Lecture 12: Predicting gene expression and splicing

Prof. Manolis Kellis

Slides credit: David Gifford, et al
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
**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

<table>
<thead>
<tr>
<th>Condition 1</th>
<th>Condition 2</th>
<th>Condition 3</th>
<th>…</th>
</tr>
</thead>
</table>

- Study resulting matrix

Each experiment measures expression of thousands of ‘spots’, typically genes

<table>
<thead>
<tr>
<th>m genes</th>
<th>n experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Expression profile of a gene

Gene similarity questions

Experiment similarity questions
Clustering vs. Classification

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

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

Conditions → Genes

Independent validation of groups that emerge:
- Chronic lymphocytic leukemia
- B-cell genes in blood cell lines
- Proliferation genes in transformed cell lines
- Lymph node genes in diffuse large B-cell lymphoma (DLBCL)

Known classes:
- Pan B cell
- Germinal Centre B cell
- T cell
- Activated B cell
- Proliferation
- Lymph node

Alizadeh, Nature 2000
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

\[ A_k = U \ \text{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)V^T \]

set smallest r-k singular values to zero

\[ A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \]

column notation: sum of rank 1 matrices

• Error:

\[ \min_{X: \text{rank}(X)=k} \| A - X \|_F = \| A - A_k \|_F = \sigma_{k+1} \]
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

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. 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.)

- 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

Pre-sampling super-resolution (SR)

Post-sampling SR

Progressive up-sampling

Iterative up-and-down sampling
D-GEX - Deep Learning for up-scaling L1000 gene expression

- 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)

Parameters
- # of hidden layers: ![Hidden Layers](image)
- # of hidden units in each hidden layer: ![Hidden Units](image)
- Dropout rate: ![Dropout Rate](image)
- Momentum coefficient: 0.5
- Initial learning rate: ![Initial Learning Rate](image)
- Minimum learning rate: ![Minimum Learning Rate](image)
- Learning rate decay factor: 0.9
- Learning scale: 3.0
- Mini-batch size: 200
- Training epoch: 200
- Weights initial range: \[-\frac{\sqrt{6}}{\sqrt{n_i+n_o}}, \frac{\sqrt{6}}{\sqrt{n_i+n_o}}\]

Gene expression inference with deep learning

Yifei Chen, Yi Li, Rajiv Narayan, Aravind Subramanian, Xiaohui Xie


Published: 11 February 2016

Article history ▼
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. 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

Efficient Generation of Transcriptomic Profiles by Random Composite Measurements

Compressed sensing recovers expression profiles from random composite measurements (RCM)

Using RCMs and compressed sensing

1. Random composite measurements
2. Infer gene module activity
3. Estimate expression

Making RCMs in the lab

- Tagged molecular probes
- Pooled staining / hybridization
- Detection and quantification

Potential applications
- mRNA profiling, CyTOF, imaging mass cytometry, screening, metabolic profiling, chromatin profiling, etc.

Algorithm: Sparse Module Activity Factorization (SMAF)

1. 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

- 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
- Histone mark 1
- Outperforms previous methods
- Meanings features selected
- Convolution, pooling, drop-out, Multi-Layer-Perceptron (MLP) alternating lin/non-linear
AttentiveChrome: Selectively attend to specific marks/positions

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

  - Attention focuses on specific positions for specific marks

- Consistent improvement over DeepChrome

### Baselines vs. AttentiveChrome Variations

<table>
<thead>
<tr>
<th>Model</th>
<th>DeepChrome (CNN) [29]</th>
<th>LSTM</th>
<th>CNN-Attn</th>
<th>CNN-(\alpha, \beta)</th>
<th>LSTMAtt + (\alpha, \beta)</th>
<th>LSTMAtt + (\alpha)</th>
<th>LSTMAtt + (\alpha, \beta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>0.8008</td>
<td>0.8052</td>
<td>0.7622</td>
<td>0.7936</td>
<td>0.8100</td>
<td>0.8133</td>
<td>0.8115</td>
</tr>
<tr>
<td>Median</td>
<td>0.8009</td>
<td>0.8036</td>
<td>0.7617</td>
<td>0.7914</td>
<td>0.8118</td>
<td>0.8143</td>
<td>0.8123</td>
</tr>
<tr>
<td>Max</td>
<td>0.9225</td>
<td>0.9185</td>
<td>0.8707</td>
<td>0.9059</td>
<td>0.9155</td>
<td>0.9218</td>
<td>0.9177</td>
</tr>
<tr>
<td>Min</td>
<td>0.6854</td>
<td>0.7073</td>
<td>0.6469</td>
<td>0.7001</td>
<td>0.7250</td>
<td>0.7250</td>
<td>0.7215</td>
</tr>
<tr>
<td>Improvement over DeepChrome [29]</td>
<td>36</td>
<td>0</td>
<td>16</td>
<td>49</td>
<td>50</td>
<td>49</td>
<td>49</td>
</tr>
</tbody>
</table>
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

Alternatively spliced exon2

Feature set: known motifs, transcript structure in target exon and adjacent exons

RNA feature extraction

Splicing code

3-class softplus prediction model:
\[ q^{\text{inc}}, q^{\text{exc}}, q^{\text{nc}} \]

Exon inclusion:
\[ t^{\text{inc}}=1, t^{\text{exc}}=0, t^{\text{nc}}=0 \]

Exon exclusion:
\[ t^{\text{inc}}=0, t^{\text{exc}}=1, t^{\text{nc}}=0 \]

[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 Bern(1 − \(\alpha\))
- 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 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 $\psi_{\text{ref}}$ and alternative $\psi_{\text{alt}}$ sequences harboring one of the 658,420 common variants
- Calculate $\Delta \psi_t = \psi_{\text{ref}}^t - \psi_{\text{alt}}^r$ 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: $(1393 \text{ inputs} + 13 \text{ outputs}) \times 10 \text{ hidden units} = 13000 \text{ parameters}$
- 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 be 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

$z = b + \sum_i x_i w_i$

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)

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

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

\[
P(v) = \frac{\sum_h \exp(-E(v, h))}{\sum_v, h \exp(-E(v, h))}
\]

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

Botzmann 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

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

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

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

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

\[
\frac{\partial \log p(v|\theta)}{\partial w_{ij}} = \sum_{n=1}^N \left[ v_i \sum_h h_j p(h|v) - \sum_{v,h} v_i h_j p(v, h) \right]
\]

\[
= \mathbb{E}_{\text{data}}[v_i h_j] - \mathbb{E}_{\text{model}}[\hat{v}_i \hat{h}_j] \equiv <v_i h_j>_{\text{data}} - <\hat{v}_i \hat{h}_j>_{\text{model}}
\]

However: \( <v_i^\hat{}, h_j^\hat{}> \) model still too large to estimate.

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

[Hinton and Osindero, 2006]
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

1st column: Sample from generative model with each label clamped on.
2nd 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
RBMs for TCGA cancer integration: Expression, miRNAs, Methylation

Hierarchical model integrates:
- gene expression (GE)
- miRNA expression (ME)
- DNA methylation (DM)

Energy function combines multiple data types
Learned patient groups show different survival/drugs

- Capture independent variables from molecular data