Deep Learning in the Life Sciences
6.874, 6.802, 20.390, 20.490, HST.506

Lecture 09: Predicting gene expression and splicing

Prof. Manolis Kellis

Guest lectures:
Flynn Chen, Mark Gerstein Lab, Yale
Prof. Xiaohui Xie, UC Irvine
Dr. Kyle Kai-How Farh, Illumina

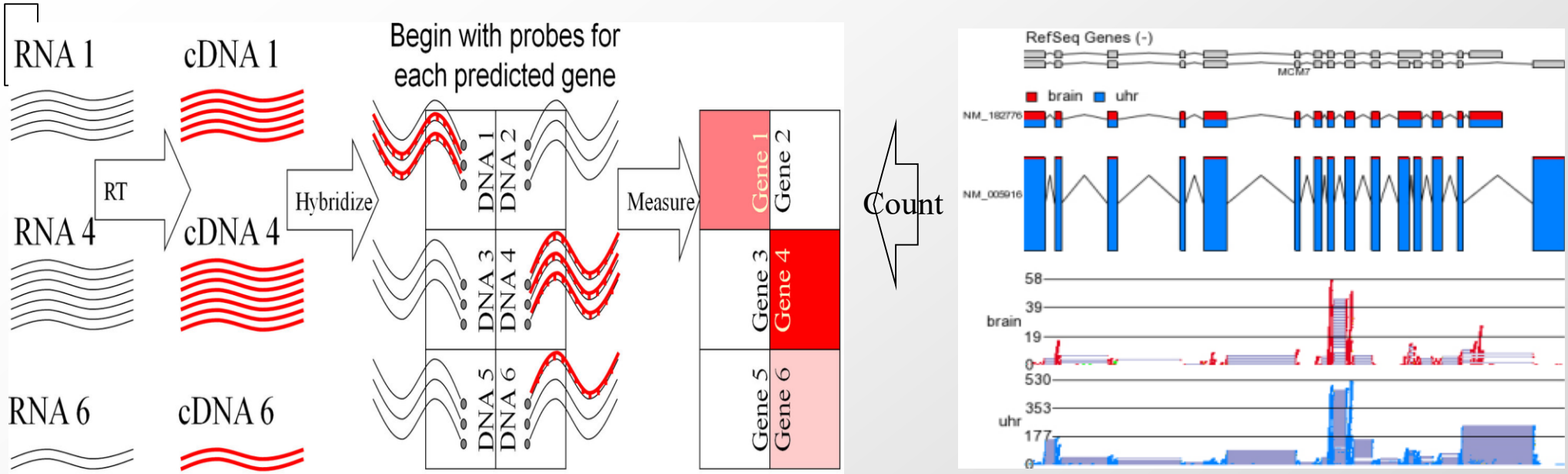
Massachusetts Institute of Technology

http://compbio.mit.edu/6874

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

RNA-Seq: De novo tx reconstruction / quantification



Microarray technology

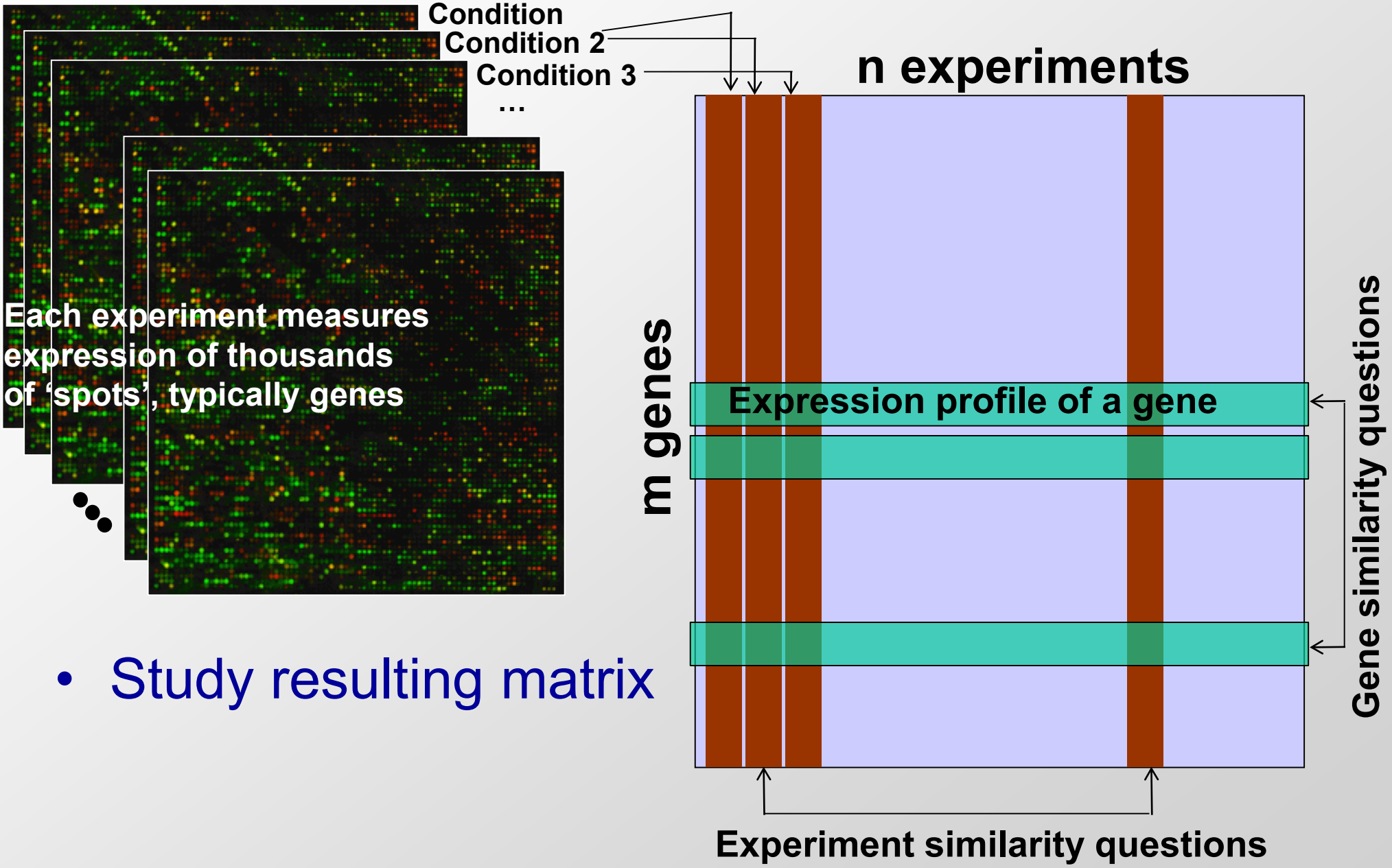
- Synthesize DNA probe array, complementary hybridization
- Variations:
  - One long probe per gene
  - Many short probes per gene
  - Tiled k-mers across genome
- Advantage:
  - Can focus on small regions, even if few molecules / cell

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

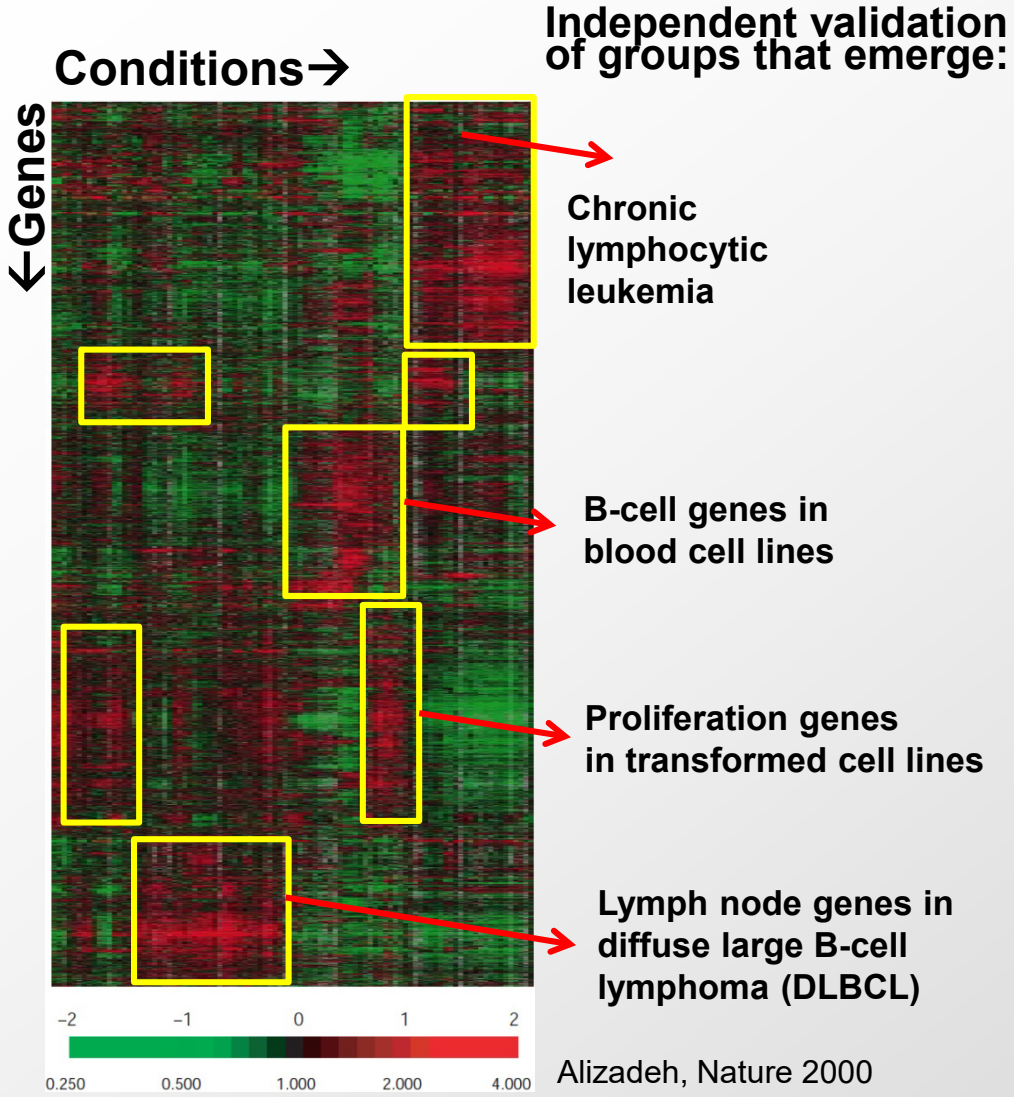
Expression Analysis Data Matrix

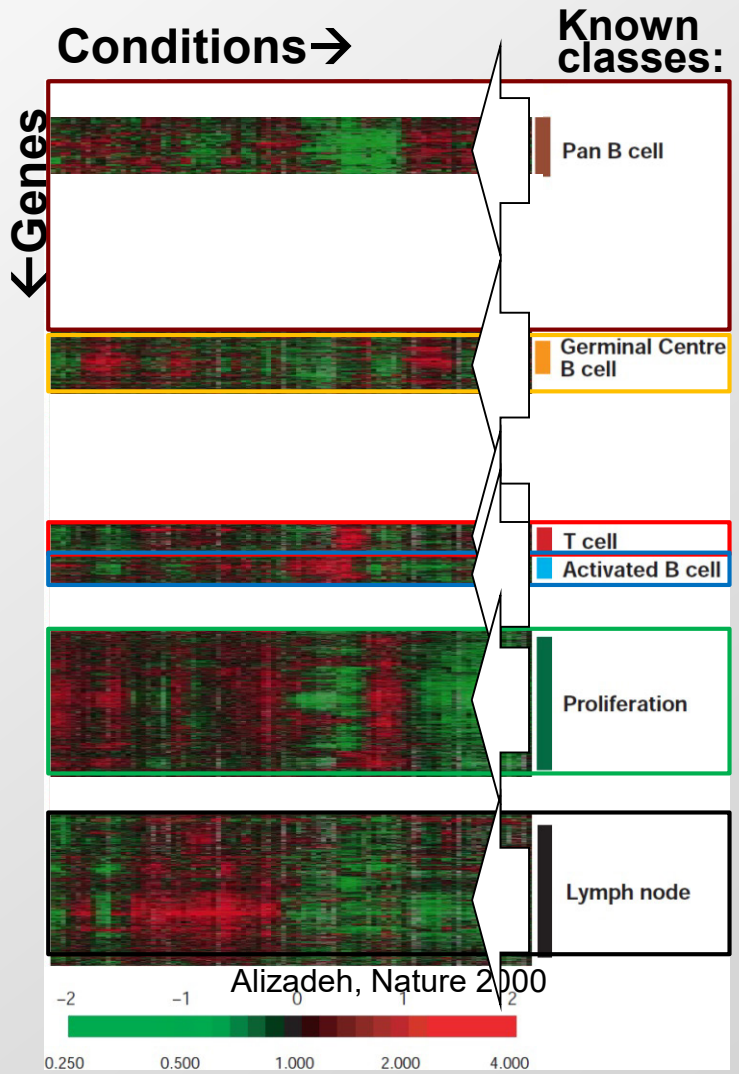
- Measure 20,000 genes in 100s of conditions



- Study resulting matrix

Clustering vs. Classification



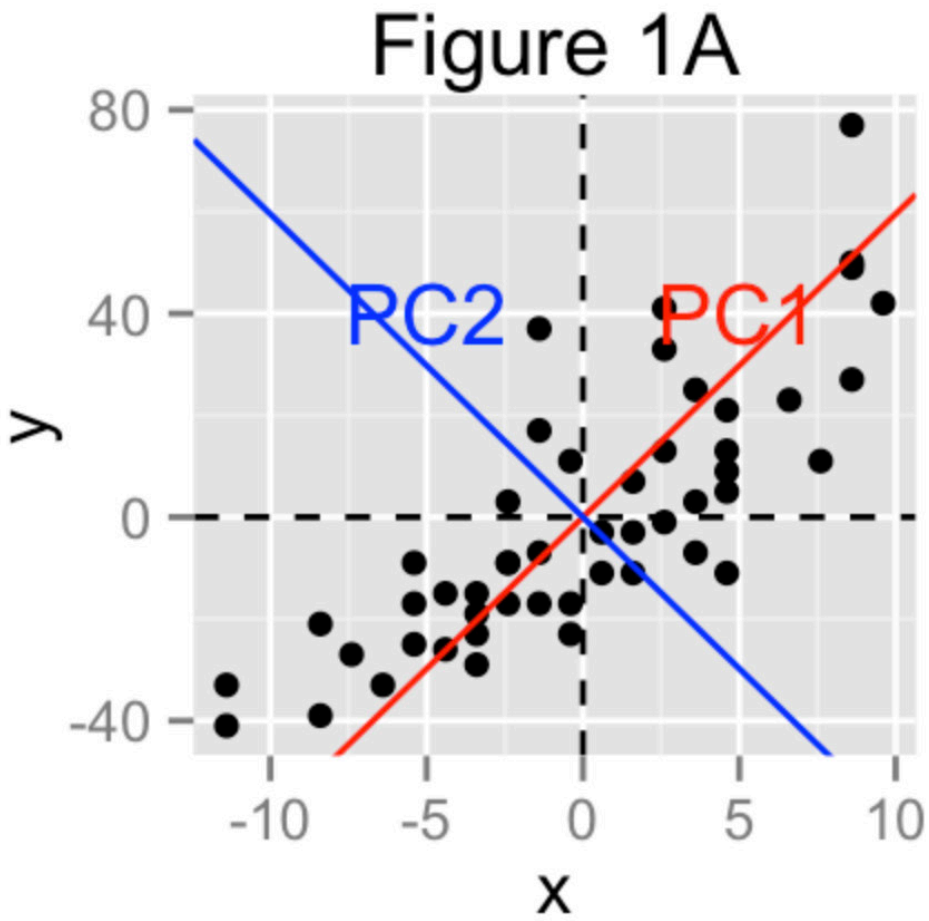


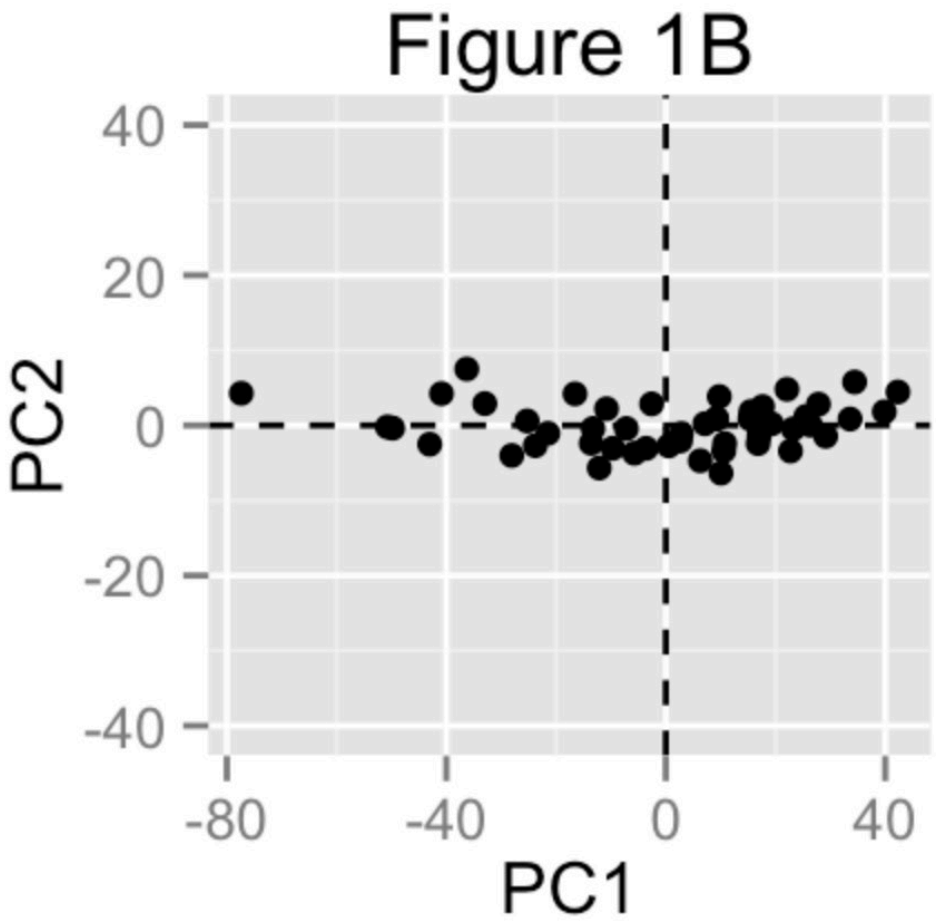
Goal of Clustering: **Group similar items** that likely come from the same category, and in doing so **reveal hidden structure**

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

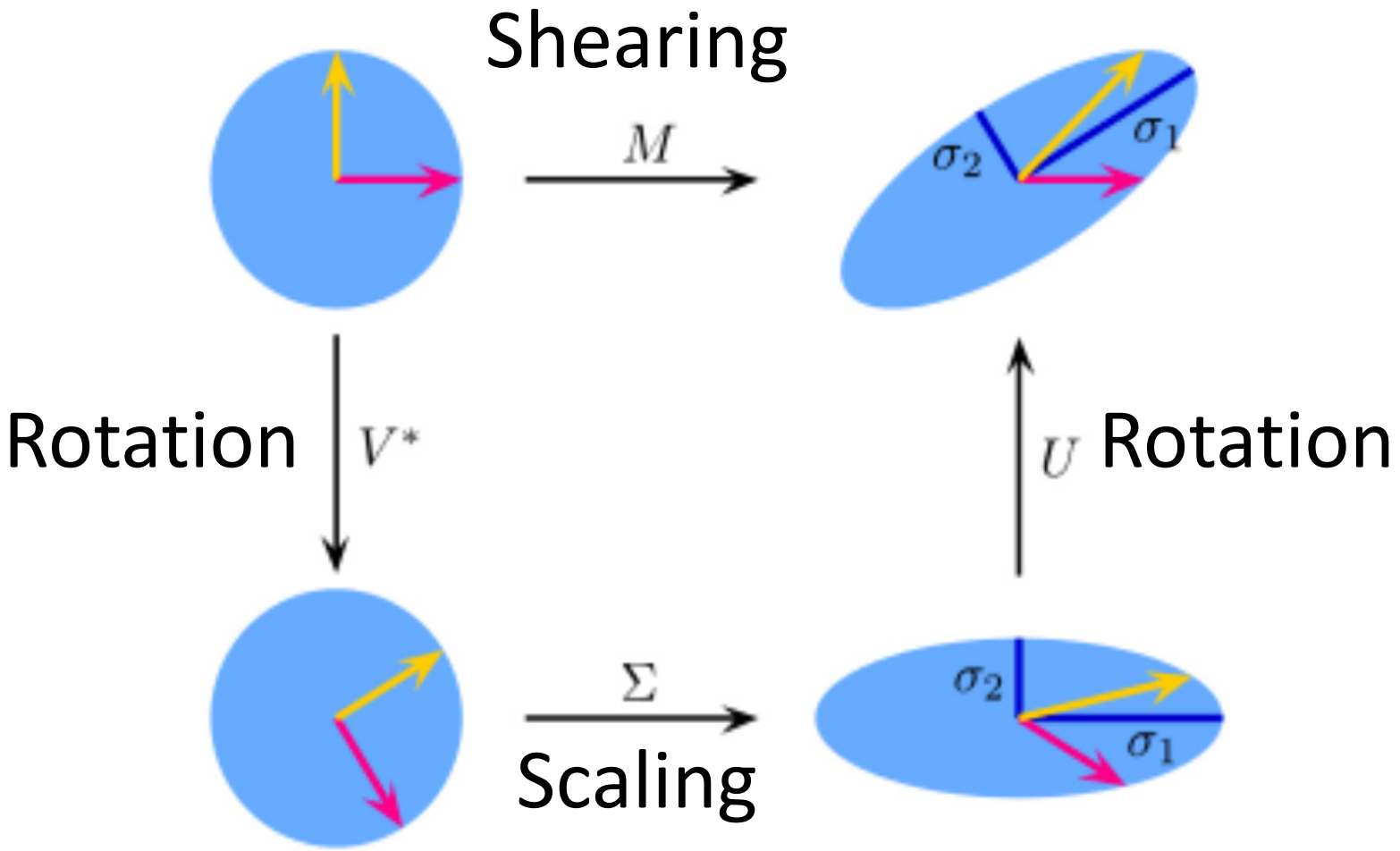
- **Unsupervised learning**
- **Supervised learning**

PCA, Dimensionality reduction





Geometric interpretation of SVD



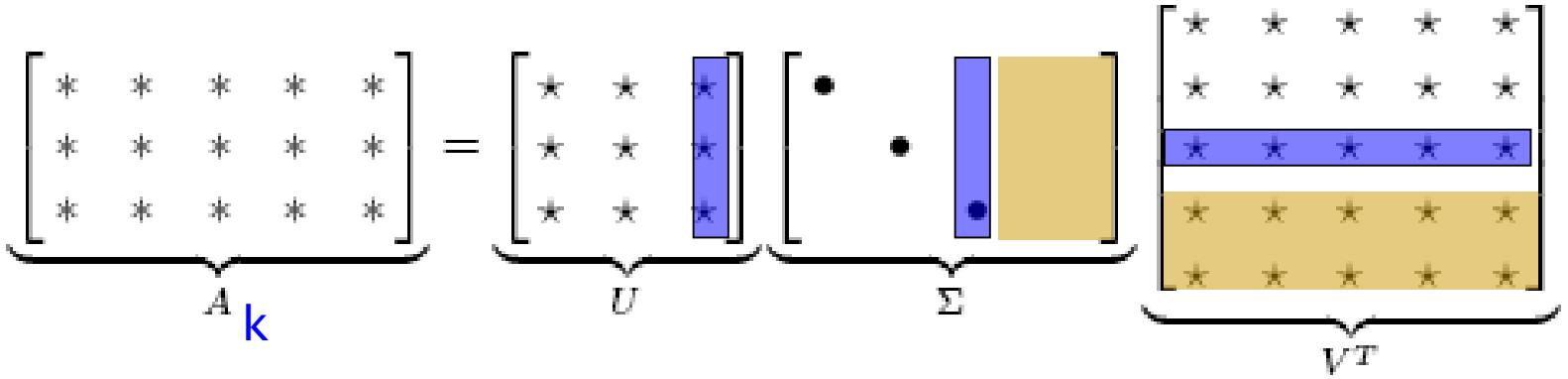
$$M = U \cdot \Sigma \cdot V^*$$

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

Low-rank Approximation

- Solution via SVD

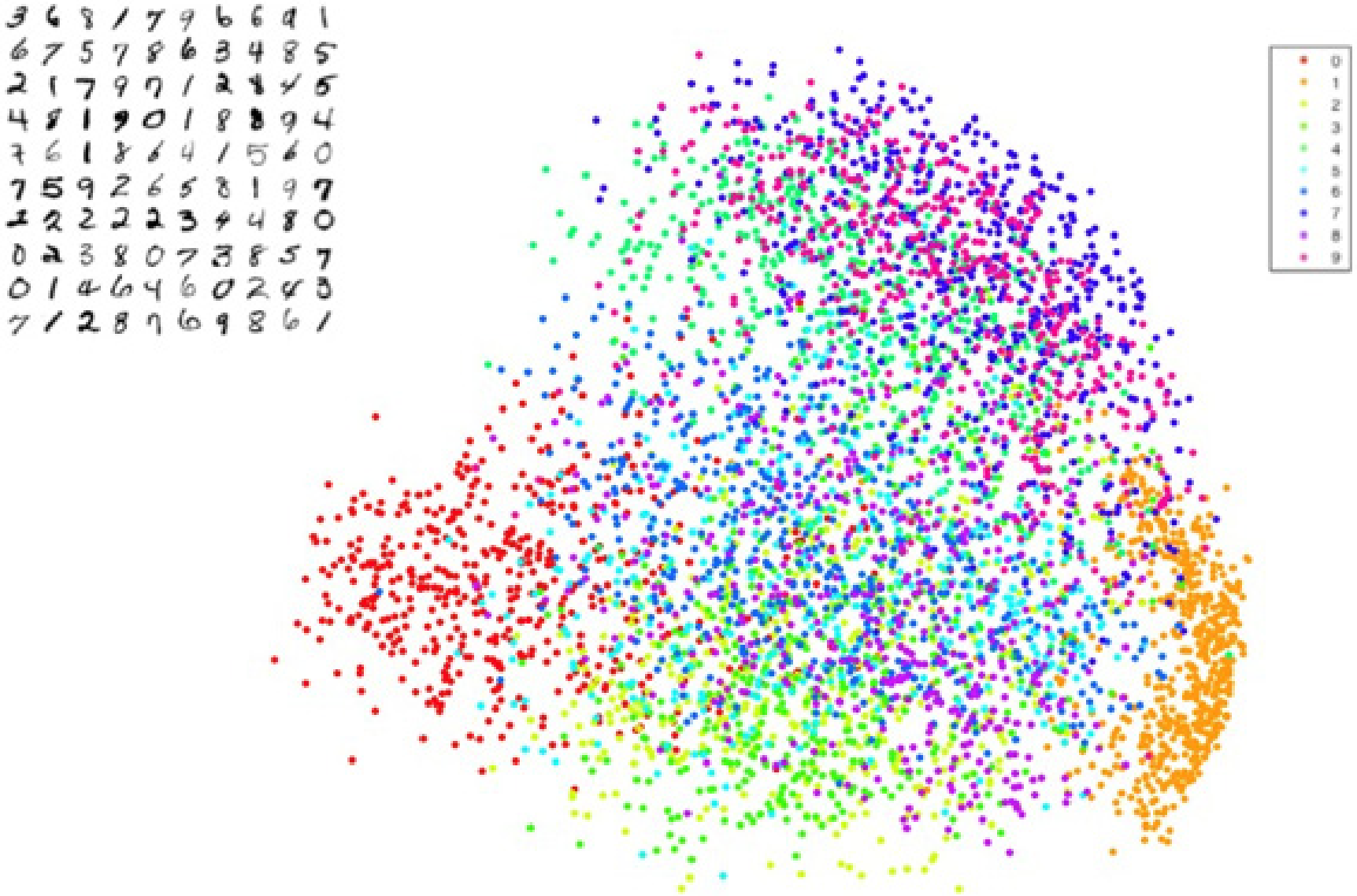
$$A_k = U\,\mathrm{diag}(\sigma_1,\ldots,\sigma_k,\underbrace{0,\ldots,0}_{\text{set smallest r-k singular values to zero}})V^T$$



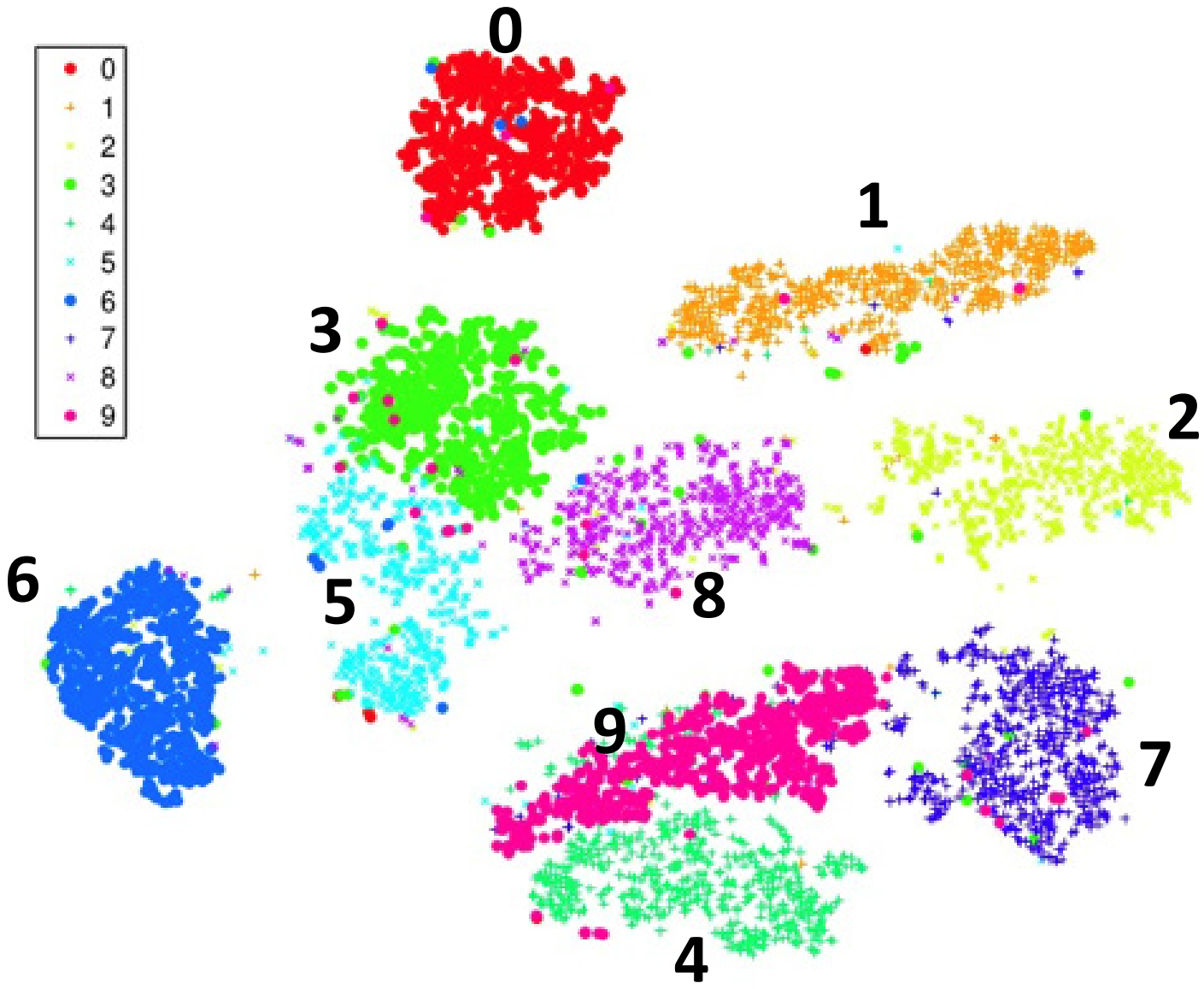
$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$ ← *column notation: sum of rank 1 matrices*

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

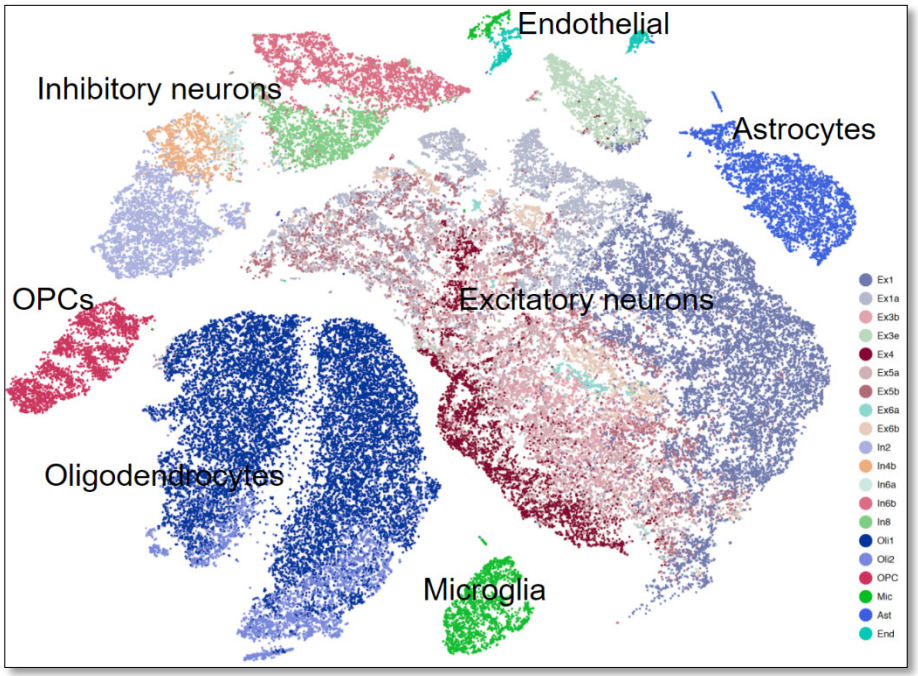
PCA of MNIST digits



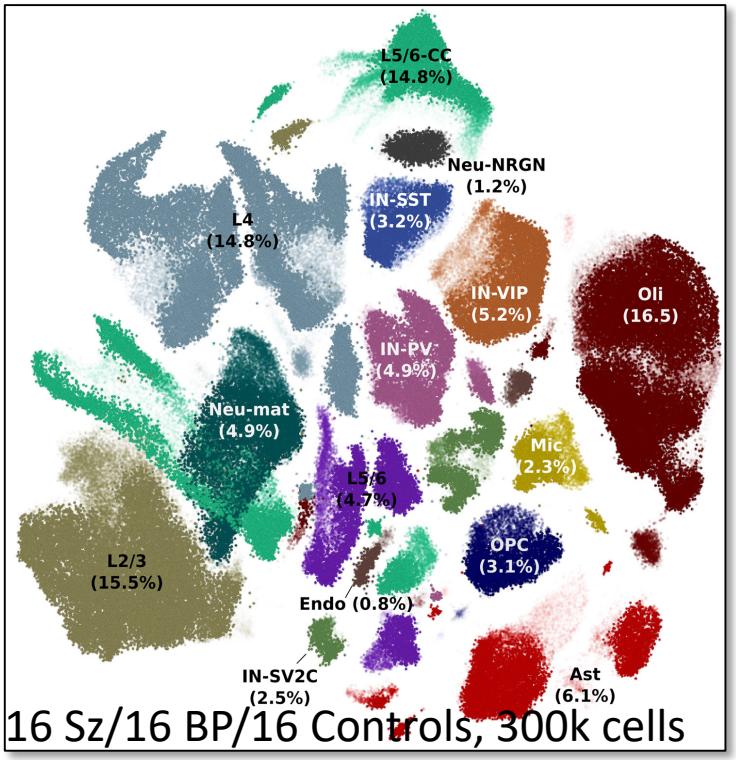
t-SNE of MNIST digits



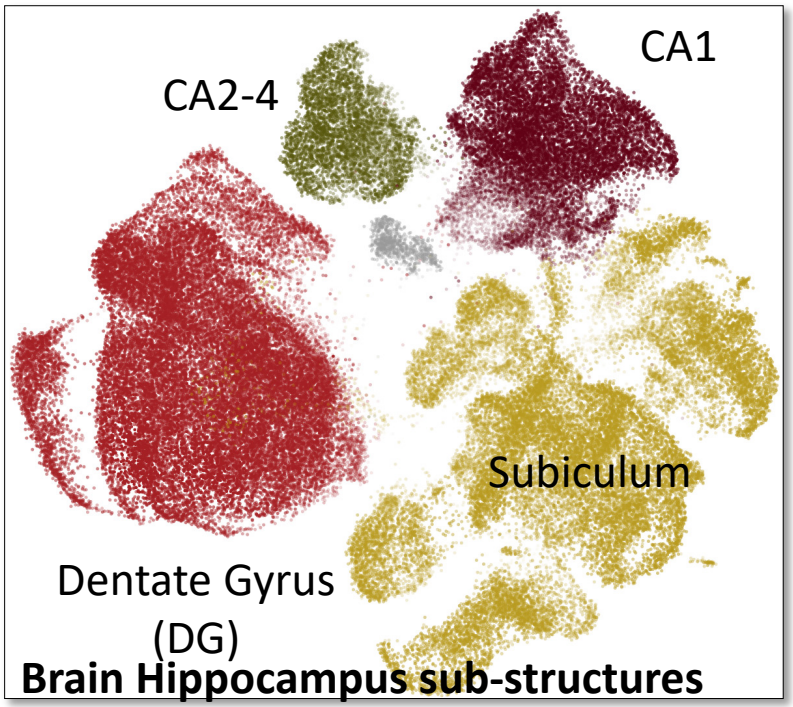
t-SNEs of single-cell Brain data



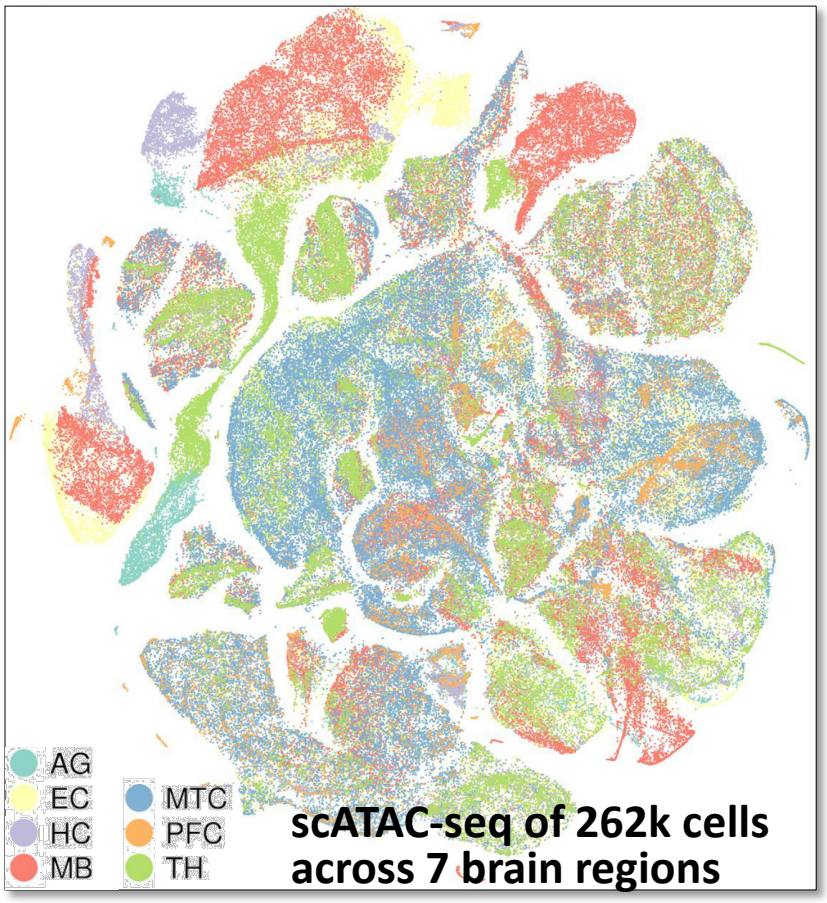
scRNA-seq in 48 individuals, 84k cells, Nature, 2019



16 Sz/16 BP/16 Controls, 300k cells

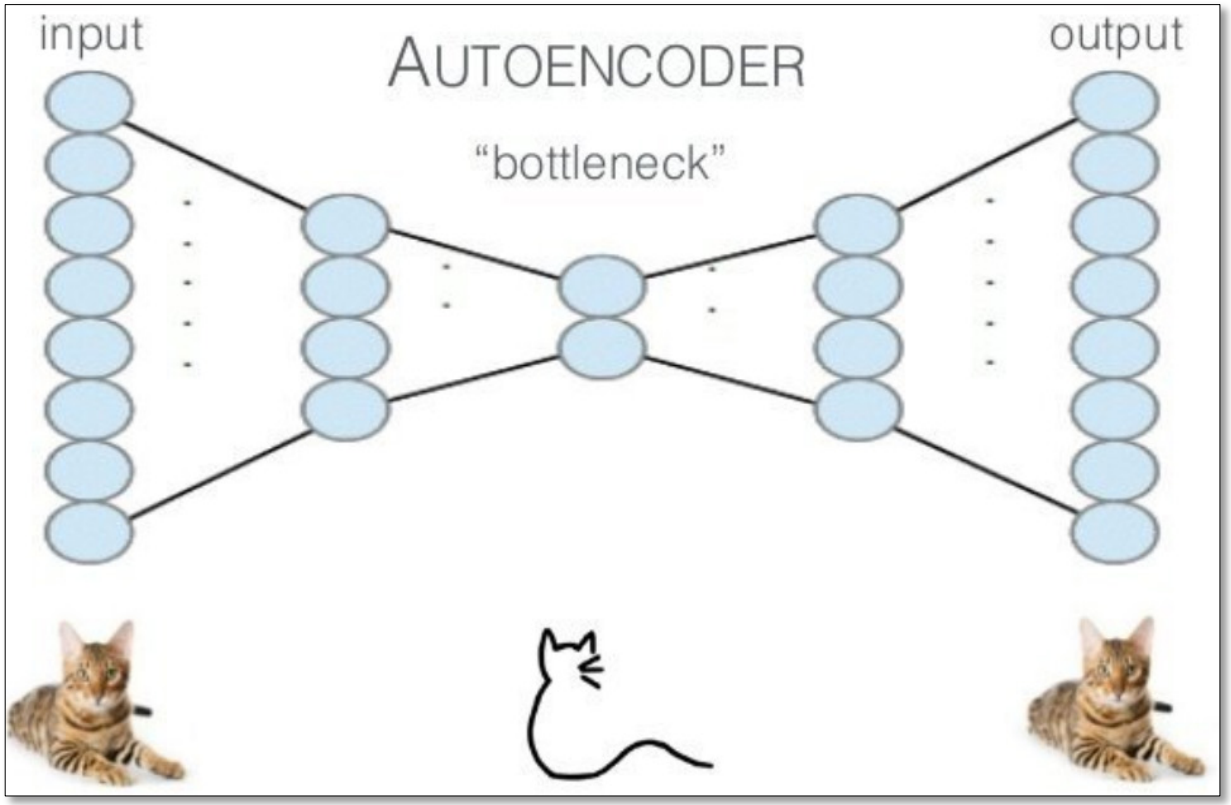


Brain Hippocampus sub-structures



scATAC-seq of 262k cells across 7 brain regions

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

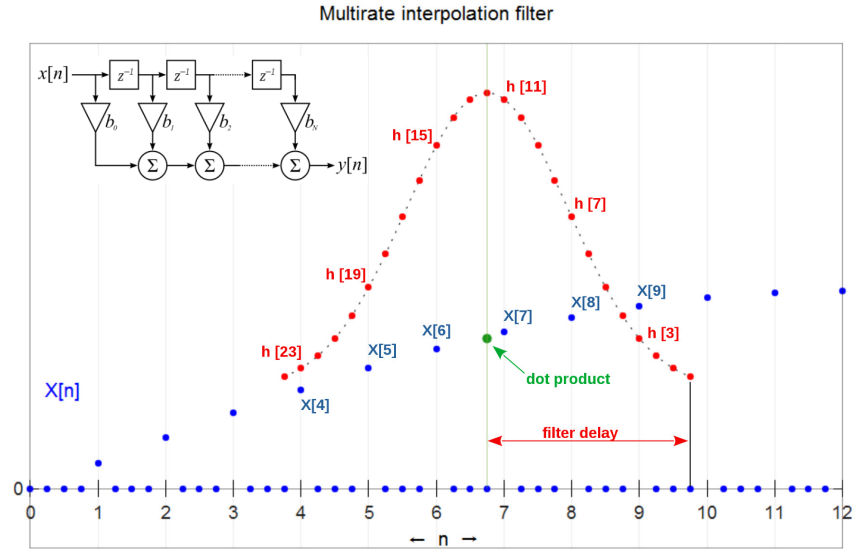
[Hinton *et al*, 2006]

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

1. Up-sampling gene expression patterns

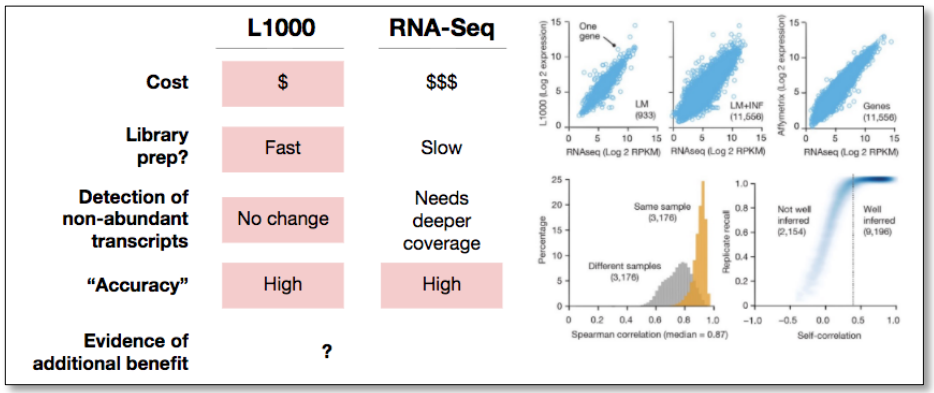
Challenge: Measure few values, infer many values



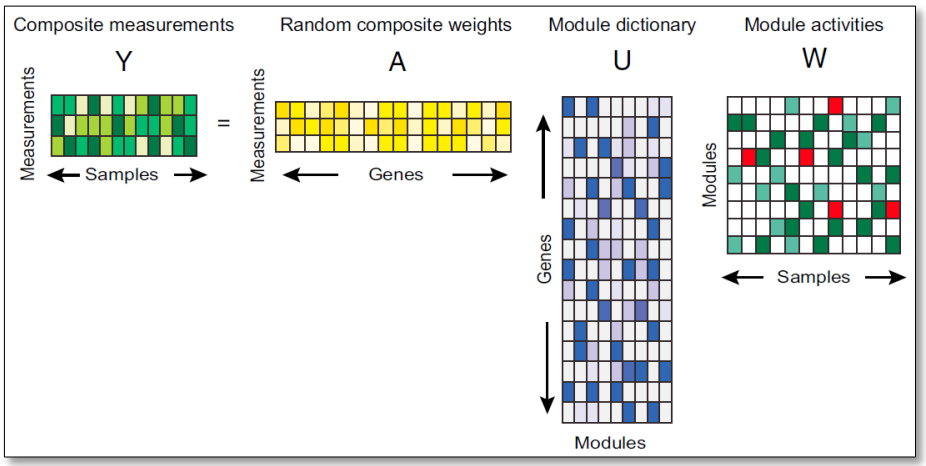
- Digital signal upscaling
  - Interpolating low-pass filter (e.g. FIR finite impulse response)
  - Low-dim. capture of higher-dim. signal
  - Nyquist rate (discrete) / freq. (contin.)

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

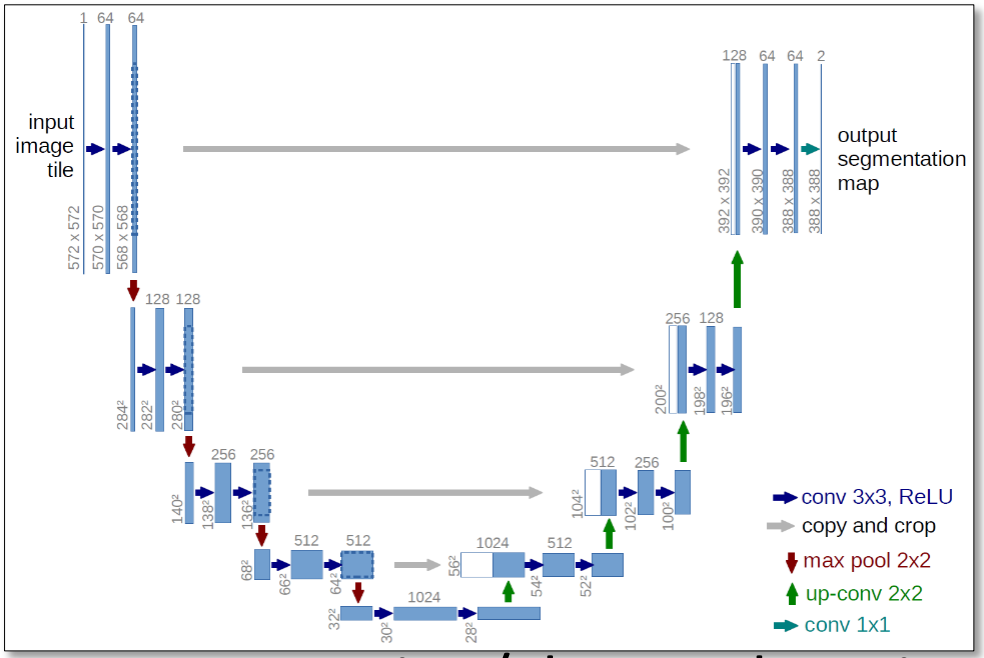


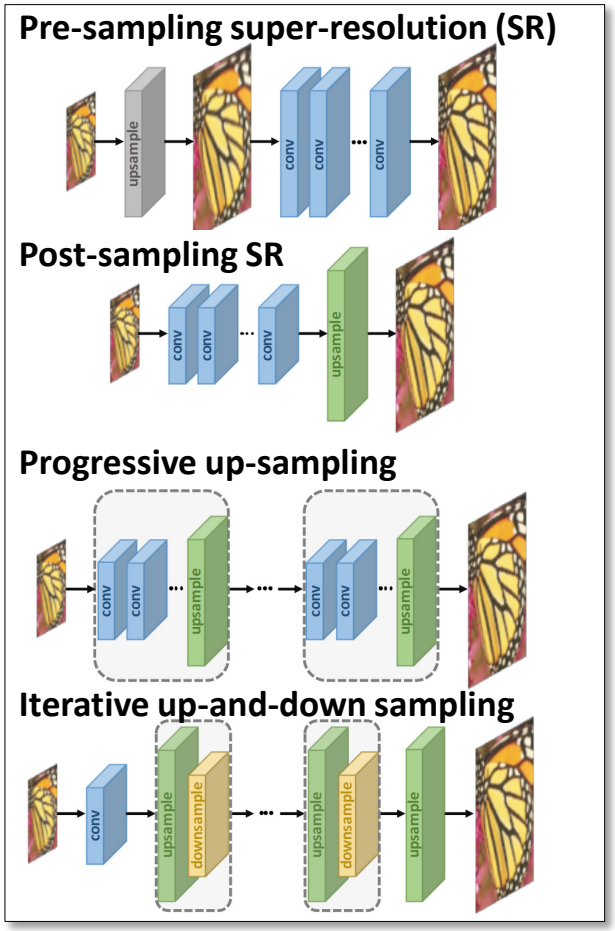
- Gene expression measurements
  - Measure 1000 genes, infer the rest
  - Rapid, cheap, reference assay
  - Apply to millions of conditions



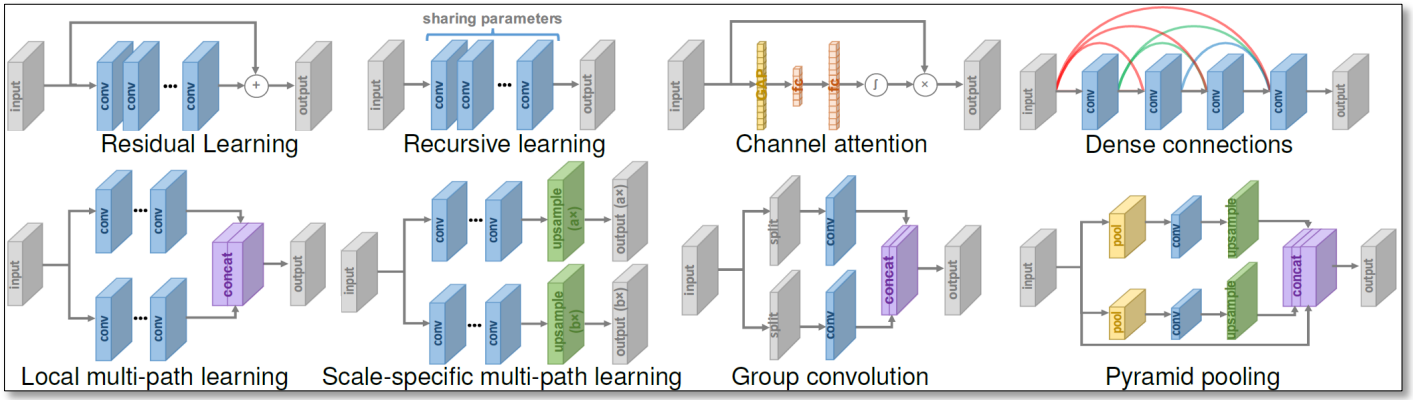
- ***Which*** 1000 genes? Compressed sensing
  - Measure few ***combinations*** of genes
  - Better capture high-dimensional vector

Deep Learning architectures for up-sampling images

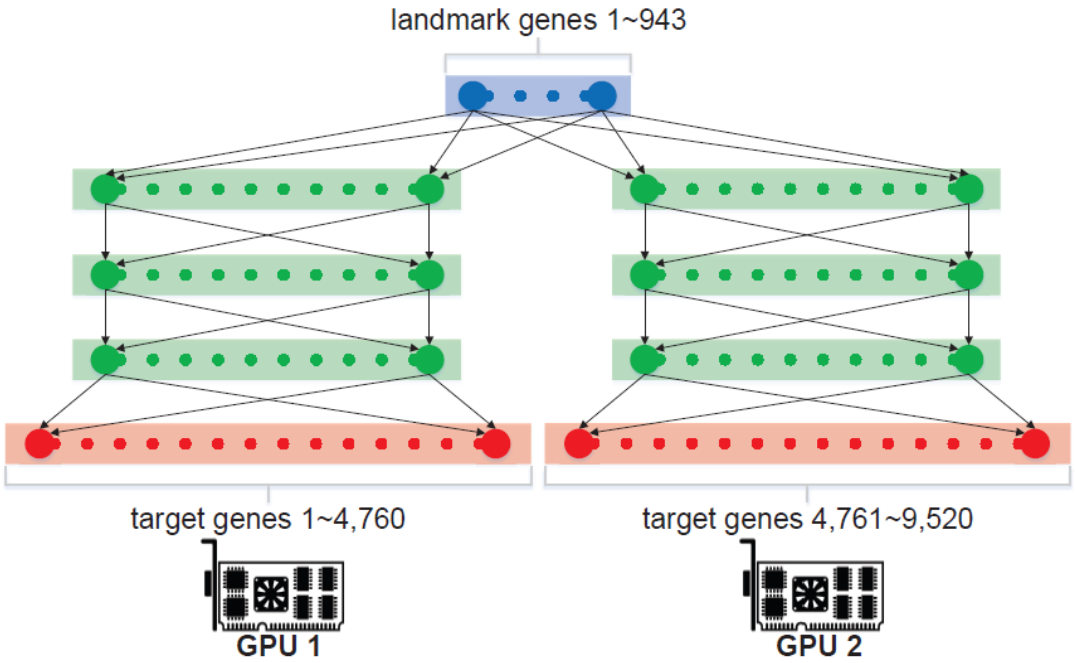




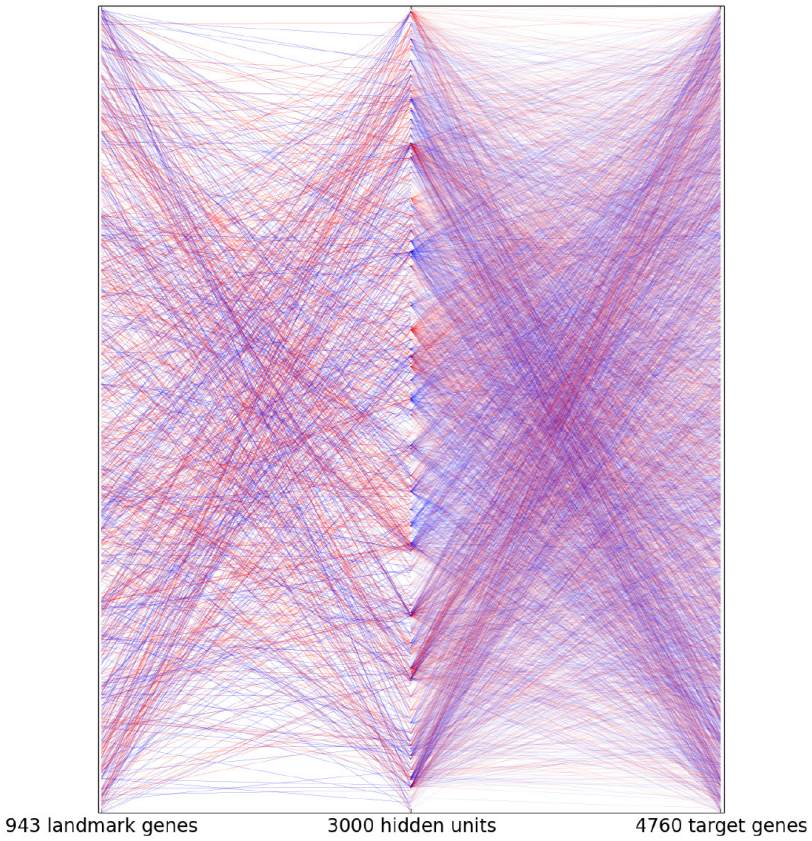
- Representation/abstract learning
  - Enables compression, re-upscaling, denoising
  - Example: autoencoder bottleneck. High-low-high
  - Modification: de-compression, up-scaling, low-high only



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 [a]	5e-4 or 3e-4
Minimum learning rate	1e-5
Learning rate decay factor	0.9
Learning scale [b]	3.0
Mini-batch size	200
Training epoch	200
Weights initial range [c]	$\left[-\frac{\sqrt{6}}{\sqrt{n_i+n_o}}, \frac{\sqrt{6}}{\sqrt{n_i+n_o}}\right]$



Gene expression inference with deep learning FREE

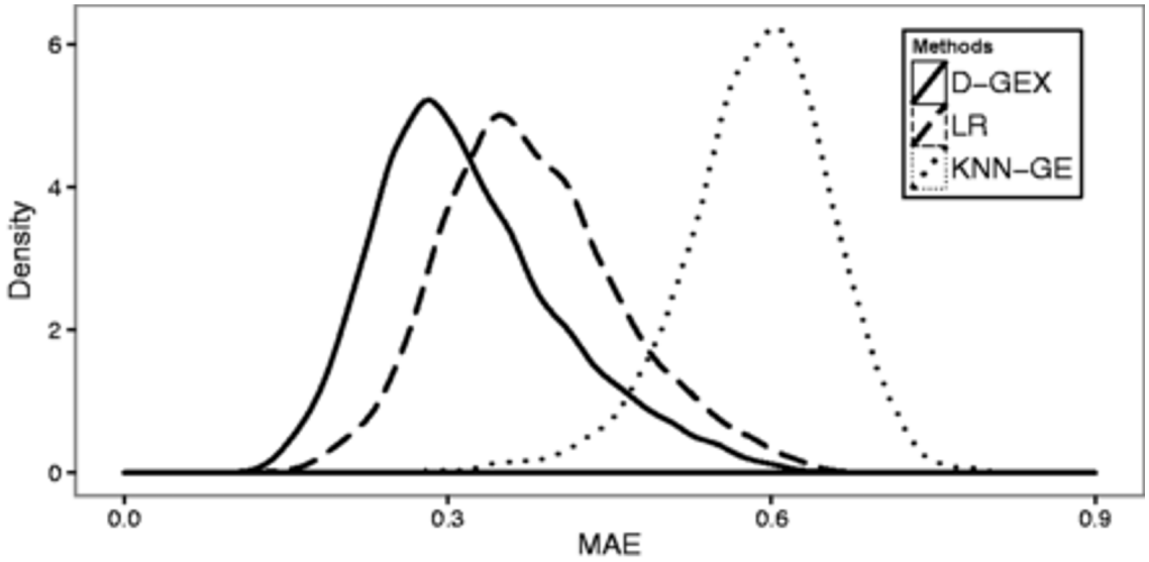
Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, Xiaohui Xie ✉ Author Notes

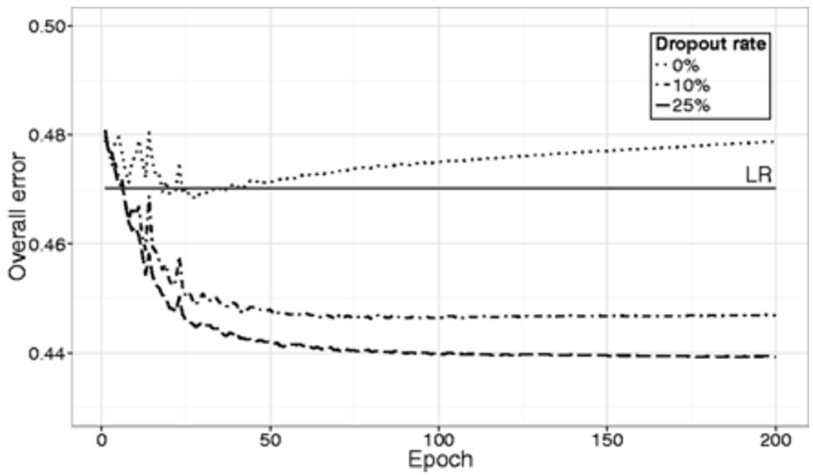
Bioinformatics, Volume 32, Issue 12, 15 June 2016, Pages 1832–1839,
https://doi.org/10.1093/bioinformatics/btw074

Published: 11 February 2016 **Article history** ▾

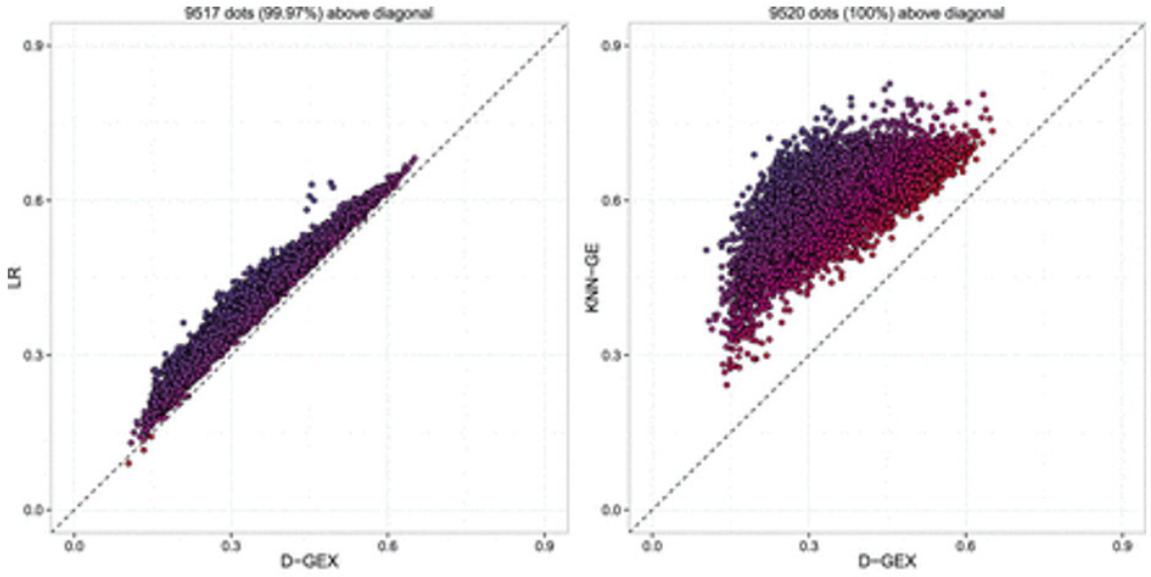
- Multi-task Multi-Layer Feed-Forward Neural Net
- Non-linear activation function (hyperbolic tangent)
- Input: 943 genes, Output: 9520 targets (partition to fit in memory)

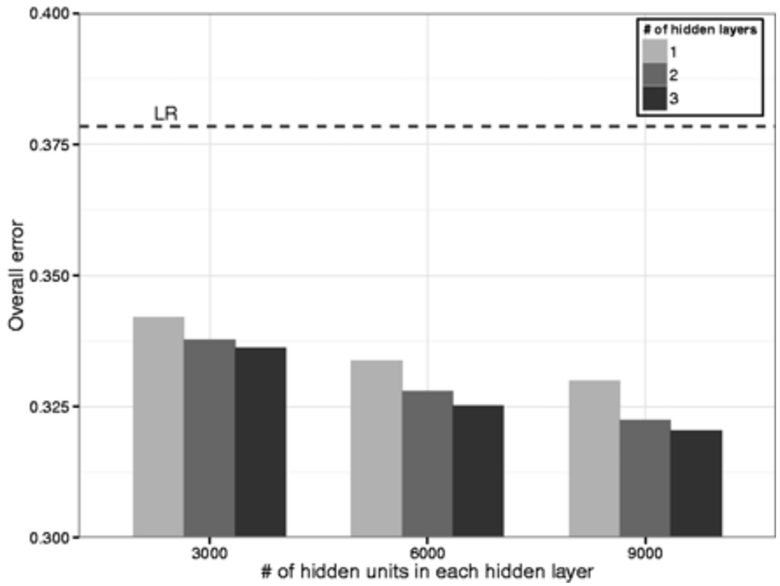
D-GEX outperforms Linear Regression or K-nearest-Neighbors





- Lower error than LR or KNN
- Training rapidly converges





- Strictly better for nearly all genes
- Deeper = better

However: performance still not great, computational limitations

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

2. Composite measurements for compressed sensing

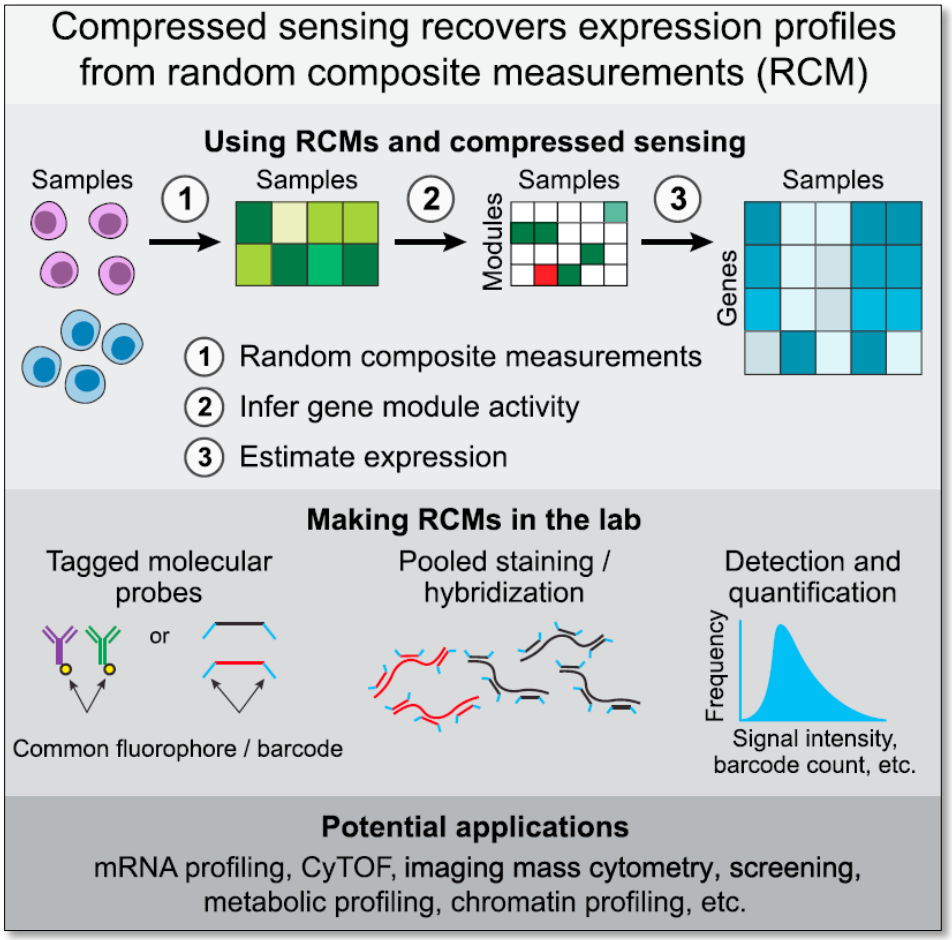
Key insight: Composite measurements better capture modules

Cell

Theory

Efficient Generation of Transcriptomic Profiles by Random Composite Measurements

Brian Cleary,[1,2] Le Cong,[1] Anthea Cheung,[1] Eric S. Lander,[1,3,4] and Aviv Regev[1,3,5,6,*]
[1]Klarman Cell Observatory, Broad Institute of MIT and Harvard, Cambridge, MA, USA
[2]Computational and Systems Biology Program, MIT, Cambridge, MA, USA
[3]Department of Biology, Massachusetts Institute of Technology, Cambridge, MA, USA
[4]Department of Systems Biology, Harvard Medical School, Boston, MA, USA



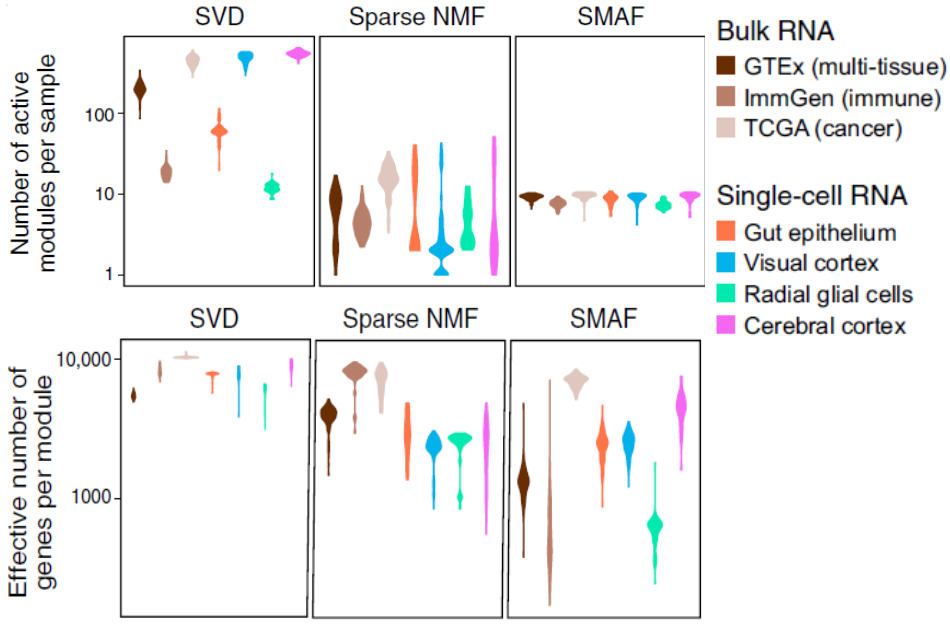
gene expression matrix $X \in \mathbb{R}^{g x n}$

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

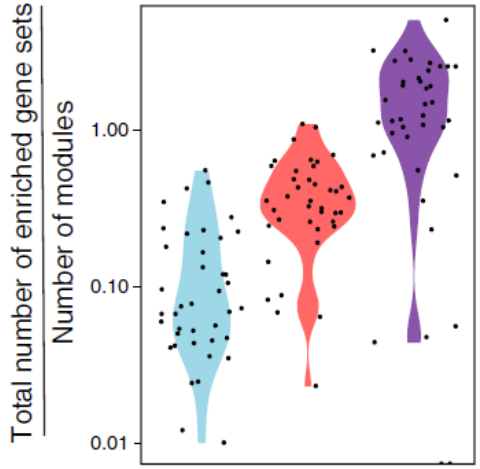
module activity matrix $W \in \mathbb{R}^{d x n}$

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

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



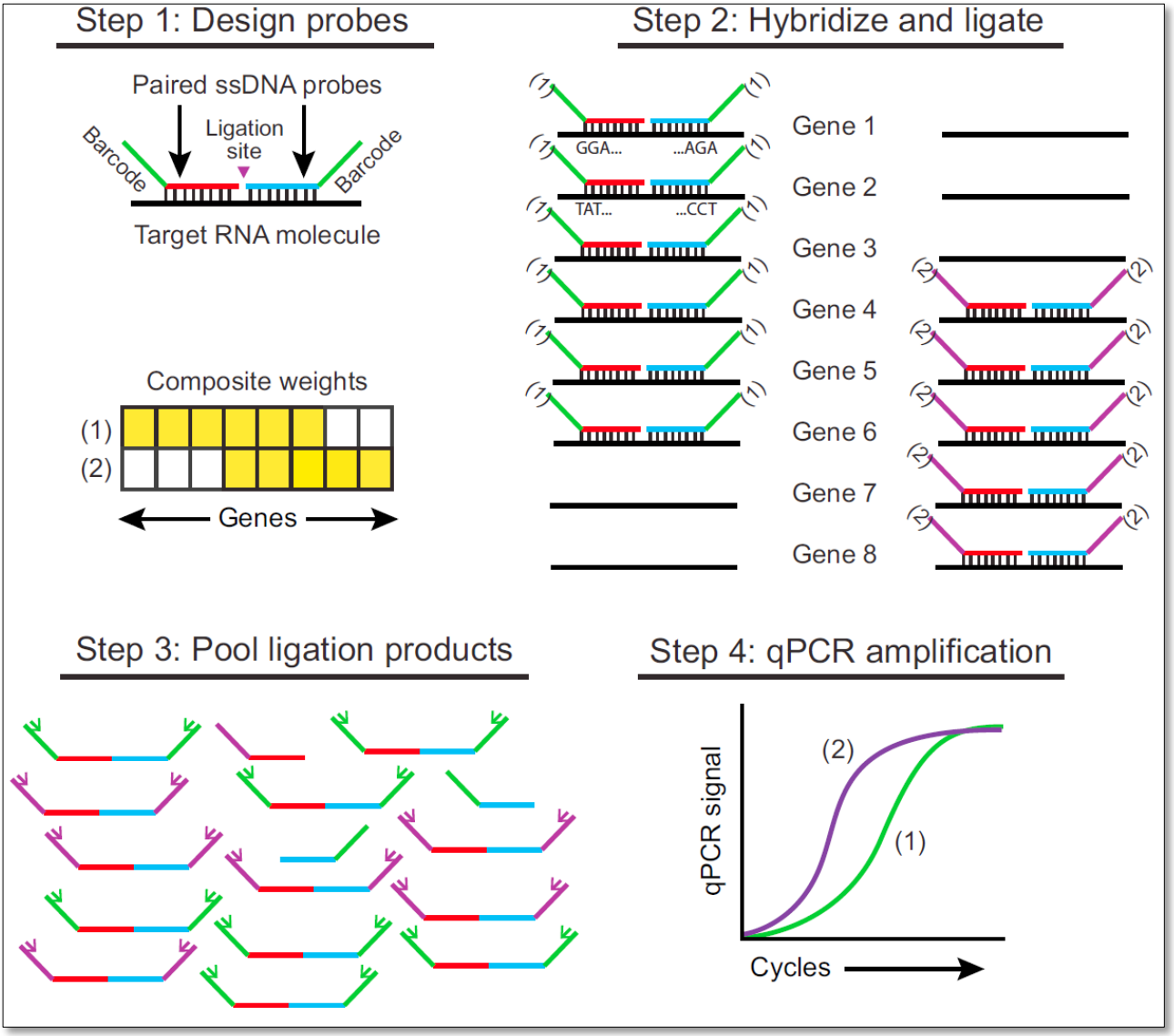
- Sparse Module Activity Factorization (SMAF)

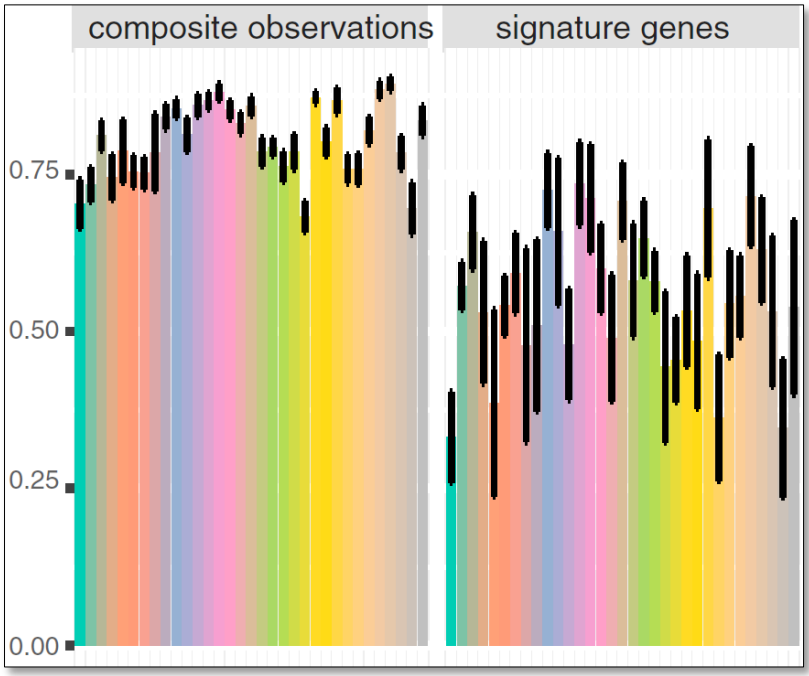


Algorithm: Sparse Module Activity Factorization

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

Making composite measurements in practice





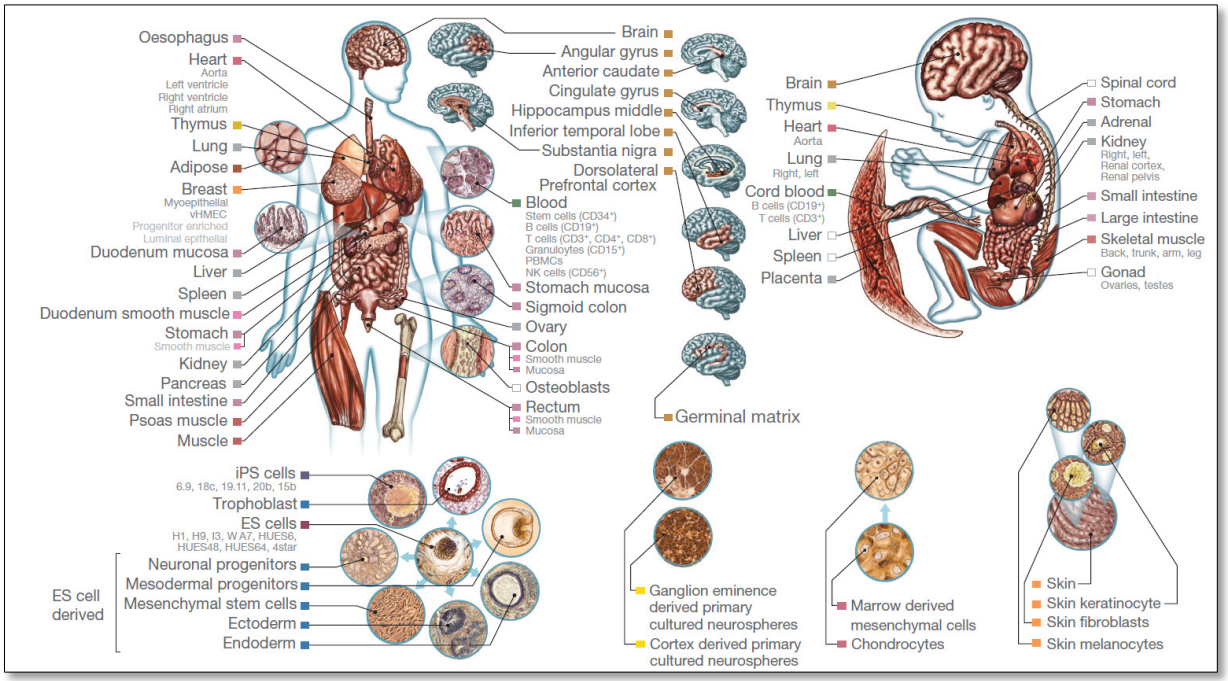
- Combinations of probes + barcodes for measurement
- More consistent signal-to-noise ratios

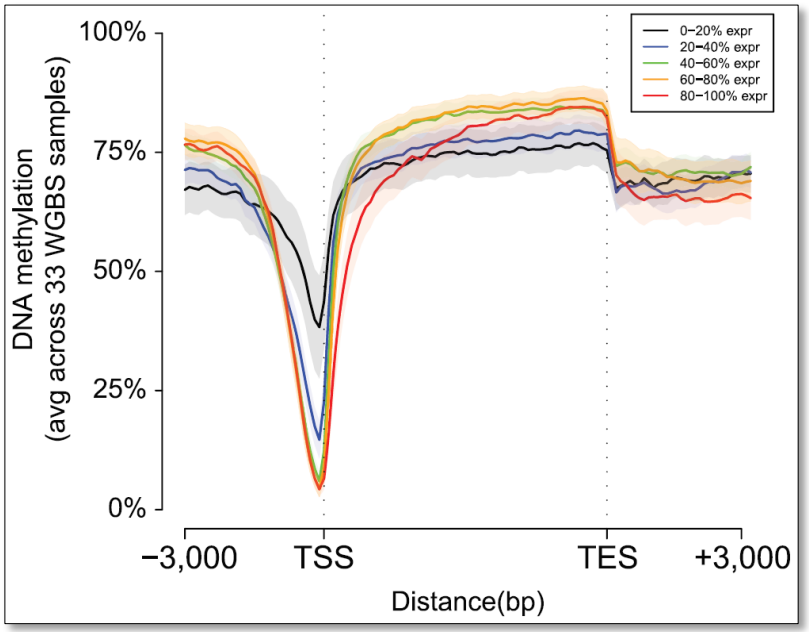
Today: Predicting gene expression and splicing

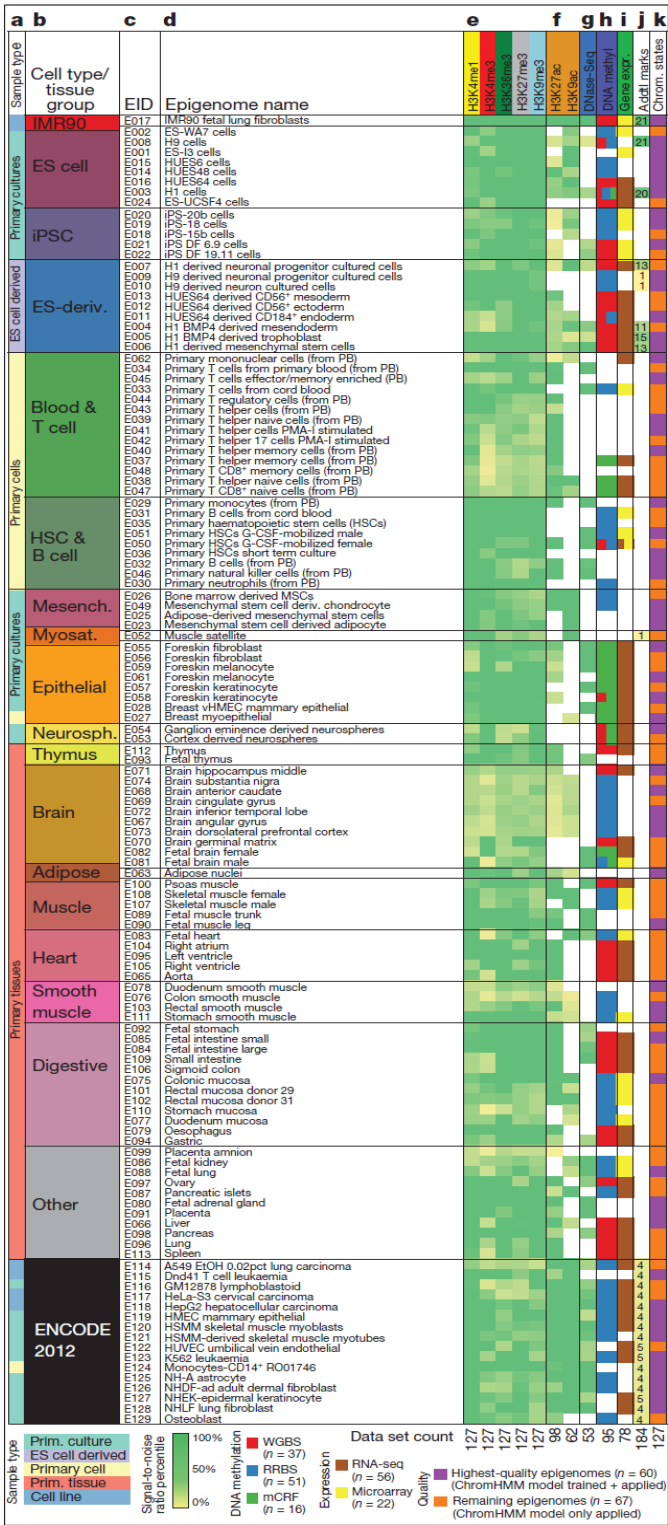
0. Intro: Expression, unsupervised learning, clustering
1. Up-sampling: predict 20,000 genes from 1000 genes
2. Compressive sensing: Composite measurements
3. DeepChrome+LSTMs: predict expression from chromatin
4. Predicting splicing from sequence: 1000s of features
5. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

3. Predicting Expression from Chromatin

Can we predict gene expression from chromatin information?

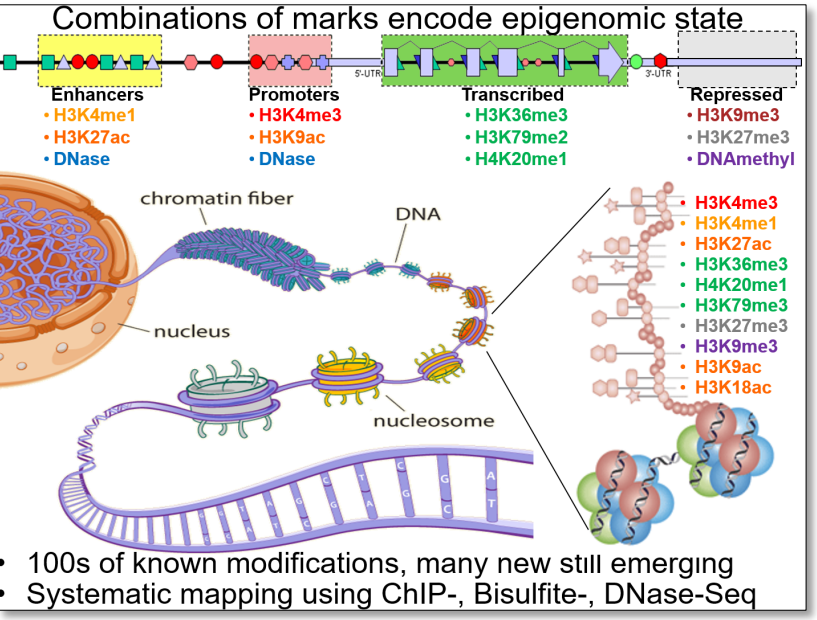


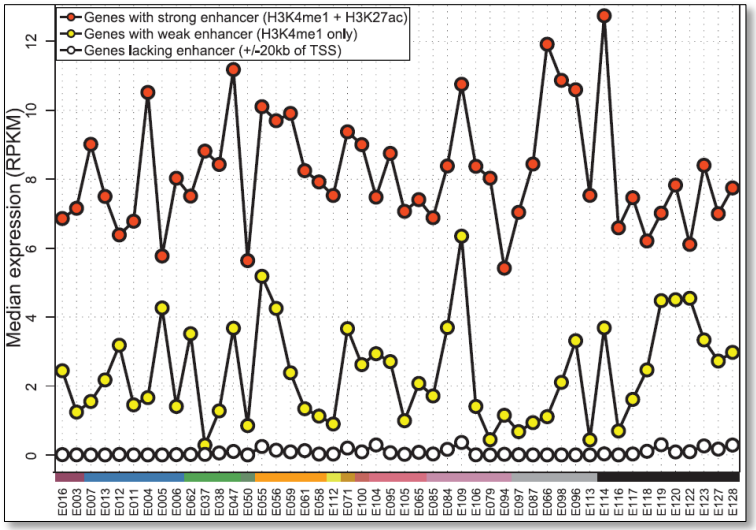


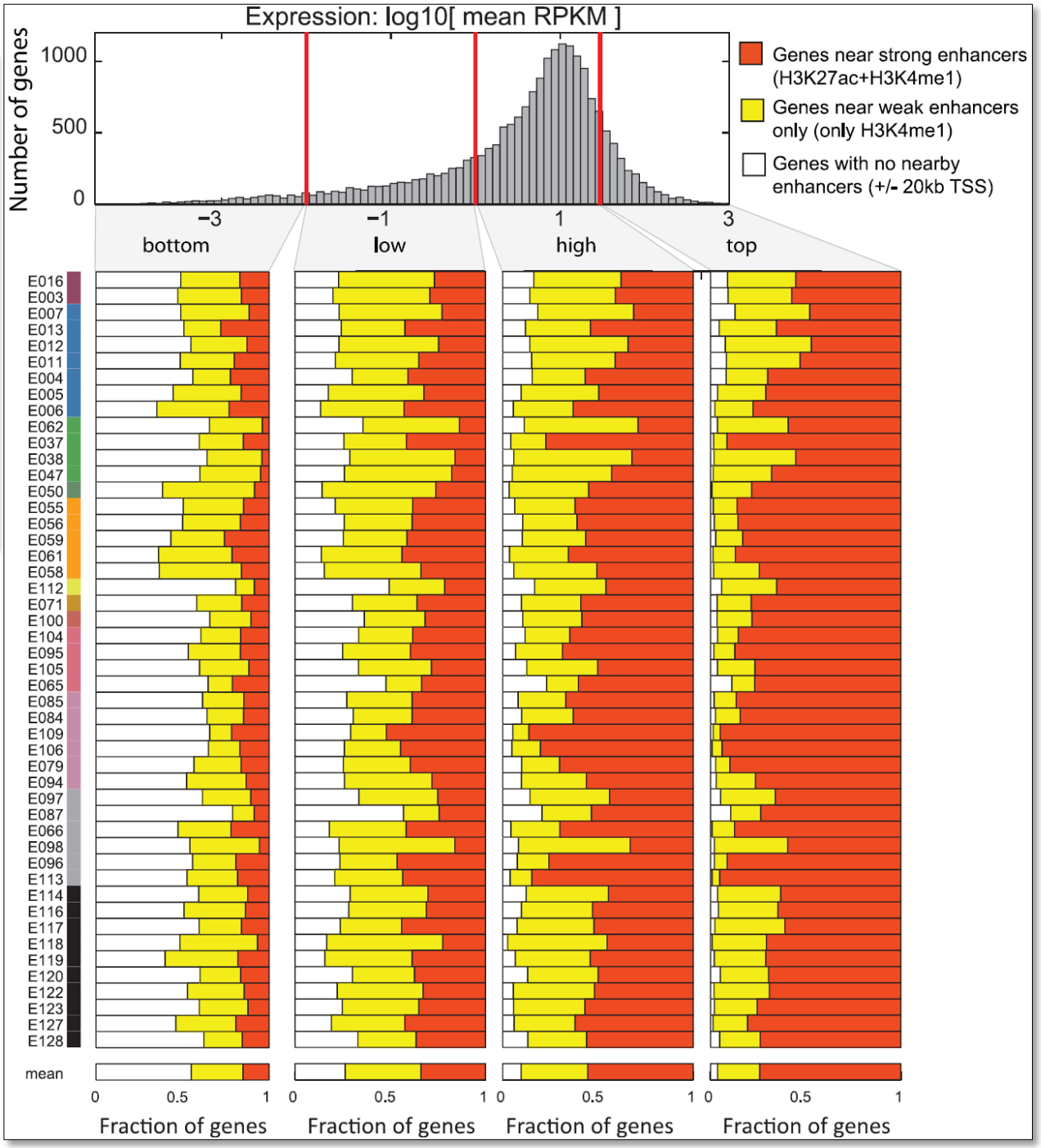


- DNA methylation vs. gene expression
- Promoters: high. Gene body: low

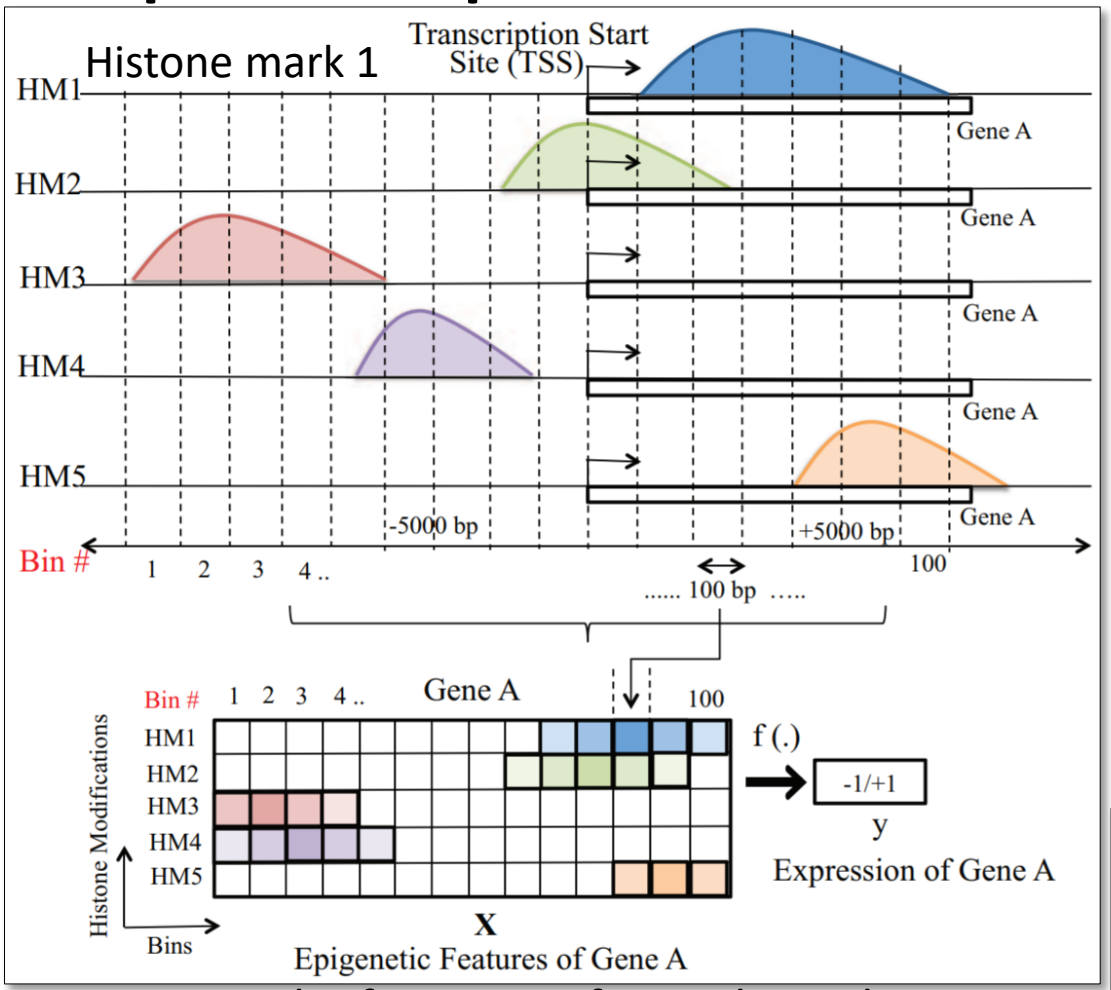
Strong enhancers (+H3K27ac) vs. weak enhancers (H3K4me1 only)



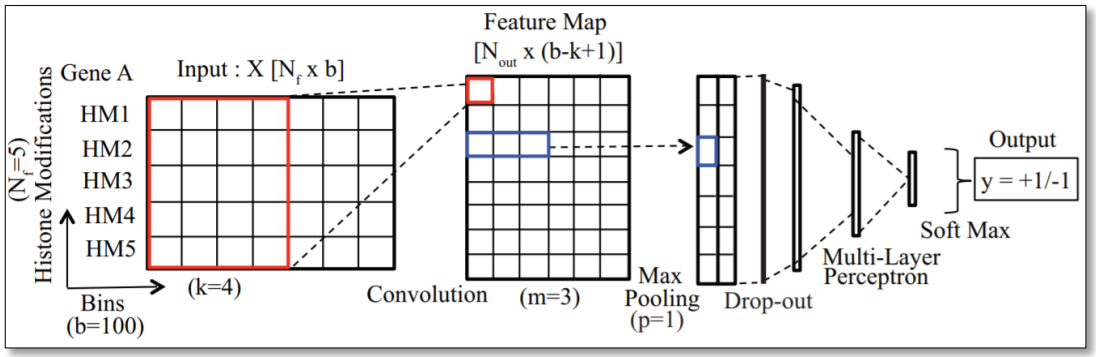




DeepChrome: positional histone features predictive of expression



- Positional information for each mark



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

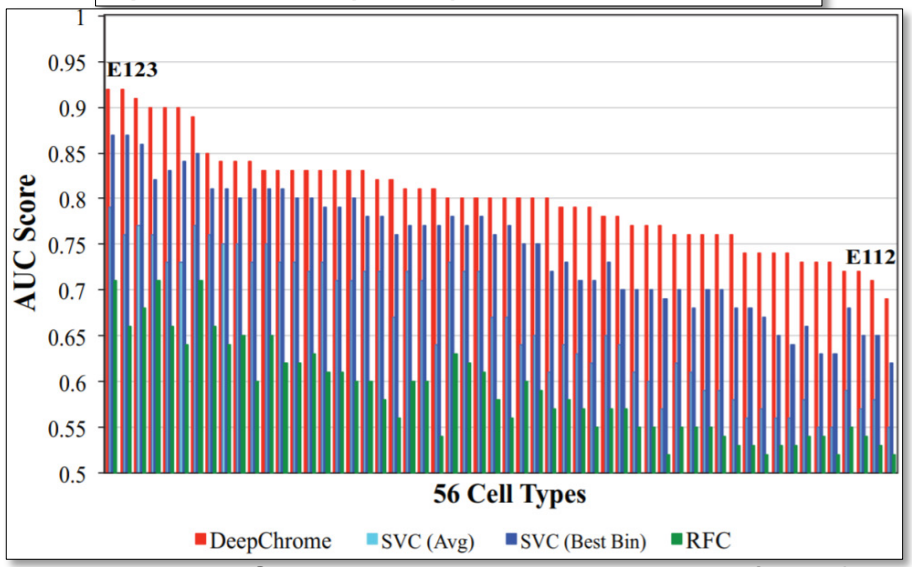
DeepChrome: Deep-learning for predicting gene expression from histone modifications.

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

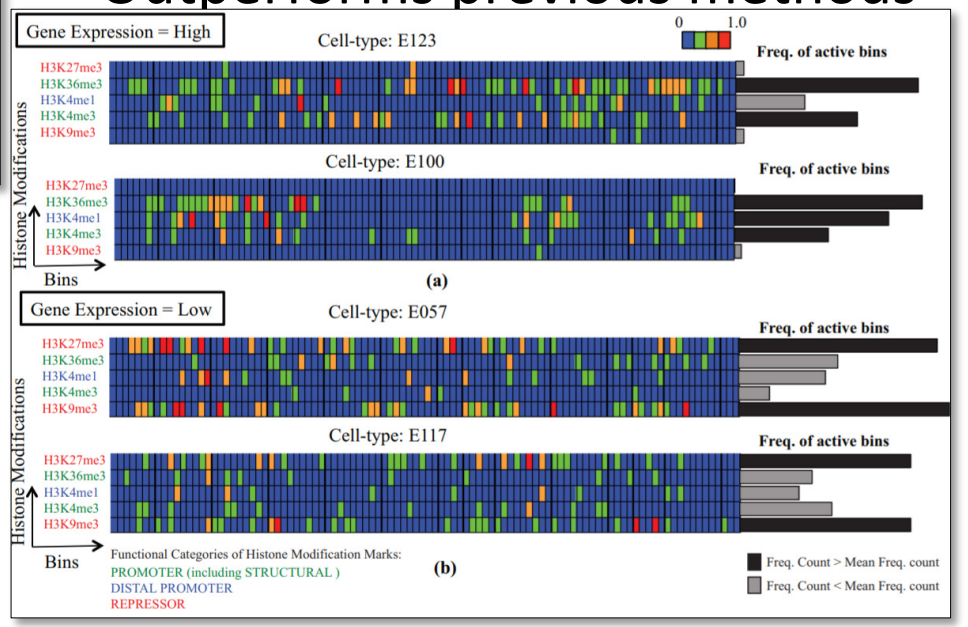
Department of Computer Science, University of Virginia, Charlottesville, VA, U.S.A

* To whom correspondence should be addressed.

This work will be published originally in Bioinformatics Journal at
http://bioinformatics.oxfordjournals.org

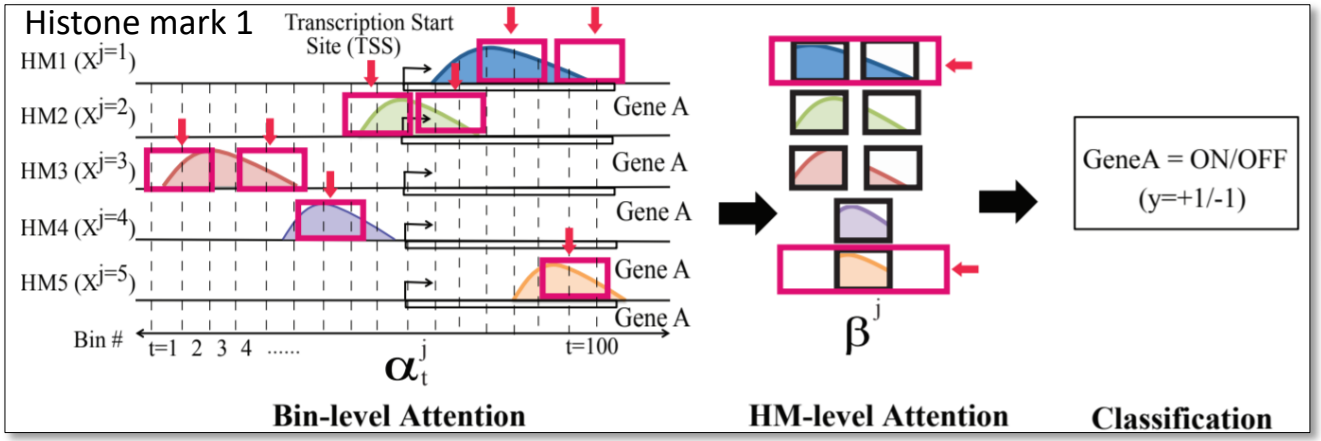


- Outperforms previous methods



- Meaningful features selected

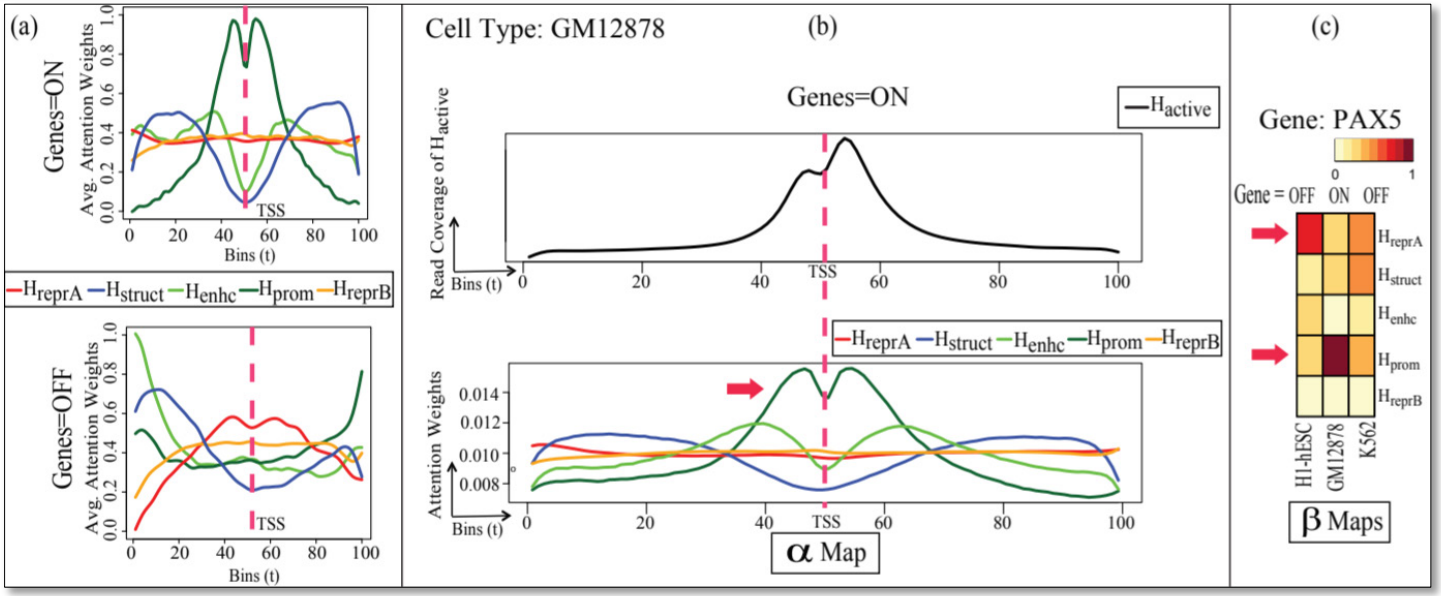
AttentiveChrome: Selectively attend to specific marks/positions



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

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

- Consistent improvement over DeepChrome

Guest lecture: Xiaohui Xie

Deep Learning for Expression/Chromatin Prediction

Xiaohui Xie
Professor, UC Irvine

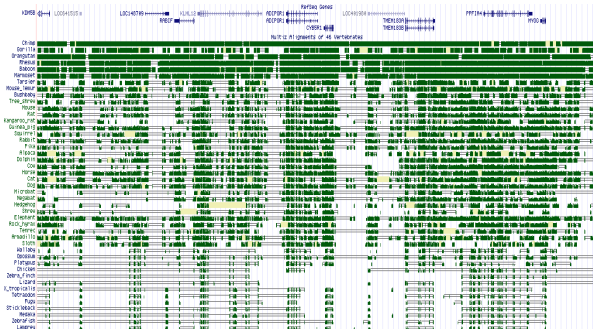
Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

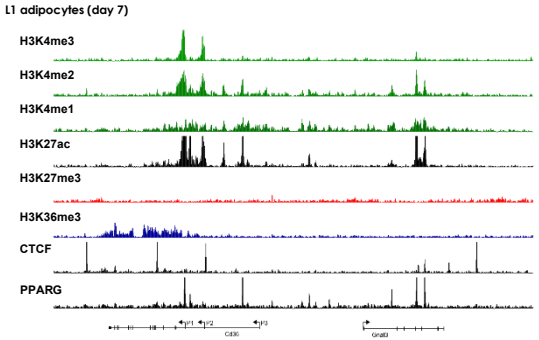
4. Predicting Reporter Expression from Chromatin Features

We can find regulatory elements … but we don't know how to read them

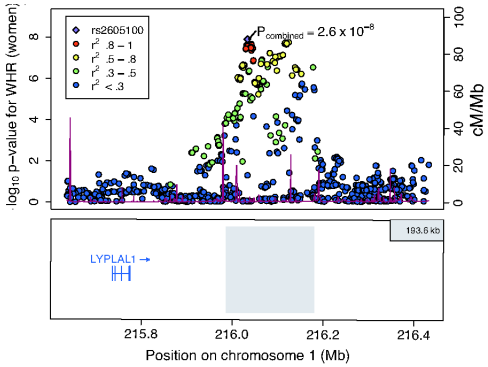
Comparative sequence analysis



Genome-wide chromatin/TF mapping



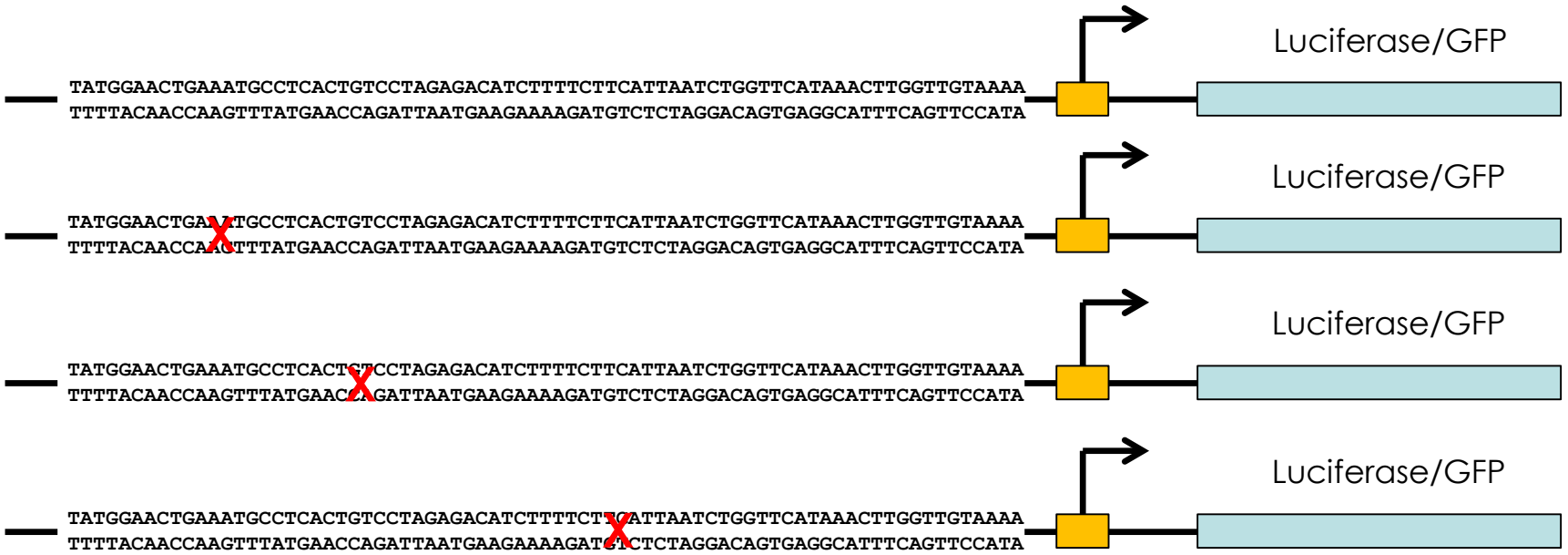
Genetics



100s of megabases of likely cis-regulatory elements

TATGGAACTGAAATGCCTCACTGTCCTAGAGACATCTTTTCTTCATTAATCTGGTTCATAAACTTGGTTGTAAAAAGCAAATTC
AAAGAATTCATCTAATAATGACAGAAAAAGAAACATTTCTGAATGAATTTGTGGAAGTGTACAATTTAATTTTTCATTTAATTT
TTATTTACAATTTTCATTTAATTTTCATATTAATTTTTCCTCTTTAATCCCATGCAAGGACACAGCAGTTAATAAATACTTAAG
AAATACTTCTCAATTGCATACCATTCTTA??????????GGATTTTATAAATCCAGTTATTAGTATCT
CAAACTTTAAATAATCAAAAGTCAATGGCAG??????????AGGTATCAACTCACCCACCCACTCACCCC
AAAGAGACTTGTCGGTACGCCCTCTGGTGAC??????????TAAATTTGAGAATGTGTAGCCCTCCAGTTCC
CAACCTTCATTCCATCTTTTTTTTTTTTTTGA??????????AACAGGTATATTAGAAAATACCAAATTCTCT
TAGAGAAAAAAAAATTGCACAACAATCTCAACACTGAGAGGACTGTCCAAGGATGAAAGCAAGGTACTCCTAGGTTAGTAGTTT
TTCAAACCTATAGATGGCATTTGGAAAGAAGTACACGCATAGGCTTCAAAAAACCCTGAACACCAGAAACCTAATTAGCACTTT
CATTTTAGGGGCACGGGCTTTCCCTTATCCTTTTCTACAGCCAACCTTAATGGGGTCCAATTAAGGCAACAATGAAAGGTTACA
AAAGTCTACAAAAAGGTCCCATTACAGCATTTAAACAGGGTTCATTTTTTTTAATCAAATAGTTTACACTGTTTTTTTGTTGTT

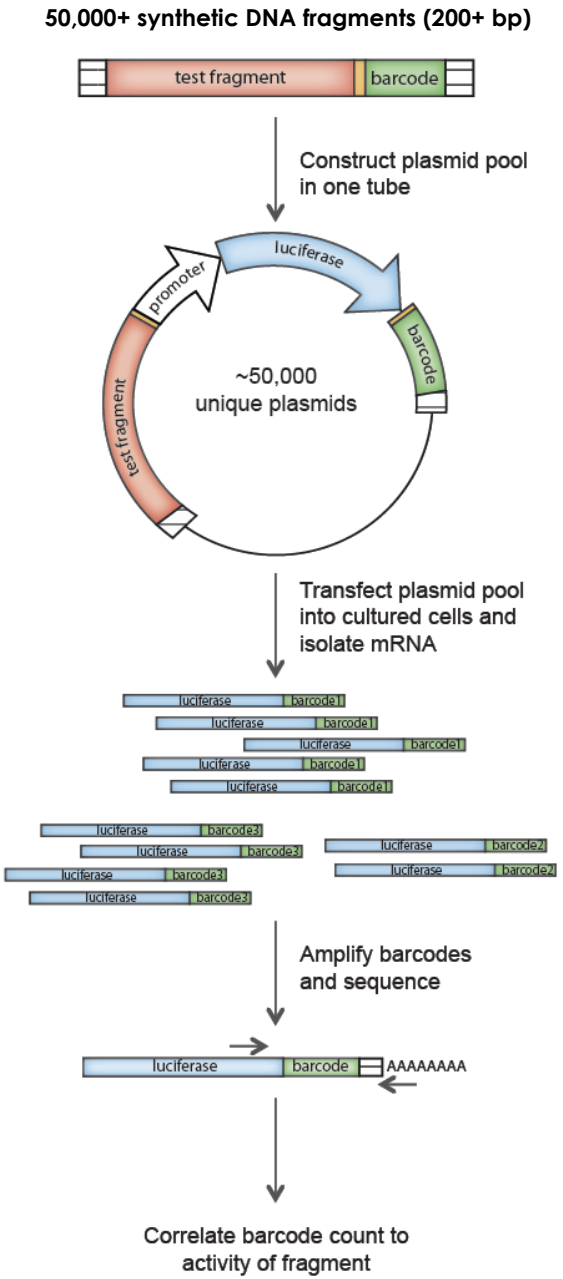
Traditional regulatory element “bashing”



Bottlenecks:

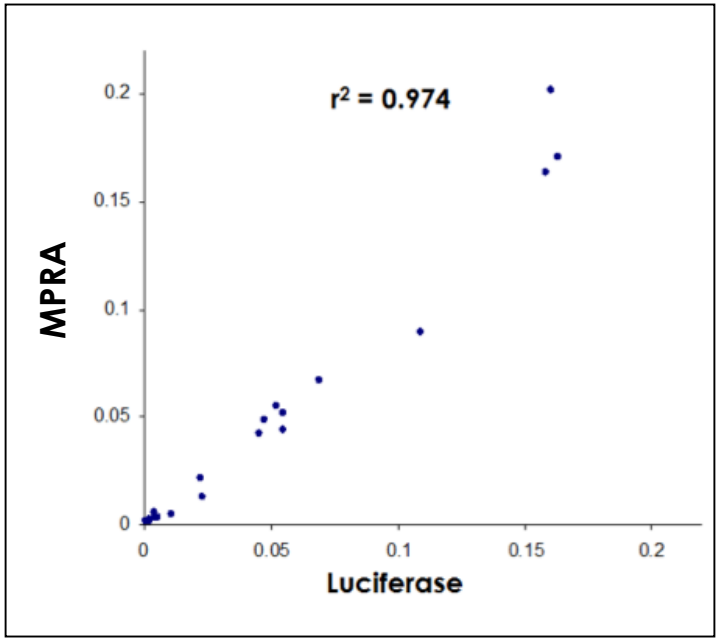
1. **Generating/cloning individual variants is tedious**
2. **Enzymatic/fluorescent reporters limit multiplexing**

Massively Parallel Reporter Assays (MPRA)



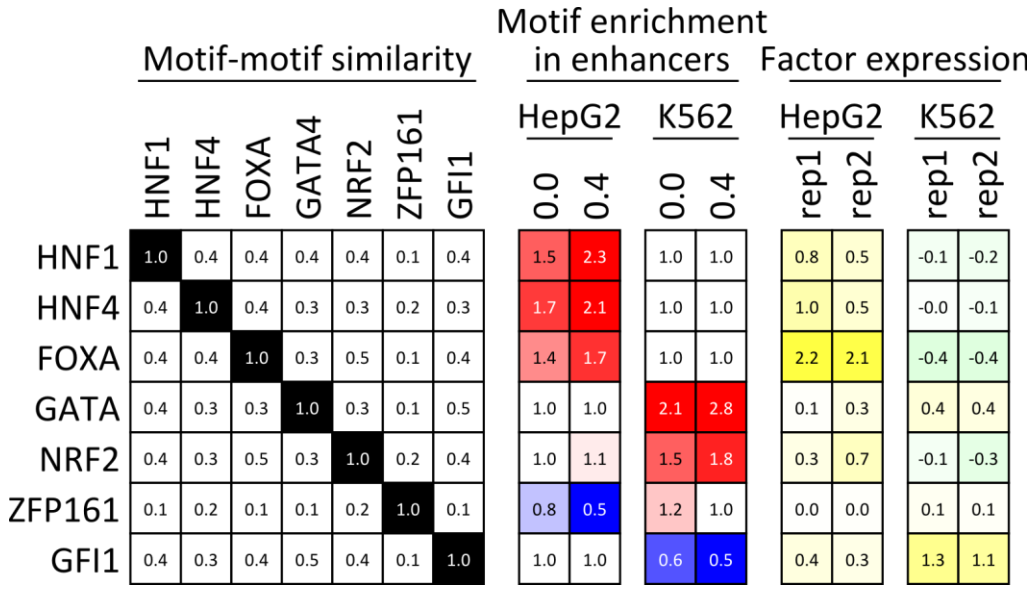
- Flexible assay format:
  Promoters, enhancers, silencers,
  Insulators, RNA stability elements, ++

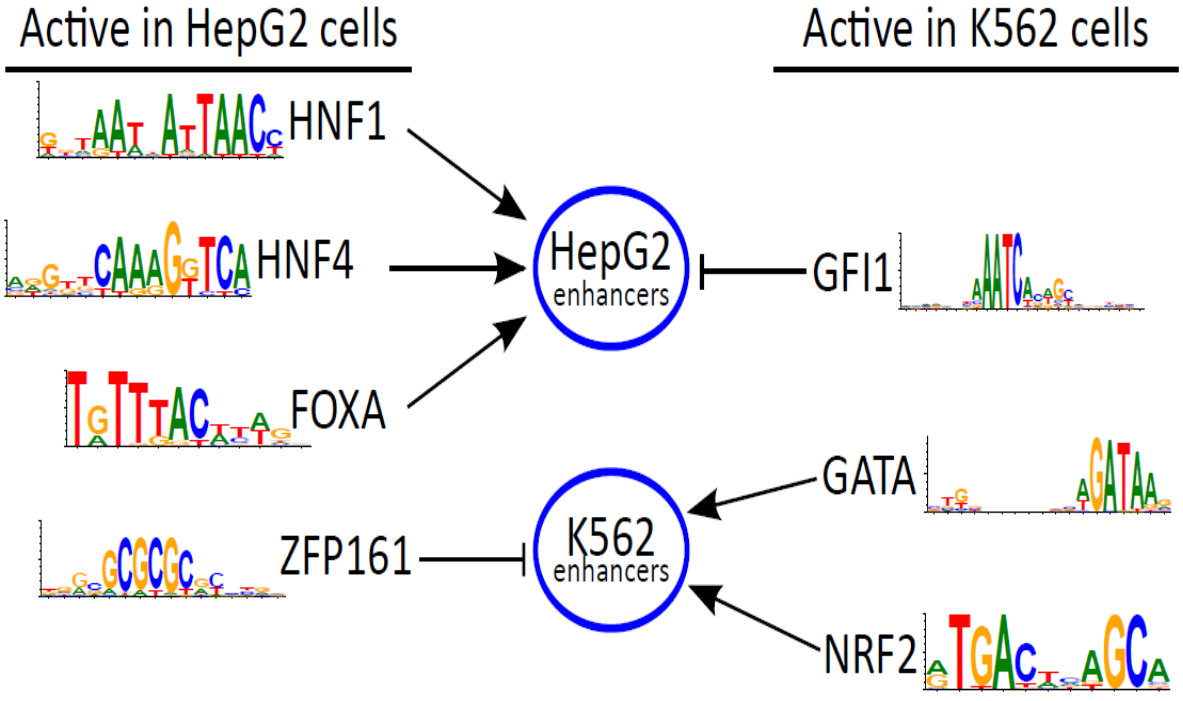
- Data is directly comparable to traditional reporter assays:

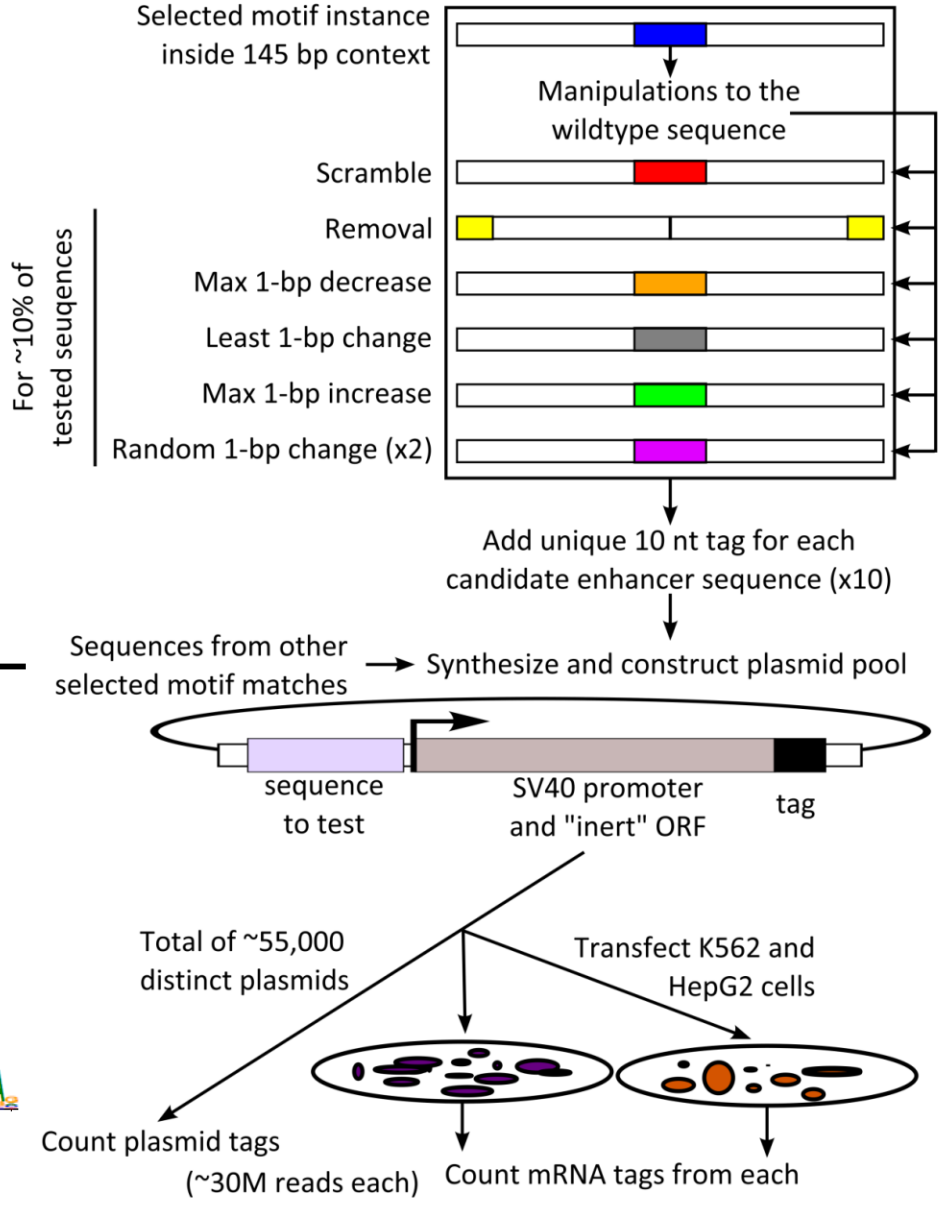


- Throughput increased by 3 orders of magnitude

Systematic motif disruption for 5 activators and 2 repressors in 2 human cell lines

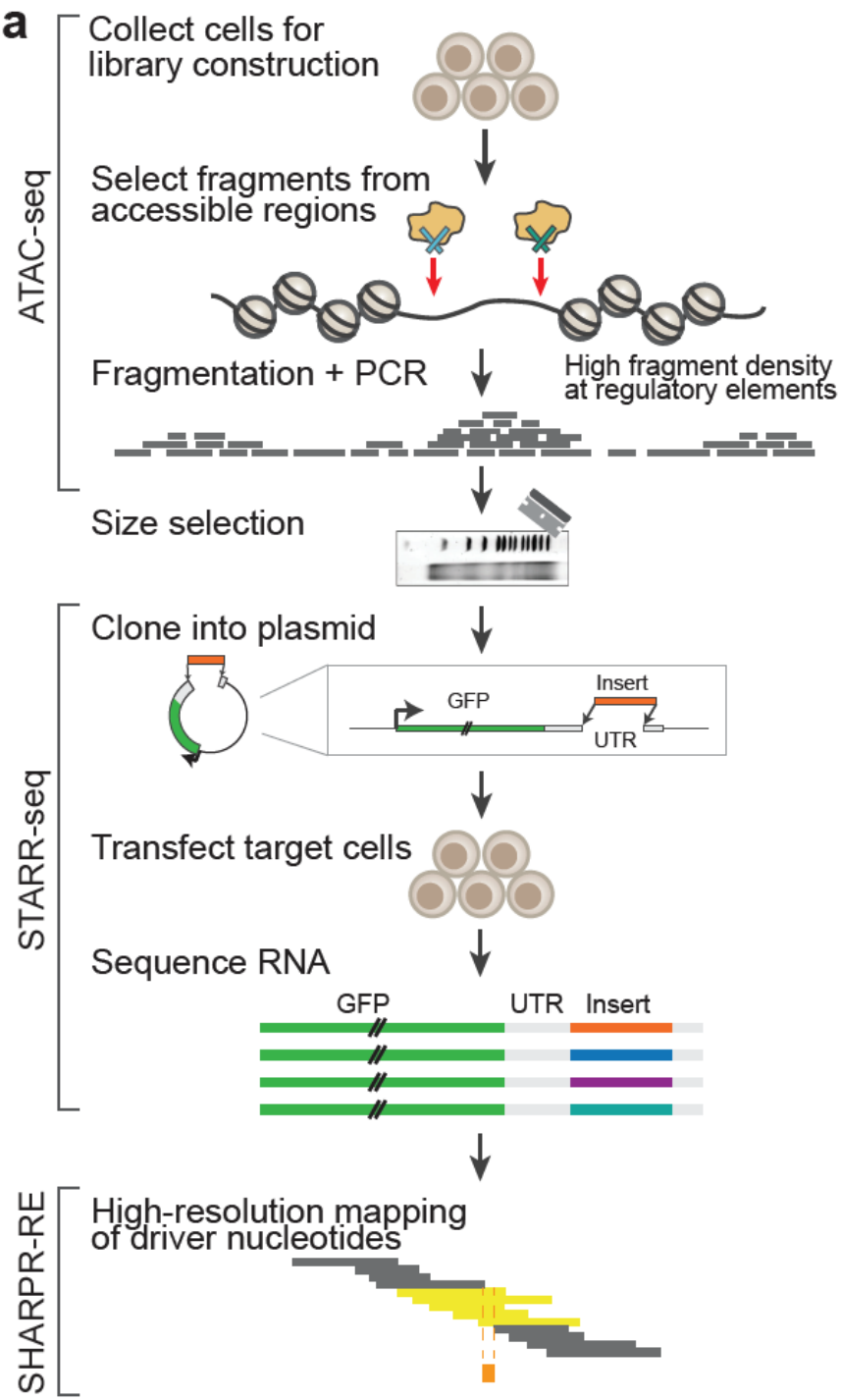






54000+ measurements (x2 cells, 2x repl)

HiDRA: High-Definition Reporter Assay



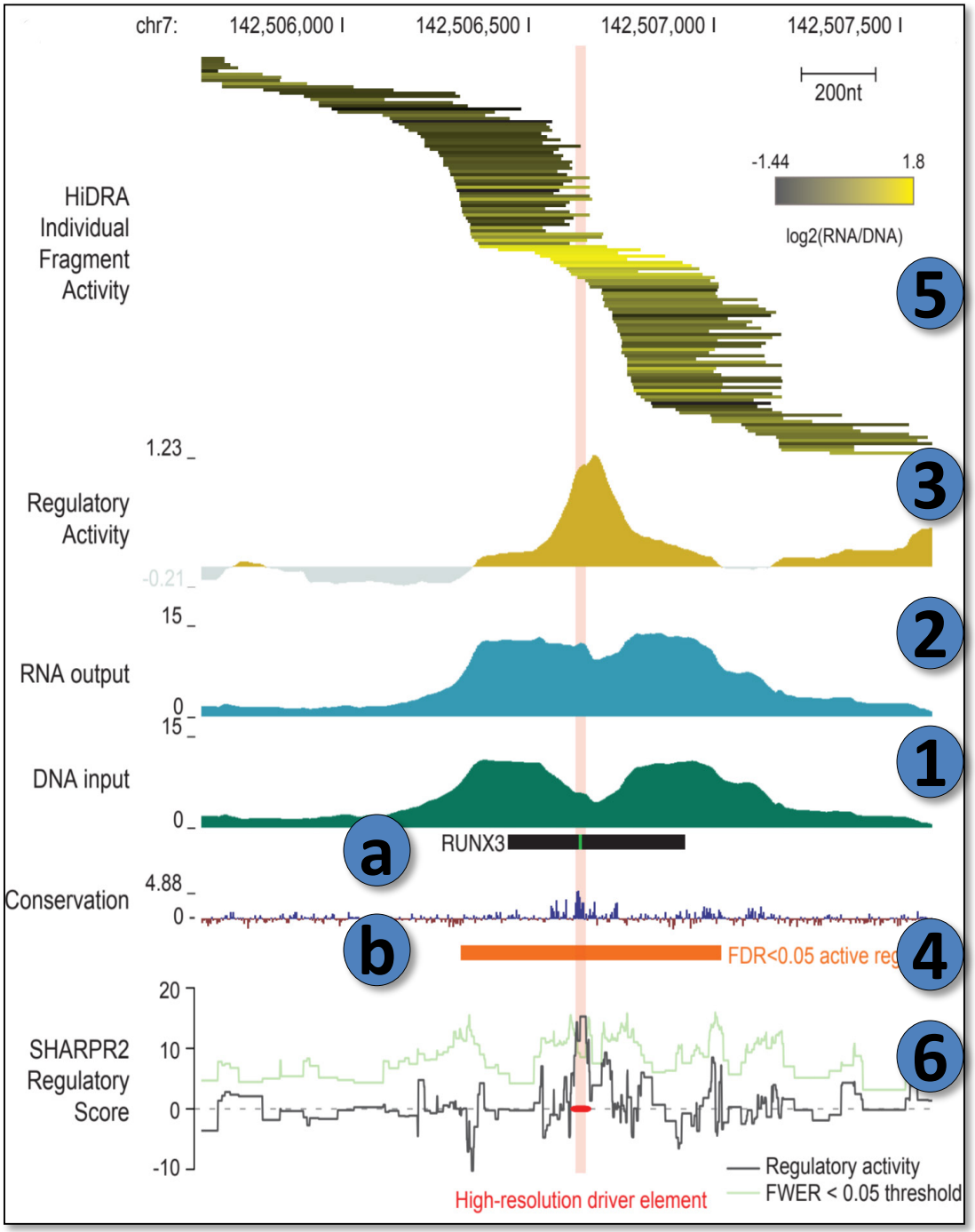
Key features:

- No synthesis → 7M fragments tested in 1 expt
- No synthesis, size-selection → Test long fragments
- Select accessible DNA regions → High sensitivity
- 3'UTR integration → self-transcribing → No barcode
- Densely-overlapping fragments → Region tiling
- Unbiased, random starts/ends → Sharpr dissection

Putting it all together:

- Testing 7M fragments in 1 experiment
- High sensitivity, high specificity, quantitative assay
- High-res inference pinpoints driver nucleotides

HiDRA data overview: DNA, RNA, Regulatory Activity, Sharpr2



1. Sequence DNA library
- Effectively a DNase/ATAC-Seq expt

2. Sequence RNA output
- How much expression does this drive

3. Take RNA/DNA ratio
- Measures regulatory activity

4. Pinpoint boundaries of active region
- FDR<0.05

5. Study activity of individual fragments
- Random start/end cuts (Transposase)

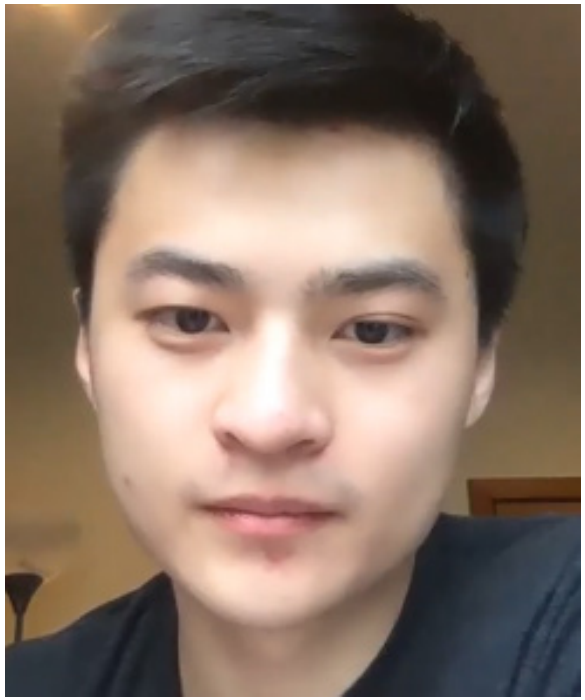
6. Infer high-resolution driver nucleotides
- Sharpr2 deconvolution algorithm
- Exploit diffs btw overlapping fragments

a. Compare with evolutionary conservation
- Capture evolutionarily-conserved nts

b. Compare with bound regulatory motifs
- Driver nucleotides are highly accurate

Guest lecture: Flynn Chen, Mark Gerstein Lab

Deep Learning for Reporter Expression Prediction



Flynn Chen
Yale University
Statistics and Data Science

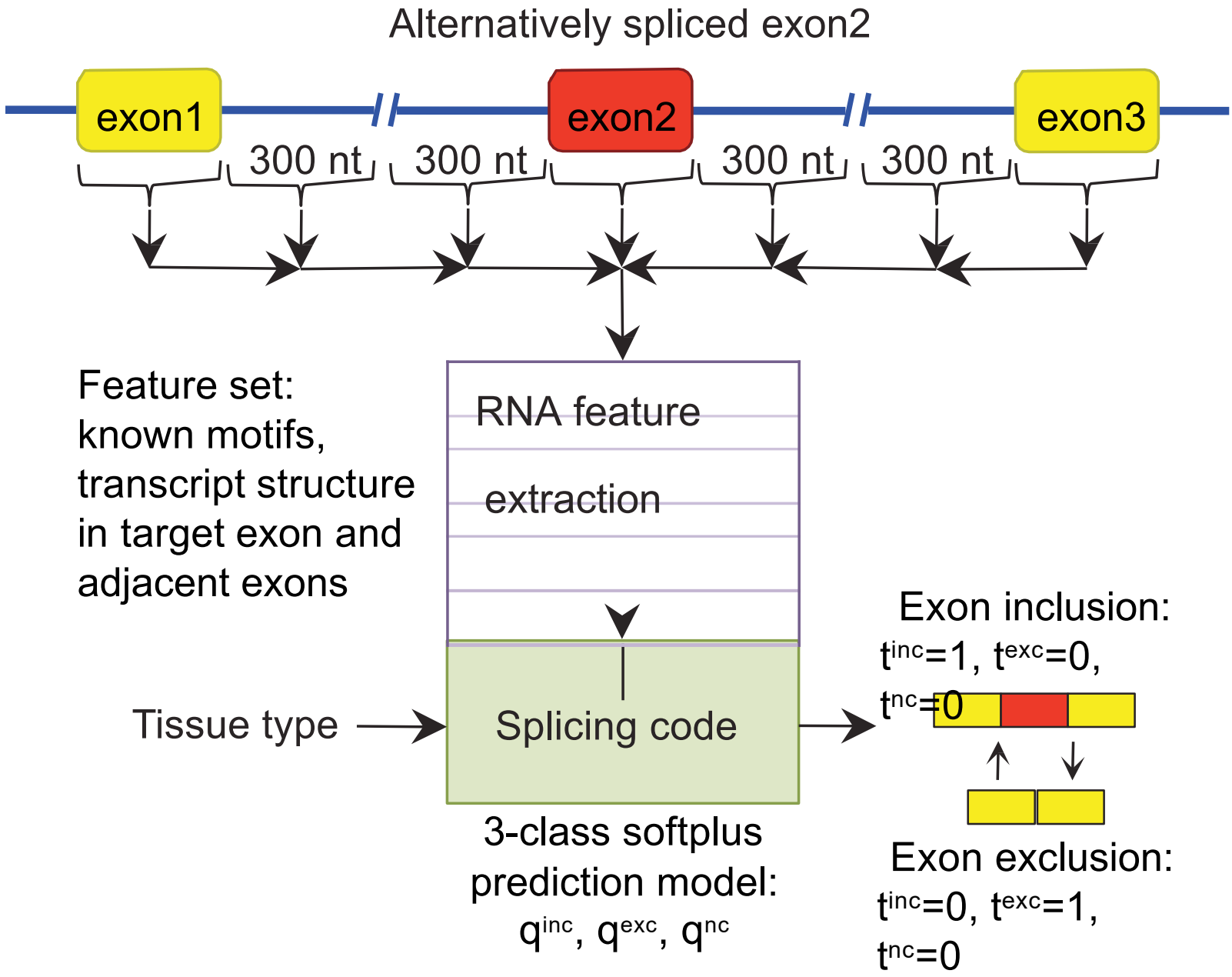
Prof. Mark Gerstein
Yale University
ENCODE, Data Science

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

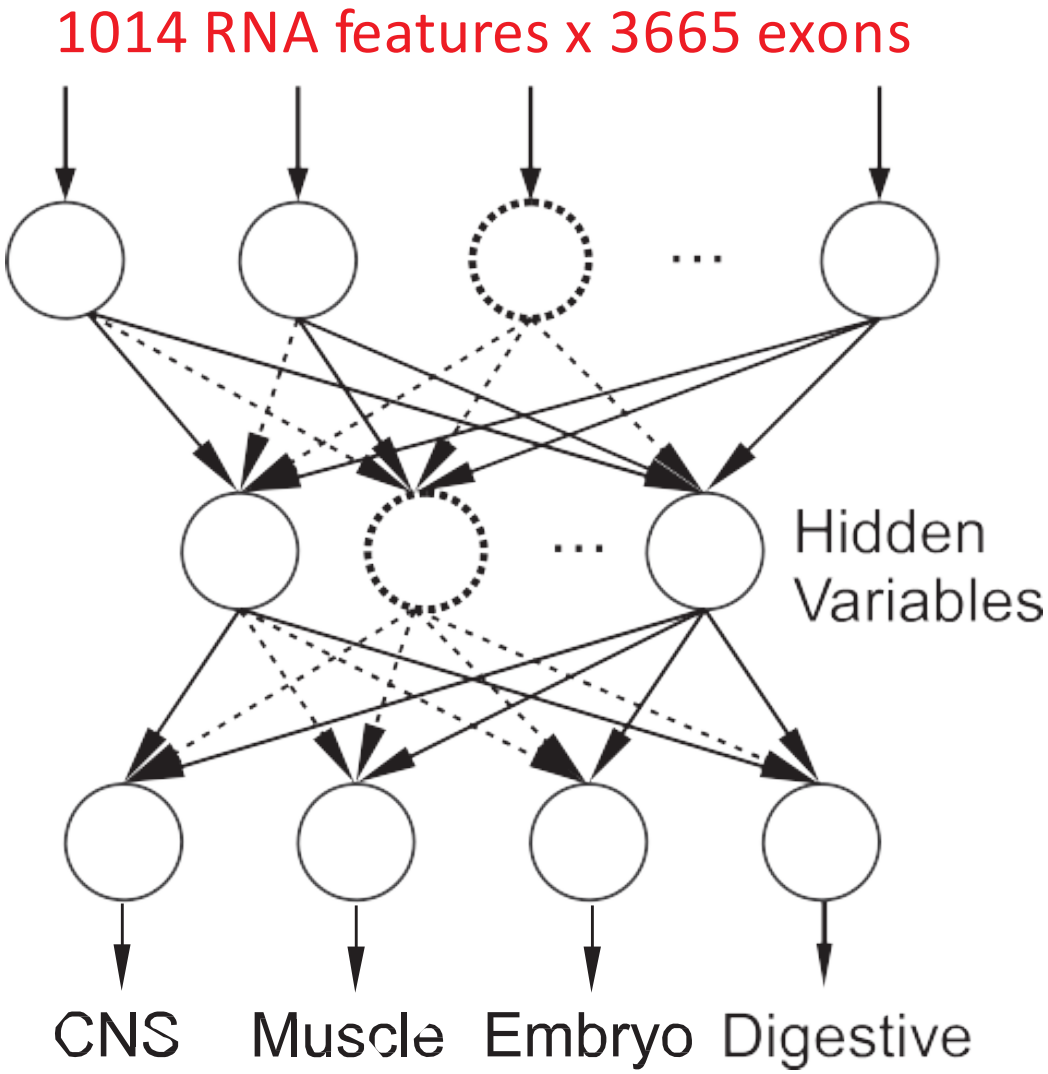
4. Predicting splicing from sequence

Deciphering tissue-specific splicing code



[Barash et al., 2010]

Bayesian neural network splicing code

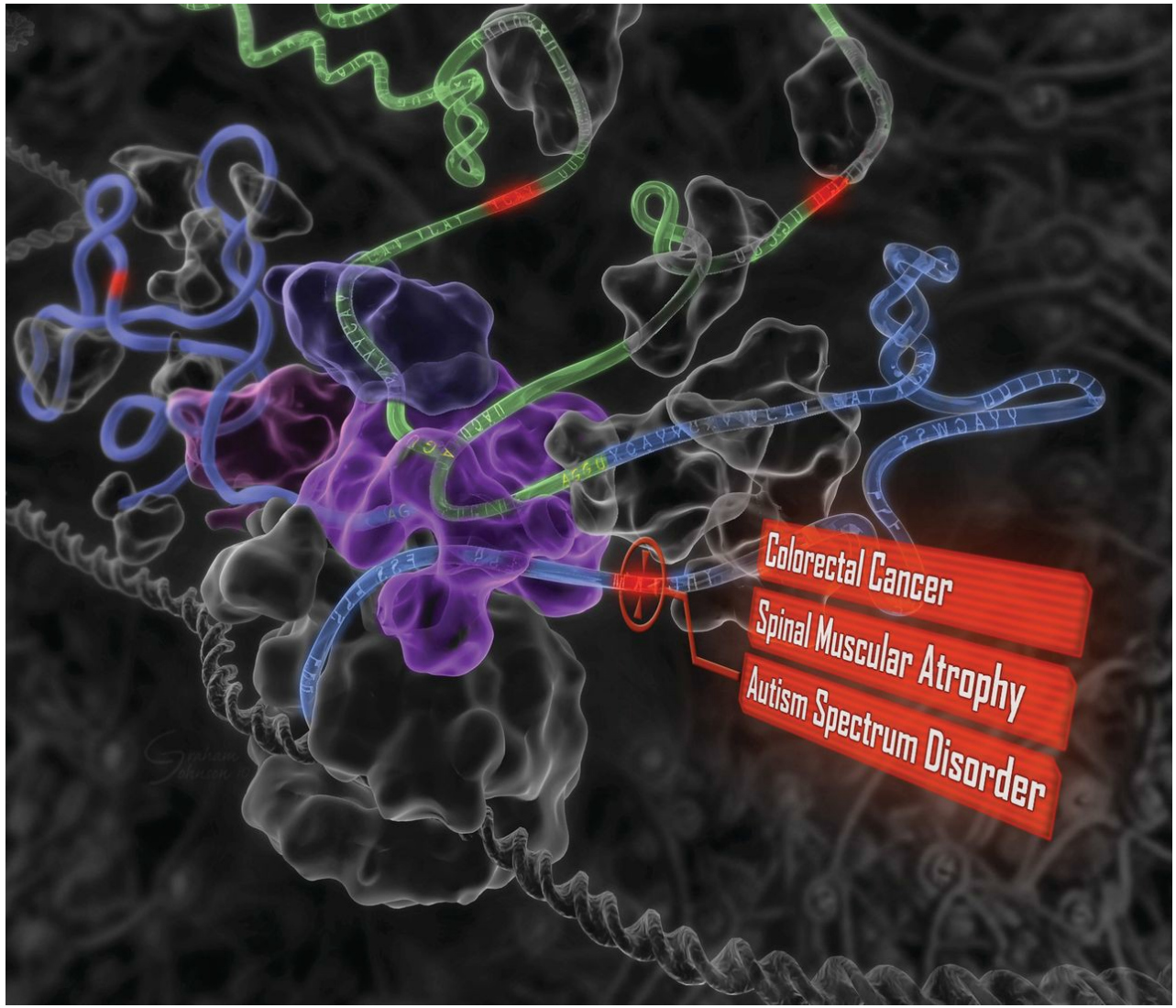


Bayesian neural network:

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

[Xiong et al., 2011]

Predicts diseasing causing mutations from splicing code

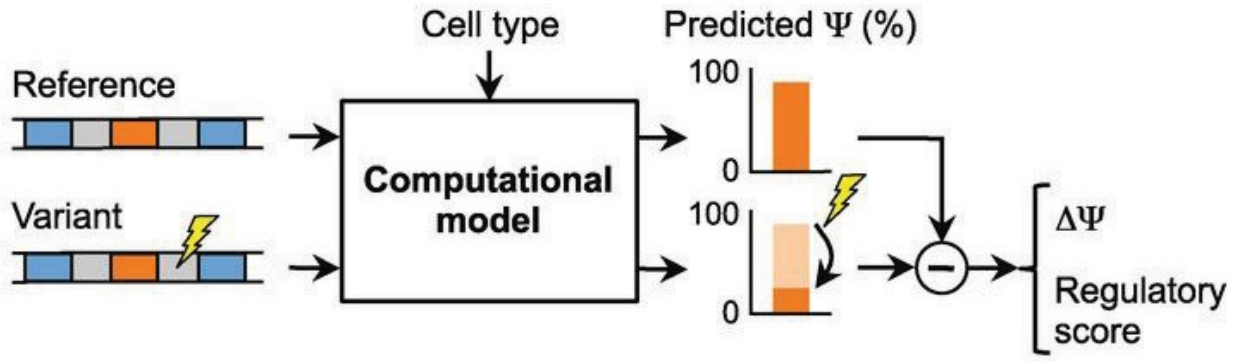


[Xiong et al., 2011]

Predicts diseasing causing mutations from splicing code

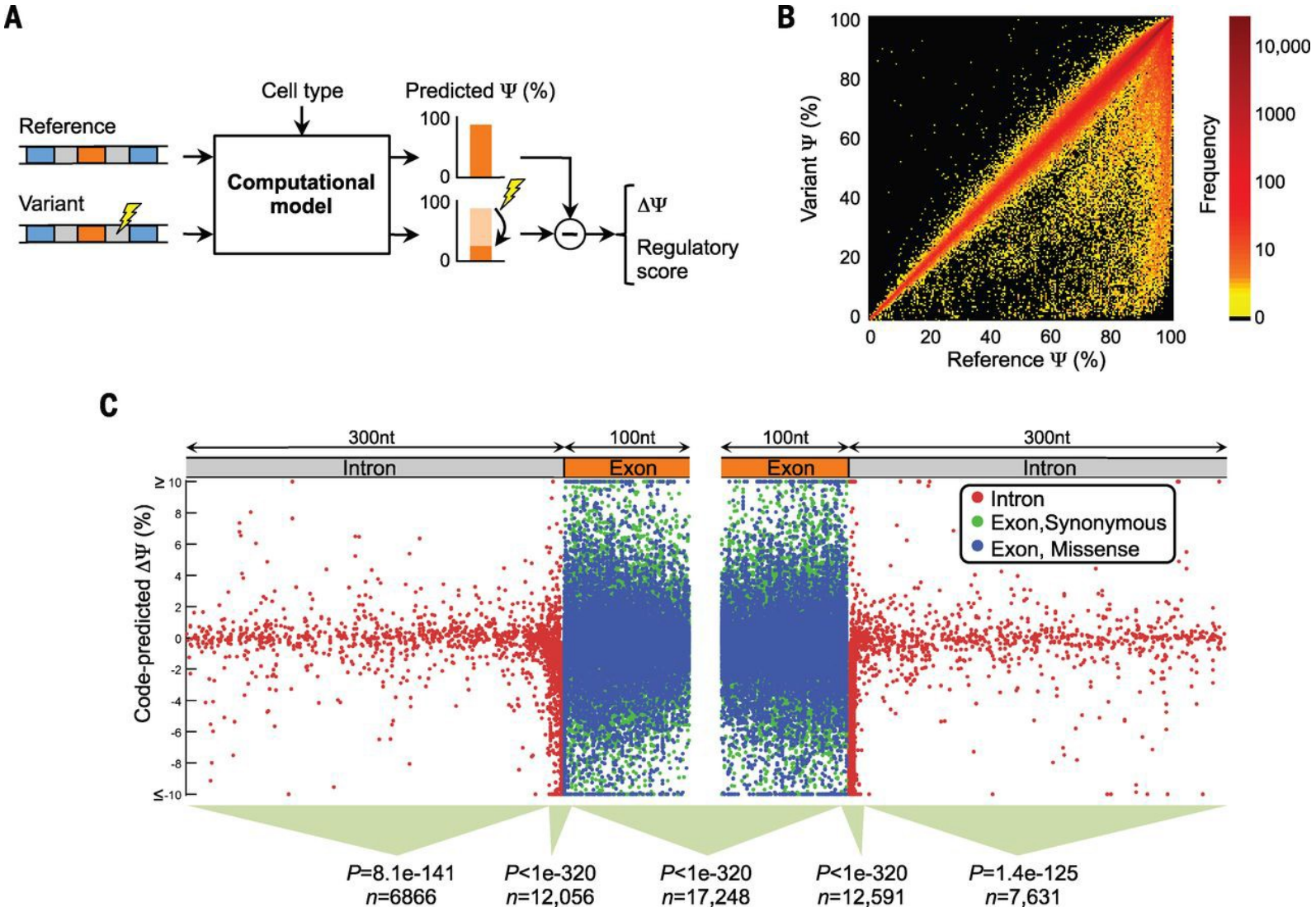
Scoring splicing changes due to SNP $\Delta\psi$:

- 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 ψ_{ref} and alternative ψ_{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

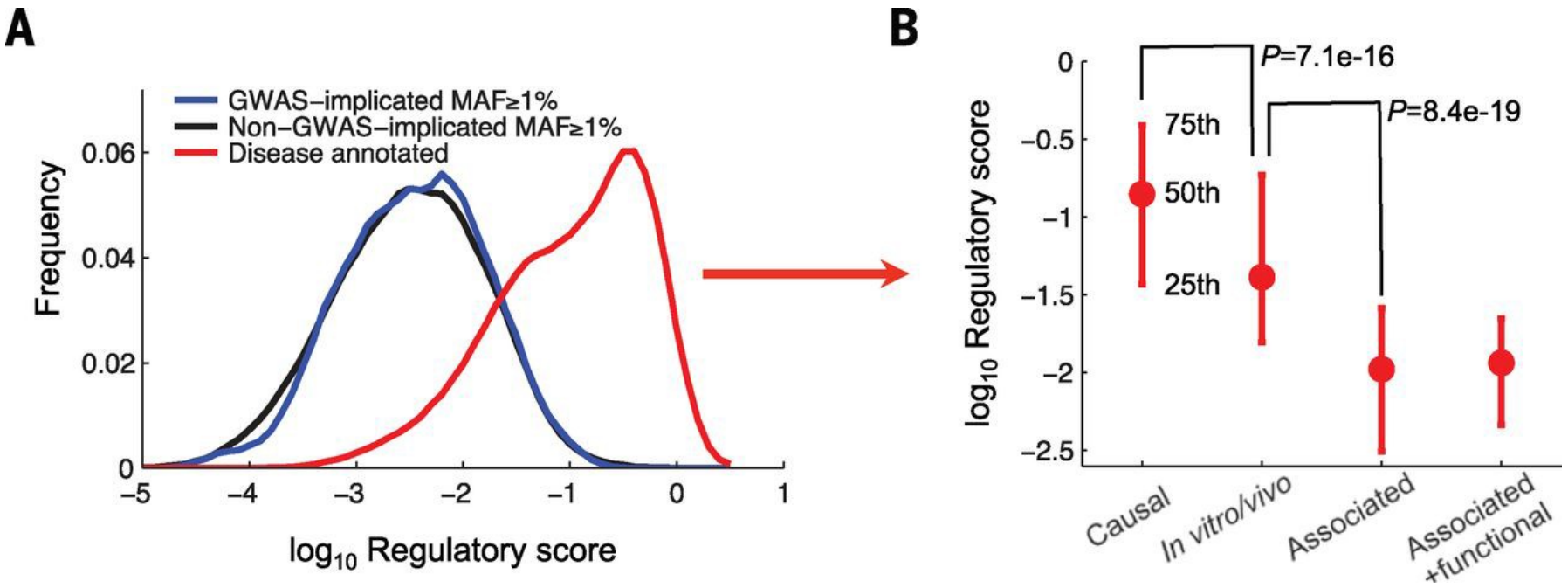


[Xiong et al., 2011]

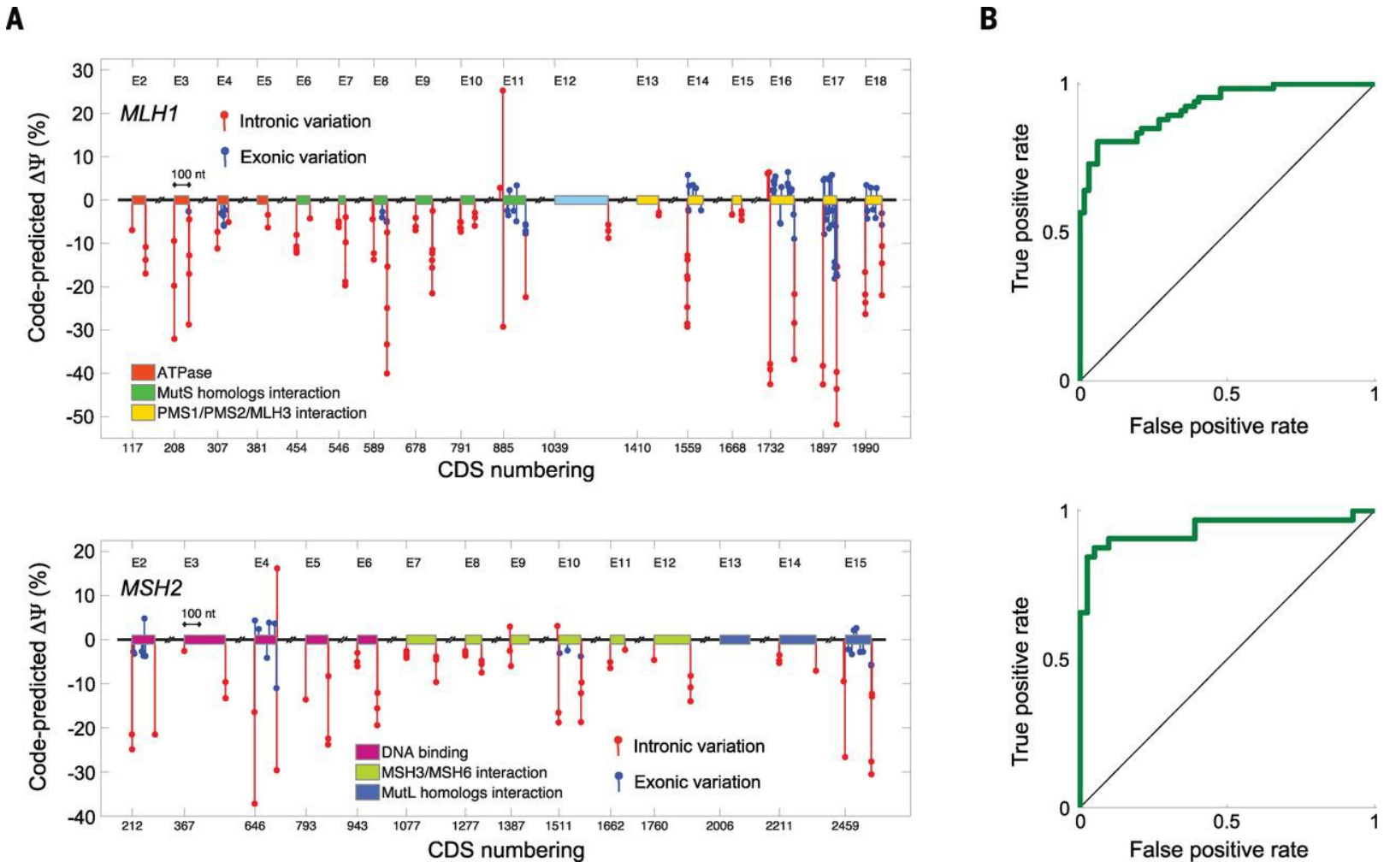
Predicted scores are indicative of disease causing mutations



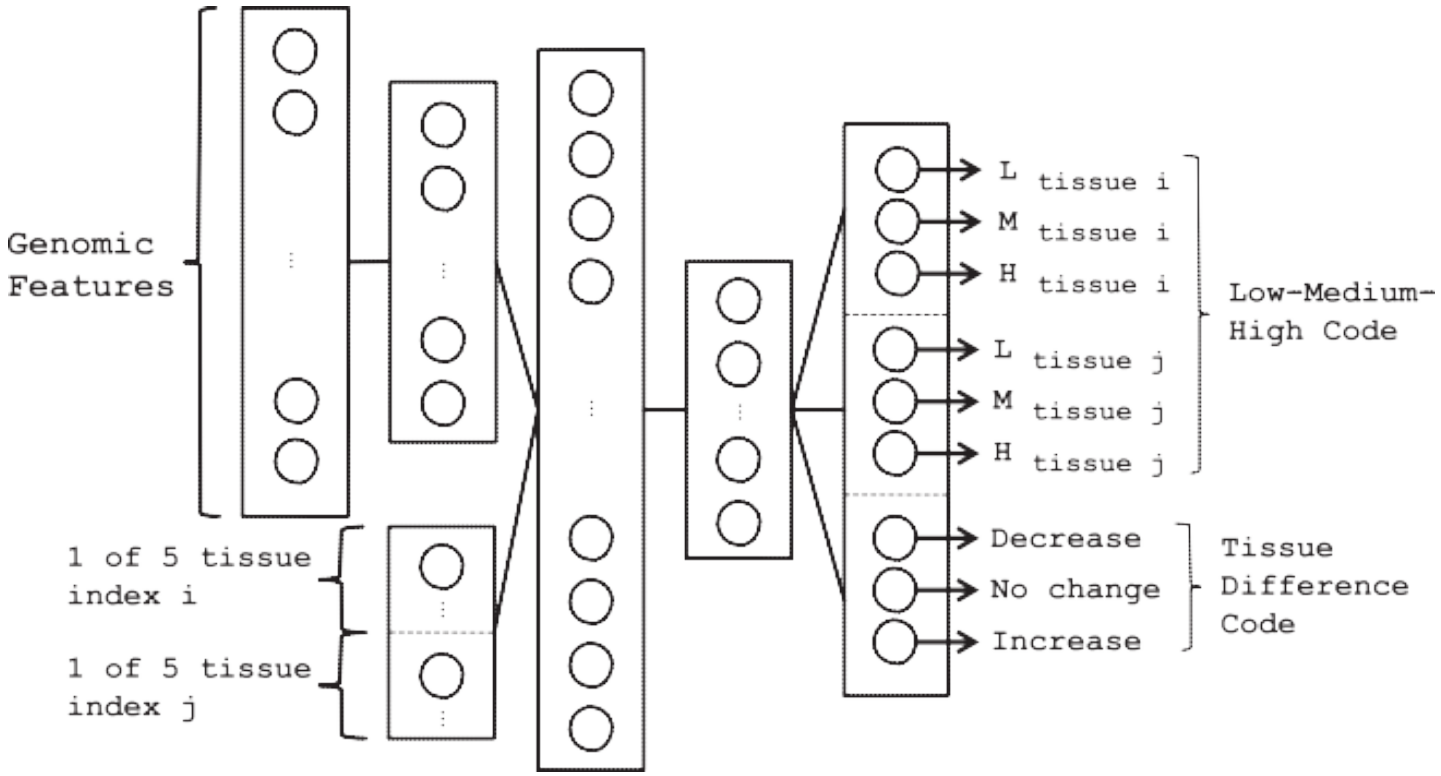
Predicted scores are indicative of disease causing mutations



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



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.

[Leung et al., 2014]

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 inputs + 13 outputs) × 10 hidden units = 13000 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 be completely reflect the underlying splicing mechanism
- Interpretation of the importance of features is not trivial

Guest lecture: Kyle Farh, Illumina
Deep Learning for Splicing Prediction

Dr. Kyle Kai-How Farh
Illumina
Harvard/MIT/Broad alum

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

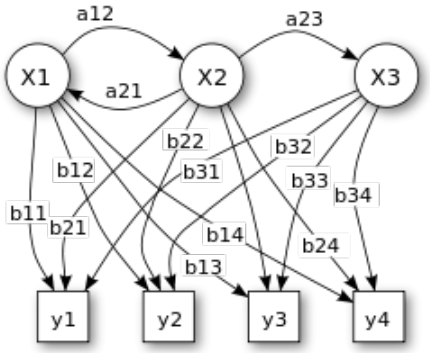
Where before What, a Weakly Supervised Framework (DECODE) for Precise Enhancer Localization

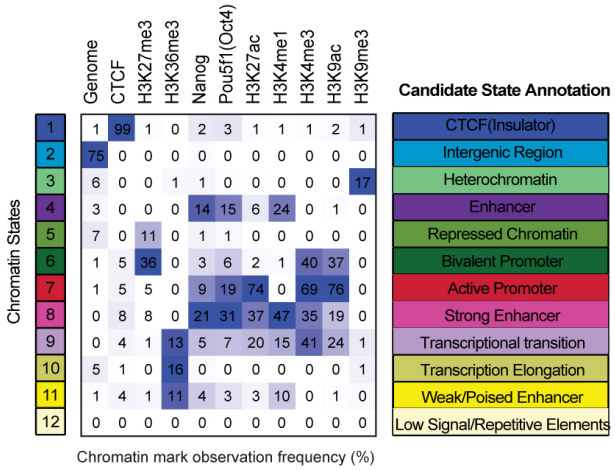
-- Zhanlin (Flynn) Chen

https://www.biorxiv.org/content/10.1101/2021.01.27.428477v2.full.pdf

Enhancer Discovery

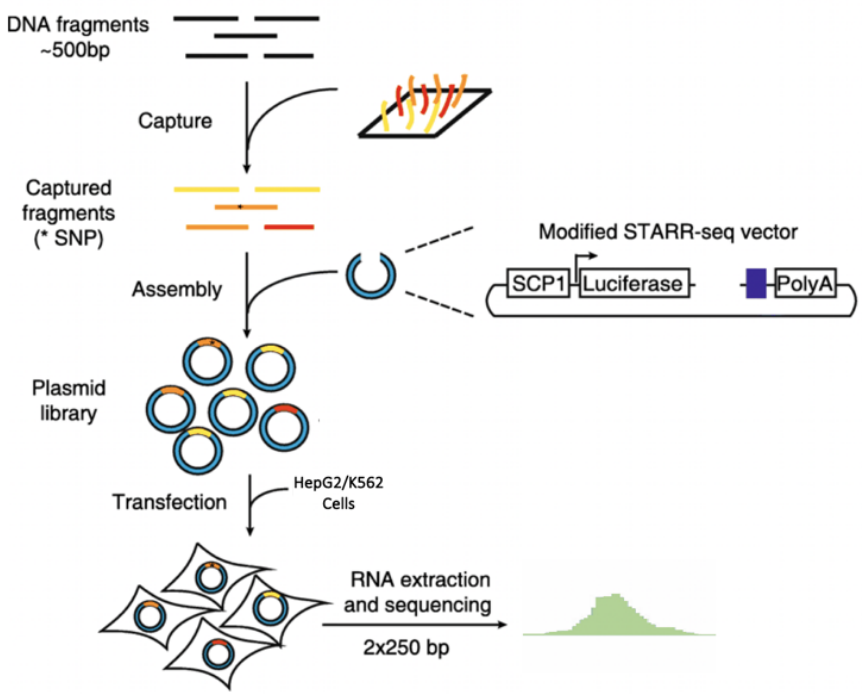
- Enhancers are a type of regulatory element that increases the transcription of a particular gene
- Mapping out cell-type specific regulatory landscape allows us to find genetic drivers for various diseases
- Earliest methods for enhancer discovery like ChromHMM focused on **unsupervised** approaches





STARR-seq Experiments

- Massively parallel reporter assay
- Identifies transcriptional enhancers directly based on their activity
  - Fragments of the genome is transfected into target cells in front of a luciferase gene
  - The ability to increase transcription of that fragment is quantified by measuring the relative expression of the luciferase gene
- Low transfection efficiency, low resolution, **evaluate fragments out of epigenetic context**
- Provide a basis for **supervised** approaches



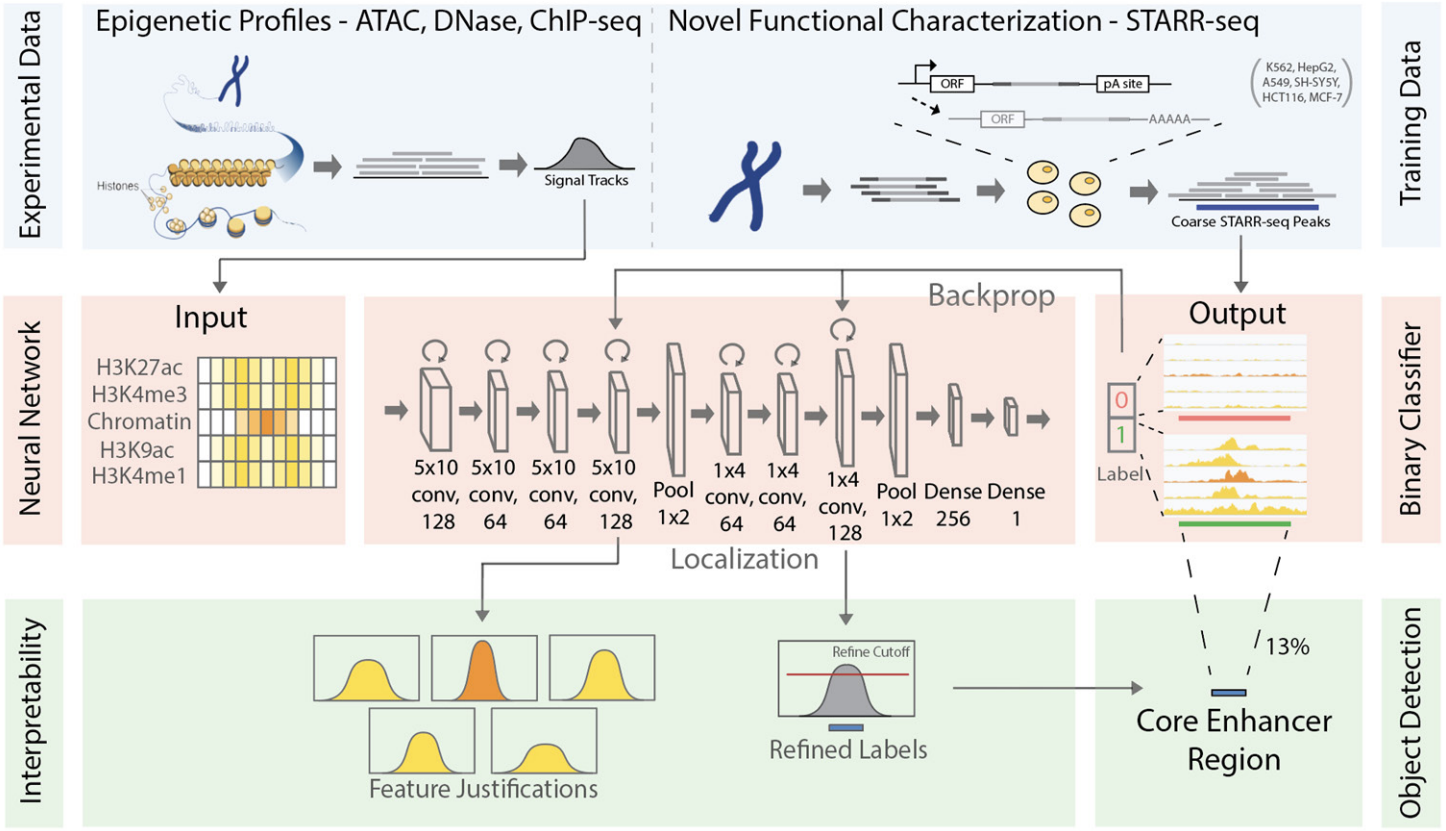
Current ENCODE Dataset (hg38)

Cell Type	STARR-seq	ATAC-seq	DNase-seq	H3K27ac ChIP-seq	H3K4me3 ChIP-seq	H3K4me1 ChIP-seq	H3K9ac ChIP-seq
K562	✓	✓	✓	✓	✓	✓	✓
HepG2	✓	✓	✓	✓	✓	✓	✓
A549	✓		✓	✓	✓	✓	✓
HCT116	✓		✓	✓	✓	✓	✓
MCF-7	✓		✓	✓	✓	✓	✓

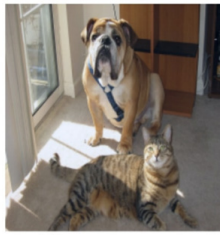
Our central hypothesis: The interactions between open chromatin and histone marks provide a platform for TF binding and enhancer activity

Workflow/Architecture

- Given only epigenetic features and coarse training labels, could we produce precise localization of enhancers?
- Operationalized enhancer discovery into an **object detection** task
- First, classify 4kb sliding windows given genomic features
- Second, use Grad-CAM for feature justification and localization



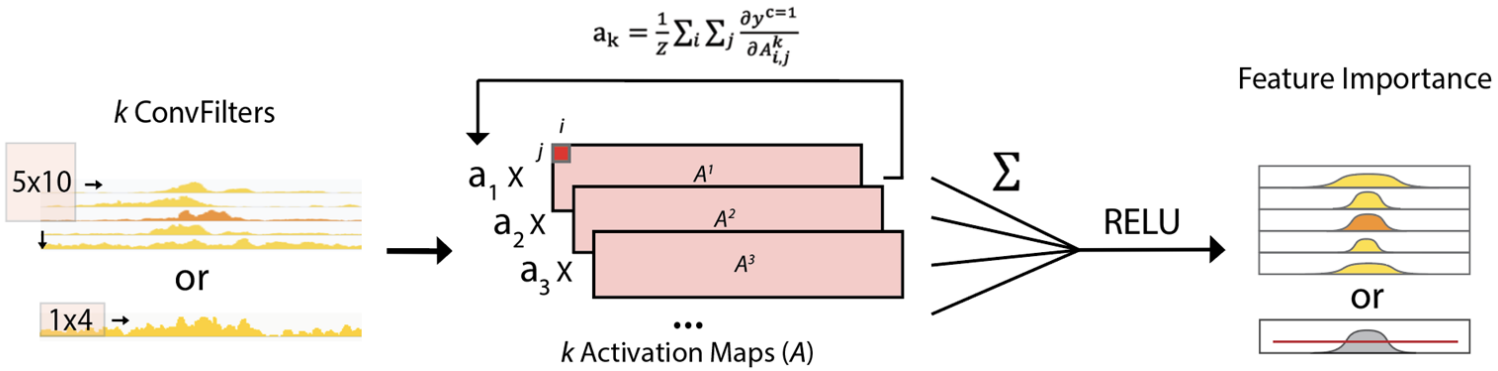
Weakly Supervised Learning (Grad-CAM)



(a) Original Image

(i) Grad-CAM 'Dog'

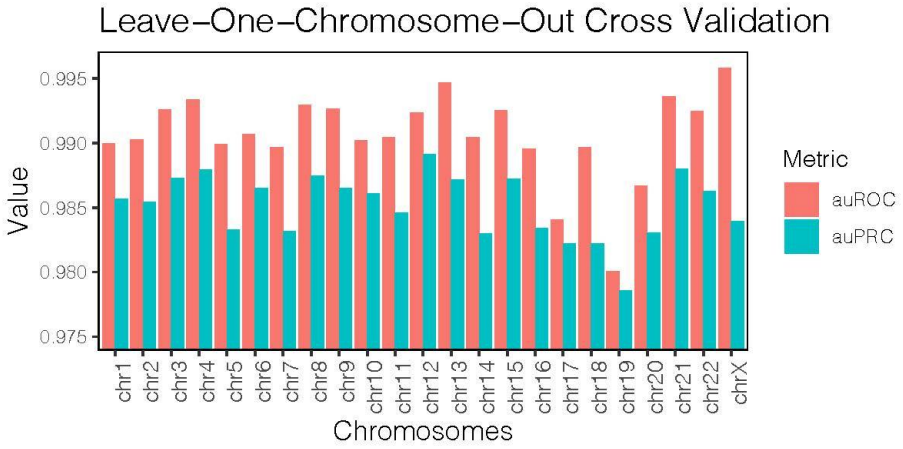
(c) Grad-CAM 'Cat'



- Applying convolutional filters on a given input produces activation maps that highlights particular features, which is a subset of the input.
- A linear combination of activation maps (weighed on the sum of all activation in the maps) produces a heatmap localization of the object.
- **Weakly supervised**: label only indicate "existence" for classification. No locations were provided in our model, yet they could be inferred.

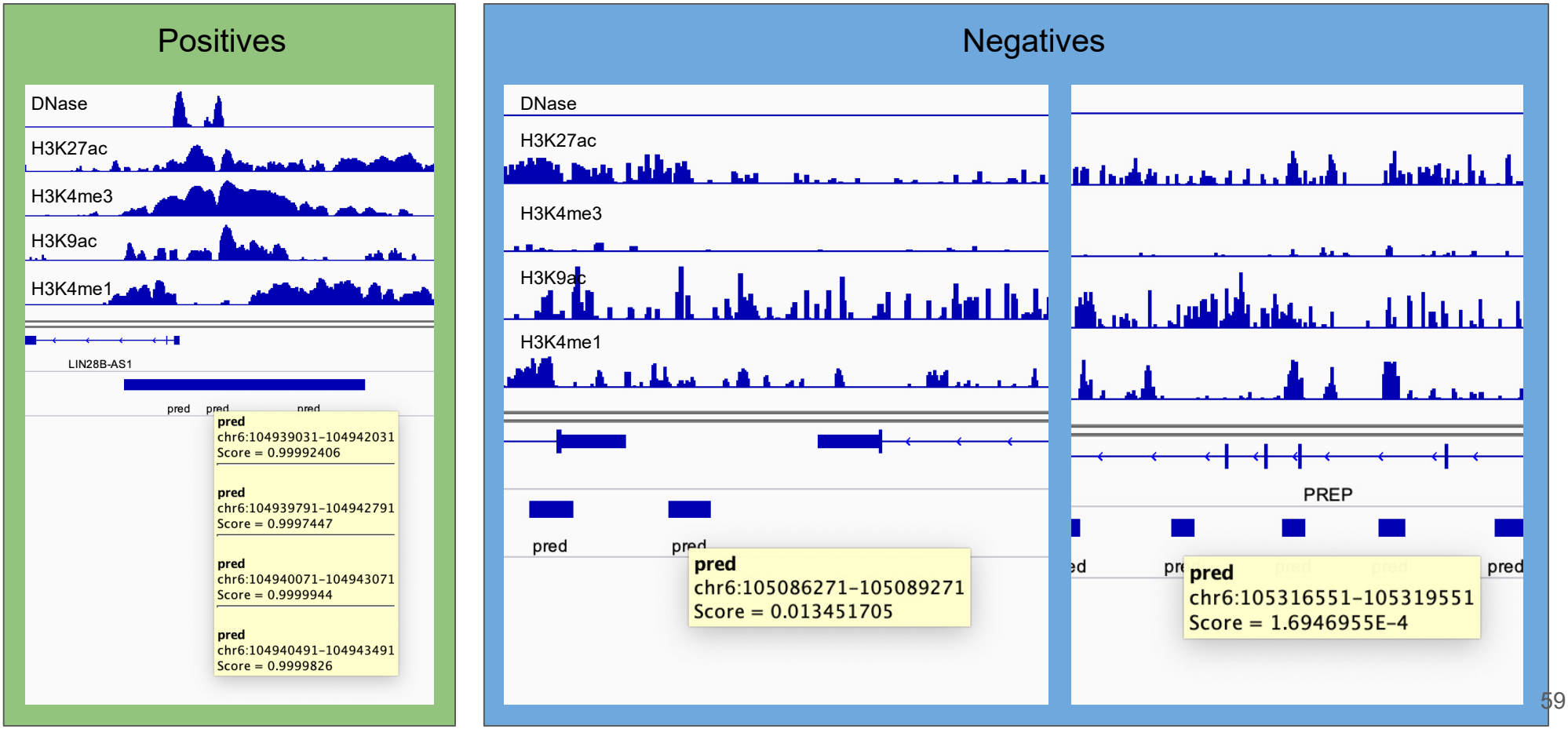
Cell-Line/Chromosome Cross Validation

Chromatin Accessibility	Cell Type	Validation Accuracy	Validation auROC	Validation auPRC
ATAC-seq	K562	0.9885	0.9972	0.9704
	HepG2	0.9908	0.9960	0.9536
DNase-seq	K562	0.9849	0.9984	0.9975
	HepG2	0.9865	0.9978	0.9972
	A549	0.9818	0.9984	0.9978
	HCT116	0.9918	0.9989	0.9981
	MCF-7	0.9897	0.9983	0.9978

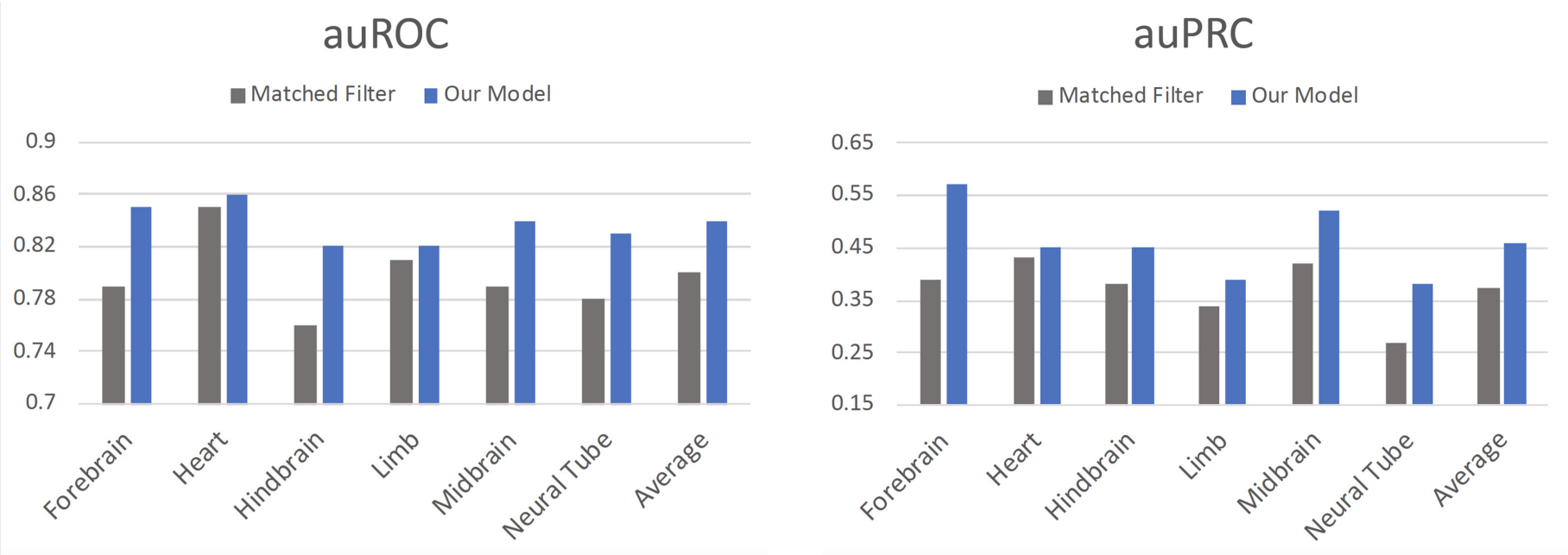


- Data from cell-lines/chromosomes were set aside for validation
- Cross cell-line validation indicate that our model can **generalize predictions to new cell-lines**
- Cross chromosome validation indicate that our model can **generalize predictions to new genomic loci**

Example Predictions

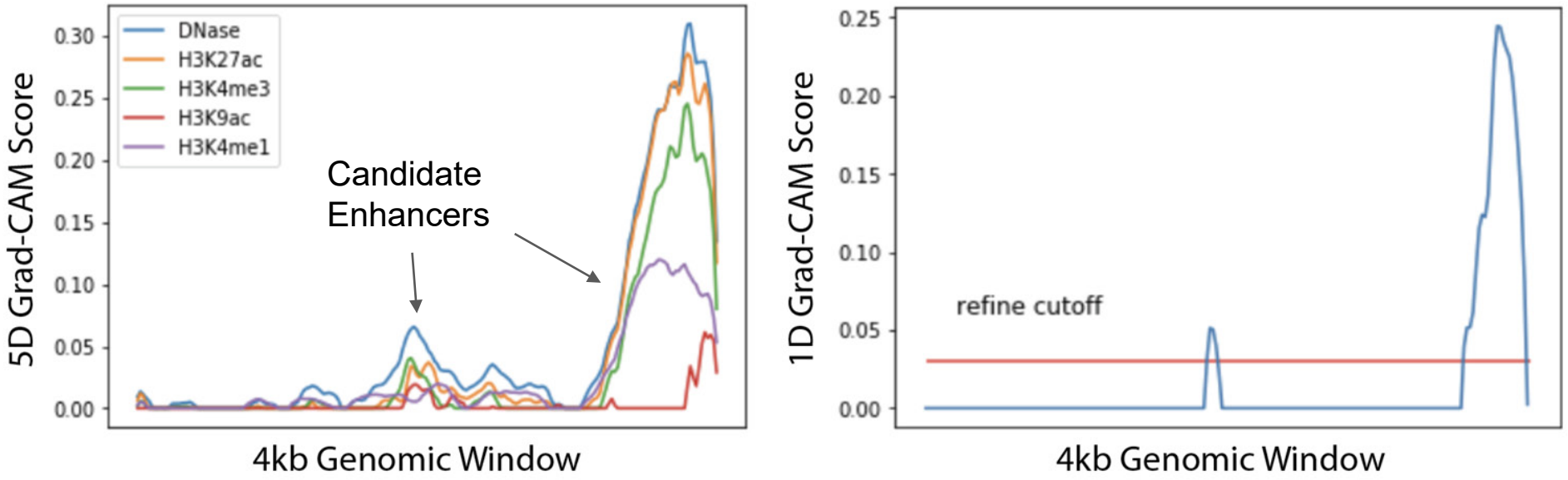


Transgenic Mouse Validation (Our model vs. SOTA)



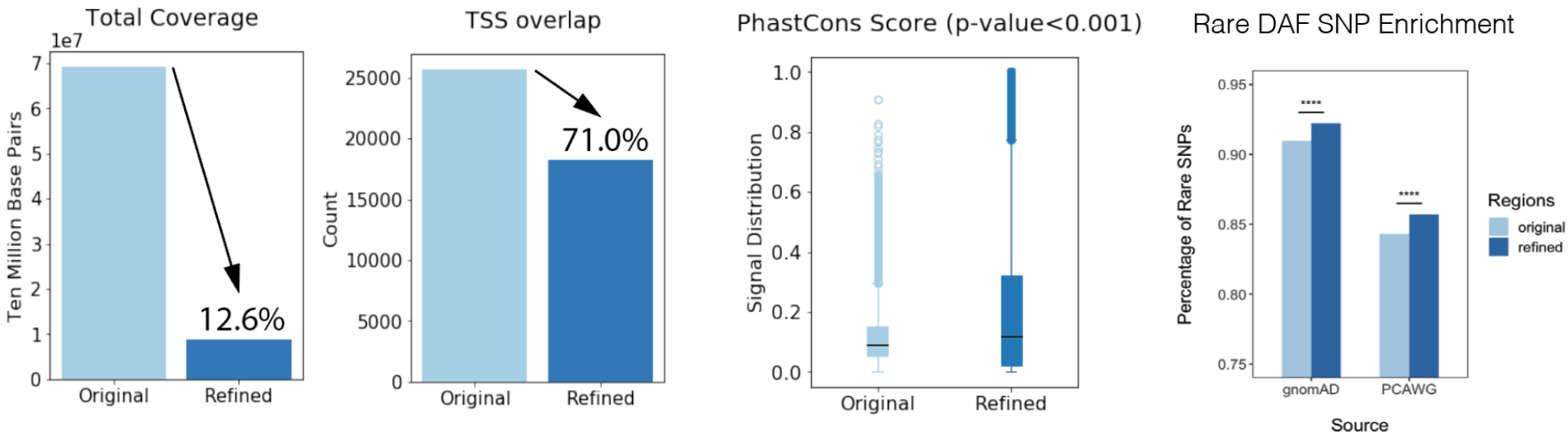
- **Matched Filter** utilized shape-matching filters for feature extraction and linear SVMs for classification
- Comparison on ENCODE Enhancer Challenge Dataset (VISTA mouse enhancer database https://enhancer.lbl.gov/)
- Outperformed Matched filter in every mm10 tissue type, some with 15-20% margin

Neural Progenitor Cell (NPC) Case Study



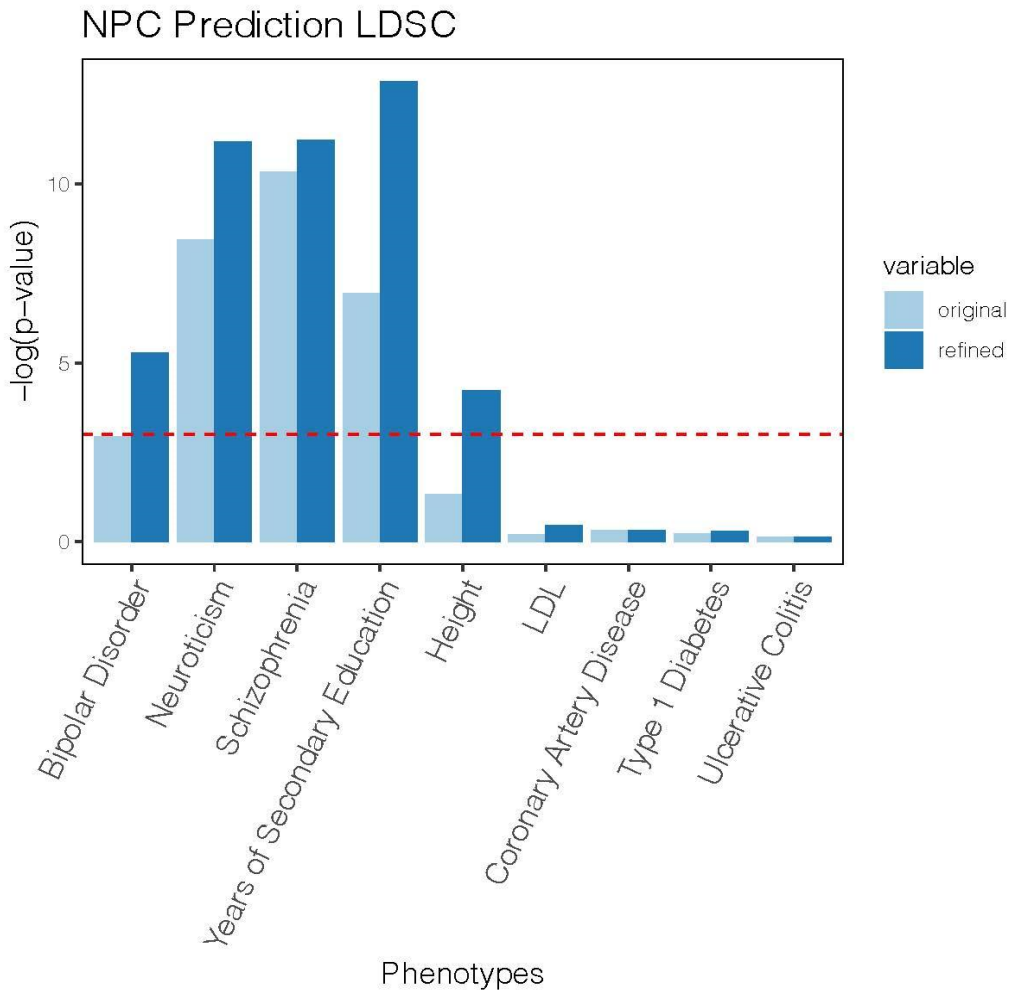
- **Feature-wise score:** importance score for each feature (left)
  - DNase, H3K27ac, and H3K4me3 were emphasized
  - Interaction of low-level features lead to high-level features
- **Position-wise score:** importance score for each location/loci (right)
  - Position-wise scores capture candidate enhancers within a subset of the 4kb input

Prediction Statistics



- Our refined predictions cover less area (12.6% of the original 4kb positive predictions), but is enriched for transcriptional start sites, indicating a strong transcriptional impact from our refinement.
- 100-way phylogenetic PhastCons shows enriched inter-specie conservation.
- Rare Derived-Allele-Frequency (DAF) SNPs enrichment indicate intra-species conservation (through negative selection).

Disease Causal-variant Mapping



- Linkage Disequilibrium Score (LDSC) determines whether the heritability of a phenotype is enriched through GWAS summary statistics.
- Our original and refined NPC enhancers are enriched mostly only for neurodevelopmental and psychiatric phenotypes.
- Our NPC refined enhancers exhibit higher LDSC enrichment compared to original enhancers in relevant GWAS phenotypes
- Increase statistical power could be attributed to our **compact** annotations

Thank you

Special Thanks to:
Mark Gerstein,
Jing Zhang,
Jason Liu,
And other members of the Gerstein Lab

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

Deep Learning in Gene Expression Analysis

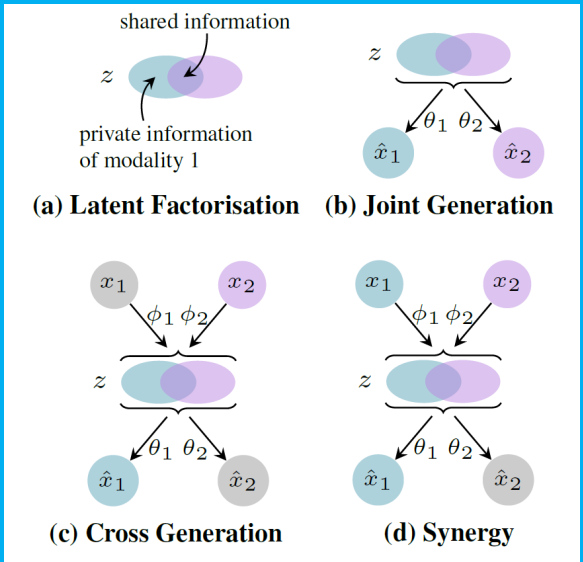
Xiaohui Xie
University of California, Irvine
xhx@uci.edu

Deep learning methods developed by Xie Lab

1. **DANQ**: deep neural network for quantifying the function of DNA sequences
2. **FactorNet**: a deep learning framework for predicting cell-type specific transcription factor binding
3. **scFAN**: predicting transcription factor binding in single cells
4. **uFold**: fact and accurate RNA secondary structure with deep learning
5. **D-GEX:** Gene Expression Prediction from subsets of genes
6. **SAILER:** autoencoder representation of expression and chromatin
7. **MVAE:** multi-modal representations with variational auto-encoders

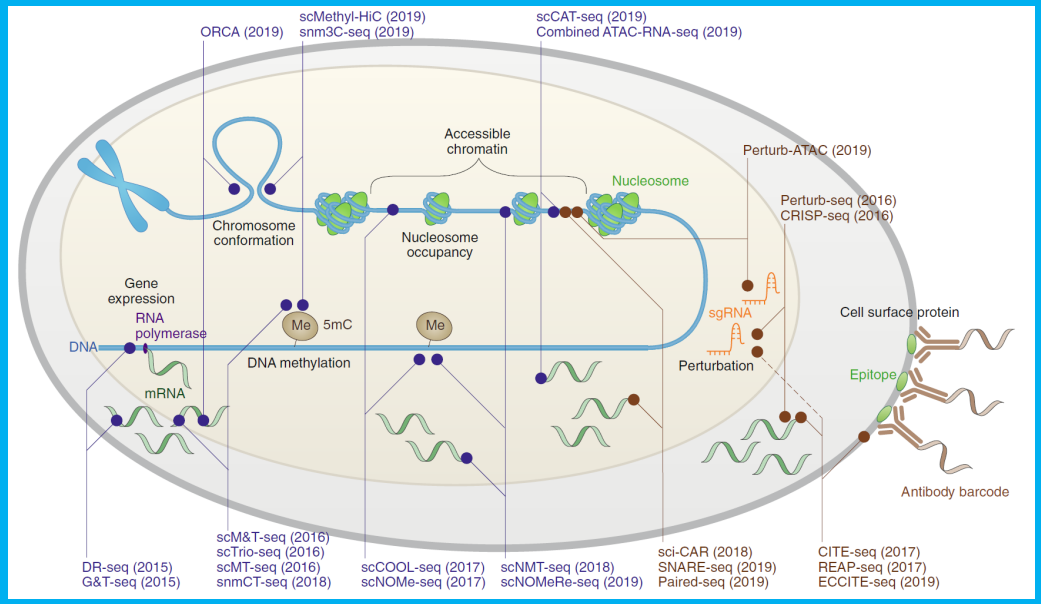
Multimodal Deep Learning for Single Cell Multimodal Omics

- Multimodal deep generative model



Shi, Yuge, et al. *NIPS* 2019

- Multimodal single cell omics methods



Zhu, Chenxu, Sebastian Preissl, and Bing Ren. *Nature methods* 2020

- Shi, Yuge, et al. "Variational mixture-of-experts autoencoders for multi-modal deep generative models." *Advances in Neural Information Processing Systems*. 2019.
- Zhu, Chenxu, Sebastian Preissl, and Bing Ren. "Single-cell multimodal omics: the power of many." *Nature methods* 17.1 (2020): 11-14.

D-GEX

Gene expression inference with deep learning

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** ▾

Connectivity Map (C-Map) Project

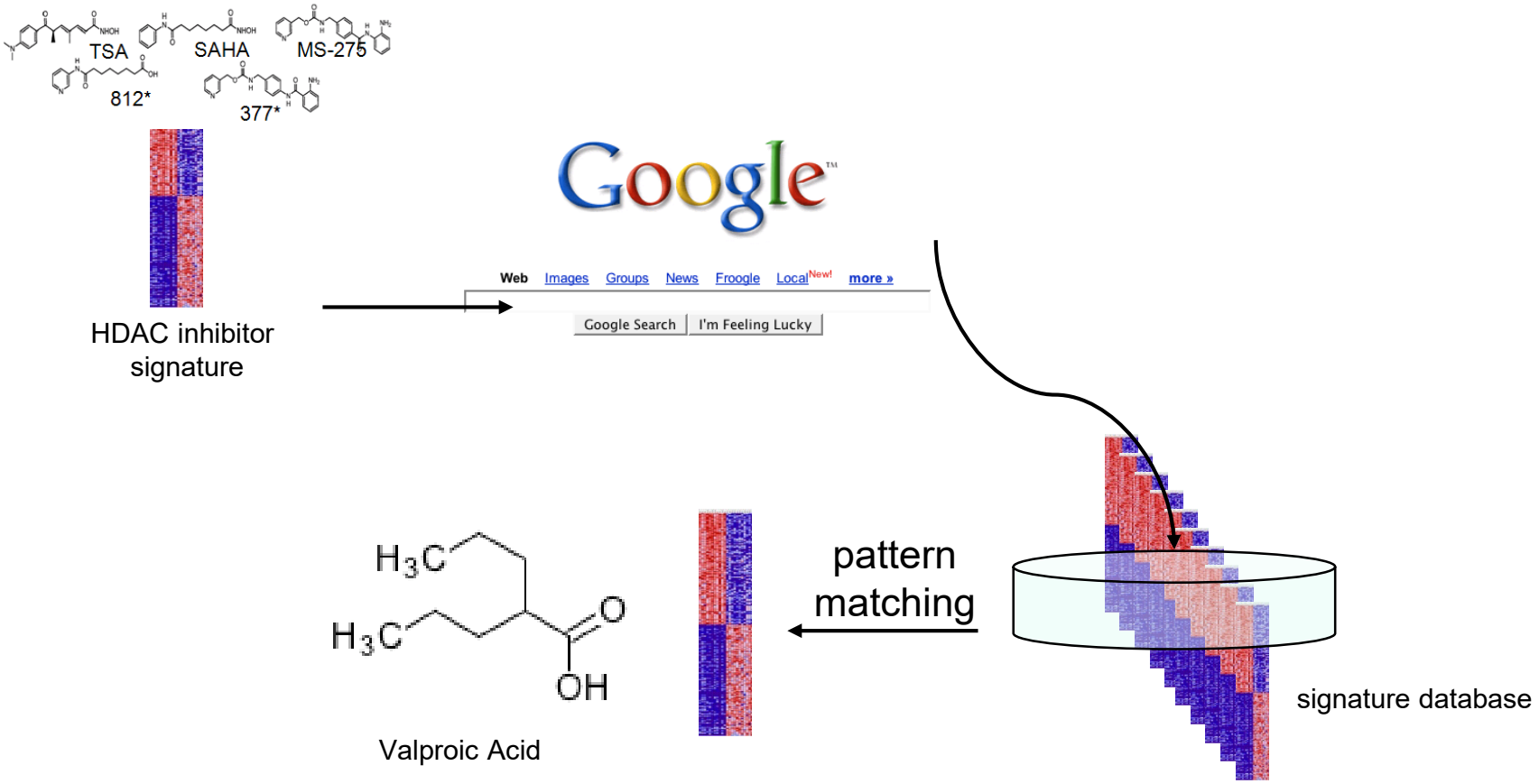
Aravind Subramanian, Justin Lamb & Todd Golub (Broad Institute)

Lamb *et al*, Science 2006

Comprehensive characterizing of cellular states: genome-wide mRNA expression profiles

The Connectivity Map (also known as cmap) is a collection of genome-wide transcriptional expression data from cultured human cells treated with bioactive small molecules and simple pattern-matching algorithms that together enable the discovery of functional connections between drugs, genes and diseases through the transitory feature o common gene-expression changes.

the c-map search engine



Histone deacetylase inhibitors (HDIs) have a long history of use in psychiatry and neurology as mood stabilizers and anti-epileptics, for example, **valproic acid**. In more recent times, HDIs are being studied as a mitigator or treatment for neurodegenerative diseases.[19][2]

connectivity map

the promise

- small molecule gene-expression profiles reveal connections b/w drugs⇔diseases⇔genes

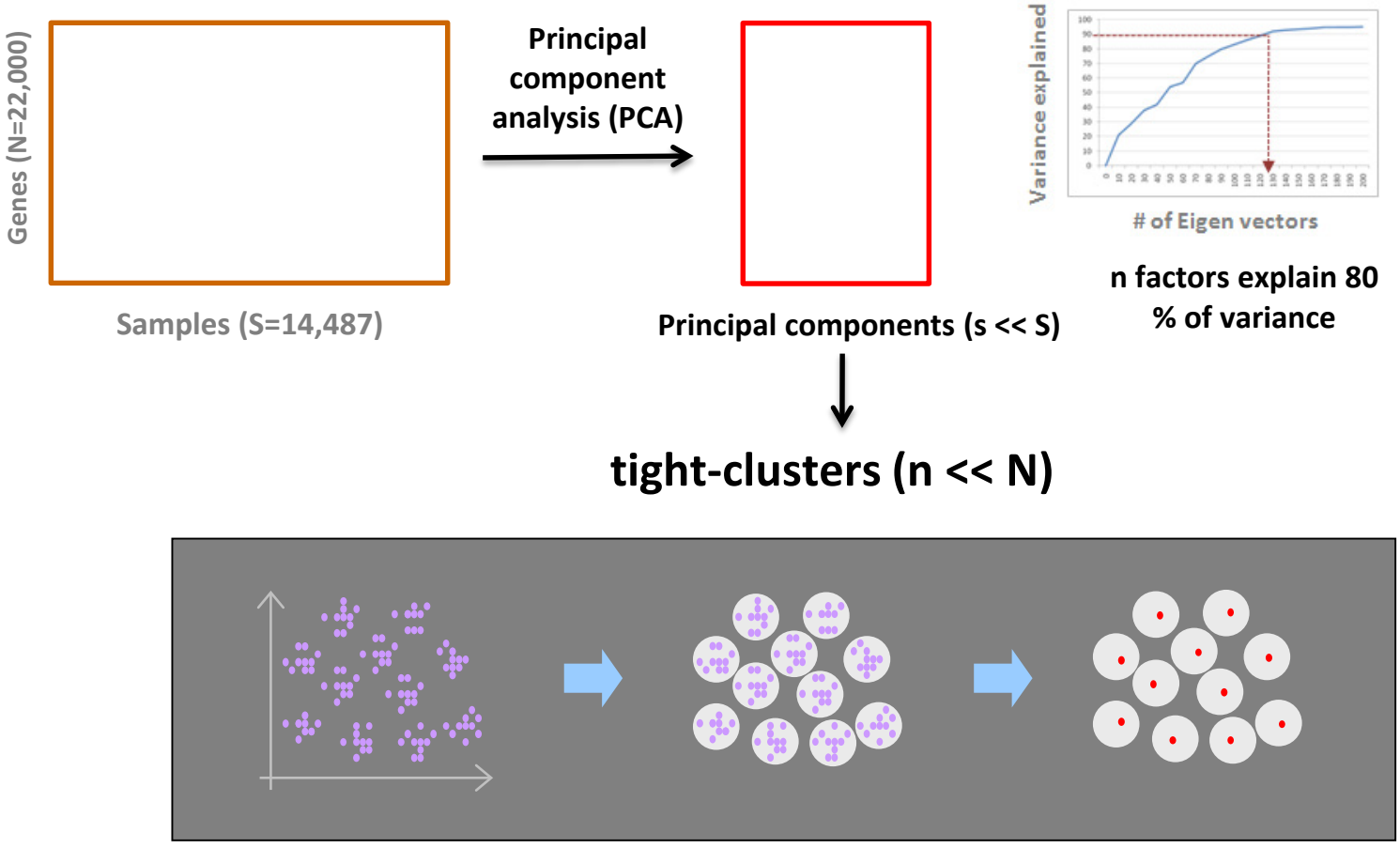
the problem

- whole-genome profiles are expensive!
- Affymetrix: ~$400 / drug in one cell line
- scaling to large chemical libraries, genotypes, cell lines *etc*, prohibitively expensive

the 1000-gene solution

- measure 1000-genes at high-throughput, low cost
- use whole-genome compendium datasets to infer the remaining genes

Dimension reduction

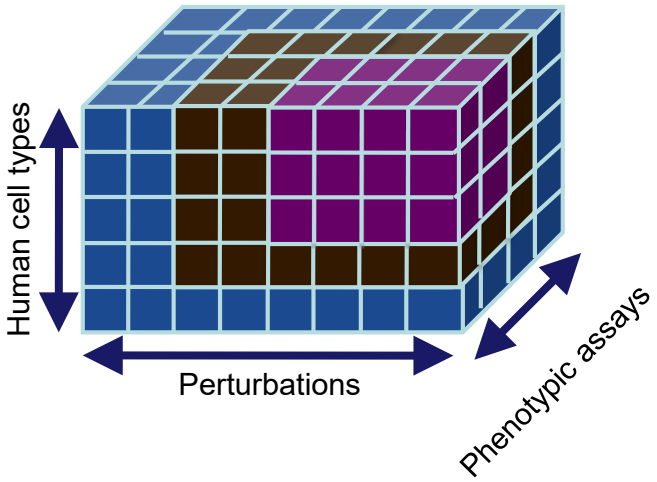


NIH LINCS Program

lincscloud

LINCS aims to inform a **network-based understanding of biological systems** in health and disease that can facilitate drug and biomarker development.

Measure 1000 'Landmark' transcripts on Luminex bead



Currently released L1000 data includes 1.3 million samples

Resource

Cell

A Next Generation Connectivity Map: L1000 Platform and the First 1,000,000 Profiles

http://lincsproject.org

NIH LINCS PROGRAM

Gene expression learning problem

Using the landmark genes to infer the entire transcriptome

Measured: 978 landmark genes $\vec{x}$

Unknown: remaining ~21K target genes $\vec{y}$

Need to learn the mapping from *x* to *y*: $\vec{y} = f(\vec{x})$

Training Data

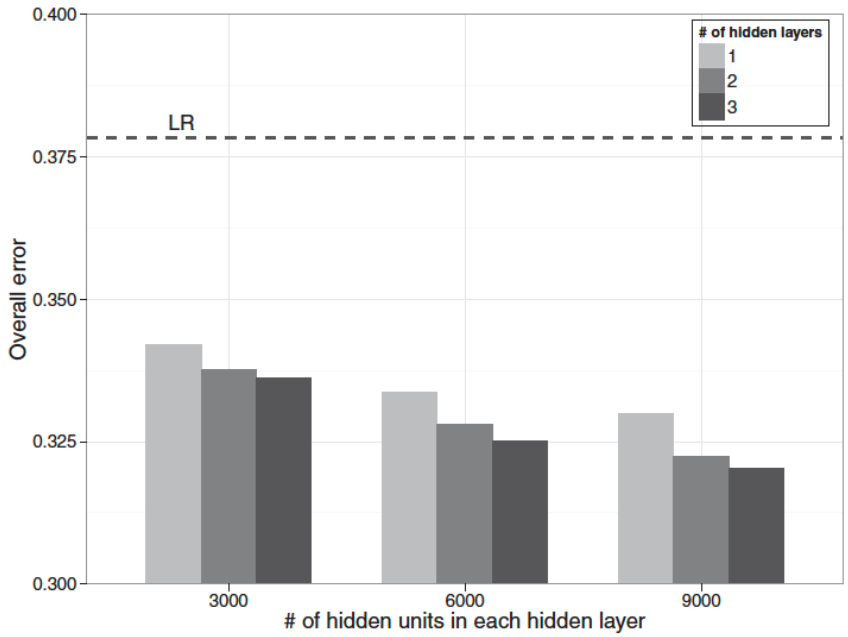
GEO data (Gene Expression Omnibus)

- Complete transcriptomes
- 129,158 samples after filtering and normalization
- Randomly partitioned into training, validation, and testing se mate ratio 8:1:1.
  - Training: train predictive models
  - Validation: model selection; parameter tuning
  - Testing: evaluate predictive models

Three methods

- Linear Regression (LRG)
  - Baseline model
  - Other variants: SVM, ridge regression, Lasso
- K Nearest Neighbor Regression (KNN)
  - K is tuned using validation data
  - Predict using average
  - Nonparametric nonlinear model
- Deep Learning

Test performance



Mean Absolute Error (MAE)

$$\mathrm{MAE}_{(t)} = \frac{1}{N'} \sum_{i=1}^{N'} |y_{i(t)} - \hat{y}_{i(t)}|$$

Fig. 1. The overall errors of D-GEX-10% with different architectures on GEO-te. The performance of LR is also included for comparison

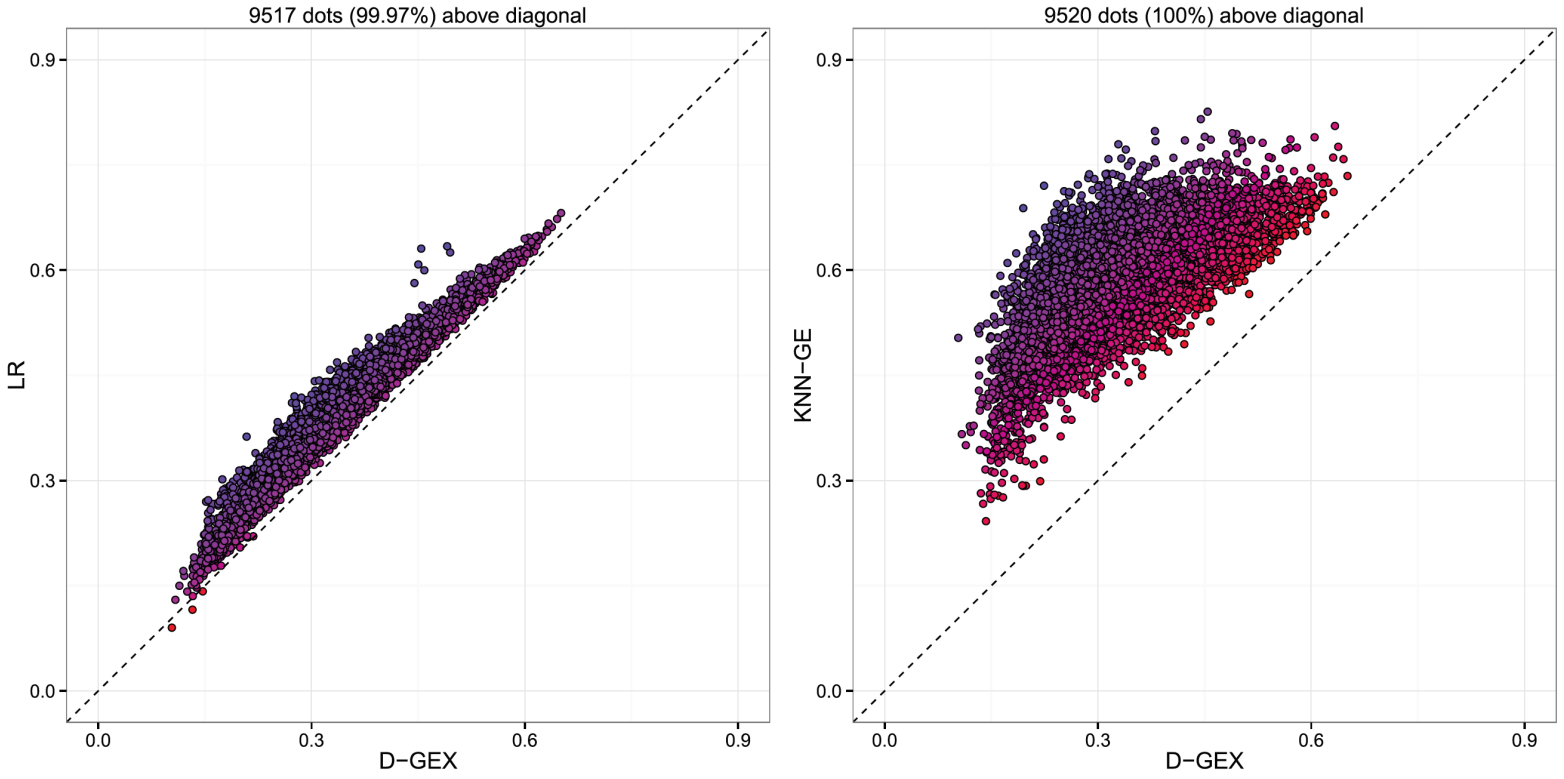


Fig. 3. The predictive errors of each target gene by GEX-10%-9000 $\times$ 3 compared with LR and KNN-GE on GEO-te. Each dot represents one out of the 9520 target genes. The *x*-axis is the MAE of each target gene by D-GEX, and the *y*-axis is the MAE of each target gene by the other method. Dots above diagonal means D-GEX achieves lower error compared with the other method. (**a**) D-GEX verse LR; (**b**) D-GEX verse KNN-GE

Table 2. The overall errors of LR, LR-L1, LR-L2, KNN-GE and D-GEX-25% with different architectures on GTEx-te

	Number of hidden units		
	3000	6000	9000
Number of hidden layers			
1	0.4507 ± 0.1231	0.4428 ± 0.1246	0.4394 ± 0.1253
2	0.4586 ± 0.1194	0.4446 ± 0.1226	**0.4393** ± **0.1239**
3	0.5160 ± 0.1157	0.4595 ± 0.1186	0.4492 ± 0.1211
LR		0.4702 ± 0.1234	
LR-L1		0.5667 ± 0.1271	
LR-L2		0.4702 ± 0.1234	
KNN-GE		0.6520 ± 0.0982	

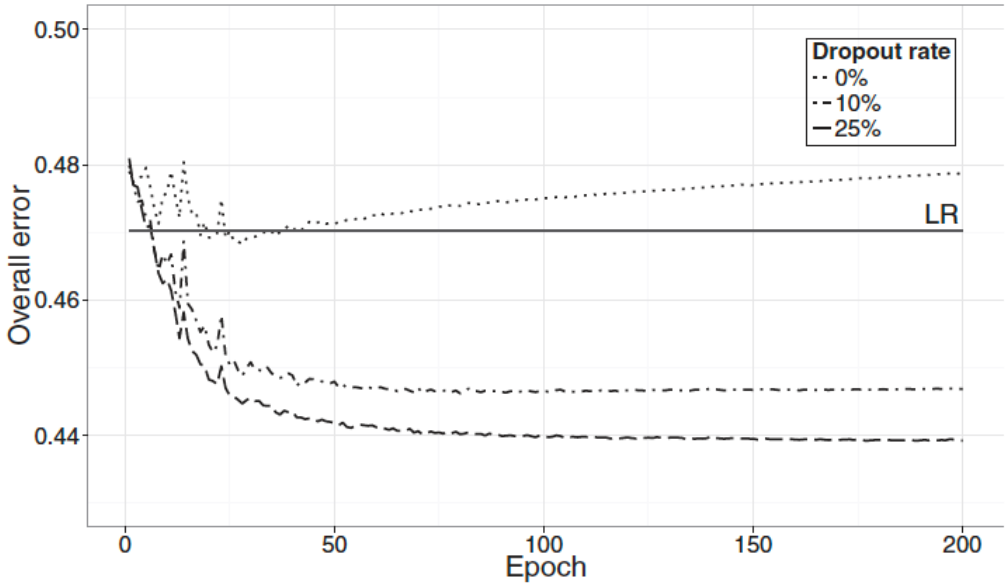
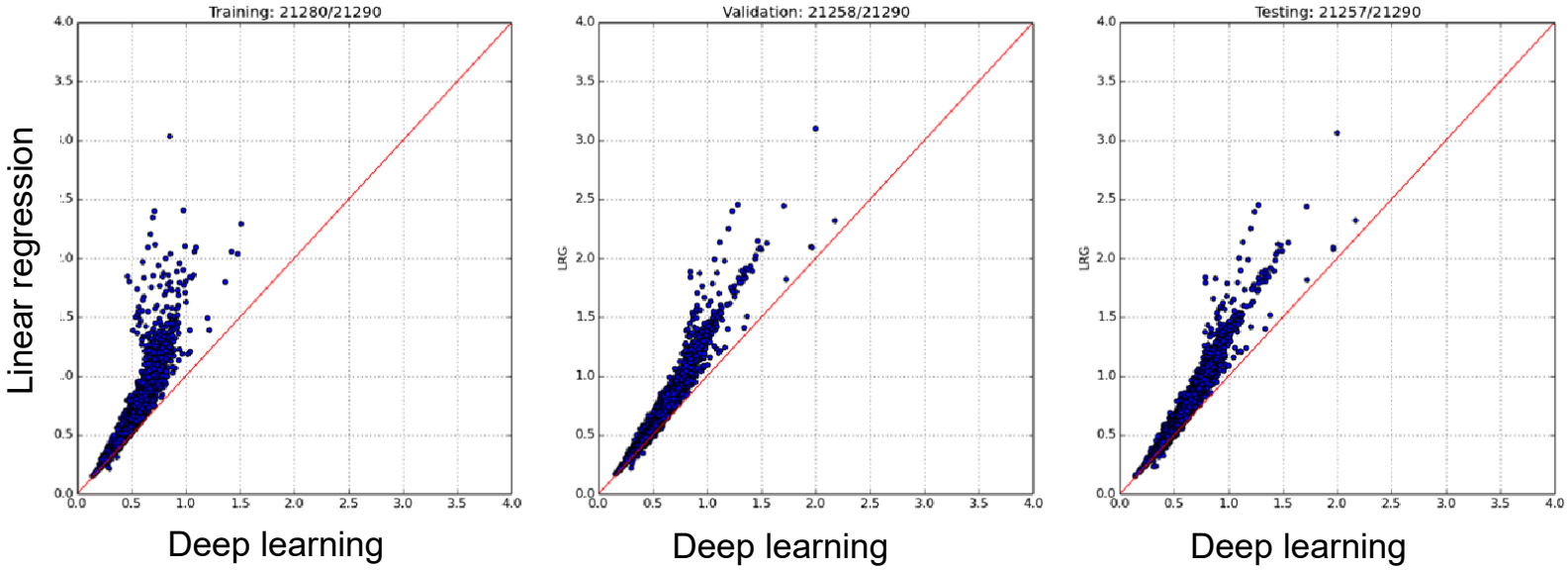


Fig. 5. The overall error decreasing curves of D-GEX-9000 $\times$ 2 on GTEx-te with different dropout rates. The *x*-axis is the training epoch and the *y*-axis is the overall error. The overall error of LR is also included for comparison

RMSE on Each Gene



Deep learning vs. linear regression

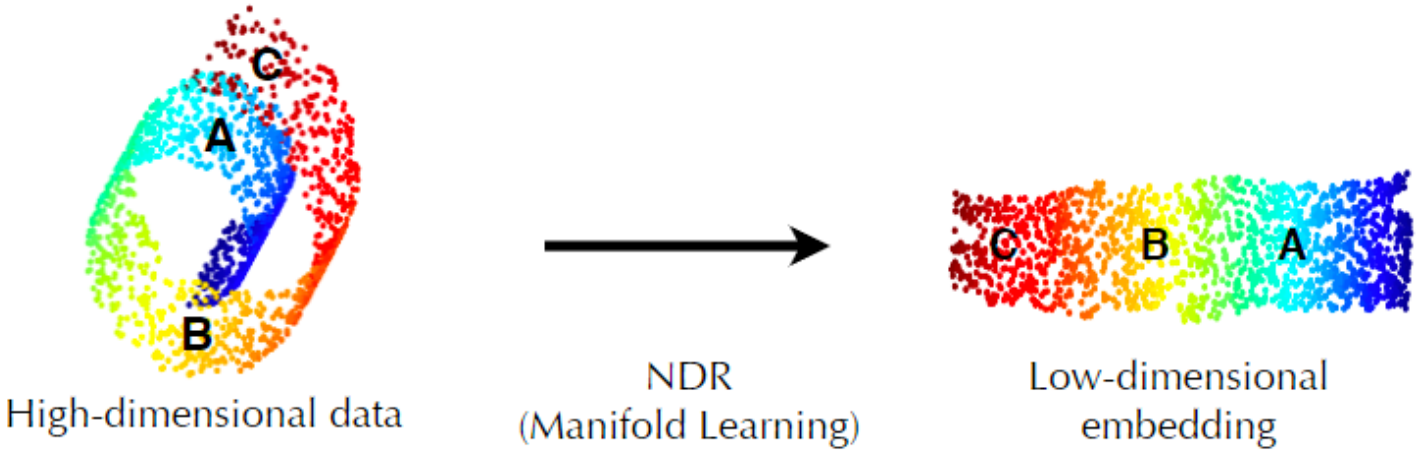
Summary Statistics

Percentage of genes on which deep learning does better than linear Regression: 99.98%

Percentage of genes on which deep learning does better than KNN: 97.90%

Deep Generative Models for Genomics

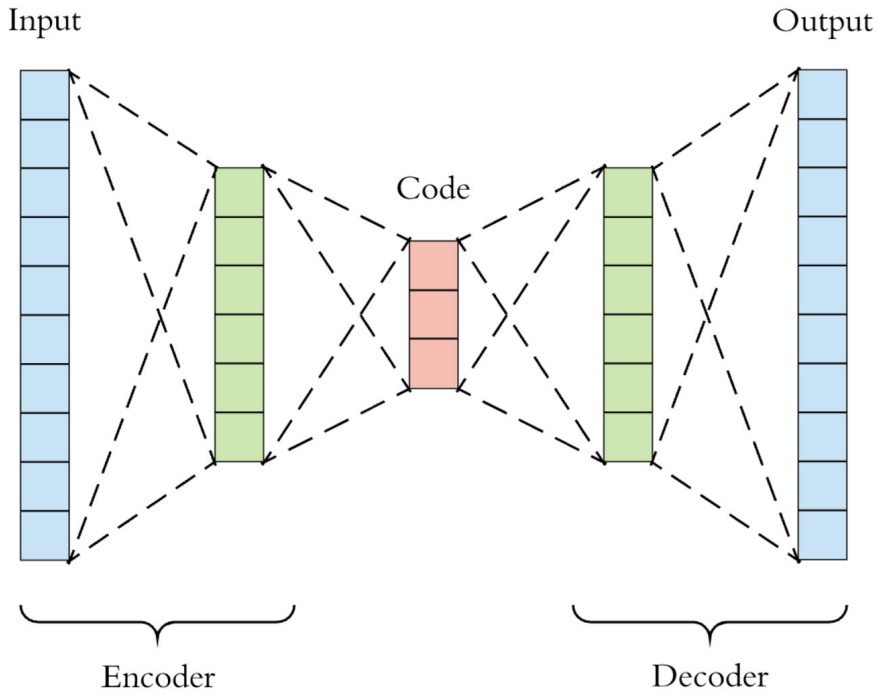
Manifold hypothesis



- Manifold hypothesis: high dimensional data (measurement) lie on low dimensional manifold embedded within the high-dimensional space.
- Need to discover the low dimensional representations (smooth manifold).
- Although biological data are complex and high-dimensional, we may understand them better if we study them within low-dimensional embedded spaces.

- Address these issues thorough manifold learning
- Manifold - smoothly varying low-dimensional structure embedded within high-dimensional ambient measurement space.
  Utilize manifold, representation, deep learning to understand large biomedical datasets.
- Give insights into diverse biological systems

Discover latent representation through autoencoder

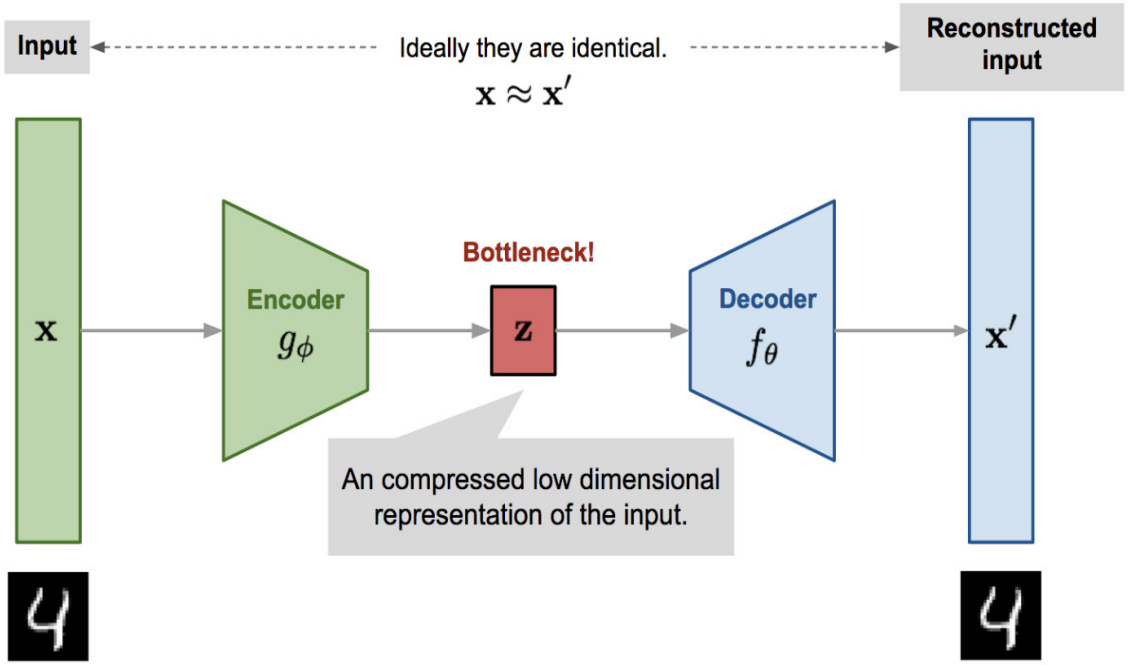


$$\phi : \mathcal{X} \to \mathcal{F}$$
$$\psi : \mathcal{F} \to \mathcal{X}$$
$$\phi, \psi = \arg\min_{\phi,\psi} \|X - (\psi \circ \phi)X\|^2$$

Disadvantages of traditional autoencoder:

1. No constraints on the latent representations, e.g., gaps in latent space.
2. Susceptible to overfitting, e.g., memorize the input.
3. Not clear how to generate a new sample.

Autoencoder model architecture

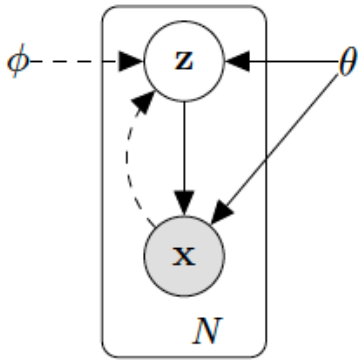


Disadvantages of traditional autoencoder:

1. No constraints on the latent representations, e.g., gaps in latent space.
2. Susceptible to overfitting, e.g., memorize the input.
3. Not clear how to generate a new sample.

https://lilianweng.github.io/lil-log/

Generative models with latent variables



Given dataset:

$$\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(n)}\}$$

Model marginal likelihood with latent variable z:

$$p_\theta(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[p_\theta(\mathbf{x}|\mathbf{z})]$$

Negative log-likelihood (NLL) function as a loss:

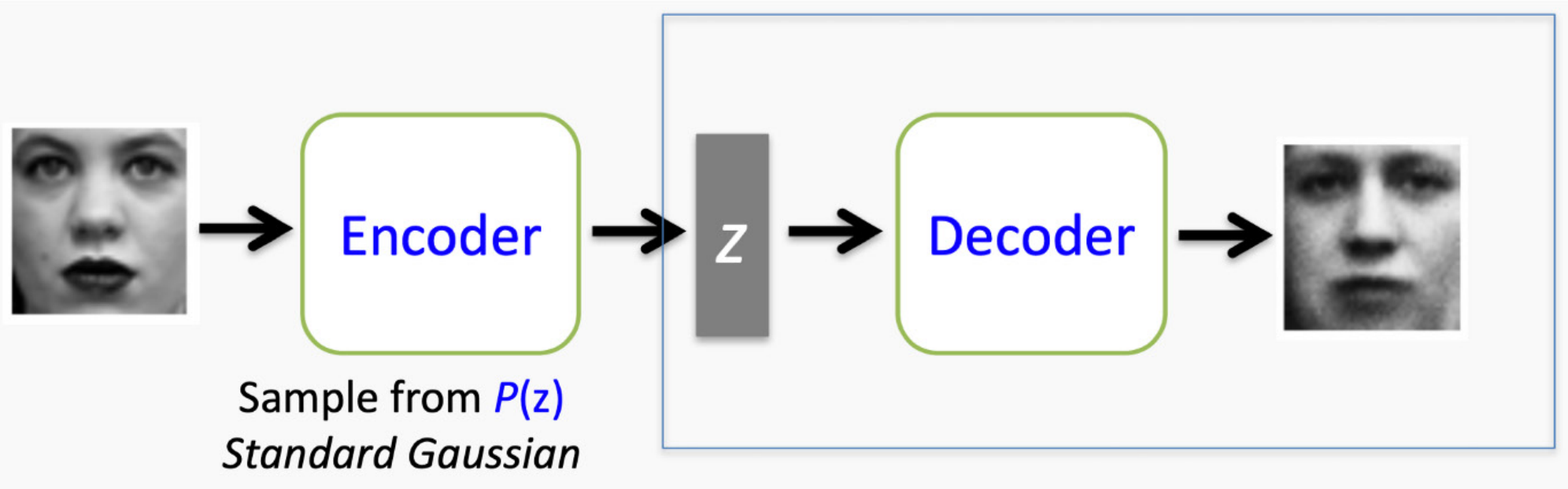
$$\mathcal{L} = -\sum_{i=1}^{n} \log p_\theta(\mathbf{x}^{(i)})$$

Variational lower bound on the marginal likelihood:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))$$

$q_\phi(\mathbf{z}|\mathbf{x})$ is a variational approximation of the intractable posterior $p_\theta(\mathbf{z}|\mathbf{x})$

Variational autoencoder (VAE)

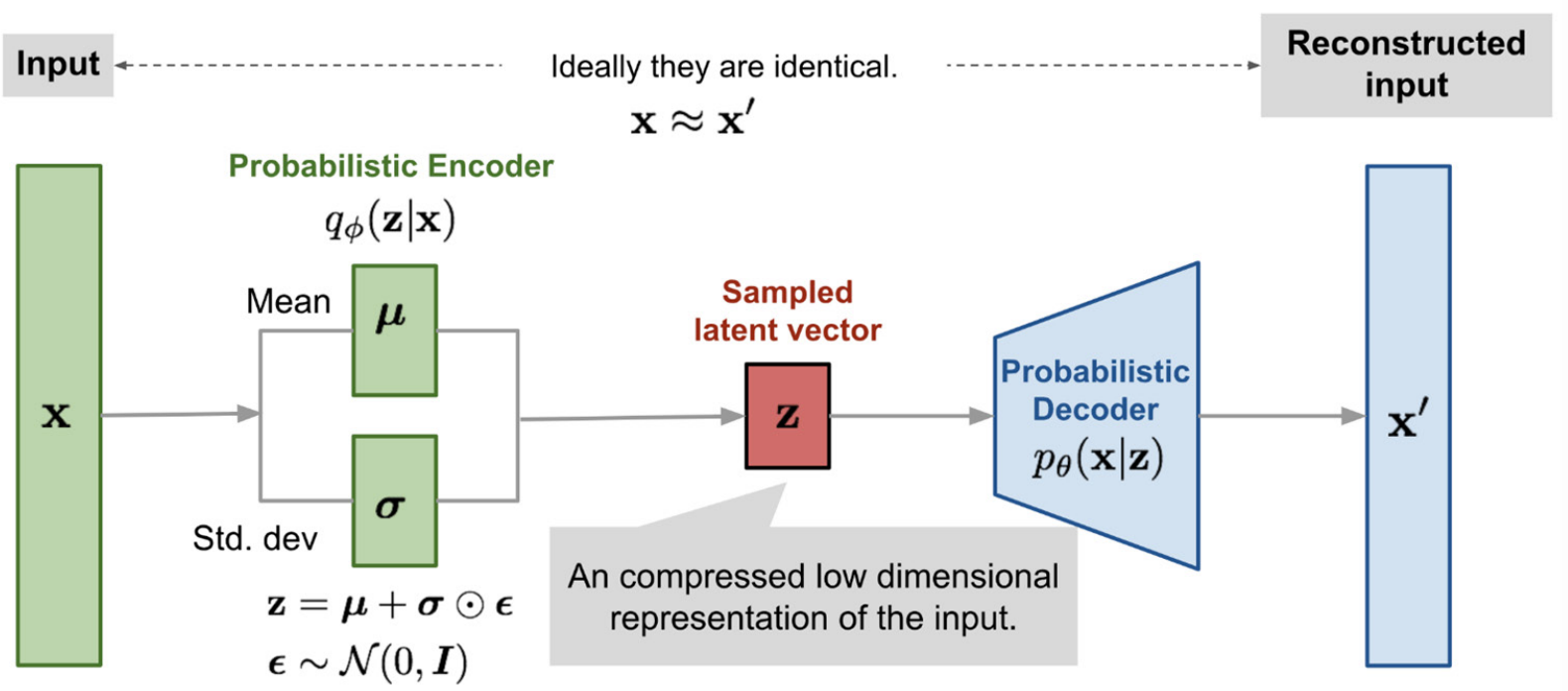


$$L_{\text{VAE}} = -\text{ELBO} = E_{q(X)}\left[-E_{q_\phi(Z|X)}p_\theta(X \mid Z) + \text{KL}\left\{q_\phi(Z \mid X)||p(Z)\right\}\right]$$

The first term is reconstruction error
The second term is the Kullback-Leibler divergence between the posterior and prior distributions of the latent variables (Z).

Kingma and Welling, 2013

VAE with Gaussian prior, reparameterization trick



https://lilianweng.github.io/lil-log/

beta-VAE

$$L_{\mathrm{BETA}}(\phi, \beta) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$

Motivations:

- Emphasize the disentanglement between different latent variables, z1, z2,..., zn.
- The prior of p(z) assumes different latent variables are independent.
- Larger weight beta on the second term leads to better disentanglement between latent variables.

Higgins et al, ICLR 2017

Deep Learning for scRNA-seq

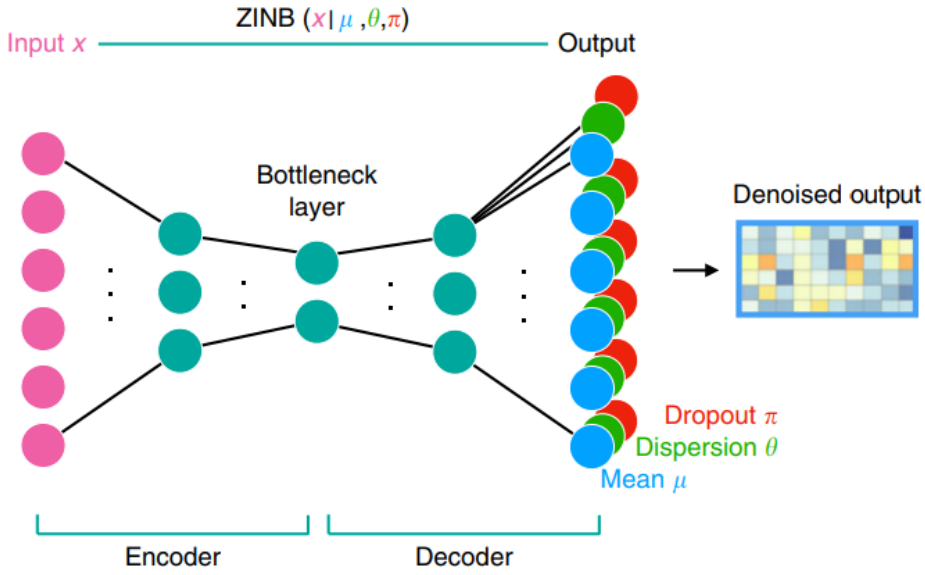
- SAUCIE
- DCA (Denoising, Imputation)
- scVI
- totalVI
- Solo (Doublet Identification)
- DeepImpute (Imputation)
- scAlign (Batch effect, Integration)

DCA—Denoising Count Autoencoder

- Autoencoder (AE) with Zero-inflated Negative Binomial (ZINB) loss function
- Negative binomial models the mean μ and dispersion θ of RNA-seq count
- Zero inflation with a point mass π models the dropout events
- ZINB provides great denoising performance, which benefits downstream analysis, including clustering, time course modeling, differential expression, protein-RNA co-expression and pseudo time analysises.

$$\mathrm{NB}(x;\mu,\theta) = \frac{\Gamma(x+\theta)}{\Gamma(\theta)}\left(\frac{\theta}{\theta+\mu}\right)^{\theta}\left(\frac{\mu}{\theta+\mu}\right)^{x}$$

$$\mathrm{ZINB}(x;\pi,\mu,\theta) = \pi\delta_0(x) + (1-\pi)\mathrm{NB}(x;\mu,\theta)$$



Eraslan, Gökcen, et al. "Single-cell RNA-seq denoising using a deep count autoencoder." *Nature communications* 10.1 (2019): 1-14.

scVI—Single cell variational inference

- Variational Autoencoder (VAE) with Zero-inflated Negative Binomial (ZINB) likelihood accounting for the count nature of RNA-seq data and dropout events during sequencing process
- Exceptional performance for imputation with ZINB loss function
- Generative modeling for Imputation and data simulation

Lopez, Romain, et al. "Deep generative modeling for single-cell transcriptomics." *Nature methods* 15.12 (2018): 1053-1058.

Regulatory and Functional Genomics

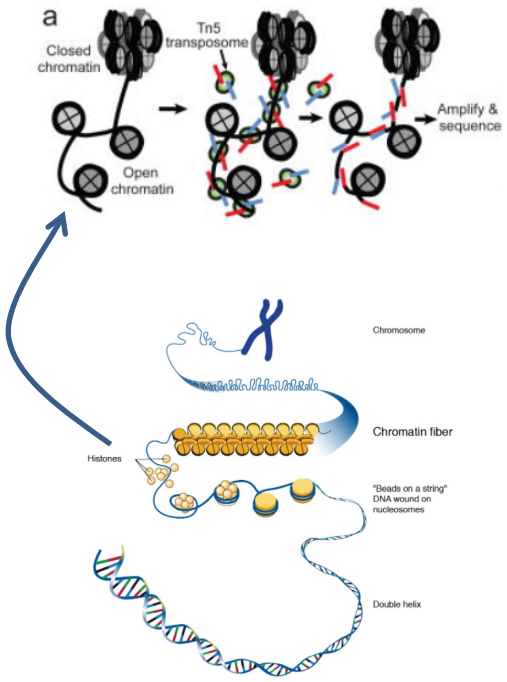
SAILER: Scalable and Accurate Invariant Representation Learning for Single-Cell ATAC-Seq Processing and Integration

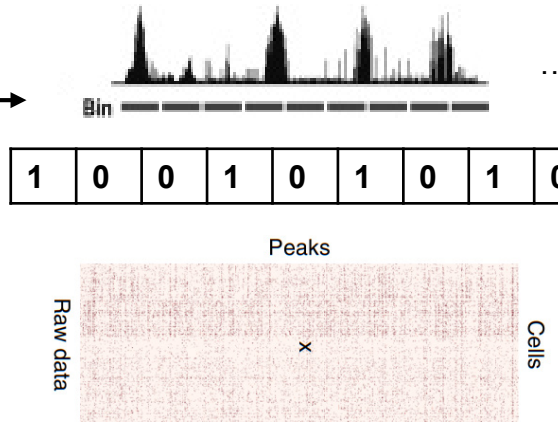
Yingxin Cao[1,5,6,†], Laiyi Fu [1,2,†], Jie Wu[3], Qinke Peng[2], Qing Nie[4,5,6], Jing Zhang[1,*], Xiaohui Xie[1,*]

https://www.biorxiv.org/content/10.1101/2021.01.28.428689v1.abstract

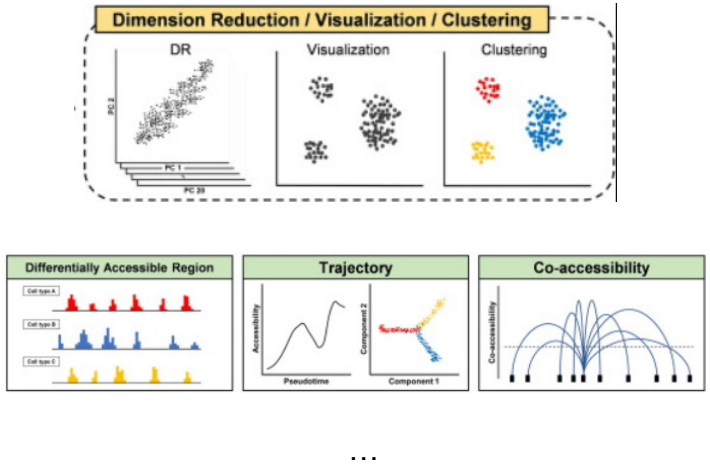
Chromatin accessibility

scATAC-seq (single cell ATAC-seq)





$x \in \{0,1\}^{n \times m}$



- High dimension, difficult to interoperate
- First step, dimensional reduction, clustering
- Goal: Learn a representation informative on biological variations, while remain invariant to confounding factors

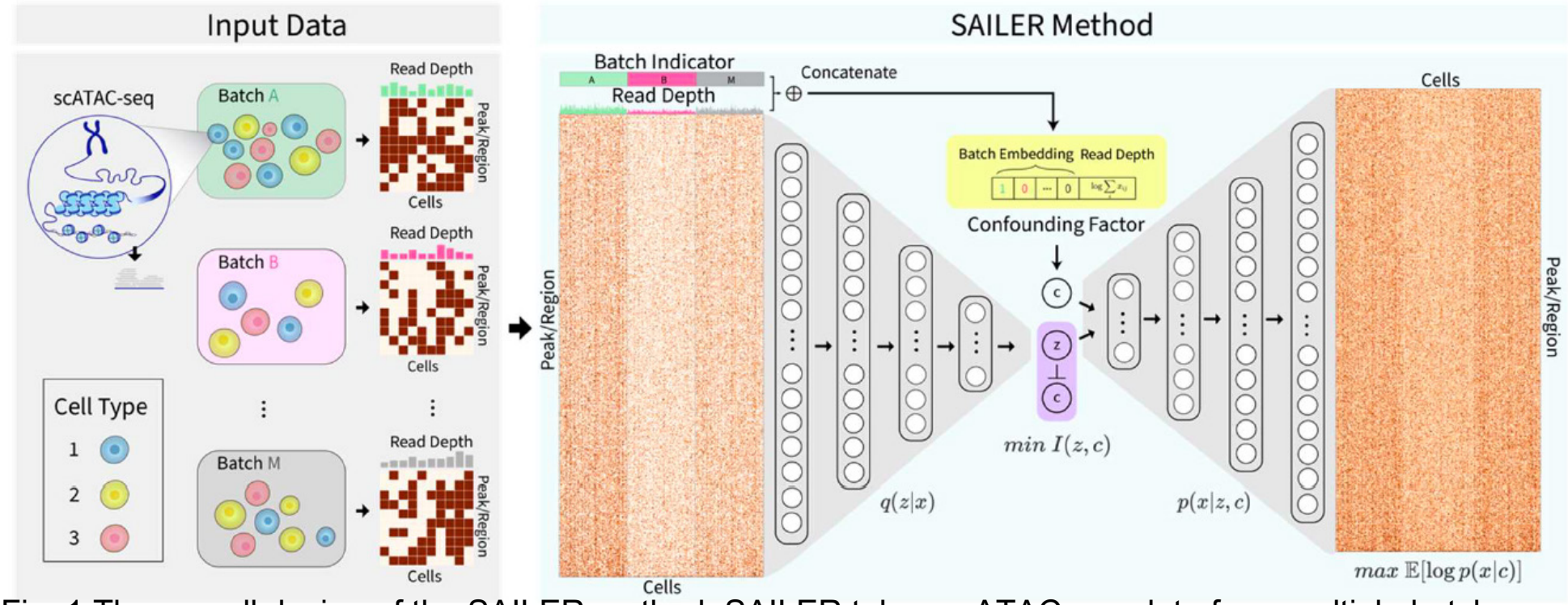
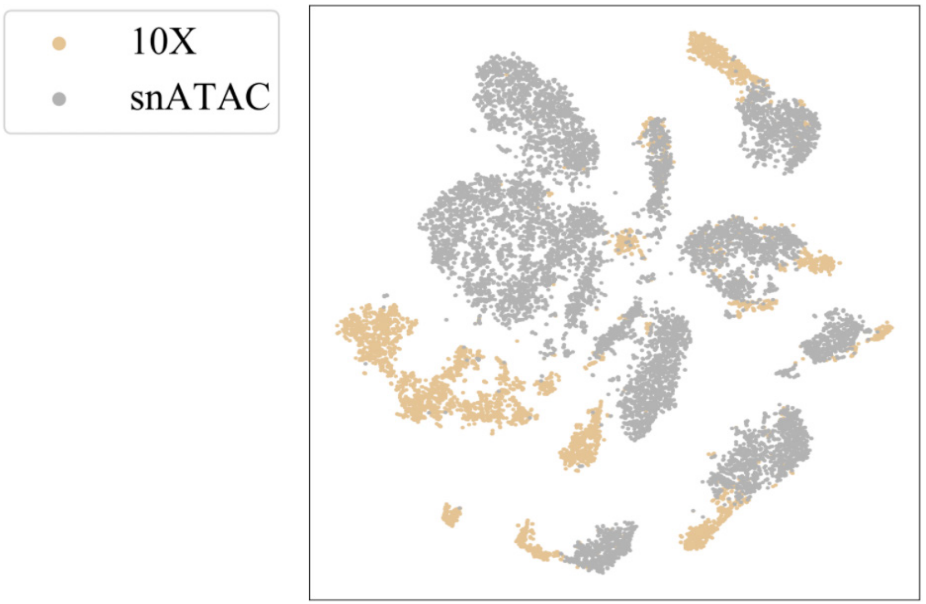
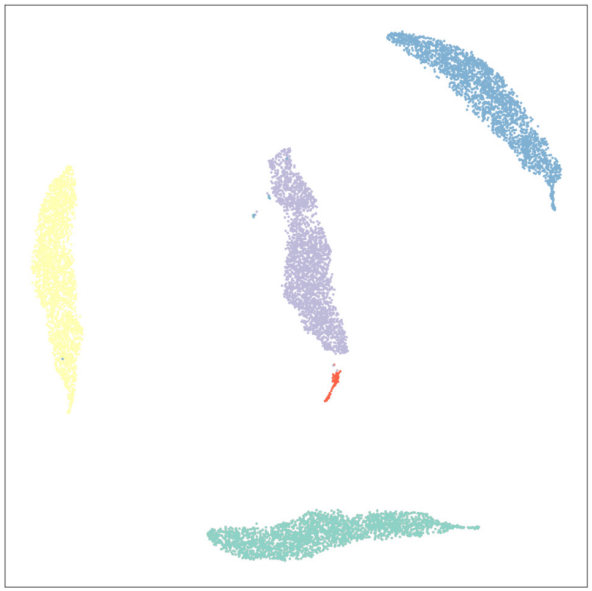


Fig. 1 The overall design of the SAILER method. SAILER takes scATAC-seq data from multiple batches as input. Raw data is pushed through the encoder network to obtain a latent representation. Confounding factors for each single cell are concatenated and fed to the decoder along with the latent representation. Batch information is indicated by a one-hot embedding, and read depth is subject to log transform and standard normalization. To learn a latent representation invariant to changes in confounding factors, mutual information between the latent variables and confounding factors are minimized during training.

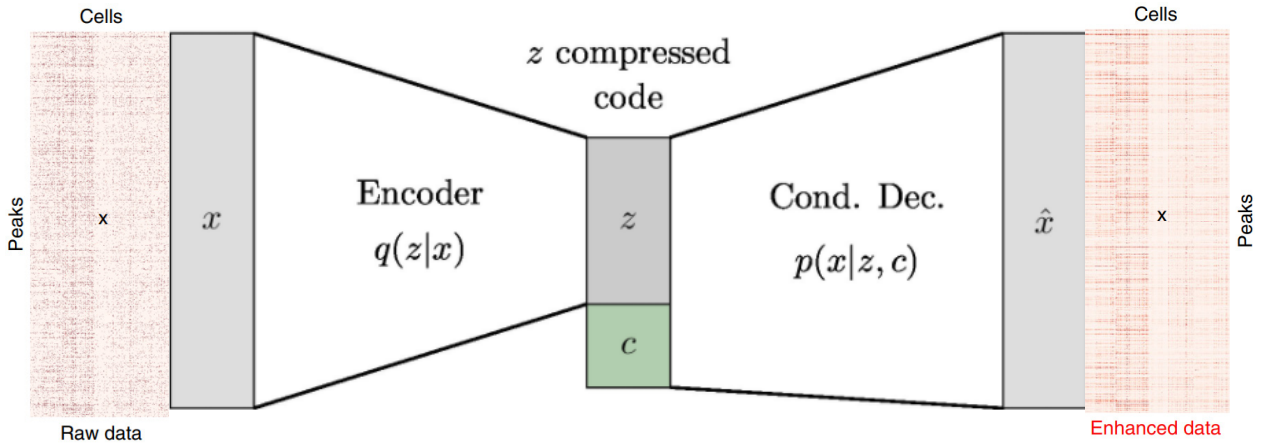
Confounding factors: batch effect



Confounding factors: read depth



Conditional VAE



Goal:

To learn a representation informative on biological variations, while remain invariant to confounding factors

Method:

Invariant Coding through VAE

Moyer,D. *et al. NIPS*, 2018

Objective:

Maximize a log-likelihood conditioned on the confounding factors, while minimize the mutual information between latent variable z and confounding factor c.

$$\max \mathbb{E}_{(x,c)}[\log p(x|c)] - \lambda I(z,c).$$

Learning invariant representations

Variational loss

$$L_{\text{VAE}} = \mathbb{E}_{\mathbf{x},\mathbf{c}\sim q(\mathbf{x},\mathbf{c})}\left[-\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z}))\right]$$

Minimizing both variation loss and mutual information between latent and conditional variables

$$L_{\text{VAE}} + \lambda I(\mathbf{z};\mathbf{c}) \qquad\qquad q_\phi(\mathbf{z},\mathbf{x},\mathbf{c}) = q(\mathbf{x},\mathbf{c})q_\phi(\mathbf{z}|\mathbf{x})$$

Approximation of the loss function:

$$L(\phi,\theta) = \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}\left[D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})) + \lambda D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z}))\right]$$
$$-(1+\lambda)\mathbb{E}_{\mathbf{x},\mathbf{c}\sim q(\mathbf{x},\mathbf{c})}\left[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})]\right]$$

$$D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z})) \approx \sum_{\mathbf{x}}\sum_{\mathbf{x}'} D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel q_\phi(\mathbf{z}|\mathbf{x}'))$$

Moyer et al, NeurIPS, 2018

SAILER learns robust latent cell representations invariant to various confounding factors

- Simulated data

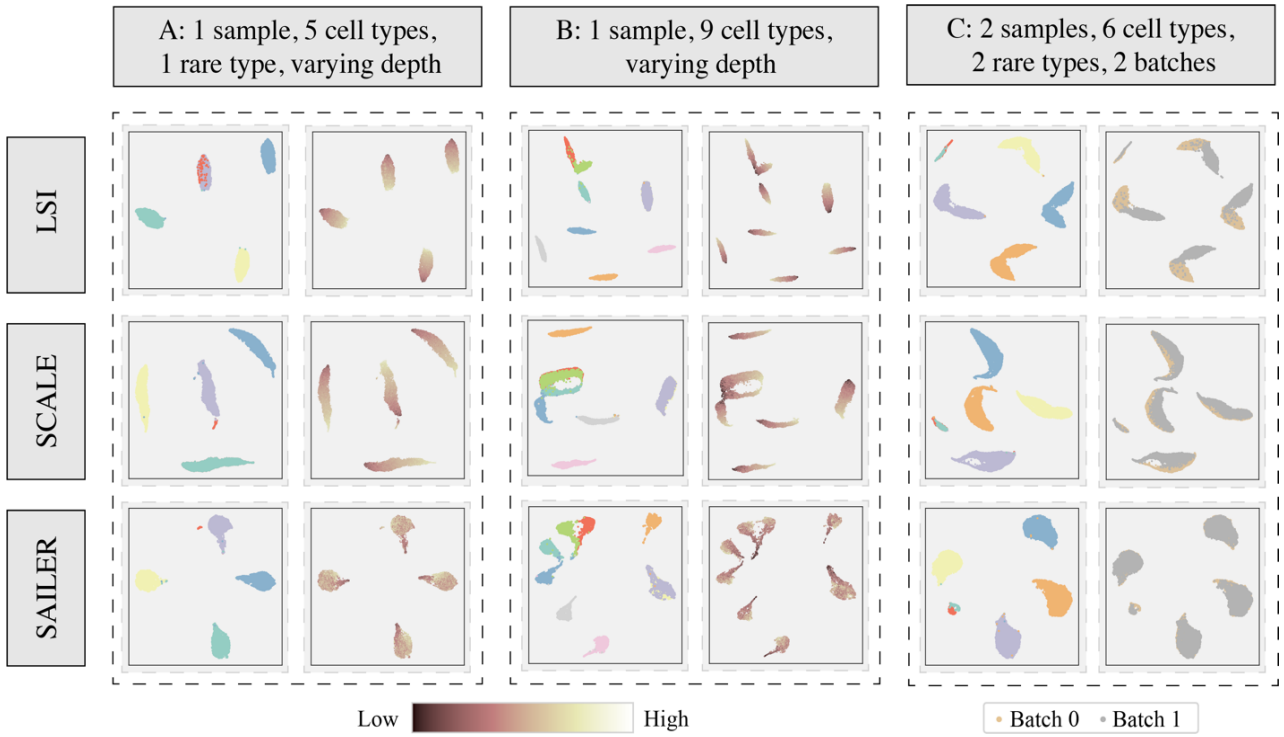


Table 1 Mutual Information between the latent representation and confounding factors on simulation datasets.

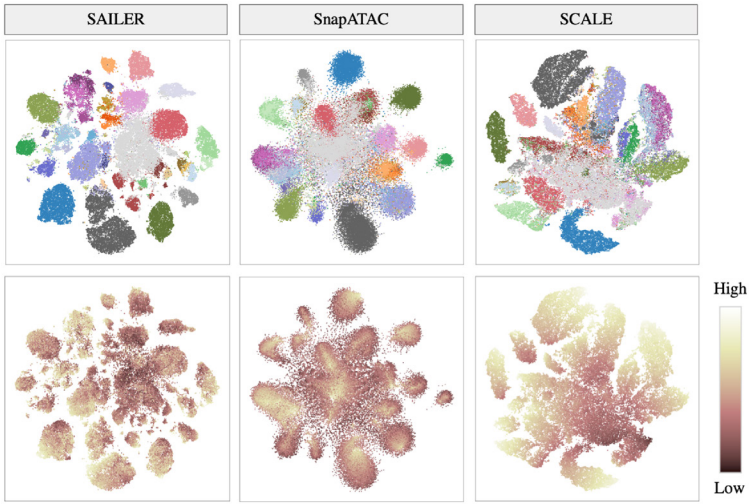
Method \ $I(\mathbf{z},\mathbf{c})$	Sim1	Sim2	Sim3
LSI	0.610	0.500	0.130
SCALE	0.290	0.224	0.087
SAILER	**0.107**	**0.100**	**0.005**

SAILER learns robust latent cell representations invariant to various confounding factors

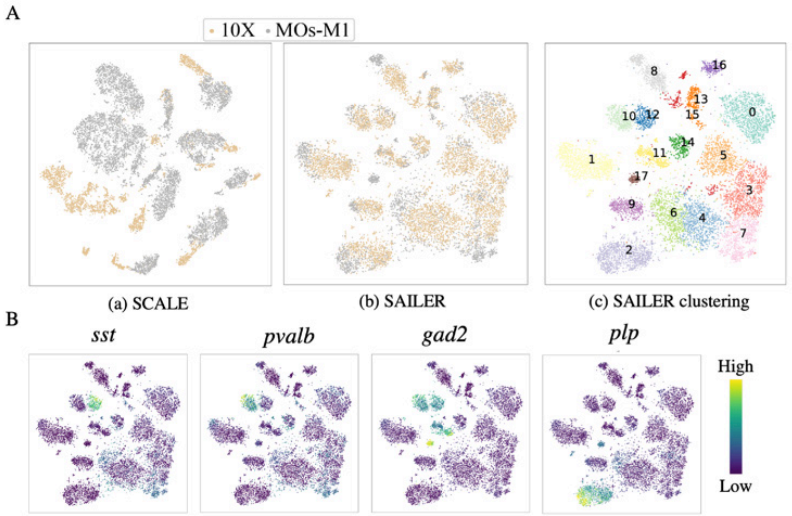
- Mouse Atlas Data

Table 2 Evaluation results on the mouse atlas dataset

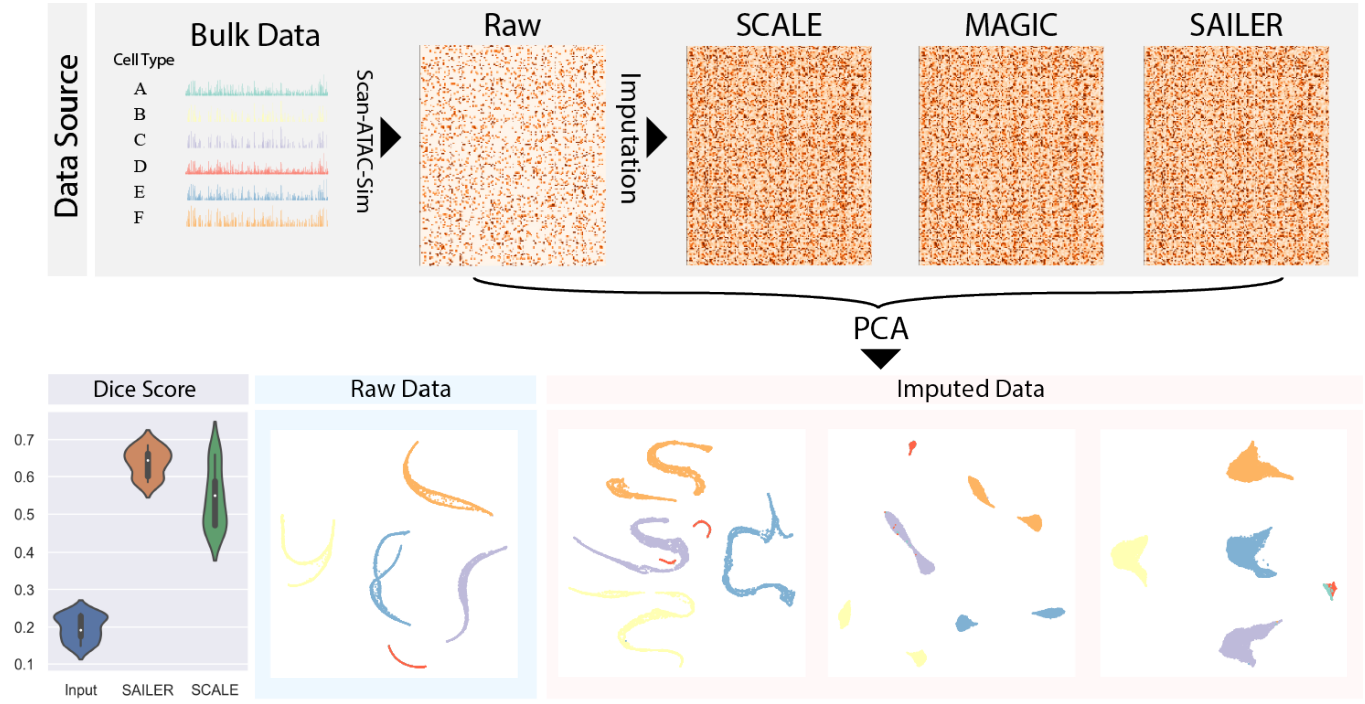
Method	ARI	NMI	$I(\boldsymbol{z}, \boldsymbol{c})$
SAILER	**0.575**	**0.799**	**0.040**
SnapATAC	0.538	0.748	0.127
SCALE	0.315	0.557	0.279



- Merging two mouse brain datasets

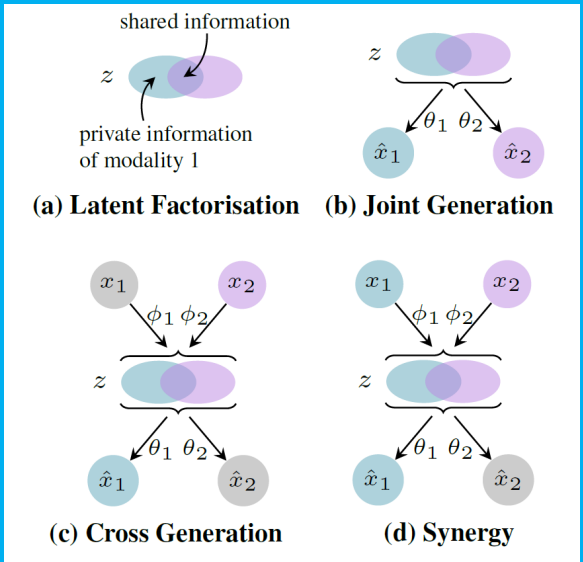


SAILER reconstructs a chromatin accessibility landscape free of various confounding factors



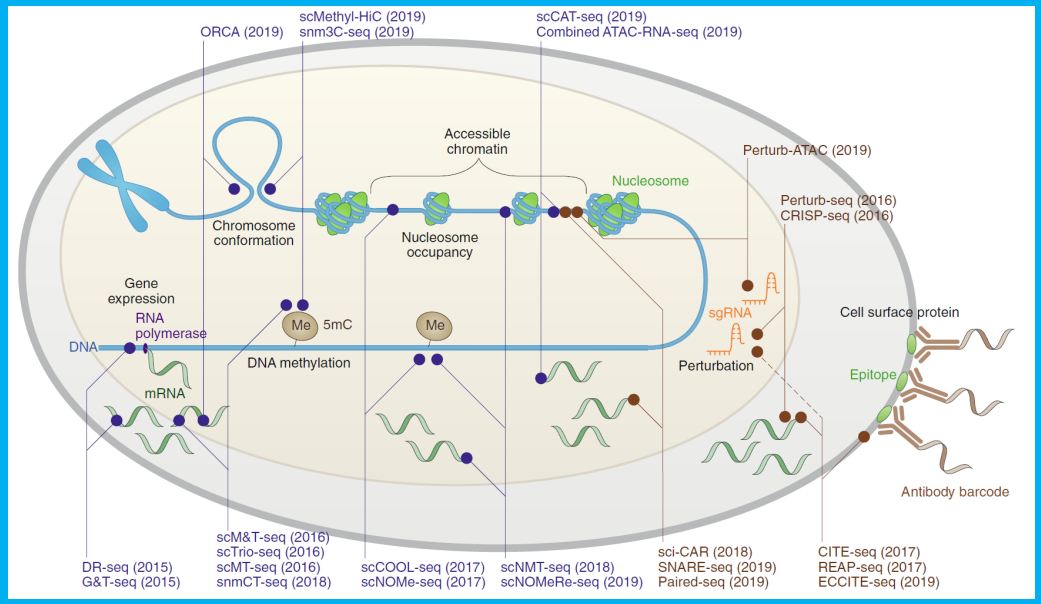
Multimodal Deep Learning for Single Cell Multimodal Omics

- Multimodal deep generative model



Shi, Yuge, et al. *NIPS* 2019

- Multimodal single cell omics methods

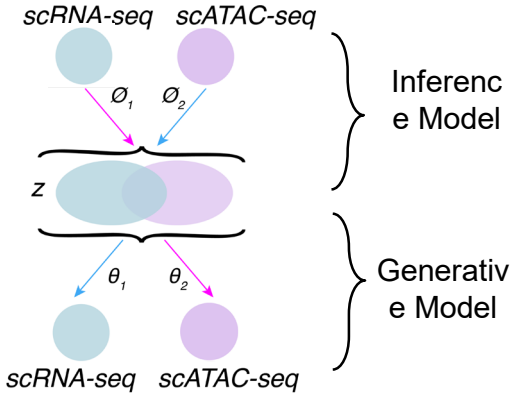


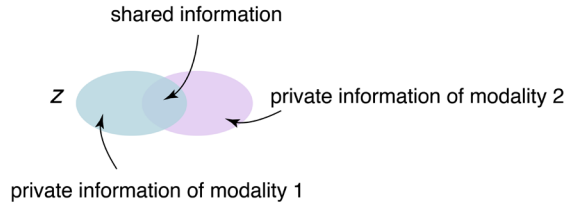
Zhu, Chenxu, Sebastian Preissl, and Bing Ren. *Nature methods* 2020

- Shi, Yuge, et al. "Variational mixture-of-experts autoencoders for multi-modal deep generative models." *Advances in Neural Information Processing Systems*. 2019.
- Zhu, Chenxu, Sebastian Preissl, and Bing Ren. "Single-cell multimodal omics: the power of many." *Nature methods* 17.1 (2020): 11-14.

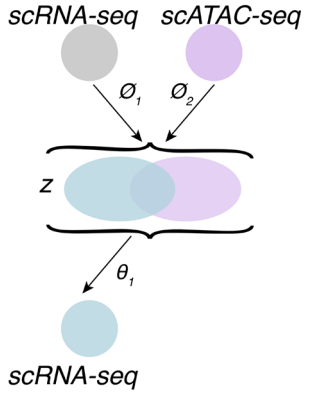
MVAE for Jointly Profiled scRNA-seq and scATAC-seq Data

- MVAE Framework

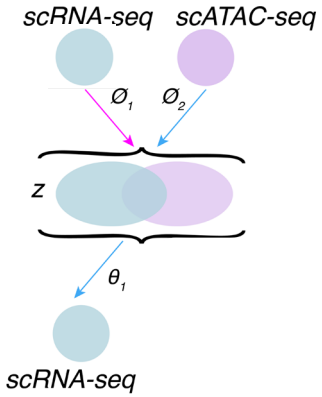




- Factorized latent space for different downstream tasks
  - Private latent space for modality specific tasks (e.g. imputation)
  - Shared latent space for common tasks (e.g. cell states, cell identity)

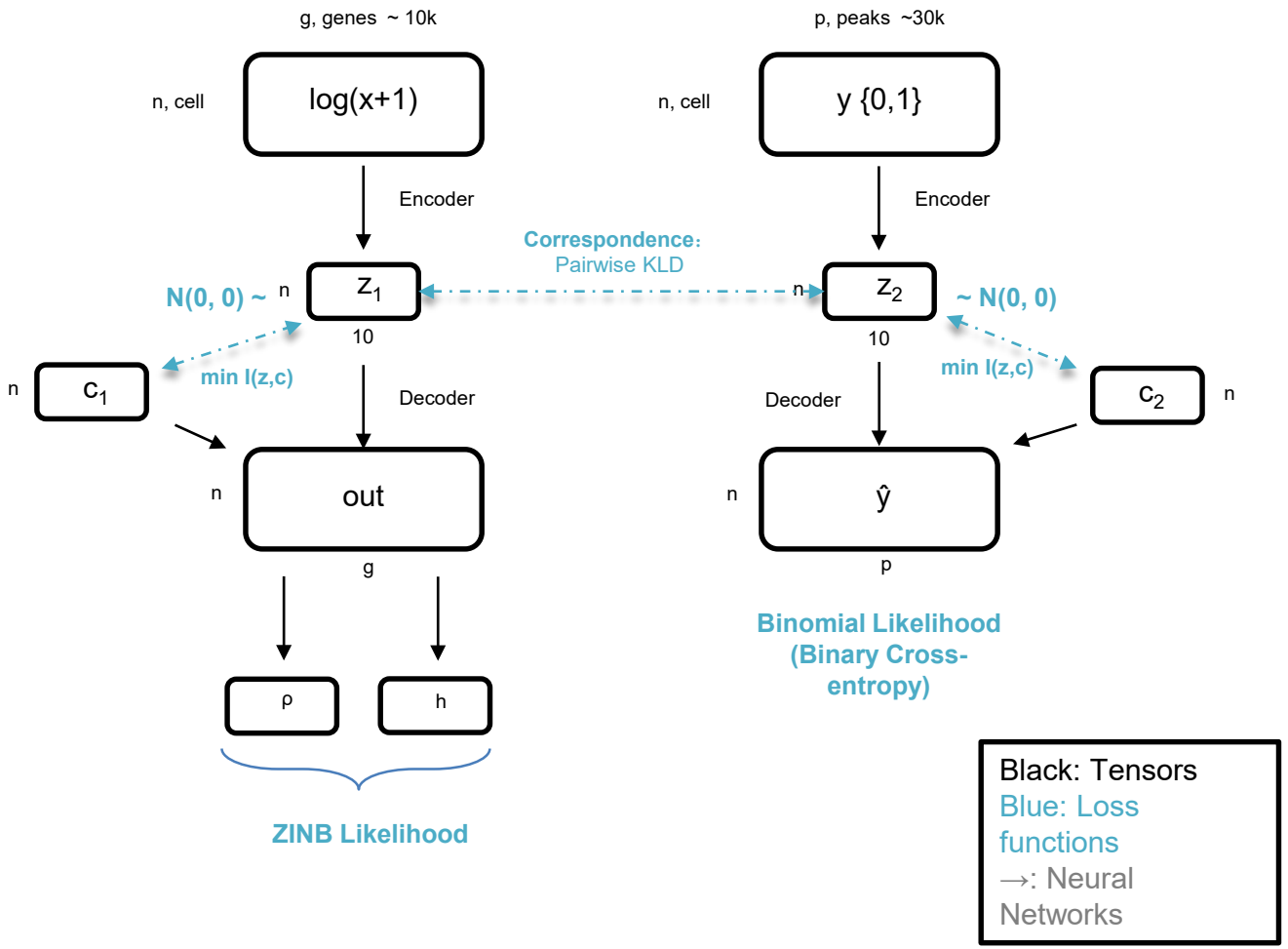


- Cross Generation
  - Generate corresponding missing modality

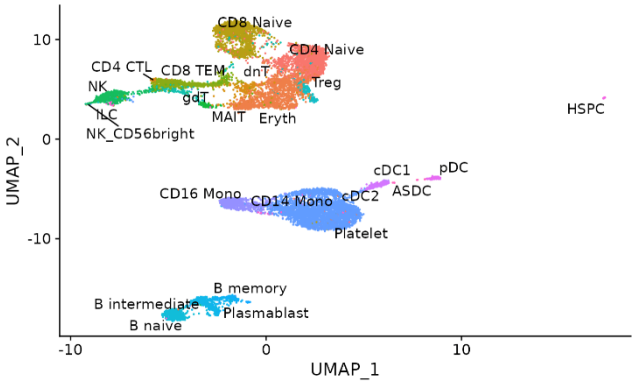


- Synergy
  - Better inference on single modality when both are observed (e.g. better clustering, imputations)

Datasets:
n = 34,774 cells for SHARE-seq
n ~ 10k cells for PBMC 10x



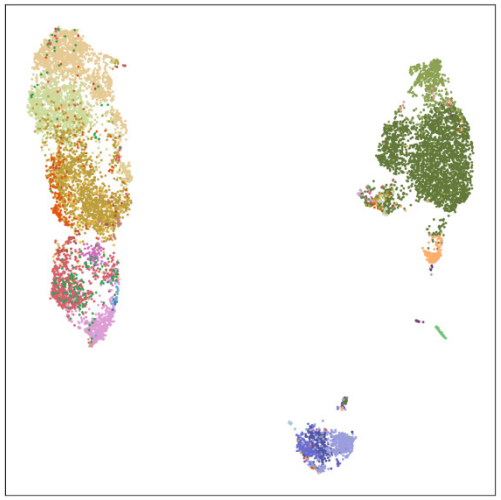
VAE

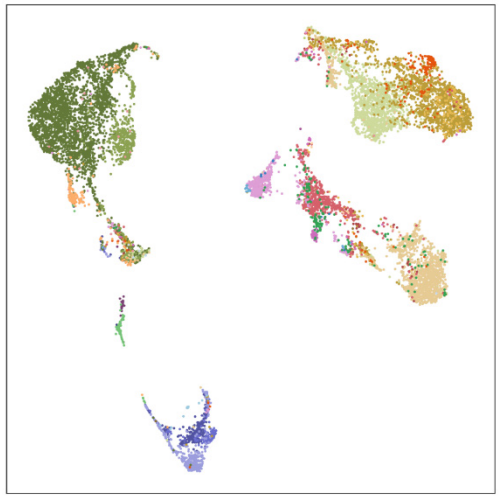


RNA

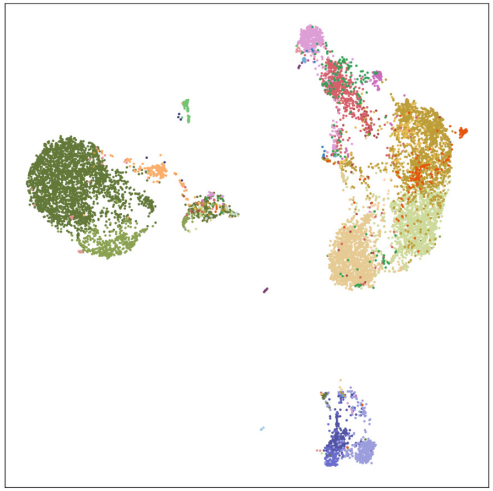
ATAC

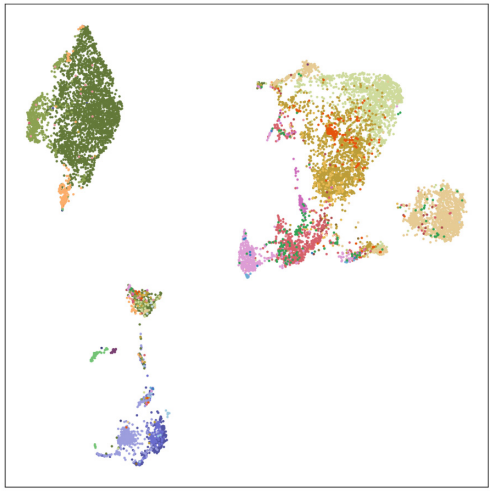
Raw



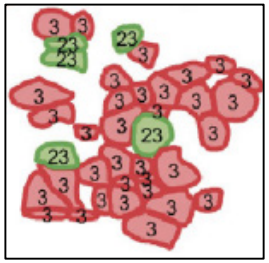


Filtered
~15,000
genes

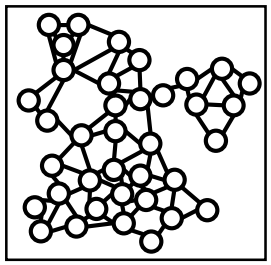




Variational Graph Autoencoders for Spatial Transcriptome Analysis

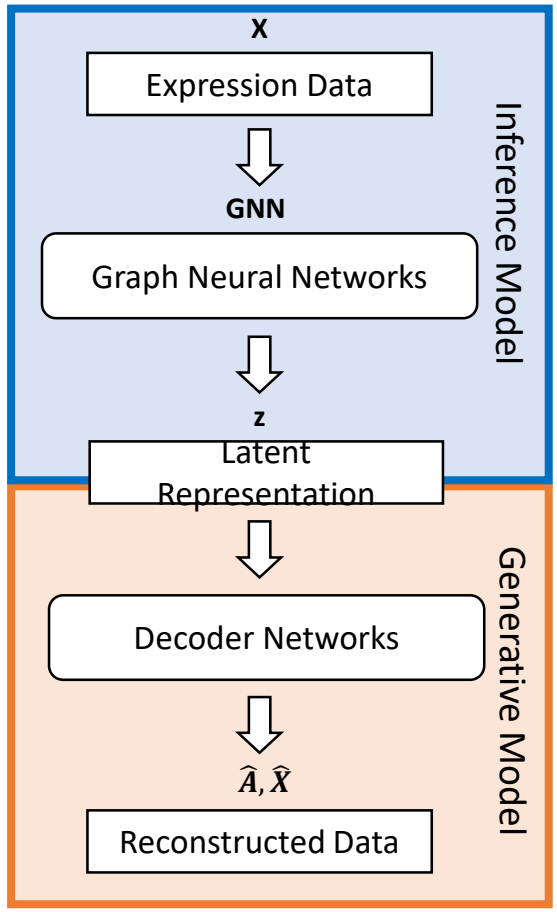


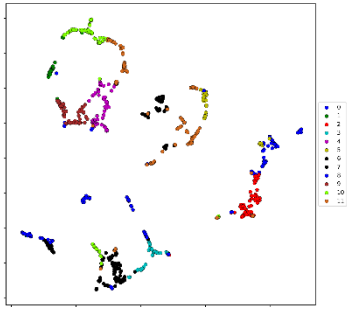
Spatial Data



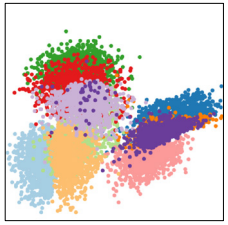
Neighborhood Graph

$$q(z_i|X,A) = \mathcal{N}\left(z_i|\mathrm{GNN}_\mu(X,A), \mathrm{diag}\left(\mathrm{GNN}_\sigma(X,A)\right)\right)$$





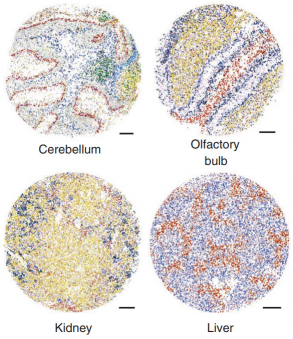
Spatial Context



Cell Identity

Slide-seq

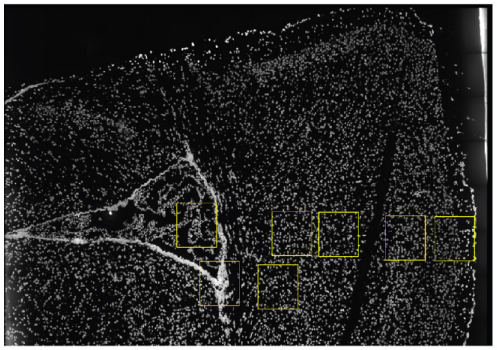
~10,000 cells, ~10,000 genes



Rodriques, Samuel G., et al. Science 2019

SeqFISH+

~1,000 cells, ~10,000 genes



Eng, Chee-Huat Linus, et al. *Nature* 2019

Eng, Chee-Huat Linus, et al. "Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+." *Nature* 568.7751 (2019): 235-239.
Rodriques, Samuel G., et al. "Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution." *Science* 363.6434 (2019): 1463-1467.

Today: Predicting gene expression and splicing

0. Intro: 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. Guest Lecture: Flynn Chen, Mark Gerstein Lab, Yale
   - Predicting Reporter Expression from Chromatin Features
6. Guest Lecture: Xiaohui Xie, UC Irvine
   - Predicting Gene Expression from partial subsets sampling
   - Representation learning for multi-omics integration
7. Guest Lecture: Kyle Kai-How Farh, Illumina
   - Predict splicing from sequence

Predicting Splicing from Primary Sequence

Kyle Farh, MD, PhD (kfarh @ illumina.com)

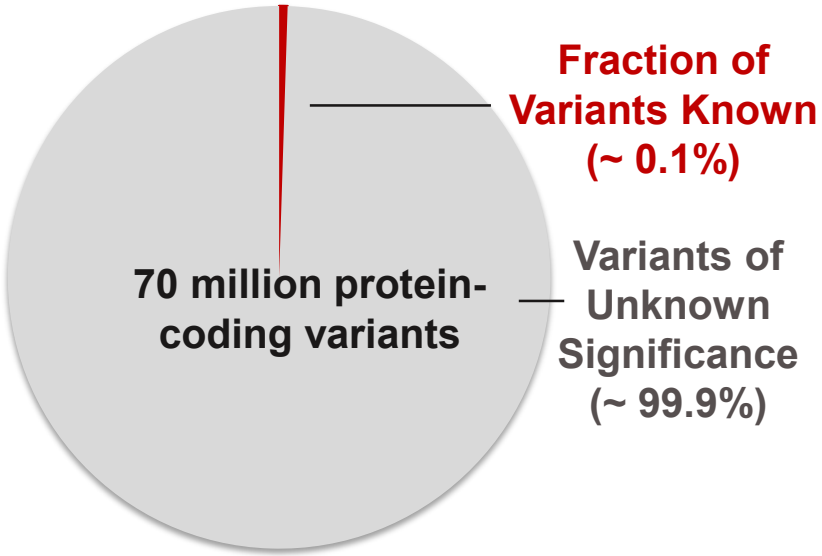
Principal Investigator, Illumina AI Lab

illumina®

Current State: Our Understanding of the Genome is Nascent

Level of actionability of whole genome is < 1% of its potential

- **Rare genetic disease**
  - Most cases remain unsolved despite WGS
  - 99% of variants are **VUS** (unknown significance)
- **Oncology**
  - 99% of the genome is noncoding, and largely uninterpreted
- **General Population**
  - Minimal actionability for common diseases
  - Little incentive for patients to be sequenced in routine care

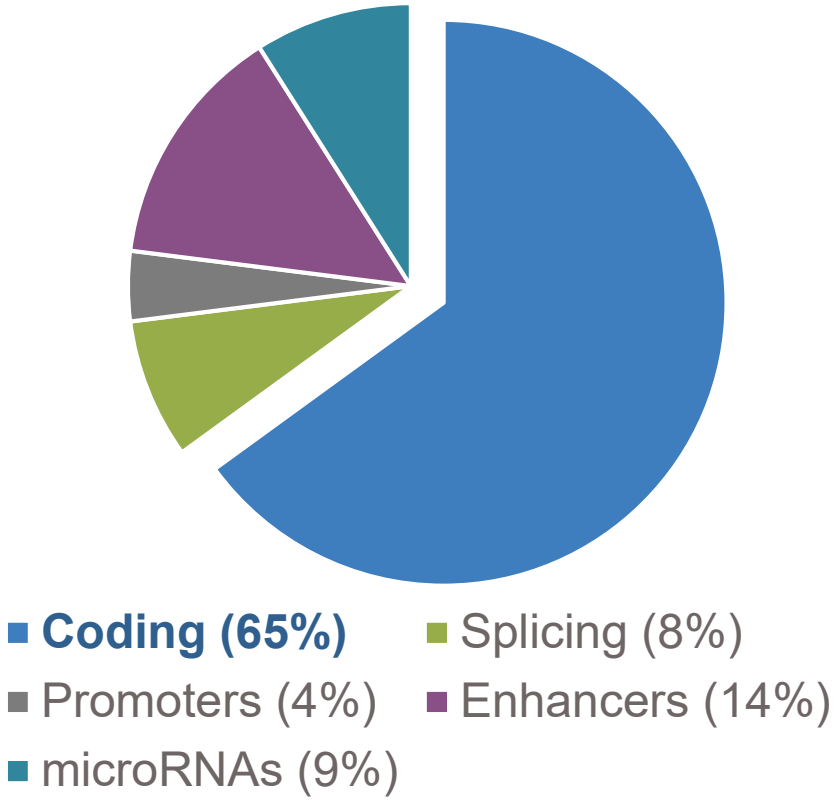


Unlock the diagnostic yield of the noncoding genome

- Currently, **99%** of the genome that is noncoding is ignored for diagnostic sequencing.

- Current diagnostic yield with exome alone for rare disease ~ **25-30%**

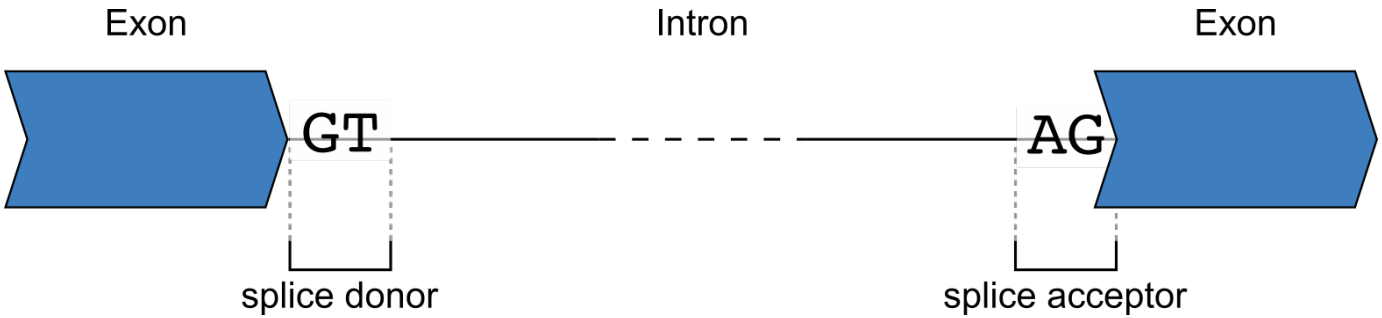
Deeply conserved sequence, PhyloP > 3



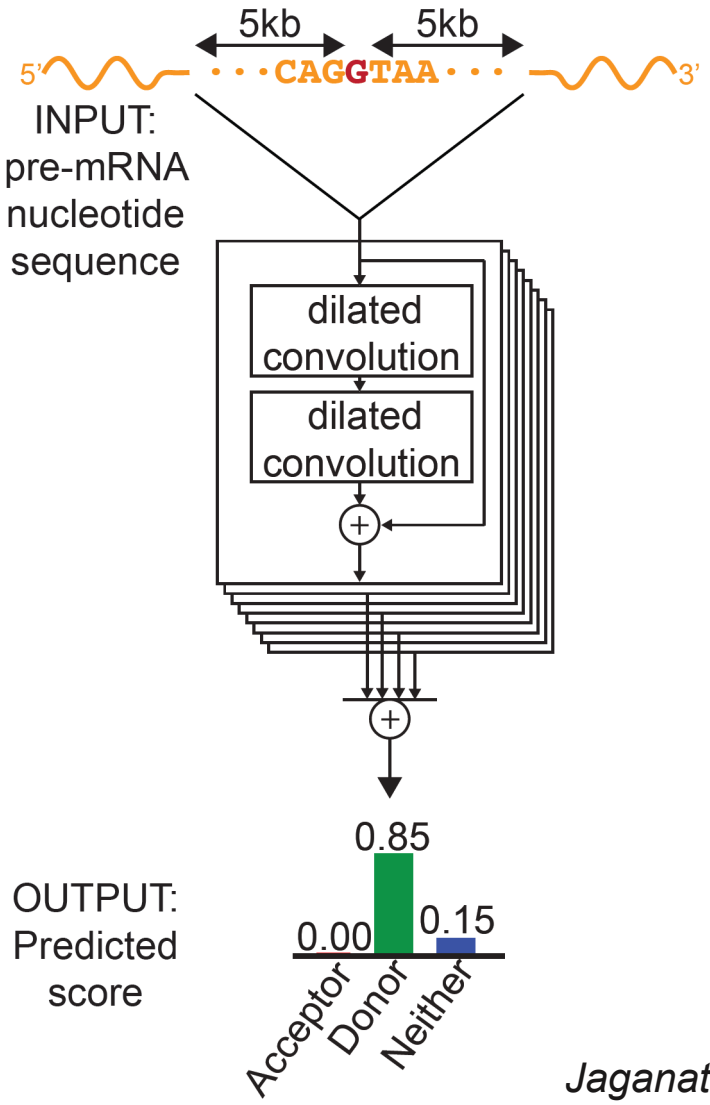
Deep learning/AI for genomics

TTGATGATCAGGTGGTGTCTCTGCCCGTCCTTCTTGAACCGGTATTTGAAGGTCTCCTCCCGGGT
CAGCTCCACCCCGTCCTTCAGCCTAGCCGGGTGGGTGGGTGGCAAGTGCTGTGGCCTCTTCTGGG
CAGATGCCCCCAACACCCATGCCCCGTGCTTCTGGAACTCACCATTTGACTTGCGCCCCCTCCTC
CGATACTTCACACTCAAACTCCACCCGCTGCCCCACCATCACCAGCTGGTCCTCCAAGGGGCGCG
TGATGAGCACAGGGGGCTCTGTCCAGGCAGGGTGAGCATGAGGGTTGGCTCCCCTGAGGCCATCT
CCTCCCCAGGTTCCCACATCCTCAGGTCCCAGGCCCACCTTTCACAAAGAGCTCCGTGCTACACT
TCTCGCCACCCACCACGCACTGGTAGGCTGCGTCGTCCGCCAATGAGCACTGGCTGATGGTCAGG
GTACGCTTGGCACCGATGGACTCAAAGATGTACCTGGGTGGGGGCTGCAGGGAAGTGGCAGGAAA
GCTGCGGACACCCCTCCGGGCCCAGTGCGCCCCATGATAATCCCTGTGCCCCCCACCCCAAGCCA
TCCAGAGGGGAACTTACTTGCTGTAGAACAGAAGGGGCCGTTGAAGTGTTCCCGACGGGAGGAAG
TGAGCCCGAGACAAAAGGAGAGAGAGAGAGGGACCGGCAGGAGCAAAAGGATGGGAAATTAGGCC
CAGAGAGATGGGGCTGAGAGCCACACCGAGTCAGAGATACGCATGTGGAGAGGGGCAGGAGGCAA
GGCTATGGGGGTCCCCACCTCCACCCGAGCCCCCCTCCCCACCCCAGGCTGCACCTGCCGCTCAT
CTGGATCTCCTGGCCATTCTTGAGCCATTTGACCTCAGCGTCATGGTCAGCCAGTTCCACGGTCA
GCCGGATCTTGTGGCCTTTGCTCACCTGGTAGGCCGGCTCCAGCTTCTTCTGAAAGGCTGAGCAC

Splice variants in disease



Jaganathan et al, Cell 2019



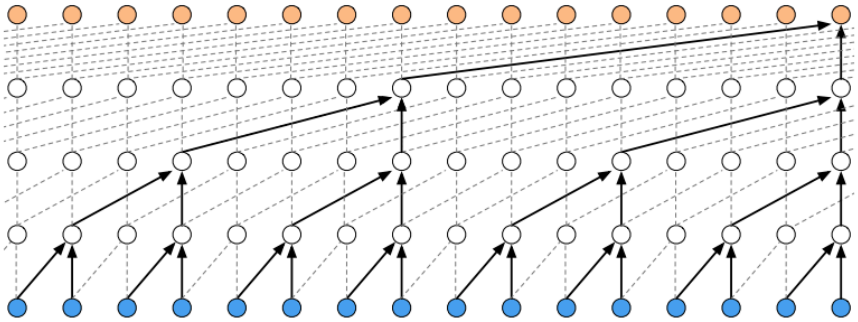
SpliceAI

- **Input**: 10K nucleotides
- **Labels**: 3-way classification, based on GENCODE annotations & RNA-seq
- **Architecture**: 32-layer convolutional neural network, 700K parameters
- Trained on half of chromosomes, withheld other half for testing, excluding paralogs

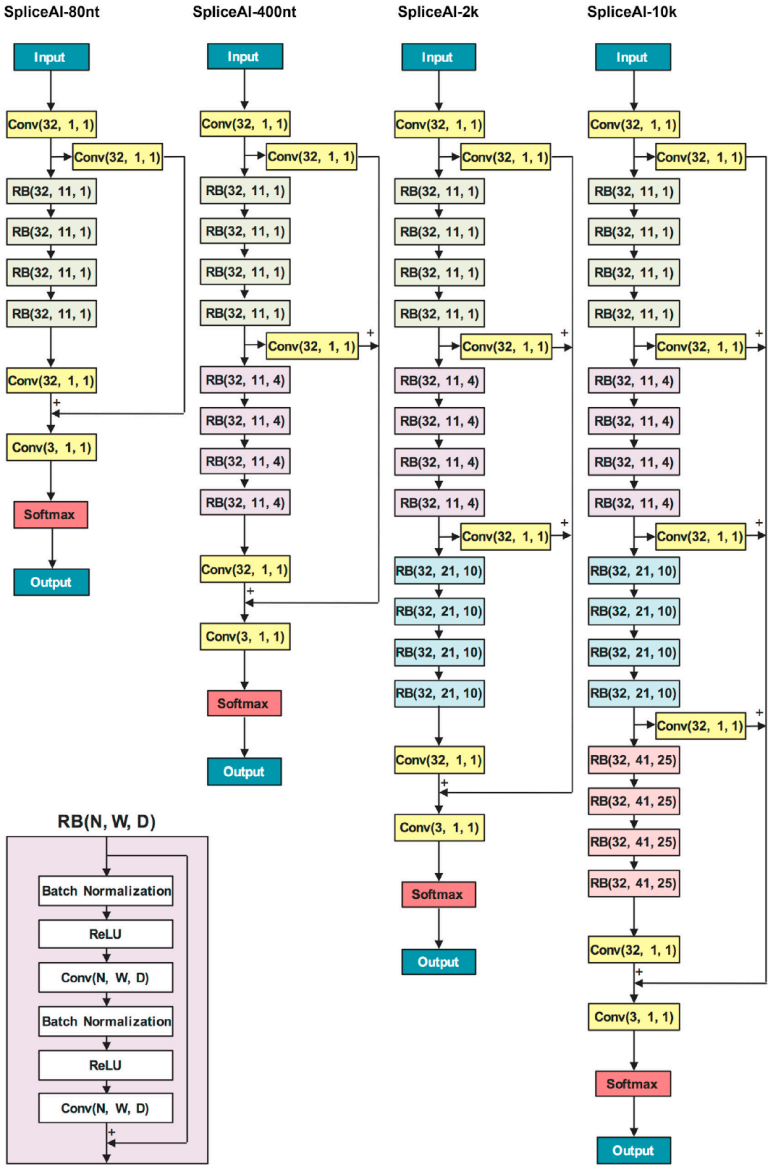
Jaganathan et al, Cell 2019

SpliceAI model

- **Sequence-to-sequence model using dilated convolutions + residual blocks**
- **Trained four models with context sequence size: 80nt, 400nt, 2000nt, and 10000nt**



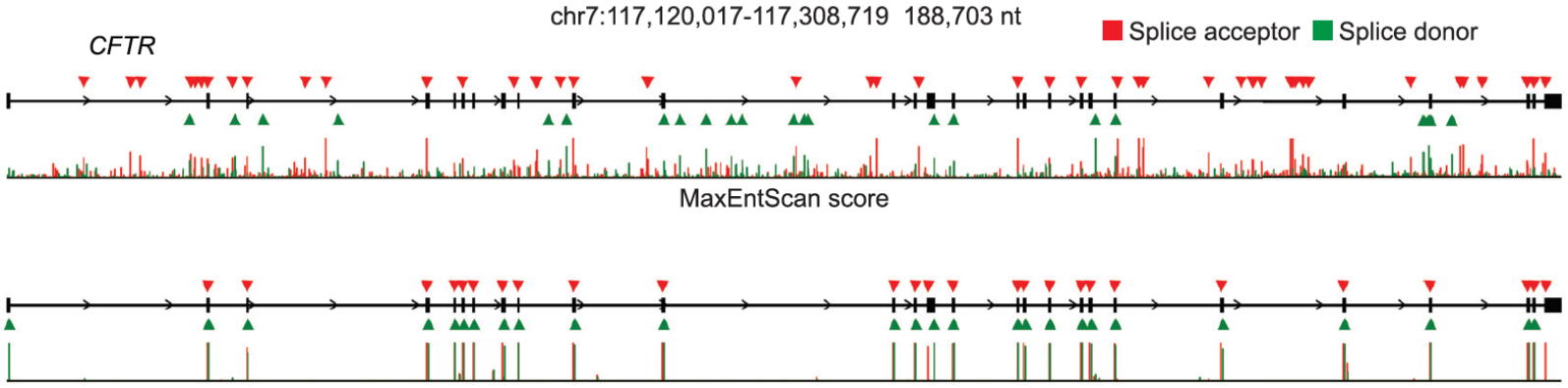
WaveNet, Van den Oord et al 2016

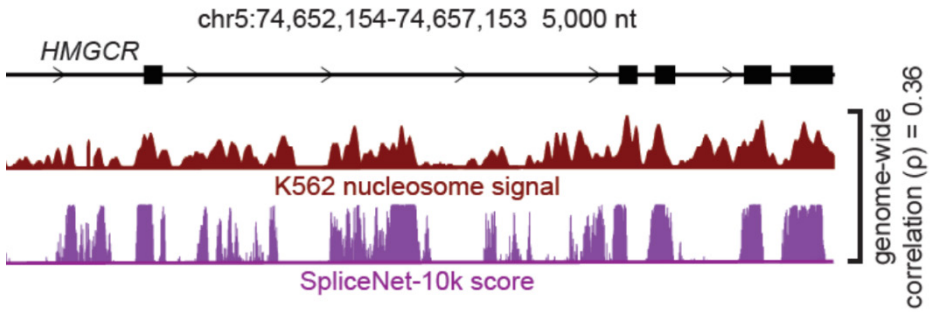


Decoding splicing with deep learning

Best previous algorithm *(**18%** accuracy)*

Illumina deep neural network *(**95%** accuracy)*





- Long range determinants up to 10kb are crucial for splicing specificity
- Intron / exon length, nucleosome positioning play major roles

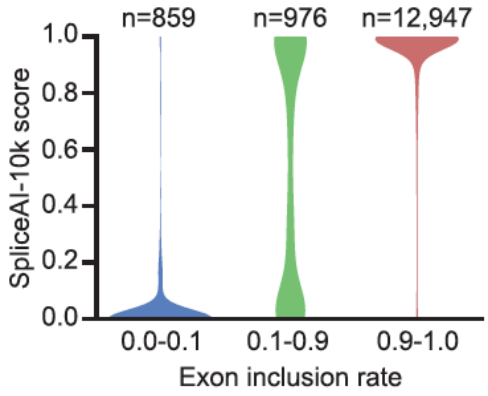
Jaganathan et al, Cell 2019

SpliceAI performance

Test accuracy

	Top-*k* accuracy
SpliceAI-80nt	0.57
SpliceAI-400nt	0.90
SpliceAI-2k	0.93
SpliceAI-10k	**0.95**
GeneSplicer	0.30
MaxEntScan	0.22
NNSplice	0.22

Accurate prediction of splicing requires very long input sequence



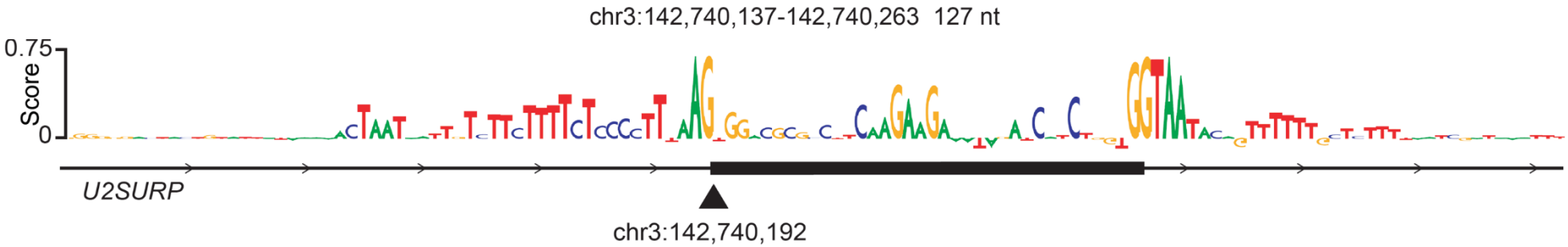
lincRNA accuracy

	Top-*k* accuracy
SpliceAI-80nt	0.51
SpliceAI-400nt	0.70
SpliceAI-2k	0.82
SpliceAI-10k	**0.84**
GeneSplicer	0.33
MaxEntScan	0.33
NNSplice	0.32

Accurate prediction of noncoding transcripts

Jaganathan et al, Cell 2019

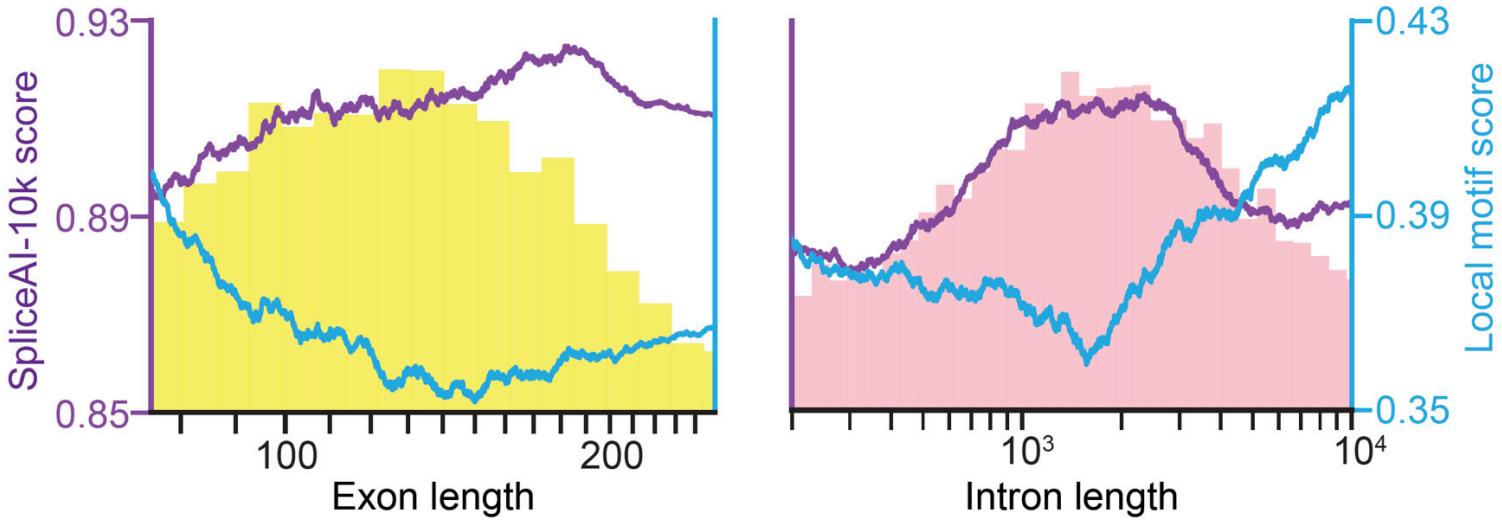
What features does SpliceAI use?



- **Impact of in-silico mutating each nucleotide around a splice acceptor**

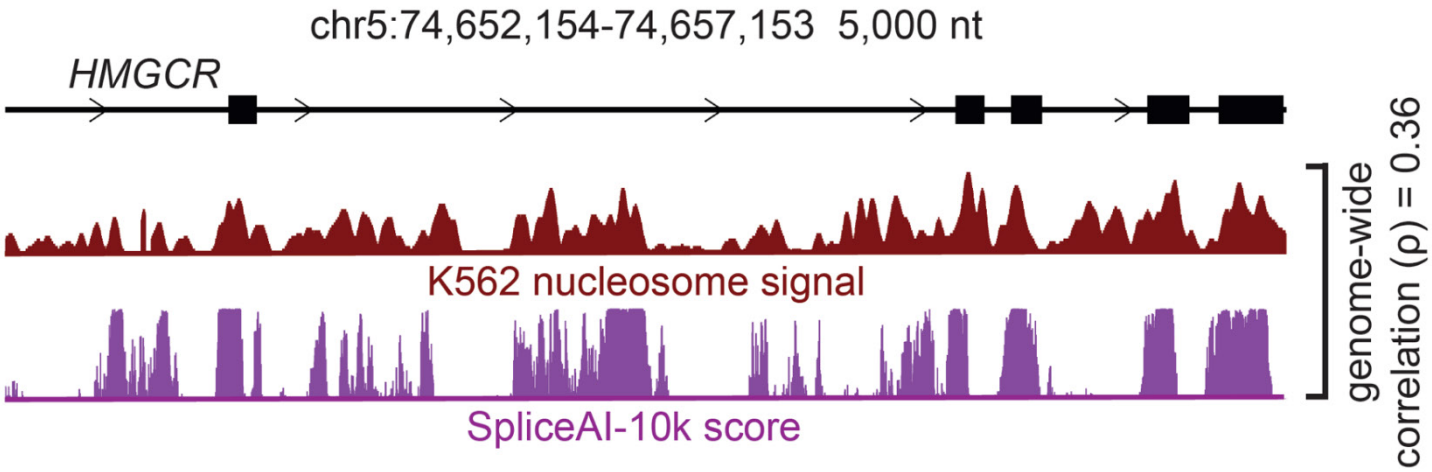
Jaganathan et al, Cell 2019

What features does SpliceAI use?



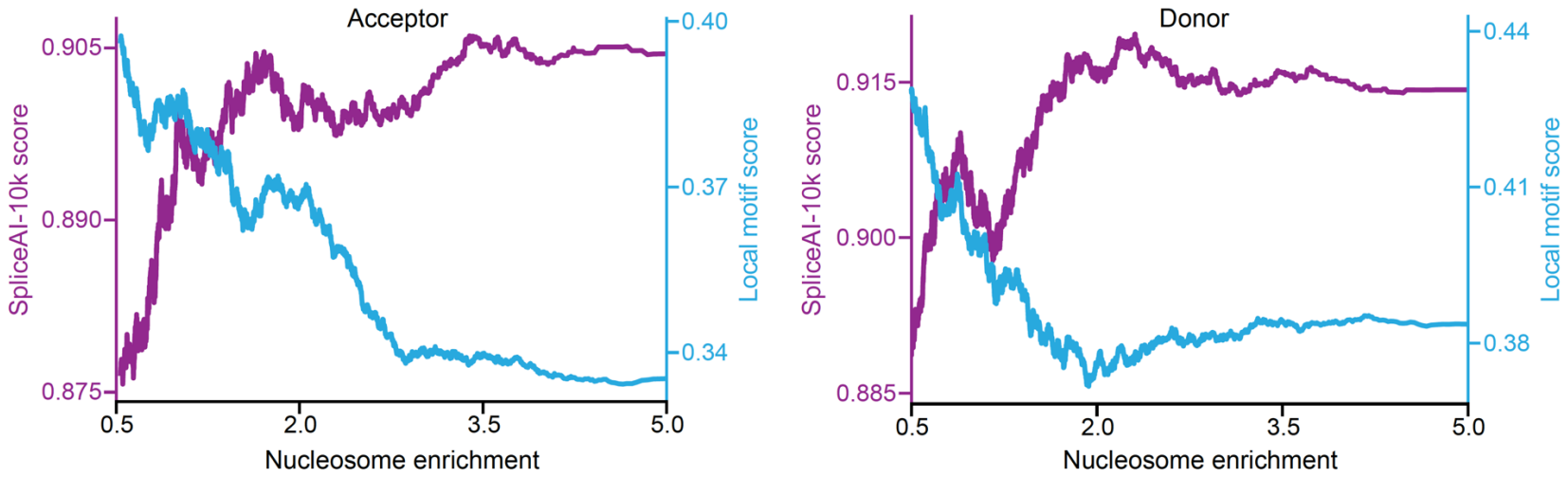
Exon/intron lengths confer additional specificity to splice sites

What features does SpliceAI use?



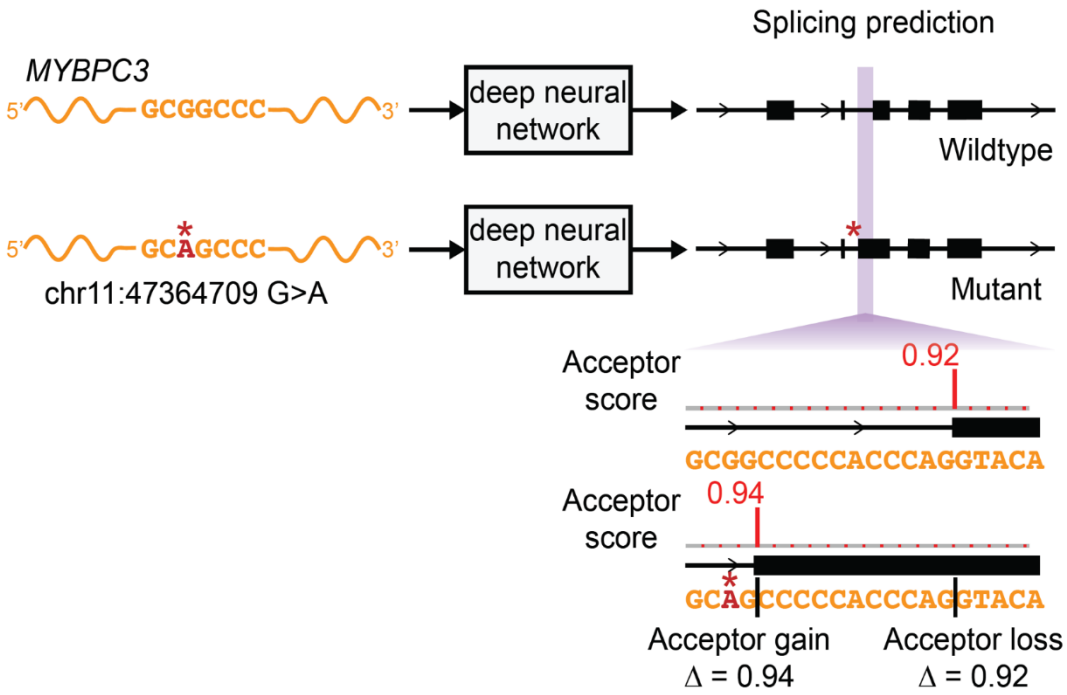
Nucleosome positioning is a specificity determinant for splicing

What features does SpliceAI use?



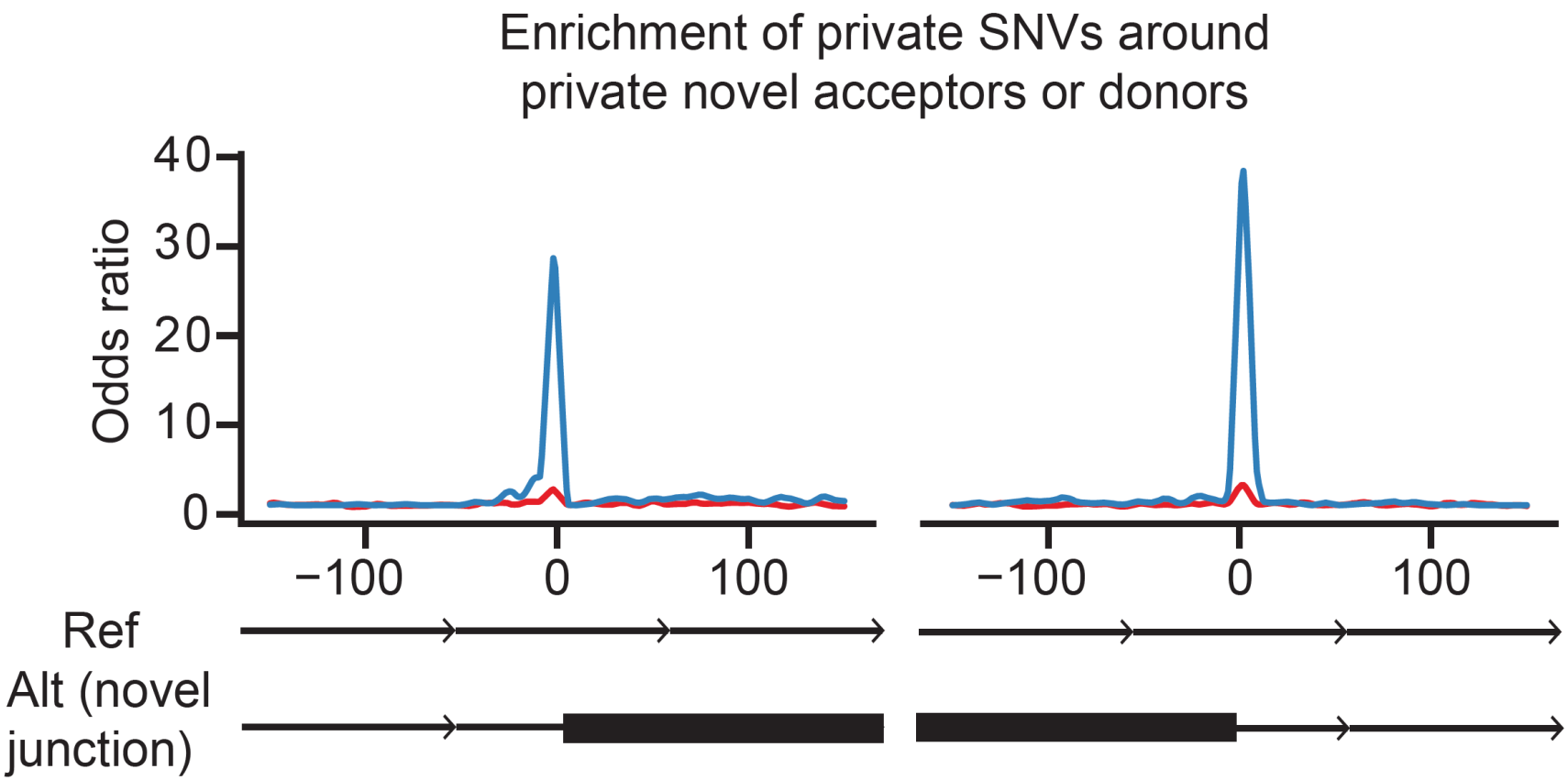
Nucleosome positioning is a specificity determinant for splicing

Scoring variants with SpliceAI

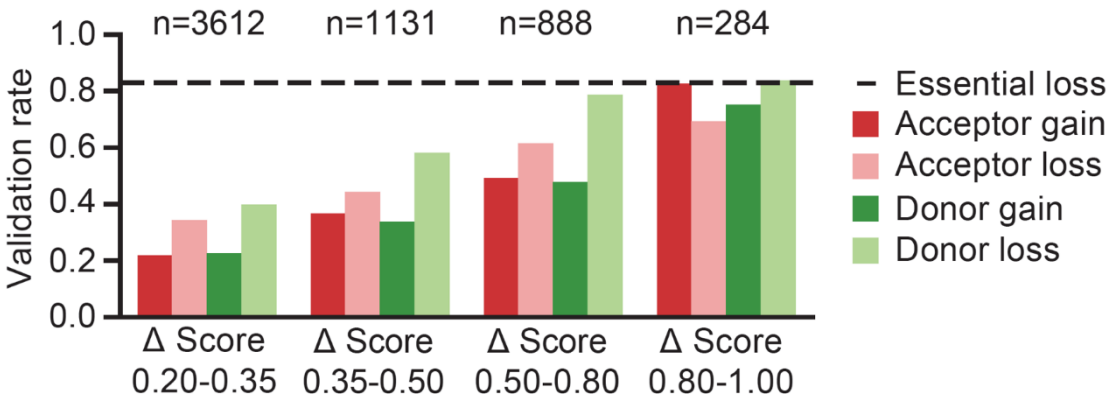


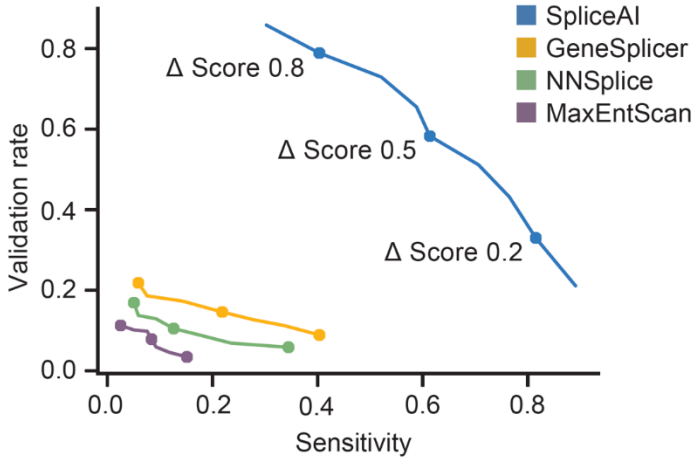
4 scores: acceptor gain, acceptor loss, donor gain, and donor loss

Validation of predicted cryptic splice variants in GTEx

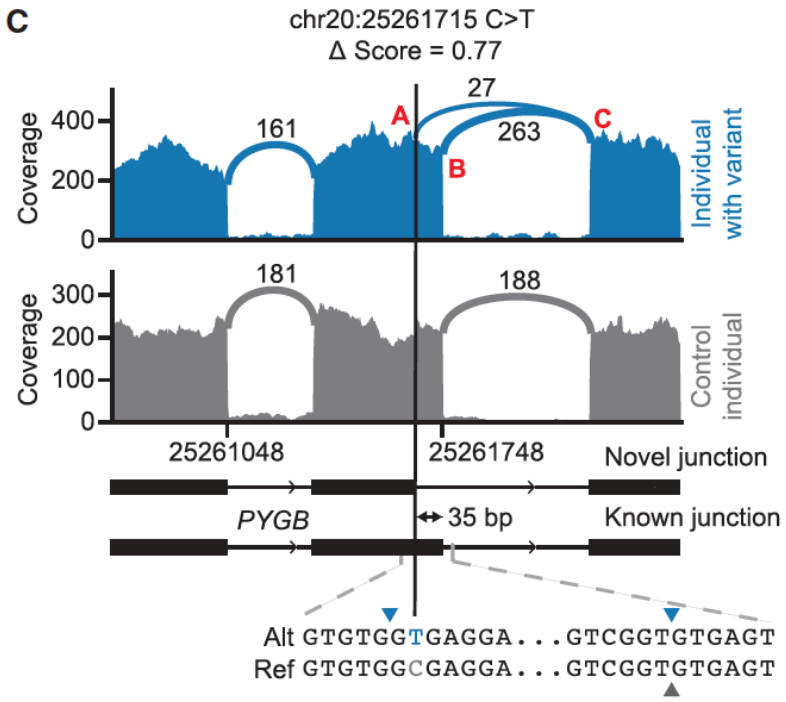


Validation of predicted cryptic splice variants in GTEx



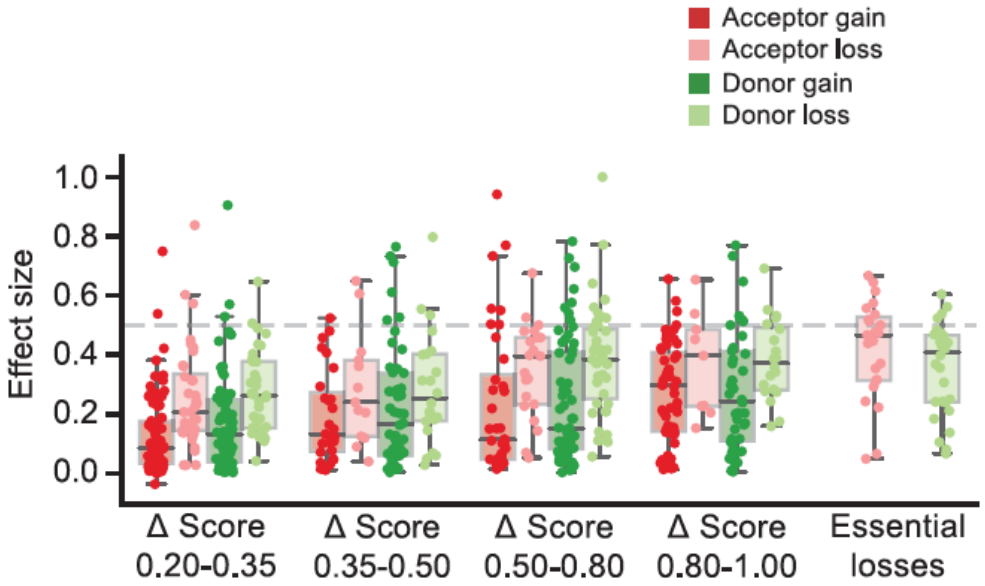


Most cryptic splice-altering variants have partial effects



Relative usage of novel junction

$$\left(\frac{AC}{AC+BC}\right)_{mut} - \left(\frac{AC}{AC+BC}\right)_{ctrl} = \frac{27}{27+263} - 0 = 0.09$$



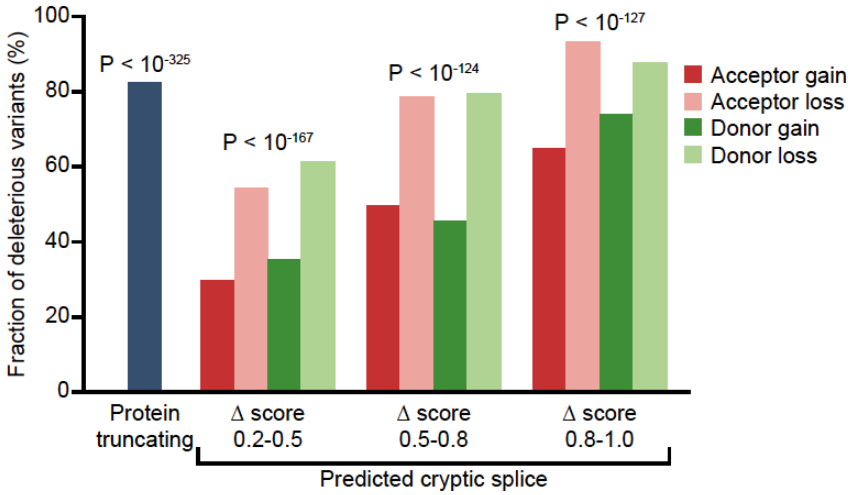
Effect size generally under-estimated due to noise, NMD, unaccounted for effects.

Cryptic splice variants are strongly deleterious

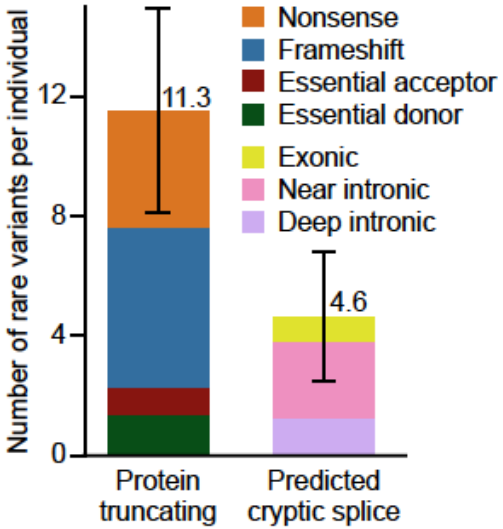
ExAC near exonic variants

	Singleton	Common (AF ≥ 0.1%)
SNVs with Δ score ≥ 0.8	10,369	212
SNVs with Δ score < 0.1	1,687,004	158,177

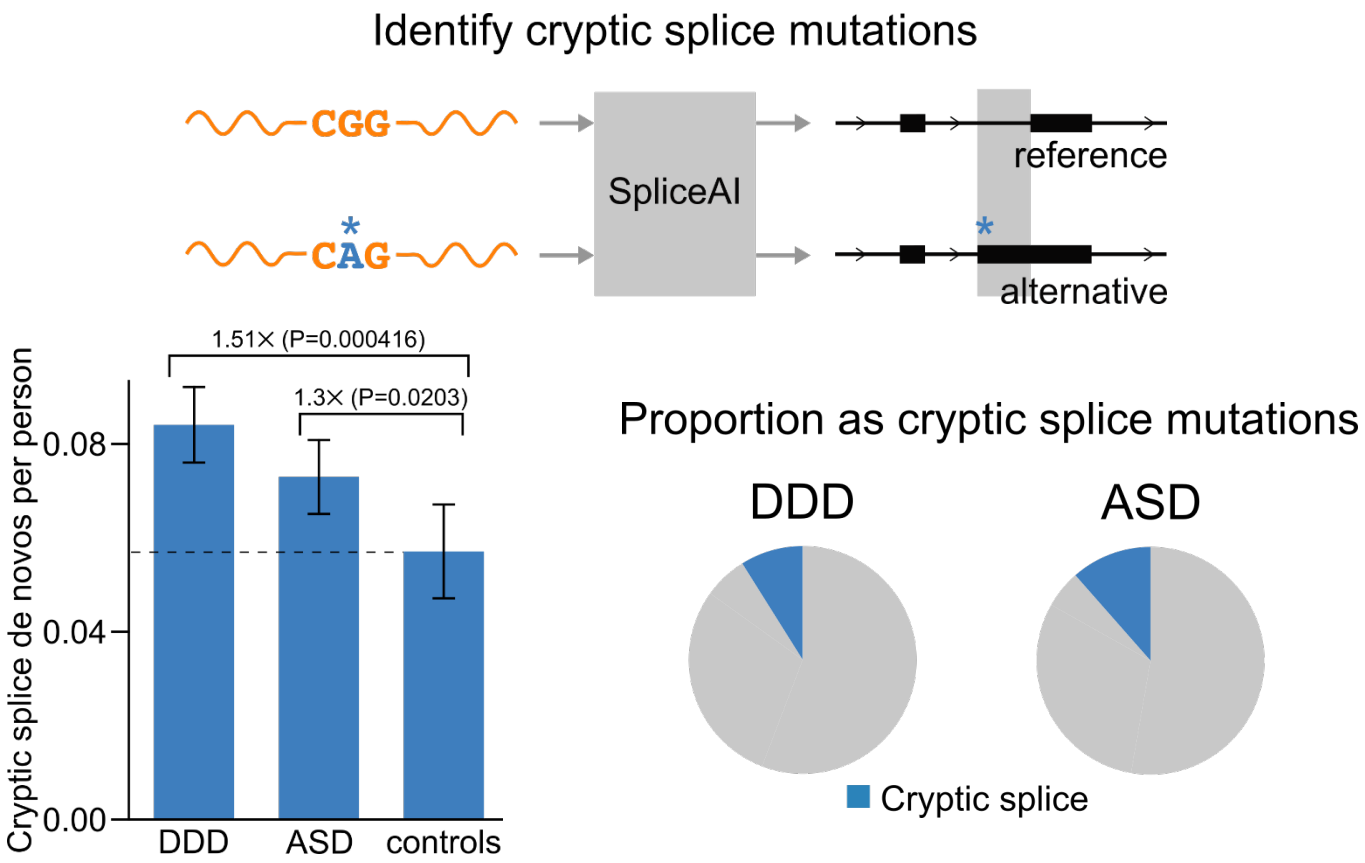
Odds Ratio (OR) = 4.58 ($P < 10^{-127}$)



Fraction deleterious variants:
(OR – 1) / OR



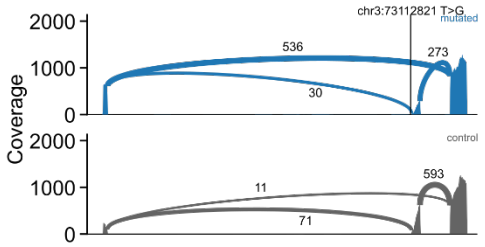
SpliceAI performance in rare disease cohorts



SpliceAI examples - exon skipping

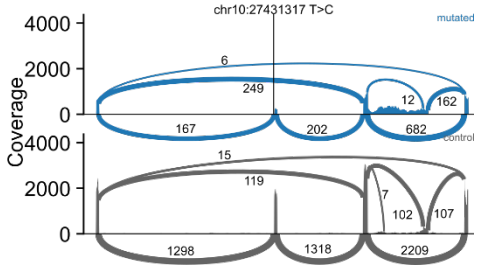
chr3:73112821 T>G splice region of *PPP4R2* TCTTATAGGA (exon)

	score	distance
acceptor gain	0.34	1
acceptor loss	0.97	3
donor gain	0.00	209
donor loss	0.45	77



chr10:27431317 T>C synonymous in *YMEL1* ATACCTGGCC (exon)

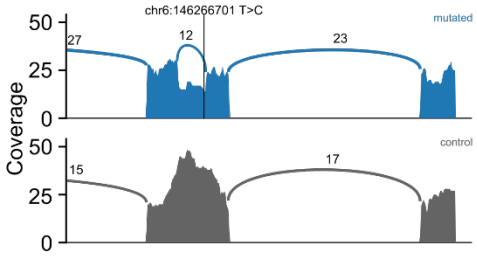
	score	distance
acceptor gain	0.00	-1
acceptor loss	0.47	97
donor gain	0.00	-24
donor loss	0.71	-1



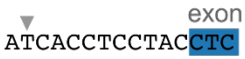
SpliceAI examples - novel junctions

chr6:146266701 T>C SHPRH p.Tyr465Cys (73 bp from exon end)

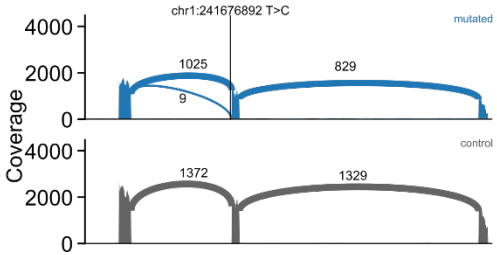
	score	distance
acceptor gain	0.46	-87
acceptor loss	0.02	-25
donor gain	0.99	1
donor loss	0.38	-3



chr1:241676892 T>C intron variant in *FH*



	score	distance
acceptor gain	0.00	121
acceptor loss	0.00	105
donor gain	0.99	1
donor loss	0.07	11



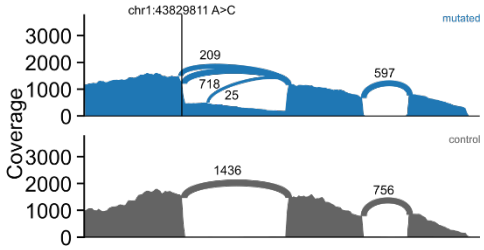
illumina®

SpliceAI examples - intron retained

chr1:43829811 A>C splice region of *MED13*

exon
ATCTAGGAT

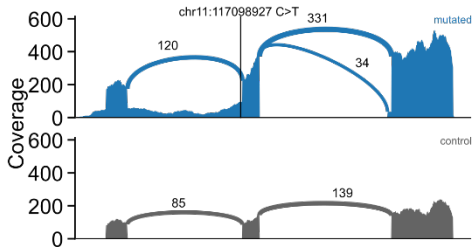
	score	distance
acceptor gain	0.33	40
acceptor loss	0.65	-3
donor gain	0.01	-198
donor loss	0.16	184



chr11:117098927 C>T splice region of *PCSK7*

exon
CCACTCACAT

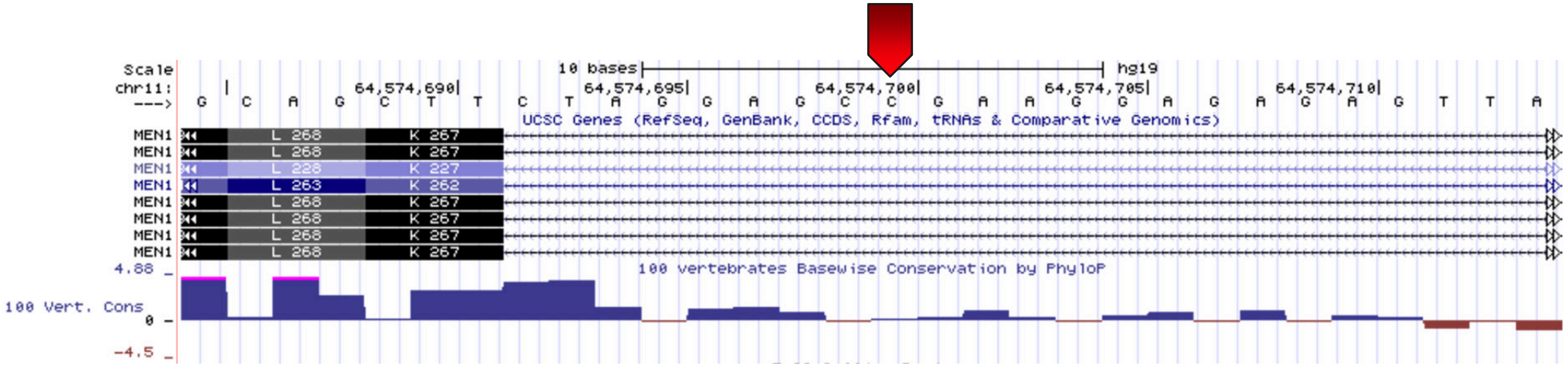
	score	distance
acceptor gain	0.00	-144
acceptor loss	0.00	139
donor gain	0.55	100
donor loss	0.39	5

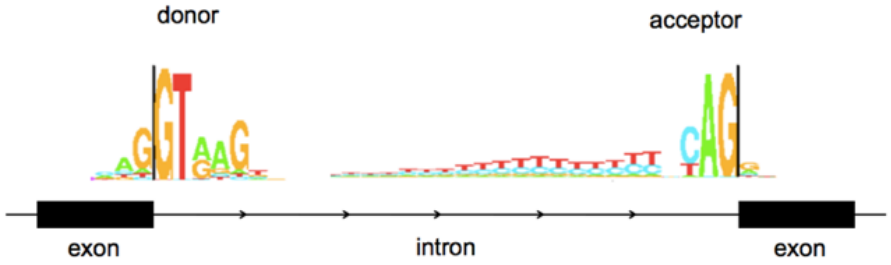


illumina®

MEN1 - Multiple Endocrine Tumors

GEL case with noncoding mutation identified with deep learning





chr11:64574700 C > T variant creates new splice acceptor upstream of canonical acceptor, frameshifting the **MEN1** tumor suppressor, and causing multiple tumors.

SpliceAI – predicting pathogenic noncoding splice variants

- Table of precomputed SNVs available online
  - basespace.illumina.com/s/5u6ThOblecrh
  - scores for SNVs genome-wide
  - table of sites with high spliceAI scores
- Code available online
  - github.com/illumina/spliceAI
- Install via command line
  - `pip install spliceai`
- Run on command line
  - spliceai -I sample.vcf -A grch37 -R genome.fa

Acknowledgements

Kyle Farh, MD, PhD
Principal Investigator
Illumina AI Lab

Internships, postdocs, research positions? kfarh@illumina.com

Illumina Artificial Intelligence Laboratory


Kishore Jaganathan


Sofia Kyriazopoulou Panagiotopoulou


Jeremy McRae


Serafim Batzoglou

UCSF

Stephan Sanders
Siavash Fazel Darbandi
Juan Arbelaez
Grace B. Schwartz
Eric D. Chow

Stanford

Jonathan Pritchard
David Knowles
Yang I. Li

Genomics England

Mark Caulfield
Damien Smedley
Augusto Rendon

illumina®

Today: Predicting gene expression and splicing

0. Intro: 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 learning: Expr+DNA+miRNA RBMs in Cancer