Fast sparse methods for genomics data

Jean-Philippe Vert

Chalmers University, April 16, 2013
Normal vs cancer cells

What goes wrong?
How to treat?
Biology is now quantitative, "high-throughput"
"The $1,000 genome, the $1 million interpretation" (B. Kopf)

High-dimensional, heterogeneous, structured data. "Large $p$"

http://aws.amazon.com/1000genomes/
In this talk

\[
\min_w R(w) + \lambda \Omega(w)
\]

where:

- \( w \) is the hypothesis we want to infer from data
- \( R(w) \) is a smooth convex "fitness" function
- \( \Omega(w) \) is a non-smooth convex penalty, which favors particular solution

1. Mapping DNA breakpoints in cancer genomes
2. Isoform detection from RNA-seq data
3. Inference of gene regulatory networks
1. Mapping DNA breakpoints in cancer genomes
2. Isoform detection from RNA-seq data
3. Inference of gene regulatory networks
Comparative Genomic Hybridization (CGH)

Motivation

- Comparative genomic hybridization (CGH) data measure the DNA copy number along the genome.
- Very useful, in particular in cancer research to observe systematically variants in DNA content.
Can we identify breakpoints and "smooth" each profile?

- A classical multiple change-point detection problem
- Should scale to lengths of order $10^6 \sim 10^9$
Can we identify breakpoints and "smooth" each profile?

- A classical multiple change-point detection problem
- Should scale to lengths of order $10^6 \sim 10^9$
An optimal solution

For a signal \( Y \in \mathbb{R}^p \), define an optimal approximation \( \beta \in \mathbb{R}^p \) with \( k \) breakpoints as the solution of

\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(\beta_{i+1} \neq \beta_i) \leq k
\]

This is an optimization problem over the \( \binom{p}{k} \) partitions...

Dynamic programming finds the solution in \( O(p^2k) \) in time and \( O(p^2) \) in memory

But: does not scale to \( p = 10^6 \sim 10^9 \)...
For a signal $Y \in \mathbb{R}^p$, define an optimal approximation $\beta \in \mathbb{R}^p$ with $k$ breakpoints as the solution of

$$\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(\beta_{i+1} \neq \beta_i) \leq k$$

This is an optimization problem over the $\binom{p}{k}$ partitions...

Dynamic programming finds the solution in $O(p^2 k)$ in time and $O(p^2)$ in memory.

But: does not scale to $p = 10^6 \sim 10^9$...
For a signal \( Y \in \mathbb{R}^p \), define an optimal approximation \( \beta \in \mathbb{R}^p \) with \( k \) breakpoints as the solution of

\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(\beta_{i+1} \neq \beta_i) \leq k
\]

This is an optimization problem over the \( \binom{p}{k} \) partitions...

Dynamic programming finds the solution in \( O(p^2k) \) in time and \( O(p^2) \) in memory...

But: does not scale to \( p = 10^6 \sim 10^9 \)...
For a signal \( Y \in \mathbb{R}^p \), define an optimal approximation \( \beta \in \mathbb{R}^p \) with \( k \) breakpoints as the solution of

\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(\beta_{i+1} \neq \beta_i) \leq k
\]

This is an optimization problem over the \( \binom{p}{k} \) partitions...

**Dynamic programming** finds the solution in \( O(p^2k) \) in time and \( O(p^2) \) in memory

**But:** does not scale to \( p = 10^6 \sim 10^9 \)...
Promoting sparsity with the $\ell_1$ penalty

The $\ell_1$ penalty (Tibshirani, 1996; Chen et al., 1998)

If $R(\beta)$ is convex and "smooth", the solution of

$$\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda \sum_{i=1}^{p} |\beta_i|$$

is usually sparse.

Geometric interpretation with $p = 2$
Promoting piecewise constant profiles penalty

The total variation / variable fusion penalty

If $R(\beta)$ is convex and "smooth", the solution of

$$\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i|$$

is usually piecewise constant (Rudin et al., 1992; Land and Friedman, 1996).

Proof:

- Change of variable $u_i = \beta_{i+1} - \beta_i$, $u_0 = \beta_1$
- We obtain a Lasso problem in $u \in \mathbb{R}^{p-1}$
- $u$ sparse means $\beta$ piecewise constant
\[
\min_{\beta \in \mathbb{R}^p} \| Y - \beta \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \leq \mu
\]

Adding additional constraints does not change the change-points:

- \( \sum_{i=1}^p |\beta_i| \leq \nu \) (Tibshirani et al., 2005; Tibshirani and Wang, 2008)
- \( \sum_{i=1}^p \beta_i^2 \leq \nu \) (Mairal et al. 2010)
Solving TV signal approximator

\[
\begin{align*}
\min_{\beta \in \mathbb{R}^p} & \| Y - \beta \|_2^2 \\
\text{such that} & \sum_{i=1}^{p-1} |\beta_{i+1} - \beta_i| \leq \mu
\end{align*}
\]

- QP with sparse linear constraints in \(O(p^2)\) \(\rightarrow 135\) min for \(p = 10^5\) (Tibshirani and Wang, 2008)
- Coordinate descent-like method \(O(p)\)? \(\rightarrow 3\) s s for \(p = 10^5\) (Friedman et al., 2007)
- With the LARS in \(O(pk)\) (Harchaoui and Levy-Leduc, 2008)
- For all \(\mu\) in \(O(p \ln p)\) (Hoefling, 2009)
- For the first \(k\) change-points in \(O(p \ln k)\) (Bleakley and V., 2010)
Problem formulation

Let \( Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n \) a signal that we wish to approximate by a piecewise-constant signal \( \mu \). We will investigate different function partitioning methods.

\[
\text{for } i = 1 \text{ to } k \text{ do}
\]

4: \( \mathcal{I}^\ast \leftarrow \arg\max_{I \in \mathcal{P}} \gamma(I) \)

5: \( \mathcal{P} \leftarrow \mathcal{P} \setminus \{I^\ast\} \)

6: \( \mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^\ast), I_R(I^\ast)\} \)

7: \text{end for}

8: \text{return } \mathcal{P}

Theorem (V. and Bleakley, 2010; see also Hoefling, 2009)

TV signal approximator performs "greedy" dichotomich segmentation

\[
\text{Algorithm 1 Greedy dichotomich segmentation}
\]

\[\text{Require: } k \text{ number of intervals, } \gamma(I) \text{ gain function to split an interval } I \text{ into } I_L(I), I_R(I)\]

1: \( I_0 \) represents the interval \([1, n]\)

2: \( \mathcal{P} = \{I_0\} \)

3: \text{for } i = 1 \text{ to } k \text{ do}

4: \( \mathcal{I}^\ast \leftarrow \arg\max_{I \in \mathcal{P}} \gamma(I) \)

5: \( \mathcal{P} \leftarrow \mathcal{P} \setminus \{I^\ast\} \)

6: \( \mathcal{P} \leftarrow \mathcal{P} \cup \{I_L(I^\ast), I_R(I^\ast)\} \)

7: \text{end for}

8: \text{return } \mathcal{P}

Apparentely greedy algorithm finds the global optimum!
Speed trial: 2 s. for $k = 100$, $p = 10^7$
Extension 1: linear discrimination / regression

Aggressive (left) vs non-aggressive (right) melanoma
Idea: find a linear predictor \( f(Y) = \beta^T Y \) that best discriminates the aggressive vs non-aggressive samples, subject to the constraints that it should be sparse and piecewise constant.

Mathematically:

\[
\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda_1 \| \beta_1 \|_1 + \lambda_2 \| \beta \|_{TV}
\]

Computationnally: proximal methods
Idea: find a linear predictor $f(Y) = \beta^T Y$ that best discriminates the aggressive vs non-aggressive samples, subject to the constraints that it should be sparse and piecewise constant.

Mathematically:

$$\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|_{TV}$$

Computationnally: proximal methods
Fused lasso for supervised classification

**Idea:** find a linear predictor \( f(Y) = \beta^T Y \) that best discriminates the aggressive vs non-aggressive samples, subject to the constraints that it should be sparse and piecewise constant.

**Mathematically:**

\[
\min_{\beta \in \mathbb{R}^p} R(\beta) + \lambda_1 \| \beta \|_1 + \lambda_2 \| \beta \|_{TV}
\]

**Computationnally:** proximal methods
Extension 2: finding multiple change points shared by several profiles
Extension 2: finding multiple change points shared by several profiles
Define the "optimal" piecewise constant approximation $\hat{U} \in \mathbb{R}^{p \times n}$ of $Y$ as the solution of

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1(U_{i+1, \bullet} \neq U_{i, \bullet}) \leq k$$

DP finds the solution in $O(p^2 kn)$ in time and $O(p^2)$ in memory.

But: does not scale to $p = 10^6 \sim 10^9$...
Selecting pre-defined groups of variables

**Group lasso (Yuan & Lin, 2006)**

If groups of covariates are likely to be selected together, the $\ell_1/\ell_2$-norm induces sparse solutions *at the group level*:

$$\Omega_{\text{group}}(w) = \sum_g \|w_g\|_2$$

$$\Omega(w_1, w_2, w_3) = \|(w_1, w_2)\|_2 + \|w_3\|_2 = \sqrt{w_1^2 + w_2^2 + w_3^2}$$

![Diagram showing the geometry of group lasso penalty](image-url)
GFLseg (Bleakley and V., 2011)

Replace

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} 1 (U_{i+1, \cdot} \neq U_{i, \cdot}) \leq k
\]

by

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1, \cdot} - U_{i, \cdot} \| \leq \mu
\]

GFLseg = Group Fused Lasso segmentation

Questions

- Practice: can we solve it efficiently?
- Theory: does it recover the correct segmentation?
Replace

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} {\mathbf{1}(U_{i+1,\bullet} \neq U_{i,\bullet})} \leq k$$

by

$$\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\bullet} - U_{i,\bullet} \| \leq \mu$$

GFLseg = Group Fused Lasso segmentation

Questions

- **Practice**: can we solve it efficiently?
- **Theory**: does it recover the correct segmentation?
GFLseg as a group Lasso problem

- Make the change of variables:
  \[ \gamma = U_{1,\bullet}, \]
  \[ \beta_{i,\bullet} = w_i (U_{i+1,\bullet} - U_{i,\bullet}) \quad \text{for } i = 1, \ldots, p - 1. \]

- TV approximator is then equivalent to the following group Lasso problem (Yuan and Lin, 2006):
  \[
  \min_{\beta \in \mathbb{R}^{(p-1) \times n}} \| \bar{Y} - \bar{X} \beta \|^2 + \lambda \sum_{i=1}^{p-1} \| \beta_{i,\bullet} \|,
  \]
  where \( \bar{Y} \) is the centered signal matrix and \( \bar{X} \) is a particular \( (p - 1) \times (p - 1) \) design matrix.
The TV approximator can be solved efficiently:

- "approximately" with the group LARS in $O(npk)$ in time and $O(np)$ in memory
- "exactly" with a block coordinate descent + active set method in $O(np)$ in memory
Although \( \bar{X} \) is \((p - 1) \times (p - 1)\):

- For any \( R \in \mathbb{R}^{p \times n} \), we can compute \( C = \bar{X}^\top R \) in \( O(np) \) operations and memory.

- For any two subset of indices \( A = (a_1, \ldots, a_{|A|}) \) and \( B = (b_1, \ldots, b_{|B|}) \) in \([1, p - 1]\), we can compute \( \bar{X}_{\bullet,A} \bar{X}_{\bullet,B} \) in \( O(|A||B|) \) in time and memory.

- For any \( A = (a_1, \ldots, a_{|A|}) \), set of distinct indices with \( 1 \leq a_1 < \ldots < a_{|A|} \leq p - 1 \), and for any \( |A| \times n \) matrix \( R \), we can compute \( C = \left( \bar{X}_{\bullet,A} \bar{X}_{\bullet,A} \right)^{-1} R \) in \( O(|A|n) \) in time and memory.
Figure 2: **Speed trials for group fused LARS (top row) and Lasso (bottom row).** *Left column:* varying \( n \), with fixed \( p = 10 \) and \( k = 10 \); *center column:* varying \( p \), with fixed \( n = 1000 \) and \( k = 10 \); *right column:* varying \( k \), with fixed \( n = 1000 \) and \( p = 10 \). Figure axes are log-log. Results are averaged over 100 trials.
Consistency

Suppose a single change-point:
- at position $u = \alpha p$
- with increments $(\beta_i)_{i=1,...,n}$ s.t. $\beta^2 = \lim_{k \to \infty} \frac{1}{n} \sum_{i=1}^{n} \beta_i^2$
- corrupted by i.i.d. Gaussian noise of variance $\sigma^2$

Does the TV approximator correctly estimate the first change-point as $p$ increases?
Consistency of the unweighted TV approximator

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} \| U_{i+1, \bullet} - U_{i, \bullet} \| \leq \mu
\]

**Theorem**

*The unweighted TV approximator finds the correct change-point with probability tending to 1 (resp. 0) as \( n \to +\infty \) if \( \sigma^2 < \tilde{\sigma}_\alpha^2 \) (resp. \( \sigma^2 > \tilde{\sigma}_\alpha^2 \)), where*

\[
\tilde{\sigma}_\alpha^2 = p\beta^2 \frac{(1 - \alpha)^2(\alpha - \frac{1}{2p})}{\alpha - \frac{1}{2} - \frac{1}{2p}}.
\]

- correct estimation on \([p\epsilon, p(1 - \epsilon)]\) with \( \epsilon = \sqrt{\frac{\sigma^2}{2p\beta^2}} + o(p^{-1/2}) \).
- wrong estimation near the boundaries
Consistency of the weighted TV approximator

\[
\min_{U \in \mathbb{R}^{p \times n}} \| Y - U \|_2^2 \quad \text{such that} \quad \sum_{i=1}^{p-1} w_i \| U_{i+1,\cdot} - U_{i,\cdot} \| \leq \mu
\]

Theorem

The weighted TV approximator with weights

\[
\forall i \in [1, p - 1], \quad w_i = \sqrt{\frac{i(p - i)}{p}}
\]

correctly finds the first change-point with probability tending to 1 as \( n \to +\infty \).

- we see the benefit of increasing \( n \)
- we see the benefit of adding weights to the TV penalty
The first change-point $\hat{i}$ found by TV approximator maximizes $F_i = \| \hat{c}_{i,\bullet} \|^2$, where

$$\hat{c} = \bar{X}^\top \bar{Y} = \bar{X}^\top \bar{X} \beta^* + \bar{X}^\top W.$$

$\hat{c}$ is Gaussian, and $F_i$ is follows a non-central $\chi^2$ distribution with

$$G_i = \frac{EF_i}{p} = \frac{i(p - i)}{pw_i^2} \sigma^2 + \frac{\beta^2}{w_i^2 w_u^2 p^2} \times \begin{cases} i^2 (p - u)^2 & \text{if } i \leq u, \\ u^2 (p - i)^2 & \text{otherwise}. \end{cases}$$

We then just check when $G_u = \max_i G_i$.
6.3 Accuracy for detecting multiple change-points

To investigate the potential for extending the results to the case of many change-points, we further simulated profiles of length $n = 100$ with a change-point at all of positions $10, 20, \ldots, 90$. We considered dimensions $p$ between 1 and 500. Jumps at each change-point are drawn from a Gaussian with mean 0 and variance 1; we then added centered Gaussian noise with $\sigma^2 \in \{0.05, 0.2, 1\}$ to each position in each profile. For each value of $p$ and $\sigma^2$, we ran 100 trials for both implementations, with or without weights, and recorded the accuracy of each method, defined as the percentage of trials where the first 9 change-points detected by the method are exactly the 9 true change-points. Results are presented in Figure 4 (from left to right, resp. $\sigma^2 = 0.05, 0.2, 1$). Clearly, the group fused Lasso outperforms the group fused LARS, and the weighted version of each algorithm outperforms the unweighted version. Although the group LARS is usually considered a reliable alternative to the exact group Lasso [21], this experiment shows that the exact optimization by block coordinate descent may be worth the computational burden if one is interested in accurate group selection. It also demonstrates that, as we conjectured in Section 5.3, the group fused Lasso can consistently estimate multiple change-points as the number of profiles increases.

6.4 Application to gain and loss detection

We now consider a possible application of our method for the detection of regions with frequent gains (positive values) and losses (negative values) among a set of DNA copy number profiles, measured by array comparative genomic hybridization (aCGH) technology [27]. We propose a two-step strategy for this purpose: first, find an adequate joint segmentation of the signals; then, check the presence of gains or losses.

Figure 3: **Single change-point accuracy for the group fused Lasso.** Accuracy as a function of the number of profiles $p$ when the change-point is placed in a variety of positions $u = 50$ to $u = 90$ (left and centre plots, resp. unweighted and weighted group fused Lasso), or: $u = 50 \pm 2$ to $u = 90 \pm 2$ (right plot, weighted with varying change-point location), for a signal of length 100.
Estimation of several change-points

Figure 4: **Multiple change-point accuracy.** Accuracy as a function of the number of profiles $p$ when change-points are placed at the nine positions $\{10, 20, \ldots, 90\}$ and the variance $\sigma^2$ of the centered Gaussian noise is either $0.05$ (left), $0.2$ (center) and $1$ (right). The profile length is $100$. 

Note that we do not assume that all profiles share exactly the same change-points, but merely see the joint segmentation as an adaptive way to reduce the dimension and remove noise from data. In practice, we used group fused LARS on each chromosome to identify a set of $100$ candidate change-points, and selected a subset of them by post-processing as described in Section 5.4. Then, in each piecewise-constant interval between successive shared change-points, we calculate the mean of the positive segments (shown in green in Figures 5(a) and 6(c)) and the mean of the negative segments (shown in red). The larger the mean of the positive segments, the more likely we are to believe that a region harbors an important common gain; the reasoning is analogous for important common losses and the mean of the negative segments. Obviously, many other statistical tests could be carried out to detect frequent gains and losses on each segment, once the joint segmentation is performed. We compare this method for detecting regions of gain and loss with the state-of-the-art H-HMM method [27], which has been shown to outperform several other methods in this setting. As [27] have provided their algorithm online with several of their data sets tested in their article, we implemented our method and theirs (H-HMM) on their benchmark data sets.

In the first data set in [27], the goal is to recover two regions – one amplified, one deleted, that are shared in $8$ short profiles, though only $6$ of the profiles exhibit each of the amplified or deleted regions. Performance is measured by area under ROC curve (AUC), following [27]. Running H-HMM with the default parameters, we obtained an AUC (averaged over $10$ trials) of $0.96 \pm 0.01$, taking on average $60.20$ seconds. The weighted group fused LARS, asked to select $100$ breakpoints and followed by dynamic programming, took $0.06$ seconds and had an AUC of $0.97$. Thus, the performance of both methods was similar, though weighted group fused LARS was around $1000$ times faster.

The second data set was a cohort of lung cancer cell lines originally published in [28, 29]. As in [27], we concentrated on the $18$ NSCLC adenocarcinoma (NA) cell lines. Figure 5 shows the scores statistic obtained on Chromosome 8 when using either weighted group fused LARS or H-HMM. Weighted group fused LARS...

---

**JP Vert (ParisTech)**

Sparse methods in genomics
Application: detection of frequent abnormalities
1. Mapping DNA breakpoints in cancer genomes

2. Isoform detection from RNA-seq data

3. Inference of gene regulatory networks
Alternative splicing: 1 gene = many proteins
The isoform identification and quantification problem

Given a biological sample (e.g., cancer tissue), can we:

1. identify the isoform(s) of each gene present in the sample?
2. quantify their abundance?
RNA-seq measures mRNA abundance by sequencing short fragments.
RNA-seq and alternative splicing

(Costa et al., 2011)
From RNA-seq to isoforms

**RNA sample transcripts**

**reads**
50-200pb

library preparation

**De Novo approaches**
- OASES (Schultz et al. 2012)
- Trinity (Grabherr et al. 2011)
- Kissplice (Sacomoto et al. 2012)

**Transcripts Quantification using annotations**
- RQuant (Bohnert et al. 2009)
- FluxCapacitor (Montgomery et al. 2010)
- IsoEM (Nicolae et al. 2011)
- eXpress (Roberts et al. 2013)

**Genome-based Transcripts Reconstruction**
- Scripture (Guttman et al. 2010)
- Cufflinks (Trapnell et al. 2010)
- IsoLasso (Li et al. 2011a)
- NSMAP (Xia et al. 2011)
- SLIDE (Li et al. 2011b)
- iReckon (Mezlini et al. 2012)
- FlipFlop
The isoform deconvolution problem

(Xia et al., 2011)
More formally

$e$ exons
$c$ candidate isoforms (up to $2^e - 1$)
$\phi \in \mathbb{R}_+^c$ the vector of abundance of isoforms (unknown!)
$U$ binary matrix:

\[
\begin{pmatrix}
\text{exon}_1 & \cdots & \text{exon}_e & \text{junction}_{1,2} & \cdots & \text{junction}_{e_1,e}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\text{isoform}_1 \\
\text{isoform}_2 \\
\vdots \\
\text{isoform}_c
\end{pmatrix} = \begin{pmatrix}
1 & \cdots & 1 & 1 & \cdots & 1 \\
1 & \cdots & 0 & 1 & \cdots & 0 \\
\vdots & & \cdots & \cdots & & \cdots \\
0 & \cdots & 1 & 0 & \cdots & 0
\end{pmatrix}
\]

$U^T \phi$ the abundance of each exon/junction.

Goal: estimate $\phi$ from the observed reads on each exon/junction
Isoform deconvolution with the Lasso

Estimate $\phi$ sparse by solving:

$$\min_{\phi \in \mathbb{R}^c_+} R(U^T \phi) + \lambda \| \phi \|_1$$

- IsoLasso (Li et al., 2011)
- NSMAP (Xia et al., 2011)
- SLIDE (Li et al., 2011)

Works well BUT computationally challenging to enumerate all candidate isoforms (up to $2^e$) for large genes!
The isoform deconvolution problem

\[
\min_{\phi \in \mathbb{R}^c_+} R(U^\top \phi) + \lambda \| \phi \|_1
\]

can be solved in polynomial time in the number of exon.

Key ideas

1. \(U^\top \phi\) corresponds to a flow on the graph
2. Reformulation as a convex cost flow problem (Mairal and Yu, 2012)
3. Recover isoforms by flow decomposition algorithm

"Feature selection on an exponential number of features in polynomial time"
From isoforms to flows

- Isoforms are paths
- Linear combinations of isoforms are flows
Isoform deconvolution as convex cost flow problem

\[
\min_{\phi \in \mathbb{R}^c_+} R(U^\top \phi) + \lambda \| \phi \|_1
\]
is equivalent to

\[
\min_{f \text{ flow}} R(f) + \lambda f_t
\]
\[ \min_{\phi \in \mathbb{R}^c_+} R(U^\top \phi) + \lambda \| \phi \|_1 \]

- **Cufflink**: *a priori* selection of isoforms (minimum graph cover)
- **IsoLasso**: pre-filtering of candidate isoforms using various heuristics
- **NSMAP, SLIDE**: limit the maximum number of exons
- **FlipFlop**: exact optimization without pre-filtering in polynomial time
Simulated data (hg19, 1137 genes on chr1, 1 million 75 bp single-end reads by transcript levels).
Speed trial

![Graph showing elapsed time vs. number of exons for different exon counts and software tools: IsoLasso, Cufflinks, FlipFlop, NSMAP, SLIDE.](Image)

<table>
<thead>
<tr>
<th>Number of EXONS</th>
<th>2−5 exons</th>
<th>5−10 exons</th>
<th>10−20 exons</th>
<th>20−116 exons</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsoLasso</td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
</tr>
<tr>
<td>Cufflinks</td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
</tr>
<tr>
<td>FlipFlop</td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
</tr>
<tr>
<td>NSMAP</td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
</tr>
<tr>
<td>SLIDE</td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
<td><img src="Graph" alt="" /></td>
</tr>
</tbody>
</table>

JP Vert (ParisTech) Sparse methods in genomics
Outline

1. Mapping DNA breakpoints in cancer genomes
2. Isoform detection from RNA-seq data
3. Inference of gene regulatory networks
Gene expression

Image adapted from: National Human Genome Research Institute.
Gene expression regulation

- Gene regulatory sequences
- Spacer DNA
- Gene regulatory proteins
- General transcription factors
- RNA polymerase
- TATA box
- Upstream
- Promoter
- Start of transcription
Gene regulatory network

Protein 3

mRNA 3

Gene 3

Protein 2

protein complex

Gene 1

Protein 1

mRNA 1

mRNA 2

Gene 2
Gene expression data
Reconstruction of gene regulatory network from expression data
De novo inference

The problem

Given a set of gene expressions, infer the regulations.

How?

- Connect "similar genes": correlation, mutual-information...
- Model-based approaches: dynamic systems, boolean networks, state-space models, Bayesian networks
- Sparse regression: regulators as the smallest set of TF necessary to predict the expression of the target (GENIE, TIGRESS...)
De novo inference

The problem

Given a set of gene expressions, infer the regulations.

How?

- Connect "similar genes": correlation, mutual-information...
- Model-based approaches: dynamic systems, boolean networks, state-space models, Bayesian networks
- Sparse regression: regulators as the smallest set of TF necessary to predict the expression of the target (GENIE, TIGRESS...)
Let $Y \in \mathbb{R}^n$ the expression of a gene, and $X_1, \ldots, X_p \in \mathbb{R}^n$ the expression of all TFs. We look for a model

$$Y = \sum_{i=1}^{p} \beta_i X_i + \text{noise}$$

where $\beta$ is sparse, i.e., only a few $\beta_i$ are non-zero.

We can estimate the sparse regression model from a matrix of expression data.

Non-zero $\beta_i$’s correspond to predicted regulators.
Feature selection with the lasso

\[ \min_{\beta \in \mathbb{R}^p} \| Y - X\beta \|^2 + \lambda \| \beta \|_1 \quad \text{where} \quad \| \beta \|_1 = \sum_{i=1}^{p} |\beta_i| \]

- No explicit solution, but this is just a quadratic program (Tibshirani, 1996; Chen et al., 1998).
- Efficient solution with the LARS (Efron et al., 2004)
- When \( t \) is not too large, the solution will usually be sparse

Geometric interpretation with \( p = 2 \)
For \( t = 1 \) to \( T \) do
- Bootstrap a random sample \( S_t \) from the training set
- Randomly reweight each feature (uniform on \([\alpha, 1]\))
- Select \( L \) features with the Lasso

The score of a feature is the number of times it was selected among the \( T \) repeats (Meinshausen and Bühlmann, 2010).

Rank features (TF-TG interactions) by decreasing area under the score curve
DREAM5: GENIE and TIGRESS ranked 1st and 2nd out of 29 on the in silico challenge
### Table: AUPR, AUROC and p-values obtained by several methods on the *in silico* dataset.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>AUPR</th>
<th>$p_{AUPR}$</th>
<th>AUROC</th>
<th>$p_{AUROC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIGRESS</td>
<td>0.3152</td>
<td>8.01e-139</td>
<td>0.7829</td>
<td>5.43e-60</td>
</tr>
<tr>
<td>GENIE3</td>
<td>0.2915</td>
<td>2.91e-105</td>
<td>0.8155</td>
<td>2.30e-107</td>
</tr>
<tr>
<td>CLR</td>
<td>0.2654</td>
<td>1.82e-73</td>
<td>0.7817</td>
<td>1.41e-58</td>
</tr>
<tr>
<td>Pearson</td>
<td>0.1887</td>
<td>3.71e-13</td>
<td>0.7568</td>
<td>1.44e-32</td>
</tr>
<tr>
<td>ARACNE</td>
<td>0.2758</td>
<td>1.73e-85</td>
<td>0.6715</td>
<td>9.82e-01</td>
</tr>
<tr>
<td>Lasso</td>
<td>0.2079</td>
<td>1.38e-23</td>
<td>0.7280</td>
<td>1.06e-12</td>
</tr>
</tbody>
</table>
Influence of $\alpha$ and scoring method

DREAM5 in silico network.
Conclusions

- Convex sparsity-inducing penalties as a way to incorporate prior knowledge
- Specific implementations for specific problems:
  - greedy dichotomic segmentation for fused lasso
  - fast group Lasso for joint segmentation
  - network flow optimization of lasso over the paths of a graph
- Often, feature selection is consistent (although we pay a price when features are very correlated), stability selection may help
- Numerous applications in bioinformatics and beyond!
Thanks!

Kevin Bleakley (INRIA), Elsa Bernard (ParisTech/Institut Curie), Laurent Jacob (UC Berkeley/CNRS), Julien Mairal (UC Berkeley/INRIA), Anne-Claire Haury, Fantine Mordelet (Duke), Paola Vera-Licona (Institut Curie)