Where do we stand in regularization for life science studies?

Veronica Tozzo [1,*], Chloé-Agathe Azencott [2, 3, 4], Samuele Fiorini [5], Emanuele Fava [6], Andrea Trucco [6] and Annalisa Barla [1]

[1] Departiment of Informatics, Bioengineering, Robotics and System Engineering - DIBRIS, University of Genoa, Genoa, 16146, Italy;

[2] Centre for Computational Biology - CBIO, MINES ParisTech, PSL Research University, Paris, 75006, France;

[3] Institut Curie, PSL Research University, Paris, 75005, France and

[4] INSERM, U900, Paris, 75005, France.

[5] Iren S.p.a, Genoa, 16122, Italy.

[6] Departiment of Electrical, Electronic, Telecommunications Engineering, and Naval Architecture (DITEN) University of Genoa, Genoa,16145, Italy.

* corresponding author

mails:veronica.tozzo@dibris.unige.it

Abstract

More and more biologists and bioinformaticians turn to machine learning to analyze large amounts of data. In this context, it is crucial to understand which is the most suitable data analysis pipeline for achieving reliable results. This process may be challenging, due to a variety of factors, the most crucial ones being the data type and the general goal of the analysis (e.g. explorative or predictive). Life science data sets require further consideration as they often contain measures with a low signal-to-noise ratio, high-dimensional observations, and relatively few samples. In this complex setting, regularization, which can be defined as the introduction of additional information to solve an ill-posed problem, is the tool of choice to obtain robust models. Different regularization practices may be used depending both on characteristics of the data and of the question asked, and different

choices may lead to different results. In this paper we provide a comprehensive description of the impact and importance of regularization techniques in life science studies. In particular, we provide an intuition of what regularization is and of the different ways it can be implemented and exploited. We propose four general life sciences problems in which regularization is fundamental and should be exploited for robustness. For each of these large families of problems, we enumerate different techniques as well as examples and case studies. Lastly, we provide a unified view of how to approach each data type with various regularization techniques.

1 Motivation

In the era of personalized medicine, biospecimen collection and biological data management is still a challenging and expensive task (Toga and Dinov, 2015). Only few large-scale research enterprises, such as ENCODE (`encodeproject.org`), ADNI (`adni.loni.usc.edu`) or TCGA (`cancergenome.nih.gov`), have sufficient financial and human resources to manage, share and distribute access of heterogeneous types of biological data. To date, many biomedical studies still rely on a small number of collected samples (McNeish and Stapleton, 2016). A number that is even lower in cases of rare diseases (Garg et al., 2016) or in high-throughput molecular data (*e.g.* genomics and proteomics) where the number of variables measured can be in the order of hundreds of thousands (Yu et al., 2013).

Asking biological or clinical questions from this data using machine learning techniques requires particular consideration of many factors, such as random fluctuations in the measurements introduced by the acquisition devices, a small number of samples, or, observed variables may not be representative of the target phenomenon. From a modeling standpoint, every combination of the factors above can be seen as *noise* affecting the data. Precautions in the model formulation process must be taken in order to achieve solutions that are *robust* to the noise effect. To this end, we can couple machine learning methods with *regularization*, a set of techniques that can be introduced independently from the learning machine (Okser et al., 2014). Regularization is of fundamental use not only to achieve robustness in the presence of noise, but also to impose consistence with prior knowledge. We will show in Section 2 that there are different methods to attain either goal, and that they can be combined.

In this review, we describe how regularization can be employed, together with machine learning methods, to successfully address complex life science questions. Unlike previous review papers on this matter (Ma and Huang, 2008; Sohail and Arif, 2020), we provide a vast range of methods incorporating the advances made in the last 10 years of research, and focus on regularization per se and how it has been successfully exploited to answer questions on various types of data including omic-data, imaging data, clinical outcomes, and much more. We provide the reader with a wide and full understanding of possible concerns and situations. More specifically, we identify four families of life science questions which occur regularly and for which regularization techniques are suitable to be employed. Although these do not cover the entirety of all possible questions that can be answered with machine learning techniques, they present some of the most common uses of regularized machine learning in the life sciences.

Such questions are the following: (Q1) How to find relations between input and output from noisy data? (Q2) Which variables are the most relevant? (Q3) Are there hidden patterns in the data? and (Q4) Are there relevant relationships between variables?

Outline In the remainder of the paper we provide background on supervised and unsupervised machine learning (Section 2), focusing on the specific ways of introducing regularization within the different methods. In Section 3 we describe the four main representative questions, and we answer each of them separately in Sections 4, 5, 6 and 7. We conclude the paper with a discussion (Section 8) on the most proper method to employ depending on the type of data, providing a list of use cases per each data type and method.

2 Learning machines and regularization

Life science problems can be tackled with a vast amount of statistical and machine learning methods. Here, we do not want to discuss how to address all the possible problems, but restrict ourselves to those that can be approached with specific regularized methods both in the supervised and unsupervised setting.

2.1 Supervised learning

Supervised learning defines a subset of machine learning methods that allows to study relationships between input-output pairs. In this setting, we denote data as $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^n = (X, \boldsymbol{y})$, where $\boldsymbol{x}_i \in \mathcal{X}$ for $i = 1, \ldots, n$ are collections of samples each sample being a d-dimensional vector of observations on d variables, and $y_i \in \mathcal{Y}$ are the related outcomes. The nature of the output space $\mathcal{Y}$ defines the problem as *classification* if the output is categorical, *e.g.* $\mathcal{Y} = \{a, b\}$ (with $a \neq b$) or *regression* if $\mathcal{Y} \subseteq \mathbb{R}$. Supervised learning methods aim at finding a function of the inputs that approximates the output $y = f(\boldsymbol{x})$ in such a way to be able to predict future data. Note that in the rest of the paper we will mainly refer to the problem of *binary* classification, but the *multi-class* case can be easily substituted (Yuan et al., 2016).

Typically both regression and classification tasks can translate into the optimization of the following problem

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f(\boldsymbol{x}_i), y_i) \tag{1}$$

where $\mathcal{F}$ is the space of possible functions (*e.g.linear* functions such as $f(\boldsymbol{x}_i) = \boldsymbol{w}\boldsymbol{x}_i + b$, where $\boldsymbol{w}$ is a vector of weights) and $L(f(\boldsymbol{x}), y)$ is the *loss function* that measures the adherence of the model to training data. Several loss functions for regression and classification problems have been proposed. Table 1 defines the most commonly adopted. Choosing the appropriate loss function for the problem at hand is crucial and there is no trivial solution for this problem. Different choices for $L(f(\boldsymbol{x}), y)$ identifies different learning machines (Hastie et al., 2009; Bishop, 2006).

2.2 Unsupervised learning

Unsupervised learning defines a subset of machine learning methods that allow to study internal patterns among possibly heterogeneous observations. In this setting, data are $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^n = X$, where each $\boldsymbol{x}_i \in \mathcal{X}$ is a d dimensional vector of observations on d variables. The most common example of unsupervised learning is *clustering*, which aims at grouping the samples such that the variability within a group is less than the variability between groups. This can help in the analysis of possibly multi-class phenomena, where the classes are unknown. Another unsupervised method is *dictionary learning* which is a matrix decomposition method that tries

to decompose the original data matrix X in two, the dictionary that explains patterns of the d variables and the coefficients that allows to reconstruct the original data matrix.

We will also discuss the problem of *network inference*, which is the problem of inferring relationships among variables through observations. Such method address the problem of understanding how the variables in play can describe the system by interacting with each other.

All the methods mentioned above entail the minimization of a loss, depending on the problem at hand the loss may change, we can generally write it as in (1) as

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f(\boldsymbol{x}_i)). \tag{2}$$

Here, $L(f(\boldsymbol{x}_i))$ is a loss function that includes only the data matrix X. Given the wide set of unsupervised methods we do not provide examples of loss functions, specific choices for Dictionary Learning and Network inference are presented in Section 6 and 7, respectively. Note that we are restricting ourselves to unsupervised scenarios where we can perform regularization.

2.3 The problem of overfitting

Learning algorithms are often prone to overfitting, which can be described as the phenomenon where the learned model is more accurate on known data (training) than on unseen data (test). Such a model will explain too precisely the known data fitting noise as well as signal, and therefore losing the ability of generalize on future examples. Overfitting is more prone to happen when learning is performed on a low number of samples, or the complexity of the model is high. Indeed, in the first case we might lose the ability to discern which information is noise and which is relevant; in the second case a high complex model is prone to fitting noise in the training data. Regularization and model selection techniques are the go to tools to prevent overfitting and obtain robust models. These two complementary sets of techniques respectively penalize overly complex models or test the model ability to generalize by evaluating its performance on a set of data not used for training (*i.e.* validation set, a part of the training set left aside for explicit evaluation of generalization properties).

Table 1: Definition of the loss function $L(f(\boldsymbol{x}), y)$ for regression and classification problems. Note that $\mathbb{I}$ denotes the indicator function.

Regression	Square	$(y - f(\boldsymbol{x}))^2$
	Absolute	$\lvert y - f(\boldsymbol{x})\rvert$
	ϵ-insensitive	$\min(\lvert y - f(\boldsymbol{x})\rvert - \epsilon, 0)$
Classification	Zero-one	$1 - \mathbb{I}(y = f(x))$
	Square	$(1 - yf(\boldsymbol{x}))^2$
	Logistic	$\log(1 + e^{-yf(\boldsymbol{x})})$
	Hinge	$\lvert 1 - yf(\boldsymbol{x})\rvert_+$

2.4 Regularization

Given Problem (1) and (2) there are many possible ways of performing regularization both to be robust to noise (*i.e.* prevent overfitting) or impose prior knowledge. They differ in the way they act on retrievieng the optimized solution: they can act on the model, on the optimization technique, or on the data.

Addition of a penalty This type of regularization acts on the model and is based on the addition of a penalty term to Problem (1), as follows:

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(f(\boldsymbol{x}_i), y_i) + \lambda\, R(f). \tag{3}$$

The term $R(f)$ is known as the *regularization penalty* and, depending on how it is defined, can impose stability on the expected function or prior knowledge on the problem (Tikhonov, 1963). With different choices for $R(f)$, different effects on the solution may be achieved. We will briefly discuss the effect of some choices, like Tikhonov, Lasso, Group Lasso, Elastic net and more in Section A1 and A2.

The scalar λ is the *regularization parameter* which controls the trade-off between the loss and the penalty terms. The addition of a penalty is related to the idea of adding a prior in Bayesian learning. Indeed, both techniques use prior knowledge or assumptions about data in order to guide the inference (Murphy, 2012, Chapter 7).

Ensemble techniques Another way of avoiding overfitting is to combine a finite set of alternative models in order to allow for higher flexibility and thus better performance. Typical ensemble techniques are *bagging* and *boosting*. The first two act on the data and involve multiple

models trained on random subsets of the input samples. They yield the final prediction by merging the predictions of the models that equally concur to the final solution. When using this approach as a regularization strategy, one must be careful to select the right number of models to learn as well as their complexity or overfitting might still occur. *Boosting* is an ensemble method that acts on the optimization process by performing predictions by sequentially fitting several base learners that cast a weighted vote (Freund, 1995). At each boosting iteration, the model is forced to learn the relationships between input and output that were previously missed as the weights corresponding to poorly predicted samples increase. From a theoretical standpoint, it is possible to boost any learning machine, nevertheless boosting methods are truly beneficial only when based on weak learners, such as stumps or linear regression (Hastie et al., 2009) — stumps are one node decision trees (Iba and Langley, 1992). Examples of these techniques are *Random forest* and *Gradient boosting*, which we will discuss in Section 4.

Dropout and data augmentation These two regularization techniques are mostly used for Neural Networks. The first one, *Dropout* (Srivastava et al., 2014) is a technique that acts on the model by temporarily deactivating a defined number of randomly chosen units of the network at training phase. This reduces the degrees of freedom of the model and it implicitly allows to achieve an ensemble of several smaller networks whose predictions are combined. *Data augmentation* acts on data as it is a pre-processing technique. It is typically used when dealing with Neural Networks and images and it consists in expanding an input data set by applying transformations as scaling or translation on the available samples. Hernández-García and König (2018) show evidence of how this method can be understood to achieve regularization as it avoids overfitting as more explicit regularization techniques.

Early stopping This is a popular regularization strategy (Prechelt, 1998), that consists in interrupting the fitting process as soon as the error on an external validation set increases (Angermueller et al., 2016). This type of regularization acts on the optimization procedure and it is typically employed on iterative methods such as gradient descent. It is based on the idea that, given a set of data on which we train the model (*training*) and a set on which we validate it (*validation*), the optimization procedure minimizes the error both for the training and the validation up to a point after which the validation error starts increasing as the model overfits

the training data.

2.5 Model selection

Each of the aforementioned regularization techniques has an intrinsic parameter that needs to be tuned. For the penalized methods we have λ, for ensemble learning we have m, the number of models, for early stopping we have the patience, i.e.: the number of iterations we allow our model not to improve its training loss, for dropout the number of units to deactivate and finally, for data augmentation, the number of data samples to add. The choice of the best parameters is crucial to achieve accurate prediction along with good generalization properties (Hastie et al., 2009).

This problem is typically referred to as *model selection*. It must be distinguished from *model evaluation*, which aims at estimating the generalization error of the chosen model on new data.

Model selection is usually performed by estimating, for a given value of a parameter, the prediction error. The simplest and most widely used method for estimating the prediction error of the model is to perform K-fold cross-validation. Given an integer K, we split the data in K parts of approximately the same size. For each of these parts in turn, we compute on the k-th part the error of the model fitted to the $K-1$ other parts. Finally, the mean prediction error on the K parts is computed.

This procedure is repeated for a certain range of parameters values, the best parameter is selected as the one that returns the lowest prediction error in average. Many other cross-validation routines are proposed in literature, we refer to Molinaro et al. (2005) for a detailed description of the most important cross-validation strategies.

In contrast with cross-validation, multiple methods have been developed to perform an analytical estimation of the prediction error of a model. Some of the most widely used of these methods are the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) (Vrieze, 2012). Both methods are based on the idea of minimizing the loss function (or maximizing the likelihood) while penalizing such quantity depending on the degrees of freedom of the problem. As an example, consider the well-known clustering method K-means, which divides the data points in K clusters. For K equal to the number of samples we would reach a perfect fit in terms of value of the loss function, but this would overfit on the samples. Thus,

using methods as AIC or BIC we add to the error a penalty proportional to the value K, in order to obtain balance between the error and the number of degrees of freedom of the problem.

3 From biological questions to learning tasks

In applied life science, it is crucial to choose the right approach to not incur bias and obtain robust results.

We identified four recurring biological questions that, even though they do not completely cover the complex variety of problems related to life science data, are the most amenable to regularized learning techniques. We provide in Figure 1 a schematic explanation of how to reach a particular question starting from the data and the problem at hand.

Q1: How to find relations between input and output from noisy data? Starting from a collection of input measures that are likely to be related to a certain output (*e.g.*, some pathological phenotype), a typical final goal is to develop a model that represents the relationship between input and target. Many possible examples of this type of problem exist, for instance in molecular (Angermueller et al., 2016; Okser et al., 2014) or radiomics/imaging studies (Min et al., 2016). Biological questions of this class are usually approached with supervised learning models. In the context of life science studies, where the available data are often scarce and noisy, models can suffer from overfitting. Therefore, the use of appropriate regularization strategies is recommended. We provide a list of suitable methods to address this problem in Section 4.

Q2: Which variables are the most relevant? A complementary question revolves around the interpretability of the predictive model. In particular, when dealing with high-dimensional biological data, the main goal can be to identify a relevant subset of meaningful variables for the observed phenomenon (Tang et al., 2017; Climente-González et al., 2019). This may improve prediction power as well as promote model interpretability, *i.e.* the ability of understanding and interpret the parameters of the inferred model in order to extract new biological knowledge from the analyzed data. Thanks to their flexibility, sparse regularization methods have been effectively used in biological contexts, dealing with high-throughput data (Silver et al.,

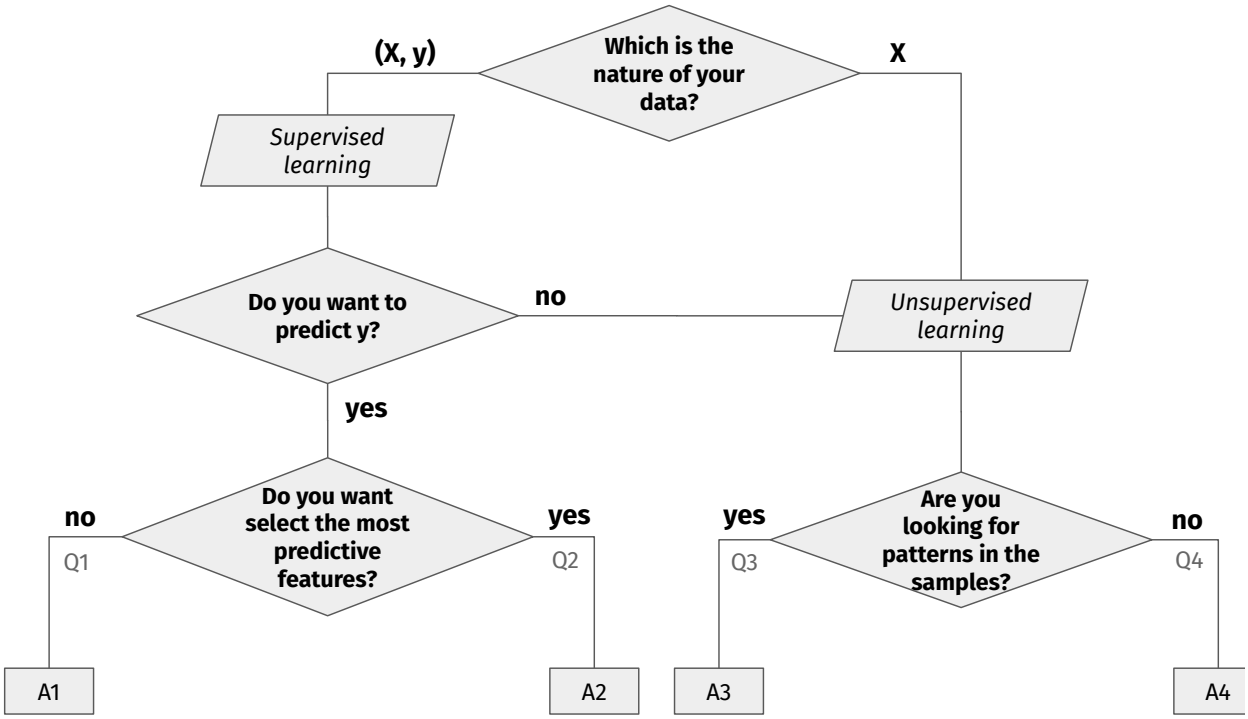

Figure 1: Flux diagram explaining how to reach a specific question. In practice we first need to distinguish if we have labelled data or not, in the first case we are in a supervised learning setting while in the second we are in an unsupervised setting. In the supervised setting we want to predict the labels, and we can simply do this in the best possible way or we may ask which are the best variables to predict. In the unsupervised setting, we can look for patterns in the samples or for relationships among the features.

2013; Mascelli et al., 2013; Giraud, 2014) From a methodological standpoint, this topic will be introduced in Section 5.

Q3: Are there hidden patterns in the data? Often, we observe a phenomenon that does not necessary have a related outcome. We observe components of the phenomenon and we want to understand whether there are underlying hidden repeated patterns. One very common way of looking for patterns in the observations is to *cluster* them, by aggregating the observations that are most similar to each other under the definition of some measures. Nonetheless, clustering is an approach typically performed on the samples and thus, it does not provide further insights on the features values. Often we recur to *matrix factorization* that simultaneously provides a

new data-driven representation of the data while also giving intuition of the underlying patterns (Alexandrov et al., 2013) From a methodological standpoint, this topic will be introduced in Section 6.

Q4: Are there relevant relationships between variables? Another common problem which arises in data analysis is how the measured variables are related to each other, or in other words how they interact. The study of these interactions can present different patterns across samples. Indeed, searching for complex patterns in the data may offer insights on the behavior of variables in diverse contexts, such as diverse biological conditions in biomedical studies. Interactions are usually modeled as a network (or graph), *i.e.,* a set of variables (nodes) connected with each other based on a particular type of relation (links). The graphical modeling of the variables offers a compact and efficient representation which helps to identify the variability patterns in the data (Monti et al., 2014). An overview of this class of methods will be provided in Section 7.

In all these questions regularization plays a key role either for robustness to impose prior knowledge on the solution. The regularization schemes presented in the previous section can be used in different ways to address all these questions, sometimes combined and sometimes alone.

4 How to find relations between input and output from noisy data? (A1)

This problem lies in the macro-category of supervised problems and it is one of the most largely discussed. We provide a variety of well known techniques that differ both in the way they approach regularization as well as the type of data they can handle.

4.1 Tikhonov regularization

This regularization strategy is based on the addition of an ℓ_2-norm penalty, that can be employed when the function $f(\boldsymbol{x})$ is linear in $\boldsymbol{x}$ (Tikhonov, 1963).

$$R_{\ell_2}(\boldsymbol{w}) = \sum_{j=1}^{d} (w_j)^2 = ||\boldsymbol{w}||_2^2 \tag{4}$$

This penalty shrinks the coefficients toward zero, but it does not achieve a parsimonious representation, as it tends to keep all the variables in the model. This penalty is typically applied to the square loss, thus taking the name of *Ridge regression* (Hoerl and Kennard, 1970) but it is known under several different names, among which we recall, *weight decay* (Krogh and Hertz, 1992) and *Regularization Network* (Evgeniou et al., 2000). t is easy to show that Ridge regression is equivalent to a Bayesian approach to linear regression where we impose a Normal prior on the regression coefficients (Murphy, 2012, Chapter 7).

Applications This model is successfully applied in a variety of biological studies mainly involving regression problems. For instance, in Kratsch and McHardy (2014) the authors propose a Ridge regression-based method to estimate the trees of mutations within a species from the ancestors of the species to the present, while in Bøvelstad et al. (2007) this technique is used to predict survival of patients from gene expression data. Tikhonov regularization can also be combined with other types of regularization as in Fiorini et al. (2017) where they exploit the addition of a Nuclear Norm penalty to perform temporal prediction of possible responses of patients affected from Multiple Scleroris.

4.2 Random Forests

RFs are ensembles of decision trees, each grown on a subset of samples randomly chosen with replacement from training data. Decision trees are interpretable models where each node can be seen as a particular *question* on a single feature that leads to partition the training data into subsets. The feature that yields the best split in terms of a pre-selected metric is chosen to create a new node — we refer to Qi (2012) for possible choices of such metric that are suitable for different biological problems. Each path from root to leaf is called classification rule.

Decision trees alone tend to not perform well, which led to the introduction of random forests in 2001 (Breiman, 2001). The final prediction is made by aggregating the prediction of m trees,

either by a majority vote in the case of classification problems, or by averaging predictions in the case of regression problems. Several techniques for applying regularization to random forests have been proposed. These techniques broadly fall under two categories: (1) cost-complexity pruning, which consists in limiting tree depth, resulting in less complex models (Kulkarni and Sinha, 2012); and (2) Gini index penalization, which weights the probabilities of each class in order to favor large partitions (Liu et al., 2014a).

Applications Random forests can handle both numerical and categorical variables, multiple scales, and non-linearities. This makes them popular for the analysis of diverse types of biological data, such as gene expression, sequencing, GWAS (Genome-Wide Association Study) or mass spectrometry data. A detailed review specific to random forest is provided in (Qi, 2012). Deng and Runger (2013) and Kursa (2014) use regularized and robust random forest for the selection of genes in classification tasks. Random forests can be used also for regression, as in Johann et al. (2019) where the authors aim at quantify tumor purity or, for learning interactions between non coding RNA and messenger RNA (Soulé et al., 2020).

4.3 Gradient Boosting

Gradient boosting is an ensemble method that performs predictions by sequentially fitting several base learners that cast a weighted vote (Freund, 1995). At each boosting iteration, a new model is created by giving increasing weight to the errors made by previous models, so that each model is forced to learn the relationships between input and output that were previously missed as the weights corresponding to poorly predicted samples increase. From a theoretical standpoint, it is possible to boost any learning machine, nevertheless boosting methods are truly beneficial only when based on weak learners, such as stumps or linear regression (Hastie et al., 2009). Gradient boosting (Friedman, 2001) is one of the most widely applied boosting methods in biological problems.

Gradient boosting has several desirable properties (Mayr et al., 2014), such as its capability to learn nonlinear input/output relationship, its ability to embed a feature importance measure and its stability in case of high-dimensional data (Buehlmann, 2006).

Boosting methods may suffer overfitting. The main regularization parameter to control is the number of boosting iterations m, *i.e.,* the number of base learners, fitted on the training

data. Careful consideration should also be put on tuning the complexity of the base learners that are used.

Applications Approaches based on gradient boosting classification are used to detect *de novo* mutations showing an improved specificity and sensitivity with respect to state-of-the-art methods (Liu et al., 2014b). When combined with stability selection (Meinshausen and Bühlmann, 2010), gradient boosting has demonstrated to be a very resourceful method for variable selection, leading to an effective control of the false discovery rate. This strategy was followed to associate overall survival with single-nucleotide polymorphisms of patients affected by cutaneous melanoma (He et al., 2016) and to detect differentially expressed amino acid pathways in autism spectrum disorder patients (Hofner et al., 2015).

4.4 Deep learning

Deep Learning (DL) methods are a broad class of machine learning techniques that, starting from raw data, aim at learning a suitable feature representation (see Section 7) and a prediction function, at the same time (LeCun et al., 2015). DL methods can be seen as an extension of classical Neural Networks (NN), where the final prediction is achieved by composing several layers of nonlinear transformations. DL architectures can be devised to tackle binary/multi-category classification (Angermueller et al., 2016; Leung et al., 2014) as well as single/multiple-output regression tasks (Chen et al., 2016).

Particular attention must be paid when fitting deep models as they can be prone to overfit the training set (Angermueller et al., 2016). This is particularly true in healthcare contexts in which the available data set dimension can be small. Regularization in DL methods can be achieved by penalizing the weights of the network. The most common regularization strategy consists in adding an ℓ_2-norm penalty in the objective function, as in Equation (4). In the DL community this procedure is known as weight decay (Krogh and Hertz, 1992). Although less common, the ℓ_1-norm can also be adopted as regularization penalty, as in (Leung et al., 2014).

Applications Deep learning can be regularized in many different ways. For example weight decay is adopted in (Chen et al., 2015) to train a deep architecture on rat cell responses to given stimuli with the final aim to predict human cell responses in the same conditions. More-

over, weight decay is also adopted in (Yuan et al., 2016) to train *DeepGene*, *i.e.,* a simple fully connected network known as multi layer perceptron (LeCun et al., 2015), which is designed to classify the tumor type from a set of somatic point mutations. Furthermore, weight decay is used in (Fakhry et al., 2016) to train a DL architecture for brain electron microscopy image segmentation. Although less common, the ℓ_1-norm can also be adopted as regularization penalty, as in (Leung et al., 2014).

these methods iteratively update the weights of the network in order to decrease the training error. The use of dropout alone can improve the generalization properties, as in (Chen et al., 2016), where the authors propose *D-GEX*, DL regression architecture trained to predict the expression of a number of target genes. Dropout can also be used in combination with weight decay or other forms of regularization, as in (Leung et al., 2014), where the authors propose to use a deep network to achieve splicing pattern prediction. Dropout is combined with early stopping in Fiorini et al. (2019) where they use textual representation of medical prescriptions to classify the patients that would likely worsen their diabetes in the future. DL methods are nowadays becoming a standard for most biomedical imaging applications. In such context regularization plays a key role, as it allows to learn robust models for automatic image retrieval, segmentation and disease prediction. One of the main drawbacks of DL methods is that, in order to learn a prediction function that does not simply overfit the training set, the number of training data should be *large* (*e.g.* in the order of tens of thousands). In the context of biomedical images, retrieving a large dataset may be hard. To cope with this issue we can employ data augmentation (Schlemper et al., 2017). An interesting property of DL architectures is that, when properly trained on a given collection of images, they can learn both specific and a specific features. So, in general, it is possible to re-use (or fine-tune) the weights learned by a network from some dataset, to another case. This strategy is known as *transfer learning* and, among others, it was successfully exploited by (Li et al., 2018) to classify subjects with autism spectrum disorder from medical images. As transfer learning helps to prevent overfitting it can be considered, to some extent, a regularization strategy.

For a complete review on the impact of DL on this subject we refer to (Lundervold and Lundervold, 2019). When model intepretabilty is as important as prediction performance, DL methods must be trained with particular care. This relevant topic is addressed in (Plumb et al.,

2019), where the authors propose a regularization term that encourages explainability of the trained model in the neighborhood of the training points without significantly affecting the predicting performance. On the same line, (Tong et al., 2018) recently introduced the so-called *Graph Spectral Regularization* that, applied on neurons activations of an arbitrary NN, can be used to enforce a meaningful graph structure. This method is successfully applied to learn gene markers correlations in a single-cell RNA-sequencing dataset. For a specific review clarifying the role of DL in biology we refer the reader to (Ching et al., 2018), where the authors analyze the application of DL to many tasks among which clinical outcome forecasting, biological processes, treatment discovery and neuroscience.

5 Which variables are the most relevant? (A2)

When dealing with health science problems, often we want to learn which are the best predictors for a certain outcome. Typically, the regularized solution to this problem is to add sparsity-inducing penalties on the loss of the specific machine learning method. A model is said to be *sparse* when it is defined upon a small number of features (Hastie et al., 2015).

5.1 Lasso and Elastic-net

There are many penalties that can be added to enforce sparsity. All these penalties are based of the *Lasso* (Tibshirani, 1996) penalty or ℓ_1-norm:

$$R_{\ell_1}(\boldsymbol{w}) = \sum_{j=1}^{d} |w_j| = ||\boldsymbol{w}||_1. \tag{5}$$

Sparsity can also be achieved through other feature selection techniques besides regularization. Those include filtering techniques, which score features according to their individual relationship to the outcome (for example, through correlations or statistical association testing) and only keep the highest-scoring ones, or wrapper techniques, which assess subsets of variables according to their usefulness to a given learner. By contrast, embedded methods such as the Lasso directly satisfy the sparsity constraint while optimizing the model, which is more efficient. All three family of approaches are reviewed in Guyon and Elisseeff (2003).

As for the ℓ_2 regularization, the Lasso has an equivalent under the Bayesian setting, and

corresponds to using a Laplace prior on the weights of the predictors (Murphy, 2012). When used for variable selection, the Lasso has two major drawbacks. First, in presence of groups of correlated variables, this method tends to select only one variable per group. Secondly, the method cannot select more variables than the sample size (Waldmann et al., 2013; De Mol et al., 2009b).

The Elastic-Net (De Mol et al., 2009a; Zou and Hastie, 2005) method can be formulated as a least square problem penalized by a convex combination of the Lasso (ℓ_1) and the Ridge regression (ℓ_2) penalties (Equation (6)).

$$R_{\ell_1\ell_2}(\boldsymbol{w}) = \sum_{j=1}^{d}((1-\alpha)\ |w_j| + \alpha\ w_j^2) = (1-\alpha)\ ||\boldsymbol{w}||_1 + \alpha\ ||\boldsymbol{w}||_2^2 \tag{6}$$

The combined presence of the ℓ_1- and ℓ_2-norms promotes sparse solutions where groups of correlated variables can be simultaneously selected. It is easy to see that fitting the Elastic-Net model for $\alpha = 1$ or $\alpha = 0$ is equivalent to Tikhonov or Lasso regularization, respectively.

Applications A popular application of the Lasso is to perform shrinkage and variable selection in survival analysis for Cox proportional hazard regression and additive risk models. Such penalized methods were extensively applied in literature to predict survival time from molecular data collected from patients affected by different kinds of tumor (Tang et al., 2017; Ma and Huang, 2007). The Elastic-Net method is successfully applied in several biomedical fields (Waldmann et al., 2013). For example, De Mol et al. (2009b) exploited an incremental version of elastic net to identify nested groups of correlated genes and Hughey and Butte (2015) exploit it to distinguish between four lung cancer subtypes.In (Csala et al., 2017) the authors propose an iterative algorithm that exploits the variable selection capabilities of this method to estimate explanatory variables weights in order to explain the variability in gene expressions by epigenomic data (*i.e.,* methylation markers) collected from blood leukocytes of Marfan Syndrome patients.

5.2 Lasso extensions

It is also possible to design regularizers that force the features that are assigned non-zero weights to follow a given underlying structure (Micchelli et al., 2013). This structure can be defined by

arranging features in *groups* (typically for bioinformatics applications, biological pathways) or *graphs* (typically, biological networks). In the case of groups, the regularizer constrains entire groups of features to be either all selected or all discarded. When the groups are disjoint, this can be implemented by the group Lasso (Yuan and Lin, 2006). Suppose that the d features are grouped into L groups, with d_l the number of features in group l. Let us denote by $X_l \in \mathbb{R}^{n \times d_l}$ the input data restricted to the features belonging to group l. The Group Lasso uses the following penalty:

$$R_{\text{gl}}(\boldsymbol{w}) = \sum_{l=1}^{L} \sqrt{d_l} ||w_l||_2 \tag{7}$$

where the same weight w_l is associated to all variables from group l. The Group Lasso was later extended to the case where the groups can overlap (Jacob et al., 2009) or be hierarchical (Jenatton et al., 2011)

In the case of networks, the regularizer encourages features that are connected on the network to be selected together. This can be implemented directly with the overlapping group Lasso, by defining groups as pairs of features connected by an edge (Jacob et al., 2009). Another way to smooth regression weights along the edges of a predefined network, while enforcing sparsity, is a variant of the generalized fused Lasso (Tibshirani et al., 2005). The corresponding penalty is given by Equation (8)

$$R_{\text{gfl}}(\boldsymbol{w}) = \sum_{p \sim q} |w_p - w_q| + \eta ||\boldsymbol{w}||_1, \tag{8}$$

where η is a regularization parameter. We use the notation $p \sim q$ to denote that vertex p and vertex q form an edge in the graph considered. However this can get computationally intensive in the case of large networks, and other methods based on graph Laplacians have been developed. Given a graph G of adjacency matrix $A \in \mathbb{R}^{d \times d}$, the Laplacian of G is defined as $L := D - A$, where D is a $d \times d$ diagonal matrix with diagonal entries $D_{ii} = \sum_{j=1}^{d} A_{ij}$. The graph Laplacian is analog to the Laplacian operator in multivariable calculus, and similarly measures to what extent a graph differs at one vertex from its values at nearby vertices. Given a function $f : \mathbb{R}^d \mapsto \mathbb{R}$, $f^\top L f$ quantifies how *smoothly* f varies over the graph (Smola and Kondor, 2003). Grace (Li and Li, 2010) uses a penalty based on the graph Laplacian L of the biological network, which encourages the coefficients $\boldsymbol{\beta}$ to be smooth on the graph structure. This regularizer is given by Eq.(9). The aGrace variant (Li and Li, 2010) allows connected

features to have effects of opposite directions.

$$R_{\text{grace}}(\boldsymbol{w}) = \boldsymbol{w}^\top L \boldsymbol{w} = \sum_{p,q} A_{pq}(w_p - w_q)^2 \tag{9}$$

These approaches are rather sensitive to the quality of the network they use, and might suffer from bias due to graph misspecification (Yang et al., 2012b). GOSCAR (Yang et al., 2012b) was proposed to address this issue, and replaces the term $|w_p - w_q|$ in Eq.(8) with a non-convex penalty: $\max(|w_p|, |w_q|) = \frac{1}{2}(|w_p + w_q| + |w_p - w_q|)$.

Applications Hierarchical Group Lasso was used in a classification setting to localize the brain regions involved in the processing of visual stimuli from fMRI (Jenatton et al., 2012). In Xin et al. (2014) the authors successfully applied network Lasso to Alzheimer's disease diagnostics from brain images. A more detailed review of these approaches and their applications to bioinformatics problems can be found in (Azencott, 2016), which also presents how these regularizers can be used in the context of filter approaches to feature selection.

5.3 Evaluation

As for the other methods presented in this review, we need to perform model selection also when utilizing the penalties described in this section. Nonetheless, when adopting sparse techniqes it is necessary to evaluate if the model recover sthe correct features. In bioinformatics, there usually is no ground truth for this question, which can hence only be answered on synthetic data: if the feature selection process is stable it should retrieve the same features on overlapping subsets of the same data set.

The set of selected features can only be interpreted if it remains robust to slight variations in the data. Do multiple repeats of the algorithm, for instance on cross-validation training folds, yield the same sets of features? A variety of measures have been developed to evaluate the stability of a feature selection algorithm.

while predictivity is typically assessed by cross-validation (Guyon et al., 2002). It is important to highlight that variable/feature selection should not be considered as a pre-processing step. In fact, using the same dataset to select the most important features and to evaluate the model performance leads to an over-optimistic predictive capability. This phenomenon is known as *selection bias* (Ambroise and McLachlan, 2002).

6 Are there hidden patterns in the data? (A3)

Pattern recognition is a very general machine learning problem that comprehends tasks as clustering of samples or retrieval of basic signals within the features. Nonetheless, in life science settings while it is useful to obtain information on samples (typically patients), it may also be useful to retrieve patterns from the features. Using clustering methods in these settings will be harder as they typically assume samples that belong to the same cluster to be i.i.d. Features, on the other hand, may have complex dependency patterns difficult to interpret with standard clusering algorithms. In signal analysis the possibility to detect latent patterns present in sampled signals has been studied in deep for the possibility to obtain a better representation of data. The most common ways to decompose a signal are Principal Component Analysis (PCA) (Wold et al., 1987) and its derivatives. Nonetheless, they typically assume strong prior on the patterns, for example in PCA all the patterns have to be orthogonal to each other. In some context this assumption can prevent the analysis to detect factors which do not satisfy the requirements imposed.

6.1 Dictionary learning

We therefore discuss a technique called matrix factorization, which given an input matrix $\mathbf{X}$ of n signals in d dimensions aims at decomposing it into two (or more) sub-matrices, one representing the patterns of features *dictionary* and one the *coefficients*. The original samples are obtained as linear combination of the atoms weighted by the coefficients; if the combination has few non-zero coefficients we have a *sparse coding* (Olshausen and Field, 1997). The dictionary learning problem, without regularization can be written as

$$\min_{\mathbf{C}\in\mathcal{C},\mathbf{D}\in\mathcal{D}} \quad \|\mathbf{X}-\mathbf{C}\mathbf{D}\|_F^2 \tag{10}$$

where $C \in \mathbb{R}^{n\times k}$ is the matrix of coefficients, $\mathbf{D} \in \mathbb{R}^{k\times d}$ is the dictionary matrix and the two convex sets $\mathcal{D}$ and $\mathcal{C}$ can be used to constrain the solution to specific sets. The number k is the number of atoms of the problem and it is a parameter that need to be found through cross-validation techniques.

We can assume that the dictionary is known *a priori*, mimicking signal decomposition

techniques as Fourier transform or Wavelet transform. In this case the problem is called sparse coding and it is a convex problem. In general, we do not know the underling patterns and we therefore need to learn the dictionary too.

This type of techniques allows to perform a variety of different tasks as clustering, dictionary learning, sparse coding, data integration, matrix completion and others. This methods can be regularized through the addition of a penalty both on the patterns or the coefficients

$$\min_{\mathbf{C}\in\mathcal{C},\mathbf{D}\in\mathcal{D}} \|\mathbf{X}-\mathbf{D}\mathbf{C}\|_F^2 + R_1(\mathbf{C}) + R_2(\mathbf{D}). \tag{11}$$

where R_1 and R_2 are penalties chosen by the user to impose regularization. Common choices are R_{ℓ_1} and R_{ℓ_2}. It is often associated to bagging techniques to prevent overfitting on the data and the initialization as, in case of learning both the dictionary and the coefficients it is a non-convex problem.

Applications Dictionary learning is widely used to analyze biological data, in particular it is mostly exploited for the analysis of biomedical images. It was exploited for the reconstruction of Magnetic Resonance images from under-sampled data (Ravishankar and Bresler, 2010), and also for the detection of microaneurysm in retinal images Zhou et al. (2017). Dictionary learning can be also used for other type of data, as in Nowak et al. (2011); Masecchia et al. (2013) where they use a fused Lasso Dictionary learning approach to perform subtyping of cancer patients analysing Copy Number Variation (CNV) data.

6.2 Non-negative matrix factorization and discriminative dictionary learning

The dictionary learning problem allows to be specialized in many forms. One of the most popular specialization is the so-called *Non-negative matrix factorization* which has the same exact form of Problem (11) but the sets in which we are optimizing the coefficient and the dictionary are restricted to the positive space with $\mathcal{D} = \mathcal{C} = \mathcal{R}_+$. This approach was first proposed in (Lee and Seung, 2001) and it is widely used in biological applications as, the main assumption is that, natural signals cannot typically derive from negative patterns, where we define signal as the measurable expression of the system under analysis. Imposing a non-

negativity constraint forces the algorithm to detect only positive patterns as well as positive weights thus reducing cancellation effects (Lee and Seung, 2001).

The second problem is *discriminative dictionary learning* where the coefficients are used as a new representation for the original signal in a new problem such as classification or regression. The possibility to learn the dictionary, the coefficients and the classification parameters at the same time was first proposed by Huang and Aviyente (2007). In this specialization the functional becomes

$$\min_{\mathbf{C}\in\mathcal{C},\mathbf{D}\in\mathcal{D},\mathbf{w}\in\mathbb{R}^k} \|\mathbf{X}-\mathbf{C}\mathbf{D}\|_F^2 + L(\mathbf{y},\mathbf{w},\mathbf{C}) + R_1(\mathbf{C}) + R_2(\mathbf{D}) + R_3(\mathbf{w}) \tag{12}$$

where L is a classification/regression loss as the ones in Table 1 and $R_1,\ R_2$ and R_3 are penalties as for Equation (11).

Applications In Piaggio et al. (2019) they exploit penalized non-negative matrix factorization to find patterns of somatic mutations specific of uveal melanoma from SNPs data. In Javidi et al. (2017) they exploit discriminative dictionary learning and sparse representation based on Lasso penalty to to perform vessel segmentation on retinal images. In Li et al. (2017) they employ multi-modal dictionary learning with lasso penalty to distinguish between stages of Alzheimer's disease.

7 Are there relevant relationships between variables? (A4)

Network inference is the process of estimating a graph from real world measurements. The inferred graph is the mathematical abstraction of a system where nodes represent the variables and edges may represent different types of relations according to the system under analysis. Often, in real world scenarios, the graph structure is not known and, in fields such as computational biology, network inference plays a key role in understanding how molecular interaction works. At the cellular level, for example, we may seek for evidence of regulatory functions (Lozano et al., 2009), co-expression edges, metabolic influence (Kanehisa, 2001) as well as protein-protein interaction networks (Huang et al., 2016). Learning the network struc-

ture from data may be hard due to the ratio between number of features and samples. The research in this area has increased in the last years and many methods that tackle some of these problems have been proposed. These methods include Bayesian Networks (BN) (Nielsen and Jensen, 2009), Gaussian Graphical Models (GGMs), Differential Equation (DE) and (de Hoon et al., 2002), Mutual Information (MI) (Margolin et al., 2006) based methods. In this section we focus on GGMs as a specific example of a wider sets of probabilistic methods that naturally leverage regularization to infer networks. GGMs are based on penalized Maximum Likelihood Estimation (MLE) and can be written as in Equation (3). GGMs can also easily be adapted to many different regularization strategies. Regularization in these methods helps to cope with high-dimensionality of the data and identifiability and interpretability of the resulting network. Moreover, GGMs are suited to both the inference of co-expression (Friedman et al., 2008) and regulatory networks (Krämer et al., 2009). This class of methods can also be easily adapted to non-Gaussian data through appropriate data manipulation.

7.1 Graphical Lasso

Graphical Lasso is the most representative example of penalized MLE method for network inference. It assumes the variables in the system to be distributed according to a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma)$. The problem translates in inferring the connections between the variables. The Gaussian assumption simplifies this inference as the connections can be read in the *precision matrix*, *i.e.,* the inverse of the covariance matrix $\Theta = \Sigma^{-1}$. Indeed, two variables i and j are conditionally independent, given the other variables, if and only if $\Theta_{ij} = 0$. Therefore, the precision matrix can be seen as the adjacency matrix of a graph. Another main assumption is that the underlying network is sparse, *i.e.,* only few edges are necessary to fully describe the system. Graphical Lasso (Friedman et al., 2008) can be formalized as:

$$\underset{\Theta}{\text{minimize}} \quad \text{trace}(S\Theta) - \text{logdet}(\Theta) + \lambda \left\|\Theta\right\|_{od,1}, \tag{13}$$

where $\left\|\cdot\right\|_{od,1}$ is the off-diagonal ℓ_1-norm, promoting sparsity in the off-diagonal part of the precision matrix, S is the empirical covariance matrix and the terms trace and logdet derive from the computation of the Gaussian log-likelihood. Equation (13) can be solved using a modified Lasso regression on each variable in turn (see Section 5) with a simple, efficient and fast procedure (Friedman et al., 2008). This is for instance the case of (Menéndez et al.,

2010), where the authors exploit this method to reverse engineer five gene regulatory networks within the context of DREAM4 challenge[1]. It is easy to modify the algorithm to have specific penalties λ_{ik} for each edge. A value $\lambda_{ik} \to \infty$ forces nodes x_i and x_j to be disconnected. This is particularly relevant in biology, when two variables (such as genes) are known not to interact directly. It is worth mentioning that the l_1 norm helps both in terms of understandability and identifiability of the result. Nevertheless, often the final graph may present some differences under different sub-sampling of the data as it is extremely data dependent. In (Liu et al., 2010) the authors suggest a method to select the regularization parameter λ based on the stability of the result under many sub-sampling of the data that was proved effective in many contexts.

Applications An example is the work proposed in (Ramanan et al., 2016) where the authors inferred a network demonstrating an antagonistic relationship between Clostridiales and Bacteroidales communities from the Human Microbiome Project. Since it was first proposed the Graphical Lasso has received much attention for its application in biology, we refer the reader to this review (Kuismin and Sillanpää, 2017) that compares it with other network inference methods in the context of system biology.

7.2 Graphical Lasso extensions

Many extensions of Equation (13) were proposed over the years to model systems of increasing complexity. These extensions are widely based on the addition of further penalties that force the graph structures to respect certain constraints. One notable example is the extension to the multi-task/multi-class case in which the graphs share a common structure but they differ in some connections (Danaher et al., 2014). These methods are mainly based on the group Lasso or fused Lasso penalties and they were succesfully applied in genomics (Xie et al., 2016) and neuroscience (Belilovsky et al., 2016). To include the dynamical properties of systems, (Zhou et al., 2010) propose a weighted method to estimate the graph temporal evolution. Whereas (Hallac et al., 2017) propose evolving precision matrices in time, similarly to (Danaher et al., 2014). Here, again, the extension is performed by applying a regularization term that enforces similarities between graphs close in time. The graphical Lasso has also been extended in order to consider hidden and unmeasurable variables that influence the system through the nuclear

[1] Source: `http://dreamchallenges.org`

norm penalty (Chandrasekaran et al., 2010). The dynamical and latent aspects were fused together in (Tomasi et al., 2018) where the authors show the ability to detect perturbation in cellular system subject to external stimuli.

Graphical Lasso can be further extended to consider the multi-layer case, which integrates components of the cellular system that can act at different scales or time in order to obtain a more precise overview.

Applications In Cheng et al. (2017) they propose a regularized extension that translates into a group Lasso penalty on the entries of the precision matrix. This method is able to detect pathway-pathway and gene-gene interactions. Monti et al. (2014) employed a dynamical Graphical Lasso to detect brain functional connections from fMRI images. Libbrecht et al. (2015) performed semi automated genome annotation by inferring a network with graph-based regularization.

7.3 Lasso in the non-Gaussian case

The Gaussian assumption allows to provide easy and computationally tractable algorithms and extensions but it imposes a limitation in the type of data that can be analyzed. Several methods consider non-Gaussian data distributions simply manipulating the input data through $\log_2$ or copula transforms (Liu et al., 2012).

Applications Research has also moved towards the use of other distributions and models, *e.g.* the Ising model for discrete variables or the Poisson model that provides a better modeling of next generation sequencing data (Yang et al., 2012a). These methods are powerful and they allow to consider graphs, for example gene-gene interactions, that are generated from different data measurements as copy number variation, gene expression or single nucleotide polymorphism data. In this context, a method that integrates the network obtained from diverse measurements assuming the best distribution has been proposed in (Žitnik and Zupan, 2015) where they showed that it allows to recover a more detailed network. Zitnik and Leskovec (2017) exploit a similar method to perform prediction of multi-cellular function by inferring multi-layer tissue networks regularized throguh ℓ_2-norm.

8 Conclusion

This paper clarifies the importance of regularized methods for life science studies from different perspectives. We covered both supervised settings, where the expected outcome is to predict some target variable, as well as unsupervised scenarios, where the aim is to infer the topology of the network modeling the interactions between the observed variables. Moreover, we showed how prior knowledge on the problem at hand can be embedded into a regularization penalty, allowing to identify meaningful and interpretable solutions. Moreover, we also highlighted how, thanks to different regularization penalties, it is possible to overcome the issues faced by standard statistical methods in settings where the amount of variables outnumbers the available samples ($n \ll p$).

We summerized the applications cited in the paper in Table 2 and 3. We highlighted that regularization is heavily employed for the analysis of omic-data (Table 2), which is due to the natural high-dimensionality of these types of data. Furthermore, we cannot identify one specific type of method or regularization type that is more used in general for omic-data. Indeed, the choice of regularization method depends on a variety of additional considerations. In Table 3 we report other types of data; a clear preference for Deep Learning and Dictionary Learning emerges when it comes to the analysis of biomedical images. Such behavior is expected, indeed both Deep Learning and Dictionary Learning learn representations of meaningful parts of the input signal, which is crucial in image analysis as we may want the model to have suitable properties, *e.g.* translation-invariance.

Regularization is a key aspect in all these works, and in many others. In the era of large scale data, it is very much worth to invest effort in adopting suitable regularization techniques when developing an analysis pipeline in order to obtain robust, reliable and interpretable results.

9 Disclosure

No competing financial interests exist.

Data type	Citation	Method	Regularization type
Gene expression (Microarrays)	Guyon et al. (2002)	Support Vector Machines	Recursive Feature Elimination
	Bøvelstad et al. (2007)	Ridge regression	Tikhonov
	Kursa (2014)	Random Forest	Tree regularization
	Deng and Runger (2013)	Random Forest	Gini index regularization
	Chen et al. (2016)	Depp learning	Dropout
	Mascelli et al. (2013)	Regularized Least Squares	Elastic-net
	Ma and Huang (2007)	Regularized Least Squares	Lasso
	De Mol et al. (2009b)	Regularized Least Squares	Elastic-net
	Hughey and Butte (2015)	Regularized Least Squares	Elastic net
	Krämer et al. (2009)	Network inference	Lasso
Gene expression (RNA-Seq)	citeyu2013shrinkage	Negative binomial distribution	Tikhonov
	Leung et al. (2014)	Deep learning	Lasso, Dropout
	Tang et al. (2017)	Cox model	Lasso
	Cheng et al. (2017)	Network inference	Group Lasso
	Yang et al. (2012a)	Network inference	Lasso
Gene expression, CNV	Žitnik and Zupan (2015)	Network inference	Network integration
ncRNA- mRNA	Soulé et al. (2020)	Random Forest	Ensemble
SNPs	Yuan et al. (2016)	Deep learning	Tikhonov (weigh decay)
	Kratsch and McHardy (2014)	Regularized Least Squares	Tikhonov
	Silver et al. (2013)	Regularized Least Squares	Group Lasso
	He et al. (2016)	Gradient boosting	Boosting and Lasso
	Alexandrov et al. (2013)	Dictionary learning	Lasso
	Piaggio et al. (2019)	Dictionary learning	Lasso
SNPs, Copy number, methylation	Aben et al. (2016)	Regularized Least Squares	Elastic net
Methylation	Johann et al. (2019)	Random Forest	Bagging
Methylation, gene expression	Csala et al. (2017)	Regularized Least Squares	Elastic net
DNA sequence	Liu et al. (2014b)	Gradient boosting	Bagging
DNA sequence	Libbrecht et al. (2015)	Network inference	Graph-based regularization
proteomic	Chen et al. (2015)	Deep learning	Tikhonov (weigh decay)
Microbioma	Ramanan et al. (2016)	Network inference	Lasso
Protein, tissue and function information	Zitnik and Leskovec (2017)	Multi-layer network inference	Tikhonov
CNV	Nowak et al. (2011); Masecchia et al. (2013)	Dictionary learning	Fused Lasso

Table 2: Applications related to the analysis of omic-data of various nature. For each type of datum we provided the specific type of analyzed data, the citation, the machine learning method and the type of regularization. Note that Recursive Feature Elimination was never explicit mentioned but it is part of the sparsity inducing regularization techniques, details can be found in Guyon and Elisseeff (2003).

Data category	**Data type**	**Citation**	**Method**	**Regularization type**
Texts	Clinical records	Garg et al. (2016)	AdaBoost	Bootstrap
Structured text	Insurance claims	Fiorini et al. (2019)	Deep learning	Early stopping and dropout
Clinical	Patient Centered Outcomes	Fiorini et al. (2017)	RLS	Nuclear Norm, Elastic Net
Images	Brain electron microscopy	Fakhry et al. (2016)	Deep learning	Tikhonov (weigh decay)
	MRI	Schlemper et al. (2017)	Deep learning	Data augmentation
	MRI	Li et al. (2018)	Deep learning	Transfer learning
	MRI	Tong et al. (2018)	Deep learning	Graph spectral regularization
	fMRI	Jenatton et al. (2012)	Generalized linear model	Hierarchical group Lasso
	MRI	Xin et al. (2014)	RLS	Generalized fused Lasso
	fMRI	Monti et al. (2014)	Network inference	Joint Lasso
	Retinal images	Javidi et al. (2017)	Dictionary learning	Lasso
	sMRI	Li et al. (2017)	Dictionary Learning	Lasso
	Retinal images	Zhou et al. (2017)	Dictionary learning	Group Lasso
	MR	Ravishankar and Bresler (2010)	Dictionary learning	ℓ_0 penalty

Table 3: Applications related to the analysis of biomedical images and textual/clinical data. For each type of datum we provided the specific type of analysed data, the citation, the machine learning method and the type of regularization.

References

Aben, N., Vis, D. J., Michaut, M., and Wessels, L. F. (2016). Tandem: a two-stage approach to maximize interpretability of drug response models based on multiple molecular data types. *Bioinformatics*, 32(17):i413–i420.

Alexandrov, L. B., Nik-Zainal, S., Wedge, D. C., Campbell, P. J., and Stratton, M. R. (2013). Deciphering signatures of mutational processes operative in human cancer. *Cell reports*, 3(1):246–259.

Ambroise, C. and McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the national academy of sciences*, 99(10):6562–6566.

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Molecular systems biology*, 12(7):878.

Azencott, C.-A. (2016). Network-guided biomarker discovery. In *Machine Learning for Health Informatics*, pages 319–336. Springer.

Belilovsky, E., Varoquaux, G., and Blaschko, M. B. (2016). Testing for differences in gaussian graphical models: applications to brain connectivity. In *Advances in Neural Information Processing Systems*, pages 595–603.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Bøvelstad, H., Nygård, S., Størvold, H., Aldrin, M., Borgan, Ø., Frigessi, A., and Lingjærde, O. (2007). Predicting survival from microarray data—a comparative study. *Bioinformatics*, 23(16):2080–2087.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Buehlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, pages 559–583.

Chandrasekaran, V., Parrilo, P. A., and Willsky, A. S. (2010). Latent variable graphical model selection via convex optimization. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1610–1613. IEEE.

Chen, L., Cai, C., Chen, V., and Lu, X. (2015). Trans-species learning of cellular signaling systems with bimodal deep belief networks. *Bioinformatics*, 31 18:3008–15.

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, 32 12:1832–9.

Cheng, L., Shan, L., and Kim, I. (2017). Multilevel gaussian graphical model for multilevel networks. *Journal of Statistical Planning and Inference*, 190:1–14.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., Ferrero, E., Agapow, P.-M., Zietz, M., Hoffman, M. M., et al. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387.

Climente-González, H., Azencott, C.-A., Kaski, S., and Yamada, M. (2019). Block hsic lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics*, 35(14):i427–i435.

Csala, A., Voorbraak, F. P., Zwinderman, A. H., and Hof, M. H. (2017). Sparse redundancy analysis of high-dimensional genetic and genomic data. *Bioinformatics*, 33(20):3228–3234.

Danaher, P., Wang, P., and Witten, D. M. (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397.

de Hoon, M., Imoto, S., and Miyano, S. (2002). Inferring gene regulatory networks from time-ordered gene expression data using differential equations. In *International Conference on Discovery Science*, pages 267–274. Springer.

De Mol, C., De Vito, E., and Rosasco, L. (2009a). Elastic-net regularization in learning theory. *Journal of Complexity*, 25(2):201–230.

De Mol, C., Mosci, S., Traskine, M., and Verri, A. (2009b). A regularized method for selecting nested groups of relevant genes from microarray data. *Journal of Computational Biology*, 16(5):677–690.

Deng, H. and Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition*, 46(12):3483–3489.

Evgeniou, T., Pontil, M., and Poggio, T. (2000). Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50.

Fakhry, A., Peng, H., and Ji, S. (2016). Deep models for brain em image segmentation: novel insights and improved performance. *Bioinformatics*, 32 15:2352–8.

Fiorini, S., Hajati, F., Barla, A., and Girosi, F. (2019). Predicting diabetes second-line therapy initiation in the australian population via time span-guided neural attention network. *PloS one*, 14(10).

Fiorini, S., Verri, A., Barla, A., Tacchino, A., and Brichetto, G. (2017). Temporal prediction of multiple sclerosis evolution from patient-centered outcomes. In *Machine Learning for Healthcare Conference*, pages 112–125.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Garg, R. P., Dong, S., Shah, S. J., and Jonnalagadda, S. R. (2016). A bootstrap machine learning approach to identify rare disease patients from electronic health records. *CoRR*, abs/1609.01586.

Giraud, C. (2014). *Introduction to high-dimensional statistics*, volume 138. CRC Press.

Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422.

Hallac, D., Park, Y., Boyd, S., and Leskovec, J. (2017). Network inference via the time-varying graphical lasso. In *Proc. of the 23rd ACM SIGKDD*, pages 205–213. ACM.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning*, volume 2. Springer.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC Press.

He, K., Li, Y., Zhu, J., Liu, H., Lee, J. E., Amos, C. I., Hyslop, T., Jin, J., Lin, H., Wei, Q., et al. (2016). Component-wise gradient boosting and false discovery control in survival analysis with high-dimensional covariates. *Bioinformatics*, 32(1):50–57.

Hernández-García, A. and König, P. (2018). Data augmentation instead of explicit regularization. *CoRR*, abs/1806.03852.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Hofner, B., Boccuto, L., and Göker, M. (2015). Controlling false discoveries in high-dimensional situations: boosting with stability selection. *BMC bioinformatics*, 16(1):144.

Huang, K. and Aviyente, S. (2007). Sparse representation for signal classification. In *Advances in neural information processing systems*, pages 609–616.

Huang, L., Liao, L., and Wu, C. H. (2016). Inference of protein-protein interaction networks from multiple heterogeneous data. *EURASIP Journal on Bioinformatics and Systems Biology*, 2016(1):1–9.

Hughey, J. J. and Butte, A. J. (2015). Robust meta-analysis of gene expression using the elastic net. *Nucleic Acids Research*, 43(12):e79.

Iba, W. and Langley, P. (1992). Induction of one-level decision trees. In *Machine Learning Proceedings 1992*, pages 233–240. Elsevier.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proc. of the 26th ICML*, ICML '09, pages 433–440, New York, NY. ACM.

Javidi, M., Pourreza, H.-R., and Harati, A. (2017). Vessel segmentation and microaneurysm detection using discriminative dictionary learning and sparse representation. *Computer methods and programs in biomedicine*, 139:93–108.

Jenatton, R., Gramfort, A., Michel, V., Obozinski, G., Eger, E., Bach, F., and Thirion, B. (2012). Multiscale mining of fmri data with hierarchical structured sparsity. *SIAM Journal on Imaging Sciences*, 5(3):835–856.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2011). Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12(Jul):2297–2334.

Johann, P. D., Jäger, N., Pfister, S. M., and Sill, M. (2019). Rf_purify: a novel tool for comprehensive analysis of tumor-purity in methylation array data based on random forest regression. *BMC bioinformatics*, 20(1):1–9.

Kanehisa, M. (2001). Prediction of higher order functional networks from genomic data. *Pharmacogenomics*, 2(4):373–385.

Krämer, N., Schäfer, J., and Boulesteix, A.-L. (2009). Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC bioinformatics*, 10(1):384.

Kratsch, C. and McHardy, A. C. (2014). Ridgerace: ridge regression for continuous ancestral character estimation on phylogenetic trees. *Bioinformatics*, 30(17):i527–i533.

Krogh, A. and Hertz, J. A. (1992). A simple weight decay can improve generalization. In *NIPS*, pages 950–957.

Kuismin, M. O. and Sillanpää, M. J. (2017). Estimation of covariance and precision matrix, network structure, and a view toward systems biology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(6):e1415.

Kulkarni, V. Y. and Sinha, P. K. (2012). Pruning of Random Forest classifiers: A survey and future directions. In *2012 International Conference on Data Science Engineering (ICDSE)*, pages 64–68.

Kursa, M. B. (2014). Robustness of Random Forest-based gene selection methods. *BMC Bioinformatics*, 15:8.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562.

Leung, M. K., Xiong, H. Y., Lee, L. J., and Frey, B. J. (2014). Deep learning of the tissue-regulated splicing code. *Bioinformatics*, 30(12):i121–i129.

Li, C. and Li, H. (2010). Variable selection and regression analysis for graph-structured covariates with an application to genomics. *The annals of applied statistics*, 4(3):1498–1516.

Li, H., Parikh, N. A., and He, L. (2018). A novel transfer learning approach to enhance deep neural network classification of brain functional connectomes. *Frontiers in neuroscience*, 12:491.

Li, Q., Wu, X., Xu, L., Chen, K., Yao, L., and Li, R. (2017). Multi-modal discriminative dictionary learning for alzheimer's disease and mild cognitive impairment. *Computer methods and programs in biomedicine*, 150:1–8.

Libbrecht, M. W., Ay, F., Hoffman, M. M., Gilbert, D. M., Bilmes, J. A., and Noble, W. S. (2015). Joint annotation of chromatin state and chromatin conformation reveals relationships among domain types and identifies domains of cell-type-specific expression. *Genome research*, 25(4):544–557.

Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L., et al. (2012). High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326.

Liu, H., Roeder, K., and Wasserman, L. (2010). Stability approach to regularization selection (stars) for high dimensional graphical models. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1432–1440. Curran Associates, Inc.

Liu, S., Dissanayake, S., Patel, S., Dang, X., Mlsna, T., Chen, Y., and Wilkins, D. (2014a). Learning accurate and interpretable models based on regularized random forests regression. *BMC Systems Biology*, 8(3):S5.

Liu, Y., Li, B., Tan, R., Zhu, X., and Wang, Y. (2014b). A gradient boosting approach for filtering de novo mutations in parent-offspring trios. *Bioinformatics*, page btu141.

Lozano, A. C., Abe, N., Liu, Y., and Rosset, S. (2009). Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics*, 25(12):i110–i118.

Lundervold, A. S. and Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127.

Ma, S. and Huang, J. (2007). Additive risk survival model with microarray data. *BMC bioinformatics*, 8(1):192.

Ma, S. and Huang, J. (2008). Penalized feature selection and classification in bioinformatics. *Briefings in bioinformatics*, 9(5):392–403.

Margolin, A. A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., and Califano, A. (2006). Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. In *BMC bioinformatics*, volume 7, page S7. BioMed Central.

Mascelli, S., Barla, A., Raso, A., Mosci, S., Nozza, P., Biassoni, R., Morana, G., Huber, M., Mircean, C., Fasulo, D., et al. (2013). Molecular fingerprinting reflects different histotypes and brain region in low grade gliomas. *BMC cancer*, 13(1):387.

Masecchia, S., Barla, A., Salzo, S., and Verri, A. (2013). Dictionary learning improves subtyping of breast cancer acgh data. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 604–607. IEEE.

Mayr, A., Binder, H., Gefeller, O., Schmid, M., et al. (2014). The evolution of boosting algorithms. *Methods of Information in Medicine*, 53(6):419–427.

McNeish, D. M. and Stapleton, L. M. (2016). The effect of small sample size on two-level model estimates: A review and illustration. *Educational Psychology Review*, 28(2):295–314.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Menéndez, P., Kourmpetis, Y. A., ter Braak, C. J., and van Eeuwijk, F. A. (2010). Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the dream4 challenge. *PloS one*, 5(12):e14147.

Micchelli, C. A., Morales, J. M., and Pontil, M. (2013). Regularizers for structured sparsity. *Advances in Computational Mathematics*, 38(3):455–489.

Min, S., Lee, B., and Yoon, S. (2016). Deep learning in bioinformatics. *arXiv preprint arXiv:1603.06430*.

Molinaro, A. M., Simon, R., and Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307.

Monti, R. P., Hellyer, P., Sharp, D., Leech, R., Anagnostopoulos, C., and Montana, G. (2014). Estimating time-varying brain connectivity networks from functional mri time series. *NeuroImage*, 103:427–443.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nielsen, T. D. and Jensen, F. V. (2009). *Bayesian networks and decision graphs*. Springer Science & Business Media.

Nowak, G., Hastie, T., Pollack, J. R., and Tibshirani, R. (2011). A fused lasso latent feature model for analyzing multi-sample acgh data. *Biostatistics*, 12(4):776–791.

Okser, S., Pahikkala, T., Airola, A., Salakoski, T., Ripatti, S., and Aittokallio, T. (2014). Regularized machine learning in the genetic prediction of complex traits. *PLoS Genet*, 10(11):e1004754.

Olshausen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325.

Piaggio, F., Tozzo, V., Bernardi, C., Croce, M., Puzone, R., Viaggi, S., Patrone, S., Barla, A., Coviello, D., J Jager, M., et al. (2019). Secondary somatic mutations in g-protein-related pathways and mutation signatures in uveal melanoma. *Cancers*, 11(11):1688.

Plumb, G., Al-Shedivat, M., Xing, E., and Talwalkar, A. (2019). Regularizing black-box models for improved interpretability. *arXiv preprint arXiv:1902.06787*.

Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Qi, Y. (2012). Random forest for bioinformatics. In *Ensemble machine learning*, pages 307–323. Springer.

Ramanan, D., Bowcutt, R., Lee, S. C., Tang, M. S., Kurtz, Z. D., Ding, Y., Honda, K., Gause, W. C., Blaser, M. J., Bonneau, R. A., Lim, Y. A., Loke, P., and Cadwell, K. (2016). Helminth infection promotes colonization resistance via type 2 immunity. *Science*, 352(6285):608–612.

Ravishankar, S. and Bresler, Y. (2010). Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE transactions on medical imaging*, 30(5):1028–1041.

Schlemper, J., Caballero, J., Hajnal, J. V., Price, A. N., and Rueckert, D. (2017). A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE transactions on Medical Imaging*, 37(2):491–503.

Silver, M., Chen, P., Li, R., Cheng, C.-Y., Wong, T.-Y., Tai, E.-S., Teo, Y.-Y., and Montana, G. (2013). Pathways-driven sparse regression identifies pathways and genes associated with high-density lipoprotein cholesterol in two asian cohorts. *PLoS genetics*, 9(11):e1003939.

Smola, A. J. and Kondor, R. (2003). Kernels and Regularization on Graphs. In Schölkopf, B. and Warmuth, M. K., editors, *Learning Theory and Kernel Machines*, volume 2777, pages 144–158. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sohail, A. and Arif, F. (2020). Supervised and unsupervised algorithms for bioinformatics and data science. *Progress in Biophysics and Molecular Biology*, 151:14–22.

Soulé, A., Steyaert, J.-M., and Waldispühl, J. (2020). A nested 2-level cross-validation ensemble learning pipeline suggests a negative pressure against crosstalk snorna-mrna interactions in saccharomyces cerevisiae. *Journal of Computational Biology*, 27(3):390–402.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Tang, Z., Shen, Y., Zhang, X., and Yi, N. (2017). The spike-and-slab lasso cox model for survival prediction and associated genes detection. *Bioinformatics*, page btx300.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108.

Tikhonov, A. (1963). Solution of incorrectly formulated problems and the regularization method. In *Soviet Math. Dokl.*, volume 5, pages 1035–1038.

Toga, A. W. and Dinov, I. D. (2015). Sharing big biomedical data. *Journal of big data*, 2(1):7.

Tomasi, F., Tozzo, V., Salzo, S., and Verri, A. (2018). Latent variable time-varying network inference. In *Proc. of the 24th ACM SIGKDD*, pages 2338–2346. ACM.

Tong, A., van Dijk, D., III, J. S. S., Amodio, M., Yim, K., Muhle, R., Noonan, J., Wolf, G., and Krishnaswamy, S. (2018). Interpretable neuron structuring with graph spectral regularization.

Vrieze, S. I. (2012). Model selection and psychological theory: a discussion of the differences between the akaike information criterion (aic) and the bayesian information criterion (bic). *Psychological methods*, 17(2):228.

Waldmann, P., Mészáros, G., Gredler, B., Fuerst, C., and Sölkner, J. (2013). Evaluation of the lasso and the elastic net in genome-wide association studies. *Frontiers in genetics*, 4:270.

Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Xie, Y., Liu, Y., and Valdar, W. (2016). Joint estimation of multiple dependent gaussian graphical models with applications to mouse genomics. *Biometrika*, 103(3):493–511.

Xin, B., Kawahara, Y., Wang, Y., and Gao, W. (2014). Efficient Generalized Fused Lasso and its Application to the Diagnosis of Alzheimer's Disease. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Yang, E., Allen, G., Liu, Z., and Ravikumar, P. K. (2012a). Graphical models via generalized linear models. In *NIPS*, pages 1358–1366.

Yang, S., Yuan, L., Lai, Y.-C., Shen, X., Wonka, P., and Ye, J. (2012b). Feature grouping and selection over an undirected graph. In *Proc. of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 922–930, New York, NY, USA. ACM.

Yu, D., Huber, W., and Vitek, O. (2013). Shrinkage estimation of dispersion in negative binomial models for rna-seq experiments with small sample size. *Bioinformatics*, 29(10):1275–1282.

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.

Yuan, Y., Shi, Y., Li, C., Kim, J., Cai, T. W., Han, Z., and Feng, D. D. (2016). Deepgene: an advanced cancer type classifier based on deep learning and somatic point mutations. In *BMC Bioinformatics*.

Zhou, S., Lafferty, J., and Wasserman, L. (2010). Time varying undirected graphs. *Machine Learning*, 80(2-3):295–319.

Zhou, W., Wu, C., Chen, D., Wang, Z., Yi, Y., and Du, W. (2017). Automatic microaneurysms detection based on multifeature fusion dictionary learning. *Computational and mathematical methods in medicine*, 2017.

Zitnik, M. and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, 33(14):i190–i198.

Žitnik, M. and Zupan, B. (2015). Gene network inference by fusing data from diverse distributions. *Bioinformatics*, 31(12):i230–i239.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.