1 body = $10^{14}$ human cells (and 100x more non-human cells)
1 cell = $6 \times 10^9$ ACGT coding for 20,000 genes
Sequencing revolution

Cost per Genome

$100M
$10M
$1M
$100K
$10K
$1K

NIH National Human Genome Research Institute

genome.gov/sequencingcosts
Many various data
A cancer cell
A cancer cell
A cancer cell
Opportunities

- What is your risk of developing a cancer? (*prevention*)
- After diagnosis and treatment, what is the risk of relapse? (*prognosis*)
- What specific treatment will cure your cancer? (*personalized medicine*)
Machine learning formulation
Machine learning formulation
Machine learning formulation
Machine learning formulation
Challenges

- High dimension
- Few samples
- Structured data
- Heterogeneous data
- Prior knowledge
- Fast and scalable implementations
- Interpretable models
Learning with regularization

Learn

\[ f_\beta(x) = \beta^\top x \]

by solving

\[ \min_{\beta \in \mathbb{R}^p} R(f_\beta) + \lambda \Omega(\beta) \]

- \( R(f_\beta) \) empirical risk
- \( \Omega(\beta) \) penalty
1. FlipFlop: fast isoform prediction from RNA-seq data

2. Learning molecular classifiers with network information

3. Kernel bilinear regression for toxicogenomics
1. FlipFlop: fast isoform prediction from RNA-seq data
2. Learning molecular classifiers with network information
3. Kernel bilinear regression for toxicogenomics
Joint work with...

Elsa Bernard (Mines ParisTech / Institut Curie), Laurent Jacob (CNRS / LBBE), Julien Mairal (INRIA)
Alternative splicing: 1 gene = many proteins

In human, 28k genes give 120k known transcripts (Pal et al., 2012)
Opportunities for drug developments...

(Pal et al., 2012)
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

http://rnaseq.uoregon.edu
RNA-seq and alternative splicing

(Costa et al., 2011)
Let a gene with $e$ exons

Suppose there are $c$ candidate isoform ($c$ large, up to $2^e$)

Let $\phi \in \mathbb{R}^c$ the unknown $c$-dimensional vector of abundance

Let $L(\phi)$ quantify whether $\phi$ explains well the observed read counts (e.g., minus log-likelihood)

Find a sparse vector of abundances by solving (e.g., IsoLasso, SLIDE, NSMAP...)

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

Computational problem: Lasso problem with $2^e$ variables
Lasso-based estimation of isoforms

Let a gene with $e$ exons
Suppose there are $c$ candidate isoform ($c$ large, up to $2^e$)
Let $\phi \in \mathbb{R}^c$ the unknown $c$-dimensional vector of abundance
Let $L(\phi)$ quantify whether $\phi$ explains well the observed read counts (e.g., minus log-likelihood)
Find a sparse vector of abundances by solving (e.g., IsoLasso, SLIDE, NSMAP...)

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

Computational problem: Lasso problem with $2^e$ variables
Theorem (Bernard, Mairal, Jacob and V., 2014)

The isoform deconvolution problem

$$\min_{\phi \in \mathbb{R}_+^c} L(\phi) + \lambda \| \phi \|_1$$

can be solved in \textit{polynomial time} in the number of exon.

Key ideas

1. Reformulation as a \textit{convex cost flow problem} (Mairal and Yu, 2012)

2. Recover isoforms by flow decomposition algorithm

"Feature selection on an exponential number of features in polynomial time"
Isoforms are Paths in a Graph
Isoforms are Paths in a Graph
Isoforms are Paths in a Graph
Isoforms are Paths in a Graph
Combinations of isoforms are flows

\[ (a) \text{ Reads at every node corresponding to one isoform.} \]

\[ (b) \text{ Reads at every node after adding another isoform.} \]

- \( L(\phi) \) depends only on the values of the flow on the vertices
- \( \|\phi\|_1 = f_t \)

Therefore,

\[
\min_{\phi \in \mathbb{R}_+^c} L(\phi) + \lambda \|\phi\|_1
\]

is equivalent to

\[
\min_{f \text{ flow}} R(f) + \lambda f_t
\]
Human Simulation: Precision/Recall

hg19, 1137 genes on chr1, 1 million 75 bp single-end reads by transcript levels.

Simulator: [http://alumni.cs.uc.rder.edu/~liw/rnaseqreadsimulator.html](http://alumni.cs.uc.rder.edu/~liw/rnaseqreadsimulator.html)
Performance increases with read length

![Graph showing performance increases with read length.](image)

**Legend:**
- Isolasso
- Cufflinks
- FlipFlop
- N5MAP
- 1 transcripts
- 2 transcripts
- 3–4 transcripts
- 5–7 transcripts
- 8–43 transcripts

**Figure 2:** Illustration of the graph construction for a gene with 4 exons. The graph is constructed by connecting consecutive vertices along the path, and each vertex represents a junction between exons. The length of each vertex is proportional to the length of the corresponding exon. The graph is designed to represent the flow of information through the gene, with the source node (s) injecting flow into the network and the sink node (t) receiving flow. The network is designed to capture isoform candidates, such that (4) holds. Moreover, there exists linear-time algorithms to solve this problem.

**Flow Interpretation of Isoforms:**

We now have all the tools in hand to turn (3) into a flow problem. Given a flow $f$ on a graph $G$, let us define the amount of flow incoming to a node $v$ by following Mairal and Yu (2012). Given a flow $f$ on a graph $G$, we can define the amount of flow incoming to a node $v$ by the following equation:

$$
\sum_{u \in V} f(u,v) - \sum_{v \in V} f(v,u) = \text{inflow of } v
$$

This construction corresponds to sending an amount of flow equal to the abundance of the isoform with abundance $s$, from the source $s$ to the sink $t$. Such conservation property leads to a flow interpretation for isoforms. For example, let us consider a gene with 4 exons, where the length of exon 3 is smaller than the read length. When all exons are bigger than the read length, the resulting graph in (b) is the graph in (a). When the length of exon 3 is smaller than the read length, the resulting graph in (b) is the graph in (a) when all exons are bigger than the read length.
Performance increases with coverage
Extension to paired-end reads OK.
Speed trial

- **Graph**: Elapsed time (s) vs. Number of EXONS
  - Axes: X-axis = Number of EXONS, Y-axis = Elapsed time (s)
  - Legend: Flipflop, NSMAP

- **Table**: CPU time (ms) by gene
  - Columns: 2–5 exons, 5–10 exons, 10–20 exons, 20–116 exons
  - Genes: IsoLasso, Cufflinks, FlipFlop, NSMAP, SLIDE
  - Colors: green, purple, blue, red, cyan

- **Summary**: The graph shows the elapsed time for different gene sizes, with Flipflop and NSMAP plotted. The table provides CPU time for each gene classification.
FlipFlop summary

- Fast method for exact Lasso-based isoform detection and quantification
- [http://cbio.mines-paristech.fr/flipflop](http://cbio.mines-paristech.fr/flipflop)
- Available as an R package

```r
> source("http://bioconductor.org/biocLite.R")
> biocLite("flipflop")
```


- Ongoing: extension to multiple samples and differential analysis
Outline

1. FlipFlop: fast isoform prediction from RNA-seq data
2. Learning molecular classifiers with network information
3. Kernel bilinear regression for toxicogenomics
Joint work with...

Franck Rapaport, Emmanuel Barillot, Andrei Zinovyev, Anne-Claire Haury, Laurent Jacob, Guillaume Obozinski
Breast cancer prognosis
The idea

- We look for a **limited set** of genes that are sufficient for prediction.
- Selected genes should inform us about the underlying biology.
Lack of stability of signatures

Haury et al. (2011)
Gene networks and expression data

**Motivation**

- Basic biological functions usually involve the **coordinated action of several proteins**:
  - Formation of **protein complexes**
  - Activation of metabolic, signalling or regulatory **pathways**
- Many pathways and protein-protein interactions are **already known**
- **Hypothesis**: the weights of the classifier should be “coherent” with respect to this **prior knowledge**

![Graphical representation of gene networks and expression data](image)
Graph based penalty

\[ f_\beta(x) = \beta^\top x \quad \min_{\beta} R(f_\beta) + \lambda \Omega(\beta) \]

Prior hypothesis

Genes near each other on the graph should have similar weights.

An idea (Rapaport et al., 2007)

\[ \Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2, \]

\[ \min_{\beta \in \mathbb{R}^p} R(f_\beta) + \lambda \sum_{i \sim j} (\beta_i - \beta_j)^2. \]
Graph based penalty

\[ f_\beta(x) = \beta^T x \quad \min_{\beta} R(f_\beta) + \lambda \Omega(\beta) \]

Prior hypothesis

Genes near each other on the graph should have similar weights.

An idea (Rapaport et al., 2007)

\[ \Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2, \]

\[ \min_{\beta \in \mathbb{R}^p} R(f_\beta) + \lambda \sum_{i \sim j} (\beta_i - \beta_j)^2. \]

Fig. 5. Work was supported by the grant ACI-IMPBIO-2004-47 of the French Ministry for Research and New Technologies.
Spectral penalty as a kernel

Theorem

The function $f(x) = \beta^T x$ where $\beta$ is solution of

$$
\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell \left( \beta^T x_i, y_i \right) + \lambda \sum_{i \sim j} (\beta_i - \beta_j)^2
$$

is equal to $g(x) = \gamma^T \Phi(x)$ where $\gamma$ is solution of

$$
\min_{\gamma \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \ell \left( \gamma^T \Phi(x_i), y_i \right) + \lambda \gamma^T \gamma,
$$

and where

$$
\Phi(x)^T \Phi(x') = x^T K_G x'
$$

for $K_G = L^*$, the pseudo-inverse of the graph Laplacian.
Graph Laplacian

Definition

The Laplacian of the graph is the matrix \( L = D - A \).

\[
L = D - A = \begin{pmatrix}
1 & 0 & -1 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 \\
-1 & -1 & 3 & -1 & 0 \\
0 & 0 & -1 & 2 & -1 \\
0 & 0 & 0 & 1 & 1
\end{pmatrix}
\]
Pseudo-inverse of the Laplacian

$L^* = \begin{pmatrix}
0.88 & -0.12 & 0.08 & -0.32 & -0.52 \\
-0.12 & 0.88 & 0.08 & -0.32 & -0.52 \\
0.08 & 0.08 & 0.28 & -0.12 & -0.32 \\
-0.32 & -0.32 & -0.12 & 0.48 & 0.28 \\
-0.52 & -0.52 & -0.32 & 0.28 & 1.08 \\
\end{pmatrix}$
Other penalties with kernels

\[ \Phi(x)^\top \Phi(x') = x^\top K_G x' \]

with:

- \( K_G = (c + L)^{-1} \) leads to

\[ \Omega(\beta) = c \sum_{i=1}^{p} \beta_i^2 + \sum_{i \sim j} (\beta_i - \beta_j)^2. \]

- The diffusion kernel:

\[ K_G = \exp_M(-2tL). \]

penalizes high frequencies of \( \beta \) in the Fourier domain.
Other penalties without kernels

- Gene selection + Piecewise constant on the graph

\[ \Omega(\beta) = \sum_{i \sim j} |\beta_i - \beta_j| + \sum_{i=1}^{p} |\beta_i| \]

- Gene selection + smooth on the graph

\[ \Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2 + \sum_{i=1}^{p} |\beta_i| \]
Example: classification of DNA copy number profiles

Aggressive (left) vs non-aggressive (right) melanoma
Fused lasso solution (Rapaport et al., 2008)

\[ \Omega(\beta) = \sum_{i \sim j} |\beta_i - \beta_j| + \sum_{i=1}^{p} |\beta_i| \]
Graph-based structured feature selection

\[ \Omega_1(\beta) = \sum_{i \sim j} \sqrt{\beta_i^2 + \beta_j^2}, \quad \text{(Jenatton et al., 2009)} \]

\[ \Omega_2(\beta) = \sup_{\alpha \in \mathbb{R}^p : \forall i \sim j, \|\alpha_i^2 + \alpha_j^2\| \leq 1} \alpha^\top \beta. \quad \text{(Jacob et al., 2008)} \]
Lasso signature (accuracy 0.61)

Breast cancer prognosis
Graph Lasso signature (accuracy 0.64)

Breast cancer prognosis
Disjoint feature selection

\[ W = (w_i)_{i \in V} \in \mathbb{R}^{p \times V} \]

\[ \Omega(W) = \min_{-H \leq W \leq H} \sum_{i \sim j} K_{ij} \left| h_i^\top h_j \right| \]

(Vervier et al, 2014)
Example: multiclass classification of MS spectra

(Vervier et al, 2013, unpublished)
Outline

1. FlipFlop: fast isoform prediction from RNA-seq data
2. Learning molecular classifiers with network information
3. Kernel bilinear regression for toxicogenomics
Elsa Bernard, Erwan Scornet, Yunlong Jiao, Véronique Stoven, Thomas Walter
Pharmacogenomics / Toxicogenomics

Patients with same condition

DNA Profiling

Good responders

Bad side effects

No Responders
Genotypes from the 1000 genome project
RNASeq from the Geuvadis project
Bilinear regression

- Cell line $X$, chemical $Y$, toxicity $Z$.
- Bilinear regression model:

$$Z = f(X, Y) + b(Y) + \epsilon,$$

- Estimation by kernel ridge regression:

$$\min_{f \in \mathcal{H}, b \in \mathbb{R}^p} \sum_{i=1}^{n} \sum_{j=1}^{p} (f(x_i, y_j) + b_j - z_{ij})^2 + \lambda \|f\|^2,$$
**Theorem 1.** Let \( Z \in \mathbb{R}^{n \times p} \) be the response matrix, and \( K_X \in \mathbb{R}^{n \times n} \) and \( K_Y \in \mathbb{R}^{p \times p} \) be the kernel Gram matrices of the \( n \) cell lines and \( p \) chemicals, with respective eigenvalue decompositions \( K_X = U_X D_X U_X^\top \) and \( K_Y = U_Y D_Y U_Y^\top \). Let \( \gamma = U_X^\top 1_n \) and \( S \in \mathbb{R}^{n \times p} \) be defined by \( S_{ij} = 1/ \left( \lambda + D_X^i D_Y^j \right) \), where \( D_X^i \) (resp. \( D_Y^i \)) denotes the \( i \)-th diagonal term of \( D_X \) (resp. \( D_Y \)). Then the solution \( (f^*, b^*) \) of (2) is given by

\[
 b^* = U_Y \text{Diag} \left( S^\top \gamma^2 \right)^{-1} \left( S^\top \circ \left( U_Y^\top Z^\top U_X \right) \right) \gamma \tag{3}
\]

and

\[
 \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, \quad f^*(x, y) = \sum_{i=1}^n \sum_{j=1}^p \alpha_{i,j}^* K_X(x_i, x) K_Y(y_i, y), \tag{4}
\]

where

\[
 \alpha^* = U_X \left( S \circ \left( U_X^\top \left( Z - 1_n b^*^\top \right) U_Y \right) \right) U_Y^\top \tag{5}
\]
Kernel Trick

cell line descriptors

Kcell

kernelized

Kdrug

drug descriptors

jeudi 7 novembre 13
Kernel Trick

- **cell line descriptors**
- **drug descriptors**

Kernelized

**Kcell**

**Kdrug**

Kernel bilinear regression

\[ \hat{f} \]
Kernel Trick

- cell line descriptors
- drug descriptors
- Kcell
- kernelized
- Kdrug
- kernel bilinear regression
- $\hat{f}$

Kernel choice?
- descriptors
- data integration
- missing data
Kernel choice

1. $K_{\text{cell}}$:
   - 29 cell line kernels tested
   - 1 kernel that integrate all information
   - deal with missing data

2. $K_{\text{drug}}$:
   - 48 drug kernels tested
   - multi-task kernels
Kernel choice

1. $K_{\text{cell}}$ :
   \[ \rightarrow \text{29 cell line kernels tested} \]
   \[ \rightarrow \text{1 kernel that integrate all information} \]
   \[ \rightarrow \text{deal with missing data} \]

2. $K_{\text{drug}}$ :
   \[ \rightarrow \text{48 drug kernels tested} \]
   \[ \rightarrow \text{multi-task kernels} \]
Cell line data integration

**Covariates**
- linear kernel

**SNPs**
- 10 gaussian kernels

**RNA-seq**
- 10 gaussian kernels
Cell line data integration

**Covariates**
- linear kernel

**SNPs**
- 10 gaussian kernels

**RNA-seq**
- 10 gaussian kernels

**Integrated kernel**
Multi-task drug kernels

- Dirac
- Multi-Task
- Feature-based
- Empirical
- Integrated

independent regression for each drug
Multi-task drug kernels

1. Dirac
2. Multi-Task
3. Feature-based
4. Empirical
5. Integrated

sharing information across drugs
Multi-task drug kernels

1. Dirac
2. Multi-Task
3. Feature-based
4. Empirical
5. Integrated

Linear kernel and 10 gaussian kernels based on features:

- CDK (160 descriptors) and SIRMS (9272 descriptors)
- Graph kernel for molecules (2D walk kernel)
- Fingerprint of 2D substructures (881 descriptors)
- Ability to bind human proteins (1554 descriptors)
Multi-task drug kernels

1. Dirac
2. Multi-Task
3. Feature-based
4. **Empirical**
5. Integrated
Multi-task drug kernels

1. Dirac
2. Multi-Task
3. Feature-based
4. Empirical
5. Integrated

Integrated kernel:

\[ K_{int} = \sum_i K_i \]

- Combine all information on drugs
29x48 kernel combinations: CV results

Value

0 150

Color Key and Histogram

integrated and covariates kernels
29x48 kernel combinations: CV results

Color Key
and Histogram

Count

Value

CI

sightly multi-task
on drugs

covariates kernel
on cell lines

jeudi 7 novembre 13

Kmultitask11
Ksubstructure.txt
Kchemcpp.txt
Kmultitask1
KsirmsRbf1.txt
KcdkRbf1.txt
KpredtargetRbf1.txt
KsirmsRbf2.txt
Kmultitask3
Kmultitask2
KsirmsRbf3.txt
KpredtargetRbf2.txt
KcdkRbf2.txt
KpredtargetRbf3.txt
KcdkRbf3.txt
Kmultitask4
Kmultitask5
KsirmsRbf3.txt
KpredtargetRbf4.txt
KpredtargetRbf5.txt
KcdkRbf4.txt
KcdkRbf6.txt
KcdkRbf5.txt
KcdkRbf7.txt
KsirmsRbf6.txt
KsirmsRbf7.txt
Kmultitask6
Kmultitask8
Kmultitask9
KsirmsRbf4.txt
KsirmsRbf5.txt
KpredtargetMean
KcdkMean
KsirmsMean
Kcovariates.txt
KcovariatesPopulation.txt
KcovariatesBatch.txt
KrnaseqMean
KrnaseqRbf4.txt
KrnaseqRbf9.txt
KrnaseqRbf10.txt
KrnaseqRbf8.txt
KrnaseqRbf5.txt
KrnaseqRbf7.txt
KrnaseqRbf6.txt
Kdirac
Kuniform
KsnpMean
KsnpRbf1.txt
KsnpRbf3.txt
KsnpRbf2.txt
KsnpRbf4.txt
KsnpRbf5.txt
KsnpRbf8.txt
KsnpRbf7.txt
KsnpRbf6.txt
Kernel on cell lines: CV results

Mean CI for cell line kernels

integrated kernel

batch effect

jeudi 7 novembre 13
Mean CI for chemicals kernels
Mean CI for chemicals kernels

Empirical kernel on drugs

Integrated kernel on cell lines

vendredi 8 novembre 13
Many new problems and lots of data in computational genomics
Computational constraints $\rightarrow$ fast sparse models (FlipFlop)
Small $n$ large $p$ $\rightarrow$ regularized models with prior knowledge
Heterogeneous data integration $\rightarrow$ kernel methods
Personalized medicine promising but difficult!
Thanks

Alexandre d’Aspremont, Emmanuel Barillot, Anne-Claire Haury, Laurent Jacob, Pierre Mahé, Julien Mairal, Guillaume Obozinski, Franck Rapaport, Jean-Baptiste Veyrieras, Andrei Zynoviev
... and all CBIO@Mines