Approches statistiques pour une sélection de variables stable et une modélisation prédictive de processus complexes

Frédéric Bertrand[1]

[1]Cedric, Conservatoire National des Arts et Métiers
Paris, France

05th December 2025

le cnam

Sommaire

Contents

SelectBoost (Bertrand, 2020) is a new general algorithm which improves the precision of any existing variable selection method.

- This algorithm is based on highly intensive simulations and takes into account the correlation structure of the data.
- Our algorithm can either produce a confidence index for variable selection or be used to design experiments.
- We showed the performance of our algorithm on both simulated and real data and apply it in two different ways to improve biological network reverse-engineering.

More results and use cases in the article published in *Bioinformatics*.

Setting and Aim

Variable selection is a commonly required step

Technological innovations make it possible to measure large amounts of data in a single observation.

As a consequence, problems in which the number **P** of variables is larger than the number **N** of observations have become common.

As reviewed by Fan and Li (Fan and Li, 2006), such situations arise in many fields from fundamental sciences to social science, and **variable selection is required to tackle these issues**.

A use case of the algorithm:
Robust reverse-engineering of networks.

Sparsity is a well-known feature of most biological networks: a node can only be regulated by a small number of other nodes, whereas it may regulate any number of other nodes.
Hence, variable selection methods for linear models, such as the lasso, ensures that sparsity feature and are often core components of most of the biological network reverse-engineering tools.

As a consequence, we propose to apply the SelectBoost algorithm in two different ways in order to improve the biological network reverse-engineering:

1. as a post-processing step post-inference (Fig. 1) or
2. during the inference itself in order to select the most stable predictors for each node in the network (Fig. 2).

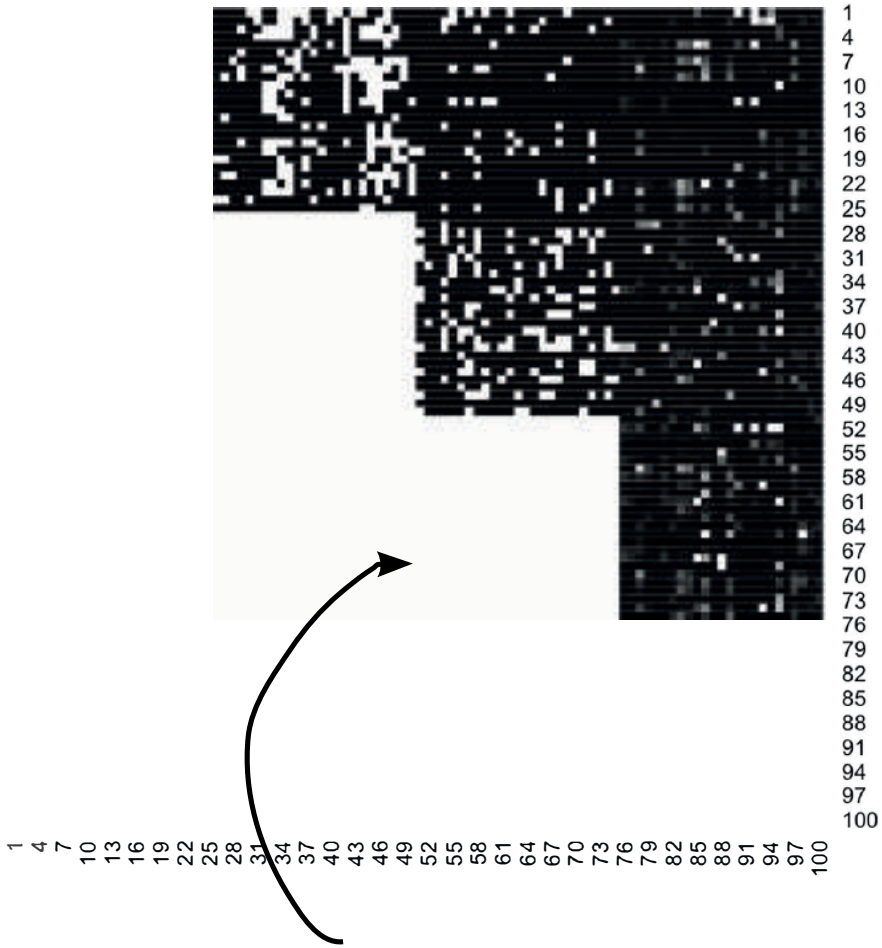


Fig. 1. Post inference analysis of a cascade network. The darker, the lower the confidence.

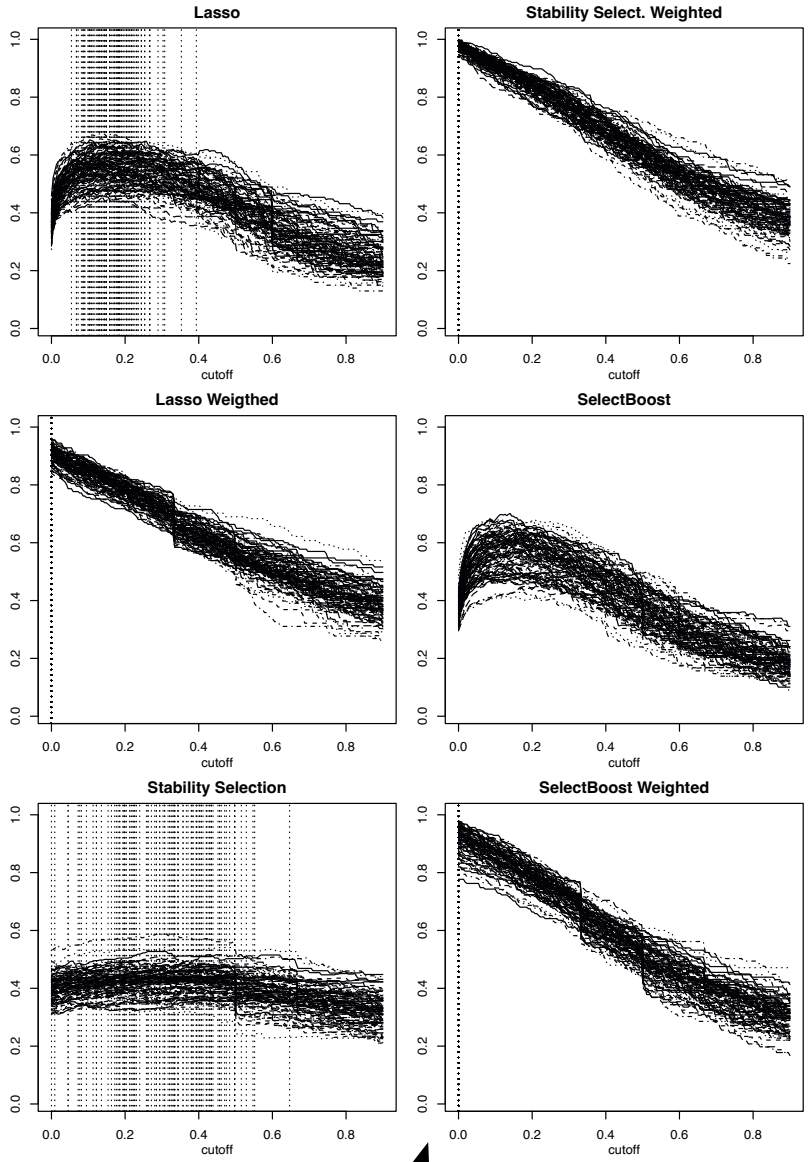


Fig. 2. F-score as a function of the thresholding value for network inference based on the SelectBoost, the stability selection and the regular lasso and their weighted counterparts.

Penalized likelihood 1/2

There is a vast literature dealing with the problem of variable selection in both statistical and machine learning areas (Fan and Lv, 2010).

The main variable selection methods can be gathered in the common framework of penalized likelihood. The estimate $\hat{\beta}$ is then given by:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^P}{\arg\min} \left[-\ell_N(\boldsymbol{\beta}) + \sum_{p=1}^{P} \text{pen}_\lambda(\beta_p) \right]$$

where $\ell_N(.)$ is the log-likelihood function, $\text{pen}_\lambda(.)$ is a penalty function and $\lambda \in \mathbb{R}$ is the regularization parameter.

Penalized likelihood 2/2

As the goal is to obtain a sparse estimation of the vector of parameters β, a natural choice for the penalty function is to use the so-called $\mathcal{L}_0$ norm ($\|.\|_0$) which corresponds to the number of non-vanishing elements of a vector:

$$\begin{array}{rcccl} \text{pen}_\lambda & : & \mathbb{R} & \mapsto & \{0, \lambda\} \\ & & x & \mapsto & \begin{cases} \text{pen}_\lambda(x) = \lambda & \text{if } x \neq 0 \\ \text{pen}_\lambda(x) = 0 & \text{else} \end{cases} \end{array}$$

which induces $\sum_{p=1}^{P} \text{pen}_\lambda(\beta_p) = \lambda \|\beta\|_0$.

For example, when $\lambda = 1$, we get the Akaike Information Criterion (AIC, Akaike1974) and when $\lambda = \frac{\log(N)}{2}$ we get the Bayesian Information Criterion (BIC, Schwartz 1978).

Failure due to linear correlation

The goal of identifying the correct support of the regression is complicated and the reason why **variable selection methods fail** to select the set of non-zero variables can easily be summarized in two words:

linear correlation!

Notations

N the number of observations	P the number of variables
$\mathbf{y} = (y_1, ..., y_N)$ the response variable	
$\mathbf{X} = (\mathbf{x}_{1.}, ..., \mathbf{x}_{P.})$ a variable matrix of size $N \times P$	
$\boldsymbol{\beta} = (\beta_1, ..., \beta_P)'$ a vector of parameters	

Stability selection

How robust is a given selection?

Study the occurrences (or not) of variable selections in a model, for a given penalisation, on random subsamples (Meinshausen, 2010).

Linear model with Lasso estimator or, more generally, of the `glmnet` type.

$\rightarrow$ counts how many times a variable is selected for a given value of the penalisation.

$\rightarrow$ asymptotic convergence properties towards the correct selection.

Contents

Method

Overcome linear correlation

The SelectBoost algorithm has been designed in a general framework to avoid to select non-predictive correlated features. The main goal is to improve the PPV, i.e. the proportion of selected variables which truly belong to the model.

Correlated resampling

The main idea of our algorithm is to consider that any observed value of a group of linearly correlated variables of the $\mathbf{X}$ matrix is the independent realization of a given random function, which is then used to perturb the observed values of the relevant correlated variables.

Sherical distribution

As we assume that the variables are centered and that $\|\mathbf{x}_{p.}\|^2 = 1$ for $p = 1, ..., P$, we know that:

$$\mathbf{x}_{p.} \in \mathcal{S}^{N-2}.$$

As a consequence, we use a spherical distribution for resampling.

For instance, a multivariate Gaussian distribution assumption for $\mathbf{X} = (\mathbf{x}_{1.}, ..., \mathbf{x}_{P.})$ advocates for the use a von Mises-Fisher distribution (Sra, 2012).

The SelectBoost algorithm

Pseudo-code

Algorithm 1 Pseudo-code for the SelectBoost algorithm

Require: $gr_{c_0}, select, B, c_0$

$\boldsymbol{\zeta} \leftarrow \mathbf{0_P}$

for $b = 1, ..., B$ **do**

$\mathbf{X}^{(b)} \leftarrow \mathbf{X}$

for $p = 1, ..., P$ **do**

$\mathbf{x}^{(b)}_{\mathbf{p}.} \leftarrow \phi^{-1}\left(\text{random-vMF}\left(\hat{\boldsymbol{\mu}}(\phi(\mathbf{X}_{\mathbf{gr_{c_0}(p)}})), \hat{\kappa}(\phi(\mathbf{X}_{\mathbf{gr_{c_0}(p)}})\right)\right)$

end for

$\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta} + select(\mathbf{X}^{(b)}, \mathbf{y})$

end for

$\boldsymbol{\zeta} \leftarrow \boldsymbol{\zeta}/B$

The SelectBoost algorithm (Step 1)

1. To use the SB algorithm, we need a grouping method gr_{c_0} depending on a user-provided constant $0 \leq c_0 \leq 1$. It must satisfy:

$$\forall p \in \{1, ...P\} : gr_1(p) = \{p\} \quad \text{and} \quad gr_0(p) = \{1, ...P\}.$$

Let $gr_{c_0}(p)$ be the set of all variables which are considered to be linked to the variable $\mathbf{x}_p$. $\mathbf{X}_{gr_{c_0}(p)}$ is the submatrix of $\mathbf{X}$ with the columns which indices are in $gr_{c_0}(p)$.

The SelectBoost algorithm (Steps 2 & 3)

2. Furthermore, we need to have a selection method:

$$select \quad : \quad \mathbb{R}^{N \times P} \times \mathbb{R}^{N} \quad \rightarrow \quad \{0,1\}^{P}$$

which maps the design matrix $\mathbf{X}$ and the response variable $\mathbf{y}$ to a 0-1 vector of length P with 1 at position p if the method selects the variable p and 0 otherwise.

3. We then use the von Mises-Fisher distribution to generate replacement of the original variables by some simulations to create B new design matrices $\mathbf{X}^{(1)}, ..., \mathbf{X}^{(B)}$.

The SelectBoost algorithm (Steps 4 & 5)

4. The SB algorithm then applies the variable selection method $select$ to each of these matrices and returns a vector of length P with the frequency of apparition of each variable.
   The frequency of apparition of variable $\mathbf{x}_{p.}$, noted ζ_p is assumed to be an estimator of the probability of $\mathbf{x}_{p.}$ being in the true model.
5. The choice of c_0 is crucial. The model might not perturbed enough (too large c_0) or the variables are chosen at random (too small c_0).

The SB algorithm (**Algorithm 1**) returns the vector $\boldsymbol{\zeta} = (\zeta_1, ..., \zeta_P)'$.

Compare these values $\boldsymbol{\zeta} = (\zeta_1, ..., \zeta_P)'$ to a threshold $\zeta_{\min}$ to determine which variables are selected.

Contents

Simulation studies

The choice of the threshold is critical and the algorithm can be improved if we enforce that the ζ_p values are non-increasing functions of c_0, see **Fig. 3**, **4** and **5**.

Fig. 3. displays the evolution of the recall, PPV and F-score as a function of $1 - c_0$ for LASSO based SelectBoost with a nonincreasing post-processing step and a threshold min = 1.

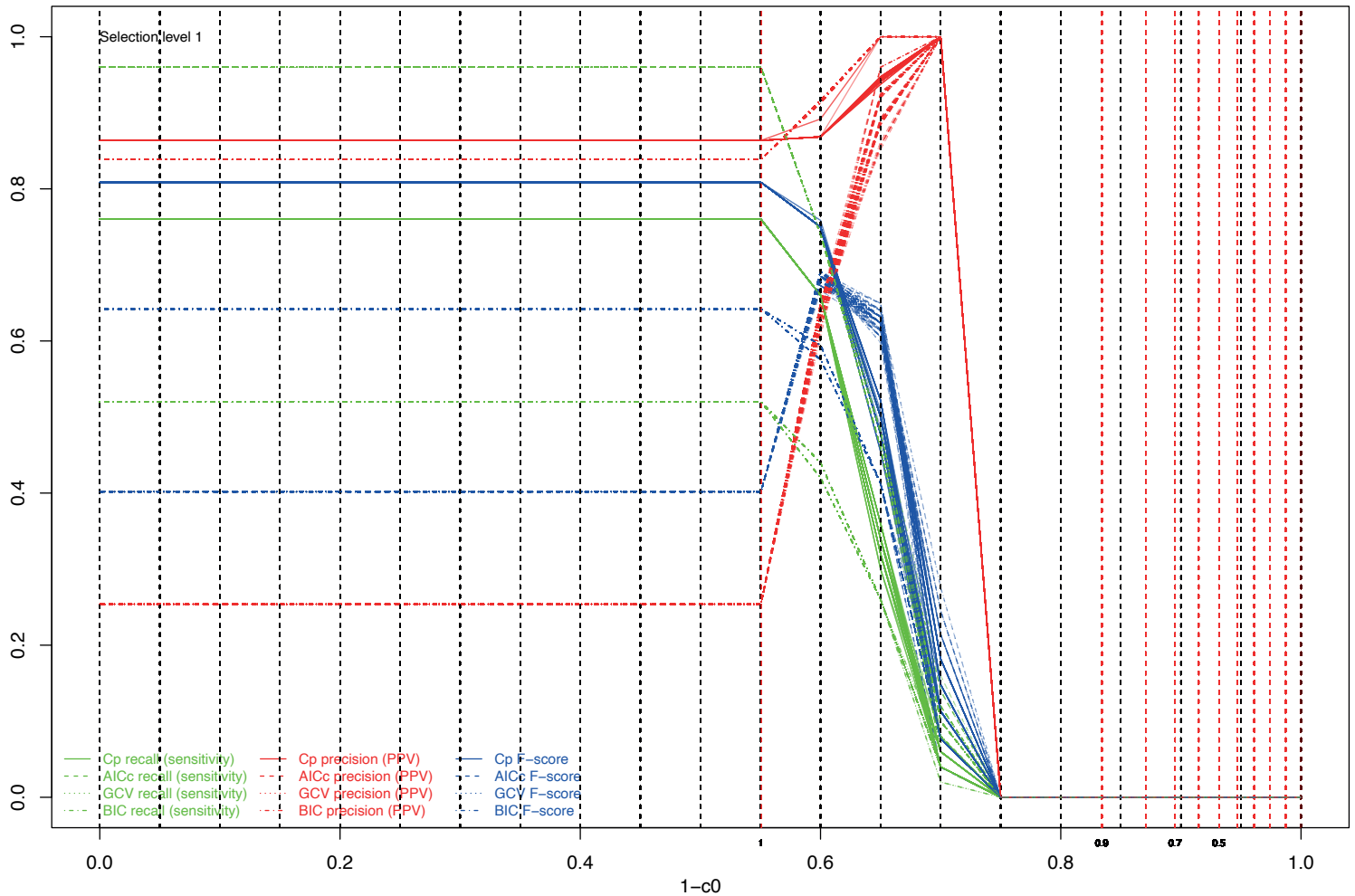


Fig. 3. Evolution of the recall, PPV and F-score as a function of $1 - c_0$.

Average number of identified variables

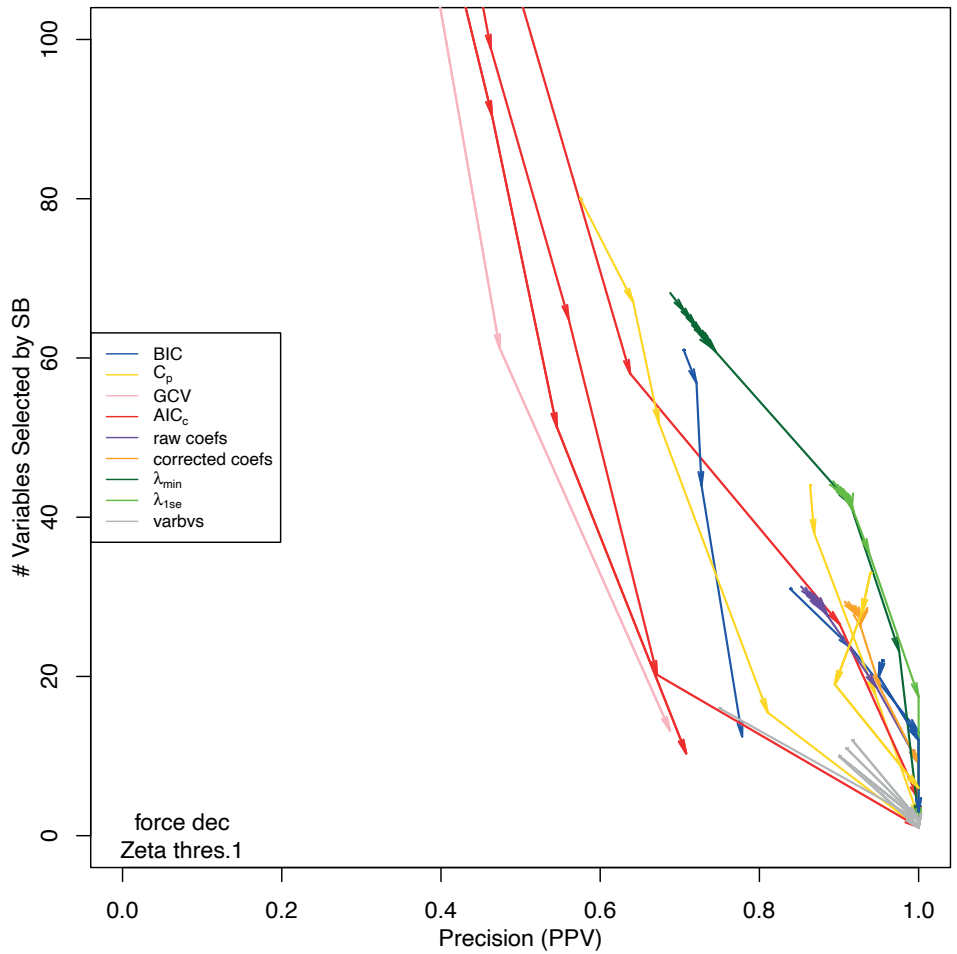


Fig. 4. Average number of identified variables as a function of the proportion of correctly identified variables for simulated data.

le cnam

Recall and precision

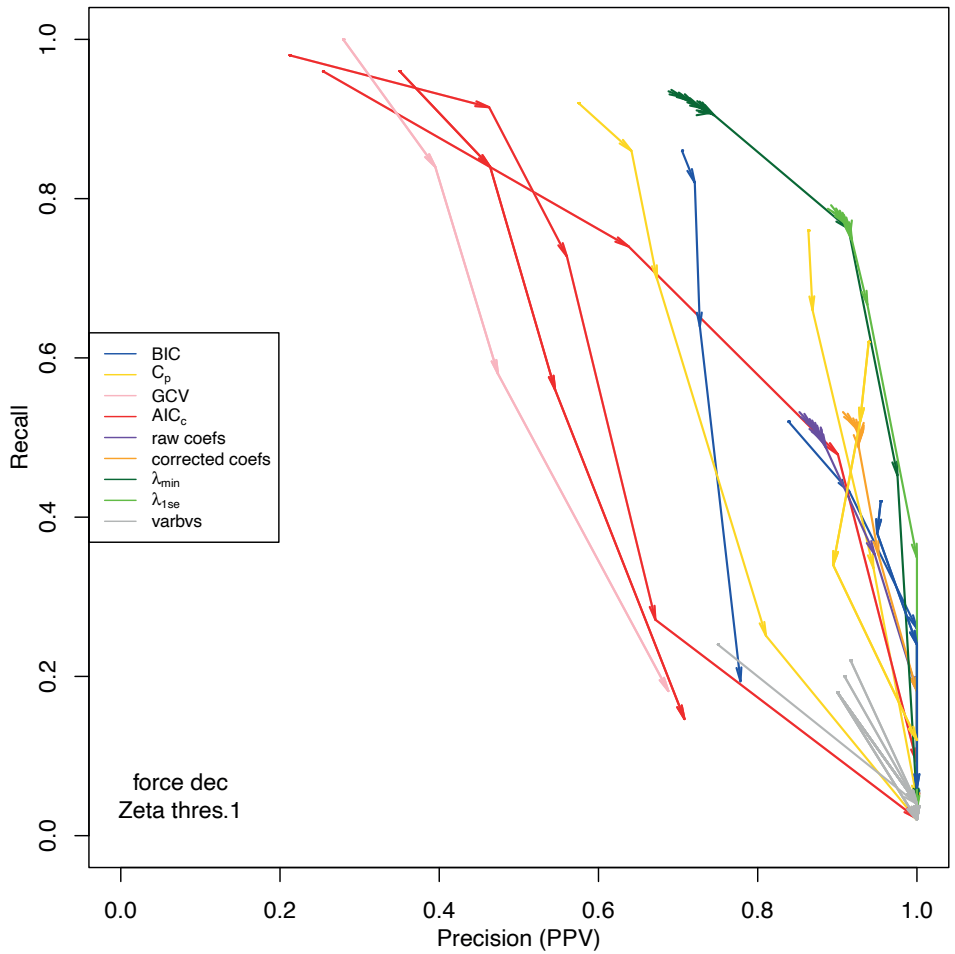


Fig. 5. Recall plotted as a function of the precision for simulated data.

SelectBoost

How robust is a given selection?

Study whether variables are selected or not in a model when the data are resampled while taking into account the correlation between variables.

$\rightarrow$ counts how many times a variable is selected for a given level of "agitation".

Details in Bertrand et al. Bioinformatics (2021), and package available on the CRAN at `https://cran.r-project.org/web/packages/SelectBoost/index.html`.

Linear model with Lasso estimator, or more generally any type of model as long as **the predictors are continuous quantitative variables**.

From GLMs to Beta regression

Goal. Extend SelectBoost—a stability selection scheme under correlated predictors—to responses in $(0, 1)$ via Beta regression.

Model. For $y_i \in (0, 1)$,

$$y_i \sim \mathrm{Beta}(\mu_i\phi,\ (1-\mu_i)\phi), \qquad g(\mu_i) = \eta_i = \mathbf{x}_i^\top \beta, \quad g = \text{logit},$$

with precision $\phi > 0$ (constant or updated).

Key tweaks.

- *Squeeze* y into $(0, 1)$: $\tilde{y} = \frac{y(n-1)+0.5}{n}$ to avoid boundary issues.
- Keep SelectBoost's correlated-resampling over a grid $c_0 \downarrow$; only the *base selector* changes.
- Selection frequencies per c_0 unchanged; interpret as in the original method.

le cnam

IC-based Beta selectors

Mean submodel uses `betareg` with logit link for μ.

Greedy add/drop (robust to `betareg`)

At each step, test adds/drops and select the move that minimizes

$$\mathrm{AIC} = -2\ell + 2k, \quad \mathrm{BIC} = -2\ell + k \log n, \quad \mathrm{AICc} = \mathrm{AIC} + \frac{2k(k+1)}{n-k-1},$$

stopping when improvement $< \varepsilon$.

- Handles p moderate; transparent model path.
- Output: named vector $(\beta_0, \boldsymbol{\beta})$ (zeros for nonselected).

IRLS + `glmnet` (pure path)

Working system (Beta IRLS for mean link):

$$z = \eta + \frac{y - \mu}{\mu(1-\mu)}, \qquad W = \phi\,\mu(1-\mu).$$

Fit $\eta \approx \beta_0 + \mathbf{X}\boldsymbol{\beta}$ by weighted least squares using `glmnet` on $(z, \mathbf{X})$ with weights W.

- Choose λ by BIC/AIC or CV; optional pre-standardization of $\mathbf{X}$ for speed.
- Update ϕ (precision) by a few Newton steps for stability.
- Scales to $p \gg n$; returns $(\beta_0, \boldsymbol{\beta})$.

SelectBoost workflow (Beta response)

Inputs: $\mathbf{X} \in \mathbb{R}^{n \times p}$, $y \in (0,1)$ (or intervals $[y_i^-, y_i^+]$), selector S.

1. Compute correlation matrix; build groups by threshold c_0.
2. Generate B correlated surrogates of $\mathbf{X}$ via groupwise resampling (with caching).
3. If interval data: draw $y^{(b)}$ by midpoint or uniform sampling in $[y_i^-, y_i^+]$.
4. Run S on each surrogate to get $\widehat{\beta}^{(b)}$; mark nonzeros.
5. Aggregate selection frequencies per variable; sweep c_0 grid from $1 \to 0$.

Outputs: frequency matrix (rows: c_0, cols: variables) with attributes selector, c0.seq, B, interval, diagnostics.

Practical guidance

- Clamp/squeeze y; logit link by default.
- Prefer BIC/AICc stepwise for sparse, interpretable models; use IRLS+`glmnet` for high-p.
- Interval responses encode measurement uncertainty yet fit seamlessly into stability selection.
- Parallelize across resamples; cache correlated draws per c_0 for speed.
- Thresholding of frequencies follows original SelectBoost practice; report the path over c_0.

Why GAMLSS + SelectBoost?

Setting. GAMLSS models the full distribution of Y via up to four parameters (μ, σ, ν, τ) with links g_θ and predictors $\eta_\theta = g_\theta(\theta)$, $\theta \in \mu, \sigma, \nu, \tau$.

Challenge. High-dimensional X, strong collinearity, and distinct parameter-specific design spaces make variable selection unstable and hard to reproduce.

Idea. *SelectBoost* wraps any selector with (i) correlation-aware grouping, (ii) correlated resampling, and (iii) stability tallies across a correlation threshold $c_0 \in [0, 1]$, delivering robust term discovery for each GAMLSS parameter.

Algorithmic Extension to GAMLSS

Inputs. Data $(y_i, \mathbf{x}_i)$; candidate bases per parameter (e.g. linear, interactions, splines $pb(\cdot)$). A selector per parameter.

1) Normalise & correlate. Standardise pooled design $X^\star$ (or per-parameter X_θ); compute $C = \mathrm{cor}(X^\star)$.

2) Grouping by c_0. Build correlation groups $G_j(c_0)$ from $|C|$.

3) Correlated resampling ($b = 1, \ldots, B$).

1. Subsample rows; within each group $G_j(c_0)$ sample a representative column;
2. For each $\theta \in \mu, \sigma, \nu, \tau$, run the chosen selector (*e.g.*, stepwise AIC/BIC, (group-)LASSO/Ridge/Elastic Net, grpreg/sgl);
3. Record selected terms for that θ.

Output. Stability proportions $\hat{\pi}_{\theta j}(c_0)$ for every term j in each parameter.

le cnam

Selectors & Engines

Mean (μ) and **dispersion/shape** (σ, ν, τ) **may use different engines.**

- **Stepwise** on IRLS fits (AIC/BIC); **betareg**-style for beta-type means.
- **Penalised GLM/GAMLSS** paths: Ridge, LASSO, Elastic Net; **group**-LASSO on spline blocks.
- **grpreg** / **sgl** families for grouped or sparse-group penalties on basis expansions.
- **Custom scopes** per parameter: e.g. $\eta_\mu \sim pb(x)$, $\eta_\sigma \sim pb(x) + z$; heredity constraints optional.

Tuning inside SelectBoost. Use B resamples and a grid of c_0; optionally cross-validate selector hyperparameters within each resample.

Confidence Functionals & Plots

Decision rule (per parameter). For chosen c_0 and threshold π_{thr}, keep $j : \hat{\pi}_{\theta j}(c_0) \geq \pi_{\text{thr}}$ and refit GAMLSS.

Confidence functionals (summarise stability over c_0):

- *Area above threshold*: $A_{\theta j} = \int \max \hat{\pi}_{\theta j}(c_0) - \pi_{\text{thr}}, 0, dc_0$;
- *Quantile index*: $Q_{\theta j} = \text{Quantile}_{\alpha} \hat{\pi}_{\theta j}(c_0)$;
- *Early-rise score*: slope of $\hat{\pi}_{\theta j}(c_0)$ near small c_0.

Plots. Stability curves per parameter; barplots of $A_{\theta j}/Q_{\theta j}$; stacked views comparing μ vs. $\sigma/\nu/\tau$.

Workflow & Practical Tips

Workflow.

1. Encode bases per parameter (splines, interactions, groups); standardise predictors.
2. Run SelectBoost-GAMLSS with a grid of c_0 and B resamples.
3. Choose π_{thr}; rank via functionals (A, Q, early-rise).
4. Refit final GAMLSS on selected terms; report curves, ranks and coefficients.

Tips.

- Use ($B \geq 50$) for stable tallies; widen c_0 grid if collinearity is strong.
- Group entire spline bases to respect smoothness; consider heredity across parameters.
- For responses at boundaries, apply squeezing or a zero/one-inflated family as appropriate.

Contents

Context

An observation

After a biological stimulation mimicking cancer

- in one group the cells proliferate at day +4,
- in the other the cells do not proliferate.

$\rightarrow$ There is therefore a “visible” difference.

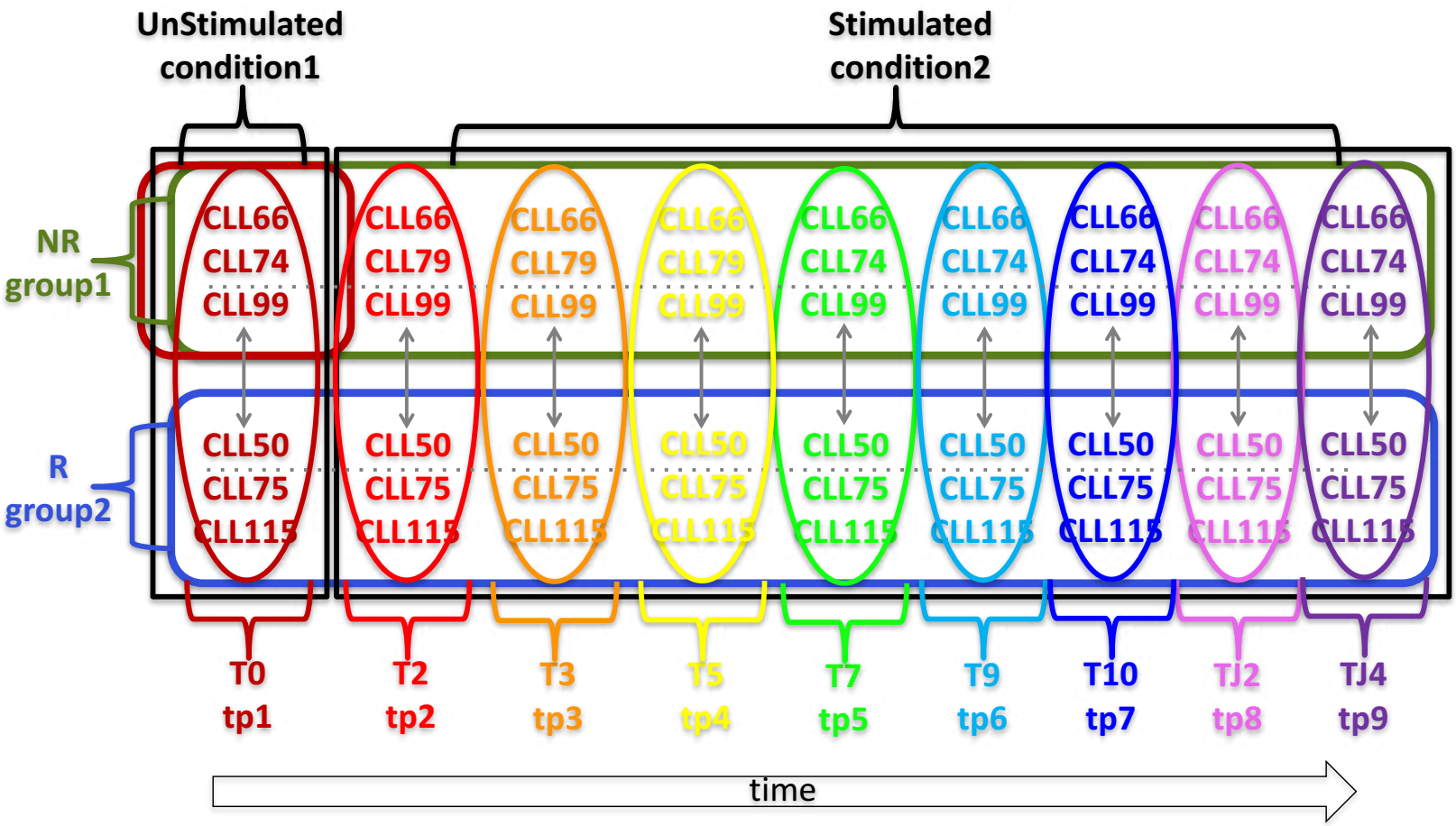
$\rightarrow$ How can we decode the signal (i.e., the sequence of gene and protein activations) following the stimulation?

$\rightarrow$ Reduce cell proliferation.

$\rightarrow$ Cure this type of leukemia (major therapeutic interest but a distant prospect).

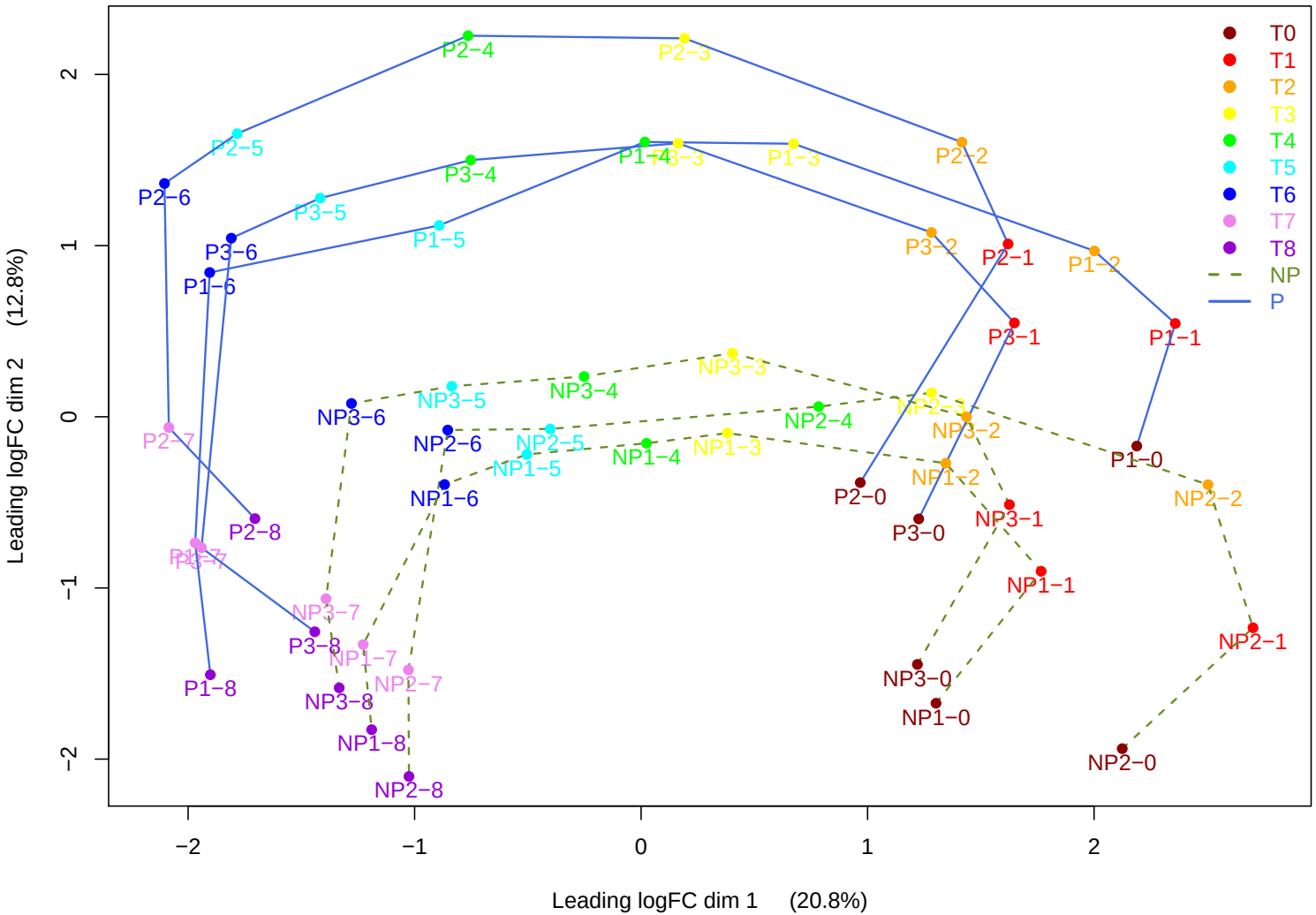
Context

Design of the experiment



Context

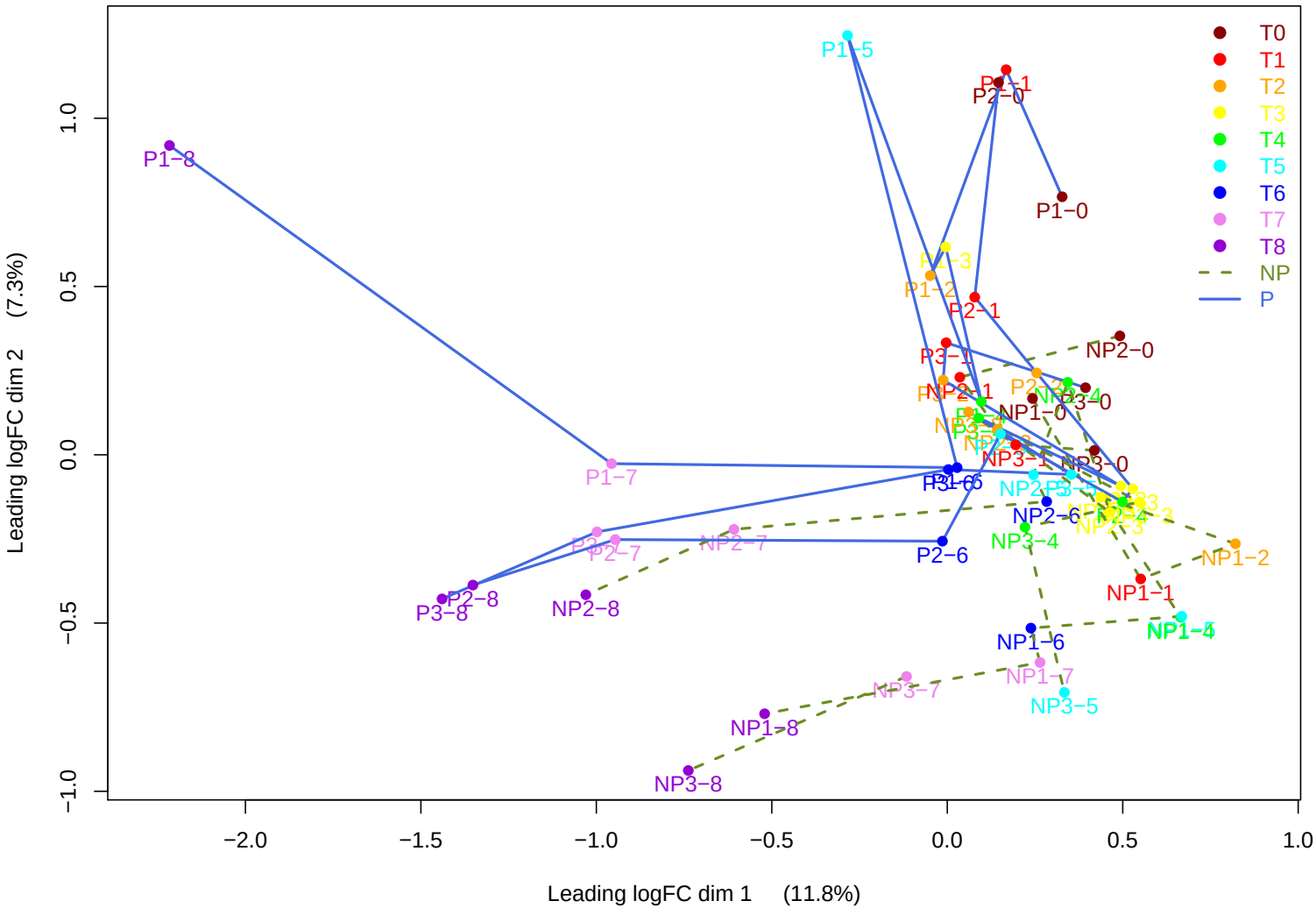
Genomic data



Multidimensional scaling plot of distances between profiles, `plotMDS`, `limma`.

Context

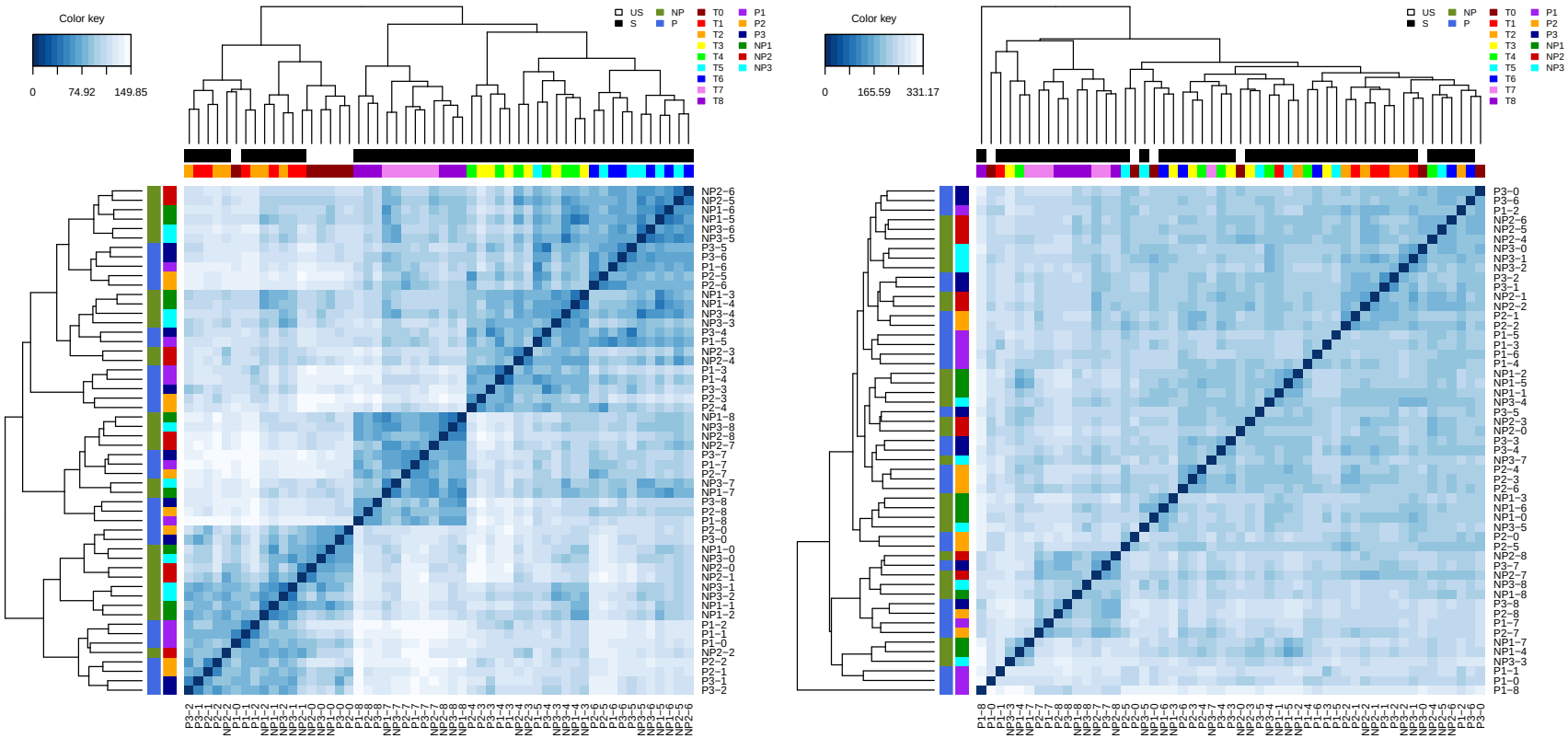
Proteomic data



Multidimensional scaling plot of distances between protein abundancies profiles, `plotMDS`, `limma`.

Context

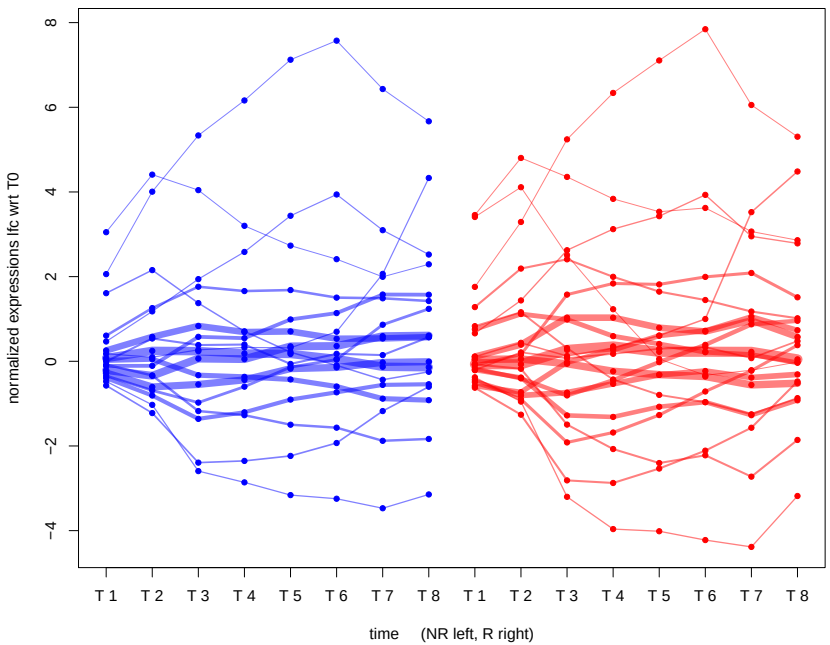
Genomic (g) and proteomic (d) data

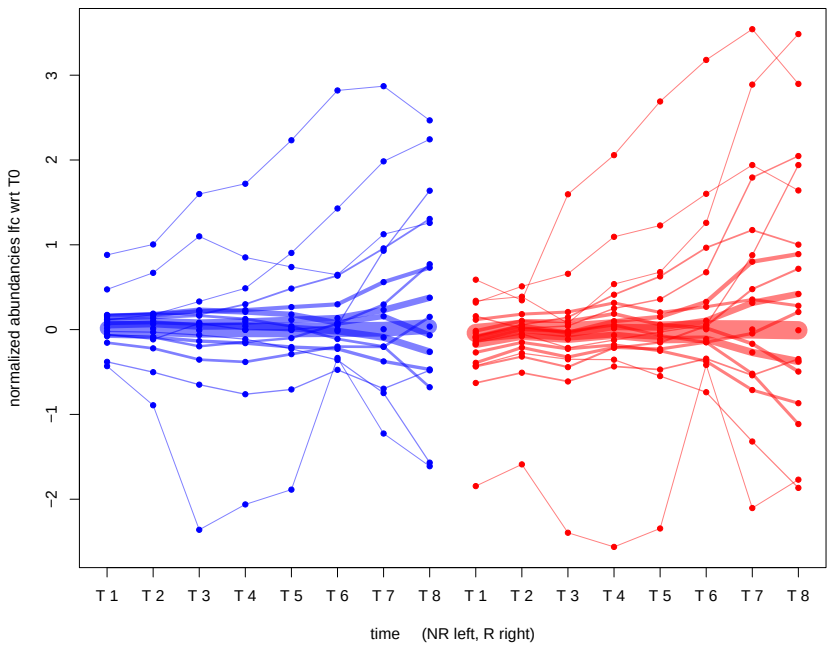


Clustered Image Maps (CIMs) ("Heat Maps"), `mixOmics`.

Context

Genomic (g) and proteomic (d) data



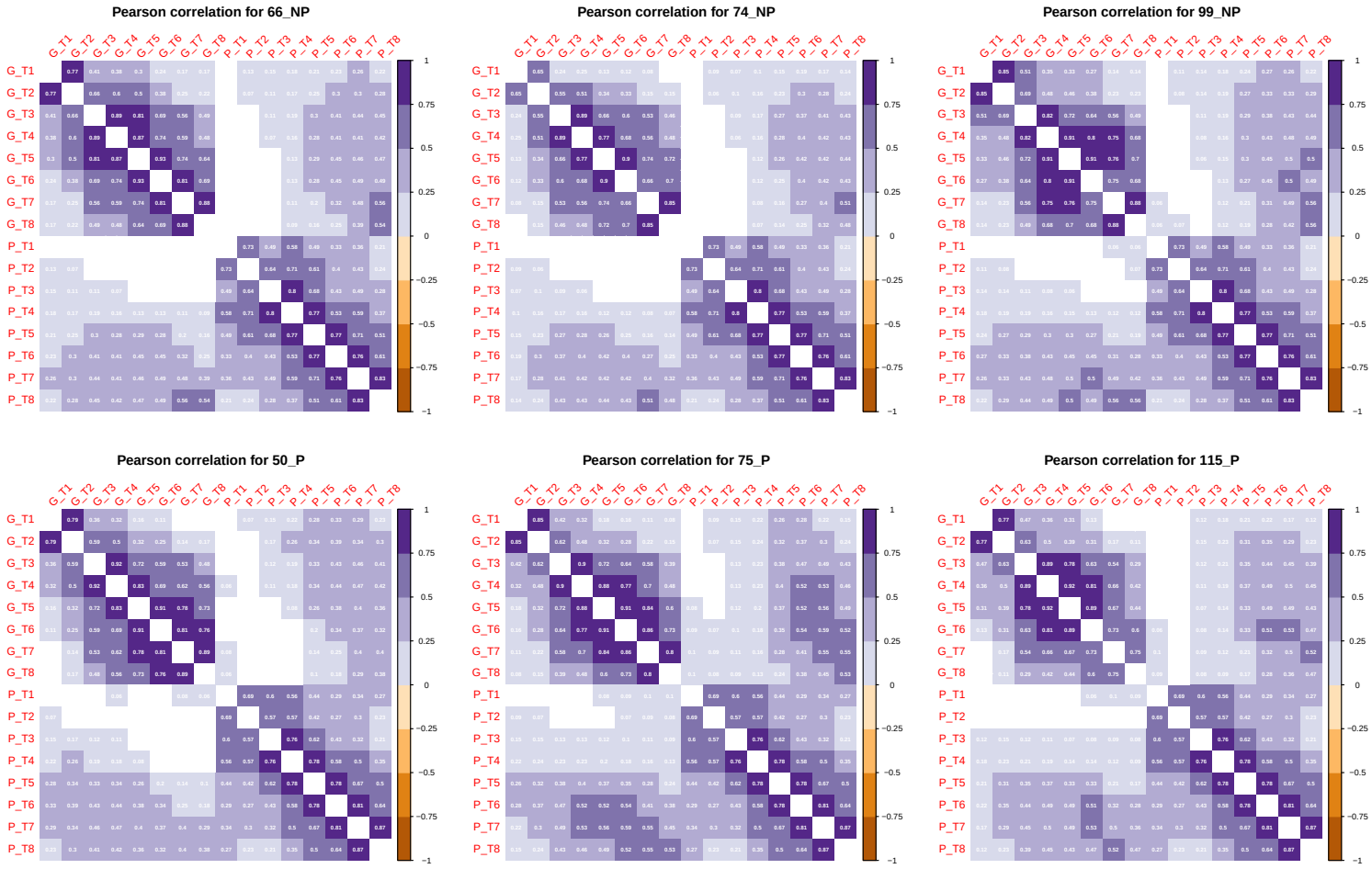


Groups of genes and proteins can be identified. `Mfuzz` (Futschik 2005; Kumar 2007).

le cnam

Context

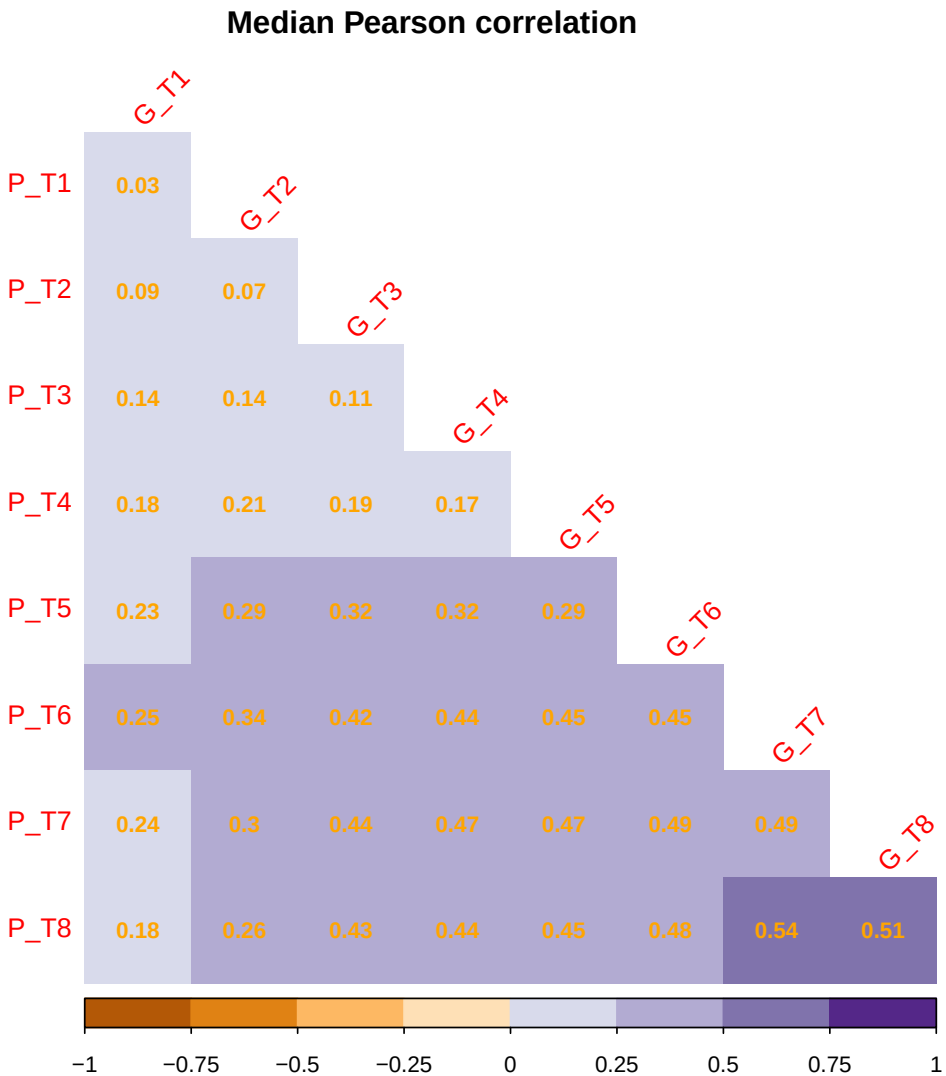
Genes and proteins correlations



Correlations for each subject.

Context

Genes and proteins correlations



Contents

Joint inference of biological networks

Objectives

It was impossible to use the methodological approaches available at the time and impossible to enlarge the experimental design.

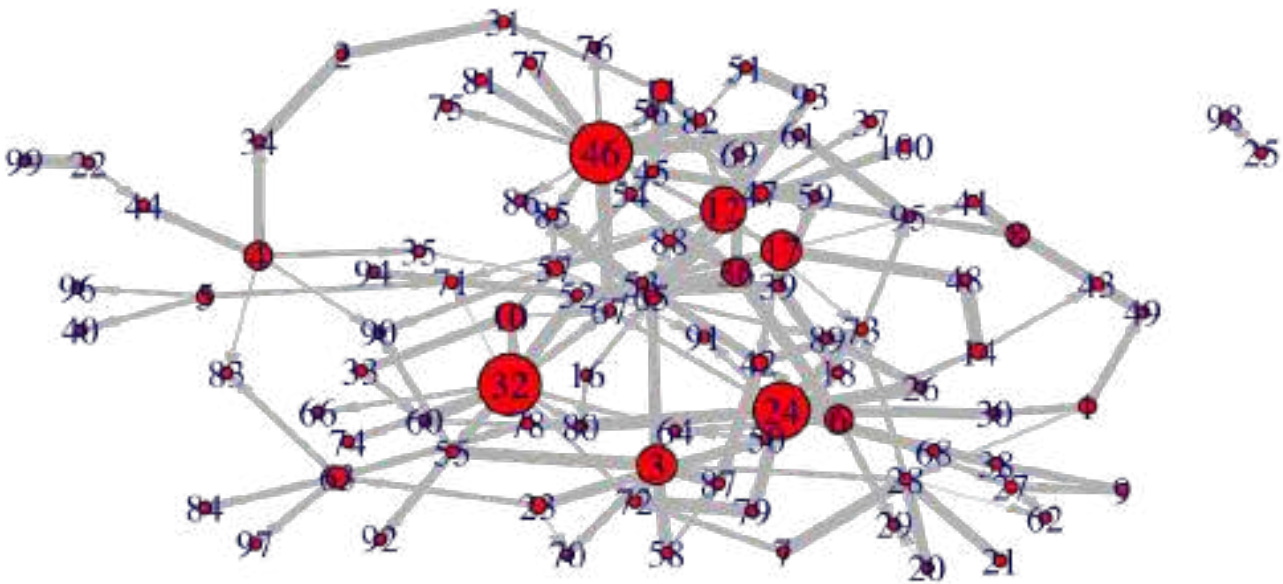
- experimental cost greater than 300k€
- cells (limited in quantity) sampled from real subjects (limited in number).

Twofold problem of **inferring** both

- the **propagation of the signal** in the network, and
- the **network** itself.

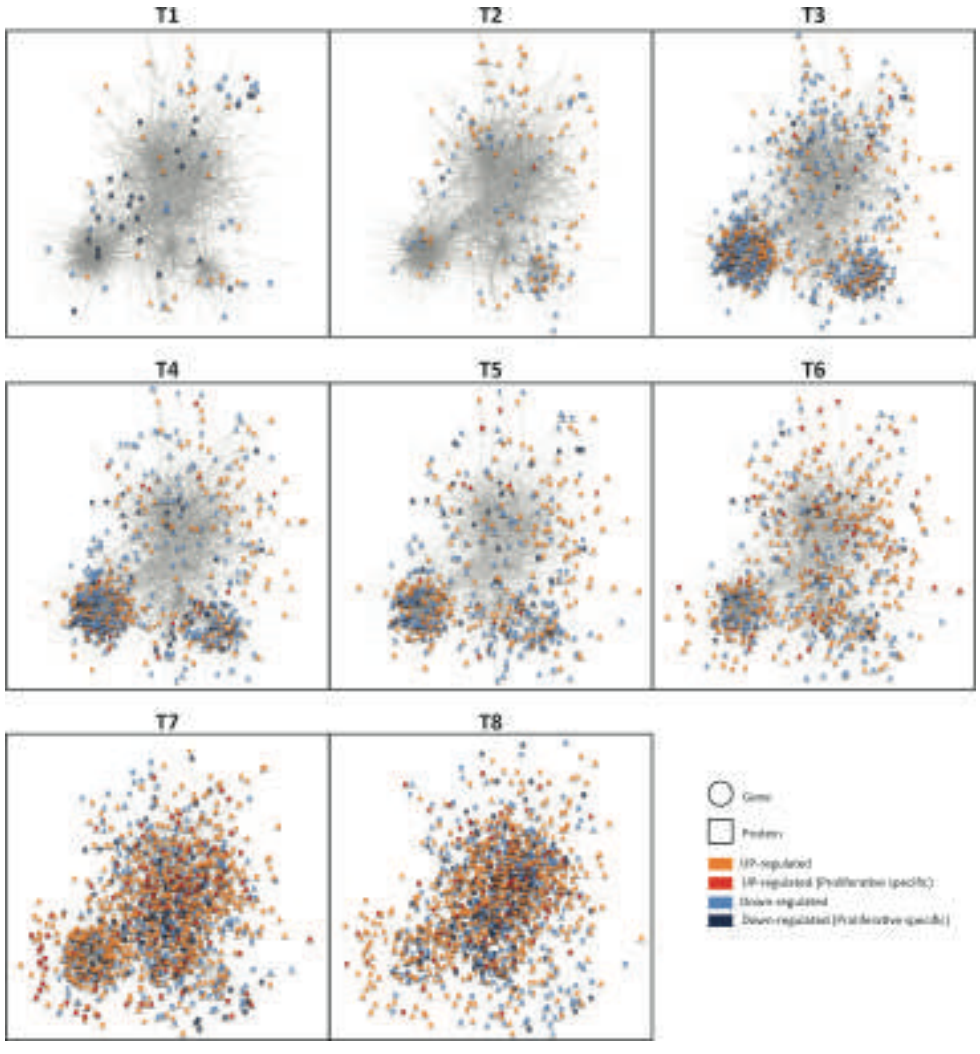
Joint inference of biological networks

Objectives: inferring the network



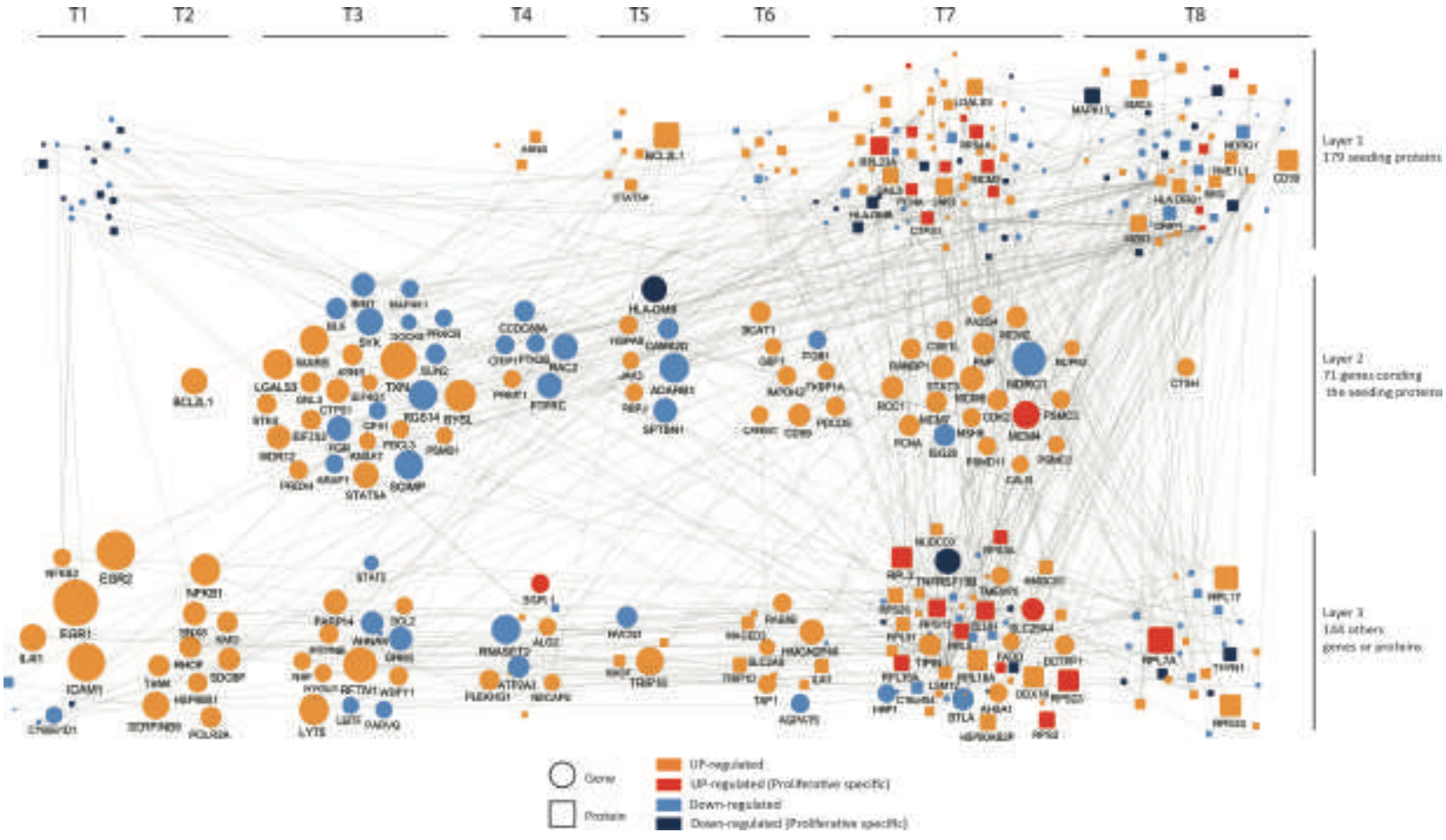
Joint inference of biological networks

Objectives: inferring the signal



Joint inference of biological networks

Objectives: isolating the proliferative sub-network



Contents

Joint inference of biological networks

Methodology

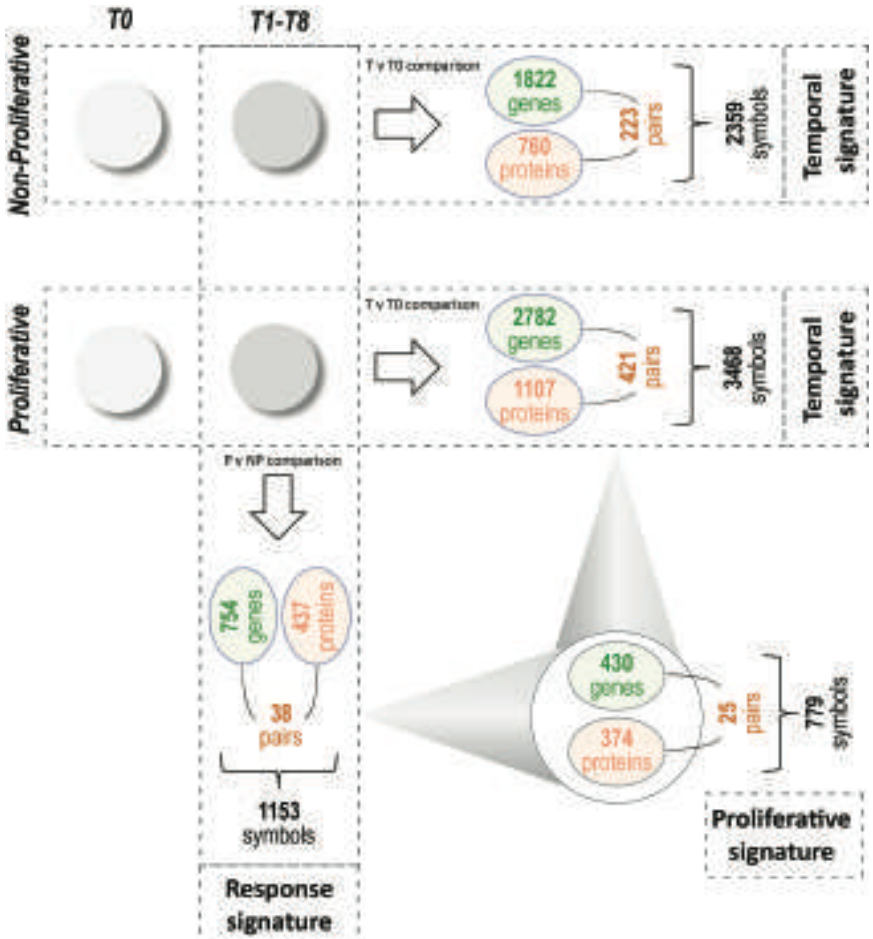
First individual-level modelling of the temporal evolution of expression levels and abundances of gene–protein pairs.

Several methodological bottlenecks.

1. **Choosing** the entities to be modelled (23,442 genes and 4,664 proteins).
2. **Reconciling** two types of entities measured in different ways.
3. Collecting information to **weight** the inference.
4. Retaining the the most **reliable** links.
5. Respecting the **cascade** structure induced by the stimulation and integrating two types of temporal **irregularities**.

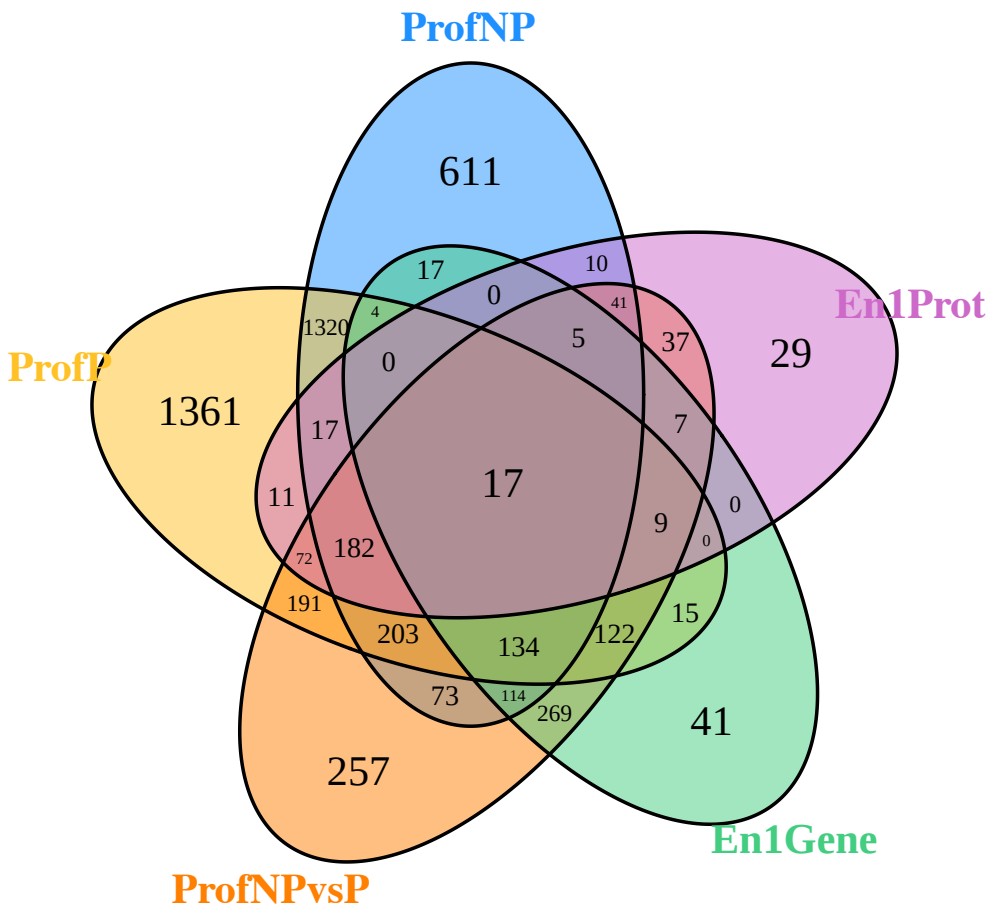
Modelling genomic and proteomic data

Selection of entities: differential fold changes



Modelling genomic and proteomic data

Selection of entities: enrichments



Modelling genomic and proteomic data

Reconciling the two types of entities

- **Matching** genes and protein groups (a priori non-bijective).
- RNA-seq (**counts**) and protein groups (**abundances** inferred from peptide intensities).
- **Transforming** the measurements so that they become **sufficiently** similar to be **integrated** in the **same model**.

Counts + `voom` $\rightarrow$ continuous quantitative values.

Adjust the ranges of the distributions using a multiplicative factor (50% trimmed mean of the ratios of ranges, when available).

Modelling genomic and proteomic data

Using biological information and weighting

Use of the **RegNetwork** database (341,207 links, Liu 2015).

Two types of **information** on each link:

- confidence (high, medium, low), and
- evidence (experimental observation, link predicted by a model).

Modelling genomic and proteomic data

Selecting the most reliable links, confidence scores on the links

Model fitting was carried out using a weighted version of

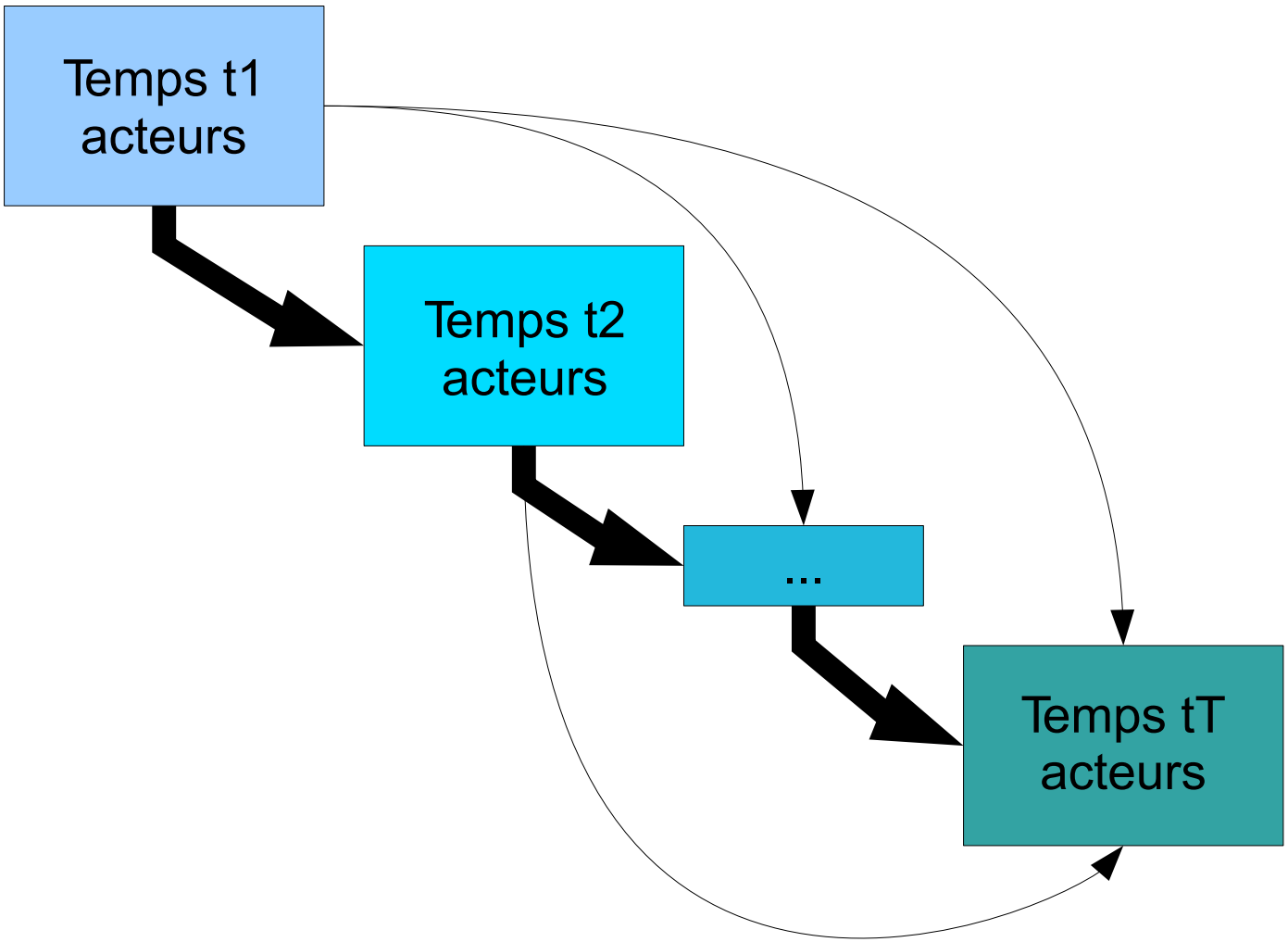
- the **stability selection** algorithm (Meinshausen, 2010), or
- the **selectboost** algorithm (Bertrand, 2021).

$\Rightarrow$ retain only the most reliable links.% in the matrix ω

selectboost $\Rightarrow$ computation of a confidence index in the presence of correlated variables.

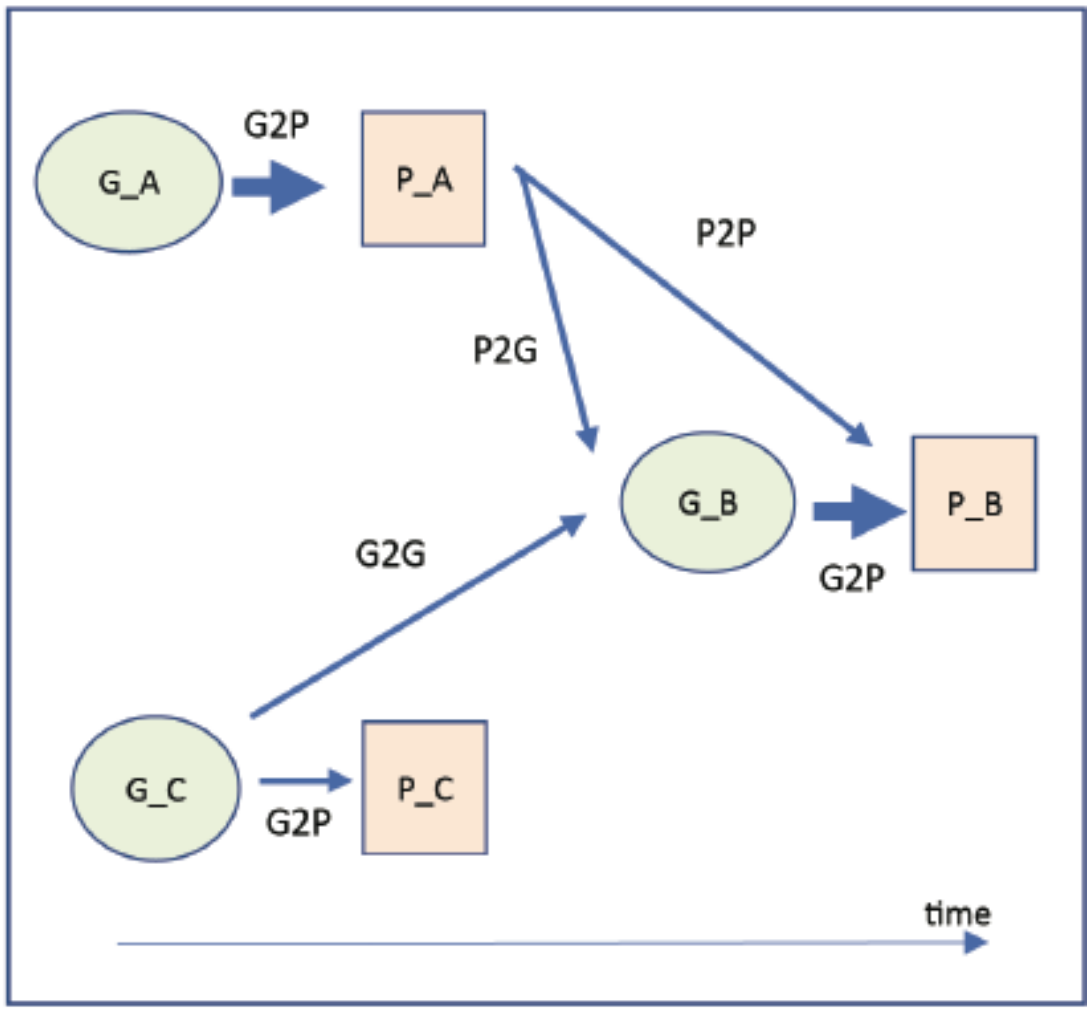
Joint inference of biological networks

Methodology: cascade structure



Context

Methodology: planned actions



Contents

Joint inference of biological networks

Mathematical model

Let:

- N be the number of **actors** (genes or proteins),
- P the number of **subjects**,
- T the number of measurement **times**,
- G **groups** of actors (partition, measurement similarities).

We denote:

- x_{npt} the observed value of actor n measured on subject p at time t,
- $\tilde{\boldsymbol{x}}_{np\cdot}$ the vector of length T formed by the observed values of actor n measured on subject p at the T time points.

Joint inference of biological networks

Mathematical model

Data of the biological problem:

- **6** subjects (2 groups of 3),
- **9** measurement times (t_i, $0 \leqslant i \leqslant 8$),
- **48** ($= 6 \times 8$) observations vs t_0,
- **7747** variables,

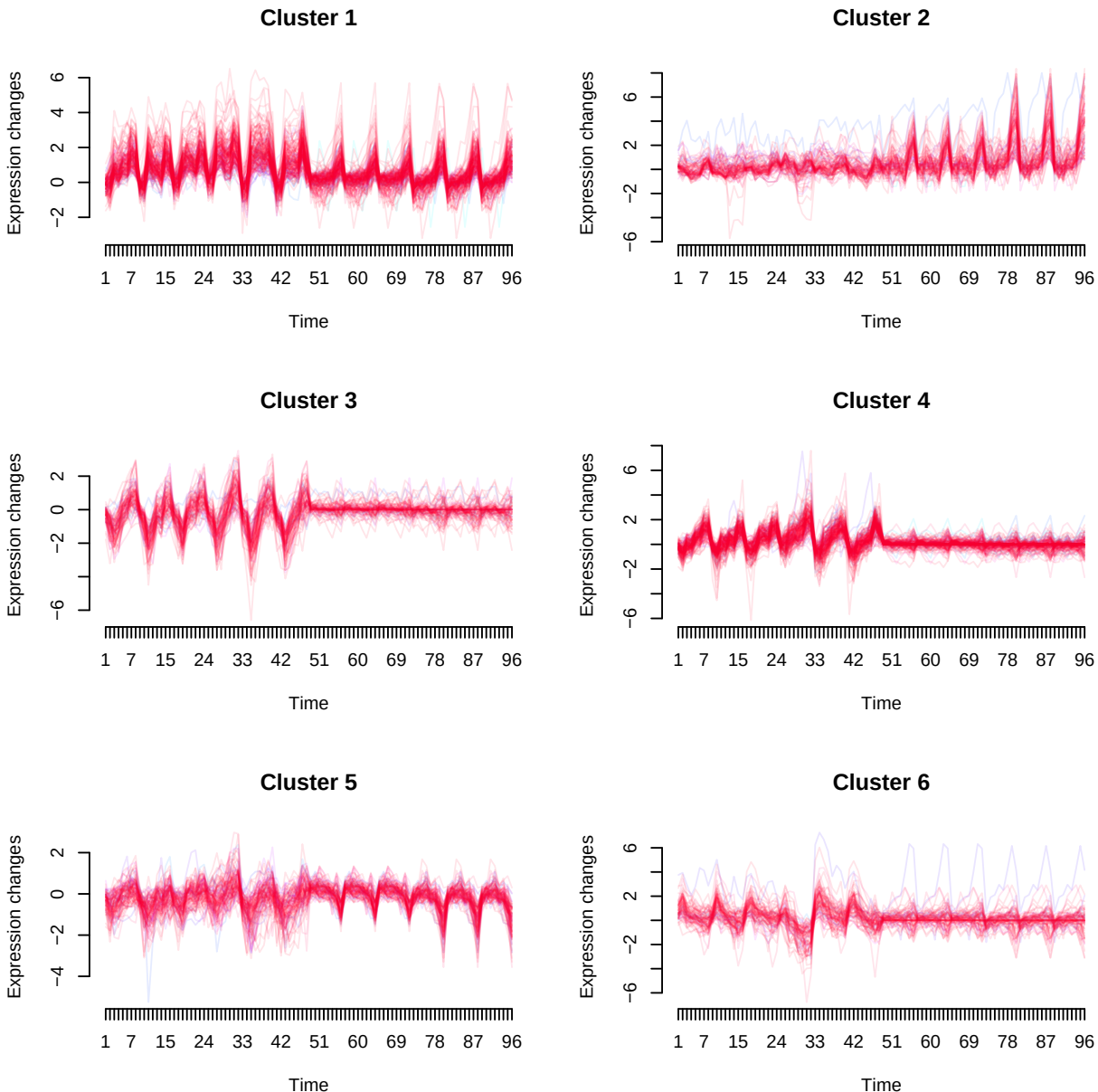
$\rightarrow$ network decoding.

For example, if $n = 1$ the actor is the **AAMP gene** and, for patient $p = 1$, its differential measurements at the 8 time points are:

$$\tilde{\boldsymbol{x}}_{11\cdot} = (-0.20, -0.08, 0.31, -0.02, 0.56, 0.34, 0.47, 0.65).$$

Joint inference of biological networks

Mathematical model: construction of the groups



Joint inference of biological networks

Mathematical model: construction of the groups

7747 variables = **5722 genes** + **2025 proteins**

- Step 1: cluster the 5722 genes $\rightarrow$ **20 groups**.
- Step 2: cluster the 2015 actors for which protein measurements are available (based on both types of measurements) $\rightarrow$ **21 groups** of genes and **21 groups** of proteins.

Total: **62 groups**.

For each fixed actor n, chosen among the N (=7747), the proposed model is written as:

$$\tilde{\boldsymbol{x}}_{np\cdot} = \sum_{n'=1}^{N} \omega_{n'n} \boldsymbol{F}_{m(n')m(n)} \tilde{\boldsymbol{x}}_{n'p\cdot} + \boldsymbol{\epsilon}_{np}, \qquad 1 \leqslant p \leqslant P.$$

1. $n \mapsto m(n)$ associates each actor with its group;
2. $\boldsymbol{F}_{ij}$ is a square matrix describing the **effect** of actors from group i on actors from group j;
3. the entry ω_{kl} of the matrix $\boldsymbol{\omega}$ is the strength of the **link** from actor k to actor l;
4. $\boldsymbol{\epsilon}_{np}$ is a centered random vector with T components and variance $\boldsymbol{I_T}$.

A versatile modelling tool.

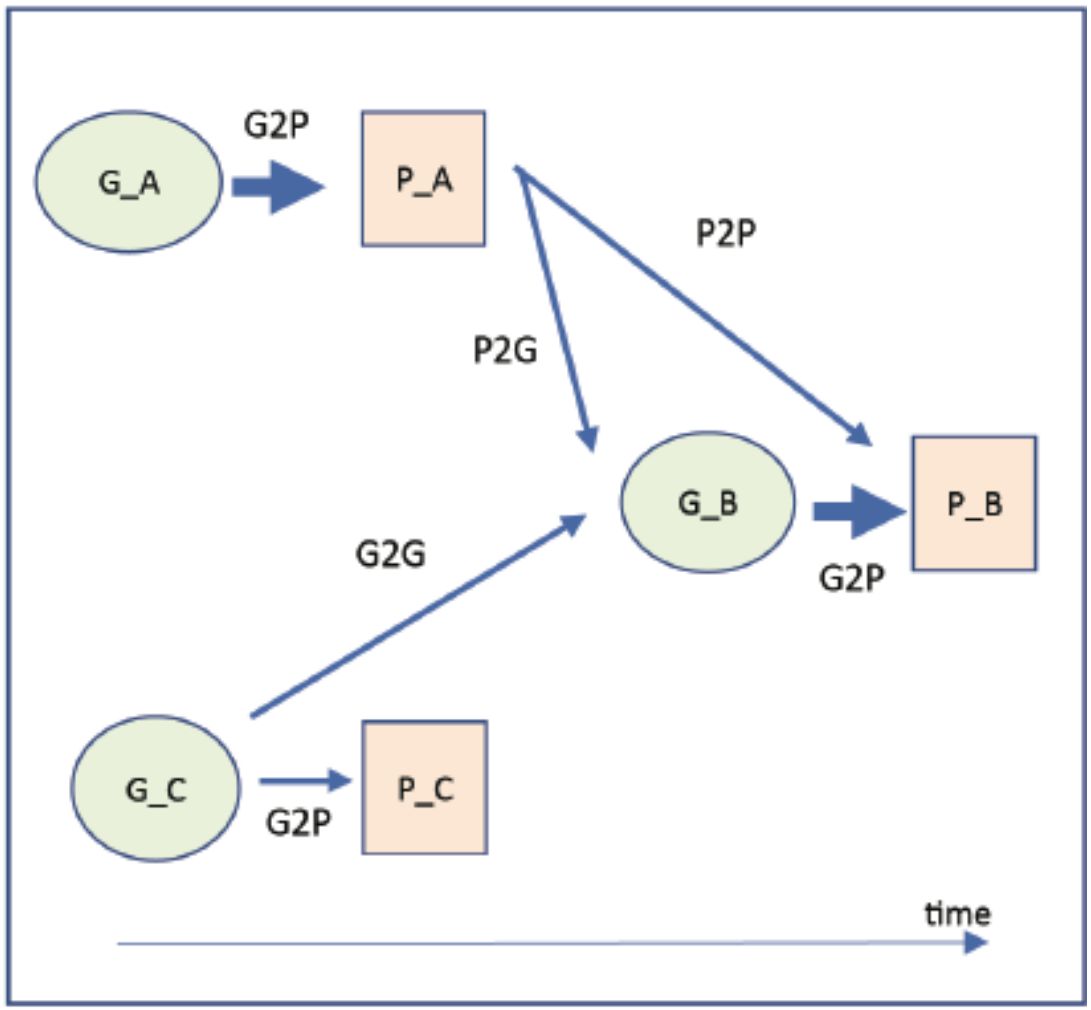
Many inference functions can be used to get specific features for the inferred network such as

- sparsity,
- robust links,
- high confidence links or
- stable through resampling links.

using various inference tools such as **lasso**, **spls**, **elasticnet**, **stability selection**, **robust lasso**, **selectboost** or any of their **weighted** counterparts.

Joint inference of biological networks

Mathematical model: actions



Joint inference of biological networks

Mathematical model: translating the actions

	1	2	3	4	5	6	7	8
1	p_{ij}^F	0	0	0	0	0	0	0
2	q_{ij}^F	r_{ij}^F	0	0	0	0	0	0
3	0	0	r_{ij}^F	0	0	0	0	0
4	0	0	0	r_{ij}^F	0	0	0	0
5	0	0	0	0	r_{ij}^F	0	0	0
6	0	0	0	0	0	r_{ij}^F	0	0
7	0	0	0	0	0	0	r_{ij}^F	0
8	0	0	0	0	0	0	0	r_{ij}^F

F matrix for a G2P action

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	a_{ij}^F	0	0	0	0	0	0	0
3	b_{ij}^F	a_{ij}^F	0	0	0	0	0	0
4	c_{ij}^F	b_{ij}^F	a_{ij}^F	0	0	0	0	0
5	d_{ij}^F	c_{ij}^F	b_{ij}^F	a_{ij}^F	0	0	0	0
6	0	e_{ij}^F	f_{ij}^F	i_{ij}^F	k_{ij}^F	0	0	0
7	0	0	0	j_{ij}^F	l_{ij}^F	m_{ij}^F	0	0
8	0	0	0	0	0	n_{ij}^F	o_{ij}^F	0

F matrix for a G2G, P2G, P2P action

Joint inference of biological networks

Use of biological information and weighting

Use of the RegNetwork database (341,207 links, Liu 2015).
Two types of information on each link:

- confidence (high, medium, low), and
- evidence (experimental observation, link predicted by a model).

$\rightarrow$ In agreement with the biologist collaborators, quantification of the value of a weight, denoted $\pi_{n'n}$, lying between:

- 0, systematic **presence** of the link,
- 1, **neutral** value, and
- $+\infty$, systematic **forbidden** link.

Joint inference of biological networks

Use of biological information and weighting

Integration of other biological constraints:

- Whenever possible, use protein **abundance** for the action of X and not the **expression** of X (biological dogma).
- No action within the same group (reasonable restriction: same group $\rightarrow$ same temporal profiles $\rightarrow$ activation at the same moment in the cascade).
- No **feedback** (from an actor on itself, although an action of a gene on its protein is still allowed).

Weighting is used when fitting the model, typically via the *lasso*.

Joint inference of biological networks

Cascade structure

The core of the statistical model combines

- the matrix $\boldsymbol{\omega} \in \mathcal{M}_N(\mathbb{R}) \quad (N = 7747)$
  $\rightarrow$ intensity and sign of the link between two specific actors.
  There are 3844 matrices F_{ij} to be estimated, i.e. 246,016 parameters, to which are added those of the matrix ω, for a total of 253,763 parameters. As we shall see later, I used the cascade structure of the signal to drastically reduce this number of parameters.
  $\rightarrow$ reconstruction of the network.
- the matrix $\boldsymbol{F} \in \mathcal{M}_{TG}(\mathbb{R}_+) \quad (T \times G = 8 \times 62 = 496)$
  $\rightarrow$ actions between the groups over time,
  $\rightarrow$ describes the propagation of the signal through the network.

Cascade structure of the signal to drastically reduce the potential number of parameters.

Joint inference of biological networks

Form of the cell matrices $\boldsymbol{F}_{ij} \in \mathcal{M}_T(\mathbb{R}_+)$ $(T = 8)$

Determination of the first activation time C_i of the actors in cluster i.

If $C_i \geqslant C_j$, then the matrix $\boldsymbol{F}_{ij} = 0$ (arrow of time).

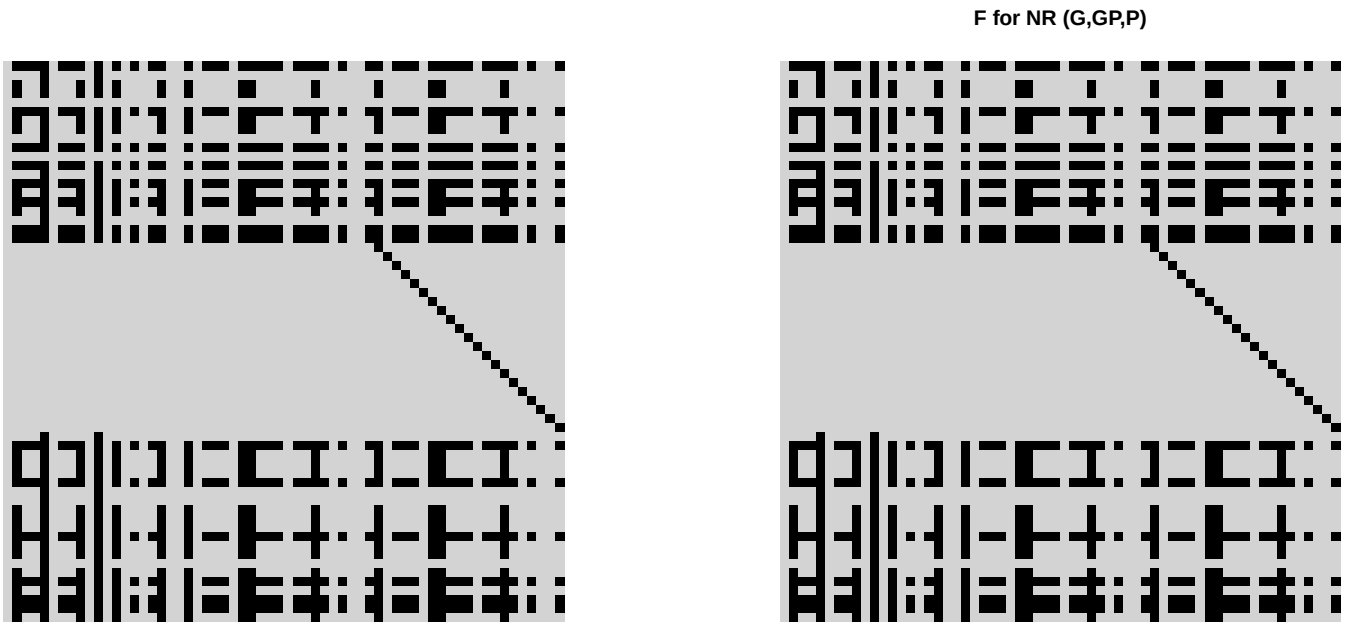


Figure: Example of a matrix $\boldsymbol{F}$; in black, the non-zero cells

Joint inference of biological networks

Form of the cell matrices $\boldsymbol{F}_{ij} \in \mathcal{M}_T(\mathbb{R}_+)$ $(T = 8)$

If $C_i < C_j$, then the square matrix $\boldsymbol{F}_{ij}$ of order $T = 8$ is

- lower triangular (arrow of time),
- strictly lower triangular except in the case of an action at the same time, for example that of a gene on its protein.

	1	2	3	4	5	6	7	8
1	p_{ij}^F	0	0	0	0	0	0	0
2	q_{ij}^F	r_{ij}^F	0	0	0	0	0	0
3	0	0	r_{ij}^F	0	0	0	0	0
4	0	0	0	r_{ij}^F	0	0	0	0
5	0	0	0	0	r_{ij}^F	0	0	0
6	0	0	0	0	0	r_{ij}^F	0	0
7	0	0	0	0	0	0	r_{ij}^F	0
8	0	0	0	0	0	0	0	r_{ij}^F

F matrix for a G2P action

	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0
2	a_{ij}^F	0	0	0	0	0	0	0
3	b_{ij}^F	a_{ij}^F	0	0	0	0	0	0
4	c_{ij}^F	b_{ij}^F	a_{ij}^F	0	0	0	0	0
5	d_{ij}^F	c_{ij}^F	b_{ij}^F	a_{ij}^F	0	0	0	0
6	0	e_{ij}^F	f_{ij}^F	i_{ij}^F	k_{ij}^F	0	0	0
7	0	0	0	j_{ij}^F	l_{ij}^F	m_{ij}^F	0	0
8	0	0	0	0	0	n_{ij}^F	o_{ij}^F	0

F matrix for a G2G, P2G, P2P action

$\rightarrow$ the measurement of an actor at time t_k influences the measurement of another actor at time t_{k_0} if and only if $k \leqslant k_0$.

Modelling genomic and proteomic data

Form of the cell matrices $\boldsymbol{F}_{ij} \in \mathcal{M}_T(\mathbb{R}_+)$ $(T = 8)$

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
\left[\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_{ij}^F & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
b_{ij}^F & a_{ij}^F & 0 & 0 & 0 & 0 & 0 & 0 \\
c_{ij}^F & b_{ij}^F & a_{ij}^F & 0 & 0 & 0 & 0 & 0 \\
d_{ij}^F & c_{ij}^F & b_{ij}^F & a_{ij}^F & 0 & 0 & 0 & 0 \\
0 & e_{ij}^F & f_{ij}^F & i_{ij}^F & k_{ij}^F & 0 & 0 & 0 \\
0 & 0 & 0 & j_{ij}^F & l_{ij}^F & m_{ij}^F & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & n_{ij}^F & o_{ij}^F & 0
\end{array}\right]
\end{array}
$$

Figure: Example of a cell matrix $\boldsymbol{F}_{ij}$

le cnam

Joint inference of biological networks

Selecting the most reliable links, confidence indices on the links

Model fitting was carried out using a weighted version of

- the **stability selection** algorithm (Meinshausen, 2010), or
- the **selectboost** algorithm (Bertrand, 2021).

$\rightarrow$ retain only the most reliable links in the matrix $\boldsymbol{\omega}$.

combined with *non-negative least squares* to estimate the coefficients of $\boldsymbol{F}$.

selectboost $\rightarrow$ computation of a confidence index in the presence of correlated variables.

Joint inference of biological networks

Interpretation of the inference results

The result obtained can be interpreted in three ways:

1. a **connectivity network** with the non-zero entries of $\hat{\omega}$: $\hat{\omega}_{n'n} \neq 0$ means that an action of n' on n has been detected;
2. each matrix $\hat{\boldsymbol{F}}_{ij}$ models the **actions between groups** and the times at which they occur;
3. the evolution over time of the action of an actor n on an actor n', which corresponds to the **propagation of the signal** through the network. It is obtained by computing the product $\hat{\omega}_{n'n}\hat{\boldsymbol{F}}_{m(n')m(n)}$.

Contents

Validations and perspectives

Performance tests

Impact of a good or poor specification of the **weighting** of the links between the actors in the network.

1. **Simulation** of networks inspired by preferential attachment (Albert, 2002; Barabasi, 2004),
2. adapted to nested temporal networks (**cascades**),
3. initial conditions of the actors in the network defined using Laplace distributions (classical in this context).

Validations and perspectives

Theoretical F and estimation of F by weighted lasso

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.20	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	0.20	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00	0.20	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.06	0.00	0.00	1.00	0.20	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.30	0.06	0.00	0.00	1.00	0.20	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.30	0.00	0.00	0.00	0.58	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.05	1.00	0.00	0.00	0.68	0.30	0.00	0.00	0.45	0.58	0.00	0.00
0.00	0.00	0.00	0.00	0.05	0.05	1.00	0.00	0.06	0.68	0.30	0.00	0.74	0.45	0.58	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.53	0.00	0.00	0.00	0.15	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.53	0.00	0.00	0.48	0.15	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.08	0.11	0.53	0.00	0.22	0.48	0.15	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.47	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.47	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.06	0.07	0.47	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure: F-score of the network decoding methods.

Validations and perspectives

Definitions: sensitivity, PPV and *F*-score

1. **Sensitivity** (= recall): probability that the link is detected if the link exists

$$\text{Sensitivity} = \frac{TP}{TP + FN}.$$

2. **Positive predictive value** (= precision): probability that the link is present when it is detected

$$\text{PPV} = \frac{TP}{TP + FP}.$$

3. ***F*-score**: popular measure that combines PPV and sensitivity by computing their harmonic mean

$$F = 2 \cdot \frac{(\text{PPV} \cdot \text{Sensitivity})}{(\text{PPV} + \text{Sensitivity})}.$$

Validations and perspectives

Sensitivity

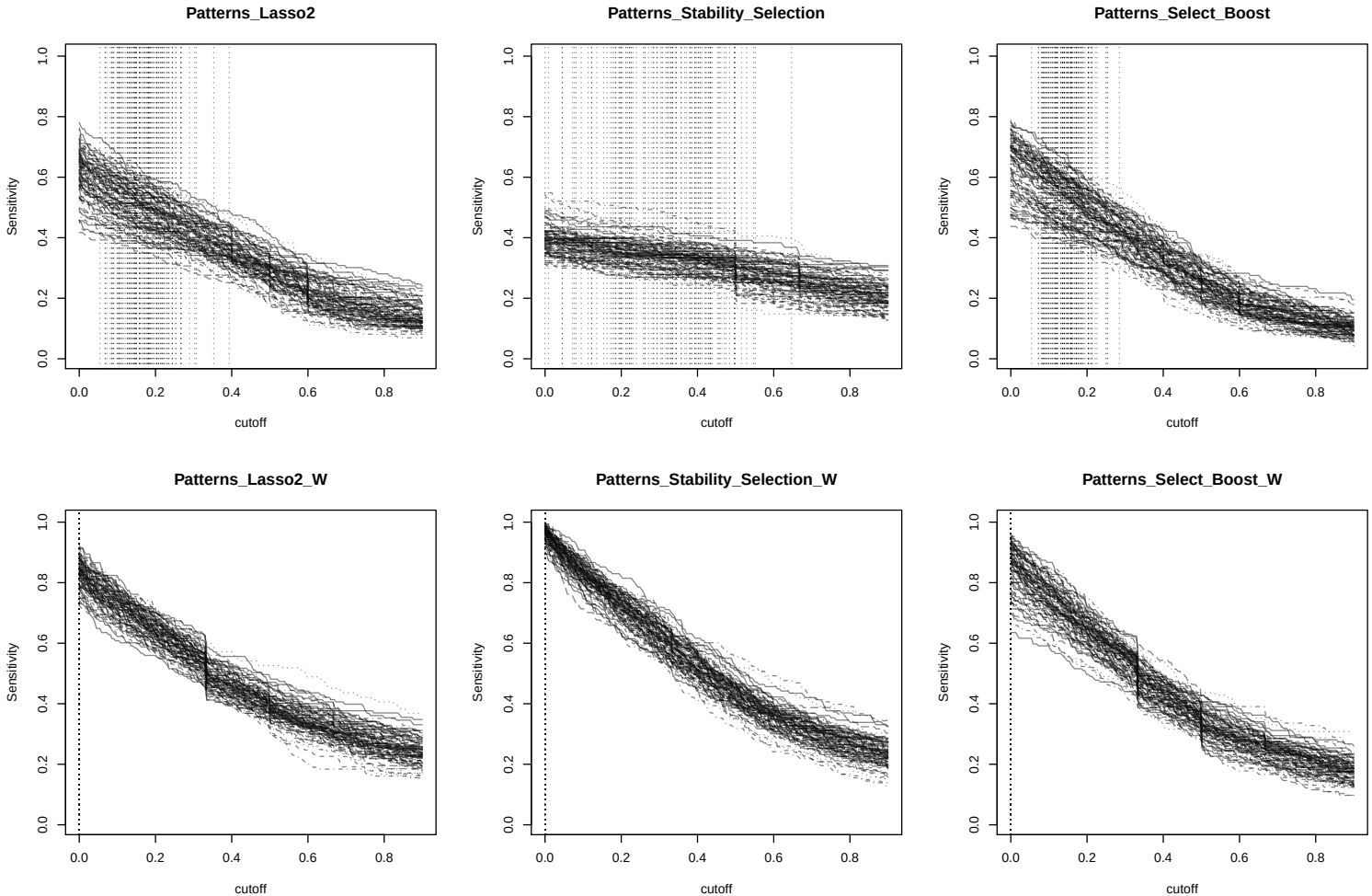


Figure: Sensitivity of the network decoding methods.

le cnam

Validations and perspectives

Positive predictive value

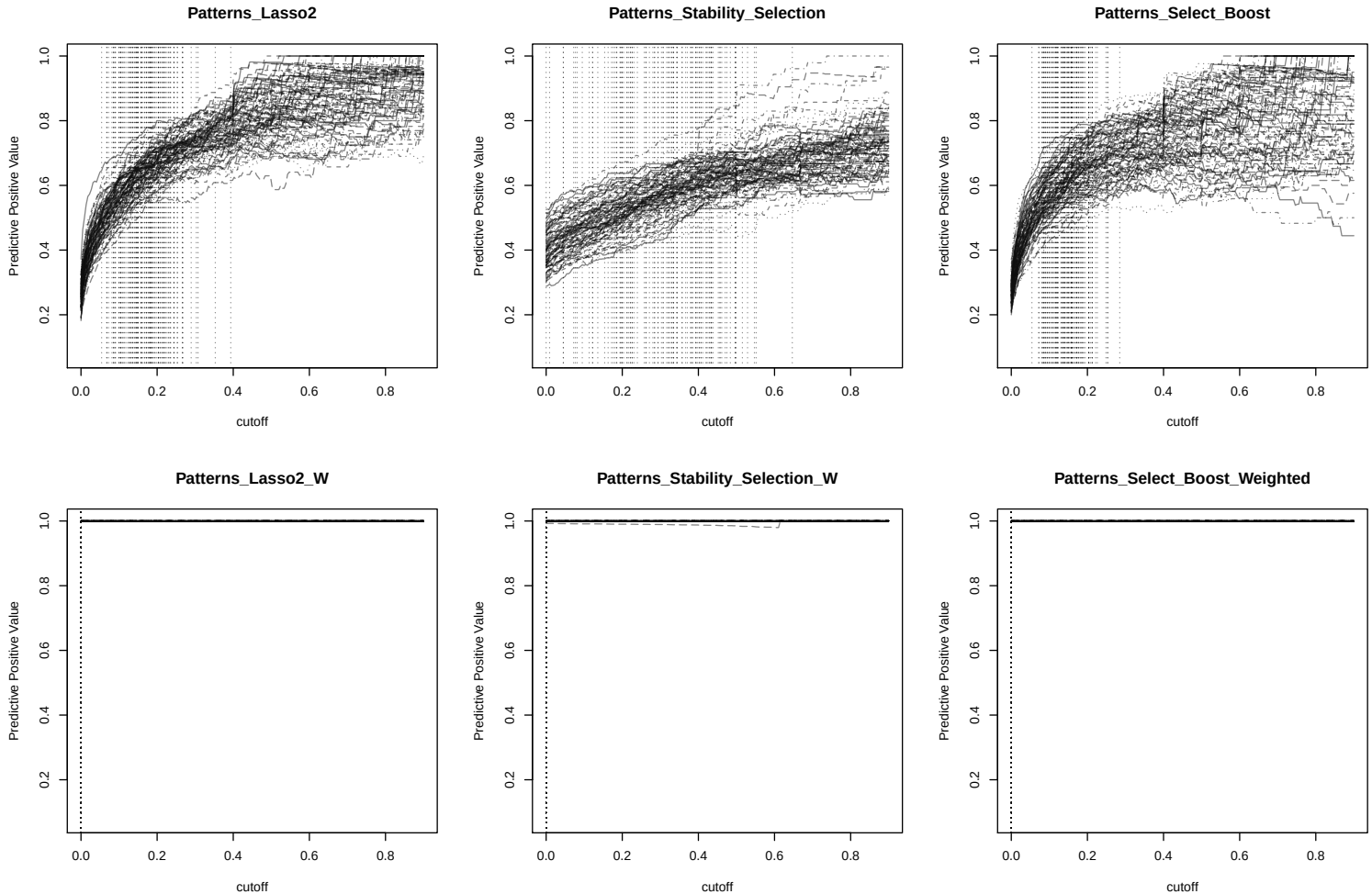


Figure: PPV of the network decoding methods.

Validations and perspectives

F-score

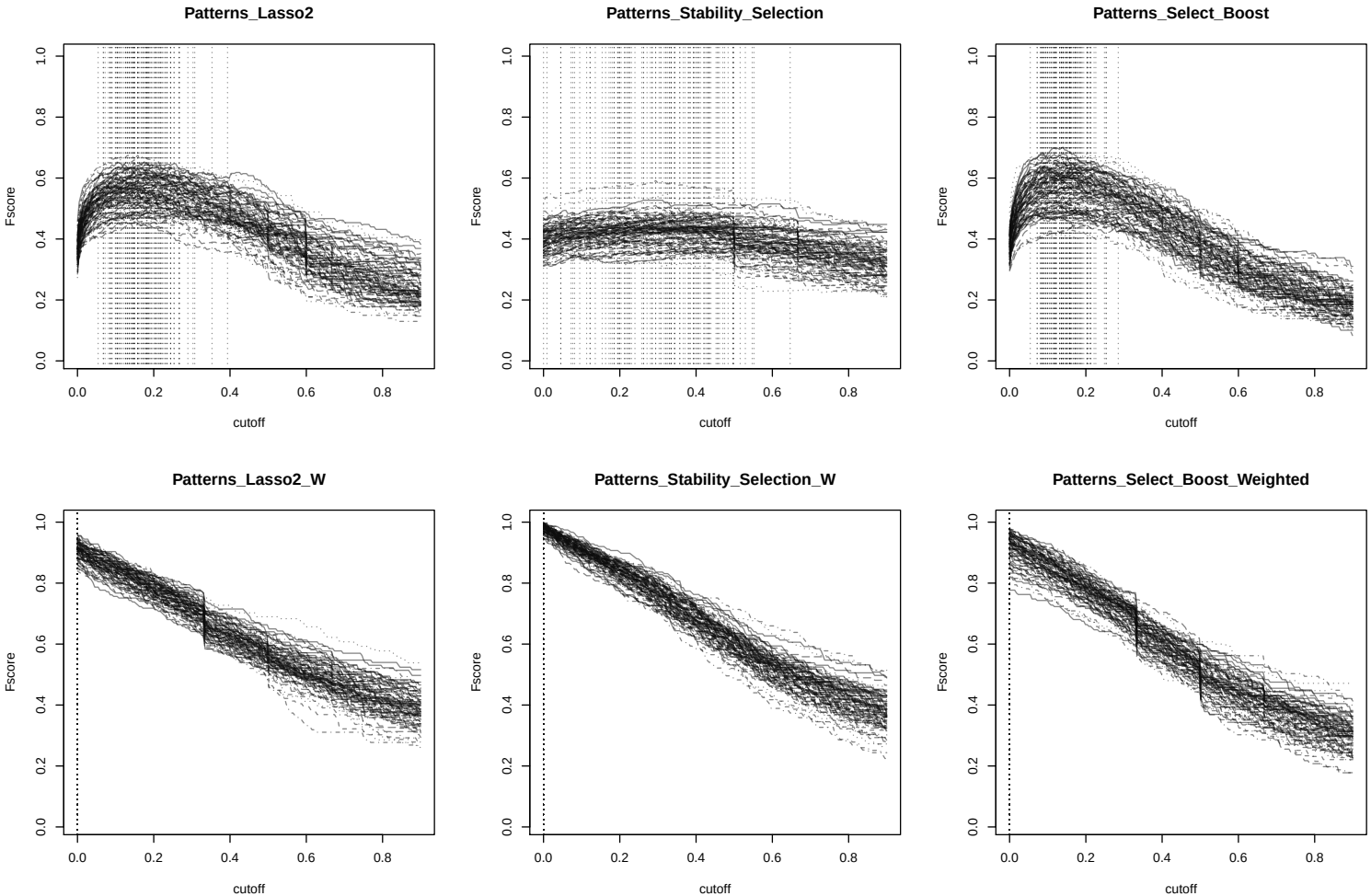


Figure: *F*-score of the network decoding methods.

le cnam

References

Akaike, H. (1974) A new look at the statistical model identification, *IEEE Transactions on Automatic Control*, **19**(6), 716-723.

Bertrand, F., Aouadi, I., Jung, N., Carapito, R., Vallat, L., Bahram, S. and Maumy-Bertrand, M. (2021) selectBoost: a general algorithm to enhance the performance of variable selection methods, *Bioinformatics*, **37**(5), 659-668.

Fan, J. and Lv, J. (2010) A selective overview of variable selection in high dimensional feature space, *Statistica Sinica*, **20**(1), 101.

Sra, S. (2012) A short note on parameter approximation for von Mises-Fisher distributions: and a fast implementation of Is(x), *Computational Statistics*, **27**(1), 177-190.

Schwarz, G. (1978) Estimating the dimension of a model, *The Annals of Statistics*, **6**(2), 461-464.

References

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

Bertrand, F., Aouadi, I., Jung, N., Carapito, R., Vallat, L., Bahram, S. and Maumy-Bertrand, M. (2020), SelectBoost : a general algorithm to enhance the performance of variable selection methods, *Bioinformatics*, btaa855.

Chun, H. and Keles, S. (2010) Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *JRSSB*, **72**(1), 3–25.

Liu, Z.P., Wu, C., Miao, H. and Wu, H. (2015). RegNetwork: an integrated database of transcriptional and posttranscriptional regulatory networks in human and mouse. *Database*, **2015**.

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

Schleiss, C., ..., Maumy-Bertrand, M., Bahram, S., Bertrand, F. and Vallat, L., (2021), Temporal multiomic modelling reveals a B-cell receptor proliferative program in chronic lymphocytic leukemia, *Leukemia*, **35**(5):1463-1474.

Thank you for watching this talk.

Any question?

Visit

https://cran.r-project.org/web/packages/Cascade
https://cran.r-project.org/web/packages/Patterns
https://cran.r-project.org/web/packages/SelectBoost
https://cran.r-project.org/web/packages/SelectBoost.beta
https://cran.r-project.org/web/packages/SelectBoost.gamlss

Contact

frederic.bertrand@lecnam.net