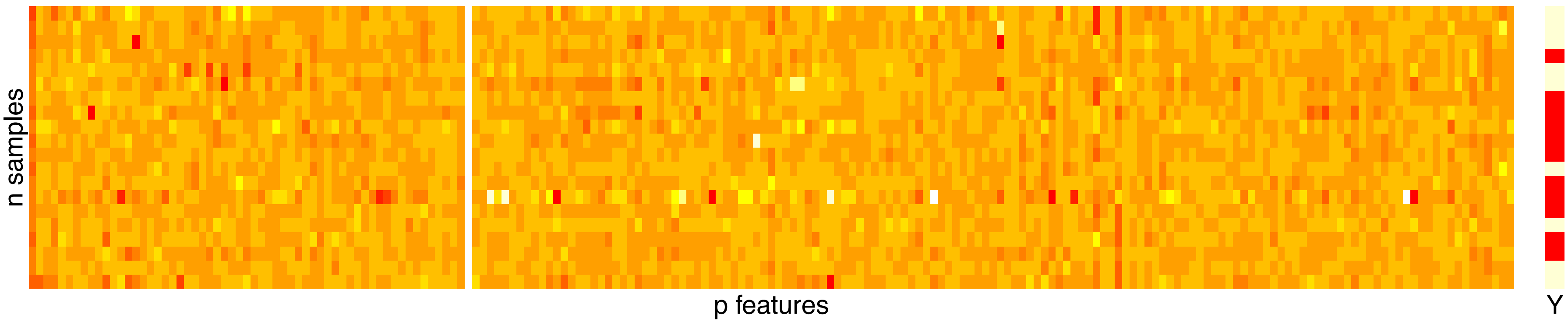
Learning in high dimension

Jean-Philippe Vert

The « n << p » problem



n = 1E2 ~ 1E4
(patients)

p = 1E4 ~ 1E7
(genes, mutations, copy numbers, …)

How to learn with n<<p?

1. **Simplify data: pairwise comparisons**

2. **Add prior knowledge: structured feature selection**

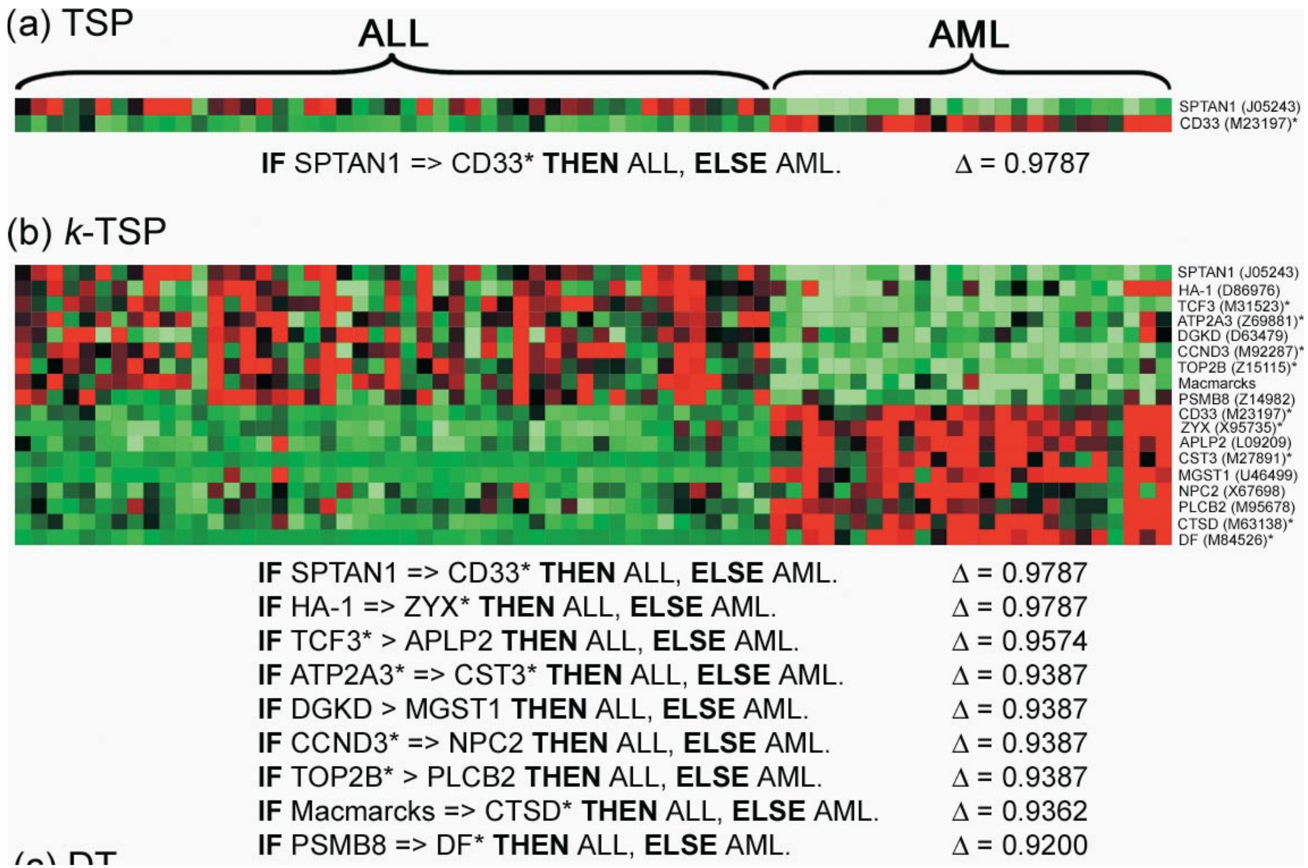
#my2cents

How to learn with n<<p?

1. **Simplify data: pairwise comparisons**

2. **Add prior knowledge: structured feature selection**

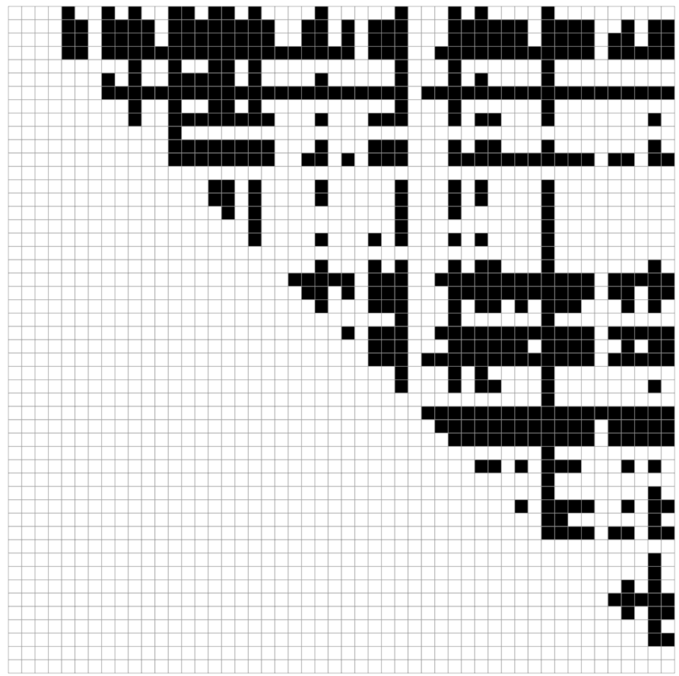
Top Scoring Pairs (TSP)



(Geman et al., 2004; Tan et al., 2005; Leek, 2009;…)

Generalization of TSP

One sample x
p features

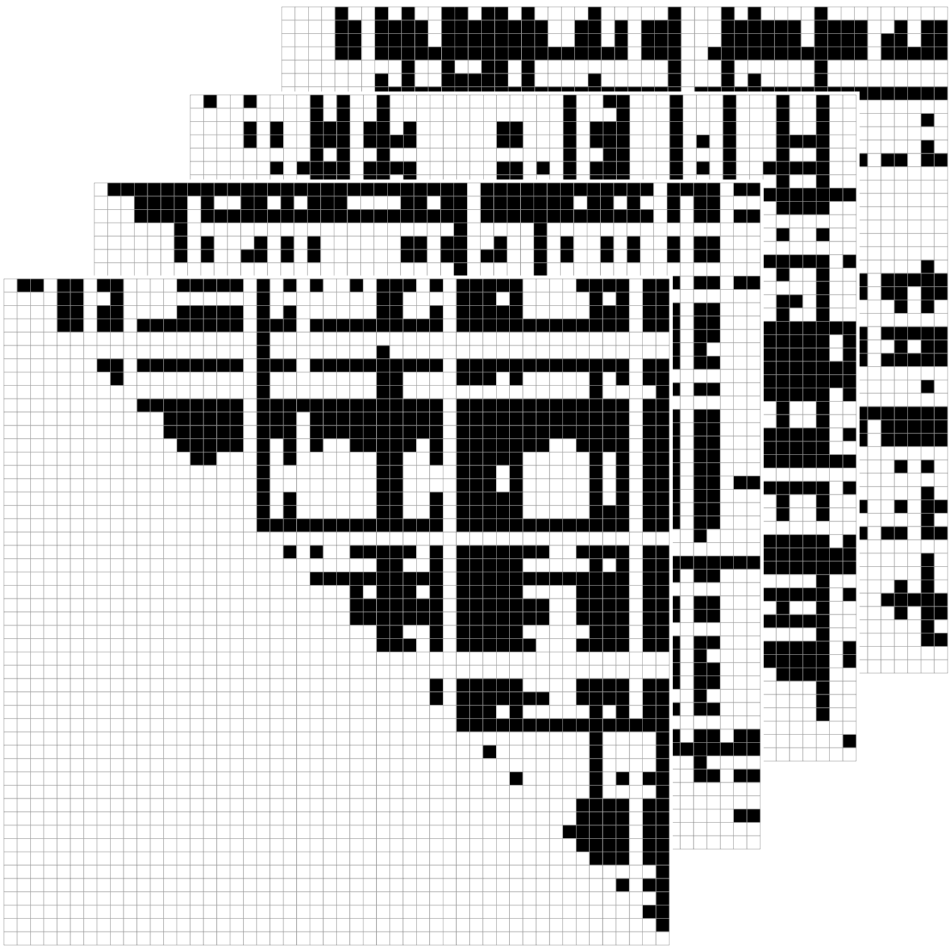


Mapping f(x)
p(p-1)/2 bits

Select features
- **TSP**
- **k-TSP**
- **…**

Linear model
- **logistic regression**
- **ridge regression**
- **SVM**
- **…**

Practical problem

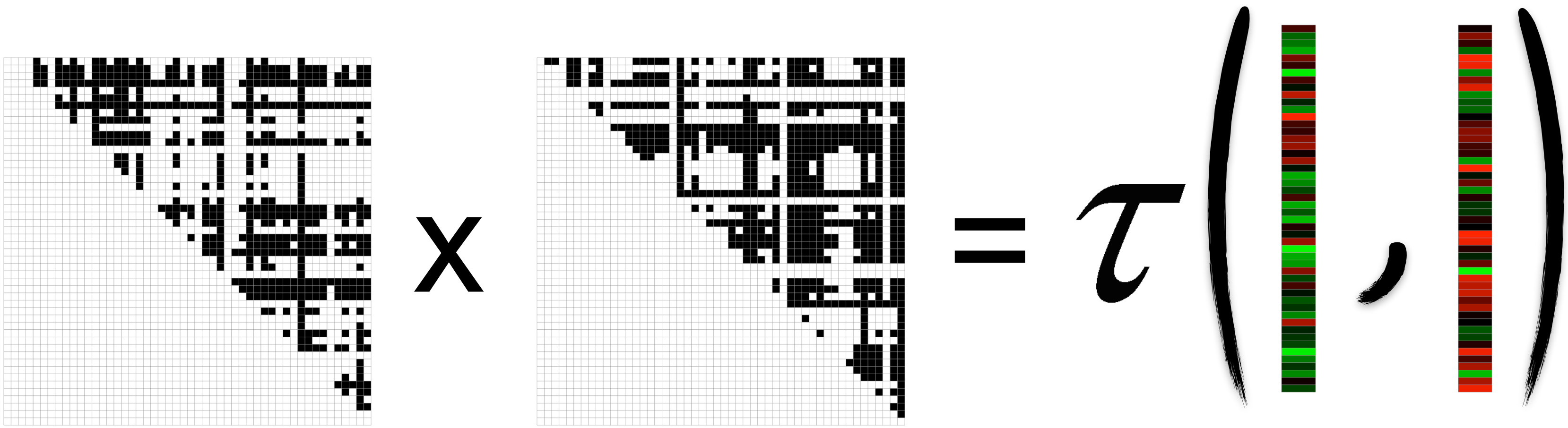


Storing O(p^2) bits per sample

Training a linear model in O(p^2) dimensions

(Jiao and V., 2015)

A trick



O(p^2) O(p log(p))

+kernel trick = we can train linear models efficiently

Experiment

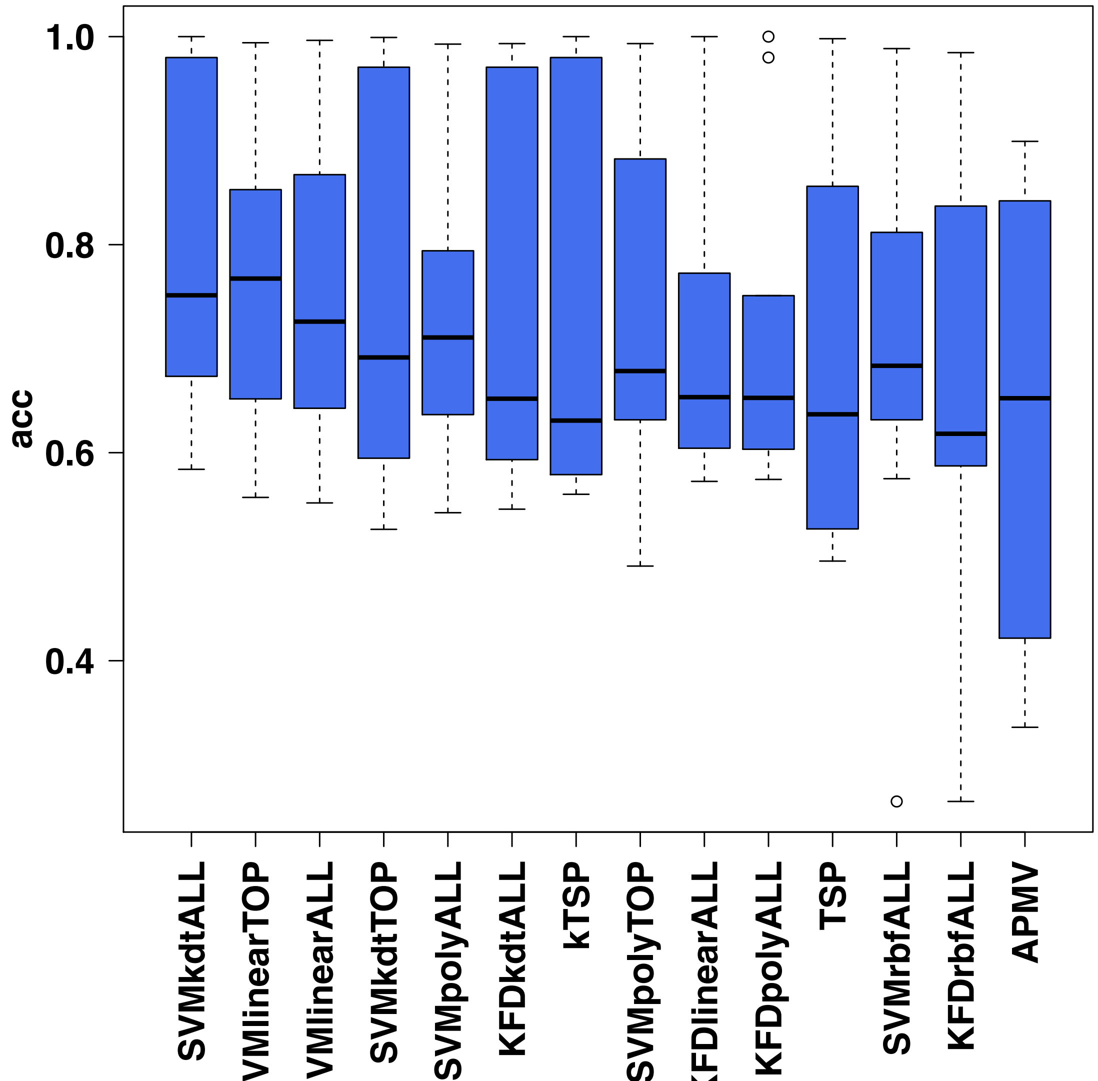
Datasets

Dataset	No. of features	No. of samples (training/test)	
		C_1	C_2
Breast Cancer 1	23624	44/7 (Non-relapse)	32/12 (Relapse)
Breast Cancer 2	22283	142 (Non-relapse)	56 (Relapse)
Breast Cancer 3	22283	71 (Poor Prognosis)	138 (Good Prognosis)
Colon Tumor	2000	40 (Tumor)	22 (Normal)
Lung Cancer 1	7129	24 (Poor Prognosis)	62 (Good Prognosis)
Lung Cancer 2	12533	16/134 (ADCA)	16/15 (MPM)
Medulloblastoma	7129	39 (Failure)	21 (Survivor)
Ovarian Cancer	15154	162 (Cancer)	91 (Normal)
Prostate Cancer 1	12600	50/9 (Normal)	52/25 (Tumor)
Prostate Cancer 2	12600	13 (Non-relapse)	8 (Relapse)

Methods

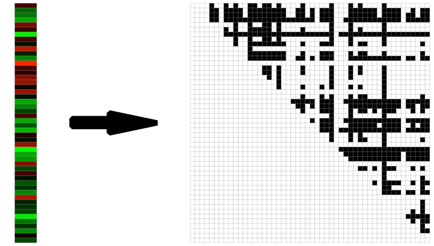
- Kernel machines Support Vector Machines (SVM) and Kernel Fisher Discriminant (KFD) with Kendall kernel, linear kernel, Gaussian RBF kernel, polynomial kernel.
- Top Scoring Pairs (TSP) classifiers [Tan et al., 2005].
- Hybrid scheme of SVM + TSP feature selection algorithm.

Results



	Average
SVMkdtALL	79.39
SVMlinearTOP	77.16
SVMlinearALL	76.09
SVMkdtTOP	75.5
SVMpolyALL	74.54
KFDkdtALL	74.33
kTSP	74.03
SVMpolyTOP	73.99
KFDlinearALL	71.81
KFDpolyALL	71.39
TSP	69.71
SVMrbfALL	69.31
KFDrbfALL	66.39
APMV	61.91

Summary



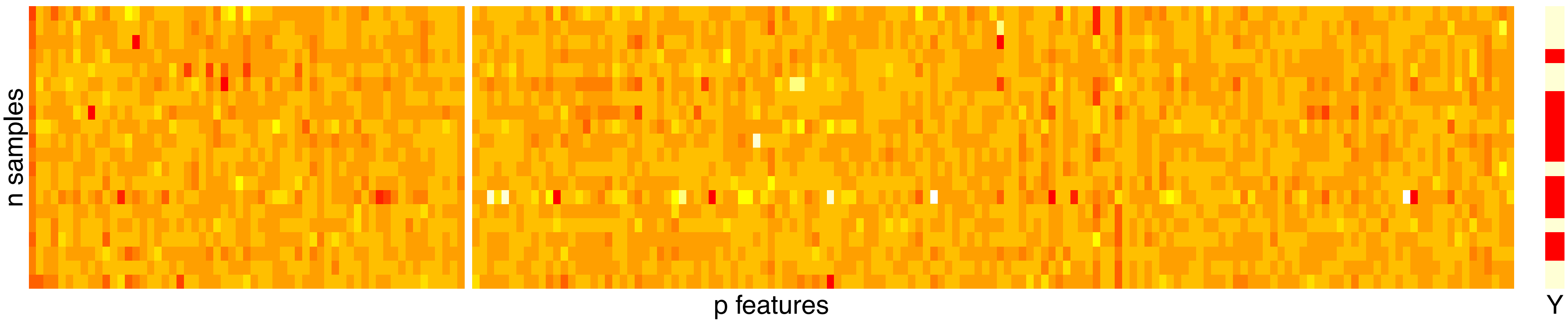
- Robust representation as O(p^2) bits
- Computationally efficient (Kendall kernel)
- Good accuracy
- Extension to missing values OK
- Extension to « fuzzy comparison » OK
- Open questions:
  - robustness across technologies (Patil et al., 2015) ?
  - correction for batch / structure?

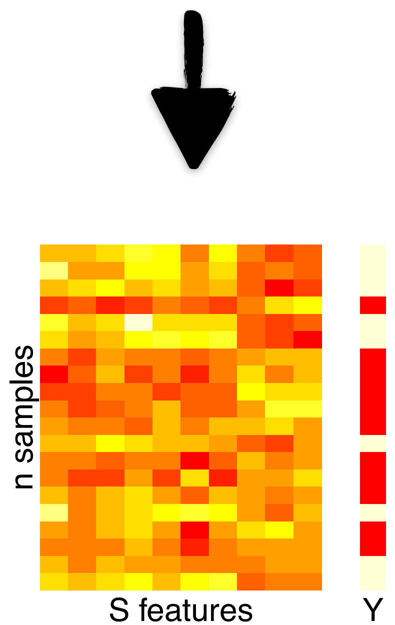
How to learn with n<<p?

1. **Simplify data: pairwise comparisons**

2. **Add prior knowledge: structured feature selection**

Feature Selection





« Molecular signature »

Also relevant for

- *isoform identification from RNA-seq data (IsoLasso, FlipFlop etc…)*
- *gene network inference (GENIE3, TIGRESS, etc…)*

Early disappointments…

Gene expression profiling predicts clinical outcome of breast cancer

Laura J. van 't Veer*†, Hongyue Dai†‡, Marc J. van de Vijver*†, Yudong D. He‡, Augustinus A. M. Hart*, Mao Mao‡, Hans L. Peterse*, Karin van der Kooy*, Matthew J. Marton‡, Anke T. Witteveen*, George J. Schreiber‡, Ron M. Kerkhoven*, Chris Roberts‡, Peter S. Linsley‡, René Bernards* & Stephen H. Friend‡

70 genes (Nature, 2002)

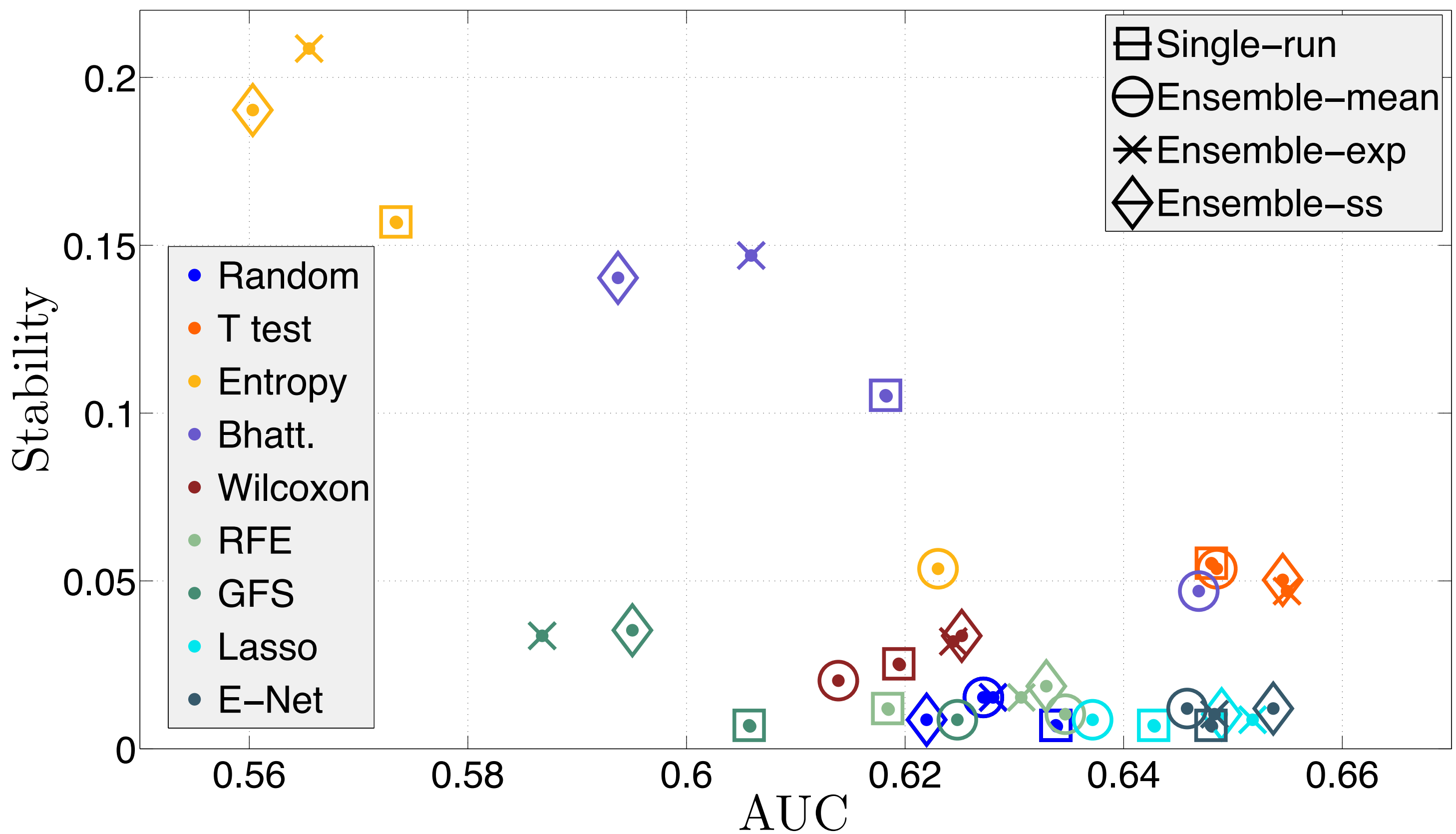
Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer

Yixin Wang, Jan G M Klijn, Yi Zhang, Anieta M Sieuwerts, Maxime P Look, Fei Yang, Dmitri Talantov, Mieke Timmermans, Marion E Meijer-van Gelder, Jack Yu, Tim Jatkoe, Els M J J Berns, David Atkins, John A Foekens

76 genes (Lancet, 2005)

3 genes in common

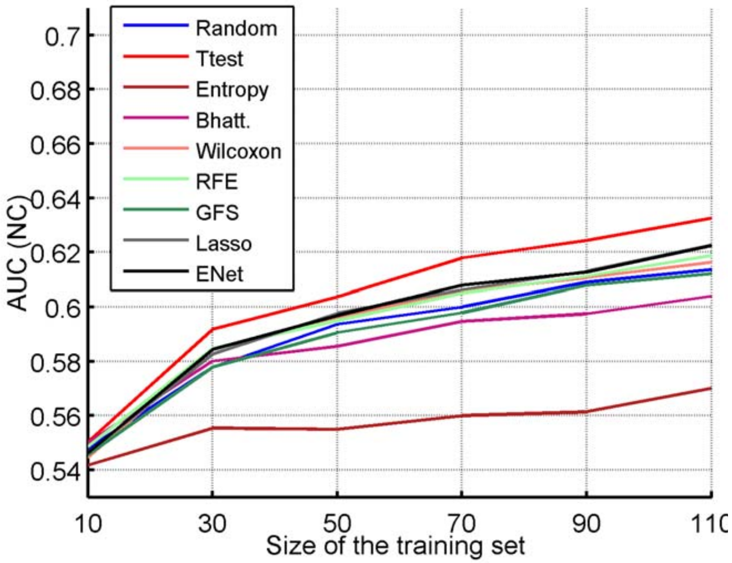
Not because of feature selection method

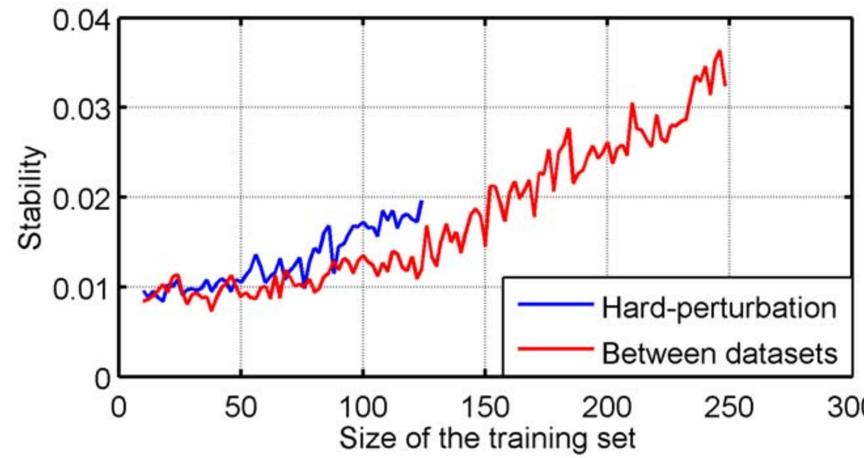


(Haury et al., 2011)

What's wrong?

Increasing n helps





Can we try to « decrease p »?

Add prior knowledge,

Structured feature selection

Sparsity with the LASSO

- Linear model

f(x) = w1 x1 + w2 x2 + … + wP xP

- Sparse when wK=0 for many K’s

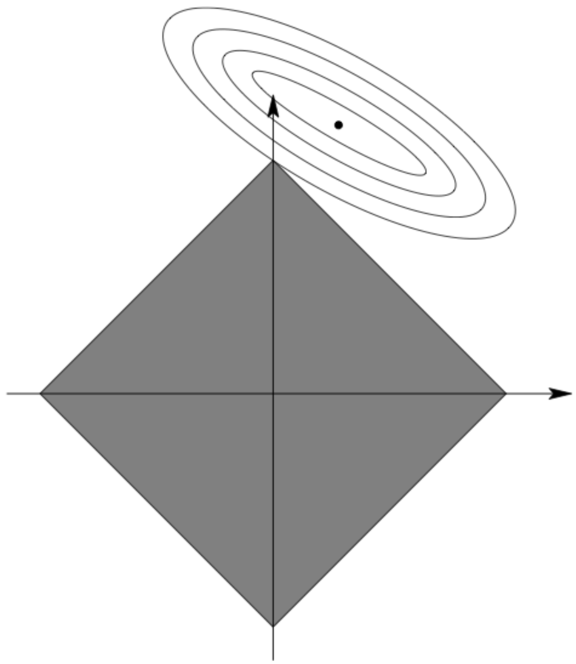
- Learn a sparse model by

minimize Error(w)

such that

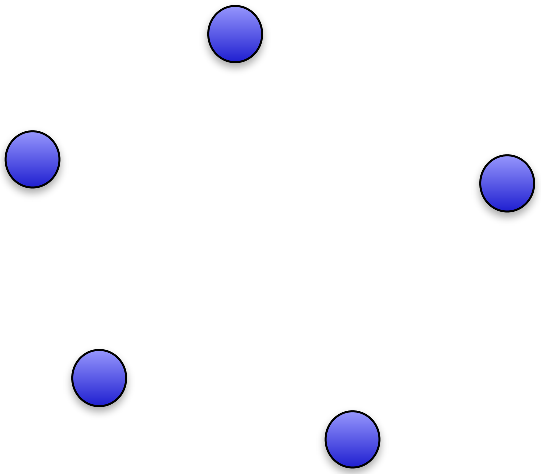
w is in the grey box O

- O is convex -> **efficient algorith**
- O has edges -> **sparsity**



Structured sparsity with atomic norms

1) Choose a set of ATOMS

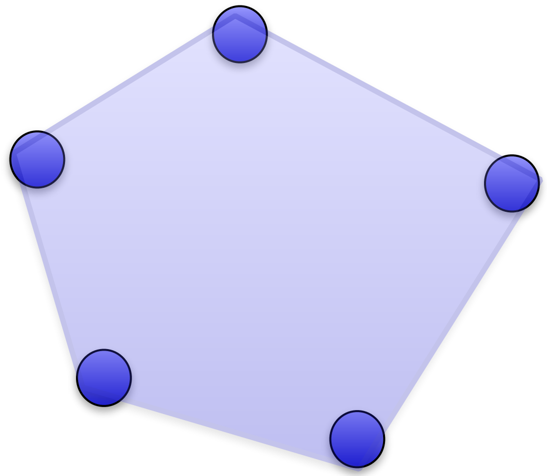


(Chandrasekaran et al., 2012, …)

Structured sparsity with atomic norms

1) Choose a set of ATOMS

2) Take the convex hull O



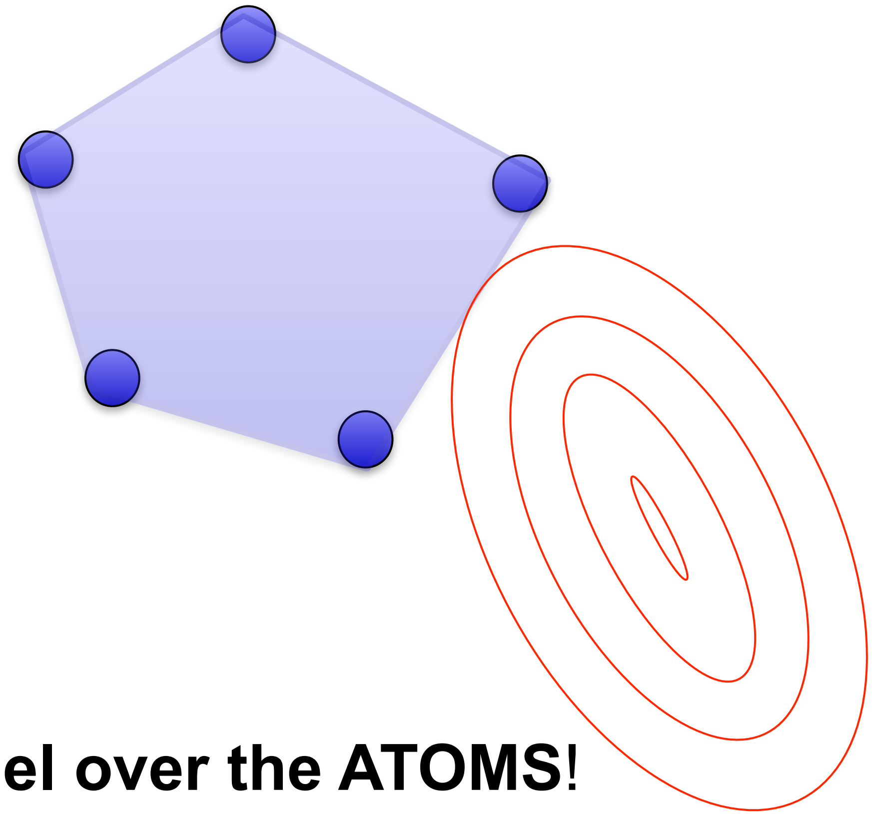
(Chandrasekaran et al., 2012, …)

Structured sparsity with atomic norms

1) Choose a set of ATOMS

2) Take the convex hull

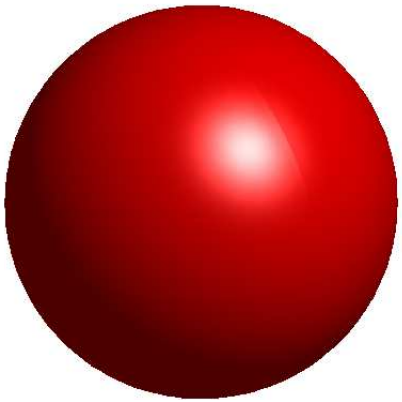
3) Minimize Error(w)
such that
w is in the convex hull



The solution is a **sparse model over the ATOMS**!

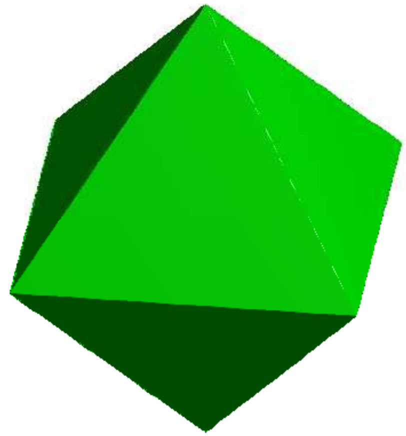
(Chandrasekaran et al., 2012, …)

Quizz: where are the atoms?



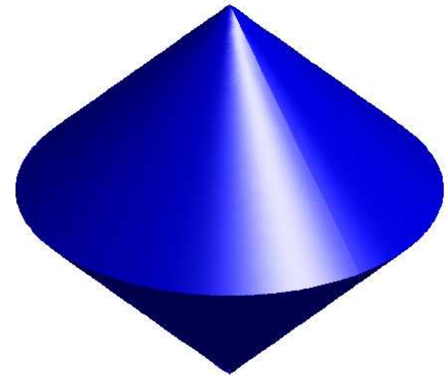
$\|w\|_2$

Ridge



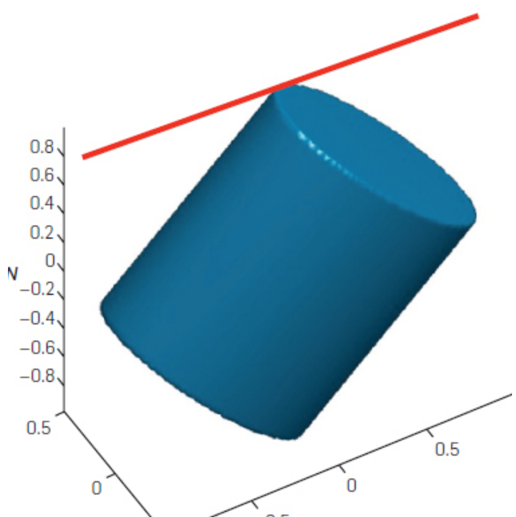
$\|w\|_1$

Lasso



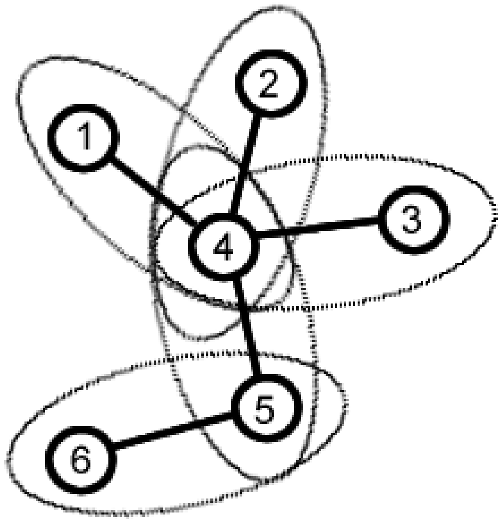
$\sqrt{w_1^2 + w_2^2} + |w_3|$

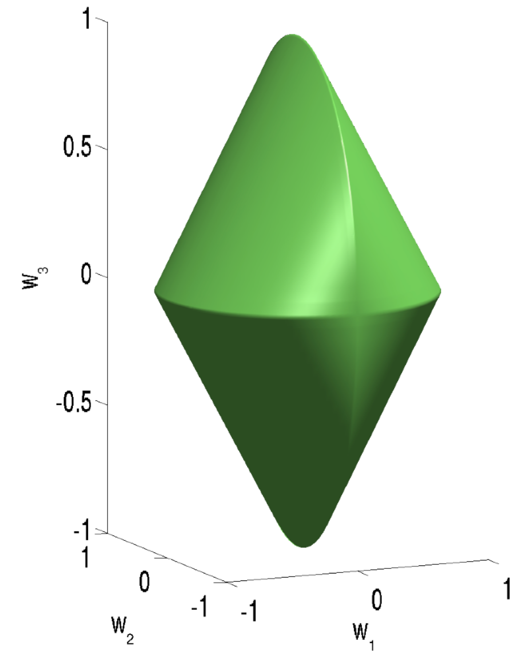
Group Lasso



Trace norm

Graph Lasso





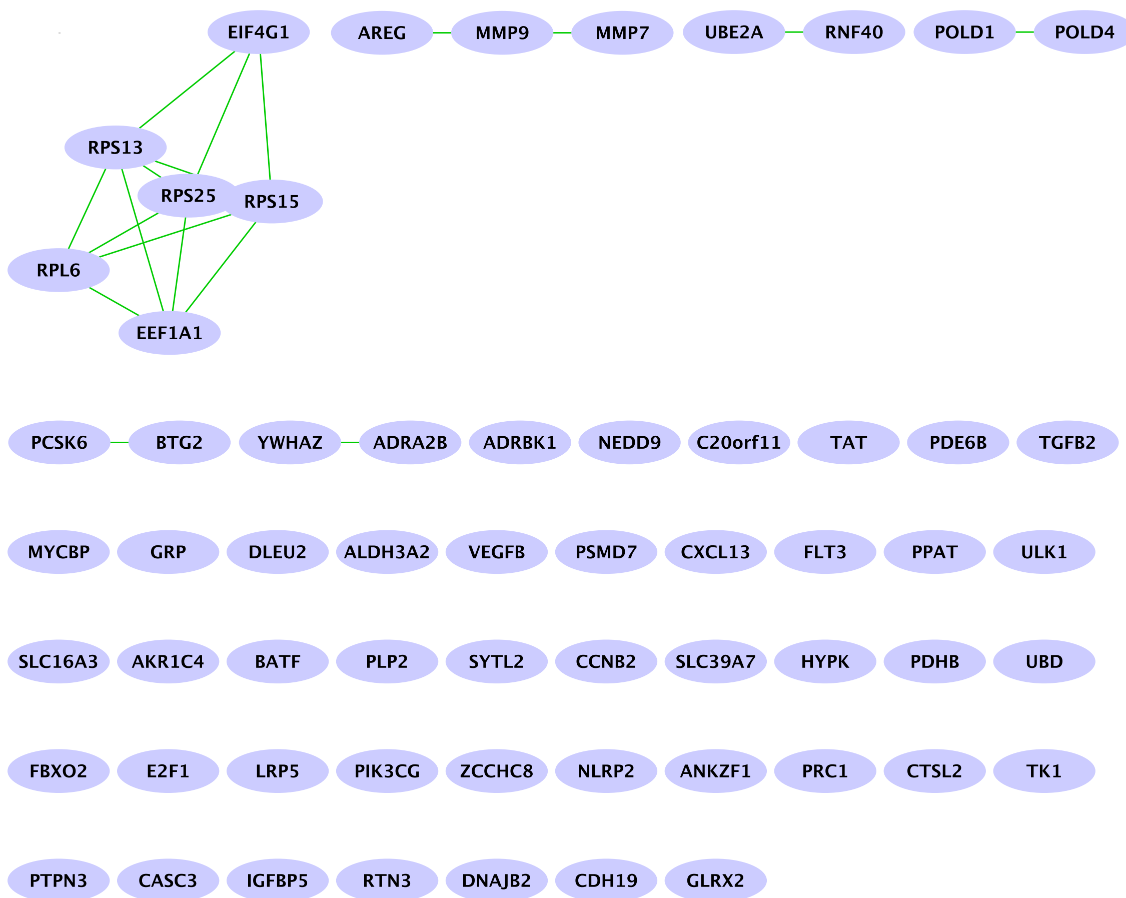

L. Jacob


G. Obozinski

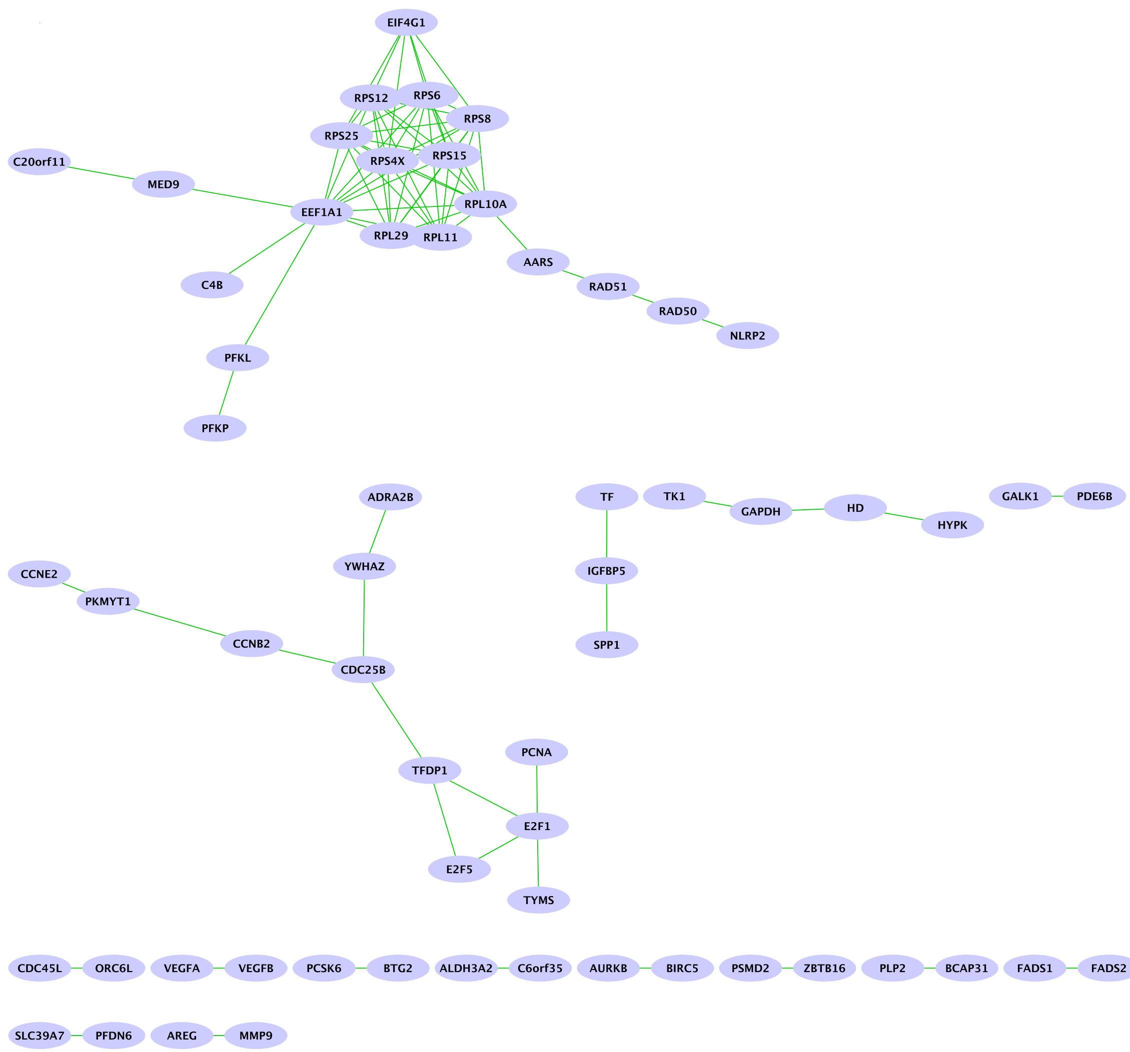
To select features that tend to be connected over a given network

(Jacob et al., 2009)

Breast cancer prognosis signature with Lasso (accuracy=61%)



Breast cancer prognosis signature with Graph Lasso (accuracy=64%)



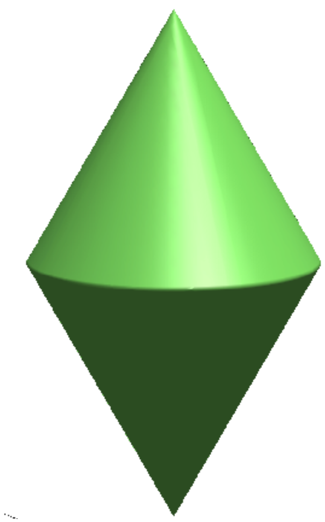
Joint isoform detection from multiple RNA-Seq samples

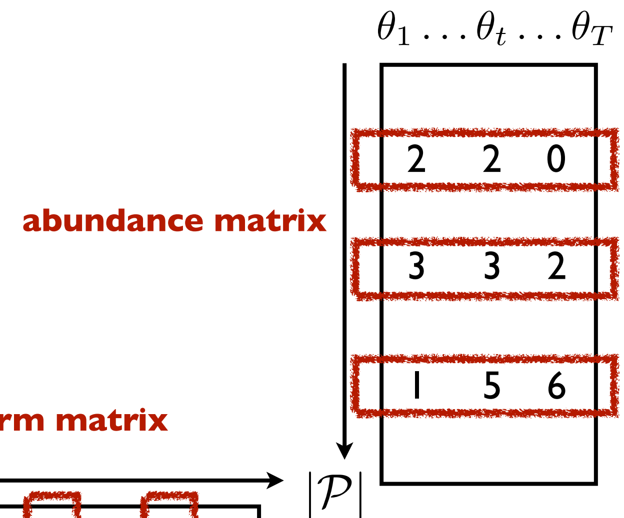

Elsa Bernard

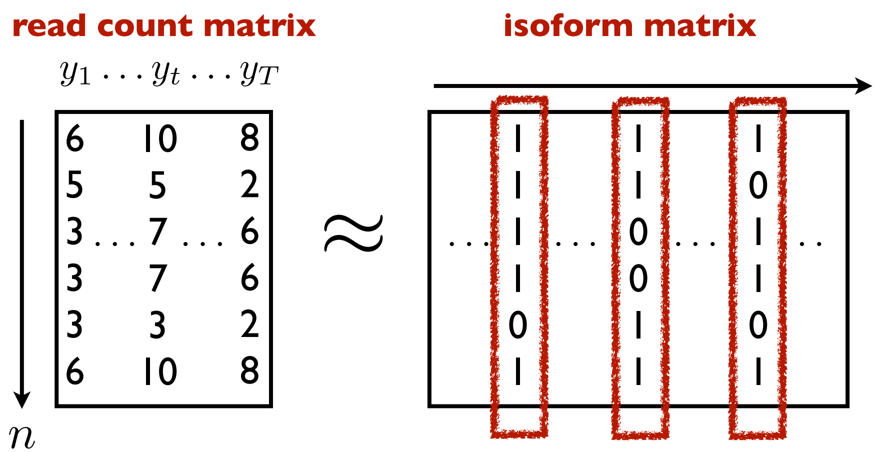

Laurent Jacob


Julien Mairal


Eric Viara







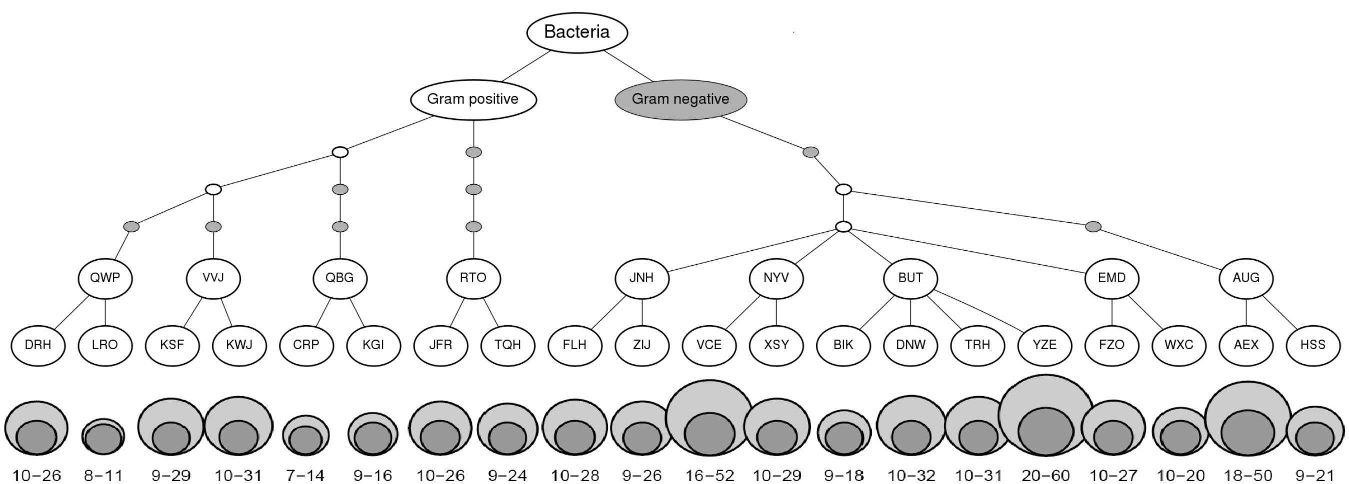

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("flipflop")
```

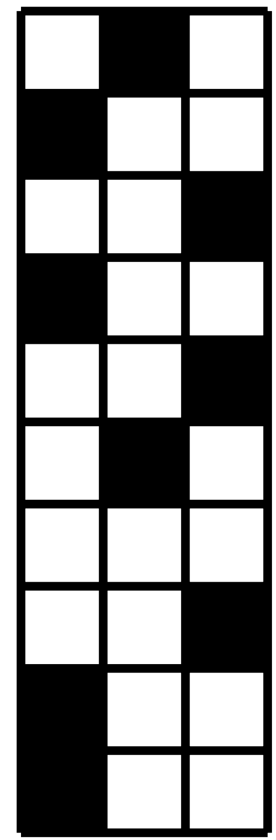
(Bernard et al., 2015)

Learning sparse models with disjoint support ?

Motivation

- Multiclass or multi-task classification problems
- Eg: cascade of classifiers

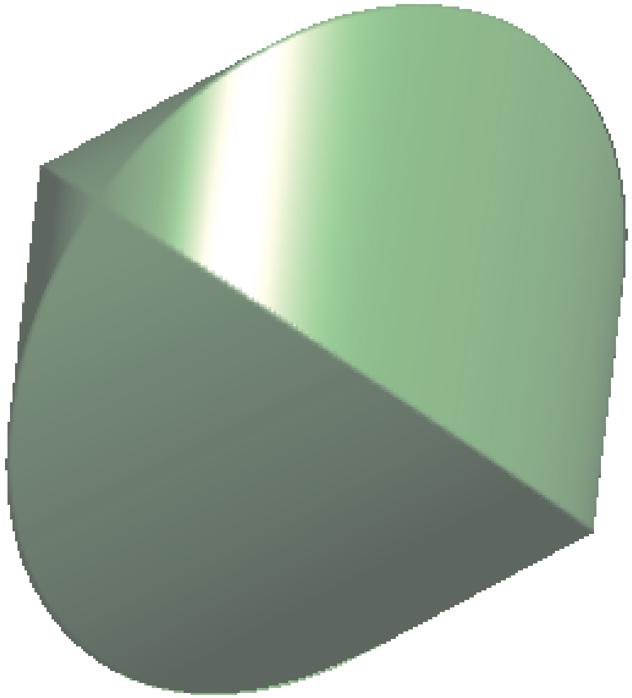


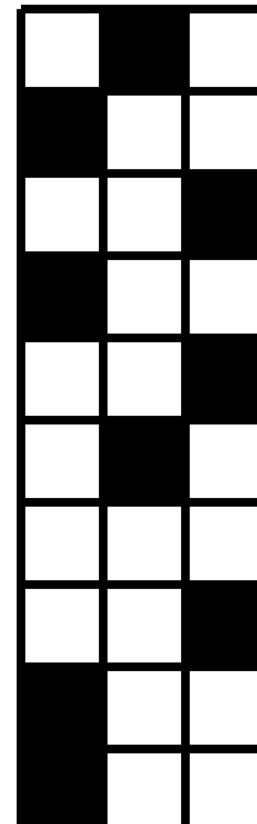
$X =$ 

An atomic norm

K. Vervier

A. d'Aspremont

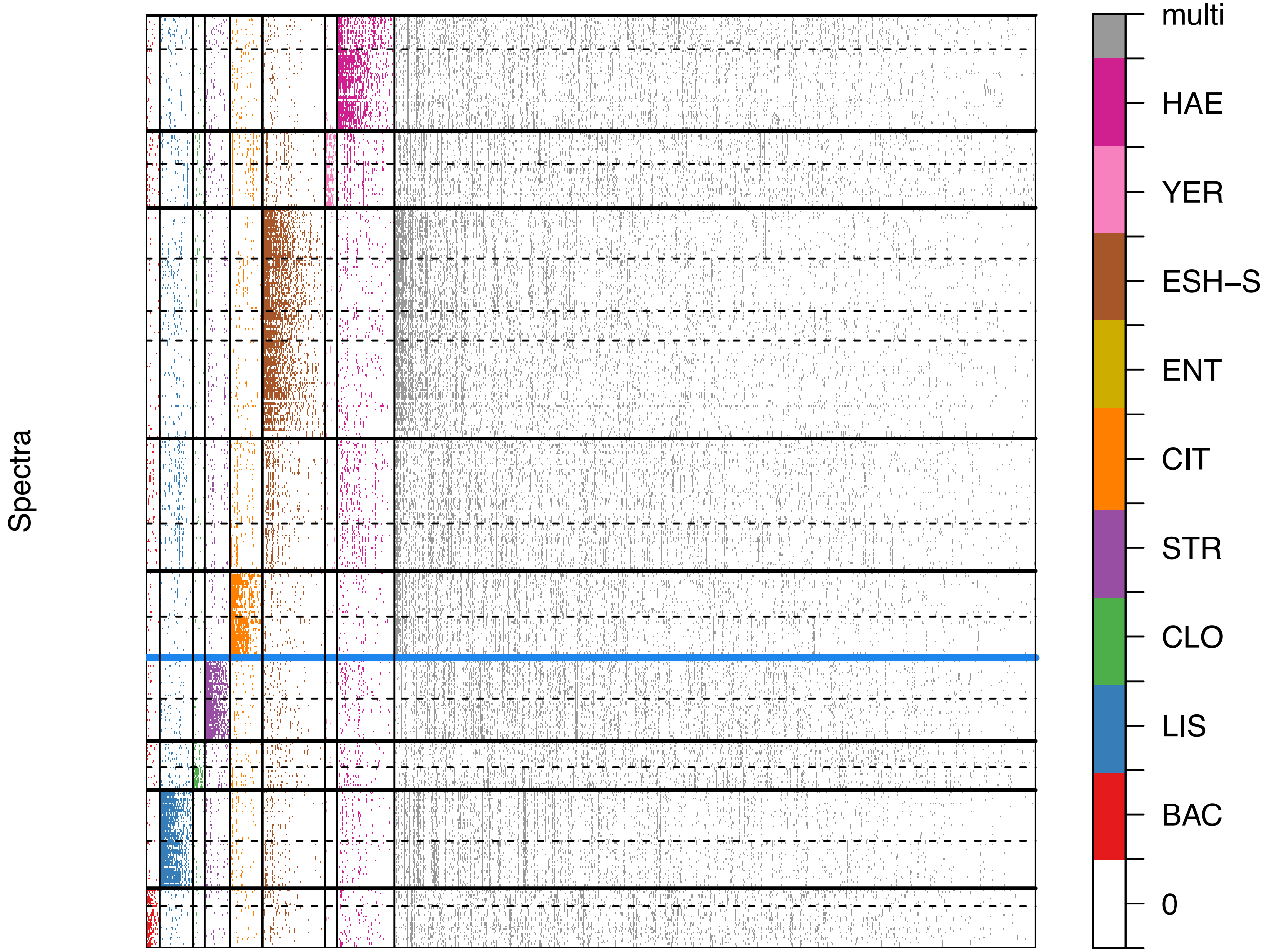


$X =$ 

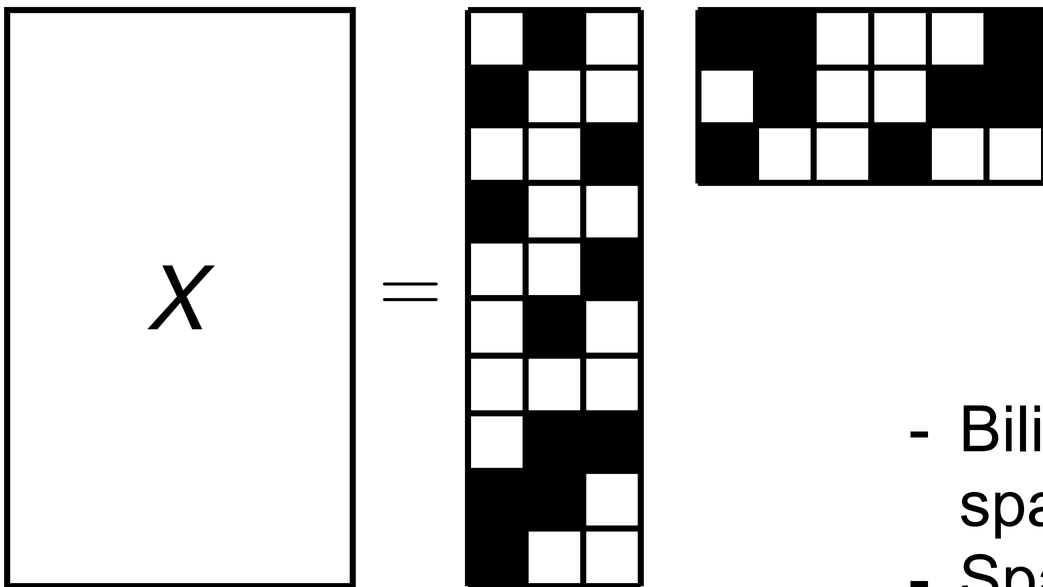
$$\Omega_K(X) = \sum_{i=1}^{p} K_{ii} \| x_i \|^2 + \sum_{i \neq j} K_{ij} \left| x_i^\top x_j \right|$$

(Vervier et al., 2014)

Application: Microbial identification from MALDI-TOF MS spectra



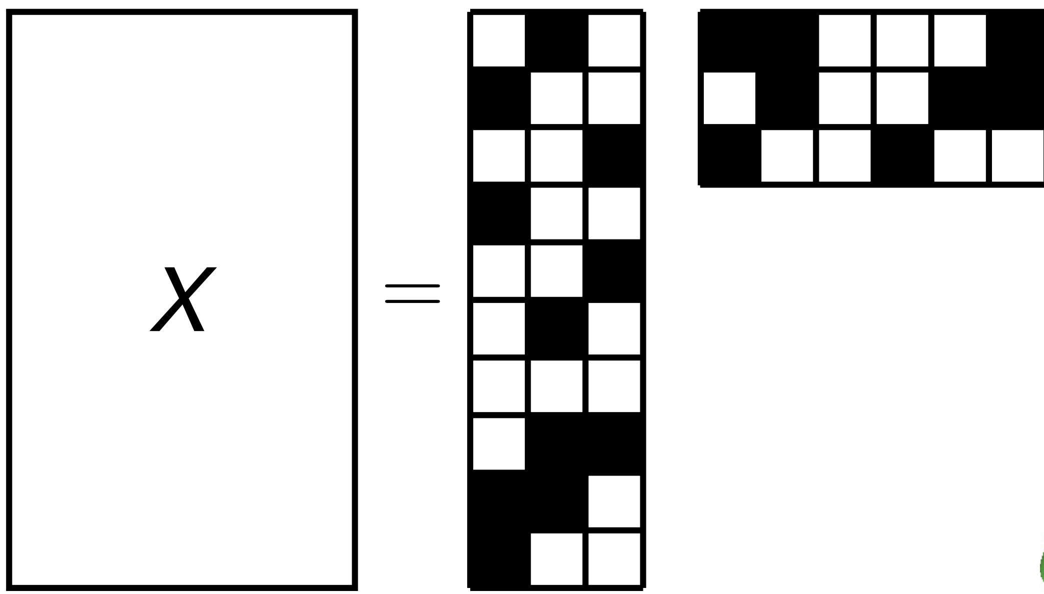
Learning low-rank matrices with sparse factors ?



$$X = \sum_{i=1}^{r} u_i v_i^\top$$

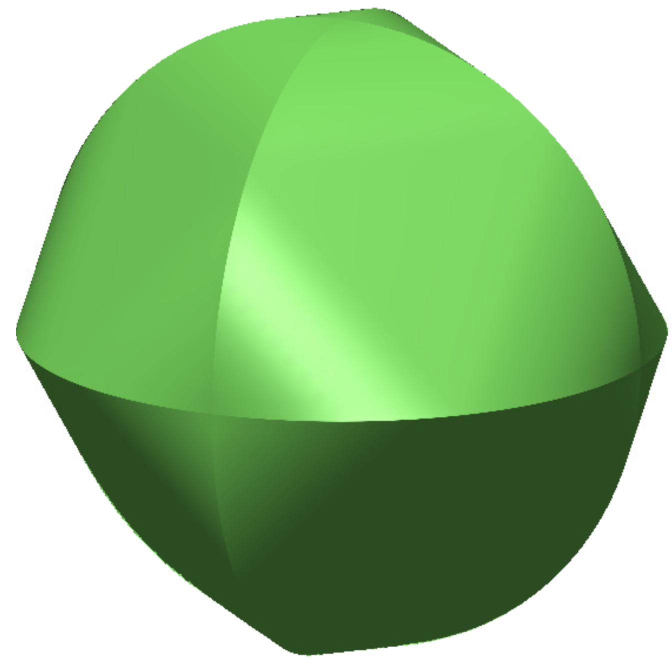
- Bilinear regression with sparse latent factors
- Sparse PCA
- Sparse CCA
- Hidden clique problem
- Community detection in networks

An atomic norm



E. Richard

G. Obozinski

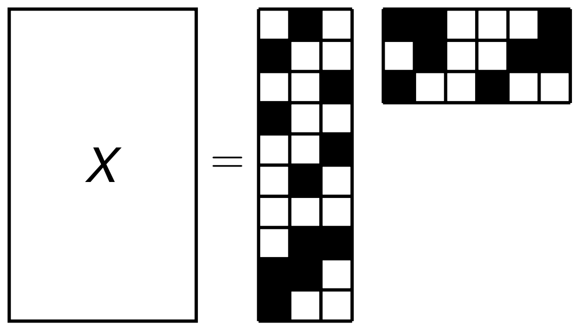


$$X = \sum_{i=1}^{r} u_i v_i^\top$$

$$\Omega_{k,q}(Z) = \inf\left\{ \sum_{(I,J)\in\mathcal{G}_{k,q}} \|A^{(IJ)}\|_* \;:\; Z = \sum_{(I,J)\in\mathcal{G}_{k,q}} A^{(IJ)}\;,\; \mathrm{supp}(A^{(IJ)}) \subset I \times J \right\}$$

(Richard et al., 2014)

An atomic norm



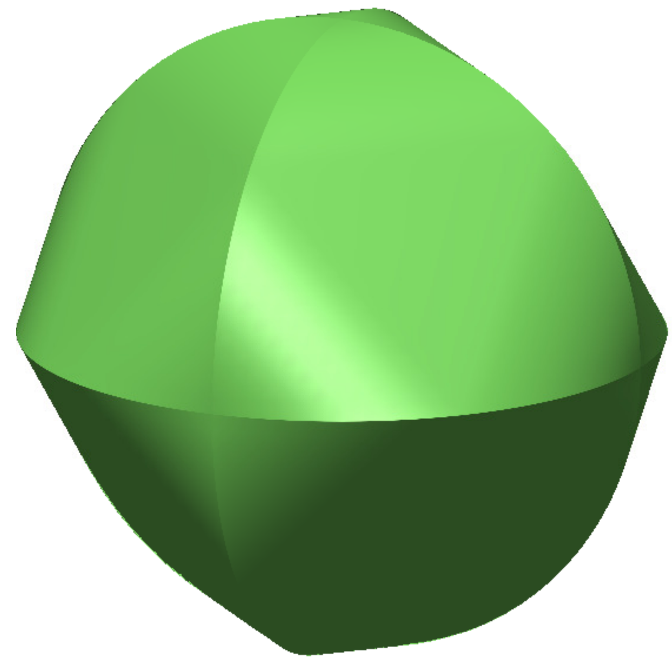
$$X = \sum_{i=1}^{r} u_i v_i^\top$$


E. Richard


G. Obozinski

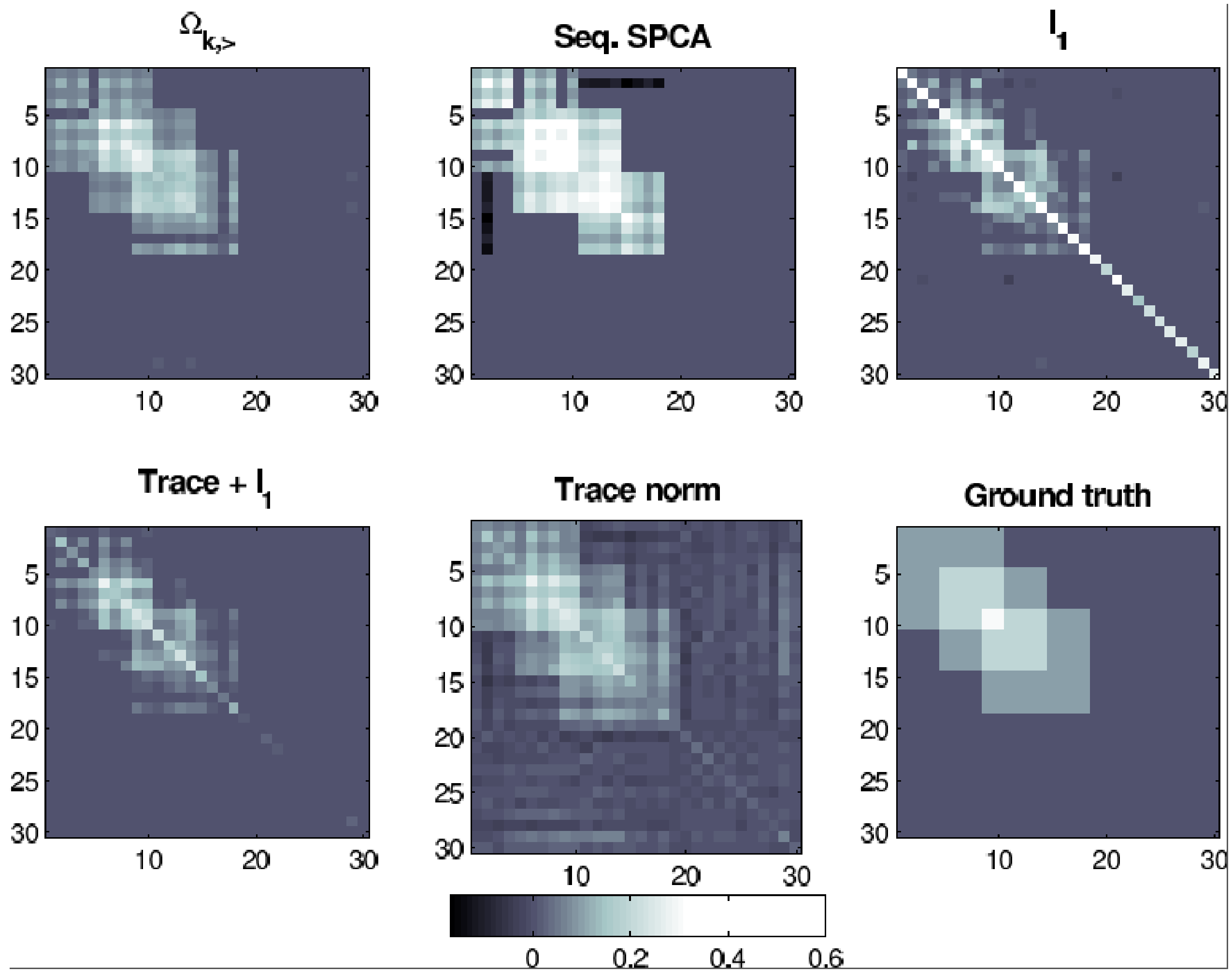
Theorem
Learning with this norm is « statistically optimal » to infer sparse low-rank matrices

But
Convex but NP-hard



(Richard et al., 2014)

Preliminary results on sparse PCA

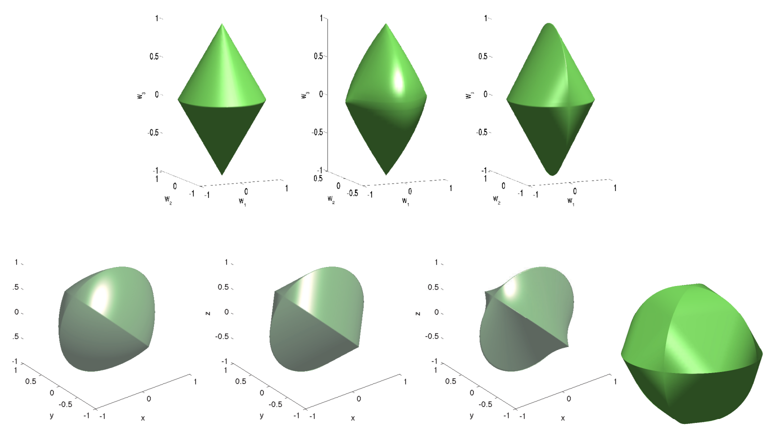


Sample covariance	Trace	ℓ_1	Trace + ℓ_1	Sequential	$\Omega_{k,\succeq}$
4.20 ± 0.02	0.98 ± 0.01	2.07 ± 0.01	0.96 ± 0.01	0.93 ± 0.08	$\mathbf{0.59 \pm 0.03}$

(Richard et al., 2014)

Summary

- Include prior knowledge: « sparse on some dictionary »
- Convex, (usually) computationally efficient
- Leads to interpretable model
- Good framework for data integration



Thanks

Future

- Find representations simple (for statistical reasons), robust to artefacts (batch, technology, …)

- n<<p still far from solved