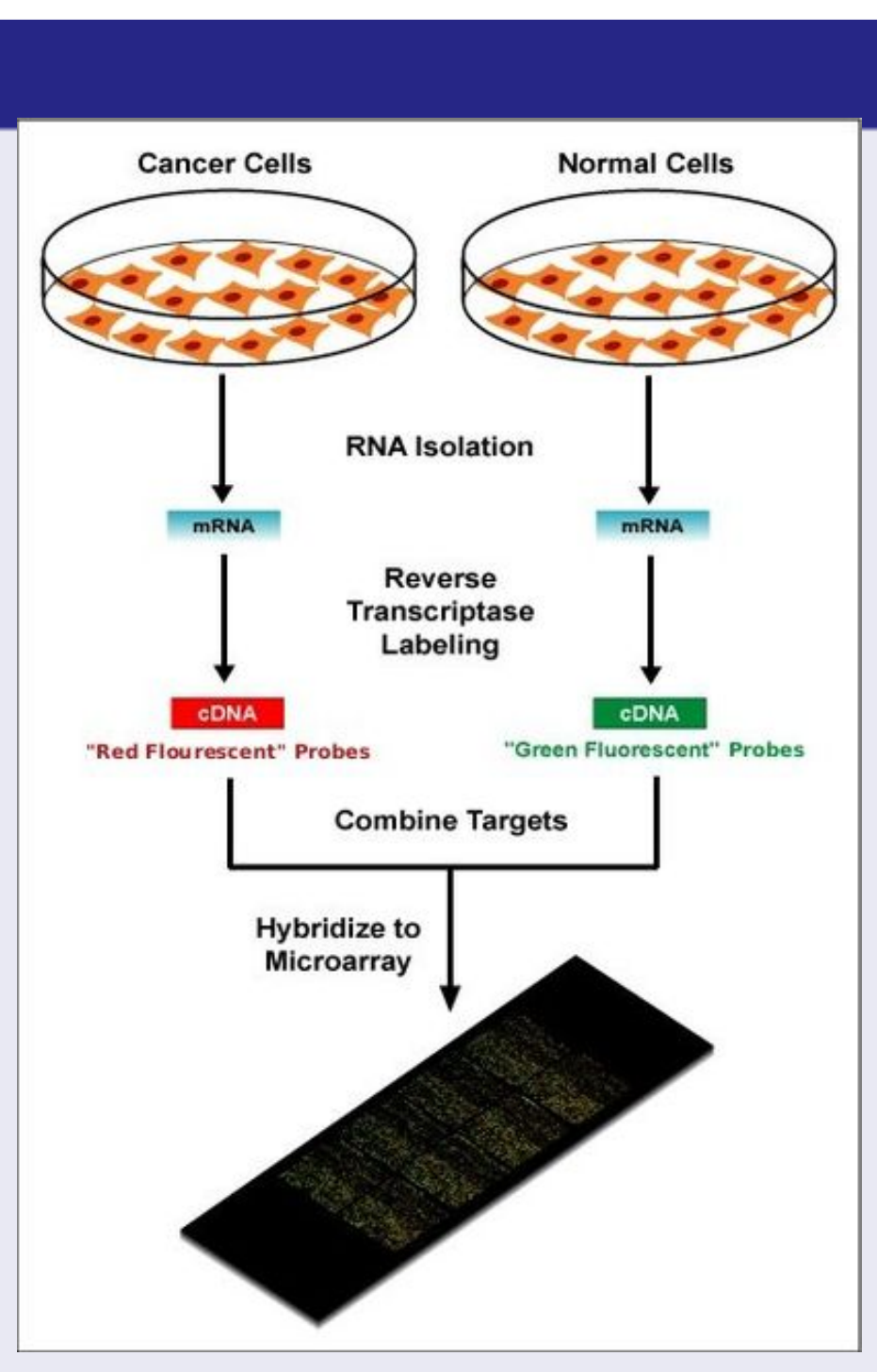
Analyzing How Protein Interaction Networks Improve Classification Performance in Gene Expression Data Analysis

Adrin Jalali
Supervised by: Nico Pfeifer

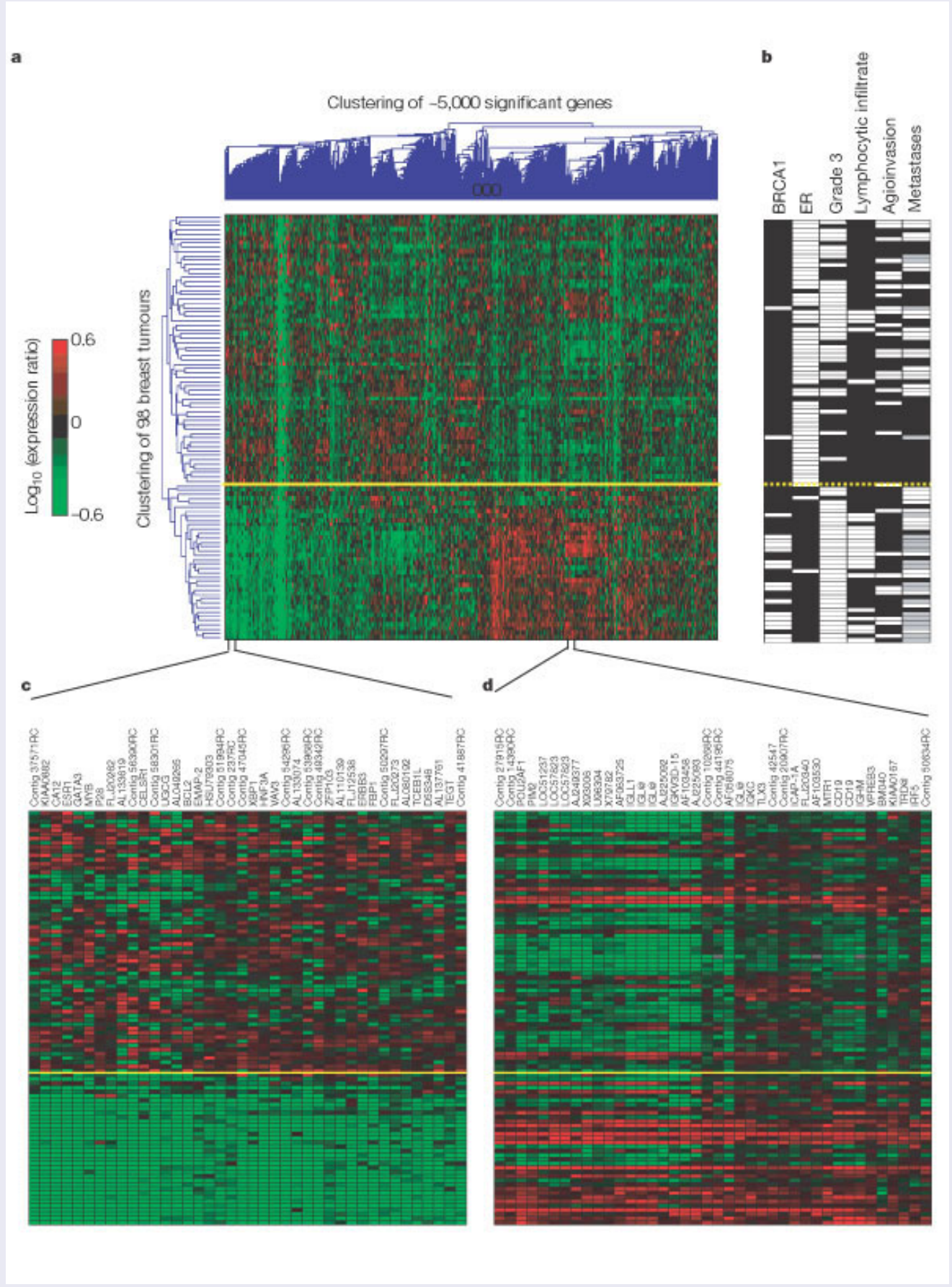
MPI for Informatics

July 12, 2013

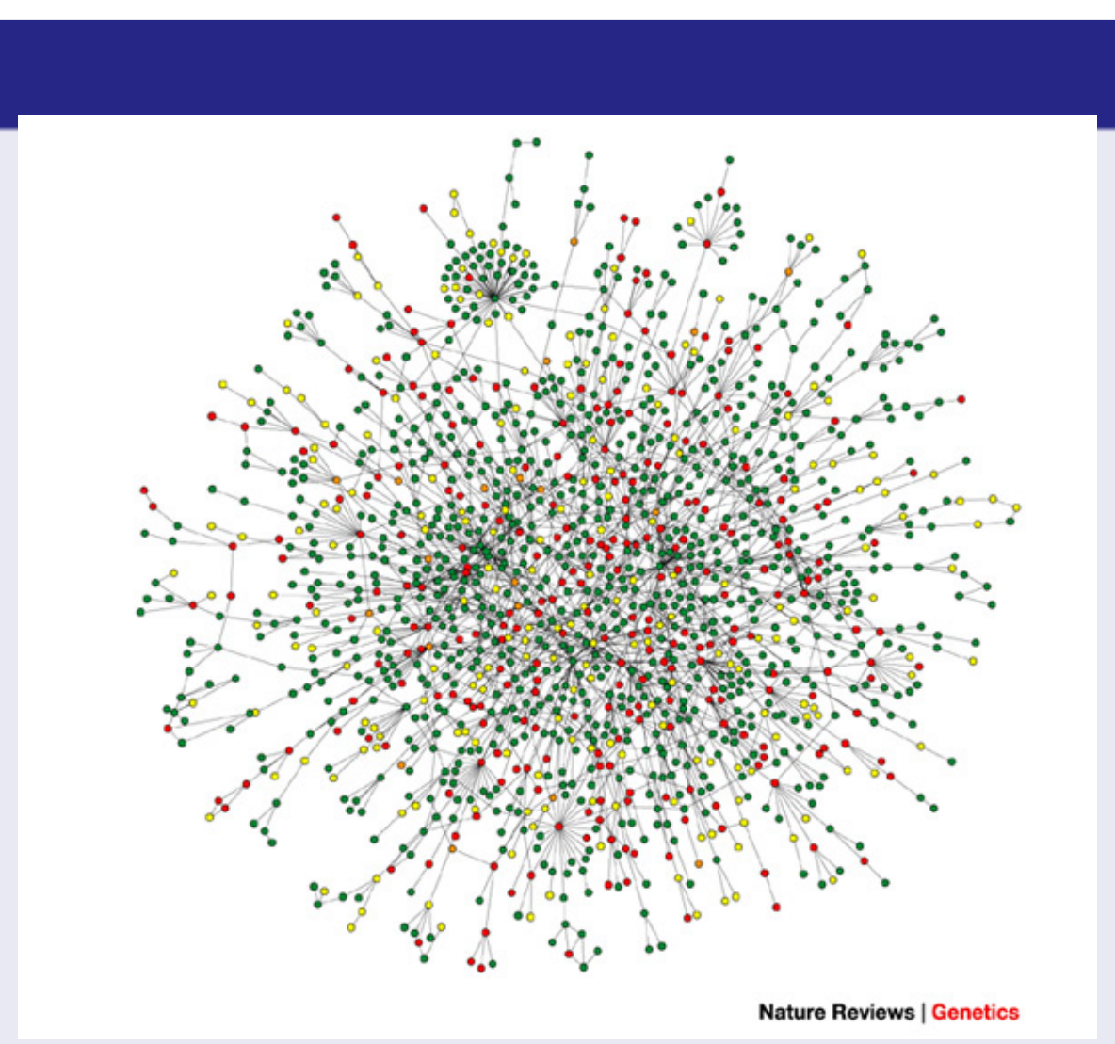
Data



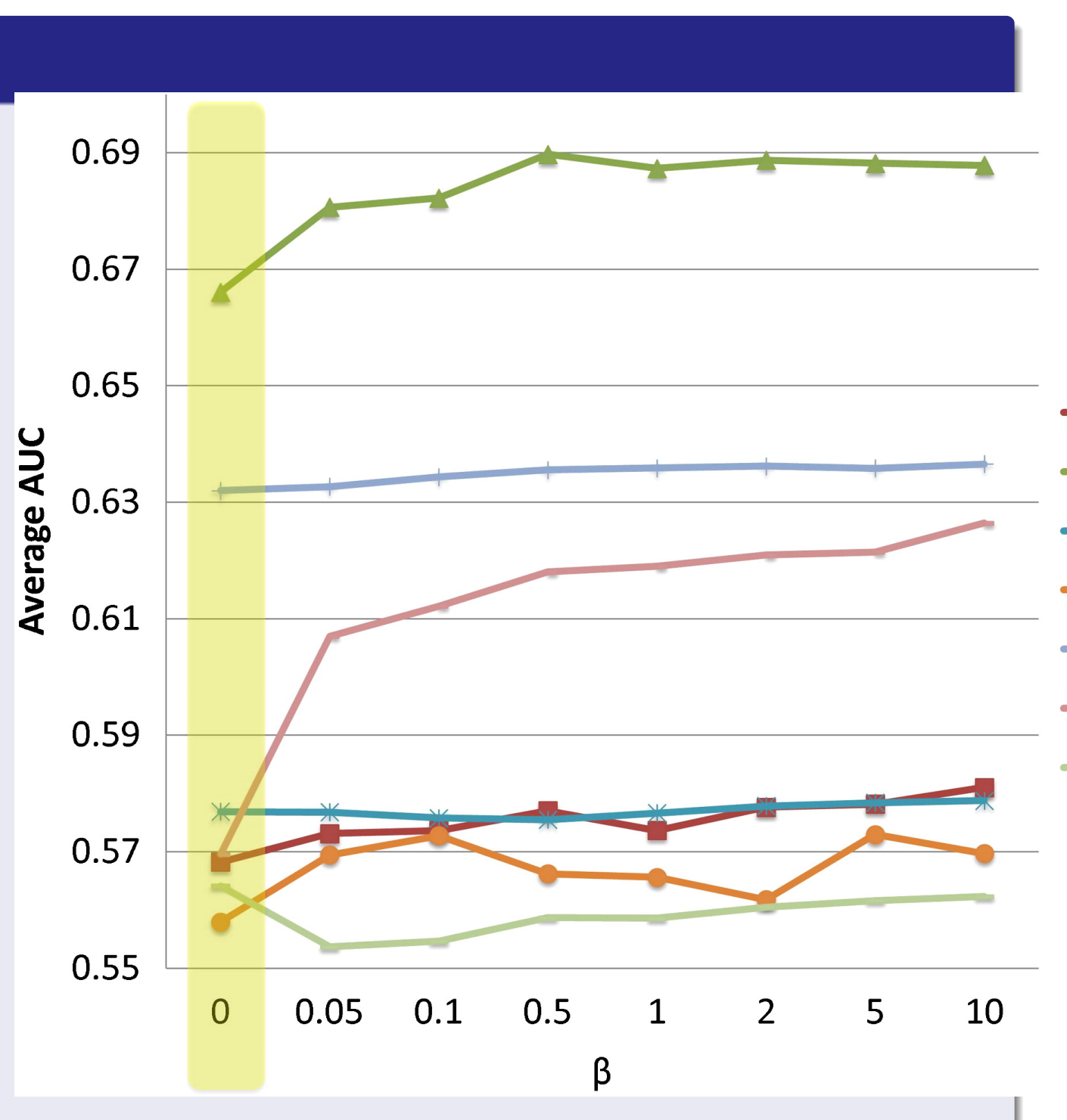
Microarray Gene Expression Data



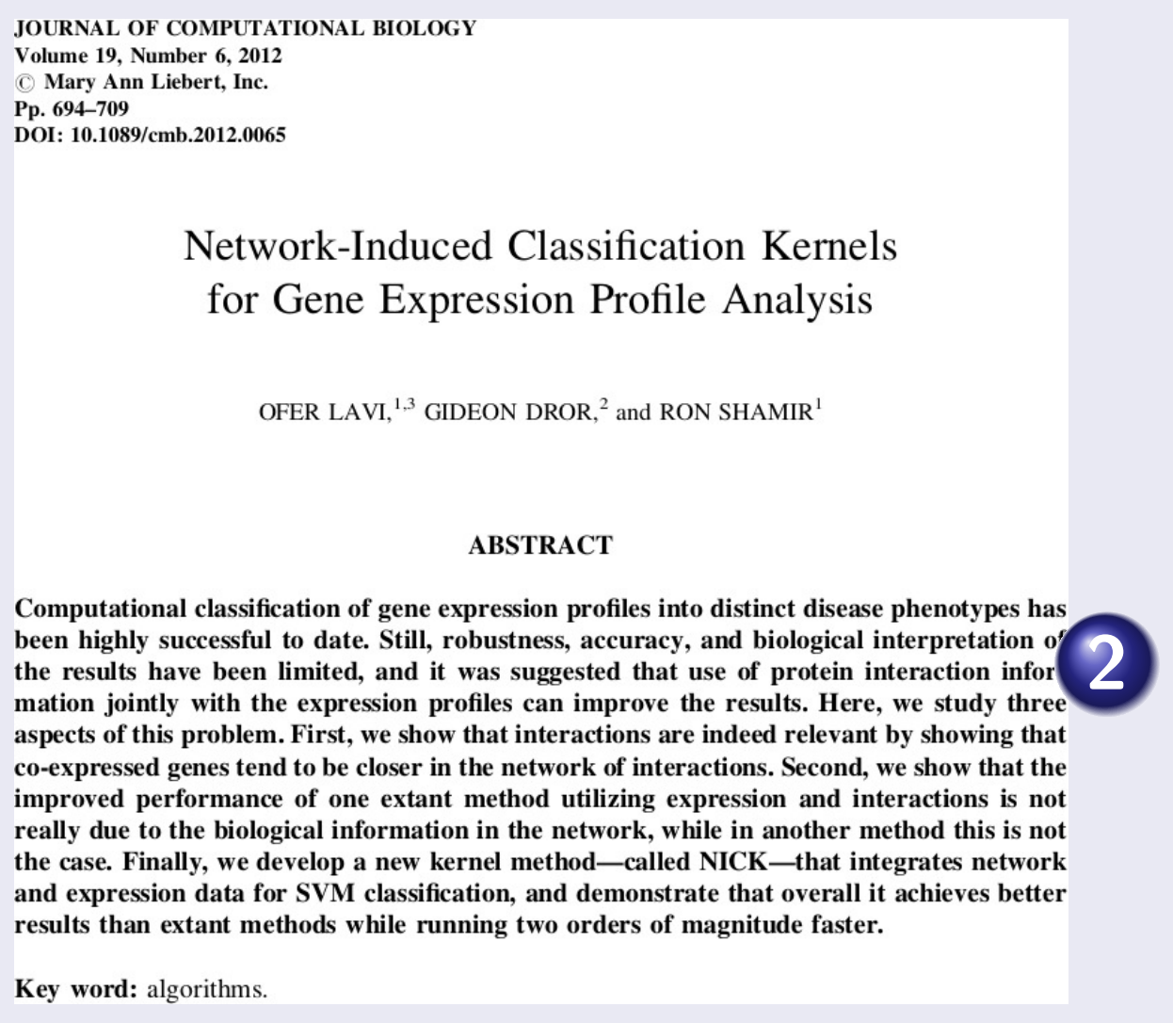
1. Input: Gene expression data 2. Output: Prognosis (Poor vs. Good), Metastases 3. Goal: Classify and find important genes 4. Issue: Hard to classify due to huge number of features (genes) compared to number of samples ($\sim 22000 \gg 98$)
- Laura J. van 't Veer et.al. Nature, (2002)



Yeast protein interaction network. The colour of a node indicates the phenotypic effect of removing the corresponding protein (red = lethal, green = non-lethal, orange = slow growth, yellow = unknown), A. BarabÃ¡si, Z. Oltvai, Nature Reviews Genetics, (2004)



Method



JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 19, Number 6, 2012
© Mary Ann Liebert, Inc.
Pp. 694–709
DOI: 10.1089/cmb.2012.0065

Network-Induced Classification Kernels for Gene Expression Profile Analysis

OFER LAVI,[1,3] GIDEON DROR,[2] and RON SHAMIR[1]

ABSTRACT

Computational classification of gene expression profiles into distinct disease phenotypes has been highly successful to date. Still, robustness, accuracy, and biological interpretation of the results have been limited, and it was suggested that use of protein interaction information jointly with the expression profiles can improve the results. Here, we study three aspects of this problem. First, we show that interactions are indeed relevant by showing that co-expressed genes tend to be closer in the network of interactions. Second, we show that the improved performance of one extant method utilizing expression and interactions is not really due to the biological information in the network, while in another method this is not the case. Finally, we develop a new kernel method—called NICK—that integrates network and expression data for SVM classification, and demonstrate that overall it achieves better results than extant methods while running two orders of magnitude faster.

Key word: algorithms.

1. It's shown:
   - Co-expressed genes tend to be close in the PPI-Network.
   - Exploit this fact to modify the SVM objective function - called NICK
2. What can be done:
   - Reverse engineer the learned machine to extract important genes after using the network information.

NICK

1. SVM modified objective function

$\min_{\mathbf{w},w_0}\left\{\frac{1}{2}\|\mathbf{w}\|^2+\frac{1}{2}\beta\sum_{(j,k)\in E}(w_j-\right.$

s.t.:

$\forall i\in\{1,\cdots,n\}:(\mathbf{w}\mathbf{x}_i+w_0)y_i\geq 1$

2. Dual Problem

$$\max_{\alpha}\left\{\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i^T\mathbf{L})(\mathbf{L}^T\mathbf{x}_j)\right\}$$

$\mathbf{L}\mathbf{L}^T=(\mathbf{I}+\beta\mathbf{B})^{-1}$

s.t.:

$$\forall i\in\{1,\cdots,n\}:\sum_{i=1}^{n}\alpha_i y_i=0$$

$\forall i\in\{1,\cdots,n\}:\alpha_i\geq 0$

Laplacian matrix:

$\mathbf{B}=\mathbf{D}-\mathbf{A}$

3. Dual to Primal

$\mathbf{w}=(\mathbf{I}+\beta\mathbf{B})^{-1}\sum_{i=1}^{n}\alpha_i y_i\mathbf{x}_i$

Synthesize Data

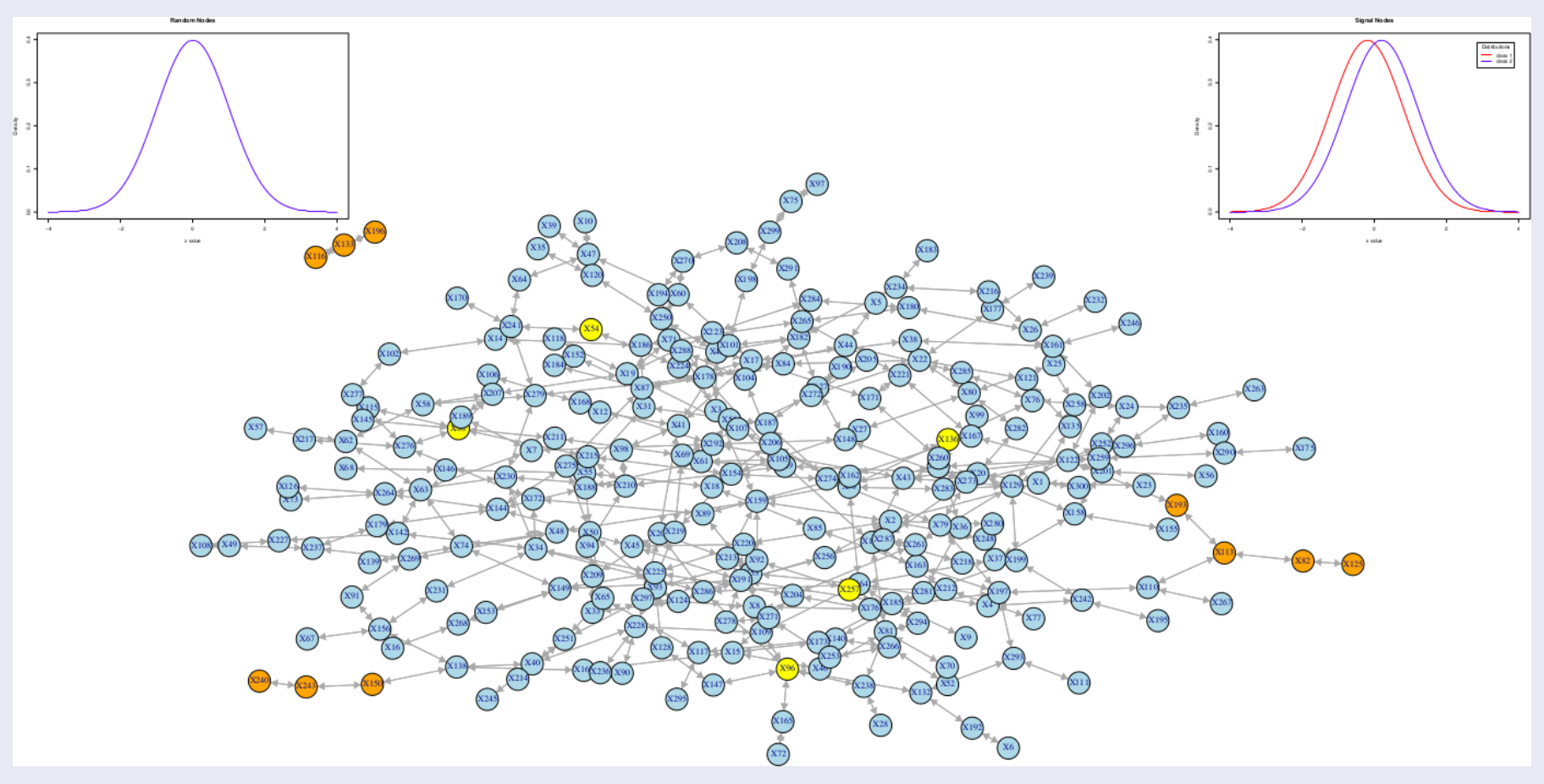
1. A random graph (PPI-Network)
2. Signal nodes (genes):

$$f(n)=\begin{cases}N(-\mu,1) & \text{if } n \text{ is in class } 1\\ N(\mu,1) & \text{if } n \text{ is in class } 2\end{cases}$$

3. Random nodes (non-informative genes):

$$f(n)=N(0,1)$$

4. Pathway: 2, 3, or 4 connected signal nodes.



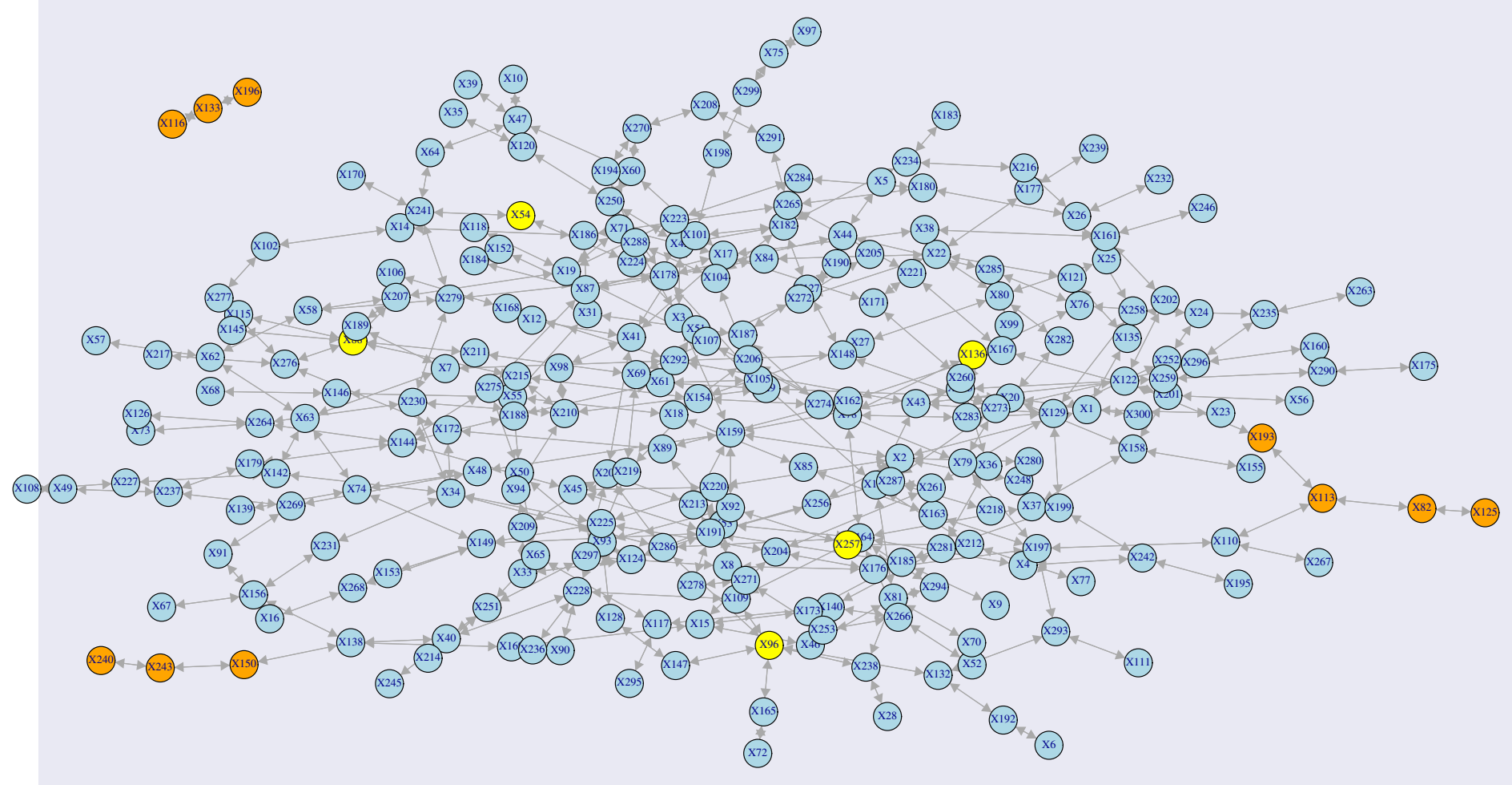
Blue: random gene, Orange: Signal node being a member of a pathway of signal nodes, Yellow: A lonely signal node

- Solve SVM problem for original and transformed data.
- Calculate $\mathbf{w}$ for both models.
- Compute for each pair of nodes, for each model:

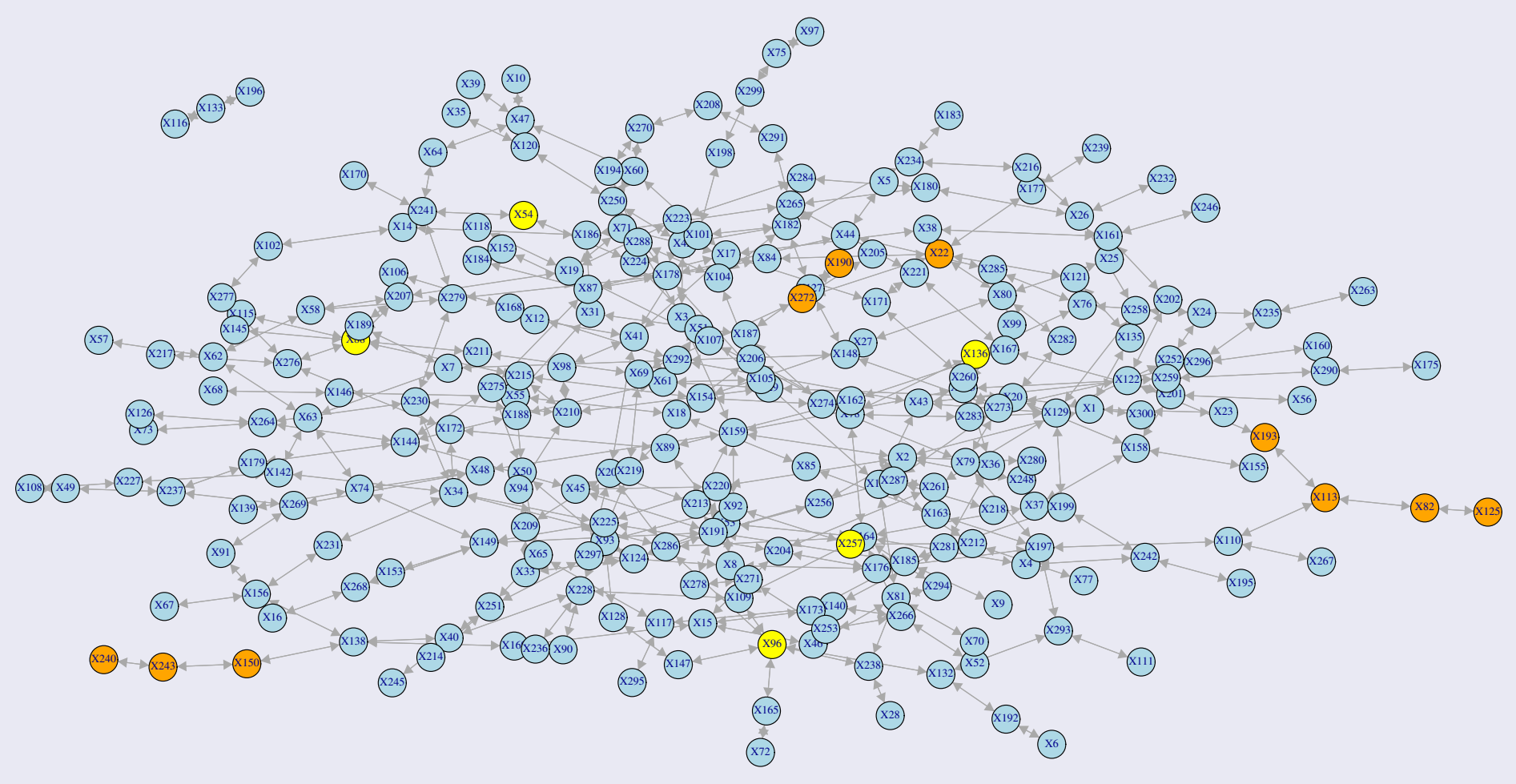
$$Score(i,j)=\frac{|w_i|+|w_j|}{2}\times e^{max(d_G(i,j),1)}$$

- Report pairs with highest scores for both trained models.

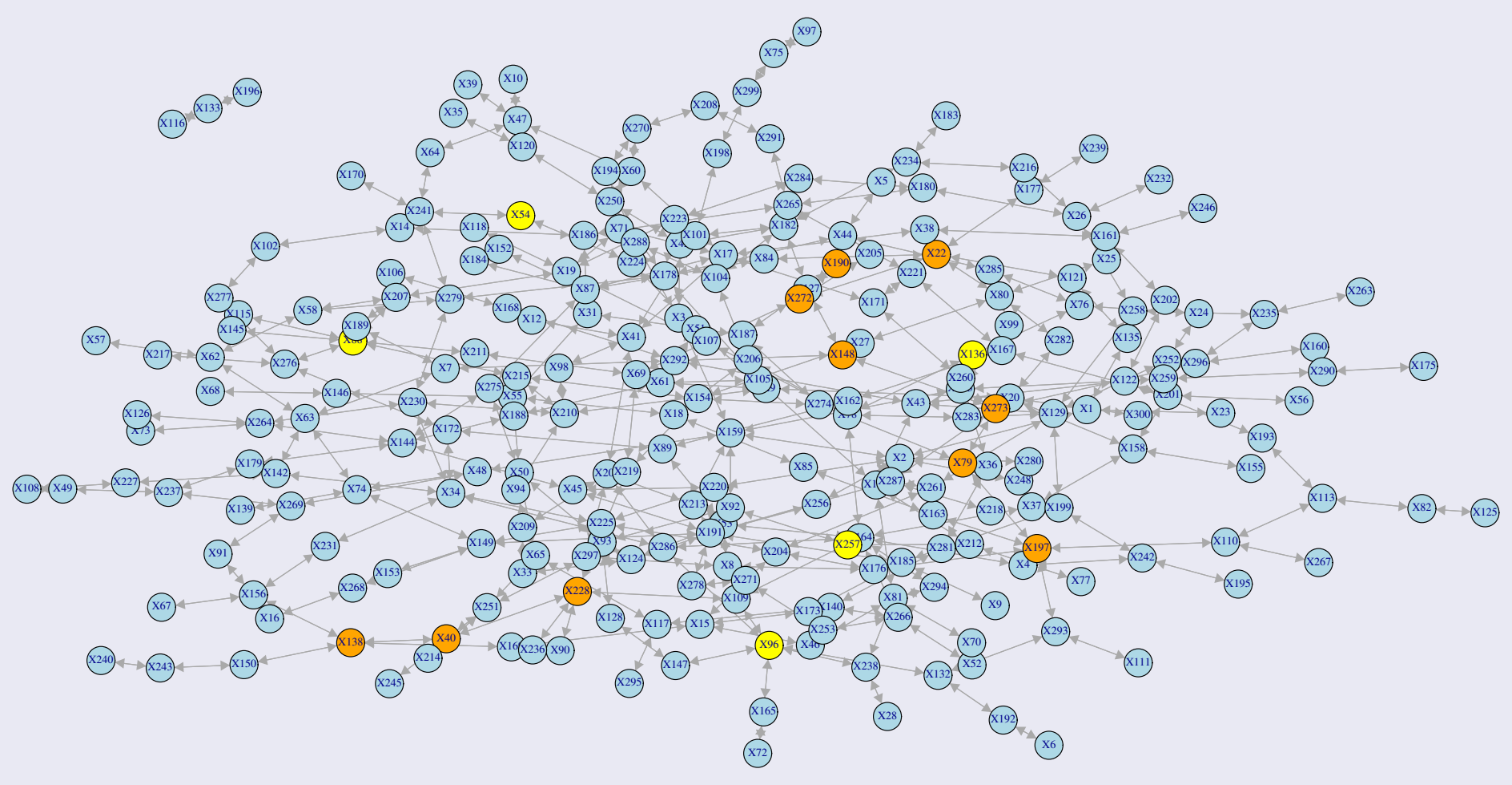
Results



Easy



Medium



Hard

Easy

Original			
X196	X196	X53	X53
X233	X233	X39	X39
X88	X88	X196	X133
X116	X116	X127	X127
X197	X197	X127	X148
X148	X148	X150	X150
X148	X273	X116	X133

Transformed			
X196	X196	X233	X233
X196	X133	X133	X133
X133	X116	X116	X116
X95	X95	X240	X240
X39	X39	X240	X243
X59	X59	X106	X106
X243	X243	X106	X168

Medium

Original			
X190	X190	X104	X104
X233	X233	X190	X272
X277	X277	X88	X88
X190	X127	X165	X165
X272	X272	X272	X22
X106	X106	X165	X96
X150	X150	X250	X250

Transformed			
X233	X233	X190	X190
X112	X112	X240	X240
X190	X272	X240	X243
X86	X86	X243	X243
X243	X150	X190	X127
X150	X150	X272	X272
X246	X246	X298	X298

Hard

Original			
X190	X190	X101	X101
X233	X233	X190	X272
X88	X88	X297	X297
X190	X127	X93	X93
X26	X26	X138	X138
X272	X272	X272	X22
X101	X41	X123	X123

Transformed			
X233	X233	X190	X190
X112	X112	X190	X272
X86	X86	X190	X127
X272	X272	X272	X205
X205	X205	X146	X146
X146	X68	X68	X68
X298	X298	X272	X22

Easy

AUC (Original):	60.6
AUC (Transformed):	62.4
wc p-value (paired):	5.669e-09

Medium

AUC (Original):	60.1
AUC (Transformed):	61.5
wc p-value (paired):	1.383e-06

Hard

AUC (Original):	60.6
AUC (Transformed):	62.4
wc p-value (paired):	5.669e-09