An overview on penalized network regression approaches (with applications in genomics)

Claudia Angelini
Istituto per le Applicazioni del Calcolo “M. Picone”

28 February 2019

Introduction

Regression approaches

Regression is a well-known statistical frameworks that allows to explain and predict the behaviour of a dependent (response) random variable Y as a function $f()$ of p (explanatory) variables $X_1, ...X_p$, i.e.,

$$Y = f(X_1, ..., X_p, \beta_1, ..., \beta_p, \epsilon)$$

1. **Estimate** $\beta = (\beta_1, ..., \beta_p)$ with $\hat{\beta}$ given n observed samples

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & .... & x_{1,p} \\ x_{2,1} & x_{2,2} & .... & x_{2,p} \\ .... & .... & .... & .... \\ .... & .... & .... & .... \\ x_{n,1} & x_{n,2} & .... & x_{n,p} \end{bmatrix} \in \mathbf{R}^{n \times p}$$

$$\mathbf{Y} = [y_1, ..., y_n]^T \in R^{n \times 1}$$

2. **Predict**

$$\hat{y}_{new} = f(x_{new,1}, ..., x_{new,p}, \hat{\beta})$$

Note

In this context, $\mathbf{Y}$ can be either a continuous or a discrete variable.

Regression approaches

There exists several regression models

- Linear models
- Generalized linear models
- Cox-Regression
- Many others models, including non linear relationships

$\hat{\beta} \in R^p$ can be obtained by minimizing an **objective function** $S(\beta)$, i.e,

$$\hat{\beta} = argmin_{\beta \in R^p} \; S(\beta).$$

$S(\beta)$ is often a **Loss function** $L(\beta)$ that can be the negative log-likelihood or the negative partial log-likelihood.

Note

In alternative to frequentist approaches, it is also possible to consider **Bayesian** approaches, in such cases the inference is carried out on posterior probability (i.e., MAP).

Linear regression to fix the ideas

Linear regression assumes that $f()$ is linear, i.e.,

$$y_i = \beta_0 1 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i, \quad 1 = 1, ..., n.$$

where ϵ_i are *i.i.d* zero-mean random variables, representing the noise.

$\Rightarrow \hat{\beta}$ is obtained by minimizing

$$L(\beta) = \sum_{i=1}^{n} (y_i - \mathbf{X}_i^T \beta)^2,$$

i.e., if $\mathbf{X}^T\mathbf{X}$ is non singular,

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Under the **Gauss-Markov properties**, $\hat{\beta}$ is the **BLUE**.

Note

Usually, we assume that the predictors are standardized and the response is centered.

High-dimensional regression and penalization

Note

We can distinguish two main **frameworks**:
- **Classical** setting: $p < n$
- **High-dimensional** setting: $p >> n$

When $p > n$, OLSE does not perform well due to overfitting. Therefore, some form of **regularization** is required.

The general form the objective function $S(\beta)$ becomes

$$S_\lambda(\beta) = L(\beta) + P_\lambda(\beta).$$

where P_λ penalizes values of the unknown parameters β tobalance the **bias-variance trade-off** $\Rightarrow$ $P_\lambda(\beta)$ imposes **smoothness** and/or **sparsity**.

Note

λ is a vector containing one or more regularization parameters that tune the effect of the penalty $\Rightarrow$ $\hat{\beta}_\lambda$ depends on the choice of λ that need to be chosen from the data, for example using cross-validation.

$\Rightarrow$ For a review, see (Buhlmann and van de Geer, 2011).

Well-known examples of penalization approaches

- **Ridge regression** (Hoerl and Kennard, 1970) penalizes large values in the estimated coefficients, β, by using ℓ_2 regularization, i.e.,

$$P(\beta) = \lambda \sqrt{\sum_{i=1}^{p} \beta_i^2} = \lambda ||\beta||_2$$

  Therefore, it shrinks the coefficients, but it does not perform feature selection.

- **Lasso** (Tibshirani, 1996) aims to produce a sparse coefficient vector by using $\ell_1$penalty,

$$P(\beta) = \lambda \sum_{i=1}^{p} |\beta_i| = \lambda ||\beta||_1,$$

  So, it performs both variable selection and regularization.

- **Elastic Net** (Zou and Hastie,2005) linearly combines the ℓ_1 and ℓ_2 penalties, i.e.,

$$P(\beta) = \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sqrt{\sum_{i=1}^{p} \beta_i^2} = \lambda_1 ||\beta||_1 + \lambda_2 ||\beta||_2.$$

  It performs both variable selection and regularization, and has an implicit grouping effect.

- **Other penalties**: SCAD (Fan and Li, 2001); Dantzig Selector (Candes and Tao, 2007); Fused-lasso (Tibshirani et al, 2005); Grouped-lasso (Yuan and Li, 2006), etc.

Omics data

In the recent years, the advent of **high-throughput technologies** such as microarrays, next generation sequencing, mass spectrometry, etc., has allowed to measure the level of expression of genes or proteins, to monitor the level of DNA methylation, to identify other genomic structural variants such as SNPs, at a **genome-wide** scale for **different individuals** or **under different experimental/physiological conditions**.

One of the central problems in omics research is to identify genes, regulatory elements or pathways involved in disease etiology and progression by linking different omic data to various clinical outcomes.

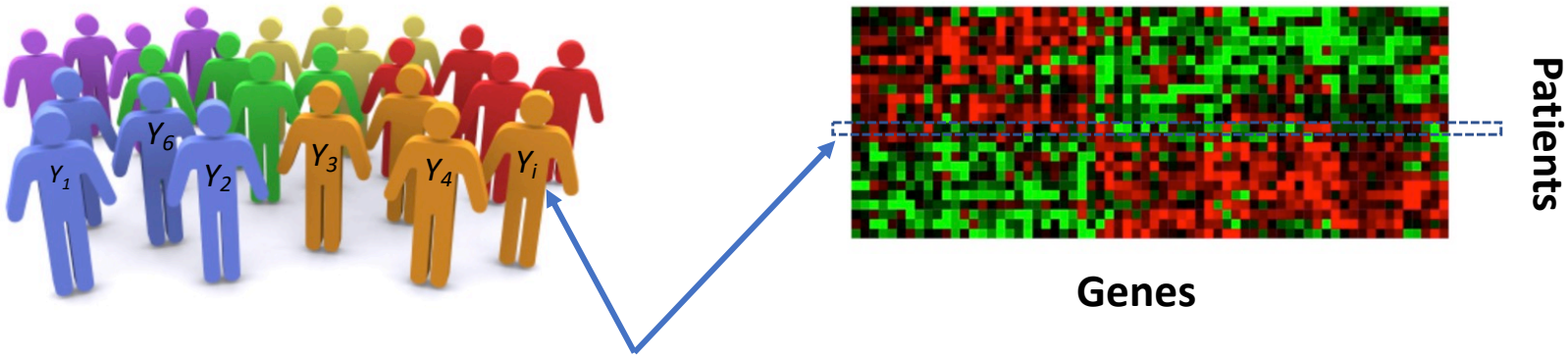
Typical examples in Omics science:

$\Rightarrow$ Y : a clinical outcome and X_j: expression level of gene j. Then, $\hat{\beta}_j \neq 0$ implies that the gene j is related to the clinical outcome; $\{\hat{\beta}_s : \quad s \in S\}$ is a potential gene-signature that could be used for predicting novel outcomes.

Note

Omics data are high-dimensional, i.e., $p \approx 10^4 - 10^5$ or even higher, althought the sample size is usually relatively small i.e., $n \approx 10 - 10^3$.

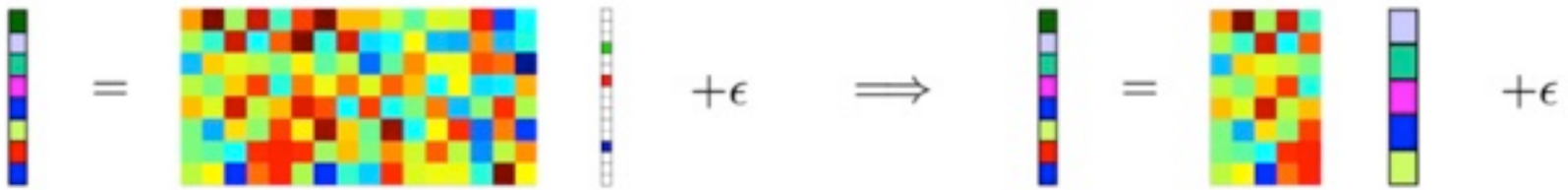
Omics data in figure



The aim is

- To identify a subset of ***biomakers*** that can be predictive of a given outcome
- To **predict novel outcomes**

$$y = X\beta + \epsilon \implies y = X_S\beta_S + \epsilon$$



Networks in Omics

Graph and networks are a common way of depicting information.

In biology, many different processes can be represented by graphs such as regulatory networks, protein-protein interaction networks, or metabolic pathways, etc. Moreover, a large body of information is available in well-known databases such as:

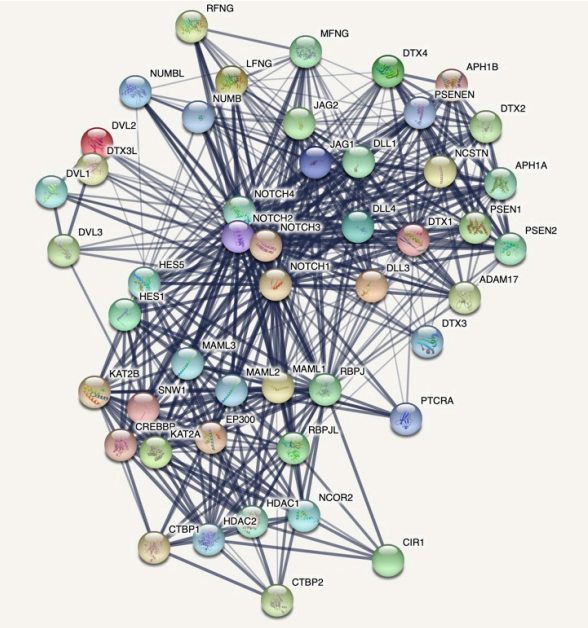
- KEGG
- Gene Ontology
- Reactome
- BioCarta
- String
- Cosmic
- and many others

Key Ideas:

It is reasonable to assume that two neighboring genes in a network are more likely to participate together in the same biological process than two genes far away in the network.

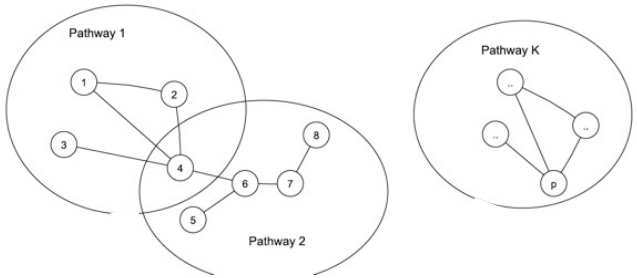
Classical penalization approaches **do not utilize prior biological knowledge** $\Rightarrow$ The explicit use of **network structured** information might lead to a better selection.

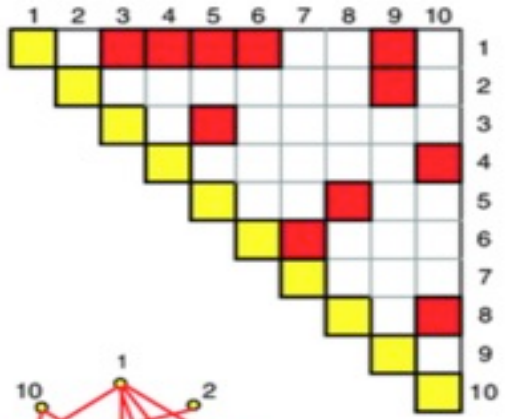
Example

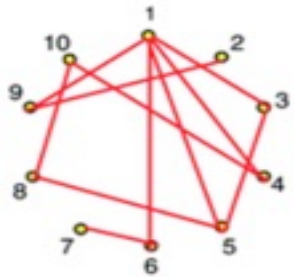


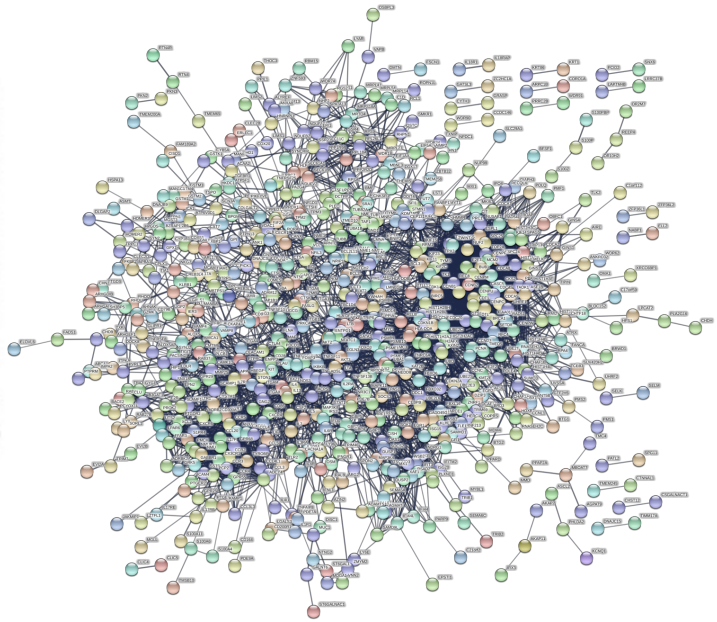
Notch signaling is an evolutionarily conserved, intercellular signaling mechanism that plays myriad roles during vascular development and physiology in vertebrates and in T cell differentiation.

It is known that the genes do not work in isolation or independently with each other; they function coordinately in **pathways** or **networks**.









Networkws & Notations

A **network** can be represented by a **weighed undirect graph** $G = (V, E, W)$, where V is the set of vertices that correspond to the p predictors, $E = \{u \sim v\}$ is the set of edges indicating that the predictors u and v are linked on the network, and W contains the weights $w(u, v)$ of the edge $e = (u \sim v)$

The **adjacency matrix A** $\in [0, 1]^{p \times p}$ is such that

$$A_{u,v} = \begin{cases} w(u,v) & \text{if} \quad (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

The (diagonal) **degree** matrix $\mathbf{D} \in R^{p \times p}$ as $Diag(D)_v = d_v = \sum_{u \sim v} w(u, v)$

The **normalized Laplacian matrix L** $= D - A$,

$$L_{u,v} = \begin{cases} 1 - w(u,v)/d_u & \text{if} \quad u = v \quad \text{and} \quad d_u \neq 0 \\ -\frac{w(u,v)}{\sqrt{d_u d_v}} & \text{if} \quad (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

Note: **L** is always non-negative definite and its spectrum reflects many properties of the graph (Chung, 1997).

Network-based penalized regression

Motivating ideas

The idea is modeling a **phenotype** through **gene expression profiles** while accounting for coordinated functioning of genes in the form of **biological pathways or networks**.

- The network is assumed to be known and to reflect the available biological information.

Network assumption/prior:

If two genes $u \sim v$ in a network, then the relationship between the corresponding regression coefficients could be incorporated in the model using the penalty

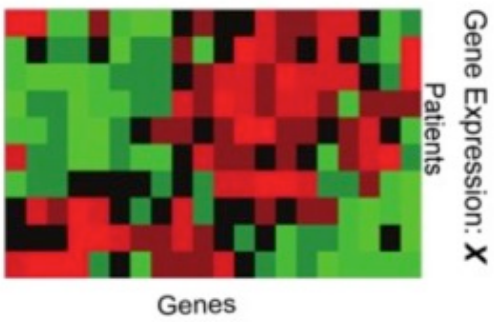
$$S_\lambda(\beta) = -l(\beta) + P_\lambda(\beta)$$

The penalty can be written as a **mixture penalty**:

$$P_\lambda(\beta) = \lambda_1 \Psi(\beta) + \lambda_2 \Phi(\beta) = \lambda\big(\alpha\Psi(\beta) + (1-\alpha))\Phi(\beta)\big)$$

where $\Psi()$ is a function that should induce **sparsity** and $\Phi()$ **smoothness** across the network.

Motivating ideas in figure



Response **Y**

1	1	1	1	0	1	1	1	0	0
3	5	10	14	20	27	30	36	38	45

δ: event indicator

t: followup time

Goodness of fit term i.e., negative log-likelihood

Variable selection term i.e., lasso-type

Network term

$$S_\lambda(\beta) = -l(\beta) + \lambda_1 \sum_{u=1}^{p} \psi(\beta_u) + \lambda_2 \sum_{u \sim v} \phi(\beta_u, \beta_v)$$

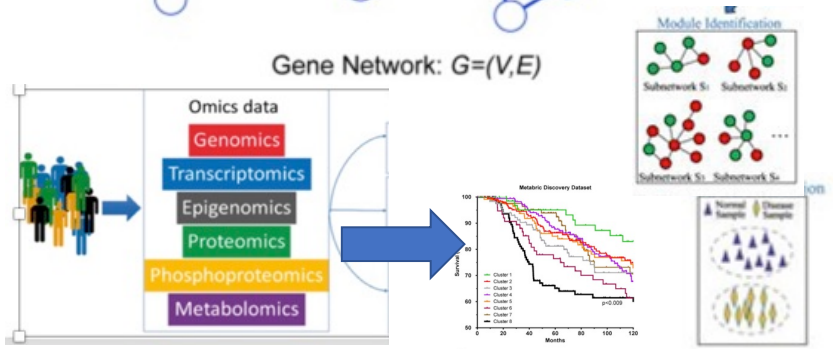
Regression analysis, **variable selection** and '***smoothness***' when the covariates are linked on a **graph**

1) Select predictors that are linked in the network (i.e., *grouping effect*) to form **sub-module** or pathways

2) Give higher probability to **hub-genes**

3) Consider either **positive and negative relationships**

4) Handle network uncertainity



Network penalized approaches & Linear Regression

GRAph Constrained Estimation: Grace (Li and Li, 2008)

Given $\lambda_1 \geq 0$ and $\lambda_1 \geq 0$ the penalty function is given by

$$P(\beta) = \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{u \sim v} \left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_v}{\sqrt{d_v}} \right)^2 w(u,v) = \lambda_1 ||\beta||_1 + \lambda_2 \beta^T \mathbf{L} \beta$$

It contains two terms: an ℓ_1 term for variable selection and a second term that performs the network penalization. It assumes $\frac{\beta_u}{d_u} \approx \frac{\beta_v}{d_v} \quad u \sim v$.

The penalty is strictly convex for $\lambda_2 > 0$ and encourages genes with a higher degree in the network (e.g., hub genes) to have larger coefficients.

Since $\mathbf{L}$ is non-negative definite, the problem can be solved using coordinate descend algorithm (Friedman et al, 2007; Wu and Lange, 2008).

Adaptive GRAph-Constrained Estimation - AGRACE (Li and Li, 2010)

One drawback of the original Grace approach is that it performs poorly when the coefficients of two linked predictors have different signs.

Let $\tilde{\beta}$ be an initial estimate obtained through OLSE if $p < n$ or Elastic Net otherwise. The penalty function is then

$$P(\beta) = \lambda_1 \sum_{i=1}^{p} |\beta_i| + \lambda_2 \sum_{u \sim v} \left(\frac{sign(\tilde{\beta}_u)\beta_u}{\sqrt{d_u}} - \frac{sign(\tilde{\beta}_v)\beta_v}{\sqrt{d_v}} \right)^2 w(u,v) = \lambda_1 ||\beta||_1 + \lambda_2 \beta^T \mathbf{L}^* \beta$$

where

$$L^*_{u,v} = \begin{cases} 1 - w(u,v)/d_u & \text{if} \quad u = v \quad \text{and} \quad d_u \neq 0 \\ -\frac{sign(\tilde{\beta}_u) sign(\tilde{\beta}_v) w(u,v)}{\sqrt{d_u d_v}} & \text{if} \quad (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

Note that $\mathbf{L}^*$ is still positive semi-defined, therefore coordinate descendent algorithm can be applied to retrieve the solution.

Generalized-Boosted Lasso: GBLASSO (Pan et al. 2010 and Luo et al. 2012)

Consider a family of penalties functions that induce (group-wise) sparsity using a weighed ℓ_γ norm, as

$$P(\beta) = \lambda 2^{1/\gamma'} \sum_{u \sim v} \left(\frac{|\beta_u|^\gamma}{w_u} + \frac{|\beta_v|^\gamma}{w_v} \right)^{1/\gamma},$$

where $\lambda > 0$; $\gamma > 1$ and γ' satisfies $\frac{1}{\gamma'} + \frac{1}{\gamma} = 1$.

w_u is a weight function attributed to each node.

It tends to select $\frac{|\hat{\beta}_u|}{w_u} = \frac{|\hat{\beta}_v|}{w_u}$ or $\hat{\beta}_u = \hat{\beta}_v = 0$ if $u \sim v$.

The **simplified GBLasso** penalty is,

$$P(\beta) = \lambda \sum_{u \sim v} \left[\left(\frac{|\beta_u|}{w_u} \right)^\gamma + \left(\frac{|\beta_v|}{w_v} \right)^\gamma \right]^{1/\gamma}.$$

For $\gamma = 2$ and $w_u = w_w = 1$ we have a **grouped-lasso penalty**. Particularly interesting is $w_i = \sqrt{d_i}$.

Pan et al. 2010, approximated the solution using the generalized Boosted-Lasso algorithm; Luo et al 2012, proposed a general convex programming algorithm (disciplinate convex programming).

Other penalties

Let $\gamma \to \infty$ and $w_i = \sqrt{d_i}$

- **Linf** L_∞ (Luo et al 2012)

$$P(\beta) = \lambda \sum_{u \sim v} \max \left(\frac{|\beta_u|}{\sqrt{d_u}}, \frac{|\beta_v|}{\sqrt{d_v}} \right).$$

The adaptive penalty version is

- **aLinf** aL_∞ (Luo et al 2012)

$$P(\beta) = \lambda \sum_{u \sim v} \left| \frac{sign(\tilde{\beta}_u)\beta_u}{\sqrt{d_u}} - \frac{sign(\tilde{\beta}_v)\beta_v}{\sqrt{d_v}} \right|,$$

GBLasso, Linf and aLinf penalties produce good variable selection properties, but biased estimates.

Other penalties

An kind of ideal penalty could be ℓ_0 for sparsest variable selection and unbiased paramarer estimation

$$P(\beta) = \lambda_1 \sum_{u=1}^{p} I\Big(|\beta_u| \neq 0\Big) + \lambda_2 \sum_{u \sim v} \left| I\left(\frac{|\beta_u|}{w_u} \neq 0\right) - I\left(\frac{|\beta_v|}{w_v} \neq 0\right) \right|$$

i.e, by imposing either $\beta_u = \beta_v = 0$ or both different from zero. Unfortunately $I()$ is not continuous.

Consider the **Truncated Lasso penalty** $J_\tau(|z|) = \min\left(\frac{|z|}{\tau}, 1\right)$; note

$J_\tau(|z|) \to I(|z| \neq 0) \quad \tau \to 0^+$, we have

- $TTLP_I$ (Kim et al 2013) Given $\lambda_1, \lambda_2, \tau$

$$P(\beta) = \lambda_1 \sum_{u=1}^{p} J_\tau\Big(|\beta_u|\Big) + \lambda_2 \sum_{u \sim v} \left| J_\tau\left(\frac{|\beta_u|}{w_u}\right) - J_\tau\left(\frac{|\beta_v|}{w_v}\right) \right|$$

- $LTLP_I$ (Kim et al 2013)

$$P(\beta) = \lambda_1 \sum_{u=1}^{p} |\beta_u| + \lambda_2 \sum_{u \sim v} \left| J_\tau\left(\frac{|\beta_u|}{w_u}\right) - J_\tau\left(\frac{|\beta_v|}{w_v}\right) \right|$$

An algorithm based on difference convex programming (DC) can be used to minimixe the objective function with both penalties. However, since the penalties are not convex, the local minimum of the DC algorithm converges depend on the starting values.

Some considerations

- Simulations and real data analysis show that network-penalized regression outperform classical penalized approaches and tend to select fewer isolated variables with respect to classical penalty terms
- Different approaches use different penalties, migth select different variables and provide different estimates.
- Each method exhibits its strengths and weaknesses when processing different datasets and it is hard to choose which one to use on a finite size dataset for which the ground truth is not known.
- Moreover, each method requires to tune several regularization parameters $\Rightarrow$ Cross-validation might be problematic and time consuming for small sample size and/or for many regularization parameters. Moreover, a bad choice of the regularization parameters might strongly affect the results.

$\Rightarrow$ **Such considerations motivated our novel orchestrated parameter tuning approach.**

Novel orchestrated parameter tuning approach

Aim: To combine an ensamble of different regression methods $\mathbf{M_1}, ..., \mathbf{M_K}$, into a single **consensus solution** based on

- Voting approach based on the output of different methods (**Composite Voting Regression**)
- Iterative, simultaneous and cooperative hyperparameter tuning approach to estimate the reguarization parameters (**Orchestrated tuning paramenters**)

$\Rightarrow$ Therefore, we first choose the parameters of the K methods to improve their similarities, then we select the variables of interest using a voting approach, finally we fit the data using OLS or elastic-net method on the best-voted variables.

For details see

S. Daskalov, A. Iuliano, K. Bliznakova, P. Liò, C. Angelini, *Orchestrated Hyperparameter Tuning of Regression Method Ensembles*. In preparation (2019).

- Morever, a Python software* that implements all the procedures is also under preparation

*Please, ask Sivo Daskalov for the preliminary version of the code.

Composite Voting Regression

Let K be different regression methods $\mathbf{M}_i, \quad i = 1, .., K$ processing a given dataset with p predictors, and producing estimates $\hat{\beta}_{M_i}$.

Let $\mathbf{B} \in R^{K \times p}$, such that $B_{i,j} = \mathbf{M}_i(\beta_j) = \hat{\beta}_{M_i,j}$

	X_1	X_2	...	X_p
$\mathbf{M}_1$	$M_1(\beta_1)$	$M_1(\beta_2)$	...	$M_1(\beta_p)$
$\mathbf{M}_2$	$M_2(\beta_1)$	$M_2(\beta_2)$	...	$M_2(\beta_p)$
...	...	...	...	...
$\mathbf{M}_K$	$M_K(\beta_1)$	$M_K(\beta_2)$	...	$M_K(\beta_p)$

We define the **fraction of votes** (FV statistic) as

$$FV_j = \frac{\sum_{i=1}^{K} [B_{i,j} \neq 0]}{K} \quad j = 1, ..., p.$$

Note $FV_j \in [0, 1] \Rightarrow$. Values near 1 indicate agreement of the predictor's importance, those near 0 suggest irrelevance of the predictor, while those near 0.5 hint for disagreement across the regression methods regarding the predictor's importance.

Orchestrated parameters tuning algorithm

Our goal is to develop a hyperparameter optimization approach that uses an ensemble of regression methods to perform simultaneous and cooperative hyperparameter tuning **to improve the similarities across different methods**.

Let us denote M_i^t the regression methods M_i applied at an arbitrary iteration t, P_i^t the current vectors of paramenters and $\hat{\beta}_{j,t}$ the corresponding estimated vector of coefficients.

The following sequence of steps are performed until convergence:

1. Consider the set of coefficient vectors $\hat{\beta}_{j,t-1}$ estimated by all other regression methods $M_{j\neq i}^{t-1}$ for the previous iteration $t-1$. Define the **target coefficient vector** $\beta_{i,t}^* = \frac{1}{k-1}\sum_{j\neq i}\hat{\beta}_{j,t-1}$
2. Consider the candidate hyperparameter values located in search grid proximity to P_i^{t-1}.
3. Choose as P_i^t the ones that maximizes correlation between the estimated coefficient vector and the target vector $\beta_{i,t}^*$. Then, estimate $\hat{\beta}_{i,t}$ using P_i^t

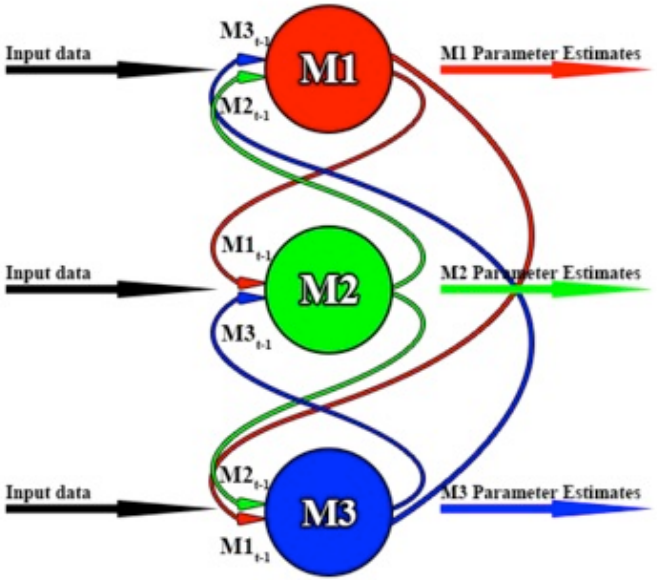
Note

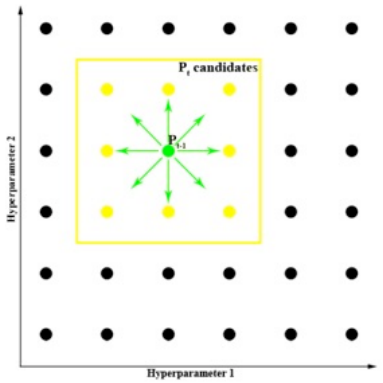
In order to reinforce sparsity, median can be used for definig the target vector and Jaccard index instead of correlation.

Orchestrated parameters tuning in figures

- Initially, each method M_i is trained using some hyperparameter combinations in their respective search grids, forming the method M_i^0 for the iteration at time t=0, for i=1,…,K
- Then, each consecutive iteration $t=1,\ldots,T_{max}$ of the tuning process, the method attempts to increase the similarity between the coefficients estimated by M_i^t and the coefficients of the other regression approaches M_j^{t-1} (for j=1…,i-1,i+1,..K)
- At each time **t** the optimization is carried out in a n neighboring of the previous parameter values P_i^{t-1}

Several, initializing choices and stopping criteria are discussed



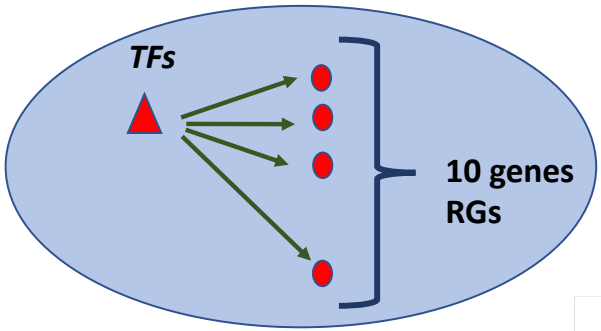


Some preliminary results

Standard versus orchestrated approaches were evaluated in simuation:

- ✓ **50** independent **TFs** regulate **10** independent genes,
- ✓ **only 4 modules were active** (i.e., associated to the response Y)

$$X = [TF_1|RG_{1,1}|...|RG_{1,10}|TF_2|RG_{2,1}|...|RG_{2,10}|TF_3|...|RG_{49,10}|TF_{50}|RG_{50,1}|...|RG_{50,10}]$$



The resulting network contains 550 nodes and 500 edges

p=550 and n=300

Active modules have non zero regression coefficients,
The regulation can be either **positive** or **negative** and with **different strenght**
We considered **4** different scenarios *(A,B,C,D)*

$$X_{TF_j} \sim N(\mu = 0, \sigma = 1)$$

$$X_{RG} \sim N(\mu = 0.7 * X_{TF}, \sigma = 0.71)$$

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \quad \epsilon \sim N\left(0, \sqrt{\frac{\sum_{j=1}^{p} \beta_j^2}{4}}\right)$$

Set-up similar to Li and Li 2008

A

$$\beta = (\ 5, \overbrace{\tfrac{5}{\sqrt{10}}, \ldots, \tfrac{5}{\sqrt{10}}}^{10\text{ genes}}, \overbrace{-5, \tfrac{-5}{\sqrt{10}}, \ldots, \tfrac{-5}{\sqrt{10}}}^{10\text{ genes}}, 3, \tfrac{3}{\sqrt{10}}, \ldots, \tfrac{3}{\sqrt{10}}, -3, \tfrac{-3}{\sqrt{10}}, \ldots, \tfrac{-3}{\sqrt{10}}, 0, \ldots, 0) \quad (26)$$

B

$$\beta = (\ 5, \tfrac{-5}{\sqrt{10}}, \tfrac{-5}{\sqrt{10}}, \tfrac{-5}{\sqrt{10}}, \overbrace{\tfrac{5}{\sqrt{10}}, \ldots, \tfrac{5}{\sqrt{10}}}^{7\text{ genes}}, -5, \tfrac{5}{\sqrt{10}}, \tfrac{5}{\sqrt{10}}, \tfrac{5}{\sqrt{10}}, \tfrac{-5}{\sqrt{10}}, \ldots, \tfrac{-5}{\sqrt{10}}, 3, \tfrac{-3}{\sqrt{10}}, \tfrac{-3}{\sqrt{10}}, \tfrac{-3}{\sqrt{10}}, \tfrac{3}{\sqrt{10}}, \ldots, \tfrac{3}{\sqrt{10}}, -3, \tfrac{3}{\sqrt{10}}, \tfrac{3}{\sqrt{10}}, \tfrac{3}{\sqrt{10}}, \tfrac{-3}{\sqrt{10}}, \ldots, \tfrac{-3}{\sqrt{10}}, 0, \ldots, 0) \quad (27)$$

C

$$\beta = (\ 5, \overbrace{\tfrac{5}{10}, \ldots, \tfrac{5}{10}}^{10\text{ genes}}, \overbrace{-5, \tfrac{-5}{10}, \ldots, \tfrac{-5}{10}}^{10\text{ genes}}, 3, \tfrac{3}{10}, \ldots, \tfrac{3}{10}, -3, \tfrac{-3}{10}, \ldots, \tfrac{-3}{10}, 0, \ldots, 0) \quad (28)$$

D

$$\beta = (\ 5, \tfrac{-5}{10}, \tfrac{-5}{10}, \tfrac{-5}{10}, \overbrace{\tfrac{5}{10}, \ldots, \tfrac{5}{10}}^{7\text{ genes}}, -5, \tfrac{5}{10}, \tfrac{5}{10}, \tfrac{5}{10}, \tfrac{-5}{10}, \ldots, \tfrac{-5}{10}, 3, \tfrac{-3}{10}, \tfrac{-3}{10}, \tfrac{-3}{10}, \tfrac{3}{10}, \ldots, \tfrac{3}{10}, -3, \tfrac{3}{10}, \tfrac{3}{10}, \tfrac{3}{10}, \tfrac{-3}{10}, \ldots, \tfrac{-3}{10}, 0, \ldots, 0) \quad (29)$$

Some preliminary results

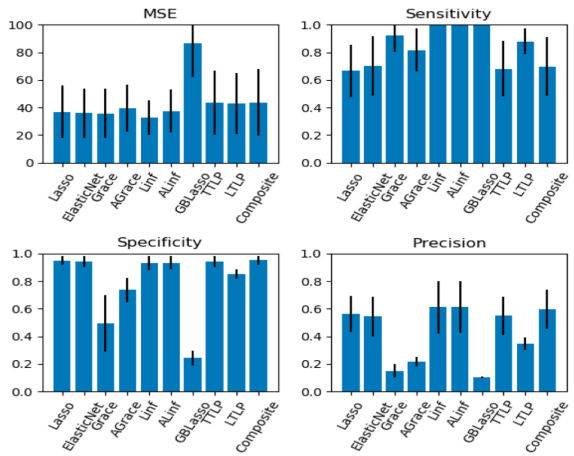


Figure 2. CV-MSE tuning mean model metrics with standard deviation error bars for 20 synthetic datasets

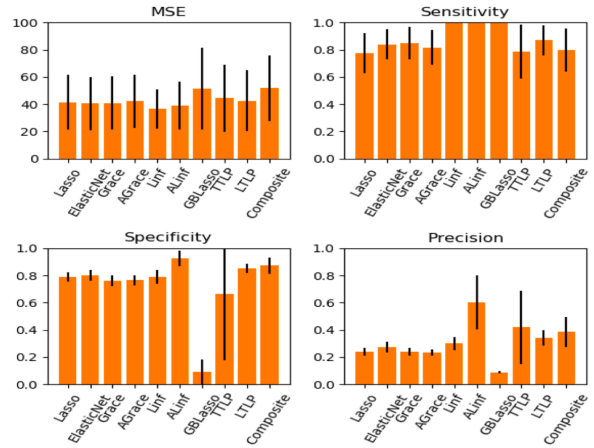


Figure 3. Orchestrated tuning mean model metrics with standard deviation error bars for 20 synthetic datasets

We have evaluated

- ✓ individual approaches versus composite
- ✓ Individual tuning versus orchestrate tuning

With respect to:

1) MSE
2) Sensitivity
3) Specificity
4) Precision/accuracy
5) Correlation between pairs of methods

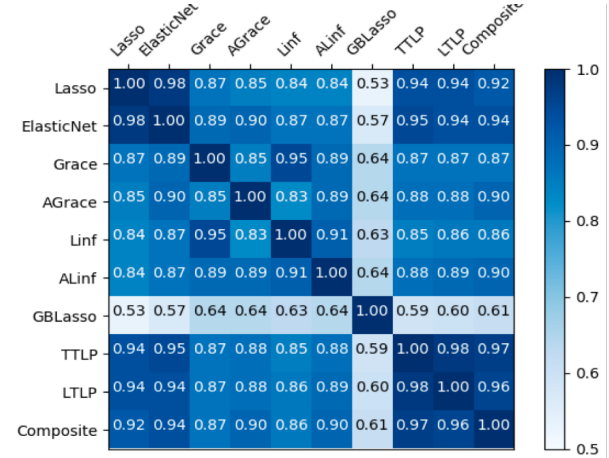


Figure 4. CV-MSE tuning mean regression method similarities

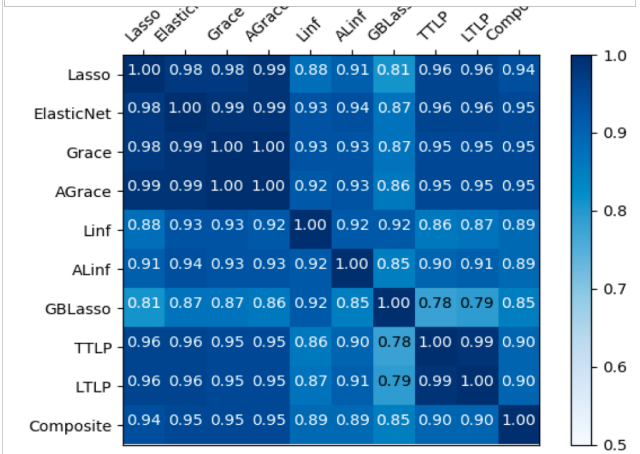


Figure 5. Orchestrated tuning mean regression method similarities

Other considerations

- Extension to other regression contexts (GLM, Cox-regression), in particular **Cox-regression** is interesting due to the potentiality of predicting survival in cancer studies
- Practical applications to genomic data requires a **pre-selction** of a set of genes of interest to reduce both the dimensionality and the nextwork complexity to a moderate scale before applying the penalized algorithm. $\Rightarrow$ **Variable screening** (Fan and Lv, 2008, Fan et al., 2009) might provide a useful framework to achieve such aim.
- Different types of omics data are often measured on the same samples. In many cases they might refers to the same biological variables, i.e., genes (expression, methylation, SNPs, etc) sharing the same network structure. **Integrating multiomic data** is expected to improve the inference.

Network penalized approaches & Cox-Regression

Cox-Regression

Cox model describes the relationship between the patient's survival times and predictor variables (i.e. molecular and clinical information) (Cox, 1972).

Let denote

- T_i : Survival time $\Rightarrow t_i = \min(T_i, C_i)$: **observed survival time**
- C_i : Censoring time $\Rightarrow \delta_i = I(T_i \leq C_i)$: **censoring indicator** such that $\delta_i = 1$ if the survival time is observed $\delta_i = 0$ if the survival time is censored
- $\mathbf{X}_i^T = (x_{i,1}, ..., x_{i,p})^T \in R^p$ patient specific **genome-wide profile**

Assume we observe $(t_i, \delta_i, \mathbf{X}_i^T) \quad i = 1, ..., n$, then **hazard function** $h(t)$ is modeled as

$$h(t|\mathbf{X}_i) = h_0(t)\exp\Big(\sum_{j=1}^{p} x_{i,j}\beta_j\Big) = h_0(t)\exp\Big(\mathbf{X}_i^T\boldsymbol{\beta}\Big)$$

$h_0(t)$: **baseline hazard function** that describes the risk for $\mathbf{X}_i = 0$,

$\exp\Big(\sum_{j=1}^{p} x_{i,j}\beta_j\Big)$: **relative risk**, i.e., proportional to increase or decrease of x_{ij}

$\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^T$: the vector of coefficients.

Cox-regression

- **Classical settings**: the regression parameters are estimated by maximizing the **Cox's partial likelihood**, or equivalently, by minimizing **Cox's log-partial likelihood** $-l(\beta)$

$$\hat{\beta} = \text{argmin}_{\beta}\left[-l(\beta)\right] = \text{argmin}_{\beta}\left[-\frac{1}{n}\sum_{i=1}^{n}\delta_i\left\{\mathbf{X}_i^T\beta - \log\left[\sum_{j\in R(t_i)}\exp(\mathbf{X}_j^T\beta)\right]\right\}\right]$$

  where $R(t_i)$ denotes the risk set at time t_i (i.e., the set of all patients who still survived prior to time t_i).
- **Penalized approaches**

$$l_{pen}(\beta) = -l(\beta) + P_\lambda(\beta)$$

Note

Classical Penalized approaches $P_\lambda(\beta)$ can be the extended to Cox-regression see, LASSO (Tibshirani, 1997, Gui and Li 2005); Elastic-net (Engler and Li, 2009,Simon et al. 2011, Wu 2012); SCAD (Fan and Li, 2002); adaptive Lasso (Zhang and Lu, 2007); Dantzing selector (Antoniadis et al. 2010), however extension is not straightforward due to the semiparametric nature of Cox-Regression.

Laplacian Net, Adaptive Laplacian net and Absolute Laplacian net (Sun, et al 2014)

It naturally extends **Grace** and **aGrace** to Cox-regression.

Laplacian Net (Lnet):

$$P_\lambda(\beta) = \lambda_1 \sum_{u=1}^{p} |\beta_u| + \lambda_2 \sum_{u\sim v} w_{u,v}\left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_v}{\sqrt{d_v}}\right)^2 = \lambda_1 ||\beta||_1 + \lambda_2 \beta^T \mathbf{L} \beta$$

where **L** is the **Normalized Laplacian matrix**,

Adaptive Laplacian net (AdaLnet) $P_\lambda(\beta) =$

$$\lambda_1 \sum_{u=1}^{p} |\beta_u| + \lambda_2 \sum_{u\sim v} w_{u,v}\left(\frac{\mathrm{sgn}(\tilde{\beta}_u)\beta_i}{\sqrt{d_u}} - \frac{\mathrm{sgn}(\tilde{\beta}_v)\beta_v}{\sqrt{d_v}}\right)^2 = \lambda_1 ||\beta||_1 + \lambda_2 \beta^T \mathbf{L}^* \beta$$

Absolute Laplacian net

$$P_\lambda(\beta) = \lambda_1 \sum_{u=1}^{p} |\beta_u| + \lambda_2 \sum_{u\sim v} w_{u,v}\left(\frac{|\beta_u|}{\sqrt{d_u}} - \frac{|\beta_v|}{\sqrt{d_v}}\right)^2 = \lambda_1 ||\beta||_1 + \lambda_2 |\beta|^T \tilde{\mathbf{L}} |\beta|$$

Lnet and **AdaLNet** are convex penalties, therefore the solution can be efficiently computed using Coordinate Descendent algorithms. The Absolute Laplacian net is not convex, therefore it poses both theoretical and computational challenges.

A different algorithm for solving adaptive/laplacian network is given by Alternating Direction method of Multipliers - **ADMMNET** (Boyde, 2011)

Other approaches

- **Net-Cox (Zhang et al., 2013)**: It smooths the coefficients over the network, but it does not perform variable selection

$$P_\lambda(\beta) = \lambda_1 \sum_{u=1}^{p} \beta_u^2 + \lambda_2 \sum_{u \sim v} w_{u,v} \left(\frac{\beta_u}{\sqrt{d_u}} - \frac{\beta_v}{\sqrt{d_v}} \right)^2 = \lambda_1 ||\beta||_2 + \lambda_2 \beta^T \mathbf{L} \beta$$

  It can be computed very efficiently using Newton-Raphson method.

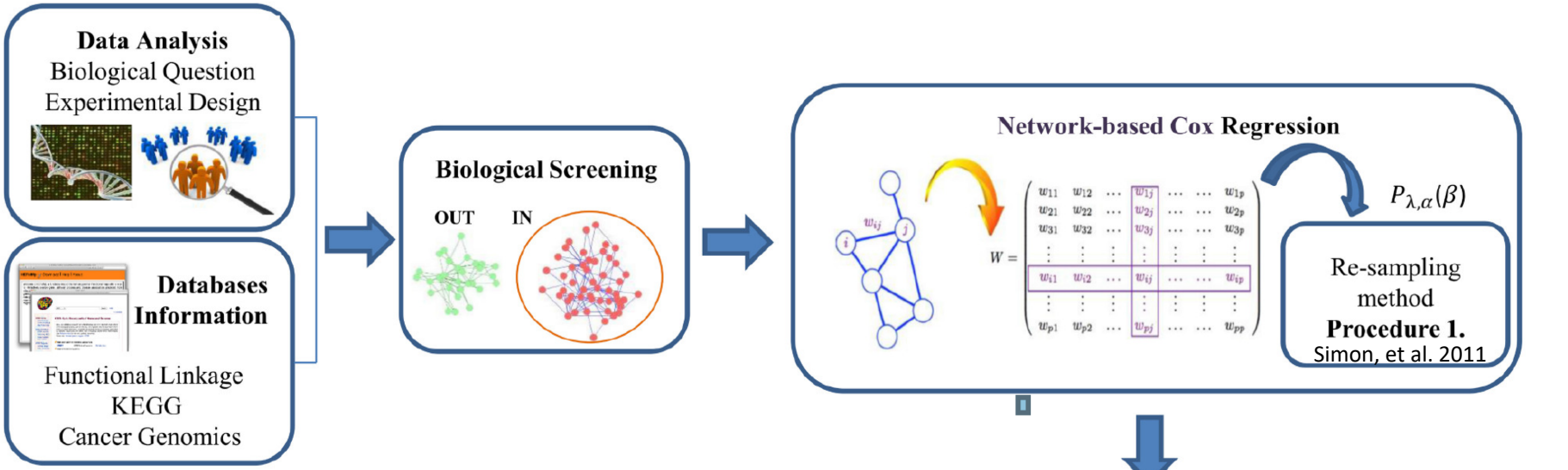
- **DrCOX (Wu et al., 2013; Gong et al., 2014)** It uses grouped-lasso penalization on the a partition $P_1, ...P_K$ on the genes in K pathways, as follows

$$P_\lambda(\beta) = \lambda_1 \sum_{k=1}^{K} \sum_{u \in P_k} |\beta_u| + \lambda_2 \sum_{k=1}^{K} \sqrt{\sum_{u \in P_k} \beta_u^2} = \lambda_1 \sum_{k=1}^{K} ||\beta_k||_1 + \lambda_2 \sum_{k=1}^{K} ||\beta_k||_2$$

  Then it can be adapted for handling the case of overlapping partitions. Moreover, it can be solved using Coordinate descendent method.

- $L_{1/2}$**-Laplacian Network and its adaptive version: (Jiang and Liang, 2018)** $P_\lambda(\beta) = \lambda_1 ||\beta||_{1/2} + \lambda_2 \beta^T \mathbf{L} \beta$

Network penalized regression in practical applications



Typical results are

1. Gene signature → vector of **non-zero coefficients**
2. Network/pathway relationship (i.e., **sub-modules**)
3. Prognostic index ***PI**** to be used for frediction

- ✓ **Training the algorithm on a dataset**
- ✓ **Validate the findings on one or more independent datasets**

Some results

A. Iuliano, A. Occhipinti, C. Angelini, I. De Feis, P. Liò. *Cancer markers selection using network-based Cox regression: A methodological and computational practice.* Frontiers in physiology 7, 208, (2016).

Network penalized regression in practical applications

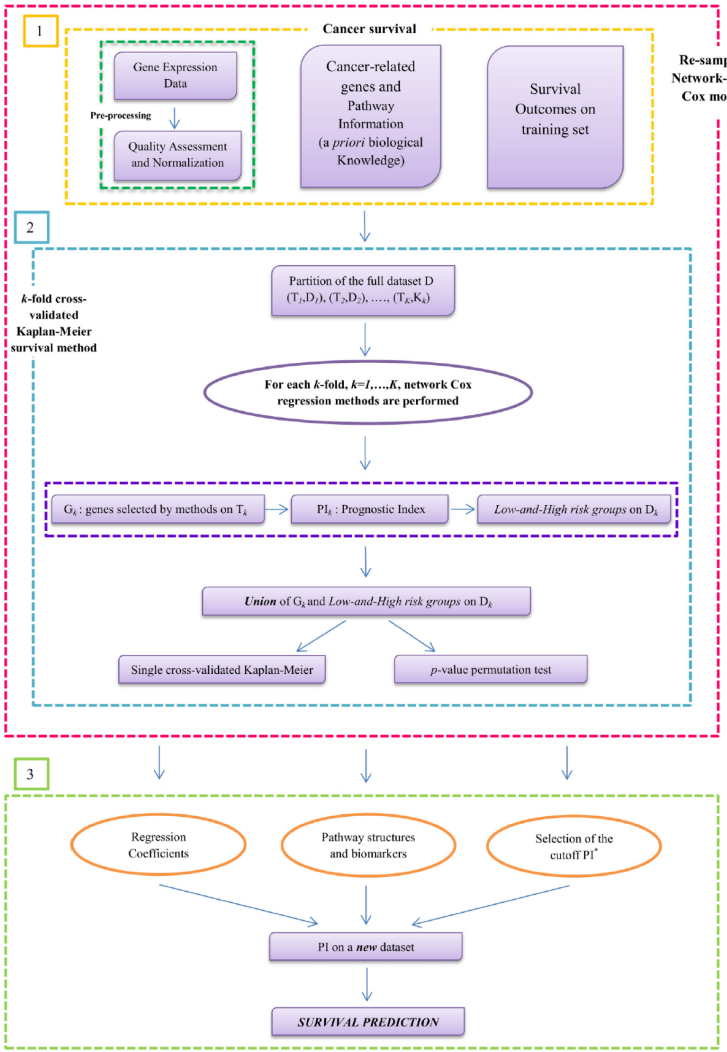
On a Training-set: T

1. Pre-selection of a subset of potential relevant genes based on the significance of the relation between the gene and the disease of interest using the functional linkage database **HEFalMp** (Huttenhower et al., 2009) and evaluation of **L**
2. Network-based penalized regression using **Net-Cox**, **fastCox** (i.e., elastic-net for cox-regression), and **AdaLnet**
3. Identification of a subset of potential biomarkers, or **gene signature S**, i.e., $\hat{\beta}_I \neq 0 \quad I \in S$ and estraction of the **network sub-modules**
4. Given the identified **gene signature** ($\hat{\beta}_I \neq 0 \quad I \in S$) evaluation of the **prognostic index** $PI_i = \mathbf{X}_i^T \hat{\beta}; \quad i \in T$ and identification of a cut-off PI^* for distinguish between **low** and **high-risk patients**

On a Validation-set: V

1. Evaluation of the prognostic index $PI_i; \quad i \in V$ on the novel patients and comparision with PI^*.
2. Significance assessed using **Kaplan-Meier** and **log-rank test**

Algorithm in figures



Training-set **T**

The training T set is divided in k-fold $(T_1,D_1),\ldots,(T_k,D_k)$

The learning process is performed independently on each fold, then results are combined in order to obtain

1. **Gene signature** (i.e,, vector of coefficients)
2. Corresponding **network/pathway** structure
3. Prognostic index cut-off ***PI**** for distinguish between ***low and high-risk patients***

Note: there are two nested cross-validation loops.
a) an internal loop for selecting the regularization parameters on each (T_l,D_l);
b) an external loop for selecting *PI** (based on Simon, et al. 2011 procedure for evaluating the predictive accuracy of survival risk classifiers)

Validation-set **V** (usually an independent dataset)

Novel patients are divided in low and high-risk groups on the basis of their prognostic index
The significance of the results are assessed using **Kaplan-Meier** curves and the **log-rank test**

Some results

TABLE 1 | Microarray Dataset Summary (OS = overall survival).

Datasets	Ref.	Sample number	Platform	Genes number
GSE26712	Bonome et al., 2008	185	Affymetrix U133A	13104
OV-TCGA	The Cancer Genome Atlas Research Network, 2011	578	Affymetrix U133A	13104
GSE20685	Kao et al., 2011	327	Affymetrix U133Plus2	21686
GSE7390	Desmedt et al., 2007	198	Affymetrix U133A	13718

TABLE 2 | Significant genes number selected using HEFalMp tool.

Datasets	Genes number
GSE26712	1068
OV-TCGA	1068
GSE20685	536
GSE7390	536

TABLE 5 | Optimal α cross-validated value calculated on the k training sets.

Datasets	k Partitions	Net-Cox		AdaLnet		fastcox	
		α	Genes selected	α	Genes selected	α	Genes selected
GSE26712	5	0.2	101	0.5	23	0.01	453
OV-TCGA	5	0.5	99	0.5	38	0.1	623
GSE20685	5	0.5	76	0.5	28	0.01	298
GSE7390	5	0.5	89	0.5	14	0.01	423

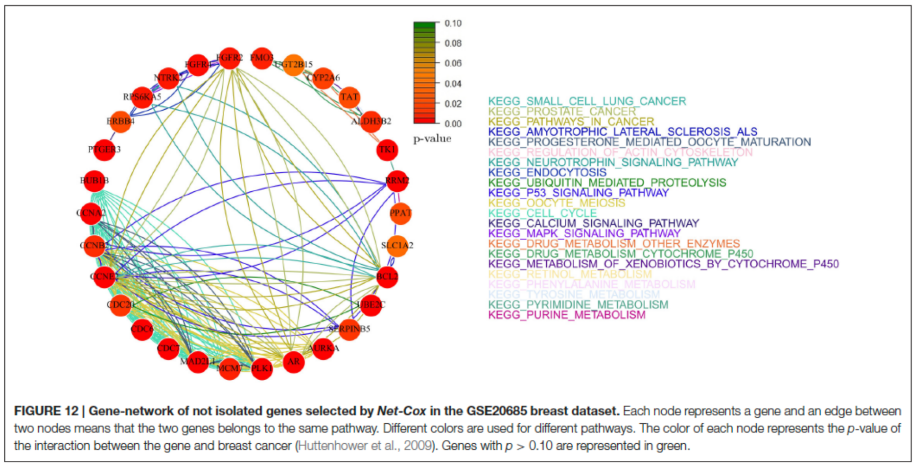


FIGURE 12 | Gene-network of not isolated genes selected by *Net-Cox* in the GSE20685 breast dataset. Each node represents a gene and an edge between two nodes means that the two genes belongs to the same pathway. Different colors are used for different pathways. The color of each node represents the p-value of the interaction between the gene and breast cancer (Huttenhower et al., 2009). Genes with $p > 0.10$ are represented in green.

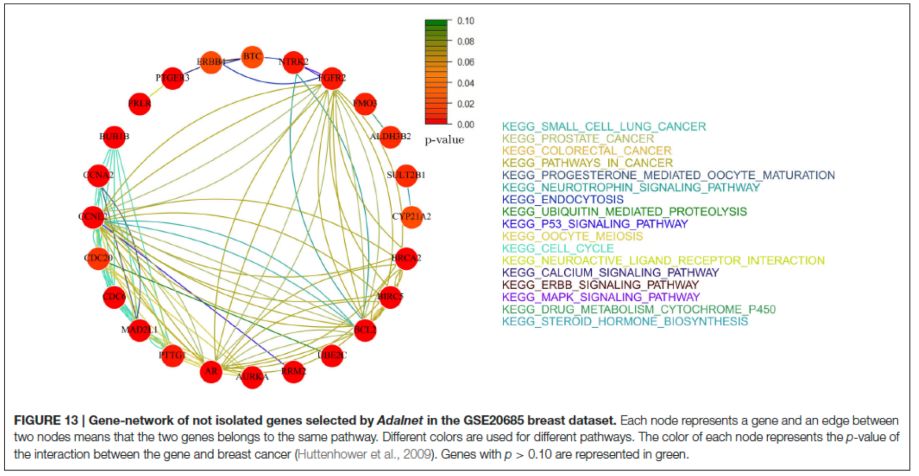


FIGURE 13 | Gene-network of not isolated genes selected by *Adalnet* in the GSE20685 breast dataset. Each node represents a gene and an edge between two nodes means that the two genes belongs to the same pathway. Different colors are used for different pathways. The color of each node represents the p-value of the interaction between the gene and breast cancer (Huttenhower et al., 2009). Genes with $p > 0.10$ are represented in green.

Statistical screening & network penalized regression

A two-stage computational-statistical model to identify potential biomarkers and predict patient clinical outcomes by using cancer omics data.

This novel approach is based on the two following steps:

A. **Dimensional reduction** step to reduce the number of variables from a high-dimensional space p to a moderate-dimension space d by using different types of **screening** techniques:

1. Biomedical-driven screening $\Rightarrow$ BMD-screening;
2. Data-driven screening $\Rightarrow$ DAD-screening;
3. Combination of BMD-and-DAD-screening $\Rightarrow$ BMD+DAD-screening;

B. **Network-regression** step to incorporate an a-priori biological knowledge into the model by applying network penalized Cox regression methods:

1. AdaLNet
2. ADMMNET

See

A. Iuliano, A. Occhipinti, C. Angelini, I. De Feis, P. Liò. *Combining pathway identification and breast cancer survival prediction via screening-network methods*. Frontiers in Genetics 9, (2018).

Variable Screening Step

The aim of the variable screening step is to reduce the number of variables form a high- to a moderate scale. i.e., identify a subset

$$\{x_j; \quad j \in \mathcal{I}\}$$

as the subset of the **screened variables** such that

$$d = |\mathcal{I}| < p$$

BMD-screening: It uses information available from the literature

$$\mathcal{I}_{BMD} = \{1 \leq k \leq p : p_k < 0.05\}$$

where p_k is the **p-value** measuring the significance of the relation between the gene and the disease of interest computed according to **HEFalMp**

HEFalMp: (Human Experimental/Functional Mapper, Huttenhower et al., 2009) is a database containing maps describing the genes functional activity and interaction networks in over 200 areas of human cellular biology with information from about 30.000 genome-wide experiments of different types.

Variable Screening Step

Following the idea of Fan et al., 2008 about **statistical screening**,

DAD-screening: It considers evidence emerging from the data

$$\mathcal{I}_{DAD} = \{1 \leq k \leq p : |\beta_k^M| \geq \delta_n\},$$

where δ_n is a threshold chosen to pick the top ranked covariates and β_k^M is the maximum marginal likelihood estimator (MMLE), i.e., the maximizer of the log-partial likelihood with a single covariate

$$\beta_k^M = \arg\max_{\beta_k} \sum_{i=1}^{n} \delta_i \left\{ \mathbf{x}_{ki}^T \beta_k - \log \left[\sum_{j \in R(t_i)} \exp(\mathbf{x}_{kj}^T \beta_k) \right] \right\}, \quad k = 1, \ldots, p.$$

DAD-screening can be easily generalized to **Pathway screening** (Gong, et al. 2014), or **SKY-Cox** (Liu, et al. 2017)

BMD+DAD-screening: It combines information available from the literature with novel evidence emerging from the data

$$\mathcal{I}_{BMD+DAD} = I_{BMD} \cup I_{DAD}$$

Algorithm

On a Training-set: T

1. Selection of the screened variables using **BMD, DAD** or **BMD+DAD screening**; evaluation of **L** using the functional linkage database **HEFalMp** (Huttenhower et al., 2009)
2. Network-based penalized regression using **AdaLnet** method with either a) **Coxnet** or b) **ADMMNET** algorithm. Cross-validation for estimating the regularization parameters.
3. Identification of a subset of potential biomarkers or **gene signature S** , i.e., $\hat{\beta}_I \neq 0 \quad I \in S$
4. Given the identified **gene signature** ($\hat{\beta}_I \neq 0 \quad I \in S$) evaluation of the **prognostic index** $PI_i = \mathbf{X}_i^T \hat{\beta}; \quad i \in T$ and identification of a cut-off PI^* for distinguish between **low** and **high-risk patients**

On a Validation-set: V

1. Evaluation of the prognostic index $PI_i, \quad i \in D$ on the novel patients and comparison with PI^*
2. Significance assessed using **Kaplan-Meier** and **log-rank test**

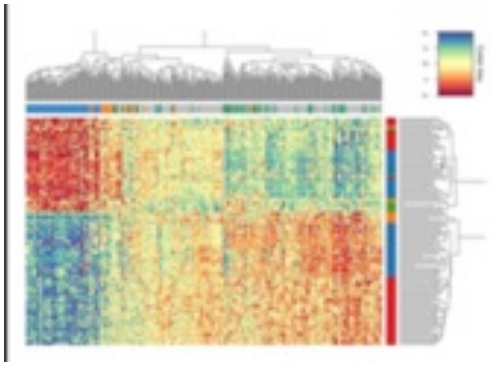
The analysis of METABRIC dataset

The dataset

Breast Cancer (METABRIC, Nature 2012 & Nat Commun 2016)

The genomic profiles (somatic mutations [targeted sequencing], copy number alterations, and gene expression)

It contains genome-wide profiles of about **2000** samples



Long-term follow-up data:

- **OS-MONTHS** (Q1 = 60:78, Median = 116:10, Q3 = 184:90);
- **OS-STATUS**: Died of Disease=1, Living=0, Died of Other Causes=0.

Omics data:

- **mRNA expression** data (mRNA);
- **copy-number aberrations** data from DNA copy (CNAs).

Omics data	Training set (*T*)		Testing set (*D*)	
	Samples	*# Genes*	*Samples*	*# Genes*
mRNA	997	19151	995	19151
mRNA+CNAs	997	18006	995	18006

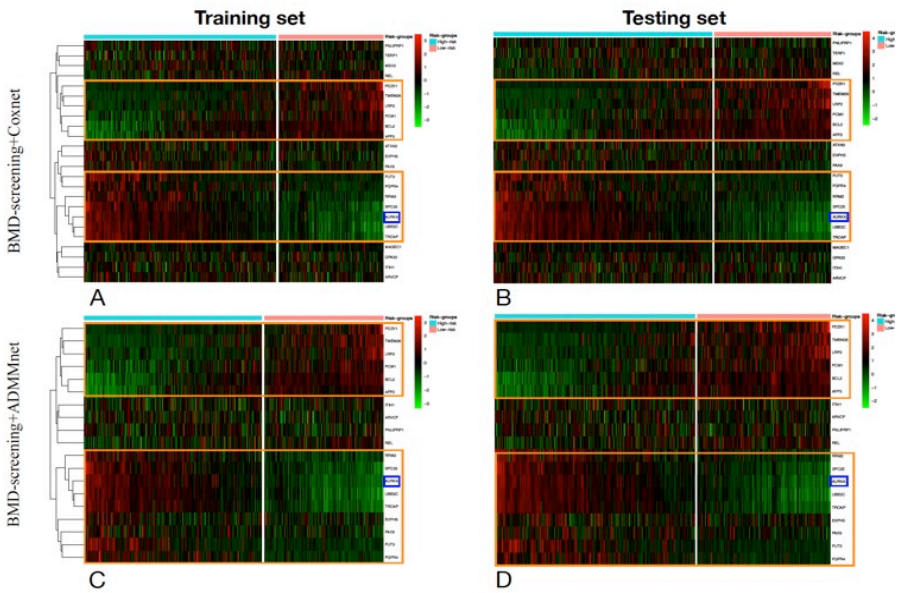
→ Raw data were pre-processed, normalized between arrays and standardized, before applying the proposed two-stage procedure

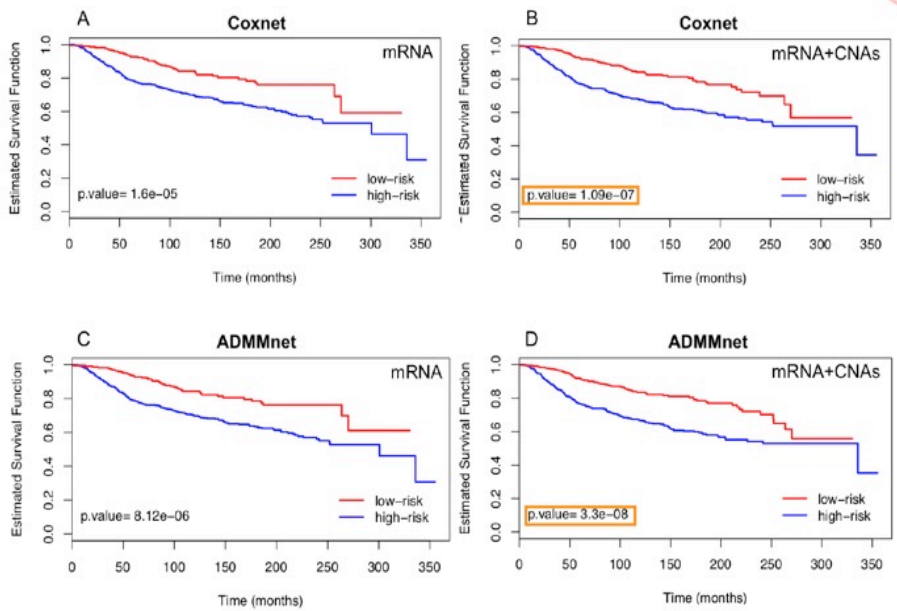
→ In our study, mRNA and CNAs data were integrated by using **MANCIE** package (Zang et al., 2016) to obtain a single integreted predictor matrix **X**.

Results

Omics data	Methods	# BMD-genes	*p*-value	α	λ	$\mathcal{I}_{BMD}$
mRNA	Coxnet	38	1.6e-05	0.5	0.07934	528
	ADMMnet	43	8.12e-06	0.5	0.07695	
mRNA+CNAs	Coxnet	24	1.09e-07	0.5	0.09338	526
	ADMMnet	19	3.3e-08	0.5	0.10170	

BMD-Screening





Red color indicates a *high level* of expression in breast cancer a **green color** indicates a *low level of expression*.

- *Kaplan-Meier curves* obtained on the testing set *D*. The patients are divided in **high**-and-**low** risk group according to the PI_D.

Results

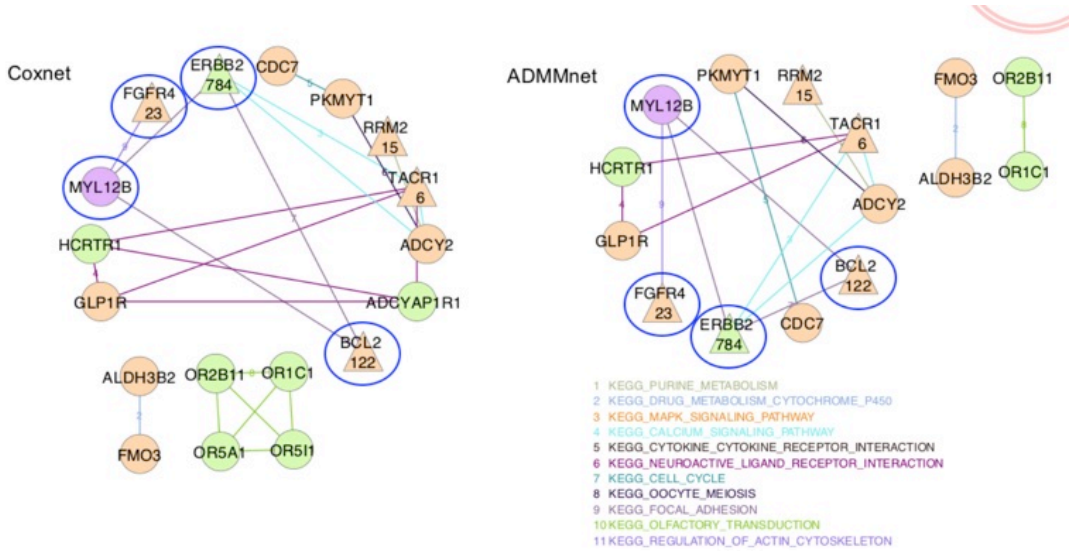
The **BMD+DAD-screening** outperforms (in terms of *p-value*) the other two screenings allowing:

- better separation between high-and-low-risk groups;
- identification of novel potential biomarkers.

Moreover, results results confirm that the prediction capability improves when two omic layers (mRNA + CNAs) are used instead of a single omic layer (mRNA).

orange color for genes-HEFalMp-high (p<0.05),
green color for genes-HEFalMp-low (p>0.05),
purple color for genes-no-HEFalMp

Triangular-shaped nodes indicate the genes identified in literature as breast-cancer associated genes. The number of papers is also reported in the triangular nodes.



Most of the genes (i.e., orange nodes) confirm that the BMD+DAD screening identifies well-known breast cancer related genes, however it also identifies potentially novel genes (green and purple) nodes.
In particular **MYL12B** is a high risk mutated gene detected by COSMIC and it is predominantly expressed in Triple-Negative Breast Cancer

Conclusions and What Next?

Conclusions

- **Network-penalized approaches** constitute **a valid and general framework** to integrate in a regression model biological information available in form of network/graph $\rightarrow$ They allows to identify sub-modules of connected variables that are much likely to belong to the same biological pathway.
- **Statistical screening** approaches can be used to pre-select a subset of genes/elements of interest allowing to reduce the computational effort and improve the quality of the selection $\rightarrow$ **BMD+DAD-screening** provides better results and allowes to identify novel potential biomarkers.
- **Integrating data** from multiple omic sources improves the prediction and opens novel interesting statistical challenges

Cosmonet R package

Iuliano et al. COSMONET: A R package for survival analysis using screening-network methods. In preparation (2019), **Have a look at poster session**.

What Next? Some ideas for future works

- **Incorporating patient clinical information to remove potentially confounding/interacting variables**

$\Rightarrow$ Extending the model $f(\mathbf{X}, \beta, \mathbf{Z}, \delta, \epsilon)$ considering $\mathbf{X}$ and $\mathbf{Z}$ (with possible interactions)

- **Multi-omics data integration** and/or **Multi-studies data integration**:
  $\mathbf{Y}^r = f(\mathbf{X}^r, \beta^r, \epsilon) \quad r = 1, ..., R$

where $\mathbf{X} \in R^{n_r \times p_r}$ can be either the different omics matrices on the same samples or the same omic across different studies, or both possibilities (see, Lin et al., 2014). Assume that most of the variables (i.e., genes) are measured across all/most of the omics/studies

$\Rightarrow$ **Sparse-group multi-task regression + Network penalization** could constitute a suitable statistical framework for data integration (Cao et al, 2018)

- **Comparison with other approaches/frameworks**

$\Rightarrow$ e.g., **Bayesian approaches** provide an alternative framework for embedding network information in regression (Stingo et al. 2011, Peterson et al., 2016), or for multi-omic data integration (Chekouo et al., 2017)

- **Joint network estimation and network penalization**

ACKNOWLEDGEMENTS

BioInfoLaB team at IAC-CNR

- Antonella Iuliano
- Italia De Feis
- All other lab members for useful discussion (Monika, Dario, Anna, Eugenio,...)

Cambridge University

- Pietro Liò
- Annalisa Occhipinti

Technical University of Varna

- Sivo Daskalov
- Kristina Bliznakova

COST Action CA15109.
COST is supported by the
EU Framework Programme
Horizon 2020.