

6.874, 6.802, 20.390, 20.490, HST.506

Computational Systems Biology

Deep Learning in the Life Sciences

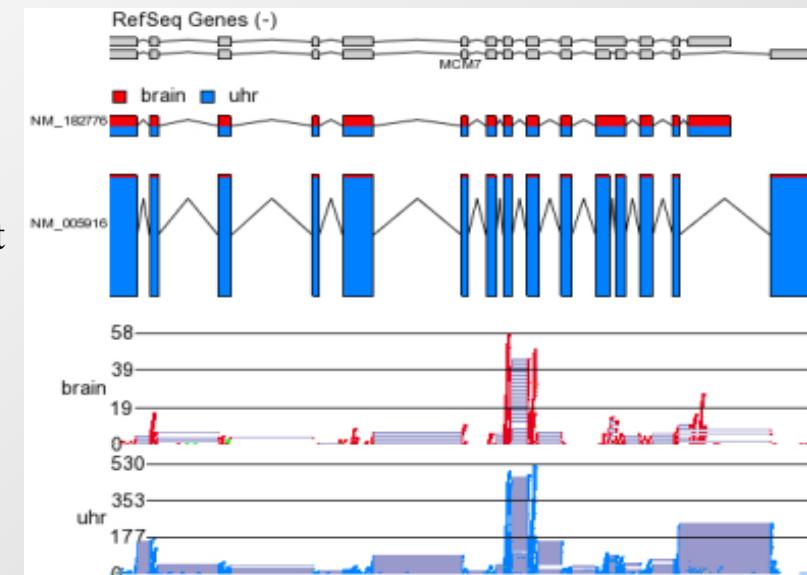
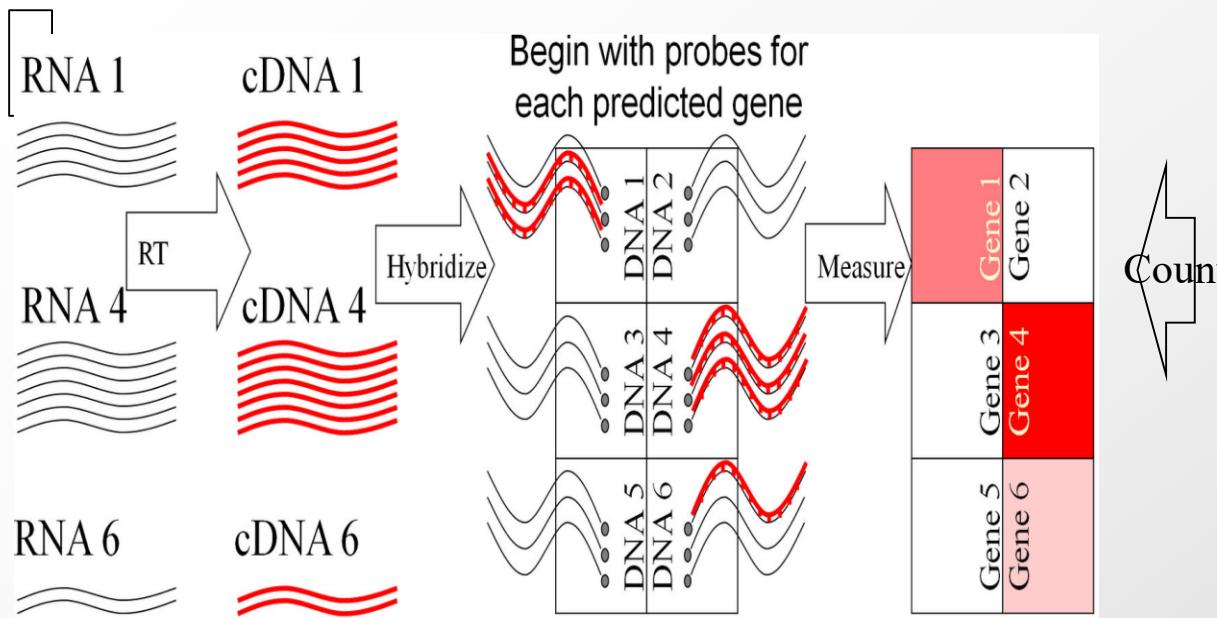
# Lecture 12: Predicting gene expression and splicing

Prof. Manolis Kellis

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

# RNA-Seq: De novo tx reconstruction / quantification



## Microarray technology

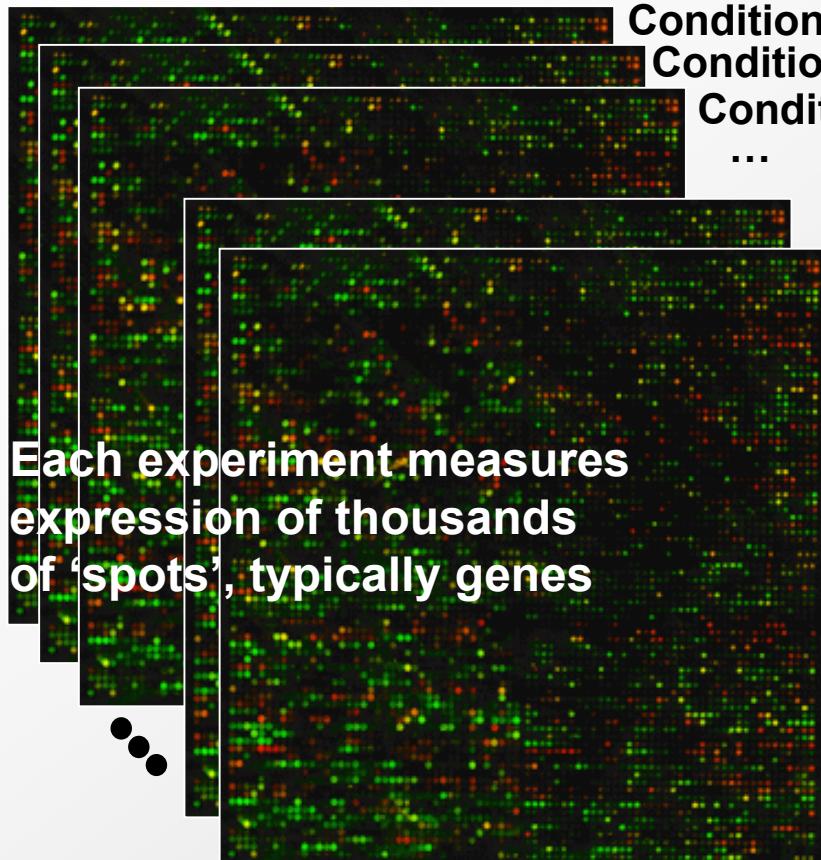
- Synthesize DNA probe array, complementary hybridization
- Variations:
  - One long probe per gene
  - Many short probes per gene
  - Tiled k-mers across genome
- Advantage:
  - Can focus on small regions, even if few molecules / cell

## RNA-Seq technology:

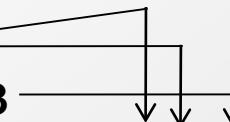
- Sequence short reads from mRNA, map to genome
- Variations:
  - Count reads mapping to each known gene
  - Reconstruct transcriptome *de novo* in each experiment
- Advantage:
  - Digital measurements, *de novo*

# Expression Analysis Data Matrix

- Measure 20,000 genes in 100s of conditions



Condition  
Condition 2  
Condition 3  
...



**n experiments**

**m genes**

Expression profile of a gene

Gene similarity questions

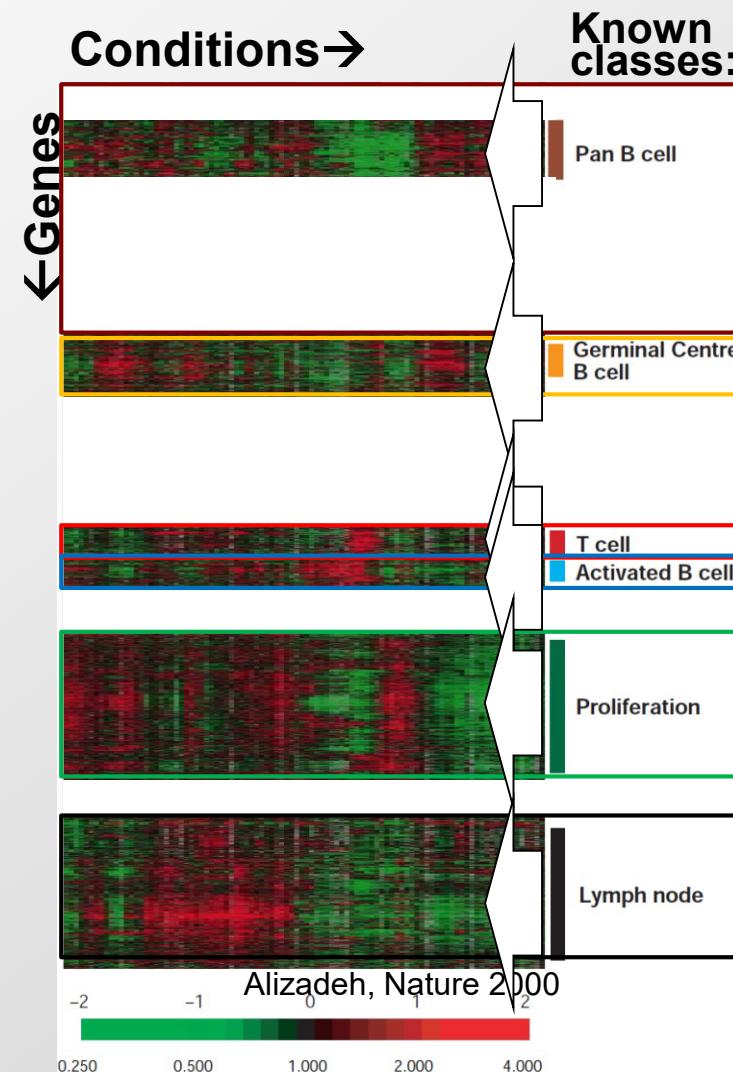
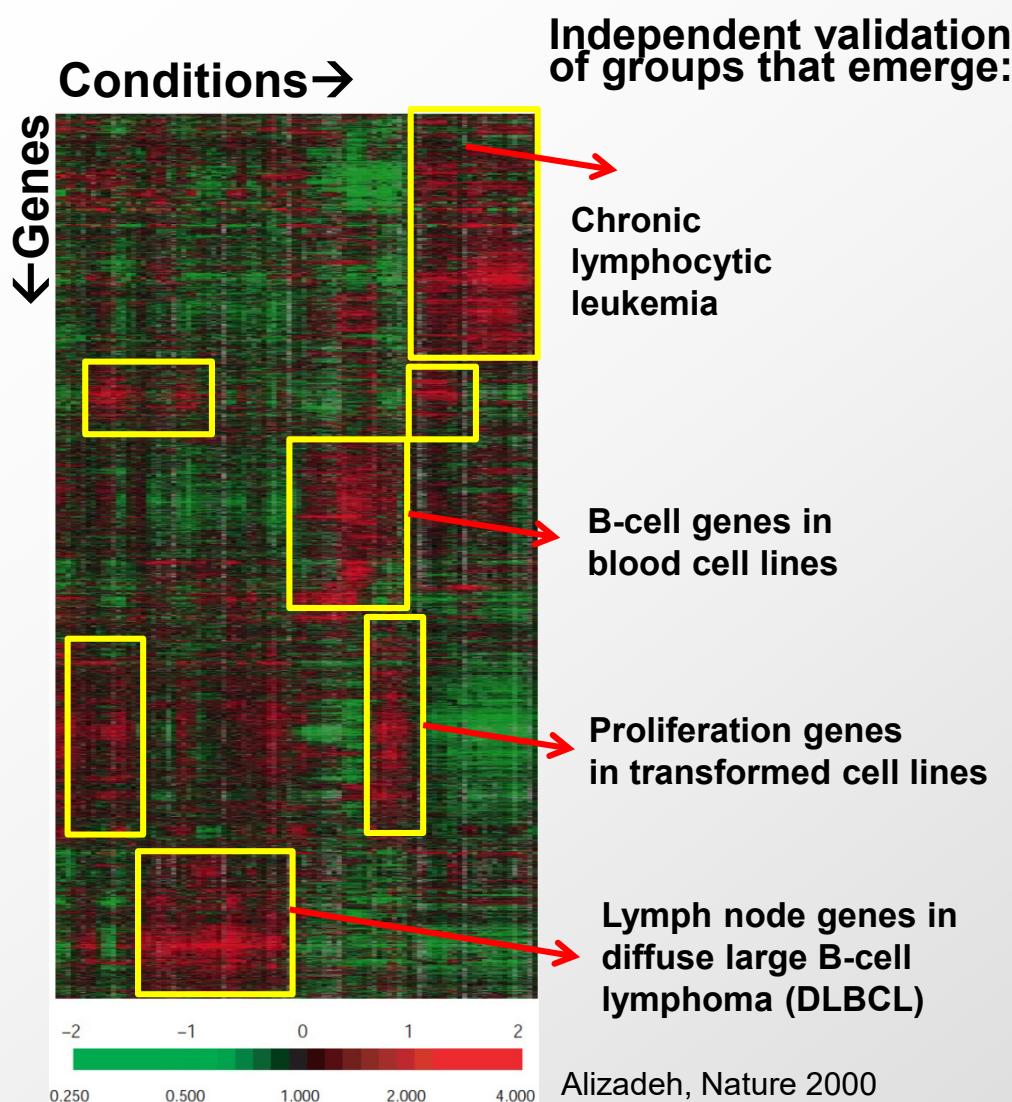
Experiment similarity questions

- Study resulting matrix

# Clustering

vs.

# Classification



**Goal of Clustering:** Group similar items that likely come from the same category, and in doing so reveal hidden structure

- Unsupervised learning

**Goal of Classification:** Extract features from the data that best assign new elements to  $\geq 1$  of well-defined classes

- Supervised learning

# PCA, Dimensionality reduction

Figure 1A

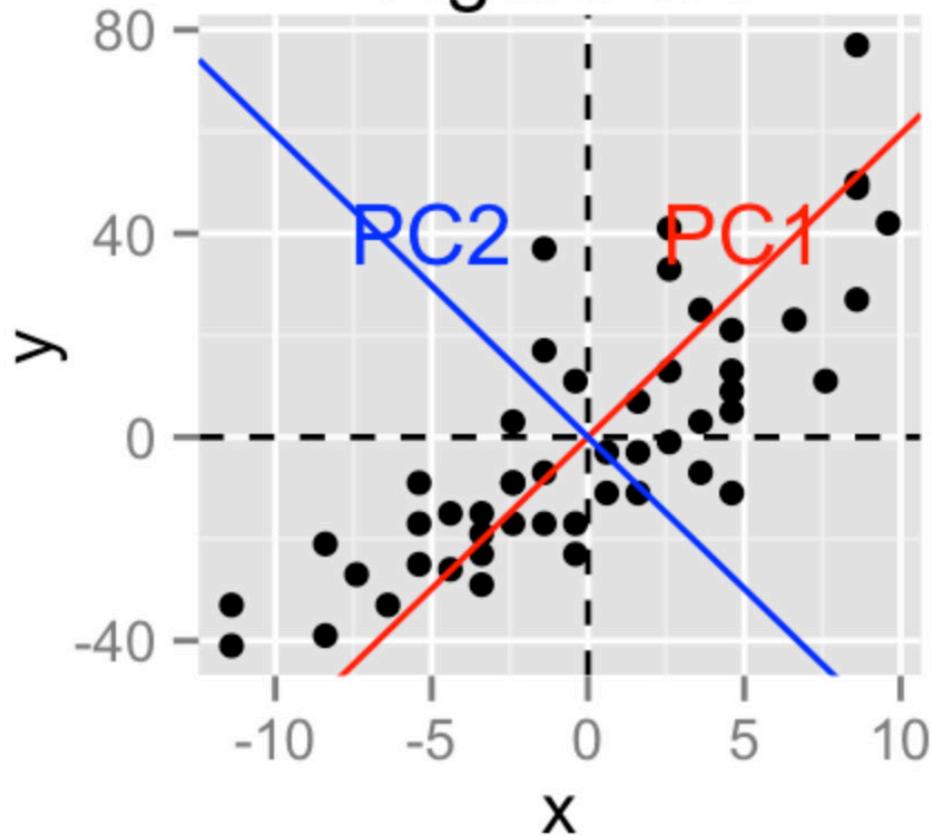
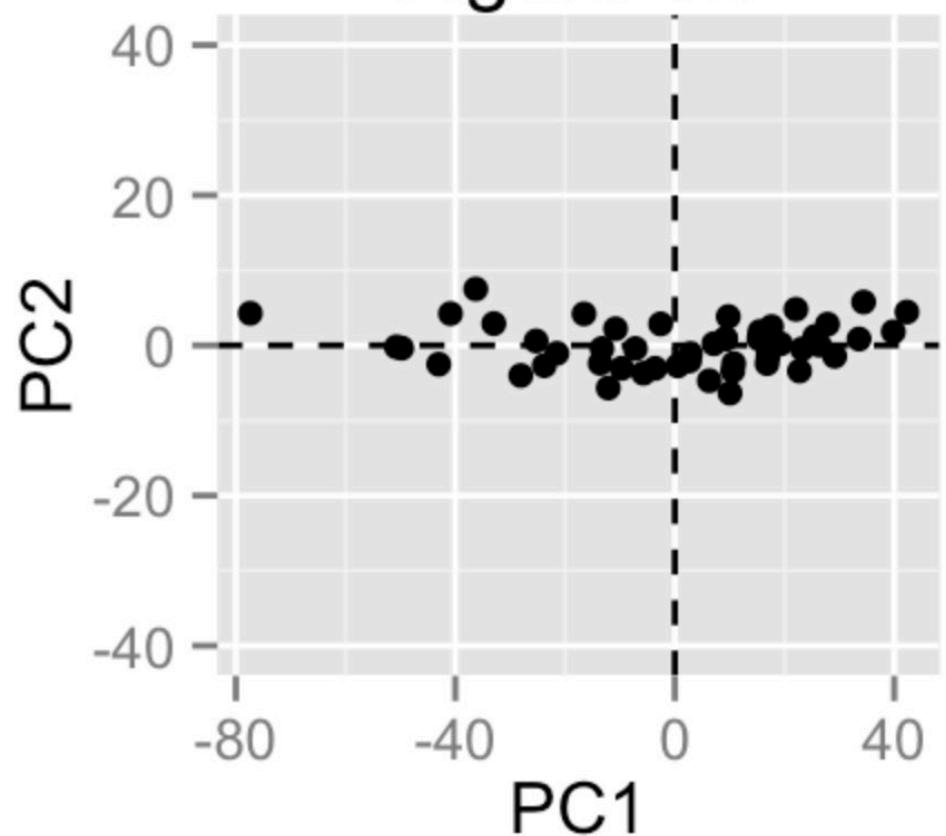
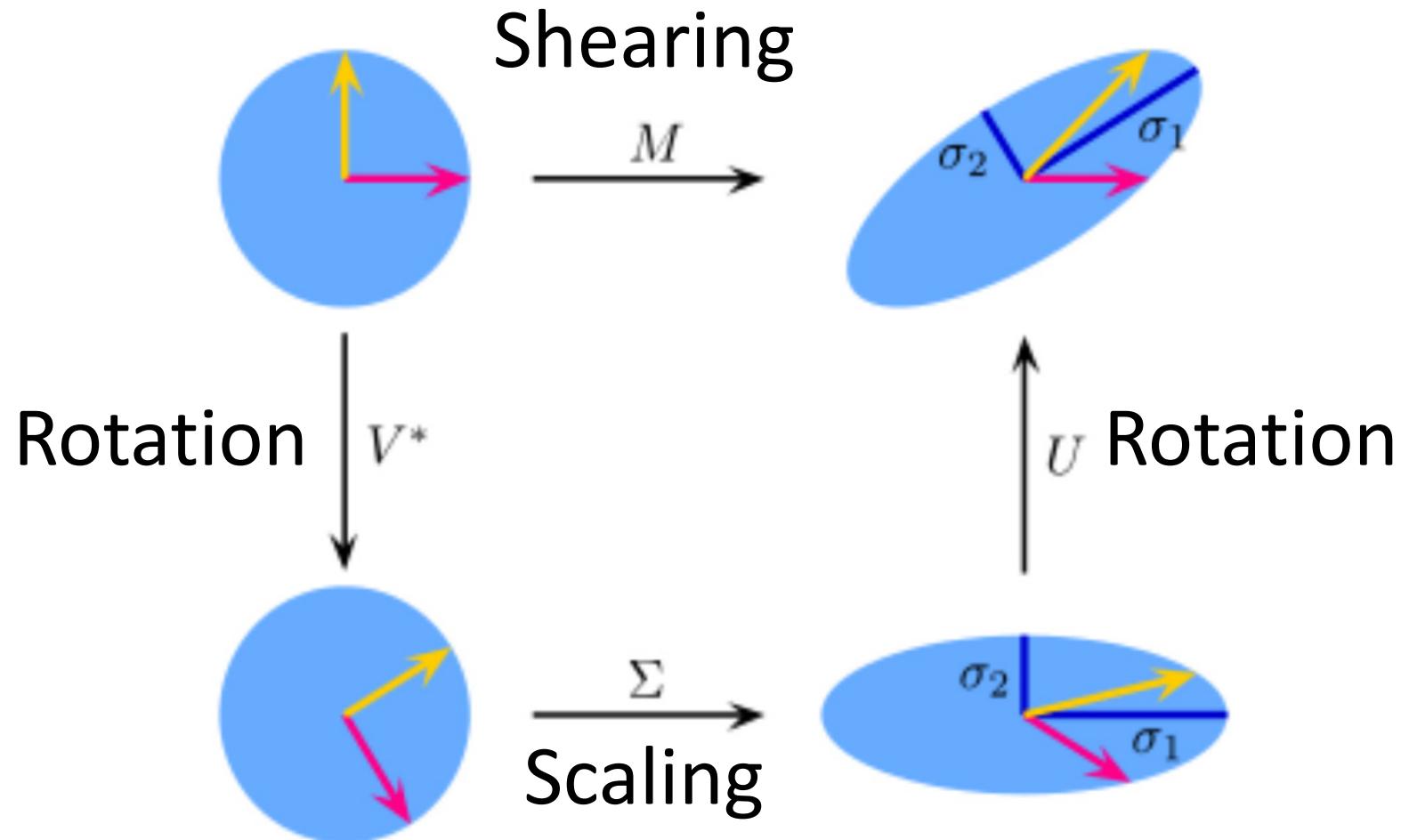


Figure 1B



# Geometric interpretation of SVD



$$M = U \cdot \Sigma \cdot V^*$$

$$Mx = M(x) = U( S( V^*(x) ) )$$

# Low-rank Approximation

- Solution via SVD

$$A_k = U \operatorname{diag}(\sigma_1, \dots, \sigma_k, \underbrace{0, \dots, 0}_{\text{set smallest } r-k \text{ singular values to zero}}) V^T$$

*set smallest  $r-k$  singular values to zero*

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{A_k} = \underbrace{\begin{bmatrix} \star & \star & \color{blue}{\star} \\ \star & \star & \color{blue}{\star} \\ \star & \star & \color{blue}{\star} \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & & \\ & \bullet & & \\ & & \color{blue}{\bullet} & \\ & & & \color{brown}{\bullet} \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \\ \color{blue}{\star} & \color{blue}{\star} & \color{blue}{\star} & \color{blue}{\star} & \color{blue}{\star} \\ \star & \star & \star & \star & \star \\ \star & \star & \star & \star & \star \end{bmatrix}}_{V^T}$$

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T \quad \xleftarrow{\text{column notation: sum of rank 1 matrices}}$$

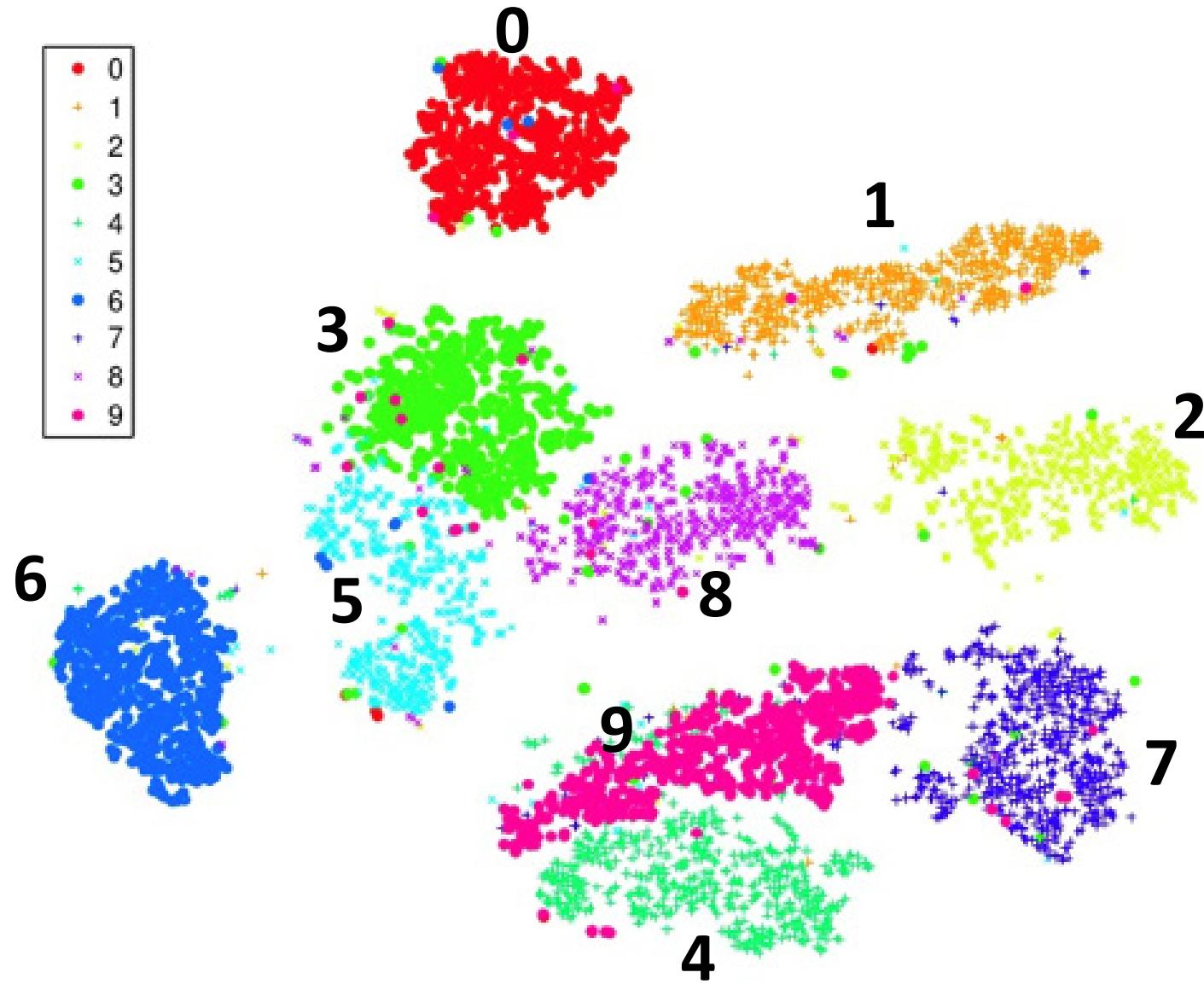
- Error:  $\min_{X: \operatorname{rank}(X)=k} \|A - X\|_F = \|A - A_k\|_F = \sigma_{k+1}$

# PCA of MNIST digits

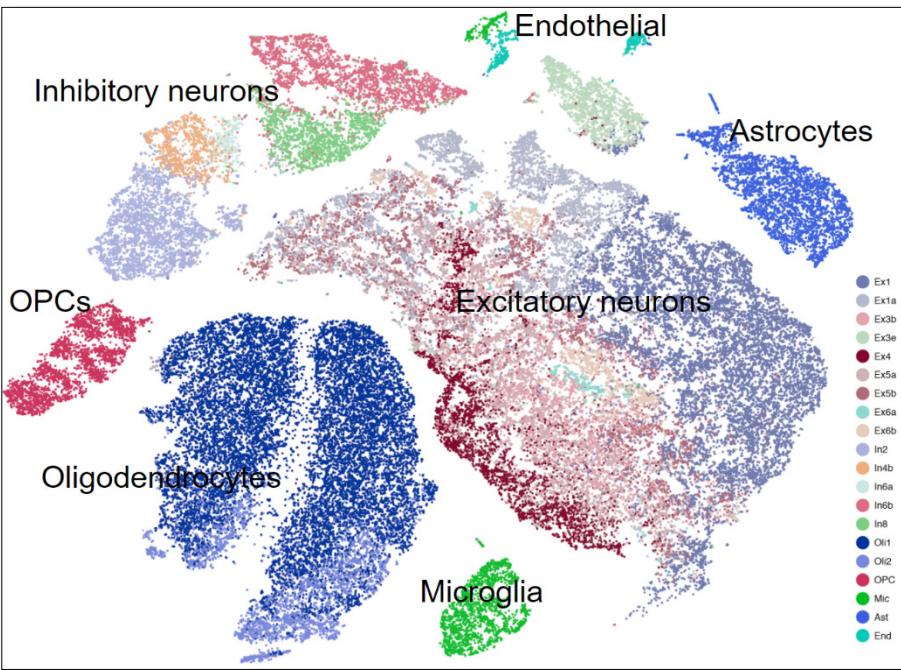
3 6 8 1 7 9 6 6 4 1  
6 7 5 7 8 6 3 4 8 5  
2 1 7 9 7 1 2 1 4 5  
4 8 1 9 0 1 8 3 9 4  
7 6 1 8 1 4 1 5 1 0  
7 5 9 2 6 5 8 1 9 7  
1 2 2 2 2 3 4 4 8 0  
0 2 3 8 0 7 3 8 5 7  
0 1 4 6 4 6 0 2 8 3  
7 1 2 8 7 6 9 8 6 1



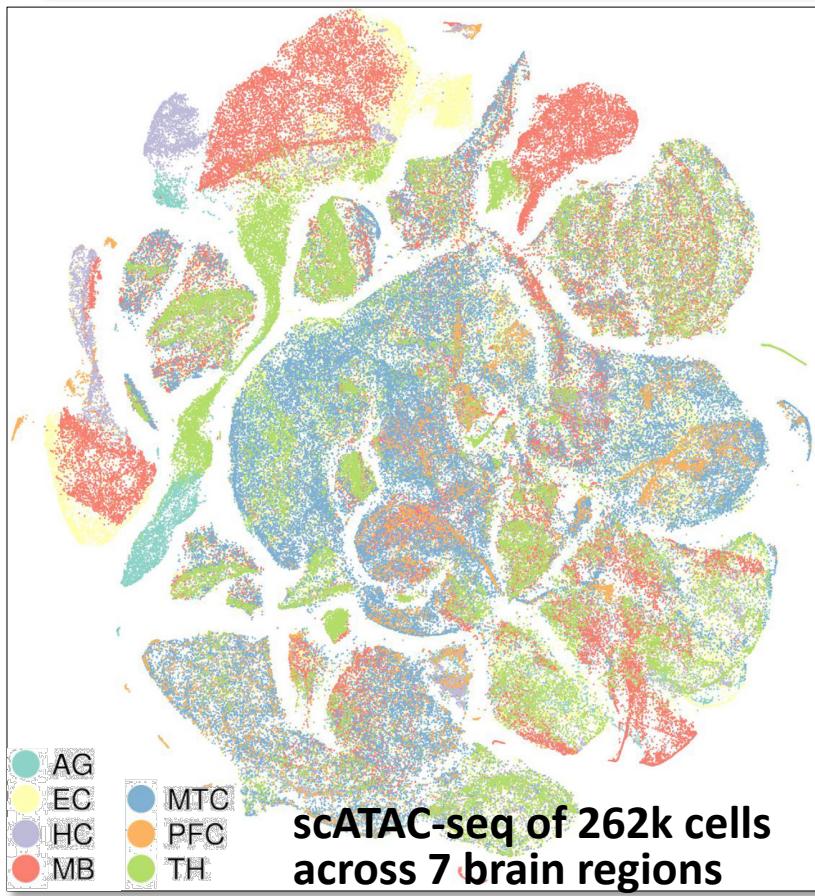
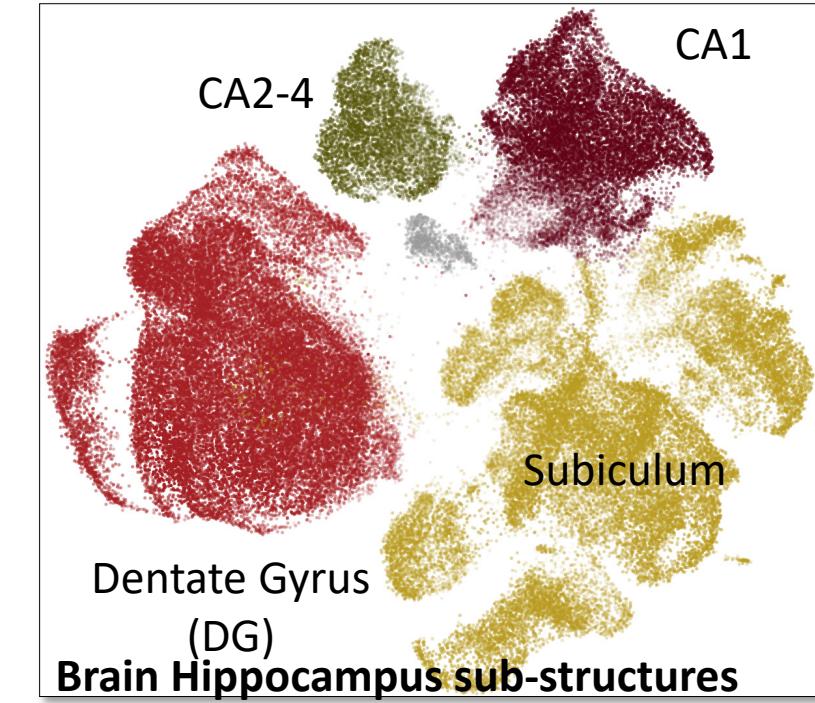
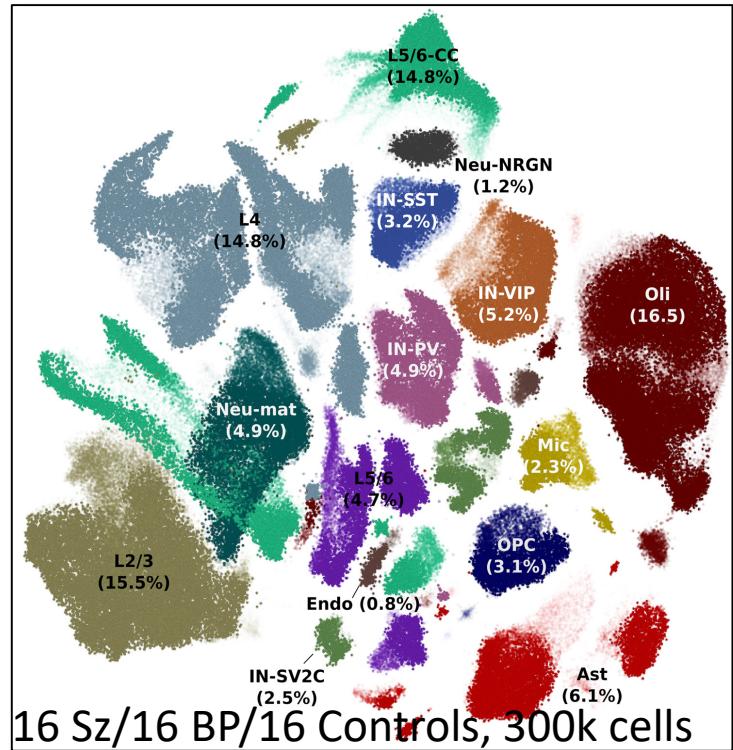
# t-SNE of MNIST digits



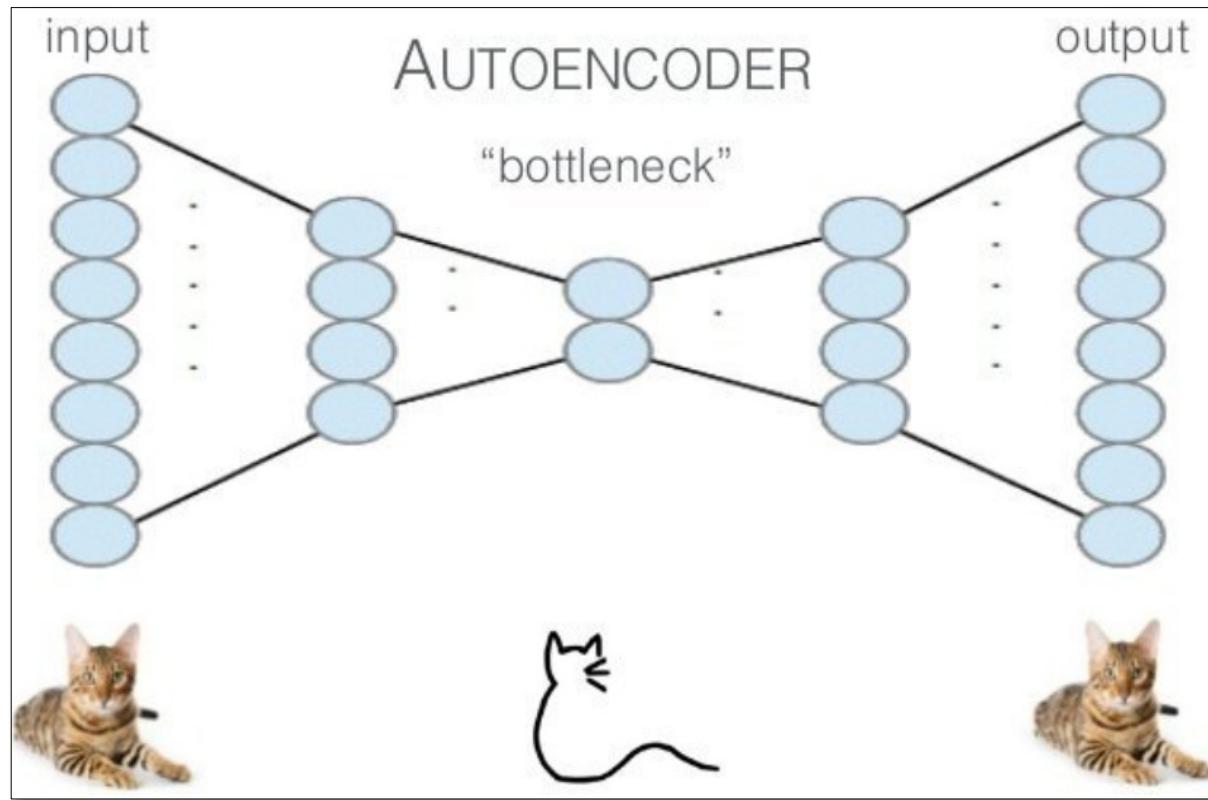
# t-SNEs of single-cell Brain data



scRNA-seq in 48 individuals, 84k cells, Nature, 2019



# Autoencoder: dimensionality reduction with neural net



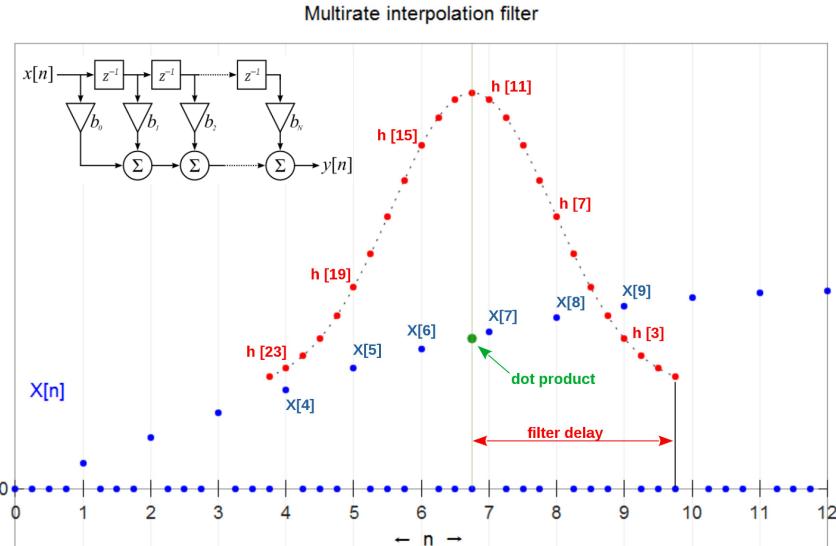
- Tricking a **supervised** learning algorithm to work in **unsupervised** fashion
- Feed input as output function to be learned. **But!** Constrain model complexity
- **Pretraining** with RBMs to learn representations for future supervised tasks. Use RBM output as “data” for training the next layer in stack
- After pretraining, “unroll” RBMs to create deep autoencoder
- Fine-tune using backpropagation

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

# 1. Up-sampling gene expression patterns

# Challenge: Measure few values, infer many values



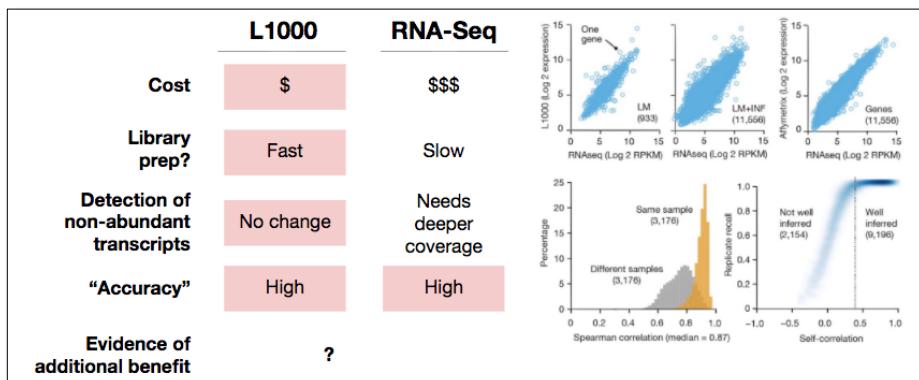
- Digital signal upscaling
  - Interpolating low-pass filter (e.g. FIR finite impulse response)
  - Low-dim. capture of higher-dim. signal
  - Nyquist rate (discrete) / freq. (contin.)



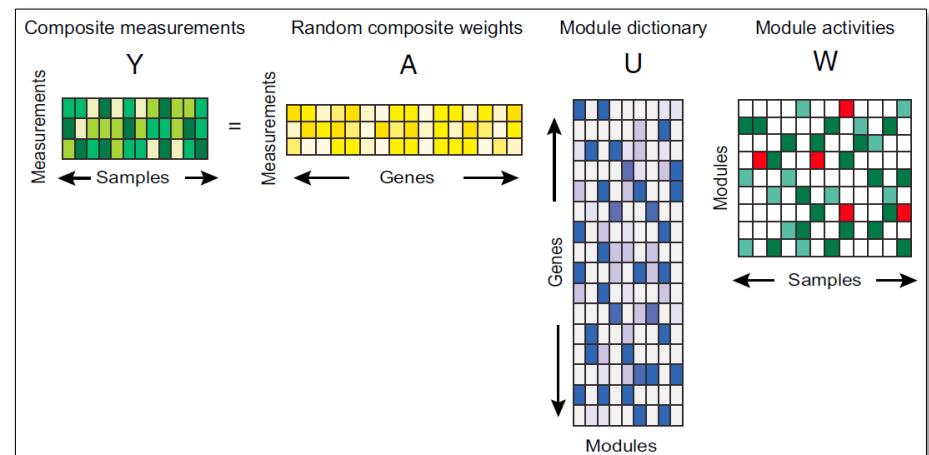
<https://arxiv.org/pdf/1902.06068.pdf>



- Image up-scaling
  - Inverse of convolution (de-convolution)
  - Transfer learning from corpus of images
  - Low-dim. re-projection to high-dim. img

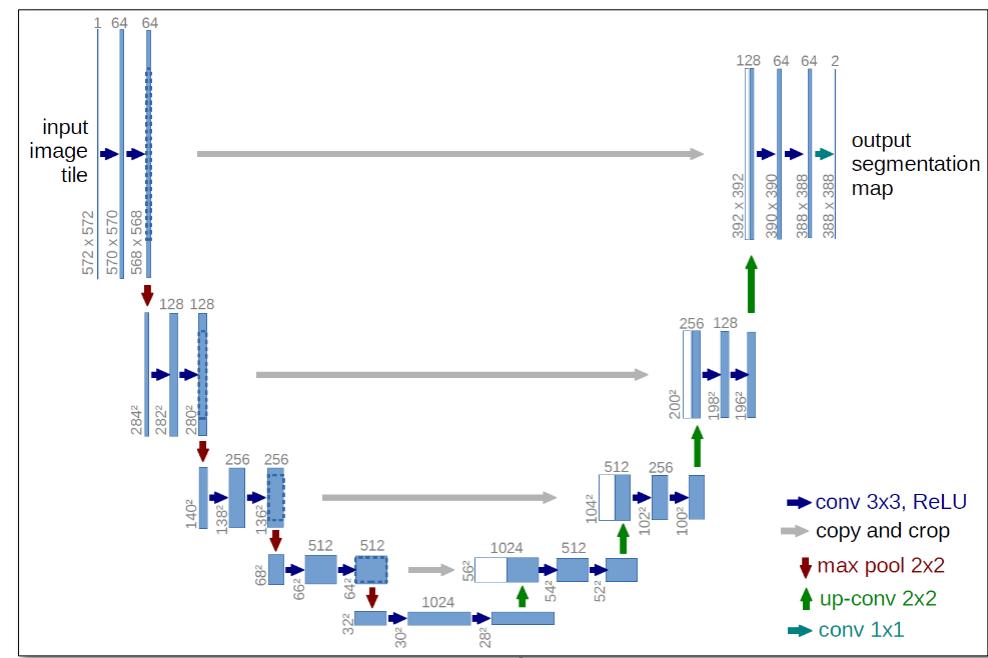


- Gene expression measurements
  - Measure 1000 genes, infer the rest
  - Rapid, cheap, reference assay
  - Apply to millions of conditions

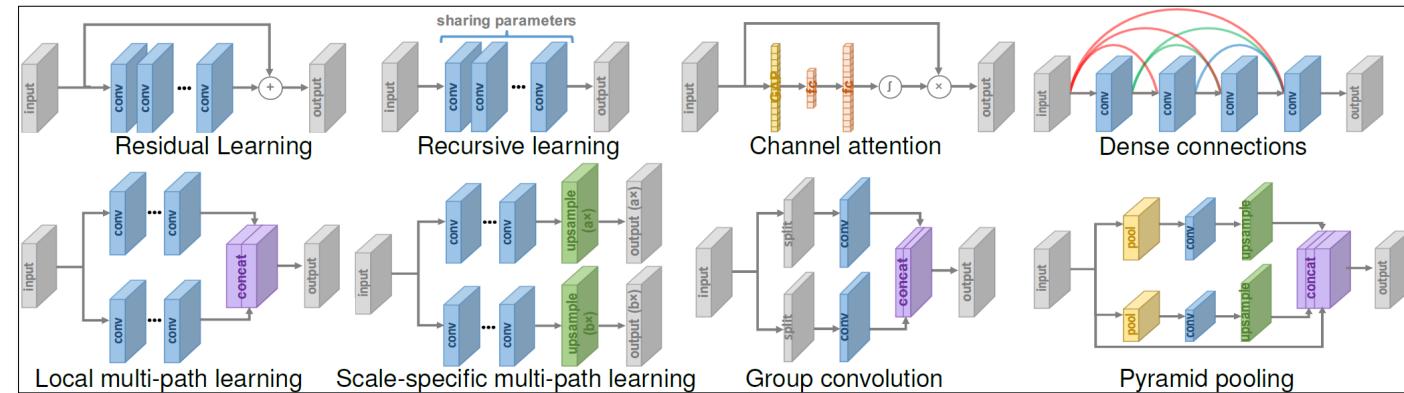
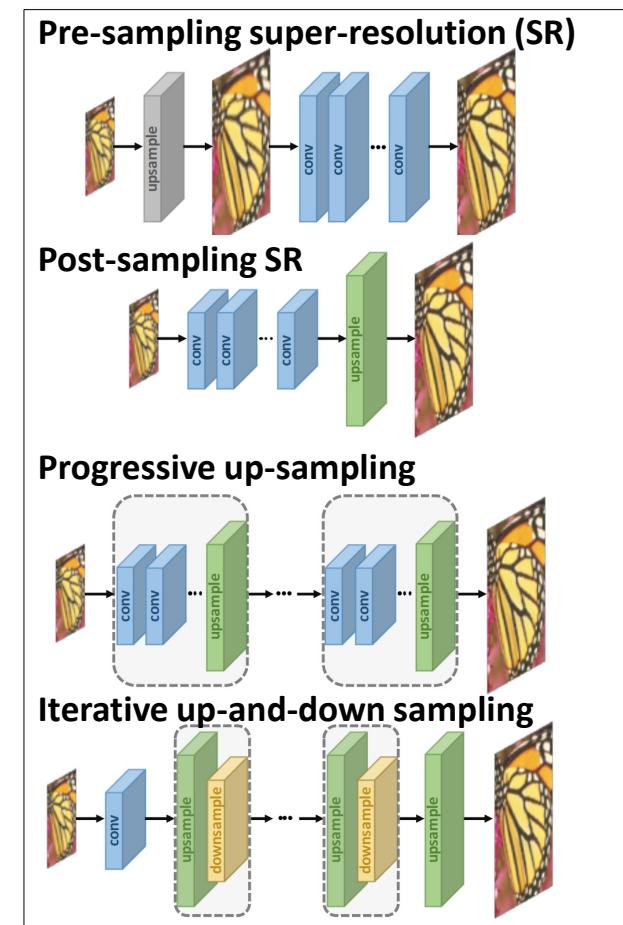


- **Which** 1000 genes? Compressed sensing
  - Measure few **combinations** of genes
  - Better capture high-dimensional vector

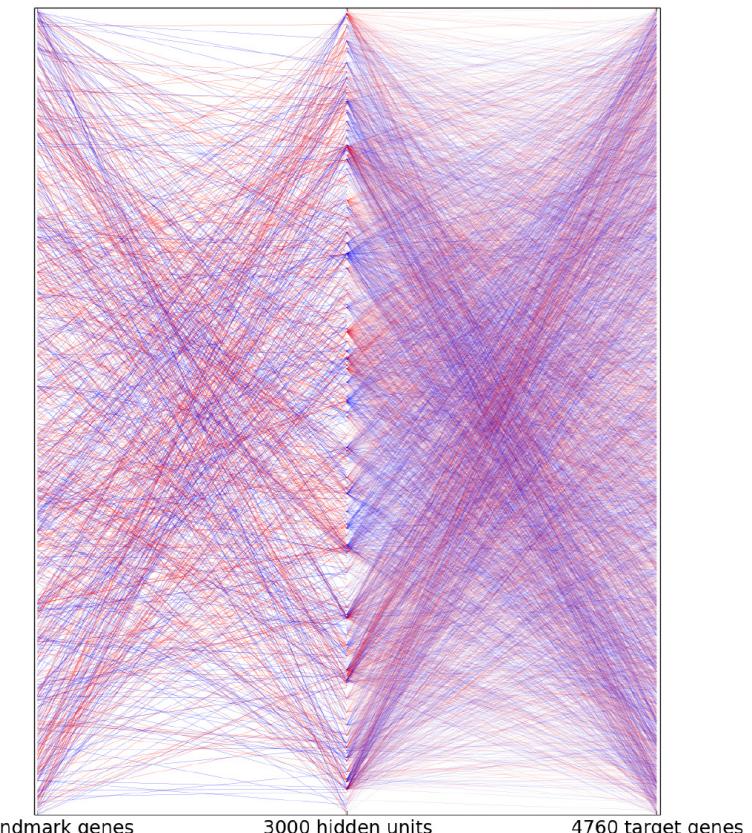
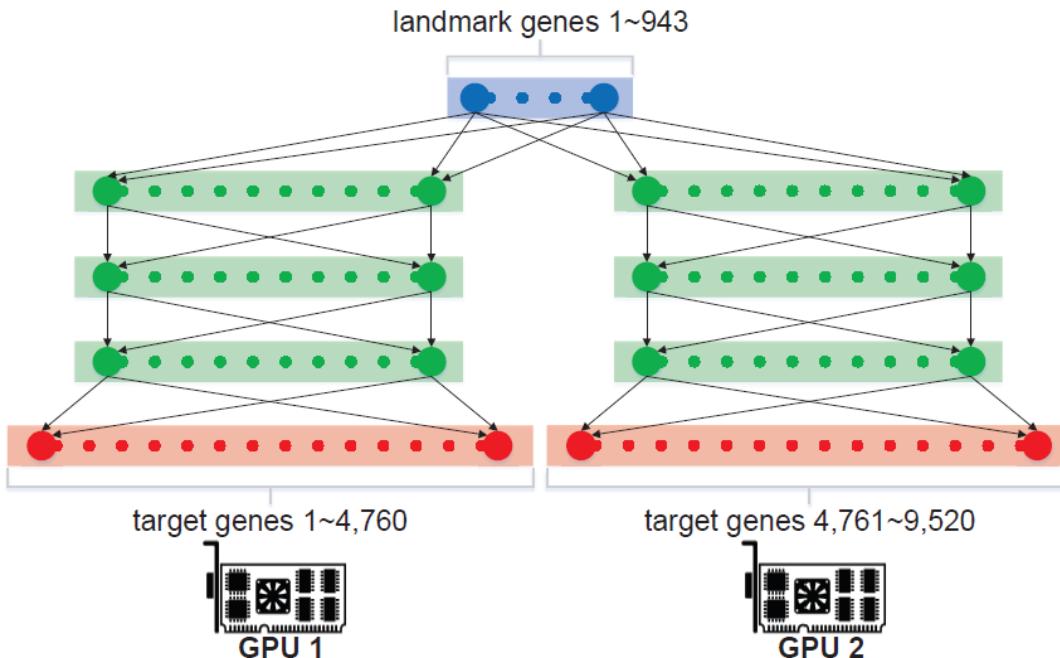
# Deep Learning architectures for up-sampling images



- Representation/abstract learning
  - Enables compression, re-upscaling, denoising
  - Example: autoencoder bottleneck. High-low-high
  - Modification: de-compression, up-scaling, low-high only



# D-GEX - Deep Learning for up-scaling L1000 gene expression



## Parameters

# of hidden layers	[1, 2, 3]
# of hidden units in each hidden layer	[3000, 6000, 9000]
Dropout rate	[0%, 10%, 25%]
Momentum coefficient	0.5
Initial learning rate <sup>a</sup>	5e-4 or 3e-4
Minimum learning rate	1e-5
Learning rate decay factor	0.9
Learning scale <sup>b</sup>	3.0
Mini-batch size	200
Training epoch	200
Weights initial range <sup>c</sup>	$\left[ -\frac{\sqrt{6}}{\sqrt{n_i+n_o}}, \frac{\sqrt{6}}{\sqrt{n_i+n_o}} \right]$

## Gene expression inference with deep learning FREE

Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, Xiaohui Xie ✉ [Author Notes](#)

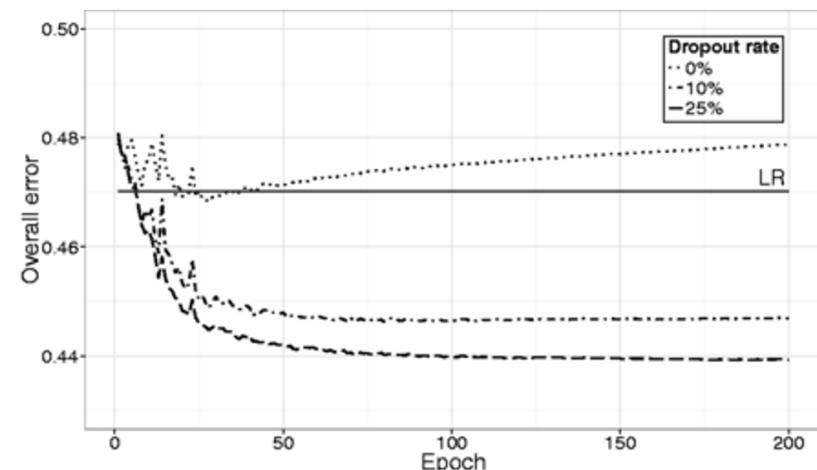
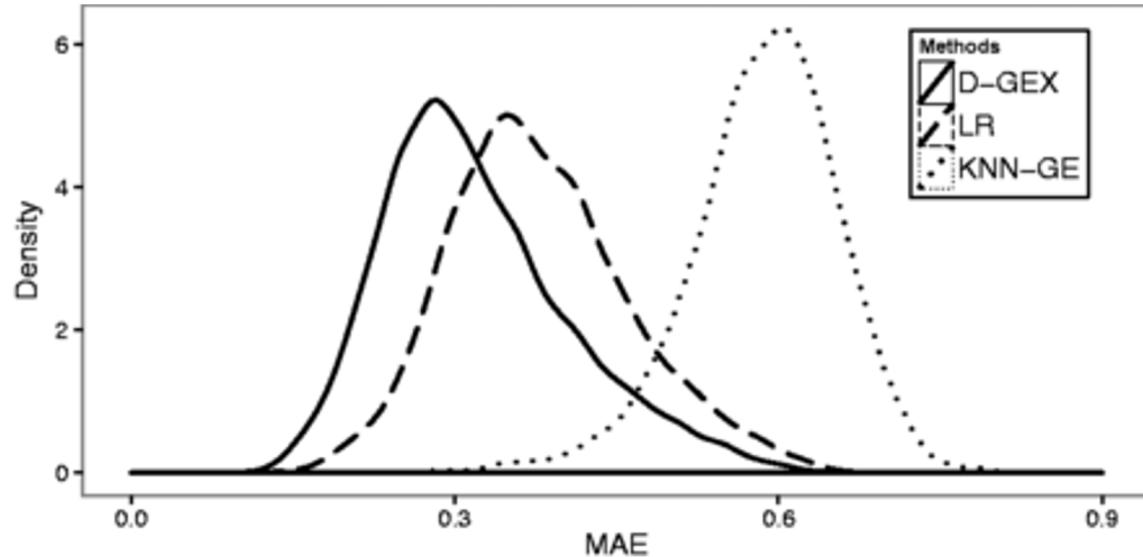
*Bioinformatics*, Volume 32, Issue 12, 15 June 2016, Pages 1832–1839,

<https://doi.org/10.1093/bioinformatics/btw074>

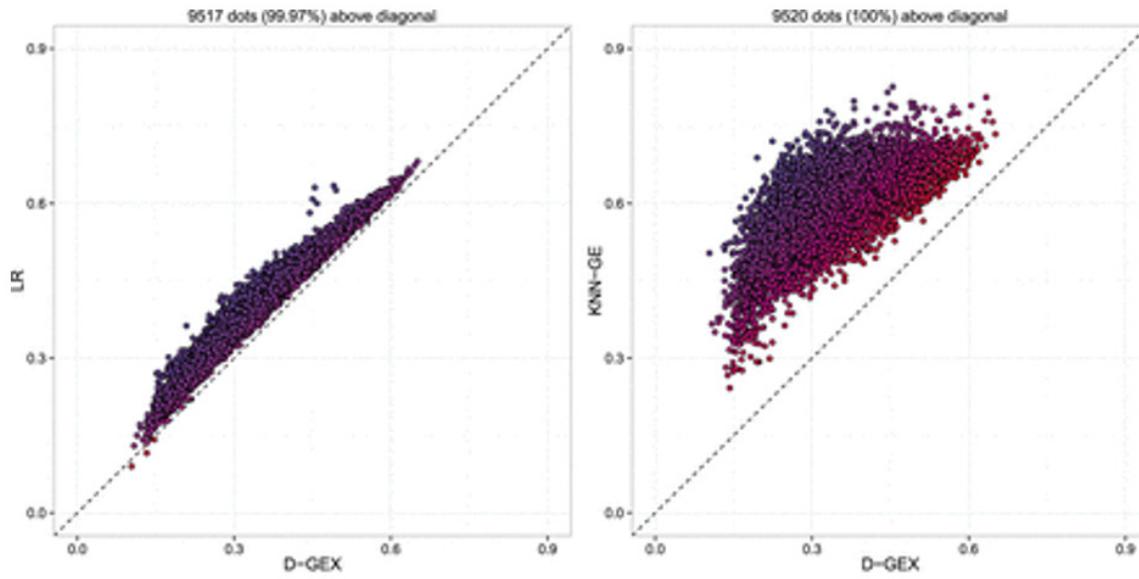
Published: 11 February 2016 Article history ▾

- Multi-task Multi-Layer Feed-Forward Neural Net
- Non-linear activation function (hyperbolic tangent)
- Input: 943 genes, Output: 9520 targets (partition to fit in memory)

# D-GEX outperforms Linear Regression or K-nearest-Neighbors

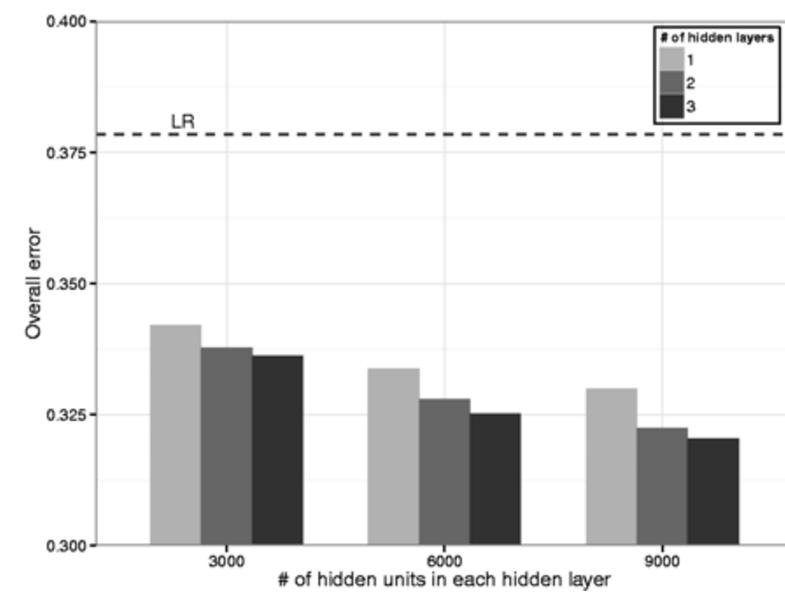


- Lower error than LR or KNN
- Training rapidly converges



- Strictly better for nearly all genes

However: performance still not great, computational limitations



- Deeper = better

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

## 2. Composite measurements for compressed sensing

# Key insight: Composite measurements better capture modules

Cell

## Theory Efficient Generation of Transcriptomic Profiles by Random Composite Measurements

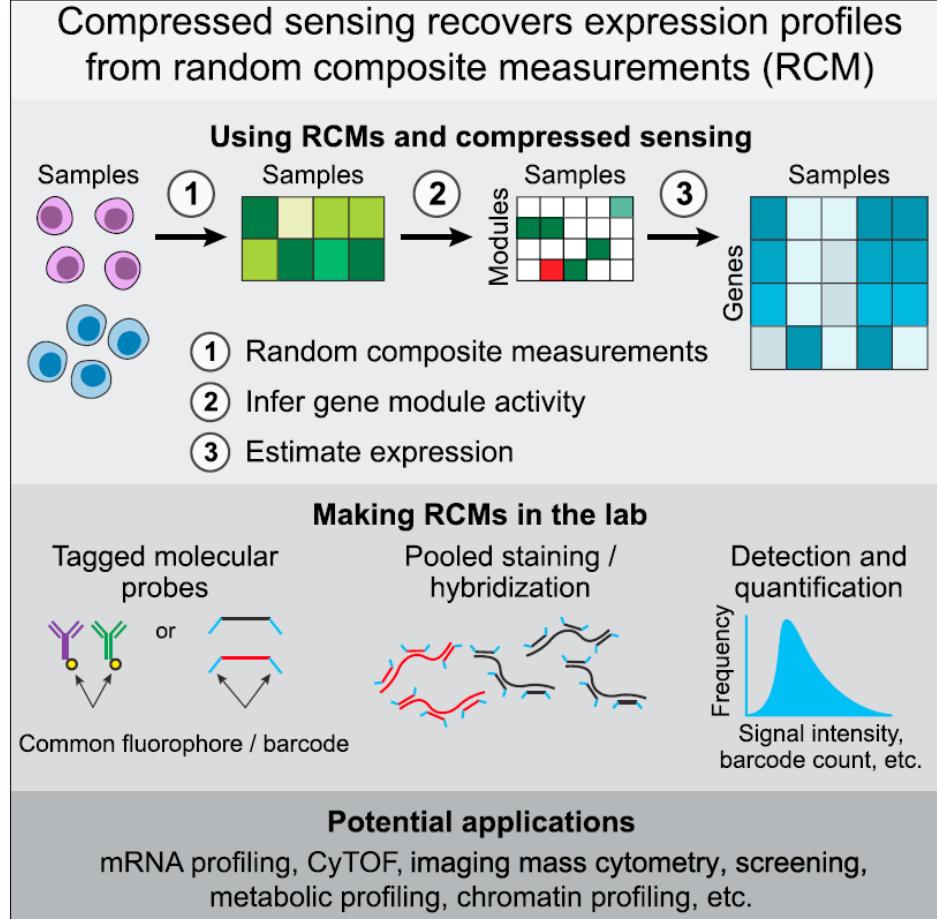
Brian Cleary,<sup>1,2</sup> Le Cong,<sup>1</sup> Anthea Cheung,<sup>1</sup> Eric S. Lander,<sup>1,3,4</sup> and Aviv Regev<sup>1,3,5,6,\*</sup>

<sup>1</sup>Klarman Cell Observatory, Broad Institute of MIT and Harvard, Cambridge, MA, USA

<sup>2</sup>Computational and Systems Biology Program, MIT, Cambridge, MA, USA

<sup>3</sup>Department of Biology, Massachusetts Institute of Technology, Cambridge, MA, USA

<sup>4</sup>Department of Systems Biology, Harvard Medical School, Boston, MA, USA



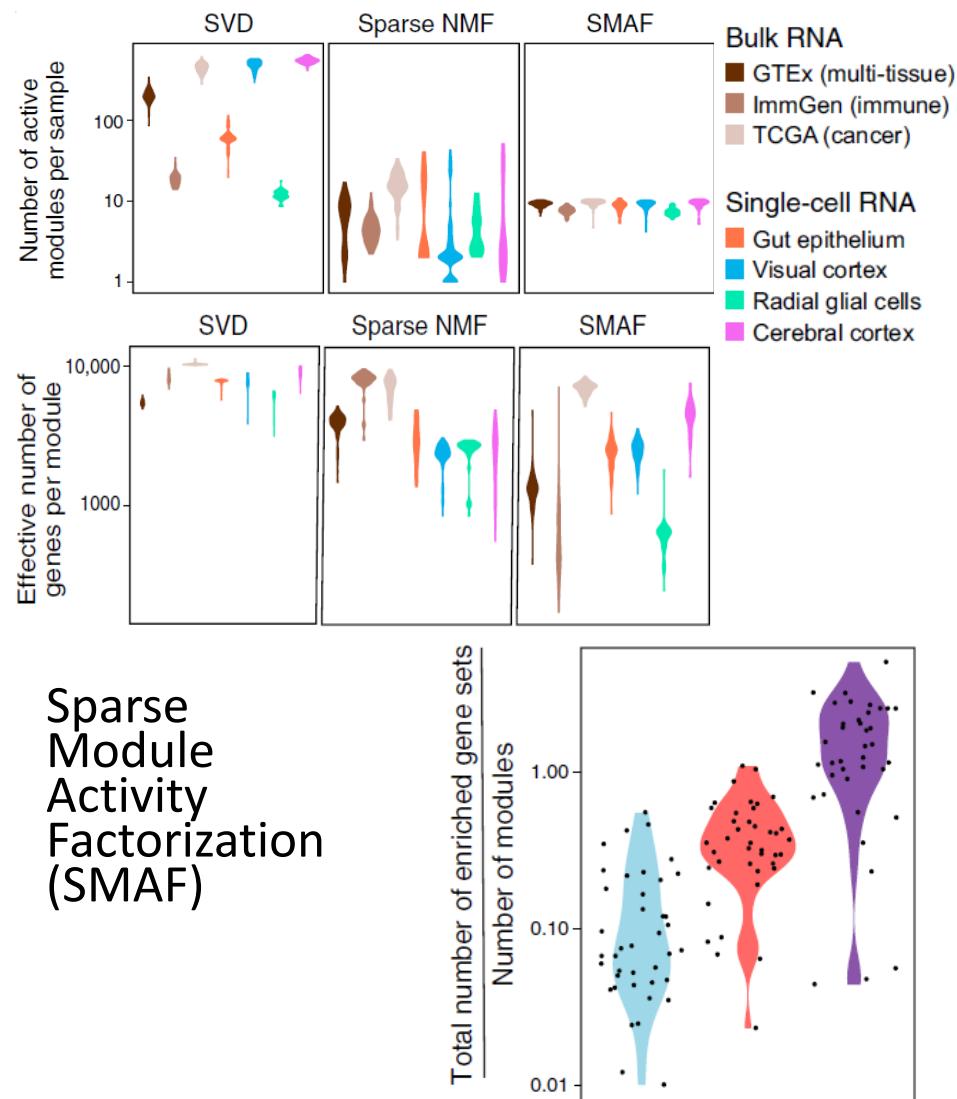
gene expression matrix  $X \in \mathbb{R}^{g \times n}$

module dictionary  $U \in \mathbb{R}^{g \times d}$

module activity matrix  $W \in \mathbb{R}^{d \times n}$

$$\min_{U,W} \|X - UW\|^2 + \lambda \|U\|_1$$

such that  $u_{ij} \geq 0$ ,  $\|u_{\cdot j}\| = 1$ , and  $\|w_i\|_0 \leq k \forall i \in \{1, \dots, n\}$

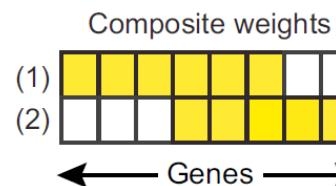
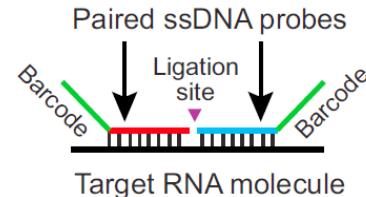


**Algorithm:** Sparse Module Activity Factorization

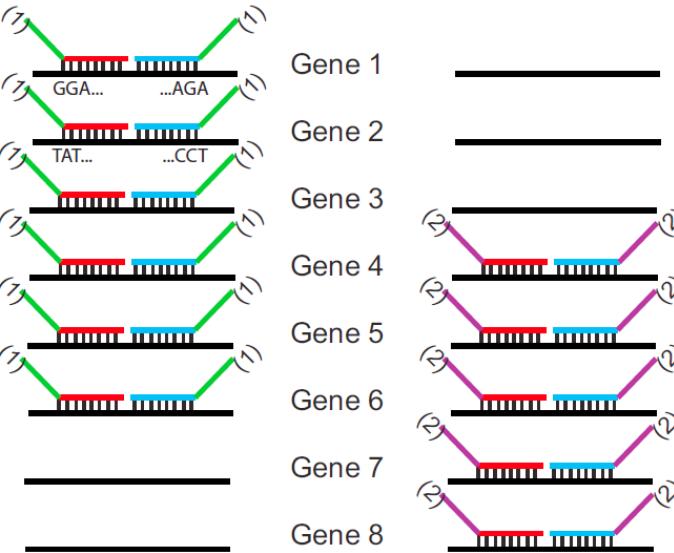
1.  $\text{SMAF}(X, d, \lambda, k)$
2. Initialize  $U \in \mathbb{R}^{g \times d}$  and  $W \in \mathbb{R}^{d \times n}$  randomly.
3. For 10 iterations:
  - a. Update the module dictionary as  $U = \text{LassoNonnegative}(X, W, \lambda)$ .
  - b. Normalize each module so that  $\|u_i\|_2 = 1$ .
  - c. Update the activity levels as  $W = \text{OMP}(X, U, k)$ .
4. Return  $U, W$ .

# Making composite measurements in practice

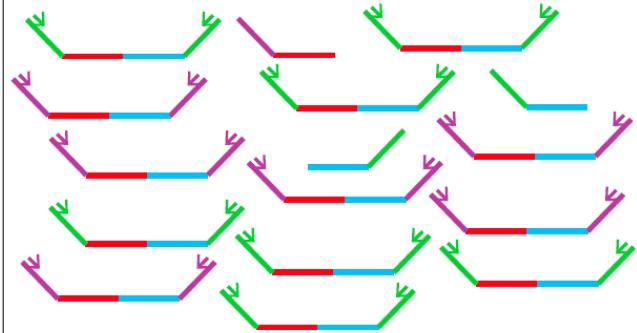
## Step 1: Design probes



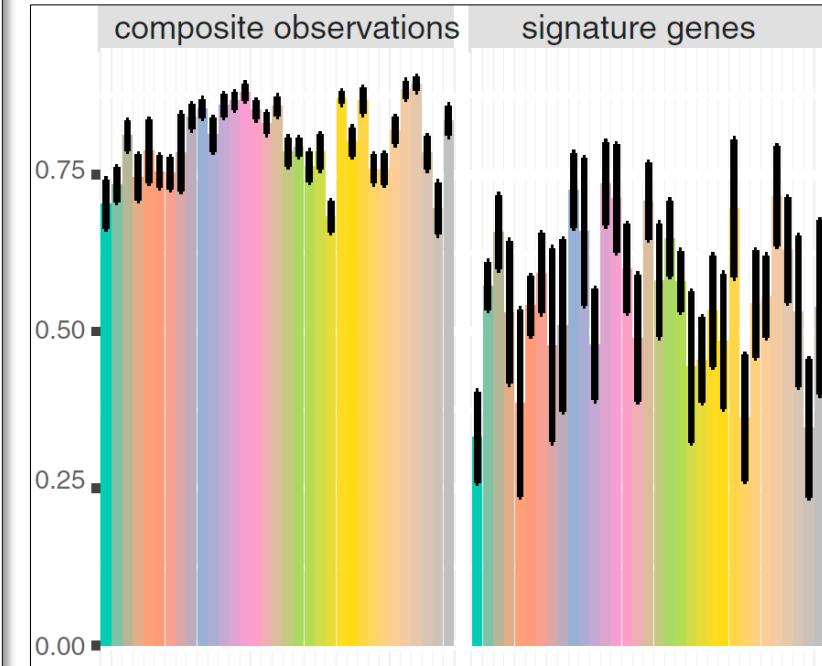
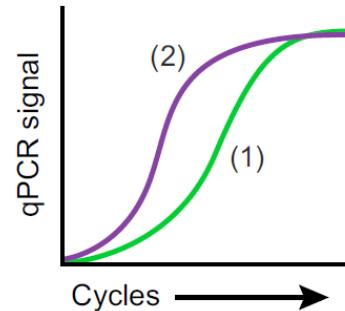
## Step 2: Hybridize and ligate



## Step 3: Pool ligation products



## Step 4: qPCR amplification



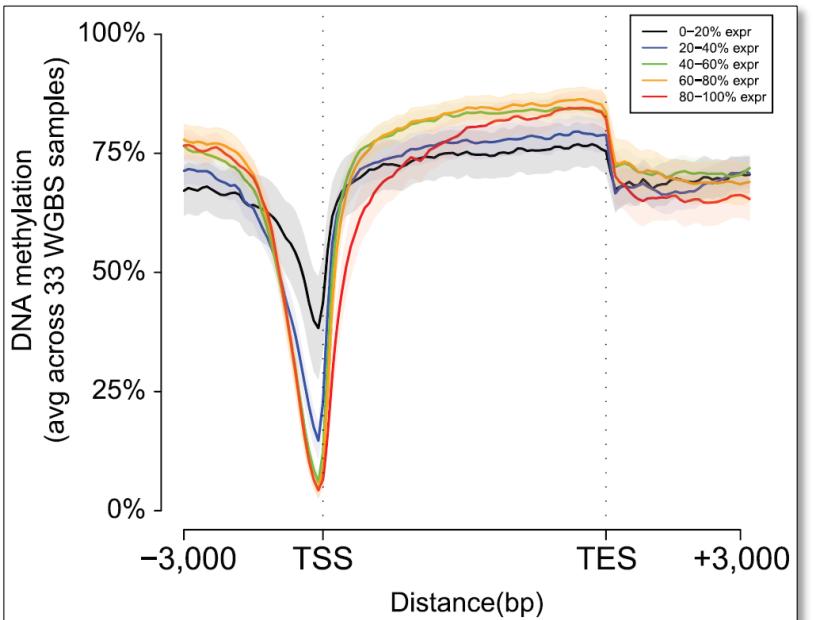
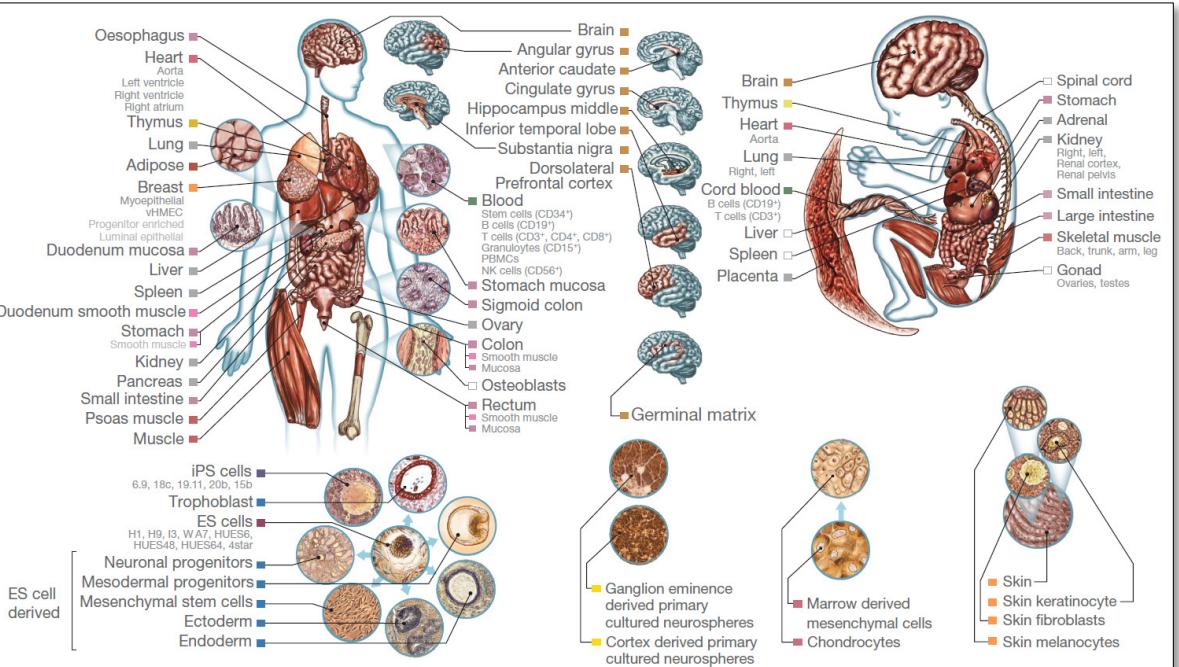
- Combinations of probes + barcodes for measurement
- More consistent signal-to-noise ratios

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

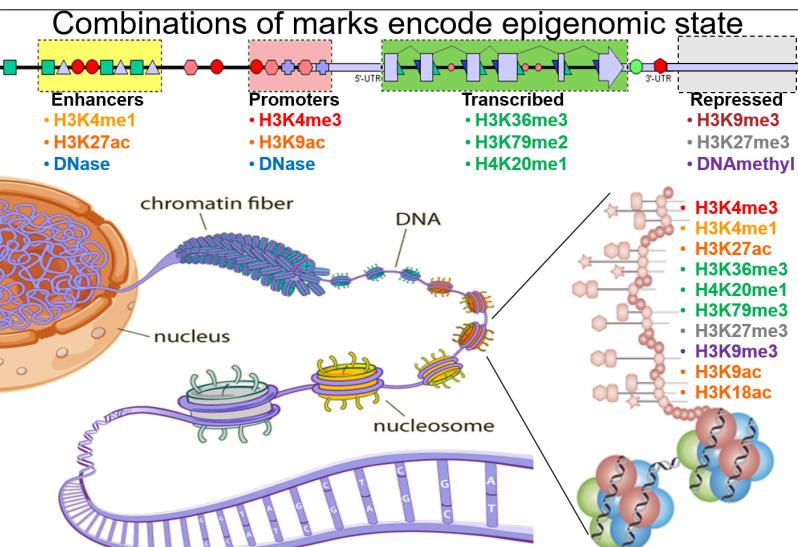
### 3. Predicting Expression from Chromatin

# Can we predict gene expression from chromatin information?

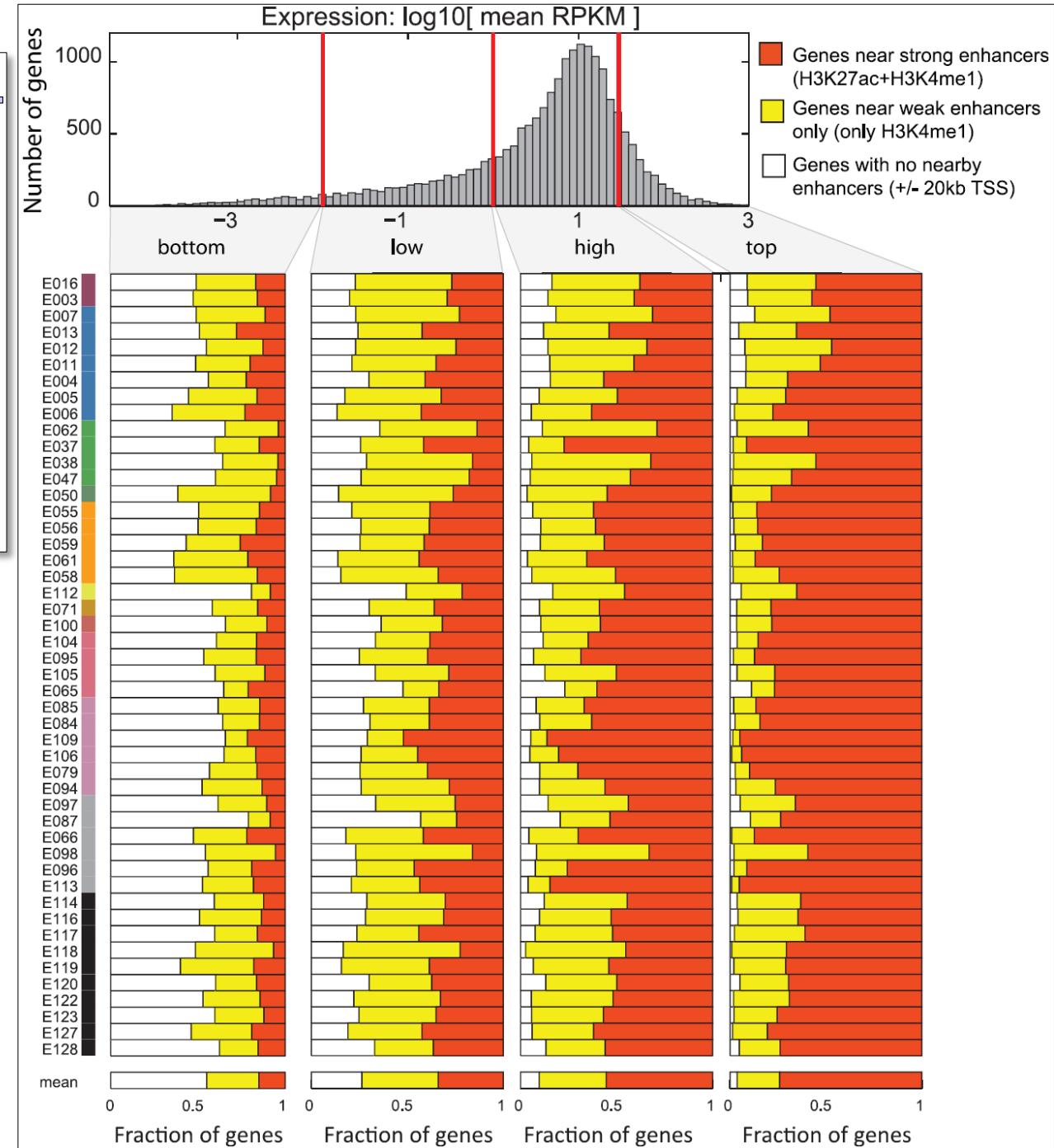
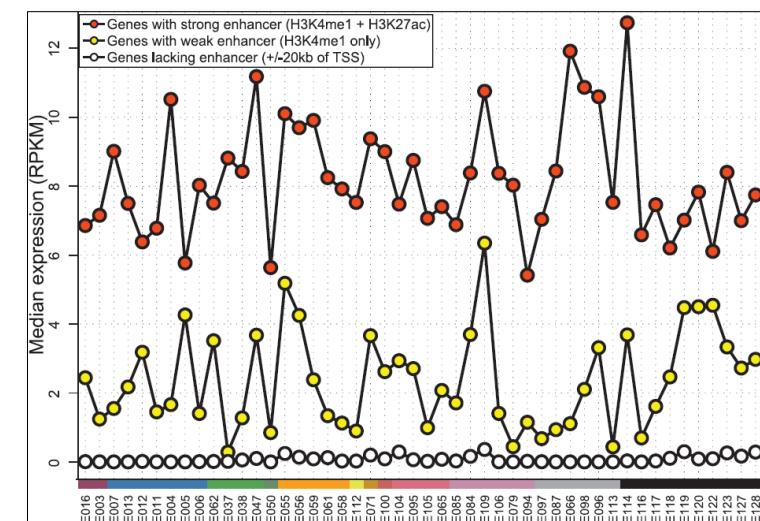


- DNA methylation vs. gene expression
  - Promoters: high. Gene body: low

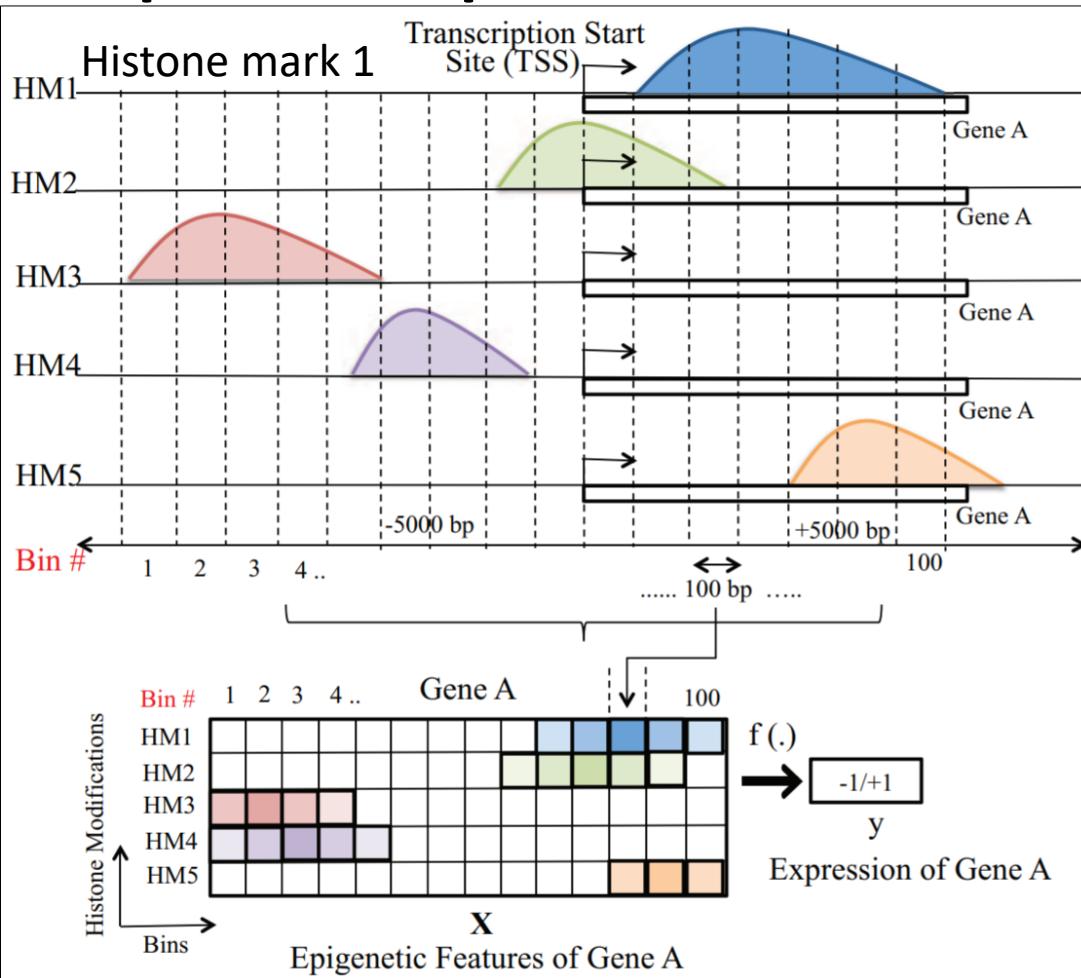
## Strong enhancers (+H3K27ac) vs. weak enhancers (H3K4me1 only)



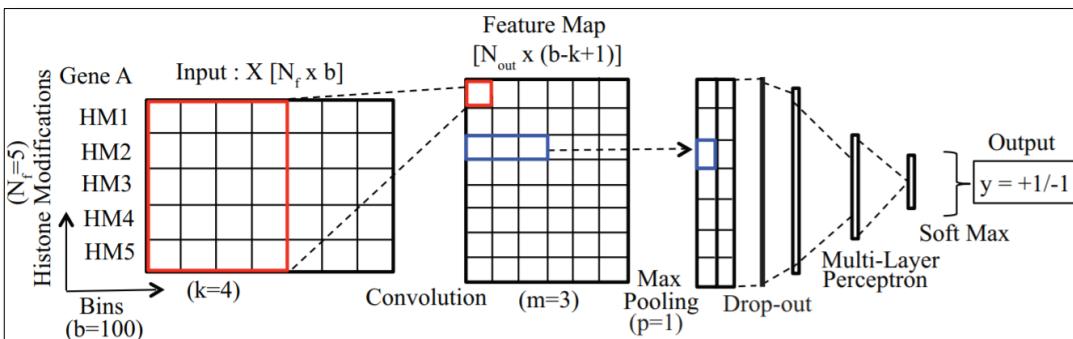
- 100s of known modifications, many new still emerging
  - Systematic mapping using ChIP-, Bisulfite-, DNase-Seq



# DeepChrome: positional histone features predictive of expression



- Positional information for each mark



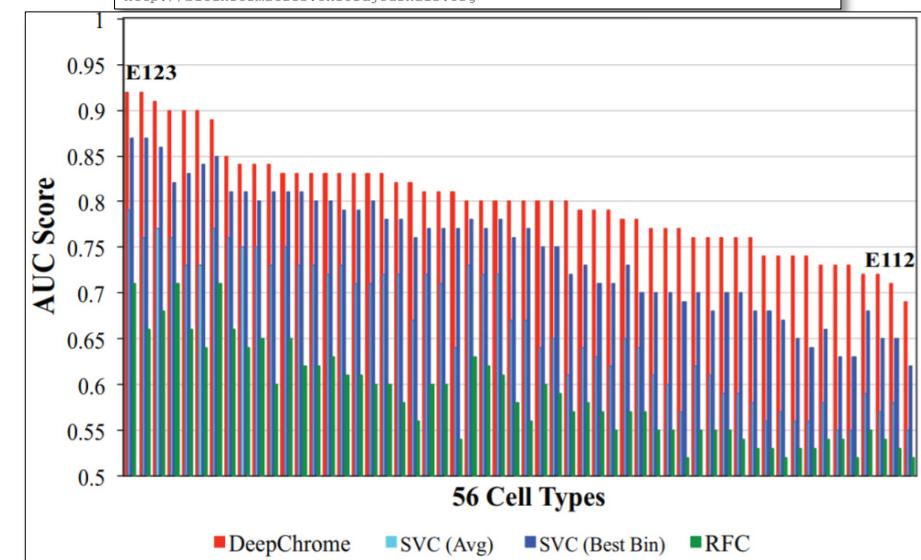
- Convolution, pooling, drop-out, Multi-Layer Perceptron (MLP) alternating lin/non-linear

**DeepChrome: Deep-learning for predicting gene expression from histone modifications.**

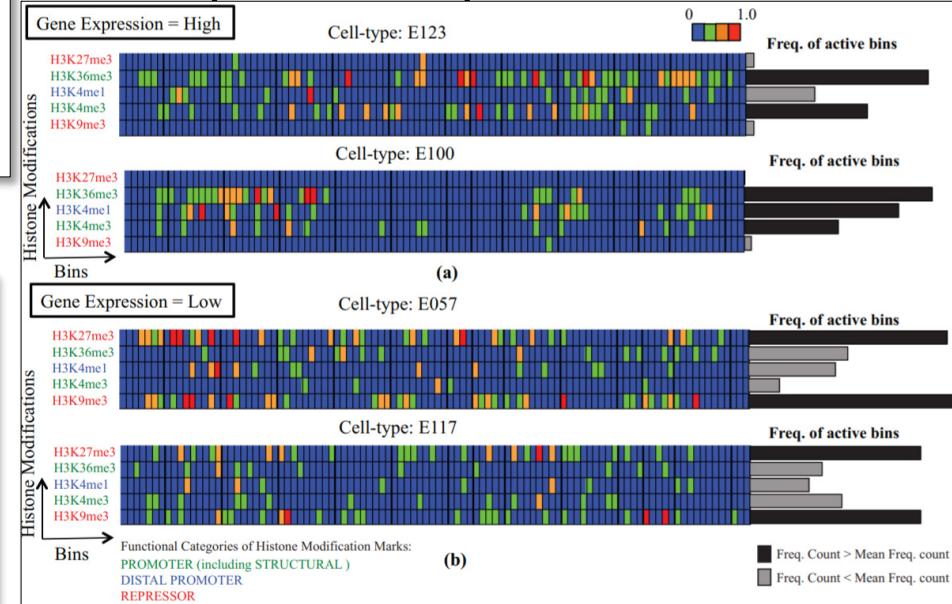
Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi \*

Department of Computer Science, University of Virginia, Charlottesville, VA, U.S.A

\* To whom correspondence should be addressed.  
This work will be published originally in Bioinformatics Journal at  
<http://bioinformatics.oxfordjournals.org>

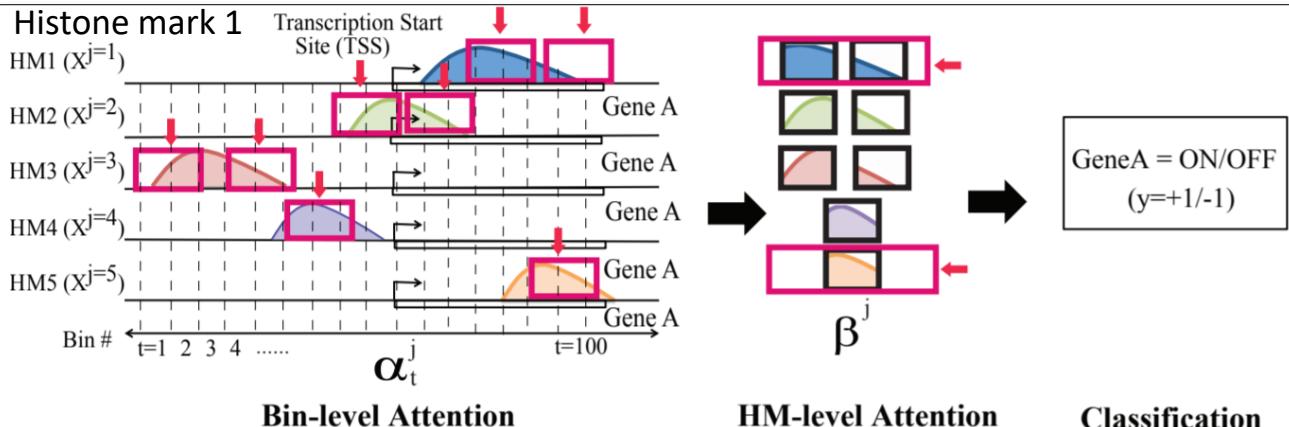


- Outperforms previous methods



- Meaningful features selected

# AttentiveChrome: Selectively attend to specific marks/positions

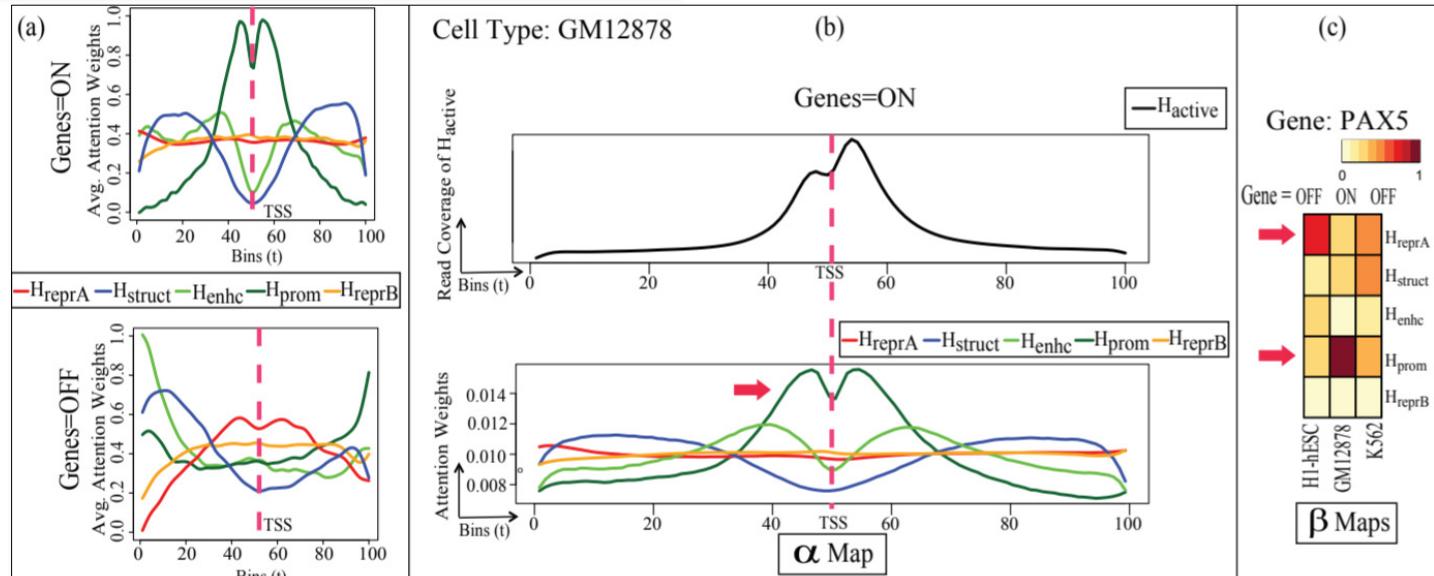


**Attend and Predict: Understanding Gene Regulation by Selective Attention on Chromatin**

Ritambhara Singh, Jack Lanchantin, Arshdeep Sekhon, Yanjun Qi  
Department of Computer Science  
University of Virginia  
yanjun@virginia.edu

- Attention: LSTM: Long short-term memory module
- Hierarchical LSTM modules: interactions across marks

- Attention focuses on specific positions for specific marks



Model	Baselines			AttentiveChrome Variations				
	DeepChrome (CNN) [29]	LSTM	CNN-Attn	CNN- $\alpha, \beta$	LSTM-Attn	LSTM- $\alpha$	LSTM- $\alpha, \beta$	
Mean	0.8008	0.8052	0.7622	0.7936	0.8100	<b>0.8133</b>	0.8115	
Median	0.8009	0.8036	0.7617	0.7914	0.8118	<b>0.8143</b>	0.8123	
Max	<b>0.9225</b>	0.9185	0.8707	0.9059	0.9155	0.9218	0.9177	
Min	0.6854	0.7073	0.6469	0.7001	<b>0.7237</b>	0.7250	0.7215	
Improvement over DeepChrome [29] (out of 56 cell types)		36	0	16	49	<b>50</b>	49	

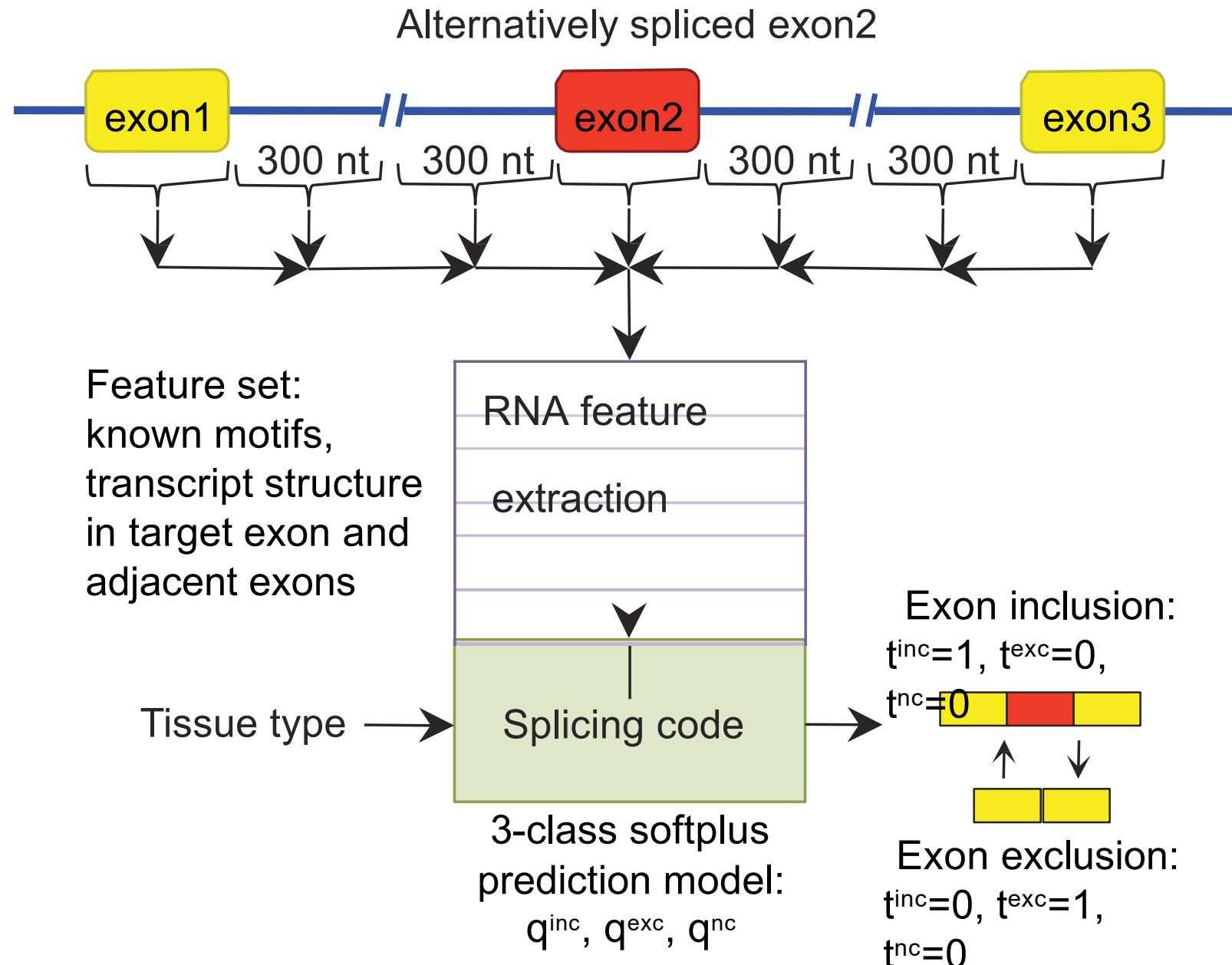
- Consistent improvement over DeepChrome

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

## 4. Predicting splicing from sequence

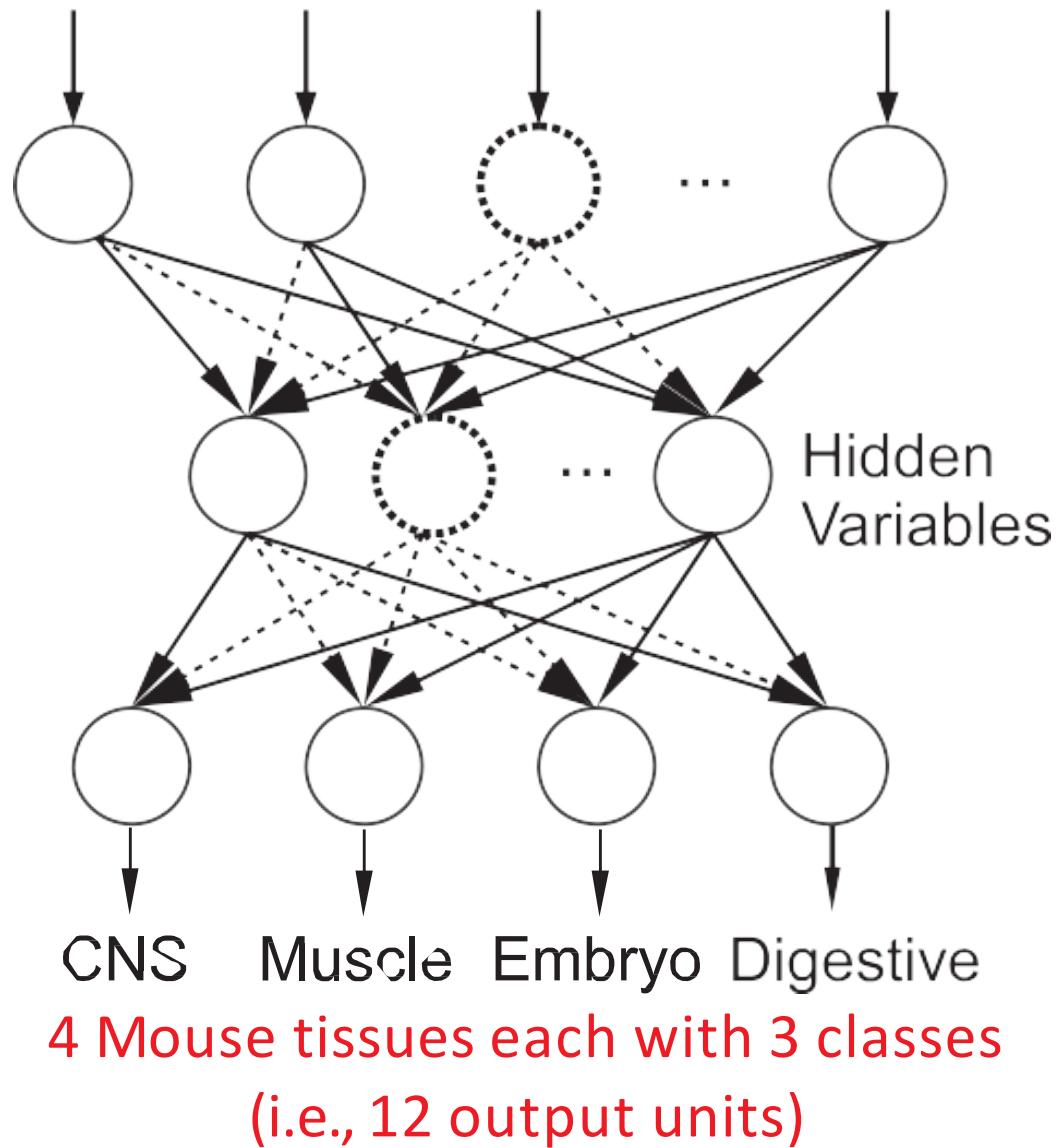
# Deciphering tissue-specific splicing code



[Barash et al., 2010]

# Bayesian neural network splicing code

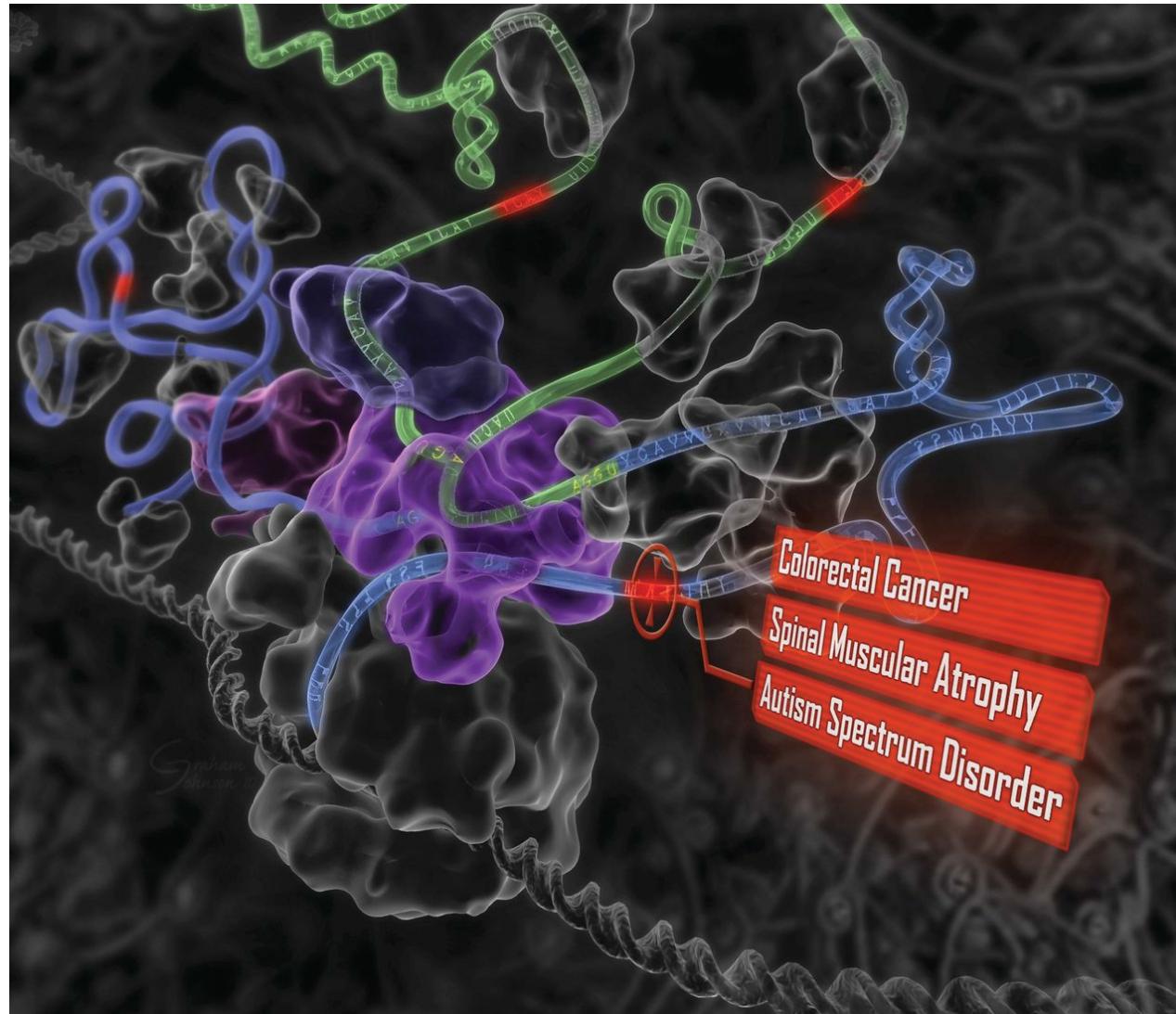
1014 RNA features x 3665 exons



Bayesian neural network:

- # hidden units follows Poisson( $\lambda$ )
- Network weights follows spike-and-slab prior  $\text{Bern}(1 - a)$
- Likelihood is cross-entropy
- Network weights are sampled from the posterior

# Predicts disease-causing mutations from splicing code

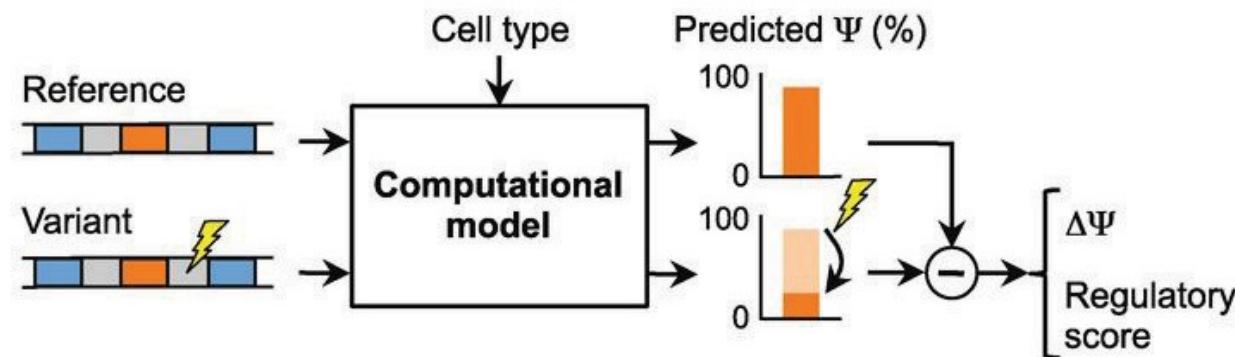


[Xiong et al., 2011]

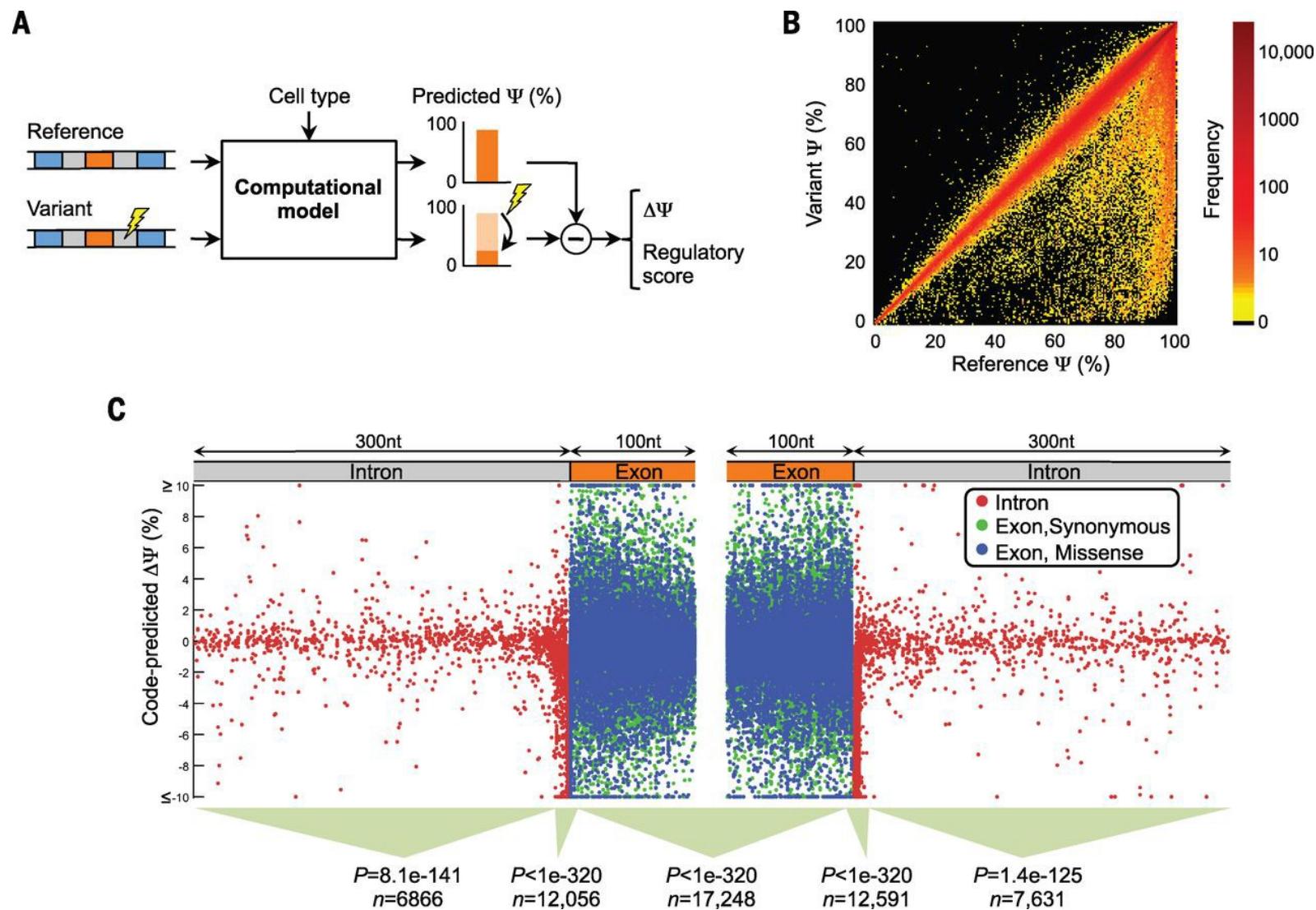
# Predicts disease-causing mutations from splicing code

Scoring splicing changes due to SNP  $\Delta\psi$ :

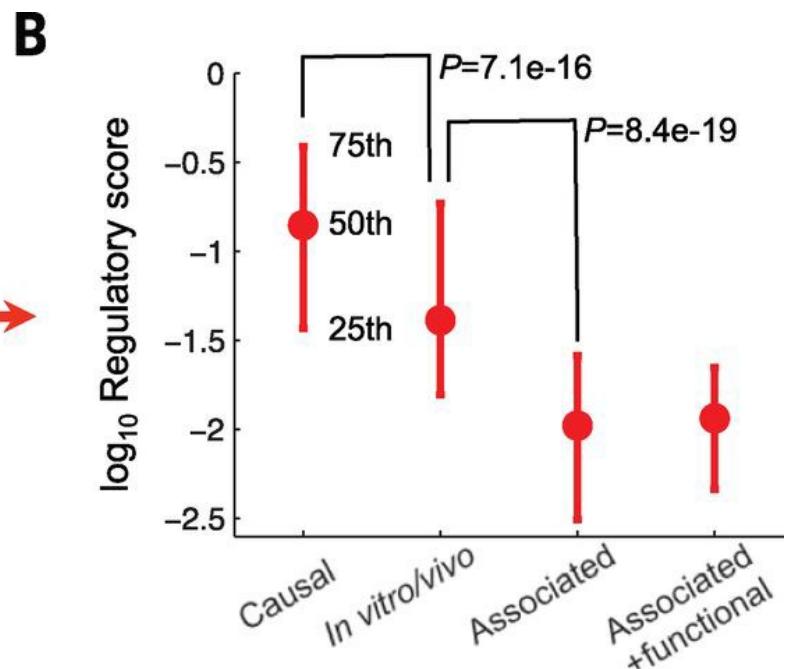
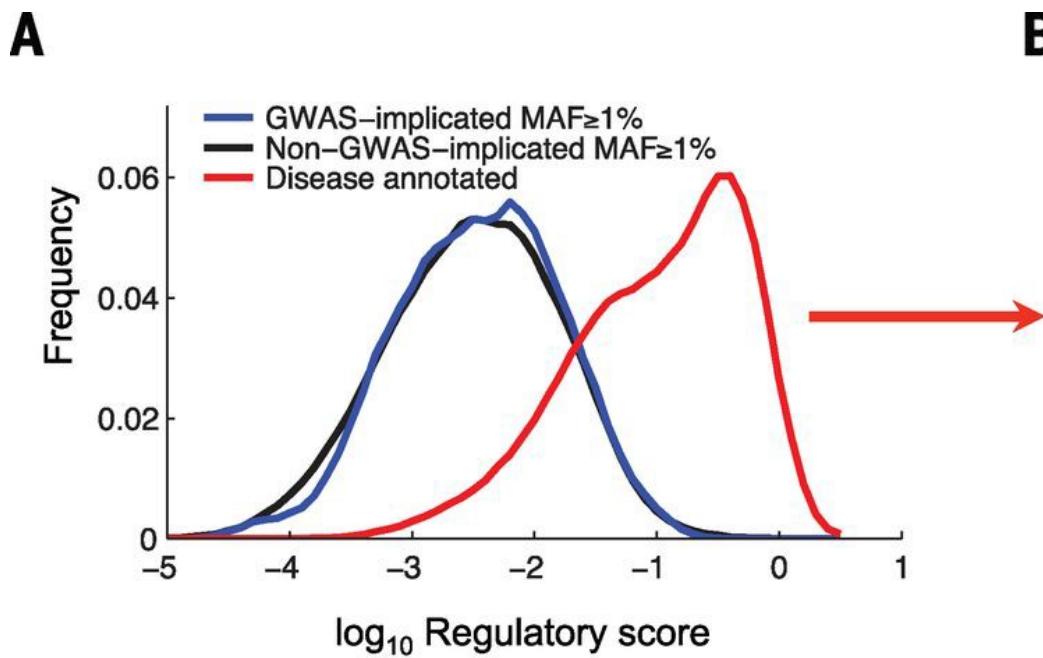
- Train splice code model on 10,689 exons to predict the 3 splicing classes over 16 human tissues using 1393 sequence features (motifs & RNA structures)
- Score both the reference  $\psi_{ref}$  and alternative  $\psi_{alt}$  sequences harboring one of the 658,420 common variants
- Calculate  $\Delta\psi_t = \psi^t_{ref} - \psi^r_{alt}$  over each tissue t
- Obtain largest absolute or aggregate  $\Delta\psi_t$  to score effects of SNPs



# Predicted scores are indicative of disease causing mutations

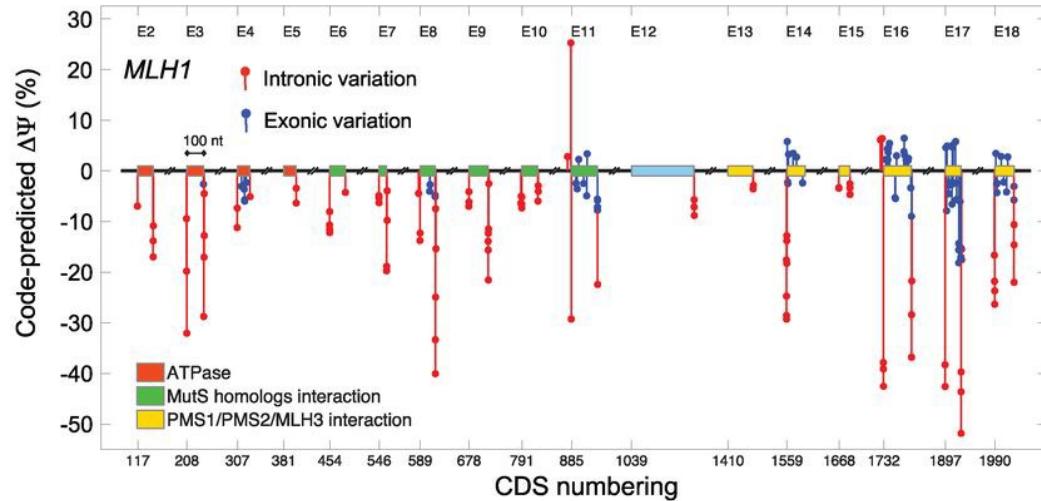


# Predicted scores are indicative of disease causing mutations

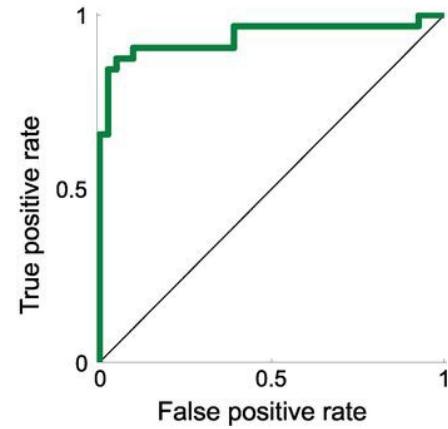
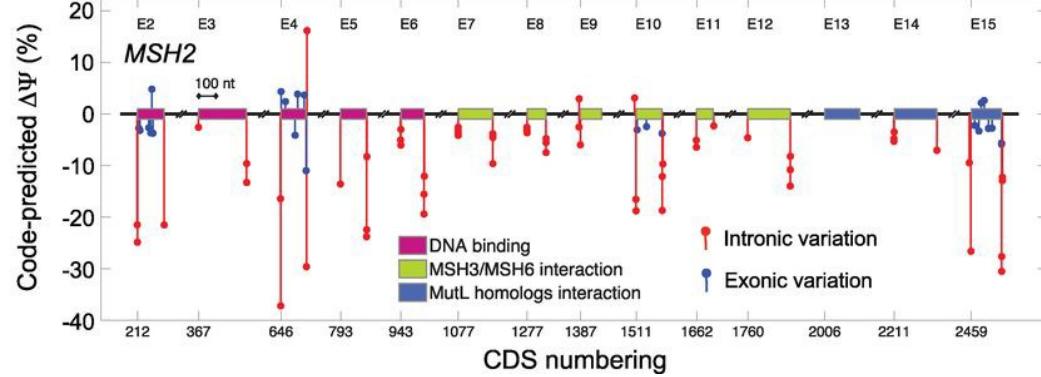
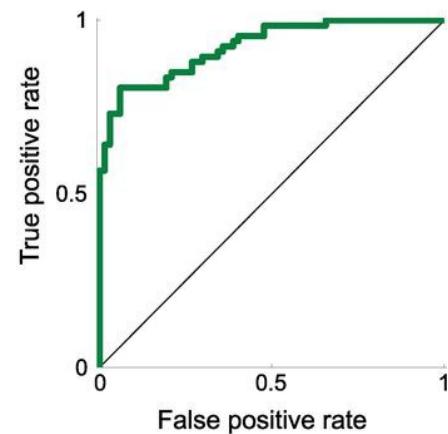


# Predicted mutations in MLH1,2 in nonpolyposis colorectal cancer patients are validated via RT-PCR

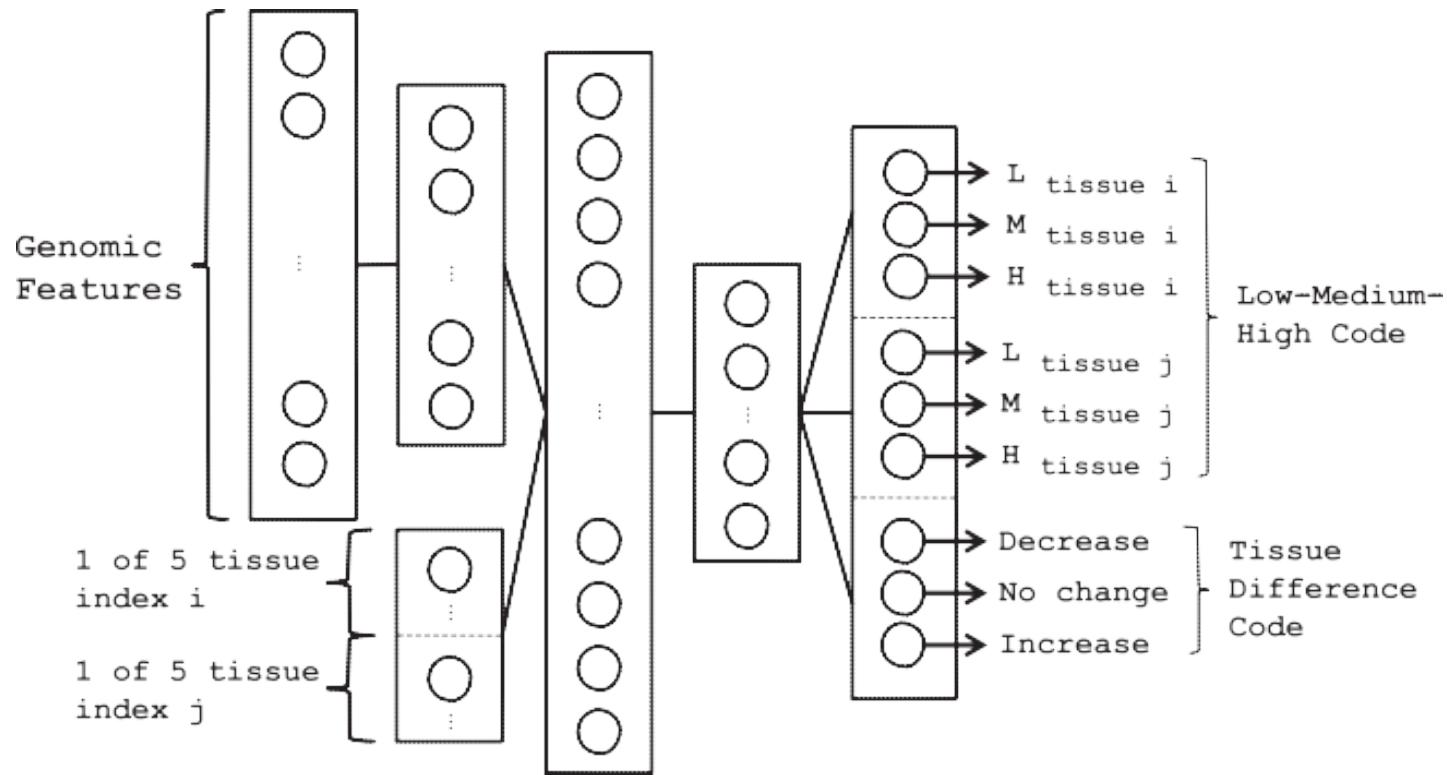
A



B



# Splice code goes deep



Architecture of the new network to predict alternative splicing between two tissues. It contains three hidden layers, with hidden variables that jointly represent genomic features and tissue types.

# Limitations of the splice code model

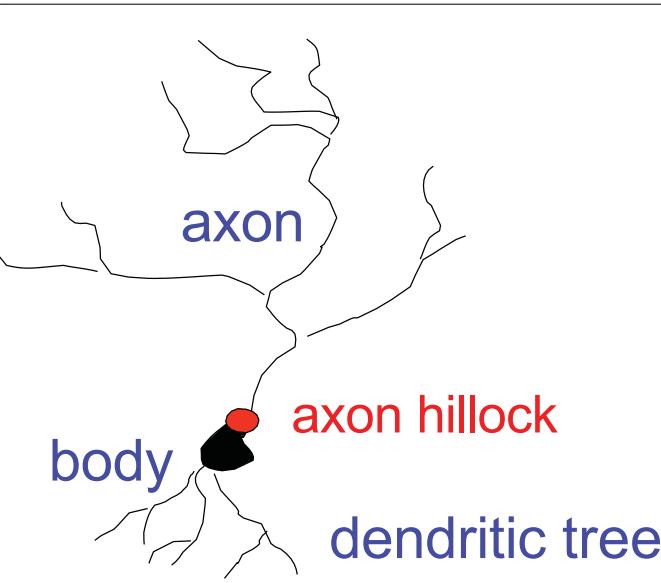
- Require threshold to define discrete splicing targets
- Not taking into account exon expression level in specific tissue types
- Fully connected neural network potentially impose a large number of parameters:  $(1393 \text{ inputs} + 13 \text{ outputs}) \times 10 \text{ hidden units} = 13000 \text{ parameters}$
- Although authors showed that neural network performs the best a softplus/Dirichlet multivariate linear regression may achieve similar performance
- The features are pre-defined and thus may not completely reflect the underlying splicing mechanism
- Interpretation of the importance of features is not trivial

# Today: Predicting gene expression and splicing

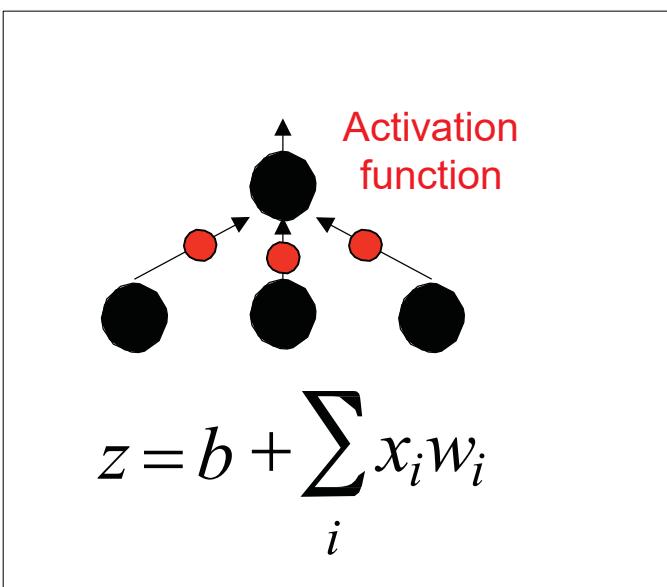
0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

# 5. Unsupervised deep learning with Restricted Boltzmann Machines (RBMs)

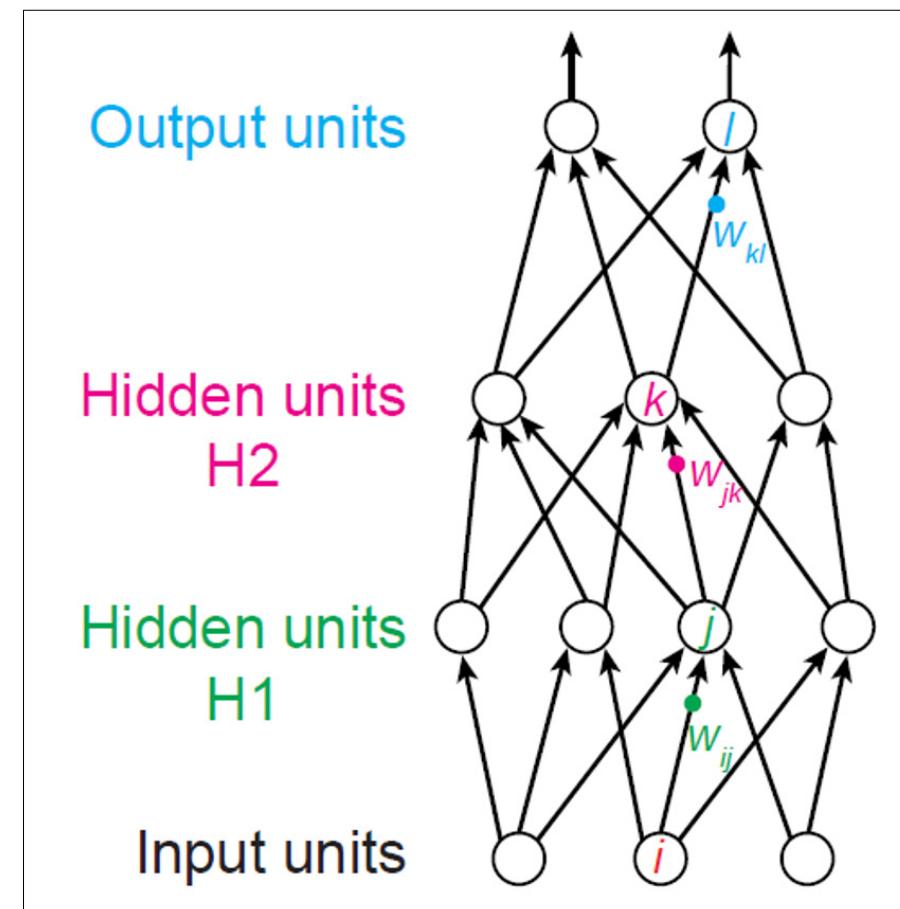
# How the brain works inspired artificial “neural” networks



Biological neuron



Artificial perceptron

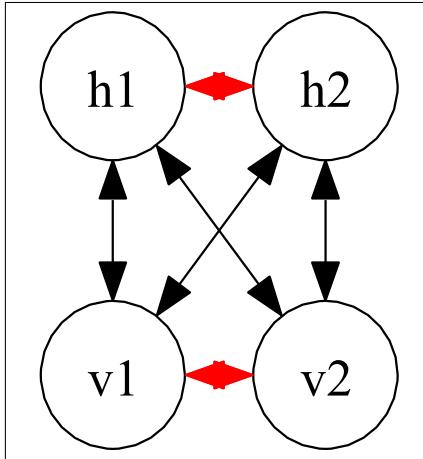


Neural Network (e.g. 4-layers ‘deep’)

Deep multi-layer neural networks can  
‘learn’ almost any function

→ Deep ‘unsupervised’ learning?

# General Boltzmann Machine: Unsupervised learning



- Symmetrically connected network (no target ‘output’)
- Each binary unit makes stochastic on/off decision
- Network weights learn relationships between variables
- Configuration dictates “energy”. At equilibrium, follows Boltzmann distribution (exponentiated negative energy)

$$P(\mathbf{v} = \mathbf{v}, \mathbf{h} = \mathbf{h}) = \frac{1}{Z} \exp(-E(\mathbf{v}, \mathbf{h}))$$

[Ackley et al., 1985; Le Roux, Bengio, 2008]

$$E(\mathbf{v}, \mathbf{h}) = - \sum_i s_i b_i - \sum_{i < j} s_i s_j w_{ij}$$

$$P(\mathbf{v}) = \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))}{\sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}))}$$

**v=visible** units; **h=hidden** units,

$E(\mathbf{v}, \mathbf{h})$  energy function

$w_{ij}$ =connection weights

$s_i \in \{0, 1\}$ =state of unit  $i$ ,

$b_i$ =bias of unit  $i$  in global

$P(\mathbf{v})$  energy-dependent prob. function

Goal: Given  $\mathbf{v}$ , learn weights  $w_{ij}$  to maximize  $P(\mathbf{v})$ .

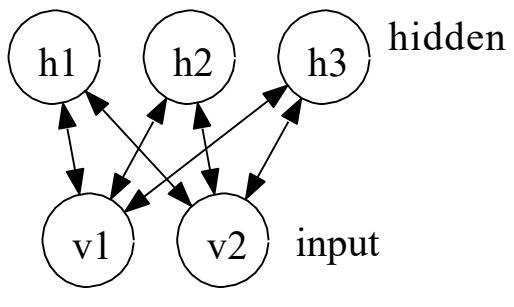
Boltzmann machine becomes universal approximator  
of probability mass functions over discrete variables.

Adv: Local learning rules, infer each variable based on neighbors only.

No need for example annotations, no output function.

Problem: Difficult to train, dependencies between hidden units

# Restricted Boltzmann Machine (RBM)



- Bipartite graph. No  $h \Leftrightarrow h$  and no  $v \Leftrightarrow v$  connections
- 1 layer of hidden units, 1 layer of visible units.
- Simple **unsupervised** learning module
- Much easier to train than GBM: no circularities

Objective function:

$$E(\mathbf{v}, \mathbf{h}|\theta) = - \sum_{ij} w_{ij} v_i h_j - \sum_i b_i v_i - \sum_j b_j h_j$$

$$p(\mathbf{v}|\theta) = \prod_{n=1}^N \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}|\theta) = \prod_{n=1}^N \frac{\sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta))}{\sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta))}$$

$$\log p(\mathbf{v}|\theta) = \sum_{n=1}^N \left( \log \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta)) - \log \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}|\theta)) \right)$$

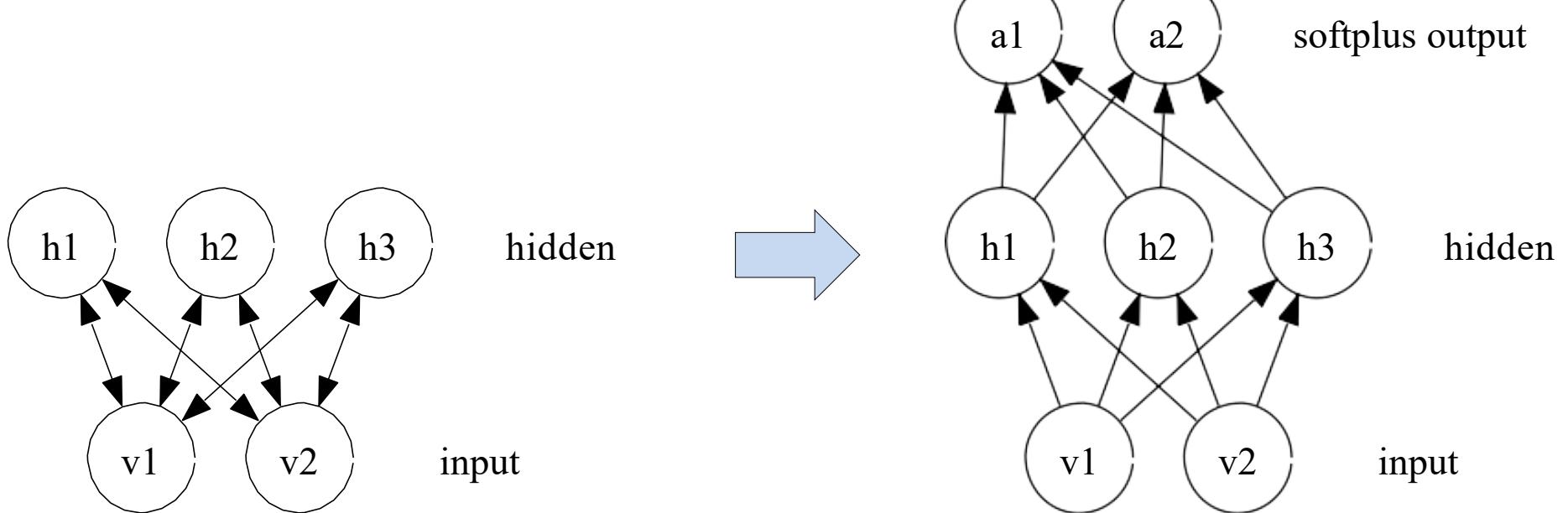
$$\begin{aligned} \frac{\partial \log p(\mathbf{v}|\theta)}{\partial w_{ij}} &= \sum_{n=1}^N \left[ v_i \sum_{\mathbf{h}} h_j p(\mathbf{h}|\mathbf{v}) - \sum_{\mathbf{v}, \mathbf{h}} v_i h_j p(\mathbf{v}, \mathbf{h}) \right] \\ &= \mathbb{E}_{\text{data}}[v_i h_j] - \mathbb{E}_{\text{model}}[\hat{v}_i \hat{h}_j] \equiv \langle v_i h_j \rangle_{\text{data}} - \langle \hat{v}_i \hat{h}_j \rangle_{\text{model}} \end{aligned}$$

However:  $\langle \hat{v}_i, \hat{h}_j \rangle$  model still too large to estimate.

→ apply Markov Chain Monte Carlo (MCMC) (i.e., Gibbs sampling)

# Stacking RBMs → Deep belief network

1. First apply RBM to find a sensible set of weights using unlabelled data.
2. Then use the pre-trained weight to perform backpropagation to classify labelled data



# Look into the mind of the network: **generative** model



1<sup>st</sup> column: Sample from generative model with each label clamped on.

2<sup>nd</sup> column: 20 iterations of alternating Gibbs sampling in associative memory.  
etc... (Figure 9, Hinton et al., 2006).

Interactive visualization of network learning:

<http://www.cs.toronto.edu/~hinton/digits.html>

# Today: Predicting gene expression and splicing

0. Review: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Unsupervised deep learning: Restricted Boltzmann mach.
6. Multi-modal programs: Expr+DNA+miRNA RMBs Liang

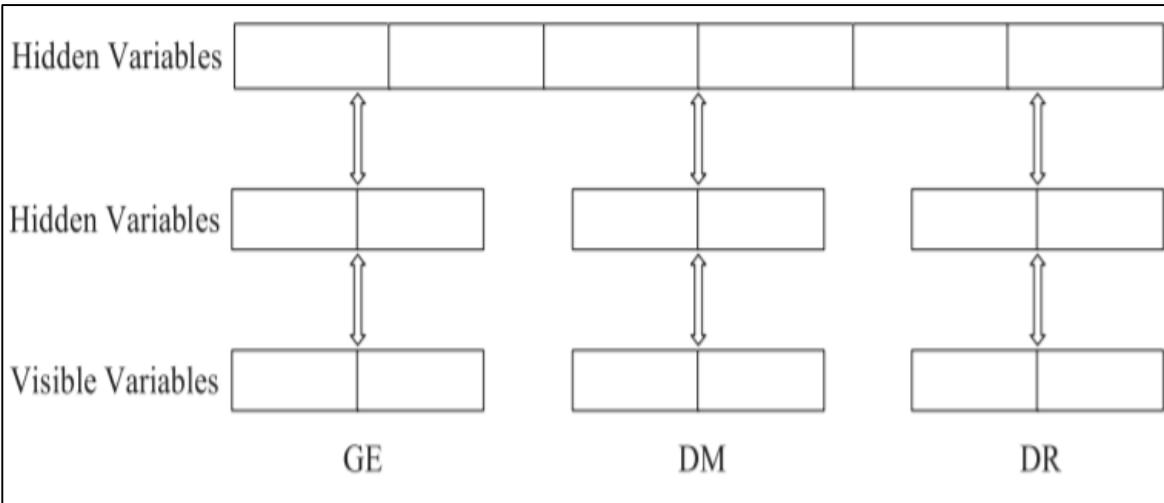
## 6. Multimodal unsupervised deep learning for data integration with RBMs

# RBM<sub>s</sub> for TCGA cancer integration: Expression, miRNAs, Methylation

IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 12, NO. 4, JULY/AUGUST 2015

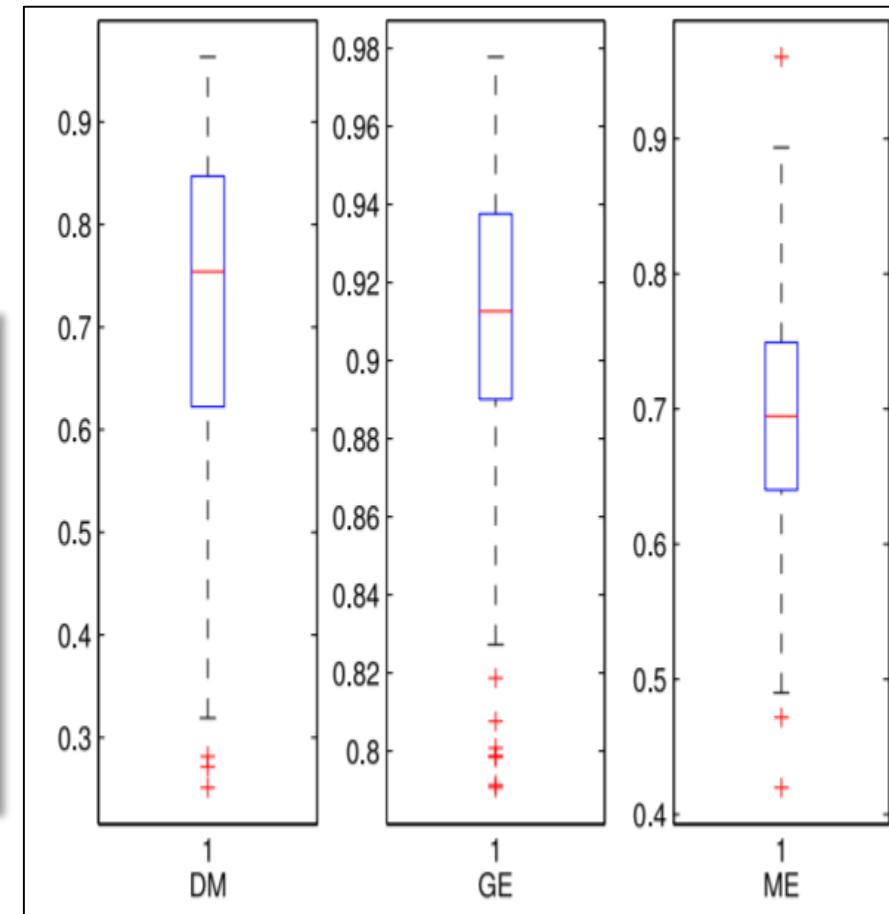
## Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach

Muxuan Liang, Zhizhong Li, Ting Chen, and Jianyang Zeng



Hierarchical model integrates:

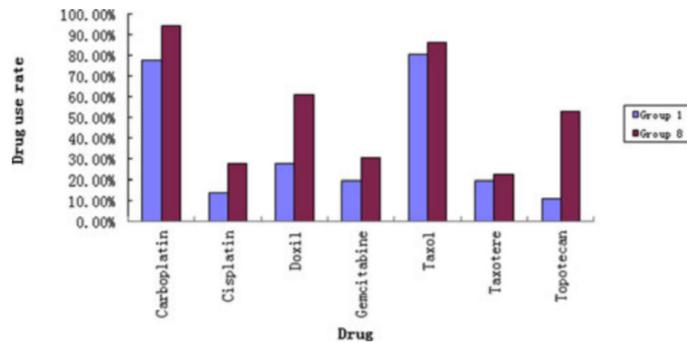
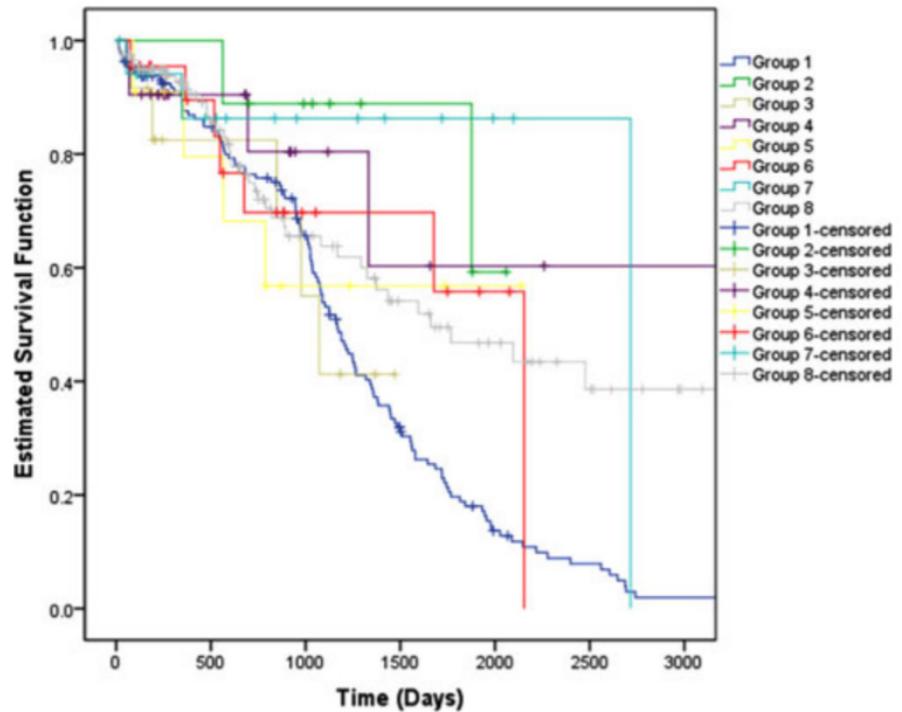
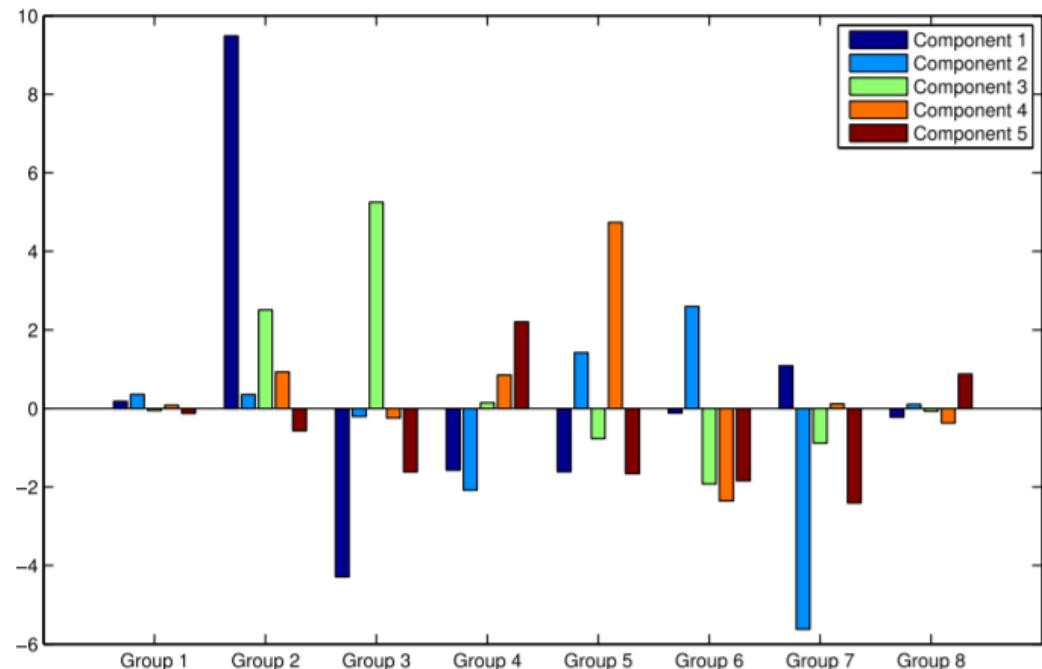
- gene expression (GE)
- miRNA expression (ME)
- DNA methylation (DM)



$$E(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) = \sum_{i=1}^m \frac{(v_i - a_i)^2}{2\sigma_i^2} + \sum_{i=1}^m \sum_{j=1}^g \frac{v_i}{\sigma_i} W_{ij} h_j + \sum_{j=1}^g b_j h_j$$

Energy function combines multiple data types

# Learned patient groups show different survival/drugs



- Capture independent variables from molecular data