Topological Grammars for data analysis

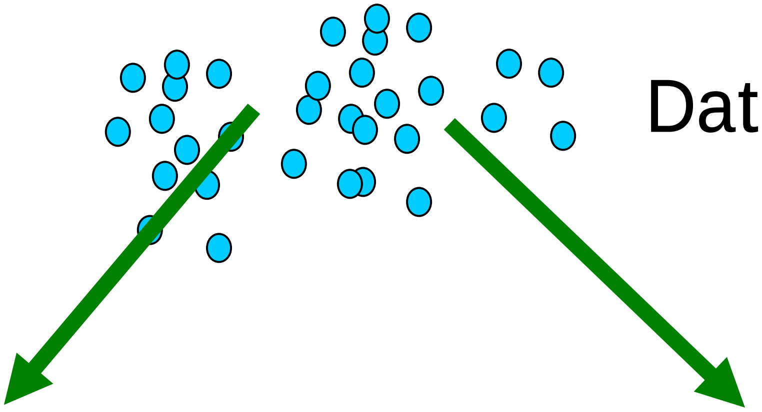
Alexander Gorban, **Leicester**

*with Andrei Zinovyev, **Paris***
*and Neil Sumner, **Leicester***

Plan of the talk

- Two paradigms for data analysis: statistics and modelling
- Clustering and K-means
- Self Organizing Maps
- Constructing PMs: elastic maps
- Adaptation and grammars
- Examples

Two basic paradigms for data analysis



Data set

Statistical Analysis

Data Modelling

Statistical Analysis

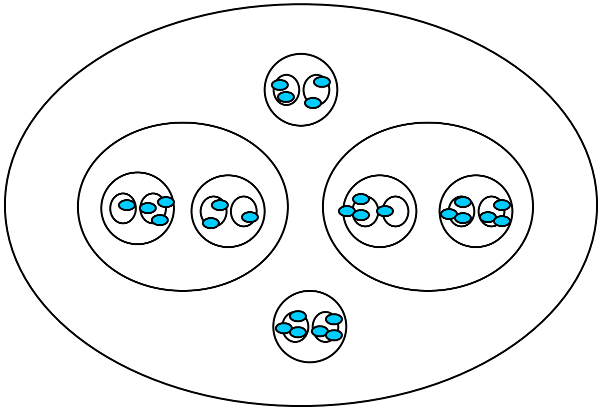
- Existence of a Probability Distribution;
- Statistical Hypothesis about Data Generation;
- Verification/Falsification of Hypothesises about Hidden Properties of Data Distribution

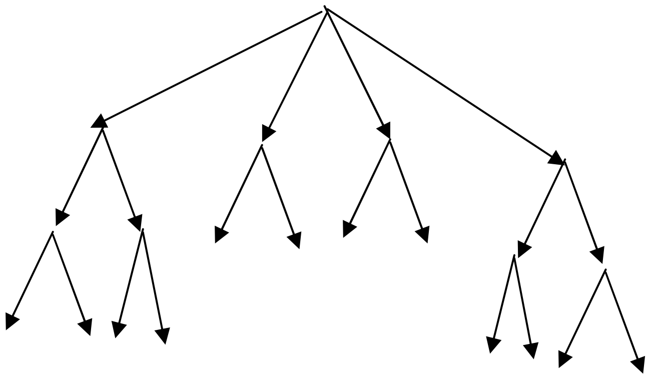
Data Modelling

Universe
of models

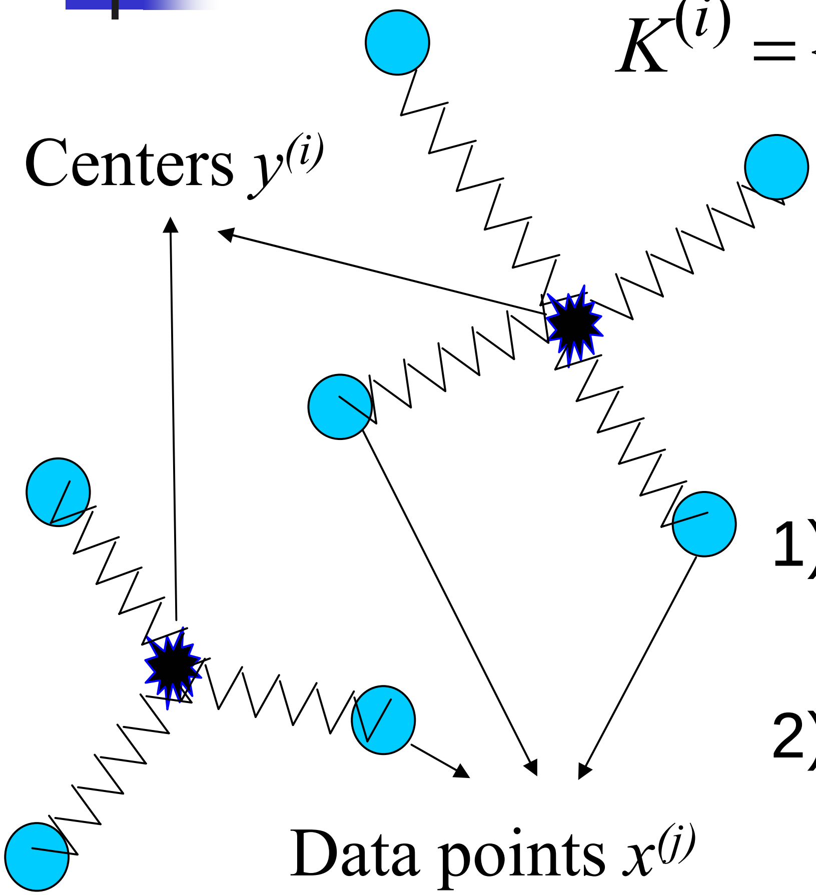
- We should find the Best Model for Data description;
- We know the Universe of Models;
- We know the Fitting Criteria;
- Learning Errors and Generalization Errors analysis for the Model Verification

Example: Simplest Clustering





K-means algorithm



$$K^{(i)} = \{x^{(j)} : \left\| x^{(j)} - y^{(i)} \right\| \le \left\| x^{(j)} - y^{(m)} \right\| \forall m\}$$

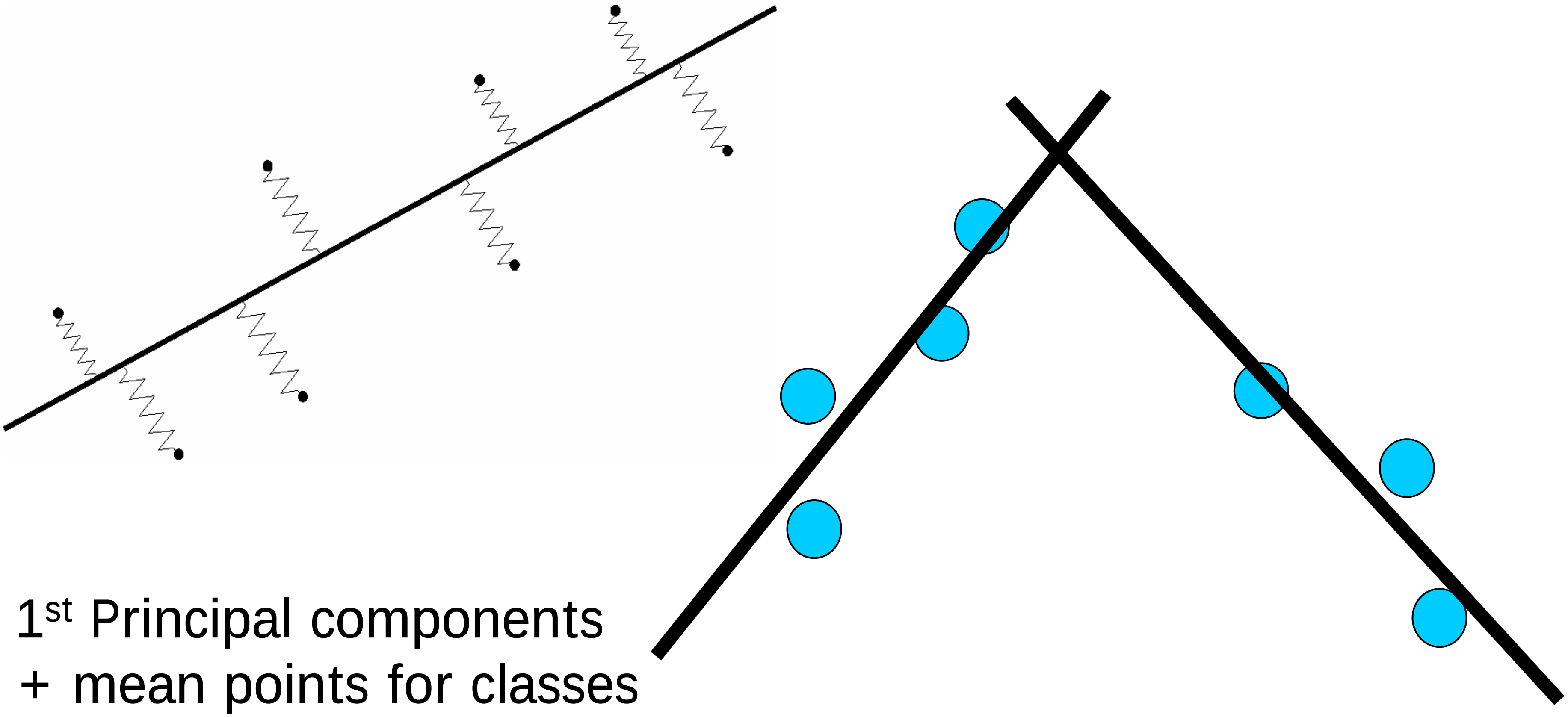
$$U = \frac{1}{N} \sum_{i=1}^{p} \sum_{x^{(j)} \in K^{(i)}} \left\| x^{(j)} - y^{(i)} \right\|^2$$

1) Minimize U for given $\{K^{(i)}\}$ (find centers);

2) Minimize $\underline{U}$ for given $\{y^{(i)}\}$ (find classes);

3) If $\{K^{(i)}\}$ change, then go to step 1.

"Centers" can be lines, manifolds,… with the same algorithm



1st Principal components
+ mean points for classes

instead of simplest means

SOM - Self Organizing Maps

- Set of nodes is a finite metric space with distance $d(N,M)$;
- 0) Map set of nodes into dataspace $N \to f_0(N)$;
- 1) Select a datapoint X (random);
- 2) Find a nearest $f_i(N)$ $(N=N_X)$;
- 3) $f_{i+1}(N) = f_i(N) + w_i(d(N, N_X))(X - f_i(N))$,
  where $w_i(d)$ $(0<w_i(d)<1)$ is a decreasing cutting function.

The closest node to X is moved the most in the direction of X, while other nodes are moved by smaller amounts depending on their distance from the closest node in the initial geometry.

PCA and Local PCA

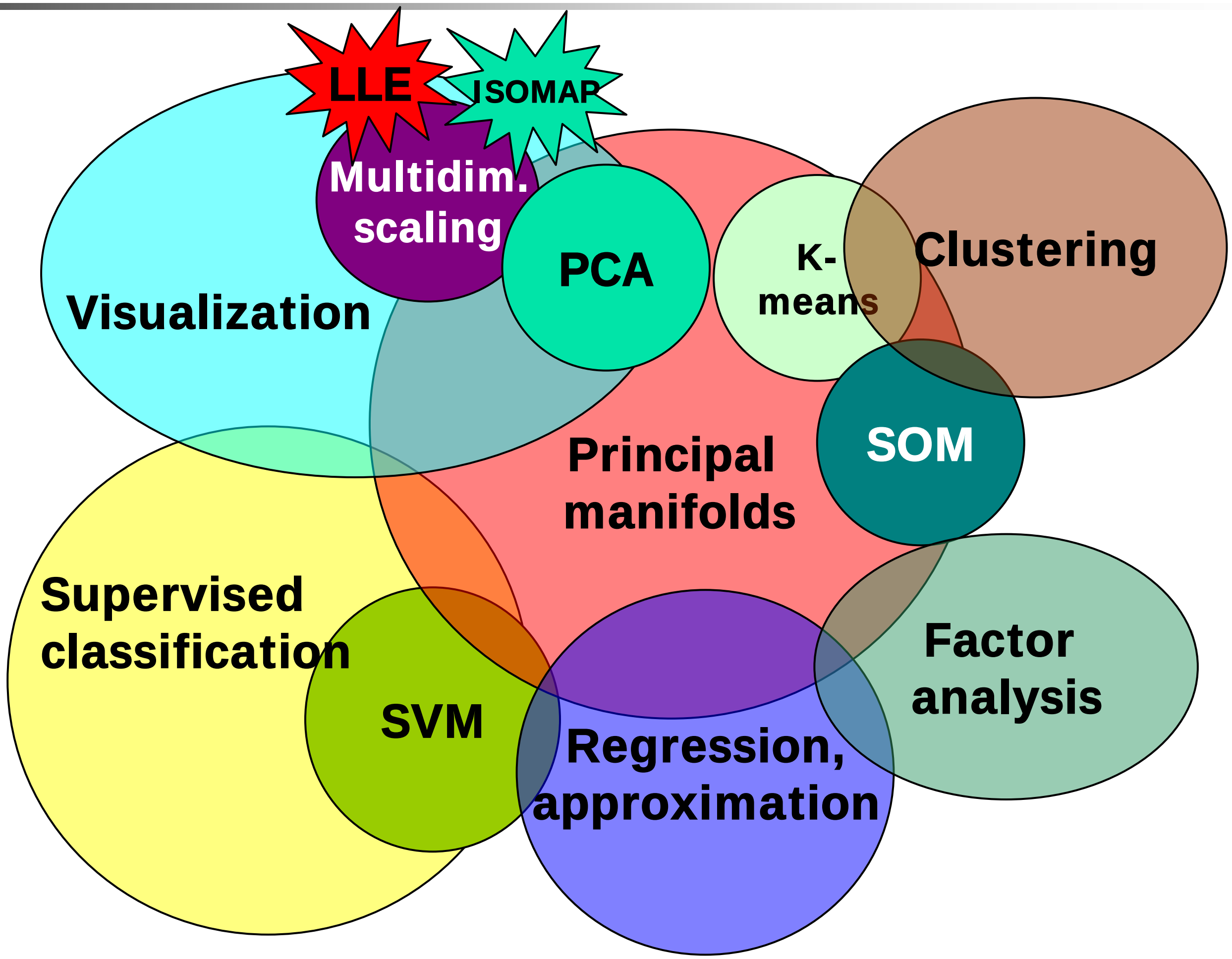
A top secret: the difference between two basic paradigms is not crucial

(Almost) Back to Statistics:

- Quasi-statistics:
  1) delete one point from the dataset,
  2) fitting,
  3) analysis of the error for the deleted data;
- The *overfitting* problem and *smoothed data points* (it is very close to non-parametric statistics)

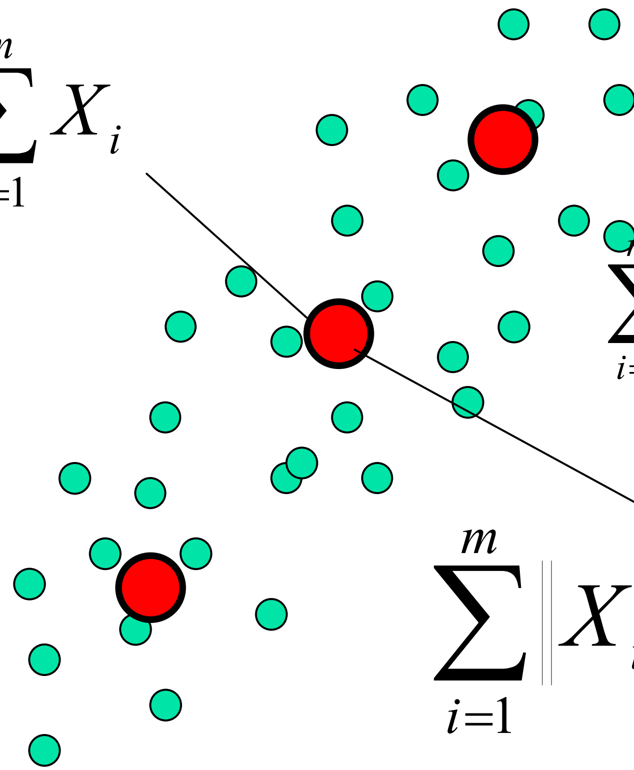
Principal manifolds
Elastic maps framework

Non-linear
Data-mining methods



Mean point

$$\langle X \rangle = \frac{1}{m} \sum_{i=1}^{m} X_i$$



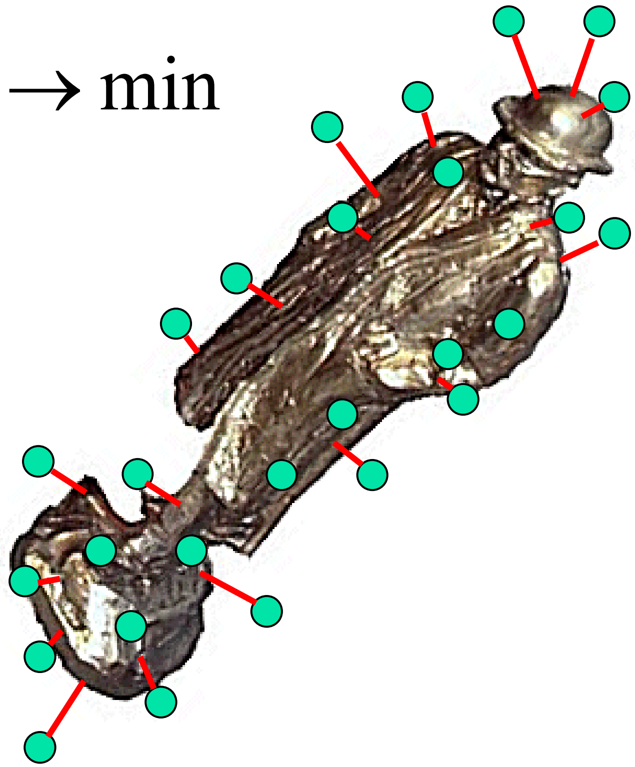
K-means clustering

$$\sum_{i=1}^{m} \left\| X_i - closest\ Y \right\|^2 \to \min$$

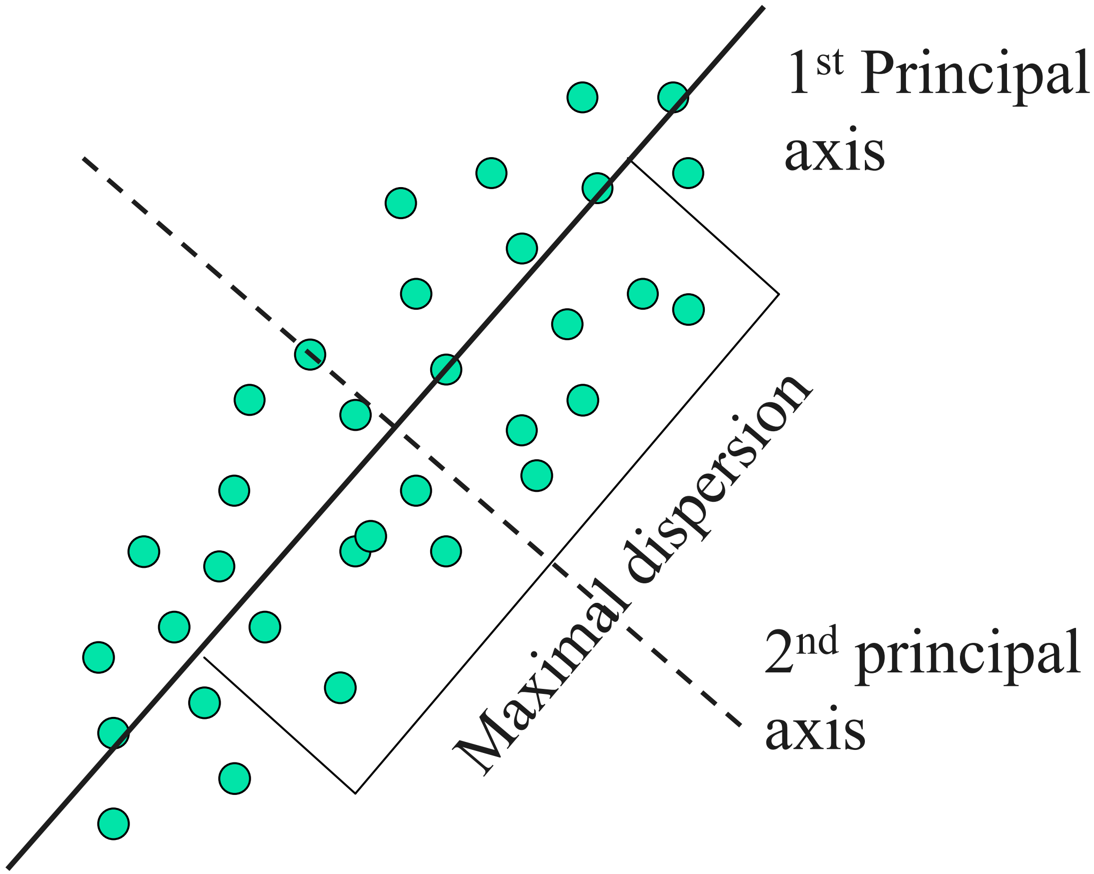
$$\sum_{i=1}^{m} \left\| X_i - \langle X \rangle \right\|^2 \to \min$$

Principal “Object”

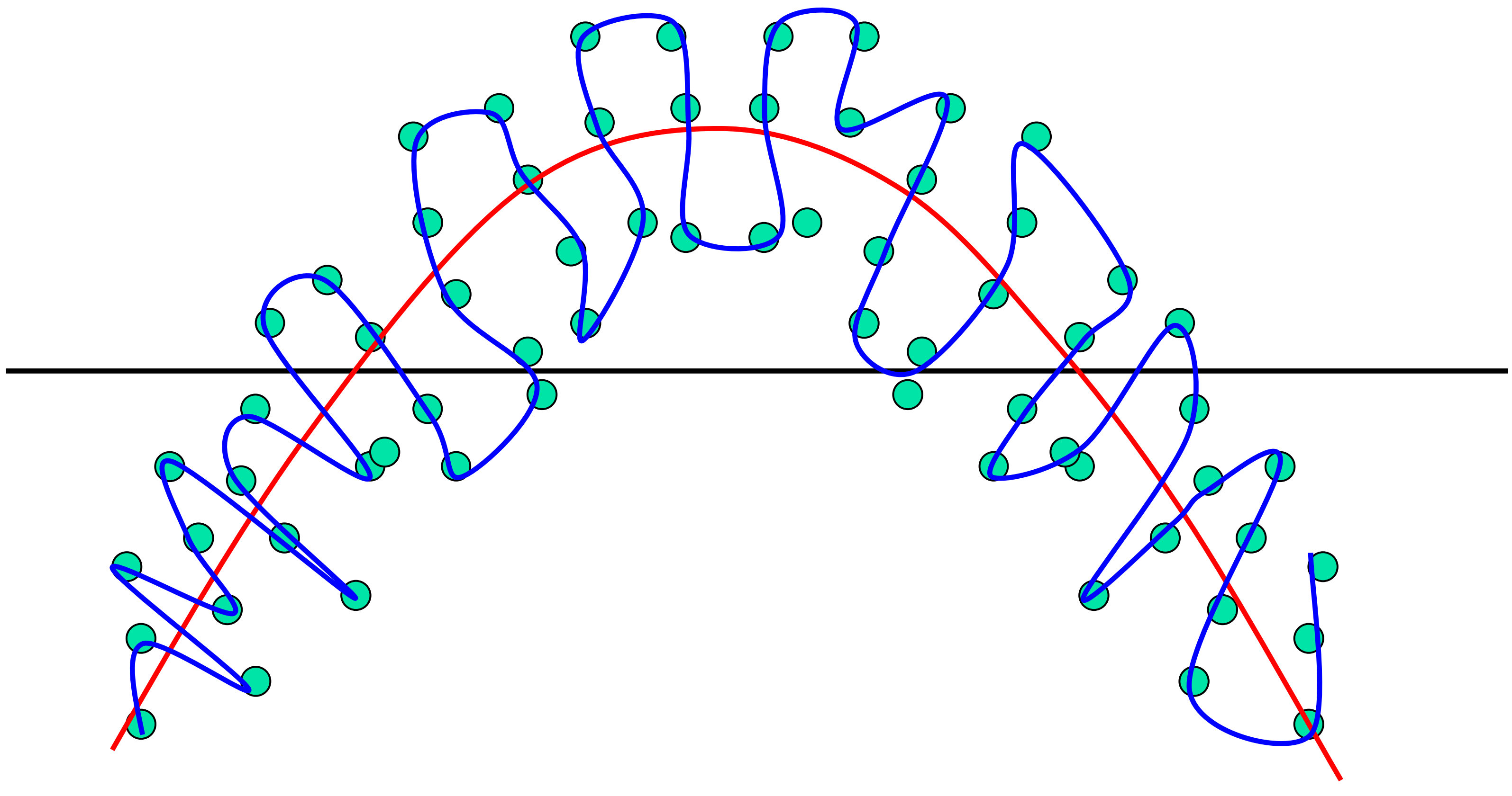
$$\sum_{i=1}^{m} \| \quad \|^2 \to \min$$



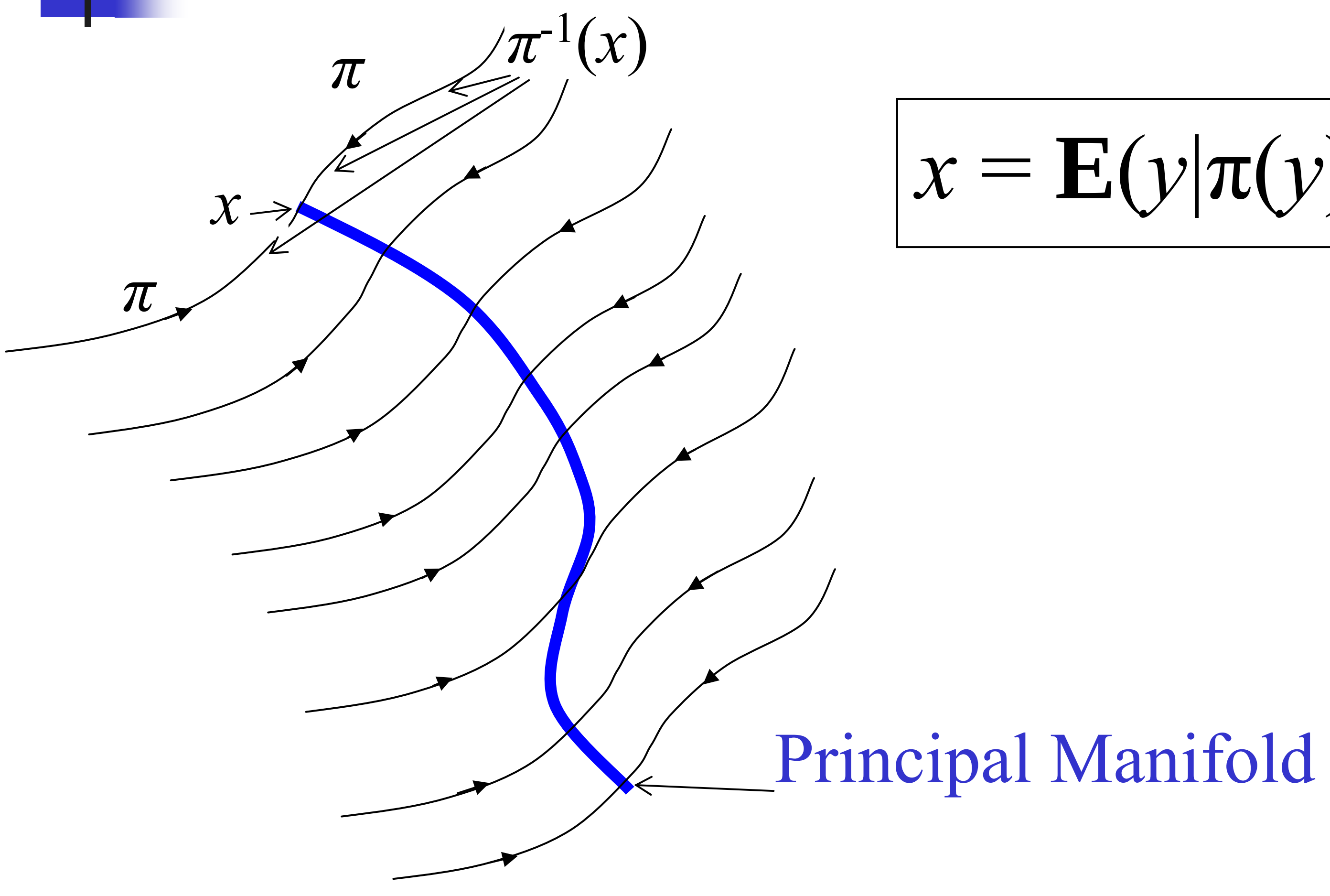
Principal Component Analysis



Principal manifold



Statistical Self-consistency

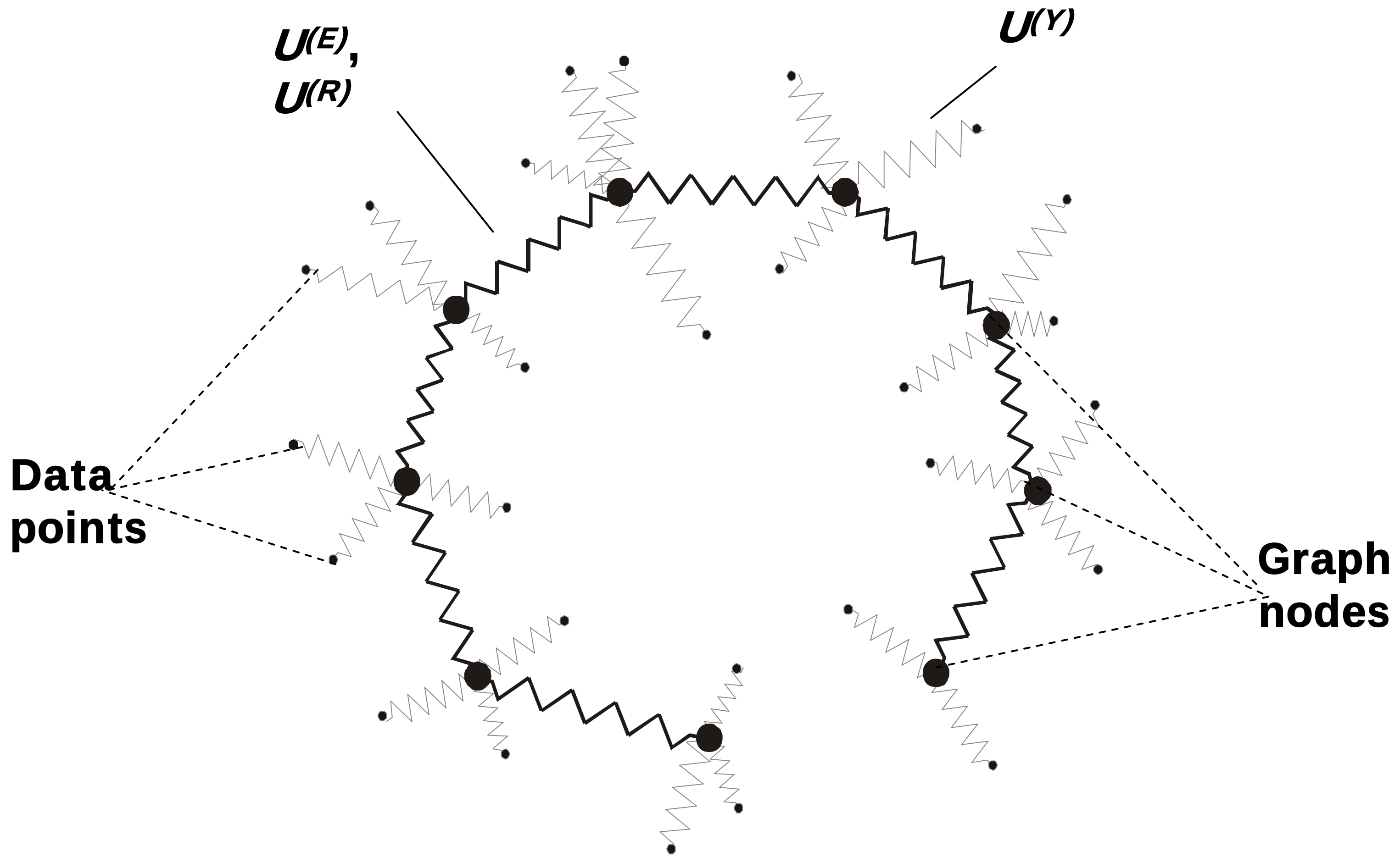


$$x = \mathbf{E}(y \mid \pi(y) = x)$$

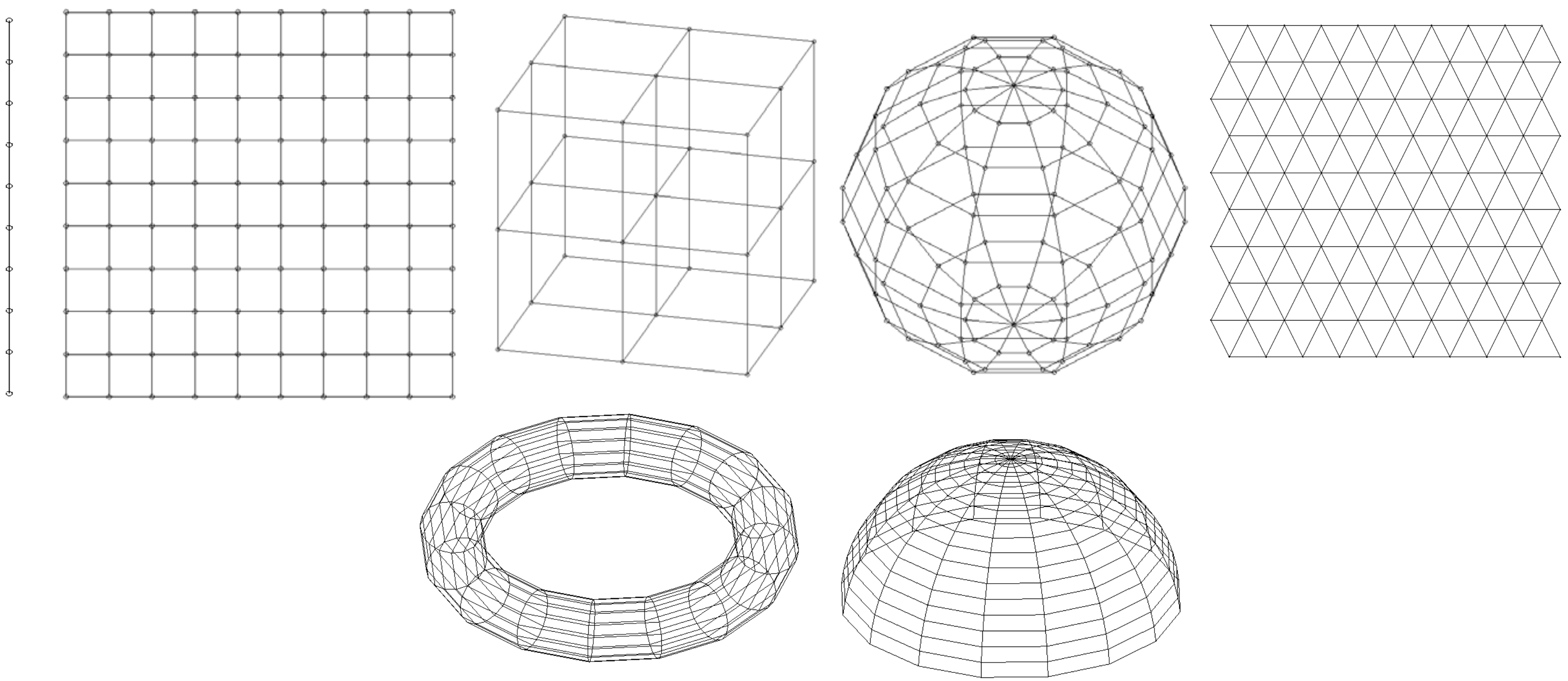
What do we want?

- Non-linear surface (1D, 2D, 3D ...)
- Smooth and not twisted
- The data model is unknown
- Speed (time linear with Nm)
- Uniqueness
- Fast way to project datapoints

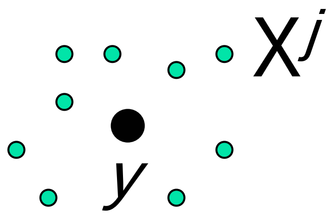
Metaphor of elasticity



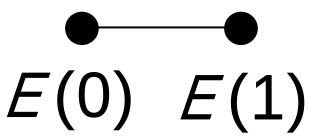
Constructing elastic nets



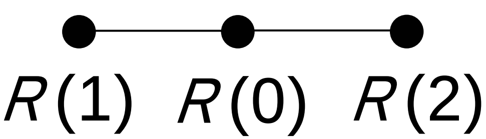
Definition of elastic energy



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



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

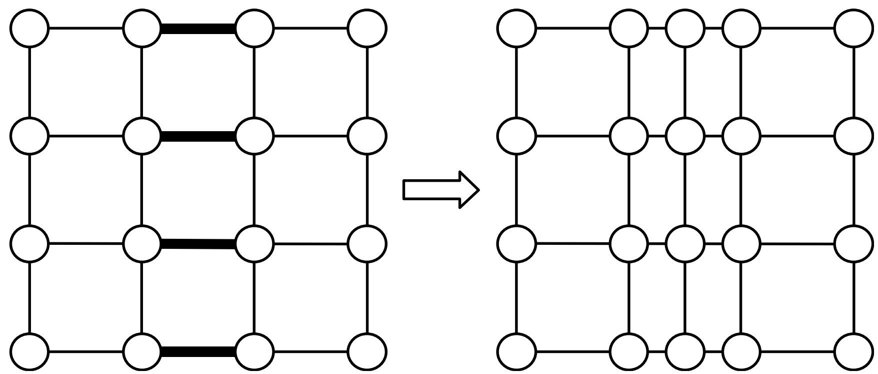


$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| R^{(i)}(1) + R^{(i)}(2) - 2R^{(i)}(0) \right\|^2$$

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

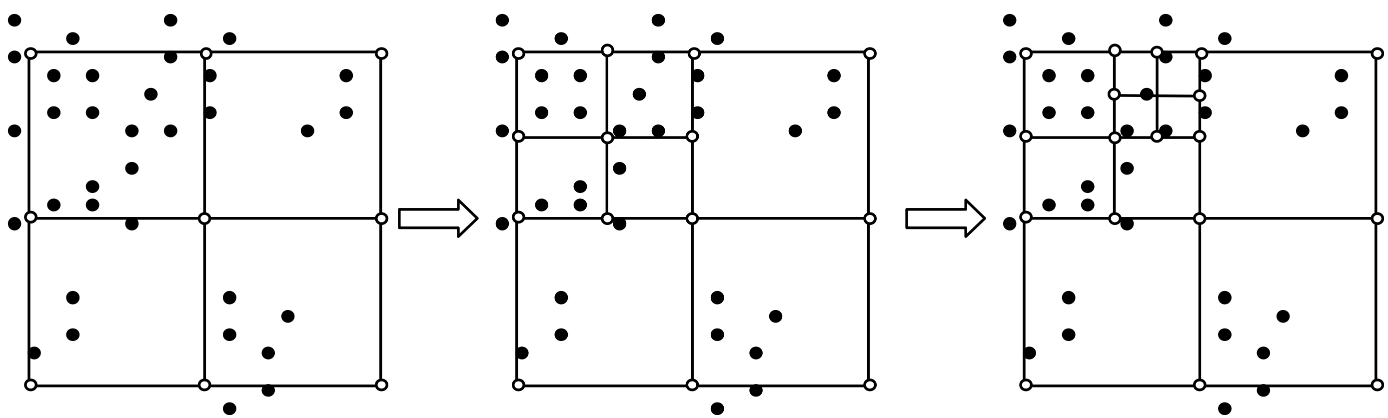
Adaptive algorithms

Refining net:



Growing net

Idea of scaling:



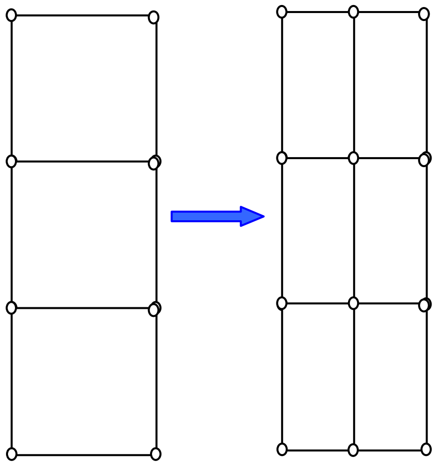
Adaptive net

Grammars of Construction

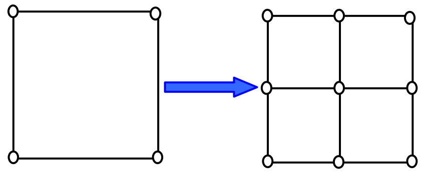
Substitution rules

Examples:

1) For net refining: substitutions of columns and rows

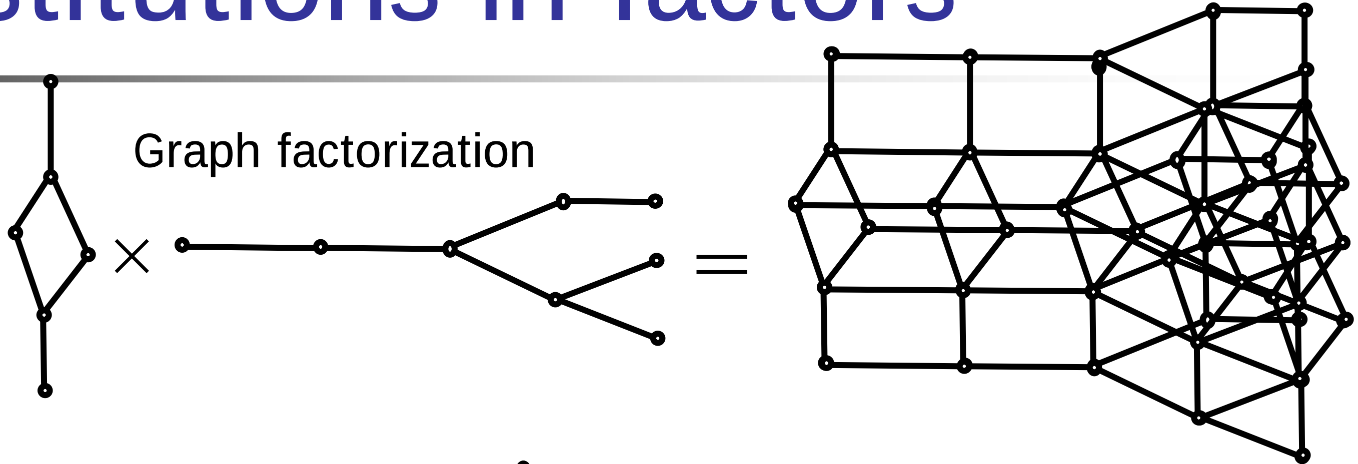


2) For growing nets: substitutions of elementary cells.

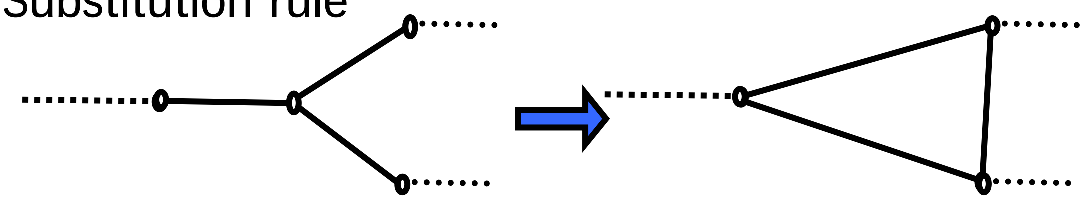


Substitutions in factors

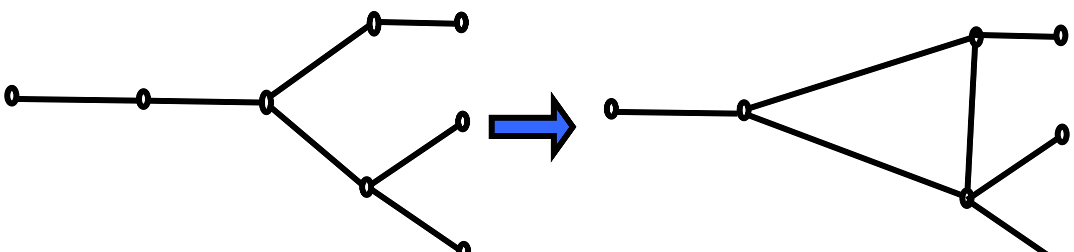
Graph factorization



Substitution rule

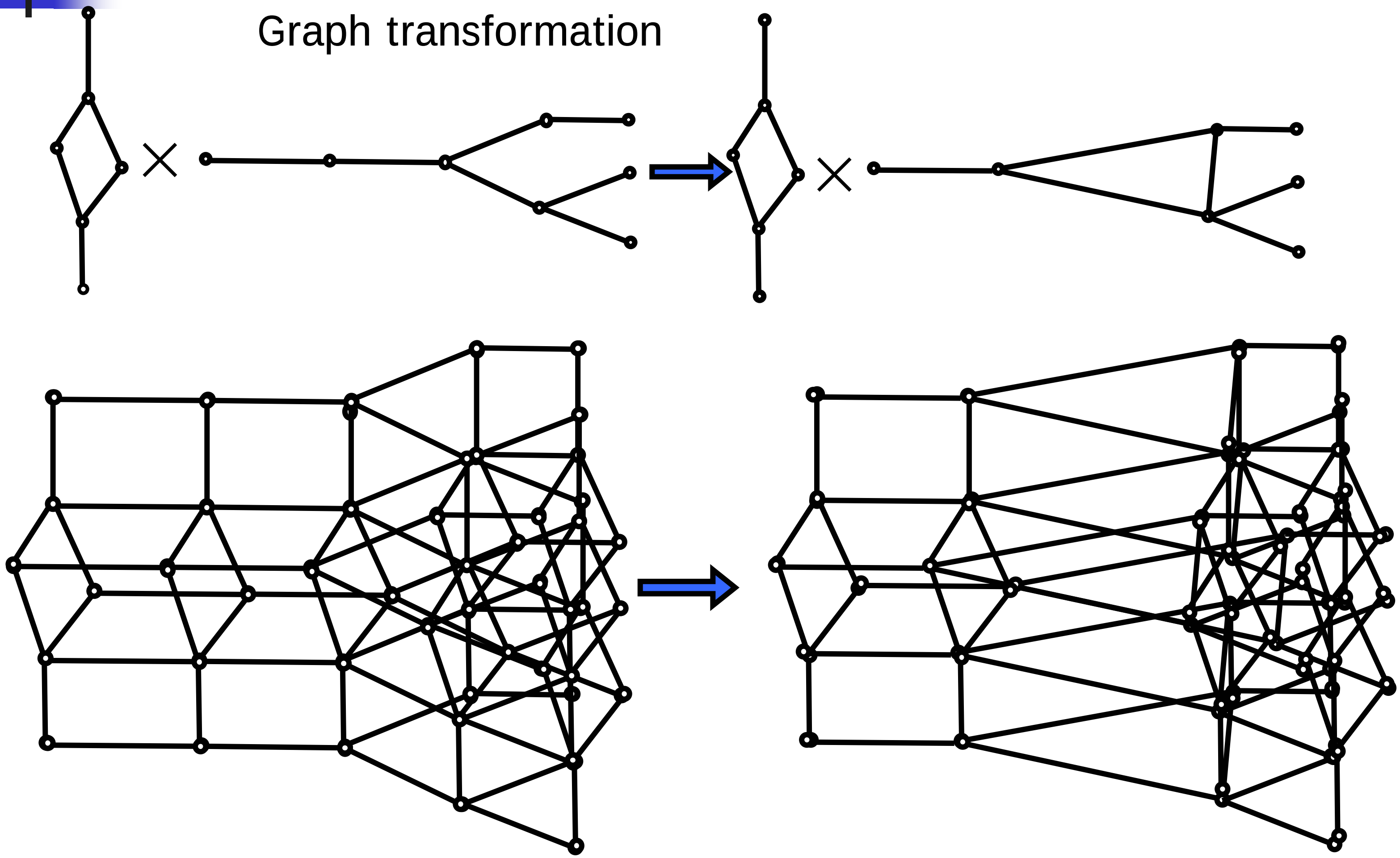


Transformation of factor



Substitutions in factors

Graph transformation



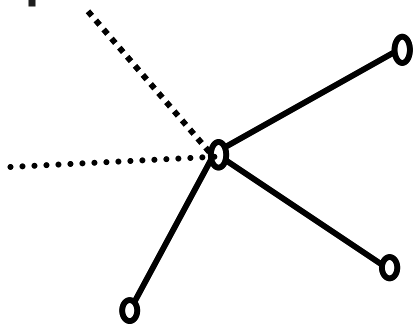
Transformation selection

A grammar is a list of elementary graph transformations.

Energetic criterion: we select and apply an elementary applicable transformation that provides the maximal energy decrease (after a fitting step).

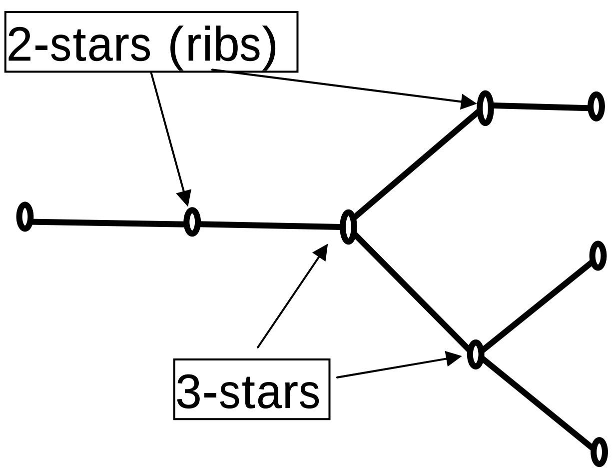
The number of operations for this selection should be in order O(N) or less, where N is the number of vertexes

Primitive elastic graphs



Elastic k-star (k edges, k+1 nodes).
The branching energy is

$$u_{k\text{-star}} = \mu_k \left(k y_0 - \sum_{i=1}^{k} y_i \right)^2$$



Primitive elastic graph: all non-terminal nodes with k edges are elastic k-stars.
The graph energy is

$$U_G = \sum_{\text{edges}} u_{\text{edge}} + \sum_{k} \sum_{k-\text{stars}} u_{\text{star}}$$

A grammar: “add a node to a node or bisect an edge”

Production:
“add a node to a node:”

A production rule applicable to any graph node y:

If y is a terminal node then add a new node z, a new edge (y,z), and a new 2-star with centre in y;

If y is a centre of a k-star then add a new node z, a new edge (y,z), and change the k-star with centre in y to (k+1)-star.

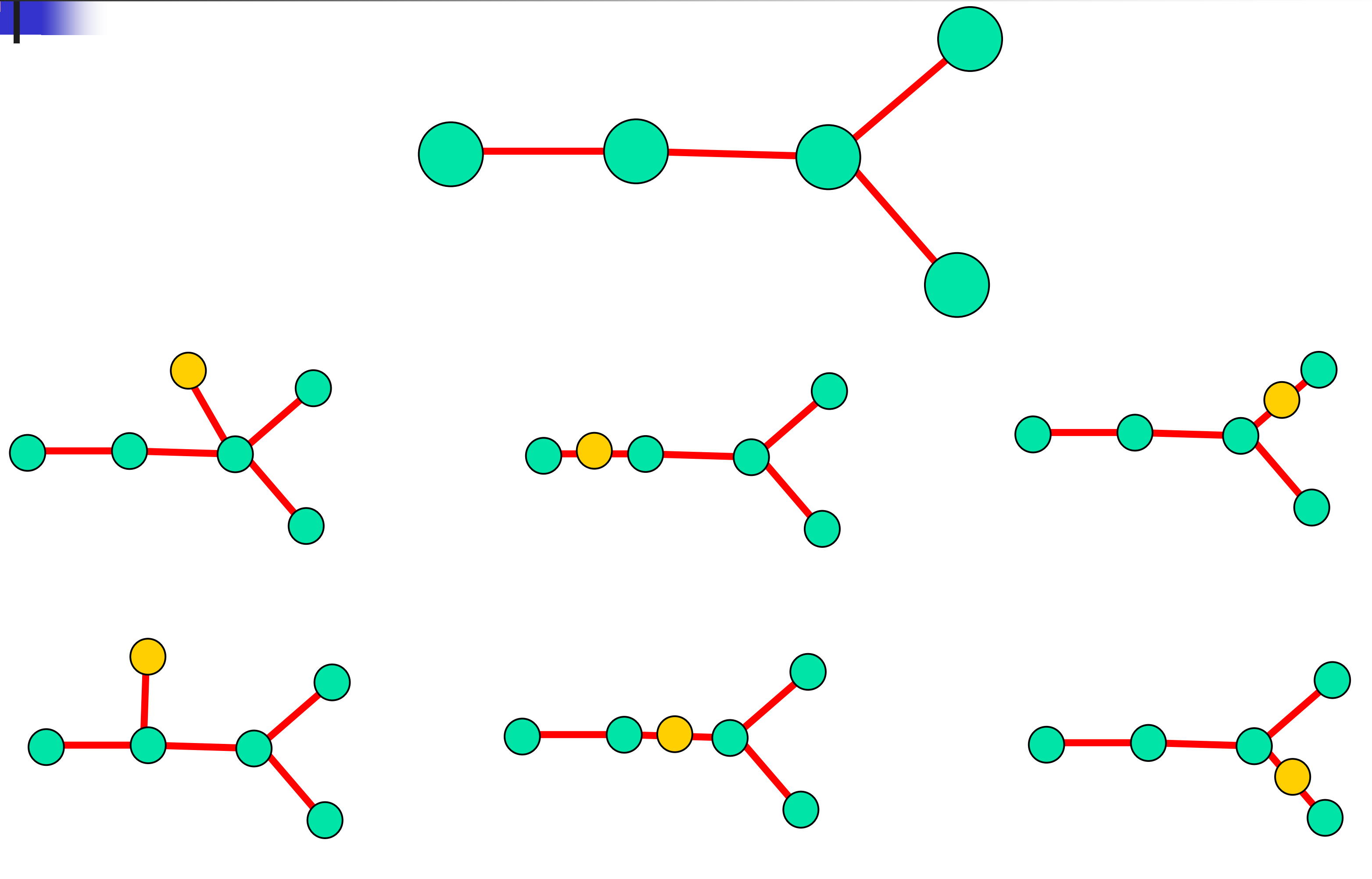
Production:
“bisect an edge:”

A production rule applicable to any graph edge (y,y’):

Delete edge (y,y’), add a vertex z, two edges, (y,z) and (z,y’), and a 2-star with the centre z.

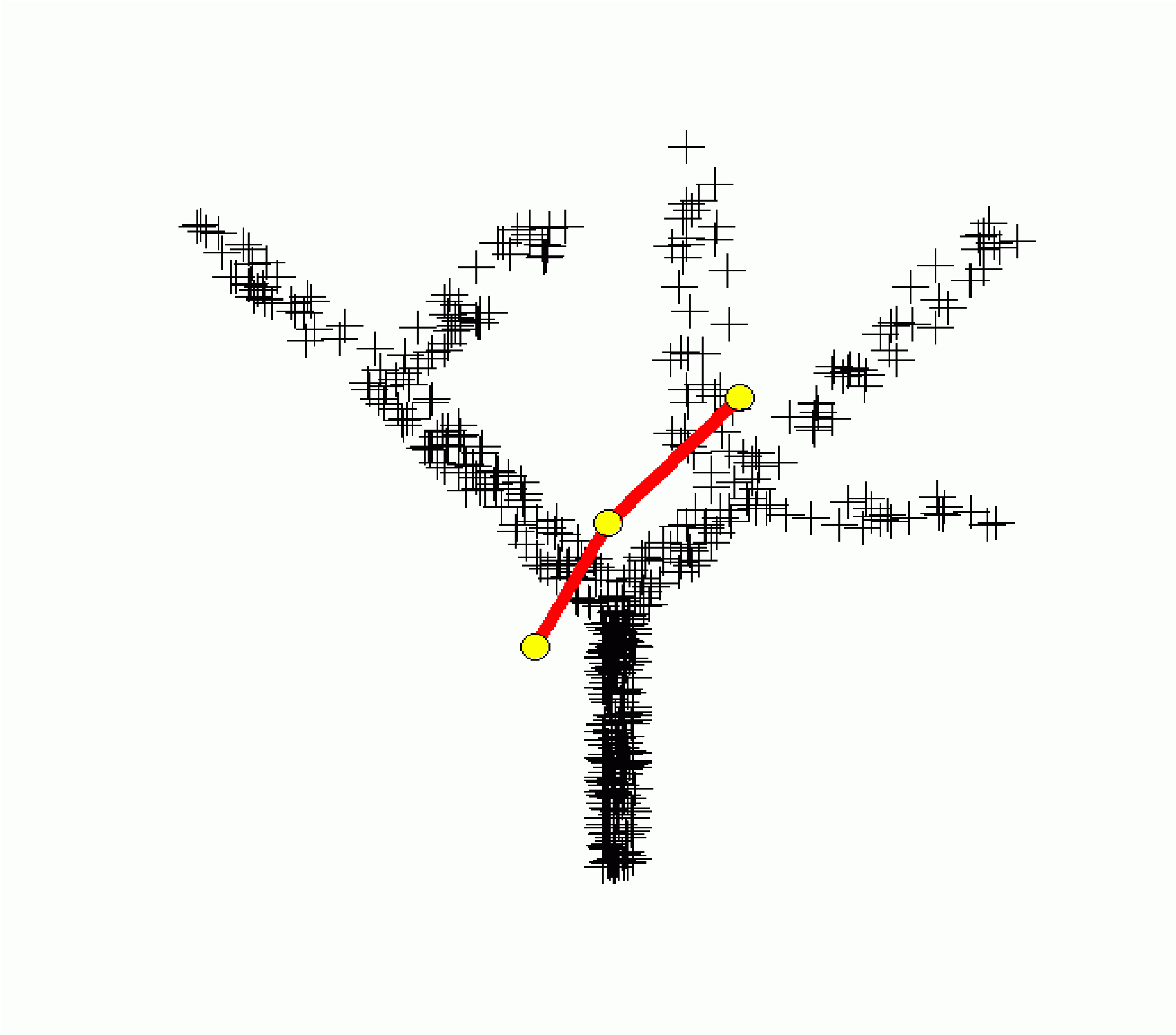
If y or y’ are centres of k-stars, change them to (k+1)- stars.

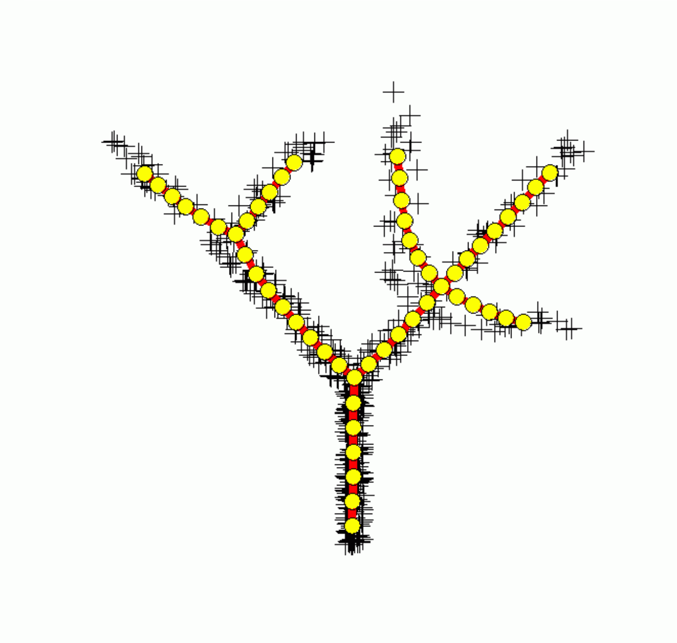
Transformations



Growing principal tree:

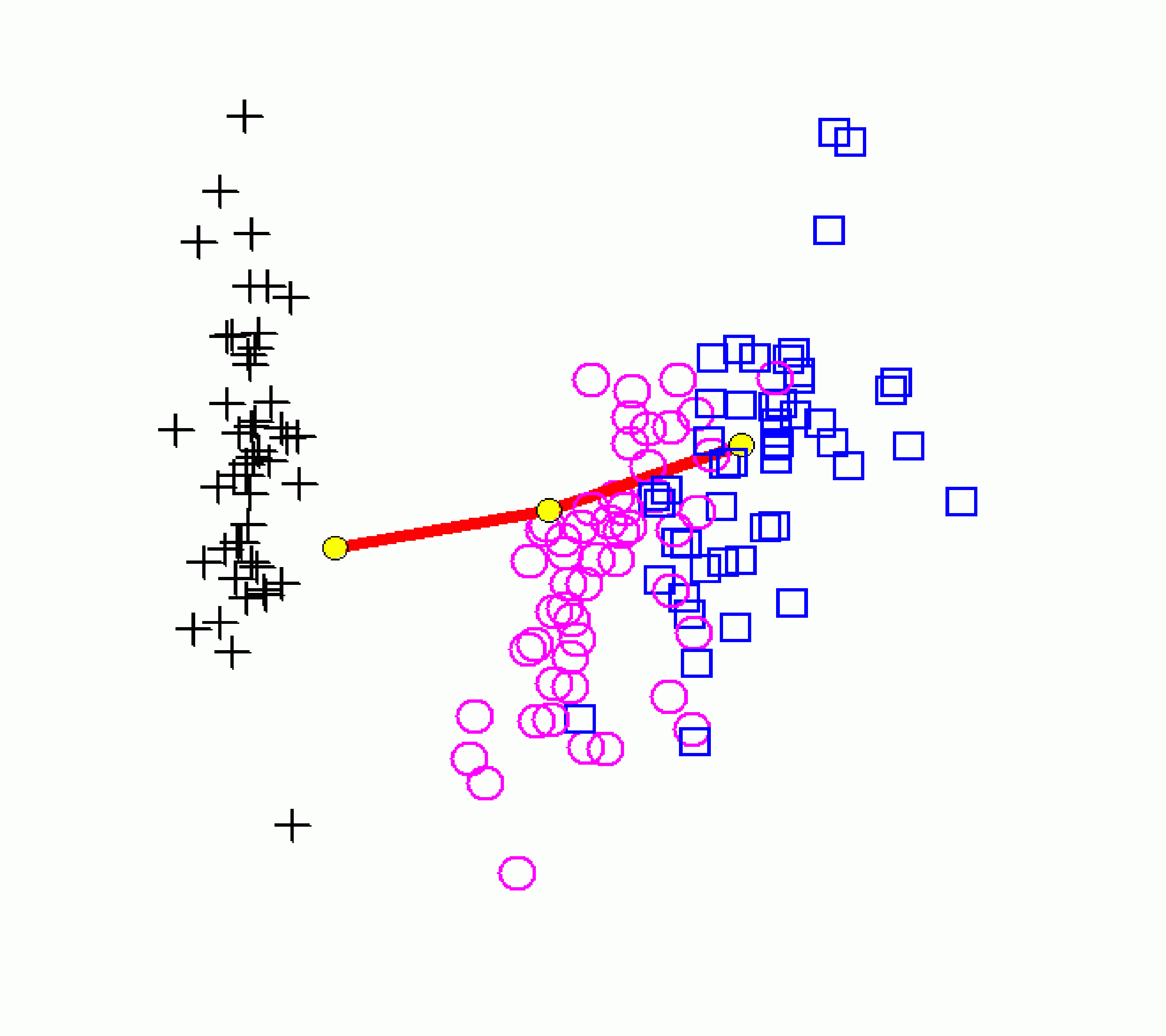
branching data distribution

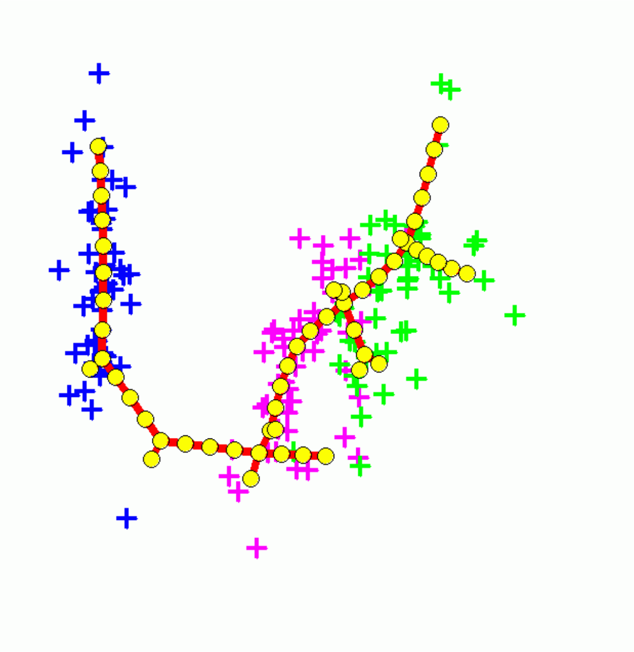




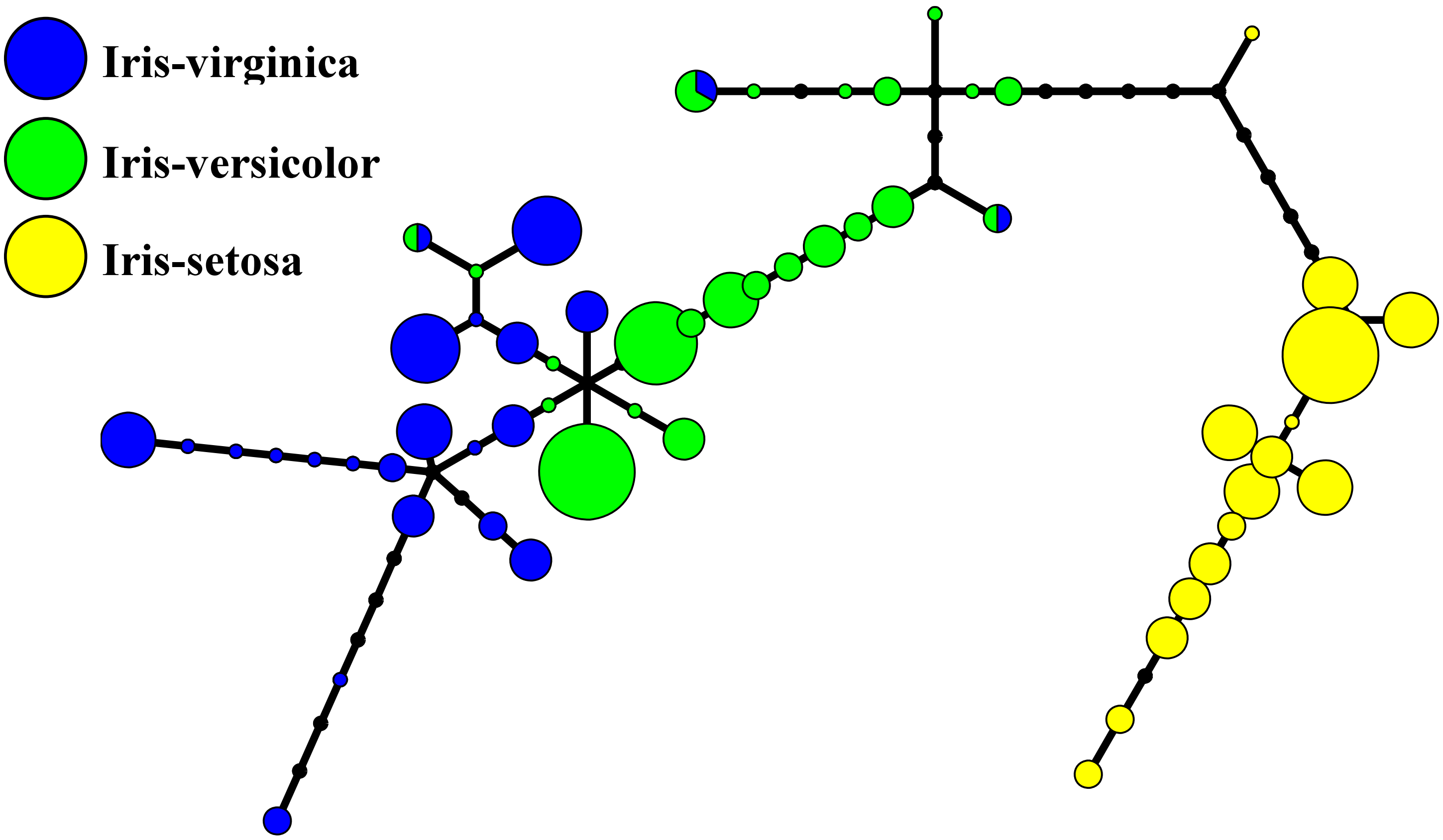
Growing principal tree:

Iris 4D dataset, PCA view



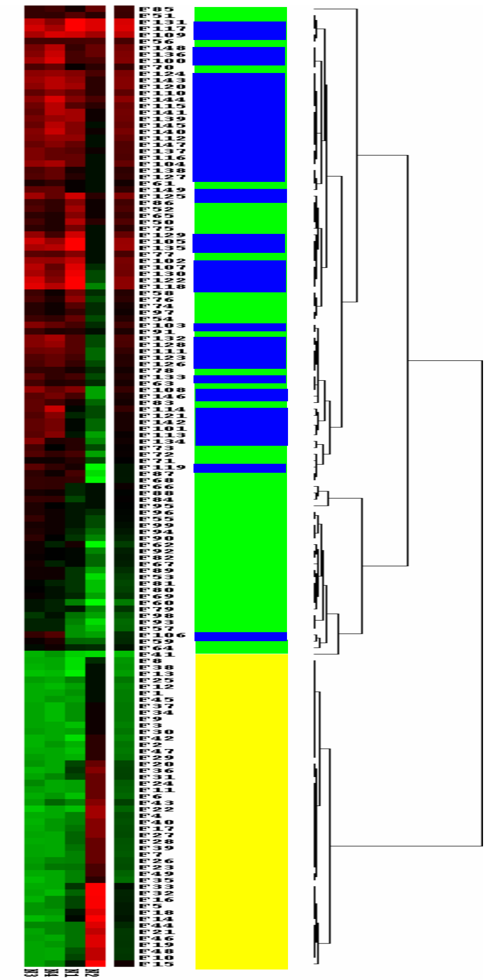


Principal coordinates: tree on plane

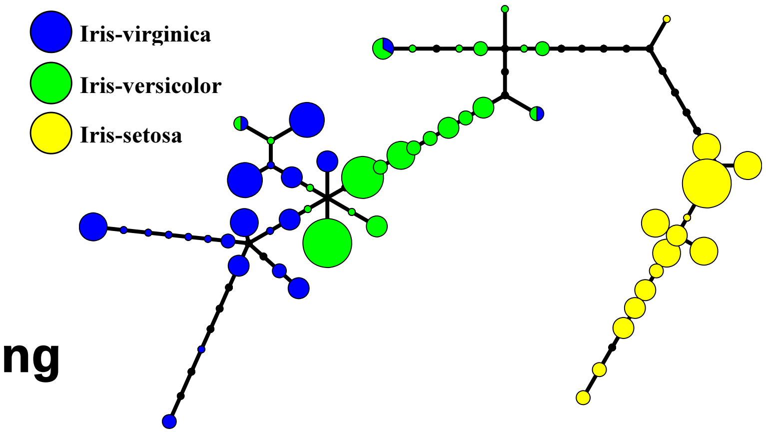


HC vs Principal Trees

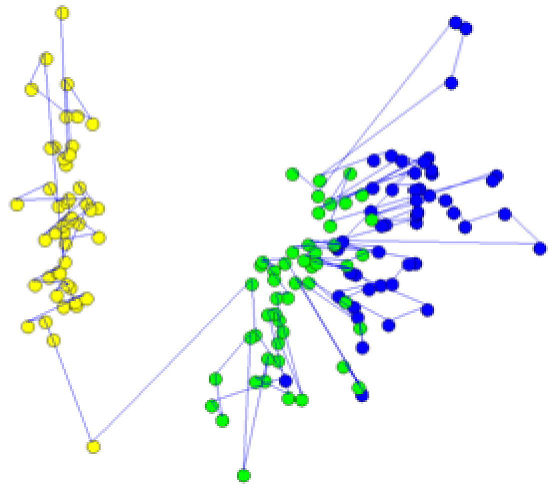
"Genealogy tree"



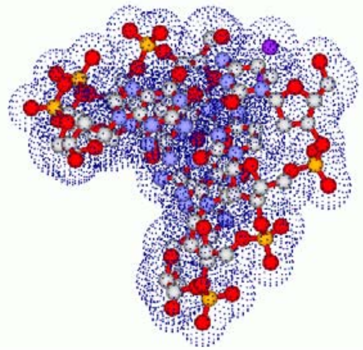
"Metro map"

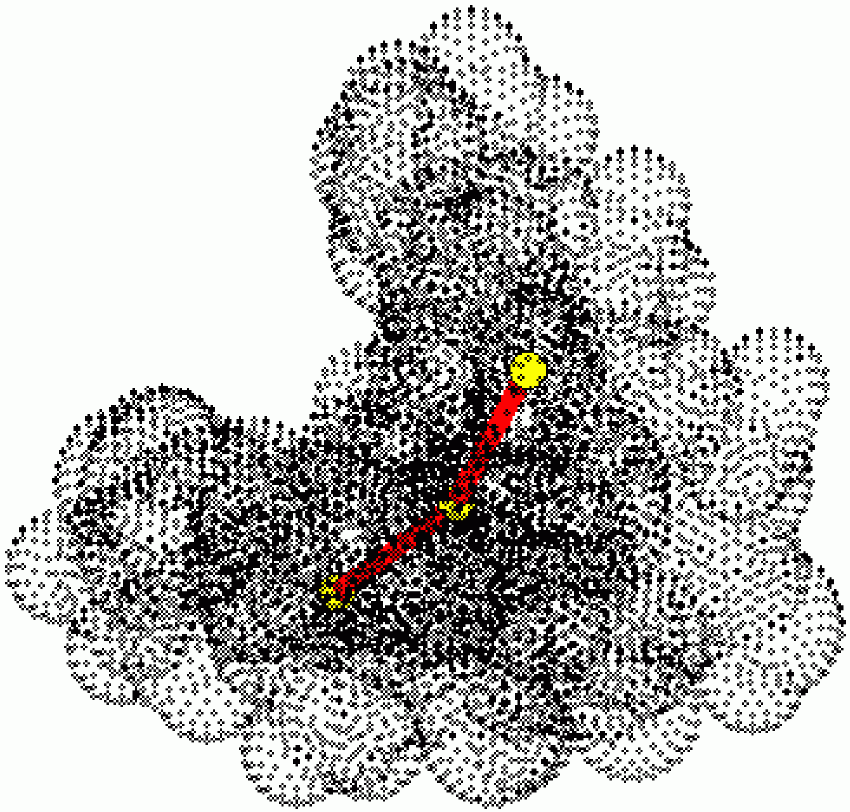


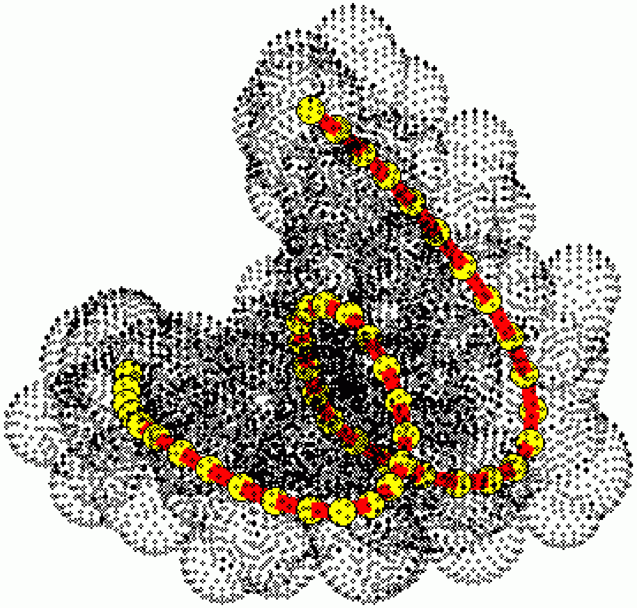
PCA, HC ordering



Growing principal tree:
DNA molecular surface

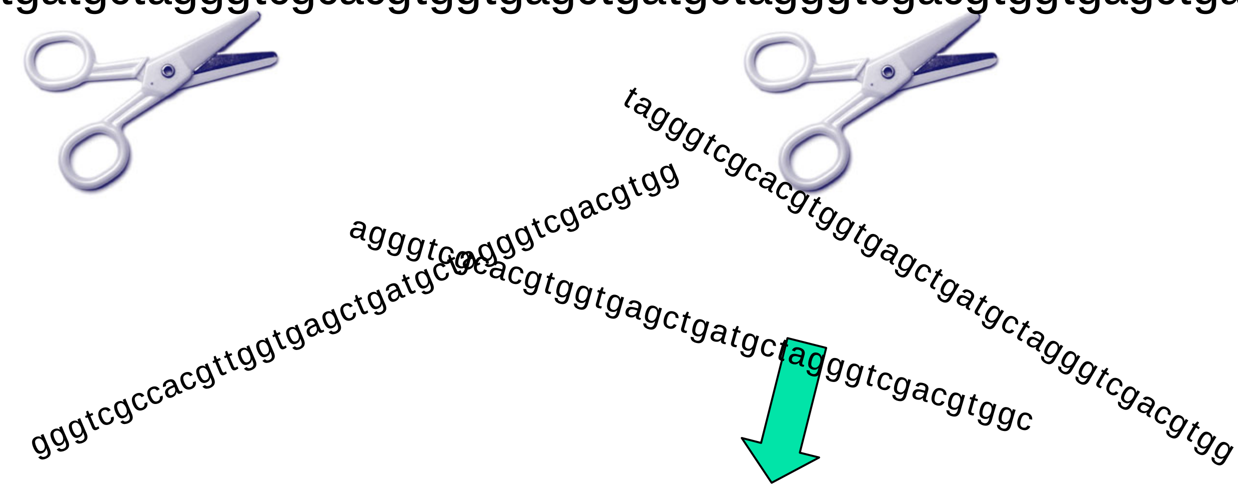






Genomic sequence
and frequency dictionaries

cgtggtgagctgatgctagggtcgcacgtggtgagctgatgctagggtcgacgtggtgagctgatgctagggtcgc



tagggtcgcacgtggtgagctgatgctaggg

frequency dictionaries:

t a g g g t c g c a c g t g g t g a g c t g a t g c t a g g g $N = 4 = 4^1$

ta gg gt cg ca cg tg gt ga gc tg at gc ta gg $N = 16 = 4^2$

tag ggt cgc acg tgg tga gct gat gct agg $N = 64 = 4^3$

tagg gtcg cacg tggt gagc tgat gcta gggt $N = 256 = 4^4$

From text to geometry

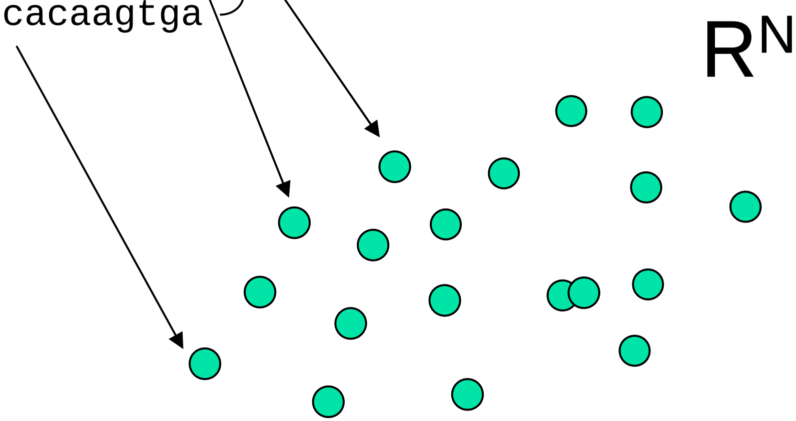
cgtggtgagctgatgctagggtcgcacgtggtgagctgatgctagggtcgacgtggtgagctgatgctagggtcgc

10^7

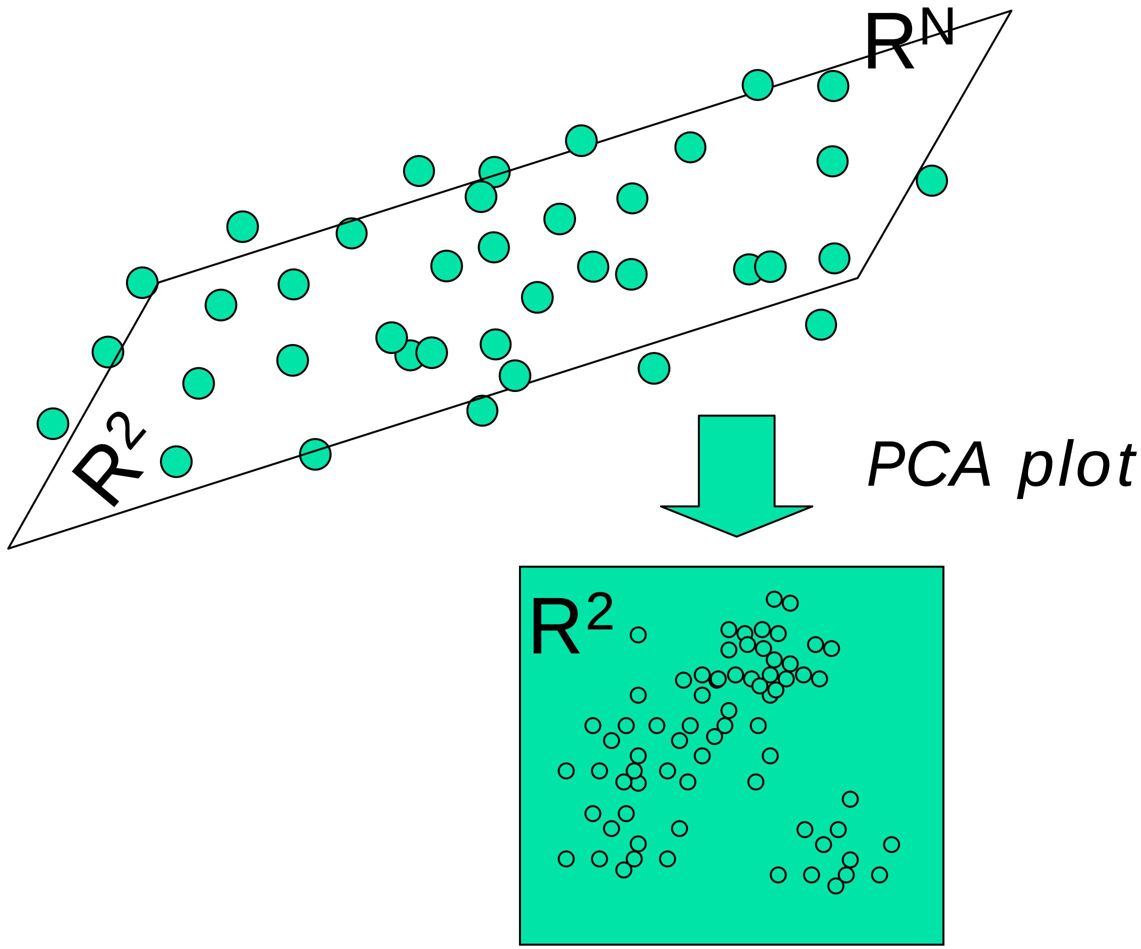
length~300-400

```
cgtggtgagctgatgctagggtcgcac
ggtgagctgatgctagggtcgcacact
tgagctgatgctagggtcgcacaattc
gtgagctgatgctagggtcgcacggtg
……
gagctgatgctagggtcgcacaagtga
```

3000-4000 fragments



Method of visualization
principal components analysis



Caulobacter crescentus

singles
N=4

doublets
N=16

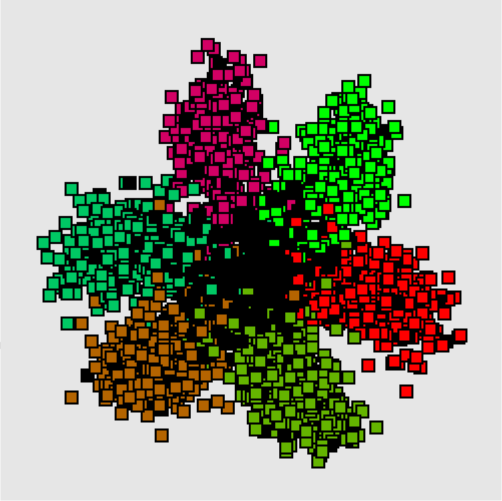
triplets
N=64

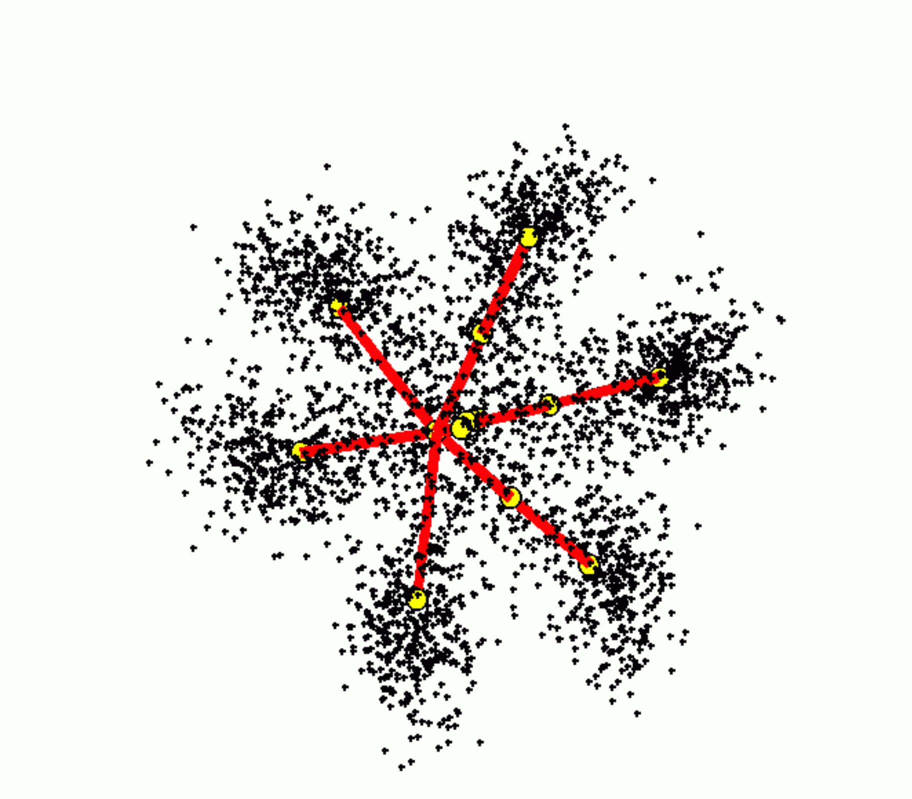
quadruplets
N=256

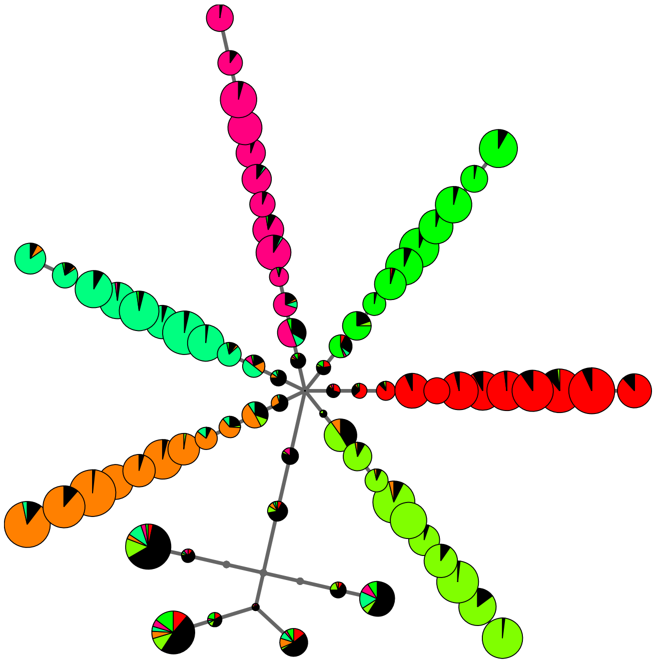
!!!

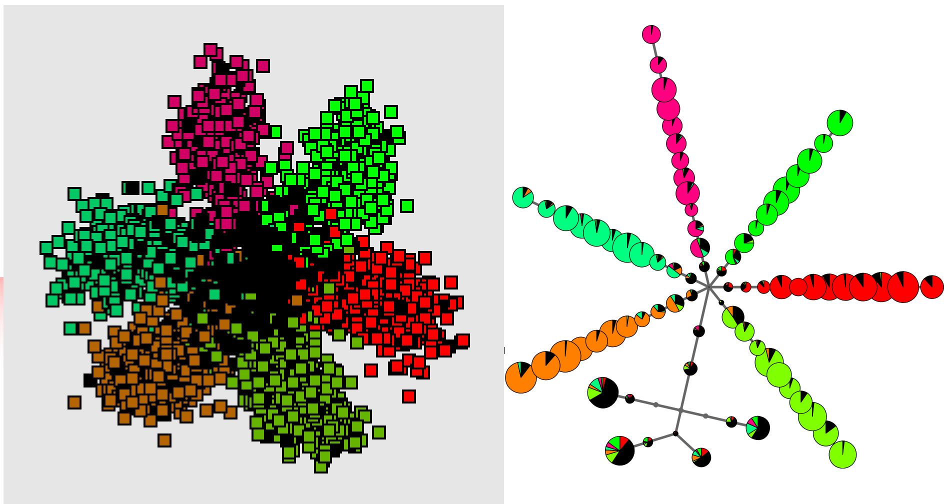
the information in genomic sequence is encoded by non-overlapping triplets

Streptomyces coelicolor
7-clusters structure

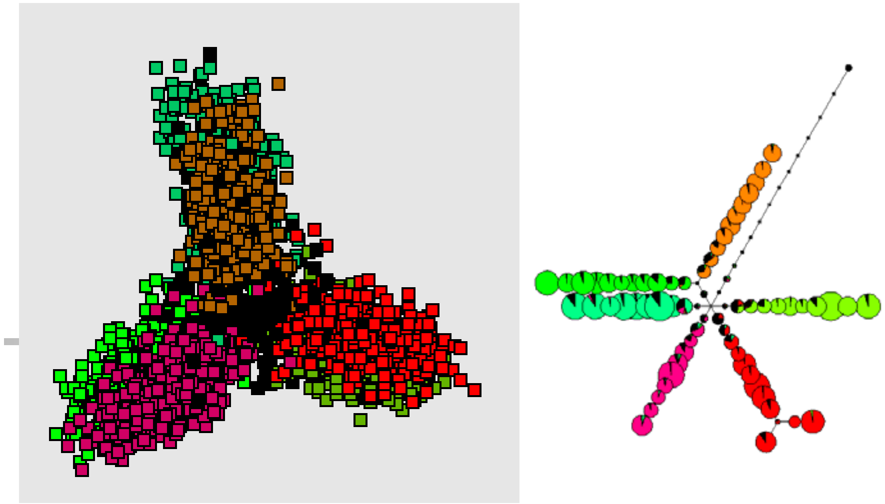




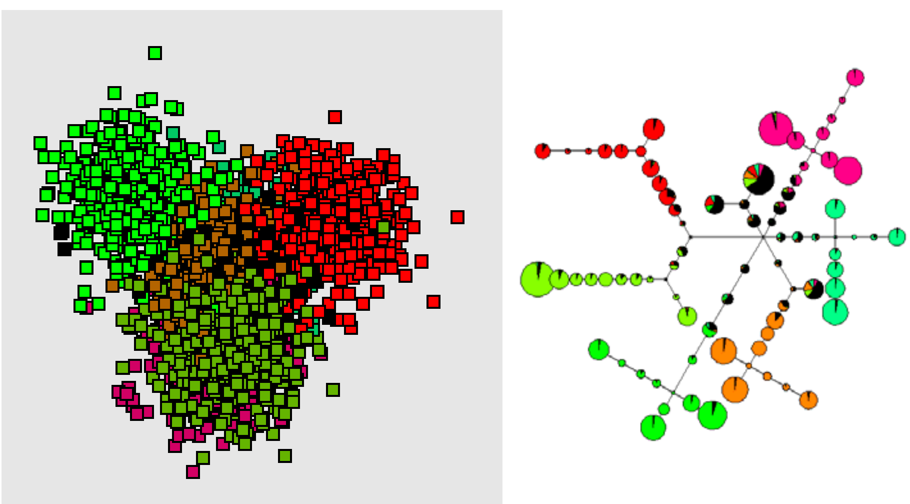




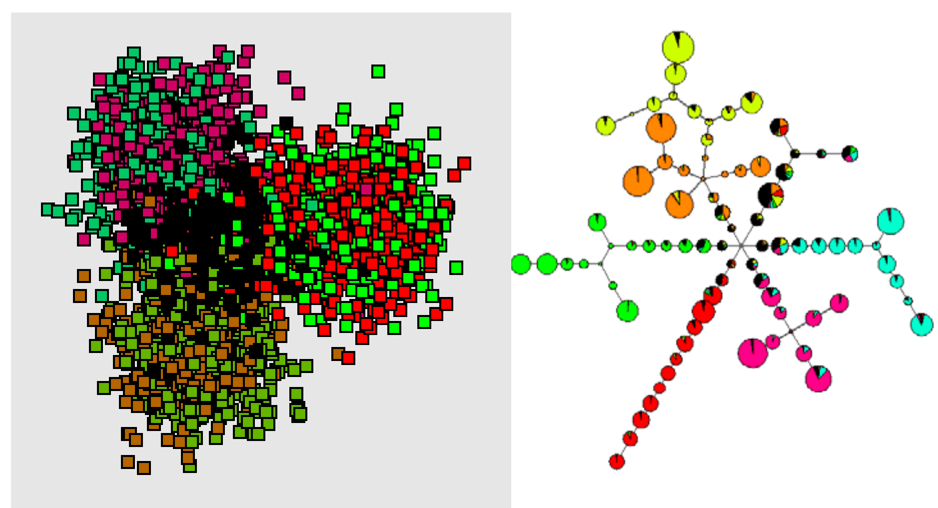
Streptomyces coelicolor



Fusobacterium nucleatum

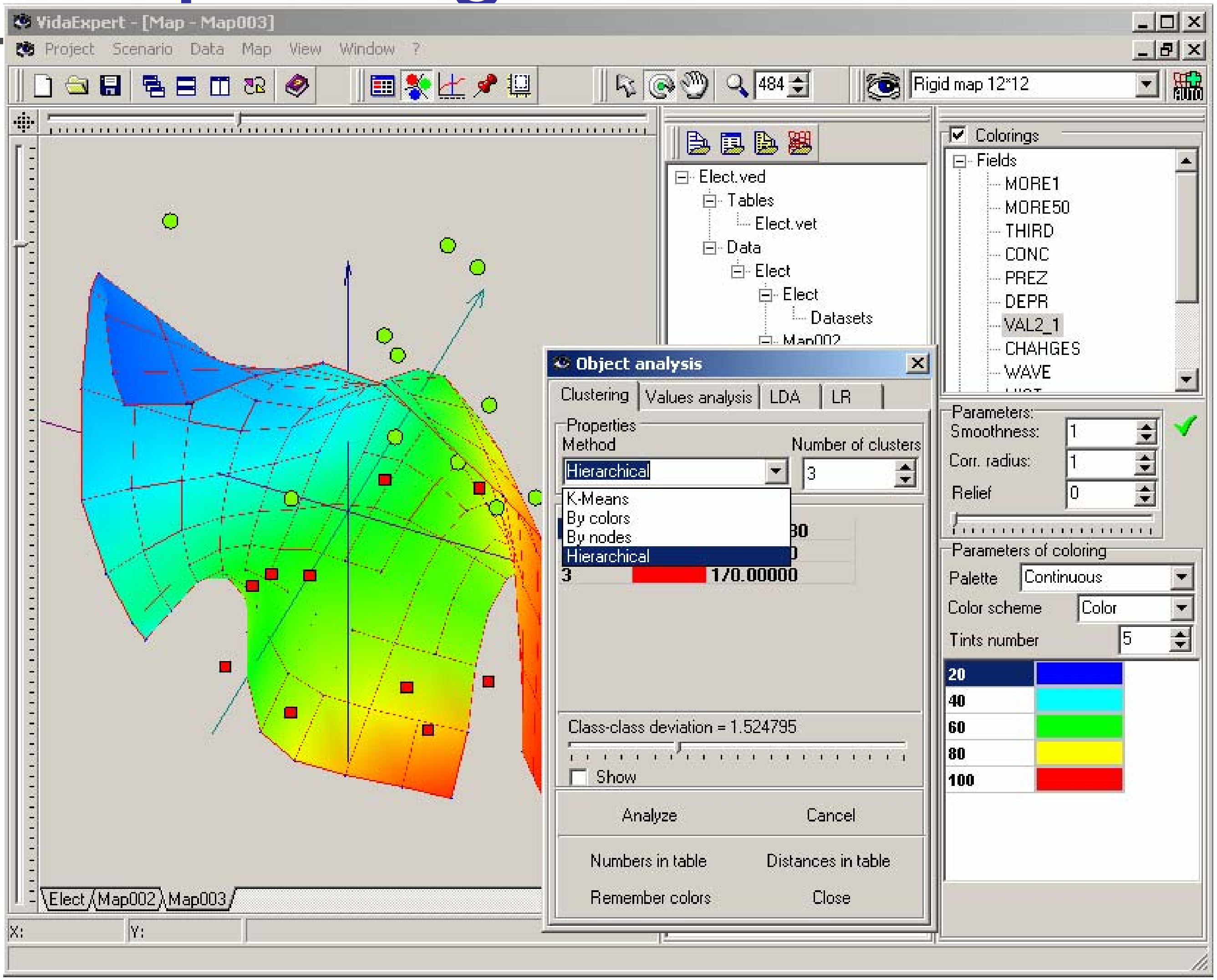


Bacillus halodurans



Ercherichia coli

VIDAExpert tool and *elmap* C++ package



Iterative error mapping

For a given elastic manifold and a datapoint $x^{(i)}$ the error vector is

$$x_{err}^{(i)} = x^{(i)} - P(x^{(i)})$$

where $P(x)$ is the projection of data point $x^{(i)}$ onto the manifold.

The errors form a new dataset, and we can construct another map, getting regular model of errors. So we have *the first* map that models the data itself, *the second* map that models errors of the first model, ... and so on. Every point x in the initial data space is modeled by the vector

$$\tilde{x} = P(x) + P_2(x - P(x)) + P_3(x - P(x) - P_2(x - P(x))) + \ldots$$

Conclusion

- Complex topology, quadratic functionals, simple algorithm.
- The whole approach can be interpreted as a intermediate between absolutely flexible neural gas and significantly more restrictive elastic map.
- It includes as the simplest limit cases the k-means clustering algorithm and classical PCA.

Useful links

- Principal components and factor analysis
  http://www.statsoft.com/textbook/stfacan.html
  http://149.170.199.144/multivar/pca.htm
- Principal curves and surfaces
  http://www.slac.stanford.edu/pubs/slacreports/slac-r-276.html
  http://www.iro.umontreal.ca/~kegl/research/pcurves/
- Self Organizing Maps
  http://www.mlab.uiah.fi/~timo/som/
  http://davis.wpi.edu/~matt/courses/soms/
  http://www.english.ucsb.edu/grad/student-pages/jdouglass/coursework/hyperliterature/soms/
- Elastic maps
  http://www.ihes.fr/~zinovyev/
  http://www.math.le.ac.uk/~ag153/homepage/

Several names

- K-means clustering: MacQueen, 1967;
- SOM: T. Kohonen, 1981;
- Principal curves: T. Hastie and W. Stuetzle, 1989;
- Elastic maps: A. Gorban, A. Zinovyev, A. Rossiev, 1996,1998;
- Polygonal models for principal curves: B. Kégl, 1999;
- Local PCA for principal curves construction: J. J. Verbeek, N. Vlassis, and B. Kröse, 2000.

Three of them are Authors

Thank you for your attention!

- Questions?