Machine learning for patient stratification from genomic data

Jean-Philippe Vert

MINES ParisTech

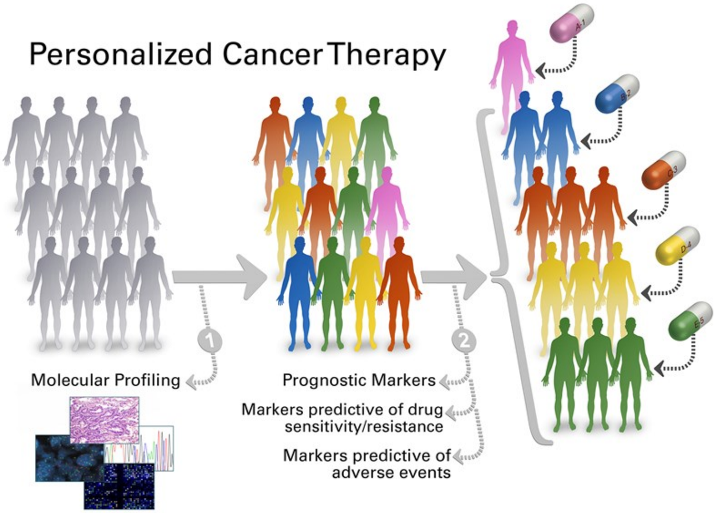
institut Curie
Together, let's beat cancer.

ENS
ÉCOLE NORMALE SUPÉRIEURE

PSL
RESEARCH UNIVERSITY PARIS

IHES, March 9, 2018

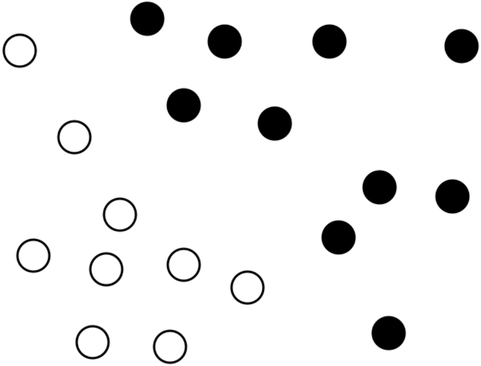
Goal


Personalized Cancer Therapy
1
Molecular Profiling
2
Prognostic Markers
Markers predictive of drug sensitivity/resistance
Markers predictive of adverse events
A-1
B-2
C-3
D-4
E-5

https://pct.mdanderson.org

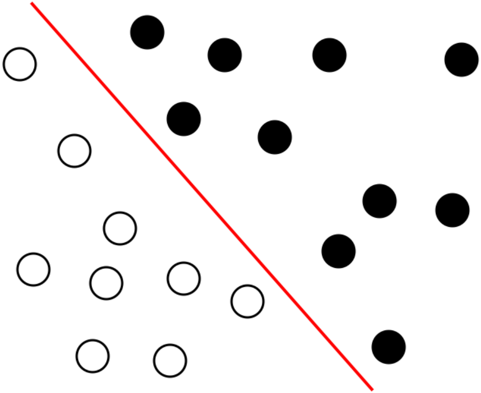
Mathematical model

- Patients with VS without relapse in 5 years
- n (=19) patients $>>$ p (=2) markers



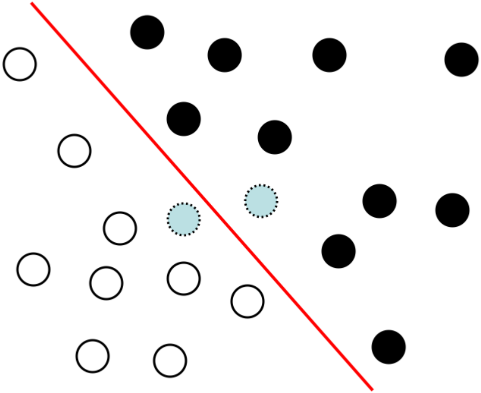
Mathematical model

- Patients with VS without relapse in 5 years
- n (=19) patients $>>$ p (=2) markers



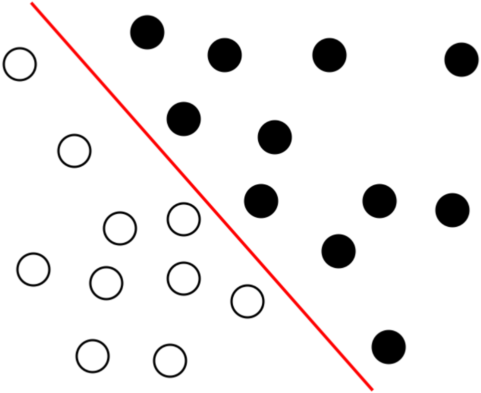
Mathematical model

- Patients with VS without relapse in 5 years
- n (=19) patients $>>$ p (=2) markers



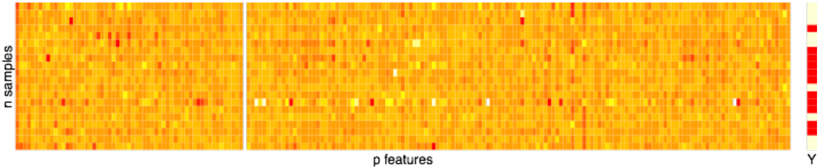
Mathematical model

- Patients with VS without relapse in 5 years
- *n* (=19) patients $>>$ *p* (=2) markers

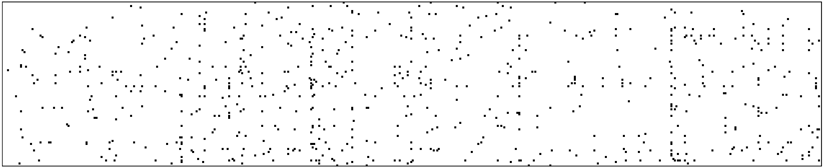


Real data: $n << p$

- Gene expression



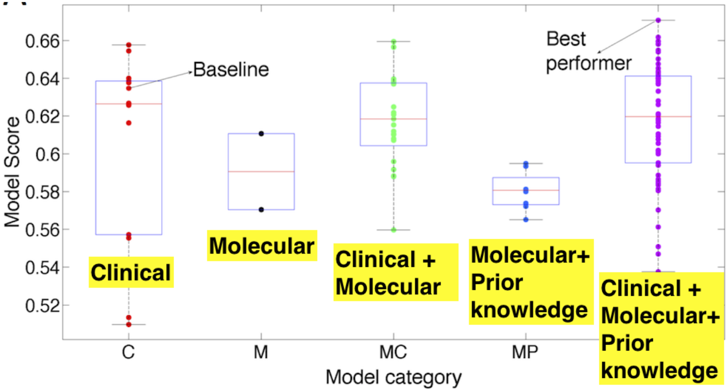
- Somatic mutations



- $n = 10^2 \sim 10^4$ (patients)
- $p = 10^4 \sim 10^7$ (genes, mutations, copy number, ...)
- Data of various nature (continuous, discrete, structured, ...)
- Data of variable quality (technical/batch variations, noise, ...)

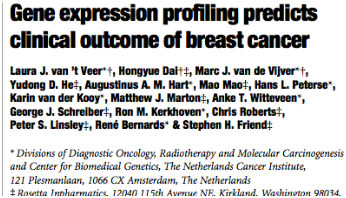
Consequence: limited accuracy

Breast cancer prognosis competition, $n = 2000$ (Bilal et al., 2013)



- C: 16 standard clinical data (age, tumor size, ...)
- M: 80k molecular features (gene expression, DNA copy number)

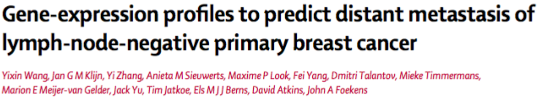
Consequence: unstable biomarker selection



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‡

** Divisions of Diagnostic Oncology, Radiotherapy and Molecular Carcinogenesis and Center for Biomedical Genetics, The Netherlands Cancer Institute, 121 Plesmanlaan, 1066 CX Amsterdam, The Netherlands*
‡ Rosetta Inpharmatics, 12040 115th Avenue NE, Kirkland, Washington 98034,



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

70 genes (Nature, 2002)

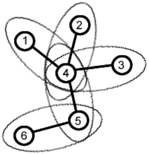
76 genes (Lancet, 2005)

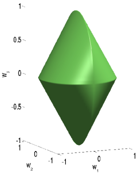
3 genes in common

van 't Veer et al. (2002); Wang et al. (2005)

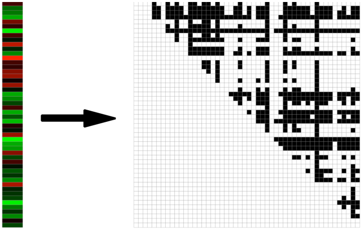
Some research directions

- Regularize and incorporate prior knowledge





- Find a better representation



Outline

1. Regularize

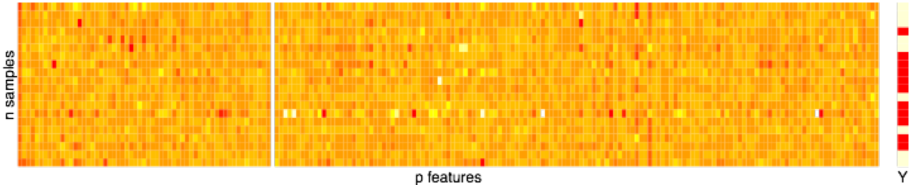
2. Change representation

Outline

1. Regularize

2. Change representation

Typical problem

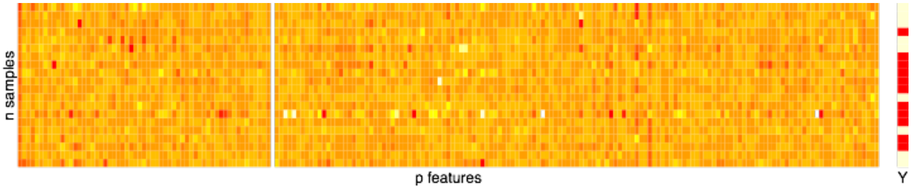


- n samples (patients), p features (genes)
- $X \in \mathbb{R}^{n \times p}$ gene expression profile of each patient
- $Y \in \mathcal{Y}^n$ survival information of each patient
- Fit a linear model for a sample $x \in \mathbb{R}^p$:

$$f(x) = \beta^\top x = \sum_{i=1}^{p} \beta_i x_i$$

- Standard methods (least squares or logistic regression) won't work because $n < p$

Regularized linear models



In high dimension, estimate β by solving

$$\min_{\beta \in \mathbb{R}^p} R(Y, X\beta) + \lambda J(\beta)\,,$$

where

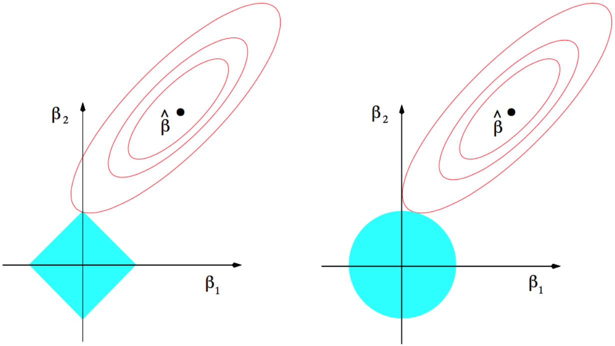
- $R(Y, X\beta)$ is an empirical risk to measures the fit to the training data
- $J(\beta)$ is a penalty to control the complexity of the model
- $\lambda > 0$ is a regularization parameter

Standard regularizations

$$\min_{\beta \in \mathbb{R}^p} R(Y, X\beta) + \lambda J(\beta)$$

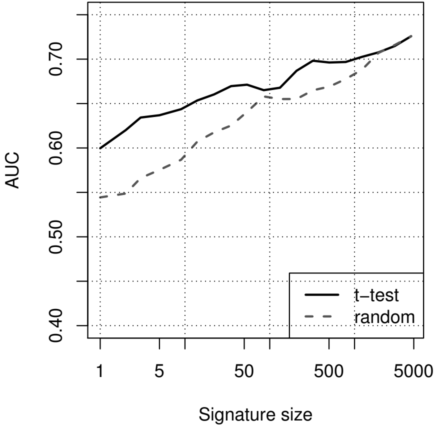
where

- Lasso: $J(\beta) = \|\beta\|_1$ for gene selection.
- Ridge: $J(\beta) = \|\beta\|_2^2$ to address $n \gg m$.
- Elastic net: $J(\beta) = \alpha\|\beta\|_2^2 + (1 - \alpha)\|\beta\|_1$

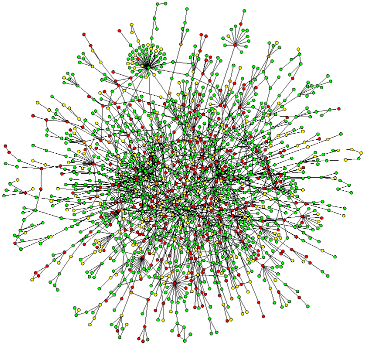


Which regularization is the best?

- Feature selection (lasso, t-tests, ...) is popular, it leads to a limited set of genes that form a molecular signatures
- Ridge is less interpretable but often leads to better performance... e.g., breast cancer prognosis ($n = 286$):

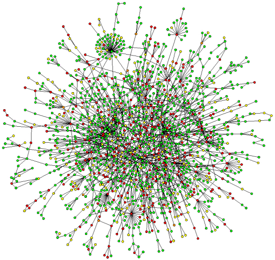


Adding prior knowledge: network-based regularizations



- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ a graph of genes (PPI, metabolic, signaling, regulatory network...)
- Prior knowledge:
  - β should be "smooth" on the graph?
  - Selected genes should be connected?

Examples of network-based regularizations



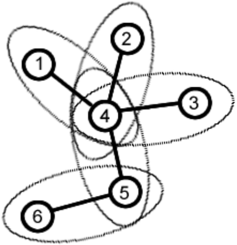
$$J_{\mathcal{G}}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2$$ (Rapaport et al., 2007)

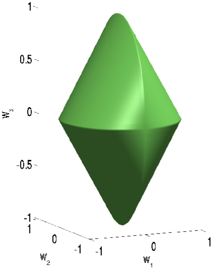
$$J_{\mathcal{G}}(\beta) = a\|\beta\|_1 + (1-a)\sum_{i \sim j} (\beta_i - \beta_j)^2$$ (Li and Li, 2008)

$$J_{\mathcal{G}}(\beta) = \sup_{\alpha \in \mathbb{R}^p : \forall i \sim j \; \alpha_i^2 + \alpha_j^2 \leq 1} \alpha^\top \beta$$ (Jacob et al., 2009)

$$J_{\mathcal{G}}(\beta) = a\|\beta\|_1 + (1-a)\sum_{i \sim j} |\beta_i - \beta_j|$$ (Hoefling, 2010)

Gene selection with the graph lasso

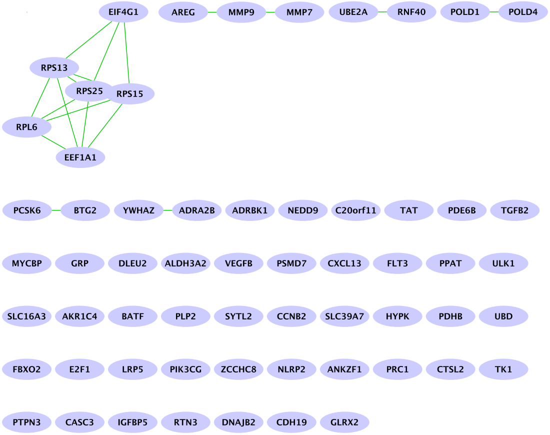




$$J_{\mathcal{G}}(\beta) = \sup_{\alpha\in\mathbb{R}^p:\forall i\sim j,\|\alpha_i^2+\alpha_j^2\|\leq 1} \alpha^\top \beta$$

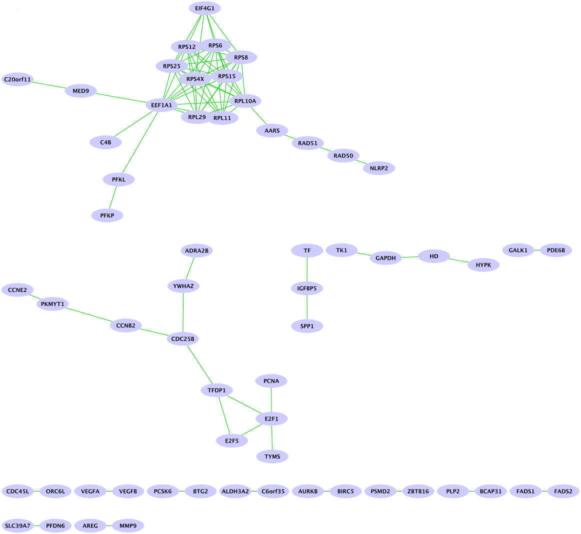
Jacob et al. (2009)

BC prognosis: Lasso signature (accuracy 0.61)



Jacob et al. (2009)

BC prognosis: Graph Lasso signature (accuracy 0.64)



Jacob et al. (2009)

Smoothness regularization and Fourier transform

- "Connected genes have similar weights" (Rapaport et al., 2007; Li and Li, 2008)

$$J_{\mathcal{G}}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2$$

- No feature selection
- Reinterpretation in the Fourier domain (Rapaport et al., 2007):

$$\sum_{i \sim j} (\beta_i - \beta_j)^2 = \sum_{i=1}^{p} \lambda_i \hat{\beta}_i^2$$

where
  - $\hat{\beta}_i$ is the i-th Fourier coefficient of β
  - λ_i is the i-th frequency
- "β has little energy at high frequency" and is therefore smooth on the graph

Smoothness regularization and Fourier transform

- "Connected genes have similar weights" (Rapaport et al., 2007; Li and Li, 2008)

$$J_{\mathcal{G}}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2$$

- No feature selection
- Reinterpretation in the Fourier domain (Rapaport et al., 2007):

$$\sum_{i \sim j} (\beta_i - \beta_j)^2 = \sum_{i=1}^{p} \lambda_i \hat{\beta}_i^2$$

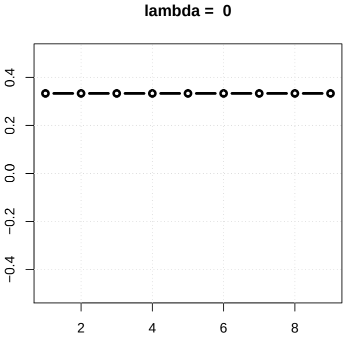
  where
  - $\hat{\beta}_i$ is the *i*-th Fourier coefficient of β
  - λ_i is the *i*-th frequency
- "β has little energy at high frequency" and is therefore smooth on the graph

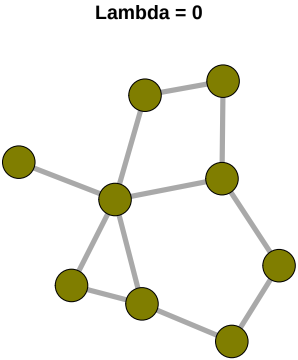
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = U^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis



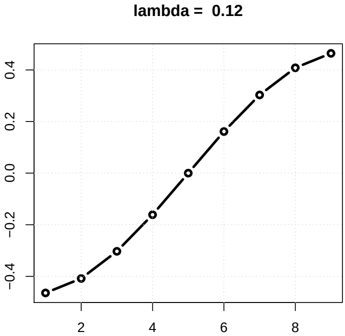


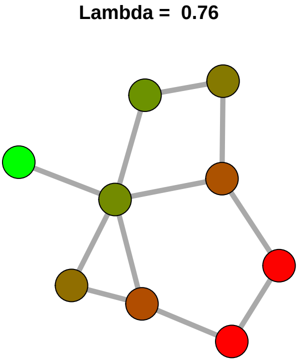
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = U^\top \beta$$

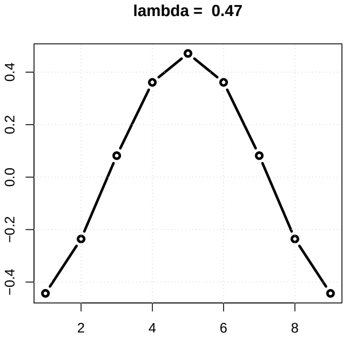
- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis

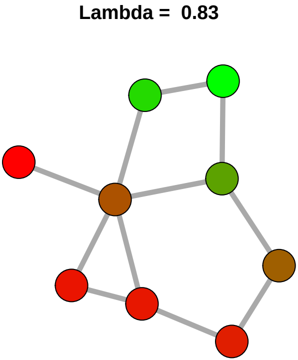




Graph Fourier transform $\hat{\beta}$?

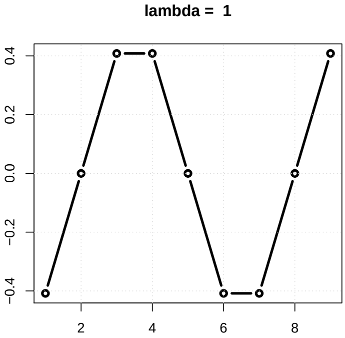
- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:
$$\hat{\beta} = U^\top \beta$$
- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis

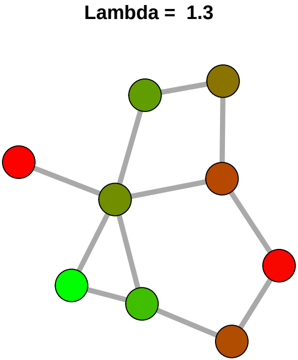




Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:
$$\hat{\beta} = U^\top \beta$$
- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis



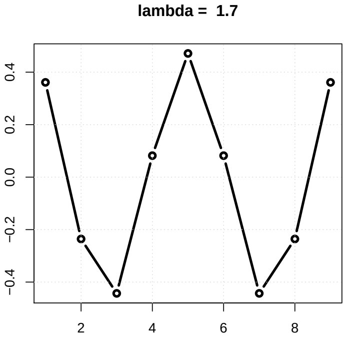


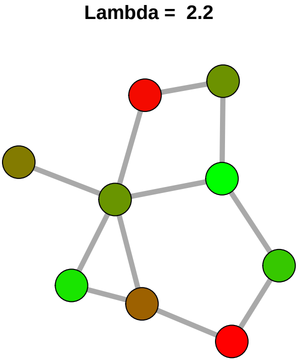
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = \boldsymbol{U}^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis





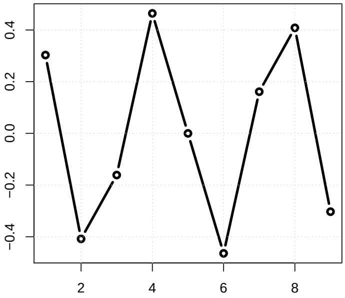
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors *U* of the graph Laplacian matrix form the Fourier basis:

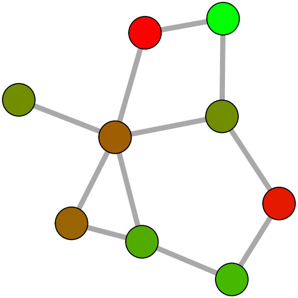
$$\hat{\beta} = \boldsymbol{U}^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis

lambda = 2.3



Lambda = 2.8

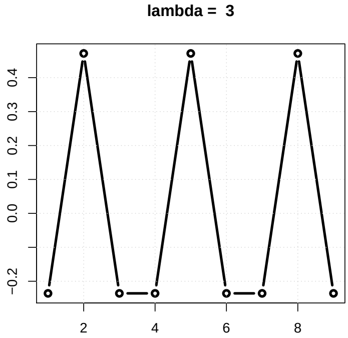


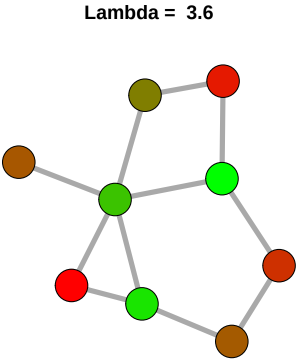
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = U^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis



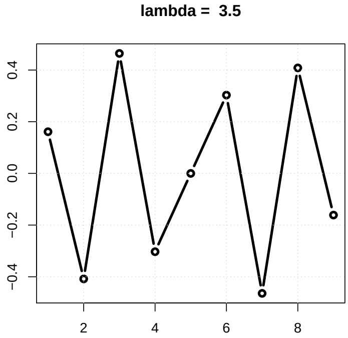


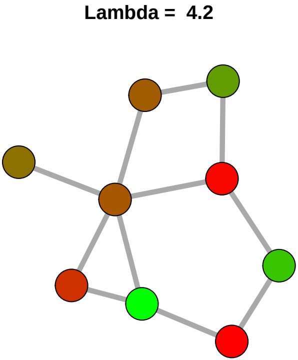
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = U^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis



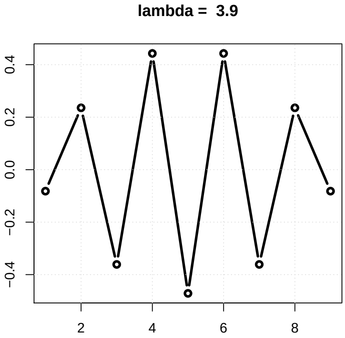


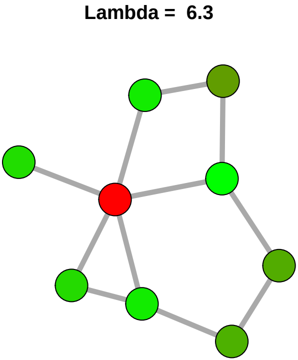
Graph Fourier transform $\hat{\beta}$?

- Eigenvectors U of the graph Laplacian matrix form the Fourier basis:

$$\hat{\beta} = U^\top \beta$$

- Eigenvalues $\Lambda = (0 = \lambda_1 \leq \ldots \leq \lambda_p)$ represent the "frequencies" of the Fourier basis


lambda = 3.9


Lambda = 6.3

Smoothness in the Fourier domain: extensions

- Rapaport et al. (2007) extends

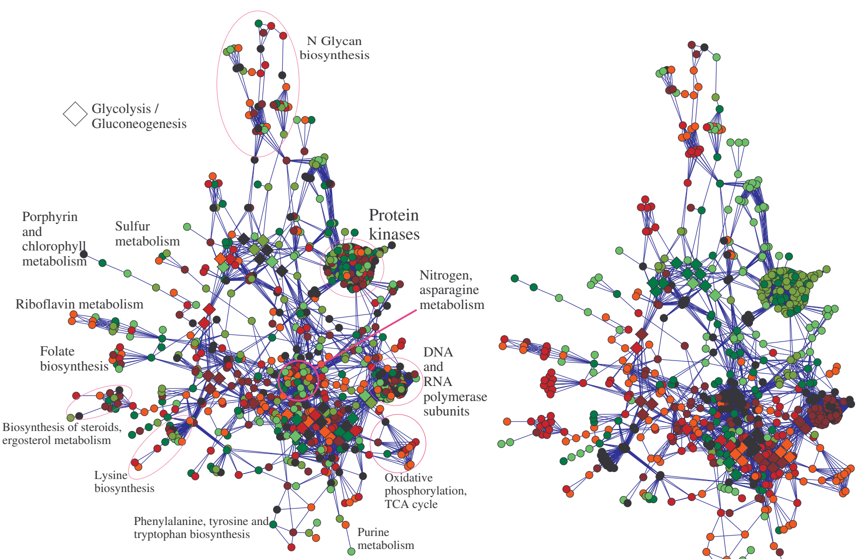
  $$\sum_{i \sim j} (\beta_i - \beta_j)^2 = \sum_{i=1}^{p} \lambda_i \hat{\beta}_i^2$$

  to

  $$\sum_{i=1}^{p} \phi(\lambda_i) \hat{\beta}_i^2$$

  for $\phi : \mathbb{R}^+ \to \mathbb{R}^+$ non-decreasing.
- Example: $\phi(\lambda) = \exp(-\gamma\lambda)$ linked to the diffusion kernel on the graph.

Classifiers

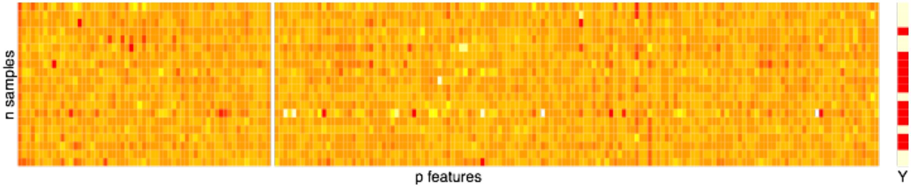


Outline

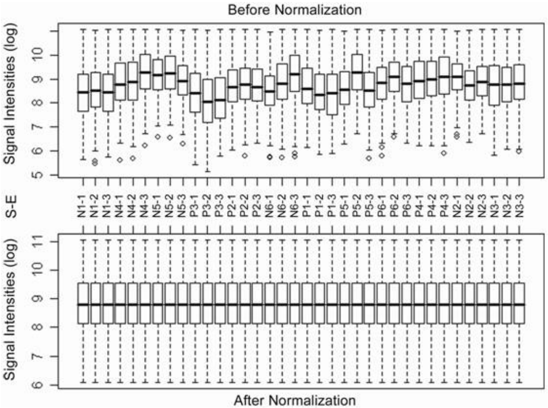
1. Regularize

2. Change representation

Back to the data

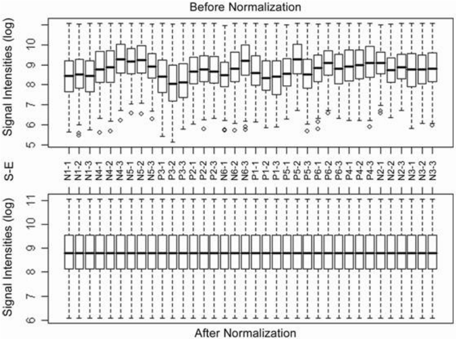


From raw data to *X*



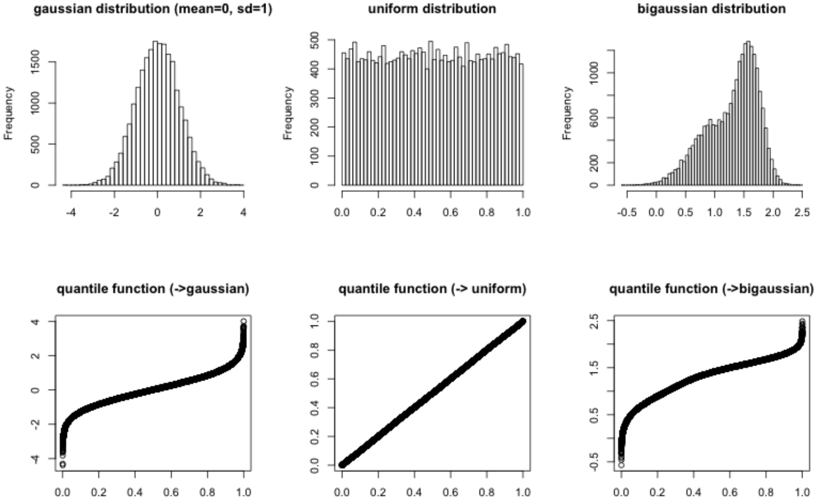
- Between-sample variability: batch effect, drift over time, ...
- Typical pre-processing: Quantile normalization per sample

Standard QN



- Fix a target quantile $f \in \mathbb{R}^n$
- Transform $x \in \mathbb{R}^p$ to $\Phi_f(x)$ such that:
  - The ranking of entries in x and $\Phi_f(x)$ are the same
  - The distribution of entries in $\Phi_f(x)$ follows f
- See also: images (Gonzalez and Woods, 2008), MRI scans (Shinohara et al., 2014), speech (Hilger and Ney, 2006)

How to choose a "good" target distribution?



From QN to supervised QN (Le Morvan and Vert, 2017)

Standard approaches: learn model after QN preprocessing:

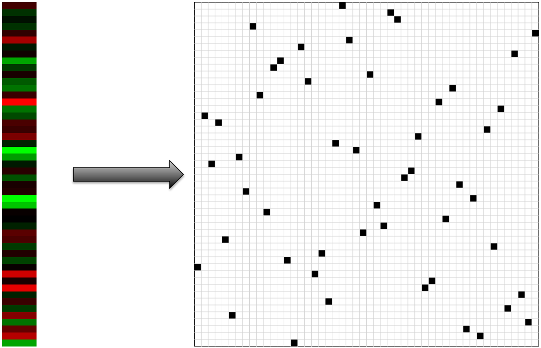
1. Fix f arbitrarily (typically, mean empirical quantile function)
2. QN all samples to get $\Phi_f(x_1), \ldots, \Phi_f(x_n)$
3. Learn a model on normalized data, e.g.:

$$\min_{w,b} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(w^\top \Phi_f(x_i) + b \right) + \lambda \Omega(w) \right\}$$

SUQUAN: jointly learn f and the model:

$$\min_{w,b,f} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell_i \left(w^\top \Phi_f(x_i) + b \right) + \lambda \Omega(w) + \gamma \Omega_2(f) \right\}$$

Computing $\Phi_f(x)$



For $x \in \mathbb{R}^p$ let

$$[\Pi_x]_{ij} = \begin{cases} 1 & \text{if } x_j \text{ has rank } i\,, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\Phi_f(x) = \Pi_x f$$

Linear SUQAN as rank-1 matrix regression

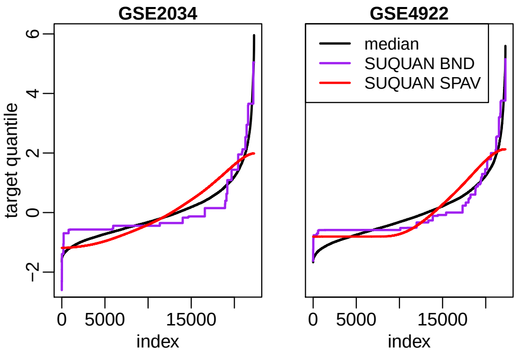
- Linear SUQAN therefore solves

$$
\begin{aligned}
&\min_{w,b,f}\left\{\frac{1}{n}\sum_{i=1}^{n}\ell_i\left(w^\top\Phi_f(x_i)+b\right)+\lambda\Omega(w)+\gamma\Omega_2(f)\right\}\\
&=\min_{w,b,f}\left\{\frac{1}{n}\sum_{i=1}^{n}\ell\left(w^\top\Pi_{x_i}f+b\right)+\lambda\Omega(w)+\gamma\Omega_2(f)\right\}\\
&=\min_{w,b,f}\left\{\frac{1}{n}\sum_{i=1}^{n}\ell\left(<wf^\top,\Pi_{x_i}>_{\text{Frobenius}}+b\right)+\lambda\Omega(w)+\gamma\Omega_2(f)\right\}
\end{aligned}
$$

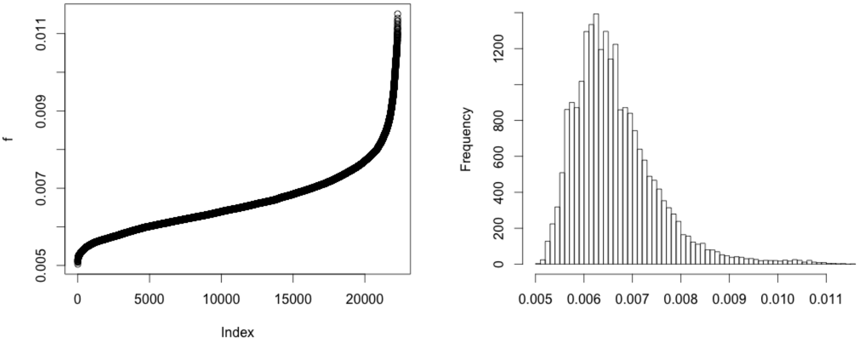
- A particular linear model to estimate a rank-1 matrix $M = wf^\top$
- Each sample $x \in \mathbb{R}^p$ is represented by the matrix $\Pi_x \in \mathbb{R}^{p\times p}$
- Non-convex
- Alternative optimization of f and w is easy

Results: gene expression data

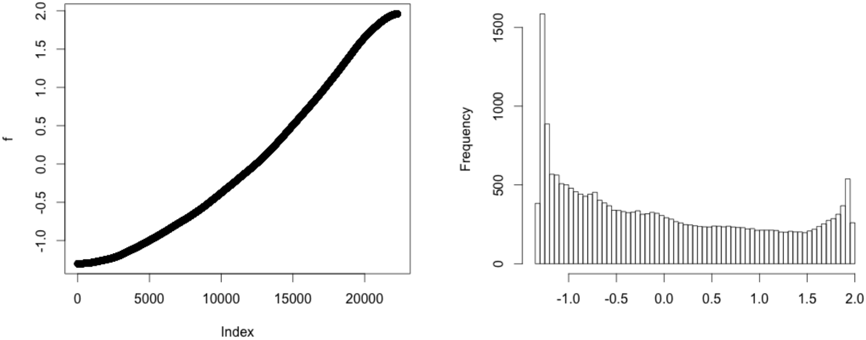
	LOGISTIC REGRESSION							SUQUAN		
	RAW	RMA	CAUCHY	EXP.	UNIF.	GAUS.	MEDIAN	SVD	BND	SPAV
GSE1456	65.94	68.73	59.56	68.86	68.72	69.00	69.06	57.60	**71.44**	69.60
GSE2034	74.52	75.42	61.91	74.53	75.22	**76.45**	74.92	52.61	70.50	76.11
GSE2990	57.01	60.43	54.72	**61.25**	56.25	58.66	59.72	52.51	59.22	59.94
GSE4922	58.52	58.86	55.24	58.81	55.66	60.01	59.18	52.39	**61.82**	61.41
AVERAGE	64.00	65.86	57.86	65.86	63.96	66.03	65.72	53.78	65.75	**66.77**



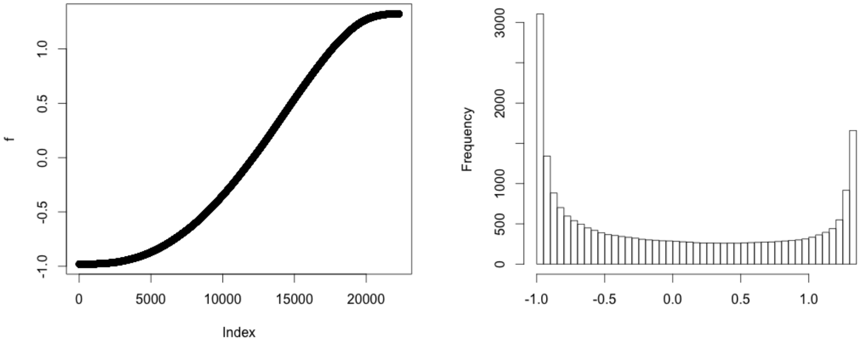
Estimated quantile function: iteration=0



Estimated quantile function: iteration=1

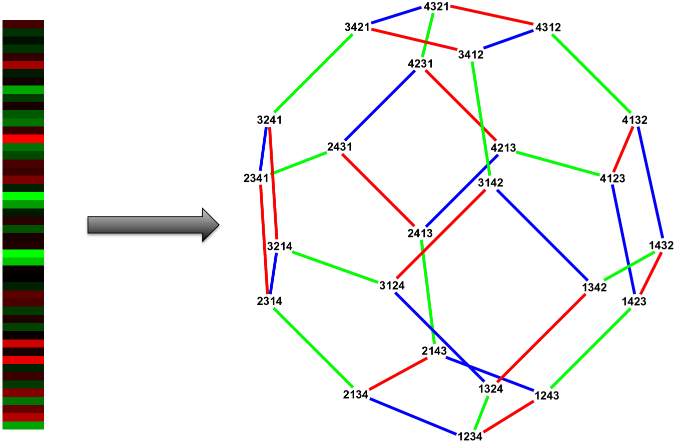


Estimated quantile function: iteration=2

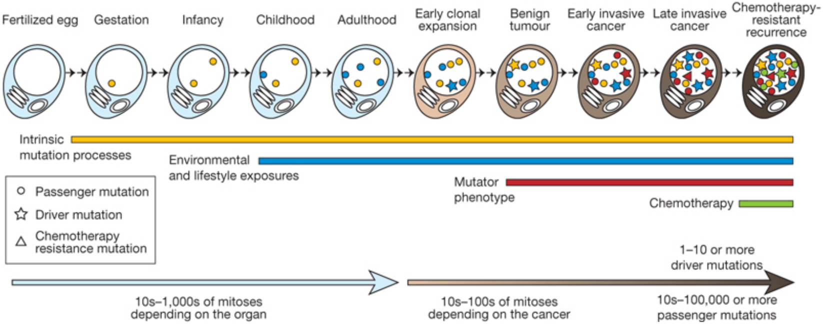


Remark: embedding $\mathbb{R}^n$ to $\mathbb{S}_n$

- Remark: each sample $x \in \mathbb{R}^p$ was represented by the permutation of genes $\sigma \in S_p$
- Many other possibilities when we decide to embed data to the symmetric group $\mathbb{S}_n$



Somatic mutations in cancer



Fertilized egg, Gestation, Infancy, Childhood, Adulthood, Early clonal expansion, Benign tumour, Early invasive cancer, Late invasive cancer, Chemotherapy-resistant recurrence

Intrinsic mutation processes

Environmental and lifestyle exposures

Mutator phenotype

Chemotherapy

○ Passenger mutation

☆ Driver mutation

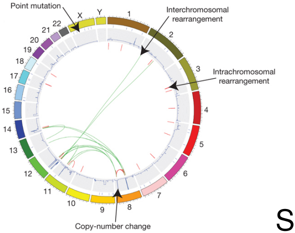
△ Chemotherapy resistance mutation

10s–1,000s of mitoses depending on the organ

10s–100s of mitoses depending on the cancer

1–10 or more driver mutations

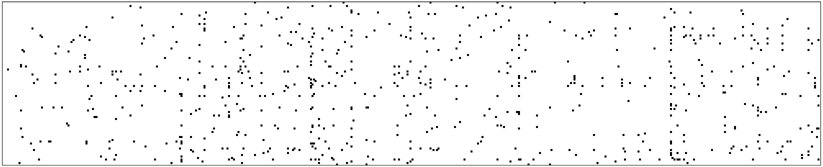
10s–100,000 or more passenger mutations



Stratton et al. (2009)

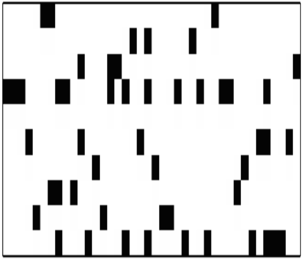
Large-scale efforts to collect somatic mutations

- 3, 378 samples with survival information from 8 cancer types
- downloaded from the TCGA / cBioPortal portals.



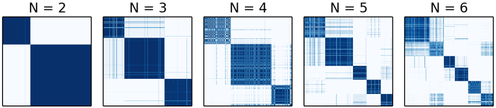
Cancer type	Patients	Genes
LUAD (Lung adenocarcinoma)	430	20 596
SKCM (Skin cutaneous melanoma)	307	17 463
GBM (Glioblastoma multiforme)	265	14 750
BRCA (Breast invasive carcinoma)	945	16 806
KIRC (Kidney renal clear cell carcinoma)	411	10 609
HNSC (Head and Neck squamous cell carcinoma)	388	17 022
LUSC (Lung squamous cell carcinoma)	169	13 590
OV (Ovarian serous cystadenocarcinoma)	363	10 195

Patient stratification (unsupervised) from raw mutation profiles

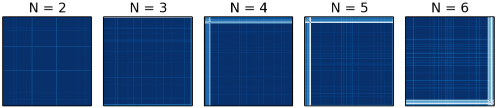


✓ Non-Negative matrix factorisation (NMF)

✓ Desired behaviour:



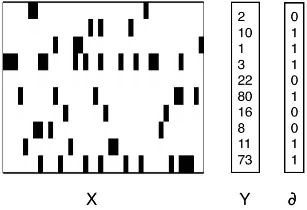
✓ Observed behaviour:

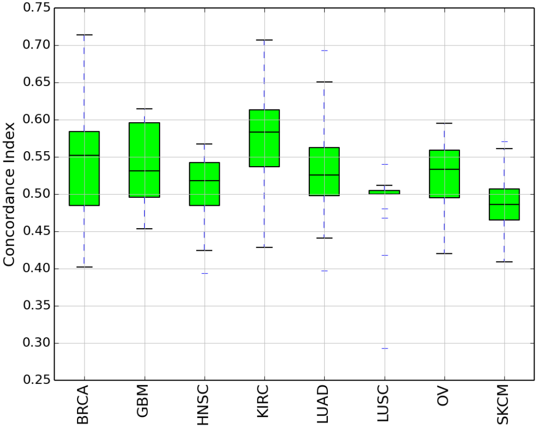


Patients share very few mutated genes!

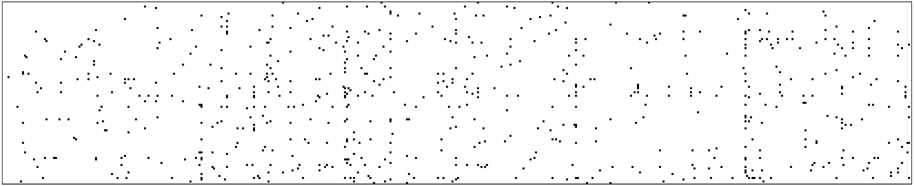
Survival prediction from raw mutation profiles

- Each patient is a binary vector: each gene is mutated (1) or not (2)
- Silent mutations are removed
- Survival model estimated with sparse survival SVM
- Results on 5-fold cross-validation repeated 4 times





Approach: change representation?



Can we replace

$$x \in \{0, 1\}^p \quad \text{with } p \text{ very large, very sparse}$$

by a representation with more information shared between samples

$$\Phi(x) \in \mathcal{H}$$

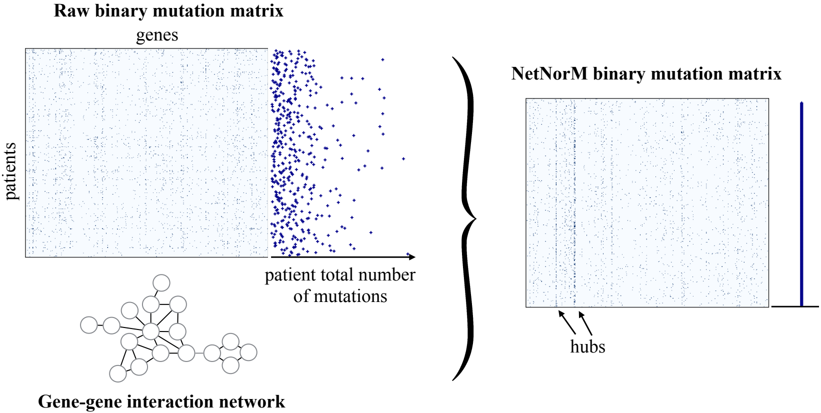
that would allow better supervised and unsupervised classification?

NetNorm Overview (Le Morvan et al., 2017)

Take

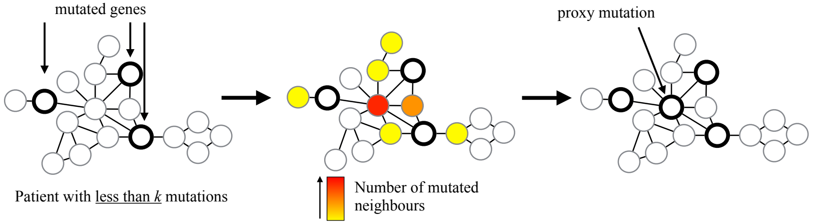
$$\mathcal{H} = \left\{ x \in \{0,1\}^p \, : \, \sum_{i=1}^{p} x_i = K \right\}$$

and use a gene network to transform x to $\phi(x) \in \mathcal{H}$ by adding/removing mutations

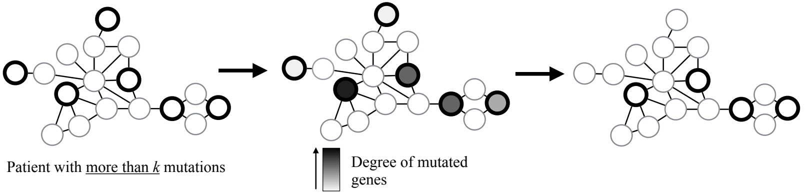

Raw binary mutation matrix
genes
patients
patient total number of mutations
Gene-gene interaction network
NetNorM binary mutation matrix
hubs

NetNorm detail (k=4)

1. Add mutations for patients with few (less than *K*) mutations

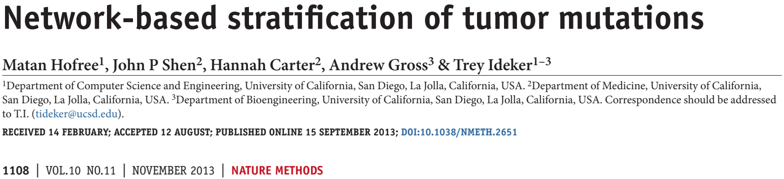

mutated genes
proxy mutation
Patient with less than *k* mutations
Number of mutated neighbours

2. Remove mutations for patients for many (more than *K*) mutations



In practice, *K* is a free parameter optimized on the training set, typically a few 100's.

Related work (Hofree et al., 2013)



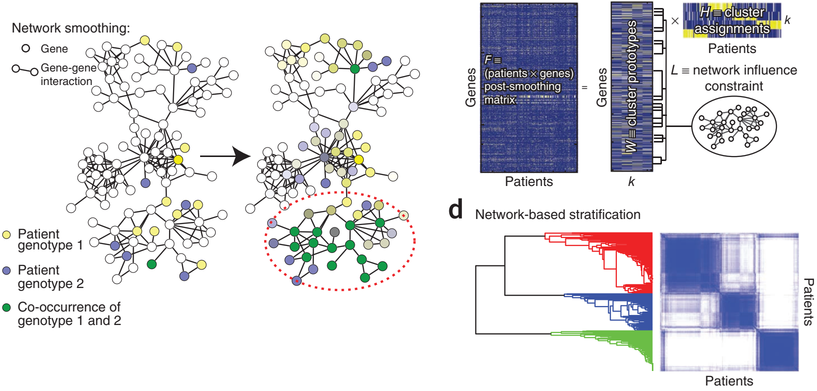
Network-based stratification of tumor mutations

Matan Hofree[1], John P Shen[2], Hannah Carter[2], Andrew Gross[3] & Trey Ideker[1–3]

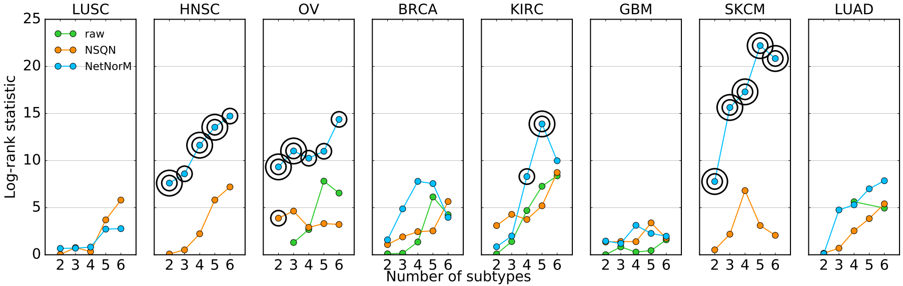
[1]Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California, USA. [2]Department of Medicine, University of California, San Diego, La Jolla, California, USA. [3]Department of Bioengineering, University of California, San Diego, La Jolla, California, USA. Correspondence should be addressed to T.I. (tideker@ucsd.edu).

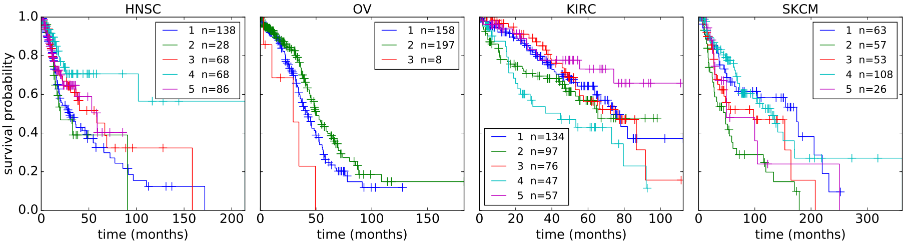
RECEIVED 14 FEBRUARY; ACCEPTED 12 AUGUST; PUBLISHED ONLINE 15 SEPTEMBER 2013; DOI:10.1038/NMETH.2651

1108 | VOL.10 NO.11 | NOVEMBER 2013 | NATURE METHODS

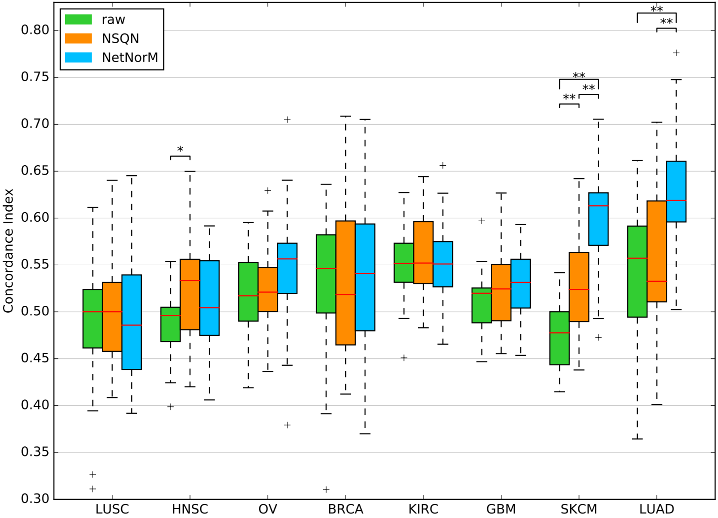


Results: unsupervised classification





Results: survival prediction

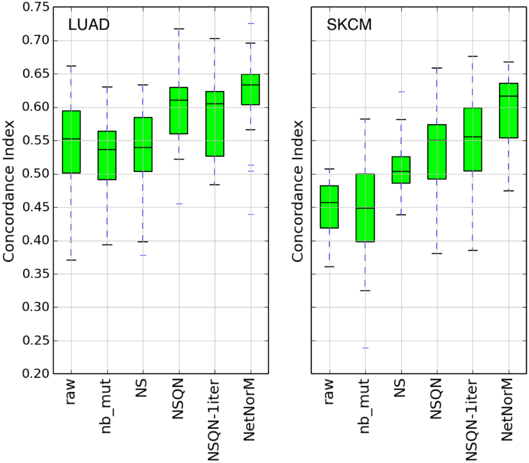


Use Pathway Commons as gene network.
NSQN = Network Smoothing / Quantile Normalization (Hofree et al., 2013)

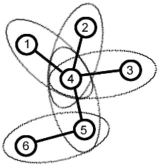
QN matters...

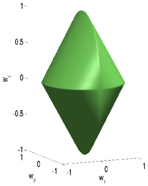
Both NetNorm and NSQN transforms follow a 2-step a approach:

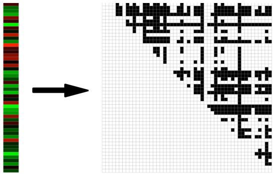
1. Smooth the raw data onto the gene network (NS)
2. Quantile normalize the smoothed profile (QN)



Conclusion







- Learning from genomic data is challenging
- Regularization is needed in high dimension
- A good representation is worth a thousand learning algorithms
- Subtle interplay between biology and math/CS
- Impact on the final quality/performance of the model
- Recent trend: learn the representation

Thanks

erc

SEVENTH FRAMEWORK PROGRAMME

INVESTISSEMENTS D'AVENIR

AGENCE NATIONALE DE LA RECHERCHE ANR

BIOMÉRIEUX

MINES ParisTech

institutCurie
Together, let's beat cancer.

Inserm
Institut national
de la santé et de la recherche médicale

PSL★ RESEARCH UNIVERSITY

SANOFI

THE UNIVERSITY OF CALIFORNIA BERKELEY · LET THERE BE LIGHT · 1868

The Adolph C. and Mary Sprague
Miller Institute for Basic
Research in Science
University of California, Berkeley

SIMONS INSTITUTE
for the Theory of Computing

1794
ENS
ÉCOLE NORMALE SUPÉRIEURE

References

E. Bilal, J. Dutkowski, J. Guinney, I. S. Jang, B. A. Logsdon, G. Pandey, B. A. Sauerwine, Y. Shimoni, H. K. Moen V., B. H. Mecham, O. M. Rueda, J. Tost, C. Curtis, M. J. Alvarez, V. N. Kristensen, S. Aparicio, A.-L. BÃÿrresen-Dale, C. Caldas, A. Califano, S. H. Friend, T. Ideker, E. E. Schadt, G. A. Stolovitzky, and A. A. Margolin. Improving breast cancer survival analysis through competition-based multidimensional modeling. *PLoS computational biology*, 9: e1003047, 2013. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1003047. URL `http://dx.doi.org/10.1371/journal.pcbi.1003047`.

R. C. Gonzalez and R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice Hall, 2008.

F. Hilger and H. Ney. Quantile based histogram equalization for noise robust large vocabulary speech recognition. *IEEE Trans. Audio, Speech, Language Process.*, 14(3):845–854, 2006. doi: 10.1109/TSA.2005.857792. URL `http://dx.doi.org/10.1109/TSA.2005.857792`.

H. Hoefling. A path algorithm for the Fused Lasso Signal Approximator. *J. Comput. Graph. Stat.*, 19(4):984–1006, 2010. doi: 10.1198/jcgs.2010.09208. URL `http://dx.doi.org/10.1198/jcgs.2010.09208`.

M. Hofree, J. P. Shen, H. Carter, A. Gross, and T. Ideker. Network-based stratification of tumor mutations. *Nat Methods*, 10(11):1108–1115, Nov 2013. doi: 10.1038/nmeth.2651. URL `http://dx.doi.org/10.1038/nmeth.2651`.

L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 433–440, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553431. URL `http://dx.doi.org/10.1145/1553374.1553431`.

References (cont.)

Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. In *Proceedings of The 32nd International Conference on Machine Learning*, volume 37 of *JMLR:W&CP*, pages 1935–1944, 2015. URL `http://jmlr.org/proceedings/papers/v37/jiao15.html`.

Y. Jiao and J.-P. Vert. The Kendall and Mallows kernels for permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. doi: 10.1109/TPAMI.2017.2719680. URL `http://dx.doi.org/10.1109/TPAMI.2017.2719680`.

W. R. Knight. A computer method for calculating Kendall's tau with ungrouped data. *J. Am. Stat. Assoc.*, 61(314):436–439, 1966. URL `http://www.jstor.org/stable/2282833`.

M. Le Morvan and J.-P. Vert. Supervised quantile normalisation. Technical Report 1706.00244, arXiv, 2017.

M. Le Morvan, A. Zinovyev, and J.-P. Vert. NetNorM: capturing cancer-relevant information in somatic exome mutation data with gene networks for cancer stratification and prognosis. *PLoS Comp. Bio.*, 13(6):e1005573, 2017. URL `http://hal.archives-ouvertes.fr/hal-01341856`.

C. Li and H. Li. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, 24:1175–1182, May 2008. ISSN 1367-4811. doi: 10.1093/bioinformatics/btn081.

F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J.-P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:35, 2007. doi: 10.1186/1471-2105-8-35. URL `http://dx.doi.org/10.1186/1471-2105-8-35`.

References (cont.)

R. T. Shinohara, E. M. Sweeney, J. Goldsmith, N. Shiee, F. J. Mateen, P. A. Calabresi, S. Jarso, D. L. Pham, D. S. Reich, C. M. Crainiceanu, A. I. B. L. F. S. o. A. , and A. D. N. I. . Statistical normalization techniques for magnetic resonance imaging. *Neuroimage Clin*, 6:9–19, 2014. doi: 10.1016/j.nicl.2014.08.008. URL `http://dx.doi.org/10.1016/j.nicl.2014.08.008`.

M. R. Stratton, P. J. Campbell, and P. A. Futreal. The cancer genome. *Nature*, 458(7239): 719–724, Apr 2009. doi: 10.1038/nature07943. URL `http://dx.doi.org/10.1038/nature07943`.

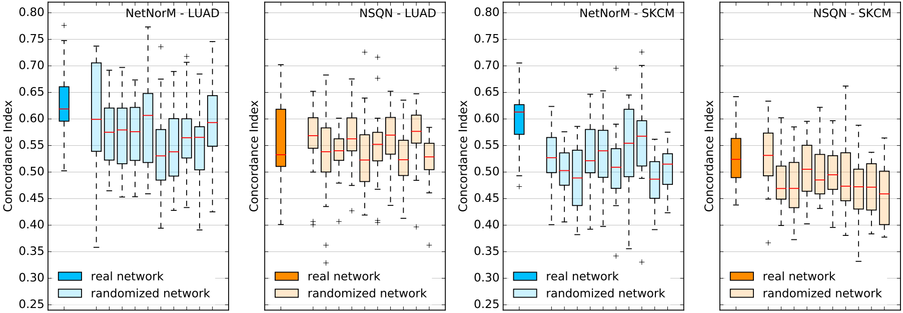
R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B*, 58(1): 267–288, 1996. URL `http://www.jstor.org/stable/2346178`.

L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancers. *Nature*, 415(6871):530–536, Jan 2002. doi: 10.1038/415530a. URL `http://dx.doi.org/10.1038/415530a`.

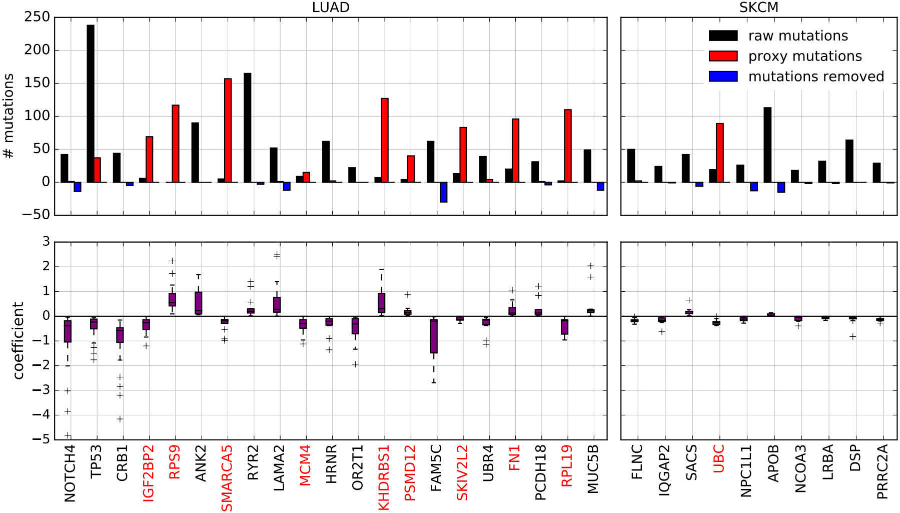
Y. Wang, J. Klijn, Y. Zhang, A. Sieuwerts, M. Look, F. Yang, D. Talantov, M. Timmermans, M. Meijer-van Gelder, J. Yu, T. Jatkoe, E. Berns, D. Atkins, and J. Foekens. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancers. *Lancet*, 365(9460):671–679, 2005. doi: 10.1016/S0140-6736(05)17947-1. URL `http://dx.doi.org/10.1016/S0140-6736(05)17947-1`.

NetNorM and NSQN benefit from biological information in the gene network

Comparison with 10 randomly permuted networks:

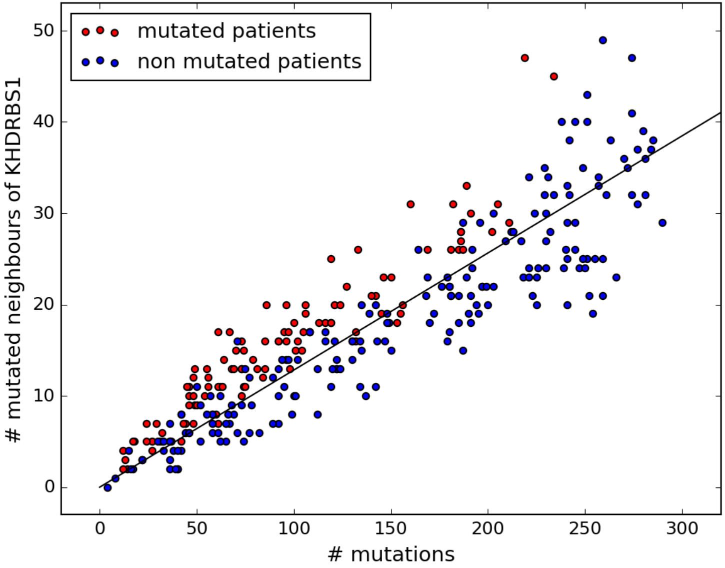


Selected genes represent "true" or "proxy" mutations

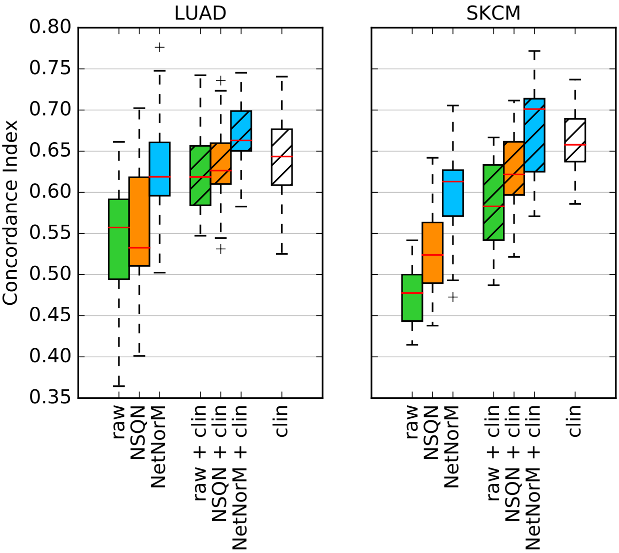


Genes selected in at least 50% of the cross-validated sparse SVM model

Proxy mutations encode both total number of mutations and local mutational burden

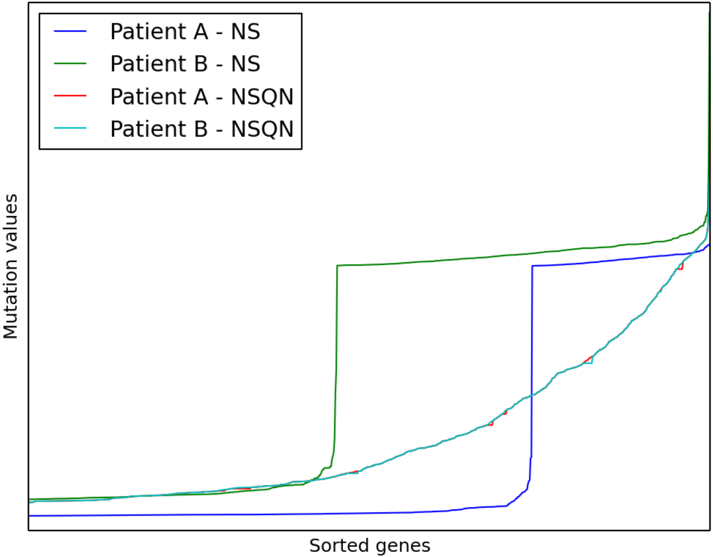


Adding good old clinical factors

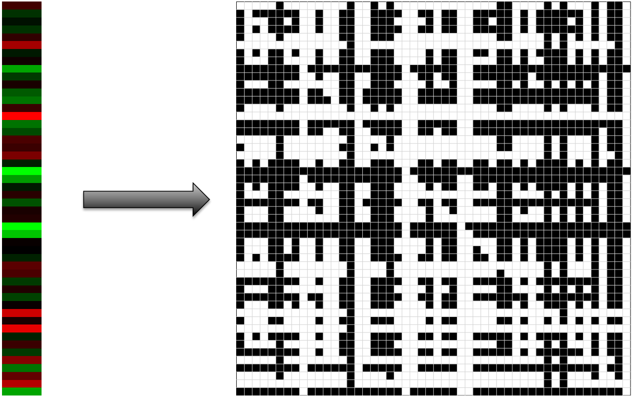


Combination by averaging predictions

QN after network smoothing

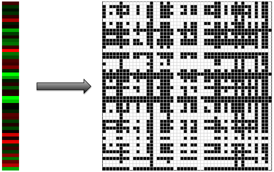


Another representation



$$\Phi_{i,j}(x) = \begin{cases} 1 & \text{if } x_i \leq x_j \, , \\ 0 & \text{otherwise.} \end{cases}$$

Geometry of the embedding



For any two permutations $\sigma, \sigma' \in \mathbb{S}_n$:

- Inner product

$$\Phi(\sigma)^\top \Phi(\sigma') = \sum_{1 \leq i \neq j \leq n} \mathbb{1}_{\sigma(i)<\sigma(j)} \mathbb{1}_{\sigma'(i)<\sigma'(j)} = n_c(\sigma, \sigma')$$

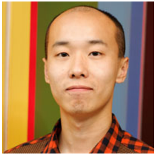
  n_c = number of concordant pairs

- Distance

$$\| \Phi(\sigma) - \Phi(\sigma') \|^2 = \sum_{1 \leq i,j \leq n} (\mathbb{1}_{\sigma(i)<\sigma(j)} - \mathbb{1}_{\sigma'(i)<\sigma'(j)})^2 = 2n_d(\sigma, \sigma')$$

  n_d = number of discordant pairs

Kendall and Mallows kernels (Jiao and Vert, 2017)



- The Kendall kernel is

$$K_\tau(\sigma, \sigma') = n_c(\sigma, \sigma')$$

- The Mallows kernel is

$$\forall \lambda \geq 0 \quad K_M^\lambda(\sigma, \sigma') = e^{-\lambda n_d(\sigma, \sigma')}$$

Theorem (Jiao and Vert, 2015, 2017)

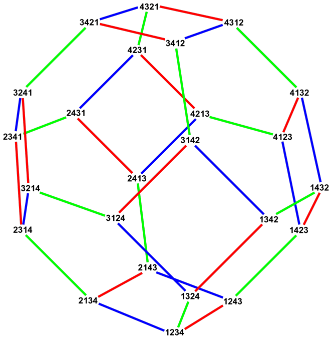
The Kendall and Mallows kernels are positive definite.

Theorem (Knight, 1966)

These two kernels for permutations can be evaluated in $O(n \log n)$ time.

Kernel trick useful with few samples in large dimensions

Related work



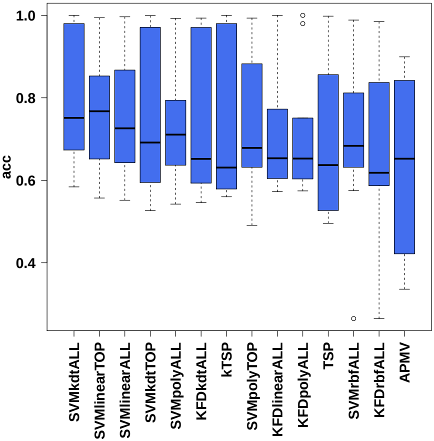
Cayley graph of $\mathbb{S}_4$

- Kondor and Barbarosa (2010) proposed the diffusion kernel on the Cayley graph of the symmetric group generated by adjacent transpositions.
- Computationally intensive ($O(n^{2n})$)
- Mallows kernel is written as

  $$K_M^\lambda(\sigma, \sigma') = e^{-\lambda n_d(\sigma, \sigma')},$$

  where $n_d(\sigma, \sigma')$ is the shortest path distance on the Cayley graph.
- It can be computed in $O(n \log n)$

Applications



Average performance on 10 microarray classification problems (Jiao and Vert, 2017).