Structured priors for supervised learning in computational biology

Laurent Jacob

To cite this version:

Laurent Jacob. Structured priors for supervised learning in computational biology . Life Sciences [q-bio]. École Nationale Supérieure des Mines de Paris, 2009. English. NNT : 2009ENMP1644 . pastel-00005743

HAL Id: pastel-00005743

https://pastel.hal.science/pastel-00005743

Submitted on 21 Jan 2010

MINES
ParisTech

ED n°431 : Information, Communication, Modélisation et Simulation

THÈSE

pour obtenir le grade de

DOCTEUR DE L'ÉCOLE NATIONALE SUPÉRIEURE DES MINES DE PARIS

Spécialité "Bioinformatique"

présentée et soutenue publiquement par

Laurent JACOB

le 25 Novembre 2009

STRUCTURED PRIORS FOR SUPERVISED LEARNING IN COMPUTATIONAL BIOLOGY

Directeur de thèse : Jean-Philippe Vert

Jury

M.	Christophe Ambroise	Université d'Évry	Président
M.	Massimiliano Pontil	University College London	Rapporteur
M.	Didier Rognan	Université Strasbourg I	Rapporteur
M.	Francis Bach	INRIA - École Normale Supérieure	Examinateur
M.	Yves Grandvalet	Université de Technologie de Compiègne	Examinateur
M.	Jean-Philippe Vert	Mines ParisTech	Examinateur

Only for you, children of doctrine and learning, have we written this work. Examine this book, ponder the meaning we have dispersed in various places and gathered again; what we have concealed in one place we have disclosed in another, that it may be understood by your wisdom.

H. C. A. von Nettesheim, *De occulta philosophia*, 3, 65.

I have understood. And the certainty that there is nothing to understand should be my peace, my triumph. But I am here, and They are looking for me, thinking I possess the revelation They sordidly desire. It isn't enough to have understood, if others refuse and continue to interrogate.

U. Eco, *Foucault's Pendulum.*, *Malkhut*, 120.

Remerciements

Je tiens en premier lieu à remercier Jean-Philippe, qui m'a encadré pendant la durée de cette thèse. Jean-Philippe a toujours su m'indiquer de bonnes directions de recherche quand il le fallait, me laisser chercher seul quand il le fallait, et a ce talent de savoir expliquer les concepts les plus techniques en des termes très intuitifs. Je pense avoir énormément appris à son contact, j'ai sincèrement apprécié de travailler avec lui et lui suis reconnaissant pour le temps qu'il ma consacré et toutes les opportunités qu'il m'a données au cours de cette thèse.

Mes remerciements vont également à Francis Bach, Guillaume Obozinski, Véronique Stoven et Martial Hue qui sans être mes directeurs de thèse, ont grandement contribué à ma formation en répondant avec patience à mes questions de jeune naïf et inculte sur l'optimisation convexe, les statistiques, la biologie structurale et l'algèbre.

Je suis également très reconnaissant à Massimiliano Pontil et Didier Rognan d'avoir accepté d'être rapporteurs de cette thèse, ainsi qu'aux autres membres du jury, Christophe Ambroise, Francis Bach et Yves Grandvalet pour toutes leurs questions, remarques et suggestions.

Par ailleurs je voudrais remercier Emmanuel Barillot, directeur de l'U900 Inserm, pour m'avoir accueilli dans son laboratoire avant même la création de tout partenariat officiel entre le CBIO des Mines et l'institut Curie.

J'ai eu la chance au cours de ma thèse de travailler quelque temps à l'Université de Berkeley, et remercie les professeurs Bin Yu et Michael Jordan pour m'avoir accueilli et avoir financé mon séjour. Je remercie également Pierre, Agathe et Naël pour leur gentillesse tout au long de ce séjour.

Que ce soit en France ou à Berkeley, j'ai eu la chance tout au long de ces trois années d'être aidé par des assistantes administratives extrêmement efficaces et qui m'ont rendu la vie plus facile. Merci beaucoup donc à Isabelle et Nathalie à Fontainebleau, à Jennifer et Marie à Curie, et à Debbie à Berkeley.

Je remercie aussi tous mes collègues des Mines, Pierre, Caroline, Franck, Misha, Martial, Véronique, Christian, Fantine, Brice, Yoshi, Philippe, Kevin, Anne-Claire et Toby. Le CBIO, c'est bien. Je ne me risquerai pas à citer tous mes collègues de l'institut Curie parce que je serais certain d'en oublier, mais je n'en pense pas moins, je vous suis sincèrement reconnaissant d'être aussi funky (et beaux et intelligents, cela va sans dire). Je remercie tout particulièrement le système, Franck (déjà cité dans la liste des Mines mais comme la pudeur et la peur que quelqu'un lise un jour ces remerciements en entier m'interdisent d'écrire tout le bien que je pense de lui, je le cite deux fois à la place), Laurence pour m'avoir materné toutes ces années durant, Gautier même si son pays sera dissous quand Khadafi sera maître du monde, ainsi que Cécile et Pierre G. pour avoir gardé Anne quand je n'étais pas là et mon BFF Alexandre. Merci aussi aux autres étudiants avec qui j'ai eu l'occasion d'interagir, en particulier Zaïd Harchaoui, Julien Mairal, Rodolphe Jenatton, Marie Szafranski et Matthieu Kowalski.

Un grand merci à mes amis, en particulier Arthur, Guillaume, Alice, Brice, Clément, Camille, Sandrine et Fabrice (à peu près par ordre d'apparition), qui ont continué à m'appeller pour sortir alors même que je ne donnais plus signe de vie et que je dormais debout quand je ne faisais pas des blagues de maths. Un remerciement spécial à Guillaume Pinot qui a bien voulu relire ce manuscrit, et m'a appris les bases de LaTeX et d'Unix il y a quelques années de cela. Merci à ma famille, notamment à mes parents Benoit et Claire, ma sœur Gaëlle et mon frère Tom dont l'affection m'a rendu la vie vraiment plus agréable, et mes grands-parents et mon arrière-grand-mère, qui ont eu la gentillesse de toujours s'intéresser à ce que je faisais même si l'intérêt de ce que je faisais ne devait a priori pas être flagrant.

Merci enfin à Anne pour avoir su toujours me soutenir et parfois garder le moral à ma place.

Contents

List of Tables

List of Figures

Abstract

Supervised learning methods are used to build functions which accurately predict the behavior of new objects from observed data. They are therefore extremely useful in several computational biology problems, where they can exploit the increasing amount of empirical data generated by high-throughput technologies, or the accumulation of experimental knowledge in public databases.

A very popular example is DNA microarrays, which are used to measure the expression of thousands of genes, for example in tumoral cells of a patient. Typical studies involve such measures for hundreds of patients, and it is hoped that supervised learning methods using this data can help build functions which differentiate between *e.g.*, good prognosis and bad prognosis tumors based on their gene expression. In vaccine design, *in silico* methods for the prediction of antigenic peptides binding to MHC class I molecules play an increasingly important role in the identification of T-cell epitopes. Statistical and machine learning methods in particular are widely used to score candidate binders based on their similarity with known binders and non-binders. Similarly, predicting interactions between small molecules and proteins is a crucial ingredient of the drug discovery process. In particular, accurate predictive models are increasingly used to preselect potential lead compounds from large molecule databases, or to screen for side-effects.

In these three examples however, the amount of training data is rarely sufficient to deal with the complexity of the learning problem. Gene expression datasets often report measures for only few patients with respect to the number of genes. In addition, these expression measures are often noisy, and several biological effects like disease subtypes

and gene expression regulation mechanisms suggest that the best possible function based on gene expression can be very complex. Similarly, the genes coding for the MHC molecules, are highly polymorphic, and statistical methods have difficulties building models for alleles with few known binders. The same applies to drug discovery problems, where little or no training data is available for some targets of interest.

On the other hand ill-posed problems, in particular those involving less data points than dimensions, are not new in statistics and statistical machine learning. They are classically addressed using *regularization* approaches, or equivalently in a Bayesian perspective, using a *prior* on what the function should be like. We build on this principle and propose new regularization methods based on biological prior knowledge and available information for each problem.

For example, while classical *in silico* drug design approaches focus on predicting interactions with a given specific target, new chemogenomics approaches adopt cross-target views and have demonstrated the utility of leveraging information across therapeutical targets to improve the performance of the prediction. Using the prior biological knowledge that similar targets bind similar ligands, and the large amount of data available for some targets, it is therefore possible to improve dramatically the prediction accuracy for the targets with little known ligands, and even to make predictions for targets with no known ligand, provided that some ligands are known for other targets which are similar in some sense. Building on recent developments in the use of kernel methods in bio- and chemoinformatics, we present a systematic framework to screen the chemical space of small molecules for interaction with the biological space of proteins. We show that this framework allows information sharing across the targets, resulting in a dramatic improvement of ligand prediction accuracy for three important classes of drug targets: enzymes, GPCR and ion channels. Here again, the same idea applies to vaccine design.

In order to exploit more efficiently the fact that similar targets bind similar ligands, it is useful to make the idea more precise: in more realistic settings, the binding behaviors of some targets are very close to each others while some others are independent or even opposite, thereby defining clusters of related targets. Sharing information between tasks of the same clusters is expected to be beneficial while between tasks of different clusters, it may damage the performances. Since the clusters are unknown beforehand, we design a new spectral norm that encodes this restricted sharing assumption without the prior knowledge of the partition of tasks into groups. This results in a new convex optimization formulation for multi-task learning, which jointly minimizes the classification error and a relaxed clustering cost. We show in simulations on synthetic examples and on the IEDB MHC-I binding dataset, that our approach outperforms well-known convex methods for multi-task learning, as well as related non-convex methods dedicated to the same problem.

Concerning DNA microarray data, it is known that gene expressions are not independent. In particular, some groups of genes are known to be involved in the same biological functions, and tend to have very correlated expressions. Alternatively, several types of gene networks like regulatory networks or protein-protein interaction networks give informations about potential correlations between the expression of genes. To exploit this information, we devise a norm which, when used as regularization for empirical risk minimization procedures, leads to sparse estimators, the support of the sparse vector typically being a union of potentially overlapping groups of covariates defined a priori, or a set of covariates which tend to be connected to each other when a graph of covariates is given. When this penalty is used with gene sets corresponding to biological functions, or graphs encoding biological information about the correlation structure between gene expressions, it could simultaneously guide the learning process and make the solution more interpretable. We study theoretical properties of the estimator which follows from this penalty, and illustrate its behavior on simulated and breast cancer gene expression data.

Résumé

Les méthodes d'apprentissage supervisé sont utilisées pour construire des fonctions prédisant efficacement le comportement de nouvelles données à partir de données déjà observées. Elles sont de ce fait extrêmement utiles dans de nombreux problèmes de biologie computationnelle, où elles permettent d'exploiter la quantité grandissante de données expérimentales générée par les technologies à haut débit, ou l'accumulation de connaissances expérimentales contenue dans les bases de données publiques.

On peut donner en exemple les puces à ADN, qui sont utilisées pour mesurer l'expression de milliers de gènes, en particulier dans les cellules tumorales de patients. La plupart des études exploitant cette technologie effectuent ces mesures pour des centaines de patients, et on peut espérer que les méthodes d'apprentissage supervisé utilisant ces données peuvent à terme conduire à des functions permettant par exemple de différencier les tumeurs à bon et à mauvais pronostic en se basant uniquement sur l'expression de leurs gènes. Lors de la conception de vaccins, les méthodes *in silico* pour la prédiction de peptides antigéniques se liant aux molécules du MHC-I jouent un rôle de plus en plus important dans l'identifications des épitopes de lymphocytes T. En particulier, les méthodes d'apprentissage supervisé sont couramment utilisées pour noter les peptides candidats en fonction de leur similarité avec les ligands et non-ligands connus. De la même façon, la prédiction d'interactions entre des petites molécules et certaines protéines est un élément crucial dans le recherche de nouveaux médicaments. En particulier, de plus en plus de modèles prédictifs sont utilisés afin de pré-sélectionner des molécules potentiellement actives à partir de grandes bases de données, ou pour détecter des risques d'effets secondaires.

Dans chacun de ces trois exemples cependant, la quantité de données d'entraînement disponible est rarement suffisante par rapport à la complexité du problème d'apprentissage. Les études impliquant des expressions de gènes donnent généralement des mesures pour quelques centaines de patients, à rapporter au millier de gènes potentiellement impliqués. Par ailleurs, ces mesures sont généralement bruitées et de nombreux effets biologiques tels que les sous-types d'une maladie et les mécanismes de régulation de l'expression des gènes suggèrent qu'une fonction de prédiction basée sur ces valeurs peut être assez complexe. De la même manière, les gènes codant pour les molécules du MHC sont hautement polymorphes, et il peut être difficile aux méthodes statistiques de construire de bons modèles pour les allèles ayant peu de ligands connus à utiliser en entraînement. On retrouve le même problème lors de la recherche de nouveaux médicaments, où certaines cibles thérapeutiques ont peu voire aucun ligand connu.

Heureusement, les problèmes mal posés, en particulier ceux impliquant moins d'individus que de dimensions, ne sont pas nouveaux en statistiques et apprentissage automatique. Une approche classique est d'utiliser des méthodes de *régularisation*, ou de manière équivalente dans une perspective Bayesienne, d'introduire un *a priori* sur la forme que la fonction devrait avoir. Partant de ces principes, nous proposons de nouvelles méthodes de régularisation basées sur des connaissances biologiques et sur les informations a priori disponibles pour chaque problème.

Par exemple, alors que les approches classiques de conception *in silico* de médicament se concentrent sur la prédiction d'interactions avec une cible spécifique, les approches récentes dites *chémogénomiques* considèrent plusieurs cibles simultanément et ont prouvé que partager l'information entre les cibles pouvait conduire à de meilleures prédictions. L'utilisation de cette connaissance *a priori* que des cibles similaires lient des ligands similaires et de la grande quantité de données disponibles pour certaines cibles permet ainsi d'améliorer grandement les prédictions pour certaines cibles pour lesquelles peu de ligands sont connus. Cette approche peut même conduire à de bonnes prédictions pour certaines cibles sans ligand connu, sous réserve que des ligands soient disponibles pour d'autres cibles, similaires à la cible d'intérêt. En nous basant sur les récents développements de noyaux en bio- et chémoinformatique, nous proposons un cadre systématique pour cribler l'espace des molécules contre l'espace biologique des protéines, afin de détecter de potentielles interactions. Nous montrons que ce cadre permet un partage de l'information entre les cibles et conduit à une forte amélioration de la précision en prédiction de ligands pour trois classes importantes de cibles thérapeutiques : les enzymes, les GPCR et les canaux ioniques. Ici encore, la même idée s'applique également à la conception de vaccins.

Afin d'exploiter plus efficacement le fait que les cibles similaires lient des ligands similaires, il convient ensuite de préciser cette idée : dans un cadre plus réaliste, certaines cibles ont une comportement de liaison très proche tandis que d'autres ont des comportements indépendants voire opposés, ce qui définit des groupes de cibles au même comportement. On peut s'attendre à ce que le partage d'information entre des cibles du même groupe soit bénéfique à l'apprentissage, alors qu'au contraire partager l'information entre des cibles de groupes différents risque d'altérer les performances. Les groupes étant inconnus *a priori*, nous avons conçu une nouvelle norme spectrale qui représente cette hypothèse de partage restreint sans nécessiter la connaissance de l'information de partition des cibles. Il en résulte une nouvelle formulation d'optimisation convexe pour l'apprentissage multitâche, qui minimise conjointement l'erreur de classification et une relaxation du coût de clustering. Nous montrons par des simulations sur des exemples synthétiques ainsi que sur les données de liaison MHC-I de la base de données IEDB que cette approche donne de meilleures performances que les autres méthodes convexes pour l'apprentissage multitâche, ainsi que les méthodes non-convexes appliquant la même idée.

En ce qui concerne les puces à ADN, les connaissances en génétique moderne suggèrent que les expressions des gènes ne sont pas indépendantes. En particulier, certains groupes de gènes sont connus pour être impliqués dans les mêmes fonctions biologiques, et ont ainsi tendance à avoir des expressions corrélées. Par ailleurs, certains types de réseaux de gènes tels que les réseaux de régulation ou les réseaux d'interaction de protéines fournissent des informations quant à la corrélation potentielles entre l'expression des gènes. Afin d'exploiter ces informations, nous construisons une norme qui, lorsqu'elle est utilisée pour régulariser un problème de minimisation du risque empirique, conduit à des estimateurs parcimonieux, dont le support est typiquement une union de groupes de variables potentiellement chevauchants définis *a priori*, ou un ensemble de variables ayant tendance à être connectées sur un graphe également défini *a priori*. Lorsque cette pénalité est utilisée avec des ensembles de gènes correspondant à des fonctions biologiques ou avec des graphes représentant une information biologique sur les structures de corrélation entre les expressions de gènes, elle pourrait simultanément guider le processus d'apprentissage et rendre la solution plus interprétable. Nous étudions les propriétés théoriques de l'estimateur résultant de cette pénalité, et illustrons son comportement sur des données simulées ainsi que sur des données d'expression de gènes dans des tumeurs du sein.

Chapter 1

Context

This preliminary chapter introduces the main concepts and existing work this thesis builds on. We start by a very general introduction to statistical machine learning, followed by a section on its applications to computational biology, with an emphasis on vaccine, drug design and outcome prediction from gene expression data, which are the biological problems we tackle throughout this thesis. Then, we briefly introduce kernel methods, along with a presentation of the existing kernels for proteins and for small molecules. Finally, we present the notion of regularization and prior knowledge for statistical machine learning, with a focus on multi-task learning and sparsity-based regularization.

1.1 Statistical machine learning

1.1.1 General definition

Machine learning is concerned with analyzing data arising from some phenomenon. The objective can be purely descriptive, *e.g.*, finding a good way to summarize the phenomenon or isolating interesting trends. It can also be inferential, *i.e.*, the goal can be to learn the relation between the observed data and another phenomenon in order to accurately predict the phenomenon from new data for which it is not observed. The methods addressing the former problems are often refered to as *unsupervised* learning, whereas the methods addressing the latter are refered to as *supervised* learning.

In this thesis, this data will be generally described by observations $x_i \in \mathcal{X}$, denoted $\{x_1, \ldots, x_n\}$ for n observations. Unless otherwise stated, these observations will be vectors of dimension d, *i.e.*, $\mathcal{X} = \mathbb{R}^d$. $X \in \mathbb{R}^{n \times d}$ will denote the matrix whose lines are the observations.

The remainder of this section introduces the basic notions which underly this thesis. For a more detailed overview, the reader is referred to, *e.g.*, Vapnik (1998) and Hastie et al. (2001).

1.1.2 Supervised learning

In supervised learning, an output $y_i \in \mathcal{Y}$ is associated to each corresponding observation x_i for $i = 1, \ldots, n$. The set of observed data-output pairs $\{(x_i, y_i)_{i=1,\ldots,n}\}$ is referred to as the *training set*.

The goal is to learn a function from this training set, which accurately predicts the output y of a new input x. More formally, considering that:

- The observations and the corresponding outputs follow a joint distribution $\mathbb{P}(x, y)$.
- We are given a *loss* function
$$L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R},$$
such that $L(y, y')$ quantifies the cost of predicting the output y' when the right output is y.

Then supervised learning aims at finding the function f which minimizes

$$R(f) = \int_{\mathcal{X} \times \mathcal{Y}} L(y, f(x)) d\mathbb{P}, \tag{1.1}$$

where R is the *risk* of f, *i.e.*, the mean cost of using it to predict y from x on the joint distribution.

Of course, this risk cannot be computed in practice because the joint distribution $\mathbb{P}$ is unknown. Practical algorithms therefore have to use the training set to estimate it, for example by the *empirical risk* :

$$\hat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)). \tag{1.2}$$

Choosing the f which minimizes $\hat{R}_n$ is a procedure known as *empirical risk minimization*. However, for a finite sample size n, the function which minimizes $\hat{R}_n$ may not be unique, and some of the minimizers may have a very high risk R. For example in a regression problem where $\mathcal{Y} = \mathbb{R}$, any function f taking the correct values at the training points, *i.e.*, $f(x_i) = y_i, i = 1, \ldots, n$ and any value everywhere else will have a zero empirical risk for any reasonable loss function (*e.g.* the squared loss $(y_i - f(x_i))^2$). The theory of statistical learning (Vapnik, 1995, 1998) gives bounds on the distance between $\hat{R}_n(f)$ and $R(f)$ as a function of the sample size n and the complexity of the class of functions $\mathcal{H}$ on which the empirical risk is minimized. The key to obtain a function having a low true risk and therefore better generalization abilities on (x, y) pairs which were not in the training set is to control the complexity of $\mathcal{H}$. Indeed when looking for the best classifier f in a given space of functions $\mathcal{H}$, the Bayes regret $R(f) - R^*$, where R^* is the risk of the Bayes rule which is the optimal classifier knowing the true distribution, can classically be decomposed as :

$$R(f) - R^* = \left(R(f) - \inf_{g\in\mathcal{H}} R(g)\right) + \left(\inf_{g\in\mathcal{H}} R(g) - R^*\right). \tag{1.3}$$

The second term is called the *approximation error* and corresponds to the minimum excess risk which can be achieved by using a member of $\mathcal{H}$. It is a bias term, which does not depend on the data but only on the richness of $\mathcal{H}$, or at least on its ability to approximate the Bayes rule. The first term is called the *estimation error*, and corresponds to the excess risk of f with respect to the best possible function in $\mathcal{H}$.

Interestingly, when choosing f by empirical risk minimization over $\mathcal{H}$, it can be shown that if f_n^* minimizes $\hat{R}_n$, then :

$$R(f_n^*) - \inf_{f\in\mathcal{H}} R(f) \leq 2\sup_{f\in\mathcal{H}} |R_n(f) - R(f)|,$$

so the estimation error can be thought of like a variance term, which grows with the size, or more precisely the complexity of $\mathcal{H}$.

The variance term in the Bayes regret decomposition (1.3) penalizes complex function spaces, while the bias term penalizes spaces which do not contain good approximations of the true function. In order to obtain a low risk when minimizing the empirical risk, it is therefore necessary to minimize it over function spaces which are not too complex, but are rich enough to contain good approximations of the true function.

This bias-variance trade-off is generally dealt with using the following *structural risk minimization* procedure (Vapnik and Chervonenkis, 1974) :

1. Define a structured family of function complexity classes,
2. Find the empirical risk minimizer on each complexity class,
3. Choose the minimizer giving the best generalization performances.

In practice, step 3 is done by estimating the empirical risk of the selected function on a hold out part of the training set which was not used in the two previous steps. Step 1 is typically done by varying the strength of a penalty constraining the optimal function : each level of constraint defines a new function space in which the empirical risk minimization can be performed, with the hope that one of them is simple enough to have a small variance term and rich enough to have a small bias term. This procedure underlines the importance of designing good penalties : since it defines the sequence of function spaces in which empirical risk minimization is performed, an excellent penalty may give function spaces which are simple yet contain good approximations of the true function. In particular, this may happen if the penalty enforces some *a priori* about what type of regularity the true function should have. This thesis presents some of these priors, and the corresponding penalties.

1.1.3 Unsupervised learning

Another family of methods, often termed *unsupervised learning* aim at analyzing the data based on the descriptions $\{x_1, \dots, x_n\}$ only. Among these, *clustering* methods attempt to identify groups of observations which are significantly close to each others in the description space which can be a first step for preprocessing, compressing, or can reveal a hidden phenomenon. A very popular example of such a clustering algorithm is k-means (Hartigan and Wong, 1979), which iteratively estimates the cluster centers from the cluster assignments by taking the mean over the observations in each cluster, and the assignments from the centers by associating each observation with the closest cluster center. Each step of this fast heuristic decreases the within sum of squares criterion :

$$\sum_{c=1}^{C} \sum_{i \in \mathcal{I}_c} \|x_i - \bar{x}_c\|_2^2, \tag{1.4}$$

where C is the number of clusters, $\mathcal{I}_c$ the indices of the observations belonging to cluster c and $\bar{x}_c$ the center of cluster c. More recently, approaches based on the spectral decomposition of the Gram matrix $XX^\top$ have been shown to give good results (Ng et al., 2001; Bach and Jordan, 2003; von Luxburg, 2007).

Other unsupervised methods try to identify trends or axes which characterize the set of observations. For example, *principal component analysis* (PCA) (Pearson, 1901) gives the directions which best explain the variance of the data while *independent component analysis* (ICA) (Comon, 1994) identifies axes which are as independent as possible, for a given independency criterion. Non-parametric version of these methods using positive definite kernels (which will be introduced in Section 1.3.1) were proposed in Schölkopf et al. (1999) and Bach and Jordan (2002) respectively.

Some approaches formally combine unsupervised and supervised learning, *e.g.* for classifier fusion or combination (Kuncheva, 2004) which involves the identification of clusters in the description space followed by the identification of the locally optimal classifier on each cluster from a family of clusters.

1.1.4 ℓ_p norms

As they will be used to build several penalties which are presented in this thesis, it is useful to recall the definition of the ℓ_p norms. The ℓ_p norm of a vector $x \in \mathbb{R}^d$ is defined by :

$$\|x\|_p = \left(\sum_{j=1}^{d} |x_j|^p \right)^{\frac{1}{p}}. \tag{1.5}$$

$\|.\|_p$ is a valid norm for $1 \leq p \leq \infty$. The ℓ_2 norm is the usual euclidean norm. Taking the limit $p \to \infty$ yields the max norm $\|x\|_\infty = \max_{j=1,\dots,d} \|x_j\|$. The last four images of Figure 1.1 show the unit balls of $\|.\|_p$ for $p = 1, \frac{3}{2}, 2, \infty$, *i.e.*, the set $\{w \in \mathbb{R}^2, \|w\|_p \leq 1\}$.

For $0 < p < 1$ on the other hand, $\|.\|_p$ is not a norm anymore because it does not verify the triangle inequality, but it is still a quasi-norm. Taking $p \to 0$ with the convention $0^0 = 0$ gives :

$$\|x\|_0 = \sum_{j=1}^{d} \mathbf{1}_{\{x_j \neq 0\}}, \tag{1.6}$$

often referred to as the ℓ_0 norm although it is not a proper norm because it is not positive homogeneous. The first two images of Figure 1.1 show the unit balls of $\|.\|_p$ for $p = 0, \frac{2}{3}$.

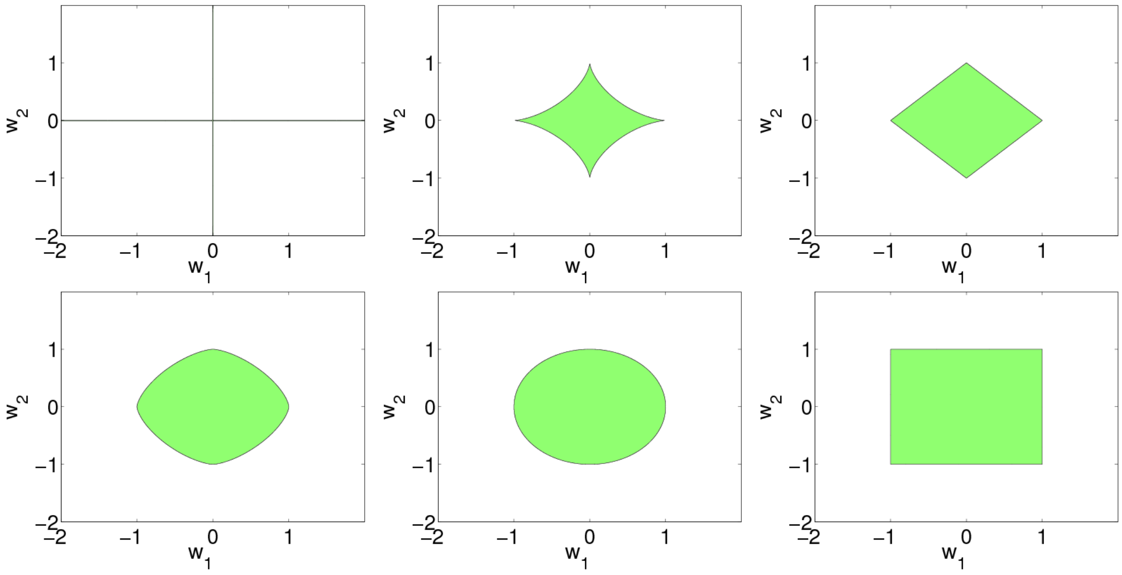

Figure 1.1: Unit balls for the ℓ_p norms in two dimensions, for $p = 0, \frac{2}{3}, 1, \frac{3}{2}, 2, \infty$.

1.2 Supervised learning in computational biology

1.2.1 Overview

The growing quantity of data available to analyze various problems in biology have made possible and sometimes necessary the use of statistical tools like the ones we presented in

Section 1.1. A typical example is the case of microarray data, which we further detail in section 1.2.4 : while technology only allowed to study the expression of one or few genes in one or few patients, there was no point in detecting statistical trends in the data. When microarrays allowed to measure the expression level of thousand of genes for hundreds of patients, the data became impossible to analyze by just looking at it, but it became relevant to look at clusters of genes which had the same behavior across the patients, or to try to identify which genes best explained a given variable on the patients in average.

For example, supervised learning methods were used to study the evolutionary relatedness of biological sequences (Felsenstein, 1981; Carlo et al., 1999; Rohlf, 2005; Bouchard-Côté et al., 2008) in *phylogenetics* , to find recurrent motifs in a set of sequence (Bailey and Elkan, 1994; Roth et al., 1998; Xing et al., 2004; Xing and Karp, 2004; Frith et al., 2008; Fu et al., 2009), and to predict the family (Bejerano and Yona, 1999; Bhasin and Raghava, 2004a; Cai et al., 2004; Leslie et al., 2004; Qiu et al., 2007) or the structure of a new protein (Kumar et al., 2005; Zhang, 2008).

The contributions of this thesis are focused on two classes of computational biology problems. The first one is to predict pairwise interactions in a biological system, which can also be thought of as infering missing edges in a particular biological networks (Vert and Yamanishi, 2005). While this framework encompasses the completion of metabolic (Yamanishi et al., 2005; Bleakley et al., 2007; Vert et al., 2007), protein-protein interaction (Ben-Hur and Noble, 2005; Bleakley et al., 2007; Vert et al., 2007) and regulatory networks (Qian et al., 2003; Auliac et al., 2008; Mordelet and Vert, 2008), we focused on methods for *vaccine design*, where one wants to predict interaction between small pathogen fragments and MHC molecules, and *drug design*, where one wants to predict interaction between small molecules and proteins. The second problem is to make good diagnosis or prognosis based on molecular data. In particular, we study the problem of predicting tumor metastasis based on gene expression data. The remainder of this section gives a more detailed presentation of these two problems.

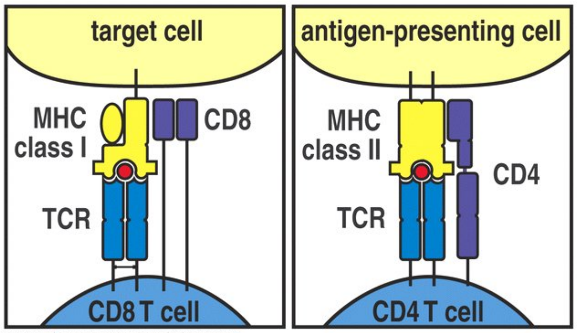

Figure 3-12 The Immune System, 2/e (© Garland Science 2005)

Figure 1.2: T cell – HLA molecule interaction

1.2.2 Vaccine design

The immune system and cytotoxic T Lymphocytes

The *immune system* is the set of mechanisms that protect our organism against all kinds of infectious agents. These mechanisms are organized in two main branches: innate or non-specific and adaptive or specific immune system.

The idea of a vaccine is to artificially trigger active immunity to a disease, so we focus our interest on specific immune system, *i.e.*, lymphocytes and specific antibodies. Since experiments in this thesis focused on HIV vaccines, for which the most efficient mechanism seems to be cytotoxic reactions (McMichael and Hanke, 2002; Parham, 2004), we will be even more specific and mostly describe this last mechanism.

T-cells are special lymphocytes that play a major role in the cell-mediated response, by contrast with the humoral immunity ruled by the antibodies. They all express the *T-Cell Receptor* (TCR). Cytotoxic T-cells are involved in the destruction of virally infected cells. Since most of them express the CD8 glycoprotein, they are also known as CD8+ T-cells.

A key step in the cell-mediated immune response is the *activation* of the T-cells through the interaction of the TCR with a specific MHC-antigen complex. This is illustrated on Figure 1.2. Basically the T-cell “recognizes” an antigen, which is a peptide, *i.e.*, a fragment of protein, that is presented by a cell. Since the only viable T-cells are those who do not recognize the organism-specific peptides, the recognition means that the presenting cell is either not from the organism, like a bacteria, or has been infected by a virus and presents its proteins.

The naive T-cells that recognize an antigen both divide and mature into effector cells. Activated cytotoxic T lymphocytes (CTL) are then able to kill specifically the infected cells they recognize. This is done through the release of effector proteins such as perforin and granzymes, or via the binding of Fas in the target cell membrane by the Fas ligand that leads to activation of caspases. All these molecules induce apoptosis in the target cells. The killing process is illustrated on Figure 1.3.

The MHC-epitope binding

As we explained above, both the activation of the T-cells and their action imply the recognition of a specific MHC-antigen complex. We now describe this complex more precisely.

The MHC is a large gene family involving around 140 genes subdivided into three groups or classes. CD8+ T-cells recognize antigen bound with class I MHC molecules[1]. These molecules are heterodimers, consisting of a single transmembrane polypeptide chain (the α-chain) and a β_2 microglobulin (which is encoded elsewhere, not in the MHC). The schematic representation on Figure 1.4 shows the peptide-binding groove formed by the two polymorphic domains α_1, α_2. This part of the molecule, whose shape depends on the corresponding MHC genes allele presents an antigen to the TCR of the T-cells. The binding mechanism is shown on Figure 1.5.

As one can see on Figure 1.6, the shape of the epitope must be compatible with the shape of the groove. In other words, the potential epitopes can be different if the molecules are different, which is likely to occur if the corresponding MHC gene alleles are different.

The problem is that the MHC harbors much allelic diversity, *i.e.*, one finds many different genotypes for the MHC genes. This implies different phenotypes, which means that one finds a large variety of MHC presenting molecules, each of them being able to complex with different peptides because of its different structure.

Interaction prediction for intelligent vaccine design

To summarize, a key step in the immune response to pathogen invasion is the activation of cytotoxic T-cells, which is triggered by the recognition of a short peptide, called epitope,

[1] In humans, the subset of the MHC genes that code for presenting molecules is also known as HLA for *human leukocyte antigen*.

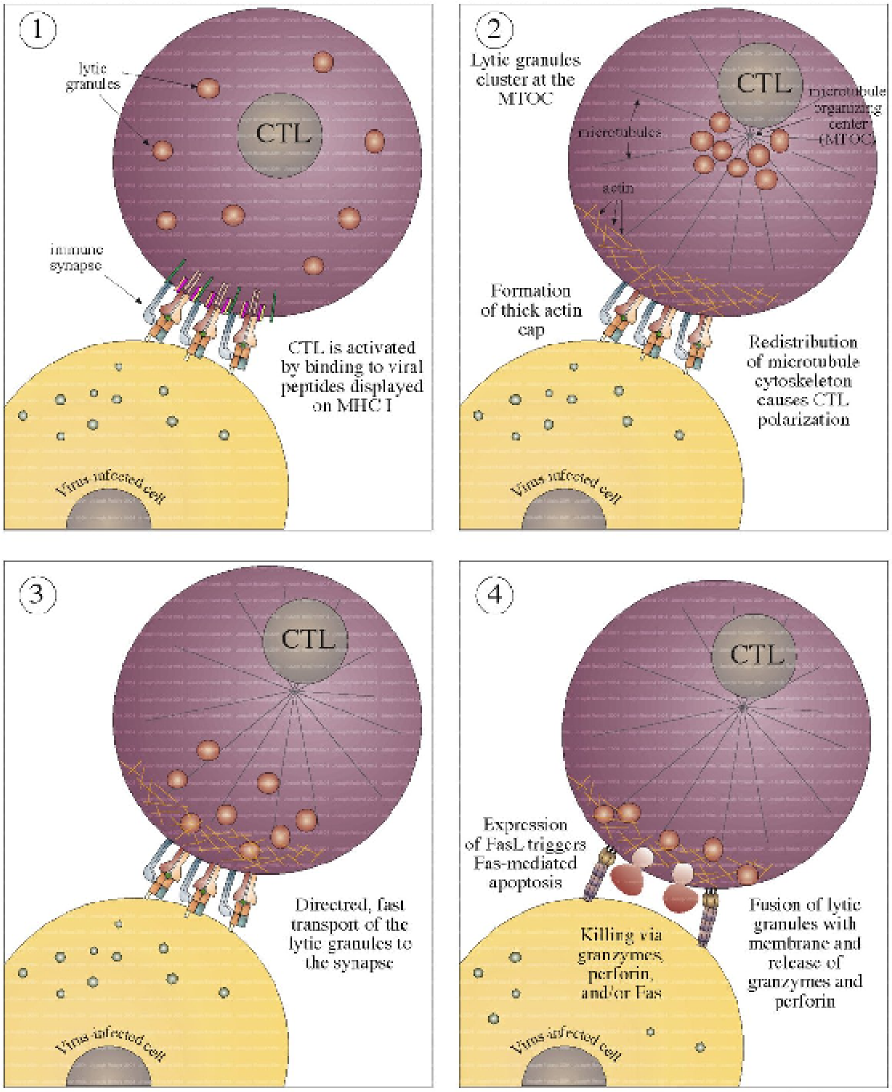
1
lytic granules
CTL
immune synapse
CTL is activated by binding to viral peptides displayed on MHC I
Virus infected cell
2
Lytic granules cluster at the MTOC
CTL
microtubule organizing center (MTOC)
microtubules
actin
Formation of thick actin cap
Redistribution of microtubule cytoskeleton causes CTL polarization
Virus infected cell
3
CTL
Directed, fast transport of the lytic granules to the synapse
Virus infected cell
4
CTL
Expression of FasL triggers Fas-mediated apoptosis
Killing via granzymes, perforin, and/or Fas
Fusion of lytic granules with membrane and release of granzymes and perforin
Virus infected cell

Figure 1.3: Killing by Cytotoxic T Lymphocyte

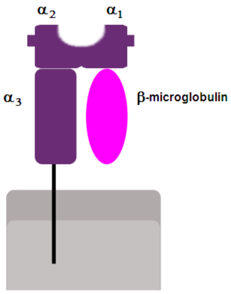

Figure 1.4: Schematic representation of MHC class I molecule (source: Wikipedia).

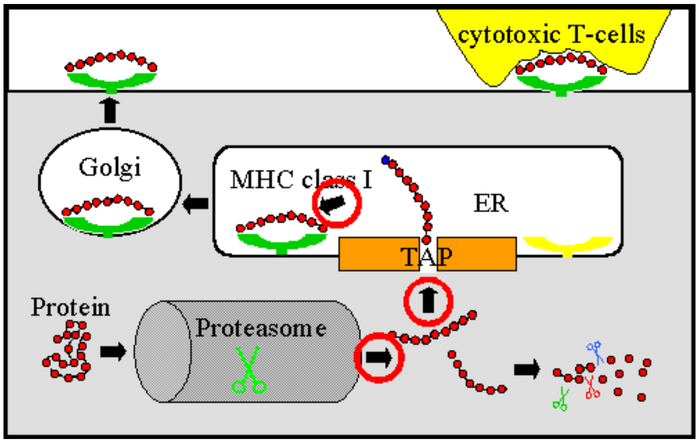

Figure 1.5: Construction of the MHC-antigen complex

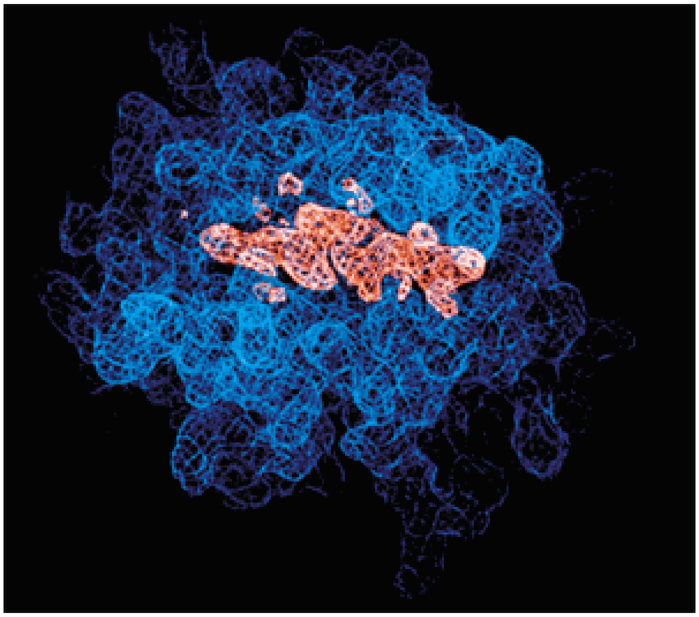

Figure 1.6: Epitope presented in the groove of a MHC class I molecule

bound to Major Histocompatibility Complex (MHC) class I molecules and presented to the T-cells. This recognition is supposed to trigger cloning and activation of cytotoxic lymphocytes able to identify and destroy the pathogen or infected cells. MHC class I epitopes are therefore potential tools for the development of peptide vaccines, in particular for AIDS vaccines (McMichael and Hanke, 2002). They are also potential tools for diagnosis and treatment of cancer (Wang, 1999; Sette et al., 2001).

Identifying MHC class I epitope in a pathogen genome is therefore crucial for vaccine design. However, not all peptides of a pathogen can bind to the MHC molecule to be presented to T-cells: it is estimated that only 1 in 100 or 200 peptides actually binds to a particular MHC (Yewdell and Bennink, 1999). In order to alleviate the cost and time required to identify epitopes experimentally, *in silico* computational methods for epitope prediction are therefore increasingly used. Structural approaches, on the one hand, try to evaluate how well a candidate epitope fit in the binding groove of a MHC molecule, by various threading or docking approaches (Rosenfeld et al., 1995; Schueler-Furman et al., 2000; Tong et al., 2006; Bui et al., 2006). Sequence-based approaches, on the other hand, estimate predictive models for epitopes by analyzing and learning from sets of known epitopes and non-epitopes. Models can be based on motifs (Rötzschke et al., 1992; Rammensee et al., 1995), profiles (Parker et al., 1994; Rammensee et al., 1999; Reche et al., 2002), or machine learning methods like artificial neural networks (Honeyman et al., 1998; Milik et al., 1998; Brusic et al., 2002; Buus et al., 2003; Nielsen et al., 2003; Zhang et al., 2005), hidden Markov models (Mamitsuka, 1998), support vector machines (SVM) (Dönnes and Elofsson, 2002; Zhao et al., 2003; Bhasin and Raghava, 2004b; Salomon and Flower, 2006), boosted metric learning (Hertz and Yanover, 2006) or logistic regression (Heckerman et al., 2007). Finally, some authors have recently proposed to combine structural and sequence-based approaches (Antes et al., 2006; Jojic et al., 2006). Although comparison is difficult, sequence-based approaches that learn a model from the analysis of known epitopes benefit from the accumulation of experimentally validated epitopes and will certainly continue to improve as more data become available.

The binding affinity of a peptide depends on the MHC molecule's 3D structure and physicochemical properties, which in turns vary between MHC alleles. This compels any prediction method to be allele-specific: indeed, the fact that a peptide can bind to an allele is neither sufficient nor necessary for it to bind to another allele. Since MHC genes are highly polymorphic, little training data if any is available for some alleles. Thus, though achieving good precisions in general, classical statistical and machine learning-based MHC-peptide binding prediction methods fail to efficiently predict bindings for these alleles.

Some alleles, however, can share binding properties. In particular, experimental work (Sidney et al., 1995, 1996; Sette and Sidney, 1998, 1999) shows that different alleles have overlapping peptide repertoires. This fact, together with the posterior observation of structural similarities among the alleles sharing their repertoires allowed the definition of HLA allele supertypes, which are families of alleles exhibiting the same behavior in terms of peptide binding. This suggests that sharing information about known epitopes across different but similar alleles has the potential to improve predictive models by increasing the quantity of data used to establish the model. For example, Zhu et al. (2006) show that simply pooling together known epitopes for different alleles of a given supertype to train a model can improve the accuracy of the model. Hertz and Yanover (2006) pool together epitope data for all alleles simultaneously to learn a metric between peptides, which is then used to build predictive models for each allele. Finally, Heckerman et al. (2007) show that leveraging the information across MHC alleles and supertypes considerably improves individual allele prediction accuracy. This idea was extended to MHC-II binding prediction in Pfeifer and Kohlbacher (2008), which is a harder problem since the MHC-II binding clefts are not closed, using multiple instance learning (Dietterich et al., 1997).

1.2.3 Virtual screening for drug discovery

Interaction prediction for drug design in general

Predicting interactions between small molecules and proteins is a key element in the drug discovery process. In particular, several classes of proteins such as G-protein-coupled receptors (GPCR), enzymes and ion channels represent a large fraction of current drug targets and important targets for new drug development (Hopkins and Groom, 2002). Understanding and predicting the interactions between small molecules and such proteins could therefore help in the discovery of new lead compounds.

Various approaches have already been developed and have proved very useful to address this *in silico* prediction issue (Manly et al., 2001). The classical paradigm is to predict the modulators of a given target, considering each target independently from other proteins. Usual methods are classified into *ligand-based* and *structure-based* or *docking* approaches. Ligand-based approaches compare a candidate ligand to the known ligands of the target to make their prediction, typically using machine learning algorithms (Butina et al., 2002; Byvatov et al., 2003) whereas structure-based approaches use the 3D-structure of the target to determine how well each candidate binds the target (Halperin et al., 2002; Kellenberger et al., 2004).

Ligand-based approaches require the knowledge of sufficient ligands of a given target with respect to the complexity of the ligand/non-ligand separation to produce accurate predictors. If few or no ligands are known for a target, one is compelled to use docking approaches, which in turn require the 3D structure of the target and are very time consuming. If for a given target with unavailable 3D structure no ligand is known, none of the classical approaches can be applied. This is the case for many GPCR as very few structures have been crystallized so far, see Ballesteros and Palczewski (2001) for the first structure, and Weis and Kobilka (2008); Mustafi and Palczewski (2009) and Topiol and Sabio (2009) for more recent reviews. In addition, many of these receptors, referred to as *orphan* GPCR, have no known ligand.

An interesting idea to overcome this issue is to stop considering each protein target independently from other proteins, and rather take the point of view of *chemogenomics* (Kubinyi et al., 2004; Jaroch and Weinmann, 2006). Roughly speaking, chemogenomics aims at mining the entire *chemical space*, which corresponds to the set of all small molecules, for interactions with the *biological space*, i.e., the set of all proteins or at least protein families, in particular drug targets. A salient motivation of the chemogenomics approach is the realization that some classes of molecules can bind "similar" proteins, suggesting that the knowledge of some ligands for a target can be helpful to determine ligands for similar targets. Besides, this type of method allows for a more rational approach to design drugs since controlling a whole ligand's selectivity profile is crucial to make sure that no side effect occurs and that the compound is compatible with therapeutical usage.

Recent reviews (Kubinyi et al., 2004; Jaroch and Weinmann, 2006; Klabunde, 2007; Rognan, 2007) describe several chemogenomic approaches to predict interactions between compounds and targets. A first class of approaches, called *ligand-based chemogenomics* by Rognan (2007), pool together targets at the level of families (such as GPCR) or subfamilies (such as purinergic GPCR) and learn a model for ligands at the level of the family (Balakin et al., 2002; Klabunde, 2006). Other approaches, termed *target-based chemogenomic* approaches by Rognan (2007), cluster receptors based on ligand binding site similarity and again pool together known ligands for each cluster to infer shared ligands (Frimurer et al., 2005). Finally, a third strategy termed *target-ligand* approach by Rognan (2007) attempts to predict ligands for a given target by leveraging binding information for other targets in a single step, that is, without first attempting to define a particular set of similar receptors. For example, Bock and Gough (2005) merge descriptors of ligands and targets to describe putative ligand-receptor complexes, and use machine learning methods to discriminate real complexes from ligand-receptors pairs that do not form complexes This idea was further developed in Weill and Rognan (2009). Erhan et al. (2006) show how the same idea can be casted in the framework of neural networks and support vector machines (SVM), In particular they show that a given set of receptor descriptors can be combined with a given set of ligand descriptors in a computationally efficient framework, offering in principle a large flexibility in the choice of the receptor and ligand descriptors.

Interaction prediction for drug design: the GPCR case

The G-protein coupled receptor (GPCR) superfamily is comprised of an estimated 600-1,000 members and is the largest known class of molecular targets with proven therapeutic value. They are ubiquitous in our body, being involved in regulation of every major mammalian physiological system (Bockaert and Pin, 1999), and play a role in a wide range of disorders including allergies, cardiovascular dysfunction, depression, obesity, cancer, pain, diabetes, and a variety of central nervous system disorders (Deshpande and Penn, 2006; Hill, 2006; Catapano and Manji, 2007). They are integral membrane proteins sharing a common global topology that consists of seven transmembrane alpha helices, an intracellular C-terminal, an extracellular N-terminal, three intracellular loops and three extracellular loops. There are four main classes of GPCRs (A, B, C and D) defined in terms of sequence similarity (Horn et al., 2003). Their location on the cell surface makes them readily accessible to drugs, and 30 GPCRs are the targets for the majority of best-selling drugs, representing about 40% of all prescription pharmaceuticals on the market (Fredholm et al., 2007). Besides, the human genome contains several hundreds unique GPCRs which have yet to be assigned a clear cellular function, suggesting that they are likely to remain an important target class for new drugs in the future (Lin and Civelli, 2004).

Predicting interactions *in silico* between small molecules and GPCRs is not only of particular interest for the drug industry, but also a useful step for the elucidation of many biological process. First, it may help to decipher the function of so-called *orphan* GPCRs, for which no natural ligand is known. Second, once a particular GPCR is selected as a target, it may help in the selection of promising molecule candidates to be screened *in vitro* against the target for lead identification.

In silico virtual screening of GPCRs is however a daunting task, both for receptor-based approaches (also called docking) and for ligand-based approaches. The former relies on the prior knowledge of the 3D structure of the protein, in a context where only two GPCR structures are currently known (bovine rhodopsin and human β_2-adrenergic receptor). Indeed, GPCRs, like other membrane proteins, are notoriously difficult to crystallize. As a result, docking strategies for screening small molecules against GPCRs are often limited by the difficulty to model correctly the 3D structure of the target. To circumvent the lack of experimental structures, various studies have used 3D structural models of GPCRs built by homology modeling using bovine rhodopsin as a template structure. Docking a library of molecules into these modeled structures allowed the recovery of known ligands (Evers and Klabunde, 2005; Cavasotto et al., 2003; Shacham et al., 2004; Bissantz et al., 2003), and even identification of new ligands (Becker et al., 2004; Cavasotto et al., 2008). However, docking methods still suffer from docking and scoring inaccuracies, and homology models are not always reliable-enough to be employed in target-based virtual screening. Methods have been proposed to enhance the quality of the models for docking studies by global optimization and flexible docking (Cavasotto et al., 2003), or by using different sets of receptor models (Bissantz et al., 2003). Nevertheless, these methods have been applied only to class A receptors and they are expected to show limited performances for GPCRs sharing lower sequence similarity with rhodopsin, especially in the case of receptors belonging to classes B, C and D. Alternatively, ligand-based strategies, in particular quantitative structure-activity relationship (QSAR), attempt to predict new ligands from previously known ligands, often using statistical or machine learning approaches. Ligand-based approaches are interesting because they do not require the knowledge of the target 3D structure and can benefit from the discovery of new ligands. However, their accuracy is fundamentally limited by the amount of known ligands, and degrades when few ligands are known. Although these methods were successfully used to retrieve strong GPCR binders (Rolland et al., 2005), they are efficient for lead optimization within a previously identified molecular scaffold, but are not appropriate to identify new families of ligands for a target. At the extreme, they cannot be pursued for the screening of orphan GPCRs. In this thesis, we will present a contribution to the screening of GPCRs, that is complementary to the above docking and ligand-based approaches. The method is related to ligand-based approaches, but because it allows to share information between different GPCRs, it can be used for orphan GPCRs, possibly in parallel to docking methods in order to increase the prediction quality.

1.2.4 Outcome prediction from molecular data

Another important application field of supervised learning in computational biology is the exploitation of molecular data, either to understand biological mechanisms like gene regulation in an organism, or to analyze and predict phenomenons influenced by such mechanisms. An example of particular interest is the prediction of outcome, *e.g.* diagnostic or prognosis of a disease, based on gene expression or copy number data. Using this type of information is crucial for diseases like cancers, which are known to be strongly related to high genomic instability.

DNA microarrays

DNA microarrays are used to measure the expression level of several genes in a tissue simultaneously. Several technologies exist, but the general idea is to use DNA segments fixed on a solid array, each segment being located at a known place on the array and used as a probe for a given gene. Typically, each segment is a single strand of DNA containing a subsequence of the gene. RNA is then extracted from the studied cells, converted to dyed cDNA by reverse-transcription and *hybridized* on the array. By complementarity, if a piece of cDNA strand contains a gene, it will be hybridized at the location of the corresponding probe. Two-channel technologies mix the reverse-transcripted RNA of two different sources (*e.g.*, a control tissue and a tested tissue) using a different fluorochrome to dye each of them, whereas one-channel technologies directly measure the absolute expression of the genes in a single tissue. The dye intensity of each spot is then measured, reflecting the expression level of the corresponding gene. Figure 1.7 shows an example of such an hybridized array. For a general presentation of DNA microarrays, see Brown and Botstein (2000).

A large effort has been made to normalize and correct various biases of the raw data generated by these technologies (Yang et al., 2002; Benito et al., 2004). In terms of applications, it has been used for a wide range of problems, including exploration of biological mechanisms (DeRisi et al., 1997; Ferea et al., 1999; Gasch et al., 2001; Le Roch et al., 2004), gene network inference (Beal et al., 2005; Bansal et al., 2007), pathway analysis (Curtis et al., 2005) or drug discovery (Debouck and Goodfellow, 1999). A very

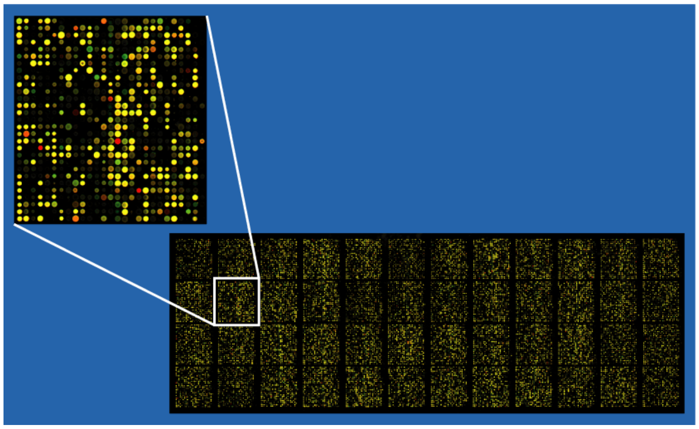

Figure 1.7: Example of an approximately 37,500 probe spotted oligo microarray with enlarged inset to show detail (source: Wikipedia).

important application is the study of cancerous cells. Early work (Golub et al., 1999; Alizadeh et al., 2000) showed that expression data could be used to efficiently differentiate distinct tumour types, and give insight on the genes which were involved in the disease. More generaly, a large effort has been made to propose efficient methods for tumor classification (Mukherjee et al., 1998; Ben-Dor et al., 2000; Bhattacharjee et al., 2001; van 't Veer et al., 2002; Rapaport et al., 2007), to characterize them (Perou et al., 2000) or extract molecular signatures (van de Vijver et al., 2002; Bild et al., 2006) based on gene expression data. In this thesis, we contribute to this effort by proposing methods which simultaneously identify good metastasis signatures in terms of pre-defined biologically meaningful gene groups and build a metastasis prediction function based on this signature, with the hope to improve model interpretability and robustness.

Other technologies quantifying molecules

In this thesis, outcome prediction and biomarker discovery experiments were only based on gene expression data. However, several other technologies allow to measure other quantities of interest. For example, *array comparative genomic hybridization* (aCGH) measure the number of copies of each gene in a cell (Pinkel et al., 1998; van Beers and Nederlof, 2006; Chin et al., 2006, 2007). In a normal diploid cell, the expected number of copy is 2, but phenomenons like microdeletions or duplications can decrease or increase it. Since these phenomenons have an influence on the outcome (*e.g.* in the case of a tumor suppressor deletion), their measure is important both for prediction and analysis purposes. In particular, several methods have been proposed to detect breakpoints on the CGH profiles (Jong et al., 2004; Hupé et al., 2004), classify patients based on the copy number measure (Aliferis et al., 2002), or do both simultaneously (Rapaport et al., 2008; Tibshirani and Wang, 2008). Other relevant data include the quantity of each protein, measured by mass spectrometry-based proteomics methods (Wilkins et al., 1996; Aebersold and Mann, 2003; Dhingra et al., 2005), and epigenetic data such as DNA methylations, which are known to influence gene expression (Jones, 2002; Jaenisch and Bird, 2003). Besides, recent high-throughput technologies allow to measure at a better base-scale resolution the number of DNA (Mardis, 2008) or RNA (Mortazavi et al., 2008; Wang et al., 2009) sequences.

1.3 Kernel methods in computational biology

Among the many approaches in machine learning that have been investigated to attack problems arising in computational biology, kernel methods have emerged as a powerful and principled tool to manipulate data such as molecules or proteins whose explicit description by a real-valued vector is sometimes delicate. In this section largely inspired from Vert and Jacob (2008), we introduce the notions of positive definite kernel and kernel methods, and give an overview of existing kernels for small molecules and kernels for proteins. These notions will be central in the methods introduced in Chapter 2. For a more detailed presentation, the reader is referred to Saitoh (1988) for positive definite kernels, Cristianini and Shawe-Taylor (2000) or Schölkopf and Smola (2002) for kernel methods and Schölkopf et al. (2004) for the applications to computational biology.

1.3.1 Kernels and kernel methods

Many widely-used statistical and machine learning algorithms, including for example PLS or ANN, are designed to manipulate vector data. Using these tools to manipulate and analyze data such as proteins or small molecules which are not intrinsically vectorial therefore poses the problem of representing these data as vectors or, equivalently, defining a set of binary or real-valued descriptors for these data and stacking them to form a vector. The design of molecular descriptors to describe various features of proteins or small molecules has indeed been much investigated over the last decades, and many such descriptors are nowadays routinely used in combination with statistical methods to correlate the structures of molecules with their physicochemical or biological properties. The explicit computation of a finite number p of molecular descriptors to represent a molecule x by a vector $\Phi(x) = (\Phi_1(x), \ldots, \Phi_p(x))$ nevertheless raises several challenges, including the problem of choosing a set of descriptors sufficiently large to capture the relevant features for a given problem and sufficiently small to allow fast computation and efficient storage.

Kernel methods, including SVM, are a class of algorithms that follow a slightly different strategy to solve the problem of data representation (Schölkopf and Smola, 2002; Schölkopf et al., 2004; Shawe-Taylor and Cristianini, 2004). Data do not need to be represented individually as vectors, they need instead to be compared to each other. More formally, instead of converting each protein or small molecule x into a p-dimensional vector $\Phi(x)$ for further processing, kernel methods can manipulate data only through a function $k(x, x')$ that compares any two proteins (or small molecules) x and x' and returns a real number. The function k is called a *kernel*, hence the name kernel methods. As a result, when a set of n proteins (or of n small molecules) $x_1, \ldots, x_n$ is given as input to a kernel method, the algorithm can only manipulate the data through the *Gram matrix*, which is the square $n \times n$ matrix K of all pairwise similarities, whose entry $K_{i,j}$ is equal to $k(x_i, x_j)$.

Positive definite kernels

Only a certain class of functions k, however, can be used in combination with kernel methods. These kernels are often called *positive definite kernels* or more simply *valid kernel*. The technical conditions that a function $k(x, x')$ must fulfill to be a valid kernel over a space $\mathcal{X}$ are :

1. to be symmetric, *i.e.*, $\forall x, x' \in \mathcal{X}, k(x, x') = k(x', x)$,

2. to be positive definite, *i.e.*,$\forall n \in \mathbb{N}, \forall x_1, \ldots, x_n \in \mathcal{X},$ and $\forall a_1, \ldots, a_n \in \mathbb{R}$,

$$\sum_{i,j=1}^{n} a_i a_j k(x_i, x_j) \geq 0.$$

Although the second condition can sometimes be difficult to assess for a newly defined function k, mathematics textbook abound on examples of valid kernels and on systematic techniques to create them (Aronszajn, 1950; Berg et al., 1984; Berlinet and Thomas-Agnan, 2003). For example, given any representation of a molecule x by a vector of p descriptors $\Phi(x)$, the inner product between two vectors $\Phi(x)$ and $\Phi(x')$ representing two molecules x and x' is a valid kernel:

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle = \sum_{i=1}^{p} \Phi_i(x) \Phi_i(x') \,. \tag{1.7}$$

When such kernels are used, the vectors of descriptors $\Phi(x)$ are explicitly computed prior to the computation of inner products, and kernel methods like SVM are not fundamentally different from other methods such as PLS or ANN.

Reproducing kernel Hilbert spaces

Interestingly, it can be shown that, conversely, any valid kernel $k(x, x')$ can be written as an inner product (1.7), for some vector representation $\Phi(x)$ (Aronszajn, 1950) :

Theorem 1 (Aronszajn, 1950)**.** *K is a p.d. kernel on the set $\mathcal{X}$ if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping*

$$\Phi : \mathcal{X} \mapsto \mathcal{H},$$

such that, for any x, x' in $\mathcal{X}$:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{H}} \,,$$

where $\langle ., . \rangle_{\mathcal{H}}$ denotes the inner product in $\mathcal{H}$.

The proof involves the explicit construction of $\mathcal{H}$ from the set of functions of form :

$$\forall x \in \mathcal{X}, f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x),$$

together with their limit under the norm $\|f\|_{\mathcal{H}}^2 = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)$. This space can be proved to be a Hilbert space endowed with the following dot product :

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_i \beta_j k(x_i, x_j),$$

between any two functions $f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$ and $g(x) = \sum_{i=1}^{m} \beta_i k(x, x_i)$. In particular, taking $g(x') = k(x, x')$ for any particular $x \in \mathcal{X}$ gives the following expression of any function f of $\mathcal{H}$ as a dot product in $\mathcal{H}$:

$$\forall (f, x) \in \mathcal{H} \times \mathcal{X}, f(x) = \langle f, k(x, .) \rangle_{\mathcal{H}}.$$

A direct consequence, known as the *reproducing property*, is that the kernel evaluation between two points of $\mathcal{X}$ can be written as a dot product in $\mathcal{H}$:

$$\forall x, x' \in \mathcal{X}, k(x, x') = \langle k(x, .), k(x', .) \rangle_{\mathcal{H}},$$

which makes the connection with Theorem 1 : k is the dot product in the space where each point $x \in \mathcal{X}$ is mapped to the function $k(x, .)$. Because of this property, $\mathcal{H}$ is often called the *reproducing kernel Hilbert space* (RKHS) associated to k.

Practical use : kernel trick and representer theorem

This analysis apparently establishes an equivalence between the use of valid kernels, on the one hand, and the use of explicit vector representations, on the other hand. In the converse statement, however, the vector $\Phi(x)$ are not necessarily of finite dimension, they can also involve an infinite number of descriptors. In that case, there is obviously no hope to compute the infinitely many descriptors explicitly and store them in a computer, and a

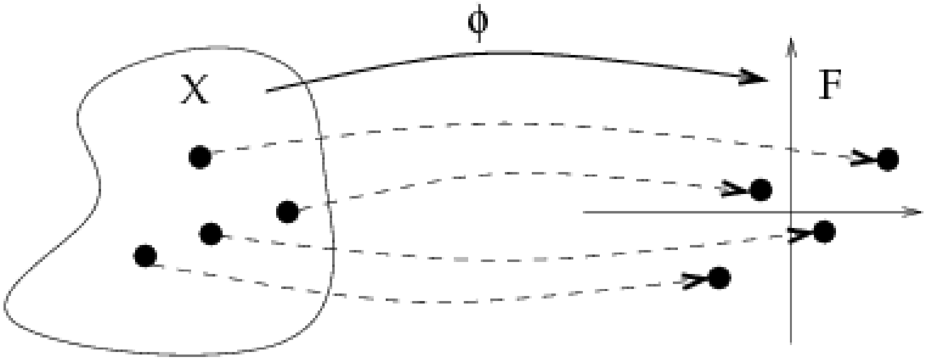

Figure 1.8: Defining a kernel over a space $\mathcal{X}$, such as the space of all small molecules or the space of all proteins, is equivalent to embedding $\mathcal{X}$ in a vector space F of finite or infinite dimension through a mapping $\Phi : \mathcal{X} \mapsto F$. The kernel between two points in $\mathcal{X}$ is equal to the inner products of their images in F, as shown in (1.7)

computational trick must be found to compute directly the kernel $k(x, x')$ without computing $\Phi(x)$ and $\Phi(x')$. We review several examples of such kernels in the next two sections. As a result, the kernel approach can be seen as a generalization of the descriptor vector approach, where the number of descriptors can be finite or infinite (Figure 1.8).

Valid kernels therefore always define a vector space structure over the set of molecules to be manipulated. This structure can either be defined *explicitly*, when molecular descriptors are computed in order to evaluate the kernel similarity through inner products of Tanimoto coefficients between fingerprints, or *implicitly*, when a valid kernel function $k(x, x')$ is directly computed to compare two molecules x and x'. Yet this implicit construction is sufficient to perform various data processing and manipulation in the vector space. As a simple illustration let us consider the problem of computing the distance between two feature vectors $\Phi(x)$ and $\Phi(x')$ corresponding to two data points x and x', as illustrated in Figure 1.9. A simple computation shows that:

$$\begin{aligned} \|\Phi(x) - \Phi(x')\|^2 &= \langle \Phi(x), \Phi(x) \rangle + \langle \Phi(x'), \Phi(x') \rangle - 2\langle \Phi(x), \Phi(x') \rangle \\ &= k(x, x) + k(x', x') - 2k(x, x') \,, \end{aligned} \tag{1.8}$$

where k is the kernel associated to the vector Φ through (1.7). This equation shows that

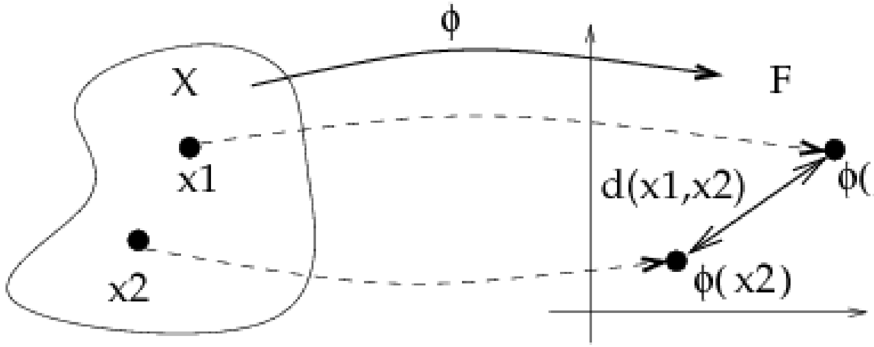

Figure 1.9: We can define the distance between two objects x_1 and x_1, such as two molecules or proteins, as the Euclidean distance between their images $\Phi(x_1)$ and $\Phi(x_2)$. If the mapping Φ is defined by a valid kernel k, then this distance can be computed easily without computing $\Phi(x_1)$ and $\Phi(x_2)$, as shown in (1.8). This *kernel trick* can be extended to a variety of linear algorithms that only manipulate the data through inner products.

in order to compute the distance between points in the feature space, one does not necessarily need to first compute explicitly the vectors themselves, and can rely instead on the corresponding kernel. This trick, known as the *kernel trick*, can be applied to any algorithm for vectors that can be expressed in terms of inner products: replacing each inner product by the respective kernel evaluation allows to execute the algorithm implicitly in the feature space defined by a valid kernel. A surprising variety of methods, collectively known as *kernel methods*, can benefit from this trick. Besides the evaluation of distances using (1.8), one can mention for example dimensionality reduction with principal component analysis (PCA) (Schölkopf et al., 1999), regression and pattern recognition with Gaussian process (Williams, 1998; Seeger, 2004), PLS (Rosipal and Trejo, 2001), SVM (Boser et al., 1992; Vapnik, 1998), or outlier detection with one-class SVM (Schölkopf et al., 2001b). We refer the reader to various textbooks and survey articles for more details on these algorithms (Schölkopf and Smola, 2002; Schölkopf et al., 2004; Shawe-Taylor and Cristianini, 2004).

Another important practical property of kernels and their RKHS, is given by the *representer theorem* :

Theorem 2. *Let $\mathcal{X}$ be a set endowed with a kernel k and $\mathcal{S} = \{x_1, \ldots, x_n\} \subset \mathcal{X}$ a finite set of objects. Let $\Psi : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ be a function of $n+1$ arguments, strictly monotonic* *increasing in its last argument. Then any solution of the problem :*

$$\min_{f \in \mathcal{H}} \Psi(f(x_1), \dots, f(x_n), \|f\|_{\mathcal{H}}),$$

where $\{\mathcal{H}, \|.\|_{\mathcal{H}}\}$ *is the RKHS associated with* k*, admits a representation of the form :*

$$\forall x \in \mathcal{X}, f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i),$$

for some $\alpha_1, \dots, \alpha_n \in \mathbb{R}$*.*

This theorem was stated in this form in Kimeldorf and Wahba (1971), generalized in Schölkopf et al. (2001a) and extended to some non-Hilbertian norms in Abernethy et al. (2008) and Argyriou et al. (2008a). It shows that when optimizing a functional over a RKHS with a penalty on the RKHS norm, the solution always lies in a n-dimension space where n is the number of data points in the problem, even if the RKHS is of infinite dimension. It can also be used in practice to "kernelize" optimization problems as an alternative to the kernel trick.

In summary, the definition of a positive definite kernel for certain types of data defines explicitly or implicitly a mapping of the data to a vector space, possibly of high or infinite dimension. Yet a variety of data processing and analysis algorithm can be performed in this feature space thanks to the kernel trick, without the need to compute and store the vector representing the objects. In the next two sections, we review some recent work focusing on the definition of valid kernels for proteins and small molecules, respectively, to illustrate the possibilities offered by kernels to define implicitly "biological" and "chemical" spaces.

1.3.2 Kernels for proteins

Bioinformatics has historically been one of the first application domain for SVM and kernel methods (Schölkopf et al., 2004), and has triggered a lot of research focused on the design of valid kernels for non-vectorial object, such as proteins. The simplest representation of a protein is the sequence of amino acids it contains, which mathematically is a string in an alphabet of 20 letters. Alternatively, when available or predicted, one can represent a protein by its 3D structure, which is likely to contain more relevant information related to physical interactions with other proteins or ligands. As summarized in Table 1.1, three main strategies have been followed to define kernels between proteins: (i) computing an inner product with descriptors explicitly defined, (ii) deriving a kernel from a probabilistic model, and (iii) adapting widely used measures of similarity between biological sequences or 3D structures. We now review in more details these different strategies, starting with kernels defined for amino acid or nucleotide sequences.

Strategy	Input data	Examples
Define a list of descriptors	Sequence	Physico-chemical kernels (Wang et al., 2004; Zhang et al., 2003) Spectrum, mismatch kernels (Leslie et al., 2002; Leslie and Kuang, 2004; Kuang et al., 2004, 2005) Pairwise, motif kernel (Logan et al., 2001; Ben-Hur and Brutlag, 2003; Liao and Noble, 2003)
	3D Structure	Kernel based on 3D descriptors (Dobson and Doig, 2005)
Derive a kernel from a generative model	Sequence	Fisher, TOP kernel (Jaakkola et al., 2000; Tsuda et al., 2002a) Mutual information kernels (Cuturi and Vert, 2005) Marginalized kernels (Tsuda et al., 2002b; Vert et al., 2006; Kin et al., 2002)
	3D Structure	Random walk kernel(Borgwardt et al., 2005)
Derive a kernel from a measure of similarity	Sequence	Local alignment kernels (Haussler, 1999; Watkins, 2000; Vert et al., 2004; Saigo et al., 2004; Rangwala and Karypis, 2005)
	3D Structure	Structure alignment kernel (Qiu et al., 2007)

Table 1.1: A typology of kernels for proteins.

The first strategy to make a kernel is to define a set of descriptors to characterize various features of protein sequences, and to compute the inner products between the resulting vectors to obtain a kernel. As an example, Leslie et al. (2002) uses as descriptors how many times each sequence of n letters occurs consecutively in the string, for a fixed integer n (a sequence of n contiguous letters is called a n-mer). These descriptors could be relevant to detect homologous proteins, which are likely to contain similar contents of the various n-mers, or to predict biological properties that depend on short motifs of amino acids. Taking for instance $n = 2$, the DNA sequence $x = AATCGCAACT$ is represented by the 16-dimensional vector $\Phi(x) = (2, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0)$, where the numbers are the counts of occurrences of each 2-mer $AA, AC, \ldots, TG, TT$ lexicographically ordered. The dimension of $\Phi(x)$ is then 4^n for nucleotide sequences, and 20^n for amino acid sequences, which can be prohibitively large for, e.g., $n = 5$. Fortunately, Leslie et al. (2002) show that a computational trick allows to compute the kernel between two sequences with a complexity in time and memory linear with respect to the sum of the length of the sequences, independently from the dimension of $\Phi(x)$ (Leslie et al., 2002; Vishwanathan and Smola, 2004). Approximated approaches (Kuksa et al., 2008) allow to further improve the scalability of this type of this kernek. Several variants have also been proposed, including kernels based on counts of n-mers appearing with up to a few mismatches in the sequences (Leslie et al., 2004), matching of n-mers with the possibility of gaps or substitution (Leslie and Kuang, 2004), or counts of n-mers derived from a profile instead of a single sequence (Kuang et al., 2004, 2005). Alternatively one can first replace each amino acid by one or several numerical features, such as physico-chemical properties, and then extract features from the resulting variable-length numerical time series using classical signal processing techniques such as Fourier transforms (Wang et al., 2004) or autocorrelation analysis (Zhang et al., 2003). The resulting features can be explicitly computed to form a vector, and any valid kernel for vector can then be used. These descriptors are interesting to encode information about the variations of physico-chemical properties along the sequence, e.g., to detect elements of the secondary structure. Finally, another popular approach to design features and therefore kernels for biological sequences is to "project" them onto a fixed dictionary of sequences or motifs, using classical similarity measures, and to use the resulting vector of similarities as the feature vector. For example, Logan et al. (2001) represent each sequence by a 10,000-dimensional vector indicating the presence of 10,000 motifs of the BLOCKS database; similarly, Ben-Hur and Brutlag (2003) use a vector that indicates the presence or absence of about 500,000 motifs in the eMOTIF database, requiring the use of a tree structure to compute efficiently the kernel without explicitly storing the 500,000 features; and Liao and Noble (2003) represent each sequence by a vector of sequence similarities with a fixed set of sequences. The choice of sequences or motifs to be included in the dictionary is crucial and may be problem dependent, as it allows to extract for example the occurrences of particular functional or structural motifs in the protein sequences.

A second strategy to design kernels for amino acid sequences has been to derive them from probabilistic models. Indeed, before the interest on string kernels grew, a number of ingenious probabilistic models had been defined to represent biological sequences or families of sequences, including for example Markov and hidden Markov models for protein sequences, or stochastic context-free grammars for RNA sequences (Durbin et al., 1998). Several authors have therefore explored the possibility to use such models to make kernels, starting with the seminal work of Jaakkola et al. (2000) that introduced the *Fisher kernel*. This kernel uses a parametric probabilistic model to explicitly extract features from each sequence. The features for a sequence x are related to the influence of each parameter of the model on the probability of x. The resulting vector of features, known as the Fisher score vector in statistics, has a fixed dimension equal to the number of parameters in the model, and therefore provides a principled way to map sequences of different length to a vector of fixed length. The Fisher kernel was generalized by the Tangent Of Posterior (TOP) kernel (Tsuda et al., 2002a). Intuitively, the descriptors encoded in the Fisher and TOP kernel describe how each individual sequence differs from a model supposed to represent an “average” sequence, and the choice of the model and its parameters influence therefore a lot the kernel. A second line of thought to make a kernel out of a parametric probabilistic model is to use the concept of mutual information (MI) kernels (Seeger, 2002). Contrary to the Fisher kernel, MI kernels do not provide an explicit finite-dimensional representation for each sequence. Instead the dimensions of the feature space are indexed by all possible values of the model parameters, and the feature $\Phi_\theta(x)$ extracted from the sequence x for the parameter θ is the probability of x under the model P_θ, i.e., $\Phi_\theta(x) = P(x|\theta)$. The computation of this kernel involves a summation over all parameters, i.e., takes the form:

$$K(x, x') = \int P_\theta(x) P_\theta(x') d\mu(\theta),$$

where $d\mu$ is a prior distribution on the parameter space. Hence for practical applications one must chose probabilistic models that allow the computation of the above integral. This was carried out by Cuturi and Vert (2005) who present a family of variable-length Markov models for strings and an algorithm to perform the integral over parameters and models at the same time, resulting in a string kernel with linear complexity in time and memory with respect to the total length of the sequences. There exists an information-theoretic interpretation of mutual information kernels: they quantify how much information is shared between two sequences, in particular if the knowledge of a sequence can be helpful to compress another one. Finally, a third strategy to derive valid kernels from probabilistic models with latent variables, such as HMM, is to build a *marginalized kernel* (Tsuda et al., 2002b). Latent variables in probabilistic models often represent meaningful information, such as the local structure or function of a protein sequence. The basic idea behind marginalized kernel is to first design a kernel over the latent and observed variables, as if the latent ones were observed, and then to take the expectation of the kernel with respect to the conditional distribution of the latent variable given the sequences. As for the MI kernel, this kernel can only be computed for judicious choices of random models. Several beautiful examples of such kernels for various probabilistic models have been worked out, including hidden Markov models for sequences (Tsuda et al., 2002b; Vert et al., 2006) or stochastic context-free grammars for RNA sequences (Kin et al., 2002).

A third strategy to define a kernel is to go back to the interpretation of kernels as "measure of similarity", and try to adapt well-known measures of similarities between biological sequences to make valid kernels. This idea was pioneered by Haussler (1999) introduced the concept of *convolution kernels* for structured objects that can be decomposed into subparts, such as sequences that can be decomposed into subsequences concatenated to each other (see also Vert et al. (2004)). Convolution kernels offer the possibility to combine several kernels adapted to each subpart of the sequences into a single kernel for the whole sequence. Besides proving the validity of convolution kernels, Haussler (1999); Watkins (2000) give several examples of convolution kernels relevant for biological sequences. This work is extended by Vert et al. (2004); Saigo et al. (2004) where a valid convolution kernel based on the alignment of two sequences is proposed. This kernel, named *local alignment kernel*, is a close relative of the widely used Smith-Waterman local alignment score (Smith and Waterman, 1981), and gives excellent results on the problem of detecting remote homologs of proteins. This work was later extended to alignment kernels for sequence profiles (Rangwala and Karypis, 2005). This strategy is particularly relevant when the kernel is used by a SVM to predict a property that is conserved across "similar" sequences. In particular, the local alignment score attempts to quantify a measure of evolutionary distance, and the local alignment kernel is therefore particularly adapted to predict biological properties conserved during evolution.

While kernels for sequences, that implicitly map proteins to a feature space through their primary structure, have by far attracted the largest attention so far, several groups have recently attempted to map protein 3D structures through the construction of kernels between structures. Dobson and Doig (2005) explicitly represent each structure by a vector made of carefully chosen features, such as secondary structure content, amino acid propensities, surface properties, etc. Alternatively, Borgwardt et al. (2005) use a representation based upon walks defined on a graph of secondary structural elements, while Qiu et al. (2007) show that a kernel derived from a measure of structure superpositions is more efficient to relate the structure of a protein to its function.

These kernels for proteins have been widely applied, often in combination with SVM, to various classification tasks in computational biology, including for example the prediction of structural or functional classes (Ding and Dubchak, 2001; Jaakkola et al., 2000; Vert et al., 2004; Karchin et al., 2002; Cai et al., 2003; Dobson and Doig, 2005; Borgwardt et al., 2005; Qiu et al., 2007) or the prediction of the subcellular localization of proteins (Hua and Sun, 2001; Park and Kanehisa, 2003; Matsuda et al., 2005). The performance reported in these studies are often state-of-the-art, which might be in large part due to the efficacy of algorithms like SVM to estimate classification or regression function. While each kernel for proteins corresponds to a particular embedding of the space of proteins in a vector space, it has been observed that the choice of the kernel, hence of the embedding, can have an important effect on the final performance of the algorithm. For example, in the context of remote protein homology detection, Vert et al. (2004) compared different kernels and observed that the local alignment kernel was particularly efficient for this application. Besides the performance criterion, different kernels can have different computational complexities which might become prohibitive if large datasets are to be processed. Hence in practical applications the choice of a particular kernel is often a trade-off between computational consideration and performance.

1.3.3 Kernels for small molecules

Kernel methods are also increasingly used in chemoinformatics for various analysis, regression or classification tasks with small molecules. We now review the main recent contributions in this field, as summarized in Table 1.2.

Strategy	Input data	Examples
Use classical fingerprints of molecular descriptors	1D to 4D structure	Tanimoto or inner products between fingerprints (Burbidge et al., 2001; Ralaivola et al., 2005; Azencott et al., 2007)
Use an infinite number of descriptors and a computational trick	2D structure	Walk kernels (Kashima et al., 2003, 2004; Gärtner et al., 2003; Mahé et al., 2005) Shortest-path fragment kernel (Borgwardt and Kriegel, 2005) Subtree kernel (Ramon and Gärtner, 2003; Mahé and Vert, 2006) Cyclic fragment kernel (Horváth et al., 2004)
	3D structure	Pharmacophore kernel (Mahé et al., 2006)
Use a measure of similarity	2D structure	Optimal assignment kernel (Fröhlich et al., 2005)

Table 1.2: A typology of kernels for small molecules

The problem of explicitly representing and storing small molecules as finite-dimensional vectors has a long history in chemoinformatics, and a multitude of molecular descriptors have been proposed (Todeschini and Consonni, 2002). These descriptors include in particular physicochemical properties of the molecules, such as its solubility or logP, descriptors derived from the 2D structure of the molecule, such as fragment counts or structural fingerprints, or descriptors extracted from the 3D structure. All classical vector fingerprint and vector representations of molecules define an explicit “chemical space” where each molecule is represented by a finite-dimensional vector, and these vector representations can obviously be used as such to define kernels between molecules (Burbidge et al., 2001; Azencott et al., 2007).

Alternatively, several groups have investigated different strategies to build implicit chemical spaces by defining kernels between molecules that do not require the explicit computation of vector representations. These attempts were pioneered simultaneously and independently by Kashima et al. (2003, 2004) and Gärtner et al. (2003) who proposed to represent the 2D structure of a molecule by an infinite-dimensional vector of linear fragment counts and showed how SVM can handle this representation with the kernel trick. Mahé et al. (2005) extended these works by showing how irrelevant fragments can be filtered out and proposing a trick to increase the dimension of the feature space to make the fragments more specific while simultaneously increasing the speed of the kernel computation. Ralaivola et al. (2005) also tested several variants of these kernels, and showed in particular that the Tanimoto index, widely used in chemoinformatics, is a valid kernel. Borgwardt and Kriegel (2005) investigated the possibility to restrict the fragment counts to shortest-path fragments. Several groups have also tried to extend the substructures extracted from the molecular graphs beyond linear fragments, and therefore to trade some increase in expressiveness against loss in computational efficiency (Ramon and Gärtner, 2003). For example, Horváth et al. (2004) considers kernels based on cyclic fragments, while Ramon and Gärtner (2003) suggests to consider tree fragments instead of linear fragments, an idea that was later extended and validated in (Mahé and Vert, 2006). Finally, Fröhlich et al. (2005) defines a kernel between molecular graphs by scoring an optimal matching between the atoms of two molecules to be compared; this kernel, however, is not a valid one (Vert, 2008).

A few attempts to define kernels based on the 3D structure of molecules have also been proposed. Mahé et al. (2006) design a kernel focused on the detection of 3D pharmacophores, while Swamidass et al. (2005) considers similarity measures between histograms of pairwise distances between atom classes and Azencott et al. (2007) use Delaunay tetrahedrization and other techniques from computational geometry to characterize the 3D structures of small molecules and make kernels. Finally, Azencott et al. (2007) shows how kernels can also handle multiple 3D conformations and demonstrates the relevance of this idea on several QSAR experiments.

Although the construction of valid kernels for molecules is a young discipline, it has witnessed impressive progresses in just a few years, triggered by potential application in chemoinformatics and drug design. Large avenues that could be relevant for kernel design remain however largely unexplored, such as the modeling of 3D surfaces, their electrostaticity and polarity, or the dynamics of the structures. We expect fast progresses in this field in the coming years.

1.4 Prior knowledge and regularization

Another dominant theme in modern statistical machine learning is the regularization of empirical risk minimization problems, generally guided by some prior knowledge on what the solution should be like. In this section, we introduce this concept and present existing attempts to regularize empirical risk minimization problems using various kinds of prior knowledge. Regularization is a central notion in this thesis, and underlies all the methods which are presented throughout Chapter 2 to 4.

1.4.1 Overview

Motivations

Regularization-based approaches are increasingly popular in machine learning and statistics, providing an intuitive and principled tool for learning from high-dimensional data. The underlying idea of these methods, which we started to motivated in section 1.1.2, is to bias estimation problems towards more regular solutions, where the notion of regularity depends on some prior knowledge of what the solution should be. This allows to turn ill-posed problems into well-posed problems, and to avoid overfitting when little data is available with respect to the complexity of the problem.

Historically, it stems from the notion of ill-posed problem by J. Hadamard (Hadamard, 1902, 1923). A well posed problem is a problem such that :

1. A solution exists,

2. The solution is unique,

3. The solution depends continuously on the data, in some reasonable topology.

A typical example of nonwell posed problem is the Cauchy problem for the Laplace equation, given in Hadamard (1923). For a long time, little effort was made by mathematicians to address this type of problem, partially because many of them considered it as a waste of time : a problem for which any of these conditions failed could not be of any physical interest (Levine, 1979).

The first solutions to this problem appear in the independent work of A. N. Tikhonov and D. L. Philips on integral equations (Tikhonov, 1943; Philips, 1962; Tikhonov, 1963; Tikhonov and Arsenin, 1977). The idea was to *regularize* the minimization problem which was solved when computing the solution of integral equations by adding a term penalizing the Hilbertian norm of the function, hence improving the conditioning of the problem.

In modern machine learning and statistics, regularization has naturally emerged as a dominant theme because of the new type of problems that have to be addressed and the new type of data that have become available. While multivariate linear regression was traditionally used to learn few parameters from a reasonably large number of data points ("small p, large n" problems), typical computational biology or computer vision problems involve more parameters than training data points, which makes them ill-posed. However, as early as in the 1950s, statisticians were already facing ill-posed problems when solving their multivariate linear regression problems with rank-deficient or ill-conditioned matrices (Hoerl and Kennard, 1982) :

> We were charging $90/day for our time, but had to charge $450/hour for computer time [...], we found that we had both encountered the same phenomenon, one that had caused some embarrassment with clients. We found that multiple linear regression coefficients computed using least squares didn't always make sense when put into the context of the process generating the data. The coefficients tended to be too large in absolute value, some would even have the wrong sign, and they could be unstable with very small changes in the data.

The ridge regression

This motivated the introduction of the *ridge regression* (independently from Tikhonov's and Philips' work) for the parametric case (Hoerl, 1962; Hoerl and Kennard, 1970). The idea of ridge regression is to penalize the least-square minimization problem by the ℓ_2 norm of the parameter vector :

$$\min_{w \in \mathbb{R}^d} \sum_i (w^\top x_i - y_i)^2 + \lambda \|w\|_2^2, \tag{1.9}$$

where $\|.\|_2$ denotes the usual ℓ_2 norm of a vector, $\|w\|_2 = \left(\sum_{j=1}^d w_j^2 \right)^{\frac{1}{2}}$. This results in the estimate $\hat{w} = (X^\top X + \lambda I)^{-1} X^\top Y$, which has several interesting properties :

- For $\lambda \neq 0$, it is defined even if $X^\top X$ is rank-deficient. If $(X^\top X)^{-1}$ is defined, it recovers the solution of the unpenalized least-squares estimator for $\lambda = 0$.
- It adds an offset to the eigenvalues of $X^\top X$, which makes the inverted matrix better conditioned.
- It is *smooth* in the sense that two close points x, x' have close evaluations by $\hat{w}$. Indeed, by the Cauchy-Schwarz inequality,

  $$|\hat{w}^\top x - \hat{w}^\top x'| \leq \|\hat{w}\|.\|x - x'\|,$$

  which is small because the penalty in the ridge regression enforces a small $\|\hat{w}\|$.

The first two points are related to the definition of the problem and its numerical conditioning. Intuitively, if the problem is unpenalized and we try to find a linear function which fits few points in high dimension, the problem is ill-defined because there is no unique solution (as long as $n < p$). The regularization makes the solution unique by imposing to choose the one with the smallest ℓ_2 norm. But even if the solution is uniquely defined, the problem can have a poor conditioning, *i.e.*, the matrix $X^\top X$ can have very small eigenvalues, which would cause a dramatic sensitivity of the solution to the noise in the input Y. In particular, this happens if some variables are correlated, which is often the case in practice. The regularizer solves this problem because it improves the conditioning of the matrix.

The last point on the other hand relates to the notion of *overfitting*. Indeed, the regularization doesn't make the solution unique by randomly picking one of the functions which correctly fit to the data, it choses the smoothest one. Therefore, the function is forced to have a similar behavior on similar data, and will have better generalization properties, *i.e.* it will be more accurate in evaluating y from x points which were not present in the optimization problem. This is a direct application of Occam's razor principle, which states that when competing hypotheses are equal in other respects, the simplest one should be chosen. This is also a way to deal with the well known bias-variance dilemna in statistics, which was presented in section 1.1.2 : by introducing a bias in the estimator towards 0, we decrease its variance across the choice of the points which are used to build it. Even if its introduction was motivated by numerical conditioning issues more than statistical learning concepts, the ridge regression when used with an adequate regularization parameter selection procedure exactly applies the principle of structural risk minimization formalized later in Vapnik and Chervonenkis (1974).

The ridge penalty

While further theoretical of the ridge regression was still carried on later (Wahba, 1990), the importance of this work in the statistics and machine learning field mostly originates from its applicability to a wide range of problems. Girosi et al. (1995) studied the ridge penalty in the context of neural networks. The ℓ_2 penalty was also used to regularize classification algorithms like logistic regression (Hastie et al., 2001) and support vector machines (Boser et al., 1992; Vapnik, 1998), yielding in the latter case an interesting *large margin* interpretation.

Bayesian point of view

Although the idea of regularizing an estimator by this ℓ_2 penalty historically stems from very practical considerations, it also admits a very simple interpretation from a Bayesian probabilistic point of view. In a Bayesian probabilitistic model, a *prior* distribution $p(w)$ of the linear function w is chosen, as well as a *likelihood* model $p(D|w)$ of how the data D should be distributed. Then, a *posterior* distribution of w is computed through the Bayes rule,

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)} \propto p(D|w)p(w),$$

and w is estimated from this distribution, for example by finding the w which maximizes the posterior distribution,

$$\hat{w}_{MAP} = \underset{w}{\operatorname{argmax}}\ p(w|D) = \underset{w}{\operatorname{argmax}}\ p(D|w)p(w), \tag{1.10}$$

also known as the maximum a posteriori (MAP) estimator[2] (De Groot, 1970; Berger, 1985). Now for a given likelihood model, if one chooses a centered Gaussian distribution for the prior, $p(w) \sim \mathcal{N}(0, \sigma^2 I)$, using the $\log$ of the posterior in (1.10) it is straightforward to see that the Gaussian prior becomes a ℓ_2 penalty in the maximization problem. In particular, for a Gaussian likelihood $p(D|w)$, the problem is exactly equivalent to (1.9). In other words, regularizing a regression problem by the ℓ_2 norm is equivalent choosing the most likely function given the data under the prior that the parameters w_i are independent and normally distributed around 0, *i.e.*, are not likely to have large values unless the data need it. This Bayesian framework has been used in parallel to optimization-based regularization approaches to find ways to improve learning performances by introducing prior knowledge on what the solution should look like. While regularization-based approaches try to formulate this prior knowledge as a penalty in an optimization problem, Bayesian methods formulate it as a prior probability distribution of the learned function. Throughout the next sections on sparse methods and multi-task learning, we will show some specific example of such priors.

Non-parametric extension

The ridge regression as it was proposed in (Hoerl and Kennard, 1970) was a parametric model dealing with linear functions. Using the positive definite kernels we introduced in section 1.3.1, it is possible to make it non-parametric. Indeed recall that

$$\hat{w} = (X^\top X + \lambda I_p)^{-1} X^\top Y = X^\top (X X^\top + \lambda I_n)^{-1} Y,$$

[2] In a pure Bayesian setting however, w is only manipulated through its distribution and predictions are made by averaging over all possible w wieghted by their prior probability.

so the prediction Y_{test} from new points X_{test} is

$$Y_{test} = X_{test}\hat{w} = X_{test}X^\top(XX^\top + \lambda I_n)^{-1}Y,$$

which only depends on x through dot products, so the kernel trick applies. Replacing all the dot products by kernel evaluations in this prediction is equivalent to using the solution of the non-parametric problem

$$\min_{f\in\mathcal{H}} \sum_{i=1}^{n}(f(x_i) - y_i)^2 + \lambda\|f\|_{\mathcal{H}},$$

where $\mathcal{H}$ is the RKHS corresponding to the chosen kernel. In particular, by the same Cauchy-Schwarz-based argument as in the linear case, the optimal function has the same smoothness (hence good generalization) properties, because data points whose mappings to $\mathcal{H}$ are close (in the metric of $\mathcal{H}$) have close evaluations by f. Note that similar arguments can be used in the non-parametric Bayesian framework to turn the parametric Bayesian model presented above into a non-parametric *Gaussian process* model (Rasmussen and Williams, 2005).

An interesting example of such a parametric extension is the case where a graph on the variables is known, with the prior that parameters corresponding variables which are close with respect to the graph topology should have similar values, *i.e.*, the function should be smooth on the graph. If $L = E\Lambda E^\top$ is the spectral decomposition of the graph Laplacian L, then this effect can be obtained by penalizing the Hilbert norm of the following function space :

$$\Phi(x) = \Lambda^{\frac{1}{2}}E^\top x, \tag{1.11}$$

i.e., the projection of each data point on the square root of the graph Laplacian. Penalizing the Hilbert norm in this space (which is actually still Euclidean) gives :

$$\|w\|_{\mathcal{H}} = \Phi(w)^\top\Phi(w) = w^\top Lw = \sum_{l\sim k}\|w_l - w_k\|_2^2, \tag{1.12}$$

where $l \sim k$ indicates that the l-th and the k-th variables are connected on the graph. In other words, enforcing smoothness in the description space (1.11) is equivalent to enforcing smoothness on the graph topology. This type of penalty was used in Rapaport et al. (2007) to obtain interpretable classifiers of tumor expression data based on a biological graph.

While the regularization by Hilbertian norms improves the problem conditioning and the generalization ability of the solution, it only relies on the assumption that the function should be as smooth as possible or in the parametric case that the parameters have reasonable values. If more information is available on the expected structure of the solution, building a regularization which takes it into account could further improve the performance of the learned function. In the next two sections we present two particular families of assumptions, namely that the true model is sparse, and that the true models of several problems are related.

1.4.2 Sparsity-inducing regularizations

Motivation

A particular and very popular assumption is that the true model is *sparse*, *i.e.*, has a lot of zero parameters. This assumption is especially helpful when trying to learn a function in high dimension because most phenomenon only involve few of the many features injected in the model, and learning a non-zero weight for all the other features may only add noise. Besides, selecting a small set of explicative features makes the model interpretable and is as important as learning an accurate classifier in some applications, especially in computational biology.

Early selection methods, known as model selection approaches, like the AIC criterion (Akaike, 1973), C_p (Mallows, 1973) or BIC (Schwarz, 1978), try to achieve this sparsity by penalizing the dimension of the model. Although they are based on a search over all possible subset selections, they are shown to give optimal prediction performances in some settings, and are still an active field of research (Birgé and Massart, 2006; Baraud et al., 2009).

On the other hand, a possible formalization of this sparsity assumption in terms of regularization of an empirical risk minimization problem for a given loss function L (Foster and George (1994) for the least-square case) is the following :

$$\min_{w} L(w) + \lambda \|w\|_0, \tag{1.13}$$

which has the same form as the Tikhonov formulation, but where we regularize by the ℓ_0 instead of the ℓ_2 norm. In other words, formulation (1.13) penalizes the number of non-zero elements of w, which enforces the sparsity assumption.

A severe shortcoming of the ℓ_0 regularizer however is that like model selection approaches, it is non-convex (Natarajan, 1995) and therefore subject to the local minima problem. A natural paradigm is to try to minimize this non-convex problem directly using greedy methods like the matching pursuit (Mallat and Zhang, 1993) or the orthogonal matching pursuit (Tropp, 2004). It is still unclear in practice how much and under which exact settings this non-convexity harms the prediction performances and model recovery, as greedy methods empirically perform well in many cases.

Sparsity induction by the ℓ_1 norm

Another popular approach is to use a surrogate constraint which is convex has the same effect as the ℓ_0 constraint. A possible convex relaxation of the ℓ_0 constraint is the ℓ_1 constraint, as shown on Figure 1.1. This is actually the tightest convex relaxation among all the ℓ_p norms. Regularization by the ℓ_1 constraint was introduced independently in the statistics (Tibshirani, 1996) and the signal processing (Chen et al., 1998) literatures[3] in two very close formulations. In statistics, it was first proposed in a regularized regression problem, the *least absolute shrinkage and selection operator* (Lasso) :

$$\begin{cases} \min\limits_{w} \sum\limits_{i=1}^{n} (y_i - x_i w)^2 \\ \|w\|_1 \leq C, \end{cases} \tag{1.14}$$

[3] Historically, the first use of a ℓ_1 constraint was by the Nobel Prize winner Harry Markowitz, in his work on portfolio optimization (Markowitz, 1952), although no explicit mention to the notion of regularization was made in this work.

whereas in signal processing it was introduced as the *basis pursuit* :

$$\begin{cases} \min\limits_{w} \|w\|_1 \\ Xw = y, \end{cases} \tag{1.15}$$

as a mean to recover exactly the signal w from a given overcomplete dictionnary X (the *basis pursuit denoising* formulation is equivalent to the Lasso). Note that by a Lagrangian argument (Boyd and Vandenberghe, 2004), (1.14) can be written under the same Tikhonov form as the ridge regression and the ℓ_0 regularization (1.13) :

$$\min_{w} \sum_{i=1}^{n} (y_i - x_i w)^2 + \lambda \|w\|_1, \tag{1.16}$$

and for all C, there exists $\lambda(C)$ such that the two problems are equivalent.

The two following arguments explain why penalizing by the ℓ_1 norm favors sparse solutions :

Geometric argument : Figure 1.10 illustrates in two dimensions the minimization of a smooth function, represented by the contour lines under ℓ_1 and ℓ_2 constraints represented by the green zones. If w_{min} is the minimizer of the smooth function (the center of the ellipses), the point inside the ℓ_2 ball of radius C which minimizes the function is the projection of w_{min} on the ball, $C.\frac{w_{min}}{\|w_{min}\|}$, which is colinear to w_{min} and has no reason the have any zero coordinate (unless w_{min} itself has zero coordinates, which has probability zero under any reasonable noise setting).

Under the ℓ_1 constraint on the other hand, the w minimizing the smooth function, *i.e.*, the projection of w_{min} on the ℓ_1 ball, generally lies on one of the singularities of the ball, so the constrained solution typically has zero coordinates.

KKT argument : Consider the minimization of the following general problem :

$$\min_{w} L(w) + \lambda \|w\|_p, \tag{1.17}$$

for $1 \leq p \leq \infty$, where $\|w\|_p = \left(\sum_{j=1}^{d} |w_j|^p\right)^{\frac{1}{p}}$, as defined in (1.5). This problem is

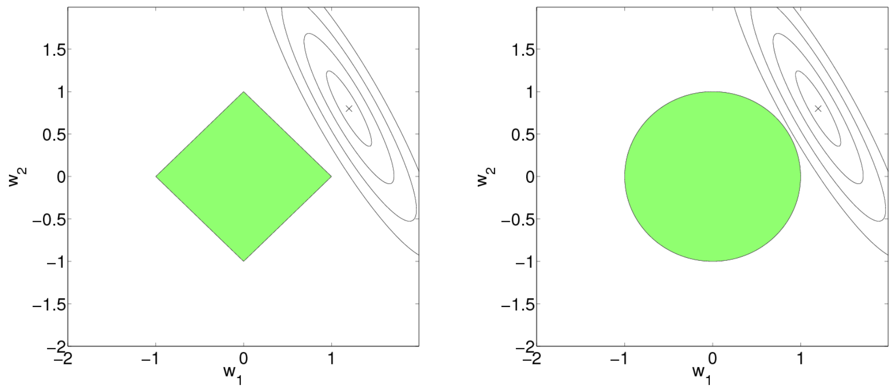

Figure 1.10: Sparsity induction by the ℓ_1 norm.

equivalent to $\min_w L(w) + \tilde{\lambda}\|w\|_p^p$ for some $\tilde{\lambda}$ and generalizes the ridge regression, the Lasso, and several other regularized problems.

For a convex loss function L, since $q \geq 1$, (1.17) is a convex optimization problem, whose solution is characterized by its KKT conditions (Boyd and Vandenberghe, 2004). In particular, at the optimum, w must satisfy the stationarity conditions. For $p > 1$, $\|w\|_p$ is differentiable everywhere, and the stationarity conditions are :

$$\forall j, \frac{\partial L}{\partial w_j} = -\lambda \frac{\partial \|w\|_p}{\partial w_j} = -\lambda \text{sign}(w_j) \frac{|w_j|^{p-1}}{\|w\|_p^{p-1}}, \tag{1.18}$$

where sign(w_j) is 1 for a positive w_j, -1 for a negative w_j and 0 for $w_j = 0$. This condition simply means that at the optimum, the derivative of the loss function with respect to each parameter is cancelled out by the absolute value of the parameter (weighted by the strength λ of the constraint), which intuitively makes sense : the former tries to increase the parameter value in order to minimize the loss, whereas the latter penalizes large values of the parameter. As a consequence, for a w_j to be 0 at the optimum for any $\lambda > 0$, the KKT conditions impose that the corresponding $\frac{\partial L}{\partial w_j}$ be 0 as well, which has probability 0 under any non-idealized setting.

For $p = 1$ on the other hand, $|w_j|^p$ is non-differentiable at 0, so for $w_j = 0$, the stationarity condition, in terms of the subdifferential ∂_{w_j} of the loss and the penalty functions becomes :

$$0 \in \partial_{w_j} \left(L(w) + \lambda \|w\|_p \right) \Leftrightarrow \left| \frac{\partial L}{\partial w_j} \right| \leq \lambda, \tag{1.19}$$

because the subdifferential of the absolute value is the $[-1, 1]$ set. Therefore, any parameter with respect to which the gradient of the loss has an amplitude less than λ will be 0 at the optimum. Note that the zone where the parameter is left to 0 because it doesn't help enough the loss decrease is created by the non-differentiability of the penalty. This will be a useful fact to define more elaborate sparsity-inducing norms.

Following the geometric intuition, 1.1 shows that using ℓ_p constraints for $p < 1$ would favor stronger sparsity, because the cost induced by the corresponding metrics is closer to the ℓ_0 cost, *i.e.*, the smaller p, the more expensive it is to add variables to the model. At the other extreme, the ℓ_∞ norm only penalizes the largest parameter in absolute value, so using it to penalize an empirical risk minimization problem results in solutions where all the parameters have the same absolute value.

Algorithms

While the differentiability of the ℓ_2 norm allows to use any gradient-based descent method for the minimization, ℓ_1-penalized problems may seem much more difficult to optimize as it is known that subgradient iterations are slower than gradient descent methods. However, two elements make the ℓ_1 minimization problem amenable in practice :

- For large penalizations, the above analysis shows that the optimal function is very sparse, *i.e.*, few parameters are involved.
- The *regularization path* of the Lasso is piecewise linear. In other words, to describe all the values taken by the parameters across all possible λ, it is sufficient to describe their values at a finite number of points. This was shown in Efron et al. (2004) and generalized in Rosset and Zhu (2007) to any minimization of an affine combination of a piecewise quadratic function and a piecewise linear function.

The main families of algorithms to minimize the Lasso are :

Coordinate descent methods : First introduced by Fu (1998) and re-discovered in Daubechies et al. (2004), they simply consist in iterating over the parameters w_j, and for each parameter to minimize the Lasso objective with respect to the parameter only. This partial minimization can be performed in closed form by soft-thresholding (Donoho, 1994). For this reason, in spite of its naive aspect, and the fact that it does not use the piecewise linearity of the regularization path, this method is very efficient and difficult to improve on in practice.

Homotopy methods : A second family of approaches, pioneered by Osborne et al. (1999a,b) and further developed by Efron et al. (2004) follow the continuous path of each parameter along the regularization values. The idea of the LARS Efron et al. (2004) is to use an active set of parameters which is empty at the beginning (corresponding to a large penalty), then add the variable most correlated with the output and increase the corresponding parameter until the correlation of another variable with the residual is as strong as the correlation of the first variable, then add the new variable to the active set and increase the value of the two parameters until a new variable becomes correlated enough with the residual, *etc*. Under minor modifications, this algorithm can be proved to follow the Lasso regularization path.

Projected gradient methods : Alternatively, it is possible to consider the Lasso problem in its original constrained form (1.14) and minimize it by descending the gradient projected on the ℓ_1 ball (Duchi et al., 2008).

Consistency of the Lasso

Once the practical minimization of the Lasso is solved, the remaining problem, which has concentrated a lot of efforts recently, is its model consistency : under which settings does the Lasso recover the correct sparsity pattern when the number of data points grows? An interesting result in Leng et al. (2004) shows that selecting the regularization parameter based on the regression performances does not lead to consistent models. More recently however, Zhao and Yu (2006) and Yuan and Lin (2007b) show that in the case of a finite number of parameters and for an adequate choice of the regularization parameter, the Lasso is consistent under some irrepresentable conditions, stating that when the variables of the model are not too correlated with the variables which are not in the model. To address these restrictions, Zou (2006) proposes an adaptive version of the Lasso, while Bach (2008c) shows that a bootstrapped version of the Lasso can be consistent even when the irrepresentable conditions are not met. Meinshausen and Buehlmann (2009) adopt a similar approach, but add randomized weights which acts like a soft bootstrapping of the variables.

Note that as a non-asymptotic counterpart to this asymptotical result, recent analysis from the compressed sensing field (Candes and Tao, 2005; Candes et al., 2006; Candes, 2008; Foucart and Lai, 2009) give conditions on the design matrix X under which ℓ_1 minimization leads to exact recovery in the noiseless case. For noisy recovery, the analysis gives a tight bound on the ℓ_1 error made by the estimator. The conditions on X, known as *restricted isometry properties* express that X should be close enough to an isometry in the sense that mapping a vector by X should not change too much the euclidean norm of the vector.

Variations on the Lasso

Several modifications of the Lasso have been proposed. As a direct alternative to the Lasso, Candes and Tao (2007) introduced the *Dantzig selector* :

$$\begin{cases} \min_{w} \|w\|_1 \\ \|X^\top (Y - Xw)\|_\infty \leq C, \end{cases} \tag{1.20}$$

which interestingly relates to the Lasso through the following alternative formulation of the Lasso (Osborne et al., 1999b) :

$$\begin{cases} \min_{w} \frac{1}{2} w^\top X^\top X w \\ \|X^\top (Y - Xw)\|_\infty \leq .C \end{cases} \tag{1.21}$$

Meinshausen et al. (2007) thoroughly discusses the differences of these two approaches.

A growing literature uses the ℓ_1 penalty to recover the structure of a Gaussian graphical model (Meinshausen and Bühlmann, 2006; Friedman et al., 2007; Yuan and Lin, 2007a; Ravikumar et al., 2009). It is indeed well known that the inverse covariance matrix of a Gaussian graphical model has zeros at positions corresponding to couples of variables which are not connected. Penalizing the ℓ_1 norm of this inverse covariance matrix is therefore a natural way to estimate the structure of the graphical model.

In the Bayesian setting we introduced for the ridge regression, penalizing by the ℓ_1 norm can be seen as doing MAP estimation with a Laplacian prior instead of the Gaussian prior. In a nonparametric Bayesian setting, Ravikumar et al. (2008) give a sparse version of the generalized additive models proposed by Hastie and Tibshirani (1999) named SpAM (Sparse Additive Model). While generalized additive models try to explain the output by a sum of non-parametric functions of the individual variables $Y_i = \sum_{j=1}^d m(X_{ij}) + \epsilon$, SpAM adds a constraint on $\sum_{j=1}^d \sqrt{\mathbb{E}(m_j^2(X_j))}$, which gives a non-parametric equivalent to the Lasso. They propose an optimization algorithm and prove the model selection consistency of the approach in the same paper.

As mentioned in the section on Lasso consistency, adaptive (Zou, 2006) as well as bootstrapped (Bach, 2008c) variants of the Lasso have been proposed to improve the model recovery abilities under settings where some variables of the models would be correlated with variables outside the model. On the other hand, a known weakness of the Lasso is that if two variables in the model are too correlated, the Lasso selects only one of them. In an attempt to solve this problem, Zou and Hastie (2005) proposed to combine the ℓ_1 and the ℓ_2 norms : $\Omega_{elastic}(w) = \lambda_1\|w\|_1 + \lambda_2\|w\|_2$. The resulting *elastic net* penalty selects all the variables which are correlated enough instead of selecting only the variable most correlated with the loss gradient and leaving the others. Following the geometric argument we presented above, the projection of the optimum w on the ball of the elastic net, presented on Figure 1.11 doesn't fall on the corners if the unpenalized optimal is close enough to the $w_1 = w_2$ axis. The minimal correlation that the variables must have to be selected together depends on the (λ_1, λ_2) hyperparameters.

Structured sparsity

All the regularizations presented so far only assume that the true model is sparse, without any assumption about how the non-zero coefficients are organized. Various models have been devised to use this type of information when available.

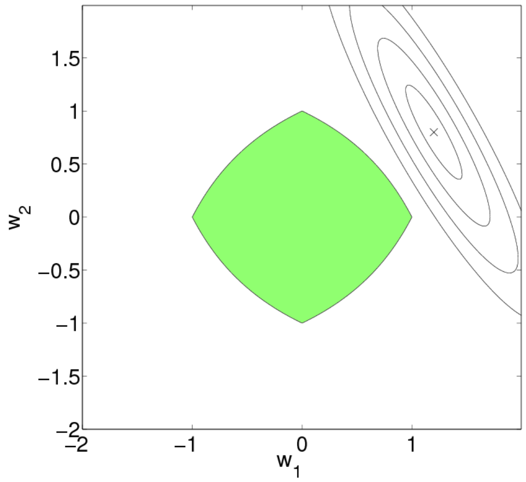

Figure 1.11: Unit ball of the elastic net $\lambda_1\|w\|_1 + \lambda_2\|w\|_2$ for $\lambda_1 = \lambda_2$.

In some applications, a natural order of the variables can be defined, with the assumption that the model is *piecewise constant* on this order. A typical example in computational biology is CGH data analysis, where the variables are copy numbers of each gene. Since duplications and deletions of a chromosome segment are generally not limited to a single gene, it is likely that the genes neighboring a given gene on the chromosome have the same number of copies. Rapaport et al. (2008) and Tibshirani and Wang (2008) proposed to take this information into account by using a fused penalty which combines the ℓ_1 norm of the function and the ℓ_1 norm of the differences of parameters corresponding to successive genes :

$$\Omega_{fused}(w) = \lambda_1\|w\|_1 + \lambda_2 \sum_{j=1}^{p-1} |w_j - w_{j+1}|. \tag{1.22}$$

This penalty had first been proposed in Land and Friedman (1997) without the $\lambda_1\|w\|_1$ term and used under this form in Harchaoui and Levy-Leduc (2008) for change-point detection. It was studied under the complete form (1.22) in Tibshirani et al. (2005) where it was applied to expression and mass spectrometry data. It was further refined in Rinaldo (2009) where an adaptive formulation was proposed. Intuitively, penalizing the ℓ_1 norm of differences favors solutions which have several zero differences, *i.e.*, constant segments. Adding the regular ℓ_1 penalty leads to solutions which are both sparse and piecewise constant. Figure 1.12 shows that following the same geometric argument as for the Lasso confirms this property, as the singularities of the fused ball happen at points where either one parameter is 0 or the two parameters have the same value.

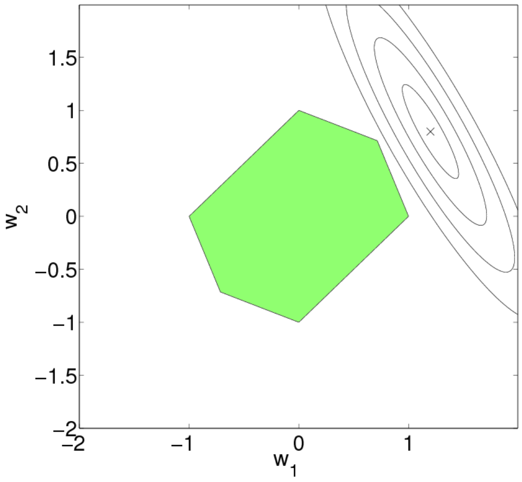

Figure 1.12: Unit ball of the fused penalty (1.22) for $\lambda_1 = \frac{2}{3}, \lambda_2 = \frac{1}{3}$.

Another type of prior information which can be available on the sparsity structure is that some pre-defined groups of variables are likely to be selected together. In this case, regularizing a learning problem by the following *group-lasso* penalty (Yuan and Lin, 2006) favors solutions such that parameters within a group are either all zero or all non-zero :

$$\Omega_{\text{group}} = \sum_{g \in \mathcal{G}} \|w_g\|_2, \tag{1.23}$$

where $\mathcal{G}$ is the set of pre-defined groups forming a partition of the variables. Penalty (1.23) is a mixed ℓ_1/ℓ_2 norm : it is the ℓ_1 norm of the vector formed by the ℓ_2 norms of the groups of variables. Therefore using this penalty favors solutions in which several $\|w_g\|_2$ are 0. For these groups, all the parameters are necessarily 0. On the other hand, all the parameters within groups such that $\|w_g\|_2 \neq 0$ are non-zero with probability 1 for reasons similar to those presented in the above KKT arguments for the Lasso sparsity (within each group, the penalty is ℓ_2 so sparsity is not favored). Here again, it is possible to check on Figure 1.13

that the singularities of the group-lasso ball occur at points where some groups have zero norm. More generally, penalizing an empirical risk minimization problem by a ℓ_p/ℓ_q mixt norm $\left(\sum_{g\in\mathcal{G}} \|w_g\|_q^p\right)^{\frac{1}{p}}$ induces the effect of the ℓ_p norm at the group level, and the effect of the ℓ_q norm within each group.

Contrarily to the Lasso, the group-lasso does not have a piecewise linear regularization path. It is therefore not possible to generalize the homotopy algorithms of the Lasso, but several efficient block-coordinate descent algorithms have been proposed (Yuan and Lin, 2006; Meier et al., 2008; Roth and Fischer, 2008). The latter one uses an active set and is able to deal with millions of variables. Lanckriet et al. (2004b) proposed a non-parametric version of this penalty in the positive definite kernel framework. This *multiple kernel learning* (MKL) selects a small number of kernels instead of selectioning groups of variables. The original formulation was an expensive SDP, but Bach et al. (2004a,b) introduced a scalable SOCP formulation (Lobo et al., 1998) based on conic duality and Moreau-Yosida smoothing (Lemarechal et al., 1997). More recently, Sonnenburg et al. (2006) and Rakotomamonjy et al. (2007, 2008) proposed fast algorithms for MKL minimization, based on a variational formulation. Bach (2008a) gives consistency conditions for both the parametric group-lasso and the MKL minimizations, yielding similar irrepresentable conditions as for the Lasso. Huang and Zhang (2009) gives non-asymptotic results as tight bounds on the ℓ_2 error of the group-lasso under group-RIP conditions, and show that the group-lasso can be inferior to the Lasso if the variables of the model belong to small groups. Finally, Bach (2009) proposes an approximated version of the MKL which is able to deal with a very large number of kernels if a hierarchical structure on the kernels is given.

Parametric variants of the group-lasso include the CAP penalty Zhao et al. (2009), which enforces that a variable enters the model only if its parents enter the model as well for a given hierarchical structure of the variables, and the model of Szafranski et al. (2008b) which introduces sparsity within the groups. Szafranski et al. (2008a) generalizes this model to the non-parametric and allows to increase sparsity at the cost of losing the convexity of the penalty.

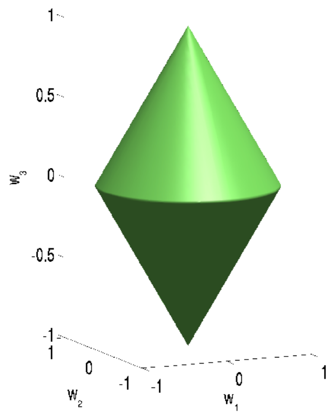

Figure 1.13: Unit ball of the group-lasso in 3 dimensions with $\mathcal{G} = \{\{1, 2\}, \{3\}\}$.

1.4.3 Multi-task and pairwise learning

Another type of prior information

In some practical applications, training data is available for several related yet different learning problems. In marketing, one may want to predict the preferences of several customers. In medicine, one may be interested in predicting the efficiency of various treatments for different patients. Therefore, a new type of prior information becomes available when considering all these learning problems simultaneously, *i.e.*, as a single optimization problem, and new regularizers favoring similar classification functions for related learning tasks can be designed. More precisely when T related learning tasks are available, it is possible to consider the joint minimization of their risk under some regularity constraint :

$$\min_{w_1,\ldots,w_T} \sum_{t=1}^{T} L(w_t) + \lambda\Omega(w_1, \ldots, w_T), \tag{1.24}$$

where w_t is the classification function for task t. A first observation is that if the regularizer Ω is separable on the tasks, *i.e.*, if $\Omega(w_1, \ldots, w_T) = \sum_{t=1}^{T} \tilde{\Omega}(w_t)$, then problem (1.24) can be decomposed as T smaller minimization problems. In particular, if $\Omega(w_1, \ldots, w_T) = \sum_{t=1}^{T} \|w_t\|_2^2$, then problem (1.24) boils down to T independant ridge-regularized learning problems. On the other hand, various non-separable joint penalties can enforce various forms of similarity among the w_t. In the remaining of this section, some existing penalties together with the corresponding type of relation among the tasks they assume will be presented. While this presentation is made from a regularization point of view to be consistent with the contributions of this thesis, several ideas in this domain come from other frameworks like artificial neural networks, hierarchical bayes or non-parametric bayes. The corresponding models will be presented with their regularization-based counterpart.

Context

The idea of leveraging across several estimation problem long has been considered in econometrics and statistics literature. Zellner (1962) and Srivastava and Dwivedi (1979) studied *seemingly unrelated regression* which specifically consists in jointly solving several linear regressions, which is shown to be more efficient than individual solving when the residual terms are correlated. Brown and Zidek (1980) proposed a multivariate ridge regression models which extends the ridge regression to the case where several outputs are associated with each data points.

In the machine learning literature, this idea first appeared as the *bias learning* (Baxter, 1996a,b, 2000), or *learning to learn* (Baxter, 1997; Thrun and Pratt, 1998) problem. In these models, the bias denotes the hypotheses space in which the classification function can be chosen and the idea is to learn which hypotheses space is optimal from an hypotheses space family and several related learning problems. In practice, this idea was implemented by artificial neural networks with several output neurons corresponding to the different functions, and in which a hidden layer with a given architecture is shared by all the outputs. Optimizing the weights between the inputs and this hidden layer across the tasks results in learning a common low dimension representation, *i.e.* implicitly a hypothesis space. Caruana (1993, 1997) pioneered such networks and gave the denomination *multitask learning* to this approach.

Related problems include recommender systems or *collaborative filtering* (Breese et al., 1998; Heckerman et al., 2000), where the goal is to predict new products that a customer could like based on his previous purchases or ratings and those of other customers. In marketing, *conjoint analysis* (Green and Srinivasan, 1978; Chapelle and Harchaoui, 2005) tries to identify which features a new product should have based on the trade-off made by several customers. Multi-task learning is also related to *transfer learning* (Pan and Yang, 2008), where the goal is to apply a classifier learned on a learning task with a given distribution and set of features to a related task with a possibly different distribution and feature space.

Multi-task learning was studied from a learning theory point of view in Baxter (1997, 2000) and Ben-David and Schuller (2003). Baxter (2000) and Ben-David and Schuller (2003) extended results of Vapnik (1998), using a notion of extended VC-dimension to derive bounds on the generalization error of several tasks learned simultaneously, while Baxter (1997) analyzed multi-task learning in an information theoretic framework.

Task relatedness

Recent contributions in machine learning have built on these ideas to propose new models which take into account the fact that several learning tasks are related to improve their performances. Notwithstanding the different frameworks in which these models were proposed like artificial neural networks, hierarchical bayes or regularized risk minimization, an important distinction between them is the assumption they make about what is shared among the learning tasks. The most common assumptions are presented below :

Low variance assumption : A first class of models assumes that the linear functions of all the learning tasks are close to each others in ℓ_2 norm. This assumption stems in the seemingly unrelated regression of Zellner (1962) presented above, and the *mixed effect models* in statistics. In the regularization framework, Evgeniou and Pontil (2004) and Evgeniou et al. (2005) introduced the following penalty :

$$\Omega_{\text{variance}} = \alpha \sum_{t=1}^{T} \|w_t\|_2^2 + (1-\alpha) \sum_{t=1}^{T} \|w_t - \bar{w}\|_2^2, \tag{1.25}$$

for $\alpha \in [0, 1]$, where $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$. When used to regularize a risk minimization problem, Ω_{variance} penalizes more the functions $(w_1, ..., w_T)$ which are far from their mean in average thereby favoring solutions such that all the w_t are close to each others. The first term of the penalty is a regular individual ridge penalty, and the hyperparameter α trades off between individual ℓ_2 regularity and low variance among the w_t.

The hierarchical Gaussian process model proposed by Yu et al. (2005), with one Gaussian process by task and a Gaussian process prior whose parameters are learned across the tasks, implicitly makes the same assumption : the mean function accounts for the common effects shared by all the learning tasks, and the individual functions for the specific effects of each task. The Gaussian distribution on the specific task enforces a low variance of the individual functions around their mean, and the Gaussian prior on the mean enforces a global regularity. Yu et al. (2007) extended this model to t-processes, which are similar to Gaussian processes but have heavier tails, for robust multi-task learning.

Shared low-dimension representation assumption : Another common assumption is that the linear functions of all the learning tasks live in the same low-dimension linear subspace. This is equivalent to assuming that all the linear classifiers can be expressed in terms of a few variables given by linear combinations of the original descriptors. If this assumption is true, forcing all the functions to belong to the same subspace can guide the learning process and provide an insight of which descriptors best describe the problem. An interesting difference with the low-variance prior is that under this assumption, two linear classifiers can be anti-correlated and still live in the same linear subspace, whereas their ℓ_2 distance becomes maximal.

Early neural-networks approaches to multi-task learning (Caruana, 1993, 1997) are based on this assumption since the way they share information among the tasks is by learning a low-dimension representation in the hidden layer shared by all the tasks.

In the context of collaborative filtering, the most basic setting would only provide the output y_{it} (*e.g.* a rating) for each task t (*e.g.* a customer) and some point x_i (*e.g.* an object). Typically, this output matrix is very incomplete, *e.g.*, only the ratings of certain objects by certain users are available. In this case, searching for a *low-rank decomposition* of the output matrix $Y = UV^\top$, $U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{T\times k}$ is equivalent to finding a k-dimension space in which each line u_i of U describes the corresponding point x_i and each line v_t of V contains the linear prediction function of the corresponding task (Srebro and Jaakkola, 2003). However, the rank constraint is non-convex, and Srebro and Shraibman (2005); Srebro et al. (2005) propose to relax it by the trace norm, defined by :

$$\Omega_{\text{trace}}(W) = \sum_{k=1}^{\min(T,p)} |\sigma_k(W)|, \tag{1.26}$$

where $\sigma_k(W)$ denotes the k-th largest singular value of W. Therefore, Ω_{trace} is the ℓ_1 norm of the spectrum of the matrix and relaxes the rank penalty which would constrain the ℓ_0 norm of the spectrum, like the usual ℓ_1 norm relaxes the usual ℓ_0 norm. In practice, the trace penalty indeed leads to low-rank matrices (Fazel et al., 2001). More generally, when the data x_i are described by features, penalizing the joint risk minimization problem by the trace norm of W is equivalent to making the (relaxed) assumption that all the linear classifiers belong to the same low-dimension linear subspace. Indeed, if there are more descriptors than tasks, constraining the rank of W will impose that all the w_t can be expressed in a small shared basis. Another insight on the effect of the trace-norm constraint is given by the following equivalent formulation :

$$\Omega_{\text{trace}}(W) = \min_{U,V,\, W=UV} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right), \tag{1.27}$$

where $\|.\|_F^2$ denotes the squared Frobenius matrix norm, *i.e.*, the sum of the squared elements of the matrix. For a loss function $L(w_t) = \tilde{L}(X^t w_t)$ which only depends on w_t through dot products with the data points of the task X^t, the joint minimization problem therefore writes :

$$\min_{W} \sum_{t=1}^{T} \tilde{L}(X^t w_t) + \lambda \Omega_{\text{trace}}(W) \tag{1.28}$$

$$= \min_{W,U,V,\, W=UV} \sum_{t=1}^{T} \tilde{L}(X^t w_t) + \lambda \frac{1}{2} \left(\|U\|_F^2 + \|V\|_F^2 \right) \tag{1.29}$$

$$= \min_{U,V} \sum_{t=1}^{T} \tilde{L}(X^t U v_t) + \lambda \frac{1}{2} \left(\|U\|_F^2 + \|V\|_F^2 \right), \tag{1.30}$$

where v_t are the columns of V. In other words, penalizing by the trace norm can be seen as learning jointly a linear map U of the original descriptors across the tasks and linear functions v_t in this shared feature space. Because of the relaxation, the number of shared features is not explicitly controlled and the complexity of the mapping is only penalized through the Frobenius norm of U. The Frobenius norm of V, $\|V\|_F^2 = \sum_{t=1}^{T} \|v_t\|_2^2$ controls the ℓ_2 regularity of the linear functions in the shared space. This approach was used in Argyriou et al. (2007), and extended in Argyriou et al. (2008b) to a more general class of spectral functions. Consistency results for the trace norm minimization were given in Bach (2008b).

Alternatively, Ando et al. (2005) propose a model in which the linear classifier of each task is a combination of a ridge-regularized component in the original feature space and a component in another feature space. This other feature space is made of h normalized and orthogonal linear combinations of the original features, where h is an hyperparameter of the model :

$$\min_{\{w-t,v_t\}_{t=1,\dots,T},\Theta} \sum_{t=1}^{T} L(w_t + \Theta^\top v_t) + \lambda_t \|w_t\|_2^2, \tag{1.31}$$
$$\text{s.t. } \Theta\Theta^\top = I_h,$$

which they optimize by an efficient alternating optimization scheme. The problem is convex in w_t, v_{tt} and in Θ, but not jointly convex because of the $\Theta^\top v_t$ product. A convex relaxation was proposed in Chen et al. (2009).

Shared sparsity pattern assumption : A last family of approaches makes the assumption that all the linear functions are sparse, and that the variables in the model are the same for all of them. Like for the previous prior family, these methods assume that the linear functions of all the tasks live in a low-dimension space, but they further assume that this space is a restriction of the original feature space to certain variables.

A first model enforcing this prior was proposed in Jebara (2004) in the maximum entropy discrimination framework (Jaakkola et al., 1999). In the framework of regularized learning, it is possible to enforce this prior by simply using a group-lasso penalty where the groups are each variable considered across the tasks (Obozinski et al., 2009) :

$$\Omega_{\text{joint}}(W) = \|W\|_{1,2} = \sum_{j=1}^{d} \|w^{(j)}\|_2, \tag{1.32}$$

where $w^{(j)} = (w_1^{(j)}, \ldots, w_T^{(j)})$ is the vector formed by the j-th parameter across the tasks. As argumented in Section 1.4.2, this penalty will favor solutions such that several $w^{(j)}$ are 0, resulting in w_t which are sparse with a joint sparsity pattern.

A ℓ_1/ℓ_∞ version of this penalty was proposed in Turlach et al. (2005), studied in Tropp et al. (2006) for sparse appoximations in signal processing and for regression, and in Zhang et al. (2008) for classification. In the non-parametric Bayes framework, a generalization of the SpAM was proposed in Liu et al. (2009) using the ℓ_1/ℓ_∞ norm. The model selection consistency was studied for the ℓ_1/ℓ_2 regression model in Obozinski et al. (2008). Finally, Lounici et al. (2009) proposed a non-asymptotic bound on the ℓ_1/ℓ_2 error of the corresponding estimate.

Of course, these assumptions do not exclude each others. For example, some hierarchical Bayes models (Heskes, 2000; Bakker and Heskes, 2003) combine the first two assumptions. Like the neural networks of Caruana (1997), they learn a common low dimension representation but in addition, they impose a common Gaussian prior to the task parameters. Since the prior is learned jointly by taking its maximum of likelihood estimator across the tasks, this controls the variance of the task parameters around the cross-task mean (as in the non-parametric counterpart of Yu et al. (2005)). Abernethy et al. (2008), which is presented later in this section, explicitely combines penalties (1.25) and (1.26) among others.

An alternative approach based on covariate shift (Shimodaira, 2000; Bickel et al., 2007) was proposed by Bickel et al. (2008). Instead of jointly minimizing the empirical risks of all the tasks under some relatedness constraint, they learn for each task t the probability $p(t|x_i, y_i)$ of a data point and its output to be generated from the task, and then learn a model for the task using all the points weighted by this probability. They show that this is equivalent to sampling the training points from the task distribution instead of the mixture of tasks one.

Clustered multi-task

Some models consider that this low variance assumption only holds within some *clusters* of tasks : Bakker and Heskes (2003) propose a version of their hierarchical Bayesian model where the prior is a mixture of Gaussian instead of a single Gaussian. In the non-parametric Bayesian framework, Xue et al. (2007b) used a Dirichlet process prior, which is known to have a clustering effect. In Xue et al. (2007a) and Dunson et al. (2008), the idea was extended to the case where the clustering is not the same for the different parameters. Deodhar and Ghosh (2007) propose to alternate between clustering the tasks and learning a multi-task model using this clustering. Daume (2009) uses a hierarchy structure instead of a clustering.

Pairwise learning

As suggested in Evgeniou et al. (2005) and detailed in Appendix A.1, it is possible to express the multi-task learning formulation (1.25) by considering pairs formed by one task and one individual as the inputs of any kernel method, using a product kernel of the form

$$K((x,t),(x',t')) = K_{\text{data}}(x,x')K_{\text{task}}(t,t'), \tag{1.33}$$

for the particular choice of K_{task}.

If some prior information about which tasks are similar is available under the form of a kernel or of task descriptors, the problem becomes symmetric, *i.e.*, the problem is to learn a function which discriminates between "positive" pairs (the pairs formed by an x and a t

such that x is a positive data for task t) from the negative ones. Using a product of kernel and a regular kernel method with an ℓ_2 penalty, this approach was described in Bonilla et al. (2007), is known as *pairwise learning*, and has been used with some success in bioinformatics (Martin et al., 2005; Ben-Hur and Noble, 2005; Vert et al., 2007). In terms of prior, using this ℓ_2 penalty in the joint space means that we expect the function to be smooth across the pairs. Slight changes in either x or t (in terms of the K_{data} and K_{task} metrics) should not result in large variations of the functions.

More generally, Abernethy et al. (2008) cast this problem of learning a discriminative function on pairs as an operator estimation problem, and propose to regularize it by a convex combination of the ℓ_2 (Frobenius) norm of the operator and its trace norm. As for the ℓ_2 penalty, the interpretation of the trace norm contraint generalizes to the joint feature space, to the idea that the true function should be expressed in terms of few linear combinations of the joint features. Since a classical results on tensor products states that :

$$\begin{aligned} K_{\text{data}}(x,x')K_{\text{task}}(t,t') &= \Phi_{\text{data}}(x)^\top \Phi_{\text{data}}(x') \times \Phi_{\text{task}}(t)^\top \Phi_{\text{task}}(t') \\ &= \left(\Phi_{\text{data}}(x) \otimes \Phi_{\text{task}}(t)\right)^\top \left(\Phi_{\text{data}}(x') \otimes \Phi_{\text{task}}(t')\right), \end{aligned} \tag{1.34}$$

these joint features are products of the data features and the task features.

In the non-parametric Bayesian framework, Bonilla et al. (2008) use a Gaussian process model to learn a function in the joint function space, and learn an optimal representation for the tasks.

Structured output learning

A problem very close to multi-task learning and in which the same type of prior can be used is *structured output learning* (Taskar et al., 2004; Tsochantaridis et al., 2005). In structured output learning, each data point x_i is associated to an output y_i which instead of just being a binary variable like in classification or a real number like in regression has its own structure and can be described with some features, like the input. Typical applications include natural language processing, when one wants to predict the syntactic tree of a given sentence, or bioinformatics when one wants to predict the splicing of a protein in some process. Multi-class learning can also be formulated as a basic form of structured output learning, where the structure which has to be predicted is a vector of the canonical basis indicating the class. Formally, the only difference with the multi-task learning problem is that each input is positive for a unique structure, whereas in multi-task, each data point can be positive for several tasks. Therefore, both problems can be formulated in a joint data-structure or data-task feature space, but instead of the sum of risks which is minimized in multi-task (1.24), the structured output problem is to minimize :

$$\sum_{i=1}^{n} \max_{y \neq y_i} (w^\top \Phi(x_i, y) - w^\top \Phi(x_i, y_i)), \tag{1.35}$$

that is, to make sure that w gives a higher score to the right pair (x_i, y_i) than to any other pair formed by x_i with another y. Several relaxations of this risk function following the same large margin principle as the SVM were proposed in Tsochantaridis et al. (2005). Once such a risk is chosen, any regularizer that was presented for the multi-task can be used for the structured output problem. In particular, Amit et al. (2007) used the trace norm in the context of multi-class learning.

1.5 Contributions of this thesis

1.5.1 Pairwise learning for interaction prediction

Both vaccine and drug design involve a screening step, whose purpose is to identify within a very large list of potential binders (peptides for the vaccines, small molecules for the drugs) which ones are the most likely to actually bind a given target (MHC molecule for the vaccines, proteins for the drugs). A common way to formulate this problem is as a binary classification problem for each target, where the positive data are the binders and the negative data are the non-binders for this target. Note that another dominant way to address this screening problem is to use *docking* methods which use the 3D-structure of the target to determine how well each candidate binds the target.

One contribution of this thesis, when binding data is available for several targets, is to reformulate this problem as a binary classification on binder-target pairs, where the positive points are the pairs which interact, *i.e.*, which involve a given target and one of its ligands, and the negative points are the pairs consisting of non-interacting elements. The purpose of this reformulation is to improve the performances for targets which have little or no available training data. The central idea of the method is to find descriptors for the pairs, which reflect its potential to interact or not. These descriptors are built from descriptors for the targets in addition to the traditional descriptors for the ligands. We propose several target descriptor both for MHC molecules (vaccine design) and proteins (drug design). This was detailed in Jacob and Vert (2008a,b) and Jacob et al. (2008), which form Chapter 2.

Note that the framework as it is proposed in this thesis only considers *asymetric* pairs, in the sense that the two elements in the pair do not have the same nature. Problems involving symmetric pairs, *e.g.*, protein-protein interaction prediction, require a slightly different handling, like counting each pair twice (one time in each order), or adding a symmetrization term in the resulting scalar product.

Experimentally, this method outperformed state-of-the-art prediction methods for vaccine and drug design on various benchmarks

1.5.2 Clustered multi-task learning

Multi-task learning involves considering several related problems simultaneously, with the hope of improving performance by sharing information across these problems or “tasks”. A common strategy as outlined in Section 1.4.3 is to penalize the variance across the classification functions of all the tasks, which can help guide learning when little data is available.

In more realistic settings, it may be that certain inference problems are related but others are arbitrarily different. In such cases, penalizing the overall variance may harm the performance as it would force classifiers of very different problems to be close to each others. If this clustering structure on the learning tasks were known, one would like to penalize the variance only within clusters of related problems.

As these clusters are generally unknown *a priori*, we have proposed in Jacob et al. (2009a) a criterion which penalizes the variance of functions within clusters, and optimized jointly with respect to both the classification functions and clustering. However, clustering is by nature a non-convex problem, because it consists of finding the optimum assignment of points (here the linear functions corresponding to the tasks) within a discrete set of possible assignements. We have therefore proposed a convex relaxation, which we show improved the prediction performances.

This work is presented in Chapter 3.

1.5.3 Structured priors for expression data analysis

A well known problem in bioinformatics is to predict the class of a tumour from gene expression measurements with microarrays, and simultaneously select a small number of genes to establish a predictive signature. Selecting a few genes that either belong to the same functional groups (where the groups are given *a priori* and may overlap *e.g.*, biological pathways) or tend to be connected to each other in a given biological network, may lead to increased interpretability of the signature and potentially to better performance when little data is available.

To this end, we proposed and studied in Jacob et al. (2009b) a new penalty which generalizes the ℓ_1/ℓ_2 norm to overlapping groups, and cast the problem of selecting connected covariates in a graph as the problem of selecting a union of overlapping groups, with adequate definition of groups. More precisely, this method can be used in cases where either groups of covariates are given (potentially with overlap between the groups) and we wish to estimate a model whose support is a union of groups, or when a graph with covariates as vertices is given and we wish to estimate a model whose support contains covariates which tend to be connected to each other on the graph.

We illustrated the behavior of this method on a well known benchmark of breast cancer tumors. When used with canonical pathways, it led to solutions involving much less pathways, and when used with biological graphs, to solutions which were more connected.

This work is presented in Chapter 4.

1.5.4 List of articles published during the thesis

- L. Jacob and J.-P. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, Feb 2008a.

- L. Jacob and J.-P. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008b.
- L. Jacob, B. Hoffmann, B. Stoven, and J.-P. Vert. Virtual screening of GPCRs: an *in silico* chemogenomics approach. *BMC Bioinformatics*, 9:363, 2008.
- J.-P. Vert and L. Jacob. Machine learning for in silico virtual screening and chemical genomics: New strategies. *Combinatorial Chemistry & High Throughput Screening*, 11(8):677–685, September 2008.
- L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems 21*, pages 745–752. MIT Press, 2009a.
- L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlaps and graph lasso. In *ICML'09 Proceedings of the 26th international conference on Machine learning*, 2009b.

Chapter 2

Pairwise learning for interaction prediction

This chapter presents the work which was published in Jacob and Vert (2008a,b); Jacob et al. (2008) and Vert and Jacob (2008). The order of the sections as well as the content of some of them has been modified to give consistency to the chapter.

2.1 Interaction prediction in computational biology

In vaccine design, immunologists are interested in having accurate predictions of which peptides bind to MHC molecules. This is crucial to discover which peptides of a pathogen can trigger an immunological response and therefore give protection against the given pathogen. Different MHC alleles bind different peptides.

In drug discovery, biologists try to find small molecules which interact with given therapeutical targets such as enzymes or GPCRs. The goal is to use these molecules as drugs to regulate the target whose abnormal behavior causes a disease.

In both cases, traditional prediction methods build one classifier for each target (MHC molecule or drug target) separately. Using a kernel-based approach which casts the problem as predicting whether each pair, *e.g.* (peptide,MHC) or (molecule,target) interacts or not (Vert and Jacob, 2008), we obtained significant prediction improvement in accuracy for the targets with few known binders. We have proposed some specific kernels for each problem, and shown that this approach improves the prediction accuracy for both the MHC (Jacob and Vert, 2008a) and drug discovery problems (Jacob and Vert, 2008b). In Jacob et al.

(2008), we propose some additional kernels for the GPCR case.

2.2 Kernel methods for interaction prediction

We formulate the typical *in silico* interaction prediction problem as the following learning problem : given a collection of n target/ligand pairs $(t_1, l_1), \dots (t_n, l_n)$ known to interact or not, estimate a function $f(t, l)$ that would predict whether any ligand l binds to any target t. Here, the words ligand and target are taken in their most general meaning : in the *in silico* chemogenomics problem, the ligands will be small molecules and the target will be proteins whose activity has to be modulated whereas in the case of vaccine design, the ligands will be short peptides and the target will be MHC molecules for different alleles. In this section we propose a rigorous and general framework to solve this problems, building on recent developments of kernel methods in bio- and chemoinformatics.

2.2.1 From single-target screening to interaction prediction

Much effort in chemoinformatics has been devoted to the more restricted problem of mining the chemical space for interaction with a single drug target, and in immunoinformatics the space of the potential epitopes for a single MHC allele, using a training set of ligands $l_1, \dots, l_n$ known to interact or not with the target. Machine learning approaches, such as artificial neural networks (ANN) or support vector machines (SVM), often provide competitive models for such problems. The simplest linear models start by representing each ligand l by a vector representation $\Phi(l)$, before estimating a linear function $f_t(l) = w_t^\top \Phi(l)$ whose sign (positive or negative) is used to predict whether or not the ligand l interacts with the target t. The weight vector w_t is typically estimated based on its ability to correctly predict the classes of molecules in the training set.

The *in silico* interaction prediction problem is more general because data involving different targets are available to train a model which must be able to predict interactions between any ligand and any target. In order to extend the previous machine learning approaches to this setting, we need to represent a *pair* (t, l) of target t and ligand l by a vector $\Phi(t, l)$, then estimate a linear function $f(t, l) = w^\top \Phi(t, l)$ whose sign is used to predict whether or not l can bind to t. As before the vector w can be estimated from the training set of interacting and non-interacting pairs, using any linear machine learning algorithm.

To summarize, we propose to cast the *in silico* interaction prediction problem as a learning problem in the ligand-target space thus making it suitable to any classical linear machine learning approach as soon as a vector representation $\Phi(t, l)$ is chosen for target/ligand pairs. We propose in the next sections a systematic way to design such a representation.

2.2.2 Vector representation of target/ligand pairs

A large literature in chemoinformatics has been devoted to the problem of representing a molecule c by a vector $\Phi_{lig}(c) \in \mathbb{R}^{d_c}$, e.g., using various molecular descriptors (Todeschini and Consonni, 2002). These descriptors encode several features related to the physico-chemical and structural properties of the molecules, and are widely used to model interactions between the small molecules and a single target using linear models described in the previous section (Gasteiger and Engel, 2003). On the other hand, much work in computational biology has been devoted to the construction of descriptors for genes and proteins, in order to represent a given protein t by a vector $\Phi_{tar}(t) \in \mathbb{R}^{d_t}$. The descriptors typically capture properties of the sequence or structure of the protein, and can be used to infer models to predict, *e.g.*, the structural or functional class of a protein. Similarly for the epitope prediction problem, it is possible to design descriptors for small peptides which would be informative of their ability to bind a given molecule, and descriptors of MHC molecules which would be related to their epitope repertoire. A more detailed presentation of these descriptors will be given in the next sections.

For our *in silico* interaction prediction problem we need to represent each pair (l, t) of ligand and target by a single vector $\Phi(l, t)$. In order to capture interactions between features of the ligand and of the target that may be useful predictors for the interaction between l and t, we propose to consider features for the pair (l, t) obtained by multiplying a descriptor of l with a descriptor of t. Intuitively, if for example the descriptors are binary indicators of specific structural features in each ligand and target, then the product of two such features indicates that both the ligand and the target carry specific features, which may be strongly correlated with the fact that they interact. More generally, if a molecule l

is represented by a vector of descriptors $\Phi_{lig}(l) \in \mathbb{R}^{d_l}$ and a target protein by a vector of descriptors $\Phi_{tar}(t) \in \mathbb{R}^{d_t}$, this suggests to represent the pair (l, t) by the set of all possible products of features of l and t, i.e., by the tensor product:

$$\Phi(l, t) = \Phi_{lig}(l) \otimes \Phi_{tar}(t) \,. \tag{2.1}$$

Remember that the tensor product in (2.1) is a $d_l \times d_t$ vector whose (i, j)-th entry is exactly the product of the i-th entry of $\Phi_{lig}(l)$ by the j-th entry of $\Phi_{tar}(t)$. This representation can be used to combine in an algorithmic way any vector representation of ligands with any vector representation of targets, for the purpose of *in silico* interaction prediction or any other task involving pairs of ligand/target. A potential issue with this approach, however, is that the size of the vector representation for a pair may be prohibitively large for practical computation and storage. For example, using a vector of molecular descriptors of size 1024 for molecules and representing a protein by the vector of counts of all 2-mers of amino-acids in its sequence ($d_t = 20 \times 20 = 400$) results in more than 400k dimensions for the representation of a pair. In order to circumvent this issue we now show how kernel methods such as SVM can efficiently work in such large spaces.

2.2.3 Kernels for target/ligand pairs

SVM is an algorithm to estimate linear binary classifiers from a training set of patterns with known class (Boser et al., 1992; Vapnik, 1998). A salient feature of SVM, often referred to as the *kernel trick*, is its ability to process large- or even infinite-dimensional patterns as soon as the inner product between any two patterns can be efficiently computed. This property is shared by a large number of popular linear algorithms, collectively referred to as *kernel methods*, including for example algorithms for regression, clustering or outlier detection (Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004).

In order to apply kernel methods such as SVM for *in silico* interaction prediction, we therefore need to show how to efficiently compute the inner product between the vector representations of two ligand/target pairs. Interestingly, a classical property of tensor products allows us to factorize the inner product between two tensor product vectors as follows:

$$\begin{aligned}&\left(\Phi_{lig}(l) \otimes \Phi_{tar}(t)\right)^\top \left(\Phi_{lig}(l') \otimes \Phi_{tar}(t')\right)\\&= \Phi_{lig}(l)^\top \Phi_{lig}(l') \times \Phi_{tar}(t)^\top \Phi_{tar}(t') \,.\end{aligned} \tag{2.2}$$

This factorization dramatically reduces the burden of working with tensor products in large dimensions. For example, in our previous example where the dimensions of the small molecule and proteins are vectors of respective dimensions 1024 and 400, the inner product in $> 400k$ dimensions between tensor products is simply obtained from (2.2) by computing two inner products, respectively in dimensions 1024 and 400, before taking their product.

Even more interestingly, this reasoning extends to the case where inner products between vector representations of ligands and targets can themselves be efficiently computed with the help of *positive definite* kernels (Vapnik, 1998), as explained in the next sections. Positive definite kernels are linked to inner products by a fundamental result (Aronszajn, 1950): the kernel between two points is equivalent to an inner product between the points mapped to a Hilbert space uniquely defined by the kernel. Now by denoting

$$K_{ligand}(l, l') = \Phi_{lig}(l)^\top \Phi_{lig}(l'), \tag{2.3}$$

$$K_{target}(t, t') = \Phi_{tar}(t)^\top \Phi_{tar}(t'), \tag{2.4}$$

we obtain the inner product between tensor products by:

$$K\left((c, t), (c', t')\right) = K_{target}(t, t') \times K_{ligand}(c, c'). \tag{2.5}$$

In summary, as soon as two kernels K_{ligand} and K_{target} corresponding to two implicit embeddings of the ligand and target spaces in two Hilbert spaces are chosen, we can solve the *in silico* interaction prediction problem with an SVM (or any other relevant kernel method) using the product kernel (2.5) between pairs. The particular kernels K_{ligand} and K_{target} should ideally encode properties related to the ability of similar ligands to bind similar targets. We review in the next two sections possible choices for such kernels.

2.3 Kernels for epitope prediction

2.3.1 Kernels for peptides

We consider in this work mainly peptides made of 9 amino acids, although extensions to variable-length peptides poses no difficulty in principle (Salomon and Flower, 2006). The classical way to represent these 9-mers as fixed length vectors is to encode the letter at each position by a 20-dimensional binary vector indicating which amino acid is present, resulting in a 180-dimensional vector representations. In terms of kernel, the inner product between two peptides in this representation is simply the number of letters they have in common at the same positions, which we take as our baseline kernel:

$$K_{linseq}(x, x') = \sum_{i=1}^{l} \delta(x[i]x'[i]),$$

where l is the length of the peptides (9 in our case), $x[i]$ is the i-th residue in x and $\delta(x[i]x'[i])$ is 1 if $x[i] = x'[i]$, 0 otherwise.

Alternatively, several authors have noted that nonlinear variants of the linear kernel can improve the performance of SVM for epitope prediction (Dönnes and Elofsson, 2002; Zhao et al., 2003; Bhasin and Raghava, 2004b). In particular, using a polynomial kernel of degree p over the baseline kernel is equivalent, in terms of feature space, to encoding p-order interactions between amino acids at different positions. In order to assess the relevance of such non-linear extensions we tested a polynomial kernel of degree 5, *i.e.*,

$$K_{seq5}(x, x') = (K_{linseq}(x, x') + 1)^5.$$

In order to limit the risk of overfitting to the benchmark data we restrict ourselves to the evaluation of the baseline linear kernel and its nonlinear polynomial extension. Designing a specific peptide kernel for epitope prediction, *e.g.*, by weighting differently the positions known to be critical in the MHC-peptide complex, is however an interesting research topic that could bring further improvements in the future.

2.3.2 Kernels for MHC molecules

Although the question of kernel design for peptides has been raised in previous studies involving SVM for epitope prediction (Dönnes and Elofsson, 2002; Zhao et al., 2003; Bhasin and Raghava, 2004b; Salomon and Flower, 2006), the question of kernel design for alleles is new to our knowledge. We tested several choices that correspond to previously published approaches:

- The *Dirac* kernel is:

  $$K_{Dirac}(a,a') = \begin{cases} 1 & \text{if } a = a' \,, \\ 0 & \text{otherwise.} \end{cases}$$

  With the Dirac kernel, no information is shared across alleles and the SVM learns one model for each allele independently from the others. Therefore this corresponds to the classical setting of learning epitope prediction models per allele with SVM.

- The *uniform* kernel is:

  $$K_{uniform}(a,a') = 1 \text{ for all } a, a' \,.$$

  With this kernel all alleles are considered the same, and a unique model is created by pooling together the data available for all alleles.

- The *multitask* kernel is:

  $$K_{multitask}(a,a') = K_{dirac}(a,a') + K_{uniform}(a,a') \,.$$

  As explained in the previous section and in Evgeniou et al. (2005) this is the simplest way to train different but related models. The SVM learns one model for each allele, using known epitopes and non-epitopes for the allele, but using also known epitopes and non-epitope for all other alleles with a smaller contribution. The training peptides are shared uniformly across different alleles.

- The *supertype* kernel is

  $$K_{supertype}(a, a') = K_{multitask} + \delta_s(a, a')\,,$$

  where $\delta_s(a, a')$ is 1 if a and a' are in the same supertype, 0 otherwise. As explained in the previous section this scheme trains a specific models for each allele using training peptides from different alleles, but here the training peptides are more shared across alleles withing a supertype than across alleles in different supertypes. This is used by Heckerman et al. (2007), without the kernel formulation, to train a logistic regression model.

Heckerman et al. (2007) show that the supertype kernel generally improves the performance of logistic regression models compared to the uniform or Dirac kernel. Intuitively it seems to be an interesting way to include prior knowledge about alleles. However, one should be careful since the definition of supertypes is based on the comparison of epitopes of different alleles, which suggests that the supertype information might be based on some information used to assess the performance of the method in the benchmark experiment. In order to overcome this issue, and illustrate the possibilities offered by our formulation, we also tested a kernel between alleles which tries to quantify the similarity of alleles without using known epitope information. For that purpose we reasoned that alleles with similar residues at the positions involved in the peptide binding were more likely to have similar epitopes, and decided to make a kernel between alleles based on this information. For each locus we gathered from Doytchinova et al. (2004) the list of positions involved in the binding site of the peptide (Table 2.1). Taking the union of these sets of positions we then represented each allele by the list of residues at these positions, and used a polynomial kernel of degree 7 to compare two lists of residues associated to two alleles, *i.e*,

$$K_{bsite7}(a, a') = \left(\sum_{i \in \text{bsite}} \delta(a[i]a'[i]) + 1\right)^7,$$

where bsite is the set of residues implied in the binding site for one of the three allele groups HLA-A, B, C, $a[i]$ is the i-th residue in a and $\delta(a[i]a'[i])$ is 1 if $a[i] = a'[i]$, 0 otherwise.

2.4 Kernels for compound prediction

2.4.1 Kernels for small molecules

The problem of explicitly representing and storing small molecules as finite-dimensional vectors has a long history in chemoinformatics, and a multitude of molecular descriptors have been proposed (Todeschini and Consonni, 2002). These descriptors include in particular physicochemical properties of the molecules, such as its solubility or logP, descriptors derived from the 2D structure of the molecule, such as fragment counts or structural fingerprints, or descriptors extracted from the 3D structure (Gasteiger and Engel, 2003). Each classical fingerprint vector and vector representation of molecules define an explicit "chemical space" in which each molecule is represented by a finite-dimensional vector, and these vector representations can obviously be used as such to define kernels between molecules (Azencott et al., 2007). Alternatively, some authors have recently proposed some kernels that generalize some of these sets of descriptors and correspond to inner products between large- or even infinite-dimensional vectors of descriptors. These descriptors encode, for example, the counts of an infinite number of walks on the graph describing the 2D structure of the molecules (Kashima et al., 2004; Gärtner et al., 2003; Mahé et al., 2005), or various features extracted from the 3D structures (Mahé et al., 2006; Azencott et al., 2007). For a more detailed review of the kernels for small molecule, we refer the reader to section 1.3.3.

In this study we select two existing kernels, encoding respectively 2D and 3D structural information of the small molecules, and propose a new 3D kernel:

- *The 2D Tanimoto kernel.* Our first set of descriptors is meant to characterize the 2D structure of the molecules. For a small molecule m, we define the vector $\Phi_{mol}(m)$ as the binary vector whose bits indicate the presence or absence of all linear graph of length u or less as subgraphs of the 2D structure of l. We chose $u = 8$ in our experiment, i.e., characterize the molecules by the occurrences of linear subgraphs of length 8 or less, a value previously observed to give good results in several virtual screening tasks (Mahé et al., 2005). Moreover, instead of directly taking the inner product between vectors as in (2.3), we use the Tanimoto kernel:

$$K_{ligand}(l, l') = \frac{\Phi_{lig}(l)^\top \Phi_{lig}(c')}{\Phi_{lig}(l)^\top \Phi_{lig}(l) + \Phi_{lig}(l')^\top \Phi_{lig}(l') - \Phi_{lig}(l)^\top \Phi_{lig}(l')}, \tag{2.6}$$

which was proven to be a valid inner product by Ralaivola et al. (2005), giving very competitive results on a variety of QSAR or toxicity prediction experiments.

- *3D pharmacophore kernel* While 2D structures are known to be very competitive in ligand-based virtual screening (Azencott et al., 2007), we reasoned that some specific 3D conformations of a few atoms or functional groups may be responsible for the interaction with the target. Thus, we decided to test descriptors representing the presence of potential 3-point pharmacophores. For this, we used the 3D pharmacophore kernel proposed by Mahé et al. (2006), that generalizes 3D pharmacophore fingerprint descriptors. This approach implies the choice of a 3D conformer for each molecule. In absence of sufficient data available for bound ligands in GPCR structures, we chose to build a 3D version of the ligand base in which molecules are represented in an estimated minimum energy conformation. For each of the 2446 retained ligands, 25 conformers were generated with the Omega program (OpenEye Scientific Software) using standard parameters, except for a 1 RMSD clustering of the conformers, instead of the 0.8 default value. A 3D ligand base was generated by keeping the conformer of lowest energy for each ligand. Partial charges were calculated for all atoms using the molcharge program (OpenEye Scientific Software) with standard parameters. This ligand base was then used to calculate a 3D pharmacophore kernel for molecules (Mahé et al., 2006).

We used the freely and publicly available *ChemCPP*[1] software to compute the 2D and 3D pharmacophore kernel.

2.4.2 Kernels for protein targets

SVM and kernel methods are also widely used in bioinformatics (Schölkopf et al., 2004), and a variety of approaches have been proposed to design kernels between proteins, ranging

[1] Available at `http://chemcpp.sourceforge.net`.

from kernels based on the amino-acid sequence of a protein (Jaakkola et al., 2000; Leslie et al., 2002; Tsuda et al., 2002b; Leslie et al., 2004; Vert et al., 2004; Kuang et al., 2005; Cuturi and Vert, 2005) to kernels based on the 3D structures of proteins (Dobson and Doig, 2005; Borgwardt et al., 2005; Qiu et al., 2007) or the pattern of occurrences of proteins in multiple sequenced genomes (Vert, 2002). A more detailed review of existing kernels for proteins is proposed in section 1.3.2.

These kernels have been used in conjunction with SVM or other kernel methods for various tasks related to structural or functional classification of proteins.

While any of these kernels can theoretically be used as a target kernel in (2.5), we investigate in this chapter a restricted list of specific kernels described below, aimed at illustrating the flexibility of our framework and testing various hypothesis.

- The *Dirac* kernel between two targets t, t' is:

  $$K_{Dirac}(t,t') = \begin{cases} 1 & \text{if } t = t' , \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

  This basic kernel simply represents different targets as orthonormal vectors. From (2.5) we see that orthogonality between two proteins t and t' implies orthogonality between all pairs (c, t) and (c', t') for any two small molecules c and c'. This means that a linear classifier for pairs (c, t) with this kernel decomposes as a set of independent linear classifiers for interactions between molecules and each target protein, which are trained without sharing any information of known ligands between different targets. In other words, using Dirac kernel for proteins amounts to performing classical learning independently for each target, which is our baseline approach.

- The *multitask* kernel between two targets t, t' is defined as:

  $$K_{multitask}(t,t') = 1 + K_{Dirac}(t,t') .$$

  This kernel, originally proposed in the context of multitask learning (Evgeniou et al., 2005), removes the orthogonality of different proteins to allow sharing of information. As explained in Evgeniou et al. (2005), plugging $K_{multitask}$ in (2.5) amounts to decomposing the linear function used to predict interactions as a sum of a linear function common to all targets and of a linear function specific to each target:

$$f(c,t) = w^\top \Phi(c,t) = w_{general}^\top \Phi_{lig}(c) + w_t^\top \Phi_{lig}(c)\,. \tag{2.8}$$

A consequence is that only data related to the the target t are used to estimate the specific vector w_t, while all data are used to estimate the common vector $w_{general}$. In our framework this classifier is therefore the combination of a target-specific part accounting for target-specific properties of the ligands and a global part accounting for general properties of the ligands across the targets. The latter term allows to share information during the learning process, while the former ensures that specificities of the ligands for each target are not lost.

- While the multitask kernel provides a basic framework to share information across proteins, it does not allow to weight differently how known interactions with a protein t should contribute to predict interactions with a target t'. Empirical observations underlying chemogenomics, on the other hand, suggest that molecules binding a ligand t are only likely to bind ligand t' similar to t in terms of structure or evolutionary history. In terms of kernels this suggest to plug into (2.5) a kernel for proteins that quantifies this notion of similarity between proteins, which can for example be detected by comparing the sequences of proteins. In order to test this approach, we therefore tested two commonly-used kernels between protein sequences: the mismatch kernel (Leslie et al., 2004), which compares proteins in terms of common short sequences of amino acids up to some mismatches, and the local alignment kernel (Vert et al., 2004) which measures the similarity between proteins as an alignment score between their primary sequences. In our experiments involving the mismatch kernel, we use the classical choice of 3-mers with a maximum of 1 mismatch, and for the datasets where some sequences were not available in the database, we added $K_{Dirac}(t,t')$ to the kernel (and normalized at 1 on the diagonal) in order to keep it valid.

- Alternatively we propose a new kernel aimed at encoding the similarity of proteins with respect to the ligands they bind. Indeed, for most major classes of drug targets such as the ones investigated in this study (GPCR, enzymes and ion channels), proteins have been organized into hierarchies that typically describe the precise functions of the proteins within each family. Enzymes are labeled with *Enzyme Commission numbers* (EC numbers) defined in International Union of Biochemistry and Molecular Biology (1992), that classify the chemical reaction they catalyze, forming a 4-level hierarchy encoded into 4 numbers. For example $EC\ 1$ includes oxidoreductases, $EC\ 1.2$ includes oxidoreductases that act on the aldehyde or oxo group of donors, $EC\ 1.2.2$ is a subclass of $EC\ 1.2$ with $NAD+$ or $NADP+$ as acceptor and $EC\ 1.2.2.1$ is a subgroup of enzymes catalyzing the oxidation of formate to bicarbonate. These number define a natural and very informative hierarchy on enzymes: one can expect that enzymes that are closer in the hierarchy will tend to have more similar ligands. Similarly, GPCRs are grouped into 4 classes based on sequence homology and functional similarity: the *rhodopsin* family (class A), the *secretin* family (class B), the *metabotropic* family (class C) and a last class regrouping more diverse receptors (class D). The KEGG database (Kanehisa et al., 2002) subdivides the large rhodopsin family in three subgroups (amine receptors, peptide receptors and other receptors) and adds a second level of classification based on the type of ligands or known subdivisions. For example, the rhodopsin family with amine receptors is subdivided into cholinergic receptors, adrenergic receptors, *etc*. This also defines a natural hierarchy that we could use to compare GPCRs. Finally, KEGG also provides a classification of ion channels. Classification of ion channels is a less simple task since some of them can be classified according to different criteria like voltage dependence or ligand-gating. The classification proposed by KEGG includes *Cys-loop superfamily, glutamate-gated cation channels, epithelial and related Na+ channels, voltage-gated cation channels, related to voltage-gated cation channels, related to inward rectifier K+ channels, chloride channels* and *related to ATPase-linked transporters* and each of these classes is further subdivided according for example to the type of ligands (*e.g.*, glutamate receptor) or to the type of ion passing through the channel (*e.g.*, Na+ channel). Here again, this hierarchy can be used to define a meaningful similarity in terms of interaction behavior.

For each of the three target families, we define the hierarchy kernel between two targets of the family as the number of common ancestors in the corresponding hierarchy plus one, that is,

$$K_{hierarchy}(t,t') = \langle \Phi_h(t), \Phi_h(t') \rangle,$$

where $\Phi_h(t)$ contains as many features as there are nodes in the hierarchy, each being set to 1 if the corresponding node is part of t's hierarchy and 0 otherwise, plus one feature constantly set to one that accounts for the "plus one" term of the kernel. One might not expect the EC classification to be a good similarity measure in terms of binding since it does not closely reflect evolutionary or mechanistic similarities except for the case of identical subclasses with different serial numbers. However, using the full hierarchy gave a better accuracy in our experiments. Even if the hierarchy itself is not fully relevant in this case, the improvement can be explained, on the one hand, by the multitask effect, *i.e.*, by the fact that we use the data from the target and the data from other targets with a smaller weight, and on the other hand by the fact that we give more weight to the enzymes with the same serial number than to the other enzymes.

- The *binding pocket* kernel. Because the protein-ligand recognition process occurs in 3D space in a pocket involving a limited number of residues, we tried to describe the GPCR space using a representation of this pocket. The difficulty resides in the fact that although the GPCR sequences are known, the residues forming this pocket and its precise geometry are *a priori* unknown. However, the two available X-Ray structures, together with mutagenesis data showed that the binding pockets are situated in a similar region for all GPCRs (Kratochwil et al., 2005). In order to identify residues potentially involved in the binding pocket of GPCRs of unknown structure studied in this work, we proceeded in several steps. (a) The two known structures (PDB entries 1U19 and 2RH1) were superimposed using the STAMP algorithm (Russell and Barton, 1992). In the superimposed structures, the retinal and 3-(isopropylamino)propan- 2-ol ligands are very close, which is in agreement with global conservation of binding pockets, as shown on Figure 2.1. (b) The structural alignment of bovine rhodopsin and of human β_2-adrenergic receptor was used to generate a sequence alignment of these two proteins. (c) For both structures, in order to identify residues potentially involved in stabilizing interactions with the ligand (residues of the pocket), we selected residues that presented at least one atom situated at less than 6 from at least one atom of the ligand. Figure 2.1 shows that these two pockets clearly overlap, as expected. (d) Residues of the two pockets (as defined in (c)) were labeled in this structural sequence alignment. These residues were found to form small sequence clusters that were in correspondence in this alignment. These clusters were situated mainly in the apical region of transmembrane segments and included a few extracellular residues. (e) All studied GPCR sequences, including bovine rhodopsin and of human β_2-adrenergic receptor were aligned using CLUSTALW (Chenna et al., 2003) with Blosum matrices (Henikoff and Henikoff, 1992). For each protein, residues in correspondence with a residue of the binding pocket (as defined above) of either bovine rhodopsin or human β_2-adrenergic receptor were retained. This lead to a different number of residues per protein, because of sequence variability. For example, in extracellular regions, some residues from bovine rhodopsin or human β_2-adrenergic receptor had a corresponding residue in some sequences but not in others. In order to provide a homogeneous description of all GPCRs, in the list of residues initially retained for each protein, only residues situated at positions conserved in almost all GPCRs were kept. (f) Each protein was then represented by a vector whose elements corresponded to a potential conserved pocket. This description, although appearing as a linear vector filled with amino acid residues, implicitly codes for a 3D information on the receptor pocket, as illustrated on Figure 2.2. Note that another approach to identify binding pocket residues was previously proposed in Surgand et al. (2006).

These vectors were then used to build a kernel that allows comparison of binding pockets. The classical way to represent motifs of constant length as fixed length vectors is to encode the letter at each position by a 20-dimensional binary vector indicating which amino acid is present, resulting in a 180-dimensional vector representations. In terms of kernel, the inner product between two binding pocket motifs in this representation is simply the number of letters they have in common at the same positions:

$$K_{pb}(x,x') = \sum_{i=1}^{l} \delta(x[i], x'[i]),$$

where l is the length of the binding pocket motifs (31 in our case), $x[i]$ is the i-th residue in x and $\delta(x[i], x'[i])$ is 1 if $x[i] = x'[i]$, 0 otherwise. This is the baseline pocket binding kernel. Alternatively, using a polynomial kernel of degree p over the baseline kernel is equivalent, in terms of feature space, to encoding p-order interactions between amino acids at different positions. In order to assess the relevance of such non-linear extensions we tested this polynomial pocket binding kernel,

$$K_{ppb}(x,x') = \left(K_{pb}(x,x') + 1\right)^p .$$

We only used a degree $p = 2$, although a more careful choice of this parameter could further improve the performances.

- The *binding pocket hierarchy* kernel. Because of the link between binding pockets and ligand recognition, we also defined a new hierarchy based on the sequence alignment of the binding pocket amino acid vectors without gaps. To do this, we used a PAM matrix with high values of gap insertion and extension to compare each couple of GPCR vectors. The obtained scores were used in UPGMA (Unweighted Pair Group Method with Arithmetic mean) to determine a binding pocket similarity based hierarchy. We obtained a tree comparable to phylogenetic trees, and that happens to be share many substructures with the GLIDA hierarchy.

2.5 Experiments

2.5.1 Epitope prediction

Data

In order to evaluate both the performance of our method and the impact of using various kernels for the peptides or the alleles, we test our method on three different benchmark

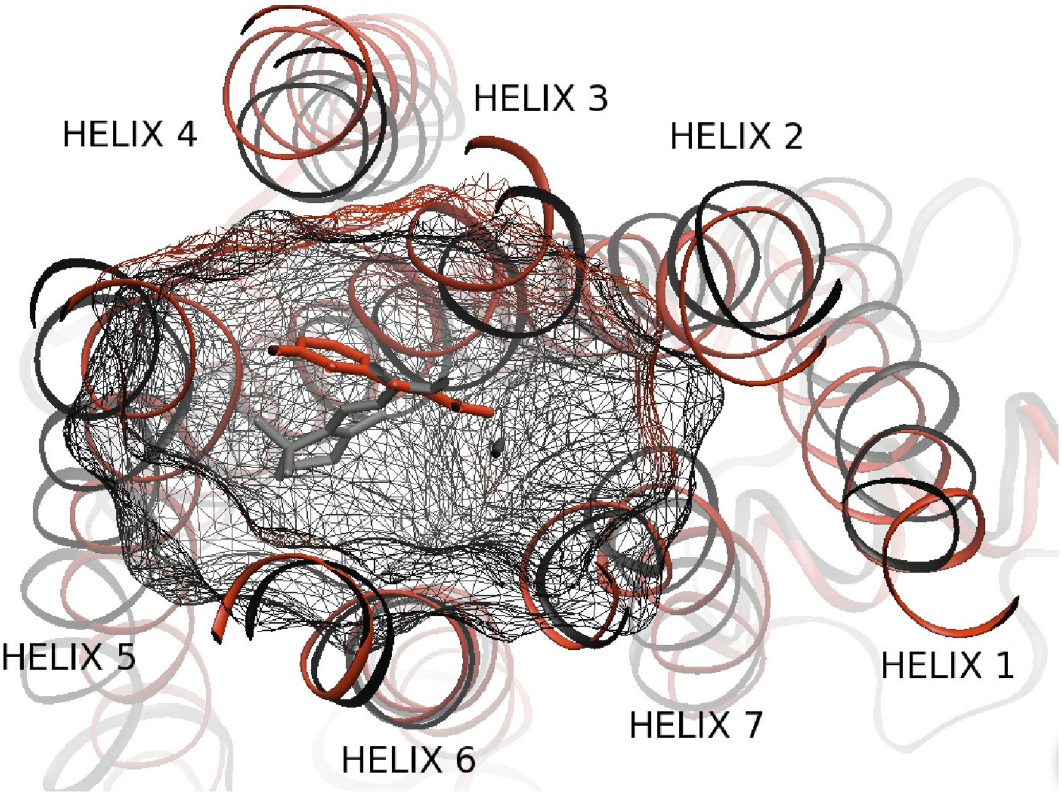

Figure 2.1: Representation of the binding pocket of β_2-adrenergic receptor (in red) and bovine Rhodopsin (in black) viewed from the extracellular surface. On the center of the pocket, 3-(isopropylamino)propan-2-ol and cis-retinal have been represented to show the size and the position of the pocket around each ligand. Figure drawn with VMD (Humphrey et al., 1996).

datasets that have been compiled recently to compare the performance of epitope prediction algorithms.

We first use two datasets compiled by Heckerman et al. (2007), where it is already shown that leveraging improves prediction accuracy with respect to the best published results.The first dataset, called SYFPEITHY+LANL, combines experimentally confirmed positive epitopes from the SYFPEITHY database (see Rammensee et al., 1999, available at `http://www.syfpeithy.de`) and from the Los Alamos HIV database (`http://www.hiv.lanl.gov`) and negative example randomly drawn from the HLA and amino acid distribution in the positive examples, for a total of 3152 data points. For more details, see Heckerman et al. (2007) where this dataset is used to compare the leveraged

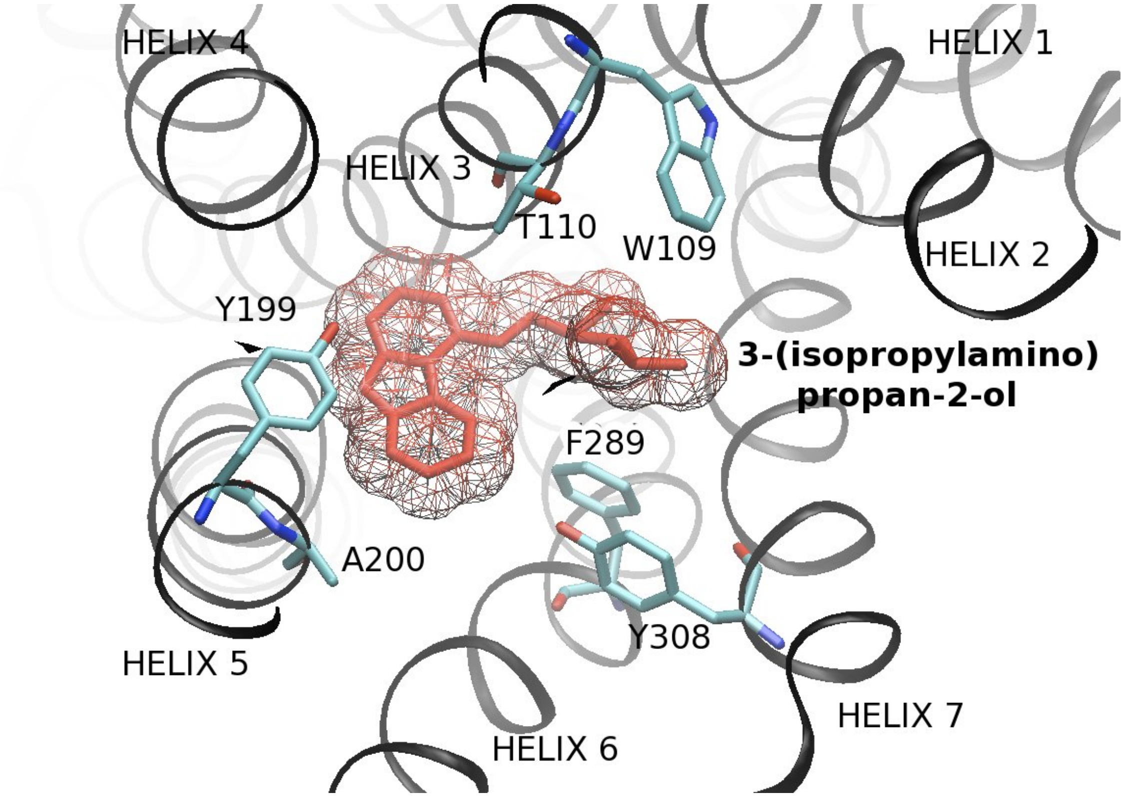

Figure 2.2: 3-(isopropylamino)propan-2-ol and the protein environment of β_2-adrenergic receptor as viewed from the extracellular surface. Amino acid side chains are represented for 6 of the 31 residues (in cyan, blue and red) of the binding pocket motif. Transmembranes helix and 3-(isopropylamino)propan-2-ol are colored in black and red respectively. Figure drawn with VMD (Humphrey et al., 1996).

logistic regression with *DistBoost*. Since this dataset is quite small and was already used as a benchmark, we use it as a first performance evaluation, and to compare our kernels.

The second dataset of Heckerman et al. (2007) contains $160,085$ peptides including those from SYSFPEITHY+LANL and others from the MHCBN data repository (see Bhasin et al., 2003, available at `http://www.imtech.res.in/raghava/mhcbn/index.html`). This corresponds to $1,585$ experimentally validated epitopes, and $158,500$ randomly generated non-binders (100 for each positive). We only kept 50 negative for each positive in the interest of time and assuming this would not deteriorate too much the performance of our algorithm. In the worst case, it is only a handicap for our methods.

Finally, we assess the performance of our method on the MHC-peptide binding benchmark recently proposed by Peters et al. (2006) who gathered quantitative peptide-binding affinity measurements for various species, MHC class I alleles and peptide lengths, which makes it an excellent tool to compare MHC-peptide binding learning methods. Since our method was first designed for binary classification of HLA epitopes, we focused on the 9-mer peptides for the 35 human alleles and thresholded at IC$50 = 500$. Nevertheless, the application of our method to other species or peptide lengths would be straightforward, and generalization to quantitative prediction should not be too problematic either. The benchmark contained 29336 9-mer.

The first dataset is 5-folded, the second 10-folded, so that the test be only performed on HIV (LANL) data. The third dataset is 5-folded. We used the same folds as Heckerman et al. (2007), available at `ftp://ftp.research.microsoft.com/users/heckerma/recomb06` for the first two datasets and the same folds as Peters et al. (2006) available at `http://mhcbindingpredictions.immuneepitope.org/` for the third one.

Molecule-based allele kernels require the amino-acid sequences corresponding to each allele. These sequences are available in various databases, including `http://www.anthonynolan.org.uk/` and Robinson et al. (2000). We used the peptide-sequence alignment for HLA-A, HLA-B and HLA-C loci. Each sequence was restricted to residues at positions involved in the binding site of one of the three loci, see Table 2.1. Preliminary experiments showed that using this restriction instead of the whole sequences didn't change the performance significantly, but it speeds up the calculation of the kernel. We were not able to find the sequence of a few molecules of the two datasets of Heckerman et al. (2007), so in the experiments implying these datasets and a molecule-based allele kernel, we used $K_{bsite7}(a, a') + K_{multitask}(a, a')$ instead of simply using $K_{bsite7}(a, a')$, with a sentinel value of $K_{bsite7}(a, a') = 0$ in these cases. This is the sum of two kernels, so still a positive definite kernel and actually exactly the same thing as $K_{supertype}$ with K_{bsite7} instead of δ_s.

Locus	Positions
HLA-A	5, 7, 9, 24, 25, 34, 45, 59, 63, 66, 67, 70, 74, 77, 80, 81, 84, 97, 99, 113, 114, 116, 123, 133, 143, 146, 147, 152, 155, 156, 159, 160, 163, 167, 171
HLA-B	5, 7, 8, 9, 24, 45, 59, 62, 63, 65, 66, 67, 70, 73, 74, 76, 77, 80, 81, 84, 95, 97, 99, 114, 116, 123, 143, 146, 147, 152, 155, 156, 159, 160, 163, 167, 171
HLA-C	5, 7, 9, 22, 59, 62, 64, 66, 67, 69, 70, 73, 74, 77, 80, 81, 84, 95, 97, 99, 116, 123, 124, 143, 146, 147, 156, 159, 163, 164, 167, 171

Table 2.1: Residue positions involved in the binding site for the three loci, according to Doytchinova et al. (2004)

Results

We first use K_{linseq} and K_{seq5} for the peptides and $K_{uniform}$ (one SVM for all the alleles), K_{Dirac} (one SVM for each allele), $K_{multitask}$, $K_{supertype}$ and K_{bsite7} for the alleles on the small SYFPEITHI+LANL dataset. Using combinations of molecule-based and non-molecule-based kernels for K_{all} didn't improve the prediction, generally the result was as good as or slightly worse than the result obtained with the best of the two combined kernels. Results are displayed on Table 2.2, and ROC curves for $K_{linseq} \times K_{Dirac}$, $K_{linseq} \times K_{supertype}$, $K_{seq5} \times K_{supertype}$ and $K_{seq5} \times K_{bsite7}$ on Figure 2.3.

Table 2.2 demonstrates the benefits of carefully sharing information across alleles. The *Dirac* allele kernel being the baseline kernel corresponding to independent training of SVM on different alleles, we observe an improvement of at least 2% when information is shared across alleles during training (with the *multitask,supertype* or *bsite7* strategies). It should be noted, however, that the *uniform* strategies which amount to training a single model for all alleles perform considerably worse than the *Dirac* strategies, justifying the fact that it is still better to build individual models than a single model for all alleles. Among the strategies to share information across alleles, the *supertype* allele kernel seems to work slightly better than the two other ones. However, one should keep in mind that there is a possible bias in the performance of the *supertype* kernel, because some peptides in the test sets might have contributed to the definition of the allele supertypes. Among the *multitask* kernel, which considers all different alleles as equally similar, and the *bsite7* kernel, which shares more information between alleles that have similar residues at key positions, we observe a slight benefit for the *bsite7* kernel, which justifies the idea that including biological knowledge in our framework is simple and powerful. Finally, we observe that for all allele kernels, the nonlinear *seq5* peptide kernel outperforms the baseline *linseq* kernel, confirming that linear models based on position-specific score matrices might be a too restrictive set of models to predict accurately epitopes.

In terms of absolute value, all three allele kernels that share information across alleles combined with the nonlinear *seq5* peptide kernel (AUC $= 0.943 \pm 0.015$) strongly outperform the leveraged logistic regression of Heckerman et al. (2007) (AUC $= 0.906 \pm 0.016$) and the boosted distance metric learning algorithm of Hertz and Yanover (2006) (AUC $= 0.819 \pm 0.055$). This corresponds to a decrease of roughly 40% of the area above the ROC curve compared to the best method. As the boosted distance metric learning approach was shown to be superior to a variety of state-of-the-art other methods by Hertz and Yanover (2006), this suggest that our approach can compete if not overcome the best methods in terms of accuracy.

As we can clearly see in Table 2.2, two factors are involved in the improvement over the leveraged logistic regression of Heckerman et al. (2007):

- The use of an SVM instead of a logistic regression, since this is the only difference between the leveraged logistic regression and our SVM with a $K_{linseq} \times K_{supertype}$ kernel. This, however, may not be intrinsic to the algorithms, but caused by optimization issues for the logistic regression in high dimension.

- The use of a non-linear kernel for the peptide, as we observe a clear improvement in the case of SVM (this improvement might therefore also appear if the logistic regression was replaced by a kernel logistic regression model with the adequate kernel).

Figure 2.3 illustrates the various improvement underlined by this experiment: first from the individual SVM ($K_{linseq} \times K_{Dirac}$), to the $K_{linseq} \times K_{supertype}$ SVM which is the SVM equivalent of leveraged logistic regression, and finally to $K_{seq5} \times K_{supertype}$ and $K_{seq5} \times K_{bsite7}$ SVM that both give better performances than $K_{linseq} \times K_{supertype}$ SVM because they use a nonlinear kernel to compare the peptides. It is also worth noting that the *supertype*

$K_{all} \backslash K_{pep}$	linseq	seq5
uniform	0.826 ± 0.010	0.883 ± 0.011
Dirac	0.891 ± 0.014	0.893 ± 0.024
multitask	0.910 ± 0.008	0.936 ± 0.008
supertype	0.923 ± 0.011	0.943 ± 0.015
bsite7	0.919 ± 0.011	0.943 ± 0.009

Table 2.2: AUC results for an SVM trained on the SYFPEITHI+LANL with various kernel and estimated error on the 5 folds.

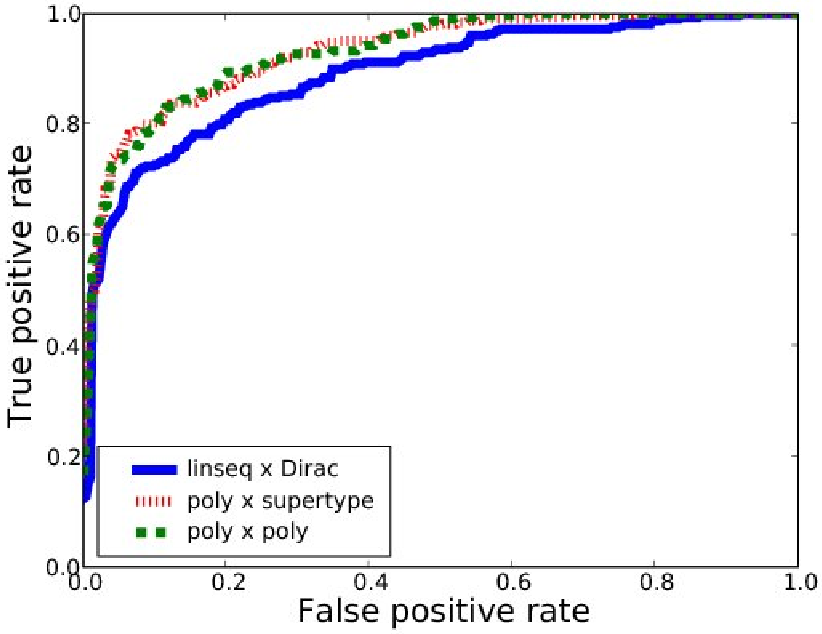

Figure 2.3: ROC curves on the pooled five folds of the SYFPEITHI+LANL benchmark.

and the *bsite7* strategies give very similar results, which makes them two good strategies to leverage efficiently across the alleles with different information.

These results are confirmed by the MHCBN+SYFPEITHI+LANL benchmark, for which the results are displayed in Table 2.3. Again, the use of SVM with our product kernels clearly improves the performance with respect to Heckerman et al. (2007) (from 0.906 to 0.938). Moreover, we again observe that learning a leveraged predictor using the data from all the alleles improves the global performance very strongly, hence the important

Method	AUC
Leveraged LR	0.906
$K_{linseq} \times K_{stype}$	0.916 ± 0.008
$K_{seq5} \times K_{dirac}$	0.867 ± 0.010
$K_{seq5} \times K_{multitask}$	0.934 ± 0.006
$K_{seq5} \times K_{stype}$	0.939 ± 0.006
$K_{seq5} \times K_{bsite7}$	0.938 ± 0.006

Table 2.3: AUC results for an SVM trained on the MHCBN+SYFPEITHI+LANL benchmark with various kernel and estimated error on the 10 folds.

step between Dirac (0.867) and all the multitask-based methods, including the simplest multitask kernel (0.934). It is worth reminding here that the multitask kernel is nothing but the sum of the Dirac and uniform kernels, *i.e.*, that it contains no additional biological information: the improvement is caused by the mere fact of using roughly (with a weighting of 0.5) the points of other alleles to learn the predictor of one allele. Figure 2.4 shows the ROC curves for SVM with $K_{seq5} \times K_{Dirac}$, $K_{seq5} \times K_{supertype}$ and $K_{seq5} \times K_{bsite7}$ kernels on this benchmark. Again, we clearly see the strong improvement between leveraged and non-leveraged strategies. The difference between the $K_{seq5} \times K_{Dirac}$ and the two others is only caused by leveraging, since in the three case the same nonlinear strategy was used for the peptide part. On the other hand, the figure illustrates once again that our two high-level (*i.e.*, more sophisticated than *multitask*) strategies for leveraging across alleles give almost the same result.

Finally, Table 2.4 presents the performance on the IEDB benchmark proposed in Peters et al. (2006). The indicated performance corresponds, for each method, to the average on the AUC for each of the 35 alleles. This gives an indication of the global performances of each methods. The ANN field is the tool proposed in Peters et al. (2006) giving the best results on the 9-mer dataset, an artificial neural network proposed in Nielsen et al. (2003), while the ADT field refers to the adaptive double threading approach recently proposed in Jojic et al. (2006) and tested on the same benchmark. These tools were compared to and significantly outperformed other tools in the comprehensive study of Peters et al. (2006), specifically Peters and Sette (2005) and Bui et al. (2005), that are both scoring-matrix-based. Our approach gives equivalent results in terms of global performances as Nielsen

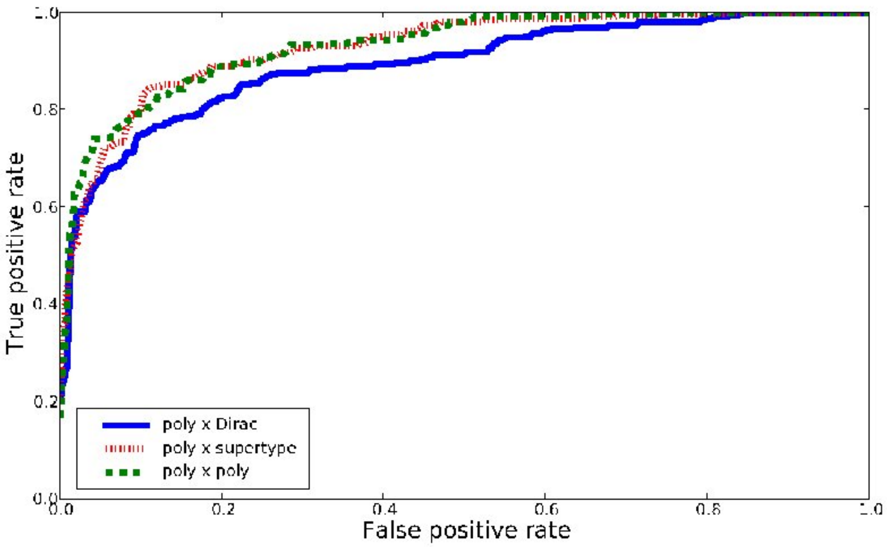

Figure 2.4: ROC curves on the pooled ten folds of the MHCBN+SYFPEITHI+LANL benchmark.

et al. (2003), and therefore outperforms the other internal methods.

Table 2.5 presents the performances on the 10 alleles with less than 200 training points, together with the performances of the best internal tool, Nielsen et al. (2003) ANN, and the adaptive double threading model that gave good prediction performances on the alleles with few training data. Except for one case, our SVM outperforms both models. This means of course that our approach does not perform as well as Nielsen et al. (2003) on the alleles with a large training set, but nothing prevents an immunologist from using one tool for some alleles and another tool for other alleles. As we said in introduction, our original concern was to improve binding prediction for alleles with few training points, and for which it is hard to generalize. This was the main point of using a multitask learning approach. The results on this last benchmark suggest that the leveraging approaches succeed in improving prediction performances when few training points are available.

Discussion and concluding remarks

In these experiments, we used the general framework of pairwise learning introduced in this chapter to share efficiently the binding information available for various alleles by simply

Method	AUC
SVM with $K_{seq5} \times K_{Dirac}$	0.804
SVM with $K_{seq5} \times K_{supertype}$	0.877
SVM with $K_{seq5} \times K_{bsite7}$	0.892
ADT	0.874
ANN	0.897

Table 2.4: AUC results for an SVM trained on the IEDB benchmark with various methods.

Allele	Peptide number	$K_{seq5} \times K_{bsite7}$	ADT	ANN
A_2301	104	0.887 ± 0.021	0.804	0.852
A_2402	197	0.826 ± 0.025	0.785	0.825
A_2902	160	0.948 ± 0.015	0.887	0.935
A_3002	92	0.826 ± 0.048	0.763	0.744
B_1801	118	0.866 ± 0.020	0.869	0.838
B_4002	118	0.796 ± 0.025	0.819	0.754
B_4402	119	0.782 ± 0.084	0.678	0.778
B_4403	119	0.796 ± 0.042	0.624	0.763
B_4501	114	0.889 ± 0.029	0.801	0.862
B_5701	59	0.938 ± 0.046	0.832	0.926

Table 2.5: Detail of the IEDB benchmark for the 10 alleles with less than 200 training points (9-mer data).

defining a kernel for the peptides, and another one for the alleles. The result is a simple model for MHC-peptide binding prediction that uses information from the whole dataset to make specific prediction for any of the alleles. Our approach is simple, general and both easy to adapt to a specific problem by using more adequate kernels, and to implement, by running any SVM implementation with these kernels. Everything is performed in low dimension and with no need for feature selection.

We presented performances on three benchmarks. On the first two benchmark, our approach performed considerably better than the state-of-the-art, which illustrates the good general behavior in terms of prediction accuracy. Besides, these experiments clearly confirmed the interest of leveraging the information across the alleles. On the last benchmark, the results were globally comparable to the best state-of-the-art tested in Peters et al. (2006), with a strong improvement on the alleles for which few training points were available, probably, as it was already observed, because of the fact that our model uses all the points from all the alleles for each allele-specific prediction.

Another contribution is the use of allele sequences, which allows us to improve the prediction accuracy and to do as well as what was done with the supertype information. Supertype is a crucial information and a key concept in the development of epitope-based vaccines, for example to find epitopes that bind several alleles instead of just one. However, one should be careful when using it to learn an automatic epitope predictor because even if the idea behind a supertype definition is to represent a general ligand trend, the intuition is always guided by the fact that some alleles have overlapping repertoires of known binders, and it is not easy to figure out to which extent the known epitopes used to assess the predictor performances were used to design the supertypes.

Because of these overfitting issues and the fact that supertypes are difficult to define, the good performances of molecule-based allele kernel with respect to the supertype-based allele kernels are good news. This potentially allows us to leverage efficiently across alleles even when the supertype is unknown, which is often the case, and we don't take the risk to use overfitted information when learning on large epitope databases.

Although the kernels we used already gave good performances, there is still room for improvement. A first way to improve the performances would be to use more adequate kernels to compare the peptides and, probably more important, to compare the alleles. In other words answering the question, what does it mean in the context of MHC-peptide binding prediction for two alleles to be similar? Possible answers should probably involve better kernels for the allele sequences, and structural information which could be crucial to predict binding and, as we said in introduction, is already used in some models. Another interesting possibility is, as it was suggested in Hertz and Yanover (2007), the use of true non-binders, that could make the predictor more accurate than randomly generated peptides since these experimentally assessed peptides are in general close to the known binders. Finally, it could be useful to incorporate the quantitative IC50 information when available, instead of simply thresholding as we did for the last benchmark.

This leads us to the possible generalizations we hope to work on, besides these improvements. Using the binding affinity information, it is obviously possible to apply our general framework to predict quantitative values, using regression models with the same type of kernels. This framework could also be used for a lot of similar problems involving binding, like MHC-type-II-peptide binding where sequences can have variable length and the alignment of epitopes usually performed as pre-processing can be ambiguous. Salomon and Flower (2006) already proposed a kernel for this case. Another interesting application would be prediction of a virus susceptibility to a panel of drugs for various mutations of the virus.

2.5.2 Compound prediction: KEGG benchmark

Data

We extracted compound interaction data from the KEGG BRITE Database (Kanehisa et al., 2002, 2004) concerning enzyme, GPCR and ion channel, three target classes particularly relevant for novel drug development.

For each family, the database provides a list of known compounds for each target. Depending on the target families, various categories of compounds are defined to indicate the type of interaction between each target and each compound. These are for example *inhibitor, cofactor* and *effector* for enzyme ligands, *antagonist* or *(full/partial) agonist* for GPCR and *pore blocker, (positive/negative) allosteric modulator, agonist* or *antagonist* for ion channels. The list is not exhaustive for the latter since numerous categories exist. Although different types of interactions on a given target might correspond to different binding sites, it is theoretically possible for a non-linear classifier like SVM with non-linear kernels to learn classes consisting of several disconnected sets. Therefore, for the sake of clarity of our analysis, we do not differentiate between the categories of compounds.

For each target class, we retained only one protein by element of the hierarchy. In particular, we did not take into account the different orthologs of the targets, and the different enzymes corresponding to the same EC number. We then eliminated all compounds for which no molecular descriptor was available (principally peptide compounds), and all the targets for which no compound was known. For each target, we generated as many negative ligand-target pairs as we had known ligands forming positive pairs by combining the target with a ligand randomly chosen among the other targets' ligands (excluding those that were known to interact with the given target). This protocol generates false negative data since some ligands could actually interact with the target although they have not been experimentally tested, and our method could benefit from experimentally confirmed negative pairs.

This resulted in 2436 data points for enzymes (1218 known enzyme-ligand pairs and 1218 generated negative points) representing interactions between 675 enzymes and 524 compounds, 798 training data points for GPCRs representing interactions between 100 receptors and 219 compounds and 2330 ion channel data points representing interactions between 114 channels and 462 compounds. Besides, Figure 2.5 shows the distribution of the number of known ligands per target for each dataset and illustrates the fact that for most of them, few compounds are known.

For each target t in each family, we carried out two experiments. First, all data points corresponding to other targets in the family were used for training only and the n_t points corresponding to t were k-folded with $k = \min(n_t, 10)$. That is, for each fold, an SVM classifier was trained on all points involving other targets of the family plus a fraction of the points involving t, then the performances of the classifier were tested on the remaining fraction of data points for t. This protocol is intended to assess the incidence of using ligands from other targets on the accuracy of the learned classifier for a given target. Second, for each target t we trained an SVM classifier using only interactions that did not involve t and tested on the points that involved t. This is intended to simulate the behavior of our framework when making predictions for orphan targets, *i.e.*, for targets for which no ligand is known.

For both experiments, we used the area under the ROC curve (AUC) as a performance measure. The ROC curve was computed for each target using the test points pooled from all the folds. For the first protocol, since training an SVM with only one training point does not really make sense and can lead to "anti-learning" less than 0.5 performances, we set all results r involving the Dirac target kernel on targets with only 1 known ligand to $\max(r, 0.5)$. This is to avoid any artefactual penalization of the Dirac approach and make sure we measure the actual improvement brought by sharing information across targets.

Results

We first discuss the results obtained on the three datasets for the first experiment, assessing how using training points from other targets of the family improves prediction accuracy with respect to individual (Dirac-based) learning. Table 2.6 shows the mean AUC across the family targets for an SVM with a product kernel using the Tanimoto kernel for ligands and various kernels for proteins. For the enzymes and ion channels datasets, we observe significant improvements when the multitask kernel is used in place of the Dirac kernel, on the one hand, and when the hierarchy kernel replaces the multitask kernel, on the other hand. For example, the Dirac kernel only performs at an average AUC of 77% for the ion channel dataset, while the multitask kernel increases the AUC to 87.3% and the hierarchy kernel brings it to 92.5%. For the enzymes, a global improvement of 30.9% is observed between the Dirac and the hierarchy approaches. This clearly demonstrates the benefits of sharing information among known ligands of different targets, on the one hand, and the relevance of incorporating prior information into the kernels, on the other hand.

On the GPCR dataset though, the multitask kernel performs slightly worse than the Dirac kernel, probably because some targets in different subclasses show very different binding behavior which results in adding more noise than information when sharing naively with this kernel. However a more careful handling of the similarities between GPCRs through the hierarchy kernel again results in significant improvement over the Dirac kernel (from 75% to 92.6%), again demonstrating the relevance of the approach.

Sequence-based target kernels do not achieve the same performance as the hierarchy kernel, although they perform relatively well for the ion channel dataset, and give better results than the multitask kernel for both GPCR and ion channel datasets. In the case of enzymes, it can be explained by the diversity of the proteins in the family and for the GPCR, by the well known fact that the receptors do not share overall sequence homology (Gether, 2000). Figure 2.6 shows 3 of the tested target kernels for the ion channel dataset. The hierarchy kernel adds some structure information with respect to the multitask kernel, which explains the increase in AUC. The local alignment sequence-based kernels fail to precisely re-build this structure but retain some substructures. In the cases of GPCR and enzymes,

$K_{tar}\backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.646 ± 0.009	0.750 ± 0.023	0.770 ± 0.020
multitask	0.931 ± 0.006	0.749 ± 0.022	0.873 ± 0.015
hierarchy	0.955 ± 0.005	0.926 ± 0.015	0.925 ± 0.012
mismatch	0.725 ± 0.009	0.805 ± 0.023	0.875 ± 0.015
local alignment	0.676 ± 0.009	0.824 ± 0.021	0.901 ± 0.013

Table 2.6: AUC for the first protocol on each dataset with various target kernels.

almost no structure is found by the sequence kernels, which, as alluded to above, was expected and suggests that more subtle comparison of the sequences would be required to exploit the information they contain.

Figure 2.7 illustrates the influence of the number of training points for a target on the improvement brought by using information from similar targets. As one could expect, the improvement is very strong when few ligands are known and decreases when enough training points become available. After a certain point (around 30 training points), using similar targets can even impair the performances. This suggests that the method could be globally improved by learning for each target independently how much information should be shared, for example through kernel learning approaches (Lanckriet et al., 2004a).

The second experiment aims at pushing this remark to its limit by assessing how each strategy is able to predict ligands for proteins with no known ligand. Table 2.7 shows the results in that case. As expected, the classifiers using Dirac kernels show random behavior in this case since using a Dirac kernel with no data for the target amounts to learning with no training data at all. In particular, in the SVM implementation that we used, the classifier learned with no data from the task gave constant scores to all the test points, hence the 0.500 ± 0.000 AUC on the test data. On the other hand we note that it is still possible to obtain reasonable results using adequate target kernels. In particular, the hierarchy kernel loses only 7.2% of AUC for the ion channel dataset, 5.1% for the GPCR dataset and 1.7% for the enzymes compared to the first experiment where known ligands were used, suggesting that if a target with no known compound is placed in the hierarchy through, *e.g.* in the case of GPCR homology detection with known members of the family using specific GPCR alignment algorithms (Kratochwil et al., 2005) or fingerprint analysis (Attwood et al., 2003), it is possible to predict some of its ligands almost as accurately as if some of them were already available.

In this second setting, our approach when using the hierarchy kernel on the targets is closely related to annotation transfer. Indeed, the learned predictor in this case will predict a molecule to be a ligand of a given target if the molecule is similar to the known ligands of close targets in the hierarchy. In particular, it will predict that the ligands of the target's direct neighbors are ligands of the target (which is an intuitive and natural way to choose new candidate binders). A major difference however is that an annotation transfer approach will not predict as a ligand a molecule that is very similar to close target's ligands but that is not itself a close target's ligand. In particular if the candidate molecule is not present anywhere else in the ligand database, it will never be predicted to be a ligand. Exemples can be found in each of the considered target classes. The 4-Aminopyridine is a blocker of the ion channel KCJN5, a potassium inwardly-rectifying channel. Although this molecule is a known blocker of other channels (in particular, many potassium channels), it is not a known ligand of any other channel of KCJN5's superfamily. However, the most similar molecule in the database, in the sense of the Tanimoto kernel, is the Pinacidil, which happens to be a known ligand of two direct neighbors of KCJN5. This allows our method to predict 4-Aminopyridine as a ligand for this target. Similarly, N-Acetyl-D-glucosamine 1,6-bisphosphate is the only known effector of phosphoacetylglucosamine mutase, an enzyme of the isomerase family. This molecule is not a known ligand of any other enzyme in the database, so a direct annotation transfer approach would never predict it as a ligand. Our method, on the other hand, predicts it correctly, taking advantage of the fact that very similar molecules like D-Ribose 1,5-bisphosphate or alpha-D-Glucose 1,6-bisphosphate are known ligands of direct neighbors. The same observation can be made for several GPCRs, including the prostaglandin F receptor whose 3 known ligands are not ligands of any other GPCR but whose direct neighbors have similar ligands.

Discussion

We propose a general method to combine the chemical and the biological space in an algorithmic way and predict interaction between any small molecule and any target, which makes it a vary valuable tool for drug discovery. The method allows one to represent systematically a ligand-target pair, including information on the interaction between the ligand

$K_{tar} \backslash$ Target	Enzymes	GPCR	Channels
Dirac	0.500 ± 0.000	0.500 ± 0.000	0.500 ± 0.000
multitask	0.902 ± 0.008	0.576 ± 0.026	0.704 ± 0.026
hierarchy	0.938 ± 0.006	0.875 ± 0.020	0.853 ± 0.019
mismatch	0.602 ± 0.008	0.703 ± 0.027	0.729 ± 0.024
local alignment	0.535 ± 0.005	0.751 ± 0.025	0.772 ± 0.023

Table 2.7: AUC for the second protocol on each dataset with various target kernels.

and the target. Prediction is then performed by any machine learning algorithm (an SVM in our case) in the joint space, which makes targets with few known ligands benefit from the data points of similar targets, and which allows one to make predictions for targets with no known ligand. Our information sharing process therefore simply relies on a choice of description for the ligands, another one for the targets and on classical machine learning methods: everything is done by casting the problem in a joint space and no explicit procedure to select which part of the information is shared is needed. Since it subdivides the representation problem into two subproblems, our approach makes use of previous work on kernels for molecular graphs and kernels for biological targets. For the same reason, it will automatically benefit from future improvements in both fields. This leaves plenty of room to increase the performance.

Results on experimental ligand datasets show that using target kernels allowing to share information across the targets considerably improve the prediction, especially in the case of targets with few known ligands. The improvement is particularly strong when the target kernel uses prior information on the structure between the targets, *e.g.*, a hierarchy defined on a target class. Although the usage of a kernel based on the hierarchy is restricted to protein families where hierarchical classification schemes exist, it applies to the three main classes of proteins targeted by drugs, and others like cytochromes and abc transporters. Sequence kernels, on the other hand, did not give very good results in our experiments. However, we believe using the target sequence information could be an interesting alternative or complement to the hierarchy kernel. For example, Jacob et al. (2008) used a kernel based on the sequence of the GPCR that performed as well as the kernel based on the GPCR hierarchy. Further improvement could come from the use of kernel for structures in the cases where 3D structure information is available (*e.g.* for the enzymes, but not for the

GPCR). Our method also shows good performances even when no ligand at all is known for a given target, which is excellent news since classical ligand based approaches fail to predict ligand for these targets in the one hand, and docking approaches are computationally expensive and not feasible when the target 3D structure is unknown which is the case of GPCR in the other hand.

In future work, it could be interesting to apply this framework to quantitative prediction of binding affinity using regression methods in the joint space. It would also be important to confirm predicted ligands experimentally or at least by docking approaches when the target 3D structure is available.

2.5.3 Compound prediction: GLIDA benchmark

Data

For a more extensive study on GPCR, we used the GLIDA GPCR-ligand database Okuno et al. (2006) which includes 22964 known ligands for 3738 GPCRs from human, rat and mouse. The ligand database contains highly diverse molecules, from ions and very small molecules up to peptides, and a significant number of duplicates. These redundancies were eliminated. Elimination of duplicates present in the GLIDA database was important here because it could have led to over-optimistic evaluation in the cross-validation procedure described below. The remaining molecules were further filtered in order to satisfy two constraints. First, our method relies on the evaluation of similarities between molecules using kernels, which makes sense only if the molecules are comparable in size. Second, since the long term goal is to identify drug candidates targeting GPCRs, it was important to retain drug-like compounds, i.e. molecules having the adequate physico-chemical characteristics to be potential drugs candidates satisfying ADME criteria Caldwell et al. (1995). Therefore, to only keep drug-like compounds, we filtered the GLIDA database using the filter program (OpenEye Scientific Software) with standard parameters, which removes molecules according to calculated properties such as molecular weight, hydrogen bond donor and acceptor count, number of rotatable bonds, ring size and number etc... as discussed in Lipinski et al. (2001); Egan et al. (2000); Veber et al. (2002); Martin (2005). For example, only molecules of molecular weights ranging from 150 Da to 450 Da were kept (the classically accepted range for drugs), since the aim was to evaluate if statistical learning was possible on drug-like compounds. Another example was the elimination of molecules with more than 10 rotatable bonds (although most of them being already filtered out on the molecular weight criterion). Indeed, they correspond to very flexible molecules that are not suitable for the use of 3D descriptors. Overall these filters retained 2446 molecules, available under a 2D description file in the GLIDA data bank, and giving 4051 interactions with the human GPCRs. The number of molecules retained is only a small fraction of the GLIDA database, but it corresponds to all drug-like compounds of this database. For each positive interaction given by this restricted set, we generated a negative interaction involving the same receptor and one of the ligands that was in the database and that was not indicated as one of its ligands. This may have generated a few false negative points in our benchmark, and it would be interesting to use experimentally tested negative interactions. However, the mean similarity between the different ligands in the database using the Tanimoto kernel, a classical normalized similarity measure for ligands which is later used in our method, is quite low (0.13). Besides, only 6.7% of the ligands have a mean similarity of more than 0.2 to the other ligands. This suggests that even if false negative have to be expected, this method to generate negative interaction is a reasonable approximation. We loaded the sequences of all GPCRs that are able to bind any of these ligands, which resulted in 80 sequences, all corresponding to human GPCRs. The retained GPCRs were significantly diverse in sequence, most of them sharing 15% to 50% pairwise sequence similarities. Furthermore, they belong to various families, according to the GLIDA classification. They are found in several subfamilies of class A (rhodopsin-like receptors), classes B (secretin family) and C (metabotropic family). In the GLIDA database, GPCRs are classified in hierarchy (as mentioned above) which was also loaded for use in the hierarchy kernel.

Results

We ran two different sets of experiments on this dataset in order to illustrate two important points. In a first set of experiments, for each GPCR, we 5-folded the data available, *i.e.*, the line of the interaction matrix corresponding to this GPCR. The classifier was trained with four folds and the whole data from the other GPCRs, *i.e.*, all other lines of the interaction matrix. The prediction accuracy for the GPCR under study was

$K_{tar} \backslash K_{lig}$	2D Tanimoto	3D pharmacophore
Dirac	86.2 ± 1.9	84.4 ± 2.0
multitask	88.8 ± 1.9	85.0 ± 2.3
hierarchy	93.1 ± 1.3	88.5 ± 2.0
binding pocket	90.3 ± 1.9	87.1 ± 2.3
poly binding pocket	92.1 ± 1.5	87.4 ± 2.2
binding pocket hierarchy	93.0 ± 1.4	90.0 ± 2.1

Table 2.8: Prediction accuracy for the first experiment with various ligand and target kernels.

then tested on the remaining fold. The goal of these first experiments was to evaluate if using data from other GPCRs improved the prediction accuracy for a given GPCR. In a second set of experiments, for each GPCR we ignored ligand data available for this particular GPCR, we trained a classifier on the whole data from the other GPCRs, and tested on the data of the considered GPCR. The goal was to assess how efficient our chemogenomics approach would be to predict the ligands of orphan GPCRs. In both experiments, the C parameter of the SVM was selected by internal cross validation on the training set among $2^i, i \in \{-8, -7, \ldots, 5, 6\}$. The data and source code (under GPL license) are publicly available at `http://www.biomedcentral.com/content/supplementary/1471-2105-9-363-s2.tgz`.

For the first experiment, since learning an SVM with only one training point does not really make sense and can lead to "anti-learning" less than 0.5 performances, we set all results r involving the Dirac GPCR kernel on GPCRs with only 1 known ligand to $\max(r, 0.5)$. This is to avoid any artefactual penalization of the Dirac approach and make sure that we measure the actual improvement brought by sharing information across GPCRs.

Table 2.8 shows the results of the first experiments with all the ligand and GPCR kernel combinations. For all the ligand kernels, one observes an improvement between the individual approach (Dirac GPCR kernel, 86.2%) and the baseline multitask approach (multitask GPCR kernel, 88.8%). The latter kernel is merely modeling the fact that each GPCR is uniformly similar to all other GPCRs, and twice more similar to itself. It does not use any prior information on the GPCRs, and yet, using it improves the global performance with respect to individual learning. Using more informative GPCR kernels further improves the prediction accuracy. In particular, the hierarchy kernel add more than 4.5% of precision with respect to naive multitask approach. All the other informative GPCR kernels also improve the performance. The polynomial binding pocket kernel is almost as efficient as the hierarchy kernel, which is an interesting result. Indeed, one could fear that using the hierarchy kernel, for the construction of which some knowledge of the ligands may have been used, could have introduced bias in the results. Such bias is certainly absent in the binding pocket kernel. The fact that the same performance can be reached with kernels based on the mere sequence of GPCRs' pockets is therefore an important result. Figure 2.8 shows three of the GPCR kernels. The baseline multitask is shown as a comparison. Interestingly, many of the subgroups defined in the hierarchy can be found in the binding pocket kernel, that is, they are retrieved from the simple information of the binding pocket sequence.

The 3D kernel for the ligands, on the other hand, did not perform as well as the 2D kernel. This can be either explained by the fact the the pharmacophore kernel is not suited to this problem, or by the fact that choosing the conformer of the ligand is not a trivial task. This point is discussed below.

Figure 2.9 illustrates how the improvement brought by the chemogenomics approach varies with the number of available training points. As one could have expected, the strongest improvement is observed for the GPCRs with few (less than 20) training points (*i.e.*, less than 10 known ligands since for each known ligand an artificial non-ligand was generated). When more training points become available, the improvement is less important, and sharing the information across the GPCRs can even degrade the performances. This is an important point, first because, as showed on Figure 2.10, many GPCRs have few known ligands (in particular, 11 of them have only two training points), and second because it shows that when enough training points are available, individual learning will probably perform as well as or better than our chemogenomics approach.

Our second experiment intends to assess how our chemogenomics approach can perform when predicting ligands for orphan GPCRs, *i.e.*, with no training data available for the GPCR of interest. Table 2.9 shows that in this setting, individual learning performs random prediction. Naive multitask approach provides modest improvement of the performance, but informative kernels such as hierarchical and binding pocket kernels achieve 77.4% and 78.1% of precision respectively, that is, almost 30% better than the random approach one would get when no data is available. Here again, the fact that the binding pocket kernel

$K_{tar} \backslash K_{lig}$	2D Tanimoto	3D pharmacophore
Dirac	50.0 ± 0.0	50.0 ± 0.0
multitask	56.8 ± 2.5	58.2 ± 2.2
hierarchy	77.4 ± 2.4	76.2 ± 2.2
binding pocket	78.1 ± 2.3	76.6 ± 2.2
poly binding pocket	76.4 ± 2.4	74.9 ± 2.3
binding pocket hierarchy	75.5 ± 2.4	76.5 ± 2.2

Table 2.9: Prediction accuracy for the second experiment with various ligand and target kernels.

Family $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin peptide receptors (18)	73.7	80.0	85.8	83.8	83.7
Rhodopsin amine receptors (35)	91.1	92.1	94.0	93.9	94.1
Rhodospin other receptors (17)	83.6	88.0	95.7	95.9	95.9
Metabotropic glutamate family (9)	73.1	93.5	98.9	83.3	93.3
Secretin family (1)	50.0	100.0	100.0	50.0	100.0

Table 2.10: Mean prediction accuracy for each GPCR family for the first experiment with the 2D Tanimoto ligand kernel and various target kernels. The numbers in bracket are the numbers of receptors considered in the experiment for each family. BP is the binding pocket kernel and PBP the poly binding pocket kernel.

Family $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin peptide receptors (18)	50.0	50.6	66.7	74.0	65.3
Rhodopsin amine receptors (35)	50.0	56.0	73.7	74.0	73.1
Rhodospin other receptors (17)	50.0	50.2	86.5	87.6	85.5
Metabotropic glutamate family (9)	50.0	79.7	93.9	87.2	91.3
Secretin family (1)	50.0	100.0	100.0	50.0	100.0

Table 2.11: Mean prediction accuracy for each GPCR family for the second experiment with the 2D Tanimoto ligand kernel and various target kernels. The numbers in bracket are the numbers of receptors considered in the experiment for each family. BP is the binding pocket kernel and PBP the poly binding pocket kernel.

that only uses the sequence of the receptor pocket performs as well as the hierarchy-based kernel is encouraging. It suggests that given a receptor for which nothing is known except its sequence, it is possible to make reasonable ligand predictions.

Discussion

Our results demonstrate that chemogenomic approaches outperform individual approach, in particular in cases where very limited or no ligand information is available, as shown in Table 2.9 and Figure 2.9.

In the case of well studied GPCRs, more classical ligand-based methods (QSAR) may be better suited to predict new strong binders from a large number of known ligands, as shown in Figure 2.9. Consistent with this observation, Tables 2.10 and 2.11 show that in the two types of experiments, the improvement is observed for all subfamilies of GPCRs retained in this study. This is an interesting result since most of published virtual screening studies on GPCRs were applied to class A GPCRs.

Since our chemogenomic approach is a ligand-based approach, it would probably be interesting to use it in combination with docking. Indeed, although prior known ligands can help tuning docking procedures to the receptor under study, it can in principle be used with little or no ligand information. When more experimental 3D structures become available for GPCRs in the future, this will help building reliable models for a wider range of GPCRs that would be suitable for docking studies. Joint use of ligand-based chemogenomic and docking would certainly improve predictions.

We chose to use a binary descriptor for the receptor-ligand interaction, while QSAR or docking methods usually try to rank molecules according to their predicted affinity for the receptor. However, affinity prediction is still a subject of research at the level of a single receptor, at least when using methods whose calculation times are compatible with the screening of large molecular databanks. In this context, we feel that in chemogenomic approaches, where information is shared between different proteins, such quantitative prediction is even more challenging. This led us to retain the binary binding and non-binding descriptors, although it would formally have been straightforward to use a regression algorithm instead of a classification one to make quantitative predictions.

It is not always easy to compare the performances of a new method to other existing methods, and particularly in the case of GPCRs. Indeed, at least to our knowledge, there is up to now no public complete data from previous screening studies available as a benchmark to compare different screening methods on the same data. This urged us to give public access to the ligand and receptor databases used in this study, to the detailed experimental protocol of the study, and to the predictions made by our chemogenomic approach for each GPCR, see Tables B.2 and B.3, summarized by GPCR family in Table 2.10 and Table 2.11. This provides a benchmark which we hope will contribute to a fair evaluation of different methods and trigger new developments. This benchmark could be used to compare predictions made by other methods.

Our approach boils down to the application of well-known machine learning methods in the constructed chemogenomics space. We used a systematic way to build such a space by combining a given representation of the ligands with a given representation of the GPCRs into a binding-prediction-oriented GPCR-ligand couple representation. This allows to use any ligand or GPCR descriptor or kernel existing in the chemoinformatics or bioinformatics literature, or new ones containing other prior information as we tried to propose in this chapter. Our experiments showed that the choice of the descriptors was crucial for the prediction, and more sophisticated features for either the ligands or the GPCRs could probably further improve the performances. Among these features, improvements in the 3D ligand descriptors could probably be obtained. Indeed, 3D pharmacophore kernels did not always reach the performance of 2D kernels for the ligands. This is apparently in contradiction with the idea that protein-ligand interaction is a process occurring in the 3D space, and with previous work in our group Mahé et al. (2006). Different explanations can be proposed. First, it is possible that the bioactive conformation was not correctly predicted for all molecules used in this study. For the two ligands for which it was known, *i.e.*, retinal and 3-(isopropylamino)propan- 2-ol from PDB entries 1U19 and 2RH1 respectively, we found that the predicted conformation, using the same method as for all other molecules, was very close to the experimental conformation, with RMSD values of less than 1. However, in absence of any other information on bound ligand conformations, it is not possible to rule out the possibility that for other molecules, the prediction was not correct. Although more complete conformational space exploration for all ligands was clearly out of the scope of this work and would be a study by itself, work in this direction could improve the method. In particular, since 2D ligand-based methods are not easily suitable to make predictions outside of the molecular scaffolds for which information is known, ligand-based methods using 3D description are of particular interest, because they are expected to allow better predictions on molecules presenting diverse molecular patterns. Synergy between our method and docking would provide a means for the choice of a conformer. The principle could be to build homology models for the GPCRs, dock the molecular database in the modeled binding pockets, and derive a 3D database using, for each molecule, the conformer associated to the best docking solution. However, conformer generation and selection is a major drawback of using 3D descriptors, especially in the case of large ligands with many free torsion angles.

Various evidence suggest that, within a common global architecture, a generic binding pocket mainly involving transmembrane regions hosts agonists, antagonists and allosteric modulators. In order to identify this pocket automatically, other studies report the use of sequence alignment and the prediction of transmembrane helices. Kratochwil et al. (2005) detected hypervariable positions in transmembrane helices for identification of residues forming the binding pocket, although some positions were more conserved. Indeed, conserved residues are probably important for structural stabilization of the pocket, while variable positions are involved in ligand binding, in order to accommodate the wide spectrum of molecules that are GPCR substrates. Analyzing the positions of variable positions, these authors proposed potential binding pockets for GPCRs, and found that the corresponding residues were frequently in the GRAP mutant database for GPCRs Kristiansen et al. (1996). Interestingly, they pointed that residues at hypervariable positions were found within a distance of 6 from retinal in the rhodopsin X-Ray structure, which is also a classical distance cutoff above which it is admitted that protein-ligand interactions become negligible. Therefore, this inspired the simple and automatic method used in the present work for extracting GPCRs potential binding pockets, and our results are in good agreement with this study. It is also important to note that GPCRs are known to exist in dynamic equilibrium between inactive- and several active-state conformations Kobilka (2007), and different ligands sometimes trigger distinct conformational changes and stabilize different receptor conformations Yao et al. (2006). Taking into account receptor plasticity constitutes in itself a research domain in docking. Its use is of particular interest for screening GPCR homology models since residue positions are not exactly known. Therefore flexible docking procedures have been proposed and applied on GPCR proteins Cavasotto et al. (2003);

Chen et al. (2007). Moreover, a modeling method has been proposed to get insights on transmembrane bundle plasticity Deupi et al. (2007). In our case, receptor flexibility might influence the definition of the binding pocket, since it initially relies on the identification of residues in the two reference structures (1U19 and 2RH1) that present at least one atom situated at less than 6 of the ligand. Therefore, we made the implicit hypothesis that receptor conformational changes upon ligand binding does not drastically affect this list of residues. When more structures become available in this family of proteins, a better appreciation of such conformational rearrangements will be possible, which could be taken into account in the binding pocket definition and could help to improve the method. Kristiansen et al. (1996) found that hierarchical tree representations of GPCR subfamilies calculated with full-length GPCR sequences or with only binding pocket residues were similar, and that locally, the latter was in better agreement with functional data although their binding pocket included only 35 residues. This result is also in good agreement with our finding that the hierarchy kernel based on full length sequence (from GLIDA) and the kernel based on the binding pocket provided very similar performances. As mentioned in the Results section, it is however important to note that the kernels based on the binding pocket were built without any ligand information that could lead to some bias and artificially better performance.

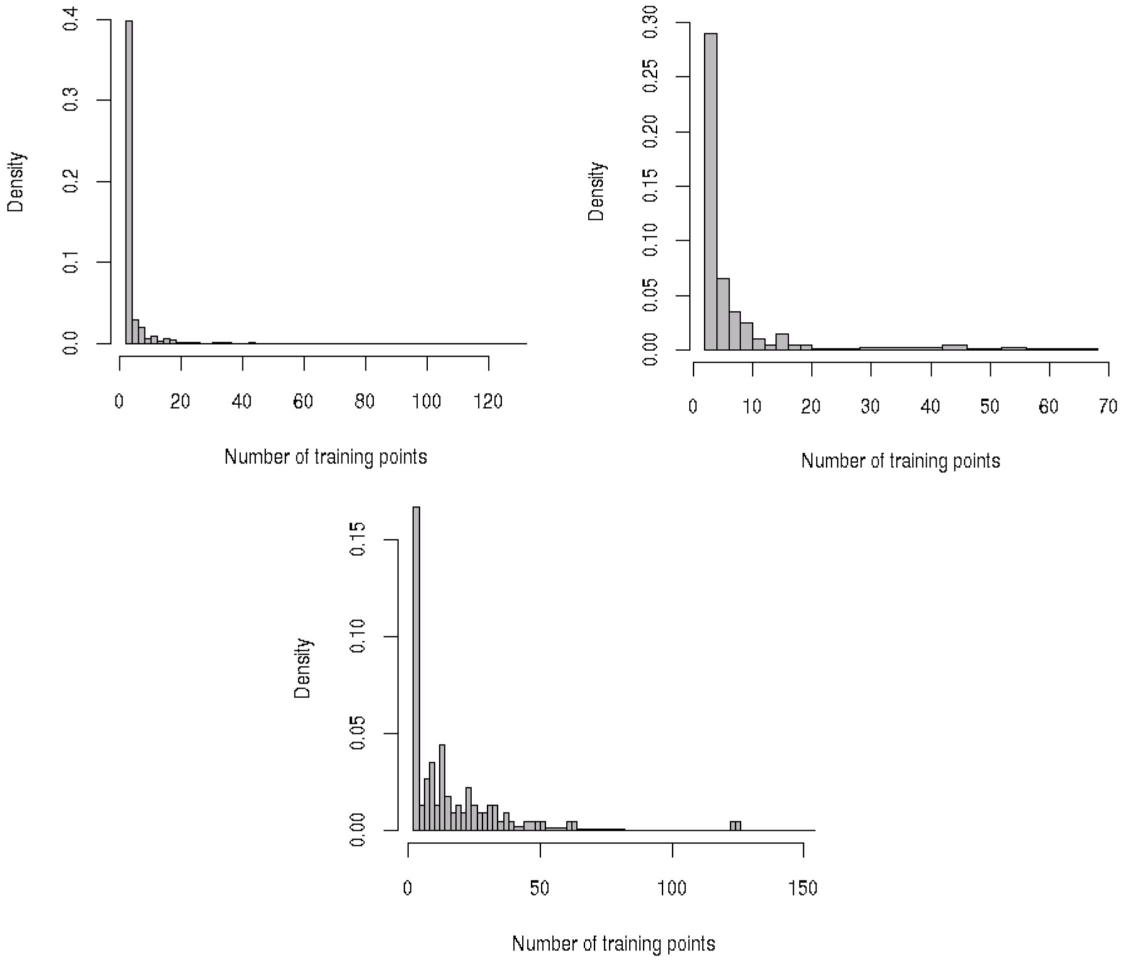

Figure 2.5: Distribution of the number of training points for a target for the enzymes, GPCR and ion channel datasets. Each bar indicates the proportion of targets in the family for which a given (x-axis) number of data points is available.

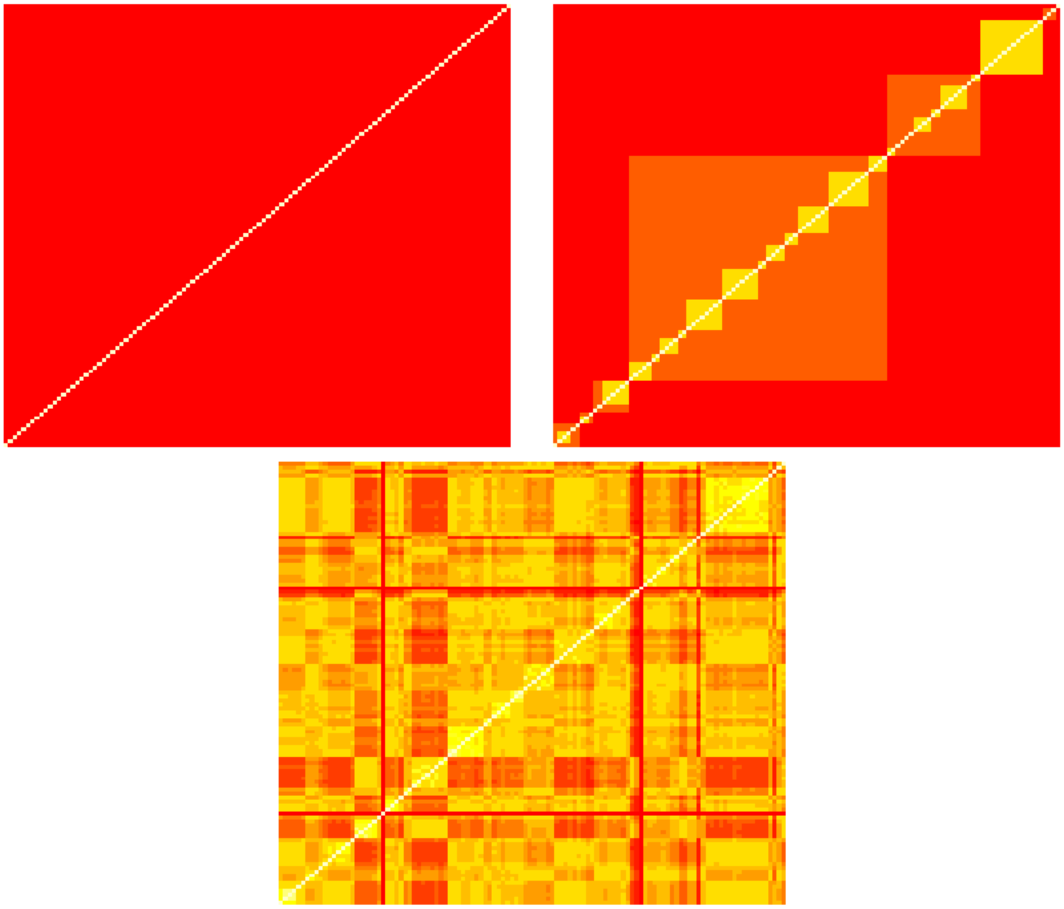

Figure 2.6: Target kernel Gram matrices (K_{tar}) for ion channels with multitask, hierarchy and local alignment kernels.

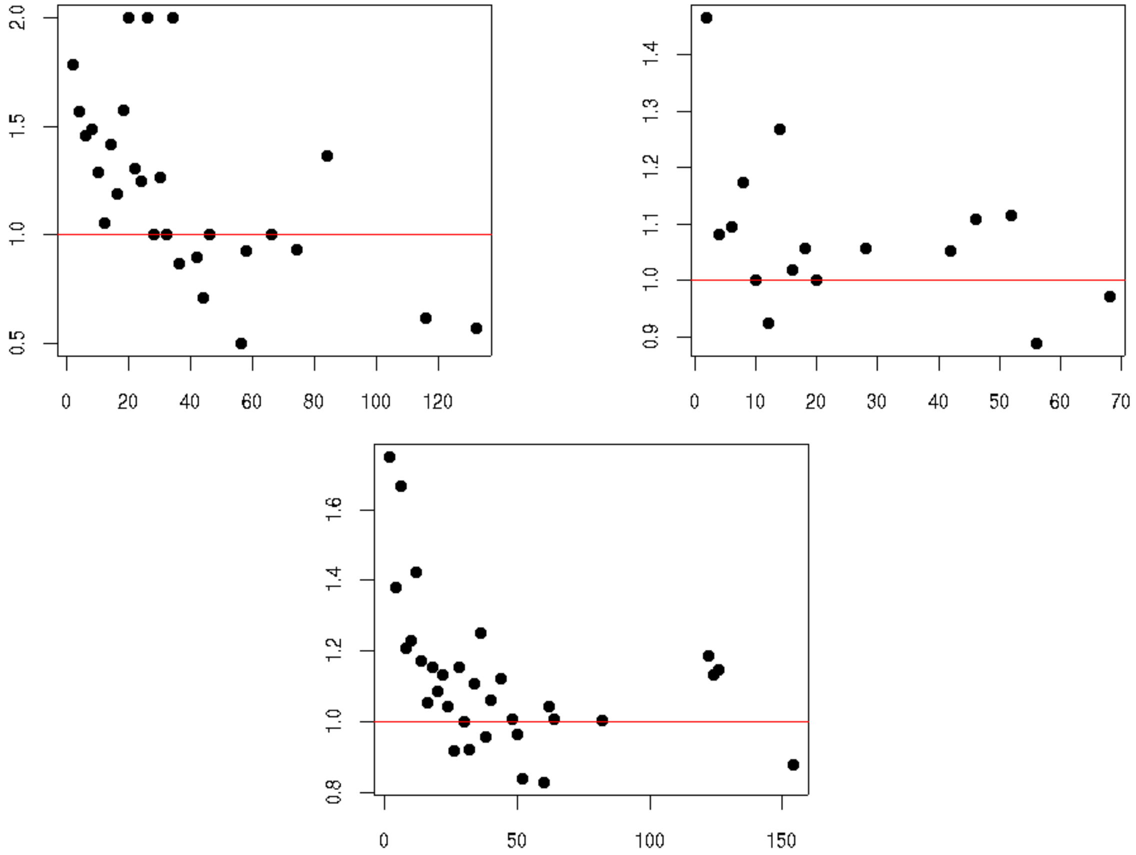

Figure 2.7: Relative improvement of the *hierarchy* kernel against the *Dirac* kernel as a function of the number of known ligands for enzymes, GPCR and ion channel datasets. Each point indicates the mean performance ratio between individual and *hierarchy* approaches across the targets of the family for which a given (x-axis) number of training points was available.

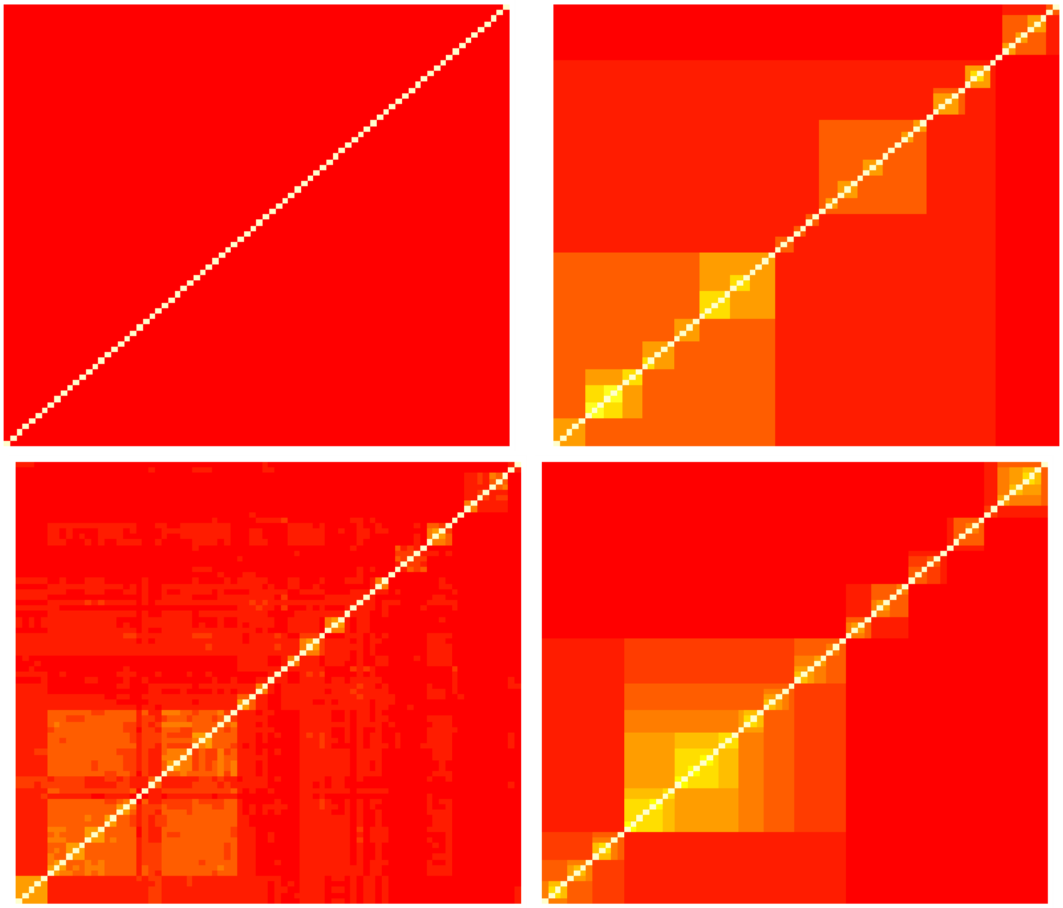

Figure 2.8: GPCR kernel Gram matrices (K_{tar}) for the GLIDA GPCR data with multitask, hierarchy, binding pocket and binding pocket hierarchy kernels.

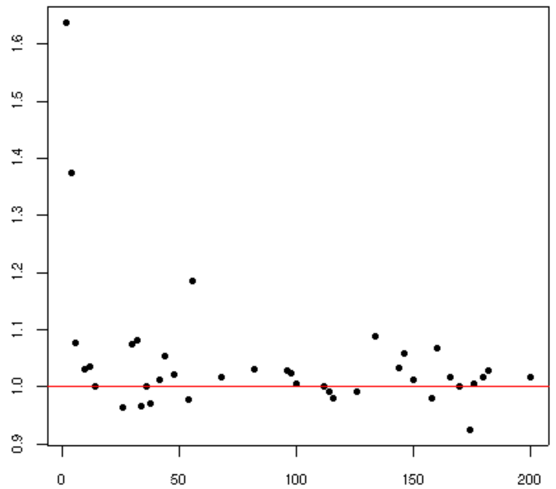

Figure 2.9: Improvement (as a performance ratio) of the hierarchy GPCR kernel against the Dirac GPCR kernel as a function of the number of training samples available. Restricted to $[2 - 200]$ samples for the sake of readability.

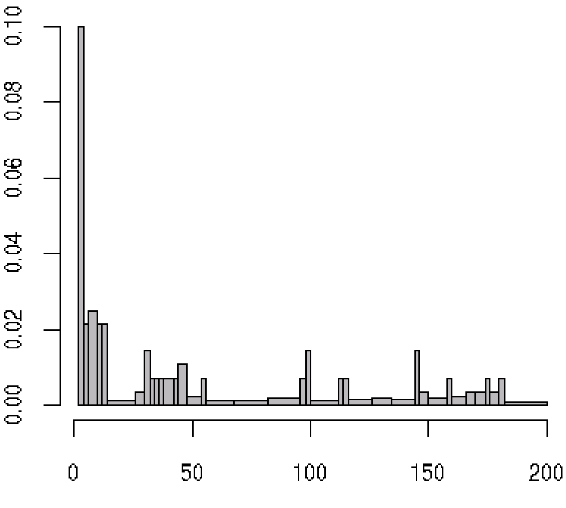

Figure 2.10: Distribution of the number of training points for a GPCR. Restricted to $[2 - 200]$ samples for the sake of readability.

Chapter 3

Clustered multi-task learning

The material presented in this section was published under a slightly different form in Jacob et al. (2009a).

3.1 Introduction

Regularization has emerged as a dominant theme in machine learning and statistics, providing an intuitive and principled tool for learning from high-dimensional data. In particular, regularization by squared Euclidean norms or squared Hilbert norms has been thoroughly studied in various settings, leading to efficient practical algorithms based on linear algebra, and to very good theoretical understanding (see, e.g., Wahba (1990); Girosi et al. (1995)). In recent years, regularization by non Hilbert norms, such as ℓ^p norms with $p \neq 2$, has also generated considerable interest for the inference of linear functions in supervised classification or regression. Indeed, such norms can sometimes both make the problem statistically and numerically better-behaved, and impose various prior knowledge on the problem. For example, the ℓ^1-norm (the sum of absolute values) imposes some of the components to be equal to zero and is widely used to estimate sparse functions Tibshirani (1996), while various combinations of ℓ^p norms can be defined to impose various sparsity patterns.

While most recent work has focused on studying the properties of simple well-known norms, we take the opposite approach in this paper. That is, assuming a given prior knowledge, how can we design a norm that will enforce it?

More precisely, we consider the problem of multi-task learning, which has recently emerged as a very promising research direction for various applications Bakker and Heskes (2003). In multi-task learning several related inference tasks are considered simultaneously, with the hope that by an appropriate sharing of information across tasks, each one may benefit from the others. When linear functions are estimated, each task is associated with a weight vector, and a common strategy to design multi-task learning algorithm is to translate some prior hypothesis about how the tasks are related to each other into constraints on the different weight vectors. For example, such constraints are typically that the weight vectors of the different tasks belong (a) to a Euclidean ball centered at the origin Evgeniou et al. (2005), which implies no sharing of information between tasks apart from the size of the different vectors, *i.e.*, the amount of regularization, (b) to a ball of unknown center Evgeniou et al. (2005), which enforces a similarity between the different weight vectors, or (c) to an unknown low-dimensional subspace Abernethy et al. (2006); Argyriou et al. (2007).

In this paper, we consider a different prior hypothesis that we believe could be more relevant in some applications: the hypothesis that *the different tasks are in fact clustered into different groups, and that the weight vectors of tasks within a group are similar to each other*. A key difference with Evgeniou et al. (2005), where a similar hypothesis is studied, is that we don't assume that the groups are known *a priori*, and in a sense our goal is both to identify the clusters and to use them for multi-task learning. An important situation that motivates this hypothesis is the case where most of the tasks are indeed related to each other, but a few "outlier" tasks are very different, in which case it may be better to impose similarity or low-dimensional constraints only to a subset of the tasks (thus forming a cluster) rather than to all tasks. Another situation of interest is when one can expect a natural organization of the tasks into clusters, such as when one wants to model the preferences of customers and believes that there are a few general types of customers with similar preferences within each type, although one does not know beforehand which customers belong to which types. Besides an improved performance if the hypothesis turns out to be correct, we also expect this approach to be able to identify the cluster structure among the tasks as a by-product of the inference step, e.g., to identify outliers or groups of customers, which can be of interest for further understanding of the structure of the problem.

In order to translate this hypothesis into a working algorithm, we follow the general strategy mentioned above which is to design a norm or a penalty over the set of weights which can be used as regularization in classical inference algorithms. We construct such a penalty by first assuming that the partition of the tasks into clusters is known, similarly to Evgeniou et al. (2005). We then attempt to optimize the objective function of the inference algorithm over the set of partitions, a strategy that has proved useful in other contexts such as multiple kernel learning Lanckriet et al. (2004b). This optimization problem over the set of partitions being computationally challenging, we propose a convex relaxation of the problem which results in an efficient algorithm.

3.2 Multi-task learning with clustered tasks

We consider m related inference tasks that attempt to learn linear functions over $\mathcal{X} = \mathbb{R}^d$ from a training set of input/output pairs $(x_i, y_i)_{i=1,\dots,n}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. In the case of binary classification we usually take $\mathcal{Y} = \{-1, +1\}$, while in the case of regression we take $\mathcal{Y} = \mathbb{R}$. Each training example (x_i, y_i) is associated to a particular task $t \in [1, m]$, and we denote by $\mathcal{I}(t) \subset [1, n]$ the set of indices of training examples associated to the task t. Our goal is to infer m linear functions $f_t(x) = w_t^\top x$, for $t = 1, \dots, m$, associated to the different tasks. We denote by $W = (w_1 \dots w_m)$ the $d \times m$ matrix whose columns are the successive vectors we want to estimate.

We fix a loss function $l : \mathbb{R} \times \mathcal{Y} \mapsto \mathbb{R}$ that quantifies by $l(f(x), y)$ the cost of predicting $f(x)$ for the input x when the correct output is y. Typical loss functions include the square error in regression $l(u, y) = \frac{1}{2}(u - y)^2$ or the hinge loss in binary classification $l(u, y) = \max(0, 1 - uy)$ with $y \in \{-1, 1\}$. The empirical risk of a set of linear classifiers given in the matrix W is then defined as the average loss over the training set:

$$\ell(W) = \frac{1}{n} \sum_{t=1}^{m} \sum_{i \in \mathcal{I}(t)} l(w_t^\top x_i, y_i) \,. \tag{3.1}$$

In the sequel, we will often use the $m{\times}1$ vector $\mathbf{1}$ composed of ones, the $m{\times}m$ projection matrices $U = \mathbf{1}\mathbf{1}^\top/m$ whose entries are all equal to $1/m$, as well as the projection matrix $\Pi = I - U$.

In order to learn simultaneously the m tasks, we follow the now well-established approach which looks for a set of weight vectors W that minimizes the empirical risk regularized by a penalty functional, *i.e.*, we consider the problem:

$$\min_{W \in \mathbb{R}^{d \times m}} \ell(W) + \lambda \Omega(W) \,, \tag{3.2}$$

where $\Omega(W)$ can be designed from prior knowledge to constrain some sharing of information between tasks. For example, Evgeniou et al. (2005) suggests to penalize both the norms of the w_i's and their variance, *i.e.*, to consider a function of the form:

$$\Omega_{variance}(W) = \|\bar{w}\|^2 + \frac{\beta}{m} \sum_{i=1}^{m} \|w_i - \bar{w}\|^2 \,, \tag{3.3}$$

where $\bar{w} = \left(\sum_{i=1}^{n} w_i\right)/m$ is the mean weight vector. This penalty enforces a clustering of the $w_i's$ towards their mean when β increases. Alternatively, Argyriou et al. (2007) propose to penalize the trace norm of W:

$$\Omega_{trace}(W) = \sum_{i=1}^{\min(d,m)} \sigma_i(W) \,, \tag{3.4}$$

where $\sigma_1(W), \ldots, \sigma_{\min(d,m)}(W)$ are the successive singular values of W. This enforces a low-rank solution in W, *i.e.*, constrains the different w_i's to live in a low-dimensional subspace.

Here we would like to define a penalty function $\Omega(W)$ that encodes as prior knowledge that tasks are clustered into $r < m$ groups. To do so, let us first assume that we know beforehand the clusters, *i.e.*, we have a partition of the set of tasks into r groups. In that case we can follow an approach proposed by Evgeniou et al. (2005) which for clarity we rephrase with our notations and slightly generalize now. For a given cluster $c \in [1, r]$, let us denote $\mathcal{I}(c) \subset [1, m]$ the set of tasks in c, $m_c = |\mathcal{I}(c)|$ the number of tasks in the cluster c, and E the $m \times r$ binary matrix which describes the cluster assignment for the m tasks, *i.e.*, $E_{ij} = 1$ if task i is in cluster j, 0 otherwise. Let us further denote by $\bar{w}_c = (\sum_{i \in \mathcal{I}(c)} w_i)/m_c$ the average weight vector for the tasks in c, and recall that $\bar{w} = \left(\sum_{i=1}^{m} w_i\right)/m$ denotes the average weight vector over all tasks. Finally it will be convenient to introduce the matrix

$M = E(E^\top E)^{-1}E^\top$. M can also be written $I - L$, where L is the normalized Laplacian of the graph G whose nodes are the tasks connected by an edge if and only if they are in the same cluster. Then we can define three semi-norms of interest on W that quantify different orthogonal aspects:

- A global penalty, which measures on average how large the weight vectors are:

$$\Omega_{mean}(W) = n\|\bar{w}\|^2 = \mathrm{tr} W U W^\top\,.$$

- A measure of between-cluster variance, which quantifies how close to each other the different clusters are:

$$\Omega_{between}(W) = \sum_{c=1}^{r} m_c \|\bar{w}_c - \bar{w}\|^2 = \mathrm{tr} W(M-U)W^\top.$$

- A measure of within-cluster variance, which quantifies the compactness of the clusters:

$$\Omega_{within}(W) = \sum_{c=1}^{r} \left\{ \sum_{i\in\mathcal{I}(c)} \|w_i - \bar{w}_c\|^2 \right\} = \mathrm{tr} W(I-M)W^\top\,.$$

We note that both $\Omega_{between}(W)$ and $\Omega_{within}(W)$ depend on the particular choice of clusters E, or equivalently of M. We now propose to consider the following general penalty function:

$$\Omega(W) = \varepsilon_M \Omega_{mean}(W) + \varepsilon_B \Omega_{between}(W) + \varepsilon_W \Omega_{within}(W)\,, \tag{3.5}$$

where $\varepsilon_M, \varepsilon_B$ and ε_W are non-negative parameters that can balance the importance of the components of the penalty. Plugging this quadratic penalty into (3.2) leads to the general problem:

$$\min_{W\in\mathbb{R}^{d\times m}} \ell(W) + \lambda \mathrm{tr} W \Sigma(M)^{-1} W^\top\,, \tag{3.6}$$

where

$$\Sigma(M)^{-1} = \varepsilon_M U + \varepsilon_B (M-U) + \varepsilon_W (I-M)\,. \tag{3.7}$$

Here we use the notation $\Sigma(M)$ to insist on the fact that this quadratic penalty depends on the cluster structure through the matrix M. Observing that the matrices U, $M-U$ and

$I - M$ are orthogonal projections onto orthogonal supplementary subspaces, we easily get from (3.7):

$$\Sigma(M) = \varepsilon_M^{-1}U + \varepsilon_B^{-1}(M-U) + \varepsilon_W^{-1}(I-M) = \varepsilon_W^{-1}I + (\varepsilon_M^{-1} - \varepsilon_B^{-1})U + (\varepsilon_B^{-1} - \varepsilon_W^{-1})M\,. \quad (3.8)$$

By choosing particular values for $\varepsilon_M, \varepsilon_B$ and ε_W we can recover several situations, In particular:

- For $\varepsilon_W = \varepsilon_B = \varepsilon_M = \varepsilon$, we simply recover the Frobenius norm of W, which does not put any constraint on the relationship between the different tasks:

$$\Omega(W) = \varepsilon \mathrm{tr} W W^\top = \varepsilon \sum_{i=1}^m \|w_i\|^2\,.$$

- For $\varepsilon_W = \varepsilon_B > \varepsilon_M$, we recover the penalty of Evgeniou et al. (2005) without clusters:

$$\Omega(W) = \mathrm{tr} W \left(\varepsilon_M U + \varepsilon_B (I - U)\right) W^\top = \varepsilon_M n \|\bar{w}\|^2 + \varepsilon_B \sum_{i=1}^m \|w_i - \bar{w}\|^2\,.$$

  In that case, a global similarity between tasks is enforced, in addition to the general constraint on their mean. The structure in clusters plays no role since the sum of the between- and within-cluster variance is independent of the particular choice of clusters.

- For $\varepsilon_W > \varepsilon_B = \varepsilon_M$ we recover the penalty of Evgeniou et al. (2005) with clusters:

$$\Omega(W) = \mathrm{tr} W \left(\varepsilon_M M + \varepsilon_W (I - M)\right) W^\top = \varepsilon_M \sum_{c=1}^r \left\{ m_c \|\bar{w}_c\|^2 + \frac{\varepsilon_W}{\varepsilon_M} \sum_{i \in \mathcal{I}(c)} \|w_i - \bar{w}_c\|^2 \right\}\,.$$

In order to enforce a cluster hypothesis on the tasks, we therefore see that a natural choice is to take $\varepsilon_W > \varepsilon_B > \varepsilon_M$ in (3.5). This would have the effect of penalizing more the within-cluster variance than the between-cluster variance, hence promoting compact clusters. Of course, a major limitation at this point is that we assumed the cluster structure known *a* *priori* (through the matrix E, or equivalently M). In many cases of interest, we would like instead to learn the cluster structure itself from the data. We propose to learn the cluster structure in our framework by optimizing our objective function (3.6) both in W and M, *i.e.*, to consider the problem:

$$\min_{W \in \mathbb{R}^{d \times m}, M \in \mathcal{M}_r} \ell(W) + \lambda \mathrm{tr} W \Sigma(M)^{-1} W^\top , \tag{3.9}$$

where $\mathcal{M}_r$ denotes the set of matrices $M = E(E^\top E)^{-1} E^\top$ defined by a clustering of the m tasks into r clusters and $\Sigma(M)$ is defined in (3.8). Denoting by $\mathcal{S}_r = \{\Sigma(M) : M \in \mathcal{M}_r\}$ the corresponding set of positive semidefinite matrices, we can equivalently rewrite the problem as:

$$\min_{W \in \mathbb{R}^{d \times m}, \Sigma \in \mathcal{S}_r} \ell(W) + \lambda \mathrm{tr} W \Sigma^{-1} W^\top . \tag{3.10}$$

The objective function in (3.10) is jointly convex in $W \in \mathbb{R}^{d \times m}$ and $\Sigma \in \mathcal{S}_+^m$, the set of $m \times m$ positive semidefinite matrices, however the (finite) set $\mathcal{S}_r$ is not convex, making this problem intractable. We are now going to propose a convex relaxation of (3.10) by optimizing over a convex set of positive semidefinite matrices that contains $\mathcal{S}_r$.

3.3 Convex relaxation

In order to formulate a convex relaxation of (3.10), we observe that in the penalty term (3.5) the cluster structure only contributes to the second and third terms $\Omega_{between}(W)$ and $\Omega_{within}(W)$, and that these penalties only depend on the centered version of W. In terms of matrices, only the last two terms of $\Sigma(M)^{-1}$ in (3.7) depend on M, *i.e.*, on the clustering, and these terms can be re-written as:

$$\varepsilon_B(M - U) + \varepsilon_W(I - M) = \Pi(\varepsilon_B M + \varepsilon_W(I - M))\Pi. \tag{3.11}$$

Indeed, it is easy to check that $M - U = M\Pi = \Pi M \Pi$, and that $I - M = I - U - (M - U) = \Pi - \Pi M \Pi = \Pi(I - M)\Pi$. Intuitively, multiplying by Π on the right (*resp.* on the left) centers the rows (*resp.* the columns) of a matrix, and both $M - U$ and $I - M$ are row- and column-centered.

To simplify notations, let us introduce $\widetilde{M} = \Pi M \Pi$. Plugging (3.11) in (3.7) and (3.9), we get the penalty

$$\mathrm{tr} W\Sigma(M)^{-1}W^\top = \varepsilon_M \left(\mathrm{tr} W^\top W U\right) + (W\Pi)(\varepsilon_B \widetilde{M} + \varepsilon_W (I - \widetilde{M}))(W\Pi)^\top, \tag{3.12}$$

in which, again, only the second part needs to be optimized with respect to the clustering M. Denoting $\Sigma_c^{-1}(M) = \varepsilon_B \widetilde{M} + \varepsilon_W (I - \widetilde{M})$, one can express $\Sigma_c(M)$, using the fact that $\widetilde{M}$ is a projection:

$$\Sigma_c(M) = \left(\varepsilon_B^{-1} - \varepsilon_W^{-1}\right)\widetilde{M} + \varepsilon_W^{-1} I. \tag{3.13}$$

Σ_c is characterized by $\widetilde{M} = \Pi M \Pi$, that is discrete by construction, hence the non-convexity of $\mathcal{S}_r$. We have the natural constraints $M \geq 0$ (*i.e.*, $\widetilde{M} \geq -U$), $0 \preceq M \preceq I$ (*i.e.*, $0 \preceq \widetilde{M} \preceq \Pi$) and $\mathrm{tr} M = r$ (*i.e.*, $\mathrm{tr}\widetilde{M} = r - 1$). A possible convex relaxation of the discrete set of matrices $\widetilde{M}$ is therefore $\{\widetilde{M} : 0 \preceq \widetilde{M} \preceq I,\ \mathrm{tr}\widetilde{M} = r - 1\}$. This gives an equivalent convex set $\mathcal{S}_c$ for Σ_c, namely:

$$\mathcal{S}_c = \left\{\Sigma_c \in \mathcal{S}_+^m : \alpha I \preceq \Sigma_c \preceq \beta I, \mathrm{tr}\Sigma_c = \gamma\right\}\ , \tag{3.14}$$

with $\alpha = \varepsilon_W^{-1}$, $\beta = \varepsilon_B^{-1}$ and $\gamma = (m - r + 1)\varepsilon_W^{-1} + (r - 1)\varepsilon_B^{-1}$. Incorporating the first part of the penalty (3.12) into the empirical risk term by defining $\ell_c(W) = \lambda \ell(W) + \varepsilon_M \left(\mathrm{tr} W^\top W U\right)$, we are now ready to state our relaxation of (3.10):

$$\min_{W \in \mathbb{R}^{d\times m}, \Sigma_c \in \mathcal{S}_c} \ell_c(W) + \lambda \mathrm{tr} W \Pi \Sigma_c^{-1} (W\Pi)^\top\ . \tag{3.15}$$

3.3.1 Reinterpretation in terms of norms

We denote $\|W\|_c^2 = \min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W \Sigma_c^{-1} W^T$ the *cluster norm* (CN). For any convex set $\mathcal{S}_c$, we obtain a norm on W (that we apply here to its centered version). By putting some different constraints on the set $\mathcal{S}_c$, we obtain different norms on W, and in fact all previous multi-task formulations may be cast in this way, *i.e.*, by choosing a specific set of positive matrices $\mathcal{S}_c$ (*e.g.*, trace constraint for the trace norm, and simply a singleton for the Frobenius norm). Thus, designing norms for multi-task learning is equivalent to designing a set of positive matrices. In this paper, we have investigated a specific set adapted for clustered-tasks, but other sets could be designed in other situations.

Note that we have selected a simple *spectral* convex set $\mathcal{S}_c$ in order to make the optimization simpler in Section 3.3.3, but we could also add some additional constraints that encode the point-wise positivity of the matrix M. Finally, when $r = 1$ (one cluster) and $r = m$ (one cluster per task), we get back the formulation of Evgeniou et al. (2005).

3.3.2 Reinterpretation as a convex relaxation of K-means

In this section we show that the semi-norm $\|W\Pi\|_c^2$ that we have designed earlier, can be interpreted as a convex relaxation of K-means on the tasks Deodhar and Ghosh (2007). Indeed, given $W \in \mathbb{R}^{d\times m}$, K-means aims to decompose it in the form $W = \mu E^\top$ where $\mu \in \mathbb{R}^{d\times r}$ are cluster centers and E represents a partition. Given E, μ is found by minimizing $\min_\mu \|W^\top - E\mu^\top\|_F^2$. Thus, a natural strategy outlined by Deodhar and Ghosh (2007), is to alternate between optimizing μ, the partition E and the weight vectors W. We now show that our convex norm is obtained when minimizing in closed form with respect to μ and relaxing.

By translation invariance, this is equivalent to minimizing $\min_\mu \|\Pi W^\top - \Pi E\mu^\top\|_F^2$. If we add a penalization on μ of the form $\lambda \mathrm{tr} E^\top E\mu\mu^\top$, then a short calculation shows that the minimum with respect to μ (*i.e.*, after optimization of the cluster centers) is equal to

$$\mathrm{tr} W\Pi(\Pi E(E^\top E)^{-1}E^\top\Pi/\lambda + I)^{-1}\Pi W^\top = \mathrm{tr} W\Pi(\Pi M\Pi/\lambda + I)^{-1}\Pi W^\top.$$

By comparing with Eq. (3.13), we see that our formulation is indeed a convex relaxation of K-means.

3.3.3 Primal optimization

Let us now show in more details how (3.15) can be solved efficiently. Whereas a dual formulation could be easily derived following Lanckriet et al. (2004b), a direct approach is to rewrite (3.15) as

$$\min_{W\in\mathbb{R}^{d\times m}} \left(\ell_c(W) + \min_{\Sigma_c\in\mathcal{S}_c} \mathrm{tr} W\Pi\Sigma_c^{-1}(W\Pi)^\top \right) \tag{3.16}$$

which, if ℓ_c is differentiable, can be directly optimized by gradient-based methods on W since $\|W\Pi\|_c^2 = \min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W\Pi\Sigma_c^{-1}(W\Pi)^\top$ is a quadratic semi-norm of $W\Pi$. This regularization term $\mathrm{tr}W\Pi\Sigma_c^{-1}(W\Pi)^\top$ can be computed efficiently using a semi-closed form. Indeed, since Σ_c as defined in (3.14) is a spectral set (*i.e.*, it does depend only on eigenvalues of covariance matrices), we obtain a function of the singular values of $W\Pi$ (or equivalently the eigenvalues of $W\Pi W^\top$):

$$\min_{\Sigma_c \in \mathcal{S}_c} \mathrm{tr} W\Pi\Sigma_c^{-1}(W\Pi)^\top = \min_{\lambda \in \mathbb{R}^m,\ \alpha \leq \lambda_i \leq \beta,\ \lambda \mathbf{1} = \gamma,\ V \in \mathcal{O}^m} \mathrm{tr} W\Pi V \operatorname{diag}(\lambda)^{-1} V^\top (W\Pi)^\top,$$

where $\mathcal{O}^m$ is the set of orthogonal matrices in $\mathbb{R}^{m \times m}$. The optimal V is the matrix of the eigenvectors of $W\Pi W^\top$, and we obtain the value of the objective function at the optimum:

$$\min_{\Sigma \in S} \mathrm{tr} W\Pi\Sigma^{-1}(W\Pi)^\top = \min_{\lambda \in \mathbb{R}^m,\ \alpha \leq \lambda_i \leq \beta,\ \lambda \mathbf{1} = \gamma} \sum_{i=1}^m \frac{\sigma_i^2}{\lambda_i},$$

where σ and λ are the vectors containing the singular values of $W\Pi$ and Σ respectively. Now, we simply need to be able to compute this function of the singular values.

The only coupling in this formulation comes from the trace constraint. The Lagrangian corresponding to this constraint is:

$$\mathcal{L}(\lambda, \nu) = \sum_{i=1}^m \frac{\sigma_i^2}{\lambda_i} + \nu \left(\sum_{i=1}^m \lambda_i - \gamma \right). \tag{3.17}$$

For $\nu \leq 0$, this is a decreasing function of λ_i, so the minimum on $\lambda_i \in [\alpha, \beta]$ is reached for $\lambda_i = \beta$. The dual function is then a linear non-decreasing function of ν (since $\alpha \leq \gamma/m \leq \beta$ from the definition of α, β, γ in (3.14)), which reaches it maximum value (on $\nu \leq 0$) at $\nu = 0$. Let us therefore now consider the dual for $\nu \geq 0$. (3.17) is then a convex function of λ_i. Canceling its derivative with respect to λ_i gives that the minimum in $\lambda \in \mathbb{R}$ is reached for $\lambda_i = \sigma_i/\sqrt{\nu}$. Now this may not be in the constraint set (α, β), so if $\sigma_i < \alpha\sqrt{\nu}$ then the minimum in $\lambda_i \in [\alpha, \beta]$ of (3.17) is reached for $\lambda_i = \alpha$, and if $\sigma_i > \beta\sqrt{\nu}$ it is reached for $\lambda_i = \beta$. Otherwise, it is reached for $\lambda_i = \sigma_i/\sqrt{\nu}$. Reporting this in (3.17), the dual problem is therefore

$$\max_{\nu \geq 0} \sum_{i, \alpha\sqrt{\nu} \leq \sigma_i \leq \beta\sqrt{\nu}} 2\sigma_i\sqrt{\nu} + \sum_{i, \sigma_i < \alpha\sqrt{\nu}} \left(\frac{\sigma_i^2}{\alpha} + \nu\alpha\right) + \sum_{i, \beta\sqrt{\nu} < \sigma_i} \left(\frac{\sigma_i^2}{\beta} + \nu\beta\right) - \nu\gamma\,. \tag{3.18}$$

Since a closed form for this expression is known for each fixed value of ν, one can obtain $\|W\Pi\|_c^2$ (and the eigenvalues of Σ^*) by Algorithm 1. The cancellation condition in Algorithm 1 is that the value canceling the derivative belongs to (a, b), *i.e.*,

Algorithm 1 Computing $\|A\|_c^2$

```
Require: $A, \alpha, \beta, \gamma$.
Ensure: $\|A\|_c^2$, $\lambda^*$.
  Compute the singular values $\sigma_i$ of $A$.
  Order the $\frac{\sigma_i^2}{\alpha^2}, \frac{\sigma_i^2}{\beta^2}$ in a vector $I$ (with an additional 0 at the beginning).
  for all interval $(a, b)$ of $I$ do
    if $\frac{\partial \mathcal{L}(\lambda^*, \nu)}{\partial \nu}$ is canceled on $\nu \in (a, b)$ then
      Replace $\nu^*$ in the dual function $\mathcal{L}(\lambda^*, \nu)$ to get $\|A\|_c^2$, compute $\lambda^*$ on $(a, b)$.
      return $\|A\|_c^2$, $\lambda^*$.
    end if
  end for
```

$$\nu = \left(\frac{\sum_{i, \alpha\sqrt{\nu} \leq \sigma_i \leq \beta\sqrt{\nu}} \sigma_i}{\gamma - (\alpha n^- + \beta n^+)}\right)^2 \in (a, b)\,,$$

where n^- and n^+ are the number of $\sigma_i < \alpha\sqrt{\nu}$ and $\sigma_i > \beta\sqrt{\nu}$ respectively. Denoting $\|A\|_c^2 = F(A, \Sigma^*(A))$, $\nabla_A F = \partial_A F + \partial_\Sigma F \partial_A \Sigma$ cannot be computed because of the non-differentiable constraints on Σ for F. We followed an alternative direction, using only the $\partial_A F$ part.

3.4 Experiments

3.4.1 Artificial data

We generated synthetic data consisting of two clusters of two tasks. The tasks are vectors of $\mathbb{R}^d,\ d = 30$. For each cluster, a center $\bar{w}_c$ was generated in $\mathbb{R}^{d-2}$, so that the two clusters be orthogonal. More precisely, each $\bar{w}_c$ had $(d-2)/2$ random features randomly drawn from $\mathcal{N}(0, \sigma_r^2),\ \sigma_r^2 = 900$, and $(d-2)/2$ zero features. Then, each tasks t was computed as $w_t + \bar{w}_c(t)$, where $c(t)$ was the cluster of t. w_t had the same zero feature as its cluster center, and the other features were drawn from $\mathcal{N}(0, \sigma_c^2),\ \sigma_c^2 = 16$. The last two features were non-zero for all the tasks and drawn from $\mathcal{N}(0, \sigma_c^2)$. For each task, 2000 points were generated and a normal noise of variance $\sigma_n^2 = 150$ was added.

In a first experiment, we compared our cluster norm $\|.\|_c^2$ with the single-task learning given by the Frobenius norm, and with the trace norm, that corresponds to the assumption that the tasks live in a low-dimension space. The multi-task kernel approach being a special case of CN, its performance will always be between the performance of the single task and the performance of CN.

In a second setting, we compare CN to alternative methods that differ in the way they learn Σ:

- The *True metric* approach, that simply plugs the actual clustering in E and optimizes W using this fixed metric. This necessitates to know the true clustering *a priori*, and can be thought of like a golden standard.

- The *k-means* approach, that alternates between optimizing the tasks in W given the metric Σ and re-learning Σ by clustering the tasks w_i Deodhar and Ghosh (2007). The clustering is done by a k-means run 3 times. This is a non convex approach, and different initialization of k-means may result in different local minima.

We also tried one run of CN followed by a run of *True metric* using the learned Σ reprojected in $\mathcal{S}_r$ by rounding, *i.e.*, by performing k-means on the eigenvectors of the learned Σ (*Reprojected* approach), and a run of *k-means* starting from the relaxed solution (*CNinit* approach).

Only the first method requires to know the true clustering a priori, all the other methods can be run without any knowledge of the clustering structure of the tasks.

Each method was run with different numbers of training points. The training points were equally separated between the two clusters and for each cluster, $5/6$th of the points were used for the first task and $1/6$th for the second, in order to simulate a natural setting were some tasks have fewer data. We used the 2000 points of each task to build 3 training folds, and the remaining points were used for testing. We used the mean RMSE across the tasks as a criterion, and a quadratic loss for $\ell(W)$.

The results of the first experiment are shown on Figure 3.1 (left). As expected, both multi-task approaches perform better than the approach that learns each task independently. CN penalization on the other hand always gives better testing error than the trace norm penalization, with a stronger advantage when very few training points are available. When more training points become available, all the methods give more and more similar performances. In particular, with large samples, it is not useful anymore to use a multi-task approach.

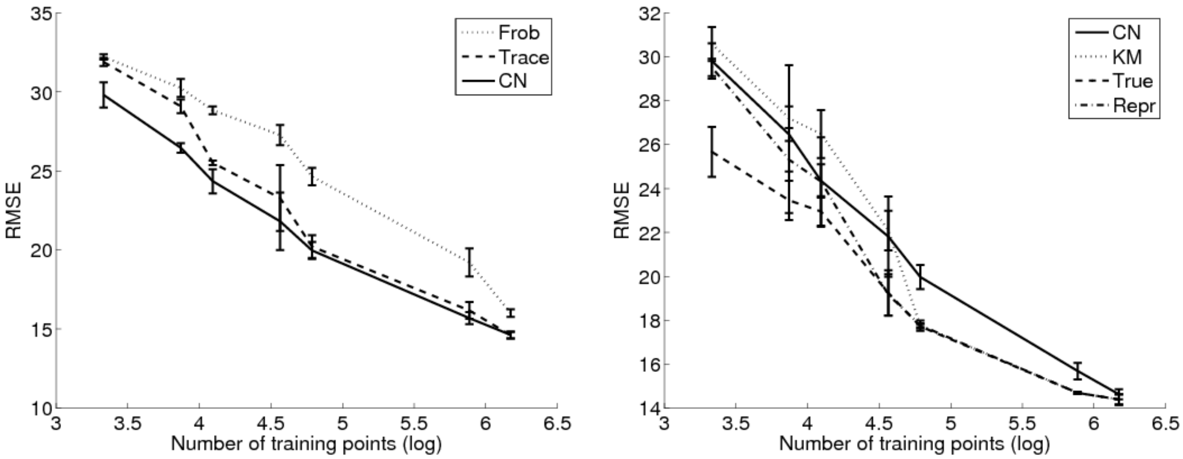

Figure 3.1: RMSE versus number of training points for the tested methods.

Figure 3.1 (right) shows the results of the second experiment. Using the true metric always gives the best results. For 28 training points, no method recovers the correct clustering structure, as displayed on Figure 3.2, although CN performs slightly better than the

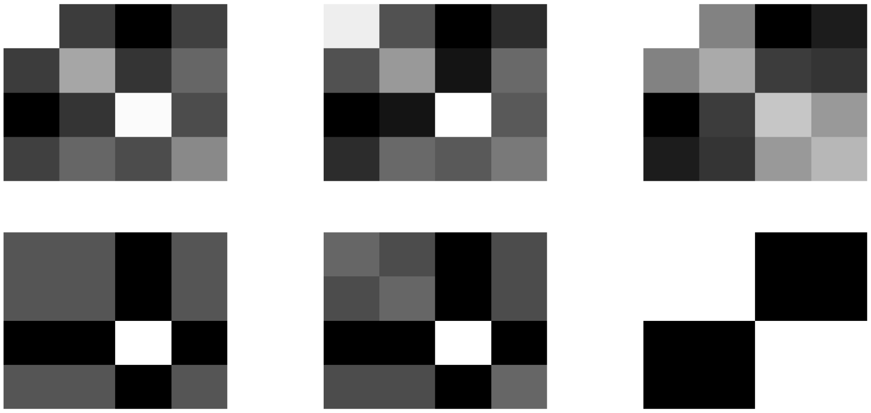

Figure 3.2: Recovered Σ with CN (upper line) and k-means (lower line) for 28, 50 and 100 points.

k-means approach since the metric it learns is more diffuse. For 50 training points, CN performs much better than the *k-means* approach, which completely fails to recover the clustering structure as illustrated by the Σ learned for 28 and 50 training points on Figure 3.2. In the latter setting, CN partially recovers the clusters. When more training points become available, the *k-means* approach perfectly recovers the clustering structure and outperforms the relaxed approach. The reprojected approach, on the other hand, performs always as well as the best of the two other methods. The CNinit approach results are not displayed since the are the same as for the reprojected method.

3.4.2 MHC-I binding data

We also applied our method to the IEDB MHC-I peptide binding benchmark proposed in Peters et al. (2006). This database contains binding affinities of various peptides, *i.e.*, short amino-acid sequences, with different MHC-I molecules. This binding process is central in the immune system, and predicting it is crucial, for example to design vaccines. The affinities are thresholded to give a prediction problem. Each MHC-I molecule is considered as a task, and the goal is to predict whether a peptide binds a molecule. We used an orthogonal coding of the amino acids to represent the peptides and balanced the data by keeping only one negative example for each positive point, resulting in 15236 points involving 35

Table 3.1: Prediction error for the 10 molecules with less than 200 training peptides in IEDB.

Method	Pooling	Frobenius	MT kernel	Trace norm	Cluster Norm
Test error	$26.53\% \pm 2.0$	$11.62\% \pm 1.4$	$10.10\% \pm 1.4$	$9.20\% \pm 1.3$	$8.71\% \pm 1.5$

different molecules. We chose a logistic loss for $\ell(W)$.

Multi-task learning approaches have already proved useful for this problem, see for example Heckerman et al. (2007); Jacob and Vert (2008a). Besides, it is well known in the vaccine design community that some molecules can be grouped into empirically defined *supertypes* known to have similar binding behaviors.

Jacob and Vert (2008a) showed in particular that the multi-task approaches were very useful for molecules with few known binders. Following this observation, we consider the mean error on the 10 molecules with less than 200 known ligands, and report the results in Table 3.1. We did not select the parameters by internal cross validation, but chose them among a small set of values in order to avoid overfitting. More accurate results could arise from such a cross validation, in particular concerning the number of clusters (here we limited the choice to 2 or 10 clusters).

The pooling approach simply considers one global prediction problem by pooling together the data available for all molecules. The results illustrate that it is better to consider individual models than one unique pooled model.On the other hand, all the multitask approaches improve the accuracy, the cluster norm giving the best performance. The learned Σ, however, did not recover the known supertypes, although it may contain some relevant information on the binding behavior of the molecules.

3.5 Conclusion

We have presented a convex approach to clustered multi-task learning, based on the design of a dedicated norm. Promising results were presented on synthetic examples and on the IEDB dataset. We are currently investigating more refined convex relaxations and the natural extension to non-linear multi-task learning as well as the inclusion of specific features on the tasks, which has shown to improve performance in other settings Abernethy et al. (2006).

Chapter 4

Structured priors for expression data analysis

This work was presented under a slightly modified form in Jacob et al. (2009b).

4.1 Introduction

Estimation of sparse linear models by the minimization of an empirical error penalized by a regularization term is a very popular and successful approach in statistics and machine learning. Controlling the trade-off between data fitting and regularization, one can obtain estimators with good statistical properties, even in very large dimension. Moreover, sparse classifiers lend themselves particularly well to interpretation, which is often of primary importance in many applications such as biology or social sciences. A popular example is the penalization of a ℓ_2 criterion by the ℓ_1 norm of the estimator, known as *Lasso* Tibshirani (1996) or *basis pursuit* Chen et al. (1998). Interestingly, the Lasso is able to recover the exact support of a sparse model from data generated by this model if the covariates are not too correlated Zhao and Yu (2006); Wainwright (2006).

While the ℓ_1 norm penalty leads to sparse models, it does not contain any prior information about, *e.g.*, possible groups of covariates that one may wish to see selected jointly. Several authors have recently proposed new penalties to enforce the estimation of models with specific sparsity patterns. For example, when the covariates are partitioned into groups, the *group lasso* leads to the selection of groups of covariates Yuan and Lin (2006). The group lasso penalty for a model, also called ℓ_1/ℓ_2 penalty, is the sum (*i.e.*, ℓ_1 norm) of the ℓ_2 norms of the restrictions of the model to the different groups of covariates. It recovers the support of a model if the support is a union of groups and if covariates of different groups are not too correlated. It can be generalized to an infinite-dimensional setting Bach (2008a). Other variants of the group lasso include joint selection of covariates for multi-task learning Obozinski et al. (2009) and penalties to enforce hierarchical selection of covariates, *e.g.*, when one has a hierarchy over the covariates and wants to select covariates only if their ancestors in the hierarchy are also selected Zhao et al. (2009); Bach (2009).

In this chapter, we are interested in a more general situation. We assume that either (i) groups of covariates are given, potentially with overlap between the groups, and we wish to estimate a model whose support is a union of groups, or (ii) that a graph with covariates as vertices is given, and we wish to estimate a model whose support contains covariates which tend to be connected to each others on the graph. Although quite general, this framework is motivated in particular by applications in bioinformatics, when we have to solve classification or regression problems with few samples in high dimension, such as predicting the class of a tumour from gene expression measurements with microarrays, and simultaneously select a few genes to establish a predictive signature Roth (2002); Ghosh and Chinnaiyan (2005). Selecting a few genes that either belong to the same functional groups, where the groups are given *a priori* and may overlap, or tend to be connected to each other in a given biological network, could then lead to increased interpretability of the signature and potential better performances Rapaport et al. (2007).

To reach this goal, we propose and study a new penalty which generalizes the ℓ_1/ℓ_2 norm to overlapping groups for the first case, and propose to cast the problem of selecting connected covariates in a graph as the problem of selecting a union of overlapping groups, with adequate definition of groups, for the second case. We mention various properties of this penalty, and provide conditions for the consistency of support estimation in the regression setting. Finally, we report promising results on both simulated and real data.

4.2 Problem and notations

For any vector $w \in \mathbb{R}^p$, $\|w\|$ denotes the Euclidean norm of w, and $\operatorname{supp}(w) \subset [1,p]$ denotes the support of w, *i.e.*, the set of covariates $i \in [1,p]$ such that $w_i \neq 0$. A group of covariates is a subset $g \subset [1,p]$. The set of all possible groups is therefore $\mathcal{P}([1,p])$, the power set of $[1,p]$. Throughout the chapter, $\mathcal{G} \subset \mathcal{P}([1,p])$ denotes a set of groups, usually fixed in advance for each application. We say that two groups overlap if they have at least one covariate in common. For any vector $w \in \mathbb{R}^p$, and any group $g \in \mathcal{G}$, we denote $w_g \in \mathbb{R}^p$ the vector whose entries are the same as w for the covariates in g, and are 0 for other other covariates. However, we use a different convention for elements of $\mathcal{V}_{\mathcal{G}} \subset \mathbb{R}^{p \times \mathcal{G}}$ the set of $|\mathcal{G}|$-tuples of vectors $\mathbf{v} = (v_g)_{g \in \mathcal{G}}$, where each v_g is this time a separate vector in $\mathbb{R}^p$, which satisfies $\operatorname{supp}(v_g) \subset g$ for each $g \in \mathcal{G}$. For any differentiable function $f : \mathbb{R}^p \to \mathbb{R}$, we denote by $\nabla f(w) \in \mathbb{R}^p$ the gradient of f at $w \in \mathbb{R}^p$ and by $\nabla_g f(w) \in \mathbb{R}^g$ the partial gradient of f with respect to to the covariates in g.

4.3 Group lasso with overlapping groups

When the groups in $\mathcal{G}$ do not overlap, the group lasso penalty Yuan and Lin (2006) is defined as:

$$\forall w \in \mathbb{R}^p , \quad \Omega^{\mathcal{G}}_{\text{group}}(w) = \sum_{g \in \mathcal{G}} \|w_g\| \ . \tag{4.1}$$

When the groups in $\mathcal{G}$ form a partition of the set of covariates, then $\Omega^{\mathcal{G}}_{\text{group}}(w)$ is a norm whose balls have singularities when some w_g are equal to zero. Minimizing a smooth convex loss functional L over such a ball :

$$\min_w L(w) + \lambda \sum_{g \in \mathcal{G}} \|w_g\|_2 \tag{4.2}$$

often leads to a solution that lies on a singularity, *i.e.*, to a vector w such that $w_g = 0$ for some of the g in $\mathcal{G}$. The hyperparameter $\lambda \geq 0$ in (4.2) is used to adjust the tradeoff between minimizing the risk and finding a solution which is very sparse at the group level.

When some of the groups in $\mathcal{G}$ overlap, the penalty (4.1) is still a norm (if all covariates

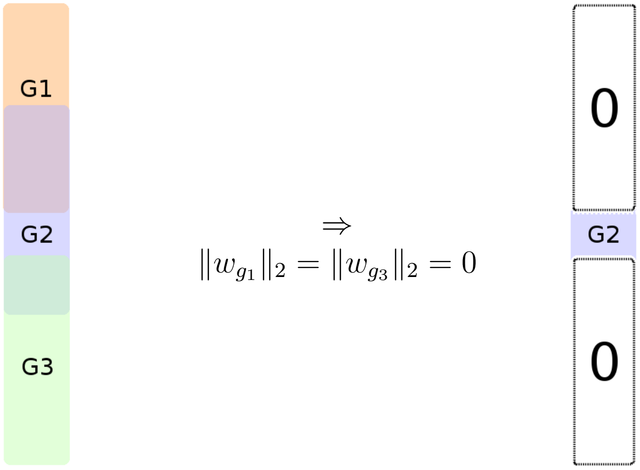

Figure 4.1: Effect of penalty (4.1) on the support. Removing *any* group containing a variable removes the variable from the support.

are in at least one group) whose ball has singularities when some w_g are equal to zero. Indeed, for a vector w, if we denote by $\mathcal{G}_0 \subset \mathcal{G}$ the set of groups such that $w_g = 0$, then

$$\text{supp}\,(w) \subset \Big(\bigcup_{g \in \mathcal{G}_0} g \Big)^c .$$

This is illustrated on a simple example in Figure 4.1. We see that this penalty induces the estimation of sparse vectors, whose support in typically the complement of a union of groups. Although this may be relevant for some applications, with appropriately designed families of groups — as considered by Jenatton et al. (2009) — , we are interested in this chapter in penalties which induce the opposite effect: that the support of w be a union of groups. For that purpose, we introduce one latent variable v_g by group and propose instead to solve the following problem :

$$\begin{cases} \min\limits_{w,v} L(w) + \lambda \sum\limits_{g \in \mathcal{G}} \|v_g\|_2 \\ w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(v_g) \subseteq g. \end{cases} \tag{4.3}$$

Figure 4.2 illustrates the idea of (4.3). Each group $g \in \mathcal{G}$ is assigned a latent variable

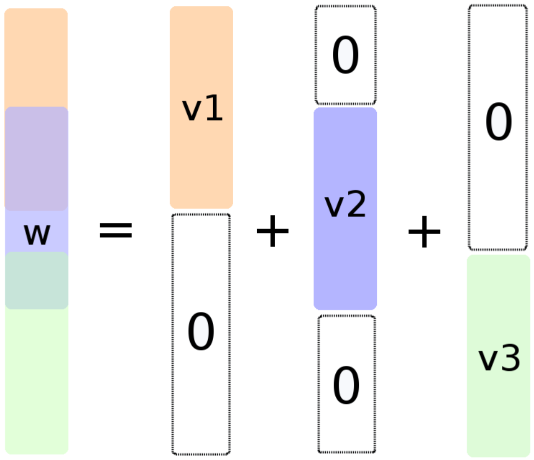

Figure 4.2: Latent decomposition of w over $(v_g)_{v \in \mathcal{G}}$. Applying the ℓ_1/ℓ_2 penalty to the decomposition instead of applying it to the w_g removes only the variables which do not belong to *any* selected group.

$v_g \in \mathbb{R}^p$ whose support is restricted to the group by the last constraint. Applying the ℓ_1/ℓ_2 penalty to these v_g favors solutions which have several $\|v_g\| = 0$. On the other hand, since we enforce w to be the *sum* of these v_g, a variable can be non-zero as long as it belongs to at least one selected group. More precisely, if we denote by $\mathcal{G}_1 \subset \mathcal{G}$ the set of groups g with $v_g \neq 0$, then we immediately get $w = \sum_{g \in \mathcal{G}_1} v_g$, and therefore:

$$\text{supp}\,(w) \subset \bigcup_{g \in \mathcal{G}_1} g\,.$$

In other words, this formulation leads to sparse solutions whose support is typically a union of groups, matching the setting of applications that motivate this work.

Interestingly, solving this expanded problem can be thought of as a minimization of L constrained by a particular penalty function. This can be seen directly by separating the $\min$ over v from the rest in (4.3) :

$$\begin{cases} \min\limits_{w,v} L(w) + \lambda \sum\limits_{g \in \mathcal{G}} \|v_g\|_2 \\ w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(v_g) \subseteq g, \end{cases} \quad = \quad \min_w L(w) + \lambda \Omega^{\mathcal{G}}_{\text{overlap}}\,(w)\,, \tag{4.4}$$

with

$$\Omega^{\mathcal{G}}_{\text{overlap}}(w) = \min_{\mathbf{v}\in\mathcal{V}_{\mathcal{G}}, \sum_{g\in\mathcal{G}} v_g = w} \sum_{g\in\mathcal{G}} \|v_g\| \,. \tag{4.5}$$

$\Omega^{\mathcal{G}}_{\text{overlap}}(w)$ is a penalty function of $w \in \mathbb{R}^p$ which is defined as the solution of a constrained minimization problem. When used instead of the ℓ_1/ℓ_2 penalty (4.1) to constrain the solution of a learning problem, it leads to solution whose support is included in a union of groups. When the groups do not overlap and form a partition of $[1, p]$, there exists a unique decomposition of $w \in \mathbb{R}^p$ as $w = \sum_{g\in\mathcal{G}} v_g$ with $\text{supp}(v_g) \subset g$, namely, $v_g = w_g$ for all $g \in \mathcal{G}$. In that case, both penalties (4.1) and (4.5) are the same. If some groups overlap, then we have shown that this penalty induces the selection of w that can be decomposed as $w = \sum_{g\in\mathcal{G}} v_g$ where some v_g are equal to 0.

Figure 4.3 shows the ball for both norms in $\mathbb{R}^3$ with groups $\mathcal{G} = \{\{1, 2\}, \{2, 3\}\}$. The pillow shaped ball of $\Omega^{\mathcal{G}}_{\text{group}}(\cdot)$ has four singularities corresponding to cases where either only w_1 or only w_3 is non-zero. By contrast, $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$ has two circular sets of singularities corresponding to cases where (w_1, w_2) only or (w_2, w_3) only is non zero.

In the remaining of this chapter, we therefore investigate in more details $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$, both theoretically and empirically.

4.4 Some properties of $\Omega^{\mathcal{G}}_{\textbf{overlap}}(.)$

We first analyze the decomposition of a vector $w \in \mathbb{R}^p$ as $\sum_{g\in\mathcal{G}} v_g$ induced by (4.5). For that purpose, let $\mathbf{V}(w) \subset \mathcal{V}_{\mathcal{G}}$ be the set of $|\mathcal{G}|$-tuples of vectors $\mathbf{v} = (v_g)_{g\in\mathcal{G}}$ which reach the minimum in (4.5), *i.e.*, which satisfy

$$w = \sum_{g\in\mathcal{G}} v_g \quad \text{and} \quad \Omega^{\mathcal{G}}_{\text{overlap}}(w) = \sum_{g\in\mathcal{G}} \|v_g\| \,.$$

The optimization problem (4.5) defining $\Omega^{\mathcal{G}}_{\text{overlap}}(w)$ is a convex problem and its objective is coercive, so that the set of solutions $\mathbf{V}(w)$ is non-empty and convex. Moreover,

Lemma 1. *$w \mapsto \Omega^{\mathcal{G}}_{overlap}(w)$ is a norm.*

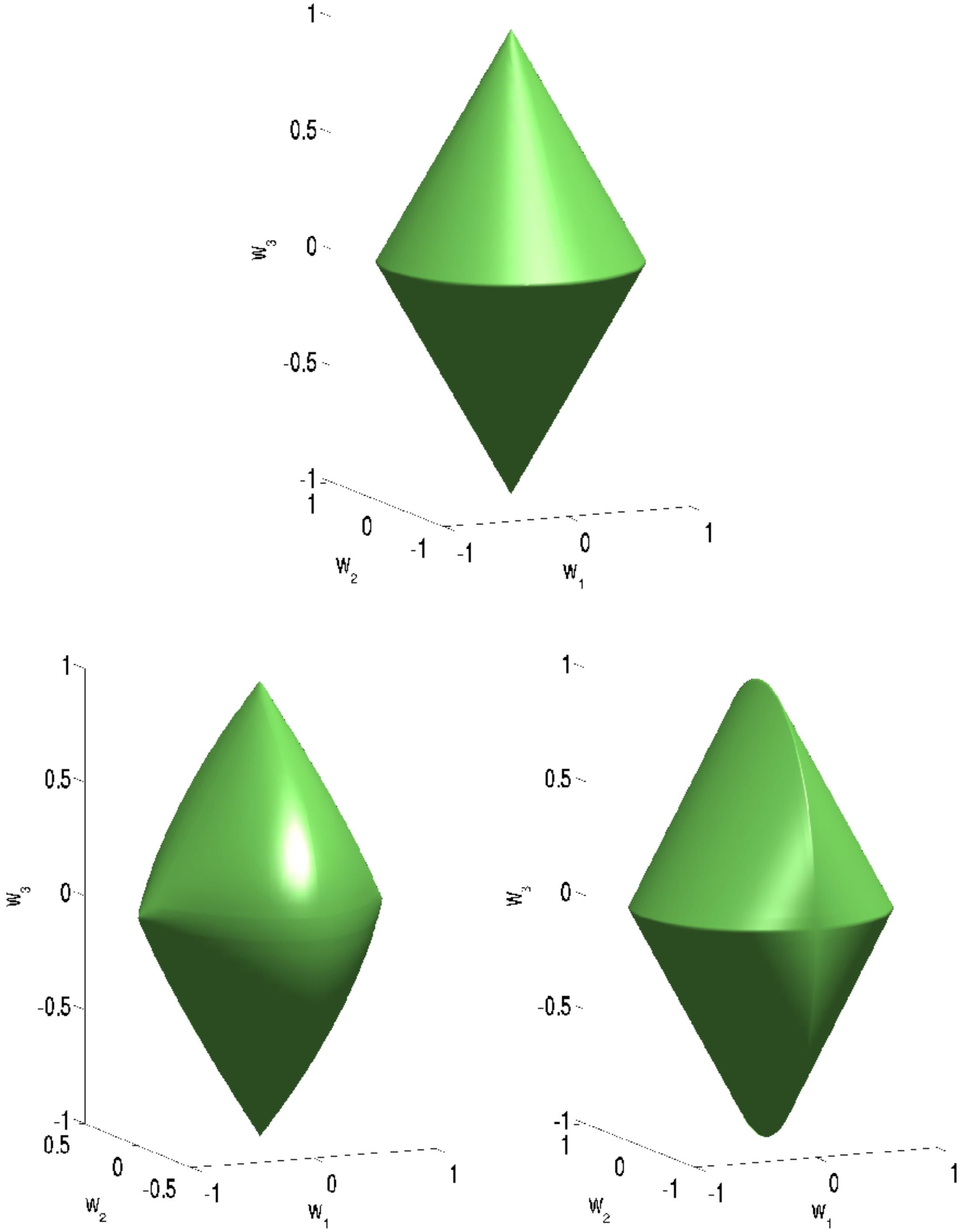

Figure 4.3: Bottom: balls for $\Omega^{\mathcal{G}}_{\text{group}}(\cdot)$ (left) and $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$ (right) for the groups $\mathcal{G} = \{\{1,2\},\{2,3\}\}$ where w_2 is represented as the vertical coordinate. Up: group-lasso ($\mathcal{G} = \{\{1,2\},\{3\}\}$), for comparison.

Proof. Positive homogeneity and positive definiteness hold trivially. We show the triangular inequality. Consider $w, w' \in \mathbb{R}^p$; let $(v_g)_{g\in\mathcal{G}}$ and $(v'_g)_{g\in\mathcal{G}}$ be respectively optimal decompositions of w and w' so that $\Omega^{\mathcal{G}}_{\text{overlap}}(w) = \sum_g \|v_g\|$ and $\Omega^{\mathcal{G}}_{\text{overlap}}(w') = \sum_g \|v'_g\|$. Since $(v_g + v'_g)_{g\in\mathcal{G}}$ is a (a priori non-optimal) decomposition of $w + w'$, we clearly have :

$$\Omega^{\mathcal{G}}_{\text{overlap}}(w + w') \leq \sum_{g\in\mathcal{G}} \|v_g + v'_g\| \leq \sum_g (\|v_g\| + \|v'_g\|) = \Omega^{\mathcal{G}}_{\text{overlap}}(w) + \Omega^{\mathcal{G}}_{\text{overlap}}(w') \,.$$

□

Using the conic dual of (4.5), we give another formulation of the norm $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ yelding some important properties.

Lemma 2. *1. It holds that:*

$$\Omega^{\mathcal{G}}_{\textit{overlap}}(w) = \sup_{\alpha\in\mathbb{R}^p:\forall g\in\mathcal{G}, \|\alpha_g\|\leq 1} \alpha^\top w \,. \tag{4.6}$$

2. *A vector $\alpha \in \mathbb{R}^p$ is a solution of* (4.6) *if and only if there exists $\mathbf{v} = (v_g)_{g\in\mathcal{G}} \in \mathbf{V}(w)$ such that:*

$$\forall g \in \mathcal{G}\,,\ \textit{if}\, v_g \neq 0,\ \alpha_g = \frac{v_g}{\|v_g\|}\ \textit{else}\ \|\alpha_g\| \leq 1 \tag{4.7}$$

3. *Conversely, a $\mathcal{G}$-tuple of vectors $\mathbf{v} = (v_g)_{g\in\mathcal{G}} \in \mathcal{V}_{\mathcal{G}}$ such that $w = \sum_g v_g$ is a solution to* (4.5) *if and only if there exists a vector $\alpha \in \mathbb{R}^p$ such that* (4.7) *holds.*

Proof. Let us introduce slack variables $\mathbf{t} = (t_g)_{g\in\mathcal{G}} \in \mathbb{R}^{\mathcal{G}}$ and rewrite the optimization problem (4.5) as follows:

$$\min_{\mathbf{t}\in\mathbb{R}^{\mathcal{G}}, \mathbf{v}\in\mathcal{V}_{\mathcal{G}}} \sum_{g\in\mathcal{G}} t_g \text{ s.t. } \sum_{g\in\mathcal{G}} v_g = w \text{ and } \forall g \in \mathcal{G}, \|v_g\| \leq t_g \,.$$

We can form a Lagrangian (Boyd and Vandenberghe, 2004) for this problem with the dual variables $\alpha \in \mathbb{R}^p$ for the constraint $\sum_{g\in\mathcal{G}} v_g = w$, and $(\beta, \gamma) \in \mathcal{V}_{\mathcal{G}} \times \mathbb{R}^{\mathcal{G}}$ with $\|\beta_g\| \leq \gamma_g$ for the conic constraints $\|v_g\| \leq t_g$, and get:

$$L = \sum_{g\in\mathcal{G}} t_g + \alpha^\top \Big(w - \sum_{g\in\mathcal{G}} v_g\Big) - \sum_{g\in\mathcal{G}} \left(\beta_g^\top v_g + \gamma_g t_g\right) .$$

The minimum of L with respect to the primal variables $\mathbf{t}$ and $\mathbf{v}$ is non trivial only if $\gamma_g = 1$ and $\alpha_g = -\beta_g$ for any $g \in \mathcal{G}$. Therefore, we get the dual function:

$$\min_{\mathbf{t},\mathbf{v}} L = \begin{cases} \alpha^\top w & \text{if } \gamma_g = 1 \text{ and } \alpha_g = -\beta_g \text{ for all } g \in \mathcal{G}\,, \\ -\infty & \text{otherwise.} \end{cases}$$

By strong duality (since, *e.g.*, Slater's condition is fulfilled), the optimal value $\Omega^{\mathcal{G}}_{\text{overlap}}(w)$ of the primal is equal to the maximum of the dual problem. Maximizing this dual function over $\gamma_g = 1$, $\|\beta_g\| \leq \gamma_g$ and $\alpha_g = -\beta_g$ is equivalent to maximizing $\alpha^\top w$ over the vectors $\alpha \in \mathbb{R}^p$ such that $\|\alpha_g\| \leq 1$ for all $g \in \mathcal{G}$, which proves (4.6). To prove the second point, we note that the variables $(\mathbf{t}, \mathbf{v}, \alpha, \beta, \gamma)$ are primal/dual optimal for this convex optimization problem if and only if the Karush-Kuhn-Tucker (KKT) conditions are satisfied, *i.e.*, if and only if, for all $g \in \mathcal{G}$:

$$\begin{cases} \text{supp}\,(v_g) = g, \|v_g\| \leq t_g \quad \text{and} \quad w = \sum_{g \in \mathcal{G}} v_g \\ \text{supp}\,(\beta_g) = g, \|\beta_g\| \leq \gamma_g \\ \alpha_g = -\beta_g \text{ and } \gamma_g = 1 \\ \beta_g^\top v_g + \gamma_g t_g = 0 \end{cases}$$

Eliminating β and γ with the stationarity conditions, all conditions are fulfilled if and only if $w = \sum_{g \in \mathcal{G}} v_g$ and for all $g \in \mathcal{G}$, (i) either $v_g = 0$ and $\|\alpha_g\| \leq 1$, (ii) or $v_g \neq 0$ and $\alpha_g = v_g/\|v_g\|$. If a pair $(\alpha, \mathbf{v})$ fulfills these conditions, then we obtain a primal/dual solution by taking $t_g = \|v_g\|$, $\beta_g = -\alpha_g$ and $\gamma_g = 1$. This proves points 2 and 3. □

Denote by $\mathcal{G}_1$ the group-support of w, i.e., the set of groups belonging to the support of at least one optimal decomposition of w: $\mathcal{G}_1 = \{g \in \mathcal{G} \mid \exists \mathbf{v} = (v_g)_g \in \mathbf{V}(w),\ v_g \neq 0\}$ and J_1 the corresponding set of variables $J_1 = \cup_{g \in \mathcal{G}_1} g$.

Lemma 3. *Let α be an optimum in the formulation* (4.6) *of the $\Omega^{\mathcal{G}}_{overlap}(\cdot)$ norm, then α_{J_1} is uniquely defined.*

Proof. Consider any solution $\mathbf{v} = (v_g)_{g \in \mathcal{G}}$ of (4.5). Let α be any optimal solution of (4.6). Since $(\mathbf{v}, \alpha)$ form a primal/dual pair, they must satisfy the KKT conditions. In particular, for all g such that $v_g \neq 0$, α_g is defined uniquely by $\alpha_g = \frac{v_g}{\|v_g\|}$. Since this is true for all solutions $\mathbf{v} \in \mathbf{V}(w)$, α_{J_1} is uniquely defined. □

Corollary 1. *For any* $\mathbf{v}, \mathbf{v}' \in \mathbf{V}(w)$ *and for any* $g \in \mathcal{G}$,

$$\|v_g\| \times \|v'_g\| = 0 \quad \textit{or} \quad \exists \gamma_g \geq 0 \textit{ s.t. } v'_g = \gamma v_g \,. \tag{4.8}$$

Proof. If $v_g \neq 0$ and $v'_g \neq 0$, let α be solution of (4.6), by the previous lemma α_g is unique and $\alpha_g = \frac{v_g}{\|v_g\|} = \frac{v'_g}{\|v'_g\|}$. □

4.5 Using $\Omega^{\mathcal{G}}_{\textbf{overlap}}(.)$ as a penalty

We now consider a learning scenario where we use $\Omega^{\mathcal{G}}_{\text{overlap}}(w)$ as a regularization term to the minimization of an objective function $L(w)$, typically an empirical risk. We assume that $L(w)$ is convex and differentiable in w, and consider the optimization problem:

$$\min_{w \in \mathbb{R}^p} L(w) + \lambda \Omega^{\mathcal{G}}_{\text{overlap}}(w) \ , \tag{4.9}$$

where $\lambda > 0$ is a regularization parameter. We first derive optimality conditions for any solution of (4.9). For that purpose, let us denote $\mathcal{A}_{\mathcal{G}}(w)$ the set of vectors $\alpha \in \mathbb{R}^p$ solution of (4.6).

Lemma 4. *A vector* $w \in \mathbb{R}^p$ *is a solution of* (4.9) *if and only if* $-\nabla L(w)/\lambda \in \mathcal{A}_{\mathcal{G}}(w)$.

Proof. The proof follows from the same Lagrangian based derivation as for Lemma 2, adding only the loss term. □

Remark 1. *By point* 2 *of Lemma 2, an equivalent formulation is the following: a vector* $w \in \mathbb{R}^p$ *is a solution of* (4.9) *if and only if it can be decomposed as* $w = \sum_{g \in \mathcal{G}} v_g$ *where, for any* $g \in \mathcal{G}$, $v_g \in \mathbb{R}^p$, *supp* $(v_g) = g$, *and if* $v_g = 0$ *then* $\|\nabla_g L(w)\| \leq \lambda$, *and* $\nabla_g L(w) = -\lambda v_g / \|v_g\|$ *otherwise.*

4.6 Consistency

Before we present a consistency result on $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$, we will need the following lemma.

Lemma 5. *Assume that for all w' in a small neighborhood U of w, w' admits a unique decomposition $(v'_g)_{g\in\mathcal{G}}$ of minimal norm supported by the same set of groups $\mathcal{G}_1$ as w. Writing $\eta_g = \|v_g\|$, there exists a neighborhood U_0 of w_{J_1} in $\mathbb{R}^{|J_1|}$ and a neighborhood U'_0 of $(\alpha_{J_1}, \eta_{\mathcal{G}_1})$ in $\mathbb{R}^{|J_1|\times|\mathcal{G}_1|}$ such that there exists a unique continuous function*

$$\phi : w_{J_1} \mapsto (\alpha_{J_1}(w), \eta_{\mathcal{G}_1}(w))$$

from U_0 to U'_0.

Proof. The dual problem (4.6) is equivalent to the saddle-point problem

$$\min_{\alpha} \max_{\eta} L'(\alpha, \eta, w) \text{ s.t. } \eta_g \in \mathbb{R}_+,$$

with Lagrangian

$$L'(\alpha, \eta, w) = -\alpha^\top w + \sum_{g\in\mathcal{G}} \frac{\eta_g}{2}(\|\alpha_g\|^2 - 1)$$

and KKT conditions:

$$\begin{cases} \forall g \in \mathcal{G}, \|\alpha_g\|^2 \leq 1, & \text{(primal feas.)} \\ \forall g \in \mathcal{G}, \eta_g \geq 0, & \text{(dual feas.)} \\ \forall i \in [1,p], -w_i + \left(\sum_{g\ni i} \eta_g\right) \alpha_i = 0, & \text{(stationarity)} \\ \forall g \in \mathcal{G}, \eta_g(\|\alpha_g\|^2 - 1) = 0, & \text{(comp.slack.)} \end{cases}$$

By stationarity, $(v_g)_{g\in\mathcal{G}}$ defined by $v_g = \eta_g \alpha_g$ is a decomposition of w; it is optimal because it satisfies property 3 of lemma 2; finally we have $\eta_g = \|v_g\|$ consistently with our definition of $\eta_g(w)$. For any w with the same set of supporting groups $\mathcal{G}_1$, we have $\|\alpha_g(w)\| = 1$ for all $g \in \mathcal{G}_1$ and $\eta_g = 0$ for all $g \in \mathcal{G}\backslash\mathcal{G}_1$. For all w_{J_1} with group-support no smaller than $\mathcal{G}_1$, the corresponding pair $(\alpha_{J_1}(w), \eta_{\mathcal{G}_1}(w))$ is therefore a solution of the set of non-linear equations:

$$\begin{cases} \forall i \in J_1, -w_i + \left(\sum_{g\ni i} \eta_g\right) \alpha_i = 0 \\ \forall g \in \mathcal{G}_1, \|\alpha_g\|^2 - 1 = 0 \end{cases} \tag{4.10}$$

In other words consider the function

$$\begin{aligned} F : \mathbb{R}^{|J_1| \times |J_1| \times |\mathcal{G}_1|} &\longrightarrow \mathbb{R}^{|J_1| \times |\mathcal{G}_1|} \\ (w_{J_1}, \alpha_{J_1}, \eta_{\mathcal{G}_1}) &\mapsto \begin{pmatrix} \left(-w_i + \left[\sum_{g \ni i} \eta_g\right] \alpha_i\right)_{i \in J_1} \\ (\|\alpha_g\|^2 - 1)_{g \in \mathcal{G}_1} \end{pmatrix}, \end{aligned}$$

then (4.10) is equivalent to $F(w_{J_1}, \alpha_{J_1}, \eta_{\mathcal{G}_1}) = 0$. We use the implicit function theorem for non-differentiable function of Kumagai (1980). The theorem states that for a continuous function

$$F : \mathbb{R}^{|J_1|} \times \mathbb{R}^{|J_1| \times |\mathcal{G}_1|} \to \mathbb{R}^{|J_1| \times |\mathcal{G}_1|},$$

such that $F(w_0, (\alpha_0, \eta_0)) = 0$, if there exist open neighborhoods $U \subset R^{|J_1|}$ and $U' \subset R^{|J_1| \times |\mathcal{G}_1|}$ of w_0 and (α_0, η_0) respectively, such that, for all $w \in U$, $F(w, \cdot) : U' \to R^{|J_1| \times |\mathcal{G}_1|}$ is locally one-to-one then there exist open neighborhoods $U_0 \subset R^{|J_1|}$ and $U'_0 \subset R^{|J_1| \times |\mathcal{G}_1|}$ of w_0 and (α_0, η_0), such that, for all $w \in U_0$, the equation $F(w, (\alpha, \eta)) = 0$ has a unique solution $(\alpha, \eta) = \phi(w) \in U'_0$, where ϕ is a continuous function from U_0 into U'_0. By continuity of the addition, the product and the Euclidean norm, the above defined F is continuous. For each w fixed, $F(w, \cdot)$ is bijective, because of the assumption of the existence of a unique decomposition in a neighborhood of w. Applying the theorem of Kumagai (1980) then yields the desired result.

□

We are now ready to prove the consistency of $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$. Consider the linear regression model $Y = X\bar{w} + \epsilon$, where $X \in \mathbb{R}^{n \times p}$ is a design matrix, $Y \in \mathbb{R}^p$ is the response vector and $\epsilon \in \mathbb{R}^p$ is a vector of i.i.d. random variables with mean 0 and finite variance. We denote the true regression function by $\bar{w}$. We assume that

1. (H1) $\quad \Sigma := \frac{1}{n} X^\top X \succ 0$
2. (H2) There exists a neighborhood of $\bar{w}$ in which (4.5) has a unique solution.

If $\mathcal{G}_1$ is the set of group supporting the unique solution of (4.5), we denote $\mathcal{G}_2 \triangleq \mathcal{G} \backslash \mathcal{G}_1$ and $J_2 \triangleq [1, p] \backslash J_1$. For convenience, for any group of covariates g we note X_g the $n \times |g|$ design matrix restricted to the predictors in g, and for any two groups g, g' we note $\Sigma_{gg'} =$

$X_g^\top X_{g'}$. We can then provide a condition under which minimizing the least-square error penalized by $\Omega^{\mathcal{G}}_{\text{overlap}}(w)$ leads to an estimator with the correct support. Consider the two conditions:

$$\forall g \in \mathcal{G}_2,\ \|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(\bar{w})\| \leq 1 \tag{C1}$$

$$\forall g \in \mathcal{G}_2,\ \|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(\bar{w})\| < 1 \tag{C2}$$

Lemma 6. *With assumptions (H1-2), for $\lambda_n \to 0$ and $\lambda_n n^{1/2} \to \infty$, conditions* (C1) *and* (C2) *are respectively necessary and sufficient for the solution of* (4.9) *to estimate consistently the group-support of $\bar{w}$.*

Proof. We follow the line of proof of Bach (2008a) but consider a fixed design for simplicity of notations. Let us first consider the subproblem of estimating a vector only on the support of $\bar{w}$ by using only the groups in J_1 in the penalty, *i.e.*, consider $w_1 \in \mathbb{R}^{J_1}$ a solution of $\min_{w_{J_1}\in\mathbb{R}^{J_1}} \frac{1}{2n}\|Y - X_{J_1}w_{J_1}\|^2 + \lambda_n\Omega^{\mathcal{G}_1}_{\text{overlap}}(w_{J_1})$. By standard arguments, we can prove that w_1 converges in Euclidean norm to $\bar{w}$ restricted to J_1 as n tends to infinity Knight and Fu (2000). In the rest of the proof we show how to construct a vector $w \in \mathbb{R}^p$ from w_1 which under condition (C2) is with high probability a solution to (4.9). By adding null components to w_1, we obtain a vector $w \in \mathbb{R}^p$ whose support is also J_1, and $u = w - \bar{w}$ therefore satisfies $\text{supp}(u) \subset J_1$. A direct computation of the gradient of the loss $L(w) = \|Y - Xw\|^2$ gives $\nabla L(w) = \Sigma u - W$, where $W = \frac{1}{n}X\epsilon$. From this we deduce that $u = \Sigma_{J_1J_1}^{-1}(\nabla_{J_1}L(w) + W_{J_1})$, and since $\nabla_{J_1}L(w) = -\lambda_n\alpha_{J_1}(w)$ we have :

$$\nabla_{J_2}L(w) = \Sigma_{J_2J_1}\Sigma_{J_1J_1}^{-1}(W_{J_1} - \lambda_n\alpha_{J_1}(w)) - W_{J_2}\,.$$

To show that w is a feasible solution to (4.9) it is enough to show that $\forall g \in \mathcal{G}_2, \|\nabla_g L(w)\| \leq \lambda_n$. Moreover, since the noise has bounded variance,

$$\Sigma_{J_2J_1}\Sigma_{J_1J_1}^{-1}W_{J_1} - W_{J_2} = X_{J_2}^\top\left[\frac{1}{n}X_{J_1}\Sigma_{J_1J_1}^{-1}X_{J_1}^\top - I\right]\epsilon$$

is $\sqrt{n}$-consistent and

$$\frac{1}{\lambda_n}\|\nabla_g L(w)\| \leq \|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(w)\| + \mathcal{O}_p(\lambda_n^{-1}n^{-1/2}).$$

By Lemma 5, we have that α_{J_1} is a continuous function of w in a neighborhood of $\bar{w}$ so that $w_{J_1} \xrightarrow{\mathbb{P}} \bar{w}_{J_1}$ implies $\alpha_{J_1}(w) \xrightarrow{\mathbb{P}} \alpha_{J_1}(\bar{w})$. Since we chose λ_n such that $\lambda_n^{-1} n^{-1/2} \to 0$, we have

$$\frac{1}{\lambda_n}\|\nabla_g L(w)\| \leq \|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(\bar{w})\| + o_p(1).$$

Hence the result for the sufficient condition. Symmetrically, for the necessary condition we have

$$\frac{1}{\lambda_n}\|\nabla_g L(w)\| \geq \|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(\bar{w})\| - o_p(1).$$

□

4.7 Graph lasso

We now consider the situation where we have a simple undirected graph (I, E), where the set of vertices $I = [1, k]$ is the set of covariates and $E \subset I \times I$ is a set of edges that connect covariates. We suppose that we wish to estimate a sparse model such that selected covariates tend to be connected to each other, *i.e.*, form a limited number of connected components on the graph. An obvious approach is to consider the prior $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ where $\mathcal{G}$ is a set that generates by union the connected components. For example, we may consider for $\mathcal{G}$ the set of edges, cliques, or small linear subgraphs. As an example, considering all edges, *i.e.*, $\mathcal{G} = E$ leads to :

$$\Omega_{\text{graph}}(w) = \min_{v \in \mathcal{V}_E} \sum_{e \in E} \|v_e\| \quad \text{s.t.} \sum_{e \in E} v_e = w,\ \text{supp}\,(v_e) = e\,.$$

Alternatively, we will consider in the experiments the set of all linear subgraphs of length $k \geq 1$. Although we have no formal statement on how to chose k, it intuitively controls the size of the groups of connected variables which are selected, and should therefore be typically chosen to be slightly smaller than the size of the minimal connected component expected in the support of the model.

4.8 Implementation

We consider loss functions L which only depend on w through dot products with the data points X_i, *i.e.* $L(w) = \tilde{L}(Xw)$, which is the case of many loss functions of interest.

In this case, a simple way to implement empirical risk minimization using $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ as the regularizer is to explicitly duplicate the variables in the design matrix. Using the latent variable formulation of (4.4) and eliminating w by plugging the first constraint into $\tilde{L}(Xw)$, we can write :

$$\begin{cases} \min\limits_{w,v} \tilde{L}(Xw) + \lambda \sum\limits_g \|v_g\|_2 \\ w = \sum_g v_g \\ \text{supp}\,(v_g) \subseteq g. \end{cases} = \min_{\tilde{v}} \tilde{L}(\tilde{X}\tilde{v}) + \lambda \sum_g \|\tilde{v}_g\|_2,$$

where $\tilde{X} \in \mathbb{R}^{n \times \sum |g|}$ is defined by the concatenation of copies of the design matrix restricted each to a certain group g, i.e., $\tilde{X} = [X_{g_1}, X_{g_2}, ..., X_{g_{|\mathcal{G}|}}]$, with $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$, and where we denote $\tilde{v}_g = (v_{gi})_{i \in g}$ and $\tilde{\mathbf{v}} = (\tilde{v}^\top_{g_1}, \ldots, \tilde{v}^\top_{g_{|\mathcal{G}|}})^\top$.

On our simple example with 3 overlapping groups, this gives :

$$Xw = X.\begin{pmatrix}\tilde{v}_1\\0\end{pmatrix} + X.\begin{pmatrix}0\\\tilde{v}_2\\0\end{pmatrix} + X.\begin{pmatrix}0\\\tilde{v}_3\end{pmatrix} = (X_{g_1}, X_{g_2}, X_{g_3})\,.\begin{pmatrix}\tilde{v}_1\\\tilde{v}_2\\\tilde{v}_3\end{pmatrix} \triangleq \tilde{X}\tilde{v}.$$

That way the vector $\tilde{\mathbf{v}} \in \mathbb{R}^{\sum |g|}$ can be directly estimated from $\tilde{X}$ with a classical group lasso for non-overlapping groups. We implemented the approach of Meier et al. (2008) to estimate the group lasso in the expanded space. Note that Roth and Fischer (2008) provides a faster algorithm for the group Lasso. When there are many groups with important overlap however, an alternative implementation without explicit data duplication, *e.g.*, with a variational formulation similar to the one of Rakotomamonjy et al. (2008) might be more scalable.

4.9 Experiments

4.9.1 Synthetic data: given overlapping groups

To assess the performance of our method when overlapping groups are given as *a priori*, we simulated data with $p = 82$ variables, covered by 10 groups of 10 variables with 2 variables of overlap between two successive groups: $\{1, \dots, 10\}, \{9, \dots, 18\}, \dots, \{73, \dots, 82\}$. We chose the support of w to be the union of groups 4 and 5 and sampled both the support weights and the offset from i.i.d. Gaussian variables. Note that in this setting, the support can be expressed as a union of groups, but not as the complement of a union. Therefore, $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ can recover the right support, whereas by construction $\Omega^{\mathcal{G}}_{\text{group}}(\cdot)$ using the same groups would be unable to recover it.

The model is learned from n data points (x_i, y_i), with $y_i = w^\top x_i + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma = |\mathbb{E}(Xw + b)|$. Using an ℓ_2 loss $L(w) = \|Y - Xw - b\|^2$, we learn models from 50 such training sets. On Figure 4.4, for each variable (on the vertical axis), we plot its frequency of selection in levels of gray as a function of the regularization parameter λ, both for the Lasso penalty and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$.

For any choice of λ the Lasso frequently misses some variables from the support, while $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ never misses any variable from the support for a large part of the regularization path. Besides, we observed that over the replicates, the Lasso never selected the exact correct pattern for $n < 100$. For $n = 100$, the right pattern was selected with low frequency on a small part of the regularization path. $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ on the other hand selected it up to 92% of the times for $n = 50$ and more than 99% on more than one third of the path for $n = 100$. We tried the same experiment for various n and as long as n was too small for the Lasso to recover the right support, the group regularization always helped.

Figure 4.5 shows the root mean squared error of both methods for various n. For both methods, the full regularization path is computed and tested on three replicates of n training and 100 testing points. The best average parameter is selected and used to train and test a model on a fourth replicate. On a large range of n, $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$, not only helps to recover the

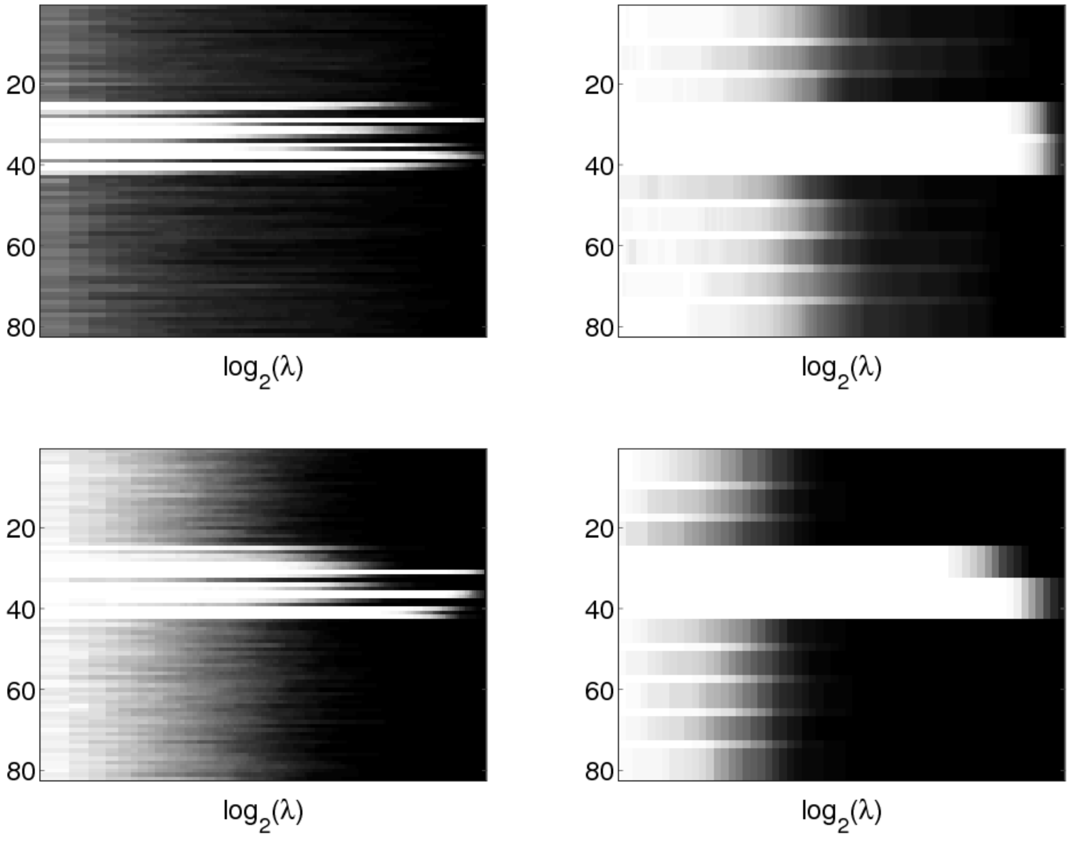

Figure 4.4: Frequency of selection of each variable with the Lasso (left) and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ (right) for $n = 50$ (top) and 100 (bottom).

right pattern, improves the regression performance. A possible explanation is that if several variables from the support are correlated in the design matrix X, the Lasso selects one and is less robust than $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ which uses all the variables. Note that when enough training points become available (last point on Figure 4.5), Figure 4.4 shows that the selected model is generally better but still not correct whereas $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ selects the right model, even if it does not give much lower error anymore.

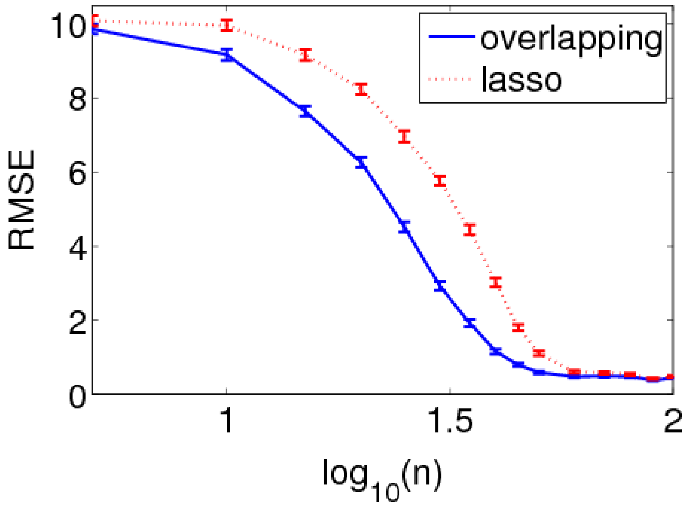

Figure 4.5: Root mean squared error of overlapped group lasso and Lasso as a function of the number of training points.

4.9.2 Synthetic data: given linear graph structure

We now consider that the prior given on the variables is a graph structure and that we are interested by solutions which are connected components on this graph. As a first simple illustration, we consider a chain. We use $w \in \mathbb{R}^p$, $p = 100$, $\operatorname{supp}(w) = [20, 40]$. The nodes of the graph are the variables w_i, the edges are all the pairs $(w_i, w_{i+1}), i = 1, \dots, n$. The model's weights, offset and the 50 training examples (x, y) are drawn using the same protocol as in the previous experiment. We take for the groups all the sub-chains of length k. We present the results for various choices of k and compare to the Lasso ($k = 1$).

Figure 4.6 shows the frequency of each variable selection over 20 replications. Here again, using a group prior helps the pattern recovery. We also observe as expected that the choice of k plays a role in the improvement.

4.9.3 Synthetic data: given non-linear graph structure

Here we consider the same setting as in the linear case, except that instead of a chain we are given a grid structure on the variables. Each node is connected to the 4 nodes above,

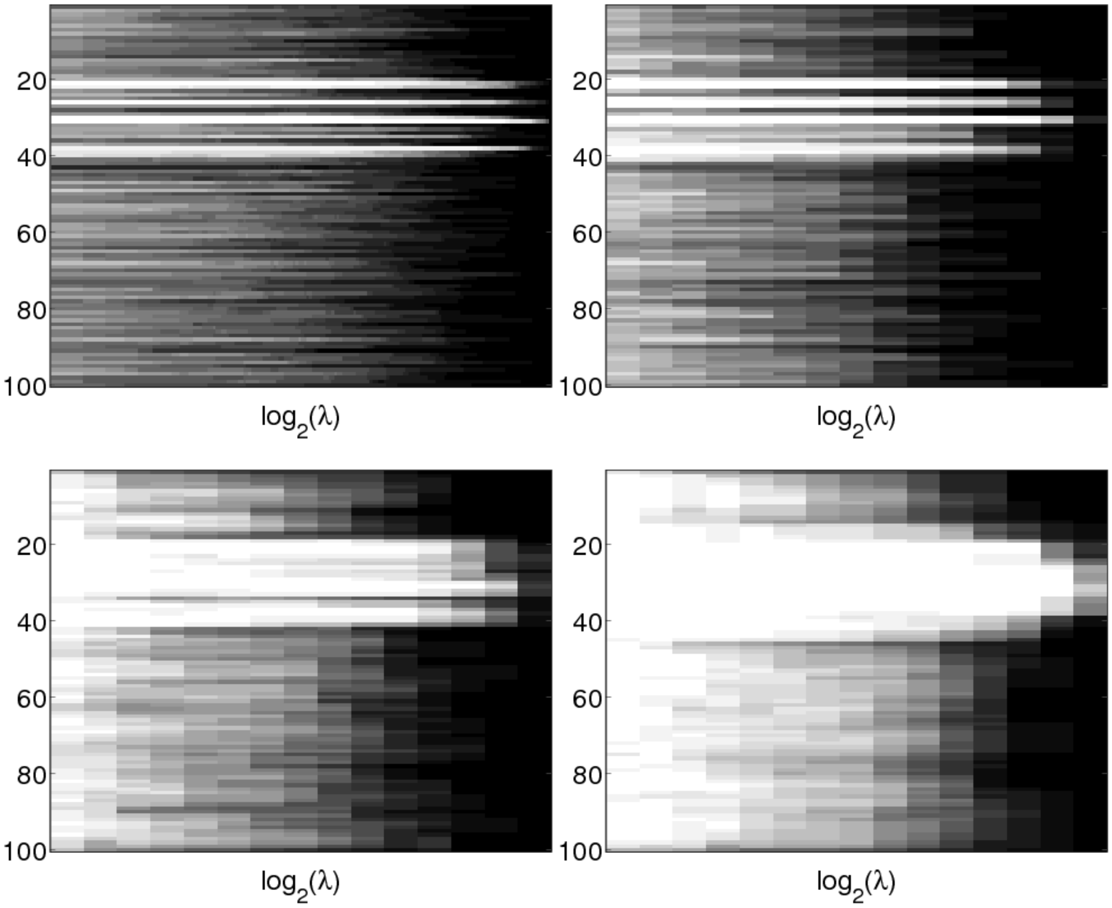

Figure 4.6: Variable selection frequency with $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ using the chains of length k (left) as groups, for $k = 1, 2, 4, 8$.

below, left and right. The support is a 20-variable region in the center of the grid, x-axis 4 to 7, y-axis 4 to 8. As groups, we use all the 4-cycles, which is a natural prior given the graph topology and the expected pattern.

Figure 4.7 shows the variable selection frequency of each variable for both methods at a fixed λ (chosen in both cases to give the best behavior). $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ seems to generally give better selection performances than Lasso.

Besides, we observed that on each run, variables incorrectly selected where always unions of groups whereas the Lasso selected disconnected variables on the graph. We made the same observation for the linear graph case. This is an expected property of our method, and implies that even if variables which are not in the model are selected, they enter the model as large connected components, whereas the false positive of the Lasso are more randomly distributed on the graph, often as isolated variables. This is an interesting property for real applications because it may then be easier to discard manually a few large connected components of false positives, than many isolated variables (assuming of course that the right variables are selected as well).

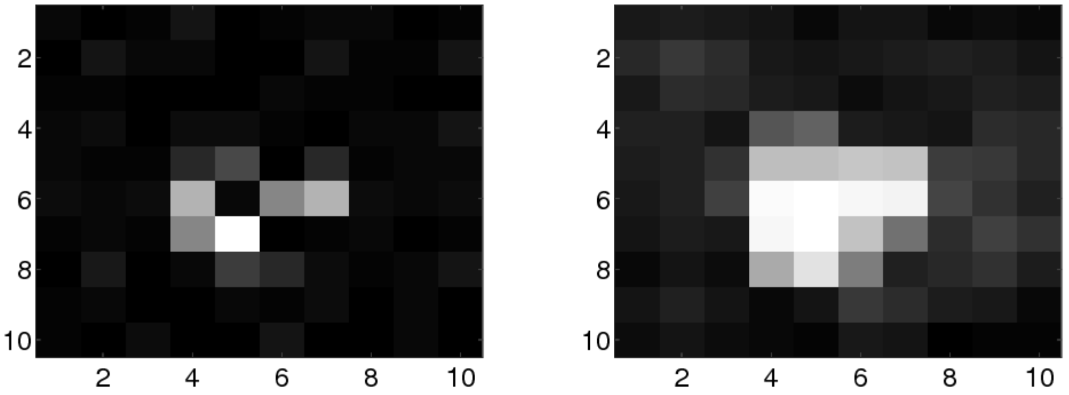

Figure 4.7: Grid view of the variable selection frequencies with the non-linear graph setting. Left: Lasso, right: $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ using 4-cycles as groups. $n = 30$ training points, λ is arbitrarily fixed.

4.9.4 Breast cancer data: pathway analysis

An important motivation for our method is the possibility to perform gene selection from microarray data using priors which are overlapping groups. For example, one may want to analyse microarrays in terms of biologically meaningful gene sets. In most such analysis, genes discriminating the classes (*e.g.* tumors leading to metastasis versus non-metastasis) are selected in a first step, then enrichment analysis is performed by looking for gene sets in which selected genes are overrepresented Subramanian et al. (2005). Several organizations of the genes into gene sets are available in various databases. We use the canonical pathways from MSigDB Subramanian et al. (2005) containing 639 groups of genes, 637 of which involve genes from our study.

Table 4.1: Classification error, number and proportion of pathways selected by the ℓ_1 and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ on the 3 folds.

METHOD	ℓ_1	$\Omega^{\mathcal{G}}_{\text{OVERLAP}}(.)$
ERROR	0.38 ± 0.04	0.36 ± 0.03
$\sharp$ PATH.	$148, 58, 183$	$6, 5, 78$
PROP. PATH.	$0.32, 0.14, 0.41$	$0.01, 0.01, 0.17$

We use the breast cancer dataset compiled by van de Vijver et al. (2002), which consists of gene expression data for $8,141$ genes in 295 breast cancer tumors (78 metastatic and 217 non-metastatic). We restrict the analysis to the 3510 genes which are in at least one pathway. Since the dataset is very unbalanced, we balance it by using 3 replicates of each metastasis patient (keeping all duplicates in the same fold during cross-validation).

We estimate by 3-fold cross validation the accuracy of a logistic regression with ℓ_1 and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ penalties, using the pathways as groups. As a pre-processing, we keep the 300 genes most correlated with the output (on each training set). λ is selected by cross validation on each training set.

Table 4.1 shows the results of both methods. Using $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ instead of the ℓ_1 penalty leads to a slight improvement in the prediction performances, and much sparser solutions at the pathway level, which makes the selected model easier to interpret.

4.9.5 Breast cancer data: graph analysis

Another important application in microarray data analysis is the search for potential drug targets. In order to identify genes which are related to a disease, one would like to find groups of genes forming connected components on a graph carrying biological information such as regulation, involvement in the same chain of metabolic reactions, or protein-protein interaction. Similarly to what is done in pathway analysis, Chuang et al. (2007) built a network by compiling several biological networks and performed such graph analysis by identifying discriminant subnetworks in one step and using these subnetworks to learn a classifier in a separate step. We use this network and the approach described in section 4.7, taking all the edges on the network as the groups, on the breast cancer dataset. Here again,

Table 4.2: Classification error and average size of the connected components selected by the ℓ_1 and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ on the 3 folds.

METHOD	ℓ_1	$\Omega^{\mathcal{G}}_{\text{OVERLAP}}(.)$
ERROR	0.39 ± 0.04	0.36 ± 0.01
AV. SIZE C.C.	$1.1, 1, 1.0$	$1.3, 1.4, 1.2$

we restrict the data to the 7910 genes which are present in the network, and use the same correlation-based pre-processing as for the pathway analysis.

Table 4.2 shows the results of the logistic regression with ℓ_1 and $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$. Here again, both methods give similar performances, with a slight advantage for $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$. On the other hand, while the ℓ_1 mostly selects disconnected variables on the graph, $\Omega^{\mathcal{G}}_{\text{overlap}}(.)$ tends to select variables which are grouped into larger connected components on the graph. This would make the interpretation and the search for new drug targets easier.

4.10 Discussion

We have presented a generalization of the group lasso penalty, which leads to sparse models with sparsity patterns that are unions of pre-defined groups of covariates, or, given a graph of covariates, groups of connected covariates in the graph. We obtained promising results on both simulated and real data.

From a theoretical point of view, we gave both sufficient and necessary conditions for the correct recovery of the same union of groups as in the decomposition induced by $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$ on the true optimal parameter vector. It still remains to characterize when the latter decomposition has the smallest number of groups. The situation where several decompositions exist should be analyzed. Also, the construction of an adaptive version of the Group Lasso with overlap that could possibly generalize the scheme proposed by Bach (2008a) would be of interest.

From a practical point of view, although algorithms for the standard group Lasso can be used to implement $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$, more dedicated and scalable algorithms could be designed for cases with large overlaps.

Future work should compare more systematically $\Omega^{\mathcal{G}}_{\text{overlap}}(\cdot)$ and $\Omega^{\mathcal{G}}_{\text{group}}(\cdot)$ empirically and theoretically.

Conclusion

The successive chapters of this thesis have illustrated how supervised learning methods could take advantage of the available prior knowledge in computational biology problems. Chapter 2 has showed how to practically use the fact that similar targets bind similar ligands in the context of vaccine design and drug discovery. This involved working with target-ligand pairs, which could technically be done directly by using product kernels. Experiments showed that this information sharing across the targets helped when little data was available, even allowing to learn when no data is available for a given target. They also showed however that this information sharing was pointless or even harmful when enough data was available. Chapter 3 has introduced a practical algorithm to deal with the cases where one only wants to share information across certain sets of tasks, where the sets are unknown beforehand. We provided an efficient algorithm to deal with the resulting optimization problem, and gave interpretations both in terms of non-Hilbertian norm of the classification functions, and of convex relaxation of the original k-means objective. Finally in Chapter 4, we introduced a penalty which induces solutions involving few groups among some predefined overlapping groups of covariates, or few connected components in a given graph on the covariates when used to regularize a convex smooth optimization problem. This allowed us to build functions predicting breast cancer outcome from gene expressions, and involving only few gene pathways, resulting in more interpretable signatures. Technically, we provided practical algorithms to solve the resulting optimization problem and showed some properties of this new estimator, in particular its consistency under certain conditions.

From a technical point of view, all the algorithms we introduced can be expressed under the usual Tikhonov regularization form of the minimization of a loss functional penalized by a regularization term. It is worth noting that in Chapter 3 and 4, these regularization terms are novel non-Hilbertian norms which where built to impose the desired type of regularization. More precisely, these norms are given by the solution of some convex optimization problems, which in turn define some quantity of interest for analysis purpose: a partition of the learning problems in the first case, and an optimal decomposition of the support of the linear function over some pre-defined overlapping groups in the second case.

For each of these problems, several improvement directions remain to be explored. For the interaction prediction problem, in the context of which we used Hilbertian norms in a joint description space, the most fragile element from our point of view is not really the method itself, but the design of the benchmark on which it is tested. The practical purpose of the ligand-based approach we presented is to mine large databases of drug (or vaccine) candidates. It is not easy to simulate this setting, because the existing databases contain a lot of redundant information which is not straightforward to filter automatically, in particular some receptors are extremely close and share the same binding repertoire in databases. Overall, it is very easy to build a benchmark which over-represents some pieces of the chemical or receptor spaces (namely, those which are best studied and published in public databases). Quantifying the prediction accuracy of a method on such a benchmark may give biased results. This doesn't invalidate our results at all, because our purpose was only to show that sharing information helps learning for targets with little or no training data, but a consequence of this remark is that the importance of the improvement brought by sharing information may be different in practical cases depending on the quantity and quality of training data available for similar enough targets. A related comment on these experiments concerns the way the practical problem is classicaly modeled in machine learning. Ligand-based approaches cast the problem of enriching a set of ligand candidates in a binary classification framework, *i.e.*, as the problem of separating the true binders from the non-binders. In practice, this implies to have both positive (binders) and negative (non-binders) training data whereas most databases contain only lists of ligands for some targets. Casting the problem as a binary classification therefore makes it necessary to find a way to designate some molecules as being non-ligands. It may be worth finding a better and systematic way to learn from positive data only. Concerning the method itself, Chapter 2 only illustrated the effect of considering the problem of predicting interactions as a unique joint learning problem. Several improvements could be brought by integrating recent advances in machine learning, such as using more recent kernels for targets or ligands. In particular, it seems crucial to find good ways to compare targets in terms of their binding abilities, which could necessitate more thorough methods to compare binding pockets. Learning the kernel or using more recent multi-task learning approaches, as suggested in Chapter 3, may also improve the performances on this problem.

Concerning the penalty we proposed in Chapter 3, which is based on a convex relaxation of the clustering problem, direct improvements could be brought by finding tighter relaxations, involving less parameters, and by faster optimization schemes. A limitation of this approach is indeed that it transforms T learning problems of dimension p, corresponding to the T tasks in a single joint problem of dimension pT and that, contrarily to what happens with the trace norm minimization, the problems do not decouple even when fixing the metric Σ. Other possible improvements include taking into account existing features on the tasks, which could guide the clustering process, and enforcing that antithetic tasks learn from each others, *i.e.*, that the problem is invariant by flipping the labels. Once these improvements are brought, it would be interesting to apply the method to chemogenomics data, and in particular to see if the resulting clustering corresponds to known receptor families or gives interesting insights in terms of target grouping.

Several directions emerge from the penalty we proposed in Chapter 4. First, this work overall makes it necessary to define what the optimal decomposition of the support of a classifier over a set of groups of covariates is when the groups overlaps. In particular, depending on the number of available training points, it may be preferable to include, *e.g.*, one large group containing the full support and few other variables than several small groups whose union is exactly the support, thereby introducing a small bias in the model selection at the benefit of a better stability. An appropriate weighting of the groups should allow to deal with this trade-off. In addition, since the optimal decomposition over the groups may not be unique even for some simple patterns, it would be important to generalize our consistency result to the such cases. Concerning the implementation, we used the straightforward approach to duplicate the variables which were in several groups and apply non-overlapping group-lasso because it was sufficient for our experiments, but several finer approaches could be devised, including any multiple kernel learning algorithm, which would have a better scaling in the size of the group intersections than the duplication approach. Using these algorithms would by definition also allow to introduce non-linearities among variables which belong to the same group. Allowing non-linear effects between variables which belong to different groups could be even more interesting, but is less straightforward. Finally, while using this new penalty improved the learning accuracy on synthetic data, it failed to do so on real datasets like the breast cancer benchmark we used in our experiments. Achieving sparsity at the gene set level, hence more interpretability, was one of the objectives, which was met in the experiments, but enforcing this biological prior did not lead to the expected improvement. This is not extremely surprising since this dataset is known to be a difficult prediction problem, on which basically all methods level up at the same performance, but it leaves unanswered the question of how to better predict cancer outcome from molecular data and whether or not this is compatible with enforcing interpretable priors like the one we used here. A last related point which remains to be verified, is whether using this penalty confers more robustness to the learned classifier, *i.e.*, whether it gives reasonable performances on a new independant breast cancer dataset.

Appendix A

Context

A.1 Relation between learning in a joint feature space and controlling the variance of the individual tasks

Let us choose $K_{\text{task}} = 1 + \mu\delta_{\{t=t'\}}$ for some $\mu(\alpha)$, where $\delta_{\{t=t'\}}$ is 1 if $t = t'$, 0 otherwise. Considering the linear kernel $K_{\text{data}}(x, x')$ to simplify the notations, one can check that

$$\begin{aligned} K_{\text{data}}(x, x')(1 + \mu\delta_{\{t=t'\}}) &= \langle x, x' \rangle (1 + \mu\delta_{\{t=t'\}}) \\ &= \langle \Phi(x), \Phi(x') \rangle, \end{aligned}$$

with $\Phi(x) = x \otimes (1 \quad \sqrt{\mu} e_{t(x)})$, $t(x)$ being the task of point x, $\otimes$ denoting the tensor product and e_t the t-th vector of the canonical basis. Therefore describing the pairs by this product of kernels is equivalent to describing each data-task pair (x, t) by a large vector of size $p(T + 1)$ containing only zeros except at the first p positions and between the $tp + 1$ and the $(t + 1)p$ positions, where it contains the description of the data. In this joint feature space, the prediction function is :

$$w^\top \Phi(x) = \underbrace{\begin{bmatrix} w_g \\ w_1 \\ \vdots \\ w_{t(x)} \\ \vdots \\ w_T \end{bmatrix}}_{w} \cdot \underbrace{\begin{bmatrix} x \\ 0 \\ \vdots \\ x\sqrt{\mu} \\ \vdots \\ 0 \end{bmatrix}}_{\Phi(x)},$$

and penalizing the squared ℓ_2 norm of this w will give :

$$\|w\|_2^2 = \|w_g\|_2^2 + \mu \sum_{t=1}^{T} \|w_t\|_2^2 = \|w_g\|_2^2 + \mu \sum_{t=1}^{T} \|v_t - w_g\|_2^2, \tag{A.1}$$

where $v_t = w_g + w_t$ is the linear function learned for task t, *i.e.*, $w^\top \Phi(x) = v_t^\top x$. It is straightforward to check from (A.1) that at the minimum, $w_g = \frac{1}{1+T} \sum_{t=1}^{T}$, that is, the part of w which is shared by all the tasks is a shrinked mean of the individual functions. Therefeore, (A.1) exactly states that penalizing the ℓ_2 norm in the joint data-task space is equivalent to enforcing a low variance of the individual functions across their mean, and a small individual ℓ_2 norm (since the v_t are close to w_g, penalizing the norm of w_g controls the norm of the v_t).

Appendix B

Interaction prediction

B.1 GPCR binding pocket

B.2 Prediction accuracy by GPCR for the first experiment

B.3 Prediction accuracy by GPCR for the second experiment

positions on β_2-adrenergic receptor	82	**109**	**110**	113	114	115	116	117	118	121	175	183	195	**199**	**200**
β_2-adrenergic receptor	M	**W**	**T**	D	V	L	C	V	T	I	R	N	T	**Y**	**A**
5-hydroxytryptamine 5A receptor	V	**W**	**I**	D	V	L	C	C	T	I	I	E	S	**Y**	**A**
Adenosine A2b receptor	V	**L**	**A**	V	L	V	L	T	Q	I	I	K	K	**M**	**V**
γ-aminobutyric acid type B receptor	E	**D**	**E**	E	A	V	E	G	H	T	L	G	S	**F**	**D**
Relaxin 3 receptor 2	L	**V**	**L**	T	V	L	N	V	Y	I	V	G	L	**Y**	**Q**

positions on β_2-adrenergic receptor	203	204	207	208	212	282	286	**289**	290	293	**308**	311	312	313	315	316
β_2-adrenergic receptor	S	S	S	F	L	F	W	**F**	F	N	**Y**	L	N	W	G	Y
5-hydroxytryptamine 5A receptor	S	T	A	F	L	F	W	**F**	F	E	**K**	F	L	W	G	Y
Adenosine A2b receptor	N	F	C	V	L	F	W	**V**	H	N	**M**	A	I	L	S	H
γ-aminobutyric acid type B receptor	G	S	A	W	E	F	L	**Y**	H	R	**L**	T	V	G	L	V
Relaxin 3 receptor 2	R	V	A	F	L	F	W	**N**	H	T	**F**	T	T	C	A	H

Table B.1: Residues of 5-hydroxytryptamine 5A receptor, Adenosine A2b receptor, Gamma-aminobutyric acid type B receptor and Relaxin 3 receptor 2 (shown as examples) aligned with β_2-adrenergic receptor binding site amino acids. The binding pocket motif of β_2-adrenergic receptor has been used as reference to determine residues involved in the formation of the binding site of the 79 other GPCRs. Bold columns correspond to the residues shown on Figure 2.2.

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin peptide receptors					
AG2R(5)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
CCKAR(6)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
CML2(1)	50.0 ± 0.0	50.0 ± 35.4	100.0 ± 0.0	50.0 ± 35.4	50.0 ± 35.4
CXCR3(1)	50.0 ± 35.4	0.0 ± 0.0	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4
EDNRA(50)	100.0 ± 0.0	99.0 ± 0.9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
EDNRB(48)	96.9 ± 1.1	91.8 ± 3.4	98.0 ± 1.1	99.0 ± 0.9	99.0 ± 0.9
GASR(2)	100.0 ± 0.0	75.0 ± 21.7	75.0 ± 21.7	75.0 ± 21.7	75.0 ± 21.7
GPR7(1)	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4
LSHR(4)	70.0 ± 11.0	70.0 ± 11.0	70.0 ± 11.0	70.0 ± 11.0	70.0 ± 11.0
NK1R(24)	92.0 ± 4.4	82.0 ± 5.2	86.0 ± 5.4	88.0 ± 3.3	86.0 ± 3.6
NK2R(1)	50.0 ± 35.4	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
NK3R(1)	50.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
OPRD(27)	92.3 ± 1.7	86.7 ± 4.4	90.3 ± 4.9	90.3 ± 2.8	90.3 ± 2.8
OPRK(24)	96.0 ± 3.6	98.0 ± 1.8	98.0 ± 1.8	98.0 ± 1.8	98.0 ± 1.8
OPRM(21)	100.0 ± 0.0	97.5 ± 2.2	97.5 ± 2.2	97.5 ± 2.2	97.5 ± 2.2
OXYR(3)	90.0 ± 8.9	100.0 ± 0.0	90.0 ± 8.9	100.0 ± 0.0	100.0 ± 0.0
SSR1(3)	90.0 ± 8.9	90.0 ± 8.9	90.0 ± 8.9	90.0 ± 8.9	90.0 ± 8.9
CCR3(1)	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4
Rhodopsin amine receptors (1/2)					
5HT1A(196)	91.6 ± 1.3	90.1 ± 2.2	88.8 ± 0.8	91.8 ± 1.5	90.8 ± 1.7
5HT1B(28)	82.7 ± 3.0	96.0 ± 3.6	98.0 ± 1.8	100.0 ± 0.0	100.0 ± 0.0
5HT1D(172)	93.3 ± 1.0	92.4 ± 0.9	92.7 ± 0.9	94.8 ± 0.7	94.8 ± 0.7
5HT1E(16)	87.5 ± 5.5	90.8 ± 3.4	96.7 ± 3.0	90.8 ± 3.4	90.8 ± 3.4
5HT1F(49)	86.7 ± 1.2	90.9 ± 0.8	88.8 ± 1.7	92.9 ± 1.1	91.7 ± 2.1
5HT2A(79)	94.9 ± 1.4	95.6 ± 1.4	93.0 ± 1.7	94.3 ± 1.7	94.9 ± 1.4
5HT2B(72)	81.2 ± 3.3	78.3 ± 2.9	83.9 ± 1.8	83.2 ± 2.0	83.2 ± 2.0
5HT2C(198)	88.6 ± 1.2	86.8 ± 1.2	89.4 ± 1.4	89.6 ± 0.8	90.1 ± 1.3
5HT4R(87)	92.5 ± 2.0	86.7 ± 2.5	85.7 ± 2.0	87.9 ± 2.1	89.0 ± 2.0
5HT5A(7)	80.0 ± 8.4	75.0 ± 10.0	75.0 ± 10.0	75.0 ± 10.0	75.0 ± 10.0
5HT6R(13)	95.0 ± 4.5	96.7 ± 3.0	91.7 ± 4.7	95.0 ± 4.5	100.0 ± 0.0
5HT7R(15)	90.0 ± 6.0	90.0 ± 3.7	96.7 ± 3.0	93.3 ± 3.7	93.3 ± 3.7
ACM1(527)	96.7 ± 0.6	94.3 ± 0.9	95.5 ± 1.0	96.1 ± 0.7	96.1 ± 0.8
ACM2(24)	82.0 ± 5.2	90.0 ± 2.8	92.0 ± 3.3	94.0 ± 3.6	92.0 ± 3.3
ACM3(58)	93.2 ± 2.6	90.5 ± 0.7	91.3 ± 1.3	96.4 ± 1.5	95.6 ± 1.3
ACM4(21)	90.0 ± 5.5	95.0 ± 2.7	95.0 ± 2.7	92.5 ± 2.7	95.0 ± 2.7
ACM5(16)	94.2 ± 3.2	94.2 ± 3.2	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin amine receptors (2/2)					
ADA1A(80)	93.1 ± 2.1	98.8 ± 0.7	99.4 ± 0.6	98.1 ± 0.7	98.8 ± 0.7
ADA1B(67)	90.5 ± 3.7	95.7 ± 1.9	98.6 ± 0.8	97.0 ± 0.7	97.0 ± 0.7
ADA1D(73)	90.4 ± 2.4	96.0 ± 1.1	98.7 ± 0.7	98.0 ± 0.7	98.0 ± 0.7
ADA2A(234)	95.7 ± 0.5	96.8 ± 0.3	98.5 ± 0.2	98.5 ± 0.2	98.5 ± 0.2
ADA2B(224)	95.1 ± 1.2	95.5 ± 1.3	98.2 ± 0.7	98.2 ± 0.7	98.0 ± 0.7
ADA2C(225)	95.3 ± 0.4	96.4 ± 0.4	97.6 ± 0.4	97.6 ± 0.4	97.8 ± 0.3
ADRB1(50)	98.0 ± 1.1	97.0 ± 1.8	99.0 ± 0.9	99.0 ± 0.9	99.0 ± 0.9
ADRB2(48)	92.8 ± 1.9	95.9 ± 0.9	96.9 ± 1.1	98.0 ± 1.1	98.0 ± 1.1
ADRB3(57)	98.2 ± 1.0	95.5 ± 2.2	97.3 ± 1.6	97.3 ± 1.6	97.3 ± 1.6
DRD1(100)	93.5 ± 1.8	94.5 ± 1.5	95.0 ± 1.4	94.5 ± 1.3	94.5 ± 1.3
DRD2(106)	93.4 ± 0.8	92.9 ± 1.8	92.4 ± 1.6	91.5 ± 1.7	91.9 ± 1.9
DRD3(41)	86.7 ± 2.6	89.2 ± 3.1	89.3 ± 3.8	90.4 ± 3.2	91.5 ± 2.8
DRD4(143)	92.3 ± 0.8	92.7 ± 1.1	93.7 ± 1.3	93.7 ± 1.4	94.1 ± 1.3
DRD5(7)	95.0 ± 4.5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
HRH1(19)	89.2 ± 4.3	92.5 ± 2.7	86.7 ± 0.7	92.5 ± 2.7	92.5 ± 2.7
HRH2(22)	91.0 ± 3.5	93.5 ± 3.7	96.0 ± 3.6	96.0 ± 3.6	96.0 ± 3.6
HRH3(88)	97.2 ± 0.8	96.1 ± 1.3	97.7 ± 0.9	97.7 ± 0.5	97.7 ± 0.5
HRH4(5)	80.0 ± 11.0	70.0 ± 17.9	100.0 ± 0.0	80.0 ± 11.0	80.0 ± 11.0

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin other receptors					
AA1R(56)	96.4 ± 1.5	96.4 ± 0.8	96.4 ± 1.5	97.3 ± 1.0	97.3 ± 1.0
AA2AR(73)	96.0 ± 1.7	97.3 ± 1.1	98.6 ± 0.8	98.0 ± 1.2	98.0 ± 1.2
AA2BR(83)	97.6 ± 1.0	98.2 ± 0.7	99.4 ± 0.6	99.4 ± 0.6	99.4 ± 0.6
AA3R(17)	97.5 ± 2.2	82.5 ± 1.8	94.2 ± 3.2	95.0 ± 4.5	95.0 ± 4.5
CLTR1(18)	89.2 ± 2.5	84.2 ± 4.1	89.2 ± 4.3	91.7 ± 3.1	91.7 ± 3.1
LT4R1(2)	50.0 ± 25.0	50.0 ± 25.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
LT4R2(2)	50.0 ± 25.0	50.0 ± 25.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MTR1A(91)	97.3 ± 1.1	96.8 ± 1.4	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MTR1B(90)	97.8 ± 0.9	97.8 ± 0.9	99.4 ± 0.5	99.4 ± 0.5	99.4 ± 0.5
MTR1L(75)	98.7 ± 0.7	99.3 ± 0.6	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PAFR(1)	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4	50.0 ± 35.4
PE2R1(5)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PE2R2(7)	100.0 ± 0.0	95.0 ± 4.5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PE2R3(5)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PE2R4(5)	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
R3R2(1)	50.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
TA2R(63)	100.0 ± 0.0	99.2 ± 0.7	99.2 ± 0.7	100.0 ± 0.0	100.0 ± 0.0
Metabotropic glutamate family					
GABR1(1)	50.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 35.4	100.0 ± 0.0
GABR2(1)	50.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	0.0 ± 0.0	50.0 ± 35.4
MGR1(34)	98.3 ± 1.5	91.4 ± 4.7	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR2(6)	95.0 ± 4.5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR3(5)	100.0 ± 0.0	90.0 ± 8.9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR5(5)	90.0 ± 8.9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR6(5)	100.0 ± 0.0	90.0 ± 8.9	90.0 ± 8.9	100.0 ± 0.0	90.0 ± 8.9
MGR7(6)	95.0 ± 4.5	90.0 ± 8.9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR8(3)	80.0 ± 17.9	80.0 ± 17.9	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
Secretin family					
VIPR1(1)	50.0 ± 35.4	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 35.4	100.0 ± 0.0

Table B.2: Prediction accuracy by GPCR for the first experiment. Mean prediction accuracy for each GPCR for the first experiment with the 2D Tanimoto ligand kernel and various target kernels. The GPCR identifiers are the GLIDA references. The numbers in bracket are the numbers ligands considered in the experiment for each GPCR. BP is the binding pocket kernel and PBP the poly binding pocket kernel.

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin peptide receptors					
AG2R(5)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0
CCKAR(6)	50.0 ± 50.0	50.0 ± 50.0	66.7 ± 47.1	50.0 ± 50.0	50.0 ± 50.0
CML2(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
CXCR3(1)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0
EDNRA(50)	50.0 ± 50.0	46.0 ± 49.8	100.0 ± 0.0	100.0 ± 0.0	94.0 ± 23.7
EDNRB(48)	50.0 ± 50.0	22.9 ± 42.0	74.0 ± 43.9	95.8 ± 20.0	75.0 ± 43.3
GASR(2)	50.0 ± 50.0	25.0 ± 43.3	75.0 ± 43.3	75.0 ± 43.3	50.0 ± 50.0
GPR7(1)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	100.0 ± 0.0	50.0 ± 50.0
LSHR(4)	50.0 ± 50.0	0.0 ± 0.0	37.5 ± 48.4	50.0 ± 50.0	37.5 ± 48.4
NK1R(24)	50.0 ± 50.0	27.1 ± 44.4	33.3 ± 47.1	60.4 ± 48.9	39.6 ± 48.9
NK2R(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
NK3R(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
OPRD(27)	50.0 ± 50.0	37.0 ± 48.3	44.4 ± 49.7	55.6 ± 49.7	55.6 ± 49.7
OPRK(24)	50.0 ± 50.0	47.9 ± 50.0	81.2 ± 39.0	87.5 ± 33.1	83.3 ± 37.3
OPRM(21)	50.0 ± 50.0	54.8 ± 49.8	88.1 ± 32.4	90.5 ± 29.4	90.5 ± 29.4
OXYR(3)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	66.7 ± 47.1	50.0 ± 50.0
SSR1(3)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0
CCR3(1)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0
Rhodopsin amine receptors (1/2)					
5HT1A(196)	50.0 ± 50.0	39.8 ± 48.9	53.3 ± 49.9	49.0 ± 50.0	45.7 ± 49.8
5HT1B(28)	50.0 ± 50.0	50.0 ± 50.0	89.3 ± 30.9	91.1 ± 28.5	89.3 ± 30.9
5HT1D(172)	50.0 ± 50.0	29.7 ± 45.7	56.7 ± 49.6	59.3 ± 49.1	57.3 ± 49.5
5HT1E(16)	50.0 ± 50.0	68.8 ± 46.4	93.8 ± 24.2	93.8 ± 24.2	90.6 ± 29.1
5HT1F(49)	50.0 ± 50.0	41.8 ± 49.3	58.2 ± 49.3	60.2 ± 48.9	56.1 ± 49.6
5HT2A(79)	50.0 ± 50.0	68.4 ± 46.5	76.6 ± 42.3	77.2 ± 41.9	76.6 ± 42.3
5HT2B(72)	50.0 ± 50.0	31.2 ± 46.4	70.8 ± 45.5	56.9 ± 49.5	55.6 ± 49.7
5HT2C(198)	50.0 ± 50.0	35.1 ± 47.7	60.4 ± 48.9	52.0 ± 50.0	48.2 ± 50.0
5HT4R(87)	50.0 ± 50.0	20.1 ± 40.1	29.9 ± 45.8	34.5 ± 47.5	31.6 ± 46.5
5HT5A(7)	50.0 ± 50.0	78.6 ± 41.0	64.3 ± 47.9	78.6 ± 41.0	78.6 ± 41.0
5HT6R(13)	50.0 ± 50.0	92.3 ± 26.6	80.8 ± 39.4	84.6 ± 36.1	88.5 ± 31.9
5HT7R(15)	50.0 ± 50.0	93.3 ± 24.9	93.3 ± 24.9	90.0 ± 30.0	90.0 ± 30.0
ACM1(527)	50.0 ± 50.0	30.2 ± 45.9	43.3 ± 49.5	48.8 ± 50.0	45.6 ± 49.8
ACM2(24)	50.0 ± 50.0	58.3 ± 49.3	81.2 ± 39.0	91.7 ± 27.6	87.5 ± 33.1
ACM3(58)	50.0 ± 50.0	35.3 ± 47.8	59.5 ± 49.1	73.3 ± 44.3	70.7 ± 45.5
ACM4(21)	50.0 ± 50.0	90.5 ± 29.4	76.2 ± 42.6	78.6 ± 41.0	81.0 ± 39.3
ACM5(16)	50.0 ± 50.0	84.4 ± 36.3	84.4 ± 36.3	78.1 ± 41.3	84.4 ± 36.3

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin amine receptors (2/2)					
ADA1A(80)	50.0 ± 50.0	76.9 ± 42.2	97.5 ± 15.6	96.2 ± 19.0	96.2 ± 19.0
ADA1B(67)	50.0 ± 50.0	74.6 ± 43.5	91.8 ± 27.5	91.0 ± 28.6	91.8 ± 27.5
ADA1D(73)	50.0 ± 50.0	79.5 ± 40.4	98.6 ± 11.6	97.9 ± 14.2	97.9 ± 14.2
ADA2A(234)	50.0 ± 50.0	58.3 ± 49.3	93.8 ± 24.1	91.0 ± 28.6	89.7 ± 30.3
ADA2B(224)	50.0 ± 50.0	57.4 ± 49.5	97.5 ± 15.5	97.3 ± 16.1	95.3 ± 21.1
ADA2C(225)	50.0 ± 50.0	59.6 ± 49.1	94.9 ± 22.0	94.2 ± 23.3	94.7 ± 22.5
ADRB1(50)	50.0 ± 50.0	44.0 ± 49.6	87.0 ± 33.6	84.0 ± 36.7	84.0 ± 36.7
ADRB2(48)	50.0 ± 50.0	52.1 ± 50.0	95.8 ± 20.0	89.6 ± 30.5	90.6 ± 29.1
ADRB3(57)	50.0 ± 50.0	46.5 ± 49.9	86.0 ± 34.7	82.5 ± 38.0	81.6 ± 38.8
DRD1(100)	50.0 ± 50.0	43.0 ± 49.5	45.5 ± 49.8	43.0 ± 49.5	43.0 ± 49.5
DRD2(106)	50.0 ± 50.0	50.5 ± 50.0	54.7 ± 49.8	59.9 ± 49.0	55.7 ± 49.7
DRD3(41)	50.0 ± 50.0	57.3 ± 49.5	70.7 ± 45.5	74.4 ± 43.6	69.5 ± 46.0
DRD4(143)	50.0 ± 50.0	45.8 ± 49.8	45.1 ± 49.8	51.0 ± 50.0	49.3 ± 50.0
DRD5(7)	50.0 ± 50.0	57.1 ± 49.5	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
HRH1(19)	50.0 ± 50.0	55.3 ± 49.7	57.9 ± 49.4	68.4 ± 46.5	68.4 ± 46.5
HRH2(22)	50.0 ± 50.0	45.5 ± 49.8	52.3 ± 49.9	56.8 ± 49.5	56.8 ± 49.5
HRH3(88)	50.0 ± 50.0	39.2 ± 48.8	49.4 ± 50.0	46.0 ± 49.8	46.0 ± 49.8
HRH4(5)	50.0 ± 50.0	70.0 ± 45.8	90.0 ± 30.0	70.0 ± 45.8	70.0 ± 45.8

GPCR $\backslash K_{tar}$	Dirac	multitask	hierarchy	BP	PBP
Rhodopsin other receptors					
AA1R(56)	50.0 ± 50.0	39.3 ± 48.8	91.1 ± 28.5	92.9 ± 25.8	86.6 ± 34.1
AA2AR(73)	50.0 ± 50.0	46.6 ± 49.9	94.5 ± 22.8	96.6 ± 18.2	95.2 ± 21.4
AA2BR(83)	50.0 ± 50.0	37.3 ± 48.4	87.3 ± 33.2	98.2 ± 13.3	89.2 ± 31.1
AA3R(17)	50.0 ± 50.0	38.2 ± 48.6	64.7 ± 47.8	70.6 ± 45.6	52.9 ± 49.9
CLTR1(18)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	52.8 ± 49.9
LT4R1(2)	50.0 ± 50.0	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
LT4R2(2)	50.0 ± 50.0	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MTR1A(91)	50.0 ± 50.0	43.4 ± 49.6	97.3 ± 16.3	97.3 ± 16.3	95.6 ± 20.5
MTR1B(90)	50.0 ± 50.0	47.2 ± 49.9	95.6 ± 20.6	97.8 ± 14.7	97.8 ± 14.7
MTR1L(75)	50.0 ± 50.0	46.7 ± 49.9	99.3 ± 8.1	100.0 ± 0.0	99.3 ± 8.1
PAFR(1)	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0	50.0 ± 50.0
PE2R1(5)	50.0 ± 50.0	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PE2R2(7)	50.0 ± 50.0	42.9 ± 49.5	92.9 ± 25.8	85.7 ± 35.0	85.7 ± 35.0
PE2R3(5)	50.0 ± 50.0	60.0 ± 49.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
PE2R4(5)	50.0 ± 50.0	60.0 ± 49.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
R3R2(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
TA2R(63)	50.0 ± 50.0	42.1 ± 49.4	47.6 ± 49.9	50.8 ± 50.0	49.2 ± 50.0
Metabotropic glutamate family					
GABR1(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
GABR2(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 50.0	100.0 ± 0.0
MGR1(34)	50.0 ± 50.0	42.6 ± 49.5	63.2 ± 48.2	61.8 ± 48.6	64.7 ± 47.8
MGR2(6)	50.0 ± 50.0	58.3 ± 49.3	100.0 ± 0.0	100.0 ± 0.0	83.3 ± 37.3
MGR3(5)	50.0 ± 50.0	70.0 ± 45.8	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR5(5)	50.0 ± 50.0	90.0 ± 30.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
MGR6(5)	50.0 ± 50.0	90.0 ± 30.0	90.0 ± 30.0	90.0 ± 30.0	90.0 ± 30.0
MGR7(6)	50.0 ± 50.0	83.3 ± 37.3	91.7 ± 27.6	83.3 ± 37.3	83.3 ± 37.3
MGR8(3)	50.0 ± 50.0	83.3 ± 37.3	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
Secretin family					
VIPR1(1)	50.0 ± 50.0	100.0 ± 0.0	100.0 ± 0.0	50.0 ± 50.0	100.0 ± 0.0

Table B.3: Prediction accuracy by GPCR for the second experiment. Mean prediction accuracy for each GPCR for the second experiment with the 2D Tanimoto ligand kernel and various target kernels. The GPCR identifiers are the GLIDA references. The numbers in bracket are the numbers ligands considered in the experiment for each GPCR. BP is the binding pocket kernel and PBP the poly binding pocket kernel.

Bibliography

J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. Low-rank matrix factorization with attributes. Technical Report cs/0611124, arXiv, 2006.

J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: operator estimation with spectral regularization. *J. Mach. Learn. Res.*, 2008. In press.

R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422(6928): 198–207, Mar 2003. URL `http://dx.doi.org/10.1038/nature01511`.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In P. B. N. and C. F., editors, *Proc. of the 2nd Int. Symp. on Information Theory*, pages 267–281, 1973.

C. Aliferis, D. Hardin, and P. Massion. Machine Learning Models For Lung Cancer Classification Using Array Comparative Genomic Hybridization. In *Proceedings of the 2002 American Medical Informatics Association (AMIA) Annual Symposium*, pages 7–11, 2002.

A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, Feb 2000. URL http://dx.doi.org/10.1038/35000501.

Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 17–24, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. URL http://doi.acm.org/10.1145/1273496.1273499.

R. K. Ando, T. Zhang, and P. Bartlett. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.

I. Antes, S. W. I. Siu, and T. Lengauer. DynaPred: a structure and sequence based method for the prediction of MHC class I binding peptide sequences and conformations. *Bioinformatics*, 22(14):e16–e24, Jul 2006. URL http://dx.doi.org/10.1093/bioinformatics/btl216.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Adv. Neural. Inform. Process Syst. 19*, pages 41–48, Cambridge, MA, 2007. MIT Press.

A. Argyriou, C. A. Micchelli, and M. Pontil. When is there a representer theorem? vector versus matrix regularizers. *CoRR*, abs/0809.1590, 2008a.

A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 25–32. MIT Press, Cambridge, MA, 2008b.

N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337 – 404, 1950.

T. K. Attwood, P. Bradley, D. R. Flower, A. Gaulton, N. Maudling, A. L. Mitchell, G. Moulton, A. Nordle, K. Paine, P. Taylor, A. Uddin, and C. Zygouri. Prints and its automatic supplement, preprints. *Nucleic Acids Res.*, 31(1):400–402, Jan 2003.

C. Auliac, V. Frouin, X. Gidrol, and F. d'Alché Buc. Evolutionary approaches for the reverse-engineering of gene regulatory networks: A study on a biologically realistic dataset. *BMC Bioinformatics*, 9, 2008.

C.-A. Azencott, A. Ksikes, S. J. Swamidass, J. H. Chen, L. Ralaivola, and P. Baldi. One- to four-dimensional kernels for virtual screening and the prediction of physical, chemical, and biological properties. *J. Chem. Inform. Model.*, 47(3):965–974, 2007. URL `http://dx.doi.org/10.1021/ci600397p`.

F. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008a. URL `http://jmlr.csail.mit.edu/papers/v9/bach08b.html`.

F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Adv. Neural. Inform. Process Syst.*, volume 21, 2009.

F. R. Bach. Consistency of trace norm minimization. *J. Mach. Learn. Res.*, 9:1019–1048, 2008b. URL `http://jmlr.csail.mit.edu/papers/volume9/bach08a/bach08a.pdf`.

F. R. Bach, G. Lanckriet, and M. I. Jordan. Fast kernel learning using sequential minimal optimization. Technical Report UCB/CSD-04-1307, Computer Science Division, UC Berkeley, February 2004a.

F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 6, New York, NY, USA, 2004b. ACM. URL `http://doi.acm.org/10.1145/1015330.1015424`.

F. Bach and M. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3:1–48, 2002. URL `http://jmlr.csail.mit.edu/papers/v3/bach02a.html`.

F. R. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *ICML*

'08: Proceedings of the 25th international conference on Machine learning, pages 33–40, New York, NY, USA, 2008c. ACM. ISBN 978-1-60558-205-4. URL `http://doi.acm.org/10.1145/1390156.1390161`.

F. R. Bach and M. I. Jordan. Learning spectral clustering. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

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 ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 2:28–36, 1994. ISSN 1553-0833. URL `http://view.ncbi.nlm.nih.gov/pubmed/7584402`.

B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *J. Mach. Learn. Res.*, 4:83–99, 2003. ISSN 1533-7928.

K. V. Balakin, S. E. Tkachenko, S. A. Lang, I. Okun, A. A. Ivashchenko, and N. P. Savchuk. Property-based design of GPCR-targeted library. *J. Chem. Inf. Comput. Sci.*, 42(6):1332–1342, 2002.

J. Ballesteros and K. Palczewski. G protein-coupled receptor drug discovery: implications from the crystal structure of rhodopsin. *Curr. Opin. Drug Discov. Devel.*, 4(5):561–574, Sep 2001.

M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo. How to infer gene networks from expression profiles. *Mol Syst Biol*, 3:78, 2007. URL `http://dx.doi.org/10.1038/msb4100120`.

Y. Baraud, C. Giraud, and S. Huet. Gaussian model selection with an unknown variance. *Annals Of Statistics, to appear*, 37:630, 2009. URL `http://doi:10.1214/07-AOS573`.

J. Baxter. A bayesian/information theoretic model of bias learning. In *COLT '96: Proceedings of the ninth annual conference on Computational learning theory*, pages 77–88, New York, NY, USA, 1996a. ACM Press. ISBN 0-89791-811-8. URL `http://doi.acm.org/10.1145/238061.238071`.

J. Baxter. Learning model bias. In *Advances in Neural Information Processing Systems*, pages 169–175. MIT Press, 1996b.

J. Baxter. A bayesian/information theoretic model of learning to learn via multiple task sampling. In *Machine Learning*, pages 7–39, 1997.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000. URL http://citeseer.ist.psu.edu/article/baxter00model.html.

M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild. A Bayesian approach to reconstructing genetic regulatory networks with hidden factors. *Bioinformatics*, 21(3): 349–356, Feb 2005. URL http://dx.doi.org/10.1093/bioinformatics/bti014.

O. M. Becker, Y. Marantz, S. Shacham, B. Inbal, A. Heifetz, O. Kalid, S. Bar-Haim, D. Warshaviak, M. Fichman, and S. Noiman. G protein-coupled receptors: in silico drug discovery in 3D. *Proc. Natl. Acad. Sci. USA*, 101(31):11304–11309, Aug 2004. URL http://dx.doi.org/10.1073/pnas.0401862101.

G. Bejerano and G. Yona. Modeling protein families using probabilistic suffix trees. In *Proceedings of RECOMB 1999*, pages 15–24. ACM Press, 1999.

S. Ben-David and R. Schuller. Exploiting task relatedness for multiple task learning, 2003. URL http://citeseer.ist.psu.edu/ben-david03exploiting.html.

A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *J. Comput. Biol.*, 7(3-4):559–583, 2000. URL http://www.liebertonline.com/doi/abs/10.1089/106652700750050943.

A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Bioinformatics*, 19(Suppl. 1):i26–i33, 2003. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/19/suppl_1/i26.

A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(Suppl. 1):i38–i46, Jun 2005. URL `http://dx.doi.org/10.1093/bioinformatics/bti1016`.

M. Benito, J. Parker, Q. Du, J. Wu, D. Xiang, C. M. Perou, and J. S. Marron. Adjustment of systematic microarray data biases. *Bioinformatics*, 20(1):105–14, Jan 2004.

C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic analysis on semigroups*. Springer-Verlag, New-York, 1984.

J. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, 1985.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2003.

M. Bhasin and G. P. S. Raghava. GPCRpred: an SVM-based method for prediction of families and subfamilies of G-protein coupled receptors. *Nucl. Acids Res.*, 32(Supp.2): W383–389, 2004a. URL `http://dx.doi.org/10.1093/nar/gkh416`.

M. Bhasin and G. P. S. Raghava. Prediction of CTL epitopes using QM, SVM and ANN techniques. *Vaccine*, 22(23-24):3195–3204, 2004b. URL `http://dx.doi.org/10.1016/j.vaccine.2004.02.005`.

M. Bhasin, H. Singh, and G. P. S. Raghava. MHCBN: a comprehensive database of MHC binding and non-binding peptides. *Bioinformatics*, 19(5):665–666, Mar 2003.

A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. A. Sugarbaker, and M. Meyerson. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl. Acad. Sci. USA*, 98(24):13790–13795, Nov 2001. URL `http://dx.doi.org/10.1073/pnas.191502998`.

S. Bickel, M. Brückner, and T. Scheffer. Discriminative learning for differing training and test distributions. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 81–88. ACM Press, 2007.

S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for hiv therapy screening. In *ICML'08: Proceedings of the 25th international conference on Machine learning*, pages 56–63, 2008.

A. H. Bild, G. Yao, J. T. Chang, Q. Wang, A. Potti, D. Chasse, M. B. Joshi, D. Harpole, J. M. Lancaster, A. Berchuck, J. Olson, J. A., J. R. Marks, H. K. Dressman, M. West, and J. R. Nevins. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature*, 439(7074):353–7, 2006. URL `http://dx.doi.org/10.1038/nature04296`.

L. Birgé and P. Massart. Minimal penalties for gaussian model selection. *Probab. Theory Relat. Fields*, 138:33–73, 2006. URL `http://dx.doi.org/10.1007/s00440-006-0011-8`.

C. Bissantz, P. Bernard, M. Hibert, and D. Rognan. Protein-based virtual screening of chemical databases. II. are homology models of G-protein coupled receptors suitable targets? *Proteins*, 50(1):5–25, Jan 2003. URL `http://dx.doi.org/10.1002/prot.10237`.

K. Bleakley, G. Biau, and J.-P. Vert. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65, Jul 2007. URL `http://dx.doi.org/10.1093/bioinformatics/btm204`.

J. R. Bock and D. A. Gough. Virtual screen for ligands of orphan G protein-coupled receptors. *J. Chem. Inform. Model.*, 45(5):1402–1414, 2005. URL `http://dx.doi.org/10.1021/ci050006d`.

J. Bockaert and J. P. Pin. Molecular tinkering of G protein-coupled receptors: an evolutionary success. *EMBO J.*, 18(7):1723–1729, Apr 1999. URL `http://dx.doi.org/10.1093/emboj/18.7.1723`.

E. Bonilla, K. M. Chai, and C. Williams. Multi-task gaussian process prediction. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

E. V. Bonilla, F. V. Agakov, and C. K. I. Williams. Kernel multi-task learning using task-specific features. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*. Omnipress, March 2007.

K. M. Borgwardt and H.-P. Kriegel. Shortest-path kernels on graphs. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2278-5. URL `http://dx.doi.org/10.1109/ICDM.2005.132`.

K. Borgwardt, C. Ong, S. Schönauer, S. Vishwanathan, A. Smola, and H.-P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(Suppl. 1):i47–i56, Jun 2005. URL `http://dx.doi.org/10.1093/bioinformatics/bti1007`.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th annual ACM workshop on Computational Learning Theory*, pages 144–152, New York, NY, USA, 1992. ACM Press. URL `http://www.clopinet.com/isabelle/Papers/colt92.ps.Z`.

A. Bouchard-Côté, M. I. Jordan, and D. Klein. Efficient inference in phylogenetic indel trees. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *NIPS*, pages 177–184. MIT Press, 2008. URL `http://dblp.uni-trier.de/db/conf/nips/nips2008.html#Bouchard-CoteJK08`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, W.I., 1998. Morgan Kaufman.

P. J. Brown and J. V. Zidek. Adaptive multivariate ridge regression. *Ann. Statist.*, 8(1): 64–74, 1980.

P. Brown and D. Botstein. Exploring the new world of the genome with DNA microarrays. *Nat. Genet.*, 21:33–37, 2000. URL `http://www.nature.com/ng/journal/v21/n1s/abs/ng0199supp_33.html`.

V. Brusic, N. Petrovsky, G. Zhang, and V. B. Bajic. Prediction of promiscuous peptides that bind HLA class I molecules. *Immunol. Cell Biol.*, 80(3):280–285, Jun 2002.

H.-H. Bui, A. J. Schiewe, H. von Grafenstein, and I. S. Haworth. Structural prediction of peptides binding to MHC class I molecules. *Proteins*, 63(1):43–52, Apr 2006. URL `http://dx.doi.org/10.1002/prot.20870`.

H.-H. Bui, J. Sidney, B. Peters, M. Sathiamurthy, A. Sinichi, K.-A. Purton, B. R. Mothé, F. V. Chisari, D. I. Watkins, and A. Sette. Automated generation and evaluation of specific mhc binding predictive tools: Arb matrix applications. *Immunogenetics*, 57(5):304–314, Jun 2005. URL `http://dx.doi.org/10.1007/s00251-005-0798-y`.

R. Burbidge, M. Trotter, B. Buxton, and S. Holden. Drug design by machine learning: support vector machines for pharmaceutical data analysis. *Comput. Chem.*, 26 (1):4–15, December 2001. URL `http://stats.ma.ic.ac.uk/~rdb/pubs/candc-aisb00-rbmt-final.pdf`.

D. Butina, M. D. Segall, and K. Frankcombe. Predicting ADME properties in silico: methods and models. *Drug Discov Today*, 7(11 Suppl):S83–S88, Jun 2002.

S. Buus, S. L. Lauemøller, P. Worning, C. Kesmir, T. Frimurer, S. Corbet, A. Fomsgaard, J. Hilden, A. Holm, and S. Brunak. Sensitive quantitative predictions of peptide-MHC binding by a 'query by committee' artificial neural network approach. *Tissue Antigens*, 62(5):378–384, Nov 2003.

E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. *J Chem Inf Comput Sci*, 43(6):1882–9, 2003. URL `http://dx.doi.org/10.1021/ci0341161`.

C. Cai, W. Wang, L. Sun, and Y. Chen. Protein function classification via support vector machine approach. *Math. Biosci.*, 185(2):111–122, 2003. URL `10.1016/S0025-5564(03)00096-8`.

C. Cai, L. Han, Z. Ji, and Y. Chen. Enzyme family classification by support vector machines. *Proteins*, 55(1):66–76, 2004. URL http://dx.doi.org/10.1002/prot.20045.

J. Caldwell, I. Gardner, and N. Swales. An introduction to drug disposition: the basic principles of absorption, distribution, metabolism, and excretion. *Toxicol. Pathol.*, 23 (2):102–114, 1995.

E. Candes. The restricted isometry property. *Compte Rendus de l'Académie des Sciences, Paris*, 1(346):589–592, 2008.

E. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Ann. Stat.*, 35(6):2313–2351, 2007. URL http://dx.doi.org/10.1214/009053606000001523.

E. Candes, J. K. Romberk, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006.

M. Carlo, S. Li, D. K. Pearl, and H. Doss. Phylogenetic tree construction using markov chain monte carlo. *Journal of the American Statistical Association*, 95:493–508, 1999.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann, 1993.

L. A. Catapano and H. K. Manji. G protein-coupled receptors in major psychiatric disorders. *Biochim. Biophys. Acta*, 1768(4):976–993, Apr 2007. URL http://dx.doi.org/10.1016/j.bbamem.2006.09.025.

C. N. Cavasotto, A. J. W. Orry, and R. A. Abagyan. Structure-based identification of binding sites, native ligands and potential inhibitors for G-protein coupled receptors. *Proteins*, 51(3):423–433, May 2003. URL `http://dx.doi.org/10.1002/prot.10362`.

C. N. Cavasotto, A. J. W. Orry, N. J. Murgolo, M. F. Czarniecki, S. A. Kocsi, B. E. Hawes, K. A. O'Neill, H. Hine, M. S. Burton, J. H. Voigt, R. A. Abagyan, M. L. Bayne, and F. J. Monsma. Discovery of novel chemotypes to a G-protein-coupled receptor through ligand-steered homology modeling and structure-based virtual screening. *J. Med. Chem.*, 51(3):581–588, Feb 2008. URL `http://dx.doi.org/10.1021/jm070759m`.

O. Chapelle and Z. Harchaoui. A machine learning approach to conjoint analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 257–264. MIT Press, Cambridge, MA, 2005.

J.-Z. Chen, J. Wang, and X.-Q. Xie. Gpcr structure-based virtual screening approach for cb2 antagonist search. *J. Chem. Inf. Model.*, 47(4):1626–1637, 2007. URL `http://dx.doi.org/10.1021/ci7000814`.

J. Chen, L. Tang, J. Liu, and J. Ye. A convex formulation for learning shared structures from multiple tasks. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 137–144, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. URL `http://doi.acm.org/10.1145/1553374.1553392`.

S. S. Chen, D. L. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998. URL `http://dx.doi.org/10.1137/S1064827596304010`.

R. Chenna, H. Sugawara, T. Koike, R. Lopez, T. J. Gibson, D. G. Higgins, and J. D. Thompson. Multiple sequence alignment with the Clustal series of programs. *Nucleic Acids Res.*, 31(13):3497–3500, Jul 2003.

S.-F. Chin, Y. Wang, N. P. Thorne, A. E. Teschendorff, S. E. Pinder, M. Vias, A. Naderi,

I. Roberts, N. L. Barbosa-Morais, M. J. Garcia, N. G. Iyer, T. Kranjac, J. F. R. Robertson, S. Aparicio, S. Tavare, I. Ellis, J. D. Brenton, and C. Caldas. Using array-comparative genomic hybridization to define molecular portraits of primary breast cancers. *Oncogene*, 26(13):1959–1970, September 2006. ISSN 0950-9232. URL `http://dx.doi.org/10.1038/sj.onc.1209985`.

S. F. Chin, A. E. Teschendorff, J. C. Marioni, Y. Wang, N. L. Barbosa-Morais, N. P. Thorne, J. L. Costa, S. E. Pinder, M. A. van de Wiel, A. R. Green, I. O. Ellis, P. L. Porter, S. Tavaré, J. D. Brenton, B. Ylstra, and C. Caldas. High-resolution aCGH and expression profiling identifies a novel genomic subtype of ER negative breast cancer. *Genome Biol.*, 8(10):R215, 2007. URL `http://dx.doi.org/10.1186/gb-2007-8-10-r215`.

H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.*, 3:140, 2007. URL `http://dx.doi.org/10.1038/msb4100180`.

P. Comon. Independent component analysis: a new concept? *Signal Processing*, 36(3): 287–314, 1994.

N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. URL `http://www.support-vector.net`.

K. R. Curtis, M. Oresic, and A. Vidal-Puig. Pathways to the analysis of microarray data. *Trends in Biotechnology*, 23(8):429–435, 2005. URL `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=15950303`.

M. Cuturi and J.-P. Vert. The context-tree kernel for strings. *Neural Network.*, 18(4):1111–1123, 2005. URL `http://dx.doi.org/10.1016/j.neunet.2005.07.010`.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

H. Daume. Bayesian Multitask Learning with Latent Hierarchies. In *25th Conference on Uncertainty in Artificial Intelligence*, 2009.

M. H. De Groot. *Optimal statistical decisions / Morris H. De Groot*. McGraw-Hill, New York :,, 1970.

C. Debouck and P. N. Goodfellow. DNA microarrays in drug discovery and development. *Nat. Genet.*, 21(1 Suppl):48–50, Jan 1999. URL `http://dx.doi.org/10.1038/4475`.

M. Deodhar and J. Ghosh. A framework for simultaneous co-clustering and learning from complex data. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 250–259, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-609-7. URL `http://doi.acm.org/10.1145/1281192.1281222`.

J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, 1997. URL `http://www.sciencemag.org/cgi/reprint/278/5338/680.pdf`.

D. A. Deshpande and R. B. Penn. Targeting G protein-coupled receptor signaling in asthma. *Cell. Signal.*, 18(12):2105–2120, Dec 2006. URL `http://dx.doi.org/10.1016/j.cellsig.2006.04.008`.

X. Deupi, N. Dölker, M. L. Lòpez-Rodrìguez, M. Campillo, J. A. Ballesteros, and L. Pardo. Structural models of class a G protein-coupled receptors as a tool for drug design: insights on transmembrane bundle plasticity. *Curr. Top. Med. Chem.*, 7(10):991–998, 2007.

V. Dhingra, M. Gupta, T. Andacht, and Z. F. Fu. New frontiers in proteomics research: a perspective. *Int. J. Pharm.*, 299(1-2):1–18, Aug 2005. URL `http://dx.doi.org/10.1016/j.ijpharm.2005.04.010`.

T. Dietterich, R. Lathrop, and T. Lozano-Perez. Solving the Multiple Instance Problem with Axis-Parallel Rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

C. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, 17:349–358, 2001. URL http://bioinformatics.oupjournals.org/cgi/reprint/17/4/349.pdf.

P. Dobson and A. Doig. Predicting enzyme class from protein structure without alignments. *J. Mol. Biol.*, 345(1):187–199, Jan 2005. URL http://dx.doi.org/10.1016/j.jmb.2004.10.024.

P. Dönnes and A. Elofsson. Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics*, 3(1):25, Sep 2002. URL http://www.biomedcentral.com/1471-2105/3/25/abstract.

D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. IT*, 41(3):613–627, 1994.

I. A. Doytchinova, P. Guan, and D. R. Flower. Identifying human MHC supertypes using bioinformatic methods. *J. Immunol.*, 172(7):4314–4323, Apr 2004.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In A. McCallum and S. Roweis, editors, *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 272–279. Omnipress, 2008.

D. Dunson, Y. Xue, and L. Carin. The matrix stick-breaking process: Flexible bayes meta-analysis. *Journal of the American Statistical Association*, 103(481):317–327, March 2008. URL http://www.ingentaconnect.com/content/asa/jasa/2008/00000103/00000481/art00036.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32 (2):407–499, 2004.

W. J. Egan, K. M. Merz, and J. J. Baldwin. Prediction of drug absorption using multivariate statistics. *J. Med. Chem.*, 43(21):3867–3877, Oct 2000.

D. Erhan, P.-J. L'heureux, S. Y. Yue, and Y. Bengio. Collaborative filtering on a family of biological targets. *J. Chem. Inf. Model.*, 46(2):626–635, 2006. URL `http://dx.doi.org/10.1021/ci050367t`.

A. Evers and T. Klabunde. Structure-based drug discovery using GPCR homology modeling: successful virtual screening for antagonists of the alpha1A adrenergic receptor. *J. Med. Chem.*, 48(4):1088–1097, Feb 2005. URL `http://dx.doi.org/10.1021/jm0491804`.

T. Evgeniou, C. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637, 2005. URL `http://jmlr.csail.mit.edu/papers/volume6/evgeniou05a`.

T. Evgeniou and M. Pontil. Regularized multi–task learning. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. URL `http://doi.acm.org/10.1145/1014052.1014067`.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference*, volume 6, pages 4734–4739, 2001. URL `http://dx.doi.org/10.1109/ACC.2001.945730`.

J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.

T. L. Ferea, D. Botstein, P. O. Brown, and R. F. Rosenzweig. Systematic changes in gene expression patterns following adaptive evolution in yeast. *Proc. Natl. Acad. Sci. USA*, 96 (17):9721–9726, 1999. URL `http://www.pnas.org/cgi/reprint/96/17/9721.pdf`.

D. P. Foster and E. I. George. The risk inflation criterion for multiple regression. *The Annals of Statistics*, 22(4):1947–1975, 1994. ISSN 00905364. URL `http://dx.doi.org/10.2307/2242493`.

S. Foucart and M.-J. Lai. Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$. *Applied and Computational Harmonic Analysis*, 26 (3):395–407, May 2009. URL http://dx.doi.org/10.1016/j.acha.2008.09.001.

B. B. Fredholm, T. Hökfelt, and G. Milligan. G-protein-coupled receptors: an update. *Acta Physiol.*, 190(1):3–7, May 2007. URL http://dx.doi.org/10.1111/j.1365-201X.2007.01689.x.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, pages kxm045+, December 2007. URL http://dx.doi.org/10.1093/biostatistics/kxm045.

T. M. Frimurer, T. Ulven, C. E. Elling, L.-O. Gerlach, E. Kostenis, and T. Högberg. A physicogenetic method to assign ligand-binding relationships between 7tm receptors. *Bioorg. Med. Chem. Lett.*, 15(16):3707–3712, Aug 2005. URL http://dx.doi.org/10.1016/j.bmcl.2005.05.102.

M. C. Frith, N. F. W. Saunders, B. Kobe, and T. L. Bailey. Discovering sequence motifs with arbitrary insertions and deletions. *PLoS Comput. Biol.*, 4(5):e1000071+, May 2008. ISSN 1553-7358. URL http://dx.doi.org/10.1371/journal.pcbi.1000071.

H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd international conference on Machine learning*, pages 225 – 232, New York, NY, USA, 2005. ACM Press. URL http://doi.acm.org/10.1145/1102351.1102380.

W. Fu. Penalized regressions: the bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7:397–416, 1998.

W. Fu, P. Ray, and E. P. Xing. Discover: a feature-based discriminative method for motif search in complex genomes. *Bioinformatics (Oxford, England)*, 25(12): i321–329, June 2009. ISSN 1460-2059. URL http://dx.doi.org/10.1093/bioinformatics/btp230.

T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: hardness results and efficient alternatives. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory and the Seventh Annual Workshop on Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 129–143, Heidelberg, 2003. Springer. URL http://dx.doi.org/10.1007/b12006.

A. Gasch, M. Huang, S. Metzner, D. Botstein, S. Elledge, and P. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol. Biol. Cell*, 12(10):2987–3003, 2001. URL http://www.molbiolcell.org/cgi/content/full/12/10/2987.

J. Gasteiger and T. Engel, editors. *Chemoinformatics : a Textbook*. Wiley, New York, NY, USA, 2003.

U. Gether. Uncovering molecular mechanisms involved in activation of g protein-coupled receptors. *Endocr Rev*, 21(1):90–113, Feb 2000.

D. Ghosh and A. M. Chinnaiyan. Classification and Selection of Biomarkers in Genomic Data Using LASSO. *J Biomed Biotechnol*, 2005(2):147–54, 2005. URL http://dx.doi.org/10.1155/JBB.2005.147.

F. Girosi, M. Jones, and T. Poggio. Regularization Theory and Neural Networks Architectures. *Neural Comput.*, 7(2):219–269, 1995. URL http://citeseer.nj.nec.com/girosi95regularization.html.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286:531–537, 1999. URL http://www.sciencemag.org/cgi/reprint/286/5439/531.pdf.

P. E. Green and V. Srinivasan. Conjoint analysis in consumer research: Issues and outlook. *The Journal of Consumer Research*, 5(2):103–123, 1978. URL http://www.jstor.org/stable/2489001.

J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.

J. Hadamard. *Lectures on Cauchy's Problem: In Linear Partial Differential Equations*. Dover Publications, 1923.

I. Halperin, B. Ma, H. Wolfson, and R. Nussinov. Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins*, 47(4):409–443, Jun 2002. URL `http://dx.doi.org/10.1002/prot.10115`.

Z. Harchaoui and C. Levy-Leduc. Catching change-points with lasso. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 617–624. MIT Press, Cambridge, MA, 2008.

J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28: 100–108, 1979.

T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, UK, 1999.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2001.

D. Haussler. Convolution Kernels on Discrete Structures. Technical Report UCSC-CRL-99-10, UC Santa Cruz, 1999. URL `http://www.cse.ucsc.edu/~haussler/convolutions.ps`.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, 1:49–75, 2000.

D. Heckerman, D. Kadie, and J. Listgarten. Leveraging information across HLA alleles/supertypes improves epitope prediction. *J. Comput. Biol.*, 14(6):736–746, 2007. URL `http://dx.doi.org/10.1089/cmb.2007.R013`.

S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, 89(22):10915–10919, Nov 1992.

T. Hertz and C. Yanover. PepDist: a new framework for protein-peptide binding prediction based on learning peptide distance functions. *BMC Bioinformatics*, 7 Suppl 1:S3, 2006. URL `http://dx.doi.org/10.1186/1471-2105-7-S1-S3`.

T. Hertz and C. Yanover. Identifying hla supertypes by learning distance functions. *Bioinformatics*, 23(2):e148–e155, Jan 2007. URL `http://dx.doi.org/10.1093/Bioinformatics/btl324`.

T. Heskes. Empirical bayes for learning to learn. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 367–374, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2.

S. J. Hill. G-protein-coupled receptors: past, present and future. *Br. J. Pharmacol.*, 147 Suppl 1:S27–S37, Jan 2006. URL `http://dx.doi.org/10.1038/sj.bjp.0706455`.

A. E. Hoerl. Application of ridge regression analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

A. E. Hoerl and R. W. Kennard. Ridge regression : biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

A. E. Hoerl and R. W. Kennard. Citation classic - ridge regression : biased estimation for nonorthogonal problems. *CC/Eng. Tech. Appl. Sci.*, 35:18–18, 1982.

M. C. Honeyman, V. Brusic, N. L. Stone, and L. C. Harrison. Neural network-based prediction of candidate T-cell epitopes. *Nat. Biotechnol.*, 16(10):966–969, Oct 1998. URL `http://dx.doi.org/10.1038/nbt1098-966`.

A. L. Hopkins and C. R. Groom. The druggable genome. *Nat. Rev. Drug Discov.*, 1(9): 727–730, Sep 2002. URL `http://dx.doi.org/10.1038/nrd892`.

F. Horn, E. Bettler, L. Oliveira, F. Campagne, F. E. Cohen, and G. Vriend. GPCRDB information system for G protein-coupled receptors. *Nucl. Acids Res.*, 31(1): 294–297, 2003. URL http://nar.oxfordjournals.org/cgi/content/abstract/31/1/294.

T. Horváth, T. Gärtner, and S. Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167, New York, NY, USA, 2004. ACM Press. URL http://doi.acm.org/10.1145/1014052.1014072.

S. Hua and Z. Sun. Support vector machine approach for protein subcellular localization prediction. *Bioinformatics*, 17(8):721–728, 2001. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/17/8/721.

J. Huang and T. Zhang. The benefit of group sparsity, 2009. URL http://www.citebase.org/abstract?id=oai:arXiv.org:0901.2962.

W. Humphrey, A. Dalke, and K. Schulten. VMD: visual molecular dynamics. *J. Mol. Graph.*, 14(1):33–8, 27–8, Feb 1996.

P. Hupé, N. Stransky, J.-P. Thiery, F. Radvanyi, and E. Barillot. Analysis of array CGH data: from signal ratio to gain and loss of dna regions. *Bioinformatics*, 20(18):3413–3422, Dec 2004. URL http://dx.doi.org/10.1093/bioinformatics/bth418.

International Union of Biochemistry and Molecular Biology. *Enzyme Nomenclature 1992*. Academic Press, San Diego, California, United States, August 1992. ISBN 0122271645. URL http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&amp;path=ASIN/0122271645.

T. Jaakkola, M. Diekhans, and D. Haussler. A Discriminative Framework for Detecting Remote Protein Homologies. *J. Comput. Biol.*, 7(1,2):95–114, 2000. URL http://www.cse.ucsc.edu/research/compbio/discriminative/Jaakola2-1998.ps.

T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In *Adv. Neural Inform. Process. Syst.*, volume 12. MIT Press, Cambridge, MA, 1999.

L. Jacob and J.-P. Vert. Efficient peptide-MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, Feb 2008a. URL http://dx.doi.org/10.1093/bioinformatics/btm611.

L. Jacob and J.-P. Vert. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics*, 24(19):2149–2156, 2008b. URL http://bioinformatics.oxfordjournals.org/cgi/reprint/btn409.

L. Jacob, B. Hoffmann, B. Stoven, and J.-P. Vert. Virtual screening of GPCRs: an *in silico* chemogenomics approach. *BMC Bioinformatics*, 9:363, 2008. URL http://dx.doi.org/10.1186/1471-2105-9-363.

L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems 21*, pages 745–752. MIT Press, 2009a. URL http://books.nips.cc/papers/files/nips21/NIPS2008_0680.pdf.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-516-1. URL http://doi.acm.org/10.1145/1553374.1553431.

R. Jaenisch and A. Bird. Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nat. Genet.*, 33 Suppl:245–254, Mar 2003. URL http://dx.doi.org/10.1038/ng1089.

S. E. Jaroch and H. Weinmann, editors. *Chemical Genomics: Small Molecule Probes to Study Cellular Function*. Ernst Schering Research Foundation Workshop. Springer, Berlin, 2006.

T. Jebara. Multi-task feature and kernel selection for svms. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 55, New York, NY, USA, 2004. ACM. ISBN 1-58113-828-5. URL http://doi.acm.org/10.1145/1015330.1015426.

R. Jenatton, J.-Y. Audibert, and F. Bach. Structured Variable Selection with Sparsity-Inducing Norms. Research report, WILLOW - INRIA Rocquencourt - INRIA - Ecole Normale Supérieure de Paris - Ecole Nationale des Ponts et Chaussées - CNRS : UMR8548 - Imagine - Université Paris-Est, 2009. URL http://hal.inria.fr/inria-00377732/en/.

N. Jojic, M. Reyes-Gomez, D. Heckerman, C. Kadie, and O. Schueler-Furman. Learning MHC I–peptide binding. *Bioinformatics*, 22(14):e227–e235, Jul 2006. URL http://dx.doi.org/10.1093/bioinformatics/btl255.

P. A. Jones. Dna methylation and cancer. *Oncogene*, 21(35):5358–5360, Aug 2002. URL http://dx.doi.org/10.1038/sj.onc.1205597.

K. Jong, E. Marchiori, G. Meijer, A. V. D. Vaart, and B. Ylstra. Breakpoint identification and smoothing of array comparative genomic hybridization data. *Bioinformatics*, 20(18):3636–3637, Dec 2004. URL http://dx.doi.org/10.1093/bioinformatics/bth355.

M. Kanehisa, S. Goto, S. Kawashima, and A. Nakaya. The KEGG databases at GenomeNet. *Nucleic Acids Res.*, 30:42–46, 2002. URL http://nar.oupjournals.org/cgi/content/full/30/1/42.

M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Res.*, 32(Database issue):D277–80, Jan 2004. URL http://dx.doi.org/10.1093/nar/gkh063.

R. Karchin, K. Karplus, and D. Haussler. Classifying G-protein coupled receptors with support vector machines. *Bioinformatics*, 18:147–159, 2002. URL http://bioinformatics.oupjournals.org/cgi/reprint/18/1/147.

H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized Kernels between Labeled Graphs. In T. Faucett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328, New York, NY, USA, 2003. AAAI Press.

H. Kashima, K. Tsuda, and A. Inokuchi. Kernels for graphs. In B. Schölkopf, K. Tsuda, and J. Vert, editors, *Kernel Methods in Computational Biology*, pages 155–170. MIT Press, The MIT Press, Cambridge, Massachussetts, 2004.

E. Kellenberger, J. Rodrigo, P. Muller, and D. Rognan. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. *Proteins*, 57(2):225–242, Nov 2004. URL `http://dx.doi.org/10.1002/prot.20149`.

G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.*, 33:82–95, 1971.

T. Kin, K. Tsuda, and K. Asai. Marginalized kernels for RNA sequence data analysis. In R. Lathtop, K. Nakai, S. Miyano, T. Takagi, and M. Kanehisa, editors, *Genome Informatics 2002*, pages 112–122. Universal Academic Press, 2002. URL `http://www.jsbi.org/journal/GIW02/GIW02F012.html`.

T. Klabunde. Chemogenomics approaches to ligand design. In *Ligand Design for G Protein-coupled Receptors*, chapter 7, pages 115–135. Wiley-VCH, Great Britain, 2006.

T. Klabunde. Chemogenomic approaches to drug discovery: similar receptors bind similar ligands. *Br. J. Pharmacol.*, 152:5–7, May 2007. URL `http://dx.doi.org/10.1038/sj.bjp.0707308`.

K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Ann. Stat.*, 28(5):1356–1378, 2000. URL `http://dx.doi.org/10.1214/aos/1015957397`.

B. K. Kobilka. G protein coupled receptor structure and activation. *Biochim. Biophys. Acta*, 1768(4):794–807, Apr 2007. URL `http://dx.doi.org/10.1016/j.bbamem.2006.10.021`.

N. A. Kratochwil, P. Malherbe, L. Lindemann, M. Ebeling, M. C. Hoener, A. Mühlemann, R. H. P. Porter, M. Stahl, and P. R. Gerber. An automated system for the analysis of G protein-coupled receptor transmembrane binding pockets: alignment, receptor-based pharmacophores, and their application. *J. Chem. Inf. Model.*, 45(5):1324–1336, 2005. URL `http://dx.doi.org/10.1021/ci050221u`.

K. Kristiansen, S. G. Dahl, and O. Edvardsen. A database of mutants and effects of site-directed mutagenesis experiments on G protein-coupled receptors. *Proteins*, 26(1):81–94, Sep 1996. URL `http://dx.doi.org/3.0.CO;2-J`.

R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *Proc IEEE Comput Syst Bioinform Conf*, pages 152–160, 2004.

R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *J. Bioinform. Comput. Biol.*, 3(3):527–550, Jun 2005.

H. Kubinyi, G. Müller, R. Mannhold, and G. Folkers, editors. *Chemogenomics in Drug Discovery: A Medicinal Chemistry Perspective*. Methods and Principles in Medicinal Chemistry. Wiley-VCH, New York, 2004.

P. P. Kuksa, P.-H. Huang, and V. Pavlovic. Scalable algorithms for string kernels with inexact matching. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *NIPS*, pages 881–888. MIT Press, 2008. URL `http://dblp.uni-trier.de/db/conf/nips/nips2008.html#KuksaHP08`.

S. Kumagai. An implicit function theorem: Comment. *Journal of Optimization Theory and Applications*, 31:285–288, Jun 1980.

M. Kumar, M. Bhasin, N. K. Natt, and G. P. S. Raghava. BhairPred: prediction of beta-hairpins in a protein from multiple alignment information using ANN and SVM techniques. *Nucleic Acids Res*, 33(Web Server issue):W154–9, Jul 2005. URL `http://dx.doi.org/doi:10.1093/nar/gki588`.

L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004. ISBN 0471210781.

G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004a. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/20/16/2626.

G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *J. Mach. Learn. Res.*, 5:27–72, 2004b. URL http://www.jmlr.org/papers/v5/lanckriet04a.html.

S. R. Land and J. H. Friedman. Variable fusion: A new adaptive signal regression method. Technical Report 656, Department of Statistics, Carnegie Mellon University Pittsburgh, 1997.

K. G. Le Roch, Y. Zhou, P. L. Blair, M. Grainger, J. K. Moch, J. D. Haynes, P. De la Vega, A. A. Holder, S. Batalov, D. J. Carucci, and E. A. Winzeler. Discovery of Gene Function by Expression Profiling of the Malaria Parasite Life Cycle. *Science*, 301(5639):1503–1508, 2004. URL http://www.sciencemag.org/cgi/content/full/301/5639/1503.

C. Lemarechal, C. Sagastizábal, Echal, C. S. Abal, and P. S. Practical aspects of the moreau-yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7:367–385, 1997.

C. Leng, C. Leng, Y. Lin, Y. Lin, G. Wahba, and G. Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273,1284, 2004.

C. Leslie and R. Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

C. Leslie, E. Eskin, and W. Noble. The spectrum kernel: a string kernel for SVM protein classification. In R. B. Altman, A. K. Dunker, L. Hunter, K. Lauerdale, and T. E. Klein, editors, *Proceedings of the Pacific Symposium on Biocomputing 2002*, pages 564–575, Singapore, 2002. World Scientific.

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,

2004. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/20/4/467.

H. A. Levine. Review of : Solutions of ill posed problems. *Bull. Amer. Math. Soc.*, 1: 521–524, 1979.

L. Liao and W. Noble. Combining Pairwise Sequence Similarity and Support Vector Machines for Detecting Remote Protein Evolutionary and Structural Relationships. *J. Comput. Biol.*, 10(6):857–868, 2003. URL http://www.liebertonline.com/doi/abs/10.1089/106652703322756113.

S. H. S. Lin and O. Civelli. Orphan G protein-coupled receptors: targets for new therapeutic interventions. *Ann. Med.*, 36(3):204–214, 2004. URL http://dx.doi.org/10.1080/07853890310024668.

C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug. Deliv. Rev*, 46(1-3):3–26, Mar 2001.

H. Liu, J. Lafferty, and L. Wasserman. Nonparametric regression and classification with joint sparsity constraints. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 969–976. MIT Press, 2009.

M. S. Lobo, L. I, L. Vandenberghe, H. Lebret, and S. Boyd. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, November 1998.

B. Logan, P. Moreno, B. Suzek, Z. Weng, and S. Kasif. A Study of Remote Homology Detection. Technical Report CRL 2001/05, Compaq Cambridge Research laboratory, June 2001.

K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer. Taking advantage of sparsity in multi-task learning. In *Proceedings of COLT*, 2009.

P. Mahé and J.-P. Vert. Graph kernels based on tree patterns for molecules. Technical Report ccsd-00095488, HAL, September 2006. URL https://hal.ccsd.cnrs.fr/ccsd-00095488.

P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, and J.-P. Vert. Graph kernels for molecular structure-activity relationship analysis with support vector machines. *J. Chem. Inf. Model.*, 45(4):939–51, 2005. URL http://dx.doi.org/10.1021/ci050039t.

P. Mahé, L. Ralaivola, V. Stoven, and J.-P. Vert. The pharmacophore kernel for virtual screening with support vector machines. *J. Chem. Inf. Model.*, 46(5):2003–2014, 2006. URL http://dx.doi.org/10.1021/ci060138m.

S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993. URL http://dx.doi.org/10.1109/78.258082.

C. L. Mallows. Some comments on c_p. *Technometrics*, 15:661–675, 1973. URL http://www.math.tau.ac.il/~yekutiel/MA%20seminar/Malows%202000.pdf.

H. Mamitsuka. Predicting peptides that bind to MHC molecules using supervised learning of hidden Markov models. *Proteins*, 33(4):460–474, Dec 1998.

C. Manly, S. Louise-May, and J. Hammer. The impact of informatics and computational chemistry on synthesis and screening. *Drug Discov. Today*, 6(21):1101–1110, Nov 2001.

E. R. Mardis. Next-generation dna sequencing methods. *Annu. Rev. Genomics Hum. Genet.*, 9:387–402, 2008. URL http://dx.doi.org/10.1146/annurev.genom.9.081307.164359.

H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, March 1952.

S. Martin, D. Roe, and J.-L. Faulon. Predicting protein-protein interactions using signature products. *Bioinformatics*, 21(2):218–226, Jan 2005. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/21/2/218.

Y. C. Martin. A bioavailability score. *J. Med. Chem.*, 48(9):3164–3170, May 2005. URL http://dx.doi.org/10.1021/jm0492002.

A. Matsuda, J.-P. Vert, H. Saigo, N. Ueda, H. Toh, and T. Akutsu. A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Sci.*, 14(11):2804–2813, 2005. URL http://dx.doi.org/10.1110/ps.051597405.

A. McMichael and T. Hanke. The quest for an AIDS vaccine: is the CD8+ T-cell approach feasible? *Nat. Rev. Immunol.*, 2(4):283–291, Apr 2002. URL http://dx.doi.org/10.1038/nri779.

L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *J. R. Stat. Soc. Ser. B*, 70(1):53–71, 2008. URL http://dx.doi.org/10.1111/j.1467-9868.2007.00627.x.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Ann. Stat.*, 34:1436–1462, 2006. URL http://dx.doi.org/10.1214/009053606000000281.

N. Meinshausen, G. Rocha, and B. Yu. Discussion: A tale of three cousins: Lasso, l2boosting and dantzig. *ANNALS OF STATISTICS*, 35:2373, 2007. URL http://doi:10.1214/009053607000000460.

N. Meinshausen and P. Buehlmann. Stability selection, May 2009. URL http://arxiv.org/abs/0809.2932.

M. Milik, D. Sauer, A. P. Brunmark, L. Yuan, A. Vitiello, M. R. Jackson, P. A. Peterson, J. Skolnick, and C. A. Glass. Application of an artificial neural network to predict specific class I MHC binding peptide sequences. *Nat. Biotechnol.*, 16(8):753–756, Aug 1998. URL http://dx.doi.org/10.1038/nbt0898-753.

F. Mordelet and J.-P. Vert. Sirene: Supervised inference of regulatory networks. *Bioinformatics*, 24(16):i76–i82, 2008. URL http://dx.doi.org/10.1093/bioinformatics/btn273.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, 5(7):621–628, Jul 2008. URL `http://dx.doi.org/10.1038/nmeth.1226`.

S. Mukherjee, P. Tamayo, J. P. Mesirov, D. Slonim, A. Verri, and T. Poggio. Support vector machine classification of microarray data. Technical Report 182, C.B.L.C., 1998. URL `http://citeseer.nj.nec.com/437379.html`. A.I. Memo 1677.

D. Mustafi and K. Palczewski. Topology of class a g protein-coupled receptors: insights gained from crystal structures of rhodopsins, adrenergic and adenosine receptors. *Mol Pharmacol*, 75(1):1–12, Jan 2009. URL `http://dx.doi.org/10.1124/mol.108.051938`.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995. ISSN 0097-5397. URL `http://dx.doi.org/10.1137/S0097539792240406`.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

M. Nielsen, C. Lundegaard, P. Worning, S. L. Lauemøller, K. Lamberth, S. Buus, S. Brunak, and O. Lund. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci.*, 12(5):1007–1017, May 2003.

G. Obozinski, M. J. Wainwright, and M. I. Jordan. Union support recovery in high-dimensional multivariate regression. Technical Report 0808.0711v1, arXiv, August 2008.

G. Obozinski, B. Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 2009. To appear.

Y. Okuno, J. Yang, K. Taneishi, H. Yabuuchi, and G. Tsujimoto. GLIDA: GPCR-ligand database for chemical genomic drug discovery. *Nucleic Acids Res.*, 34(Database issue): D673–D677, Jan 2006. URL `http://dx.doi.org/10.1093/nar/gkj028`.

M. R. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20:389–404, 1999a.

M. R. Osborne, B. Presnell, and B. A. Turlach. On the lasso and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337, 1999b.

S. J. Pan and Q. Yang. A survey on transfer learning. Technical Report HKUST-CS08-08, Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China, November 2008. URL `http://www.cse.ust.hk/~sinnopan/publications/TLsurvey_0822.pdf`.

P. Parham. *The Immune System*. Garland Science Publishing, 2004.

K.-J. Park and M. Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19 (13):1656–1663, 2003. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/19/13/1656`.

K. C. Parker, M. A. Bednarek, and J. E. Coligan. Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *J. Immunol.*, 152(1):163–175, Jan 1994.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2(6):559–572, 1901.

C. M. Perou, T. Sørlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, O. Fluge, A. Pergamenschikov, C. Williams, S. X. Zhu, P. E. Lønning, A. L. Børresen-Dale, P. O. Brown, and D. Botstein. Molecular portraits of human breast tumours. *Nature*, 406(6797):747–752, Aug 2000. URL `http://dx.doi.org/10.1038/35021093`.

B. Peters and A. Sette. Generating quantitative models describing the sequence specificity of biological processes with the stabilized matrix method. *BMC Bioinformatics*, 6:132, 2005. URL `http://dx.doi.org/10.1186/1471-2105-6-132`.

B. Peters, H.-H. Bui, S. Frankild, M. Nielson, C. Lundegaard, E. Kostem, D. Basch, K. Lamberth, M. Harndahl, W. Fleri, S. S. Wilson, J. Sidney, O. Lund, S. Buus, and A. Sette. A community resource benchmarking predictions of peptide binding to MHC-I molecules. *PLoS Comput. Biol.*, 2(6):e65, Jun 2006. URL `http://dx.doi.org/10.1371/journal.pcbi.0020065`.

N. Pfeifer and O. Kohlbacher. Multiple instance learning allows mhc class ii epitope predictions across alleles. In *WABI '08: Proceedings of the 8th international workshop on Algorithms in Bioinformatics*, pages 210–221, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87360-0. URL `http://dx.doi.org/10.1007/978-3-540-87361-7_18`.

D. L. Philips. A technique for the numerical solution of certain integral equations of the first kind. *J. Assoc. Comput. Mach.*, 9:84–97, 1962.

D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W. L. Kuo, C. Chen, Y. Zhai, S. H. Dairkee, B. M. Ljung, J. W. Gray, and D. G. Albertson. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.*, 20(2):207–211, Oct 1998. URL `http://dx.doi.org/10.1038/2524`.

J. Qian, J. Lin, N. M. Luscombe, H. Yu, and M. Gerstein. Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data. *Bioinformatics*, 19(15):1917–1926, 2003. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/19/15/1917`.

J. Qiu, J. Hue, A. Ben-Hur, J.-P. Vert, and W. S. Noble. A structural alignment kernel for protein structures. *Bioinformatics*, 23(9):1090–1098, May 2007. URL `http://dx.doi.org/10.1093/bioinformatics/btl642`.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *J. Mach. Learn. Res.*, 9:2491–2521, 2008.

A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on*

Machine learning, pages 775–782, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. URL `http://doi.acm.org/10.1145/1273496.1273594`.

L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical informatics. *Neural Netw.*, 18(8):1093–1110, Sep 2005. URL `http://dx.doi.org/10.1016/j.neunet.2005.07.009`.

H. Rammensee, J. Bachmann, N. P. Emmerich, O. A. Bachor, and S. Stevanović. Syfpeithi: database for MHC ligands and peptide motifs. *Immunogenetics*, 50(3-4):213–219, Nov 1999.

H. G. Rammensee, T. Friede, and S. Stevanoviíc. MHC ligands and peptide motifs: first listing. *Immunogenetics*, 41(4):178–228, 1995.

J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In T. Washio and L. De Raedt, editors, *Proceedings of the First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

H. Rangwala and G. Karypis. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247, Dec 2005. URL `http://dx.doi.org/10.1093/bioinformatics/bti687`.

F. Rapaport, A. Zynoviev, M. Dutreix, E. Barillot, and J.-P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007.

F. Rapaport, E. Barillot, and J.-P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, Jul 2008. URL `http://dx.doi.org/10.1093/bioinformatics/btn188`.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, December 2005. ISBN 026218253X.

P. Ravikumar, H. Liu, J. Lafferty, and L. Wasserman. Spam: Sparse additive models. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1201–1208. MIT Press, Cambridge, MA, 2008.

P. Ravikumar, G. Raskutti, M. Wainwright, and B. Yu. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized mle. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1329–1336. MIT Press, 2009.

P. A. Reche, J.-P. Glutting, and E. L. Reinherz. Prediction of MHC class I binding peptides using profile motifs. *Hum. Immunol.*, 63(9):701–709, Sep 2002.

A. Rinaldo. Properties and refinements of the fused lasso. *Annals of Statistics*, 37(5B): 2922–2952, 2009. URL `http://doi:10.1214/08-AOS665`.

J. Robinson, A. Malik, P. Parham, J. G. Bodmer, and S. G. Marsh. IMGT/HLA database–a sequence database for the human major histocompatibility complex. *Tissue Antigens*, 55 (3):280–287, Mar 2000.

D. Rognan. Chemogenomic approaches to rational drug design. *Br. J. Pharmacol.*, 152: 38–52, May 2007. URL `http://dx.doi.org/10.1038/sj.bjp.0707307`.

F. J. Rohlf. J. felsenstein, inferring phylogenies, sinauer assoc., 2004, pp. xx + 664. *J. Classif.*, 22(1):139–142, 2005. ISSN 0176-4268. URL `http://dx.doi.org/10.1007/s00357-005-0009-4`.

C. Rolland, R. Gozalbes, A. Nicolaï, M.-F. Paugam, L. Coussy, F. Barbosa, D. Horvath, and F. Revah. G-protein-coupled receptor affinity prediction based on the use of a profiling dataset: Qsar design, synthesis, and experimental validation. *J. Med. Chem.*, 48(21): 6563–6574, Oct 2005. URL `http://dx.doi.org/10.1021/jm0500673`.

R. Rosenfeld, Q. Zheng, S. Vajda, and C. DeLisi. Flexible docking of peptides to class I major-histocompatibility-complex receptors. *Genet. Anal.*, 12(1):1–21, Mar 1995.

R. Rosipal and L. J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *J. Mach. Learn. Res.*, 2:97–123, 2001.

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, 35 (3):1012–1030, 2007.

F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding dna regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation. *Nat. Biotechnol.*, 16(10):939–945, October 1998. ISSN 1087-0156. URL http://dx.doi.org/10.1038/nbt1098-939.

V. Roth. The generalized lasso: a wrapper approach to gene selection for microarray data. In *Proc. CADE-14, 252–255*, 2002.

V. Roth and B. Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 848–855, 2008.

O. Rötzschke, K. Falk, S. Stevanović, G. Jung, and H. C. Rammensee. Peptide motifs of closely related HLA class I molecules encompass substantial differences. *Eur. J. Immunol.*, 22(9):2453–2456, Sep 1992.

R. B. Russell and G. J. Barton. Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, 14(2):309–323, Oct 1992. URL http://dx.doi.org/10.1002/prot.340140216.

H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu. Protein homology detection using string alignment kernels. *Bioinformatics*, 20(11):1682–1689, 2004. URL http://bioinformatics.oupjournals.org/cgi/content/abstract/20/11/1682.

S. Saitoh. *Theory of reproducing Kernels and its applications*. Longman Scientific & Technical, Harlow, UK, 1988.

J. Salomon and D. R. Flower. Predicting Class II MHC-Peptide binding: a kernel based approach using similarity scores. *BMC Bioinformatics*, 7:501, 2006. URL http://dx.doi.org/10.1186/1471-2105-7-501.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, 2002. URL http://www.learning-with-kernels.org.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 327–352. MIT Press, 1999. URL http://www.kyb.tuebingen.mpg.de/bu/people/bs/papers/kpca_nc.ps.gz.

B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, volume 2011 of *Lecture Notes in Computer Science*, pages 416–426, Berlin / Heidelberg, 2001a. Springer. URL http://dx.doi.org/10.1007/3-540-44581-1.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.*, 13:1443–1471, 2001b.

B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, The MIT Press, Cambridge, Massachussetts, 2004.

O. Schueler-Furman, Y. Altuvia, A. Sette, and H. Margalit. Structure-based prediction of binding peptides to MHC class I molecules: application to a broad range of MHC alleles. *Protein Sci.*, 9(9):1838–1846, Sep 2000.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

M. Seeger. Covariance Kernels from Bayesian Generative Models. In *Adv. Neural Inform. Process. Syst.*, volume 14, pages 905–912, 2002.

M. Seeger. Gaussian processes for machine learning. *Int J Neural Syst*, 14(2):69–106, Apr 2004.

A. Sette and J. Sidney. HLA supertypes and supermotifs: a functional perspective on HLA polymorphism. *Curr. Opin. Immunol.*, 10(4):478–482, Aug 1998.

A. Sette and J. Sidney. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. *Immunogenetics*, 50(3-4):201–212, Nov 1999.

A. Sette, R. Chesnut, and J. Fikes. HLA expression in cancer: implications for T cell-based immunotherapy. *Immunogenetics*, 53(4):255–263, 2001.

S. Shacham, Y. Marantz, S. Bar-Haim, O. Kalid, D. Warshaviak, N. Avisar, B. Inbal, A. Heifetz, M. Fichman, M. Topf, Z. Naor, S. Noiman, and O. M. Becker. PREDICT modeling and in-silico screening for G-protein coupled receptors. *Proteins*, 57(1):51–86, Oct 2004. URL `http://dx.doi.org/10.1002/prot.20195`.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, October 2000. ISSN 03783758. URL `http://dx.doi.org/10.1016/S0378-3758(00)00115-4`.

J. Sidney, M. F. del Guercio, S. Southwood, V. H. Engelhard, E. Appella, H. G. Rammensee, K. Falk, O. Rötzschke, M. Takiguchi, and R. T. Kubo. Several HLA alleles share overlapping peptide specificities. *J. Immunol.*, 154(1):247–259, Jan 1995.

J. Sidney, H. M. Grey, S. Southwood, E. Celis, P. A. Wentworth, M. F. del Guercio, R. T. Kubo, R. W. Chesnut, and A. Sette. Definition of an HLA-A3-like supermotif demonstrates the overlapping peptide-binding repertoires of common HLA molecules. *Hum Immunol*, 45(2):79–93, Feb 1996.

T. Smith and M. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *J. Mach. Learn. Res.*, 7:1531–1565, 2006. ISSN 1533-7928.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In T. Fawcett and N. Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Adv. Neural. Inform. Process Syst. 17*, pages 1329–1336, Cambridge, MA, 2005. MIT Press.

N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *COLT*, pages 545–560, 2005.

V. K. Srivastava and T. D. Dwivedi. Estimation of seemingly unrelated regression equations : A brief survey. *Journal of Econometrics*, 10(1):15–32, April 1979. URL http://ideas.repec.org/a/eee/econom/v10y1979i1p15-32.html.

A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, 102(43):15545–15550, Oct 2005. URL http://dx.doi.org/10.1073/pnas.0506580102.

J.-S. Surgand, J. Rodrigo, E. Kellenberger, and D. Rognan. A chemogenomic analysis of the transmembrane binding cavity of human g-protein-coupled receptors. *Proteins*, 62 (2):509–538, Feb 2006. URL http://dx.doi.org/10.1002/prot.20768.

S. J. Swamidass, J. Chen, J. Bruand, P. Phung, L. Ralaivola, and P. Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity and anti-cancer activity. *Bioinformatics*, 21(Suppl. 1):i359–i368, Jun 2005. URL http://dx.doi.org/10.1093/bioinformatics/bti1055.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite kernel learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, Helsinki Finlande, 07 2008a. URL http://hal.archives-ouvertes.fr/hal-00316016/en/.

M. Szafranski, Y. Grandvalet, and P. Morizet-Mahoudeaux. Hierarchical penalization. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1457–1464. MIT Press, Cambridge, MA, 2008b.

B. Taskar, C. Guestrin, and D. Koller. Max-Margin Markov Networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

S. Thrun and L. Pratt, editors. *Learning to learn*. Kluwer Academic Publishers, Norwell, MA, USA, 1998. ISBN 0-7923-8047-9.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc. B.*, 58(1):267–288, 1996.

R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics (Oxford, England)*, 9(1):18–29, January 2008. ISSN 1465-4644. URL `http://dx.doi.org/10.1093/biostatistics/kxm013`.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(1):91–108, 2005. URL `http://ideas.repec.org/a/bla/jorssb/v67y2005i1p91-108.html`.

A. Tikhonov. On the stability of inverse problems. *Doklady Akademii nauk SSSR*, 39(5): 195–198, 1943.

A. Tikhonov. Solution of incorrectly problems and the regularization method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. W.H. Winston, Washington, D.C., 1977.

R. Todeschini and V. Consonni. *Handbook of Molecular Descriptors*. Wiley-VCH, New York, 2002.

J. C. Tong, G. L. Zhang, T. W. Tan, J. T. August, V. Brusic, and S. Ranganathan. Prediction of HLA-DQ3.2beta ligands: evidence of multiple registers in class II binding peptides. *Bioinformatics*, 22(10):1232–1238, May 2006. URL `http://dx.doi.org/10.1093/bioinformatics/btl071`.

S. Topiol and M. Sabio. X-ray structure breakthroughs in the GPCR transmembrane region. *Biochem Pharmacol*, 78(1):11–20, Jul 2009. URL `http://dx.doi.org/10.1016/j.bcp.2009.02.012`.

J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inform. Theory*, 50:2231–2242, 2004.

J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation: part i: Greedy pursuit. *Signal Process.*, 86(3):572–588, 2006. ISSN 0165-1684. URL `http://dx.doi.org/10.1016/j.sigpro.2005.05.030`.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6:1453–1484, 2005. URL `http://jmlr.csail.mit.edu/papers/v6/tsochantaridis05a.html`.

K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller. A new discriminative kernel from probabilistic models. *Neural Computation*, 14(10):2397–2414, 2002a. URL `http://dx.doi.org/10.1162/08997660260293274`.

K. Tsuda, T. Kin, and K. Asai. Marginalized Kernels for Biological Sequences. *Bioinformatics*, 18:S268–S275, 2002b.

B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

E. van Beers and P. Nederlof. Array-CGH and breast cancer. *Breast Cancer Research*, 8(3): 210, 2006. ISSN 1465-5411. URL `http://breast-cancer-research.com/content/8/3/210`.

M. J. van de Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. M. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.*, 347(25):1999–2009, Dec 2002. URL `http://dx.doi.org/10.1056/NEJMoa021967`.

L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven,

C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancers. *Nature*, 415(6871):530–536, Jan 2002. URL `http://dx.doi.org/10.1038/415530a`.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New-York, 1998.

V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0387945598. URL `http://portal.acm.org/citation.cfm?id=211359`.

V. Vapnik and A. Y. Chervonenkis. Teoriya raspoznavaniya obrazov: Statisticheskie problemy obucheniya. (Russian) [Theory of Pattern Recognition: Statistical Problems of Learning]. Moscow: Nauka, 1974.

D. F. Veber, S. R. Johnson, H.-Y. Cheng, B. R. Smith, K. W. Ward, and K. D. Kopple. Molecular properties that influence the oral bioavailability of drug candidates. *J. Med. Chem.*, 45(12):2615–2623, Jun 2002.

J.-P. Vert. A tree kernel to analyze phylogenetic profiles. *Bioinformatics*, 18: S276–S284, 2002. URL `http://cbio.ensmp.fr/~jvert/publi/ismb02/index.html`.

J.-P. Vert. The optimal assignment kernel is not positive definite. Technical Report 0801.4061, Arxiv, 2008. URL `http://hal.archives-ouvertes.fr/hal-00218278`.

J.-P. Vert and L. Jacob. Machine learning for in silico virtual screening and chemical genomics: New strategies. *Combinatorial Chemistry & High Throughput Screening*, 11 (8):677–685, September 2008. ISSN 1386-2073. URL `http://dx.doi.org/10.2174/138620708785739899`.

J.-P. Vert and Y. Yamanishi. Supervised graph inference. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Adv. Neural Inform. Process. Syst.*, volume 17, pages 1433–1440. MIT Press, Cambridge, MA, 2005.

J.-P. Vert, H. Saigo, and T. Akutsu. Local alignment kernels for biological sequences. In B. Schölkopf, K. Tsuda, and J. Vert, editors, *Kernel Methods in Computational Biology*, pages 131–154. MIT Press, The MIT Press, Cambridge, Massachussetts, 2004.

J.-P. Vert, R. Thurman, and W. S. Noble. Kernels for gene regulatory regions. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Adv. Neural. Inform. Process Syst.*, volume 18, pages 1401–1408, Cambridge, MA, 2006. MIT Press.

J.-P. Vert, J. Qiu, and W. S. Noble. A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, 8 Suppl 10:S8, 2007. URL `http://dx.doi.org/10.1186/1471-2105-8-S10-S8`.

S. V. N. Vishwanathan and A. J. Smola. Fast kernels for string and tree matching. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel methods in computational biology*, pages 113–130. MIT Press, 2004.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007. URL `http://dx.doi.org/10.1007/s11222-007-9033-z`.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1990.

M. J. Wainwright. Sharp thresholds for high-dimensional and noisy recovery of sparsity. Technical Report 709, UC Berkeley, Department of Statistics, 2006. URL `http://www.stat.berkeley.edu/tech-reports/709.pdf`.

M. Wang, J. Yang, G.-P. Liu, Z.-J. Xu, and K.-C. Chou. Weighted-support vector machines for predicting membrane protein types based on pseudo-amino acid composition. *Protein Eng. Des. Sel.*, 17(6):509–516, 2004. URL `http://dx.doi.org/10.1093/protein/gzh061`.

R. F. Wang. Human tumor antigens: implications for cancer vaccine development. *J. Mol. Med.*, 77(9):640–655, Sep 1999.

Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, 10(1):57–63, Jan 2009. URL `http://dx.doi.org/10.1038/nrg2484`.

C. Watkins. Dynamic alignment kernels. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 39–50. MIT Press, Cambridge, MA, 2000. URL `http://www.cs.rhbnc.ac.uk/home/chrisw/dynk.ps.gz`.

N. Weill and D. Rognan. Development and Validation of a Novel Protein– Ligand Fingerprint To Mine Chemogenomic Space: Application to G Protein-Coupled Receptors and Their Ligands. *Journal of Chemical Information and Modeling*, 49(4):1049–1062, 2009.

W. I. Weis and B. K. Kobilka. Structural insights into G-protein-coupled receptor activation. *Curr Opin Struct Biol*, 18(6):734–740, Dec 2008. URL `http://dx.doi.org/10.1016/j.sbi.2008.09.010`.

M. R. Wilkins, C. Pasquali, R. D. Appel, K. Ou, O. Golaz, J. C. Sanchez, J. X. Yan, A. A. Gooley, G. Hughes, I. Humphery-Smith, K. L. Williams, and D. F. Hochstrasser. From proteins to proteomes: large scale protein identification by two-dimensional electrophoresis and amino acid analysis. *Biotechnology (N Y)*, 14(1):61–65, Jan 1996.

C. Williams. Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond. In M. Jordan, editor, *Learning and Inference in Graphical Models*. Kluwer Academic Press, 1998.

E. Xing and R. Karp. Motifprototyper: A bayesian profile model for motif families. *PNAS*, 101(29):10523–10528, 2004.

E. P. Xing, W. Wu, M. I. Jordan, and R. M. Karp. LOGOS: A modular Bayesian model for de novo motif detection. *J. Bioinform. Comput. Biol.*, 2:127–154, 2004. URL `http://dx.doi.org/10.1142/S0219720004000508`.

Y. Xue, D. Dunson, and L. Carin. The matrix stick-breaking process for flexible multi-task learning. In *ICML '07: Proceedings of the 24th international conference on Machine* *learning*, pages 1063–1070, New York, NY, USA, June 2007a. ACM. ISBN 978-1-59593-793-3. URL `http://doi.acm.org/10.1145/1273496.1273630`.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8:2007, January 2007b.

Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics*, 21:i468–i477, 2005. URL `http://dx.doi.org/10.1093/bioinformatics/bti1012`.

Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, 30(4), February 2002. ISSN 1362-4962. URL `http://dx.doi.org/10.1093/nar/30.4.e15`.

X. Yao, C. Parnot, X. Deupi, V. R. P. Ratnala, G. Swaminath, D. Farrens, and B. Kobilka. Coupling ligand structure to specific conformational switches in the beta2-adrenoceptor. *Nat. Chem. Biol.*, 2(8):417–422, Aug 2006. URL `http://dx.doi.org/10.1038/nchembio801`.

J. W. Yewdell and J. R. Bennink. Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. *Annu. Rev. Immunol.*, 17:51–88, 1999. URL `http://dx.doi.org/10.1146/annurev.immunol.17.1.51`.

K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 1012–1019, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. URL `http://doi.acm.org/10.1145/1102351.1102479`.

S. Yu, V. Tresp, and K. Yu. Robust multi-task learning with t-processes. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1103–1110, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. URL `http://doi.acm.org/10.1145/1273496.1273635`.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, 68(1):49–67, 2006.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007a. URL `http://ideas.repec.org/a/oup/biomet/v94y2007i1p19-35.html`.

M. Yuan and Y. Lin. On the non-negative garrotte estimator. *Journal Of The Royal Statistical Society Series B*, 69(2):143–161, 2007b. URL `http://ideas.repec.org/a/bla/jorssb/v69y2007i2p143-161.html`.

A. Zellner. An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *J. Am. Stat. Assoc.*, 57(298):348–368, 1962. URL `http://dx.doi.org/10.2307/2281644`.

G. L. Zhang, A. M. Khan, K. N. Srinivasan, J. T. August, and V. Brusic. MULTIPRED: a computational system for prediction of promiscuous HLA binding peptides. *Nucleic Acids Res/*, 33(Web Server issue):W172–W179, Jul 2005. URL `http://dx.doi.org/10.1093/nar/gki452`.

H. H. Zhang, Y. Liu, Y. Wu, and J. Zhu. Variable selection for multicategory SVM via adaptive sup-norm regularization. *Electronic Journal of Statistics*, 2:149–167, 2008. URL `http://dx.doi.org/10.1214/08-EJS122`.

S.-W. Zhang, Q. Pan, H.-C. Zhang, Y.-L. Zhang, and H.-Y. Wang. Classification of protein quaternary structure with support vector machine. *Bioinformatics*, 19(18):2390–2396, 2003. URL `http://bioinformatics.oupjournals.org/cgi/content/abstract/19/18/2390`.

Y. Zhang. Progress and challenges in protein structure prediction. *Curr. Opin. Struct. Biol.*, 18(3):342–348, June 2008. URL `http://dx.doi.org/10.1016/j.sbi.2008.02.004`.

P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541, 2006. URL `http://jmlr.csail.mit.edu/papers/v7/zhao06a.html`.

P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Ann. Stat.*, 37(6A):3468–3497, 2009.

Y. Zhao, C. Pinilla, D. Valmori, R. Martin, and R. Simon. Application of support vector machines for T-cell epitopes prediction. *Bioinformatics*, 19(15):1978–1984, Oct 2003.

S. Zhu, K. Udaka, J. Sidney, A. Sette, K. F. Aoki-Kinoshita, and H. Mamitsuka. Improving MHC binding peptide prediction by incorporating binding data of auxiliary MHC molecules. *Bioinformatics*, 22(13):1648–1655, 2006.

H. Zou. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.*, 101:1418–1429, December 2006. URL `http://ideas.repec.org/a/bes/jnlasa/v101y2006p1418-1429.html`.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.1596`.