Feature selection from gene expression data
Molecular signatures for breast cancer prognosis and inference of gene regulatory networks.

Anne-Claire Haury

Centre for Computational Biology
Mines ParisTech, INSERM U900, Institut Curie

PhD Defense, December 14, 2012
Introduction
=> RNA as a **proxy** to measure gene expression.

**Figure:** Central Dogma of Molecular Biology (Source: Wikipedia)
Microarrays: measuring gene expression

- Microarrays: one option.
- RNA hybridized onto a chip.
- Quantification of gene activity in different conditions at the genome scale.
- Resulting data: gene expression data.
Feature selection: extract relevant information

- Genome $\approx 25,000$ genes
- From gene expression, explain a phenomenon
- **Feature selection**: deciding which variables/genes are relevant.
Feature selection: extract relevant information

- Genome $\approx 25,000$ genes
- From gene expression, **explain** a phenomenon
- **Feature selection**: deciding which variables/genes are relevant.

Mathematically:

Explain response $Y$ using variables $(X_j)_{j=1...p}$

$$Y = f(X_1, X_2, X_3, X_4, \ldots, X_p) + \varepsilon$$
Feature selection: extract relevant information

- Genome ≈ 25,000 genes
- From gene expression, explain a phenomenon
- **Feature selection**: deciding which variables/genes are relevant.

Mathematically:

Explain response $Y$ using relevant variables amongst $(X_j)_{j=1...p}$

$$Y = f(X_1, X_2, X_3, X_4 \ldots, X_p) + \varepsilon$$

- **Objective 1**: more accurate predictors
- **Objective 2**: more interpretable predictors
- **Objective 3**: faster algorithms
Contributions of this thesis

Gene Regulatory Network Inference
- TIGRESS: new method based on local feature selection.
- Ranked 3rd/29 at DREAM5 challenge.
- Linear method, competitive with more complex algorithms

Molecular signatures for breast cancer prognosis
- Select biomarkers to predict metastasis/relapse in breast cancer patients.
- Complete benchmark of feature selection methods.
- Investigation of the stability issue.
Gene Regulatory Network Inference with TIGRESS

Gene Regulatory Networks

- **Gene regulation**: control gene expression.
- **Transcription factors (TF)** activate or repress **target genes (TG)**.
- **Gene Regulatory Network (GRN)**: representation of the regulatory interactions between genes.

*Figure: E. coli regulatory network*
DREAM network inference challenge

Network inference challenge:

DREAM5 results:

<table>
<thead>
<tr>
<th>Method</th>
<th>Network 1 AUPR</th>
<th>Network 1 AUROC</th>
<th>Network 3 AUPR</th>
<th>Network 3 AUROC</th>
<th>Network 4 AUPR</th>
<th>Network 4 AUROC</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENIE3(^1)</td>
<td>0.291</td>
<td>0.815</td>
<td>0.093</td>
<td>0.617</td>
<td>0.021</td>
<td>0.518</td>
<td>40.28</td>
</tr>
<tr>
<td>ANOVerence(^2)</td>
<td>0.245</td>
<td>0.780</td>
<td>0.119</td>
<td>0.671</td>
<td>0.022</td>
<td>0.519</td>
<td>34.02</td>
</tr>
<tr>
<td>Naive TIGRESS</td>
<td>0.301</td>
<td>0.782</td>
<td>0.069</td>
<td>0.595</td>
<td>0.020</td>
<td>0.517</td>
<td>31.1</td>
</tr>
</tbody>
</table>

\(^1\) Huynh-Thu et al., 2010

\(^2\) Kueffner et al., 2012
Purposes

- **Introduce TIGRESS**: Trustful Inference of Gene REgulation using Stability Selection.
- **Assess** the impact of the parameters.
- **Test** and **benchmark** TIGRESS on several datasets.
Outline

1 Methods
   - Regression-based inference
   - TIGRESS
   - Material

2 Results
   - In silico network results
   - In vitro networks results
   - Undirected case: DREAM4

3 Conclusion
Regression-based inference: hypotheses

**Notations**
- $n_{tf}$ transcription factors (TF), $n_{tg}$ target genes (TG), $n_{exp}$ experiments
- Expression data: $X (n_{exp} \times (n_{tf} + n_{tg}))$.
- $X_g$: expression levels of gene $g$.
- $X_G$: expression levels of genes in $G$.
- $T_g$: candidate TFs for gene $g$.

**Hypotheses**
1. The expression level $X_g$ of a TG $g$ is a function of the expression levels $X_{T_g}$ of $T_g$:

   $$X_g = f_g(X_{T_g}) + \varepsilon.$$

2. A score $s_g(t)$ can be derived from $f_g$, for all $t \in T_g$ to assess the probability of the interaction $(t, g)$. 
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

**subproblem $g$ $\iff$ find regulators $TFs(g)$ of gene $g$**

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

   
   - TF 12 $\rightarrow$ TG 17 $= 1$
   - TF 23 $\rightarrow$ TG 5 $= 0.99$
   - TF 2 $\rightarrow$ TG 1 $= 0.97$
   - ... $\rightarrow$ ... $= ...$

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

**subproblem $g \Leftrightarrow$ find regulators $TFs(g)$ of gene $g$**

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

   - TF 12 $\rightarrow$ TG 17 1
   - TF 23 $\rightarrow$ TG 5 0.99
   - TF 2 $\rightarrow$ TG 1 0.97
   - ... ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs \((n_{tg} \text{ subproblems})\)

**subproblem** \(g \Leftrightarrow \text{find regulators } TFs(g) \text{ of gene } g\)

1. For each TG, **score** all \(n_{tf}\) candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG ( n_{tg} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF ( n_{tf} )</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

\[
\begin{align*}
\text{TF 12} & \rightarrow \text{TG 17} \quad 1 \\
\text{TF 23} & \rightarrow \text{TG 5} \quad 0.99 \\
\text{TF 2} & \rightarrow \text{TG 1} \quad 0.97 \\
\end{align*}
\]

3. **Threshold** to a value or a given number \(N\) of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems) subproblem $g \Leftrightarrow$ find regulators $TFs(g)$ of gene $g$

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

- $TF_{12} \rightarrow TG_{17}$, 1
- $TF_{23} \rightarrow TG_{5}$, 0.99
- $TF_{2} \rightarrow TG_{1}$, 0.97
- ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

subproblem $g \Leftrightarrow$ find regulators $TFS(g)$ of gene $g$

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

- TF 12 $\rightarrow$ TG 17 1
- TF 23 $\rightarrow$ TG 5 0.99
- TF 2 $\rightarrow$ TG 1 0.97
- ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

**Subproblem $g \Leftrightarrow$ find regulators $TFs(g)$ of gene $g$**

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td>...</td>
<td>0.11</td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>TF$n_{tf}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0.76</td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

- TF 12 $\rightarrow$ TG 17 1
- TF 23 $\rightarrow$ TG 5 0.99
- TF 2 $\rightarrow$ TG 1 0.97
- ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

**subproblem $g \Leftrightarrow$ find regulators $TFs(g)$ of gene $g$**

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td>...</td>
<td>0.11</td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0.76</td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

- TF 12 $\rightarrow$ TG 17 1
- TF 23 $\rightarrow$ TG 5 0.99
- TF 2 $\rightarrow$ TG 1 0.97
- ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

subproblem $g \iff$ find regulators $TFs(g)$ of gene $g$

1. For each TG, **score** all $n_{tf}$ candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td>...</td>
<td>0.11</td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0.76</td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

   TF 12 $\rightarrow$ TG 17 1
   TF 23 $\rightarrow$ TG 5 0.99
   TF 2 $\rightarrow$ TG 1 0.97
   ...

3. **Threshold** to a value or a given number $N$ of edges.
Regression-based inference: main steps

Idea: consider as many problems as TGs \((n_{tg} \text{ subproblems})\)

Subproblem \(g \Leftrightarrow \text{find regulators } TFS(g) \text{ of gene } g\)

1. For each TG, **score** all \(n_{tf} \) candidate interactions:

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG (n_{tg})</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td>...</td>
<td>0.11</td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>TF (n_{tf})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0.76</td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

   - TF 12 → TG 17 1
   - TF 23 → TG 5 0.99
   - TF 2 → TG 1 0.97
   - ... ... ...

3. **Threshold** to a value or a given number \(N\) of edges.
Adding a linearity assumption

TIGRESS’ first hypothesis: regulations are linear

\[ X_g = f_g(X_{Tg}) + \varepsilon = X_{Tg} \omega^g + \varepsilon \]

Consequence: if \( \omega_t^g = 0 \), no edge between \( g \) and \( t \).
Adding a sparsity assumption

TIGRESS’ second hypothesis: few TFs regulate each TG

∀ \( g \), \( \# \{ \omega^g_t \neq 0 \} \ll n_{tf} \)

Possible algorithm: LARS with \( L \) steps \( \Rightarrow \) \( L \) TFs selected.
**Stability Selection**

**Problem:** LARS efficiency is limited:
- bad response to **correlation**;
- no confidence score for each TF.

**Solution:** Stability Selection with randomized LARS (Bach, 2008; Meinshausen and Bühlmann, 2009):
- **Resample the experiments:** run LARS many (e.g. 1,000) times with different training sets.
- “Resample” the variables: also weight the variables 
  \[ X_{it} \leftarrow W_t X_{it} \]  
  where \( W_j \sim \mathcal{U}([\alpha, 1]) \) for all \( t = 1 \ldots n_{tf} \). **The smaller** \( \alpha \), **the more randomized** the variables.
- Get a frequency of selection for each TF.
Stability Selection

Problem: LARS efficiency is limited:
- bad response to correlation;
- no confidence score for each TF.

Solution: Stability Selection with randomized LARS (Bach, 2008; Meinshausen and Bühlmann, 2009):
- Resample the experiments: run LARS many (e.g. 1,000) times with different training sets.
- “Resample” the variables: also weight the variables

\[ X_{it} \leftarrow W_t X_{it} \]  

(1)

where \( W_j \sim U([\alpha, 1]) \) for all \( t = 1 \ldots n_{tf} \). The smaller \( \alpha \), the more randomized the variables.
- Get a frequency of selection for each TF.
Stability Selection path

(example for one target gene)
Scoring

Choose $L$, then:

- **Original** scoring
- **Area** scoring (contribution)
Let $H_t$ be the **rank** of TF $t$. Then,

$$score = \mathbb{E}[\phi(H_t)]$$

with

- **Original**: $\phi(h) = 1$ if $h \leq L$, 0 otherwise
- **Area**: $\phi(h) = L + 1 - h$ if $h \leq L$, 0 otherwise

=> Area scoring takes the **value of the rank** into account.
Idea: consider as many problems as TGs ($n_{tg}$ subproblems)

**subproblem $g \Leftrightarrow$ find regulators $TFs(g)$ of gene $g$**

1. For each TG, **score** all $n_{tf}$ candidate interactions:

   LARS
   + Stab. Selection
   + Choose $L$
   + Score

<table>
<thead>
<tr>
<th></th>
<th>TG 1</th>
<th>TG 2</th>
<th>TG 3</th>
<th>...</th>
<th>TG $n_{tg}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TF 1</td>
<td>-</td>
<td>0.23</td>
<td>0</td>
<td>...</td>
<td>0.11</td>
</tr>
<tr>
<td>TF 2</td>
<td>0.97</td>
<td>-</td>
<td>0.03</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>TF $n_{tf}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0.76</td>
</tr>
</tbody>
</table>

2. **Rank** the scores altogether:

   - TF 12 $\rightarrow$ TG 17 1
   - TF 23 $\rightarrow$ TG 5 0.99
   - TF 2 $\rightarrow$ TG 1 0.97
   - ... ... ... ...

3. **Threshold** to a value or a given number $N$ of edges.
TIGRESS needs four parameters to be set:

- **scoring method** (original, area, ...);
- **number of runs** $R$: large;
- **randomization level** $\alpha$: between 0 and 1;
- **number of LARS steps** $L$: not obvious.
## Data

<table>
<thead>
<tr>
<th>Network</th>
<th># TF</th>
<th># Genes</th>
<th># Chips</th>
<th># Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>DREAM5 Net 1 (in-silico)</td>
<td>195</td>
<td>1643</td>
<td>805</td>
<td>4012</td>
</tr>
<tr>
<td>DREAM5 Net 3 (E. coli)</td>
<td>334</td>
<td>4511</td>
<td>805</td>
<td>2066</td>
</tr>
<tr>
<td>DREAM5 Net 4 (S. cerevisiae)</td>
<td>333</td>
<td>5950</td>
<td>536</td>
<td>3940</td>
</tr>
<tr>
<td><em>E. coli</em> Net from <em>Faith et al., 2007</em></td>
<td>180</td>
<td>1525</td>
<td>907</td>
<td>3812</td>
</tr>
<tr>
<td>DREAM4 Multifactorial Net 1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>176</td>
</tr>
<tr>
<td>DREAM4 Multifactorial Net 2</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>249</td>
</tr>
<tr>
<td>DREAM4 Multifactorial Net 3</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>195</td>
</tr>
<tr>
<td>DREAM4 Multifactorial Net 4</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>211</td>
</tr>
<tr>
<td>DREAM4 Multifactorial Net 5</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>193</td>
</tr>
</tbody>
</table>
Outline

1 Methods
   - Regression-based inference
   - TIGRESS
   - Material

2 Results
   - In silico network results
   - In vitro networks results
   - Undirected case: DREAM4

3 Conclusion
Impact of the parameters: results on in silico network

- **Area less sensitive** than original to $\alpha$ and $L$.
- **Area** systematically outperforms original.
- The more runs, the better
- Best values:
  - $\alpha = 0.4, L = 2, R = 10,000$. 

![Graphs showing the impact of parameters on the area with 1,000, 4,000, and 10,000 runs.](image)
How to choose $L$?

$L=2$: number of TFs/TG smaller and more variable.

$L=20$: greater number of TFs/TG, less sparsity.

$\Rightarrow L$ should depend on the expected network’s topology
TIGRESS is competitive with state-of-the-art.
Results on *E. coli* network

Random Forests-based and DREAM winner GENIE3 overperforms all methods.
False discovery analysis on *E. coli*

Main **false positive pattern** found by TIGRESS:

**Good news**: spuriously inferred edges close to true edges

**Bad news**: confusion (due to linear model?)
Undirected case: DREAM4 challenge

- DREAM4: *undirected networks* (TFs not known in advance)
- *A posteriori* comparison of Default TIGRESS and GENIE3:

<table>
<thead>
<tr>
<th>Method</th>
<th>Network 1</th>
<th>Network 2</th>
<th>Network 3</th>
<th>Network 4</th>
<th>Network 5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUPR</td>
<td>AUROC</td>
<td>AUPR</td>
<td>AUROC</td>
<td>AUPR</td>
</tr>
<tr>
<td>GENIE3</td>
<td>0.154</td>
<td>0.745</td>
<td>0.155</td>
<td>0.733</td>
<td>0.231</td>
</tr>
<tr>
<td>TIGRESS</td>
<td>0.165</td>
<td>0.769</td>
<td>0.161</td>
<td>0.717</td>
<td>0.233</td>
</tr>
</tbody>
</table>

Overall scores:
- GENIE3: 37.48
- TIGRESS: 38.85
Outline

1 Methods
   - Regression-based inference
   - TIGRESS
   - Material

2 Results
   - In silico network results
   - In vitro networks results
   - Undirected case: DREAM4

3 Conclusion
Conclusion

- **Contributions:**
  - **Automatization** and **adaptation** of Stability Selection to GRN inference.
  - **Area scoring setting:** better results and less elasticity to parameters.
  - Insights on network’s behavior

- **TIGRESS is:**
  - Linear
  - Competitive
  - Parallelizable
  - Available (http://cbio.ensmp.fr/tigress)
  - But outperformed in some cases by random forests: **limits of linearity?**

- **Perspectives:**
  - Adaptive/changeable value for $L$.
  - Group selection of TFs.
  - Use of time series/knock-out/replicates information.
Molecular signatures for breast cancer prognosis


Motivation

Prediction of breast cancer outcome
- Assist **breast cancer prognosis** based on gene expression.
- Avoid adjuvant/preventive chemotherapy when not needed.

Gene expression signature
- Data: primary site tumor expression arrays.
- Among the genome, **find the few (50-100) genes** sufficient to predict metastasis/relapse.
- Main challenge: high-dimensional data (few samples, many variables).
2002: *Van’t Veer et al.* publish 70-gene signature **MammaPrint**.

Since then: at least 47 *published signatures* ([Venet et al., 2011](#)).

**Little overlap**, if any ([Fan et al., 2006](#); [Thomassen et al., 2007](#)).

Many gene sets are **equally predictive** ([Michiels et al., 2005](#); [Ein-Dor et al., 2005](#)), even within the same dataset.

Prediction **discordances** ([Reyal et al., 2008](#))

Stability at the functional level? Not sure.

**Seeking stability**

**Accuracy is not enough** to choose the right genes.

=> Stability as a **confidence** indicator.
Background

- 2002: *Van’t Veer et al.* publish 70-gene signature **MammaPrint**.
- Since then: at least 47 **published signatures** (*Venet et al.*, 2011).
- Many gene sets are **equally predictive** (*Michiels et al.*, 2005; *Ein-Dor et al.*, 2005), even within the same dataset.
- Prediction **discordances** (*Reyal et al.*, 2008)
- Stability at the functional level? Not sure.

**Seeking stability**

**Accuracy is not enough** to choose the right genes.

=> Stability as a **confidence** indicator.
2002: Van’t Veer et al. publish 70-gene signature MammaPrint.

Since then: at least 47 published signatures (Venet et al., 2011).

Little overlap, if any (Fan et al., 2006; Thomassen et al., 2007).

Many gene sets are equally predictive (Michiels et al., 2005; Ein-Dor et al., 2005), even within the same dataset.

Prediction discordances (Reyal et al., 2008)

Stability at the functional level? Not sure.

Seeking stability

Accuracy is not enough to choose the right genes.

=> Stability as a confidence indicator.
2002: *Van’t Veer et al.* publish 70-gene signature **MammaPrint**. Since then: at least 47 published signatures ([Venet et al., 2011](#)). Little overlap, if any ([Fan et al., 2006; Thomassen et al., 2007](#)). Many gene sets are equally predictive ([Michiels et al., 2005; Ein-Dor et al., 2005](#)), even within the same dataset.

Prediction discordances ([Reyal et al., 2008](#))

Stability at the functional level? Not sure.

### Seeking stability

**Accuracy is not enough** to choose the right genes. => Stability as a **confidence** indicator.
2002: Van’t Veer *et al.* publish 70-gene signature **MammaPrint**.
Since then: at least 47 **published signatures** (*Venet et al.*, 2011).
Many gene sets are **equally predictive** (*Michiels et al.*, 2005; *Ein-Dor et al.*, 2005), even within the same dataset.
Prediction **discordances** (*Reyal et al.*, 2008)
Stability at the functional level? Not sure.

**Seeking stability**

**Accuracy is not enough** to choose the right genes.
=> Stability as a **confidence** indicator.
2002: Van’t Veer et al. publish 70-gene signature **MammaPrint**.

Since then: at least 47 **published signatures** (Venet et al., 2011).

**Little overlap**, if any (Fan et al., 2006; Thomassen et al., 2007).

Many gene sets are **equally predictive** (Michiels et al., 2005; Ein-Dor et al., 2005), even within the same dataset.

Prediction **discordances** (Reyal et al., 2008)

Stability at the functional level? Not sure.

---

**Seeking stability**

Accuracy is not enough to choose the right genes.

=> Stability as a **confidence** indicator.
2002: Van’t Veer et al. publish 70-gene signature MammaPrint.
Since then: at least 47 published signatures (Venet et al., 2011).
Little overlap, if any (Fan et al., 2006; Thomassen et al., 2007).
Many gene sets are equally predictive (Michiels et al., 2005; Ein-Dor et al., 2005), even within the same dataset.
Prediction discordances (Reyal et al., 2008)
Stability at the functional level? Not sure.

Seeking stability

Accuracy is not enough to choose the right genes.
=> Stability as a confidence indicator.
Contributions

1. **Systematic comparison** of feature selection methods in terms of:
   - predictive performance;
   - stability;
   - biological stability and interpretability.

2. **Group selection** of genes:
   - with predefined groups (Graph Lasso);
   - with latent groups (k-overlap norm).

3. **Evaluation of Ensemble methods**
Evaluation

- **Accuracy**: how well selected genes + classifier predict metastatic events on test data.
- **Stability**: how similar two lists of genes are.
- **Interpretability**: how much biological sense selected genes make.
Four public breast cancer datasets from the same technology (Affymetrix U133A):

<table>
<thead>
<tr>
<th>GEO Reference</th>
<th># genes</th>
<th># samples</th>
<th># positives</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSE1456</td>
<td>12,065</td>
<td>159</td>
<td>40</td>
</tr>
<tr>
<td>GSE2034</td>
<td>12,065</td>
<td>286</td>
<td>107</td>
</tr>
<tr>
<td>GSE2990</td>
<td>12,065</td>
<td>125</td>
<td>49</td>
</tr>
<tr>
<td>GSE4922</td>
<td>12,065</td>
<td>249</td>
<td>89</td>
</tr>
</tbody>
</table>
1. A simple start

2. An attempt at enforcing stability: Ensemble methods

3. Using prior knowledge: Graph Lasso

4. Acknowledging latent team work

5. Conclusion
### Classical feature selection/feature ranking methods

<table>
<thead>
<tr>
<th>Type</th>
<th>Characteristics</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filters</td>
<td>Univariate, fast&lt;br&gt;Only depend on the data&lt;br&gt;Do not use loss function</td>
<td>$T$-test, KL Divergence&lt;br&gt;Wilcoxon rank-sum test</td>
</tr>
<tr>
<td>Wrappers</td>
<td>Learning machine as a criterion&lt;br&gt;Computationally expensive</td>
<td>SVM RFE, Greedy FS</td>
</tr>
<tr>
<td>Embedded</td>
<td>Learning + selecting&lt;br&gt;Possible use of prior knowledge</td>
<td>Lasso, Elastic Net</td>
</tr>
</tbody>
</table>

Each of these algorithms returns a **ranked list of genes** to be thresholded.
First results

100 genes over four databases.

Accuracy vs. Stability

- Random
- Ttest
- Entropy
- Bhattacharyya
- Wilcoxon
- SVM RFE
- GFS
- Lasso
- E−Net
First conclusions

- **Random** better than tossing a coin.
- **Elastic Net** neither more stable nor more accurate than Lasso.
- **Accuracy/Stability trade-off**
- **T-test**: both simplest and best.

Next step:

Can we have a better stability without decreasing accuracy?
First conclusions

- **Random** better than tossing a coin.
- **Elastic Net** neither more stable nor more accurate than Lasso.
- **Accuracy/Stability trade-off**
- **T-test**: both simplest and best.

Next step:

Can we have a better stability without decreasing accuracy?
Outline

1. A simple start

2. An attempt at enforcing stability: Ensemble methods

3. Using prior knowledge: Graph Lasso

4. Acknowledging latent team work

5. Conclusion
Run each algorithm \( R \) times on subsamples.
Get \( R \) ranked lists of genes \((r^b)_b=1...R\).
Aggregate and get a score for each gene:

\[
S(g) = \frac{1}{R} \sum_{b=1}^{R} f(r_g^b).
\]

- average: \( f(r) = (p - r)/p \)
- exponential: \( f(r) = \exp(-\alpha r) \)
- stability selection: \( f(r) = \delta(r \leq k) \)

Sort \( S \) by decreasing order and threshold to get final signature
Results

100 genes over four databases.

Accuracy vs. Stability graph with various algorithms represented:
- Random
- Ttest
- Entropy
- Bhattacharyya
- Wilcoxon
- SVM RFE
- GFS
- Lasso
- E−Net
- Single−run
- Stab. sel
100 genes over four databases.
Expected improvement in **stability** not happening.
Slight improvement in **accuracy** in some cases.
Loss in **functional stability**.
**T-test**: still the preferred method.

**Next step:**
Can we do better by incorporating prior knowledge?
Expected improvement in **stability** not happening.

Slight improvement in **accuracy** in some cases.

Loss in **functional stability**.

**T-test**: still the preferred method.

**Next step:**

Can we do better by incorporating prior knowledge?
1. A simple start
2. An attempt at enforcing stability: Ensemble methods
3. Using prior knowledge: Graph Lasso
4. Acknowledging latent team work
5. Conclusion
• **Expression data**: Van’t Veer et al., 2002; Wang et al., 2005.

• **PPI network** with 8141 genes (Chuang et al., 2007)

• **Assumption**: genes close on the graph behave similarly

• **Idea**: instead of selecting single genes, select **edges**
Selecting groups of genes: $\ell_1$ methods

- **Lasso**: selects single genes (Tibshirani, 1996)

**Why groups?**
- selecting similar genes: improving stability and interpretability
- smoothing out noise by "averaging": improving accuracy

- **Group Lasso** (Yuan & Lin, 2006): implies *group sparsity* for groups of covariates that form a partition of $\{1...p\}$
- **Overlapping group Lasso** (Jacob et al., 2009): selects a union of *potentially overlapping* groups of covariates (e.g. gene pathways).
- **Graph Lasso**: uses groups induced by the graph (e.g. edges)
Is group sparsity enough?

- $\ell_1$ methods work well, but face serious **stability issues when groups are correlated**.
- Solution: randomization through **stability selection**.
Accuracy results

Test on data from *Wang et al., 2005*.

Neither prior knowledge nor stability selection bring any improvement!
Graph Lasso slightly improves stability.
Signature obtained using **Lasso**:
Signature obtained using **Graph Lasso + Stability Selection**:
Graph Lasso conclusion

- **Graphical prior** seems to increase stability and interpretability.
- However: no change in accuracy.

**Next step:**

Grouping increases stability. Now on to accuracy!
1. A simple start
2. An attempt at enforcing stability: Ensemble methods
3. Using prior knowledge: Graph Lasso
4. Acknowledging latent team work
5. Conclusion
Latent grouping

- **Grouping genes** makes sense.
- **Let the data tell** which genes to select together.

The k-support norm

- Introduced by *Argyriou et al., 2012*
- A **trade-off** between $\ell_1$ and $\ell_2$.
- Equivalent to **overlapping group Lasso** (*Jacob et al., 2009*) with all possible groups of size $k$.
- Results in selecting groups that are not predefined.
Extreme randomization

- Following Breiman’s **random forests**
- Sample both the examples and the covariates.
- **Less variables = less correlation.**
- Give each gene a chance to be selected.

For each of the $R$ runs:
- Bootstrap samples (classical Ensemble method)
- **Sample the covariates:** randomly choose 10% of them.
- Run FS procedure on the restricted data.

$\Rightarrow$ Compute frequency of selection: $\mathbb{P}(g \text{ selected} | g \text{ preselected})$
Accuracy or stability?

Accuracy and stability are plotted on a graph. The x-axis represents accuracy, ranging from 0.62 to 0.66, while the y-axis represents stability, ranging from 0.00 to 0.05.

Different methods and techniques are represented with distinct markers and colors:
- Random
- T-test
- Lasso
- ENet
- kSupport (k=2)
- kSupport (k=10)
- kSupport (k=20)
- Single-run
- Extreme Rand. + SS
- T-test

The graph visually compares the performance of these methods in terms of accuracy and stability.
Stability or redundancy?

![Graph showing stability vs. correlation for different methods]

- Random
- Ttest
- Lasso
- ENet
- kSupport (k=2)
- kSupport (k=10)
- kSupport (k=20)
- Single-run
- Extreme Rand. + SS

Correlation
Stability
Extreme Randomization improves accuracy.

Grouping improves stability.

But: both effects do not add up that well (redundancy)

T-test still the best trade-off?
1. A simple start
2. An attempt at enforcing stability: Ensemble methods
3. Using prior knowledge: Graph Lasso
4. Acknowledging latent team work
5. Conclusion
Contributions:

- **Step-by-step** study of FS methods behavior on several breast cancer datasets.
- **Systematic analysis** of accuracy, stability, interpretability.
- **Insights** on the accuracy/stability trade-off.

What have we learned?

- Best methods: simple t-test **or** complex black box
- **Grouping** improves gene and functional stability.
- **Randomization** improves accuracy (sometimes) but has unwanted effects on stability.
- **Accuracy/Stability Trade-off**: Stability $\Rightarrow$ redundancy $\Rightarrow$ lower accuracy.
Signature selection: perspectives

One unique signature?
- single breast cancer subtype
- many samples
- larger signature

Is expression data sufficient?
- probably not all information is there
- clinical data: same accuracy (same information?)
- possibly look at genotype, methylation, clinical and expression

Is stability important?
- not as important as accuracy + prediction concordance
- possibly not even achievable
Conclusion
Conclusion

Gene expression data:
- High-dimensional, noisy
- Possibly contains **important information**

Feature selection:
- Find the needle in the haystack.
- Output **relevant genes** to be studied further.

Main issues:
- Results are not necessarily **transferable** across datasets.
- Models rely on **hypotheses**!

Fixing:
- Testing on **many** databases.
- Keeping model hypotheses in mind / not being afraid of **black boxes**.
Conclusion

Gene expression data:
- High-dimensional, noisy
- Possibly contains **important information**

Feature selection:
- Find the needle in the haystack.
- Output **relevant genes** to be studied further.

Main issues:
- Results are not necessarily **transferable** across datasets.
- Models rely on **hypotheses**!

Fixing:
- Testing on **many** databases.
- Keeping model hypotheses in mind / not being afraid of **black boxes**.
Conclusion

Gene expression data:
- High-dimensional, noisy
- Possibly contains **important information**

Feature selection:
- Find the needle in the haystack.
- Output **relevant genes** to be studied further.

Main issues:
- Results are not necessarily **transferable** across datasets.
- Models rely on **hypotheses**!

Fixing:
- Testing on **many** databases.
- Keeping model hypotheses in mind / not being afraid of **black boxes**.
Conclusion

Gene expression data:
- High-dimensional, noisy
- Possibly contains important information

Feature selection:
- Find the needle in the haystack.
- Output relevant genes to be studied further.

Main issues:
- Results are not necessarily transferable across datasets.
- Models rely on hypotheses!

Fixing:
- Testing on many databases.
- Keeping model hypotheses in mind / not being afraid of black boxes.
Acknowledgements

Fantine Mordelet
Paola Vera-Licona
Pierre Gestraud
Laurent Jacob
The k-support norm

It can be shown that:

\[
\Omega_{k}^{sp}(\omega) = \left( \sum_{i=1}^{k-r-1} (|\omega|_{i}^\downarrow)^2 + \frac{1}{r+1} \left( \sum_{i=k-r}^{d} |\omega|_{i}^\downarrow \right)^2 \right)^{\frac{1}{2}}
\]

where \( r \) is the only integer in \( \{0, \ldots, k-1\} \) satisfying

\[
|\omega|_{k-r-1}^\downarrow > \frac{1}{r+1} \sum_{i=k-r}^{d} |\omega|_{i}^\downarrow \geq |\omega|_{k-r}^\downarrow.
\]

and \( |\omega|_{i}^\downarrow \) is the \( i \)-th largest value of \( |\omega| \) (\( |\omega|_{0}^\uparrow = +\infty \)).
The k-support norm is equivalent to the overlapping group Lasso norm

\[
\Omega^\text{sp}_k(\omega) = \min_{v \in \mathbb{R}^{p \times \mathcal{G}_k}} \left\{ \sum_{I \in \mathcal{G}_k} \| v_I \|_2 : \text{supp}(v_I) \subseteq I, \sum_{I \in \mathcal{G}_k} v_I = \omega \right\}
\]

where \( \mathcal{G}_k \) denotes all subsets of \( \{1, \ldots, d\} \) of cardinality \( k \).

- **Remark 1:** it selects at least \( k \) variables.
- **Remark 2:** the first selected group consists of the \( k \) variables most correlated with the response.
ADMM - applied to k-support problem

Our problem

\[
\begin{align*}
\min_{\omega, \beta} & \quad \hat{R}_l(\omega) + \frac{\lambda}{2} \Omega_k^{sp}(\beta)^2 \\
\text{s.t.} & \quad \omega - \beta = 0
\end{align*}
\]

Augmented Lagrangian

\[
\mathcal{L}_\rho(\omega, \beta, \mu) = \hat{R}_l(\omega) + \frac{\lambda}{2} \Omega_k^{sp}(\beta)^2 + \mu'(\omega - \beta) + \frac{\rho}{2} ||\omega - \beta||^2
\]

Algorithm

1. Initialize: \(\beta^{(1)}, \omega^{(1)}, \mu^{(1)}\)
2. for \(t = 1, 2, \ldots\), do
   \[
   \begin{align*}
   w^{(t+1)} &= \arg \min_w \left\{ \hat{R}_l(w) + \mu^{(t)T} w + \frac{\rho}{2} ||w - \beta^{(t)}||^2 \right\} \\
   \beta^{(t+1)} &= \text{prox}_{\frac{\lambda}{2\rho} \Omega_k^{sp}(\cdot)^2} \left( w^{(t+1)} + \frac{\mu^{(t)}}{\rho} \right) \\
   \mu^{(t+1)} &= \mu^{(t)} + \rho \left( w^{(t+1)} - \beta^{(t+1)} \right)
   \end{align*}
   \]
Three first order conditions:

- **Primal condition:** \( \omega^* - \beta^* = 0 \)
- **Dual condition 1:** \( \nabla \hat{R}_l(\omega^*) + \mu^* = 0 \)
- **Dual condition 2:** \( 0 \in \frac{\lambda}{2} \partial \Omega_k^{sp} (\beta^*)^2 + \mu^* \)

Resulting in a definition for the residuals at step \( t + 1 \):

- **Primal residuals:** \( r^{(t+1)} = \omega^{(t+1)} - \beta^{(t+1)} \)
- **Dual residuals:** \( s^{(t+1)} = \rho(\beta^{(t+1)} - \beta^{(t)}) \)

As the algorithm converges, the (norm of the) residuals tend to zero.
Parameter $\rho$ is **critical** in ADMM: it controls **how much** variables change. It can be seen as a **step size**.

**How to choose it?**

**Adaptive ADMM**

One solution is to let it adapt to the problem:

$$\rho^{(t+1)} = \begin{cases} 
(1 + \tau)\rho^{(t)} & \text{if } ||r^{(t+1)}||_2 > \eta||s^{(t+1)}||_2 \text{ and } t \leq t_{max} \\
\rho^{(t)}/(1 + \tau) & \text{if } \eta||r^{(t+1)}||_2 < ||s^{(t+1)}||_2 \text{ and } t \leq t_{max} \\
\rho^{(t)} & \text{otherwise}
\end{cases}$$

In practice, we use $\tau = 1$, $\eta = 10$ and $t_{max} = 100$.

$\Rightarrow$ Adaptive ADMM forces the primal and dual residuals to be of a similar amplitude.
Comparison

$k = 1$

$k = 5$

$k = 10$

$k = 100$

- ADMM - adaptive
- ADMM - \(\rho=1\)
- ADMM - \(\rho=10\)
- ADMM - \(\rho=100\)
- FISTA
Accuracy vs size of the signature

Single-run

Ensemble-Mean

Ensemble-Exponential

Ensemble-Stability Selection

Random T-test Entropy Bhatt. Wilcoxon SVM RFE GFS Lasso E-Net
Stability vs size of the signature

Single-run

Ensemble-average

Ensemble-exponential

Ensemble-stability selection

- Random
- T test
- Entropy
- Bhatt.
- Wilcoxon
- RFE
- GFS
- Lasso
- E−Net