Uncertainty quantification of explicability for Machine Learning : post-selection inference over interpretable biological features

Antoine Villie

THESE de DOCTORAT DE L'UNIVERSITE CLAUDE BERNARD LYON 1

Ecole Doctorale N°341
Évolution, Écosystèmes, Microbiologie, Modélisation

Discipline : **Mathématiques**

Soutenue publiquement le 31/03/2023, par :
Antoine Villié

Quantifier l'incertitude de l'explicabilité en apprentissage automatique : Inférence post-sélection sur des caractéristiques biologiques interprétables

---

Devant le jury composé de :

Mme FOUGERES Anne-Laure	Professeur des Universités	Université Lyon 1	Présidente
Mme LAINE Elodie	Maître de Conférences	Sorbonne Université Paris	Rapporteure
Mme VINGA Susana	Professeur Associé	Université de Lisbonne	Rapporteure
M. NEUVIAL Pierre	Directeur de Recherche	CNRS Toulouse	Rapporteur
Mme SAGOT Marie-France	Directrice de Recherche	INRIA Lyon	Examinatrice
M. MAHE Pierre	Professeur des Universités Associé	Université Grenoble Alpes / bioMérieux	Examinateur
M. JACOB Laurent	Chargé de Recherche	CNRS Lyon	Directeur de thèse
M. DE CASTRO Yohann	Professeur des Universités	Ecole Centrale de Lyon	Co-directeur de thèse
M. VEBER Philippe	Ingénieur de Recherche	CNRS Lyon	Invité

Laboratoire de Biométrie et de
Biologie Évolutive
Université Claude Bernard Lyon 1
Bâtiment Grégor Mendel
43 boulevard du 11 novembre 1918
69622 VILLEURBANNE

École doctorale Évolution Écosystèmes
Microbiologie Modélisation
43 boulevard du 11 novembre 1918
69622 VILLEURBANNE

Résumé

Ce travail de thèse cherche à participer au rapprochement entre deux littératures et communautés scientifiques, qui sont jusqu'à présent relativement étanches l'une à l'autre : l'apprentissage automatique et la biologie computationnelle.

Les réseaux de neurones artificiels, une classe particulière de méthodes d'apprentissage automatique, ont en effet été largement développés au cours des dernières années pour des applications aux séquences biologiques. Ils ont permis des avancées importantes dans des champs variés de la biologie, notamment en génomique régulatrice, par exemple en permettant de prédire les liaisons des facteurs de transcription, les niveaux d'expression des gènes, l'accessibilité de la chromatine ou encore les modifications des histones (Zhou & Troyanskaya, 2015; Kelley et al., 2018; Avsec et al., 2021a,b). Ces méthodes sont principalement utilisées pour leurs capacités de prédiction d'un trait biologique à partir d'une séquence, et sont couramment évaluées sur leurs capacités à faire des prédictions correctes à partir de données qui n'étaient pas présentes dans leurs jeux d'entraînement.

En parallèle, de nombreuses méthodes ont été développées en biologie computationnelle dans le but de tenter d'expliquer, plutôt que de prédire, ces phénotypes ; le but étant alors d'aider à leur compréhension. Dans la suite, on nommera simplement ces approches *méthodes explicatives*. On peut notamment citer les études d'association pan-génomiques (Visscher et al., 2017), qui ont pour but l'identification de variants génomiques qui corrèlent avec le trait biologique d'intérêt, par exemple en utilisant des modèles linéaires. On peut également citer le nombre croissant d'algorithmes cherchant à détecter des motifs de séquences (Bailey et al., 2015). Ces motifs, qui peuvent être compris comme de petites séquences biologiques probabilistes, sont notamment connus pour être des éléments de base en génomique régulatrice. Bien que ces méthodes aient mené à de nombreuses découverts durant les dernières années, elles restent soumises à de nombreuses limitations. Parmi d'autres, on peut citer des restrictions sur le type de variants génétiques qu'elles peuvent utiliser, ainsi que des limitations statistiques liées à la manière dont elles gèrent un grand nombre de tests, résultant d'un grand nombre de variants.

Les réseaux de neurones ont initialement été développés dans un but prédictif, et les architectures les plus utilisées ont souvent été créées pour des applications différentes des séquences biologiques, telles que la vision par ordinateur (réseaux de neurones convolutionnels (CNNs), Lecun & Bengio, 1995) ou encore le traitement automatique du langage (méchanismes d'attentions Vaswani et al., 2017). Malgré cela, ils reposent parfois sur des objets mathématiques qui peuvent facilement s'interpréter d'un point de vue biologique. Par exemple, les filtres de la première couche de convolution d'un CNN sont homogènes aux données d'entrées, et peuvent être vus comme de petits bouts d'images dans un cadre de vision par ordinateur, ou comme des motifs de séquences si le réseau est appliqué à des séquences biologiques correctement encodées. Sur des réseaux plus profonds, qui permettent de modéliser des interactions plus complexes sur les données d'entrées, où cette interprétation ne peut pas être obtenue aussi directement, de plus en plus de techniques cherchent à expliquer les prédictions en reconstruisant des caractéristiques biologiques interprétables à partir de ces réseaux (Novakovsky et al., 2022a). Ces réseaux, associés avec des méthodes d'interprétabilité, peuvent donc être considérés comme des méthodes permettant de découvrir des variants biologiques associés à un trait phénotypique, puisque ces variants en permettent la prédiction, ce qui est donc un but commun avec les méthodes explicatives.

En revanche, contrairement aux méthodes explicatives issues de la bioinformatique, qui s'attachent à quantifier statistiquement l'association entre les variants et le phénotype d'intérêt, il n'y a à notre connaissance que peu de travaux qui cherchent à quantifier l'association entre les caractéristiques apprises par un réseau de neurones et la sortie de ce réseau. Mais cela soulève des questions d'inférence post-sélection, un champ de recherche très actif en statistiques.

L'inférence post-sélection cherche à créer des procédures de test valides dans un cas où les hypothèses nulles ont été construites, ou sélectionnées, en ayant utilisé les mêmes données que celles qui seront utilisées pour tester ces hypothèses. En effet, le cadre classique de l'inférence suppose que les hypothèses ont été formulées indépendamment des données, et donc l'utilisation d'outils issus de ce cadre classique sur des hypothèses sélectives mène à des biais, et à des p-valeurs trop optimistes dans le cadre qui nous intéresse (Benjamini, 2020; Taylor & Tibshirani, 2015). L'inférence conditionnelle cherche alors à corriger la procédure de test, en prenant en compte la sélection préalable.

Cette thèse cherche donc à combiner les avancées en apprentissage automatique, notamment en termes d'explicabilité, avec les méthodes explicatives. Nous voulons montrer comment les réseaux de neurones permettent de surmonter certaines des limites inhérentes à ces méthodes, tout en prenant en compte les problématiques liées à l'inférence post-sélection.

Dans un premier temps, nous nous concentrons sur les méthodes bioinformatiques, et plus particulièrement sur les études d'association pan-génomiques. A partir d'un jeu de données rassemblant un ensemble d'individus pour lesquels un certain nombre de variants génomiques et le phénotype sont connus, ces études cherchent à détecter les variants significativement associés avec le phénotype, le plus souvent en utilisant un modèle linéaire. Le nombre de variants testés étant généralement très élevé, ces méthodes doivent donc prendre en compte la multiplicité des tests.

On peut alors identifier au moins deux facteurs limitant ces études. Premièrement, leurs découvertes sont nécessairement restreintes à la liste de variants dont elles disposent en entrée. Ces listes sont à leur tour limitées, que ce soit pour des questions aussi bien expérimentales que de modélisation. Deuxièmement, ces listes contiennent généralement énormément de variants différents, et ces études reposent généralement sur de l'inférence simultanée, en cherchant à contrôler la probabilité de faire au moins une fausse découverte, pour gérer la multiplicité des tests. Cette approche est relativement stricte et peut limiter fortement le nombre de découvertes effectuées.

Nous montrons alors comment l'apprentissage automatique permet de répondre à cette première limite. En effet, on peut souvent décomposer ces méthodes en deux étapes: construction d'une représentation des données, puis application d'un modèle linéaire sur cette nouvelle représentation. L'étape d'entraînement d'un réseau de neurones cherche alors à trouver les bons paramètres du modèle linéaire, mais également à apprendre la meilleur représentation possible pour le problème posé. Certaines représentation sont facilement interprétables, on peut par exemple penser à représenter une séquence biologique par son contenu en k-mers — des sous-séquences de longueur k. Les représentations apprises par les CNNs à une couche reposent sur la comparaison entre la séquence et des motifs de séquence, ce qui là encore donne accès à une interprétation biologique intéressante. Nous réalisons alors un état de l'art des méthodes qui cherchent à expliquer les prédictions d'un réseau de neurones à partir d'éléments biologiques. Cela nous permet de montrer le potentiel que représente l'apprentissage automatique, couplé avec les méthodes d'interprétation, dans la recherche de nouveaux variants associés avec un trait biologique.

En revanche, ces variants sont donc issus d'une procédure de sélection, qu'il faudra prendre en compte au moment de tester leur association avec le trait biologique d'intérêt. De plus, cette sélection intervient parmi un ensemble infini, ce qui rajoute une difficulté supplémentaire. Nous étudions alors les différentes approches qui permettent d'obtenir des procédures de tests valides, dans le cas où une sélection préalable des objets testés a eu lieu. Cette question est en fait très proche de la seconde limitation soulevée pour les études d'association pan-génomiques, et nous présentons donc différentes approches pour résoudre ce problème. Nous explicitons la problématique rencontrée lors des tests multiples et en présence de sélection, puis nous proposons un état de l'art des connaissances en inférence conditionnelle, tout en pointant les limites actuelles. Particulièrement, ces méthodes ne fonctionnent pas pour de la sélection parmi un ensemble infini, ce qui est pourtant le cas des variants sélectionnés par les réseaux de neurones.

La seconde moitié de la thèse est dédiée à l'introduction de la méthode SEISM — SElective Inference for Sequence Motifs. Cette méthode se base sur des réseaux de neurones pour faire de la sélection de variants génomiques, puis propose une procédure d'inférence valide pour tester l'association entre ces variants et le trait biologique. Cette méthode a fait l'objet d'un article (Villié et al., 2022) et une implémentation PyTorch est disponible : `https://gitlab.in2p3.fr/antoine.villie1/seism`.

Tout d'abord, nous commençons par introduire un cadre pour formaliser la définition des méthodes d'apprentissage automatique en tant que méthodes de sélection de variants génétiques. Nous montrons que les réseaux convolutionnels à une couche rentrent dans ce cadre, et sélectionnent effectivement des motifs de séquences. En revanche, ces méthodes sont originellement créées pour des problématiques de prédiction, et n'ont donc pas un

comportement satisfaisant pour faire de la sélection, que ce soit au niveau de la stabilité ou de la pertinence des variants sélectionnés. Nous introduisons donc plusieurs modifications afin d'améliorer leurs performances dans ce domaine, que ce soit dans l'architecture du réseau en elle-même ou en ce qui concerne son optimisation.

Nous introduisons ensuite le problème de *de-novo* motif discovery, qui cherche à déterminer des motifs de séquences associés avec un trait biologique, et nous montrons que notre procédure de sélection obtient des performances similaires aux méthodes explicatives issues de la biologie computationnelle, dont c'est le but premier. Nous soulignons également les différences de modélisation qui sous-tendent ces différentes approches. En effet, les méthodes bioinformatiques et celles issues de l'apprentissage statistique modélisent de manières différentes les distributions de k-mers à partir d'un motif de séquence, ce qui mène à des interprétations différentes.

Puis nous définissons un cadre statistique pour tester l'association entre les motifs découverts et le trait biologique d'intérêt. Nous commençons donc par introduire un modèle Gaussien, des hypothèses nulles et des statistiques de test associées. Les méthodes d'inférence conditionnelles reposent sur le concept d'événement de sélection : l'ensemble des traits biologiques qui auraient mené à la sélection des mêmes variants génomiques si on avait appliqué à ces traits la même procédure de sélection que celle appliquée au vrai phénotype, celui présent dans le jeu de données initial. Le cœur de l'inférence conditionnelle consiste à construire la distribution des statistiques de test sous l'hypothèse nulle, conditionnellement l'événement de sélection.

Mais nous montrons que dans ce cas, une expression analytique pour cette distribution semble hors d'atteinte. De plus, cet événement étant ici de probabilité nulle, le conditionnement devient alors mal défini, ce qui complique encore le problème, avec l'apparition potentielle du paradoxe de Borel-Kolmogorov (Bungert & Wacker, 2022). Les outils classiques d'inférence conditionnelle ne peuvent alors pas s'appliquer directement.

Nous contournons donc le problème en introduisant un maillage de l'espace des motifs : au lieu de tester un motif ponctuel, nous allons tester un ensemble de motifs qui lui sont proches. Cette approche entraîne alors une modification des hypothèses nulles et des statistiques de tests, avec plusieurs options possibles menant à des interprétations différentes.

Une expression analytique de la distribution nulle conditionnelle restant hors d'atteinte, nous avons recours à une procédure d'échantillonnage afin de l'approximer. Nous recourrons donc à un algorithme dit de *hit-and-run* : une procédure d'échantillonnage par rejet qui diminue considérablement le taux de rejet par rapport à une approche naïve, au prix d'une dépendance entre les points tirés.

Nous proposons également un travail autour des hypothèses nulles, qui sont composites dans ce cadre. C'est-à-dire qu'une même hypothèse nulle peut être décrite par plusieurs paramètres. Afin de faciliter l'utilisation de SEISM sur des données réelles, nous cherchons à limiter le nombre d'hypothèses nécessaires sur ces paramètres, et nous introduisons donc plusieurs invariances afin de rendre la distribution nulle conditionnelle indépendante de ces paramètres. Cette problématique dépasse le cadre de SEISM, mais représentait à notre connaissance une problématique peu étudiée jusque-là.

Grâce à plusieurs expériences, nous montrons que la procédure introduite est correctement calibrée, et plus puissante qu'une procédure basée sur une stratégie de data-split, consistant à sélectionner les variants sur une partie seulement des données, et à les tester sur la seconde partie. Cela valide l'intérêt de la procédure SEISM, notamment dans un cadre de petits jeux de données.

En revanche, l'inférence conditionnelle par échantillonnage se révèle couteuse d'un point de vue computationnel, et nous étudions l'impact de différents paramètres sur le temps de calcul. Nous montrons donc une complémentarité entre une approche conditionnelle et une approche data-split, en fonction du jeu de données et des différents paramètres de sélection.

Nous appliquons enfin SEISM sur un jeu de données réelles, et montrons que notre procédure semble robuste à l'hypothèse Gaussienne sur la distribution des traits biologiques.

Pour conclure, ce travail vise à tirer profit des récentes avancées dans le domaine de l'apprentissage automatique, de l'explicabilité de ces méthodes et en inférence post-sélection, afin de dépasser la seule notion d'explicabilité d'un réseau de neurones, et d'en faire un outil qui corresponde aux critères des méthodes de la biologie computationnelle. Il constitue une preuve de concept sur un cas d'utilisation relativement simple, et permet de lever plusieurs barrières théoriques. Mais le cadre introduit et la procédure statistique développée se veulent les plus généralistes possible, et adaptables à de nombreuses méthodes de sélection et autres types de variants génomiques. Cela ouvre la voie à de nombreux développements afin d'en accroître le champ d'application, constituant des pistes prometteuses pour de futurs travaux.

Remerciements

Tout d'abord, je tiens à remercier Laurent Jacob pour la confiance qu'il m'a accordée en me proposant ce sujet de thèse. Merci beaucoup de m'avoir accompagné et guidé au cours de ces trois années, et d'avoir partagé avec moi ton expérience en recherche scientifique. Merci également pour ta bienveillance et ta disponibilité.

Merci à Yohann de Castro, pour l'intérêt que tu as porté à ce sujet, pour ton implication et pour ton expertise. Tu nous as permis d'explorer de nombreuses directions, et travailler avec toi a vraiment été stimulant et formateur.

Merci beaucoup Philippe Veber pour ton aide sur ce projet, et pour les échanges que nous avons eu pendant ces trois années. Merci de partager ainsi ta passion, toujours avec beaucoup de pédagogie, et sur des sujets variés.

J'ai eu énormément de chance de vous avoir comme encadrants, et ces quelques années ont été très agréables grâce à vous !

Merci également aux membres du jury, Anne-Laure Fougères, Élodie Laine, Pierre Mahé, Pierre Neuvial, Marie-France Sagot et Susana Vinga. Merci pour votre temps, pour votre expertise, pour l'intérêt que vous avez porté à ce sujet et pour votre participation à ce jury.

Je tiens à remercier les membres de mon comité de suivi de thèse : Chloé Azencott, Vincent Daubin, Jean-Philippe Rasigade et plus particulièrement Benoît Cournoyer, pour votre accompagnement et vos conseils toujours bienveillants au cours de ces trois années.

Merci à François Gindraud, pour ton temps et pour tes conseils sur le développement de SEISM. Ta contribution a été précieuse, et j'ai énormément appris grâce à toi.

Merci Claire pour ton accueil dans le LBBE, j'ai vraiment apprécié travailler avec toi pendant le début de cette thèse. Ce fut une belle rencontre, et je te souhaite le meilleur, tant sur le plan professionnel que personnel !

Merci à Alexandre, Alexia, Djivan, Johanna, Luca, Mary, Maxime, Nicolas, Théo et Thibault, pour ces bons moments partagés.

Merci à toute l'équipe BAOBAB, et plus particulièrement Marie-France, Sabine, Arnaud, Vincent pour cette super équipe, avec une super ambiance.

Merci à Annaël, Caro, Damien, Jade, Kaïs, Lola, Lucas, Romain et tout le BCVIL pour m'avoir permis de décompresser sur les terrains chaque semaine, mais aussi en dehors des gymnases, et de repartir reboosté. Merci Antoine, Bruno, Cyrielle, Eloïse, Emeline, Florence, Ludo, Quentin et Tanguy pour toutes ces années d'amitiés qui représentent beaucoup pour moi. Merci pour votre capacité à comprendre les règles, à dire tout le temps oui, pour votre calme, votre fair-play, la finesse de vos analyses sportives, pour votre volonté gravir des sommets, votre présence, votre modération et votre bonne foi.

Un grand merci à mes parents et à mes sœurs, vous m'avez toujours soutenu et encouragé dans mes différents projets. C'est vous qui m'avez donné cette curiosité et cette envie d'apprendre, et cette thèse est donc en grande partie grâce à vous.

Enfin, merci Chloé, pour partager avec moi cette aventure, pour la confiance que tu m'apportes, pour ton soutien constant, ta bonne humeur et surtout ton humour irrésistible.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive numbers
$\mathbb{R}^{n\times m}$	The set of real-valued matrices of size $n \times m$
$\boldsymbol{I}_n$	The identity matrix in $\mathbb{R}^{n\times n}$
$\boldsymbol{C}_n$	The centering matrix in $\mathbb{R}^{n\times n}$
$\mathbf{0}\,,\,\mathbf{1}$	The all-zeros and all-ones vectors
$\boldsymbol{A}^T$	The transpose of matrix $\boldsymbol{A}$
$\boldsymbol{y}^\perp$	The orthogonal complement of vector $\boldsymbol{y}$
$\langle \cdot\,,\,\cdot \rangle$	The dot product
$\lVert\cdot\rVert_p$	ℓ_p norm
$\mathcal{U}(0,1)$	The uniform distribution between 0 and 1
$\mathcal{N}(\mu,\sigma^2)$	The Gaussian distribution with mean μ and standard deviation σ
$\mathbb{P}$	Probability
$\mathbb{E}$	Expected value
$\mathcal{L}$	Likelihood function

Biology Basics for the Mathematically Inclined

DNA	Deoxyribonucleic acid: a polymer carrying genetic instructions for the development, functioning, growth and reproduction of organisms. This polymer is composed of two polynucleotide chains, where each nucleotide, or base, is either adenine (A), cytosine (C), guanine (G) or thymine (T).
Phenotype	The set of observable characteristics or traits of an organism.
Genotype	The complete set of genetic material of an organism.
SNP	A single-nucleotide polymorphism is a substitution of a single nucleotide at a specific position in the genome. Such variations in the DNA sequence can affect the phenotype of an individual and can be associated with diseases.
Indels	It refers to insertion and/or deletion of nucleotides into DNA, usually less than 1000 bases long.
Translocations	When a segment of DNA is moved to a new location on the same or a different chromosome.
Copy-number variations	Gain or loss of a segment of DNA resulting in an alteration in the number of copies of a gene or set of genes.

Mobile genetic elements	Segments of DNA that can move around within or between genomes. They include transposable elements, plasmids (circular fragments of DNA) and viruses. For instance, antibiotic resistance genes, if located in a such a segment, can be transported to share genetic code with neighboring bacteria.
Microarray	A collection of DNA spots attached to a solid surface. Each spot contains a specific DNA sequence, for instance a short section of a gene, or a small sequence with a known SNP. Hybridization between the probes and the target is then detected and quantified, usually using fluorescence or chemiluminescence. Such chips can be used to measure the expression levels of genes, or to genotype a genome.
High-throughput sequencing	Also known as next-generation sequencing, it refers to sequencing methods that allow to sequence the entire genome at once, by sequencing multiple DNA molecules in parallel, enabling hundreds of millions of DNA molecules to be sequenced at a time. Usually, the genome is first fragmented into small pieces, and then multiple fragments are sequenced at once.
Reference genome	A representative example of the DNA sequence in one idealized individual organism of a species.
Core or accessory genome	The core genome represents the shared and conserved genetic material of a species, usually composed of genes that are essential for basic cellular functions, while the accessory genome represents the material shared within only one or some individuals, and usually provides specific functions or adaptations, such as antibiotic resistance.
Transcription factor	A protein that regulates the transcription of DNA into RNA by binding to specific sequences of DNA and modulating the activity of RNA polymerase. Those specific sequences are named binding sites, and sets of similar sequences can be represented using sequence motifs.
Sequence motif	A nucleotide (or amino-acid) sequence pattern, assumed to be related to some biological function. They can be mathematically represented using position weight matrices, or graphically as sequence logos.

Foreword

In the recent years, neural networks have been successfully used for making predictions from biological sequences. In particular, they have brought significant improvements in regulatory genomics, *e.g.* to predict cell-type specific transcription factor binding, gene expression, chromatin accessibility or histone modifications from a DNA sequence (Zhou & Troyanskaya, 2015; Kelley et al., 2018; Avsec et al., 2021a,b). These methods are usually evaluated based on the accuracy of their predictions or decisions, and they have made great progress in this regard. It is also worth noting that the majority of these algorithms were not initially developed for application to biological data, but rather to computer vision (convolutional neural networks (CNNs), Lecun & Bengio, 1995) or to natural language processing (attention mechanisms, Vaswani et al., 2017) problems.

Although most neural networks were initially designed for prediction, some architectures spontaneously reveal features that lend themselves easily to biological interpretations. For instance, the trained filters of elementary one-layer CNNs have a straightforward interpretation as position weight matrices, and therefore as sequence motifs. However, the lack of interpretability is a commonly outlined limitation of deeper networks, for which the ability to model complex interactions between the features results in significant performance gains. Interpretability considerations are becoming increasingly important in the machine learning community, particularly for biological applications. As a result, approaches for extracting interpretable biological features from these networks have been developed, and highlight various genetic variant types (Novakovsky et al., 2022a).

In addition to seeking to predict a biological trait, these neural networks — if necessary coupled with explainability approaches — can then be thought of as methods for selecting variants that appear to be somewhat related to this trait, since they are useful for the prediction. However finding features somewhat associated with a trait is often not enough, as an observed non-zero association can be spurious. And that's why some methods from the computational biology literature are committed to quantifying the uncertainty of those associations. These explanatory methods, which have developed greatly in recent years, seek to explain, rather than predict, certain biological traits, and help to understand the underlying biology. Genome-wide association studies (Visscher et al.,

2017) for example find genetic variants (traditionally single-nucleotide polymorphisms) correlated with a trait. We can also mention the increasing number of algorithms designed to tackle the *de-novo* motifs discovery task (Bailey et al., 2015). Sequence motifs, or small probabilistic biological sequences, are indeed historical and basic elements of regulatory genomics (Harr et al., 1983; Schneider & Stephens, 1990). In this framework, the quantification of the association between the genetic variants and the biological trait is of great importance. Although they have led to a remarkable range of discoveries in recent years, these explanatory methods face different challenges, whether it is to extend their scope, the types of genetic variants they consider or regarding statistical considerations.

Neural networks then represent a promising direction to try to overcome those limitations. However, to our knowledge, quantifying the significance of those associations between interpretable features extracted from a neural network and biological traits has only received little attention and raises many challenges.

In particular, the genetic variants resulting from the training and interpretation of neural networks have been selected among rich class of variants, and this selection has to be accounted for when it comes to the test step.

In order to bridge those different approaches, this thesis aims to go beyond interpretations, by leveraging recent development in post-selection inference to quantify the uncertainty of explicability for machine learning.

To accomplish this goal, this thesis is organized in four chapters. The first two chapters aim to give a precise state of the art of the different existing methods, whether to identify relevant genetic variants or to test them, and to identify some limitations to which we will try to provide answers in the following chapters.

- Chapter 1 first proposes a rapid overview of genome wide association studies. Once we provide a brief explanation of how they operate, we can then identify several limitations that current approaches are subject to. We will then introduce several machine learning algorithms for biological sequences and show how, in addition to properly predicting a biological trait, they learn a data representation that is relevant to the task at hand. This representation can be very complex depending on the considered algorithm, but can occasionally be explained using mathematical objects that lend themselves easily to biological interpretations. In this regard, the machine learning methods constitute a promising direction for addressing some of the restrictions of the genome wide association studies. We also conduct an overview of existing explainable machine learning methods, allowing us to derive interpretable features for more complex networks, in order to try to give a comprehensive picture of neural network contributions to the considered limitations.

- Chapter 2 focuses on the statistical issues arising when one wants to test the association between genetic variants and a trait. Genome wide association studies have to deal with a huge number of different variants to test, and the current simultaneous inference approach is being challenged. However, neural networks and interpretation methods tend to select only a limited number of genetic variants, among rich and potentially infinite classes. This places us in a post-selection inference framework, a very active field of research in recent years. Multiple testing and post-selection

inference are different approaches that can tackle similar problematics, and we will try to give a comprehensive overview of the challenges and of recent developments. This will allow us to identify some limitations of current methods, preventing their application to our framework of interest.

After having completed this overview, we will introduce SElective Inference for Sequence Motifs (SEISM), a valid statistical inference procedure for the interpretable features extracted from a trained neural network, depicted in Figure 1.

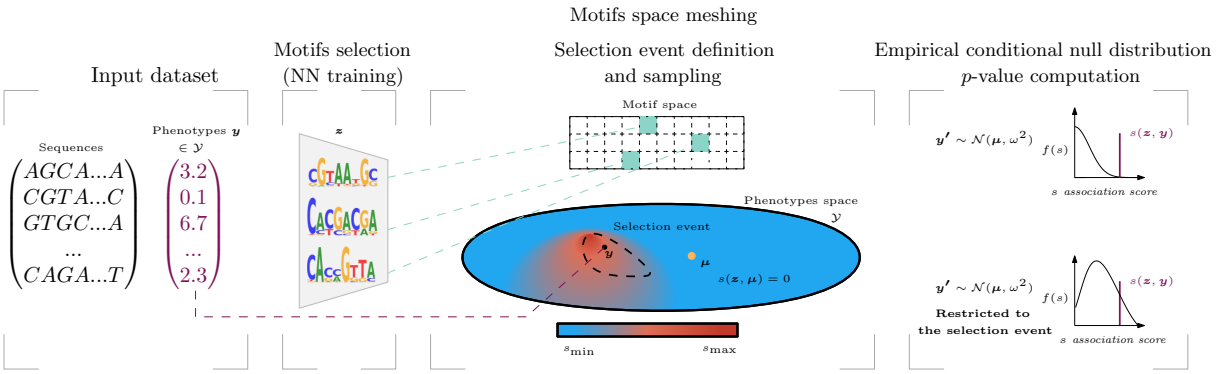


Figure 1: Overview of SEISM. (a) The input is a set of sequences and corresponding phenotypes in some space $\mathcal{Y}$ (b) It trains a convolutional neural networks to predict a phenotype from sequences, which leads to the selection of sequence motifs. (c) Then SEISM partitions the space of motifs to quantize the selection. The selection event is the set of phenotype vectors that would lead to selecting an element in the same mesh. (d) Using a sampling strategy, SEISM builds a null distribution for the test statistic, conditional to the selection event. The p-value associated with a selected motif is the quantile of its score under this distribution.

- Chapter 3 presents how SEISM can be used to discover sequence motifs. We first cast commonly used CNNs in a feature selection framework, and we propose multiple modifications to classical CNNs in order to improve their performances as feature selection tools rather than as predictive tools. We also propose several methods to optimize this network, and we show that SEISM achieves similar performance on *de-novo* motifs discovery tasks as state of the art explanatory methods from the bioinformatics literature.

- Chapter 4 aims at providing a valid statistical inference procedure for the sequence motifs discovered by SEISM. But existing methods for selective inference only apply to a selection from a finite set, while the sequence motifs are selected from a continuously infinite set. We work around this issue by quantizing our selection to a very large but finite space, making it amenable to existing strategies. We show that SEISM results in a calibrated test procedure, and compare it with a standard data-split strategy. We also work on the composite aspect of our null hypotheses, and provide invariance results suggesting a practical procedure with only a few assumptions regarding the distribution of the data. To our knowledge, there was a blind spot in sampling-based post-selection inference approaches beyond our specific context. The results of this chapter, although illustrated on the specific case

of sequence motifs, remain quite general and can be extended to other features and types of association.

We provide a preprint (Villié et al., 2022), as well as a PyTorch implementation `https://gitlab.in2p3.fr/antoine.villie1/seism`. This implementation contains all the experiments presented in this work, and it should be easy to apply on any new dataset.

CHAPTER 1

Machine learning and explainable AI for enhancing Genome Wide Association Studies

Genome Wide Association Studies have become a widely-used tool in the search for genetic variations that are associated with certain traits or diseases. Focusing on the entire genome at once, rather than on individual genes, they have provided valuable insights into the genetics of a wide range of phenotypes. In this chapter, we will first briefly describe the methodology associated with these studies, allowing us to draw attention to two of their inherent limitations. While one of those limitations relates to statistical considerations and will be the focus of Chapter 2, the other one is about the list of genetic variants that are used in those studies.

We then examine how current developments in machine learning can help us to overcome this limitation. In recent years, machine learning models for biological sequences have indeed gained popularity as a way to analyze genetic data. To that end, they learn a data representation and identify underlying patterns and relationships. The representation learned by these models can then provide insights into the underlying biology, and can also be linked to new classes of genomic variants that might be employed in Genome Wide Association Studies.

This direction can be pursued even further, thanks to explainable artificial intelligence, an increasingly important research area for machine learning, especially in biology. It refers to the use of machine learning models that can provide clear and understandable explanations for their predictions. The various associated techniques also bring to light new promising genomic variants, as we will see in a third step, after having proposed an overview of the current state of knowledge in this area.

Finally, we will see that these advances inevitably raise inference-related questions, complementary to the limitation previously identified, leading us into Chapter 2.

1.1 Methodology and limitations of GWAS

Genome Wide Association Studies (GWASs) are observational studies that aim at detecting associations between genetic variants and phenotypes. They are generally traced back to the publication of Wellcome Trust Case Control Consortium (2007), the first large scale GWAS. It leaded to the discovery of more than 20 association signals between single-nucleotide polymorphisms (SNPs) and various disorders, such as artery disease, rheumatoid arthritis and diabetes. As of 29 November 2022, the National Human Research Institute Catalog of Published GWAS (Welter et al., 2014) contained 344 498 SNP-trait significant associations at the genome-wide p-value threshold of 5×10^{-8}. But GWAS are not limited to human genetics, and the GWAS Atlas (Tian et al., 2020) offers a curated database of variant-trait associations for ten plant species and five animal species. They are therefore used in many different fields, from risk factor identification to plan breeding (Gali et al., 2019).

A typical GWAS relies on microarrays or on sequencing technologies to genotype some individuals with different phenotypes. Designing microarrays requires prior knowledge about the location of the SNPs in the genome, which prevents the study of rare SNPs, since those may be missing from the chip. High throughput sequencing overcomes this limitation, but comes with an intensive computational step as it maps all the reads to a reference genome, in order to identify the SNPs. In doing so, both approaches build a matrix of the variants' presence or absence pattern (Table 1.1).

	Person 1	Person 2	...	Person n
Variant 1	0	1		0
Variant 2	1	0		0
...				
Variant v	0	0		1
Trait	Control	Case		Control

Table 1.1: Example of a dataset used for a GWAS analysis. For instance, the association between a variant and the phenotype may be tested using a linear model.

Next, statistical analysis is carried out to indicate the significance of the association between each of the variants and the phenotype, for instance using a linear model. The results are often visualized using a *Manhattan plot* where the x-axis shows the genomic coordinates and the y-axis displays the negative logarithm of the association p-value for each SNP (Figure 1.1). Although Visscher et al. (2017) underline the remarkable range of discoveries that were facilitated using GWASs, they highlight five factors influencing the potential of a GWAS to find significantly associated variants for a particular trait.

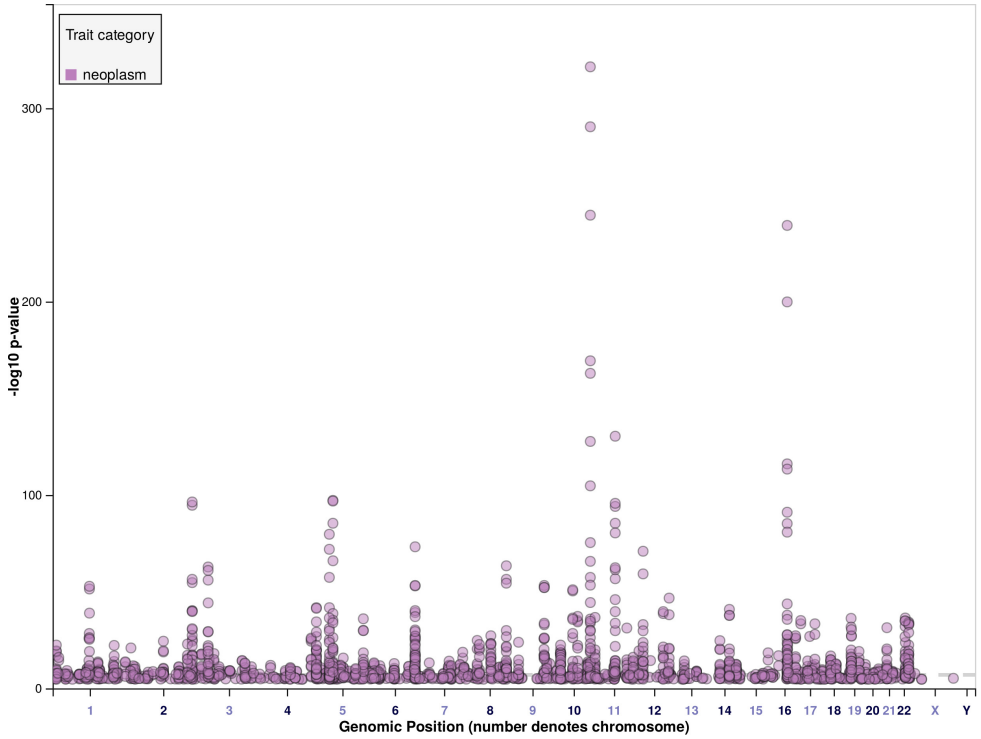


Figure 1.1: Example of *Manhattan plot* for breast carcinoma, obtained using the GWAS catalog (Welter et al., 2014). In particular, it can be seen that some SNPs on chromosomes 10 and 16 (and more precisely on the regions of the FGFR2 and CASC16 genes) are associated with the development of this cancer.

Three of them are extraneous to the design of the study:

- The number of different variants affecting this trait in the population,
- The genetic architecture, effect sizes and the frequencies of those variants,
- The heterogeneity of the trait (depending on the biology of the trait, as well as the ability to measure it precisely).

The two other factors are intrinsic: the experimental sample size, and the panel of genome-wide variants that are used in the GWAS.

While the experimental sample size may at first sight appear as extraneous, it mainly limits the success of the GWAS through the minimum achievable p-value it authorizes. A rare variant with a low effect size indeed requires a large number of samples to be detected at the genome-wide p-value 5×10^{-8}. Increasing this threshold would then lead to a greater number of detections. It is therefore necessary to understand how it was set and why it is so stringent. It was first introduced in Risch & Merikangas (1996) as a Bonferroni corrected significance threshold, to account for the multiplicity of tests, see Chapter 2 Section 2.2. The authors anticipated the evolution of knowledge about the human genome, and estimated that, in total, the human genome would contain 100 000 genes, with an average of 5 diallelic SNPs of interest in each gene, resulting in a total of 10^6 variants. Then, controlling the family-wise error rate, that is the probability of

rejecting at least one true null hypothesis, at 5×10^{-2} using the Bonferroni correction gives a nominal p-value of $5 \times 10^{-2}/10^{6} = 5 \times 10^{-8}$. Although it relies on obsolete assumptions, this threshold is still widely used and similar values have been computed using more complex assumptions and strategies (Dudbridge & Gusnanto, 2008). This demonstrates that, given a constant experimental sample size, modifying how the multiplicity of tests is accounted for may result in an increased number of findings. There exist several ways to work around this issue, and it will be the focus of Chapter 2.

High throughput sequencing overcomes some limitations related to the panel of genome-wide variants compared to microarrays. In contrast to the design of SNPs arrays, that requires some knowledge about the genome of the organism and about the location of the SNPs, this approach does not require the definition of an a priori list of SNPs, thus enabling the discovery of new variants. But it still requires a reference genome to map the reads and identify the SNPs, which limits its applications to species for which such a reference genome has already been assembled. Even the human reference genome is frequently updated, due to being incomplete (Altemose et al., 2014) and to the occurrence of a reference bias (Sousa & Hey, 2013). The use of such a reference genome becomes unsuitable for bacterial species with a large accessory genome (the portion of the genome that is not present in all strains). Moreover, relying on SNPs ignores structural variations in the sequences, such as insertion-deletions, translocations or copy-number variations. But this ignored structural variation may explain a significant number of phenotypes. That's why some recent GWAS methods rely on k-mers, the substrings of length k contained in a sequence, whose presence represents a wide class of genetic variants, including the SNPs, but also mobile genetic elements and more (Rahman et al., 2018; Jaillard et al., 2018; Roux de Bézieux et al., 2022). Such methods are frequently described as reference-free and agnostic, in the sense that no prior knowledge or alignment step is required. A promising way to push GWASs even further is then to run them on broader class of genetic variants, while finding new inference methods to improve statistical power. Machine learning techniques can then offer great opportunities to that end, as discussed in the following sections.

1.2 Machine Learning for biological sequences overview

This section aims at clarifying the existing link between GWASs and machine learning. In both cases, these methods look for associations between some input data (*e.g.* the genetic variants in the case of GWASs) and some output (the phenotype). While this link is direct in GWASs, machine learning constructs a representation of the input to predict an output. We will be able to highlight this connection thanks to the introduction of linear models. Then, we will focus on kernel methods and more advanced learning models, allowing to create new representations of the inputs and new associations with the output.

Machine learning is a subfield of artificial intelligence, that deals with the development of algorithms and statistical models that enable computers to learn from data and make predictions or decisions without explicit instructions. It involves the use of computational methods to extract knowledge from data and improve the performances of the model in

an automated way, the goal being that the model generalizes well to new and unseen data. Despite being initially created for such tasks, the models trained on biological data can be used to discover relevant genetic variants, especially taking advantage of the recent advances in explainable artificial intelligence. The different machine learning approaches are generally grouped into three major categories, relating to the nature of the problematics:

- In *supervised learning*, the algorithm is provided with a training dataset, containing both inputs $(x_1, \ldots, x_n)$ and corresponding outputs $(y_1, \ldots, y_n)$, and aims at learning a mapping function f from the input space to the outputs. If this task is successful, the algorithm should be able to precisely predict the outputs $f(x) = y$ from inputs data x that were not included in the training set. Classification problems (such as predicting whether a bacteria will resist to an antibiotic given its genotype) and regressions problems, where the outputs are not restricted to a discrete set of values, are the main tasks for supervised learning.
- On the contrary, *unsupervised learning* has no response variable, and the algorithm attempts to find structure in the inputs. While this can be a goal in itself, such as in a fraud monitoring framework using outlier detection, it can also be a step among others, for instance by discovering genetic variants that segregate a population, thus identifying suitable features which can then be used in a supervised learning framework.
- In *reinforcement learning*, the algorithm interacts with a dynamic environment and tries to perform a certain goal, through trial-and-error. Although it is widely used in other domains, such as autonomous driving, its application on biological sequences data is for now quite limited.

This thesis mainly focuses on supervised learning approaches, using biological sequences as inputs, and phenotypes as labels. In most of the approaches, the sequences are numerically represented using one-hot encoding (OHE): a sequence with length ℓ over an alphabet $\mathcal{A}$ is represented as an $|\mathcal{A}| \times \ell$ matrix, where each letter is encoded as a vector of all zeros, except in specific positions where there is a 1 (see Figure 1.2).

$$\text{ATC..GT} \xrightarrow{\text{OHE}} \begin{matrix} A \\ C \\ G \\ T \end{matrix} \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 1 \end{pmatrix}$$

Figure 1.2: One-hot encoding of a DNA sequence ($\mathcal{A} = \{A, C, G, T\}$) as a 4-row matrix.

1.2.1 Linear models, support vector machines and data representation

Linear models, such as the ridge regression and support vector machines (SVMs) belong to the simplest and most widely used machine learning models. We will go over their operations briefly in order to understand why these strategies are effective and identify some limitations.

- Ridge regression, a regularized version of linear regression, aims at predicting the output value $y \in \mathbb{R}$ for an input object $x \in \mathcal{X}$. To that end, it learns a linear prediction function $\hat{y}(x) = (\varphi^x)^T \boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^\ell$ contains some weights and $\boldsymbol{\varphi}^x \in \mathbb{R}^\ell$ is the feature vector of the object x according to a representation function $\varphi : \mathcal{X} \to \mathbb{R}^\ell$. For instance, one-hot encoding defines such a function on sequences. Ridge regression optimizes a loss function on the training set $(x_1, y_1), \ldots, (x_n, y_n) \in (\mathcal{X}, \mathbb{R})^n$ and finds the optimal $\boldsymbol{\beta}^*$:

$$\begin{aligned} \boldsymbol{\beta}^* &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^\ell} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \hat{y}\left(x_i\right)\right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \\ &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^\ell} \frac{1}{n} \|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{X}} \boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \,, \end{aligned} \tag{1.1}$$

  where $\boldsymbol{\varphi}^{\boldsymbol{X}} \in \mathbb{R}^{n \times \ell}$ such that $\boldsymbol{\varphi}^{\boldsymbol{X}}_{i,\cdot} = \boldsymbol{\varphi}^{x_i}$, and $\lambda \in \mathbb{R}^+$ is a regularization factor. This combination of a quadratic error and a L^2 penalization, has an analytical solution:

$$\boldsymbol{\beta}^* = \left((\boldsymbol{\varphi}^{\boldsymbol{X}})^T \boldsymbol{\varphi}^{\boldsymbol{X}} + \lambda n \boldsymbol{I}_\ell\right)^{-1} (\boldsymbol{\varphi}^{\boldsymbol{X}})^T \boldsymbol{y} \,, \tag{1.2}$$

  with $\boldsymbol{I}_\ell \in \mathbb{R}^{\ell \times \ell}$ the identity matrix (see Chapter 3 Section 3.1 for details).

  In (1.2), the quadratic error is an empirical risk minimization: it measures the difference between the prediction $\hat{y}(x)$ and the true output $\boldsymbol{y}$. The L^2 penalty, relying on the squared euclidean norm of $\boldsymbol{\beta}$, encourages the learned weights $\boldsymbol{\beta}$ to be small in magnitude, which reduces the generalization error, *i.e.* the error obtained on an input x' which does not belong to the training dataset. But other strategies do exist: The Lasso (Tibshirani, 1996) relies on the L^1 regularization $\|\boldsymbol{\beta}\|_1 = \sum_j |\boldsymbol{\beta}_j|$, resulting in sparse weights, with a few non-zero elements. The Elastic-net (Zou & Hastie, 2005) combines both L^1 and L^2 penalties.

- SVMs (Cortes & Vapnik, 1995) have historically been a particularly successful class of models for supervised learning, applied in a variety of domains such as handwriting recognition, face detection or text categorization. In computational biology, they have been used notably to classify biological sequences (Rätsch et al., 2006), to detect protein remote homologies, or to predict the function of a protein (Schölkopf et al., 2004).

  For the sake of simplicity and following Lin et al. (2007) we will focus on a binary classification task, where each input x is represented as a feature vector $\boldsymbol{\varphi}^x \in \mathbb{R}^\ell$ and associated to an output $y = \pm 1$, such that a linear classifier takes the sign of a function $\hat{y}(x) = (\boldsymbol{\varphi}^x)^T \boldsymbol{\beta} + b$. The hyperplane $\hat{y}(x) = 0$ then defines a decision

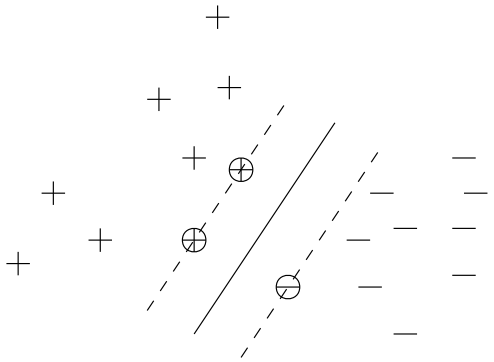


Figure 1.3: The optimal hyperplane separates positive and negative training points with the maximal margin. Its position is completely determined by the support vectors — the circled points achieving the minimal distance.

boundary in the feature space $\mathbb{R}^\ell$, and the parameters $\boldsymbol{\beta}$ and b are determined by running a learning procedure on a training set $(x_1, y_1), \ldots, (x_n, y_n) \in (\mathcal{X}, \mathcal{Y})^n$.

When the dataset is linearly separable, that is when there exists such a function $\hat{y}$ whose sign perfectly matches the classes of all training examples, the SVM chooses the one that maximizes the margin: it selects the function that maximizes the distance between the decision boundary $\hat{y}(x) = 0$ and the closest example, as mathematically described in (1.3) with a normalized dataset.

$$\begin{aligned} &\min \frac{1}{2}\|\boldsymbol{\beta}\|_2 \quad \left(\text{Geometrically, the margin is } \frac{2}{\|\boldsymbol{\beta}\|_2}\right) \\ &\text{s.t. } \forall i \in [n]\,,\ y_i((\boldsymbol{\varphi}^{x_i})^T\boldsymbol{\beta} + b) \geq 1 \end{aligned} \tag{1.3}$$

When the training examples are not linearly separable, some mistakes are allowed in (1.3) using an additional parameter that controls the compromise between large margins and small mistakes.

The optimization problem is not relevant to this thesis, however it should be noted that SVMs get their name from the fact that the solution for the choice of $\boldsymbol{\beta}$ and b depends only on the subset of points x_i that achieve the minimum distance: the *support vectors*, as described in Figure 1.3. The SVM's predictions can then be explained by a limited subset of x_i, which lends these models some sense of interpretability.

The function φ, which turns a data point x into a vector in $\mathbb{R}^\ell$, is thus a critical component of the models. The first question to address before applying any model is the choice of this representation function. One-hot encoding gives us a solution applicable to any biological sequences dataset, and kernel methods, developed in the next section, will allow us to extend such a mapping to other data types. Moreover, this mapping suffers a limitation: it is agnostic with regard to the biological problem, as it does not depend on the learning task. In Subsection 1.2.3, we will see that more complex machine learning methods can tackle this issue.

1.2.2 Kernel methods for biological sequences

Kernel methods provide a different paradigm to this representation issue: data are no longer represented individually, but rather through a set of pairwise comparisons. That is, instead of using a mapping function $\varphi : \mathcal{X} \to \mathbb{R}^\ell$, they rely on a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The training dataset is now represented as a $n \times n$ matrix of pairwise comparisons $k_{i,j} = k(x_i, x_j)$: the Gram matrix. Such a kernel function must meet some requirements in order to be usable in machine learning models, as described in Definition 1.2.1.

Definition 1.2.1 (Positive definite kernel)**.**
A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a positive definite kernel *if and only if it is:*

- *symmetric:* $\forall (x, x') \in \mathcal{X}^2\,,\ k(x, x') = k(x', x)$
- *positive definite:* $\forall n > 0\,, \forall (x_1, \ldots, x_n) \in \mathbb{R}^n$ *and* $\forall (c_1, \ldots, c_n) \in \mathbb{R}^n$ *:*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

A fundamental property of the kernels is given by Aronszajn's theorem:

Theorem 1.2.1 (Aronszajn, 1950)**.**
k is a positive definite kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping $\varphi : \mathcal{X} \to \mathcal{H}$ such that:

$$\forall (x, x') \in \mathcal{X}^2\,,\ k(x, x') = \langle \boldsymbol{\varphi}^x, \boldsymbol{\varphi}^{x'} \rangle_{\mathcal{H}},$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ in the dot product in $\mathcal{H}$.

This theorem states that using a kernel function *implicitly amounts to mapping the objects $x \in \mathcal{X}$ to a representation $\boldsymbol{\varphi}^x$* in a *feature space*. But there is a significant difference with the representation discussed in subsection 1.2.1: we can not necessarily access it. Apart from the fact that having an analytical expression for this mapping can be complicated, $\mathcal{H}$ is often an infinite dimensional space, which makes any computer storage of $\boldsymbol{\varphi}^x$ impossible.

While this might appear to be a problem at first glance, one has to remember that a wide class of machine learning methods access the input data only through pairwise dot products. This is known as the kernel trick: any algorithm for vectorial data that can be expressed only in terms of dot products between vectors can be performed implicitly in the feature space associated with any kernel, by replacing each dot product by a kernel evaluation. While this trivial statement has many important applications, the one that interests us here is the fact that it allows us to apply algorithms to non-vectorial data, such as biological sequences (Schölkopf et al., 2004, Section 1.2).

Using various kernels allows for applying the same algorithms to different representations of those data, and consequently to choose a relevant representation for the given prediction task.

For instance, a standard kernel is the Gaussian kernel:

$$\forall (x, x') \in \mathcal{X}^2\,,\ k(x, x') = e^{-\frac{d(x,x')^2}{2\sigma^2}}\,, \tag{1.4}$$

defined for some bandwidth parameter σ and using a distance defined over the objects of $\mathcal{X}$. For biological sequences, such a distance between sequences can be defined as the euclidean distance between their one-hot encoding matrices, leading to a valid kernel (Figure 1.4).

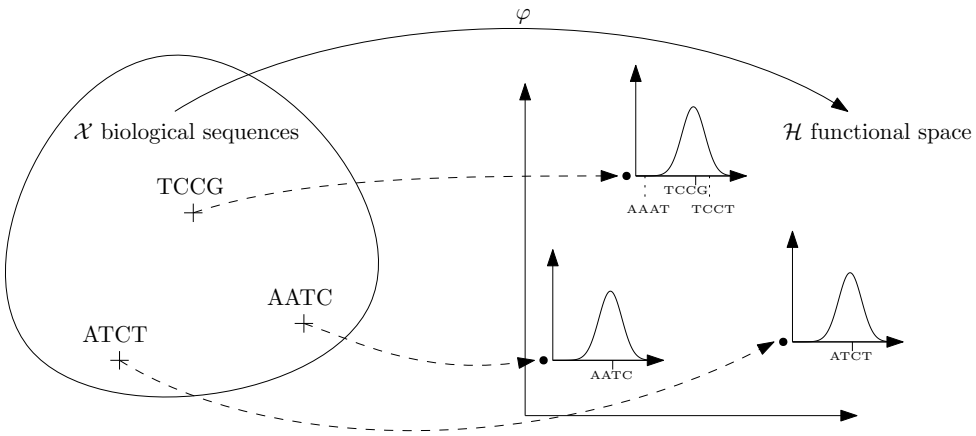


Figure 1.4: The Gaussian kernel on biological sequences $\mathcal{X}$ can be represented as an inner product after the sequences have been mapped to a functional space $\mathcal{H}$. In this feature space, each sequence x is represented as a Gaussian function over a matrix space, centered at x.

We are then left with two possibilities:

- Designing by hand a kernel associated with a representation that is well-suited for our data.

  There exist several famous kernels on biological sequences that have been designed, such as the Spectrum Kernel, introduced by Leslie et al. (2001) for application to protein classification. Based on the set of all contiguous sub-sequences of length k contained within a sequence over an alphabet $\mathcal{A}$, it defines a feature map $\varphi : \mathcal{X} \to \mathbb{R}^{|\mathcal{A}|^k}$:

  $$\boldsymbol{\varphi}^x = (\varphi^x_{\boldsymbol{u}})_{\boldsymbol{u} \in \mathcal{A}^k} \ , \tag{1.5}$$

  with $\varphi^x_{\boldsymbol{u}}$ is the number of times the k-mer $\boldsymbol{u}$ occurs in x. As the kernel function is then $k(x, x') = \langle \boldsymbol{\varphi}^x, \boldsymbol{\varphi}^{x'} \rangle$, two sequences will have a large kernel value if they share many of the same k-mers.

  This kernel, later improved by Leslie et al. (2004) to allow for some degree of mismatching in the k-mers comparisons for biological considerations, combined with SVMs formed the state-of-art approach for protein classification tasks.

- Relying on a machine learning algorithm to choose the best kernel for a given task from a given class of kernels. That is, learning a good representation that may take into account both the structure of the training points x_i and the learning objective, as discussed below.

1.2.3 Learning a relevant representation

Supervised learning models can often be decomposed as two-steps methods, as described in Figure 1.5:

- First, they represent the input data using a function φ. This representation can be straightforward: if x already is a numerical vector of features $\boldsymbol{x}$, then a trivial choice for this mapping is $\boldsymbol{\varphi}^x = \boldsymbol{x}$. However, for non-numerical data (such as biological sequences or graphs), this step is required in order to use the data in the predictor. Finally, even with numerical data, choosing a relevant representation can significantly improve the overall performance of the method. When dealing with kernel methods, this representation step might be implicit, with no $\boldsymbol{\varphi}^x$ being explicitly computed.
- Second, the prediction step is often performed using simple linear classifiers or regressions f — such as SVMs or ridge regression.

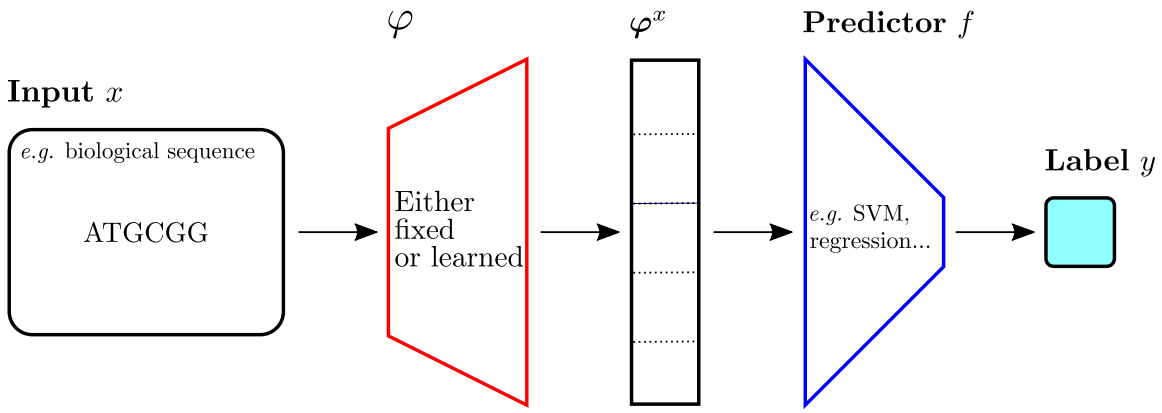


Figure 1.5: Diagram of a supervised learning method. While the predictor is always optimized using the training data, the representation φ can either be fixed beforehand (by choosing a specific kernel or by extracting some features from the data) or selected among a class of functions during the training step.

The representation can then be chosen manually, for example, via a feature engineering step, by selecting an appropriate kernel as stated in subsection 1.2.2, or by learning both the predictor and the mapping:

$$(\varphi^*, f^*) = \underset{(\varphi \in \mathcal{H}, f \in \mathcal{F})}{\arg\min} \left(n^{-1} \sum_{i=1}^{n} L\left(y_i, f \circ \varphi(x_i)\right) + \lambda \Omega(f \circ \varphi) \right), \tag{1.6}$$

where L is a loss function, f is the prediction function selected within the functional space $\mathcal{F}$ (for linear regressions, $\mathcal{F}$ is the set of linear functions), Ω is a measure of complexity used for penalization (whose impact is mitigated by λ) and $\mathcal{H}$ is a given class of mappings. This can be accomplished using methods such as multiple kernel learning (MKL) (Chapelle et al., 2002) and neural networks (NN), in which the output layer serves as a predictor while other hidden layers construct the representation. The many existing NN designs can then be viewed as ways to define different mapping classes $\mathcal{H}$.

1.2.3.1 Multiple kernel learning

Kernel methods allow for the implicit definition of a new data representation, on which will be performed a supervised learning task. Using the Spectrum Kernel (1.5) as an example, we can see that it may define several kernel functions, depending on the considered length k of the contiguous sub-sequences:

$$k_k(x, x') = \langle \boldsymbol{\varphi}_k^x \, , \boldsymbol{\varphi}_k^{x'} \rangle \, , \text{ where } \boldsymbol{\varphi}_k^x = (\varphi_{\boldsymbol{u}}^x)_{\boldsymbol{u} \in \mathcal{A}^k} \tag{1.7}$$

How can we choose the most relevant length k for a given task? One may also argue that from biological considerations the output is determined by the content of k-mers of various lengths rather than of k-mers of a specific size. As any convex combination of kernels defines a kernel, we can consider the convex combination of spectrum kernels with various lengths k:

$$k(x, x') = \sum_{k=1}^{M} w_k k_k(x, x') \, , \text{ with } w_k \geq 0 \text{ and } \sum_k w_k = 1 \tag{1.8}$$

This defines a class of kernel functions $\mathcal{K}$, containing all kernels that can be written according to (1.8) with any valid combination for the w_k. And implicitly, it defines a class of mapping $\mathcal{H}$: all the functions φ associated with a $k \in \mathcal{K}$. Finding the optimal convex combination, resulting in the optimal representation for the prediction task, is known as the *Multiple Kernel Learning* (MKL) problem. It is solved through joint optimization, as described in (1.6).

1.2.3.2 Convolutional neural networks for biological sequences

Over the last decade, convolutional neural networks (CNNs) — introduced by Lecun & Bengio (1995), have demonstrated outstanding performance in image-based predictions. Traditionally applied to analyze imaging data, they have also been successfully used for making predictions from biological sequences. In particular, they have brought significant improvements in regulatory genomics, *e.g.* to predict cell-type specific transcription factor binding, gene expression, chromatin accessibility or histone modifications from a DNA sequence (Zhou & Troyanskaya, 2015; Kelley et al., 2018; Avsec et al., 2021a,b). The convolution step allows for translation equivariance, and CNNs are therefore particularly suited to long sequences, whose relevant parts do not correlate with their positions.

Learning a CNN is typically achieved by minimizing the following objective:

$$\min_{g \in \mathcal{G}} \; n^{-1} \sum_{i=1}^{n} L(y_i, g(x_i)) + \lambda \Omega(g) \tag{1.9}$$

It jointly learns a representation and a predictor — $g = f \circ \varphi$ using notations from (1.6). In neural networks, the functions in $\mathcal{G}$ perform a sequence of linear and nonlinear operations.

For instance, DeepBind (Alipanahi et al., 2015) first maps a sequence x to a numerical vector using one-hot encoding, then applies a one-dimensional convolution with q convolution filters $\boldsymbol{Z} = (\boldsymbol{z}_i)_{i \leq q} \in \mathbb{R}^{|\mathcal{A}| \times k \times q}$, followed by a rectified linear unit and a pooling step, such as:

$$\mathcal{H} = \left\{ \varphi_{\boldsymbol{Z}} : \mathcal{X} \to \mathbb{R}^m \; : \; \exists \boldsymbol{Z} \in \mathbb{R}^{|\mathcal{A}| \times k \times q} \, , \; \varphi_{\boldsymbol{Z}}(x) = pool\left(ReLU\left(conv\left(ohe(x), \boldsymbol{Z}\right)\right)\right) \right\}$$

It then uses a linear output layer, such that $\mathcal{F}$ is the set of linear functions from $\varphi(\mathcal{X}) = \mathbb{R}^q$ to $\mathbb{R}$. Figure 1.6 provides an overview of this network.

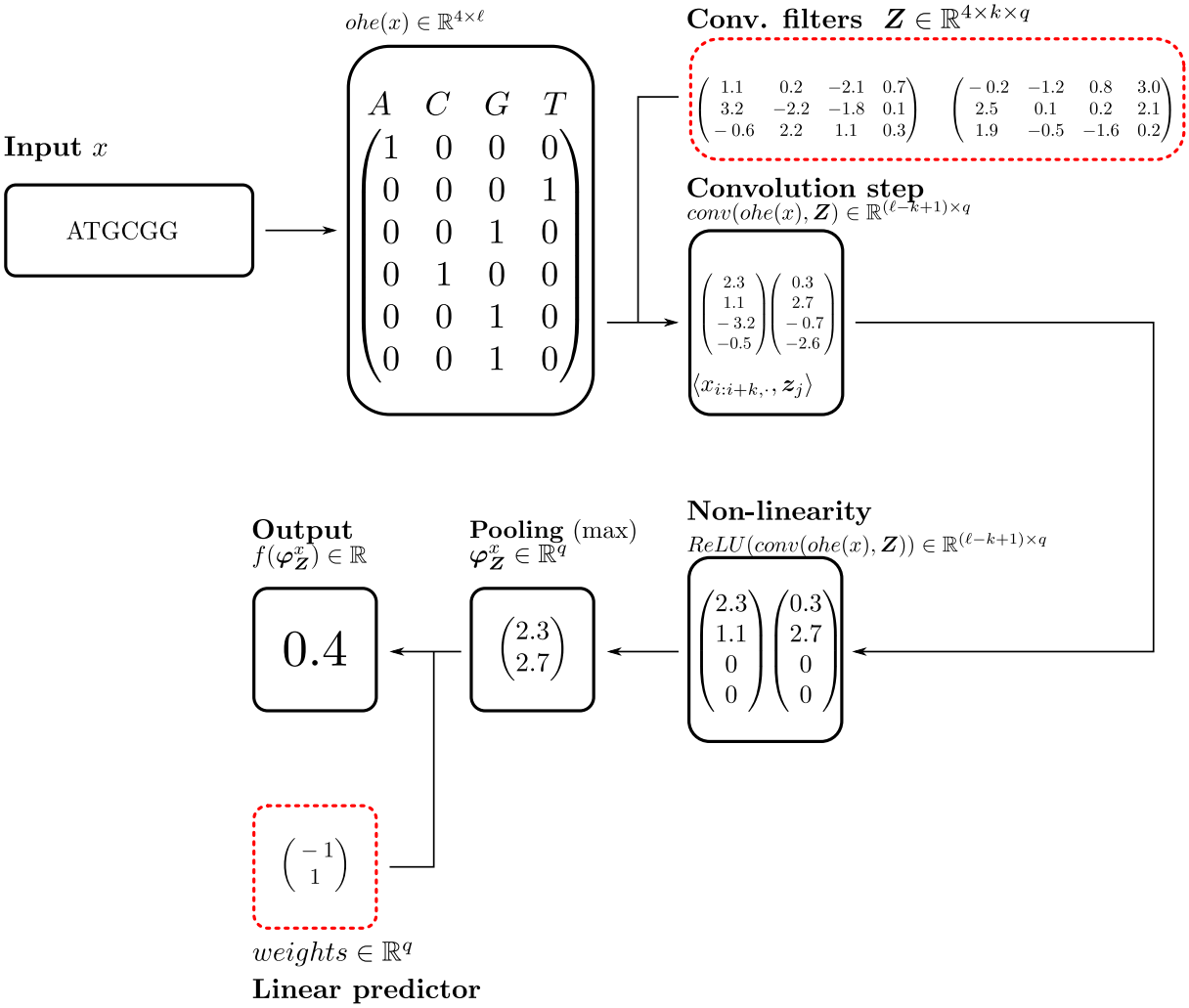


Figure 1.6: Diagram of DeepBind architecture. It starts by one-hot encoding the sequence of length $\ell = 6$, then it applies a convolution with $q = 2$ filters of length $k = 3$ followed by a non-linearity, a max pooling step and in the end a linear predictor. Both the linear weights of the predictor and the convolution filters (circled in red) are optimized during the training step.

A CNN is then a neural network that contains one or more convolutional layers. Such layers convolve their inputs with filters, that is they compare the filters with sliding windows of the inputs with same length.

• Convolutional Kernel Networks

As described in Section 1.2.2, kernels implicitly define data representations. Convolutional Kernel Networks (CKNs), introduced by Mairal et al. (2014) for images and extended to biological sequences by Chen et al. (2019a) make use of kernels whose induced mapping is related to the one obtained with CNNs.

Given two biological sequences x and x' of respective lengths ℓ and ℓ', let's consider the

following kernel

$$k(x, x') = \frac{1}{\ell\ell'} \sum_{i=1}^{\ell} \sum_{j=1}^{\ell'} k_0(\boldsymbol{u}_i^x, \boldsymbol{u}_j^{x'}) \,, \tag{1.10}$$

associated with a mapping φ, where $\boldsymbol{u}_i^x$ is the one-hot encoded k-mer of x starting at position i, and

$$k_0(\boldsymbol{u}, \boldsymbol{u}') = \|\boldsymbol{u}\|_2 \|\boldsymbol{u}'\|_2 \kappa \left(\left\langle \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|_2}, \frac{\boldsymbol{u}'}{\|\boldsymbol{u}'\|_2} \right\rangle \right) \,, \tag{1.11}$$

with $\kappa : v \to e^{\frac{1}{\omega^2}(v-1)}$ for some bandwidth parameter ω. This kernel k_0, similarly to the Gaussian kernel (to which it is actually quite similar), sends the k-mers into a functional space with infinite dimension using a mapping φ_0. This mapping $\varphi_0(\boldsymbol{u})$ contains a measure of similarity between the k-mer $\boldsymbol{u}$ and all possible matrices in $\mathbb{R}^{|\mathcal{A}| \times k}$, see Figure 1.4. These matrices can be interpreted as sequence motifs, as described below.

In addition to numerical barriers to accessing this representation, it does not take into account the learning objective. It then becomes interesting to approximate any representation in this space using its projection onto a finite-dimensional subspace $\mathcal{W}_{\boldsymbol{Z}}$ defined as the span of some anchor points $\boldsymbol{Z} = (\boldsymbol{z}_i)_{i \leq q} \in \mathbb{R}^{|\mathcal{A}| \times k \times q}$:

$$\mathcal{W}_{\boldsymbol{Z}} = Span\left(\varphi_0(\boldsymbol{z}_1), \ldots, \varphi_0(\boldsymbol{z}_q)\right) \,, \tag{1.12}$$

as illustrated in Figure 1.7.

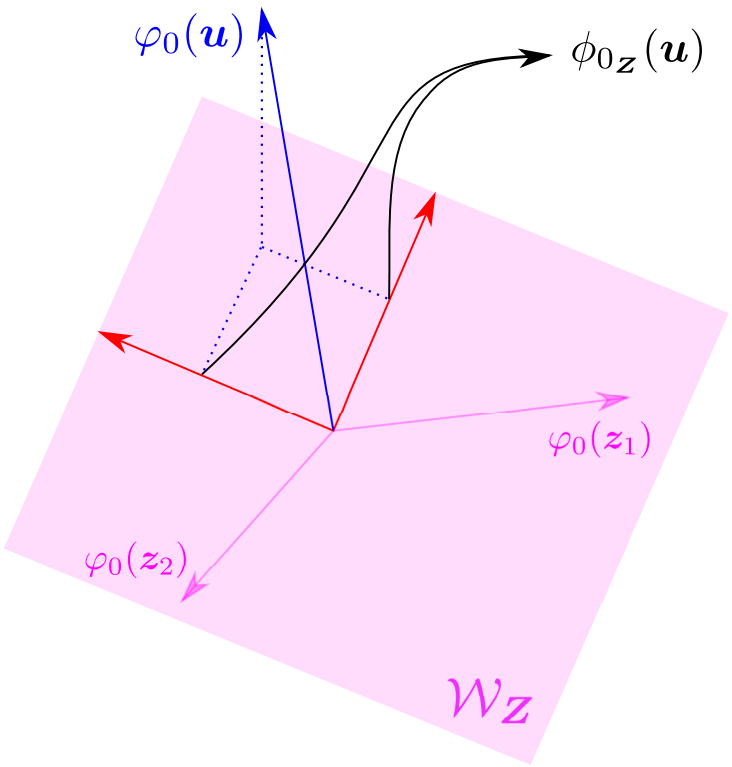


Figure 1.7: The use of anchor points $\boldsymbol{Z}$ allows the definition of a new mapping $\phi_{0_{\boldsymbol{Z}}}$, resulting in a approximation of the kernel k.

Chen et al. (2019a) introduce the following mapping, based on those anchor points:

$$\begin{aligned} \phi_{0_{\boldsymbol{Z}}} \, &: \mathbb{R}^{|\mathcal{A}|\times k} \to \mathbb{R}^q \\ &\boldsymbol{u} \mapsto \boldsymbol{K}_{\boldsymbol{ZZ}}^{-1/2} \boldsymbol{K}_{\boldsymbol{Z}}(\boldsymbol{u}) \,, \end{aligned} \tag{1.13}$$

with $\boldsymbol{K}_{\boldsymbol{ZZ}}$ the Gram matrix $[k_0(\boldsymbol{z}_i, \boldsymbol{z}_j)]_{i,j}$ and $\boldsymbol{K}_{\boldsymbol{Z}}(\boldsymbol{u}) = (k_0(\boldsymbol{z}_1, \boldsymbol{u}), \ldots, k_0(\boldsymbol{z}_q, \boldsymbol{u}))$.

They show that this mapping approximates the kernel k, introduced in (1.10):

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle \approx \langle \phi_{\boldsymbol{Z}}(x), \phi_{\boldsymbol{Z}}(x') \rangle_{\mathbb{R}^q} \text{ with } \phi_{\boldsymbol{Z}}(x) = \frac{1}{\ell} \sum_{i=1}^{\ell} \phi_{0_{\boldsymbol{Z}}}(\boldsymbol{u}_i^x) \,. \tag{1.14}$$

We then see that this representation is quite similar to the one learned by a standard CNN, as it includes a convolution step. The induced representation $\phi_{\boldsymbol{Z}}(x)$ of a sequence x is indeed described by some comparison of its k-mers (the sliding window) with points $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$, the filters, that are optimized in a supervised learning framework, similar to (1.9):

$$\min_{\substack{\boldsymbol{\beta} \in \mathbb{R}^q \\ \boldsymbol{Z} \in \mathbb{R}^{|\mathcal{A}|\times k \times q}}} \; n^{-1} \sum_{i=1}^{n} L(y_i, \boldsymbol{\beta}^\top \phi_{\boldsymbol{Z}}(x_i)) + \lambda \|\boldsymbol{\beta}\|_2^2 \,. \tag{1.15}$$

CKN demonstrated better performance than state-of-art CNNs, especially in a small- to medium-scale datasets, in particular thanks to induced regularization mechanisms (the $\boldsymbol{K}_{\boldsymbol{ZZ}}^{-1/2}$ in (1.13)).

• Sequence motifs: a promising variant for GWAS

Because CNNs and CKNs' first layer convolution filters are homogeneous to the input, they lend themselves easily to interpretation: as small picture patches for image inputs and as sequence motifs for one-hot encoded biological sequences. Such a sequence motif, a historical and basic element of regulatory genomics, can be thought of as a probabilistic k-mer. It can be represented mathematically as a position probability matrix (PPM) $\boldsymbol{z} \in \mathbb{R}^{|\mathcal{A}|\times k}$ such that

$$\forall j \leq k \,, \; \sum_{i=1}^{|\mathcal{A}|} \boldsymbol{z}_{i,j} = 1 \text{ and } \forall (i,j) \,, \; \boldsymbol{z}_{i,j} \geq 0 \,. \tag{1.16}$$

Although the trained filters are not constrained to match conditions (1.16) for optimization purposes, they can be projected onto the corresponding space at a later stage. An other possibility consists in projecting the filters over the position weight matrices (PWMs, containing log probabilities) vectorial space, and then the link between PPMs and PWMs is straightforward. The representation of a sequence induced by a convolutional network is thus dependent on the similarity between its k-mers and the motifs learnt during the training step.

In addition to this matrix representation, motifs can be displayed as sequence logos (Schneider & Stephens, 1990), in which the characters are stacked on top of each other for each position. The entire stack's height represents the information content of the position in

$$\begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} 0.11 & 0.32 & 0.01 & 0.99 & 0.67 & 0.77 & 0.00 & 0.00 \\ 0.00 & 0.38 & 0.00 & 0.00 & 0.25 & 0.01 & 0.60 & 0.00 \\ 0.01 & 0.22 & 0.13 & 0.00 & 0.05 & 0.21 & 0.40 & 0.00 \\ 0.88 & 0.08 & 0.86 & 0.01 & 0.03 & 0.01 & 0.00 & 1.00 \end{pmatrix}$$

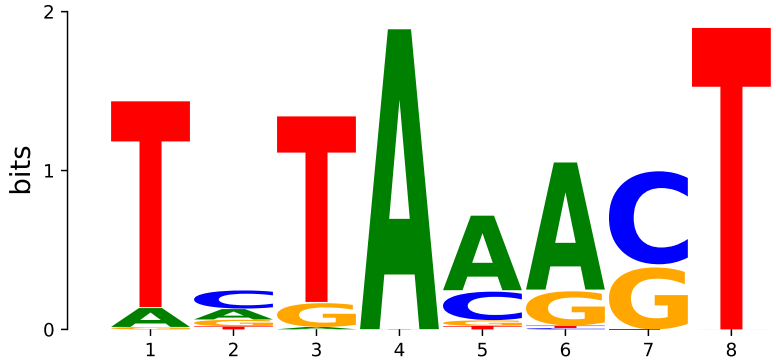


Figure 1.8: A sequence motif represented by its position probability matrix and corresponding sequence logo. The total height of the letters indicates the information content of the position (in bits), closely related to the Shannon entropy.

the sense of Shannon entropy, with a maximum of 2 bits. Each letter's height is related to its probability, as described in Figure 1.8.

As noted in Section 1.1, typical GWASs rely on SNPs to represent a sequence, whereas newer approaches use k-mers to address some of the SNP's intrinsic constraints. Sequence motifs can then be seen as a continuous generalizations of the k-mers, overcoming another limitation: the presence pattern of similar k-mers may be the best genetic determinant for a given phenotype rather than the pattern of an exact k-mer. For instance, synonymous substitutions, which are modifications of one base with no changes to the coded amino acid, may be silent.

In this case, the GWAS may output a huge list of k-mers, which is not very practical for the interpretation of the results. Moreover, in a small- to medium-scale dataset, there will be strong dilution effects with each of the many versions of the k-mers only present in a small number of individuals, resulting in a fairly low detection capacity. Sequence motifs, on the contrary, are unaffected by this issue because they are designed to account for such variations around a k-mer.

• Recurrent neural networks

Recurrent neural networks (RNNs), unlike traditional feed forward neural networks, which only process input data in a single pass, have feedback connections. They process the sequence one single letter at a time and output a value to the next artificial neuron. Thanks to their flexibility for different types of tasks — with biological sequences as inputs, they can either output a single numerical value (many-to-one) or a sequence (many-to-many) — they have been used in a variety of biological applications, such as protein secondary structure or protein-protein interaction prediction (Jurtz et al., 2017)). They are often used in combination with CNNs, resulting in architectures with one or

more convolutional layers followed by a recurrent layer.

Chen et al. (2019b) introduce a kernel for biological sequences, and show that the associated mapping is similar to the mapping induced by a RNN. While training a CKN results in selecting sequence motifs, the authors show that training this recurrent kernel network (RKN) allows for gaps in the motifs, motivated by genomics considerations. For instance, DeeperBind (Hassanzadeh & Wang, 2016) was proposed as an improved version of DeepBind, using a RNN network in addition to the CNN model.

In contrast to CNNs, which are unable to capture long-range interactions because their ability to extract local features in restricted by the filter size, RNNs are not constrained in this way. However, when dealing with long sequences, they may suffer from vanishing (or exploding) gradient problems (Pascanu et al., 2013). The gradients of the parameters become very small over the sequential process, making it difficult for the network to learn such dependencies in long sequences. Furthermore, as the length of the sequence increases, the limited capacity of their hidden state to store information becomes congested, causing a bottleneck issue when processing long sequences. Attention methods can help with both situations.

1.2.3.3 Attention networks and transformers

To get intuition on attention mechanisms (Vaswani et al., 2017), one can think about them as a way to automatically highlight the most relevant parts of the input data at each step of the processing. For instance, in a natural language processing framework, an attention mechanism might select the most relevant words in a sentence, allowing the network to focus on these words and better understand the sentence's content. This in in contrast to traditional neural networks, which evaluate all input data equally and do not allow for such a focus.

Attention mechanisms may be understood as a new strategy for weight regularization in a regression context. In the ridge penalization strategy described in Subsection 1.2.1, the weights $\boldsymbol{\beta}$ have an optimal analytical solution for (1.1): $\boldsymbol{\beta}^* = \left((\boldsymbol{\varphi}^{\boldsymbol{X}})^T\boldsymbol{\varphi}^{\boldsymbol{X}} + \lambda n \boldsymbol{I}_\ell\right)^{-1}(\boldsymbol{\varphi}^{\boldsymbol{X}})^T\boldsymbol{y}$, which leads to:

$$\begin{aligned}\hat{y}(x) &= \boldsymbol{y}^T\boldsymbol{\varphi}^{\boldsymbol{X}}\left((\boldsymbol{\varphi}^{\boldsymbol{X}})^T\boldsymbol{\varphi}^{\boldsymbol{X}} + \lambda n \boldsymbol{I}_\ell\right)^{-1}\boldsymbol{\varphi}^{x} \\ &= \sum_{i=1}^{n} \alpha_{\boldsymbol{X}}(x_i, x) y_i\,, \text{ for some function } \alpha_{\boldsymbol{X}}\,.\end{aligned}$$

The output $\hat{y}(x)$ is computed as a weighted sum of the y_i, where the corresponding weights depend on x and x_i. The function $\alpha_{\boldsymbol{X}}$ then encodes the relevance of x_i to predict for x and is specific to the ridge regression. This method is generalized by attention mechanisms, and an attention function may be defined as a mapping between a query $\boldsymbol{q}$ and a set of key-value pairs $(\boldsymbol{K}, \boldsymbol{V})$ to an output:

$$A(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V}) = \sum_{i=1}^{n} \alpha(\boldsymbol{k}_i, \boldsymbol{q})\boldsymbol{v}_i\,. \tag{1.17}$$

In the regression example, x is the query, the training points x_i are the keys, and the labels y_i are the values. In more complex networks, all $(\boldsymbol{q}, \boldsymbol{K}, \boldsymbol{V})$ may depend on trainable

parameters and/or on x, while α is fixed (Chaudhari et al. (2021) provide a summary of frequently used functions).

An example of implementation is TBiNet (Park et al., 2020). This network is used for TF-DNA binding prediction task, and its design is essentially a CNN followed by an attention layer and then a RNN, see Figure 1.9.

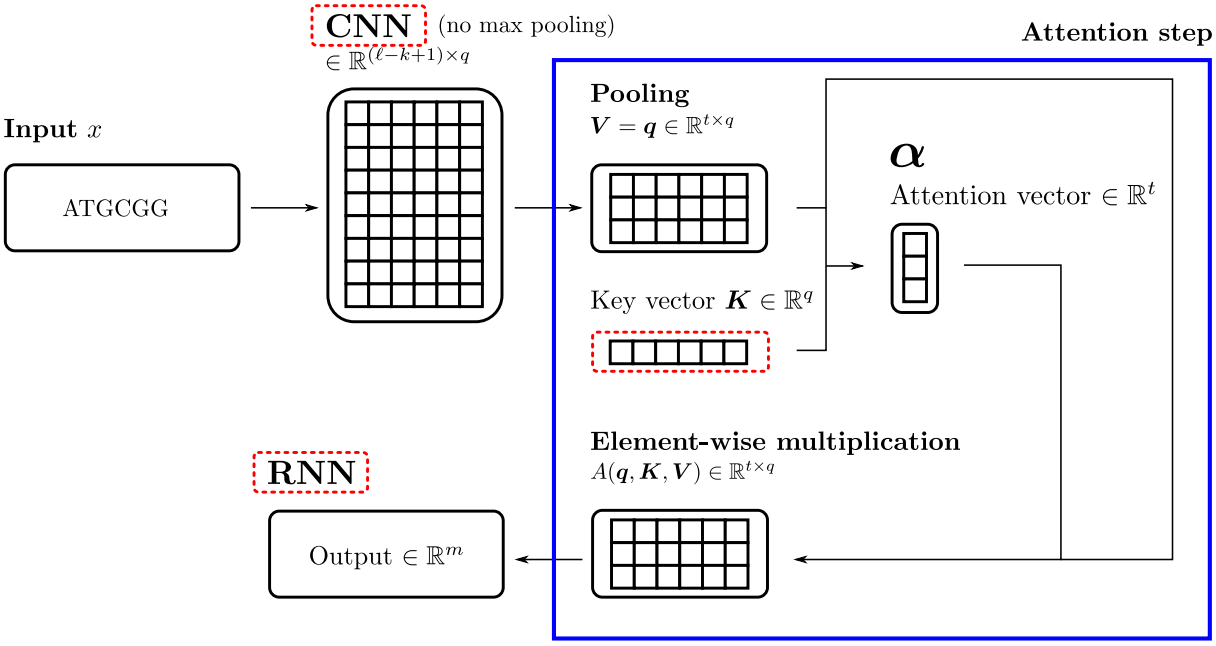


Figure 1.9: Diagram of TBiNet architecture. An input sequence is first passed into a CNN. Then, the obtained matrix serves both as value $\boldsymbol{V}$ and query $\boldsymbol{q}$ for an attention layer, while the key $\boldsymbol{K}$ parameters are determined throughout the learning process. The output is given by a final RNN. The attention layer is circled in blue, while the red boxes correspond to steps with trained parameters.

Thanks to the CNN (see Subsection 1.2.3.2), this network identifies sequence motifs that are used to compute the representations of any sequences. Moreover, for a given data point x, it highlights the most relevant part of x for the prediction: the ones with high attention scores in the activation vector (Park et al., 2020, Figure 6). Therefore, it provides new insights on the data representation.

By overcoming both the vanishing gradient problem and the bottleneck caused by the limited capacity of RNN hidden states, architectures based on attention mechanisms, like the Transformer (Vaswani et al., 2017), have advanced to the forefront of many natural language processing (Galassi et al., 2021) and computer vision (Khan et al., 2022) tasks. For biological sequences, DNABERT (Ji et al., 2021) can achieve state-of-the-art performance on prediction of promoters, splice sites and transcription factor binding sites.

In a supervised learning context, machine learning models learn a model from a dataset to make accurate predictions for unseen data points. Even when applied to biological sequences, these models are almost exclusively evaluated on the basis of their predictive performance. But some models give biological interpretations on why they perform well, and it draws a link with GWASs that aim to significantly associated variants for

a phenotype. Apart from that, interpreting machine learning models allows for model improvements and becomes a prerequisite for many real-world problematics. This is why it is becoming an increasingly important field of research.

1.3 Why are explanations important

Machine learning models become more sophisticated as they get better and better, requiring more training parameters, as illustrated in Figure 1.10. For instance, in natural language processing tasks, GPT-3 (Brown et al., 2020) has 175 billion parameters, while its predecessor GPT-2 (Radford et al., 2019) only had 1.5 billion, already 10 times more than its own predecessor GPT (Radford et al., 2018). In a more comprehensive way, (Bernstein et al., 2021, Figure 1) illustrates a similar trend: the number of parameters in recent landmark neural networks tends to grow exponentially.

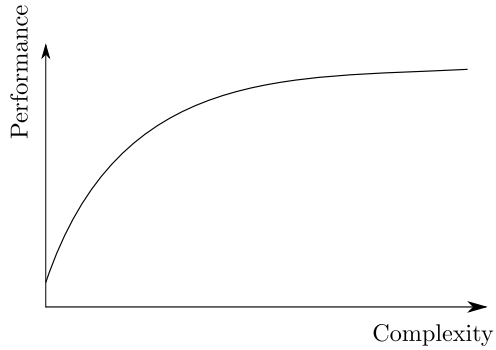


Figure 1.10: The complexity of a machine learning model and the number of parameters to be trained often increases faster than the performance.

It is necessary to define interpretability, because it is a large and poorly defined notion on its own. Kim et al. (2016) provides the following definition: "a method is interpretable if a user can correctly and efficiently predict the method's result". The greater a machine learning model's interpretability, the easier it is to understand why particular predictions or decisions were made. In this framework, we can rely on the definition given by Miller (2019) for explainable AI as: "an explanatory agent revealing underlying causes to its or another agent's decision making." It includes a broad set of methods, from the design of the algorithm to various visualization tools. But why not just trust a machine learning model and ignore why it made a certain prediction if it performs well?

For instance, according to DeGrave et al. (2021), explainable AI should be regarded as a requirement for clinical deployment of machine learning models in healthcare. The authors indeed evaluate the trustworthiness of 10 deep learning models applied to detect COVID-19 from chest radiographs. They first highlight the existence of an important gap between the performance on internal and on external test sets: the different models may appear accurate, but fail when applied to radiographs from new hospitals. Second, they use explainable AI techniques and reveal that the models' predictions heavily rely on radiographs areas that are medically irrelevant, such as some laterality markers or patient positioning.

To that end, they construct two datasets, both containing COVID-19 positive and negative radiographs. Dataset I contains COVID-19 positive radiographs from the GitHub-COVID repository, which includes images from various hospitals (with various image acquisition systems) and from publications, while the negative ones all come from a single hospital. As a result, the methods for obtaining radiographs are not uniform in this dataset. On the contrary, all the radiographs in dataset II (positive and negative) are issued from the same hospital system, see (DeGrave et al., 2021, Figure 1).

They train the models on dataset I, and compare the predictive performance on held-out radiographs from this same dataset (internal test set) to performance on radiographs from dataset II (external). Although the models attain high performance on the internal test data, *half of this performance is lost* when testing on dataset II, which suggests that the algorithms learned some spurious features. Particularly, there exists in dataset I a correlation between the image acquisition system and the COVID status, therefore learning features corresponding to those differences will have a positive impact on the internal performance while not being leverageable on the external test. This hypothesis is confirmed by the implementation of two explainable AI techniques (DeGrave et al., 2021, Figure 2):

- Using saliency maps, that are images that highlight pixels that were important from the COVID status prediction from a given radiograph, indicates that regions of the radiographs located outside the lungs are important for the algorithms' predictions
- Implementing and training neural networks that learn how to modify a radiograph with a given label such that the prediction associated with this modified image changes, reveals that spurious features, such as the positioning of a patient shoulders, impacts the prediction.

More surprisingly, the same level of degradation is observed when training the models on dataset II and comparing the performances on dataset I, suggesting that even when there are no correlations between the image acquisition system and the COVID-19 status, the models learn some spurious features. It reveals that this poor behavior persists even in a more ideal data collection environment, emphasizing the importance of deploying explainable AI tools alongside traditional prediction models.

In some cases, especially in a low-risk environment such as a movie recommendation system, a good predictive performance might be enough to use the method. But interpretability may be required:

- For safety issues, such as understanding why the machine learning system has predicted a given COVID-19 status for the patient,
- To understand and limit the impact of bias in the model, for instance by identifying patients for which the predictions are known to be poor,
- To learn from the model: if an algorithm can predict accurately a phenotype whose mechanisms are not yet known, understanding what features were used for the predictions can give insights on those mechanisms,
- To improve social acceptance, to assist in auditing and debugging...

1.4 Interpretability tools

Machine learning interpretability methods can be classified using a variety of criteria (Molnar, 2020). The ability of a model to explain its own predictions is referred to as **intrinsic** model interpretability. A model is considered to be intrinsically interpretable if it can provide a clear and understandable explanation of how it arrived at its predictions without the use of external tools. Linear regression and short decision trees are examples of such models. In contrast, **Post-hoc** model interpretability refers to strategies used to explain a model's predictions after it has been trained. Following (Novakovsky et al., 2022a), we shall concentrate on the latter type in this thesis.

On the one hand, **local** interpretation refers to methods for explaining the predictions of a model for a given data point (or for a small set of data). In other words, local interpretation provides an explanation of the model's prediction for a particular case. **Global** interpretation, on the other hand, refers to methods for explaining the model's general behavior. Such strategies can assist in identifying patterns and relationships in the data that the model has learned. In some cases, local interpretations can be aggregated to reveal a global understanding of the model's behaviors.

Finally, some methods are restricted to **specific** model classes, whilst other **model-agnostic** tools can be used regardless of the model type. Figure 1.11 depicts the relative positions of various interpretation methods in relation to these criteria.

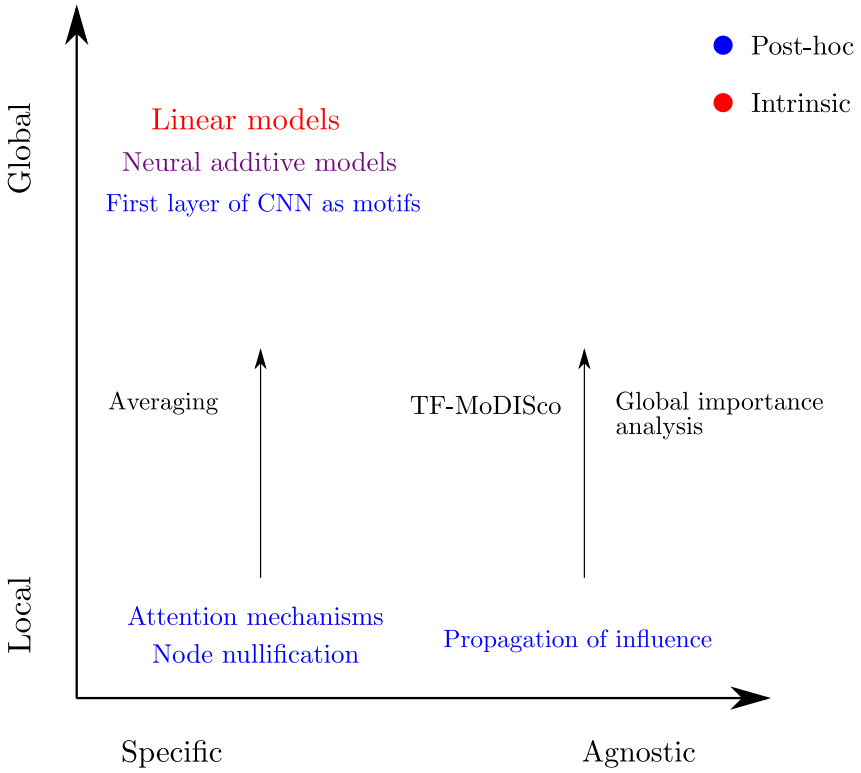


Figure 1.11: Classification of the main explainable machine learning methods.

1.4.1 Interpreting first-layer filters of CNNs as sequence motifs

CNNs and CKNs' first layer's convolution filters are homogeneous to biological sequences, as discussed in Subsection 1.2.3.2, and can thus be regarded as sequence motifs or probabilistic k-mers. Despite the fact that they do not match the requirements for PFMs given in (1.16) due to optimization considerations, they can be projected to produce PWMs or PFMs. It is worth noting that other methods for obtaining sequence motifs from first layer convolution filters exist. For instance, (Min et al., 2017) look for all possible subsequences of length k that yield convolutional activations above a given threshold, and average their one-hot encoded matrices to obtain a PFM.

This interpretation approach is global, as it provides information about the entire model rather than about a prediction for a given data point, and is specific to networks with a single convolutional layer. Although easy to implement, it also has some limitations, summarized in (Koo & Eddy, 2019):

- Deep CNNs tend to learn distributed representations of sequences motifs. They learn partial motifs, that are then assembled in deeper layers,
- The filters may learn slightly different versions of the same motif, leading to redundancies,
- As deep neural networks are over-parametrized by design, some filters learn relevant motifs and capture the signal, but the other filters do not have signal anymore to learn from, resulting in non-relevant motifs.

1.4.2 Visualizing importance of a neural network node using nullification

Some nodes of a neural network may be understood as interpretable features. But just because a given element has been learned, for instance a sequence motif corresponding to a CNN filter, does not mean that it is necessarily relevant to explain the network's behavior. The simplest approach to measure the contribution of a given filter is to nullify it, and observe the variations in prediction that ensue. Big variations indicate that this filter plays an important role for the model.

For a given input sequence, the influence of filter nullification can be measured and interpreted locally. The simplest method for forming a global interpretation is then to average the local ones.

But as deeper layers make previous layers filters dependent on each other, this strategy does not produce satisfying results due to redundancy. If several similar filters do exist with distributed importance, nullifying one of them will only result in a small impact on the final predictions. Moreover, some non-linearities may result in model saturation, as described in Figure 1.12.

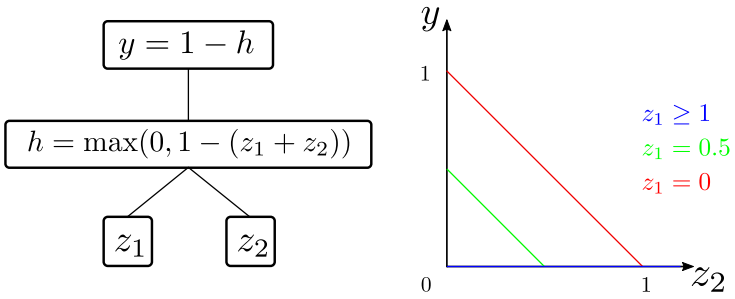


Figure 1.12: Nullifying filter z_1 has no impact on the prediction as long as the activation of z_2 is greater than one.

1.4.3 Obtaining importance from attention mechanisms

Attention mechanisms, as explained in Subsection 1.2.3.3, highlight the most relevant input features in a way dictated by the attention function and the various parameters learned, particularly the key $\boldsymbol{K}$. The attention vector (see Figure 1.9) then provides a direct estimate of the relevance of its input features. For instance, TBiNet computes attention scores along the regions of an input sequence, which serves as a proxy for their importance in the prediction. In contrast to Subsection 1.4.2, the received information no longer has a direct relationship to single nodes.

Again, the offered interpretation is local (it is specific to a given sequence), but can be averaged using several inputs to acquire more general information about the model as a whole, as seen in Figure **??**.

Finally, Serrano & Smith (2019) criticize the use of attention as a measure of the importance of features. Although the feature with the highest attention weight tends to have a higher impact on the prediction than the feature with the lower weight, relying on attention weights ranking to identify the set of features most important to the model's prediction often fails.

1.4.4 An agnostic set of methods: propagation of influence

Propagation of influence, or perturbation-based approach, entails making minor changes, or perturbations, to an input and analyzing how these affect the output of a machine-learning model. This helps in determining which parts of the input are most critical to the model's prediction. Such methods are model-agnostic since they can be applied regardless of the model's architecture.

These approaches are classified into two groups. On the one hand, the forward propagation of influence modifies the input and observes the changes in the predictions. Backward propagation of influence, on the other hand, uses a backward pass to compute the gradient (or other similar measures) of the prediction with respect to the various components of the input.

The two following subsections will follow Novakovsky et al. (2022a) and Shrikumar et al. (2017), which provide an in-depth analysis of such approaches for biological sequences.

1.4.4.1 Forward propagation of influence

Forward perturbation strategies are widely used in computer vision, and the associated philosophy can be exemplified using DeGrave et al. (2021) (introduced in Section 1.3). Among the different methods they apply to explain radiograph-based COVID status predictions, the authors have trained neural networks to slightly alter radiographs in order to inverse the output prediction. If changing a pixel (or a set of pixels) affects the prediction, then it may correspond to a key feature. As a result, the authors discovered that the various models relied on non-medical features for predicting the COVID status, including markers related to image acquisition systems.

Altering letters in biological sequences is similar to modifying pixels in a computer vision framework. This approach is termed in silico mutagenesis (ISM), and was first introduced by Zhou & Troyanskaya (2015). In this paper, the authors train a neural network to predict chromatin features from DNA sequences. To discover informative sequence features within an input sequence, they compute for each possible mutation the effect on the prediction using *log* fold change:

$$log_2\left(\frac{P_0}{1-P_0}\right) - log_2\left(\frac{P_1}{1-P_1}\right), \tag{1.18}$$

where P_0 and P_1 represent the binding probabilities predicted for the original and mutated sequence. The results can be visualized using an attribution map (see Figure 1.13.), similarly to saliency maps used in computer vision.

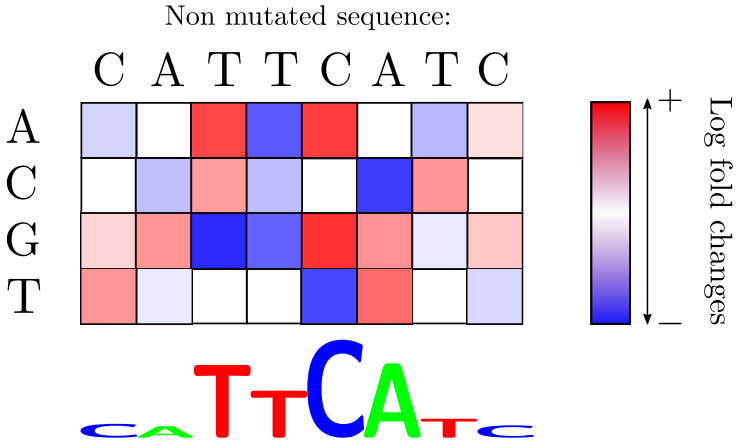


Figure 1.13: Attribution map obtained by measuring the impact of every possible single base mutation from the original input sequence CATTCATC with length 8 (3×8 in total). This result can also be represented as a logo, the information content of a position being related to the max or average *log* fold change.

The first drawback that comes to mind is the high number of forward propagations that must be computed if this method is to be applied to a big number n of sequences of length ℓ: $3 \times n \times \ell$ in total. For the sake of completeness, we can note that the search for efficient ISM algorithms is an active field of research, with notably fastISM (Nair et al., 2022) and Yuzu (Schreiber et al., 2022) that are both algorithms that speed up ISM by taking advantage of specific architectures of neural networks. Furthermore, ISM may suffer from

model saturation, as seen in Figure 1.12: changing a base that activates already saturated nodes would result in no prediction change.

Finally, rather than being limited to single base mutations, this approach has been extended to the k-mer scale to identify relevant sequence motifs with related attribution scores (Novakovsky et al., 2022a).

1.4.4.2 Backward propagation of influence

Backward methods propagate the signal from an output backwards through the machine learning model to the input in a single pass, considerably reducing the number of passes required. They rely on evaluating the derivative (or an approximation) of the model's prediction function at the input, to estimate the influence of changes in the input to the output, thus approaching ISM: instead of measuring precisely the impact of a given base change, it is estimated thanks to the gradients.

They were first developed for computer vision, and include:

- Deconvolutional Network (deconvnet, Zeiler & Fergus (2013)): a network using the same components as a CNN (filters, convolutional step, pooling) but in a reverse order. Instead of mapping an image to features, it maps the features learned by another CNN back to an image
- Directly using the gradient of the output with respect to the pixels of the image (Simonyan et al., 2014), or combining this strategy with deconvnet to generalize the latter to more architectures (Guided Backpropagation, Springenberg et al. (2015)).

When applied on biological sequences, these methods can be used to produce attribution maps, similar to Figure 1.13.

These different methods can be seen as different ways of handling non-linearities during the backward pass (Shrikumar et al., 2017), but all suffer from the saturation problem (Figure 1.12, as well as instabilities due to discontinuous gradients produced by some non-linearities). DeepLIFT (Shrikumar et al., 2017) solves both issues: it relies on introducing a reference output and on back propagating discrete gradients (using a modified version of backpropagation using chain rule like computation rules) instead of infinitesimal differences. But doing so, it breaks the Implementation Invariance: applying DeepLIFT to two networks with the same prediction functions obtained using two different architectures may lead to two different results. Approaches based on integrating the gradients along a line from the input to a reference input — Integrated Gradients (Sundararajan et al., 2017) — respect the Implementation Invariance rule, are not prone to saturation, but may suffer from instabilities. Those methods are summarized in Table 1.2.

Finally, attribution methods (both forward and backward) are model-agnostic, and deliver local interpretations. Nevertheless, some approaches like TF-MoDISco (Shrikumar et al., 2018) and Global Importance Analysis (GIA) (Koo et al., 2021) aggregate those local results to derive global interpretation about the models, while once again extracting sequence motifs:

	Saturation	Implementation Invariance	Instabilities
Deconvnet	✗	✓	✗
Gradient	✗	✓	✗
Guided backpropagation	✗	✓	✗
DeepLIFT	✓	✗	✓
Integrated Gradients	✓	✓	✗

Table 1.2: Different approaches relying on backpropagation of influence and their limitations. Red cross indicates method limitation.

- TD-MoDISco takes as input a set of importance scores on genomic sequences, for instance obtained with DeepLIFT, and starts by identifying high-importance windows within the sequences, termed *seqlets*. Then, it clusters those seqlets into sequence motifs.
- GIA starts by measuring importance scores for all possible k-mers of a given length: the more the model's prediction varies with the presence or absence of a given k-mer, the higher score it gets. Then, GIA creates an alignment of the top scoring k-mers and average them to produce sequence motifs.

1.4.5 Using prior knowledge to derive transparent models

In order to be thorough, we can cite recent works that aim to design new architectures for neural networks, based on biological knowledge. It may be done at different levels:

- When using a standard architecture with a first convolutional layer, one can initialize the first layer filters according to known motifs of interest — DanQ (Quang & Xie, 2016),
- Designing the structure and initializing the parameters of the model using knowledge about cell's subsystems — DCell (Ma et al., 2018),
- Relying on neural additive models (Agarwal et al., 2021), which rely on linear combination of smaller neural networks, allowing to take advantage of the inherent interpretability of linear models combined with the expressivity of neural networks. For instance, ExplaiNN (Novakovsky et al., 2022b) combines several CNNs, each containing only one convolutional filter.

These methods are then able to identify various biological features to provide global interpretation of their predictions. These features may include sequence motifs, but they may also include other diverse genetic variants, such as pathways and gene interactions, depending on the chosen architecture. Such features would be more difficult to highlight with traditional neural networks.

1.4.6 Limitations of interpretability

Explainable machine learning covers a heterogeneous set of methods that seek to provide insights into why a model takes certain decisions rather than others. The approach used is therefore determined by the architecture of the model, the level of explanation needed — is it to understand a specific prediction or the overall functioning of the model? — as well as the type of biological features that the user considers as interesting.

A certain amount of explainability can always be reached. However, there are some criteria for determining the level of relevance of a given explanation:

- Is the explanation stable? Let's consider a model trained over a dataset D, and an explanation f obtained using one of the aforementioned methods (either local for a given x or global). Would the same explanation be obtained if we add or remove a data point to D? If the machine learning model is robust to this modification (the prediction function remains almost unchanged) but a different explanation f' is given, it calls into question the relevance of f.
- Is an explanation complete? Does it cover all of the key aspects of the decision-making process? For instance, interpreting the first-layer filters of a CNN as sequence motifs does not provide information about the interactions between those motifs.

- Is it accurate? Does the given explanation play a significant role in the decision-making process?
- Is it actionable? How can this information be used?

The different interpretation methods make it possible to highlight biological features that seem to be relevant to explain a given phenotype. The class of features is wide and depends both on the architecture of the model and on the chosen interpretation method. In this sense, it enables the discovery of new genetic variants, which may be subsequently tested in the same way as GWASs do. Furthermore, computing the statistical significance of an explanatory element answers at least partially the outlined criteria, depending on the chosen null hypothesis. While this statistical significance has its own limitations (Wasserstein & Lazar, 2016), it often provides an intuitive scale for identifying relevant features and has only received little attention to our knowledge in the context of neural networks.

Whether using SNPs, k-mers, sequence motifs or any other feature, we see that they belong to very rich classes. Then, there exists a very large number of variants to test, and when the class is continuous this number becomes infinite. If we do not pay attention to this multiplicity, the results may be biased. The goal of the next chapter is to propose an overview of the different approaches that can be applied in such a context.

CHAPTER 2

From multiple testing to conditional inference, different strategies for valid inference procedures on high-dimensional data

Regardless of the type of the genetic variants considered in a GWAS, there are always a huge number of existing different ones. For instance, the number of interesting SNPs is above 10^6 in the human genome, and there exists an infinite number of sequence motifs, as they are matrices living in a continuous space. Testing these variants poses a significant replicability challenge due to the high-dimensional statistics and multiple testing context (Benjamini, 2020). The purpose of this chapter is therefore to explain and give intuition about this issue, and to propose an overview of the different strategies to overcome it. To that end, we will develop an example throughout the following sections, allowing us to illustrate the problem and the different methods.

2.1 Uncovering gene-phenotype associations: a case study

Let's say that we are interested in discovering the genes whose expression levels are associated with cell growth rate, denoted y. This rate, often expressed as a population doubling time, is then a continuous phenotype: $y \in \mathbb{R}$.

2.1.1 Association between a given gene and the phenotype

In this subsection, let's consider a single given gene g. We are interested in the following null hypothesis:

$\mathbb{H}_0$: "There is no association between the growth rate y and the expression level of g".

To this end, we studied n different cells $\{c_1, \dots, c_n\}$, for which we measured the growth rates $\boldsymbol{y} \in \mathbb{R}^n$, such that $\boldsymbol{y}_i = y_i$ is the growth rate of cell c_i. The expression levels for gene g in these cells are summarized in $\boldsymbol{g} = (g_i)_{i \leq n}$.

The null hypothesis described above is quite vague, as several types of association can be considered. To precise it, we suppose that y is linearly associated with g:

$$y = \beta_1 g + \beta_2 + \epsilon\,, \tag{2.1}$$

where the $\beta_i \in \mathbb{R}$, and $\epsilon \in \mathbb{R}$ is normally distributed $\epsilon \sim \mathcal{N}(0, \sigma^2)$, with $\sigma \in \mathbb{R}$. The noise ϵ is assumed to be homoscedastic.

The null hypothesis can now be formulated as:

$$\mathbb{H}_0 \ : \ \text{“}\beta_1 = 0\text{”}\,. \tag{2.2}$$

In other words, $\mathbb{H}_0$ states that there is no relationship between the growth rate and the expression level of gene g.

The true values $\boldsymbol{\beta} = (\beta_1, \beta_2)$ are unknown, so we can estimate them using the coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_j)_{j \leq 1}$ obtained by minimizing the least-squares problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}' \in \mathbb{R}^q}{\arg\min} \left(\frac{1}{n} \|\boldsymbol{y} - \boldsymbol{G}^T \boldsymbol{\beta}'\|_2^2 \right)\,, \tag{2.3}$$

where $\boldsymbol{G} \in \mathbb{R}^{2 \times n}$, such that $\boldsymbol{G}_{1,\cdot} = \boldsymbol{g}^T$ and $\boldsymbol{G}_{2,\cdot} = \mathbf{1}_n^T$ is the vector of all ones. Under a full-rank assumption, this equation admits an analytical solution:

$$\hat{\boldsymbol{\beta}}^T = \left(\boldsymbol{G}\boldsymbol{G}^T\right)^{-1} \boldsymbol{G}\boldsymbol{y}\,. \tag{2.4}$$

To test the null hypothesis (2.2), we rely on the following test statistic V:

$$V := \frac{|\hat{\beta}_1|}{\mathrm{se}(\hat{\beta}_1)}\,, \tag{2.5}$$

where $se(\hat{\beta}_1)$ is the standard error of estimation for $\hat{\beta}_1$. The test we derive consists in rejecting the null hypothesis if $V \geq t$, where t is some threshold chosen such that the probability to wrongly reject $\mathbb{H}_0$ is lower than a given risk level α.

Under the assumption that the expression level of g across the cells is distributed according to a Gaussian model, and that $\mathbb{H}_0$ (2.2) holds, we have the following result (Giraud, 2021, Chapter 10):

$$V \sim \ |\mathcal{T}(n-2)|\,, \tag{2.6}$$

with $\mathcal{T}(n-2)$ the Student's distribution with $n-2$ degrees of freedom, and we denote $F_{|\mathcal{T}(n-2)|}$ its cumulative distribution function (CDF). We now define the function p such that:

$$\begin{aligned} p\ &: \mathbb{R}^+ \to [0,1] \\ &\quad v \mapsto 1 - F_{|\mathcal{T}(n-2)|}(v)\,. \end{aligned} \tag{2.7}$$

It follows that if $\mathbb{H}_0$ holds, $p(V) \sim \mathcal{U}(0,1)$. Then, for a given $\alpha \in [0,1]$, and under the null hypothesis (2.2), we have:

$$\mathbb{P}(p(V) \leq \alpha) = \alpha\,. \tag{2.8}$$

This means that if we chose to reject the null hypothesis when $p(V)$ is lower than α, the probability of wrongly rejecting $\mathbb{H}_0$ is equal to α, and $p(V)$ consequently defines a correct p-value.

To illustrate this result, we use the following script to obtain the Q-Q plot in Figure 2.1, with $m = 1\ 000$ independent experiences under the null hypothesis and $n = 100$ samples per experience:

```
pvalues = c()
for (r in 1:m){
    # Generating a dataset under the null
    g <- rnorm(n)
    y <- rnorm(n)

    # Finding the optimal hat_beta
    model <- lm(y~g)

    # Recovering the pvalues
    pvalues <- c(pvalues, summary(model)$coefficients[2,4]
}
```

The resulting distribution fits perfectly with the uniform, which confirms the validity of the testing procedure described above.

2.1.2 What happens with multiple genes?

In this subsection, instead of studying the association for just one gene g, we collected the expression levels of q genes $\{g_1, \ldots, g_q\}$ for n samples, and we want to discover the genes that are significantly associated with the phenotype y. This dataset is summarized in Table 2.1.

We therefore generalize the framework introduced in (2.1) using a multiple linear model:

$$y = \beta_1 g_1 + \cdots + \beta_q g_q + \beta_{q+1} + \epsilon. \tag{2.9}$$

From this model, we define a set of null hypotheses $(\mathbb{H}_{0,j})_{j \leq q}$ such that:

$$\mathbb{H}_{0,j}\ :\ \text{“}\beta_j = 0\text{”}\,. \tag{2.10}$$

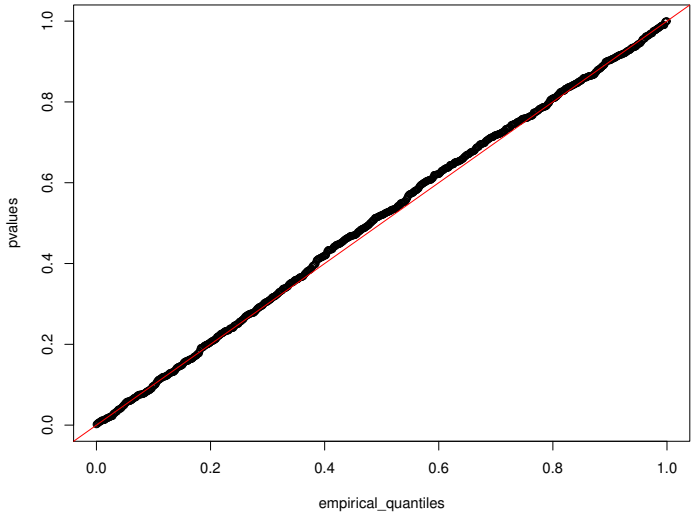


Figure 2.1: Comparing the distribution of p-values, obtained under the null for the association of gene g with the phenotype, to a uniform distribution using a Q-Q plot.

	Cell c_1	Cell c_2	...	Cell c_n
Gene g_1	$g_{1,1}$	$g_{2,1}$		$g_{1,n}$
Gene g_2	$g_{2,1}$	$g_{2,2}$		$g_{2,n}$
Gene g_q	$g_{1,q}$	$g_{2,q}$		$g_{q,n}$
Growth rate	y_1	y_2	...	y_n

Table 2.1: Example dataset to be used throughout the chapter.

As above, the β_j are unknown and will be estimated using (2.4) with $\boldsymbol{G} \in R^{(q+1)\times n}$, still under a full-rank assumption for $\boldsymbol{G}\boldsymbol{G}^T$. The test statistics are now:

$$V_j := \frac{|\hat{\beta}_j|}{\text{se}(\hat{\beta}_j)} \,. \tag{2.11}$$

Under the Gaussian assumption and the null hypothesis $\mathbb{H}_{0,j}$, we have:

$$V_j \sim |\mathcal{T}(n-(q+1))| \,. \tag{2.12}$$

Finally, we can modify (2.7) and give a new definition of the function p, using:

$$\begin{aligned} p \,&: \mathbb{R}^+ \to [0,1] \\ &v \mapsto 1 - F_{|\mathcal{T}(n-(q+1))|}(v) \,. \end{aligned} \tag{2.13}$$

This leads to a valid definition for the p-value $p(V_j)$ corresponding to the null hypothesis $\mathbb{H}_{0,j}$. From now on, the p-value associated with $\mathbb{H}_{0,j}$ will simply be denoted p_j.

For a given gene g_j, rejecting $\mathbb{H}_{0,j}$ when p_j is lower than α results in a probability of wrongly rejecting $\mathbb{H}_0$ equals to α.

• Where the problems begin

In fact, while looking at the $(\hat{\beta}_j)_{j\leq q}$ coefficients, we realize that the gene g_s seems particularly associated with the phenotype. We indeed noticed that $|\hat{\beta}_s| = \max_{j\leq q}\left(|\hat{\beta}_j|\right)$. We are now interested in computing the p-value p_s for this specific gene. To that end, we rely on (2.11,2.13).

But let's repeat this experiment a large number $m = 1\ 000$ of times, always with data under the null hypotheses $(\mathbb{H}_{0_j})_{j\leq q}$, and observe the empirical distribution of the p-values p_s. In each run, s is chosen such that $|\hat{\beta}_s| = \max_{j\leq q}(|\hat{\beta}_j|)$. With $n = 1\ 000$ samples per dataset and $q = 100$ genes, we use the following code snippet:

```
pvalues_s <- c()
for (r in 1:m){
    # Generating a dataset under the null
    G <- matrix(rnorm(n*q), nrow=n)
    y <- rnorm(n)

    # Fitting the linear model
    model <- lm(y~G)

    # Recovering all hat_beta and pvalues
    hat_beta <- unname(summary(model)$coefficients[2:q,1])
    all_pvalues <- unname(summary(model)$coefficients[2:q,4])

    # Keeping only p_s, associated with the maximum of the |hat_beta|
    pvalues_s <- c(pvalues_s,
            all_pvalues[abs(hat_beta)==max(abs(hat_beta))]
}
```

The resulting Q-Q plot is given in Figure 2.2 and shows a significant decalibration: the empirical distribution is far from $\mathcal{U}(0,1)$. With this set of parameters, rejecting $\mathbb{H}_{0,s}$ when $p_s < 0.05$ leads to rejecting this null hypothesis for more than 90% of the experiments, while the expected proportion should be 5%.

And this is where post-selection inference and multiple testing problem appears. Indeed, the classical inferential scheme, in order to give valid results, assumes that the null hypotheses have been chosen without having used the data. This is the classic scheme, where one formulates a null hypothesis, collects some relevant data, and tests this hypothesis using the data.

But recently, the increasing amount of available data has encouraged another process: one first collects a large dataset, then formulates some hypothesis — is the gene g_s, chosen using the data, associated with the phenotype? — and finally tests this hypothesis, see Table 2.2. But the classical guarantees provided by any statistical method degrade if the impact of the selection is ignored, as illustrated in Figure 2.2.

The aforementioned example is relatively straightforward and it is easy to identify the situation as problematic. In fact, choosing the gene with the highest $|\hat{\beta}_j|$ is equivalent to choosing the gene with the lowest p-value among the q genes, under the simplifying

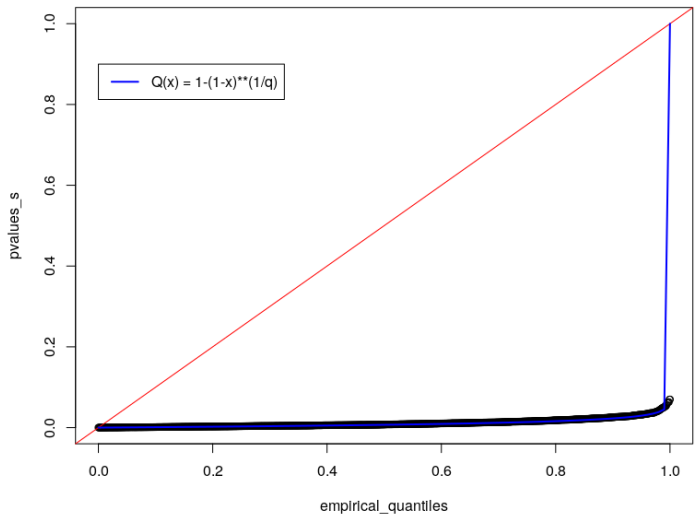


Figure 2.2: Q-Q plot comparing the distribution of p-values for the gene g_s, corresponding to the maximum weight $\hat{\beta}$, to a uniform distribution.

Classical inference	Post-selection inference
1. Devise a model	1. Collect some data
2. Collect some data	2. Select a model
3. Test the hypotheses	3. Test the hypotheses

Table 2.2: Classical and post-selection inference framework.

assumption that the standard errors are all equal. With this assumption, we can compute the probability distribution of p_s if all the null hypotheses hold:

$$\begin{aligned}
\mathbb{P}(\min_{j\leq q}(p_j) \leq x) &= \mathbb{P}(\exists j \leq q\,,\, p_j \leq x) \\
&= 1 - \mathbb{P}(\forall j \leq q\,,\, p_j > x) \\
&= 1 - \mathbb{P}\left(\bigcap_{j\leq q} p_j > x\right) \\
&= 1 - \prod_{j\leq q} \mathbb{P}(x_j > x) \text{ (independence)} \\
&= 1 - (1 - F(x))^q \text{ (with } F \text{ the CDF function of the } p\text{-values)} \\
&= 1 - (1 - x)^q \text{ (under the null, the } p\text{-values are uniform)}
\end{aligned} \tag{2.14}$$

The quantile function of this distribution is then given by:

$$Q(p) = F^{-1}_{\min_{j\leq q}(p_j)}(p) = 1 - (1 - x)^{1/q}\,, \tag{2.15}$$

which is consistent with the empirical distribution found for p_s, as shown in Figure 2.2.

While the selection is relatively easy to identify in this example, it also occurs in less obvious settings. For instance, if we train a CNN and decide to test the resulting sequence

motifs (Chapter 1 Subsection 1.2.3.2), we have to be aware that throughout the network's training step, only a small subset of possible null hypotheses was chosen to be tested: the training resulted in a given number of specific motifs from the set of all possible motifs. Moreover, in this case, the independence assumption made in (2.14) does not hold anymore.

The tools to choose some null hypothesis, such as machine learning methods, are becoming increasingly sophisticated. Benjamini (2020) provides an overview of the implications of selective inference in many fields of science, and the non-management of the selection of null hypothesis is one of the reasons invoked to explain the replication crisis (Ioannidis, 2005).

However, there are tools that provide statistical guarantees in a situation where a lot of null hypotheses (and potentially an infinite number) can be tested.

They can be classified into three categories, which will be the focus of the next sections:

- Simultaneous inference: accounting for the multiplicity of the nulls;
- Sample splitting strategies: working around the problem by using different data to select and to test the nulls;
- Conditional inference: accounting for the selection in the null distribution of the test statistics.

2.2 Simultaneous inference

Let's continue our previous example, and consider that we collected the expression levels for $q = 1\ 000$ genes and set a risk level $\alpha = 5\%$. If none of those genes is associated with the phenotype, each gene has a 5% chance to be declared as significantly associated with the growth rate. This leads to an expected number of $50 = \alpha \times q$ false discoveries — wrongly rejected null hypotheses, or false positives — see Table 2.3.

	$\mathbb{H}_0$ is false	$\mathbb{H}_0$ is true
p-value$\leq \alpha$	TP (true positive)	FP (false positive)
p-value$> \alpha$	TN (false negative)	TN (true negative)

Table 2.3: Possible outcomes when testing a null hypothesis.

As a results, considering the risk level on a test-by-test basis is no longer relevant when dealing with multiple tests.

That's why other criteria have been introduced, and we will focus on two of them:

- The False Discovery Rate (FDR):

$$\text{FDR} = \mathbb{E}\left(\frac{\text{False positives}}{\text{True positives} + \text{False positives}}\right), \tag{2.16}$$

  controlling the expected proportion of wrongly rejected null hypotheses among all the rejected ones. In this equation, $\mathbb{E}$ is the expected value.

- The Family-Wise Error Rate (FWER)

$$\text{FWER} = \mathbb{P}(\text{False positive} \geq 1), \tag{2.17}$$

  the probability of having at least one wrong discovery in the set of rejected null hypotheses.

Any procedure that controls the FWER also controls the FDR. However, if a procedure controls the FDR only, it can be less stringent (Benjamini & Hochberg, 1995, Section 2.1). Intuitively, accepting a given proportion of false discoveries out of a possibly large total of null hypotheses may imply a very high probability of at least one false discovery. However, controlling the FWER will lead to a lower number of discoveries than controlling the FDR. The tools are then used alternatively, depending on the research question and on the consequences of a false discovery.

2.2.1 The Bonferroni correction

The Bonferroni correction is a widely used method to choose the risk level α at the test level to control the FWER.

We need to define the set J_0, containing the indices j of the true null hypotheses $\mathbb{H}_{0,j}$:

$$J_0 = \{j \leq q \,:\, \mathbb{H}_{0,j} \text{ holds}\}\,. \tag{2.18}$$

The Bonferroni correction relies on the following observation (assuming that the null hypotheses are independent):

$$\begin{aligned}
\text{FWER} \;= \mathbb{P}(\text{FP} \geq 1) &= \mathbb{P}\left(\exists j \in J_0 \,:\, p_j \leq \frac{\alpha}{q}\right) \\
&\leq \sum_{j \in J_0} \mathbb{P}\left(p_j \leq \frac{\alpha}{q}\right) \text{ (Boole's inequality)} \\
&\leq \sum_{j \in J_0} \frac{\alpha}{q} \\
&\leq \alpha\,.
\end{aligned}$$

If we want to control the FWER with a risk level α, that is if we want the probability to have at least one false discovery among the q genes we study to be less than α, then using α/q as a threshold on the p-value for declaring a gene significant or not, guarantees it.

While this method provides a very simple way to control the FWER, it is quite conservative. There exist other methods for controlling the FWER while making more discoveries, the Holm-Bonferroni procedure being one of the most widely used.

As discussed in Chapter 1 Section 1.1, the widely-used genome-wide p-value for GWASs is fixed using the Bonferroni correction. Considering that there exists a total of 10^6 interesting SNPs in the human genome, if we want to control the FWER to be lower than 5%, it leads to threshold of $5 \times 10^{-8} = 5 \times 10^{-2}/10^6$. In addition to being based on a very conservative approach, seeking to control the FWER rather than the FDR in the context of GWASs is questionable (Chen et al., 2021).

2.2.2 The Benjamini-Hochberg method

The Benjamini-Hochberg procedure aims at controlling the FDR at a level α. It relies on three steps:

1. Ordering the p-values: $(p_j)_{j \leq q} \mapsto (p_{(j)})_{j \leq q}$, where $\forall j < q\,,\, p_{(j)} \leq p_{(j+1)}$.
   The null hypotheses are re-arranged using the same indices $(\mathbb{H}_{0,(j)})_{j \leq q}$.
2. Find the largest j_0 such that $p_{(j_0)} \leq \frac{j_0 \alpha}{q}$.
3. Reject all null hypotheses $H_{0,(j)}$ for $j \leq j_0$.

The proof that this procedure guarantees $FDR \leq \alpha$ is a bit tedious, and is not relevant to this thesis. It can be found in (Giraud, 2021, Theorem 10.5). But this method provides a particularly useful way to control the FDR.

Both to control the FWER or the FDR, with Bonferroni or Benjamini-Hochberg methods, we see that to reject the null hypothesis $\mathbb{H}_{0,j}$, the corresponding p-value p_j must be lower than a certain threshold, inversely proportional to q. The higher the number of genes to be tested, the lower the threshold. It implies that if the number of genetic variants to be tested increases, then this threshold risks to be low, sometimes falling below the minimum p-value attainable for a given n if applicable.

Moreover, if the number of variants is infinite — for instance, the number of sequence motifs is infinite, as the motifs live in a continuous space — then this multiple testing approach does not work anymore as $q = +\infty$.

2.3 Data-split

A data-split strategy is another approach, feasible when the original dataset can be split into two subsets: a training set and a test set, for instance see Wasserman & Roeder (2009). It relies on the following steps:

1. Randomly split the set of indices $I = \{1, \dots, n\}$ into two independent subsets I_1 and I_2 such that $I_1 \cap I_2 = \emptyset$ and $I_1 \cup I_2 = I$. Then $(\boldsymbol{y}_1, \boldsymbol{G}_1)$, containing the data for indices $i \in I_1$ will be the training set and $(\boldsymbol{y}_2, \boldsymbol{G}_2)$ the test set.

2. Perform any selection method to select any subset of null hypotheses: $J_s \subset \{j \leq q\}$, using only the training set. The number of selected hypotheses is denoted $\kappa < q$.

3. Compute the p-values associated with the null hypotheses $(\mathbb{H}_{0,j})_{j \in J_s}$, using only the test set.

4. Apply a simultaneous inference method if $\kappa > 1$ to correct for the multiplicity of tests (if more than one nulls have been selected).

Let's apply it on our example, still selecting the gene g_s (and therefore the null $\mathbb{H}_{0,s}$) such that $|\hat{\beta}_s| = \max_{j \leq q}(|\hat{\beta}_j|)$. We chose to split the initial dataset into two equal halves $\text{card}(I_1) = \text{card}(I_2)$:

```
pvalues_s <- c()
for (r in 1:m){
     # Generating a dataset under the null
     y <- rnorm(n)
     x <- matrix(rnorm(q*n), nrow=n)

     # Random split of I, and definition of the test and training sets
     I_1 <- sample(I,n/2)
     I_2 <- setdiff(I,I_1)
     x_1, x_2, y_1, y_2 <- x[I_1,], x[I_2,], y[I_1], y[I_2]

     # Training a linear model on the two datasets
     model_1, model_2 <- lm(y_1~x_1), lm(y_2~x_2)

     # Recovering the hat_beta for the training set
     hat_beta_1 <- unname(summary(model_1)$coefficients[2:q,1])

     # Recovering the pvalues for the test set
     pvalues_2 <- unname(summary(model_2)$coefficients[2:q,4])

     # Keeping only pvalue_s, corresponding to max |hat_beta|
     pvalues_s <- c(pvalues_s,
             pvalues_2[abs(hat_beta_1)==max(abs(hat_beta))])
}
```

The resulting Q-Q plot is shown in Figure 2.3, and illustrates the proximity between the distribution of the p-values p_s and the uniform.

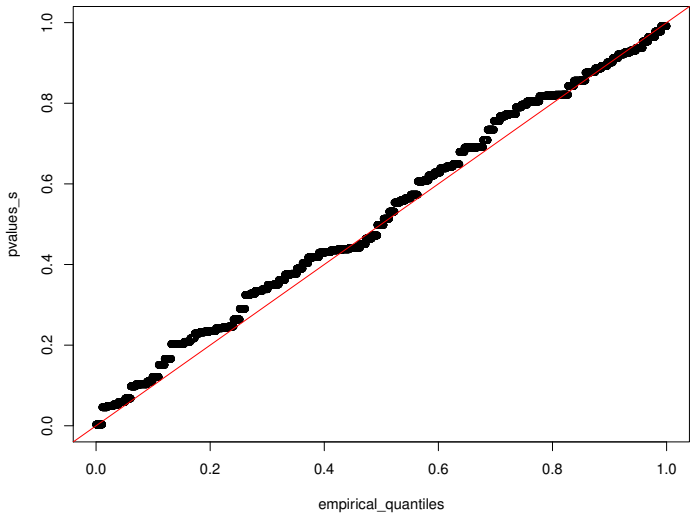


Figure 2.3: Q-Q plot of the distribution of p-values p_s, when using a data-split strategy, versus a uniform distribution.

Finally, using a data-split strategy has two main advantages: it is usually very straightforward to implement — it works automatically in many settings —, and it is fairly assumption free. However, we can identify three drawbacks to using it:

- It introduces some irreproducibility, as the resulting p-values depend on the random split performed at step 1. Two different splits may lead to different p-values.
- It is inefficient, in a sense that it only uses part of the data to select the null hypotheses, and the other part to test them. So, the training step might have lower selection performance, and the test procedure is less powerful than one that would have used all the data.
- It might be infeasible, depending on the data structure. For instance, time series are not suited to data-split, as it is hard to define independent subsets. If the original dataset contains rare observations, a data-split approach may also be blind to those observations: if the corresponding samples are only in I_1, the test of the corresponding null will have no power, and conversely, if they are in I_2, the corresponding null will not be selected.

2.4 Conditional inference

Conditional inference is a different approach, that uses all the available data both for the selection and for the test step. For each selected null hypothesis $\mathbb{H}_{0,j}$, it constructs the distribution of the test statistics under the null, conditional on the prior selection of $\mathbb{H}_{0,j}$ using a data-dependent procedure.

A well-studied case where conditional inference occurs is Pearson's χ^2 test (Fisher, 1922). Let's say we want to use this test to evaluate the goodness of fit of a random variable X, for which we have a realization $(x_1, \dots, x_n)$ — divided in k classes — with a uniform distribution $\mathcal{U}$. To test the null hypothesis $\mathbb{H}_0$: "$X \sim \mathcal{U}$", we compute the test statistics and compare it with the quantiles of a $\chi^2(k-1)$, its asymptotic distribution when $n \to \infty$.

But if we want to compare the distribution of X with a Poisson distribution $\mathcal{P}$, the first step is then to look for the parameter ν, such that the first moment of $\mathcal{P}(\nu)$ is equal to the estimated first moment of X — *i.e.* the average of $(x_1, \dots, x_n)$. Then the null hypothesis becomes $\mathbb{H}_{0,\nu}$: "$X \sim \mathcal{P}(\nu)$", and it was selected among all possible ν. The distribution of the test statistics under the null then must be adjusted to account for this selection. Under the right assumptions, it is well established that the new asymptotic distribution is $\chi^2(k-2)$. The distribution of the test statistics under the null has been changed, to account for the selection. This well-known example is intended to illustrate the modification in the distribution of the test statistic under a conditional null hypothesis, compared to an unconditional one.

But besides being very specific, this example is only valid in the asymptotic framework. We will therefore extend this approach to other testing frameworks, and to results valid for any n.

2.4.1 Getting intuition on an easy example

Let's go back to our example. Here, instead of choosing the gene g_s with the highest $|\hat{\beta}|$, we are interested in selecting all the genes g_j with $j \in J_\tau$ such that:

$$J_\tau = \left\{ j \leq q \,,\, V_j = \frac{|\hat{\beta}_j|}{\mathrm{se}(\hat{\beta}_j)} > \tau \right\} , \tag{2.19}$$

for a given threshold τ.

We denote F the CDF function of $|\mathcal{T}(n-(q+1))|$. Under the non-selective null hypothesis $\mathbb{H}_{0,j}$, we have $\mathbb{P}(V_j \leq x) = F(x)$. And the non-conditional p-value $p_{j_{\mathrm{nc}}}$ is:

$$p_{j_{\mathrm{nc}}} = 1 - F(V_j) \,. \tag{2.20}$$

Let's focus on what happens under a selective null. We can obtain the selective null distribution $F_{\mathrm{cond}}(x) = \mathbb{P}_{V_j > \tau}(V_j \leq x)$:

$$\forall x \in \mathbb{R}^+ \qquad F_{\mathrm{cond}}(x) = \frac{\mathbb{P}(V_j > \tau \cap V_j \leq x)}{\mathbb{P}(V_j > \tau)} = \begin{cases} 0 & \text{if } x < \tau \,, \\ \frac{F(x)-F(\tau)}{1-F(\tau)} & \text{otherwise} \,. \end{cases} \tag{2.21}$$

The conditional null distribution is then the non-conditional null distribution, truncated at $x = \tau$. And we obtain the conditional p-value $p_{j_{\text{cond}}}$:

$$p_{j_{\text{cond}}} = 1 - F_{\text{cond}}(V_j) = 1 - \frac{F(V) - F(\tau)}{1 - F(\tau)} = \frac{p_{j_{\text{nc}}}}{1 - F(\tau)}\,. \tag{2.22}$$

This result is illustrated using the following experiment, with $q = 1\,000$, $n = 10\,000$ and $\tau = 0.1$:

```
# Generating data under the null
y <- rnorm(n, 0, 10)
G <- matrix(rnorm(q*n), nrow=n)

# Fitting the linear regression
model <- lm(y~G)

# Recovering the hat_beta and all the (non-conditional) pvalues
hat_beta <- unname(summary(model)$coefficients[2:q,1])
pvalues = unname(summary(model)$coefficients[2:q,4])

# Keeping only the ones such that |hat_beta| > tau
pvalues <- pvalues[abs(hat_beta)>tau]
```

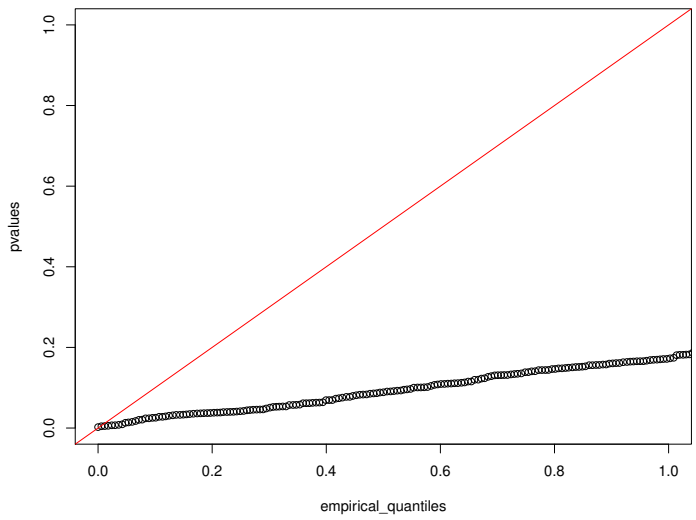


Figure 2.4: Q-Q plot of the empirical non-conditional distribution of the p-values (p_j), for $j \in J_\tau$.

The Q-Q plot provided in Figure 2.4 shows the empirical distribution of the non-conditional (but still selected using $V_j > \tau$) p-values $p_{j_{\text{nc}}}$ obtained in the experiment. While it shows the expected selection bias, the distribution's alignment on a straight line highlights the linear relationship between these unconditional p-values and the conditional p-values $p_{j_{\text{cond}}}$ (2.22) that would have produced a calibrated test.

2.4.2 Conditional inference with the LASSO

To further study our example, we now want to change the way we select the genes we want to test. Instead of solving the least-squares problem (2.3), we will solve a penalized version of it:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta'}\in\mathbb{R}^q}{\arg\min} \left(\frac{1}{n} \|\boldsymbol{y} - \boldsymbol{G}^T\boldsymbol{\beta'}\|_2^2 + \lambda \|\boldsymbol{\beta'}\|_1 \right) \tag{2.23}$$

with λ a regularization parameter.

This problem has been introduced in (Tibshirani, 1996) as the LASSO (least absolute shrinkage and selection operator). It is convex if $\text{rank}(\boldsymbol{G}) = q$ and therefore admits a global minimum, and is known to produce sparse solutions. That is, if λ is correctly chosen, only a subset of weights $\hat{\beta}_j$ will be non-zero. We denote this set M:

$$M = \left\{ j \leq q \,:\, \hat{\beta}_j \neq 0 \right\}, \tag{2.24}$$

and M will be referred to as the selected model for the phenotypes of our original dataset $\boldsymbol{y}$. For another phenotype $\boldsymbol{y'} \in \mathbb{R}^n$, we will denote $\hat{M}(\boldsymbol{y'})$ the model selected by optimizing the LASSO (2.23). We can note here that while model selection can be performed for interpretability issues — a model containing fewer predictors is easier to interpret, but can also be required when working with high dimensional data (q>n). In this case, the linear regression (2.3) becomes ill-defined, as it admits an infinite number of solutions, contrarily to the LASSO.

We then want to test the genes belonging to the selected model: $(g_j)_{j\in M}$. To that end, we will follow the procedure described in (Lee et al., 2016), (Tibshirani et al., 2015) and (Hastie et al., 2015, Chapter 6).

This method can be divided into two steps: obtaining a simple characterization for the selection of M, and obtaining an analytical expression for the CDF of the test statistics, conditionally to the selection.

• The LASSO selection event as a union of polyhedra

First, the authors define the selection event $E(M)$, that is the set of vectors $\boldsymbol{y'} \in \mathbb{R}^n$ that lead to selection of the same model M as $\boldsymbol{y}$:

$$E(M) = \left\{ \boldsymbol{y'} \in \mathbb{R}^n \,:\, \hat{M}(\boldsymbol{y'}) = \hat{M}(\boldsymbol{y}) = M \right\}. \tag{2.25}$$

They show that this set is a union of polyhedra. In other words, it can be described as a union of sets, each of which can be described using linear constraints:

Theorem 2.4.1.
Let $S = \{-1, 1\}^{|M|}$. There exists $card(S) = 2^{|M|}$ matrices $(\boldsymbol{A}(M, \boldsymbol{s}))_{\boldsymbol{s}\in S}$ and $2^{|M|}$ vectors $(\boldsymbol{b}(M, \boldsymbol{s}))_{\boldsymbol{s}\in S}$ such that:

$$E(M) = \bigcup_{\boldsymbol{s}\in S} \left\{ \boldsymbol{y'} \in \mathbb{R}^n \,:\, \boldsymbol{A}(M, \boldsymbol{s})\boldsymbol{y'} \leq \boldsymbol{b}(M, \boldsymbol{s}) \right\}. \tag{2.26}$$

While the proof of this theorem is too long for this thesis, we will give here a brief overview to have some intuition about this theorem.

It starts by introducing $\boldsymbol{s}$, the vector containing the signs of $\hat{\boldsymbol{\beta}}$. The proof is then mainly based on the Karush-Kuhn-Tucker conditions (KKT). Indeed, for $\hat{\boldsymbol{\beta}}$ to be a solution of (2.23), it is necessary and sufficient that $(\hat{\boldsymbol{\beta}}, \boldsymbol{s})$ satisfy the KKT conditions. Next, the authors take advantage of those conditions to characterize the set of $\boldsymbol{y}' \in \mathbb{R}^n$ that lead to the selection of a particular pair $(M, \boldsymbol{s})$ and show that it can be described using a set of linear constraints $\boldsymbol{A}(M, \boldsymbol{s})\boldsymbol{y}' \leq \boldsymbol{b}(M, \boldsymbol{s})$. To conclude the proof, they make the union of the sets on all sign vectors $\boldsymbol{s}$.

We then see that conditioning with respect to the pair $(M, \boldsymbol{s})$ is easier than using M alone, since the selection event boils down to a single polyhedron, *i,e* an intersection of linear constraints. In the following, we condition on the selection of genes $(g_j)_{j \in M}$ and on the signs with which those genes participate in the linear model (2.9).

Compared to Section 2.1.2, we slightly modify the test statistics, and the model. Here, we suppose that the phenotype is normally distributed $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$ a covariance matrix. We still assume homoscedasticity: $\exists \sigma \in \mathbb{R}\,,\ \boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}_n$.

While the genes $(g_j)_{j \in M}$ have been selected using the LASSO (2.23), we will test them using a standard least-squares model (2.3). The new null hypotheses $(\mathbb{H}_{0,j})_{j \in M}$ and test statistics $(V_j)_{j \in M}$ make use of the coefficients β^* of this regression, as computed in (2.4):

$$\begin{aligned} &\boldsymbol{\beta}^* = (\boldsymbol{G}_M \boldsymbol{G}_M^T)^{-1} \boldsymbol{G}_M \boldsymbol{y}\,, \\ &\mathbb{H}_{0,j} \ : \ \text{“}\beta_j^* = 0\text{”}\,, \\ &V_j = \beta_j^* = \boldsymbol{e}_j^T \boldsymbol{\beta}^*\,, \end{aligned} \tag{2.27}$$

where $\boldsymbol{G}_M$ is the data matrix restrained to genes in M, and $\boldsymbol{e}_j$ is the j^{th} vector of the canonical basis. In order to generalize, we can consider test statistics of the form $\boldsymbol{\eta}^T \boldsymbol{y}$, with $\boldsymbol{\eta} \in \mathbb{R}^n$. As a results, V_j becomes a special case and can be written in this form using $\boldsymbol{\eta} = \boldsymbol{e}_j$. Under the null hypothesis $\mathbb{H}_{0,j}$, $V_j \sim \mathcal{N}(\boldsymbol{\eta}^T \boldsymbol{\mu}, \boldsymbol{\eta}^T \boldsymbol{\Sigma} \boldsymbol{\eta})$.

To find the conditional distribution of V_j, the authors study:

$$\boldsymbol{\eta}^T \boldsymbol{y} | \left\{ \hat{M} = M, \hat{\boldsymbol{s}} = \boldsymbol{s} \right\}\,. \tag{2.28}$$

They rewrite this selection event $\left\{ \hat{M} = M, \hat{\boldsymbol{s}} = \boldsymbol{s} \ \right\} = \{\boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\}$ in terms of $\boldsymbol{\eta}^T \boldsymbol{y}$ (Lee et al., 2016, Lemma 5.1):

Lemma 2.4.1 (Polyhedra selection as truncation)**.**

$$\{\boldsymbol{A}\boldsymbol{y} \leq \boldsymbol{b}\} \ = \left\{ \mathcal{V}^-(\boldsymbol{y}) \leq \boldsymbol{\eta}^T \boldsymbol{y} \leq \mathcal{V}^+(\boldsymbol{y})\,, \mathcal{V}^0(\boldsymbol{y}) \geq 0 \right\}\,, \tag{2.29}$$

where:

- $\mathcal{V}^-(\boldsymbol{y}) = \max\limits_{j\,:\,\boldsymbol{\rho}_j > 0} \dfrac{\boldsymbol{b}_j - (\boldsymbol{A}\boldsymbol{y})_j + \boldsymbol{\rho}_j \boldsymbol{\eta}^T \boldsymbol{y}}{\boldsymbol{\rho}_j}$,
- $\mathcal{V}^+(\boldsymbol{y}) = \min\limits_{j\,:\,\boldsymbol{\rho}_j < 0} \dfrac{\boldsymbol{b}_j - (\boldsymbol{A}\boldsymbol{y})_j + \boldsymbol{\rho}_j \boldsymbol{\eta}^T \boldsymbol{y}}{\boldsymbol{\rho}_j}$,
- $\mathcal{V}^0(\boldsymbol{y}) = \max\limits_{j\,:\,\boldsymbol{\rho}_j = 0} \boldsymbol{b}_j - (\boldsymbol{A}\boldsymbol{y})_j$,

and $\boldsymbol{\rho} = \boldsymbol{A\Sigma\eta}/\boldsymbol{\eta}^T\boldsymbol{\Sigma\eta}$.

This lemma is simply obtained by decomposing $\boldsymbol{y} = \boldsymbol{c}(\boldsymbol{\eta}^T\boldsymbol{y}) + \boldsymbol{z}$ with $\boldsymbol{c} = \boldsymbol{\Sigma\eta}(\boldsymbol{\eta}^T\boldsymbol{\Sigma\eta})^{-1}$ and $\boldsymbol{z} = (\boldsymbol{I}_n - \boldsymbol{c}\boldsymbol{\eta}^T)\boldsymbol{y}$. Then, we transform $\{\boldsymbol{Ay} \leq \boldsymbol{b}\}$ into inequalities on $\boldsymbol{\eta}^T\boldsymbol{y}$.

Of note, if $\boldsymbol{\Sigma}$ can be written as $\sigma^2\boldsymbol{I}_n$, with $\sigma \in \mathbb{R}$, then $\boldsymbol{z} = \boldsymbol{P}_{\boldsymbol{\eta}^\perp}(\boldsymbol{y})$, the projection of $\boldsymbol{y}$ onto $\boldsymbol{\eta}^\perp$ and $\boldsymbol{c}(\boldsymbol{\eta}^T\boldsymbol{y}) = \boldsymbol{P}_{\boldsymbol{\eta}}(\boldsymbol{y})$. Finally, this lemma lends itself easily to a geometrical interpretation, see Figure 2.5.

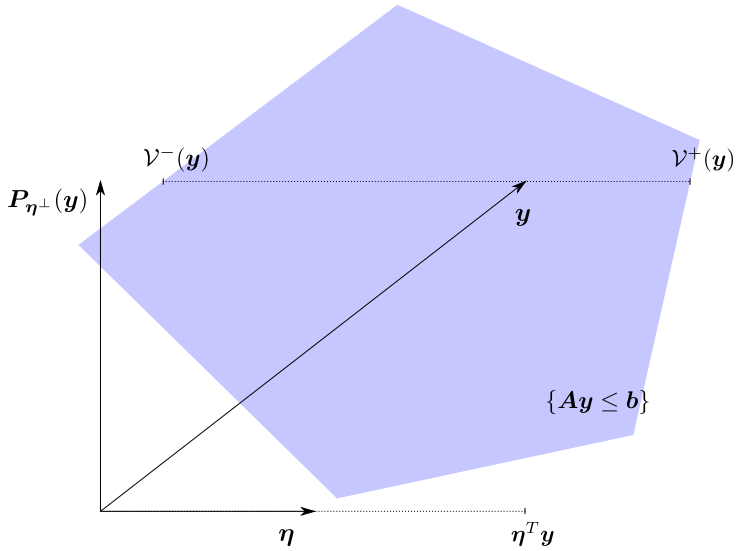


Figure 2.5: Geometrical interpretation of the polyhedral lemma, with $\boldsymbol{\Sigma} = \boldsymbol{I}_n$ and $\|\boldsymbol{\eta}\| = 1$. The event $\{\boldsymbol{Ay} \leq \boldsymbol{b}\}$ can be characterized as $\{\mathcal{V}^-(\boldsymbol{y}) \leq \boldsymbol{\eta}^T\boldsymbol{y} \leq \mathcal{V}^+(\boldsymbol{y}), \mathcal{V}^0(\boldsymbol{y}) \geq 0\}$. Conditioning on $(M, \boldsymbol{s})$ is equivalent to conditioning $\boldsymbol{\eta}^T\boldsymbol{y}$ on a certain segment. We also note that $\mathcal{V}^{+,-}(\boldsymbol{y})$ only depend on $\boldsymbol{P}_{\boldsymbol{\eta}^\perp}(\boldsymbol{y})$ and are then independent of $\boldsymbol{\eta}^T\boldsymbol{y}$.

This lemma tells us that $[\boldsymbol{\eta}^T\boldsymbol{y}|\{\boldsymbol{Ay} \leq \boldsymbol{b}\}]$ and $[\boldsymbol{\eta}^T\boldsymbol{y}|\{\ \mathcal{V}^-(\boldsymbol{y}) \leq \boldsymbol{\eta}^T\boldsymbol{y} \leq \mathcal{V}^+(\boldsymbol{y}), \mathcal{V}^0(\boldsymbol{y}) \geq 0\}]$ are equally distributed.

• **The conditional null is the truncated null**

From Lemma 2.4.1, the authors derive the following result:

Lemma 2.4.2 (Pivotal statistic after polyhedral selection)**.**
Let Φ *denote the CDF of* $\mathcal{N}(0,1)$*. Let* $F^{[a,b]}_{\theta,\sigma^2}$ *the CDF of a* $\mathcal{N}(\theta,\sigma^2)$ *random variable to lie in* $[a,b]$*, i,e*

$$\forall x\,,\ F^{[a,b]}_{\theta,\sigma^2}(x) = \frac{\Phi((x-\theta)/\sigma) - \Phi((a-\theta)/\sigma)}{\Phi((b-\theta)/\sigma) - \Phi((a-\mu)/\sigma)}\,. \tag{2.30}$$

For $\boldsymbol{\eta}^T\boldsymbol{\Sigma\eta} \neq 0$*, we have*

$$F^{[\mathcal{V}^-,\mathcal{V}^+]}_{\boldsymbol{\eta}^T\boldsymbol{\mu},\boldsymbol{\eta}^T\boldsymbol{\Sigma\eta}}(\boldsymbol{\eta}^T\boldsymbol{y})|\{\boldsymbol{Ay} \leq \boldsymbol{b}\} \sim \mathit{Unif}(0,1)\,. \tag{2.31}$$

To remove the conditioning on the signs, and condition only on M, we just need to perform the union over all sign vector $\boldsymbol{s}$:

$$F^{\bigcup_{\boldsymbol{s}}[\mathcal{V}^-_{\boldsymbol{s}},\mathcal{V}^+_{\boldsymbol{s}}]}_{0,\boldsymbol{\eta}^T\boldsymbol{\Sigma\eta}}(\boldsymbol{\eta}^T\boldsymbol{y})|\bigcup_{\boldsymbol{s}}\{\boldsymbol{A}_{\boldsymbol{s}}\boldsymbol{y} \leq \boldsymbol{b}_{\boldsymbol{s}}\} \sim \mathrm{Unif}(0,1)\,. \tag{2.32}$$

A valid test procedure for the genes $(g_j)_{j\in M}$ obtained via optimization of the lasso problem (2.23) is then to compute the p-values for the null $\mathbb{H}_{0,j}$: "$\beta_j^* = 0$" as follows:

$$p_j = 1 - F^{\bigcup_s [\mathcal{V}_s^-, \mathcal{V}_s^+]}_{\boldsymbol{\eta}^T\boldsymbol{\mu}, \boldsymbol{\eta}^T\boldsymbol{\Sigma}\boldsymbol{\eta}}(\beta_j^*)\,, \tag{2.33}$$

for a one-sided test ($\mathbb{H}_1$: "$\beta_j^* > 0$"), or to use:

$$p_j = 2 \times \min\left\{F^{\bigcup_s [\mathcal{V}_s^-, \mathcal{V}_s^+]}_{0, \boldsymbol{\eta}^T\boldsymbol{\Sigma}\boldsymbol{\eta}}(\beta_j^*), 1 - F^{\bigcup_s [\mathcal{V}_s^-, \mathcal{V}_s^+]}_{0, \boldsymbol{\eta}^T\boldsymbol{\Sigma}\boldsymbol{\eta}}(\beta_j^*)\right\}\,, \tag{2.34}$$

for a two-sided test ($\mathbb{H}_1$: "$\beta_j^* \neq 0$").

This procedure has been implemented in the **R** package 'selectiveInference' (Tibshirani et al., 2019). Let's use it on our example:

```
pvalues <- c()
for (r in 1:m){
    # Generating data under the null
    y <- rnorm(n)
    G <- scale(matrix(rnorm(q*n), nrow=n), TRUE, TRUE)

    # Fitting the LASSO
    fit <- glmnet(G, y, standardize = TRUE)

    # Extracting the coefficients for a given lambda
    lambda <- 0.8
    beta <- coef(fit, s=lambda/n, exact = TRUE, x=G, y=y)

    # Computing the pvalues
    res <- fixedLassoInf(G, y, beta, lambda, sigma =1)
    pvalues <- c(pvalues, res$pv)
```

The Q-Q plot obtained is in Figure 2.6 and confirms the validity of this conditional inference procedure.

Of note, equivalent results exist when the selection can be described as an intersection of quadratic inequalities, particularly useful for group-LASSO (Loftus & Taylor, 2015), but the support of the truncation becomes very computationally intensive $O\left(\binom{q}{\text{card}(M)}\right)$, quickly becoming intractable. To overcome this issue, sampling-based strategies can be implemented to approximate the conditional distribution, see Subsection 2.4.4.

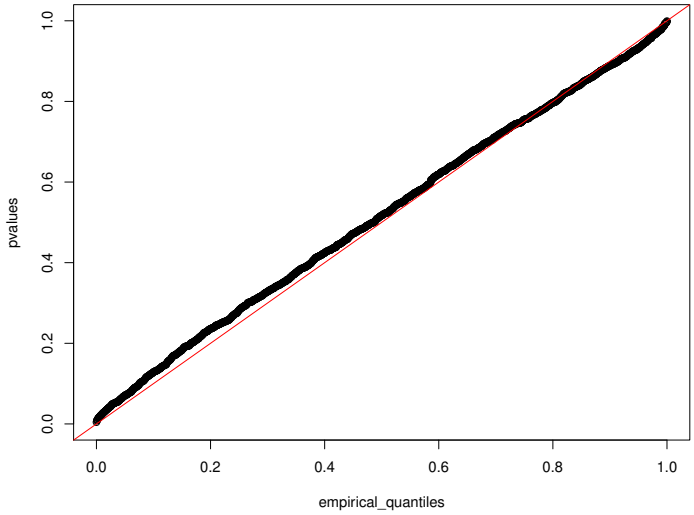


Figure 2.6: Q-Q plot of the distribution of p-values obtained with the conditional inference procedure after a LASSO selection, against a uniform distribution.

2.4.3 Some extensions in the linear case

The previous results have been extended in different directions.

• **Sequential selection procedures**

Tibshirani et al. (2015) extend this work to some sequential regression procedures, such as:

- The forward stepwise regression, which starts from a model containing zero predictors (zero genes), and incrementally adds the predictor that most improves the fit to the model.
- The Least Angle Regression, introduced by Efron et al. (2004).

Both selection procedures can be described using unions of polyhedra, and the Lemmas 2.4.1 and 2.4.2 can be applied, with modifications in matrices $\boldsymbol{A}, \boldsymbol{b}$ and functions $\mathcal{V}^-, \mathcal{V}^+, \mathcal{V}^0$.

In the context of sequential selection procedures, there are two options for testing the genes $\boldsymbol{g}_j$. Let's define M_k the selected model at step k, such that for all k $M_k \subset M_{k+1}$. Then:

- Either we test the gene g_j as soon as it enters the model at step k. Its new index in the model M_k is (j).

  Defining $\beta^*_{(j),k} = \boldsymbol{e}_{(j)}\boldsymbol{\beta}^*_k$ with $\boldsymbol{\beta}^*_k = (\boldsymbol{G}_{M_k}\boldsymbol{G}^T_{M_k})^{-1}\boldsymbol{G}_{M_k}\boldsymbol{y}$. The null hypothesis becomes $\mathrm{H}_{0,j}$: "$\beta^*_{(j),k} = 0$", the test statistics $V_j = \beta^*_{(j),k}$, and the conditioning is on the selection of M_k.

That is, the gene entering the model during step k is tested in the context of the previously entered genes.

- Or we test the gene g_j in the context of the entire model M, obtained at the final step. Then the test is performed as described in (2.27).

Both options are valid, they only lead to different interpretations. The literature tends to favor the second option: the test in the context of the final model.

• Groups of features

Our example dataset contains q genes, and for now we have considered them each separately. But it could be interesting to define groups of gene, for instance by grouping the genes that are in the same biological pathways. Reid & Tibshirani (2015); Reid et al. (2015); Loftus & Taylor (2015) define a test procedure for such groups. The method can be described as follows:

1. Group the predictors (the genes of our example), using either some knowledge considerations or a clustering method.
2. For each group, extract a *prototype*, that is a representative for the group. It can be achieved either in an unsupervised way (*e.g.* average the expressions of the genes) or in a supervised way (*e.g.* select the gene g_j with the highest marginal correlation with $\boldsymbol{y}$).
3. Select some prototypes, using forward stepwise regression (or another method) and test them using a conditional inference approach, using adapted version of Lemmas 2.4.1 and 2.4.2.

2.4.4 Extensions to the non-linear framework

Until now, we assumed that the gene expression levels were associated with the growth rate in a linear manner (2.9). But we can try to go beyond this linear framework, and test other types of association. To that end,(Yamada et al., 2018) propose a kernel-based conditional inference approach.

• Selection with HSIC criterion

In addition to allowing non-linear associations, the use of kernels (see Chapter 1 Subsection 1.2.2) allows for using non-numerical data, such as biological sequences instead of gene expressions.

In this approach, the authors estimate the discrepancy from independence between the j^{th} predictor (the gene g_j in our example, but not necessarily its expression level) and the outcome $\boldsymbol{y}$ (the phenotype): $\hat{I}(g_j, \boldsymbol{y})$, the estimate of the true discrepancy $I(g_j, \boldsymbol{y})$.

They denote $\boldsymbol{z} \in \mathbb{R}^q$ such that $\boldsymbol{z}_j = \hat{I}(g_j, \boldsymbol{y})$, and assume that it is normally distributed with mean $\boldsymbol{\mu} \in \mathbb{R}^q$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{q \times q}$:

$$\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \,. \tag{2.35}$$

To estimate the discrepancy from independence, the authors rely on the Hilbert-Schmidt Independence Criterion $I(g_j, \boldsymbol{y}) = \text{HSIC}(g_j, \boldsymbol{y})$ and rely on its empirical approximation $\widehat{\text{HSIC}}(g_j, \boldsymbol{y})$, as introduced in (Gretton et al., 2005). This criterion is known to asymptotically follow normal distribution when n goes to infinity (Zhang et al., 2018), justifying (2.35).

It can be expressed using two kernels, one on the predictors and one on the output. It therefore measures a non-linear association between $\boldsymbol{y}$ and g_j, depending on the chosen kernels. For gene g_j, the authors define the null hypothesis as the independence:

$$\mathbb{H}_{0,j} \; : \; \text{“HSIC}(g_j, \boldsymbol{y}) = 0\text{”}\,, \tag{2.36}$$

and the test statistics is $V_j = \boldsymbol{z}_j$.

They select the set of predictors M (containing a fixed number) with the highest discrepancy from independence, leading to the selection event:

$$E(M) = \left\{ \boldsymbol{y}' \in \mathbb{R}^n \,,\, \forall (m, \ell) \in M \times \{1, \dots, q\} \backslash M, \hat{I}(g_m, \boldsymbol{y}') \geq \hat{I}(g_l, \boldsymbol{y}') \right\} . \tag{2.37}$$

Then the selection event, originally expressed as a set of constraints on $\boldsymbol{y}'$, can be rewritten as a set of linear inequalities with respect to $\boldsymbol{z}$, and the results derived in Subsection 2.4.2 can be applied to find an analytical expression for the distribution of $\boldsymbol{z}_j$ under the null (2.36), conditionally to the selection of the model M.

To sum things up, (Yamada et al., 2018) enables to select and test the predictors g_j using a non-linear association with the outcome, this association being defined by the two given kernels in HSIC. The new parametrization with $\boldsymbol{z}$ allowed them to switch from non-linear constraints on $\boldsymbol{y}$ to linear constraints on $\boldsymbol{z}$, and thus to apply the aforementioned results.

• General framework for kernel selection

However, (Slim et al., 2019) go even further, and propose a general framework for selecting and testing kernels, extending the possible associations types.

As discussed in Chapter 1, a kernel defines an implicit representation of the data. If we rely on a parameterized class of kernel functions $\mathcal{K} = \{K_\theta \,, \theta \in \Theta\}$, with Θ a parameter space, then we can be interested in testing the association between K_θ and the phenotype.

For the sake of clarity, let's apply it to our original example, with $\boldsymbol{y}$ the vector containing the growth rates for n cells. But now, instead of having the gene expression levels g, let's say the inputs are the DNA sequences of those cells: $X = (x_i)_{i \leq n}$. We first define the class of kernel functions:

$$\tilde{\mathcal{K}} = \{K_\theta \; : \; \exists \theta \in \Theta \,,\, K_\theta(x, x') = \langle \varphi_\theta(x), \varphi_\theta(x') \rangle\} \,, \tag{2.38}$$

where Θ is the set of existing k-mers of a given length k, and $\varphi_\theta(x)$ is the number of occurrences of θ in x — in this simple case $\langle \varphi_\theta(x), \varphi_\theta(x') \rangle = \varphi_\theta(x) \times \varphi_\theta(x')$.

We can then define the association between a kernel K_θ and the phenotype, *e.g* by using the squared correlation between $\boldsymbol{\varphi}_\theta(X) = [\varphi_\theta(x_i)]_{i \leq n}$:

$$\text{association}(K_\theta, \boldsymbol{y}) = \boldsymbol{y}^T \boldsymbol{\varphi}_\theta(X) \boldsymbol{\varphi}_\theta(X)^T \boldsymbol{y} = \boldsymbol{y}^T \boldsymbol{K}_\theta \boldsymbol{y} \,, \tag{2.39}$$

with $\boldsymbol{K}_\theta$ the Gram matrix associated to K_θ.

Under the Gaussian assumption $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we are interested in the following null hypotheses:

$$\mathrm{H}_{0,\theta} \; : \; \text{“association}(K_\theta, \boldsymbol{\mu}) = 0\text{”}\,, \tag{2.40}$$

using $V_j = \text{association}(K_\theta, \boldsymbol{y})$ as a test statistics.

We can select a limited number of $\{\theta_j\}_{j \in M}$ by applying one of the aforementioned selection procedures to maximize the association (*e.g.* LASSO, FS or LAR) and test their respective null hypotheses, conditionally to the selection of M.

However, the authors show that the selection event

$$\left\{ \boldsymbol{y}' \in \mathbb{R}^n \; : \; \arg\max_{M'} \left(\text{association}(K_{\{\theta_j\} \in M'}, \boldsymbol{y}') \right) = M \right\}\,, \tag{2.41}$$

cannot be written as a set of linear constraints on $\boldsymbol{y}'$, but as a set of quadratic constraints.

As discussed above, obtaining the right truncation of the null distribution to account for this conditioning is then theoretically feasible, but intractable, and the authors rely on a smart rejection sampling strategy to approximate this distribution: the hypersphere direction sampling, first proposed by Berbee et al. (1987). This method will be discussed in detail in Chapter 4 Subsection 4.3.1.2, as the inference procedure we propose is based on it.

2.5 Current limitations of conditional inference

To sum things up, there exists three main classes of methods to perform valid inference in a context of multiple null hypotheses. The simultaneous inference approach is complementary to data-split and conditional inference as it gives different statistical guarantees. Data-split, when applicable, is quite straightforward, but is data-inefficient compared to the conditional inference approach, as it only uses distinct subsets to perform selection and inference.

However, we can identify several limitations to the use of conditional inference:

- It often relies on some assumptions regarding the data, and particularly the output, such a Gaussian assumption with a known variance. While some results exist for the variance (Lee et al., 2016, Section 8), we believe our results from Chapter 4 can improve this point.
- They are designed to work with selection among a finite number of predictors. But when working with continuous variants, such as sequence motifs, the selection is performed over a infinite set of features. Chapter 4 introduces a method to overcome this issue.

CHAPTER 3

Discovering sequence motifs with SEISM

Within the following chapters, we set out to go beyond explainable machine learning by introducing SElective Inference for Sequence Motifs (SEISM): a valid statistical inference procedure for features obtained using interpretability tools over machine learning models. SEISM is introduced in (Villié et al., 2022), and a PyTorch implementation is provided at `https://gitlab.in2p3.fr/antoine.villie1.seism`.

This chapter aims at defining a procedure for selecting q sequence motifs $\boldsymbol{Z} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_q\}$ that are associated with a phenotype in a given dataset $(\boldsymbol{X}, \boldsymbol{y})$ containing n samples. Each sample is composed of a one-hot encoded sequence $x_i \in \mathcal{X}$, defined over an alphabet $\mathcal{A}$ (Chapter 1 Section 1.2), and of the corresponding measurement of a biological property $y_i \in \mathcal{Y}$.

The motifs must match the constraint (1.16) provided in Chapter 1, and thus $\boldsymbol{Z} \in \mathcal{Z}^q$, where $\mathcal{Z}$ is a subset of $\mathbb{R}^{|\mathcal{A}| \times k}$, given by the simplex:

$$\mathcal{Z} = \left\{ z \in \mathbb{R}_+^{|\mathcal{A}| \times k} \ : \ \forall j \leq k \, , \ \sum_{i=1}^{|\mathcal{A}|} z_{i,j} = 1 \right\} , \tag{3.1}$$

and k is the length of the motif.

In order to do so, we cast commonly used CNNs in a feature selection framework. While achieving state-of-art performance for predictions tasks, their feature selection performance suffers limitations discussed in Chapter 1: instabilities, irrelevant, redundant or partial motifs... We will then modify those CNNs to work around these issues and show that they can be fitted into a broader analysis paradigm: association scores. By representing various association types between motifs and phenotypes using this new paradigm, we are able to create a more versatile selection approach. These scores will then be leveraged in Chapter 4 to derive a valid test procedure for the selected motifs.

In the second part of the chapter, we compare the motif selection performance of SEISM

with existing *de-novo* motifs discovery tools, and show that SEISM reaches state-of-art performance.

3.1 Association scores and link with CNNs

As described in Chapter 1 Subsection 1.2.3.2, one-layer CNNs parametrize a function $g : \mathcal{X} \mapsto \mathcal{Y}$ by q filters of length k and q weights $\boldsymbol{\beta} \in \mathbb{R}^q$. The function g can be decomposed as a linear predictor applied over a data representation:

$$\forall x \in \mathcal{X}\,,\ g(x) = \boldsymbol{\varphi}^{\boldsymbol{Z},x}\boldsymbol{\beta}\,. \tag{3.2}$$

In this section, the focus is put on the loss function, *i.e.* the function that measures the difference between the predicted output and the true one. The goal of training a model is to minimize the value of this function. The classical framework for CNNs, using the data $\{\boldsymbol{X}, \boldsymbol{y}\}$, relies on a quadratic loss measuring the empirical risk, associated with a L^2 penalty:

$$\min_{(\boldsymbol{Z},\boldsymbol{\beta})\in(\mathcal{Z},\mathbb{R}^q)} n^{-1}\|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\,, \tag{3.3}$$

with $\varphi^{\boldsymbol{Z},\boldsymbol{X}} \in \mathbb{R}^{n\times q}$ such that $[\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}]_{(i,\cdot)} = \boldsymbol{\varphi}^{\boldsymbol{Z},x_i}$ and some $\lambda > 0$.

Problem (3.3) defines a convex and differentiable function of $\boldsymbol{\beta}$, and its minimum can then be found by setting the gradient in $\boldsymbol{\beta}$ to $\mathbf{0}$:

$$\begin{aligned} 0 &= -2n^{-1}\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\left(\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\boldsymbol{\beta}\right) + 2\lambda\boldsymbol{\beta} \\ &= \left(\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} + \lambda n\boldsymbol{I}_q\right)\boldsymbol{\beta} + \left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{y}\,. \end{aligned}$$

Then we can observe that $\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ is positive, and definite if $\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ has rank q:

$$\forall \boldsymbol{v} \in \mathbb{R}^q\,,\ \boldsymbol{v}^T\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\boldsymbol{v} = \langle\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\boldsymbol{v}, \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\boldsymbol{v}\rangle \geq 0\,,$$

leading to $\left(\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} + n\lambda\boldsymbol{I}_q\right)$ being invertible. Next, we derive the optimal solution for (3.3):

$$\boldsymbol{\beta}^* = \left(\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} + \lambda n\boldsymbol{I}_q\right)^{-1}\left(\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\right)^T\boldsymbol{y}\,. \tag{3.4}$$

By plugging this solution (3.4) into (3.3), we obtain:

$$\arg\min_{\boldsymbol{Z}}\left\{\min_{\boldsymbol{\beta}}\left\{n^{-1}\|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2\right\}\right\} = \arg\max_{\boldsymbol{Z}}\left\{s_\lambda^{\text{ridge}}(\boldsymbol{Z},\boldsymbol{y})\right\}\,, \tag{3.5}$$

where s^{ridge} defines a particular quadratic association score between an outcome $\boldsymbol{y}$ and a set of filters $\boldsymbol{Z}$:

$$s_\lambda^{\text{ridge}}\left(\boldsymbol{Z},\boldsymbol{y}\right) := \boldsymbol{y}^T\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\left[\left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\right)^T\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} + \lambda n\boldsymbol{I}_q\right]^{-1}\left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\right)^T\boldsymbol{y}\,. \tag{3.6}$$

Strictly speaking, since (3.3) is a non-convex joint objective in $(\boldsymbol{Z}, \boldsymbol{\beta})$, its solution can differ from the solution of (3.5). It is indeed possible for the joint minimum to be different from the minimum in $\boldsymbol{Z}$ of the minimum in $\boldsymbol{\beta}$.

It nonetheless formalizes the training of a CNN as the selection of a set of filters whose association with the phenotype $\boldsymbol{y}$ in the sense of s_λ^{ridge} is maximal.

We adopt the concept of association score from (Slim et al., 2019), albeit the conditions we will set in Chapter 4 for a given association score to be employed in SEISM deviate slightly from the description provided by the authors. Introducing the association scores allows us the generalize the different results that follow, particularly in Chapter 4. Indeed, the developed methodology may be applied as long as such an association score between the phenotype and explanatory features of interest can be defined.

• Other association scores

In a way, association scores assess the relationship between a phenotype and an explanatory feature. We can then use different scores to measure different types of relationships. To begin, we can observe that if $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$ is centered:

$$\lim_{\lambda \to \infty} \lambda n \times s_\lambda^{\text{ridge}}(\boldsymbol{Z}, \boldsymbol{y}) = \boldsymbol{y}^T \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} (\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}})^T \boldsymbol{y} := s^{\text{HSIC}}(\boldsymbol{Z}, \boldsymbol{y}) \,, \tag{3.7}$$

so for large values of the regularization hyperparameter, selecting filters by learning a CNN is equivalent to selecting filters with the classical Hilbert-Schmidt Independence Criterion (HSIC) score (Song et al., 2012), a widely used criterion for feature selection.

Other scores can be implemented by modifying the way the representation $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$ is linked to $\boldsymbol{y}$, or by changing the way the data is represented by using alternative representation functions φ. For instance, (Ditz et al., 2022) extend CNNs to include positional information in φ, thus enabling to derive an association score between a phenotype and (motif, position) couples.

3.2 The activation function — measuring the presence of a motif in a sequence

In Chapter 1, we introduced the spectrum kernel for biological sequences (1.5). This representation basically counts the number of occurrences of existing k-mers in a given sequence. But we then see that we cannot apply the exact same strategy with sequence motifs: the occurrence of a motif in a sequence is not well defined. Different literatures employ different strategies to determine the extent to which a motif is present in a sequence. These methods are all based on a comparison of the motif with the k-mers $\boldsymbol{u}$ of the sequence.

3.2.1 Comparing a motif and a k-mer

On the one hand, most bioinformatic tools — such as the MEME suite (Bailey et al., 2015), an integrated set of tools for studying sequence motifs in biological sequences considered as state-of-art — rely on a **categorical model**, where the weight of a motif at a given position can be directly interpreted as the probability to find the corresponding letter in this position in a k-mer. One can then compute the presence of a motif in a sequence using a pooling step (either max or mean pooling), resulting in the following representation for sequences $\boldsymbol{X}$ using motifs $\boldsymbol{Z}$:

$$\forall (i,j)\,,\ [\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}_{\text{cat}}]_{(i,j)} = \text{Pooling}_{\boldsymbol{u}\in x_i}\left(\prod_{\ell=1}^{k} \boldsymbol{u}_\ell^T \boldsymbol{Z}_{(j,\ell)}\right). \tag{3.8}$$

On the other hand, the standard machine learning methods for biological sequences, for instance typical CNNs, rely on exponential activation functions:

$$\forall (i,j)\,,\ [\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}_{\text{exp}}]_{(i,j)} = \text{Pooling}_{\boldsymbol{u}\in x_i}\left(e^{\frac{1}{\omega^2}(\boldsymbol{u}^T \boldsymbol{Z}_j - 1)}\right), \tag{3.9}$$

for some bandwidth parameter ω.

In this work, we will use a slightly modified version of this activation function and rely on a Gaussian activation, where the probability distribution defined by a motif can be represented by a Gaussian over $\mathcal{Z}$, similarly to Figure 1.4. This activation empirically results in better selection performance compared to the exponential activation:

$$\forall (i,j)\,,\ [\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}_{\text{gaus}}]_{(i,j)} = \text{Pooling}_{\boldsymbol{u}\in x_i}\left(e^{-\frac{\|\boldsymbol{Z}_j - \boldsymbol{u}\|_2^2}{2\omega^2}}\right), \tag{3.10}$$

and the impact of ω will be studied in Subsection 3.3.5. As all the k-mers $\boldsymbol{u}$ have the same norm $\|\boldsymbol{u}\|^2 = k$, we can note working with normalized motifs $\boldsymbol{Z}_j$ would make (3.9) and (3.10) equivalent up to a constant factor. But using $\boldsymbol{z} \in \mathcal{Z}$ leads to slightly different results.

Furthermore, in contrast to standard CNNs, we will use a centered version of the representation $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} = \boldsymbol{C}_n \tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}$, where $\boldsymbol{C}_n = \boldsymbol{I}_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^T$ is the centering operator, $\boldsymbol{I}_n$ the identity matrix and $\mathbf{1}_n$ the all-one vector in $\mathbb{R}^n$.

In addition to connecting s^{ridge} with s^{HSIC} as discussed in Section 3.1, we observed that this centering led to the selection of more relevant sequence motifs in our experiments, as it allows SEISM to work with skewed data, since imbalanced classes will have no effect on the result.

We observe that the centering matrix is an orthogonal projection matrix onto $\mathcal{E} := \text{Range}(\boldsymbol{C}_n)$, the orthogonal of the vector line generated by the vector $\mathbf{1}_n$, and then it holds:

$$\|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\|_2^2 = \|\boldsymbol{C}_n\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\|_2^2 + \|\boldsymbol{y} - \boldsymbol{C}_n\boldsymbol{y}\|_2^2\,. \tag{3.11}$$

The solution of (3.3) is unchanged if $\boldsymbol{y}$ is replaced by $\boldsymbol{C}_n\boldsymbol{y}$, and so we can assume that $\boldsymbol{y} \in \mathcal{E}$ without any generality loss. In practice, $\boldsymbol{y}$ is centered at the very beginning of the SEISM procedure.

3.2.2 Pooling strategies

When using CNNs for prediction performance, max pooling and mean pooling are two methodologies that, depending on the task, can both produce good results. As a result, the two strategies are often implemented and left at the discretion of the user, and we began to work with these two options:

$$[\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}_{\text{mean}}]_{(i,j)} = \frac{1}{|x_i|}\sum_{\boldsymbol{u}\in x_i}\left(e^{-\frac{\|\boldsymbol{Z}_j-\boldsymbol{u}\|_2^2}{2\omega^2}}\right) \text{ and } [\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}_{\max}]_{(i,j)} = \max_{\boldsymbol{u}\in x_i}\left(e^{-\frac{\|\boldsymbol{Z}_j-\boldsymbol{u}\|_2^2}{2\omega^2}}\right) \tag{3.12}$$

However, it turns out that max pooling is superior when it comes to *de-novo* motifs discovery. Indeed, mean pooling tends to select homopolymers: motifs that look like repeated strings of one or two letters, as shown in Figure 3.1. This issue, which is not addressed in standard machine learning approaches because it does not negatively impact the prediction performance, has long been known in the bioinformatics literature (Bailey & Elkan, 1994).

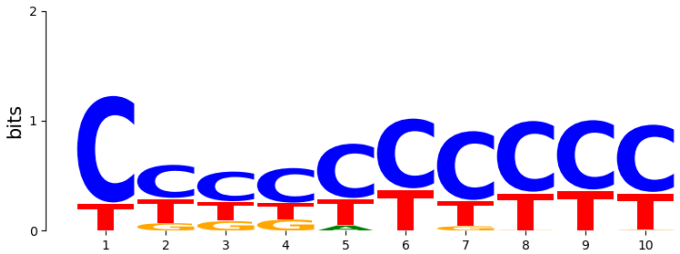


Figure 3.1: Motif discovered using SEISM with mean pooling on a simulated dataset with no signal (the phenotype was pure noise). The discovery of this homopolymer reveals a minor imbalance in the C content of the positive and negative sequences.

In our framework, this issue arises when a given nucleotide sequence content in $\boldsymbol{X}$ correlates (even slightly) with the phenotype $\boldsymbol{y}$. A prediction based on this feature would have very low generalization performance, as from a biological point of view this feature is irrelevant for most phenotypes. While they might sometimes be relevant, for instance the GC content can correlate with biological properties (Galtier & Lobry, 1997), they are most often pointless.

This problem does not occur with max pooling strategies since they do not evaluate the nucleotide content at the sequence scale. As a result, they achieve superior selection performance.

3.3 Optimizing an association score to select sequence motifs

Both s^{HSIC} and s^{ridge} scores with the Gaussian activation function are non-convex functions of $\boldsymbol{Z}$. The search for a global maximum is thus not straightforward, and in the course of this thesis, we studied three approaches for optimizing the association score.

The first one makes use of a reformulation of our optimization problem as a difference of convex functions. There exists a literature dedicated to this class of problems, and it has already been applied in a setting similar to ours. The second one consists in convexifying the problem, by generalizing it on a richer class of functions. Finally, the most standard approach relies on gradient descent algorithms to identify local optima. Each approach has its own advantages and trade-offs, and SEISM is based on this latter method.

3.3.1 Difference of convex functions

When working with s^{HSIC}, the exponential activation and only one filter $\boldsymbol{z}$, one can note that optimizing the objective function is equivalent to optimizing a difference of convex functions:

$$\begin{aligned}
\arg\max_{\boldsymbol{z}} \; s^{\text{HSIC}}(\boldsymbol{z}, \boldsymbol{y}) &= \arg\max_{\boldsymbol{z}} \left(\boldsymbol{y}^T \boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},\boldsymbol{X}} (\boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},\boldsymbol{X}})^T \boldsymbol{y} \right) \\
&= \arg\max_{\boldsymbol{z}} \left(\left((\boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},\boldsymbol{X}})^T \boldsymbol{y} \right)^2 \right) \\
&= \arg\max_{\boldsymbol{z}} \left((\boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},\boldsymbol{X}} i)^T \boldsymbol{y} \right) \quad \text{(Assumption: } (\boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},\boldsymbol{X}} i)^T \boldsymbol{y} > 0) \\
&= \arg\max_{\boldsymbol{z}} \left(\sum_{y_i>0} y_i \boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},x_i} - \sum_{y_i<0} |y_i| \boldsymbol{\varphi}_{\exp}^{\boldsymbol{z},x_i} \right) \\
&= \arg\min_{\boldsymbol{z}} \left(\sum_{y_i<0} |y_i| e^{\boldsymbol{z}^T x_i} - \sum_{y_i>0} y_i e^{\boldsymbol{z}^T x_i} \right) \\
&= \arg\min_{\boldsymbol{z}} \left(g(z) - h(z) \right) ,
\end{aligned} \tag{3.13}$$

where $g : \boldsymbol{z} \mapsto \sum_{y_i<0} |y_i| e^{\boldsymbol{z}^T x_i}$ and $h : \boldsymbol{z} \mapsto \sum_{y_i>0} y_i e^{\boldsymbol{z}^T x_i}$ are both convex functions of $\boldsymbol{z}$, and $\mathcal{Z}$ is a convex subset of $\mathbb{R}^{|\mathcal{A}| \times k}$.

Optimizing the association score over the convex set $\mathcal{Z}$ is therefore equivalent to minimizing the function $f = g - h$, which belongs to the class of *DC functions*, as introduced in Definition 3.3.1 following Horst & Thoai (1999):

Definition 3.3.1 (DC functions)**.**
Let Ω be a convex subset of $\mathbb{R}^d$. A real-valued function $f : \Omega \to \mathbb{R}$ is called DC on Ω, if there exist two convex functions $(g\,, h) : \Omega \to \mathbb{R}$ such that f can be expressed in the form:

$$f = g - h\,.$$

From this definition, the authors derive the following optimality condition:

Proposition 3.3.1 (Optimality condition for DC functions)**.**
A point $\boldsymbol{z}^*$ *is an optimal solution* $\boldsymbol{z}^* = \arg\min_z (g(z) - h(z))$ *if and only if there is* $t^* \in \mathbb{R}$ *such that:*

$$0 = \inf \{-h(\boldsymbol{z}) + t \ : \ \boldsymbol{z} \in \mathcal{Z}, t \in \mathbb{R}, g(\boldsymbol{z}) - t \leq g(\boldsymbol{z}^*) - t^*\} \ . \tag{3.14}$$

This formulation motivates a cutting plane algorithm, which iteratively identifies and add constraints to the problem, until the optimal solution is found. The main idea is to construct a sequence of nested polytopes $P^{i+1} \subseteq P^i$ which contain the optimal solution $(\boldsymbol{z}^*, t^*)$. Subsequent polytopes are then defined by cutting out the current vertex $(\boldsymbol{z}^i, t^i)$ while keeping the solution inside.

In particular, this algorithm was adapted by Argyriou et al. (2006) to perform kernel selection among a convex hull of a continuous parametrized family of kernels. Our framework also fits this definition, and the developed methodology can therefore theoretically be applied to our case.

However, Argyriou et al. (2006) apply their algorithm to find the best kernel in a family parametrized with only a few parameters, and describe a sharp increase of the required computation time with the number of parameters: on their dataset, it took between one and two minutes to select the best kernel when there was only one parameter, about five minutes to learn two parameters, and about one hour to learn four parameters. Because the number of parameters for a sequence motif of length k is $3 \times k$, this algorithm is way too slow to detect even very short motifs.

Nonetheless, considering our optimization problem with the HSIC score (3.13) as a DC problem, and particularly using the optimality condition provided by Proposition 3.3.1 could be interesting to determine whether a sequence motif is a global optimal for a given phenotype, which could be beneficial for the inference step, in particular to describe a selection event, see Chapter 4 Subsection 4.1.2. Working around this topic could thus be an interesting line of work to pursue.

3.3.2 Convexification

In this section, we will slightly modify our optimization objective (3.3) and rely on a LASSO rather that on the Ridge penalty:

$$\min_{(\boldsymbol{Z},\boldsymbol{\beta}) \in (\mathcal{Z}, \mathbb{R}^q)} n^{-1} \|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \ \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \ . \tag{3.15}$$

The idea of this section is to reparametrize the optimization problem: instead of trying to find filters $\boldsymbol{Z} = (\boldsymbol{z}_j)_{i \leq q}$ and weights $\boldsymbol{\beta}$ minimizing (3.15), we will define a richer problem and find a measure μ minimizing it. This new formulation indeed leads to a convex problem, facilitating the search for a global optimum. To our knowledge, this idea originally comes from the optimal transport literature, see for instance (Peyré & Cuturi, 2020, Chapter 2), in particular the Monge Problem and the Kantorovich relaxation. It

was recently leveraged by the machine learning community (Zhang et al., 2016; Bach, 2016).

In (3.3), the prediction function g can be expressed with parameters $\boldsymbol{Z}$ and $\boldsymbol{\beta}$:

$$g(x) = \boldsymbol{\varphi}^{\boldsymbol{Z},x}\ \boldsymbol{\beta} = \sum_{j=1}^{q} \boldsymbol{\beta}_j \boldsymbol{\varphi}^{\boldsymbol{z}_j,x}\,. \tag{3.16}$$

In this section, we introduce a prediction function h, that can be expressed using a measure $\mu \in \mathcal{M}(\mathcal{Z})$:

$$h(x) = \int_{\mathcal{Z}} \boldsymbol{\varphi}^{\boldsymbol{z},x} \mathrm{d}\mu(\boldsymbol{z})\,, \tag{3.17}$$

where $\mathcal{M}(\mathcal{Z})$ is simply the set of signed measures on $\mathcal{Z}$. We can then note that any prediction function of the form (3.16) can be expressed as a function of the form (3.17) using the following parametrization:

$$\mu = \sum_{i=1}^{q} \boldsymbol{\beta}_j \delta_{\boldsymbol{z}_j}\,, \tag{3.18}$$

where $\delta_{\boldsymbol{z}_j} : \mathcal{X} \to \mathbb{R}$ is the Dirac measure centered on $\boldsymbol{z}_j$. In this sense, (3.17) defines a richer class of functions than (3.16).

Similarly, we obtain the following mapping:

$$\boldsymbol{\varphi}^{\boldsymbol{Z},x} \mapsto \int_{\mathcal{Z}} \boldsymbol{\varphi}^{\boldsymbol{z},x} \mathrm{d}\mu(\boldsymbol{z}) = \Phi(\mu)\,. \tag{3.19}$$

We now introduce $F = (\mathcal{C}(\mathcal{Z}), \|\cdot\|_\infty)$ the normed vector space, with $\mathcal{C}(\mathcal{Z})$ the set of continuous real-valued functions on the compact set $\mathcal{Z}$, and:

$$\begin{aligned} \|\cdot\|_\infty : \mathcal{C}(\mathcal{Z}) &\to \mathbb{R} \\ f &\mapsto \sup_{\boldsymbol{z}\in\mathcal{Z}} |f(\boldsymbol{z})|\,. \end{aligned} \tag{3.20}$$

In particular, we will work on the topological dual space F^*, that is the set of continuous real-valued linear functions on F:

$$F^* = \left\{ \begin{array}{l} \rho : F \to \mathbb{R} \\ \quad f \mapsto \rho(f) \text{ linear} \end{array} \right\}\,. \tag{3.21}$$

Next, we will take advantage of the Riesz representation theorem, see for instance (Le Gall, 2006, Theorem 6.4.1):

Theorem 3.3.1 (Riesz representation theorem)**.**
Let $\rho \in F^*$ be a continuous linear function on $F = \mathcal{C}(\mathcal{Z})$. There exists a unique signed measure $\mu \in \mathcal{M}(\mathcal{Z})$ such that:

$$\forall f \in \mathcal{C}(\mathcal{Z})\,,\ \rho(f) = \int_{\mathcal{Z}} f \mathrm{d}\mu\,. \tag{3.22}$$

From this theorem, we obtain:

$$F^* = (\mathcal{C}(\mathcal{Z}), \|\cdot\|_\infty)^* = (\mathcal{M}(\mathcal{Z}), \|\cdot\|_1)\ , \tag{3.23}$$

with $\|\cdot\|_1$ the total variation, such that:

$$\|\mu\|_1 = \sup_{\|f\|_\infty \leq 1} \int_{\mathcal{Z}} f \mathrm{d}\mu. \tag{3.24}$$

Finally, this new parametrization leads us to a new formulation for (3.3):

$$\min_{\mu \in F^*} \frac{1}{n} \sum_{i=1}^{n} \left(\boldsymbol{y}_i - \int_{\mathcal{Z}} \boldsymbol{\varphi}^{\boldsymbol{z}, x_i} \mathrm{d}\mu(\boldsymbol{z}) \right)^2 + \lambda \|\mu\|_1 \,. \tag{3.25}$$

We can then recognize that this new formulation is very similar to the LASSO (see Chapter 2 Subsection 2.4.2). It corresponds actually to the Beurling LASSO (BLASSO) problem, introduced by Azais et al. (2014), and is a convex optimization objective for $\mu \in F^*$. Consequently, it admits a global minimum μ^*, and the L^1 penalization enforces this solution to be sparse:

$$\mu^* = \sum_{j=1}^{p^*} \boldsymbol{\beta}'_j \delta_{\boldsymbol{z}'_j} \,, \tag{3.26}$$

for some number of particles p^*, some weights $\boldsymbol{\beta} \in \mathbb{R}^{p^*}$ and some Dirac (atomic) measures $\delta_{\boldsymbol{z}'_j}$.

The Conic Particle Gradient Descent (CPGD) (Chizat, 2020) provides an efficient way to optimize this problem. It starts with an initial measure μ^0 described with a high number of particles (usually $p > n$) and leverages the gradient flows to modify the weights and atomic measures in order to achieve (3.25). This algorithm is illustrated in Figure 3.2.

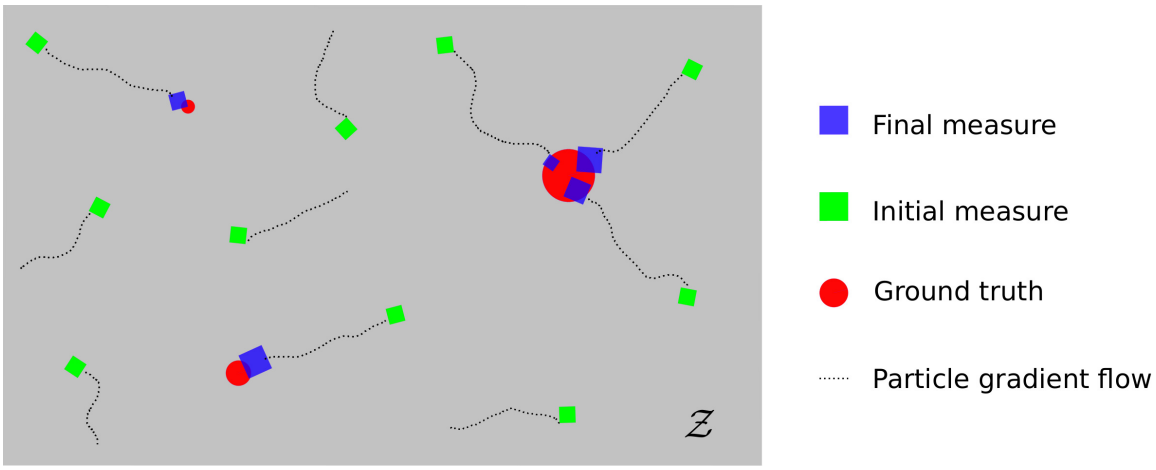


Figure 3.2: Illustration of the Conic Particle Gradient Descent algorithm. The size of a particle represents its weight $\boldsymbol{\beta}_j$. From a initial distribution of s particles with same weights, all particles follow their respective gradient flows. Some of them converge towards the optimal positions, while other vanish $\boldsymbol{\beta}_j \to 0$.

In addition to providing a global minimum, this approach allows to automatically find the right number of particles. That is, under some assumptions about the regularization, the number of particles (sequence motifs) obtained with this approach is identical to what

we would have obtained by optimizing the non-convex problem (3.15). In comparison, a classical CNN will always find a fixed and predetermined number of filters.

Another possibility for solving (3.25) is to rely on the Sliding Frank-Wolfe algorithm, introduced by Denoyelle et al. (2018): a greedy first-order optimization for constrained convex optimization. At each step, this algorithm performs a linear approximation of the objective function, and looks for a minimizer of this linear function. While the problem it solves is convex, it relies on non-convex steps.

While it contains non-convex steps, and therefore is not guaranteed to converge towards the global optimum, this latter algorithm does not suffer from a curse of dimensionality, unlike CPGD, which requires a growing initial number of particles.

To conclude, this direction emerged at an advanced stage of this thesis, and the focus was already on the next component of SEISM (the inference), so we did not have time to implement it yet. Nonetheless, it seems to be a promising direction and its implementation constitutes a future work.

3.3.3 (Stochastic) gradient descent with line search

While they may only find local optimizers instead of global ones, gradient-based strategies are traditionally used for neural networks, leading to good results for prediction. It is then common to solve (3.3) by gradient descent over the filters. As described in Chapter 1, the stability of the explanation is an important criterion, that's why we prefer to work with a full gradient strategy rather than a stochastic one. This choice prevents the use of SEISM on large datasets. While SEISM was initially developed for small-scale datasets, it will later be interesting to extend it to bigger datasets, and implementing stochastic gradient descent approaches could be a future work.

During the optimization, we work on a less constrained set than $\mathcal{Z}$, defined in (3.1), and don't enforce the positivity constraint, resulting in the following vectorial space:

$$\mathcal{Z}_{\text{uc}} = \left\{ \boldsymbol{z} \in \mathbb{R}^{|\mathcal{A}|\times k} \;:\; \forall j \in \leq k\,, \sum_{i=1}^{|\mathcal{A}|} \boldsymbol{z}_{i,j} = 1 \right\}. \tag{3.27}$$

One can note that the gradients of both s^{HSIC} and s^{ridge} with respect to $\boldsymbol{z}$ belong to $\mathcal{Z}_{\text{uc}}$. Any point obtained by taking a step in the gradient direction from a point in $\mathcal{Z}_{\text{uc}}$ also belongs to this vectorial space. It is therefore not necessary to use projected gradient strategies in this framework.

At the end of this optimization step, the resulting point is projected onto the simplex $\mathcal{Z}$ using an orthogonal projection, according to the algorithm described in Duchi et al. (2008).

The point picked for the initiation of this gradient descent is another element that might result in instability in the selected motifs. Indeed, common strategies that randomly initialize the filters might produce varying outcomes from one run to the next, as the gradient descent strategy is not guaranteed to find the global minimum. To tackle this issue, we propose to initialize the filter at the k-mer with the best association score. To

that end, all the k-mers contained in $\boldsymbol{X}$ are enumerated using the DSK software (Rizk et al., 2013), which provides an efficient method to do so. We restrict the search for the best k-mer to the 5% most prevalent k-mers in $\boldsymbol{X}$ to avoid using up too much memory. Empirically, the best k-mer appears to typically be in this subset.

Moreover, we rely on a backtracking line search approach with Armijo-Goldstein stopping criterion for the optimization (Armijo, 1966). It is a simple way to adaptively choose the step size and the number of iterations, that tends to work well in practice, see Algorithm 1.

Algorithm 1 Backtracking line search

`/* Description: The line search algorithm determines the amount to move along the gradient's direction in order to rapidly reach a maximum. */`

Inputs: *Association score s, phenotypes $\boldsymbol{y}$, motif $\boldsymbol{z}^{(i)}$ obtained after optimization step i and $\nabla_{\boldsymbol{z}} s(\boldsymbol{z}^{(i)}, \boldsymbol{y})$ the gradient of the score evaluated in $\boldsymbol{z}^{(i)}$. Hyperparameters $(c, \tau) \in (0,1)^2$: a control parameter and a diminution factor. The initial step size α_0.*

Result: The new motif $\boldsymbol{z}^{(i+1)}$ obtained after optimization step $i+1$.

$j \leftarrow 0$
$\alpha_j \leftarrow \alpha_0$
while $\left(s\left(\boldsymbol{z}^{(i)} + \alpha_j \nabla_{\boldsymbol{z}} s(\boldsymbol{z}^{(i)}, \boldsymbol{y}), \boldsymbol{y}\right) - s(\boldsymbol{z}^{(i)}, \boldsymbol{y})\right) \leq \alpha_j \times c \times \|\nabla_{\boldsymbol{z}} s(\boldsymbol{z}^{(i)}, \boldsymbol{y})\|^2$ **do**
  `/* The shrinking of the step size` α_j `continues until a value provides an increase in the objective function that matches the increase expected to be achieved based on the gradient. */`
  $j \leftarrow j + 1$
  $\alpha_j \leftarrow \tau \times \alpha_j$
end

3.3.4 Reverse complements

The DNA has a well-known double-helical structure, made up of two interwoven strands. Each strand can be thought of as an oriented chain made up of the nucleotides $\{A, C, G, T\}$. The nucleotides of the two strands interact deterministically at each site since only the pairings of the nucleotides (A, T) and (G, C) can occur. By complementing each nucleotide and reversing the order, we can determine the so-called reverse complement (RC) sequence of a strand, see Figure 3.3. During the sequencing step, only one of those two strands is randomly chosen and sequenced. Therefore, any DNA sequence may be represented equally by two RC sequences, motivating the creation of methods with RC equivalence (Mallet & Vert, 2021; Zhou et al., 2022).

To this end, the representation function of SEISM is slightly modified, in order to compare the motifs with the k-mers and their reverse complements contained within a sequence:

$$\left[\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}\right]_{(i,j)} = \max_{u \in x_i} \left(\max \left(e^{-\frac{\|\boldsymbol{Z}_j - \boldsymbol{u}\|_2^2}{2\omega^2}}, e^{-\frac{\|\boldsymbol{Z}_j - \bar{\boldsymbol{u}}\|_2^2}{2\omega^2}} \right) \right), \tag{3.28}$$

where $\bar{\boldsymbol{u}}$ is the RC version of $\boldsymbol{u}$.

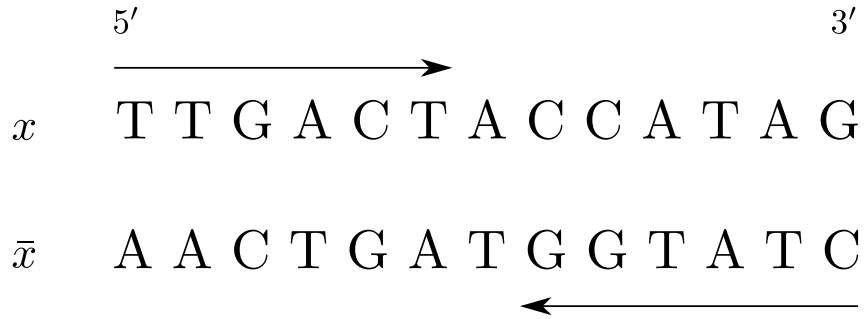


Figure 3.3: A sequence x and its reverse complement $\bar{x}$.

3.3.5 Adaptive length selection

As described in Equation (3.10), the data representation $\varphi_{\text{gaus}}^{\boldsymbol{Z},\boldsymbol{X}}$ used by SEISM depends on a hyperparameter ω controlling the bandwidth of the Gaussian non-linearity: $\exp\left(-\frac{\|\boldsymbol{z}-\boldsymbol{u}\|_2^2}{2\omega^2}\right)$. Assuming that the positions within a k-mer are independent, the expected value of the squared distance between a motif $\boldsymbol{z}$ and a k-mer $\boldsymbol{u}$ with length k is then proportional to k.

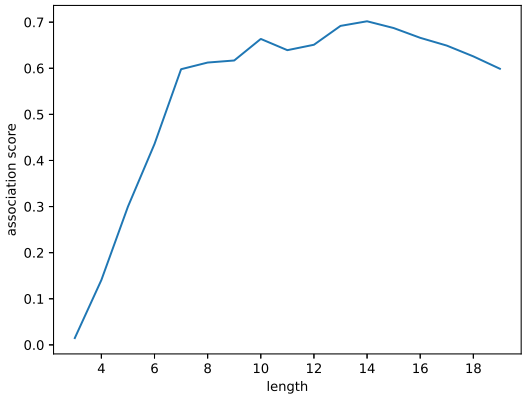
In order to get an activation that does not depend on the length of the motifs, we simply set ω to be proportional to $\sqrt{k}$. From empirical tests, we set $\omega = \frac{\sqrt{0.9k}}{2}$ to achieve good selection performance by choosing the motif that maximizes the association score among the set of possible lengths, as described in Figure 3.4.

To create this figure, we constructed a simulated dataset, with one true motif (Figure 3.4b top row) associated with the phenotype. This motif was randomly inserted in the sequences of the dataset. We then applied the SEISM procedure to select the motifs from length 3 to 19 (Figure 3.4b bottom row). Figure 3.4a summarizes the association scores of those different motifs.

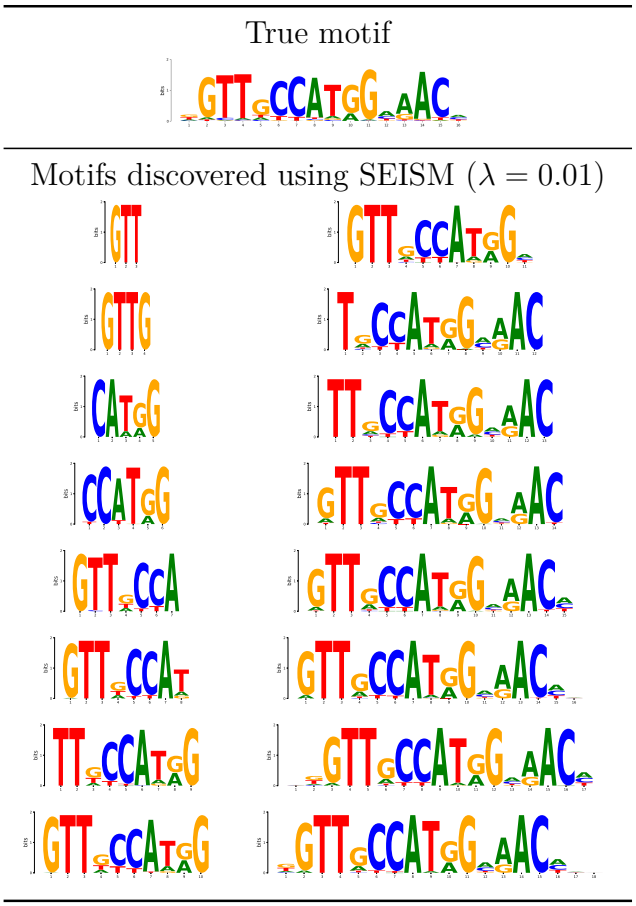
3.4 From a joint to a greedy optimization

The previous section focused on optimizing the association score for only one motif $\boldsymbol{z}$. We will now see how SEISM performs the selection of several motifs. Standard CNNs indeed commonly optimize the objective function jointly over the q filters. More generally, this approach for training a neural network with a single, large hidden layer is known to find a global optimizer at the large q limit under some assumptions (Soltanolkotabi et al., 2019). While it leads to very good performance for prediction, our objective here is slightly different: we do not necessarily aim at approximating a continuous measure with a large number of particles, but we aim at selecting a small number of particles lending themselves to interpretation. Furthermore, the number of relevant motifs for a given dataset is generally unknown. As discussed in Chapter 1, jointly optimizing the convolution filters leads to irrelevant motifs, with some actual motif split along several filters and other duplicated, as illustrated in Figure 3.5.

A possible strategy is to forego filter-level interpretation, train an over-parametrized net-



(a) Association score vs. motif length (obtained with $\lambda = 0.01$).



(b) Motifs obtained with length ranging from 3 to 19.

Figure 3.4: Using an adaptive $\omega = \frac{\sqrt{0.9k}}{2}$ allows to select the motif with the right length from several possible ones. While the true motif has length 16, the two extreme positions are uninformative, and therefore the selection of the motif with length 14 is adequate. Adding highly informative positions leads to a rapid growth of the association score (lengths 3 to 7) while adding non-informative positions degrades this score (length 11 and lengths 15 to 19).

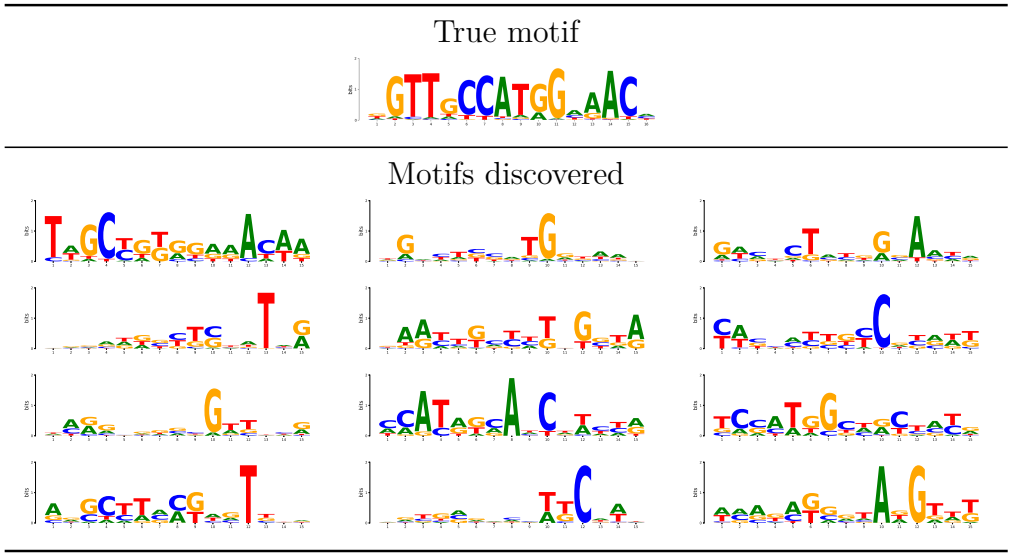


Figure 3.5: Motifs obtained by training a convolutional neural network (CKN) with 12 filters on a simulated with a known true motif. One can observe redundancies, partial or non-informative motifs.

work — with a much larger q than the expected number of motifs — and use attribution methods to extract relevant motifs from the trained network (Subsection 1.4.4).

Another possibility, coming from the bioinformatics literature, is to perform a greedy optimization: selecting the motifs one by one. For instance, once a motif has been selected during step j, STREME (Bailey, 2021) *erases* the motif from the dataset: in a given sequence, it will remove all the k-mers corresponding to the motif (k-mers with a Euclidean distance to the motif below some threshold). But this strategy requires the use of an additional hyperparameter (the threshold).

Of note, forward selection procedures over finite sets of features work around the problem by iteratively removing the selected elements from the set over which the selection is performed (Slim et al., 2019). Such a strategy is not suited to our framework, where the selection is performed over a continuous set of motifs.

With SEISM, we adopt a different strategy. Similarly to bioinformatics methods, we use a forward stepwise procedure. A first naive approach would then be to select at each of the q steps the motif that maximizes the overall score:

$$\boldsymbol{z}_j = \operatorname*{arg\,max}_{\boldsymbol{z} \in \mathcal{Z}} s\left([\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{j-1}, \boldsymbol{z}], \boldsymbol{y}\right) . \tag{3.29}$$

But one can note that iterating (3.29) using s^{HSIC} would return the same motif $\boldsymbol{z}$ at each step. With $\boldsymbol{Z} = (\boldsymbol{z}_j)_{j \leq q}$, we can indeed observe:

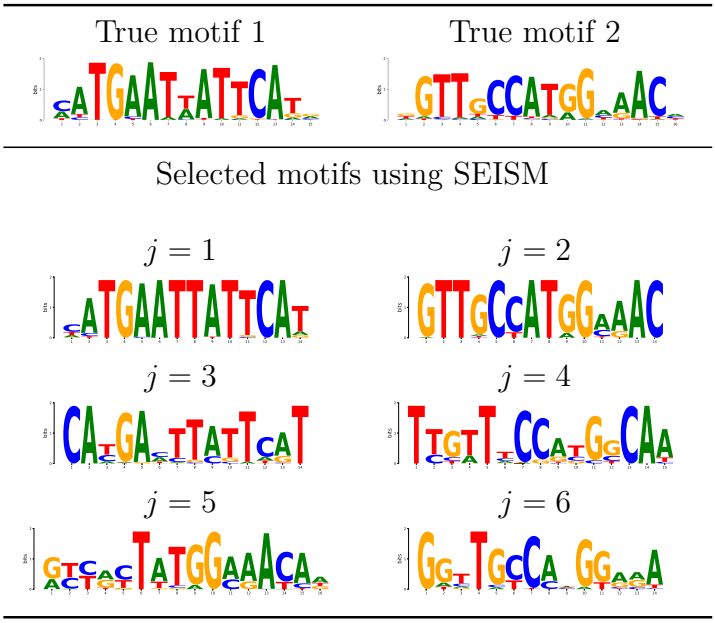


Figure 3.6: Motifs selected using SEISM's greedy procedure with $q = 6$ on a simulated with a phenotype determined by two *true* motifs. The two first discovered motifs ($j = 1$ and $j = 2$) correspond to the two true motifs, while the following ones are less similar and less informative.

$$\begin{aligned} s^{\text{HSIC}}(\boldsymbol{Z}, \boldsymbol{y}) &= \boldsymbol{y}^T \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\right)^T \boldsymbol{y} \\ &= \| \left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\right)^T \boldsymbol{y} \|_2^2 \\ &= \sum_{j=1}^{q} \left(\left(\boldsymbol{\varphi}^{\boldsymbol{z}_j,\boldsymbol{X}}\right)^T \boldsymbol{y} \right)^2 . \end{aligned}$$

The score s^{ridge} introduces some interactions between the filters thanks to the inverse term (3.6) but does not enforce a sufficient separation of the motifs.

To work around this issue, instead of modifying the sequences by removing the already selected motif sites, SEISM iteratively optimizes each of the convolution filters over the residual error left by the previous ones.

More precisely at each of the q steps, we select $\boldsymbol{z}_j$ such that:

$$\boldsymbol{z}_j = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}) \,, \tag{3.30}$$

where $\boldsymbol{P}_j$ is the projection operator onto the orthogonal of the subspace $\underset{\ell<j}{\text{Span}} \left\{ \mathbf{1}, \boldsymbol{\varphi}^{\boldsymbol{z}_\ell,\boldsymbol{X}} \right\}$.

This is how $\boldsymbol{z}_j$ is optimized over the residuals of the previous filters. The vector $\mathbf{1}$ enforces that we project $\boldsymbol{y}$ on a subspace of $\mathcal{E}$, in particular $\boldsymbol{P}_1 = \boldsymbol{C}_n$.

This strategy enables a solid separation of the selected motifs, as illustrated in Figure 3.6.

3.5 *De-novo* motif discovery

De-novo motif discovery tools are algorithms that aim to identify motifs in biological sequences. They differ from other approaches in that they do not rely on any prior knowledge about the motifs being searched for, but rather rely on statistical tools. Here is a quick summary of the various strategies used by some different tools.

In particular, these methods are widely used for motif discovery in Transcription Factor (TF) ChIP-seq datasets. ChIP-seq (chromatin immunoprecipitation sequencing, Robertson et al., 2007) is a technique for examining how proteins interact with DNA. In a nutshell, it first links the DNA with proteins (and other molecules, such as RNA), then breaks the resulting complex (the chromatin) into small fragments. Third, it leverages specific antibodies, that recognize the transcription factors of interest, to extract only the fragments containing DNA sequences that are bound with transcription factors, and finally sequences those small sequences.

A transcription factor is a protein that, by binding to a particular DNA sequence, regulates the rate at which genetic information is transcribed from DNA to messenger RNA, and therefore regulates the expression of genes. There are about 1 500 of them in the human genome (Vaquerizas et al., 2009).

A *binding motif* is then a particular group of DNA sequences that a TF prefers to bind to. TFs have a variety of binding affinities for the sequences that make up their set of binding motifs. While the TFs have been largely identified, it is still unclear which sequences they can recognize. That is why algorithms able to detect those motifs from ChIP-seq datasets are useful.

Among the different existing algorithms, we can cite:

• **MEME:** Multiple Expectation maximization for Motif Elicitation (Bailey & Elkan, 1995) is an unsupervised algorithm for discovering enriched motifs in a set of sequences. To that end, it looks for maximum likelihood estimates of the parameters of a mixture model — made up of a given background distribution and the categorical model for generating k-mers at some positions (3.31) — with respect to the dataset, using an expectation maximization technique. It discovers motifs in a greedy way by incorporating information about the motifs already discovered into the current model to avoid selecting the same motifs again.

• **Weeder** (Pavesi et al., 2004) makes use of a data structure called a generalized suffix tree (Figure 3.7) to identify all k-mers of some lengths occurring with a given number of errors. To that end, it enumerates the paths of the tree and weeds out the ones that are unlikely to contain the k-mer. Those k-mers might be aggregated as sequence motifs in a second step.

• **SMILE** (Marsan & Sagot, 2000) is an unsupervised method looking for k-mers that are present — up to a given number of mismatches (therefore being similar to motifs) — in more than a user-defined number of sequences. To that end, it also makes use of a suffix tree.

• **HOMER** (Heinz et al., 2010) is a supervised algorithm. It selects the top k-mers whose enrichment are particularly high in the positive dataset compared to the negative one,

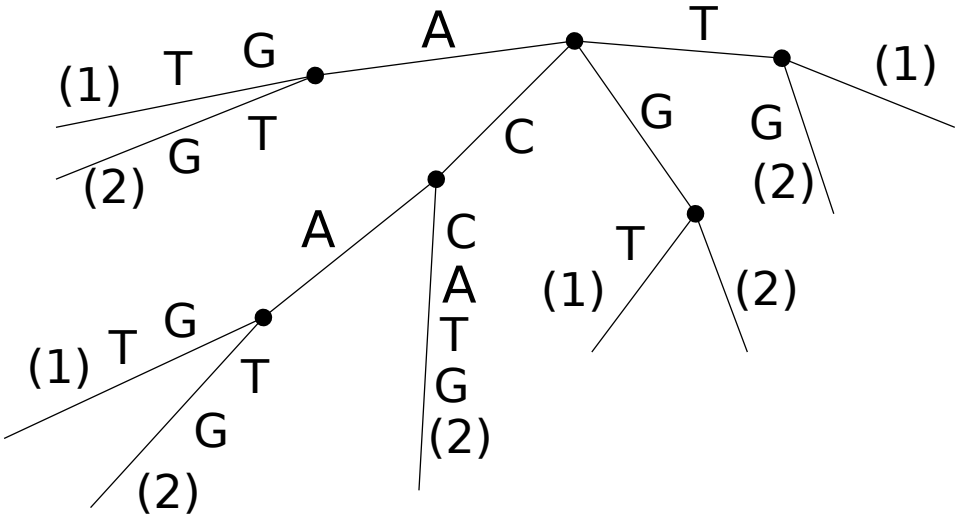


Figure 3.7: Generalized suffix tree for sequences CAGT and CCATG, with termination markers (1) and (2) respectively. To search for a k-mer in the set of sequences, for instance CAT, we start at the root, choose branch C, then branch A and then the leave starting with the T. On the contrary, TAG is not present in the sequences: starting with root and branch T, no following branch starting with an A exist in the tree. By keeping track of the numbered of encountered mismatches on each path, one can also search for a k-mer with a given number of allowed mismatches.

and converts them into motifs. It also uses the generalized suffix tree structure to speed up the counting.

• **STREME** (Bailey, 2021) is also a supervised method, that maximizes a Fisher association score between the presence of a motif and the binary class of sequences. It also derives valid p-values for the selected motifs using a data-split strategy.

In order to compare the SEISM selection performance with existing motif discovery algorithm, we followed the methodology described in Bailey (2021). This work introduces the STREME algorithm, establishing a new state-of-the-art method for *de-novo* motifs discovery tasks. The authors compare STREME with several widely used algorithms, including the majority of the aforementioned algorithms.

The evaluation method proposed in (Bailey, 2021) for those algorithms makes use of 40 TF ChIP-seq experiments in K562 cells (a human myelogenous leukemia cell line) provided by the ENCODE project (ENCODE Project Consortium, 2004). Each dataset contains between 1 233 and 56 058 sequences of 100bp regions. A known TF motif binding site is associated to each dataset, derived using completely independent assays (Jolma et al., 2013). All the sequences are associated with positive labels, as they all interact with the associated TF, and negative sequences are generated by shuffling the positive sequences.

This comprehensive comparison of those different approaches highlights that MEME is the most accurate method for small-scale datasets, while STREME outperforms other tools for middle- to large-scale datasets. In the following, we will therefore only compare SEISM with MEME and STREME.

3.6 Results comparison

To benchmark SEISM selection performance, we rely on the 40 datasets described in Subsection 3.5 and we test three sample sizes $n \in \{50, 100, 500\}$. For each dataset, $n/2$ sequences are sampled (the positive sequences) and then shuffled to create the negative sequences. For MEME, the unsupervised method, the negative sequences are useless. But MEME constructs a background model using the positive sequences, which is almost the same in this situation. In the end, 100 subsets are created per initial dataset.

We set up STREME, MEME and SEISM to select 5 sequence motifs. SEISM is run with $\lambda = 0.01$. We add CKN-seq (the neural network leveraging kernel methods and CNNs, described in Chapter 1 Subsection 1.2.3.2) to the comparison, to visualize the effects of the modifications we applied to convolutional networks with SEISM. It is parametrized to jointly optimize 128 filters (such networks notoriously lead to poor performances when a few filters are used).

We measure the accuracy of all the methods by comparing the motifs they discover with the known motif corresponding to the transcription factor binding site $\boldsymbol{m}^*$. We rely on the TOMTOM method (Gupta et al., 2007), which quantifies the probability that the Euclidean distance between a random motif and $\boldsymbol{m}^*$ is lower than the distance between the discovered motif and $\boldsymbol{m}^*$. Put simply, it gauges how close the discovered motif is to $\boldsymbol{m}^*$. More precisely, for each method we use the lowest TOMTOM p-value between the known TF binding site motif $\boldsymbol{m}^*$ and any of those discovered by the method. The TOMTOM score is then defined as $-\log_{10}$ of this p-value. We define the accuracy of a method as the proportion of experiments where the TOMTOM score between its best match and the true TF binding site motif was higher than some threshold.

Figure 3.8 (left column) shows that SEISM is as good as, if not superior to STREME at detecting sequences motifs with a threshold for TOMTOM p-values at 0.01, for any dataset size. It also performs better than MEME, except for small-scale datasets. It should be noted here that unlike STREME and MEME, SEISM provides a statistical significance for the discovered motifs without resorting to data-split strategies. This means that if we want to obtain valid p-values for the motifs discovered with STREME or MEME, we will have to set aside part of the sequences during the selection step, reducing their performance. The one-layer CNN with jointly optimized filters performs poorly in these experiments, emphasizing the importance of greedy optimization for accurate motif selection. This results confirm that the filters learned by a CNN do not correspond to very relevant motifs, and highlights the need for specific methods, such as TF-MoDISco, as discussed in Chapter 1 Subsection 1.4.2.

Figure 3.8 (right column) shows that SEISM performs slightly worse than STREME and MEME for high thresholds on TOMTOM scores. This suggests that the motif $\boldsymbol{z}$ that SEISM identifies is close enough to the PWM corresponding to the true motif, but farther away than the matrices identified by STREME or MEME. This discrepancy reflects a different usage of $\boldsymbol{z}$ for the parametrization of the k-mers distribution. This will be the focus of the following section. In practice, we observe that for a given dataset, the p-values of the best motifs discovered by SEISM and STREME/MEME are not separated by more than 2 orders of magnitude, which leads to minor differences in the motifs, as illustrated

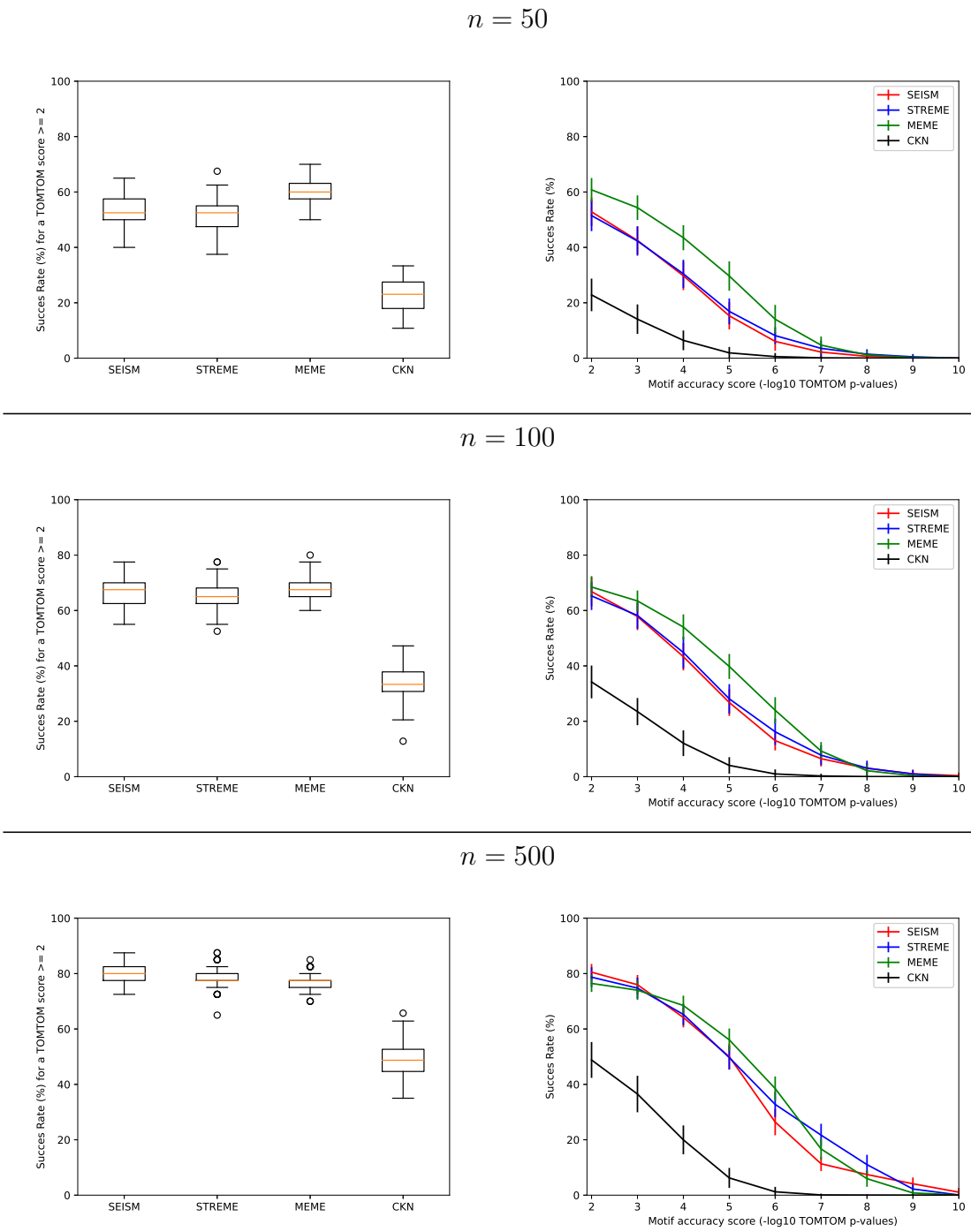


Figure 3.8: **Left:** Proportion of datasets where the true motif was detected by the designated algorithm. A true motif is said to be detected if its highest TOMTOM score with the discovered motifs is greater than 2. **Right:** Accuracy of motif discovery algorithms on ENCODE TF ChIP-seq datasets. The curves represent the proportion of ChIP-seq datasets were the best motif identified by the designated algorithm has a TOMTOM score greater than x.

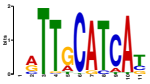
Reference motif *ATV_DBD*	SEISM motif p-value= 10^{-6}	STREME motif p-value= 3×10^{-8}
		

Figure 3.9: Comparison between two discovered sequence motifs by SEISM or STREME, and the true motif *ATV_DBD*.

in Figure 3.9.

3.7 Discussing the different models

The discovered motifs can be interpreted as representing a distribution of k-mers at the transcription factor binding site. As discussed in Subsection 3.2, both STREME and MEME rely on a categorical model, whereby the matrix $\boldsymbol{z}$ directly defines the probability of observing each letter at each of the k positions:

$$\forall(\boldsymbol{u}, \boldsymbol{z} \in \mathcal{Z})\,,\ \mathcal{L}_{\text{cat}}(\boldsymbol{u}; \boldsymbol{z}) = \prod_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{z}_i\,. \tag{3.31}$$

SEISM, on the other hand, is based on the Gaussian model. Through representation (3.10), $\boldsymbol{z}$ is meant to maximize the Gaussian likelihood of a set of k-mers, *i.e.*

$$\forall(\boldsymbol{u}, \boldsymbol{z} \in \mathcal{Z})\ \ \mathcal{L}_{\text{gaus}}(\boldsymbol{u}; \boldsymbol{z}) = C \prod_{i=1}^{k} e^{-\frac{\|\boldsymbol{u}_i - \boldsymbol{z}_i\|^2}{2\omega^2}}\,, \tag{3.32}$$

where C is a constant such that the sum of probabilities over $\mathbb{R}^{|\mathcal{A}| \times l}$ equals 1.

We now illustrate on a simple example how the same distribution of k-mers can be parametrized by different matrices under the different models. To build an easy example, we focus on k-mers of length 1, with the following distribution:

$$P(A) = 0.3\ \ P(C) = 0.4\ \ P(G) = 0.1\ \ P(T) = 0.2\,. \tag{3.33}$$

The matrix $\boldsymbol{z}_1 = (0.3, 0.4, 0.1, 0.2)^T$ used with the categorical model trivially leads to such a distribution. But using the same matrix with a Gaussian model with a bandwidth parameters ω fixed as described in Subsection 3.3.5 leads to a slightly different distribution:

$$P(A) = 0.28\ \ P(C) = 0.43\ \ P(G) = 0.11\ \ P(T) = 0.18\,. \tag{3.34}$$

A distribution closer to (3.33) can be constructed with a Gaussian model parametrized by

$$\boldsymbol{z}_2 = (0.315, 0.38, 0.08, 0.225)^T\,.$$

To clarify the relationship between those two motifs, (3.31) can be rewritten to account for the fact that $\boldsymbol{u}$ is one-hot encoded. That is, for each position i it only has one 1 for

letter $j(i)$ and 0's elsewhere:

$$\forall(\boldsymbol{u}, \boldsymbol{z} \in \mathcal{Z}) \;\; \mathcal{L}_{\text{cat}}(\boldsymbol{u}; \boldsymbol{z}) = \prod_{i=1}^{k} \boldsymbol{z}_{i,j(i)} \,. \tag{3.35}$$

Assuming that the columns of $\boldsymbol{z}$ are normalized and $\omega = 1$, we can modify (3.32):

$$\forall(\boldsymbol{u}, \boldsymbol{z} \in \mathcal{Z}) \;\; \mathcal{L}_{\text{gaus}}(\boldsymbol{u}; \boldsymbol{z}) = C \prod_{i=1}^{k} e^{-\frac{\|\boldsymbol{u}_i - \boldsymbol{z}_i\|^2}{2\omega^2}} = C_2 \prod_{i=1}^{k} e^{\boldsymbol{u}_i^T \boldsymbol{z}_i} = C_2 \prod_{i=1}^{k} e^{\boldsymbol{z}_{i,j(i)}} \,. \tag{3.36}$$

With the Gaussian model and a few assumptions, the motifs can then be seen as defining the log probability to observe each letter at each of the k sites. This gives us a new interpretation for the filters learned by CNNs and suggests that, in this framework, it might be interesting to constrain $e^{\boldsymbol{z}}$ to be in the simplex $\mathcal{Z}$ rather than $\boldsymbol{z}$.

The true TF binding site motifs used in Subsection 3.6 from Jolma et al. (2013) must be interpreted with the categorical model, since they have been derived by averaging k-mers which corresponds to a maximum likelihood estimate under this model. This can explain why the $\boldsymbol{z}$ obtained with MEME/STREME are closer to those true motifs than the ones obtained with SEISM and a different model, as shown in Figure 3.8. In this framework, we used a Gaussian activation function to fit with the classical CNNs approaches, but SEISM is generic enough to allow other activation functions based on the categorical model by using the activation function $\tilde{\boldsymbol{\varphi}}_{\text{cat}}^{\boldsymbol{Z},\boldsymbol{X}}$ introduced in (3.8), or more realistic variants (Ruan & Stormo, 2017).

CHAPTER 4

A valid post-selection inference procedure for the association between the phenotype and trained convolutional filters

We now proceed to the issue of testing the association between the trained filters of our network, that is the selected motifs $\boldsymbol{z}$ and the phenotype $\boldsymbol{y}$. In order to do so, we need to solve three interrelated problems.

First, using Chapter 3, the motifs were specifically selected for their association with the trait, which leads to the well-known post-selection inference problem, as described in Chapter 2. Any inference procedure that disregards the fact that the null hypotheses were constructed or selected using the same data as the one used for testing is likely invalid and the results may appear more significant than they actually are.

Second, we deal with a continuous selection event, because the selection described in (3.30) is performed over a continuous set $\mathcal{Z}$. By contrast, existing solutions for conditional inference address selections over finite sets.

Third, the null hypothesis commonly used for similar post-selection inference problems is composite, *i.e.*, it corresponds to several values of the parameters. Existing methods work around this issue by fixing these parameters to arbitrary values, thereby limiting the scope under which they are calibrated. In this chapter, we present our solutions to these three problems.

4.1 Setting up the statistical framework and limitations due to conditioning

This section introduces the model and the null hypotheses associated with the sequence motifs, and highlights inherent limitations due to selecting motifs from a continuous set.

4.1.1 Introduction of the Gaussian model

Let's consider the Gaussian model:

$$\boldsymbol{y} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}\,, \tag{4.1}$$

where $\boldsymbol{\mu} \in \mathcal{E}$ is the target deterministic signal and $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_n)$ is the standard Gaussian distribution on $\mathcal{E} := \text{Range}(\boldsymbol{C}_n)$. We represent the probability distribution associated with this model through its probability measure υ.

We will follow (Yamada et al., 2018) and test the association of a motif $\boldsymbol{z}$ through the following null hypothesis:

$$\mathbb{H}_0 : \text{“}s(\boldsymbol{z}, \boldsymbol{\mu}) = 0\text{”}\,, \tag{4.2}$$

for some association score s.

For a $\boldsymbol{z}$ chosen independently of the data, $\mathbb{H}_0$ could be tested by sampling replicates $\boldsymbol{y}'$ under the corresponding distribution (4.1), and using the quantile of the scores $s(\boldsymbol{z}, \boldsymbol{y}')$ corresponding to $s(\boldsymbol{z}, \boldsymbol{y})$ as a p-value — *i.e.*, the probability, when sampling a phenotype under $\mathbb{H}_0$, to observe a score as extreme as $s(\boldsymbol{z}, \boldsymbol{y})$.

As in our frameworks, $\boldsymbol{z}$ was not chosen independently, we rely on recent developments in post-selection inference, and we will make use of the concept of selection event.

4.1.2 Selection event description

Formally, our selection event $\tilde{E}$ is the set of outcomes $\boldsymbol{y}'$ that would have led to the selection of the same set of motifs $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q\}$ as the one selected using the phenotype $\boldsymbol{y}$ from the dataset, when applying the same selection procedure:

$$\tilde{E}(\boldsymbol{Z}) := \left\{ \boldsymbol{y}' \in \mathcal{E} \;:\; \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}') = \boldsymbol{z}_j,\, \forall j \leq q \right\}, \tag{4.3}$$

where $\boldsymbol{P}_j$ is the orthogonal projection onto $\text{Span}_{\ell<j} \left\{ \mathbf{1}, \boldsymbol{\varphi}^{\boldsymbol{z}_\ell, \boldsymbol{X}} \right\}^T$.

A simple rejection approach to sample from the null (4.2) conditioned to $\tilde{E}(\boldsymbol{Z})$ would be to sample $\boldsymbol{y}'$ in $\mathcal{E}$ under (4.1, 4.2) and retain only those in $\tilde{E}(\boldsymbol{Z})$.

Unfortunately, $\tilde{E}(\boldsymbol{Z})$ belongs to a strictly lower-dimensional vector space of $\mathbb{R}^n$ and is therefore a null set for the measure υ. In other words, a vector $\boldsymbol{y}'$ sampled according to (4.1) has a zero probability to belong to $\tilde{E}(\boldsymbol{Z})$.

Indeed, for a vector $\boldsymbol{y}'$ to admit $\boldsymbol{z}_j$ as a maximum for the score at step j, the filter $\boldsymbol{z}_j$ must be at least a critical point (the gradient of the score must be zero in $\boldsymbol{z}_j$). For our scores s^{HSIC} and s^{ridge}, this leads to the following necessary condition:

$$\boldsymbol{y}' \in \tilde{E}(\boldsymbol{Z}) \Rightarrow \boldsymbol{P}_j \boldsymbol{y}' \in \text{Span}\left\{\nabla_z \boldsymbol{\varphi}^{\boldsymbol{z}_j, \boldsymbol{X}}\right\}^T, \forall j \leq q\,. \tag{4.4}$$

We will denote $\mathcal{S}(\boldsymbol{Z}) = \left\{\boldsymbol{y}' \in \mathcal{E} \,:\, \boldsymbol{P}_j \boldsymbol{y}' \in \text{Span}\left\{\nabla_z \boldsymbol{\varphi}^{\boldsymbol{z}_j, \boldsymbol{X}}\right\}^T, \forall j \leq q\right\}$ the set of vectors verifying this condition. We then obtain $\tilde{E}(\boldsymbol{Z}) \subseteq \mathcal{S}(\boldsymbol{Z})$.

For instance, for $q = 1$ and assuming that all the different directions of the gradient are independent, this set is a vector subspace $\mathcal{S}(\boldsymbol{Z})$ with dimension $n - 4 \times k$.

We empirically observed that sampling from this subspace produced a non-zero proportion of $\boldsymbol{y}'$ in $\tilde{E}(\boldsymbol{Z})$, allowing a rejection sampling strategy. Nonetheless, choosing a sampling distribution on $\mathcal{S}(\boldsymbol{Z})$ that leads to the correct conditional distribution on $\tilde{E}(\boldsymbol{Z})$ after rejection sampling is not straightforward — and may not even be possible — as discussed below and illustrated with the theorem of disintegration. We can note here that this result was not intuitive for us, and that we thought at the beginning that $\tilde{E}(\boldsymbol{Z})$ was reduced to a single point.

Moreover, relying on conditional probability with respect to a null set is not well defined and may lead to the Borel-Kolmogorov paradox (Bungert & Wacker, 2022), which further complicates its use.

4.1.3 Conditioning with respect to a null set: disintegration and Borel-Kolmogorov paradox

In what follows, we seek to derive the null distribution, conditioned with respect to $\mathcal{S}(\boldsymbol{Z})$, which would allow us, with a rejection sampling strategy, to approximate the conditional null distribution on $\tilde{E}(\boldsymbol{Z})$. To that end, we will leverage the disintegration theorem, requiring a mapping from $\mathcal{E}$ to the set of possible $\mathcal{S}$.

We will then consider the set:

$$\mathcal{D} = \{\mathcal{S} \,:\, \exists \boldsymbol{Z} \in \mathcal{Z}^q\,,\, \mathcal{S} = \mathcal{S}(\boldsymbol{Z})\}\,. \tag{4.5}$$

We also consider the mapping:

$$\begin{aligned} \pi \,&: \mathcal{E} \to \mathcal{D} \\ &\boldsymbol{y}' \to \mathcal{S}(\boldsymbol{Z}^{(1)})\,, \end{aligned} \tag{4.6}$$

where $\boldsymbol{Z}^{(1)}$ is the sequence of motifs selected with phenotype $\boldsymbol{y}'$.

In order to use the disintegration theorem, this mapping must be well defined, *i.e.* to a given $\boldsymbol{y}'$ corresponds a unique subspace $\mathcal{S}$, which is not straightforward.

Indeed, π being well defined is equivalent to

$$\begin{aligned} \pi' \,&: \mathcal{E} \to \mathcal{Z}^q \\ &\boldsymbol{y}' \mapsto \boldsymbol{Z}^{(1)} \end{aligned} \tag{4.7}$$

being well defined.

The latter is not clear, as a same $\boldsymbol{y}'$ may lead to the selection of at least two different motifs sequences $\boldsymbol{Z}^{(1)}$ and $\boldsymbol{Z}^{(2)}$ if the score admits more than one global maximum.

As a first remark, we can see that the set of problematic $\boldsymbol{y}'$ is exactly

$$\mathcal{P} := \bigcup_{\boldsymbol{Z}^{(1)} \neq \boldsymbol{Z}^{(2)}} \tilde{E}(\boldsymbol{Z}^{(1)}) \cap \tilde{E}(\boldsymbol{Z}^{(2)}) \,. \tag{4.8}$$

When one assumes that the $\boldsymbol{y}'$ admits a unique maximum $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$, one implicitly assumes that $\upsilon(\mathcal{P}) = 0$. For sufficiently regular scores this is however the case. We will not comprehensively study this issue here but we will give an argument that tends to support this assumption for the scores s^{HSIC} and s^{ridge}.

Indeed, with those scores, we can circumvent this difficulty considering the Gaussian random fields indexed by $\mathcal{Z}$:

- $\boldsymbol{z} \mapsto \langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{y}' \rangle$ for s^{HSIC},
- $\boldsymbol{z} \mapsto \langle (\|\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\|^2 + \lambda n)^{-1/2} \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{y}' \rangle$ for s^{ridge}.

Of note, in both cases the autocovariance function is $\mathcal{C}^2$ — the autocovariance functions of the Gaussian random fields are given by:

$$(\boldsymbol{z}, \boldsymbol{z}') \mapsto \sigma^2 \langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{\varphi}^{\boldsymbol{z}',\boldsymbol{X}} \rangle, \tag{4.9}$$

from (4.1) for the HSIC score, and by :

$$(\boldsymbol{z}, \boldsymbol{z}') \mapsto \sigma^2 \langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} \,,\, \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} \rangle \times (\|\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\|^2 + \lambda n)^{-1/2} (\|\boldsymbol{\varphi}^{\boldsymbol{z}',\boldsymbol{X}}\|^2 + \lambda n)^{-1/2} \tag{4.10}$$

for the Ridge score.

The score is then the largest norm of this Gaussian random field. It is well established in the theory of Gaussian random fields (Azäis & Wschebor, 2009, Chapter 7) that the law of this maximum is regular and that the argument maximum is unique. Tsirel'son (1976) and Lifshits (1983) are relevant references in this regard.

In Tsirelson's theorem, the parameter set is countable. This says that the same result holds true for separable bounded Gaussian processes, since in this case, the distribution of the supremum coincides almost surely with the one of the suprema on the countable nonrandom set.

This argument does not constitute a rigorous proof, and we will then assume that almost surely the selected sequence of motifs $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$ is uniquely defined, hence π', and consequently π, are well defined.

• The disintegration steps

To sample conditionally on $\{\boldsymbol{y}' \in \mathcal{S}(\boldsymbol{Z})\}$, we need to consider the conditional probability distribution with respect to this event. We will represent it using the associated probability measure $\upsilon_{\mathcal{S}(\boldsymbol{Z})}$, depending only on υ (the non-conditional distribution), $\mathcal{S}(\boldsymbol{Z})$ and π. This law is described by the theorem of disintegration (Ambrosio et al., 2005, Theorem 5.3.1).

Let's define ν the pushforward measure of υ by π, a probability measure on $\mathcal{D}$ denoted by $\nu = \pi_{\#}\upsilon$:

$$\begin{aligned} \nu := \pi_{\#}\upsilon \ &: \mathcal{D} \to \mathbb{R}^+ \\ &\mathcal{S}(\boldsymbol{Z}) \mapsto \upsilon(\pi^{-1}(\mathcal{S}(\boldsymbol{Z}))) \,. \end{aligned} \tag{4.11}$$

By the disintegration theorem, there exists a ν-almost everywhere uniquely determined family of probability measures $\left(\upsilon_{\mathcal{S}(\boldsymbol{Z})}\right)_{\mathcal{S}(\boldsymbol{Z})\in\mathcal{D}}$ on $\mathcal{E}$ — corresponding to the conditional distributions — such that:

- for ν-almost every $\mathcal{S}(\boldsymbol{Z})$, $\upsilon_{\mathcal{S}(\boldsymbol{Z})}\{\mathcal{E} \setminus \pi^{-1}(\mathcal{S}(\boldsymbol{Z}))\} = 0$. In other words, the probability measures $\upsilon_{\boldsymbol{Z}}$ are ν-almost everywhere supported by $S(\boldsymbol{Z})$;
- it holds that, for every map $f : \mathcal{E} \to [0, +\infty]$,

$$\int_{\mathcal{E}} f d\upsilon = \int_{\mathcal{D}} \left(\int_{\pi^{-1}(\mathcal{S}(\boldsymbol{Z}))} f \mathrm{d}\upsilon_{\mathcal{S}(\boldsymbol{Z})} \right) \mathrm{d}\nu(\mathcal{S}(\boldsymbol{Z})) = \int_{\mathcal{D}} \left(\int_{\mathcal{S}(\boldsymbol{Z})} f \mathrm{d}\upsilon_{\mathcal{S}(\boldsymbol{Z})} \right) \mathrm{d}\nu(\mathcal{S}(\boldsymbol{Z})) \tag{4.12}$$

  That is, the expectation of the conditional expectation is the expectation.

Let us comment on this result regarding our purposes. First, we have mentioned that we know that the support of $\tilde{E}(\boldsymbol{Z})$ is included in some vectorial subspace $\mathcal{S}(\boldsymbol{Z})$ defined by the first order condition (4.4).

Second, although one can use a rejection sampling strategy on the subspace $\mathcal{S}(\boldsymbol{Z})$ to draw points on the support $\tilde{E}(\boldsymbol{Z})$, it is not clear at all what should be the conditional distribution on $\mathcal{S}(\boldsymbol{Z})$, represented by its probability measure $\upsilon_{\mathcal{S}(\boldsymbol{Z})}$. Indeed, the family of probability measures $\left(\upsilon_{\mathcal{S}(\boldsymbol{Z})}\right)_{\mathcal{S}(\boldsymbol{Z})\in\mathcal{D}}$ is the unique family that satisfies (4.12).

It implies that a measure $\upsilon_{\mathcal{S}(\boldsymbol{Z}^{(1)})}$ depends on the other measures $\upsilon_{\mathcal{S}(\boldsymbol{Z}^{(2)})}$ and this dependency is geometrically given by the (piecewise) topological sub-manifold given by the function $\boldsymbol{z} \mapsto \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ from $\mathcal{Z}$ to $\mathcal{E}$.

From a practical point of view, we tried various simple distributions for $\upsilon_{\mathcal{S}(\boldsymbol{Z})}$, but none of them matched the condition (4.12). It resulted in decalibrated test procedures, as illustrated in Figure 4.1.

In the next point, we recall a toy example: the disintegration of the uniform measure on the sphere is not the uniform measure. Even in this simple geometrical example, the calculus of the conditional law might be seen as tedious. We believe that the calculus of $\upsilon_{\boldsymbol{Z}}$ is somehow out of reach for our purposes, and a solution to work around this issue will be described in the following subsections.

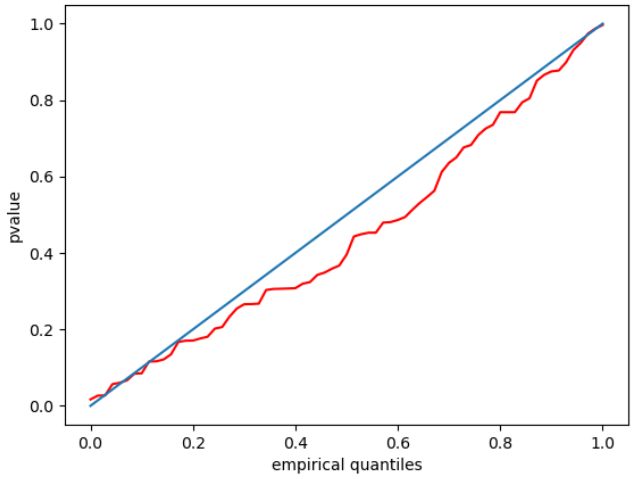


Figure 4.1: Q-Q plot obtained by applying the SEISM procedure on a simulated dataset under $\mathbb{H}_0$, using a Gaussian distribution (4.1) restricted to $\tilde{E}$. Instead of sticking to the $y = x$ line, we observe that the p-values tend to be too low. This indicates that this distribution is different from the conditional distribution.

- **A toy example on the sphere**

Let $\mathbb{S}$ be the unit sphere in $\mathbb{R}^3$, and consider υ the uniform measure on this sphere. Let $\{\mathcal{S}_\theta \,:\, \theta \in [0, \pi[\}$ be the family of planes, sharing $\mathrm{Span}\{(0,0,1)\}$ (say the south-north axis) as a revolution axis, parameterized by a longitude θ.

Let $\bar{\pi} \,:\, \mathbb{S} \to [0, \pi[$ be the function that maps a point to its longitude modulo π. We can now define the pushforward measure $\nu = \bar{\pi}_\# \upsilon$:

$$\begin{aligned} \nu := \bar{\pi}_\# \nu \,:\, & [0, \pi[\to \mathbb{R}^+ \\ & \theta \mapsto \upsilon(\bar{\pi}^{-1}(\theta)) \,. \end{aligned} \tag{4.13}$$

By spherical symmetries, this probability measure is the uniform measure on $[0, \pi[$, leading to $\mathrm{d}\nu(\theta) = (1/\pi)\mathrm{d}\theta$.

Condition (4.12) of the disintegration theorem (the left-hand side of the equality below) is given by the spherical coordinate system (the right-hand side) in:

$$\int_{\mathbb{S}} f \mathrm{d}\upsilon = \int_0^\pi \left(\int_{\bar{\pi}^{-1}(\theta)} f \mathrm{d}\upsilon_\theta \right) \mathrm{d}\nu(\theta) = \int_0^\pi \left(\int_0^{2\pi} f(\theta, \phi) \frac{|\sin \phi|}{4\pi} \mathrm{d}\phi \right) \mathrm{d}\theta \,, \tag{4.14}$$

where ϕ is the latitude.

We can note that $\bar{\pi}^{-1}(\theta) = \mathbb{S} \cap \mathcal{S}_\theta$ is in bijection with $[0, 2\pi[$, using the mapping that maps a point to its latitude.

Using this representation, we can see that the uniform probability measure on $\bar{\pi}^{-1}(\theta)$ is given by $(1/2\pi)\mathbf{1}_{[0,2\pi)}(\phi)$, with $\mathbf{1}$ the indicator function.

But the above equality shows that the conditional probability distribution, represented by its measure $\upsilon_{\theta^{(1)}}$ for the uniform probability on $\mathbb{S}$ conditioned to a fixed longitude $\theta^{(1)}$, admits $(1/4)|\sin(\phi)|\mathbf{1}_{[0,2\pi)}(\phi)$ as a density, see Figure 4.2.

It proves that the disintegration of the uniform measure on the sphere is not the uniform measure, but rather a distribution that will put a little weight around the poles and large mass around the equator.

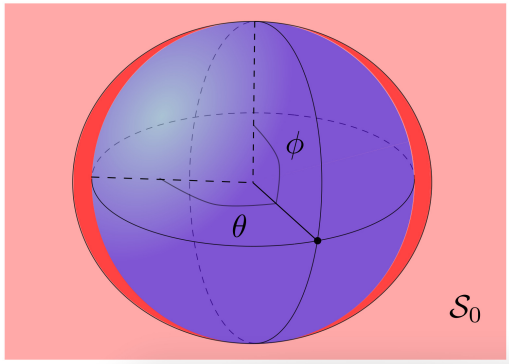


Figure 4.2: For $\theta = 0$, $\mathcal{S}_\theta$ is the light red plan, the conditional measure $\mathrm{d}\nu_0(\phi)$ is depicted with a red area and is proportional to $|\sin(\phi)|$, which is not the uniform measure.

Equation (4.14) also shows that, if we choose to condition the uniform probability on $\varphi = 0$ rather than on a fixed θ, the resulting density is uniform: $\frac{1}{\pi}\mathbf{1}_{[0,\pi[}(\theta)$. But in both cases, we condition on a great circle. We here have a first intuition on the Borel-Kolmogorov paradox: conditioning on a null set may lead to different results, depending on the chosen parameterization. We will now illustrate this paradox on an example.

• Illustration of the Borel-Kolmogorov paradox

Largely inspired by https://en.wikipedia.org/wiki/Borel-Kolmogorov_paradox*, as of 01/14/2023.*

Indeed, in addition to being out of reach, the conditional probability distribution given a zero-probability event, such as $\boldsymbol{y} \in \tilde{E}(\boldsymbol{Z})$ for a vector $\boldsymbol{y}$ under the Gaussian model (4.1) and some motifs $\boldsymbol{Z}$, may be ill-defined: let's consider two random variables X and Y and recall the definition of the conditional probability density, given the event $X = x$:

$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x,y)}{f_X(x)}\,, \tag{4.15}$$

where $f_{X,Y}(x,y)$ is the joint density of X and Y, and $f_X(x)$ is the marginal density for X. We then see that if $\mathbb{P}(X = x) = 0$, (4.15) is undefined.

We can then try to work around this problem by conditioning on "X close to x", for instance $X \in \{x - \epsilon, x + \epsilon\}$ and defining the conditional probability given $X = x$ as the

limit obtained for $\epsilon \to 0$. But the Borel-Kolmogorov paradox demonstrates that it cannot be achieved in a consistent manner.

To illustrate this paradox, let's consider a random vector $V = (X, Y, Z)$, uniformly distributed on the unit sphere $\mathbb{S}$. We will consider two events, illustrated in Figure 4.3:

- $A = \{0 < X < 1\,,\, 0 < Y < X\}$
- $B = \{Z = 0\}$ (and then $\mathbb{P}(B) = 0$).

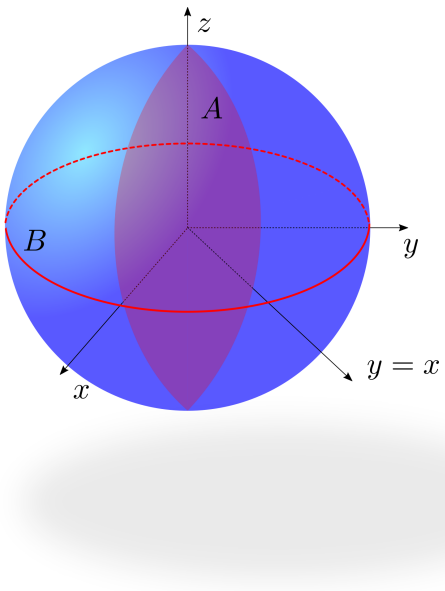


Figure 4.3: The unit sphere $\mathbb{S}$, with events A (the spherical wedge) and B (the great circle).

We will now compute $\mathbb{P}(A|B)$ using two different — but equivalent — parameterizations, described in Table 4.1. Parameterization 2 is indeed a rotation of 90° around the y-axis of parameterization 1.

With parametrization 1, we can note that as Φ and Θ are independent, the events A and B_ϵ are also independent when formulated with this parameterization. We then obtain:

$$\lim_{\epsilon \to 0} \frac{\mathbb{P}(A \cap B_\epsilon)}{\mathbb{P}(B_\epsilon)} = \lim_{\epsilon \to 0} \frac{\mathbb{P}(A) \times \mathbb{P}(B_\epsilon)}{\mathbb{P}(B_\epsilon)} = \mathbb{P}(A) = \mathbb{P}\left(\Theta \in \left]0, \frac{\pi}{4}\right[\right) = \frac{1}{8}$$

However, with parameterization 2, the calculation is less straightforward. While Φ' and Θ'

Parameterization 1	**Parameterization 2**
New coordinates, with $(\varphi,\theta)\in[-\frac{\pi}{2},\frac{\pi}{2}]\times[-\pi,\pi]$	
$x=r\cos\varphi\cos\theta$ $y=r\cos\varphi\sin\theta$ $z=r\sin\varphi$	$x=r\sin\varphi$ $y=r\cos\varphi\sin\theta$ $z=-r\cos\varphi\cos\theta$
$R=\sqrt{X^2+Y^2+Z^2}$ $\Phi=\arcsin(Z)$ $\Theta=\arctan_2\left(\frac{Y}{\sqrt{1-Z^2}},\frac{X}{\sqrt{1-Z^2}}\right)$	$R'=\sqrt{X^2+Y^2+Z^2}$ $\Phi'=\arcsin(X)$ $\Theta'=\arctan_2\left(\frac{Y}{\sqrt{1-X^2}},\frac{-Z}{\sqrt{1-X^2}}\right)$
Jacobian matrix $\boldsymbol{J}$	
$\begin{pmatrix}\cos\varphi\cos\theta & -r\cos\varphi\sin\theta & -r\sin\varphi\cos\theta\\ \cos\varphi\sin\theta & r\cos\varphi\cos\theta & -r\sin\varphi\sin\theta\\ \sin\varphi & 0 & r\cos\varphi\end{pmatrix}$	$\begin{pmatrix}\sin\varphi & 0 & r\cos\theta\\ \cos\phi\sin\theta & r\cos\phi\cos\theta & -r\sin\varphi\sin\theta\\ -\cos\varphi\cos\theta & r\cos\varphi\sin\theta & r\sin\varphi\cos\theta\end{pmatrix}$
Determinant	
$\lvert\boldsymbol{J}\rvert=r^2\cos\varphi$	$\lvert\boldsymbol{J}\rvert=r^2\cos\varphi$
Surface of a spherical cap wedge $(r=1)$	
$\text{Area}(\Theta\le\theta,\Phi\le\varphi)$ $\int_{-\pi}^{\theta}\int_{-\pi/2}^{\phi}\cos(\varphi)\mathrm{d}\varphi\mathrm{d}\theta=(1+\sin(\varphi))(\theta+\pi)$	$\text{Area}(\Theta'\le\theta,\Phi'\le\varphi)$ $(1+\sin(\varphi))(\theta+\pi)$
Joint cumulative distribution function on $\mathbb{S}$	
$F_{\Phi,\Theta}(\varphi,\theta)=\frac{1}{4\pi}(1+\sin(\varphi))(\theta+\pi)$	$F_{\Phi',\Theta'}(\varphi,\theta)=\frac{1}{4\pi}(1+\sin(\varphi))(\theta+\pi)$
Joint probability density function on $\mathbb{S}$	
$f_{\Phi,\Theta}(\varphi,\theta)=\frac{\partial^2F_{\Phi,\Theta}(\varphi,\theta)}{\partial\phi\partial\theta}=\frac{1}{4\pi}\cos\varphi$	$f_{\Phi',\Theta'}(\varphi,\theta)=\frac{1}{4\pi}\cos\varphi$
Translations of $A=\{0<X<1,0<Y<X\}$ *and* $B=\{Z=0\}$	
$A=\{0<\Theta<\frac{\pi}{4}\}$ $B=\{\Phi=0\}$	$A=\{\Theta'\in]0,\pi[,\Phi'\in]0,\frac{\pi}{2}[,\sin\Theta'<\tan\Phi'\}$ $B=\{\Theta'=-\frac{\pi}{2}\}\ \cup\{\Theta'=\frac{\pi}{2}\ \}$
Definition of events B_ϵ *and* B'_ϵ	
$B_\epsilon=\{\lvert\Phi\rvert<\epsilon\}$	$B'_\epsilon=\left\{\lvert\Theta'+\frac{\pi}{2}\rvert<\epsilon\right\}\ \cup\left\{\lvert\Theta'-\frac{\pi}{2}\rvert<\epsilon\right\}$

Table 4.1: Two equivalent parameterizations for the unit sphere $\mathbb{S}$ and their implications on the joint density functions and on events A and B.

are still independent, the events A and B (and *a fortiori* A and B_ϵ) are not independent:

$$\begin{aligned}
\lim_{\epsilon\to 0}\frac{\mathbb{P}(A\cap B'_\epsilon)}{\mathbb{P}(B'_\epsilon)} &= \lim_{\epsilon\to 0}\frac{2\pi}{4\epsilon}\mathbb{P}(A\cap B'_\epsilon)\\
&= \lim_{\epsilon\to 0}\frac{\pi}{2\epsilon}\int_{\pi/2-\epsilon}^{\pi/2+\epsilon}\int_0^{\pi/2}\mathbf{1}_{\sin(\theta)<\tan(\varphi)}f_{\Phi',\Theta'}(\varphi,\theta)\mathrm{d}\varphi\mathrm{d}\theta\\
&= \frac{\pi}{2}\lim_{\epsilon\to 0}\frac{\partial}{\partial\epsilon}\int_{\pi/2-\epsilon}^{\pi/2+\epsilon}\int_0^{\pi/2}\mathbf{1}_{\sin(\theta)<\tan(\varphi)}f_{\Phi',\Theta'}(\varphi,\theta)\mathrm{d}\varphi\mathrm{d}\theta \qquad \text{(L'Hôpital's rule)}\\
&= \pi\int_0^{\pi/2}\mathbf{1}_{1<\tan(\varphi)}f_{\Phi',\Theta'}\left(\varphi,\frac{\pi}{2}\right)\mathrm{d}\varphi \qquad \text{(Leibniz integral rule)}\\
&= \pi\int_{\pi/4}^{\pi/2}\frac{1}{4\pi}\cos(\varphi)\mathrm{d}\varphi\\
&= \frac{1}{4}\left(1-\frac{1}{\sqrt{2}}\right)
\end{aligned}$$

Parameterization 1 and 2 then lead to different values, demonstrating that $\lim_{\epsilon\to 0}\frac{\mathbb{P}(A\cap B_\epsilon)}{\mathbb{P}(B_\epsilon)}$ cannot be used as a conditional probability $\mathbb{P}(A|B)$ when $\mathbb{P}(B)=0$.

4.2 Quantization of the motif space using meshes

In this work, we choose to circumvent the issue of conditioning with respect to a null set described above by using a partition of the space $\mathcal{Z}$, over which our selection operates, into a very large but finite set of meshes: $\mathcal{Z}=\bigsqcup M_i$. This quantization solves the aforementioned problems, since the set of $\boldsymbol{y}'$ resulting in motifs in a given mesh is then of strictly positive measure. As depicted in Figure 4.4, we consider a regular partition of each coordinate into m bins.

Based in this partition into meshes, we define a quantized selection event E as follows. First, given an outcome $\boldsymbol{y}$ we define the sequence of the q selected meshes $(M_{i_1},\dots,M_{i_q})$ as

$$\forall j\leq q\,,\ \arg\max_{\boldsymbol{z}\in\mathcal{Z}} s(\boldsymbol{z},\boldsymbol{P}_j\boldsymbol{y})\in M_{i_j}\,. \tag{4.16}$$

Second, the selection event is given by

$$E(i_1,\dots,i_q):=\left\{\boldsymbol{y}'\in\mathcal{E}\ :\ \arg\max_{\boldsymbol{z}\in\mathcal{Z}} s(\boldsymbol{z},\boldsymbol{P}_j\boldsymbol{y}')\in M_{i_j}\,,\ \forall j\leq q\right\}, \tag{4.17}$$

the set of outcomes $\boldsymbol{y}'$ that would have led to the selection of motifs within the same meshes $(M_{i_1},\dots,M_{i_q})$ as the ones selected with $\boldsymbol{y}$.

This quantization leads to the definition of selection events with non-zero Lebesgue measure, and so the issues identified in the previous section are no longer present. Deriving an analytical expression for the conditional null distribution is nevertheless out of reach to our knowledge, and we will see how it can be approximated using a sampling procedure.

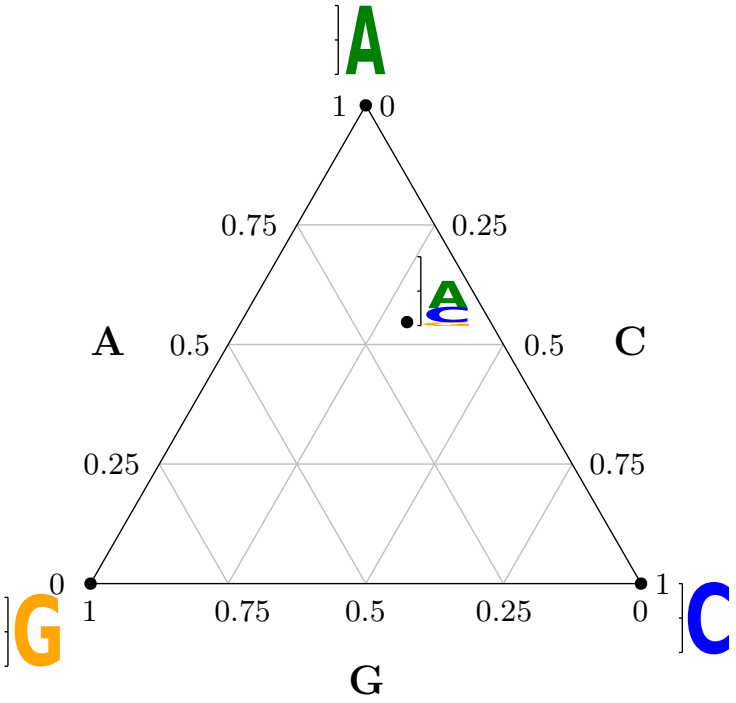


Figure 4.4: Discretization of the 3-letters alphabet simplex A, C, G, with a binning parameter for the meshes $m = 4$.

This meshing introduces several hyperparameters: in addition of the binning parameter m, our choice for the geometry of the meshes is only one possibility among others. Although if applied to the 3-letters simplex, it leads to meshes with identical sizes, as illustrated in Figure 4.4, it is not the case anymore when applied to the 4-letters simplex — for instance, with $m = 2$, the mesh located at the center of the simplex is bigger than the other meshes. We chose this option because it seemed the most straightforward, but other possibilities can be explored in the future.

In addition to modifying the interpretation of the tests, the mesh size also affects the power. Following (Fithian et al., 2017, Proposition 3), we typically sacrifice power as we move to finer selection events. For instance, in the case of LASSO, we start by conditioning on both the selected model and the signs (Chapter 2 Subsection 2.4.2) because it facilitates the access to the conditional null distribution. But in a second step, we remove the conditioning on the signs (thanks to an union), notably for the sake of power gain. In our case, conditioning on finer meshes does not allow us to access the conditional distribution. On the contrary, due to the sampling procedure, the opposite is true: it is harder to sample from a finer selection event, as discussed in Subsection 4.4.2. The choice of m is then determined by the fineness with which we want to test a motif, and the preceding arguments are in favour of selecting meshes that are not too fine.

4.3 Description of SEISM's test procedure

We now show how the quantization (4.17) of the selection event enables the definition of a valid inference procedure. We start with the simplest case, where we select only a single motif ($q = 1$).

4.3.1 Testing procedure for a single motif

In this section, considering that only the motif $\boldsymbol{z}_1$ was chosen by the SEISM selection procedure, selection event (4.17) boils down to:

$$E(i_1) := \left\{ \boldsymbol{y}' \in \mathcal{E} \ : \ \arg\max_{\boldsymbol{z} \in \mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{y}') \in M_{i_1} \right\} . \tag{4.18}$$

4.3.1.1 Definition of the null hypotheses

We use this simplified case to introduce our null hypotheses and test statistics attached to this selection event, and we consider two options:

- A first option consists in representing the mesh M_{i_1} by its center $\boldsymbol{c}_1$. Then, the corresponding null hypothesis is the following:

$$\mathbb{H}'_{0,1} \ : \ \text{“}s(\boldsymbol{c}_1, \boldsymbol{\mu}) = 0\text{”} . \tag{4.19}$$

  It can be tested using statistic $V'_1 = s(\boldsymbol{c}_1, \boldsymbol{y})$.

- A second possibility is to represent M_{i_1} by the motif with the highest association score within it. In this case, the null hypothesis becomes:

$$\mathbb{H}''_{0,1} \ : \ \text{“}s(\boldsymbol{z}, \boldsymbol{\mu} = 0) \, , \, \forall \boldsymbol{z} \in M_{i_1}\text{”} . \tag{4.20}$$

  We test it using statistic $V''_1 = \max_{\boldsymbol{z} \in \mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{y})$.

In both cases, we reject the null hypothesis if the test statistics are greater than a threshold, determined by their cumulative distributions under the nulls (4.19, 4.20) conditionally to $E(i_1)$: $\mathbb{F}'_{1,(i_1)}$ and $\mathbb{F}''_{1,(i_1)}$. In practice, we do not have closed forms for these conditional cumulative distributions, and we rely on empirical versions that we build using a hit-and-run sampler algorithm, as described in Subsection 4.3.1.2.

Hypotheses (4.19) and (4.20) lead to very similar results when the meshes are small enough, which is easily the case in practice. Hypothesis (4.19) gives us insights on one specific motif of the mesh — the center, but (4.20) tells us about whether there exists a motif within M_{i_1} associated with the phenotype. To illustrate the difference, let us consider a meshing with only one bin per coordinate, that is the meshing with only one mesh, containing all the motifs.

- Testing the center-based null hypothesis (4.19) boils down to testing the association of $\boldsymbol{\mu}$ with the motif $\boldsymbol{c}_1$ with the same probabilities for each letter of $\mathcal{A}$ at every position (the center of the simplex). It produces a p-value of 1, regardless of the data, since for any k-mer $\boldsymbol{u}$, $\|\boldsymbol{c}_1 - \boldsymbol{u}\|_2^2 = k \times (0.75^2 + 3 \times 0.25^2)$, leading to $\boldsymbol{\varphi}^{\boldsymbol{c}_1, \boldsymbol{X}} = \mathbf{0}$ for all $\boldsymbol{X} \in \mathcal{X}^n$, according to the centering step in SEISM. Finally, it leads to a zero score for any $\boldsymbol{y}' \in \mathcal{E}$, and consequently to a value of 1 for the p-value.
- By contrast, one can obtain a non-trivial p-value for (4.20), because different $\boldsymbol{y}' \in \mathcal{E}$ can lead to different scores, which means that there may exist a motif in $\mathcal{Z}$ associated with $\boldsymbol{y}$ — but does not inform us on which motif it is.

4.3.1.2 Sampling from the conditional null distribution with the hit-and-run algorithm

Even after reducing our selection event to a finite set (Subsection 4.2), a rejection sampling strategy that would draw $\boldsymbol{y}'$ from either (4.1, 4.19) or (4.1, 4.20) and only retain those leading to the selection of the same mesh as $\boldsymbol{y}$ is not tractable as the rejection rate is empirically too low. Following (Slim et al., 2019), we resort to a Hypersphere Direction strategy (Algorithm 2).

The hit-and-run algorithm produces uniform samples from an open and bounded acceptance region (Smith, 1984; Bélisle et al., 1993) — corresponding, in our case, to the selection event. It starts from any point in the acceptance region, draws a random direction from this point and performs rejection sampling along this direction until it finds one element that also falls in the acceptance region. It then follows the same procedure from this new starting point.

The hit-and-run sampler therefore also relies on rejection, but it does so along a single direction rather than over $\mathbb{R}^n$. It explores the selection event step by step, starting from a point that belongs to this event, guaranteeing a higher acceptance rate than naive rejection sampling.

But since the points produced by this algorithm are not independent from each other, we need a large number of burn-in — the first $\boldsymbol{y}'$ generated are trashed to remove the dependence on the original $\boldsymbol{y}$ — and a large number of replicates to produce a good approximation for the targeted distribution. While (Smith, 1984, Section 3) provides theoretical upper bounds on the rate of convergence towards the uniform distribution, these bounds highly overestimate the required number of iterations. In practice, this number is set heuristically. Its impact is studied in Subsection 4.4.3. Nonetheless, these bounds tell us that two parameters are critical to fix those numbers:

- The dimension of the problem n, resulting in a curse of dimensionality;
- The geometry of the acceptance region, as the bounds depend on the ratio between the volume of the acceptance region and the volume of the smallest sphere containing this region.

To speed up the procedure, we parallelize the rejection step across several computing cores. Because each point sampled by the hit-and-run procedure depends on the previous

Algorithm 2 Hypersphere Directions hit-and-run sampler

`/* Description: The Hypersphere Directions hit-and-run sampler creates a discrete-time Markov chain on an open and bounded region and is used to approximate a uniform distribution on the selection event` E. `*/`

Inputs: *Response* $\boldsymbol{y} \in E \subseteq \mathbb{R}^n$, *$B$ and R the numbers of burn-in iterations and replicates.*

Result: $\boldsymbol{y}'^{(B+1)}, ..., \boldsymbol{y}'^{(B+R)} \in E \subseteq \mathbb{R}^n$ the replicates sampled under the conditional null distribution

$\tilde{\boldsymbol{y}}'^{(0)} \leftarrow \mathbb{L}(\boldsymbol{y})$; `/* ` $\mathbb{L}$ `is the cumulative distribution function of` $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$ `*/`

for $t = 1, \ldots, B + R$ **do**

  Sample uniformly $\boldsymbol{\theta}^{(t)}$ from $\{\boldsymbol{\theta} \in \mathbb{R}^n, \ \|\boldsymbol{\theta}\| = 1\}$;

  $$a^{(t)} \leftarrow \max \left\{ \max_{\boldsymbol{\theta}_t^{(i)} > 0} -\frac{\tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}}; \max_{\boldsymbol{\theta}_t^{(i)} < 0} \frac{\mathbf{1} - \tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}} \right\};$$

  $$b^{(t)} \leftarrow \max \left\{ \min_{\boldsymbol{\theta}_t^{(i)} < 0} -\frac{\tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}}; \min_{\boldsymbol{\theta}_t^{(i)} > 0} \frac{\mathbf{1} - \tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}} \right\};$$

  `/* Sampling` $\lambda^{(t)}$ `from` $\left]a^{(t)}, b^{(t)}\right[$ `ensures that` $\tilde{\boldsymbol{y}}'^{(t-1)} + \lambda^{(t)}\boldsymbol{\theta^{(t)}} \in]0,1[^n$ `*/`

  while $\boldsymbol{y}'^{(t)} \notin E$ **do**

    `/* This loop is parallelized on several cores until one of them discovers a replicate in the selection event. */`

    Sample uniformly $\lambda^{(t)}$ from $\left]a^{(t)}, b^{(t)}\right[$;

    $\tilde{\boldsymbol{y}}'^{(t)} \leftarrow \tilde{\boldsymbol{y}}'^{(t-1)} + \lambda^{(t)}\boldsymbol{\theta^{(t)}}$;

    $\boldsymbol{y}'^{(t)} \leftarrow \mathbb{L}^{-1}(\tilde{\boldsymbol{y}}'^{(t)})$;

  end

end

one, it is impossible to parallelize the whole sampling process. By contrast, the rejection step used for computing a single replicate, once a sampling direction has been fixed, can be parallelized. We draw several distances to the initial point independently, optimizing new independent points, until one of them belongs to the selection event. This parallelization provides a significant time saving, as discussed in Subsection 4.4.2.

This algorithm is designed to produce uniform samples from an open and bounded acceptance region. While the openness requirement is ensured by the definition of the meshes, the boundedness assumption does not hold in our case, as the arg max over $\mathcal{Z}$ of the score only depends on the direction of $\boldsymbol{y}$ and not on its norm (at least for s^{HSIC}).

Following (Slim et al., 2019) again, we use the reparameterization $\tilde{\boldsymbol{y}} = \mathbb{L}(\boldsymbol{y})$, where $\mathbb{L} : \mathbb{R}^n \rightarrow]0,1[^n$ is defined as $\mathbb{L}(\boldsymbol{y})_i = \mathbb{L}_{\boldsymbol{\mu},\sigma^2}(\boldsymbol{y}_i)$ for $i = 1, \ldots, n$ and $\mathbb{L}_{\boldsymbol{\mu},\sigma^2}$ denotes the cumulative distribution function of $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$. Sampling uniform $\tilde{\boldsymbol{y}}$ from the open bounded space $]0,1[^n$ then indirectly provides normal samples from $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$, using $\mathbb{L}^{-1}(\tilde{\boldsymbol{y}})$.

Combining this sampling strategy with the quantization of the selection event described in Subsection 4.2 and the selective null hypotheses attached to this event provides a selective inference procedure for one selected motif $\boldsymbol{z}_1$ ($q = 1$) and a null defined by a given pair

$(\boldsymbol{\mu}, \sigma)$ of parameters. Our next steps are to handle the selection of multiple motifs, and the general case where several $\boldsymbol{\mu}$ describe the same null hypothesis and σ is not specified.

4.3.2 Testing procedure for $q > 1$ motifs

We now consider that we selected $q > 1$ motifs with the SEISM procedure, leading to the general selection event $E(i_1, \ldots, i_q)$ (4.17). Generalizing our single-motif strategy described above, we propose two options for defining null hypotheses (and test statistics) related to this selection event:

- The first one relies one the centers of the selected meshes

$$\mathbb{H}_{0,j} \; : \; \text{“}s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{\mu}) = 0\text{”}\,, \tag{4.21}$$

  where $\boldsymbol{\Pi}'_j$ is the orthogonal projector onto $\text{Span}_{\ell \neq j} \left\{ \boldsymbol{\varphi}^{\boldsymbol{c}_\ell, \boldsymbol{X}} \right\}^{\perp}$. In other words, it expresses that the center of the mesh M_{i_j} is associated with $\boldsymbol{\mu}$ after removing its component carried by the span of the center of the meshes corresponding to the $q-1$ other motifs.

- And the second one takes advantages of the best motifs in each mesh:

$$\mathbb{H}_{0,j} \; : \; \text{“}s\left(\boldsymbol{z}, \boldsymbol{\Pi}''\left(\left(\boldsymbol{z}^*_{i_\ell}\right)_{\ell \neq j}\right)\boldsymbol{\mu}\right) = 0\,,\; \forall (\boldsymbol{z}^*_{i_\ell})_{\ell \neq j} \in (M_{i_\ell})_{\ell \neq j}\,,\; \forall \boldsymbol{z} \in M_{i_j}\text{”}\,, \tag{4.22}$$

  with $\boldsymbol{\Pi}''\left(\left(\boldsymbol{z}^*_{i_\ell}\right)_{\ell \neq j}\right)$ being the projection onto $\text{Span}_{\ell \neq j} \left\{ \boldsymbol{\varphi}^{\boldsymbol{z}^*_{i_\ell}, \boldsymbol{X}} \right\}^{\perp}$.

Generalizing what we introduced for $q = 1$ motif (Subsection 4.3.1), we test those hypotheses using $V'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y})$ and $V''_j = \max_{\boldsymbol{z} \in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j \boldsymbol{y})$.

To that end, we rely on their cumulative distributions under the nulls (4.21, 4.22) conditionally to $E(i_1, \ldots, i_q)$: respectively $\mathbb{F}'_{1,\ldots,q,(i_1,\ldots,i_q)}$ and $\mathbb{F}''_{1,\ldots,q,(i_1,\ldots,i_q)}$, empirically approximated using Algorithm 2.

Following the work of Loftus & Taylor (2015) in the finite case, both versions of our null hypothesis are joint across the q motifs: each of them considers the association between the j-th selected motif and $\boldsymbol{\mu}$ after projection onto the span of all others, not just the ones that were selected before — using $\boldsymbol{\Pi}'$ and $\boldsymbol{\Pi}''$. This is to be contrasted to our sequential selection process, which adjusts at each step for the previously selected motifs using $\boldsymbol{P}$.

• **Description of the null hypotheses**

In order to give more insights on the null hypotheses (4.21, 4.22), we derive the following proposition:

Proposition 4.3.1 (Description of the selective nulls)**.**
Let $\boldsymbol{Z} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_q\}$ *be* q *sequence motifs. Let* $s(\cdot, \cdot)$ *be a score such that "nullity implies orthogonality" (for instance* s^{HSIC} *or* s^{ridge}*):*

($\mathbf{A_1}$) **Nullity implies orthogonality:** *If* $\{s(\boldsymbol{z}, \boldsymbol{y}) = 0\}$ *then* $\left\{\langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{y}\rangle = 0\right\}$*, for every* $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$*, and for some function* $\boldsymbol{z} \to \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} \in \mathcal{E}$*.*

Let $\boldsymbol{\mu} \in \mathcal{E}$ *and decompose* $\boldsymbol{\mu}$ *as*

$$\boldsymbol{\mu} = \sum_{j=1}^{q} \alpha_j \boldsymbol{\varphi}^{\boldsymbol{z}_j,\boldsymbol{X}} + \underline{\mu}\,, \tag{4.23}$$

with $\underline{\mu} \in \mathcal{E}$ *orthogonal to* $\mathrm{Span}(\varphi^{\boldsymbol{Z},\boldsymbol{X}})$*.*

It holds that "$s(\boldsymbol{z}_j, \boldsymbol{\Pi}_j \boldsymbol{\mu}) = 0$*" is equivalent to "*$\alpha_j = 0$*" for some decomposition* (4.23)*.*

If $\mathrm{Rank}\left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\right) = q$ then the decomposition (4.23) is unique — the greedy selection procedure described in Chapter 3 enforces this situation. We interpret this as follows: we look at a motif $\boldsymbol{z}_\ell$ and would like to test its significance; in view of property ($\mathbf{A_1}$), we can eliminate the effects that are captured by the other motifs by using the orthogonal projection onto the orthogonal of $\mathrm{Span}(\boldsymbol{\varphi}^{\boldsymbol{z}_j,\boldsymbol{X}})$, given by $\boldsymbol{\Pi}_j$ — using $\boldsymbol{\Pi}_j = \boldsymbol{\Pi}'_j$ or $\boldsymbol{\Pi}_j = \boldsymbol{\Pi}''_j\left(\left(\boldsymbol{z}^*_{i_\ell}\right)_{\ell \neq j}\right)$.

Finally, we can consider $\boldsymbol{\Pi}_j \boldsymbol{y}$ to test the association "$s(\boldsymbol{z}_j, \boldsymbol{\Pi}_j\ \boldsymbol{\mu}) = 0$", equivalently to testing "$\alpha_j = 0$" by the above proposition.

4.3.3 Sampling under selective composite hypotheses with known variance σ

The sampling strategy described in Subsection 4.3.1 builds a conditional null distribution — therefore offering a selective inference procedure — for a given $\boldsymbol{\mu}$ and σ. In practice, σ is not known, and several values of $\boldsymbol{\mu}$ can describe the selective null hypotheses (4.21, 4.22) for a given motif selection. Of note, this issue is not specific to our selective inference procedure. It will arise in any sampling-based post-selection inference strategy including data-split: even if the latter samples from a non-selective null hypothesis, it still needs given values for $\boldsymbol{\mu}$ and σ.

We leave aside the choice of σ for now, and describe how we can sample from any null distribution (4.21) or (4.22) for a given σ. Our results hold for scores verifying the following assumption — this includes both s^{HSIC} and s^{ridge}:

($\mathbf{A_2}$) **Nullity implies translation-invariant:** For every $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$, if $s(\boldsymbol{z}, \boldsymbol{y}) = 0$, then $\forall \boldsymbol{y}' \in \mathcal{E}$, $s(\boldsymbol{z}, \boldsymbol{y}') = s(\boldsymbol{z}, \boldsymbol{y} + \boldsymbol{y}')$

Under this assumption, the following proposition ensures that relying on the quantiles of the empirical distribution of scores sampled under $\boldsymbol{\mu} = \mathbf{0}$ leads to a calibrated test procedure.

Proposition 4.3.2.
Let s be an association score such that ($\mathbf{A_2}$) holds. Let $V'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y})$ and $V''_j = \max_{\boldsymbol{z} \in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j \boldsymbol{y})$, formed from $\boldsymbol{y}$ sampled according to (4.1) with any mean $\boldsymbol{\mu}$ such that $s(\boldsymbol{z}', \boldsymbol{\mu}) = 0$, any known variance $\sigma > 0$, and such that $\boldsymbol{z}' = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{y})$.

The conditional null distributions $\mathbb{F}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{F}''_{j,(i_1,\dots,i_q)}$ with means $\mathbf{0}$ and variance σ verify:

$$\mathbb{F}'_{j,(i_1,\dots,i_q)}(V'_j) \sim \mathcal{U}(0,1) \text{ and } \mathbb{F}''_{j,(i_1,\dots,i_q)}(V''_j) \sim \mathcal{U}(0,1) \tag{4.24}$$

Proof. Assumption ($\mathbf{A_2}$) under the Gaussian model (4.1) implies the following property:

$$\begin{aligned} \forall (\boldsymbol{z}, \boldsymbol{A}, \boldsymbol{y}) \in \mathcal{Z} \times \mathbb{R}^{n \times n} \times \mathcal{E} \text{ such that } \boldsymbol{y} = \boldsymbol{\mu} + \sigma \boldsymbol{\epsilon}, \\ \text{“} s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{\mu}) = 0 \text{”} \Rightarrow \text{“} s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y}) = s(\boldsymbol{z}, \sigma \boldsymbol{A}\boldsymbol{\epsilon}) \text{”}, \end{aligned} \tag{4.25}$$

which implies that, for a composite null hypothesis of the form $\mathbb{H}_0$: “$s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{\mu}) = 0$”, the distribution of $s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y})$ does not depend on the mean $\boldsymbol{\mu}$ that satisfies $\mathbb{H}_0$.

Hence, even if the hypothesis $\mathbb{H}_0$ corresponds to a set of probability distributions of $\boldsymbol{y}$ that may depend on $\boldsymbol{\mu}$, the distribution of the statistic $s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y})$ does not depend on $\boldsymbol{\mu}$ under this hypothesis. We can then conclude that if σ is known, as it is assumed to be the case in this section, then a test statistic of the form $V = s(\boldsymbol{z}, \boldsymbol{\Pi}\boldsymbol{y})$ has the same distribution as $s(\boldsymbol{z}, \sigma\boldsymbol{\Pi}\boldsymbol{\epsilon})$. □

4.3.4 Sampling under selective composite hypotheses with unknown σ

In practice, σ is often unknown. To address this issue, we rely on the normalized versions of the test statistics V' and V'' introduced in Subsection 4.3.2, defined by

$$T'_j := \frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y})}{\|\boldsymbol{y}\|_2^2} \text{ and } T''_j := \max_{\boldsymbol{z} \in M_{i_j}} \frac{s\left(\boldsymbol{z}, \boldsymbol{\Pi}''\left((\boldsymbol{z}_{i_\ell})_{\ell \neq j}\right)\boldsymbol{y}\right)}{\|\boldsymbol{y}\|_2^2}, \tag{4.26}$$

where $\boldsymbol{z}_{i_\ell} = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \ s(\boldsymbol{z}, \boldsymbol{P}_\ell \boldsymbol{y}) \in M_{i_\ell}$. We will denote $\mathbb{G}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{G}''_{j,(i_1,\dots,i_q)}$ their cumulative distribution functions under the null, conditionally to $E(i_1, \dots, i_q)$.

We will also make use of a third assumption, here again fulfilled by both s^{HSIC} and s^{ridge}:

($\mathbf{A_3}$) **Two-homogeneity:** It holds that $s(\boldsymbol{z}, t\boldsymbol{y}) = t^2 s(\boldsymbol{z}, \boldsymbol{y})$ for all $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$ and all $t \geq 0$.

Of note, normalizing the association score with respect to the labels does not affect the selection:

$$\forall \boldsymbol{y} \in \mathcal{E}\,,\ \arg\max_{\boldsymbol{z}\in\mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{y}) = \arg\max_{\boldsymbol{z}\in\mathcal{Z}} \frac{s(\boldsymbol{z}, \boldsymbol{y})}{\|\boldsymbol{y}\|_2^2}\,. \tag{4.27}$$

• A simpler case: $\boldsymbol{\mu} = \mathbf{0}$

If $\boldsymbol{\mu} = \mathbf{0}$, the distribution of the normalized statistics does not depend on σ, and the empirical cumulative distribution functions of normalized scores obtained by sampling under $\boldsymbol{\mu} = \mathbf{0}$ and any σ still provides a valid inference procedure, as stated by the following proposition.

Proposition 4.3.3.
Let s be an association score such that ($\mathbf{A_2}$) and ($\mathbf{A_3}$) hold. Let $T'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j\boldsymbol{y})/\|\boldsymbol{y}\|_2^2$ and $T''_j = \max_{\boldsymbol{z}\in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j\boldsymbol{y})/\|\boldsymbol{y}\|_2^2$, formed from $\boldsymbol{y}$ sampled from (4.1) with mean $\boldsymbol{\mu} = \mathbf{0}$, and any variance $\sigma > 0$. Then for all $\sigma' > 0$, their conditional null distributions $\mathbb{G}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{G}''_{j,(i_1,\dots,i_q)}$ with mean $\mathbf{0}$ and variance σ' verify:

$$\mathbb{G}'_{j,(i_1,\dots,i_q)}(T'_j) \sim \mathcal{U}(0,1) \text{ and } \mathbb{G}''_{j,(i_1,\dots,i_q)}(T''_j) \sim \mathcal{U}(0,1)\,. \tag{4.28}$$

Proof. Let us consider two different normal models as defined in (4.1) under the global null hypothesis "$\boldsymbol{\mu} = \mathbf{0}$" and given by

$$\boldsymbol{y}^{(1)} = \sigma^{(1)}\boldsymbol{\epsilon}^{(1)} \text{ and } \boldsymbol{y}^{(2)} = \sigma^{(2)}\boldsymbol{\epsilon}^{(2)}\,.$$

Then we have:

$$\frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j\boldsymbol{y}^{(1)})}{\|\boldsymbol{y}^{(1)}\|_2^2} \sim \frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j\boldsymbol{y}^{(2)})}{\|\boldsymbol{y}^{(2)}\|_2^2} \text{ and } \frac{s\left(\boldsymbol{z}, \boldsymbol{\Pi}''\left((\boldsymbol{z}_\ell)_{\ell\neq j}\right)\boldsymbol{y}^{(1)}\right)}{\|\boldsymbol{y}^{(1)}\|_2^2} \sim \frac{s\left(\boldsymbol{z}, \boldsymbol{\Pi}''\left((\boldsymbol{z}_\ell)_{\ell\neq j}\right)\boldsymbol{y}^{(2)}\right)}{\|\boldsymbol{y}^{(2)}\|_2^2}$$

The proof directly follows assumption ($\mathbf{A_3}$) applied with $t = \|\boldsymbol{y}^{(\cdot)}\|^2$. Proposition 4.3.3 is complementary to Proposition 4.3.2 and provides a selective inference procedure when σ is unknown, under the special hypothesis $\boldsymbol{\mu} = \mathbf{0}$. □

• For any mean $\boldsymbol{\mu}$

Our final result investigates the testing procedures for the general null hypothesis (4.21, 4.22) — not restricted to $\boldsymbol{\mu} = \mathbf{0}$ — with an unknown σ. Recall that the decision rule is to reject the null hypothesis if the observed value of the statistic is greater than a given threshold t. We show that choosing t to be a quantile for the global null hypothesis ($\boldsymbol{\mu} = \mathbf{0}$) leads to a calibrated test — for the type I error, see (4.29) — in the non-selective framework.

Proposition 4.3.4 (Global null achieves lowest observed significance)**.**
Let $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q\}$ *be* q *sequence motifs. Let* $s(\cdot, \cdot)$ *be a score such that* $(\mathbf{A_1})$, $(\mathbf{A_2})$ *and* $(\mathbf{A_3})$ *hold. Let* $\boldsymbol{\mu} \in \mathcal{E}$ *be such that:*

$$\mathbb{H}_0 \ : \ \text{“}s(\boldsymbol{Z}, \boldsymbol{\mu}) = 0\text{”}\,.$$

Then

$$\forall t > 0,\ \sup_{\boldsymbol{\mu} \in \mathbb{H}_0} \mathbb{P}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon})}{\|\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}\|_2^2} \geq t\right] = \mathbb{P}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{\epsilon})}{\|\epsilon\|_2^2} \geq t\right]\,. \tag{4.29}$$

Proof. This proof makes an ad-hoc use of Anderson's theorem (Theorem 1, Anderson, 1955) on a symmetric convex cone (whereas it is usually devoted to symmetric convex bodies).

Consider the orthogonal decomposition

$$\mathcal{E} = \mathcal{S} \bigoplus \mathcal{T},$$

where $\mathcal{T}$ is the span of $\boldsymbol{\mu}$ and $\mathcal{S} = \mathcal{T}^{\perp}$. Consider $\boldsymbol{y} \in \mathcal{E}$ and its orthogonal decomposition $\boldsymbol{y} = \boldsymbol{s} + t\boldsymbol{e}$ where $\boldsymbol{e} = \boldsymbol{\mu}/\|\boldsymbol{\mu}\|_2$ is a unit norm vector that spans $\mathcal{T}$. Let $\tau > 0$ and note that it is enough to prove that

$$\mathbb{P}_{\boldsymbol{\mu}}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{Y})}{\|\boldsymbol{Y}\|_2^2} \leq \tau\right] \geq \mathbb{P}_{\mathbf{0}}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{Y})}{\|\boldsymbol{Y}\|_2^2} \leq \tau\right]\,,$$

where $\boldsymbol{Y}$ is a random variable with the same distribution as $\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ (respectively $\sigma\boldsymbol{\epsilon}$) on the probability space defined by $\mathbb{P}_{\boldsymbol{\mu}}$ (respectively $\mathbb{P}_{\mathbf{0}}$).

Using Assumptions $(\mathbf{A_2})$ and $(\mathbf{A_3})$, one gets that $s(\boldsymbol{Z}, \boldsymbol{y}) = s(\boldsymbol{Z}, t\boldsymbol{\mu}/\|\boldsymbol{\mu}\|_2 + \boldsymbol{s}) = s(\boldsymbol{Z}, \boldsymbol{s})$. By the Pythagorean theorem, we then have the following equality:

$$\left\{\boldsymbol{y} \ : \ \frac{s(\boldsymbol{Z}, \boldsymbol{y})}{\|\boldsymbol{y}\|_2^2} \ \leq \tau\right\} = \left\{(t, \boldsymbol{s}) \ : \ s(\boldsymbol{Z}, \boldsymbol{s}) \leq \tau(t^2 + \|\boldsymbol{s}\|_2^2)\right\}\,.$$

By orthogonality, we can note that the law of $\boldsymbol{s}$ is independent from t and that this law is a centered Gaussian multivariate law. We deduce that the aforementioned probabilities are of the form

$$\mathbb{P}_{\boldsymbol{\mu}}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{Y})}{\|\boldsymbol{Y}\|_2^2} \ \leq \tau\right] \ = \int_0^{\infty} w_{\mathbf{0}}(t)\phi_{\boldsymbol{\mu}}(t)\mathrm{d}t\,,$$

where

$$w_{\mathbf{0}}(t) = \mathbb{P}_{\mathbf{0}}\left[s(\boldsymbol{Z}, \boldsymbol{s}) \leq \tau(t^2 + \|\boldsymbol{s}\|_2^2)\right] = \mathbb{P}_{\mathbf{0}}\left[s(\boldsymbol{Z}, \boldsymbol{s})/\tau - \|\boldsymbol{s}\|_2^2 \leq t^2\right]\,,$$

$$\sqrt{2\pi}\phi_{\boldsymbol{\mu}}(t) = \exp(-(t - \mu_e)^2/2) + \exp(-(t + \mu_e)^2/2)\,,$$

with $\mu_e = \langle \boldsymbol{e}, \boldsymbol{\mu} \rangle = \|\mu\|_2$. Using the right-hand side of the aforementioned definition of $w_{\mathbf{0}}$, one can check that $w_{\mathbf{0}}$ is a CDF and hence it has generalized inverse, denoted $w_{\mathbf{0}}^{-1}$.

The Fubini's equality yields:

$$\begin{aligned}\mathbb{P}_{\boldsymbol{\mu}}\left[\frac{s(\boldsymbol{Z},\boldsymbol{Y})}{\|\boldsymbol{Y}\|_2^2} \leq \tau\right] &= \int_0^\infty w_{\mathbf{0}}(t)\phi_{\boldsymbol{\mu}}(t)\mathrm{d}t \\ &= \int_0^\infty \int_0^1 \mathbf{1}_{\{u\leq w_{\mathbf{0}}(t)\}}\mathrm{d}u\phi_{\boldsymbol{\mu}}(t)\mathrm{d}t \\ &= \int_0^\infty \int_0^1 \mathbf{1}_{\{w_{\mathbf{0}}^{-1}(u)\leq t\}}\ \mathrm{d}u\phi_{\boldsymbol{\mu}}(t)\mathrm{d}t \\ &= \int_0^1 \int_{w_{\mathbf{0}}^{-1}(u)}^\infty \phi_{\boldsymbol{\mu}}(t)\mathrm{d}t\mathrm{d}u\,.\end{aligned}$$

By Anderson's theorem, the measure of the interval $[-w_{\mathbf{0}}^{-1}(u), w_{\mathbf{0}}^{-1}(u)]$ for the centered Gaussian density is greater than the one for a non-centered Gaussian density with the same variance. As a results, we deduce that

$$\int_{w_{\mathbf{0}}^{-1}(u)}^\infty \phi_{\boldsymbol{\mu}}(t)\mathrm{d}t \geq \int_{w_{\mathbf{0}}^{-1}(u)}^\infty \phi_{\mathbf{0}}(t)\mathrm{d}t\,,$$

which achieves the proof. □

Proposition 4.3.4 shows that data-split produces a calibrated procedure for testing the general null hypotheses (4.21, 4.22) when sampling the test statistics (4.26) under the global null ($\boldsymbol{\mu} = \mathbf{0}$).

We could not prove an equivalent statement for conditional null hypotheses, and Proposition 4.3.4 therefore does not guarantee the validity of a selective inference procedure sampling under the global null ($\boldsymbol{\mu} = \mathbf{0}$). Yet, we used it as a heuristic justification of SEISM and we observed that it leads to empirically calibrated procedures, see Subsection 4.4.1.

In view of Proposition 4.3.4 and its proof, one can see that the alternatives $\boldsymbol{\mu}$ such that $\|\boldsymbol{P}_q\boldsymbol{\mu}\|_2/\|\boldsymbol{\mu}\|_2$ is large have small power. As the selection procedure described in Chapter 3 achieves good results, the chosen motifs $\boldsymbol{Z}$ should capture the principal components of $\boldsymbol{\mu}$, and therefore are such that $\|\boldsymbol{P}_q\boldsymbol{\mu}\|_2/\|\boldsymbol{\mu}\|_2$ should be small.

4.4 Empirical evaluation of SEISM

4.4.1 Statistical validity and performance

In order to assess the statistical validity and the performance of the SEISM procedure with the different strategies described above, we derived the following protocol

1. We first create some simulated datasets:
    - We draw one sequence motif $\tilde{\boldsymbol{z}}$ with length $k = 8$ for each simulated dataset using a uniform distribution of $\mathcal{Z}$ restricted to motifs with an information level fixed at 10 bits.

- Then, we draw a set of $n = 30$ biological sequences $\boldsymbol{X}$ as follows: all sites are generated according to a uniform distribution over $\{A, C, G, T\}$ for all sequences. In half of those sequences, we draw one k-mer according to the categorical model (3.31) parameterized by $\tilde{\boldsymbol{z}}$, and insert it at a random position.
- The phenotypes $\boldsymbol{y}$ are drawn from $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{C}_n)$ to generate data under the null hypothesis for calibration experiments, and from $\mathcal{N}(\boldsymbol{\varphi}^{\tilde{\boldsymbol{z}}, \boldsymbol{X}}, \sigma^2 \boldsymbol{C}_n)$ to generate data under the alternative for experiments on statistical power, with $\sigma = 0.1$ in both cases.

2. We then run the SEISM procedure:
   - We select $q = 2$ sequence motifs.
   - We test them using either a data-split strategy or the conditional inference one. For both strategies, the distribution from which the replicates are sampled uses the empirical variance of $\boldsymbol{y}$ as variance parameter. Although any choice for this parameter leads to a valid procedure, as described in Subsection 4.3.4, we make this choice for numerical stability considerations. For the conditional inference strategy, we sample 50 000 replicates under the conditional null hypothesis using the hypersphere direction sampler, after 10 000 burn-in iterations.

The top row of Figure 4.5 shows the Q-Q plot of the distribution of quantiles for the p-values obtained across 1 000 datasets under the null hypothesis for the data-split strategy and 100 datasets for the hit-and-run sampler one. All the data points are well-aligned with the diagonal, which confirms the correct calibration of both the data-split and hypersphere direction sampling strategies, either considering the best motif or the center of the mesh (4.21, 4.22) and regardless of the size parameter m of the mesh.

The bottom row of Figure 4.5 shows the same Q-Q plot on data generated under the alternative hypothesis. From this figure, we observe that on small datasets, the post-selection inference strategy is more powerful than the data-split one, regardless of the size of the mesh m, or the choice concerning the definition of the null hypothesis.

The deviation observed on the curves associated with the selective inference procedure for the second motif is due to the presence of a weak residual remaining signal after the first motif, as a result of an imperfect selection step. Testing it with the best motif in the mesh captures this signal, resulting in curves under the diagonal. By contrast, focusing on the center of the meshes leads to testing motifs that do not capture this signal, placing us in the conservative situation described at the end of Subsection 4.3.4. The residual signal is not well explained by the mesh's centers, and thus its component on the orthogonal of the span of the activation vector of the second motif is important. The larger the mesh, the farther its center is to the selected motif and thus the less signal it captures, explaining the differences between the two curves.

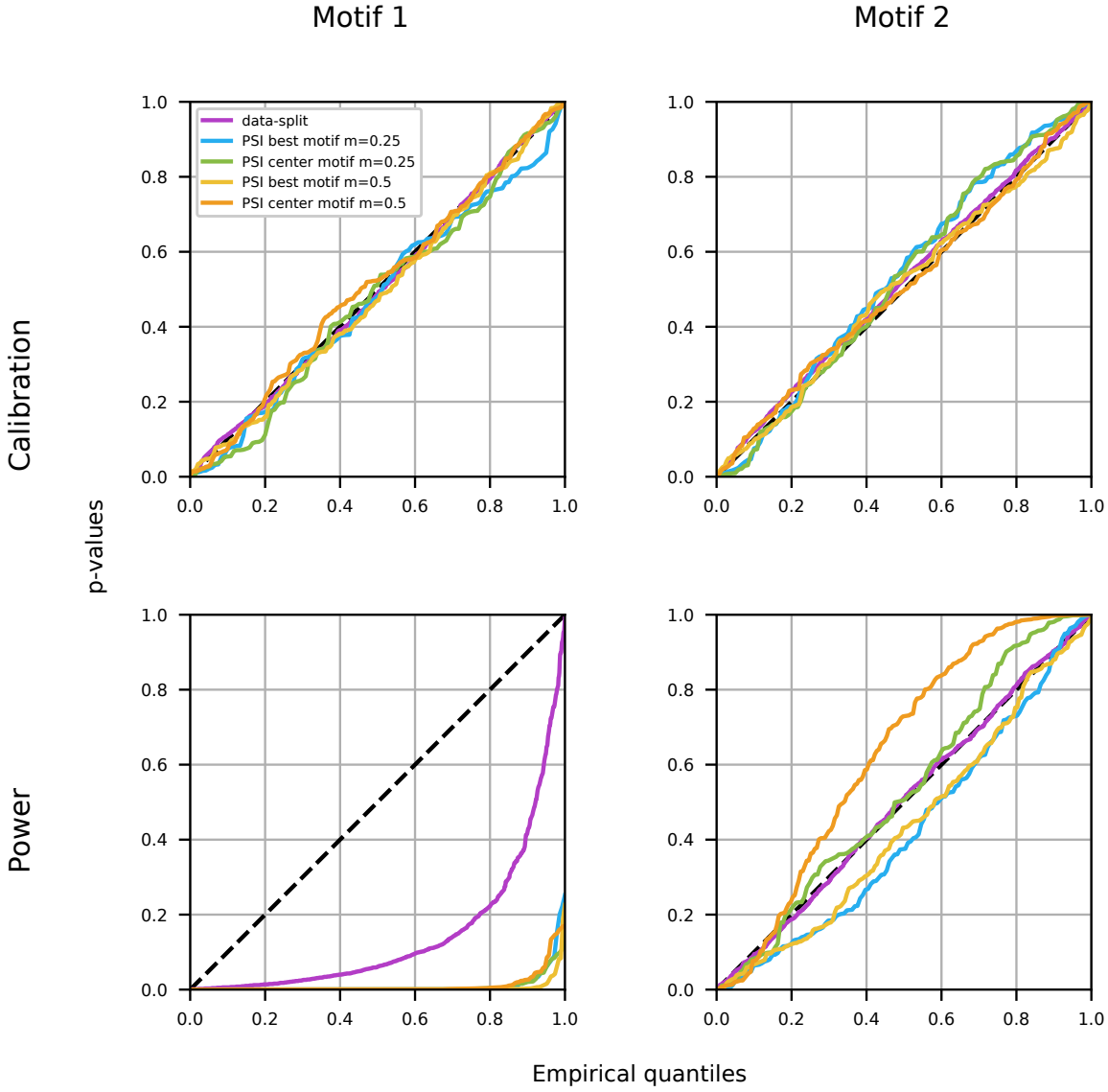


Figure 4.5: Q-Q plots obtained by applying the SEISM procedure with data-split and hit-and-run strategies with different parameters to select two motifs and test their association with an outcome. **Top:** data simulated under the null hypothesis. The proximity between the quantiles of the obtained p-values and those of the uniform distribution confirms that all SEISM strategies presented in this work are correctly calibrated. **Bottom:** data simulated under an alternative hypothesis, where the outcome depends on the activation $\boldsymbol{\varphi}^{\tilde{\boldsymbol{z}},\boldsymbol{X}}$ of a single *true* motif $\tilde{\boldsymbol{z}}$ in the sequences. The distributions of the p-values computed with the post-selection inference (PSI) strategies have a larger deviation to the uniform distributions than the p-values computed with the data-split strategy (purple).

4.4.2 Impact of the hyperparameters on computation costs

This section serves as an overview of how various user-specified parameters impact the computation time required by the post-selection inference procedure.

As discussed in Subsection 4.3.1.2, the hit-and-run algorithm is actually a rejection sampler. Its overall computation cost depends mainly on two characteristics: the cost of the selection step, that is the cost for selecting q motifs for a given $\boldsymbol{y}'$, and the acceptance rate.

Although some parameters affect the selection cost, the acceptance rate is primarily responsible for determining if a user-specified combination of parameters results in a tractable configuration for the conditional inference method in a reasonable amount of time.

This rate is high compared with a naive rejection sampler over $\mathcal{E}$, as the hit-and-run strategy reduces the dimension over which the rejection step is performed: from n with a naive sampler to 1. Nonetheless, some parameters may have a major impact on the rejection rate. To clarify it, we studied in Figure 4.6 the impact of several user-specified parameters — the number of motifs to be discovered q, the precision of the meshes m, the regularization parameter λ and the number of computation cores allowed during the rejection step of the hit-and-run sampler.

- Although the number of motifs to be found by SEISM undoubtedly affects the selection cost, we can roughly consider that this relationship is linear. The upper left curve in Figure 4.6, however, demonstrates that the influence of q on the overall computation costs is super-linear, in line with an exponential growth of the number of distinct selection events one may describe with a fixed mesh size m when q grows. As a result, the post-selection process quickly becomes intractable for testing more than a few motifs.

- We make a similar observation for mesh precision: computation time grows exponentially with the number of bins m used to define the meshing. This can be explained by the exponential relationship between the number of bins and the number of different meshes (and thus the rejection rate). Of note, mesh precision has no impact on the selection time, and therefore the difference in computation time is entirely explained by the acceptance rate.

- We observe that the greater the regularization parameter λ, the lower the computation time. This can be explained by detailing its impact on the rejection rate. To understand it, it is necessary to note that the motifs are not selected over $\mathcal{Z}$, but over a less constrained set $\mathcal{Z}_{\text{uc}}$, as described in Chapter 3 Subsection 3.3.3. They are only projected onto $\mathcal{Z}$ at the end of the whole procedure, to ease their interpretation.

  The meshes are then defined over a vectorial space, leading to an infinite number of meshes. Compared to a small regularization parameter, a higher λ favors motifs resulting in a $\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ with a higher norm (3.6). With regard to the activation function, such motifs are located closer from the k-mers, and thus from $\mathcal{Z}$, see Figure 4.7.

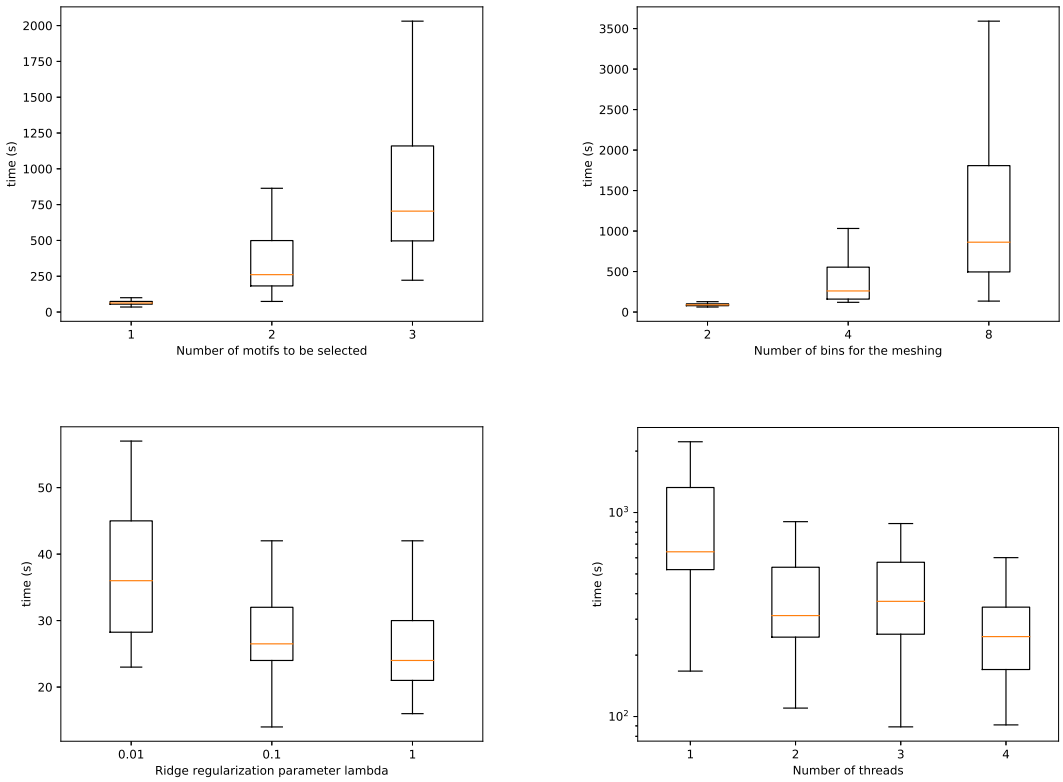


Figure 4.6: Impact of different parameters on the computation time for the post-selection inference procedure. **Upper left:** Impact of the number of motifs to be discovered q. **Upper right:** Impact of the number of bins defining the meshes m. **Bottom right:** Impact of the ridge regularization parameter λ. **Bottom right** (Log scale) Impact of the number of threads over which the hit-and-run sampler is parallelized.

Then λ has no effect on the number of existing meshes, but impacts the number of *acceptable* ones, in the sense that they have a reasonable probability to be selected. A lower λ leads to better selection performances, but to a higher number of acceptable meshes, and thus to a lower acceptance rate. We empirically set $\lambda = 0.01$ to provide a good trade-off.

- Finally, the rejection sampling step can be parallelized over several computation cores, which accelerates the whole procedure, as described in Subsection 4.3.1.2. As long as the acceptance rate is small enough, using j cores to parallelize the rejection step roughly divides the computation time by a factor j.

We can clearly identify limitations inherent to the use of the selective inference procedure. Although it is more powerful than the data-split approach, it cannot be used in every situation. The data-split approach does indeed not include any rejection step, and the only factor influencing its overall computation time is the selection time, only marginally influenced by the aforementioned parameters.

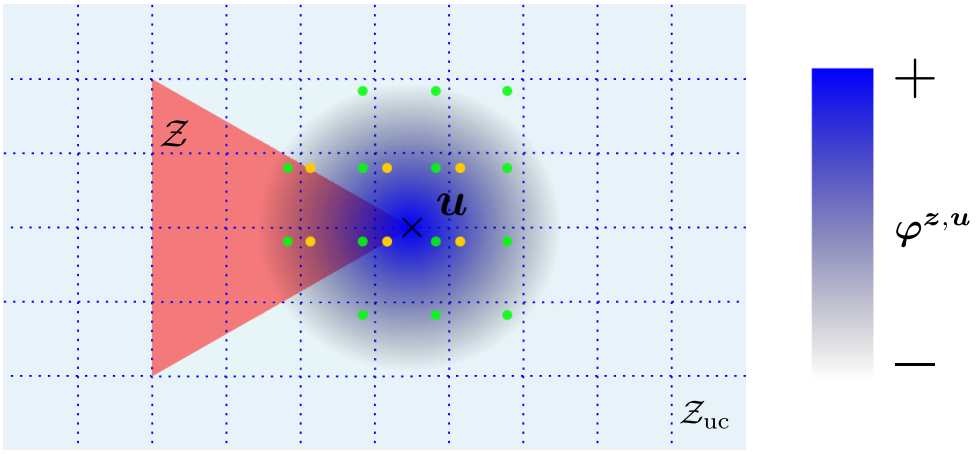


Figure 4.7: Diagram of the simplex $\mathcal{Z}$ (red) and vectorial space $\mathcal{Z}_{\text{uc}}$ (light blue). The meshes are represented by dotted lines, and a k-mer $\boldsymbol{u}$ is located on a vertex of $\mathcal{Z}$. The function $\varphi : \boldsymbol{z} \mapsto \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{u}}$ is represented in dark blue. We say that a mesh is *acceptable* if the activation function goes beyond a given threshold in this mesh. Indeed, a mesh needs to contain a motif with a high score, and then a high activation function, to be selected. If we consider two thresholds $t_1 < t_2$, we observe that 6 meshes contain at least one motif $\boldsymbol{z}$ with $\varphi(\boldsymbol{z}) > t_2$ (indicated with yellow dots), while 14 meshes contain motifs with $\varphi(\boldsymbol{z}) > t_1$ (green dots). A lower λ tends to lower this threshold.

4.4.3 Impact and choice of the number of burn-in and replicates

As described in Subsection 4.3.1.2, the hit-and-run sampler generates a Markov chain over the selection event E. It requires a large number of burn-in iterations, to reduce the dependence on the original phenotype $\boldsymbol{y}$ and a large number of replicates to provide a good approximation of the target distribution and to address the dependence between consecutive replicates. In this section, we give insights on the impact of the number of burn-in and replicates on the statistical validity and on the power of the test, and on the stability of the resulting p-values.

• Impact of the number of burn-in iterations and replicates on the calibration and statistical power

First, we generate 200 datasets under the null (with $n = 50$), and apply the SEISM procedure with different number of burn-in and replicates. The corresponding Q-Q plots are represented in Table 4.2. We can then make two different observations:

- The number of burn-in iterations has no impact on the validity of the procedure. The p-values distributions obtained with 0 burn-in are close to uniforms.
- On the contrary, using too few replicates leads to S-shaped curves — the p-values are too extreme, either too high or too low, except when the number of burn-in is set to zero. That is because with a non-zero number of burn-in, the first replicates belong to a small area of the selection event located far from the initial $\boldsymbol{y}$. The $\boldsymbol{y}'$ in this region have similar scores (as the $\boldsymbol{y}'$ are close from one another): either higher or lower scores than $s(\boldsymbol{z}, \boldsymbol{y})$. If we do not allow a sufficient number of replicates, only

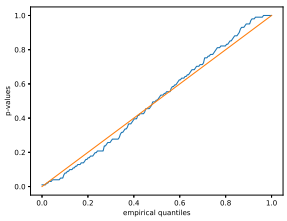
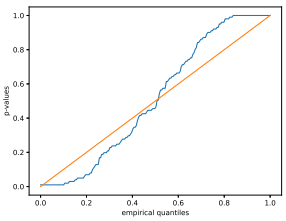
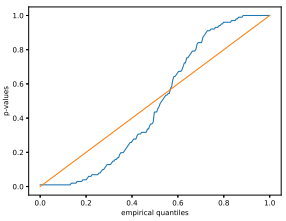
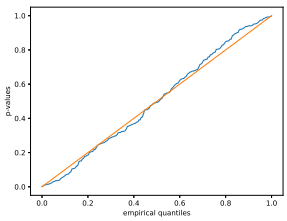
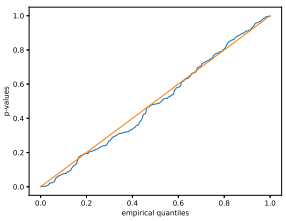
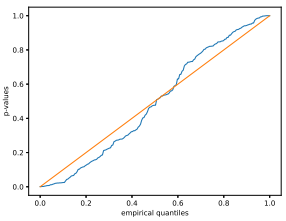
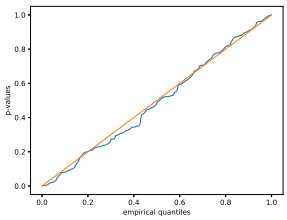
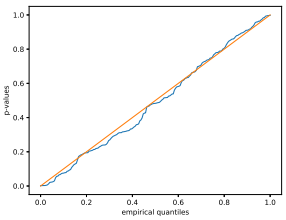
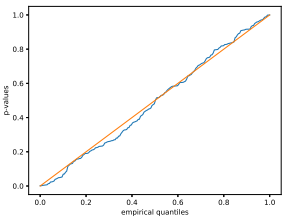
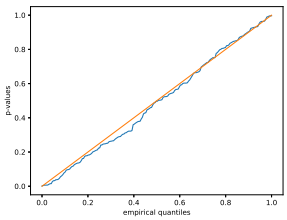
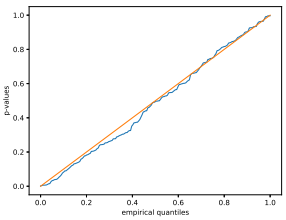
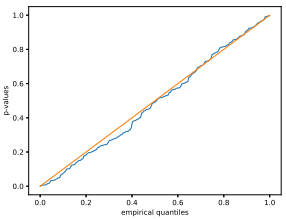
Replicates	0 burn-in	1,000 burn-in	10,000 burn-in
100			
1,000			
10,000			
50,000			

Table 4.2: Q-Q plots for a various number of burn-in iterations and replicates obtained by applying SEISM on 200 simulated datasets under the null hypothesis.

this small area will be sampled. This leads to p-values close to 0 or 1 respectively. With zero burn-in, the area contains $\boldsymbol{y}'$ close to $\boldsymbol{y}$, therefore leading to similar scores and to uniform p-values.

Then, we generate 200 datasets under an alternative hypothesis, by adding some signal in $\boldsymbol{y}$. The results are compiled in Table 4.3. Once again, we can describe the impact of the number of burn-in and replicates:

- A low number of burn-in tends to reduce the statistical power. Indeed, the first replicates will then be close to the initial $\boldsymbol{y}$, and consequently they have close scores. Keeping these points in the distribution tends to bring it closer to the uniform.
- A high number of replicates can mitigate the impact of a low number of burn-in iterations, by reducing the proportion of points close to $\boldsymbol{y}$. But if the number of

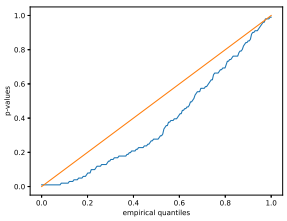
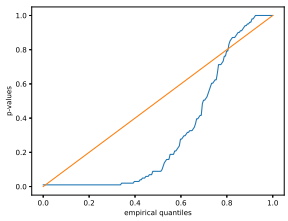
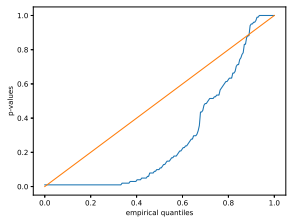
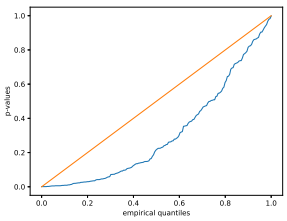
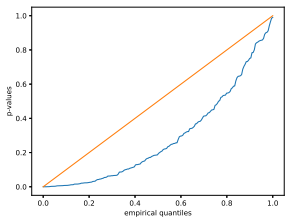
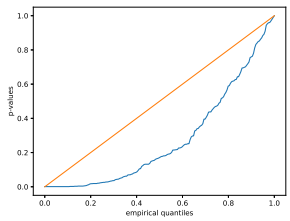
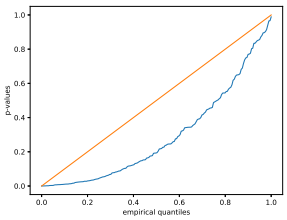
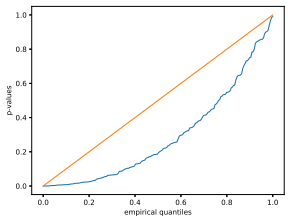
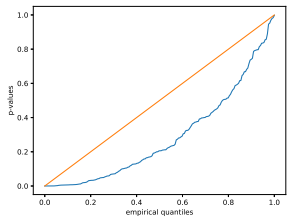
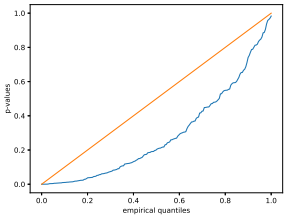
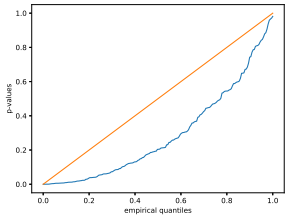
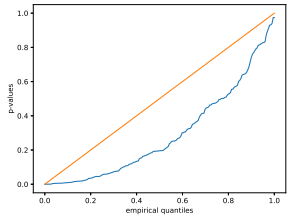
Replicates	0 burn-in	1,000 burn-in	10,000 burn-in
100			
1,000			
10,000			
50,000			

Table 4.3: Q-Q plots for a various number of burn-in iterations and replicates obtained by applying SEISM on 200 simulated datasets with some signal.

burn-in is sufficient, then the number of replicates seems to have a very limited impact on the power.

Although the number of replicates has a limited impact on the power, a low number can lead to unstable p-values, as described below.

• **Impact of the dimension n on the required number of replicates**

Because it defines the dimension of the selection event E, the number of samples n in the dataset has an impact on the required number of replicates required to approximate the uniform distribution. We are confronted here with a curse of dimensionality, as the volume of the space increases rapidly with the number of dimensions. Therefore, it is necessary to have a large number of samples to sufficiently fill this space and thus correctly approximate the distribution. This may then have an impact on the p-values.

In order to illustrate it, we create a random dataset under the null, with 50 sequences.

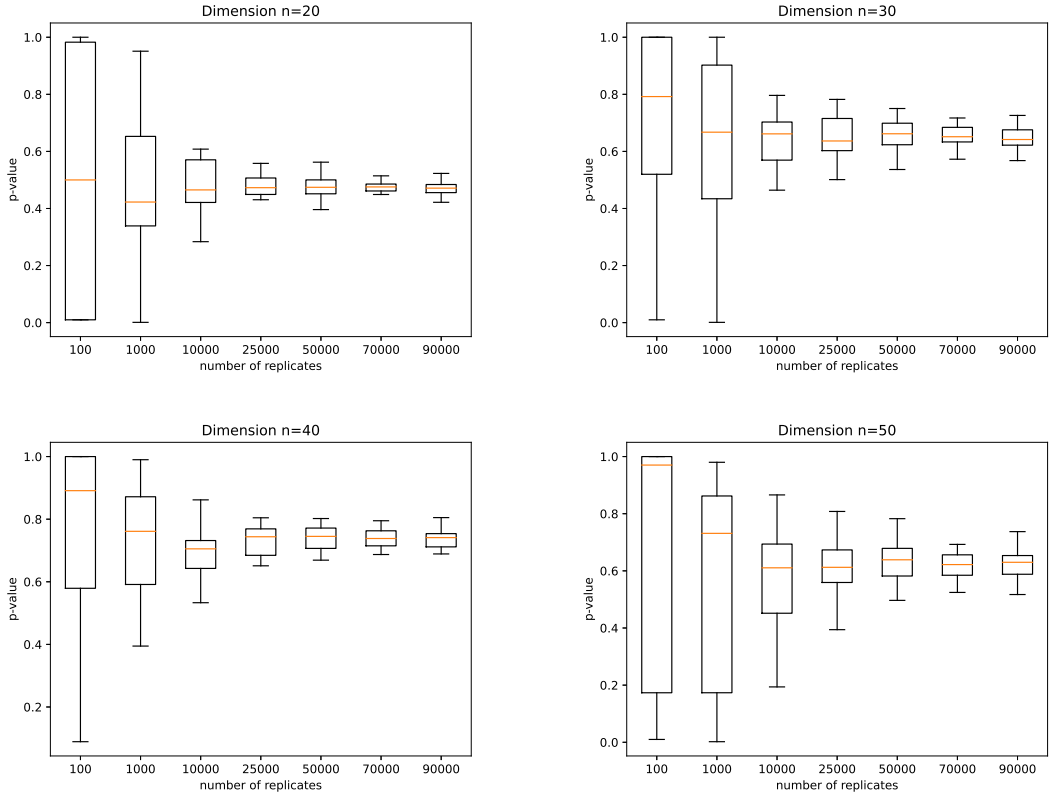


Figure 4.8: Variations of the p-values for different number of samples n and different numbers of replicates.

From this dataset, we create 3 new datasets, by randomly sampling 20, 30 and 40 individuals. We are then left with 4 datasets, with $n \in \{20, 30, 40, 50\}$. We apply 20 times the SEISM procedure on each dataset, with 10,000 burn-in iterations. As the selection procedure is stable from one run to another, the selected motifs are identical from one run to the next. But the test procedure involves random steps in the hit-and-run sampler. In particular, the replicates vary from one run to the next, resulting in different p-values.

With a sufficient number of replicates — a good approximation of the conditional distribution — the variations of the p-values should be limited. But the number of replicates required to have stable p-values is dependent on n. To illustrate this, we computed the p-values for each run with a growing number of replicates, and the results are compiled in Figure 4.8.

First, we observe that the variability of the p-values decreases with the number of replicates. Second, it shows that this decrease is faster for small datasets ($n = 20$) than for big ones ($n = 50$). Those two observations are consistent with the theory, and simply allow us to have some intuition on the behavior of the p-values.

4.4.4 Robustness of the Gaussian assumption: end-to-end application on real data

We now have a valid procedure to select sequence motifs associated with a phenotype, thanks to an adapted version of one-layer CNNs, and to test those trained filters. We now want to apply SEISM to a given real-world dataset. One question remains: even if it can be applied with no a priori for $\boldsymbol{\mu}$ and σ, is the dataset compatible with the Gaussian model (3.32)? Is the test procedure valid for this particular dataset?

In fact, Gaussian phenotypes are not expected in the vast majority of datasets. Obviously, classification problems do not fit this model, but even for continuous datasets the Gaussian assumption may be challenging. In this section, we provide a method to check whether the SEISM procedure is valid for a given dataset or not.

To this end, we will work on a real case study, using the ChIP-seq dataset from (Chatagnon et al., 2015). This experiment seeks to understand some of the mechanisms underlying cell differentiation. It is known that retinoic acid plays a role in these mechanisms, and thus the retinoic acid receptor (RAR) is an interesting transcription factor. We are then looking for the binding motifs for RAR.

We apply the SEISM procedure on this dataset, to detect 4 motifs and test them using a data-split approach

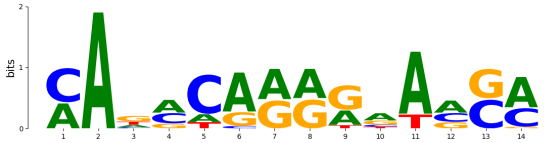
```
seism analysis fasta_file.fa --nb-motifs 4 --min-motifs-length 8
    --max-motifs-length 16 --association-score ridge --ridge-lambda 0.01
    --inferer-type data_split --ds-nb-replicates 1000 --ds-split-ratio 0.1
```

The selected motifs are represented in Table 4.4 with their respective p-values.

Motif 1 $p = 10^{-3}$

Motif 2 $p = 0.013$

Motif 3 $p = 0.47$



Motif 4 $p = 0.23$

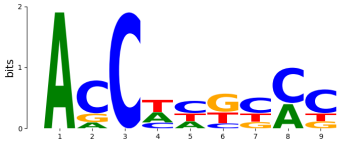


Table 4.4: Motifs and p-values obtained using the SEISM procedure (data-split) on the real ChIP-seq dataset.

We can already note that the first motif found, with the lowest p-value, actually corresponds to a known binding motif for RAR (Balmer & Blomhoff, 2005), see Figure 4.9.

But are the obtained p-values valid, since the Gaussian hypothesis does not necessarily hold for this dataset? By plotting the distribution of the phenotypes in Figure 4.10, we realize that the empirical distribution does not look like a Gaussian one (and unsurprisingly,

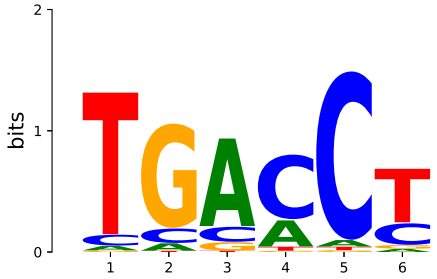


Figure 4.9: Known binding motif for RAR.

a normality test such as D'Agostino-Pearson leads to an extremely low p-value$< 10^{-100}$). In this dataset, the outcome $\boldsymbol{y}$ contains the $-\log_{10}$ of p-values resulting from a test to determine whether the corresponding sequence is associated with a high number of bindings with RAR.

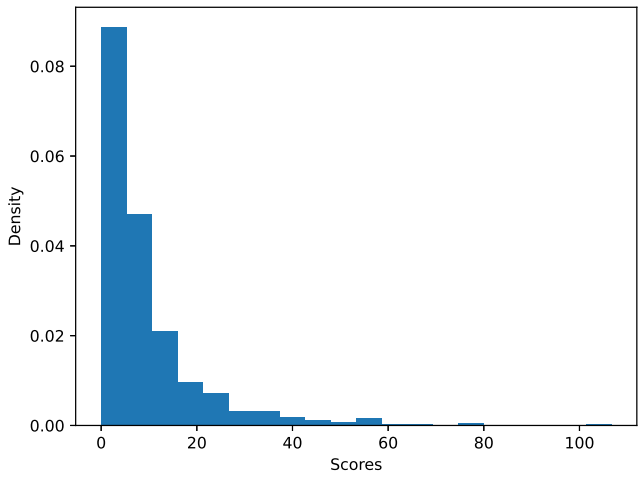


Figure 4.10: Empirical probability density of the phenotypes in the ChIP-seq dataset.

Although this problem is not limited to SEISM, we propose here an approach that determines whether the Gaussian assumption is valid for a given dataset $(\boldsymbol{X}, \boldsymbol{y})$ or not for the SEISM analysis. This method can be divided into 3 steps:

1. Create N datasets $(\boldsymbol{X}, \boldsymbol{y}^{(i)})$ derived from the original one. The sequences are unchanged, but the labels are randomly permuted versions of $\boldsymbol{y}$. If there exists motifs $\boldsymbol{Z}$ that are effectively associated with $\boldsymbol{y}$, then this permutation step breaks those associations, thus enforcing that the permuted datasets are under the null. But it will maintain the original probability distribution of $\boldsymbol{y}$.

2. Run the whole SEISM procedure on each of those permuted datasets, and collect the p-values.

3. Draw a Q-Q plot with those p-values. If the distribution is close to the uniform, then it validates the Gaussian assumption, and the p-values obtained on the non-

permuted dataset actually reflect the association between the discovered motifs and $\boldsymbol{y}$.

We applied this methodology to our dataset, and created 1 000 permuted versions. The obtained Q-Q plot is in Figure 4.11, and it confirms that the SEISM procedure is valid for this dataset.

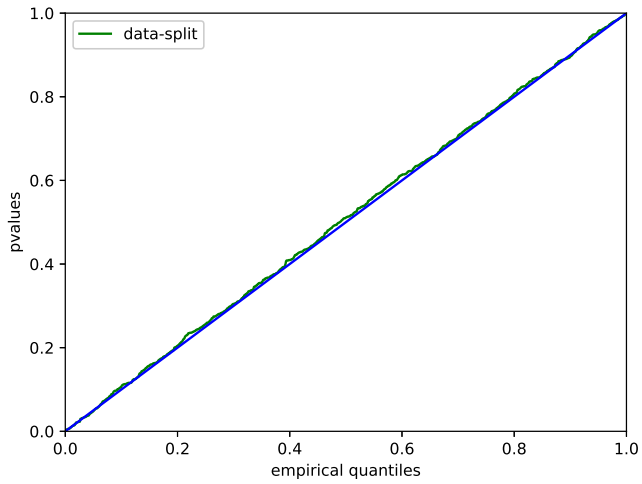


Figure 4.11: Q-Q plot obtained by applying the SEISM procedure to permuted versions of the ChIP-seq dataset

To conclude this chapter, we have introduced a new analytical framework, allowing to test the interpretable features extracted from a machine learning model. This framework is generic enough to be used with different features, as long as an association score can be defined (with a few conditions) between the feature and the phenotype. We were interested in a very wide range of possible associations.

We provided a proof of concept on a particular case, the motifs extracted from a CNN. Although in SEISM, the selection procedure is coupled with the test, it is by no means a necessity. For instance, the SEISM test procedure can be used to test motifs extracted by TF-MoDISco, with the appropriate modification of the selection event.

Finally, we have attempted to reduce the number of assumptions required to apply SEISM, either on the parameters of the null distribution, or this null distribution itself. We therefore hope that this work will facilitate the application of SEISM on real datasets.

Conclusion and future works

In this work, we introduced a new approach to quantify the association between interpretable features extracted from a trained neural network and the outcome of this network. We have indeed shown that, despite the fact that these networks were originally designed with the aim of correctly predicting a biological trait, they can be used to correctly select features associated with the phenotype. Thanks to slight modifications, or to methods coming from the explainable artificial intelligence literature, they can achieve similar performance for feature selection tasks than state of art explanatory methods from the computational biology literature.

Our procedure relies on the post-selection framework and formalizes the network training as a feature selection step. Along the way, we addressed general problems related to selective inference over composite hypotheses, which has implications beyond testing of features extracted on trained neural networks. Our strategy to normalize the statistic to make it scale-free could easily be transferred for testing the association of kernels with a trait (Slim et al., 2019), or to previous selective inference frameworks for testing groups of variables using sampling strategies (Reid et al., 2015).

Nonetheless, SEISM represents a first step in the direction of quantifying the uncertainty of explicability in machine learning. It can be considered as a proof of concept, and its development suggests lots of questions and directions to study.

• Which activation model for the motifs?

The selection step of SEISM formalizes that training a one-layer CNN is equivalent to selecting a finite set of PWMs, or sequence motifs, that have a maximal association to the outcome for some particular score. This formalization also highlights the specific way by which CNNs with exponential activation functions parametrize the distribution of k-mers at a binding site. As discussed in Chapter 3 Subsection 3.2 and Section 3.7, although the PWMs returned by most bioinformatics models represents a categorical distribution, the convolution filters parametrize a Gaussian distribution. In practice, this difference leads to discrepancies between the trained convolution filters and the motifs learned using categorical likelihoods — including those offered by databases and often used as ground

truth. This observation suggests alternative set of constraints for convolution filters, for instance instead of constraining a motif $\boldsymbol{z}$ to belong to $\mathcal{Z}$, we could try to constrain the pointwise exponential of $\boldsymbol{z}$. We could also try to modify the activation function, from the classical exponential activation to a categorical one.

• Extending SEISM to other biological features

The test and the selection procedures are linked by the selection event and the resulting conditioning. SEISM was only implemented to test sequence motifs, obtained with a modified version of a one-layer CNN. However, the test method is general enough to be theoretically applied to different selection methods.

The theoretical foundation of our test method, as well as how it was implemented in the code provided with this thesis, were designed to be modular. That is, the results provided in Chapter 4 and the code should be easily adaptable to new genetic variants and selection procedures. In a very simple way, the framework described in Chapter 4 with a data-split strategy provides a straightforward method for testing any feature, as long as an association score can be defined between the feature and the trait. But even for the conditional inference method, we tried to explicit the requirements for a feature selection method to be compatible with our test procedure, and the code has been developed trying to be as generic as possible towards the selection method. Also obviously implementation issues will arise when we will try to apply it with a different selection method, we hope that having developed this code with modularity in mind would decrease the occurrence of these issues.

The most important step in extending SEISM to other biological features or selection methods is to formulate the training of the network and the extraction of the feature as a feature selection problem, and to formalize the association between those features and the phenotype, meeting the assumptions described in Chapter 4. For instance, we may test motif interactions derived from convolutional-attention networks (Ullah & Ben-Hur, 2021), a (motif, position) couple as selected in (Ditz et al., 2022) or motifs extracted by TF-MoDISco (Shrikumar et al., 2018). For this latter case, we can note that our inference procedure can be directly applied, as we simply need to change the rejection step in the hit-and-run algorithm: in this case, we keep the replicates resulting in the selection of the same motifs using TF-MoDISco. More relevant association scores could be devised for such motifs, but our procedure is nonetheless valid.

For other features types, some practical problems may arise. First, the hit-and-run sampler requires the selection method to be stable, that is, running the selection method twice on the same input must lead to the selection of the same set of features. This property is required to guarantee the theoretical convergence of the algorithm, but may not be necessary in practice. Second, some attention may be required to avoid the computational cost to become prohibitive, in particular depending on the regularity properties of the selection event, that may lead to a higher rejection probability or to a higher required number of replicates. Granted that these technical challenges can be addressed, we are confident that extending SEISM to more general networks and corresponding features will benefit both the fields currently using these networks — such as regulatory genomics — and genome wide association studies.

• Accelerating the conditional inference procedure

The main obstacle to using SEISM on a real dataset is the time required for the inference step. To tackle this issue, we identify two promising directions:

- We can try to leverage the formulation of the optimization problem as a difference of convex functions, as discussed in Chapter 3 Subsection 3.3.1. While it does not seem relevant to use it to select the motif associated with a $\boldsymbol{y}$, we can try to make use of condition (3.14) to check if a given $\boldsymbol{y}'$ admits a motif as an argmax for the score. Indeed, during the rejection step of the hit-and-run sampler, each replicate currently goes through the whole selection procedure, to check whether it leads to the selection of the same set of motifs than the initial $\boldsymbol{y}$. But we can try to derive from this condition a quicker check method.

- As discussed in Chapter 2 Subsection 2.4.4, Yamada et al. (2018) introduce a new parametrization for the selection event. Instead of parametrizing the test with the outcome $\boldsymbol{y}$, resulting in quadratic constraints, the authors directly rely on the measure of independence — the score in our framework. They show that this new parametrization leads to linear constraints, and thus can derive an analytical expression for the truncation bounds of the null distribution. However, a few obstacles prevent us from applying the same trick. First, our selection event, even with the meshes, cannot be described with a finite number of constraints. Consequently, there are still theoretical challenges to overcome. Moreover, their method relies on a Gaussian assumption on the distribution of the scores. While this assumption is asymptotically true for the HSIC score, it does not necessarily hold with different scores. And the authors show that for a small number of samples, their procedure is not calibrated due to the violation of this assumption. As it is, our method is more general (and applicable to small datasets), but this direction is nevertheless a very interesting direction, since it could lead to substantial cuts in the computational costs.

- Third, our selection event can be written as an intersection of quadratic constraints. As discussed in Chapter 2, theoretical analytical bounds for the null distribution exist, but computing them is not tractable. It could then be interesting to try to obtain an approximation of these bounds in a reasonable amount of time, and to check whether this leads to a valid inference procedure or not.

• From SEISM to genome wide association studies: remaining challenges

Finally, this thesis attempted to create a bridge between neural networks and genome wide association studies. But many challenges remain before applying SEISM in a genome wide association study. First of all, it should be able to run on a large number of samples. While stochastic gradient descent can be implemented, it is still necessary to confirm that the hit-and-run algorithm works with an unstable selection method. Then, the activation functions we proposed in this thesis may not be relevant to detect a signal at the scale of a whole genome with millions or billions of base pairs. As of now, SEISM only gave results on small sequences, with a few hundred base pairs. In a GWAS context, we could use a CNN with long filters, resulting in selecting long sequence motifs, and thus sequences, with some flexibility, associated with the trait. Adding attention layers could help to select interactions between those sequences.

In conclusion, the development of SEISM marks, in our opinion, a significant step forward in going beyond explicability for machine learning, but its current application scope remains rather limited. However, lots of promising opportunities for improvement have been identified, and we look forward exploring them in future research.

Neural Networks beyond explainability: Selective inference for sequence motifs

Antoine Villié *antoine.villie@univ-lyon1.fr*
Université de Lyon, Université Lyon 1, CNRS, VetAgro Sup, Laboratoire de Biométrie et Biologie Evolutive, UMR5558, Villeurbanne, France

Philippe Veber *philippe.veber@univ-lyon1.fr*
Université de Lyon, Université Lyon 1, CNRS, VetAgro Sup, Laboratoire de Biométrie et Biologie Evolutive, UMR5558, Villeurbanne, France

Yohann De Castro *yohann.de-castro@ec-lyon.fr*
Université de Lyon, École Centrale de Lyon, CNRS UMR 5208, Institut Camille Jordan, France

Laurent Jacob *laurent.jacob@univ-lyon1.fr*
Université de Lyon, Université Lyon 1, CNRS, VetAgro Sup, Laboratoire de Biométrie et Biologie Evolutive, UMR5558, Villeurbanne, France

Abstract

Over the past decade, neural networks have been successful at making predictions from biological sequences, especially in the context of regulatory genomics. As in other fields of deep learning, tools have been devised to extract features such as sequence motifs that can explain the predictions made by a trained network. Here we intend to go beyond explainable machine learning and introduce SEISM, a selective inference procedure to test the association between these extracted features and the predicted phenotype. In particular, we discuss how training a one-layer convolutional network is formally equivalent to selecting motifs maximizing some association score. We adapt existing sampling-based selective inference procedures by quantizing this selection over an infinite set to a large but finite grid. Finally, we show that sampling under a specific choice of parameters is sufficient to characterize the composite null hypothesis typically used for selective inference—a result that goes well beyond our particular framework. We illustrate the behavior of our method in terms of calibration, power and speed and discuss its power/speed trade-off with a simpler data-split strategy. SEISM paves the way to an easier analysis of neural networks used in regulatory genomics, and to more powerful methods for genome wide association studies (GWAS).

1 Introduction

In the recent years, neural networks have been successfully used for making predictions from biological sequences. In particular, they have brought significant improvements in regulatory genomics, *e.g.* to predict cell-type specific transcription factor binding, gene expression, chromatin accessibility or histone modifications from a DNA sequence (Zhou & Troyanskaya, 2015; Kelley et al., 2018; Avsec et al., 2021a;b). These tasks are expected to be a good proxy for predicting the functional effect of non-coding variants, and help us in turn make better sense of the observed human genetic variation and its effect on various phenotypical traits including diseases. Most successful models have used convolutional neural networks (CNNs, LeCun & Bengio, 1998) and more recent approaches have explored self-attention mechanisms (Vaswani et al., 2017). These models have been trained from experimental data obtained from ChIP-seq, ATAC-seq, DNase-seq, or CAGE assays, that provide examples where both the DNA sequence and the outcome of interest are known.

A commonly outlined limitation of neural networks is their lack of explainability or black box aspect, *i.e.*, the contrast between their excellent prediction accuracy and the possibility to explain these in intuitive or

mechanistic terms (Ras et al., 2022; Molnar, 2022). Elementary one-layer CNNs don't face this issue, as their trained filters have a straightforward interpretation as position weight matrices (PWMs, Harr et al., 1983; Schneider & Stephens, 1990), a historical and basic element of regulatory genomics. Nonetheless, these simple models are notoriously too simple to capture the complexity of the regulatory code which requires to account not only for individual motif presence but for their long range sequence context and mutual interactions (Avsec et al., 2021b). Multi-layer CNNs and self-attention mechanisms model this additional complexity but are less straightforward to interpret. Tools inspired from the explainable deep learning literature have been adapted to extract features beyond PWMs and one-layer CNNs to explain the predicted regulatory behavior (Novakovsky et al., 2022). It is therefore often possible to explain the predictions of a trained neural network for biological sequences, either directly through estimates of its parameters or through features extracted post hoc.

Unfortunately, finding features somewhat associated to an outcome is often not enough, as an observed non-zero association can be spurious. In experimental science, it is actually common to quantify the significance of this association, *e.g.*, by testing the hypothesis that it is zero. Genome wide association studies (GWAS, Visscher et al., 2017) for example find genetic variants correlated with a trait by building a linear model explaining this trait by each variant and testing the hypothesis that the weight is zero. Statistical significance has its own limitations (Wasserstein & Lazar, 2016), but often provides an intuitive scale for identifying relevant features. Quantifying the significance of associations between interpretable features and predicted outcome is equally important in the context of neural networks but has received little attention to our knowledge.

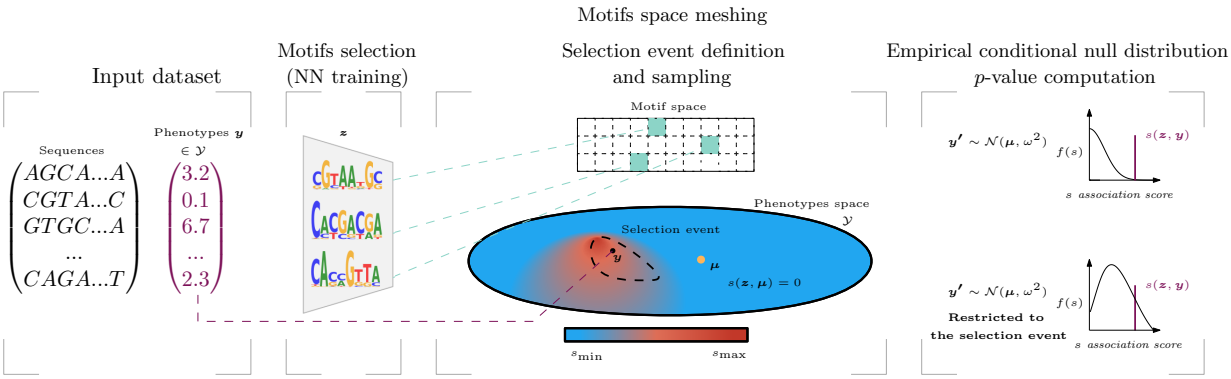


Figure 1: Overview of our SEISM procedure. (a) The input is a set of sequences and corresponding phenotypes in some space $\mathcal{Y}$ (b) It trains a convolutional neural networks to predict a phenotype from sequences, which leads to the selection of sequence motifs. (c) Then SEISM partitions the space of motifs to quantize the selection. The selection event is the set of phenotype vectors that would lead to selecting an element in the same mesh. (d) Using a sampling strategy, SEISM builds a null distribution for the test statistic, conditional to the selection event. The p-values associated with a selected motif is the quantile of its score under this distribution.

Here, we set out to go beyond explainable machine learning by introducing SElective Inference for Sequence Motifs (SEISM), depicted in Figure 1, a valid statistical inference procedure for these features. In order to do so, we cast commonly used CNNs in a feature selection framework, and show that it achieves similar selection performances as existing bioinformatics algorithms on *de novo* motifs discovery tasks. This selection needs to be accounted for when testing the association of the features with the predicted trait. This problem has been discussed and addressed in the growing literature on selective inference over the past few years (Taylor & Tibshirani, 2015; Reid et al., 2018; Slim et al., 2019), but existing methods only apply to a selection from a finite set. We work around this issue by quantizing our selection to a very large but finite space, making it amenable to existing sampling strategies. We show that the resulting procedure is well calibrated and compare it to a data-split strategy on a variety of settings.

For the sake of simplicity, we choose to restrict this presentation to simple one-layer CNNs and sequence motifs. The procedure that we introduce, however, is by no means limited to this framework, and could be applied to any of the more expressive features proposed in the explainable machine learning literature.

Our contributions are as follows:

- We formally cast one-layer CNNs into a motif discovery tool, reaching similar performances as *de-novo* motifs discovery tools from the bio-informatics literature (Section 3).
- We define a framework to perform post-selection inference dealing with selection over a continuous set of features, and thereby we make interpretable features amenable to inference (Section 4).
- The standard Gaussian framework for selective inference typically allows several means to be under the same selective null hypothesis, and require the variance to be known, both of which make more difficult the sampling under this null. We provide invariance results suggesting a practical procedure that works around these issues (Section 4.6). To our knowledge, they were a blind spot in sampling-based post-selection inference approaches beyond our specific context.
- We provide a PyTorch implementation of SEISM at:
  `https://gitlab.in2p3.fr/antoine.villie1/seism`.

2 A short overview of our SEISM procedure

SEISM aims to detect sequence motifs associated with a biological outcome, and to test the statistical significance of this association. To this end, it performs different steps which we will briefly describe here, in order to give the reader an overview of the procedure. They are summarized in Algorithm 1, and more details will be given in the following sections.

(i) SEISM takes as input biological sequences $\boldsymbol{X}$ associated with a phenotype $\boldsymbol{y}$. The user must also specify the number of motifs to find, as well as a parameter controlling the meshing of the motif space, that is the precision with which the found motifs will be tested.

(ii) The motif selection step corresponds to the maximisation of a so-called association score $s(\cdot, \cdot)$, which depends on the phenotype and on the motifs $\boldsymbol{z}$ through their activation patterns in the biological sequences $\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$. This step is formally equivalent to training a one hidden layer CNN. We implement a greedy procedure, optimizing each new filter over the residuals of the previously entered ones, using a gradient descent method initialized at the k-mer with the best score. To that end, we enumerates the k-mers contained in $\boldsymbol{X}$ using the DSK software (Rizk et al., 2013) and compute their scores $s(\cdot, \cdot)$.

(iii) SEISM splits the set of sequence motifs into meshes according to the input parameter. This step leads to the definition of a set of null hypotheses and of a selection event E, *i.e.* the set of outcomes $\boldsymbol{y}'$ that would have led to the selection of motifs within the same meshes as the ones selected in (ii), namely the sequence of meshes $(M_{i_1}, \ldots, M_{i_q})$. Formally, the selection event reads

$$E := \left\{ \boldsymbol{y}' \in \mathcal{Y} \,:\, \forall j \in [q]\,,\ \operatorname*{arg\,max}_{\boldsymbol{z} \in \mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}') \in M_{i_j} \right\}, \tag{1}$$

for some projection matrix $\boldsymbol{P}_j$, to be defined later.

(iv) It approximates the conditional null distribution of the test statistics by sampling biological outcomes $\boldsymbol{y}'$ under the null, conditionally to the selection event. This sampling is performed using a hit-and-run strategy (according to Algorithm 2), by building a discrete time Markov chain on E whose distribution converges to the uniform one.

(v) SEISM finally computes the p-values for the null hypotheses defined in (iii), associated with the selected motifs in ii, using the empirical distribution of the test statistics, and returns the motifs with their association p-values. Given these p-values, one can adjust the number of selected motifs discarding the ones with non-significant p-values. This multiple-testing issue has not been investigated in this paper, but the practitioner can use for instance a Bonferroni bound to select the number of motifs.

Algorithm 1 SEISM algorithm (general formulation)

 # Description: `SEISM` *selects a set of sequence motifs* $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$ *based on an association score* $s(\cdot, \cdot)$*, and evaluate their p-values based on a partition* $\mathcal{Z} = \bigsqcup M_i$*.*

Inputs: *Response* $\boldsymbol{y} \in \mathcal{Y} \subseteq \mathbb{R}^n$*, sequence samples* $\boldsymbol{X}$*, feature function* $\boldsymbol{z} \in \mathcal{Z} \mapsto \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}} \in \mathbb{R}^n$*, association score* $s : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$*, number of selected motifs* $q \geq 1$*, meshes* $\mathcal{Z} = \bigsqcup_{i=1} M_i$*, sampling algorithm* $\mathcal{HR}$*.*

Result: $((p_1, \boldsymbol{z}_1), \ldots, (p_q, \boldsymbol{z}_q))$, sequence of p-values and sequence motifs.

 # Selection step: *Selection of the sequence motifs* $(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q)$ *and the sequence of meshes* $(M_{i_1}, \ldots, M_{i_q})$*.*

for $j = 1, \ldots, q$ **do**
  $\boldsymbol{z}_j \leftarrow \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y})$; `//` $\boldsymbol{P}_k$ `orthogonal projection onto` $\underset{\ell<j}{\mathrm{Span}} \left\{\boldsymbol{\varphi}^{\boldsymbol{z}_\ell,\boldsymbol{X}}\right\}^\perp$
  $i_j \leftarrow i$ s.t. $\boldsymbol{z}_j \in M_i$; `//` `the mesh` M_{i_j} `is selected`
end

 # Inference step: `SEISM` *provides a p-value* p_k *on the statistical influence of the selected sequence motifs* $\boldsymbol{z}_k$ *conditional on the selection event* (1) *of observations* $\boldsymbol{y}'$ *that would have led to same selection of the sequence of meshes* $(M_{i_1}, \ldots, M_{i_q})$*.*

$\boldsymbol{y}'^{(1)}, ..., \boldsymbol{y}'^{(N)} \leftarrow \mathcal{HR}(y, (M_{i_1}, \ldots, M_{i_q}))$; `// Sampling outcomes under the selected null`

for $j = 1, \ldots, q$ **do**
  $\tilde{F}_j(\,\cdot\,; \boldsymbol{y}'^{(1)}, ..., \boldsymbol{y}'^{(N)}) \leftarrow$ empirical cumulative distribution function of $s(\boldsymbol{r}M_{i_j}, \boldsymbol{\Pi_j} \boldsymbol{y}')$ under the selected null ; `//` $\boldsymbol{r}M_{i_j}$ `is a motif representing` M_{i_j} `and` $\boldsymbol{\Pi_j}$ `the orthogonal projection onto` $\underset{\ell \neq j}{\mathrm{Span}} \left\{\boldsymbol{\varphi}^{\boldsymbol{z}_\ell,\boldsymbol{X}}\right\}^\perp$
  $p_j \leftarrow \tilde{F}_j(s(\boldsymbol{r}M_{i_j})\,; \boldsymbol{y}'^{(1)}, ..., \boldsymbol{y}'^{(N)})$; `//` `output the` j^{th} p`-value`
end

3 One hidden layer CNNs select sequence motifs maximizing an association score

One-layer CNNs have been at the core of the rising popularity of deep learning over the past decade, by enabling major improvements in computer vision tasks (Krizhevsky et al., 2012). Although they are formally a specialized fully connected feedforward networks with additional constraints on the weights, CNNs are equivalent to, and more often thought of as, a set of *convolutions* of the vectorial input with some smaller vectors referred to as filters. When applying the network, dot products are taken between each of them and successive windows of the vectorial input followed by some non-linear operation, producing an activation profile for each filter. In one-layer networks, these activations are pooled across the windows into a single scalar for each filter and these scalars are combined—typically through a linear or regular fully connected network—to provide a prediction for the input. Because convolution filters are homogeneous to the input, they easily lend themselves to interpretation: as small image patches for image inputs, and as sequence motifs for appropriately encoded biological sequence inputs. Accordingly, activation profiles reflect how much each piece of the input is similar to the filter—in the sense of the dot product—and applying a one-layer CNN amounts to applying a predictive function to a modified representation of the original data by these similarity profiles. Because convolution filters are jointly optimized with the parameterization of the predictive function, CNNs are often described as a strategy to jointly learn a data representation and a function acting on this representation, both being optimized for a prediction objective. In computer vision, the optimized filters of the first layer typically learn to detect edges with different orientations. In biological sequences, they learn short sequences whose presence anywhere in the input is predictive of the output phenotype used for training.

Unlike input sequences that are formed by a discrete succession of letters in some alphabet, trained filters are continuous and therefore account for possible variation in the predictive short sequence, *e.g.*, a T mostly followed by a C but sometimes an A or a G and so on (Figure 2). These probabilistic objects have also been

$$\begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{pmatrix} 0.11 & 0.27 & 0.01 & 0.99 & 0.67 & 0.77 & 0.00 & 0.00 \\ 0.00 & 0.48 & 0.00 & 0.00 & 0.25 & 0.01 & 0.60 & 0.00 \\ 0.01 & 0.17 & 0.13 & 0.00 & 0.05 & 0.21 & 0.40 & 0.00 \\ 0.88 & 0.08 & 0.86 & 0.01 & 0.03 & 0.01 & 0.00 & 1.00 \end{pmatrix}$$

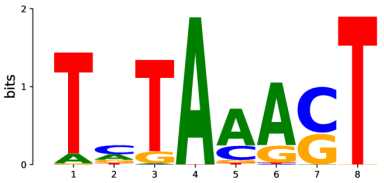


Figure 2: A motif represented by its position weight matrix and corresponding sequence logo. The total height of the letters indicates the information content of the position (in bits), closely related to the Shannon entropy.

used for a long time in the bioinformatics literature and referred to as position weight matrices (PWMs). Inferring PWMs either according to their frequency in a set of sequences (Bailey et al., 2006) or their discriminating power between two sets (Bailey, 2021) has been a major theme over the past thirty years. Here we formalize the training a one-layer CNN as equivalent to the selection of a set of sequence motifs that are optimal for some association score. This formalization will be instrumental in the definition of our hypothesis testing procedure in Section 4.

Notations Let $\boldsymbol{X}$ represent a data set of n one-hot encoded sequence samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, in a set $\mathcal{X}$ of biological sequences assumed to be over an alphabet $\mathcal{A}$—for DNA sequences, $\mathcal{A} = \{A, C, T, G\}$. One-hot encoding maps each letter in $\mathcal{A}$ to a vector in $\{0,1\}^{|\mathcal{A}|}$, with all-zero entries except for a single 1 at the coordinate corresponding to the order of the letter in $\mathcal{A}$—for DNA sequences, A is encoded as $(1,0,0,0)$. Every $\boldsymbol{x}_i$ is therefore encoded as a matrix in $\{0,1\}^{|\mathcal{A}|\times|\boldsymbol{x}_i|}$—although in practice, encoded sequences are often padded with dummy columns to have the same lengths. We denote $y_i \in \mathcal{Y}$ the measurement of a biological property associated with sequence $\boldsymbol{x}_i$, and $\boldsymbol{y} \in \mathcal{Y}^n$ the corresponding vector of outcomes. We consider one-layer CNNs with a Gaussian non-linearity with scale ω, a max global pooling and a linear prediction function. These CNNs parameterize a function $f : \mathcal{X} \to \mathcal{Y}$ by q filters of length k, namely $\boldsymbol{Z} := \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_q\} \in \mathcal{Z}^q$, where $\mathcal{Z}$ is a subset of $\mathbb{R}^{|\mathcal{A}|\times k}$, given by the simplex in this paper:

$$\mathcal{Z} = \left\{ \boldsymbol{z} \in \mathbb{R}_+^{|\mathcal{A}|\times k} \ : \ \forall j \in [k], \ \sum_{i=1}^{|\mathcal{A}|} z_{i,j} = 1 \right\}, \tag{2}$$

and q weights $\beta \in \mathbb{R}^q$.

More precisely, we define $f(\boldsymbol{x}_i) := \left(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\right)_i$, with $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \in \mathbb{R}^{n\times q}$ defined as $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} = \boldsymbol{C}_n\tilde{\boldsymbol{\varphi}}^{\boldsymbol{Z},\boldsymbol{X}}$, where $\boldsymbol{C}_n = I_n - n^{-1}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering operator, I_n the identity matrix, $\mathbf{1}_n$ the all-one vector in $\mathbb{R}^n$, and

$$\tilde{\varphi}_{i,j}^{\boldsymbol{Z},\boldsymbol{X}} := \max_{\boldsymbol{u}\in[\boldsymbol{x}_i]_\ell} \left\{ \exp\left(-\frac{||\boldsymbol{z}_j - \boldsymbol{u}||^2}{2\omega^2} \right) \right\}, \tag{3}$$

where $[\boldsymbol{x}_i]_\ell$ denotes the set of ℓ consecutive entries of the vector $\boldsymbol{x}_i$ (and of its reverse-complement counterpart), and ω is a bandwidth hyperparameter whose impact and tuning is studied in Appendix A. This model differs with a typical CNN in two ways. First, it uses a Gaussian activation function instead of an exponential one; second the use of the centering operator that sets the average of the activation to zero. These adjustments were made to improve the SEISM algorithm's selection performances.

3.1 From empirical risk minimization to association scores

The function f is learned in a classical penalized empirical risk minimization framework, using the data $\{\boldsymbol{X}, \boldsymbol{y}\}$:

$$\min_{(\boldsymbol{Z},\boldsymbol{\beta})\in(\mathcal{Z}\times\mathbb{R}^q)} n^{-1}\left\|\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}\boldsymbol{\beta}\right\|^2 + \lambda||\boldsymbol{\beta}||^2, \tag{4}$$

for some $\lambda > 0$. Equation (4) formalizes the idea that learning a one-layer CNN on one-hot encoded sequences amounts to learning a data-representation $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$ of the sequences parameterized by a set $\boldsymbol{Z}$ of

filters—corresponding to PWMs—and a linear function with weights $\boldsymbol{\beta}$ acting on this representation. Noting that there exists a unique explicit optimal $\boldsymbol{\beta}$ for Eq. (4), it follows immediately that:

$$\arg\min_{\boldsymbol{Z}} \left\{ \min_{\boldsymbol{\beta}} \left\{ n^{-1} ||\boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \boldsymbol{\beta}||^2 + \lambda ||\boldsymbol{\beta}||^2 \right\} \right\} = \arg\max_{\boldsymbol{Z}} \left\{ s_\lambda^{\text{ridge}}(\boldsymbol{Z}, \boldsymbol{y}) \right\}, \tag{5}$$

where s^{ridge} defines a particular quadratic association score between an outcome $\boldsymbol{y}$ and a set of filters $\boldsymbol{Z}$:

$$s_\lambda^{\text{ridge}}(\boldsymbol{Z}, \boldsymbol{y}) := \boldsymbol{y}^T \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \left[(\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}})^T \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} + \lambda n \boldsymbol{I}_q \right]^{-1} (\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}})^T \boldsymbol{y}. \tag{6}$$

It formalizes the training of a CNN as the selection of a set of filters whose association with $\boldsymbol{y}$ in the sense of s_λ^{ridge} is maximal. Of note, one has

$$\lim_{\lambda \to \infty} \lambda n \times s_\lambda^{\text{ridge}}(\boldsymbol{Z}, \boldsymbol{y}) = \boldsymbol{y}^T \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} (\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}})^T \boldsymbol{y} =: s^{\text{HSIC}}(\boldsymbol{Z}, \boldsymbol{y}),$$

so for large values of the regularization hyperparameter, selecting filters by learning a CNN is equivalent to selecting filters with the classical HSIC score (Song et al., 2012), because $\boldsymbol{\varphi}$ already includes a centering operator. In addition to connecting s^{ridge} with s^{HSIC}, we observed that the centering in the definition of $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$ led to the selection of better sequence motifs in our experiments. Observe that the centering matrix is an orthogonal projection matrix onto $\mathcal{E} := \text{Range}(\boldsymbol{C}_n)$, the orthogonal of the vector line generated by the vector $\mathbf{1}$, and it holds

$$\left\| \boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \boldsymbol{\beta} \right\|_n^2 = \left\| \boldsymbol{C}_n \boldsymbol{y} - \boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}} \boldsymbol{\beta} \right\|_n^2 + \left\| \boldsymbol{y} - \boldsymbol{C}_n \boldsymbol{y} \right\|_n^2. \tag{7}$$

The solution of (4) is unchanged if $\boldsymbol{y}$ is replaced by $\boldsymbol{C}_n \boldsymbol{y}$, and so we can assume that $\boldsymbol{y} \in \mathcal{E}$ without any generality loss. Furthermore, this shows that we can work with skewed data in a classification context, since imbalanced classes will have no effect on the result.

3.2 Greedy optimization

It is common to solve (4) by stochastic gradient descent (SGD) jointly over the q filters. More generally, this approach for training a neural network with a single, large hidden layer is known to find a global optimizer at the large q limit under some assumptions (Soltanolkotabi et al., 2019). Our objective here is slightly different: we do not necessarily aim at approximating a continuous measure with a large number of particules, but we aim at selecting a small number of particules lending themselves to a biological interpretation. Furthermore, the number of relevant motifs on a given dataset is generally unknown. In this context, it is known that jointly optimizing the convolution filters leads to irrelevant PWMs, with some actual motif split across several filters and other duplicated (Koo & Eddy, 2019). A possible strategy is to forego filter-level interpretation, train an overparameterized network—with a much larger q than the expected number of motifs—and use attribution methods to extract relevant motifs or other interpretable features from the trained network (Shrikumar et al., 2018). Here we adopt a different strategy using a forward stepwise procedure, where we iteratively optimize each of the convolution filters over the residual error left by the previous ones.

More precisely at each of the q steps, we select $\boldsymbol{z}_j$ such that:

$$\boldsymbol{z}_j = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} s^{\text{ridge}}(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}), \tag{8}$$

where $\boldsymbol{P}_j$ is the projection operator onto the orthogonal of the subspace $\underset{\ell < j}{\text{Span}} \left\{ \mathbf{1}, \boldsymbol{\varphi}^{\boldsymbol{z}_\ell,\boldsymbol{X}} \right\}$, see line 2 of Algorithm 1. This is how $\boldsymbol{z}_j$ is optimized over the residuals of the previous filters. The vector $\mathbf{1}$ enforces that we project on a subspace of $\mathcal{E}$, in particular $\boldsymbol{P}_1 = \boldsymbol{C}_n$. Without this projection, iterating (8) would return the same $\boldsymbol{z}$. Of note, joint optimization procedures of the q filters don't face this issue, and forward selection procedures over finite sets of features work around the problem by iteratively removing the selected elements from the set over which selection if performed (Slim et al., 2019). This sequential strategy combined with the testing procedure introduces in Section 4 provides a data-driven mean to choose the number q of relevant motifs.

In practice, we solve (8) with a standard gradient descent algorithm, initialized at the k-mer with the best association score. The k-mer list is obtained using the DSK software (Rizk et al., 2013). We work on a less

constrained set than $\mathcal{Z}$ (2) and don't enforce the positivity constraint during optimization. We project the optimized motifs onto the full $\mathcal{Z}$ at the end of the process. Our procedure also requires to choose a motif length k. We proceed adaptively by choosing the length leading to the highest score, within a user-specified range.

With the one-layer CNNs training formally cast as the successive selection of q sequence motifs optimizing an association score, we now turn to the problem of testing the significance of these associations. Of note, what follows is only based on the definition of an association score and could be applied to perform inference on other features coming from the training step of any algorithm, as long as one can define an association score between the feature and the outcome.

4 Post-selection testing of the association between the outcome and trained convolution filters

We now turn to the problem of testing the association between the selected motifs $\boldsymbol{z}$ and the trait $\boldsymbol{y}$. In order to do so, we need to solve three interrelated problems. First, the motifs were specifically selected for their association with the trait, which leads to the well known post-selection inference problem. Any inference procedure that disregards that the hypothesis was constructed using the same data used for testing is likely invalid and produces deflated p-values. Second, we deal with a continuous selection event, because (8) is performed over a continuous set $\mathcal{Z}$. By contrast, existing solutions for post-selection inference address selections over finite sets. Third, the null hypothesis commonly used for similar post-selection inference problems is composite, *i.e.*, it corresponds to several values of the parameters. Existing methods work around this issue by fixing theses parameters to arbitrary values, thereby limiting the scope under which they are calibrated. Here we present our solutions to these three problems.

Consider the Gaussian model:

$$\boldsymbol{y} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon} \tag{9}$$

where $\boldsymbol{\mu} \in \mathcal{E}$ is the target deterministic signal, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{C}_n)$ the standard Gaussian distribution on $\mathcal{E}$.

4.1 Selective null hypothesis

We follow Yamada et al. (2018) and test the association of a motif $\boldsymbol{z}$ through the following null hypothesis:

$$\mathbb{H}_0 : \text{“}s(\boldsymbol{z}, \boldsymbol{\mu}) = 0\text{”}, \tag{10}$$

for some association score s. For a $\boldsymbol{z}$ chosen independently of the data, $\mathbb{H}_0$ could be tested by sampling $\boldsymbol{y}'$ under the corresponding distribution, and using the quantile of the $s(\boldsymbol{z}, \boldsymbol{y}')$ scores corresponding to $s(\boldsymbol{z}, \boldsymbol{y})$ as a p-value—*i.e.*, the probability when sampling under $\mathbb{H}_0$ to observe a score as extreme as $s(\boldsymbol{z}, \boldsymbol{y})$. In our case, however, the motifs $\boldsymbol{z}$ in the trained convolution filters were specifically selected for their strong association with $\boldsymbol{y}$, and this procedure would not produce calibrated p-values. This problem is known as post-selection inference, and has been discussed and addressed in a growing literature over the past few years. Although data-split strategies — which split the data into two parts, and then perform the selection and the inference on the two different parts — lead to valid procedures (Wasserman & Roeder, 2009), they necessarily result in a reduction of the sample size, unsatisfying when the original sample size is limited. Alternatively, selective inference frameworks were developed in the recent years to address these issues. We refer to (Hastie et al., 2015, Chapter 6) and references therein for a general presentation. Taylor et al. (2014) and Lee et al. (2016) address scenarios where the selection event, *i.e.* the set of data outputs that would result in the selection of the same set of features, is polyhedral—determined by the finite intersection of linear constraints. Reid & Tibshirani (2013), and later Reid et al. (2015) extend this selection to clusters or groups of features, still in the linear framework. Yamada et al. (2018) extended post-selection inference to the non-linear framework, by proposing a kernel-based approach, where the selection is performed through the HSIC criterion. Slim et al. (2019) generalize this work, by allowing the selection to be carried out with a wider range of tools, making use of quadratic association scores.

To our knowledge, post-selection inference literature only addresses the problem of selecting features from a discrete collection and does not provide a solution for selections from a continuous set like our $\mathcal{Z}$. Hence,

testing (10) directly is not feasible and we resort to the quantization of the motif space to address this problem.

In addition to that, we push the analysis of the statistical model further, in order to be able to apply it with weaker assumptions on the data distribution.

4.2 Dealing with selection events over a continuous set of features

Formally, our selection event $E_{\text{cont.}}$ is the set of outcomes $\boldsymbol{y}'$ that would have led to the selection of the same set of motifs $\boldsymbol{Z} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_q\}$ than the one selected using $\boldsymbol{y}$ from the real dataset, when applying the same selection procedure:

$$E_{\text{cont.}} := \left\{ \boldsymbol{y}' \in \mathcal{E} \ : \ \forall j \in \{1, \dots, q\} \ \ \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}') = \boldsymbol{z}_j \right\}, \tag{11}$$

where $\boldsymbol{P}_j$ is the orthogonal projection onto $\underset{\ell < j}{\text{Span}} \left\{ \mathbf{1}, \boldsymbol{\varphi}^{\boldsymbol{z}_\ell, \boldsymbol{X}} \right\}^\perp$.

A simple rejection approach to sample from the null (10) conditioned to $E_{\text{cont.}}$ would be to sample $\boldsymbol{y}$ in $\mathcal{E}$ under (9, 10) and only retain those in $E_{\text{cont.}}$. Unfortunately, $E_{\text{cont.}}$ belongs to a strictly lower-dimensional vector space of $\mathbb{R}^n$ and is therefore a null set for the Lebesgue measure on $\mathbb{R}^n$. For s^{HSIC} and s^{ridge}, and noting that a maximum is also a critical point, we indeed obtain:

$$\boldsymbol{y}' \in E_{\text{cont.}} \implies \forall j \in \{1, \dots, q\} \ \ \boldsymbol{P}_j \boldsymbol{y}' \in \text{Span} \left\{ \nabla_{\boldsymbol{z}} \boldsymbol{\varphi}^{\boldsymbol{z}_j, \boldsymbol{X}} \right\}^\perp .$$

For $q = 1$ and assuming that the different directions of the gradient are independent, this spans is a vector subspace with dimension $n - 4 \times k$. We empirically observed that sampling from this subspace produced a non-zero proportion of $\boldsymbol{y}'$ in $E_{\text{cont.}}$. Nonetheless, choosing a sampling distribution that leads to the correct conditional distribution is not straightforward—and may not even be possible—as discussed in Supplementary Material B. Moreover, relying on conditional probability with respect to a null set is not well defined and may lead to the Borel-Kolmogorov paradox (Bungert & Wacker, 2022), which further complicates its use.

We choose to circumvent this issue using a partition of the space $\mathcal{Z}$ of motifs spaces, over which our selection (8) operates, into a very large but finite set of meshes: $\mathcal{Z} = \bigsqcup M_i$. As depicted in Figure 3, we consider a regular partition of each coordinates into m bins:

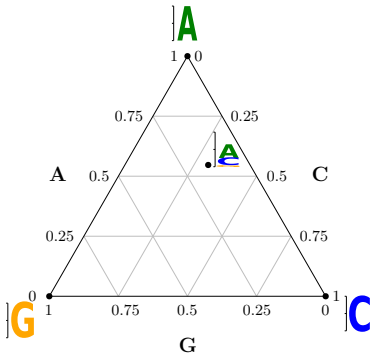


Figure 3: Discretization of the 3-letters alphabet simplex $\{A, C, G\}$, with a binning parameter for the meshes $m = 4$.

Based on this partition into meshes, we define a quantized selection event E as follows. First, given an outcome $\boldsymbol{y}$ we define the sequence of the q selected meshes $(M_{i_1}, \dots, M_{i_q})$ as

$$\forall j \in \{1, \dots, q\} \ , \ \ \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}) \in M_{i_j} \ ,$$

Second, the selection event is given by:

$$E(i_1, \dots, i_q) := \left\{ \boldsymbol{y}' \in \mathcal{Y} \,:\, \forall j \in \{1, \dots, q\} \;,\; \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_j \boldsymbol{y}') \in M_{i_j} \right\}, \tag{12}$$

the set of outcomes $\boldsymbol{y}'$ that would have led to the selection of motifs within the same meshes as the selected ones $(M_{i_1}, \dots, M_{i_q})$.

We now show how quantization (12) of the selection problem make enables the definition of a valid inference procedure. We start with the simplest case where we select a single motif ($q = 1$).

4.3 Test with only one motif $q = 1$, μ and σ fixed

In this section, considering the motif $\boldsymbol{z}_1$ was chosen by the SEISM selection procedure, selection event (12) boils down to:

$$E(i_1) := \left\{ \boldsymbol{y}' \in \mathcal{Y} \,:\, \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{y}') \in M_{i_1} \right\} \tag{13}$$

We use this simplified case to introduce our null hypotheses and test statistics attached to this selection event, and consider two options:

- A first option consists in representing the mesh M_{i_1} by its center $\boldsymbol{c}_1$. Then the corresponding null hypothesis is the following:

  $$\mathbb{H}'_{0,1} \,:\, \text{“}s(\boldsymbol{c}_1, \boldsymbol{\mu}) = 0\text{”}, \tag{14}$$

  It can be tested using statistic $V'_1 = s(\boldsymbol{c}_1, \boldsymbol{y})$.

- A second possibility is to represent M_{i_1} by the motif with the highest association score within. In this case, the null hypothesis becomes:

  $$\mathbb{H}''_{0,1} \,:\, \text{“}\forall \boldsymbol{z} \in M_{i_1} \;,\; s(\boldsymbol{z}, \boldsymbol{\mu}) = 0\text{”}, \tag{15}$$

  We test it using statistic $V''_1 = \max_{\boldsymbol{z} \in M_{i_1}} s(\boldsymbol{z}, \boldsymbol{y})$.

In both cases, we reject the null hypothesis if the test statistics are greater than a threshold, determined by their cumulative distributions under the nulls (14), (15) conditionally to $E(i_1) : \mathbb{F}'_{1,(i_1)}$ and $\mathbb{F}''_{1,(i_1)}$. In practice, there is no closed form for these conditional cumulative distributions, and we rely on an empirical version that we build using a hit-and-run sampler algorithm, as described in Section 4.4.

Hypotheses (14) and (15) lead to very similar results when the meshes are small enough, which is easily the case in practice. (14) gives us insights on one specific motif of the mesh — the center, but (15) tells us about whether there exists a motif within M_{i_1} associated with the phenotype. To illustrate the difference, let us consider a meshing with only one bin per coordinate, that is the meshing with only one mesh, containing all the motifs:

- Testing the center-based null hypothesis (14) boils down to testing the association of $\boldsymbol{\mu}$ with the motif $\boldsymbol{c}_1$ with the same probabilities for each letter of $\mathcal{A}$ at every position, and produces a p-value of 1, regardless of the data, since for any k-mer u, $\|\boldsymbol{c} - \boldsymbol{u}\|^2 = k \times (0.75^2 + 3 \times 0.25^2)$, which leads to $\forall \boldsymbol{X} \in \mathcal{X},\ \boldsymbol{\varphi}^{\boldsymbol{c}, \boldsymbol{X}} = \boldsymbol{0}$ according to the centering step, and to a zero score for any $\boldsymbol{y}' \in \mathcal{E}$.

- By contrast, one can obtain a strictly less than 1 p-value for (15), because different $\boldsymbol{y}' \in \mathcal{E}$ can lead to different scores, which means that there may exist a motif in $\mathcal{Z}$ associated with $\boldsymbol{y}$— but does not inform us on which motif it is.

Algorithm 2 Hypersphere Directions hit-and-run sampler

`/* Description: The Hypersphere Directions hit-and-run sampler creates a discrete-time Markov chain on an open and bounded region and is used to approximate a uniform distribution on the selection event` E. `*/`

Inputs: *Response* $\boldsymbol{y} \in E \subseteq \mathbb{R}^n$, B *and* R *the numbers of burn-in iterations and replicates.*

Result: $\boldsymbol{y}'^{(B+1)}, ..., \boldsymbol{y}'^{(B+R)} \in E \subseteq \mathbb{R}^n$ the replicates sampled under the conditional null distribution

$\tilde{\boldsymbol{y}}'^{(0)} \leftarrow \mathbb{L}(\boldsymbol{y})$; `/* ` $\mathbb{L}$ `is the cumulative distribution function of` $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$ `*/`

for $t = 1, \ldots, B + R$ **do**

  Sample uniformly $\boldsymbol{\theta}^{(t)}$ from $\{\boldsymbol{\theta} \in \mathbb{R}^n, \ \|\boldsymbol{\theta}\| = 1\}$;

  $a^{(t)} \leftarrow \max \left\{ \max_{\boldsymbol{\theta}_t^{(i)} > 0} -\frac{\tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}}; \max_{\boldsymbol{\theta}_t^{(i)} < 0} \frac{\mathbf{1} - \tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}} \right\}$;

  $b^{(t)} \leftarrow \max \left\{ \min_{\boldsymbol{\theta}_t^{(i)} < 0} -\frac{\tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}}; \min_{\boldsymbol{\theta}_t^{(i)} > 0} \frac{\mathbf{1} - \tilde{\boldsymbol{y}}'^{(t-1)}}{\boldsymbol{\theta_t}} \right\}$; `/* Sampling` $\lambda^{(t)}$ `from` $\left]a^{(t)}, b^{(t)}\right[$ `ensures that` $\tilde{\boldsymbol{y}}'^{(t-1)} + \lambda^{(t)} \boldsymbol{\theta^{(t)}} \in]0, 1[^n$ `*/`

  while $\boldsymbol{y}'^{(t)} \notin E$ **do**

    `/* This loop is parallelized on several cores until one of them discovers a replicates in the selection event. */`

    Sample uniformly $\lambda^{(t)}$ from $\left]a^{(t)}, b^{(t)}\right[$;

    $\tilde{\boldsymbol{y}}'^{(t)} \leftarrow \tilde{\boldsymbol{y}}'^{(t-1)} + \lambda^{(t)} \boldsymbol{\theta^{(t)}}$;

    $\boldsymbol{y}'^{(t)} \leftarrow \mathbb{L}^{-1}(\tilde{\boldsymbol{y}}'^{(t)})$;

  end

end

4.4 Sampling from the conditional null distribution with the Hit-and-Run algorithm

Even after reducing our selection to a finite set (Section 4.2), a rejection sampling strategy that would draw $\boldsymbol{y'}$ from either (9, 16) or (9, 17) and only retain those leading to the selection of the same mesh as $\boldsymbol{y}$ is not tractable as the rejection rate is empirically too low. Following Slim et al. (2019), we resort to a Hypersphere Direction strategy (Algorithm 2).

The hit-and-run algorithm produces uniform samples from an open and bounded acceptance region—corresponding, in our case, to the selection event. It starts from any point in the acceptance region, draws a random direction from this point and samples along this direction until it finds one elements that also falls in the acceptance region. It then follows the same procedure from this new starting point. The hit-and-run sampler therefore also relies on rejection but it does so along a single dimension rather than from $\mathbb{R}^n$. It explores the selection event step by step, starting from a point that belongs to this event, which guarantees a higher acceptance rate. To speed up the procedure, we parallelize the rejection step across several cores. Because each point sampled by the hit-and-run procedure depends on the previous one, it is impossible to parallelize the whole sampling process. By contrast, the rejection step used for computing a single replicate, once a sampling direction has been fixed, can be parallelized. We draw several distances to the initial point independently, optimizing new independent points, until one of them belongs to the selection event. This parallelization provides a significant time saving, as discussed in Section 5.3. Algorithm 2 produces uniform samples from an open and bounded acceptance region. The boundedness assumption does not hold in our case as the $\arg\max$ over $\mathcal{Z}$ of the score only depends on the direction of $\boldsymbol{y}$ and not on its norm. The openness requirement is ensured by the definition of the meshes. Following Slim et al. (2019) again, we use the reparameterization $\tilde{\boldsymbol{y}} = \mathbb{L}(\boldsymbol{y})$, where $\mathbb{L} : \mathbb{R}^n \rightarrow]0, 1[^n$ is defined as $\mathbb{L}(\boldsymbol{y})_i = \mathbb{L}_{\boldsymbol{\mu}, \sigma^2}(\boldsymbol{y}_i)$ for $i = 1, \ldots, n$ and $\mathbb{L}_{\boldsymbol{\mu}, \sigma^2}$ denotes the cumulative distribution function of $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$. Sampling uniform $\tilde{\boldsymbol{y}}$ from the open bounded space $]0, 1[^n$ indirectly provides normal samples from $\mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{C}_n)$.

Combining this sampling strategy with the quantization of the selection event introduced in Section 4.2 and the selective null hypotheses attached to this event introduced in Section 4.3 provides a selective inference procedure for one selected motif $\boldsymbol{z}_1$ ($q = 1$) and a null defined by a given pair $(\boldsymbol{\mu}, \sigma)$ of parameters. Our next two steps are to handle the selection of multiple motifs, and the general case where several $\boldsymbol{\mu}$ describe the same null hypothesis and σ is not specified.

4.5 Dealing with the selection of several motifs ($q > 1$)

We now consider that we selected $q > 1$ motifs with the SEISM procedure, leading to the general (12) selection event $E(i_1, \ldots, i_q)$. Generalizing our single-motif strategy of Section 4.3, we propose two options for defining null hypotheses (and test statistics) related to this selection event:

- The first one relies on the centers of the selected meshes:

$$\mathbb{H}_{0,j} \ : \ \text{“}s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_{\boldsymbol{j}}\boldsymbol{\mu}) = 0\text{”}, \tag{16}$$

  where $\boldsymbol{\Pi}'_{\boldsymbol{j}}$ is the orthogonal projector onto $\mathrm{Span}_{\ell \neq j} \left\{\boldsymbol{\varphi}^{\boldsymbol{c}_\ell, \boldsymbol{X}}\right\}^\perp$. In other words, it expresses that the center of the mesh M_{i_j} is associated with $\boldsymbol{\mu}$ after removing its component carried by the span of the centers of the meshes corresponding the the $q-1$ other motifs.

- And the second one takes advantages of the best motifs in each mesh:

$$\mathbb{H}_{0,j} \ : \ \text{“}\forall (\boldsymbol{z}^*_{i_\ell})_{\ell \neq j} \in (M_{i_\ell})_{\ell \neq j}\,, \quad \forall \boldsymbol{z} \in M_{i_j}\,,\ s(\boldsymbol{z}, \boldsymbol{\Pi''}\left(\left(\boldsymbol{z}^*_{\boldsymbol{i_\ell}}\right)_{\boldsymbol{\ell \neq j}}\right)\boldsymbol{\mu}) = 0\text{”}, \tag{17}$$

  with $\boldsymbol{\Pi''}\left(\left(\boldsymbol{z}^*_{\boldsymbol{i_\ell}}\right)_{\boldsymbol{\ell \neq j}}\right)$ being the projection onto $\mathrm{Span}_{\ell \neq j} \left\{\boldsymbol{\varphi}^{\boldsymbol{z}^*_{i_\ell}, \boldsymbol{X}}\right\}^\perp$.

Generalizing what we introduced for $q = 1$ (Section 4.3), we test those hypotheses using $V'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y})$ and $V''_j = \max_{\boldsymbol{z} \in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j \boldsymbol{y})$. To that end, we rely on their cumulative distributions under the nulls (16), (17) conditionally to $E(i_1, \ldots, i_q)$: respectively $\mathbb{F}'_{1,\ldots,q(i_1,\ldots,i_q)}$ and $\mathbb{F}''_{1,\ldots,q,(i_1,\ldots,i_q)}$, empirically approximated with Algorithm 2.

Following the work of Loftus & Taylor (2015) in the finite case, both versions of our null hypothesis are joint across the q motifs: each of them considers the association between the j-th selected motif and $\boldsymbol{\mu}$ after projecting onto the span of all others, not just the ones that were selected before — using $\boldsymbol{\Pi}'$ and $\boldsymbol{\Pi}''$. This is to be contrasted to our sequential selection process, which adjusts at each steps for the previously selected motifs using $\boldsymbol{P}$.

In order to give more insights on these null hypotheses, we derive the following proposition:

Proposition 4.1 (Description of the selective nulls)**.** *Let $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_q\}$ be q sequence motifs. Let $s(\cdot, \cdot)$ be a score such that "nullity implies orthogonality" (for instance s^{HSIC} or s^{ridge}):*

($\mathbf{A_1}$) **Nullity implies orthogonality:** *If $\{s(\boldsymbol{z}, \boldsymbol{y}) = 0\}$ then $\{\langle \boldsymbol{\varphi}^{\boldsymbol{z}, \boldsymbol{X}}, \boldsymbol{y} \rangle = 0\}$, for every $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$, and for some function $\boldsymbol{z} \to \boldsymbol{\varphi}^{\boldsymbol{z}, \boldsymbol{X}} \in \mathcal{E}$.*

Let $\boldsymbol{\mu} \in \mathcal{E}$ and decompose $\boldsymbol{\mu}$ as

$$\boldsymbol{\mu} = \sum_{j=1}^{q} \alpha_j \boldsymbol{\varphi}^{\boldsymbol{z}_j, \boldsymbol{X}} + \underline{\boldsymbol{\mu}} \tag{18}$$

with $\underline{\boldsymbol{\mu}} \in \mathcal{E}$ orthogonal to $\mathrm{Span}(\boldsymbol{\varphi}^{\boldsymbol{Z}, \boldsymbol{X}})$.

It holds that “$s(\boldsymbol{z}_j, \boldsymbol{\Pi}_{\boldsymbol{j}}\boldsymbol{\mu}) = 0$” is equivalent to “$\alpha_j = 0$” for some decomposition (18).

If $\mathrm{Rank}(\boldsymbol{\varphi}^{\boldsymbol{Z}, \boldsymbol{X}}) = q$ then the decomposition (18) is unique, and the greedy selection procedure described in Section 3 enforces this situation. We interpret this as follows: we look at a motif $\boldsymbol{z}_\ell$ and would like to test its significance; in view of property ($\mathbf{A_1}$), we can eliminate the effects that are captured by the other

motifs by using the orthogonal projection onto the orthogonal of $\mathrm{Span}(\boldsymbol{\varphi}^{\boldsymbol{z}_j,\boldsymbol{X}})$, given by $\boldsymbol{\Pi}_j$ (using $\boldsymbol{\Pi}_j = \boldsymbol{\Pi}'_j$ or $\boldsymbol{\Pi}_j = \boldsymbol{\Pi}''\left((\boldsymbol{z}^*_{i_\ell})_{\ell\neq j}\right)$), and consider $\boldsymbol{\Pi}_j\boldsymbol{y}$ to test the association "$s(\boldsymbol{z}_j, \boldsymbol{\Pi}_j\boldsymbol{\mu}) = 0$"; equivalent to testing "$\alpha_j = 0$" by the above proposition.

4.6 Sampling under selective multiple hypotheses with known σ

The sampling strategy described in Section 4.4 builds a conditional null distribution—therefore offering a selective inference procedure—for a given $\boldsymbol{\mu}$ and σ. In practice, σ is not known, and several values of $\boldsymbol{\mu}$ can describe the selective null hypotheses (16) or (17) for a given motif selection. Of note, this issue is not specific to our selective inference procedure. It will arise in any sampling-based post-selection inference strategy including data-split: even if the latter samples from a non-selective null hypothesis, it still needs concrete values for $\boldsymbol{\mu}$ and σ.

We leave aside the choice of σ for now, and describe how we can sample from any null distribution (16) or (17) using $\boldsymbol{\mu} = 0$ for a given σ. Our results holds for scores verifying the following assumption—this includes both s^{HSIC} and s^{ridge}:

- $(\mathbf{A_2})$ **Nullity implies translation-invariant:** If $s(\boldsymbol{z}, \boldsymbol{y}) = 0$ then $\forall \boldsymbol{y}' \in \mathcal{E}\,,\ s(\boldsymbol{z}, \boldsymbol{y}') = s(\boldsymbol{z}, \boldsymbol{y} + \boldsymbol{y}')$, for every $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$;

Under this assumption, the following proposition ensures that using the quantile of the empirical distribution of scores sampled under $\boldsymbol{\mu} = 0$ leads to a calibrated test procedure:

Proposition 4.2. *Let s be an association score such that $(\mathbf{A_2})$ holds. Let $V'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j\boldsymbol{y})$ and $V''_j = \max_{\boldsymbol{z}\in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j\boldsymbol{y})$, formed from $\boldsymbol{y}$ sampled from (9) with any mean $\boldsymbol{\mu}$ such that $s(\boldsymbol{z}', \boldsymbol{\mu}) = 0$, any known variance $\sigma > 0$, and such that $\boldsymbol{z}' = \arg\max_{\boldsymbol{z}\in\mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{y})$. The conditional null distributions $\mathbb{F}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{F}''_{j,(i_1,\dots,i_q)}$, with mean $\mathbf{0}$ and variance σ verify:*

$$\mathbb{F}'_{j,(i_1,\dots,i_q)}(V'_j) \sim \mathit{Unif}(0,1) \text{ and } \mathbb{F}''_{j,(i_1,\dots,i_q)}(V''_j) \sim \mathit{Unif}(0,1)$$

Proof. Assumption $(\mathbf{A_2})$ under the Gaussian model (9) implies the following property:

$$\begin{aligned} &\forall(\boldsymbol{z}, \boldsymbol{A}, \boldsymbol{y}) \in \mathcal{Z} \times \boldsymbol{A} \times \mathcal{E} \text{ such that } \boldsymbol{y} = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}, \\ &\text{“}s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{\mu}) = 0\text{”} \Longrightarrow \text{“}s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y}) = s(\boldsymbol{z}, \sigma\boldsymbol{A}\boldsymbol{\varepsilon})\text{”}, \end{aligned} \tag{19}$$

which implies that, for a composite null hypothesis of the form $\mathbb{H}_0$: "$s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{\mu}) = 0$", the distribution of $s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y})$ does not depend on the mean $\boldsymbol{\mu}$ that satisfies $\mathbb{H}_0$. Hence, even if the hypothesis $\mathbb{H}_0$ corresponds to a set of probability distributions of $\boldsymbol{y}$ that may depend on $\boldsymbol{\mu}$, the distribution of the statistic $s(\boldsymbol{z}, \boldsymbol{A}\boldsymbol{y})$ does not depend on $\boldsymbol{\mu}$ under this hypothesis. We can then conclude that if σ is known, as it is assumed to be the case in this section, then a test statistic of the form $V = s(\boldsymbol{z}, \boldsymbol{\Pi}\boldsymbol{y})$ has the same distribution as $s(\boldsymbol{z}, \sigma\boldsymbol{\Pi}\boldsymbol{\epsilon})$. □

4.7 Sampling under selective multiple hypotheses with unknown σ

In practice, σ is often unknown. To address this issue, we rely on the normalized versions of the test statistics V' and V'' introduced in Section 4.3, defined by

$$T'_j := \frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j\boldsymbol{y})}{\|\boldsymbol{y}\|^2} \quad \text{and} \quad T''_j := \max_{\boldsymbol{z}\in M_{i_j}} \frac{s\left(\boldsymbol{z}, \boldsymbol{\Pi}''\left((\boldsymbol{z}_\ell)_{\ell\neq j}\right)\boldsymbol{y}\right)}{\|\boldsymbol{y}\|^2} \tag{20}$$

where $\boldsymbol{z}_\ell = \arg\max_{\boldsymbol{z}\in\mathcal{Z}} s(\boldsymbol{z}, \boldsymbol{P}_\ell\boldsymbol{y})$. We will denote $\mathbb{G}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{G}''_{j,(i_1,\dots,i_q)}$ their cumulative distribution functions under the null, conditionally to $E(i_1, \dots, i_q)$.

We will also make use of a third assumption, here again fulfilled by s^{HSIC} and s^{ridge}:

($\mathbf{A_3}$) **Two-homogeneity:** It holds that $s(\boldsymbol{z}, t\boldsymbol{y}) = t^2 s(\boldsymbol{z}, \boldsymbol{y})$ for all $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{E} \times \mathcal{Z}$ and all $t > 0$.

Of note, normalizing the association score with respect to the labels does not affect the selection:

$$\forall \boldsymbol{y} \in \mathcal{Y}, \quad \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{y}) = \underset{\boldsymbol{z} \in \mathcal{Z}}{\arg\max}\, \frac{s(\boldsymbol{z}, \boldsymbol{y})}{\|\boldsymbol{y}\|^2} \tag{21}$$

If $\boldsymbol{\mu} = \mathbf{0}$, the distribution of the normalized statistics does not depend on σ, and the empirical cumulative distribution functions of normalized scores obtained by sampling under $\boldsymbol{\mu} = \mathbf{0}$ and any σ still provide a valid inference procedure :

Proposition 4.3. *Let s be an association score such that ($\mathbf{A_2}$) and ($\mathbf{A_3}$) hold. Let $T'_j = s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y})/\|\boldsymbol{y}\|^2$ and $T''_j = \max_{\boldsymbol{z} \in M_{i_j}} s(\boldsymbol{z}, \boldsymbol{\Pi}''_j \boldsymbol{y})/\|\boldsymbol{y}\|^2$, formed from $\boldsymbol{y}$ sampled from (9) with mean $\boldsymbol{\mu} = \mathbf{0}$, and any variance $\sigma > 0$. Then for all $\sigma' > 0$, their conditional null distributions $\mathbb{G}'_{j,(i_1,\dots,i_q)}$ and $\mathbb{G}''_{j,(i_1,\dots,i_q)}$ with mean $\mathbf{0}$ and variance σ' verify:*

$$\mathbb{G}'_{j,(i_1,\dots,i_q)}(T'_j) \sim \mathit{Unif}(0,1) \text{ and } \mathbb{G}''_{j,(i_1,\dots,i_q)}(T''_j) \sim \mathit{Unif}(0,1)$$

Proof. Let us consider two different normal models as defined in (9) under the global null hypothesis "$\boldsymbol{\mu} = \mathbf{0}$" and given by

$$y^{(1)} = \sigma^{(1)} \boldsymbol{\varepsilon}^{(1)} \quad \text{and} \quad y^{(2)} = \sigma^{(2)} \boldsymbol{\varepsilon}^{(2)}$$

Then

$$\frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y}^{(1)})}{\|\boldsymbol{y}^{(1)}\|^2} \sim \frac{s(\boldsymbol{c}_j, \boldsymbol{\Pi}'_j \boldsymbol{y}^{(2)})}{\|\boldsymbol{y}^{(2)}\|^2} \quad \text{and} \quad \frac{s(\boldsymbol{z}, \boldsymbol{\Pi}'' \left((\boldsymbol{z}_\ell)_{\ell \neq j}\right) \boldsymbol{y}^{(1)})}{\|\boldsymbol{y}^{(1)}\|^2} \sim \frac{s(\boldsymbol{z}, \boldsymbol{\Pi}'' \left((\boldsymbol{z}_\ell)_{\ell \neq j}\right) \boldsymbol{y}^{(2)})}{\|\boldsymbol{y}^{(2)}\|^2}.$$

The proof directly follows assumption ($\mathbf{A_3}$) applied with $t = \|\boldsymbol{y}^{(\cdot)}\|^2$. Proposition 4.3 is complementary to Proposition 4.2 and provides a selective inference procedure when σ is unknown, under the special null hypothesis $\boldsymbol{\mu} = 0$. □

Our final result investigates the testing procedures for the general null hypotheses (16) and (17)—not restricted to $\boldsymbol{\mu} = 0$—with an unknown σ. Recall that the decision rule is to reject the null hypothesis if the observed value of the statistic is greater than a given threshold t. We show that choosing t to be a quantile for the global null hypothesis ($\boldsymbol{\mu} = 0$) leads to a calibrated (for the type I error) non-selective procedure, see (22).

Proposition 4.4 (Global null achieves lowest observed significance)**.** *Let $\boldsymbol{Z} = \{\boldsymbol{z}_1, \dots, \boldsymbol{z}_q\}$ be q sequence motifs. Let $s(\cdot, \cdot)$ be a score such that ($\mathbf{A_1}$) and ($\mathbf{A_2}$) hold. Let $\boldsymbol{\mu} \in \mathcal{E}$ be such that*

$$\mathrm{H}_0 \ : \text{“}s(\boldsymbol{Z}, \boldsymbol{\mu}) = 0\text{”}$$

Then

$$\forall t > 0\,, \quad \sup_{\boldsymbol{\mu} \in \mathrm{H}_0} \mathbb{P}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon})}{\|\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}\|^2} \geq t\right] = \mathbb{P}\left[\frac{s(\boldsymbol{Z}, \boldsymbol{\epsilon})}{\|\boldsymbol{\epsilon}\|^2} \geq t\right] \tag{22}$$

We provide a proof in Appendix C. This proof makes an ad-hoc use of Anderson's theorem on a symmetric convex cone (whereas it is usually devoted to symmetric convex bodies).

Proposition 4.4 shows that data-split produces a calibrated procedure for testing the general null hypotheses (16) and (17) when sampling under the global null ($\boldsymbol{\mu} = 0$) the test statistics (20). We could not prove an equivalent statement for conditional null hypotheses, and Proposition 4.4 therefore does not guarantee the validity of a selective inference procedure sampling under the global null ($\boldsymbol{\mu} = 0$). Yet, we used it as a heuristic justification of SEISM and we observed that it leads to empirically calibrated procedures, see Section 5.2.

In view of Proposition 4.4 and its proof, one can see that the alternatives $\boldsymbol{\mu}$ such that $\|\boldsymbol{P}_q \boldsymbol{\mu}\|/\|\boldsymbol{\mu}\|$ is large have small power. As the selection procedure described in Section 3 achieves good results (Section 5.1), the chosen motifs $\boldsymbol{Z}$ should capture the principal components of $\boldsymbol{\mu}$, and therefore are such that $\|\boldsymbol{P}_q \boldsymbol{\mu}\|/\|\boldsymbol{\mu}\|$ should be small.

5 Results

5.1 SEISM performs as well as state-of-the-art *de novo* motif discovery methods

In order to compare the accuracy of our selection step with existing motif discovery algorithms, we use the 40 ENCODE Transcription Factors ChIP-seq datasets from K562 cells (ENCODE Project Consortium, 2004), each of which contains a known TF motif, denoted $\boldsymbol{m}^*$, derived using completely independent assays (Jolma et al., 2013). STREME (Bailey, 2021) and MEME (Bailey et al., 2006) are state-of-art bioinformatics methods for *de-novo* motifs discovery tasks. STREME identifies motifs that maximize a Fisher score of association between the presence of the motif and the binary class of sequences. By looking for maximum likelihood estimates of the parameters of a mixture model - made up of a background distribution and a model for generating k-mers at some positions - that may have produced a particular dataset using an expectation maximisation technique, MEME finds enriched motifs in this dataset. Finally CKN-seq (Chen et al., 2017) is a one-layer CNN tailored to small scale datasets. We set up STREME, MEME and SEISM to select 5 sequence motifs. SEISM is run with a regularization parameter $\lambda = 0.01$. CKN-seq jointly optimizes its filters, which notoriously leads to poor performances when few filters are used. We train it consequently over 128 filters. We measure these accuracy of all methods by comparing the motifs they discover with the known motif corresponding to the transcription factor $\boldsymbol{m}^*$. We rely on the Tomtom method (Gupta et al., 2007), which quantifies the probability that the euclidean distance between a random motif and $\boldsymbol{m}^*$ is lower than the distance between the discovered motif and $\boldsymbol{m}^*$. More precisely for each method we use the lowest Tomtom p-value between the known TF motif $\boldsymbol{m}^*$ and any of those discovered by the method. The Tomtom score is then defined as $-\log_{10}$ of the Tomtom p-value. We define the accuracy of the method as the proportion of experiments where the Tomtom score between its best match and the true TF motif was higher than some threshold.

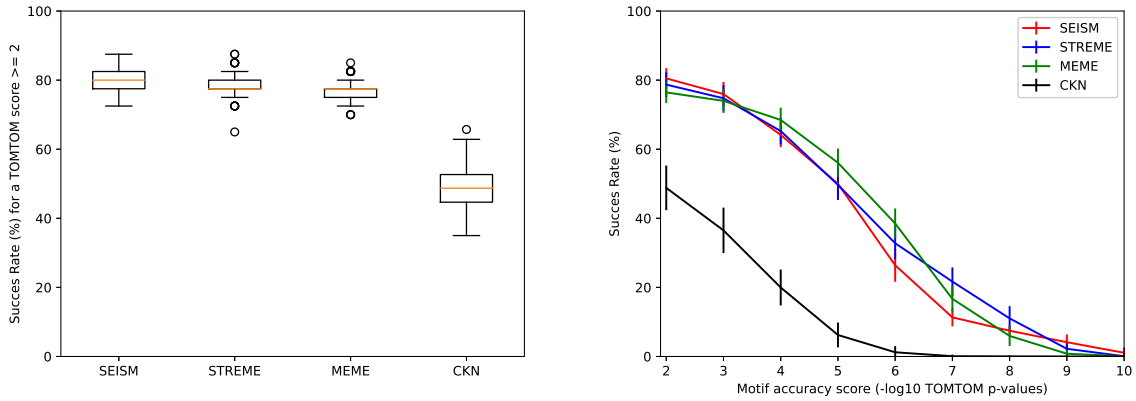


Figure 4: **Left**: Proportion of datasets where the true motif was detected by the designated algorithm. A true motif is said to be detected if its highest Tomtom score with the discovered motifs is greater than 2. **Right**: Accuracy of motif discovery algorithms on ENCODE TF ChIP-seq datasets. The curves display the proportion of ChIP-seq datasets were the best motif identified by the designated algorithm has a Tomtom score greater than x.

Figure 4 (left panel) demonstrates that SEISM is just as good as, if not superior to, state-of-the-art bioinformatics algorithms at detecting sequence motifs when thresholding Tomtom p-values at 0.01. The one-layer CNN with jointly optimized filters performs poorly in this experiments, emphasising the importance of greedy optimization for selecting the right motif.

Figure 4 (right panel) shows that SEISM performs slightly worse than STREME and MEME for high thresholds on the Tomtom scores. This suggests that the matrix z that SEISM identifies is close enough to the PWM matrix of the true motif, but farther away than the matrices identified by STREME or MEME. This discrepancy reflects a different usage of z to parameterize a distribution of k-mers. In practice, we

observe that on a given dataset, the p-values of the best motifs discovered by SEISM and STREME are not separated by more than 2 orders of magnitude, which leads to minor differences in the motifs, as illustrated in Table 1.

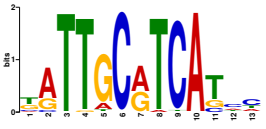
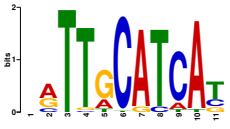
Reference motif *(ATF4_DBD)*	SEISM motif $p\text{-value}= 10^{-6}$	STREME motif $p\text{-value}= 3 \times 10^{-8}$
		

Table 1: Comparison between two discovered sequence motifs by SEISM or STREME, and the true motif *(ATF4_DBD)*

Both SEISM and MEME/STREME exploit a distribution of k-mers at the transcription factor binding site. MEME and STREME maximize the likelihood of a *categorical model*, whereby the matrix $\boldsymbol{z}$ directly defines the probability to observe each letter at each of the k sites:

$$\forall\big(\boldsymbol{u},\ \boldsymbol{z} \in \mathcal{Z}\big)\,, \quad \mathcal{L}_{\text{cat}}(\boldsymbol{u};\boldsymbol{z}) = \prod_{i=1}^{k} \boldsymbol{u}_i^T \boldsymbol{z}_i \tag{23}$$

SEISM on the other hand is based on a *Gaussian model*. Through representation (3), $\boldsymbol{z}$ is meant to maximize the Gaussian likelihood of a set of k-mers, *i.e.*

$$\forall\big(\boldsymbol{u},\ \boldsymbol{z} \in \mathcal{Z}\big)\,, \quad \mathcal{L}_{\text{gaus}}(\boldsymbol{u};\boldsymbol{z}) = C \prod_{i=1}^{k} e^{-\frac{\|\boldsymbol{u}_i - \boldsymbol{z}_i\|^2}{2\omega^2}} \tag{24}$$

where C is a constant such that the sum of probabilities over $\mathbb{R}^{4\times k}$ equals 1. If we consider a binary $\boldsymbol{y}$ to match the setting of MEME/STREME, this set is made of one k-mer for each positive sequence. Importantly, the true TF motifs from (Jolma et al., 2013) that we use to assess selection accuracies are also defined through the maximum likelihood in a categorical model, which can explain why the z obtained with MEME/STREME are closer to the true PWM than the one obtained with SEISM.

We now illustrate on a simple example how the same distribution of k-mers is parameterized by different matrices under the two models. To build an easy example, we focus on k-mers of length 1, with

$$P(A) = 0.3,\ P(C) = 0.4,\ P(G) = 0.1,\ P(T) = 0.2 \tag{25}$$

The matrix $\boldsymbol{z}_1 = (0.3, 0.4, 0.1, 0.2)^T$ used with the categorical model trivially constructs such a distribution. But using the same matrix in a Gaussian model with a parameter ω fixed as described in Appendix A leads to a slightly different distribution:

$$P(A) = 0.28,\ P(C) = 0.43,\ P(G) = 0.11,\ P(T) = 0.18 \tag{26}$$

A distribution closer to Equation (25) can be constructed with a Gaussian model parameterized by $\boldsymbol{z}_2 = (0.315, 0.38, 0.08, 0.225)^T$.

To clarify the relationships between those two motifs, we will rewrite (23) considering $\boldsymbol{u}$ is one hot encoded. That is, for each position i, it has only one 1 for letter $j(i)$ and 0's elsewhere:

$$\forall\big(\boldsymbol{u},\ \boldsymbol{z} \in \mathcal{Z}\big)\,, \quad \mathcal{L}_{\text{cat}}(\boldsymbol{u};\boldsymbol{z}) = \prod_{i=1}^{k} \boldsymbol{z}_{i,j(i)} \tag{27}$$

Assuming that the columns of $\boldsymbol{z}$ are normalized and $\omega = 1$, we can modify (24):

$$\forall\big(\boldsymbol{u},\ \boldsymbol{z} \in \mathcal{Z}\big)\,, \quad \mathcal{L}_{\text{gaus}}(\boldsymbol{u};\boldsymbol{z}) = C\prod_{i=1}^{k} e^{-\frac{\|\boldsymbol{u}_i - \boldsymbol{z}_j\|^2}{2\omega^2}} = C_2\prod_{i=1}^{k} e^{\boldsymbol{u}_i^T \boldsymbol{z}_i} = C_2\prod_{i=1}^{k} e^{\boldsymbol{z}_{i,j(i)}} \tag{28}$$

With the Gaussian model and a few assumptions, the motifs can the be seen as defining the log probability to observe each letter at each of the k sites. This gives us a new interpretation for the filters learned by CNNs and suggests that in this framework it might be interesting to constrain $e^{\boldsymbol{z}}$ rather than $\boldsymbol{z}$ to be in $\mathcal{Z}$.

We used a Gaussian activation function since it is closer to typical CNNs approaches. Our framework is generic enough to allow other activation functions based on the categorical model, or more realistic variants (Ruan & Stormo, 2017).

5.2 Statistical validity and performances

In order to assess the statistical validity and of the SEISM procedure with the different strategies, we simulate datasets under the null hypothesis. To that end, we draw one sequence motif $\tilde{\boldsymbol{z}}$ with length $k = 8$ for each simulated dataset using a uniform distribution on $\mathcal{Z}$ restricted to motifs with an information level fixed at 10 bits. Then, we draw a set of $n = 30$ biological sequences X as follows: all sites are generated according to a uniform distribution over A, C, T, G for all sequences, and for half of the sequences one k-mer is drawn according to the categorical model parameterized by $\tilde{\boldsymbol{z}}$. The phenotypes $\boldsymbol{y}$ are drawn from $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{C}_n)$ to generate data under the null hypothesis for calibration experiments, and from $\mathcal{N}(\boldsymbol{\varphi}^{\tilde{\boldsymbol{z}},\boldsymbol{X}}, \sigma^2\boldsymbol{C}_n)$ to generate data under the alternative for experiments on statistical power, with $\sigma = 0.1$ in both cases. We then run the SEISM procedure to select and test two sequence motifs. For both the data-split strategy and the hypersphere direction sampling one, the distribution from which the replicates are drawn uses the empirical variance from $\boldsymbol{y}$ as variance parameter. Although any choice for this parameter leads to a valid procedure, as described in Section 4.6, we make this choice for numerical stability considerations. For the data-split strategy, we sample 1000 replicates under the null hypothesis to compute the p-value. For SEISM, we sample $50,000$ replicates under the conditional null hypothesis using the hypersphere direction sampler, after $10,000$ burn-in iterations.

Figure 5 (top) shows the Q-Q plot of the distribution of quantiles of the uniform distribution against the p-values obtained across 1000 datasets under the null hypothesis for the data-split strategy and 100 datasets for the hypersphere direction sampling one. All the data points are well-aligned with the diagonal, which confirms the correct calibration of both the data-split and hypersphere direction sampling strategies, either considering the best motif or the center of the mesh and regardless of the size parameter.

Figure 5 (bottom) shows the same Q-Q plot on data generated under the alternative hypothesis. From this figure, we observe that on small datasets, the post-selection strategy is more powerful than the data-split one, regardless of the size of the mesh considered or the choice concerning the definition of the null hypothesis. The variance observed on the curves associated with the selective inference procedure is due to the presence of a weak residual signal after the first motif as a result of an imperfect selection step. Testing it with the best motif in the mesh captures this signal, resulting in curves under the diagonal. By contrast, focusing on the center of the meshes leads to testing motifs that do not capture this signal, placing us in the conservative situation, described at the end of Section 4.7. The residual signal is not well explained by the mesh's centers, and thus its component on the orthogonal of the span of the activation vector of the second motif is important. The larger the mesh, the farther its center is to the selected motif and thus the less signal it captures, which explains the differences between the two curves.

5.3 Computation costs

The section serves as an overview of how various user-specified parameters impact the computation time required by the post-selection inference procedure.

As discussed in 4.6, the hit-and-run algorithm is actually a rejection sampler. Its overall computation cost depends mainly on two characteristics: the cost of the selection step, that is the cost of selecting q motifs

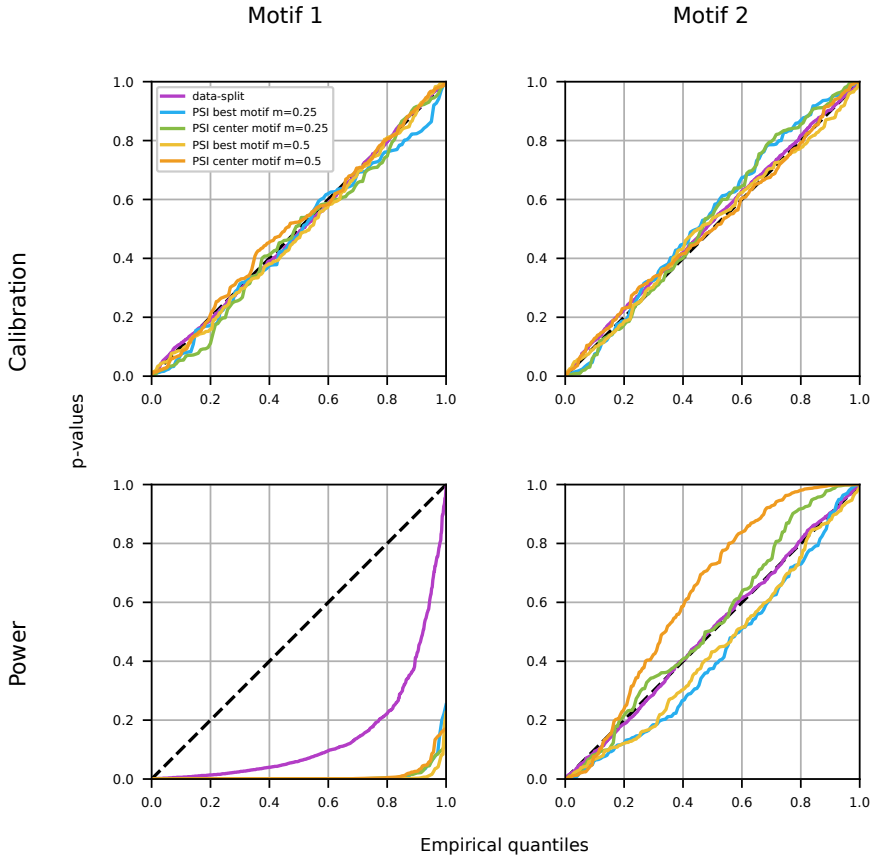


Figure 5: Q-Q plots obtained by applying data-split and different hit-and-run sampling strategies to select two motifs and test their association with an outcome. **Top**: data simulated under the null hypothesis. The proximity between the quantiles of the obtained p-values and those of the uniform distribution confirms that all SEISM strategies presented in this article are correctly calibrated. **Bottom**: data simulated under an alternative hypothesis, where the outcome depends on the activation $\varphi^{z,\tilde{\boldsymbol{X}}}$ of a single motif in the sequence. The distributions of the p-values computed with the post-selection inference (PSI) strategies have a larger deviation to the uniform distribution than the distributions of the p-values computed with the data-split strategy (purple).

for a given phenotype $\boldsymbol{y}$, and the acceptance rate. Although some parameters affect the selection cost, the acceptance rate is primarily responsible for determining if a user-specified combination of parameters results in a tractable configuration for the post-selection method in a reasonable amount of time. This rate is high compared with a naive rejection sampler over $\mathcal{E}$, as the hit-and-run strategy reduces the dimension over which the rejection step is performed: from n with a naive sampler to 1. Nonetheless some parameters may have a major impact on the rejection rate. To clarify it, we studied in Figure 6 the impact of several user-specified parameters — the number of motifs to be discovered, the precision of the meshes, the regularization parameter of the ridge score and the number of computation cores allowed during the rejection step of the hit-and-run sampler.

Although the number of motifs to be found by SEISM undoubtedly affects the selection cost, we can roughly consider that this relationship is linear. The upper left figure in Figure 6, however, demonstrates that the influence on the overall computing cost is superlinear, in line with the exponential growth of the number of distinct selection events one may describe with a fixed mesh size. As a result, the post-selection process quickly becomes intractable for discovering and test more than a few motifs.

We make a similar observation for mesh precision: computation time grows exponentially with the number of bins used to define the meshing. This can be explained by the exponential relationship between the number of bins and the number of different meshes (and thus the rejection rate). Of note, mesh precision has no impact on the selection time, and therefore the computation time is entirely explained by the acceptance rate.

We observe that the greater the regularization parameter λ, the lower the computation time. This can be explained by detailing its impact on the rejection rate. To understand it, it is necessary to note that the motifs are not selected over $\mathcal{Z}$, but over a less constrained set as described in 3. They are only projected onto $\mathcal{Z}$ at the end of the whole procedure, to ease their interpretation. The meshes are then defined over a vectorial space, leading to an infinite number of meshes. Compared to a small regularization parameter, a higher λ favors motifs resulting in a $\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ with a higher norm. With regard to the activation function, such motifs are located closer from the k-mers, and thus from $\mathcal{Z}$. λ has then no effect on the number of existing meshes, but impacts the number of *acceptable* ones, in the sense that they have a reasonable probability to be selected. A lower λ leads to better selection performances, but to a higher number of acceptable meshes, and thus to a lower acceptance rate. We empirically set $\lambda = 0.01$ to provide a good trade-off.

Finally, the rejection sampling step can be parallelized over several computation cores, which accelerates the whole procedure, as described in Section 4.4. As long as the acceptance rate is small enough, using j cores to parallelize the rejection step should roughly divide the computation time by j.

We can clearly identify limitations inherent to the use of the selective inference procedure. Although it is more powerful than the data-split approach, it can not be used in every situation. This latter approach does indeed not include any rejection step, and the only factor influencing its overall computation time is the selection time, only marginally influenced by the aforementioned parameters.

6 Discussion and future works

We have introduced a procedure to test the association between features learned by a neural network and the outcome predicted by this network. We did so by relying on the post-selection inference framework and formalizing the network training as a feature selection step. Along the way, we addressed general problems related to selective inference over composite hypotheses, which has implications beyond testing of features extracted by trained neural networks. In particular to our knowledge, all previous procedures had to work under the assumption that the variance was known. Our strategy to normalize the statistic to make it scale-free could easily be transferred to kernelPSI for testing the association of kernels with a trait, or to previous selective inference frameworks for testing groups of variables using sampling strategies (Slim et al., 2019; Reid et al., 2018).

Through the SEISM procedure, we are also drawing connections between neural networks for biological sequences and two related fields: sequence motif detection, and GWAS.

Sequence motif detection has been a major theme in bioinformatics for the past 30 years and many methods have been proposed to identify motifs that are over-represented in a set of sequence compared to some control class or background distribution. The earliest CNNs for regulatory genomics Alipanahi et al. (2015); Zhou & Troyanskaya (2015) already exploited the fact that trained convolution filters of the first layer could be interpreted as PWMs, and more recent work have sought to extract PWMs from entire multi-layer trained networks through attribution methods. The selection step of our procedure merely formalizes that training a one-layer CNN is equivalent to selecting a finite set of PWMs that have a maximal association to the outcome for some particular score. This formalization also highlights the specific way by which CNNs with exponential activation functions parameterize the distribution of k-mers at a binding site. Although the PWM returned by most bioinformatics models represents a categorical distribution—probability to draw each letter at each site, trained convolution matrices parameterize a Gaussian distribution. In practice, this difference leads to discrepancies between the trained convolution filters and PWMs learned using categorical likelihoods—including those offered by databases and often used as ground truth. This observation also suggests alternative sets of constraints for convolution filters—*e.g.*, each column of the pointwise exponential of the filter should belong to the simplex.

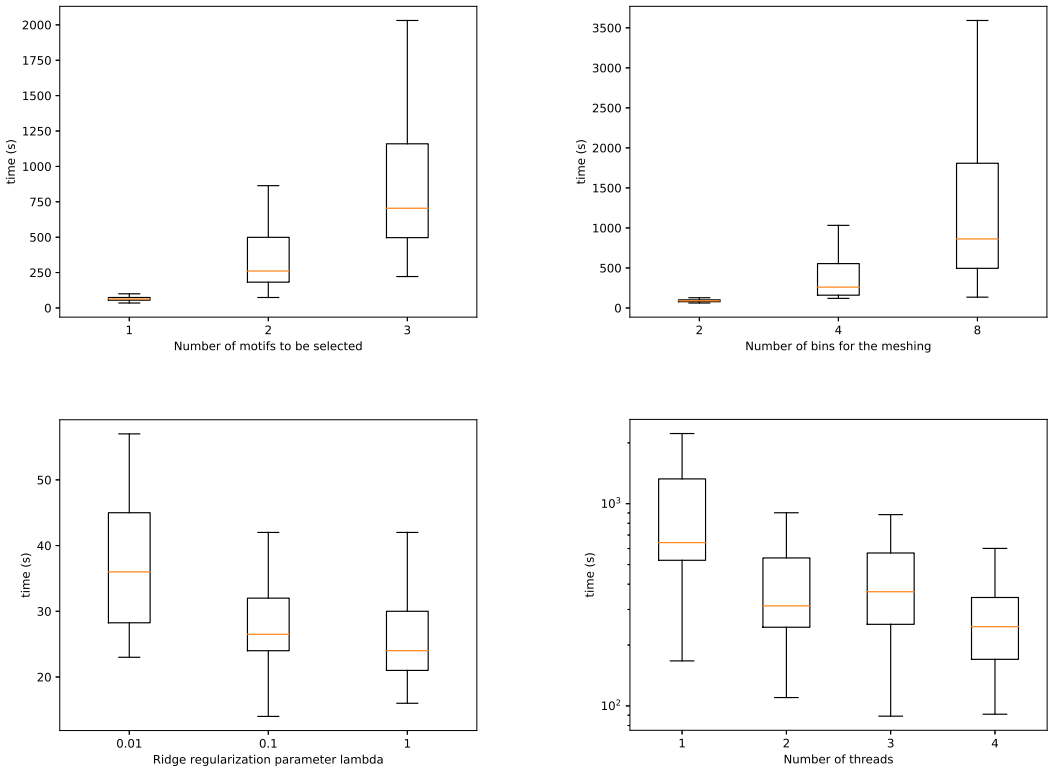


Figure 6: Impact of different parameters on the computation time for 100 replicates for the post-selection inference procedure. **Upper left:** Impact of the number of motifs to be discovered **Upper right:** Impact of the number of bins defining the meshes. **Bottom left:** Impact of the ridge regularization parameter. **Bottom right:** (Log scale) Impact of the number of threads over which the hit-and-run sampling is parallelized.

By providing an inference procedure for features extracted by the trained model, our work also connects neural networks for genomic sequences to GWAS. The good predictive performances of these neural networks is often explained by their ability to jointly learn an appropriate data representation and a regular prediction function acting on this representation. Nonetheless, the space from which these representations are learned is seldom formalized and to our knowledge the association of the extracted features with the predicted outcome is never tested. GWAS on the other hand relies on hypothesis testing, but commonly relies on relatively simple genomic variants such as single nucleotide polymorphisms (SNPs) or k-mer presence (Jaillard et al., 2018; Roux de Bézieux et al., 2022). Our framework paves the way to GWAS over richer sets of variants, *e.g.* capturing the presence of entire polymorphic genes through large convolution filters, or the interaction of simpler variants through multilayer or self-attention networks (Avsec et al., 2021a). This will require scaling to entire genomes as inputs, and making more complex networks, such as multi-layer CNNs and networks using attention mechanisms, amenable to inference. The most important step in achieving this goal is to formulate the training of these networks as a feature selection problem and formalize the association between these features and the phenotype. The inference framework might then be directly derived from this present work. For instance, we may test motif interactions derived from convolutional-attention networks (Ullah & Ben-Hur, 2021), a (motif, position) couple as selected in (Ditz et al., 2022) or motifs extracted by TF-MoDISco (Shrikumar et al., 2018). For this latter case, we can note that the inference procedure we describe is completely independent from the selection method, and we can then apply directly this procedure to TF-MoDISco's motifs. More relevant association scores than, *e.g.*, s^{ridge} could be devised for such motifs, but the procedure is nonetheless valid. For other features, the definition of a relevant association score, which meets the assumptions described in Section 4, is needed. A few practical problems may arise. First, the hit-and-run sampler requires the selection method to be stable, that is, running the selection method twice on the same input will lead to the same selection on features. This property is required to guarantee the theoretical convergence of the the algorithm but may not be necessary in practice. Second, some attention may be required to avoid the that computational cost become prohibitive, in particular depending on the regularity properties of the selection event leading to a higher rejection probability or to a higher number of replicates required. Granted that these technical challenges can be addressed, we are confident that extending SEISM to more general networks and corresponding features will benefit both the fields currently using these networks—such as regulatory genomics—and GWAS.

7 Acknowledgements

This work has been supported by ANR grants (FAST-BIG project ANR-17-CE23-0011-01 and PIECES project ANR-20-CE45-0017) and was performed using the computation facilities of the LBBE/PRABI.

We thank François Gindraud, Jean-Philippe Rasigade, Lotfi Slim and Dexiong Chen for the insightful discussions and support.

References

Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838, July 2015. ISSN 1087-0156. doi: 10.1038/nbt.3300. URL http://dx.doi.org/10.1038/nbt.3300.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, Oct 2021a. ISSN 1548-7105. doi: 10.1038/s41592-021-01252-x. URL https://doi.org/10.1038/s41592-021-01252-x.

Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nat Genet*, 53(3):354–366, February 2021b.

Timothy L Bailey. STREME: accurate and versatile sequence motif discovery. 37(18):2834–2840, 2021. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btab203. URL https://academic.oup.com/bioinformatics/article/37/18/2834/6184861.

Timothy L. Bailey, Nadya Williams, Chris Misleh, and Wilfred W. Li. MEME: discovering and analyzing DNA and protein sequence motifs. 34:W369–W373, 2006. ISSN 0305-1048. doi: 10.1093/nar/gkl198. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1538909/.

Leon Bungert and Philipp Wacker. The lion in the attic – a resolution of the borel–kolmogorov paradox, 2022. URL http://arxiv.org/abs/2009.04778.

Dexiong Chen, Laurent Jacob, and Julien Mairal. Biological sequence modeling with convolutional kernel networks. 2017. doi: 10.1101/217257. URL http://biorxiv.org/lookup/doi/10.1101/217257.

Jonas C. Ditz, Bernhard Reuter, and Nico Pfeifer. Convolutional motif kernel networks, 2022. URL http://arxiv.org/abs/2111.02272.

ENCODE Project Consortium. The ENCODE (ENCyclopedia of DNA elements) project. 306(5696):636–640, 2004. ISSN 1095-9203. doi: 10.1126/science.1105136.

Shobhit Gupta, John A. Stamatoyannopoulos, Timothy L. Bailey, and William Stafford Noble. Quantifying similarity between motifs. 8(2):R24, 2007. ISSN 1474-760X. doi: 10.1186/gb-2007-8-2-r24. URL https://doi.org/10.1186/gb-2007-8-2-r24.

R Harr, M Häggström, and P Gustafsson. Search algorithm for pattern match analysis of nucleic acid sequences. *Nucleic Acids Res*, 11(9):2943–2957, May 1983.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143:143, 2015.

Magali Jaillard, Leandro Lima, Maud Tournoud, Pierre Mahé, Alex van Belkum, Vincent Lacroix, and Laurent Jacob. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLOS Genetics*, 14(11):1–28, 11 2018. doi: 10.1371/journal.pgen.1007758. URL https://doi.org/10.1371/journal.pgen.1007758.

Arttu Jolma, Jian Yan, Thomas Whitington, Jarkko Toivonen, Kazuhiro R. Nitta, Pasi Rastas, Ekaterina Morgunova, Martin Enge, Mikko Taipale, Gonghong Wei, Kimmo Palin, Juan M. Vaquerizas, Renaud Vincentelli, Nicholas M. Luscombe, Timothy R. Hughes, Patrick Lemaire, Esko Ukkonen, Teemu Kivioja, and Jussi Taipale. DNA-binding specificities of human transcription factors. 152(1):327–339, 2013. ISSN 00928674. doi: 10.1016/j.cell.2012.12.009. URL https://linkinghub.elsevier.com/retrieve/pii/S0092867412014961.

David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res*, 28(5):739–750, March 2018.

Peter K. Koo and Sean R. Eddy. Representation learning of genomic sequence motifs with convolutional neural networks. 15(12):e1007560, 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007560. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007560. Publisher: Public Library of Science.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Yann LeCun and Yoshua Bengio. *Convolutional Networks for Images, Speech, and Time Series*, pp. 255–258. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262511029.

Jason D. Lee, Dennis L. Sun, Yuekai Sun, and Jonathan E. Taylor. Exact post-selection inference, with application to the lasso. 44(3):907–927, 2016. ISSN 0090-5364. doi: 10.1214/15-AOS1371. URL http://arxiv.org/abs/1311.6238.

M.A. Lifshits. On the absolute continuity of distributions of functionals of random processes. *Theory of Probability & Its Applications*, 27(3):600–607, 1983.

Joshua R. Loftus and Jonathan E. Taylor. Selective inference in regression models with groups of variables. *arXiv e-prints*, art. arXiv:1511.01478, November 2015.

Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book.

Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht, Wyeth W. Wasserman, and Sara Mostafavi. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, Oct 2022. ISSN 1471-0064. doi: 10.1038/s41576-022-00532-2. URL https://doi.org/10.1038/s41576-022-00532-2.

Gabrielle Ras, Ning Xie, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *J. Artif. Int. Res.*, 73, may 2022. ISSN 1076-9757. doi: 10.1613/jair.1.13200. URL https://doi.org/10.1613/jair.1.13200.

Stephen Reid and Robert Tibshirani. Sparse regression and marginal testing using cluster prototypes. 2013. URL http://arxiv.org/abs/1503.00334.

Stephen Reid, Jonathan Taylor, and Robert Tibshirani. A general framework for estimation and inference from clusters of features. 2015. URL http://arxiv.org/abs/1511.07839.

Stephen Reid, Jonathan Taylor, and Robert Tibshirani. A general framework for estimation and inference from clusters of features. *Journal of the American Statistical Association*, 113(521):280–293, 2018. doi: 10.1080/01621459.2016.1246368. URL https://doi.org/10.1080/01621459.2016.1246368.

Guillaume Rizk, Dominique Lavenier, and Rayan Chikhi. DSK: k-mer counting with very low memory usage. 29(5):652–653, 2013. ISSN 1367-4811. doi: 10.1093/bioinformatics/btt020.

Hector Roux de Bézieux, Leandro Lima, Fanny Perraudeau, Arnaud Mary, Sandrine Dudoit, and Laurent Jacob. CALDERA: finding all significant de Bruijn subgraphs for bacterial GWAS. *Bioinformatics*, 38:i36–i44, 06 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac238. URL https://doi.org/10.1093/bioinformatics/btac238.

Shuxiang Ruan and Gary D. Stormo. Inherent limitations of probabilistic models for protein-DNA binding specificity. 13(7):e1005638, 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005638. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005638. Publisher: Public Library of Science.

T D Schneider and R M Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, 18(20):6097–6100, October 1990.

Avanti Shrikumar, Katherine Tian, Žiga Avsec, Anna Shcherbina, Abhimanyu Banerjee, Mahfuza Sharmin, Surag Nair, and Anshul Kundaje. Technical note on transcription factor motif discovery from importance scores (tf-modisco) version 0.5.6.5, 2018. URL https://arxiv.org/abs/1811.00416.

Lotfi Slim, Clément Chatelain, Chloe-Agathe Azencott, and Jean-Philippe Vert. kernelPSI: a post-selection inference framework for nonlinear variable selection. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5857–5865. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/slim19a.html.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2): 742–769, 2019. doi: 10.1109/TIT.2018.2854560.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(47):1393–1434, 2012. URL http://jmlr.org/papers/v13/song12a.html.

Jonathan Taylor and Robert J. Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, 2015. doi: 10.1073/pnas.1507583112. URL https://www.pnas.org/doi/abs/10.1073/pnas.1507583112.

Jonathan Taylor, Richard Lockhart, Robert Tibshirani, and Ryan J Tibshirani. Exact post-selection inference for forward stepwise and least angle regression. pp. 32, 2014.

V.S. Tsirelson. The density of the distribution of the maximum of a gaussian process. *Theory of Probability & Its Applications*, 20(4):847–856, 1976.

Fahad Ullah and Asa Ben-Hur. A self-attention model for inferring cooperativity between regulatory features. 49(13):e77, 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab349. URL https://doi.org/10.1093/nar/gkab349.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

P M Visscher, N R Wray, Q Zhang, P Sklar, M I McCarthy, M A Brown, and J Yang. 10 years of GWAS discovery: Biology, function, and translation. *Am J Hum Genet*, 101(1):5–22, July 2017. doi: 10.1016/j.ajhg.2017.06.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5501872/.

Larry Wasserman and Kathryn Roeder. High-dimensional variable selection. 37(5):2178–2201, 2009. ISSN 0090-5364, 2168-8966. doi: 10.1214/08-AOS646. Publisher: Institute of Mathematical Statistics.

Ronald L. Wasserstein and Nicole A. Lazar. The ASA statement on p-values: Context, process, and purpose. *The American Statistician*, 70(2):129–133, 2016. doi: 10.1080/00031305.2016.1154108. URL https://doi.org/10.1080/00031305.2016.1154108.

Makoto Yamada, Yuta Umezu, Kenji Fukumizu, and Ichiro Takeuchi. Post selection inference with kernels. 2018.

Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods*, 12:931–4, 2015 Oct 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3547.

Supplemental Materials of "Neural Networks beyond explainability: Selective inference for sequence motifs"

A Tuning the activation bandwidth hyperparameter

The data representation $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$ depends on a hyperparameter ω controlling the bandwidth of the gaussian non-linearity (Equation 3): $\exp\left(-\frac{||\boldsymbol{z}_j-\boldsymbol{u}||^2}{2\omega^2}\right)$. Assuming that the positions are independant, we know that the expected value of the distance between a motif $\boldsymbol{z}$ and a k-mer u with length k is proportional to k.

In order to get an activation that does not depend on the length of the motifs, we simply set ω to be proportional to $\sqrt{k}$. From empirical tests, we set $\omega = \frac{\sqrt{0.9*k}}{2}$ to achieve good selection results by choosing the motif that maximizes the association score among a set of possible lengths.

B Disintegration of the selection event given by sequence motifs

In this section we consider the selection event:

$$E_{\text{cont.}}(\boldsymbol{Z}) := \left\{\boldsymbol{y}' \in \mathcal{E},\ \forall i \in \{1,\dots,q\}\ \ \underset{\boldsymbol{z}\in\mathcal{Z}}{\arg\max}\, s(\boldsymbol{z}, \boldsymbol{P}_i\boldsymbol{y}') = \boldsymbol{z}_i\right\}, \tag{S1}$$

given by the sequence of selected motifs $\boldsymbol{Z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_q)$. We denote by μ the law of $\boldsymbol{y}$ as given by Eq. (9), a Gaussian distribution on $\mathcal{E}$.

A first remark on the uniqueness of the selection

Consider the mapping $\pi : \mathcal{E} \to \mathcal{Z}^q$ given by $\pi(\boldsymbol{y}') = \boldsymbol{Z}$ where $\boldsymbol{Z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_q)$ is the sequence of motifs such that $\boldsymbol{y}' \in E_{\text{cont.}}(\boldsymbol{Z})$. It is not clear that π is well defined as a same $\boldsymbol{y}'$ may lead to the selection of at least two different motifs sequences $\boldsymbol{Z}$ and $\boldsymbol{Z}'$. As a first remark, we can see that the set of problematic $\boldsymbol{y}'$ is exactly

$$\mathcal{P} := \bigcup_{\boldsymbol{Z}\neq\boldsymbol{Z}'} E_{\text{cont.}}(\boldsymbol{Z}) \cap E_{\text{cont.}}(\boldsymbol{Z}')\,.$$

When one assumes that $\boldsymbol{Z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_q)$ is unique, one implicitly assumes that $\mu(\mathcal{P}) = 0$. For sufficiently regular scores, this is however the case. For sake of readability, we will not comprehensively study this issue but we will present an argument for the scores s^{HSIC} and s^{ridge}. In this case, we can circumvent this difficulty considering the Gaussian random field

$$\boldsymbol{z} \mapsto \langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{y}\rangle \text{ for (HSIC)} \quad \text{and} \quad \boldsymbol{z} \mapsto \langle \left(\|\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\|^2 + \lambda n\right)^{-1/2} \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{y}\rangle \text{ for (Ridge)}$$

indexed by $\mathcal{Z}$ where $\boldsymbol{y}$ is distributed with respect to a multivariate Gaussian distribution Eq. (9). Its autocovariance function is given by $(\boldsymbol{z},\boldsymbol{z}') \mapsto \sigma^2\langle \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}, \boldsymbol{\varphi}^{\boldsymbol{z}',\boldsymbol{X}}\rangle$ from Eq. (9) (one has to multiply by $\left(\|\boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}\|^2 + \lambda n\right)^{-1/2}\left(\|\boldsymbol{\varphi}^{\boldsymbol{z}',\boldsymbol{X}}\|^2 + \lambda n\right)^{-1/2}$ for the Ridge). The score is just the largest norm of this Gaussian random field. It is well established in theory of Gaussian random fields that the law of this maximum is regular and the argument maximum is unique. The interested reader may consult the pioneering work of Tsirelson (Tsirelson, 1976) and Lifshits (Lifshits, 1983). In Tsirelson's theorem, the parameter set is countable. This says that the same result holds true for separable bounded Gaussian processes, since in this case, the distribution of the supremum coincides a.s. with the one of the supremum on some countable nonrandom set. To avoid a cumbersome presentation, we will assume that almost surely the selected sequence motifs $\boldsymbol{Z} = (\boldsymbol{z}_1,\dots,\boldsymbol{z}_q)$ is uniquely defined, hence π is well defined.

The disintegration steps

To sample conditionally on (S1), one need to consider the conditional law with respect to this event. We will denote this law by $\mu_{\boldsymbol{Z}}$, it depends only on μ, $\boldsymbol{Z}$ and π. This law is described by the theorem of disintegration, see for instance (Ambrosio et al., 2005, Theorem 5.3.1). Denote ν the pushforward measure of μ by π, denoted by $\nu = \pi_{\#}\mu$, a probability measure on the set $\mathcal{Z}^q$ of $\boldsymbol{Z}$. By the disintegration theorem, there exists a ν-almost everywhere uniquely determined Borel family of probability measures $\mu_{\boldsymbol{Z}}$ (the though-after conditional distributions) such that

- **Supported by $E_{\text{cont.}}(\boldsymbol{Z})$:** $\mu_{\boldsymbol{Z}}\big\{\mathcal{E} \setminus \pi^{-1}(\boldsymbol{Z})\big\} = 0$ for ν-almost every $\boldsymbol{Z}$;
- **Expectation of the conditional expectation is the expectation:** It holds that, for every Borel test map $f : \mathcal{E} \to [0, +\infty]$,

$$\int_{\mathcal{E}} f \mathrm{d}\mu = \int_{\mathcal{Z}^q} \Big(\int_{\pi^{-1}(\boldsymbol{Z})} f \mathrm{d}\mu_{\boldsymbol{Z}} \Big) \mathrm{d}\nu(\boldsymbol{Z}) \,, \tag{S2}$$

where one can remark that $\pi^{-1}(\boldsymbol{Z}) = E_{\text{cont.}}(\boldsymbol{Z})$ by definition of π. Let us comment on this result regarding our purposes. First, we have mentioned that we known that the support $E_{\text{cont.}}(\boldsymbol{Z})$ is included in some subspace, say $\mathcal{S}$, defined by the first order conditions. Second, although one can use a rejection sampling strategy on the subspace $\mathcal{S}$ to draw points on the support $E_{\text{cont.}}(\boldsymbol{Z})$ (viewed as a subset of the same Hausdorff dimension as the subspace $\mathcal{S}$), it is not clear at all what should be the density of $\mu_{\boldsymbol{Z}}$. Indeed, the family of probability measures $\mu_{\boldsymbol{Z}}$ is the unique family that satisfies Eq. (S2). It implies that a measure $\mu_{\boldsymbol{Z}}$ depends on the others measures $\mu_{\boldsymbol{Z}'}$ and this dependency is geometrically given by the (piece-wise) topological sub-manifold given by the function $\boldsymbol{z} \mapsto \boldsymbol{\varphi}^{\boldsymbol{z},\boldsymbol{X}}$ from $\mathcal{Z}$ to $\mathcal{E}$.

From a practical view point, we tried various law for $\mu_{\boldsymbol{Z}}$ such as the uniform, or a rejection sampling based on the Gaussian distribution (9), but none of them matched the condition (S2). In the next subsection, we recall a toy example: the disintegration of the uniform measure on the sphere is not the uniform measure. Even in this simple geometrical example, the calculus of the conditional law might be seen as tedious. We believe that the calculus of $\mu_{\boldsymbol{Z}}$ is somehow out of reach for our purposes and our analysis with selection events defined by meshes more suited.

A toy example on the sphere

Let $\mathbb{S}$ be the 2-sphere embedded in the 3-Euclidean space. Let μ be the uniform measure on the sphere $\mathbb{S}$. Let $\{\mathcal{S}_\theta \,:\, \theta \in [0, \pi)\}$ be a family of sub-spaces of co-dimension 1 (hyper-planes) sharing $\mathrm{Span}\{(0, 0, 1)\}$ (say the north pole) as a revolution axis parameterized by θ. The parameter θ can be interpreted as the longitude.

Let $\bar{\pi}$ be the function that maps a point to its longitude modulo π. By spherical symmetries, the pushforward measure $\nu = \bar{\pi}_{\#}\mu$ is the uniform measure on $[0, \pi)$, so that $\mathrm{d}\nu(\theta) = (1/\pi)\mathrm{d}\theta$. Condition (S2) (the lhs of the equality below) is given by the coordinate integration system (the rhs) in:

$$\int_{\mathbb{S}} f \mathrm{d}\mu = \int_0^{\pi} \Big(\int_{\bar{\pi}^{-1}(\theta)} f \mathrm{d}\mu_\theta \Big) \mathrm{d}\nu(\theta) = \int_0^{\pi} \Big(\int_0^{2\pi} f(\theta, \phi) \frac{|\sin\phi|}{4\pi} \mathrm{d}\phi \Big) \mathrm{d}\theta \,,$$

where ϕ is the latitude. Note that $\bar{\pi}^{-1}(\theta) = \mathbb{S} \cap \mathcal{S}_\theta$ and it is in bijection with $[0, 2\pi)$ using the mapping that to a point maps its latitude. Using this representation, it is not hard to see that the uniform measure on $\bar{\pi}^{-1}(\theta)$ is given by $(1/2\pi)\mathbf{1}_{[0,2\pi)}(\phi)$ while the above equality shows that the conditional measure μ_θ of the uniform measure on the sphere has density $(1/4)|\sin\phi|\mathbf{1}_{[0,2\pi)}(\phi)$, see Figure S1. It proves that the disintegration of the uniform measure on the sphere is not the uniform measure, but rather a distribution that will put few mass around the poles and large mass around the equator.

C Proof of Proposition 4.4

Consider the orthogonal decompositon

$$\mathcal{E} = \mathcal{R} \oplus S \oplus \mathcal{T}$$

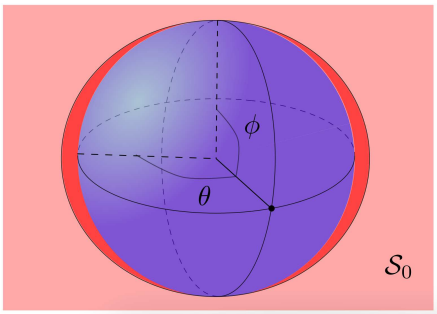


Figure S1: For $\theta = 0$, S_θ is the light red plan, the conditional measure $\mathrm{d}\mu_0(\phi)$ is depicted with a red area and is proportional to $|\sin(\phi)|$, which is not the uniform measure.

where $\mathcal{R}$ is the span of $\boldsymbol{\varphi}^{\boldsymbol{Z},\boldsymbol{X}}$, $\mathcal{T}$ is the span of $\boldsymbol{\mu}$ (orthogonal to $\mathcal{R}$ by Proposition 4.1), and $\mathcal{S}$ such that the equality holds. Consider $\boldsymbol{y} \in \mathcal{E}$ and its othogonal decomposition $\boldsymbol{y} = \boldsymbol{r} + \boldsymbol{s} + t\boldsymbol{e}$ where $\boldsymbol{e} = \boldsymbol{\mu}/\|\boldsymbol{\mu}\|_2$ is a unit norm vector that spans $\mathcal{T}$. Let $\tau > 0$ and note that it is enough to prove that

$$\mathbb{P}_{\boldsymbol{\mu}}\Big\{\frac{s(\boldsymbol{Z},\boldsymbol{Y})}{\|\boldsymbol{Y}\|^2} \leq \tau\Big\} \geq \mathbb{P}_{\mathbf{0}}\Big\{\frac{s(\boldsymbol{Z},\boldsymbol{Y})}{\|\boldsymbol{Y}\|^2} \leq \tau\Big\},$$

where $\boldsymbol{Y}$ is a random variable with the same distribution as $\boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$ (resp. $\sigma\boldsymbol{\epsilon}$) on the probability space defined by $\mathbb{P}_{\boldsymbol{\mu}}$ (resp. $\mathbb{P}_{\mathbf{0}}$). Not that the event decomposed as

$$\Big\{\boldsymbol{y} \,:\, \frac{s(\boldsymbol{Z},\boldsymbol{y})}{\|\boldsymbol{y}\|^2} \leq \tau\Big\} = \Big\{(t,\boldsymbol{r},\boldsymbol{s}) \,:\, s(\boldsymbol{Z},\boldsymbol{r}) \leq \tau(t^2\|\boldsymbol{\mu}\|^2 + \|\boldsymbol{r}\|^2 + \|\boldsymbol{s}\|^2)\Big\}$$

By othogonality, note that $\mathcal{L}_{\boldsymbol{\mu}}(\boldsymbol{r},\boldsymbol{s}) = \mathcal{L}_0(\boldsymbol{r},\boldsymbol{s})$ and this law is a centered Gaussian multivariate law. We deduce that the aforementioned probabilities are of the form

$$\mathbb{P}_{\mu}\Big\{\frac{s(\boldsymbol{Z},Y)}{\|Y\|^2} \leq \tau\Big\} = \int_0^\infty w_0(t)\varphi_\mu(t)\mathrm{d}t$$

where

$$w_0(t) = \mathbb{P}_0\Big\{s(\boldsymbol{Z},\boldsymbol{r}) \leq \tau(t^2\|\boldsymbol{\mu}\|^2 + \|\boldsymbol{r}\|^2 + \|\boldsymbol{s}\|^2)\Big\}$$
$$\varphi_\mu(t) = \exp\left(-(t-\mu_{\boldsymbol{e}})^2/2\right) + \exp\left(-(t+\mu_{\boldsymbol{e}})^2/2\right)$$

with $\mu_{\boldsymbol{e}} = \langle \boldsymbol{e}, \boldsymbol{\mu}\rangle = \|\boldsymbol{\mu}\|_2$. Note that $w_0 : (0,\infty) \to (0,1)$ is an increasing continuous function. It is an increasing homeomorphism and the Fubini's equality yields

$$\begin{aligned}
\mathbb{P}_{\mu}\Big\{\frac{s(\boldsymbol{Z},Y)}{\|Y\|^2} \leq \tau\Big\} &= \int_0^\infty w_0(t)\varphi_\mu(t)\mathrm{d}t \\
&= \int_0^\infty \int_0^1 \mathbf{1}_{\{u \leq w_0(t)\}}\mathrm{d}u\,\varphi_\mu(t)\mathrm{d}t \\
&= \int_0^\infty \int_0^1 \mathbf{1}_{\{w_0^{-1}(u) \leq t\}}\mathrm{d}u\,\varphi_\mu(t)\mathrm{d}t \\
&= \int_0^1 \int_{w_0^{-1}(u)}^\infty \varphi_\mu(t)\mathrm{d}t\mathrm{d}u
\end{aligned}$$

By Anderson's theorem, the measure of the interval $[-w_0^{-1}(u), w_0^{-1}(u)]$ for the centered Gaussian density is greater than the one for a non-centered Gaussian density with the same variance. As a result, we deduce that

$$\int_{w_0^{-1}(u)}^{\infty} \varphi_\mu(t)\mathrm{d}t \geq \int_{w_0^{-1}(u)}^{\infty} \varphi_0(t)\mathrm{d}t\,,$$

which achieves the proof.

Bibliography

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey Hinton. Neural Additive Models: Interpretable Machine Learning with Neural Nets, October 2021. URL http://arxiv.org/abs/2004.13912. arXiv:2004.13912 [cs, stat].

Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838, August 2015. ISSN 1546-1696. doi: 10.1038/nbt.3300. URL https://www.nature.com/articles/nbt.3300. Number: 8 Publisher: Nature Publishing Group.

Nicolas Altemose, Karen H. Miga, Mauro Maggioni, and Huntington F. Willard. Genomic Characterization of Large Heterochromatic Gaps in the Human Genome Assembly. *PLOS Computational Biology*, 10(5):e1003628, May 2014. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1003628. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003628. Publisher: Public Library of Science.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Lectures in mathematics ETH Zürich. Birkhäuser, Boston, 2005. ISBN 978-3-7643-2428-5.

T. W. Anderson. The Integral of a Symmetric Unimodal Function over a Symmetric Convex Set and Some Probability Inequalities. *Proceedings of the American Mathematical Society*, 6(2):170–176, 1955. ISSN 0002-9939. doi: 10.2307/2032333. URL https://www.jstor.org/stable/2032333. Publisher: American Mathematical Society.

Andreas Argyriou, Raphael Hauser, Charles A. Micchelli, and Massimiliano Pontil. A DC-programming algorithm for kernel selection. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pp. 41–48, Pittsburgh, Pennsylvania, 2006. ACM Press. ISBN 978-1-59593-383-6. doi: 10.1145/1143844.1143850. URL http://portal.acm.org/citation.cfm?doid=1143844.1143850.

Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, January 1966. ISSN 0030-8730, 0030-8730. doi: 10.2140/pjm.1966.16.1. URL http://msp.org/pjm/1966/16-1/p01.xhtml.

Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R. Ledsam, Agnieszka Grabska-Barwinska, Kyle R. Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, October 2021a. ISSN 1548-7105. doi: 10.1038/s41592-021-01252-x. URL https://www.nature.com/articles/s41592-021-01252-x. Number: 10 Publisher: Nature Publishing Group.

Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, and Julia Zeitlinger. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, March 2021b. ISSN 1546-1718. doi: 10.1038/s41588-021-00782-6. URL https://www.nature.com/articles/s41588-021-00782-6. Number: 3 Publisher: Nature Publishing Group.

Jean-Marc Azais, Yohann De Castro, and Fabrice Gamboa. Spike detection from inaccurate samplings, February 2014. URL http://arxiv.org/abs/1301.5873. arXiv:1301.5873 [math, stat].

Jean-Marc Azäis and Mario Wschebor. *Level Sets and Extrema of Random Processes and Fields: Azaïs/Level Sets and Extrema of Random Processes and Fields*. John Wiley & Sons, Inc., Hoboken, NJ, USA, February 2009. ISBN 978-0-470-43464-2 978-0-470-40933-6. doi: 10.1002/9780470434642. URL http://doi.wiley.com/10.1002/9780470434642.

Francis Bach. Breaking the Curse of Dimensionality with Convex Neural Networks, October 2016. URL http://arxiv.org/abs/1412.8690. arXiv:1412.8690 [cs, math, stat].

T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 2:28–36, 1994. ISSN 1553-0833.

T. L. Bailey and C. Elkan. The value of prior knowledge in discovering motifs with MEME. *Proceedings. International Conference on Intelligent Systems for Molecular Biology*, 3:21–29, 1995. ISSN 1553-0833.

Timothy L Bailey. STREME: accurate and versatile sequence motif discovery. *Bioinformatics*, 37(18):2834–2840, September 2021. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btab203. URL https://academic.oup.com/bioinformatics/article/37/18/2834/6184861.

Timothy L. Bailey, James Johnson, Charles E. Grant, and William S. Noble. The MEME Suite. *Nucleic Acids Research*, 43(W1):W39–W49, July 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv416. URL https://doi.org/10.1093/nar/gkv416.

J. E. Balmer and R. Blomhoff. A robust characterization of retinoic acid response elements based on a comparison of sites in three species. *The Journal of Steroid Biochemistry and Molecular Biology*, 96(5):347–354, September 2005. ISSN 0960-0760. doi: 10.1016/j.jsbmb.2005.05.005.

Yoav Benjamini. Selective Inference: The Silent Killer of Replicability. *Harvard Data Science Review*, 2(4), July 2020. ISSN 2644-2353, 2688-8513. doi: 10.1162/99608f92.fc62b261. URL https://hdsr.mitpress.mit.edu/pub/l39rpgyc/release/3.

Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, January 1995. ISSN 00359246. doi: 10.1111/j.2517-6161.1995.tb02031.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.2517-6161.1995.tb02031.x.

H. C. P. Berbee, C. G. E. Boender, A. H. G. Rinnooy Ran, C. L. Scheffer, R. L. Smith, and J. Telgen. Hit-and-run algorithms for the identification of nonredundant linear inequalities. *Mathematical Programming*, 37(2):184–207, June 1987. ISSN 1436-4646. doi: 10.1007/BF02591694. URL https://doi.org/10.1007/BF02591694.

Liane Bernstein, Alexander Sludds, Ryan Hamerly, Vivienne Sze, Joel Emer, and Dirk Englund. Freely scalable and reconfigurable optical hardware for deep learning. *Scientific Reports*, 11(1):3144, February 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-82543-3. URL https://www.nature.com/articles/s41598-021-82543-3. Number: 1 Publisher: Nature Publishing Group.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners, July 2020. URL http://arxiv.org/abs/2005.14165. arXiv:2005.14165 [cs].

Leon Bungert and Philipp Wacker. The lion in the attic – A resolution of the Borel–Kolmogorov paradox, April 2022. URL http://arxiv.org/abs/2009.04778. arXiv:2009.04778 [math].

Claude J. P. Bélisle, H. Edwin Romeijn, and Robert L. Smith. Hit-and-Run Algorithms for Generating Multivariate Distributions. *Mathematics of Operations Research*, 18(2):255–266, 1993. ISSN 0364-765X. URL https://www.jstor.org/stable/3690278. Publisher: INFORMS.

Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1):131–159, January 2002. ISSN 1573-0565. doi: 10.1023/A:1012450327387. URL https://doi.org/10.1023/A:1012450327387.

Amandine Chatagnon, Philippe Veber, Valérie Morin, Justin Bedo, Gérard Triqueneaux, Marie Sémon, Vincent Laudet, Florence d'Alché Buc, and Gérard Benoit. RAR/RXR binding dynamics distinguish pluripotency from differentiation associated cis-regulatory elements. *Nucleic Acids Research*, 43(10):4833–4854, May 2015. ISSN 1362-4962. doi: 10.1093/nar/gkv370.

Sneha Chaudhari, Varun Mithal, Gungor Polatkan, and Rohan Ramanath. An Attentive Survey of Attention Models. *ACM Transactions on Intelligent Systems and Technology*, 12(5):53:1–53:32, October 2021. ISSN 2157-6904. doi: 10.1145/3465055. URL https://doi.org/10.1145/3465055.

Dexiong Chen, Laurent Jacob, and Julien Mairal. Biological sequence modeling with convolutional kernel networks. *Bioinformatics*, 35(18):3294–3302, September 2019a. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz094. URL https://doi.org/10.1093/bioinformatics/btz094.

Dexiong Chen, Laurent Jacob, and Julien Mairal. Recurrent Kernel Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b. URL https://proceedings.neurips.cc/paper/2019/hash/d60743aab4b625940d39b3b51c3c6a78-Abstract.html.

Zhongsheng Chen, Michael Boehnke, Xiaoquan Wen, and Bhramar Mukherjee. Revisiting the genome-wide significance threshold for common variant GWAS. *G3: Genes|Genomes|Genetics*, 11(2):jkaa056, January 2021. ISSN 2160-1836. doi: 10.1093/g3journal/jkaa056. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8022962/.

Lenaic Chizat. Sparse Optimization on Measures with Over-parameterized Gradient Descent, November 2020. URL http://arxiv.org/abs/1907.10300. arXiv:1907.10300 [math, stat].

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20 (3):273–297, September 1995. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00994018. URL http://link.springer.com/10.1007/BF00994018.

Alex J. DeGrave, Joseph D. Janizek, and Su-In Lee. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, July 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00338-7. URL https://www.nature.com/articles/s42256-021-00338-7. Number: 7 Publisher: Nature Publishing Group.

Quentin Denoyelle, Vincent Duval, Gabriel Peyré, and Emmanuel Soubies. The Sliding Frank-Wolfe Algorithm and its Application to Super-Resolution Microscopy, November 2018. URL http://arxiv.org/abs/1811.06416. arXiv:1811.06416 [math].

Jonas C. Ditz, Bernhard Reuter, and Nico Pfeifer. Convolutional Motif Kernel Networks, May 2022. URL http://arxiv.org/abs/2111.02272. arXiv:2111.02272 [cs].

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1 -ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 272–279, Helsinki, Finland, 2008. ACM Press. ISBN 978-1-60558-205-4. doi: 10.1145/1390156.1390191. URL http://portal.acm.org/citation.cfm?doid=1390156.1390191.

Frank Dudbridge and Arief Gusnanto. Estimation of significance thresholds for genomewide association scans. *Genetic Epidemiology*, 32(3):227–234, April 2008. ISSN 0741-0395. doi: 10.1002/gepi.20297. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2573032/.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, April 2004. ISSN 0090-5364, 2168-8966. doi: 10.1214/009053604000000067. URL https://projecteuclid.org/journals/annals-of-statistics/volume-32/issue-2/Least-angle-regression/10.1214/009053604000000067.full. Publisher: Institute of Mathematical Statistics.

ENCODE Project Consortium. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science (New York, N.Y.)*, 306(5696):636–640, October 2004. ISSN 1095-9203. doi: 10.1126/science.1105136.

R. A. Fisher. The Goodness of Fit of Regression Formulae, and the Distribution of Regression Coefficients. *Journal of the Royal Statistical Society*, 85(4):597–612, 1922. ISSN 0952-8385. doi: 10.2307/2341124. URL https://www.jstor.org/stable/2341124. Publisher: [Wiley, Royal Statistical Society].

William Fithian, Dennis Sun, and Jonathan Taylor. Optimal Inference After Model Selection, April 2017. URL http://arxiv.org/abs/1410.2597. arXiv:1410.2597 [math, stat].

Andrea Galassi, Marco Lippi, and Paolo Torroni. Attention in Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4291–4308, October 2021. ISSN 2162-2388. doi: 10.1109/TNNLS.2020.3019893. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

Krishna Kishore Gali, Alison Sackville, Endale G. Tafesse, V.B. Reddy Lachagari, Kevin McPhee, Mick Hybl, Alexander Mikić, Petr Smýkal, Rebecca McGee, Judith Burstin, Claire Domoney, T.H. Noel Ellis, Bunyamin Tar'an, and Thomas D. Warkentin. Genome-Wide Association Mapping for Agronomic and Seed Quality Traits of Field Pea (Pisum sativum L.). *Frontiers in Plant Science*, 10:1538, November 2019. ISSN 1664-462X. doi: 10.3389/fpls.2019.01538. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6888555/.

Nicolas Galtier and J.R. Lobry. Relationships Between Genomic G+C Content, RNA Secondary Structures, and Optimal Growth Temperature in Prokaryotes. *Journal of Molecular Evolution*, 44(6):632–636, June 1997. ISSN 1432-1432. doi: 10.1007/PL00006186. URL https://doi.org/10.1007/PL00006186.

Christophe Giraud. *Introduction to High-Dimensional Statistics*. Chapman and Hall/CRC, August 2021. ISBN 978-1-00-315874-5. doi: 10.1201/9781003158745. URL https://www.taylorfrancis.com/books/mono/10.1201/9781003158745/introduction-high-dimensional-statistics-christophe-giraud.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring Statistical Dependence with Hilbert-Schmidt Norms. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita (eds.), *Algorithmic Learning Theory*, volume 3734, pp. 63–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-29242-5 978-3-540-31696-1. doi: 10.1007/11564089_7. URL http://link.springer.com/10.1007/11564089_7. Series Title: Lecture Notes in Computer Science.

Shobhit Gupta, John A. Stamatoyannopoulos, Timothy L. Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome Biology*, 8(2):R24, February 2007. ISSN 1474-760X. doi: 10.1186/gb-2007-8-2-r24. URL https://doi.org/10.1186/gb-2007-8-2-r24.

R Harr, M Häggström, and P Gustafsson. Search algorithm for pattern match analysis of nucleic acid sequences. *Nucleic Acids Research*, 11(9):2943–2957, May 1983. ISSN 0305-1048. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC325935/.

Hamid Reza Hassanzadeh and May D. Wang. DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins. pp. 178–183. IEEE Computer Society, December 2016. ISBN 978-1-5090-1611-2. doi: 10.1109/BIBM.2016.7822515. URL https://www.computer.org/csdl/proceedings-article/bibm/2016/07822515/12OmNqOwQHF.

Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity*. Chapman and Hall/CRC, New York, May 2015. ISBN 978-0-429-17158-1. doi: 10.1201/b18401.

Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C. Lin, Peter Laslo, Jason X. Cheng, Cornelis Murre, Harinder Singh, and Christopher K. Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell*, 38(4):576–589, May 2010. ISSN 1097-4164. doi: 10.1016/j.molcel.2010.05.004.

R. Horst and N. V. Thoai. DC Programming: Overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, October 1999. ISSN 1573-2878. doi: 10.1023/A:1021765131316. URL https://doi.org/10.1023/A:1021765131316.

John P. A. Ioannidis. Why Most Published Research Findings Are False. *PLoS Medicine*, 2(8):e124, August 2005. ISSN 1549-1676. doi: 10.1371/journal.pmed.0020124. URL https://dx.plos.org/10.1371/journal.pmed.0020124.

Magali Jaillard, Leandro Lima, Maud Tournoud, Pierre Mahé, Alex van Belkum, Vincent Lacroix, and Laurent Jacob. A fast and agnostic method for bacterial genome-wide association studies: Bridging the gap between k-mers and genetic events. *PLoS Genetics*, 14(11):e1007758, November 2018. ISSN 1553-7390. doi: 10.1371/journal.pgen.1007758. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6258240/.

Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37(15):2112–2120, August 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083. URL https://doi.org/10.1093/bioinformatics/btab083.

Arttu Jolma, Jian Yan, Thomas Whitington, Jarkko Toivonen, Kazuhiro R. Nitta, Pasi Rastas, Ekaterina Morgunova, Martin Enge, Mikko Taipale, Gonghong Wei, Kimmo Palin, Juan M. Vaquerizas, Renaud Vincentelli, Nicholas M. Luscombe, Timothy R. Hughes, Patrick Lemaire, Esko Ukkonen, Teemu Kivioja, and Jussi Taipale. DNA-Binding Specificities of Human Transcription Factors. *Cell*, 152(1-2):327–339, January 2013. ISSN 00928674. doi: 10.1016/j.cell.2012.12.009. URL https://linkinghub.elsevier.com/retrieve/pii/S0092867412014961.

Vanessa Isabell Jurtz, Alexander Rosenberg Johansen, Morten Nielsen, Jose Juan Almagro Armenteros, Henrik Nielsen, Casper Kaae Sønderby, Ole Winther, and Søren Kaae Sønderby. An introduction to deep learning on biological sequence data: examples and solutions. *Bioinformatics*, 33(22):3685–3690, November 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx531. URL https://doi.org/10.1093/bioinformatics/btx531.

David R. Kelley, Yakir A. Reshef, Maxwell Bileschi, David Belanger, Cory Y. McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Research*, 28(5):739–750, May 2018. ISSN 1088-9051. doi: 10.1101/gr.227819.117. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5932613/.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in Vision: A Survey. *ACM Computing Surveys*, 54(10s):200:1–200:41, September 2022. ISSN 0360-0300. doi: 10.1145/3505244. URL https://doi.org/10.1145/3505244.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/hash/5680522b8e2bb01943234bce7bf84534-Abstract.html.

Peter K. Koo and Sean R. Eddy. Representation learning of genomic sequence motifs with convolutional neural networks. *PLOS Computational Biology*, 15 (12):e1007560, December 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1007560. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007560. Publisher: Public Library of Science.

Peter K. Koo, Antonio Majdandzic, Matthew Ploenzke, Praveen Anand, and Steffan B. Paul. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLOS Computational Biology*, 17(5):e1008925, May 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008925. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008925. Publisher: Public Library of Science.

Jean-François Le Gall. *Intégration, Probabilités et Processus Aléatoires*. 2006. URL https://www.imo.universite-paris-saclay.fr/~jean-francois.le-gall/IPPA2.pdf.

Yann Lecun and Y. Bengio. Convolutional Networks for Images, Speech, and Time-Series. January 1995.

Jason D. Lee, Dennis L. Sun, Yuekai Sun, and Jonathan E. Taylor. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44 (3):907–927, June 2016. ISSN 0090-5364, 2168-8966. doi: 10.1214/15-AOS1371. URL https://projecteuclid.org/journals/annals-of-statistics/volume-44/issue-3/Exact-post-selection-inference-with-application-to-the-lasso/10.1214/15-AOS1371.full. Publisher: Institute of Mathematical Statistics.

C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, March 2004. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btg431. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btg431.

Christina Leslie, Eleazar Eskin, and William Stafford Noble. The Spectrum Kernel: a String Kernel for SVM Protein Classification. In *Biocomputing 2002*, pp. 564–575, Kauai, Hawaii, USA, December 2001. WORLD SCIENTIFIC. ISBN 978-981-02-4777-5 978-981-279-962-3. doi: 10.1142/9789812799623_0053. URL http://www.worldscientific.com/doi/abs/10.1142/9789812799623_0053.

M. A. Lifshits. On the Absolute Continuity of Distributions of Functionals of Random Processes. *Theory of Probability & Its Applications*, 27(3):600–607, January 1983. ISSN 0040-585X. doi: 10.1137/1127066. URL https://epubs.siam.org/doi/abs/10.1137/1127066. Publisher: Society for Industrial and Applied Mathematics.

Chi-Jen Lin, Patrick Haffner, Stephan Kanthak, Sören Sonnenburg, Gunnar Rätsch, Konrad Rieck, Igor Durdanovic, Eric Cosatto, Hans-Peter Graf, Elad Yom-Tov, Vikas Sindhwani, Sathiya Keerthi, Vikas Chandrakant Raykar, Ramani Duraiswami, Joaquin Quiñonero-Candela, Carl Edward Rasmussen, Christopher K. I. Williams, Stefanie Jegelka, Arthur Gretton, Ronan Collobert, Fabian Sinz, Gaëlle Loosli, Stephane Canu, Yoshua Bengio, and Yann LeCun. *Large-Scale Kernel Machines*. The MIT Press, Cambridge, Mass, illustrated edition edition, August 2007. ISBN 978-0-262-02625-3.

Joshua R. Loftus and Jonathan E. Taylor. Selective inference in regression models with groups of variables, November 2015. URL http://arxiv.org/abs/1511.01478. arXiv:1511.01478 [math, stat].

Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, and Trey Ideker. Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, 15(4):290–298, April 2018. ISSN 1548-7105. doi: 10.1038/nmeth.4627. URL https://www.nature.com/articles/nmeth.4627. Number: 4 Publisher: Nature Publishing Group.

Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional Kernel Networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/81ca0262c82e712e50c580c032d99b60-Paper.pdf.

Vincent Mallet and Jean-Philippe Vert. Reverse-Complement Equivariant Networks for DNA Sequences. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13511–13523. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/706608cfdbcc1886bb7eea5513f90133-Abstract.html.

L. Marsan and M. F. Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *Journal of Computational Biology: A Journal of Computational Molecular Cell Biology*, 7(3-4): 345–362, 2000. ISSN 1066-5277. doi: 10.1089/106652700750050826.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019. ISSN 00043702. doi: 10.1016/j.artint.2018.07.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370218305988.

Xu Min, Wanwen Zeng, Shengquan Chen, Ning Chen, Ting Chen, and Rui Jiang. Predicting enhancers with deep convolutional neural networks. *BMC Bioinformatics*, 18 (13):478, December 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1878-3. URL https://doi.org/10.1186/s12859-017-1878-3.

Christoph Molnar. *Interpretable Machine Learning*. Lulu.com, 2020. ISBN 978-0-244-76852-2. Google-Books-ID: RHjTxgEACAAJ.

Surag Nair, Avanti Shrikumar, Jacob Schreiber, and Anshul Kundaje. fastISM: performant in silico saturation mutagenesis for convolutional neural networks. *Bioinformatics*, 38(9):2397–2403, May 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac135. URL https://doi.org/10.1093/bioinformatics/btac135.

Gherman Novakovsky, Nick Dexter, Maxwell W. Libbrecht, Wyeth W. Wasserman, and Sara Mostafavi. Obtaining genetics insights from deep learning via explainable artificial intelligence. *Nature Reviews Genetics*, pp. 1–13, October 2022a. ISSN 1471-0064. doi: 10.1038/s41576-022-00532-2. URL https://www.nature.com/articles/s41576-022-00532-2. Publisher: Nature Publishing Group.

Gherman Novakovsky, Oriol Fornes, Manu Saraswat, Sara Mostafavi, and Wyeth W. Wasserman. ExplaiNN: interpretable and transparent neural networks for genomics, November 2022b. URL https://www.biorxiv.org/content/10.1101/2022.05.20.492818v3. Pages: 2022.05.20.492818 Section: New Results.

Sungjoon Park, Yookyung Koh, Hwisang Jeon, Hyunjae Kim, Yoonsun Yeo, and Jaewoo Kang. Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Scientific Reports*, 10(1):13413, August 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-70218-4. URL https://www.nature.com/articles/s41598-020-70218-4. Number: 1 Publisher: Nature Publishing Group.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, pp. III–1310–III–1318, Atlanta, GA, USA, June 2013. JMLR.org.

Giulio Pavesi, Paolo Mereghetti, Giancarlo Mauri, and Graziano Pesole. Weeder Web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. *Nucleic Acids Research*, 32(Web Server issue):W199–W203, July 2004. ISSN 0305-1048. doi: 10.1093/nar/gkh465. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC441603/.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport, March 2020. URL http://arxiv.org/abs/1803.00567. arXiv:1803.00567 [stat].

Daniel Quang and Xiaohui Xie. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research*, 44 (11):e107, June 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw226. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4914104/.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

Atif Rahman, Ingileif Hallgrímsdóttir, Michael Eisen, and Lior Pachter. Association mapping from sequencing reads using k-mers. *eLife*, 7:e32920, June 2018. ISSN 2050-084X. doi: 10.7554/eLife.32920. URL https://doi.org/10.7554/eLife.32920. Publisher: eLife Sciences Publications, Ltd.

Stephen Reid and Robert Tibshirani. Sparse regression and marginal testing using cluster prototypes, March 2015. URL http://arxiv.org/abs/1503.00334. arXiv:1503.00334 [stat].

Stephen Reid, Jonathan Taylor, and Robert Tibshirani. A general framework for estimation and inference from clusters of features, November 2015. URL http://arxiv.org/abs/1511.07839. arXiv:1511.07839 [stat].

Neil Risch and Kathleen Merikangas. The Future of Genetic Studies of Complex Human Diseases. *Science*, 273(5281):1516–1517, September 1996. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.273.5281.1516. URL https://www.science.org/doi/10.1126/science.273.5281.1516.

G. Rizk, D. Lavenier, and R. Chikhi. DSK: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653, March 2013. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btt020. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btt020.

Gordon Robertson, Martin Hirst, Matthew Bainbridge, Misha Bilenky, Yongjun Zhao, Thomas Zeng, Ghia Euskirchen, Bridget Bernier, Richard Varhol, Allen Delaney, Nina Thiessen, Obi L. Griffith, Ann He, Marco Marra, Michael Snyder, and Steven Jones. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, 4(8):651–657, August 2007. ISSN 1548-7105. doi: 10.1038/nmeth1068. URL https://www.nature.com/articles/nmeth1068. Number: 8 Publisher: Nature Publishing Group.

Hector Roux de Bézieux, Leandro Lima, Fanny Perraudeau, Arnaud Mary, Sandrine Dudoit, and Laurent Jacob. CALDERA: finding all significant de Bruijn subgraphs for bacterial GWAS. *Bioinformatics*, 38(Suppl 1):i36–i44, June 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac238. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9235473/.

Shuxiang Ruan and Gary D. Stormo. Inherent limitations of probabilistic models for protein-DNA binding specificity. *PLOS Computational Biology*, 13(7):e1005638, July 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005638. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005638. Publisher: Public Library of Science.

Gunnar Rätsch, Sören Sonnenburg, and Christin Schäfer. Learning Interpretable SVMs for Biological Sequence Classification. *BMC Bioinformatics*, 7(1):S9, March 2006. ISSN 1471-2105. doi: 10.1186/1471-2105-7-S1-S9. URL https://doi.org/10.1186/1471-2105-7-S1-S9.

T D Schneider and R M Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, October 1990. ISSN 0305-1048. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC332411/.

Jacob Schreiber, Surag Nair, Akshay Balsubramani, and Anshul Kundaje. Accelerating in silico saturation mutagenesis using compressed sensing. *Bioinformatics (Oxford, England)*, 38(14):3557–3564, July 2022. ISSN 1367-4811. doi: 10.1093/bioinformatics/btac385.

Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel Methods in Computational Biology a book by Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert.* The MIT Press, the mit press edition, July 2004. ISBN 978-0-262-19509-6. URL https://bookshop.org/p/books/kernel-methods-in-computational-biology-bernhard-scholkopf/14607449.

Sofia Serrano and Noah A. Smith. Is Attention Interpretable?, June 2019. URL http://arxiv.org/abs/1906.03731. arXiv:1906.03731 [cs].

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 3145–3153, Sydney, NSW, Australia, August 2017. JMLR.org.

Avanti Shrikumar, Katherine Tian, Žiga Avsec, Anna Shcherbina, Abhimanyu Banerjee, Mahfuza Sharmin, Surag Nair, and Anshul Kundaje. Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5. Technical report, October 2018. URL https://ui.adsabs.harvard.edu/abs/2018arXiv181100416S. Publication Title: arXiv e-prints ADS Bibcode: 2018arXiv181100416S Type: article.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, April 2014. URL http://arxiv.org/abs/1312.6034. arXiv:1312.6034 [cs].

Lotfi Slim, Clément Chatelain, Chloe-Agathe Azencott, and Jean-Philippe Vert. kernelPSI: a Post-Selection Inference Framework for Nonlinear Variable Selection. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5857–5865. PMLR, May 2019. URL https://proceedings.mlr.press/v97/slim19a.html. ISSN: 2640-3498.

Robert L. Smith. Efficient Monte Carlo Procedures for Generating Points Uniformly Distributed Over Bounded Regions. *Operations Research*, 32(6):1296–1308, 1984. ISSN 0030-364X. URL https://www.jstor.org/stable/170949. Publisher: INFORMS.

Mahdi Soltanolkotabi, Adel Javanmard, and Jason D. Lee. Theoretical Insights Into the Optimization Landscape of Over-Parameterized Shallow Neural Networks. *IEEE Transactions on Information Theory*, 65(2):742–769, February 2019. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2018.2854560. URL https://ieeexplore.ieee.org/document/8409482/.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 13 (1):1393–1434, May 2012. ISSN 1532-4435.

Vitor Sousa and Jody Hey. Understanding the origin of species with genome-scale data: modelling gene flow. *Nature Reviews Genetics*, 14(6):404–414, June 2013. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg3446. URL https://www.nature.com/articles/nrg3446.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for Simplicity: The All Convolutional Net, April 2015. URL http://arxiv.org/abs/1412.6806. arXiv:1412.6806 [cs].

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks, June 2017. URL http://arxiv.org/abs/1703.01365. arXiv:1703.01365 [cs].

Jonathan Taylor and Robert J. Tibshirani. Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112(25):7629–7634, June 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1507583112. URL http://www.pnas.org/lookup/doi/10.1073/pnas.1507583112.

Dongmei Tian, Pei Wang, Bixia Tang, Xufei Teng, Cuiping Li, Xiaonan Liu, Dong Zou, Shuhui Song, and Zhang Zhang. GWAS Atlas: a curated resource of genome-wide variant-trait associations in plants and animals. *Nucleic Acids Research*, 48(D1):D927–D932, January 2020. ISSN 0305-1048. doi: 10.1093/nar/gkz828. URL https://doi.org/10.1093/nar/gkz828.

Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 0035-9246. URL https://www.jstor.org/stable/2346178. Publisher: [Royal Statistical Society, Wiley].

Ryan Tibshirani, Rob Tibshirani, Jonathan Taylor, Joshua Loftus, Stephen Reid, and Jelena Markovic. selectiveInference: Tools for Post-Selection Inference, September 2019. URL https://CRAN.R-project.org/package=selectiveInference.

Ryan J. Tibshirani, Jonathan Taylor, Richard Lockhart, and Robert Tibshirani. Exact Post-Selection Inference for Sequential Regression Procedures, October 2015. URL http://arxiv.org/abs/1401.3889. arXiv:1401.3889 [stat].

V. S. Tsirel'son. The Density of the Distribution of the Maximum of a Gaussian Process. *Theory of Probability & Its Applications*, 20(4):847–856, September 1976. ISSN 0040-585X. doi: 10.1137/1120092. URL https://epubs.siam.org/doi/10.1137/1120092. Publisher: Society for Industrial and Applied Mathematics.

Fahad Ullah and Asa Ben-Hur. A self-attention model for inferring cooperativity between regulatory features. *Nucleic Acids Research*, 49(13):e77, July 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab349. URL https://doi.org/10.1093/nar/gkab349.

Juan M. Vaquerizas, Sarah K. Kummerfeld, Sarah A. Teichmann, and Nicholas M. Luscombe. A census of human transcription factors: function, expression and evolution. *Nature Reviews. Genetics*, 10(4):252–263, April 2009. ISSN 1471-0064. doi: 10.1038/nrg2538.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Antoine Villié, Philippe Veber, Yohann de Castro, and Laurent Jacob. Neural Networks beyond explainability: Selective inference for sequence motifs, December 2022. URL http://arxiv.org/abs/2212.12542. arXiv:2212.12542 [cs, q-bio, stat].

Peter M. Visscher, Naomi R. Wray, Qian Zhang, Pamela Sklar, Mark I. McCarthy, Matthew A. Brown, and Jian Yang. 10 Years of GWAS Discovery: Biology, Function, and Translation. *American Journal of Human Genetics*, 101(1):5–22, July 2017. ISSN 0002-9297. doi: 10.1016/j.ajhg.2017.06.005. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5501872/.

Larry Wasserman and Kathryn Roeder. High-dimensional variable selection. *The Annals of Statistics*, 37(5A):2178–2201, October 2009. ISSN 0090-5364, 2168-8966. doi: 10.1214/08-AOS646. URL https://projecteuclid.org/journals/annals-of-statistics/volume-37/issue-5A/High-dimensional-variable-selection/10.1214/08-AOS646.full. Publisher: Institute of Mathematical Statistics.

Ronald L. Wasserstein and Nicole A. Lazar. The ASA Statement on p-Values: Context, Process, and Purpose. *The American Statistician*, 70(2):129–133, April 2016. ISSN 0003-1305. doi: 10.1080/00031305.2016.1154108. URL https://doi.org/10.1080/00031305.2016.1154108. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00031305.2016.1154108.

Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, June 2007. ISSN 1476-4687. doi: 10.1038/nature05911.

Danielle Welter, Jacqueline MacArthur, Joannella Morales, Tony Burdett, Peggy Hall, Heather Junkins, Alan Klemm, Paul Flicek, Teri Manolio, Lucia Hindorff, and Helen Parkinson. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic Acids Research*, 42(Database issue):D1001–D1006, January 2014. ISSN 0305-1048. doi: 10.1093/nar/gkt1229. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3965119/.

Makoto Yamada, Yuta Umezu, Kenji Fukumizu, and Ichiro Takeuchi. Post Selection Inference with Kernels. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 152–160. PMLR, March 2018. URL https://proceedings.mlr.press/v84/yamada18a.html. ISSN: 2640-3498.

Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks, November 2013. URL http://arxiv.org/abs/1311.2901. arXiv:1311.2901 [cs].

Qinyi Zhang, Sarah Filippi, Arthur Gretton, and Dino Sejdinovic. Large-Scale Kernel Methods for Independence Testing. *Statistics and Computing*, 28(1):113–130, January 2018. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-016-9721-7. URL http://arxiv.org/abs/1606.07892. arXiv:1606.07892 [stat].

Yuchen Zhang, Percy Liang, and Martin J. Wainwright. Convexified Convolutional Neural Networks, September 2016. URL http://arxiv.org/abs/1609.01000. arXiv:1609.01000 [cs].

Hannah Zhou, Avanti Shrikumar, and Anshul Kundaje. Towards a Better Understanding of Reverse-Complement Equivariance for Deep Learning Models in Genomics. In *Proceedings of the 16th Machine Learning in Computational Biology meeting*, pp. 1–33. PMLR, January 2022. URL https://proceedings.mlr.press/v165/zhou22a.html. ISSN: 2640-3498.

Jian Zhou and Olga G. Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, October 2015. ISSN 1548-7105. doi: 10.1038/nmeth.3547. URL https://www.nature.com/articles/nmeth.3547. Number: 10 Publisher: Nature Publishing Group.

Hui Zou and Trevor Hastie. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 1369-7412. URL https://www.jstor.org/stable/3647580. Publisher: [Royal Statistical Society, Wiley].

Abstract:

Over the past decade, neural networks have been successful at making predictions from biological sequences, especially in the context of regulatory genomics. These networks are mainly evaluated on their predictive capacities, and are often criticized for their lack of interpretability. Some methods from the bioinformatics literature, on the other hand, seek to help understand the underlying biology, by selecting genomic variants significantly associated with a biological trait. Although having led to many discoveries in the recent years, these methods remain subject to many limitations. Recently, explainability tools have been devised to extract interpretable biological features from the trained neural networks. These networks can consequently be seen as methods for selecting genomic variants, and can help to overcome some of the aforementioned limitations. Quantifying the significance of associations between interpretable features and biological traits has only received little attention to our knowledge in the context of neural networks. We therefore propose to go beyond the notion of explicability for machine learning, by seeking to statistically quantify the association between variants extracted from neural networks and biological traits, in order to participate in building a bridge between machine learning methods and computational biology. In particular, we formalize the link between the training of a neural network and the selection of biological variants, and we propose different modifications to these networks, in order to improve their performances as selection methods. We also propose a valid test procedure for the selected variants, based on recent advances in post-selection inference.

Résumé :

Les réseaux de neurones artificiels ont récemment été utilisés avec succès pour faire des prédictions sur des séquences biologiques. Ces réseaux sont principalement évalués pour leurs capacités prédictives, et sont souvent critiqués pour leur manque d'interprétabilité. D'un autre côté, plusieurs méthodes issues de la littérature bioinformatique cherchent à aider à comprendre les mécanismes biologiques sous-jacents, en sélectionnant des variants génomiques significativement associés avec le trait biologique d'intérêt. Bien qu'elles aient mené à de nombreuses découvertes durant les dernières années, ces méthodes restent soumises à certaines limitations. Récemment, des outils cherchent à expliquer les prédictions des réseaux de neurones, en extrayant des caractéristiques biologiques interprétables de ces réseaux entraînés. Les réseaux de neurones peuvent alors être compris comme des méthodes permettant de sélectionner des variants génomiques, et peuvent permettre de dépasser certaines des limitations préalablement mentionnées. Mais à notre connaissance, la quantification de la significativité de l'association entre les caractéristiques biologiques extraites de ces réseaux et les traits biologiques d'intérêt n'a reçu que peu d'attention. Nous proposons donc de dépasser la notion d'explicabilité pour l'apprentissage automatique, en cherchant à quantifier statistiquement l'association entre les variants issus de réseaux de neurones et le phénotype, afin de participer à créer un lien entre les méthodes d'apprentissage automatiques et celles provenant de la biologie computationnelle. En particulier, nous formalisons le lien entre réseaux de neurones et sélection de variants biologiques, et nous proposons différentes modifications à ces réseaux, afin d'améliorer leurs performances en tant que méthodes de sélection. Nous proposons également une procédure de test valide pour les variants ainsi sélectionnés, issue des avancées récentes en inférence post-sélection.