Transcription Networks, Microarray Chips and Sparse Linear Methods

José Miguel Hernández Lobato

Abstract

This paper is an introduction to some basic concepts on transcription networks, microarray chips and sparse linear methods. We describe the application of sparse linear methods to the modeling of gene expression data. First, we review the gene expression process and how transcription networks regulate protein production rates in cells. After this, DNA microarray chips are described alongside with the specific features of the gene expresion data generated by this technology. Finally, we also introduce different sparse linear methods that are frequently used in the modeling of gene expression data.

1 Introduction

Cells are very complex devices composed of several thousands of interacting proteins, where each of these proteins is a biological molecule that acomplishes a specific task with very high precision. In particular, proteins do almost everying in a cell, from the catalyzation of chemical reactions or the maintainance of cell shape and size to the transmission and processing of information or the binding of small target molecules [1]. As an example, *Saccharomyces cerevisiae* or budding yeast is one of the most thoroughly researched microorganisms and its individual cells typically contain a few billion proteins of about 6,600 different types [2].

Throughout the lifetime of a cell, different proteins have to be produced in different situations with production rates that can also vary with time. For example, in another well studied microorganism called *Escherichia coli*, the absence of glucose and the presence of lactose triggers the production of several proteins that are required for the transport and the metabolism of lactose [2]. When the DNA molecules of a cell undergo structural damage, DNA repair proteins are produced by the cell to correct the alterations in the DNA molecules [3]. Thus, a cell constantly checks for changes in its environment and automatically fixes an adequate production rate for each protein. The process by which a protein is synthesized from the information stored in the DNA is called *gene expression*. Additionally, the main mechanism of control of the amount of proteins generated during the gene expression process is called *transcription regulation* and it is carried out by *transcription control networks* [4, 2].

DNA *microarray chips* [5] are a recent high-throughput molecular techology that allows researchers to simultaneously monitor the expression profile of thousands of genes in a single sample of cells or tissues. Microarray datasets often present three common characteristics. First, they only include a limited number n of data instances. Second, noise in the expression measurements is frequently very high and finally, microarray datasets have a large dimensionality d, where $d \gg n$. These features provide a significant challenge to the analysis of gene expression data. In particular, most machine learning methods are likely to identify irrelevant patterns in such a 'large d, small n' scenario [6]. To solve this problem, a popular approach is to consider an underlying *sparse linear model* whose input is the activity of a few relevant genes. Given a microarray dataset, the coefficients of this linear model are learned using a *sparse linear method* [7, 8, 9].

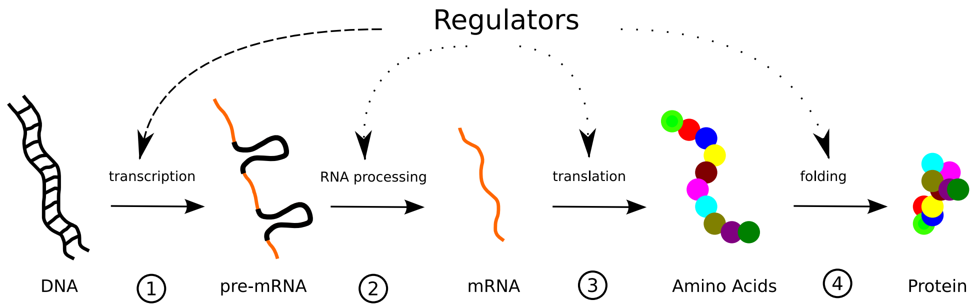

Figure 1: The four steps of the gene expression mechanism. First RNA polymerase transcribes the information coded in a gene to a pre-mRNA molecule. Second, the pre-mRNA molecule is modified, resulting in an mRNA molecule. Third, the mRNA molecule is mapped to a chain of amino acids. Finally, the polypeptide chain is folded into a 3d protein structure. Regulation of protein concentration may occur at any of these steps. However, transcriptional control is the most common regulatory mechanism.

2 Gene Expression

Gene expression is the process by which proteins are synthesized in a cell. A gene is a sequence of nucleotides in a DNA molecule with the instructions for the production of a particular protein [10]. Genes are generally preceeded in the DNA by a regulatory region called the promoter. The enzime RNA polymerase binds this promoter and synthesizes a pre-mRNA molecule or primary transcript that is a complementary copy of the gene. This first step of the gene expression process is called transcription. In a second step called RNA processing, the pre-mRNA molecule undergoes some modifications to become an mRNA molecule. After this, a step called translation generates from the mRNA molecule a corresponding polypeptide chain. The requiered mapping between RNA tri-nucleotide sequences and amino acids is known as the genetic code [1]. Finally, the polypeptide chain is folded into a three-dimensional structure, becoming a biochemically active protein [4]. Figure 1 displays these four steps of the gene expression mechanism: transcription, RNA processing, translation and folding.

Regulatory molecules can control the final concentration of each protein. For this, they modify the amount of intermediate product produced at each step of the gene expression process. Regulators are generally fully-assembled proteins but other intermediate products such as RNA molecules or polypeptides can also have a control role in the gene expression process [10, 4]. Although regulation can take place at any step, transcriptional control is the most important mechanism for regulating gene expression. In particular, a control step at the begining of the gene expression process allows the cell to save energy by preventing the production of unnecessary RNA molecules.

3 Transcription Regulation

The rate of synthesis of pre-mRNA molecules is determined by the activity of the enzime RNA polymerase (RNAP). After binding a specific DNA sequence in the promoter, RNAP opens the DNA double helix and starts copying the genetic material into a pre-mRNA molecule. Transcription ends when RNAP finds a stop sequence in the DNA. This process is illustrated in the top of Figure 2. The activity of RNAP can be altered by the action of different regulatory molecules. These regulators may modify (i) the affinity of RNAP to bind the promoter, (ii) the accessibility of RNAP to the DNA, (iii) the rate at which RNAP synthesizes the pre-mRNA molecule and (iv) the location in the DNA where RNAP terminates the copying process [10]. From these four transcriptional control mechanisms, the first one appears to be the most predominant in nature [4]. The reason for this is that this form of regulation is very configurable for specific genes.

The affinity of RNAP for binding a particular promoter can be modified by a series of regulatory proteins called *transcription factors*. These are DNA binding proteins that are able to recognize and bind specific DNA sequences called *motifs*. When a transcription factor (TF) binds a motif in the

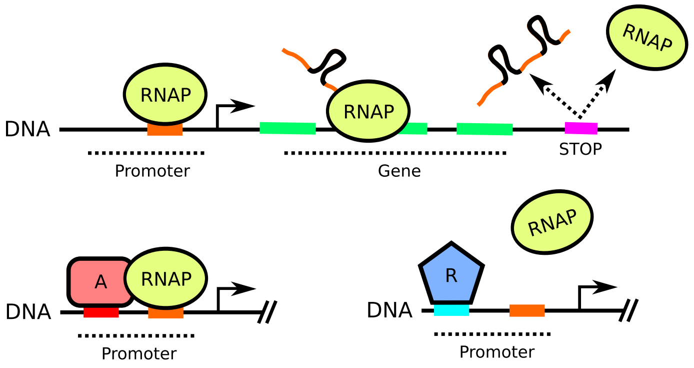

Figure 2: Transcription begins when RNAP binds a specific DNA sequence in the promoter (top). After this, RNAP copies the genetic material into a pre-mRNA molecule. Transcription ends when a particular stop sequence is recognized by RNAP. Activator proteins may bind a specific nucleotide sequence in the promoter, increasing the binding affinity of RNAP (bottom left). Repressor proteins may bind a specific nucleotide sequence in the promoter, reducing the binding affinity of RNAP (bottom right).

promoter of a target gene, it changes the probability per unit time that RNAP binds the promoter and generates a pre-mRNA molecule [2]. In particular, if the TF is an activator it can recruit RNAP to the promoter of the target gene and increase the probability of transcription initiation. By contrast, if the TF is a repressor it can prevent RNAP from binding the promoter of the target gene and the probability of transcription initiation is reduced. The bottom of Figure 2 illustrates this behavior. The molecular form of a TF can shift rapidly from active to inactive states and this depends on specific environmental signals [2]. An active TF is very likely to bind the promoter of a target gene, while the DNA binding properties of an inactive TF are significantly reduced. TFs are proteins and are also encoded in the DNA by genes. This means that the expression level of a particular TF can also be controlled by other TFs and these may additionally be regulated by yet other TFs and so on. These regulatory interactions at the transcription level between different genes and different gene products can be organized in a transcription control network.

4 Transcription Control Networks

A transcription control network (TCN) describes the transcriptional regulatory interactions between the genes of a cell [4, 2, 13]. Each node in the network corresponds to a specific gene and each edge represents a transcriptional regulatory relationship between the two connected nodes. In particular, the directed edge X $\rightarrow$ Y indicates that the protein product of gene X, typically a TF, has a direct effect on the transcription rate of gene Y. TCNs have a scale free topological structure [14, 15, 13]. This means that most genes are involved in only a reduced number of interactions, while a few hubs or key regulators are linked to a significantly higher number of genes. Figure 4 illustrates this by displaying an approximation of the TCN for budding yeast, where the directionality of the edges has been removed to improve network visualization [11]. In this TCN, the average connectivity of a few hub genes (diamon-shaped nodes) is very high: 21.75, while the average network connectivity is much lower: 3.77. Highly connected nodes or hubs are the result of a heavy tail in the connectivity distribution of scale-free networks. In particular, the probability that a randomly selected node in a scale-free network has exactly k links follows a power law $P(k) \sim k^{-\gamma}$, where γ is the scaling exponent and $\sim$ indicates 'proportional to' [14]. The left part of Figure 4 shows a histrogram for the connectivity in the TCN of budding yeast. The heavy tail in this histogram and its decreasing shape suggest a power law for the connectivity distribution. Further evidence is given by the right part of Figure 4, which shows a log-log plot of the empirical connectivity probabilities. The points in this plot are well approximated by a straight line. This is a clear signature of an underlying power law process. Another key feature of scale-free networks, and in particular TCNs, is their robustness [14]. In a scale-free network, random failure of a node generally affects the many nodes with low connectivity, which are not essential for maintaining the network integrity. However, an intentional attack to a few of the hub nodes or key connectors can significantly damage the network.

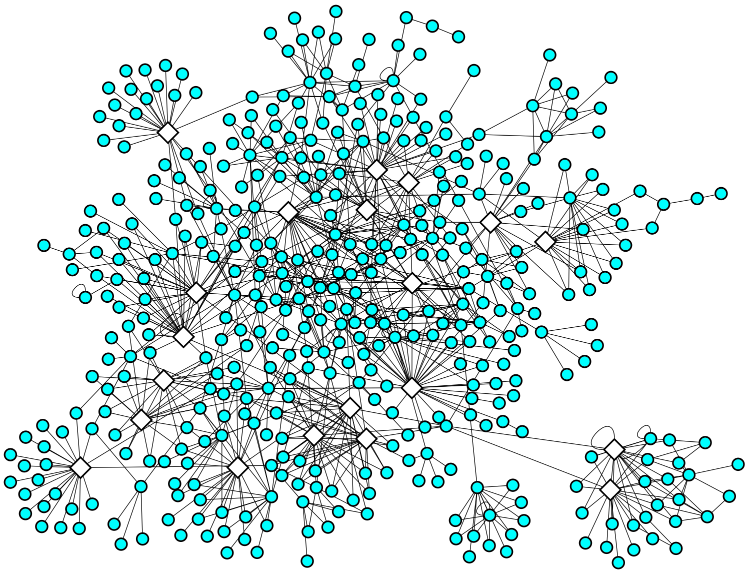

Figure 3: Main connected component of an approximation for the transcription control network of budding yeast. This network is described in [11]. Each node represents a different gene and each link represents a transcriptional regulatory interaction between two genes. White diamond-shaped nodes correspond to the 20 genes with highest connectivity. The average connectivity of these hub genes is very high: 21.75. By contrast, the average connectivity of a gene that is selected randomly in the network is much lower: 3.77. The network was visualized using the software Cytoscape [12].

4.1 Modeling the Edges in a TCN

As mentioned before, a link in a TCN represents an interaction between a regulator and a target gene. This interaction can be formally described in terms of the concentrations of regulator, pre-mRNA molecules of the target gene and enzime RNAP. For this, Michaelis-Menten interaction kinetics and the Hill equation are employed to model the chemical reactions that ocurr during the transcription process [4, 2, 16].

First, we consider a situation with no regulation in which RNAP (P) binds a nucleotide sequence in the promoter (N) of a gene to form a complex (PN). After this, transcription occurs and a pre-mRNA molecule (X) is generated. These chemical reactions are represented schematically as follows:

$$P + N \underset{k_{-1}}{\overset{k_{+1}}{\rightleftarrows}} PN \overset{k_{+2}}{\rightarrow} P + N + X \,, \tag{1}$$

where k_{+1} and k_{-1} are respectively the rates of complex formation and complex dissociation and k_{+2} is the rate of transcription. Michaelis-Menten kinetics allow us to obtain a differential equation for the concentration of X (see Appendix A), namely

$$\frac{d[X]}{dt} = \frac{V_m[P]}{[P] + K_m} - \delta[X] \,, \tag{2}$$

where δ is the rate of degradation of X, $[\cdot]$ stands for 'concentration of', $V_m = k_{+2}[N]_0$ is the maximum rate of synthesis and $K_M = (k_{-1} + k_{+2})/k_{+1}$ is referred to as the Michaelis constant. When $[P]$ exceeds K_M, the rate of syntehsis of X begins to saturate at its maximum value V_m.

The previous example can be easily extended to account for positive regulation by a TF. In particular, RNAP may not bind the promoter of a gene unless an activator (A) has already bound a nucleotide sequence in the promoter [4]. The bottom left of Figure 3 illustrates this. Additionally, the promoter may contain α copies of the DNA sequence bound by A. Under this setting, the differential equation for the concentration of X (see Appendix B) is given by

$$\frac{d[X]}{dt} = V_m \cdot \frac{[A]^\alpha}{[A]^\alpha + K_A} \cdot \frac{[P]}{[P] + K_m} - \delta[X] \,, \tag{3}$$

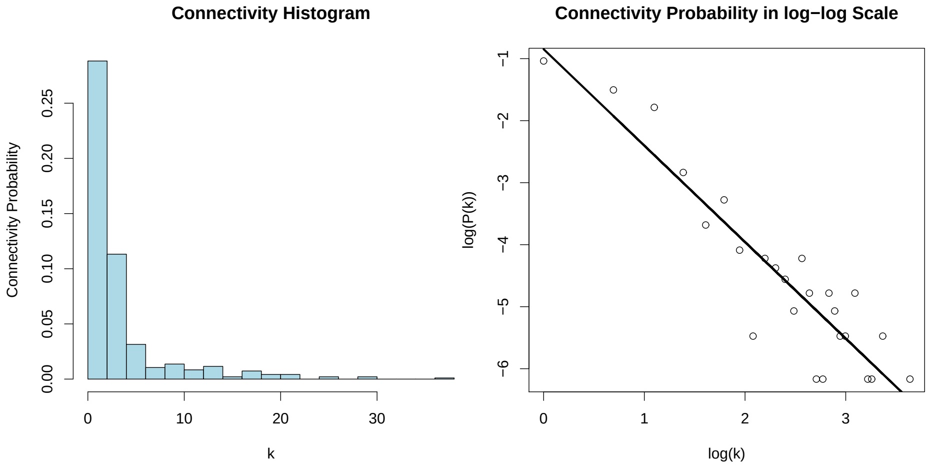

Figure 4: Left, histrogram for the connectivity in the transcription control network of budding yeast. Hubs or highly connected nodes are represented in this histrogram by a heavy tail. Right, log-log plot of the empirical connectivity probability function. The points in this plot are well approximated by a straight line. This is a clear signature of an underlying power law process.

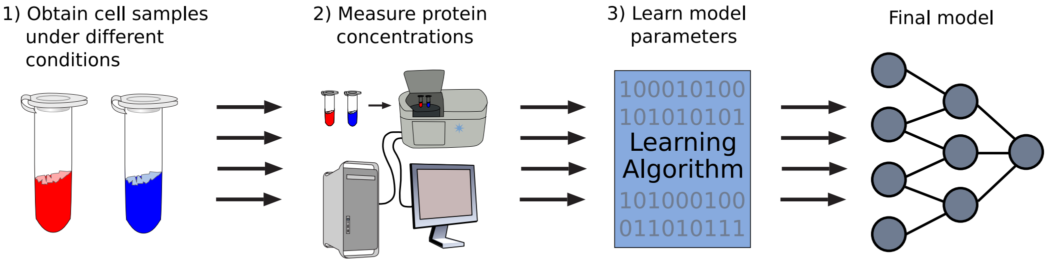

Figure 5: A general procedure for constructing a model of cell functioning. First, cell samples are obtained under different conditions, e.g. metastatic cancer cells and non-metastatic cancer cells. Second, the experimenter measures protein concentrations in the cells. Third, a learning algorithm computes the parameters of a model that describes the intrinsic structure in the data. The resulting model may then be employed in posterior analyses or predictions.

where K_A is a constant that determines the saturation threshold for A. Similarly, the transcription process can be negatively regulated by a repressor (R) that binds a particular nucleotide sequence in the promoter of a gene. Under this configuration, transcription by RNAP only occurs when R is not bound [2]. The bottom right of Figure 3 illustrates this. Additionally, the promoter of the gene may contain β copies of the nucleotide sequence bound by the repressor. The corresponding differential equation for the concentration of X (see Appendix C) is given by

$$\frac{d[X]}{dt} = V_m \cdot \frac{1}{1 + [R]^\beta / K_R} \cdot \frac{[P]}{[P] + K_m} - \delta[X] \,, \tag{4}$$

where K_R is a constant that determines how much repressor is necessary for reducing the production rate to a half of the maximal activity that would be obtained under no regulation and no transcript degradation [2].

5 Microarray Chips

The internal state of a cell is determined by the proteins that form the cell and their corresponding concentrations. Hence, we may construct a model of cell functioning (e.g. a TCN) by measuring protein concentrations in a cell under different conditions, as illustrated in Figure 5. This may require of specific machine learning and pattern recognition methods. The information encoded in the model

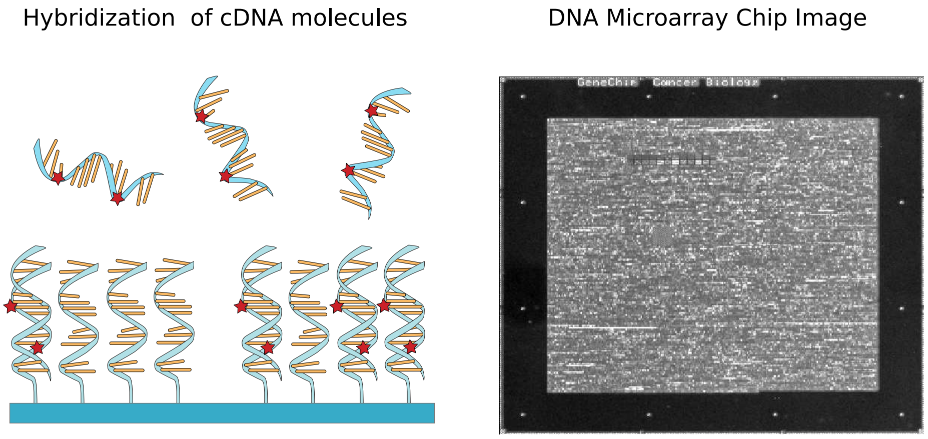

Figure 6: Left, fluorecently marked cDNA molecules bind complementary probes on the surface of the microarray chip. The relative number of these hybridizations depends on the concentration of RNA transcripts in the original sample. The four leftmost probes match a gene that is not expressed by the cells under study. Therefore, few cDNA molecules bind this group of probes. By contrast, the four rightmost probes match a gene that is significantly expressed in the original sample. Thus, several cDNA molecules bind this group of probes. Right, image of logarithmic probe intensities in a DNA microarray chip. Each dot in the image corresponds to a group of identical probes that match a specific gene subsequence. The dot intensities are given by the respective concentrations of bound cDNA molecules.

may then be employed in posterior analyses or predictions. For example, we might be interested in studying the mechanisms underlying metastasis in cancer cells. In this case, samples from metastatic and non-metastatic cancer cells are obtained and protein concentrations are measured. The resulting data are then analyzed by a machine learning method that constructs a model for the ’metastatic’ and ’non-metastatic’ states. Finally, this model is used to identify proteins and regulatory interactions involved in the transition of a cancer cell from one state to the other.

Measurements of protein concentration can be obtained indirectly using a recent high-throughput molecular technology referred to as DNA microarray chips [5]. Microarray chips allow scientists to simultaneously monitor the concentration of thousands of RNA transcripts in a sample of cells. Although RNA concentration is not the same as protein concentration, microarray measurements are expected to be representative of the relative number of fully-formed proteins in a cell population.

DNA microarrays consist of a glass slide or a silicon chip with thousands of microscopic spots, each containing a small amount of identical DNA sequences called *probes* that are attached to the surface of the chip. The probes in an individual spot correspond to a short section of a particular gene whose expression is to be monitored. Microarray experiments generally incorporate five steps. First, RNA molecules are extracted from a sample of cells or tissues by common organic extraction procedures. Second, a reverse transcription procedure translates the RNA molecules into complementary DNA (cDNA) molecules that are labeled with a flourescent marker. Third, the cDNA molecules bind to complementary probes on the surface of the microarray chip in a process referred to as *hybridization*. This is illustrated in the left of Figure 6. Fourth, after washing the surface of the chip, a scanner is used to measure the amount of fluorecent marker present on each spot. The output of this scanning process is a microarray image, see the right of Figure 6. The dots in the microarray image correspond to groups of identical probes that match a specific gene subsequence and the dot intensities are given by the respective concentrations of bound cDNA molecules. Because cDNA concentration is representative of original RNA concentration, the microarray image offers us a ’snapshot’ of all the transcriptional activity that was taking place in the biological sample. Finally, before obtaining a final expression measurement for each gene, variations in microarray data caused by technical and biological factors must be corrected in a step called *normalization*. More information about DNA chips and microarray experiments can be found in [17].

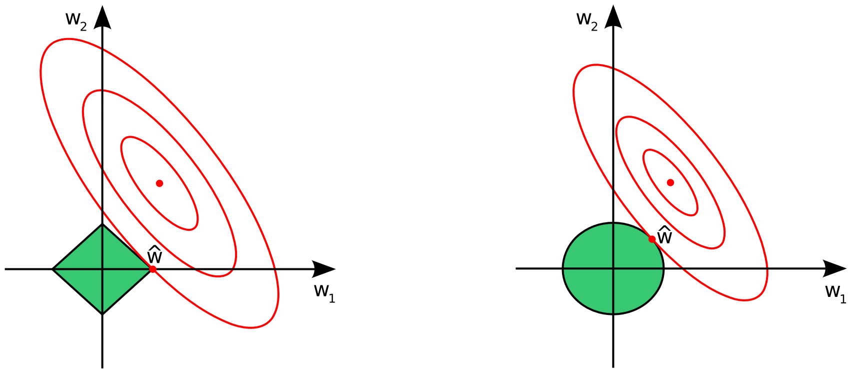

Figure 7: Elliptical countours of the residual sum of squares function (red) along with the constraint regions (green) for the lasso (left) and ridge regression (right): $|w_1| + |w_2| \leq 1$ and $w_1^2 + w_2^2 \leq 1$, respectively. Both the lasso and ridge regression select the first point $\hat{\mathbf{w}}$ where the elliptical contours hit the corresponding constraint region. In this case, the lasso returns a sparse solution $\hat{\mathbf{w}} = (\hat{w}_1, \hat{w}_2)$ where $\hat{w}_2$ is equal to zero. By contrast, the ridge regression solution is not sparse.

DNA microarray chips represent a significant technological breakthrough because, for the first time, scientists have been able to monitor the internal state of a biological system [18, 19]. Nevertheless, microarray gene expression data suffer from several drawbacks. First, microarray measurements are affected by a substantial amount of experimental and biological noise. Second, the economic cost of microarray experiments is considerably high. Hence, microarray datasets usually include a limited number n of sample instances. Finally, each microarray sample is a d-dimensional vector with the relative expression of d genes where $d \gg n$. This combination of noisy measurements, reduced number of sample instances and very high-dimensional feature space makes significantly difficult the analysis of microarray data. In particular, most machine learning methods are designed to work with large datasets of relatively low dimensionality that include minor levels of noise. When these methods are applied to microarray data 'overfitting' is a considerable problem. The conclusion is that gene expression datasets have to be analyzed by specific pattern recognition methods.

6 Sparse Linear Methods

A popular approach for modeling limited high-dimensional data is to consider an underlying sparse linear model whose output is formed by the activity of a reduced subset of features. Let us assume a standard regression framework where $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is a training set with n observations, each one formed by a d-dimensional feature vector $\mathbf{x}_i$ and a corresponding scalar target y_i. The objective is to learn a function whose input is a new feature vector $\hat{\mathbf{x}}$ and whose output is a prediction $\hat{y}$ for the true target value of $\hat{\mathbf{x}}$. A simple characterization for such function is a linear model:

$$\hat{y} = w_1 \hat{x}_1 + \cdots + w_d \hat{x}_d \,, \tag{5}$$

where $\hat{\mathbf{x}} = (\hat{x}_1, \ldots, \hat{x}_d)^{\mathrm{T}}$ and $\mathbf{w} = (w_1, \ldots, w_d)^{\mathrm{T}}$ is a vector of real coefficients. Usually, $\mathbf{w}$ is fixed by minimizing the residual sum of squares on the training data subject to a bound on its euclidian norm. This process is called *ridge* regression [20]. However, when the level of noise in the features is high and $d \gg n$, ridge regression may identify spurious patterns in $\mathcal{D}$ that do not generalize to new data instances. This is particularly the case when y_i only depends on a few components of $\mathbf{x}_i$. Under this setting, overfitting can be reduced by assuming that the optimal choice for $\mathbf{w}$ is a sparse vector with many zero components. In this case, an estimate for $\mathbf{w}$ can be obtained by running a sparse linear method [7, 8, 9] on the training data. As well as reducing overffiting, sparse techniques also allow us to identify the most relevant features for solving the learning task.

6.1 The Lasso

The lasso (least angle shrinkage and selection operator) selects $\mathbf{w}$ by minimizing the sum of squares on $\mathcal{D}$ subject to a bound on the L_1 norm of this coefficient vector [7, 20]. In particular, the lasso estimate for $\mathbf{w}$ is defined by

$$\hat{\mathbf{w}} = \underset{\mathbf{w}}{\text{argmin}} \left\{ \sum_{i=1}^{n} \left(y_i - \mathbf{w}^{\mathrm{T}} \mathbf{x}_i \right)^2 \right\} \qquad \text{subject to} \qquad \sum_{i=1}^{d} |w_i| \leq \alpha \,, \tag{6}$$

where α is a positive regularization parameter. The computation of $\hat{\mathbf{w}}$ is a cuadratic programming problem with linear inequality constraints [7] that can be solved very efficiently [21]. Additionally, $\hat{\mathbf{w}}$ is frequently a sparse vector with many components equal to zero. The origin of this sparsity lays in the specific contours of the L_1 constraint function as illustrated in Figure 7 for a two-dimensional regression problem. The residual sum of squares is a cuadratic function whose elliptical contours are centered at the full least squares estimate. The constraint regions for the lasso and ridge regression are respectively the diamond $|w_1| + |w_2| \leq 1$ and the disk $w_1^2 + w_2^2 \leq 1$. Both methods select the first point $\hat{\mathbf{w}}$ where the elliptical contours hit the constraint region. When the soluction occurs at a corner of the diamond, the corresponding component of $\hat{\mathbf{w}}$ is equal to zero. Because the disk has no corners, the ridge regression solution is never sparse. When the dimensionality is increased, the diamond turns into a rhomboid with many corners. The lasso solution is then very likely to ocurr at some of those corners, producing a sparse coefficient vector.

The regularization parameter α controls the number of components of $\hat{\mathbf{w}}$ that are zero. In particular, the smaller α is, the 'sparser' $\hat{\mathbf{w}}$ results. However, the optimal choice for α is specific of the problem under analysis and there is no straightforward rule for computing it. Hence, to fix this regularization parameter, we may perform a cross-validation search using the training data. Once α has been tuned and $\hat{\mathbf{w}}$ is available, we may identify the most relevant features for solving the regression task. These features correspond to the components of $\hat{\mathbf{w}}$ that are different from zero.

6.2 Automatic Relevance Determination

Given $\mathcal{D}$, the automatic relevance determination (ARD) framework yields a sparse estimate for $\mathbf{w}$ by regularizing the space of solutions with a data-dependent prior that forces some of the components of this coefficient vector to be zero [22, 9]. The basic ARD prior for $\mathbf{w}$ is given by

$$\mathcal{P}(\mathbf{w}) = \prod_{i=1}^{d} \mathcal{N}(w_i, 0, \alpha_i^{-1}) \,, \tag{7}$$

where $\mathcal{N}(x, \mu, \sigma^2)$ is the density function for a Gaussian distribution with mean μ and variance σ^2 evaluated at x and the α_i are non-negative hyperparameters controlling the prior variance for each unknown component of $\mathbf{w}$. In particular, when α_i is small, we expect the solution for w_i to be rather large in absolute value and when a_i is large, the solution for w_i is expected to be close to zero. The value of these hyperparameters $\alpha_1, \ldots, \alpha_d$ is computed from the data by first, marginalizing over the coefficient vector $\mathbf{w}$ and then, maximizing the type-II likelihood or evidence [9]. For this, the targets y_i are assumed to be generated by the model

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \sigma^2 \mathbf{n} \,, \tag{8}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$, $\mathbf{X}$ is the $n \times d$ matrix whose i-th row is equal to $\mathbf{x}_i^{\mathrm{T}}$, $\mathbf{n}$ is an n-dimensional random vector whose components are independent and identically distributed following a standard Gaussian distribution and σ^2 is a positive parameter that represents the level of noise in the targets. The logarithm of the evidence under this model is given by

$$\mathcal{L}(\boldsymbol{\alpha}) = \log \int \prod_{i=1}^{n} \mathcal{P}(y_i | \mathbf{x}_i, \mathbf{w}) \mathcal{P}(\mathbf{w}) \, d\mathbf{w} = -\frac{1}{2} \log |\mathbf{C}| - \frac{1}{2} \mathbf{y}^{\mathrm{T}} \mathbf{C}^{-1} \mathbf{y} + \text{constant} \,, \tag{9}$$

where $\boldsymbol{\alpha} = (a_1, \ldots, a_d)^{\mathrm{T}}$, $\mathbf{C}$ is the $n \times n$ matrix

$$\mathbf{C} = \sigma^2 \mathbf{I} + \sum_{i=1}^{d} a_i^{-1} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^{\mathrm{T}} \tag{10}$$

and $\boldsymbol{\varphi}_i$ is the i-th column of matrix $\mathbf{X}$. Appendix D describes a very efficient greedy algorithm for the local maximization of (9). Once a maximum $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \ldots, \alpha_d)^{\mathrm{T}}$ of $\mathcal{L}(\boldsymbol{\alpha})$ is obtained, the corresponding estimate for $\mathbf{w}$ is given by the posterior mean, namely

$$\hat{\mathbf{w}} = \left(\sigma^2 \mathbf{A} + \mathbf{X}^{\mathrm{T}} \mathbf{X} \right)^{-1} \mathbf{X}^{\mathrm{T}} \mathbf{y} \,, \tag{11}$$

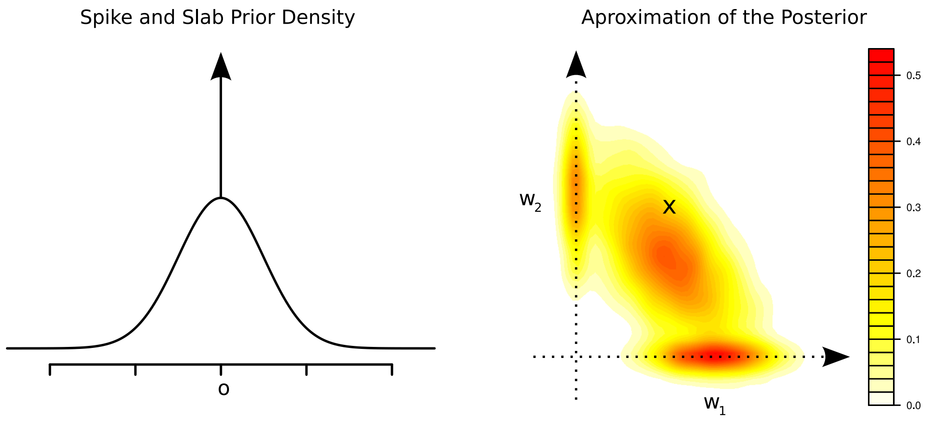

Figure 8: Left, ’spike and slab’ prior density for a single coefficient. This density is a mixture of a Gaussian and a Dirac delta function, both centered at zero. The delta function has been represented by an arrow pointing to $+\infty$. Right, approximation of the posterior distribution for $\mathbf{w}$. The training data is the same as in the two-dimensional regression problem of Figure 7 and σ^2 was fixed to 1. The prior for $\mathbf{w}$ is a spike and lab distribution where $s^2 = 1$ and $\omega = 0.5$. The maximum likelihood solution is marked in the diagram with an ’x’. The density plot was obtained by generating 10,000 samples from the posterior by Gibbs sampling [27] and then computing a kernel density estimate. Three regions of high probability can be identified in the plot. Two of them correspond to sparse solutions where either $w_1 = 0$ or $w_2 = 0$. These sparse regions are approximated as clouds with positive lengths and widths, but they are actually straight lines with only one dimension. The third region lays between the other two and corresponds to non-sparse solutions. Finally, the posterior density is maximal in the sparse region in which $w_2 = 0$.

where $\mathbf{A} = \mathrm{diag}(\hat{\alpha}_i, \ldots, \hat{\alpha}_d)$. As a result of the maximization of $\mathcal{L}(\boldsymbol{\alpha})$, several hyperparameters α_i will take an infinite value and the corresponding w_i will have a posterior distribution with mean and variance both zero. Hence, the estimate for $\mathbf{w}$ given by the ARD approach is typically a sparse vector. The origin of this sparsity can be understood from an alternative formulation of the ARD objective function $\mathcal{L}(\boldsymbol{\alpha})$ using auxiliary functions [23]. Under this new formulation, a local maximum of the objective function is obtained by solving a series of re-weighted lasso regression problems. Finally, according to ARD, the most relevant features for solving the regression task correspond to the α_i that are finite once $\mathcal{L}(\boldsymbol{\alpha})$ has been maximized.

6.3 Bayesian Sparsity

Under a Bayesian framework, sparsity is enforced by incorporating a prior on $\mathbf{w}$ that concentrates its mass on sparse parameter vectors. Some sparsifying priors are the Laplace [24] or the ’spike and slab’ [8] distributions. This latter prior works by introducing a d-dimensional vector of binary latent variables denoted $\mathbf{z} = (z_1, \ldots, z_d) \in \{-1, 1\}^d$, where $z_i = 1$ if the i-th component of $\mathbf{w}$ is not zero and $z_i = -1$ otherwise. Conditioning on $\mathbf{z}$, the prior for $\mathbf{w}$ is given by

$$\mathcal{P}(\mathbf{w}|\mathbf{z}) = \prod_{i=1}^{d} \left[\frac{z_i + 1}{2} \mathcal{N}(w_i, 0, s^2) + \frac{1 - z_i}{2} \delta(w_i) \right] , \tag{12}$$

where s^2 is the prior variance for the non-zero components of $\mathbf{w}$ and δ is the Dirac delta function which represents a point mass at zero. Note that when $z_i = -1$, the i-th component of $\mathbf{w}$ is not random because it can only take value zero. By contrast, when $z_i = 1$, the prior for w_i is a Gaussian with zero mean and variace s^2, where s^2 is significantly larger than zero. The term ’spike and slab’ has its origin in the delta function (the spike) and the broad Gaussian (the slab) that appear in (12). To complete the prior for $\mathbf{w}$, we have to fix the ’a priori’ probabilities for each component of $\mathbf{z}$. These probabilities are often described by independent Bernoulli distributions:

$$\mathcal{P}(\mathbf{z}) = \prod_{i=1}^{d} \left[\frac{z_i + 1}{2} \omega + \frac{1 - z_i}{2} (1 - \omega) \right] , \tag{13}$$

where $0 < \omega < 1$ is the prior probability that a specific component of $\mathbf{w}$ is not zero. Very frequently, this constant is fixed to 0.5 so that (13) is non-informative on $\mathbf{z}$ [25]. The unconditional prior for $\mathbf{w}$ is then obtained by marginalizing the product of (12) and (13) with respect to $\mathbf{z}$. The resulting density for each component of $\mathbf{w}$ is a mixture of a Gaussian and a delta function, as illustrated in the left of Figure 8. Finally, before performing any Bayesian analysis, we must specify a likelihood function for the model parameters. For this, we assume that the targets are generated by model (8). Under this setting, the likelihood for $\mathbf{w}$ given $\mathbf{y}$ and $\mathbf{X}$ is

$$\mathcal{P}(\mathbf{y}|\mathbf{X},\mathbf{w}) = \prod_{i=1}^{n} \mathcal{P}(y_i|\mathbf{x}_i,\mathbf{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i|\mathbf{w}^{\mathrm{t}}\mathbf{x}_i,\sigma^2)\,. \tag{14}$$

Once we have a likelihood function and a prior distribution, Bayes' theorem allows us to compute the posterior distribution for $\mathbf{w}$ when $\mathbf{y}$ and $\mathbf{X}$ are observed:

$$\mathcal{P}(\mathbf{w}|\mathbf{y},\mathbf{X}) = \frac{\sum_{\mathbf{z}} \mathcal{P}(\mathbf{y}|\mathbf{X},\mathbf{w})\mathbf{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z})}{\mathcal{P}(\mathbf{y}|\mathbf{X})}\,, \tag{15}$$

where the latent variables $\mathbf{z}$ have been marginalized out. This posterior represents our uncertainty on $\mathbf{w}$ after observing $\mathbf{y}$ and $\mathbf{X}$. The right of Figure 8 displays a plot of the posterior distribution for a two-dimensional regression problem. Three regions of the Euclidian plane have large posterior probabilities in this case. Two of them correspond to sparse parameter vectors where either $w_1 = 0$ or $w_2 = 0$. The third region lays between the other two and represents solutions that are not sparse. To compute a probabilistic prediction for the target $\hat{y}$ of a new feature vector $\hat{\mathbf{x}}$ we use

$$\mathcal{P}(\hat{y}|\mathbf{X},\mathbf{y},\hat{\mathbf{x}}) = \int \mathcal{P}(\hat{y}|\hat{\mathbf{x}},\mathbf{w})\mathcal{P}(\mathbf{w}|\mathbf{X},\mathbf{y})\,d\mathbf{w}\,. \tag{16}$$

In this formula, any possible value for $\mathbf{w}$ is considered for prediction with a weight proportional to its posterior probability. Additionally, the Bayesian machinery under the 'spike and slab' prior also allows us to perform feature selection. For this, we employ the posterior distribution for the latent vector $\mathbf{z}$, namely

$$\mathcal{P}(\mathbf{z}|\mathbf{y},\mathbf{X}) = \frac{\int \mathcal{P}(\mathbf{y}|\mathbf{X},\mathbf{w})\mathbf{P}(\mathbf{w}|\mathbf{z})\mathcal{P}(\mathbf{z})\,d\mathbf{w}}{\mathcal{P}(\mathbf{y}|\mathbf{X})}\,, \tag{17}$$

where $\mathbf{w}$ has been marginalized out. The most relevant features for solving the regression problem are then given by those z_i for which $z_i = 1$ has the largest marginal probability under (17). Finally, although the sparse Bayesian framework can be very attractive, its computational burden limits its applicability. In particular, most of the aforementioined operations require to compute summations and integrals whose cost grows exponentially in d. In practice, exact inference is not feasible under the sparse Bayesian setting and approximate methods have to be employed. These involve either sampling from the posterior distribution or approximating the posterior with a simpler distribution that is analytically tractable [27].

7 Summary

In this paper we have given a brief introduction to the mechanism by which cells produce proteins. This mechanism is called *gene expression* and consists in the synthesis of a functional polypeptide chain from the genetic information stored in the DNA. The first step in the gene expression process is called *transcription* and it involves the copying of a DNA sequence into a pre-mRNA molecule by the enzime RNA polymerase (RNAP). During the lifetime of a cell, different proteins have to be produced at different rates. This requires of a control step in the gene expression process. Control of final protein concentration is mainly performed at the transcription level by a series of regulatory molecules that modify the activity of RNAP. The most important of these molecules are a set of proteins called *transcription factors* that alter the affinity of RNAP to bind a particular sequence of DNA. Interactions between regulators and regulatees at the transcription level can be organized in a *transcription control network* (TCN), whose edges are formally described by differential equations.

For different reasons, we might be interested in constructing a model of cell functioning, e.g. a TCN. Such a model can be obtained by first, measuring protein concentrations in a cell under different conditions and second, applying specific machine learning methods to the data. Microarray chips are a novel technology that allows scientists to approximately measure protein concentrations in a sample of cells. However, microarray datasets are frequently very noisy, with a reduced number n of sample instances and a feature space of very large dimension d, where $d \gg n$. These characteristics make significantly difficult the analysis of microarray data since most machine learning methods are likely to identify irrelevant patters in this 'large d, small n' scenario. A common solution is then to consider a *sparse linear model*, whose parameters are learned using a *sparse linear method* like the Lasso, the automatic relevance determination framework or the 'spike and slab' Bayesian technique.

A Differential Equation under no Regulation

The rates of product formation for PN and X are given by

$$\frac{d[PN]}{dt} = k_{-1}[P][N] - (k_{+2} + k_{-1})[PN] = 0\,, \tag{18}$$

$$\frac{d[X]}{dt} = k_{+2}[PN] - \delta[X]\,, \tag{19}$$

where the first equation is equal to zero because we have assumed that $[PN]$ changes much more slowly than $[P]$ and $[N]$. Additionally, we have also assumed that the total concentration of promoter is constant and equal to $[N]_0$, namely $[N] + [PN] = [N]_0$. Solving for $[N]$ and substituting in (18) yields $[PN] = [P][N]_0/(K_m + [P])$, where $K_m = (k_{-1} + k_{+2})/k_{+1}$. Replacing this result in (19) allows us to obtain

$$\frac{d[X]}{dt} = \frac{V_m[P]}{[P] + K_m} - \delta[X]\,, \tag{20}$$

where $V_m = k_{+2}[N]_0$.

B Differential Equation under Positive Regulation

When the promoter N binds α molecules of A the complex αAN is formed. We ignore intermediate states where N binds less than α molecules of activator. The corresponding scheme for the chemical reactions between A and N is

$$\alpha A + N \underset{k_{-3}}{\overset{k_{+3}}{\rightleftarrows}} \alpha AN\,, \tag{21}$$

where k_{+3} and k_{-3} are respectively the rates of complex formation and complex dissociation. The rate of product formation for αAN is therefore

$$\frac{d[\alpha AN]}{dt} = k_{+3}[N][A]^\alpha - k_{-3}[\alpha AN] = 0\,, \tag{22}$$

where we have assumed that the concentration of αAN changes very slowly. Additionally, we also assume that the total concentration of promoter, either bound by A or free, is constant and equal to $[N]_0$, namely $[N] + [\alpha AN] = [N]_0$. Solving for $[N]$ and substituting in (22) allows us to obtain the concentration of promoter that can be bound by RNAP, namely

$$[\alpha AN] = \frac{[A]^\alpha [N]_0}{K_A + [A]^\alpha}\,, \tag{23}$$

where $K_A = k_{-3}/k_{+3}$. This is a particular form of the Hill equation [2] and it describes the cooperative binding of the α molecules of A. A differential equation for the rate of production of X is then obtaind by replacing $[N]_0$ in (20) with the previous expression for $[\alpha AN]$.

C Differential Equation under Negative Regulation

When the promoter N binds β molecules of repressor R the complex βRN is formed. We ignore intermediate states where N binds less than β molecules of R. The corresponding scheme for the chemical reactions between R and N is

$$\beta R + N \underset{k_{-4}}{\overset{k_{+4}}{\rightleftarrows}} \beta RN\,, \tag{24}$$

where k_{+4} and k_{-4} are respectively the rates of complex formation and complex dissociation. The rate of product formation for βRN is therefore

$$\frac{d[\beta RN]}{dt} = k_{+4}[N][R]^{\beta} - k_{-4}[\beta RN] = 0\,, \tag{25}$$

where we have assumed that the concentration of βRN changes very slowly. Additionally, we also assume that the total concentration of promoter, either bound by R or free, is constant and equal to $[N]_0$, namely $[N] + [\beta RN] = [N]_0$. Solving for $[\beta RN]$ and substituting in (25) allows us to obtain the concentration of promoter that can be bound by RNAP, namely

$$[N] = \frac{[N]_0}{1 + [R]^{\beta}/K_R}\,, \tag{26}$$

where $K_R = k_{-4}/k_{+4}$. A differential equation for the rate of production of X is then obtaind by replacing $[N]_0$ in (20) with the previous expression for $[N]$.

D Maximization of the Evidence in the ARD Framework

To maximize $\mathcal{L}(\boldsymbol{\alpha})$ with respect to $\boldsymbol{\alpha}$ we follow [26, 27]. In particular, the contribution of each α_i to the total logarithm of the evidence is separated out:

$$\mathcal{L}(\boldsymbol{\alpha}) = \mathcal{L}(\boldsymbol{\alpha}_{-i}) + \frac{1}{2}\left[\log\alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right]\,, \tag{27}$$

where $\boldsymbol{\alpha}_{-i}$ is obtained from $\boldsymbol{\alpha}$ by setting the i-th component of this vector to $+\infty$ and the quantities s_i and q_i are given by

$$s_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\boldsymbol{\varphi}_i\,, \tag{28}$$

$$q_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{-i}^{-1}\mathbf{y}\,, \tag{29}$$

where $\mathbf{C}_{-i}$ is obtained by pulling out the contribution from α_i in matrix $\mathbf{C}$, namely

$$\mathbf{C}_{-i} = \mathbf{C} - \alpha_i^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\,. \tag{30}$$

The quantities s_i and q_i are respectively called the *sparsity* and the *quality* of the i-th feature. A large value of s_i with respect to q_i means that the i-th feature is unlikely to be relevant and consequently, the optimal value for α_i is likely to be infinite. The 'sparsity' represents the redundancy of the i-th feature with respect to the other features already included in the model. The 'quality' measures the alignment of the i-th feature with respect to the errors of the model when this feature is excluded. The local maximization of $\mathcal{L}(\boldsymbol{\alpha})$ with respect to α_i occurs when the derivative of (27) with respect to α_i is equal to zero. This derivative is

$$\frac{d\mathcal{L}(\boldsymbol{\alpha})}{d\alpha_i} = \frac{\alpha_i^{-1}s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} \tag{31}$$

and there are two possible forms for the solution. When $q_i^2 \leq s_i$, the solution is obtained by setting $\alpha_i = +\infty$ and when $q_i^2 > s_i$, we maximize $\mathcal{L}(\boldsymbol{\alpha})$ with respect to α_i by setting $\alpha_i = s_i^2/(q_i^2 - s_i)$. For the computation of the quantities s_i and q_i we employ

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i}\,, \qquad q_i = \frac{\alpha_i Q_i}{\alpha_i - S_i}\,, \qquad S_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}^{-1}\boldsymbol{\varphi}_i\,, \qquad Q_i = \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{y}\,, \tag{32}$$

where $q_i = Q_i$ and $s_i = S_i$ when $\alpha_i = +\infty$. Finally, when α_i is modified we may update $\mathbf{C}^{-1}$ as

$$\mathbf{C}_{\text{new}}^{-1} = \mathbf{C}_{\text{old}}^{-1} - \frac{\mathbf{C}_{\text{old}}^{-1}\boldsymbol{\varphi}_i\boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{\text{old}}^{-1}}{\alpha_i^{\text{new}} + \boldsymbol{\varphi}_i^{\mathrm{T}}\mathbf{C}_{\text{old}}^{-1}\boldsymbol{\varphi}_i}\,. \tag{33}$$

The resulting algorithm for maximizing $\mathcal{L}(\boldsymbol{\alpha})$ implies the following steps:

1. Initialize $\alpha_i = +\infty$ for $i = 1, \ldots, d$ and $\mathbf{C}^{-1} = \sigma^{-2}\mathbf{I}$.
2. Choose an α_i to optimize and compute q_i and s_i.
3. If $q_i^2 > s_i$ set $\alpha_i = s_i^2/(q_i^2 - s_i)$ else, set $\alpha_i = +\infty$. Update $\mathbf{C}^{-1}$ if α_i was modified.
4. Go to step 2 until all the α_i have reached convergence, otherwise terminate.

The bottleneck of this algorithm lays in the update of matrix $\mathbf{C}^{-1}$, which has a computational cost equal to $\mathcal{O}(n^2)$, where n is the number of instances in the training set. A slightly different algorithm for maximizing $\mathcal{L}(\boldsymbol{\alpha})$ is described by [26, 27]. In that case, each iteration has a cost $\mathcal{O}(m^3)$, where m is the number of α_i that are finite during the current iteration.

References

[1] G. A. Petsko and D. Ringe. *Protein Structure and Function*. New Science Press, 2003.

[2] U. Alon. *An Introduction to Systems Biology*. CRC Press, 2006.

[3] M. Bénédicte. After 30 years of study, the bacterial SOS response still surprises us. *PLoS Biology*, 3(7):e255, 2005.

[4] T. Gardner and J. Faith. Reverse-engineering transcription control networks. *Physics of Life Reviews*, 2(1):65–88, 2005.

[5] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, 1995.

[6] D. K. Slonim. From patterns to pathways: gene expression data analysis comes of age. *Nature Genetics*, 32(supp):502–508, 2002.

[7] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[8] E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–374, 1997.

[9] M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[10] H. Lodish, A. Berk, C. A. Kaiser, M. Krieger, M. P. Scott, A. Bretscher, H. Ploegh, and P. Matsudaira. *Molecular Cell Biology*. W. H. Freeman and Company, 2008.

[11] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès. Topological and causal structure of the yeast transcriptional regulatory network. *Nature genetics*, 31(1):60–63, 2002.

[12] P. Shannon, A. Markieland, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498, 2003.

[13] B. H. Junker and F. Schreiber, editors. *Analysis of Biological Networks*. Wiley-Interscience, 2008.

[14] A. L. Barabási and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[15] K. Basso, A. A. Margolin, G. Stolovitzky, U. Klein, R-Dalla-Favera, and A. Califano. Reverse engineering of regulatory networks in human B cells. *Nature genetics*, 37(4):382–390, 2005.

[16] S. I. Rubinow. *Introduction to mathematical biology*. John Wiley & Sons, 1975.

[17] I. Hovatta, K. Kimppa, A. Lehmussola, T. Pasanen, J. Saarela, I. Saarikko, J. Saharinen, P. Tiikkainen, T. Toivanen, M. Tolvanen, M. Vihinen, and G. Wong. *DNA microarray data analysis*. CSC Scientific Computing Ltd, 2005.

[18] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. 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 cancer. *Nature*, 415(6871):530–536, 2002.

[19] J. P. Daily, D. Scanfeld, N. Pochet, K. Le Roch, D. Plouffe, M. Kamal, O. Sarr, S. Mboup, O. Ndir, D. Wypij, K. Levasseur, E. Thomas, P. Tamayo, C. Dong, Y. Zhou, E. S. Lander, D. Ndiaye, D. Wirth, E. A. Winzeler, J. P. Mesirov, and A. Regev. Distinct physiological states of plasmodium falciparum in malaria-infected patients. *Nature*, 450(7172):1091–1095, 2007.

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inferene, and Prediction*. Springer, 2001.

[21] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004.

[22] R. M. Neal. *Bayesian Learning for Neural Networks (Lecture Notes in Statistics)*. Springer, 1996.

[23] D. Wipf and S. Nagarajan. A new view of automatic relevance determination. In *Advances in Neural Information Processing Systems 20*, pages 1625–1632. MIT Press, 2008.

[24] M. W. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.

[25] H. Ishwaran and J.S. Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[26] M.E. Tipping and A. C. Faul. Fast marginal likelihood maximisation for sparse Bayesian models. In M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pages 3–6, Key West, FL, January 2003.

[27] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.