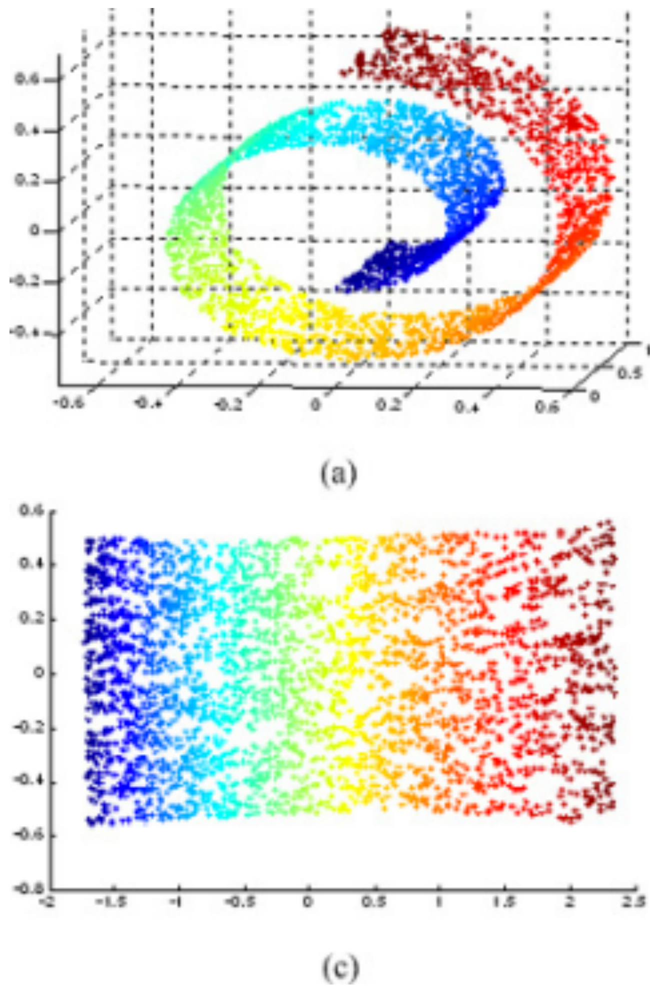
ML4Bio
Lecture #3: Dimensionality Reduction

March 9th, 2016

Quaid Morris

Review: Dimensionality reduction



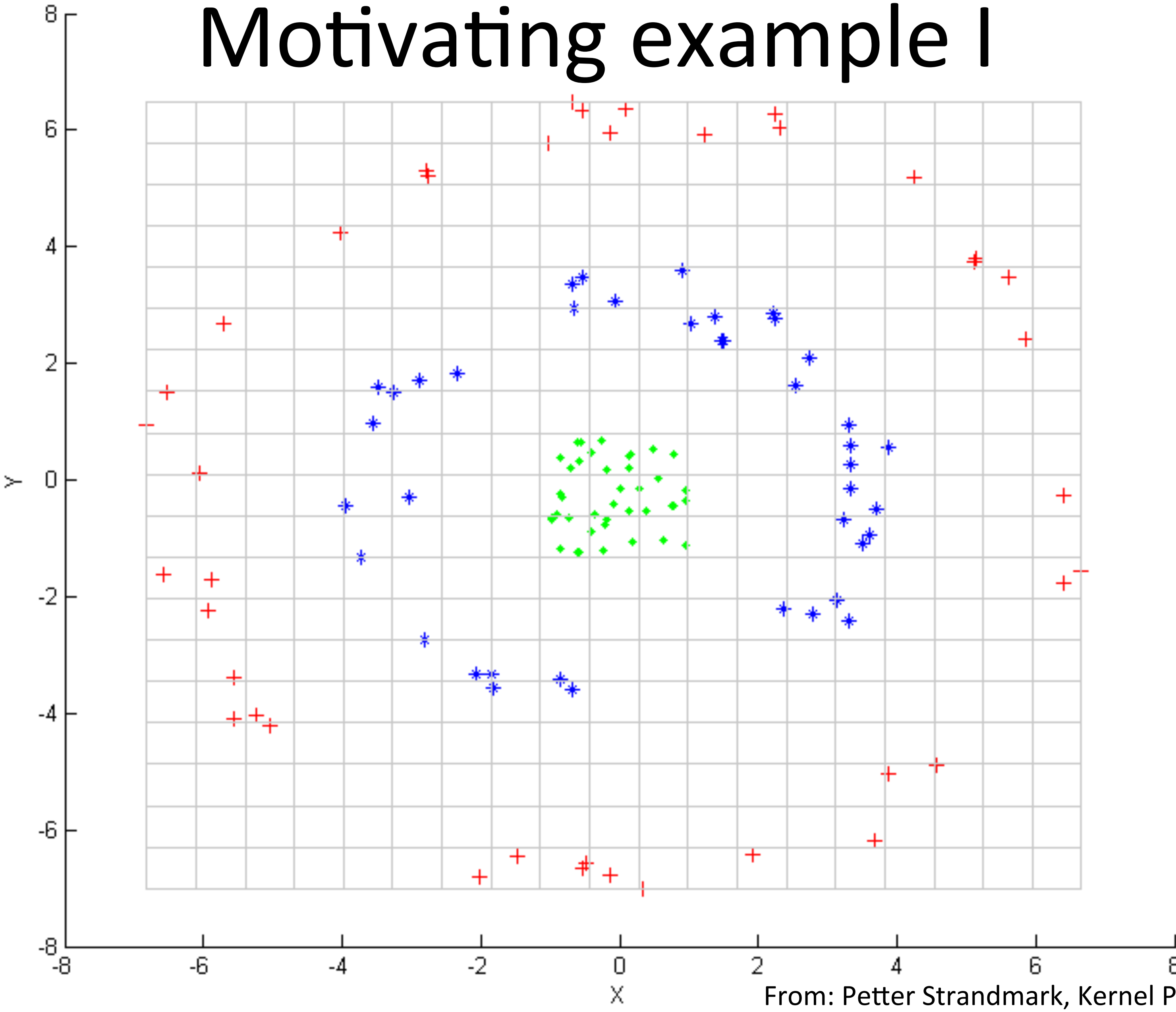
Do the datapoints lie on a lower dimensional "*manifold*"?

If so, what is the dimensionality?

How far apart are two datapoints, if you can only travel on the manifold?

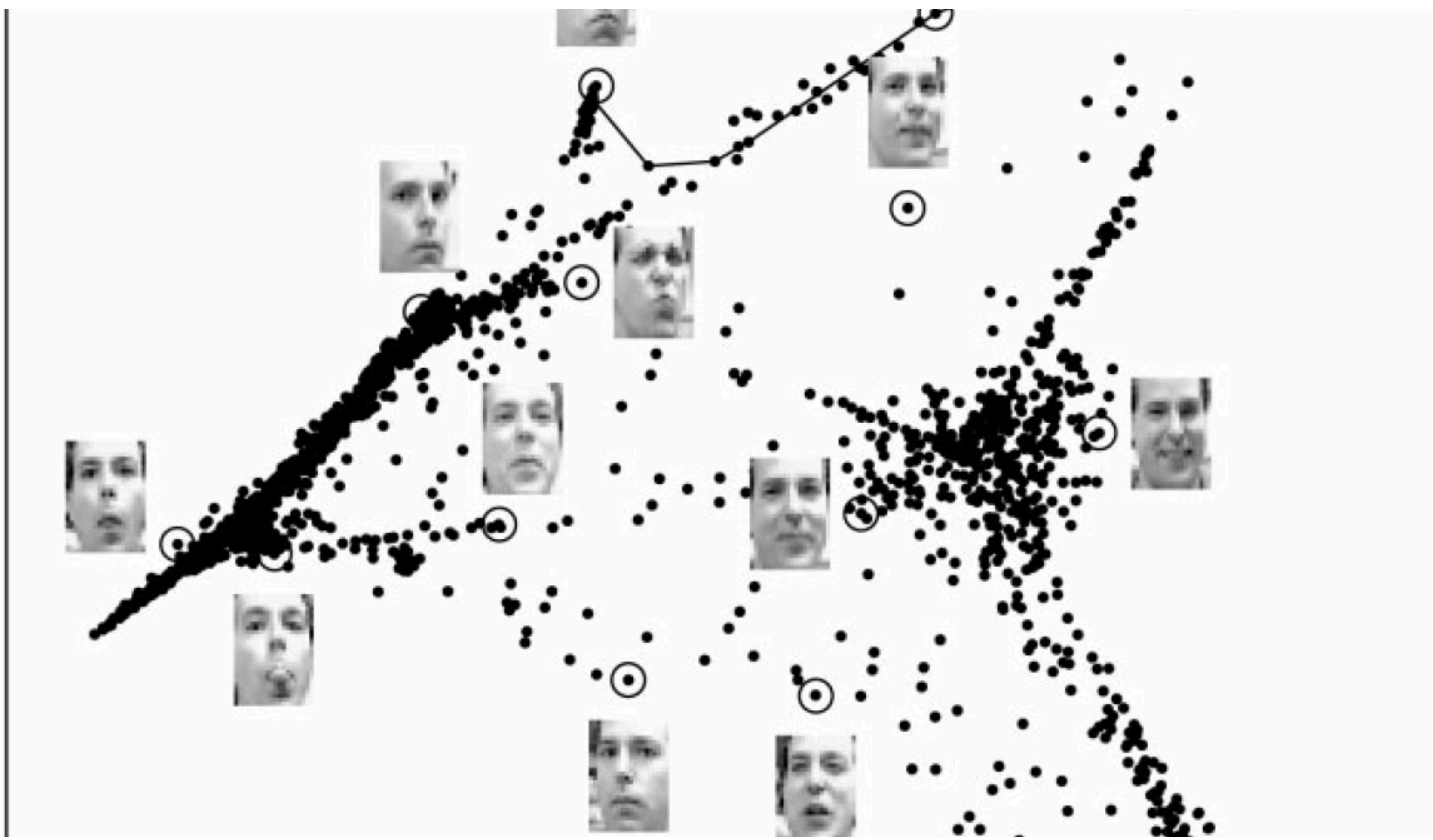
Image from PMID: 22595000

Motivating example I

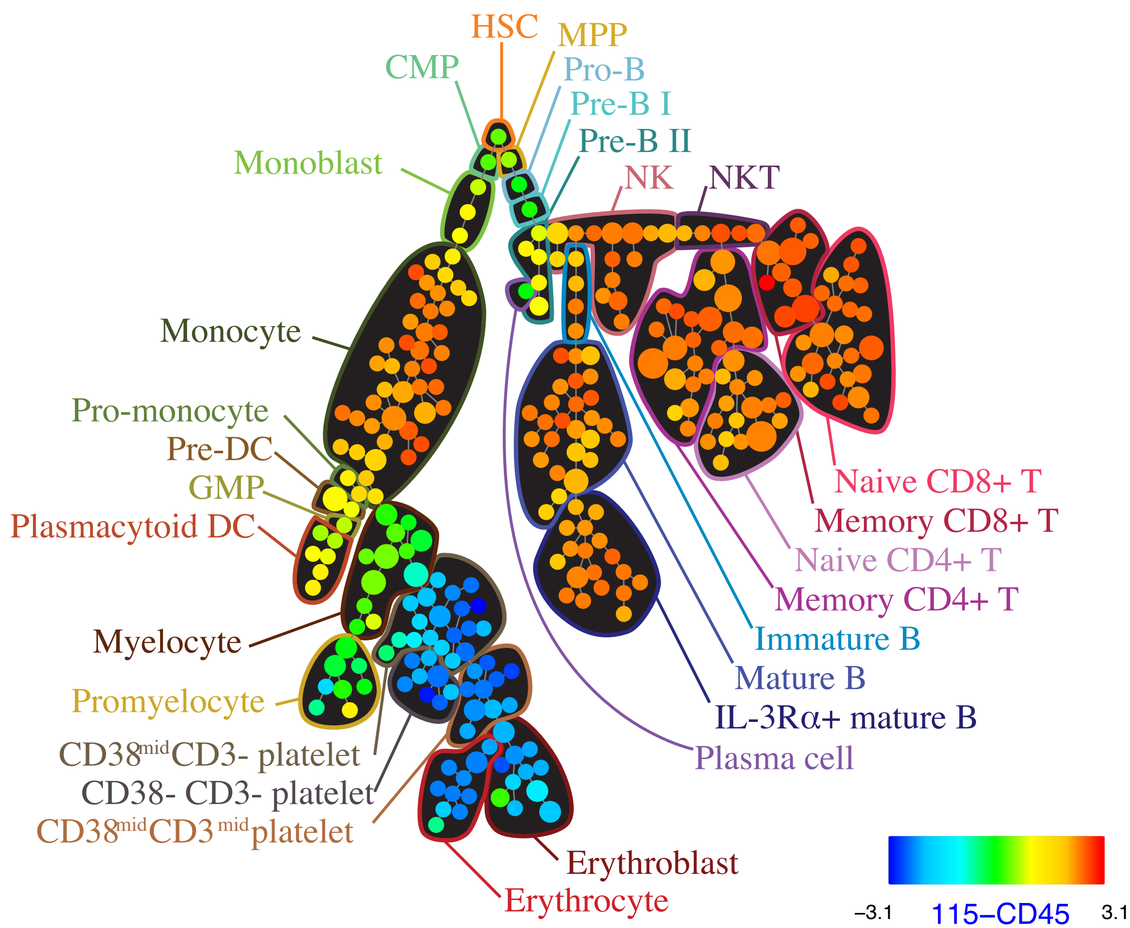


From: Petter Strandmark, Kernel PCA Wikipedia

Motivating example 2: Faces

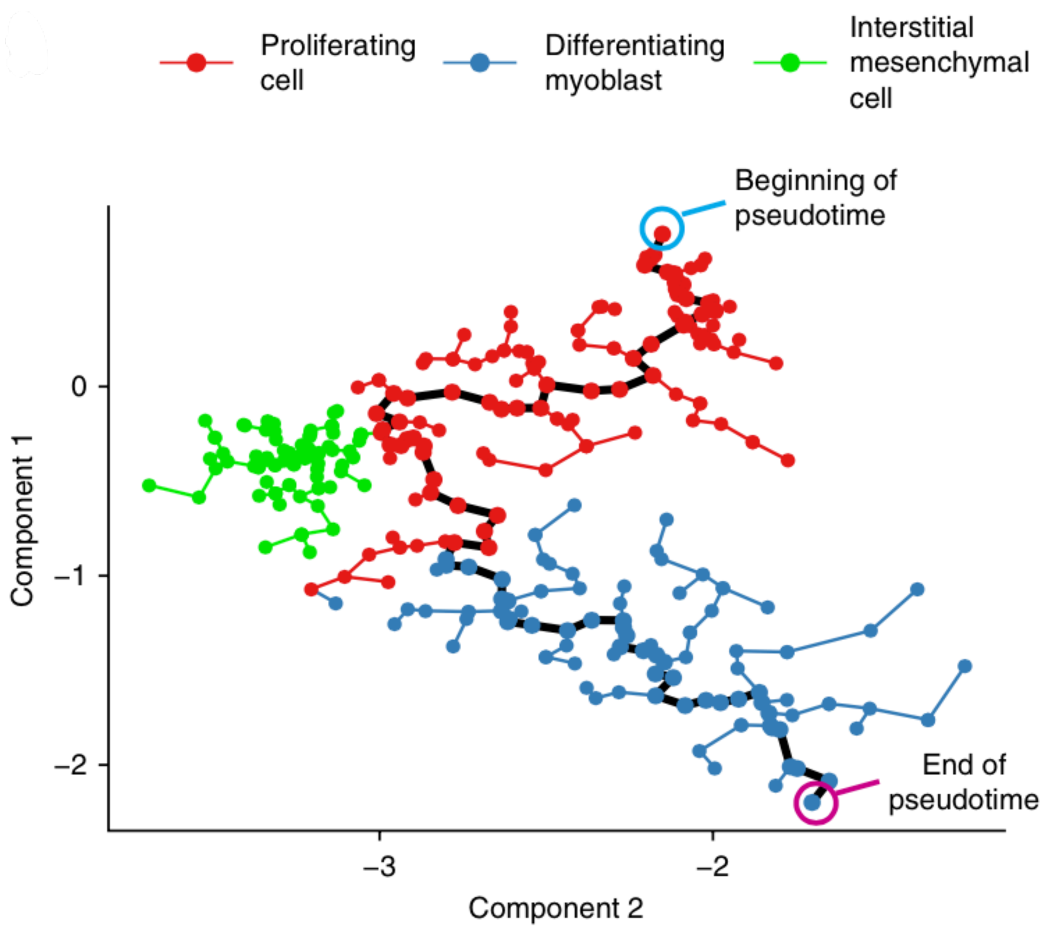


Motivating example 3: SPADE



Bendall et al, Science 2011

Motivating example 4: PCA

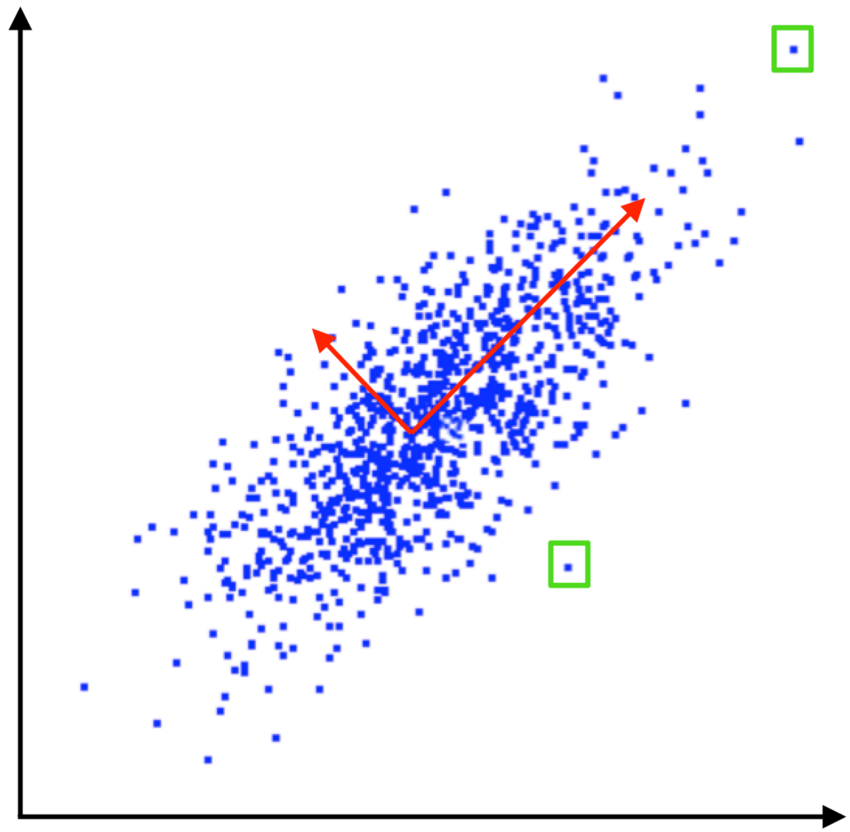


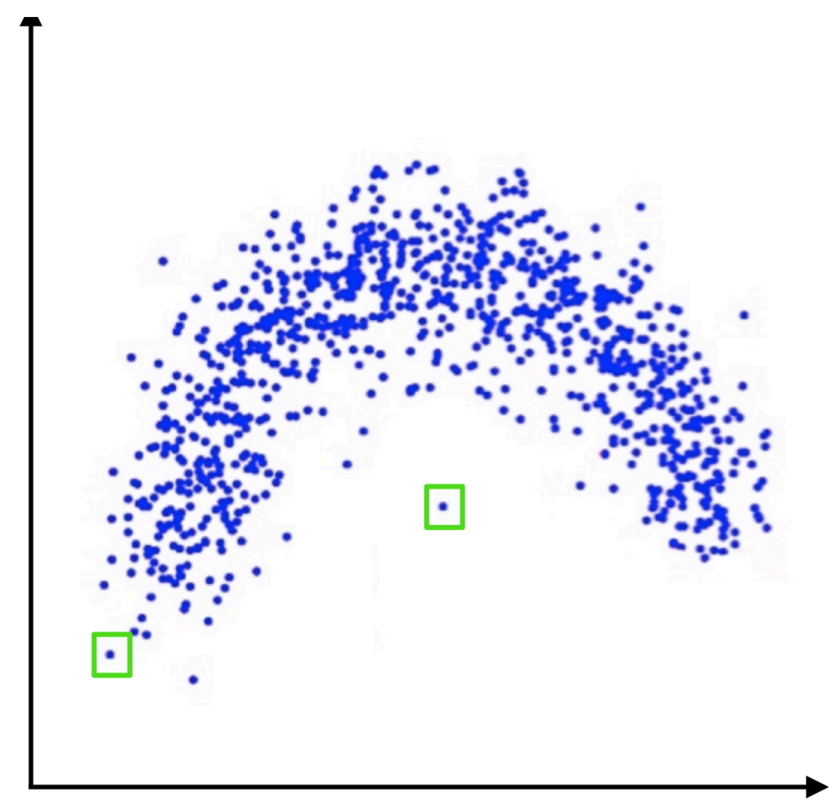
From “monocle”: Trapnell et al, NatBio 2014

What is dimensionality reduction?

Why do dimensionality reduction?

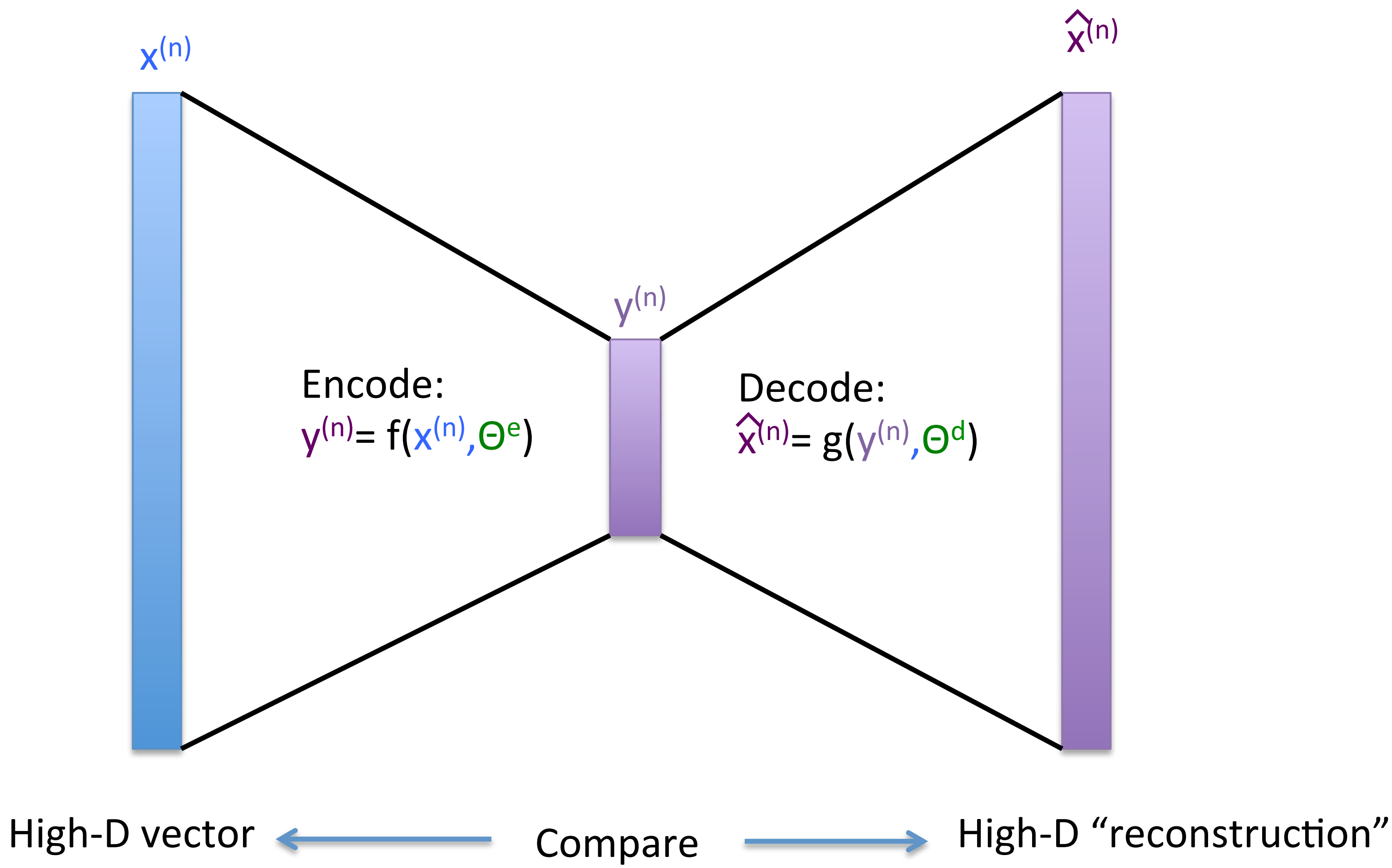
Why do dimensionality reduction?
"outlier detection"





Slide from Nil Sahin

How to do dimensionality reduction? Strategy #1: “autoencoder”



Four parts of the autoencoder:

1. Data: $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$
2. Model – encoding function $y^{(n)} = f(x^{(n)}, \Theta^e)$ and decoding function $\hat{x}^{(n)} = g(y^{(n)}, \Theta^d)$, each with their own parameters (or shared) parameters
3. Objective function –

$$E(\Theta^e, \Theta^d) = \Sigma_n (x^{(n)} - \hat{x}^{(n)})^T (x^{(n)} - \hat{x}^{(n)})$$

4. Optimization method – analytic (PCA) for linear functions, gradient descent otherwise.

Examples: PCA (linear) & deep autoencoders (non-linear)

Objective function example

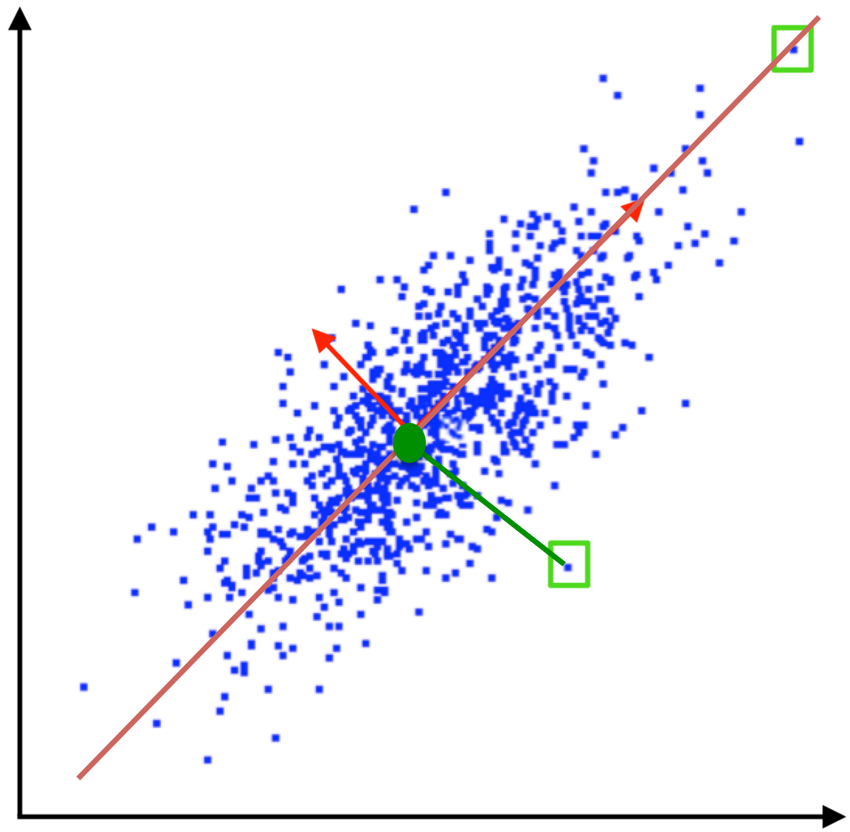
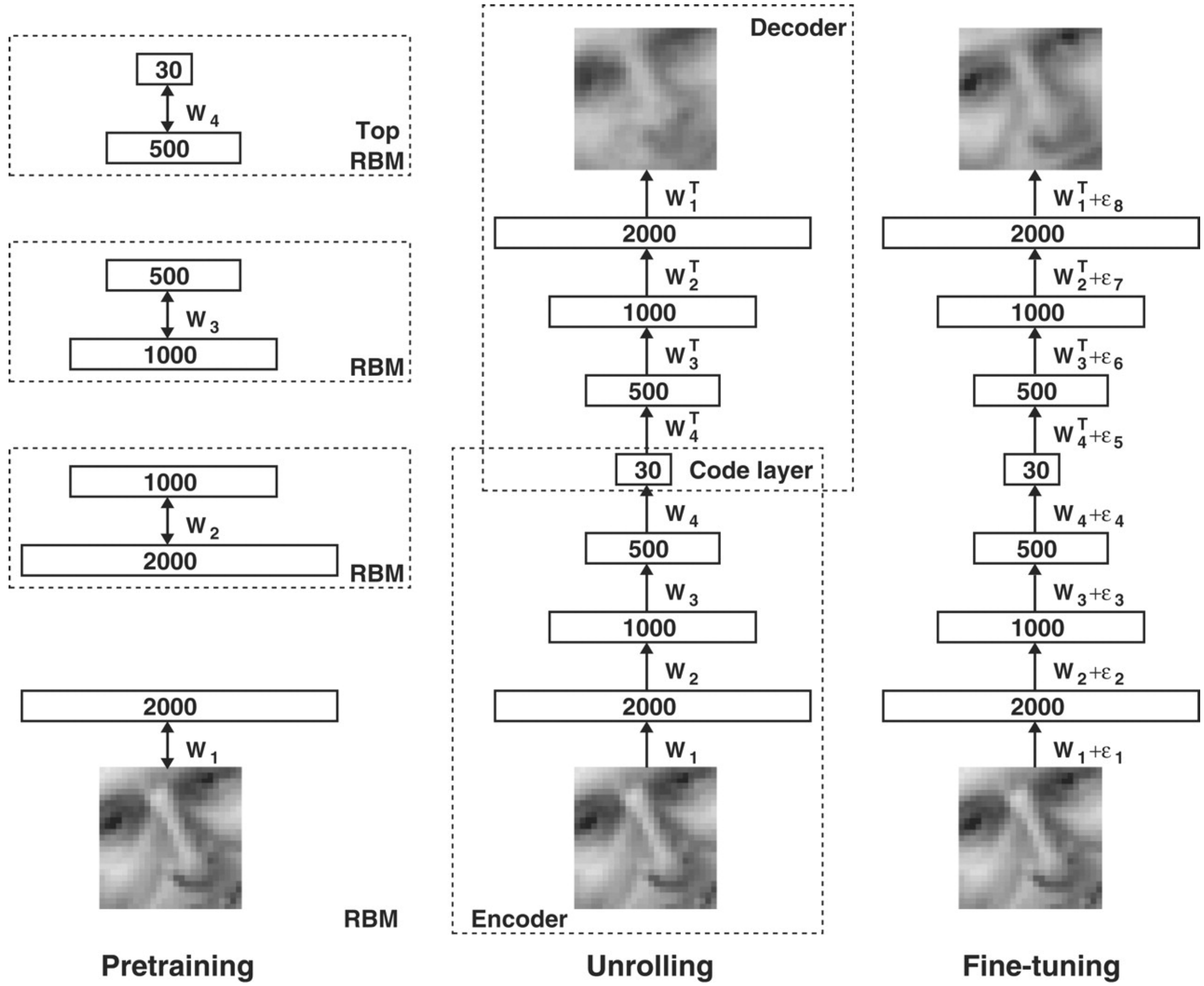


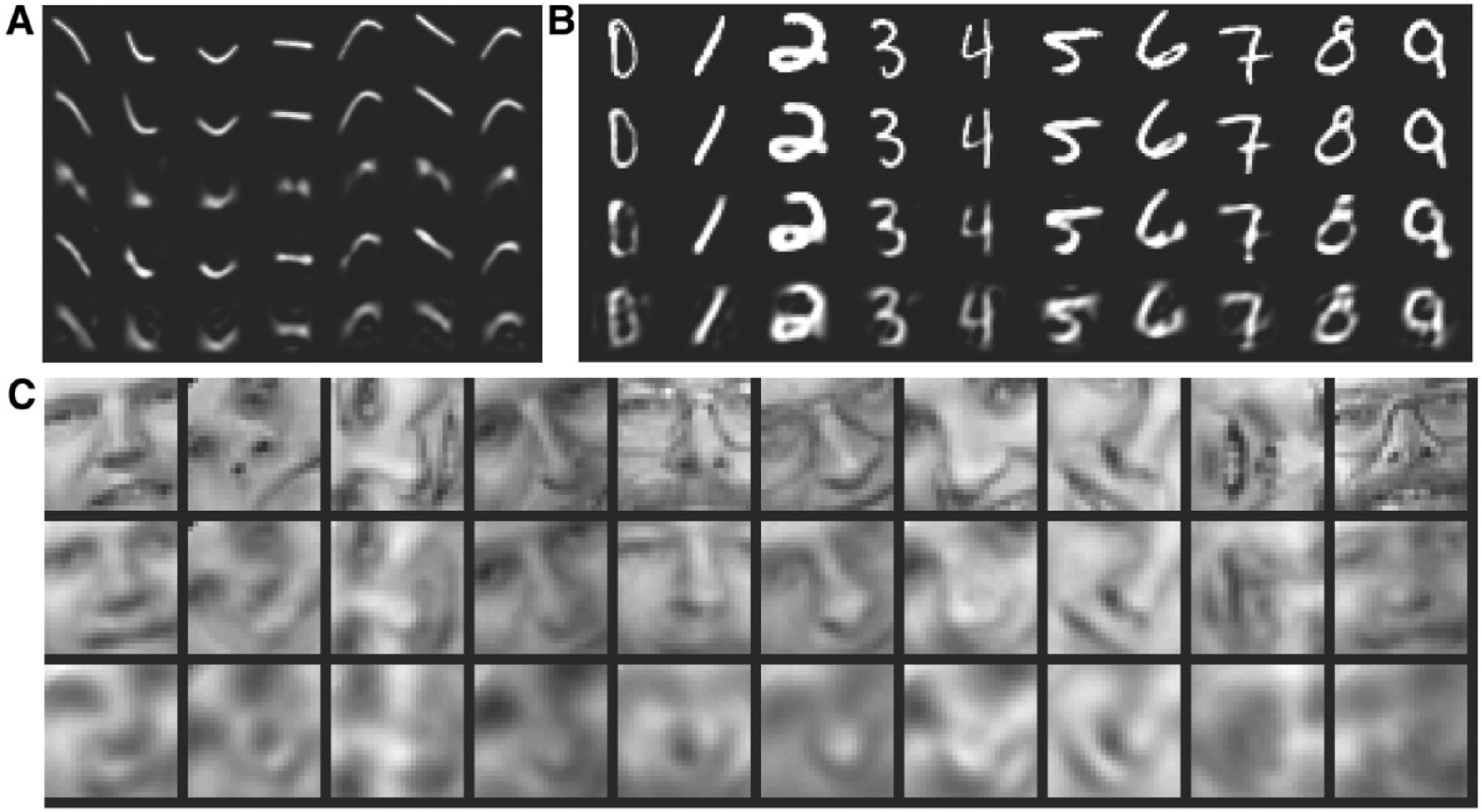
Fig. 1. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors.



G. E. Hinton, and R. R. Salakhutdinov Science 2006;313:504-507

Published by AAAS

Fig. 2. (A) Top to bottom: Random samples of curves from the test data set; reconstructions produced by the six-dimensional deep autoencoder; reconstructions by "logistic PCA" (8) using six components; reconstructions by logistic PCA and standard PCA using 18 components.

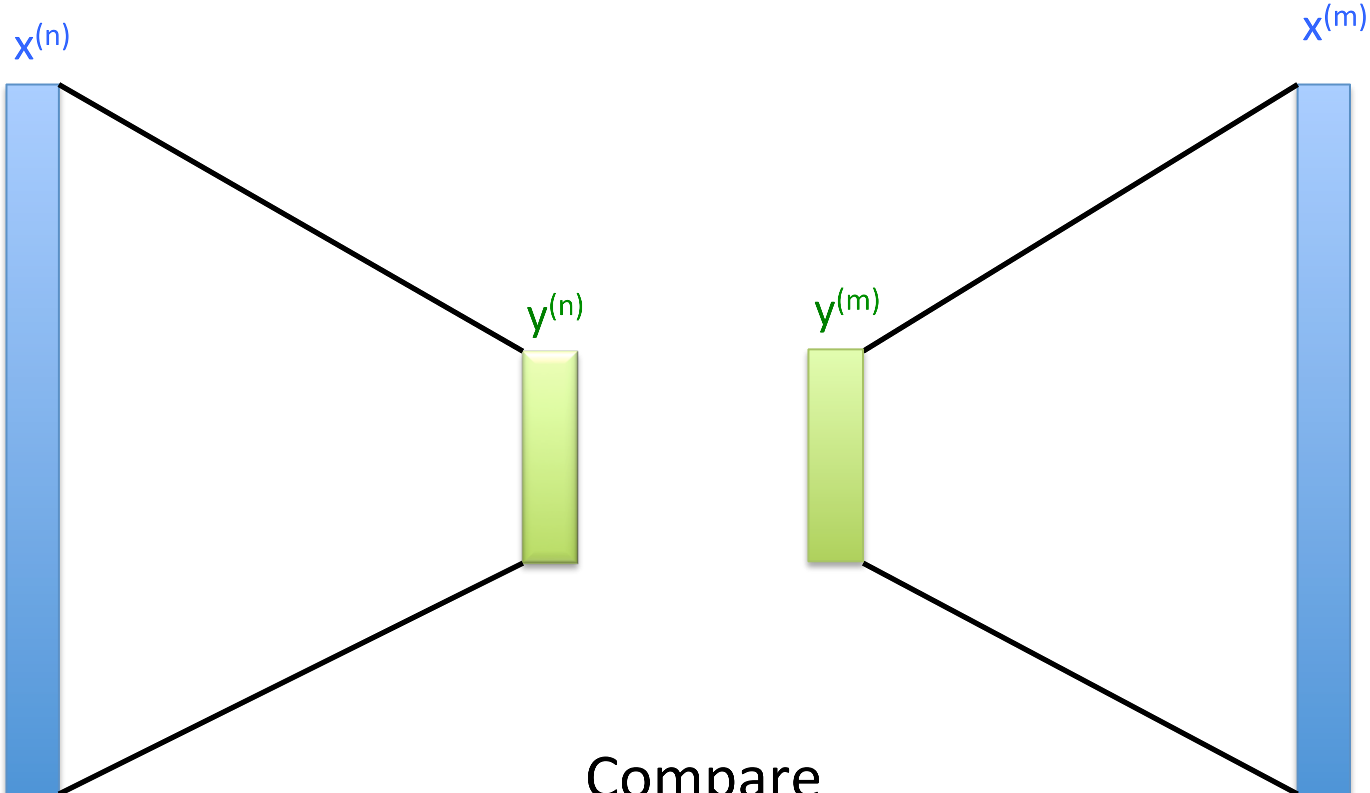


G. E. Hinton, and R. R. Salakhutdinov Science 2006;313:504-507

How to do dimensionality reduction?

Strategy #2: "recover distances"

e.g. Multidimensional scaling (MDS)



Compare

$d(x^{(n)}, x^{(m)}) = (x^{(n)} - x^{(m)})^T(x^{(n)} - x^{(m)})$

to $d(y^{(n)}, y^{(m)})$

Four parts of MDS & friends:

1. Data: $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$
2. Model – parameters are low-d coordinates of each point:

   $\Theta = \{y^{(1)}, y^{(2)}, \ldots, y^{(N)}\}$

1. Objective function –

$$E(\Theta) = \Sigma_{n,m} [d(x^{(n)}, x^{(m)}) - d(y^{(n)}, y^{(m)})]^2$$

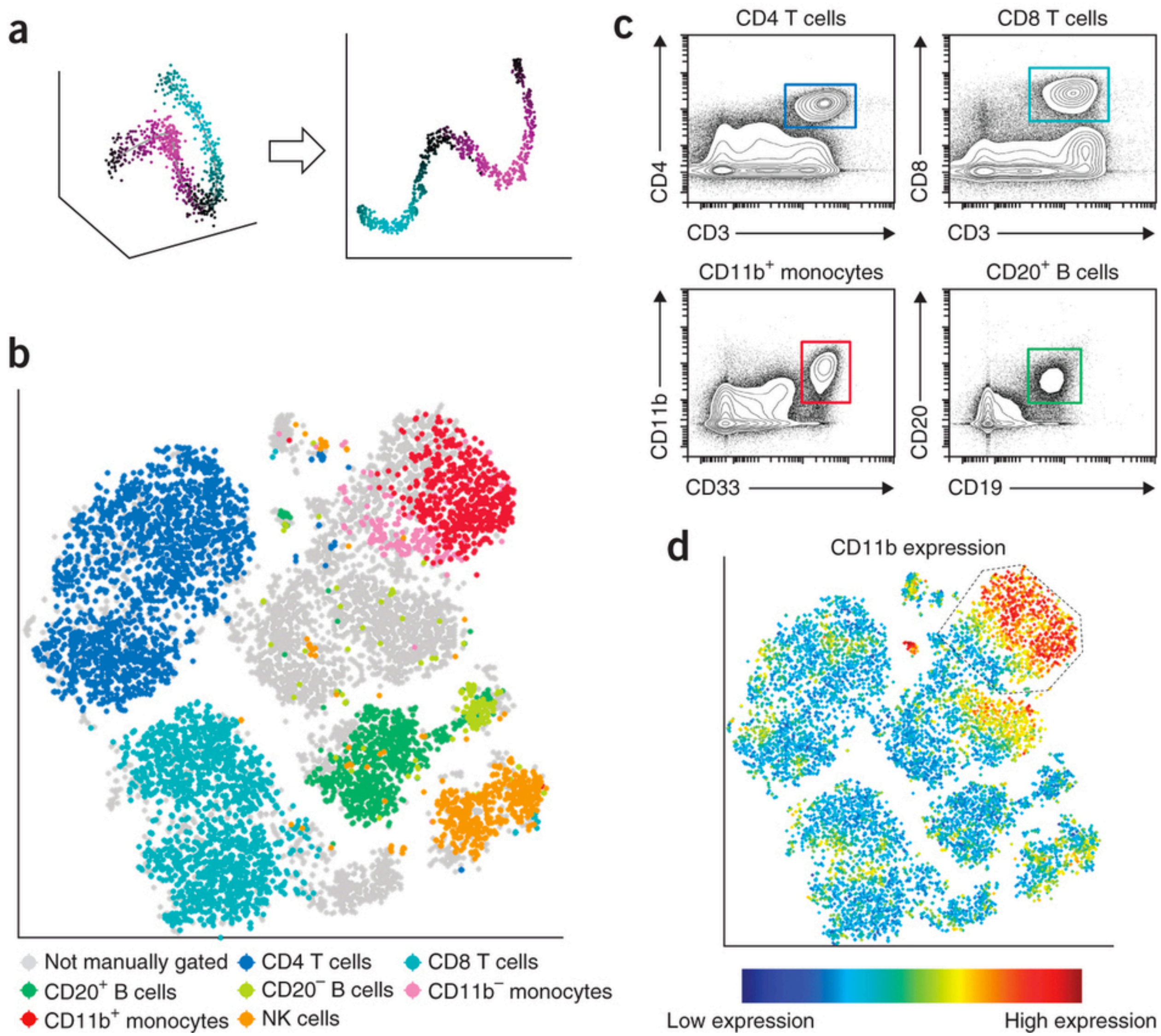
2. Optimization method – gradient descent usually.

Examples: MDS, ISOmap, Local linear embedding, SNE, t-SNE

Important variations

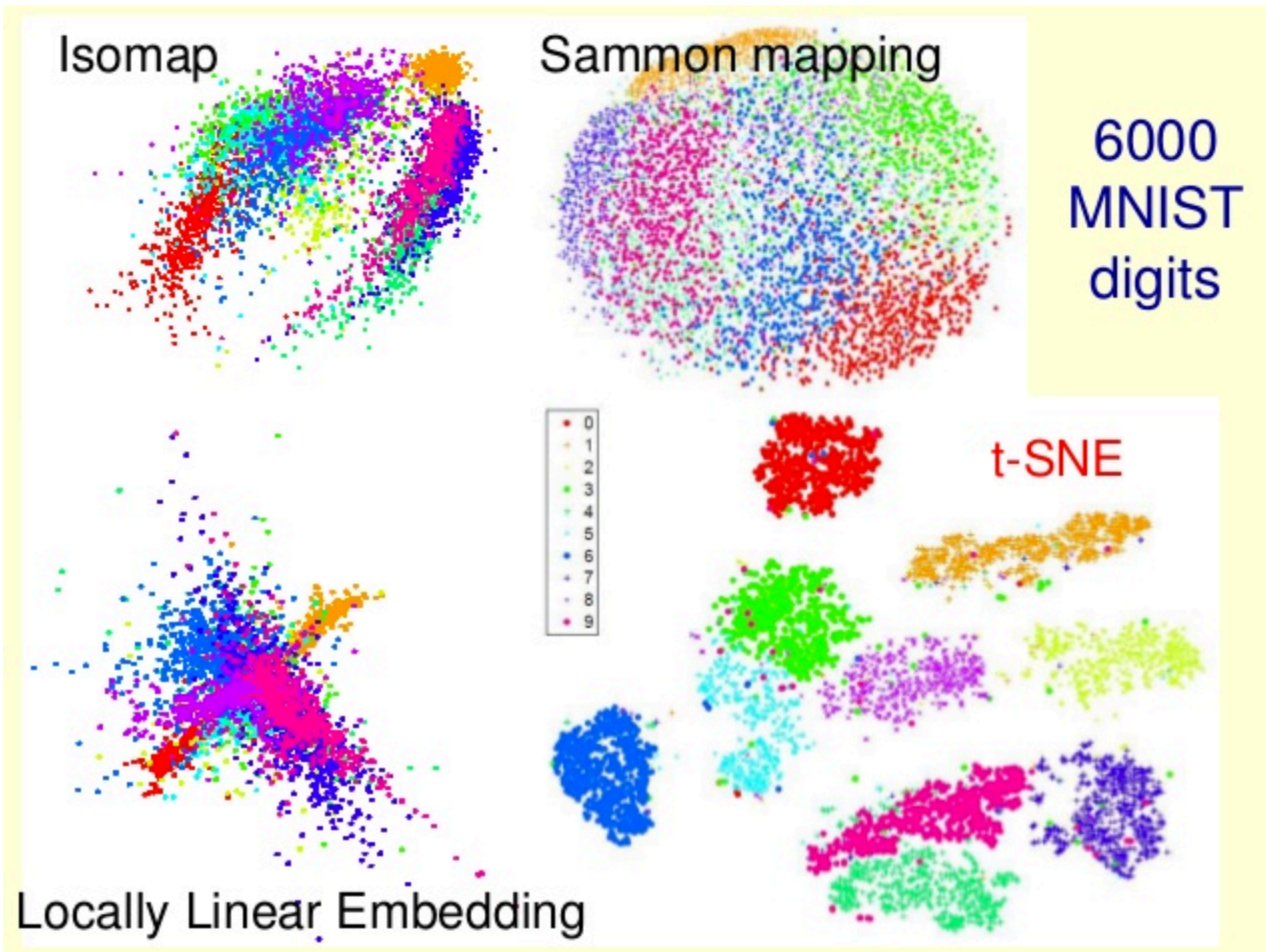
1. ISOmap (Langford, Science 1999): use “network distance” rather than Euclidean
2. SNE (Hinton): use “similarity” rather than distance, slightly different objective function
3. t-SNE (Hinton) / viSNE (Dana Pe’er): ignore long distances, use SNE-style objective function
4. LLE (Roweis, Science 1999): use local linear models, ignore long distances

viSNE example



Amir et al, NatBio 2013

Examples



Yan Xu Jun 7, 2013, Slideshare