JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 29, Number 3, 2022
Mary Ann Liebert, Inc.
Pp. 213–232
DOI: 10.1089/cmb.2019.0371

Research Articles

Where Do We Stand in Regularization for Life Science Studies?

VERONICA TOZZO,[1] CHLOÉ-AGATHE AZENCOTT,[2–4] SAMUELE FIORINI,[5] EMANUELE FAVA,[6] ANDREA TRUCCO,[6] and ANNALISA BARLA[1]

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 article, 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.

Keywords: life sciences, regularization, supervised learning, unsupervised learning.

[1]Department of Informatics, Bioengineering, Robotics and System Engineering—DIBRIS, University of Genoa, Genoa, Italy.
[2]Centre for Computational Biology—CBIO, MINES ParisTech, PSL Research University, Paris, France.
[3]Institut Curie, PSL Research University, Paris, France.
[4]INSERM, U900, Paris, France.
[5]Iren S.p.a, Genoa, Italy.
[6]Departiment of Electrical, Electronic, Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, Genoa, Italy.

1. MOTIVATION

In the era of personalized medicine, biospecimen collection and biological data management are 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 these 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 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 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 used, together with machine learning methods, to successfully address complex life science questions. Unlike previous review articles 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 that occur regularly and which regularization techniques are suitable to be used. 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 the relationships 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?

1.1. Outline

In the remainder of the article, 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 4–7. We conclude the article with a discussion (Section 8) on the most proper method to use depending on the type of data, providing a list of use cases as 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 and 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, for example, $\mathcal{Y} = \{a, b\}$ (with $a \neq b$) or *regression* if $\mathcal{Y} \subseteq R$.

Table 1. Definition of the Loss Function $L(f(\boldsymbol{x}), y)$ for Regression and Classification Problems

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

Note that $\mathbb{I}$ denotes the indicator function.

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 article we mainly refer to the problem of *binary* classification, but the *multiclass* 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 $L(f(\boldsymbol{x}), y)$ identify different learning machines (Bishop, 2006; Hastie et al., 2009).

2.2. Unsupervised learning

Unsupervised learning defines a subset of machine learning methods that allows 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 multiclass 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 allow to reconstruct the original data matrix.

We also discuss the problem of *network inference*, which is the problem of inferring relationships among variables through observations. Such method addresses 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 Equation (1):

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

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 Sections 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 to 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).

2.4. Regularization

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

2.4.1. 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 briefly discuss the effect of some choices, such as Tikhonov, Lasso, Group Lasso, Elastic-Net, and more in Sections 4 and 5.

The scalar λ is the *regularization parameter* that 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 to guide the inference (Murphy, 2012, chapter 7).

2.4.2. Ensemble techniques. Another way of avoiding overfitting is to combine a finite set of alternative models 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 discuss in Section 4.

2.4.3. Dropout and data augmentation. These two regularization techniques are mostly used for neural networks (NNs). 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 preprocessing technique. It is typically used when dealing with NNs 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 such as more explicit regularization techniques.

2.4.4. 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 used 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, that is: 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 such as AIC or BIC, we add to the error a penalty proportional to the value K, to obtain a 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 1a a schematic explanation of how to reach a particular question starting from the data and the problem at hand.

Q1: How to find relationships 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 (Okser et al., 2014; Angermueller et al., 2016) 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

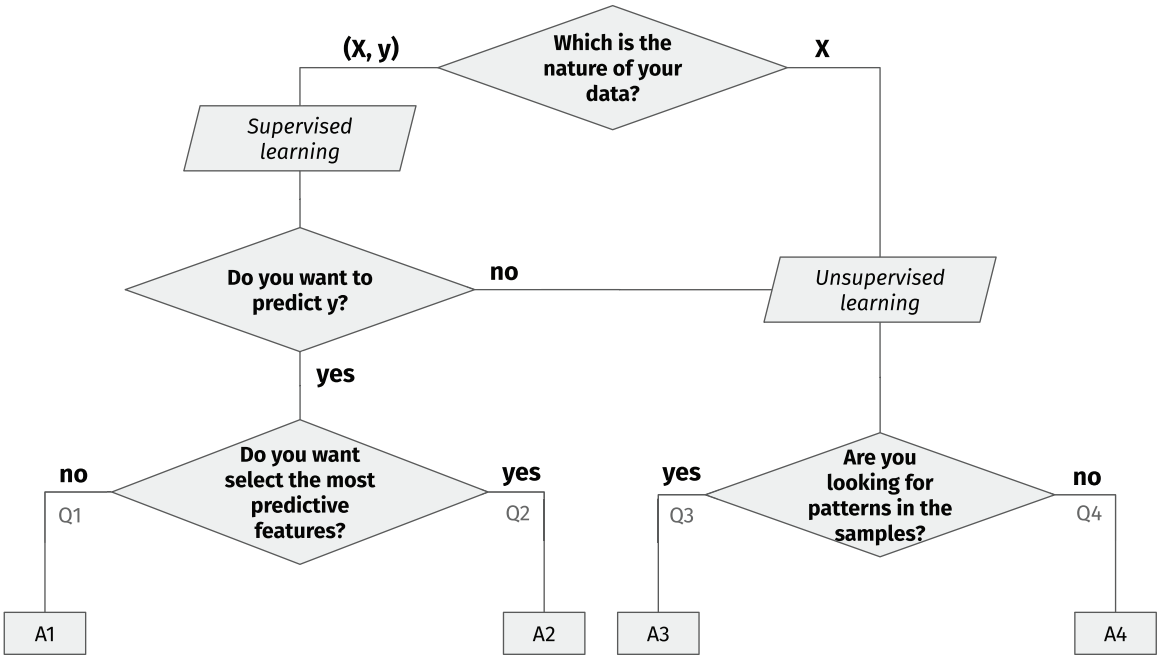

FIG. 1. Flux diagram explaining how to reach a specific question. In practice we first need to distinguish if we have labeled data or not, in the first case, we are in a supervised learning setting, while in the second we are in an unsupervised setting. In the supervised setting, we want to predict the 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.

(Tang et al., 2017; Climente-González et al., 2019). This may improve prediction power as well as promote model interpretability, that is, the ability of understanding and interpreting the parameters of the inferred model 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 (Mascelli et al., 2013; Silver et al., 2013; Giraud, 2014). From a methodological standpoint, this topic is 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 feature 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 is introduced in Section 6.

Q4: Are there relevant relationships between variables? Another common problem that 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), that is, a set of variables (nodes) connected with each other based on a particular type of relationship (links). The graphical modeling of the variables offers a compact and efficient representation that helps to identify the variability patterns in the data (Monti et al., 2014). An overview of this class of methods is provided in Section 7.

In all these questions, regularization plays a key role 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 RELATIONSHIPS BETWEEN INPUT AND OUTPUT FROM NOISY DATA? (A1)

This problem lies in the macrocategory 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 and 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 used 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). It 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).

4.1.1. 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 the 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 sclerosis.

4.2. Random forests

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 preselected 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 RFs 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 RFs 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 to favor large partitions (Liu et al., 2014a).

4.2.1. Applications. RFs can handle both numerical and categorical variables, multiple scales, and nonlinearities. 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 RF is provided in Qi (2012). Deng and Runger (2013) and Kursa (2014) use regularized and robust RF for the selection of genes in classification tasks. RFs can be used also for regression, as in Johann et al. (2019), where the authors aim at quantifying tumor purity or, for learning interactions between noncoding 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, that is, 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.

4.3.1. 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 (Section 7) and a prediction function, at the same time (LeCun et al., 2015). DL methods can be seen as an extension of classical NN, where the final prediction is achieved by composing several layers of nonlinear transformations. DL architectures can be devised to tackle binary/multicategory classification (Leung et al., 2014; Angermueller et al., 2016) 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 health care 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).

4.4.1. Applications

DL 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. Moreover, weight decay is also adopted in Yuan et al. (2016) to train *DeepGene*, that is, a simple fully connected network known as multilayer 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 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, likely to 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 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 data set may be hard. To cope with this issue, we can use 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 feature. So, in general, it is possible to reuse (or fine-tune) the weights learned by a network from some data set, 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 interpretability 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 neuron activations of an arbitrary NN, can be used to enforce a meaningful graph structure. This method is successfully applied to learn gene marker correlations in a single-cell RNA-sequencing data set. 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 are 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 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 on 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 with the outcome (e.g., 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 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 the presence of groups of correlated variables, this method tends to select only one variable per group. Second, the method cannot select more variables than the sample size (De Mol et al., 2009b; Waldmann et al., 2013).

The Elastic-Net (Zou and Hastie, 2005; De Mol et al., 2009a) 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 [Eq. (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.

5.1.1. 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 (Ma and Huang, 2007; Tang et al., 2017). 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 variable weights 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 nonzero weights to follow a given underlying structure (Micchelli et al., 2013). This structure can be defined by arranging features in *groups* (typically for bioinformatic 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:

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

where the same weight w_l is associated with 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)

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

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 analogue 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 $f : \mathbb{R}^d \mapsto \mathbb{R}$, $f^{\mathrm{T}}Lf$ 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 β to be smooth on the graph structure. This regularizer is given by Equation (9). The aGrace variant (Li and Li, 2010) allows connected features to have effects of opposite directions.

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

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 Equation (8) with a nonconvex penalty: $\max\left(|w_p|, |w_q|\right) = \frac{1}{2}\left(|w_p + w_q| + |w_p - w_q|\right)$.

5.2.1. Applications. Hierarchical Group Lasso was used in a classification setting to localize the brain regions involved in the processing of visual stimuli from functional magnetic resonance imaging (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 bioinformatic 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 techniques, it is necessary to evaluate if the model recovers the 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 a preprocessing step. In fact, using the same data set to select the most important features and to evaluate the model performance leads to an overoptimistic 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 clustering 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 contexts, this assumption can prevent the analysis to detect factors that do not satisfy the requirements imposed.

6.1. Dictionary learning

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

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

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 needs to be found through cross-validation techniques.

We can assume that the dictionary is known a priori, mimicking signal decomposition techniques such 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 such as clustering, dictionary learning, sparse coding, data integration, matrix completion, and others. These methods can be regularized through the addition of a penalty both on the patterns and on 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 with bagging techniques to prevent overfitting on the data and the initialization, as in the case of learning both the dictionary and the coefficients, is a nonconvex problem.

6.1.1. 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 undersampled 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 types of data, as in Nowak et al. (2011) and Masecchia et al. (2013), where they use a fused Lasso dictionary learning approach to perform subtyping of cancer patients analyzing 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 specializations 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).

6.2.1. Applications. In Piaggio et al. (2019), they exploit penalized non-negative matrix factorization to find patterns of somatic mutations specific of uveal melanoma from SNP data. In Javidi et al. (2017), they exploit discriminative dictionary learning and sparse representation based on Lasso penalty to perform vessel segmentation on retinal images. In Li et al. (2017), they use multimodal 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 relationships 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), coexpression edges, metabolic

influence (Kanehisa, 2001), as well as protein/protein interaction networks (Huang et al., 2016). Learning the network structure 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 network (BN)-based (Nielsen and Jensen, 2009), Gaussian graphical model (GGM)-based, differential equation (DE)-based (de Hoon et al., 2002), and mutual information (MI)-based (Margolin et al., 2006) methods. In this section, we focus on GGMs as a specific example of a wider set 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 the high dimensionality of the data and identifiability and interpretability of the resulting network. Moreover, GGMs are suited to both the inference of coexpression (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 $\mathcal{N}(\mathbf{0}, \sum)$. 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*, that is, the inverse of the covariance matrix $\Theta = \sum^{-1}$. 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, that is, only few edges are necessary to fully describe the system. Graphical Lasso (Friedman et al., 2008) can be formalized as follows:

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

where $\|\cdot\|_{od,1}$ 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 (Section 5) with a simple, efficient, and fast procedure (Friedman et al., 2008). This is, for instance, in 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 (http://dreamchallenges.org). 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 subsamplings 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 subsamplings of the data that were proven effective in many contexts.

7.1.1. 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 that 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 multitask/multiclass 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

successfully 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 to consider hidden and unmeasurable variables that influence the system through the nuclear 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 multilayer case, which integrates components of the cellular system that can act at different scales or time to obtain a more precise overview.

Table 2. Applications Related to the Analysis of Omic-Data of Various Nature

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)	RF	Tree regularization
	Deng and Runger (2013)	RF	Gini index regularization
	Chen et al. (2016)	DL	Dropout
	Mascelli et al. (2013)	RLS	Elastic-Net
	Ma and Huang (2007)	RLS	Lasso
	De Mol et al. (2009b)	RLS	Elastic-Net
	Hughey and Butte (2015)	RLS	Elastic Net
	Krämer et al. (2009)	Network inference	Lasso
Gene expression (RNA-Seq)	citeyu2013shrinkage	Negative binomial distribution	Tikhonov
	Leung et al. (2014)	DL	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)	RF	Ensemble
SNPs	Yuan et al. (2016)	DL	Tikhonov (weight decay)
	Kratsch and McHardy (2014)	RLS	Tikhonov
	Silver et al. (2013)	RLS	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)	RLS	Elastic-Net
Methylation	Johann et al. (2019)	RF	Bagging
Methylation, gene expression	Csala et al. (2017)	RLS	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)	DL	Tikhonov (weight decay)
Microbioma	Ramanan et al. (2016)	Network inference	Lasso
Protein, tissue, and function information	Zitnik and Leskovec (2017)	Multilayer network inference	Tikhonov
CNV	Nowak et al. (2011); Masecchia et al. (2013)	Dictionary learning	Fused Lasso

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 explicitly mentioned, but it is part of the sparsity inducing regularization techniques, details can be found in Guyon and Elisseeff (2003).

CNV, copy number variation; DL, deep learning; RF, Random forest; RLS, Regularized Least Squares.

7.2.1. 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) used a dynamical graphical Lasso to detect brain functional connections from fMRI images. Libbrecht et al. (2015) performed semiautomated 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).

7.3.1. Applications. Research has also moved toward the use of other distributions and models, for example, 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 such as CNV, 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 multicellular function by inferring multilayer tissue networks regularized through ℓ_2-norm.

8. CONCLUSION

This article 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, and 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 number of variables outnumbers the available samples ($n \ll p$).

Table 3. Applications Related to the Analysis of Biomedical Images and Textual/Clinical Data

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)	DL	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)	DL	Tikhonov (weight decay)
	MRI	Schlemper et al. (2017)	DL	Data augmentation
	MRI	Li et al. (2018)	DL	Transfer learning
	MRI	Tong et al. (2018)	DL	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

For each type of datum, we provided the specific type of analyzed data, the citation, the machine learning method, and the type of regularization.

MR, magnetic resonance; MRI, magnetic resonance imaging; fMRI, functional MRI; sMRI, structural MRI.

We summarized the applications cited in the articles in Tables 2 and 3. We highlighted that regularization is heavily used 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 DL and dictionary learning emerges when it comes to the analysis of biomedical images. Such behavior is expected, indeed both DL 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, for example, 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 to obtain robust, reliable, and interpretable results.

AUTHOR DISCLOSURE STATEMENT

No competing financial interests exist.

FUNDING INFORMATION

No funding was received for this article.

REFERENCES

Aben, N., Vis, D. J., Michaut, M., et al. 2016. Tandem: A two-stage approach to maximize interpretability of drug response models based on multiple molecular data types. *Bioinformatics* 32, i413–i420.

Alexandrov, L.B., Nik-Zainal, S., Wedge, D.C., et al. 2013. Deciphering signatures of mutational processes operative in human cancer. *Cell Rep.* 3, 246–259.

Ambroise, C., and McLachlan, G.J. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci.* 99, 6562–6566.

Angermueller, C., Pärnamaa, T., Parts, L., et al. 2016. Deep learning for computational biology. *Mol. Syst. Biol.* 12, 878.

Azencott, C.-A. 2016. Network-guided biomarker discovery, 319–336. In *Machine Learning for Health Informatics*. Springer, New York, NY.

Belilovsky, E., Varoquaux, G., and Blaschko, M.B. 2016. Testing for differences in gaussian graphical models: Applications to brain connectivity, 595–603. *In Advances in Neural Information Processing Systems.* Curran Associates, Inc., Red Hook, NY.

Bishop, C.M. 2006. *Pattern Recognition and Machine Learning*. Springer, New York, NY.

Bøvelstad, H., Nygård, S., Størvold, H., et al. 2007. Predicting survival from microarray dataa comparative study. *Bioinformatics* 23, 2080–2087.

Breiman, L. 2001. Random forests. *Mach. Learn.* 45, 5–32.

Buehlmann, P. 2006. Boosting for high-dimensional linear models. *Ann. Stat.* 34, 559–583.

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

Chen, L., Cai, C., Chen, V., et al. 2015. Trans-species learning of cellular signaling systems with bimodal deep belief networks. *Bioinformatics* 31, 3008–3015.

Chen, Y., Li, Y., Narayan, R., et al. 2016. Gene expression inference with deep learning. *Bioinformatics* 32, 1832–1839.

Cheng, L., Shan, L., and Kim, I. 2017. Multilevel gaussian graphical model for multilevel networks. *J. Stat. Plan. Inference* 190, 1–14.

Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., et al. 2018. Opportunities and obstacles for deep learning in biology and medicine. *J. Royal Soc. Interf.* 15, 20170387.

Climente-González, H., Azencott, C.-A., Kaski, S., et al. 2019. Block hsic lasso: Model-free biomarker detection for ultra-high dimensional data. *Bioinformatics* 35, i427–i435.

Csala, A., Voorbraak, F.P., Zwinderman, A.H., et al. 2017. Sparse redundancy analysis of high-dimensional genetic and genomic data. *Bioinformatics* 33, 3228–3234.

Danaher, P., Wang, P., and Witten, D.M. 2014. The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Stat Soc. Series B Stat. Methodol.* 76, 373–397.

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

De Mol, C., De Vito, E., and Rosasco, L. 2009a. Elastic-net regularization in learning theory. *J. Complex.* 25, 201–230.

De Mol, C., Mosci, S., Traskine, M., et al. 2009b. A regularized method for selecting nested groups of relevant genes from microarray data. *J. Comput. Biol.* 16, 677–690.

Deng, H., and Runger, G. 2013. Gene selection with guided regularized random forest. *Pattern Recogn.* 46, 3483–3489.

Evgeniou, T., Pontil, M., and Poggio, T. 2000. Regularization networks and support vector machines. *Adv. Comput. Math.* 13, 1–50.

Fakhry, A., Peng, H., and Ji, S. 2016. Deep models for brain em image segmentation: Novel insights and improved performance. *Bioinformatics* 32, 2352–2358.

Fiorini, S., Hajati, F., Barla, A., et al. 2019. Predicting diabetes second-line therapy initiation in the australian population via time span-guided neural attention network. *PLoS One* 14, e0211844.

Fiorini, S., Verri, A., Barla, A., et al. 2017. Temporal prediction of multiple sclerosis evolution from patient-centered outcomes, 112–125. *In* Doshi-Velez, F., Fackler, J., Kale, D., Ranganath, R., Wallace, B., and Wiens, J., eds. *Machine Learning for Healthcare Conference* PMLR, Boston, MA.

Freund, Y. 1995. Boosting a weak learning algorithm by majority. *Inf. Comput.* 121, 256–285.

Friedman, J., Ha stie, T., and Tibshirani, R. 2008. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9, 432–441.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* 29, 1189–1232.

Garg, R. P., Dong, S., Shah, S. J., et al. 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, Boca Raton, FL.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. Mac. Learn. Res.* 3, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., et al. 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46, 389–422.

Hallac, D., Park, Y., Boyd, S., et al. 2017. Network inference via the time-varying graphical lasso, 205–213. In *Proceedings of the 23rd ACM SIGKDD*. ACM, Nova Scotia, Canada.

Hastie, T., Tibshirani, R., and Friedman, J. 2009. *The Elements of Statistical Learning*, Volume 2. Springer, New York, NY.

Hastie, T., Tibshirani, R., and Wainwright, M. 2015. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, Boca Raton, FL.

He, K., Li, Y., Zhu, J., et al. 2016. Component-wise gradient boosting and false discovery control in survival analysis with high-dimensional covariates. *Bioinformatics* 32, 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, 55–67.

Hofner, B., Boccuto, L., and Göker, M. 2015. Controlling false discoveries in high-dimensional situations: Boosting with stability selection. *BMC Bioinformatics* 16, 144.

Huang, K., and Aviyente, S. 2007. Sparse representation for signal classification, 609–616. *In* Hoffman, T., Platt, J., and Schölkopf, B., eds. *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA.

Huang, L., Liao, L., and Wu, C.H. 2016. Inference of protein-protein interaction networks from multiple heterogeneous data. *EURASIP J. Bioinf. Syst. Biol.* 2016, 1–9.

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

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

Jacob, L., Obozinski, G., and Vert, J.-P. 2009. Group lasso with overlap and graph lasso, 433–440. In *Proceedings of the 26th ICML, ICML'09*. 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. *Comput. Methods Programs Biomed.* 139, 93–108.

Jenatton, R., Gramfort, A., Michel, V., et al. 2012. Multiscale mining of fMRI data with hierarchical structured sparsity. *SIAM J. Imaging Sci.* 5, 835–856.

Jenatton, R., Mairal, J., Obozinski, G., et al. 2011. Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res.* 12, 2297–2334.

Johann, P.D., Jäger, N., Pfister, S.M., et al. 2019. Rf_purify: A novel tool for comprehensive analysis of tumor-purity in methylation array data based on random forest regression. *BMC Bioinformatics* 20, 1–9.

Kanehisa, M. 2001. Prediction of higher order functional networks from genomic data. *Pharmacogenomics* 2, 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, 384.

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

Krogh, A., and Hertz, J.A. 1992. A simple weight decay can improve generalization, 950–957. *In* Hanson, S., Lippmann, R.P., and Moody, J., eds. *NIPS*. Morgan-Kaufmann, San Francisco, CA.

Kuismin, M.O., and Sillanpää, M.J. 2017. Estimation of covariance and precision matrix, network structure, and a view toward systems biology. *Wiley Interdiscip. Rev. Comput. Stat.* 9, e1415.

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

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, 436–444.

Lee, D.D., and Seung, H.S. 2001. Algorithms for non-negative matrix factorization, 556–562. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA.

Leung, M.K., Xiong, H.Y., Lee, L.J., et al. 2014. Deep learning of the tissue-regulated splicing code. *Bioinformatics* 30, i121–i129.

Li, C., and Li, H. 2010. Variable selection and regression analysis for graph-structured covariates with an application to genomics. *Ann. Appl. Stat.* 4, 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. *Front. Neurosci.* 12, 491.

Li, Q., Wu, X., Xu, L., et al. 2017. Multi-modal discriminative dictionary learning for Alzheimer's disease and mild cognitive impairment. *Comput. Methods Programs Biomed.* 150, 1–8.

Libbrecht, M.W., Ay, F., Hoffman, M.M., et al. 2015. Joint annotation of chromatin state and chromatin conformation reveals relationships among domain types and identifies domains of cell-type-specific expression. *Genome Res.* 25, 544–557.

Liu, H., Han, F., Yuan, M., et al. 2012. High-dimensional semiparametric gaussian copula graphical models. *Ann. Stat.* 40, 2293–2326.

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

Liu, S., Dissanayake, S., Patel, S., et al. 2014a. Learning accurate and interpretable models based on regularized random forests regression. *BMC Syst. Biol.* 8, S5.

Liu, Y., Li, B., Tan, R., et al. 2014b. A gradient boosting approach for filtering de novo mutations in parent-offspring trios. *Bioinformatics* 30, 1830–1836.

Lozano, A.C., Abe, N., Liu, Y., et al. 2009. Grouped graphical granger modeling for gene expression regulatory networks discovery. *Bioinformatics* 25, 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, 102–127.

Ma, S., and Huang, J. 2007. Additive risk survival model with microarray data. *BMC Bioinformatics* 8, 192.

Ma, S., and Huang, J. 2008. Penalized feature selection and classification in bioinformatics. *Brief. Bioinform.* 9, 392–403.

Margolin, A.A., Nemenman, I., Basso, K., et al. 2006. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* 7, S7.

Mascelli, S., Barla, A., Raso, A., et al. 2013. Molecular fingerprinting reflects different histotypes and brain region in low grade gliomas. *BMC Cancer* 13, 387.

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

Mayr, A., Binder, H., Gefeller, O., et al. 2014. The evolution of boosting algorithms. *Methods Inf. Med.* 53, 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. *Educ. Psychol. Rev.* 28, 295–314.

Meinshausen, N., and Bühlmann, P. 2010. Stability selection. *J. R. Stat. Soc. Series B Stat. Methodol.* 72, 417–473.

Menéndez, P., Kourmpetis, Y.A., ter Braak, C.J., et al. 2010. Gene regulatory networks from multifactorial perturbations using graphical lasso: Application to the dream4 challenge. *PLoS One* 5, e14147.

Micchelli, C.A., Morales, J.M., and Pontil, M. 2013. Regularizers for structured sparsity. *Adv. Comput. Math.* 38, 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, 3301–3307.

Monti, R.P., Hellyer, P., Sharp, D., et al. 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, Cambridge, MA.

Nielsen, T.D., and Jensen, F.V. 2009. *Bayesian Networks and Decision Graphs*. Springer Science & Business Media, Berlin, Germany.

Nowak, G., Hastie, T., Pollack, J.R., et al. 2011. A fused lasso latent feature model for analyzing multi-sample acgh data. *Biostatistics* 12, 776–791.

Okser, S., Pahikkala, T., Airola, A., et al. 2014. Regularized machine learning in the genetic prediction of complex traits. *PLoS Genet.* 10, e1004754.

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

Piaggio, F., Tozzo, V., Bernardi, C., et al. 2019. Secondary somatic mutations in g-protein-related pathways and mutation signatures in uveal melanoma. *Cancers* 11, 1688.

Plumb, G., Al-Shedivat, M., Xing, E., et al. 2019. Regularizing black-box models for improved interpretability. *arXiv preprint arXiv:1902.06787*.

Prechelt, L. 1998. Early stopping-but when?, 55–69. In Müller, K.-R., and Orr, G.B., eds. *Neural Networks: Tricks of the trade*. Springer, Berlin, Germany.

Qi, Y. 2012. Random forest for bioinformatics, 307–323. In Zhang, C., and Ma, Y., eds. *Ensemble Machine Learning*. Springer, Boston, MA.

Ramanan, D., Bowcutt, R., Lee, S.C., et al. 2016. Helminth infection promotes colonization resistance via type 2 immunity. *Science* 352, 608–612.

Ravishankar, S., and Bresler, Y. 2010. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Trans. Med. Imaging* 30, 1028–1041.

Schlemper, J., Caballero, J., Hajnal, J.V., et al. 2017. A deep cascade of convolutional neural networks for dynamic mr image reconstruction. *IEEE Trans. Med. Imaging* 37, 491–503.

Silver, M., Chen, P., Li, R., et al. 2013. Pathways-driven sparse regression identifies pathways and genes associated with high-density lipoprotein cholesterol in two Asian cohorts. *PLoS Genet.* 9, e1003939.

Smola, A.J., and Kondor, R. 2003. Kernels and regularization on graphs, 144–158. *In* Schlkopf, B. and Warmuth, M.K., eds. *Learning Theory and Kernel Machines*, Volume 2777. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sohail, A., and Arif, F. 2020. Supervised and unsupervised algorithms for bioinformatics and data science. *Progr. Biophys. Mol. Biol.* 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. *J. Comput. Biol.* 27, 390–402.

Srivastava, N., Hinton, G., Krizhevsky, A., et al. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.

Tang, Z., Shen, Y., Zhang, X., et al. 2017. The spike-and-slab lasso cox model for survival prediction and associated genes detection. *Bioinformatics* 33, 2799–2807.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Series B Stat. Methodol.* 267–288.

Tibshirani, R., Saunders, M., Rosset, S., et al. 2005. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Series B Stat. Methodol.* 67, 91–108.

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

Toga, A.W., and Dinov, I.D. 2015. Sharing big biomedical data. *J. Big Data* 2, 7.

Tomasi, F., Tozzo, V., Salzo, S., et al. 2018. Latent variable time-varying network inference, 2338–2346. In *Proceedings of the 24th ACM SIGKDD*. ACM, London, United Kingdom.

Tong, A., van Dijk, D.,Stanley, III, J.S., et al. 2018. Interpretable neuron structuring with graph spectral regularization. In Berthold, M.R., Feelders, A., and Krempl, G., eds. *Advances in Intelligent Data Analysis XVIII*. Springer, Kostanz, Germany.

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). *Psychol. Methods* 17, 228.

Waldmann, P., Mészáros, G., Gredler, B., et al. 2013. Evaluation of the lasso and the elastic net in genome-wide association studies. *Front. Genet.* 4, 270.

Wold, S., Esbensen, K., and Geladi, P. 1987. Principal component analysis. *Chemom. Intell. Lab. Syst.* 2, 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, 493–511.

Xin, B., Kawahara, Y., Wang, Y., et al. 2014. Efficient generalized fused lasso and its application to the diagnosis of Alzheimers disease. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAI Press, Palo Alto, CA.

Yang, E., Allen, G., Liu, Z., et al. 2012a. Graphical models via generalized linear models, 1358–1366. In *NIPS*. Curran Associates, Inc., Red Hook, NY.

Yang, S., Yuan, L., Lai, Y.-C., et al. 2012b. Feature grouping and selection over an undirected graph, 922–930. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'12. 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, 1275–1282.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Series B Stat. Methodol.* 68, 49–67.

Yuan, Y., Shi, Y., Li, C., et al. 2016. Deepgene: An advanced cancer type classifier based on deep learning and somatic point mutations. *BMC Bioinformatics*.

Zhou, S., Lafferty, J., and Wasserman, L. 2010. Time varying undirected graphs. *Mach. Learn.* 80, 295–319.

Zhou, W., Wu, C., Chen, D., et al. 2017. Automatic microaneurysms detection based on multifeature fusion dictionary learning. *Comput. Math. Methods Med.* 2017, 2483137.

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

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

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Series B Stat. Methodol.* 67, 301–320.

Address correspondence to:
Dr. Veronica Tozzo
Department of Informatics, Bioengineering, Robotics and System Engineering—DIBRIS
University of Genoa
Genoa 16146
Italy

E-mail: veronica.tozzo@dibris.unige.it