponding to the top two Principal Components (PC1 and PC2). The height of each letter corresponds to the value of the eigenvector entry associated to that amino acid. We note that PC1 corresponds to an extended electrostatic mode, showing how many important contacts are realized by electrostatic bonds between oppositely charged amino acids (see Figure 7.2(b,c)). PC2 instead identifies possible Cys-Cys bridges, in particular between the contacting residues at sites 6-11-22 and 8-15-10. Projecting the sequences onto these two PCs reveals two sub-families separated along PC1 (Fig. 7.2(a)), associated to opposite chains of alternating charges along the electrostatic mode (Figs. 7.2(b,c)). We will use our path sampling procedure to interpolate between the two sub-families, see start (white star) and end (black star) sequences in Fig. 7.2(a).

To sample transition paths, we now train an RBM over the sequences discussed above. The detials of the learning procedures are given in Appendix C.1 while the sequence logo of all the weights of the trained RBM are given in Appendix C.5.

7.2.2 Statistics of sampled paths

After training, we use $P_{\rm model}$ as an approximate expression of $p_{\rm nat}$ and use it to generate mutational paths at low temperature according to Eq. (7.2). Measuring the folding probability $p_{\rm nat}$ along the paths, we find that the algorithm is able to find excellent paths, with intermediate sequences having even higher folding probabilities than the starting and ending sequences (see inset of Figure 7.2(a)). In particular, we show that the folding probability on longer global paths is on average larger than the one of the direct paths. Repeated runs of the sampling procedure give different paths that cluster into two classes, shown in red and maroon in Fig. 7.2(a). Few paths exploit a transient introduction of Cys-Cys interaction (on sites 6, 11 and 22) to stabilize the structure while flipping the electrostatic residues (maroon cluster); while most majority introduce additional stabilizing electrostatic contacts along the path (red cluster).

The last argument can be quantified by looking at the inputs on the hidden units of the trained RBM along the path. As said in Section 2.3.4, the weights of the RBM can encode meaningful patterns of data. This is true also for the case of LP as shown in Figure 7.3. In particular, we plot the inputs of the relevant hidden unit along the sampled paths and their corresponding weights. The two selected inputs have been chosen between those showing highest variability along the paths. From Figure 7.3, we see that w_{39} recovers the extended electrostatic mode already discussed above while w_{40} encodes the Cys-Cys interaction between sites 5,6,11 and 22 (similar to what shown in PC2). Moreover, we notice how some of the paths (the maroon cluster) exploit this Cys-Cys interaction to maintain the folding.

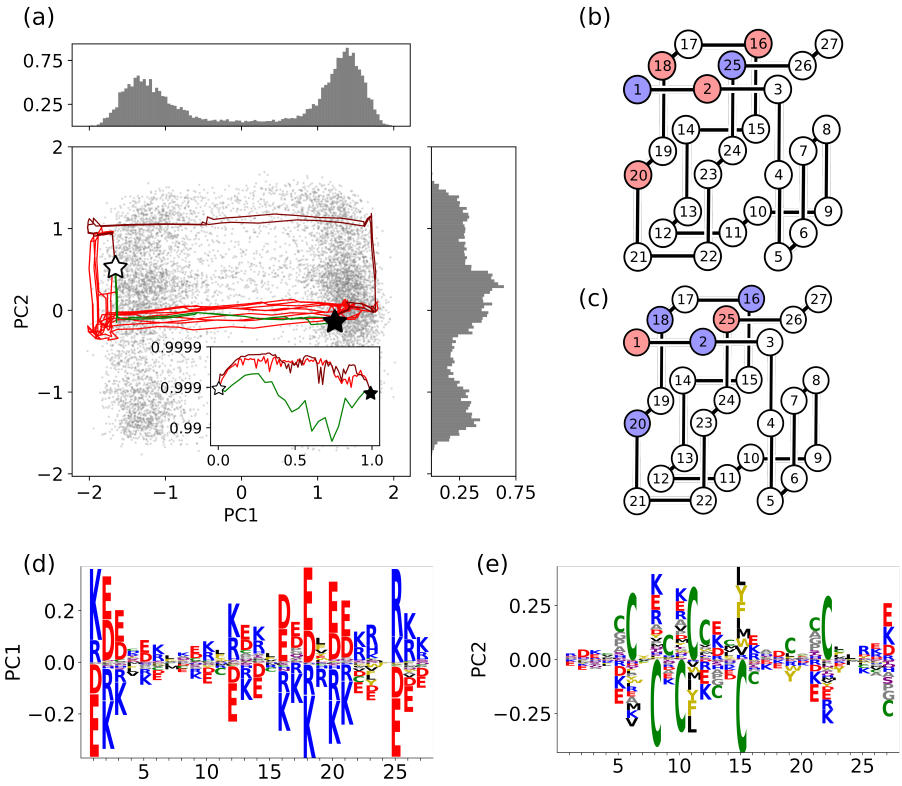


Figure 7.2: Mutational paths for lattice proteins, joining sequences (white star) = DRGIQCLAQMFEKEMRKKRRKCYLECD and (black star) = RECCAVCHQRFKDKIDEDYEDAWLKCN belonging to the family with structure shown in (b) and (c). Red and blue colors respectively correspond to negatively and positively charged amino acids. Cysteine is denoted by a green C. (a) Projections of 10^4 LP sequences in the family (grey dots) along the top two PC of their correlation matrix. Green lines represent direct paths, while red and maroon lines show some global paths sampled from Eq. (7.2). The relative numbers of maroon (2) and red (10) paths respect the statistics over all sampled paths. Parameter values: $\beta = 3$, $\Lambda = 2$, $T_{\text{global}} = 82$, $T_{\text{direct}} = 24$. Sides: histograms of projections along PC1 (top) and PC2 (right). Inset: folding probabilities p_{nat} along each path.(b,c) The fold of the LP family is stabilized by alternating configurations of charges. (d,e) Sequence logos of PC1 and PC2. Image and caption taken from [Mauri et al. (2023a)].

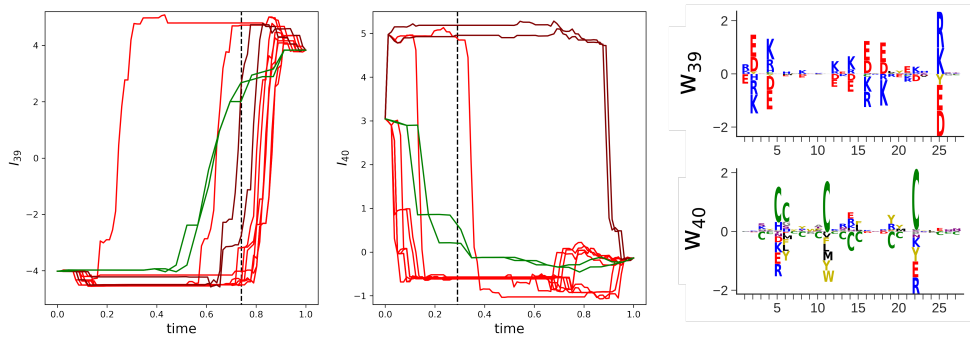


Figure 7.3: Plot of the inputs to two relevant RBM hidden units along the sampled paths already shown in Figure 7.2 (*left*) and sequence logo of their corresponding weigths (*right*). Black lines correspond to the average time at which the input switch value (> 0 for I_{39} and < 2 for I_{40}). Image taken from [Mauri et al. (2023a) (Supplemental Material)].

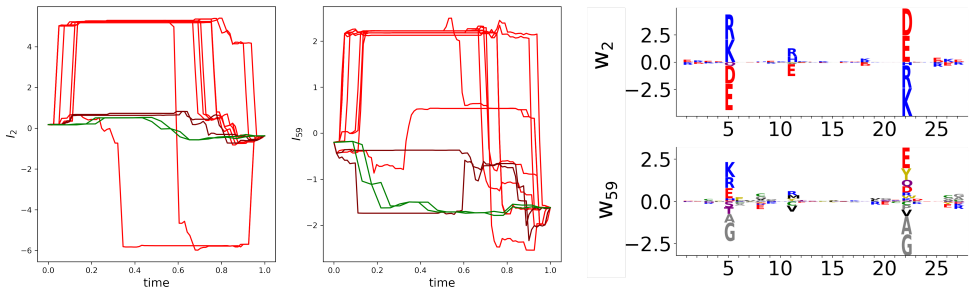


Figure 7.4: Same of Figure 7.2 but for the hidden units $\mu = 2$ and 59. Paths in the red cluster exploit electrostatic interactions not accessible in the direct space to maintain high folding probability. Image taken from [Mauri et al. (2023a) (Supplemental Material)].

Looking at the inputs showing the greater difference between the red and maroon cluster, we recognise other two relevant hidden units (see Figure 7.4). In particular, the two hidden units considered ($\mu = 2$ and 59) encode the electrostatic interactions between the contacting sites 5 and 22. Activating these hidden units, the paths in the red cluster are able to stabilise the structure. On the contrary, direct paths (shown in green in Figure 7.4) can not exploit these interactions, since the required mutations are not present in the two edge sequences. This results in the direct paths having lower folding probability than the global ones (inset Fig. 7.2).

In Chapter 9 we will discuss in more detail the difference between direct and global paths and present a more quantitative argument to explain why and when global paths are preferred to direct ones as the parameters of the transition term π in Eq. (7.2) are varied.

7.3. Mutational paths from data-driven models of natural proteins

The next point we want to show is that the sampling procedure introduced in Section 7.1 can be used in the case of real proteins. In particular we will use an RBM trained on a MSA of WW domains. In this section we are going first to quickly introduce WW domains and its binding properties as well as the performance of the trained RBM at predicting the binding affinity of WW domains. Then, we will sample paths between couples of WW domains belonging to different sub-families and benchmark the quality of the intermediate sequences using numerical methods, such as AlphaFold [Jumper et al. (2021)] and ProteinMPNN [Dauparas et al. (2022)].

7.3.1 Introduction to WW domain and its binding affinities

The WW domain is a family of small protein domains (30-40 amino acids long) whose name comes from the presence of two strongly conserved tryptophan residues in their primary structure [Sudol (1996)]. Most WW domains show binding affinity to proline-rich peptides [Zarrinpar and Lim (2000)], but they may largely differ in the specific amino acid motif to which they bind and in the biological pathway in which these bindings are implicated. Two examples of WW domain containing proteins present in the human proteome are the yes-associated protein 1 (YAP1) and the Peptidyl-prolyl cis-trans isomerase NIMA-interacting 1 (Pin1). While the former acts as a transcritptional co-activator of genes involved in cellular proliferation (with a major role in oncogenic activity [Shibata et al. (2018)]), the latter isomerizes only phospho-Serine/Threonine-Proline motifs and it has many role from cell cycle regulation to the emergence of Alzheimer's disease [Butterfield et al. (2006)].

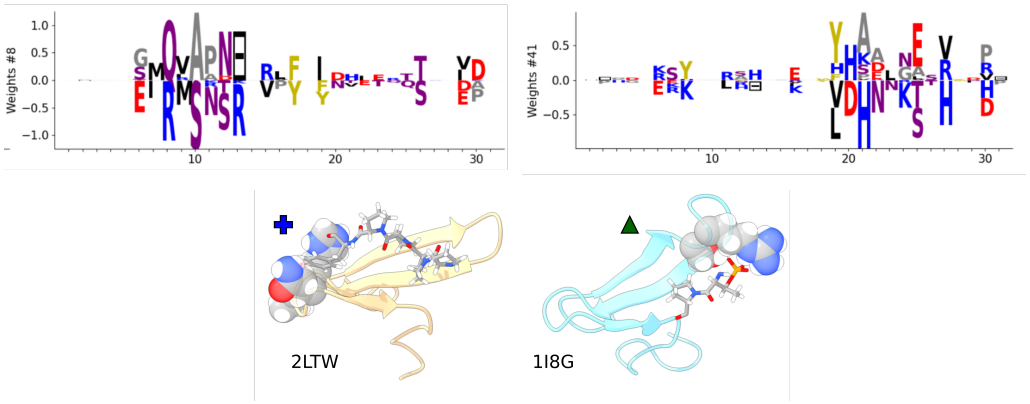


Figure 7.5: (*Top*) Sequence logo of weights corresponding to the two hidden units $\mu = 8$ and $\mu = 41$. The cluster of sites with strong weights corresponds to two different binding pockets in the WW domain. (*Bottom*) Complexes (WW domain and cognate peptides) for classes I (blue cross) and IV (green triangle) with their corresponding PDB IDs Pettersen et al. (2021). Atoms corresponding to the two binding pockets are highlighted. The two WW domains realize the binding in two different position corrsponding to the activated sites in w_8 and w_{41}. The *bottom* subfigure has been taken from [Mauri et al. (2023a)].

Past work showed that WW domains can be divided into four groups (or classes) depending on their preferential binding affinity [Sudol and Hunter (2000); Russ et al. (2005); Otte et al. (2003)], which are defined as follows:

- **Class I**: WW domains that bind to PPxY motifs (where x is any amino acid) with high affinity. The most common example of this class is the YAP1 WW domain whose protein+ligand structure is shown in Figure 7.5 (PDB ID: 2LTW).
- **Class II**: WW domains that bind specifically to PPLP motifs.
- **Class III**: WW domains that bind specifically to peptides containing proline-arginine (PR) motifs.
- **Class IV**: WW domains that bind specifically to phosphorylated serine/threonine-proline (p(S/T)P) motifs. An example of this class is the Pin1 WW domain shown in Figure 7.5 (PDB ID: 1I8G).

7.3.2 Training RBMs on WW domains

We train an RBM on a MSA of WW domains obtained from the PFAM database (PFAM ID PF00397). The data can be fund in Tubiana et al. (2019) and details about the training procedure are given in Appendix C.1. The sequence logo of the weights of the trained RBM are shown in Appendix C.4.

Following the same discussion done by Tubiana et al. (2019), we show that this RBM is able to recognise the binding class of different WW domains. In particular, the binding properties of different WW domains have been experimentally tested by Russ et al. (2005) and Otte et al. (2003). We can take these sequences and project them into the input space defined by the weights of the RBM. Then, we look for the couple of hidden units that best separate the different classes of WW domains. In Figure 7.6(a) we show the projection of the natural sequences in the input space defined by the two hidden units $\mu = 8$ and $\mu = 41$. We see how the experimentally tested sequences (plotted as colored dots) cluster in the

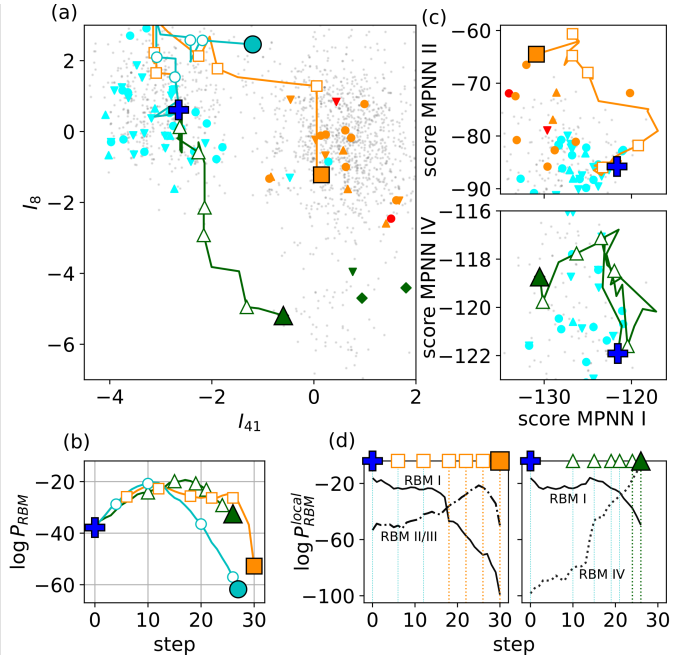


Figure 7.6: Mutational paths of the WW domain using RBM trained on the PFAM PF00397 family, see Appendix C.1 for details about implementation. (a) Natural sequences $\mathbf{v}$ (grey dots) projected onto the plane of inputs I_μ (here $M = 50$) of two hidden units clustering sequences according to the types of ligands they bind Zarrinpar and Lim (2000): I (cyan), II (red), III (orange), IV (green). Marked sequences: Upper (Lower) triangles: natural (artificial), from Russ et al. (2005). Circles: natural, from Otte et al. (2003). Blue cross represents the YAP1 domain. Lines shows the projection of three representative paths connecting YAP1 to sequences in classes I (circle), II (square) and IV (triangle). Intermediate sequences (empty symbols) are listed in Appendix C.2. Parameters: $\beta = 3$, $\Lambda = 0.1$. (c) Log $P_{\rm RBM}$ for sequences along the paths. (c) ProteinMPNN scores for binding affinity. x-axis measures the affinity to class I reference structure while y-axes show affinity to classes II/III (Top) and IV (Bottom) reference structure respectively. (d) Log-likelihood along the paths from I to II (Left) and from I to IV (Right) according to class-specific RBM trained on sequences in the three quadrants (Solid: I, Dot-dashed: II/III, Dotted: IV). Image and caption taken from [Mauri et al. (2023a)].

different corner of the input space, according to their binding class. The only exeption is between class II and III that are grouped together in Figure 7.6(a). This aspect is in agreement with past literature where the distinction between the two classes is acutally unclear, because WW domains often have high affinity with both ligands.

We plot the sequence logo of the weights corresponding to the hidden unit $\mu = 8$ and $\mu = 41$ in Figure 7.5(*Top*). In particular, we notice that the two weights define two different groups of sites along the sequence which corresponds to two different binding pockets in the WW domain, as shown by the complexes in Figure 7.5(*Bottom*) [Russ et al. (2005); Jäger et al. (2006); Kato et al. (2002)].

From the classification given by Figure 7.6(a), we can also build local models for each cluster. We partition the MSA according to the positions (x=I_{41}, y=I_8) of the representative points of sequences in the 2D plane of Figure 7.6(a). More precisely, we define three clusters in the following way: $x < -1$; $x > -1$ and $y > -3$; $x > -1$ and $y < -3$. The corresponding sub MSAs are used to train specific RBM models. This results in three RBMs (one for class I, one for class II/III and one for class IV) that we can use as a predictor of the binding affinity of each protein in the family. More details about the training procedure of these RBMs are given in Appendix C.1.

	YAP1	wt class I
YAP1	1	0.88
1	0.95	0.83
2	0.94	0.86
3	0.91	0.82
4	0.84	0.98
wt class I	0.88	1

	YAP1	wt class IV
YAP1	1	0.76
1	0.93	0.76
2	0.9	0.8
3	0.89	0.84
4	0.81	0.7
5	0.83	0.93
wt class IV	0.76	1

	YAP1	wt class II
YAP1	1	0.76
1	0.92	0.72
2	0.93	0.75
3	0.86	0.85
4	0.84	0.84
5	0.86	0.83
wt class II	0.76	1

Figure 7.7: Structure similarity scores (measured in terms of TM-scores) between some designed WW domains along the paths (rows of the table) and the natural wild-types at the edges (columns of the table). The structure of the designed proteins has been obtained using AlphaFold. Image taken from Mauri et al. (2023a) (Supplemental Material).

7.3.3 Characterising mutational paths in WW domains

We now use the RBM trained on WW domains to sample mutational paths interpolating between couples of different wild-types belonging to different binding classes. In particular, starting from the YAP1 WW domain (class I) we sample three paths connecting it to a WW domain belonging to class I, II and IV, respectively. The projection in the input space corresponding to the hidden units $\mu = 8$ and $\mu = 41$ are shown in Figure 7.6(a).

The sampled paths are predicting to be of good quality by the RBM, as shown by the log-likelihood of the intermediate sequences in Figure 7.6(b) that are higher than those of the wild-types. Interestingly, the path connecting YAP1 to a class IV WW domain (green triangle) crosses a region in the input space of Figure 7.6(a) which is lacking of natural sequences. This might be a sign of the path exploring a promiscuous region of the sequence space, where the binding pockets characterized by w_8 and w_{41} are both optimised in the intermediate sequences. Promiscuous sequences are penalized during the process of evolution, which selects proteins to work for more specific tasks. This might suggest that the sequences of class I and IV evolved from a region of promiscuous activity and specialized during evolution, leaving the ancestral region empty. This evolutionary path may eventually be approximated by the sampling procedure presented in this chapter. In Appendix C.3, we show that this prediction remains valid if we sample multiple paths with different initial and final configurations.

The last hypothesis require further investigation, in particular from the experimental point of view. However, we can use numerical methods to benchmark the quality of the intermediate sequences. Below, we will first benchmark our results using AlphaFold [Jumper et al. (2021)] and then with ProteinMPNN [Dauparas et al. (2022)] in order to predict the binding affinity of the intermediate sequences.

To further assess the specificity of sequences on the sampled path, we have also measured the log-likelihood of the intermediate sequences along the path using class-specific RBMs previously discussed in Section 7.3.2. The results are shown in Figure 7.6(d). The cross-overs between the log-likelihoods of the class-specific RBMs in Fig. 7.6(d) suggest the presence of specificity switches along the I $\to$ II/III and I $\to$ IV paths.

7.3.3.1 Benchmarking with AlphaFold

We have already introduced this tool in Section 3.1. Here, we will use it to predict the structure of the intermediate sequences along the paths and compare it with the structure of the wild-types. In particular, we will use experimental validated structures of WW domains in complex with their cognate ligands as reference. Their PDB IDs are 2LTW for class I, 1YWI for class II and 1I8G for class IV. Due to the computational cost of AlphaFold, we will only predict the structure of a subset of the intermediate sequences along each path. The sequences are given in Appendix C.2 and are represented by empty

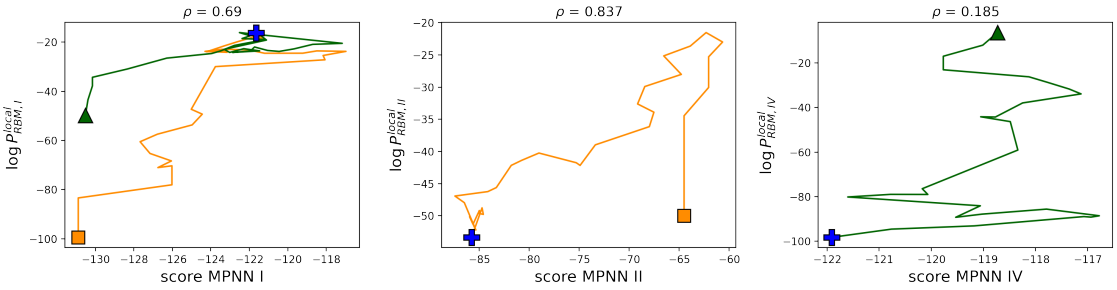


Figure 7.8: Comparison of the MPNN score and local RBM likelihoods along the paths shown in Figure 7.6. The title of each panel shows the Spearman correlation between RBM and ProteinMPNN scores. Image and caption taken from Mauri et al. (2023a) (Supplemental Material).

symbols in Figure 7.6.

To compare the inferred structures of the sampled sequences along the path with those of the target natural sequences we used the Template Modelling (TM) score developed by Zhang and Skolnick (2004) and represents a variation of the Levitt–Gerstein (LG) score [Levitt and Gerstein (1998)]. Compared to other similarity score (like root-mean-square deviation (RMSD)) it gives a more accurate measure since it relies more on the global similarity of the full sequence rather than the local similarities. Practically, we consider a target sequence of length L_{target} and a template one whose structure has to be compared with. First, we align the two sequences and we take the L_{common} pairs of residues that commonly appear aligned. Then the score is computed as

$$\text{TM-score} = \max_{\{d_i\}} \left[\frac{1}{L_{\text{target}}} \sum_{i=1}^{L_{\text{common}}} \frac{1}{1 + \left(\frac{d_i}{d_0(L_{\text{target}})} \right)^2} \right], \tag{7.10}$$

where d_i is the distance between the ith pair of residues between the template and the target structures after alignment, and $d_0(L_{\text{target}}) = 1.24(L_{\text{target}} - 15)^{1/3} - 1.8$ is a distance scale that normalizes distances. This formula gives a score between 0 and 1. if TM-score< 0.2, the two sequences are totally uncorrelated, while they can be considered to have the same structure if TM-score> 0.5.

In Figure 7.7 we show the tables of the TM-scores associated with the sequences tested above. In particular, We obtain TM-score > 0.5, indicating a high similarity between the folds of sequences sampled along the path and of natural WW.

7.3.3.2 Benchmarking with ProteinMPNN

We now aim to test our sequences using ProteinMPNN [Dauparas et al. (2022)]. As presented in Section 3.2, this tool creates a probabilistic model as a function of the sequence given a target backbone structure, P_{MPNN}. This score can be used to measure if the sequence is compatible with the target structure. For the case of the complex between a WW domain and its cognate ligand, we can interpret the MPNN score as an estimate of the binding affinity between the two proteins.

In order to compute the likelihood of a sequence given a reference backbone, the sequence has to have the same length of the wild-type from which the reference backbone was taken. Unfortunately, the transition from class I to class IV requires an insertion along the path (see Appendix C.2). So, we decided to remove those insertion to test the sequences against the class I reference backbone structure (PDB ID: 2LTW) and to substitute the gap with Serine (S) to test them against class IV reference structure (PDB ID: 1I8G).

Results are shown in Figure 7.6(c). In general, intermediates sequences show binding scores comparable (or higher) to those of the other tested natural sequences. As expected, along the path I $\rightarrow$ II/III the affinity to class I-cognate decreases while those to class II/III-cognate increases. More interestingly, the path I $\rightarrow$ IV shows a region where the affinities with respect to both complexes are high[1]. It has been experimentally shown that some natural WW domains belonging to class I have also class IV activity [Russ et al. (2005)]. This results is in agreement with the hypothesis that the path I $\rightarrow$ IV explores an ancestral,promiscuous region of the sequence space described in Section 7.3.3.

To further test the agreement between RBMs and ProteinMPNN at predicting binding affinities, we compared the log-likelihoods along the paths with respect to the local RBMs and ProteinMPNN. Results are shown in Figure 7.8. We see that the two methods are slightly in agreement, as we can see in the spearmann correlations showed in the title of each panel of Figure 7.8. The discrepancies between the two models can be understood in the following way. While RBMs express information more related to evolution (since they are trained on MSAs of homologous sequences), ProteinMPNN mostly focuses on structural information [Malbranke et al. (2023)]. This means that for some sequences containing rare mutations not present in the training dataset, their RBM score will be penalized despite the fact that they are structurally compatible with the target complex.

[1] In the sense that the scores are comparable with those of the natural wildtypes at the edges of the path.

8

Mean-field theory of paths with RBMs

In the previous chapter, we proposed a Monte Carlo algorithm to sample mutational paths that interpolate between two homologous sequences while maximizing the expected fitness, measured as a probability score, along the path. We benchmarked this method using exactly solvable models of Lattice Proteins and a model of WW domains. The quality of the sampled paths was verified using state-of-the-art numerical tools based on deep learning, such as AlphaFold and ProteinMPNN.

From a computational perspective, our algorithm is highly efficient. Using RBMs as reference models, it takes less than one minute to sample paths in WW domains (indicatively, we sampled paths with a number of steps comparable to the length of the protein, $N = 31$). The computational time is expected to scale linearly with the length of the protein and the total length of the path. However, when it comes to analyzing the statistical properties of the paths in detail, the efficiency of the algorithm diminishes. This is due to the need to sample a large number of paths, which requires parallel computation and can be computationally expensive.

To overcome this challenge, we can apply a standard approximation scheme in statistical physics known as mean-field theory. In mean-field theory, the entire ensemble of statistically relevant configurations can be described using a single set of statistics called order parameters. These parameters provide access to all the necessary information of the system without the need for sampling. Specifically, we can compute the free energy of the system, equivalent to the logarithm of the partition function, as a function of the order parameters. From this, we can compute other thermodynamic quantities, such as entropy and specific heat.

Mean-field theories are valid for large systems where the probability of each configuration can be expressed as a function of the order parameters. In the case of paths sampled with RBMs, the inputs to the hidden space of each intermediate configuration can be viewed as effective candidates for order parameters. This suggests the application of mean-field theory to the ensemble of paths described in the previous chapter. The goal of this chapter is to demonstrate how this can be accomplished and to discuss the potential applications of this approach in studying relevant properties of the paths. Specifically, two primary applications of this approach are: estimating the total number of possible paths connecting two sequences, corresponding to computing the ensemble entropy, and computing statistics relevant in evolutionary dynamics, such as transition probabilities between configurations and escape times from metastable states.

The chapter is organized as follows. First, we will describe how to apply mean-field theory to the ensemble of paths, discussing the main assumptions for its validity. Next, we will present the formulation of the free energy of the system and demonstrate how it can be used to compute relevant statistics. Finally, we will present results obtained using

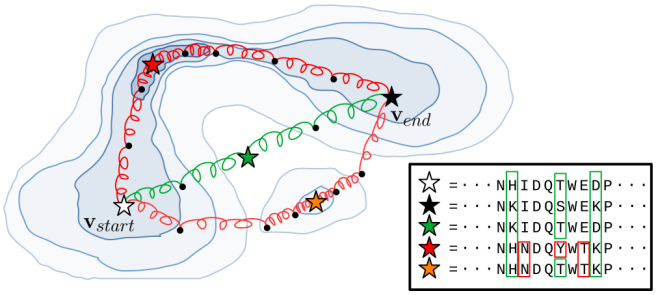


Figure 8.1: Mutational paths between two subfamilies in the sequence landscape associated to a protein family. Darker blue levels correspond to increasing values of the protein fitness. Paths are either direct (green: each site carries the amino acid present at the same position in the initial or in the final sequence) or global (red: no restriction on amino acids), making possible the exploration of high-fitness regions. Image and caption from Mauri et al. (2023b).

this approach for RBMs trained on LP and WW domains, as introduced in the previous chapter. In particular, we will also discuss the case of direct and global paths, as defined in Chapter 7. We will show how global paths can explore a larger portion of the sequence space, allowing for longer paths (in terms of total mutations applied) with higher average fitness (*i.e.*, lower energy), see Figure 8.1.

The results presented in this chapter are based on our original work, which has been published in the following papers: *Mutational Paths with Sequence-Based Models of Proteins: From Sampling to Mean-Field Characterization* and *Transition paths in Potts-like energy landscapes: general properties and application to protein sequence models* [Mauri et al. (2023a,b)].

8.1. Mean-field theory of transition paths

We recall what we have written in Section 7.1.1. In particular, we consider an energy landscape $E_{\text{model}}(\mathbf{v})$ over N-dimensional Potts configurations $\mathbf{v}$, see Fig. 8.1. E_{model} can correspond to the energy of a trained RBM, but also derived from first principles, as we will show in next chapters. Following Mauri et al. (2023a), we associate to each path $\mathcal{V} = \{\mathbf{v}_{\text{start}}, \mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_{T-1}, \mathbf{v}_{\text{end}}\}$ an energy $\mathcal{E}(\mathcal{V})$. This energy is the sum of the energies of the intermediate configurations along the path, and of elastic contributions decreasing with the similarities between pairs of successive configurations. We denote by Φ the elastic potential. Notice that compared to the previous chapter, we are rewriting the interaction term π as $\pi(\mathbf{v}, \mathbf{v}') = \exp(-N\Phi(q(\mathbf{v}, \mathbf{v}')))$, where the overlap $q(\mathbf{v}, \mathbf{v}') = \frac{1}{N}\sum_i \delta_{v_{i,t}, v_{i,t+1}}$ measures the similarity between adjacent sequences. We will expalin this choice below. The energy of a path (divided by N) is then

$$\mathcal{E}(\mathcal{V}) = \frac{1}{N}\sum_{t=1}^{T-1} E_{\text{model}}(\mathbf{v}_t) + \Phi(q(\mathbf{v}_{\text{start}}, \mathbf{v}_1)) + \\ + \sum_{t=1}^{T-2} \Phi(q(\mathbf{v}_t, \mathbf{v}_{t+1})) + \Phi(q(\mathbf{v}_{T-1}, \mathbf{v}_{\text{end}})) . \quad (8.1)$$

The probability of the path is then defined as the Boltzmann distribution

$$\mathcal{P}[\mathcal{V}] = \frac{1}{Z_{\text{path}}} e^{-\beta N \mathcal{E}(\mathcal{V})} , \quad (8.2)$$

where β is an inverse temperature and Z_{path} ensures normalization. This distribution promotes paths, where intermediate configurations have low energies E_{model}, and are not far away from each other in order to guarantee smoothness in the interpolation.

The choice of rewriting the interaction term π, *i.e.* as the exponential of an energy scaling with N, comes from the following argument. In our mean-field theory we will characterise paths by two sets of order parameters: the projection of each sequence into the input space of the RBM (as described in the introduction of the chapter) and the overlap (related to the number of mutations) between successive sequences, see Figure 8.1. Each order parameter has to be an intensive quantity, meaning that it does not scale with the size of the system, N. The constraint on the number of mutations defined in Chapter 7 is not consistent with this assumption (in particular, the overlap will scale as $1/N$). Hence, for the mean-field theory to make sense, we need to relax this condition and allow for an extensive number of mutations. Moreover, since the energy of a configuration as to be an extensive quantity (*i.e.* scaling with the size of the system), we are required to add that scaling in front of the potential Φ. We will see that the results obtained in this framework are consistent with what shown in the previous chapter.

8.1.1 Choice of the elastic potential

Clearly, the potential Φ plays a key role in this model by controlling the elastic properties of the path. In Mauri et al. (2023a), we considered two choices for Φ, corresponding to distinct scenarios for the mutational dynamics. The first one, denoted by Cont, makes sure that any two contiguous configurations along the path, $\mathbf{v}_t$ and $\mathbf{v}_{t+1}$, differ by a bounded (and small compared to N) number of sites. The second choice for Φ is inspired by Kimura's theory of neutral evolution [Kimura (1983)] and hereafter called Evo. It enforces a constant mutation rate for each variable, see below.

a) Cont scenario

The first choice for Φ is inspired by the constraint on the number of mutations defined in Chapter 7. In particular, we want the number of mutations between successive configurations to be small and bounded by a certain threshold. Moreover, our goal is to define a continuous limit as the number of steps T grows by lowering this threshold accordingly. In practice, we define the Φ as an hard-wall potential of the form

$$\Phi_{\text{Cont}}(q) = \frac{1}{T^2|q-q_c|} = \frac{1}{T^2\left|q-1+\frac{\gamma}{T}\right|}, \tag{8.3}$$

where the $1/T^2$ scaling in the potential guarantees the existence of continuous solution in the large-T limit as shown in the next chapter. Other choices of potentials with hard-wall constraints give similar results. The parameter γ controls the elasticity of the path. Its minimal value is D/N, where D is the Hamming distance between the extremities $\mathbf{v}_{\text{start}}$ and $\mathbf{v}_{\text{end}}$. Larger values of γ will authorize more flexible paths.

b) Evo scenario

In the Evo scenario, the potential Φ is chosen to emulate neutral evolution [Kimura (1983)] with a certain mutation rate μ (measured in [mutations/ genome length/ cellular division]), and is given by

$$\Phi_{\text{Evo}}(q) = (1-q)\ln\left(1+\frac{A}{e^{\mu A/(A-1)}-1}\right). \tag{8.4}$$

In this setting, paths can be seen as alternating steps of random mutations (starting from $\mathbf{v}_{\text{start}}$) and of selection, parametrized by, respectively, the mutation rate μ and the

effective 'log. fitness' $-E_{\text{model}}$. The transition path is conditioned to end in $\mathbf{v}_{\text{end}}$. In standard evolutionary dynamics, paths are not constrained by their final configuration, but only by their initial one. Such paths are anchored at one extremity only. However, if the configuration (genome) of an organism is observed after some evolutionary time, it is legitimate to ask about the distribution of putative paths followed by the organism that interpolate between this 'final' and the known initial configurations. As a result of this conditioning, the transitions paths are now anchored at both extremities. In Appendix D.1, we show in detail how this potential is derived from Kimura's neutral theory of evolution. This choice of potential is also consistent with the more complex Eigen's evolution model which takes into account also the effects of selection [Leuthäusser (1987)].

8.1.2 Computing the free energy of the model

We now turn into the problem of computing the free energy of the transition-path model in the context of mean-field theory presented in Mauri et al. (2023a,b). As said before, we aim to describe the ensemble of paths using as order parameters the overlaps between adjacent sequences along the path and the input over the hidden space of each configuration. In particular, for a given sequence along the path, $\mathbf{v}_t$, we will consider the intensive parameter $m^\mu(\mathbf{v}_t) = \frac{1}{N}\sum_i w_{i\mu}(v_{t,i})$, for $\mu = 1, \dots, M$, where $w_{i\mu}$ is the weight matrix as defined in Chapter 2 and $v_{t,i}$ is the i-th component of $\mathbf{v}_t$.

By recalling the definition of the RBM given in Eq. (2.41), we can write the partition function of the path ensemble, $Z_{\text{path}} = \sum_{\mathcal{V}} \exp\left(-\beta N \mathcal{E}(\mathcal{V})\right)$ as

$$Z_{\text{path}} = \sum_{\mathcal{V}} \exp\left(\beta \sum_{t,i} g_i(v_{i,t}) + \beta \sum_{\mu,t} \Gamma_\mu(N m_\mu(\mathbf{v}_t)) - \beta N \sum_t \Phi(q(\mathbf{v}_t, \mathbf{v}_{t+1}))\right). \tag{8.5}$$

Upon a change of variables, we can rewrite the partition function as an integral over the order parameters $\mathbf{m} = \{m_t^\mu\}$ and $\mathbf{q} = \{q_t\}$ as

$$Z_{\text{path}} = \int \mathrm{d}\mathbf{m}\, \mathrm{d}\mathbf{q} \exp\left(\beta N \sum_{\mu,t} \frac{1}{N} \Gamma_\mu(N m_t^\mu) - \beta N \sum_t \Phi(q_t) + N \mathcal{S}(\mathbf{m}, \mathbf{q})\right), \tag{8.6}$$

where we have defined $\mathcal{S}$ as

$$\mathcal{S}(\mathbf{m}, \mathbf{q}) = \frac{1}{N} \log \sum_{\mathcal{V}} e^{\beta \sum_{i,t} g_i(v_{i,t})} \prod_{\mu,t} \delta\left(\frac{1}{N}\sum_i w_{i\mu}(v_{i,t}) - m_t^\mu\right) \times \delta\left(\frac{1}{N}\sum_i \delta_{v_{i,t}, v_{i,t+1}} - q_t\right), \tag{8.7}$$

where δ is the Dirac delta function. This object corresponds to the number of configurations compatible with the order parameters $\mathbf{m}$ and $\mathbf{q}$, biased by the local fields g_i and can be seen as an entropic contribution to the free energy.

To continue with the computation of the free energy in mean-field theory, we now have to formally send $N \to \infty$, while the number of stored patterns M is kept finite. Our goal is to write the argument in the exponential of Eq. (8.6) as something that scales as $\mathcal{O}(N)$ and then use the saddle-point approximation to compute the integral. Note that in order for Γ_μ to be well defined in this limit, we are formally supposing that the local fields on the hidden space are scaling as N, *i.e.* $\gamma_{\mu,\pm},\ \theta_{\mu,\pm} = \mathcal{O}(N)$. On the contrary, the weights and local biases do not scale with N, *i.e.* $w_{i\mu},\ g_i = \mathcal{O}(1)$.

Once this assumptions are satisfied, we can use integral representation of the Dirac delta functions in Eq. (8.7), given by

$$\delta(\mathbf{x}) = \frac{1}{2\pi/N}\int \mathrm{d}\hat{\mathbf{x}}\, e^{N\hat{\mathbf{x}}\cdot\mathbf{x}}, \tag{8.8}$$

to express the term $\mathcal{S}$ as an integral over the auxiliary variables $\hat{\mathbf{m}} = \{\hat{m}_t^\mu\}$ and $\hat{\mathbf{q}} = \{\hat{q}_t\}$:

$$\mathcal{S}(\mathbf{m},\mathbf{q}) = \frac{1}{N}\log\int \frac{\mathrm{d}\hat{\mathbf{m}}\,\mathrm{d}\hat{\mathbf{q}}}{(2\pi/N)^2}\exp\left(-N\hat{\mathbf{m}}\cdot\mathbf{m} - N\hat{\mathbf{q}}\cdot\mathbf{q}\right) \times \sum_{\mathcal{V}}\prod_i \exp\left[\beta\sum_t g_i(v_{i,t}) + \sum_{\mu,t}\hat{m}_t^\mu w_{i\mu}(v_{i,t}) + \sum_t \hat{q}_t\delta_{v_{i,t},v_{i,t+1}}\right]. \tag{8.9}$$

In the large-N limit, the integral in Eq. (8.9) is dominated by the saddle-point. Hence, we can obtain

$$\mathcal{S}(\mathbf{m},\mathbf{q}) = \min_{\hat{\mathbf{m}},\hat{\mathbf{q}}}\left[-\hat{\mathbf{m}}\cdot\mathbf{m} - \hat{\mathbf{q}}\cdot\mathbf{q} + \frac{1}{N}\sum_i \log Z_i^{\mathrm{1D}}(\hat{\mathbf{m}},\hat{\mathbf{q}})\right], \tag{8.10}$$

where

$$Z_i^{\mathrm{1D}}(\hat{\mathbf{m}},\hat{\mathbf{q}}) = \sum_{\{v_t\}}\exp\left[\beta\sum_t g_i(v_t) + \sum_{\mu,t}\hat{m}_t^\mu w_{i\mu}(v_t) + \sum_t \hat{q}_t\delta_{v_t,v_{t+1}}\right] \tag{8.11}$$

is the partition function of a 1D-Potts models with nearest-neighbour interactions. Z_i^{1D} can be efficiently estimated through products of $A\times A$-dimensional transfer matrices, where A is the number of Potts states. For global paths, $A = 21$, while $A = 2$ for direct paths. We note that in the case of paths with both ends fixed the starting and final element of this sum are fixed. We can generalize this assumption for the case of paths with one free end by summing over the last element v_T of the path. We will see below how this can be useful.

At the saddle-point of Eq. (8.10), the auxiliary variables fulfill the following set of coupled implicit equations

$$\begin{aligned} m_t^\mu &= \frac{1}{N}\sum_i \frac{\partial \log Z_i^{\mathrm{1D}}}{\partial \hat{m}_t^\mu}(\hat{\mathbf{m}},\hat{\mathbf{q}})\ , \\ q_t &= \frac{1}{N}\sum_i \frac{\partial \log Z_i^{\mathrm{1D}}}{\partial \hat{q}_t}(\hat{\mathbf{m}},\hat{\mathbf{q}})\ . \end{aligned} \tag{8.12}$$

We conclude, according to Eq. (8.6), that the path free-energy is given by

$$\begin{aligned} f_{\mathrm{path}}(\beta) &= \lim_{N\to\infty} -\frac{1}{N\beta}\log Z_{\mathrm{path}}(\beta) \\ &= \min_{\mathbf{m},\mathbf{q}} f_{\mathrm{path}}(\beta,\mathbf{m},\mathbf{q}), \end{aligned} \tag{8.13}$$

where we have defined the free-energy functional

$$f_{\mathrm{path}}(\beta,\mathbf{m},\mathbf{q}) = -\frac{1}{N}\sum_{\mu,t}\Gamma_\mu(Nm_t^\mu) + \sum_t \Phi(q_t) - \frac{1}{\beta}\mathcal{S}(\mathbf{m},\mathbf{q})\ , \tag{8.14}$$

The minimum of f_{path} is reached for the roots of

$$\hat{m}_t^\mu = \beta\,\Gamma_\mu'(Nm_t^\mu)\ ,\ \hat{q}_t = -\beta\,\Phi'(q_t), \tag{8.15}$$

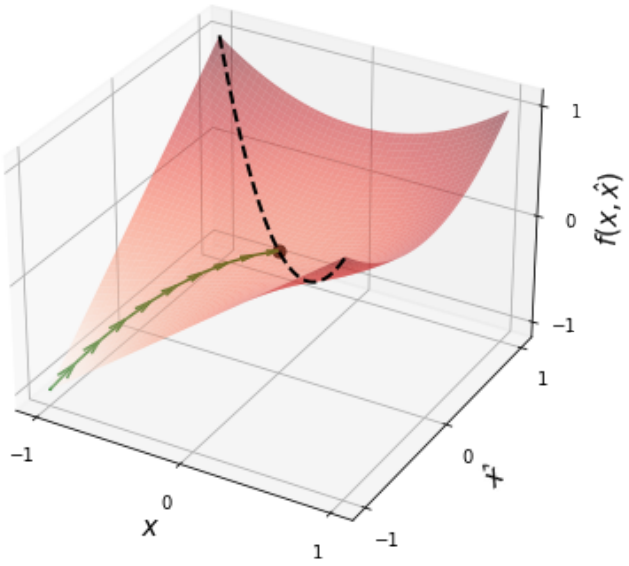


Figure 8.2: Sketch of the free energy minimisation scheme. We plot the free energy f as a function of both the order parameter x and its auxiliary variable $\hat{x}$ (similar to the sets $(\mathbf{m}, \mathbf{q})$ and $(\hat{\mathbf{m}}, \hat{\mathbf{q}})$ from the main text, respectively). Once the auxiliary variable is fixed (by solving Eq. (8.12)) for any values of x, minimising the free energy (*i.e.* solving Eq. (8.15)) corresponds to minimise f along a submanifold of the $(x, \hat{x})$ space (black dasehd line). In our scheme, we are not exploring f along this submanifold, but we approach the same saddle-point from an orthogonal direction (green arrows), resulting in a non-decreasing free energy. Equivalently stated, we are first solving Eq. (8.15), for any value of x, and then Eq. (8.12).

which, together with Eq. (8.12), form a closed set of self-consistent equations for the order parameters. Note that for the case of simpler Hopfield models, where $\Gamma_\mu(x) = \frac{1}{2N}x^2$, the same procedure can be applied and is valid as long as the number of patterns, M, stays finite compared to the size of the system, N. A model of this type will be studied in the next chapter.

8.1.3 Numerical estimation of the free energy

To solve the model numerically, we need to jointly solve the set of self-consistent equations given by Eq. (8.15) and Eq. (8.12). In principle, one should first solve Eq. (8.12) for any possible value of $\hat{\mathbf{m}}$ and $\hat{\mathbf{q}}$ in order to find the values of the auxiliary variables minimising the entropic term $\mathcal{S}$. Then, we use them to compute the functional f_{path} only as a function of the order parameters and minimize it by inverting Eq. (8.15), See Fig. 8.2. However, this procedure is computationally expensive, since it requires to solve Eq. (8.12) for each value of $\hat{\mathbf{m}}$ and $\hat{\mathbf{q}}$.

A different approach is the following: we inject Eq. (8.15) into Eq. (8.12) and we solve the resulting self-consistent equation for the order parameters using Gradient Descent. The derivatives of $\log Z_i^{\text{1D}}$ are taken using automatic differentiation technique built in the Python library JAX [Bradbury et al. (2018)].

Despite converging to a correct solution, this procedure is fundamentally different from the first approach described above, since Eq. (8.15) only holds at the correct saddle-point. In particular, as $\mathbf{m}$ and $\mathbf{q}$ are updated, the auxiliary variables $\hat{\mathbf{m}}$ and $\hat{\mathbf{q}}$ taken from Eq. (8.15) are not guaranteed to be the correct ones, *i.e.* those minimising $\mathcal{S}$. This results in the functional f_{path} not being a monotonically decreasing during the minimisation

procedure. However, we found that this approach is more efficient and allows to obtain the correct solution in a reasonable amount of time (few seconds or some minute depending on the length of the path). See Figure 8.2 for a sketch of the difference between the two approaches.

8.1.4 Computing relevant statistics

Here, we review how to compute relevant statistics of the paths using the mean-field theory described above. In the next section of this chapter, we will apply this methods to analyse the properties of paths in Lattice Proteins and WW domains modeled with RBMs.

a) Amino-acid frequencies along the path

Once the mean-field solution has been determined through minimization of f_{path} we can compute any observable, such as the average frequencies of amino acids on site i at intermediate step t on the path:

$$\langle \delta_{v_{i,t},a} \rangle = \frac{\partial f_{\text{path}}}{\partial(\beta g_{i,t}(a))} = \\ = \sum_{\{v_{t'}\}} \frac{\delta_{v_{i,t},a}}{Z_i} \exp\left(\beta \sum_{t'} g_{i,t'}(v_{t'}) \sum_{t',\mu} \hat{m}^{\mu}_{t'} w_{i\mu}(v_{t'}) + \sum_{t'} \hat{q}_{t'} \delta_{v_{t'},v_{t'+1}} \right) , \tag{8.16}$$

where $\hat{m}^{\mu}_t = \beta \Gamma'_{\mu}(N m^{\mu}_t)$ and $\hat{q}_t = -\beta \Phi'(q_t)$, see Eq. (8.15) [Mauri et al. (2023b)].

b) Entropy of paths

Another quantity of interest would be entropy of the system, corresponding to the number of relevant transition paths. Knowing this quantity would be useful for example to estimate how rare the transition between two regions of the sequence space is.

From a practical point of view, despite the care brought in numerically solving Eqs. (8.12) a small disagreement between the left and right hand sides may subsist. As the number of order parameters scales proportionally to T and M, these inaccuracies must be taken into account when estimating the entropy $\mathcal{S}_{\text{path}}$. To compute the latter we therefore estimate f_{path} at different inverse temperatures β and use the identity

$$\mathcal{S}_{\text{path}} = -\frac{\mathrm{d} f_{\text{path}}}{\mathrm{d}(1/\beta)} . \tag{8.17}$$

In the case of RBM we obtain

$$\mathcal{S}_{\text{path}} = -\frac{\beta}{N} \sum_{i,t} \langle g_{i,t}(a) \rangle + \frac{1}{N} \sum_i \log Z_i - \frac{\beta}{N} \left(\Gamma'(N\mathbf{m}) \frac{\partial}{\partial \hat{\mathbf{m}}} - \Phi'(\mathbf{q}) \frac{\partial}{\partial \hat{\mathbf{q}}} \right) \sum_i \log Z_i . \tag{8.18}$$

Notice that at the exact saddle-point, the last equation is equivalent to $S_{\text{path}} = \beta \langle \mathcal{E} \rangle - \beta f_{\text{path}}$, where $\langle \mathcal{E} \rangle$ is the average energy of the paths. Since our numerical solution is not exact, Eq. (8.18) takes care of these inaccuracies allowing for a precise estimate of the entropy [Mauri et al. (2023b)].

c) Transition and escape probabilities

Finally, we can also compute the transition probabilities between configurations and the escape times from metastable states. In particular, we can compute the probability of a transition between two configurations $\mathbf{v}$ and $\mathbf{v}'$ after T steps by comparing the statistical weights (corresponding to the Boltzmann factor $e^{-N\mathcal{E}(\mathcal{V})}$) between all the paths connecting the two configurations and those connecting $\mathbf{v}$ to anything else. This can be written as

$$P(\mathbf{v} \to \mathbf{v}'|T) = \frac{\sum_{\mathcal{V}:\mathbf{v}_{\text{end}}=\mathbf{v}'} e^{-N\mathcal{E}(\mathcal{V})}}{\sum_{\mathcal{V}} e^{-N\mathcal{E}(\mathcal{V})}} \underset{N\gg 1}{\sim} \frac{e^{-N f_{\text{path}}^{\text{const.}}(\mathbf{v},\mathbf{v}'|T)}}{e^{-N f_{\text{path}}^{\text{unconst.}}(\mathbf{v}|T)}}, \tag{8.19}$$

where $f_{\text{path}}^{\text{const.}}$ is the free energy of the path ensemble with the constraint that the path ends in $\mathbf{v}'$ and $f_{\text{path}}^{\text{unconst.}}$ is the free energy of the unconstrained ensemble [Mauri et al. (2023a)].

Similarly, we can compute the probability of a path to end inside a certain region $\mathcal{R}$ of the sequence space after T steps as

$$P_{\text{stay}}(\mathcal{R}|T) = \frac{\sum_{\mathcal{V}:\mathbf{v}_{\text{end}}\in\mathcal{R}} e^{-N\mathcal{E}(\mathcal{V})}}{\sum_{\mathcal{V}} e^{-N\mathcal{E}(\mathcal{V})}} \underset{N\gg 1}{\sim} \frac{e^{-N f_{\text{path}}^{\text{const.}}(\mathbf{v},\mathcal{R}|T)}}{e^{-N f_{\text{path}}^{\text{unconst.}}(\mathbf{v}|T)}}, \tag{8.20}$$

from which the escape probability can be computed as $P_{\text{escape}}(\mathcal{R}|T) = 1 - P_{\text{stay}}(\mathcal{R}|T)$ [Mauri et al. (2023b)]. Although both Eqs. (8.19) and (8.20) are well defined for any possible interacting potential Φ, these probabilities acquire an evolutionary interpretation in the case of the Evo potential. Consistently with Eigen's theory of evolution [Eigen (1971); Leuthäusser (1987)], they estimate probabilities resulting from an evolutionary dynamics consisting of mutations at rate μ combined with selection with probability P_{model}. Eigen's model is highly appropriate for describing the evolution of quasispecies, which can be visualized as dynamic "clouds" of genotypes evolving rapidly due to high mutation rates. In this scenario, a substantial proportion of offspring are expected to inherit one or more mutations compared to their parents. Notably, this class of quasispecies models faithfully captures the essence of SELEX experiments and viral evolution.

8.2. Application to data-driven protein models

We now apply the mean-field theory introduced in Section 8.1 to study transition paths in landscapes trained from protein sequence data using RBM. In particular, we will consider two RBMs trained on Lattice Proteins (LP) and WW domains, as introduced in the previous chapter (also look at Appendix C for more details). Despite the fact that mean-field theory is exact only in the large-N limit, it already provides a good approximation for small systems ($N = 27,\ 31$ for LP and WW, respectively) as we will see below [Mauri et al. (2023a,b)]. To improve the mean-field approximation in this context, one should evaluate the Gaussian correction around the saddle-point or, alternatively, consider the limit where both M and N are of the same order of magnitude. A possible application, in the case of RBM, would be the so-called compositional phase described in Tubiana and Monasson (2017), where each data configuration activates a finite number of hidden units. In this scenario, our aim is to describe the free energy of the system when only a finite number of patterns are active, while the others act as white noise. However, this topic goes beyond the scope of this thesis.

8.2.1 Application to Lattice Proteins

We start by considering the toy-model of Lattice Proteins (LP) [Lau and Dill (1989)]. This model has already been introduced in Section 7.2. In LP we can write the probability p_{nat} of a sequence $\mathbf{v}$ (with $N = 27$ sites) to fold in a given structure $\mathbf{S}$. We already discussed how this probability can be used to sample sequences with high folding probability that can be later used to train an RBM (details in Appendix C.1).

We aim to use this model to test our mean-field theory. We start by numerically compute solutions of paths (corresponding to the order parameters $\mathbf{m}$ and $\mathbf{q}$) connecting two far away target sequence with high $p_{\rm nat}$ for both the global and direct cases:

$\mathbf{v}_{\rm start}$ =DRGIQCLAQMFEKEMRKKRRKCYLECD,

$\mathbf{v}_{\rm end}$ =RECCAVCHQRFKDKIDEDYEDAWLKCN.

These two configurations are characterized by a flip of the charge (from negative to positive) of the amino-acids in the site 25 (from E to K) and of the neighboring sites (see Fig. 8.3(d)) to keep an attractive interaction between such sites in order to guarantee the stability of the fold. See Section 7.2 for more details about electrostatic modes in LP. The trajectories of the inputs m_t^μ and of the overlaps q_t reveal which and when latent factors of RBM enter into play in the transition.

Figure 8.3(a) shows the trajectories of inputs associated to the weights in Fig. 8.3(c) (corresponding to hidden variable $\mu = 39$ and 40). These two hidden variables are strongly activated at sites that are in contact in the tertiary structure of the protein (Fig. 8.3(d)) and are consequentially relevant for its stability. While the logo of w_{39} shows that the interaction between site 25 and its neighbors can be realized through electrostatic forces between charged amino acids w_{40} tells that contacts between sites 5,6,11 and 22 can be realized through disulfide bonds between Cysteines (C). The dynamics of the projection m^{40} (Fig. 8.3(a)) explains how global optimal paths exploit Cysteine-Cysteine interactions (not present in the initial and final sequences) in order to maintain the structural stability through transient mutations to C-C in the sites 5,6,11 of the protein. These C-C bonds are then lost in the final configuration, as clearly seen by the decrease of the projection m^{40}. Along global paths, most of the intermediate mutational steps do not abruptly changes the order parameters, with the exception of the bump in the overlap q at step $\sim$10, possibly related to the presence of preparatory mutations for the Cys-related transition in Fig. 8.3(a).

Using Eq. (8.16) we can compute the amino acids frequencies at each site along the path and use this information to estimate the average log-likelihood and $p_{\rm nat}$ at each step. To estimate the $p_{\rm nat}$ we use this frequencies to build an independent site model that approximate the true marginal distribution of sequences, then we use this model to sample many sequences at a given step t and compute the average $p_{\rm nat}$ from this samples. The results shown in Fig. 8.4 confirm very good values for the probabilities of intermediate sequences along the path, both for p_{nat} (Fig. 8.4(a)) and for the model $P_{\rm RBM}$ (Fig. 8.4(b)). We also observe that sequences along the global paths have substantially higher probabilities than along direct paths for the values of T and γ considered.

8.2.2 Application to WW domains

We apply the above approach to RBM models learnt from sequence data of the WW family extracted from public database (PFAM id: PF00397) [Sudol (1996); Sudol and Hunter (2000)] (here $N = 31$). Details about this protein family and the trained RBM are given in Sections 7.3.1 and 7.3.2. In particular we will study paths connecting sequences that have been tested to belong to different specificity classes.

Cont and Evo mean-field paths between class-specific WW domains are shown in Fig. 8.5(b). Both solutions follow similar traces in the specificity plane, in agreement with the paths in Fig. 7.6(a). However, mutations are homogeneously spread along the Cont path, with $\simeq N\gamma/T$ mutations at each step (Fig. 8.5(c-d)). Conversely, the Evo path is highly heterogeneous, with some steps accumulating many mutations and others barely any. In Appendix D.2, we provide the list of the consensus sequences computed with

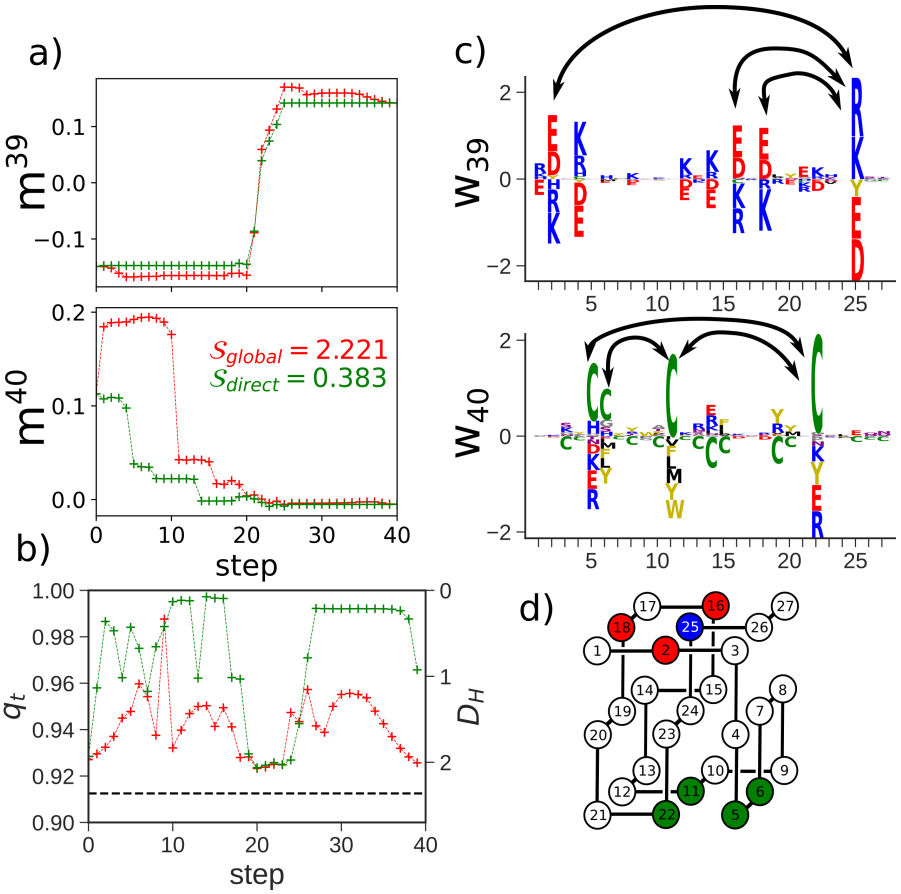


Figure 8.3: Mean-field description of mutational paths in lattice proteins with the Cont potential. (a) Values of two relevant inputs vs. number t of mutations along paths of length $T = 40$. Red and green lines correspond to, respectively, global and direct paths. Parameters: $\beta = 3$, $\gamma = 3.5$. In the inset we show the entropy for the global and direct solutions. (b) Overlap q_t (left scale) and average number of mutations $D_H = N(1-q_t)$ (right scale) between sequences at steps t and $t+1$ vs. t; The dark line shows q_c. (c) Logos of the attached weights $w_{i,\mu}(v)$. Positively charged amino acids are in blue, negatively charged ones in red. (d) Reference structure for the Lattice Protein model. Image and caption from Mauri et al. (2023b).

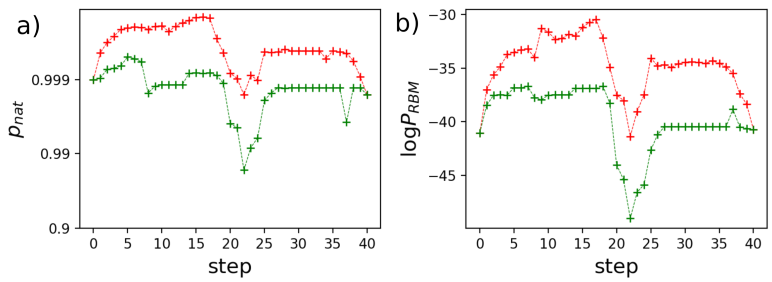


Figure 8.4: Average value of p_{nat} and log-likelihood along the paths for lattice proteins estimated from the mean-field global (red) and direct (green) solutions shown in Fig. 8.3. Image and caption from Mauri et al. (2023b).

mean-field theory. Interestingly, most steps along the Evo path I→IV are concentrated in the region characterized by promiscuous sequences binding both ligand classes as mentioned in Section 7.3.3. The linearity of Φ_{Evo} makes the transition probabilities π in $\mathcal{P}$ in Eq. (7.2) independent of the location of mutations, concentrating intermediate sequences in the region of highest fitness.

8.2.2.1 Mean-field based estimation of evolutionary distance

As an application of our mean-field approach we show how it can be used to estimate evolutionary distances between sequences with complex data-driven models, including epistatic interactions between residues. The probability that sequence $\mathbf{v}_{end}$ be reached after T steps of stochastic mutations with rate μ starting from $\mathbf{v}_{start}$ is given by Eq. (8.19).

This probability can be computed as a function of T to determine the optimal time (evolutionary distance) T^* at which it is maximal. For purely neutral evolution, $f_{path}^{free} = 0$ and the probability $P(\mathbf{v}_{start} \to \mathbf{v}_{end}|T)$ can be exactly computed; T^* then coincides with the predictions of Kimura's theory of neutral evolution [Kimura (1983)], see Appendix D.1. T^* can also be easily computed for profile models [Felsenstein (2004)], where selection acts independently from site to site, see Fig. 8.5(e) for an illustration of WW. Our mean-field theory theory allows us to go well beyond profile models, and to compute the probability P in the presence of epistatic effects in the RBM model inferred from WW sequence data. Figure 8.5(e) shows that the evolutionary distance T^* may then substantially differ from its profile counterpart, showing the effectiveness of our mean-field approach to deal with complex sequence models.

We want to stress at this point some relevant aspects of the mean-field approach presented here. In phylogenetic inference, most methods rely on models of evolution acting independently on each site (like the profile model described above) [Felsenstein (2004)]. Inferring evolutionary history from data based on coupled (i.e. epistatic) models of selection is a notoriously hard problem and literature still lacks consolidated tools for this task. Some efforts in this direction have been done by Rodríguez-Horta et al. (2022); Magee et al. (2021) (also see Chapter 13 and 16 of the book by Felsenstein (2004)). Our mean-field-based approach makes possible to estimate evolutionary distances with complex selection models at low computational cost (linear in sequence length), opening the way to ancestral reconstruction and to the prediction of phylogenetic trees with with data-driven, epistatic models. However, further validations and comparisons with other phylogeny inference approaches are needed.

In the next sections we are going to study in more details paths connecting WW domains of class I [Espanel and Sudol (1999)] to class IV [Russ et al. (2005)] (see Section 7.3.1). The amino-acid sequences are given by:

$$\mathbf{v}_{\text{start}} = \texttt{LPAGWEMAKTSS-GQRYFLNHIDQTTTWQDP} \text{ (class I)} ,$$

$$\mathbf{v}_{\text{end}} = \texttt{LPKPWIVKISRSRNRPYFFNTETHESLWEPP} \text{ (class IV)}.$$

8.2.2.2 Direct-to-global phase transition

As we already discussed previously in Sections 7.2 and 8.2.1, global paths are generally preferred to direct ones due to their ability of exploring the sequence space more efficiently and looking for more stable (*i.e.* more probable) configurations. However, this is true only if the interacting potential Φ allows for longer paths (in terms of total mutations) in order to explore efficiently the sequence space. In the limit case of very stiff potentials, paths will not be able to introduce novel mutations apart from those of the two edge sequences, and direct paths will be preferred.

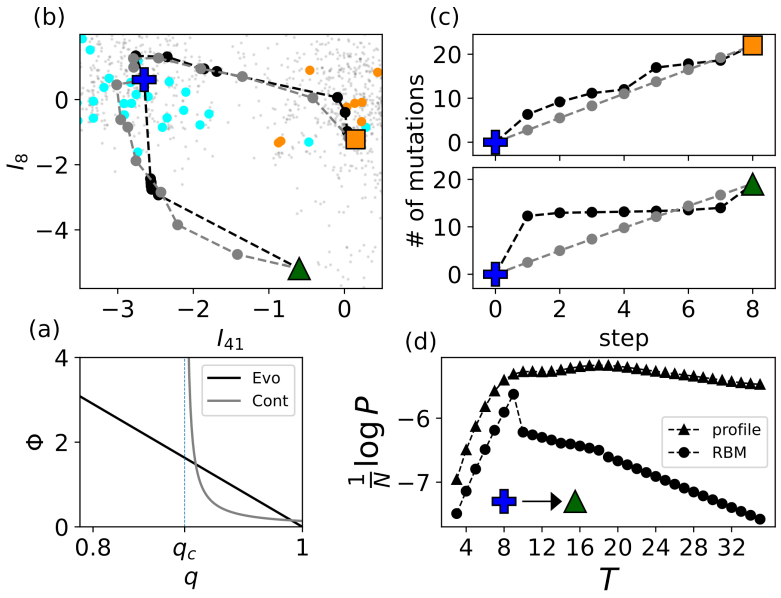


Figure 8.5: Mean-field theory of mutational paths for the RBM model trained on WW domain. (a) Sketches of the potentials Φ_{Evo} (black) and Φ_{Cont} (gray) vs. q. (b) Same two-dimensional representation as in Fig. 7.6(a) for the mean-field paths with Evo (black lines) and Cont (grey lines) potentials. (c) Cumulative numbers of mutations vs. t. Here, $\mu = 10^{-5}$ and $\gamma = 0.9$, so that the cumulative numbers match for $t = T$. (d) Log-probability of joining class I and class IV natural WW domains in T steps with the profile (triangles) and RBM (circles) models. Jumps signal the onset of several new mutations, *e.g.* 4 in the mean-field *free* path at $T = 10$. Image and caption from Mauri et al. (2023a).

Understanding this trade-off quantitatively in terms of the parameters of the paths is a natural question from a physiscist's perspective. Moreover, from an experimental point of view, this question can be helpful to understand to which extent exploring longer paths can result is better designed proteins. In the context of mean-field theory, we are particularly interested to understand the direct-to-golbal transition in the Cont scenario, since it is more closesly related to the Monte Carlo sampling of Chapter 7 and consequently to the experimental design of mutational paths. In Chapter 9, we will study a simple Hopfield-Potts model, where the direct-to-global phase transition is analytical tractable. Despite being simple, this model shows an interesting phenomenology and interpretation of such transition.

Below, we show that direct-to-global transitions are observed in mutational paths joining natural WW sequences, see Figure 8.6. This figure shows in particular the presence of a cross-over, when the path length T is kept fixed, between direct and global solution at a value of $\gamma \sim 0.92$ and another jump at $\gamma \sim 1.3$, corresponding to the insertion of a novel mutation outside the direct space.

To further study the difference between direct and global solutions at different values of γ, we can compute what and where the first relevant mutations that push the solutions outside the direct space should be considered. Differently stated, given a direct path computed for certain value of length T and potential stiffness γ, we would like to know what sites will be the first to mutate outside the direct space immediately after we release the constraint on the path to be direct (*i.e.* we compute the mean-field solution only considering as accessible sites the ones present at the target sequences). To do so, we use Eq. (8.16) to compute the frequencies of each amino acid $\langle \delta_{v_{it},a} \rangle$ in the global space (where

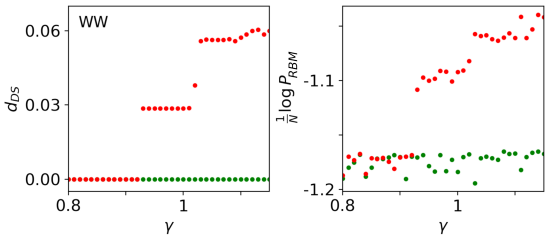


Figure 8.6: Direct-to-global phase transition in WW domain. Mean-field estimates of d_{DS} (*Left*) and of $(\log P_{\mathrm{RBM}})/N$ (*Right*; red: global paths, green: direct) vs. γ for mutational paths of the WW domain of length $T = 10$. In all panels $\beta = 3$. Image and caption from Mauri et al. (2023b).

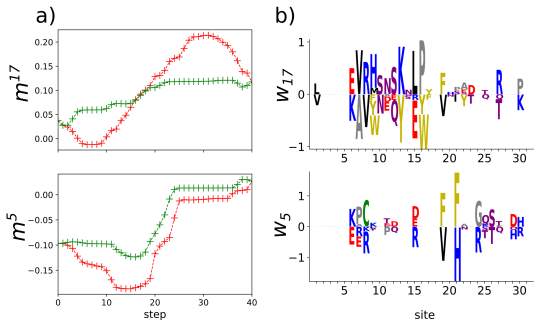


Figure 8.7: Direct and global transition path in WW domain. (a) Values of two relevant inputs. Green and red paths correspond to direct and global solutions, respectively. (b) Logo of the weights associated to the inputs. Simulation parameter: $\gamma = 1.6$, $T = 40$, $\beta = 3$. Image and caption from Mauri et al. (2023b).

the transfer matrix that defines Z_i^{1D} is of size 21×21) around the direct solution. Then we compute the probability assigned to non-direct amino acids at some point by the direct mean-field solution, $p_{\mathrm{DS}}^{\mathrm{out}}(i,t)$, as:

$$p_{\mathrm{DS}}^{\mathrm{out}}(i,t) = 1 - \langle \delta_{v_{it},v_{\mathrm{start},i}} \rangle_{\#_{\mathrm{dir}}} - \langle \delta_{v_{it},v_{\mathrm{end},i}} \rangle_{\#_{\mathrm{dir}}} , \tag{8.21}$$

where $\langle \cdot \rangle_{\#_{\mathrm{dir}}}$ indicates that the average is computed only over the direct solution.

Results for different values of γ are shown in Fig. 8.8. As expected for higher values of γ the interaction potential Φ_{Cont} becomes less stiff and allows the emergence of more mutations escaping the direct space. In the case of $\gamma = 1$ the Cont potential is stiff enough to allow only one mutation outside the direct space. In particular this mutation appears in the middle of the path and stays until the very end (before returning to the final state at step 10), showing that the path has to reach a proper region of the sequence space before engaging non-direct mutations. The difference between these global mutations computed on the direct solution and the global solution is shown in Fig. 8.9, where we used Eq. (8.16) to compute the frequencies of each amino acid. This approach can be useful to improve mutagenesis experiments by suggesting a minimal number of mutations outside the direct space that can already improve the quality of the intermediate sequences.

Differences between direct and global solutions in the case of WW domain can be observed in Fig. 8.7. Here we plot the values of two relevant inputs along both type of paths. The two weights have been chosen to between those that maximise the difference between direct and global solutions. In particular, we see that the projection along the weight w_{17} for the direct solution remains almost constant compared to the global case. On the other hand, projection along the weight w_5 shows a switch in both cases, with global solutions showing a stronger activity [Mauri et al. (2023b)].

8.2.2.3 Entropy of doubly-anchored paths

In Section 8.1.4, we showed how to compute the number of relevant transition paths, corresponding to the entropy of the system, $\mathcal{S}_{\mathrm{path}}$. Estimates of $\mathcal{S}_{\mathrm{path}}$ in the Cont and Evo scenarios are shown in Fig. 8.10(a). The first important aspect to be noted regards the scaling of $\mathcal{S}_{\mathrm{path}}$ with the path length T: while in the Evo scenario the entropy seems to grow linearly with T, we notice a slower growth with T in the Cont scenario. This

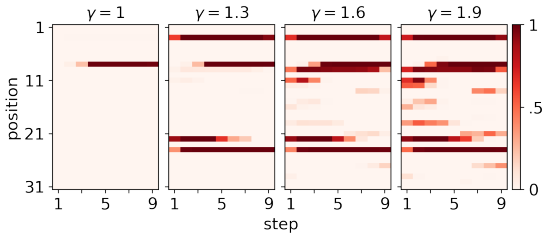


Figure 8.8: Probability of non-direct amino acids along direct paths as a function of the step t (x-axis) and of the sequence site i (y-axis) for the WW domain. Results are shown for four values of γ, see panels. Parameter: $\beta = 3$. Image and caption from Mauri et al. (2023b).

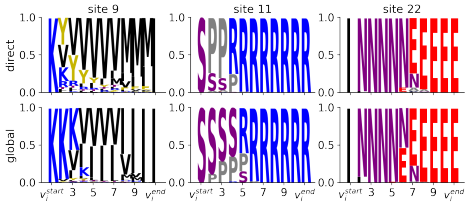


Figure 8.9: Logos of the amino-acid frequencies at three arbitrarily chosen sites along a path of length $T = 10$ joining two WW domains. (*Top*) logos computed using the MF direct solution; the two amino acids allowed on each site in the direct subspace are the ones corresponding to v^{start} and v^{end}. The other amino acids are candidate for mutations outside the direct space. (*Bottom*) logos computed using the MF global solution. Here $\gamma = 1.6$ and $\beta = 3$. Image and caption from Mauri et al. (2023b).

behaviour can be understood in the following toy model. We consider a uniform (flat) landscape P_{model}, without constraint on the final sequence. In the Evo scenario, it is easy to show that each time step corresponds, on average, to a constant number of mutations whose value depends on μ and on A only. Hence, the entropy is approximately added to the logarithm of this number at each step, and the total entropy will scale linearly with T. In the Cont scenario, the number of possible configurations at each step is bounded from above by the hard wall in Φ_{Cont}, defined by the overlap $q_c = 1 - \gamma/T$. Considering that $\Phi_{\text{Cont}}(q > q_c) \ll 1$, each sequence along a path will have on average ρN mutations with respect to the previous sequence, where $\rho = \gamma/T$ is the mutation probability per site. We then estimate the entropy of a binary variable (mutation or no mutation on each site with probability ρ) as $-\rho \log \rho - (1-\rho)\log(1-\rho) \simeq \frac{\gamma}{T} \log T$ for large T. Hence, the total entropy (per site) of the paths of length T is expected to scale as $\sim \log T$ [Mauri et al. (2023b)].

8.2.2.4 Case of paths anchored at the origin

The partition function for paths in Eq. (8.2) is computed on the ensemble of paths fixed at both ends to be equal to sequence $\mathbf{v}_{start}$ and $\mathbf{v}_{end}$. One can easily redo the computation when the last extremity is left free. We show in Fig. 8.10(a) the entropies of these partially unconstrained paths for the Cont and Evo potentials.

In the Evo scenario the unconstrained solution shows lower entropy than the constrained one, while it has a higher entropy in the Cont scenario as intuitively expected. This apparently surprising finding can be explained as follows. For the constrained, doubly-anchored paths $\mathbf{v}_{\text{end}}$ has relatively high energy (see Fig. 8.4(b)), and many paths connect this last sequence to $\mathbf{v}_{\text{start}}$. Conversely, in the unconstrained case, paths are attracted to a lower free-energy minimum, and there are fewer paths connecting the initial configuration to this final region. The presence of a hard wall in the Cont scenario forbids both solutions to remain in the same configurations for long times and to then jump directly to another distant point in sequence space. Hence, Cont solutions will explore many more different configurations making their entropy higher with respect to their Evo counterparts. Moreover, since the constrained solution in the Cont case has to smoothly interpolate between distant regions in such a way that the energy along the path is optimized, this makes the number of accessible paths lower than in the unconstrained solution.

Our mean-field formalism allows us to compute the probability to go from $\mathbf{v}_{start}$ to $\mathbf{v}_{end}$

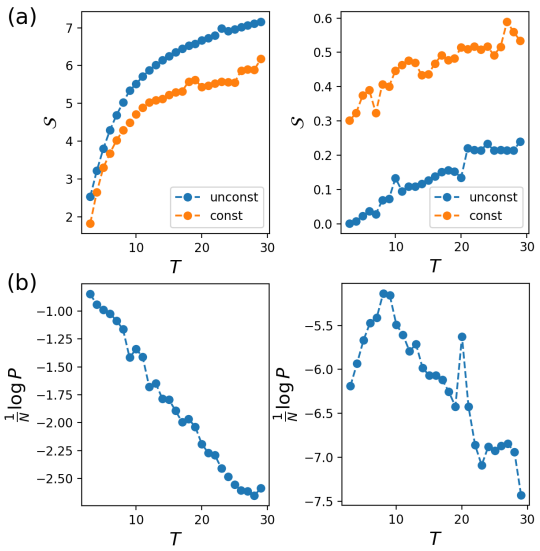


Figure 8.10: Entropies and probabilities of transition for the Cont (left) and Evo (right) potentials. (a) Entropy $\mathcal{S}_{\text{path}}$ of paths as a function of T. Results are shown for paths joining the two WW domain wild-type sequences (constrained) and paths anchored by the starting sequence and free at the other extremity (unconstrained). (b) Probability of a transition path as a function of T. Parameters for Evo: $\mu = 10^{-4}$, $\beta = 1$; for Cont: $\gamma = 3$, $\beta = 1$. Image and caption from Mauri et al. (2023b).

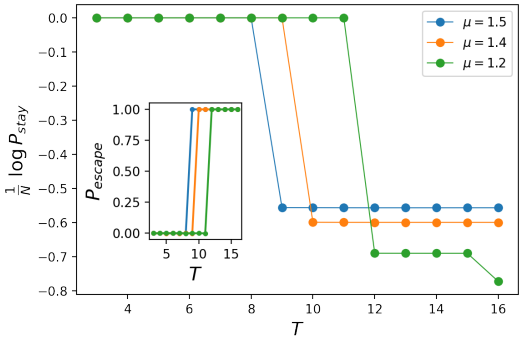


Figure 8.11: Probability of remaining in (main panel, log. scale along y-axis) and of escaping from (inset) the neighborhood of $\mathbf{v}_{start}$ for the WW domain, computed using Eq.(8.20) in the Evo scenario ($\beta = 1$). Three different values of the mutation rate μ are considered. Image and caption from Mauri et al. (2023b).

in T dynamical steps, see Mauri et al. (2023a) and Sections 8.1.4–8.2.2.1. As we said earlier, this probability acquires an evolutionary interpretation in the case of the Evo potential. It estimates the probability to join the two sequences in T steps consisting of mutations at rate μ (per step) combined with selection with probability P_{model} [Leuthäusser (1987)]. We show in Figure 8.10(b) the transition probabilities for the Cont and Evo scenarios. The Evo scenario shows an optimal length T^* for which the probability is maximised, while, in the Cont scenario, the transition probability decreases linearly with T. This may be explained from the fact that the Evo potential emulates a mutational dynamics in which T^* plays the role of an evolutionary distance between the two edge sequences (see Section 8.2.2.1). On the contrary the emergence of this optimal T^* is forbidden in Cont scenario by the stiffness of Φ_{Cont}, which increases with T.

Furthermore, we also discussed in Section 8.1.4 how the mean-field framework allows us to compute the probability of remaining in the minimum of the free-energy landscape corresponding to the starting sequence towards some region $\mathcal{R}$ of the sequence space in T steps. We define $P_{\text{stay}}(\mathcal{R}|T)$ as in Eq. 8.20. In Fig. 8.11 we plot the probability of remaining in the region associated to $\mathbf{v}_{\text{start}}$ for the WW domain energy landscape and in the Evo scenario. For different values of μ we are able to estimate at which time an evolving configuration is supposed to escape from the minimum. We observe the existence of a trade-off between the time and the probability of sojourn in the starting region depending on the value of μ [Mauri et al. (2023b)].

9

Direct-to-global phase transition in simple Hopfield models

We introduced in Chapter 8 a mean-field framework to study mutational paths in energy landscapes defined by Restricted Boltzmann Machines. In this framework, each path can be seen as a succession of points in a multi-dimensional space, with dimension corresponding to the number of hidden neurons M of the RBM, and interacting between each other through an elastic potential Φ (Figure 8.1 in Chapter 8). Different dynamics can be modeled through different choices of Φ: the first one, referred to as Cont, ensures a smooth interpolation between sequences along the path and avoid 'jumps' between configurations. The second potential, called Evo is inspired by evolutionary biology *i.e.* it mimics random mutations at a constant rate μ [Leuthäusser (1987)], while the energy landscape plays the role of the selective pressure driving the evolution.

Mean-field approximation allows us to access a number of useful statistics of the paths: from the total number of accessible paths, to transition probabilities and escape times from metastable regions of the sequence space. In particular, we discussed the difference between direct and global paths as defined in the introduction of Chapter 7. If the elastic potential is not to stiff, the system will allow for more flexible, longer global paths that explore the sequence space looking for low energy (high fitness) intermediate configurations. In Section 8.2.2.2, we discussed a direct-to-global transition for paths interpolating different specificity classes in WW domains, as a function of the stiffness of the elastic potential (in the Cont scenario defined in Section 8.1.1).

Understanding more precisely the transition between direct and global paths is the main goal of this Chapter. We will introduced a toy Hopfield-Potts model (or Minimal Hopfield-Potts model, MHP) with $P = 2$ non-orthogonal patterns. This system allows us to study transition paths analytically using mean-field theory. In the Cont case, we will unveil the existence of a direct-to-global phase diagram controlled by the stiffness of the interacting potential, γ (controlling the number of mutations allowed at each step), and the total number of steps of the path, together with the inner structure of the energy landscape. In particular, we will stress that the overlap between the patterns of the model is crucial for the existence of a direct-to-global transition, suggesting a possible interpretation on how the same phenomenon occurs in more complex models.

The chapter is organized as follows. To help the reader to follow the rather technical discussion, we will start by giving a brief introduction of the model and summurize briefly the main results that will be analytically derived in the following sections. We will then apply the mean-field framework introduced in Chapter 8 to the MHP model, deriving the exact phase diagram for the direct-to-global transition in the Cont scenario and discussing other features of the model in the case of paths only anchored at one end. We will finally discuss possible interpretation of these results as weel as possible future directions of research.

The results presented in this Chapter are based on our original work published in *Transition paths in Potts-like energy landscapes: general properties and application to protein sequence models* [Mauri et al. (2023b)].

9.1. Definitions and overview of the results

We recall the definition of the energy of a path $\mathcal{V} = \{\mathbf{v}_0 = \mathbf{v}_{\text{start}}, \mathbf{v}_1, \dots, \mathbf{v}_{T-1}, \mathbf{v}_T = \mathbf{v}_{\text{end}}\}$ in a Potts-like energy landscape E_{model}, as introduced in Chapter 8, Eq. 8.1:

$$\mathcal{E}(\mathcal{V}) = \sum_{i=0}^{T-1} \Phi(q(\mathbf{v}_i, \mathbf{v}_{i+1})) + \frac{1}{N} \sum_{i=0}^{T} E_{\text{model}}(\mathbf{v}_i)\,, \tag{9.1}$$

where the interacting potential can be either the Cont potential, Eq. 8.3, or the Evo potential, Eq. 8.4:

$$\Phi_{\text{Cont}}(q) = \frac{1}{T^2 |q - 1 - \gamma/T|}\,, \tag{9.2}$$

$$\Phi_{\text{Evo}}(q) = (1-q) \ln\left(1 + \frac{A}{e^{\mu A/(A-1)} - 1}\right), \tag{9.3}$$

where γ is the stiffness of the potential, controlling the number of mutations allowed at each step, while A is the alphabet size (*e.g.*, total number of amino acids in the case of protein models) and μ is the mutation rate. We now introduce a minimal setting, where the properties of transition paths can be analytically characterized.

9.1.1 Minimal Hopfield-Potts model

We first define the energy landscape for Potts configurations. We consider a Hopfield model for categorical data, hereafter referred to as Hopfield-Potts. There are $A \geq 3$ states per site (called a, b and c and so on). The energy of our Minimal Hopfield-Potts (MHP) model reads

$$E_{\text{MHP}}(\mathbf{v}) = -\frac{J}{2N} \sum_{i,j} \left(w_{1i}(v_i) w_{1j}(v_j) + w_{2i}(v_i) w_{2j}(v_j)\right), \tag{9.4}$$

where the two patterns $\mathbf{w}$ are constructed as follows:

$$\begin{aligned} w_{1i}(v_i) &= \delta_{v_i,a} + \omega\, \delta_{v_i,c}\,, \\ w_{2i}(v_i) &= \delta_{v_i,b} + \omega\, \delta_{v_i,c}\,, \end{aligned} \tag{9.5}$$

and ω is a positive parameter that controls how much the two patterns overlap. The coupling strength J is supposed to be large, but its precise value does not affect the qualitative description below. The energy of a configuration $\mathbf{v}$ is a quadratic function of its two projections along the patterns, denoted as $m^\mu(\mathbf{v}) = \frac{1}{N} \sum_i w_{i\mu}(v_i)$ $(\mu = 1, 2)$:

$$E_{\text{MHP}}(\mathbf{v}) = -\frac{J}{2} N \left[\left(m^1(\mathbf{v})\right)^2 + \left(m^2(\mathbf{v})\right)^2\right], \tag{9.6}$$

The MHP model is therefore intrinsically of mean-field nature, and can be easily solved in the large-N limit.

A sketch of the free-energy of the MHP model in the (m^1, m^2) plane is shown in Figure 9.1. Depending on the value of ω two cases must be be distinguished:

- For $\omega < \frac{1}{2}$, the only minima of the free energy are $(m^1, m^2) = (m^*, 0)$ and $(0, m^*)$, with $m^* \simeq 1$. As a consequence, the only configuration with non–negligible probabilities are the two patterns themselves.

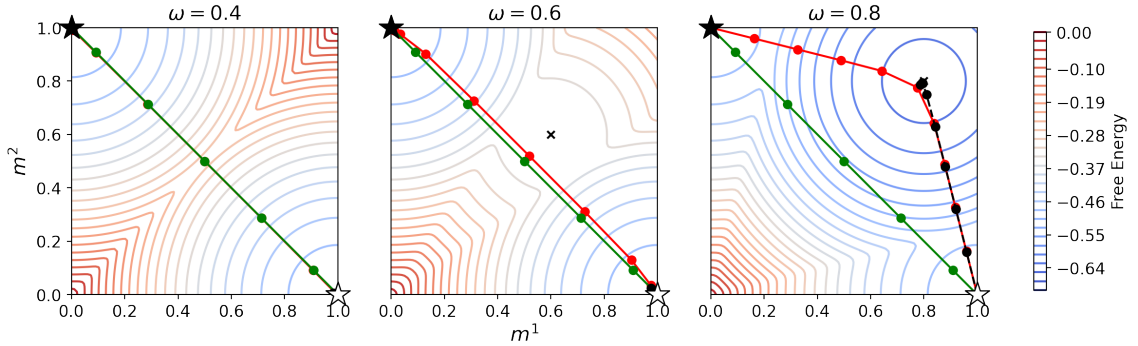


Figure 9.1: Sketch of the direct-to-global phase transition in the MHP model. Green paths correspond to path constrained in the direct space, while red are global (free to explore any configuration). Stars represent the initial and final configurations, see corners of the free-energy landscape. Each plot represents the MHP free energy landscape for a fixed value of ω, showing the crossover between direct and global transition paths as the symmetric minimum of the free energy becomes more and more attractive, *i.e.* as the overlap between the patterns ω increases and as the length of the path increases. The free energy landscape is represented by iso-free energy lines (see colorbar on the right). Black dashed lines correspond to paths computed without fixing the end, showing a transition between paths staying close to initial minimum (*i.e.* the white star) and paths that jump into the intermediate minimum when this becomes stable for higher values of ω. Here the parameters β and γ appearing in Eqs. (8.2) and (8.3) are equal to, respectively, 6 and 3. The length of all paths defined in Eq. (8.1) is set to $T = 10$, but only points that are different from the endpoints (*i.e.* white and black stars) are shown. Image and caption taken from Mauri et al. (2023b).

- For $\omega > \frac{1}{2}$, a new local minimum will appear at $(m^1, m^2) = (\omega, \omega)$, which we refer to as symmetric minimum later. This local minimum becomes global when $\omega > \frac{1}{\sqrt{2}}$. Therefore, the energy landscape includes a region, far away from the pattern-associated configuration, which is energetically favorable.

9.1.2 Transition paths anchored at both extremities

In this landscape, we will consider paths of configurations anchored at both extremities, *i.e.* such that $\mathbf{v}_{\text{start}} = \{a, a, a, ..., a\}$ and $\mathbf{v}_{\text{end}} = \{b, b, b, ..., b\}$. For the sake of simplicity, we will restrict ourselves to the Cont potential, see Eq.(9.2). As the Hamming distance between the two edges of the path is equal to $D = N$, the flexibility parameter γ must be larger than 1.

The properties of the mutational paths associated to this energy landscape can be analytically characterized. The mean-field theory associated to paths is more sophisticated than for single configurations. Explicit expressions can nevertheless be derived for the average projections m_t^1, m_t^2 of intermediate configurations $\mathbf{v}_t$ and for the average overlap q_t between successive sequences $\mathbf{v}_t, \mathbf{v}_{t+1}$; Detailed calculations and results are reported in Section 9.2. Briefly speaking, we find that, see Fig. 9.1:

- For $\omega < \frac{1}{2}$, the optimal path connecting the starting and ending configurations is direct. Due to the absence of favorable regions in the landscape outside the neighborhoods of the anchors paths have no incentive to explore the landscape: they directly interpolate between $\mathbf{v}_{\text{start}}$ and $\mathbf{v}_{\text{end}}$ to minimize their elastic energy.

- For $\omega > \frac{1}{2}$, the symmetric minimum attracts mutational paths and make them leave the direct space. If paths are sufficiently long and flexible they are deviated by this minimum, and explore the global configuration space.

While the precise locus of the paths are specific to the MHP model, the coincidence of the onset of the transition with the existence of favorable regions in the configurations space is a general phenomenon. The nature of optimal transition paths is therefore intimately related to the structure of the energy landscape.

9.1.3 Transition paths anchored at one extremity

We consider the case in which $\mathbf{v}_{\text{end}}$ is not fixed. In this scenario, this final configuration is very likely, for large N, to lie in one of the global minima of the free energy. If the starting configuration is attached to another minimum, then paths can explore the space in its diversity, see Fig. 9.1(right) for an illustration.

As shown in Chapter 8, our mean-field theory can be adapted to the case of paths anchored at one extremity, and allows us to estimate the time, *i.e.*, the minimal length necessary for a path to escape some region $\mathcal{R}$ of the configuration space. In practice, a region is defined as the local minimum of the mean-field free energy containing $\mathbf{v}_{\text{start}}$. We define the probability of paths of length T to stay in $\mathcal{R}$ through the ratio of the statistical weight, defined in Eq. (8.2), of all the paths constrained to end in $\mathcal{R}$ and the weight associated to unconstrained paths, *i.e.* free to wander in the configuration space. This probability reads (corresponding to Eq. (8.20) of the previous chapter)

$$P_{\text{stay}}(\mathcal{R}|T) = \frac{\sum_{\mathcal{V}:\mathbf{v}_{\text{end}}\in\mathcal{R}} e^{-N\mathcal{E}(\mathcal{V})}}{\sum_{\mathcal{V}} e^{-N\mathcal{E}(\mathcal{V})}} \underset{N\gg 1}{\sim} \frac{e^{-Nf_{\text{path}}^{\text{const.}}(\mathbf{v}_{\text{start}},\mathcal{R}|T)}}{e^{-Nf_{\text{path}}^{\text{unconst.}}(\mathbf{v}_{\text{start}}|T)}}, \tag{9.7}$$

where $f_{\text{path}}^{\text{const.}}$ and $f_{\text{path}}^{\text{unconst.}}$ are the free energies associated to, respectively, constrained and unconstrained paths. The calculation of these free energies is reported in Section 9.2.1 and 9.2.4. For the MHP model, we observe that, above a certain crossover value of ω (depending on T), the path is very likely to escape from the local minimum at $(m^1,m^2) = (m^*,0)$ and to end up in (ω,ω), see Fig. 9.1.

9.2. Mean-field theory and direct-to-global transition for the MHP model

In this section we describe the mean-field theory treatment of paths in the MHP landscape following Mauri et al. (2023b), solve the corresponding self-consistent equations for the order parameters $\{m_t^1, m_t^2, q_t\}$ along the path, and then characterize the nature of the transition. The computation of the free energy of paths for general Restricted Boltzmann Machines have been already reported in Section 8.1. MHP model is a special case of RBM, with $M = 2$ patterns defined in Eq. (9.5), no local fields on the visible units, $g_i = 0$, and quadratic activation function of the hidden unit, $\Gamma_\mu(m) = \frac{m^2}{2N}$.

9.2.1 Free-energy for paths

To make the theory easier to interpret in the case of the MHP model, we introduce the three projections of a configuration (denoted by $\tilde{m}_t^\mu$, with $\mu = 1,2,3$) along the vectors $\delta_{v_i,a}$, $\delta_{v_i,b}$, $\omega\,\delta_{v_i,c}$, see Fig. 9.2. While introducing an additional order parameter compared to the number of patterns makes the computation slightly more lengthy, it offers the major advantage to allow for immediate distinction between direct ($\tilde{m}^3 = 0$) and global ($\tilde{m}^3 > 0$)

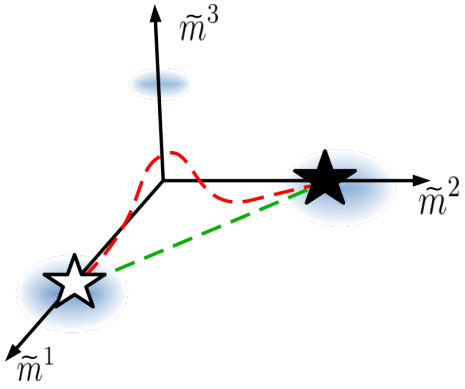


Figure 9.2: Transition paths in the 3D-space of projections $\tilde{\mathbf{m}}$ for the MHP model presented in Section 9.1. Direct solutions (in green) linearly interpolate in the input space the two minima of the energy landscape. The global solutions are pushed away from the direct ones by the presence of a third minima emerging from the overlap ω between the two patterns of the model defined in Eq. (9.5). Image and caption taken from Mauri et al. (2023b).

paths. With this choice, we rewrite the free energy of the path as

$$f_{\text{path}}(\beta, \tilde{\mathbf{m}}, \mathbf{q}) = \sum_t \Big(\frac{1}{2}(\tilde{m}_t^1)^2 + \frac{1}{2}(\tilde{m}_t^2)^2 + (\tilde{m}_t^3)^2 + \\ + \tilde{m}_t^3(\tilde{m}_t^1 + \tilde{m}_t^2) \Big) + \sum_t (\Phi(q_t) - q_t \Phi'(q_t)) - \frac{1}{\beta} \log Z_{1\text{D}}\,, \quad (9.8)$$

where

$$Z_{1\text{D}} = \sum_{\{v_t\}} \exp \Big[\beta \Big(\sum_t \delta_{v_t,a}(\tilde{m}_t^1 + \tilde{m}_t^3) + \delta_{v_t,b}(\tilde{m}_t^2 + \tilde{m}_t^3) + \\ + \omega \delta_{v_t,c}(\tilde{m}_t^1 + \tilde{m}_t^2 + 2\tilde{m}_t^3) - \Phi'(q_t)\delta_{v_t,v_{t+1}} \Big) \Big] \quad (9.9)$$

As we shall see, this model undergoes a first order phase transition in the regime where $\beta \times T$ is large controlled by the overlap between patterns, ω, the length of the path, T, and the stiffness of the Cont potential, γ. We will show the existence of a stretched regime when either T and ω are small or γ is large. In this regime the minimum of the free energy corresponds to the direct solution from $\mathbf{v}_{\text{start}}$ to $\mathbf{v}_{\text{end}}$ that one obtains by restricting the sum in $Z_{1\text{D}}$ over the first two colors only. We will refer to this solution as $\#_{\text{dir}}$. If either T and ω are large or γ is small, a floppy regime arises and $\#_{\text{dir}}$ is no longer a minimum of the free energy, and the latter is minimized by global paths introducing novel mutations at intermediate steps with non zero value of $\tilde{m}_t^3$.

9.2.2 Minimization of the path free-energy in the direct subspace

To understand this phase transition, we first have to find a solution of the direct problem $\#_2$, that is, the set of parameters $\{\tilde{m}_t^{1,\text{dir}}, \tilde{m}_t^{2,\text{dir}}, q_t^{\text{dir}}\}$. The direct solution is found

by solving the following coupled equations similar to Eq.(8.12):

$$\tilde{m}_t^{1,\mathrm{dir}} = \frac{1}{Z_{1\mathrm{D}}^{\mathrm{dir}}} \sum_{\{v_t=a,b\}} \delta_{v_t,a}\, e^{-\beta E_{1\mathrm{D}}(\{v_t\})} \ , \tag{9.10}$$

$$\tilde{m}_t^{2,\mathrm{dir}} = \frac{1}{Z_{1\mathrm{D}}^{\mathrm{dir}}} \sum_{\{v_t=a,b\}} \delta_{v_t,b}\, e^{-\beta E_{1\mathrm{D}}(\{v_t\})} \ , \tag{9.11}$$

$$q_t^{\mathrm{dir}} = \frac{1}{Z_{1\mathrm{D}}^{\mathrm{dir}}} \sum_{\{v_t=a,b\}} \delta_{v_t,v_{t+1}}\, e^{-\beta E_{1\mathrm{D}}(\{v_t\})} \ . \tag{9.12}$$

where

$$E_{1\mathrm{D}} = -\sum_t \left(\tilde{m}_t^{1,\mathrm{dir}} \delta_{v_t,a} + \tilde{m}_t^{2,\mathrm{dir}} \delta_{v_t,b} - \Phi'(q_t^{\mathrm{dir}}) \delta_{v_t,v_{t+1}} \right). \tag{9.13}$$

The partition function $Z_{1\mathrm{D}}^{\mathrm{dir}}$ is the same as in Eq.(9.9) with the sum running over the states a, b only, and $\tilde{m}^3 = 0$.

We now derive the analytical expression for the mean-field solution when $T \gg 1$ (remember N was sent to infinity first). Due to exchange symmetry $a \leftrightarrow b$ we have $\tilde{m}_t^{2,\mathrm{dir}} = 1 - \tilde{m}_t^{1,\mathrm{dir}}$. We then look for a direct solution of the form

$$\tilde{m}_t^{1,\mathrm{dir}} = \tilde{m}\left(\tau = \frac{t}{T}\right) \ , \tag{9.14}$$

where

$$\tilde{m}(\tau) = \begin{cases} 1 & \text{for } \tau < \hat{x} \\ 1 - \frac{\tau - \hat{x}}{1-2\hat{x}} + \eta(\tau) & \text{for } \tau \in (\hat{x}, 1-\hat{x}) \\ 0 & \text{for } \tau > 1-\hat{x} \end{cases} \tag{9.15}$$

where $\hat{x}$ depends on T and the function $\eta(\tau)$ vanishes at large T; We will show below that η is of the order of $1/\sqrt{T}$.

As the number of mutations at each step t is equivalent to the difference in the projection $\tilde{m}^{1,\mathrm{dir}}$ between steps t and $t+1$, we write

$$q_t^{\mathrm{dir}} = 1 - \frac{\text{nb. mutations}}{N} = 1 + \tilde{m}_{t+1}^{1,\mathrm{dir}} - \tilde{m}_t^{1,\mathrm{dir}} = 1 + \frac{1}{T}\,\partial_\tau \tilde{m}(\tau) \tag{9.16}$$

to dominant order in T. Hence, the overlap order parameters are fully determined once the projection is, with the explicit expression $q_t^{\mathrm{dir}} = q(\tau = t/T)$ and

$$q(\tau) = \begin{cases} 1 & \text{for } \tau < \hat{x} \\ 1 + \frac{1}{T}\left(\frac{-1}{1-2\hat{x}} + \eta'(\tau)\right) & \text{for } \tau \in (\hat{x}, 1-\hat{x}) \ , \\ 1 & \text{for } \tau > 1-\hat{x} \ . \end{cases} \tag{9.17}$$

Our goal is to inject the above Ansätze into Eq. (9.12) and determine the function η and the value of $\hat{x}$ that solve the equation at the 0-th order in T. First, we expect the effective coupling $-\Phi'(q(\tau))$ between neighbouring v_t, v_{t+1} in the energy $E_{1\mathrm{D}}$ to scale linearly with the size of the system T. The reason is that, given a configuration $\{v_t\}$ appearing in the sum of $Z_{1\mathrm{D}}^{\mathrm{dir}}$, every couple of adjacent sites v_t and v_{t+1} occupying different states, *i.e.* for every mutation along the path, would produce an energetic penalty $-\Phi'(q_t)\delta_{v_t,v_{t+1}}$ of the order of T. The partition function will thus be dominated by the configurations $v_t = a$ for $\tau < \hat{x}$ and $v_t = b$ for $\tau > 1-\hat{x}$, that is, by configurations with a single mutation along the

path. In fact, adding more mutations would increase the energy by terms that are linear in T, while the entropy will increase only by terms of order $\log T$. Hence, configurations with more mutations will have a higher free energy and are not expected to contribute to the partition function.

Computing the derivative of the Cont potential, we obtain $-\Phi'(q(\tau)) = |\gamma - 1/(1-2\hat{x}) + \eta'(\tau)|^{-2}$. Therefore, we expect

$$\gamma - \frac{1}{1-2\hat{x}} + \eta'(\tau) \equiv \frac{\xi(\tau)}{\sqrt{T}} \ . \tag{9.18}$$

The partition function can then be rewritten as

$$Z_{\rm 1D}^{\rm dir} = T \int_0^1 {\rm d}\tau \exp\left[\beta T \left(\int_0^\tau {\rm d}y\, \tilde{m}(y) + \int_\tau^1 {\rm d}y\, (1-\tilde{m}(y)) - \frac{1}{\xi(\tau)^2} \right)\right] \tag{9.19}$$

where we explicitly integrate over the reduced 'time' τ at which the $a \to b$ mutation occurs. When $\beta T \gg 1$, the exponential integral in the partition function should not depend on τ as the mutation may take place with uniform probability in the interval $(\hat{x}, 1-\hat{x})$; hence, the mutations will happen at different times depending on the site i. Differentiating the term in factor of βT with respect to τ, we obtain the following differential equation for $\tau \in (\hat{x}, 1-\hat{x})$:

$$\tilde{m}(\tau) - (1-\tilde{m}(\tau)) - \frac{d}{d\tau}\left(\frac{1}{\xi(\tau)^2}\right) = 0\,, \tag{9.20}$$

or, equivalently in the large T limit,

$$1 - 2\frac{\tau - \hat{x}}{1-2\hat{x}} + 2\frac{\xi'(\tau)}{\xi(\tau)^3} = 0\,. \tag{9.21}$$

Solving this differential equation leads to

$$\xi(\tau) = \left[\frac{1}{\xi(\hat{x})^2} - \frac{\tau^2 - \hat{x}^2 - (\tau - \hat{x})}{(1-2\hat{x})} \right]^{-\frac{1}{2}} \ . \tag{9.22}$$

In order to ensure the continuity of $\Phi'(q(\tau))$ in $\tau = \hat{x}$, we choose $\xi(\hat{x}) = \gamma\sqrt{T}$. Integrating Eq.(9.18) over τ, we obtain

$$\eta(\tau) - \eta(\hat{x}) = \frac{1}{T^{1/2}} \int_{\hat{x}}^\tau \xi(y){\rm d}y + \left(\frac{1}{1-2\hat{x}} - \gamma\right)(\tau - \hat{x})\,. \tag{9.23}$$

Last of all, upon imposing the boundary condition $\eta(\hat{x}) = \eta(1-\hat{x}) = 0$, we also determine $\hat{x}$ as a function of γ and of T. In particular, we can expand $\hat{x}$ for large T as

$$\hat{x} = \frac{1}{2} - \frac{1}{2\gamma} - \frac{\pi}{2\sqrt{\gamma^3 T}} + o\big(T^{-\frac{1}{2}}\big)\,. \tag{9.24}$$

Consequently, $\tilde{m}(\tau) = \tilde{m}^\infty(\tau) + \mathcal{O}\left(T^{-\frac{1}{2}}\right)$ with $\tilde{m}^\infty(\tau) = 1$ if $\tau < \hat{x}^\infty = \frac{1}{2}(1 - \frac{1}{\gamma})$, $\tilde{m}^\infty(\tau) = 0$ if $\tau > 1 - \hat{x}^\infty$, and

$$\tilde{m}^\infty(\tau) = 1 - \gamma\,(\tau - \hat{x}^\infty) \tag{9.25}$$

if $\hat{x}^\infty \le \tau \le 1 - \hat{x}^\infty$. It is easy to check that Eqs. (9.10),(9.12) are fulfilled at zeroth order by this solution.

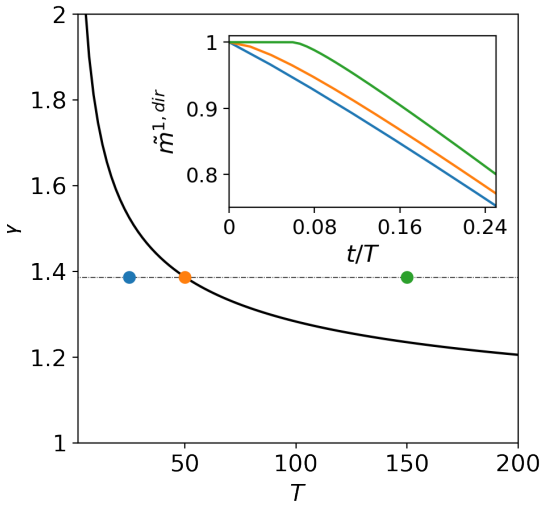


Figure 9.3: The 'understretched' and 'overstretched' sub-regimes for direct paths. The solid black line represents the root $\gamma^*(T)$ of Eq. (9.26). The three colored dots on the black dashed line $\gamma = 1.387$ correspond to $T = 25$ (blue), 50 (orange), 150 (green). The blue and green dots respectively correspond to the overstretched ($\hat{x} = 0$, $\gamma < \gamma^*(T)$) and understretched ($\hat{x} > 0$, $\gamma > \gamma^*(T)$) direct regimes. The orange dot locates the crossover point ($\gamma = \gamma^*(T)$). Inset: Numerical solutions for $\tilde{m}_t^{1,\mathrm{dir}}$ with those combinations of parameters are shown in the inset plot for $t/T \leq 0.24$. In the simulations $\beta = 6$. Image and caption taken from Mauri et al. (2023b).

The solution above holds as long as $\hat{x}$ does not hit the boundary, *i.e.* provided $\hat{x} > 0$. When $\hat{x} = 0$, using Eq.(9.23) and integrating function ξ, we find that γ has to satisfy the equation

$$\gamma = 1 + \frac{2}{\sqrt{T}} \arctan\left(\frac{\gamma\sqrt{T}}{2}\right) . \tag{9.26}$$

The root of this equation, which we denote by $\gamma^*(T)$ is plotted in Figure 9.3. We may now conclude:

- If $\gamma < \gamma^*(T)$ we have $\hat{x} = 0$: the projection $\tilde{m}(\tau)$ is smaller than 1 as soon as $\tau > 0$, see inset in Figure 9.3. For such small γ the paths are not flexible enough and the full 'time' T at their disposal is needed to join the anchoring edges. We call this regime *overstretched*. Notice that the boundary conditions $\eta(\hat{x} = 0) = 0$ in Eq. (9.23) can be satisfied by fixing the initial value of the function ξ, *i.e.* $\xi(0)$. In particular, we find

$$\xi(\hat{x} = 0) = 2\tan\left(\frac{\sqrt{T}(\gamma - 1)}{2}\right) . \tag{9.27}$$

- If $\gamma > \gamma^*(T)$, we have $\hat{x} > 0$. The available number of intermediate sequences along the path, T, is larger than what is actually needed to join the two edges. A fraction ($= 2\hat{x}$) of these intermediate sequences are mere copies of the initial and final configurations, see inset of Figure 9.3. We hereafter call this regime *understretched*. All the analytical results reported in Eqs. (9.24,9.25) are in excellent agreement with the numerical resolution of the self-consistent equations for the order parameters, see Figure 9.4.

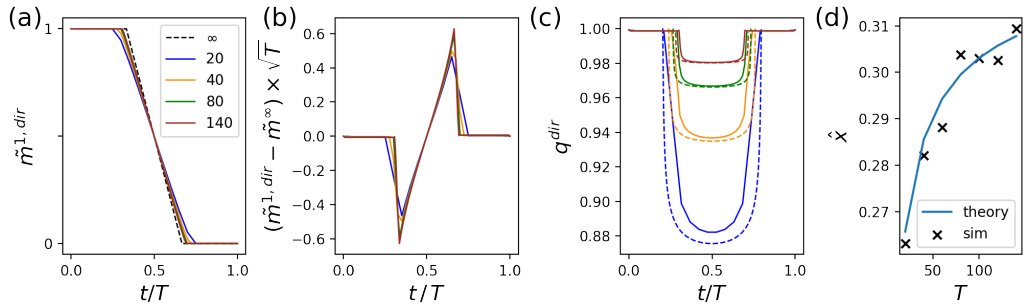


Figure 9.4: Mean-field solution of the MHP model in the understretched regime for direct paths. (a) Numerical solutions for $\tilde{m}_t^{1,\mathrm{dir}}$ for different values of T (values showed in legend) compared with the limit solution $\tilde{m}^{\infty}$ in Eq.(9.25) for $T \to \infty$ (black dashed line). (b) Scaling of the difference between $\tilde{m}_t^{1,\mathrm{dir}}$ computed numerically and $\tilde{m}_t^{\infty}$ for large T. (c) Numerical solutions for q_t^{dir} (solid lines) compared with the respective theoretical estimation (dashed lines) evaluated using $\hat{x}$ according to Eq.(9.24). (d) Numerical estimation of $\hat{x}$ (black crosses: the value corresponds to the moment $\tilde{m}_t^{1,\mathrm{dir}}$ becomes < 1) vs. theoretical scaling from Eq.(9.24) (blue line). The parameters of the simulations are $\beta = 6$, $\gamma = 3$. Image and caption taken from Mauri et al. (2023b).

9.2.3 The direct-to-global phase transition

The solution $\#_{\mathrm{dir}}$ we have derived above assumes that $\tilde{m}_3$ vanishes at all time. This assumption is correct as long as the minimum of the free energy f_{path} is located in $\tilde{m}_3 = 0$. We compute below the first derivative of the free energy along the third projection $\tilde{m}_t^3$:

$$\left.\frac{\partial f_{\mathrm{path}}}{\partial \tilde{m}_t^3}\right|_{\#_{\mathrm{dir}}} = 1 - \langle \delta_{v_t,a} + \delta_{v_t,b} + 2\omega\delta_{v_t,c} \rangle_{1\mathrm{D}}|_{\#_{\mathrm{dir}}} \,. \tag{9.28}$$

By studying the sign of this derivative we will show the existence of a critical value of ω appearing in the patterns of the HP model, see Eq. (9.5). This critical value, hereafter denoted by ω_c, separating a regime where the direct solution is stable ($\omega < \omega_c$) and a regime where it is not and the true mean-field solution is global ($\omega > \omega_c$).

Two classes of competing configurations must be considered: the direct (*dir*) ones, which start in $v^{\mathrm{start}} = a$ and turn into $v^{\mathrm{end}} = b$ at some time $\tau \in (\hat{x}, 1-\hat{x})$.; the global (*glob*) ones, which start in a then change to c at some time $\tau \equiv x \in (0, 1/2)$, then turn into b when $\tau = 1 - x$. We estimate below the energies E_{dir} and E_{glob} corresponding to the two scenarios. In particular, when $E_{\mathrm{dir}} < E_{\mathrm{glob}}$, the direct configurations dominate the average on the right hand side of Eq. (9.28), leading to

$$\left.\frac{\partial f_{\mathrm{path}}}{\partial \tilde{m}_t^3}\right|_{\#_{\mathrm{dir}}} = 0 \;\forall\; t \,. \tag{9.29}$$

Conversely, when $E_{\mathrm{dir}} > E_{\mathrm{glob}}$, we will have

$$\left.\frac{\partial f_{\mathrm{path}}}{\partial \tilde{m}_t^3}\right|_{\#_{\mathrm{dir}}} = 1 - 2\omega \text{ for } t \in (x, 1-x)\,,\ 0 \text{ otherwise.} \tag{9.30}$$

Hence, the direct solution will be unstable if, in addition, $\omega > \frac{1}{2}$. As we shall check explicitly below this condition is always met when $E_{\mathrm{dir}} > E_{\mathrm{glob}}$.

9.2.3.1 Understretched regime

The energy of the direct configurations (for $T \gg 1$) is given by:

$$E_{\mathrm{dir}} = -T\left(\hat{x} + \frac{1}{2}\right) + \frac{1}{\gamma^2}\,, \tag{9.31}$$

while the global ones have energy

$$E_{\rm glob}(x) = \begin{cases} -T\,(2x+\omega(1-2x)) + \dfrac{2}{\gamma^2} & \text{for } x \le \hat{x} \\ -T\Big(2\hat{x} + 2\displaystyle\int_{\hat{x}}^{x} {\rm d}y\,(1-\frac{y-\hat{x}}{1-2\hat{x}}) + \\ +\omega(1-2x) - \dfrac{2}{|\xi(x)|^2}\Big) & \text{for } x \in (\hat{x}, 1/2) \end{cases} \tag{9.32}$$

which is minimal for $x = \hat{x}$ when $\omega \in (1/4, 1)$ and for $x = 0$ when $\omega > 1$. Here the condition $E_{\rm glob} < E_{\rm dir}$ provides the critical value of ω for the phase transition:

$$\omega_c^{\rm under}(\gamma, T) = \frac{1}{2} + \frac{1}{T\gamma^2(1-2\hat{x})} \simeq \frac{1}{2} + \frac{1}{T\gamma} \tag{9.33}$$

for large T.

9.2.3.2 Overstretched regime

In the overstretched case, the energy of the direct configurations is given by

$$E_{\rm dir} = -\frac{T}{2} + \frac{T}{\xi(0)^2}\,, \tag{9.34}$$

while the global configurations correspond to energy

$$E_{\rm glob} = -T\omega + \frac{2T}{\xi(0)^2}\,. \tag{9.35}$$

Here, $\xi(0)$ is given by Eq.(9.27). The condition $E_{\rm glob} < E_{\rm dir}$ leads to a new critical value for ω:

$$\omega_c^{\rm over}(\gamma, T) = \frac{1}{2} + \frac{1}{4\tan^2(\sqrt{T}(\gamma-1)/2)}\,. \tag{9.36}$$

9.2.3.3 Comparison with numerics

Putting together the two regimes studied above, we find that the transition takes place at

$$\omega_c(\gamma, T) = \begin{cases} \omega_c^{\rm over}(\gamma, T) & \text{for } \gamma < \gamma^*(T) \\ \omega_c^{\rm under}(\gamma, T) & \text{for } \gamma > \gamma^*(T) \end{cases}\,. \tag{9.37}$$

The phase diagram in the (ω, T) plane is shown in Figure 9.5 for different values of the flexibility parameter γ.

While the transition formally takes place in the limit $\beta \times T \to \infty$, a cross-over is observed for finite T and β. We show in Figure 9.6 the coincidence of the average log-likelihoods of intermediate sequences along direct and global paths at large T for small ω, and the higher quality of global paths for large ω. Notice that these results are valid when T is sent to large values while keeping β fixed. If β is small, e.g. of the order of $\frac{1}{T}$, the domination of global paths on direct paths is due to the larger entropy of the former. Figure 9.6 shows that, for small $\beta \times T$, global paths are indeed of lesser quality (probability) than their direct counterparts, even at high ω.

To better distinguish global from direct paths, we introduce the distance

$$d_{\rm DS}(\mathbf{v}) \quad = \quad \frac{1}{N}\sum_i (1-\delta_{(v_{\rm start})_i, v_i})(1-\delta_{(v_{\rm end})_i, v_i})\,. \tag{9.38}$$

By definition, $d_{\rm DS}$ vanishes if the configuration is within the direct subspace, and is strictly positive otherwise. Its maximal value is 1. We show in Fig. 9.6(inset) the behavior of d_{DS} for two values of the flexibility parameter controlling the Cont potential, below and above the transition point.

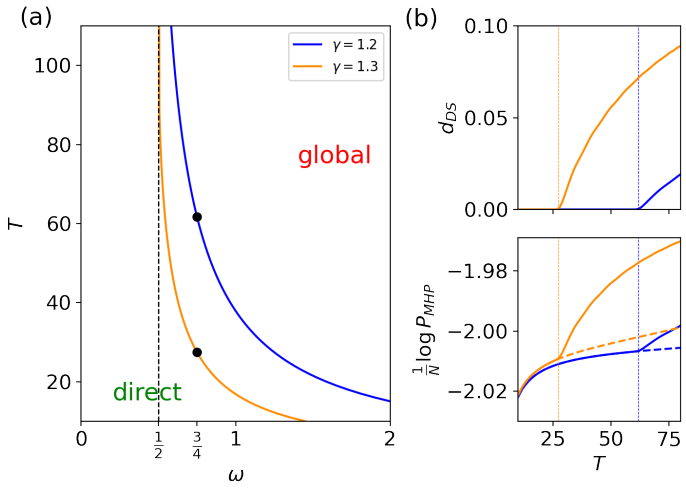


Figure 9.5: Crossover between direct and global transition paths in the MHP model. (a). Critical line $\omega_c(\gamma, T)$ vs. T for two values of γ, see Eq. (9.37). The black dots show the crossovers for $\omega = \frac{3}{4}$. (b) Distance d_{DS} to the direct space (Top) and $(\log P_{\mathrm{MHP}})/N$ averaged over intermediate sequences (Bottom; solid line: global, dashed: direct) vs. path length T; same parameters as in (a). Image and caption taken from Mauri et al. (2023b).

9.2.4 Escaping from local minimum: paths anchored at origin

We have so far considered paths anchored at both extremities. Our mean-field formalism can be extended to the case of paths in which the final configuration is not fixed. In this context the goal is to characterize the most likely behavior of a path in the energy landscape under a mutational dynamics encoded in the interaction potential Φ.

A natural question in this scenario is to estimate when and in which conditions a configuration escape from a local minimum to reach a more stable configuration. In the MHP model, we consider paths starting in $\mathbf{v}_{\mathrm{start}} = \{a, a, a, \ldots, a\}$, and unconstrained at the other extremity. The properties of these paths can be computed through Eq. (9.8), upon relaxing the condition at the extremity when computing Z_i^{1D} using transfer matrix in Eq. (9.9). Compared to what done in the rest of the chapter, we decide to study the escape problem both in Evo and Cont scenario.

To estimate the escape probability, we define the region $\mathcal{R}$ associated to the minimum of the free-energy landscape close to the initial configuration $\mathbf{m}^{\mathrm{start}} = (1, 0)$. Then, we evaluate the probability P_{stay} of remaining in that region after a certain number of steps T using Eq. (9.7). The escape probability is computed as $P_{\mathrm{escape}} = 1 - P_{\mathrm{stay}}$. In Figure 9.7 we show the estimated $\log P_{\mathrm{stay}}$ in the Cont and Evo scenarios for different values of ω and T. When $\omega > 1/\sqrt{2}$ the local minimum in (ω, ω) depicted in Fig. 9.1 becomes global, and the path is attracted towards this minimum. For finite T higher values of ω are required to overcome the elastic constraint due to Φ to remain close to $\mathbf{v}_{\mathrm{start}}$.

9.2.5 Conclusive remarks

In this Chapter, we have unveiled the existence of a direct-to-global phase diagram controlled by the stiffness of the interacting potential (in the Cont scenario) and the total number of steps of the path, together with the inner structure of the energy landscape. We have analytically described this phase diagram for the so-called Hopfield-Potts model, with only two interaction patterns with projections outside the direct subspace of controlable amplitude. We have analytically located the direct-to-global phase transition in a low temperature/high length regime as a trade-off between long, flexible paths with low energy

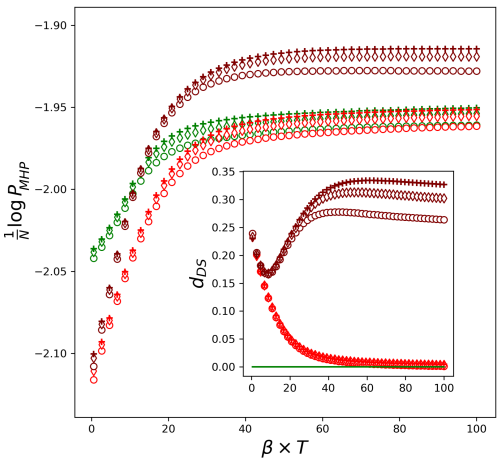


Figure 9.6: Average log-likelihood along the paths for the MHP model as a function of $\beta \times T$. Inset plot shows the average distance to direct space. Symbols stand for different T (circles for $T = 20$, diamonds for $T = 30$ and pluses for $T = 40$). Green symbols represent direct paths (which are of course independent of ω), Red symbols represent global paths with $\omega = 0.4$ and maroon symbols represent global paths for $\omega = 0.7$. Here $\gamma = 2$ (Cont potential). For high values of $\beta \times T$, we see that the global paths for $\omega < 0.5$ converge towards the direct ones, while, for $\omega > 0.5$, the two classes of paths remain separated, in agreement with the phase transition shown in Fig. 9.5. Image and caption taken from Mauri et al. (2023b).

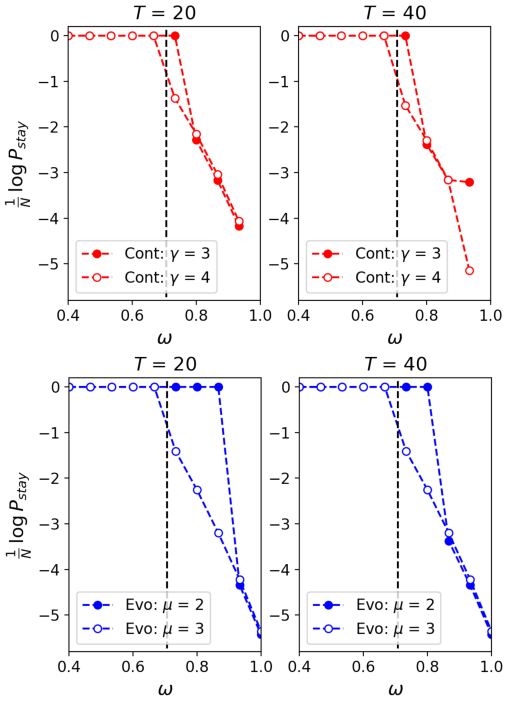


Figure 9.7: Probability of stay in the initial local minimum close to the configuration $\mathbf{m}^{\text{start}} = (1,0)$ after $T = 20$ (*left*) and $T = 40$ (*right*) steps. the probabilities are plotted against different values of the overlap ω. Cont (*up*) and Evo (*bottom*) scenario are respectively plotted in red and blue. The dark dashed line correspond to $\omega = 1/\sqrt{2}$, when the minimum at $\mathbf{m} = (\omega, \omega)$ becomes global. Here $\beta = 6$. Image and caption taken from Mauri et al. (2023b).

intermediate configurations and short, stiff paths minimizing the number of mutations to go from one sequence to another. In this low temperature regime, the direct-to-global transition is essentially not affected by the number A of Potts states (colors). More precisely, in the absence of strong thermal fluctuations, due to the low temperature, all paths will tend to minimize the energy and, as a consequence, each residue along the path will only occupy one of the three states a, b, c that are relevant in the MHP landscape (see Eq. 9.5). Conversely, in the high temperature regime, that is, if the fluctuations of the energy are smaller than, or comparable to the inverse of the path length, paths tend to be global due to thermal fluctuations and the entropy of the system will depend on the total number of accessible state A per site.

This direct-to-global phase transition is properly defined when conditioning on the final extremity of the paths. While evolutionary paths are generally not constrained in this way, there exist relevant situations in which conditioning is important. For instance, consider a directed evolution experiment starting from a wild-type sequence (of DNA, RNA, protein). Samples of the pool of sequences are retained at each round of selections/mutations. After several rounds, a sequence is obtained, and one asks for the possible transition paths that led to this outcome from the wild type. This well-posed question can be addressed with the methods proposed in this thesis, and confronted to sequences sampled at intermediate

rounds. In addition, irrespective of conditioning at the end of the path, we have shown that the direct-to-global transition is intimately related to the presence of attractive region in the energy/fitness landscape (Fig. 9.2).

From a statistical mechanics point of view, the mean-field approach followed here computes transition paths for a given realization of the quenched disorder. This is made possible by the fact that the number M of patterns in the Hopfield-Potts model (or the number of hidden units in the RBM) remains finite as $N \to \infty$. As we have already seen in Chapter 8, for RBM models trained on real protein families, the number of hidden units M is usually of the same order as N. A natural extension of this theoretical work would involve studying direct-to-global phase transitions in models with an extensive number of patterns, whose weights are random but with controlled overlaps. Such models could be explored in the compositional phase, which we briefly discussed at the beginning of Section 8.2.

IV

Experimental validation of mutational paths

Summary

The final objective concerning the study of mutational paths presented in Part III is to experimentally test the sampled intermediate sequences to validate the quality of the entire paths. In this last part of the thesis, we will present experimental results obtained by our collaborators, which show promising outcomes.

Chapter 10 focuses on the experimental validation of mutational paths in WW domains (previously studied in Chapter 7 and 8) through *in vitro* experiments. This research project is conducted in collaboration with Professor Marco Ribezzi-Crivellari and Ahmed Rehan from ESPCI in Paris, who are developing the experimental set-up for measuring the binding affinity of designed WW domains, which are extracted from sampled paths, to different peptides and compare them with natural sequences.

Chapter 11 centers on the experimental validation of mutational paths in proteases. Our primary interest lies in studying the transition between two classes of proteases: trypsin and chymotrypsin. The trypsin-chymotrypsin transition is a significant challenge in structural biology, and identifying sequences with promiscuous activity is considered extremely difficult. In collaboration with Professor Clement Nizak, Professor Olivier Rivoire, and Amaury Paveyranne from Sorbonne University in Paris, we aim to experimentally test proteases along mutational paths between trypsins and chymotrypsins using a microfluidic experiment of their design. The experimental set-up is currently under development, and we expect the first results in the coming months. In this chapter, we will present preliminary results obtained with a simpler experimental set-up.

10

in vitro validation of WW domains

In Part III, we explored the sampling of mutational paths between two homologous proteins using sequence-based models trained on multiple sequence alignments, particularly utilizing Restricted Boltzmann Machines. This approach is justified by the ability of these models to encode relevant information about the real fitness landscape that guided the evolution of the protein family, thus enabling the reconstruction of evolutionary trajectories connecting different homologs to their common ancestor. Although we have extensively benchmarked the quality of the paths using various numerical methods, direct experimental validation of the paths remains lacking. This serves as the primary objective of the final part of this thesis.

Specifically, this chapter focuses on the experimental validation of paths in WW domains, which we have already discussed and numerically validated in Chapter 7. We will begin by describing the experimental setup for the *in vitro* characterization of the WW domains, which has been developed by Professor Marco Ribezzi-Crivellari and Ahmed Rehan from ESPCI in Paris. Subsequently, we will present the paths that we intend to experimentally test, along with preliminary experimental results.

One of the noteworthy outcomes from this preliminary experiment is the demonstration of promiscuous activity along paths connecting WW domains of class I to class IV, consistent with our previous predictions from the numerical analysis in Section 7.3.3.

10.1. Experimental setup

For each WW domain, a poly-nucleotide molecule has been ordered. This molecule contains the DNA sequence corresponding to the WW domain, along with a promoter and a ribosome-binding site (RBS) upstream which is necessary to start the replication process with PCR. Downstream of the WW domain, a sequence encoding a linker and a SNAPtag has been added, while a stop codon has been put at the end of the molecule. The SNAPtag is a protein that can be covalently linked to a fluorescent molecule [Kolberg et al. (2013)]. A scheme of this molecule is given in Figure 10.1.

In parallel to the poly-nucleotide molecule, the substrate containing the specific peptide

Figure 10.1: Scheme of the poly-nucleotide molecule used for the experimental validation of mutational paths in WW domains. Image courtesy of Marco Ribezzi-Crivellari and Ahmed Rehan.

Class	Sequence
I	GESPPPPYSRYPMD
II	APPTPPPLPPD
IV	EQPLpTPVTDL

Table 10.1: Peptides used for the experimental validation of mutational paths in WW domains. Highlighted in red are residues that bind to the WW domain. The phosphorylated Threonine in the peptide of class IV is indicated by pT.

to which the WW is suppose to bind is prepared. More precisely, it consists of a metallic bead containing on the surface many copies of a protein containing the peptide of interest plus a fluorescent dye.

Once the protein containing the WW domain and the SNAPtag is prepared it is presented to the substrate (at a given concentration). If the WW domain binds to the peptide, the SNAPtag will be covalently linked to the fluorescent dye, and the fluorescence of the substrate will increase. The fluorescence of the substrate is measured using a microscope and used to estimate the binding affinity of the WW domain to the peptide.

Each WW domain will be tested against three peptides, each one corresponding to a different class of WW domains. The three peptides are given in Table 10.1.

We urge to point that we encountered a problem when ordering the peptides. This resulted in the peptide of class IV having a phosphorylated tyrosine (pY) instead of a threonine (pT) in the active site. Unfortunately, we realised this issue after the experiments have been completed. As a consequence, the activity against this peptide will be much lower compared to the others as we will show below. Despite that, we anticipate that some sequences still have significant response above noise levels, notably for those in the path interplating between class I and class IV.

10.2. Tested sequences

For the experimental validation of WW domains, we have sent three batches of sequences to be tested with this set-up. For the first batch, our collaborators had some problems related to the phosphorylated peptide of class IV that seemed to hydrolyze very quickly. On the contrary, results for the third batch are still pending. For this reason, the next section will only present the experimental results for the second batch of sequences.

All the sampled paths are taken using the Monte-Carlo method presented in Chapter 7 or the Mean-field theory as described in Appendix D.2. The three batches are composed as follows (see Appendix E for the full list of sequences):

- **Batch 1**: 10 sequences taken from a path connecting WW domains of class I to class IV. the parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 26$
- **Batch 2**:
  - 8 wild-type WW domains of class I.
  - 8 sequences taken from a path connecting two WW domains of class I. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 16$.
  - 8 sequences taken from a path connecting WW domains of class I to class II. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 30$.
  - 8 sequences taken from a path connecting WW domains of class I to class IV. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 21$.

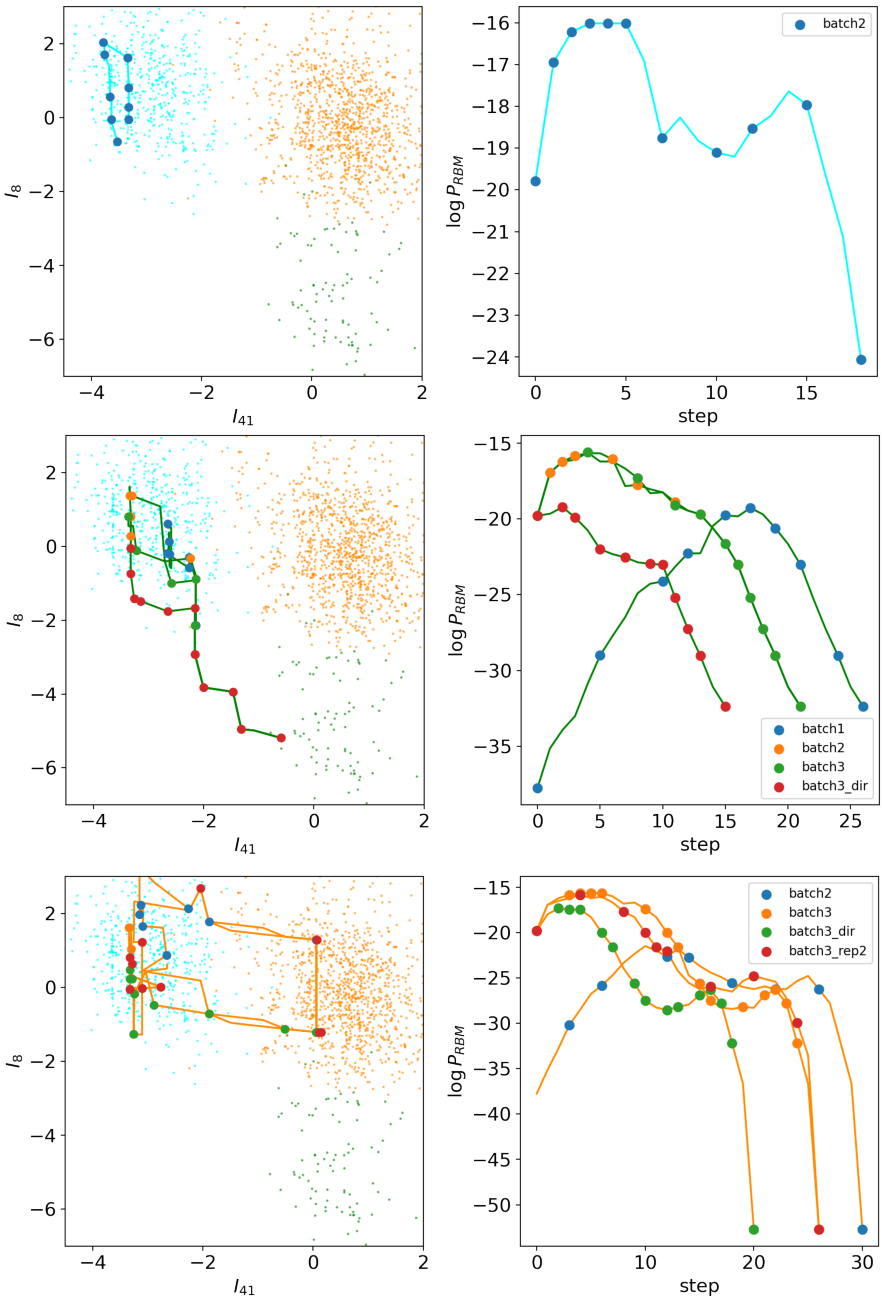


Figure 10.2: Projection of the paths from batch 2 and 3 connecting WW domains along two inputs of the trained RBM (same as Figure 7.6). Top, Center and bottom panels correspond respectively to paths from class I to class I, from class I to class II and from class I to class IV. Colored dots along the paths represent sequences that are going to be tested, with colors corresponding to their batches (see legends). The smaller points corresponds to the training data of the RBM and their color correspond to the predicted specificity class according to the local RBMs described in Section 7.3.2 (same color code of Figure 7.6).

- **Batch 3**:
  - 8 sequences taken from a path connecting WW domains of class I to class IV. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 21$.
  - 8 sequences taken from a DIRECT path connecting WW domains of class I to class IV. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 15$.
  - 16 (8+8) sequences taken from two paths connecting WW domains of class I to class II. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 26$.
  - 8 sequences taken from a DIRECT path connecting WW domains of class I to class II. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 20$.
  - 16 (8+8) sequences taken from two paths connecting WW domains of class I to class IV from the mean-field solution in Cont and Evo scenario, respectively. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 8,\ \gamma = 1,\ \mu = 2 \times 10^{-5}$.
  - 16 (8+8) sequences taken from two paths connecting WW domains of class I to class II from the mean-field solution in Cont and Evo scenario, respectively. The parameters of the path are: $\Lambda = 0.1,\ \beta = 3,\ T = 8,\ \gamma = 1,\ \mu = 2 \times 10^{-5}$.

The projection of the paths from the last two batches along the two-dimensional input space that cluster the WW domains according to their class (as in Figure 7.6) is given in Figure 10.2. We notice that all the paths from class I to class IV (Figure 10.2(*center*)) cross a region of the input space lacking of natural sequences, consistently with what presented in Section 7.3.3. Moreover, global I→IV paths from second and third batch are very similar to each other, with many identical sequences, and therefore appear overlapped in Figure 10.2(*center,right*). As a final remark, we notice the I→IV path from batch 1 and the I→II path from batch 2 start from the same sequence (the YAP1 WW domain), while the others start from a different class I wildtype. This explains the differences in the $\log P_{\mathrm{RBM}}$ profiles in Figure 10.2(*center,right*) and (*bottom,right*).

10.3. Experimental results for the 2nd batch

We now present the experimental results for the second batch of sequences, see Figures 10.3 and 10.4. The first aspect we want to discuss is related to the measurements of binding affinity in class I wild-types. We see that some of the natural wild-type do not show significant response to any peptide. This is in particular the case of the YAP1 WW domain that we studied in Chapter 7 and which was already tested in the first batch of sequences (but we did not test it in the second batch). This may be due to the fact that the binding mechanism imposed by this experiment could be largely different from the natural context in which this protein is functional. For example, many WW domains (included the YAP1) are not alone in the proteins but come in tandem with other WW domains which play an important role in promoting the binding with the peptide.

The path from class I to class II show low activity for the initial sequences (which are supposed to be of class I). This is consistent with what said above since the starting sequence of this path is exactly the YAP1 WW domain[1]. On the contrary the final sequences along this path shows good activity, as expected, against the peptide of class II. This suggests that, in general, paths show good quality only if the edge sequences show

[1]Notice that the YAP1 WW domain has not been tested in the second batch. In fact, for each path we only tested a subsample of sequences. For the case of the path I→II, we only tested the sequences at poistion $t = 3, 6, 12, 14, 18, 22, 26, 30$ (the last one corresponding to the wildtype of class II) as shown in Figures 10.3 and 10.4

good activity. In the third batch we aim to test another path from class I to class II starting from a wild type that shows significant activity in this experimetnal setup.

The measurements against the class IV peptide show a much smaller fluorescent intensity compared to all the other sequences as we anticipated at the end of Section 10.1. Despite that, both edge sequences in the I→IV path show good activity against their respective peptide. Moreover, the sequences along the path show also significant and promiscuous activity, consistently with our argument given in Section 7.3.3 (see Figure 10.5).

As written in the caption, Figure 10.4 report the same experiment of Figure 10.3, but with double the peptide concentration. As expected, the responses are overall stronger than in Figure 10.3. Despite that, the two measurements are not linearly correlated, with sequences that showed weaker responses in the first experiment being much more active than others in the second experiment. As an example, sequences at position $t = 26$ and $t = 30$ in the 1to2 path showed only strong activity against the class II peptide in the first experiment, but when the concentration is doubled they show also strong activity against the class I peptide (and class IV peptide for sequence at $t = 26$). On the contrary the class I wildtype #12, who showed the stronger activity in the first experiment, shows a weaker response compared to other proteins in the second experiment.

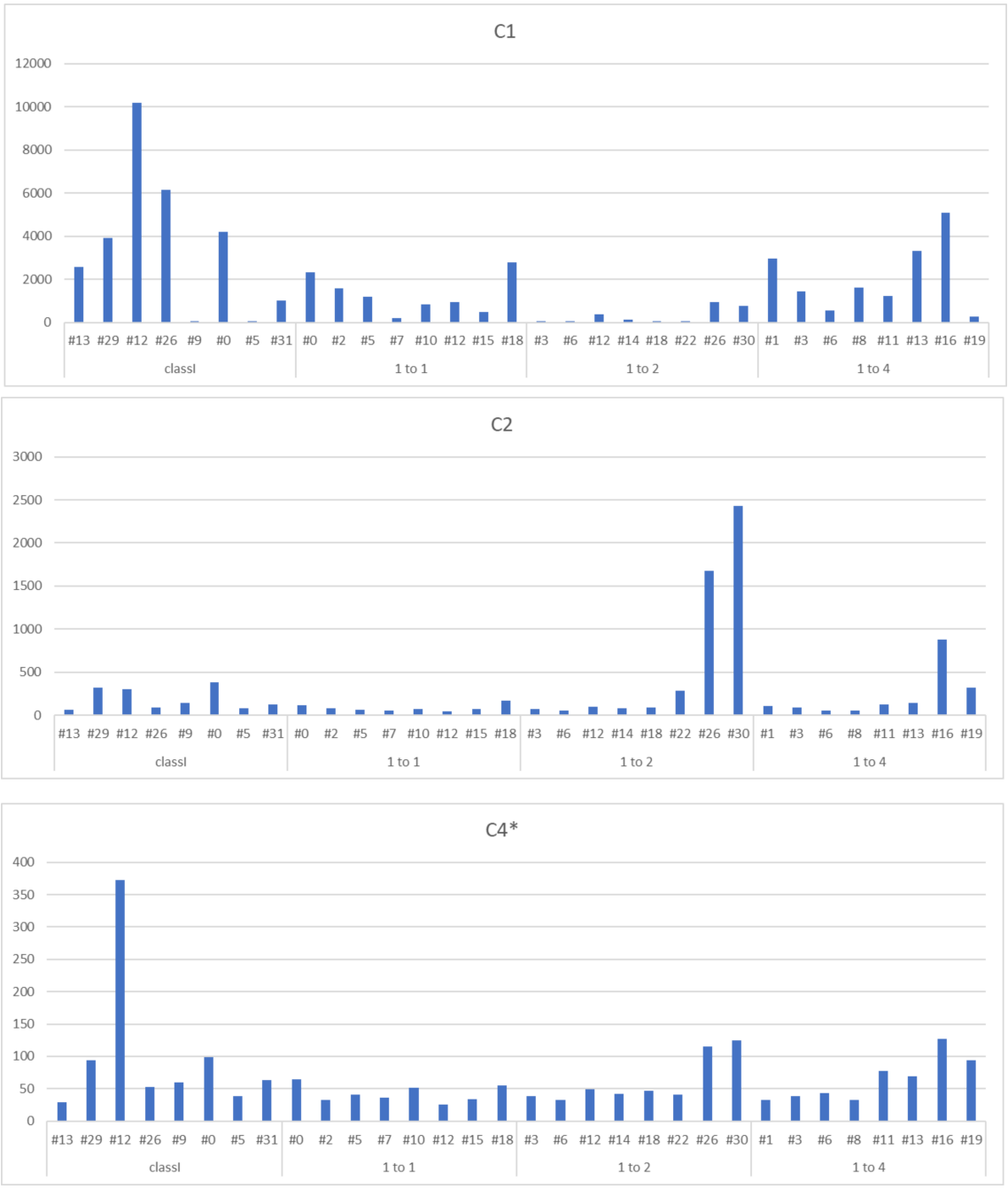


Figure 10.3: Experimental results for the second batch of sequences. The x-axis each panel corresponds to the tested sequences: (classI) wildtypes of class I, (1to1) path from class I to class I, (1to2) path from class I to class II, (1to4) path from class I to class IV. In particular for the 1to1, 1to2 and 1to4 paths, the x-axis labels correspond to the position of the sequence along the path. The y-axis corresponds to the binding affinity of the WW domain to each peptide (measured as fluorescence intensity): C1, C2, C4* panels correspond to measured activity against class I, class II and class IV peptides, respectively. Image courtesy of Marco Ribezzi-Crivellari and Ahmed Rehan.

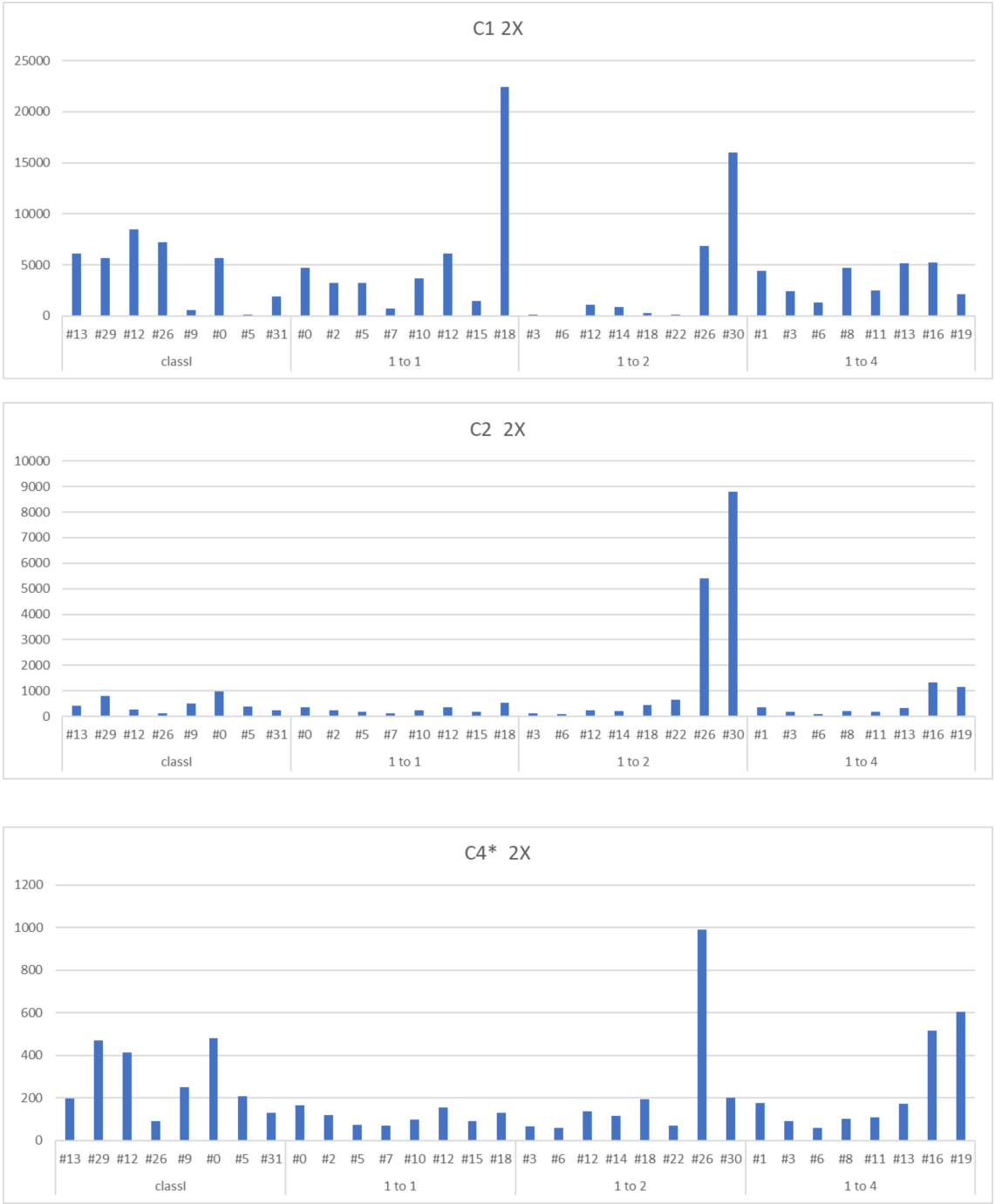


Figure 10.4: Experimental results for the same sequences as in Figure 10.3, but with a concentration of peptide doubled. Image courtesy of Marco Ribezzi-Crivellari and Ahmed Rehan.

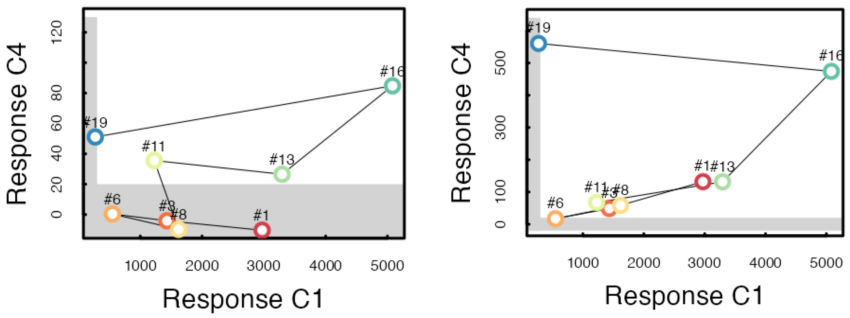


Figure 10.5: (*Left*) Response activity (measured as fluorescence intensity) against class I (x-axis) and class IV (y-axis) for the I→IV path in the second batch. Horizontal and vertical grey regions correspond to non significant response for class I and class IV peptide, respectively. (*Right*) same as left panel but for the case where the concentration of peptide is doubled. Image courtesy of Marco Ribezzi-Crivellari and Ahmed Rehan.

11

Mutational paths in proteases

Up to now, we have studied the capabilities of Restricted Boltzmann Machines to design sequences and paths of sequences in relatively short protein families ($N = 27$ for Lattice Proteins, $N = 31$ for WW domains). Another important goal would be to characterize, both numerically and experimentally, the performance of our path sampling model described in Part III for longer proteins where the inference of a reliable landscape from sequence data is much harder.

In particular, we will consider the case of serine proteases, a large family of enzymes that catalyze the hydrolysis of peptide bonds. Proteases are involved in many biological processes such as digestion, blood coagulation, apoptosis, and many others. Our goal will be to design paths of sequences between proteases belonging to two different subfamilies called Trypsins and Chymotrypsins, which is considered a major challenge in structural biology and protein design [Hedstrom et al. (1994); Halabi et al. (2009)].

After a brief introduction to the family of serine proteases and the Trypsin/Chymotrypsin subfamilies, we will describe how we trained an RBM on a multiple sequence alignment of proteases and discuss the performance of the model for sampling Trypsin/Chymotrypsin transition paths. We will then present the experimental set-up, based on a microfluid device developed by Clement Nizak's group at ESPCI, that is being used to test the designed sequences. Since we still do not have definitive measurements with this set-up, we will present preliminary results based on the catalytic activity of a few designed sequences in a standard biochemical assay. Finally, we will discuss the results and the perspectives of this work.

11.1. Introduction to serine proteases

As mentioned earlier, serine proteases are enzymes (of ∼200-300 amino acids) whose biological function is to break peptide bonds of other proteins and play an important role in digestion and immune response [Hedstrom (2002)]. The catalytic triad is the main player in the catalytic activity of serine proteases. It consists of three amino acids (Serine 195, Histidine 57, and Aspartate 102) that are spatially close in the protein structure and highly conserved in all superfamilies of serine proteases. Serine 195, in particular, performs the nucleophilic attack on the peptide bond to be cleaved, giving the name to the entire class of enzymes.

Serine proteases share a distinctive structure composed of two β-barrel domains that converge at the catalytic active site (Figure 11.1(B)). Moreover, they can be categorized based on their substrate specificity, i.e., the type of amino acid to which they prefer to bind to perform the cleavage. Two of these categories are Trypsin-like and Chymotrypsin-like proteases. Despite having similar sequences and structures, they possess different substrate

A

Enzyme	k_{cat}/K_m ($M^{-1}\,s^{-1}$)				
	Phe	Tyr	Trp	Leu	Lys
Ch	1.6×10^6	4.5×10^6	6.8×10^6	1.2×10^5	850
Tr	4.5	1.8	0.2	0.2	1.2×10^6

B

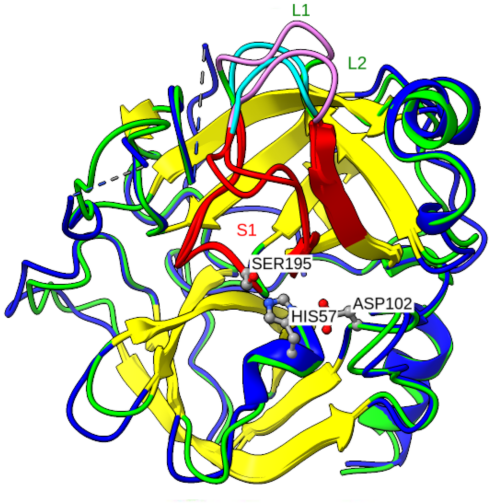


Figure 11.1: (A) Specificity values for bovine chymotrypsin (Ch) and rat trypsin (Tr) from Hedstrom et al. (1994), measured in terms of k_{cat}/K_m from Michaelis–Menten catalytic reaction mechanism (high/low k_{cat}/K_m corresponds to high/low affinity) whose meaning is explaind in Appendix F. Colored circles indicate the enzyme which prefers to bind to each amino acid (green=trypsin, blue=chymotrypsin). N.B.: the catalytic activity against Arg is not reported in this table. (B) Superposition of trypsin (green) and chymotrypsin (blue) structures. The S1 pocket appears in red, the β-barrels are in yellow, while the loops of trypsin and chymotrypsin are respectively in purple and cyan. Active site residues of trypsin are shown in ball and stick.

specificities: Chymotrypsins favor hydrophobic amino acids, such as phenylalanine, tyrosine, tryptophan, and leucine [Appel (1986)]; Trypsins prefer positively charged amino acids (e.g., arginine, lysine) [Olsen et al. (2004)]. Refer to Figure 11.1(A) for details.

In the chymotrypsin index, which is the canonical numbering of serine proteases based on that of bovine chymotrypsin, His-57, Asp-102, and Ser-195 constitute the catalytic triad (as mentioned earlier), while residues 189–195, 214–220, and 225–228 form the binding pocket (called S1). The two surface loops near the S1 pocket, sites 185–188 and 221–224, are respectively called L1 and L2 loops (Figure 11.1(B)) and were described by Hedstrom et al. (1992) to be important in the catalytic activity, see below. The binding pockets of trypsin and chymotrypsin notably differ at site 189, where trypsin has a negatively charged Asp (D) and chymotrypsin has a polar Ser (S). In particular, Asp helps bind positively charged amino acids in trypsin-like S1, while Ser helps accommodate large hydrophobic amino acids in Chymotrypsins.

Modifying the specificity of a protease has proven to be a very challenging task since the preference of substrate between trypsins and chymotrypsins is very strong (10^4-fold). Redesigning only the binding pocket by mutating residue 189 to change the specificity of a trypsin to that of chymotrypsin resulted in the loss of catalytic function [Craik et al. (1985)]. However, Hedstrom et al. (1992, 1994) demonstrated that replacing both the binding pocket S1 and the surface loops, L1 and L2, of trypsin with those of chymotrypsin, the new mutant enzyme showed chymotrypsin-like activity. This result demonstrated the importance of allostery, which defines the correlation between sites in contact with the

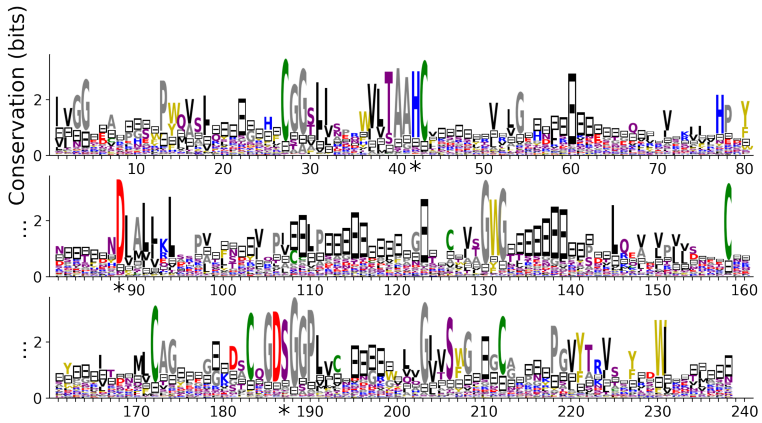


Figure 11.2: Sequence logo of the alignment of serine proteases (see Section 2.1 for definition). In this alignment, the catalytic triad (Ser 195, His 57, Asp 102) is at positions 186, 43, and 88, respectively (see asterisks), while the residue 189 in the standard reference is at position 181.

substrate and those that are far away, in this kind of activity transition. The opposite transition (i.e., changing the specificity of a chymotrypsin to that of a trypsin) was also challenging. A successful effort was conducted by Jelinek et al. (2004), showing that the mutations S189D and A226G in rat chymotrypsin B resulted in a trypsin-like activity. Finally, efforts to modify the specificity of trypsin-like proteases (increasing or decreasing its affinity for arginine or lysine) resulted in a loss of catalytic activity [Evnin et al. (1990)].

In Halabi et al. (2009), sectors, which are large groups of independently coevolving residues with specific biological functions, have been identified using Principal Component analysis on an appropriately reweighted pairwise Correlation matrix of serine protease alignments. This approach helped reveal long-range interactions between protein sites. Similarly, our goal is to utilize the trained RBM to identify groups of residues with essential roles in catalytic activity and specificity (by analyzing the weight matrix, as described in Section 2.3.4). We will use this information to characterize the optimal transition path using our algorithm presented in Chapter 7.

11.2. Training an RBM on serine proteases

To train an RBM, we need to construct a dataset of sequences that represents the family of serine proteases from the Pfam database (identified by ID PF00089). For this purpose, we will utilize the alignment obtained in Leah Friedman's Master's thesis [Friedman and Cocco (2021)]. The process of building this alignment is as follows: starting from the alignment of Halabi et al. (2009), which contains 1390 manually corrected and annotated serine proteases, sequences corresponding to snake venom serine proteases (due to their unusual activity) have been removed. Subsequently, the MAFFT algorithm [Katoh et al. (2002)] is applied to add new sequences extracted from Pfam to the existing alignment.

As a result, we obtain an alignment with 36,468 sequences, but many of the columns have a significant number of gaps (more than 50% of columns are almost entirely gapped). Since RBMs do not model gaps well, Friedman decided to retain only columns in the alignment that correspond to wild-type rat anionic trypsin 3TGI sites and sites where the average presence of the gap symbol was less than 50 percent. Additionally, the sequence coding for the propeptide region, which is cleaved during maturation, was removed.

To have a model working well also for the Chymotrypsins, we decided to retain also all the columns in the alignment that correspond to wild-type bovine chymotrypsin 1T8O. This

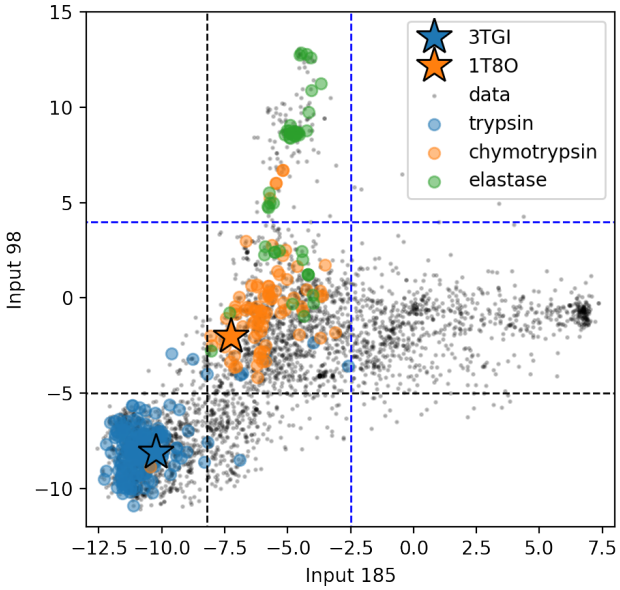


Figure 11.3: Projection along the 2D input space that better separates labeled chymotrypsin and trypsin sequences. The black and blue dashed lines delimit the regions used to group natural MSA sequences into the trypsin-specific or chymotrypsin-specific subfamilies.

decision was made to help the trained RBM better describe both Trypsin and Chymotrypsin subfamilies. As a result, the final alignment contains 238 columns. The sequence logo of such alignment is given in Figure 11.2.

This multiple sequence alignment (MSA) is utilized to train an RBM with $M = 500$ hidden units, and the hyperparameters can be found in Appendix F. Similar to the approach used in Section 7.3.3, we can cluster trypsins and chymotrypsins in the global RBM input space. Specifically, we select two hidden neurons whose inputs yield the best separation between the two classes, as tested on the labeled sequences from Halabi et al. (2009). Subsequently, we filter the full training set and create two new training sets comprising all the sequences that are closer to trypsins and chymotrypsins, respectively (see the quadrants in Figure 11.3). In particular, the dataset of chymotrypsin-like enzymes contains 11417 sequences, while the trypsin-like dataset has 11332 sequences.

Lastly, we train two local RBMs (one for trypsins and one for chymotrypsins) using these new datasets, with the number of hidden units set to $M = 100$, while keeping the other parameters the same as in the global RBM.

11.3. Tested sequences

We now describe the sequences that are in course of experimental validation. Together with paths connecting trypsin and chymotrypsin wild-type (PDB IDs: 3TGI and 1T8O, respectively) sequences, we also designed single sequences that are expected to be catalytically active. The sampling techniques are the following:

- **Path sampling.** We use the path sampling algorithm from Chapter 7 to design paths between 3TGI and 1T8O at inverse temperature $\beta = 3$.

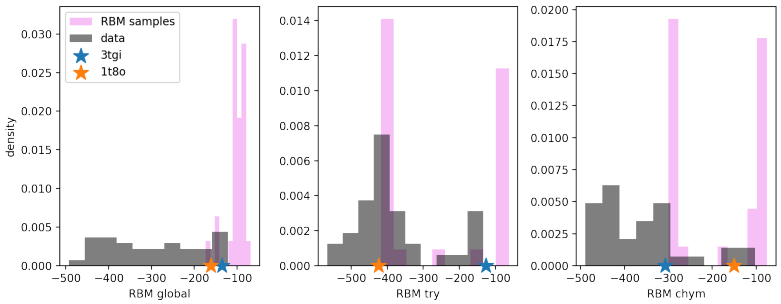


Figure 11.4: Histograms of the RBM scores of generated sequences from the first batch (excluding those from Friedman's alignment) and training data. Each panel corresponds to the scores measured with global RBM (left) and the two local RBMs (center and right). Scores of the wild-types are shown as stars on the x-axis.

- **Sampling close to wild-types.** We want to design enzymes with either trypsin or chymotrypsin specificity. To this aim, we introduce some biases $\lambda = 2$ in the local fields corresponding to wild-type amino acids and then sampling at $\beta = 3$. All the sampled sequences have between 10 and 20 mutations with respect to their specific wt. More precisely, we wirte the sampling probability, up to an additive constant, as follows:

$$\frac{1}{\beta}\log P_{\text{sampling}}(\mathbf{v}|\mathbf{v}_{\text{wt}}) = \log P_{RBM}(\mathbf{v}) + \lambda \sum_{i=1}^{N} \delta_{v_i, v_i^{\text{wt}}} \,. \tag{11.1}$$

  Figure 11.4 shows the histograms of RBM scores (energies) of generated data. The choice of λ Described above garantees a good trade-off between the number of mutations and the score of the sampled sequences. Lower values of λ would result in sequences completely different from the wildtype, while higher values would result in too few mutations.

- **Sampling Chymiotrypsin sequences with Trypsin-length and vice-versa.** We sample sequences at low temperature from an energy landscape which is given by the sum of the global RBM and the local RBM trained on chymotrypsins and adding a penalty for the gap (i.e. negative bias for gapped sites) and a bias on the amino acids present on 3tgi. More precisely, we write the sampling probability, up to an additive constant, as follows:

$$\frac{1}{\beta}\log P^{\text{chym}}_{\text{sampling}}(\mathbf{v}|\mathbf{v}_{\text{3TGI}}) = \log P_{\text{RBM}}(\mathbf{v}) + \log P^{\text{chym}}_{\text{RBM}}(\mathbf{v}) + \sum_{i=1}^{N} \left(\lambda \delta_{v_i, v_i^{\text{3TGI}}} + \lambda_{\text{gap}} \delta_{v_i, \text{gap}}\right) , \tag{11.2}$$

  where λ_{gap} is the penalty for the gap, λ is the bias on the amino acids present on 3tgi and $\log P^{\text{chym}}_{\text{RBM}}$ is the local RBM trained on chymotrypsins. We do the opposite to sample trypsins close to 1t8o (see figure 11.5). In Eq. (11.2), we could also have added a control parameter in front of the $\log P^{\text{chym}}_{\text{RBM}}$ term, but we found that increasing the strength of this term results in very poor sequences (with respect to $\log P_{\text{RBM}}$), while decreasing it make the sampling too similar to the original wildtype. In addition, also the parameters γ and λ_{gap} have been chosen heuristically to guarantee a good trade-off between the number of mutations and the score of the sampled sequences.

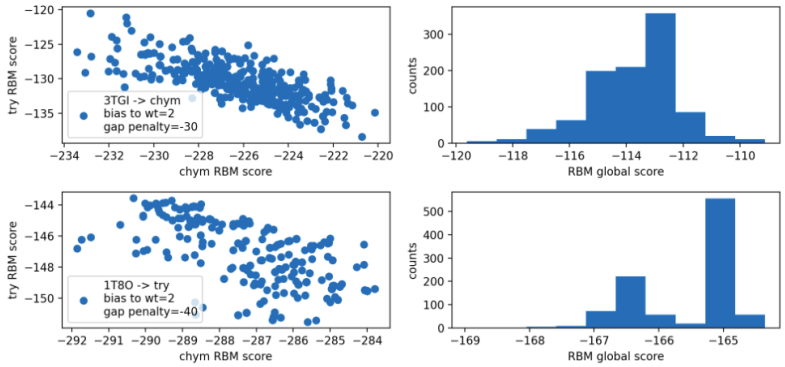


Figure 11.5: Cross-sampling chymotrypsins(trypsins) close to 3TGI(1T8O) in order to find specificity switches. Here $\beta = 2$. Left column compares scores given by the local RBM models for the sampled sequences, while right column shows the histograms with the scores of the global RBM. For both samples, we chose randomly 10 sequences to be put in the second batch.

Using these sample techniques, we prepared two batches of sequences that we aim to test in the following months. The complete list of sequences can be found in Appendix F. Here we give a brief description of the two batches:

- **First batch.**
  - 10 sequences taken from a path connecting 3TGI and 1T8O using our global RBM. parameters: $T = 430$, $\beta = 3$, $\Lambda = 1$. These sequences are labeld with PEXXXX, where XXXX is replaced with the RBM score (−log-probability).
  - 10 sequences sampled close to 3TGI using our global RBM (parameters as above). These sequences are labeled with TEXXXX, where XXXX is replaced with the RBM score (−log-probability).
  - 10 sequences sampled close to 1T8O using our global RBM (parameters as above). These sequences are labeled with CEXXXX, where XXXX is replaced with the RBM score (−log-probability).
  - 10 sequences sampled close to 3TGI using the RBM trained with the alignment from Friedman and Cocco (2021). As we stated above, this alignment only retains sites that are not gapped in the 3TGI. The training hyperparameters are the same of our global RBM and also the sampling parameters do not change. These sequences are labeled with TLXXXX, where XXXX is replaced with the RBM score (−log-probability).
- **Second batch.** These new sequences are designed starting from some of the sequences from the first batch (which are called TE85.22, PE95.13). The choice of this sequences is motivated by the preliminary results obtained by tested some of the sequences from the 1st batch and that we are going to explain in the next section.
  - 10 sequences sampled close to PE95.13 (predicted to be trypsin-like) with chymotrypsin specificity. Here, bias to wildtype is set to 2, while the penalty gap is set to -30.
  - 10 sequences sampled close to TE85.22 with chymotrypsin specificity. Parameters from Figure 11.5(top-left).
  - 10 sequences sampled close to 3TGI with chymotrypsin specificity. Parameters from Figure 11.5(top-left).

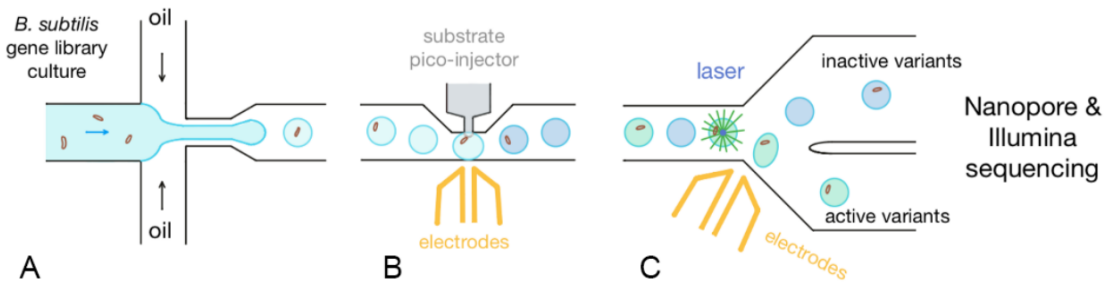


Figure 11.6: Scheme of Clément Nizak's high throughput experiment to measure the activity of enzymes. Courtesy of Valentin Senlis from Clément Nizak's lab.

- 10 sequences sampled close to 1T8O with trypsin specificity. Parameters from Figure 11.5(bottom-left).
- 30(10 x3) sequences along path 3TGI to 1T8O close to PE95.13
- 16 sequences along path 3TGI to TE85.22 ($\beta = 3$, $\Lambda = 1$)
- 11 sequences along path TE85.22 to 1T8O ($\beta = 3$, $\Lambda = 1$, penalty gap $= 2$).
- 9 sequences along the path from 3TGI to 1T8O (global) ($\beta = 3$, $\Lambda = 1$, penalty gap $= 2$).
- 8 sequences along the path from 3TGI to 1T8O (direct) ($\beta = 3$, $\Lambda = 1$, penalty gap $= 2$).

Sampling paths in proteases (where $N = 238$) is harder than sampling paths in small proteins like WW domains ($N = 31$) using only simulated annealing, as described in Chapter 7. To make the simulation faster and more efficient, we exploit the mean-field framework described in Chapter 8. We first compute the mean-field solution of a path with $T \sim 10$ steps (which does not take more than some seconds). Subsequently, we compute for each step the consensus sequence, as described in Appendix D.2. Finally, we build a discrete path by connecting all the intermediate consensus sequences with a succession of direct mutations and use this configuration as the starting point of the Monte-Carlo sampling (which take some minutes). Supposing that this inital configuration is already close enough to equilibrium, we just need one round (i.e. one temperature) of simulated annealing to obtain a good path.

11.4. Experimental set-up

In this section, we will first describe the microfluidics device that Clement Nizak's group is currently developping and from which we will receive the experimental results in the following moths. Then, we will present the simpler biochemical assay that we used to test a few sequences from the first batch.

a) Microfluidics device

The aim of this device is to measure catalytic activity of $10^3 - 10^4$ different enzymes against a range of substrates. Genes coding for the mutant enzymes are first introduced in *Bacillus subtilis* which are later encapsulated into water droplets. Here the enzyme is first expressed and then secreted by the bacteria (Figure 11.6(A)). Subsequently, fluorogenic substrates (become fluorescent when processed by an active enzyme) are injected into each droplet (Figure 11.6(B)). If the enzyme reacts with some of the substrates, the droplet fluorescence increases and it can be detected through different channels (Figure 11.6(C)).

Based on their fluorescence profile, droplets are sorted and the corresponding enzymes genes are deep-sequenced. This allows to measure the activity of many enzymes in parallel as enrichment scores between the different substrates.

b) Biochemical assay

A simpler approach to experimentally test the catalytic activity of single enzymes is through a biochemical assay. It consists of a classic enzymatic test, more precise (measuring the reaction rate in $s^{-}1$ instead of having a single time point measurement) but much less high throughput (less than 20 enzymes tested) and much more expensive in substrate (1000 times more than with the microfluidics device).

In this case, the enzyme is first expressed in vitro and is then incubated with a fluorescent substrate. The fluorescent substrate is composed by two peptides with sequence AAPx (x representing any amino acid) attached to a fluorescent dye. When the enzyme breaks both peptides, the dye becomes fluorescent. The concentration of the reaction product can be obtained by measuring the fluorescence of the solution.

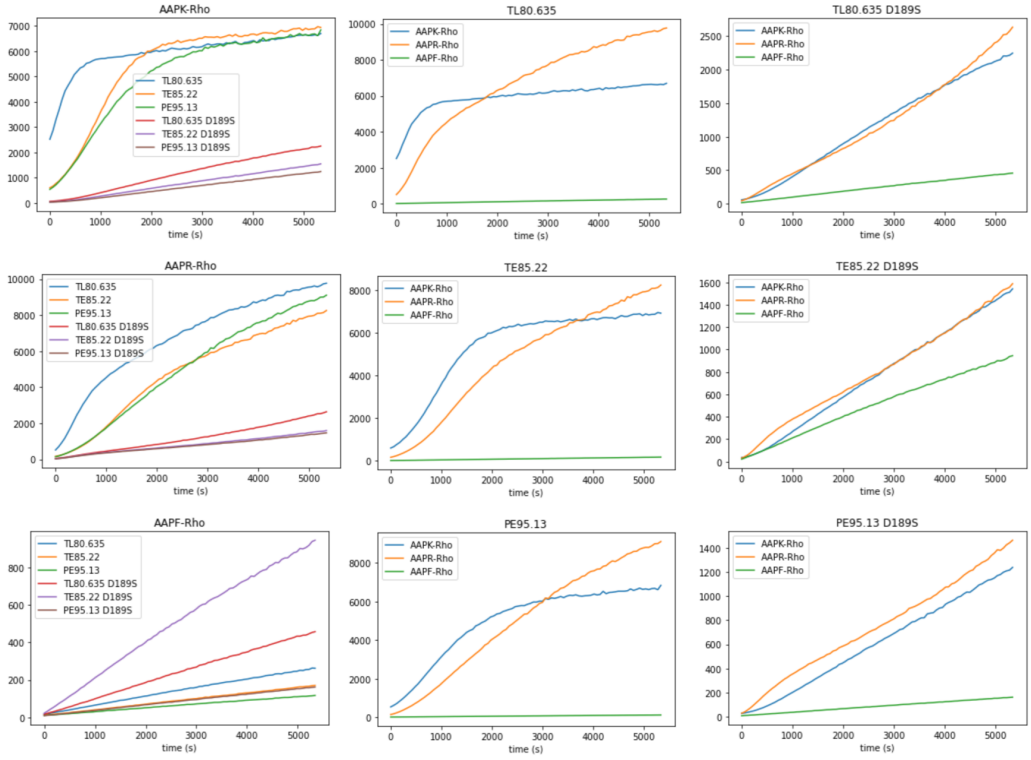
11.5. Preliminary results

We show now the preliminary results obtained through biochemical assays of the sequences TE85.22, PE95.13 and TL80.635, taken from the first batch. We tested them against three different substrates: AAPR, AAPK and AAPF. The first two substrate are specific for trypsin-like enzymes, while the third one is specific for chymotrypsin-like enzymes. The results are shown in Figure 11.7.

From Figure 11.7a and 11.7b, we see that all the three sequences are trypsin-like, showing good response to the substrates AAPR and AAPK, while low response to AAPF. To further analyse those sequences, we also tested the same enzyme where the D189 residue (using the standard positioning reference) is mutated into S, see Section 11.1. Results are shown in Figure 11.7a and 11.7c. Notably, we see that, despite the activity against AAPK and AAPR is decreased, it is not completely lost. Moreover, the activity against AAPF is increased and for the case of TE85.22(D189S) the activity against AAPF is comparable with the one against AAPK and AAPR (see Figure 11.7c(second row)). This means that TE85.22(D189S) is a promiscuous enzyme, which is able to cleave trypsin-like and chymotrypsin-like substrates without any preference.

This last point is of particular interest from the point of view of structural biology. As we stated above, Modifying directly the binding pocket (through D189S mutation) in wild-type trypsin 3TGI would result in a complete loss of activity of the enzyme. It has been consequently thought that altering the specificity profile of a protease is a very challenging task [Hedstrom et al. (1994)]. However, we showed that sampling at low temperature and close to the wild-type sequence using our RBM, we can obtain sequences that are much more tolerant to mutations done in the binding pocket. Supposing that this property is general for sequences with low predicted energy that are close to 3TGI, we might have found a simple way to design many promiscuous enzymes with relative small effort.

The goal at the moment would be to investigate more deeply the generative power of RBMs by studying paths of sequences between 3TGI and 1T8O using the microfluidics device described above. What makes this task particularly challenging, from the point of view of the RBM, is that the two wild-type sequences are very different and present many deletions and insertions (*i.e.* some sites are gapped in one sequence but not the other). This might be an issue since this kind of sequence-based models are not well suited for modelling gapped sites, as discussed in Section 3.4. We tried to mitigate this problem in



(a) (b) (c)

Figure 11.7: Experimental validation of some sequences using biochemical "in vitro" assay. (a) Each panel shows the fluorescence intensity as function of time for different sequences against a particular substrate. (b) Each panel shows the fluorescence intensity as function of a given sequence from the 1st batch against different substrates. (c) Each panel shows the fluorescence intensity as function of a given sequence from the 1st batch (with the D189S mutation) against different substrates. Image courtesy of Clement Nizak.

the two batches of sequences described in Section 11.3 by introducing a gap penalty in the RBM during the sampling procedure. However, future work will be needed to better understand how to deal with this issue.

Conclusions and perspectives

In this Ph.D. thesis, we have studied the power of sequence-based models trained on multi-sequence alignments (in particular DCA and RBM) to recover relevant information about the evolution of homologous proteins. The main assumption behind this approach is that the genotype distribution for a given protein family is related in some way to the fitness landscape driving the evolution of each individual sequence. This approach has proved fruitful in past works for different tasks such as protein structure prediction, mutagenesis experiments, and *de novo* protein design.

In this context, we have focused on two main problems:

- In which limit can important evolutionary constraints, such as epistatic interactions between pairs of distant residues, be inferred from sequence data? More generally, in which limit can the global fitness landscape be better recovered by inferring the genotype distribution of a protein family?

- How can the genotype distribution inferred from sequence data be used to understand how distant homologous proteins evolved apart from each other? In other words, we are addressing the problem of finding mutational paths between homologous proteins. This problem is interesting not only from an evolutionary point of view but also for practical applications in protein engineering. Specifically, we are asking how to consistently find successions of mutations that interpolate between proteins with different functions in such a way that the intermediate sequences are also functional.

The first question is addressed in the context of the *Quasi-linkage equilibrium* (QLE) regime of evolution, where selective pressures to fixate the fittest genotype in the population are weaker than reshuffling terms, such as mutations and/or recombinations.

In this limit, and for the case of binary genomes, we first introduced a Gaussian ansatz that simplifies the evolutionary dynamics of a population into a closed set of differential equations for the first (*i.e.*, average spin value) and second (*i.e.*, correlations between loci) order moments of the distribution. Within this approximation, we extensively studied the evolution of a population in a short-range epistatic fitness landscape, unveiling the existence of a phase transition between a non-localized phase and a phase where the population tends to align along two competitive and equally fit distant genotypes.

Secondly, we studied a numerical model for the evolution of spin-like genomes in a random epistatic fitness landscape with tunable mutation and recombination rates. We have shown that when the system reaches QLE, the genotype distribution reaches an asymptotic stationary state which can be reconstructed using DCA or Gaussian approximations. Moreover, past work has shown that in the high-recombination limit, the pairwise interactions between loci are directly related to the epistatic interactions in the fitness landscape. We have extended this result to the case where mutation and recombination strengths are comparable, leading to an updated formula to infer epistatic interactions

from sequence data. We have furthermore corroborated this analysis through extensive numerical simulations.

The main limitation of our analysis is that we only focused on a quite simplistic model for the evolution of spin-like genomes. Further work is necessary to extend it to the case of categorical data (*i.e.* Potts variables) and to benchmark it against real models of evolution. In this direction, a possible idea would be to study SELEX experiments with very high mutation rates and to compare inferred epistatic interactions with the ones measured experimentally.

Regarding the Gaussian closure scheme, it would also be interesting to extend it to the case of categorical data. Moreover, it would be necessary to understand how this approach can be extended to study the breakdown of QLE in the presence of strong selection. If in QLE, we can see the population as a unique cloud of sequences evolving together around a certain 'average' genome, we could similarly see the Clonal Condensation phase as a sum of competing clouds (as already described in Section 5.4). A possible working hypothesis would be to consider the shape of each cloud as something evolving independently from the others, while the height of each peak (*i.e.*, the abundance of each clone) is determined by the competition between the different clouds.

To answer the second question, we first developed a Monte-Carlo algorithm to sample mutational paths between two homologous proteins. The goal of this procedure is to maximize the probability of all the intermediate sequences along the path, given a certain model probability distribution inferred from the data. Using RBMs trained on Lattice Proteins and WW domain families, we have shown that this algorithm is able to find good mutational paths, with intermediate sequences sometimes better than the wild-types used as edge sequences. For the case of WW domains, we used the algorithm to design paths interpolating between different sub-families with different binding specificities. In one particular case, such a transition path crosses a region lacking natural sequences, suggesting the presence of an ancestral promiscuous region of proteins that disappeared during evolution. We benchmarked our algorithm using state-of-the-art numerical approaches based on structural information, such as ProteinMPNN and AlphaFold.

To further exploit the interpretative character of RBMs, we developed a mean-field theory that characterizes paths through the overlap between neighboring sequences and the projection of each sequence in the representation space defined by the input to each hidden unit. Depending on the interaction potential between adjacent sequences, different mutational dynamics can be modeled: one acting to limit the total number of mutations along the path (Cont scenario), and one replicating a mutational dynamics of an infinite population diffusing in a fitness landscape at a given mutation rate (Evo scenario). We showed that this approach opens up the possibility to compute many statistics of interest for paths with both ends fixed and paths with only one end fixed. We applied our mean-field framework to the case of LP and WW domains. Moreover, for a minimal Hopfield-Potts model, we unveiled the existence of a transition between a phase where paths are constrained in the space defined by 2^D mutants containing the amino acids appearing in the two edge sequences (differing on D sites), called *direct* paths, and an unconstrained phase where paths can freely explore the sequence space in order to minimize the energy, called *global* paths. At high temperature, this transition is controlled by the stiffness of the interacting potential, the length of the path, and the inner structure of the energy landscape, and does not depend on the number of categorical variables accessible at each site.

Finally, we presented the experimental set-ups developed by Marco Ribezzi's and Clement Nizak's groups. The first aims to test mutational paths in WW domains, while the second focuses on serine proteases, with the final goal of characterizing the transition

between trypsin-like and chymotrypsin-like proteases. Both experiments are still ongoing, but promising preliminary results have been presented.

The part of the research project focused on transition paths is clearly the one that reached the most advanced stage, with final experimental validation expected in a few months. However, different issues might necessitate further investigation in future works.

First, as previously discussed, energy-based models such as RBMs depend on the alignment in the training set. If the alignment contains too many gaps, this might disrupt the generative performances of the RBM since it treats gaps as any other amino acid. To address this issue, it would be useful to develop models similar to RBMs in terms of interpretability but that do not depend on a specific alignment of the training data. In particular, we have discussed in Chapter 3 about possible alignment-free models based on deep learning (and in particular on language models) that reached promising results Seo et al. (2018); Weißenow et al. (2022).

Second, as discussed at the end of Chapter 9, it would be useful to introduce a modified mean-field theory that accounts for the number of patterns in the RBM scaling with the number of sites. This would allow us to have a more realistic approximation scheme, which can be used to have a deeper and more precise characterization of transition paths.

Finally, we have discussed how mean-field theory can be used to compute the transition probability between two sequences under a given mutational dynamics. This application could be valuable in the context of phylogeny. Specifically, the transition probability could serve as an estimation of the evolutionary distance between two sequences, enabling us to construct phylogenetic trees using distance-based methods. This would likely be the most significant outcome of our mean-field framework, as it allows us to tackle the problem of phylogeny reconstruction using complex evolutionary models that consider potentially strong epistatic interactions between sites. However, the main challenge lies in making this approach feasible, which requires developing a more efficient mean-field theory capable of evaluating in parallel many transition probabilities between different pairs of sequences.

V

Appendix

A

Appendix to Chapter 5

1.1. Derivation of the Gaussian closure equations

Below we present the extensive calculation in the Gaussian closure scheme to obtain Eqs. (5.6) and (5.7) according to our original work [Mauri et al. (2021)]. Here, we report the calculations as replicated by Dichio and collaborators in Refs. [Dichio (2020); Dichio et al. (2023)].

A.1.1 Derivation of Eq.(5.6)

Consider the dynamics of the first cumulants, Eq.(4.15). $\forall i \in 1,\dots,L$

$$
\begin{aligned}
\dot{\chi}_i &\overset{(a)}{=} \sum_j f_j\chi_{ij} + \sum_{j\neq i} f_{ij}\chi_j + \sum_{\substack{j<k\\ j,k\neq i}} f_{jk}[\chi_i(\chi_j\chi_k+\chi_{jk})+\chi_j\chi_{ik}+\chi_k\chi_{ij}] + \\
&\quad - \sum_{j<k} f_{jk}\chi_i(\chi_{jk}+\chi_j\chi_k) - 2\mu\chi_i \\
&= \sum_j f_j\chi_{ij} + \sum_{j\neq i} f_{ij}\chi_j - \sum_{j\neq i} f_{ij}\chi_i(\chi_{ij}+\chi_i\chi_j) + \sum_{\substack{j<k\\ j,k\neq i}} f_{jk}(\chi_j\chi_{ik}+\chi_k\chi_{ij}) - 2\mu\chi_i \\
&= \sum_j f_j\chi_{ij} - \sum_{j\neq i} f_{ij}\chi_i\chi_{ij} + \sum_{j\neq i} f_{ij}\chi_j(1-\chi_i^2) + \sum_{\substack{j\neq k\\ j,k\neq i}} f_{jk}\chi_j\chi_{ik} - 2\mu\chi_i \\
&\overset{(b)}{=} \sum_j f_j\chi_{ij} - \sum_{j\neq i} f_{ij}\chi_i\chi_{ij} + \sum_{j\neq i} f_{ij}\chi_j\chi_{ii} + \sum_{j\neq i}\sum_{\substack{k\neq i\\ k\neq j}} f_{jk}\chi_j\chi_{ik} - 2\mu\chi_i \\
&\overset{(c)}{=} \sum_j f_j\chi_{ij} - \sum_{j\neq i} f_{ij}\chi_i\chi_{ij} + \sum_{j\neq i}\sum_{k\neq j} f_{jk}\chi_j\chi_{ik} \pm \sum_{k\neq i} f_{ik}\chi_i\chi_{ik} - 2\mu\chi_i \\
&= \sum_j f_j\chi_{ij} - 2\sum_{j\neq i} f_{ij}\chi_i\chi_{ij} + \sum_j\sum_{k\neq j} f_{jk}\chi_k\chi_{ij} - 2\mu\chi_i \\
&\overset{(d)}{=} \sum_j \chi_{ij}(f_j + \sum_k f_{jk}\chi_k - 2f_{ij}\chi_i) - 2\mu\chi_i \ .
\end{aligned}
$$

In (a) we expanded $\langle s_j s_k\rangle$ and, after distinguishing the case where $i\neq j\neq k$, we exploited Eq.(5.4); in (b) we used $\chi_{ii} = \langle s_i^2\rangle - \langle s_i\rangle^2 = 1-\chi_i^2$; in (c) we added and subtracted a sum; in (d) we used $f_{ii} = 0\ \forall i$.

A.1.2 Derivation of Eq.(5.7)

We start by substituting in Eq.(4.19) the result we have just derived for $\dot{\chi}_i$, Eq.(5.6). $\forall i,j \in 1,\dots L$ with $(i \neq j)$:

$$\begin{aligned}\dot{\chi}_{ij} = &\sum_k f_k\langle s_i s_j s_k\rangle + \sum_{k<l} f_{kl}\langle s_i s_j s_k s_l\rangle - \langle s_i s_j\rangle\Big(\sum_k f_k\chi_k + \sum_{k<l} f_{kl}\langle s_k s_l\rangle\Big) + \\ &- \chi_i \sum_k \chi_{jk}(\hat{f}_k - 2f_{jk}\chi_j) - \chi_j \sum_k \chi_{ik}(\hat{f}_k - 2f_{ik}\chi_i) - (4\mu + rc_{ij})\chi_{ij} \qquad \text{(A.1)}\end{aligned}$$

where we have defined $\hat{f}_k = f_k + \sum_j f_{jk}\chi_j$. We will now analyze separately the terms highlighted in blue (B), red (R) and violet (V) and cyan (C). In order to substitute Eq.(5.4-5.5) we again decompose the sums distinguishing cases where some of the indices are equal.

$$\begin{aligned}V &= (\chi_{ij} + \chi_i\chi_j)\Big(\sum_k f_k\chi_k + \sum_{k<l} f_{kl}(\chi_{kl} - \chi_k\chi_l)\Big) \\ &= (\chi_{ij} + \chi_i\chi_j)\Big(\sum_{k\neq i,j} f_k\chi_k + f_i\chi_i + f_j\chi_j + \sum_{\substack{k<l \\ k,l\neq i,j}} f_{kl}(\chi_{kl} + \chi_k\chi_l) + \\ &\quad + \sum_{k\neq i,j}\left[f_{ik}(\chi_{ik} + \chi_i\chi_k) + f_{jk}(\chi_{jk} + \chi_j\chi_k)\right] + f_{ij}(\chi_{ij} + \chi_i\chi_j)\Big) \\ B &= \sum_{k\neq i,j} f_k\langle s_i s_j s_k\rangle + f_i\chi_j + f_j\chi_i \\ &= \sum_{k\neq i,j} f_k(\chi_i\chi_j\chi_k + \chi_i\chi_{jk} + \chi_j\chi_{ik} + \chi_k\chi_{ij}) + f_i\chi_j + f_j\chi_i \\ R &= \sum_{\substack{k<l \\ k,l\neq i,j}} f_{kl}\langle s_i s_j s_k s_l\rangle + \sum_{k\neq i,j}\left[f_{ik}\langle s_j s_k\rangle + f_{jk}\langle s_i s_k\rangle\right] + f_{ij} \\ &= \sum_{\substack{k<l \\ k,l\neq i,j}} f_{kl}(\chi_i\chi_j\chi_k\chi_l + \chi_i\chi_j\chi_{kl} + \chi_i\chi_k\chi_{jl} + \chi_i\chi_l\chi_{jk} + \chi_j\chi_k\chi_{il} + \chi_j\chi_l\chi_{ik} + \\ &\quad + \chi_k\chi_l\chi_{ij} + \chi_{ij}\chi_{kl} + \chi_{ik}\chi_{jl} + \chi_{il}\chi_{jk}) + \sum_{k\neq i,j}\left[f_{ik}(\chi_{jk} + \chi_j\chi_k) + \right. \\ &\quad \left. + f_{jk}(\chi_{ik} - \chi_i\chi_k)\right] + f_{ij} \\ C &= \chi_i\sum_k \chi_{jk}(f_k + \sum_l f_{kl}\chi_l - 2f_{jk}\chi_j) \\ &= \chi_i\sum_{k\neq i,j}\chi_{jk}(f_k + \sum_l f_{kl}\chi_l - 2f_{jk}\chi_j) + \chi_i\chi_{ij}(f_i + \sum_l \chi_{il}\chi_l - 2f_{ij}\chi_j) + \\ &\quad + \chi_i(1-\chi_j^2)(f_j + \sum_l f_{jl}\chi_l)\end{aligned}$$

In the last line we have used $\chi_{ii} = 1 - \chi_i^2$, $f_{ii} = 0\ \forall i$ and the definition of $\hat{f}_i$. In addition, note that there is a term in Eq.(A.1) which is nothing but (C) after exchanging $i \leftrightarrow j$.

Summing all the terms in Eq.(A.1) and simplifying:

$$\begin{aligned}\dot{\chi}_{ij} =& -(4\mu + rc_{ij})\chi_{ij} - 2f_i\chi_i\chi_{ij} - 2f_j\chi_j\chi_{ij} + f_{ij}(1 - \chi_{ij}^2 - \chi_i^2\chi_j^2 + 2\chi_i\chi_j\chi_{ij}) + \\ &+ \sum_{k\neq i,j} f_{ik}(\chi_{jk} + \chi_j\chi_k - \chi_{ij}\chi_{ik} + \chi_{ik}\chi_i\chi_j - \chi_{ij}\chi_i\chi_k - \chi_i^2\chi_j\chi_k) + \\ &+ \sum_{k\neq i,j} f_{jk}(\chi_{ik} + \chi_i\chi_k - \chi_{ij}\chi_{jk} + \chi_{jk}\chi_i\chi_j - \chi_{ij}\chi_j\chi_k - \chi_i\chi_j^2\chi_k) + \\ &+ (\chi_i^2\chi_j - \chi_j - \chi_i\chi_{ij})\sum_l f_{il}\chi_l + (\chi_i\chi_j^2 - \chi_i - \chi_j\chi_{ij})\sum_l f_{jl}\chi_l + \\ &+ \sum_{\substack{k<l \\ k,l\neq i,j}} f_{kl}(\chi_{ik}\chi_j\chi_l + \chi_{jk}\chi_i\chi_l + \chi_{il}\chi_j\chi_k + \chi_{jl}\chi_i\chi_k + \chi_{ik}\chi_{jl} + \chi_{il}\chi_{jk}) + \\ &- \sum_{k\neq i,j} \chi_i\chi_{jk}\sum_l \chi_l f_{kl} - \sum_{k\neq i,j} \chi_j\chi_{ik}\sum_l \chi_l f_{kl} \ . \end{aligned} \tag{A.2}$$

This is the final result for the dynamics of the second-order cumulants, where all sums have no equal indices. For the sake of elegance, it is possible to rewind this "decomposition" and the result is precisely Eq.(5.7).

B

Appendix to Chapter 6

2.1. FFPopSim settings

We present here the FFPopSim [Zanini and Neher (2012)] settings used to simulate the data used in Chapter 6 and repsented in [Zeng et al. (2021)].

We use the class `haploid_highd` i.e. *individual-based* simulations that handle the population as a set of *clones* $(g_i, n_i(t))$ where g_i is a genotype and $n_i(t)$ is the number of individuals with genotype g_i at time t (only existing clones are tracked). At each generation, the size of each clone is first updated $n_i(t) \to n_i(t+1) \sim \mathcal{P}_\lambda$ where $\mathcal{P}$ is the Poisson distribution with parameter $\lambda = \frac{1}{\langle e^F \rangle} e^{F(g_i)+1-\frac{1}{N}\sum_j n_j(t)}$, N is the carrying capacity and $F(g)$ is the fitness function. A fraction r^* (outcrossing rate) of the resulting offspring is destined to the recombination step, paired and reshuffled. Finally, each individual is allowed to mutate with probability $1-e^{-L\mu}$, where μ is the recombination rate, the exact number of mutations being Poisson distributed $\mathcal{P}_{L\mu}$.
We have used FFPoSim in a similar manner as in Zeng and Aurell (2020) and we will here only list the settings. Parameters which are the same in all simulations reported in this paper are listed in Tab. B.1. Parameters that have been varied (not all variations reported in the paper) are listed in Tab. B.2.

It is important to notice that the out-crossing rate r^* in FFPopSim *a priori* differs from our recombination rate, r, appearing in eq. (4.28). In the simulation package, dynamics is discrete in time (with time step of one generation) and r^* is a probability taking value between 0 and 1. In our theory, r is a rate, which can take any positive value. In the examples given in Zanini and Neher (2012), e.g. Fig 2 in the main text and Fig 2 in Supplementary Information the out-crossing probability does not exceed 10^{-2}. For such low values r^* coincides with a rate (since the time step is equal to unity), which justifies its denomination. We use this correspondence $r^* = 1-e^{-r} \sim r$ between the out-crossing rate r^* in FFPopSim and our recombination rate r, valid for small values, to produce the scatter plots in Figs. 6.1 and 6.4.

Notice that this correspondence breaks down for large recombination rates. Indeed, even for out-crossing rate $r^* = 1$ in the simulation package, mutations and fitness effects can still be quite large, depending on the values of the f_{ij}'s and of μ, and QLE is not recovered. In the theory, however, all fitness and mutation effects become relatively weak, of the order of $1/r$.

In addition to forward simulations, a subsequent release of the original FFPopSim package allows for the possibility of tracking the genealogy of loci e.g. that of the central locus. Such information can be used in the first place to draw a coalescent tree Neher et al. (2013): technically, this is done by converting the genealogy in a `BioPython` tree and using

Table B.1: **Main default parameters of FFPopSim used in the simulations.**

number of loci (L)	25
number of traits	1
circular	False
carrying capacity (N)	200
generation	$10,000$
recombination model	CROSSOVERS
crossover rate (ρ)	0.5
fitness additive(coefficients)	Gaussian random number with $\sigma(\{f_i\}) = 0.05$

Table B.2: **Variable parameters of FFPopSim used in the simulation.**

initial genotypes	binary random numbers
out-crossing rate (r)	$[0.,\ 1.0]$
mutation rate (μ)	$[0.05,\ 0.5]$
epistatic fitness	Gaussian random number with $\sigma(\{f_{ij}\}) \in [0.004,\ 0.04]$

the module `Bio.Phylo` for plotting purposes.

A quantitative analysis of such trees can be carried out, for instance the Time to the Most Recent Common Ancestor T_{MRCA} of a group of individuals at time t is nothing but the temporal distance of the leaves (individuals) from the root (common ancestor) in the corresponding coalescent tree $\mathcal{CT}$, as shown in Fig. B.1. In the same vein, we are able to evaluate the average pair coalescent time $\langle T_2 \rangle$: we sample $n_2 = 10$ pairs of leaves, per each of them we extract the information about their subtree $\mathcal{CT}_2$ and evaluate the $T_{MRCA}(\mathcal{CT}_2)$, which now corresponds to the difference between the present and the time in the past when the two branches stemming from the chosen leaves merge. Averaging over the sample of size n_2 gives an estimate of the desired quantity.

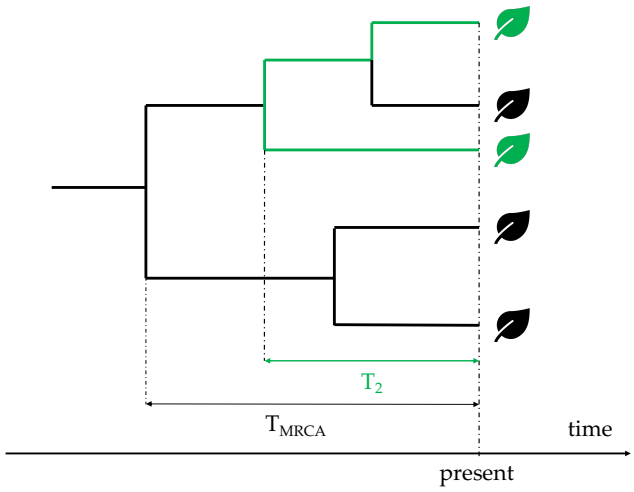
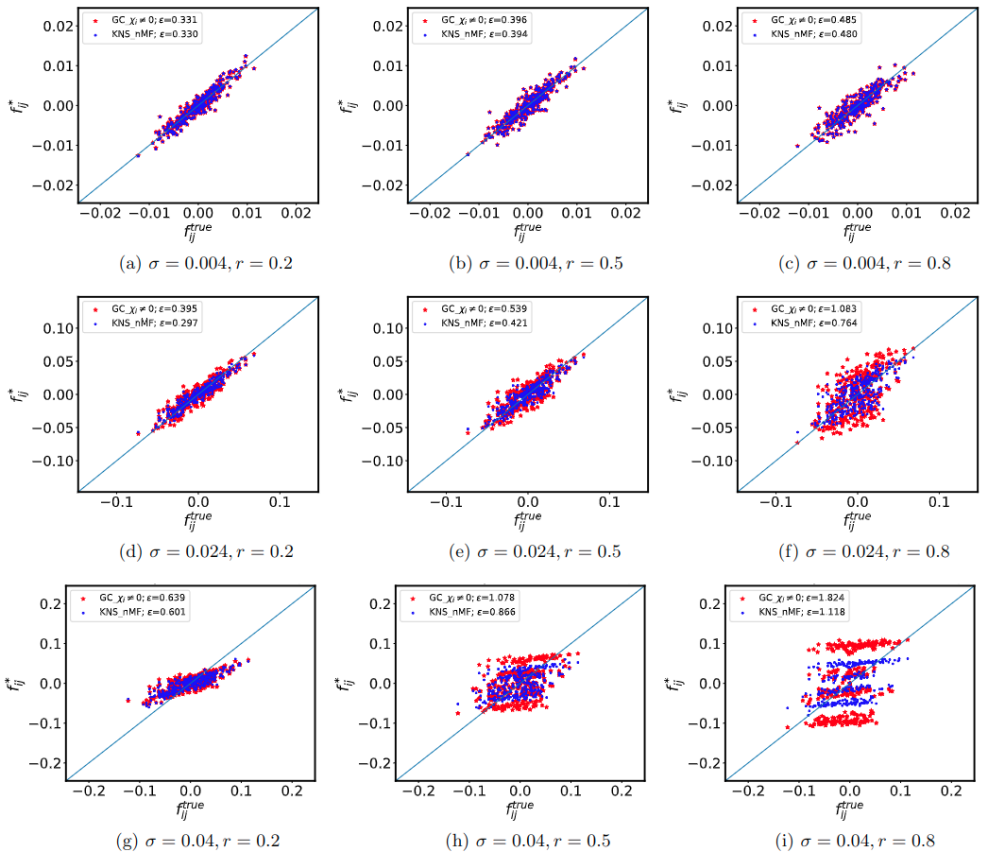


Figure B.1: Illustrative coalescent tree. The Time to the Most Recent Common Ancestor T_{MRCA} of a tree is the difference between the current time and the time point where all the branches merge. The pair coalescent time T_2 for two chosen leaves (individuals) is the T_{MRCA} with respect to their subtree (highlighted in green). Image taken from Zeng et al. (2021).

2.2. Numerical comparison between Eq. (6.7) with nMF and SIE couplings

Here we compare the results on the inference of epistatic fitness using the extended formula Eq. (6.7) with nMF and SIE couplings as done in [Zeng et al. (2021)]. We present the numerical simulations in Fig. B.2 with a fixed mutation rate $\mu = 0.2$ while different $\sigma(\{f_{ij}\})$s and recombination rate r. The blue dots are for eq. (6.7) with nMF couplings while the red stars are with SIE couplings. As shown in the top row of Fig. B.2 (a), (b) and (c), two methods perform almost the same for weak epistatic fitness $\sigma(\{f_{ij}\}) = 0.004$. When increasing $\sigma(\{f_{ij}\})$ and for sufficiently low recombination rates as in Fig. B.2 (d), (e), (g) , we observe that using nMF couplings is slightly better than SIE couplings, as it is evident from the smaller reconstruction error of the former with respect to the latter. Finally, none of them works for large $\sigma(\{f_{ij}\})$ and high r as shown in Fig. B.2 (f), (h) and (i). The parameters for these cases are located in the white area of Fig. 6.5 where the system may not be in the QLE state and both the reconstructions (Neher-Shraiman and Gaussian closure) fail. This part with strong correlations has been studied extensively in Dichio's Master's thesis [Dichio (2020); Dichio et al. (2021)].



(a) $\sigma = 0.004, r = 0.2$ (b) $\sigma = 0.004, r = 0.5$ (c) $\sigma = 0.004, r = 0.8$

(d) $\sigma = 0.024, r = 0.2$ (e) $\sigma = 0.024, r = 0.5$ (f) $\sigma = 0.024, r = 0.8$

(g) $\sigma = 0.04, r = 0.2$ (h) $\sigma = 0.04, r = 0.5$ (i) $\sigma = 0.04, r = 0.8$

Figure B.2: Scatter plots for testing and reconstructed f_{ij}s. The standard deviation $\sigma(\{f_{ij}\}^{true})$ increases from top to bottom rows $(0.004, 0.024$ and 0.04 respectively) and recombination rate r enlarges in columns from left to right $(0.2, 0.5$ and 0.8 respectively). Red stars for $f_{ij}^* = \chi_{ij} \cdot (4\mu + rc_{ij})/((1-\chi_i^2)(1-\chi_i^2))$ and blue dots for $f_{ij}^* = (4\mu + rc_{ij}) \cdot J_{ij}^{*,nMF}$. The other parameters are the same to those in Fig. 6.4. In the regime of weak σ and r, the reconstructions are equivalent. Increasing σ for sufficiently small r as in (d),(e),(g) the mean field reconstruction outperforms the Gaussian one. However, both reconstructions fail for sufficiently high σ, r, as in (f),(g),(h),(i), where strong correlations emerge between loci that drive the system out of the QLE phase [Dichio (2020); Dichio et al. (2021)]. One realization of the fitness terms f_{ij} and f_i for each parameter value. Image taken from Zeng et al. (2021).

C

Appendix to Chapter 7

3.1. RBM training details for LP and WW domains

We train RBMs on multiple sequence alignemnts of Lattice Proteins and WW domains. In particular, each visible unit of the RBM can take one out of 21 possible states (20 amino acids + gap), while the potential over the hidden unit are determined by dReLu potentials as described in Section 2.3. We use Persistent Contrastive Divergence with L_1^2 regularization for the training. The code and the data for the training can be found in Tubiana (2018a).

Below we report the hyper-parameters used for learning.

- For WW (N=31):
  - M = 50
  - Batch size = 100
  - Number of epochs = 500
  - Learning rate = 5×10^{-3} (which has a decay rate of 0.5 after 50% of iterations)
  - $L_1 b$ regularization = 0.25
  - Number of MC step between each update = 10
- For Lattice Protein (N=27):
  - $M = 100$
  - Batch size = 100
  - Number of epochs = 100
  - Learning rate = 5×10^{-3} (which has a decay rate of 0.5 after 50% of iterations)
  - $L_1 b$ regularization = 0.025
  - Number of MC step between each update = 5

For the local RBMs trained respectively on the three specificity classes of WW sequences predicted by the original RBM, we use $M = 30$ and keep all the other hyper-parameters unchanged.

3.2. Lists of the tested sequences

Here we present the reference sequences sampled with the MC algorithm and tested using AlphaFold (note that the first sequence of each list represent the YAP1 wild-type sequence from Espanel and Sudol (1999), while the last is the natural wild-type specific for each class) together with the predicted specificity using local RBMs:

- From YAP1 to wild-type protein in class II:
  - LPAGWEMAKTSS-GQRYFLNHIDQTTTWQDP
  - LPAGWEMAKTSD-GERYFINHNTKTTTWQDP predicted I
  - LPPGWEEARTPD-GRVYFINHNTKTTTWQDP predicted I
  - LPPGWEEARAPD-GRTYYYNHNTKTTTWEKP predicted II/III
  - LPPGWTEHKAPD-GRTYYYNHNTKTSTWEKP predicted II/III
  - LPSGWTEHKAPD-GRTYYYNTETKQSTWEKP predicted II/III
  - AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
- From YAP1 to wild-type protein in class IV:
  - LPAGWEMAKTSS-GQRYFLNHIDQTTTWQDP
  - LPAGWEMRRTPS-GRVYFVNHITRTTQWEDP predicted I
  - LPPGWEERRDPS-GRVYYVNHITRTTQWERP predicted I
  - LPPGWEERVSRS-GRVYYVNHITRTTQWERP predicted I
  - LPPGWEKRMSRS-GRVYYVNHITRTTQWERP predicted I
  - LPPGWEKRMSRSSGRVYYVNHITRASQWERP predicted IV
  - LPPGWEKRMSRSSGRVYYFNHITNASQWERP
- From YAP1 to wild-type protein in class I:
  - LPAGWEMAKTSS-GQRYFLNHIDQTTTWQDP
  - LPAGWEMAKTSE-GQRYFINHNTQTTTWQDP
  - LPPGWEMAYTPE-GERYFINHNTKTTTWLDP
  - LPPGWEMGITRG-GRVFFINHETKSTTWLDP
  - LPRSWTYGITRG-GRVFFINHEAKSTTWLHP
  - LPRSWTYGITRG-GRVFFINEEAKSTTWLHP

3.3. Additional paths associated to I→IV transition

We show in Figure C.1 different paths sampled for different couples of class I and class IV wild-types using the same simulation parameters as in Figure 7.6. This figure shows that all the paths cross a region of the input space where natural sequences are lacking.

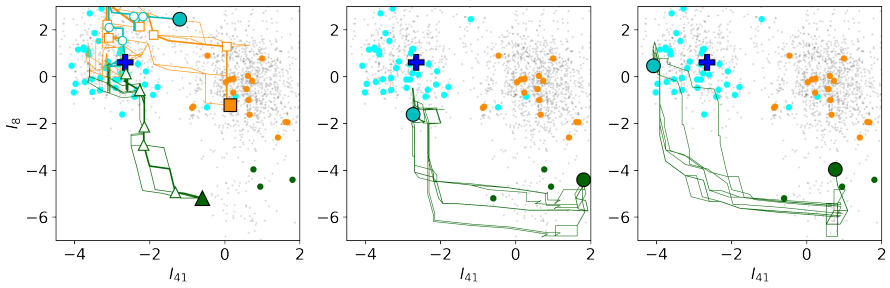
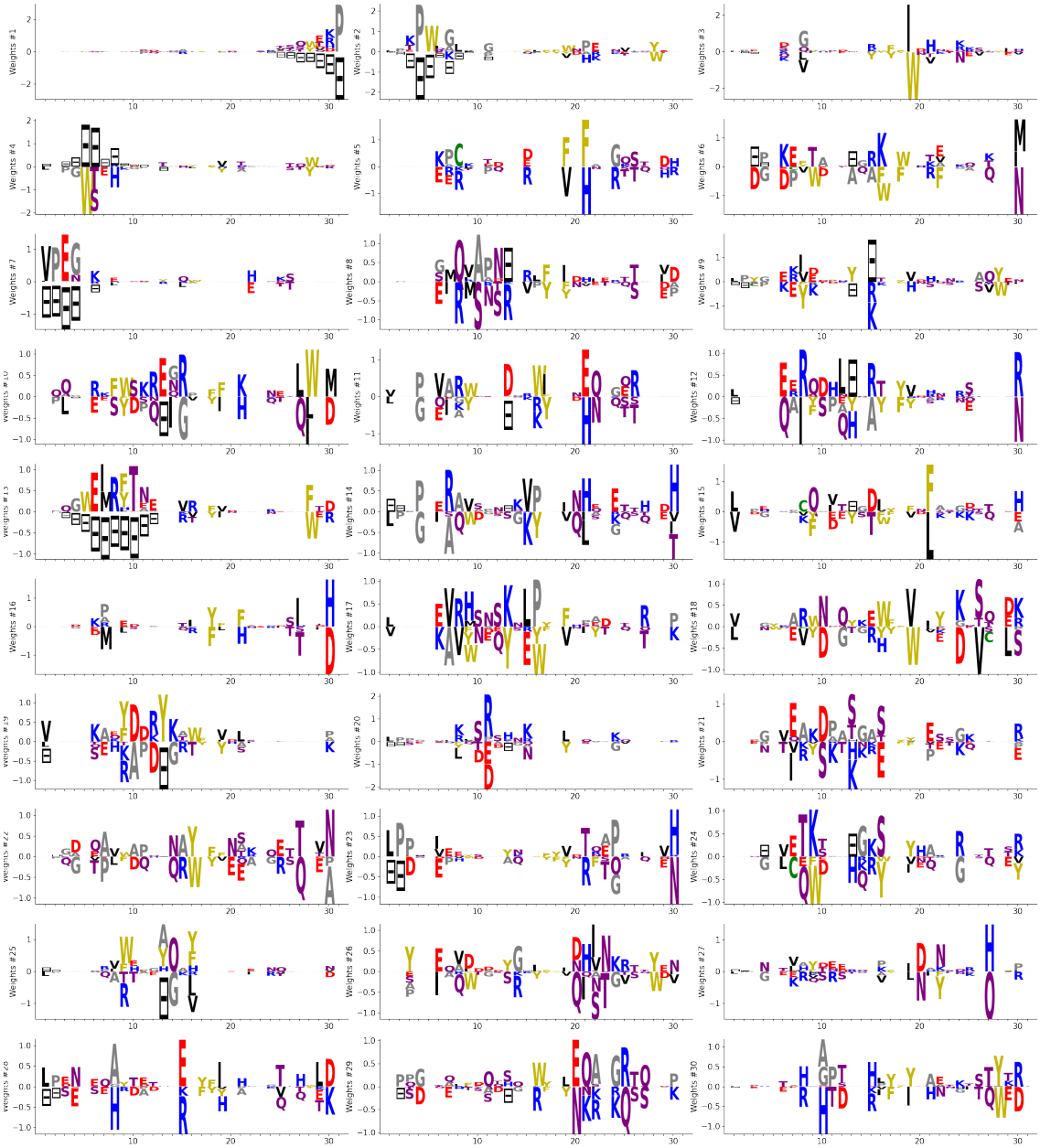
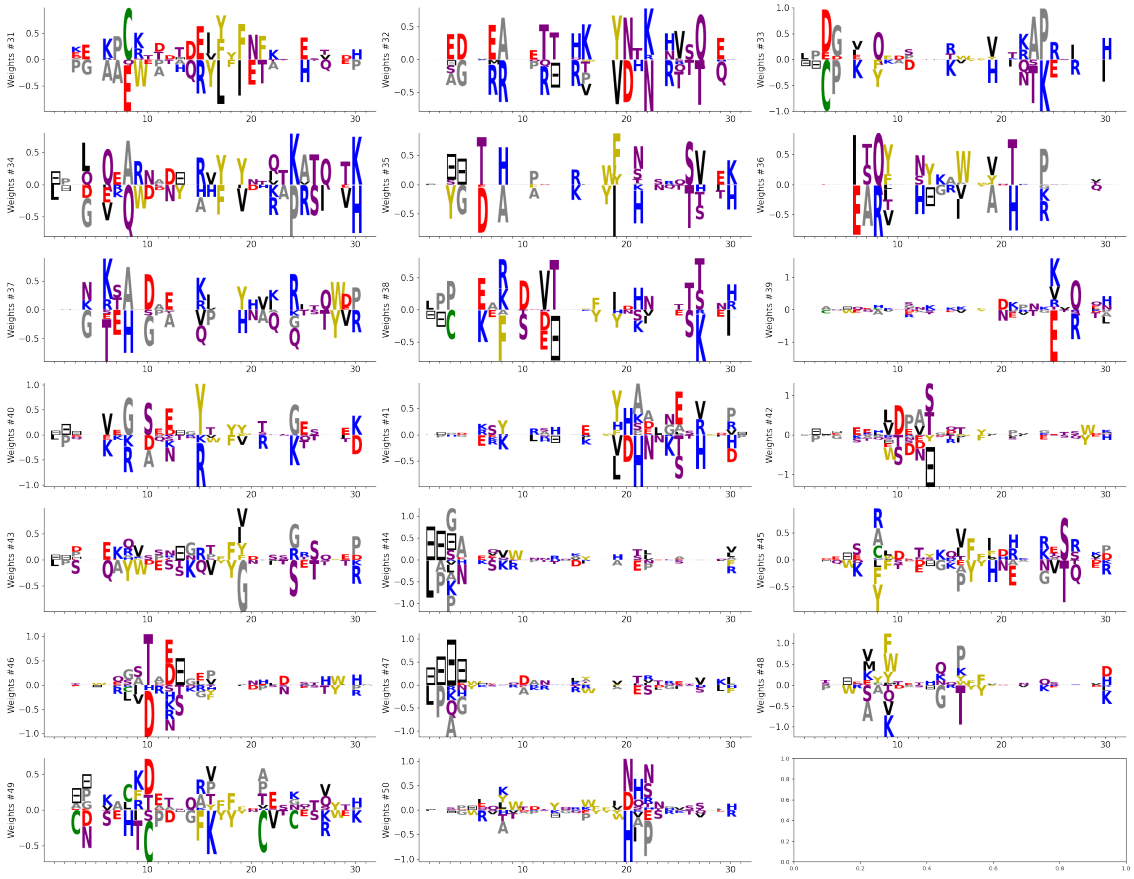


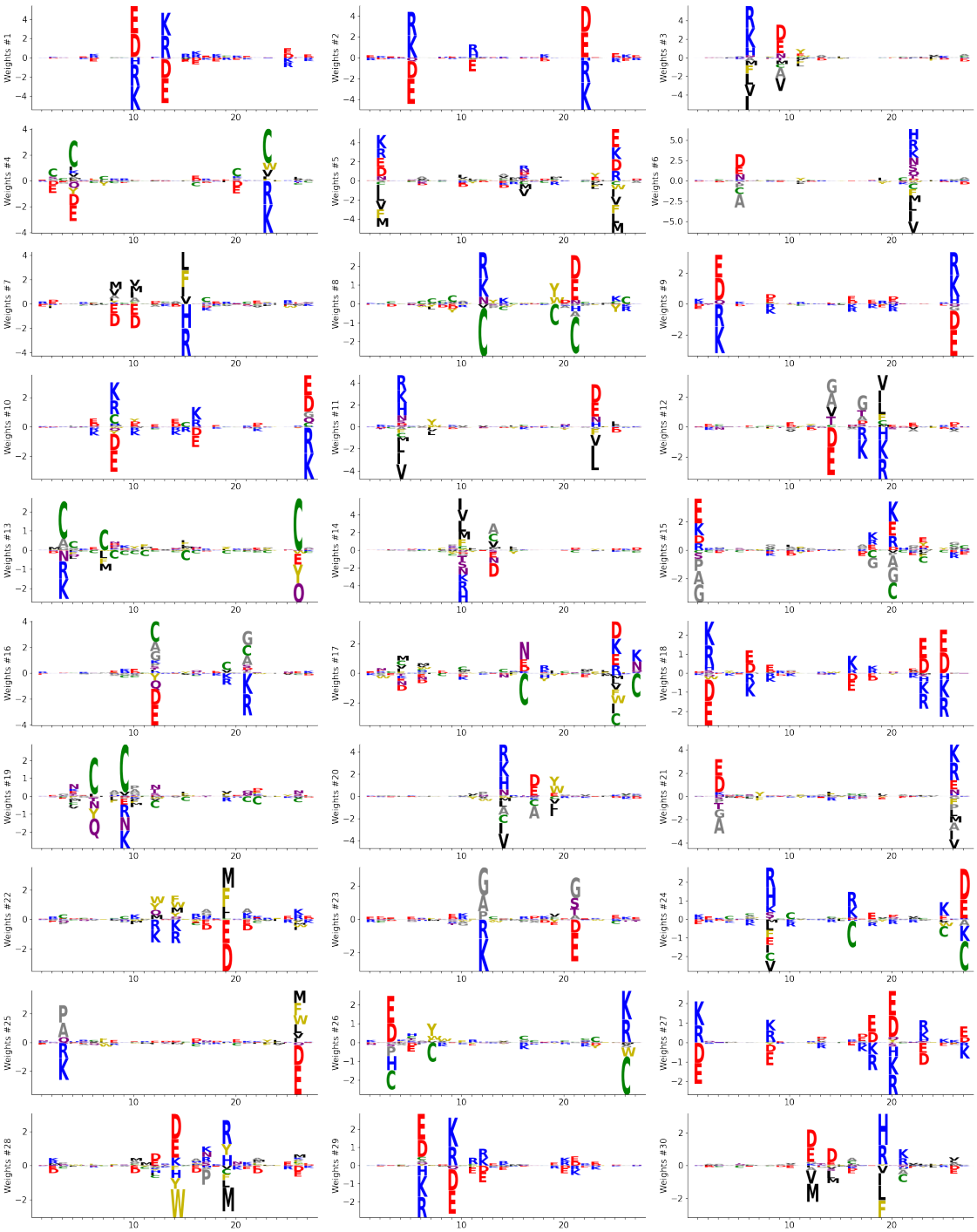
Figure C.1: (a) Same as Figure 7.6(a). 10 new samples of paths connecting specificity class I to class IV (PDB IDs: 2LTW and 1I8G, respectively) and class I to class II/III are shown. (b)-(c) 10 paths joining two pairs of wild-type sequences in classes I and IV obtained with the same sampling algorithm as in Figure 2 in the main text. Image and caption taken from Mauri et al. (2023a) (Supplemental Material).

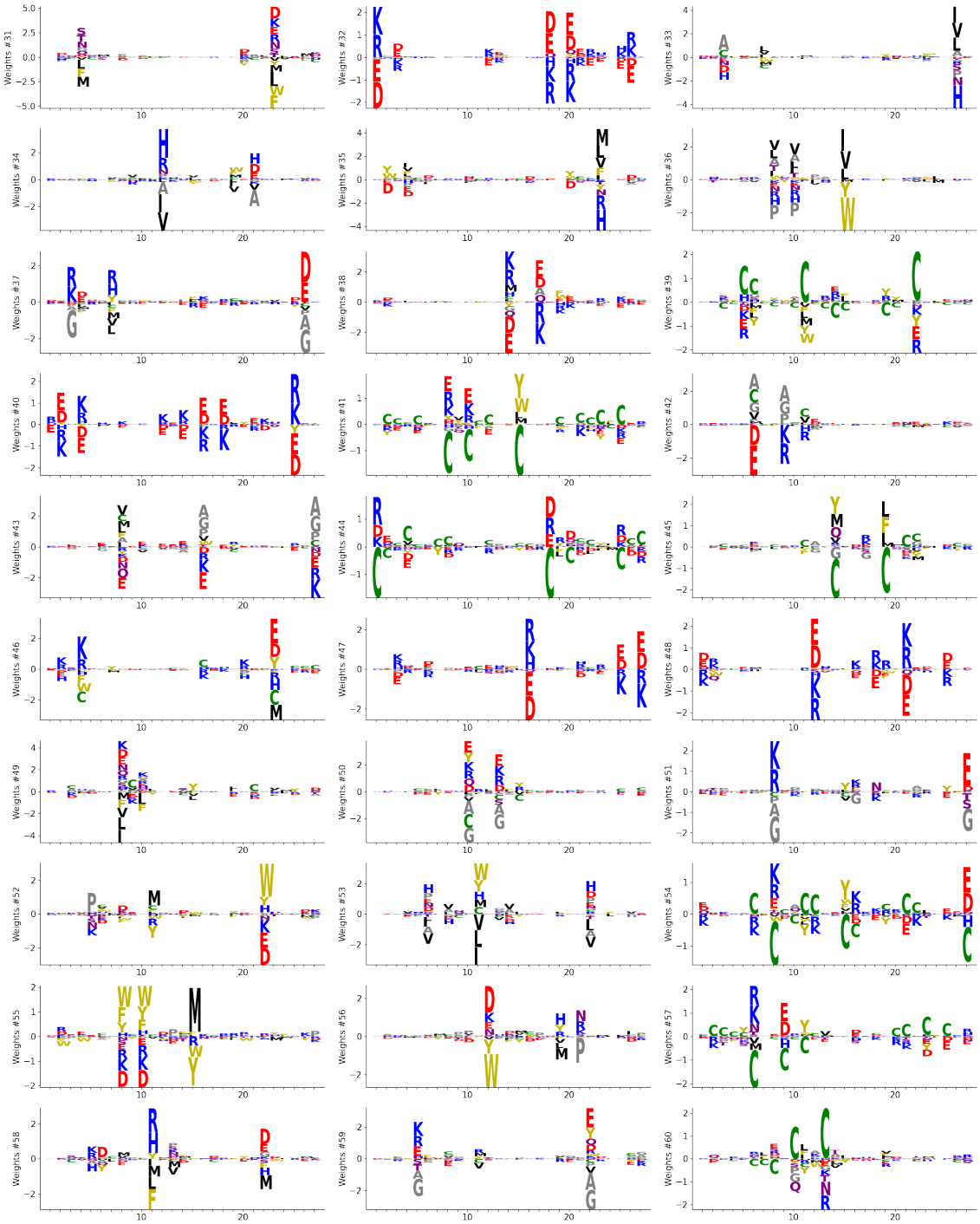
3.4. Weights Logo for the WW domain

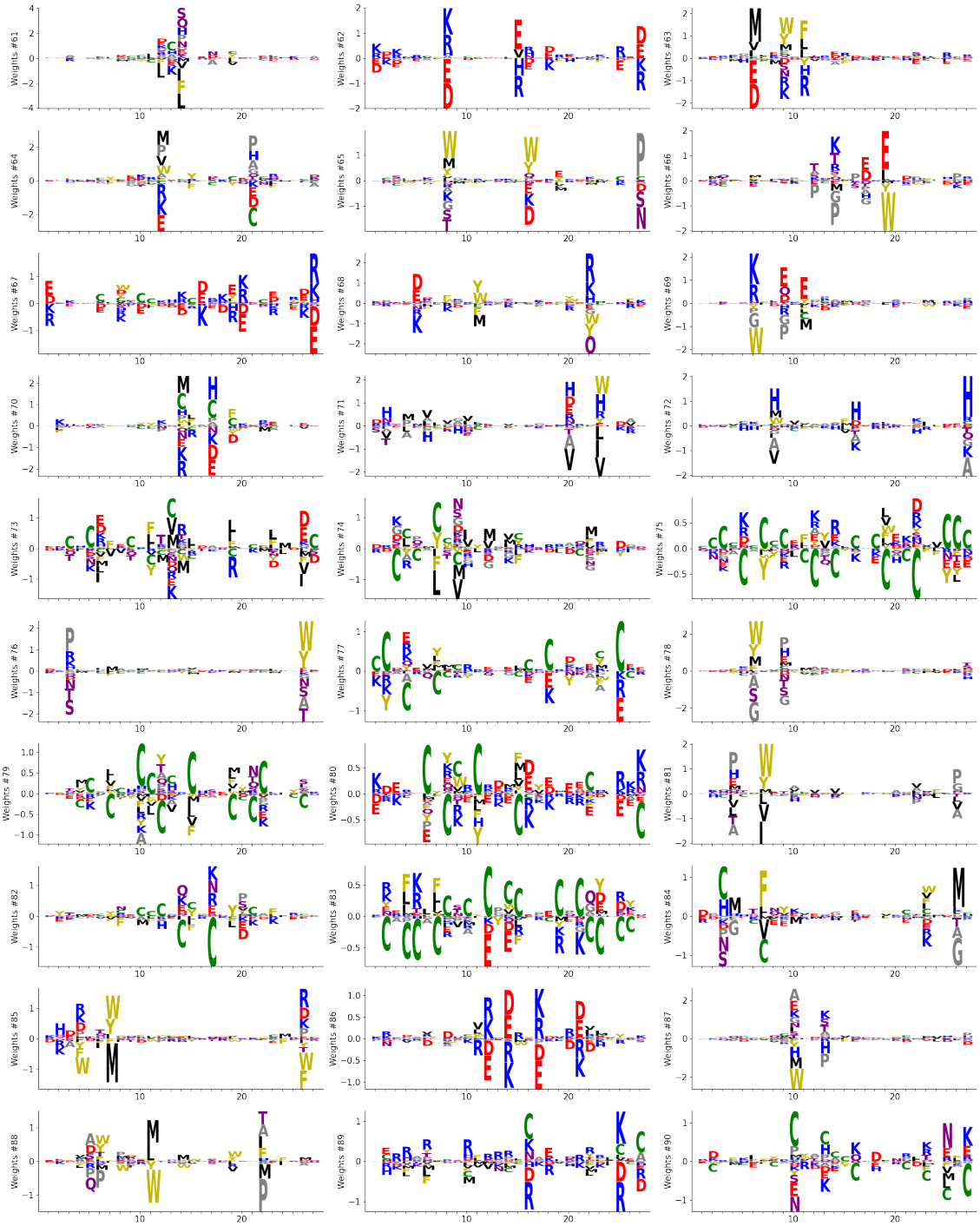


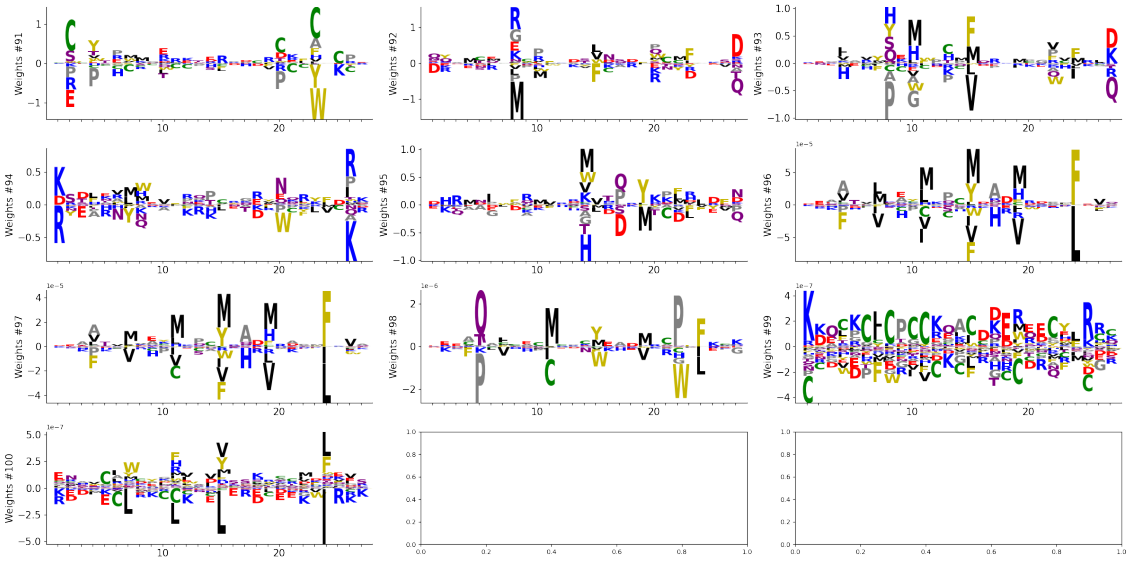

Weights #31
Weights #32
Weights #33
Weights #34
Weights #35
Weights #36
Weights #37
Weights #38
Weights #39
Weights #40
Weights #41
Weights #42
Weights #43
Weights #44
Weights #45
Weights #46
Weights #47
Weights #48
Weights #49
Weights #50

3.5. Weights Logo for the Lattice Proteins




Weights #31
Weights #32
Weights #33
Weights #34
Weights #35
Weights #36
Weights #37
Weights #38
Weights #39
Weights #40
Weights #41
Weights #42
Weights #43
Weights #44
Weights #45
Weights #46
Weights #47
Weights #48
Weights #49
Weights #50
Weights #51
Weights #52
Weights #53
Weights #54
Weights #55
Weights #56
Weights #57
Weights #58
Weights #59
Weights #60


Weights #61
Weights #62
Weights #63
Weights #64
Weights #65
Weights #66
Weights #67
Weights #68
Weights #69
Weights #70
Weights #71
Weights #72
Weights #73
Weights #74
Weights #75
Weights #76
Weights #77
Weights #78
Weights #79
Weights #80
Weights #81
Weights #82
Weights #83
Weights #84
Weights #85
Weights #86
Weights #87
Weights #88
Weights #89
Weights #90


Weights #91
Weights #92
Weights #93
Weights #94
Weights #95
Weights #96
Weights #97
Weights #98
Weights #99
Weights #100

D

Appendix to Chapter 8

4.1. Neutral theory of evolution

In this section, we present Kimura neutral theory of evolution as presented in Mauri et al. (2023a) (Supplemental Material).

Let's consider a sequence (with A number of states per site) evolving under mutations only. Given a site i along the sequence the probability of that site to be in a given state a at time t, $x_a^i(t)$, evolves through time under the following equation:

$$\frac{d}{dt}x_a^i(t) = -\mu x_a^i(t) + \frac{\mu}{A}\sum_{b\neq a} x_b^i(t) = \sum_b W_{a,b}x_b^i(t) \tag{D.1}$$

where μ is the mutation rate. Solving the linear differential equation, we can compute the probability that a site mutates into a specific new state in the time interval Δt as

$$p_{\neq} = \frac{1}{A}\left(1 - e^{\frac{A\mu\Delta t}{1-A}}\right), \tag{D.2}$$

while the probability of not mutating is $p_{=} = 1-(A-1)p_{\neq}$. Here we set $\Delta t = 1$. Hence the probability of evolving from a sequence $\mathbf{v}$ to $\mathbf{v}'$ is

$$\pi(\mathbf{v},\mathbf{v}') = p_{=}^{Nq}p_{\neq}^{N(1-q)} = e^{-N\Phi(q)}, \tag{D.3}$$

where q is the overlap between the two sequence and $\Phi(q) = q\log\frac{p_{\neq}}{p_{=}} - \log p_{\neq}$ (which is equal to Eq. (8.4) up to an irrelevant constant). Hence, the probability to go from $\mathbf{v}_0$ to $\mathbf{v}_T$ in T steps is

$$P(\mathbf{v}_0 \to \mathbf{v}_T;T) = \sum_{\{\mathbf{v}\}_{t=1}^{T-1}} \pi(\mathbf{v}_0,\mathbf{v}_1)\pi(\mathbf{v}_1,\mathbf{v}_2)\ldots\pi(\mathbf{v}_{T-1},\mathbf{v}_T) = \sum_{\{\mathbf{v}\}_{t=1}^{T-1}} e^{-N\sum_t \Phi(q_t)}, \tag{D.4}$$

which can be computed exactly as

$$P(\mathbf{v}_0 \to \mathbf{v}_T;T) = \frac{p_{\neq}^{TN}}{A^N}\left[\left(\frac{p_{=}}{p_{\neq}} + A - 1\right)^T - \left(\frac{p_{=}}{p_{\neq}} - 1\right)^T\right]^D \times \left[\left(\frac{p_{=}}{p_{\neq}} + A - 1\right)^T - (1-A)\left(\frac{p_{=}}{p_{\neq}} - 1\right)^T\right]^{N-D}, \tag{D.5}$$

where D is the Hamming distance between the two sequences. The last equation corresponds to Kimura's theory of neutral evolution Kimura (1983). The optimal time for which this probability is maximised, T^*, and converges to $1/A^N$ for $T \to \infty$.

4.2. Consensus sequence from MF solutions and the case for WW domain

In this section, we present the computation of consensus sequences from the mean-field solution as presented in Mauri et al. (2023a) (Supplemental Material).

To obtain the average distance from the direct space, we need to compute at each time at each site the probability of a specific state $a = 1, ..., A$. This can be computed as

$$\begin{aligned} f_{i,t}(a|\{\mathbf{m}_t, q_t\}) &= \frac{\partial}{\partial \beta g_{it}(a)} \log Z_{path} = \\ &= \frac{\partial}{\partial \beta g_{it}(a)} \sum_{\mathcal{V}} \exp\left[\sum_{i,t,a} \beta g_{it}(a)\delta_{a,v_{it}} + N\beta \sum_{\mu,t} \Gamma(m_\mu^t) - N\beta \sum_t \Phi(q_t)\right] = \\ &= \frac{\partial}{\partial \beta g_{it}(a)} \log Z_i \,, \end{aligned} \tag{D.6}$$

where we write $g_{it}(a) = g_i(a)$ for any t. Once $f_{i,t}$ is computed, we obtain the consensus sequence $\mathbf{v}_t = \{v_{i,t} = \text{argmax}_a f_{i,t}\}$. The consensus sequences for the MF path shown in Figure 8.7 are:

- I→IV Cont scenario:
  - `LPAGWEMAKTSS-GQRYFLNHITRTTTWQDP`
  - `LPAGWEMRKTSS-GQVYFLNHITRTTTWEDP`
  - `LPPGWEMRKTSS-GRVYFLNHITRTTQWEDP`
  - `LPPGWEMRKSRS-GRVYFLNHITRTTQWEDP`
  - `LPPGWEKRKSRS-GRVYYLNHITRTTQWERP`
  - `LPPGWEKRMSRS-GRVYYLNHITRTTQWERP`
  - `LPPGWEKRMSRS-GRVYYFNHITRASQWERP`
- I→IV Evo scenario:
  - `LPPGWEKRKSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRKSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRKSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRKSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRKSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRMSRS-GRVYYLNHITKTTQWERP`
  - `LPPGWEKRMSRS-GRVYYLNHITKTTQWERP`
- I→II Cont scenario:
  - `LPAGWEMAKTSD-GQRYFLNHITQTTTWQDP`
  - `LPAGWEMAKTPD-GQRYFLNHITKTTTWEDP`
  - `LPAGWEEAKTPD-GRTYFYNHITKTTTWEDP`
  - `LPAGWEEAKTPD-GRTYYYNHITKTTTWEKP`
  - `LPAGWTEHKTPD-GRTYYYNHITKTTTWEKP`
  - `LPSGWTEHKTPD-GRTYYYNTITKQSTWEKP`

  - LPSGWTEHKSPD-GRTYYYNTETKQSTWEKP

- I→II Evo scenario:
  - LPAGWEEAKTPD-GRRYFLNHITKTTTWEDP
  - LPAGWEEAKTPD-GRTYYYNHITKTTTWEDP
  - LPAGWEEAKTPD-GRTYYYNHITKTTTWEKP
  - LPAGWEEAKTPD-GRTYYYNHITKTTTWEKP
  - LPSGWTEHKTPD-GRTYYYNTITKQSTWEKP
  - LPSGWTEHKTPD-GRTYYYNTETKQSTWEKP
  - LPSGWTEHKSPD-GRTYYYNTETKQSTWEKP

E

Appendix to Chapter 10

5.1. Full list of tested sequences

All highlighted sequences are natural wild-types.

E.1.0.1 1st batch

first batch of tested sequences. Parameters: $\Lambda = 0.1$, $\beta = 3$, $T = 27$.

>1to4_t0_s37.73
LPAGWEMAKTSS-GQRYFLNHIDQTTTWQDP
>1to4_t5_s28.99
LPAGWEMRKTSS-GRRYFVNHITRTTTWQDP
>1to4_t10_s24.11
LPAGWEMRRTPS-GRVYFVNHITRTTQWEDP
>1to4_t12_s22.26
LPPGWEERRTPS-GRVYFVNHITRTTQWEDP
>1to4_t15_s19.74
LPPGWEERRDPS-GRVYYVNHITRTTQWERP
>1to4_t17_s19.25
LPPGWEERVDPS-GRVYYVNHITRTTQWERP
>1to4_t19_s20.59
LPPGWEERVSRS-GRVYYVNHITRTTQWERP
>1to4_t21_s23.0
LPPGWEKRMSRS-GRVYYVNHITRTTQWERP
>1to4_t24_s29.01
LPPGWEKRMSRSSGRVYYVNHITRASQWERP
>1to4_t26_s32.36
LPPGWEKRMSRSSGRVYYFNHITNASQWERP

E.1.0.2 2nd batch

Parameters: $\Lambda = 0.1$ (only for the paths), $\beta = 3$. In this case the 1 to 4 path ends in the same sequence of the 1st batch but it has not been tested.

>classI_13_s20.147064208984375
LPPGWEQRYTPE-GRPYFVDHNTRTTTWVDP
>classI_29_s23.194305419921875
LPPGWERRTDNF-GRTYYVDHNTRTTTWKRP
>classI_12_s24.534210205078125
LPSGWEERKDAK-GRTYYVNHNNRTTTWTRP
>classI_26_s24.639556884765625

LPSGWEQRFTPE-GRAYFVDHNTRTTTWVDP
>classI_9_s24.893524169921875
LPPGWEMKYTSE-GVRYFVDHNTRTTTFKDP
>classI_0_s25.825653076171875
LPPGWEIRKDGR-GRVYYVDHNTRKTTWQRP
>classI_5_s26.013275146484375
LPPGWEERTHTD-GRVFFINHNIKKTQWEDP
>classI_31_s26.285308837890625
LPPGWEKRTDSN-GRVYFVNHNTRITQWEDP
>1to1_0_s19.71026611328125
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to1_2_s16.1439208984375
LPPGWERRVDPL-GRTYYVDHNTRTTTWTRP
>1to1_5_s15.9388427734375
LPPGWERRVDPN-GRTYYVDHNTRTTTWTRP
>1to1_7_s18.679412841796875
LPPGWERRVDPN-GRVYYVDHNTRTTTWTDP
>1to1_10_s19.027801513671875
LPPGWERRYTPN-GRVYFVDHNTRTTTWTDP
>1to1_12_s18.45770263671875
LPPGWEIRYTPE-GRVYFVDHNTRTTTWTDP
>1to1_15_s17.8856201171875
LPPGWEIRYTPE-GVRYFVDHNTRTTTFTDP
>1to1_18_s23.983245849609375
LPEGWEIRYTRE-GVRYFVDHNTRTTTFKDP
>1to2_3_s30.1531982421875
LPAGWEMAKTSS-GQRYFINHNTQTTTWQDP
>1to2_6_s25.78289794921875
LPAGWEMAKTSD-GERYFINHNTKTTTWQDP
>1to2_12_s22.609832763671875
LPPGWEEARTPD-GRVYFINHNTKTTTWQDP
>1to2_14_s22.72491455078125
LPPGWEEARTPD-GRVYYINHNTKTTTWEDP
>1to2_18_s25.520660400390625
LPPGWEEARAPD-GRTYYYNHNTKTTTWEKP
>1to2_22_s26.201904296875
LPPGWTEHKAPD-GRTYYYNHNTKTSTWEKP
>1to2_26_s26.1959228515625
LPSGWTEHKAPD-GRTYYYNTETKQSTWEKP
>1to2_30_s52.654571533203125
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to4_1_s16.872100830078125
LPPGWERRVDSL-GRTYYVDHNTRTTTWTRP
>1to4_3_s15.768341064453125
LPPGWERRVDPL-GRTYYVDHNTRTTTWQRP
>1to4_6_s15.98455810546875
LPPGWERRVDPN-GRTYYVDHNTRTTTWERP
>1to4_8_s17.68353271484375
LPPGWEERVDPN-GRTYYVDHNTRTTQWERP

>1to4_11_s18.824005126953125
LPPGWEERVDPS-GRTYYVNHITRTTQWERP
>1to4_13_s19.622283935546875
LPPGWEERVDRS-GRVYYVNHITRTTQWERP
>1to4_16_s22.94757080078125
LPPGWEKRMSRS-GRVYYVNHITRTTQWERP
>1to4_19_s28.957672119140625
LPPGWEKRMSRSSGRVYYVNHITRASQWERP

E.1.0.3 3rd batch

Parameters: $\Lambda = 0.1$, $\beta = 3$.

>1to4(rep2)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to4(rep2)_4_15.553558
LPPGWERRVDPR-GRTYYVDHNTRTTTWQRP
>1to4(rep2)_8_17.2417
LPPGWEERVDPS-GRTYYVDHNTRTTTWERP
>1to4(rep2)_11_19.035675
LPPGWEERVDRS-GRTYYVNHNTRTTQWERP
>1to4(rep2)_13_19.63678
LPPGWEERVDRS-GRVYYVNHITRTTQWERP
>1to4(rep2)_15_21.587952
LPPGWEKRVSRS-GRVYYVNHITRTTQWERP
>1to4(rep2)_18_27.183807
LPPGWEKRMSRS-GRVYYVNHITRASQWERP
>1to4(rep2)_21_32.32129(typeIV wt)
LPPGWEKRMSRSSGRVYYFNHITNASQWERP
>1to2(rep1)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to2(rep1)_6_15.608826
LPPGWERRVDPN-GRTYYVDHNTRTTTWQRP
>1to2(rep1)_10_17.344177
LPPGWEERKDPN-GRTYYVDHNTRTTTWERP
>1to2(rep1)_13_21.524017
LPPGWEERKDPD-GRTYYVNHNTKTTTWERP
>1to2(rep1)_16_27.44638
LPPGWEEHKSPD-GRTYYVNHNTKTTTWEKP
>1to2(rep1)_19_28.134277
LPPGWTEHKSPD-GRTYYYNHNTKTSTWEKP
>1to2(rep1)_22_26.252502
LPPGWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(rep1)_26_52.654571(typeII wt)
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(rep2)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to2(rep2)_4_15.782837
LPPGWERRVDPL-GRTYYVDHNTRTTTWQRP
>1to2(rep2)_8_17.634186
LPPGWEERVDPD-GRTYYVDHNTRTTTWERP

```
>1to2(rep2)_12_22.014587
LPPGWEERKAPD-GRTYYVNHNTKTTTWERP
>1to2(rep2)_16_25.869446
LPPGWTEHKAPD-GRTYYYNHNTKTTTWEKP
>1to2(rep2)_20_24.72174
LPPGWTEHKAPD-GRTYYYNTETKQSTWEKP
>1to2(rep2)_24_29.899872
-KSGWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(rep2)_26_52.654571(typeII wt)
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to4(dir)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to4(dir)_3_19.841064
LPPGWERRMDSS-GRTYYVDHNTRTTTWERP
>1to4(dir)_5_21.93518
LPPGWERRMDRS-GRTYYVDHNTRTTQWERP
>1to4(dir)_7_22.47174
LPPGWERRMDRS-GRVYYVNHNTRTTQWERP
>1to4(dir)_9_22.904633
LPPGWEKRMDRS-GRVYYVNHITRTTQWERP
>1to4(dir)_11_25.149902
LPPGWEKRMSRS-GRVYYVNHITRTSQWERP
>1to4(dir)_13_28.972168
LPPGWEKRMSRSSGRVYYVNHITRASQWERP
>1to4(dir)_15_32.32129(typeIV wt)
LPPGWEKRMSRSSGRVYYFNHITNASQWERP
>1to2(dir)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to2(dir)_3_17.419312
LPPGWERRKDPL-GRTYYVDHNTRTTTWERP
>1to2(dir)_6_19.964874
LPPGWEERKDPD-GRTYYVNHNTRTTTWERP
>1to2(dir)_9_25.546936
LPPGWEERKSPD-GRTYYVNHNTKTTTWEKP
>1to2(dir)_12_28.469604
LPPGWTEHKSPD-GRTYYVNHNTKTSTWEKP
>1to2(dir)_15_26.825195
LPPGWTEHKSPD-GRTYYYNTNTKQSTWEKP
>1to2(dir)_18_32.14113
LKSGWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(dir)_20_52.654571(typeII wt)
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to4(mfevo)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to4(mfevo)_1_19.724762
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to4(mfevo)_2_19.306091
LPPGWERRMDSL-GRTYYVDHNTRTTTWTRP
>1to4(mfevo)_3_19.162354
```

LPPGWERRMDSL-GRTYYVDHNTRTTTWERP
>1to4(mfevo)_4_19.162354
LPPGWERRMDSL-GRTYYVDHNTRTTTWERP
>1to4(mfevo)_5_19.162354
LPPGWERRMDSL-GRTYYVDHNTRTTTWERP
>1to4(mfevo)_6_19.162354
LPPGWERRMDSL-GRTYYVDHNTRTTTWERP
>1to4(mfevo)_7_32.32129(typeIV wt)
LPPGWEKRMSRSSGRVYYFNHITNASQWERP
>1to4(mfcont)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to4(mfcont)_1_21.341278
LPPGWERRQDRL-GRTYYVNHNTRTTQWTRP
>1to4(mfcont)_2_20.144714
LPPGWERRQDRL-GRTYYVNHNTRTTQWERP
>1to4(mfcont)_3_20.144714
LPPGWERRQDRL-GRTYYVNHNTRTTQWERP
>1to4(mfcont)_4_21.639069
LPPGWEKRQDRS-GRVYYVNHITRTTQWERP
>1to4(mfcont)_5_22.962067
LPPGWEKRMSRS-GRVYYVNHITRTTQWERP
>1to4(mfcont)_6_27.183807
LPPGWEKRMSRS-GRVYYVNHITRASQWERP
>1to4(mfcont)_7_32.32129(typeIV wt)
LPPGWEKRMSRSSGRVYYFNHITNASQWERP
>1to2(mfcont)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to2(mfcont)_1_17.39331
LPPGWEERKDPL-GRTYYVDHNTRTTTWTRP
>1to2(mfcont)_2_19.71637
LPPGWEERKAPD-GRTYYVDHNTRTTTWERP
>1to2(mfcont)_3_22.014587
LPPGWEERKAPD-GRTYYVNHNTKTTTWERP
>1to2(mfcont)_4_25.869446
LPPGWTEHKAPD-GRTYYYNHNTKTTTWEKP
>1to2(mfcont)_5_26.16455
LPPGWTEHKAPD-GRTYYYNHNTKQSTWEKP
>1to2(mfcont)_6_27.7146
LPSGWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(mfcont)_7_52.64276(typeII wt)
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
>1to2(mfevo)_0_19.724762(typeI wt)
LPPGWERRADSL-GRTYYVDHNTRTTTWTRP
>1to2(mfevo)_1_17.27475
LPPGWERRKDPL-GRTYYVDHNTRTTTWTRP
>1to2(mfevo)_2_17.27475
LPPGWERRKDPL-GRTYYVDHNTRTTTWTRP
>1to2(mfevo)_3_17.27475
LPPGWERRKDPL-GRTYYVDHNTRTTTWTRP

```
>1to2(mfevo)_4_17.419312
LPPGWERRKDPL-GRTYYVDHNTRTTTWERP
>1to2(mfevo)_5_17.390045
LPPGWEERKDPL-GRTYYVDHNTRTTTWERP
>1to2(mfevo)_6_17.390045
LPPGWEERKDPL-GRTYYVDHNTRTTTWERP
>1to2(mfevo)_7_52.64276(typeII wt)
AKSMWTEHKSPD-GRTYYYNTETKQSTWEKP
```

F

Appendix to Chapter 11

6.1. Training hyperparameters for RBM

List of hyperparameters defined to train the RBM:

- number of hidden units, $M = 200$
- learning rate $= 0.005$
- batch size $= 500$
- decay of learning rate, $= 0.33$
- L_1^2 regularization over weights, $l1b = 0.8$
- L_2 regularization over fields, $l2 = 0.0001$
- number of MonteCarlo step for contrastive divergence, $N_{MC} = 4$
- number of epochs $= 1000$

6.2. Michaelis–Menten kinetics

The Michaelis–Menten kinetics is the simplest model of enzyme kinetics [Michaelis et al. (1913)]. It considers a single substrate S and a single enzyme E that can bind to form a comple ES and then dissociate to form the product P and the free enzyme E:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} E + P \tag{F.1}$$

In this context, 5 assumptions can be made:

1. Only the intial velocity of the reaction is monitored.
2. There is no product at the beginning of the reaction. Thus, we can ignore the reaction $E + P \rightarrow ES$ (equivalent to set $k_{-2} = 0$) since we are only looking at the initial reaction rates.
3. Briggs-Haldane assumption: the rate of formation of the complex ES is equivalent to the rate of its dissociation.
4. The concentration of the enzyme is much lower than the concentration of the substrate. Thus, the concentration of the enzyme is constant and equal to its initial concentration, namely $[E]_{tot}$. This assumption also necessitate assumption 1.

5. The enzyme is either free or in complex with the substrate. Hence, $[E]_{tot} = [E] + [ES]$, where $[\cdot]$ indicates the molar concentration.

Starting from these assumptions we can write:

$$\frac{d[ES]}{dt} = k_1[E][S] - k_{-1}[ES] - k_2[ES] = 0\,, \tag{F.2}$$

where we used assumptions 2 and 3. By solving the last equation for $[ES]$, we obtain

$$[ES] = \frac{[E]_{tot}[S]}{[S] + K_m}\,, \tag{F.3}$$

where we have defined the Michaelis constant $K_m \equiv (k_{-1} + k_2)/k_1$. By renaming $k_2 \equiv k_{cat}$, we can now write the rate of formation of the product P as:

$$\frac{d[P]}{dt} = k_{cat}[ES] = \frac{k_{cat}[E]_{tot}[S]}{[S] + K_m}\,. \tag{F.4}$$

In particular, for small $[S]$ we have:

$$\frac{d[P]}{dt} = \frac{k_{cat}}{K_m}[E]_{tot}[S]\,, \tag{F.5}$$

where k_{cat}/K_m is called the specificity constant (or catalytic efficiency) and measures how efficiently an enzyme converts substrates into products.

6.3. List of tested sequences

We report here the list of the sequences that are going to be tested. Notice that the gaps from the alignements have been removed.

F.3.1 1st batch

>PE68.06
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINSQWVVSAAHCYKSRIQVRLGEHNIA
VLEGTEQFINSAKVIRHPNYNSYTLDNDIMLIKLSSPATLNSYVSTVSLPTSCAAAG
TQCLISGWGNTLSSGSNYPDLLQCLDAPILSDADCRNSYPGQITSNMFCVGFLEGG
KDSCQGDSGGPVVCNGELQGIVSWGYGCAQKNKPGVYTKVCNYVSWIQDTIAA
N
>PE95.13
IVGGYECVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIQ
VLEGNEQFINAAKIIRHPKFNSKTIDNDIMLIKLATPATLNSYVSPVALPTSCAAAG
TQCLISGWGNTLSSGANYPDLLQCLDAPILSNADCKKSYPGKITDNMICAGFLEGG
KDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGYPGVYTKVCNYVSWIQQTIAA
N
>PE145.07
IVGGYECVPNSVPYQVSLNDGYHFCGGSLINEQWVVSAAHCYKSSIQVVLGEHNI
QVLEGNEQFIKAAKIIKHPKFNSKTIDNDIMLIKLATPATFNSYVSPVCLPSSDSPG
TLCLISGWGNTLSSGANYPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGFLEG
GKDSCQGDSGGPVVCNGTLQGIVSWGYGCAQPGYPGVYTKVCNYVPWIQQTIA
AN
>PE172.44
IVGGEEAVPNSWPWQVSLQDTGFHFCGGSLINEQWVVSAAHCYVSSIQVVLGEH

NIQVLEGNEQVIKAAKIIKHPKFNSKTIDNDITLIKLATPATFNSYVSPVCLPSASDS
FPAGTLCVITGWGNTKSSGANYPDLLQCVALPLLSNADCKKSYGGKITDNMICAG
FLEGGKDSCQGDSGGPLVCQGGAWTLVGIVSWGYGCAQPGYPGVYTRVTNYVP
WIQQTIAAN
>PE142.64
IVGGEEAVPNSWPWQVSLQDTGFHFCGGSLINEQWVVSAAHCGVSSIQVVLGEH
NIQSLEGNEQVIKIAKIIKHPKFNSKTIDNDITLIKLATPATFNSYVSPVCLPSASDSF
PAGTLCVTTGWGKTKYNGANTPDLLQQAALPLLSNADCKKSWGSKITDNMICA
GASGVSSCMGDSGGPLVCQKGGAWTLVGIVSWGSSTCSTSYPGVYTRVTELVPW
IQQTIAAN
>PE91.82
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVSSHRVVLGE
HDRQSSEENIQVLKIAKVFKHPKFNSFTINNDITLIKLATPARFSSTVSPVCLPSASD
DFPAGTLCVTTGWGKTKYNAAKTPDKLQQAALPLLSNADCKKYWGSKITDVMI
CAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTELV
PWVQQTIAAN
>PE83.41
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVRSHRVVLGE
HDRQSDEENIQVLKIAKVFKHPKFNSFTINNDITLLKLATPARFSQTVSPVCLPSAS
DDFPAGTLCVTTGWGKTKYNANKTPDKLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LRPWVDQILAAN
>PE79.47
IVNGEDAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVRTSDVVVAG
EFDQSDEEDIQVLKIAKVFKNPKFNMFTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTLCATTGWGKTKYNANKTPDKLQQAALPLLSNADCKKFWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LRPWVDQILAAN
>PE92.64
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVRTSDVVVAG
EFDQGSSSEDIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTLCVTTGWGLTKYTAANTPDKLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LVPWVQQTLAAN
>PE110.75
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVTTSDVVVAG
EFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPSA
SDDFPAGTTCVTTGWGLTRYTAANTPDRLQQAALPLLSNTDCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVNWVQQTLAAN
>TL80.637
IVGGYTCQENSVPYQVSLNSGGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINAAKIIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDADCRAAYPGQITSNMFCVGFL
EGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVDWIQE
TIAAN
>TL80.635
IVGGYTCQENSVPYQVSLNSQGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINAAKIIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPTSCAA

AGTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDAECEAAYPGQITSNMICVGFL
EGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQET
IAAN
>TL80.634
IVGGYTCAENSVPYQVSLNSGGYHFCGGSLINDQWVVSAAHCYKSGRIQVRLGEH
NIEVLEGTEQFINAAKIIRHPNYNSWTLDNDIMLIKLSTPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGSNYPDLLQCLDAPVLSDAECRAAYPGQITSNMFCVGFL
EGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQQT
IAAN
>TL80.473
IVGGYTCQENSVPYQVSLNSGGYHFCGGSLINSQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINAAKIIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDADCEAAYPGQITSNMFCLGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQET
IAAN
>TL80.439
IVGGYTCQENSVPYQVSLNSGGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGNEQFINAAKIIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGSNYPDLLQCLDAPILSDAECRAAYPGQITSNMICVGFLE
GGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQETI
AAN
>TL80.424
IVGGYTCQENSVPYQVSLNSKGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINSAKIIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGSNYPDLLQCLDAPVLSDADCRASYPGQITSNMFCLGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQETI
AAN
>TL80.375
IVGGYTCTKNSVPYQVSLNSGGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIAVLEGTEQFINAAKIIRHPSYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPTSCAA
AGTQCLISGWGNTLSSGSNYPDLLQCLDAPVLSDADCRAAYPGQITSNMFCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQDT
IAAN
>TL80.319
IVGGYTCQENSVPYQVSLNSGGYHFCGGSLISDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINAAKIIRHPSYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCAA
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDAECEAAYPGQITSNMFCLGFL
EGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQDT
IAAN
>TL80.285
IVGGYTCQENSVPYQVSLNSGGYHFCGGSLINDQWVVSAAHCYKSSRIQVRLGEH
NIAVLEGTEQFINSAKIIRHPNYNSYTLDNDIMLIKLSSPATLNSRVSTVSLPTSCAA
AGTQCLISGWGNTLSSGSNYPDLLQCLDAPVLSDAECEAAYPGQITSNMFCVGFL
EGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQET
IAAN
>TL79.959
IVGGYTCQKNSVPYQVSLNSGGYHFCGGSLISDQWVVSAAHCYKSSRIQVRLGEH
NIEVLEGTEQFINAAKVIRHPNYNSWTLDNDIMLIKLSSPATLNSRVSTVSLPSSCA
AAGTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDADCRAAYPGQITSNMFCLGF

LEGGKDSCQGDSGGPVVCNGQLQGIVSWGYGCAQKGKPGVYTKVCNYVSWIQS
TIAAN
>TE87.17
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSQADCEASYPGKITSNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>TE86.04
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLDNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKNKPGVYTKVCNYVDWIQDTI
AAN
>TE93.28
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
AVLEGNEQFINAAKIIRHPNFDSKTLNNDIMLIKLSSPATLNARVATVSLPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSDADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>TE88.21
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLDNDIMLIKLSSPATLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>TE88.26
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLDNDIMLIKLSSPVTLNSRVATVSLPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKNKPGVYTKVCNYVDWIQDTI
AAN
>TE93.56
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDSKTLDNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLNAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>TE86.80
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSWTLDNDIMLIKLSSPAKLNSRVATVSLPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>TE91.13
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYDSKTLDNDIMLIKLSSPATLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITSNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI

AAN
>TE88.05
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLDNDIMLIKLSSPATLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPVLSQADCEASYPGKITSNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGYPGVYTKVCNYVDWIQDTI
AAN
>TE85.22
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCENSYPGKITDNMFCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>CE101.16
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVTTSDVVVAG
EFDQGSSSEDIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTTCVTTGWGLTRYTAANTPDRLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LVPWVQQTLAAN
>CE103.59
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVTTSDVVVAG
EFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTTCVTTGWGLTRYTAANTPDKLQQAALPLLSNAECKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVNWVQQTLAAN
>CE103.34
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVTTSDVVVA
GEFDQGSDSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPS
ASDDFPAGTLCVTTGWGLTKYTAAKTPDRLQQAALPLLSNADCKKYWGSKITD
VMICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVT
ALVNWVQQTLAAN
>CE101.13
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVTTSDVVVAG
EFDQGSSSESIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPSAS
DDFPAGTTCVTTGWGLTRYNAANTPDKLQQAALPLLSNADCKKYWGSKITDVM
ICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTSLR
NWVQQTLAAN
>CE101.93
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVRTSDVVVAG
EFDQGSSSEVIQVLKIAKVFKNPKYNSLTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTTCVTTGWGLTKYTAANTPDKLQQAALPLLSNTDCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVPWVQQTLAAN
>CE103.94
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVRTSDVVVA
GEFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLSTPASFSQTVSAVCLPS
ASDDFPAGTLCVTTGWGLTRYNAANTPDTLQQAALPLLSNADCKKYWGSKITD
VMICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVT
ELVSWVQQTLAAN

>CE107.07
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLISENWVVTAAHCGVTTSDVVVA
GEFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPS
ASDDFPAGTLCVTTGWGLTRYTAANTPDRLQQAALPLLSNTQCKKYWGSKITD
VMICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVT
ELVPWVQQTLAAN
>CE107.59
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVTTSDVVVAG
EFDQGSSSENIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPSAS
DDFPAGTTCVTTGWGLTRYNAANTPDRLQQAALPLLSNTDCKKYWGSKITDAM
ICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTAL
VPWVQQTLAAN
>CE105.28
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVTTSDVVVAG
EFDQGSSSEEIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPSAS
DDFPAGTTCVTTGWGLTKYTAANTPDRLQQAALPLLSNADCKKYWGSKITDVM
ICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTAL
VNWVQQTLAAN
>CE101.97
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVRTSDVVVAG
EFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPSA
SDDFPAGTLCVTTGWGLTKYTASNTPDRLQQAALPLLSNTDCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVPWVQQTLAAN

F.3.2 2nd batch

>TE85.22D189S
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCENSYPGKITDNMFCVGFLE
GGKSSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>TE85.22
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCENSYPGKITDNMFCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>1T8O
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVTTSDVVVA
GEFDQGSSSEKIQKLKIAKVFKNSKYNSLTINNDITLLKLSTAASFSQTVSAVCLPSA
SDDFAAGTTCVTTGWGLTRYTNANTPDRLQQASLPLLSNTNCKKYWGTKIKDA
MICAGASGVSSCMGDSGGPLVCKKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVNWVQQTLAAN
>3TGI
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFVNAAKIIKHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLPQADCEASYPGKITDNMVCVGFL

EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDNPGVYTKVCNYVDWIQDT
IAAN
>sT85C_0_120.671
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_1_119.286
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTQCLVSGWGNTLSPGVNYPDTLQCLDIPILSDEECKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_2_119.662
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKIIRHPKYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_3_123.093
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKIIRHPPYNSKTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTTCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_4_121.38
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKIIRHPPYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_5_120.222
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
FLEGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQ
ETMAAN
>sT85C_6_117.858
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPKYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sT85C_7_119.457
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPPYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET

MAAN
>sT85C_8_122.051
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPPYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYLDWIQET
MAAN
>sT85C_9_119.276
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPPYNSYTIDNDIMLIKLSRPATLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLNIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKSSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>pT85C_10_89.125
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
AVLEGNEQFINAAKVIRHPKYNSKTLNNDIMLIKLSSPATLNSRVSTVALPSSCAAA
GTQCLISGWGNTLSSGANYPDLLQCLDAPLLSDADCKNSYPGKITDNMFCVGFLE
GGKSSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQQTI
AAN
>pT85C_25_133.123
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QGVLEGNEQFINAAKVIRHPKYNSKTLNNDIMLIKLSTPATLNSRVSTVCALPSAS
DSCAAPGTQCLISGWGNTLSTGANYPDLLQCLDAPLLSNADCKKSYPGKITDNMI
CVGFLEGGKSSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVD
WIQQTIAAN
>pT85C_40_152.37
IVGGYTCVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCYKSRIQVVLGEH
NIQGVLEGNEQVINAAKVIKNPKYNSKTLNNDIMLIKLSTPATLNSRVSTVCALPS
ASDSFPAGTQCLISGWGNTLSTGANYPDLLQCLDAPLLSNADCKKSYPGKITDNM
ICVGFLEGGKSSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVD
WIQQTIAAN
>pT85C_55_176.694
IVGGETAVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCYVSRIQVVLGEH
NIQGVLEGNEQVINAAKVIKNPKYNSKTLNNDIMLIKLSTPATLNSRVSTVCALPS
ASDSFPAGTQCLISGWGNTLSTGANYPDLLQCLDAPLLSNADCKKSYGGKITDNM
ICVGFLEGGKSSCQGDSGGPLVCQKNGAWTLVGIVSWGYGCALKGKPGVYTRV
TNYVDWIQQTIAAN
>pT85C_70_177.467
IVNGETAVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCYVSRIQVVLGEH
NIQGVLEGNEQVINAAKVIKNPKYNSKTLNNDIMLIKLSTPATLNSTVSTVCALPS
ASDSFPAGTQCVTSGWGNTLSTGANYPDLLQCLAAPLLSNADCKKSWGGKITDN
MICVGASGVSSCQGDSGGPLVCQKNGAWTLVGIVSWGSSGCATKGKPGVYTRVT
NYVDWIQQTIAAN
>pT85C_85_168.141
IVNGETAVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCYVSTIQVVLGEH
NIQGSLEGNEQVINIAKVFKNPKYNSKTINNDIMLIKLSTPATLNSTVSTVCALPSA
SDSFPAGTLCVTSGWGLTLSTGANTPDLLQQAALPLLSNADCKKSWGGKITDNMI
CAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSGCSTSGKPGVYTRVTNY
VDWIQQTIAAN

>pT85C_100_142.437
IVNGETAVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCGVSTIQVVLGEH
NIQGSSEENIQVLNIAKVFKNPKYNSKTINNDITLIKLATPATLNSTVSTVCALPSAS
DSFPAGTLCVTSGWGLTLSTAANTPDLLQQAALPLLSNADCKKSWGSKITDVMIC
AGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYTRVTKLVD
WIQQTIAAN
>pT85C_115_113.596
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINEQWVVSAAHCGVSTHQVVLGE
HNIQGSSEENIQVLKIAKVFKNPKYNSKTINNDITLIKLATPATLNSTVSPVCLPSAS
DSFPAGTLCVTTGWGLTRYTAANTPDKLQQAALPLLSNADCKKYWGSKITDVMI
CAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTKLV
PWIQQTIAAN
>pT85C_130_95.114
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVSTHTVVLGE
HDIQGSSEENIQVLKIAKVFKNPKYNSFTINNDITLLKLATPARLSSTVSPVCLPSAS
DDFPAGTLCVTTGWGLTRYNAAKTPDKLQQAALPLLSNADCKKYWGSKITDVM
ICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTELR
PWVQQTLAAN
>pT85C_145_88.028
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVRTSDVVVAG
EFDQGSDEEDIQVLKIAKVFKNPKYNSFTINNDITLLKLATPARFSQTVSPVCLPSA
SDDFPAGTLCVTTGWGLTKYNAAKTPDKLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LRPWVQQTLAAN
>pT85C_160_106.793
IVNGEEAVPGSWPWQVSLQDSTGFHFCGGSLINENWVVTAAHCGVTTSDVVVAG
EFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLATPASFSQTVSAVCLPSA
SDDFPAGTTCVTTGWGLTRYTAANTPDRLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVPWVQQTLAAN
>pTT85_1_129.184
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFVNAAKIIKHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLPQADCEASYPGKITDNMVCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDNPGVYTKVCNYVDWIQDT
IAAN
>pTT85_2_124.596
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIKHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLPQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDNPGVYTKVCNYVDWIQDTI
AAN
>pTT85_3_120.761
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIKHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLPQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_4_116.607

IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIKHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_5_112.844
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDRKTLNNDIMLIKLSSPVKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_6_109.514
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDRKTLNNDIMLIKLSSPAKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_7_106.24
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDSKTLNNDIMLIKLSSPAKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_8_102.876
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDSKTLNNDIMLIKLSSPAKLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_9_100.057
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFDSKTLNNDIMLIKLSSPATLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_10_97.635
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFNSKTLNNDIMLIKLSSPATLNARVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_11_95.726
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNFNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_12_93.698
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI

NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKDKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_13_91.085
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_14_89.073
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCEASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_15_87.851
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVATVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCEASYPGKITDNMFCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>pTT85_16_86.023
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFINAAKIIRHPNYNSKTLNNDIMLIKLSSPATLNSRVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSDADCEASYPGKITDNMFCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALKGKPGVYTKVCNYVDWIQDTI
AAN
>pTCd_10_118.815
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKVIKHPNFNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDNPGVYTKVCNYVDWIQDT
IAAN
>pTCd_17_119.724
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKFNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDNPGVYTKVCNYVDWIQQT
IAAN
>pTCd_30_137.206
IVNGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLKAAKVIKHPKFNSLTLNNDIMLIKLSTPAKLNATVSTVALPSSCAP
AGTTCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGFL
EGGKDSCQGDSGGPVVCNGTLQGIVSWGYGCALSDTPGVYTKVCNYVDWIQQT
IAAN
>pTCd_40_158.132
IVNGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLKAAKVIKHPKFNSLTLNNDIMLIKLSTPAKLNATVSTVALPSSCAP

AGTTCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGAS
GGSSCQGDSGGPVVCNGTLQGIVSWGSSGCATSDTPGVYTRVCNYVDWIQQTIA
AN
>pTCd_50_166.798
IVNGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVVLGEHNI
QVSEGNEQFLKAAKVIKHPKFNSLTLNNDIMLIKLSTPAKLNATVSTVALPSSCAP
AGTTCVTSGWGNTLSTGVNTPDLLQCLDLPLLSNADCKKSWGGKITDNMICAGA
SGVSSCMGDSGGPVVCNGTLQGIVSWGSSGCATSTPGVYTRVTNYVDWIQQTIA
AN
>pTCd_100_153.615
IVNGETAVPNSWPWQVSLQDKTGFHFCGGSLINEQWVVSAAHCGVSTIQVVLGE
HNIQSSEENIQFLKIAKVFKHPKFNSLTINNDITLIKLSTPAKLNATVSAVCLPSASD
SFAAGTTCVTTGWGLTLSTGANTPDLLQQADLPLLSNADCKKYWGGKITDNMIC
AGASGVSSCMGDSGGPLVCKKNGAWTLVGIVSWGSSTCSTSTPGVYARVTNLVD
WIQQTIAAN
>pTCd_120_140.188
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVSTSQVVVA
GEFDQGSSEENIQFLKIAKVFKNPKFNSLTINNDITLIKLSTPAKLSQTVSAVCLPSA
SDDFAAGTTCVTTGWGLTRYTGANTPDLLQQADLPLLSNADCKKYWGGKITDN
MICAGASGVSSCMGDSGGPLVCKKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVDWVQQTIAAN
>pTCd_130_138.337
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVTTSDVVVA
GEFDQGSSEENIQFLKIAKVFKNPKFNSLTINNDITLLKLSTPASFSQTVSAVCLPSA
SDDFAAGTTCVTTGWGLTRYTGANTPDRLQQASLPLLSNADCKKYWGGKITDA
MICAGASGVSSCMGDSGGPLVCKKNGAWTLVGIVSWGSSTCSTSTPGVYARVTA
LVNWVQQTLAAN
>pTCg_18_107.104
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
QVLEGTEQFLNAAKVIKHPNFNSKTLNNDIMLIKLSSPAKLNAYVSTVALPSSCAP
AGTQCLISGWGNTLSSGANYPDLLQCLDAPLLSNADCKASYPGKITDNMFCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVSWIQQTI
AAN
>pTCg_37_131.874
IVNGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QGVLEGTEQFLKAAKVIKHPKFNSLTLNNDIMLIKLSTPASLNAYVSTVALPDSCA
PPGTTCLISGWGNTLSTGANYPDLLQCLDAPLLSNAQCKKSYPGKITDNMICAGF
LEGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVNWIQQ
TIAAN
>pTCg_56_172.711
IVNGYTCVPNSVPYQVSLQDSTGFHFCGGSLINEQWVVSAAHCYVSRIQVRLGEH
NIQGVLEGTEQVLKAAKVIKNPKFNSLTLNNDIMLIKLSTPASLNAYVSAVCALPS
ASDSFPGGTTCLISGWGNTLSTGANYPDLLQCLDAPLLSNAQCKKSYPGKITDNM
ICAGFLEGGKDSCQGDSGGPVVCNGTLQGIVSWGYGCALPDKPGVYTKVCNYV
NWIQQTIAAN
>pTCg_75_187.698
IVNGETAVPGSWPWQVSLQDSTGFHFCGGSLINEQWVVSAAHCYVSRIQVRLGE
HNIQGVLEGTEQVLKIAKVIKNPKFNSLTLNNDITLIKLSTPASLNAYVSAVCLPSA
SDSFPGGTTCLISGWGNTLSTGANYPDLLQCLDAPLLSNAQCKKSYGGKITDNMI

CAGFLEGGKDSCQGDSGGPLVCQKNGAWTLAGIVSWGYGCALPDKPGVYTRVT
NYVNWIQQTIAAN
>pTCg_94_177.274
IVNGETAVPGSWPWQVSLQDSTGFHFCGGSLINEQWVVSAAHCGVSTIQVILGEH
NIQGVLEGTEQVLKIAKVIKNPKFNSLTLNNDITLIKLSTPASLNAYVSAVCLPSAS
DSFPGGTTCVTTGWGNTLSTGANTPDLLQQLALPLLSNAQCKKSWGSKITDNMI
CAGASGVDSCQGDSGGPLVCQKNGAWTLAGIVSWGSGCALPDKPGVYTRVTNY
VNWIQQTIAAN
>pTCg_113_145.673
IVNGETAVPGSWPWQVSLQDSTGFHFCGGSLINEQWVVSAAHCGVSTIQVILGEH
NIQGSDEETIQVLKIAKVIKNPKFNSLTLNNDITLIKLSTPASLNATVSAVCLPSASD
SFPGGTTCVTTGWGLTLSTGANTPDLLQQAALPLLSNAQCKKSWGSKITDVMIC
AGASGVSSCMGDSGGPLVCQKNGAWTLAGIVSWGSSTCSTSTPGVYTRVTELVN
WIQQTIAAN
>pTCg_132_109.684
IVNGEDAVPGSWPWQVSLQDSTGFHFCGGSLISEQWVVTAAHCGVSTSQVVLGE
HNIQGSDEETIQVLKIAKVFKNPKFNSLTINNDITLIKLATPARLSATVSAVCLPSAS
DSFPAGTTCVTTGWGLTKYNAANTPDKLQQAALPLLSNAQCKKYWGSKITDVM
ICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVTELV
NWIQQTIAAN
>pTCg_151_89.169
IVNGEDAVPGSWPWQVSLQDSTGFHFCGGSLISENWVVTAAHCGVRTSRVVVAG
EFDQGSDEEDIQVLKIAKVFKNPKFNSLTINNDITLLKLATPARFSQTVSAVCLPSA
SDDFPAGTLCVTTGWGLTKYNAANTPDKLQQAALPLLSNADCKKYWGSKITDV
MICAGASGVSSCMGDSGGPLVCQKDGAWTLVGIVSWGSSTCSTSTPGVYARVTE
LVNWVQQTLAAN
>pTCg_170_116.681
IVNGEEAVPGSWPWQVSLQDKTGFHFCGGSLINENWVVTAAHCGVTTSDVVVA
GEFDQGSSSEKIQVLKIAKVFKNPKYNSLTINNDITLLKLSTPASFSQTVSAVCLPS
ASDDFPAGTTCVTTGWGLTRYTAANTPDRLQQAALPLLSNTDCKKYWGSKITD
VMICAGASGVSSCMGDSGGPLVCQKNGAWTLVGIVSWGSSTCSTSTPGVYARVT
ALVNWVQQTLAAN
>sCT_0_165.032
IVGGQDAPAGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGVSGCARPLPGVYTRVSQ
YQSWINSHIASN
>sCT_1_166.181
IVGGQDAPAGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYNSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFNGCARPLPGVYTRVS
QYQSWINSHIASN
>sCT_2_164.537
IVGGQDAPAGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYNSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGVNGCARPLPGVYTRVS

QYQSWINSHIASN
>sCT_3_167.153
IVGGQDAPAGSWPWQVSLQSRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVNFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHGTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFSGCARPLPGVYTRVSQ
YQSWINSHIASN
>sCT_4_166.085
IVGGQDAPAGSWPWQVSLQVRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVNFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHGTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFNGCARPLPGVYTRVS
QYQSWINSHIASN
>sCT_5_166.067
IVGGQDAPAGSWPWQVSLQVRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFNGCARPLPGVYTRVS
QYQSWINSHIASN
>sCT_6_165.741
IVGGQDAPPGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVNFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGSSGCARPLPGVYTRVSQ
YQSWINSHIASN
>sCT_7_167.921
IVGGQDAPPGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFNGCARPLPGVYTRVS
QYQSWINSHIAAN
>sCT_8_167.016
IVGGQDAPPGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGFNGCARPLPGVYTRVS
QYQSWINSHIASN
>sCT_9_165.601
IVGGQDAPPGSWPWQVSLQRRRGGHFCGGSLINDQWVLTAAHCFQSWLTVYLG
RQNLQPGLNPNEVSRTVAKIIVHPNYNSNTNNNDIALLKLSSPVTFTDYIRPVCLA
ASGSVFNNGTDSWVTGWGNIVEEGLPPPYTLQEVEVPVVGNRQCNCLYHSTITD
NMICAGVLAADSCQGDSGGPMVSKQGSVWIQSGIVSFGSSGCARPLPGVYTRVSQ
YQSWINSHIASN
>sTC_0_119.612
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVVEGTEQFINAAKVIRHPPYNSKTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEECKKAYPGKITDNMVCAGF
LEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYLDWIQET
MAAN

>sTC_1_115.551
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
FLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQ
ETMAAN
>sTC_2_118.394
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
FLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYLDWIQE
TMAAN
>sTC_3_117.814
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
FLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYVDWIQ
ETMAAN
>sTC_4_118.652
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
YLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYVDWIQ
ETMAAN
>sTC_5_115.858
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEECKKAYPGKITDNMVCAGF
LEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQE
TMAAN
>sTC_6_116.588
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLNIPILSDEDCKKAYPGKITDNMVCAG
FLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCKYVDWIQ
ETMAAN
>sTC_7_115.724
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINAAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLNIPILSDEECKKAYPGKITDNMVCAGF
LEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQE
TMAAN
>sTC_8_116.111
IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPKYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCA
QPGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAG
YLEGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCSYVDWIQ
ETMAAN
>sTC_9_114.844

IVGGYECVPHSQPWQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
WVWEGTEQFINSAKVIRHPPYNSYTIDNDIMLIKLSRPARLNQYVQPVPLPSRCAQ
PGTMCLVSGWGTTSSPGVNYPDTLQCLDIPILSDEDCKKAYPGKITDNMVCAGFL
EGGKDSCQGDSGGPLVCNGELQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_0_103.815
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VNEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_1_104.847
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VQEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_2_104.873
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VQEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDAECKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_3_105.813
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VQEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLSRPATLNQYVQPVALPSRCAQA
GTQCLISGWGNTLSPGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_4_107.419
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVALPSRCAQA
GTQCLISGWGNTLSPGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_5_105.562
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVALPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_6_107.451
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSPGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_7_105.581
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS

VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_8_105.613
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLARPATLNQYVQPVPLPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDAECKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>sPC_9_104.743
IVGGYECVPHSQPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNIS
VWEGTEQFINAAKIIRHPKYNSKTIDNDIMLIKLSRPATLNQYVQPVALPSRCAQA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPILSDADCKKAYPGKITDNMVCAGFLE
GGKDSCQGDSGGPLVCNGQLQGIVSWGYGCAQPNKPGVYTKVCNYVDWIQET
MAAN
>pTCg1_10_115.033
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKIIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMVCVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQDTI
AAN
>pTCg1_12_114.606
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKVIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMVCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQQT
IAAN
>pTCg1_14_114.769
IVGGYTCQENSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMVCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQQT
IAAN
>pTCg1_16_115.055
IVGGYTCVENSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMVCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQQT
IAAN
>pTCg1_18_115.625
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNEPDLLQCLDAPLLSNADCKASYPGKITDNMVCVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQQT
IAAN
>pTCg1_21_118.548
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLKAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP

AGTQCLISGWGNTLSSGVNTPDLLQCLDAPLLSNADCKASYPGKITDNMVCAGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYTKVCNYVDWIQQT
IAAN
>pTCg1_23_121.908
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLKAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKASYPGKITDNMVCAGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDYPGVYAKVCNYVDWIQQT
IAAN
>pTCg1_25_126.118
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLKAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPDSCA
PAGTQCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKASYPGKITDNMVCAG
FLEGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALSDYPGVYAKVCNYVDWIQ
QTIAAN
>pTCg1_27_130.743
IVNGYTCVPNSVPYQVSLNDGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLKAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPDSCA
PAGTQCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKASYPGKITDNMVCAG
FLEGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALSDYPGVYAKVCNYVDWIQ
QTIAAN
>pTCg1_30_138.933
IVNGYTCVPNSVPYQVSLNDGFHFCGGSLINEQWVVSAAHCYVSRIQVVLGEHNI
QVLEGNEQFLKAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPDSCA
PAGTQCLISGWGNTLSTGVNTPDLLQCLDAPLLSNADCKASYPGKITDNMVCAG
FLEGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALSDYPGVYAKVCNYVDWIQ
QTIAAN
>pTCg2_10_112.906
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKIIKHPNFNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNLPDLLQCLDAPLLSQADCEASYPGKITDNMICVGFLEG
GKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTIA
AN
>pTCg2_12_112.211
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKIIKHPNFNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNLPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTCg2_14_111.684
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKVIKHPNFNSKTLNNDIMLIKLSSPARLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNLPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTCg2_16_111.645
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKVIKHPNFNSKTLNNDIMLIKLSSPARLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE

GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTCg2_18_111.55
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKFNSKTLNNDIMLIKLSSPARLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTCg2_21_112.139
IVGGYTCVKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKFNSKTLNNDIMLIKLSTPARLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQTI
AAN
>pTCg2_23_114.158
IVGGYTCVKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKFNSKTLNNDIMLIKLSTPARLNARVSTVALPSSCAP
AGTTCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKKSYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQTI
AAN
>pTCg2_25_115.635
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKFNSKTLNNDIMLIKLSTPARLNARVSTVALPSSCAP
AGTTCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQTI
AAN
>pTCg2_27_118.311
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLKAAKVIKHPKFNSKTLNNDIMLIKLSTPARLNARVSTVALPSSCAP
AGTTCLISGWGNTLSSGANLPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGYL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg2_30_123.218
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLKAAKVIKHPKFNSKTLNNDIMLIKLSTPARLNARVSTVALPDSCAP
AGTTCLISGWGNTLSTGANLPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGYL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_10_107.656
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFVNAAKIIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSQADCEASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDT
IAAN
>pTCg3_12_107.24
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINDQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFVNAAKIIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDT

IAAN
>pTCg3_14_107.05
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKIIKHPNYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAPA
GTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFLE
GGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDTI
AAN
>pTCg3_16_106.864
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
NVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQDT
IAAN
>pTCg3_18_106.815
IVGGYTCQKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSSPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKASYPGKITDNMICVGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_21_108.235
IVGGYTCVKNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKASYPGKITDNMICAGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_23_109.918
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVLEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTQCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_25_112.202
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLNAAKVIKHPKYNSKTLNNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTTCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGFL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_27_114.597
IVGGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLNAAKVIKHPKYNSKTINNDIMLIKLSTPAKLNARVSTVALPSSCAP
AGTTCLISGWGNTLSSGVNYPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGYL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN
>pTCg3_30_121.036
IVNGYTCVPNSVPYQVSLNSGYHFCGGSLINEQWVVSAAHCYKSRIQVRLGEHNI
QVSEGNEQFLNAAKVIKHPKYNSKTINNDIMLIKLSTPAKLNATVSTVALPSSCAP
AGTTCLISGWGNTLSTGVNYPDLLQCLDAPLLSNADCKKSYPGKITDNMICAGYL
EGGKDSCQGDSGGPVVCNGELQGIVSWGYGCALPDKPGVYTKVCNYVDWIQQT
IAAN

Bibliography

1. Agoritsas, E., Catania, G., Decelle, A., and Seoane, B. (2023). Explaining the effects of non-convergent sampling in the training of energy-based models. *arXiv preprint arXiv:2301.09428*.

2. Anfinsen, C. B. (1973). Principles that Govern the Folding of Protein Chains. *Science*, 181(4096):223–230.

3. Appel, W. (1986). Chymotrypsin: molecular and catalytic properties. *Clinical biochemistry*, 19(6):317–322.

4. Azer, E. S., Ebrahimabadi, M. H., Malikić, S., Khardon, R., and Sahinalp, S. C. (2020). Tumor phylogeny topology inference via deep learning. *Iscience*, 23(11):101655.

5. Baake, E., Baake, M., and Wagner, H. (1997). Ising quantum chain is equivalent to a model of biological evolution. *Physical Review Letters*, 78(3):559.

6. Baldassi, C., Zamparo, M., Feinauer, C., Procaccini, A., Zecchina, R., Weigt, M., and Pagnani, A. (2014). Fast and accurate multivariate gaussian modeling of protein families: Predicting residue contacts and protein-interaction partners. *PLoS One*, 9(3):e92721.

7. Barton, J. P., De Leonardis, E., Coucke, A., and Cocco, S. (2016). Ace: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics*, 32(20):3089–3097.

8. Barton, N. H. and Turelli, M. (1991). Natural and sexual selection on many loci. *Genetics*, 127(1):229–255.

9. Beissinger, M. and Buchner, J. (1998). How chaperones fold proteins. *Biological chemistry*, 379(3):245–259.

10. Bengio, Y. and Delalleau, O. (2009). Justifying and generalizing contrastive divergence. *Neural computation*, 21(6):1601–1621.

11. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, 28(1):235–242.

12. Blythe, R. A. and McKane, A. J. (2007). Stochastic models of evolution in genetics, ecology and linguistics. *J. Stat. Mech.: Theory Exp.*, 2007(07):P07018.

13. Bolhuis, P. G., Chandler, D., Dellago, C., and Geissler, P. L. (2002). Transition path sampling: Throwing ropes over rough mountain passes, in the dark. *Annual review of physical chemistry*, 53(1):291–318.

14. Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs. Available online at: `http://github.com/google/jax`.

15. Brandsdal, B. O., Österberg, F., Almlöf, M., Feierberg, I., Luzhkov, V. B., and Åqvist, J. (2003). Free Energy Calculations and Ligand Binding. In *Advances in Protein Chemistry*, volume 66 of *Protein Simulations*, pages 123–158. Academic Press.

16. Bray, A. J. (2002). Theory of phase-ordering kinetics. *Advances in Physics*, 51(2):481–587.

17. Burger, L. and Van Nimwegen, E. (2008). Accurate prediction of protein–protein interactions from sequence alignments using a bayesian method. *Molecular systems biology*, 4(1):165.

18. Burger, L. and van Nimwegen, E. (2010). Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS Comput. Biol.*, 6(1):e1000633.

19. Butterfield, D. A., Abdul, H. M., Opii, W., Newman, S. F., Joshi, G., Ansari, M. A., and Sultana, R. (2006). Review: Pin1 in alzheimer's disease. *Journal of Neurochemistry*, 98(6):1697–1706.

20. Camin, J. H. and Sokal, R. R. (1965). A method for deducing branching sequences in phylogeny. *Evolution*, pages 311–326.

21. Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., and Weigt, M. (2018). Inverse statistical physics of protein sequences: a key issues review. *Reports on Progress in Physics*, 81(3):032601.

22. Cocco, S. and Monasson, R. (2012). Adaptive cluster expansion for the inverse ising problem: convergence, algorithm and tests. *Journal of Statistical Physics*, 147(2):252–314.

23. Craik, C. S., Largman, C., Fletcher, T., Roczniak, S., Barr, P. J., Fletterick, R., and Rutter, W. J. (1985). Redesigning trypsin: alteration of substrate specificity. *Science*, 228(4697):291–297.

24. Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. (2022). Robust deep learning–based protein sequence design using proteinmpnn. *Science*, page eadd2187.

25. Dean, S. N. and Walper, S. A. (2020). Variational autoencoder for generation of antimicrobial peptides. *ACS omega*, 5(33):20746–20754.

26. Decelle, A., Rosset, L., and Seoane, B. (2023). Unsupervised hierarchical clustering using the learning dynamics of rbms. *arXiv preprint arXiv:2302.01851*.

27. Delgado, J., Radusky, L. G., Cianferoni, D., and Serrano, L. (2019). FoldX 5.0: Working with RNA, small molecules and a new graphical interface. *Bioinformatics*, 35(20):4168–4169.

28. Dichio, V. (2020). Statistical Genetics and DCA Inference beyond the Quasi Linkage Equilibrium. Master Thesis, University of Trieste, Italy.

29. Dichio, V., Zeng, H.-L., and Aurell, E. (2021). Statistical genetics within and beyond the Quasi-Linkage Equilibrium. in preparation.

30. Dichio, V., Zeng, H.-L., and Aurell, E. (2023). Statistical genetics in and out of quasi-linkage equilibrium. *Reports on Progress in Physics*, 86(5):052601.

31. Dobson, C. M. (2003). Protein folding and misfolding. *Nature*, 426(6968):884–890.

32. Du, Z., Su, H., Wang, W., Ye, L., Wei, H., Peng, Z., Anishchenko, I., Baker, D., and Yang, J. (2021a). The trRosetta server for fast and accurate protein structure prediction. *Nature protocols*, 16(12):5634–5651.

33. Du, Z., Su, H., Wang, W., Ye, L., Wei, H., Peng, Z., Anishchenko, I., Baker, D., and Yang, J. (2021b). The trRosetta server for fast and accurate protein structure prediction. *Nature Protocols*, 16(12):5634–5651.

34. Dunn, S., Wahl, L., and Gloor, G. (2008). Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340.

35. Edgar, R. C. (2004). Muscle: multiple sequence alignment with improved accuracy and speed. In *Proceedings. 2004 IEEE Computational Systems Bioinformatics Conference, 2004. CSB 2004.*, pages 728–729. IEEE.

36. Edwards, A. W. (1963). Reconstruction of evolution. In *Heredity*, page 553. BLACKWELL SCIENCE LTD PO BOX 88, OSNEY MEAD, OXFORD OX2 0NE, OXON, ENGLAND.

37. Eigen, M. (1971). Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523.

38. Ekeberg, M., Hartonen, T., and Aurell, E. (2014). Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *J. Comput. Phys.*, 276:341–356.

39. Espanel, X. and Sudol, M. (1999). A single point mutation in a group i ww domain shifts its specificity to that of group ii ww domains. *Journal of Biological Chemistry*, 274(24):17284–17289.

40. Evnin, L. B., Vásquez, J. R., and Craik, C. S. (1990). Substrate specificity of trypsin investigated by using a genetic selection. *Proceedings of the National Academy of Sciences*, 87(17):6659–6663.

41. Felsenstein, J. (2004). *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA.

42. Fernandez-de Cossio-Diaz, J., Cocco, S., and Monasson, R. (2023). Disentangling representations in restricted boltzmann machines without adversaries. *Physical Review X*, 13(2):021003.

43. Figliuzzi, M., Jacquier, H., Schug, A., Tenaillon, O., and Weigt, M. (2016). Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1. *Mol. Biol. Evol.*, 33(1):268.

44. Finn, R. D., Clements, J., and Eddy, S. R. (2011). Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37.

45. Fischer, A. and Igel, C. (2012). An introduction to restricted boltzmann machines. In *Iberoamerican congress on pattern recognition*, pages 14–36, Berlin, Germany. Springer, Springer.

46. Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*. Clarendon.

47. Fitch, W. M. and Margoliash, E. (1967). Construction of phylogenetic trees: a method based on mutation distances as estimated from cytochrome c sequences is of general applicability. *Science*, 155(3760):279–284.

48. Flajolet, P. and Sedgewick, R. (2009). *Analytic combinatorics*. cambridge University press.

49. Friedman, L. and Cocco, S. (2021). Machine learning and protein specificity. Master's thesis, Ecole Normale Superieure.

50. Gao, C.-Y., Cecconi, F., Vulpiani, A., Zhou, H.-J., and Aurell, E. (2019). DCA for genome-wide epistasis analysis: the statistical genetics perspective. *Phys. Biol.*, 16(2):026002.

51. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

52. Greenbury, S. F., Louis, A. A., and Ahnert, S. E. (2021). The structure of genotype-phenotype maps makes fitness landscapes navigable. *bioRxiv*.

53. Greener, J. G., Moffat, L., and Jones, D. T. (2018). Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, 8(1):16189.

54. Halabi, N., Rivoire, O., Leibler, S., and Ranganathan, R. (2009). Protein sectors: evolutionary units of three-dimensional structure. *Cell*, 138(4):774–786.

55. Hansen, T. F. (2006). The evolution of genetic architecture. *Annu. Rev. Ecol. Evol. Syst.*, 37:123–157.

56. Hansen, T. F. and Wagner, G. P. (2001). Modeling genetic architecture: a multilinear theory of gene interaction. *Theoretical population biology*, 59(1):61–86.

57. Hawkins-Hooker, A., Depardieu, F., Baur, S., Couairon, G., Chen, A., and Bikard, D. (2021). Generating functional protein variants with variational autoencoders. *PLOS Computational Biology*, 17(2):1–23.

58. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

59. Hedstrom, L. (2002). Serine protease mechanism and specificity. *Chemical reviews*, 102(12):4501–4524.

60. Hedstrom, L., Farr-Jones, S., Kettner, C. A., and Rutter, W. J. (1994). Converting trypsin to chymotrypsin: ground-state binding does not determine substrate specificity. *Biochemistry*, 33(29):8764–8769.

61. Hedstrom, L., Szilagyi, L., and Rutter, W. J. (1992). Converting trypsin to chymotrypsin: the role of surface loops. *Science*, 255(5049):1249–1253.

62. Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800.

63. Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Scharfe, C. P. I., Springer, M., Sander, C., and Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.*, 35(2):128–135.

64. Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558.

65. Hordijk, W. and Stadler, P. F. (1998). Amplitude spectra of fitness landscapes. *Advances in Complex Systems*, 1(01):39–66.

66. Jacquin, H., Gilson, A., Shakhnovich, E., Cocco, S., and Monasson, R. (2016). Benchmarking inverse statistical approaches for protein structure and design with exactly solvable models. *PLOS Computational Biology*, 12(5):1–18.

67. Jäger, M., Zhang, Y., Bieschke, J., Nguyen, H., Dendle, M., Bowman, M. E., Noel, J. P., Gruebele, M., and Kelly, J. W. (2006). Structure–function–folding relationship in a ww domain. *Proceedings of the National Academy of Sciences*, 103(28):10648–10653.

68. Jelinek, B., Antal, J., Venekei, I., and Gráf, L. (2004). Ala226 to gly and ser189 to asp mutations convert rat chymotrypsin b to a trypsin-like protease. *Protein Engineering Design and Selection*, 17(2):127–131.

69. Jensen, J. L. and Pedersen, A.-M. K. (2000). Probabilistic models of dna sequence evolution with context dependent rates of substitution. *Advances in Applied Probability*, 32(2):499–517.

70. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589.

71. Kageyama, H., Kondo, T., and Iwasaki, H. (2003). Circadian Formation of Clock Protein Complexes by KaiA, KaiB, KaiC, and SasA in Cyanobacteria *. *Journal of Biological Chemistry*, 278(4):2388–2395.

72. Kato, Y., Ito, M., Kawai, K., Nagata, K., and Tanokura, M. (2002). Determinants of ligand specificity in groups i and iv ww domains as studied by surface plasmon resonance and model building. *Journal of Biological Chemistry*, 277(12):10173–10177.

73. Katoh, K., Misawa, K., Kuma, K.-i., and Miyata, T. (2002). Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic acids research*, 30(14):3059–3066.

74. Kauzmann, W. (1959). Some Factors in the Interpretation of Protein Denaturation11The preparation of this article has been assisted by a grant from the National Science Foundation. In Anfinsen, C. B., Anson, M. L., Bailey, K., and Edsall, J. T., editors, *Advances in Protein Chemistry*, volume 14, pages 1–63. Academic Press.

75. Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R., Wyckoff, H., and Phillips, D. C. (1958). A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature*, 181(4610):662–666.

76. Khersonsky, O. and Tawfik, D. S. (2010). Enzyme promiscuity: A mechanistic and evolutionary perspective. *Annual Review of Biochemistry*, 79(1):471–505.

77. Kimura, M. (1956). A model of a genetic system which leads to closer linkage by natural selection. *Evolution*, 10(3):278–287.

78. Kimura, M. (1965). Attainment of quasi linkage equilibrium when gene frequencies are changing by natural selection. *Genetics*, 52(5):875–890.

79. Kimura, M. (1983). *The neutral theory of molecular evolution*. Cambridge University Press.

80. Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

81. Kolberg, K., Puettmann, C., Pardo, A., Fitting, J., and Barth, S. (2013). Snap-tag technology: a general introduction. *Curr. Pharm. Des*, 19(30):5406–5413.

82. Korf, I., Yandell, M., and Bedell, J. (2003). *Blast*. " O'Reilly Media, Inc.".

83. Kuchner, O. and Arnold, F. H. (1997). Directed evolution of enzyme catalysts. *Trends in Biotechnology*, 15(12):523–530.

84. Kussell, E. and Vucelja, M. (2014). Non-equilibrium physics and evolution—adaptation, extinction, and ecology: a key issues review. *Reports on Progress in Physics*, 77(10):102602.

85. Lau, K. F. and Dill, K. A. (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997.

86. Le Roux, N. and Bengio, Y. (2008). Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649.

87. Leuthäusser, I. (1987). Statistical mechanics of eigen's evolution model. *Journal of statistical physics*, 48(1):343–360.

88. Levitt, M. and Gerstein, M. (1998). A unified statistical framework for sequence comparison and structure comparison. *Proceedings of the National Academy of sciences*, 95(11):5913–5920.

89. Magee, A. F., Hilton, S. K., and DeWitt, W. S. (2021). Robustness of Phylogenetic Inference to Model Misspecification Caused by Pairwise Epistasis. *Molecular Biology and Evolution*, 38(10):4603–4615.

90. Malbranke, C., Rostain, W., Depardieu, F., Cocco, S., Monasson, R., and Bikard, D. (2023). Computational design of novel cas9 pam-interacting domains using evolution-based modelling and structural quality assessment. *bioRxiv*, pages 2023–03.

91. Mariz, B. d. P., Carvalho, S., Batalha, I. L., and Pina, A. S. (2021). Artificial enzymes bringing together computational design and directed evolution. *Org. Biomol. Chem.*, 19:1915–1925.

92. Marks, D. S., Colwell, L. J., Sheridan, R., Hopf, T. A., Pagnani, A., Zecchina, R., and Sander, C. (2011). Protein 3D structure computed from evolutionary sequence variation. *PLoS One*, 6(12):e28766.

93. Mauri, E., Cocco, S., and Monasson, R. (2021). Gaussian closure scheme in the quasi-linkage equilibrium regime of evolving genome populations. *EPL (Europhysics Letters)*, 132(5):56001.

94. Mauri, E., Cocco, S., and Monasson, R. (2023a). Mutational paths with sequence-based models of proteins: from sampling to mean-field characterization. *Physical Review Letters*, 130(15):158402.

95. Mauri, E., Cocco, S., and Monasson, R. (2023b). Transition paths in potts-like energy landscapes: General properties and application to protein sequence models. *Physical Review E*, 108(2):024141.

96. McBride, J. M., Polev, K., Reinharz, V., Grzybowski, B. A., and Tlusty, T. (2022). Alphafold2 can predict structural and phenotypic effects of single mutations. *arXiv preprint arXiv:2204.06860*.

97. McCandlish, D. M., Otwinowski, J., and Plotkin, J. B. (2015). Detecting epistasis from an ensemble of adapting populations. *Evolution*, 69(9):2359–2370.

98. McQuarrie, D. A. (2000). *Statistical mechanics*. Sterling Publishing Company.

99. Meilă, M. and Jaakkola, T. (2006). Tractable bayesian learning of tree belief networks. *Statistics and Computing*, 16:77–92.

100. Mézard, M. and Montanari, A. (2009). *Information, Physics, and Computation*. Oxford University Press.

101. Mézard, M., Parisi, G., and Virasoro, M. A. (1987). *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*. World Scientific.

102. Michaelis, L., Menten, M. L., et al. (1913). Die kinetik der invertinwirkung. *Biochem. z*, 49(333-369):352.

103. Michener, C. D. and Sokal, R. R. (1957). A quantitative approach to a problem in classification. *Evolution*, 11(2):130–162.

104. Miyazawa, S. and Jernigan, R. L. (1996). Residue – Residue Potentials with a Favorable Contact Pair Term and an Unfavorable High Packing Density Term, for Simulation and Threading. *J. Mol. Biol.*, 256(3):623–644.

105. Mora, T., Walczak, A. M., and Zamponi, F. (2012). Transition path sampling algorithm for discrete many-body systems. *Physical Review E*, 85(3):036710.

106. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl. Acad. Sci.*, 108(49):E1293–E1301.

107. Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, 11:125–139.

108. Neher, R. A. and Shraiman, B. I. (2011a). Statistical genetics and evolution of quantitative traits. *Rev. Mod. Phys.*, 83:1283–1300.

109. Neher, R. A. and Shraiman, B. I. (2011b). Statistical genetics and evolution of quantitative traits. *Reviews of Modern Physics*, 83(4):1283.

110. Neher, R. A., Vucelja, M., Mezard, M., and Shraiman, B. I. (2013). Emergence of clones in sexual populations. *Journal of Statistical Mechanics: Theory and Experiment*, 2013(01):P01008.

111. Noble, D. (2002). The rise of computational biology. *Nature Reviews Molecular Cell Biology*, 3(6):459–463.

112. Olsen, J. V., Ong, S.-E., and Mann, M. (2004). Trypsin cleaves exclusively c-terminal to arginine and lysine residues. *Molecular & cellular proteomics*, 3(6):608–614.

113. Opper, M. and Saad, D. (2001). *Advanced mean field methods: Theory and practice*. MIT press.

114. Otte, L., Wiedemann, U., Schlegel, B., Pires, J. R., Beyermann, M., Schmieder, P., Krause, G., Volkmer-Engert, R., Schneider-Mergener, J., and Oschkinat, H. (2003). Ww domain sequence activity relationships identified using ligand recognition propensities of 42 ww domains. *Protein Science*, 12(3):491–500.

115. Otwinowski, J., McCandlish, D. M., and Plotkin, J. B. (2018). Inferring the shape of global epistasis. *Proceedings of the National Academy of Sciences*, 115(32):E7550–E7558.

116. Pauling, L. and Corey, R. B. (1951). Atomic Coordinates and Structure Factors for Two Helical Configurations of Polypeptide Chains. *Proceedings of the National Academy of Sciences*, 37(5):235–240.

117. Pettersen, E. F., Goddard, T. D., Huang, C. C., Meng, E. C., Couch, G. S., Croll, T. I., Morris, J. H., and Ferrin, T. E. (2021). Ucsf chimerax: Structure visualization for researchers, educators, and developers. *Protein Science*, 30(1):70–82.

118. Poelwijk, F. J., Socolich, M., and Ranganathan, R. (2019). Learning the pattern of epistasis linking genotype and phenotype in a protein - Nature Communications. *Nat. Commun.*, 10(4213):1–11.

119. Porter, L. L. and Looger, L. L. (2018). Extant fold-switching proteins are widespread. *Proceedings of the National Academy of Sciences of the United States of America*, 115(23):5968–5973.

120. Posani, L., Rizzato, F., Monasson, R., and Cocco, S. (2022). Infer global, predict local: quantity-quality trade-off in protein fitness predictions from sequence data. *bioRxiv*, pages 2022–12.

121. Ramachandran, G. N., Ramakrishnan, C., and Sasisekharan, V. (1963). Stereochemistry of polypeptide chain configurations. *Journal of Molecular Biology*, 7:95–99.

122. Ravikumar, P., Wainwright, M. J., and Lafferty, J. D. (2010). High-dimensional ising model selection using ℓ_1-regularized logistic regression. *Ann. Stat.*, 38(3):1287–1319.

123. Riesselman, A. J., Ingraham, J. B., and Marks, D. S. (2018). Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822.

124. Rodríguez-Horta, E., Lage-Castellanos, A., and Mulet, R. (2022). Ancestral sequence reconstruction for co-evolutionary models. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(1):013502.

125. Rodriguez-Rivas, J., Croce, G., Muscat, M., and Weigt, M. (2022). Epistatic models predict mutable sites in sars-cov-2 proteins and epitopes. *Proceedings of the National Academy of Sciences*, 119(4):e2113118119.

126. Russ, W. P., Figliuzzi, M., Stocker, C., Barrat-Charlaix, P., Socolich, M., Kast, P., Hilvert, D., Monasson, R., Cocco, S., Weigt, M., and Ranganathan, R. (2020). An evolution-based model for designing chorismate mutase enzymes. *Science*, 369:440–5.

127. Russ, W. P., Lowery, D. M., Mishra, P., Yaffe, M. B., and Ranganathan, R. (2005). Natural-like function in artificial ww domains. *Nature*, 437(7058):579–583.

128. Salakhutdinov, R. and Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879.

129. Šali, A., Shakhnovich, E., and Karplus, M. (1994). How does a protein fold. *nature*, 369(6477):248–251.

130. Schnoerr, D., Sanguinetti, G., and Grima, R. (2015). Comparison of different moment-closure approximations for stochastic chemical kinetics. *The Journal of Chemical Physics*, 143(18):11B610_1.

131. Schöniger, M. and Von Haeseler, A. (1994). A stochastic model for the evolution of autocorrelated dna sequences. *Molecular phylogenetics and evolution*, 3(3):240–247.

132. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W. R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., and Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710.

133. Seo, S., Oh, M., Park, Y., and Kim, S. (2018). Deepfam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, 34(13):i254–i262.

134. Shakhnovich, E. and Gutin, A. (1990). Enumeration of all compact conformations of copolymers with random sequence of links. *The Journal of Chemical Physics*, 93(8):5967–5971.

135. Shakhnovich, E. I. (1997). Theoretical studies of protein-folding thermodynamics and kinetics. *Current opinion in structural biology*, 7(1):29–40.

136. Shakhnovich, E. I. and Gutin, A. M. (1993). Engineering of stable and fast-folding sequences of model proteins. *Proceedings of the National Academy of Sciences*, 90(15):7195–7199.

137. Shekhar, K., Ruberman, C. F., Ferguson, A. L., Barton, J. P., Kardar, M., and Chakraborty, A. K. (2013). Spin models inferred from patient-derived viral sequence data faithfully describe hiv fitness landscapes. *Physical review E*, 88(6):062705.

138. Sherrington, D. and Kirkpatrick, S. (1975). Solvable model of a spin-glass. *Phys. Rev. Lett.*, 35:1792–1796.

139. Shi, D., Nannenga, B. L., Iadanza, M. G., and Gonen, T. (2013). Three-dimensional electron crystallography of protein microcrystals. *elife*, 2:e01345.

140. Shibata, M., Ham, K., and Hoque, M. O. (2018). A time for yap1: Tumorigenesis, immunosuppression and targeted therapy. *International journal of cancer*, 143(9):2133–2144.

141. Sinclair, J. F., Ziegler, M. M., and Baldwin, T. O. (1994). Kinetic partitioning during protein folding yields multiple native states. *Nature Structural Biology*, 1(5):320–326.

142. Socolich, M., Lockless, S., Russ, W., Lee, H., Gardner, K., and Ranganathan, R. (2005). Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–8.

143. Stadler, P. F. and Wagner, G. P. (1997). Algebraic theory of recombination spaces. *Evolutionary computation*, 5(3):241–275.

144. Storkey, A. (1997). Increasing the capacity of a hopfield network without sacrificing functionality. In *Artificial Neural Networks—ICANN'97: 7th International Conference Lausanne, Switzerland, October 8–10, 1997 Proceeedings 7*, pages 451–456. Springer.

145. Sudol, M. (1996). Structure and function of the ww domain. *Progress in biophysics and molecular biology*, 65(1-2):113–132.

146. Sudol, M. and Hunter, T. (2000). New wrinkles for an old domain. *Cell*, 103(7):1001–1004.

147. Thompson, J. D., Gibson, T. J., and Higgins, D. G. (2003). Multiple sequence alignment using clustalw and clustalx. *Current Protocols in Bioinformatics*, 00(1):2.3.1–2.3.22.

148. Tian, P. and Best, R. B. (2020). Exploring the sequence fitness landscape of a bridge between protein folds. *PLOS Computational Biology*, 16(10):1–19.

149. Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. Association for Computing Machinery, New York, NY, USA.

150. Tubiana, J. (2018a). Probabilistic graphical models (pgm). Available online at: `https://github.com/jertubiana/PGM`.

151. Tubiana, J. (2018b). *Restricted Boltzmann machines : from compositional representations to protein sequence analysis*. PhD thesis, Université Paris sciences et lettres, Paris, France.

152. Tubiana, J., Cocco, S., and Monasson, R. (2019). Learning protein constitutive motifs from sequence data. *eLife*, 8:e39397.

153. Tubiana, J. and Monasson, R. (2017). Emergence of compositional representations in restricted boltzmann machines. *Physical review letters*, 118(13):138301.

154. Vanden-Eijnden, E. et al. (2010). Transition-path theory and path-finding algorithms for the study of rare events. *Annual review of physical chemistry*, 61:391–420.

155. Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., Žídek, A., Green, T., Tunyasuvunakool, K., Petersen, S., Jumper, J., Clancy, E., Green, R., Vora, A., Lutfi, M., Figurnov, M., Cowie, A., Hobbs, N., Kohli, P., Kleywegt, G., Birney, E., Hassabis, D., and Velankar, S. (2021). AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444.

156. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

157. Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I. M., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. (2022). Language models generalize beyond natural proteins. *bioRxiv*.

158. Wang, S., Sun, S., Li, Z., Zhang, R., and Xu, J. (2017). Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS computational biology*, 13(1):e1005324.

159. Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of direct residue contacts in protein-protein interaction by message passing. *Proc. Natl. Acad. Sci.*, 106(1):67–72.

160. Weinberger, E. D. (1991). Fourier and taylor series on fitness landscapes. *Biological cybernetics*, 65(5):321–330.

161. Weißenow, K., Heinzinger, M., and Rost, B. (2022). Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction. *Structure*, 30(8):1169–1177.

162. Wick, G.-C. (1950). The evaluation of the collision matrix. *Physical review*, 80(2):268.

163. Wierenga, R. K. (2001). The TIM-barrel fold: A versatile framework for efficient enzymes. *FEBS Letters*, 492(3):193–198.

164. Wolf, J. B., Brodie, E. D., Wade, M. J., Wade, M. J., et al. (2000). *Epistasis and the evolutionary process*. Oxford University Press, USA.

165. Woolfson, M. M. and Woolfson, M. M. (1997). *An introduction to X-ray crystallography*. Cambridge University Press.

166. Wright, S. (1931). Evolution in mendelian populations. *Genetics*, 16(2):97.

167. Wu, F.-Y. (1982). The potts model. *Reviews of modern physics*, 54(1):235.

168. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O., and Sun, R. (2016). Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965.

169. Wüthrich, K. (1989). Protein structure determination in solution by nuclear magnetic resonance spectroscopy. *Science*, 243(4887):45–50.

170. Zanini, F. and Neher, R. A. (2012). FFPopSim: an efficient forward simulation package for the evolution of large populations. *Bioinformatics*, 28(24):3332–3333.

171. Zarrinpar, A. and Lim, W. A. (2000). Converging on proline: the mechanism of ww domain peptide recognition. *Nature structural biology*, 7(8):611–613.

172. Zeng, H.-L. and Aurell, E. (2020). Inferring genetic fitness from genomic data. *Phys. Rev. E*, 101:052409.

173. Zeng, H.-L., Mauri, E., Dichio, V., Cocco, S., Monasson, R., and Aurell, E. (2021). Inferring epistasis from genomic data with comparable mutation and outcrossing rate. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(8):083501.

174. Zhang, Y. and Skolnick, J. (2004). Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710.

RÉSUMÉ

Tout au long de cette thèse de doctorat, nous explorons les potentialités des modèles basés sur les séquences pour les polymères biologiques, en particulier les protéines. Notre principal objectif est de comprendre leur capacité à inférer les contraintes évolutives essentielles à partir des données de séquence générées par ces modèles.

Initialement, nous étudions la façon dont les données de séquence sont façonnées par l'évolution dans le régime de Quasi Équilibre de Liaison (QLE), où la distribution des génotypes ressemble à une distribution de Boltzmann avec des sites faiblement interactifs. Nous introduisons et illustrons un schéma de fermeture gaussienne pour un modèle épistatique à courte portée de l'évolution. De plus, nous explorons les conditions dans lesquelles des informations évolutives critiques, telles que le paysage de fitness, peuvent être correctement inférées à partir des données résultantes, en utilisant des outils tels que l'analyse des couplages directs (DCA) ou notre schéma de fermeture gaussienne.

En second lieu, nous abordons la caractérisation des chemins mutationnels entre les protéines homologues distantes en utilisant les machines de Boltzmann restreintes (RBM). Nous développons un algorithme de Monte-Carlo efficace pour échantillonner les chemins mutationnels, maximisant la probabilité des séquences intermédiaires tout en tenant compte de modèles complexes basés sur les séquences. En utilisant des RBM entraînées sur des protéines sur réseau et des familles de domaines WW, nous identifions avec succès des chemins mutationnels significatifs, y compris la découverte de régions ancestrales de protéines ayant disparu au cours de l'évolution. De plus, nous introduisons une théorie du champ moyen qui caractérise les chemins selon différentes dynamiques mutationnelles, fournissant des informations précieuses sur l'évolution de diverses familles de protéines. Des résultats expérimentaux préliminaires prometteurs pour la validation des chemins mutationnels dans les domaines WW et les sérine protéases sont également présentés.

MOTS CLÉS

Évolution des protéines, Modèles basés sur les séquences, Quasi Équilibre de Liaison, Chemins mutationnels

ABSTRACT

Throughout this Ph.D. thesis, we investigate the capabilities of sequence-based models for biological polymers, particularly proteins. Our main focus is to understand their ability to infer essential evolutionary constraints from the sequence data generated by them.

Firstly, we explore the shaping of sequence data by evolution in the regime of Quasi-linkage Equilibrium (QLE), where the genotype distribution resembles a Boltzmann distribution with weakly interacting sites. We introduce and illustrate a Gaussian closure scheme for a short-range epistatic model of evolution. Furthermore, we explore the conditions under which critical evolutionary information, such as the fitness landscape, can be accurately inferred from the resulting data, using tools such as Direct Coupling Analysis (DCA) or our Gaussian closure scheme.

Secondly, we address the characterization of mutational paths between distant homologous proteins using Restricted Boltzmann Machines (RBMs). We develop an efficient Monte-Carlo algorithm to sample mutational paths, maximizing the probability of intermediate sequences while considering complex sequence-based models. By utilizing RBMs trained on Lattice Proteins and WW domain families, we successfully identify meaningful mutational paths, including the possible discovery of ancestral promiscuous regions of proteins that disappeared during evolution. Additionally, we introduce a mean-field theory that characterizes paths under different mutational dynamics, providing valuable insights into the evolution of diverse protein families. Promising preliminary experimental results for the validation of mutational paths in WW domains and serine proteases are also presented.

KEYWORDS

Protein Evolution, Sequence-based Models, Quasi-linkage Equilibrium, Mutational Paths