Linear models for data analysis

Wolfram Liebermeister
MPI für molekulare Genetik

The problem

- Data matrix X with variables x_i in rows
- Transform variables x_i to more convenient coordinates s_l

$$\vec{x} = f(\vec{s}) + \vec{\eta}$$

- Estimate transformation from the data
  (unlike e.g. smoothing, fourier or wavelet transform)

What is convenient?

- Reduce dimensionality (keeping maximal information) for
  - visualisation
  - further processing (classification, discrimination, regression)
  - storing/transmitting
- Simplify data
  - separate effects
  - simpler coding (possibly in more dimensions)
- Estimate underlying distribution
  - denoising
  - regression
  - estimate "real" underlying factors

Linear models

- Probabilistic model:

$$\vec{x} = \vec{\mu} + A\ \vec{s} + \vec{\eta}$$

  with mean $\vec{\mu}$, $<\vec{s}>= 0$, $\vec{\eta}$ independent gaussian noise

- Data transformation to *components* s

$$x_{il} = \mu_i + \sum_k A_{ik}\ s_{kl} + \eta_{il}$$

  with estimates of μ, A, η

- Centering (use empirical center of mass as estimate of μ)
  and transformation to new basis ("loadings", columns of A)
  (may be under- or overcomplete)

- matrix factorisation is underdetermined

$$AS = ATT^{-1}S = A'S'$$

  further constraints are necessary $\rightarrow$ different linear methods

Principal component analysis (PCA)

Basic idea

Explain most of the data variance by a small subspace

Calculation

- assumption: data are multivariate normal with $p \propto exp(-1/2\ x'\ \Sigma\ x)$
- estimate Σ^{-1} by empirical covariance matrix $C =< (x - \mu)(x - \mu)' >$
- C is symmetric $\rightarrow$ orthogonal eigenvectors, eigenvalues = variances
- use eigenvectors (ordered by variances) as the new basis A

Properties

- centering and rotation of the data
- solution is unique (unless different directions show the same variance)
- first n components explain as much variance as possible
- eliminate high components $\rightarrow$ linear dimension reduction with minimal loss of variance

Factor analysis

Basic idea

Estimate a small number of interpretable factors, as well as measurement noise

- underlying model
$$\vec{x} = \vec{\mu} + A\ \vec{s} + \vec{\eta}$$
with s independent gaussian (less components than variables) with unit covariance and η independent gaussian (with different variances)
- estimate factor subspace and measurement noise using the correlation matrix
- estimate significant number of factors using likelihood ratio test
- Achieve "simple structure" of loadings matrix (large vs. small values) by rotation
- "varimax" criterion: maximize sum of squared loadings

Independent component analysis (ICA) and projection pursuit

Basic idea

Find non-gaussian components with minimal statistical dependencies
Use higher-order (covariance is second-order) dependencies for the estimation

Projection pursuit (Friedman and Tukey, 1974)

- Project data to low-dimensional space such that "interesting" features (e.g. clusters) become visible
- Central limit theorem $\rightarrow$ in high dimensions, almost all (random) projections yield almost normal data
- "Interesting" means non-normal $\rightarrow$ maximise some higher-order measure of non-normality

Independent component analysis (ICA) and projection pursuit

Basic idea

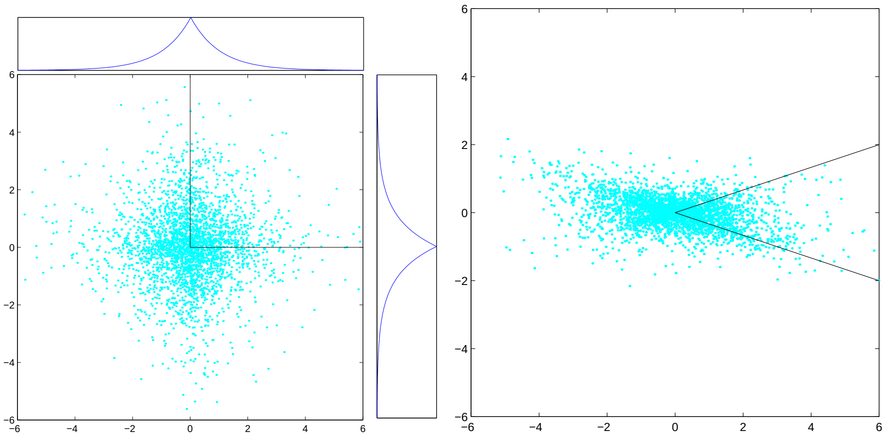
Find non-gaussian components with minimal statistical dependencies
Use higher-order (more than covariance) dependencies for the estimation

ICA

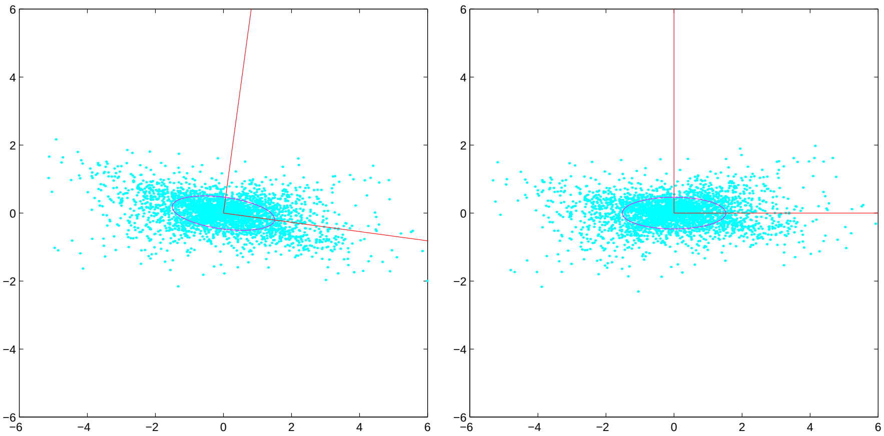
- Basic model
$$\vec{x} = \vec{\mu} + A\ \vec{s}$$
where s_k (same number as variables) are independent, but *not* gaussian (sub- or supergaussian)
- distribution $p(\vec{s}) = \Pi\ p_k(s_k)$
- Estimation:
minimize the Kullback "distance" between empirical distribution and model distribution
$\leftrightarrow$ minimize the (empirical) mutual information between components s
$\leftrightarrow$ minimize the sum of marginal entropies
- `fastica` algorithm (A. Hyvärinen):
maximize "contrast" (dissimilarity between (unknown) marginal distributions from normal)
by a gradient descent search

Illustrative example: reconstruct two Laplacian-distributed variables

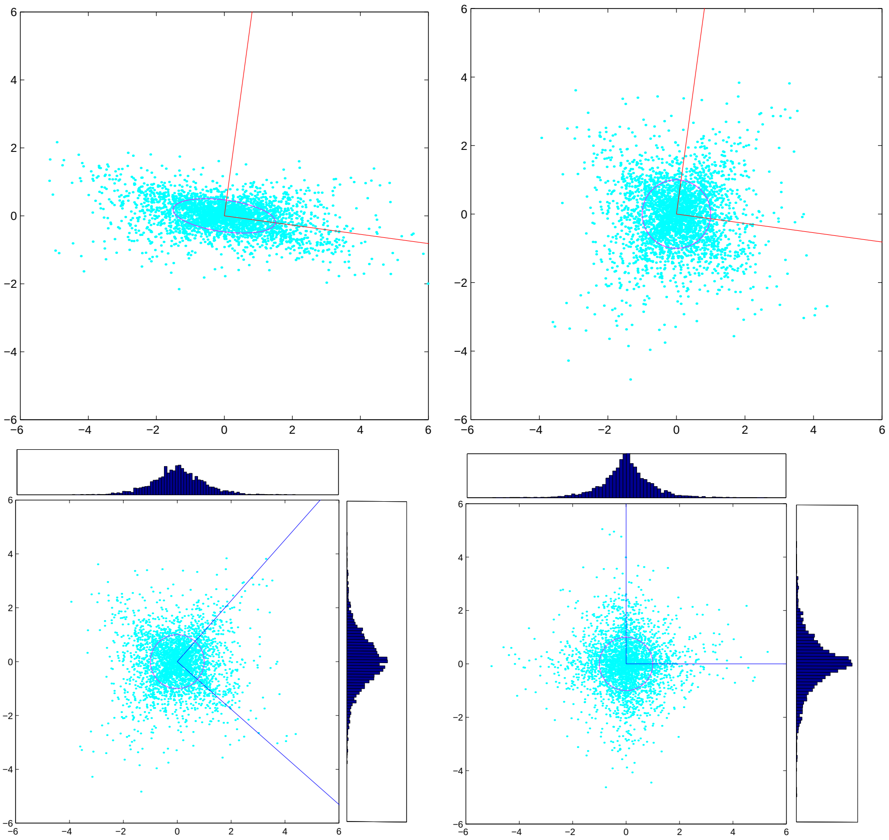
Produce artificial data:



What PCA does:



What ICA does (fastica):



Linear correlations

- ICA removes linear correlations
- With gaussian data, the solution is not unique $\rightarrow$ bad convergence

Degeneracies

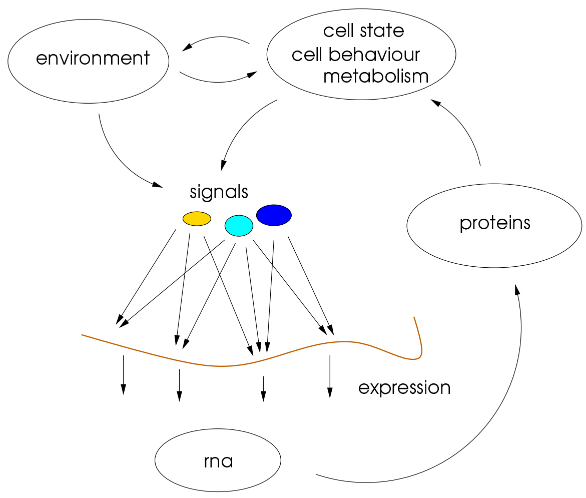
- The original variables are assumed to have zero mean.
- IC are scaled to variance=1 by convention.
- The signs can be chosen arbitrarily.
- There is no natural order of the IC (use variance, contrast, or other)

Applications

- Blind source separation
- Suited to find almost sparse components
- Noisy and overcomplete variants, and variants with priors on A exist

see
A. Hyvärinen, Survey on independent component analysis [5]
A. Hyvärinen, E. Oja, Independent component analysis: a tutorial [1]

Assumptions on gene expression



- A cell/tissue state is characterized by q variables ("expression mode levels").
- The genes' log expression levels are functions of (some of) them.
- The genes' input functions can be approximated by linear functions.

Sparseness assumption (ICA etc.)

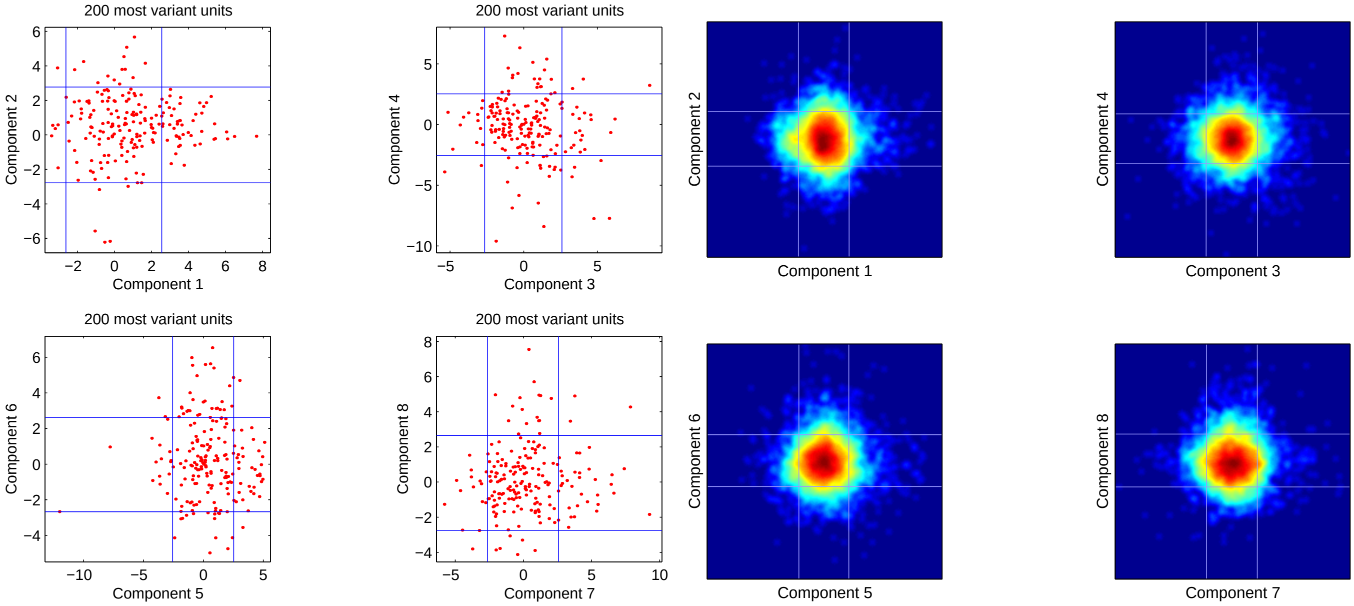
- The influence weights of different modes are approximately independent and sparse.
- If N(experiments) $>>$ N(genes)
  $\rightarrow$ use factor analysis instead

A biological example

see *H. Causton, Remodeling of yeast genome expression in response to environmental changes* [2]

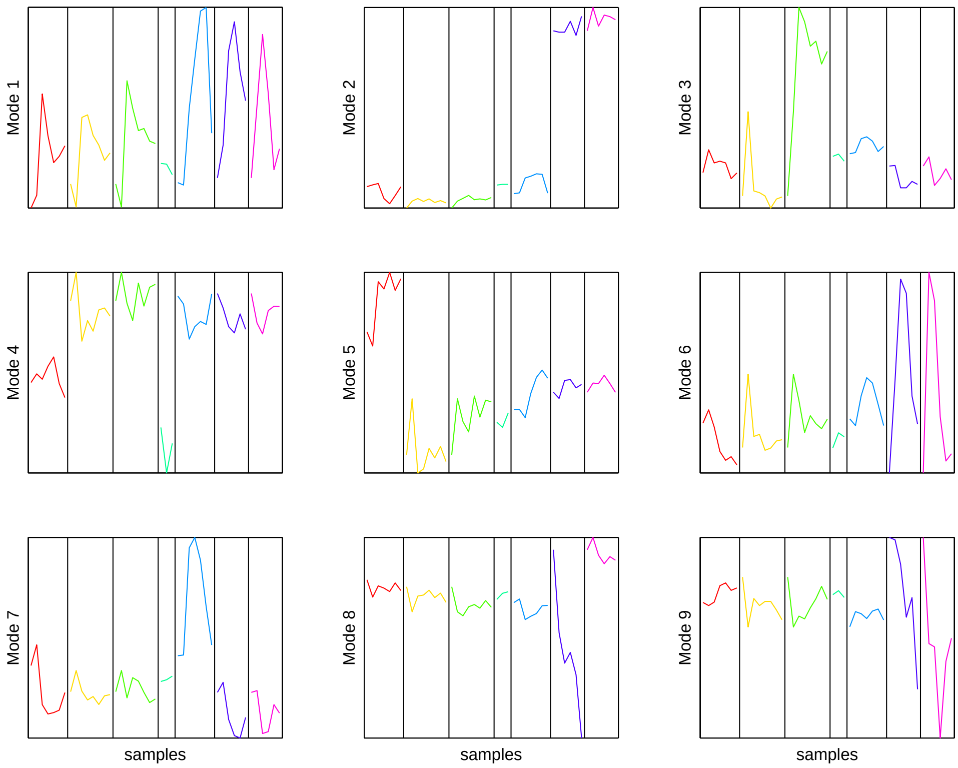
Expression in yeast after shock treatments:
heat, acid, alkali, msn 2/4 deletion + acid, hydrogen peroxide, NaCl, sorbitol

Components



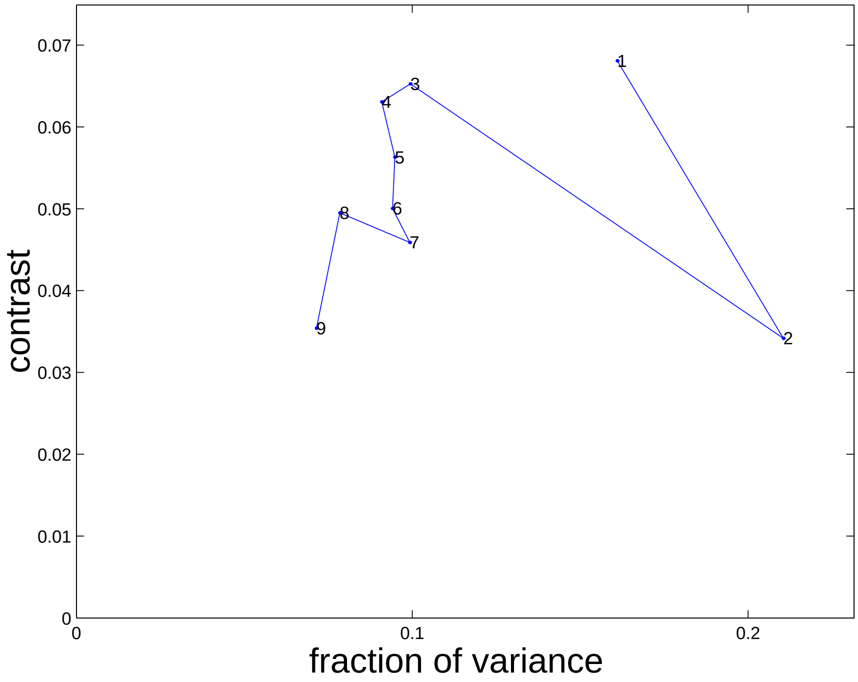
A biological example

Loadings



A biological example

Scores



Some other linear models

- Topographic ICA

  see *A. Hyvärinen et al., Topographic Independent Component Analysis* [7]
  Assume graph topology between components.
  Estimate components such that dependencies of squared data are located between neighbour components.

- Non-negative matrix factorisation

  see *D. Lee and H. Seung, Learning the parts of objects by non-negative matrix factorization* [9]
  data, loadings and components are constrained to be non-negative $\rightarrow$ almost sparse representation

- Overcomplete representations

  see *M. Lewicki, T. Sejnowski, Learning overcomplete representations* [10]
  more components than variables: prior needed to make the model identifiable
  sparse representation

- Bayesian decomposition

  see *T. D. Moloshok et al., Application of Bayesian decomposition for analysing microarray data* [11]

- ...

Only to mention some nonlinear models...

- Self-organised feature maps (SOM)

  Map data points to a discrete n-dimensional grid

- Non-linear component analysis

  see *R. Duda, P. Hart, D. Stork, Pattern classification* [3]
  5-layer neural autoencoder network (maps data to themselves)
  The (low-dimensional) middle layer represents the components.

- Nonlinear ICA

  see *Harri Lappalainen et al., Nonlinear independent component analysis using ensemble learning: experiments and discussion* [8]

  $$\vec{x} = f(\vec{s})$$

  where f represented by a neural network and s is non-gaussian and independent

References

[1] Erkki Oja Aapo Hyvärinen. Independent component analysis: a tutorial. unpublished.

[2] Helen C. Causton et al. Remodeling of yeast genome expression in response to environmental changes. *Molecular Biology of the cell*, 12:323–337, 2001.

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, 2 edition, 2001.

[4] Aapo Hyvärinen. Gaussian moments for noisy independent component analysis. unpublished.

[5] Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, pages 94–128, 1999.

[6] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492, 1997.

[7] A. Hyvrinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1525–1558, 2001.

[8] Harri Lappalainen et al. Nonlinear independent component analysis using ensemble learning: experiments and discussion.

[9] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788, 1999.

[10] Terence J. Sejnowski Michael S. Lewicki. Learning overcomplete representations. unpublished.

[11] T. D. Moloshok et al. Application of bayesian decomposition for analysing microarray data. *Bioinformatics*, 18(4):566–575, 2002.

The idea behind `fastica`

The goal:
Given the data matrix X, find a mixing matrix A
to minimize the statistical dependence between the "independent components" (rows of S).

Assumption: the joint distribution *factorizes* into a product of component distributions.

- Decompose A into
$$A = BR$$
where R is a rotation and $B = (X^T X)^{1/2}$ produces the linear correlations.
Use the decorrelated ("whitened") data.
- Statistical dependence is quantified by the **mutual information** between the components.
- mutual information is minimal *iff* entropy of the components is minimal
- entropy is approximated by a **contrast function** J_G (dissimilarity from normal distribution)
$$J_G(s) = |< G(s) > - < G(y) > |$$
where the test function G is an even, non-quadratic smooth function, y is normally distributed.
Robustness depends on the choice of G.

The fastica algorithm

see
Aapo Hyvärinen and Erkki Oja, A fast fixed-point algorithm for independent component analysis [6]

- Remove mean and linear correlations from the data matrix X:
  force $< \mathbf{x} >= 0$ and $< \mathbf{x}^T \mathbf{x} >= I$.
- Guess initial $W = A^{-1}$ with columns $\mathbf{w}$
- Iterate
  1. new $\mathbf{w} =< \mathbf{x}^T \; g(\mathbf{x}\mathbf{w}) > -\mathbf{w} < g'(\mathbf{x}\mathbf{w}) >$
     where g is the derivative of the test function G.
  2. Compute expectation values using batches of input data
  3. Orthogonalize W

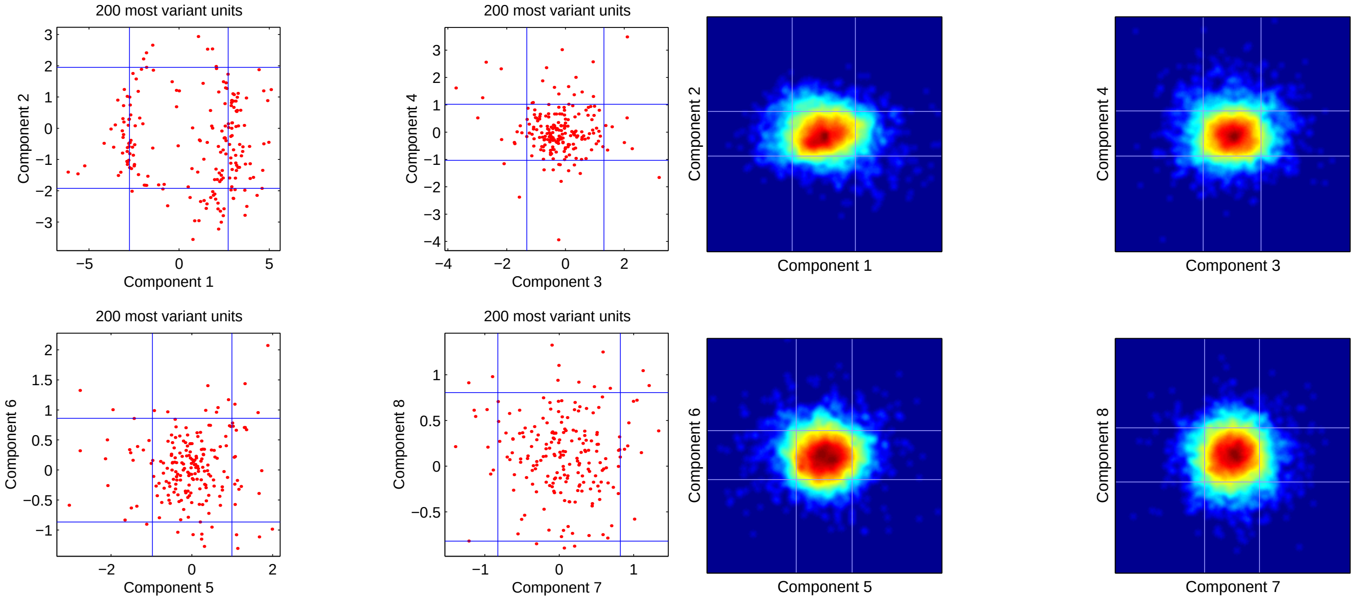
  until convergencence.

Properties of `fastica`:

- Good results for artificial data (even with moderate noise)
- Bad convergence for gaussian data
- A robust estimation of A is achieved using gaussian moments [4] as nonlinearity g.

A biological example: PCA

Components



A biological example: PCA

Loadings