Elastic Maps
for Data Analysis

Alexander Gorban, **Leicester**
with Andrei Zinovyev, **Paris**

Plan of the talk

1. Principal manifolds and elastic maps
- The notion of of principal manifold (PM)
- Constructing PMs: elastic maps
- Adaptation and grammars

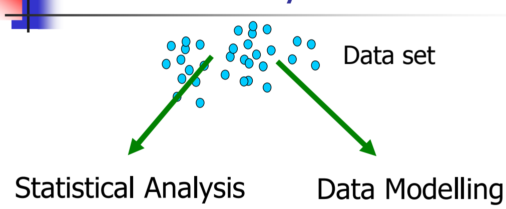
2. Application technique
- Projection and regression
- Maps and visualization of functions

3. Implementation and examples

Plan of the talk

INTRODUCTION
- Two paradigms for data analysis: statistics and modelling
- Clustering and K-means
- Self Organizing Maps
- PCA and local PCA

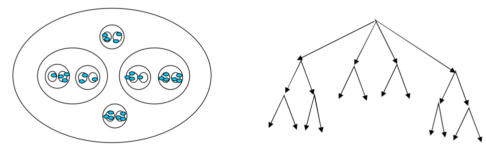
Two basic paradigms for data analysis



Statistical Analysis

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

Example: Simplest Clustering

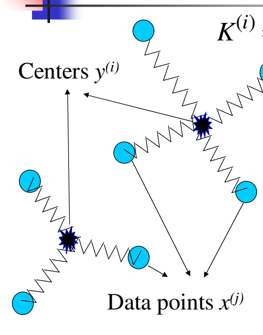


Data Modelling

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

K-means algorithm

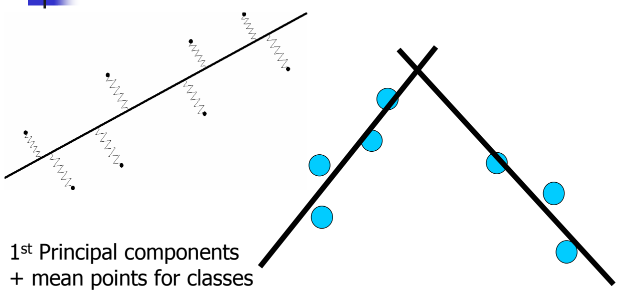


$$K^{(i)} = \{x^{(j)} : \left\| x^{(j)} - y^{(i)} \right\| \le \left\| x^{(j)} - y^{(m)} \right\| \forall m\}$$

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

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



1^{st} Principal components
+ mean points for classes

instead of simplest means

PCA and Local PCA

The covariance matrix is positive definite (X^q are datapoints)

$$\mathrm{cov}(x_i, x_j) = \frac{1}{p-1}\sum_{q=1}^{p}(X_i^q - \bar{X}_i)(X_j^q - \bar{X}_j)$$

Principal components: eigenvectors of the covariance matrix:

$$e_i, \lambda_i; \lambda_1 \geq \lambda_2 \geq \ldots \geq 0$$

The local covariance matrix (w is a positive cutting function)

$$\mathrm{cov}_y(x_i, x_j) = \frac{1}{p-1}\sum_{q=1}^{p} w(y - X^q)(X_i^q - \bar{X}_i)(X_j^q - \bar{X}_j)$$

The field of principal components: eigenvectors of the local covariance matrix, $e_i(y)$. Trajectories of these vector-fields present geometry of local data structure.

SOM - Self Organizing Maps

- Set of nodes is a finite metric space with distance $d(N,M)$;
- 0) Map set of nodes into dataspace $N \rightarrow f_0(N)$;
- 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.

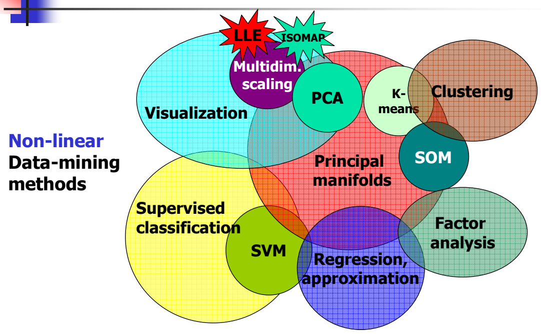
The closest node to X is moved the most in the direction of X, while other nodes are moved by smaller amounts depending on their distance from the closest node in the initial geometry.

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

(Almost) Back to Statistics:

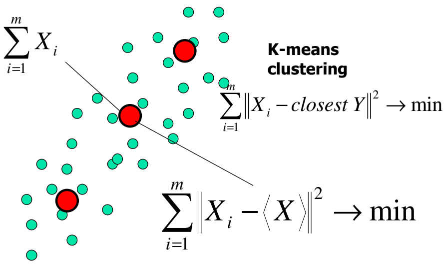
- Quasi-statistics:
  1) delete one point from the dataset,
  2) fitting,
  3) analysis of the error for the deleted data;
- The *overfitting* problem and *smoothed data points* (it is very close to non-parametric statistics)

Principal manifolds
Elastic maps framework



Mean point

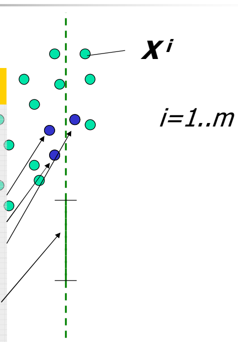
$$\langle X\rangle = \frac{1}{m}\sum_{i=1}^{m} X_i$$



Finite set of objects in R^N

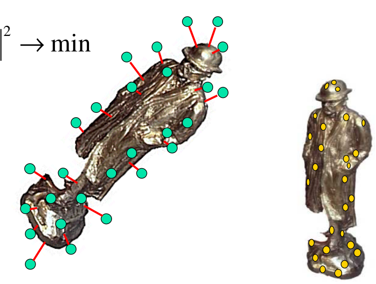
IRIS database

Petal heght	Petal width	Sepal width	Sepal height	SPECIES
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.3	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	X	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica
6.3	2.9	5.6	1.8	Iris-virginica

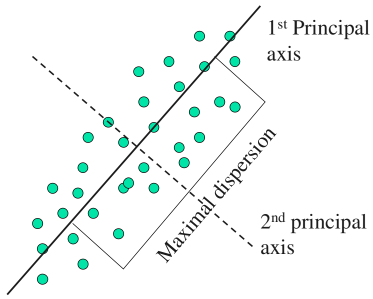


Principal "Object"

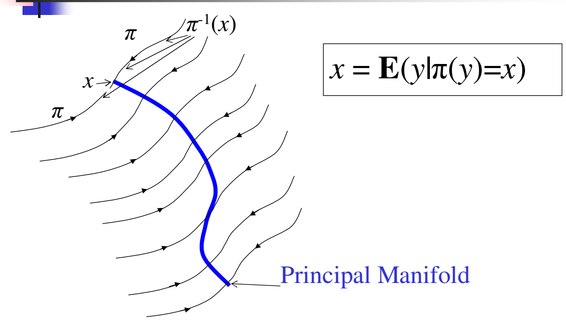
$$\sum_{i=1}^{m}\| \text{———} \|^2 \to \min$$



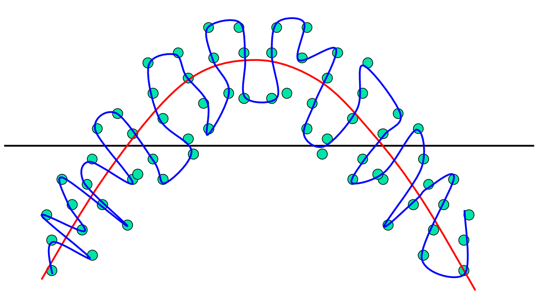
Principal Component Analysis



Statistical Self-consistency



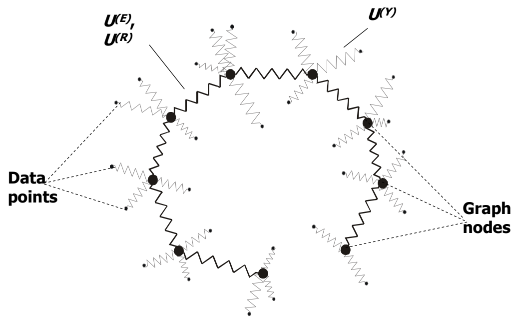
Principal manifold



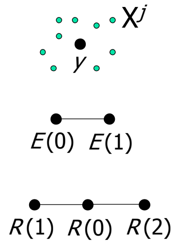
What do we want?

- 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



Definition of elastic energy



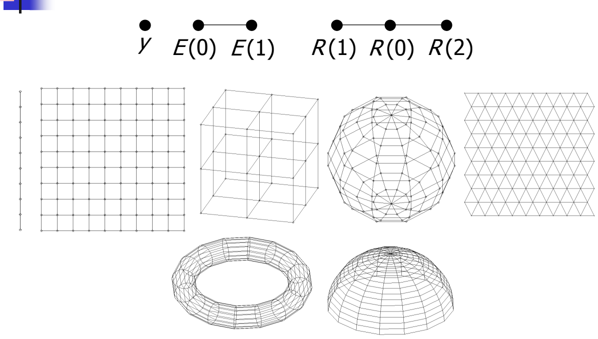
$$U^{(Y)} = \frac{1}{N}\sum_{i=1}^{p}\sum_{x^{(j)}\in K^{(i)}} \left\| X^j - y^{(i)} \right\|^2$$

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

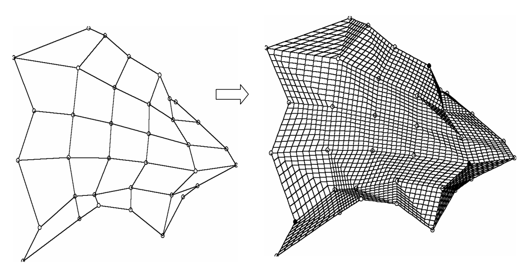
$$U^{(R)} = \sum_{i=1}^{r} \mu_i \left\| R^{(i)}(1) + R^{(i)}(2) - 2R^{(i)}(0) \right\|^2$$

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

Constructing elastic nets



Elastic manifold



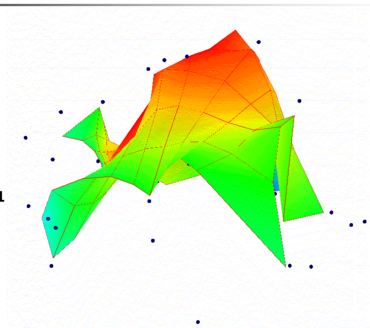
Global minimum and softening

$\lambda_0, \mu_0 \approx 10^3$

$\lambda_0, \mu_0 \approx 10^2$

$\lambda_0, \mu_0 \approx 10^1$

$\lambda_0, \mu_0 \approx 10^{-1}$



Scaling Rules

For uniform d-dimensional net from the condition of constant energy density we obtain:

$$\lambda_1 = \lambda_2 = \ldots = \lambda_s = \lambda(s);$$

$$\mu_1 = \mu_2 = \ldots = \mu_r = \mu(r)$$

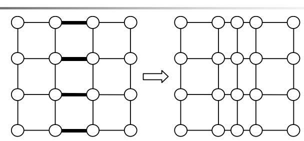
$$\lambda = \lambda_0 s^{\frac{2-d}{d}}$$

$$\mu = \mu_0 r^{\frac{4-d}{d}}$$

s is number of edges,
r is number of ribs
in a given volume

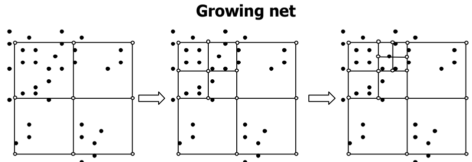
Adaptive algorithms

Refining net:



Growing net

Idea of scaling:



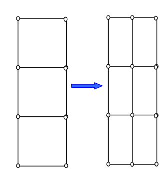
Adaptive net

Grammars of Construction

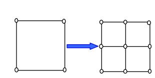
Substitution rules

Examples:

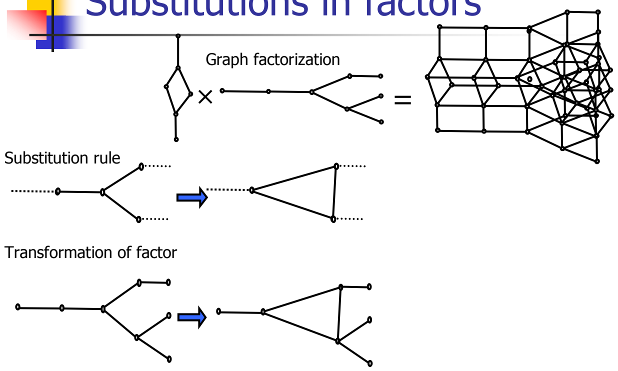
1) For net refining: substitutions of columns and rows



2) For growing nets: substitutions of elementary cells.



Substitutions in factors



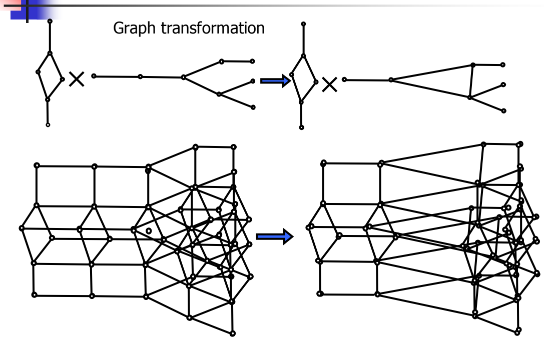
Transformation selection

A grammar is a list of elementary graph transformations.

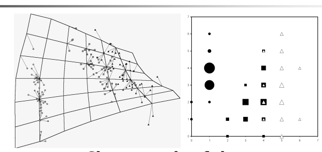
Energetic criterion: we select and apply an elementary applicable transformation that provides the maximal energy decrease (after a fitting step).

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

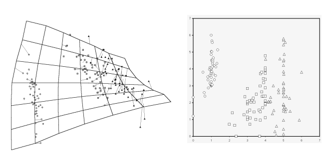
Substitutions in factors



Projection onto the manifold

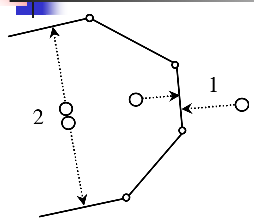


Closest node of the net



Closest point of the manifold

Mapping distortions

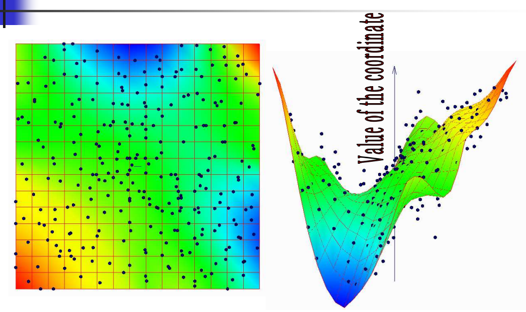


Two basic types of distortion:

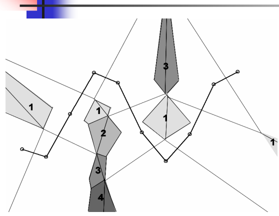
1) Projecting distant points in the close ones (**bad resolution**)

2) Projecting close points in the distant ones (**bad topology compliance**)

Colorings: visualize any function



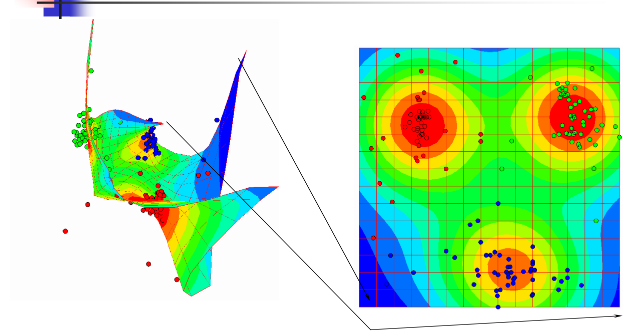
Instability of projection



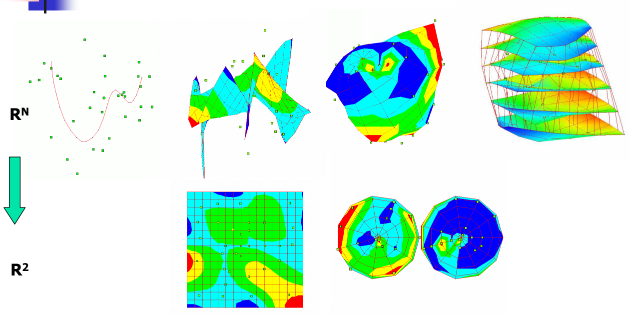
Best Matching Unit (BMU) for a data point is the closest node of the graph, BMU2 is the second-close node. If BMU and BMU2 are not adjacent on the graph, then the data point is *unstable*.

Gray polygons are the areas of instability. Numbers denote the degree of instability, how many nodes separate BMU from BMU2.

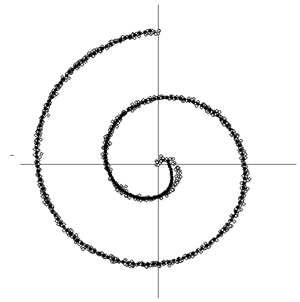
Density visualization



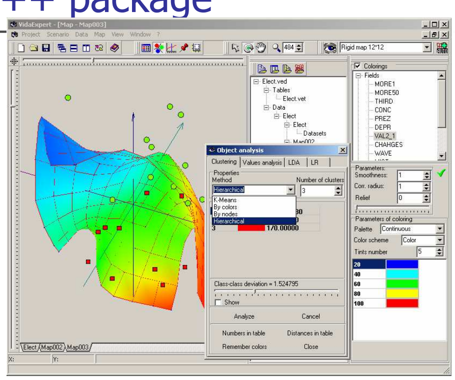
Example: different topologies



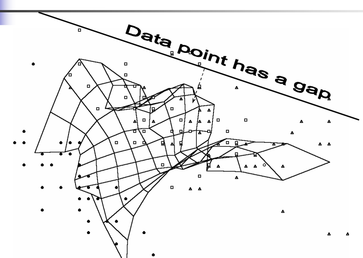
Regression and principal manifolds



VIDAExpert tool and *elmap* C++ package



Projection and regression



Data with gaps are modelled as affine manifolds, the nearest point on the manifold provides the optimal filling of gaps.

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

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

Image skeletonization or clustering around curves

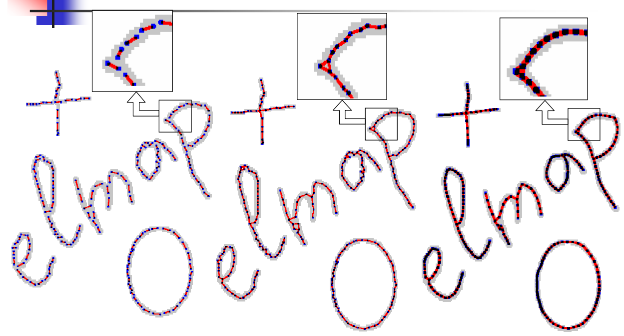
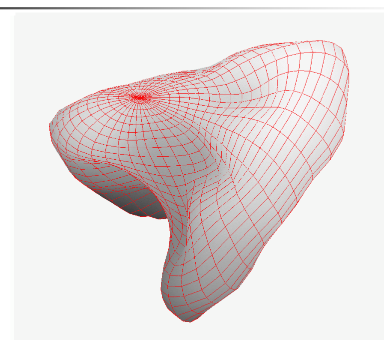
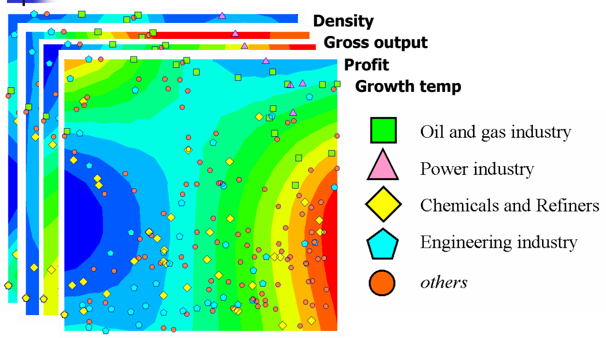


Image skeletonization or clustering around curves

Approximation of molecular surfaces

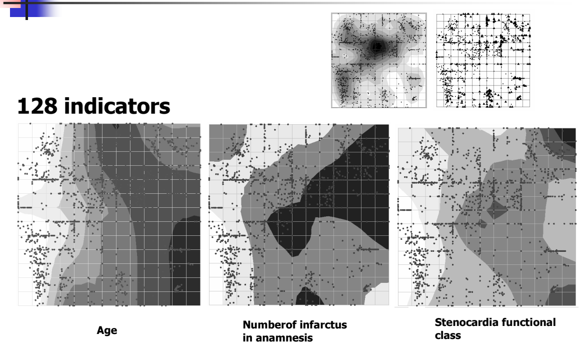


Application: economical data



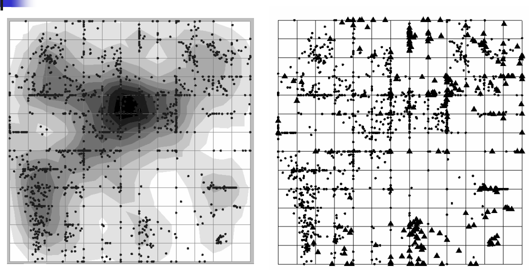
Medical table

1700 patients with **infarctus myocarde**



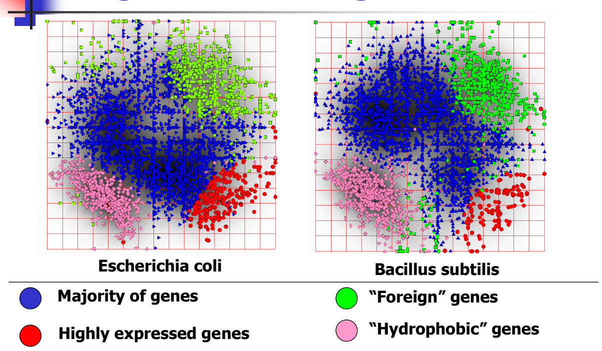
Medical table

1700 patients with **infarctus myocarde**



Patients map, density **Lethal cases**

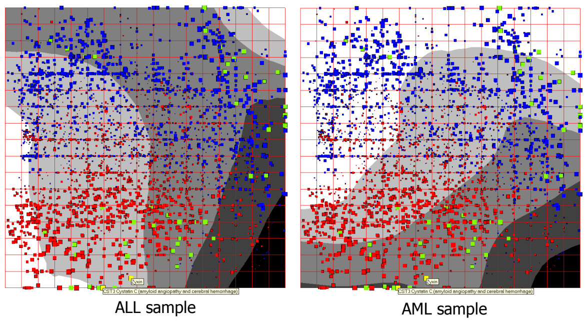
Codon usage in all genes of one genome



Golub's leukemia dataset

3051 genes, 38 samples (ALL/B-cell,ALL/T-cell,AML)

Map of genes: vote for ALL, vote for AML, used by T.Golub, used by W.Lie

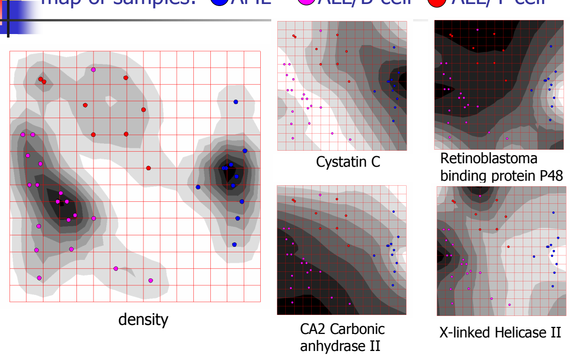


Useful links

- 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/

Golub's leukemia dataset

map of samples: AML ALL/B-cell ALL/T-cell



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, 1998;
- Polygonal models for principal curves: B. Kégl, 1999;
- Local PCA for orincipal curves construction: J. J. Verbeek, N. Vlassis, and B. Kröse, 2000.

Two of them are Authors

Thank you for your attention!

- Questions?