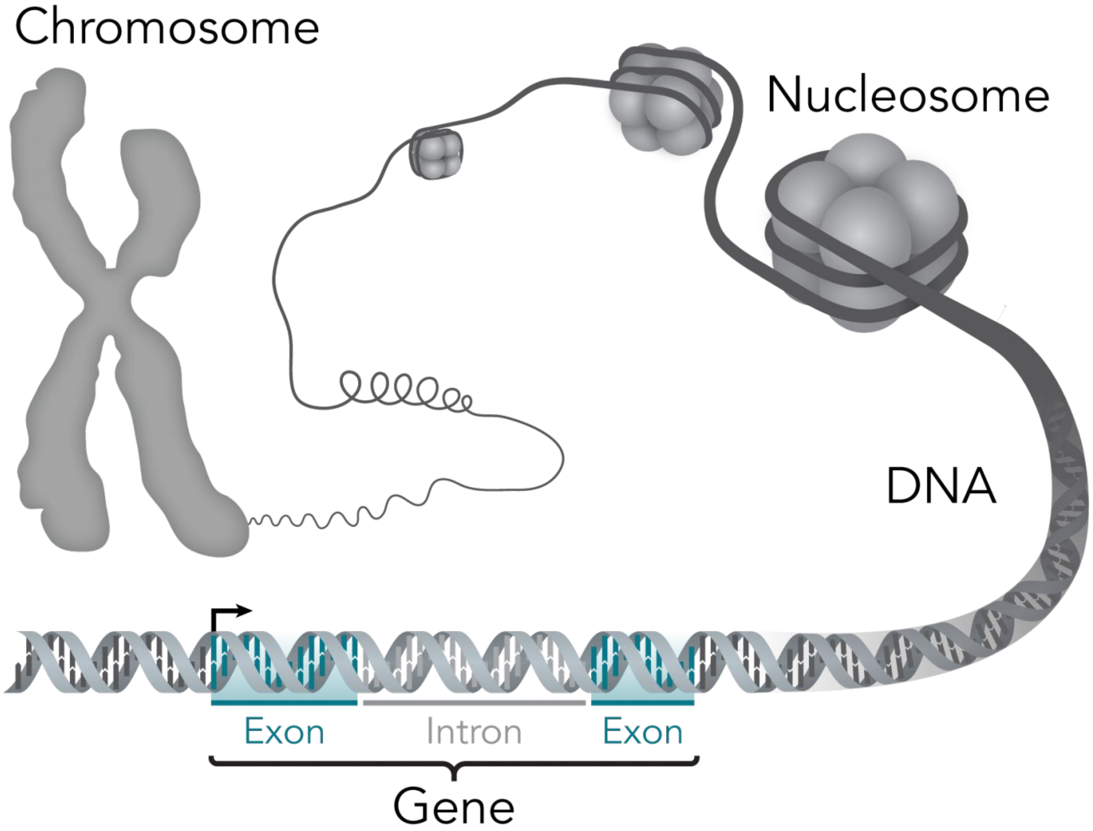
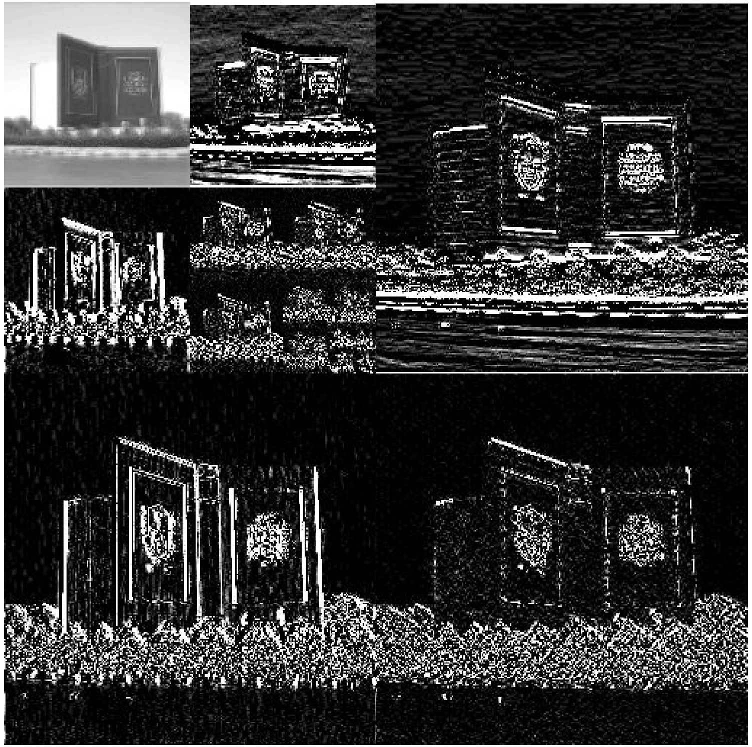
Learning structure in gene expression data using deep architectures, with an application to gene clustering

Nipun Agarwala, Oliver Bear Don’t Walk,
David Cohn, Yuki Inoue, Axel Sly

Gene Expression



Previous Work

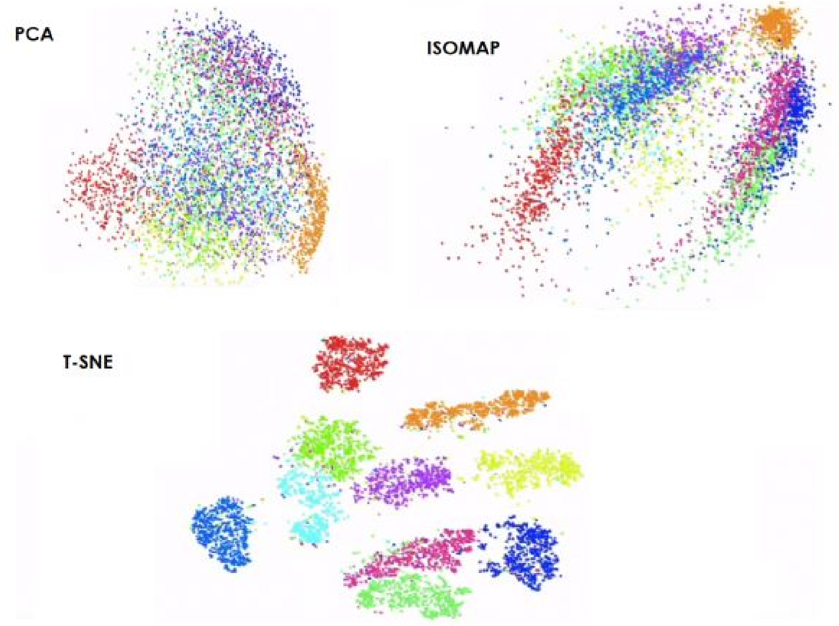


- Wavelet transformations
- Missing Value Imputation
  - Bayesian approach
  - Least Squares approach
- PCA
  - Did not improve cluster quality
- Autoencoder used to featurize breast cancer data

Motivation

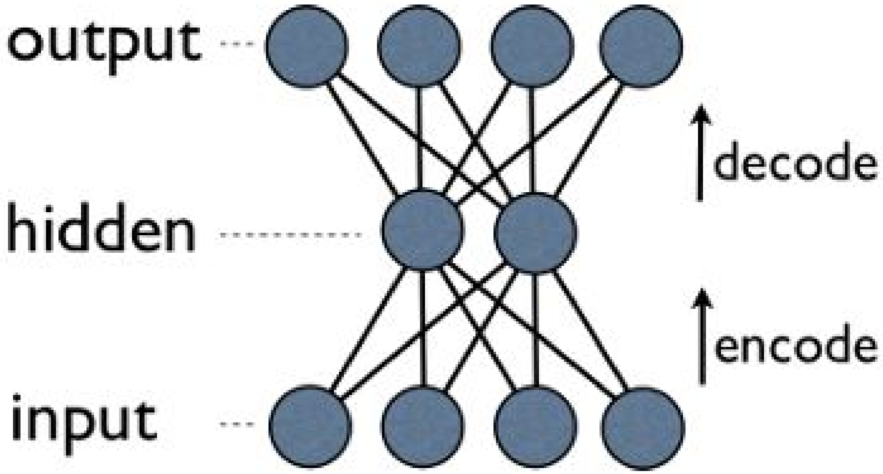
- **Objective**: Learn interesting patterns in the input distribution of gene expression profiles using deep networks with denoising autoencoders

- No microarray data denoising
- Learn and generalize
- Clustering

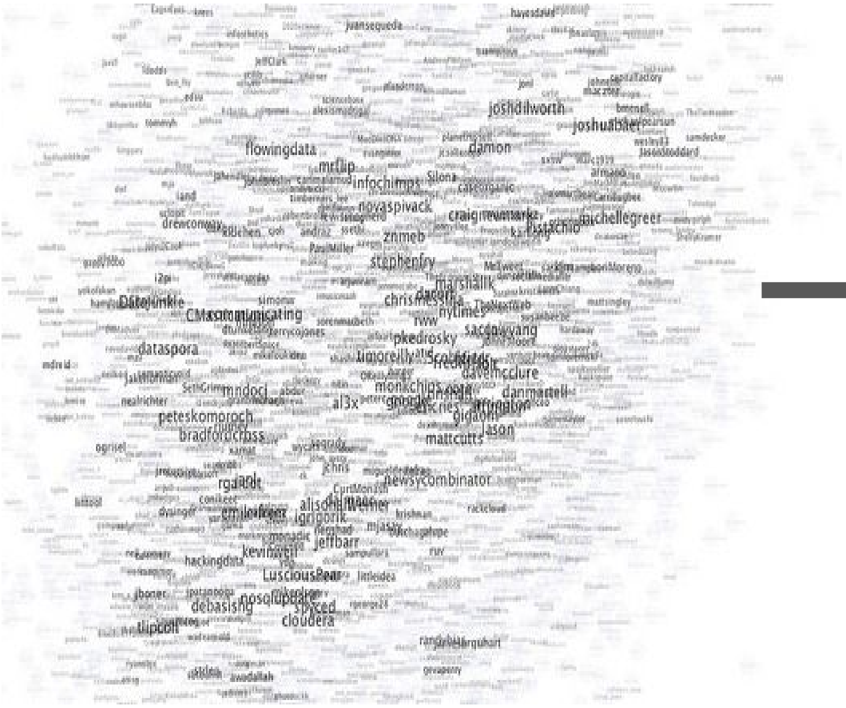


What is an Auto-encoder?

- Tries to learn $h_{W,b}(x) \simeq x$, so that the output $\hat{x}$ is close to x
- Typically, learn **lower dimension representation** of features i.e. hidden layers have lower dimension than input
- In some cases, hidden layers **can** have higher dimension, with an **additional sparse** (regularization) constraint, like KL divergence

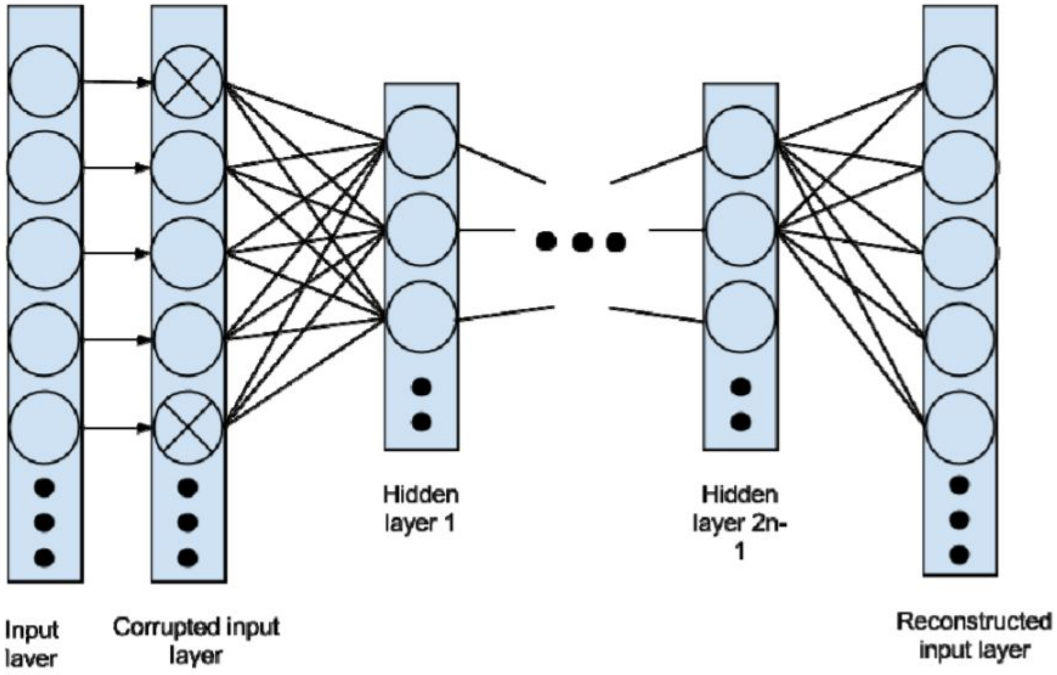


Why Autoencoders?



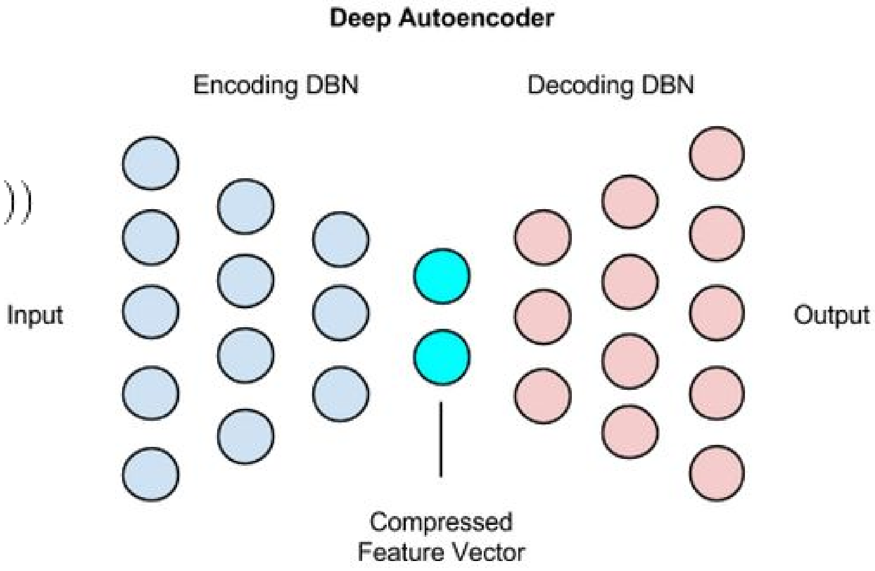
Additions to Vanilla Autoencoder

- Two additional changes are made to the vanilla autoencoders.
  - Stacking Autoencoders
  - Denoising Autoencoders



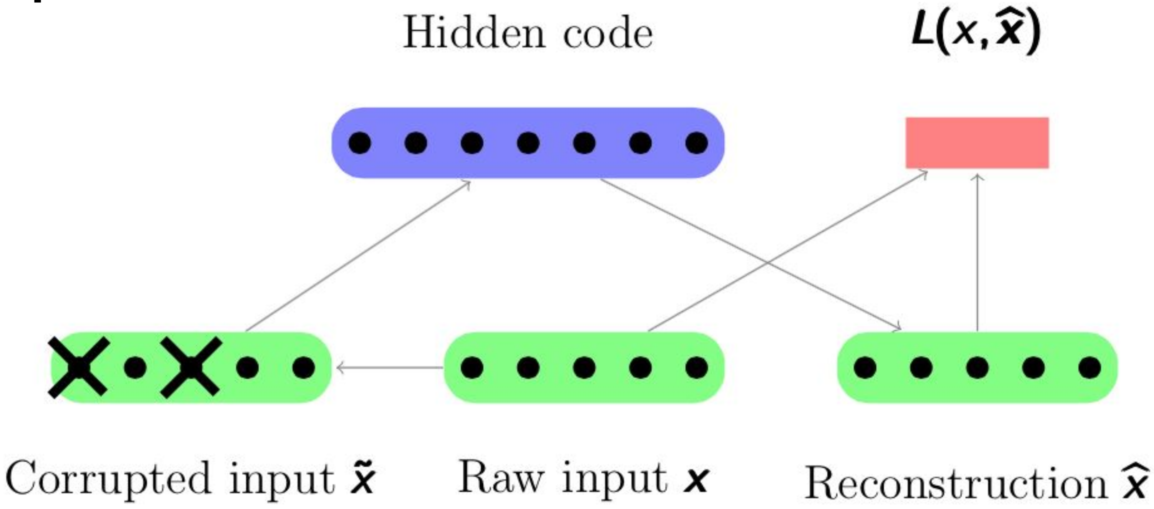
Deep Autoencoders

- Sigmoid layers present in between each layer, for non-linearities
- $\text{Transformation matrix} = f_1(W_1 * f_2(W_2 \ldots))$



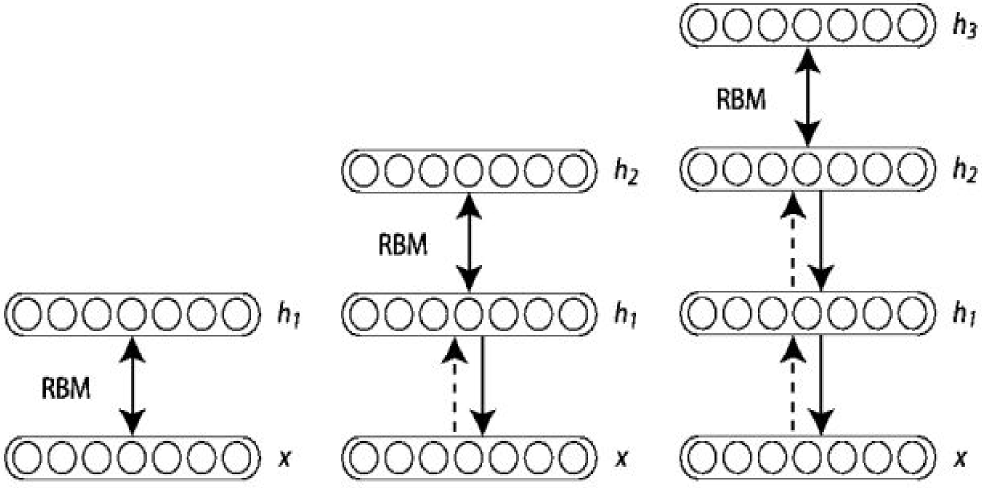
Denoising Autoencoders

- To **avoid learning an identity mapping**, two methods are usually used:
    - **Lower dimensionality** for the **hidden layer**
    - Train with **corrupt input**
- 2 Types of noise used:
    - Gaussian
    - Masking



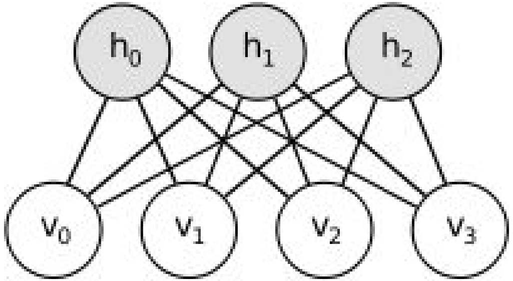
What were the improvements made?

- The authors decided to **stack multiple hidden layers** for the autoencoders
- The **training was done greedily**, training 1 layer at a time and minimizing the reconstruction error each time.
- By adding noise and stacking the layers, the encoders are able to generalize properties and learn interesting features.
- Similar to how **Deep Belief Network** is trained



Restricted Boltzmann Machine (RBM)

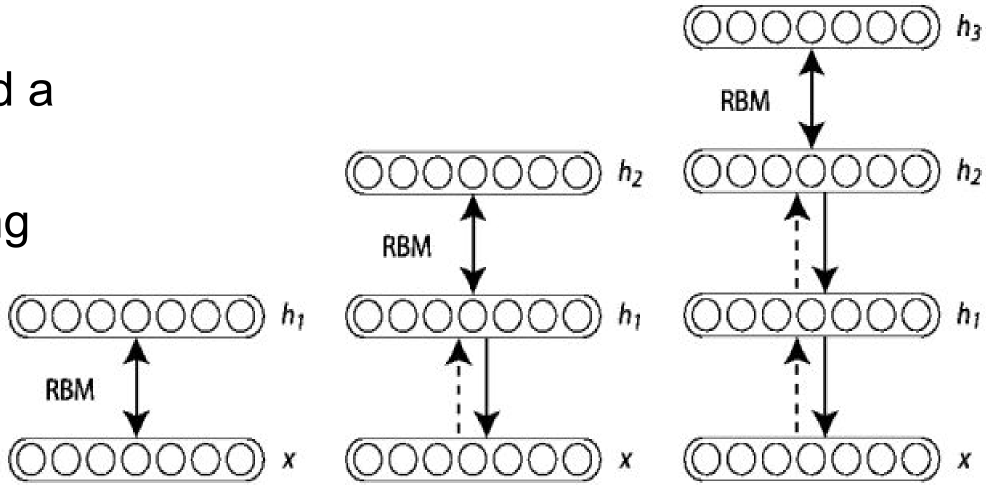
- Has a **visible layer** and a **hidden layer**, neither of which are connected to nodes from its own layer
- RBM is a **generative model**, tries to capture the probability distribution of the sample
- The **visible layer emulates the samples**
- As a result, the **hidden layer learns interesting features** of the samples
- The training is done to **minimize the "energy" equation**.



$$E = -\left(\sum_{i<j} w_{ij}\, s_i\, s_j + \sum_{i} \theta_i\, s_i\right)$$

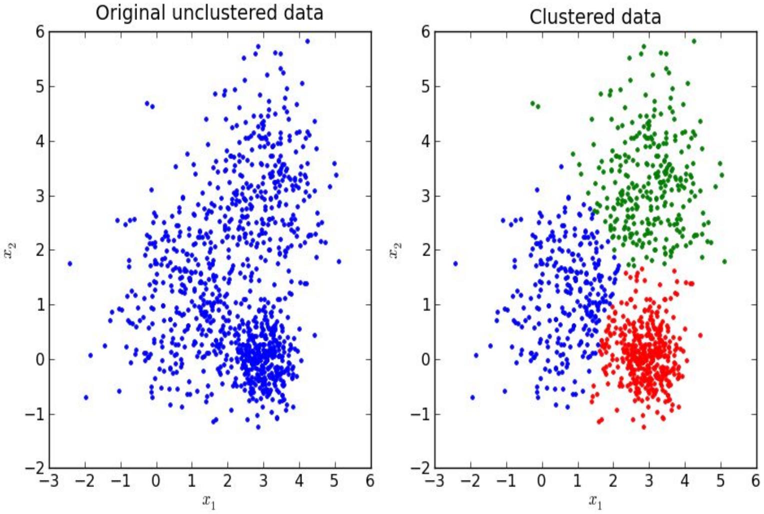
Deep Belief Network

- Another method to **preprocess the data**
- Created using stacking **multiple layers of RBM**
- Visible layers are taken as the inputs, and the Hidden layers are taken as the outputs to the next layer.
- **Training** is done layer by layer, and a global-parameter turning at the end
- Each layer of RBM learns interesting features of the first visible layer



Unsupervised Clustering: K-means

- **Cluster** the lower-dimensional points so that similar points are together. Centroids can be treated as “representative” points.
- Assign random (k < n) points as centroids. **Assign clusters based on the closest centroid** and then update centroid as mean of the vectors in that cluster.
- Helps find non-readily apparent patterns

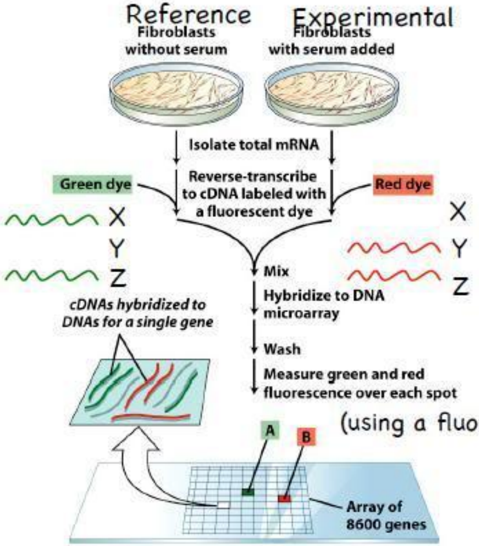


Yeast Cell Cycle Datasets

- Gupta et al. evaluated their autoencoder-based methodology on two yeast cell cycle data sets (Yeung et al. 2001)
- **Two data sets** derived from gene expression data for **6000 genes across 17 time points**; of 6000 total genes, **380 genes identifiably peak in expression** during a single phase of mitotic cell cycle
- Expression data **normalized**, with mean 0 and variance 1

DNA Microarrays

DNA microarray analysis of gene expression


Reference
Fibroblasts without serum
Experimental
Fibroblasts with serum added
Isolate total mRNA
Reverse-transcribe to cDNA labeled with a fluorescent dye
Green dye
Red dye
X
Y
Z
X
Y
Z
Mix
Hybridize to DNA microarray
Wash
Measure green and red fluorescence over each spot
cDNAs hybridized to DNAs for a single gene
A
B
Array of 8600 genes

(using a fluorescence scanner)

A If a spot is green, expression of that gene decreases in cells after serum addition

B If a spot is red, expression of that gene increases in cells after serum addition

Small region of a microarray representing expression

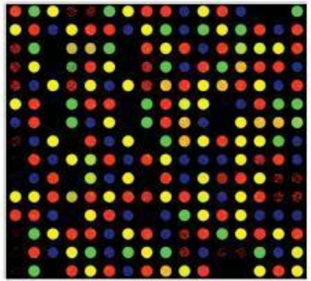
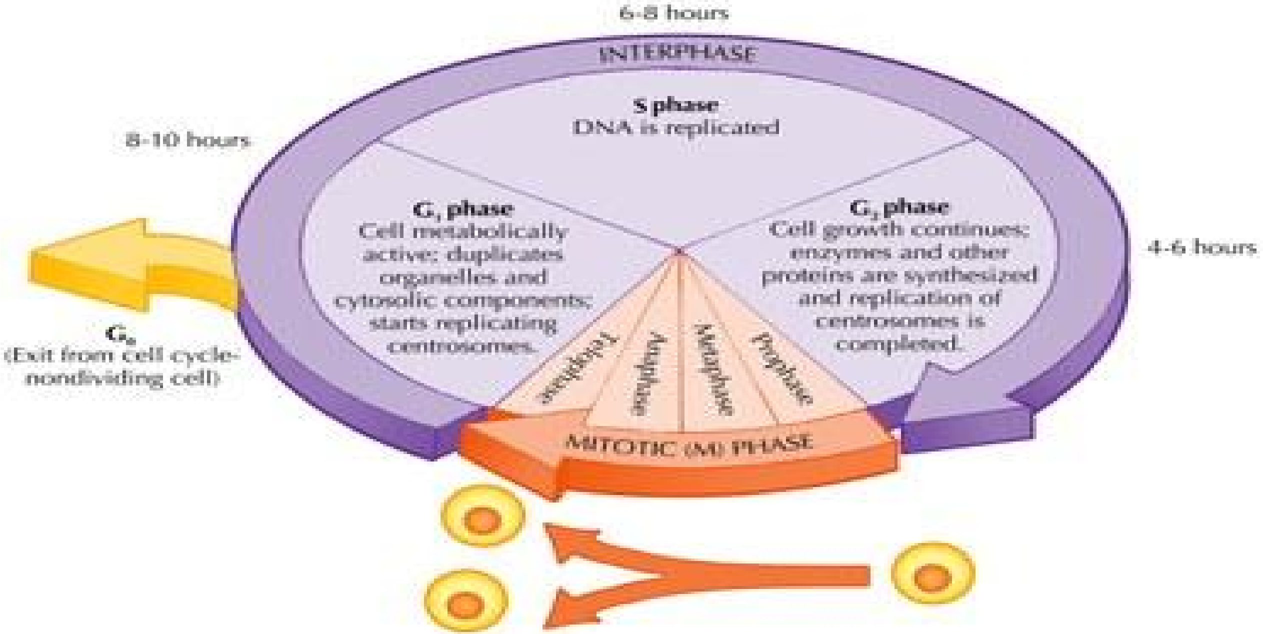


Figure 5-29b
Molecular Cell Biology, Sixth Edition
© 2008 W.H. Freeman and Company

Yeast Mitotic Cell Cycle


6-8 hours
INTERPHASE
S phase
DNA is replicated
8-10 hours
G_1 phase
Cell metabolically active; duplicates organelles and cytosolic components; starts replicating centrosomes.
G_2 phase
Cell growth continues; enzymes and other proteins are synthesized and replication of centrosomes is completed.
4-6 hours
G_0
(Exit from cell cycle-nondividing cell)
Telophase
Anaphase
Metaphase
Prophase
MITOTIC (M) PHASE

Hyper-parameter Tuning

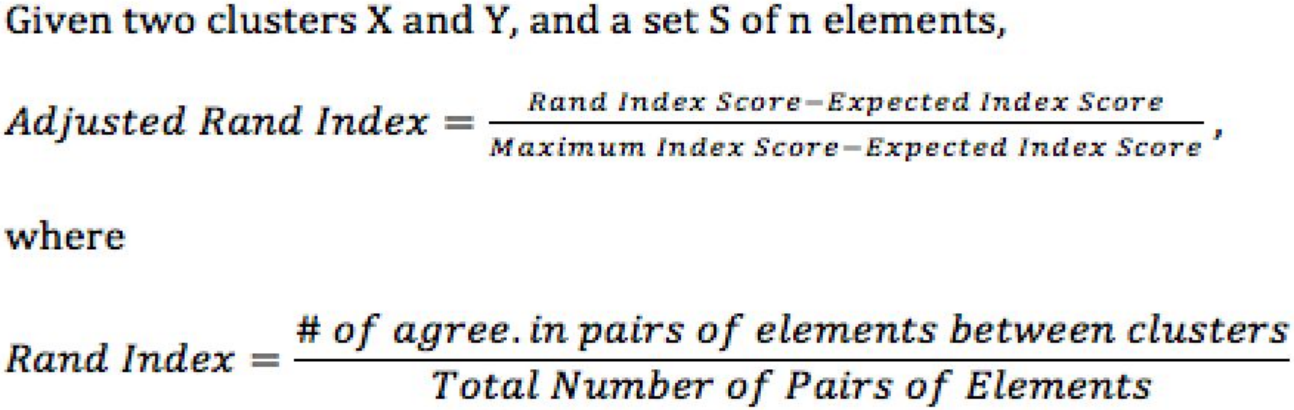
- **Hidden Layer Size** and **Corruption Level** contributed the most to the **variance** in results
- Performance of all hyper-parameter combinations considered in selection of parameter values
- Selected parameter values based on **“best results”** (validation set?)
- Number of hyper-parameters vs. dataset size

Parameters	Values Used
Batch Size	4,8,12
Number of Training Epochs	2000, 5000
Number of Hidden Nodes	Number between 4-17
Corruption Level	0, 0.05, 0.1, 0.15, 0.2
Learning Rate	0.05, 0.1

Evaluation Criteria

Adjusted Rand Index

- Quantitative measure of the **similarity in composition between two clusters**
- The “corrected-for-chance” version of the Rand Index
- Used to **assess performance of auto-encoder based clusters**, as compared to “gold-standard” cluster labels



Given two clusters X and Y, and a set S of n elements,

$$Adjusted\ Rand\ Index = \frac{Rand\ Index\ Score - Expected\ Index\ Score}{Maximum\ Index\ Score - Expected\ Index\ Score},$$

where

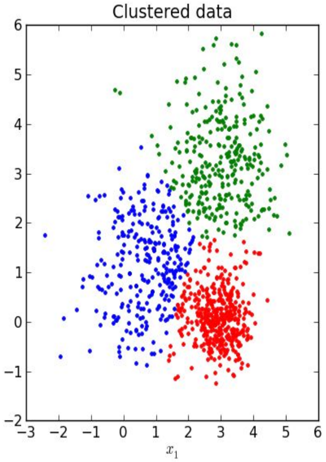
$$Rand\ Index = \frac{\#\ of\ agree.in\ pairs\ of\ elements\ between\ clusters}{Total\ Number\ of\ Pairs\ of\ Elements}$$

Clustering Algorithms and Implementation

Clustering

- K-means and spectral clustering

SciPy

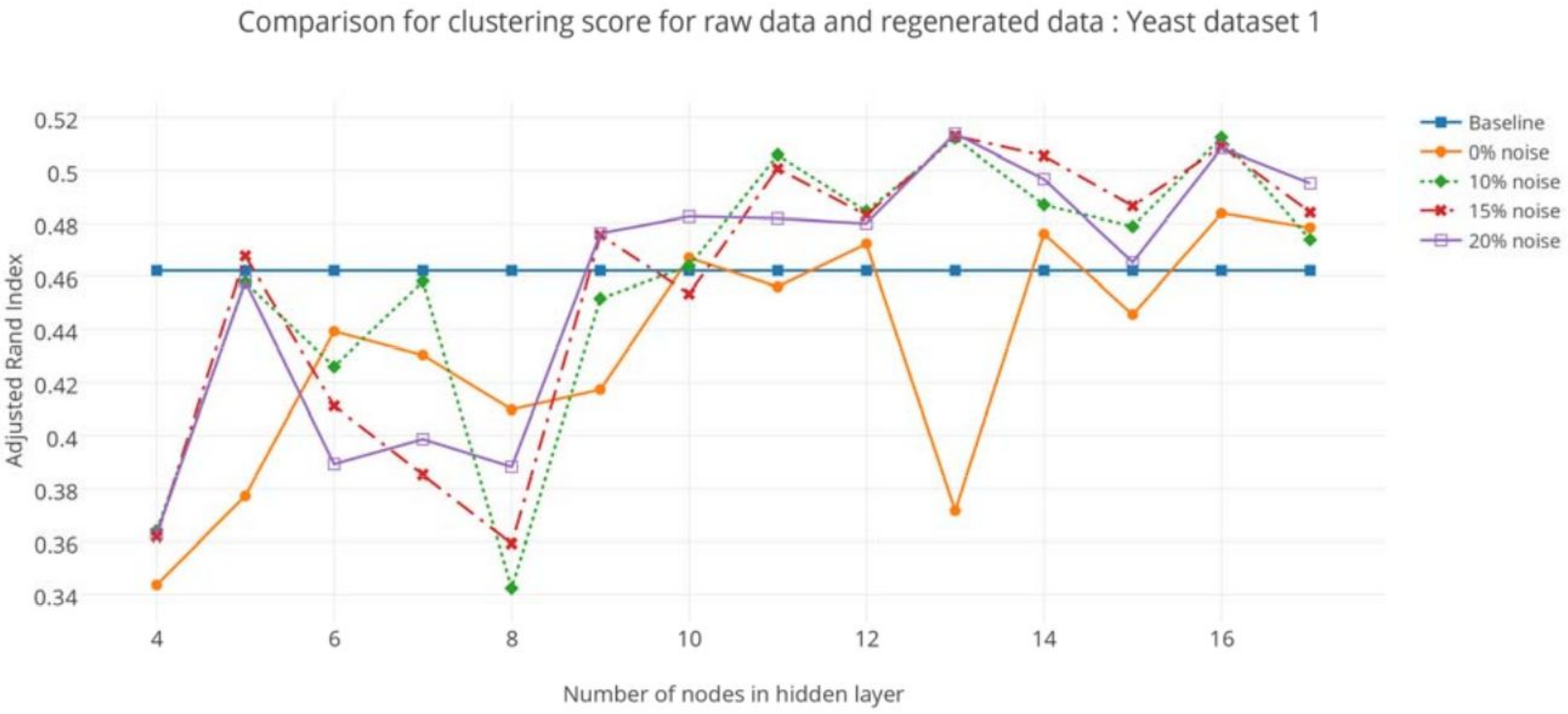


Implementation

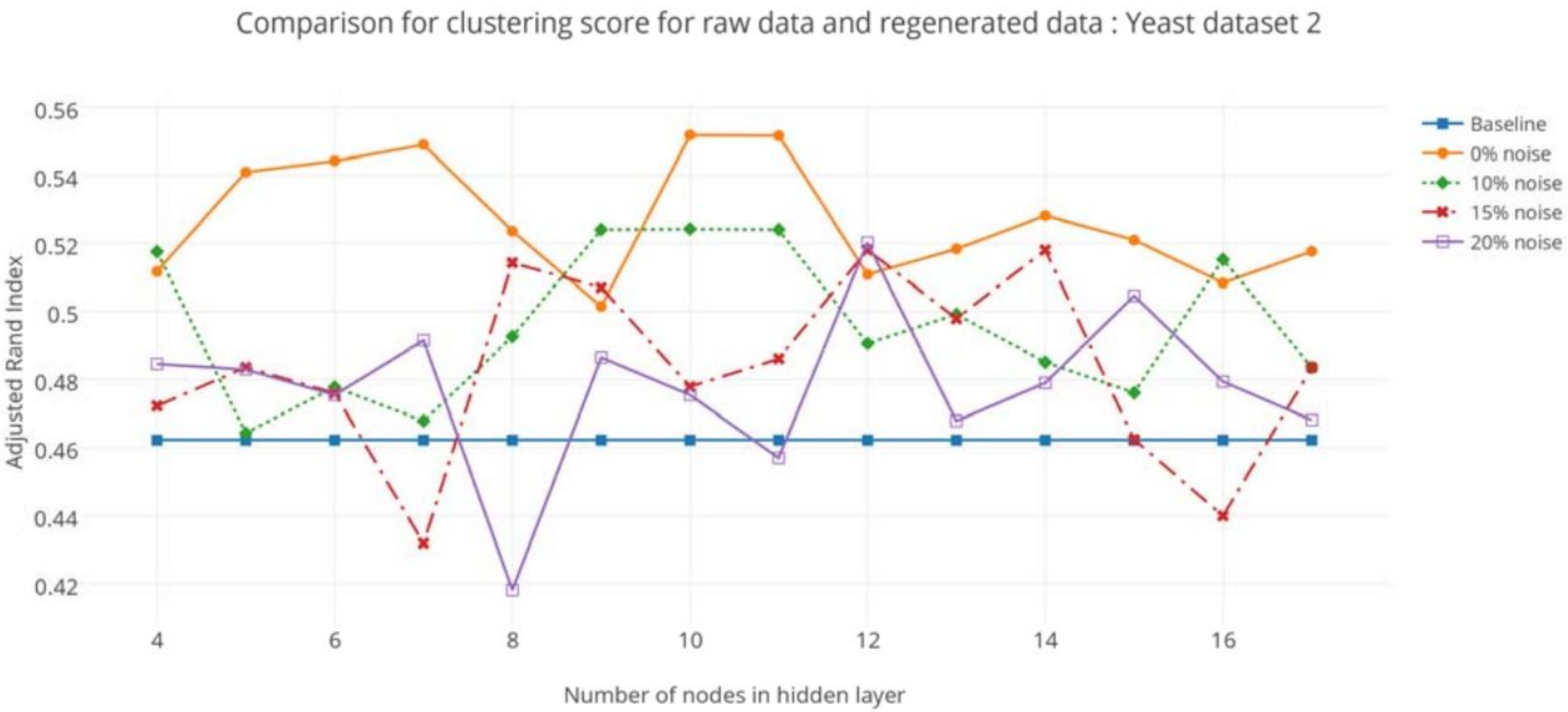
- Single hidden layer used, as opposed to three hidden layers, based largely on computation time, but also superior performance

theano

Experimental Results



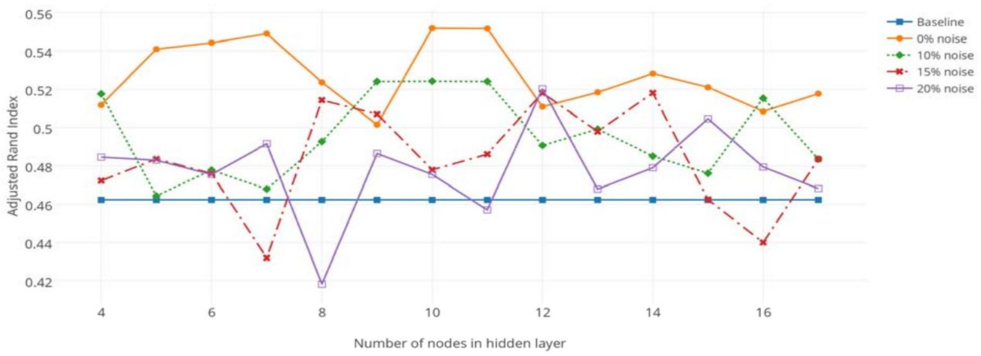
Experimental Results



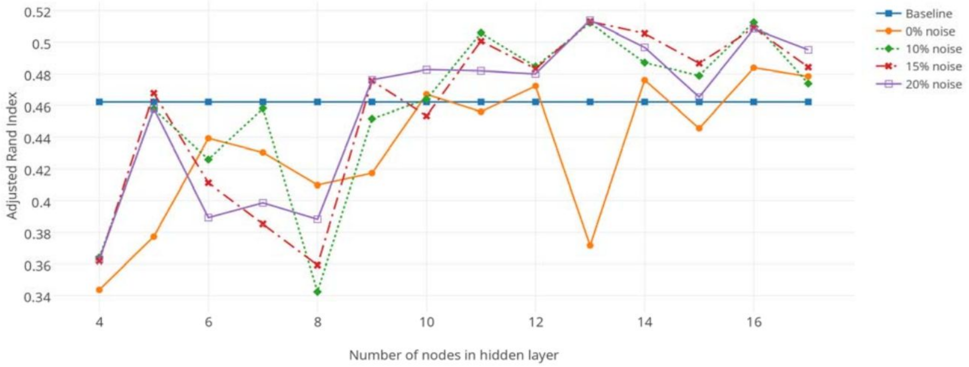
Shortcomings

- Will this generalize?
- Clustering
- Weak Evidence

Comparison for clustering score for raw data and regenerated data : Yeast dataset 2



Comparison for clustering score for raw data and regenerated data : Yeast dataset 1



Questions?