Fundamentals of AI

Manifold learning and non-linear dimred

Self-organizing maps (SOM)
Principal Curve
Principal manifold
Principal graphs

Injective methods

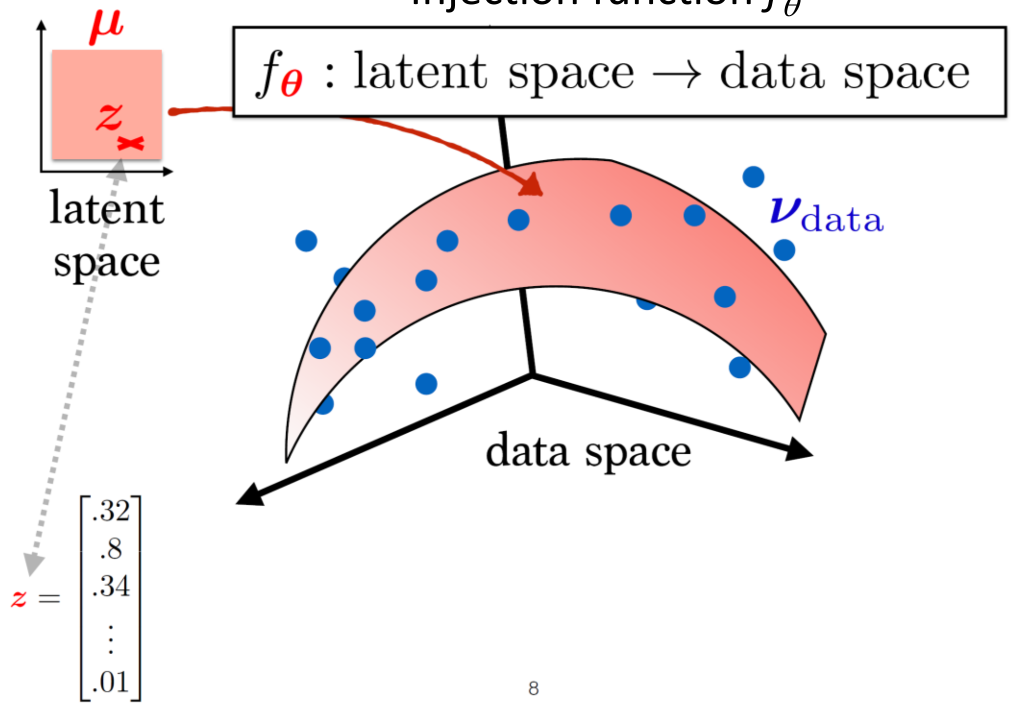
Manifold appears explicitly

Manifold can be analysed independently on the data (for example, can be colored to visualize any function in the data space)

Can be used to generate new data

Classical methods are limited to *m*=1,2,3,4

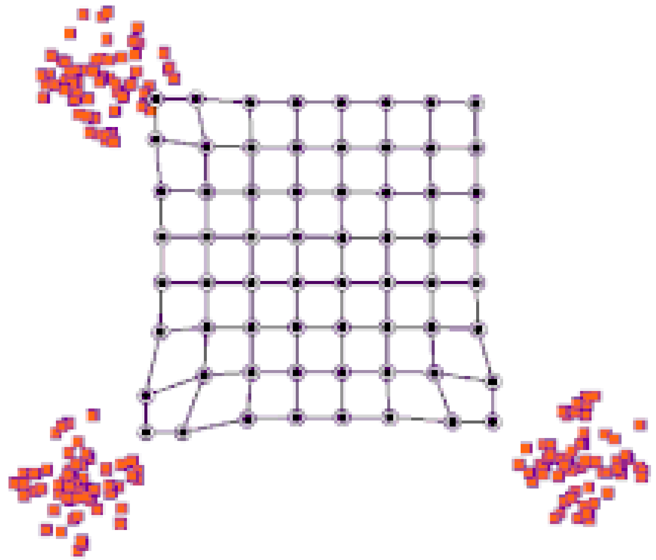
Injection function f_θ



From https://marcocuturi.net/

Self-organizing Map (SOM)

- Introduced by Teuvo Kohonen in 1982 as a type of (unsupervised) artificial neural network
- Sometimes considered as a clustering technique
- ‘Clusters’ are organized in a regular grid
- Main application – data visualization
- Approximates principal manifold
- SOM is not an optimization-based method! (it does not optimize any function)



Self-organizing Map (SOM) algorithm

1) Randomize the node weight vectors in a map

2) Randomly pick an input vector x_i

3) Traverse each node in the map

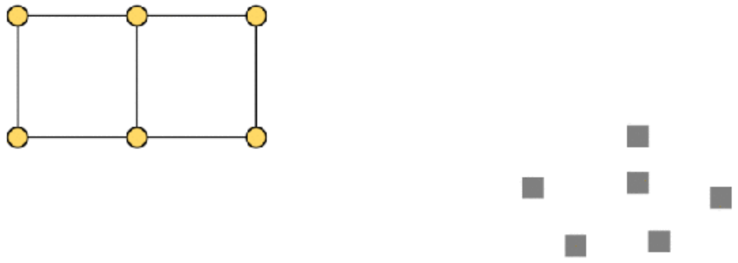
- Use the Euclidean distance formula to find the similarity between the input vector and the map's node's weight vector
- Track the node ***u*** that produces the smallest distance (this node is the best matching unit, BMU)

$$u = arg\min_{v} ||x_i - W_v||$$

4) Update the weight vectors of the nodes in the neighborhood of the BMU (including the BMU itself) by pulling them closer to the input vector

$$W_v \leftarrow W_v + \alpha\theta(u,v)\ (x_i - W_v)$$

5) Repeat from step 2 for certain number of iterations (many!)

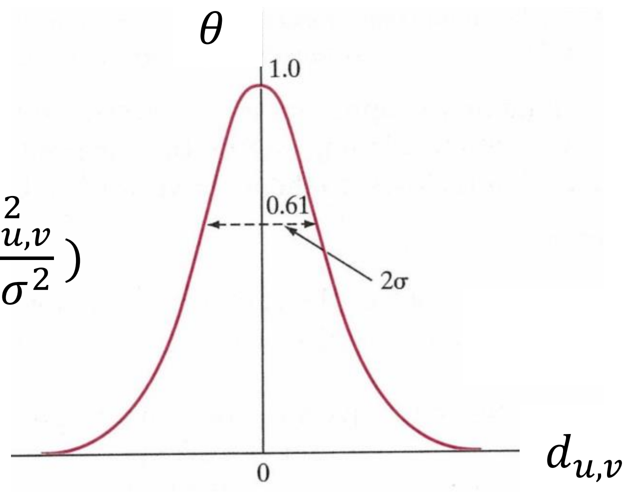


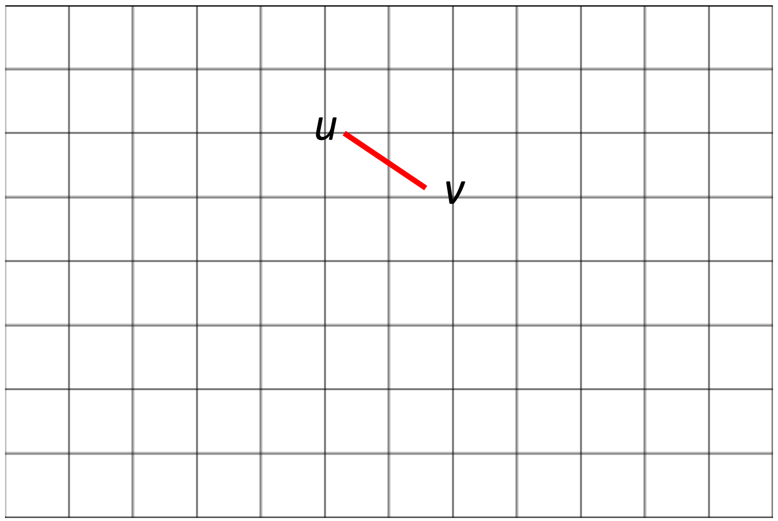
$$W_v \leftarrow W_v + \alpha\theta(u,v)\,(x_i - W_v)$$

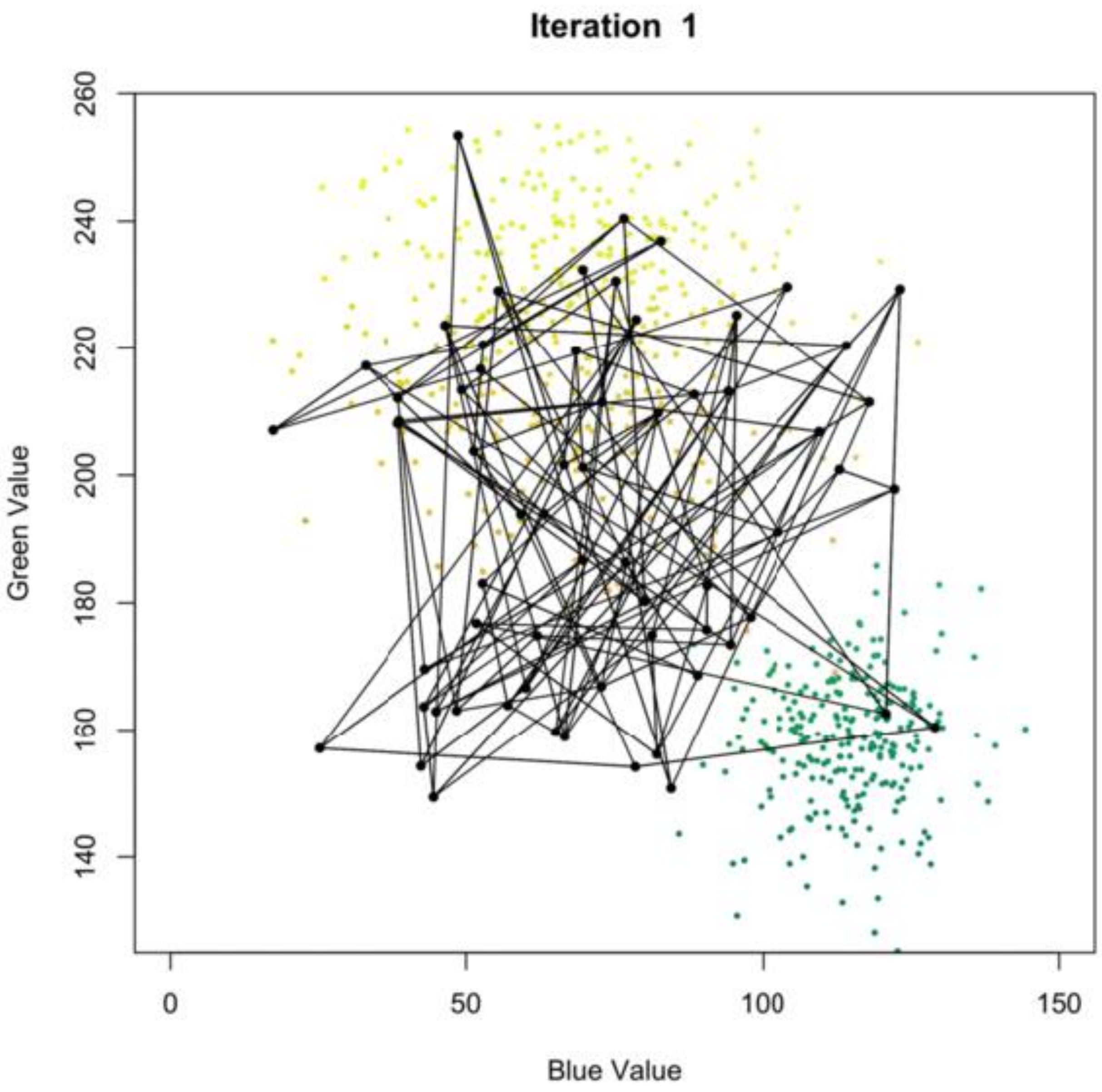
α - learning rate (usually, gradually decreasing)

$\theta(u,v)$ – neighbourhood function on the grid (internal space)

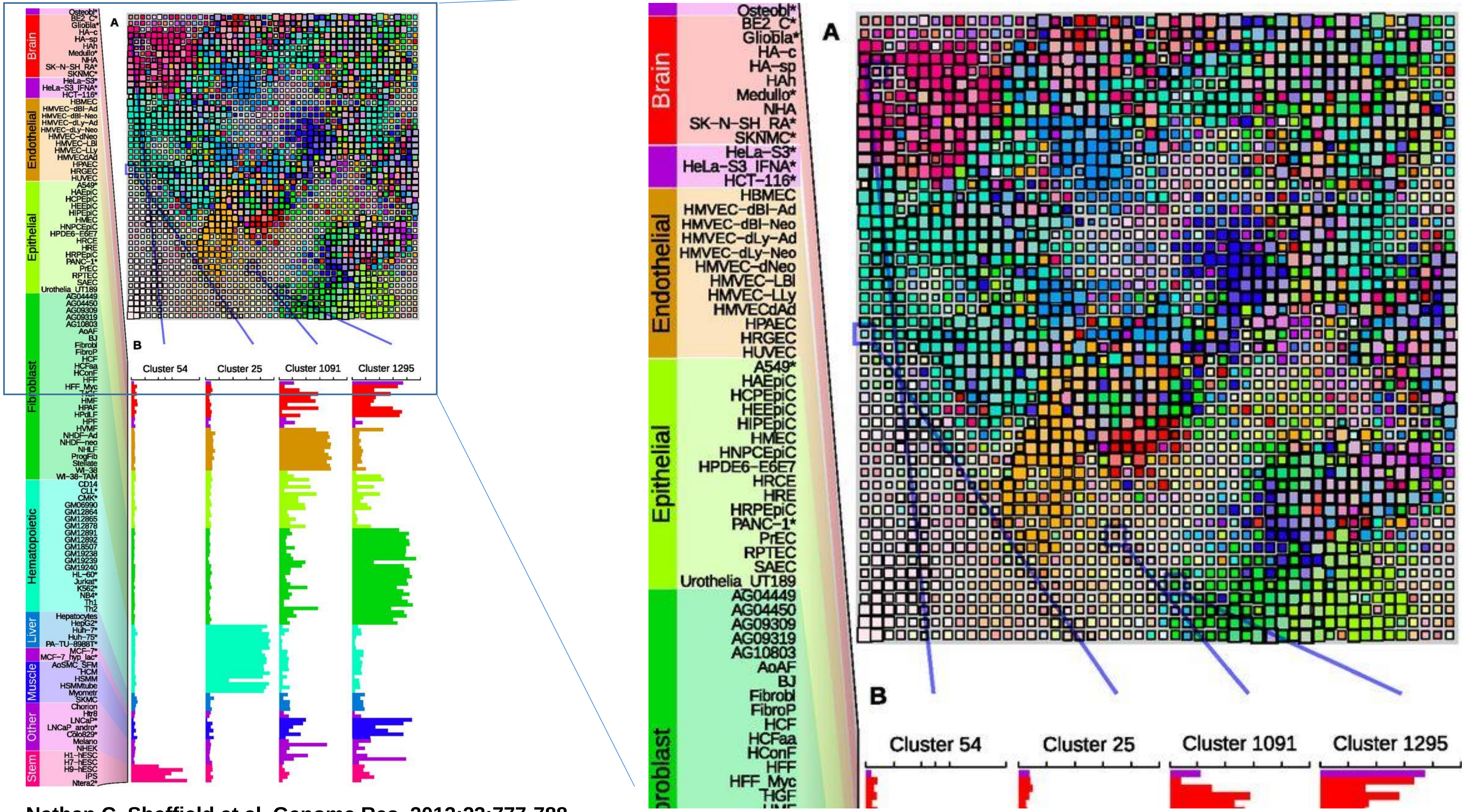
$$\theta(u,v) = \exp\left(-\frac{d_{u,v}^2}{2\sigma^2}\right)$$





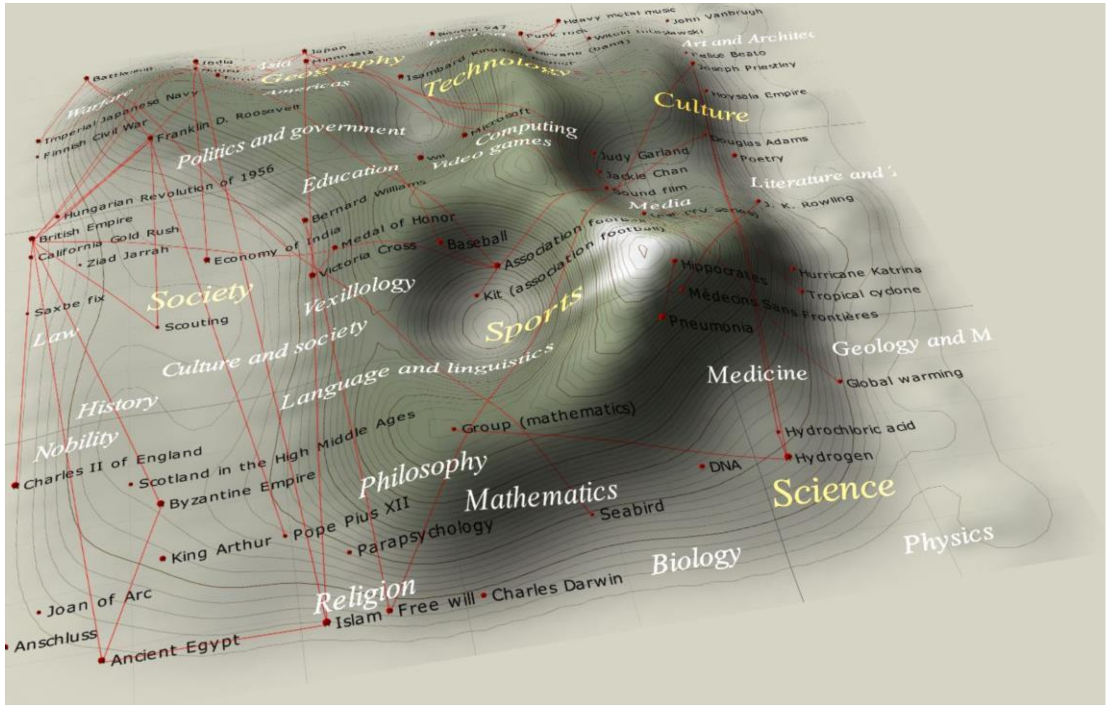

Iteration 1
Green Value
260
240
220
200
180
160
140
Blue Value
0
50
100
150

SOM clustering of DHS profiles.

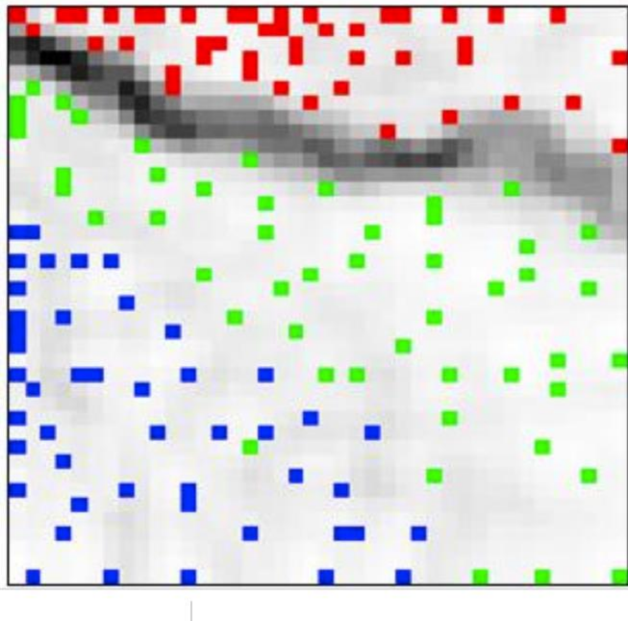


Nathan C. Sheffield et al. Genome Res. 2013;23:777-788

SOM of Wikipedia articles



SOM of the Fisher's Iris Flower Data Set



Hundreds of variants of SOMs

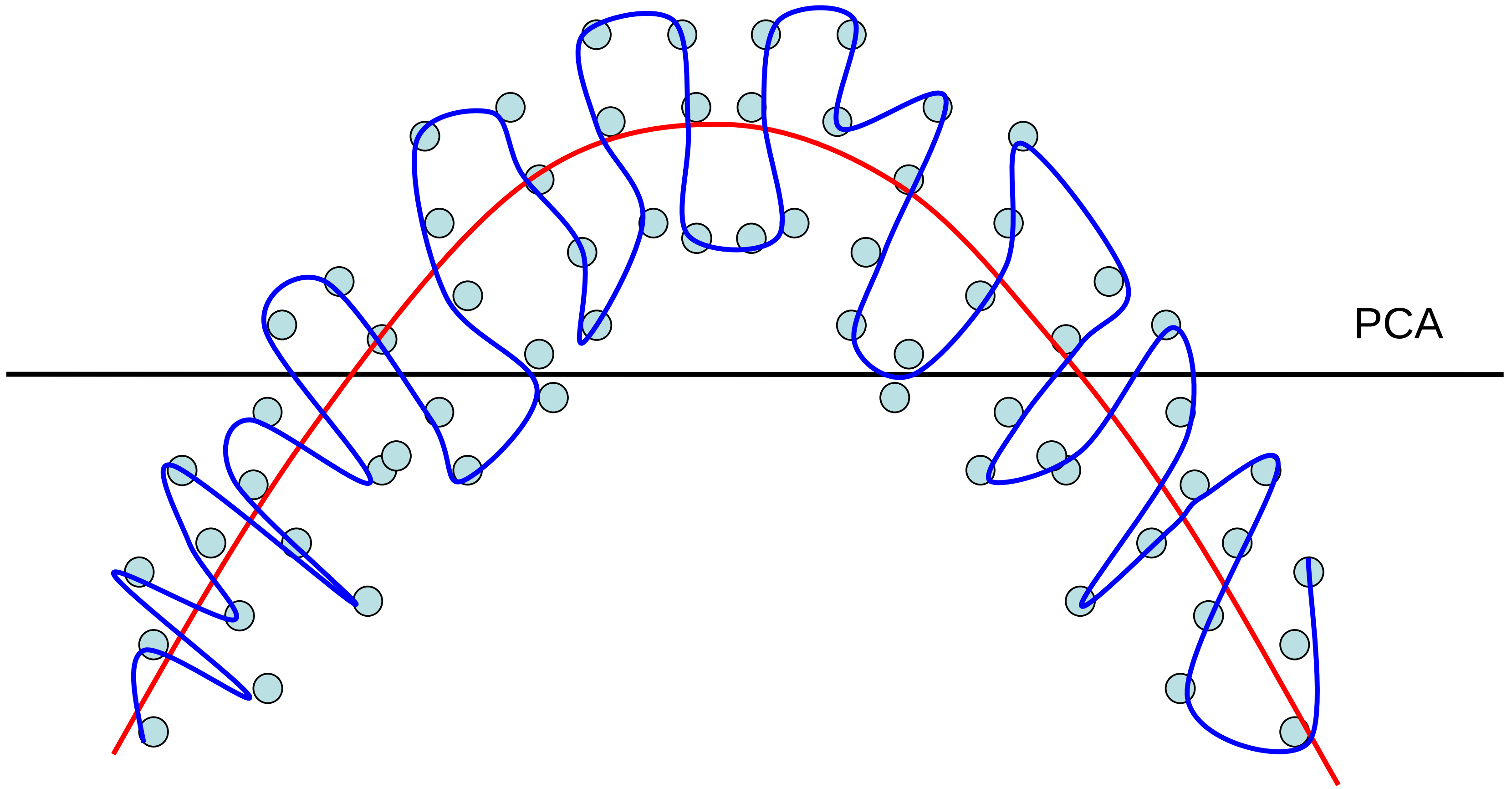
- Batch SOM
- Online SOM
- Growing SOM
- …

Optimization-based generalizations of SOMs:

Generative Topographic Mapping (GTM) – probabilistic approach

Elastic Maps – spline-based approach

Principal Curve: balance between regular properties and approximation accuracy



Principal curve as a non-linear ‘regression’

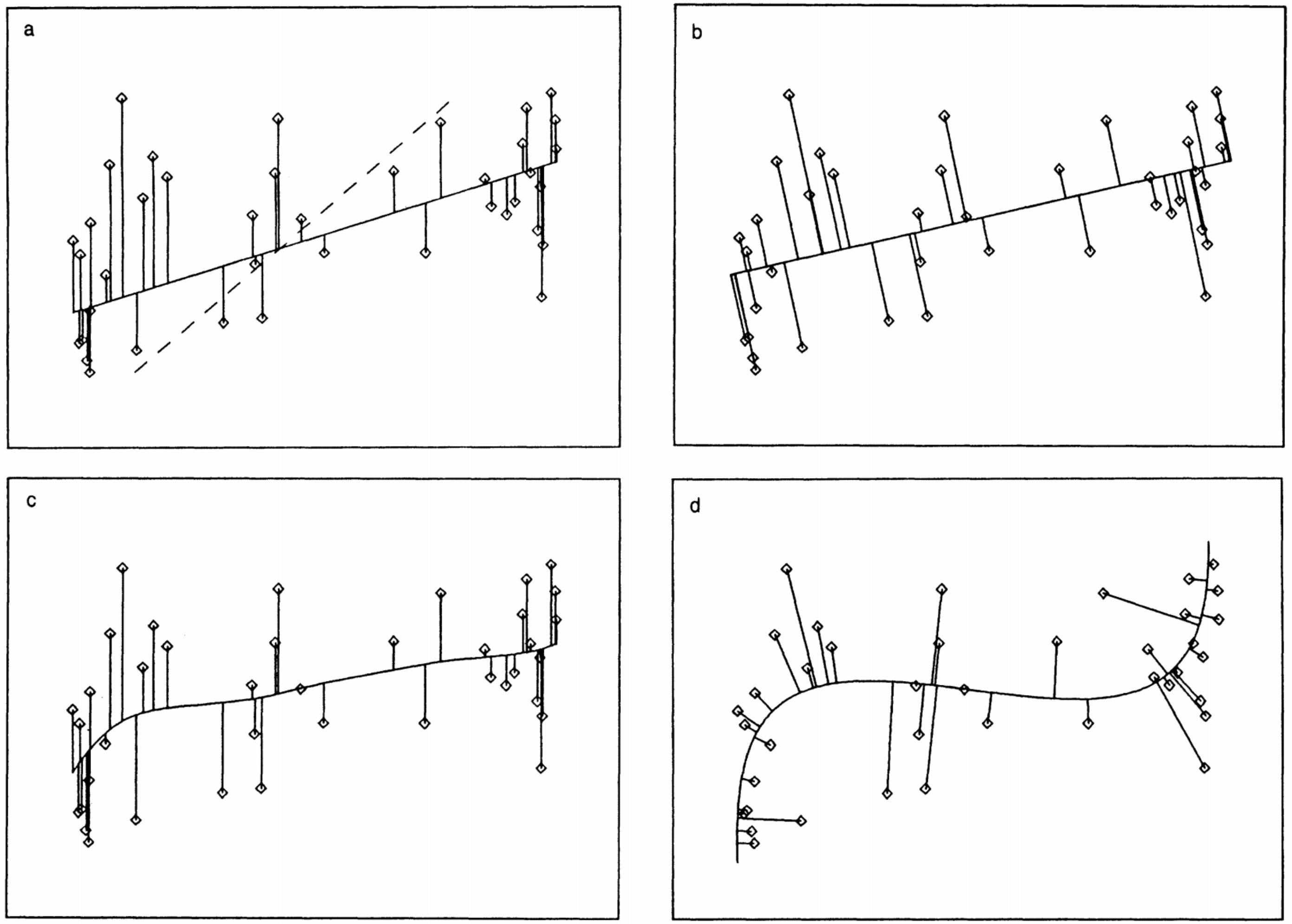
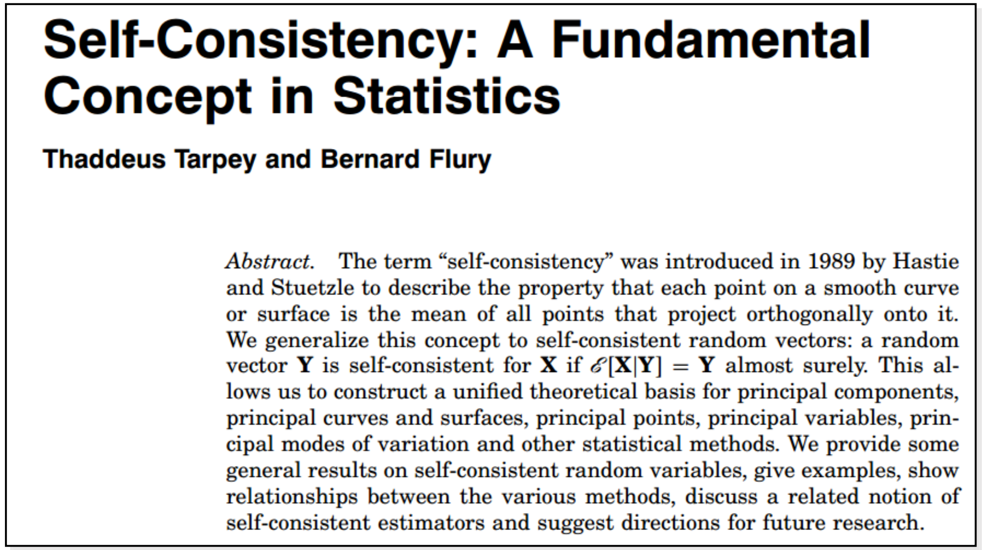


Figure 1. (a) The linear regression line minimizes the sum of squared deviations in the response variable. (b) The principal-component line minimizes the sum of squared deviations in all of the variables. (c) The smooth regression curve minimizes the sum of squared deviations in the response variable, subject to smoothness constraints. (d) The principal curve minimizes the sum of squared deviations in all of the variables, subject to smoothness constraints.

Hastie's principal curves (1989)

- Notion of self-consistency



Self-Consistency: A Fundamental Concept in Statistics

Thaddeus Tarpey and Bernard Flury

Abstract. The term "self-consistency" was introduced in 1989 by Hastie and Stuetzle to describe the property that each point on a smooth curve or surface is the mean of all points that project orthogonally onto it. We generalize this concept to self-consistent random vectors: a random vector $\mathbf{Y}$ is self-consistent for $\mathbf{X}$ if $\mathcal{E}[\mathbf{X}|\mathbf{Y}] = \mathbf{Y}$ almost surely. This allows us to construct a unified theoretical basis for principal components, principal curves and surfaces, principal points, principal variables, principal modes of variation and other statistical methods. We provide some general results on self-consistent random variables, give examples, show relationships between the various methods, discuss a related notion of self-consistent estimators and suggest directions for future research.

Let us denote some set of points **Y** = $\{\boldsymbol{y}_i\} \in R^p$ (can be infinite)

[X|$\boldsymbol{y}_i$] : all data points **X** for which y_i is the closest one among all possible **y**$\in$**Y**

If **E[X|$\boldsymbol{y}_i$] = $\boldsymbol{y}_i$** then y is **self-consistent**

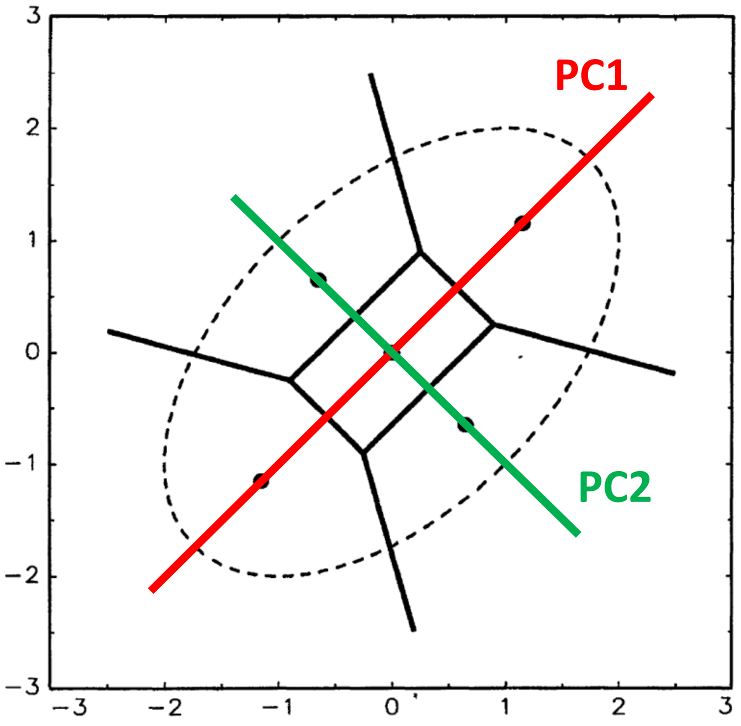


FIG. 6. *k = 5 self-consistent points of the bivariate normal distribution from Figure 2, along with the partition of* $\mathbb{R}^2$ *by domains of attraction of the five points.*

Self-consistent points = principal points
Self-consistent lines = principal components

Hastie and Stuetzle: Principal Curves

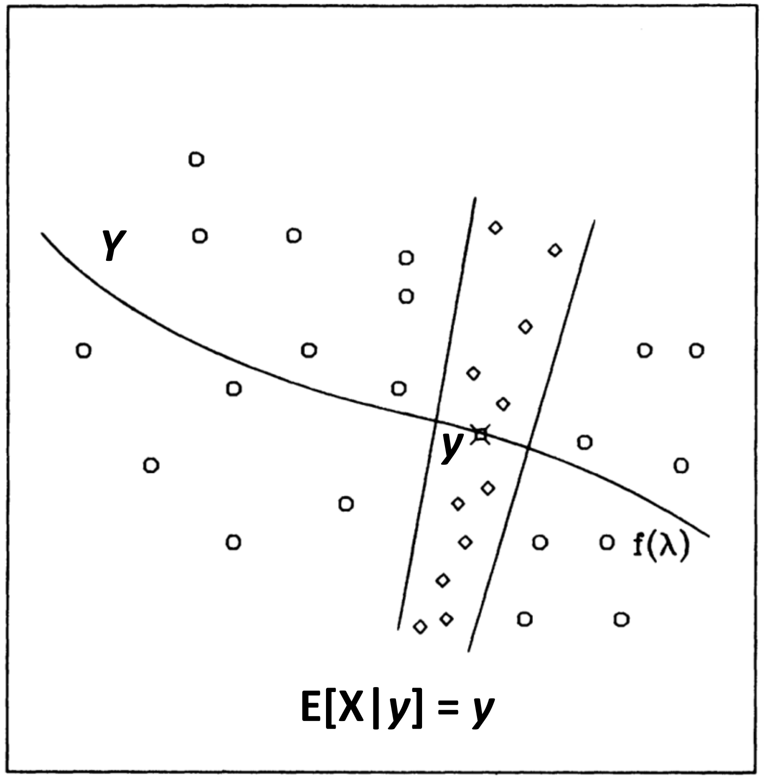


Figure 3. Each point on a principal curve is the average of the points that project there.

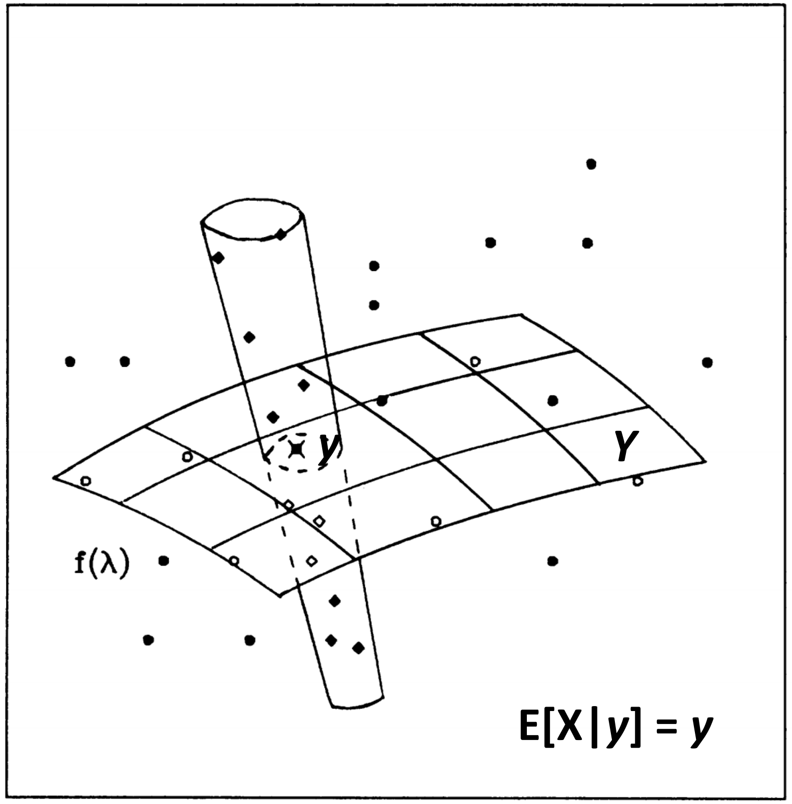
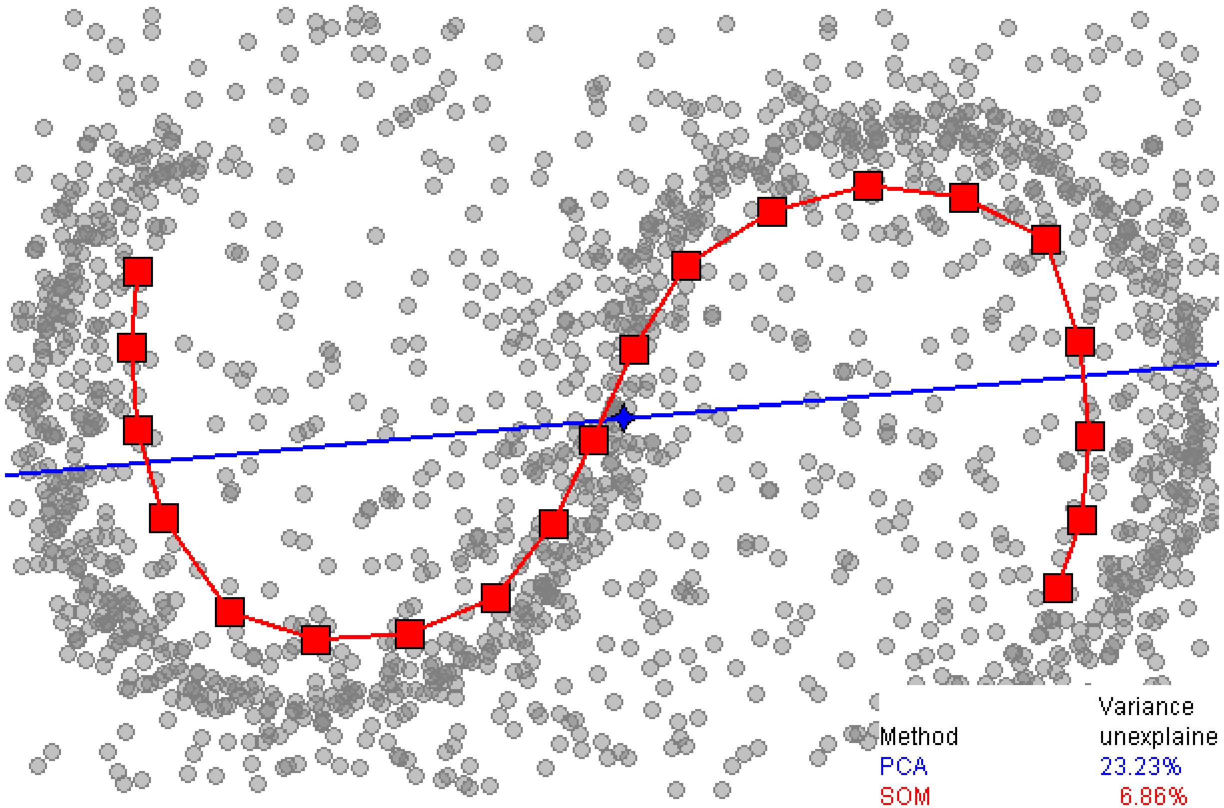


Figure 13. Each point on a principal surface is the average of the points that project there.

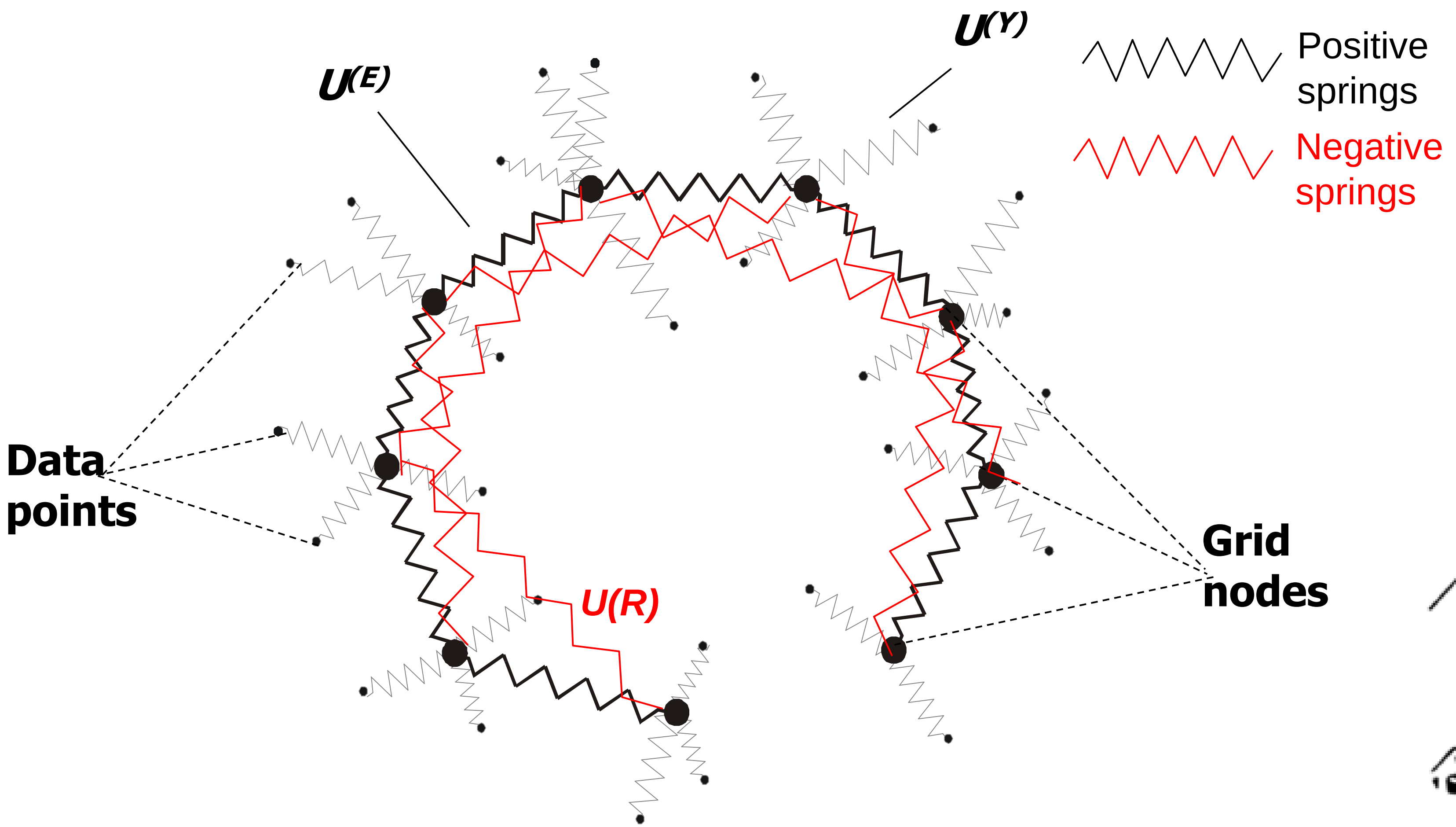
Self-consistency is enough to define the principal curve if we know PDF

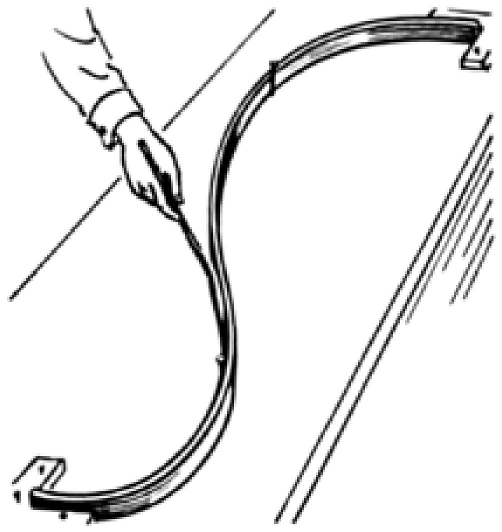
In the case of data point cloud, we need to introduce data 'smoothers' (similar to KDE)

One-dimensional SOM

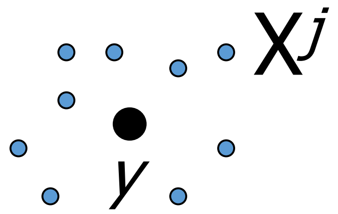


Elastic principal curve (one-dimensional elastic map)

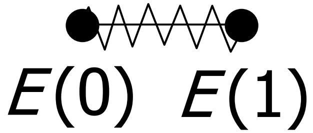




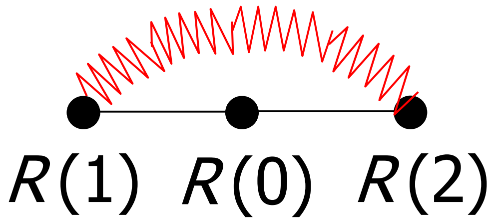
Definition of elastic energy



$$U^{(Y)} = \frac{1}{N} \sum_{i=1}^{p} \sum_{x^{(j)} \in K^{(i)}} \left\| X^{j} - y^{(i)} \right\|^2$$



$$U^{(E)} = \sum_{i=1}^{s} \lambda_i \left\| E^{(i)}(1) - E^{(i)}(0) \right\|^2$$



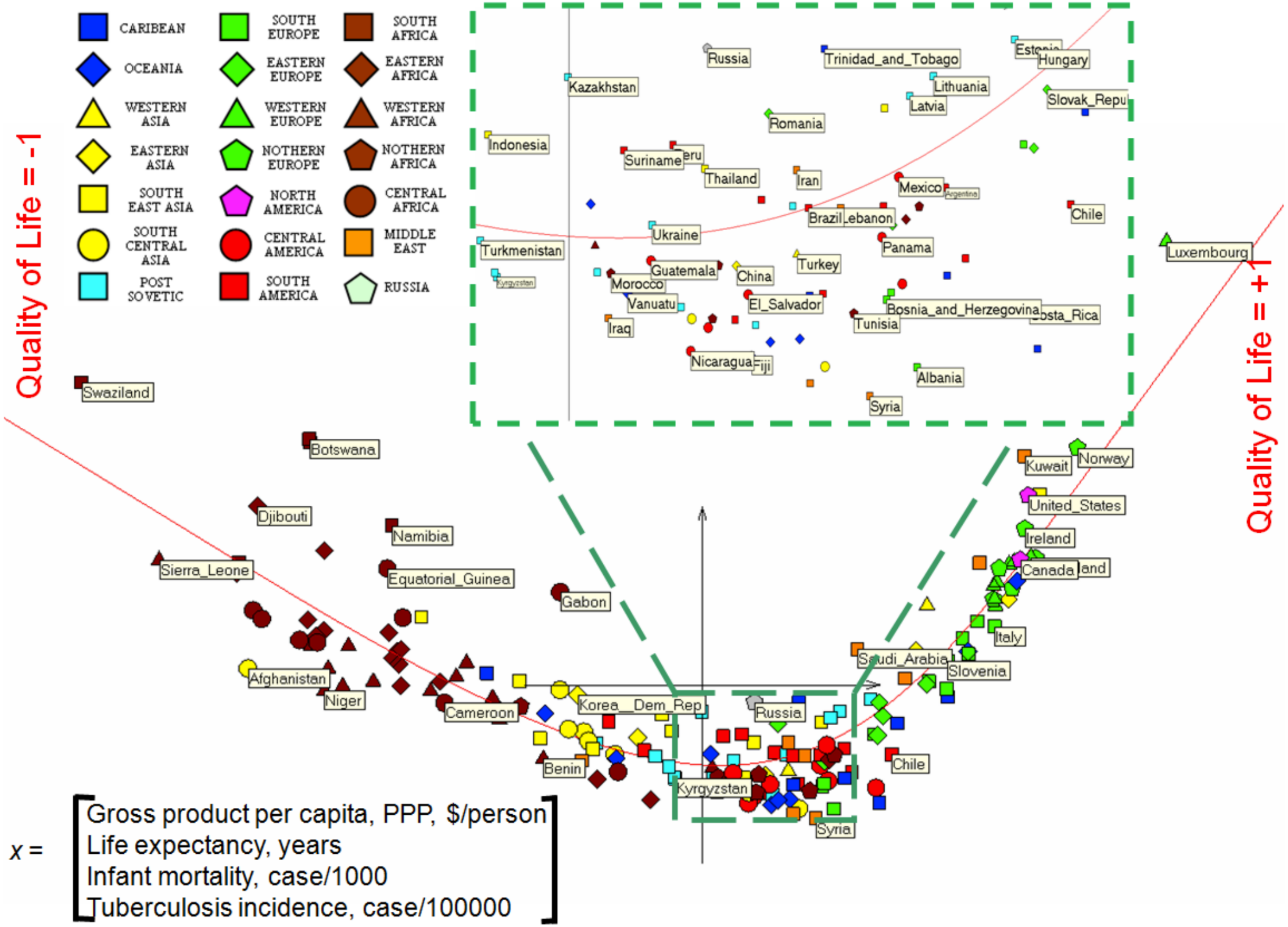
$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| \frac{R^{(i)}(1) + R^{(i)}(2)}{2} - R^{(i)}(0) \right\|^2$$

$$U = U^{(Y)} + U^{(E)} + U^{(R)} \qquad \lambda_i = \lambda_0, \quad \mu_i = \mu_0$$

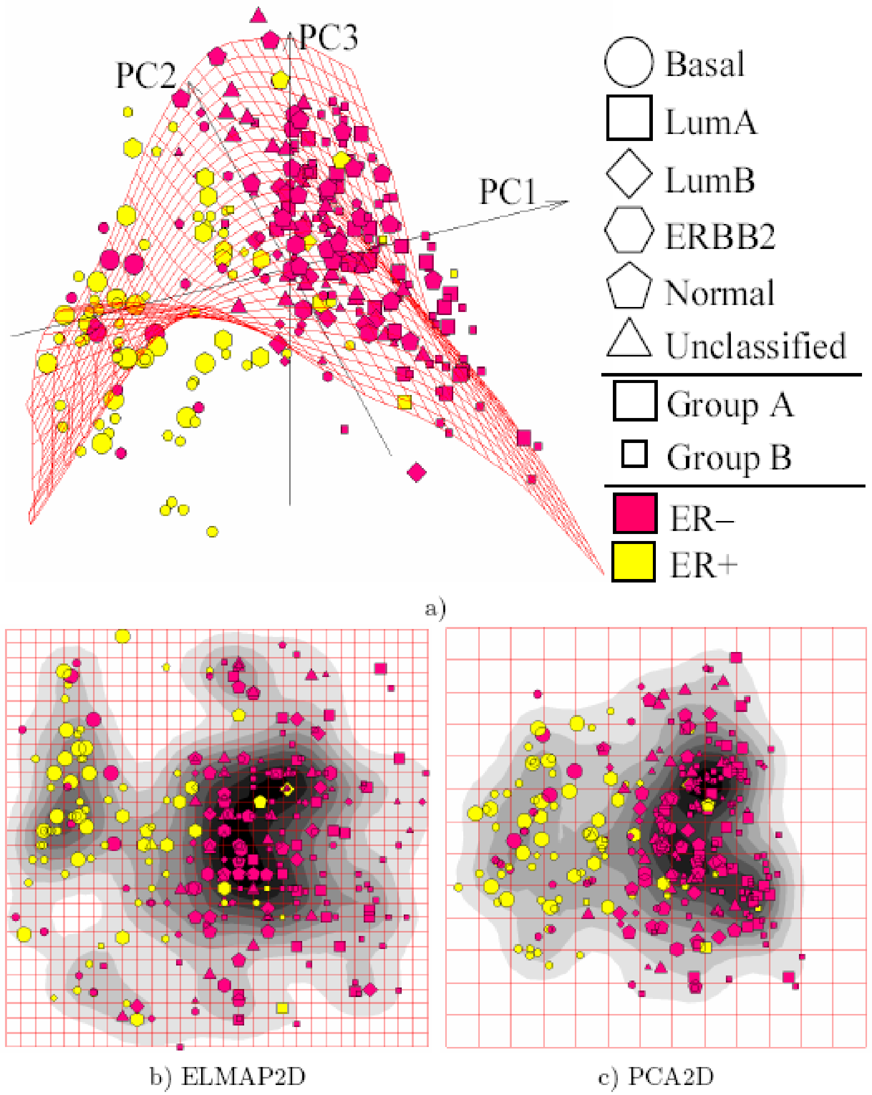
Example of a principal curve

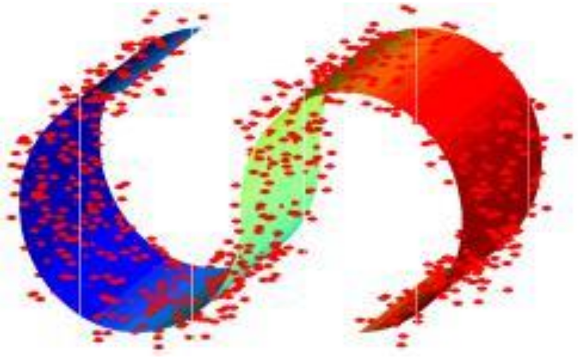
Zinovyev and Gorban, Non-linear quality of life index

https://arxiv.org/ftp/arxiv/papers/1008/1008.4063.pdf



Principal Manifold





What is Principal Manifold

1. Intuitively, a smooth **manifold** going through the middle of data cloud; formally, there exist several definitions for the case of data distributions: 1) Hastie and Stuelze's **principal manifold**s are self-consistent curves and surfaces; 2) Kegl's **principal** curves provide the minimal mean squared error given the limited curve length; 3) Tibshirani's **principal** curves maximize the likelihood of the additive noise data model; 4) Gorban and Zinovyev elastic **principal manifold**s minimize a mean square error functional regularized by addition of energy of **manifold** stretching and bending; 5) Smola's regularized **principal manifold**s minimize some form of a regularized quantization error functional; and some other definitions. Learn more in: Principal Graphs and Manifolds

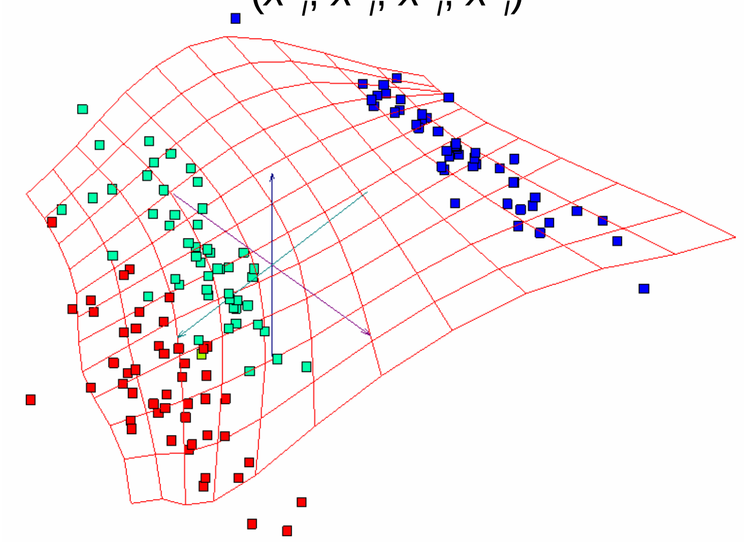
https://www.igi-global.com/dictionary/principal-graphs-manifolds/23373

Visualize any multidimensional function

in any point of the low-dimensional intrinsic space!

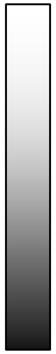
Four dimensional space

$(x^1_i, x^2_i, x^3_i, x^4_i)$

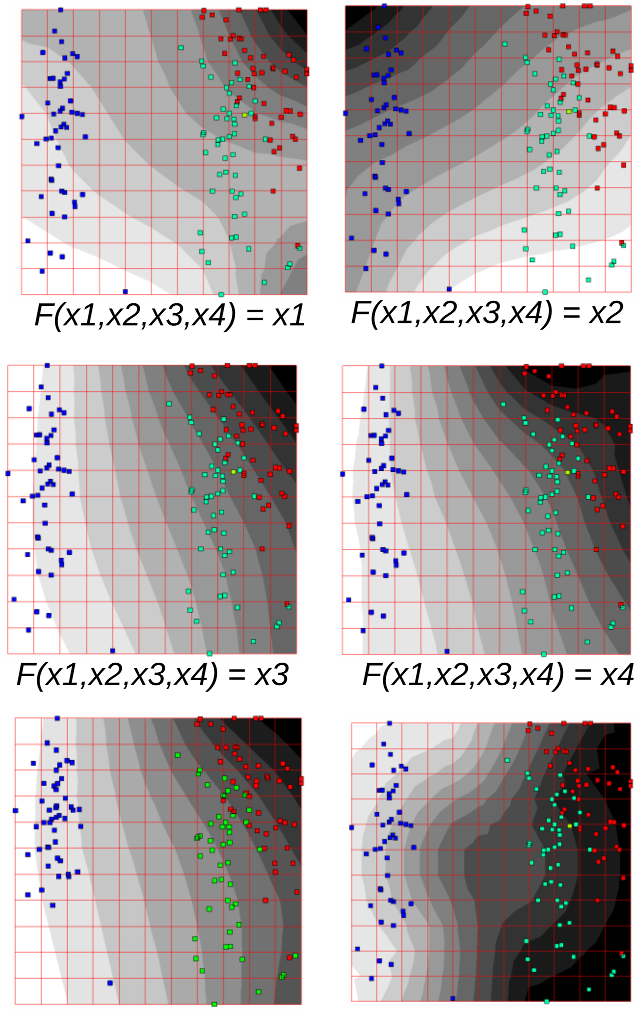


F = F(x1,x2,x3,x4)

max



min



F(x1,x2,x3,x4) = x1

F(x1,x2,x3,x4) = x2

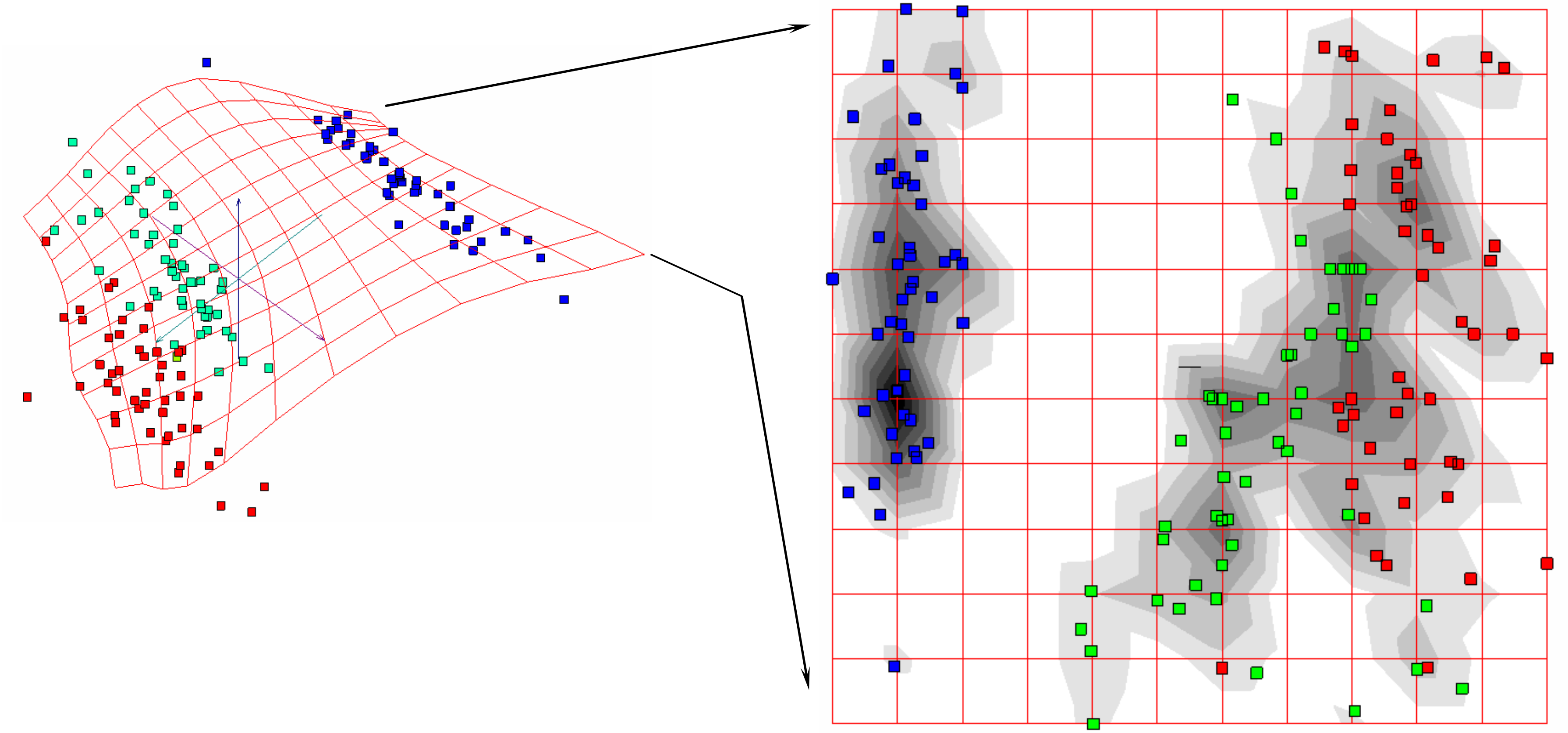
F(x1,x2,x3,x4) = x3

F(x1,x2,x3,x4) = x4

F= -0.4x1+0.3x2-0.5x3-0.5x4

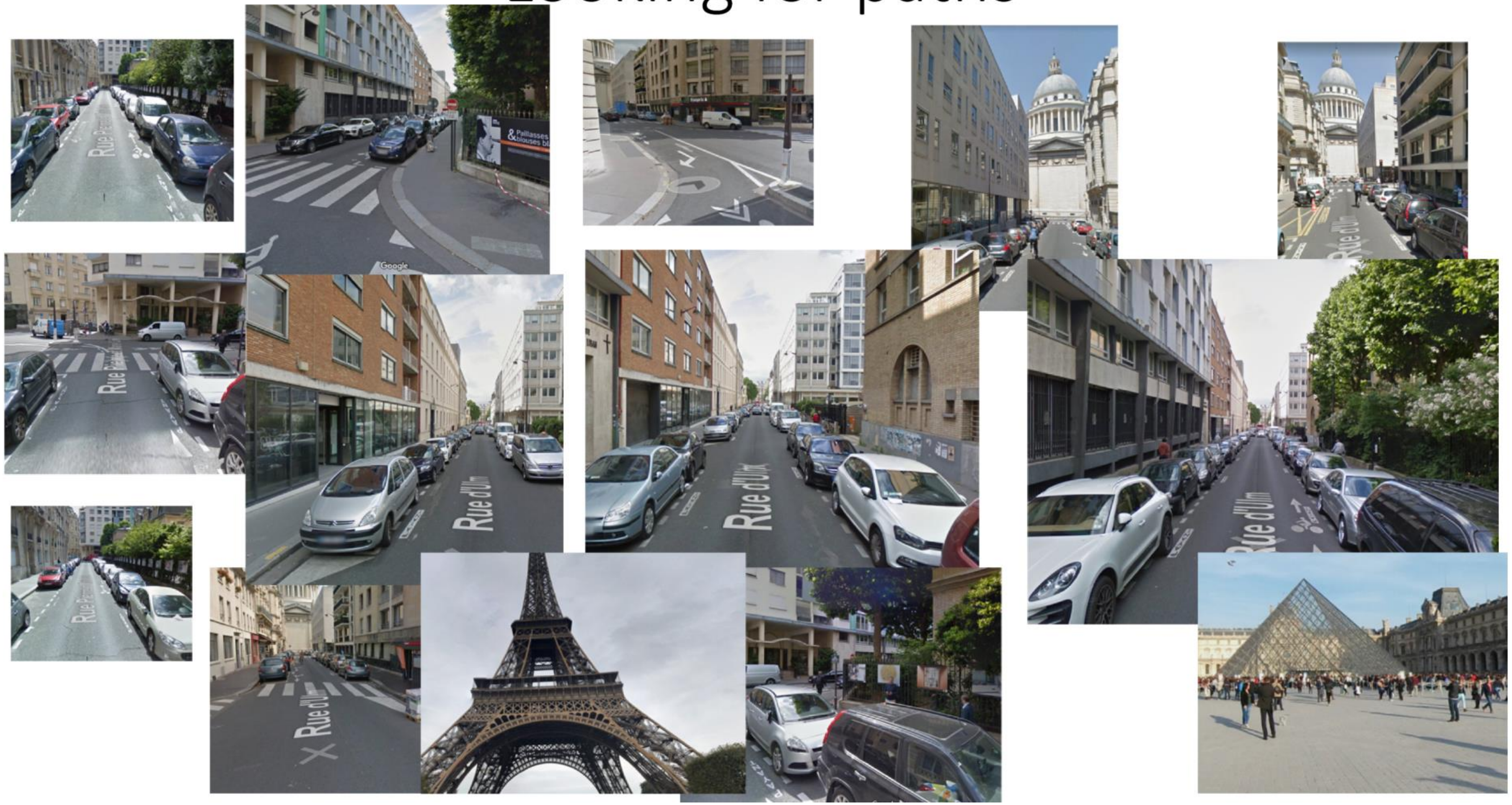
F= $x1^2+x2^2+x3^2+x4^2$

Example of complex function: probability density function estimate



‘Branching’ data

Looking for paths

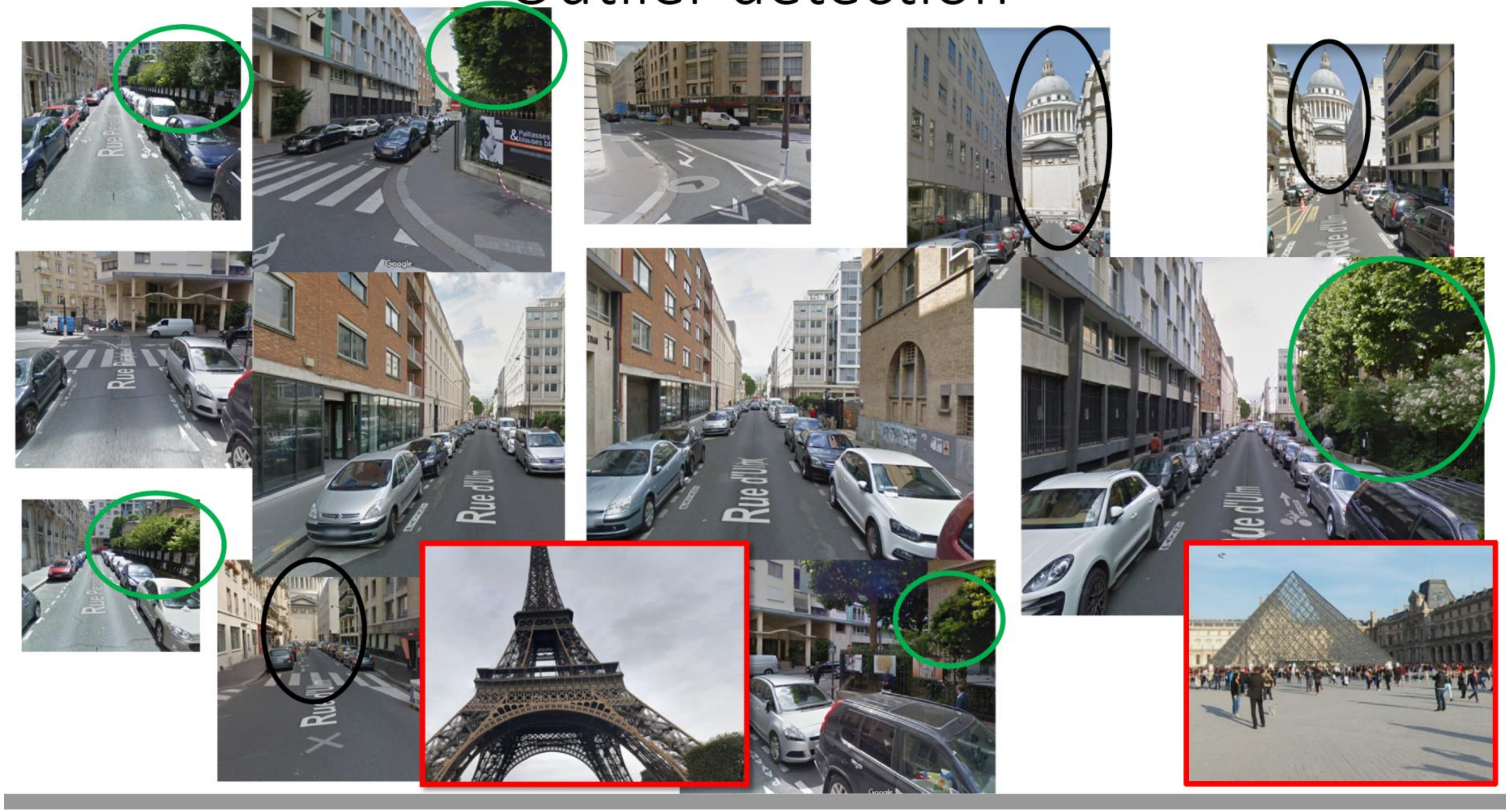


‘Branching’ data

Feature selection

‘Branching’ data

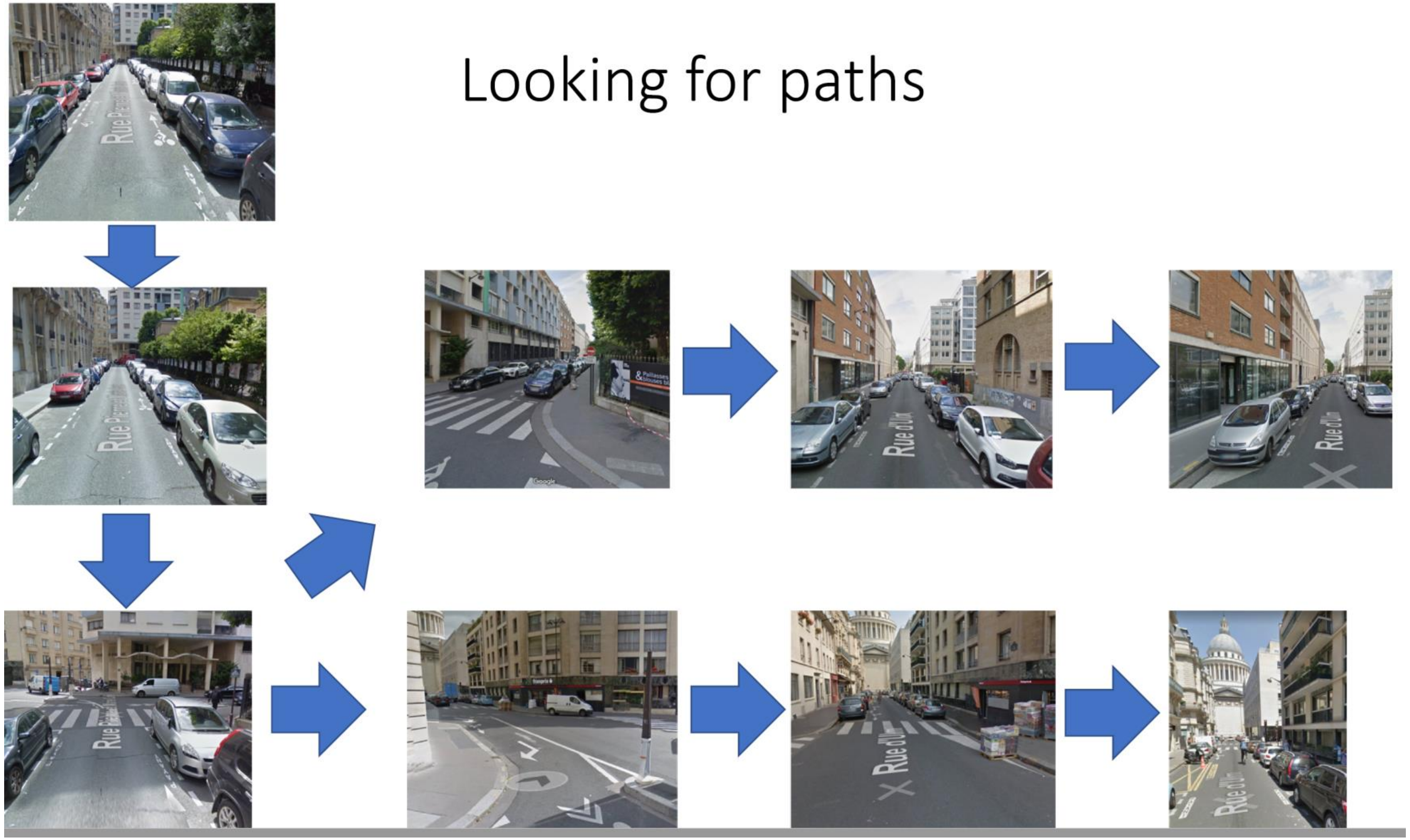
Outlier detection



‘Branching’ data

questions : how many ‘bifurcations’, where the beginning, what is the right sequence?

Looking for paths



Single cell data cartography of Planarian (Plass et al, 2018)

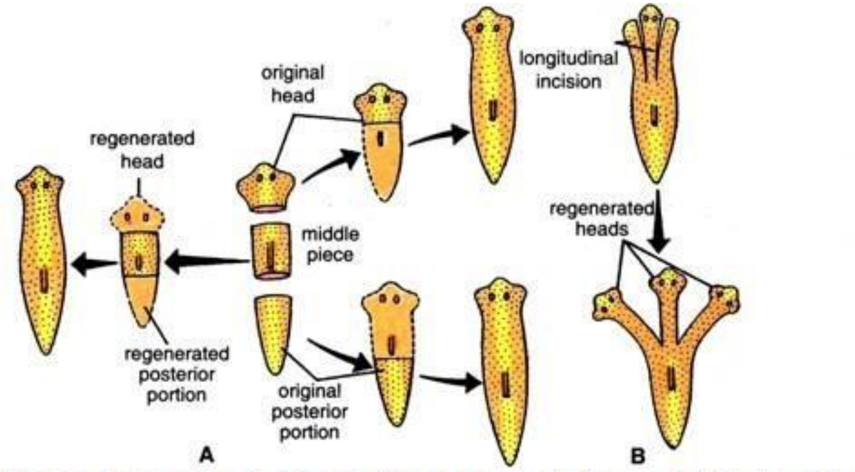
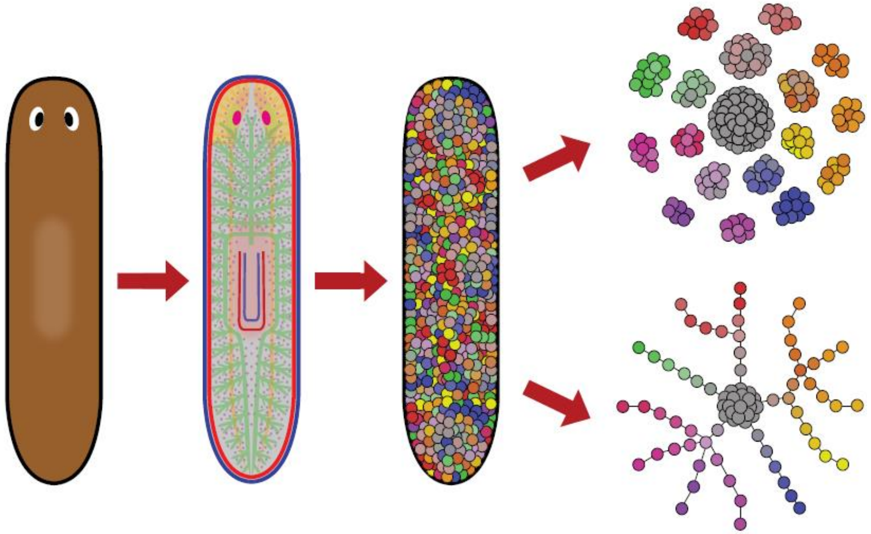
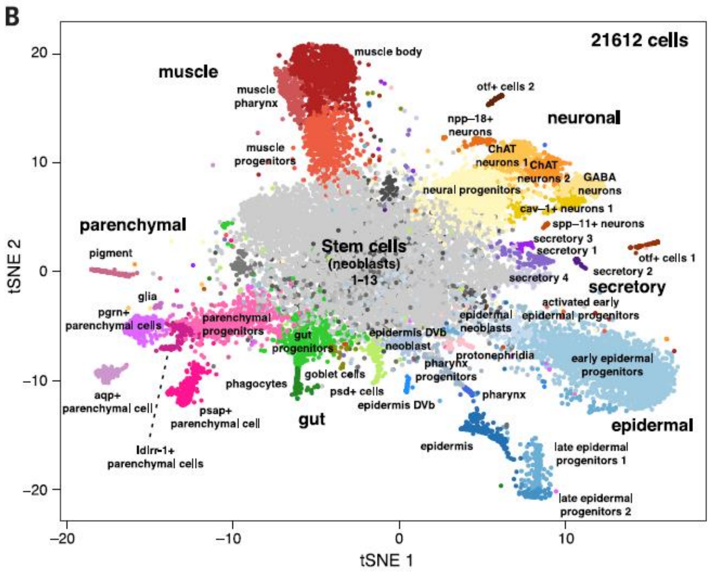
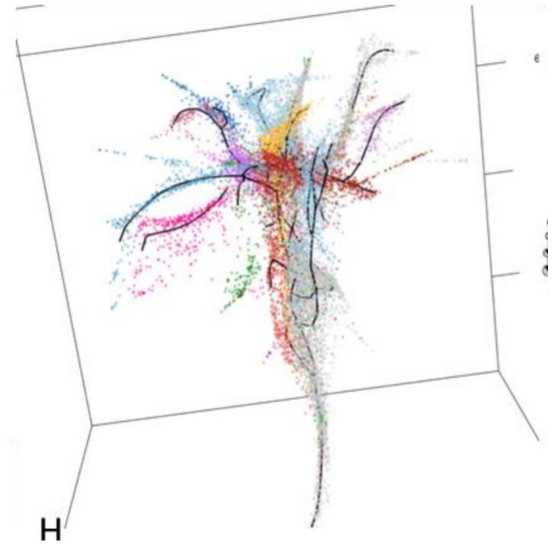


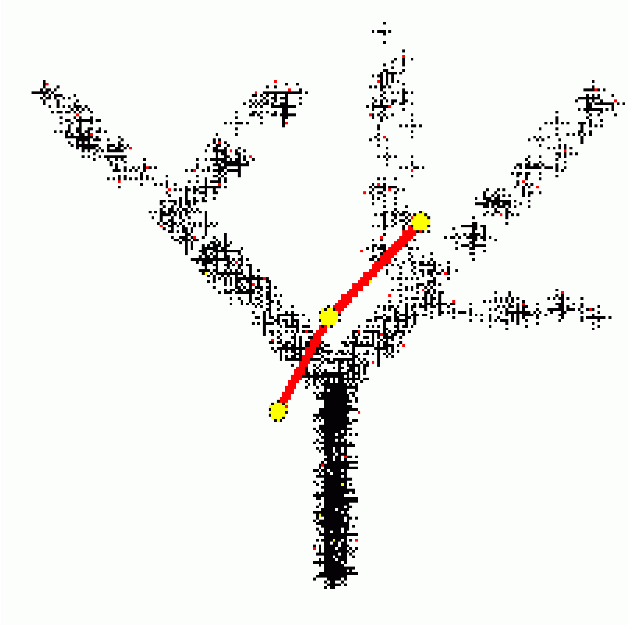
Fig. 39.17. *Dugesia*. Regeneration. A—Three individuals regenerate from an individual cut into three parts; B—Formation of a heteromorph with three heads.

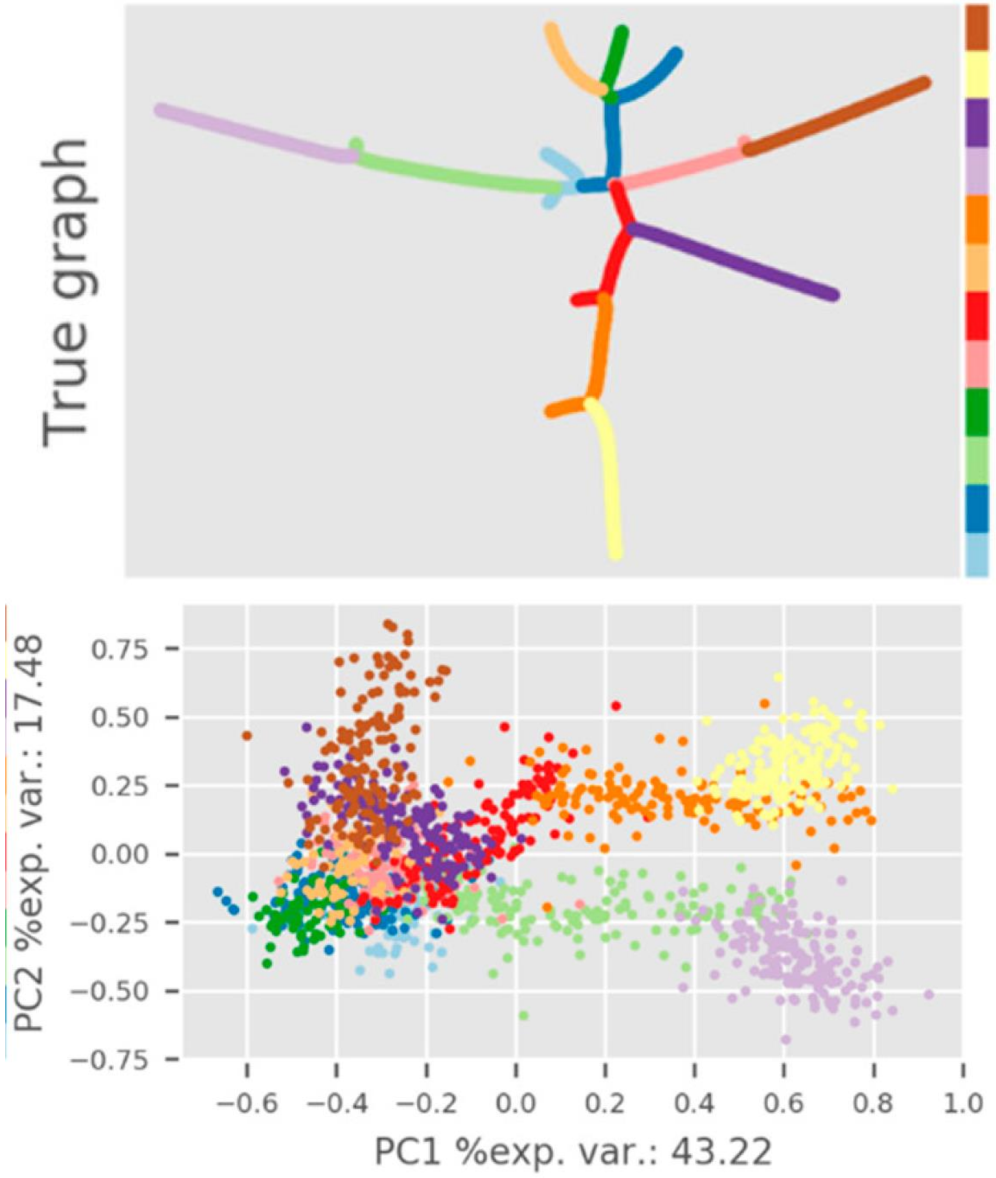




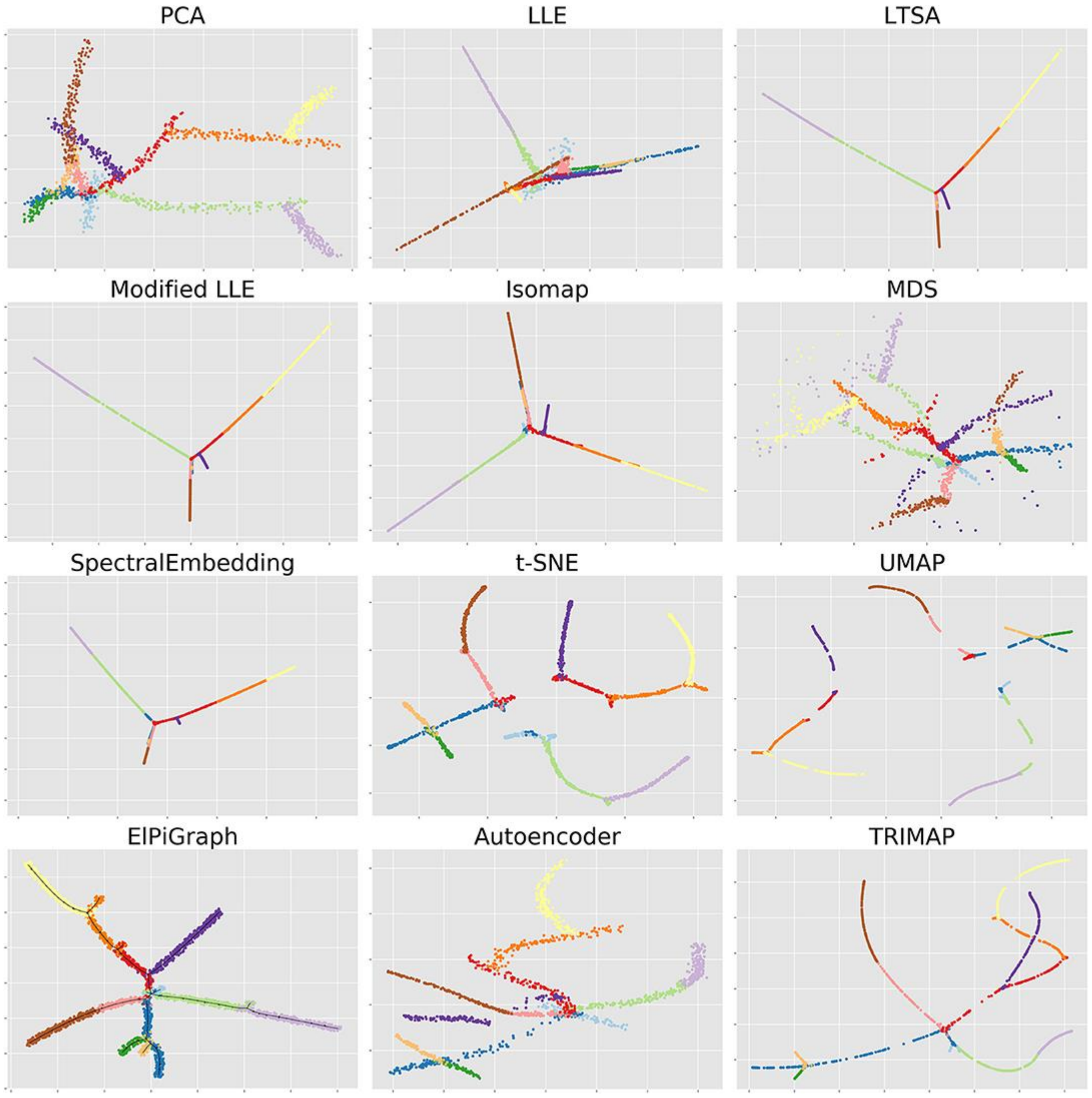


‘Branching data’





From *Albergante et al, Robust and Scalable Learning of Complex Intrinsic Dataset Geometry via ElPiGraph, Entropy, 2020*



‘Branching’ data distributions

Elastic principal graph (ElPiGraph)

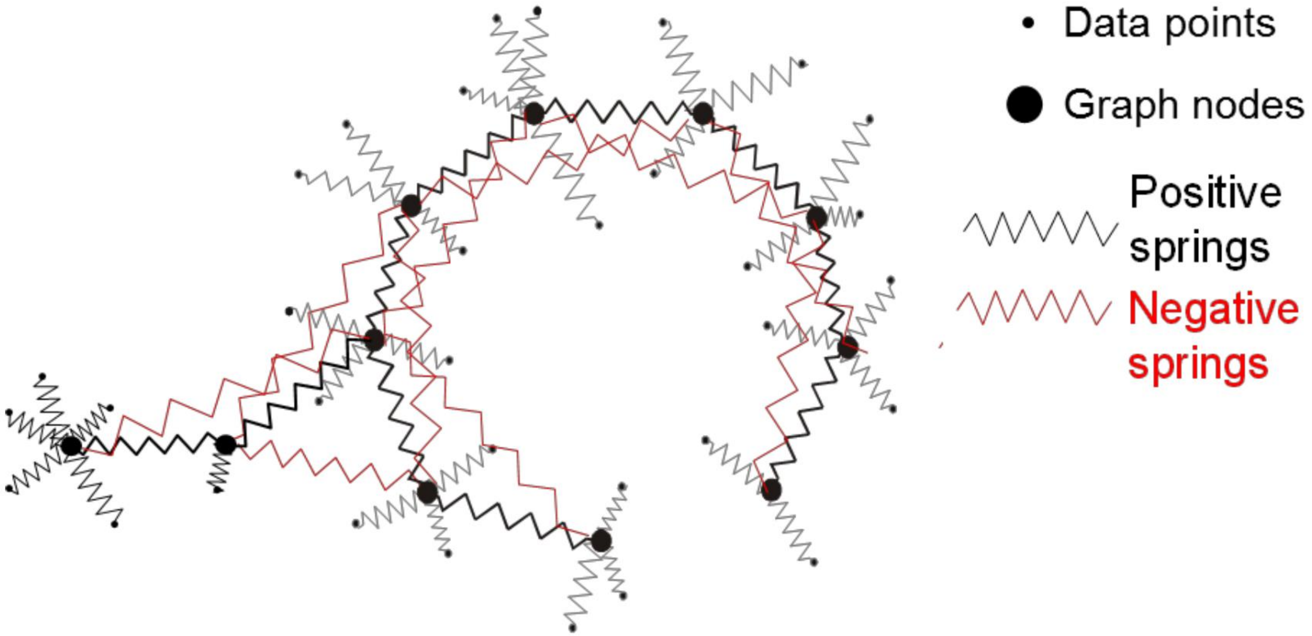
Two tasks have to be solved simultaneously:

1) Find the right graph ‘topology’
2) Fit the graph of given ‘topology’ to the data

Let us start from the second

Elastic principal graph (ElPiGraph)

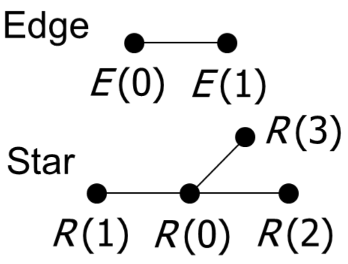
how to fit a graph to the data



We minimize the Mean Squared Distance to the neareast graph node (just like k-means)!

Penalty on **total length**:

Penalty on deviation from **harmonicity**:



$$U^{(E)} = \sum_{i=1}^{s} \lambda_i \left\| E^{(i)}(1) - E^{(i)}(0) \right\|^2$$

$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| R^{(i)}(0) - \frac{1}{k} \sum_{j=1..k} R^{(i)}(j) \right\|^2$$

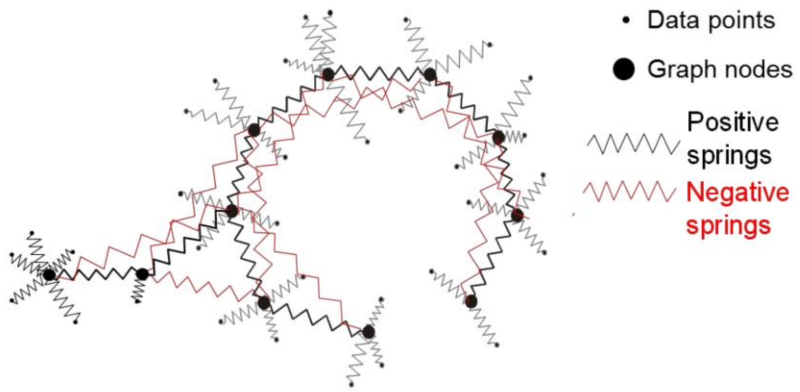
Elastic principal graph (ElPiGraph)

how to fit a graph to the data

- Splitting algorithm (Expectation-Maximization-like, very similar to k-means)

1) Start with an initial guess for the node positions
2) Perform the nearest node search for all data points
3) Solve linear equations to minimize the quadratic form

4) Repeat 2-3 until convergence

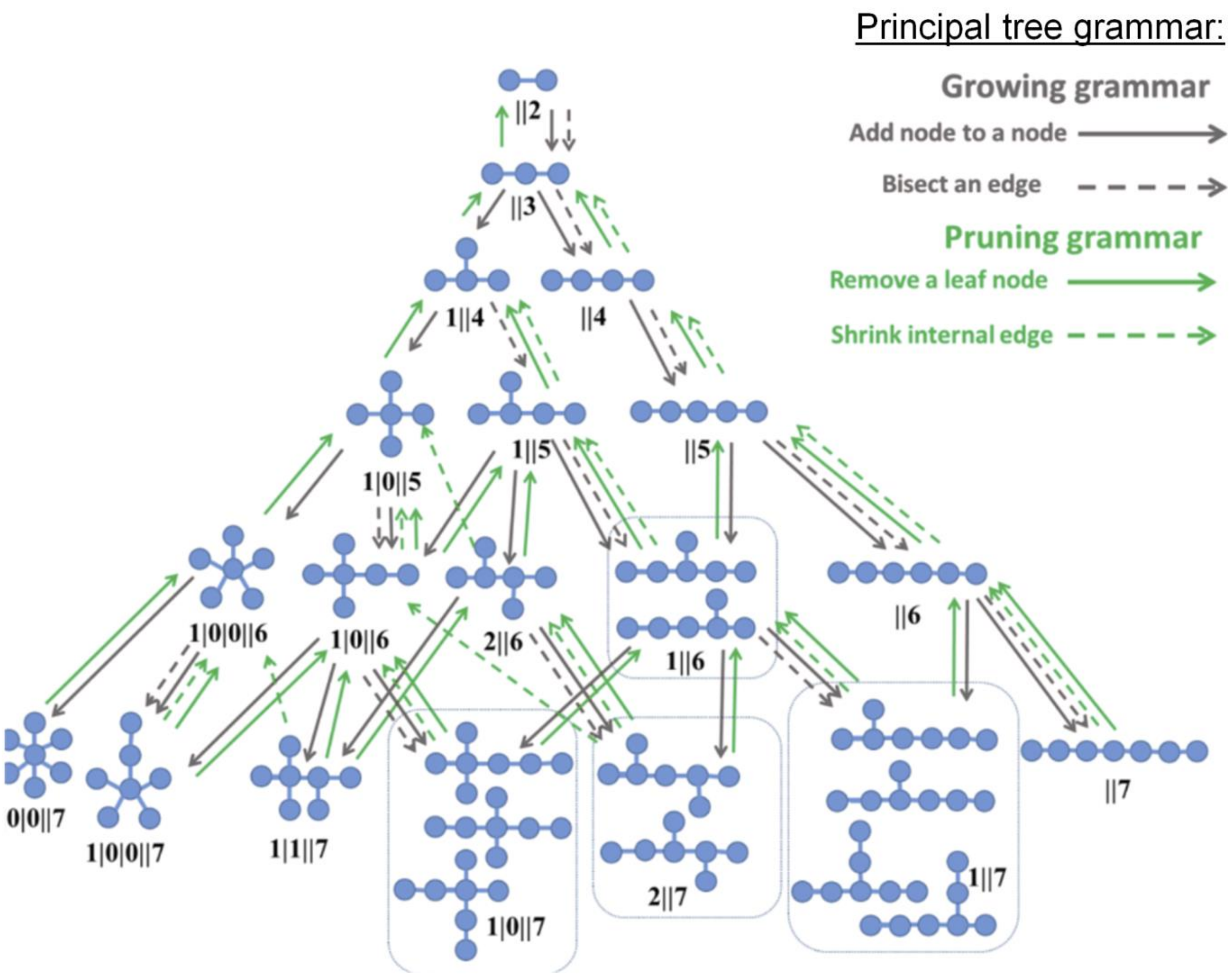


$$U^{(E)} = \sum_{i=1}^{s} \lambda_i \left\| E^{(i)}(1) - E^{(i)}(0) \right\|^2$$

$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| R^{(i)}(0) - \frac{1}{k} \sum_{j=1..k} R^{(i)}(j) \right\|^2$$

Elastic principal graph (ElPiGraph)

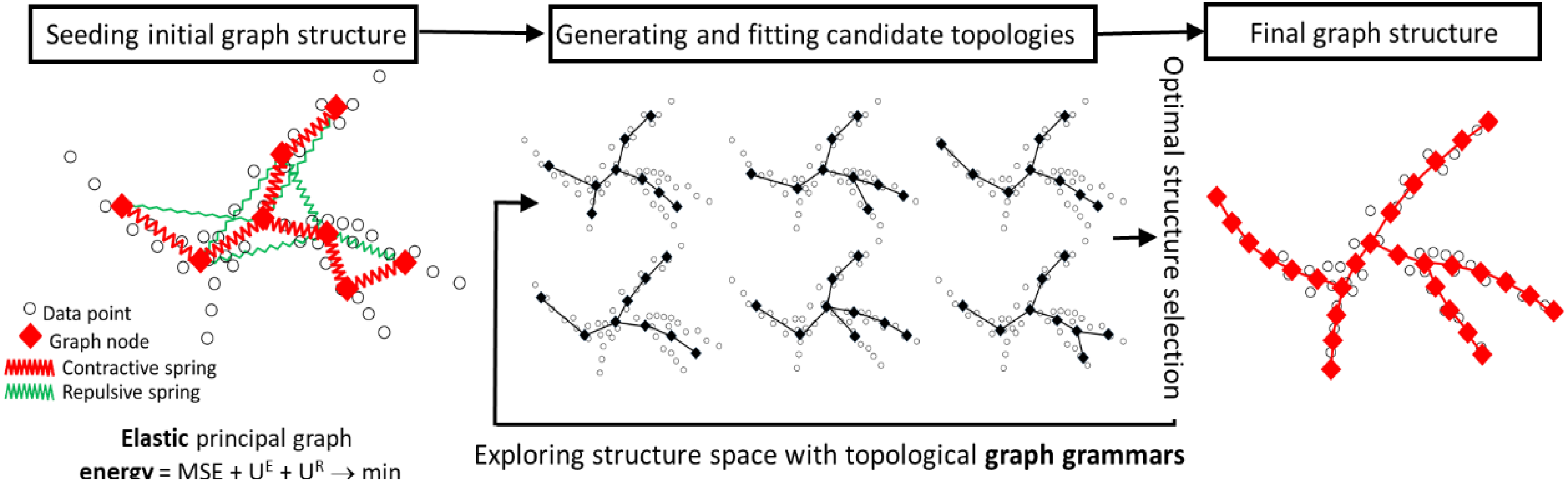
how to find the right graph structure



Topological grammars and gradient-based descent in the discrete space of graph structures

Elastic principal graph (ElPiGraph)

final algorithm



Small demo

Java applet, can be downloaded from :
http://www.ihes.fr/~zinovyev/FundamentalsOfAI2020_lectures/applet/