Learning and sampling complex landscapes with restricted Boltzmann machines : from theory to the fitness of the TEM-1 protein

Clément Roussel

▶ **To cite this version:**

Clément Roussel. Learning and sampling complex landscapes with restricted Boltzmann machines : from theory to the fitness of the TEM-1 protein. Physics [physics]. Université Paris sciences et lettres, 2021. English. NNT : 2021UPSLE019 . tel-03774086

HAL Id: tel-03774086
https://theses.hal.science/tel-03774086

Submitted on 9 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'Ecole Normale Supérieure

Learning and Sampling Complex Landscapes with Restricted Boltzmann Machines: from Theory to the Fitness of the TEM-1 Protein

Soutenue par
Clément Roussel
Le 2 décembre 2021

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

Spécialité
Physique

Composition du jury :

François COSTE INRIA	*Président du jury*
Adriano BARRA Università del Salento	*Rapporteur*
Eugene SHAKHNOVICH Harvard University	*Rapporteur*
Anne-Florence BITBOL École Polytechnique Fédérale de Lausanne	*Examinatrice*
Aurélien DECELLE Universidad Complutense de Madrid	*Examinateur*
Olivier TENAILLON Université de Paris	*Invité*
Simona COCCO École Normale Supérieure	*Directrice de thèse*
Rémi MONASSON École Normale Supérieure	*Directeur de thèse*

ENS | PSL

LPENS
LABORATOIRE DE PHYSIQUE DE L'ÉCOLE NORMALE SUPÉRIEURE

DGA

Remerciements

Je tiens tout d'abord à remercier mes deux directeurs de thèse, Simona Cocco et Rémi Monasson pour m'avoir encadré durant les trois dernières années. Merci de m'avoir fait profiter d'un encadrement de haut niveau sur des sujets intéressants pour mon court passage dans le monde de la recherche, et d'avoir été compréhensifs sur mes absences liées à ma formation au Corps de l'Armement.

Je remercie aussi le Corps de l'Armement, ainsi que la Direction Générale de l'Armement de m'avoir permis d'effectuer mon premier poste d'Ingénieur de l'Armement en tant que doctorant à l'ENS. Cette expérience a été très enrichissante et me sera d'une grande utilité pour mes postes à venir.

Je tiens également à remercier Olivier Tenaillon avec lequel j'ai eu la chance de collaborer sur notre projet sur la β-lactamase TEM-1. Merci d'avoir pris le temps de répondre à mes questions élémentaires en biologie, et d'avoir épluché avec moi des centaines de figures de modèles de stabilité. Je tiens aussi à remercier Hervé Jacquier pour ses précieux conseils sur les β-lactamases, entre deux gardes à l'hôpital.

Je remercie aussi Adriano Barra et Eugene Shakhnovich d'avoir accepté d'être les rapporteurs de ce manuscrit ainsi qu'Anne-Florence Bitbol, François Coste et Aurélien Decelle pour leur présence en tant qu'examinateurs lors de la soutenance.

Je voudrais aussi remercier Jean-François Allemand et Francis Corson, respectivement mon parrain et mon tuteur scientifique de l'école doctorale, pour avoir participé à mes comités de suivi et pour leurs conseils scientifiques. Je remercie aussi Laurent Taysse, mon tuteur de la Direction Générale de l'Armement, et j'espère ne pas l'avoir effrayé avec mes modèles de physique statistique qui peuvent sembler éloignés de prime abord de la biologie.

Je tiens aussi particulièrement à remercier les membres passés ou actuels de notre groupe. Arnaud, pour nos nombreuses discussions et notre victoire au data challenge de l'ENS sur le Rubik's Cube $2 \times 2 \times 2$. Jérôme, pour ses nombreux conseils sur les RBM, toujours à l'écoute et réactif, même depuis Israël. Marco, pour sa bonne humeur légendaire et son fameux tiramisu. Aldo, pour sa touche italienne et ses conseils. Sébastien, pour son "petit concert entre potes" au Zénith de Paris. Andrea, pour nos belles parties de pétanque. Eugenio, qui semble bien parti pour réaliser ce qui était initialement mon sujet de thèse. Et aussi tous les autres, Lorenzo, Francesca, Barbara, Cyril, Jorge, Tobias, Hugo, Leah, Mariia, Massimiliano ...

À mes copains du laboratoire, Arnaud, Hugo, Victor, Stéphane, Tristan, pour avoir partagé de nombreux cafés, repas et réflexions plus ou moins enjouées sur la recherche, dans à-peu-près tous les restaurants des alentours.

À mes camarades de bureau, Victor, Kévin, Lorenzo, Marco, Simone, Éli, Ling, Ludwig et Gabriel, pour m'avoir vu progresser de la porte du bureau vers la grande fenêtre avec la vue sur le jardin du département. Un grand merci à Kévin, pour son beau modèle de thèse que j'utilise.

À mes camarades d'hackathon quantique, Tristan, Bruno, Brice et Lionel. Après deux deuxièmes places, j'espère que la troisième fois sera la bonne.

J'aimerais remercier mes camarades avec qui j'ai eu la chance d'aller au Japon pour une formidable école d'automne : Maria, Meriem, Kevin, Tristan, Hugo et Victor. Promis, on trouvera le temple d'or à Kyoto la prochaine fois.

À mes camarades du Corps de l'Armement, avec qui j'ai eu la chance de traverser l'Atlantique sur le PHA Tonnerre, ce qui a été une bonne aération après ma première année de thèse : Maylis, Florence, Camille, Jean-Baptiste, Hugo, Antoine, Jean-Roch et Michel.

À mes amis de prépa du lycée Louis-le-Grand, majoritairement thésards et désormais docteurs (désolé Théo), pour nos excursions et voyages : François-Pierre, Thibaut, Théo et Armand.

À mes amis du 45 et compagnons de Pot. À Ulysse et Émile pour avoir tenté de m'initier à la philosophie lors de leur préparation à l'agrégation. Et à Thibault, pour m'avoir fait profiter de la vue du DMA.

À Alexandra et Armand, pour leur amitié et nos footings au parc de Sceaux.

J'aimerais aussi remercier mes camarades du cours de Physique Pour Tous : Tristan, Bruno, Thibaud et Brice. J'espère que nos remplaçants arriveront à comprendre le lien entre Nietzsche et la relativité.

J'aimerais remercier Mathieu pour ses explications sur la détermination des structures expérimentales des protéines, ainsi que pour ses gâteaux toujours très légers en beurre.

Merci aussi à mes illustres et antiques camarades, Alexandre, José et Clément.

À mes amis physiciens, Baptiste et Vincent, pour nos discussions physico-politiques autour d'une bière du Pantalon ou du libanais place Monge.

À mes amis libanais de la place Monge, pour avoir égayé mes repas deux fois par semaine avec le fameux foie de volailles, tapenade aux figues et crème d'ail, taboulé boulghour moussaka sans harissa.

À mes différents colocataires, eux-aussi thésards, avec qui j'ai partagé une partie de cette aventure avant de m'installer avec Flora : Vincent, François-Pierre et Armand.

Merci à mon beau-père, Michel, pour la relecture de ce manuscrit et ses nombreux conseils avisés. Merci à ma belle-mère, Geneviève, pour ses nombreux conseils cinématographiques qui ont égayé nos sorties au cinéma.

À mes oncles et tantes, et à mes grand-parents, pour leur soutien inconditionnel.

Merci à mes parents, et à mon frère Aurélien, pour m'avoir supporté depuis toutes ces années. On forme une très bonne équipe, et ça ne risque pas de s'arrêter de sitôt !

Et pour conclure, un grand merci à mon incroyable Flora, ma copine (et désormais compagne !), pour m'avoir encouragé et soutenu durant ces années. Sans toi, la vie aurait été réellement différente.

Et maintenant, attaquons-nous au vif du sujet !

Introduction

This thesis will be devoted to the study of the sampling of Restricted Boltzmann Machines (RBM) and the study of epistasis on an alpha-helix of the TEM-1 β-lactamase. These two themes may at first seem rather distant, but these two themes will converge together when we use RBM to study β-lactamases in Chapter 9.

This thesis is divided into four distinct parts.

Part I will be devoted to a general introduction to the Restricted Boltzmann Machines from a machine learning and physics point of view. The purpose of this part is to introduce the concepts that will be useful for Part II.

- Chapter 1 introduces the RBM, their different influencing parameters, and the different learning algorithms used to train them. This chapter reviews the different applications, past or present, of RBM in the machine learning community.
- Chapter 2 introduces the Hopfield network and different results on storing memories with it. It links with RBM and shows different ways to store and retrieve memories with a RBM. It also reviews the various known results on RBM from the statistical physics community.

Part II will be devoted to the study of the sampling of an energy landscape with RBM and gathers the main theoretical and numerical results that we found during this thesis. This section highlights that Alternating Gibbs Sampling is just as inefficient as Metropolis-Hastings to sample an energy landscape. However, we show that the sampling can be improved by taking advantage of the representations learned by the RBM in its hidden space.

- Chapter 3 is devoted to the study of the canonical sampling algorithm, Alternating Gibbs Sampling. We show that this algorithm is just as inefficient as a classical Metropolis-Hastings. Nevertheless, the space of hidden units allows extracting a useful representation of the data, and the use of Metropolis-Hastings in this space improves the sampling. This chapter is based on our following publication, accepted in *Physical Review E*:

  [1] Roussel, C., Cocco, S., and Monasson, R. (2021). Barriers and Dynamical Paths in alternating Gibbs sampling of restricted Boltzmann machines, *Physical Review E*

- Chapter 4 is devoted to the study of RBM sampling using the Deep Tempering algorithm. This algorithm is based on a stack of RBM that can exchange their configurations to improve the sampling. We show that this overcomes the limitations observed by our sampling in the hidden space in Chapter 3. This chapter is based on

our paper, *in preparation*:

[2] Roussel, C., Cocco, S., and Monasson, R. (2021). Improving Sampling of Restricted Boltzmann Machines with Deep Tempering, *In preparation*

Part III will be devoted to a general introduction on proteins, and the use of Potts models and neural networks to predict their structures and the effects of mutations. This part is intended to be introductory, and to bring useful elements of understanding for the Part IV devoted to class A β-lactamases.

- Chapter 5 is an introduction to proteins for physicists. It highlights the importance of the 3D structure of proteins on their functionalities and the experimental and theoretical difficulties of obtaining the structure. In addition, it introduces the concept of protein co-evolution, which gives hope to the determination of the protein structure from its sequence.
- Chapter 6 introduces in detail the Direct Coupling Analysis, based on Potts models, which allows predicting the protein structure from the sequences of a protein family, and to score them. It details the different technical aspects of this method, and also presents some recent advances using neural networks.

Part IV will be devoted to a particular protein family, the β-lactamases.

- Chapter 7 is devoted to the study of epistasis on the α-helix of the TEM-1 β-lactamase. We show that a two-state model well captures this epistasis and that coupled to a Potts model trained on a multiple sequence alignment, it allows determining the sign of the epistasis. This chapter is based on our paper, *in preparation*:

  [3] Birgy, A.†, Roussel, C.†, Kemble, H., Mullaert, J., Panigoni, K., Chapron, A., Chatel, J., Magnan, M., Jacquier, H., Cocco, S., Monasson, R., Tenaillon, O., Origins and breadth of pairwise epistasis in an α-helix of β-lactamase TEM-1, *In preparation* (†: joint first authors)

- Chapter 8 presents preliminary results on the influence of amoxicillin concentration in TEM-1 log-fitness and epistasis.
- Chapter 9 presents how RBM can identify through their weights relevant information concerning the phylogeny, the functionality, and the structure of class A β-lactamases.

Contents

List of Figures

List of Algorithms

Résumé substantiel en français

Dans cette thèse, mon attention s'est portée à la fois sur les machines de Boltzmann restreintes (RBM) ainsi que sur une protéine nommée TEM-1 issue de la famille des β-lactamases. Ces deux sujets peuvent sembler de prime abord relativement disjoints, mais comme nous le verrons par la suite, les machines de Boltzmann restreintes, et plus généralement la physique statistique, se sont trouvées fort utiles pour étudier ladite protéine.

Résultats obtenus sur les machines de Boltzmann restreintes

Présentation succincte des machines de Boltzmann restreintes

Les machines de Boltzmann restreintes sont des réseaux de neurones artificiels inventés en 1986 par Paul Smolensky (Smolensky, 1986). Ces machines, comme leur nom l'indique, sont un cas particulier des machines de Boltzmann inventées trois ans plus tôt par Geoffrey Hinton et Terrence Sejnowski (Hinton and Sejnowski, 1983).

Les machines de Boltzmann restreintes sont des réseaux de neurones artificiels à deux couches, l'une nommée visible $\mathbf{v}$, qui représente les données, l'autre appelée cachée $\mathbf{h}$, qui est l'espace des représentations des données (Fig. 1(a)). Ces deux couches forment un graphe non orienté, et l'ensemble des variables aléatoires $\mathbf{v} = \{v_i\}_{i=1\dots N}$ et $\mathbf{h} = \{h_\mu\}_{\mu=1\dots M}$ vérifie une propriété de Markov relativement à ce graphe. Les deux couches sont connectées par une matrice de poids $\mathbf{W}$, et on peut définir une probabilité jointe de Boltzmann sur les configurations $\mathbf{v}$ et $\mathbf{h}$

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\left(-E(\mathbf{v},\mathbf{h})\right), \tag{1}$$

où l'énergie $E(\mathbf{v},\mathbf{h})$ s'écrit sous la forme

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{N}\sum_{\mu=1}^{M} W_{i\mu}v_i h_\mu + \sum_{\mu=1}^{M}\mathcal{U}_\mu(h_\mu) + \sum_{i=1}^{N}\mathcal{V}_i(v_i). \tag{2}$$

Le caractère "restreint" des RBM provient du fait que le graphe est biparti, c'est-à-dire qu'il n'y a des connexions qu'entre les unités visibles et cachées, et non pas directement entre les unités visibles d'une part, et les unités cachées d'autre part.

Les machines de Boltzmann restreintes sont entraînées en maximisant la log-vraisemblance sur les données d'entraînement. Après l'apprentissage, les paramètres de la machine sont fixés, ce qui détermine la distribution de probabilité (Eq. (1)) ainsi que la représentation des données.

Une fois entraînées, ces machines peuvent générer de nouvelles données grâce à l'échantillonnage alterné de Gibbs (Fig. 1(b)). Celui-ci se décompose en deux étapes :

- En partant d'une configuration visible $\mathbf{v}^t$ au temps t, une configuration cachée $\mathbf{h}^{t+1}$ est échantillonnée selon $P(\mathbf{h}|\mathbf{v}^t)$. Cette étape peut-être vue comme une extraction stochastique des représentations de la configuration $\mathbf{v}^t$.

(a) Machines de Boltzmann restreintes (b) Échantillonnage alterné de Gibbs

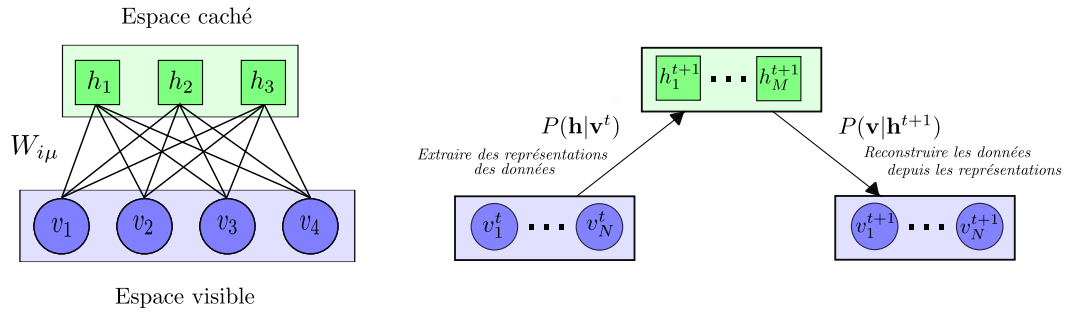


Figure 1: Description des machines de Boltzmann restreintes ainsi que l'échantillonnage alterné de Gibbs. (a) Architecture bipartite des machines de Boltzmann restreintes avec la couche visible en bleu et la couche cachée en vert. (b) Échantillonnage alterné de Gibbs. Les couches visibles et cachées sont échantillonnées alternativement.

- Une nouvelle configuration visible $\mathbf{v}^{t+1}$ est échantillonnée selon $P\left(\mathbf{v}|\mathbf{h}^{t+1}\right)$. Cette étape peut-être vue comme une reconstruction stochastique de $\mathbf{v}$ à partir de la représentation $\mathbf{h}^{t+1}$.

Résultats obtenus

Dans cette thèse, nous nous sommes intéressés aux propriétés de l'échantillonnage alterné de Gibbs. Ces résultats sont détaillés dans le papier suivant (Roussel et al., 2021) publié dans *Physical Review E*.

Dans un premier temps, nous avons étudié les trajectoires dans l'espace visible obtenues à l'aide de l'échantillonnage alterné de Gibbs entre deux minima locaux d'un paysage énergétique $E^{\text{eff}}(\mathbf{v})$ défini via

$$P(\mathbf{v}) = \int d\mathbf{h}\, P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\Big(-E^{\text{eff}}(\mathbf{v})\Big), \tag{3}$$

Nous avons montré que l'échantillonnage alterné de Gibbs permet de trouver des trajectoires optimales entre les minima locaux de $E^{\text{eff}}(\mathbf{v})$. Néanmoins, de larges barrières d'énergie libre, extensives dans la taille du système, sont présentes le long de la trajectoire. Par conséquent, l'échantillonnage alterné de Gibbs est tout aussi inefficace que l'échantillonnage de $E^{\text{eff}}(\mathbf{v})$ à l'aide d'un algorithme de Metropolis-Hastings classique. De plus, pour un paysage $E^{\text{eff}}(\mathbf{v})$ donné, ce résultat ne dépend pas de la représentation des configurations visibles dans l'espace des unités cachées, c'est-à-dire que les barrières sont les mêmes pour tout paysage $E(\mathbf{v},\mathbf{h})$ qui a comme distribution marginale $P(\mathbf{v}) = \frac{1}{Z}\exp\Big(-E^{\text{eff}}(\mathbf{v})\Big)$.

Ce résultat est valide pour des modèles en champs moyens, et ne semble pas dépendre de la structure des minima. En effet, que ce soit pour des modèles où les minima sont décorrélés ou bien reliés par symétrie globale, comme par exemple pour le modèle de Curie-Weiss ou de Hopfield (1982), ou bien pour des modèles avec structure, où les minima ont une organisation non triviale et qui représente mieux ce qui est observé sur des données, de larges barrières d'énergie libre sont observées le long de la trajectoire. Pour les modèles avec structure, l'organisation non triviale des minima gouvernent les trajectoires optimales dans $E^{\text{eff}}(\mathbf{v})$. Ces trajectoires peuvent être interprétées plus ou moins facilement dans l'espace caché. Contrairement aux barrières qui ne dépendent que de $E^{\text{eff}}(\mathbf{v})$, cette interprétation dépend des représentations apprises par la RBM et donc de $E(\mathbf{v},\mathbf{h})$.

Nous avons aussi montré que dans le cas où les représentations dans l'espace des unités cachées encodent des modes collectifs d'unités visibles relativement indépendantes, et si le nombre D d'unités cachées à changer d'états est faible pour passer d'un minima de $E^{\text{eff}}(\mathbf{v})$ à un autre, utiliser l'algorithme de Metropolis-Hastings dans l'espace des unités cachées permet d'accélérer l'échantillonnage. Cette dimension D dépend à la fois des données et de la représentation apprise par la RBM. Utiliser des pénalisations de la log-vraisemblance lors de l'entraînement de la RBM pour imposer une représentation où D est faible est possible dans certains cas. Néanmoins, dans certains cas, D est du même ordre que le nombre M d'unités cachées et notre algorithme est inefficace.

Dans un second papier, en cours d'écriture, nous avons étudié une stratégie d'échantillonnage pour palier le cas où un nombre macroscopique D d'unités cachées doivent changer d'états au même moment pour changer de mode. Notre stratégie repose sur une pile de machines de Boltzmann restreintes ainsi que sur l'algorithme "Deep Tempering" développé par Desjardins et al. (2014) pour entraîner des réseaux profonds (Fig. 2).

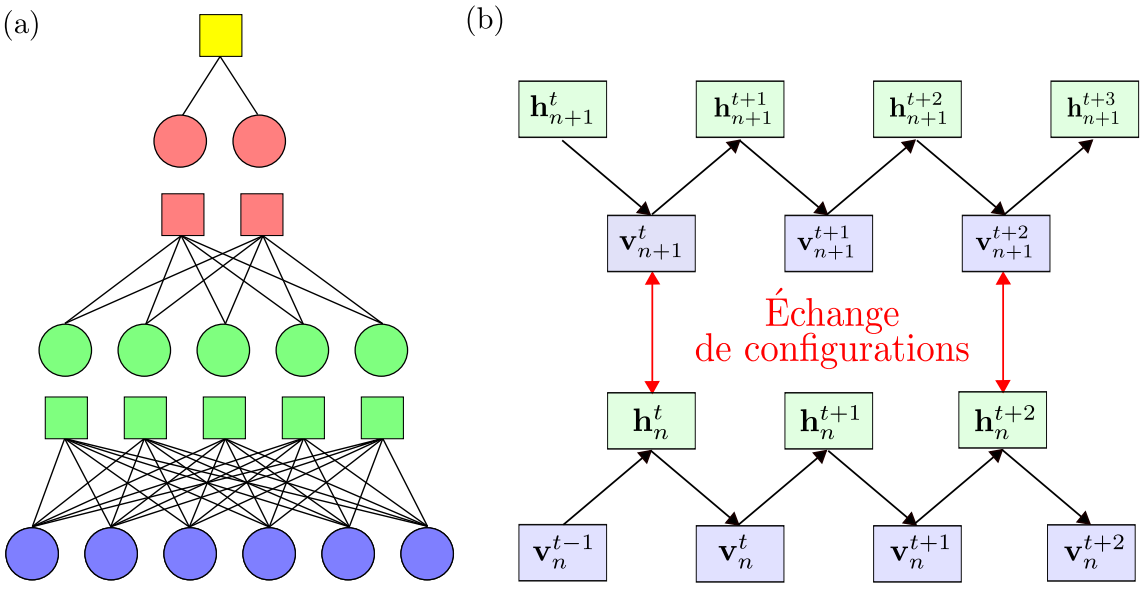


Figure 2: (a) Exemple avec trois machines de Boltzmann restreintes. Le nombre d'unités visibles de $(n+1)^{\text{ème}}$ RBM est égal au nombre d'unités cachées de la $n^{\text{ème}}$ RBM. (b) Illustration du Deep Tempering (ici entre la $n^{\text{ème}}$ et la $(n+1)^{\text{ème}}$ RBM). L'échantillonnage alterné de Gibbs est utilisé pour générer les configurations. Les dynamiques sont couplées et des échanges de configurations sont possibles entre les configurations visibles $\mathbf{v}_{n+1}$ de la $(n+1)^{\text{ème}}$ RBM et les configurations cachées $\mathbf{h}_n$ de la $n^{\text{ème}}$ RBM. Ces configurations sont échangées avec une probabilité $A_n(\mathbf{h}^t_n, \mathbf{v}^t_{n+1})$ (Eq. (1.18)). Cet algorithme a été initialement développé par Desjardins et al. (2014) pour entraîner des réseaux profonds.

Comme nous avons vu que les machines de Boltzmann restreintes étaient capables de détecter des modes collectifs dans les données, nous allons utiliser des machines de Boltzmann restreintes pour détecter des modes collectifs d'unités cachées d'une RBM donnée pour améliorer son échantillonnage (la RBM en bas de la Figure 2(a)).

Nous avons étudié théoriquement l'échantillonnage de Gibbs alterné pour une RBM entraînée sur des données représentées par des clusters orthogonaux, et nous l'avons comparé à l'échantillonnage à l'aide d'une seconde RBM couplée avec l'algorithme du Deep Tempering (Desjardins et al., 2014). Nous avons montré que diminuer le nombre

d'unités cachées de la seconde RBM et limiter la norme de sa matrice de poids permet d'accélérer l'échantillonnage entre les différents modes. De plus, sur des vraies données, comme MNIST 0/1 (LeCun, 1998) ou les Lattice Protein (Shakhnovich and Gutin, 1990; Mirny and Shakhnovich, 2001), où un nombre macroscopique D d'unités cachées doivent changer d'états au même moment pour changer de modes, nous avons montré que cet algorithme permet en effet d'échantillonner entre des modes distants, là où l'échantillonnage de Gibbs alterné ou notre version modifiée avec le Metropolis-Hastings dans l'espace caché échoue à échantillonner correctement entre les modes distants.

Résultats obtenus sur les β-lactamases

Notre travail sur les β-lactamases, et plus particulièrement sur TEM-1 est le fruit d'une collaboration avec le groupe d'Olivier Tenaillon de l'hôpital Bichat. Les β-lactamases sont des protéines responsables de la résistance aux antibiotiques à base de β-lactamines, couramment utilisés pour le traitement des infections à pneumocoques.

Quel est le lien entre cette protéine et la physique statistique ?

Les protéines sont caractérisées par longue séquence de briques élémentaires, appelées acides aminés. Ces protéines ont une structure 3D qui détermine leurs propriétés. Déterminer la structure de la protéine, et donc son phénotype, est un enjeu crucial en biologie ou en pharmacologie. Expérimentalement, il n'est pas simple d'obtenir la structure d'une protéine, car cela repose sur des procédés chronophages et coûteux, comme par exemple la cristallographie. Néanmoins, depuis une vingtaine d'années, les progrès dans le domaine du séquençage ont permis de constituer de larges bases de données de séquences de protéines. De nombreuses techniques ont été développées pour essayer de prédire la structure à partir de la séquence.

La majeure partie de ces techniques reposent sur un alignement de séquences (MSA). Un alignement de séquences regroupe plusieurs protéines issues de différents types d'organisme, mais ayant la même fonctionnalité, et donc ayant probablement la même structure. Ces différentes protéines ont des séquences distinctes du fait de l'évolution. Néanmoins, en analysant ces alignements de séquences, certaines régularités apparaissent, notamment, certains sites ont l'air de coévoluer, c'est-à-dire qu'il y a des corrélations à deux points fortes dans l'alignement. Une hypothèse majeure est de considérer que deux sites qui coévoluent ont de fortes chances d'être proches dans la structure 3D de la protéine. Néanmoins, analyser directement les corrélations à deux points pour déterminer les sites en contact n'est pas satisfaisant : tout comme en physique statistique, des couplages entre deux sites voisins peuvent créer des corrélations à longue distance. L'idée de Weigt et al. (2009), appelée Direct Coupling Analysis (DCA), est de trouver les couplages responsables de ces corrélations, et de les utiliser pour prédire si deux sites sont en contact ou non. Cette technique repose sur l'inférence d'un modèle de Potts sur l'alignement de séquences et a montré ses preuves dans la prédiction de sites en contact sur de nombreuses familles de protéines. De plus, le modèle de Potts apprend une distribution de Boltzmann : plus l'énergie du modèle est basse, plus une séquence a de chance d'appartenir à la famille de protéines sur laquelle a été entraînée le modèle de Potts. Cela permet donc d'évaluer théoriquement l'effet des mutations d'une protéine sur sa fonctionnalité. Et c'est ce que nous avons fait sur les expériences réalisées par le groupe d'Olivier Tenaillon. Il est aussi possible d'entraîner des machines de Boltzmann restreintes sur ces alignements de séquences, pour à la fois donner un score à des séquences via l'énergie de la RBM, et aussi pour apprendre des représentations de la famille de protéines en question. Comme nous allons le voir par la suite, les poids de la RBM dans sa phase compositionnelle peuvent encoder

des fonctionnalités (Tubiana et al., 2019a,b), qui permettent de séparer et classifier les différentes sous-familles d'une famille de protéines.

Résultats

Leurs expériences consistent à mesurer les effets des mutations d'acide aminés sur une hélice α de la protéine TEM-1 à travers l'évaluation de la valeur sélective du mutant. La valeur sélective mesure le taux de croissance d'une bactérie porteur de ce mutant dans un milieu avec antibiotique. Plus cette valeur sélective est élevée, plus le mutant dégrade le médicament, et donc plus la bactérie est résistante à l'antibiotique. Nous avons développé une procédure d'inférence pour déterminer cette valeur sélective à partir des données expérimentales.

De notre côté, nous avons pu donner un score à chacun de ces mutants grâce à nos modèles de Potts, ou bien nos RBM. Les prédictions théoriques et expérimentales sont fortement corrélées, mais non linéairement.

L'objectif de l'expérience était aussi de caractériser l'épistasie. L'épistasie fait référence à la dépendance contextuelle des effets de la mutation, et plus précisément dans notre cas, aux interactions entre mutations qui se traduisent par la non-additivité des effets sur la valeur sélective (en échelle logarithmique). L'épistasie entre les mutations A et B peut être estimée comme l'écart entre le logarithme de valeur sélective observée des doubles mutants, AB, et la somme des logarithmes des valeurs sélectives des deux mutations individuelles (A et B).

La relation entre la valeur sélective et les énergies de nos modèles étant non-linéaire, les prédictions d'épistasie à partir de la différence des énergies n'est pas un indicateur pertinent.

Pour expliquer l'épistasie mesurée dans l'expérience, nous avons développé un modèle à deux niveaux

$$\log\left(\frac{W}{W_0}\right) = \log\left(1+\exp\left(\frac{\Delta G_0}{RT}\right)\right) - \log\left(1+\exp\left(\frac{\Delta G_0+\Delta\Delta G}{RT}\right)\right). \tag{4}$$

où W (respectivement W_0) est la valeur sélective du mutant (respectivement de TEM-1). Nous avons inféré les paramètres du modèle de stabilité $\Delta\Delta G$ et ΔG_0. ΔG_0 est un paramètre global et chaque mutant simple A a son propre $\Delta\Delta G_A$. Une hypothèse importante est l'additivité de ces paramètres : le paramètre du double mutant AB s'écrit comme la somme des paramètres des mutants simples A et B ($\Delta\Delta G_{AB} = \Delta\Delta G_A + \Delta\Delta G_B$). Ce modèle reproduit bien les données observées expérimentales, malgré certaines déviations. Un résultat intéressant est la linéarité observée entre les paramètres du modèle de stabilité et l'énergie de nos modèles entraînés sur les alignements de séquences : en combinant les énergies de nos modèles et le modèle à deux niveaux, nous sommes en mesure de prédire le signe de l'épistasie expérimentale pour un grand nombre de mutants. Ces résultats sont détaillés dans une publication à venir.

Nous avons aussi analysé l'évolution de la valeur sélective et de l'épistasie en fonction de la concentration d'antibiotique dans le milieu. Nous avons observé des effets de saturation de la valeur sélective : un mutant est viable jusqu'à une certaine concentration d'antibiotique. Ces effets de saturation ont un effet direct sur l'évolution de l'épistasie en fonction de la concentration, créant des effets de saturation ainsi qu'une évolution non monotone en fonction de la concentration, tout en gardant un signe constant. Ces effets sont qualitativement reproduits par un modèle à deux niveaux où ΔG_0 dépend désormais de la concentration.

Nous avons enfin utilisé nos machines de Boltzmann restreintes pour analyser la classe

A de la famille des β-lactamases. En nous plaçant dans la phase compositionnelle des RBM (Tubiana and Monasson, 2017), phase où un nombre faible d'unités cachées sont activées simultanément pour représenter une séquence, nous avons pu extraire des poids ayant un sens biologique. De plus, nous avons pu nous servir de ces poids pour séparer les différentes séquences en différentes sous-familles caractérisées expérimentalement par Philippon et al. (2016, 2019).

Restricted Boltzmann Machines: an overview from a machine learning and statistical mechanics perspectives

Part I summary

Restricted Boltzmann Machines are neural networks that have been studied by several scientific communities, be it information, statistical physicists, or neuroscientists, for different purposes.

This first part aims to introduce Restricted Boltzmann Machines, and to understand their influential parameters, and why they have attracted different communities.

In Chapter 1, we will present in detail the different training algorithms of these machines, which are mainly based on variants of Alternating Gibbs Sampling, which will be the main subject of study in Chapter 3.

Chapter 2 will make the link between classical models of statistical physics and RBM. As we will see, the RBM has an additional richness compared to these models thanks to its representations in its hidden space. These representations will be the key to go further than Alternating Gibbs Sampling, and improve the sampling of these models. We will also detail the known results on RBM derived by the statistical physics community, which will be useful later on, when we will study the sampling properties of RBM in Chapters 3 and 4, and we use RBM for analyzing class A β-lactamases in Chapter 9.

This part will give the necessary tools to understand Part II, which is the heart of my work on the RBM presented in this manuscript.

1

Restricted Boltzmann Machines: an introduction

This chapter presents the Restricted Boltzmann Machines (RBM) from different points of view. First, from a historical point of view, to understand why RBM are such particular objects of study and so appreciated by statistical physicists. Then we will describe these networks and focusing on technical issues, which concern the training of RBM and the evaluation of its partition function. Finally, we describe briefly the various applications of RBM and datasets we used for numerical experiments.

1.1. Historical background

Since the XIX$^{\text{th}}$ century and the work of the pioneers in thermodynamics James Clerk Maxwell, Ludwig Boltzmann, and Josiah Willard Gibbs, statistical physicists have been interested in the macroscopic properties of large ensembles of microscopic particles. These works have led to important theoretical advances and the development of many concepts, such as the Boltzmann distribution or the entropy of a system. As Philip Warren Anderson pointed out in his famous article "More is different": "the behavior of large and complex aggregates of elementary particles, it turns out, is not to be understood in terms of a simple extrapolation of the properties of a few particles." (Anderson, 1972). Concepts like symmetry breaking or phase transitions have been introduced to understand these differences between finite and infinite numbers of particles. Many models of interacting particles, later called spins, have been developed to understand these critical phenomena, such as magnetism (Ising, 1925; Onsager, 1944). To model these phenomena, these models describe the behavior of a large number N of spins $v_i = \pm 1$ ($\mathbf{v} = \{v_i\}_{i=1...N}$) via a Hamiltonian $E(\mathbf{v}|\Theta)$ which describes the interactions between the spins. Θ describe here the ensemble of parameters of the Hamiltonian. Critical phenomena emerge in the so-called thermodynamic limit $N \to \infty$ and can be understood as changes of some quantities, called order parameters.

In the early 1980s, these models were applied to other fields, such as neuroscience and artificial intelligence. From the 1980s onwards, these models of interacting spins have been used to describe neural networks. These spins $v_i \pm = 1$ represent McCulloch Pitts neurons (McCulloch and Pitts, 1943), and the interactions described in the Hamiltonian $E(\mathbf{v}|\Theta)$ correspond to the synapses connecting these different neurons. One of the first milestones is due to John Hopfield and William Little, who showed that a neural network could store memories (called patterns) (Little, 1974; Little and Shaw, 1975; Hopfield, 1982). In the Hopfield model, the parameters Θ of the Hamiltonian are fixed by Hebb's rule (Hebb, 1949), which states that "cells that fire together, wire together.". In 1985, Geoffrey Hinton and Terrence Sejnowski introduced the Boltzmann Machines (BM) (Hinton and Sejnowski, 1983), where the parameters of the Hamiltonian are restricted to fields and

couplings ($\Theta = \{g_i, J_{ij}\}$) and can be tuned through Monte Carlo simulations (Ackley et al., 1985). Instead of having a fixed learning rule such as Hebb's rule in the Hopfield model, the parameters Θ of the BM are learned to store the patterns. In 1986, Paul Smolensky introduced the Restricted Boltzmann Machines (called Harmonium initially), which are a special case of BM with two distinct layers (Smolensky, 1986).

Although BM and RBM have shown their ability in several tasks, they were abandoned at the end of the 1980s because of their difficult training. They were supplemented by other models with different architectures whose learning is based on backpropagation (Rumelhart et al., 1986). In 2002, Geoffrey Hinton developed an efficient algorithm for training RBM called Contrastive Divergence (CD) (Hinton, 2002). This algorithm caused a surge in the use of RBM: RBM will be used as building blocks for deeper networks, such as Deep Belief Networks (DBN) (Hinton and Salakhutdinov, 2006; Hinton et al., 2006; Salakhutdinov and Murray, 2008) or Deep Boltzmann Machines (DBM) (Salakhutdinov and Hinton, 2009; Salakhutdinov and Larochelle, 2010). These deep structures based on a stack of RBM have shown their ability to learn useful data representations and to generate new data (Bengio et al., 2013).

Nevertheless, these networks were gradually abandoned by the machine learning community at the end of the 2010s, due in part to technical improvements in hardware, as well as to the constitution of large training databases, which allowed the emergence of new deep neural networks, such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), Convolutional Neural Networks (CNN) (LeCun et al., 1998), Variational Auto Encoders (VAE) (Kingma and Welling, 2014) or Generative Adversarial Network (GAN) (Goodfellow et al., 2014). For an introduction to Deep Learning, we invite the reader to read the book "Deep Learning" by Goodfellow et al. (2016).

However, RBM have remained very popular among physicists as an object of theoretical studies (Agliari et al., 2012; Tubiana and Monasson, 2017; Decelle et al., 2017, 2018; Barra et al., 2018; Hartnett et al., 2018; Decelle and Furtlehner, 2020a,b; Alberici et al., 2020; Leonelli et al., 2021; Roussel et al., 2021; Decelle et al., 2021). They have also shown their usefulness in some fields, such as extracting interesting representations in proteins families or for antigens (Tubiana et al., 2019a,b; Shimagaki and Weigt, 2019; Bravi et al., 2021a,b).

1.2. Description of the Restricted Boltzmann Machines

1.2.1 Boltzmann Distribution

Restricted Boltzmann Machines are undirected probabilistic graphical models with two layers. A visible layer $\mathbf{v}$, which represents the data, is connected to a hidden layer $\mathbf{h}$ through a weight matrix $\mathbf{W}$ (Fig. 1.1(a)).

The visible layer includes N units v_i, and the hidden layer M units h_μ, which can take discrete or continuous values. The joint probability distribution of the visible configuration $\mathbf{v} = \{v_i\}_{i=1\dots N}$ and of the hidden configuration $\mathbf{h} = \{h_\mu\}_{\mu=1\dots M}$ reads

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\left(-E(\mathbf{v},\mathbf{h})\right), \tag{1.1}$$

The energy $E(\mathbf{v},\mathbf{h})$ is equal to

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{N}\sum_{\mu=1}^{M} W_{i\mu} v_i h_\mu + \sum_{\mu=1}^{M}\mathcal{U}_\mu(h_\mu) + \sum_{i=1}^{N}\mathcal{V}_i(v_i). \tag{1.2}$$

In the formula above, $\mathcal{U}_\mu$ and $\mathcal{V}_i$ are potentials acting on, respectively, h_μ and v_i.

Depending on the nature of the data and on the applications, many parametric potentials are used in the literature, such as

(a) Restricted Boltzmann Machines (b) Alternating Gibbs Sampling

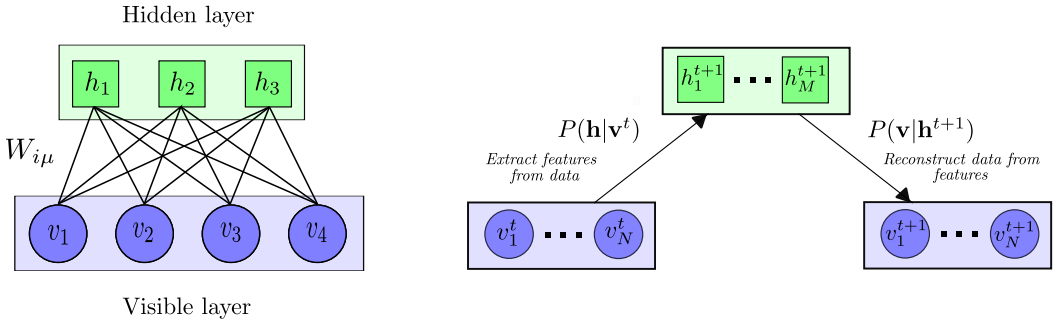


Figure 1.1: Description of the Restricted Boltzmann Machines and of Alternating Gibbs Sampling. (a) Bipartite architecture of RBM, with the visible (blue) and hidden (green) layers. (b) Alternating Gibbs Sampling: hidden and visible configurations are conditionally sampled from one another.

- Bernoulli potential: $U(x) = -gx$, with $x \in \{0,1\}$
- Spin potential: $U(x) = -gx$, with $x \in \{-1,1\}$
- Potts potential with q-states: $U(x) = -g(x)$, with $x \in [\![1,q]\!]$
- Gaussian potential: $U(x) = \frac{1}{2}\gamma x^2 + \theta x$, $x \in \mathbb{R}$
- ReLU potential: $U(x) = \frac{1}{2}\gamma_+ x^2 + \theta_+ x$, $x \in \mathbb{R}^+$
- dReLU potential: $U(x) = \frac{1}{2}\gamma_+ x_+^2 + \theta_+ x_+ + \frac{1}{2}\gamma_- x_-^2 + \theta_- x_-$, $x \in \mathbb{R}^+$, $x \in \mathbb{R}, x_+ = \max(0,x), x_- = \min(0,x)$.

The properties of the RBM depend crucially on these potentials. In the following, when the knowledge of potentials is essential, we will put them forward in the following way: Spin-Gaussian RBM denotes a RBM with Spin potential acting on the visible units and Gaussian potential acting on the hidden units. As an example, Spin-Gaussian RBM can represent the Hopfield model, (Agliari et al., 2012) and Bernoulli-Bernoulli RBM are known to be universal approximators (i.e., can approximate any distribution over the visible variables) when its number of hidden units goes to infinity ($M \to \infty$) (Le Roux and Bengio, 2008). We will come back to these two models in detail in Chapter 2.

RBM learn a joint Boltzmann distribution $P(\mathbf{v},\mathbf{h})$ by maximizing the log-likelihood of the data configurations:

$$P(\mathbf{v}) = \int d\mathbf{h}\, P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\Big(-E^{\text{eff}}(\mathbf{v})\Big), \tag{1.3}$$

with

$$E^{\text{eff}}(\mathbf{v}) = \sum_{i=1}^{N} \mathcal{V}_i(v_i) - \sum_{\mu=1}^{M} \Gamma_\mu(I_\mu(\mathbf{v})), \tag{1.4}$$

where

$$I_\mu(\mathbf{v}) = \sum_{i=1}^{N} W_{i\mu} v_i, \tag{1.5}$$

is the input received by hidden unit h_μ and

$$\Gamma_\mu(I) = \log\left(\int dh \exp\left(-\mathcal{U}_\mu(h) + hI\right)\right), \tag{1.6}$$

is the cumulative generative function associated with the potential $\mathcal{U}_\mu$. Parameters $\Theta \equiv \{W_{i\mu}, \mathcal{U}_\mu, \mathcal{V}_i\}$ modulate the energy landscape $E^{\text{eff}}(\mathbf{v})$.

Similarly, we can define

$$P(\mathbf{h}) = \int d\mathbf{v}\, P(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp\Big(-E^{\text{eff}}(\mathbf{h})\Big), \tag{1.7}$$

with

$$E^{\text{eff}}(\mathbf{h}) = \sum_{\mu=1}^{M} \mathcal{U}_\mu(h_\mu) - \sum_{i=1}^{N} \Gamma_i(I_i(\mathbf{h})), \tag{1.8}$$

where

$$I_i(\mathbf{h}) = \sum_{\mu=1}^{M} W_{i\mu} h_\mu, \tag{1.9}$$

is the input received by visible unit v_i and

$$\Gamma_i(I) = \log\left(\int dv \exp\left(-\mathcal{V}_i(v) + vI\right)\right). \tag{1.10}$$

1.2.2 Sampling

If the set of parameters Θ is known, the RBM model distribution is fully defined. Then, Alternating Gibbs Sampling between the visible and hidden layers is used to generate samples from $P(\mathbf{v})$.

The pseudocode of AGS is given in Algorithm 1 (Fig. 1.1(b)). It is mainly composed of two steps:

- Starting from a visible configuration $\mathbf{v}^t$ at time t, a hidden configuration $\mathbf{h}^{t+1}$ is drawn from $P(\mathbf{h}|\mathbf{v}^t)$. This step can be seen as a stochastic feature extraction from the configuration $\mathbf{v}^t$.
- A new visible configuration $\mathbf{v}^{t+1}$ is drawn from $P\left(\mathbf{v}|\mathbf{h}^{t+1}\right)$. This step can be seen as a stochastic reconstruction of $\mathbf{v}$ from the latent configuration $\mathbf{h}^{t+1}$.

Algorithm 1: Alternating Gibbs Sampling

Choose a random vector $\mathbf{v}^0$;
for $t \in [\![0, T]\!]$ **do**
  $\mathbf{h}^{t+1} \sim P(\mathbf{h}|\mathbf{v}^t)$;
  $\mathbf{v}^{t+1} \sim P\left(\mathbf{v}|\mathbf{h}^{t+1}\right)$;
end

The properties of AGS for sampling an energy landscape will be studied in detail in Chapter 3.

1.2.3 Representation in the hidden space

As seen in the previous part, it is easy to go from the visible space to the hidden space (and vice versa) with RBM, through the stochastic mappings $P(\mathbf{h}|\mathbf{v})$ and $P(\mathbf{v}|\mathbf{h})$ which can be factorized. It is therefore possible to define the representation of a vector $\mathbf{v}$ of the visible layer in the space of the hidden units as the most probable $\mathbf{h}$ given $\mathbf{v}$

$$h_\mu^* = \underset{h_\mu}{\operatorname{argmax}} P(h_\mu|\mathbf{v}) \equiv f_\mu(\mathbf{v}) = (\mathcal{U}'_\mu)^{-1}(I_\mu(\mathbf{v})). \tag{1.11}$$

We will see in Chapters 3 and 4 that these representations can be used to improve the sampling of $E^{\text{eff}}(\mathbf{v})$.

1.2.4 Log-likelihood estimation

In the general case, as the evaluation of the partition function Z required the summation over an exponential number of terms (*e.g.*, 2^N terms for binary visible units), it is impossible to compute the RBM partition function Z, and therefore, to have direct access to the log-likelihood. Nevertheless, it is essential to be able to estimate this log-likelihood to compare the performances of our different RBM during the training.

One way to estimate the log-likelihood directly is to use a strategy called Annealed Importance Sampling (AIS) (Jarzynski, 1997; Neal, 2001; Salakhutdinov and Murray, 2008), and was used by Jérôme Tubiana during his PhD Thesis (Tubiana, 2018)[1].

The idea of this method is to progressively interpolate our distribution of interest $P(\mathbf{v})$ with a distribution $P_0(\mathbf{v})$ whose partition function can be calculated exactly. $P_0(\mathbf{v})$ is the distribution of a RBM with no weights, chosen as the closest independent distribution to the empirical distribution of the data in terms of Kullback-Leibler divergence. A set of R intermediate distribution $P_{\beta_r}(\mathbf{v}) \propto P(\mathbf{v})^{\beta_r} P_0(\mathbf{v})^{1-\beta_r}$ is built, with $\beta_1 = 0$, $\beta_{r+1} > \beta_r$ and $\beta_R = 1$. As $P(\mathbf{v})$ and $P_0(\mathbf{v})$ are Boltzmann distributions, $P_{\beta_r}(\mathbf{v})$ is also a Boltzmann distribution and its energy is simply a linear interpolation with weight β_r between the energy of $P(\mathbf{v})$ and $P_0(\mathbf{v})$. As

$$\frac{Z}{Z_0} = \left\langle \frac{P(\mathbf{v})}{P_0(\mathbf{v})} \right\rangle_{\mathbf{v}\sim P_0} = \left\langle \prod_{r=1}^{R-1} \frac{P_{\beta_{r+1}}(\mathbf{v})}{P_{\beta_r}(\mathbf{v})} \right\rangle_{\mathbf{v}\sim P_0}, \tag{1.12}$$

the partition function Z can be computed by starting with configuration $\mathbf{v}$ sampled from $P_0(\mathbf{v})$ and progressively annealed to $P(\mathbf{v})$ through $P_{\beta_r}(\mathbf{v})$ with Alternating Gibbs Sampling.

1.3. Training Restricted Boltzmann Machines

For a given training set of K samples, $\{\mathbf{v}^k\}_{k=1\ldots K}$, the parameters Θ are found by maximizing the log-likelihood of the data,

$$\text{LL} \equiv \langle \log P(\mathbf{v}) \rangle_{\text{data}} \equiv \frac{1}{K} \sum_{k=1}^{K} \log P(\mathbf{v}^k). \tag{1.13}$$

[1] Another possible way is to use a proxy to the log-likelihood, called pseudolikelihood, see Section 6.3.2 for its description in the case of Potts model.

The maximization is done by gradient ascent. The general expression for the gradients is

$$\frac{\partial \mathrm{LL}}{\partial \Theta} = -\left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{data}} + \left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{model}}, \tag{1.14}$$

where $\langle . \rangle_{\mathrm{data}}$ denotes the expected value over the data and $\langle . \rangle_{\mathrm{model}}$ over the model.

In practice, Stochastic Gradient Ascent (SGA) (Bottou, 2010; Ruder, 2017) is used rather than Gradient Ascent (GA). This means that the expected value over the data is not calculated on all the data, but on a random subset of size B of the training data, called minibatch, see Algorithm 2. SGA is known to be quicker and has better behavior for non-convex optimization. η denotes the learning rate. We use learning rate annealing during the training, *i.e.* the learning rate is a decreasing function over time, to ensure convergence to a local maximum of the log-likelihood. For more details and tips on training RBM, we recommend the following papers (Hinton, 2002; Fischer and Igel, 2014; Tubiana, 2018).

Algorithm 2: Stochastic Gradient Ascent

for $t \in [\![1, T]\!]$ **do**
- Choose a random minibatch of size B $\{\mathbf{v}^b\}_{b=1\ldots B}$ in $\{\mathbf{v}^k\}_{k=1\ldots K}$;
- Compute $\left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{minibatch}}$;
- Compute $\left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{model}}$;
- Compute $\frac{\partial \mathrm{LL}}{\partial \Theta}$ (Eq. (1.14));
- $\Theta = \Theta + \eta \frac{\partial \mathrm{LL}}{\partial \Theta}$

end

$\left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{minibatch}}$ can be calculated quickly at each time step and depends only on the training data. However, expected values over the distribution $\left\langle \frac{\partial E^{\mathrm{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\mathrm{model}}$ are generally not tractable, because they require the summation over an exponential number of terms, like the partition function Z. Therefore, these expected values are estimated through Monte Carlo (MC) methods (Metropolis and Ulam, 1949; Hastings, 1970). The basic idea behind MC methods is to generate samples from $P(\mathbf{v})$ in order to compute these expected values. A particular branch of these MC methods is the Monte Carlo Markov Chains (MCMC). The idea is to build a Markov Chain with $P(\mathbf{v})$ as equilibrium distribution, and to record the different states of the chain in order to have samples from $P(\mathbf{v})$. AGS described in Algorithm 1 is such a Markov Chain. It is well known that these methods suffer from several problems. First, it is necessary to wait a certain number of time steps to reach the equilibrium distribution $P(\mathbf{v})$. This number of time steps, called burn-in time, is difficult to evaluate theoretically and can be very large. Therefore, without a good initialization of these chains, the convergence to the equilibrium distribution is very expensive in time and computation without a good initialization of these chains. This first point explains why RBM were abandoned in the late 1980s. Second, these methods can suffer from poor mixing: sampled configurations can be trapped in one of the regions of high probability, *i.e.*, of low energy, while other favorable regions are not dynamically explored. This second point will be discussed in Chapter 3.

Nevertheless, methods have been developed to overcome these limitations and allow training RBM. These algorithms, mostly based on Alternating Gibbs Sampling between

the visible and hidden layers, are used to generate samples from $P(\mathbf{v})$ to compute these expected values. We will quickly describe these different algorithms.

1.3.1 Contrastive Divergence

Geoffrey Hinton introduced Contrastive Divergence in 2002 (Hinton, 2002; Hinton et al., 2006). Instead of using random configuration $\mathbf{v}$ as initialization of Markov Chains, the idea of CD-k (Contrastive Divergence with k steps) is to use the data of the current minibatch $\{\mathbf{v}^b\}_{b=1...B}$ and then to run AGS for k steps. The expected values $\left\langle \frac{\partial E^{\text{eff}}(\mathbf{v})}{\partial \Theta} \right\rangle_{\text{minibatch}}$ are computed on the B samples obtained after the k steps of AGS. In practice, k is small (from 1 to 10). Maximizing log-likelihood means maximizing the probability of the data. Therefore, when initializing the Markov Chain with a training data, the chain must be in a high probability region. Therefore, CD-k is much faster than MCMC with random initialization because it avoids the long burn-in time. This algorithm allowed to train the RBM on real datasets, such as the famous MNIST handwritten digits' dataset (LeCun, 1998). Nevertheless, the advantage of this algorithm is also its main weakness. By initializing the chains next to the training data, only the configurations close to them are sampled. Therefore, if spurious maxima of $P(\mathbf{v})$ are created far away from the data during the training, they are never sampled and therefore persist during training (as the expected value on the data does not take them into account, and as CD-k can not sample configurations near them). Once the RBM is trained, by sampling new configurations from random configurations $\mathbf{v}$, RBM can get stuck in these spurious maxima.

1.3.2 Persistent Contrastive Divergence

Tijmen Tieleman introduced a variant of CD, called Persistent Contrastive Divergence (PCD) (Tieleman, 2008; Tieleman and Hinton, 2009). Instead of resetting the Markov chains at each time step with the training data, the last visible configurations used to compute the expected values over the model are kept in memory and used as initialization for the next time step. As in CD-k, PCD-k (Persistent Contrastive Divergence with k steps) consists of k steps of AGS with these initial configurations. The idea behind this is to consider that if these data are well sampled according to the distribution $P(\mathbf{v})$ at a time t, after one SGA step, these data should be almost at equilibrium provided that the learning rate is small enough (Younes, 1999). Thus, theoretically, PCD-k allows the exploration of remote regions of the training data. However, in practice, if the energy barriers are too large, the dynamics is trapped in a local minimum of the energy landscape and can not explore efficiently the landscape. Therefore, as CD-k, PCD-k can lead to bad solutions and to divergence of the likelihood (Fischer and Igel, 2010).

Nevertheless, we use this algorithm for our trainings, because it is a good compromise between the computation time and the performances reached.

1.3.3 Parallel Tempering

Another possible algorithm is the Parallel Tempering (PT), also known as replica exchanges MCMC (Swendsen and Wang, 1986; Geyer, 1991). The idea is to run several Markov Chains at different temperatures and allows exchanges of configurations between them at each time step. The distribution at inverse temperature β reads

$$P_\beta(\mathbf{v},\mathbf{h}) = \frac{1}{Z_\beta} \exp\left(-\beta E(\mathbf{v},\mathbf{h})\right). \tag{1.15}$$

$\beta = 1$ corresponds to the original distribution (Eq. 1.1). Markov chains at high temperature, *i.e.*, low inverse temperature $\beta < 1$, mixes better than between the regions of

high probability than the one at $\beta = 1$. However, at low inverse temperature $\beta < 1$, $P_\beta(\mathbf{v}, \mathbf{h})$ losts track of the details of the distribution. Replica exchanges between the different chains exploit the fast-mixing properties at low β and high-quality model at $\beta = 1$. The algorithm was adapted to train RBM (Salakhutdinov, 2009; Desjardins et al., 2010b; Cho et al., 2010).

In practice, the marginal

$$\tilde{P}_\beta(\mathbf{v}) = \int d\mathbf{h}\, P_\beta(\mathbf{v}, \mathbf{h}), \tag{1.16}$$

is used with the same initialization of the Markov Chains as in PCD-k. R chains at temperature $\beta_r \in [0,1]$ are used, with $\beta_1 = 1$ and $\beta_{r+1} < \beta_r$. $\mathbf{v}_{\beta_r}$ denotes configurations generated at inverse temperature β_r with AGS. During sampling, swaps between $\{\mathbf{v}_{\beta_r} = \mathbf{v}^t_{\beta_r}, \mathbf{v}_{\beta_{r+1}} = \mathbf{v}^t_{\beta_{r+1}}\}$ and $\{\mathbf{v}_{\beta_r} = \mathbf{v}^t_{\beta_{r+1}}, \mathbf{v}_{\beta_{r+1}} = \mathbf{v}^t_{\beta_r}\}$ are possible with an acceptance ratio

$$\begin{aligned} & A_r\left(\{\mathbf{v}_{\beta_r} = \mathbf{v}^t_{\beta_r}, \mathbf{v}_{\beta_{r+1}} = \mathbf{v}^t_{\beta_{r+1}}\} \to \{\mathbf{v}_{\beta_r} = \mathbf{v}^t_{\beta_{r+1}}, \mathbf{v}_{\beta_{r+1}} = \mathbf{v}^t_{\beta_r}\}\right) \\ = \quad & \min\left(1, \frac{\tilde{P}_{\beta_r}(\mathbf{v}^t_{\beta_{r+1}})\tilde{P}_{\beta_{r+1}}(\mathbf{v}^t_{\beta_r})}{\tilde{P}_{\beta_{r+1}}(\mathbf{v}^t_{\beta_{r+1}})\tilde{P}_{\beta_r}(\mathbf{v}^t_{\beta_r})}\right), \end{aligned} \tag{1.17}$$

in order to satisfy the detailed balance. Parallel Tempering improves trainings but has some disadvantages. First, for each chain, unlike CD and PCD, we have to simulate R Markov chains instead of one, which is numerically expensive. Secondly, it is difficult to choose correctly the inverse temperatures, as a linear interpolation between 0 and 1 is not necessarily a good idea.

1.3.4 Deep Tempering

The last method based on sampling is called Deep Tempering (DT) (Desjardins et al., 2014) (Fig. 1.2(b)). This algorithm was first introduced to train Deep Belief Networks (DBN) (Hinton and Salakhutdinov, 2006; Salakhutdinov and Murray, 2008). DBN are stacks of N RBM (Fig. 1.2(a)). The number of hidden units of the n^{th} RBM is equal to the number of visible units of the $(n+1)^{\text{th}}$ RBM. Usually, these networks are trained greedily, one RBM by one RBM, from bottom to top (Hinton et al., 2006; Bengio et al., 2007). The bottom RBM is trained on the data $\{\mathbf{v}^k\}_{k=1\ldots K}$. After the training, the hidden representations $\{\mathbf{h}^k_1\}_{k=1\ldots K}$ are drawn from $P_1\left(\mathbf{h}|\mathbf{v}^k\right)$. These hidden representations are used to train the second RBM of the stack. And so on.

Each RBM of the stack has its own visible landscape $E^v_n(\mathbf{v})$ (respectively hidden landscape $E^h_n(\mathbf{h})$) associated with the Boltzmann distribution $P^v_n(\mathbf{v})$ (respectively $P^h_n(\mathbf{h})$).

Deep Tempering consists in training the N RBM of the stack at the same time. The idea is to used Alternating Gibbs Sampling for all RBM. A Gibbs step at time t is defined by $\mathbf{h}^t_n \sim P_n\left(\mathbf{h}|\mathbf{v}^{t-1}_n\right)$ and $\mathbf{v}^t_n \sim P_n\left(\mathbf{v}|\mathbf{h}^t_n\right)$.

In the manner of Parallel Tempering, replica exchange between the visible configuration $\mathbf{v}^t_{n+1}$ of the $(n+1)^{\text{th}}$ RBM and the hidden configuration $\mathbf{h}^t_n$ of the n^{th} RBM is allowed. These configurations are swapped with probability

$$\begin{aligned} & A_n\left(\{\mathbf{h}_n = \mathbf{h}^t_n, \mathbf{v}_{n+1} = \mathbf{v}^t_{n+1}\} \to \{\mathbf{h}_n = \mathbf{v}^t_{n+1}, \mathbf{v}_{n+1} = \mathbf{h}^t_n\}\right) \\ = \quad & \min\left(1, \frac{P^v_{n+1}(\mathbf{h}^t_n)P^h_n(\mathbf{v}^t_{n+1})}{P^v_{n+1}(\mathbf{v}^t_{n+1})P^h_n(\mathbf{h}^t_n)}\right) = \min\left(1, \frac{\exp\left(-E^v_{n+1}(\mathbf{h}^t_n) - E^h_n(\mathbf{v}^t_{n+1})\right)}{\exp\left(-E^v_{n+1}(\mathbf{v}^t_{n+1}) - E^h_n(\mathbf{h}^t_n)\right)}\right) \end{aligned} \tag{1.18}$$

With this acceptance ratio, the detailed balance is satisfied (Appendix C.1). In practice, we have not used this algorithm to train RBM. Nevertheless, we have used it to improve the sampling of RBM after their training, see Chapter 4 and Algorithm 6.

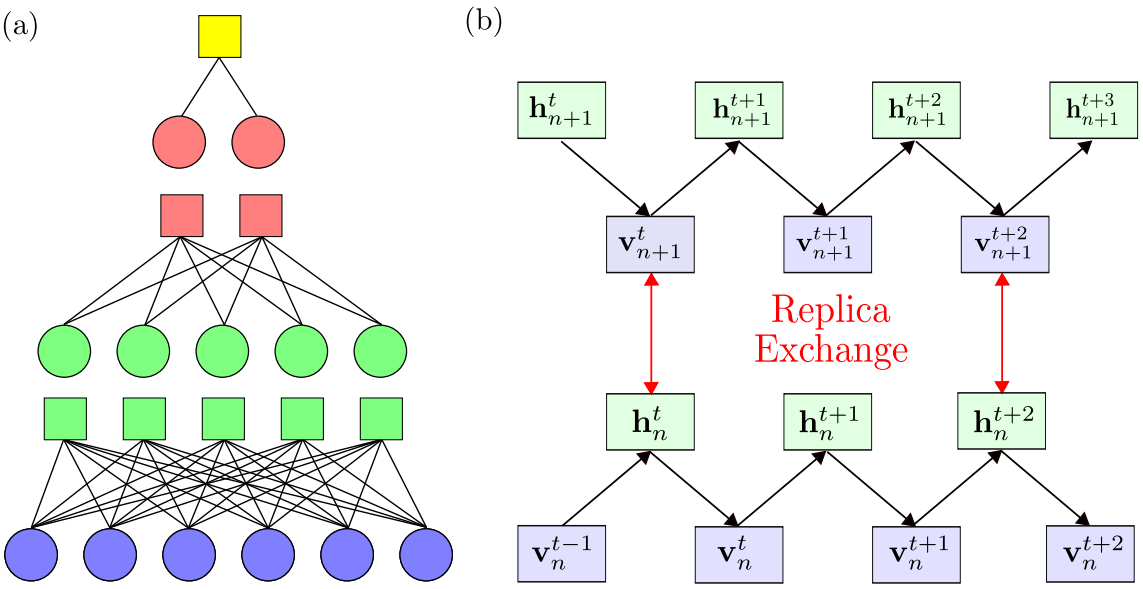


Figure 1.2: (a) Example with three RBM. The number of visible units of the $(n+1)^{\text{th}}$ RBM is equal to the number of hidden units of the n^{th} RBM. (b) Illustration of Deep Tempering (here for the n^{th} and $(n+1)^{\text{th}}$ RBM). Alternative Gibbs sampling is used to generate configurations. The dynamics are coupled. Replica exchanges are possible between the visible configurations $\mathbf{v}_{n+1}$ of the $(n+1)^{\text{th}}$ RBM and the hidden configurations $\mathbf{h}_n$ of the n^{th} RBM. These configurations are swapped with a probability $A_n(\mathbf{h}^t_n, \mathbf{v}^t_{n+1})$ (Eq. (1.18)). This algorithm was first introduced in (Desjardins et al., 2014) to train Deep Belief Networks.

1.3.5 Mean-field and TAP methods

Some methods based on high-temperature approximation of the log partition $\log Z$ have been developed to train RBM (Welling and Hinton, 2002; Gabrié et al., 2015; Tramel et al., 2018). This approximation has been initially developed for spin glasses (Georges and Yedidia, 1991). Basically, in the case of RBM, the expected value of the model is replaced by the average on the fixed points of the following self-consistent equations. At first order, for a Spin-Spin RBM without any fields, the self-consistent equations read

$$\mathbf{m}_v = \tanh\left(\mathbf{W}\cdot\mathbf{m}_h\right), \tag{1.19}$$

$$\mathbf{m}_h = \tanh\left(\mathbf{W}^T\cdot\mathbf{m}_v\right). \tag{1.20}$$

This approximation is justified only when the weights $W_{i\mu}$ are weak or in tree-like graphs of interactions.

1.3.6 Wasserstein metric method

Maximizing the log-likelihood is equivalent to minimizing the Kullback-Leibler (KL) divergence between $P(\mathbf{v})$ and the empirical distribution $P_d(\mathbf{v}) = \frac{1}{K}\sum_{k=1}^{K}\prod_{i=1}^{N}\delta_{v_i,v_i^k}$ as

$$D_{KL}(P_d(\mathbf{v})||P(\mathbf{v})) = \sum_{\mathbf{v}} P_d(\mathbf{v}) \log\left(\frac{P_d(\mathbf{v})}{P(\mathbf{v})}\right) \tag{1.21}$$

$$= -\mathrm{LL} + \sum_{\mathbf{v}} P_d(\mathbf{v}) \log\left(P_d(\mathbf{v})\right), \tag{1.22}$$

Nevertheless, Kullback-Leibler divergence is not the only measure of similarity between two distributions. Wasserstein metric used in optimal transport is also a relevant similarity measure between two distributions (Monge, 1781; Kantorovich, 1942). This distance is widely used to train GAN (Arjovsky et al., 2017) but has also been used for training RBM (Montavon et al., 2016) and does not have the same properties as KL divergence (Fig. 1.3). Wasserstein RBM relies on PCD to generate configurations and seems to have the same performance as the maximization of the log-likelihood. For training on MNIST, Wasserstein RBM produce nice digits but are less diverse than digits in the dataset or digits generated with a classical RBM (Montavon et al., 2016).

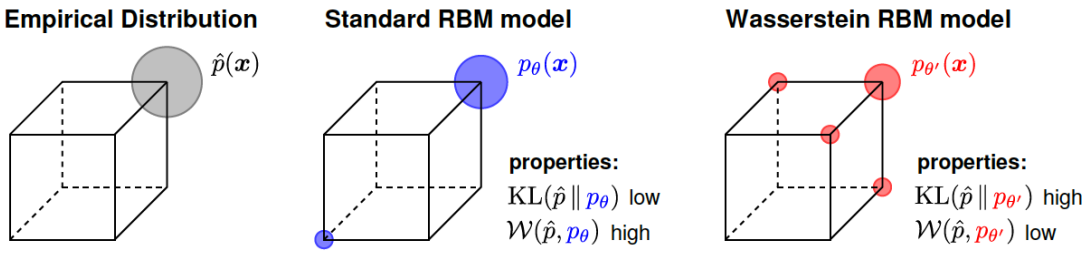


Figure 1.3: Empirical distribution $\hat{p}(x)$ (gray) defined on the set of states $\{0,1\}^d$, with $d=3$ shown next to two possible modeled distributions defined on the same set of states. The size of the circles indicates the probability mass allocated to each state. The first modeled distribution $p_\theta(x)$ (blue) has low KL divergence and high Wasserstein distance from the empirical distribution. The second one $p_{\theta'}(x)$ (red) has high KL divergence and low Wasserstein distance, and thus incorporates the desired metric. Figure and caption from Montavon et al. (2016)

.

1.3.7 With quantum annealer

More recently, a new variant has been proposed by Dixit et al. (2021), where the configurations used to compute the gradient are sampled using a D-Wave quantum annealer[2], although with the current technical limitations concerning the number of qubits, this solution only allows to train small size networks.

1.4. What are RBM for?

This section will quickly review the possible applications of RBM, and its different variations, in the machine learning community.

1.4.1 RBM as building blocks of deep neural networks

Back in the 2010s, RBM were used as building block of deep neural networks. As explained in Section 1.3.4, RBM can be stacked to form a Deep Belief Network. DBN can

[2] https://www.dwavesys.com/

be trained quickly, one RBM by one RBM, from the bottom RBM to the top one (Hinton et al., 2006; Bengio et al., 2007). This procedure allows to quickly train a DBN even if it is not optimal, as the weights of the RBM are not trained simultaneously, but only one layer by one layer. This is why this procedure is often considered as a pretraining of the DBN. After the pretraining, DBN is a generative network, and its weights can be fine-tuned with the wake-sleep algorithm (Hinton et al., 1995, 2006). Once trained, DBN can generate new data with Gibbs sampling. Another fashionable application in the 2000s was to consider DBN as feedforward neural networks, with as input the visible layer of the bottom RBM, and as output the hidden layer of the top RBM. This analogy allows to train quickly feedforward neural network. The idea is the following: consider a dataset with data and labels, and you want to train a fully connected feedforward neural network to predict the labels. Considering only the architecture (and not the dynamics), the feedforward neural network has the same architecture as the DBN. So, you can use unsupervised pretraining of DBN on the data, by maximizing the log-likelihood of each RBM one by one, in order to extract useful features from the data (Erhan et al., 2010). Then, consider DBN as a feedforward network, its weights can be fine-tuned with backpropagation, to predict the labels from the data. As DBN has learned useful representation of the data, it is easier to fine-tune its weights than to train the feedforward network from scratch. This procedure has been widely used to quickly train feedforward networks for the purpose of classification, with numerous applications in speech recognition (Dahl et al., 2010, 2012), acoustic modeling (Mohamed et al., 2012) or images (Ranzato et al., 2011). Thanks to the increase in the amount of data available and the improvement of the hardware, this unsupervised pretraining of feedforward network has been gradually abandoned and is no longer used today.

1.4.2 RBM as generative models

RBM, and its many variants, as Convolutional RBM (Desjardins and Bengio, 2008), Recurrent Neural Network RBM (Boulanger-Lewandowski et al., 2012) or GAN RBM (Fisher et al., 2018) can be used to generate new data once trained, with many applications, such as in image denoising (Tang et al., 2012), generation of textures (Courville et al., 2011; Kivinen and Williams, 2012), proteins with putative properties (Tubiana et al., 2019a,b) and even seasonal cooking recipes (Deudon, 2020).

1.4.3 RBM as features extractors and classifiers

RBM are also used as feature extractors: the M columns of the weight matrix $\mathbf{W}$ can be seen as M features extracted from the training data. In a certain regime, called the compositional phase, these features can be informative and easily interpreted. This is for example the case when the RBM learns stokes of digits on MNIST (Tubiana and Monasson, 2017) or group of coevolving amino-acids in proteins or antigens (Tubiana et al., 2019a,b; Bravi et al., 2021a,b).

In many cases, representations of data in the hidden space of RBM have been proven to be more interpretable and useful for classification than the raw data in the visible space, for example in character recognition (Larochelle and Bengio, 2008; Coates et al., 2011), or collaborative filtering (Netflix problem) (Salakhutdinov et al., 2007).

1.5. Datasets

We use different datasets to illustrate our theoretical results in the thesis. For all datasets, we train RBM using the learning algorithm of Tubiana and Monasson (2017), available from https://github.com/jertubiana/PGM.

1.5.1 Bars and Stripes

Bars and Stripes (BAS) dataset (MacKay, 2003) is made of $L \times L$ binary synthetic images which contain either exclusively bars or exclusively stripes. There are $2^{L+1} - 1$ possible configurations (Fig. 1.4(a)).

1.5.2 MNIST

MNIST dataset (LeCun, 1998) is a large dataset of 28×28 pixel images of handwritten digits. We limit ourselves to zeros and ones (Fig. 1.4(b)), two graphically far digits. We use the binarized version of MNIST: each pixel is either white or black.

1.5.3 Lattice Protein

Lattice Proteins (LP) are artificial proteins used to investigate protein design (Shakhnovich and Gutin, 1990; Shakhnovich et al., 1991; Mirny and Shakhnovich, 2001) and benchmarking inverse modeling procedures (Jacquin et al., 2016). Proteins are sequences of amino acids, whose 3D structures encode their functionalities. In this model, a structure is defined as a self-avoiding path of 27 amino-acid-long chains ($\mathbf{v}$ represents a sequence) on the $3 \times 3 \times 3$ lattice cube. There are $\mathcal{N} = 103,406$ distinct structures (up to global symmetry). The probability that a protein sequence $\mathbf{v}$ folds in a given structure S is given by

$$P_{\text{nat}}(\mathbf{v}|S) = \frac{\exp\left(-E(\mathbf{v},S)\right)}{\sum\limits_{S'} \exp\left(-E(\mathbf{v},S')\right)}, \tag{1.23}$$

where the energy of the sequence $\mathbf{v}$ in a structure S is defined through

$$E(\mathbf{v},S) = \sum_{i<j} c^S_{i,j}\, E_{MJ}(v_i, v_j). \tag{1.24}$$

In the previous formula, $c^S_{i,j} = 1$ if the sites i and j are in contact (neighbors on the cube) in structure S; there are 28 contacts between the amino acids for each structure[3]. Otherwise, $c^S_{i,j} = 0$. The pairwise energy $E_{MJ}(v_i, v_j)$ represents the physico-chemical interactions between the amino acids, given by the Miyazawa-Jernigan (MJ) potential (Miyazawa and Jernigan, 1996). Here, we focus on two structures, S_A and S_B, which define two protein families (Fig. 1.4(c)). For each structure, we sample $\sim 10^4$ sequences that have a high probability to fold in this structure ($P_{\text{nat}}(\mathbf{v}|S) > 0.99$) to build our datasets (Jacquin et al., 2016).

[3] Contacts along the chain are discarded, as their contribution to the energy is structure independent and, hence, does not affect the value of P_{nat}.

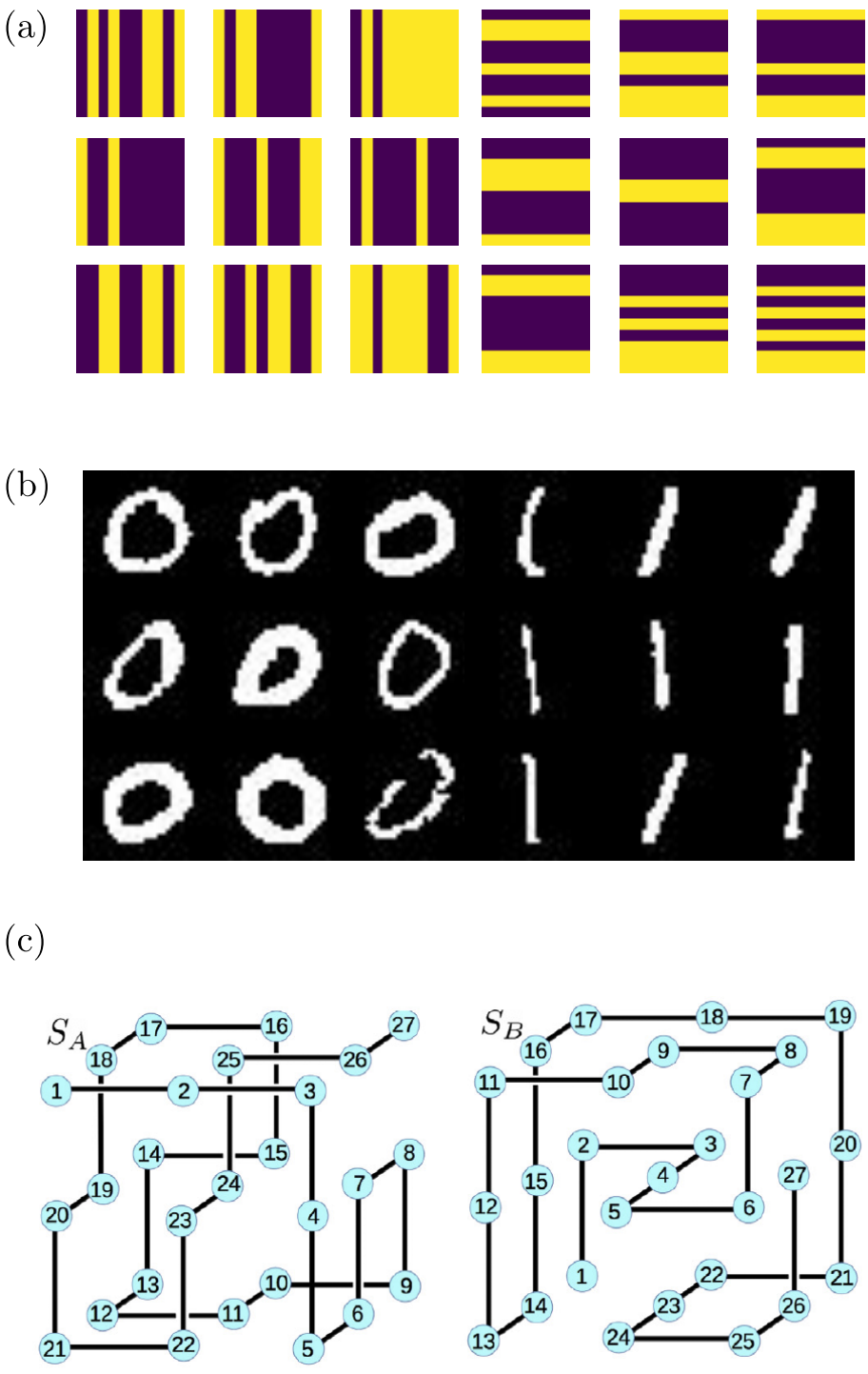


Figure 1.4: (a) BAS: examples of bars (left) and stripes (right) for $L = 10$. (b) MNIST: examples of handwritten 0 and 1 digits. (c) Lattice Proteins: two structures S_A and S_B defining two families of sequences having large P_{nat} with either fold, see Eq. (1.23). Structures from Jacquin et al. (2016).

2

Storing patterns and statistical mechanics of Restricted Boltzmann Machines

The first four sections of this chapter focus on how to store patterns with Restricted Boltzmann Machines. The last section focuses on the different properties of RBM deducted thanks to statistical mechanics. This chapter aims to introduce the concepts that will be useful in Chapters 3 and 4, which will present our results on sampling an energy landscape with RBM.

In more detail, the first section outlines famous results on autoassociative memories and how these results can be related to RBM. The second section focuses on hetero-associative memories and their link with RBM. The third section, on the universal approximation theorem of RBM. And the fourth section compares the different solutions proposed in the previous parts.

2.1. Autoassociative memory: Little and Hopfield models

This section presents the ideas and results of work conducted by Little (1974); Little and Shaw (1975) and Hopfield (1982) to build artificial neural networks capable of storing patterns. These networks are recurrent artificial neural networks, and we will refer to them as autoassociative memory. We will also show how these two models can be represented by RBM, and thus how RBM can also be used to store patterns.

2.1.1 From neural networks to statistical physics

In both cases, their model is composed of N neurons, all connected to each other by synapses (we speak in this case of a fully connected model). McCulloch Pitts neurons (McCulloch and Pitts, 1943) can take two values ($v_i = \pm 1$), where $v_i = 1$ corresponds to an active neuron and $v_i = -1$ to a silent one. A neuron v_i is connected to a neuron v_j by a synaptic connectivity of intensity J_{ij}. This synaptic connectivity is symmetric ($J_{ij} = J_{ji}$). Although this hypothesis may be questionable from a biological point of view, it has the advantage of linking these artificial neural networks to the statistical physics of disordered systems. This link allows to describe these neural networks by an energy, and to use the wide range of tools of statistical physics[1].

Note that this assumption has also been made previously in the case of RBM, where the intensity of the relations between a hidden unit h_μ and a visible unit v_i is characterized

[1] as explained in Chapter 7 of "Modeling Brain Function" by Amit (1989), the results derived below are fairly robust if the symmetry constraint is relaxed. The reader is invited to look at the works of Parisi (1986); Brunetti et al. (1992a,b) in the case of asymmetric J_{ij} or Shinomoto (1987) in the case where J_{ij} satisfies Dale's law (Eccles, 1964), *i.e.*, that a neuron i is only excitatory or inhibitory ($J_{ij} > 0$ or $J_{ij} < 0$ for all j).

by a single quantity $W_{i\mu}$.

In the case of the Hopfield model, the energy reads

$$E^{\text{Hop}}(\mathbf{v}) = -\frac{\beta}{2}\sum_{i,j} J_{ij} v_i v_j, \tag{2.1}$$

where $\beta = \frac{1}{T}$ is the inverse temperature of the model. And for the Little model, as described in Peretto (1984)

$$E^{\text{Little}}(\mathbf{v}) = -\sum_{i=1}^{N} \log 2\cosh\left(\beta \sum_{j=1}^{N} J_{ij} v_j\right). \tag{2.2}$$

As we will see in Section 2.1.5, these two models can be represented using a RBM.

The K patterns to store $\boldsymbol{\xi}^k$ $(k = 1...K)$ are random and chosen according to

$$P(\xi_i^k) = \frac{1}{2}\delta_{\xi_i^k,1} + \frac{1}{2}\delta_{\xi_i^k,-1}. \tag{2.3}$$

Therefore, in the thermodynamic limit $N \to \infty$, these patterns are on average orthogonal two by two

$$\left\langle \boldsymbol{\xi}^{k^T} \cdot \boldsymbol{\xi}^{k'} \right\rangle = N\delta_{k,k'}. \tag{2.4}$$

This property is only true on average, and as we will in Section 2.1.4, the fact that the patterns are not exactly orthogonal has a crucial importance when the number of patterns K is of the same order of magnitude as the number N of neurons in the network.

The idea is to find some parametrization of synaptic connectivity J_{ij} so that neural networks can retrieval these K patterns. In the noiseless limit $\beta \to \infty$, by retrieval, we mean that the K (finite) patterns are global minima of the previously defined energies. It also corresponds to fixed points of the following dynamics, where the probability of finding v_i in a given state is given by

$$P(v_i) = \frac{1}{2}\left(1 + \tanh\left(\frac{\beta}{2} v_i \sum_{j\neq i} J_{ij} v_j\right)\right) \underset{\beta\to\infty}{\longrightarrow} \begin{cases} P(v_i = 1) & \text{if } \sum\limits_{j\neq i} J_{ij} v_j > 0 \\ P(v_i = 1) = P(v_i = -1) = \frac{1}{2} & \text{if } \sum\limits_{j\neq i} J_{ij} v_j = 0 \\ P(v_i = -1) & \text{otherwise.} \end{cases} \tag{2.5}$$

Consequently, in the noiseless limit, a spin v_i is stable if and only if its magnetic field due to the $N-1$ other spins, $\sum\limits_{j\neq i} J_{ij} v_j$, is aligned with it (*i.e.*, $v_i \sum\limits_{j\neq i} J_{ij} v_j > 0$). Originally, the dynamics of the Hopfield model was designed only in this limit $\beta \to \infty$, and asynchronously: each v_i is updated one by one. For the Little model, the dynamics is synchronous: all spins are updated at the same time. Each update according to this dynamics decreases the total energy of the system. These different dynamics could cause major differences between the two models, but as it was shown in the case of a finite number of patterns, these two models have the same thermodynamic properties (Amit et al., 1985a).

In the noisy case, or when $\frac{K}{N}$ is finite, the patterns can not be perfectly retrieved due to the fluctuations. However, it is still possible to define the retrieval m_k using overlaps with the memorized pattern $\boldsymbol{\xi}^k$ as

$$m_k = \frac{1}{N}\sum_{i=1}^{N} \langle v_i \rangle \xi_i^k. \tag{2.6}$$

2.1.2 Choice of the synaptic connectivity

Donald Hebb proposed in 1949 a learning rule based on synaptic plasticity (Hebb, 1949). This rule suggests that when two neurons are jointly excited or silent, a link between them is created or reinforced. This rule is often summarized as follows: "cells that fire together, wire together", and is known as Hebb's rule. From a more formal point of view, this rule can be translated as follows

$$\begin{aligned} J_{ij} &= \frac{1}{N}\sum_k \xi_i^k \xi_j^k, \\ J_{ii} &= 0. \end{aligned} \tag{2.7}$$

For each memory, this indeed corresponds to the rule recommended by Hebb: $J_{ij} = \xi_i^k \xi_j^k > 0$, *i.e.*, the connectivity is reinforced, if and only if $\xi_i^k = \xi_j^k$, *i.e.*, if the two neurons are jointly excited or silent. This rule is summed over the K patterns, and the factor N^{-1} is here to get intensive quantities in the large N limit. This rule has several advantages. It is local, *i.e.*, the connection between two spins v_i and v_j depends only on the patterns at sites i and j. It is additive, so it is easy to add progressively new patterns over time. Nevertheless, as we will see in Section 2.1.4.2, this rule is not optimal in the sense that it does not allow to store a maximum number of patterns.

With Hebb's rule, Eq. (2.1) reads

$$E^{\text{Hop}}(\mathbf{v}) = -\frac{\beta}{2N}\sum_{\substack{i,j=1\\ i\neq j}}^{N}\left(\sum_{k=1}^{K}\xi_i^k\xi_j^k\right)v_i v_j, \tag{2.8}$$

and Eq. (2.2) reads

$$E^{\text{Little}}(\mathbf{v}) = -\sum_{i=1}^{N}\log 2\cosh\left(\frac{\beta}{N}\sum_{\substack{j=1\\ j\neq i}}^{N}\left(\sum_{k=1}^{K}\xi_i^k\xi_j^k\right)v_j\right). \tag{2.9}$$

2.1.3 Finite number of patterns

In this section, we will focus on the case where the number of patterns is finite and therefore negligible versus the number of neurons ($\frac{K}{N} \underset{N\to\infty}{\to} 0$). This regime is often called low memory loading. This regime is of particular interest to us because we will find ourselves in a similar regime in the various results we will present later for RBM in Chapter 4.

2.1.3.1 A quick look at the Curie-Weiss and Mattis models

Before looking at the general results, we will start with the straightforward case where $K = 1$. In this case, by injecting Hebb's rule (Eq. (2.7)) into the energy of the Hopfield model (Eq. (2.1)), and choosing $\xi_i^k = 1$ for all i, up to a irrelevant additive constant[2], we get:

$$E^{\text{CW}}(\mathbf{v}) = -\frac{\beta}{2N}\sum_{i,j=1}^{N} v_i v_j. \tag{2.10}$$

[2]Here, $J_{ii} = \frac{1}{N}$, contrary to the Hebb's rule (Eq. 2.7). However, this is irrelevant as $E^{\text{CW}}(\mathbf{v}) = -\frac{\beta}{2N}\sum_{i,j=1}^{N} v_i v_j = \frac{\beta}{2N}\sum_{\substack{i,j=1\\ i\neq j}}^{N} v_i v_j - \frac{\beta}{2}$.

This model is called the Curie-Weiss model and is a mean-field version of the Ising model (Ising, 1925) where $J_{ij} = \frac{1}{N}$ (see Ellis (1985) for details).

For $T < 1$ and infinite-size limit $N \to \infty$, the average magnetization of the spins, $m = \frac{1}{N}\sum_{i=1}^{N} v_i$, spontaneously acquires a nonzero value. The value of this order parameter is determined by minimizing the free energy (per spin), $f(m) = -\frac{w^2}{2}m^2 - \mathcal{S}(m)$, where

$$\mathcal{S}(m) = -\sum_{\sigma=\pm 1} \frac{1+\sigma m}{2} \log\left(\frac{1+\sigma m}{2}\right), \tag{2.11}$$

is the entropy at fixed magnetization. The free energy $f(m)$ is an even function of m, with a double-well shape. The two opposite values of the spontaneous magnetization, roots of $f'(m^*) = 0$, define two collective states of the system. Notice that $m = 0$ is a local maximum of the free energy. The dynamics defined in Eq.(2.5) converges to m^* or $-m^*$: the system retrieves the memory.

The Curie-Weiss model can be modified to store one specific pattern $\boldsymbol{\xi}^1$ of $\{-1,1\}^N$ by defining $J_{ij} = \frac{1}{N}\xi_i^1\xi_j^1$ [3]. This model is called Mattis model (Mattis, 1976). With this parametrization, the global minima of the free energy are reached for configurations with $\pm m^*$ magnetization along $\boldsymbol{\xi}^1$: the system retrieves the memory $\boldsymbol{\xi}^1$.

2.1.3.2 Results for $K > 1$

The main results presented here are from Amit et al. (1985a).

For the infinite-size limit, $N \to \infty$, the energy of the Hopfield model (Eq. (2.1)) with Hebb's rule reads

$$E^{\text{Hop}}(\mathbf{m}) = \beta\sum_k \frac{m_k^2}{2} - \frac{1}{N}\sum_i \log 2\cosh\left(\beta\sum_k \xi_i^k m_k\right), \tag{2.12}$$

where m_k are the magnetization along the patterns defined in Eq. (2.6). For $T > 1$, there is a single global minimum for $m_k = 0$. The patterns can not be retrieved. In the case where $T < 1$, the previous energy has $2K$ global minima, which are the Mattis states for the K patterns (Procesi and Tirozzi, 1990). This result is also valid in the case of Little's model. Consequently, the patterns can be retrieved, and the Hopfield and the Little models work as autoassociate memory. For the global minima, the vector of magnetization $\mathbf{m}$ has only one non-zero component.

The energy landscape is much richer than in the case of the Curie Weiss model or the Mattis model: the landscape has a large variety of saddle points or local minima whose nature depends on β. These landscape properties will be beneficial in Chapter 4 to compute the characteristic times associated with Alternating Gibbs Sampling of RBM.

An interesting class of these critical points are the symmetric spurious patterns, which can be written $\mathbf{m} = m_r(\underbrace{1,1,\dots 1}_{r},\underbrace{0,0,\dots,0}_{K-r})$. There is complete symmetry of the solutions under the permutations of the components of $\boldsymbol{m}$ as well as under the change of sign of any of them. There are 3^K such states of being compared with the $2K$ Mattis states. Therefore, for large $K \ll N$, these symmetric spurious patterns outnumber the Mattis states.

In the noiseless limit $T = 0$, it can be shown that the symmetric spurious patterns with an odd number of components are local minima of the energy landscape: therefore, they are metastable states, and the dynamics can be trapped in one of these states. As long as

[3] By defining $m = \frac{1}{N}\sum_{i=1}^{N} v_i\xi_i^1$, the free energy of the Curie-Weiss model is indeed retrieved.

$0.461 < T < 1$, all the states are saddle points. As pointed out by Daniel J. Amit in his well-known book "Modeling Brain Function: The World of Attractor Neural Networks" (Amit, 1989), the noise has here a positive role, as it eliminates the spurious states, contrary to the noiseless limit. Nevertheless, the intensity of the Mattis states also depends on the temperature. The higher the β, the higher the Mattis magnetization, and thus the more accurately the memory is retrieved.

There are also asymmetric spurious patterns for $T < 0.57$ but none of them are stable as long as long as $T > 0.461$.

2.1.4 Infinite number of patterns

In this section, we will focus on the case where the number of patterns is non-negligible compared to the number of neurons ($\frac{K}{N} \underset{N\to\infty}{\to} \alpha > 0$, where α is the load.) This regime is different from the previous one because from now on, it is not possible to neglect that the patterns are orthogonal two by two only on average. With Hebb's rule, even in the noiseless limit, patterns $\boldsymbol{\xi}$ may be unstable with the dynamics defined in Eq. (2.5). Indeed, the local field received by ξ_1^1 when $\mathbf{v} = \boldsymbol{\xi}^1$ reads

$$\frac{1}{N} \sum_{j \neq i} \sum_{k} \xi_i^k \xi_j^k \xi_j^1, \tag{2.13}$$

and the stability condition reads

$$\xi_1^1 \frac{1}{N} \sum_{j \neq i} \sum_{\mu} \xi_i^k \xi_j^k \xi_j^1 = \underbrace{\frac{N-1}{N}}_{\text{signal}} + \underbrace{\frac{1}{N} \sum_{j \neq i} \sum_{k>1} \xi_1^1 \xi_i^k \xi_j^k \xi_j^1}_{\text{noise}} > 0 \tag{2.14}$$

The second term can be identified as a noise term due to the non-orthogonality between the patterns. Its intensity is of order $\frac{\sqrt{NK}}{N} \underset{N\to\infty}{\to} \alpha$, compared to intensity of order 1 for the signal. Therefore, in the regime with $\alpha > 0$, the patterns may not be retrieved.

2.1.4.1 Phase diagram

As derived in Amit et al. (1985b), three distinct phases exist depending on temperature and load (Fig. 2.1(a)):

- the paramagnetic phase: at high temperature, the noise dominates and $\langle v_i \rangle = 0$. No patterns could be retrieved.
- the spin-glass phase: at low temperature and high load, neurons have a non-zero polarization $\langle v_i \rangle \neq 0$, but not aligned with a specific memory: they have weak overlap with all patterns, and therefore patterns can not be retrieved. This phase is similar to the one described in Kirkpatrick and Sherrington (1978).
- the ferromagnetic phase: at low temperature and high load, neurons have a non-zero polarization with on the memory $\boldsymbol{\xi}^k$. The ferromagnetic phase is cut in two: at very low temperatures, retrieval states are global minima of the free energy, and at higher temperatures, there are only local minima.

There exists a critical load $\alpha_c \simeq 0.138$. This means that the Hopfield model with N neurons can store up to $K = 0.138N$ patterns. The case studied in Section 2.1.3 where the number of patterns is finite corresponds to $\alpha = 0$. For $T < 1$, the Mattis states are indeed global minima of the energy. It should be noted that in the paramagnetic phase

with $0 < \alpha < \alpha_c$ and $T = 0$, the behavior is different from the case where the number of patterns is finite ($\alpha = 0$). As shown in Fig. 2.1(b), patterns are not perfectly retrieved: there is some noise due to the other patterns, which act as an effective temperature (Amit, 1989).

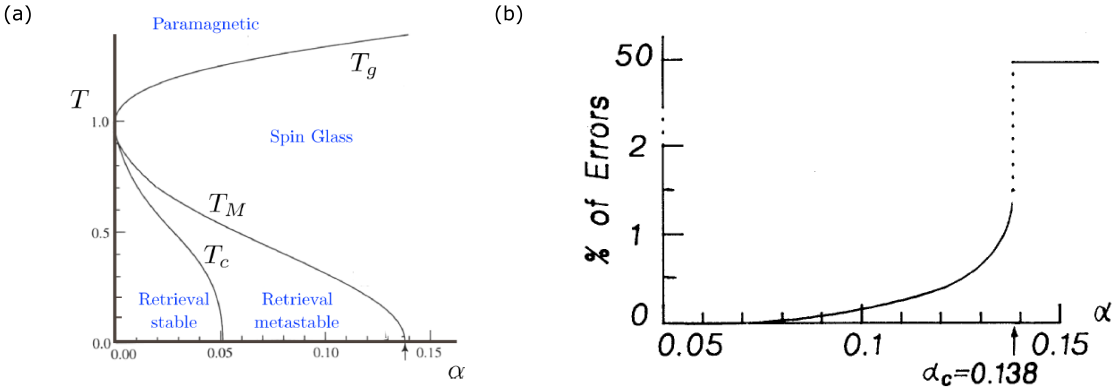


Figure 2.1: (a) Phase diagram of the Hopfield model derived in Amit et al. (1985b). Figure from Mézard (2017). (b) Average percentage of errors in the ferromagnetic phase for $\beta \to \infty$. Figure from Amit et al. (1985b).

2.1.4.2 Improvement of the capacity

Other learning rules of the synaptic weights have been used to improve the performance of autoassociative memories, such as the pseudo inverse rule (Personnaz et al., 1986; Kanter and Sompolinsky, 1987) which allows storing correlated patterns. Unlike Hebb's rule, the pseudo inverse rule is not a local, as the synaptic couplings depend on the inverse of the correlation matrix of the patterns. Elizabeth Gardner has shown that the optimal rule achieves a load $\alpha_c = 2$ for uncorrelated patterns (Gardner, 1987, 1988; Gardner and Derrida, 1988). Nevertheless, for this optimal storage, there is no explicit formula for synaptic couplings. However, it is possible to find this solution numerically, thanks to an extension of the perceptron learning rule (Rosenblatt, 1958; Gardner, 1988).

2.1.5 Links between Little and Hopfield model and Restricted Boltzmann Machines

The Hopfield model can be represented with a Spin-Gaussian RBM with N visible units $v_i = \pm 1$ (with potentials $\mathcal{V}_i = 0$) and $M = K$ continuous hidden units subject to the quadratic potential $\mathcal{U}(h) = \frac{h^2}{2}$ (Barra et al., 2012; Agliari et al., 2012; Leonelli et al., 2021). The energy of the RBM in Eq. (1.2) reads

$$E^{\text{Hop}}(\mathbf{v}, \mathbf{h}) = -\sum_{i,\mu} W_{i\mu} v_i h_\mu + \sum_\mu \frac{h_\mu^2}{2}. \tag{2.15}$$

It is straightforward to check, after integration over the M hidden units, that the effective energy in Eq. (1.4) coincides with the Hopfield energy in Eq. (2.8) provided the weights fulfill the constraints

$$\sum_\mu W_{i\mu} W_{j\mu} = \frac{w^2}{N} \sum_\mu \xi_i^\mu \xi_j^\mu. \tag{2.16}$$

These conditions do not uniquely define the weight matrix $\mathbf{W}$. The energy is invariant under any transformation $\mathbf{W} \to \mathbf{W} \times \mathbf{O}$, where $\mathbf{O}$ is an orthogonal matrix. It is also

interesting to note, that in the $M \ll N$ regime, the $E^{\text{eff}}(\mathbf{v})$ landscape of a Spin-Spin RBM is equivalent to the Hopfield model, provided that $W_{i\mu} = \xi_i^\mu$ (Agliari et al., 2012).

The Little model can be represented with a Spin-Spin RBM with N visible units $v_i = \pm 1$ (with potentials $\mathcal{V}_i = 0$) and $M = N$ hidden units $h_\mu = \pm 1$ ($\mathcal{U}_\mu(h) = 0$). The energy of the RBM in Eq. (1.2) reads

$$E^{\text{Little}}(\mathbf{v}, \mathbf{h}) = -\sum_{i,\mu} W_{i\mu} v_i h_\mu. \tag{2.17}$$

It is straightforward to check, after integration over the M hidden units, that the effective energy in Eq. (1.4) coincides with the Little energy in Eq. (2.9) provided $W_{i\mu} = W_{ij} = \beta J_{ij}$, where J_{ij} is defined by Hebb's rule (Eq. (2.7)).

Therefore, RBM can be used to store patterns, similar to Hopfield's model and Little's model. Nevertheless, a major difference exists between RBM and these two models. Indeed, Hopfield's and Little's models are autoassociative memories and thus have only N neurons v_i, which allow storing the patterns. RBM are richer, and have in addition to its N neurons v_i M neurons h_μ. Therefore, each memory $\boldsymbol{\xi}^\mu$ has a representation in the space of hidden units, as seen in Section. 1.2.3. Concerning the Hopfield model, as there is an invariance for the weight matrix $\mathbf{W}$ (Eq. (2.16)), the hidden representation depends on the parametrization. As we will show in Chapter 3, this choice does not influence the sampling on the landscape $E^{\text{Hop}}(\mathbf{v})$ with a RBM with Alternating Gibbs Sampling. Nevertheless, to improve the sampling and go beyond AGS, a simple representation of the Mattis states in the space of hidden units will play a crucial role. This representation is achieved for

$$W_{i\mu} = \frac{w}{\sqrt{N}} \xi_i^\mu. \tag{2.18}$$

In that particular case, a hidden unit h_μ is strongly activated only when $\mathbf{v}$ is aligned with $\boldsymbol{\xi}^\mu$.

Concerning the Little model, in the limit $\beta \to \infty$, the hidden representation of $\mathbf{v} = \boldsymbol{\xi}^\mu$ is $\mathbf{h} = \boldsymbol{\xi}^\mu$. The representation in the hidden space of the memory is identical to the memory itself. RBM store the memory twice: once in its visible space, another time in its hidden space. We will see in the next section how to go further than this model, using heteroassociative memory, another kind of associative memory.

2.2. Bidirectional Associative Memory

2.2.1 Description

Bidirectional Associate Memory (BAM) is a recurrent neural network introduced by Bart Kosko (Kosko, 1987, 1988). Like autoassociative memories, his idea is to build a recurrent network capable of storing patterns. However, unlike autoassociative memories which use a single population of N neurons, BAM uses two distinct populations of N and M neurons respectively, and is used to store K memory pairs $\{\boldsymbol{\xi}^k, \hat{\boldsymbol{\xi}}^k\}_{k=1\dots K}$, where $\boldsymbol{\xi}^k \in \{-1,1\}^N$ and $\hat{\boldsymbol{\xi}}^k \in \{-1,1\}^M$. Let us call v_i the neurons of the first population and h_μ the neurons of the second population. The two populations are connected through a weight matrix $\mathbf{W}$ of size $N \times M$. There are no couplings within a population. BAM dynamics is

$$\begin{aligned} \mathbf{h} &= \text{sign}\left(\mathbf{W}^T \cdot \mathbf{v}\right), \\ \mathbf{v} &= \text{sign}\left(\mathbf{W} \cdot \mathbf{h}\right). \end{aligned} \tag{2.19}$$

The matrix $\mathbf{W}$ must be tailored such that the pairs of patterns $\{\boldsymbol{\xi}^k, \hat{\boldsymbol{\xi}}^k\}$ are fixed points of the previous equations. If the patterns are random and drawn from Eq. (2.3), one possible choice is to use Hebb's rule

$$W_{i\mu} = \frac{1}{N}\sum_{k=1}^{K} \xi_i^k \hat{\xi}_\mu^k. \tag{2.20}$$

If $N = M$, and $\boldsymbol{\xi}^k = \hat{\boldsymbol{\xi}}^k$, BAM is like an autoassociative memory at $T = 0$ with Hebb's rule, where both populations of neurons play the same role. Therefore, the patterns are stored in both populations, as when we use a RBM to represent Little's model. Just like the Hopfield model, the optimal load $\alpha = \frac{K}{N}$ can be computed. In the particular case of $N = M$, $T = 0$, (Tanaka et al., 2000) have that there is a critical load $\alpha_c = 0.1998$. This critical load is slightly larger than the optimal load of the Hopfield model ($\alpha_c = 0.138$).

2.2.2 Link to Restricted Boltzmann Machines

The equations of the BAM dynamics described in Eq. (2.19) are closely related to the equations of the dynamics of a Spin-Spin RBM in the limit of large couplings (with $\mathcal{V}_i = \mathcal{U}_\mu = 0$). Indeed, if we note as $\langle \mathbf{h}|\mathbf{v}\rangle$ the average value of the unit $\mathbf{h}$ which receives its input from $\mathbf{v}$ and $\langle \mathbf{v}|\mathbf{h}\rangle$ the average value of the unit $\mathbf{v}$ which receives its input from $\mathbf{h}$

$$\langle \mathbf{h}|\mathbf{v}\rangle = \tanh\left(\mathbf{W}^T \cdot \mathbf{v}\right), \tag{2.21}$$

$$\langle \mathbf{v}|\mathbf{h}\rangle = \tanh\left(\mathbf{W} \cdot \mathbf{h}\right). \tag{2.22}$$

Therefore, if the inputs received are large, as $\tanh(I) \underset{I\to\infty}{\sim} \text{sign}(I)$, the equations of the BAM dynamics described in Eq. (2.19) are retrieved.

RBM can therefore be used to store memory pairs $\{\boldsymbol{\xi}^k, \hat{\boldsymbol{\xi}}^k\}$, as a BAM. From this observation, a natural question appears. When training RBM, we can see the data as patterns $\boldsymbol{\xi}^k$. Contrary to BAM, the patterns $\hat{\boldsymbol{\xi}}^k$ are not fixed. How are they chosen? This question will be partially addressed, based on numerical experiments and theoretical computations, in Chapter 4.

2.3. Bernoulli-Bernoulli Restricted Boltzmann Machines are universal approximators

2.3.1 Construction of the solution: a geometric interpretation

As derived in Le Roux and Bengio (2008), Bernoulli-Bernoulli RBM are universal approximators. It means that a RBM can learn any distribution on the hypercube $\{0,1\}^N$ and therefore Bernoulli-Bernoulli RBM can be used to store patterns. To understand how works their proof, let's consider a simple example where the support of the distribution is K vectors $\{\boldsymbol{\xi}^k\}_{k=1..K}$, with a probability distribution $P(\mathbf{v}) = \frac{1}{K}\sum_{k=1}^{K}\prod_{i=1}^{N}\delta_{v_i,\xi_i^k}$. For a Bernoulli-Bernoulli RBM, the energy $E^{\text{eff}}(\mathbf{v})$ (Eq. (1.4)) reads

$$E^{\text{eff}}(\mathbf{v}) = -\sum_{i=1}^{N} g_i v_i - \sum_{\mu=1}^{M} \log\left(1 + \exp\left(\sum_{i=1}^{N} c_\mu + W_{i\mu} v_i\right)\right). \tag{2.23}$$

If the sum of the input received by the hidden unit h_μ, $I_\mu(\mathbf{v}) = \sum_{i=1}^{N} W_{i\mu} v_i$ and its field c_μ is large enough ($|I_\mu(\mathbf{v}) + c_\mu| \gg 1$), the energy can be written as

$$E^{\text{eff}}(\mathbf{v}) \simeq -\sum_{i=1}^{N} g_i v_i - \sum_{\mu=1}^{M} \max\left(0, I_\mu(\mathbf{v}) + c_\mu\right). \tag{2.24}$$

We can interpret terms in the sum over the hidden units in a geometrical way. Let us define the hyperplane $\mathcal{H}_\mu = \{\mathbf{x} \in \mathbb{R}^N, c_\mu + \sum_{i=1}^{N} W_{i\mu} x_i = 0\}$, and its normal $\mathbf{n}_\mu$

$$\mathbf{n}_\mu = \begin{pmatrix} w_{1\mu} \\ \vdots \\ w_{N\mu} \end{pmatrix}. \tag{2.25}$$

- If $\mathbf{v}^T \cdot \mathbf{n}_\mu + c_\mu > 0$ (i.e $h_\mu = 1$), then $\max(0, c_\mu + I_\mu(\mathbf{v})) = c_\mu + I_\mu(\mathbf{v}) > 0$ is proportional to the distance of $\mathbf{v}$ to the hyperplane $\mathcal{H}_\mu$.
- If $\mathbf{v}^T \cdot \mathbf{n}_\mu + c_\mu < 0$ (i.e $h_\mu = 0$), $\max(0, c_\mu + I_\mu(\mathbf{v})) = 0$.

The idea of Bengio is to choose the hyperplane $\mathcal{H}_\mu$ such that $\mathbf{v}^T \cdot \mathbf{n}_\mu + c_\mu > 0$ if and only if $\mathbf{v} = \boldsymbol{\xi}^\mu$ (the μ^{th} pattern to store). Here, fields on the visible layer are unnecessary ($\forall i, g_i = 0$). In that case, the energy reads

$$E^{\text{eff}}(\mathbf{v}) \quad = \quad -\sum_{\mu=1}^{M} (I_\mu(\mathbf{v}) + c_\mu) \prod_{i=1}^{N} \delta_{v_i, \xi_i^k}, \tag{2.26}$$

This proof can be easily adapted in the case of Spin-Spin RBM. This idea is still to cut the hypercube $\{-1,1\}^N$ with K hyperplanes, and have $h_\mu = 1$ if and only if $\mathbf{v} = \boldsymbol{\xi}^\mu$. However, visible fields g_i are needed in order to compensate the effects of $h_\mu = -1$, see Appendix A.

2.3.2 Representation and sampling

From the construction of this solution, we can draw two important conclusions:

- Landscape $E^{\text{eff}}(\mathbf{v})$ is flat with K holes corresponding to $\{\boldsymbol{\xi}^k\}_{k=1..K}$.
- $h_\mu = 1$ if and only if $\mathbf{v} = \boldsymbol{\xi}^\mu$.

In this solution, each hidden unit h_μ has a precise role: detecting a single vector $\mathbf{v}^\mu$. Each hidden unit h_μ is therefore a "grandmother cell" (Barlow, 1972; Churchland, 1986; Gross, 2002; Bowers, 2011), *i.e.*, a cell which has a particular role[4]. Although this solution perfectly stores the patterns, it's inefficient in terms of patterns retrieval. In fact, the basins of attractions of $\boldsymbol{\xi}^\mu$ is limited to itself (Fig. 2.2(a)). With Alternating Gibbs Sampling, starting with $\boldsymbol{\xi}^\mu$ triggers only one hidden unit h_μ. Sampling back the visible layer leads to the same pattern $\boldsymbol{\xi}^\mu$. However, starting with a random vector $\mathbf{v}$ of $\{-1,1\}^N$, none of the hidden units will be triggered except if $\mathbf{v}$ is equal to one of $\{\boldsymbol{\xi}^k\}_{k=1..K}$. In the case where none of the hidden units are triggered, sampling back the visible layer leads to another random vector $\mathbf{v}$ of $\{-1,1\}^N$. Therefore, with this solution patterns retrieval is very unlikely: at each time step of the Alternating Gibbs Sampling, a random $\mathbf{v}$ of the hypercube is drawn uniformly, up to reach one of the $\{\boldsymbol{\xi}^k\}_{k=1..K}$. $\mathcal{O}(K^{-1}2^N)$ steps are needed to retrieve a pattern. Furthermore, once a pattern has been retrieved, the dynamics is stuck and can not efficiently find the other ones.

We adapted the proof in order to have finite-size basins of attraction (Appendix A.1.1.1). The basic idea is to have a hidden unit h_μ which triggers for all patterns $\mathbf{v}$ in a neighborhood

[4]Grandmother cells are hypothetical neurons are very specific, and they activate only for a very specific input, such as when you see your grandmother in a photo.

of $\boldsymbol{\xi}^\mu$ ($\forall \mathbf{v}$ such that $\text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) < dN$, where dN denotes a distance). This solution has a geometric interpretation. Each hyperplane $\mathcal{H}_\mu$ still separate the hypercube in two, but instead of having one vector (the pattern) on one side of the hyperplane and $2^N - 1$ on the other side, now there are $2^{N\mathcal{S}(d)}$ vectors on one side and $2^N - 2^{N\mathcal{S}(d)}$ vectors on the other side. $\mathcal{S}$ denotes the binary entropy. For orthogonal patterns, $d \sim \frac{N}{4}$. The energy landscape $E^{\text{eff}}(\mathbf{v})$ is still flat, but now the K holes have some width. Each of the K holes contains $2^{N\mathcal{S}(d)}$ vectors (Fig. 2.2(b)). Nonetheless, despite the number of points in the basins of attractions are exponential in the dimension N, in the thermodynamic limit, $2^{N(K\mathcal{S}(d)-1)}$ goes to 0.

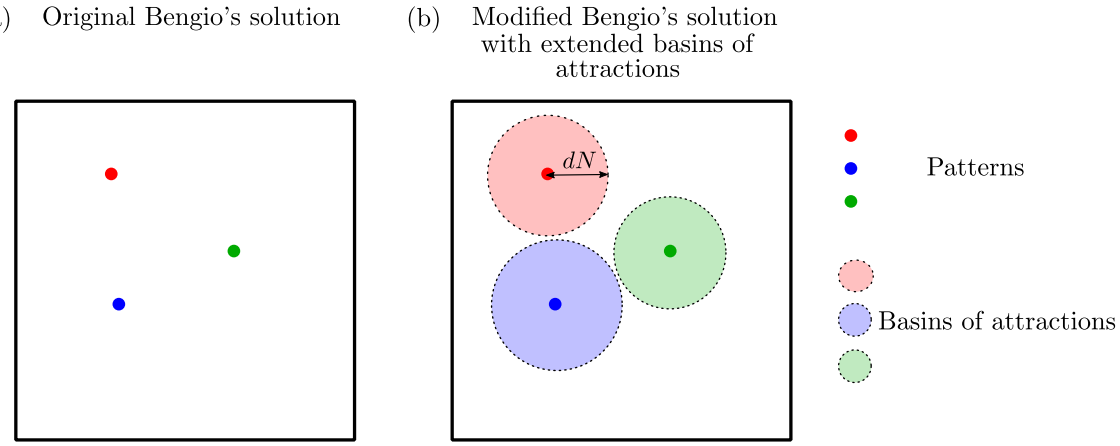


Figure 2.2: Square boxes represent the space of configuration. (a) Original Bengio's solution: basins of attractions of the patterns are reduced to the patterns only. (b) Modified Bengio's with extended basins of attractions.

2.4. Representation and sampling

As discussed in the previous sections, there are different ways to store patterns using RBM. All these solutions are not equivalent. The solution proposed by Le Roux and Bengio (2008) allows storing exactly the $\boldsymbol{\xi}^k$ patterns, without any spurious memory. Nevertheless, this solution suffers from several defects. Starting from a random configuration $\mathbf{v}$, AGS does not allow reaching quickly one of the patterns. Moreover, this solution is theoretical, and it is impossible to reach it dynamically by training a RBM by maximizing the log-likelihood on the patterns.

The comparison between RBM and associative memories also has its advantages and disadvantages. As we will show numerically in Chapters 3 and 4, these solutions are indeed found when a RBM is trained by log-likelihood maximization, in the limit $K \ll M, N$. Moreover, starting from a random configuration $\mathbf{v}$, AGS allows converging to one of the patterns. Nevertheless, there are spurious patterns, and the dynamic can get stuck in one of its unwanted states. Moreover, the results on associative memories are well understood when the patterns are random, which is not the case in data in general, where some structure emerges.

Concerning the representations of the patterns in the hidden layer, we will show in Chapter 3 that they do not matter when considering only AGS for a RBM. Nonetheless, when we want to improve the sampling, by adding for example a dynamic in the space of the hidden units (Chapter 3) or by using a stack of RBM coupled with Deep Tempering (Chapter 4), the representations are crucial to improve the sampling performances of the RBM.

2.5. What does statistical mechanics tell us about RBM?

This section discusses the results deduced on RBM using methods from statistical mechanics. These results shed light on the different representations accessible by RBM, as well as on the different phases and their learning capabilities. Decelle and Furtlehner (2020b) recently published a detailed review on the subject.

Tubiana and Monasson (2017) describe the behavior of RBM when the weights $W_{i\mu}$ are drawn randomly, with probability $\frac{p}{2}$ to be equal to 1 or -1, and probability $1-p$ to be equal to 0. The hidden units are continuous with a potential ReLU, and they study the regime where the number of hidden units is of the same order as the number of visible units ($\alpha = \frac{M}{N} = \mathcal{O}(1)$).

In addition to the traditional ferromagnetic and spin-glass phases, they show that a compositional phase exists. In this phase, a number $1 \ll L \ll M$ of hidden units is strongly activated simultaneously, and interpolates between the ferromagnetic phase, where only one hidden unit is strongly activated, and the spin-glass phase where all units are weakly activated (Fig. 2.3). In this compositional phase, data can be represented using L features. As shown in a series of papers, these features are interpretable, and encode for example strokes of digits (Tubiana and Monasson, 2017), or contacts and biological features of proteins (Tubiana et al., 2019a,b; Bravi et al., 2021a,b). Therefore, this regime is suitable for representing data, as it is halfway between a prototypical representation of data (ferromagnetic phase), where representation is trivial, and a completely delocalized representation of data (spin-glass phase), where representation is not interpretable. We will place ourselves in this phase to study the class A β-lactamases family in Chapter 9.

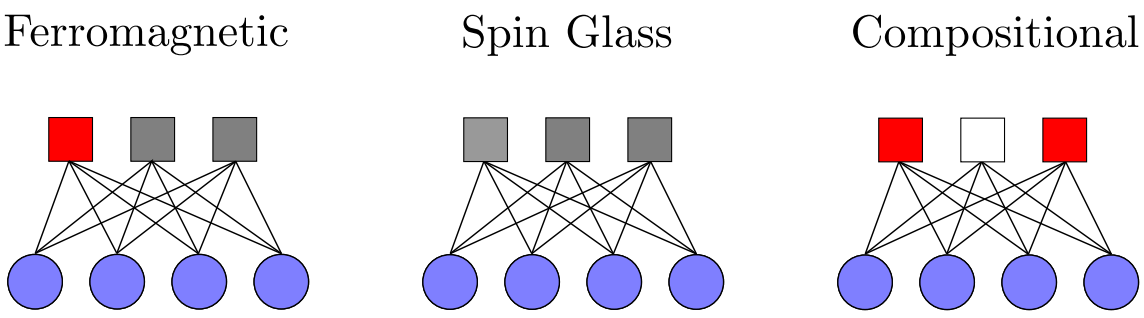


Figure 2.3: The three regimes of operation of Random RBM. Red, gray and white hidden units symbolize, respectively, strong ($h = \mathcal{O}(\sqrt{N})$), weak ($h = \mathcal{O}(1)$) and null ($h = 0$) activations. In the ferromagnetic phase, one hidden unit is strongly activated, and the others are weakly activated. The number of attractors is linear in N. In the spin-glass phase, all hidden units are weakly activated, with many metastable states. In the compositional phase, several hidden units are strongly activated, and the others are quiet. The number of attractors is polynomial in N. Figure and caption adapted from Tubiana (2018).

Agliari et al. (2012) also described the behavior of RBM with random weights (same distribution as Tubiana and Monasson (2017)), but in the case of Spin-Spin RBM, in the regime where $\alpha = 0$. In this regime, $E^{\text{eff}}(\mathbf{v})$ is equivalent to the Hamiltonian of the Hopfield model. In the regime where $p \ll 1$, which corresponds to a strong dilution of the RBM weights $W_{i\mu}$[5], an interesting phase appears. RBM is able to remember several patterns at

[5]It is important to note that even for a strong dilution of the RBM weights, the v_i spins are still connected. This dilution is quite different from the dilution of the Hopfield model, where the connections between the v_i are randomly removed (Sompolinsky, 1986). In this case, it is a Hopfield model on random graph sparse. While for the RBM weight dilution, the spin connection graph is fully connected, but of lower intensities. The properties of the networks are therefore different from each other.

the same time, through parallel states and hybrid states. These states are different from the traditional spurious states of the Hopfield model, as they correspond to global minima of the free energy, and at least one of the patterns is perfectly retrieved. Agliari et al. (2013) introduced these networks in immunology, to illustrate that a sparse lymphocyte network between B cell and T cell allows targeting several pathogens at the same time (through these parallel states and hybrid states).

Decelle et al. (2017, 2018) have also studied the Spin-Spin RBM with $\alpha = \mathcal{O}(1)$, but with another assumption on the distribution of the weight matrix. By writing the singular value decomposition of the weight matrix, it encodes $K \ll M, N$ singular values, and an iid Gaussian noise perturbs each term of the matrix. They derive a phase diagram for RBM similar to the Hopfield model. Moreover, they show the evolution of the RBM in this phase diagram during training on real data, and the transition from the paramagnetic to the ferromagnetic phase: during training, a small number K of singular values emerge from bulk. We also observe this phenomenon in Chapter 4.

II

Sampling an energy landscape with Restricted Boltzmann Machines

Part II summary

The landscape $E^{\text{eff}}(\mathbf{v})$ of the RBM can represent many models from statistical physics. Nevertheless, RBM has additional degrees of freedom through its landscape $E(\mathbf{v},\mathbf{h})$ which controls the representations of a vector $\mathbf{v}$ in the hidden space through $P(\mathbf{h}|\mathbf{v})$. These representations are not exploited by the canonical RBM sampling algorithm, Alternating Gibbs Sampling. Without exploiting these representations, we show that this algorithm is just as inefficient as Metropolis-Hastings.

These representations are an asset of RBM, and can allow better interpretation of trajectories in $E^{\text{eff}}(\mathbf{v})$, as well as speed up its sampling, as we will see in Chapters 3 and 4.

- Chapter 3 is based on our following publication:

  [1] Roussel, C., Cocco, S., and Monasson, R. (2021). Barriers and Dynamical Paths in alternating Gibbs sampling of restricted Boltzmann machines, *Physics Review E*

- Chapter 4 is based on our paper, *in preparation*:

  [2] Roussel, C., Cocco, S., and Monasson, R. (2021). Improving Sampling of restricted Boltzmann machines with Deep Tempering, *In preparation*

3

Barriers and Dynamical Paths in Alternating Gibbs Sampling of Restricted Boltzmann Machines

3.1. Alternating Gibbs Sampling of multi-modal distributions

This section examines how long it takes for AGS to sample complex energy landscapes with several states associated with multi-modal distributions, and how the hidden representations learned by the RBM can be used to improve the sampling. This section repeats the results presented in Roussel et al. (2021).

We consider first the Curie-Weiss model at low temperature, where two ferromagnetic states with opposite magnetizations coexist. We then turn to the case of the Hopfield model, in which different, uncorrelated states coexist. We finally study the general, more complex situation, in which multiple correlated states are present. We show that AGS is just as inefficient as a Metropolis-Hastings algorithm for sampling these energy landscapes. Nevertheless, RBM can learn useful representations, and the hidden units can encode collective modes of the data. Using these representations, it is possible to improve AGS by adding Metropolis Hastings to the space of hidden units.

3.1.1 Case of bi-modal distributions

We consider the Curie-Weiss (CW) model over N spins, $v_i = \pm 1$. The energy function is defined in Eq. (2.10), with $w^2 = \beta$.

The CW model can be represented with a RBM with N visible units (with potentials $\mathcal{V}_i = 0$) and $M = 1$ hidden unit with a quadratic potential $\mathcal{U}(h) = \frac{h^2}{2}$. The weights $W_{i,\mu=1}$ are uniform and equal to $\frac{w}{\sqrt{N}}$[1]. The energy of the RBM in Eq. (1.2) reads

$$E^{\mathrm{CW}}(\mathbf{v}, \mathbf{h}) = -\frac{w}{\sqrt{N}} \sum_{i=1}^{N} v_i h + \frac{h^2}{2}. \tag{3.1}$$

After integration over h, it is straightforward to check that the effective energy in Eq. (1.4) coincides with the CW energy in Eq. (2.10).

3.1.1.1 Barriers and sampling time for MH procedures

As explained in Chapter 2, for $w^2 > 1$ and infinite-size limit $N \to \infty$, the average magnetization of the spins, $m = \frac{1}{N} \sum_{i=1}^{N} v_i$, spontaneously acquires a nonzero value. The free energy $f(m)$ is an even function of m, with a double-well shape. The two opposite values

[1] We have checked that numerical experiments with RBM trained by gradient ascent on data sampled from the Curie-Weiss model converge to this solution.

of the spontaneous magnetization, roots of $f'(m^*) = 0$, define two collective states of the system. Notice that $m = 0$ is a local maximum of the free energy.

To go from one mode of the distribution to the other, a macroscopic number of spins has to be flipped. Local sampling processes, such as Metropolis-Hastings described in Algorithm 3[2] take exponential-in-N time to do so:

$$\tau \sim \exp(N\Delta f), \quad \text{where} \quad \Delta f \equiv f(\pm m^*) - f(0), \tag{3.2}$$

is the free energy barrier between the minima $m = \pm m^*$ and the local maximum $m = 0$ of the free energy landscape. Consequently, for large N, the system is stuck in one state/mode for long times, and thermalization is practically impossible.

Algorithm 3: Metropolis-Hastings algorithm

Pick $\mathbf{v}^0 \in \{-1,1\}^N$ at random ;
for $t \in [\![0,T]\!]$ **do**
  $\mathbf{v}^{'} = \mathbf{v}^t$;
  Choose $i \in [\![1,N]\!]$ uniformly at random;
  $v_i' = -v_i^t$;
  Generate a uniform random $u \in [0,1]$;
  if $u \leq \min\Big(1, \exp\big[-(E^{\mathit{eff}}(\mathbf{v}') - E^{\mathit{eff}}(\mathbf{v}^t))\big]\Big)$ **then**
    $\mathbf{v}^{t+1} = \mathbf{v}'$;
  else
    $\mathbf{v}^{t+1} = \mathbf{v}^t$;
  end
end

3.1.1.2 Optimal sampling paths with AGS

The AGS procedure can be entirely described in terms of the magnetizations m of the visible configurations and of the values h of the hidden unit. To get intensive quantities in the large N limit, we rescale $h \to h/\sqrt{N}$. The conditional configuration of the hidden unit h^{t+1} given a visible configuration with magnetization m^t then simply reads,

$$P(h^{t+1}|m^t) = \frac{1}{\sqrt{2\pi/N}} \exp\left(-\frac{N}{2}(h^{t+1} - w\,m^t)^2\right). \tag{3.3}$$

Some care must be taken to write the conditional distribution of the magnetization m^t given the hidden unit h^t. First, the conditional probability of $\mathbf{v}^t$ is

$$\begin{aligned} P(\mathbf{v}^t|h^t) &= \prod_{i=1}^{N} \frac{\exp(w\,h^t\,v_i^t)}{2\cosh(w\,h^t)} \\ &= \exp\Big(N\big(w\,h^t\,m^t - \log 2\cosh(w\,h^t)\big)\Big), \end{aligned} \tag{3.4}$$

which depends on m^t as expected. Second, to turn the probability over visible configurations into a probability over magnetizations, we have to take into account the entropies of the latter. We end up with the normalized (to dominant order in N) conditional probability,

$$\begin{aligned} P(m^t|h^t) &= \exp\Big(N\big(w\,h^t\,m^t - \log 2\cosh(w\,h^t)\big)\Big) \\ &\times \exp\Big(N\,\mathcal{S}(m^t)\Big). \end{aligned} \tag{3.5}$$

[2] The specific choice of the Metropolis rule is irrelevant here; other choices, such as Glauber rule, (Glauber, 1963), do not affect the leading behavior of τ.

We may now express the probability to go from one minimum of the free energy landscape to the other in T steps of AGS. To do so, we compute the probability $P\left(m^T|m^0\right)$ that, given magnetization $m^0 = m^*$ at time $t = 0$, the dynamics associated with AGS reaches magnetization $m^T = -m^*$ at time $t = T$. This conditional probability may be computed by means of the saddle-point method in the thermodynamic limit $N \to \infty$ (for finite T):

$$\begin{aligned} P\left(m^T|m^0\right) &= \int dh^1 \dots dh^T \int dm^1 \dots dm^{T-1} \prod_{t=0}^{T-1} P(m^{t+1}|h^{t+1})\, P(h^{t+1}|m^t) \\ &= \exp\left(-N \min_{\{m^t,h^t\}} \Phi\left(\{m^t,h^t\}\right)\right), \end{aligned} \tag{3.6}$$

where

$$\Phi\left(\{m^t,h^t\}\right) = \sum_{t=0}^{T-1} \delta\Phi\left(t \to t+1\right), \tag{3.7}$$

and, according to Eqs. (3.3) and (3.5),

$$\begin{aligned} \delta\Phi\left(t \to t+1\right) &= \frac{1}{2}(h^{t+1} - w\, m^t)^2 + \log\left(2\cosh\left(w\, h^{t+1}\right)\right) \\ &- w\, m^{t+1}\, h^{t+1} - \mathcal{S}(m^{t+1}). \end{aligned} \tag{3.8}$$

The set of magnetizations m^t and hidden-unit values h^t minimizing the action Φ in Eq. (3.6) define the most likely path, with AGS, capable of moving the system from one state to another in T alternating sampling steps. They are solutions of the following extremization equations for Φ, which must be fulfilled at all steps $1 \le t \le T-1$:

$$\begin{aligned} w(m^{t+1} + m^t) &= h^{t+1} + w\tanh(w\, h^{t+1}), \\ \text{arctanh}\,(m^t) &= w(h^t + h^{t+1}) - w^2 m^t. \end{aligned} \tag{3.9}$$

An example of transition path obtained through brute force numerical minimization of $\Phi\left(\{m^t,h^t\}\right)$ is shown in Fig. 3.1(a). It is composed of two portions:

- an initial part of the trajectory ascending the free energy landscape from one stable state, say, $+m^*$ up to the free energy local maximum, $m = 0$. This part is associated with an exponentially small probability, *i.e.*, to a positive contribution to the action, $\delta\Phi > 0$ (Fig. 3.1(b)).

- a final part of the trajectory descending the free energy landscape from the local maximum $m = 0$ down to the other stable state, say, $-m^*$. This stretch does not seem to contribute to the action, $\delta\Phi \simeq 0$ (Fig. 3.1(b)).

As the number T of steps increases the total action decreases, as expected, and quickly converges toward a minimal value (Fig. 3.1(c)). We show below that the scenario above can be analytically understood when T is sent to infinity.

3.1.1.3 Analytical expressions of the optimal trajectories in the $T \to \infty$ limit

In the infinite T limit, the equations of motion (3.9) admit two distinct solutions that correspond to the two-fold behavior empirically observed for finite T.

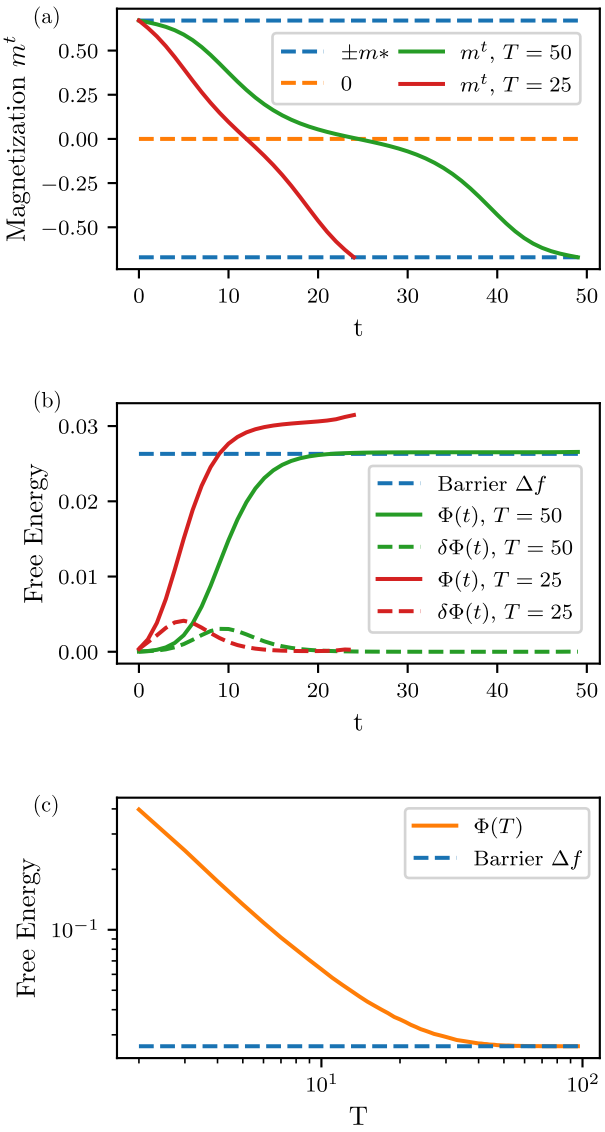


Figure 3.1: Numerical minimization of $\Phi(\{m^t, h^t\})$ for $w = 1.1$ with boundary conditions $m^0 = -m^T = m^*$. (a) Optimal time course of the magnetizations for $T = 25$ (red) and $T = 50$ (green) AGS steps. (b) Contributions $\delta\Phi(t)$ and full action $\Phi(t)$ as a function of the number of AGS steps for the optimal paths of duration $T = 25$ and $T = 50$. (c) Cost Φ of the optimal path as a function of T. For large T, Φ reaches from above a plateau equals to the free energy barrier Δf of the CW model, see Eq. (3.2). The convergence is exponentially fast, with decay time $T_{\text{decay}} \sim 1/\log(w^2)$.

a) Instanton-like trajectories

The ascending trajectories correspond to instantons, connecting a local minimum of the free energy to the local maximum, and are described by

$$\begin{aligned} m^{t+1} &= \frac{1}{w^2}\operatorname{arctanh}(m^t), \\ h^{t+1} &= w\, m^{t+1}. \end{aligned} \tag{3.10}$$

Inserting these equations into Eq. (3.8), the contribution to the action associated with one AGS step reads, after some algebra,

$$\delta\Phi = f(m^{t+1}) - f(m^t), \tag{3.11}$$

where $f(m)$ is the free energy of the CW model for magnetization m. The only stable fixed point of this dynamics is the local maximum of $f(m)$ in $m=0$. Starting from $m^0 = m^*$, the dynamics converges to $m=0$ for $T \to \infty$. Along this path, $\Phi\left(\{m^t, h^t\}\right) \underset{T\to\infty}{\longrightarrow} f(0) - f(m^*) = \Delta f$ (Fig. 3.1(c)). Hence, this path has a log-probability (per variable) equal to minus the free energy barrier separating the minima of the landscape.

b) Thermalization-like trajectories

The descending portion of the trajectory corresponds to relaxation toward the other minimum of the free energy and is described by the following solution of the extremization equations:

$$\begin{aligned} m^{t+1} &= \tanh\left(w^2 m^t\right), \\ h^{t+1} &= w\, m^t. \end{aligned} \tag{3.12}$$

We find that the contribution of an alternating step of AGS to the action vanishes

$$\delta\Phi = 0. \tag{3.13}$$

The stable fixed points of the dynamics are the two minima of $f(m)$. Starting from $m^0 = 0$ at time $t=0$, the dynamics converges, when $T \to \infty$, to the spontaneous magnetization $\pm m^*$ associated with the minima of $f(m)$. Along this relaxation part of the trajectory, $\Phi\left(\{m^t, h^t\}\right) = 0$.

As a summary, the probability that a sequence of T steps of Alternating Gibbs Sampling brings the system from one minimum of the free energy to the other is given, to the dominant order in N, by $\exp(-N\Delta f)$. This result holds when N and T are very large (but with $T \ll N$). We conclude that it will take the same time τ as with the MH procedure, see Eq. (3.2), for the system to switch state. In other words, AGS is as inefficient as MH for sampling the bi-modal distribution associated with the CW model.

3.1.2 Case of unstructured multi-modal distributions

We now consider the case of a multi-modal distribution, where more than two states have high probabilities.

3.1.2.1 Hopfield model

Let us call $\boldsymbol{\xi}^\mu$ ($\mu = 1...M$) (M finite) the centers of the states, which we suppose to be orthogonal in the infinite N limit. We assume that $\xi_i^\mu = \pm 1$. The order parameter is the M-dimensional vector of magnetizations along the centers, called patterns,

$$m_\mu = \frac{1}{N}\sum_{i=1}^{N} \langle v_i \rangle \xi_i^\mu. \tag{3.14}$$

We will hereafter consider the limit $\frac{M}{N} \to 0$. To be more precise, the energy over the visible configurations corresponds to the Hopfield model (Eq. (2.8)) with $\beta = w^2$. By inserting Eq. (3.14) into Eq. (2.8), the free energy (per site) can be written as a function of the magnetizations along the centers $\mathbf{m}$

$$f(\mathbf{m}) = -\frac{w^2}{2}\sum_{\mu=1}^{M} m_\mu^2 - \mathcal{S}^{\mathrm{Hop}}(\mathbf{m}), \tag{3.15}$$

where $\mathcal{S}^{\rm Hop}(\mathbf{m})$ denotes the entropy of the visible configurations at fixed magnetizations. It can be computed from the following Legendre formula,

$$\mathcal{S}^{\rm Hop}(\mathbf{m}) = \min_{\boldsymbol{\lambda}} \left(\frac{1}{N} \sum_{i=1}^{N} \log 2\cosh \left(\sum_{\mu=1}^{M} \xi_i^\mu \lambda_\mu \right) - \sum_{\mu=1}^{M} \lambda_\mu m_\mu \right). \tag{3.16}$$

The minimum is reached in the unique $\boldsymbol{\lambda}^*$ such that

$$m_\mu = \frac{1}{N} \sum_i \xi_i^\mu \tanh \left(\sum_\nu \xi_i^\nu \lambda_\nu^* \right), \tag{3.17}$$

for all μ's. $\mathcal{S}^{\rm Hop}(\mathbf{m})$ can be expressed as a function of $\boldsymbol{\lambda}^*$ and the binary entropy $\mathcal{S}(m)$ defined in Eq. (2.11)

$$\mathcal{S}^{\rm Hop}(\mathbf{m}) = \frac{1}{N} \sum_i \mathcal{S} \left(\tanh \left(\sum_\mu \xi_i^\mu \lambda_\mu^* \right) \right). \tag{3.18}$$

The Hopfield model can be represented with a RBM with N visible units (with potentials $\mathcal{V}_i = 0$) and M hidden units subject to the quadratic potential $\mathcal{U}(h) = \frac{h^2}{2}$ (Eq. (2.15)), and the weights must full the constraints defined in Eq. (2.16).

As explained in Chapter 2, these conditions do not uniquely define the weight matrix $\mathbf{W}$. The energy is invariant under any transformation $\mathbf{W} \to \mathbf{W} \times \mathbf{O}$, where $\mathbf{O}$ is an orthogonal matrix. We choose for now the following parametrization for the weight matrix $\mathbf{W}$:

$$W_{i\mu} = \frac{w}{\sqrt{N}} \xi_i^\mu. \tag{3.19}$$

Alternative choices will be discussed later.

3.1.2.2 Optimal sampling with AGS

The AGS procedure can be entirely described in terms of M magnetizations $\mathbf{m}$ of the visible configurations and of the values $\mathbf{h}$ of the M hidden units. As in the case of the CW model, to get intensive quantities in the large N limit, we rescale $\mathbf{h} \to \mathbf{h}/\sqrt{N}$. The conditional configuration of the hidden unit $\mathbf{h}^{t+1}$ given a visible configuration with magnetization $\mathbf{m}^t$ is factorized, and reads

$$P(h_\mu^{t+1} | \mathbf{m}^t) = \frac{1}{\sqrt{2\pi/N}} \exp \left(-\frac{N}{2} \left(h_\mu^{t+1} - w\, m_\mu^t \right)^2 \right). \tag{3.20}$$

The conditional probability of $\mathbf{m}^t$ given the hidden unit $\mathbf{h}^t$ can be easily written to the leading order in N, with the result

$$\begin{aligned} P(m_\mu^t | h_\mu^t) &= \exp \left(-\sum_{i=1}^{N} \log 2\cosh \left(w \sum_{\mu=1}^{M} \xi_i^\mu h_\mu^t \right) \right) \\ &\times \exp \left(N \left(w \sum_{\mu=1}^{M} h_\mu^t m_\mu^t + \mathcal{S}^{\rm Hop}(\mathbf{m}^t) \right) \right). \end{aligned} \tag{3.21}$$

Similarly to the CW case, the probability of going from one minimum of the free energy landscape to another in T steps of AGS can be expressed as

$$P\left(\mathbf{m}^T | \mathbf{m}^0\right) = \exp \left(-N \min_{\{m^t, h^t\}} \Phi \left(\{\mathbf{m}^t, \mathbf{h}^t\} \right) \right), \tag{3.22}$$

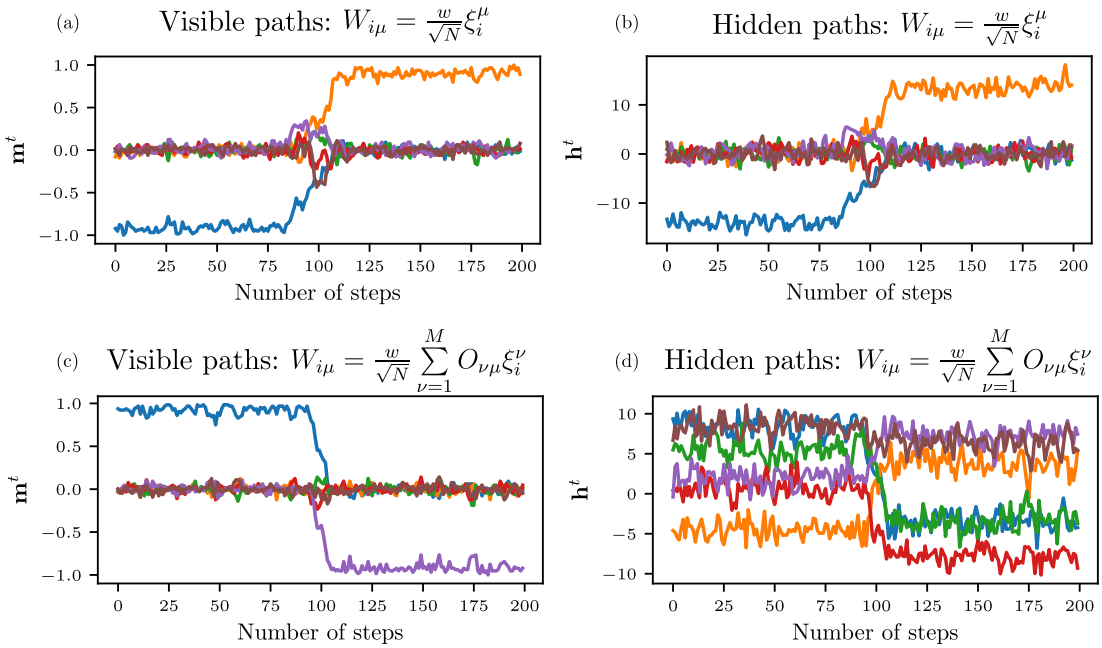


Figure 3.2: Hopfield model with $N = 128$ spins, $M = 6$ patterns and $w = 1.35$; each color refers to one index μ. Examples of transition between two states for $W_{i\mu} = \frac{w}{\sqrt{N}}\xi_i^\mu$ (panels a-b) and for $W_{i\mu} = \frac{w}{\sqrt{N}}\sum_\nu O_{\nu\mu}\xi_i^\nu$ (panels c-d). (a-c) Magnetizations m_μ along the patterns as functions of the number of AGS steps. (b-d) Hidden unit values h_μ as functions of the number of AGS steps for the same transitions as in panels (a-c).

where the action $\Phi\left(\{\mathbf{m}^t, \mathbf{h}^t\}\right)$ is the sum of

$$\begin{aligned}\delta\Phi\left(t \to t+1\right) &= \frac{1}{2}\sum_\mu (h_\mu^{t+1} - w\, m_\mu^t)^2 + \frac{1}{N}\sum_i \log 2\cosh\left(w\sum_\mu \xi_i^\mu h_\mu^{t+1}\right) \\ &- w\sum_\mu m_\mu^{t+1} h_\mu^{t+1} - \mathcal{S}^{\text{Hop}}(\mathbf{m}^{t+1}). \end{aligned} \tag{3.23}$$

The set of magnetizations $\mathbf{m}^t$ and hidden-unit values $\mathbf{h}^t$ minimizing the action Φ define the most likely path interpolating between two states in T AGS steps. They are solutions of the following extremization equations for Φ, which must be fulfilled at all steps $1 \le t \le T-1$:

$$\begin{aligned}(\lambda^*)_\mu^t &= w(h_\mu^t + h_\mu^{t+1}) - w^2 m_\mu^t, \\ w(m_\mu^{t+1} + m_\mu^t) &= h_\mu^{t+1} + \frac{w}{N}\sum_i \xi_i^\mu \tanh\left(w\sum_\nu \xi_i^\nu h_\nu^{t+1}\right). \end{aligned} \tag{3.24}$$

3.1.2.3 Analytical expressions of the optimal trajectories in the $T \to \infty$ limit

As for the CW model, we find

a) Instanton-like trajectories

These are defined by

$$\begin{aligned} h_\mu^{t+1} &= w\, m_\mu^{t+1} = \frac{1}{w}(\lambda^*)_\mu^t, \\ m_\mu^t &= \frac{1}{N}\sum_{i=1}^{N} \xi_i^\mu \tanh\left(w^2 \sum_{\mu=1}^{M} \xi_i^\mu m_\mu^{t+1}\right). \end{aligned} \tag{3.25}$$

The contribution to the action associated with this AGS step reads

$$\delta\Phi = f(\mathbf{m}^{t+1}) - f(\mathbf{m}^{t}). \tag{3.26}$$

b) Thermalization-like trajectories

These correspond to

$$\begin{aligned} h_\mu^{t+1} &= w\, m_\mu^t = \frac{1}{w}(\lambda^*)_\mu^{t+1}, \\ m_\mu^{t+1} &= \frac{1}{N}\sum_{i=1}^{N} \xi_i^\mu \tanh\left(w^2 \sum_{\mu=1}^{M} \xi_i^\mu m_\mu^t \right). \end{aligned} \tag{3.27}$$

The contribution to the action associated with such an AGS step vanishes:

$$\delta\Phi = 0. \tag{3.28}$$

c) Orthogonal transformation of the weight matrix

The computation can be repeated for a weight matrix $\tilde{\mathbf{W}} = \mathbf{W} \times \mathbf{O}$ where $\mathbf{O}$ is an orthogonal matrix. In the limit $T \to \infty$, instanton-like and themalization-like trajectories are found, and contributions to the action for both trajectories are the same as for $\mathbf{W}$. Therefore, the barriers are identical for all rotations $\mathbf{O}$. However, contrary to the previous case where the hidden unit h_μ codes for the magnetization m_μ only ($h_\mu = w\, m_\mu$), under an orthogonal transformation of the weight matrix, the hidden unit h_μ represents a superposition: $h_\mu = w \sum_{\nu=1}^{M} O_{\nu\mu} m_\nu$.

3.1.2.4 Transition paths between Mattis states

In the thermodynamic limit, the $\boldsymbol{\xi}^\mu$ are orthogonal. The free energy landscape $f(\mathbf{m})$ (Eq. (3.15)) exhibits a large variety of critical points when $w^2 > 1$ (Amit et al., 1985a; Amit, 1989), defined through Eq. (3.14), with

$$\langle v_i \rangle = \tanh\left(w^2 \sum_{\mu=1}^{M} \xi_i^\mu m_\mu \right). \tag{3.29}$$

Global minima of Eq. (3.15) are reached for magnetization with only one nonzero component, called Mattis states (Procesi and Tirozzi, 1990). Numerical experiments for finite N exhibit transitions between the Mattis states, for all orthogonal transformation $\tilde{\mathbf{W}} = \mathbf{W} \times \mathbf{O}$ (Figs. 3.2(a) and (c)). However, the hidden representations of the path between Mattis states may be easy or difficult to interpret depending on the orthogonal transformation (Figs. 3.2(b) and (d)).

Furthermore, as for CW, for large T and N (with $T \ll N$), the probability to go from one Mattis state to another scale as $\exp(-N\Delta f)$. The barrier Δf depends on w and is always positive for $w^2 > 1$ (Amit et al., 1985a). Therefore, AGS is as inefficient as MH for sampling the Hopfield model.

3.1.3 Case of structured multi-modal distributions

We now turn to a more complex case of multi-modal distributions, in which the free energy minima do not correspond to orthogonal pockets of configurations in the visible space but are structured. In addition, contrary to the previous models, the hidden units h_μ, which can be discrete or continuous, are now subject to an arbitrary, not necessarily quadratic potential $\mathcal{U}_\mu(h_\mu)$. Common potentials in the machine learning community are

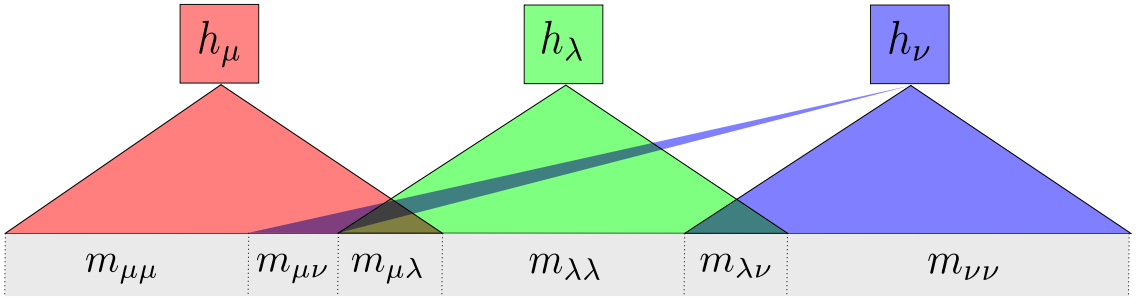


Figure 3.3: Illustration of the structured model for $M = 3$ hidden units. The structural overlap matrix $\boldsymbol{\alpha}$ divides the visible layer into six different areas labeled by μ,ν, with $1 \le \mu \le \nu \le M$. For each area, we define the corresponding normalized magnetization $m_{\mu\nu}$.

Bernoulli or ReLU potentials (Nair and Hinton, 2010; Tubiana and Monasson, 2017), see Appendix B.1.

The $N \to \infty$ visible units v_i are ± 1 variables, and no potential acts on them ($\mathcal{V}_i = 0$). A visible unit v_i is connected to one or two hidden units with equal weights $\frac{w}{\sqrt{N}}$, following a pattern of connections shown in Fig. 3.3. We define the adjacency matrix $\boldsymbol{a}$ of our model as:

$$a_{i\mu} = \begin{cases} 1 & \text{if } W_{i\mu} = \frac{w}{\sqrt{N}} \\ 0 & \text{otherwise.} \end{cases} \tag{3.30}$$

From the adjacency matrix $\boldsymbol{a}$, we define the overlap matrix $\boldsymbol{\alpha}$ and the magnetization matrix $\boldsymbol{m}$:

$$\alpha_{\mu\mu} = \frac{1}{N}\sum_{i=1}^{N} a_{i\mu} \prod_{\nu\neq\mu}(1-a_{i\nu}), \tag{3.31}$$

$$\alpha_{\mu\nu} = \frac{1}{N}\sum_{i=1}^{N} a_{i\mu}a_{i\nu}, \tag{3.32}$$

$$m_{\mu\mu} = \frac{1}{\alpha_{\mu\mu}N}\sum_{i=1}^{N} \langle v_i\rangle\, a_{i\mu} \prod_{\nu\neq\mu}(1-a_{i\nu}), \tag{3.33}$$

$$m_{\mu\nu} = \frac{1}{\alpha_{\mu\nu}N}\sum_{i=1}^{N} \langle v_i\rangle\, a_{i\mu}a_{i\nu}. \tag{3.34}$$

In other words, there are $\alpha_{\mu\mu}N$ visible units connected only to h_μ, and $\alpha_{\mu\nu}N$ visible units connected to both h_μ and h_ν. The overlap matrix $\boldsymbol{\alpha}$ partitions the visible layer into $\frac{M(M+1)}{2}$ subsets with associated magnetizations $\mathbf{m}$ (Fig. 3.3).

It is straightforward to write down the free energy per variable $f(\mathbf{m})$ as a function of the $\frac{M(M+1)}{2}$ magnetizations, with the result

$$f(\mathbf{m}) = -\sum_{\mu=1}^{M} \hat{\Gamma}_\mu\left(w\sum_{\nu=1}^{M}\alpha_{\mu\nu}\, m_{\mu\nu}\right) - \sum_{\nu\le\mu}\alpha_{\mu\nu}\,\mathcal{S}(m_{\mu\nu}), \tag{3.35}$$

where $\hat{\Gamma}_\mu$ is the rescaled cumulative generative function associated with the hidden potential $\mathcal{U}_\mu$, see Eq. (1.4) and Appendix B.1, and $\mathcal{S}(m)$ is the entropy associated with a single ± 1 variable with magnetization m. The minima of $f(\mathbf{m})$ obey the following self-consistent

equations,

$$\begin{aligned} m^*_{\mu\mu} &= \tanh\left(w\, f_\mu\left(I^*_\mu\right)\right), \\ m^*_{\mu\nu} &= \frac{m^*_{\mu\mu}+m^*_{\nu\nu}}{1+m^*_{\mu\mu}m^*_{\nu\nu}}, \end{aligned} \tag{3.36}$$

where $I^*_\mu = w\sum_{\nu=1}^{M}\alpha_{\mu\nu}m^*_{\mu\nu}$ is the input received by the hidden unit h_μ and $f_\mu = \hat{\Gamma}'_\mu$ is the transfer function associated with the hidden unit h_μ.

3.1.3.1 Optimal sampling paths with AGS

We may now express the conditional probabilities of the magnetization matrix $\mathbf{m}$ (of dimension $M\times M$) and of the hidden-unit value vector $\mathbf{h}$ (of dimension M) following what was done for the simpler models in the previous sections. We first write the conditional probability of the hidden configuration given a set of visible activities,

$$\begin{aligned} P(h^{t+1}_\mu|\mathbf{m}^t) &= \frac{\exp\left(-N\left(\mathcal{U}_\mu(h^{t+1}_\mu)-h^{t+1}_\mu I^t_\mu\right)\right)}{\int dh\exp\left(-N\left(\mathcal{U}_\mu(h)-hI^t_\mu\right)\right)} \\ &\simeq \exp\left(-N\left(\mathcal{U}_\mu(h^{t+1}_\mu)-h^{t+1}_\mu I^t_\mu\right)\right) \\ &\times \exp\left(-N\hat{\Gamma}_\mu\left(I^t_\mu\right)\right), \end{aligned} \tag{3.37}$$

where we have defined the input $I^t_\mu = w\sum_{\nu=1}^{M}\alpha_{\mu\nu}m^t_{\mu\nu}$ received by the hidden unit h_μ given the magnetization matrix $\mathbf{m}^t$.

In turn, we write the conditional probability over magnetizations given the set of hidden-unit values (to dominant order in N),

$$\begin{aligned} P(\mathbf{m}^t|\mathbf{h}^t) &\simeq \exp\left(N\left(\sum_{\mu=1}^{M} I^t_\mu h^t_\mu - \alpha_{\mu\mu}\log 2\cosh\left(w\,h^t_\mu\right)\right)\right) \\ &\times \exp\left(N\left(-\sum_{\mu\le\nu}\alpha_{\mu\nu}\log 2\cosh\left(w(h^t_\mu+h^t_\nu)\right)+\alpha_{\mu\nu}\,\mathcal{S}(m^t_{\mu\nu})\right)\right). \end{aligned} \tag{3.38}$$

The probability to go from one minimum of the free energy landscape to another in T steps of AGS, $P\left(\mathbf{m}^T|\mathbf{m}^0\right)$, takes the same form as Eq. (3.6), where the action $\Phi\left(\{\mathbf{m}^t,\mathbf{h}^t\}\right)$ is the sum of

$$\begin{aligned} \delta\Phi\left(t\to t+1\right) &= \sum_{\mu=1}^{M}\mathcal{U}_\mu(h^{t+1}_\mu)+\sum_{\mu=1}^{M}\hat{\Gamma}_\mu\left(I^t_\mu\right)+\sum_{\mu=1}^{M}\alpha_{\mu\mu}\log 2\cosh\left(w\,h^{t+1}_\mu\right) \\ &+ \sum_{\mu\le\nu}\alpha_{\mu\nu}\log 2\cosh\left(w(h^{t+1}_\mu+h^{t+1}_\nu)\right)-\sum_{\mu=1}^{M}(I^{t+1}_\mu+I^t_\mu)h^{t+1}_\mu\sum_{\mu\le\nu}\alpha_{\mu\nu}\,\mathcal{S}(m^{t+1}_{\mu\nu}). \end{aligned} \tag{3.39}$$

Notice that the previous expression extends the model studied in Section 3.1.1, which can be recovered for $M=1$, $\alpha_{11}=1$ with a quadratic potential $\mathcal{U}(h)=\frac{h^2}{2}$.

We show the best path found through minimization of Φ in the case of $M=2$ hidden units, quadratic $\mathcal{U}(h)$, $w>1$, and small positive overlap α_{12}. The free energy landscape $f(\mathbf{m})$ represents two coupled Curie-Weiss models (Fig. 3.4(a)), and displays two global

minima and two local minima. The green trajectory shows the most likely path connecting the two global minima in $T = 100$ steps. Along this path, m_{11}^t and m_{22}^t, and therefore h_1^t and h_2^t, have asymmetric behaviors. In contradistinction, trajectories along which m_{11}^t and m_{22}^t are equal, have exponentially smaller probabilities, see the red path. We elucidate this behavior below.

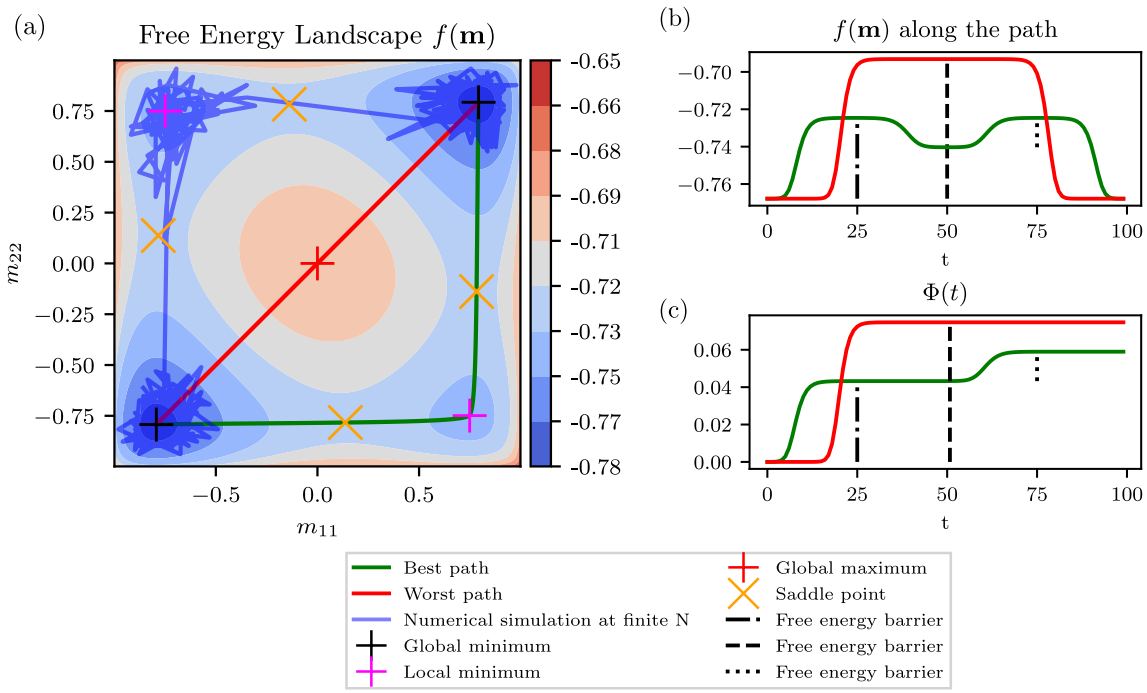


Figure 3.4: (a) Free energy landscape for a coupled Curie-Weiss model with two global minima and two local minima. $M = 2$, $\mathcal{U}(h) = \frac{h^2}{2}$, $w = 1.15\sqrt{2}$ and $\alpha_{12} = 0.02$. Among the many paths connecting the two global minima in $T = 100$ steps, the green path is the optimal one. The red path is another path, along which both magnetizations m_{11} and m_{22} are equal at all times. The blue path is a representative trajectory found by simulating AGS for $N = 400$ and 10^5 steps. (b) Free energy $f(\mathbf{m}^t)$ along the different paths. (c) Cost $\Phi\left(\{\mathbf{m}^t, \mathbf{h}^t\}\right)$ for the different paths.

3.1.3.2 Optimal trajectories in the $T \to \infty$ limit

The set of magnetizations $\mathbf{m}^t$ and hidden-unit values $\mathbf{h}^t$ minimizing the action Φ define the most likely path, with AGS, capable of moving the system from one state to another in T alternating sampling steps. They are solutions of the following extremization equations for Φ, which must be fulfilled at all steps $1 \leq t \leq T-1$:

$$\begin{aligned} I_\mu^t + I_\mu^{t+1} &= \mathcal{U}_\mu{}'(h_\mu^{t+1}) + w\,\alpha_{\mu\mu}\tanh(w h_\mu^{t+1}) \\ &+ w\sum_{\nu\neq\mu}\alpha_{\mu\nu}\tanh\left(w\left(h_\mu^{t+1}+h_\nu^{t+1}\right)\right), \end{aligned} \tag{3.40}$$

$$m_{\mu\mu}^t = \tanh\left(w\left(h_\mu^{t+1}+h_\mu^t\right) - w\hat{\Gamma}_\mu'\left(I_\mu^t\right)\right), \tag{3.41}$$

$$m_{\mu\nu}^t = \frac{m_{\mu\mu}^t + m_{\nu\nu}^t}{1 + m_{\mu\mu}^t m_{\nu\nu}^t}. \tag{3.42}$$

In the infinite T limit, these equations of motion admit two distinct solutions.

a) Instanton-like trajectories

These solutions correspond to an increase of free energy from a local minimum, to a saddle-point of $f(\mathbf{m})$. These solutions can be written as

$$\begin{aligned} h_\mu^{t+1} &= f_\mu(I_\mu^{t+1}), \\ m_{\mu\mu}^{t} &= \tanh(w f_\mu(I_\mu^{t+1})). \end{aligned} \tag{3.43}$$

Inserting these equations into Eq. (3.39):

$$\delta\Phi = f(\mathbf{m}^{t+1}) - f(\mathbf{m}^t). \tag{3.44}$$

b) Thermalization-like trajectories

These solutions make the free energy decrease until a local minimum is reached. The relaxation solution can be written as:

$$\begin{aligned} h_\mu^{t+1} &= f_\mu(I_\mu^{t}), \\ m_{\mu\mu}^{t+1} &= \tanh(w\, f_\mu(I_\mu^{t})). \end{aligned} \tag{3.45}$$

Inserting these equations into Eq. (3.39):

$$\delta\Phi = 0. \tag{3.46}$$

While instantonic and thermalization trajectories are, strictly speaking, defined for $T \to \infty$ qualitatively analogous bouts of trajectories are observed for finite T, see Fig. 3.4(b) and (c) for the $M = 2$ example above. The green and the red paths are each composed of a sequence of instantonic and thermalization stretches. In the case of the red path, starting from a global minimum, the instantonic dynamics leads to the global maximum of $f(\mathbf{m})$. The relaxation dynamics then brings the system down to the other global minimum. In the case of the green path, starting from a global minimum, the instantonic solution leads to a saddle point of $f(\mathbf{m})$, which is unstable for the instantonic and the thermalization dynamics. Then, the relaxation dynamics leads to a local minimum of $f(\mathbf{m})$. Through another pair of instantonic/relaxation dynamics, the second global minimum is finally reached. Thus, for the green and the red paths, the action $\Phi\left(\{\mathbf{m}^t, \mathbf{h}^t\}\right)$ corresponds to the sum of the free energy barriers along the paths (Figs. 3.4(b) and (c)). These theoretical findings are corroborated by running AGS on a RBM with $N = 400$ spins, with the same overlap matrix $\boldsymbol{\alpha}$. Along the transition path allowing the RBM to interpolate from one global state to the other, hidden units are preferentially flipped one by one, see the blue path in Fig. 3.4(a).

3.1.3.3 Dependence of barrier upon structural overlap α

This section examines the influence of the structural overlap on the free energy barrier (and on the transition time) separating states. For the sake of simplicity, we focus on the case of $M = 2$ hidden units subject to quadratic potentials and restrict ourselves to small overlap values, $\alpha = \alpha_{12} \ll 1$. For $\alpha = 0$ the two global minima of $f(\mathbf{m})$ are $\mathbf{m}^*$ and $-\mathbf{m}^*$, where

$$\mathbf{m}^* = \begin{bmatrix} m_{11} = m^* \\ m_{22} = m^* \end{bmatrix}. \tag{3.47}$$

An optimal path between these two global minima follows the sequence of critical points:

$$\begin{bmatrix} m^* \\ m^* \end{bmatrix} \to \begin{bmatrix} 0 \\ m^* \end{bmatrix} \to \begin{bmatrix} -m^* \\ m^* \end{bmatrix} \to \begin{bmatrix} -m^* \\ 0 \end{bmatrix} \to \begin{bmatrix} -m^* \\ -m^* \end{bmatrix}, \tag{3.48}$$

and, for large T, Φ equals the sum of the free energy barriers along the path

$$\begin{aligned}\Phi &= -f\left(\begin{bmatrix} m^* \\ m^* \end{bmatrix}\right) + 2f\left(\begin{bmatrix} 0 \\ m^* \end{bmatrix}\right) - f\left(\begin{bmatrix} -m^* \\ m^* \end{bmatrix}\right) \\ &= -\log 2 + \frac{w^2}{2}(m^*)^2 + \mathcal{S}(m^*). \end{aligned} \tag{3.49}$$

Assume now we make small changes to the weight and overlap values, *i.e.* $w \to w + dw, \alpha \to d\alpha$. We denote the displacement of the critical points of $f(\mathbf{m})$ by $d\mathbf{m}$, and the variations of the free energy by $df(\mathbf{m})$. We will consider only contributions to the first order in $d\alpha$ and dw,

$$d\mathbf{m} = \mathbf{m}^w dw + \mathbf{m}^\alpha d\alpha, \tag{3.50}$$

$$df(\mathbf{m}) = f^w(\mathbf{m}) dw + f^\alpha(\mathbf{m}) d\alpha. \tag{3.51}$$

Expressions for $\mathbf{m}^w$, $\mathbf{m}^\alpha$, $f^w(\mathbf{m})$ and $f^\alpha(\mathbf{m})$ are given in Appendix B.2.

As the variation of α changes the critical points of $f(\mathbf{m})$, we have to change w in order to keep fixed the two global minima $\pm\mathbf{m}^*$ of $f(\mathbf{m})$. Therefore, the variation of the cost Φ between an optimal path for $\alpha = d\alpha$ and one for $\alpha = 0$ defined in Eq. (3.48) reads

$$d\Phi = -df\left(\begin{bmatrix} m^* \\ m^* \end{bmatrix}\right) + 2\,df\left(\begin{bmatrix} 0 \\ m^* \end{bmatrix}\right) - df\left(\begin{bmatrix} -m^* \\ m^* \end{bmatrix}\right). \tag{3.52}$$

As we observe in Fig. 3.5, a small overlap α reduces the cost for a wide range of w and therefore helps reduce the transition time between the global minima of f.

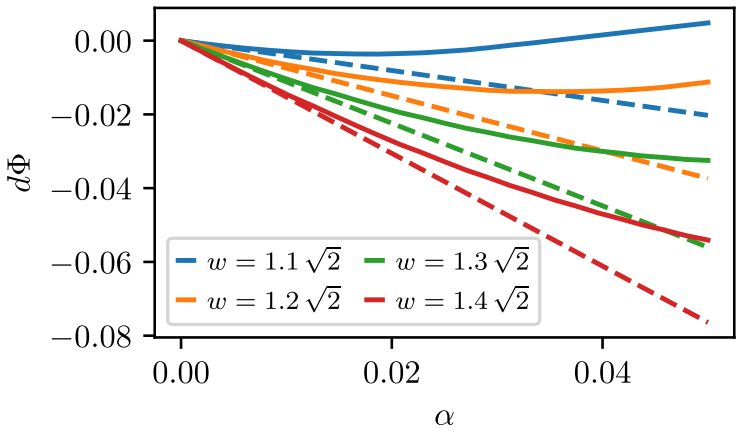


Figure 3.5: Solid lines: numerical evaluation of $d\Phi$. Dashed lines: first order perturbation theory evaluated with Eq. (3.52).

3.1.3.4 Time ordering of hidden-unit changes on sampling path

As the optimal paths for the Alternating Gibbs Sampling are the ones that minimize the sum of the free energy barriers along the paths, the optimal paths depend strongly on the overlap matrix between the hidden units. If we impose a 1d structure with periodic boundary conditions for the overlap matrix, *i.e.* $\alpha_{\mu\nu} = \alpha$ for $\nu = \mu - 1$ and $\nu = \mu + 1, \alpha_{\mu\mu} = \frac{1}{M} - \frac{M-1}{2}\alpha > 0$ (the hidden units are on a circle and have an overlap only with their two neighbors), the optimal path corresponds to an asymmetric behavior of the hidden units: they evolve one by one, according to their orders on the circle (h_μ evolves then $h_{\mu+1}$ then $h_{\mu+2}$ ...), see Fig. 3.6.

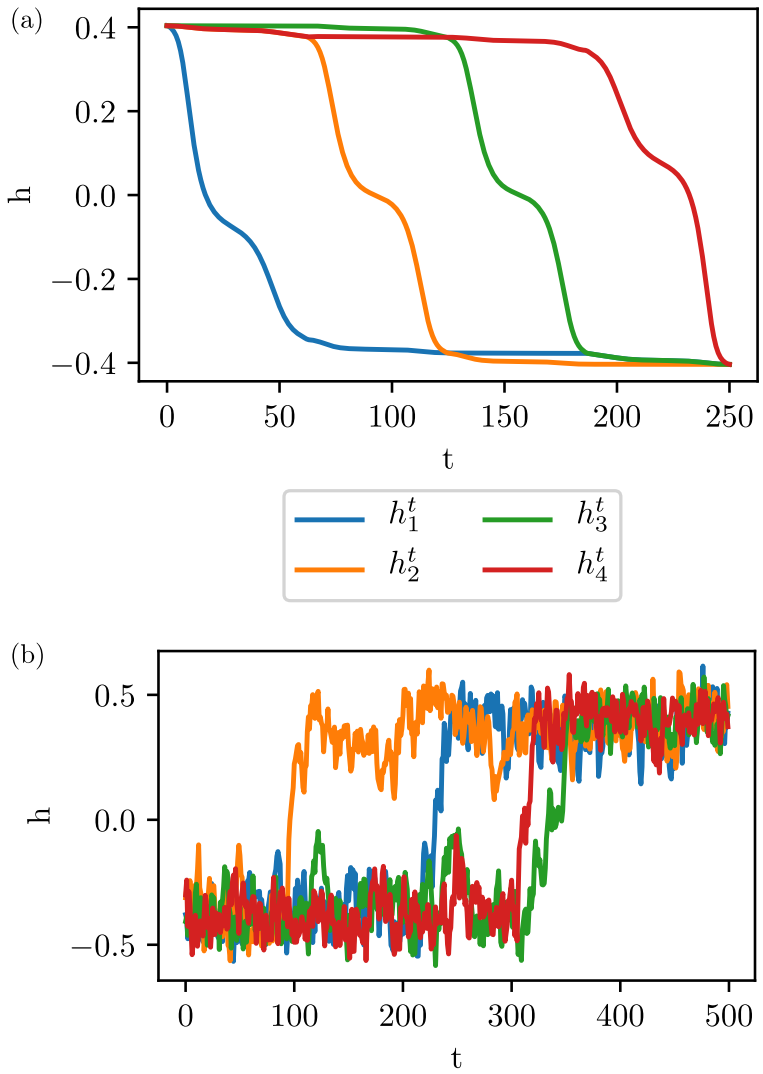


Figure 3.6: Sampling paths for structured states. $M = 4$ hidden units are arranged on a ring, with $w = 2.2$ and $\alpha = 0.02$. (a) Numerical minimization of $\Phi(\{\mathbf{m}^t, \mathbf{h}^t\})$ for $T = 250$. Hidden units are flipped according to their ordering on the ring $(h_1 \to h_2 \to h_3 \to h_4)$. There are $2M$ equivalent optimal paths. (b) Numerical experiment on a RBM with $N = 400$ visible units. Hidden units are flipped according to their ordering on the ring $(h_2 \to h_1 \to h_4 \to h_3)$.

3.1.4 Numerical experiments

We train RBM with the datasets defined in Section 1.5, then test the performances of Alternating Gibbs Sampling. The different RBM can generate high-quality configurations, but the dynamics associated with AGS struggles to mix efficiently between the data modes.

3.1.4.1 BAS

We train RBM with $2L$ real hidden units subject to quadratic potentials and ± 1 visible units. A L_1 regularization is added to the log-likelihood to enforce the sparsity of the weights. With this regularization, each hidden unit focuses on a given bar or a given stripe, see Section 3.2.2 for further details. Hidden units identify the relevant degrees of freedom of the visible units. For an image of bars, the hidden units encoding the bars are strongly magnetized, and the hidden units encoding the stripes are weakly magnetized (they are silent). It is essential to use real hidden units because each hidden unit must have more

than two equilibrium positions (strongly magnetized with positive or negative value, and weakly magnetized with positive or negative value). This behavior is not possible with discrete units like Bernoulli or Spin. AGS is inefficient for large L and long training, and the dynamics gets stuck in a bar or stripe configuration (Fig. 3.7). For short training, dynamics can escape from a given configuration, but sampled configurations are noisy.

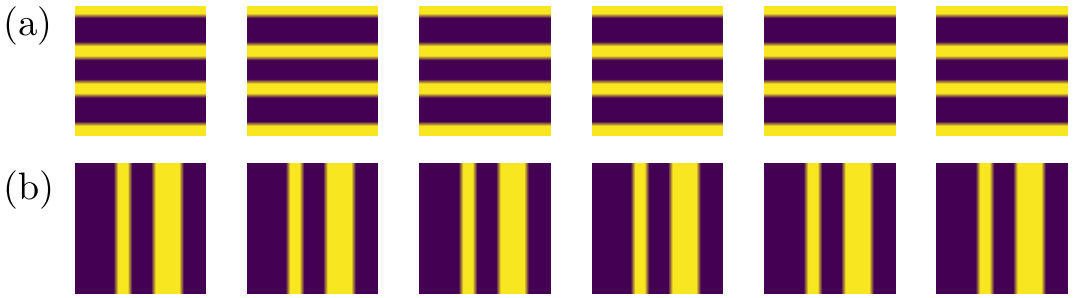


Figure 3.7: Example of configurations obtain with AGS starting from an image with stripes (a) or an image with bars (b). 1000 steps between each frame.

3.1.4.2 MNIST 0/1

We train Spin-Spin RBM (hidden and visible units are ± 1 spins). The weights of the RBM encode the digits' strokes. Zeros have many strokes in common, and so have ones. Therefore, the hidden representations of each digit are close to each other (in terms of Hamming distance). AGS is efficient to sample within a digit class and generate high-quality data, see Figs. 3.8(a) and (b). However, hidden representations of the zeros and the ones are far away from each other. Therefore, many hidden units should be simultaneously flipped to go from one class to another, which is very unlikely with AGS: the dynamics remains confined to one digit class, see Fig. 3.8(c). Notice that this observation crucially depends on the restriction of MNIST to 0-1 digits done here. RBM trained on all ten digits sample much more efficiently all classes and can reach 1 from 0 or vice versa (Desjardins et al., 2010a; Tubiana and Monasson, 2017), as other digits carve interpolating paths in the energy landscape.

3.1.4.3 Lattice Proteins

To encode amino acids (which may take 20 values), we introduce RBM with categorical (Potts) visible units. Couplings between the hidden layer and the visible layer are represented by a $M \times N \times 20$ tensor. Thus, the energy of the RBM can be written as:

$$E(\mathbf{v},\mathbf{h}) \;=\; -\sum_{i=1}^{N}\sum_{\mu=1}^{M} W_{i\mu}(v_i)h_\mu + \sum_{\mu=1}^{M}\mathcal{U}_\mu(h_\mu) + \sum_{i=1}^{N}\mathcal{V}_i(v_i). \tag{3.53}$$

The weights of the RBM encode the constraints, such as contacts between different amino acids defined by the structure. Contrary to the two previous examples, to generate high-quality proteins with the RBM, *i.e.*, proteins with a high probability to fold in a given structure, the landscape has to be sampled at low temperatures. Using the trick introduced in Tubiana et al. (2019b), we copy each hidden unit $\beta \in \mathbb{N}$ times and multiply the visible fields by the same factor β:

$$P_\beta(\mathbf{v}) \propto \int \prod_{\mu=1}^{M}\prod_{c=1}^{\beta} P(\mathbf{v}|h_\mu^c) = P(\mathbf{v})^\beta. \tag{3.54}$$

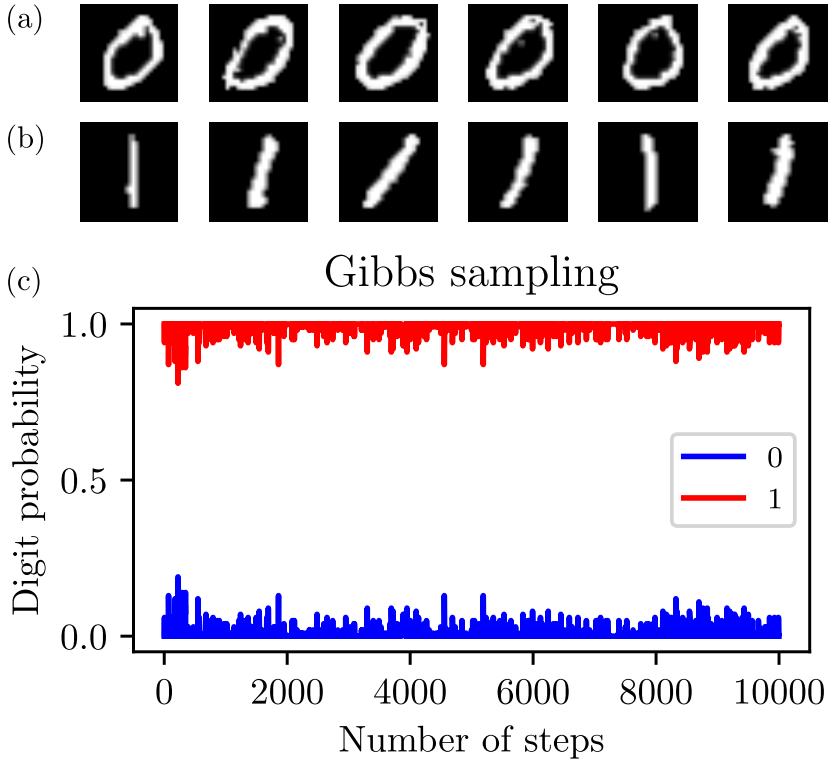


Figure 3.8: Examples of digits obtained with AGS starting from a 0 (a) and from a 1 (b); 1000 steps between each frame. (c) Probabilities that the visible unit configurations sampled by the RBM at different times are 0 (blue) or 1 (red), estimated by a random forest classifier trained on 0-1 data (Ho, 1995; Breiman, 2001). The dynamics is stuck in a given mode.

With this modification, it is possible to sample the landscape $P(\mathbf{v})$ at inverse temperature β. RBM generate high-quality proteins but struggles to mix between two families with essentially dissimilar contact maps, such as structures S_A and S_B defined in Fig. 1.4, see Fig. 3.9. Many hidden units would have to change at once, a very unlikely update with AGS to go from one family to another.

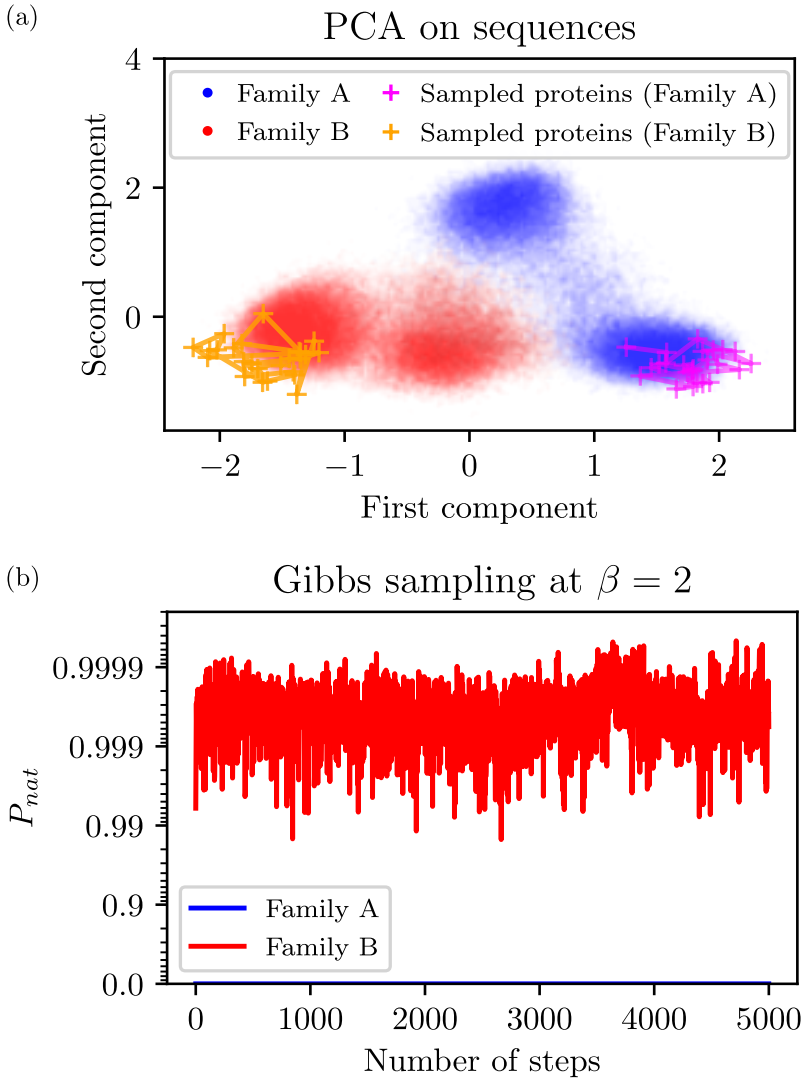


Figure 3.9: (a) Principal Component Analysis in the sequence spaces, showing the cluster structure of each family (blue and red colors). Fuchsia and orange paths are the projection of sampled proteins with AGS, starting respectively from a protein in family A and B. Sampled proteins are stuck in a given family; 250 Gibbs steps between each cross. This number of steps is larger than the decorrelation time estimated from the Hamming distance between sequences $\mathbf{v}^t$. (b) $P_{\text{nat}}(\mathbf{v}|S)$ of sampled proteins with AGS, for S_A and S_B, for an initial protein in the S_B family (orange path in panel a). RBM generates high-quality and diverse proteins, which are different from the training data.

3.2. Alternating Gibbs Sampling and dynamics in the latent space

3.2.1 Principle of the algorithm

We have shown in the previous Section 3.1 that AGS was as efficient as the local MH procedure to sample the landscape over the visible configurations, defined by the effective energy $E^{\text{eff}}(\mathbf{v})$. However, RBM offer more than this landscape, and it is natural to wonder if the representations of data could be exploited to enhance sampling performance. To do so, we propose a sampling algorithm combining AGS and moves in the *hidden unit* space, see Fig. 3.10 and Algorithm 4. The main idea is to exploit the fact that hidden units can encode specific features of the data. By doing Metropolis steps in the hidden space, we try

to flip the hidden units one by one, or by blocks, for switching on/off the features they encode. This flipping procedure must obviously preserve detailed balance. We therefore need to know the effective energy over hidden configurations, $E^{\text{eff}}(\mathbf{h})$ defined in Eq. (1.8).

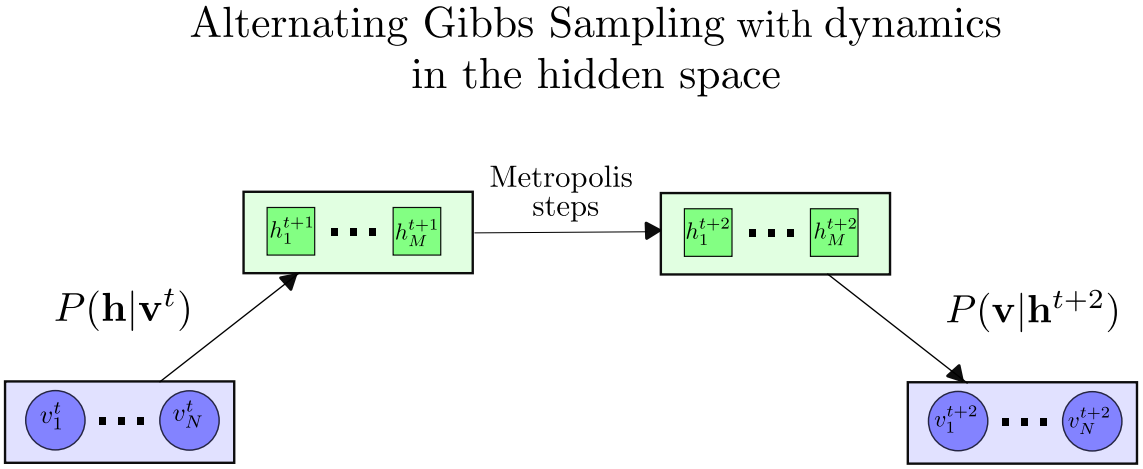


Figure 3.10: Modified Alternating Gibbs Sampling with dynamics in the hidden configuration space.

Algorithm 4: Alternating Gibbs Sampling with Metropolis-Hastings steps in latent space

Pick $\mathbf{v}^0$ in the training set;
for $t \in [\![0, T]\!]$ **do**
  $\mathbf{h}^{t+1} \sim P(\mathbf{h}|\mathbf{v}^t)$;
  π = random permutation of $[\![1, M]\!]$;
  for $i = 1...M$ **do**
    $\mu = \pi(i)$;
    $h_\mu^{t+1} \sim P(h_\mu|\mathbf{h}_{\neg\mu}^{t+1})$;
  end
  $\mathbf{v}^{t+1} \sim P(\mathbf{v}|\mathbf{h}^{t+1})$;
end

We can gain intuition about the exponential speed up offered by the algorithm in the latent space by considering first the CW model. In the absence of any bias (external field) between the + and − states of the visible variables, the effective energy $E^{\text{eff}}(\mathbf{h})$ is an even function of the hidden unit value h. A step of the sampling algorithm in the hidden space, see Algorithm 4, has thus probability $\frac{1}{2}$ to flip the hidden unit. Sampling back the visible layer will change the state of a macroscopic number of visible variables. Using MH algorithm in the hidden space is similar to using cluster algorithms for the visible spins (Swendsen and Wang, 1987; Wolff, 1989). For ferromagnetic models, these algorithms are known to be much more efficient than local MH over spins (Ray et al., 1989; Persky et al., 1996; Long et al., 2014). The latent variable is here attached to the relevant collective mode (global reversal) of the spin variables.

For the mean-field structured models defined in Section 3.1.3, as long as the overlap between the hidden units is weak, the hidden units could be flipped one by one for moderate system size N. We define the potential acting on one hidden unit, say h_μ, conditional to

the other units $\mathbf{h}_{\neg\mu}$ through

$$e_\mu(h_\mu|\mathbf{h}_{\neg\mu}) = \frac{1}{N} E^{\text{eff}}(\mathbf{h} = (h_\mu, \mathbf{h}_{\neg\mu})). \tag{3.55}$$

Each flip of a hidden unit corresponds to a move from one local minimum to another in the landscape $e_\mu(h_\mu|\mathbf{h}_{\neg\mu})$, see Fig. 3.11. Metropolis steps in the hidden space can speed up the dynamics: the free energy barrier for Metropolis-Hastings in the hidden space, $N\Delta e_{\text{MH}}$, where,

$$\Delta e_{\text{MH}} = -\frac{1}{N} \log \left[\frac{\int_0^\infty dh\, e^{-N e_\mu(h|\mathbf{h}_{\neg\mu})}}{\int_{-\infty}^0 dh\, e^{-N e_\mu(h|\mathbf{h}_{\neg\mu})}} \right], \tag{3.56}$$

is smaller than the free energy barrier $N\Delta f$ 'seen' by Alternating Gibbs Sampling.

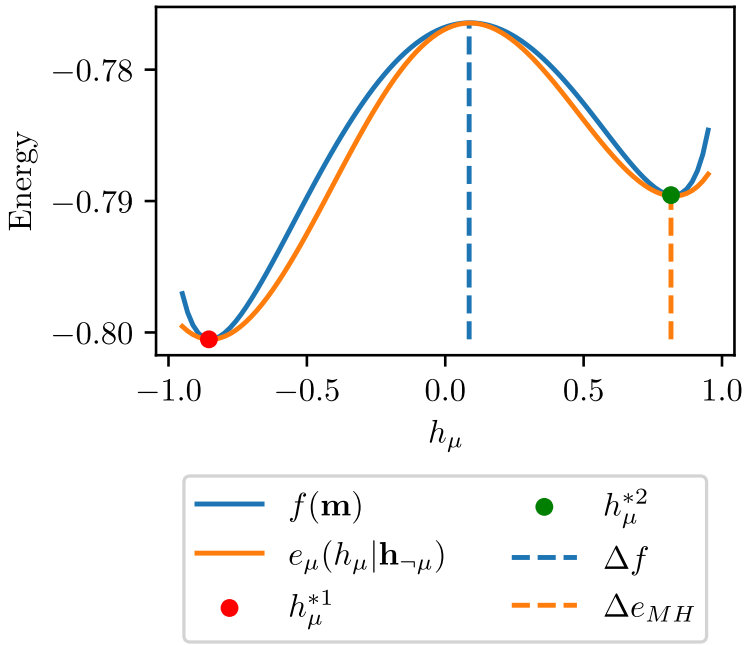


Figure 3.11: Barriers in a structured model with $M = 5$ hidden units, with $w = 1.2\sqrt{5}$, $\alpha_{\mu\nu} = 0.03$ for all pairs $\mu \neq \nu$. All hidden units are frozen except h_μ. For small overlap between the hidden units, the potential $e_\mu(h_\mu|\mathbf{h}_{\neg\mu})$ has two local minima for two different values of h_μ, h_μ^{1*} and h_μ^{2*}. By sampling back the visible layer $P(\mathbf{m}|\mathbf{h})$, we see that there are two local minima for $f(\mathbf{m})$. Flipping the hidden unit h_μ allows one to go from one local minimum to another. The free energy barrier in the hidden space with Metropolis-Hastings algorithm Δe_{MH} is smaller than the free energy barrier of the Alternating Gibbs Sampling Δf.

3.2.2 Application to BAS

We train RBM on BAS with a L_1 regularization to enforce the sparsity of the weights. Each hidden unit focuses on a given bar or a given stripe thanks to the regularization (Fig. 3.12(a)). The change $h_\mu \leftarrow -h_\mu$ leaves the energy $E^{\text{eff}}(\mathbf{h})$ unchanged: a bar or a

stripe can be present or not (Fig. 3.12(c)). We use a Gibbs sampling in the hidden space where one hidden unit is updated according to Algorithm 4. Our algorithm efficiently switches on/off these hidden units (Fig. 3.13(a)).

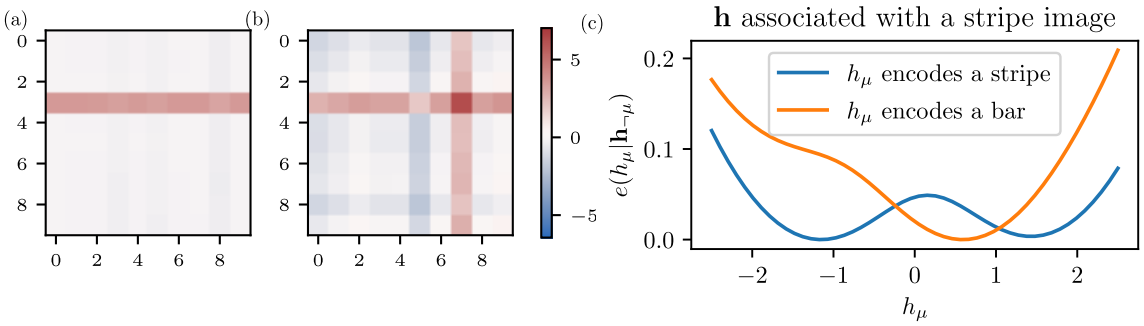


Figure 3.12: Example of weights learned by RBM on BAS, $L = 10$. (a) With L_1 regularization. Each hidden unit focuses on a bar or stripe. (b) Without L_1 regularization. Each hidden unit focuses on several bars and stripes. (c) Potential $e_\mu(h_\mu|\mathbf{h}_{\neg\mu})$ for $\mathbf{h}$ associated with a stripe image; the minimum of the energy is set to zero. Solid blue line: hidden unit h_μ encoding a stripe; the two minima coding from the on/off stripe have roughly the same energy. Solid orange line: hidden unit h_μ encoding a bar, the minimum encoding the on bar has an energy much higher than the one corresponding to the off bar.

Notice that, without regularization, each hidden unit would focus on several bars and stripes (Fig. 3.12(b)). In that case, allowing for steps in the hidden-unit space does not help, and our algorithm is inefficient (Fig. 3.13(b)).

Figure 3.13: Visible configurations obtained with Alternating Gibbs Sampling and Metropolis-Hastings algorithm in the hidden space, $L = 10$. 25 Gibbs steps between each frame. (a) With L_1 regularization. (b) Without L_1 regularization.

3.2.3 Application to the Hopfield model

We have seen in Section 3.1.2 that, for large enough weight amplitude w, the AGS dynamics is stuck in one Mattis state of the Hopfield model, *i.e.*, the magnetization $\mathbf{m}$ has only one component different from zero in the infinite size limit. The behavior of the hidden-unit configurations depends on the prescription of the weights, which may or may not be aligned with the states $\boldsymbol{\xi}^\mu$ (Eq. (2.16)).

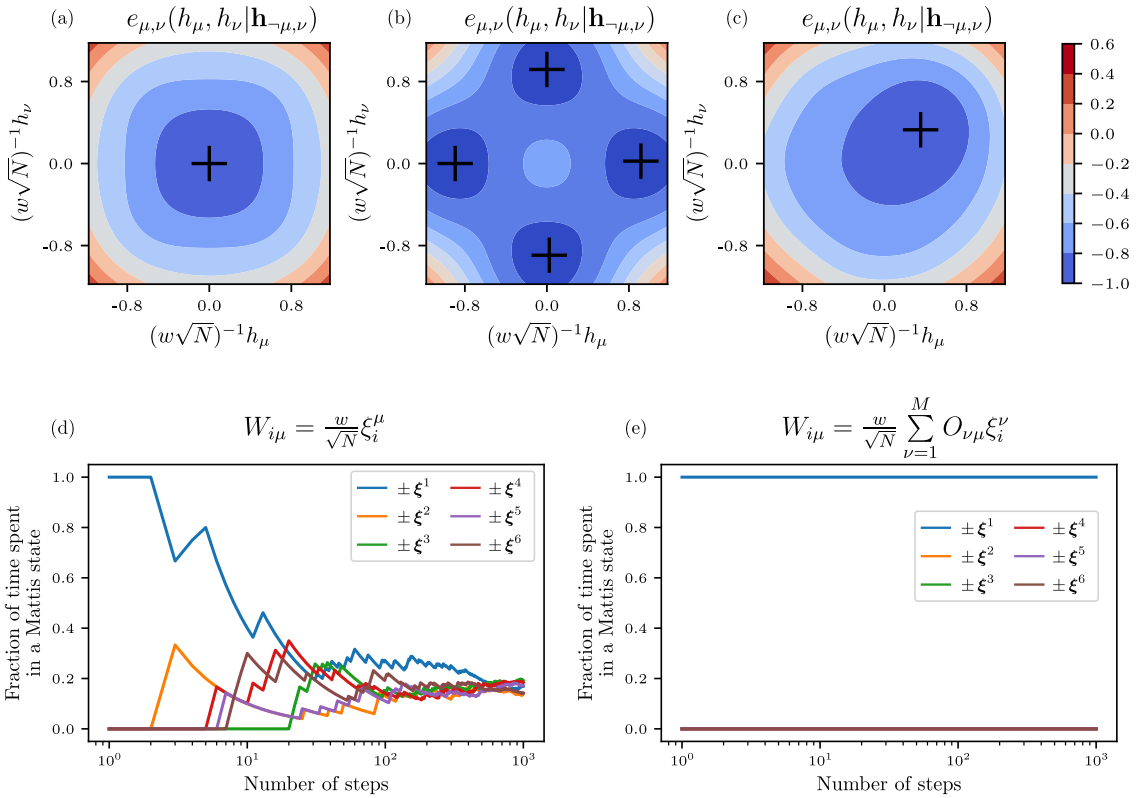


Figure 3.14: Hopfield model encoded by a RBM with $N = 128$, $M = 6$ and $w = 1.5$ and orthogonal $\boldsymbol{\xi}^\mu$. (a), (b) and (c) represent the landscape $e_{\mu,\nu}(h_\mu, h_\nu|\mathbf{h}_{\neg\mu,\nu})$, where the $M-2$ other components of $\mathbf{h}$ are fixed. Black dots represent minima of the landscape. (a) $W_{i\mu} = \frac{w}{\sqrt{N}}\xi_i^\mu$. Initial configuration is h_λ strongly magnetized and $h_\mu \sim h_\nu = \mathcal{O}(1)$. Minimum is reached for $h_\mu \sim h_\nu = \mathcal{O}(1)$. (b) $W_{i\mu} = \frac{w}{\sqrt{N}}\xi_i^\mu$. Initial configuration is h_μ strongly magnetized and $h_\nu = \mathcal{O}(1)$. Four minima exist corresponding to the four possible Mattis states. (c) Case $W_{i\mu} = \frac{w}{\sqrt{N}}\sum_{\nu=1}^{N} O_{\mu\nu}\xi_i^\nu$. There exist only one minimum. (d) and (e) $\mathbf{v}^t$ are generated with AGS with MH steps in the hidden space. The fraction of time spent in a Mattis state is measured through time. (d) $W_{i\mu} = \frac{w}{\sqrt{N}}\xi_i^\mu$: the visible configuration $\mathbf{v}^t$ eventually visits all Mattis states with equal probabilities. (e) $W_{i\mu} = \frac{w}{\sqrt{N}}\sum_{\nu=1}^{N} O_{\mu\nu}\xi_i^\nu$: the dynamics gets stuck in a given Mattis state.

3.2.3.1 Aligned weights

Let us first assume that the weights are aligned with the states, *i.e.*, that Eq. (2.18) holds. The effective energy over the hidden configurations reads

$$E^{\text{eff}}(\mathbf{h}) = \sum_\mu \frac{h_\mu^2}{2} - \sum_i \log 2\cosh\left(\frac{w}{\sqrt{N}}\sum_\mu \xi_i^\mu h_\mu\right). \tag{3.57}$$

Identifying $\frac{h_\mu}{w\sqrt{N}} = m_\mu$, the effective energy is equal to the free energy of the Hopfield model derived in Amit et al. (1985a) at inverse temperature w^2. The representations of the Mattis states are very simple in the hidden space of the RBM. In the presence of $\boldsymbol{\xi}^\mu$ on the visible layer, one hidden unit, say, $\mu = 1$, is strongly magnetized: $h_1 = \mathcal{O}(\sqrt{N})$. The

$M-1$ other hidden units are weakly activated: $h_\nu = \mathcal{O}(1)$ for $\nu \geq 2$. $E^{\text{eff}}(\mathbf{h})$ has $2M$ global minima corresponding to the $2M$ Mattis states.

a) Single unit potential

According to Eq. (3.57) the potential over the strongly magnetized hidden unit $\mu = 1$ reads, after rescaling $h_1 \to h_1/\sqrt{N}$,

$$e_1(h_1|\mathbf{h}_{\neg 1}) = \frac{h_1^2}{2} - \log 2\cosh(wh_1), \tag{3.58}$$

up to an additive constant. This potential has two global, opposed minima for $w^2 > 1$. The situation is similar to the CW model studied above: MH steps in the hidden-unit space allow for efficient sampling on the states $\boldsymbol{\xi}^1$ and $-\boldsymbol{\xi}^1$.

The potential on the other hidden units $\nu \neq 1$ is given by, up to an irrelevant additive constant and in the large-N limit, after rescaling $h_\nu \to h_\nu/\sqrt{N}$,

$$e_\nu(h_\nu|\mathbf{h}_{\neg\nu}) = \frac{h_\nu^2}{2} - (1-m_1^2)\left(\frac{1}{N}\sum_i \xi_i^1\xi_i^\nu\right)h_\nu. \tag{3.59}$$

Sampling this quadratic potential allows to better explore the Mattis state around $\boldsymbol{\xi}^1$, but it does not help to change state.

b) Two-unit potential

To speed up exploration of different states, we introduce the two-unit potentials,

$$e_{\mu,\nu}(h_\mu, h_\nu|\mathbf{h}_{\neg\mu,\nu}) = \frac{1}{N}E^{\text{eff}}(\mathbf{h} = (h_\mu, h_\nu, \mathbf{h}_{\neg\mu,\nu})), \tag{3.60}$$

where all but two hidden units are kept fixed. These potentials are plotted in Fig. 3.14. Two typical behaviors are encountered:

- μ, ν are both different from 1. The two-unit potential $e_{\mu,\nu}$ is simply the sum of the single-unit potentials e_μ and e_ν, see Eq. (3.59). Therefore, $e_{\mu,\nu}$ has only one global minimum (Fig. 3.14(a)). Changing h_μ or h_ν does not allow for moving outside the state condensed $\boldsymbol{\xi}^1$.
- $\mu = 1$ and $\nu \neq 1$. Contrary to the previous case, h_1 is now a free parameter. Therefore, by tuning h_1 and h_ν, four global minima of $e_{1,\nu}$ can be reached, corresponding to the cases where h_1 or h_ν are strongly magnetized (with positive or negative values), see Fig. 3.14(b). We can exploit this structure by introducing a block Gibbs sampling in the hidden space, where two hidden units are updated simultaneously, see Algorithm 5. The dynamics can now explore all the Mattis states very efficiently, see Fig. 3.14(d).

3.2.3.2 Rotated weights

As already mentioned in Section 3.1.2, the conditions in Eq. (2.16) do not uniquely define the weight matrix $\mathbf{W}$. The Hopfield model energy is invariant under any transformation $\mathbf{W} \to \mathbf{W} \times \mathbf{O}$, where $\mathbf{O}$ is an orthogonal matrix. After this orthogonal transformation, the hidden representation of a Mattis state is delocalized: each component of $\mathbf{h}$ is strongly magnetized (of the order of $\sqrt{N}$). Single or two-unit potentials have one global minimum (Fig. 3.14(c)). Therefore, Metropolis-steps in the hidden space do not speed up sampling (Fig. 3.14(e)) unless all M hidden units are simultaneously updated.

Numerical experiments with RBM trained by gradient ascent on data sampled from the Hopfield model generally converge to a solution, where the hidden representation of a

Algorithm 5: Alternating Gibbs Sampling with Metropolis-Hastings updates of two hidden units

Pick $\mathbf{v}^0$ in the training set;
for $t \in [\![0, T]\!]$ **do**
  $\mathbf{h}^{t+1} \sim P(\mathbf{h}|\mathbf{v}^t)$;
  π = random pairing of $[\![1, M]\!]$, defining $M/2$ pairs of elements ;
  for $i \in [\![1, M/2]\!]$) **do**
    $\mu, \nu = \pi(i)$;
    $h_\mu^{t+1}, h_\nu^{t+1} \sim P(h_\mu, h_\nu | \mathbf{h}_{\neg\mu,\nu}^{t+1})$;
  end
  $\mathbf{v}^{t+1} \sim P(\mathbf{v}|\mathbf{h}^{t+1})$;
end

Mattis state is delocalized (Fig. 3.15(a)) (Decelle et al., 2019). By adding the following penalty term in the log-likelihood, it is possible to ensure that only one hidden unit is strongly magnetized and encodes for a specific pattern $\boldsymbol{\xi}^\mu$, see Fig. 3.15(b):

$$\mathrm{LL}^{\mathrm{pen}} = -\frac{\lambda_{\mathrm{pen}}}{L} \sum_{\ell=1}^{L} \sum_{\mu \neq \nu} |f_\mu(\mathbf{v}^\ell) f_\nu(\mathbf{v}^\ell)|, \tag{3.61}$$

where $\{\mathbf{v}^\ell\}_{\ell=1\ldots L}$ are the L samples in the training set. This penalty favors solutions where only one hidden unit is strongly magnetized. Its intensity is set by the parameter λ_{pen}.

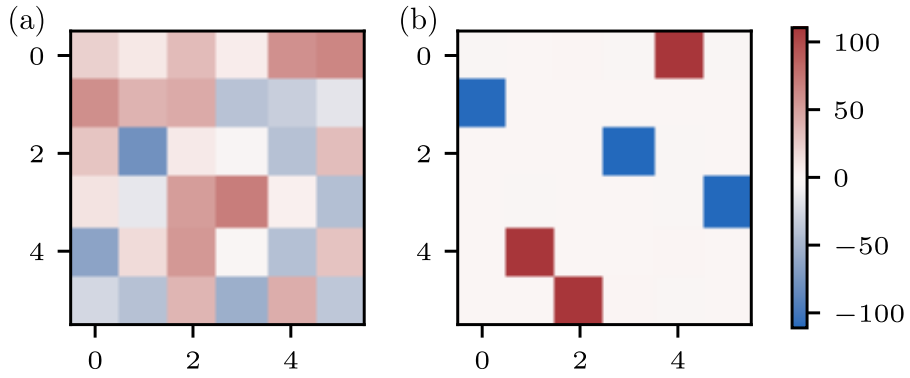


Figure 3.15: Matrix product between the weight matrix $\mathbf{W}^T$ (size $M \times N$) and the matrix of patterns $\boldsymbol{\xi}$ (size $N \times M$). $N = 128$ and $M = 6$. (a) Without regularization, $\lambda_{\mathrm{pen}} = 0$. Each pattern $\boldsymbol{\xi}^\mu$ has a delocalized representation in the hidden space. (b) With regularization, $\lambda_{\mathrm{pen}} = 0.001$. Each pattern $\boldsymbol{\xi}^\mu$ strongly magnetizes only one hidden unit.

3.3. Conclusion

This work presents a combination of analytical and numerical results on the dynamics defined by Alternating Gibbs Sampling of Restricted Boltzmann Machines and applied to several mean-field models. We have shown how this sampling procedure can find optimal transition paths between the local minima of the free energy landscape over the visible configurations. However, large free energy barriers, extensive in the system size, have to be crossed to go from one state to another. As a result, AGS is not more efficient than standard local Metropolis sampling of the effective energy of the visible configurations.

Notice that our analytical results were derived in a double large-size setting, where the asymptotics on the size N of the system was considered first, and the time T of transition paths was made large afterward. In practice, the probabilities that these transitions paths successfully interpolate between states are exponentially small in N, which implies, in turn, that transitions almost surely happen on times scales growing exponentially in N (and equal to the inverse probabilities). As shown in Fig. 3.4(a), the system spends most of this exponential time attempting to escape local minima of the free energy landscapes, while transitions between the minima are actually fast (but rare).

The inability of AGS to outperform local sampling procedures in mixing between states calls for some comments. First, it does not seem to be affected by the presence of structure in the free energy landscape. Both in the unstructured case, in which the minima of the free energy are uncorrelated (or related through global symmetries) and in the structured case, in which the minima exhibit a non-trivial organization (as observed for real data), large barriers are encountered. For structured distributions, however, the non-trivial organization of the minima leads to existence of optimal sampling paths, whose interpretation can be simpler in the hidden space of the RBM. Second, AGS, with Contrastive Divergence or Persistent Contrastive Divergence, remains an efficient training algorithm for RBM. These two procedures authorize initializations of the dynamics in different local minima close to the training data. Thus, even if AGS suffers from poor mixing between far away minima, the different minima close to the data may be well sampled. Third, AGS can be efficient when the different modes of data are connected through energy valleys. For example, AGS of RBM trained on all digits of MNIST can generate a transition between 0 and 1. However, these transitions go through different intermediate states, which are other digits. When training RBM on zeros and ones only, as done in this paper, intermediate states do not exist: the two modes are not connected by low energy funnels, and transitions are unlikely to occur. Last of all, RBM are supposed to encode meaningful (hidden) representations, coding for collective features in the data. It is tempting to see these features as modes of excitation that could be flipped at once, similarly to what cluster algorithms achieve for ferromagnetic models.

In the case of entangled representations, in which all (or a large number of) the hidden units are strongly magnetized (with different degrees of activation from one state to another), our combined AGS-MH procedure is inefficient, as flipping a small number of hidden units is unable to change the identity of the state, and determining new, adequate configurations of a large number of hidden units would be computationally prohibitive. This phenomenon was illustrated on the Hopfield model in the case of 'rotated' weights, compare Figs. 3.14(d) and (e). In much the same way, MH updates of a small subset of the hidden units of RBM trained on MNIST 0/1 or Lattice Proteins do not significantly enhance mixing performances. Hidden units capture features of the data, such as digit strokes for MNIST, which are correlated. Changing state demands to tune a large number of hidden units, see Fig. 3.16. In other words, the very existence of collective modes of hidden units prevents the success of our AGS-MH procedure, which is local in the hidden space. Another illustration of these collective modes in the hidden space is provided by RBM trained on BAS. Even if our algorithm is efficient to sample within a given class (bars or stripes), it cannot go efficiently from one class to another. To go from an image of bars to an image of stripes, the hidden units encoding the bars have to be silent, and the hidden units encoding the stripes have to be strongly magnetized. These define two collective modes of the hidden units, which AGS-MH cannot change. We stress that the inability of AGS to achieve rapid mixing is not limited to mean-field-like models. Even in the case of RBM tailored to encode finite-dimensional models with high-order ferromagnetic

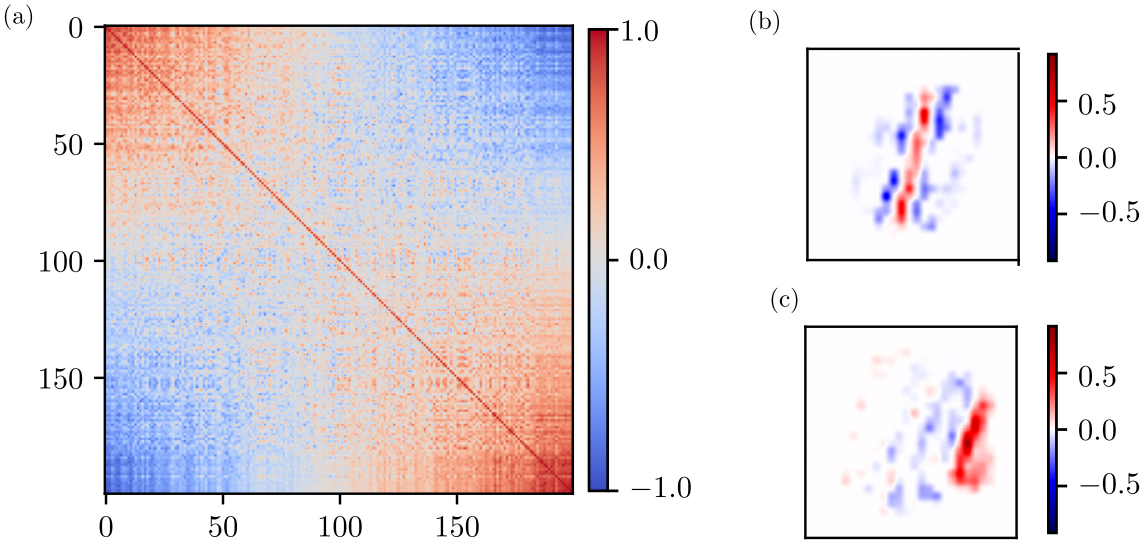


Figure 3.16: RBM trained on MNIST 0/1, with $M = 200$ hidden units. (a) Correlation matrix of the inputs received by the hidden units on the training data. Hidden units are sorted according to the components of the top eigenvector on this matrix. Two clusters emerge, corresponding to 0's and 1's: each digit is attached to roughly half the hidden units. (b) Example of weights $W_{i\mu}$ for a hidden unit μ associated with 1, corresponding to a stroke specific to 1. (c) Example of weights $W_{i\mu}$ for a hidden unit μ associated with 0, corresponding to a stroke specific to 0.

interactions, AGS suffers from poor mixing, and efficient sampling could only be obtained by combining with cluster algorithms such as the Swendsen-Wang procedure (Yoshioka et al., 2019). In Chapter 4, we show how stack of RBM, with ideas proposed in Bengio et al. (2013); Desjardins et al. (2014), can detect collective modes of hidden units and thus improve the sampling of the energy landscape.

4

Improving Sampling of Restricted Boltzmann Machines with Deep Tempering

This chapter discusses how the Deep Tempering algorithm developed by Desjardins et al. (2014) can improve the sampling of an energy landscape with RBM. This chapter is based on three observations.

First, numerical experiments on deep neural networks have shown their ability to progressively disentangle the principal modes in the data as their depths increase (Bengio et al., 2013). Representations of the data in the deep layers are more understandable and disentangled. Better disentanglement leads to better mixing between the different modes.

Second, interesting links between RBM and the renormalization group (RG) have been shown in a series of papers (Mehta and Schwab, 2014; Koch-Janusz and Ringel, 2018; Lenggenhager et al., 2020). In these papers, RBM are used to identify relevant degrees of freedom in some Hamiltonian, as for example Ising or dimmers models, and seems to perform a RG coarse graining-step.

Third, machine learning algorithms have been developed recently to detect relevant MC updates in condensed matter models (Liu et al., 2017; Xu et al., 2017; Huang and Wang, 2017; Nagai et al., 2017; Shen et al., 2018; Nagai et al., 2020a,b). Artificial neural networks are used to efficiently generate (with MC methods) low-energy configurations of approximate versions of target Hamiltonian.

Here, we will use the Deep Tempering algorithm differently from what was initially proposed by Desjardins et al. (2014). In their paper, Deep Tempering is a training algorithm for Deep Belief Networks, and in their numerical experiments, the number of hidden units of the different RBM are constant, and they increase the regularization of their RBM with the depth. DBN obtained with this algorithm are better than DBN trained greedily. In this chapter, the underlying mechanism is different. Deep Tempering is used as a sampling of a RBM (called bottom RBM in the following), once it has been trained on data. The deeper representations are not meant to disentangle the underlying factors of variation of the data, but to compress the bottom RBM's hidden representations. Each mode of the data has a few distinct representations in the hidden space of the top RBM. By reducing the number of hidden units with the depth, the hidden representations are progressively compressed, and the sampling between distinct modes of the data improves.

4.1. Deep Tempering algorithm

As explained in Section 1.3.4, Deep Tempering was first introduced as a new algorithm to train Deep Belief Networks (DBN) (Hinton and Salakhutdinov, 2006; Salakhutdinov and Murray, 2008). DBN are stacks of N RBM (Fig 1.2(a)). The number of hidden units of the n^{th} RBM is equal to the number of visible units of the $(n+1)^{\text{th}}$ RBM.

In our case, we will not use Deep Tempering to train a DBN, but as an algorithm to sample the energy landscape of our RBM of interest (called bottom RBM). To do this, once our bottom RBM is trained on the data $\{\mathbf{v}^k\}_{k=1...K}$, we train a second RBM on the hidden representations of the bottom RBM $\{\mathbf{h}_1^k\}_{k=1...K}$ drawn from $P_1\left(\mathbf{h}|\mathbf{v}^k\right)$. If necessary, we train a third RBM on the hidden representations of the second RBM, and so on. This learning algorithm is similar to the greedy training of DBN (Hinton et al., 2006; Bengio et al., 2007). Once all the RBM are trained, we can use the Deep Tempering algorithm to sample the visible landscape of the bottom RBM (Algorithm 6). We denote as $\alpha_n = \frac{M_n}{N_n}$ the aspect ratio of the n^{th} RBM of the stack. Each RBM of the stack has its own visible landscape $E_n^v(\mathbf{v})$ (respectively hidden landscape $E_n^h(\mathbf{h})$) associated with the Boltzmann distribution $P_n^v(\mathbf{v})$ (respectively $P_n^h(\mathbf{h})$). Deep Tempering allows adjacent RBM in the stack to exchange their configuration with the following acceptance ratio (the detailed balance is satisfied, see Appendix C.1)

$$\begin{aligned} & A_n\left(\{\mathbf{h}_n = \mathbf{h}_n^t, \mathbf{v}_{n+1} = \mathbf{v}_{n+1}^t\} \to \{\mathbf{h}_n = \mathbf{v}_{n+1}^t, \mathbf{v}_{n+1} = \mathbf{h}_n^t\}\right) \\ = \quad & \min\left(1, \frac{P_{n+1}^v(\mathbf{h}_n^t)P_n^h(\mathbf{v}_{n+1}^t)}{P_{n+1}^v(\mathbf{v}_{n+1}^t)P_n^h(\mathbf{h}_n^t)}\right) = \min\left(1, \frac{\exp\left(-E_{n+1}^v(\mathbf{h}_n^t) - E_n^h(\mathbf{v}_{n+1}^t)\right)}{\exp\left(-E_{n+1}^v(\mathbf{v}_{n+1}^t) - E_n^h(\mathbf{h}_n^t)\right)}\right) \end{aligned} \tag{4.1}$$

This algorithm will be particularly efficient in the case where the data are composed of several distant clusters, and where the hidden representations learned by the RBM bottom are also composed of several distant clusters, as for example in the case of MNIST 0/1 or the Lattice Protein mentioned in the Chapter 3.

In our cases of interest, the bottom RBM learns a complex energy landscape that allows for high-quality data generation, but whose large energy barriers between distant modes make Alternating Gibbs Sampling ineffective for switching between modes.

Deep Tempering, seen as an algorithm for sampling the energy landscape $E_1^v(\mathbf{v})$ of the RBM bottom, is therefore there to help sample between these distant modes.

In order to use Deep Tempering effectively, there are two criteria to be met. It is necessary that the RBM on top of the stack can switch from one mode to another faster than the bottom RBM, *i.e.* that the energy barriers between two modes in $E_n^v(\mathbf{v})$ are lower than in $E_1^v(\mathbf{v})$. At the same time, the acceptance ratio must be high, to ensure that the RBM exchange their configurations well, *i.e.* $E_{n+1}^v(\mathbf{v})$ must be a good approximation of $E_n^h(\mathbf{h})$. These criteria are similar to those necessary for the proper application of Parallel Tempering, where temperatures must be set to ensure better exchange between modes, while ensuring that the acceptance ratio between the different temperatures remains high.

Here, instead of choosing temperatures, we need to make sure that $E_{n+1}^v(\mathbf{v})$ is a good approximation of $E_n^h(\mathbf{h})$, to have a high acceptance ratio, while $E_{n+1}^v(\mathbf{v})$ having smaller barriers between modes than $E_n^h(\mathbf{h})$ to mix better between modes. So there is a trade-off. If $E_n^h(\mathbf{h}) = E_{n+1}^v(\mathbf{v})$, the acceptance ratio is high, but the barriers are identical: there is no interest in Deep Tempering. If $E_{n+1}^v(\mathbf{v})$ is a poor approximation of $E_n^h(\mathbf{h})$, in the sense that a class of data (*e.g.* the zeros or the ones of MNIST), which corresponds to a complex set of minima of $E_n^h(\mathbf{h})$, corresponds to only one minimum of $E_{n+1}^v(\mathbf{v})$, then the barriers will be smaller between two classes of data in $E_{n+1}^v(\mathbf{v})$ than in $E_n^h(\mathbf{h})$. However, it will be impossible to exchange the configurations generated by the n^{th} RBM and $(n+1)^{\text{th}}$ RBM, because the visible configurations $\mathbf{v}_{n+1}$ generated by the $(n+1)^{\text{th}}$ RBM will not correspond to minima of $E_n^h(\mathbf{h})$. It is therefore necessary to keep a balance, to decrease the size of the barriers while having a high acceptance ratio. To reduce the barriers, $E_{n+1}^v(\mathbf{v})$ must lose some details of $E_n^h(\mathbf{h})$, *i.e.* several minima of $E_n^h(\mathbf{h})$ are grouped into only one minimum in

$E^v_{n+1}(\mathbf{v})$. In terms of representations, this means that several vectors in $\{\mathbf{h}^k_n\}_{k=1...K}$ have the same representation $\mathbf{h}_{n+1}$. We speak then of compression of representations.

Here, we will put ourselves in a particular configuration where $\forall n > 1, \alpha_n < 1$, and we add during the training a regularization of the RBM weight matrix to control its norm, in order to compress both the representations and to have a smoother $E^v_{n+1}(\mathbf{v})$ landscape than the $E^h_n(\mathbf{h})$ landscape. By compression of representations, we mean that the number of unique vectors in $\{\mathbf{h}^k_{n+1}\}_{k=1...K}$ is smaller than the number of unique vectors in $\{\mathbf{h}^k_n\}_{k=1...K}$, *i.e.* several vectors in $\{\mathbf{h}^k_n\}_{k=1...K}$ have the same representation $\mathbf{h}_{n+1}$. And we consider that $E^v_{n+1}(\mathbf{v})$ is smoother than $E^h_n(\mathbf{h})$ if the typical barrier size $E^v_{n+1}(\mathbf{v})$ is smaller than in $E^h_n(\mathbf{h})$.

Algorithm 6: Deep Tempering

```
isodd = True ;
for t ∈ [[0,T]] do
    if isodd then
        n = 1 ;
        while n < N do
            Swap h_n^t and v_{n+1}^t with probability
              A_n({h_n = h_n^t, v_{n+1} = v_{n+1}^t} → {h_n = v_{n+1}^t, v_{n+1} = h_n^t}) ;
            v_n^t ~ P_n(v|h_n^t) ;
            n = n + 2
        end
    else
        n = 2 ;
        while n < N do
            Swap h_n^t and v_{n+1}^t with probability
              A_n({h_n = h_n^t, v_{n+1} = v_{n+1}^t} → {h_n = v_{n+1}^t, v_{n+1} = h_n^t}) ;
            v_n^t ~ P_n(v|h_n^t) ;
            n = n + 2
        end
    end
    isodd ← not isodd ;
    for n ∈ [[1,N]] do
        h_n^{t+1} ~ P_n(h|v_n^t) ;
        v_n^{t+1} ~ P_n(v|h_n^{t+1}) ;
    end
end
```

4.2. Numerical experiments on real data

We illustrate this idea on two different datasets: the zeros and the ones of the MNIST dataset and artificial proteins sampled from Lattice Protein models. We have shown in Chapter 3 that Alternating Gibbs Sampling is efficient to sample within a class and generate high-quality data, but it is improbable to go from one mode to another (Figs. (3.8) and (3.9)).

In these two examples, data are grouped into two distinct classes. The projection of the zeros and the ones of the MNIST dataset onto its first and second principal components reveals two clusters (Fig. 4.1(a)). For Lattice Protein models, the sequences came from two distinct structures, projections onto the two first components of the PCA distinguish

the two families, although the two clusters overlap (Fig. 4.2(a)). Hidden representations of the data are perfectly separated with PCA (Fig. 4.2(b)). In that case, the bottom RBM has learned useful representations of the data.

Numerically, we remark that the different RBM progressively compress the representations of the data. The number of distinct hidden representations decreases with the depth, and AGS is more efficient for the top RBM. In these two examples, Deep Tempering is a more efficient sampling algorithm than AGS to sample the energy landscape of the bottom RBM.

The influential parameters of Deep Tempering, be it the number of RBM in the stack, the number of hidden units as well as the intensity of the regularization were chosen empirically, in order to guarantee a good acceptance ratio while reducing energies barriers.

4.2.1 Deep Tempering for MNIST 0/1

The stack comprises four RBM, with respectively 200, 100, 25, and 10 hidden units. Deep Tempering improves the mixing between the modes (Fig. 4.1(c-f)).

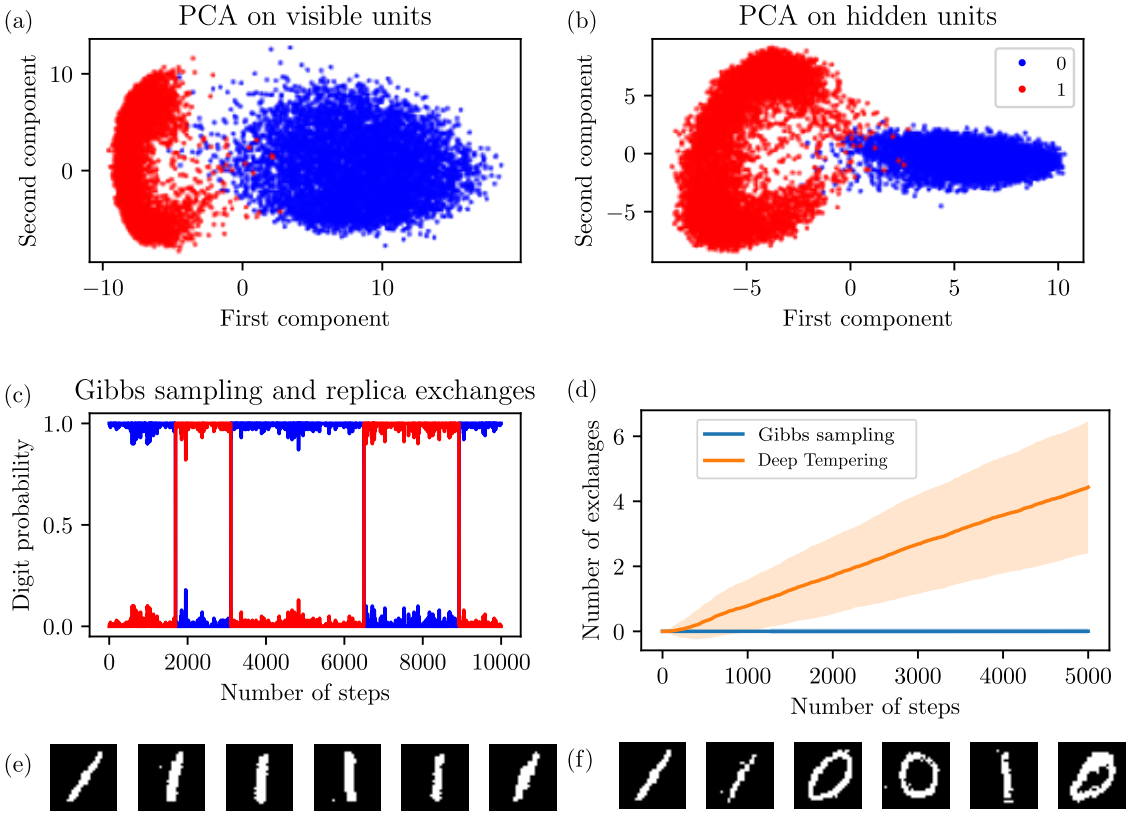


Figure 4.1: (a) Projection onto the two first components of the PCA for the data. (b) Projection onto the two first components of the PCA for the hidden representations of the data. (c) A random forest classifier is trained on 0/1 MNIST. The classifier predicts the digit's probability of the data sampled by the RBM. Deep Tempering algorithm generates high-quality digits and mixes well between the two classes. (d) Mean number of swaps between the two digits classes for the two dynamics. The initial configurations of the dynamics are random digits of 0/1 MNIST. The mean is computed with 2000 random initial configurations. (e-f) Example of sampled digits. Digits are displayed every 750 Gibbs steps. The two dynamics have the same initial configuration. (e) Alternating Gibbs Sampling. (f) Deep Tempering.

4.2.2 Lattice Protein

The stack is made of three RBM, with respectively 800, 50 and 25 hidden units. Deep Tempering improves the mixing between the modes (Fig. 4.2(c-d)).

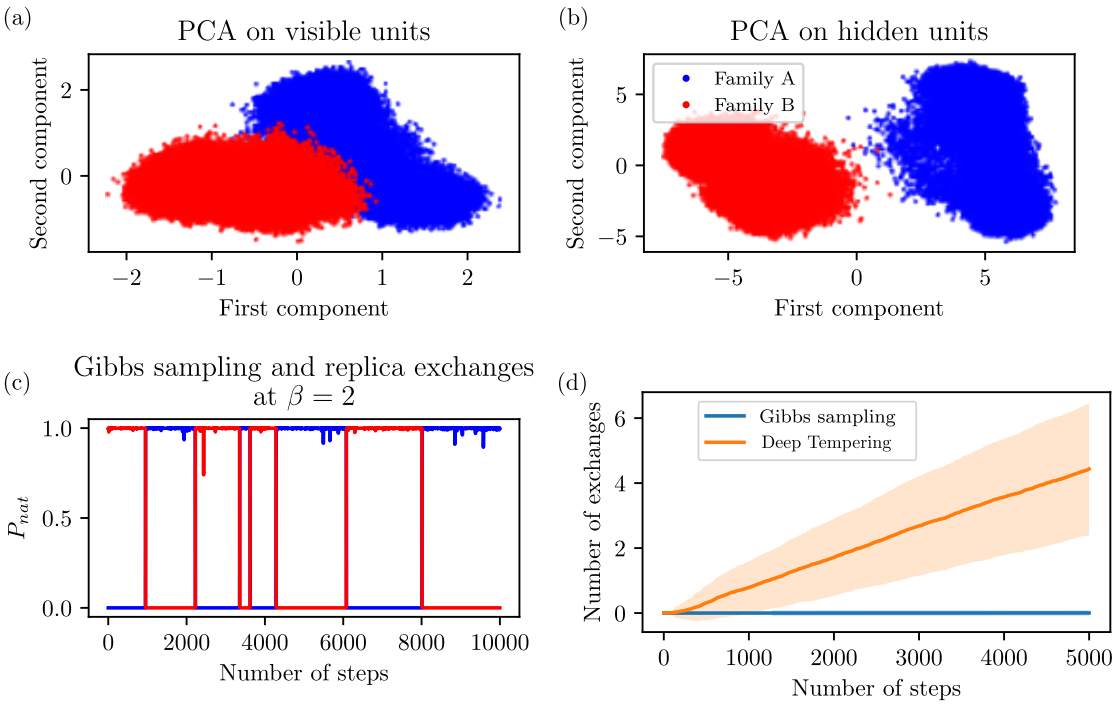


Figure 4.2: (a) Projection onto the two first components of the PCA for the data. (b) Projection onto the two first components of the PCA for the hidden representations of the data. (c) $P_{\text{nat}}(\mathbf{v}|S)$ of sampled proteins with DBN for the two families S_A and S_B. Deep Tempering algorithm generates high-quality proteins and mixes well between the two families. (d) Mean number of swaps between the two families for the two dynamics. The initial configurations of the dynamics are random proteins of the training set. The mean is computed with 500 random initial configurations.

4.3. What are the parameters influencing the compression of the representations?

In this section, we will numerically investigate the influence of the number of stacked RBM and the number of hidden units on the compression of representations. To do this, we introduce the $\{\boldsymbol{\xi}_1^k\}_{k=1\ldots K}$ patterns that represent the data. In order to have a realistic structure for the patterns, the visible patterns fall into a hierarchical tree: clusters of patterns are divided into subclusters, which can also be divided into subclusters ... This specific structure of patterns, called ultra-metricity, appears in the Sherrington-Kirkpatrick model of spin-glass (Sherrington and Kirkpatrick, 1975; Mézard et al., 1984; Rammal et al., 1986). Fig. 4.3(a) depicts the correlation matrix of the patterns we have used, with two main clusters divided into three subclusters. We train several stacks of RBM on these artificial data. For each RBM, we use the same number of iterations and the same learning rate during training. Depending on the depth of the stacks and the number of hidden units, the hidden representations exhibit different behaviors (Fig. 4.3). For the first RBM, the correlation matrix of $\mathbf{h}_1^k$ mimics the structure of the correlation matrix of $\boldsymbol{\xi}_1^k$, but the level of details depends on its number of hidden units M_1. If M_1 is small, each pattern in

a given cluster has an identical hidden representation (Fig. 4.3(b)). Then, by increasing the number of hidden units M_1, each pattern in a given subcluster has an identical hidden representation (Fig. 4.3(c)). And finally, for an even larger number of hidden units M_1, each pattern has its own hidden representation (Fig. 4.3(d)).

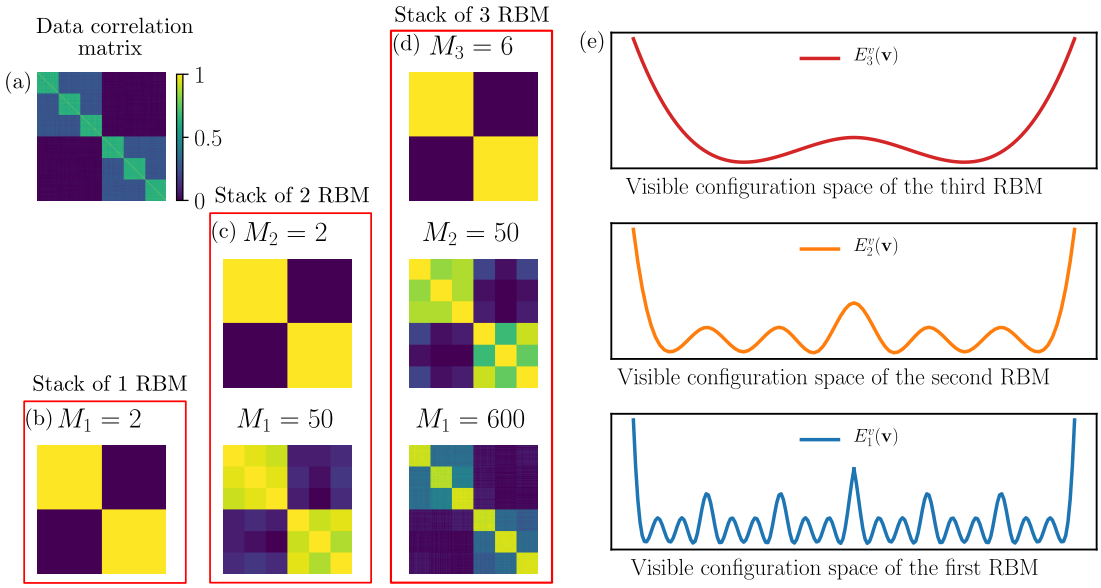


Figure 4.3: The color map is the same for the different correlation matrices (from dark blue to yellow). (a) Correlation matrix of 600 visible patterns $\boldsymbol{\xi}_1^k$, $N = 1000$. The patterns are divided into two clusters. Each cluster also has three subclusters. (b) Correlation matrix of the hidden patterns $\mathbf{h}_1^k$. (c) From the bottom to the top: correlation matrices of the hidden patterns $\mathbf{h}_1^k$ and $\mathbf{h}_2^k$. (d) From the bottom to the top: correlation matrices of the hidden patterns $\mathbf{h}_1^k$, $\mathbf{h}_2^k$ and $\mathbf{h}_3^k$. (e) Schematic representation of the landscapes $E_1^v(\mathbf{v})$, $E_2^v(\mathbf{v})$ and $E_3^v(\mathbf{v})$ learned by the different RBM represented in the panel (d). The details of the landscape are progressively smoothing out, but at the same time the free energy barriers between the different modes are decreasing.

Therefore, by tuning the number of hidden units, we can control the magnitude of the compression of the representations. It is important to note that we do not want to impose any constraints on the bottom RBM, as its goal is to learn an energy landscape capable of reproducing the data as closely as possible (in the sense of maximizing the log-likelihood), so that it can then be capable of generating high-quality data. However, we are free to impose constraints on the number of hidden units as well as on the magnitude of the weights for the other RBM in the stack.

If the compression is too important, *i.e.*, if the number of hidden units is too small, the RBM learns a poor representation of the data. Consequently, $E_{n+1}^v(\mathbf{v})$ is a crude approximation of $E_n^h(\mathbf{h})$, and therefore the replica exchange rate is low and the dynamics of the RBM are decoupled: in that case, Deep Tempering would be as efficient as AGS. Fig. 4.3(d) shows an example where each RBM of the stack learns a different level of representation of the hierarchical tree: the bottom RBM learns one representation per pattern, the second RBM one representation per subcluster and the top RBM one representation per cluster. Fig. 4.3(e) exhibits schematic representations of the different landscapes learned by the RBM in the stack. $E_1^v(\mathbf{v})$ has local minimum per patterns, $E_2^v(\mathbf{v})$ per subcluster and $E_3^v(\mathbf{v})$ per cluster: $E_{n+1}^v(\mathbf{v})$ has to be a smooth approximation

of $E_n^h(\mathbf{h})$ in order to have lower barriers while remaining a good approximation to keep the acceptance ratio high between the different RBM.

4.4. Compression of representations with Restricted Boltzmann Machines

In this section, we will study the compression of representations. To do so, we will first define our analytical framework to study this effect, then study numerically the structure taken by the weight matrix $\mathbf{W}$, and finally show the compression analytically in the particular case where $K = 2$.

4.4.1 Analytical framework

To study in more detail, in a numerical and analytical way, the effect of compression of the representations as well as the smoothing of the energy landscapes, we decided to place ourselves in the following configuration.

The number of hidden units M, as well as the number of visible units N, will be taken in the thermodynamic limit, with a finite aspect ratio $\frac{M}{N} \underset{M,N\to\infty}{\longrightarrow} \alpha$. The number of patterns K will be considered finite. This last assumption may seem risky, because on real data, the number of training examples may be arbitrarily large. Nevertheless, for example in the case of MNIST 0/1, most of the training examples can be seen as variations of a small number of prototypes, which would represent typical 0's and 1's of the training data. What we call K here, is the number of these prototypes. This hypothesis has already been formulated, as for example in the work of Decelle et al. (2017, 2018), where the weight matrix $\mathbf{W}$ has K (finite) dominant isolated singular values, disturbed by noise. Therefore, in our study, we restrict ourselves to K. Nevertheless, it will be interesting later to study the K infinite case. which has already been partly studied by Leonelli et al. (2021) for a different purpose.

4.4.2 Numerical experiments

In this section, we study numerically the form of the matrix $\mathbf{W}$. To do so, we train RBM with Contrastive Divergence on a set of K mutually orthogonal vectors $\{\boldsymbol{\xi}^k\}_{k=1\dots K}$, in the limit $K \ll M, N$ and $\frac{M}{N} \underset{M,N\to\infty}{\longrightarrow} \alpha < 1$.

The weight matrix $\mathbf{W}^0$ is initialized with small random values chosen from a zero-mean Gaussian with a standard deviation σ, typically $\sigma = \mathcal{O}(N^{-1})$.

At the end of the training, by computing the singular value decomposition of the weight matrix $\mathbf{W}$, we remark that K singular values dominate (Figs. 4.4(a) and (b)). By adding a L_2 regularization, the norms of the $M - K$ other singular values decrease during training (Fig. 4.4(b)).

The matrix can be written as $\mathbf{W} = \mathbf{W}^{\parallel} + \mathbf{W}^{\perp}$, where $\mathbf{W}^{\parallel} = \sum_{k=1}^{K} \tilde{w}_k \tilde{\mathbf{v}}^k \cdot \tilde{\mathbf{h}}^{k^T}$ corresponds to the top K modes, and $\mathbf{W}^{\perp}$ to the $M - K$ other modes ($\mathbf{W}^{\parallel} \cdot \mathbf{W}{\perp}^T = 0$). With L_2 regularization, the norm of $\mathbf{W}^{\perp}$ decreases over time, and therefore we will neglect it in our theoretical computations.

For the hidden fields, c_μ is an increasing odd function of $\sum_{k=1}^{K} \hat{\xi}_\mu^k$ (Fig. 4.4(d)). The hidden fields are small compared to the inputs of the hidden units $I_\mu(\boldsymbol{\xi}^k)$ but enough to break the symmetry between $E(\mathbf{v})$ and $E(-\mathbf{v})$.

We remark that $\text{span}(\{\tilde{\mathbf{v}}^k\}_{k=1\dots K}) = \text{span}(\{\boldsymbol{\xi}^k\}_{k=1\dots K})$, where $\text{span}(S)$ denotes the linear span of the set of vectors S. Therefore, we can write $\{\tilde{\mathbf{v}}^k\}_{k=1\dots K}$ in the basis formed by $\{\boldsymbol{\xi}^k\}_{k=1\dots K}$, and rewrite the $\tilde{w}_k$ in this basis: each $\boldsymbol{\xi}^k$ is associated with a w_k. The w_k's

are of the same order, but different. It is because the $\boldsymbol{\xi}^k$'s are not strictly orthogonal in practice. If there are mutually orthogonal, the w_k's are equal.

The final hidden representations of the patterns depend strongly on the initial weight matrix $\mathbf{W}^0$. We see a strong correlation between sign $\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^k\right)$ and sign $\left(\mathbf{W}^T \cdot \boldsymbol{\xi}^k\right)$. As $\mathbf{W}^0$ is a random matrix, and the $\boldsymbol{\xi}^k$'s are orthogonal, the initial representations of the data are nearly orthogonal. At the end of the training, the data representations are still nearly orthogonal (Fig. 4.4(c)).

Similar results are obtained when training an RBM on non-orthogonal patterns: the weight matrix $\mathbf{W}$ has K singular values emerging from the bulk. However, contrary to the orthogonal case, the w_k are no longer equal. Furthermore, at the end of the training, the correlation matrix of the hidden representations is different from the correlation matrix of the hidden representations at the beginning of the training. In Section 4.4.4, we will find the optimal correlation in the case $K = 2$.

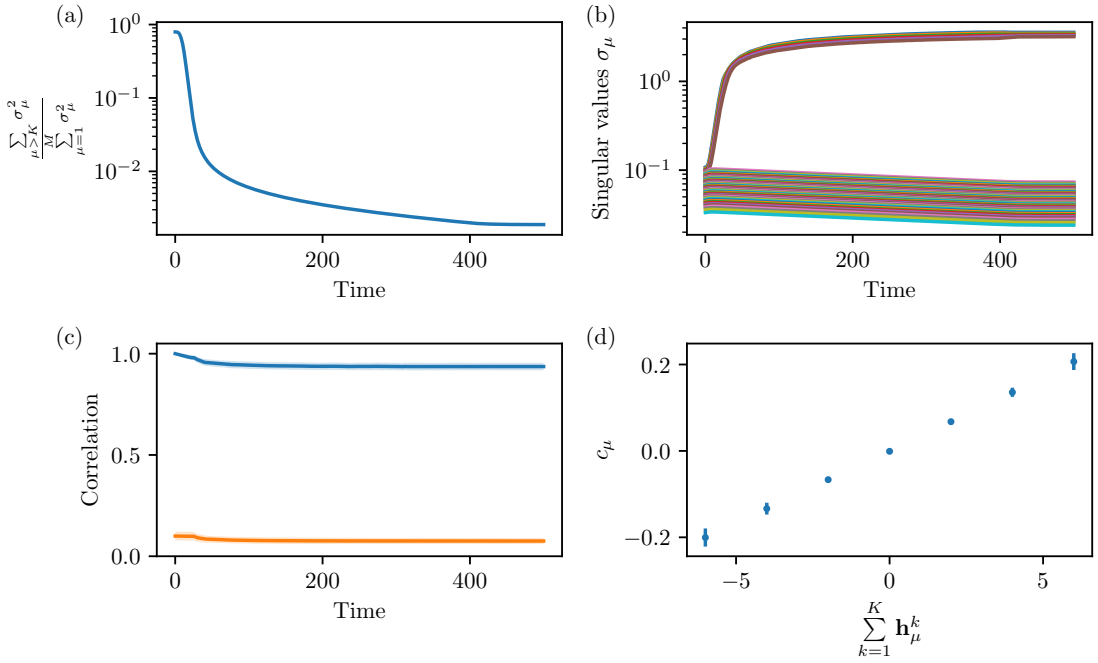


Figure 4.4: Numerical experiments with $N = 200$, $M = 60$, $K = 6$. Quantities of interests are average over 50 different initialization of the $\boldsymbol{\xi}^k$'s and $\mathbf{W}^0$. (a) Evolution of $\left(\sum_{\mu>K} \sigma_\mu^2\right)\left(\sum_{\mu=1}^{M} \sigma_\mu^2\right)^{-1}$ through time. The M singular values σ_μ are ranking in descending order. (b) Evolution of the singular values through time. K singular values emerge from the bulk. The $(M-K)$ singular values decrease over time due to the L_2 regularization. (c) Blue line: evolution of $\frac{1}{M}\text{sign}\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^k\right)^T \cdot \text{sign}\left(\mathbf{W}^T \boldsymbol{\xi}^k\right)$. The representations of the patterns are closed to their initial representations. Orange line: evolution of $\frac{1}{M}|\text{sign}\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^k\right)^T \cdot \text{sign}\left(\mathbf{W}^T \cdot \boldsymbol{\xi}^{k'}\right)|$ for $k \neq k'$. The representations are nearly orthogonal. (d) Mean value of c_μ as a function of $\sum_{k=1}^{K} \hat{\xi}_\mu^k$.

4.4.3 Hypothesis on the structure of the weight matrix

In the following, we consider that the weight matrix $\mathbf{W}$ can be put in the following form

$$W_{i\mu} = \frac{1}{N}\sum_{k=1}^{K} w_k \xi_i^k \hat{\xi}_\mu^k, \tag{4.2}$$

where $\{\boldsymbol{\xi}^k\}_{k=1...K}$ are the K patterns. $\{\hat{\boldsymbol{\xi}}^k\}_{k=1...K}$ are K vectors of $\{-1,1\}^M$ and $w_k = \mathcal{O}(1)$. This particular form of the weight matrix $\mathbf{W}$ is justified by numerical experiments (Section 4.4.2). This hypothesis is similar to the one mentioned in Decelle et al. (2017, 2018) although being more simplistic, because we do not consider any iid noise perturbation to $W_{i\mu}$.

The correlation matrix of visible patterns is fixed by the data $\{\boldsymbol{\xi}^k\}_{k=1...K}$. The weight matrix $\mathbf{W}$ is similar to the one learned from the Hebb's rule, except that in our case we have as additional degrees of freedom, the $\{\hat{\boldsymbol{\xi}}^k\}_{k=1...K}$. A main question is what are the optimal representations, depending on the aspect ratio α and $\{w_k\}_{k=1...K}$. In the Section 4.4.4, we will study this in detail for the $K = 2$ case.

Concerning the fields on the visible layer g_i and the hidden layer c_μ, we will consider that $g_i = 0$. The c_μ fields will be non-zero, enough to break the symmetry between $E(\mathbf{v})$ and $E(-\mathbf{v})$, but are small compared to the inputs of the hidden units ($|c_\mu| \ll |I_\mu(\boldsymbol{\xi}^k)|$). Therefore, we neglect the fields c_μ in the theoretical computations of the rest of the chapter.

4.4.4 Optimal hidden representations of two correlated patterns

4.4.4.1 Theoretical results

In this section, we describe the case of two patterns, $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ with an arbitrary correlation x ($\boldsymbol{\xi}^{1^T} \cdot \boldsymbol{\xi}^2 = N\,x$). We compute the optimal correlation y between the two hidden representations $\hat{\boldsymbol{\xi}}^1$ and $\hat{\boldsymbol{\xi}}^2$ ($\hat{\boldsymbol{\xi}}^{1^T} \cdot \hat{\boldsymbol{\xi}}^2 = M\,y$) for a RBM with N visible and $M = \alpha N$ hidden units. In that case, the weight matrix (Eq. 4.2) reads

$$W_{i\mu} = \frac{w}{N}\sum_{k=1,2} \xi_i^k \hat{\xi}_\mu^k, \tag{4.3}$$

We will show that, depending on α and w, there exists a regime where, for $x > x_c$, the representations are compressed ($y = 1$).

First, the log-likelihood of the two patterns reads

$$\mathrm{LL} = \frac{1}{2}\left(\log P(\boldsymbol{\xi}^1) + \log P(\boldsymbol{\xi}^2)\right) = -\frac{1}{2}\left(E^{\mathrm{eff}}(\boldsymbol{\xi}^1) + E^{\mathrm{eff}}(\boldsymbol{\xi}^2) + 2\log Z\right). \tag{4.4}$$

In the thermodynamic limit $N \to \infty$, the partition function can be computed by means of saddle point (see the general expression for any finite K in Eq. 4.12). By defining $\Delta_\pm = q^1 \pm q^2$, we have for the log-likelihood

$$\begin{aligned}\frac{1}{N}\mathrm{LL} &= \frac{1}{2}\Bigg(\alpha(1+y)\Big(\log\left(\cosh\left(w(1+x)\right)\right)-\log\left(\cosh(w\,\Delta_+)\right)\Big) \\ &+ \alpha(1-y)\Big(\log\left(\cosh\left(w(1-x)\right)\right)-\log\left(\cosh(w\,\Delta_-)\right)\Big)\Bigg) \\ &+ \frac{1}{4}\Bigg(\Delta_+\log\left(\frac{1+x+\Delta_+}{1+x-\Delta_+}\right)+(1+x)\log\left(1-\frac{\Delta_+^2}{(1+x)^2}\right) \\ &+ \Delta_-\log\left(\frac{1-x+\Delta_-}{1-x-\Delta_-}\right)+(1-x)\log\left(1-\frac{\Delta_-^2}{(1-x)^2}\right)\Bigg)-\log 2,\end{aligned} \tag{4.5}$$

with

$$\Delta_\pm = (1\pm x)\,\tanh\left(\alpha\, w(1\pm y)\tanh(w\,\Delta_\pm)\right). \tag{4.6}$$

The optimal correlation of the hidden representations $y^*(w,x,\alpha)$ is defined as $\frac{1}{N}\frac{\partial \mathrm{LL}}{\partial y}\Big|_{y=y^*}=0$ and depends on x,α and w. Without loss of generality, we assume that $x>0$ (by symmetry, the following results can be extended to $x<0$).

RBM exhibits two typical behaviors depending on the value of w. As long as $\alpha\, w^2(1\pm x)(1\pm y)<1, \Delta_\pm=0$. In this regime, $\forall x<1, y^*(w,x,\alpha)=1$. RBM has the same hidden representation for the two patterns.

For large $\alpha\, w$, *i.e.* when $\Delta_+=(1+\alpha)$, Eq. (4.5) can be simplified as follows:

$$\frac{1}{N}\mathrm{LL} \underset{\alpha\, w\gg 1}{\sim} -\tfrac{1}{2}\exp\left(-2\alpha\, w(1-y)\right)\left(\alpha\, w(1-y)(1-x)\exp\left(-2\alpha\, w(1-x)\right)+(1-x)\right) \tag{4.7}$$

In that case, $\forall x<1, y^*(w,x,\alpha)=0$: the hidden representations of the patterns are orthogonal. By computing the derivative of Eq. (4.4) with respect to y, we remark that as long as $2\alpha\, w(1-y)\gg 1$, y is a solution with a log-likelihood near 0: there exists a broad range of y with a log-likelihood near the optimum.

The log-likelihood (Eq. (4.5)) can be optimized numerically with respect to y. $y^*(w,x,\alpha)$ is a decreasing function of α and w, and an increasing function of x (Figs. 4.5(a-c)). In Fig. 4.5(c) we remark there exists some $x<1$ with $y=1$. The threshold x_c such that $\forall x>x_c, y=1$ is an increasing function of α and w (Appendix C.2.1 for the computation of x_c). If $x>x_c$, the two patterns have the same hidden representation: the representations are compressed. We remark also that if α and w are big enough, RBM tend to orthogonalize the hidden representations even if $x>0$.

The energy landscape $E^{\mathrm{eff}}(\mathbf{v})$ learned by the RBM depends on y:

- For $y<1$, $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ are global minima of $E^{\mathrm{eff}}(\mathbf{v})$, separated by energy barriers.
- For $y=1$, the two patterns are still global minima of $E^{\mathrm{eff}}(\mathbf{v})$ but are included in a large basin of low-energy: all $\mathbf{v}$ such that $\Delta_+=1+x$ have the same energy.

For numerical simulations, we use Annealed Importance Sampling (Jarzynski, 1997; Neal, 2001; Salakhutdinov and Murray, 2008) to estimate the RBM partition function (Appendix C.2.2.1). These numerical results are in agreement with the theoretical ones.

For $K=2$, it is important to note that

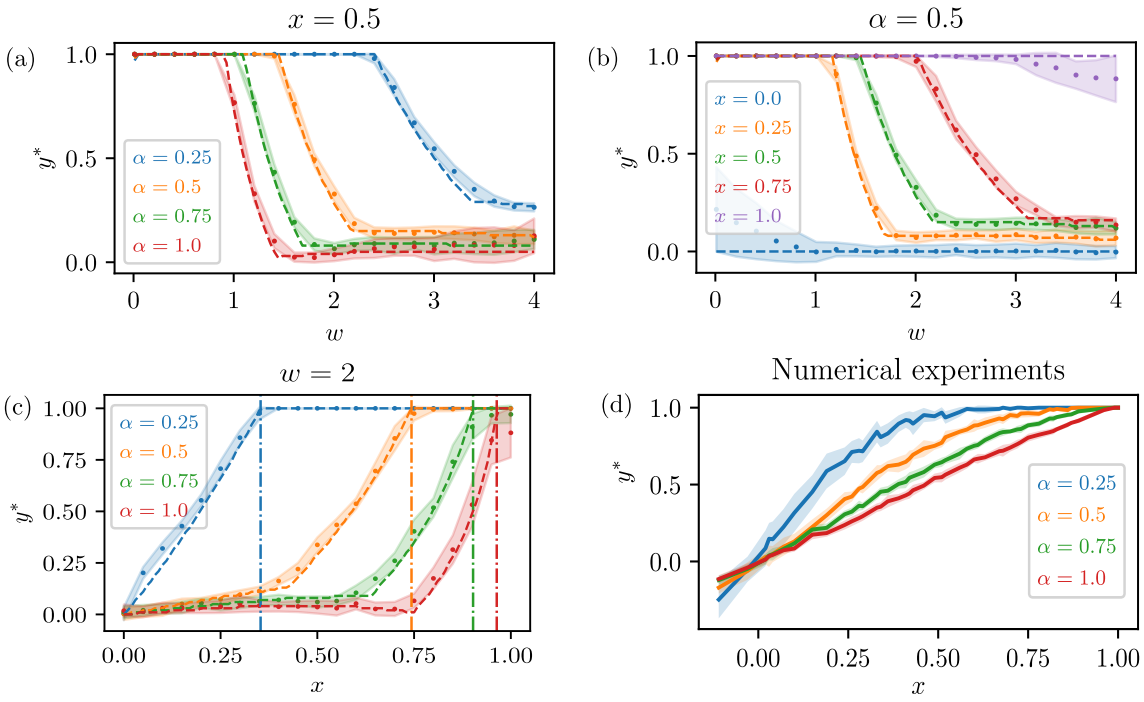


Figure 4.5: (a-c) Optimal overlap $y^*(w, x, \alpha)$. Dashed lines: theoretical results obtained by minimized Eq. (4.5) with respect to y. Dots: numerical results with AIS ($N = 200$) (mean value over 25 initialization of $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$ with a correlation equal to x). Shaded areas correspond to the empirical error bars. (a) Fixed $x = 0.5$ (b) Fixed $\alpha = 0.5$. (c) Fixed $w = 2$. Dashdotted lines: theoretical x_c. (d) y vs x for numerical maximization of the log-likelihood with Contrastive Divergence ($N = 200$).

$$\mathbf{h}^{1^T} \cdot \mathbf{h}^2 = \text{sign}\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^1\right)^T \cdot \text{sign}\left(\mathbf{W}^T \cdot \boldsymbol{\xi}^2\right) = y, \tag{4.8}$$

i.e. that the correlation between the hidden representations $\mathbf{h}^1$ and $\mathbf{h}^2$ is the same as the one between $\hat{\boldsymbol{\xi}}^1$ and $\hat{\boldsymbol{\xi}}^2$.

We have shown the existence of a critical threshold x_c, which is an increasing function of α and w, above which the hidden representations are compressed ($y = 1$).

4.4.4.2 Numerical results

We can train RBM with CD or PCD to maximize the log-likelihood (Eq. (4.4)). At the beginning of the training, the weight matrix $\mathbf{W}^0$ is initialized with small random values chosen from a zero-mean Gaussian with a standard deviation σ, typically $\sigma = \mathcal{O}(N^{-1})$. By defining $\mathbf{h}^1 = \text{sign}\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^1\right)$ and $\mathbf{h}^2 = \text{sign}\left(\mathbf{W}^{0^T} \cdot \boldsymbol{\xi}^2\right)$, the initial correlation y^0 can be computed (Appendix. C.2.2.2):

$$y^0 = 1 - \frac{2\cos^{-1}(x)}{\pi}. \tag{4.9}$$

During the training, by computing the singular values decomposition of $\mathbf{W}^t$, we observe that two singular values emerge from the bulk as long as $y < 1$ (Fig. C.1(d)). We neglect the effects of the $M - 2$ other singular values. $\mathbf{W}^t$ is of the form of Eq. (4.3), where w^t,

$\mathbf{h}^{1^t}$ and $\mathbf{h}^{2^t}$ (and therefore their correlation y^t) depend on time. α and x are fixed during the training, but w increases: $y^*(w^t, x, \alpha)$ also depends on time.

During the first steps of the training, as $\Delta_+ = \Delta_- = 0$, the optimal y^* is equal to 1. Then y^t and w^t are increasing during the training (Fig. C.1(b)). As w^t increases, the optimal $y^*(w^t, x, \alpha)$ changes. However, numerically, this optimal correlation is never reached. In the regime where $\alpha w \gg 1$, the evolution of y^t is very slow: y^t is in the range of correlation with a log-likelihood near the optimum. As w^t increases over time, the range of y which has a log-likelihood near 0 is increasing: with finite time steps, y^t never reaches the optimal $y^*(w^t, x, \alpha)$ (Fig. 4.5(d)) but corresponds to a solution with a log-likelihood near 0 (Fig. C.1(c)).

At the end of the training, for a finite number of time steps, the correlation y found with CD is an increasing function of x and a decreasing function of α. The smaller α is, the bigger y is. For α small, $y(x)$ is above the bisecting line: the hidden representations are closer than the initial patterns. Moreover, we notice the existence of a threshold x_c from which for $\forall x > x_c$, $y^* = 1$. In this case, there is a compression of the representations.

4.5. Deep Tempering: barriers and replica exchange

In this section, we will calculate the free energy for an arbitrary number of K (finite) patterns. This will allow us to compute the characteristic times of Deep Tempering and to compare it to those obtained with Alternating Gibbs Sampling in the case where the data are represented by K orthogonal clusters. We then show that Deep Tempering allows sampling more efficiently than Alternating Gibbs Sampling.

4.5.1 Computation of the partition function

Within the hypothesis defined in Section 4.4.1, we want to compute the partition function of the RBM

$$Z = \sum_{\{v_i, h_\mu = \pm 1\}} \exp\left(\sum_{i,\mu} W_{i\mu} v_i h_\mu\right). \tag{4.10}$$

To do so we introduce the order parameters

$$m_k = \frac{1}{N} \sum_{i=1}^{N} v_i \xi_i^k, \tag{4.11}$$

where m_k is the magnetization along the pattern $\boldsymbol{\xi}^k$. Consequently, the partition function Z reads

$$
\begin{aligned}
Z &= \int \prod_k \frac{dm_k d\hat{m}_k}{(2\pi/N)^{\frac{K}{2}}} \sum_{\{h_\mu\}} \exp\Big(\sum_\mu h_\mu \sum_k w_k m_k \hat{\xi}_\mu^k\Big) \sum_{\{v_i\}} \exp\Big(\sum_k \hat{m}_k (\sum_i v_i^k v_i - N m_k)\Big) \\
&= \int \prod_k \frac{dm_k d\hat{m}_k}{(2\pi/N)^{\frac{K}{2}}} \sum_{\boldsymbol{\sigma}} \Big(2\cosh(\sum_k w_k \sigma_k m_k)\Big)^{M\gamma_h(\boldsymbol{\sigma})} \sum_{\{v_i\}} \exp\Big(\sum_k \hat{m}_k (\sum_i \xi_i^k v_i - N m_k)\Big) \\
&= \int \prod_k \frac{dm_k d\hat{m}_k}{(2\pi/N)^{\frac{K}{2}}} \sum_{\boldsymbol{\sigma}} \Big(2\cosh(\sum_k w_k \sigma_k m_k)\Big)^{M\gamma_h(\boldsymbol{\sigma})} \Big(2\cosh(\sum_k w_k \sigma_k \hat{m}_k)\Big)^{N\gamma_v(\boldsymbol{\sigma})} \\
&\quad \times \exp\Big(-N \sum_k m_k \hat{m}_k\Big) \\
&= \int \prod_k \frac{dm_k d\hat{m}_k}{(2\pi/N)^{\frac{K}{2}}} \exp\left(-N F(\mathbf{m}, \hat{\mathbf{m}})\right), \qquad (4.12)
\end{aligned}
$$

where $\sum_{\boldsymbol{\sigma}}$ runs over the 2^K vectors $\boldsymbol{\sigma}$ of length K with binary coefficients (± 1). The K vectors $\boldsymbol{\xi}^k$ can be written in a matrix of size $N \times K$, and $\gamma_v(\boldsymbol{\sigma})$ is the frequency of $\boldsymbol{\sigma}$ among the N lines of this matrix. In the same way, we define $\gamma_h(\boldsymbol{\sigma})$ for the K vectors $\hat{\boldsymbol{\xi}}^k$. We have

$$
\begin{aligned}
\boldsymbol{\xi}^{a^T} \cdot \boldsymbol{\xi}^b &= N \sum_{\boldsymbol{\sigma}} \gamma_v(\boldsymbol{\sigma}) \sigma_a \sigma_b, \qquad (4.13) \\
\hat{\boldsymbol{\xi}}^{a^T} \cdot \hat{\boldsymbol{\xi}}^b &= M \sum_{\boldsymbol{\sigma}} \gamma_h(\boldsymbol{\sigma}) \sigma_a \sigma_b. \qquad (4.14)
\end{aligned}
$$

By rescaling $\alpha\, w_k \hat{m}_k \leftarrow \hat{m}_k$, the free-energy $F(\mathbf{m}, \hat{\mathbf{m}})$ reads

$$
\begin{aligned}
F(\mathbf{m}, \hat{\mathbf{m}}) &= -\sum_{\boldsymbol{\sigma}} \left(\alpha \gamma_h(\boldsymbol{\sigma}) \log 2\cosh\left(\sum_{k=1}^K w_k \sigma_k m_k\right) + \gamma_v(\boldsymbol{\sigma}) \log 2\cosh\left(\alpha \sum_{k=1}^K w_k \sigma_k \hat{m}_k\right)\right) \\
&\quad + \sum_{k=1}^K \alpha\, w_k \hat{m}_k m_k \qquad (4.15)
\end{aligned}
$$

where we can identify

$$
\hat{m}_k = \frac{1}{M} \sum_{\mu=1}^M h_\mu \hat{\xi}_\mu^k. \qquad (4.16)
$$

Therefore, $\hat{m}_k$ is the magnetization along the pattern $\hat{\boldsymbol{\xi}}^k$.

The knowledge of the free energy $F(\mathbf{m}, \hat{\mathbf{m}})$ will allow us to calculate the log-likelihood $\mathrm{LL} = \frac{1}{K} \sum_k \log P(\boldsymbol{\xi}^k)$, and therefore to determine the optimal representations $\hat{\boldsymbol{\xi}}^k$, as we have done in the Section 4.4.4 for $K = 2$.

Moreover, by determining the critical points of $F(\mathbf{m}, \hat{\mathbf{m}})$ (Section 4.5.2), we will be able to determine the size of the barriers in this energy landscape, which will be particularly useful for theoretical comparison between Alternating Gibbs Sampling and Deep Tempering (Section 4.5.4).

4.5.2 Critical points of $F(\mathbf{m}, \hat{\mathbf{m}})$

To determine the critical points of $F(\mathbf{m}, \hat{\mathbf{m}})$, we compute $\frac{\partial F(\mathbf{m},\hat{\mathbf{m}})}{\partial \mathbf{m}} = \mathbf{0}$ and $\frac{\partial F(\mathbf{m},\hat{\mathbf{m}})}{\partial \hat{\mathbf{m}}} = \mathbf{0}$, and end up with following self-consistent equations

$$\begin{aligned} \hat{m}_k &= \sum_{\boldsymbol{\sigma}} \gamma_h(\boldsymbol{\sigma}) \sigma_k \tanh\left(\sum_{k=1}^{K} w_k \sigma_k m_k \right), \\ m_k &= \sum_{\boldsymbol{\sigma}} \gamma_v(\boldsymbol{\sigma}) \sigma_k \tanh\left(\alpha \sum_{k=1}^{K} w_k \sigma_k \hat{m}_k \right). \end{aligned} \tag{4.17}$$

In the specific case where the patterns $\boldsymbol{\xi}^k$ are orthogonal, and the hidden representations $\hat{\boldsymbol{\xi}}^k$ are also orthogonal, $\gamma_v(\boldsymbol{\sigma}) = \gamma_h(\boldsymbol{\sigma}) = 2^{-K}$, and by symmetry, $w_k = w$.

In that case, we can solve the self-consistent equations (Eqs. 4.17).

As in the Hopfield model with a finite number of patterns finite number of patterns (Amit et al., 1985a), we are interested in symmetric spurious patterns of the form $\boldsymbol{\xi} = \xi_r(\underbrace{1,1,\ldots 1}_{r},\underbrace{0,0,\ldots,0}_{K-r})$. There is complete symmetry of the solutions under the permutations of the components of $\boldsymbol{m}$ as well as under the change of sign of any of them. We consider that the r components different from zero are the r first ones and $m_r > 0$. Using Eq. (4.17), we find

$$\hat{m}_r = \begin{cases} 2^{-r} \sum_{\boldsymbol{\sigma}} \tanh\left(w \left(m_r + \sum_{\substack{i=1 \\ i \neq r}}^{r} \sigma_i m_r \right) \right) & \text{if } k \leq r \\ 2^{-r} \sum_{\boldsymbol{\sigma}} \tanh\left(w \left(\sum_{i=1}^{r} \sigma_i m_r \right) \right) = 0 & \text{if } k > r \end{cases} \tag{4.18}$$

where $\sum_{\boldsymbol{\sigma}}$ runs over the 2^r vectors $\boldsymbol{\sigma}$ of length r with binary coefficients (± 1). Thus, we can write $\hat{\boldsymbol{m}} = \hat{m}_r(\underbrace{1,1,\ldots 1}_{r},\underbrace{0,0,\ldots,0}_{K-r})$. Using Eq. (4.17), we get

$$m_r = 2^{-r} \sum_{\boldsymbol{\sigma}} \tanh\left(\alpha w \left(\hat{m}_r + \sum_{\substack{i=0 \\ i \neq r}}^{r} \sigma_i \hat{m}_r \right) \right). \tag{4.19}$$

There exists a solution with $m_r > 0$ if $\alpha w^2 > 1$. For $\alpha w \hat{m}_r \gg 1$ and $w m_r \gg 1$, we can solve it analytically (Fig. 4.6(a)):

$$\hat{m}_r = m_r = \frac{1}{2^{2c}} \binom{2c}{c}, \tag{4.20}$$

where $r = 2c$ for even r and $r = 2c+1$ for odd r. This result is similar to the symmetric spurious memories of the Hopfield model at $T = 0$ (Amit et al., 1985a).

We can compute the energy F_r of the symmetric spurious patterns with r components:

$$F_r = \begin{cases} -\alpha w r m_r^2 & \text{if r is odd} \\ -\left(\alpha w r m_r^2 + (1+\alpha) m_r \log 2\right) & \text{if r is even} \end{cases} \tag{4.21}$$

We can order these energies

$$F_1 < F_3 < F_5 < \ldots < F_4 < F_2. \tag{4.22}$$

For even r, the energy is a decreasing function of r. For odd r, the energy is an increasing function of r (Fig. 4.6(b)). The K fixed points aligned with one of the $\boldsymbol{\xi}^k$'s have the lowest energy. Therefore, with this parametrization of the weights, the RBM has learned the patterns. The second-lowest energy is reached for $r = 3$. Numerically, asymmetric critical points exist, but their energies are higher than the energy of the symmetric critical points with $r = 3$. These results are also true for finite values of αw (Figs. 4.6(c) and (d)).

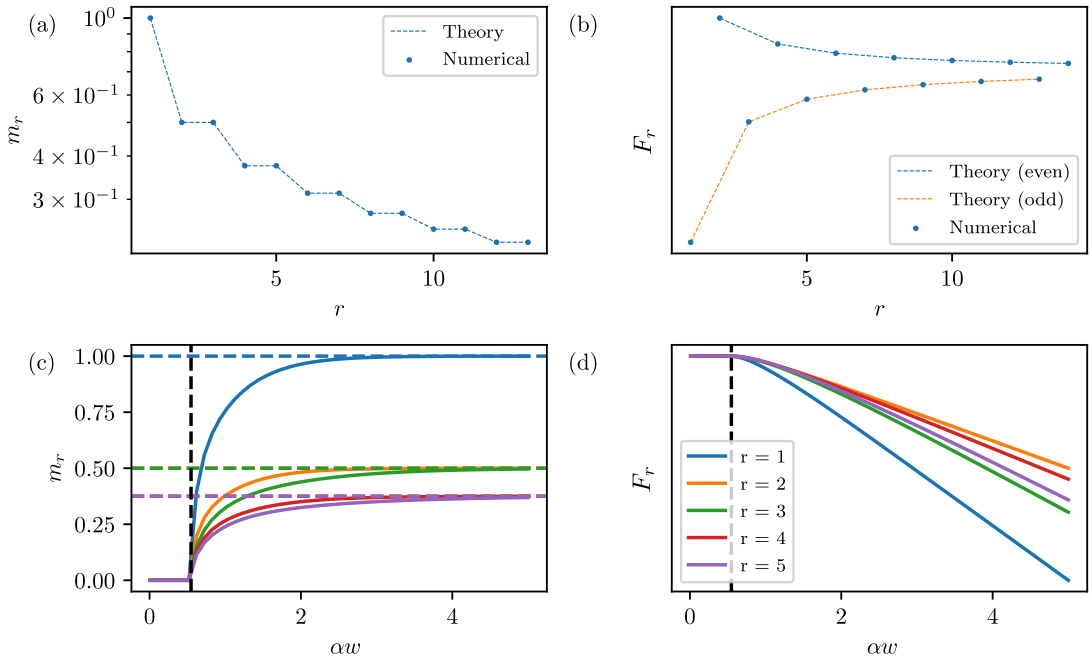


Figure 4.6: (a-b) m_r and F_r for $\alpha w \hat{m}_r \gg 1$ and $w m_r \gg 1$. Dashed lines: theoretical results (Eqs. (4.20) and (4.21)). Dots: numerical results. (a) Value of m_r vs r. (b) Energy F_r of the symmetric spurious patterns with r components vs r. (c-d) m_r and F_r against αw. Black dashed line: threshold where $\alpha w^2 = 1$. (c) Full lines: numerical m_r for different values of r (see legend of panel (d)). Dashed lines: theoretical value for large w. (d) Full lines: numerical F_r for different values of r.

4.5.3 Gradient of the log-likelihood

In the thermodynamic limit $N \to \infty$, thanks to the saddle point in the computation of the partition function Z (Eq. 4.12), the average over the model $\langle \frac{\partial E^{\text{eff}}(\mathbf{v})}{\partial \Theta} \rangle_{\text{model}}$ is replaced by $\langle \frac{\partial F(\mathbf{m},\hat{\mathbf{m}})}{\partial \Theta} \rangle_{\text{fixed points}}$, where the mean value $\langle . \rangle_{\text{fixed points}}$ denotes the expected value over the fixed points defined in Eqs. (4.17). For an operator O depending on $\{\mathbf{m}, \hat{\mathbf{m}}\}$, we get

$$\langle O \rangle_{\text{fixed points}} = \int \prod_k dm_k d\hat{m}_k O \frac{\exp\left(-NF(\mathbf{m},\hat{\mathbf{m}})\right)}{Z}, \tag{4.23}$$

$$Z = \int \prod_k dm_k d\hat{m}_k \exp\left(-NF(\mathbf{m},\hat{\mathbf{m}})\right),. \tag{4.24}$$

The gradients $\frac{\partial \text{LL}}{\partial \mathbf{W}}$ (Eq. (1.14)), for a given training set of K samples $\{\boldsymbol{\xi}^k\}_{k=1\ldots K}$ read

$$\frac{\partial \mathrm{LL}}{\partial \mathbf{W}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\xi}^k \tanh\left(\mathbf{W}^T \cdot \boldsymbol{\xi}^k\right)^T - \langle \mathbf{m} \cdot \hat{\mathbf{m}} \rangle_{\text{fixed points}}. \tag{4.25}$$

The knowledge of this gradient will allow us to ensure that the weight matrices proposed in Section 4.5.4 correspond to maxima of the log-likelihood.

4.5.4 Comparison of Alternating Gibbs Sampling and Deep Tempering to sample K orthogonal clusters

In this section, we show in a simple case how the Deep Tempering presented in Section 1.3.4 and fully defined by Algorithm 6 could help the sampling process. We point out that in the following computation, there is no compression of the representations. Nevertheless, this computation is still useful to highlight the trade-off that exists between guaranteeing a high acceptance ratio while having a top RBM that mixes better than the bottom one.

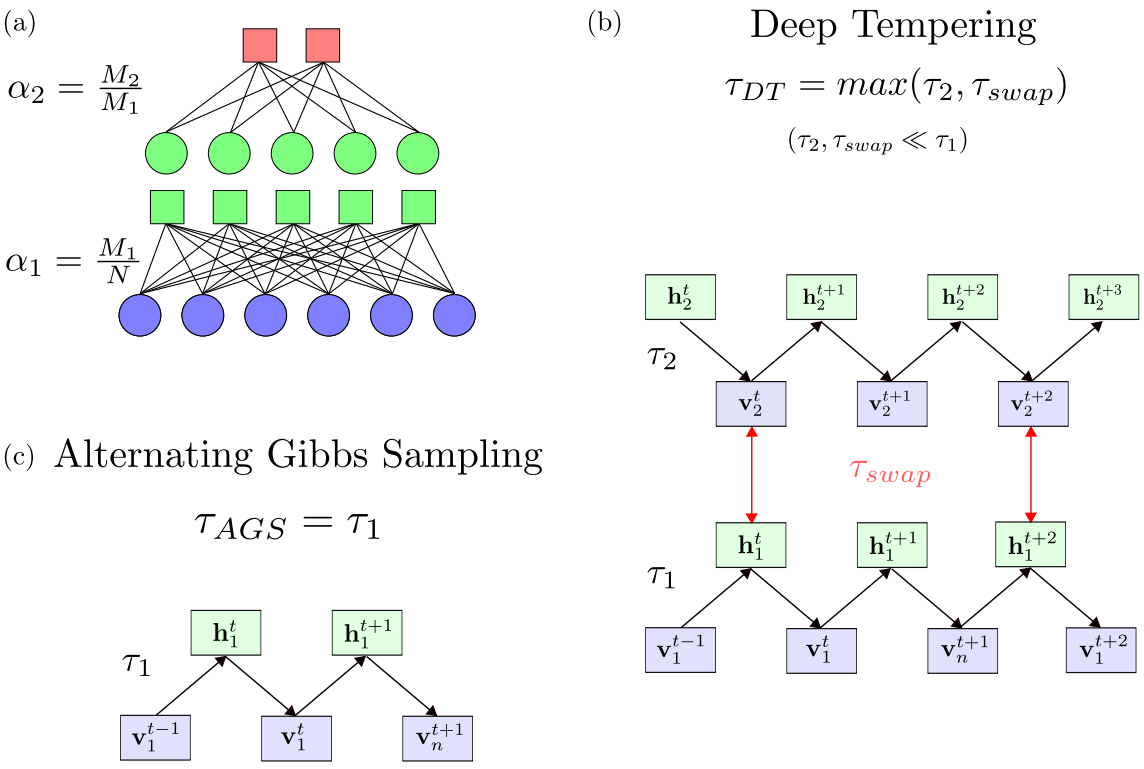


Figure 4.7: Definition of characteristic times of Alternating Gibbs Sampling and Deep Tempering. (a) Definition of the aspect ratios. (b) Definition of characteristic times of Deep Tempering. (c) Definition of characteristic time of Alternating Gibbs Sampling.

We consider data formed by K orthogonal clusters $\mathcal{C}_k$. These K clusters can be stored with a single RBM. We compute the characteristic time scale τ_1 to go from one cluster to another in $E_1^v(\mathbf{v})$ with AGS (Fig. 4.7(c)). A second RBM can be stacked on top of the first one, and Deep Tempering can be used to sample the energy landscape $E_1^v(\mathbf{v})$ of the first RBM. We compute the characteristic time scale τ_{DT} to go from one cluster to another in $E_1^v(\mathbf{v})$ with Deep Tempering (Algorithm 6). As long as $\tau_{\mathrm{DT}} \ll \tau_1$, τ_{DT} is equal to $\max(\tau_2, \tau_{\mathrm{swap}})$ where τ_2 is the time scale to go from one cluster to another in $E_2^v(\mathbf{v})$ for the second RBM with AGS and τ_{swap} is the time scale between two replica exchanges between

$\mathbf{h}_1^t$ and $\mathbf{v}_2^t$, defined as $\langle A_1\left(\{\mathbf{h}_1=\mathbf{h}_1^t,\mathbf{v}_2=\mathbf{v}_2^t\}\rightarrow\{\mathbf{h}_1=\mathbf{v}_2^t,\mathbf{v}_2=\mathbf{h}_1^t\}\right)\rangle^{-1}_{\{\mathbf{h}_1^t,\mathbf{v}_2^t\}}$ (Eq. (1.18), Fig. 4.7(b)). In fact, to move from one cluster to another with Deep Tempering, if $\tau_{\text{DT}}\ll\tau_1$, $\mathbf{v}_2^t$ must change of clusters in $E_2^v(\mathbf{v})$ (with a characteristic time of τ_2) and $\mathbf{v}_2^t$ must be exchange with $\mathbf{h}_1^t$ (with a characteristic time of τ_{swap}). As these two times are exponential, the typical time scale is therefore $\max(\tau_2,\tau_{\text{swap}})$. All these characteristic times can be calculated analytically and evaluated numerically.

Each cluster $\mathcal{C}_k$ is characterized by its center $\boldsymbol{\xi}_1^k\in\{-1,1\}^N$. The centers $\{\boldsymbol{\xi}_1^k\}_{k=1\dots K}$ are a set of mutually orthogonal vectors ($\forall k,k',k\neq k',\boldsymbol{\xi}_1^{k^T}\cdot\boldsymbol{\xi}_1^{k'}=0$). Inside a given cluster $\mathcal{C}_k$, each spin $v_{1,i}$ of a vector $\mathbf{v}_1$ is drawn from $P(v_{1,i})=(1-d)\,\delta(v_{1,i}-\xi_{1,i}^k)+d\,\delta(v_{1,i}+\xi_{1,i}^k)$. The mean Hamming distance between a vector $\mathbf{v}_1\in\mathcal{C}_k$ and the center $\boldsymbol{\xi}_1^k$ is dN.

A RBM with N visible units and $M_1=\alpha_1 N$ hidden units can encode the K clusters (Fig. 4.7(a)). There is no potential acting on the visible units. There are hidden fields $\mathbf{c}_1$ acting on the hidden units. As shown in Section 4.4.1, the weight matrix $\mathbf{W}_1$ is given by:

$$\mathbf{W}_1=\frac{w_1}{N}\sum_{k=1}^{K}\boldsymbol{\xi}_1^k\cdot\hat{\boldsymbol{\xi}}_1^{k^T}, \tag{4.26}$$

where $\{\hat{\boldsymbol{\xi}}_1^k\}_{k=1\dots K}$ is a set of mutually orthogonal vectors. Without hidden fields, $\mathbf{v_1}$ and $-\mathbf{v_1}$ have the same energy. The hidden fields $c_{1,\mu}$ are an increasing odd function of $\sum_{k=1}^{K}\hat{\xi}_{1,\mu}^k$: the hidden fields break this symmetry, the patterns $\boldsymbol{\xi}_1^k$ have a lower free energy than $-\boldsymbol{\xi}_1^k$. In the thermodynamic limit $N\rightarrow\infty$, all the configurations $\mathbf{v}\in\mathcal{C}_k$ have the same hidden representations $\hat{\boldsymbol{\xi}}_1^k$ (for $d\ll\frac{1}{\sqrt{K-1+1}}$). To enforce that the visible configurations drawn from $P(\mathbf{v}|\hat{\boldsymbol{\xi}}_1^k)$ belong to the cluster $\mathcal{C}_k$, w_1 must be equal to

$$w_1=\frac{\tanh^{-1}(1-2d)}{\alpha_1}, \tag{4.27}$$

With this parametrization, the log-likelihood of the data reaches a maximum ($\frac{\partial \text{LL}_1}{\partial \mathbf{W}}\Big|_{\mathbf{W}=\mathbf{W}_1}=0$, Eq. (4.25)).

For this RBM with AGS, the characteristic time scale τ_1 to go from one cluster to another in $E_1^v(\mathbf{v})$ scales as

$$\log(\tau_1)\sim N\mathcal{B}(\alpha_1 w_1,K), \tag{4.28}$$

where the function $\mathcal{B}$ is defined in Appendix C.3.1. $\mathcal{B}(\alpha_1 w_1,K)$ is an increasing function of $\alpha_1 w_1$.

A second RBM with $M_2=\alpha_2 M_1$ hidden units is trained on the K representations $\{\hat{\boldsymbol{\xi}}_1^k\}_{k=1\dots K}$ (Fig. 4.7(a)). Its weight matrix $\mathbf{W}_2$ can be written

$$\mathbf{W}_2=\frac{w_2}{M_1}\sum_{k=1}^{K}\hat{\boldsymbol{\xi}}_1^k\cdot\hat{\boldsymbol{\xi}}_2^{k^T}, \tag{4.29}$$

where $\{\hat{\boldsymbol{\xi}}_2^k\}_{k=1\dots K}$ is a set of mutually orthogonal vectors. This second RBM has also hidden fields $c_{2,\mu}$ to break the symmetry between $\mathbf{h}_1$ and $-\mathbf{h}_1$. The hidden fields $c_{2,\mu}$ are an increasing odd function of $\sum_{k=1}^{K}\hat{\xi}_{2,\mu}^k$. The main role of this second RBM is to improve the

sampling of the first RBM between the different clusters $\mathcal{C}_k$. w_2 is a free-parameter which can be tuned by adding a regularization during the training. With this parametrization, the log-likelihood reaches a maximum ($\frac{\partial \mathrm{LL}_2}{\partial \mathbf{W}}\Big|_{\mathbf{W}=\mathbf{W}_2} = 0$, Eq. (4.25)). For this second RBM with AGS, the characteristic time scale τ_2 to go from one cluster to another in $E_2^v(\mathbf{v})$ scales as

$$\log(\tau_2) \sim M_1 \mathcal{B}(\alpha_2 w_2, K), \tag{4.30}$$

$\mathcal{B}(\alpha_2 w_2, K)$ is an increasing function of $\alpha_2 w_2$. With AGS, first RBM generates configurations $\{\mathbf{v}_1^t, \mathbf{h}_1^t\}$ and second RBM generates $\{\mathbf{v}_2^t, \mathbf{h}_2^t\}$. These two chains are coupled as described in Fig. 1.2(b). Swap between $\mathbf{h}_1^t$ and $\mathbf{v}_2^t$ is accepted with an acceptance ratio $A_1(\mathbf{h}_1^t, \mathbf{v}_2^t)$. The mean value of this acceptance ratio can be computed, and we can define a characteristic time τ_{swap} between two replica exchanges (Appendix C.3.2):

$$\log(\tau_{\mathrm{swap}}) \sim M_1 \log(1 + \exp(-2\alpha_2 w_2)). \tag{4.31}$$

- For $\alpha_2 w_2 \ll 1$, $\tau_2 \ll \tau_{\mathrm{swap}}$: the dynamics of the second RBM can visit the different minima of $E_2^v(\mathbf{v})$. However, as the acceptance ratio is small, the dynamics of the two RBM are decoupled. $E_2^{\mathbf{v}}(\mathbf{v})$ is a poor approximation of $E_1^h(\mathbf{h})$.
- For $\alpha_2 w_2 \gg 1$, $\tau_2 \gg \tau_{\mathrm{swap}}$: the acceptance ratio is high, the dynamics of the two RBM are coupled. However, the dynamics of the second RBM is stuck in a given minimum of $E_2^v(\mathbf{v})$. $E_2^v(\mathbf{v})$ is a similar to $E_1^h(\mathbf{h})$ and has also high free energy barriers.

There exists an optimal $\alpha_2 w_2^*$ (Figs. 4.8(c) and (f)). At this optimal value, $\max(\tau_2, \tau_{\mathrm{swap}}) \ll \tau_1$. In that case, the coupled dynamics with the two RBM improve the sampling. In this example, at the optimal $\alpha_2 w_2^*$, the landscape $E_2^v(\mathbf{v})$ is a smooth approximation of $E_1^h(\mathbf{h})$ and the acceptance ratio is high. Choosing the optimal $\alpha_2 w_2^*$ is similar to choosing the optimal temperature in parallel tempering.

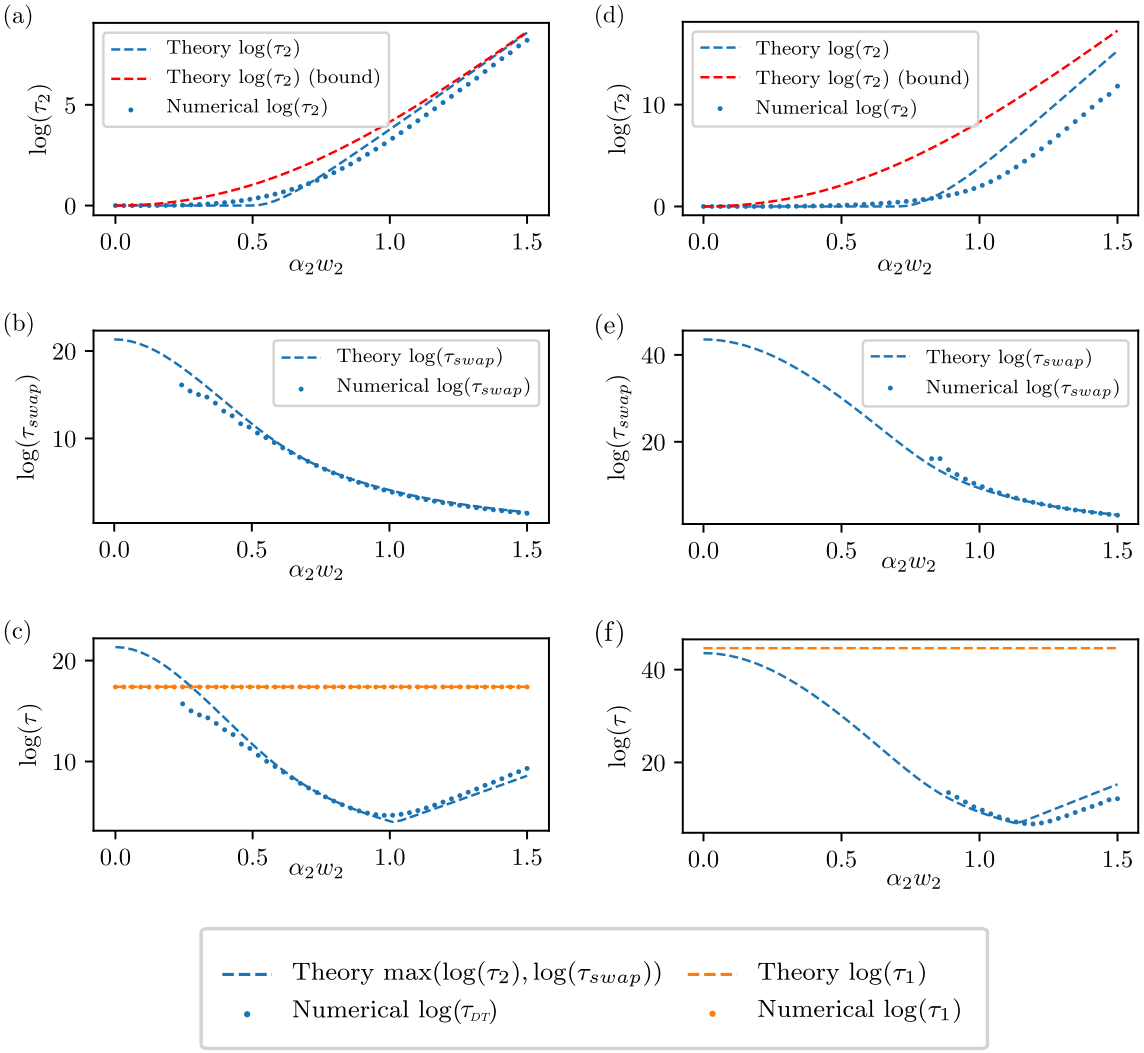


Figure 4.8: (a-c) Example with $K = 3$, $N = 128$, $M_1 = 32$ and $M_2 = 8$ and (d-f) $K = 5$, $N = 256$, $M_1 = 64$ and $M_2 = 32$. Dashed lines: theoretical results. Dots: numerical estimations (Appendix. C.3.4). (a) and (d) Characteristic time τ_2 vs $\alpha_2 w_2$. The offset between the theory and the numerical results for large value of $\alpha_2 w_2$ is equal to $-(K-2)\log(2)$. This term corresponds to the logarithm of the number of optimal distinct paths between two global minima. (b) and (e) Characteristic time τ_{swap} vs $\alpha_2 w_2$. (c) and (f) Characteristic time of the dynamics vs $\alpha_2 w_2$. τ_1 for the Alternating Gibbs Sampling for the first RBM. τ_{DT} for the Deep Tempering.

4.6. Conclusion

We have shown numerically that the Deep Tempering algorithm can improve the sampling efficiency when a large number of hidden D units must change states at the same time, to switch from one data mode to another. The influential parameters of Deep Tempering, be it the number of RBM in the stack, the number of hidden units as well as the intensity of the regularization were chosen empirically, in order to guarantee a good acceptance ratio while reducing energies barriers. Although the theoretical results allow us to have an idea about the choice of these parameters, we currently do not have an efficient method to select all these parameters automatically. It would be interesting to see how to automate this parameter selection, so that Deep Tempering can be used effectively as a

sampling algorithm.

Theoretically, we have explained the smoothing of the energies barriers in a very simple example, where the data modes are orthogonal clusters. In this particular example, there is no compression of representations. We have also explained the compression of the hidden representations in the specific case $K = 2$. It would be interesting to combine these two approaches in a more general framework like for example with a data structure similar to the hierarchical structure discussed in Section 4.3. We started to study this case, but unfortunately we did not have time to finish the computation, so we cannot present it in this manuscript. Moreover, it would be also interesting to relax the assumption of a finite number of patterns K, and to study properly the infinite case K.

This study concludes our analysis of Alternating Gibbs Sampling. We can make the following conclusions. When large valleys connect the different minima of the free energy landscape, AGS is effective in sampling between the minima. When large barriers are to be crossed, this is no longer true. Using the representations learned from RBM, it is possible to improve the sampling. If a small number D of hidden units is sufficient to change minima, then the Metropolis Hastings introduced in Chapter 3 is sufficient to improve sampling. If D is of the order of M, the Deep Tempering algorithm can improve the sampling, by progressively compressing the representations and decreasing the free energy barriers.

An introduction to proteins modeling

Part III summary

This part aims to introduce proteins to a public of physicists, and to try to make them understand their primordial role, as well as the many stakes and difficulties that arise from it, and specifically the passage from genotype to phenotype, *i.e.* from the sequence of the protein to its functionality.

Nevertheless, all hope is not lost, and many important advances have been made. We mainly detail the advances related to Direct Coupling Analysis using the Potts model. We also briefly mention the advances related to deep learning, to satisfy the reader's curiosity.

This part aims to give the necessary reading keys to understand the Part IV, which is dedicated to class A β-lactamase, and presents the main results on the biology counterpart of my thesis work.

5

An introduction to proteins for physicists

The purpose of this chapter is to quickly introduce proteins and give key concepts to help physicists better understand protein issues.

5.1. Proteins: the basis of life

Proteins are long polymer chains built with elementary blocks called amino acids. The amino acids linked by peptide bonds define the sequence of a protein. The number of amino acids depends on the proteins and ranges from about ten to several thousands (Brocchieri and Karlin, 2005).

Twenty different amino acids, with different chemical properties, are the basic elements of all proteins in all living cells. Each human cell contains several billion proteins (Beck et al., 2011), many are identical, but the distribution of the number of copies of unique proteins in a single cell is broad: from few copies to several million (Milo, 2013). Proteins are involved in most biological functions within organisms, such as catalyzing metabolic reactions, structuring the cell, DNA replication, responding to stimuli, transporting molecules into the cells ...

5.1.1 How are the proteins built?

Inside the double-stranded DNA, each protein is encoded by a gene that defines its nucleotide sequence. Each amino acid is defined by a sequence of three nucleotides, called codons. Four different nucleotides exit (adenine (A), cytosine (C), guanine (G) and thymine (T)), and then $4^3 = 64$ possible codons. Therefore, there is some redundancy: some of the twenty different amino acids are encoded by more than one codon.

Genes in the DNA are copied into single-stranded messenger-RNA through a mechanism called transcription. Then, the messenger-RNA is read by ribosomes. Finally, ribosomes perform the messenger-RNA translation, assembling several amino acids to make the protein described by the messenger-RNA. Put in crude terms, messenger-RNA are proteins' blueprints and ribosomes the factories which build them. It turns out that it is possible to design and deliver specific messenger-RNA to the ribosomes to synthesize a given protein in cells. Since the 1990s, and the seminal works of the biochemist Katalin Karikó, (Karikó et al., 2005), scientists and pharmaceutical industries have used this mechanism to try to design new vaccines (Pardi et al., 2018; Keener, 2018). The basic idea is to use ribosomes to build specific antigens of a given virus and then triggering an immune response to train T-cells (a type of lymphocyte) to recognize the antigens, and therefore the virus.

These long-term efforts have born fruits recently in the fight against the COVID-19 (SARS-CoV-2), and have led to two different vaccines, one from Pfizer-BioNTech (Polack et al., 2020) and another from Moderna (Jackson et al., 2020). These two vaccines aim at

the spike protein (S protein) of the coronavirus, the protein that allows the virus to enter into the cells (Huang et al., 2020).

5.1.2 On the importance of the three-dimensional structure

After its transcription by the ribosomes, the protein quickly folds into a unique three-dimensional structure called native conformation, on a time scale of about a few milliseconds (Kubelka et al., 2004). A protein becomes biologically functional after its folding: its properties depend mainly on its three-dimensional structure.

As we can expect from the short time scale needed for folding, a protein does not explore all the configurations space of the conformations to find the good one. In his thought experiment in his well-known paper (Levinthal, 1969), Cyrus Levinthal argues that a protein with 100 residues typically has 10^{300} possible conformations due to the large degree of freedom of the peptide bonds between amino acids (basically the angles between them). Even if a protein samples each conformation with a typical time scale of the picosecond, it would need a time larger than the universe's age to find its correct conformation. The Nobel Prize Christian Anfinsen exposed the general principles which govern the folding of the proteins after its experiments on ribonucleases (Haber and Anfinsen, 1962; Anfinsen, 1973): the native structure of a protein is the thermodynamically stable structure. The native structure depends only on the protein sequence and the solution where it folds (its temperature, pressure, pH, etc...). Interestingly, it does not depend on the way the protein folds: a protein folds into its native structure after being transcripted by the ribosomes, if the protein is helped or not by chaperone molecules (proteins which help other proteins to fold properly), or if the protein was unfolded and refolded in a test tube. This work has some important consequences (Dill et al., 2008).

From an experimental perspective, protein folding studies can be conducted *in vitro*, *i.e.*, in controlled experiments in test tubes. It is easier and cheaper than experiments *in vivo*, *i.e.*, in living organisms.

From a theoretical point of view, although the space of configurations of the proteins is tremendously large, there are some hopes to describe and predict the structure of the proteins from their sequences. As there are 20 amino acids, there exists $20^{100} \simeq 10^{130}$ proteins with 100 residues. For comparison purposes, it is estimated that there are 10^{82} atoms in the observable universe. As state functions in the theory of thermodynamics at equilibrium in physics, folding depends only on its equilibrium thermodynamic state and not on the path that took the protein to reach its native conformation. To push the analogy further, in the kinetic theory of gases developed in the XIX$^{\text{th}}$ century, only a few state variables, such as temperature, pressure, and volume, are needed to describe the evolution of a macroscopic number of particles. Thus, finding the minimal set of state variables, if there exists one, could lead to an elegant and predictive theory of protein folding.

Furthermore, there exist some regularities in structure among all the proteins. There are four distinct levels of protein structure:

- primary structure: linear sequence of amino acid along the polypeptide backbone,
- secondary structure: highly regular sub-structures resulting from hydrogen bonds in the backbone. There exist two main types of secondary structure, the α-helices, and the β-sheets. Linus Pauling predicted the two structures before their experimental findings (Pauling and Corey, 1951; Pauling et al., 1951). Chapter 7 is dedicated to the study of the epistasis on an α-helix.
- tertiary structure: three-dimensional native conformation of the protein. The α-helices and the β-sheets are folded in a compact structure. Many non-covalent

interactions are critical, such as hydrophobic interactions, salt bridges, hydrogens bonds, or disulfide bonds between two cysteines,

- quaternary structure: three-dimensional structure of several interacting proteins that operate together as a single unit called multimer.

Numerical simulations for small proteins with Lattice Protein, shows that folding is possible thanks to two things: the rapid transition from an exploration phase among the random coil states to an exploration phase among the semi-contact globule states, which drastically reduces the number of possible conformations, as well as the existence of numerous transition states leading to the native structure (Sali et al., 1994a,b; Abkevich et al., 1994; Shakhnovich, 1997).

Nevertheless, despite all these apparent regularities in structure and the findings of Anfinsen, the theoretical prediction of the protein structure is one of the most unsolved difficult problems in science (Science, 2005). We will discuss in the following why it isn't easy from an experimental and theoretical point of view. Nevertheless, DeepMind recently accomplished a major milestone in structure prediction (Senior et al., 2020; Jumper et al., 2021; Tunyasuvunakool et al., 2021; Evans et al., 2021) (Section 6.4).

Knowing how to predict the structure of a protein from its sequence would have important consequences in biology and pharmacology. As the protein structure is essential for its functionality, it would be possible to link the genotype (the sequence) to the phenotype of a protein (its properties). Progress has been made in this direction, and proteins with new properties have been manufactured from scratch (Gandhi et al., 2019). Protein misfolding is also the cause of many neurodegenerative diseases, such as Alzheimer's and Huntington's disease (Kuhlman and Bradley, 2019). Understanding the causes of this protein misfolding could lead to possible treatments for diseases that are currently incurable.

5.1.3 Why is it difficult to predict the structure of a protein given its sequence?

5.1.3.1 From an experimental point of view

From an experimental perspective, determining the protein structure is a complicated, costly, and time-consuming process. Several methods exist, such as X-ray crystallography (Shi, 2014; Maveyraud and Mourey, 2020), nuclear magnetic resonance (NMR) in solid (Quinn et al., 2018), NMR in solution (Orts and Gossert, 2018), or more recently, cryo-electron microscopy (Nobel Prize in Chemistry 2017) (Murata and Wolf, 2018; Danev et al., 2019). X-ray crystallography and NMR require the crystallization of the proteins, which can take up to several months (McPherson and Gavira, 2013). Cryo-electron microscopy is promising, but is currently at a lower resolution than crystallography: median resolution reached by X-ray crystallography is 2.05Å compared to $3-4$Å for cryo-electron microscopy (Yip et al., 2020). However, these two techniques can be combined to take advantage of their respective benefits (Wang and Wang, 2017).

On June 8, 2021, on the one hand, 178.451 structures are stored in the Protein Data Bank (PDB)[1] (Berman et al., 2000). On the other hand, in the Uniprot database[2], 564.638 annotated by hand sequences (SwissProt) and 214.406.399 automatically annotated (TrEMBL) are available (The UniProt Consortium, 2021). The ratio of available structures to available sequences decreases with time: thanks to improvements in DNA sequence techniques, it is much easier to determine the sequence of a protein rather than its structure. Predicting the structure from the sequence is therefore a hot topic.

[1] https://www.rcsb.org/
[2] https://www.uniprot.org/

5.1.3.2 From a theoretical point of view

As Anfinsen's experiments showed, the folding of a protein is determined by its free energy decrease. From a classical physical point of view, one of the possible methods to solve that kind of problem is writing a Hamiltonian taking into account the interactions between the atoms and finding the conformation that minimizes the energy. These methods belong to Molecular Dynamics (Karplus and Kuriyan, 2005). The basic idea is to initialize the atomic model, compute the initial molecular forces acting on each atom, move each atom according to these forces and then advance the simulation time by 1 femtosecond (and repeat the last three steps until convergence) (Durrant and McCammon, 2011).

Instead of taking into account the interactions at the atomic level, coarse-grained models also exist that model the interactions at the scale of amino acids. The 20 amino acids (for a complete list, see Table 5.1) have a common structure, but their side chains differ. Their properties are well-known: amino acids could be hydrophilic or hydrophobic, positively or negatively charged, basic or acidic, polar, aromatic, and have various sizes. Hamiltonian must also consider interactions with solvents because some amino acids are hydrophilic and some others hydrophobic. In many models in classical physics, Hamiltonian can be simplified to keep only dominant interaction terms (*e.g.*, to study the interactions between a proton and an electron, the force of gravity can be neglect compared to electromagnetic force). However, in the case of protein folding, a majority of interactions have to take into account (Yang et al., 2007). These interactions are complex and could have opposite effects. For example, hydrophobic residues tend to be buried into the protein's core, away from the aqueous solvent, but polar residues tend to be at the surface of the protein to interact through hydrogen bonds with the water molecules of the solvent: if they are away from the solvent, they must form hydrogen bonds to compensate. Simulating these effects lead to high computational costs.

At the atomic scale or amino acids scale, the use of supercomputers, or collaborative projects based on distributed computing (such Rosetta@home[3] or Folding@home[4]), allow tackling this computational burden partially (Lindorff-Larsen et al., 2011; Conchuir et al., 2015; Huang et al., 2016). Nonetheless, these simulations are limited by the size of the proteins and the duration time of the simulation (a few milliseconds).

Furthermore, even if the native conformation is found, it is hard to predict the effects of mutations. On the one hand, two sequences that differ only in one amino acid could have very different native conformation: for example, inserting a proline in an α-helix, is known to destabilize it (von Heijne, 1991) (see Chapter 7 for example on TEM-1 α-helix). Therefore, it changes the secondary structure and consequently the tertiary one and modifies the properties of the protein as a whole. On the other hand, two proteins that share only 20% of their amino acids could have the same native conformation.

Advancements in genomics over the last 25 years have led to an important diminution in the cost of sequencing. According to the National Human Genome Research Institute, sequencing a human genome cost about \$100.000.000 in 2002 and \$1.000 nowadays[5]. As expressed below, there are currently 215 million sequences in the Uniprot database, but only about 0.0025% are annotated by hand, *i.e.*, sequences that scientists have studied. Each sequence does not have its own unique structure: for a given structure, thousands of sequences can share it. Indeed, sequences from distinct kingdoms (animals, bacteria, fungi, ...) could have the same functionality but have different sequences due to evolution. All these sequences are likely to share the same structure. Inside a given kingdom, sequences

[3] https://boinc.bakerlab.org/
[4] https://foldingathome.org
[5] https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data

can also differ from one species to another due to evolution. For example, two bacteria may have a common ancestor in the phylogenetic tree, but mutations have led to differences in their protein sequences. These sequences are grouped and called families of homologous proteins. The Pfam database[6] (Finn et al., 2014), On June 8, 2021, contains 19.179 protein domains. A protein domain is a subpart of a protein that folds independently of the others. A protein could have one or several domains. Each of these domains has around 10^2 to 10^5 sequences in the Pfam database.

Exploiting the statistical information contained in these sequences is called co-evolution. Conserved residues, *i.e.*, residues present in all sequences with a given functionality, can indicate functional importance. Correlation between two amino acids can reveal that they are closed in the three-dimensional structure. Analyzing this co-evolution is an active field of research for over fifteen years in bioinformatics, statistical physics, and machine learning. We will discuss some key concepts in the next section.

Amino Acids	Three Letters Code	Single letter code	Properties
Alanine	Ala	A	Non-polar, hydrophobic
Arginine	Arg	R	Positively charged, hydrophilic
Asparagine	Asn	N	Polar, hydrophilic
Aspartic Acid	Asp	D	Negatively charged, hydrophilic
Cysteine	Cys	C	Polar, hydrophobic
Glutamic Acid	Glu	E	Negatively charged, hydrophilic
Glutamine	Gln	Q	Polar, hydrophilic
Glycine	Gly	G	Aliphatic, hydrophobic
Histidine	His	H	Positively charged, hydrophilic
Isoleucine	Ile	I	Aliphatic, hydrophobic
Leucine	Leu	L	Aliphatic, hydrophobic
Lysine	Lys	K	Positively charged, hydrophilic
Methionine	Meth	M	Aliphatic, hydrophobic
Phenylalanine	Phe	F	Aromatic, hydrophobic
Proline	Pro	P	Hydrophilic
Serine	Ser	S	Polar, hydrophilic
Threonine	Thr	T	Polar, hydrophilic
Tryptophan	Trp	W	Aromatic, hydrophobic
Tyrosine	Tyr	Y	Aromatic, hydrophobic
Valine	Val	V	Aliphatic, hydrophobic

Table 5.1: List of amino acids, their three letters code, single letter code and main properties.

5.2. Protein co-evolution

As expressed above, thousands of sequences can share the same native conformation. As different types of living organisms express these proteins and due to natural selection through the course of evolution, for a given structure, sequences have a lot of diversity, with only 20% - 40% sequence identity: for two sequences with the same structure, they share on average only 20% - 40% of their amino acids. This similarity percentage may seem low, but if amino acids were drawn randomly according to a uniform distribution, sequence

[6] http://pfam.xfam.org/

identity would be only 5% (there are 20 amino acids). Therefore, there is a statistical signal in the data, and several methods have been developed to extract it.

Before introducing these different methods, we will start to discuss the various means at our disposal to represent these homologous families through an example: the class A β-lactamases. This family of proteins will accompany us until the end of this manuscript. For now, we only need to know that these proteins provide resistance β-lactams antibiotics, such as penicillin, (Neu, 1969; Kong et al., 2010). β-lactams antibiotics are largely used to treat pneumococcal infections (McLinn and Williams, 1996).

5.2.1 Multiple Sequence Alignment

For a set of P sequences of a given family of homologous, sequences have various lengths (*i.e.*, number of amino acids). For example, for our 817 sequences of class A β-lactamases extracted from (Philippon et al., 2016, 2019) coming from 739 different species: the mean length is equal to 299 with a standard deviation of 12. These sequences can be put into a matrix a_i^p called Multiple Sequence Alignment (MSA), with P lines and L columns (Fig. 5.1(a)). In a given family, two sequences may differ due to substitution (mutation of amino acid), insertion (of amino acid), or deletion (of amino acid). To deal with deletion, gaps are introduced (symbol '-') and act as an 21^{th} amino acid. To build a MSA, the sequences are aligned to be as similar as possible, *i.e.*, matching the conserved sites while penalizing the total number of gaps. Building efficiently a MSA is still a field of research, and several algorithms have been developed to tackle this problem. In our case, we used the MAFFT software[7] (Katoh et al., 2002). On average, the sequences have 37% of residues in common, with a standard deviation of 10%. Once the MSA is built, we can create its profile with a Hidden Markov model (HMM) (Durbin et al., 1998; Eddy, 1998). To do that, we used the HMMer software[8] (Finn et al., 2011). Basically, HMM give a score based on single-residue conservation. Once trained on a MSA, the profile is used for searching homologs with high score given the profile in databases. This operation allows us to enrich our dataset with sequences that were not present initially.

5.2.2 Sequence logo

Once the MSA is built, the conservation score can be computed. For the i^{th} column of the MSA, we can compute the observed frequency $f_i(a)$ of each amino acid a (including the gap). The conservation score reads:

$$C_i = \log 21 + \sum_a f_i(a) \log f_i(a), \tag{5.1}$$

where $\sum_a f_i(a) \log f_i(a)$ corresponds to the Shannon entropy (Shannon, 1948) (with a minus sign). If an amino acid is completely conserved, $C_i = \log 21$, if all amino acids are uniformly distributed, $C_i = 0$. This conservation score is used in HMM.

With this score, we can build the so-called sequence logo of the MSA. Each column of the sequence logo corresponds to a column of a MSA; amino acids are stacked with a height proportional to $f_i(a)$, the total height of the stack is equal to C_i. Sequence logo is a powerful tool for visualizing conserved sites (Fig. 5.1(b)).

[7] https://mafft.cbrc.jp/alignment/software/
[8] http://hmmer.org/

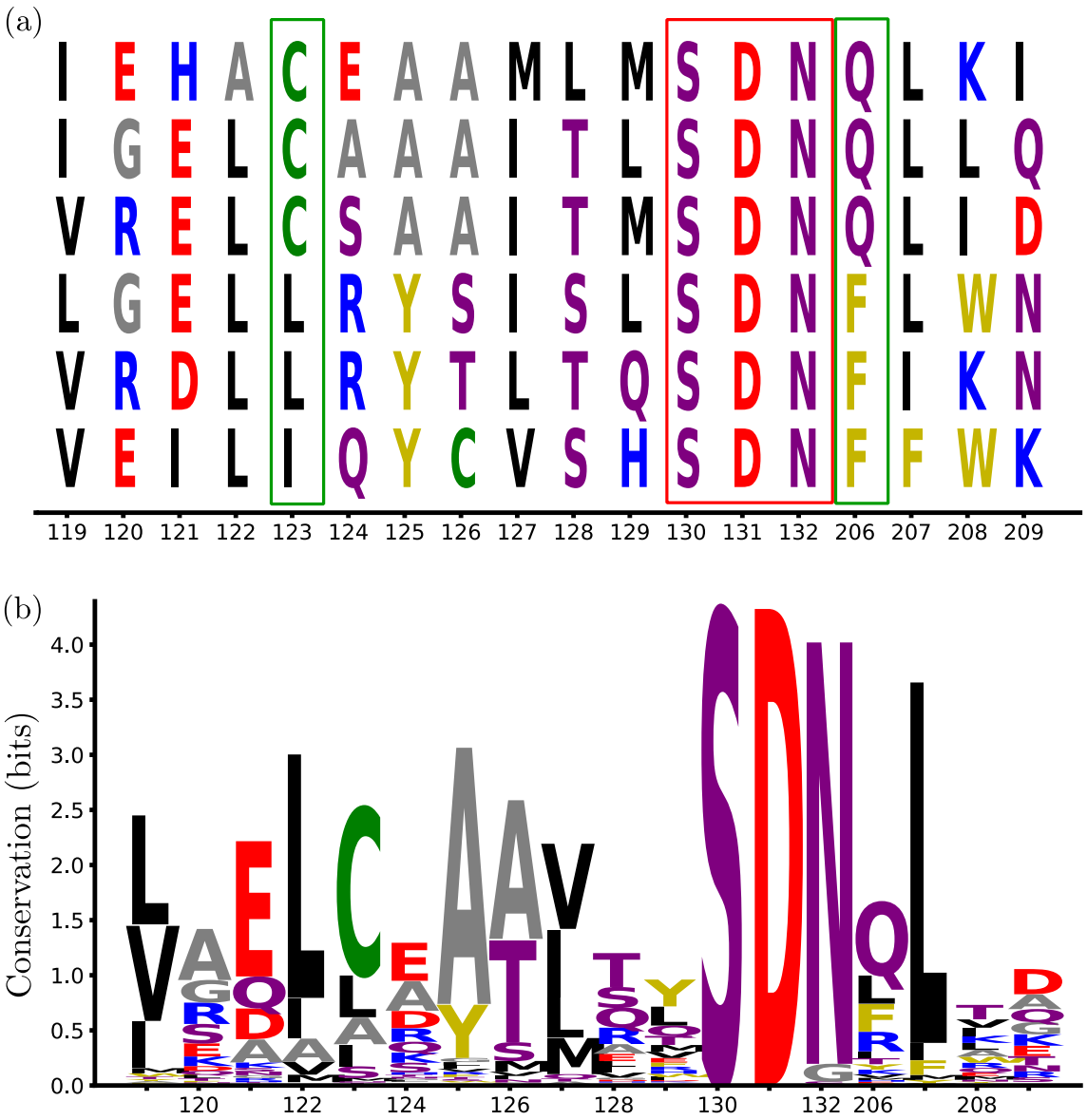


Figure 5.1: All site numbers are given in the standard Ambler numbering of β-lactamases (Ambler et al., 1991). Figures were made with a modified version of the code used by Jérôme Tubiana in his thesis (Tubiana, 2018) and available at the following link, https://github.com/jertubiana/ProteinMotifRBM. Amino acids with the same physico-chemical properties are represented with the same color. Red = negative charge (E, D), blue = positive charge (H, K, R), purple = non-charged polar (hydrophilic) (N, T, S, Q), yellow = aromatic (F, W, Y), black = aliphatic, hydrophobic (I, L, M, V), green = cysteine (C), gray = other, tiny (A, G, P). (a) 6 aligned sequences from the MSA. MSA has in total 253 columns. Here, only a subpart is represented. From site 119 to 129, an α-helix. Single and double mutations of this α-helix are studied in detail in Chapter 7. From site 130 to 132, 'SDN' loop, motif linked to catalytic mechanisms and substrate binding. From site 206 to 209, another α-helix, next to the first one in the native 3D conformation. Red frames highlight sites that are conserved. Green frames highlight coevolved residues. (b) Sequence logo for the entire MSA. SDN 'loop' is almost conserved for all sequences in the alignment.

5.2.3 Conservation and correlation

As we can see from the MSA and the sequence logo of Figure 5.1, some amino acids seem to be conserved by almost all the sequences of the class A β-lactamases. For example, in sites from 130 to 132, a specific motif called 'SDN' loop, made of Serine, Aspartic acid, and Asparagine, is highly conserved. These three sites are part of the active site of the protein (Doucet et al., 2007). Ser130 is assumed to be involved in the proton transfer from Ser70 to the β-lactam core during acylation. Crystallographic data indicate that Ser130 and Lys234 would be linked by a hydrogen bond, connecting the protein's two domains, and contribute to stabilizing the active site. These three sites are therefore of major importance for the functionality of the protein. If these sites were not important for functionality because of mutations through evolution, we should observe variability at these sites because of mutations through evolution. Here, this is not the case. For a protein to belong to class A β-lactamases, it must have this 'SDN' loop at sites 130 to 132.

We can also observe higher-order correlations between amino acids, for example, between the 123 and 206 sites. If a cysteine is present at site 123, a glutamine is present at site 206. If an aliphatic amino acid (isoleucine or leucine) is at site 123, a phenylalanine is at site 206. These patterns create second-order correlations, $f_{ij}(a, b)$, between amino acid a at site i and amino acid b at site j. Interestingly, sites 123 and 206 are close in the native conformation of the protein. The main assumption between co-evolution and structure is the following: sites which coevolve are more likely to be coupled from a functional point of view and thus close in the protein's tertiary structure. Therefore, it would be possible to extract the structure of a protein from a MSA.

Chapter 6 details the different methods used to extract information on the protein structure from MSA. We will use these methods in Chapter 7 to predict the effects of single and double mutations on an α-helix of TEM-1.

6

Protein structure and fitness prediction

This chapter discusses different methods used to predict the structure of a protein family from its MSA. For more details, we invite the reader to read the review of Cocco et al. (2018).

6.1. Correct for finite size and bias in MSA: reweighting and pseudocount

Before detailing the different methods used, we start by defining the frequency $f_i(a)$ and the second-order correlation $f_{ij}(a,b)$ introduced in Chapter 5. Furthermore, we present two useful corrections to compensate for the finite size and bias in MSA traditionally used during inference.

For a MSA with P sequences of length L, *i.e.*, a P times L matrix a_i^p, we define

$$f_i(a) = \frac{1}{P_{\text{eff}}} \sum_{p=1}^{P} w_p \, \delta_{a,a_i^p}, \tag{6.1}$$

$$f_{ij}(a,b) = \frac{1}{P_{\text{eff}}} \sum_{p=1}^{P} w_p \, \delta_{a,a_i^p} \, \delta_{b,a_j^p}, \tag{6.2}$$

where w_p is a weight and $P_{\text{eff}} = \sum_{p=1}^{P} w_p$ is the effective sequence number (Morcos et al., 2011). Note that if $w_p = w$, $f_i(a)$ and $f_{ij}(a,b)$ are simply the mean and the two point correlation of the MSA. The idea behind this reweighting procedure is to try to eliminate possible phylogenetic bias in the data (Wollenberg and Atchley, 2000; Tillier and Lui, 2003). Indeed, what interests us is the evolution of proteins across many species with a given functionality. Therefore, we do not want this signal to be biased by over-representing one species over another.

There are typically two kinds of biases. First, MSA is made of homologous sequences that have a common ancestor. If an ancestor of several sequences is recent (in terms of evolution), these sequences can be very similar because they have not had time to evolve. Second, proteins are more often sequenced in some species than in others. In fact, some species, such as *Escherichia coli*, drosophila, or zebrafish, have been much more studied by biologists than species that are difficult to access, such as species living submarine hydrothermal vent (*e.g.*, *Thermococcus gammatolerans*). These causes create biases in the MSA, and thus in the statistics extracted from the MSA. The idea behind reweighting is to give a lower weight to close sequences (in terms of Hamming distance). w_p is defined as the inverse number of sequences at a distance less than xL from the sequence $\mathbf{a}^p$. Typically,

choosing $x = 0.2$ was found to be optimal across several protein families (Morcos et al., 2011).

Another useful MSA regularization is the so-called pseudocount

$$f_i(a) \leftarrow (1-\alpha) f_i(a) + \frac{\alpha}{q}, \tag{6.3}$$

$$f_{ij}(a,b) \leftarrow (1-\alpha) f_{ij}(a,b) + \frac{\alpha}{q^2}, \tag{6.4}$$

where $q = 21$ corresponds to the number of amino acids. Pseudocount is used to limit the undersampling of proteins in a given family due to the finite size of the MSA. This transformation corresponds to adding to the MSA proteins drawn randomly according to a uniform distribution; it corresponds to a Dirichlet prior (Cocco et al., 2018). Pseudocount is useful for mean-field inference on MSA (Barton et al., 2014).

Theoretically, reweighting and pseudocount should vanish when the size P of the MSA is going to infinity.

6.2. Mutual information

Twenty-five years ago, one of the first scores used to extract structural information from the MSA was based on mutual information (Göbel et al., 1994)

$$MI_{ij} = \sum_{a,b} f_{ij}(a,b) \log \left(\frac{f_{ij}(a,b)}{f_i(a) f_j(b)} \right). \tag{6.5}$$

In information theory, mutual information is a measure of the mutual dependence between two variables. It corresponds to the Kullback-Leibler divergence between the joint distribution $f_{ij}(a,b)$ and the product of the marginals $f_i(a) f_j(b)$. $MI_{ij} \geq 0$, and $MI_{ij} = 0$ if and only if $f_{ij}(a,b) = f_i(a) f_j(b)$. Sites i and j which coevolve are likely to have a high mutual information MI_{ij}, as the joint distribution $f_{ij}(a,b)$ is not explained by the product of the marginals $f_i(a) f_j(b)$. Thus, this metric was used to predict sites that are in contact. However, using this metric leads to many false positives (Morcos et al., 2011), *i.e.*, pairs of sites predicted to be in contact but are actually far apart in the structure.

The main reason for this result is that a strong correlation between two sites is not due solely to their proximity. It is a well-known result in statistical physics: local couplings can create long-distance correlations. Take for example the famous 2D Ising model on $\mathbb{Z}^2$ with nearest neighbors couplings. At the critical temperature, the correlation length diverges (Onsager, 1944). Roughly speaking, an important correlation between sites i and j may be caused by a third-party site (Fig. 6.1(b)). Therefore, decoupling the correlations due to direct couplings from indirect couplings is necessary to improve predictions. Direct Coupling Analysis (DCA) (Weigt et al., 2009; Morcos et al., 2011) has made a significant improvement in that direction.

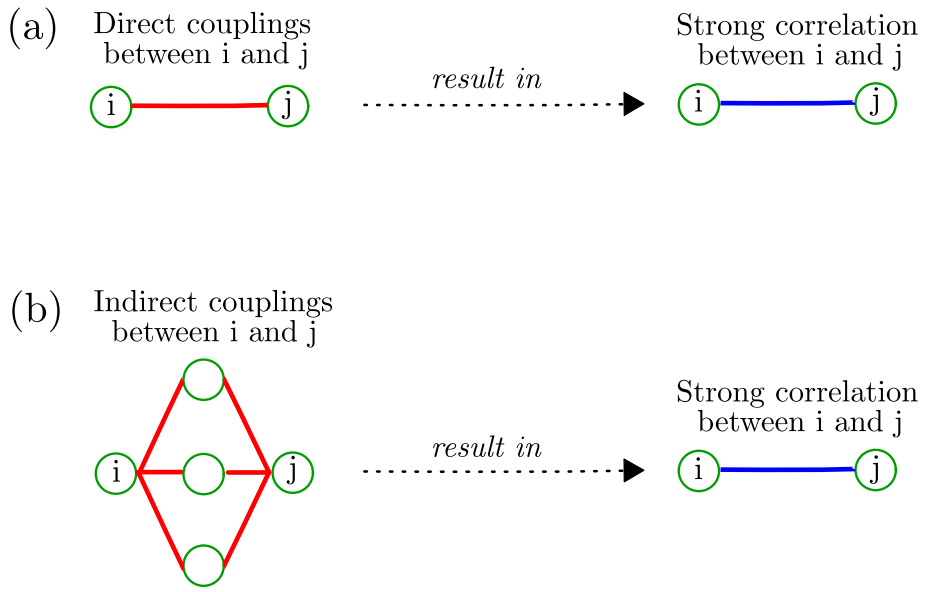


Figure 6.1: (a) Direct couplings between sites i and j result in strong correlation between them. (b) Indirect couplings between sites i and j result in strong correlation between them.

6.3. Direct Coupling Analysis

Unlike mutual information where the largest correlations are directly extracted from the MSA, the idea of DCA is to learn a probability distribution $P(\mathbf{a}|\Theta)$ to model the MSA, where $\mathbf{a}$ is a sequence of size L and Θ a set of parameters which depends on the MSA. The support of $P(\mathbf{a}|\Theta)$ is the 21^L possible sequences of size L. By tuning $\theta \in \Theta$, we want the P sequences of the MSA $\mathbf{a}^p$ to have a high probability in the model $P(\mathbf{a}^p|\theta)$. For technical reasons, the 21 amino acids (20 natural amino acids and gap) are mapped to an integer between 0 and 20.

The advantages of modeling an MSA by a probability distribution are multiple. First, we hope to understand better the interactions between amino acids from the set of parameters Θ than from the correlations observed in the MSA: this would allow to more easily predict the structure of the protein. Second, once the parameters θ are set, we can score the 21^L sequences. If the parameters θ are set "correctly", sequences of the MSA would have a high probability. But others sequences, which are not seen in the MSA, would also have a high probability. According to the model, these sequences are just as good as the MSA sequences. Therefore, the distribution $P(\mathbf{a}|\theta)$ can be used to predict whether a mutation in a given protein will have a significant effect on its functionality (see, for example, Chapter 7). Furthermore, by sampling the distribution $P(\mathbf{a}|\theta)$, it could be possible to design proteins not seen in nature with particular functionality (Russ et al., 2020).

DCA has made it possible to advance in these two directions: it has improved the prediction of the structure from the MSA (Morcos et al., 2011) and has allowed designing new proteins with specific properties (Russ et al., 2020), or to predict of mutations on a given protein (Figliuzzi et al., 2016).

From a theoretical point of view, modeling an MSA with a probability distribution $P(\mathbf{a}|\Theta)$ is not an easy thing. As pointed out by George Box, "all models are wrong, but some are useful" (Box, 1976). Here, the problem is twofold: choosing the set of parameters Θ and choosing the correct $\theta \in \Theta$ given the MSA. Indeed, we can model the MSA with the

empirical distribution $P(\mathbf{a}) = \frac{1}{P}\sum_{p=1}^{P}\prod_{i=1}^{L}\delta_{a_i,a_i^p}$, but it is not instructive at all.

For DCA, Potts model (Teller and Ashkin, 1943; Wu, 1982) is used to model the MSA.

$$P(\mathbf{a}|\Theta) = \frac{1}{Z(\Theta)}\exp\left(-E(\mathbf{a}|\Theta)\right) \tag{6.6}$$

$$= \frac{1}{Z(\Theta)}\exp\left(\sum_{i=1}^{L}h_i(a_i) + \sum_{1\leq i<\leq j\leq L}J_{ij}(a_i,a_j)\right), \tag{6.7}$$

where $Z(\Theta)$ is the partition function which ensures that the distribution is properly normalized, $\sum_{\mathbf{a}}P(\mathbf{a}|\Theta) = 1$, where $\sum_{\mathbf{a}}$ stands for the sum over the 21^L possible sequences. Each a_i can take $q = 21$ values. $h_i(a)$ are local fields acting on a single variable ($q \times 1$ vector) and $J_{ij}(a,b)$ are direct couplings between two variables ($q \times q$ matrix). The Potts model is a generalization of the Ising model (Ising, 1925), with pairwise couplings between fully connected spins, which can take q different values (called colors, $q = 2$ for Ising model). The lower the energy is, the more likely the sequence is. We will show briefly in the rest of this chapter how this probability distribution can be justified, how the parameters $h_i(a)$ and $J_{ij}(a,b)$ can be set, and how the contacts can be predicted from the couplings $J_{ij}(a,b)$.

6.3.1 Maximum entropy model

As we have seen in Section 5.2, amino acid conservation and correlations, seem to be important for modeling a protein family. This hypothesis has been experimentally tested for the WW domain (Sudol et al., 1995). In Russ et al. (2005), they artificially generated sequences by recombining natural sequences while respecting empirical means and two-point correlations. Many of these sequences were functional: they could fold and had the same properties as natural sequences. Therefore, it seems important that the probability distribution $P(\mathbf{a}|\Theta)$ respects these statistics:

$$\sum_{\mathbf{a}}\delta_{a_i,b}P(\mathbf{a}|\Theta) = f_i(b), \tag{6.8}$$

$$\sum_{\mathbf{a}}\delta_{a_i,b}\delta_{a_j,c}P(\mathbf{a}|\Theta) = f_{ij}(b,c). \tag{6.9}$$

Nevertheless, these criteria are insufficient to characterize a unique distribution: an infinite number of distributions satisfy these marginals (such as the empirical distribution, for example). So, how to choose a probability distribution among this infinity of distributions? Edwin Thompson Jaynes gave a possible answer in Jaynes (1957a,b): choose the least constrained distribution, *i.e.*, the one which maximizes the Shannon entropy (Shannon, 1948) (from a more philosophical point of view, we can rephrase this proposition from Occam's razor, "entities should not be multiplied without necessity"). The Shannon entropy was defined in information theory as

$$\mathcal{S} = -\sum_{\mathbf{a}}P(\mathbf{a}|\Theta)\log P(\mathbf{a}|\Theta). \tag{6.10}$$

To maximize the Shannon entropy while satisfying the constraints defined in Eqs. (6.8) and (6.9), we introduce Lagrange's multipliers λ, $h_i(a)$ and $J_{ij}(a,b)$. The optimization problem reads

$$
\begin{aligned}
A &= -\sum_{\mathbf{a}} P(\mathbf{a}|\Theta)\log P(\mathbf{a}|\Theta) + \lambda\left(\sum_{\mathbf{a}} P(\mathbf{a}|\Theta) - 1\right) \\
&+ \sum_i \sum_b h_i(b)\left(\sum_{\mathbf{a}} \delta_{a_i,b} P(\mathbf{a}|\Theta) - f_i(b)\right) \\
&+ \sum_{1\leq i<j\leq L} \sum_{b,c} J_{ij}(b,c)\left(\sum_{\mathbf{a}} \delta_{a_i,b}\delta_{a_j,c} P(\mathbf{a}|\Theta) - f_{ij}(b,c)\right). \qquad (6.11)
\end{aligned}
$$

By differentiating the previous equation with respect to $\mathbf{a}$, the optimal probability distribution found is the one defined in Eq. (6.6). This justifies the choice of the Potts model to model the MSA. Now that this choice is justified, we will see how to determine the fields and couplings from the MSA.

6.3.2 Inverse statistical problems

Finding fields and couplings from the MSA falls into the category of inverse problem. Schematically, from a set of observations, in our case the frequencies and the two-point correlations, we want to find the causal factors, here the fields and couplings, that produced them.

Usually, in statistical physics, our focus is on direct/forward problems. For example, when we studied Alternating Gibbs Sampling in Chapter 3, we first defined the model and then calculated observables. In our case, observables were the free energy barriers and the characteristic times associated with Alternating Gibbs Sampling.

Nevertheless, we have already tackled inverse problems in this manuscript, but we have not presented it in this way. Indeed, training an RBM solves an inverse problem: finding the optimal RBM parameters to reproduce the training data.

As we will see, to find the optimal parameters, we will maximize a log-likelihood as in the case of RBM. As for the RBM, since the probability distribution of the Potts model is also a Boltzmann distribution, we will not be able to exactly maximize the log-likelihood numerically because of our inability to evaluate the partition function. Therefore, we will have to resort to approximations to solve this maximization problem.

First, we can write the log-likelihood as

$$
\begin{aligned}
\mathrm{LL}(\Theta) &= \frac{1}{P_{\text{eff}}}\sum_{p=1}^{P} w_p \log P(\mathbf{a}^p|\Theta) \qquad (6.12) \\
&= \sum_i \sum_a h_i(a) f_i(a) + \sum_{1\leq i<\leq j\leq L} \sum_{a,b} J_{ij}(a,b) f_{ij}(a,b) - \log Z(\Theta).
\end{aligned}
$$

The first two terms correspond to the minus average energy on the data, and the last term to the minus free energy of the model. Therefore, the sum of three terms corresponds to minus an entropy: log-likelihood maximization is often called cross-entropy minimization in the statistical physics community. We can also reformulate this problem as the minimization of the Kullback-Leibler divergence between the empirical distribution and the Potts model distribution.

By writing the gradients with respect to $h_i(a_i)$ and $J_{ij}(a_i, a_j)$

$$\frac{\partial \mathrm{LL}(\Theta)}{\partial h_i(a)} = 0 \implies f_i(a) = \langle \delta_{a,a_i} \rangle_{P(\mathbf{a}|\Theta)}, \tag{6.13}$$

$$\frac{\partial \mathrm{LL}(\Theta)}{\partial J_{ij}(a,b)} = 0 \implies f_{ij}(a,b) = \langle \delta_{a,a_i} \delta_{b,b_j} \rangle_{P(\mathbf{a}|\Theta)}, \tag{6.14}$$

where $\langle . \rangle_{P(\mathbf{a}|\Theta)}$ denotes the expected value over the Potts model. These moment-matching conditions are exactly those imposed in the equations (6.8) and (6.9). Unfortunately, these expected values cannot be calculated numerically, as they involve the summation of q^L terms. As for the training of RBM, approximate methods have been developed to solve this problem. Here is a non-exhaustive list of the different methods used in practice: Boltzmann learning (Ackley et al., 1985), Pseudolikelihood maximization (PLM) (Ekeberg et al., 2013, 2014), mean-field (Roudi et al., 2009; Morcos et al., 2011), message-passing algorithm (Pearl, 1982; Weigt et al., 2009), Adaptive Cluster Expansion (ACE) (Cocco and Monasson, 2011, 2012; Barton et al., 2016). We will discuss the two first methods (that we used in our training). For more details, the reader can refer to the following reviews (Nguyen et al., 2017; Cocco et al., 2018).

- **Boltzmann learning**: as we have seen for the training of RBM (Section 1.3), Markov chain Monte Carlo methods can be used to evaluate the expected values $\langle . \rangle_{P(\mathbf{a}|\Theta)}$, and log-likelihood can be maximized with gradient ascent:

$$h_i^{t+1}(a) = h_i^t(a) + \eta \left(f_i(a) - \langle \delta_{a,a_i} \rangle_{P(\mathbf{a}|\Theta)} \right), \tag{6.15}$$

$$J_{ij}^{t+1}(a,b) = J_{ij}^t(a,b) + \eta \left(f_{ij}(a,b) - \langle \delta_{a,a_i} \delta_{b,a_j} \rangle_{P(\mathbf{a}|\Theta)} \right). \tag{6.16}$$

  This method is much less computationally intensive than the exact evaluation of the partition function, but it still has a high computational cost. Many variants of this algorithm have been used to improve the learning: in Sutto et al. (2015), gradient ascent with Nesterov momentum is used (Nesterov, 2004), and in Haldane et al. (2016), quasi-Newton method is used to approximate the Hessian (Dennis and Moré, 1977). These algorithms are used for protein families with up to $L = 200$ sites.

- **Pseudolikelihood maximization**: this approximation is based on considering L independent single-variable problems, each conditioned to the value of the $L-1$ others spins. The log-likelihood (Eq. (6.12)) is replaced by:

$$\mathrm{LL}^{\mathrm{PLM}} = \frac{1}{P_{\mathrm{eff}}} \sum_{i=1}^{L} \sum_{p=1}^{P} w_p \log P(a_i^p | \mathbf{a}_{-i}^p). \tag{6.17}$$

  $P(a_i^p | \mathbf{a}_{-i}^p)$ is a distribution over a single-variable: contrary to the initial log-likelihood, where the evaluation of the partition function requires the evaluation of q^L , here only $O(qLP)$ evaluations are required. Although $\mathrm{LL}^{\mathrm{PLM}}$ is not an approximation of the initial log-likelihood, in the limit of infinite data sampled from a Potts model, the true fields and couplings are recovering by maximizing $\mathrm{LL}^{\mathrm{PLM}}$ (Ravikumar et al., 2010). Even more interestingly, this result is still true if the maximization of the function is performed for the L terms of the sum independently: this algorithm can therefore be used in parallel. Nevertheless, this procedure creates asymmetry in the couplings ($J_{ij}(a,b) \neq J_{ji}(b,a)$) so in practice couplings, after the inference, $J_{ij}(a,b)$ are estimated as $\frac{1}{2}\left(J_{ij}(a,b) + J_{ji}(b,a)\right)$. This algorithm has shown good results in contact prediction and fitness evaluation (Ekeberg et al., 2014; Hopf et al., 2017).

6.3.3 Contact prediction with DCA

The main idea of the DCA is to disentangle the direct couplings from the indirect couplings. It results in a fourth rank tensor $J_{ij}(a,b)$ that takes into account the interactions between two amino acids, a and b, at two different sites, i and j. To predict contacts, information must be extracted at the site level, *i.e.*, interactions between sites i and j, regardless of the amino acids present at these sites. In the first place, direct information was used (Weigt et al., 2009; Morcos et al., 2011). Direct information is similar to mutual information: however, $f_{ij}(a,b)$ is replaced by a probability depending on the couplings $J_{ij}(a,b)$. More recently, an advance has been made in predicting contacts with DCA in using the Frobenius norm of the coupling matrix (Ekeberg et al., 2013) with an average product correction (APC) (Dunn et al., 2008):

$$F_{ij} = \sqrt{\sum_{a,b} J_{ij}(a,b)^2}, \tag{6.18}$$

$$F_{ij}^{\mathrm{APC}} = F_{ij} - \frac{\sum_l F_{il} \sum_k F_{kj}}{\sum_{k,l} F_{kl}}. \tag{6.19}$$

APC was introduced to suppress effects from phylogenetic biases.

6.3.4 Regularization of the Potts model

As we have seen in Section 6.1, reweighting and pseudocount can be used to eliminate biases of data. In our particular case of class A β-lactamase ($L = 253$), about 1.410^7 must be inferred when training the Potts model, while about 10^4 sequences are available in the MSA (after the enrichment on the Uniprot database). To avoid overfitting, the model must therefore be regularized. Terms are added to the log-likelihood (Eq. (6.12)) to ensure regularization ($\mathrm{LL}(\Theta) \to \mathrm{LL}(\Theta) - \Delta\mathrm{LL}(\Theta)$). Several regulation schemes are commonly used:

- L_1 regularization, corresponding to a Laplacian prior, favors sparse networks, forcing some parameters to be exactly null

$$\Delta\mathrm{LL}(\Theta) = \gamma_h \sum_{i=1}^{L} |h_i(a_i)| + \gamma_J \sum_{1 \le i < \le j \le L} |J_{ij}(a_i,a_j)|. \tag{6.20}$$

- L_2 regularization, corresponding to a Gaussian prior, penalizes large absolute value of parameters

$$\Delta\mathrm{LL}(\Theta) = \gamma_h \sum_{i=1}^{L} h_i(a_i)^2 + \gamma_J \sum_{1 \le i < \le j \le L} J_{ij}(a_i,a_j)^2. \tag{6.21}$$

Another possible regularization scheme, called color compression, has been introduced recently (Rizzato et al., 2020). Instead of having a fixed number of colors for all sites ($q = 21$), each site i has its own number of colors ($q_i \le 21$): the amino acids the least present in the data at a given site ($f_i(a) < f_0$, where f_0 is a threshold), are grouped into a unique Potts state q', with an empirical frequency $f_i(q') = \sum_a f_i(a) H(f_0 - f_i(a))$, where $H(.)$ denotes the Heaviside step function. The Potts model is inferred on these compressed

MSA, which reduces its number of parameters. After the inference procedure, the model can be decompressed to obtain all fields and couplings.

In our work on predicting epistatic effects on TEM-1 β-lactamase presented in Chapter 7, we use L_2 regularization as well as color compression.

6.3.5 Gauge invariance

The Lq frequencies $f_i(a)$ and the $\frac{1}{2}L(L-1)q^2$ correlations $f_{ij}(a,b)$ are not independent. Indeed, $\forall i, \sum_a f_i(a) = 1$ and $\forall i,a, \sum_j \sum_b f_{ij}(a,b) = f_i(a)$. Therefore, only $L(q-1) + \frac{1}{2}L(L-1)(q-1)^2$ parameters are independent. Due to this overparametrization, there is gauge invariance. $P(\mathbf{a})$ is unchanged, and consequently all its marginals, by the following change of variables:

$$h_i(a) \leftarrow h_i(a) - g_i + \sum_{j\neq i}(K_{ij}(a) + K_{ji}(a)), \tag{6.22}$$

$$J_{ij}(a,b) \leftarrow J_{ij}(a,b) - K_{ij}(a) - K_{ji}(b) + c_{ij}, \tag{6.23}$$

where $K_{ij}(a)$, g_i and c_{ij} are arbitrary functions. Two given gauges are used:

- **Zero-sum gauge**: $\forall i,j,a, \sum_b h_i(b) = 0$ and $\sum_b J_{ij}(a,b) = 0$. This gauge is used to calculate the Frobenius norm (Eq. (6.18)) and therefore is useful for contact prediction, as it is the gauge in which this norm is minimal. The transformation to obtain this gauge is as follows,

$$\begin{aligned} h_i(a) &\leftarrow h_i(a) - \frac{1}{q}\sum_b h_i(b) + \sum_{j\neq i}\left(\frac{1}{q}\sum_b J_{ij}(a,b) - \frac{1}{q^2}\sum_{b,c} J_{ij}(c,b)\right) \\ J_{ij}(a,b) &\leftarrow J_{ij}(a,b) - \frac{1}{q}\sum_c J_{ij}(a,c) - \frac{1}{q}\sum_c J_{ij}(c,b) + \frac{1}{q^2}\sum_{c,d} J_{ij}(c,d) \end{aligned}$$

- **Reference gauge**: $\forall i,j,a, h_i(\tilde{a}_i) = 0, J_{ij}(a,\tilde{a}_j) = 0$ and $J_{ij}(\tilde{a}_i,a) = 0$. This gauge sets the energy of a sequence of references to zero ($E(\tilde{a}) = 0$). This gauge is useful for predicting the effects of mutations with respect to a given sequence. The transformation to obtain this gauge is as follows

$$h_i(a) \leftarrow h_i(a) - h_i(\tilde{a}_i) + \sum_{j\neq i}\left(J_{ij}(a,\tilde{a}_j) - J_{ij}(\tilde{a}_i,\tilde{a}_j)\right), \tag{6.24}$$

$$J_{ij}(a,b) \leftarrow J_{ij}(a,b) - J_{ij}(a,\tilde{a}_j) - J_{ij}(\tilde{a}_i,b) + J_{ij}(\tilde{a}_i,\tilde{a}_j). \tag{6.25}$$

6.3.6 DCA and co-evolution: achievements and limits

Co-evolution within a given protein family appears to be important in determining the structure and functionalities of proteins. Recently, this finding has been used to determine the structure of proteins experimentally in a new way. Thanks to recent advances in sequencing, it is now possible to make all single and double mutants of a given protein (provided it is short) and measure these mutants' activities in a single experiment. This method is called deep mutational scanning (Fowler and Fields, 2014) allows identifying the sites that coevolve experimentally to deduce the protein structure (Schmiedel and Lehner, 2019; Rollins et al., 2019). Nevertheless, not all effects of a protein can be predicted with co-evolution. Some higher-order interactions appear to be crucial for proteins (Weinreich et al., 2013, 2018; Poelwijk et al., 2019; Yang et al., 2019).

From a theoretical point of view, DCA has shown its ability to predict contacts (Morcos et al., 2011), to design new proteins (Russ et al., 2020), to predict mutations' effects on a given protein (Figliuzzi et al., 2016; Hopf et al., 2017), to predict protein−protein interactions (Bitbol et al., 2016; Marmier et al., 2019) or to align sequences (Talibart and Coste, 2021). Contact predictions are now implemented in most molecular dynamics software, and have improved the predictions of these models (Ovchinnikov et al., 2016, 2017; Zhang et al., 2018).

Potts models relative simplicity, which use only fields and couplings, can encode a wide variability of effects, such as collective modes of amino acids at more than two sites. Nevertheless, it isn't easy to extract this information from the couplings, which makes their study particularly complex. Several techniques have emerged from analyzing and detecting collective modes of amino acids. Some, directly on the MSA, such as statistical coupling analysis (SCA) (Lockless and Ranganathan, 1999; Halabi et al., 2009; Rivoire et al., 2016), based on the spectral decomposition of a reweighted correlation matrix of the MSA. The idea behind this technique is to identify independently evolving subgroups of amino acids, called "sectors". In class A β-lactamases, SCA finds two different sectors that form physically contiguous structural units (Rivoire et al., 2016). RBM are effective in detecting important modes in several protein and RNA families (Tubiana, 2018; Tubiana et al., 2019a,b; Bravi et al., 2021b). We will use RBM in Chapter 9 to analyze class A β-lactamases.

In recent years, models developed in deep learning have been used to predict the structure of proteins, generate proteins or assist in experiments. The following section briefly describes these different advances.

6.4. Some advances with deep neural networks

The biggest and most exciting advance was made by Deep Mind at Critical Assessment of protein Structure Prediction[1] 13 and 14 (CASP 13 and 14) (Moult et al., 1995; Kryshtafovych et al., 2019), with respectively AlphaFold and AlphaFold 2. CASP is a biennial structure prediction competition regrouping more than 100 research groups. In CASP13 (Senior et al., 2020) and even more in CASP14 (Jumper et al., 2021), DeepMind has made tremendous progress in structure prediction using deep neural networks. Their models are trained in a supervised way on structures extracted from Protein Data Bank. In Senior et al. (2020), given a sequence and a MSA of its family, they used a ResNet (He et al., 2015), a convolutional neural using skip connections between layers intended to reproduce pyramidal cells in the cerebral cortex, to predict a distance matrix between pairs of amino acids and dihedral angles for each amino acid (more precisely, they obtain a distribution on distances and angles). They used various features as inputs to their ResNet, such as Potts models parameters trained on the MSA, the Frobenius norm of the couplings, HMM profile and other evolutionary profiles. Once trained, with this network, from a sequence and a MSA, they can predict an initial structure. After, they build a potential and minimize it by gradient descent to find the final structure. The potential is based on the negative log-likelihood of the distances and angles (predicted by the network) and on some other physical interactions between amino acids, such as van der Walls interactions (they use Rosetta software to model it). With this procedure, in both categories of CASP, the template-based modeling (TBM, where a protein with is similar sequence has a known structure) and free-modeling (FM, no homologous structure is known), AlphaFold outperformed their competitors. In CASP, the main metric used is the Global Distance Test (GDT), which ranges between 0 and 100. Schematically, this score

[1] https://www.predictioncenter.org

corresponds to the fraction of amino acids correctly predicted within a threshold distance of the experimental structure. In the free-modeling category, for CASP14, AlphaFold 2 outperformed AlphaFold, with a median GDT of 87 compared to about 60. For CASP14, Jumper et al. (2021) used an attention-based neural network (Vaswani et al., 2017), a neural network intended to mimic cognitive attention. Tunyasuvunakool et al. (2021), in a collaboration with European Molecular Biology Laboratory[2] apply this attention-based neural network at the scale of the human proteome (all the proteins in the human body) to predict the different structures. According to them, they are confident on 58% of the predicted structures, compared to 18% of the known experimental structures[3]. Another variant of the algorithm, called AlphaFold-Multimer, was recently introduced to predict protein complex (Evans et al., 2021). The use of deep networks has led to improved predictions of protein structures. Nevertheless, these techniques still rely on information from MSA and evolutionary data. They can extract information from several families and are not limited like DCA to a single indicator that is the Frobenius norm of the couplings. See Torrisi et al. (2020) for a review of the different types of networks used in practice for structure prediction. In the case of RNA structure prediction, recent progress has also been made using deep neural networks (Townshend et al., 2021).

Advances have also been made in the field of directed evolution (DE) (Chen and Arnold, 1993; Romero and Arnold, 2009)[4] DE mimics in vitro natural evolution cycle. The main objective is to produce new sequences with a given specificity. The experimental protocol is the following. From reference sequences, a library of mutants is made experimentally. Then, these mutants are selected according to a specific criterion (binding affinity, catalytic activity, fitness, ...). The best performing mutants serve as reference sequences, and the process is repeated. To create the mutants, several strategies are used, such as point mutation, insertion or deletion, or gene recombination of top mutants. These procedures are costly and time-consuming, mainly due to numerous possible mutations. To reduce these burdens, an *in silico* mutant selection phase was introduced (Cadet et al., 2018; Wu et al., 2019). In Wu et al. (2019), machine learning algorithms were trained on the experimental data up to the t^{th} round, and predict sequences which would be most likely to have a given specificity for the $(t+1)^{\text{th}}$ round of selection. This technique makes it easier to explore the space of configurations and to obtain more diverse sequences.

Deep generative neural networks, such as Variational Auto Encoders (VAE) (Kingma and Welling, 2014), Generative Adversarial Network (GAN) (Goodfellow et al., 2014), Protein Language Models (Rives et al., 2021; Meier et al., 2021) or RBM have used to predict mutations effects or generate new proteins. For predicting mutations effects (Riesselman et al., 2017; Sinai et al., 2018), VAE have comparable, though slightly better, results than predictions from Potts model. However, the distribution of VAE cannot be estimated exactly, and approximations have to be used, such as Evidence Lower Bound. One of the advantages of VAE over Potts' models is that it is possible to see in its latent space the different phylogenetic groups. On our data on the mutations of the α-helix of TEM-1 (Chapter 7), the performance of the RBM is similar to that of the DCA. RBM can however capture useful biological features of the class A β-lactamases (Chapter 9). Concerning the sampling of new proteins, GAN (Repecka et al., 2021) and VAE (Hawkins-Hooker et al., 2021) have been used in the sampling of new proteins and have demonstrated their ability to sample diverse, functional new proteins with desired properties.

[2] https://www.embl.org/

[3] https://swissmodel.expasy.org/repository/species/9606

[4] Frances Arnold's work on the use of directed evolution to engineer enzymes was awarded the Nobel Prize in Chemistry in 2018.

IV

A journey with β-lactamase TEM-1

Part IV summary

This part is dedicated to the class A β-lactamase, and more specifically to TEM-1. Thanks to an enriching collaboration with Olivier Tenaillon's group, we first studied epistasis. We show that this can be understood by a global two-state model of the protein, which can be related to the energy of a Potts model trained on a sequence alignment.

- Chapter 7 is based on our paper, *in preparation*:

  [3] Birgy, A.†, Roussel, C.†, Kemble, H., Mullaert, J., Panigoni, K., Chapron, A., Chatel, J., Magnan, M., Jacquier, H., Cocco, S., Monasson, R., Tenaillon, O., Origins and breadth of pairwise epistasis in an α-helix of β-lactamase TEM-1, *In preparation* (†: joint first authors)

Chapter 8 presents preliminary results on the influence of amoxicillin concentration on log-fitness and epistasis.

And finally, Chapter 9 presents how to use the compositional phase of RBM to isolate and interpret influential modes of amino acids of subfamilies of class A β-lactamase.

7

Pairwise epistasis in α-helix of β-lactamase TEM-1

This work is the result of a collaboration with Olivier Tenaillon's group[1] at Bichat Hospital. They performed the experiments on TEM-1 mutants, and have started to analyze the data.

We then took over the log-fitness inference procedure, the two-state model parameter inference procedure, the data analysis and the Potts model inference.

7.1. Motivations

As explained in Chapters 5 and 6, sequences of the first proteins triggered the emergence of molecular evolution and bioinformatics in the 1960s (Hagen, 2000).

Yet, more than 50 years later, despite a massive number of available protein sequences and a pressing demand from human genetic disease and synthetic biology, the prediction of nonsynonymous mutation effects, mutation that changes an amino acid of sequence, remains a challenging task.

Nonetheless, over the last decade, two independent approaches have offered new perspectives on the study of nonsynonymous mutation effects. Experimentally, protein deep mutational scans, in which the impacts of all possible single amino acid changes in a protein are investigated, have gained momentum allowing to study not only single mutants but also multiple mutants (Fowler and Fields, 2014). One way to measure the impact of these mutations is to measure the effect on the exponential growth rate of a bacterium carrying this mutant protein subjected to a selection pressure, in our case a medium with a certain concentration of amoxicillin. This growth rate, called absolute fitness W_i, reads

$$N_i(t+1) = W_i N_i(t), \tag{7.1}$$

where $N_i(t)$ denotes the population of the mutant i at time t. If $W_i > 1$, population increases over time, otherwise the population decreases. To compare the different absolute fitness, we define as reference fitness the one of the wild-type W_{WT} (here, TEM-1). The relative log-fitness $\log(w_i)$ with respect to the wild-type reads

$$\log(w_i) = \log(W_i) - \log(W_{WT}), \tag{7.2}$$

therefore, if $\log(w_i) > 0$, the mutant grows faster than the wild-type.

At the bioinformatics level, massive protein databases have allowed using multiple sequence alignment to infer the amino acids that are tolerated or not at a site. Interestingly,

[1] https://www.iame-research.center/eq1/research-interests/

experimental and data-driven approaches revealed immediately that mutation impact could vary with genetic background (Jacquier et al., 2013; Bank et al., 2015, 2016). It was for instance shown that as little as a single mutation could change quite drastically the impact of many other mutations throughout a protein (Bloom et al., 2005; Jacquier et al., 2013). These observations called for a more comprehensive understanding of mutations' effects and especially of their interactions.

Epistasis refers to the context-dependency of mutation effects. In population genetics, pairwise epistasis refers more precisely to mutation interactions that translate in non-additivity of log-fitness effects. Epistasis between mutation A and B can be quantitatively estimated as the deviation between the observed log-fitness of the double mutants, AB, and the sum of the log-fitness of both individual mutations (A and B) (Fig. 7.4(a)). Under this strict definition, epistasis has been predicted to impact significantly many facets of evolution, from the evolution of mutation rate and recombination (de Visser and Elena, 2007), to the diversity of adaptive path and the repeatability of adaptation (de Visser and Krug, 2014). These undoubtful significant consequences of epistasis now call for an integrated and mechanistic understanding of epistasis causes.

An integrated vision of epistasis may be obtained from a top-down perspective, with phenomenological models that capture its global properties. These models have shown that all forms of epistasis mentioned in Figure 7.4(a) can emerge from a simple nonlinear mapping of phenotype to fitness even if the phenotype is additive. For instance, all possible forms of pairwise epistasis are observed in the Fisher Geometric Model (Martin et al., 2007; Gros et al., 2009; Blanquart et al., 2014; Tenaillon, 2014), a smooth singled peaked phenotypic landscape in which fitness is a Gaussian function of the distance to an optimum phenotype. These observations motivated the research of an underlying simple phenotype that could explain globally the pattern of epistasis observed. Accordingly, statistical analysis of large datasets of multiple mutants have revealed epistasis to be largely described by an underlying additive phenotype (Otwinowski et al., 2018).

As proteins generally operate in a folded state, mutations' impacts on protein have mainly been investigated through their effects on that fold or its affinity with a substrate. For epistatic interactions, two mutually non-exclusive mechanistic visions have emerged. With compensatory mutations, characterized by two independently deleterious mutations that, when combined, outcompete at least a single mutant, the idea of key-lock local interactions suggested itself. Alternatively, the existence of mutations with a global impact on protein stability (Bloom et al., 2005) hinted that the cooperative nature of protein stability could also result in epistatic effects, this time at a more global level (Wylie and Shakhnovich, 2011). The extent of both types of interactions and the overall prevalence of epistatic interactions remain however unclear.

To investigate the molecular determinant of epistatic interactions, Olivier Tenaillon's group generated a comprehensive library of more than 15,000 single and double mutants within an α-helix of β-lactamase TEM-1. TEM-1 is a highly successful antibiotic resistance gene present in about 35% of *Escherichia coli* natural isolates (EARS-Net France). We focused on an 11 amino acid α-helix, from residue 119 to 129 (Fig. 7.4(b)), as α-helices are the most characterized and frequent secondary structure in protein folds. For the sake of generality, this α-helix is not involved in the active site; it is just a structural component of the enzyme. The mutants, which cover more than 76% of all possible double mutants, were analyzed for their impact on protein activity, measured through the minimum inhibitory concentration (MIC), and more importantly, through their effects on fitness, allowing a proper estimation of epistasis.

7.2. Aim of experiment and experimental protocol

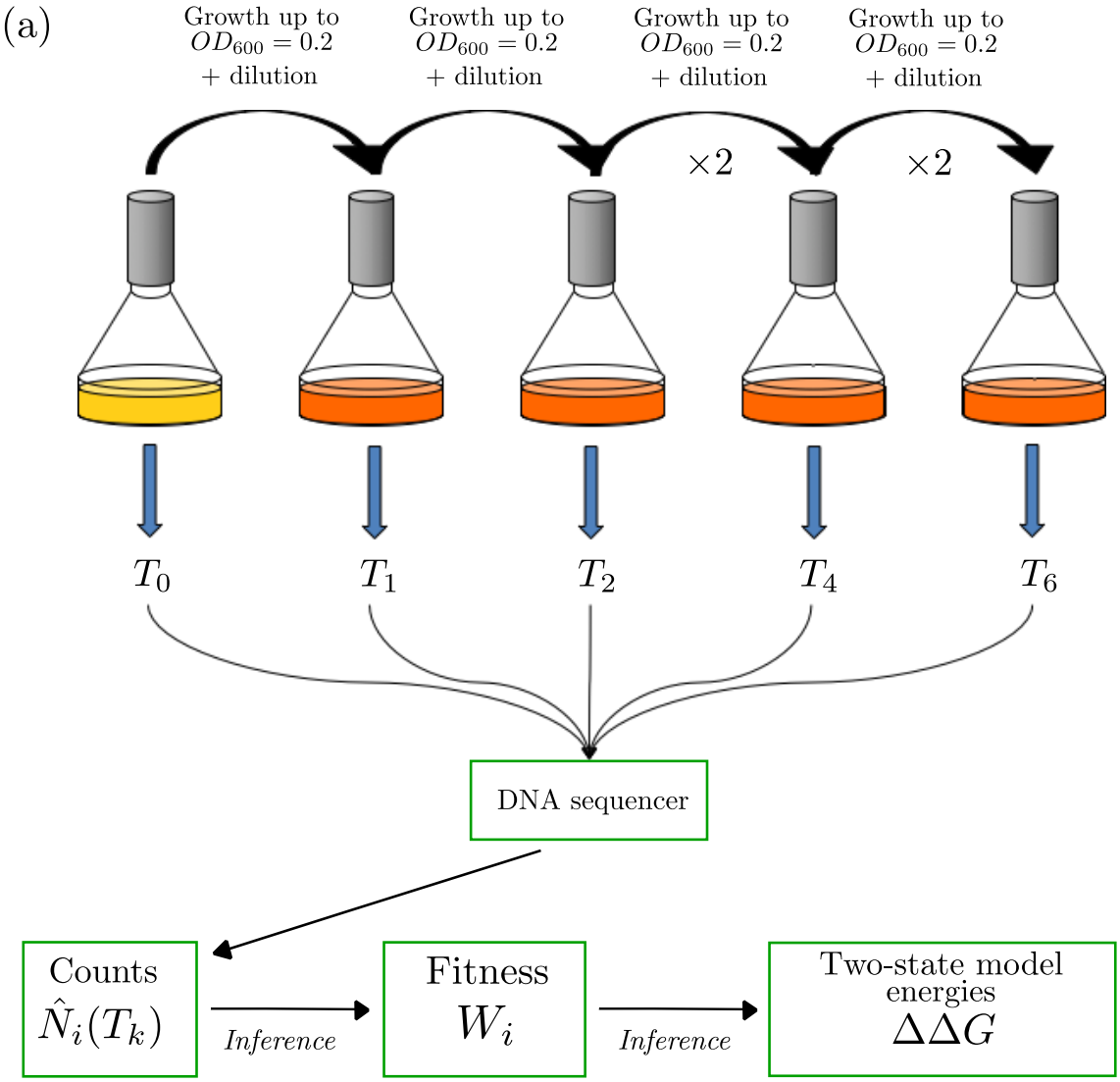
In this section, we will describe the aim of the experiment and the experimental protocol. We will try to be pedagogical, understandable for a physicist audience, while being precise. The details of this protocol are available in the Supplementary Materials of the paper.

7.2.1 Objectives

The main objective of this experiment is to measure the log-fitness of single and double mutants $\log(w_i)$ (Eq. 7.2) of an α-helix of TEM-1, in a medium with a concentration of amoxicillin of 8 g.L^{-1}.

The idea to measure the log-fitness is to confer to bacteria, here *Escherichia coli*, the ability to express one of the mutants. This mutant will bring or not to the bacterium the capacity to resist the drug, thus multiplying or not in the environment. The plasmids give the ability to express a given mutant. A plasmid is a small extrachromosomal DNA (0.5 to 5% of the chromosomal DNA) within the cell. In this experiment, plasmids encode TEM-1, and therefore, are mutated to express the wild-type's single and double mutants. In addition, these plasmids contain 20 degenerate nucleotides, called barcodes. Each barcode is assigned to a given mutant. Several barcodes can be used for the same mutant. These barcodes are read by a DNA sequencer, which allows counting the number of plasmids for a given mutant. For each mutant, what is measured is the number of plasmids encoding this mutant. In practice, as the sequencing is not perfect, a bioinformatics treatment is used to count the number of plasmids. This treatment is developed in the appendices of the paper.

The experiment we are interested in here was performed with a concentration of 8 g.L^{-1} of amoxicillin. The plasmids were sequenced at different times, called T_k from T_0 to T_6. First, bacteria are placed in an antibiotic-free medium until they reach an optical density of 0.4 at 600 nm ($OD_{600} = 0.4$), which defines the time T_0. Then, a part of this solution is taken and sequenced to obtain the number of plasmids for the different mutants at T_0. Then, another part of the solution is diluted (the dilution factor is called d). The bacteria are placed in an antibiotic medium until they reach $OD_{600} = 0.2$, which defines the time T_1. This phase of growth, measurement, dilution is repeated until time T_6. Between time T_0 and T_1, there are four population-averaged generations (the total number of plasmids is multiplied by 2^4). Between time T_k and $T_{k+1}, k \leq 1$, there are 5 population-averaged generation. The experiment is depicted in Fig. 7.1(a).



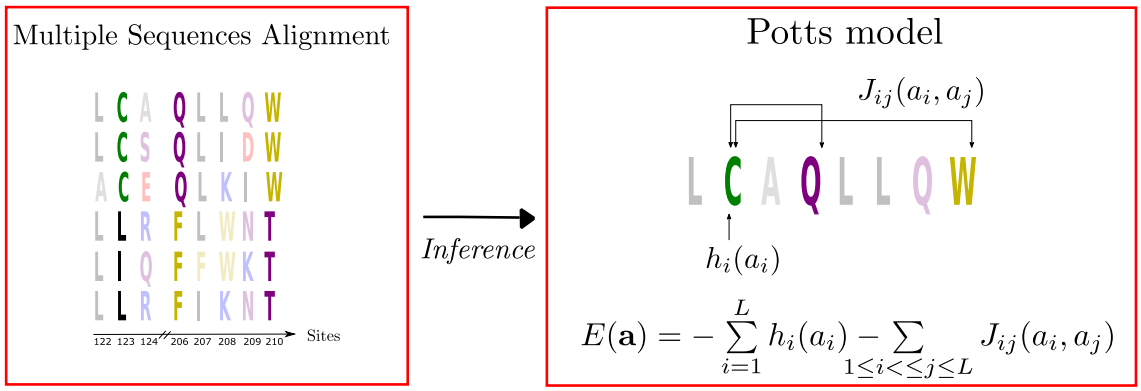


Figure 7.1: (a) Scheme of the experiment and inference from experimental data. From mutations to fitness W_i and two-state model energies $\Delta\Delta G$. (b) Inference of Potts model from multiple sequence alignment, from sequences to Potts energies $E(\mathbf{a})$.

7.3. Inference procedure of the log-fitness

In this section, we detail the inference procedure we developed to estimate the log-fitness of each mutant. We denote as $\hat{N}_i(T_k)$ the number of plasmids measured at time T_k carrying the mutation i (the different barcodes encoding for one given mutation are grouped together).

7.3.1 Modelization of the DNA sequencer

To infer the fitness of each mutation i, we modeled the evolution of the number of plasmids over time. $N_i(T_k)$ denotes the total population of plasmids carrying the mutation i at time T_k

According to the definition of the absolute fitness, $N_i(T_k)$ follows an exponential growth (Eq. 7.1). However, we do not have access directly to $N_i(T_k)$, the true population, but to $\hat{N}_i(T_k)$, the result of the measurement by the DNA sequencer.

According to the experimental protocol, at each time T_k, due to the different dilutions after each measure, the population of plasmids in the DNA sequencer at time T_k can be written as $d^k W_i^{T_k} N_i$, where d is the dilution factor ($d = \frac{1}{32}$).

The DNA sequencer does not sample all the plasmids, but samples only a fraction of them. We use the following modeling for the sampling: each measurement of $\hat{N}_i(T_k)$ is a realization of a binomial distribution $B(d^k W_i^{T_k} N_i, p_k)$, where $d^k W_i^{T_k} N_i$ is the theoretical population of plasmids in the DNA sequencer carrying the mutation i at time T_k, and p_k is the inferred sampling rate of the DNA sequencer at time T_k. We can estimate the sampling rate p_k at different times

$$p_k = \frac{\sum_i \hat{N}_i(T_k)}{N_{OD}(T_k)}, \tag{7.3}$$

where $\sum_i \hat{N}_i(T_k)$ is the total population sampled with the DNA sequencer at time T_k, and $N_{OD}(T_k)$ is the theoretical population in the DNA sequencer at time T_k. The ratio of these two quantities is an estimation of the sampling rate of the DNA sequencer. In practice, as p_k is of order 10^{-2}, $\sum_i \hat{N}_i(T_k)$ never exceeds $N_{OD}(T_k)$: during its inference, p_k is always properly defined ($p_k < 1$).

7.3.2 Time parameterization

According to the measurement protocol, the different times steps T_k correspond to a number of population-averaged generations ($T_0 = 0$, $T_1 = 4$, $T_2 = 9$, $T_4 = 19$ and $T_6 = 29$)[2]. We have the following evolution for total number of plasmids

$$N(T_k) = 2^{T_k - T_{k-1}} N(T_{k-1}). \tag{7.4}$$

However, as a significant fraction of mutants may die due to the antibiotic, the number of generations may be underestimated during a cycle. Therefore, we redefined the time scale by the number of generations the wild-type did. Within this new definition of time T^{WT}, the absolute fitness of the wild-type W_{WT} must be equal to 2. We found indeed this value with our inference procedure (Fig. 7.2).

[2]The dilution factor $d = \frac{1}{32} = \frac{1}{2^5}$ corresponds to 5 population-averaged doubling, which explains why $T_{k+1} - T_k = 5$, except for $T_1 - T_0 = 4$, as at T_0 $OD_{600} = 0.4$, and therefore after the dilution only 4 population-averaged doubling are needed to reach $OD_{600} = 0.2$.

Between T_{k-1} and T_k, we expect to have a $T_k - T_{k-1}$ population-averaged doubling. During this cycle, the wild-type has made $T_k^{WT} - T_{k-1}^{WT}$ doubling. $\hat{N}^{WT}(T_k)$ is the measured population of wild-type at time T_k. If $f(k)$ is the frequency of wild-type after the k^{th} cycle

$$f(k) = \frac{\hat{N}^{WT}(T_k)}{\sum_i \hat{N}_i(T_k)}, \tag{7.5}$$

as

$$2^{T_k^{WT} - T_{k-1}^{WT}} f(T_{k-1}) \sum_i \hat{N}_i(T_{k-1}) = f(T_k) \sum_i \hat{N}_i(T_k), \tag{7.6}$$

we get,

$$T_k^{WT} - T_{k-1}^{WT} = T_k - T_{k-1} + \log_2 \frac{f(T_k)}{f(T_{k-1})}. \tag{7.7}$$

And then

$$T_k^{WT} = \sum_{k'=1}^{k} (T_{k'} - T_{k'-1}) + \log_2 \frac{f(T_{k'})}{f(T_{k'-1})} \tag{7.8}$$

$$= T_k + \log_2 \frac{f(k)}{f(0)}. \tag{7.9}$$

The results are reported in Table 7.1.

	k = 0	k = 1	k = 2	k = 4	k = 6
T_k	0	4	9	19	29
T_k^{WT}	0	6.6	11.9	22.0	32.1

Table 7.1: Comparison between T_k and T_k^{WT}

In the following, we use T_k^{WT} as reference time scale, but we note it as T_k to simplify the notations.

7.3.3 Computation of the likelihood

For a given mutation i, we want to estimate the absolute fitness W_i knowing the measurements of the population $\{\hat{N}_i(T_k)\}_k$ at different times T_k. Within our model, the probability of $\{\hat{N}_i(T_k)\}_k$ at different times T_k knowing W_i can be written as

$$P(\{\hat{N}_i(T_k)\}_k | W_i) \propto \sum_{N_i} \prod_k \binom{d^k W_i^{T_k} N_i}{\hat{N}_i(T_k)} (1-p_k)^{d^k W_i^{T_k} N_i - \hat{N}_i(T_k)} p_k^{\hat{N}_i(T_k)}, \tag{7.10}$$

Without a priori knowledge of the distribution of W_i, using Bayes' theorem, the likelihood can be written as

$$P(W_i | \{\hat{N}_i(T_k)\}_k) \propto P(\{\hat{N}_i(T_k)\}_k | W_i). \tag{7.11}$$

The likelihood for all the mutations reads

$$P_{\text{model}}(W_i|\{\hat{N}_i(T_k)\}_k) \propto \prod_i P(W_i|\{\hat{N}_i(T_k)\}_k). \tag{7.12}$$

In the most general case, because of the binomial coefficients in equation 7.10, the exact likelihood can not be computed analytically or numerically. Nonetheless, we can use a Gaussian approximation for the likelihood. Within this approximation, we have

$$P_{\text{model}}(W_i|\{\hat{N}_i(T_k)\}_k) \quad \propto \quad \max_{\{N_i\}} \exp\left(\Phi(\{W_i, \hat{N}_i(T_k), N_i\}) - \frac{1}{2}\log|\Phi''(\{W_i, \hat{N}_i(T_k), N_i\})|\right),$$

where

$$\begin{aligned}
\Phi(\{W_i, \hat{N}_i(T_k), N_i\}) &= \sum_i \sum_k \log\left(d^k W_i^{T_k} N_i\right)\left(\frac{1}{2} + W_i^{T_k} N_i\right) \\
&- \log\left(d^k W_i^{T_k} N_i - \hat{N}_i(T_k)\right)\left(\frac{1}{2} + d^k W_i^{T_k} N_i - \hat{N}_i(T_k)\right) \\
&- \log\left(\hat{N}_i(T_k)\right)\left(\frac{1}{2} + \hat{N}_i(T_k)\right) \\
&+ \log(1-p_k)\left(d^k W_i^{T_k} N_i - \hat{N}_i(T_k)\right) + \log(p_k)\hat{N}_i(T_k).
\end{aligned} \tag{7.13}$$

The likelihood is maximized numerically with respect to W_i and N_i. In some cases, we can compute exactly the true likelihood, and our Gaussian approximation is in good agreement with the true likelihood (Fig. 7.2).

Once the parameters have been inferred, we have the following log-likelihood

$$LL(\{W_i, \hat{N}_i(T_k), N_i\}) = \Phi(\{W_i, \hat{N}_i(T_k), N_i\}) - \frac{1}{2}\log|\Phi''(\{W_i, \hat{N}_i(T_k), N_i\})|. \tag{7.14}$$

Therefore, we can estimate the uncertainty on the parameter W_i as

$$\sigma_{W_i} = \sqrt{\frac{1}{\frac{\partial^2 LL(\{W_i, \hat{N}_i(T_k), N_i\})}{\partial W_i^2}}}. \tag{7.15}$$

As by definition of the time, $W_{WT} = 2$, and $\frac{\sigma_{W_i}}{W_i} \ll 1$, the standard deviation $\sigma_{\log(w_i)}$ associated with the relative fitness is equal to $\frac{\sigma_{W_i}}{W_i}$. In practice, we estimate log-fitness only between T_0 and T_2, because at longer times, as yet not understood effects seem to disrupt the exponential growth of bacteria.

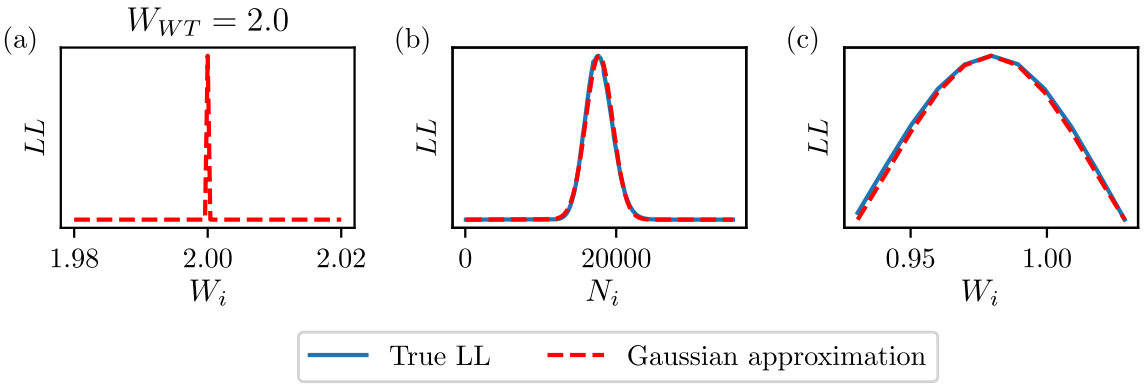


Figure 7.2: (a). Estimation of W_{WT}. $W_{WT} = 2$ in agreement with the definition of the T_k. (b) and (c). Comparison between the exact log-likelihood and the Gaussian approximation for a given mutant. (b) Comparison between the exact log-likelihood and the Gaussian approximation for the estimation of N_i. (c) Comparison between the exact log-likelihood and the Gaussian approximation for the estimation of W_i.

7.3.4 Consistency with Minimum Inhibitory Concentrations and replicas

We can compare our estimations of the log-fitness with measures of Minimum Inhibitory Concentrations (MIC)[3] through experiments at 1, 2, 4, 8, and 16 g/L of amoxicillin. MIC and log-fitness are very high correlated for both single (Spearman correlation, $\rho = 0.98$) and double mutants ($\rho = 0.76$), although in a non-linear way (Fig. D.1).

Furthermore, Olivier Tenaillon's group performed a second experiment to measure again the log-fitness (a second replicate) and for both replicates, the log-fitness of the single mutants, double mutants, and epistasis are highly correlated (respectively, r^2=1.0, r^2=0.95, r^2=0.99, Fig. D.2).

In addition, the uncertainty of the log-fitness inferred by our inference procedure as well as that estimated from the two replicates is correlated ($\rho = 0.64$ for the single mutants, $\rho = 0.58$ for the double mutants, Fig. 7.3).

7.3.5 Definition of the lethality threshold

Mutants were considered lethal under a theoretical lower threshold for log-fitness equals to $-\log(2)$. At this specific value, bacteria do not grow in the solution. In practice, to limit the noise, we used a higher threshold equals to -0.6.

[3]MIC is the lowest concentration of a drug that prevents visible growth of bacteria.

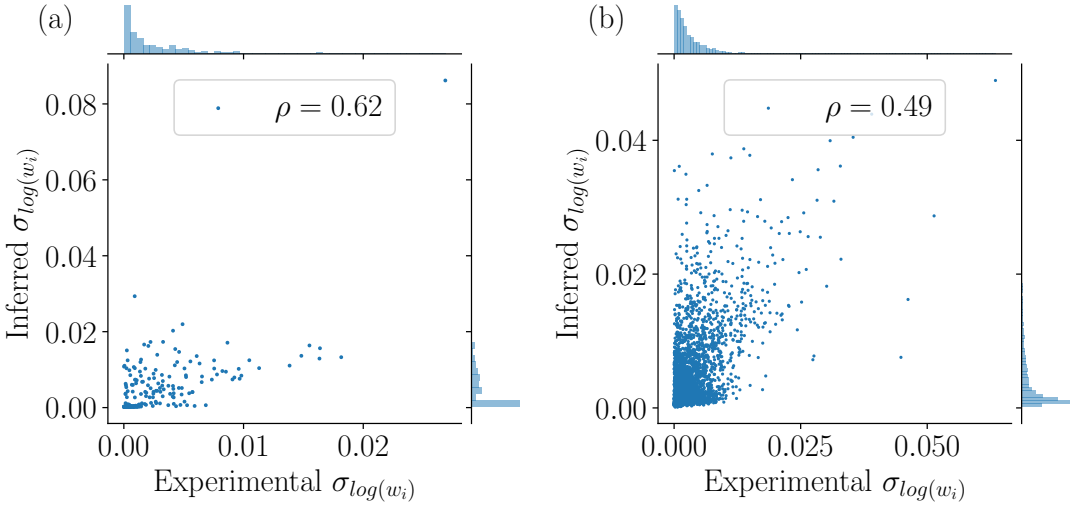


Figure 7.3: Comparison between the experimental error and the inferred error of the log-fitness. (a) For single mutants. (b) For double mutants.

7.4. Results

7.4.1 Distribution of the log-fitness and epistasis

The distribution of log-fitness effects of single mutants had a bimodal structure with close to 50% lethal mutants (log-fitness < -0.6) (Fig. 7.4(d)). This suggested an overall important role of that α-helix. The different residues had very different patterns, with four sites permissive to mutations, while the others were much more sensitive (Fig. 7.4(c)). As expected proline, which is known to be incompatible with α-helix structure (von Heijne, 1991), was lethal or close to lethal at all sites (log-fitness < -0.55) (Fig. 7.4(c)). The distribution of double mutant effects appeared to be tri-modal with an even more significant fraction of lethal genotypes (78%) (Fig. 7.4(e)). A dominance effect emerged: mutant combinations including a lethal mutation were lethal. Out of the 10.887 double mutants involving at least a lethal mutant, only 105 (1.0%) had a log-fitness higher than -0.5 (Fig. 7.5(a)). Only 2 (0.02% of total) resulted from the combination of two deleterious mutations, an instance of sign epistasis in which one of the mutations is deleterious in one background and beneficial in another. This general dominance effect clarifies the partial success of methods based on residue conservation (Ng and Henikoff, 2003; Adzhubei et al., 2013) to predict mutation effect: significant effects such as inserting a proline within an α-helix are effectively context-independent. This suggests that the key-lock epistatic compensations, characterized by two independently deleterious mutations that when combined outcompete at least a single mutant, are rare in the alpha-helix under study.

We then focused on quantifying epistasis (Figs. 7.5(a) and (b)) and noticed that double mutants' log-fitness deviated substantially from the one expected, *i.e.* the sum of log-fitness of the two single mutants.

Epistasis could be estimated with high resolution only for non-lethal double mutants with non-lethal single mutants. Restricting the dataset to these mutants, we could compute a distribution of epistasis that was both broadly distributed around zero and biased towards negative values (Fig. 7.5(b)), as observed on other experiments based on proxies of protein function rather than on true fitness, *i.e.* binding, or fluorescent protein (Sarkisyan et al.,

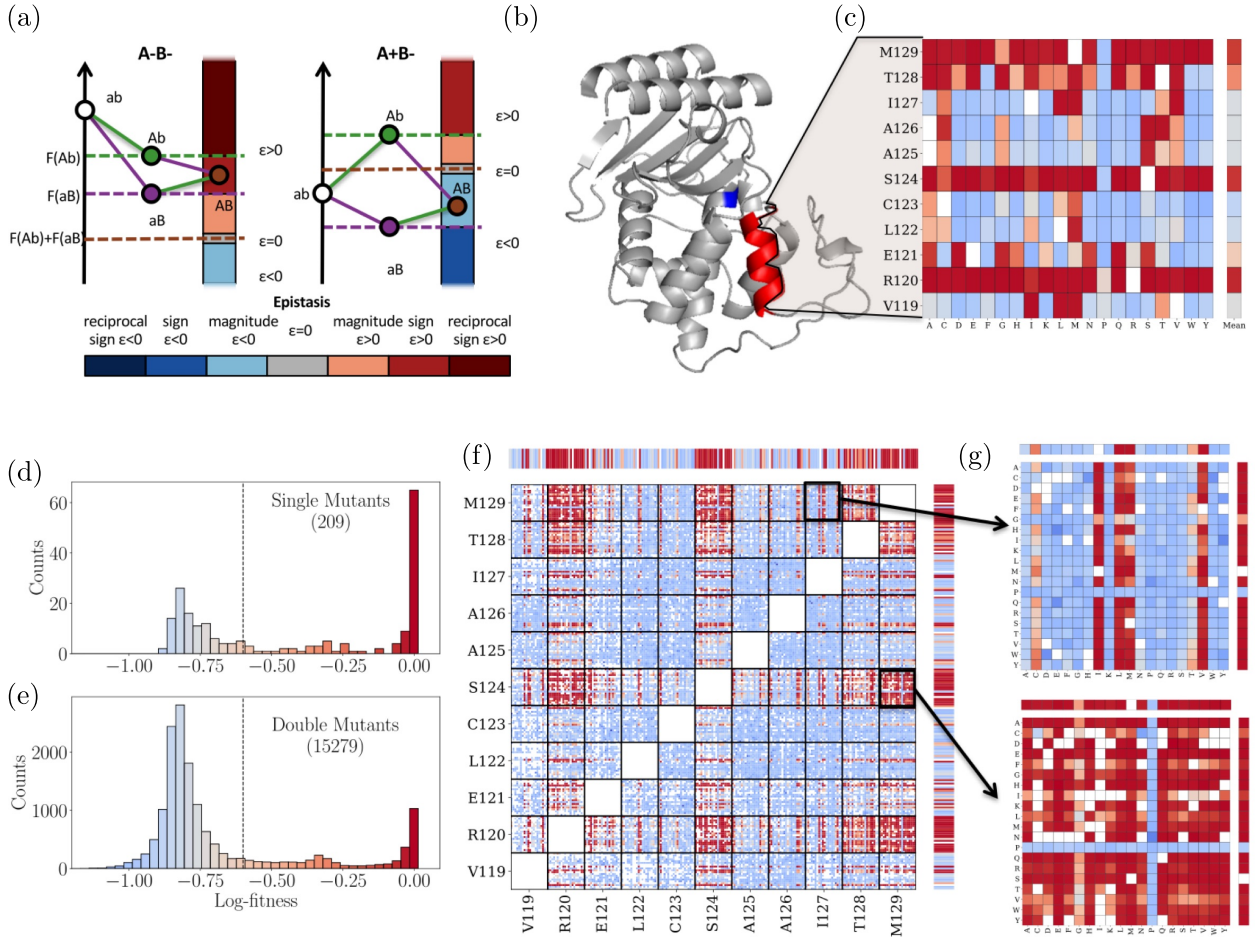


Figure 7.4: Single and double mutants' log-fitness effects. (a) Pairwise epistasis measures the deviation of the observed log-fitness of a double mutant from the sum of the log-fitness of its single constituent mutants. It can also be qualitatively categorized as magnitude, sign, and reciprocal sign as well as positive or negative. The figures illustrate how this categorization functions in the case of a pair of deleterious mutations on the left and a pair including a deleterious (b to B) and a beneficial mutation (a to A) on the right (b) 3D structure of β-lactamase TEM-1. In red the α-helix of interest, and in blue the Serine residue of the active site. (c) The effects on the log-fitness of all single mutants per residue. Color scale is given in panels (d-e). (d-e) Distribution of log-fitness effects. Below the dotted line (log-fitness = -0.6), mutants are considered non-functional. (d) For single mutants. (e) For double mutants. (f) Log-fitness of the double mutants with missing data in white. Color scale is given in panels (d-e). (g) Zoom on the double mutants log-fitness involving residues I127 and M129 on top and S124 and M129 at the bottom.

2016). Yet, some large positive epistasis were also found, especially among pairs including a beneficial mutation and a deleterious one (Fig. 7.5(c)).

We then looked at the log-fitness effect of individual mutations across all different backgrounds. For a given single mutant A, we plotted the log-fitness of the double mutants AB minus the log-fitness of the single mutant B (called focal mutation relative log-fitness) versus the log-fitness of single mutants B (called background log-fitness), see Figure 7.5(e). In this figure, the white area corresponds to mutants with high resolution on log-fitness for double mutants AB and single mutant B (log-fitness > -0.6). The blue region corresponds to lethal double mutants. And finally, the orange area corresponds to lethal single mutant

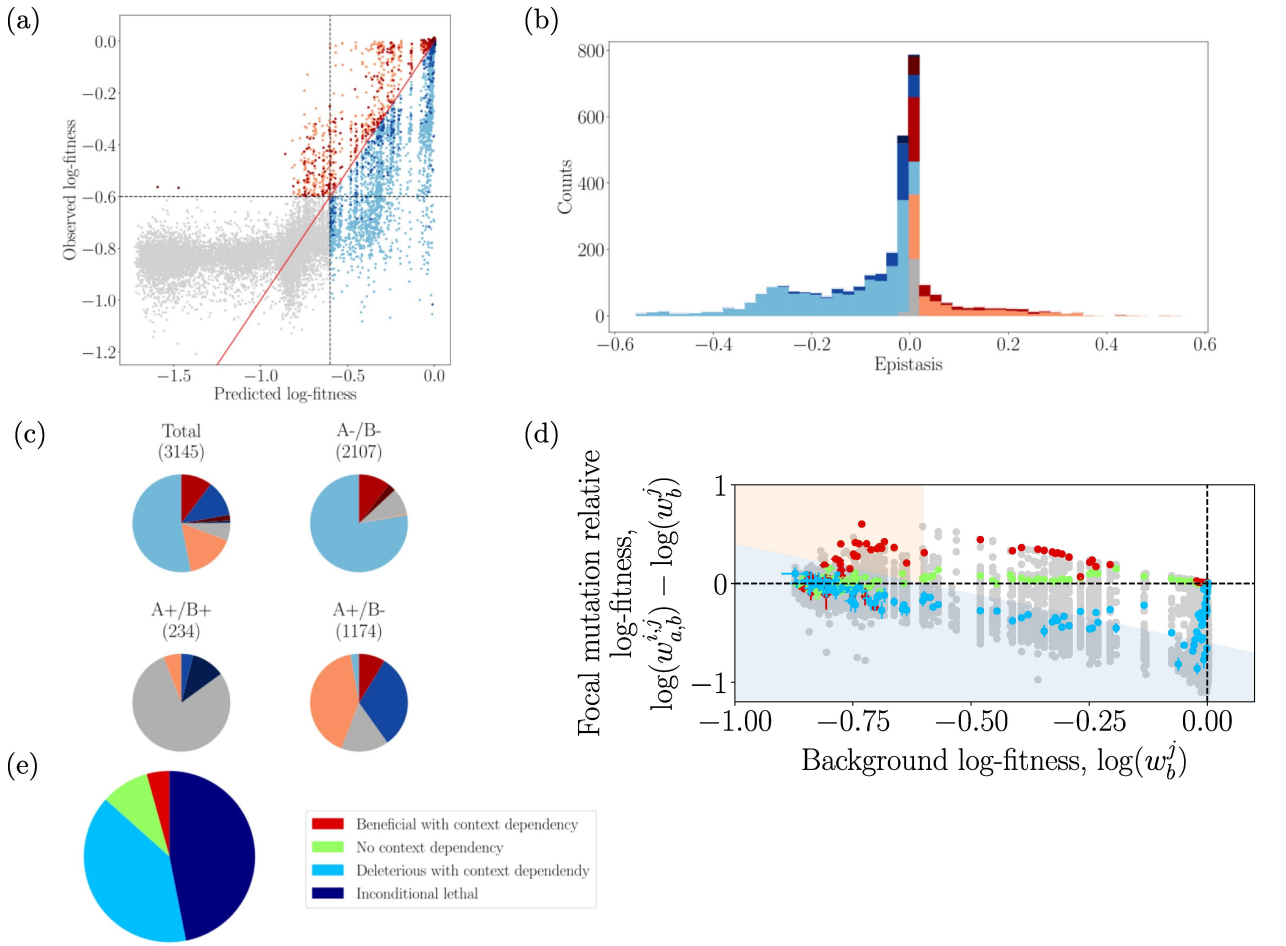


Figure 7.5: Pairwise epistasis. (a) Log-fitness of effects of double mutants, against the sum of the single mutants' log-fitness. Grey mutants of observed log-fitness and predicted log-fitness lower than -0.6 can not be used to compute epistasis. The colors of the other points represent the form of epistasis detected using the color code defined in Fig 7.4(a). (b) Distribution of epistasis using the same color code, excluding mutants with non-measurable epistasis. (c) Categorization of epistasis for all mutations, pairs of deleterious (A-/B-), pairs involving one deleterious and one beneficial (A+/B-), or pairs of beneficial (A+/B+). (d) Relative log-fitness effect of all mutations against the log-fitness of the different backgrounds in which they were found. The values for three focal mutations, L122A, R120K, and S124E, are highlighted in blue, green, and red respectively. (e) The fraction of mutations falling into unconditionally inactivating, deleterious with context-dependency, no context dependency, and beneficial with context-dependency is presented.

B but where the double mutants AB have log-fitness greater than -0.6. Due to the high resolution of log-fitness in the white area, we are mainly interested in the patterns that exhibit the mutations in this area. These plots exhibit mutations with very contrasted and structured patterns that we grouped in four distinct categories (Fig. 7.5(e)).

Among the 209 possible single mutants, 98 (47%) are lethal across all backgrounds (the single mutants and all the double mutants including these single mutants have a log-fitness lower than -0.6). Due to the resolution of our experiments, we can not say so much about them. 83 (40%) single mutants are deleterious mutations, *i.e.* have negative epistasis, see for example blue points (Fig. 7.5(d)). 19 (9%) mutations showed an overall context-independent mutation effect, *i.e.* have no epistasis, which correspond to a straight

line with null slope in Figure 7.5(d), see for example green points (Fig. 7.5(d)). These mutants had a minor impact on log-fitness (less than a 1% effect on log-fitness). Finally, 9 (4%) mutations with marginally increased log-fitness effects in the ancestral background, *i.e.* have positive epistasis, see for example red points (Fig. 7.5(d)).

Strikingly, excluding the 98 mutations that were lethal in all backgrounds, 83% of the mutations exhibited some strong form context dependencies that were structured by background log-fitness. A majority of double mutants AB associated with a given single mutant A exhibit either positive epistasis or null epistasis or negative epistasis, but not all three at once. This consistency suggests a macroscopic force at play, such as protein stability.

7.4.2 Protein two-state model

One paradigm in protein analysis is that most residues in protein maintain the functional fold, and therefore mutations at these sites mainly alter its stability but not the activity (DePristo et al., 2005). Protein stability can be represented by a two-state model, corresponding to a functional folded state and to several nonfunctional unfolded states (Privalov, 1979; Wylie and Shakhnovich, 2011) (Fig. 7.6(a) and (b)). The fraction of time spent in the functional fold reads

$$P_{\text{nat}} = \frac{1}{1+\exp\left(\frac{\Delta G_0+\Delta\Delta G}{RT}\right)}. \tag{7.16}$$

Upon change of stability, the amount of functioning protein changes according to the free energy of the wild-type (ΔG_0) and the impact of the mutation ($\Delta\Delta G$).

One of the main hypothesis in this model is the additivity of the $\Delta\Delta G$,

$$\Delta\Delta G^{a,b}_{i,j} = \Delta\Delta G^{a}_{i} + \Delta\Delta G^{b}_{j}, \tag{7.17}$$

where $\Delta\Delta G^{a,b}_{i,j}$ is associated with the double mutations at sites i and j with amino acids a and b, $\Delta\Delta G^{a}_{i}$ is associated with the single mutation at sites i with amino acids a, and $\Delta\Delta G^{b}_{j}$ is associated with the single mutation at sites j with amino acids b.

P_{nat} can be directly connected to fitness in the case of an antibiotic resistance gene (Jacquier et al., 2013) and is proportional to the absolution fitness of the mutant. Therefore, the resulting log-fitness of a mutant can be computed as

$$\log\left(\frac{W}{W_{WT}}\right) = \log\left(1+\exp\left(\frac{\Delta G_0}{RT}\right)\right) - \log\left(1+\exp\left(\frac{\Delta G_0+\Delta\Delta G}{RT}\right)\right). \tag{7.18}$$

Depending on the mutant $\Delta\Delta G$, this model produces patterns of log-fitness effects according to background log-fitness similar to the one observed in the data (Fig. 7.7(a)).

To have the best possible estimate of the parameters, we decided to estimate the $\Delta\Delta G$ and ΔG_0 from the log-fitness of single and double mutants (Appendix D.3). As we accurately measure the log-fitness only above a threshold of -0.6, we keep only the 111 single mutants (53% of the total) with a log-fitness greater than -0.6. For each pair of previously chosen single mutants, the associated double mutant is kept if it has been measured experimentally. Its log-fitness is thresholded at -0.6. The two-state model is itself thresholded at -0.6 during the inference. Keeping the lethal double mutants allows a better estimation of the $\Delta\Delta G$.

We found $\Delta G_0 = -4.55$ kcal.mol^{-1}. For both biological semi-replicates, $\Delta\Delta G$ are highly correlated ($r^2 = 0.99$, Fig. D.2(c)). We found a correlation of $\rho = 0.91$, $r^2 = 0.87$ between

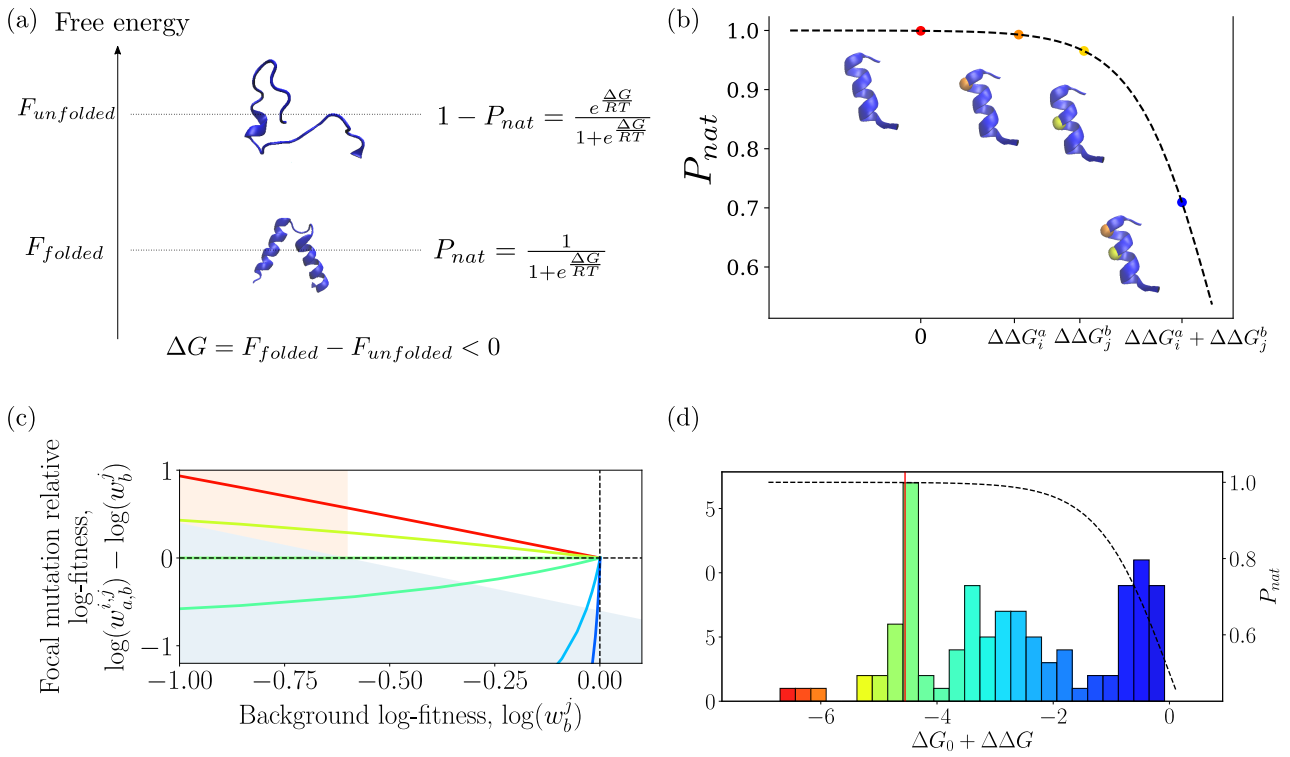


Figure 7.6: Stability and context-dependency. (a) Two-state model. P_{nat} is the probability that the protein folds. (b) Effects of the mutations on the stability. Black dotted line corresponds to P_{nat}. Red dot corresponds to the wild-type. Orange dot corresponds to a single mutation on the α-helix, with $\Delta\Delta G_i^a$. Yellow dot corresponds to a single mutation on the α-helix, with $\Delta\Delta G_j^b$. Blue dot corresponds to double mutations on the α-helix, with $\Delta\Delta G_i^a + \Delta\Delta G_j^b$. Mutations are considered as additive in $\Delta\Delta G$. However, this results in non-additive effect in P_{nat}. (c) The relationship between background log-fitness and mutant's relative log-fitness predicted by the model of stability is presented. The protein modeled has a free energy of -4.55 kcal.mol^{-1}, and the impact of mutations, $\Delta\Delta G$, is -2, -0.5, 0, 0.5, 2 and 3 kcal.mol^{-1} from red to blue. (d) Histogram of the 111 $\Delta\Delta G$ estimated. Red line corresponds to ΔG_0. Black dashed line corresponds to P_{nat} as a function of $\Delta G_0 + \Delta\Delta G$.

the observed and predicted log-fitness under the two-state model that has to be compared to a $\rho = 0.87$, $r^2 = 0.65$ correlation under the assumption that there is no epistasis. Hence, the two-state model is giving an improvement. Most importantly, the two-state model captures the overall background dependency of the mutants (Fig. 7.7(a)), reproduces the shape and breadth of the distribution of epistasis (Fig. 7.7(b)), with correlation $\rho = 0.81$, $r^2 = 0.55$, between observed and predicted epistasis (Fig. 7.8(a)). Therefore, it suggests that a significant fraction of epistasis between nonsynonymous mutations arises not through local and specific interactions between amino acids but mainly through global interactions, which are captured by the two-state model. Moreover, these results are consistent with previous experiments: R120G is known to have a stabilizing effect (Bershtein et al., 2008; Salverda et al., 2010) and this effect is indeed capture by the model, with $\Delta\Delta G = -1.85$ kcal.mol^{-1} (negative $\Delta\Delta G$ corresponds to stabilizing mutation).

However, a deeper look at the data suggests that the two-state model is not sufficient. First, keeping only the residues at less than 6Å, the correlation decreased to $\rho = 0.88$, $r^2 = 0.80$, while when only distant pairs (>6Å) were considered the correlation improved to $\rho = 0.95$, $r^2 = 0.89$. This implies that our model explained less well the interactions

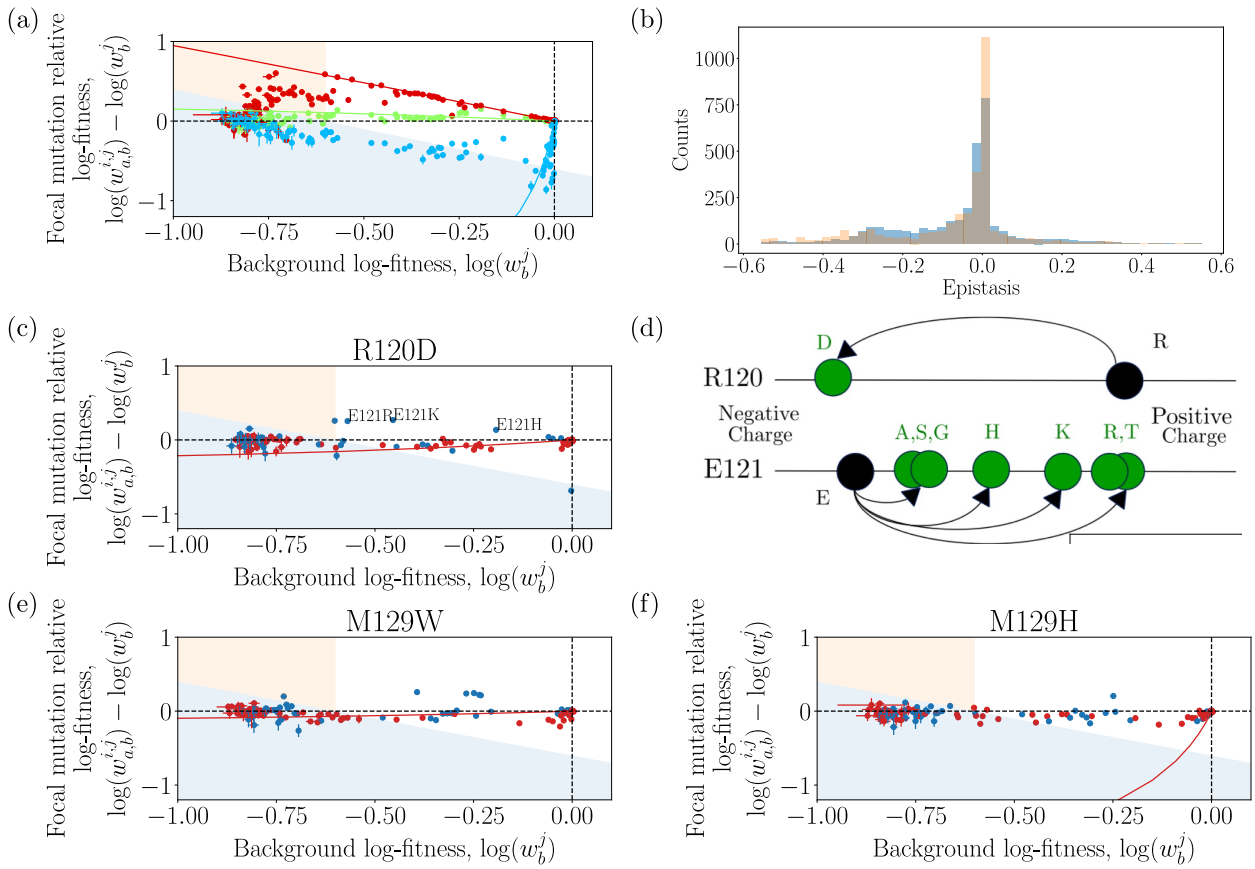


Figure 7.7: (a) The lines represent the fit of the model for the three mutants from Fig. 7.5(d). Due to the resolution of our experiments, the lines are valid only in the white area. (b) In blue is the distribution of epistasis as presented in Figure 7.5(b), and overlaid on it in orange is the distribution of epistasis obtained with the fitted two-state model. Deviations from the two-state model. Relative log-fitness according to background fitness for three mutants: R120D (c), M129W (e), and M129H (f). Red dots represent distant sites and blue dots nearby sites. (d) At residue 120, the decrease of charge associated with R to D mutation compensates mutations at residue 121 that increased the charge.

between nearby sites than distant sites.

Accordingly, a maximum likelihood model (Appendix D.4) was used to quantify the error to the two-state model. A model with two different errors was best supported, and found an error for sites at less than 6Å 1.28 times greater than the one found for sites further away. For some local interactions, other forces seemed to be at play. For instance, mutation R120D and M129W showed signs of both positive and negative epistasis, the positive effects on epistasis being restricted to residues in direct contact (Figs. 7.7(c) and (e)). R120D mutation leads to a change in charge, deleterious for distant interactions, but became beneficial when associated with departure from E121 charged amino acid, the neighboring amino acid (Fig. 7.7(d)). These interactions not captured by the two-state model represent what we refer to as idiosyncratic epistasis. They may result from the non-additivity of the $\Delta\Delta G$ and the existence of some local forces at play that are not directly linked to stability but are local. To determine pairs of sites with the strongest idiosyncratic epistasis, we use as proxy the mean square error between the experimental log-fitness $\log(w^{a,b}_{i,j})$ and the log-fitness predicted with the two-state model $\log(\hat{w}^{a,b}_{i,j})$ (Eq. 7.18),

$$D_{i,j} = \sqrt{\frac{1}{N_{i,j}} \sum_{a,b} \left(\log(w_{i,j}^{a,b}) - \log(\hat{w}_{i,j}^{a,b}) \right)^2}, \tag{7.19}$$

where $N_{i,j}$ is the total number of double mutants for which we can calculate the log-fitness according to the two-state model between sites i and j. The larger the score $D_{i,j}$, the larger the deviations from the two-state model, and thus the more the assumption of linearity of the $\Delta\Delta G$ is no longer valid (Eq. 7.17). The five pair of sites with the largest idiosyncratic epistasis are: 128-129, 124-128, 123-127, 127-128 and 120-123. Among these five pairs, four correspond to the residues at less than 6Å.

Theoretically, we can estimate $\Delta\Delta G$ from the log-fitness of single mutants only by inverting the equation (7.18). After the estimation of $\Delta\Delta G$ only on single mutants, it is possible to predict the log-fitness of double mutants from equations (7.17) and (7.18) and thus predict epistasis. Therefore, the model becomes predictive, because it is possible to estimate the effect of double mutants without having to experimentally measure their log-fitness: only measurements of single mutant's log-fitness are required. With this method, we obtain satisfactory results to predict the epistasis (Fig. 7.8(b)). We are also able to predict the sign of the epistasis from the $\Delta\Delta G$ inferred only on single mutants. Measures of the performances of our predictions are quantified by a ROC curve (Fig. 7.8(c)). We used a threshold for the epistasis, keeping only experimental epistasis above this threshold in absolute value (Fig. 7.10(e) represents the AUC for different values of the threshold).

However, by inferring only on single mutants, the estimation of the parameters is less precise for two main reasons. First, accurate estimation of ΔG_0 is complicated from the single mutants only. Nonetheless, by taking a ΔG_0 varying from -7 kcal.mol^{-1} to -3 kcal.mol^{-1}, we obtain quite robust results (Table 7.2). Second, as stabilizing mutants have a minor effect on log-fitness as they are on the plateau side of the energies to log-fitness (Fig. 7.6(b)), noise in the log-fitness estimation can result in a significant change in the $\Delta\Delta G$ estimation. Furthermore, for single mutant with stabilizing effect ($\Delta\Delta G < 0$), *i.e.* linked with positive epistasis, we underestimate their energies as their log-fitness is close to 0 (and therefore associated with $\Delta\Delta G = 0$): their stabilizing effect only appears when double mutants are included.

	Prediction epistasis, Spearman (ρ)	Prediction epistasis Pearson (r^2)	Prediction sign epistasis (AUC)
$\Delta\Delta G$ (single mutants)	0.6 to 0.7	0.21 to 0.49	0.80 to 0.81
$\Delta\Delta G$ (single and double mutants)	0.81	0.55	0.88

Table 7.2: Comparison of the prediction of the epistasis with the two-state model for $\Delta\Delta G$ inferred on single mutants only, and $\Delta\Delta G$ inferred on single and double mutants. For estimation on single mutants, ΔG_0 is varying from -7 kcal.mol^{-1} to -3 kcal.mol^{-1}. For estimation on single and double mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}.

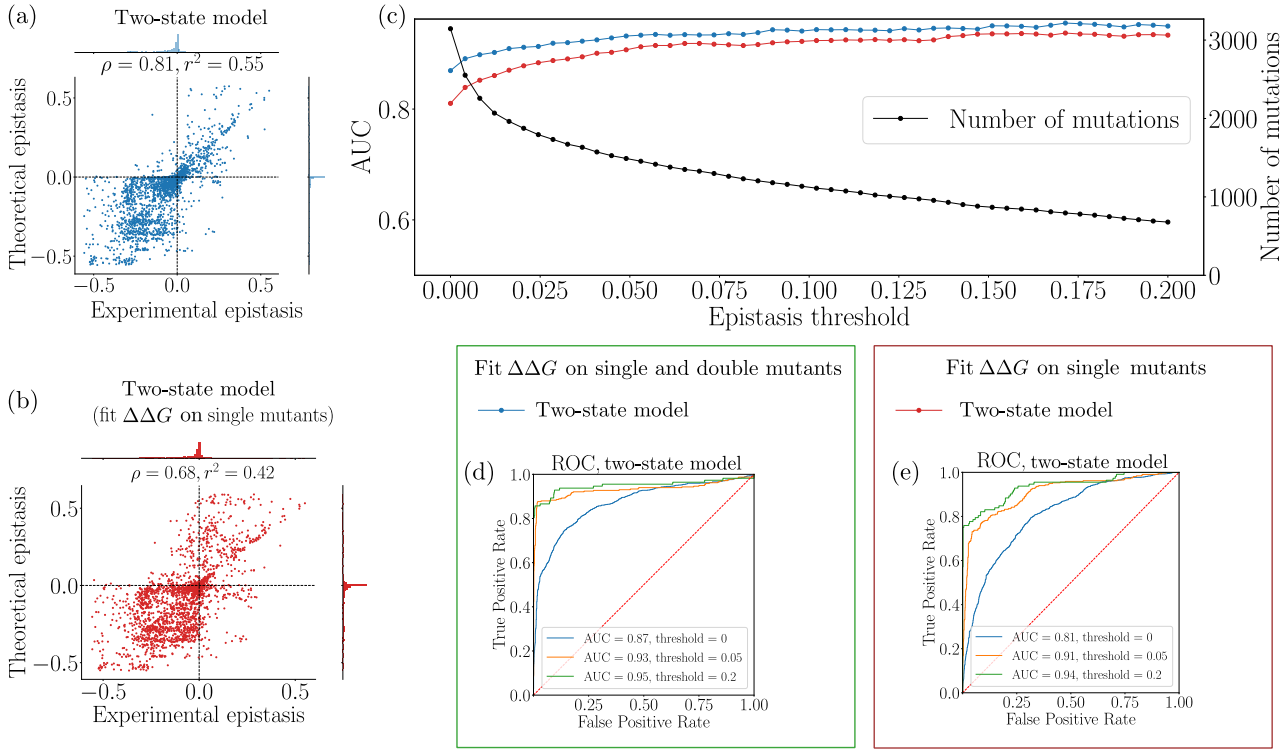


Figure 7.8: Prediction of pairwise epistasis with two-state model. (a) Epistasis predicted with two-state model against experimental epistasis. (b) Epistasis predicted with two-state model against experimental epistasis ($\Delta\Delta G$ inferred only on the single mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}). (c) AUC against epistasis' threshold for the two-state model. (d) ROC curves for the two-state model for different epistasis' threshold. (e) ROC curves for the two-state model for different epistasis' threshold ($\Delta\Delta G$ inferred only on the single mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}).

7.4.3 Prediction from MSA

To complete our analysis and to see if it is possible to predict before the experiment the effects observed on the log-fitness and epistasis of the mutants, we trained independent models and Potts models on multiple sequence alignment built on high-quality homologs of class A β-lactamases cleaned by hand (Philippon et al., 2016, 2019) and enriched on SwissProt and TrEMBL (The UniProt Consortium, 2021). Once trained, we score all the single and double mutants according to their energies $E(\mathbf{a})$ (Fig. 7.1(b)). For the independent model, $E(\mathbf{a})$ reads

$$E(\mathbf{a}) \quad = \quad -\sum_{i=1}^{L} h_i(a_i), \tag{7.20}$$

and for Potts model, $E(\mathbf{a})$ reads

$$E(\mathbf{a}) \quad = \quad -\sum_{i=1}^{L} h_i(a_i) - \sum_{1 \leq i < j \leq L} J_{ij}(a_i, a_j). \tag{7.21}$$

To compare the predictions $E(\mathbf{a})$ and the results of the experiments, we need a proxy to link the two quantities. The most common proxy is to use the difference of log-

likelihood between the mutant $\mathbf{a}_{mut}$ and the wild-type $\mathbf{a}_{WT}$ as a proxy for the results of the experiments (Figliuzzi et al., 2016; Hopf et al., 2017; Zhao et al., 2021)

$$\log P(\mathbf{a}_{mut}) - \log P(\mathbf{a}_{WT}) = -E(\mathbf{a}_{mut}) + E(\mathbf{a}_{WT}). \tag{7.22}$$

For the Potts model, we found a correlation $\rho = 0.86$ for the 209 single mutants and $\rho = 0.64$ for the 15.279 double mutants, to be compared with $\rho = 0.81$ and $\rho = 0.59$ for the independent model. Although the independent model leads to comparable results, they are slightly worse. The Potts model, thanks to its couplings $J_{ij}(a,b)$, allows having a better estimation of the effects of the mutations, because the couplings take into account the background of TEM-1, instead of having an average global effect due to all the class A β-lactamases, as in the case of the independent model.

We noticed a typical "S" shape between Potts energies and the log-fitness (Figs. 7.9(a) and (b)). Similar results are obtained using RBM (Figs. D.4(a) and (b)). For independent model, the relationship is even more bimodal (Figs. D.3(a) and (b)). The relation between MIC and our proxy is more linear for mutations with a high MIC, but saturates for those with a low MIC (Fig. D.5).

The relation between the log-fitness and our proxy is highly nonlinear. Indeed, our proxy does not depend on the quantity measured in the experiment. It is well known that Potts energies can be correlated to MIC (Figliuzzi et al., 2016), specificity constant ($\frac{k_{\text{cat}}}{K_M}$) (Zhao et al., 2021), log-fitness (Hopf et al., 2017), or binding energies (Salinas and Ranganathan, 2018). All these quantities are indeed correlated, but in a non-linear way (see for example in our case, the non-linear relation between log-fitness and MIC, Fig. D.1).

Therefore, although the strong correlation between the experimental log-fitness and the predictions of our models, due to the nonlinearity between these quantities and as epistasis is a linear function of the log-fitness, Potts models fails to predict the epistatic effects with our proxy: $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) - E(\mathbf{a}_{WT}) + E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{mut_j^b})$ ($\rho = -0.06$). This result holds also for RBM.

The typical "S" shape between the log-fitness and the energies is reminiscent of the relationship described by the two-state model: the first plateau, corresponding to mutants with energy close to the one of the wild-type and log-fitness close to 0, implies that changes in energy have minor effect on the log-fitness. Then, there is the part where changes in energies implies changes in log-fitness. And finally, for there is a second plateau, corresponding to mutants with high energies and lethal in our experiments. In this regime, our experiments do not have enough resolution (for log-fitness < -0.6) and consequently, the log-fitness saturates. We can not distinguish experimentally lethal mutants from those predicted to be even more lethal according to their Potts model's energies.

Therefore, considering the relationship between the energies and $\Delta\Delta G$, instead of the log-fitness, we observed a much more linear relation with our predictions, for the single mutants

$$\Delta\Delta G_i^a = \gamma(-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})), \tag{7.23}$$

with $r^2 = 0.67$, and $\gamma = -0.71$, and for the double mutants

$$\Delta\Delta G_i^a + \Delta\Delta G_j^b = \gamma(-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})), \tag{7.24}$$

with $r^2 = 0.66$ (Figs. 7.9(c) and (d)), keeping only the mutations where the $\Delta\Delta G$ are estimated and where the amino acids are available in the MSA. The relation between $\Delta\Delta G$ and the energy of the independent model is less linear ($r^2 = 0.37$ for the single mutants and

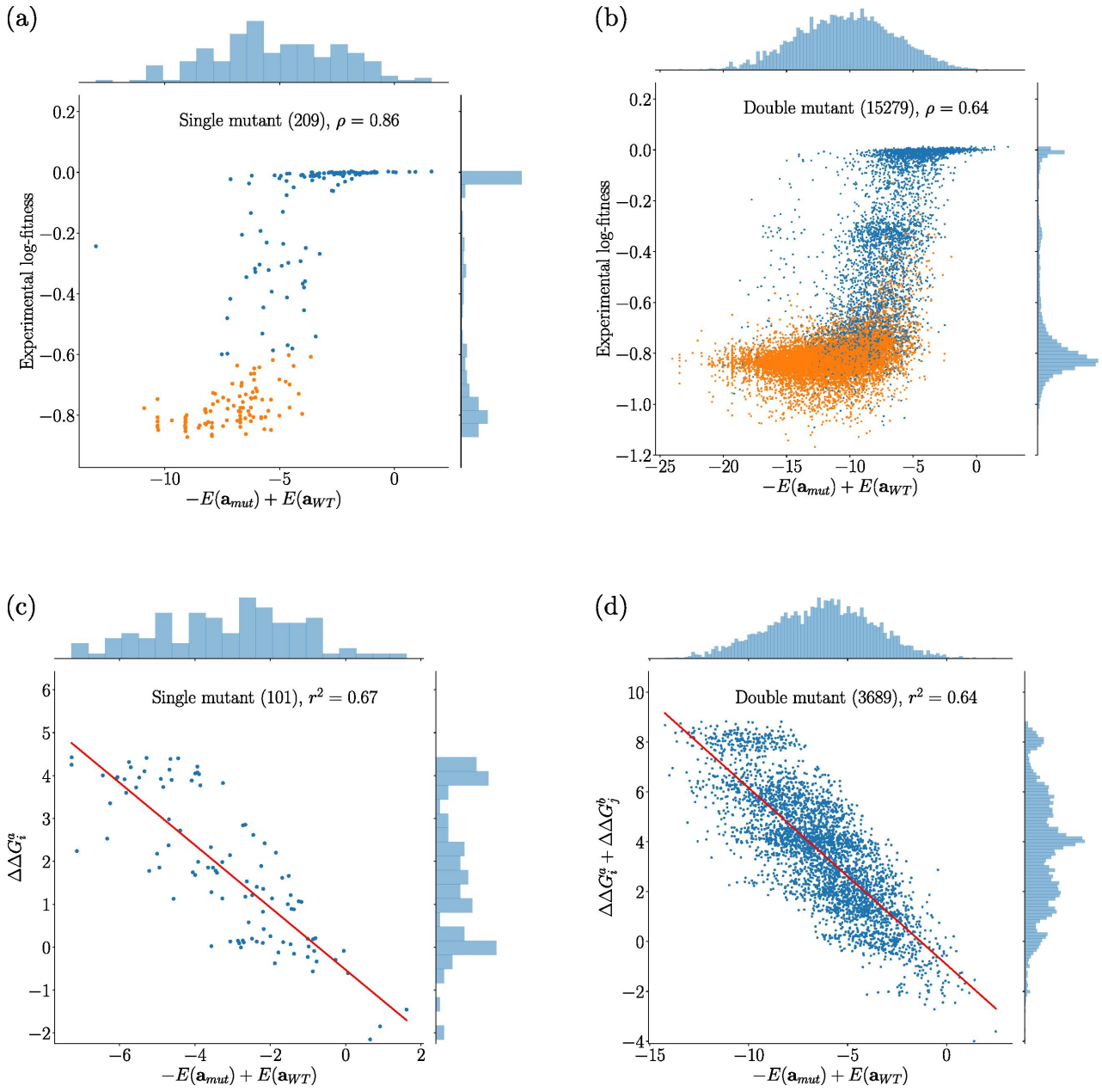


Figure 7.9: Potts' energies versus experimental quantities. Blue points are common mutations in panel (a) and (c) (respectively (b) and (d)), and correspond to the mutations we used to estimate $\Delta\Delta G$. Orange points are the other experimental mutations. (a) Experimental log-fitness against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$ for single mutants. (b) Experimental log-fitness against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$ for double mutants. (c) $\Delta\Delta G_i^a$ against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$. (d) $\Delta\Delta G_i^a + \Delta\Delta G_j^b$ against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$.

$r^2 = 0.38$ for the double mutants, Figs. D.3(c) and (d)), showing that the couplings $J_{ij}(a,b)$ are paramount to have an accurate estimation of the effects, which takes into account the TEM-1 background.

After having inferred $\gamma = -0.71$ and $\Delta G_0 = -4.55$ kcal.mol^{-1}, we can predict the $\Delta\Delta G$ for the single and double mutants from their energies according to the Potts model by using equations (7.23) and (7.24). Then, by using two-state model (Eq. 7.18), we can predict the log-fitness of single and double mutants, and consequently, the epistasis.

The predicted epistasis and the experimental one are correlated ($\rho = 0.44$, Fig. 7.10(b)). This is an improvement compared to the direct predictions from the log-likelihood ($\rho = -0.06$), but the model seems to capture only the sign of the epistasis and not its value (Fig. 7.10(b)). Nonetheless, we can use our estimation of experimental epistasis to predict the sign of the experimental one. As in Fig. 7.8(c), we measure the performances of our predictions with ROC curve (Fig. 7.10(g)).

As mentioned before, without the two-state model, the Potts model fails to capture the epistatic effects ($\rho = -0.06$ between experimental values and predictions). Nevertheless, we noticed that the couplings, at the scale of the interactions between sites, seem to capture the pairs of sites that have the most important idiosyncratic epistasis, *i.e.* not explained by the two-state model.

For Potts model, the canonical proxy to measure the interactions between two specific sites is the Frobenius norm of the couplings matrices $F_{ij} = \sqrt{\sum_{a,b} J_{ij}(a,b)^2}$ (with the average-product correction (Dunn et al., 2008)). The top couplings of this metric are traditionally used to predict the tertiary contacts (Morcos et al., 2011).

We found that among the five pairs of sites with the largest Frobenius norm, there are three pairs with significant idiosyncratic epistasis: 124-128, 127-128, 128-129 (Section 7.4.2). Under the assumption that there is no link between these two quantities, it leads to a p-value equals to 0.0036 (see Appendix D.8 for the computation).

Therefore, it seems that the most interacting pairs of sites predicted by our models within the α-helix correspond to the pairs of sites where the two-state model is the less predictive: local idiosyncratic interactions seem to result in the long term in some specific coevolution patterns between pairs of sites, which are captured by Potts model. However, these effects are not captured at the scale of the interactions between two specific sites and two specific amino acids, but at the scale of the sites.

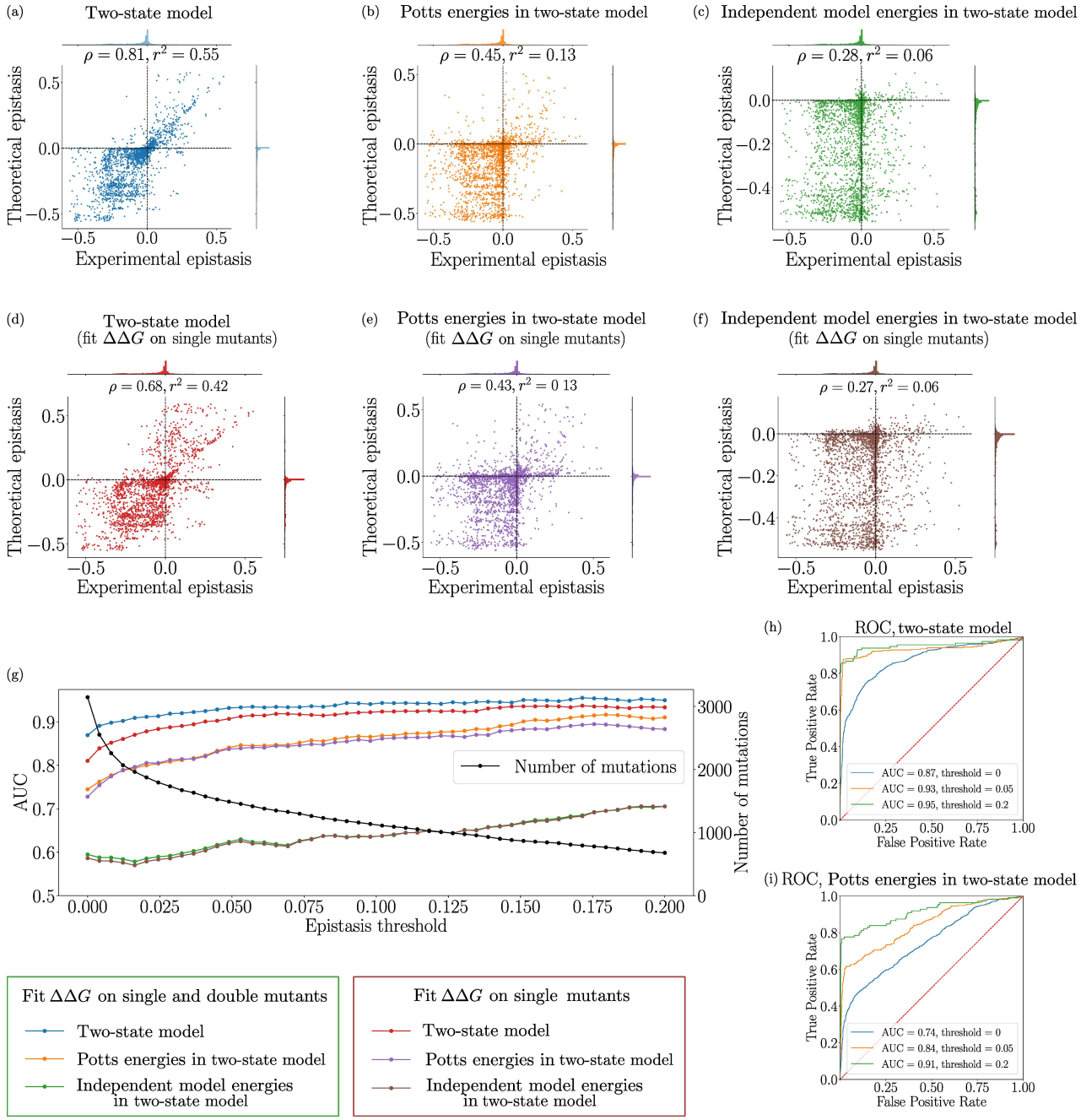


Figure 7.10: Prediction of pairwise epistasis. (a) Epistasis predicted with two-state model against experimental epistasis. (b) Estimated epistasis with Potts energies in two-state model against experimental epistasis. Our predictions capture the sign of the experimental epistasis. (c) Estimated epistasis with independent model energies in two-state model against experimental epistasis. (d) Epistasis predicted with two-state model against experimental epistasis ($\Delta\Delta G$ inferred only on the single mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}). (e) Estimated epistasis with Potts energies in two-state model against experimental epistasis ($\Delta\Delta G$ inferred only on the single mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}). (f) Estimated epistasis with independent model energies in two-state model against experimental epistasis ($\Delta\Delta G$ inferred only on the single mutants, $\Delta G_0 = -4.55$ kcal.mol^{-1}). (g) AUC against epistasis' threshold for the different models. (h) ROC curves for the two-state model for different epistasis' threshold. (i) ROC curves for the Potts energies in the two-state model for different epistasis' threshold.

7.5. Discussion

The deep mutational scan we have performed here to study mutation effects in a local alpha-helix of the β-lactamase TEM-1 reveals that epistasis is pervasive. We found that once we exclude mutations carrying irrevocable loss of function, 83% of mutations showed some strong signature of epistasis. Interestingly, though we work on a small fraction of the protein, most epistasis do not result from idiosyncratic interactions between sites, but are mostly captured by a global model of epistasis. In that model, the phenotypic impact of the mutant adds up in double mutants, but the non-linear translation of phenotype to fitness results in epistasis (Wylie and Shakhnovich, 2011; Otwinowski et al., 2018). The functional form of the non-linear mapping between the fitness and the phenotype may reflect the global impact of the mutations on the protein stability, in particular for the secondary structure component under investigation, and on its functionality. The phenotype to fitness mapping therefore the environmental pressure on the activity of the protein, tuned by the experimental conditions, here determined by the antibiotic concentration (Stiffler et al., 2015; Otwinowski et al., 2018). Using the two-state model and the single and double mutations scan, we could estimate for each single mutant a phenotypic effect in the form of an energy change, $\Delta\Delta G$. Within this model, we could explain mutants, both qualitatively and quantitatively, a large fraction of the observed epistasis ($\rho = 0.81$). Moreover, as, according to the two-state model, the mutational effects on the phenotype are additive, we could fit the $\Delta\Delta G$ parameters only from the single mutational data, to predict epistasis with a good accuracy as estimated with a Spearman correlation ranging from 0.6 to 0.7. The large contribution of this global epistasis we observed despite our focus on a local structure of the protein is remarkable and further emphasizes the importance of this form of epistasis, whose overall relative contribution should only increase as we consider larger fractions of the protein. The importance of these macroscopic form of epistasis at the protein level is reminiscent of the negative epistasis found genome-wide in experimental evolution (Chou et al., 2011; Khan et al., 2011; Wiser et al., 2013; Kryazhimskiy et al., 2014).

Our precise estimates of log-fitness allowed us to identify some deviations to the two-state model. Interestingly, there was also some consistency in these deviations that were more likely to occur between residues in direct contacts in the protein structure. We found for instance some examples of local interactions linked to charge conservation. Deviation from the additivity at the phenotypic level may generate these deviations from macroscopic epistasis. We would like to point out that our alpha helix is not included in the active site of the protein. We believe that our two-state model would be less predictive for sites included in the active site, where activity would predominate over global epistasis (Rodrigues et al., 2016). However, we estimate that for a majority of sites, this global epistasis dominates.

Both global epistasis and deviation from it seem to be connected to the 3D structure of the alpha-helix under investigation, either through the impact of mutations on protein stability or through contacts between the residues. Because such structure is highly conserved, we then questioned whether the determinants of epistasis were conserved enough to be detected from the analysis of Multiple Sequence Alignments (MSA) of distant homologues that share the same fold. Interestingly, both the signature of the macroscopic model and the patterns of deviations were recovered through the integration of MSA in the Potts model. First, the estimated $\Delta\Delta G$ correlated linearly with the Potts model mutation energy predictions. However, because macroscopic epistasis results from a precise non-linear mapping of phenotype to fitness, the Potts model estimates of $\Delta\Delta G$ had to be inserted in the two-state model to have some predictive power on the observed epistasis (mostly on the sign of epistasis). Second, pairs of sites that showed the strongest signal of

coevolution through evolutionary times (as measured through the Frobenius norm of the couplings of Potts model) were the ones that deviated the most from the macroscopic model. These idiosyncratic epistatic interactions seem therefore to generate in the long-term some co-evolution patterns between pairs of sites that can be captured by models trained on MSA.

The fact that the experimental epistasis we characterized as either global or idiosyncratic can both be recovered to some extent from the analysis of distant homologues is telling that the molecular determinants of epistasis are long-lasting. It suggests that the persistence of the underlying mechanistic selective pressures has been long and strong enough to shape the long-term evolution of the protein family. Despite the wide-spread level of epistasis we recovered in our data, these observations reject a model in which epistatic interactions are fully volatile and change quickly with protein sequence as suggested for instance in the NK model (Kauffman and Weinberger, 1989). Our data suggest a rather smooth and consistent protein mutational landscape. This offers the hope that its property could be tractable and extrapolated from one homologue to another using combinations of mutational scans and in-depth multiple sequence alignment analysis.

8

Analysis of the effects of amoxicillin concentration

To further analyze the effects of α-helix mutations, experiments presented in Chapter 7 were performed again by the group of Olivier Tenaillon, with different concentrations of amoxicillin: 2, 4, 5, 6, 8, and 10 g.L^{-1}. For each concentration, the log-fitness of single and double mutants can be estimated with the inference procedure described in Section 7.3.3. Therefore, we can compare log-fitness and epistasis for given mutants at different concentrations.

The detailed study of the various effects is unfortunately not yet complete, and we present preliminary but encouraging results here. By comparing the log-fitness at different concentrations, we notice that structured patterns appear: the mutants become deleterious from a certain concentration of drug, with a threshold that depends on the mutant. This results in structured patterns when comparing epistasis at different concentrations. By modifying our two-state model presented in Chapter 7, we are able to qualitatively capture these patterns.

8.1. Effects of amoxicillin concentration on log-fitness

As expected, the selection pressure on the mutants increases with the concentration of amoxicillin, and consequently, the fraction of lethal mutants increases with it (Fig. 8.1(a)). This effect is also noticed in Stiffler et al. (2015)'s experiments. In their experiments, they subject single mutants of TEM-1 to different ampicillin[1] concentrations and notice that as the concentration increases, the fraction of lethal mutants also increases in turn.

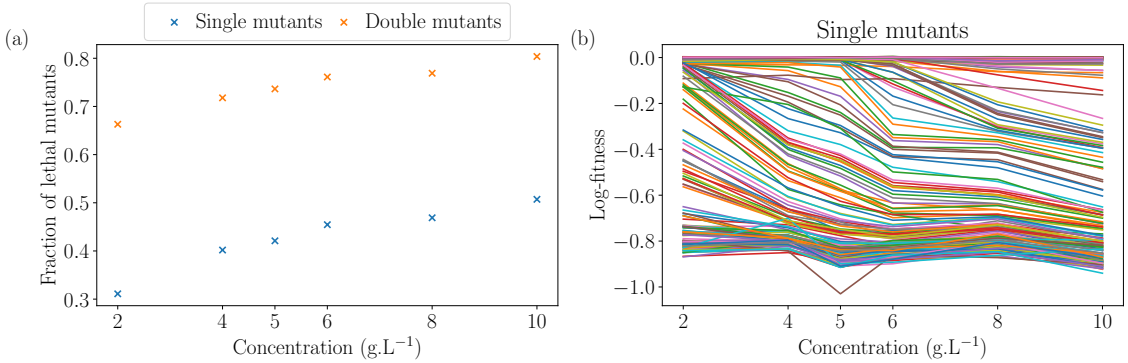


Figure 8.1: Effects of concentration on log-fitness. (a) Evolution of the fraction of lethal mutants against concentration. (b) Evolution of the log-fitness of the 209 single mutants against concentration.

[1] As amoxicillin, ampicillin is a drug that inhibits β-lactamase.

To understand this phenomenon, they propose a kinetic model. Schematically, the drug within the cell neutralizes Penicillin Binding Proteins (PBP)[2]. PBP activity can be related to the concentration of the drug in the cell, and this concentration depends on both the concentration in the medium and the ability of the mutant to hydrolyze the drug. For each mutant, their model depends on its turnover number k_{cat} and Michaelis constant K_m, which are fitted to the experimental data. The relationship between their parameters (mutant activity) and fitness is not linear. Fitness saturates for high mutants' activity. The saturation threshold depends on the concentration and increases with it. This explains the increasing fraction of lethal mutants with increasing concentration.

Comparing the MIC results to the log-fitness at different concentrations, we also notice this saturation effect. The lower the concentration, the more the relationship between log-fitness and MIC reaches its plateau at low concentrations (Fig. E.1).

Furthermore, we also observe that some mutants have log-fitness close to 0 for low concentrations, and then become increasingly deleterious above a certain concentration threshold (Fig. 8.1(b)). Other mutants are lethal at any concentration. Surprisingly, we notice that the log-fitness do not cross much (Fig. 8.1(b)): this implies that in our concentration range if one mutant is more deleterious than another at a specific concentration, it is deleterious over the whole concentration range.

For double mutants, we note a nonlinear relationship between the log-fitness measured at different concentrations (Fig. 8.2). For the figures above the diagonal, the concentration on the abscissa is greater than the concentration on the ordinate. We observe this saturation phenomenon, where mutants with a log-fitness close to 0 at low concentrations are deleterious at higher concentrations (log-fitness < 0).

To capture the saturation of the log-fitness as a function of the concentration, we take another approach than the one proposed in Stiffler et al. (2015).

First, we naively model the behavior of log-fitness as a function of concentration with a ReLU function, with three parameters

$$\text{ReLU}(x) = \min(a, bx + c) \tag{8.1}$$

where a denotes the offset, b the slope, and c the threshold. The dropout, *i.e.*, the value of x such as $bx + c = 0$ is defined as $-\frac{c}{b}$. Therefore, for mutants with a log-fitness near 0 at low concentrations, the dropout corresponds to the concentration where their log-fitness starts to decrease. There seems to be a relationship between slope and dropout (Fig. 8.3(d)). Up to a dropout of 5 g.L^{-1}, the slope is a slightly decreasing function of the dropout. As we will see in Section 8.2, this will have consequences on the epistasis. After 5 g.L^{-1}, the slope is an increasing function of the dropout.

We propose modifying the two-state model (Eq. (7.18)) to take into account the effect of concentration. The idea is still to infer one $\Delta\Delta G$ by single mutants, but to have one $\Delta G_0(c)$ for each concentration. Therefore, the two-state model reads

$$\log\left(\frac{W(c)}{W_{WT}(c)}\right) = \log\left(1 + \exp\left(\frac{\Delta G_0(c)}{RT}\right)\right) - \log\left(1 + \exp\left(\frac{\Delta G_0(c) + \Delta\Delta G}{RT}\right)\right). \tag{8.2}$$

where $W(c)$ is the absolute fitness of the mutant at concentration c, $W_{WT}(c)$ is the absolute fitness of wild-type at concentration c. By having $\Delta G_0(c)$ that depends on the concentration, we can modulate the threshold at which the function described in the equation (8.2) reaches its plateau.

[2]PDP are essential in cell-wall synthesis. Therefore, without PBP, a cell has no wall and is not viable.

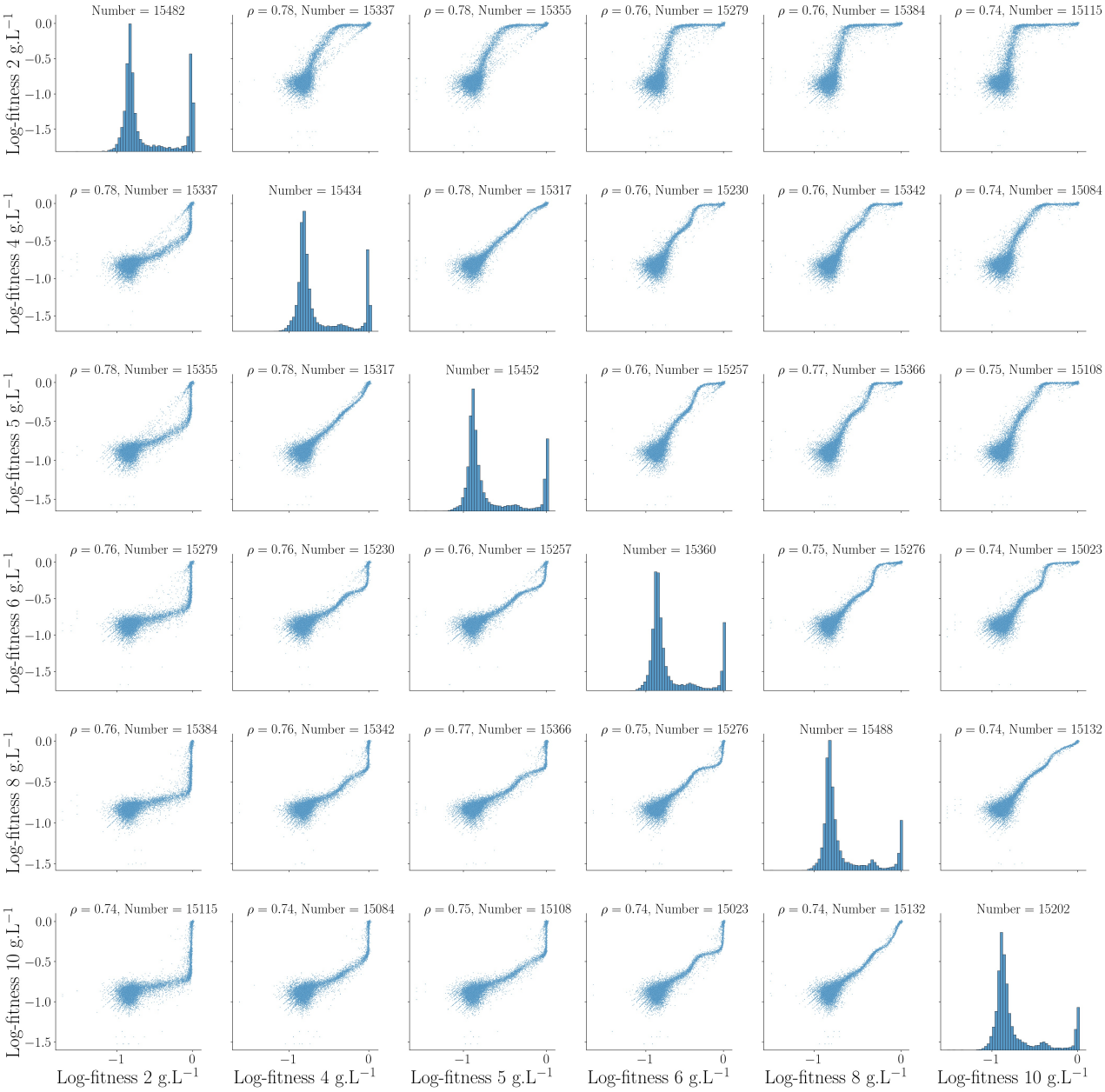


Figure 8.2: Comparison of log-fitness at concentrations 2, 4, 5, 6, 8, and 10 g.L^{-1} of amoxicillin.

We still assume that there is the additivity of the $\Delta\Delta G$,

$$\Delta\Delta G_{i,j}^{a,b} = \Delta\Delta G_i^a + \Delta\Delta G_j^b, \tag{8.3}$$

As we accurately measure the log-fitness only above a threshold of -0.6, we keep only the 103 single mutants (49% of the total) with a log-fitness greater than -0.6 for all concentrations. For each pair of previously chosen single mutants, the associated double mutant is kept if it has been measured experimentally for all concentrations. Its log-fitness is thresholded at -0.6. The two-state model is itself thresholded at -0.6 during the inference. Therefore, since we now have six concentrations available to us instead of one as in the Chapter 7, we have multiplied roughly the number of data by 6 (22.968 experimental log-fitness), but we have added only five parameters to our two-state model (109 free

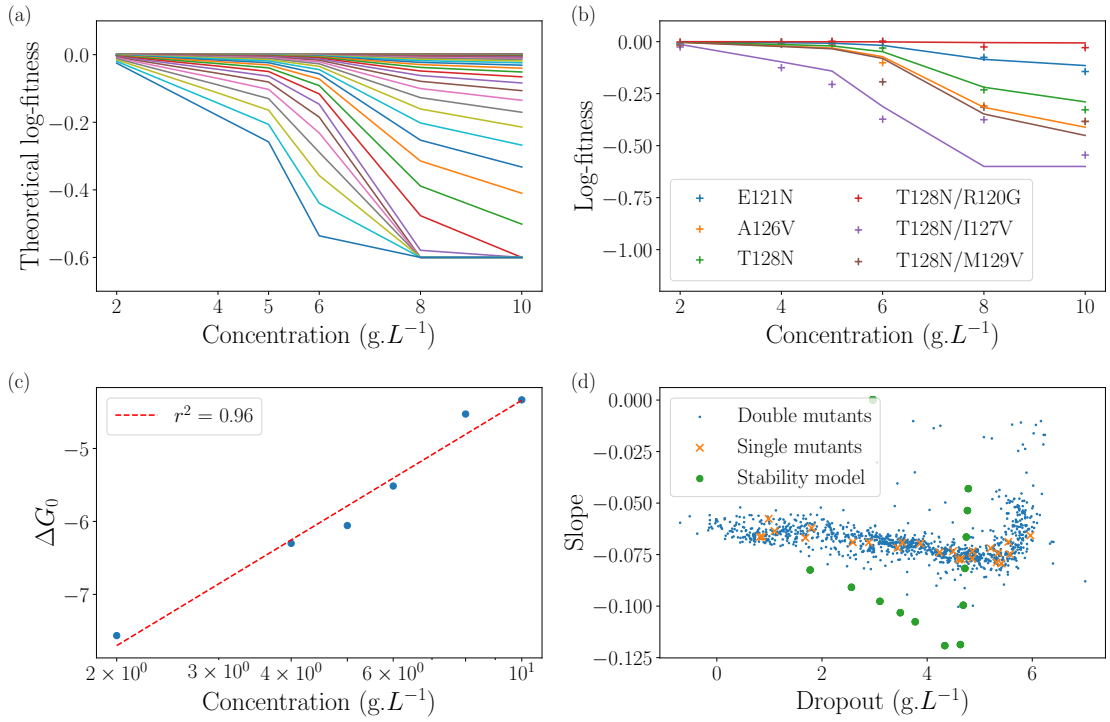


Figure 8.3: (a) Log-fitness obtained with the two-state model for ΔG_0 inferred on the data, and for $\Delta\Delta G$ varying between -2.4 and 5.3 kcal.mol^{-1} (extreme values obtained on the data). Each curve corresponds to different $\Delta\Delta G$. (b) Cross: experimental data. Dashed lines: two-state models. (c) ΔG_0 versus $\log(c)$. (d) Relation between dropout and slope.

parameters : $103\,\Delta\Delta G$ and $6\,\Delta G_0(c)$): the model is largely overconstrained.

This model reproduces qualitatively the shape of log-fitness observed on the data (compare Fig. 8.3(a) and Fig. 8.1(b)). Furthermore, the model reproduces well the log-fitness data of single mutants ($r^2 = 0.93$, and see Fig. 8.3(b) for some examples), as well the log-fitness data of double mutants ($r^2 = 0.77$, and see Fig. 8.3(b) for some examples). We further note that $\Delta G_0(c)$ is an increasing linear function of the logarithm of the concentration (Fig. 8.3(c)): the lower the concentration, the more stable the protein is, and therefore the weaker the effects of mutations on log-fitness. With the two-state model, we also observe a relationship between dropout and slope, similar to that observed on the data (Fig. 8.3(d)). Moreover, by plotting the predictions at different concentrations of the two-state model (Fig. E.2), we do find the log-fitness saturation effect present in the data (Fig. 8.2).

Furthermore, with this concentration dependence of $\Delta G_0(c)$, the model captures epistasis well at different concentrations (Fig. 8.4).

We can compare these results with those presented in Chapter 7, where the $\Delta\Delta G$ and ΔG_0 are inferred only to the concentration 8 g.L^{-1}. We find $\Delta G_0(8) = -4.53$ kcal.mol^{-1} compared to $\Delta G_0 = -4.55$ kcal.mol^{-1}, and the $\Delta\Delta G$ are very correlated ($\rho = 0.96$, $r^2 = 0.93$). Epistasis predictions are slightly worse ($\rho = 0.70$, $r^2 = 0.49$ compared to $\rho = 0.81$, $r^2 = 0.55$).

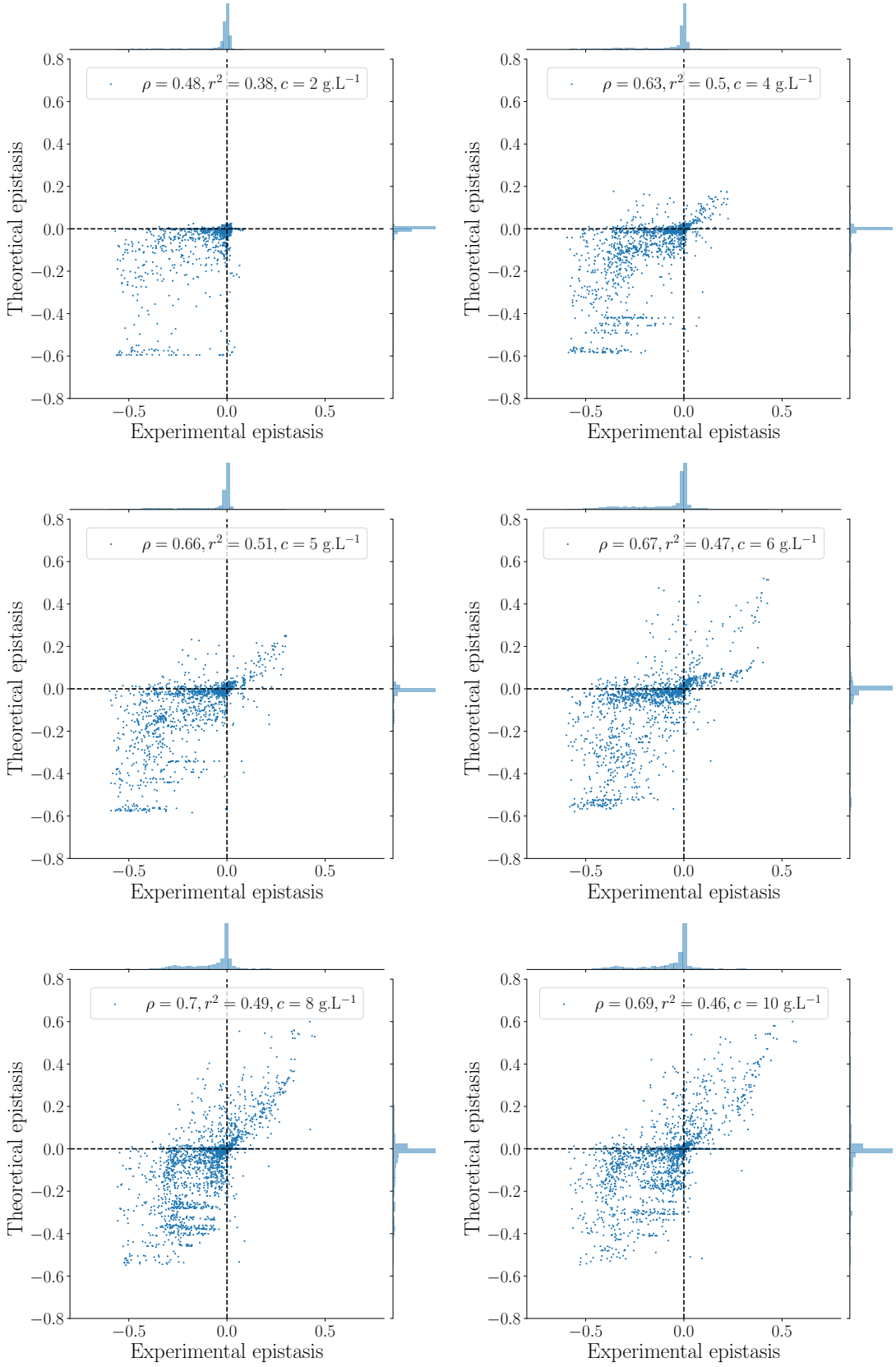


Figure 8.4: Predicted epistasis versus experimental epistasis, depending on the concentration from 2 g.L^{-1} (top left) to 10 g.L^{-1} (bottom right).

8.2. Effects of amoxicillin concentration on epistasis

The distribution of epistasis depends on concentration: strong positive epistasis is only visible at high concentrations (Fig. E.3). Moreover, the dependence of log-fitness on concentration as well as the dependence between dropout and slope create non-trivial relationships between epistasis at different concentrations (Fig. 8.5).

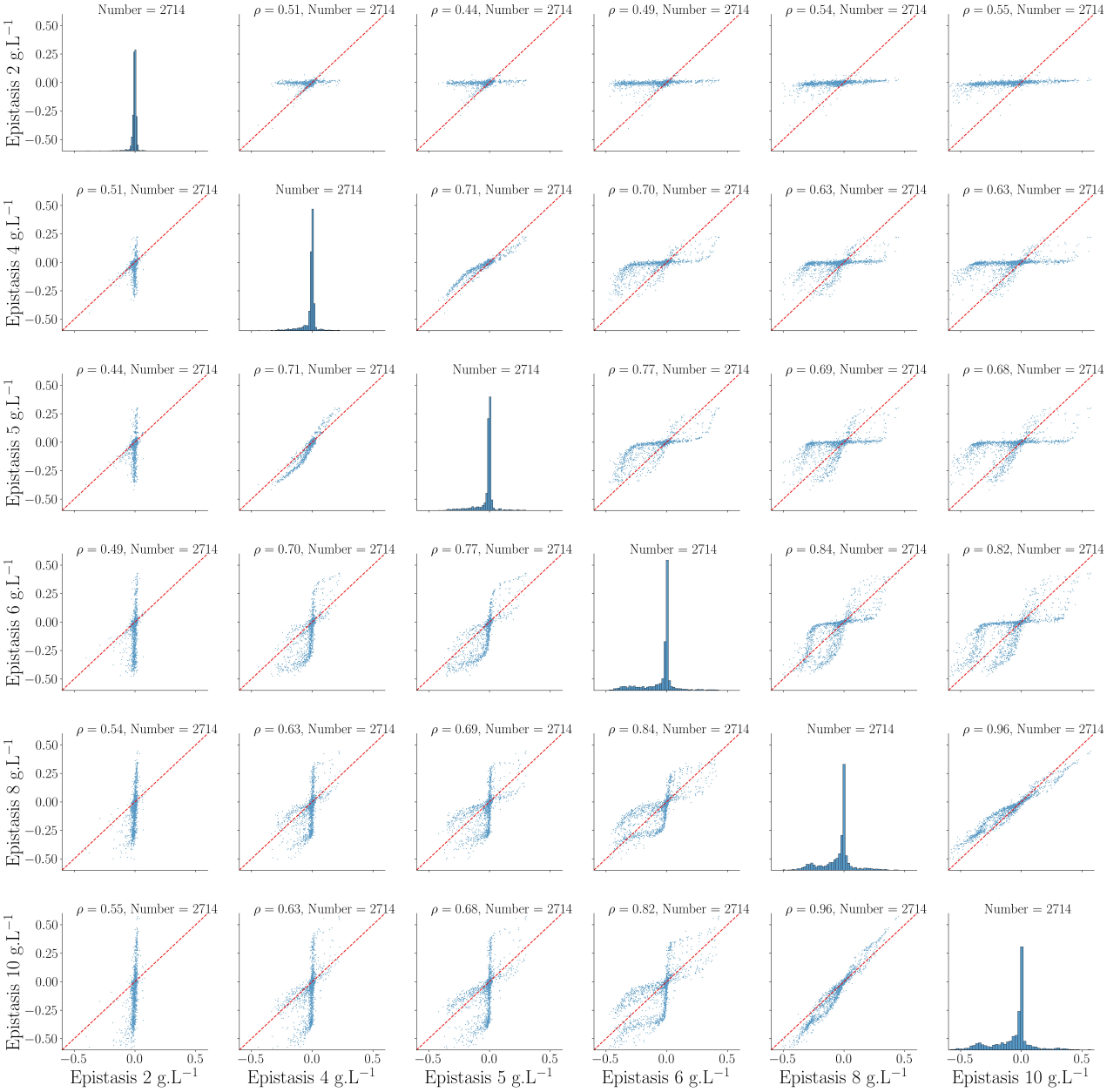


Figure 8.5: Comparison of epistasis at concentrations 2, 4, 5, 6, 8, and 10 g.L^{-1} of amoxicillin.

These relationships can be separated into two categories, as we have done in Figure 8.6(a): in blue, double mutants that have greater epistasis at high concentration than at low concentration, and in orange, double mutants that have greater epistasis at low concentration than at high concentration.

The plateau we observe in Figure 8.6(a), where the epistasis at low concentration is zero, and the epistasis at high concentration is non-zero, is a direct consequence of the

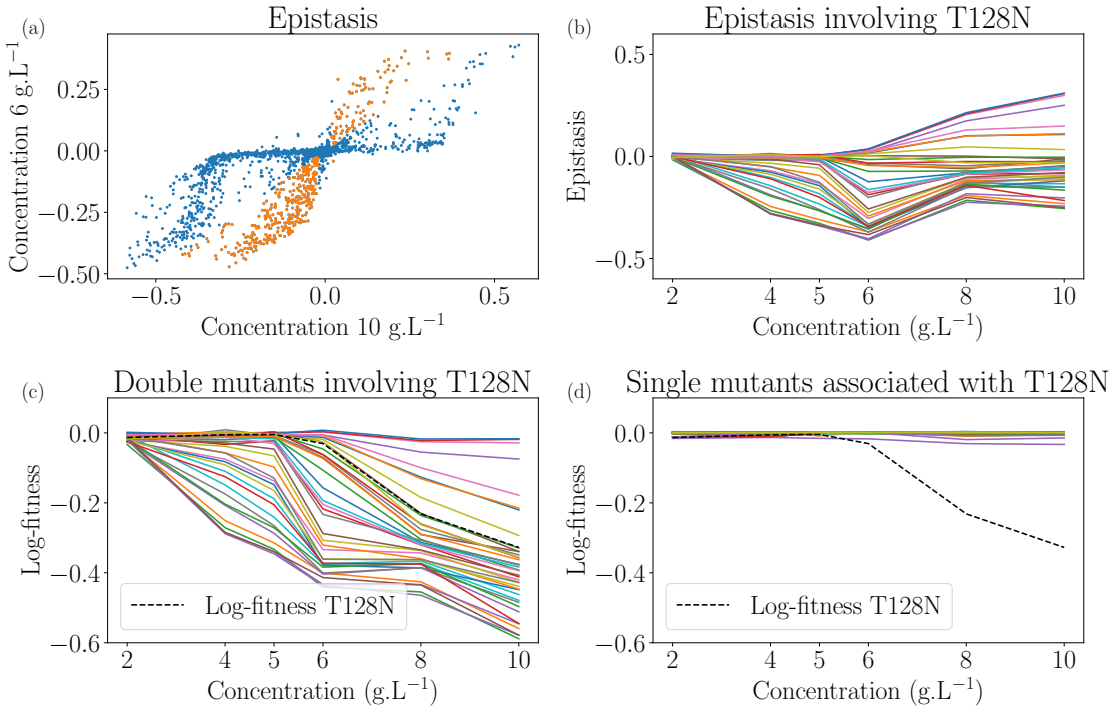


Figure 8.6: (a) Epistasis at 6 g.L^{-1} against epistasis at 10 g.L^{-1}. (b-d) Each color corresponds to a given mutant. Black dashed line corresponds to T128N. (b) Epistasis involving T128N. (c) Log-fitness of double mutants involving T128N. (d) Log-fitness of single mutants associated with T128N.

log-fitness saturation we observe in Figure 8.2: at low concentrations, the single mutants and the associated double mutant have a log-fitness close to 0, and the epistasis is therefore zero. Epistasis is only visible at higher concentrations when one (or several) of these mutants begin to be deleterious. We also observe that the highest epistasis at 10 g.L^{-1} correspond to high epistasis at 6 g.L^{-1}.

We also observe a less intuitive phenomenon, corresponding to the orange dots discussed earlier, where epistasis at low concentration is greater than epistasis at high concentration. This phenomenon can be interpreted with the help of Figures 8.6(b-d). In the case of the T128N mutation, we observe the epistasis is a non-monotonic function of the concentration for several double mutants (Fig. 8.6(b)), and that the single mutants associated with T128N have a log-fitness close to 0 at any concentration (Fig. 8.6(d)). Therefore, the epistasis depends on the log-fitness of the double mutant as well as the single mutant T128N. We observe that several double mutants start to be deleterious at low concentration while the single mutant T128N still has a log-fitness close to 0 (Fig. 8.6(c)). Therefore, we observe negative epistasis. At higher concentration, the single mutant T128N also begins to be deleterious. As seen in Figure 8.6(c), in this regime, the log-fitness decay is greater than for the mutants that started to be deleterious at lower concentrations. This explains why epistasis is non-monotonic. Furthermore, as discussed in Section 8.1, the log-fitness curves do not cross very much (Fig. 8.6(c)). Therefore, although non-monotonic, epistasis does not change sign. We also observe that some double mutants become deleterious after the T128N single mutant (Fig. 8.6(c)). The log-fitness decay of these mutants is lower than that of the single mutant, so we observe positive and monotonic epistasis (Fig. 8.6(b)).

The effects we observe are robust to changes in the time step of our inference, or else to redefining the times (Section 7.3). These effects are also captured by the two-state model (Fig. 8.7). We can see the distinction between mutations whose epistasis is stronger at high concentration than at low concentration (the blue dots of Fig. 8.7(a)), and mutations whose epistasis is stronger at low concentration than at high concentration (the orange dots of Fig. 8.7(a)). Within the model, single mutant T128N and double mutants associated with also begin to be deleterious at different concentrations, creating these specific patterns of epistasis.

Our model captures the overall effects, albeit imperfectly. Indeed, the effect of the model for the orange dots is more pronounced than in the data. Because the two-state model saturates at -0.6, the epistasis is even zero at high concentration (as single mutants and the double mutant associated with are lethal within the model), while it is non-zero at lower concentration. Concerning the blue dots, we retrieve the plateau we observe in Figure 8.6(a), but this one has now a slope lower than 1: the epistasis at low concentration is lower than the epistasis at high concentration (Fig. 8.7(a)). The model captures also that the highest epistasis at 10 g.L^{-1} correspond to high epistasis at 6 g.L^{-1}, as in the data.

More generally, by plotting the predictions of epistasis at different concentrations of the two-state model (Fig. E.4), we do find, in part, the specific patterns of epistasis present in the data (Fig. 8.5).

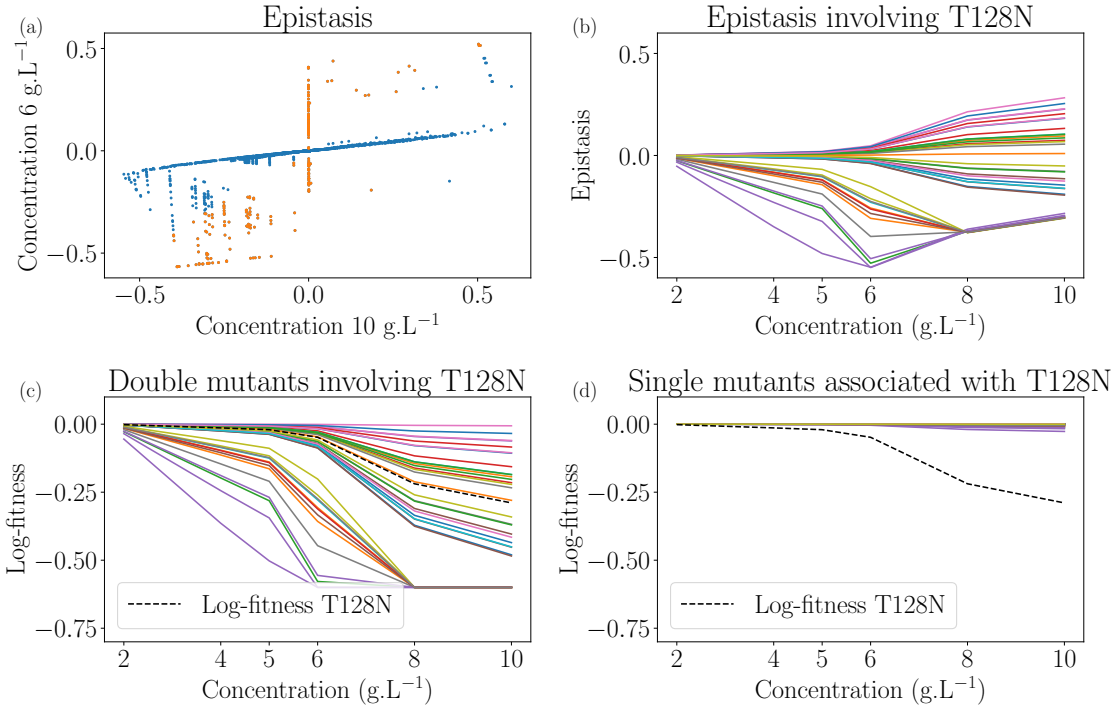


Figure 8.7: (a) Predicted epistasis at 6 g.L^{-1} against predicted epistasis at 10 g.L^{-1}. (b-d) Each color corresponds to a given mutant. Black dashed line corresponds to T128N. (b) Predicted epistasis involving T128N. (c) Predicted log-fitness of double mutants involving T128N. (d) Predicted log-fitness of single mutants associated with T128N.

8.3. Conclusion

The study of log-fitness and epistasis at different concentrations has highlighted several phenomena. There appears to be saturation phenomena of log-fitness as a function of the concentration. Up to a certain concentration threshold, some mutants have no deleterious effects on the protein. Beyond this threshold, they start to be deleterious. This saturation of log-fitness creates patterns when comparing epistasis at different concentrations. A first pattern, directly related to log-fitness saturation, is that epistasis at low concentrations may be zero, and become non-zero at high concentrations when the mutants in question become deleterious. A second, less intuitive reason is that epistasis at low concentration may be greater than at high concentration. This is because single and double mutants do not begin to be deleterious at the same concentration, what makes epistasis a non-monotonic function of concentration. The mentioned effects can be captured by a two-state model, where the effect of the concentration is captured by ΔG_0 depending on it. It appears that ΔG_0 is an increasing linear function of the logarithm of the concentration.

9

Analysis of class A β-lactamase families with Restricted Boltzmann Machines

In this chapter, we will use the compositional phase of RBM to analyze the β-lactamase class A family. As shown by Tubiana and Monasson (2017), in the compositional phase, RBM have sparse weights and each visible configuration is encoded by several (but finite) strongly activated hidden units (Section 2.5). These sparse weights can encode for different biological features, and unlike Potts' model where the interactions between amino acids are pairwise, in the case of RBM the weights take into account several amino acids and are therefore more easily interpretable.

As explained in Section 3.1.4.3 dedicated to Lattice Protein, couplings $\mathbf{W}$ between the hidden layer and the visible layer are represented by a $M \times N \times q$ tensor, as each amino acid is encoded by a Potts state with $q = 21$ colors. Therefore, the energy can be written as in Eq. (3.53). In practice, dReLU potentials are used for the hidden potentials (Section 1.2.1). As $\mathbf{W}$ is a third rank tensor, for a given hidden unit, we can extract a weight matrix associated with it: these weight matrices could encode the biological features (Figs. 9.1(b) and (c)). We use the same representation as in the thesis of Tubiana (2018): at each site i, the height of each amino acid a is proportional to the corresponding weight coefficient $W_{i\mu}(a)$. The amino acids are colored with the same color code as defined in Figure 5.1.

As for the Potts model, RBM are trained by maximizing the log-likelihood defined in Eq. (6.12), which considers the reweighting of the sequences (Section 6.1 for more details). We use the zero-sum gauge for the weights and fields.

The sparsity of weights does not emerge naturally during training, and the RBM is therefore not in the compositional phase. It isn't easy to interpret the features learned by the RBM from a biological point of view as they are very delocalized and intricate (Fig. (9.1)(b)). Therefore, in addition to a L_2 regularization on the g_i fields of the visible layer, a penalty L_1^2 is introduced during training on the couplings, where γ controls its intensity

$$\Delta \mathrm{LL} = \gamma \sum_{\mu} \left(\sum_{i,a} |W_{i\mu}(a)|^2 \right). \tag{9.1}$$

This penalty was introduced by Tubiana et al. (2019b), has several interests. It allows avoiding overfitting and to reach the compositional phase of the RBM. Nevertheless, if the regularization is too important, the log-likelihood of the model becomes low. There is thus a trade-off between a good representation (sparse weights) and a good performance of the model (a high log-likelihood), see Fig. 9.1.

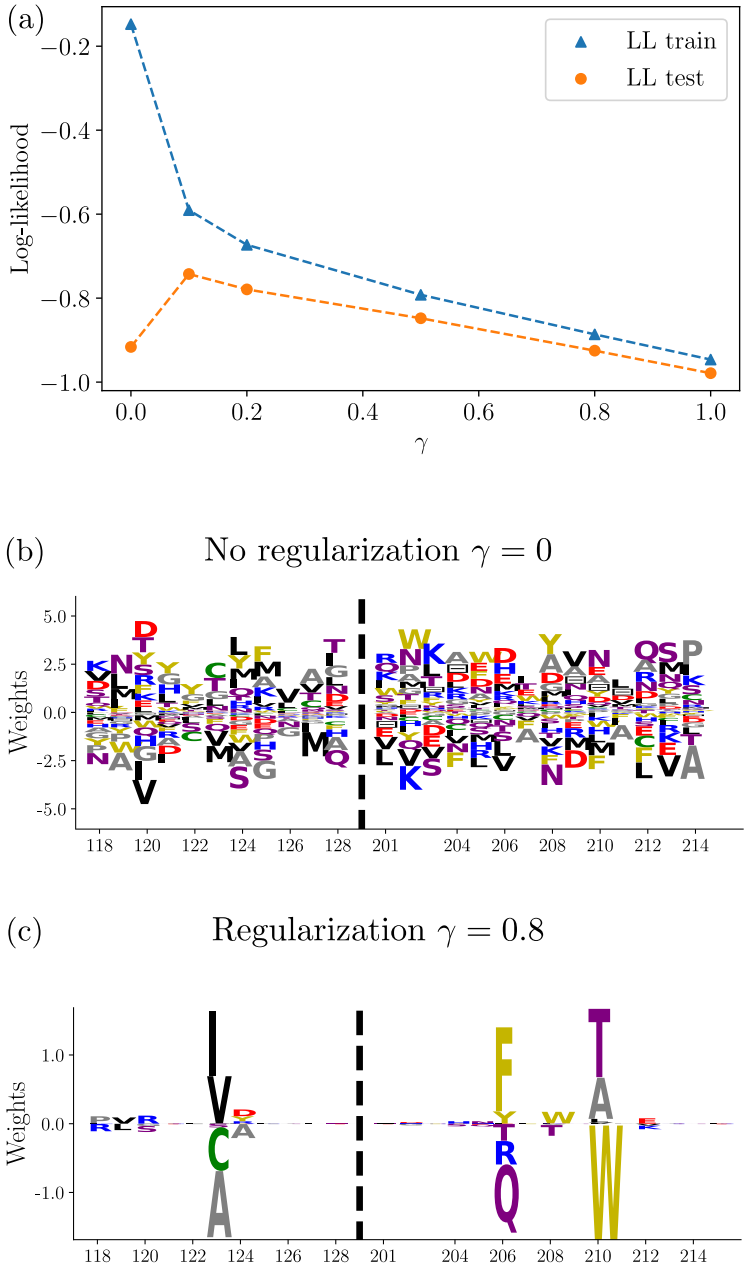


Figure 9.1: (a) Train log-likelihood (on training data) and test log-likelihood (on test data, not seen during the training) for different training with different γ evaluated with Annealed Importance Sampling (Section 1.2.4). There is an optimal $\gamma = 0.1$. However, we use a stronger regularization to have sparse weights, causing a slight decrease in the log-likelihood. (b) Example of weights for a given hidden unit ($\gamma = 0$). Weights are delocalized and can not be interpreted. (c) Example of weights for a given hidden unit ($\gamma = 0.8$). Weights are sparse and can be interpreted.

9.1. Description of class A β-lactamase

Before turning to the RBM weights, we will begin by briefly describing the β-lactamases. Two parallel classifications exist for them. The first one was introduced by Bush et al. (1995); Bush and Jacoby (2010) and is based on the functional characteristics of the proteins, *i.e.*, the type of drugs that proteins are able to hydrolyze. This classification has

three different classes, from 1 to 3, with a consequent number of subclasses.

The second one was introduced by Ambler et al. (1980, 1991) and is based on the similarity of the sequences. This classification has four different classes, from A to D. Classes A, B and D are composed of serine enzymes[1], with different 3D conformations for the three classes. Class B is composed of metalloenzymes[2], subdivided into 3 subclasses (B1 to B3).

These two classifications are compatible, and there are regularities from one to the other: class 1 is composed of class C, class 2 of classes A and D, and class 3 of class D.

TEM-1 β-lactamase belongs to class A as defined by Amber and to class 2b as defined by Bush. Class A β-lactamases contain highly conserved motifs linked to catalytic mechanisms and substrate binding: S70xxK[3], S130DN, K234TG. This conservation is visible in the sequence logo, as shown in Figure 5.1(b) for the S130DN motif. Class A is also characterized by an Ω-loop (from site 161 to 179), where the E166 site plays a crucial role in the catalytic cycle of hydrolysis (Egorov et al., 2019).

More recently, Philippon et al. (2016, 2019) proposed a refinement of the A class initially introduced by Amber: class A is subdivided into two classes A1 and A2. These two classes have many similarities in their sequences, but differ in others, allowing them to be separated. Class A1 is itself divided into 5 groups (group B to group F). The differences between these subgroups are due to residues that change the action spectrum of β-lactamases (from limited-spectrum β-lactamases to wider spectrum β-lactamases). It is possible to partly relate these classes to phylogeny, as shown in Figure 9.2: classes A1 and A2 do not correspond to the same phylum[4]. Group D belongs to a different phylum than the other groups of class A1 because it includes proteins from Gram-positive bacteria[5]. Group B, C, E, and F are Gram-negative bacteria and belong to the same phylum.

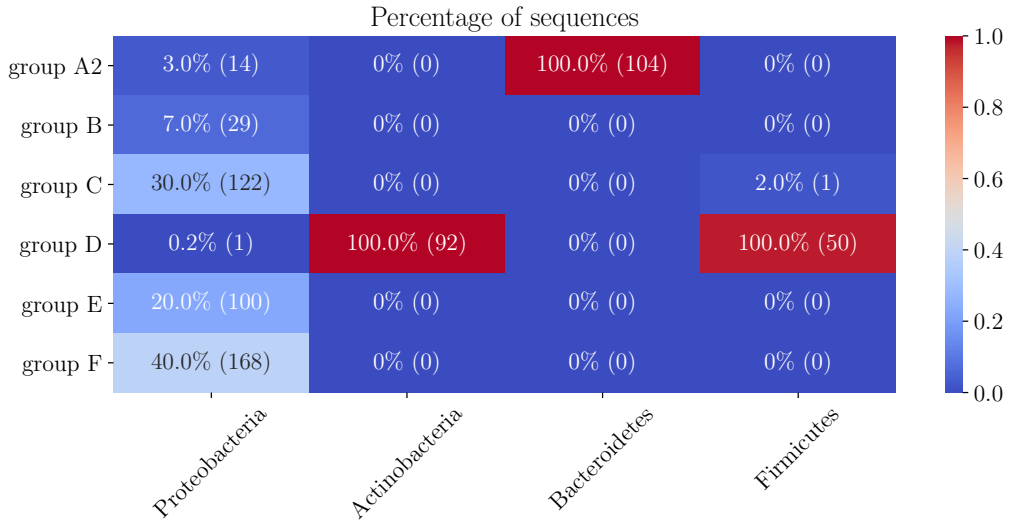


Figure 9.2: Percentage of given phylum in the different groups. The sum of the percentages in a given column is equal to 100%, and the number of sequences is indicated in brackets.

[1]The active site of the protein contains a serine.
[2]The active site of the protein contains zinc ions.
[3]x denotes a variable site
[4]Phylum is a level of classification below kingdom. Here, all proteins belong to the bacteria kingdom.
[5]Gram-positive bacteria have only one membrane, contrary to Gram-negative bacteria which have two membranes.

9.2. Results

We will show in this section that RBM are able to identify through their weights relevant information concerning the phylogeny, the functionality, and the structure of class A β-lactamases. As our MSA has 253 sites, we highlight only the most important sites, which are selected automatically: we show only sites i such that $\sum_{q=1}^{21} |W_{i\mu}(q)| \geq 0.5 \max_i \sum_{q=1}^{21} |W_{i\mu}(q)|$

9.2.1 A1 and A2 families

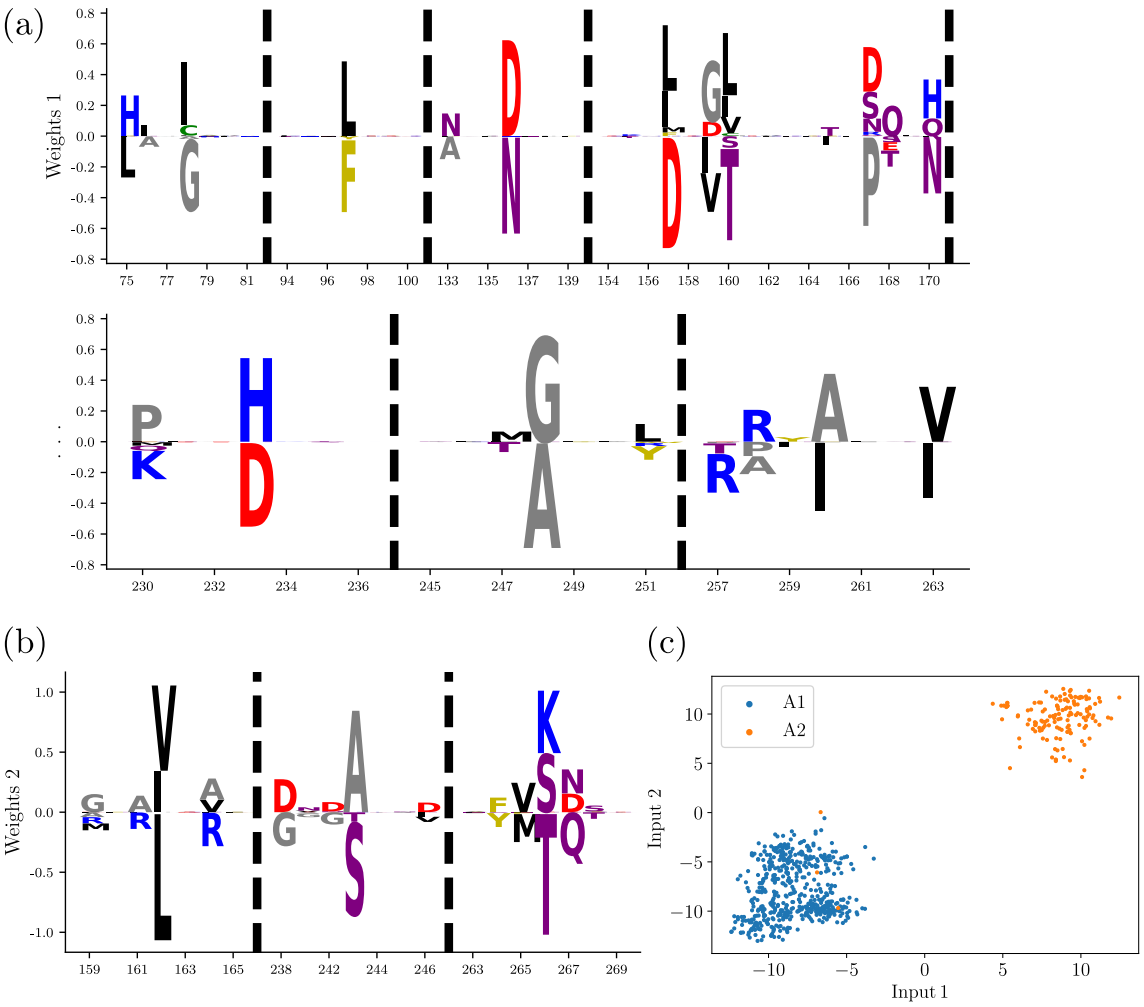


Figure 9.3: (a) and (b). Weight separating classes A1 and A2. (c) Scatter plot of the inputs of hidden units associated with the weights defined in (a) and (b).

We have identified in our trainings several weights allowing to separate the two families A1 and A2.

Weights shown in Figure 9.3(a) focus near the highly conserved motifs linked to catalytic mechanisms, substrate binding, and the Ω-loop. Philippon et al. (2019) identified that these sites allowed to discriminate the two families, A1 and A2. Among others, L75, N136, D157, T160, P167, D233 and G248 are highly conserved for sequences from class A1 and H75, D136, H233, G248 are highly conserved for sequences from class A2. These amino acids are indeed found by the RBM and therefore, input $I_1 = \sum_{i=1}^{253} W_{i1}(a_i)$ is positive for

sequences belonging to A2 and negative for sequences belonging to A1. Likewise, weights shown in Figure 9.3(b) focus on three sites. L162, T243 and T266 are highly conserved for sequences from class A1 and I162, A243 and S266 are highly conserved for sequences from class A2. Therefore, the inputs of the hidden units associated with these two weights make it easy to distinguish the two families A1 and A2 (Fig. 9.3(c)).

9.2.2 Gram-negative and Gram-positive bacteria

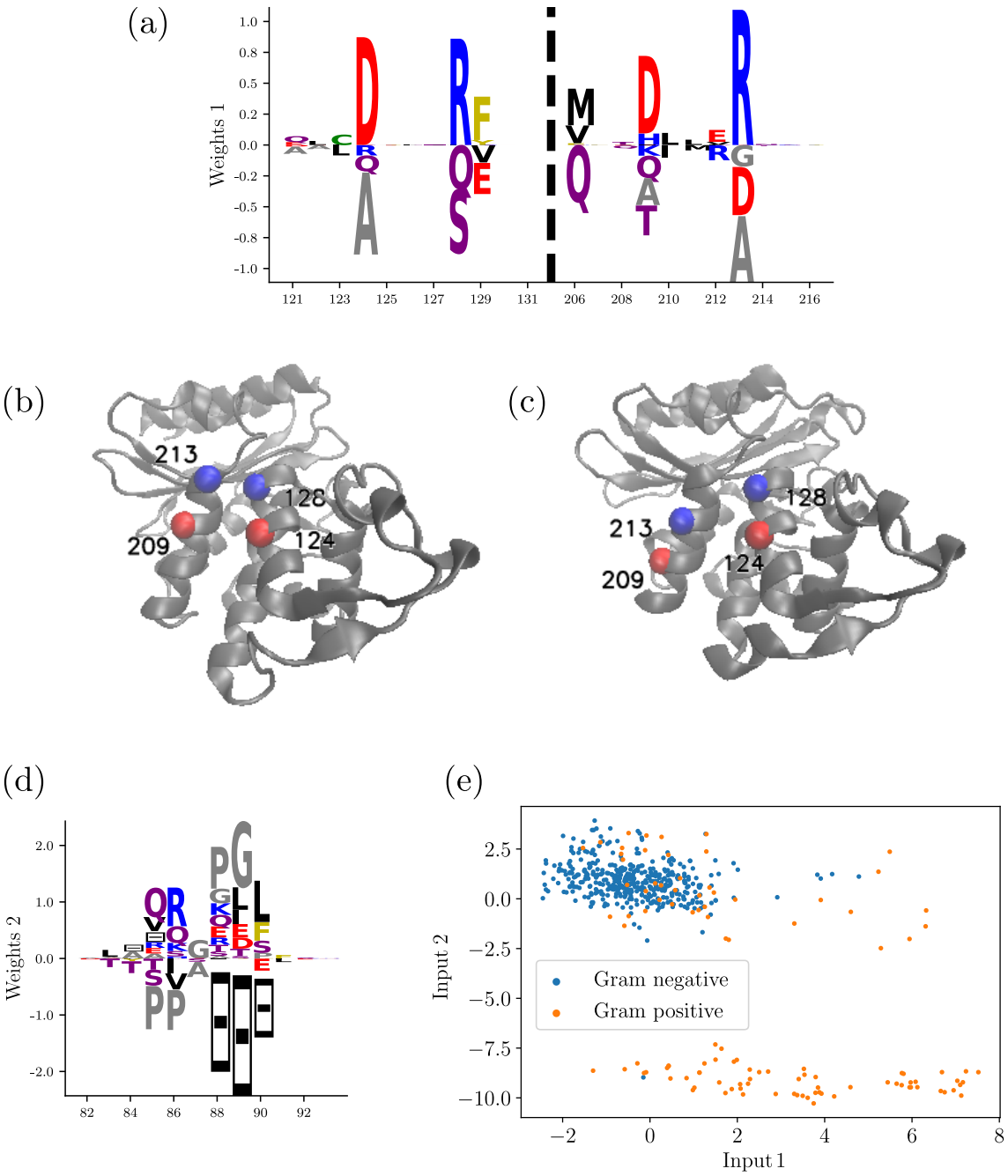


Figure 9.4: (a) Electrostatic modes. (b) TEM-1 structure (Gram-negative bacteria). PDB: 1BTL, Jelsch et al. (1993). (c) MYC1 structure (Gram-positive bacteria). PDB: 2GDN, Wang et al. (2006). (d) Gap modes. (e) Scatter plot of the inputs of hidden units associated with the weights defined in (a) and (d).

Weights shown in Figure 9.4(a) focus on two neighboring α-helices in the 3D conformation. We identify an electrostatic mode including sites 124, 128, 209 and 213. This mode is present for a large part of Gram-positive bacteria but not for Gram-negative bacteria (Fig. 9.4(e)). We have represented this mode on the 3D conformation of TEM-1, (Gram-negative bacteria, Fig. 9.4(b)). We see that this mode is not compatible with this structure, because the positive charges, as well as the negative charges, are face to face, which creates an electrostatic repulsion that destabilizes the structure. In contrast, for MYC1 (Gram-positive bacteria), the structure is slightly different: the negative charges are aligned with the positive charges (Fig. 9.4(c)). The electrostatic mode thus stabilizes the protein structure.

Weights shown in Figure 9.4(d) represent a gap mode. This mode corresponds to the insertion of three additional residues (sites 88 to 90) for the proteins found in Gram-negative bacteria. Therefore, during the sequence alignment procedure, proteins from Gram-positive bacteria have gaps at sites 88 to 91.

9.2.3 Several groups for Gram-negative bacteria

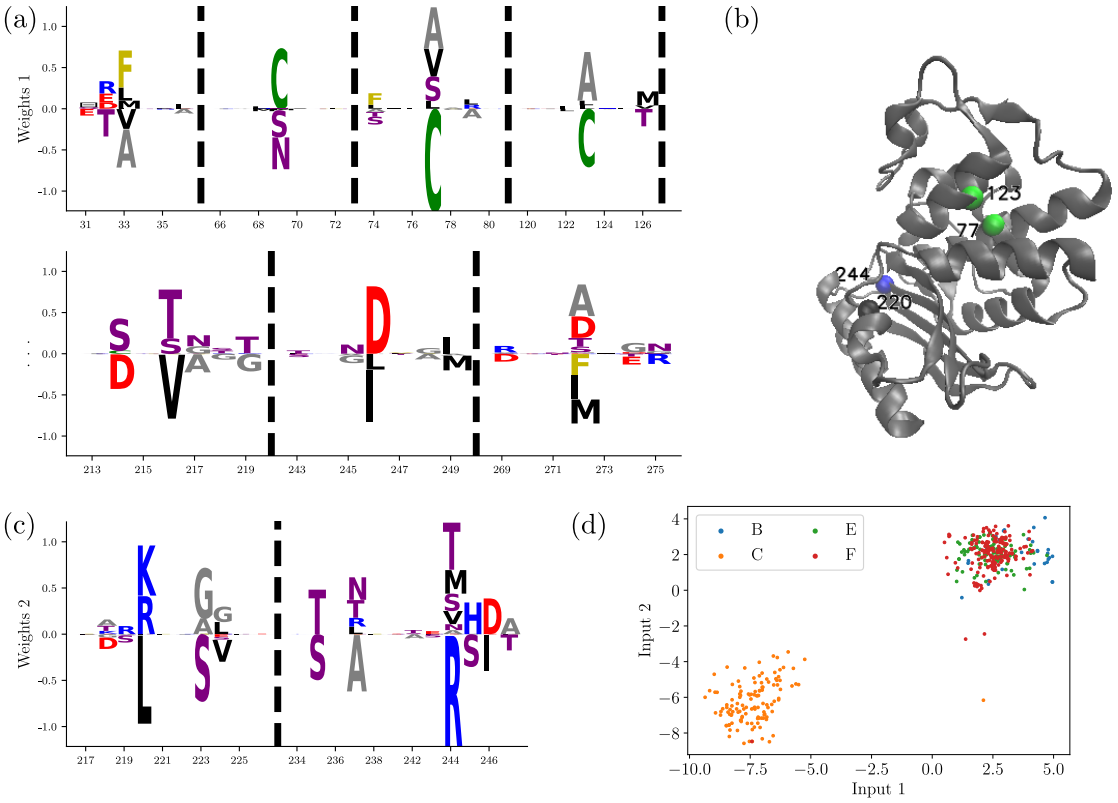


Figure 9.5: (a) Weights with a disulfide bond (C-C) between sites 79 and 123. (b) Structure of TEM-1 (group C). (c) Weights with couplings between sites 220 and 244. (d). Scatter plot of the inputs of hidden units associated with the weights defined in (a) and (c).

We are now interested in the proteins present in Gram-negative bacteria. These proteins are separated into four distinct groups: B, C, D and F (Philippon et al., 2019).

Group C proteins, of which TEM-1 is a member, are the limited-spectrum β-lactamases (LSBL). One of the characteristics that separates these proteins from the other groups is the strong conservation of C77 and C123 between two neighboring α-helices, supposedly forming a disulfide bond (Figs. 9.5(a) and (b)). Matagne et al. (1998) reported that if an

Arginine (R) is present at site 244, a Leucine (L) or Asparagine (N) is present at site 220 or 276. This is visible through the weights depicted in Figure 9.5(c). With the help of coupling between site 220 and 244, and the disulfide bond between sites 77 and 123, it is possible to separate the C group from the others.

Group E proteins are the wider-spectrum β-lactamases (WSBL). Among the highly conserved residues in this group E are Q128, Y129, F160 and T171. This mode is detected by a specific hidden unit and allows separating the group C from the others (Fig. 9.6).

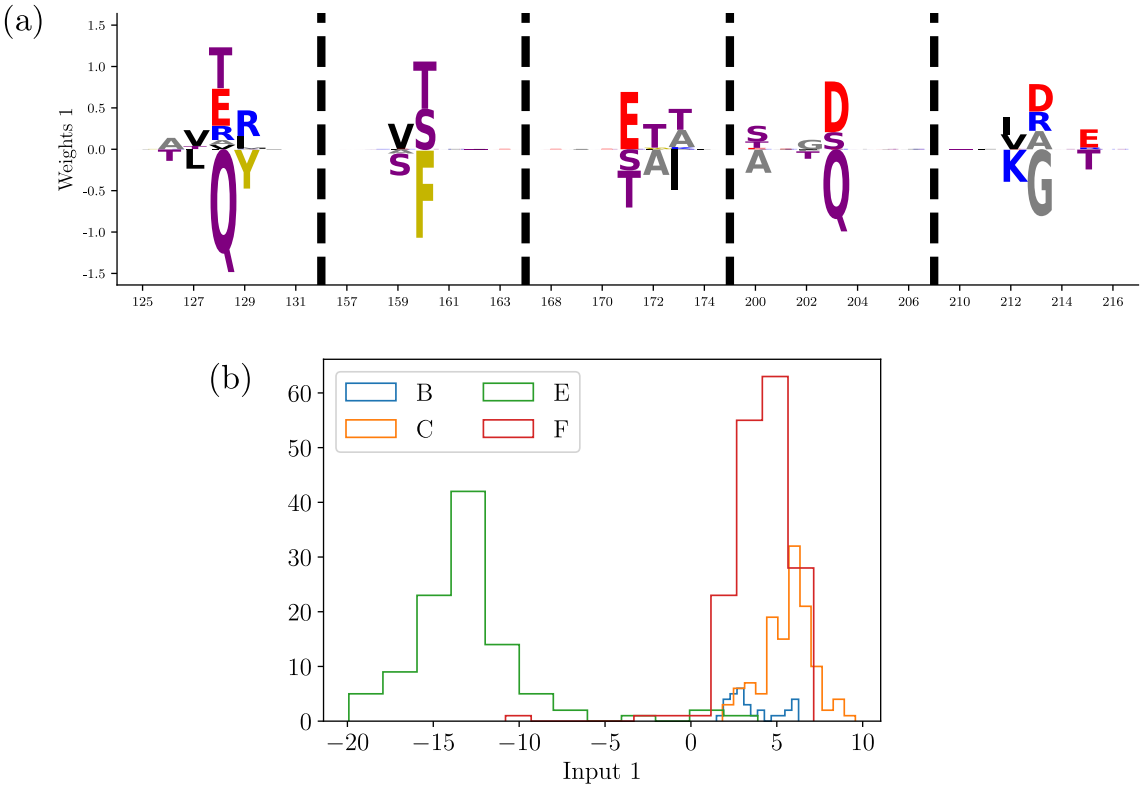


Figure 9.6: (a) Weights on sites 128, 129, 160 and 171. (b) Histogram of the input of hidden units associated with the weights defined in (a).

9.3. Comparison with Principal Component Analysis

Another approach to isolate groups of coevolved sites is to use Principal Component Analysis (PCA) (Pearson, 1901; Russ et al., 2005; Halabi et al., 2009; Rausell et al., 2010).

The first two components of the PCA separate the two subfamilies A1 and A2, as well as some subfamilies of class A1 (Fig. 9.7(b)), which shows that PCA is a useful method to separate subfamilies.

Nevertheless, the features learned through PCA are difficult to interpret for different reasons. The components of the PCA are highly delocalized, which makes the interpretation of biological features relatively difficult (Fig. 9.7(a)). Another drawback of PCA is that it assumes that the weights are orthogonal, which is not true in general: it is therefore difficult to decouple the true biological features with it. Therefore, RBM in the compositional phase, which can learn delocalized modes as well as pairwise couplings, turns out to be a more flexible and easily interpretable method than PCA. Nevertheless, PCA, because of its ease of use, remains an important tool in the detection of coevolutionary sites.

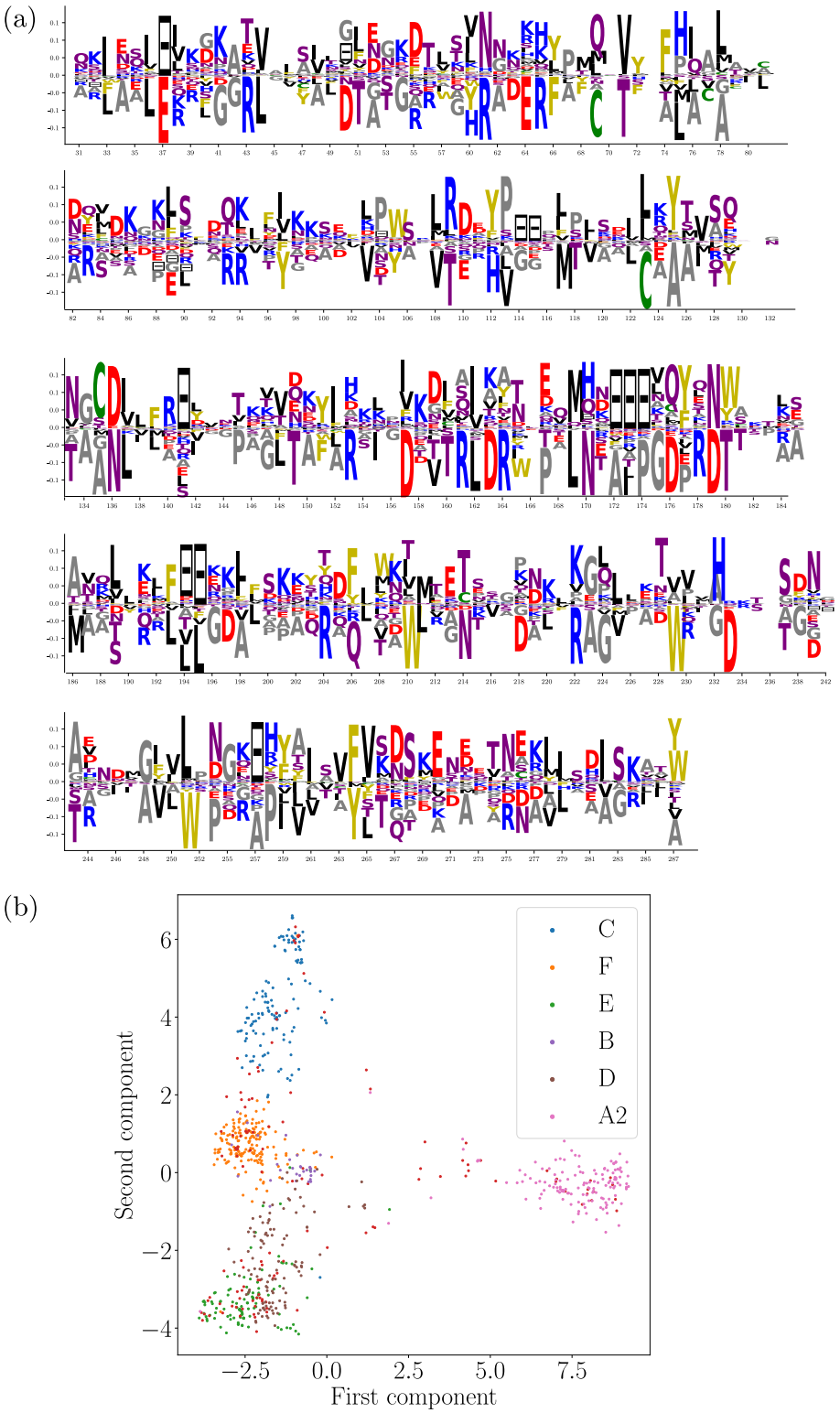


Figure 9.7: PCA on class A β-lactamases. (a) First component of the PCA. (b) Scatter plot of the projections of the sequences along the two first components of the PCA.

Conclusion and perspectives

In this Ph.D. thesis, we have studied the sampling properties of Restricted Boltzmann Machines. In the case of the canonical sampling algorithm of this neural network, the Alternating Gibbs Sampling, we have shown that it is possible to find optimal trajectories between local minima of the energy landscape, but that these trajectories pass through large free energy barriers. The characteristic time to go from one minimum to another is exponential in the size of the system. Therefore, this algorithm is just as inefficient as a naive sampling based on the Metropolis-Hastings algorithm.

We have also shown that it is possible to use the representations learned by the Restricted Boltzmann Machines to speed up the sampling. When hidden units encode essentially independent features of the data, or are block-correlated, updating of one, or a small number of hidden units with Metropolis-Hastings algorithm in the hidden space allows for a macroscopic change of visible units and offers rapid mixing between minima.

In the case of entangled representation, a large number of hidden units have to be updated simultaneously and therefore the Metropolis-Hastings algorithm in the hidden space is computationally prohibitive. In that case, to improve the sampling, we used a stack of coupled RBM via the Deep Tempering algorithm. By adding RBM on top of the initial RBM, while reducing their number of hidden units progressively and regulating the weights during their training, the representations of the initial RBM are progressively clustered, and the energy landscape becomes smoother with depth. Deep Tempering algorithm couples the RBM in the stack by allowing them to exchange configurations: swaps between the different layers allow combining the advantages of the top RBM fast-mixing properties and bottom RBM high-quality samples. In-between RBM are here to ensure a reasonable replica-exchange rate.

There is an extensive number of research directions to undertake regarding the sampling of RBM. First, a better understanding of the phenomenon of compression of representations would make it possible to improve its use on real data. Second, even though Alternating Gibbs Sampling suffers from poor mixing between far away minima, AGS, with Contrastive Divergence or Persistent Contrastive Divergence, remains an efficient training algorithm for RBM, as these two procedures authorize initialization of the dynamics in different local minima close to the training data. Nevertheless, these algorithms could be improved by including a dynamic that considers the representations identified by the RBM during training.

In this thesis, we were also interested in β-lactamase TEM-1 protein. Thanks to a collaboration with Olivier Tenaillon's group, we were able to study epistasis on a TEM-1 α-helix. We have shown that mutant's log-fitness are correlated with the energies of our inferred models on a sequence alignment in a nonlinear way, making it difficult to predict epistasis from the models. However, most mutations have a macroscopic pattern of epistasis which can be captured by a simple biophysical two-state model that predicts the emergence

of epistasis based on the additive effects of mutation. As the mutational effects on the phenotype are additive, we could fit the parameters only from the single mutational data, to predict epistasis with a good accuracy. Furthermore, the parameters of this model are linearly correlated with the energies of our inferred models on a sequence alignment. Idiosyncratic epistasis, not captured by the two-state model, may lead in the long term to some co-evolution patterns between pairs of sites. These interactions are not captured by the models at the level of interactions between amino acids, but are captured at the level of interaction between specific sites.

We also put forward saturation effects of log-fitness as a function of amoxicillin concentration in the medium. These effects create interesting patterns when comparing epistasis at different concentrations. We show that a two-state model can capture these effects if we allow ΔG_0 to depend on the concentration.

We were also able to isolate interesting residues motifs in the class A β-lactamase family using Restricted Boltzmann Machines. Just as Potts' models have shown their ability to design new proteins with a given functionality, it would be interesting to use RBM for this purpose. The advantage of RBM over Potts' model is that it is possible to fix the activity of some hidden units coding for a given protein subfamily, and thus in the case of β-lactamases effective against a given drug. In this case, it is theoretically possible to sample specifically this subfamily with the RBM, and thus to generate new proteins belonging to it.

To conclude, it would be interesting to use the advances we have made on RBM sampling for biological applications. The idea, which was initially the subject of this thesis, would be to sample transition paths between proteins using RBM and to see if this path can be interpreted from an evolutionary and phylogenetic perspective. This would require exploiting the representations learned by RBM and creating a biologically plausible sampling algorithm. This work is a long-term one, but I think it is worth the effort.

V

Appendix

Appendix summary

This part gathers various technical details concerning the previous chapters, as well as figures complementary to those exposed in the main text.

A

Appendix to Chapter 2

1.1. Spin-Spin RBM are universal approximators

A.1.1 Spin-Spin solution

The solution proposed in Le Roux and Bengio (2008) can be adapted in the case of Spin-Spin RBM. It can be done by setting

$$W_{i\mu} = \gamma \frac{a_\mu}{2} \xi_i^\mu, \tag{A.1}$$

$$c_\mu = \gamma(\lambda - \frac{N a_\mu}{2}), \tag{A.2}$$

$$g_i = \gamma \sum_{\mu=1}^{M} \frac{a_\mu}{2} \xi_i^\mu. \tag{A.3}$$

Then, in the case where $\gamma \to \infty$

$$\langle h_\mu | \mathbf{v} \rangle = \text{sign}\left(-a_\mu \text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) + \lambda\right), \tag{A.4}$$

where $\text{dist}(\mathbf{u}, \mathbf{v})$ is the Hamming distance between the vectors $\mathbf{u}$ and $\mathbf{v}$. a and λ, are tuned such that $\langle h_\mu | \mathbf{v} \rangle = 1$ if and only if $\mathbf{v} = \boldsymbol{\xi}^\mu$, so $a_\mu > \lambda > 0$.

And:

$$E^{\text{eff}}(\mathbf{v}) = \sum_{\mu=1}^{M} c_\mu \left(1 - 2 \prod_{i=1}^{N} \delta_{v_i, \xi_i^\mu}\right). \tag{A.5}$$

A.1.1.1 Increasing the size of basins of attractions

As each of the $\boldsymbol{\xi}^k$ has the same statistical weight, $a_\mu = a$ and we can rewrite Eq. (A.4)

$$\langle h_\mu | v \rangle = \text{sign}\left(-a \text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) + \lambda\right). \tag{A.6}$$

If $a = \frac{\lambda}{dN + \epsilon}$ where $\epsilon > 0$, we get that $\langle h_\mu | v \rangle = 1$ if and only if $\text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) \leq dN$. How can we choose the parameter d? To keep this "grandmother solution", we do not want that a vector $\mathbf{v}$ triggers two or more hidden units, so $dN < \frac{1}{2} \min_{\mu \neq \nu} \text{dist}(\boldsymbol{\xi}^\mu, \boldsymbol{\xi}^\nu)$ (because with this solution, a vector $\mathbf{v}$ which triggers two or more hidden unis could have a lower energy than the energy of the patterns).

For the energy we get $\forall \mathbf{v}$ such that $\forall \boldsymbol{\xi}^\mu, \text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) > dN$

$$E^{\text{eff}}(\mathbf{v}) \quad = \quad -\gamma(MN\frac{a}{2} - M\lambda) = \sum_{\mu=1}^{M} c_\mu. \tag{A.7}$$

And $\forall \mathbf{v}$ such that $\text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) = Nd' < Nd$:

$$E^{\text{eff}}(\mathbf{v})) \quad = \quad -\gamma(MN\frac{a}{2} - (M-2)\lambda - 2aNd'). \tag{A.8}$$

So the gap between $\mathbf{xi}^\mu$ and a vector $\mathbf{v}$ at distance $Nd' < Nd$ is $\frac{2\gamma\lambda Nd'}{Nd+\epsilon}$. The energy landscape $E(\mathbf{v})$ is flat, but now the K holes have some width.

In that case, $\forall \mathbf{v}$, such that $\text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) \leq dN$ $\mathbf{v}$ triggers h_μ. We can compute the relative weight of a vector $\boldsymbol{\xi}^k$ compared to the others vectors in the hole. We called $\mathbf{v}^{d'}$ a vector at distance d' of $\boldsymbol{\xi}^k$:

$$\frac{P(\boldsymbol{\xi}^k)}{\sum_{d'=0}^{dN} \binom{N}{d'} P(\mathbf{v}^{d'})} \quad \underset{dN \gg 1}{\simeq} \quad (1+\exp(-2a))^N \underset{a \gg 1}{\simeq} \exp(N\exp(-2a)). \tag{A.9}$$

As $a = \frac{\lambda}{dN+\epsilon}$, we can set λ such that $N\exp(-2a) \to 0$. In that case, sampling back the visible layer from h_μ leads to the pattern $\boldsymbol{\xi}^\mu$. Therefore, $\forall \mathbf{v}$, such that $\text{dist}(\mathbf{v}, \boldsymbol{\xi}^\mu) \leq dN$, $\mathbf{v}$ is in the basin of attraction of $\boldsymbol{\xi}^\mu$.

If the patterns are random vectors, $d \sim \frac{1}{4}$. Then, if $N \to \infty$, $\sum_{d'=0}^{Nd} \binom{N}{d'} < 2^{N\mathcal{S}(\frac{1}{4})}$ (with $\mathcal{S}$ is the binary entropy function). So the fraction of $\mathbf{v}$ in the basins of attraction of the patterns can be bounded by: $\frac{K2^{N\mathcal{S}(\frac{1}{4})}}{2^N} \underset{N \to +\infty}{\longrightarrow} 0$. The size of the basins of attraction are larger than in the previous case, but are still small compared to the total number of vectors.

B

Appendix to Chapter 3

2.1. General hidden-unit potentials

We consider below three different potentials acting on hidden units, and how they should scale when $N \to \infty$.

B.1.0.1 Quadratic potential

The quadratic potential is defined as $\mathcal{U}_\mu(h_\mu) = \frac{h_\mu^2}{2}$. In that case, we should rescale $h_\mu \to h_\mu/\sqrt{N}$. We get:

$$P(h_\mu|\mathbf{m}) = \frac{1}{\sqrt{2\pi/N}} \exp\left(-\frac{N}{2}(h_\mu - I_\mu)^2\right), \tag{B.1}$$

$$\hat{\Gamma}_\mu(I) = \frac{I^2}{2}, \quad f_\mu(I) = I. \tag{B.2}$$

B.1.0.2 ReLU potential

We can use the so-called ReLU (Rectified Linear Unit) potential $\mathcal{U}_\mu(h_\mu) = \frac{1}{2}\gamma^+ h_\mu^{+2} + \theta^+ h_\mu^+$ where $h_\mu^+ = \max(h_\mu, 0)$, see for instance (Tubiana et al., 2019b). We should rescale $h_\mu \to h_\mu/\sqrt{N}$ and $\theta_\mu^+ \to \theta_\mu^+/\sqrt{N}$. We get:

$$P(h_\mu|\mathbf{m}) = \mathcal{TN}\left(N\frac{I_\mu - \theta_\mu^+}{\gamma^+}, \frac{1}{\gamma^+}, \mathbb{R}^+\right), \tag{B.3}$$

$$\hat{\Gamma}_\mu(I) = \max\left(0, \frac{1}{2}\left(\frac{I-\theta_\mu^+}{\gamma_\mu^+}\right)^2\right), \quad f_\mu(I) = \max\left(0, \frac{I-\theta_\mu^+}{\gamma_\mu^+}\right). \tag{B.4}$$

$\mathcal{TN}(\mu, \sigma^2, \mathbb{R}^+)$ denotes the truncated Gaussian distribution of mode μ, width σ and support $\mathbb{R}^+$. This potential is called ReLU because its transfer function is a ReLU function.

B.1.0.3 Binary hidden units

If the hidden units are spins, *i.e.* $h_\mu \in \{-1, 1\}$, the potential can be written as a field $\mathcal{U}_\mu(h_\mu) = -c_\mu h_\mu$. In that case, we should rescale $c_\mu \to c_\mu/N$, $w \to w\sqrt{N}$. We get

$$P(h_\mu|\mathbf{m}) = \frac{1}{2}(1 + h_\mu \tanh(N(I_\mu + c_\mu))), \tag{B.5}$$

$$\hat{\Gamma}_\mu(I) = |I + c_\mu|, \quad f_\mu(I) = \text{sign}(I + c_\mu). \tag{B.6}$$

If the hidden units are Bernoulli units, *i.e.*, $h_\mu \in \{0, 1\}$, the potential acting on the hidden units is the same as for spins variables, and we get:

$$P(h_\mu|\mathbf{m}) = \frac{\exp\left(Nh_\mu(I_\mu+c_\mu)\right)}{1+\exp\left(N(I_\mu+c_\mu)\right)}, \tag{B.7}$$

$$\hat{\Gamma}_\mu(I) = \max(0, I+c_\mu), \quad f_\mu(I) = H\left(I+c_\mu\right). \tag{B.8}$$

$H(x)$ is the Heaviside step function.

2.2. Expansion of barrier height to first order in parameter changes

By using first order perturbation theory with the self-consistent equation defined in Eq. (3.36), we end up with:

$$\mathbf{m}^\alpha = \begin{bmatrix} g_\alpha(m_{11}, m_{22}) \\ g_\alpha(m_{22}, m_{11}) \end{bmatrix} \quad , \qquad \mathbf{m}^w = \begin{bmatrix} g_w(m_{11}) \\ g_w(m_{22}) \end{bmatrix} \quad , \tag{B.9}$$

with

$$g_\alpha(x,y) = \left(-\frac{x}{2}+\frac{x+y}{1+xy}\right)\left(\frac{2w^2(1-x^2)}{2-w^2(1-x^2)}\right), \tag{B.10}$$

$$g_w(x) = wx\left(\frac{2(1-x^2)}{2-w^2(1-x^2)}\right). \tag{B.11}$$

Inserting these results in the expression of $f(\mathbf{m})$ (Eq. (3.35)) leads to:

$$\begin{aligned} f^\alpha(\mathbf{m}) &= -\frac{w^2}{2}m_{11}\left(\frac{m_{11}+m_{22}}{1+m_{11}m_{22}}+\frac{g_\alpha(m_{11},m_{22})-m_{11}}{2}\right) \\ &- \frac{w^2}{2}m_{22}\left(\frac{m_{11}+m_{22}}{1+m_{11}m_{22}}+\frac{g_\alpha(m_{22},m_{11})-m_{22}}{2}\right) \\ &+ \frac{g_\alpha(m_{11},m_{22})}{2}\operatorname{arctanh}(m_{11})+\frac{g_\alpha(m_{22},m_{11})}{2}\operatorname{arctanh}(m_{22}) \\ &+ \frac{\mathcal{S}(m_{11})+\mathcal{S}(m_{22})}{2}-\mathcal{S}(m_{12}) \quad , \end{aligned} \tag{B.12}$$

$$\begin{aligned} f^w(\mathbf{m}) &= -\frac{w^2}{2}\left(m_{11}\frac{g_w(m_{11})}{2}+m_{22}\frac{g_w(m_{22})}{2}\right)-\frac{w}{4}\left(m_{11}^2+m_{22}^2\right) \\ &+ \frac{g_w(m_{11})}{2}\operatorname{arctanh}(m_{11})+\frac{g_w(m_{22})}{2}\operatorname{arctanh}(m_{22}) \quad . \end{aligned} \tag{B.13}$$

2.3. Sampling in the hidden space

Numerically, $P(h_\mu|\mathbf{h}_{\neg\mu})$ (Algorithm 4) and $P(h_\mu,h_\nu|\mathbf{h}_{\neg\mu,\nu})$ (Algorithm 5) are discretized, and the new candidate is drawn from the discretized distribution with the tower sampling algorithm (Krauth, 2006).

Let us denote the acceptance probability from a configuration $\mathbf{h}$ to a configuration $\mathbf{h}'$ by $A_h(\mathbf{h}\to\mathbf{h}')$. The Metropolis-Hastings algorithm and Gibbs sampling satisfy detailed balance in $E^{\text{eff}}(\mathbf{h})$, hence

$$P(\mathbf{h})A_h(\mathbf{h}\to\mathbf{h}') = P(\mathbf{h}')A_h(\mathbf{h}'\to\mathbf{h}'). \tag{B.14}$$

For the dynamics defined in Fig. 3.10, we have the following acceptance probability from a configuration $\mathbf{v}$ to a configuration $\mathbf{v}'$

$$A_v(\mathbf{v} \to \mathbf{v}') = \int d\mathbf{h} d\mathbf{h}' P(\mathbf{h}|\mathbf{v}) A_h(\mathbf{h} \to \mathbf{h}') P(\mathbf{v}'|\mathbf{h}'). \tag{B.15}$$

Therefore,

$$\begin{aligned} P(\mathbf{v}) A_v(\mathbf{v} \to \mathbf{v}') &= \int d\mathbf{h} d\mathbf{h}' P(\mathbf{v}) P(\mathbf{h}|\mathbf{v}) A_h(\mathbf{h} \to \mathbf{h}') P(\mathbf{v}'|\mathbf{h}') \\ &= \int d\mathbf{h} d\mathbf{h}' P(\mathbf{v}) \frac{P(\mathbf{v},\mathbf{h})}{P(\mathbf{v})} \frac{P(\mathbf{h}') A_h(\mathbf{h}' \to \mathbf{h})}{P(\mathbf{h})} \frac{P(\mathbf{v}',\mathbf{h}')}{P(\mathbf{h}')} \\ &= \int d\mathbf{h} d\mathbf{h}' P(\mathbf{v}|\mathbf{h}) A_h(\mathbf{h}' \to \mathbf{h}) P(\mathbf{v}',\mathbf{h}') \\ &= P(\mathbf{v}') A_v(\mathbf{v}' \to \mathbf{v}). \end{aligned} \tag{B.16}$$

As a consequence, our algorithm satisfies the detailed balance condition.

C

Appendix to Chapter 4

3.1. Detailed balance

In this part, we prove the detailed balance condition for Deep Tempering with two RBM. Moreover, the proof can be easily adapted to the general case of N RBM. First, we lighten the notation for the acceptance ratio

$$A_n\left(\{\mathbf{h}_n=\mathbf{h}_n^t,\mathbf{v}_{n+1}=\mathbf{v}_{n+1}^t\}\to\{\mathbf{h}_n=\mathbf{v}_{n+1}^t,\mathbf{v}_{n+1}=\mathbf{h}_n^t\}\right)\equiv A_n\left(\mathbf{h}_n^t,\mathbf{v}_{n+1}^t\right), \tag{C.1}$$

and we also get rid of temporal indices. First, the swap satisfies the detailed balance condition

$$P_{n+1}^v(\mathbf{v}_{n+1})P_n^h(\mathbf{h}_n)A_n\left(\mathbf{h}_n,\mathbf{v}_{n+1}\right)=P_{n+1}^v(\mathbf{h}_n)P_n^h(\mathbf{v}_{n+1})A_n\left(\mathbf{v}_{n+1},\mathbf{h}_n\right). \tag{C.2}$$

Now, we show that combining AGS and the swap also satisfy the detailed balance condition. We want to prove that

$$\begin{aligned} & P_n^v(\mathbf{v}_n)P_{n+1}^h(\mathbf{h}_{n+1})A\left(\{\mathbf{v}_n=\mathbf{v}_n,\mathbf{h}_{n+1}=\mathbf{h}_{n+1}\}\to\{\mathbf{v}_n=\mathbf{v}_n^{'},\mathbf{h}_{n+1}=\mathbf{h}_{n+1}^{'}\}\right)\\ = \;& P_n^v(\mathbf{v}_n^{'})P_{n+1}^h(\mathbf{h}_{n+1}^{'})A\left(\{\mathbf{v}_n=\mathbf{v}_n^{'},\mathbf{h}_{n+1}=\mathbf{h}_{n+1}^{'}\}\to\{\mathbf{v}_n=\mathbf{v}_n,\mathbf{h}_{n+1}=\mathbf{h}_{n+1}\}\right), \end{aligned}$$

where the acceptance ratio including AGS and the swap reads

$$\begin{aligned} & A\left(\{\mathbf{v}_n=\mathbf{v}_n,\mathbf{h}_{n+1}=\mathbf{h}_{n+1}\}\to\{\mathbf{v}_n=\mathbf{v}_n^{'},\mathbf{h}_{n+1}=\mathbf{h}_{n+1}^{'}\}\right) \\ = \;& \int d\mathbf{h}_n d\mathbf{v}_{n+1}P_n(\mathbf{h}_n|\mathbf{v}_n)P_{n+1}(\mathbf{v}_{n+1}|\mathbf{h}_{n+1})A_n\left(\mathbf{h}_n,\mathbf{v}_{n+1}\right)P_n(\mathbf{v}_n^{'}|\mathbf{h}^{'})P_{n+1}(\mathbf{h}_{n+1}^{'}|\mathbf{v}^{'}), \end{aligned} \tag{C.3}$$

where $\mathbf{v}^{'}=\mathbf{v}_{n+1}$ and $\mathbf{h}^{'}=\mathbf{h}_n$ if the swap does not occur and $\mathbf{v}^{'}=\mathbf{h}_n$ and $\mathbf{h}^{'}=\mathbf{v}_{n+1}$ otherwise. In the first case, the two RBM are independent and therefore, as AGS satisfies the detailed balance for each RBM, the detailed balance condition is fulfilled. For the latter case, we have

$$
\begin{aligned}
& P_n^v(\mathbf{v}_n)P_{n+1}^h(\mathbf{h}_{n+1})P_n(\mathbf{h}_n|\mathbf{v}_n)P_{n+1}(\mathbf{v}_{n+1}|\mathbf{h}_{n+1}) \\
& \times A_n(\mathbf{h}_n,\mathbf{v}_{n+1})\,P_n(\mathbf{v}_n^{'}|\mathbf{v}_{n+1})P_{n+1}(\mathbf{h}_{n+1}^{'}|\mathbf{h}_n) \\
= \quad & P_n^v(\mathbf{v}_n)P_{n+1}^h(\mathbf{h}_{n+1})\frac{P_n(\mathbf{h}_n,\mathbf{v}_n)}{P_n^v(\mathbf{v}_n)}\frac{P_{n+1}(\mathbf{v}_{n+1},\mathbf{h}_{n+1})}{P_{n+1}^h(\mathbf{h}_{n+1})} \\
& \times\frac{P_{n+1}^v(\mathbf{h}_n)P_n^h(\mathbf{v}_{n+1})A_n(\mathbf{v}_{n+1},\mathbf{h}_n)}{P_{n+1}^v(\mathbf{v}_{n+1})P_n^h(\mathbf{h}_n)}\frac{P_n(\mathbf{v}_n^{'},\mathbf{v}_{n+1})}{P_n^h(\mathbf{v}_{n+1})}\frac{P_{n+1}(\mathbf{h}_{n+1}^{'},\mathbf{h}_n)}{P_{n+1}^v(\mathbf{h}_n)} \\
= \quad & P_n^v(\mathbf{v}_n^{'})P_{n+1}^h(\mathbf{h}_{n+1}^{'})P_n(\mathbf{v}_{n+1}|\mathbf{v}_n^{'})P_{n+1}(\mathbf{h}_n|\mathbf{h}_{n+1}^{'}) \\
& \times A_n(\mathbf{v}_{n+1},\mathbf{h}_n)\,P_{n+1}(\mathbf{h}_{n+1}|\mathbf{v}_{n+1})P_n(\mathbf{v}_n|\mathbf{h}_n).
\end{aligned}
\tag{C.4}
$$

Consequently, the detailed balance condition is fulfilled in this case, and therefore Deep Tempering satisfies the detailed balance.

3.2. Correlated patterns

C.2.1 Computation of x_c

We can compute the derivative of the log-likelihood with respect to y

$$
\frac{\partial\left(\frac{\mathrm{LL}}{N}\right)}{\partial y} = \frac{1}{2}\log\left(\frac{\cosh\left(w(1+x)\right)\cosh\left(w\Delta_-\right)}{\cosh\left(w(1-x)\right)\cosh\left(w\Delta_+\right)}\right). \tag{C.5}
$$

x_c is defined such that $\left.\frac{\partial\left(\frac{\mathrm{LL}}{N}\right)}{\partial y}\right|_{y=1} = 0$. Then,

$$
\frac{\cosh\left(w(1+x_c)\right)}{\cosh\left(w(1-x_c)\right)\cosh\left(w\Delta_+\right)} = 1. \tag{C.6}
$$

x_c is therefore the solution of the following self-consistent equation:

$$
\begin{aligned}
& \frac{1}{w}\cosh^{-1}\left(\frac{\cosh(w(1+x_c)}{\cosh(w(1-x_c)}\right) \\
= \quad & (1+x_c)\tanh\left(2\alpha w\tanh\left(\cosh^{-1}\left(\frac{\cosh(w(1+x_c)}{\cosh(w(1-x_c)}\right)\right)\right).
\end{aligned}
\tag{C.7}
$$

C.2.2 Numerical experiments

C.2.2.1 Evaluation of the partition function with Annealed Importance Sampling

To estimate the optimal $y^*(w,x,\alpha)$ with AIS, for a fixed w, x and α, we sample random vectors $\boldsymbol{\xi}^1$, $\boldsymbol{\xi}^2$ with correlation x. For $y\in[0,1]$, we sample random vectors $\hat{\boldsymbol{\xi}}^1$, $\hat{\boldsymbol{\xi}}^2$ with a correlation y. The weight matrix is given by Eq. (4.3). The partition function is evaluated with AIS. The optimal y^* is the one which maximizes the log-likelihood (Eq. (4.5)). In practice, the segment $[0,1]$ is discretized, and the procedure is repeated 25 times for each x. Dots in Fig. 4.5(c) are the mean value of the optimal y^*, and the shaded areas show the standard deviation of the optimal y^*.

C.2.2.2 Initial correlation

The weight matrix $\mathbf{W}^0$ is initialized with small random values chosen from a zero-mean Gaussian with a standard deviation σ. $I_\mu(\boldsymbol{\xi}^1)$ and $I_\mu(\boldsymbol{\xi}^2)$ have a bivariate normal distribution with x as covariance. Therefore, $P(I_\mu(\boldsymbol{\xi}^1)I_\mu(\boldsymbol{\xi}^2)>0) = 1 - \frac{\cos^{-1}(x)}{\pi}$.

By defining $\hat{\boldsymbol{\xi}}^1 = \text{sign}\left(\mathbf{W}^0 \cdot \boldsymbol{\xi}^1\right)$ and $\hat{\boldsymbol{\xi}}^2 = \text{sign}\left(\mathbf{W}^0 \cdot \boldsymbol{\xi}^2\right)$, we have

$$\begin{aligned} y^0 &= \langle h_\mu^1 = h_\mu^2 \rangle - \langle h_\mu^1 \neq h_\mu^2 \rangle \\ &= P(I_\mu(\boldsymbol{\xi}^1)I_\mu(\boldsymbol{\xi}^2)>0) - P(I_\mu(\boldsymbol{\xi}^1)I_\mu(\boldsymbol{\xi}^2)<0) \\ &= 1 - \frac{2\cos^{-1}(x)}{\pi}. \end{aligned} \tag{C.8}$$

C.2.2.3 Numerical results

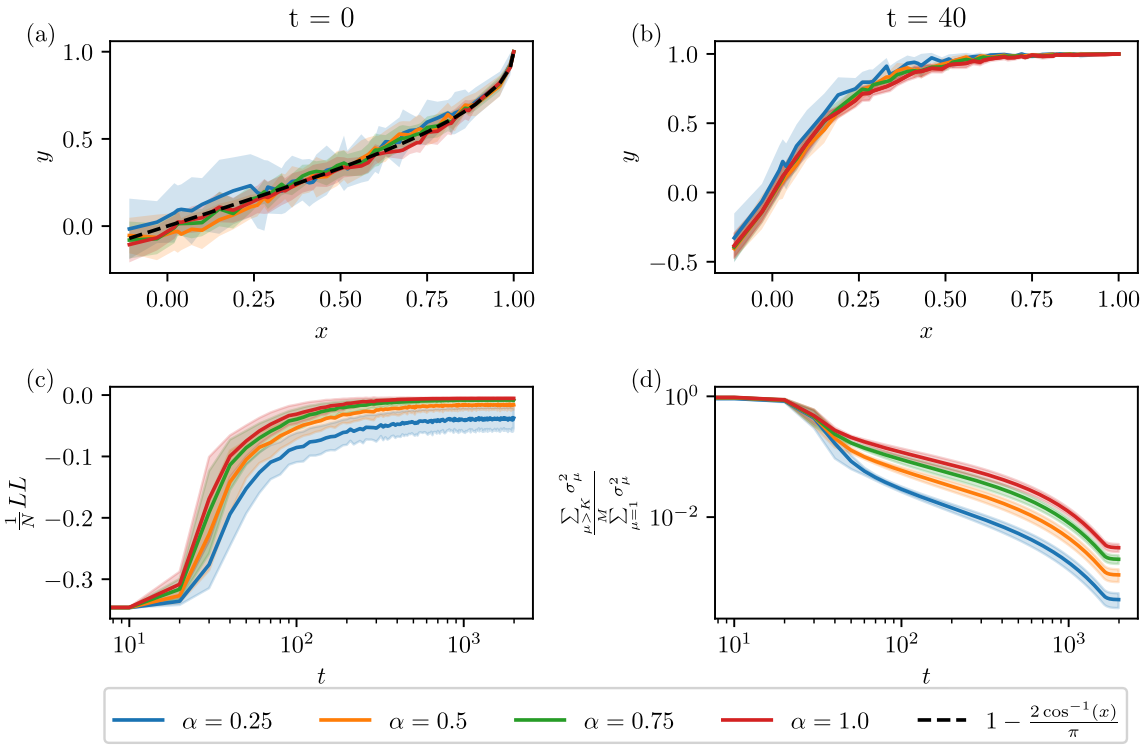


Figure C.1: Numerical experiments with $N = 200$. RBM are trained with Contrastive Divergence. For each value of x, 10 experiments are conducted with random vectors $\{\boldsymbol{\xi}^1, \boldsymbol{\xi}^2\}$ with a correlation x. Shaded areas depict the standard deviation. (a) Initial correlation y^0. Due to the initialization of the weight matrix, the initial correlation is equal to $1 - \frac{2\cos^{-1}(x)}{\pi}$ and does not depend on α. (b) Correlation y^t after 40 steps. During these first time steps, $\Delta_+ = \Delta_- = 0$. (c) Evolution of $\frac{1}{N}$LL through time. The results are averaged over all the correlation x. The dynamics reaches quickly a good solution. The convergence to the optimal solution is therefore very slow. (d) Evolution of $\left(\sum_{\mu>K}\sigma_\mu^2\right)\left(\sum_{\mu=1}^{M}\sigma_\mu^2\right)^{-1}$ through time. The M singular values σ_μ are ranking in descending order: K singular values dominate.

We train several RBM with Contrastive Divergence on $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$. After the training, two singular values dominate the SVD decomposition of the weight matrix (Fig. C.1(d)).

The first left eigenvector is aligned with $\boldsymbol{\xi}^1 + \boldsymbol{\xi}^2$ and the second one with $\boldsymbol{\xi}^1 - \boldsymbol{\xi}^2$. By writing the weight matrix in the basis $\{\boldsymbol{\xi}^1, \boldsymbol{\xi}^2\}$, the two first singular values are equal. We observe different phases during the training. First, the initial correlation y^0 is fixed by initializing the weight matrix and the correlation x (Fig. C.1(a)). Then, during the first steps of the training, y^t is increasing and independent of α (Fig. C.1(b)). Then, y^t depend on α (Fig. 4.5(c)). For finite time steps, y^t never reaches its optimal value but reaches a good value in terms of log-likelihood (Fig. C.1(c)).

3.3. Computation of the characteristic time scales

In this section, we derive the expression for the different time scales τ_1, τ_2 and τ_{swap}.

C.3.1 Computation of τ_1 and τ_2

In this part, we compute the characteristic times τ_1 and τ_2 for the bottom and the top RBM. τ_1 is the typical time scale between a transition between two modes $\boldsymbol{\xi}_1^1$ and $\boldsymbol{\xi}_1^2$ with the AGS for the bottom RBM. τ_2 is the typical time scale between a transition between two modes $\hat{\boldsymbol{\xi}}_1^1$ and $\hat{\boldsymbol{\xi}}_1^2$ with the AGS for the top RBM. Here, we compute τ_1. τ_2 has the same form, but one needs to change $M_1 \leftarrow N$, $M_2 \leftarrow M_1$ and $w_2 \leftarrow w_1$. As explained in Section 4.5.2, configurations aligned with one of the patterns have the lowest energy, and the second-lowest energy critical points of the landscape correspond to configurations aligned with three different patterns. Therefore

$$\log(\tau_1) = N\mathcal{B}(\alpha_1 w_1, K) = N(F_3 - F_1), \tag{C.9}$$

where F_r is defined in equation (4.21). This result is valid as long as the symmetric spurious patterns aligned with $r = 3$ patterns are saddle points of the energy landscape, which is the case in our numerical experiments.

$F_3 - F_1$ can be bounded by

$$F_3 - F_1 \quad \leq \quad \left(\log\cosh(\alpha w) - \frac{1}{4}\log\cosh(\frac{3}{2}\alpha w) - \frac{3}{4}\log\cosh(\frac{1}{2}\alpha w) \right).$$

In numerical experiments, the number of different optimal paths has to take into account in the barrier. This term decreases the barrier by a constant $-(K-2)\log(2)$.

C.3.2 Computation of τ_{swap}

In this part, we compute the characteristic time τ_{swap} between two replica exchanges between configurations $\mathbf{v}_2^t$ sampled by the bottom RBM and $\mathbf{h}_1^t$ sampled by the top RBM.

With AGS, the bottom RBM generates $\mathbf{v}_1^t$ and $\mathbf{h}_1^t$. The top RBM generates $\mathbf{v}_2^t$ and $\mathbf{h}_2^t$. The two chains are coupled with replica exchanges between $\mathbf{h}_1^t$ and $\mathbf{v}_2^t$ with an acceptance ratio $A_1(\mathbf{h}_1^t, \mathbf{v}_2^t)$ (Eq. (1.18)).

As $P_1^h(\hat{\boldsymbol{\xi}}_1^k)$ is very peaked around the $\hat{\boldsymbol{\xi}}_1^k$'s, $\mathbf{h}_1^t$ is equal to one of the $\hat{\boldsymbol{\xi}}_1^k$'s. It means, in order to have a swap, $\mathbf{v}_2^t$ must be equal to one of the $\hat{\boldsymbol{\xi}}_1^k$'s. Therefore, the mean value of acceptance ratio is equal to $\langle A_1(\mathbf{h}_1^t, \mathbf{v}_2^t)\rangle_{\{\mathbf{h}_1^t, \mathbf{v}_2^t\}} = \sum_{k=1}^{K} P_2^v(\hat{\boldsymbol{\xi}}_1^k)$. We do not take into account the energy term due to the hidden fields. This term is small compared to M_1 and can be neglect in this computation. As $E_2^{\mathbf{v}}(\hat{\boldsymbol{\xi}}_1^k) = -M_2\log(2\cosh(w_2))$, as $K \ll M_1$, for a configuration $\mathbf{h}_1^d$ at distance d of $\hat{\boldsymbol{\xi}}_1^k$, we have $E_2^{\mathbf{v}}(\mathbf{h}_1) = -M_2\log\left(2\cosh((1-\frac{2d}{M_1})w_2)\right)$ (we can neglect the effects of the others patterns which scale as $w_2\sqrt{\frac{2d(K-1)}{M_1}}$ in the hyperbolic cosine). We get:

$$\frac{P_2^v(\hat{\boldsymbol{\xi}}_1^k)}{\sum_{d=0}^{\frac{M_1}{2}} \binom{M_1}{d} P_2^v(\mathbf{h}_1^d)} \underset{w_2 \gg 1}{\simeq} (1+\exp(-2\alpha_2 w_2))^{-M_1}. \tag{C.10}$$

Thus, $P_2^v(\hat{\boldsymbol{\xi}}_1^k) = \frac{1}{K}(1+\exp(-2\alpha_2 w_2))^{-M_1}$ and we get that the mean acceptance ratio is equal to $\langle A_1(\mathbf{h}_1^t, \mathbf{v}_2^t)\rangle_{\{\mathbf{h}_1^t, \mathbf{v}_2^t\}} = (1+\exp(-2\alpha_2 w_2))^{-M_1}$. Therefore, we can time define the time scale

$$\tau_{\text{swap}} = \frac{1}{\langle A_1(\mathbf{h}_1^t, \mathbf{v}_2^t)\rangle_{\{\mathbf{h}_1^t, \mathbf{v}_2^t\}}} = \exp(M_1 \log(1+\exp(-2\alpha_2 w_2))). \tag{C.11}$$

C.3.3 Optimal $\alpha_2 w_2$

As τ_{swap} is an increasing function of $w_2\alpha_2$ and τ_2 is an decreasing function of $w_2\alpha_2$, the minimum of $\max(\tau_{\text{swap}}, \tau_2)$ is reached when $\tau_{\text{swap}} = \tau_2$. If the bound Eq. (C.10) is tight

$$\log(1+\exp(-2\alpha_2 w_2)) = \frac{1}{4}\left(\log(\cosh(\frac{3}{2}\alpha_2 w_2)) + 3\log(\cosh(\frac{1}{2}\alpha_2 w_2))\right). \tag{C.12}$$

Numerically, we find $\alpha_2 w_2 \simeq 0.771064$.

C.3.4 Numerical estimations of the characteristic time scales

We estimate τ_1 by sampling $\{\mathbf{v}_1^t, \mathbf{h}_1^t\}$ with AGS. τ_1 is the mean time spent by $\mathbf{v}_1^t$ in a given cluster $\mathcal{C}_k$ before going to another. τ_2 is estimated similarly with $\{\mathbf{v}_2^t, \mathbf{h}_2^t\}$.

We estimate τ_{swap} by computing the mean time between two replica exchanges with the Deep Tempering algorithm.

We estimate τ_{DT} by sampling configurations $\{\mathbf{v}_1^t, \mathbf{h}_1^t, \mathbf{v}_2^t, \mathbf{h}_2^t\}$ with Deep Tempering algorithm. τ_{DT} is the mean time spent by $\mathbf{v}_1^t$ in a given cluster $\mathcal{C}_k$ before going to another.

D

Appendix to Chapter 7

4.1. Comparison between log-fitness and MIC

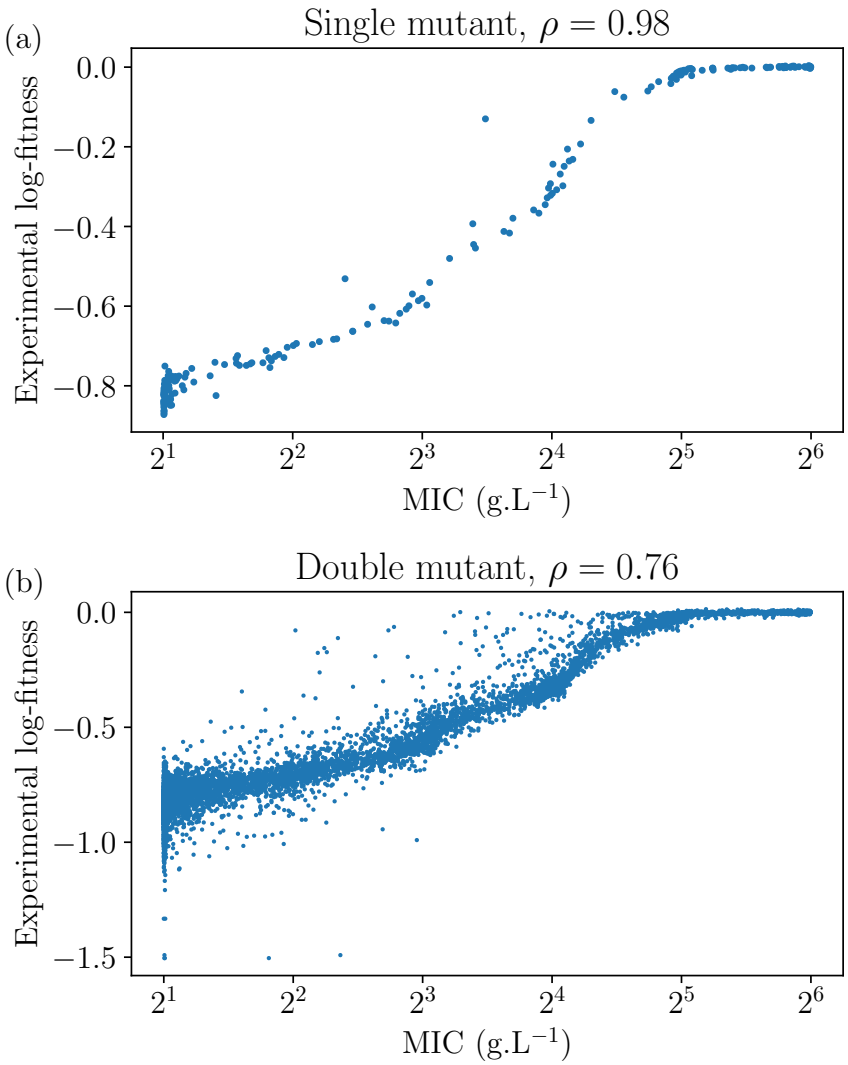


Figure D.1: Comparison between experimental log-fitness and MIC. (a) For single mutants. (b) For double mutants.

4.2. Comparison between two biological semi-replicates

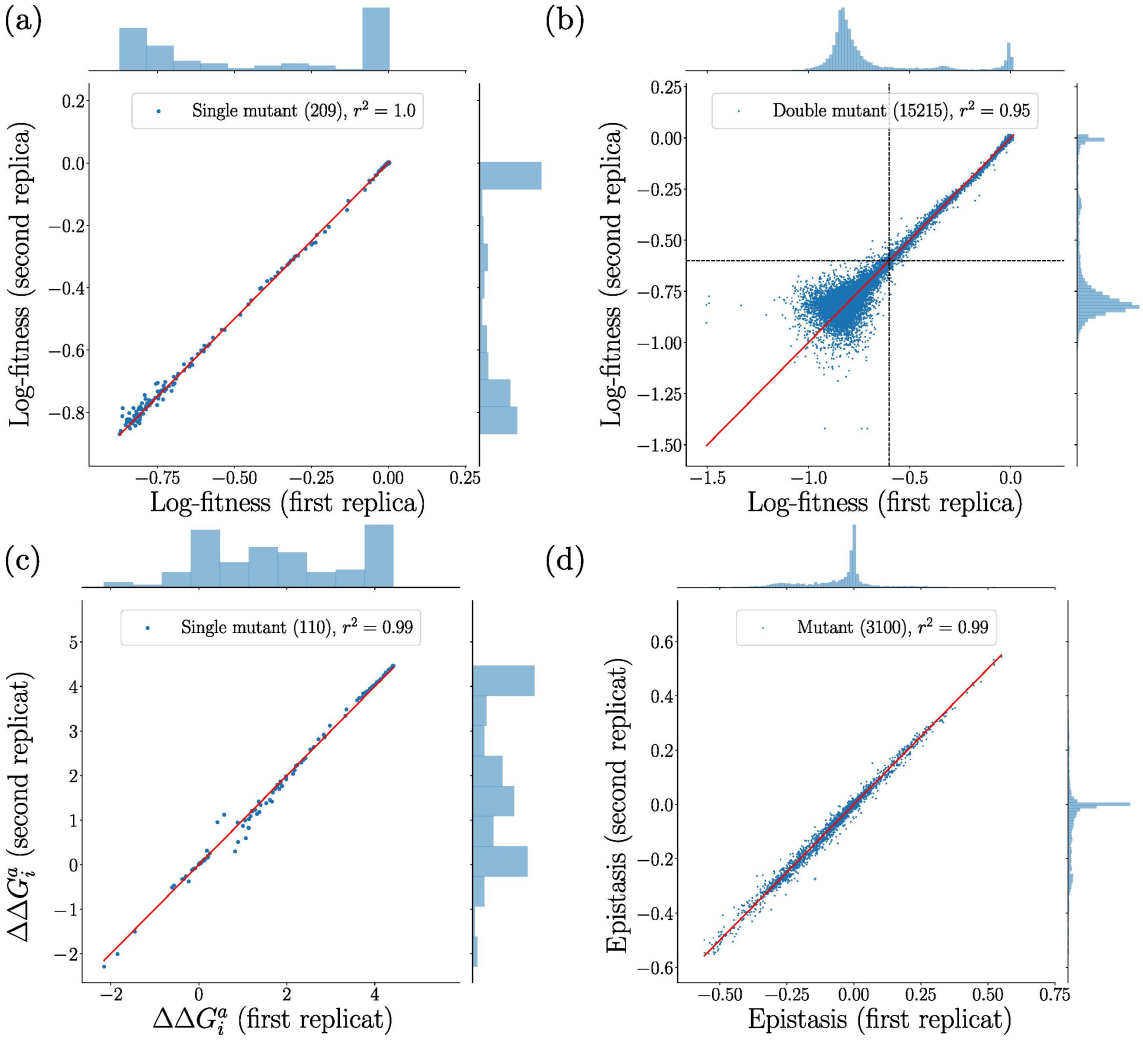


Figure D.2: Comparison between two biological semi-replicates. (a) Single mutants' log-fitness. (b) Double mutants' log-fitness. (c) $\Delta\Delta G$. (d) Epistasis.

4.3. Inference of the two-state model

We denote as $w_i(a)$ the relative fitness of the mutant that as the amino acid a at site i of the α-helix. We denote as $w_{i,j}(a,b)$ the relative fitness of the mutant that as the amino acid a at site i of the alpha-helix and amino acid b at site j.

The two-state model reads:

$$\log(\hat{w}_i^a) = \log\left(1+\exp\left(\frac{\Delta G_0}{RT}\right)\right) - \log\left(1+\exp\left(\frac{\Delta G_0 + \Delta\Delta G_i^a}{RT}\right)\right), \tag{D.1}$$

$$\log(\hat{w}_{i,j}^{a,b}) = \log\left(1+\exp\left(\frac{\Delta G_0}{RT}\right)\right) - \log\left(1+\exp\left(\frac{\Delta G_0 + \Delta\Delta G_i^a + \Delta\Delta G_j^b}{RT}\right)\right) \tag{D.2}$$

To fit the parameters, we had to assign every single mutant a free entropy value, $\Delta\Delta G$, and a value ΔG_0 characterizing the wild-type. Though measures of ΔG_0 have been done

in vitro, the cellular environment in which the mutants are evaluated could substantially affect the value. We, therefore, have also estimated ΔG_0. Ideally, estimated log-fitness is directly connected to $\Delta\Delta G$ for the single mutants, but is limited, as explained in the main text.

For the inference of the $\Delta\Delta G$ we keep only the single mutant with a log-fitness greater than -0.6. For each pair of previously chosen single mutants, the associated double mutant is kept if it exists. Its relative log-fitness is thresholded at -0.6. The two-state model is itself thresholded at -0.6 during the inference.

The model is over-constrained. By defining the residue associated with a given mutation

$$r_i = \frac{\log(w) - \log(\hat{w})}{\sigma_{\log(w)}}, \tag{D.3}$$

with the following cost function

$$C(\Delta G_0, \{\Delta\Delta G_i^a\}) = \frac{1}{2}\sum_i \alpha_i T^2 \Phi(\frac{r_i^2}{T^2}). \tag{D.4}$$

α_i is a statistical weight. For the double mutants, $\alpha_i = 2$. For the single mutants, α_i is equal to the number of double mutants with this single mutation. The weighting gives an equivalent weight for the single mutants and the double mutants.

We use $\Phi(x) = \arctan(x)$ in order to penalize the strong outliers. T is a threshold that controls the importance of the regularization of the outliers and is chosen such that 30% of the mutations are considered as outliers. The results are consistent for a wide range of thresholds T.

4.4. Estimation of the error part of the two-state model prediction

Using the whole data set, we could estimate an error to the model using a maximum likelihood framework. $\Delta\Delta G$ values were fixed. We estimated that the deviation of the observed log-fitness to the one predicted with the two-state model resulted from an overall random deviation from the model. This deviation to the model could be either the same for all pairs of mutations or could be different for residues in contact or not, *i.e.* a two model parameters. The two errors model was always much better than the single error, and always suggested a higher deviation to the model for residues in contact compared to distant residues.

For the single error model:

$$\sigma_m = \sqrt{N^{-1}\sum_{i,j,a,b}\left(\log(w_{i,j}(a,b)) - \log(\hat{w}_{i,j}^{a,b})\right)^2}, \tag{D.5}$$

where N^{-1} is the number of terms in the previous sum.

For the double error model:

$$\sigma_{md} = \sqrt{N_d^{-1}\sum_{i,j,a,b}(1-\delta_{i,j})\left(\log(w_{i,j}(a,b)) - \log(\hat{w}_{i,j}^{a,b})\right)^2}, \tag{D.6}$$

$$\sigma_{mn} = \sqrt{N_n^{-1}\sum_{i,j,a,b}\delta_{i,j}\left(\log(w_{i,j}(a,b)) - \log(\hat{w}_{i,j}^{a,b})\right)^2}, \tag{D.7}$$

with $\delta_{i,j}$ if the chains of residues carrying mutation i and j are less than 6Å away and 0 otherwise. $N_d = \sum_{i,j,a,b} (1-\delta_{i,j})$ and $N_n = \sum_{i,j,a,b} \delta_{i,j}$, $(N = N_d + N_n)$.

We used also a model with a specific error between two given sites:

$$\sigma_{i,j} = \sqrt{N_{i,j}^{-1} \sum_{a,b} \left(\log(w_{i,j}(a,b)) - \log(\hat{w}_{i,j}^{a,b})\right)^2}, \tag{D.8}$$

where $N_{i,j}$ is the number of double mutations between the sites i and j.

4.5. Results for independent model

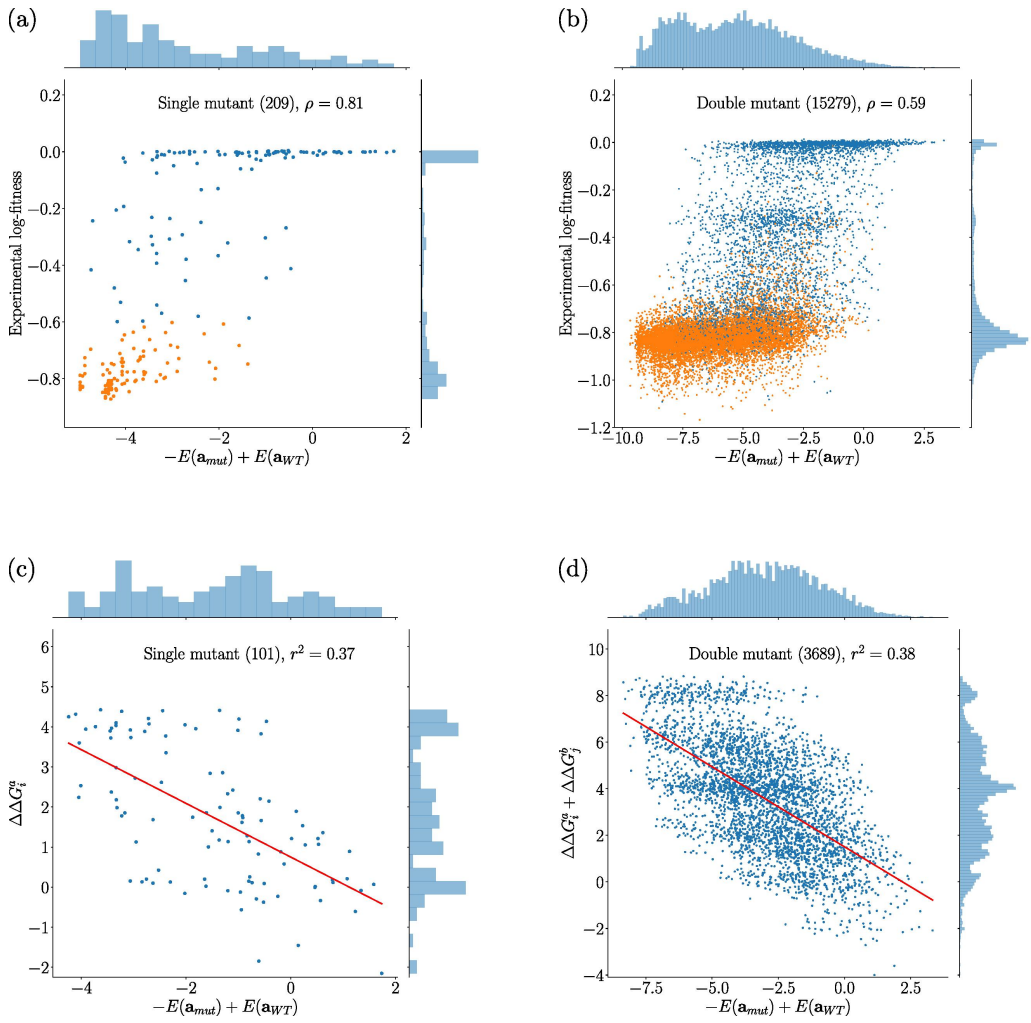


Figure D.3: Independent model' energies versus experimental quantities. Blue points are common mutations in panel (a) and (c) (respectively (b) and (d)), and correspond to the mutations we used to estimate $\Delta\Delta G$. Orange points are the other experimental mutations. (a) Experimental log-fitness against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$ for single mutants. (b) Experimental log-fitness against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$ for double mutants. (c) $\Delta\Delta G_i^a$ against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$. (d) $\Delta\Delta G_i^a + \Delta\Delta G_j^b$ against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$.

4.6. Results for RBM

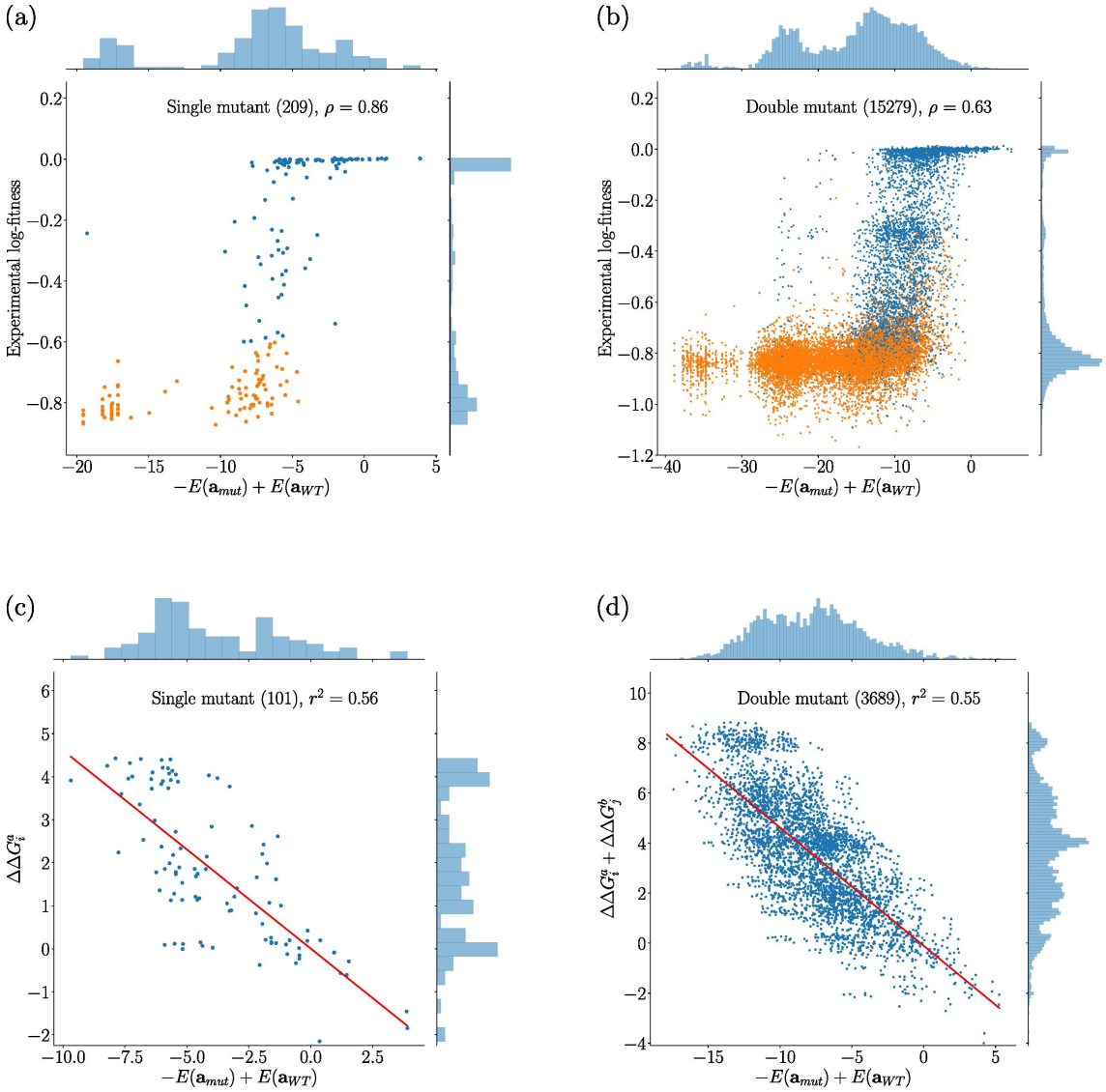


Figure D.4: RBM' energies versus experimental quantities. Blue points are common mutations in panel (a) and (c) (respectively (b) and (d)), and correspond to the mutations we used to estimate $\Delta\Delta G$. Orange points are the other experimental mutations. (a) Experimental log-fitness against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$ for single mutants. (b) Experimental log-fitness against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$ for double mutants. (c) $\Delta\Delta G_i^a$ against $-E(\mathbf{a}_{mut_i^a}) + E(\mathbf{a}_{WT})$. (d) $\Delta\Delta G_i^a + \Delta\Delta G_j^b$ against $-E(\mathbf{a}_{mut_{i,j}^{a,b}}) + E(\mathbf{a}_{WT})$.

4.7. Comparison between MIC and Potts' energies

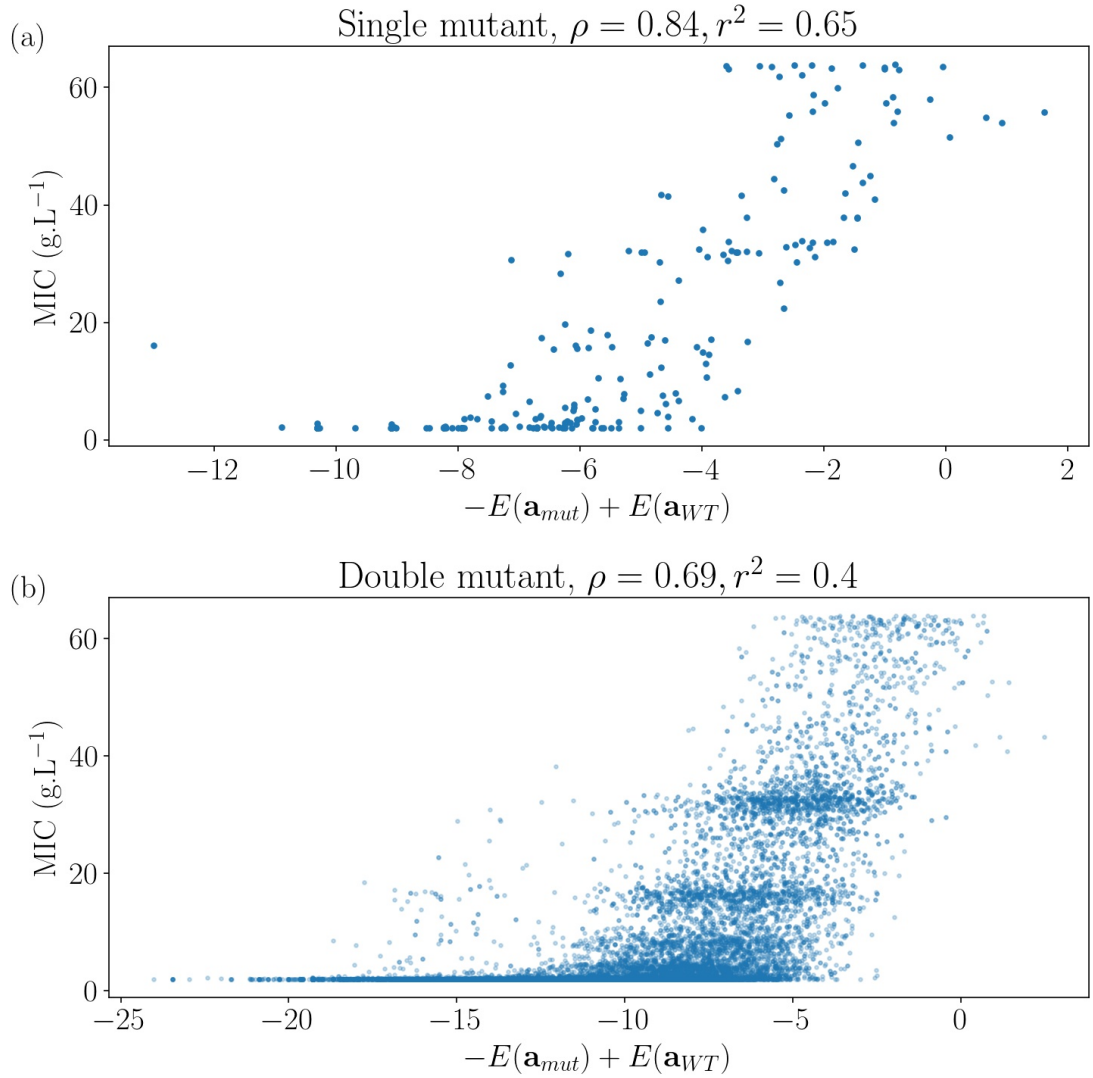


Figure D.5: Comparison between MIC and Potts' energies. (a) For single mutants. (b) For double mutants.

4.8. Computation of the p-value

If we choose as a null hypothesis that there is no relationship between pairs of sites with the largest idiosyncratic epistasis and pairs of sites with the largest Frobenius norm, the probability that among L pairs of sites with the largest idiosyncratic epistasis, c are also present in the L pairs of sites with the largest Frobenius norm follows an hypergeometric distribution with parameters N, L and c, where N is the total number of possible pairs. Therefore, the p-value reads

$$p = \sum_{k=c}^{L} \frac{\binom{L}{k}\binom{N-k}{L-k}}{\binom{N}{L}} \tag{D.9}$$

E

Appendix to Chapter 8

5.1. Comparison between log-fitness at different concentrations and MIC

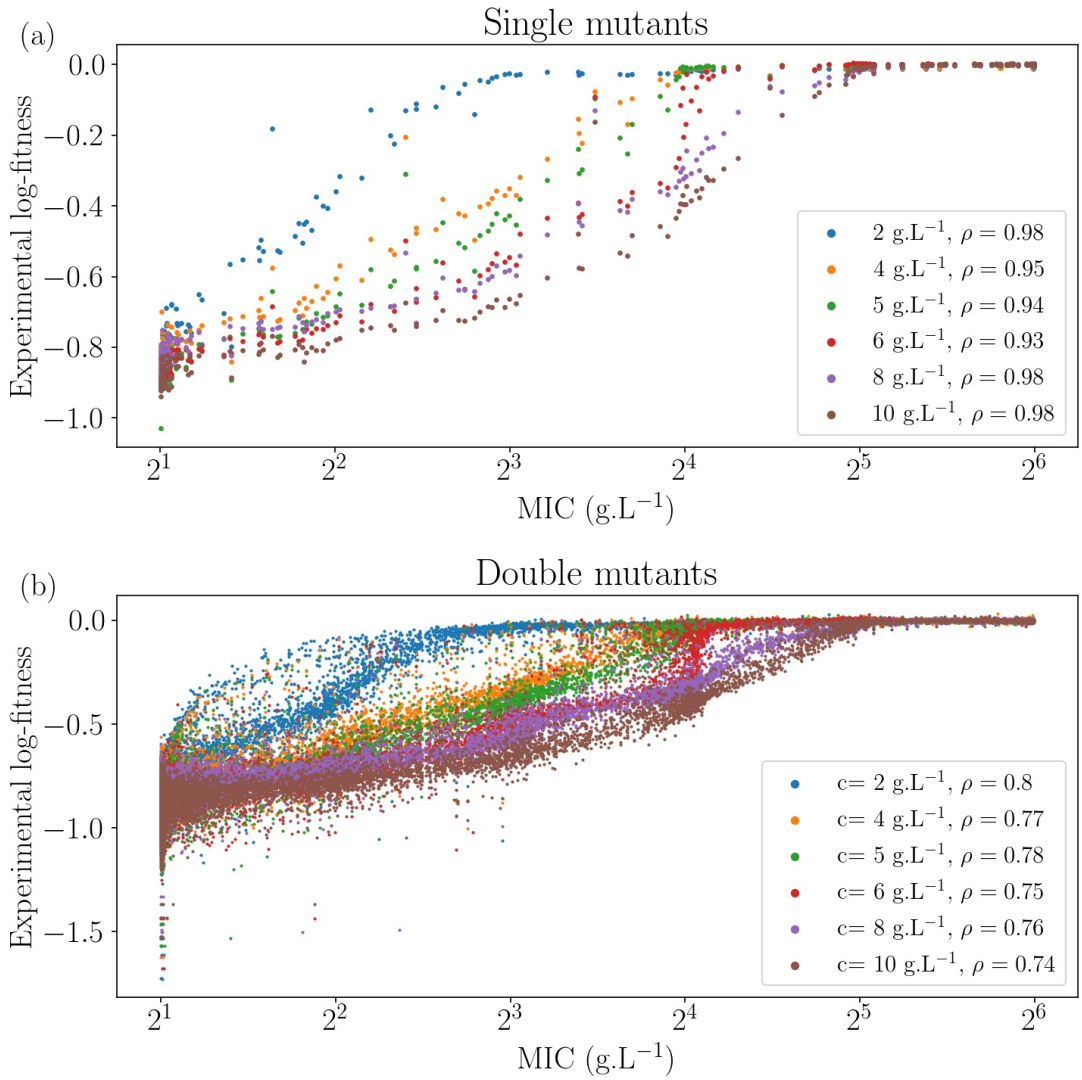


Figure E.1: Comparison between log-fitness at different concentrations and MIC. (a) For single mutants. (b) For double mutants.

5.2. Predicted log-fitness for several concentrations of amoxicillin

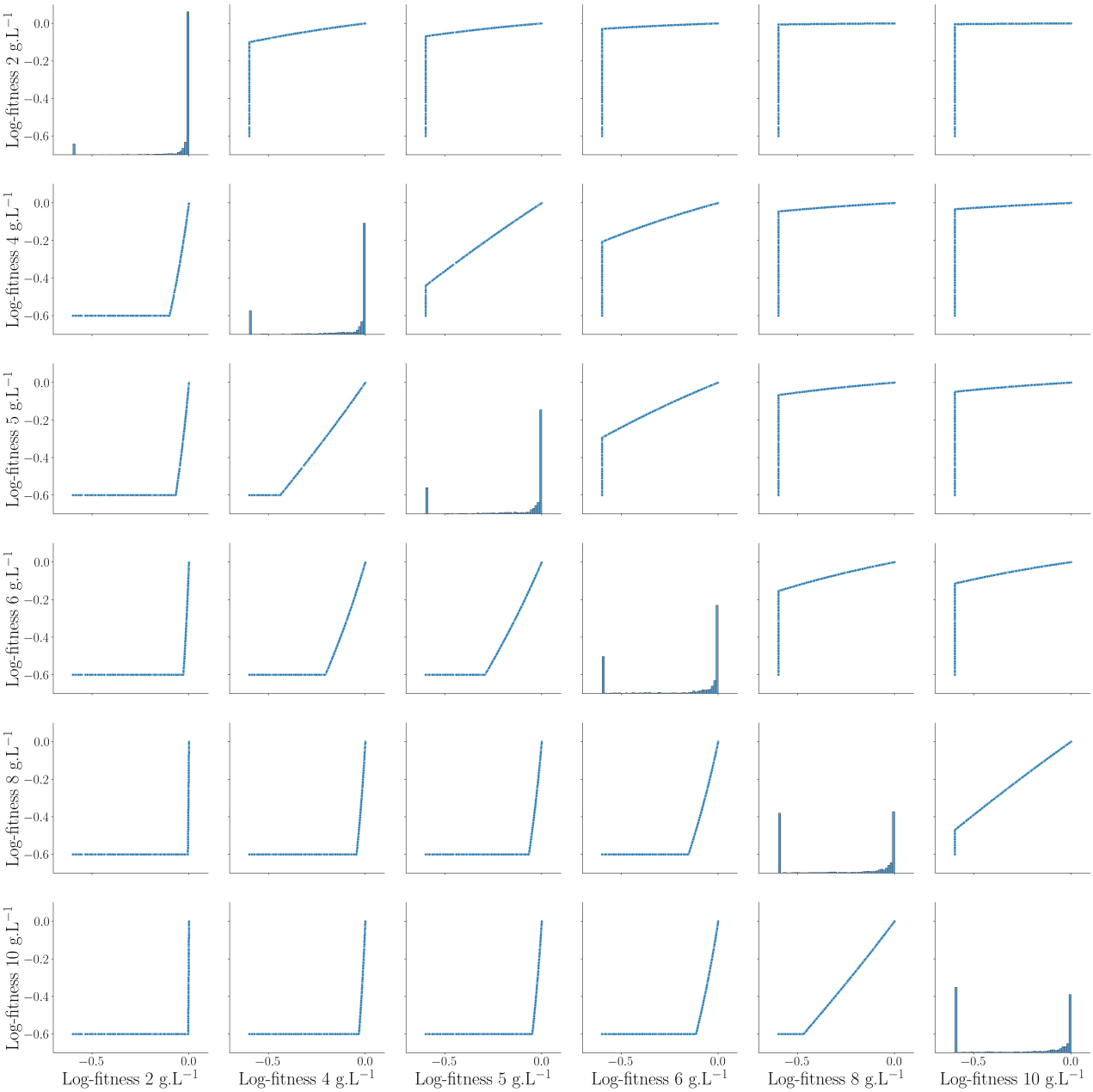


Figure E.2: Comparison of predicted log-fitness at concentrations 2, 4, 5, 6, 8, and 10 g.L^{-1} of amoxicillin.

5.3. Distribution of epistasis for several concentrations of amoxicillin

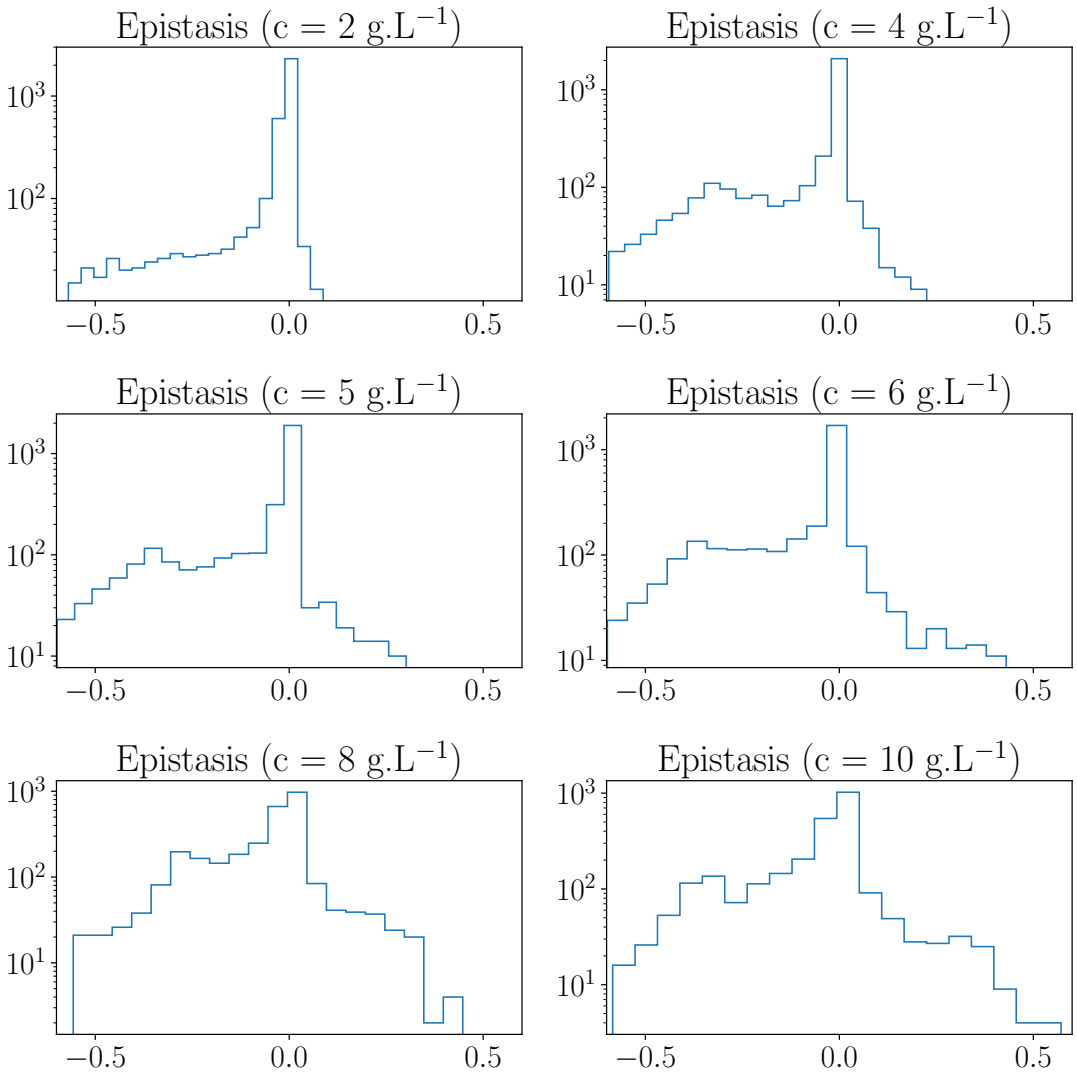


Figure E.3: Distribution of epistasis for 2, 4, 5, 6, 8, and 10 g.L^{-1} of amoxicillin.

5.4. Predicted epistasis for several concentrations of amoxicillin

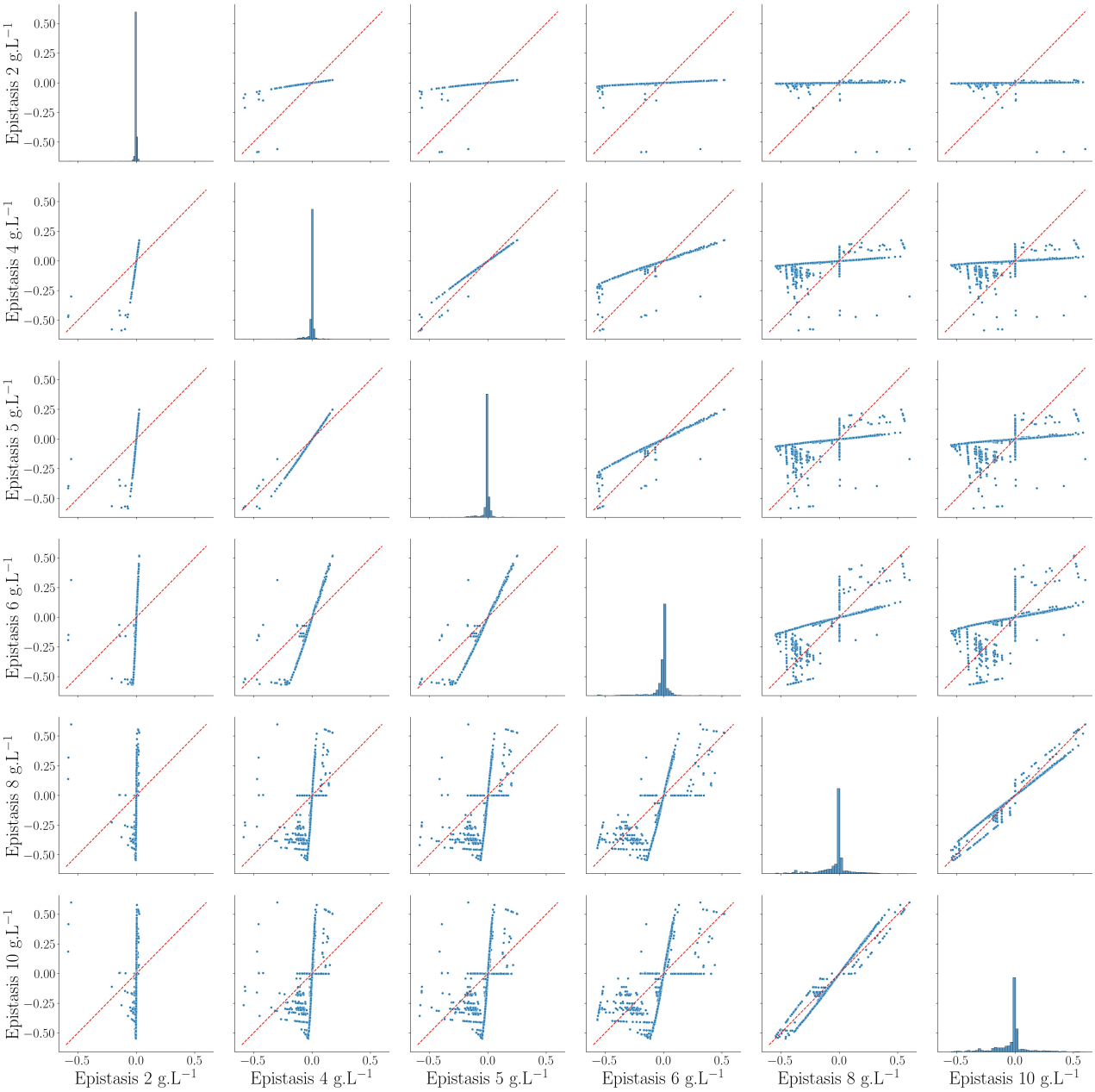


Figure E.4: Comparison of predicted epistasis at concentrations 2, 4, 5, 6, 8, and 10 g.L^{-1} of amoxicillin.

Bibliography

1. Abkevich, V. I., Gutin, A. M., and Shakhnovich, E. I. (1994). Specific nucleus as the transition state for protein folding: Evidence from the lattice model. *Biochemistry*, 33(33):10026–10036.

2. Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

3. Adzhubei, I., Jordan, D. M., and Sunyaev, S. R. (2013). Predicting Functional Effect of Human Missense Mutations Using PolyPhen-2. *Current protocols in human genetics*.

4. Agliari, E., Barra, A., Bartolucci, S., Galluzzi, A., Guerra, F., and Moauro, F. (2013). Parallel processing in immune networks. *Physical Review E*, 87(4):042701.

5. Agliari, E., Barra, A., Galluzzi, A., Guerra, F., and Moauro, F. (2012). Multitasking Associative Networks. *Physical Review Letters*, 109(26):268101.

6. Alberici, D., Barra, A., Contucci, P., and Mingione, E. (2020). Annealing and Replica-Symmetry in Deep Boltzmann Machines. *Journal of Statistical Physics*, 180(1):665–677.

7. Ambler, R. P., Baddiley, J., and Abraham, E. P. (1980). The structure of β-lactamases. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 289(1036):321–331.

8. Ambler, R. P., Coulson, A. F., Frère, J. M., Ghuysen, J. M., Joris, B., Forsman, M., Levesque, R. C., Tiraby, G., and Waley, S. G. (1991). A standard numbering scheme for the class A beta-lactamases. *Biochemical Journal*, 276(Pt 1):269–270.

9. Amit, D. J. (1989). Modeling Brain Function: The World of Attractor Neural Networks. Cambridge University Press, Cambridge.

10. Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985a). Spin-glass models of neural networks. *Physical Review A*, 32(2):1007–1018.

11. Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985b). Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks. *Physical Review Letters*, 55(14):1530–1533.

12. Anderson, P. W. (1972). More Is Different. *Science*, 177(4047):393–396.

13. Anfinsen, C. B. (1973). Principles that govern the folding of protein chains. *Science (New York, N.Y.)*, 181(4096):223–230.

14. Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein Generative Adversarial Networks. *ICML'17: Proceedings of the 34th International Conference on Machine Learning*.

15. Bank, C., Hietpas, R. T., Jensen, J. D., and Bolon, D. N. (2015). A Systematic Survey of an Intragenic Epistatic Landscape. *Molecular Biology and Evolution*, 32(1):229–238.

16. Bank, C., Matuszewski, S., Hietpas, R. T., and Jensen, J. D. (2016). On the (un)predictability of a large intragenic fitness landscape. *Proceedings of the National Academy of Sciences*, 113(49):14085–14090.

17. Barlow, H. B. (1972). Single Units and Sensation: A Neuron Doctrine for Perceptual Psychology? *Perception*, 1(4):371–394.

18. Barra, A., Bernacchia, A., Santucci, E., and Contucci, P. (2012). On the equivalence of Hopfield networks and Boltzmann Machines. *Neural Networks*, 34:1–9.

19. Barra, A., Genovese, G., Sollich, P., and Tantari, D. (2018). Phase diagram of restricted Boltzmann machines and generalized Hopfield networks with arbitrary priors. *Physical Review E*, 97(2):022310.

20. Barton, J. P., Cocco, S., De Leonardis, E., and Monasson, R. (2014). Large pseudocounts and L2-norm penalties are necessary for the mean-field inference of Ising and Potts models. *Physical Review E*, 90(1):012132.

21. Barton, J. P., De Leonardis, E., Coucke, A., and Cocco, S. (2016). ACE: Adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics (Oxford, England)*, 32(20):3089–3097.

22. Beck, M., Schmidt, A., Malmstroem, J., Claassen, M., Ori, A., Szymborska, A., Herzog, F., Rinner, O., Ellenberg, J., and Aebersold, R. (2011). The quantitative proteome of a human cell line. *Molecular Systems Biology*, 7:549.

23. Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. In Schölkopf, B., Platt, J., and Hoffman, T., editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

24. Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. (2013). Better Mixing via Deep Representations. In *International Conference on Machine Learning*, pages 552–560. PMLR.

25. 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.

26. Bershtein, S., Goldin, K., and Tawfik, D. S. (2008). Intense neutral drifts yield robust and evolvable consensus proteins. *Journal of Molecular Biology*, 379(5):1029–1044.

27. Bitbol, A.-F., Dwyer, R. S., Colwell, L. J., and Wingreen, N. S. (2016). Inferring interaction partners from protein sequences. *Proceedings of the National Academy of Sciences*, 113(43):12180–12185.

28. Blanquart, F., Achaz, G., Bataillon, T., and Tenaillon, O. (2014). Properties of selected mutations and genotypic landscapes under Fisher's geometric model. *Evolution*, 68(12):3537–3554.

29. Bloom, J. D., Silberg, J. J., Wilke, C. O., Drummond, D. A., Adami, C., and Arnold, F. H. (2005). Thermodynamic prediction of protein neutrality. *Proceedings of the National Academy of Sciences of the United States of America*, 102(3):606–611.

30. Bottou, L. (2010). Large-Scale Machine Learning with Stochastic Gradient Descent. In Lechevallier, Y. and Saporta, G., editors, *Proceedings of COMPSTAT'2010*, pages 177–186.

31. Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. (2012). Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, pages 1881–1888, Madison, WI, USA. Omnipress.

32. Bowers, J. S. (2011). What is a grandmother cell? And how would you know if you found one? *Connection Science*, 23(2):91–95.

33. Box, G. E. P. (1976). Science and Statistics. *Journal of the American Statistical Association*, 71(356):791–799.

34. Bravi, B., Balachandran, V. P., Greenbaum, B. D., Walczak, A. M., Mora, T., Monasson, R., and Cocco, S. (2021a). Probing T-cell response by sequence-based probabilistic modeling. *PLOS Computational Biology*, 17(9):e1009297.

35. Bravi, B., Tubiana, J., Cocco, S., Monasson, R., Mora, T., and Walczak, A. M. (2021b). RBM-MHC: A Semi-Supervised Machine-Learning Method for Sample-Specific Prediction of Antigen Presentation by HLA-I Alleles. *Cell Systems*, 12(2):195–202.e9.

36. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

37. Brocchieri, L. and Karlin, S. (2005). Protein length in eukaryotic and prokaryotic proteomes. *Nucleic Acids Research*, 33(10):3390–3400.

38. Brunetti, R., Parisi, G., and Ritort, F. (1992a). Asymmetric Little spin-glass model. *Physical Review B*, 46(9):5339–5350.

39. Brunetti, R., Parisi, G., and Ritort, F. (1992b). Study of the asymmetric Little model. *Physica A: Statistical Mechanics and its Applications*, 185(1):247–253.

40. Bush, K. and Jacoby, G. A. (2010). Updated functional classification of beta-lactamases. *Antimicrobial Agents and Chemotherapy*, 54(3):969–976.

41. Bush, K., Jacoby, G. A., and Medeiros, A. A. (1995). A functional classification scheme for beta-lactamases and its correlation with molecular structure. *Antimicrobial Agents and Chemotherapy*, 39(6):1211–1233.

42. Cadet, F., Fontaine, N., Li, G., Sanchis, J., Ng Fuk Chong, M., Pandjaitan, R., Vetrivel, I., Offmann, B., and Reetz, M. T. (2018). A machine learning approach for reliable prediction of amino acid interactions and its application in the directed evolution of enantioselective enzymes. *Scientific Reports*, 8.

43. Chen, K. and Arnold, F. H. (1993). Tuning the activity of an enzyme for unusual environments: Sequential random mutagenesis of subtilisin E for catalysis in dimethylformamide. *Proceedings of the National Academy of Sciences of the United States of America*, 90(12):5618–5622.

44. Cho, K., Raiko, T., and Ilin, A. (2010). Parallel tempering is efficient for learning restricted Boltzmann machines. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

45. Chou, H.-H., Chiu, H.-C., Delaney, N. F., Segrè, D., and Marx, C. J. (2011). Diminishing returns epistasis among beneficial mutations decelerates adaptation. *Science (New York, N.Y.)*, 332(6034):1190–1192.

46. Churchland, P. S. (1986). Neurophilosophy: Toward A Unified Science of the Mind-Brain. MIT Press.

47. Coates, A., Ng, A., and Lee, H. (2011). An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 215–223. JMLR Workshop and Conference Proceedings.

48. 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.

49. Cocco, S. and Monasson, R. (2011). Adaptive Cluster Expansion for Inferring Boltzmann Machines with Noisy Data. *Physical Review Letters*, 106(9):090601.

50. 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.

51. Conchuir, S. O., Barlow, K. A., Pache, R. A., Ollikainen, N., Kundert, K., O'Meara, M. J., Smith, C. A., and Kortemme, T. (2015). A Web Resource for Standardized Benchmark Datasets, Metrics, and Rosetta Protocols for Macromolecular Modeling and Design. *PLOS ONE*, 10(9):e0130433.

52. Courville, A., Bergstra, J., and Bengio, Y. (2011). Unsupervised models of images by spike-and-slab RBMs. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 1145–1152.

53. Dahl, G. E., Ranzato, M., Mohamed, A., and Hinton, G. (2010). Phone recognition with the mean-covariance restricted Boltzmann Machine. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 469–477.

54. Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42.

55. Danev, R., Yanagisawa, H., and Kikkawa, M. (2019). Cryo-Electron Microscopy Methodology: Current Aspects and Future Directions. *Trends in Biochemical Sciences*, 44(10):837–848.

56. de Visser, J. A. G. M. and Elena, S. F. (2007). The evolution of sex: Empirical insights into the roles of epistasis and drift. *Nature Reviews. Genetics*, 8(2):139–149.

57. de Visser, J. A. G. M. and Krug, J. (2014). Empirical fitness landscapes and the predictability of evolution. *Nature Reviews. Genetics*, 15(7):480–490.

58. Decelle, A., Fissore, G., and Furtlehner, C. (2017). Spectral dynamics of learning in restricted Boltzmann machines. *EPL (Europhysics Letters)*, 119(6):60001.

59. Decelle, A., Fissore, G., and Furtlehner, C. (2018). Thermodynamics of Restricted Boltzmann Machines and Related Learning Dynamics. *Journal of Statistical Physics*, 172(6):1576–1608.

60. Decelle, A. and Furtlehner, C. (2020a). Gaussian-spherical restricted Boltzmann machines. *Journal of Physics A: Mathematical and Theoretical*, 53(18):184002.

61. Decelle, A. and Furtlehner, C. (2020b). Restricted Boltzmann Machine, recent advances and mean-field theory. *Chinese Physics B*.

62. Decelle, A., Furtlehner, C., and Seoane, B. (2021). Equilibrium and non-Equilibrium regimes in the learning of Restricted Boltzmann Machines. *arXiv:2105.13889 [cond-mat]*.

63. Decelle, A., Hwang, S., Rocchi, J., and Tantari, D. (2019). Inverse problems for structured datasets using parallel TAP equations and RBM. *arXiv:1906.11988 [cond-mat]*.

64. Dennis, J. E. and Moré, J. J. (1977). Quasi-Newton Methods, Motivation and Theory. *SIAM Review*, 19(1):46–89.

65. DePristo, M. A., Weinreich, D. M., and Hartl, D. L. (2005). Missense meanderings in sequence space: A biophysical view of protein evolution. *Nature Reviews Genetics*, 6(9):678–687.

66. Desjardins, G. and Bengio, Y. (2008). Empirical Evaluation of Convolutional RBMs for Vision. *Technical Report 1327, Département d'Informatique et de Recherche Opérationnelle, Université de Montréal*, page 13.

67. Desjardins, G., Courville, A., Bengio, Y., Vincent, P., and Delalleau, O. (2010a). Parallel tempering for training of restricted Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 145–152. MIT Press Cambridge, MA.

68. Desjardins, G., Courville, A., Bengio, Y., Vincent, P., and Delalleau, O. (2010b). Tempered Markov Chain Monte Carlo for training of Restricted Boltzmann Machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 145–152. JMLR Workshop and Conference Proceedings.

69. Desjardins, G., Luo, H., Courville, A., and Bengio, Y. (2014). Deep Tempering. *arXiv:1410.0123 [cs, stat]*.

70. Deudon, M. (2020). On food, bias and seasons: A recipe for sustainability. *HAL Archives Ouvertes*.

71. Dill, K. A., Ozkan, S. B., Shell, M. S., and Weikl, T. R. (2008). The Protein Folding Problem. *Annual review of biophysics*, 37:289–316.

72. Dixit, V., Selvarajan, R., Alam, M. A., Humble, T. S., and Kais, S. (2021). Training Restricted Boltzmann Machines With a D-Wave Quantum Annealer. *Frontiers in Physics*, 9:374.

73. Doucet, N., Savard, P.-Y., Pelletier, J. N., and Gagné, S. M. (2007). NMR investigation of Tyr105 mutants in TEM-1 beta-lactamase: Dynamics are correlated with function. *The Journal of Biological Chemistry*, 282(29):21448–21459.

74. 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.

75. Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambrigde University Press. Cambrigde University Press.

76. Durrant, J. D. and McCammon, J. A. (2011). Molecular dynamics simulations and drug discovery. *BMC Biology*, 9:71.

77. Eccles, J. C. (1964). The Physiology of Synapses. Springer-Verlag, Berlin Heidelberg.

78. Eddy, S. R. (1998). Profile hidden Markov models. *Bioinformatics*, 14(9):755–763.

79. Egorov, A., Rubtsova, M., Grigorenko, V., Uporov, I., and Veselovsky, A. (2019). The Role of the Ω-Loop in Regulation of the Catalytic Activity of TEM-Type β-Lactamases. *Biomolecules*, 9(12):854.

80. Ekeberg, M., Hartonen, T., and Aurell, E. (2014). Fast pseudolikelihood maximization for direct-coupling analysis of protein structure from many homologous amino-acid sequences. *Journal of Computational Physics*, 276:341–356.

81. Ekeberg, M., Lövkvist, C., Lan, Y., Weigt, M., and Aurell, E. (2013). Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, 87(1):012707.

82. Ellis, R. (1985). Entropy, Large Deviations, and Statistical Mechanics. Classics in Mathematics. Springer-Verlag.

83. Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11(19):625–660.

84. Evans, R., O'Neill, M., Pritzel, A., Antropova, N., Senior, A. W., Green, T., Žídek, A., Bates, R., Blackwell, S., Yim, J., Ronneberger, O., Bodenstein, S., Zielinski, M., Bridgland, A., Potapenko, A., Cowie, A., Tunyasuvunakool, K., Jain, R., Clancy, E., Kohli, P., Jumper, J., and Hassabis, D. (2021). Protein complex prediction with AlphaFold-Multimer.

85. 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. *Molecular Biology and Evolution*, 33(1):268–280.

86. Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., Sonnhammer, E. L. L., Tate, J., and Punta, M. (2014). Pfam: The protein families database. *Nucleic Acids Research*, 42:D222–D230.

87. Finn, R. D., Clements, J., and Eddy, S. R. (2011). HMMER web server: Interactive sequence similarity searching. *Nucleic Acids Research*, 39(Web Server issue):W29–37.

88. Fischer, A. and Igel, C. (2010). Empirical Analysis of the Divergence of Gibbs Sampling Based Learning Algorithms for Restricted Boltzmann Machines. In *Artificial Neural Networks – ICANN 2010*, Lecture Notes in Computer Science, pages 208–217. Springer.

89. Fischer, A. and Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39.

90. Fisher, C. K., Smith, A. M., and Walsh, J. R. (2018). Boltzmann Encoded Adversarial Machines. *arXiv:1804.08682 [cs, stat]*.

91. Fowler, D. M. and Fields, S. (2014). Deep mutational scanning: A new style of protein science. *Nature Methods*, 11(8):801–807.

92. Gabrié, M., Tramel, E. W., and Krzakala, F. (2015). Training Restricted Boltzmann Machines via the Thouless-Anderson-Palmer Free Energy. *arXiv:1506.02914 [cond-mat, stat]*.

93. Gandhi, J., Antonelli, A. C., Afridi, A., Vatsia, S., Joshi, G., Romanov, V., Murray, I. V. J., and Khan, S. A. (2019). Protein misfolding and aggregation in neurodegenerative diseases: A review of pathogeneses, novel detection strategies, and potential therapeutics. *Reviews in the Neurosciences*, 30(4):339–358.

94. Gardner, E. (1987). Maximum Storage Capacity in Neural Networks. *Europhysics Letters (EPL)*, 4(4):481–485.

95. Gardner, E. (1988). The space of interactions in neural network models. *Journal of Physics A: Mathematical and General*, 21(1):257–270.

96. Gardner, E. and Derrida, B. (1988). Optimal storage properties of neural network models. *Journal of Physics A: Mathematical and General*, 21(1):271–284.

97. Georges, A. and Yedidia, J. S. (1991). How to expand around mean-field theory using high-temperature expansions. *Journal of Physics A: Mathematical and General*, 24(9):2173–2192.

98. Geyer, C. J. (1991). Markov Chain Monte Carlo Maximum Likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*. Interface Foundation of North America.

99. Glauber, R. J. (1963). Time Dependent Statistics of the Ising Model. *Journal of Mathematical Physics*, 4(2):294.

100. Göbel, U., Sander, C., Schneider, R., and Valencia, A. (1994). Correlated mutations and residue contacts in proteins. *Proteins*, 18(4):309–317.

101. Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, MA, USA.

102. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*.

103. Gros, P.-A., Le Nagard, H., and Tenaillon, O. (2009). The evolution of epistasis and its links with genetic robustness, complexity and drift in a phenotypic model of adaptation. *Genetics*, 182(1):277–293.

104. Gross, C. G. (2002). Genealogy of the “Grandmother Cell”. *The Neuroscientist*, 8(5):512–518.

105. Haber, E. and Anfinsen, C. B. (1962). Side-chain interactions governing the pairing of half-cystine residues in ribonuclease. *The Journal of Biological Chemistry*, 237:1839–1844.

106. Hagen, J. B. (2000). The origins of bioinformatics. *Nature Reviews. Genetics*, 1(3):231–236.

107. Halabi, N., Rivoire, O., Leibler, S., and Ranganathan, R. (2009). Protein sectors: Evolutionary units of three-dimensional structure. *Cell*, 138(4):774–786.

108. Haldane, A., Flynn, W. F., He, P., Vijayan, R. S. K., and Levy, R. M. (2016). Structural propensities of kinase family proteins from a Potts model of residue co-variation. *Protein Science: A Publication of the Protein Society*, 25(8):1378–1384.

109. Hartnett, G. S., Parker, E., and Geist, E. (2018). Replica symmetry breaking in bipartite spin glasses and neural networks. *Physical Review E*, 98(2):022116.

110. Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, page 13.

111. 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):e1008736.

112. He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*.

113. Hebb, D. O. (1949). The organization of behavior: A neuropsychological theory. Wiley, New York.

114. Hinton, G., Dayan, P., Frey, B., and Neal, R. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science*.

115. Hinton, G. E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, page 30.

116. Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554.

117. Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.

118. Hinton, G. E. and Sejnowski, T. J. (1983). Optimal Perceptual Inference. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 448:6.

119. Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

120. Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

121. Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., and Marks, D. S. (2017). Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135.

122. 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.

123. Huang, L. and Wang, L. (2017). Accelerate Monte Carlo Simulations with Restricted Boltzmann Machines. *Physical Review B*, 95(3):035105.

124. Huang, P.-S., Boyken, S. E., and Baker, D. (2016). The coming of age of de novo protein design. *Nature*, 537(7620):320–327.

125. Huang, Y., Yang, C., Xu, X.-f., Xu, W., and Liu, S.-w. (2020). Structural and functional properties of SARS-CoV-2 spike protein: Potential antivirus drug development for COVID-19. *Acta Pharmacologica Sinica*, 41(9):1141–1149.

126. Ising, E. (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258.

127. Jackson, L. A., Anderson, E. J., Rouphael, N. G., Roberts, P. C., Makhene, M., Coler, R. N., McCullough, M. P., Chappell, J. D., Denison, M. R., Stevens, L. J., Pruijssers, A. J., McDermott, A., Flach, B., Doria-Rose, N. A., Corbett, K. S., Morabito, K. M., O'Dell, S., Schmidt, S. D., Swanson, P. A., Padilla, M., Mascola, J. R., Neuzil, K. M., Bennett, H., Sun, W., Peters, E., Makowski, M., Albert, J., Cross, K., Buchanan, W., Pikaart-Tautges, R., Ledgerwood, J. E., Graham, B. S., Beigel, J. H., and mRNA-1273 Study Group (2020). An mRNA Vaccine against SARS-CoV-2 - Preliminary Report. *The New England Journal of Medicine*, 383(20):1920–1931.

128. Jacquier, H., Birgy, A., Le Nagard, H., Mechulam, Y., Schmitt, E., Glodt, J., Bercot, B., Petit, E., Poulain, J., Barnaud, G., Gros, P.-A., and Tenaillon, O. (2013). Capturing the mutational landscape of the beta-lactamase TEM-1. *Proceedings of the National Academy of Sciences of the United States of America*, 110(32):13067–13072.

129. 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):e1004889.

130. Jarzynski, C. (1997). Nonequilibrium Equality for Free Energy Differences. *Physical Review Letters*, 78(14):2690–2693.

131. Jaynes, E. T. (1957a). Information Theory and Statistical Mechanics. *Physical Review*, 106(4):620–630.

132. Jaynes, E. T. (1957b). Information Theory and Statistical Mechanics. II. *Physical Review*, 108(2):171–190.

133. Jelsch, C., Mourey, L., Masson, J. M., and Samama, J. P. (1993). Crystal structure of Escherichia coli TEM1 beta-lactamase at 1.8 A resolution. *Proteins*, 16(4):364–383.

134. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R.,

Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*.

135. Kanter, I. and Sompolinsky, H. (1987). Associative recall of memory without errors. *Physical Review A*, 35(1):380–392.

136. Kantorovich, L. (1942). On the transfer of masses (in russian). *Doklady Akademii*.

137. Karikó, K., Buckstein, M., Ni, H., and Weissman, D. (2005). Suppression of RNA Recognition by Toll-like Receptors: The Impact of Nucleoside Modification and the Evolutionary Origin of RNA. *Immunity*, 23(2):165–175.

138. Karplus, M. and Kuriyan, J. (2005). Molecular dynamics and protein function. *Proceedings of the National Academy of Sciences of the United States of America*, 102(19):6679–6685.

139. 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.

140. Kauffman, S. A. and Weinberger, E. D. (1989). The NK model of rugged fitness landscapes and its application to maturation of the immune response. *Journal of Theoretical Biology*, 141(2):211–245.

141. Keener, A. B. (2018). Just the messenger. *Nature Medicine*, 24(9):1297–1300.

142. Khan, A. I., Dinh, D. M., Schneider, D., Lenski, R. E., and Cooper, T. F. (2011). Negative epistasis between beneficial mutations in an evolving bacterial population. *Science (New York, N.Y.)*, 332(6034):1193–1196.

143. Kingma, D. P. and Welling, M. (2014). Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*.

144. Kirkpatrick, S. and Sherrington, D. (1978). Infinite-ranged models of spin-glasses. *Physical Review B*, 17(11):4384–4403.

145. Kivinen, J. and Williams, C. (2012). Multiple Texture Boltzmann Machines. In *Artificial Intelligence and Statistics*, pages 638–646. PMLR.

146. Koch-Janusz, M. and Ringel, Z. (2018). Mutual information, neural networks and the renormalization group. *Nature Physics*, 14(6):578–582.

147. Kong, K.-F., Schneper, L., and Mathee, K. (2010). Beta-lactam Antibiotics: From Antibiosis to Resistance and Bacteriology. *APMIS : acta pathologica, microbiologica, et immunologica Scandinavica*, 118(1):1–36.

148. Kosko, B. (1987). Adaptive bidirectional associative memories. *Applied Optics*, 26(23):4947–4960.

149. Kosko, B. (1988). Bidirectional Associative Memories. *Ieee Transactions on Systems, Man, and Cybernetics*, 18(1):49–60.

150. Krauth, W. (2006). Statistical Mechanics: Algorithms and Computations. Oxford Master Series in Physics.

151. Kryazhimskiy, S., Rice, D. P., Jerison, E. R., and Desai, M. M. (2014). Microbial evolution. Global epistasis makes adaptation predictable despite sequence-level stochasticity. *Science (New York, N.Y.)*, 344(6191):1519–1522.

152. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., and Moult, J. (2019). Critical assessment of methods of protein structure prediction (CASP)-Round XIII. *Proteins*, 87(12):1011–1020.

153. Kubelka, J., Hofrichter, J., and Eaton, W. A. (2004). The protein folding 'speed limit'. *Current Opinion in Structural Biology*, 14(1):76–88.

154. Kuhlman, B. and Bradley, P. (2019). Advances in protein structure prediction and design. *Nature Reviews Molecular Cell Biology*, 20(11):681–697.

155. Larochelle, H. and Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, pages 536–543, Helsinki, Finland. ACM Press.

156. Le Roux, N. and Bengio, Y. (2008). Representational Power of Restricted Boltzmann Machines and Deep Belief Networks. *Neural Computation*, 20(6):1631–1649.

157. LeCun, Y. (1998). The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

158. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323.

159. Lenggenhager, P. M., Gökmen, D. E., Ringel, Z., Huber, S. D., and Koch-Janusz, M. (2020). Optimal Renormalization Group Transformation from Information Theory. *Physical Review X*, 10(1):011037.

160. Leonelli, F. E., Agliari, E., Albanese, L., and Barra, A. (2021). On the effective initialisation for restricted Boltzmann machines via duality with Hopfield model. *Neural Networks*, 143:314–326.

161. Levinthal, C. (1969). How to fold graciously. *Mossbauer spectroscopy in biological systems*, 67:22–24.

162. Lindorff-Larsen, K., Piana, S., Dror, R. O., and Shaw, D. E. (2011). How Fast-Folding Proteins Fold. *Science*, 334(6055):517–520.

163. Little, W. A. (1974). The existence of persistent states in the brain. *Mathematical Biosciences*, 19(1):101–120.

164. Little, W. A. and Shaw, G. L. (1975). A statistical theory of short and long term memory. *Behavioral Biology*, 14(2):115–133.

165. Liu, J., Qi, Y., Meng, Z. Y., and Fu, L. (2017). Self-learning Monte Carlo method. *Physical Review B*, 95(4):041101.

166. Lockless, S. W. and Ranganathan, R. (1999). Evolutionarily Conserved Pathways of Energetic Connectivity in Protein Families. *Science*, 286(5438):295–299.

167. Long, Y., Nachmias, A., Ning, W., and Peres, Y. (2014). A power law of order 1/4 for critical mean field Swendsen-Wang dynamics. *Memoirs of the American Mathematical Society*, 232.

168. MacKay, D. J. C. (2003). Information Theory, Inference and Learning Algorithms. Cambridge University Press.

169. Marmier, G., Weigt, M., and Bitbol, A.-F. (2019). Phylogenetic correlations can suffice to infer protein partners from sequences. *PLOS Computational Biology*, 15(10):e1007179.

170. Martin, G., Elena, S. F., and Lenormand, T. (2007). Distributions of epistasis in microbes fit predictions from a fitness landscape model. *Nature Genetics*, 39(4):555–560.

171. Matagne, A., Lamotte-Brasseur, J., and Frère, J.-M. (1998). Catalytic properties of class A β-lactamases: Efficiency and diversity. *Biochemical Journal*, 330(2):581–598.

172. Mattis, D. C. (1976). Solvable spin systems with random interactions. *Physics Letters A*, 56(5):421–422.

173. Maveyraud, L. and Mourey, L. (2020). Protein X-ray Crystallography and Drug Discovery. *Molecules*, 25(5).

174. McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

175. McLinn, S. and Williams, D. (1996). Incidence of antibiotic-resistant Streptococcus pneumoniae and beta-lactamase-positive Haemophilus influenzae in clinical isolates from patients with otitis media. *The Pediatric Infectious Disease Journal*, 15(9 Suppl):S3–9.

176. McPherson, A. and Gavira, J. A. (2013). Introduction to protein crystallization. *Acta Crystallographica. Section F, Structural Biology Communications*, 70(Pt 1):2–20.

177. Mehta, P. and Schwab, D. J. (2014). An exact mapping between the Variational Renormalization Group and Deep Learning. *arXiv:1410.3831 [cond-mat, stat]*.

178. Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*.

179. Metropolis, N. and Ulam, S. (1949). The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341.

180. Mézard, M. (2017). Mean-field message-passing equations in the Hopfield model and its generalizations. *Physical Review E*, 95(2):022117.

181. Mézard, M., Parisi, G., Sourlas, N., Toulouse, G., and Virasoro, M. (1984). Nature of the Spin-Glass Phase. *Physical Review Letters*, 52(13):1156–1159.

182. Milo, R. (2013). What is the total number of protein molecules per cell volume? A call to rethink some published values. *BioEssays: News and Reviews in Molecular, Cellular and Developmental Biology*, 35(12):1050–1055.

183. Mirny, L. and Shakhnovich, E. (2001). Protein Folding Theory: From Lattice to All-Atom Models. *Annual Review of Biophysics and Biomolecular Structure*, 30(1):361–396.

184. 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. *Journal of Molecular Biology*, 256(3):623–644.

185. Mohamed, A., Dahl, G. E., and Hinton, G. (2012). Acoustic Modeling Using Deep Belief Networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22.

186. Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. De l'Imprimerie Royale.

187. Montavon, G., Müller, K.-R., and Cuturi, M. (2016). Wasserstein Training of Restricted Boltzmann Machines. *Advances in Neural Information Processing Systems*, page 9.

188. Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301.

189. Moult, J., Pedersen, J. T., Judson, R., and Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods. *Proteins*, 23(3):ii–v.

190. Murata, K. and Wolf, M. (2018). Cryo-electron microscopy for structural analysis of dynamic biological macromolecules. *Biochimica et Biophysica Acta (BBA) - General Subjects*, 1862(2):324–334.

191. Nagai, Y., Okumura, M., Kobayashi, K., and Shiga, M. (2020a). Self-learning hybrid Monte Carlo: A first-principles approach. *Physical Review B*, 102(4):041124.

192. Nagai, Y., Okumura, M., and Tanaka, A. (2020b). Self-learning Monte Carlo method with Behler-Parrinello neural networks. *Physical Review B*, 101(11):115111.

193. Nagai, Y., Shen, H., Qi, Y., Liu, J., and Fu, L. (2017). Self-learning Monte Carlo method: Continuous-time algorithm. *Physical Review B*, 96(16):161102.

194. Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, Madison, WI, USA.

195. Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

196. Nesterov, Y. (2004). Introductory Lectures on Convex Optimization: A Basic Course. Applied Optimization. Springer US.

197. Neu, H. C. (1969). Effect of beta-Lactamase Location in Escherichia coli on Penicillin Synergy. *Applied Microbiology*, 17(6):783–786.

198. Ng, P. C. and Henikoff, S. (2003). SIFT: Predicting amino acid changes that affect protein function. *Nucleic Acids Research*, 31(13):3812–3814.

199. Nguyen, H. C., Zecchina, R., and Berg, J. (2017). Inverse statistical problems: From the inverse Ising problem to data science. *Advances in Physics*, 66(3):197–261.

200. Onsager, L. (1944). Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition. *Physical Review*, 65(3-4):117–149.

201. Orts, J. and Gossert, A. D. (2018). Structure determination of protein-ligand complexes by NMR in solution. *Methods*, 138-139:3–25.

202. Otwinowski, J., McCandlish, D. M., and Plotkin, J. B. (2018). Inferring the shape of global epistasis. *Proceedings of the National Academy of Sciences of the United States of America*, 115(32):E7550–E7558.

203. Ovchinnikov, S., Kim, D. E., Wang, R. Y.-R., Liu, Y., DiMaio, F., and Baker, D. (2016). Improved de novo structure prediction in CASP11 by incorporating coevolution information into Rosetta. *Proteins*, 84 Suppl 1:67–75.

204. Ovchinnikov, S., Park, H., Varghese, N., Huang, P.-S., Pavlopoulos, G. A., Kim, D. E., Kamisetty, H., Kyrpides, N. C., and Baker, D. (2017). Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298.

205. Pardi, N., Hogan, M. J., Porter, F. W., and Weissman, D. (2018). mRNA vaccines — a new era in vaccinology. *Nature Reviews Drug Discovery*, 17(4):261–279.

206. Parisi, G. (1986). Asymmetric neural networks and the process of learning. *Journal of Physics A: Mathematical and General*, pages 444–449.

207. 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 of the United States of America*, 37(5):235–240.

208. Pauling, L., Corey, R. B., and Branson, H. R. (1951). The structure of proteins; two hydrogen-bonded helical configurations of the polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 37(4):205–211.

209. Pearl, J. (1982). Reverend bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the Second AAAI Conference on Artificial Intelligence*, AAAI'82, pages 133–136, Pittsburgh, Pennsylvania. AAAI Press.

210. Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

211. Peretto, P. (1984). Collective properties of neural networks: A statistical physics approach. *Biological Cybernetics*, 50(1):51–62.

212. Persky, N., Ben-Av, R., Kanter, I., and Domany, E. (1996). Mean-field behavior of cluster dynamics. *Physical Review E*, 54(3):2351–2358.

213. Personnaz, L., Guyon, I., and Dreyfus, G. (1986). Collective computational properties of neural networks: New learning mechanisms. *Physical Review A*, 34(5):4217–4228.

214. Philippon, A., Jacquier, H., Ruppé, E., and Labia, R. (2019). Structure-based classification of class A beta-lactamases, an update. *Current Research in Translational Medicine*, 67(4):115–122.

215. Philippon, A., Slama, P., Dény, P., and Labia, R. (2016). A Structure-Based Classification of Class A beta-Lactamases, a Broadly Diverse Family of Enzymes. *Clinical Microbiology Reviews*, 29(1):29–57.

216. Poelwijk, F. J., Socolich, M., and Ranganathan, R. (2019). Learning the pattern of epistasis linking genotype and phenotype in a protein. *Nature Communications*, 10(1):4213.

217. Polack, F. P., Thomas, S. J., Kitchin, N., Absalon, J., Gurtman, A., Lockhart, S., Perez, J. L., Pérez Marc, G., Moreira, E. D., Zerbini, C., Bailey, R., Swanson, K. A., Roychoudhury, S., Koury, K., Li, P., Kalina, W. V., Cooper, D., Frenck, R. W., Hammitt, L. L., Türeci, O., Nell, H., Schaefer, A., Unal, S., Tresnan, D. B., Mather, S., Dormitzer, P. R., Sahin, U., Jansen, K. U., Gruber, W. C., and C4591001 Clinical Trial Group (2020). Safety and Efficacy of the BNT162b2 mRNA Covid-19 Vaccine. *The New England Journal of Medicine*, 383(27):2603–2615.

218. Privalov, P. L. (1979). Stability of proteins: Small globular proteins. *Advances in Protein Chemistry*, 33:167–241.

219. Procesi, C. and Tirozzi, B. (1990). Metastable states in the hopfield model. *International Journal of Modern Physics B*, 04(01):143–150.

220. Quinn, C. M., Wang, M., and Polenova, T. (2018). NMR of Macromolecular Assemblies and Machines at 1 GHz and Beyond: New Transformative Opportunities for Molecular Structural Biology. *Methods in Molecular Biology (Clifton, N.J.)*, 1688:1–35.

221. Rammal, R., Toulouse, G., and Virasoro, M. A. (1986). Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765–788.

222. Ranzato, M., Susskind, J., Mnih, V., and Hinton, G. (2011). On deep generative models with applications to recognition. In *CVPR 2011*, pages 2857–2864.

223. Rausell, A., Juan, D., Pazos, F., and Valencia, A. (2010). Protein interactions and ligand binding: From protein subfamilies to functional specificity. *Proceedings of the National Academy of Sciences*, 107(5):1995–2000.

224. Ravikumar, P., Wainwright, M. J., and Lafferty, J. D. (2010). High-dimensional Ising model selection using L1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319.

225. Ray, T. S., Tamayo, P., and Klein, W. (1989). Mean-field study of the Swendsen-Wang dynamics. *Physical Review A*, 39(11):5949–5953.

226. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., Savolainen, O., Meskys, R., Engqvist, M. K. M., and Zelezniak, A. (2021). Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333.

227. Riesselman, A. J., Ingraham, J. B., and Marks, D. S. (2017). Deep generative models of genetic variation capture mutation effects. *arXiv:1712.06527 [cond-mat, physics:physics, q-bio, stat]*.

228. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15).

229. Rivoire, O., Reynolds, K. A., and Ranganathan, R. (2016). Evolution-Based Functional Decomposition of Proteins. *PLOS Computational Biology*, 12(6):e1004817.

230. Rizzato, F., Coucke, A., de Leonardis, E., Barton, J. P., Tubiana, J., Monasson, R., and Cocco, S. (2020). Inference of compressed Potts graphical models. *Physical Review E*, 101(1):012309.

231. Rodrigues, J. V., Bershtein, S., Li, A., Lozovsky, E. R., Hartl, D. L., and Shakhnovich, E. I. (2016). Biophysical principles predict fitness landscapes of drug resistance. *Proceedings of the National Academy of Sciences of the United States of America*, 113(11):E1470–1478.

232. Rollins, N. J., Brock, K. P., Poelwijk, F. J., Stiffler, M. A., Gauthier, N. P., Sander, C., and Marks, D. S. (2019). Inferring protein 3D structure from deep mutation scans. *Nature Genetics*, 51(7):1170–1176.

233. Romero, P. A. and Arnold, F. H. (2009). Exploring protein fitness landscapes by directed evolution. *Nature reviews. Molecular cell biology*, 10(12):866–876.

234. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

235. Roudi, Y., Tyrcha, J., and Hertz, J. (2009). Ising model for neural data: Model quality and approximate methods for extracting functional connectivity. *Physical Review E*, 79(5):051915.

236. Roussel, C., Cocco, S., and Monasson, R. (2021). Barriers and dynamical paths in alternating Gibbs sampling of restricted Boltzmann machines. *Physical Review E*, 104(3):034109.

237. Ruder, S. (2017). An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*.

238. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

239. 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(6502):440–445.

240. 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.

241. Salakhutdinov, R. and Hinton, G. (2009). Deep Boltzmann Machines. In *Artificial Intelligence and Statistics*, pages 448–455. PMLR.

242. Salakhutdinov, R. and Larochelle, H. (2010). Efficient Learning of Deep Boltzmann Machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 693–700. JMLR Workshop and Conference Proceedings.

243. Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 791–798, New York, NY, USA. Association for Computing Machinery.

244. Salakhutdinov, R. and Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 872–879.

245. Salakhutdinov, R. R. (2009). Learning in Markov Random Fields using Tempered Transitions. *Advances in Neural Information Processing Systems*, 22.

246. Sali, A., Shakhnovich, E., and Karplus, M. (1994a). How does a protein fold? *Nature*, 369(6477):248–251.

247. Sali, A., Shakhnovich, E., and Karplus, M. (1994b). Kinetics of protein folding: A lattice model study of the requirements for folding to the native state. *Journal of Molecular Biology*, 235(5):1614–1636.

248. Salinas, V. H. and Ranganathan, R. (2018). Coevolution-based inference of amino acid interactions underlying protein function. *eLife*, 7:e34300.

249. Salverda, M. L. M., De Visser, J. A. G. M., and Barlow, M. (2010). Natural evolution of TEM-1 β-lactamase: Experimental reconstruction and clinical relevance. *FEMS microbiology reviews*, 34(6):1015–1036.

250. Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., Bogatyreva, N. S., Vlasov, P. K., Egorov, E. S., Logacheva, M. D., Kondrashov, A. S., Chudakov, D. M., Putintseva, E. V., Mamedov, I. Z., Tawfik, D. S., Lukyanov, K. A., and Kondrashov, F. A. (2016). Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401.

251. Schmiedel, J. M. and Lehner, B. (2019). Determining protein structures using deep mutagenesis. *Nature Genetics*, 51(7):1177–1186.

252. Science (2005). So Much More to Know. *Science*, 309(5731):78–102.

253. Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Zidek, 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.

254. Shakhnovich, E., Farztdinov, G., Gutin, A. M., and Karplus, M. (1991). Protein folding bottlenecks: A lattice Monte Carlo simulation. *Physical Review Letters*, 67(12):1665–1668.

255. 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.

256. Shakhnovich, E. I. (1997). Theoretical studies of protein-folding thermodynamics and kinetics. *Current Opinion in Structural Biology*, 7(1):29–40.

257. Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

258. Shen, H., Liu, J., and Fu, L. (2018). Self-learning Monte Carlo with deep neural networks. *Physical Review B*, 97(20):205140.

259. Sherrington, D. and Kirkpatrick, S. (1975). Solvable Model of a Spin-Glass. *Physical Review Letters*, 35(26):1792–1796.

260. Shi, Y. (2014). A glimpse of structural biology through X-ray crystallography. *Cell*, 159(5):995–1014.

261. Shimagaki, K. and Weigt, M. (2019). Collective-variable selection and generative Hopfield-Potts models for protein-sequence families. *arXiv:1905.11848 [cond-mat, q-bio]*.

262. Shinomoto, S. (1987). A cognitive and associative memory. *Biological Cybernetics*, 57(3):197–206.

263. Sinai, S., Kelsic, E., Church, G. M., and Nowak, M. A. (2018). Variational auto-encoding of protein sequences. *arXiv:1712.03346 [cs, q-bio]*.

264. Smolensky, P. (1986). Chapter 6: Information Processing in Dynamical Systems: Foundations of Harmony Theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations.*, pages 194–281. MIT Press Cambridge, MA.

265. Sompolinsky, H. (1986). Neural networks with nonlinear synapses and a static noise. *Physical Review A*, 34(3):2571–2574.

266. Stiffler, M. A., Hekstra, D. R., and Ranganathan, R. (2015). Evolvability as a Function of Purifying Selection in TEM-1 β-Lactamase. *Cell*, 160(5):882–892.

267. Sudol, M., Chen, H. I., Bougeret, C., Einbond, A., and Bork, P. (1995). Characterization of a novel protein-binding module — the WW domain. *FEBS Letters*, 369(1):67–71.

268. Sutto, L., Marsili, S., Valencia, A., and Gervasio, F. L. (2015). From residue coevolution to protein conformational ensembles and functional dynamics. *Proceedings of the National Academy of Sciences*, 112(44):13567–13572.

269. Swendsen, R. and Wang, J.-S. (1986). Replica Monte Carlo Simulation of Spin-Glasses. *Physical Review Letters*, 57:2607–2609.

270. Swendsen, R. H. and Wang, J.-S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Physical Review Letters*, 58(2):86–88.

271. Talibart, H. and Coste, F. (2021). PPalign: Optimal alignment of Potts models representing proteins with direct coupling information. *BMC Bioinformatics*, 22(1):317.

272. Tanaka, T., Kakiya, S., and Kabashima, Y. (2000). Capacity analysis of bidirectional associative memory. *Proc. Seventh Int. Conf. Neural Information Processing, Taejon, Korea*, 2:779–784.

273. Tang, Y., Salakhutdinov, R., and Hinton, G. (2012). Robust Boltzmann Machines for recognition and denoising. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2264–2271.

274. Teller, E. and Ashkin, J. (1943). Statistics of Two-Dimensional Lattices with Four Components. *Physical Review*, 64(5-6):178–184.

275. Tenaillon, O. (2014). The Utility of Fisher's Geometric Model in Evolutionary Genetics. *Annual Review of Ecology, Evolution, and Systematics*, 45(1):179–201.

276. The UniProt Consortium (2021). UniProt: The universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489.

277. Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, pages 1064–1071.

278. Tieleman, T. and Hinton, G. E. (2009). Using fast weights to improve persistent contrastive divergence. In *ICML*.

279. Tillier, E. R. and Lui, T. W. (2003). Using multiple interdependency to separate functional from phylogenetic correlations in protein alignments. *Bioinformatics*, 19(6):750–755.

280. Torrisi, M., Pollastri, G., and Le, Q. (2020). Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal*, 18:1301–1310.

281. Townshend, R. J. L., Eismann, S., Watkins, A. M., Rangan, R., Karelina, M., Das, R., and Dror, R. O. (2021). Geometric deep learning of RNA structure. *Science*, 373(6558):1047–1051.

282. Tramel, E. W., Gabrié, M., Manoel, A., Caltagirone, F., and Krzakala, F. (2018). A Deterministic and Generalized Framework for Unsupervised Learning with Restricted Boltzmann Machines. *Physical Review X*, 8(4):041006.

283. Tubiana, J. (2018). Restricted Boltzmann machines : From compositional representations to protein sequence analysis. These de doctorat, Paris Sciences et Lettres (ComUE).

284. Tubiana, J., Cocco, S., and Monasson, R. (2019a). Learning Compositional Representations of Interacting Systems with Restricted Boltzmann Machines: Comparative Study of Lattice Proteins. *Neural Computation*, 31(8):1671–1717.

285. Tubiana, J., Cocco, S., and Monasson, R. (2019b). Learning protein constitutive motifs from sequence data. *eLife*, 8:e39397.

286. Tubiana, J. and Monasson, R. (2017). Emergence of Compositional Representations in Restricted Boltzmann Machines. *Physical Review Letters*, 118(13):138301.

287. Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Zidek, A., Bridgland, A., Cowie, A., Meyer, C., Laydon, A., Velankar, S., Kleywegt, G. J., Bateman, A., Evans, R., Pritzel, A., Figurnov, M., Ronneberger, O., Bates, R., Kohl, S. A. A., Potapenko, A., Ballard, A. J., Romera-Paredes, B., Nikolov, S., Jain, R., Clancy, E., Reiman, D., Petersen, S., Senior, A. W., Kavukcuoglu, K., Birney, E., Kohli, P., Jumper, J., and Hassabis, D. (2021). Highly accurate protein structure prediction for the human proteome. *Nature*, pages 1–9.

288. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6000–6010.

289. von Heijne, G. (1991). Proline kinks in transmembrane alpha-helices. *Journal of Molecular Biology*, 218(3):499–503.

290. Wang, F., Cassidy, C., and Sacchettini, J. C. (2006). Crystal Structure and Activity Studies of the Mycobacterium tuberculosis β-Lactamase Reveal Its Critical Role in Resistance to β-Lactam Antibiotics. *Antimicrobial Agents and Chemotherapy*, 50(8):2762–2771.

291. Wang, H.-W. and Wang, J.-W. (2017). How cryo-electron microscopy and X-ray crystallography complement each other. *Protein Science: A Publication of the Protein Society*, 26(1):32–39.

292. Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. (2009). Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72.

293. Weinreich, D. M., Lan, Y., Jaffe, J., and Heckendorn, R. B. (2018). The Influence of Higher-Order Epistasis on Biological Fitness Landscape Topography. *Journal of Statistical Physics*, 172(1):208–225.

294. Weinreich, D. M., Lan, Y., Wylie, C. S., and Heckendorn, R. B. (2013). Should evolutionary geneticists worry about higher-order epistasis? *Current Opinion in Genetics & Development*, 23(6):700–707.

295. Welling, M. and Hinton, G. E. (2002). A New Learning Algorithm for Mean Field Boltzmann Machines. In Dorronsoro, J. R., editor, *Artificial Neural Networks — ICANN 2002*, Lecture Notes in Computer Science, pages 351–357, Berlin, Heidelberg. Springer.

296. Wiser, M. J., Ribeck, N., and Lenski, R. E. (2013). Long-term dynamics of adaptation in asexual populations. *Science (New York, N.Y.)*, 342(6164):1364–1367.

297. Wolff, U. (1989). Collective Monte Carlo Updating for Spin Systems. *Physical Review Letters*, 62(4):361–364.

298. Wollenberg, K. R. and Atchley, W. R. (2000). Separation of phylogenetic and functional associations in biological sequences by using the parametric bootstrap. *Proceedings of the National Academy of Sciences*, 97(7):3288–3291.

299. Wu, F. Y. (1982). The Potts model. *Reviews of Modern Physics*, 54(1):235–268.

300. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J., and Arnold, F. H. (2019). Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences of the United States of America*, 116(18):8852–8858.

301. Wylie, C. S. and Shakhnovich, E. I. (2011). A biophysical protein folding model accounts for most mutational fitness effects in viruses. *Proceedings of the National Academy of Sciences of the United States of America*, 108(24):9916–9921.

302. Xu, X. Y., Qi, Y., Liu, J., Fu, L., and Meng, Z. Y. (2017). Self-learning quantum Monte Carlo method in interacting fermion systems. *Physical Review B*, 96(4):041119.

303. Yang, G., Anderson, D. W., Baier, F., Dohmen, E., Hong, N., Carr, P. D., Kamerlin, S. C. L., Jackson, C. J., Bornberg-Bauer, E., and Tokuriki, N. (2019). Higher-order epistasis shapes the fitness landscape of a xenobiotic-degrading enzyme. *Nature Chemical Biology*, 15(11):1120–1128.

304. Yang, J. S., Chen, W. W., Skolnick, J., and Shakhnovich, E. I. (2007). All-atom ab initio folding of a diverse set of proteins. *Structure*, 15(1):53–63.

305. Yip, K. M., Fischer, N., Paknia, E., Chari, A., and Stark, H. (2020). Atomic-resolution protein structure determination by cryo-EM. *Nature*, 587(7832):157–161.

306. Yoshioka, N., Akagi, Y., and Katsura, H. (2019). Transforming generalized Ising models into Boltzmann machines. *Physical Review E*, 99(3):032113.

307. Younes, L. (1999). On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics and Stochastic Reports*, 65(3-4):177–228.

308. Zhang, C., Mortuza, S. M., He, B., Wang, Y., and Zhang, Y. (2018). Template-based and free modeling of I-TASSER and QUARK pipelines using predicted contact maps in CASP12. *Proteins*, 86 Suppl 1:136–151.

309. Zhao, V. Y., Rodrigues, J. V., Lozovsky, E. R., Hartl, D. L., and Shakhnovich, E. I. (2021). Switching an active site helix in dihydrofolate reductase reveals limits to sub-domain modularity. *Biophysical Journal*.

RÉSUMÉ

Tout au long de cette thèse de doctorat, nous étudierons les propriétés d'échantillonnage des machines de Boltzmann restreintes (RBM), des réseaux de neurones à deux couches utilisés pour l'apprentissage non supervisé de distributions de modèles à partir de données. Dans le cas de l'algorithme d'échantillonnage canonique de ces réseaux de neurones, l'échantillonnage alterné de Gibbs, nous montrerons qu'il est possible de trouver des trajectoires optimales entre des minima locaux du paysage énergétique, mais que ces trajectoires passent par de grandes barrières d'énergie libre. Le temps caractéristique pour passer d'un minimum à l'autre est exponentiel dans la taille du système. Par conséquent, cet algorithme est tout aussi inefficace qu'un échantillonnage naïf basé sur l'algorithme de Metropolis-Hastings.

Nous allons montrer qu'il est possible d'utiliser les représentations apprises par les machines de Boltzmann restreintes pour accélérer l'échantillonnage. Lorsque les unités cachées codent des caractéristiques essentiellement indépendantes des données, ou sont corrélées par blocs de faible dimension, la mise à jour d'une, ou d'un petit nombre d'unités cachées avec l'algorithme de Metropolis-Hastings dans l'espace caché permet un changement macroscopique des unités visibles et offre un mélange rapide entre les minima. Dans le cas d'une représentation intriquée, l'utilisation d'une pile de RBM couplées via l'algorithme de Deep Tempering améliore l'échantillonnage.

Nous nous intéresserons également à la protéine β-lactamase TEM-1 et montrerons que la plupart des mutations présentent un schéma macroscopique d'épistasie qui peut être capturé par un modèle biophysique simple à deux niveaux, qui prédit l'émergence de l'épistasie sur la base des effets additifs des mutations. Nous utiliserons de plus des modèles issus de la physique statistique, comme les RBM entrainées sur des alignements de séquences, pour étudier théoriquement les effets de ces mutations et identifier des groupes d'acides aminés encodant des fonctionnalités particulières de la classe A des β-lactamases.

MOTS CLÉS

Machines de Boltzmann Restreintes, Échantillonage, Apprentissage de représentations, β-lactamases, TEM-1, Épistasie

ABSTRACT

Throughout this Ph.D. thesis, we will study the sampling properties of Restricted Boltzmann Machines (RBM), bi-layer neural networks used for the unsupervised learning of model distributions from data. In the case of the canonical sampling algorithm of this neural network, the Alternating Gibbs Sampling, we will show that it is possible to find optimal trajectories between local minima of the energy landscape, but that these trajectories pass through large free energy barriers. The characteristic time to go from one minimum to another is exponential in the size of the system. Therefore, this algorithm is just as inefficient as a naive sampling based on the Metropolis-Hastings algorithm.

We will show that using the representations learned by the Restricted Boltzmann Machines is possible to speed up the sampling. When hidden units encode essentially independent data features or are low dimensional block-correlated, updating of one or a small number of hidden units with Metropolis-Hastings algorithm in the hidden space allows for a macroscopic change of visible units and offers rapid mixing between minima. Furthermore, using a stack of coupled RBM via the Deep Tempering algorithm improves the sampling in the case of entangled representation.

We will also focus our interest on β-lactamase TEM-1 protein and show that most mutations have a macroscopic pattern of epistasis which can be captured by a simple biophysical two-state model that predicts the emergence of epistasis based on the additive effects of mutation. We will also use models from statistical physics, such as RBM trained on sequence alignments, to theoretically study the effects of these mutations and identify amino acid clusters encoding particular functionalities of the class A β-lactamases.

KEYWORDS

Restricted Boltzmann Machines, Sampling, Representation learning, β-lactamases, TEM-1, Epistasis