Journal of Machine Learning Research 9 (2008) 2579-2605 Submitted 5/08; Revised 9/08; Published 11/08

Visualizing Data using t-SNE

Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Editor: Yoshua Bengio

Paper Presentation
Toronto Deep Learning Series (Classics)

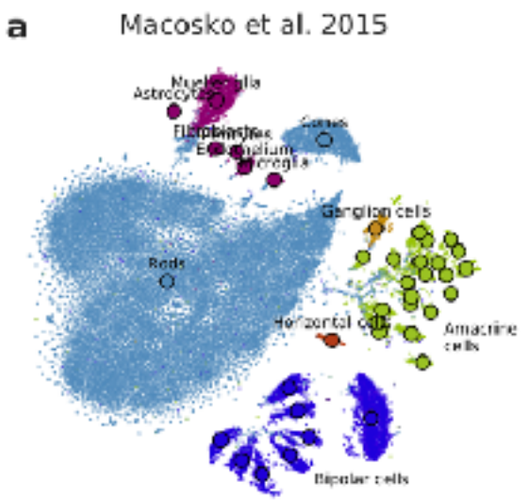
Presenter: Sabyasachi Dasgupta, University of Toronto

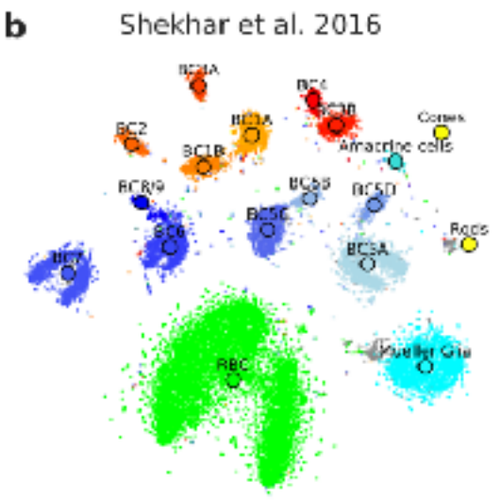
Outline of the talk

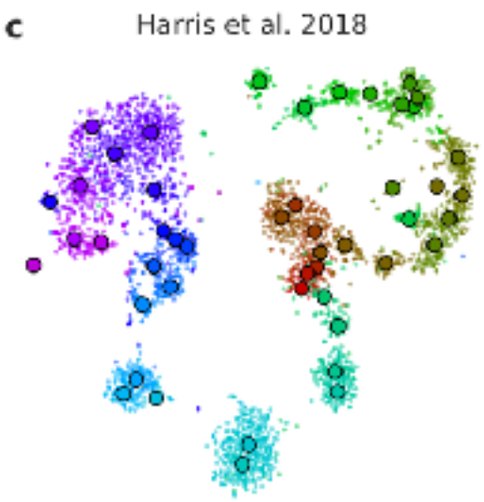
- Theoretical foundation of SNE — Hinton & Roweis, 2002
- Mutation of SNE gives a variant t-SNE — Maaten & Hinton, 2008
- t-SNE Practicals
- Using t-SNE effectively. — Wattenberg, et al., Distill, 2016.
- t-SNE pitfalls
- Discussions:
  - (PCA vs SNE vs t-SNE)
  - Hyper-parameters (P or perplexity, Learning rate, Iterations, Momentum etc.)
  - Approximate K-NN / K = 3P for attractive forces. — Maaten 2014
  - Barnes Hutt approximation for faster repulsive force calculations.
  - Fit-SNE implementation using a FFT transform for the repulsive interactions. — Lindermann 2017

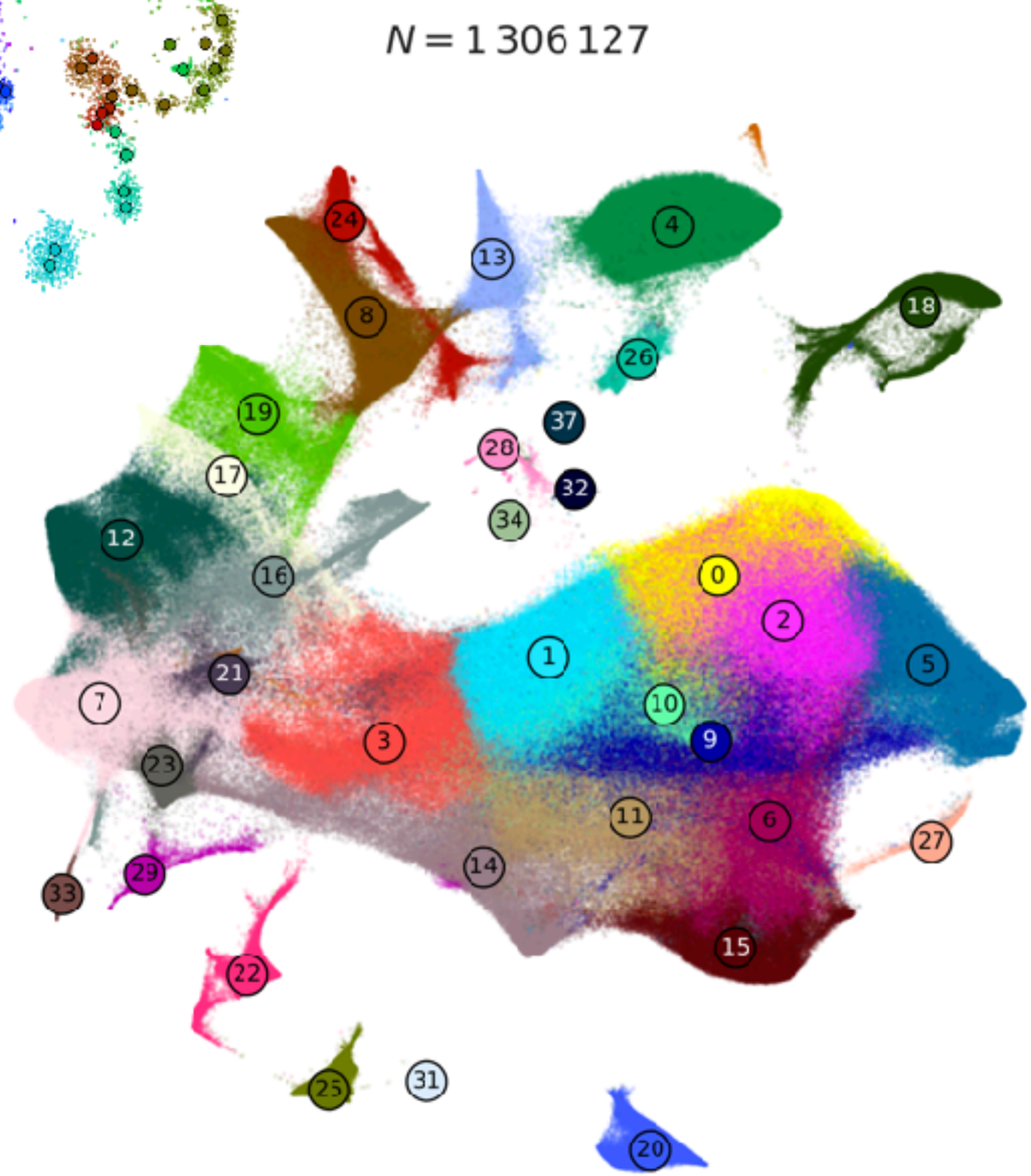
t-SNE Teasers!

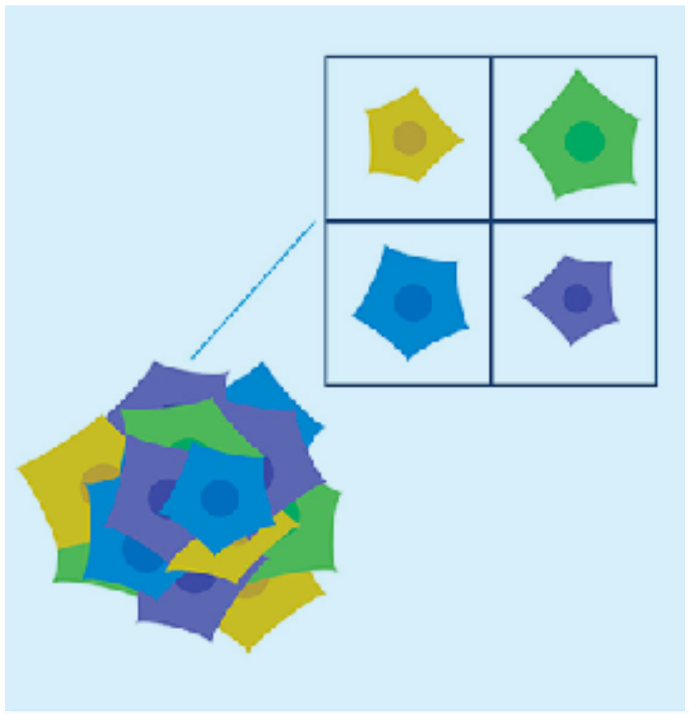
Genomics





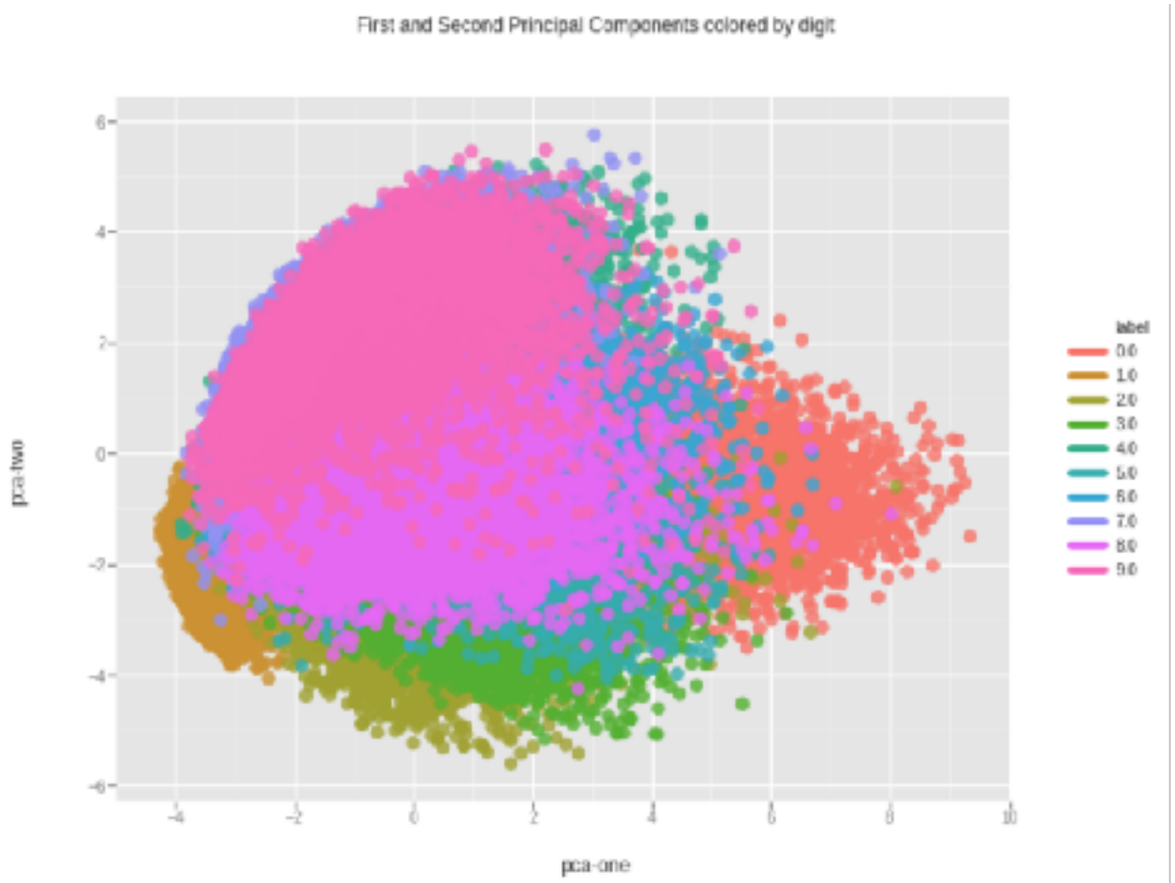


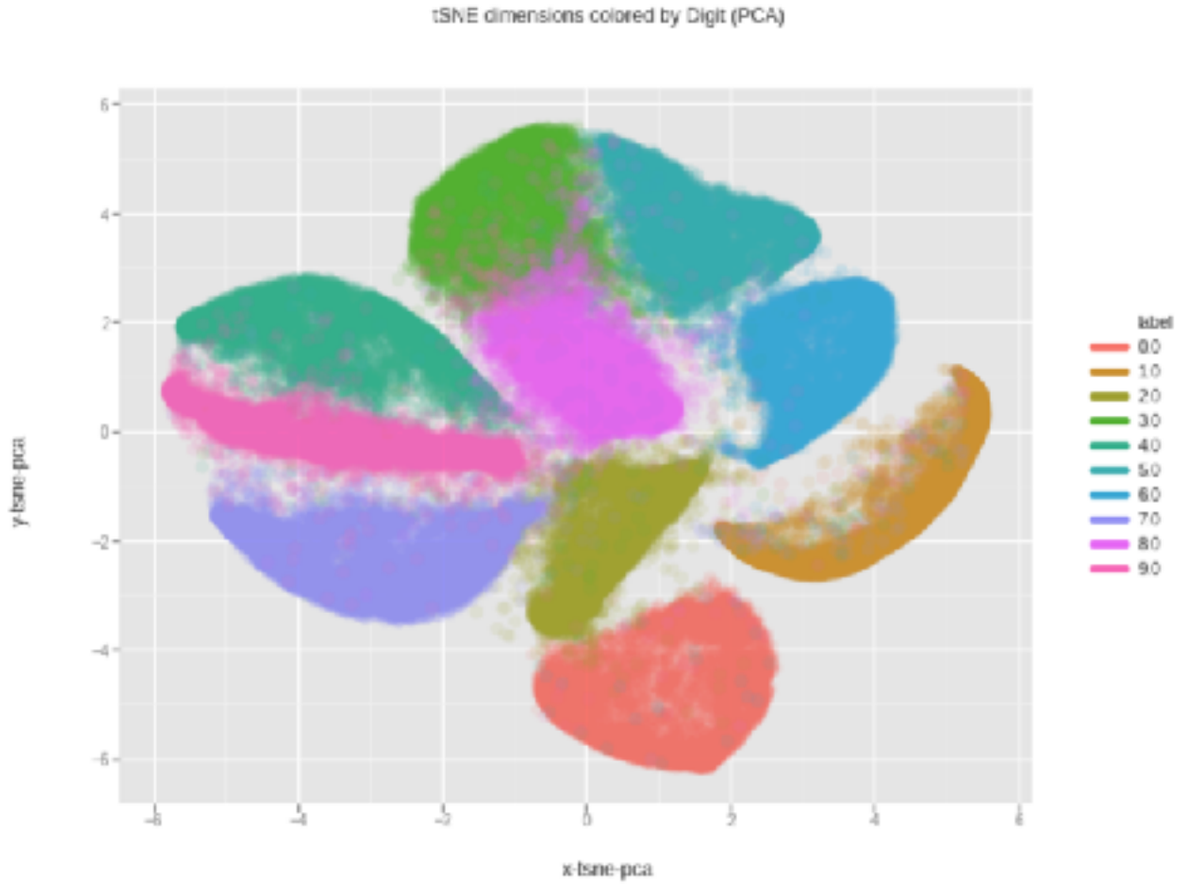




https://www.10xgenomics.com/solutions/sir

PCA vs t-SNE [MNIST dataset]





Visualization of High-Dim data

https://cs.stanford.edu/people/karpathy/cnnembed/

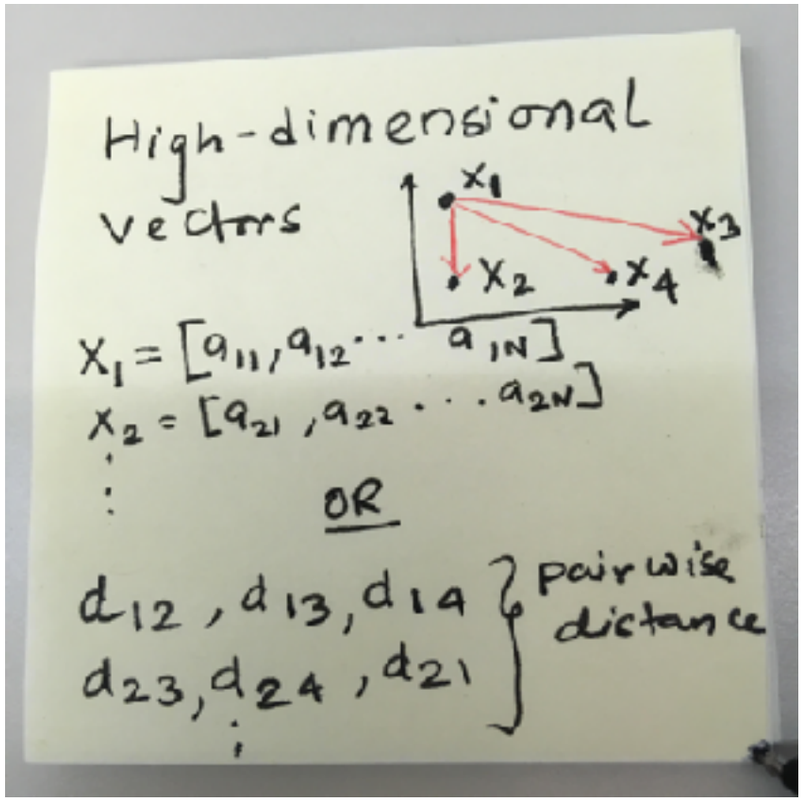
Visualization methods

Use a distance metric

High Dim Data

$[\hat{a1}, \hat{a2}, \hat{a3}..N]$

transform



Linear	Non linear
PCA	
MDS	
	LLE
	ISOMAP
	SNE/t-SNE
	D-Map
	U-Map

manifold

Low Dim Map

$[\hat{b1}, \hat{b2}, \hat{b3}]$

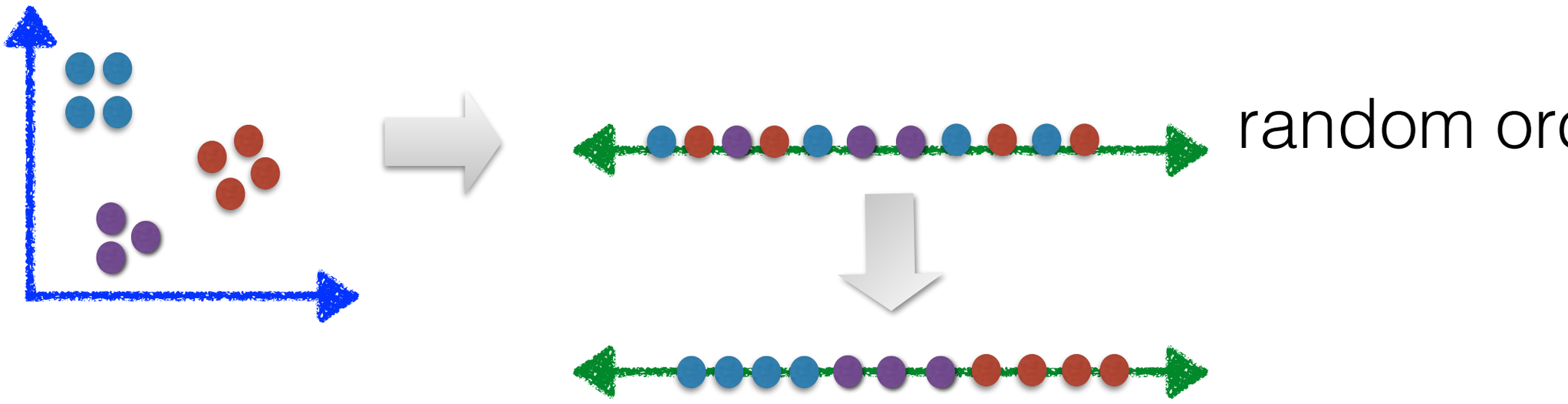
Visualize in 2/3 dimensions

SNE uses either vectors / pairwise dissimilarities to create probabilistic approach to reduce dimensioned visualize data.

Discuss- Local vs. Global

Visual interpretation

Informally, the algorithm places all points on the 2D plane, initially at random positions, and lets them interact as if they were physical particles. The interaction is governed by two “laws”: first, all points are repelled from each other; second, each point is attracted to its nearest neighbours.

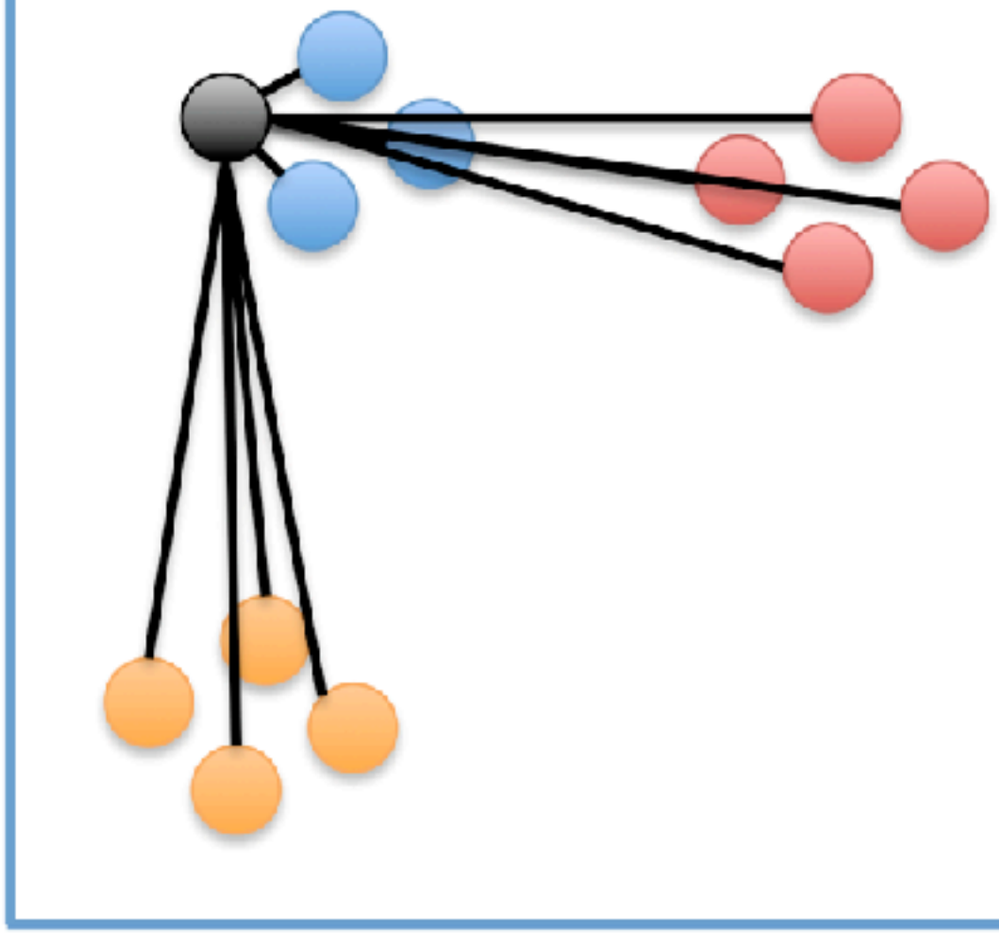


random order

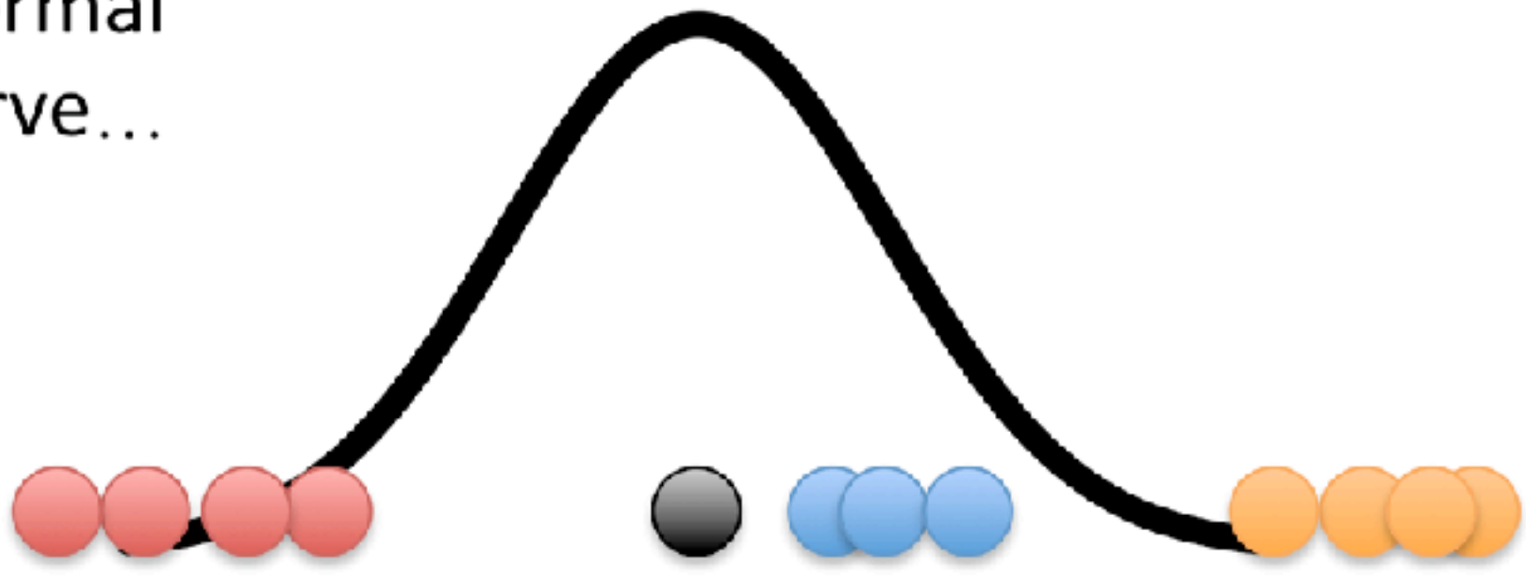
pairwise interactions

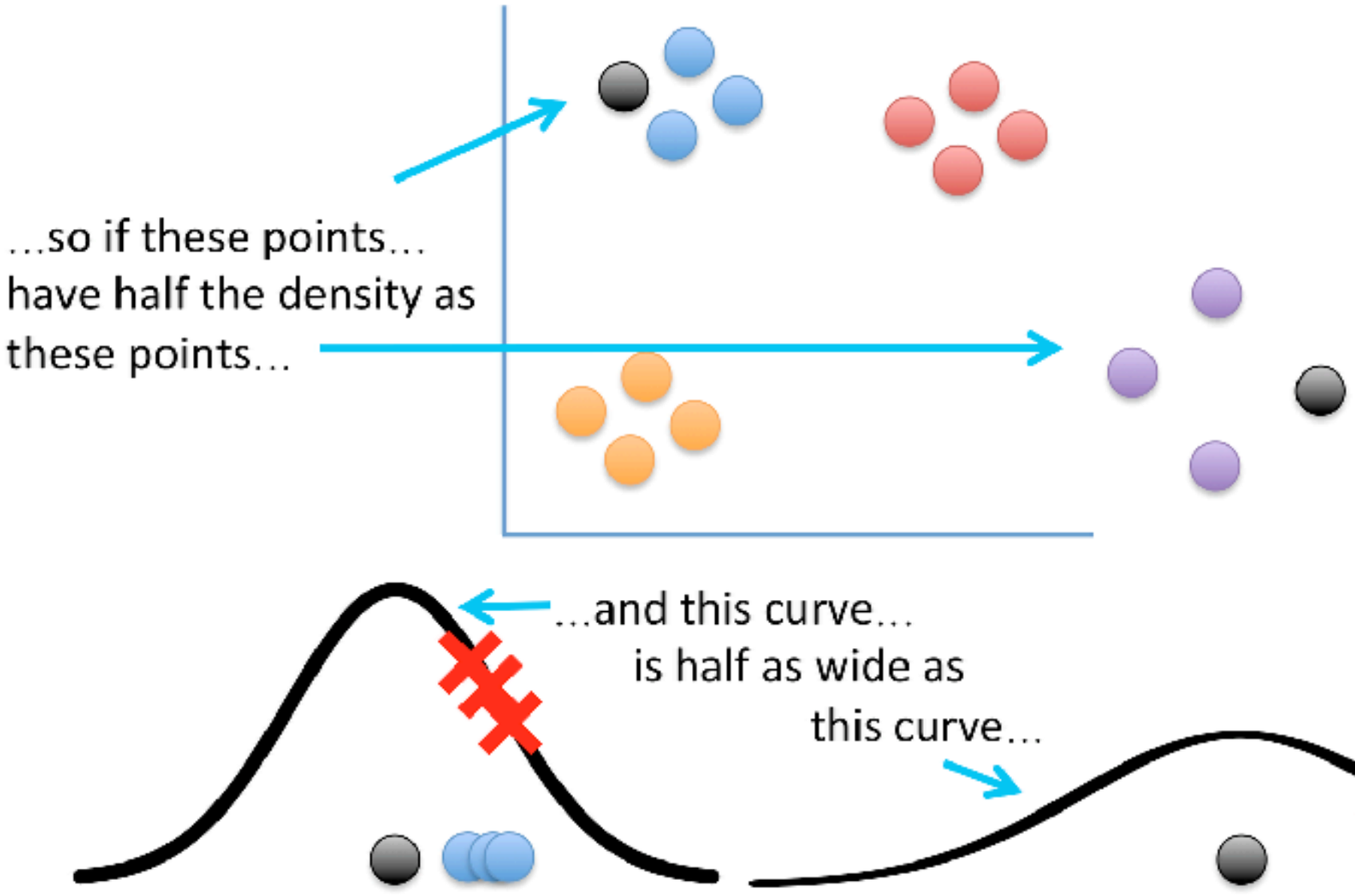
1. attractive
2. repulsive

Ultimately, we measure the distances between all of the points and the point of interest…



Plot them on the normal curve…

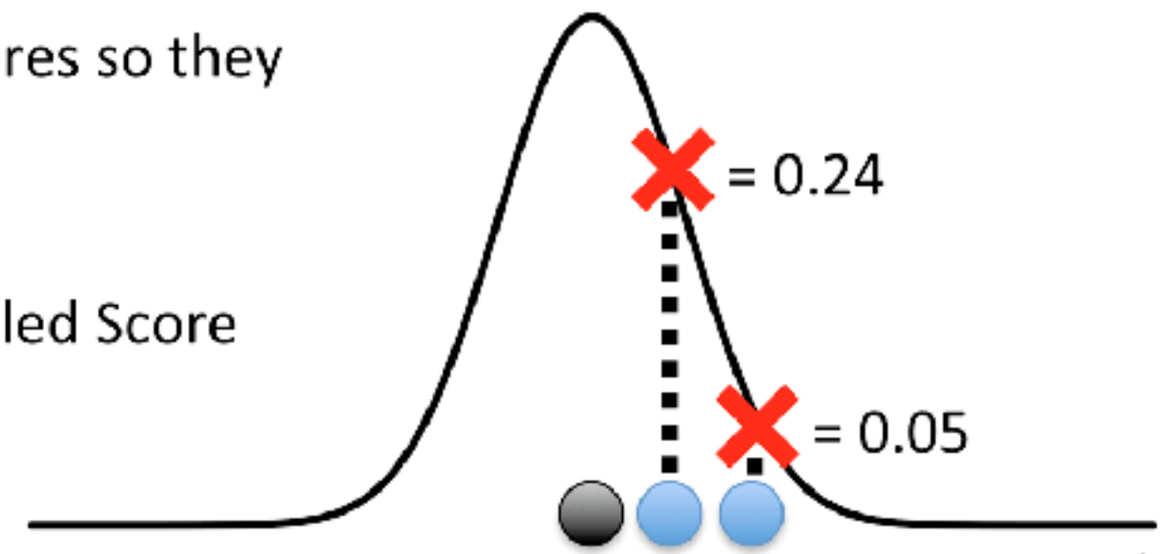




…then scaling the similarity scores will make them the same for both clusters.

To scale the similarity scores so they sum to 1:

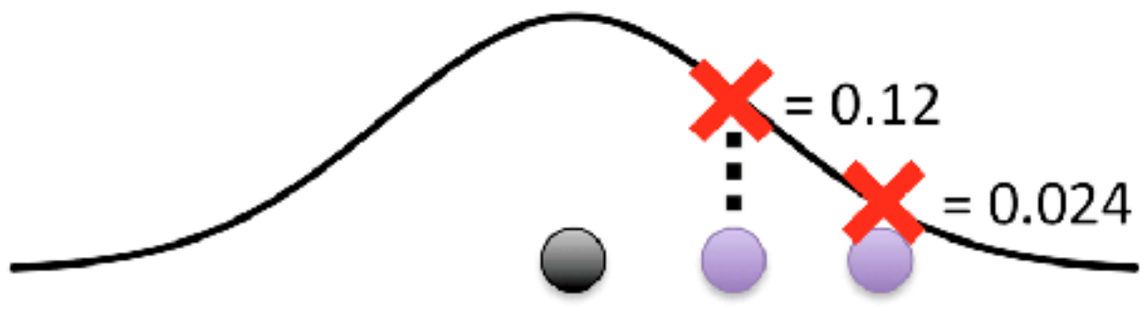
$$\frac{\text{Score}}{\text{Sum of all scores}} = \text{Scaled Score}$$



$$\frac{0.24}{0.24 + 0.05} = 0.82$$

$$\frac{0.05}{0.24 + 0.05} = 0.18$$

These are the same as these!



$$\frac{0.12}{0.12 + 0.024} = 0.82$$

$$\frac{0.024}{0.12 + 0.024} = 0.18$$

Theoretical premise-I

Eucledian metric [local linearity assumption]

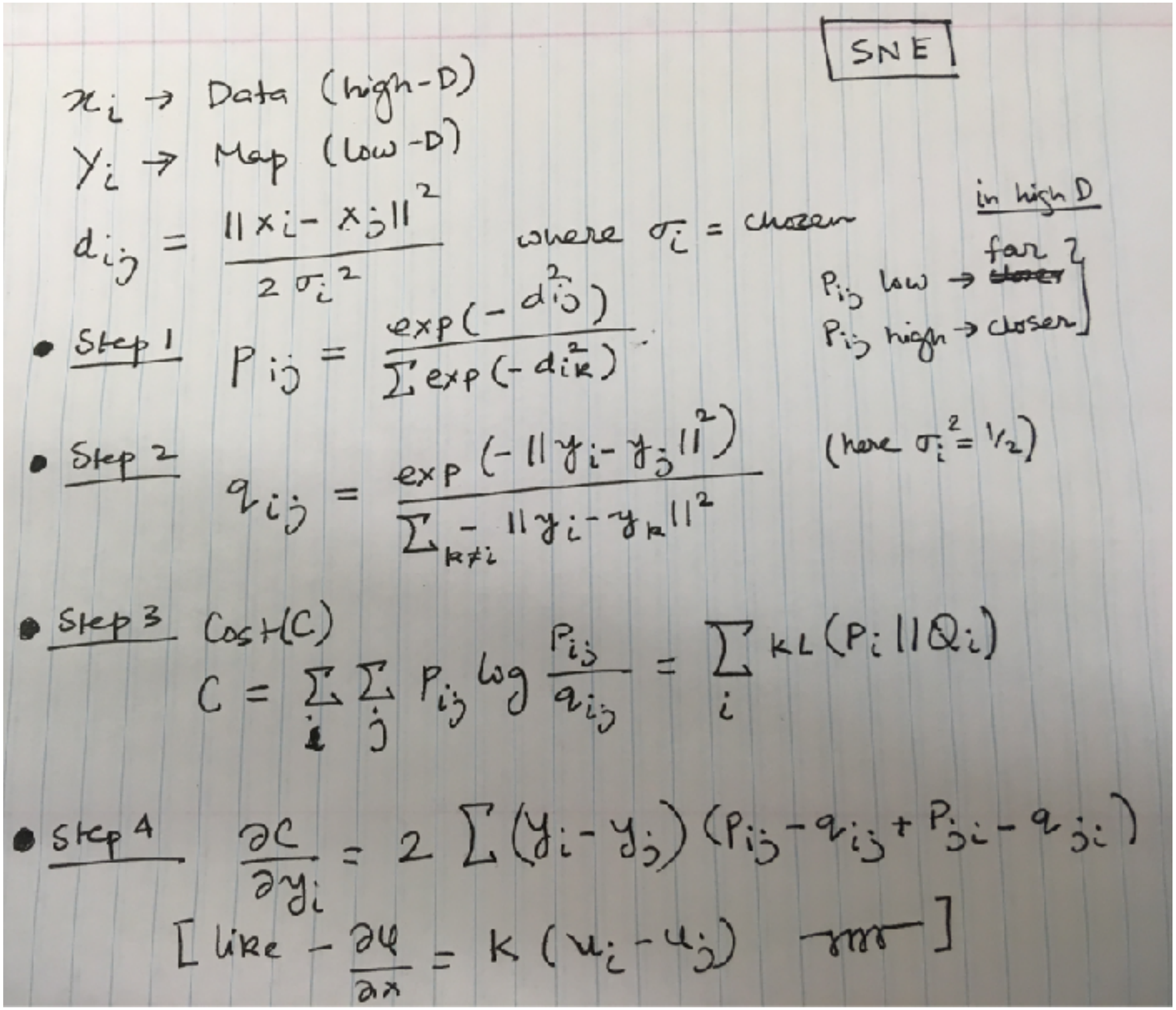
$\text{Perplexity} = 2^{-\sum_j p_{ji} \log_2 p_{ji}}$

Assymetric Probability

Guassian neighborhood in Low D

Cost function: Minimize the sum of KL divergence terms for each point

Points move in lower manifold due to gradient term.



Theoretical premise-III

- **Step 5** Gradient minimization update

$$y^{(t)} = y^{(t-1)} + \eta \frac{\partial C}{\partial y} + \alpha(t)\left(y^{(t-1)} - y^{(t-2)}\right)$$

η: learning rate; $\alpha(t)$: momentum

+ gaussian noise ~ simulated annealing

Variation ① Symmetric SNE

$$p_{ij} = p_{ji} \quad \left.\begin{array}{l} \\ \\ \Rightarrow \sum_j p_{ij} > 1/2n \end{array}\right\} + p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \Bigg\} \text{outlier control mechanism}$$

$$q_{ij} = q_{ji}$$

gradient term $$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)$$

Theoretical premise-II

1) P_{ij} are assymetric

2) Closer points $C_A \rightarrow P_{ij}$ high
far points $C_B \rightarrow P_{ij}$ low

as $C = KL(P_i || Q_i)$

$$= \sum_j P_{ij} \log \frac{P_{ij}}{q_{ij}}$$

$$= \sum_j P_{ij} \log P_{ij} - \sum_j P_{ij} \log q_{ij}$$

$$= \text{const} - \sum P_{ij} \log q_{ij}$$

$C_A > C_B \Rightarrow$ In low dim more local embedding is preferred than furthur points placed together.

Widely separated points can be collapsed as nearby points in Map [Drawback]

Crowding

2D → 1D

But generally not possible

1D – 2 E.D. points

2D → 2+1 E.D. points.

10 D → 11 points

But folding it back in 2D → hard without compression

Case 2

Diff. distribution of pairwise distances in HiD & LowD

Solution

in Hi/Dim { Near points → close in Map
Modest p → furthur in Map

Theoretical premise-IV

t-SNE

- **Step 1** $P_{ij} = \frac{P_{j|i} + P_{i|j}}{2n}$ Calculate ($P_{j|i}$ as SNE)
- **Step 2** $q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$ [choice of t-dist. avoids overcrowding]
- **Step 3** Cost function

$$C = KL(P \| Q) =$$

- **Step 4** $\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}$

Algorithm

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.

begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=*1* **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)} \right)$
  end
end

Box 1: The t-SNE algorithm The t-SNE algorithm (van der Maaten and Hinton, 2008) is based on the SNE framework (Hinton and Roweis, 2003). For any given point i, SNE introduces a notion of directional similarity of point j to point i,

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$

defining a probability distribution over all points $j \neq i$ (all $p_{i|i}$ are set to zero). The variance of the Gaussian kernel σ_i^2 is chosen such that the perplexity of this probability distribution

$$\exp\Big(-\ln(2)\cdot\sum_{j\neq i} p_{j|i}\log_2 p_{j|i}\Big)$$

has some pre-specified value. The larger the perplexity, the larger the variance of the kernel, with the largest possible perplexity value equal to $N-1$ corresponding to $\sigma_i^2=\infty$ and the uniform probability distribution (N is the number of points in the data set). Importantly, for any given perplexity value P, all but $\sim P$ nearest neighbours of point i will have $p_{j|i}$ very close to zero. For mathematical and computational convenience, *symmetric SNE* defines undirectional similarities

$$p_{ij} = \frac{p_{i|j}+p_{j|i}}{2N}$$

such that $\sum_{i,j} p_{ij} = 1$, i.e. this is a valid probability distribution on the set of all pairs (i,j).

The main idea of SNE and its modifications is to arrange the N points in a low-dimensional space such that the similarities q_{ij} between low-dimensional points match p_{ij} as close as possible in terms of the Kullback-Leibler divergence. The loss function is thus

$$\mathcal{L} = \sum_{i,j} p_{ij}\log\frac{p_{ij}}{q_{ij}}.$$

The main idea of t-SNE was to use a t-distribution with one degree of freedom (also known as Cauchy distribution) as the low-dimensional similarity kernel:

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = \frac{1}{1+\|\mathbf{y}_i - \mathbf{y}_j\|^2}, \quad Z = \sum_{k\neq l} w_{kl},$$

where $\mathbf{y}_i$ are low-dimensional coordinates (and $q_{ii}=0$). As a matter of definition, we consider any method that uses the t-distribution as the output kernel and Kullback-Leibler divergence as the loss function to be "t-SNE"; similarities $p_{j|i}$ can in principle be computed using non-Euclidean distances instead of $\|\mathbf{x}_i - \mathbf{x}_j\|$ or can use non-perplexity-based calibrations.

To justify our intuitive explanation in terms of attractive and repulsive forces, we can rewrite the loss function as follows:

$$\mathcal{L} = \sum_{i,j} p_{ij}\log\frac{p_{ij}}{q_{ij}} = \text{const} - \sum_{i,j} p_{ij}\log q_{ij},$$

and dropping the constant,

$$-\sum_{i,j} p_{ij}\log\frac{w_{ij}}{Z} = -\sum_{i,j} p_{ij}\log w_{ij} + \sum_{i,j} p_{ij}\log Z = -\sum_{i,j} p_{ij}\log w_{ij} + \log\sum_{i,j} w_{ij}.$$

To minimise $\mathcal{L}$, the first sum should be as large possible, which means large w_{ij}, i.e. small $\|\mathbf{y}_i - \mathbf{y}_j\|$, meaning an attractive force between points i and j whenever $p_{ij}\neq 0$. At the same time, the second term should be as small as possible, meaning small w_{ij} and a repulsive force between any two points i and j, independent of the value of p_{ij}.

t-SNE Math in nutshell !

Kobak & Berens, 2018

Practical use of t-SNE

- Use PCA initialization (recommended 50 Components) and reduce to 2 dims randomly with co-ordinates drawn from Gaussian distribution with standard deviation = 0.0001.

- Perplexity (attraction to nearest neighbour)- Range [2,50] and typically set to 30/50. Ideally, the t-sne map should be robust to perplexity chosen.

- Iterations = 1000.

- Learning rate = 200.

- Momentum = 0.5 first 250 iterations then 0.8 for remaining.

- Exaggeration: Change in cost function for the attractive term by alpha = 4.

- Seed- Stochastic maps hence fix the seed to reproduce results.

To approximate the t-SNE gradient, we start by splitting the gradient into two parts

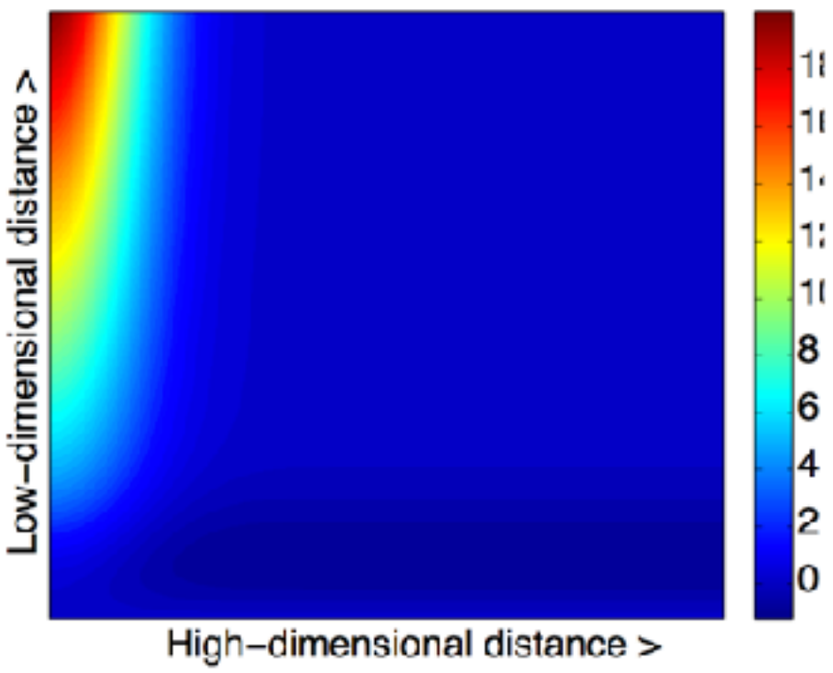
$$\frac{\partial C}{\partial \mathbf{y}_i} = 4(F_{attr} + F_{rep}) = 4\left(\sum_{j \neq i} p_{ij} q_{ij} Z(\mathbf{y}_i - \mathbf{y}_j) - \sum_{j \neq i} q_{ij}^2 Z(\mathbf{y}_i - \mathbf{y}_j)\right), \qquad (3)$$

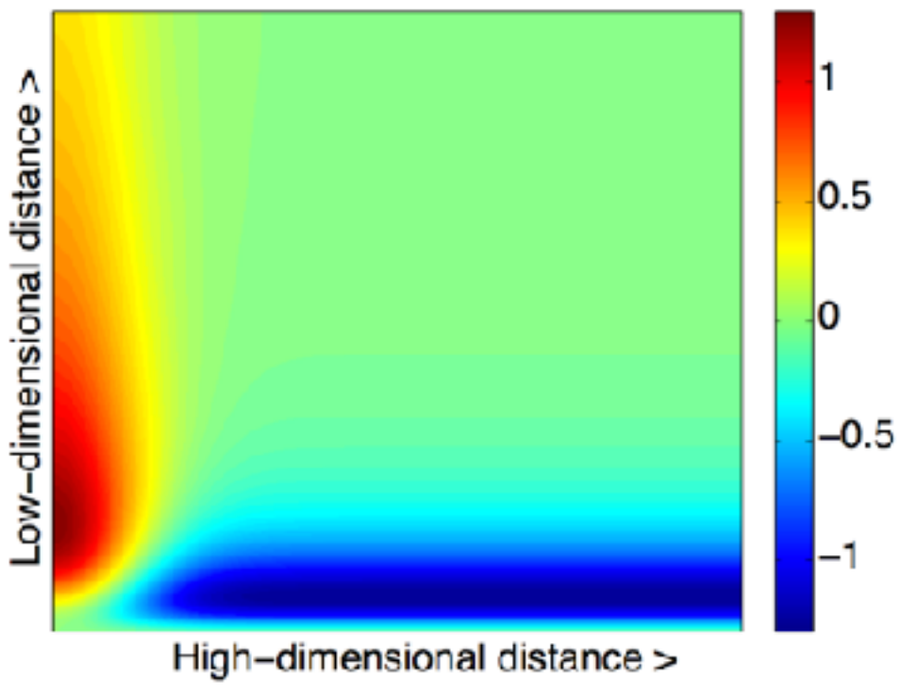
PCA vs. t-SNE

- PCA is linear transformation. / non-linear
- Can be used for Visualization + feature engineering./ only visualization
- Preserves global order (t-SNE doesn’t).
- Fails to embed local structure/ (t-SNE does).
- Example- Swiss-roll (global distances have no meaning,PCA fails but local distances more useful thus t-SNE is more useful.)

SNE vs t-SNE

- Overcrowding at center of map.
- T-distribution allows better local embedding.
- Symmetric version of SNE used - which takes care of outliers by giving equal weightage to all points (repulsive forces)
- SNE cost function is difficult to optimize.





Physical intutions

- Perplexity - 3 times #No. of neighbors used for calculating the attractive term. [Choice of σ_i^2]
- Original manifold: Closer points - $p_{j|i}$ high and further points $p_{j|i}$ low.
- Reduced Manifold: closer points are in denser clusters and Sparse clusters in less dense clusters ?
- Scaled distances - So density of cluster should not matter.
- Size of cluster in t-sne or local distances between clusters have no meaning.
- .. more ?

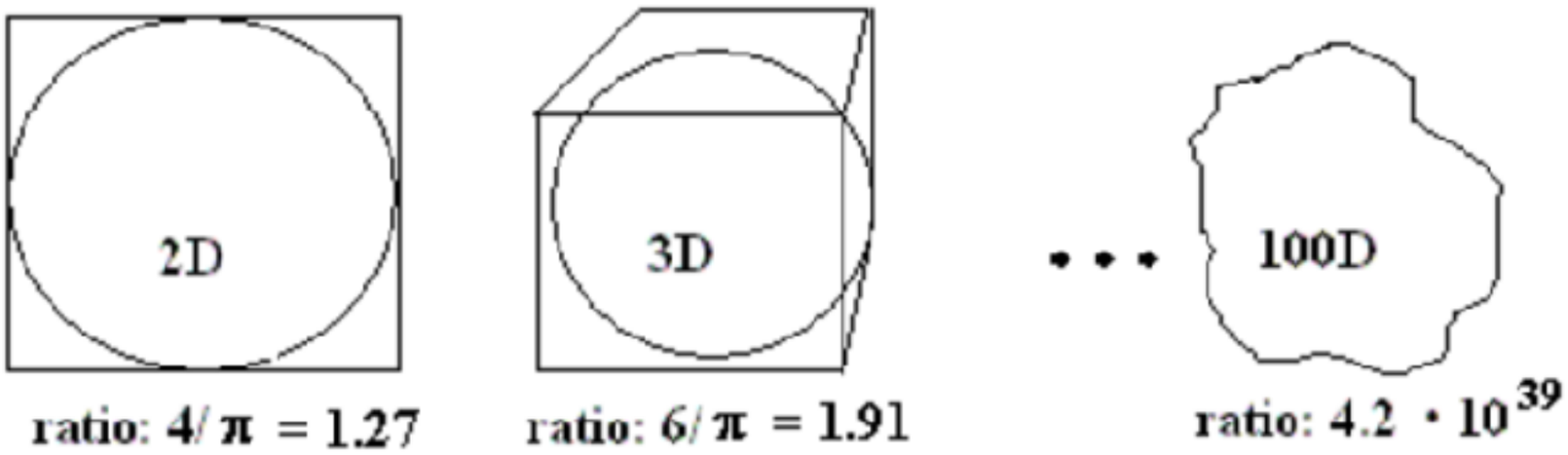
Weakness of t-SNE-1

- Can only be used to reduce to d = 2/3 dimensions. Hence cannot be used more generally due to heavy tails of t-distributions which may comprise large probability mass in higher D and thus loss of local embedding efficiency. Solution: Use t-distributions with higher degree of freedom.

Spiky Spheres in high-dimensions

Most high dimensional data stay at much less intrinsic dimensional space - this is why t-SNE/such methods work.

Physicists are aware of this aspect and call it the “spiky sphere” behavior.



Curse- of dimensionality- data stays in the close to the corners of they hyper cube in N dimensions.

https://www.cs.auckland.ac.nz/courses/compsci708s1c/lectures/Glect-html/topic5c708FSC.htm

Weakness of t-SNE-2

- Local linearity assumption- when does not hold true. E.g. Images of faces in 100 dims or Genomics data over 1000 genes (dims) for each data (Cell).

- Using auto encoders before using t-SNE because they can learn highly varying manifolds better than a local method.

- By definition, it is not possible theoretically to reduce high intrinsic dim objects to 2/3 dims. So, cut your losses and bad luck with visualizing such beasts :)

Weakness of t-SNE

- t-SNE cost function in Non -Convex unlike other methods like LLE, Isomap, D-Maps etc.

- Thus optimization parameter choice influences the local minima search and its a stochastic search so no guarantee you will reach global minima but.…

- However, using the prescribed list of hyper-parameters- its shown various seemingly different datasets can be visualized effectively.

- So authors conclude-rather than convex methods producing “worse visualizations” its better to have t-SNE with its optimization parameter producing better visualizations.

Technical Pitfalls of t-SNE

- Intrinsic higher dimensions for the complex data. (Use Autoencoders before using the output layer for t-SNE exploration)
- Fails to preserve global order. Relative positions of clusters have no physical interpretation.
- Closely spaced points in original space are closer in final map (Overcrowding) but distant points are less distant. [Has to do with the K-L Cost function penalizing $p_{j|i}$ high (close points) more than $p_{j|i}$ low.]
- t-SNE is slow for N> 10^3 and impractical for N>10^4 . O(N^2) in original prescription. (See, Haaten 2014)
- The exact t-SNE computes N^2 similarities p_{ij} and N^2 pairwise attractive and repulsive forces on each gradient descent step. This becomes infeasible for N » 10 000. 2 suggestions were made: (a) For perplexity P, use k= 3P nearest neighbors for the attractive forces to set $p_{j|i}$ = 0 for N-3P (later approximate nearest neighbor algorithms were used without loss of effectiveness and made it faster. (b) For the repulsive forces, BH approx. works as O(N log N)is used for $N \sim 10^5$
- Most recent implementations: Fit-SNE uses a FFT that works upto a million points. O(n)

Papers to look out for!

2004

Stochastic Neighbor Embedding

Geoffrey Hinton and Sam Roweis
Department of Computer Science, University of Toronto
10 Kirg's College Road, Toronto, M5S 3G5 Canada
{*hinton,roweis*}*@cs.toronto.edu*

2008

Journal of Machine Learning Research 9 (2008) 2579-2605 Submitted 5/08; Revised 9/08; Published 11/08

Visualizing Data using t-SNE

Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

2014

Journal of Machine Learning Research 15 (2014) 3221-3245 Submitted 6/13; Published 10/14

Accelerating t-SNE using Tree-Based Algorithms

Laurens van der Maaten LVDMAATEN@GMAIL.COM

2017

EFFICIENT ALGORITHMS FOR T-DISTRIBUTED STOCHASTIC NEIGHBORHOOD EMBEDDING

GEORGE C. LINDERMAN, MANAS RACHH, JEREMY G. HOSKINS,
STEFAN STEINERBERGER, AND YUVAL KLUGER

http://scikit-learn.org/stable/modules/manifold.html#t-sne

2018

Distill

How to Use t-SNE Effectively

Although extremely useful for visualizing high-dimensional data, t-SNE plots can sometimes be mysterious or misleading. By exploring how it behaves in simple cases, we can learn to use it more effectively.

The art of using t-SNE for single-cell transcriptomics

Dmitry Kobak[1] and Philipp Berens[1]

Outlook

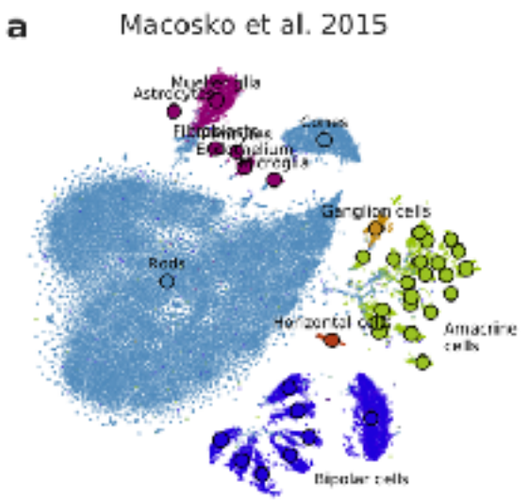
- Retaining- both local and global structure is hard while simultaneously reducing the dimensions of the data.

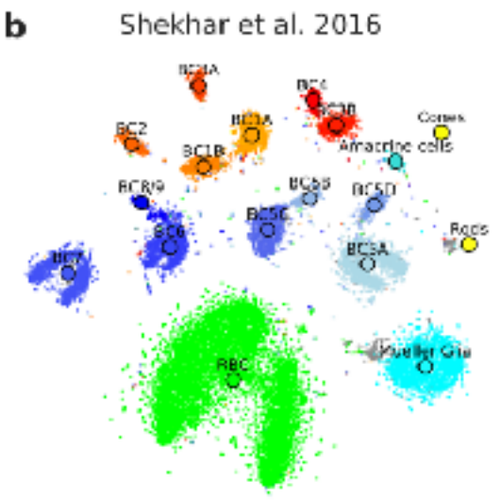
- Honestly, it is impossible is the data is really high dimensional.

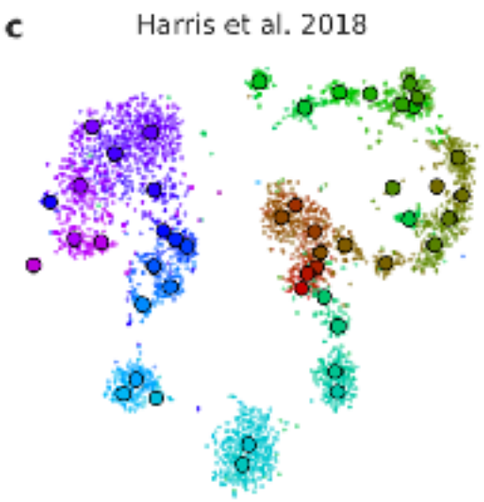
- Strategy is change distance metric first in linear methods, then in non-linear methods. t-SNE for example uses simple Eucledian metric + Non-linear map but there is no functional map. Thus no feature engineering is possible.

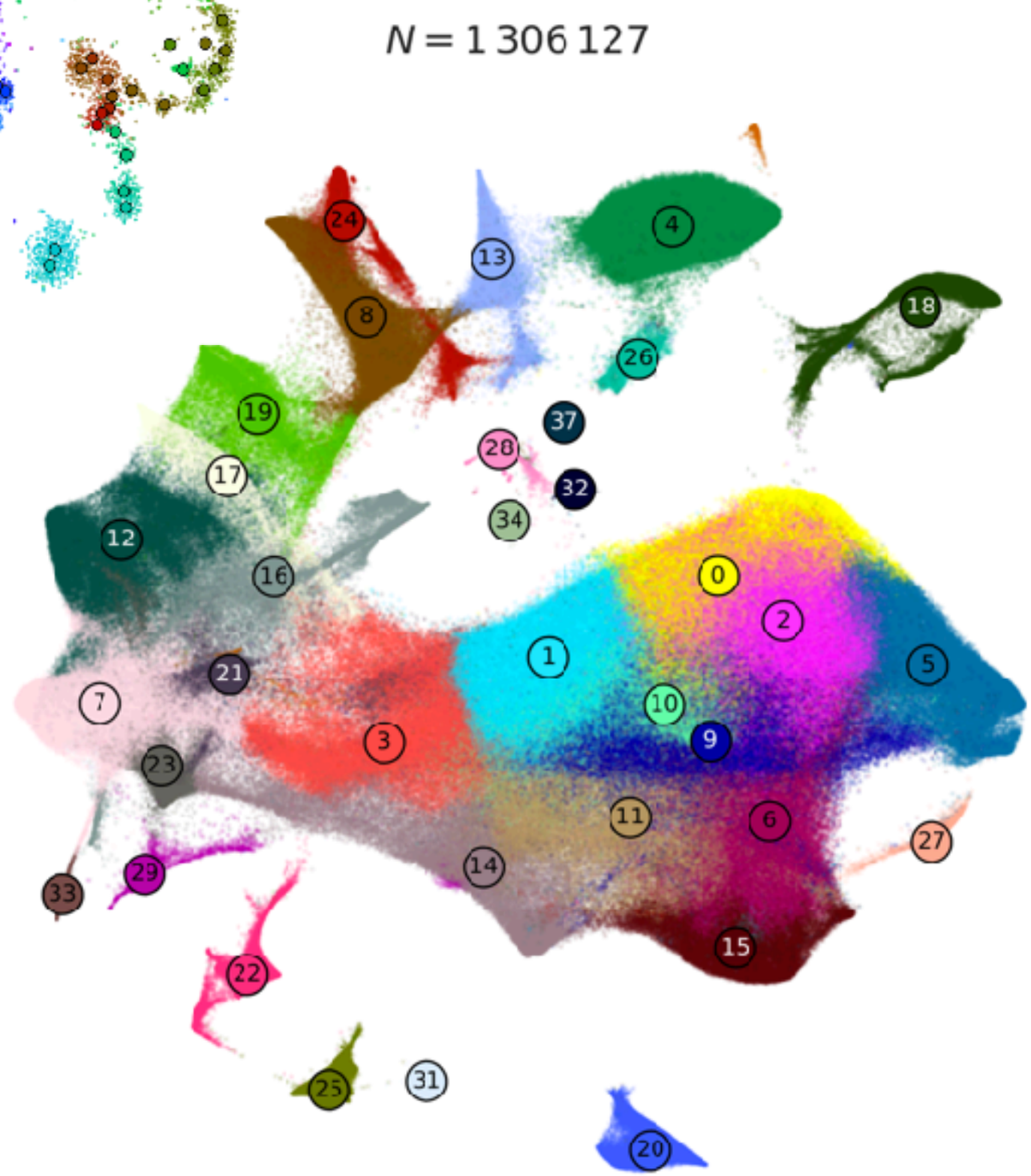
- Future: UMAP's- Changing distance metric as a function of varying manifold and allows feature extraction. Better local vs global order balance.

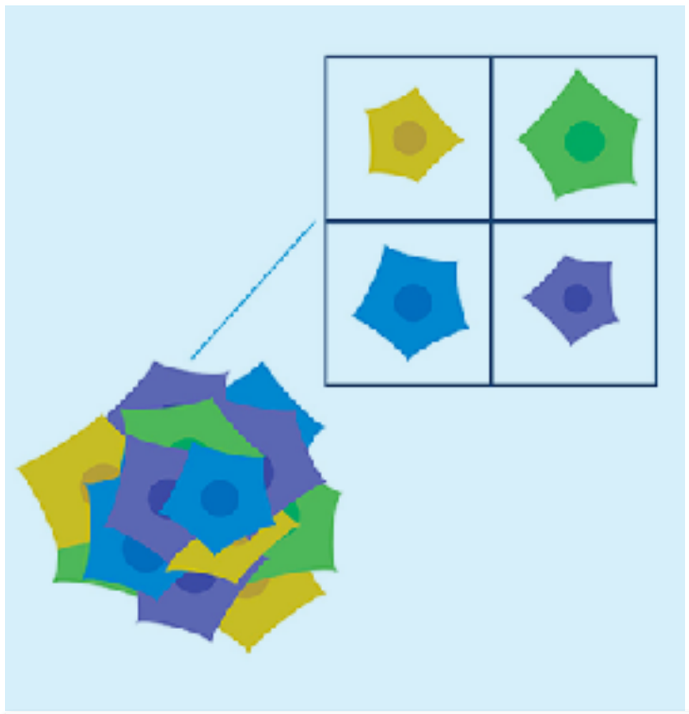
Genomics











https://www.10xgenomics.com/solutions/sir

t-SNE is a fancy MAthematicianN's Scatter Plot !

- Thank you!