

# Sparse component analysis – applications in multichannel medical imaging and bioinformatics

**Ivica Kopriva**

**Ruđer Bošković Institute**

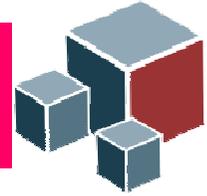
**e-mail:** [ikopriva@irb.hr](mailto:ikopriva@irb.hr) [ikopriva@gmail.com](mailto:ikopriva@gmail.com)

**Web:** <http://www.lair.irb.hr/ikopriva/>

**Acknowledgments:**

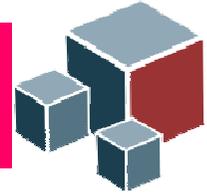
**Croatian Science Foundation** Grant 9.01/232 "Nonlinear component analysis with applications in chemometrics and pathology"

**Ministry of Science, Education and Sports, Republic of Croatia,** Grant 098-0982903-2558 "Multispectral data analysis".

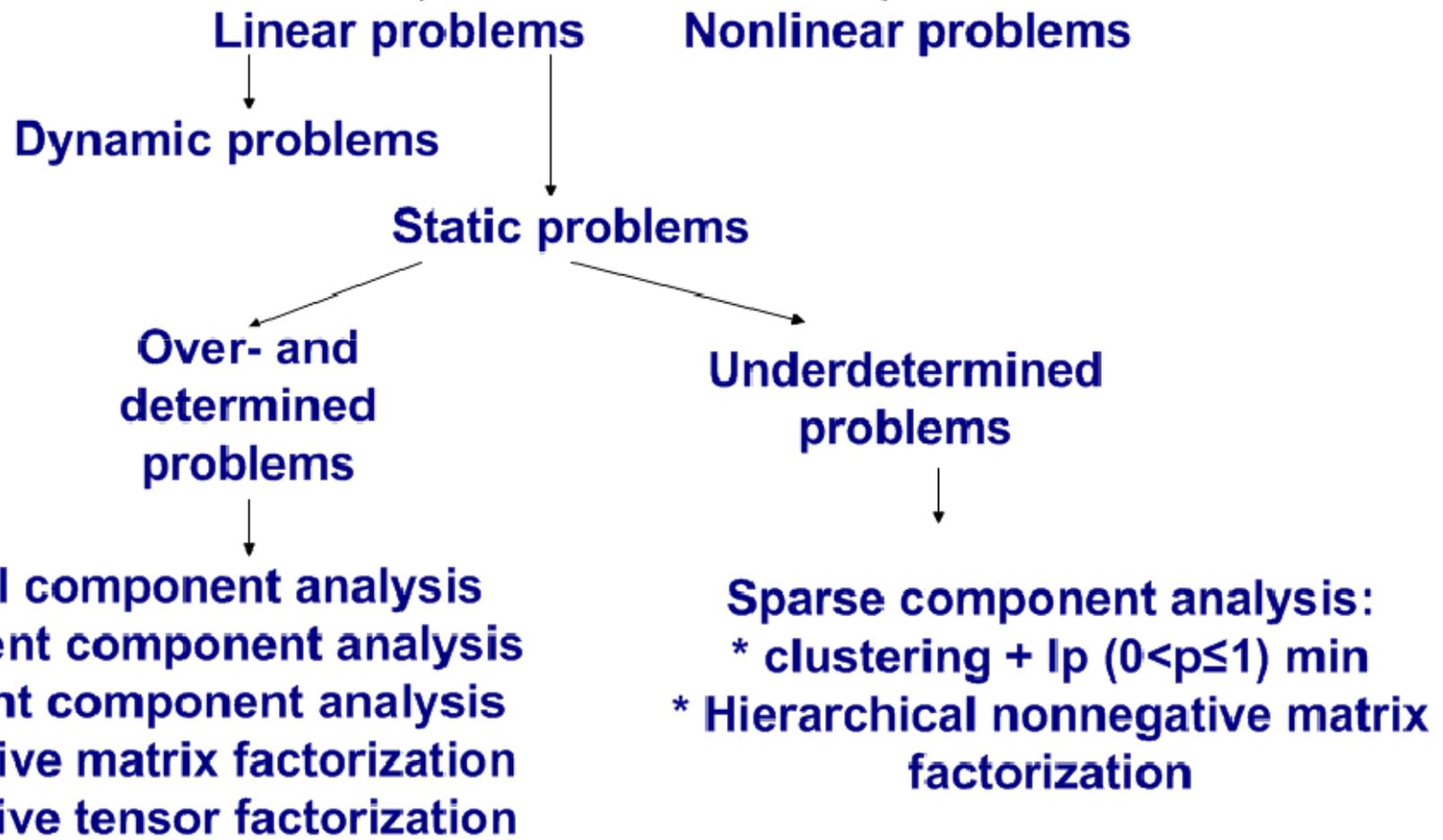


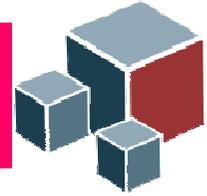
## Talk outline

- ◆ Instantaneous blind source separation (BSS):–  
problem definition and overview of main methods
- ◆ underdetermined BSS (uBSS) and sparse  
component analysis (SCA):
  - ◆ asymptotic results from compressed sensing theory
  - ◆ SCA by data clustering and  $L_p$ -norm minimization
  - ◆ SCA by sparseness constrained non-negative matrix factorization (NMF)
- ◆ Applications in multispectral and magnetic  
resonance image decomposition, chemometrics and  
bioinformatics



## Blind separation of sources





## Blind Source Separation – linear static problem

Recovery of signals from their multichannel linear superposition using minimum of a priori information i.e. multichannel measurements only.

**Problem:**

$$\mathbf{X}=\mathbf{A}\mathbf{S} \quad \mathbf{X}\in\mathbb{R}^{N\times T}, \quad \mathbf{A}\in\mathbb{R}^{N\times M}, \quad \mathbf{S}\in\mathbb{R}^{M\times T}$$

N - number of sensors/mixtures;

M - unknown number of sources

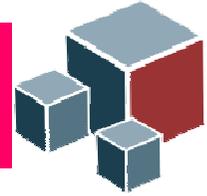
T - number of samples/observations

**Goal:** find **S**, **A** and number of sources M based on **X** only.

A. Hyvarinen, J. Karhunen, E. Oja, "Independent Component Analysis," John Wiley, 2001.

A. Cichocki, S. Amari, "Adaptive Blind Signal and Image Processing," John Wiley, 2002.

P. Comon, C. Jutten, editors, "Handbook of Blind Source Separation," Elsevier, 2010.



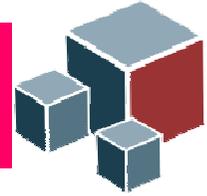
## Blind Source Separation – linear static problem

$\mathbf{X}=\mathbf{AS}$  and  $\mathbf{X}=\mathbf{ATT}^{-1}\mathbf{S}$  are equivalent for any square invertible matrix  $\mathbf{T}$ . There are infinitely many pairs  $(\mathbf{AT}, \mathbf{T}^{-1}\mathbf{S})$  satisfying linear mixture model  $\mathbf{X}=\mathbf{AS}$ . Solutions unique up to permutation and scaling indeterminacies,  $\mathbf{T}=\mathbf{P}\mathbf{\Lambda}$ , are meaningful. For such solutions constraints must be imposed on  $\mathbf{A}$  and/or  $\mathbf{S}$ .

Independent component analysis (ICA) solves BSS problem provided that: source signals  $\mathbf{S}$  are statistically independent and non-Gaussian; mixing matrix  $\mathbf{A}$  is full column rank i.e.  $M \leq N$ .

Dependent component analysis (DCA) improves accuracy of ICA when sources are not statistically independent. Linear high-pass filtering type of preprocessing transform is applied row-wise to  $\mathbf{X}$ :  $L(\mathbf{X})=\mathbf{AL}(\mathbf{S})$ . ICA is applied to  $L(\mathbf{X})$  to estimate  $\mathbf{A}$  and  $L(\mathbf{S})$ .  $\mathbf{S}$  is estimated from  $\mathbf{S} \approx \mathbf{A}^{-1}\mathbf{X}$ .

Matlab implementation of many ICA algorithms can be found in the ICALAB:  
<http://www.bsp.brain.riken.go.jp/ICALAB/>



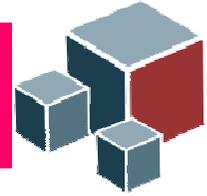
## Blind Source Separation – linear static problem

**Sparse component analysis (SCA)** solves BSS problem imposing sparseness constraints on source signals  $\mathbf{S}$ .  $M$  can be less than, equal to or greater than  $N$ .

Thus, SCA can be used to solve underdetermined BSS problems where number of source signals is greater than number of mixtures.

**Nonnegative matrix factorization (NMF)** solves BSS problem imposing nonnegativity, sparseness, smoothness or constraints on source signals. NMF algorithms that enforce sparse decomposition of  $\mathbf{X}$  can be seen as SCA algorithms.

Matlab implementation of many NMF algorithms can be found in the NMFLAB:  
<http://www.bsp.brain.riken.jp/ICALAB/nmflab.html>

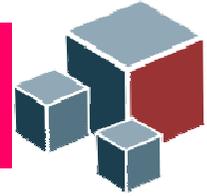


## Underdetermined BSS

- SCA-based solution of the uBSS problem is obtained in two stages:
  - 1) estimate basis or mixing matrix  $\mathbf{A}$  using data clustering.
  - 2) estimating sources, with estimated  $\mathbf{A}$ , one at a time  $\mathbf{s}_t$ ,  $t=1, \dots, T$  or simultaneously solving underdetermined linear systems of equations  $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t$ . Provided that  $\mathbf{s}_t$  is sparse enough, solution is obtained at the minimum of  $L_p$ -norm,  $\|\mathbf{s}_t\|_p$ ,  $0 \leq p \leq 1$ .

Here: 
$$\|\mathbf{s}_t\|_p = \left( \sum_{m=1}^M |s_{mt}|^p \right)^{1/p}.$$

- NMF-based solution yields  $\mathbf{A}$  and  $\mathbf{S}$  simultaneously through sparseness and nonnegativity constrained factorization of  $\mathbf{X}$ .



## When uBSS problems can(not) be solved?

Let us focus on undetermined linear system:

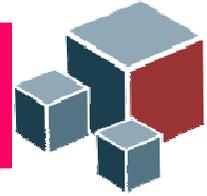
$$\mathbf{x}=\mathbf{A}\mathbf{s}, \mathbf{x}\in\mathbb{R}^N, \mathbf{s}\in\mathbb{R}^M, M>N$$

Let  $\mathbf{s}$  be  $K$ -sparse i.e.  $K=||\mathbf{s}||_0$ .

Provided that  $\mathbf{A}$  is random, with entries from Gaussian or Bernoulli distributions, compressed sensing theory has established necessary and sufficient condition on  $N$ ,  $M$  and  $K$  to obtain, with probability one, unique solution at the minimum of  $L_1$ -norm of  $\mathbf{s}$ , ref. [a]:

$$N\approx K\log(M/K)$$

a) Candès E, Tao T. Near optimal signal recovery from random projections: universal encoding strategy?. *IEEE Trans. Information Theory* 2006; **52**: 5406-5425.



## When uBSS problems can(not) be solved?

When  $L_p$ -norm of  $\mathbf{s}$  is minimized, the condition on number of measurements  $N$  is:

$$N \geq C_1(p)K + pC_2(p)K \log(M/K),$$

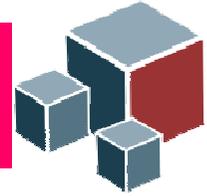
where  $C_1(p)$  and  $C_2(p)$  are norm-dependent constants, ref [a]:

a) Chartran R, Staneva V. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems* 2008; 24: 035020 (14 pages).

Hence,  $\lim_{p \rightarrow 0} N \geq C_1(0)K$ . Thus, for  $p=0$  number of measurements  $N$  does not depend on  $M$ !!!! That explains good results of  $L_0$ -norm constrained algorithms when compared against  $L_1$ -norm constrained algorithms when  $K$  is increasing, ref [b, c]:

b) Pehaz R, Pernkopf, F. Sparse nonnegative matrix factorization with  $\ell^0$ -constraints. *Neurocomputing*. 2012; **80**: 38-46.

c) Mohimani H, Babie-Zadeh B, Jutten C. A Fast Approach for Overcomplete Sparse Decomposition Based on  $\ell_0$  Smoothed Norm. *IEEE Trans. Sig. Proc.* 2009; **57**(1): 289-301.



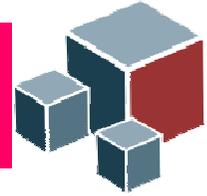
## When uBSS problems can(not) be solved?

However in BSS problems  $\mathbf{A}$  is not random matrix but deterministic matrix with a structure. For example, in multispectral imaging it contains spectral profiles of the objects/materials present in the image, ref. [a]:

a) Kopriva I, Cichocki A. Blind decomposition of low-dimensional multi-spectral image by sparse component analysis. *J. Chemometrics* 2009; **23** (11): 590-597.

In chemometrics  $\mathbf{A}$  contains concentration profiles of pure components present in the mixtures, ref. [b]:

b) Kopriva I, Jerić I. Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis. *Anal. Chem.* 2010; **82**: 1911-1920.



## When uBSS problems can(not) be solved?

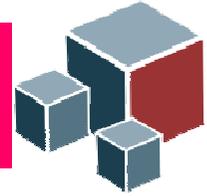
To the best of our knowledge, the only known result for deterministic  $\mathbf{A}$  is given in ref. [a]:

a) DeVore R. A. Deterministic constructions of compressed sensing matrices. *Journal of Complexity* 2007; **23**: 918-925.

For cyclic polynomial matrix  $\mathbf{A}$  it applies  $N=O(K^2)$ . That is significantly worse than  $N \approx K \log(M/K)$  for random  $\mathbf{A}$ .

$K$  corresponds with number of sources that are active/present at the specific coordinate  $t$  (time, pixel,  $m/z$  variable, frequency, etc).

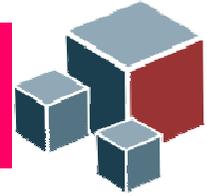
Thus,  $K$  is application dependent.



## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

- Signal  $\mathbf{s}$  is  $K$ -sparse if it has  $K$  non-zero components, i.e.  $K = \|\mathbf{s}\|_0$ .
- If uBSS problem is not sparse in original domain it ought to be transformed in domain where enough level of sparseness can be achieved:  $T(\mathbf{x}) = \mathbf{A}T(\mathbf{s})$ .
- Time-frequency and time-scale (wavelet) bases are employed for this purpose quite often.
- In addition to sparseness requirement on  $\mathbf{s}$  certain degree of incoherence of the mixing matrix  $\mathbf{A}$  is required as well. Mutual coherence is defined as the largest absolute and normalized inner product between different columns in  $\mathbf{A}$ , what reads as

$$\mu\{\mathbf{A}\} = \max_{1 \leq i, j \leq M \text{ and } i \neq j} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\| \|\mathbf{a}_j\|}$$



## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

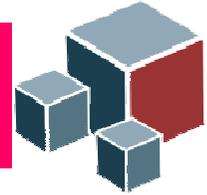
The mutual coherence provides a worst case measure of similarity between the basis vectors. It indicates how much two closely related vectors may confuse any pursuit algorithm (solver of the underdetermined linear system of equations). The worst-case perfect recovery condition for  $\mathbf{s}$  relates sparseness requirement on  $\mathbf{s}$  and coherence of  $\mathbf{A}$ , ref. [a,b]:

$$\|\mathbf{s}\|_0 < \frac{1}{2} \left( 1 + \frac{1}{\mu\{\mathbf{A}\}} \right)$$

- a) R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory* **49**, 3320-3325 (2003).
- b) J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory* **50**, 2231-2242 (2004).

**In:** I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS, a re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol.45, no.3, pp. 600–616, Mar. 1997.

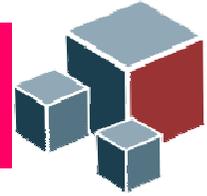
another uniqueness theorem has been stated. If  $\mathbf{A}$  has unique representation property, that is if all  $N \times N$  sub-matrices are full rank, the unique solution of  $\mathbf{x} = \mathbf{A}\mathbf{s}$  exists if:  $\|\mathbf{s}\|_0 \leq N/2$ .



## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

In blind source separation scenario properties of the mixing matrix  $\mathbf{A}$  can not be predefined i.e. they are problem dependent. Yet,  $\mathbf{A}$  dictates a level of sparseness of  $\mathbf{s}$  that is necessary to obtain possibly unique solution of the uBSS problem:  $\mathbf{x}=\mathbf{A}\mathbf{s}$ . To obtain such solution it is necessary to:

- estimate  $\mathbf{A}$  as accurately as possible.
- find representation (transformation)  $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$  where  $T(\mathbf{s})$  is as sparse as possible.
- construct algorithms for solving underdetermined system of equations  $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$  that are robust with respect to the presence of noise i.e. errors in sparse approximation of  $T(\mathbf{s})$ :  $T(\mathbf{s})$  is approximately K-sparse with  $K$  dominant and number of small coefficients. If possible performance of the algorithm should remain robust if  $K$  increases.



## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

Solving underdetermined system of linear equations  $\mathbf{x} = \mathbf{A}\mathbf{s}$  amounts to solving:

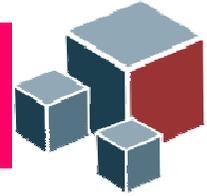
$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t.} \quad \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1, \dots, T$$

or for problems with noise or approximation error:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \frac{1}{2} \|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\|_2^2 + \lambda \|\mathbf{s}(t)\|_0 \quad \forall t = 1, \dots, T$$

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t.} \quad \|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\|_2^2 \leq \varepsilon \quad \forall t = 1, \dots, T$$

Direct minimization of  $L_0$ -norm of  $\mathbf{s}$  is combinatorial problem that is NP-hard. For larger dimension  $M$  it becomes computationally infeasible.



## uBSS – $L_1$ norm minimization

Replacement of  $L_0$ -norm by  $L_1$ -norm is done quite often. That is known as convex relaxation of the minimum  $L_0$ -norm problem. It leads to linear program:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \sum_{m=1}^{\hat{M}} s_m(t) \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1, \dots, \quad \text{s.t. } \mathbf{s}(t) \geq 0$$

$L_1$ -regularized least square problem ref.[a,b]:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \frac{1}{2} \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 + \lambda \left\| \mathbf{s}(t) \right\|_1 \quad \forall t = 1, \dots, T$$

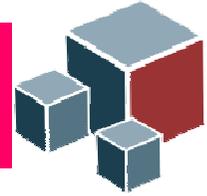
and  $L_2$ -regularized linear problem [b,c]:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \left\| \mathbf{s}(t) \right\|_1 \quad \text{s.t. } \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 \leq \varepsilon \quad \forall t = 1, \dots, T$$

a) S..J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale  $L_1$ -Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing* **1**, 606-617 (2007), [http://www.stanford.edu/~boyd/l1\\_ls/](http://www.stanford.edu/~boyd/l1_ls/).

b) E. van den Berg, M.P. Friedlander, "Probing the Pareto Frontier for Basis Pursuit Solutions," *SIAM J. Sci. Comput.* **31**, 890-912 (2008).

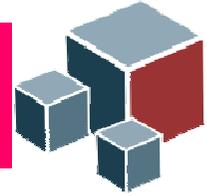
c) M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE Journal on Selected Topics in Signal Processing* **1**, 586-597 (2007).



## uBSS – $L_1$ norm minimization

Provided that prior on  $\mathbf{s}(t)$  is Laplacian, maximum likelihood approach to maximization of posterior probability  $P(\mathbf{s}|\mathbf{x},\mathbf{A})$  yields minimum  $L_1$ -norm as the solution:

$$\begin{aligned}
 \hat{\mathbf{s}}(t) &= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{s}(t) \mid \mathbf{x}(t), \hat{\mathbf{A}}\right) \\
 &= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{x}(t) \mid \mathbf{s}(t), \hat{\mathbf{A}}\right) P(\mathbf{s}(t)) \\
 &\propto \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P(\mathbf{s}(t)) \\
 &= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \exp-\left(\left|\mathbf{s}_1(t)\right| + \dots + \left|\mathbf{s}_M(t)\right|\right) \\
 &= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left|\mathbf{s}_1(t)\right| + \dots + \left|\mathbf{s}_M(t)\right| \\
 &= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left\|\mathbf{s}(t)\right\|_1
 \end{aligned}$$



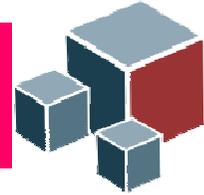
## uBSS – $L_1$ norm minimization

Sequence of MATLAB commands for solution of the problem  $\mathbf{x}=\mathbf{A}\mathbf{s}$  using command `linprog`:

```
% Linear programming solution
% solves linear program min(x) f'*x s.t. Ax=b, lb<=x<=ub.
f = ones(M,1);
lb = zeros(M,1);
ub = 1000*ones(M,1);

for m=1:T
    x=X(:,m);
    [sh,fval,exitflag,output]=linprog(f,[],[],A,x,lb,ub,[]);
    SH(:,m)=sh;
end
```

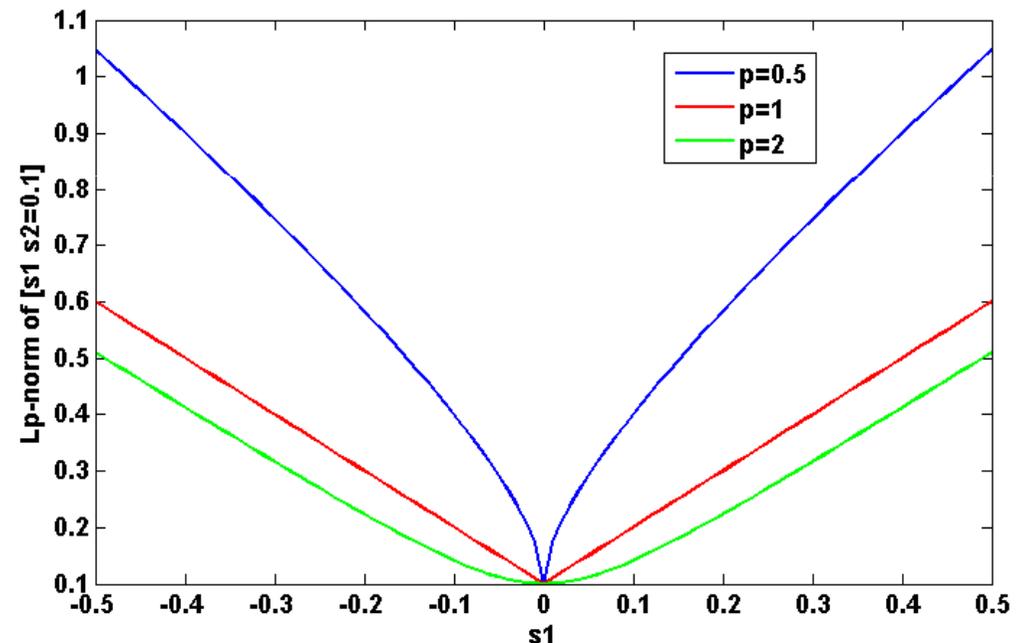
What happens if  $P(\mathbf{s})$  is not Laplacian? For distributions  $P(\mathbf{s})$  sparser than Laplacian, minimum  $L_1$ -norm approach will not yield the sparsest solution!!!!



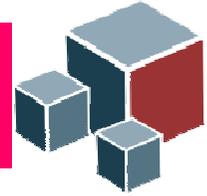
## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

Minimizing  $L_p$ -norm,  $0 < p < 1$ , of  $\mathbf{s}$  yields better performance when solving underdetermined system  $\mathbf{x} = \mathbf{A}\mathbf{s}$  than when using  $L_1$ -norm minimization.

This occurs despite the fact that minimization of  $L_p$ -norm,  $0 < p < 1$  is non-convex problem. Yet, in practical setting (when noise or approximation errors are present) its local minimum can be smaller than global minimum of  $L_1$  i.e.  $\min L_p$ -norm solution is sparser than  $\min L_1$ -norm solution.



$$L_p\text{-norm of } [s_1 \ 0.1] : \quad \|\mathbf{s}\|_p = \left( \sum_{m=1}^M |s_m|^p \right)^{1/p}$$



## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

The idea of ref. [a] was to replace  $L_0$ -norm by continuous parametric approximation:

$$\|\mathbf{s}\|_0 \approx M - F_\sigma(\mathbf{s})$$

where:

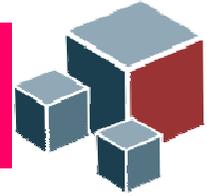
$$F_\sigma(\mathbf{s}) = \sum_m f_\sigma(s_m)$$

and:

$$f_\sigma(s_m) = \exp\left(-\frac{s_m^2}{2\sigma^2}\right)$$

approximates indicator function of a set  $\{0\}$ .

a) H. Mohimani, M. Babaie-Zadeh, C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed  $L_0$  norm," *IEEE Trans. Signal Process.* **57** (2009) 289-301.



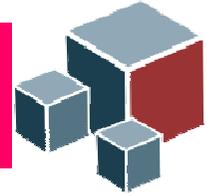
## uBSS – $L_p$ norm minimization: $0 < p \leq 1$

Smaller parameter  $\sigma$  brings us closer to  $L_0(\mathbf{s})$ , while larger  $\sigma$  yields smoother approximation that is easier to optimize.

Minimizing approximation of  $L_0(\mathbf{s})$  is equivalent to maximize  $F_\sigma(\mathbf{s})$ . The idea is to maximize  $F_\sigma(\mathbf{s})$  for large  $\sigma$  and then use obtained solution as initial value for next maximization of  $F_\sigma(\mathbf{s})$  for smaller  $\sigma$ .

After each iteration computed approximation of  $\mathbf{s}$  is projected back onto the constraining set  $\mathbf{A}\mathbf{s}=\mathbf{x}$ :

$$\mathbf{s} \leftarrow \mathbf{s} - \mathbf{A}^T \left( \mathbf{A} \mathbf{A}^T \right)^{-1} \left( \mathbf{A} \mathbf{s} - \mathbf{x} \right)$$



## Iteratively reweighted least square (IRLS) algorithm outline

$$\min \| \mathbf{s} \|_p \quad s.t. \quad \mathbf{A} \mathbf{s} = \mathbf{x} \quad \rightarrow \quad \min \sum_{m=1}^M w_m s_m^p$$

**Initialize:**  $\varepsilon=1$ ,  $\mathbf{s}^{(0)} = \text{pinv}(\mathbf{A})\mathbf{x}$ ,  $k=1$ .

**do**

**repeat**

$$w_m = \left( \left( s_m^{(k-1)} \right)^2 + \varepsilon \right)^{p/2-1}$$

$$\mathbf{Q}_k = \text{diag} \{ 1 / w_m \}$$

$$\mathbf{s}^{(k)} = \mathbf{Q}_k \mathbf{A}^T \left( \mathbf{A} \mathbf{Q}_k \mathbf{A}^T \right)^{-1} \mathbf{x}$$

$$k = k + 1$$

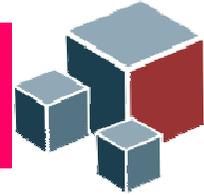
**until**  $\| \mathbf{s}^{(k)} - \mathbf{s}^{(k-1)} \|_2 < \sqrt{\varepsilon} / 100$

$\varepsilon = \varepsilon / 10$

**while**  $\varepsilon > 10^{-8}$

R. Chartrand, Exact reconstructions of sparse signals via nonconvex minimization, *IEEE Signal Process. Let.*, **14** (2007), 707-710.

I. Daubechies, R. Devore, M. Fornassier, C. S. Gunturk “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. **LXIII** (2010) 1-38.



## Iterative soft/hard thresholding

$L_1$ -regularized least square problem:

$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \frac{1}{2} \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 + \lambda \left\| \mathbf{s}(t) \right\|_1 \quad \forall t = 1, \dots, T$$

can be reformulated within analytic soft thresholding representation theory [a, b]:

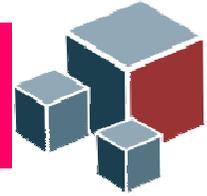
$$B(\mathbf{s}^{(k)}(t)) = \mathbf{s}^{(k)} + \mathbf{A}^T \left( \mathbf{x}(t) - \mathbf{A}\mathbf{s}^{(k)}(t) \right)$$

$$s_m^{(k+1)}(t) = \begin{cases} B(\mathbf{s}^{(k)}(t))_m - \text{sign}(B(\mathbf{s}^{(k)}(t))_m)\lambda / 2, & |B(\mathbf{s}^{(k)}(t))_m| > \lambda / 2 \\ 0, & \text{otherwise} \end{cases}$$

where  $\lambda = \sigma^2$  provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

a) D. L. Donoho, Denoising by soft-thresholding, *IEEE Trans. Information Theory*, **41** (1995), 613-627.

b) I. Daubechies, M. Defrise, D.M. Christine, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Comm. Pure and Appl. Math.*, **LVII** (2004) 1413-1457.



## Iterative soft/hard thresholding

$L_0$ -regularized least square problem:

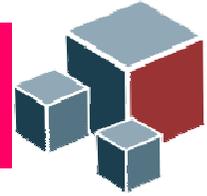
$$\hat{\mathbf{s}}(t) = \arg \min_{\mathbf{s}(t)} \frac{1}{2} \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 + \lambda \left\| \mathbf{s}(t) \right\|_0 \quad \forall t = 1, \dots, T$$

can be reformulated within analytic hard thresholding representation theory [a]:

$$s_m^{(k+1)}(t) = \begin{cases} s_m^{(k)}(t) - \text{sign}(s_m^{(k)}(t))\lambda / 2, & |s_m^{(k)}(t)| > \lambda / 2 \\ 0, & \text{otherwise} \end{cases}$$

where  $\lambda = \sigma^2$  provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

a) R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, *Inverse Problems*, **24** (2008) 1-14.



## Comparative performance analysis

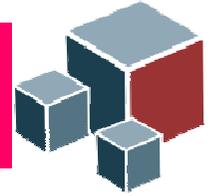
Several best algorithms for solving nonnegative uBSS problem:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}$$

were compared, whereas  $N=350$ ,  $M=1000$ ,  $T=1000$ ,  $\text{SNR [dB]} \in \{10, \text{inf}\}$ ,  $K=100$ .

Each source was generated with probability of being zero equal to 0.9. The nonzero state was generated by uniform distribution on interval  $(0,1]$ .

Entries of the  $N \times M$  mixing matrix were drawn from uniform distribution on interval  $(0,1]$ . The mixing matrix has been column normalized to unit  $L_2$  norm.

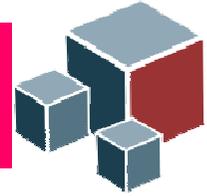


## Comparative performance analysis

Performance measure:

$$\varepsilon(\mathbf{S}) = 10 \log_{10} \left( \frac{\| \text{diag}(\mathbf{R}_{\text{SS}}) \|_2^2}{\| \text{diag}(\mathbf{R}_{\text{SS}} - \mathbf{R}_{\text{S}\hat{\text{S}}}) \|_2^2} \right)$$

has been averaged after 100 runs, where  $\mathbf{R}_{\text{SS}}$  and  $\mathbf{R}_{\text{S}\hat{\text{S}}}$  are respectively autocovariance and cross-covariance matrices calculated after permutation indeterminacy has been resolved.



## Comparative performance analysis

The following algorithms were compared:

Smoothed  $L_0$  (SL0) algorithm:

H. Mohimani, M. Babaie-Zadeh, C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed  $L_0$  norm," *IEEE Trans. Signal Process.* **57** (2009) 289-301. <http://ee.sharif.ir/~SLzero/>

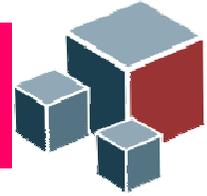
Improved SL0 algorithm (ISL0) for problems where noise is significant:

M. Hayder, K. Mahata, "An Improved Smoothed Approximation Algorithm for Sparse Representation," *IEEE Trans. Sig. Proc.*, vol. **58**, No. 4, pp. 2194-2205, 2010.

The ISL0 improves performance of SL0 when noise is negligible.

Iterative recursive least squares (IRLS):

R. Chartrand, W. Yin, "Iteratively Reweighted Algorithms for Compressive Sensing," *IEEE ICASSP*, pp. 3869-3872, 2008.



## Comparative performance analysis

### Orthogonal matching pursuit (OMP) algorithm:

M. Elad, „Orthogonal Matching Pursuit,” section 3.1.2 in *Sparse and Redundant Representations*, Springer, 2010.

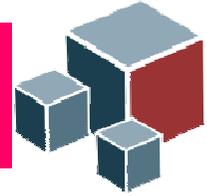
Y.C. Pati, R. Rezaifar, P.S. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

### Forward Backward (FoBa) algorithm:

N. B. Karahanoglu, H. Erdogan, "Forward-Backward Search for Compressed Sensing Signal Recovery," *Proc. EUSIPCO 2012*, pp. 1429-1433, Bucharest, Romania, August 27-31, 2012.

<http://www.lair.irb.hr/ikopriva/data.html>

This is the greedy algorithm that combines forward (adding atoms to signal support) and backward (removing atoms from signal support) selection/correction steps. Thereby the size of forward and backward steps may be greater than 1. Sparsity level  $K$  is not required to be known/specified. Only difference between size of forward and backward steps is required.



## Comparative performance analysis

### Fast Iterative Shrinkage Thresholding (Fast\_IST) algorithm:

Beck, M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Image. Sci.*, Vol. **2**, No. 1, pp. 183-202, 2009. <http://ie.technion.ac.il/Home/Users/becka.html>

This algorithm uses  $L_1$ -based regularization of least square approximation problem.

### Sparse Bayesian Learning (SBL) algorithm:

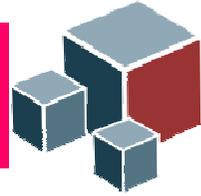
Z. Zhang, B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning", *IEEE J. Selected Topics Sig. Proc.*, (2011), **5**: 912-926. <http://dsp.ucsd.edu/~zhilin/TMSBL.html>

### Bernoulli-Gaussian approximate message passing algorithm (BG-GAMP):

J. Vila, P. Schniter, "Expectation-maximization Bernoulli-Gaussian approximate message passing", *Proc. Asilomar Conf. on Signals, Systems, and Computers* (Pacific Grove, CA), Nov. 2011.

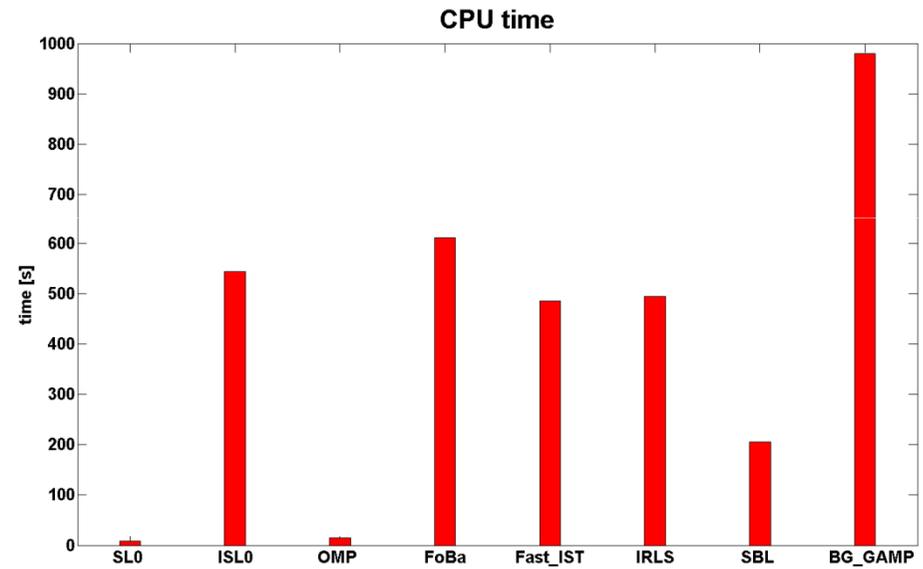
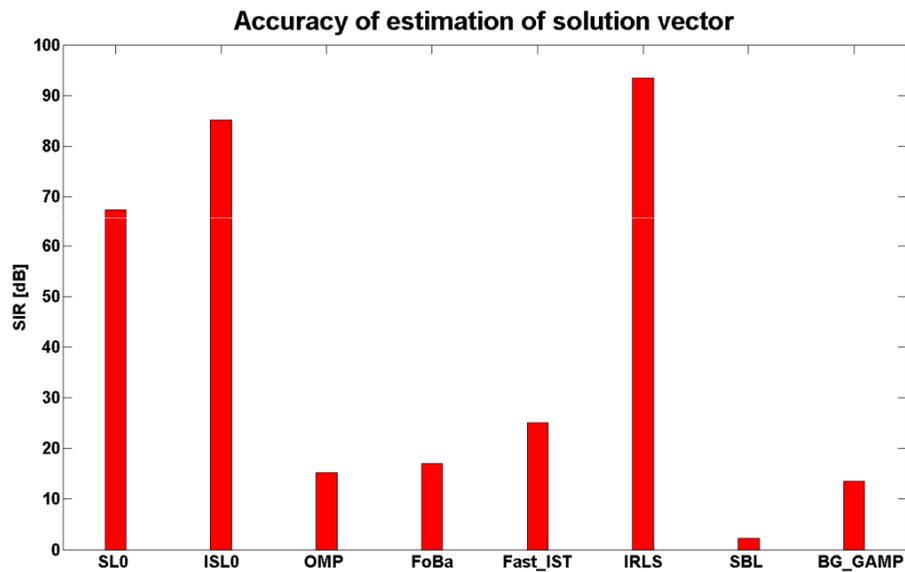
Source signal is modeled as Bernoulli-Gaussian with unknown sparsity, mean and variance, and noise is modeled as Gaussian with unknown variance. Hence, the BG-GAMP algorithm. Expectation maximization is used to learn the signal and noise parameters as algorithm iterates.

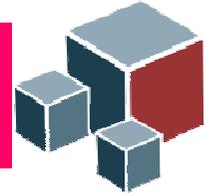
<http://www2.ece.ohio-state.edu/~vilaj/EMBGAMP/EMBGAMP.html>



## Comparative performance analysis

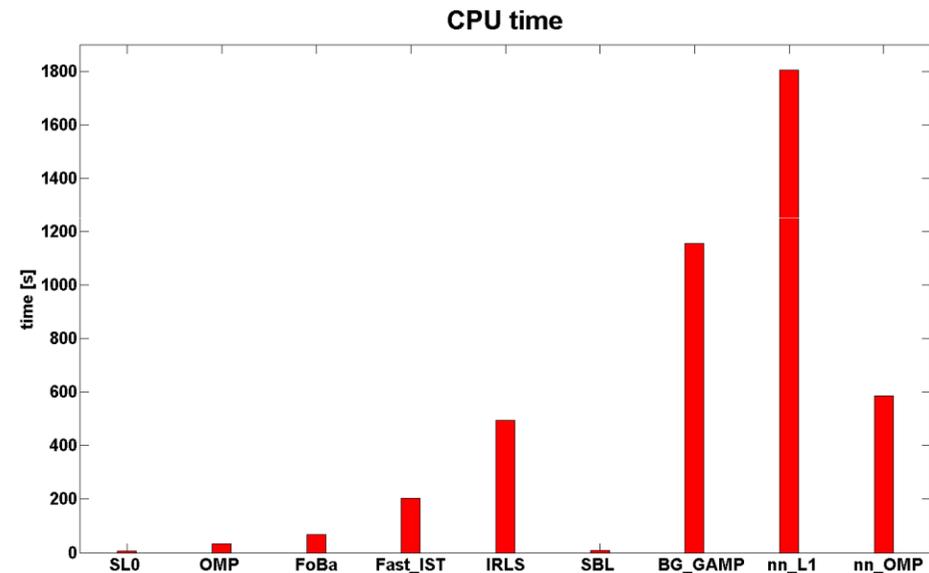
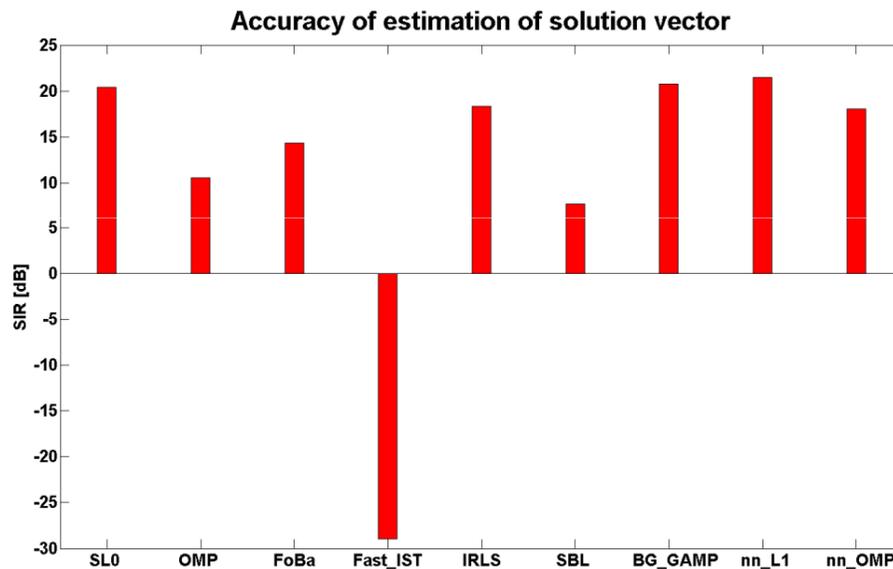
SNR=inf





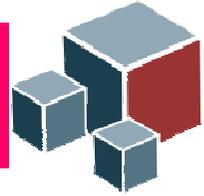
## Comparative performance analysis

SNR=10dB



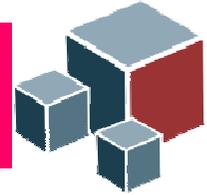
nn\_OMP – nonnegative OMP

nn\_L1 - nonnegative  $L_1$  –constrained least square algorithm.



## Comparative performance analysis - conclusions

- SL0, IRLS, BG\_GAMP, nonnegative OMP and nonnegative  $L_1$ -constrained least square algorithm yield the best accuracy.
- The SL0 algorithm has least computational complexity. Sometimes even two or three orders of magnitude less than other competitors.



## Estimation of the mixing matrix: "single source points"

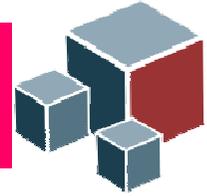
Single source points (SSP), a.k.a. pure pixels in hyperspectral image analysis, are points where only single source is active or dominantly present. At these points  $\|\mathbf{s}\|_0 \approx 1$ . This is approximately correct in medical imaging modalities with good spatial resolution.

For complex signals such points can be located using:

$$\left| \frac{R\{\mathbf{x}_t\}^T I\{\mathbf{x}_t\}}{\|R\{\mathbf{x}_t\}\| \|I\{\mathbf{x}_t\}\|} \right| \geq \cos(\Delta\theta)$$

where  $R\{\mathbf{x}_t\}$  and  $I\{\mathbf{x}_t\}$  denote real and imaginary part of  $\mathbf{x}_t$ , and  $\Delta\theta$  denotes angular displacement from a direction of 0 or  $\pi$  radians.

At SSPs mixture vectors  $\mathbf{x}_t$  and mixing vectors  $\mathbf{a}_m$  are collinear i.e.  $\mathbf{x}_t \approx \mathbf{a}_m s_{mt}$ . Hence, clustering of a set of SSPs yields estimate of the mixing matrix  $\mathbf{A}$ .



## Estimation of the mixing matrix: clustering

F. M. Naini, G. H. Mohimani, M. Babaie-Zadeh, C. Jutten, "Estimating the mixing matrix in sparse component analysis (SCA) based on partial k-dimensional subspace clustering," *Neurocomputing*, vol. 71, pp. 2330-2343, 2008.

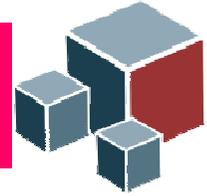
Assuming unit  $L_2$ -norm of  $\mathbf{a}_m$  and  $N=2$  we can parameterize column vectors in a plane by one angle

$$\mathbf{a}_m = [\cos(\varphi_m) \quad \sin(\varphi_m)]^T$$

Assuming that  $\mathbf{s}$  is 1-sparse in representation domain estimation of  $\mathbf{A}$  and  $M$  is obtained by means of data clustering algorithm.

We remove all data points close to the origin for which applies:  $\{|\mathbf{x}(t)|_2 \leq \varepsilon\}_{t=1}^T$  where  $\varepsilon$  represents some predefined threshold.

Normalize to unit  $L_2$ -norm remaining data points  $\mathbf{x}(t)$ , i.e.,  $\{\mathbf{x}(t) \rightarrow \mathbf{x}(t)/|\mathbf{x}(t)|_2\}_{t=1}^{\bar{T}}$



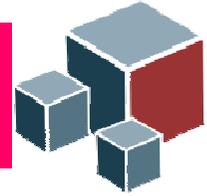
## Estimation of the mixing matrix: clustering

Calculate function  $f(\mathbf{a})$ :

$$f(\mathbf{a}) = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2(\mathbf{x}(t), \mathbf{a})}{2\sigma^2}\right)$$

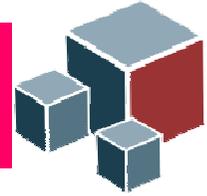
where  $d(\mathbf{x}(t), \mathbf{a}) = \sqrt{1 - (\mathbf{x}(t) \cdot \mathbf{a})^2}$  and  $(\mathbf{x}(t) \cdot \mathbf{a})$  denotes inner product. Parameter  $\sigma$  is called dispersion. If  $\sigma$  is set to sufficiently small value the value of the function  $f(\mathbf{a})$  will approximately equal the number of data points close to  $\mathbf{a}$ . Thus by varying mixing angle  $\varphi$  we effectively cluster data.

- Number of peaks of the function  $f(\mathbf{a})$  corresponds with the estimated number of sources  $M$ . Locations of the peaks correspond with the estimates of the mixing angles  $\{(\hat{\varphi}_m)\}_{m=1}^{\hat{M}}$ , i.e., mixing vectors  $\{\hat{\mathbf{a}}_m(\hat{\varphi}_m)\}_{m=1}^{\hat{M}}$ .



## Estimation of the mixing matrix: clustering

- hierarchical clustering by MATLAB function `clusterdata`. It is assumed that number of clusters (sources) is given (known). The method is deterministic and memory demanding.
- k-means clustering by MATLAB function `kmeans`. It is assumed that a number of clusters  $M$  (corresponds with number of sources) is given.  $k$ -means clustering is a first order method it is sensitive on initial choice of cluster centers (centroids).



## Mixing matrix estimation

N. Gillis and S.A. Vavasis, "Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization", 2012. <http://arxiv.org/abs/1208.1237>

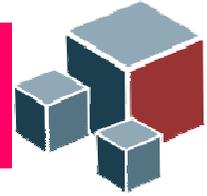
MATLAB Code: <https://sites.google.com/site/nicolasgillis/code>.

This method estimates the mixing matrix by generalizing several hyperspectral unmixing algorithms based on pure pixels (single source points) assumption.

The algorithm is recursive and fast (Fast\_NMF), i.e. it estimates one mixing vector at a time. There are no parameters required to be chosen *a priori* or to be tuned. The method works even when data matrix is not approximately separable, i.e. pure pixels do not exist.

The method identifies 'M' columns of data matrix whose convex hull has encompasses the data.

Mixing matrix must be full rank, i.e. number of components has to be less than or equal to the number of mixtures.



## Mixing matrix estimation

G. H. Ritter, G. Urcid, "A lattice matrix method for hyperspectral image unmixing," *Information Sciences*, vol. 181, pp. 1787-1803, 2011.

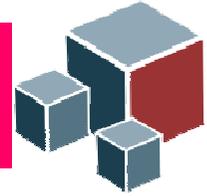
MATLAB code:

[http://www.ehu.es/ccwintco/index.php/Endmember\\_Induction\\_Algorithms\\_\(EIAs\)\\_for\\_MATLAB\\_and\\_SCILAB](http://www.ehu.es/ccwintco/index.php/Endmember_Induction_Algorithms_(EIAs)_for_MATLAB_and_SCILAB)

Autonomous endmember determination algorithm using lattice associate memory (LAM) theory.

Unlike many methods in hyperspectral image analysis it does not assume/require existence of pure pixels. Instead, it searches for the least contaminated pixels.

Also, it does not require number of endmembers (sources) to be known in advance.



## Comparative performance analysis

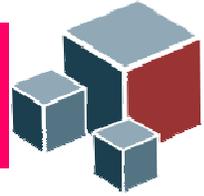
Fast\_NMF and LAM algorithms were compared on mixing matrix estimation using model:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}$$

where  $N=250$ ,  $M=250$ ,  $T=1000$ ,  $\text{SNR [dB]} \in \{10, 20, 30, \text{inf}\}$ ,  $K \in \{10, 30, 50\}$ .

Each source was generated with probability of being zero equal to 0.8. The nonzero state was generated by uniform distribution on interval (0,1].

Entries of the  $N \times M$  mixing matrix were drawn from uniform distribution on interval (0,1]. The mixing matrix has been column normalized to unit  $L_2$  norm.

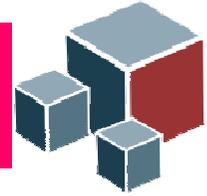


## Comparative performance analysis

Performance measure:

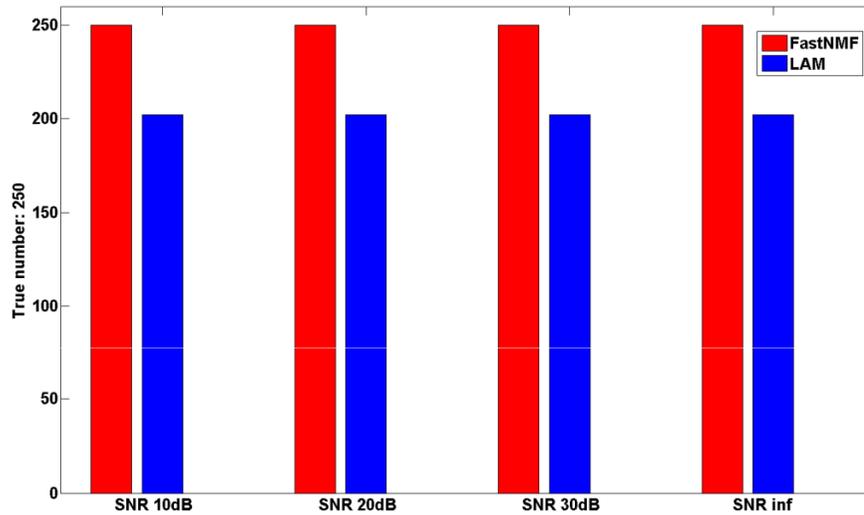
$$\varepsilon(\mathbf{A}) = 10 \log_{10} \left( \frac{\| \text{diag}(\mathbf{R}_{\mathbf{AA}}) \|_2^2}{\| \text{diag}(\mathbf{R}_{\mathbf{AA}} - \mathbf{R}_{\mathbf{A}\hat{\mathbf{A}}}) \|_2^2} \right)$$

has been averaged after 100 runs, where  $\mathbf{R}_{\mathbf{AA}}$  and  $\mathbf{R}_{\mathbf{A}\hat{\mathbf{A}}}$  are respectively autocovariance and cross-covariance matrices calculated after permutation indeterminacy has been resolved.

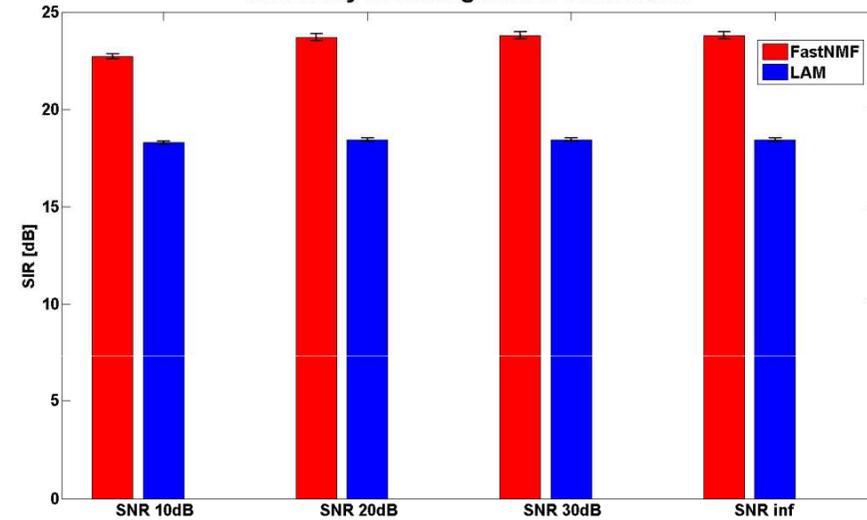


## Comparative performance analysis

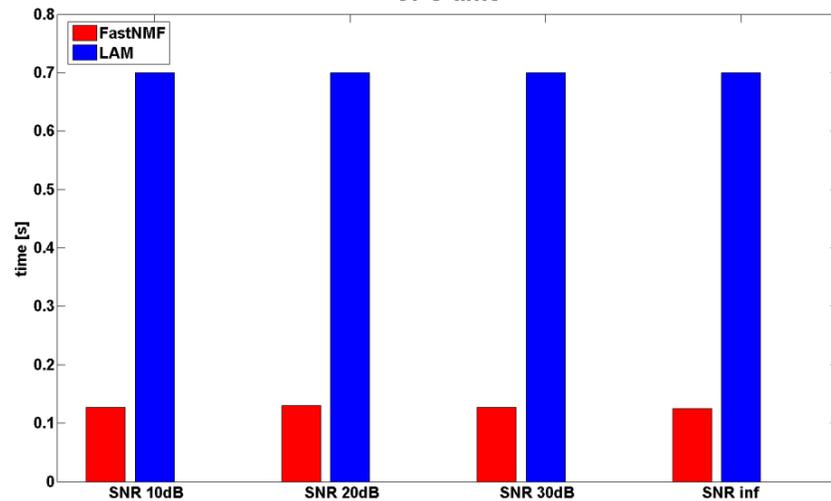
Accuracy of estimation of number of sources



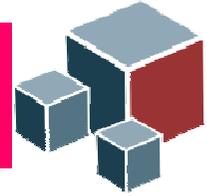
Accuracy of mixing matrix estimation



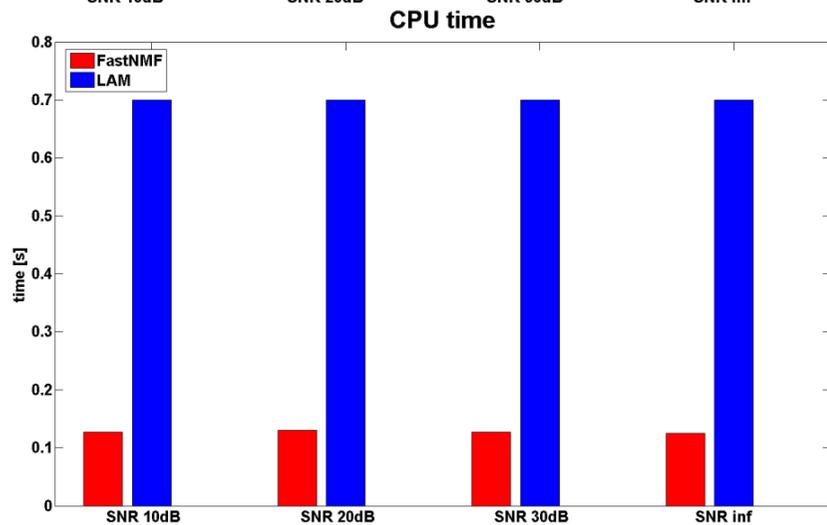
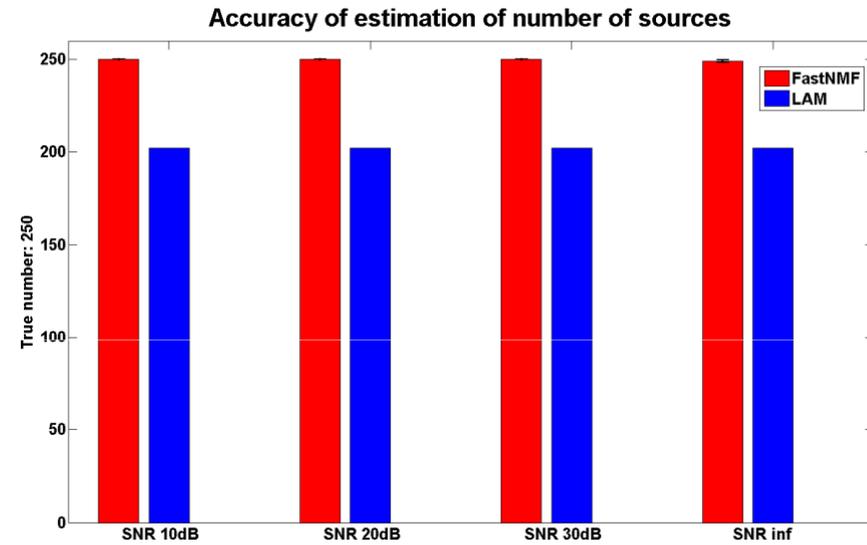
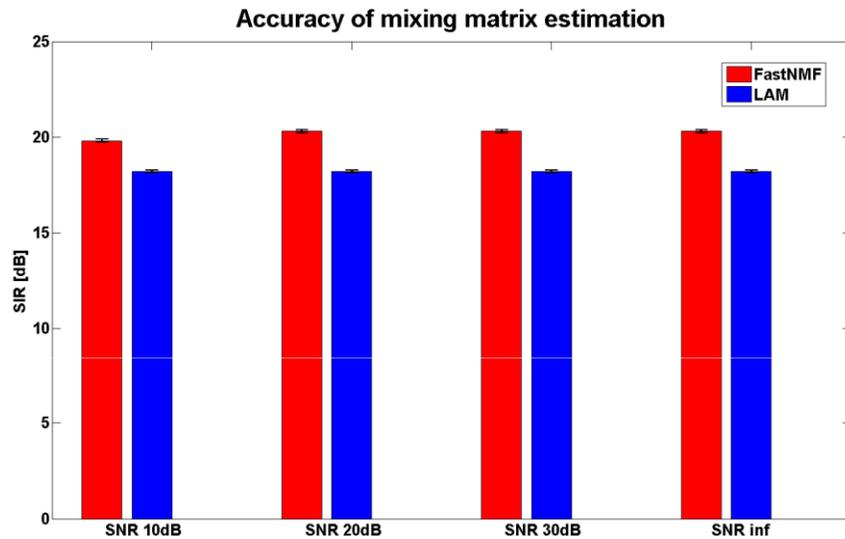
CPU time



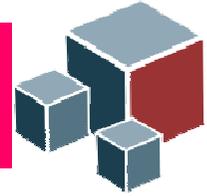
K=10



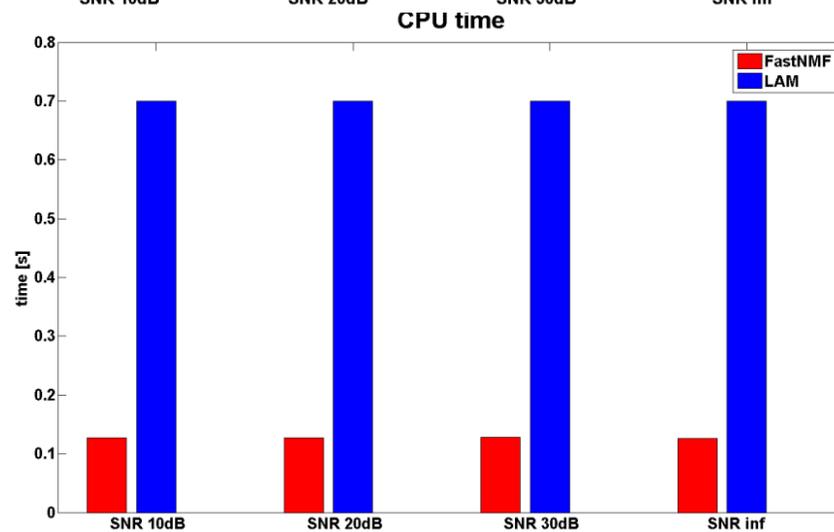
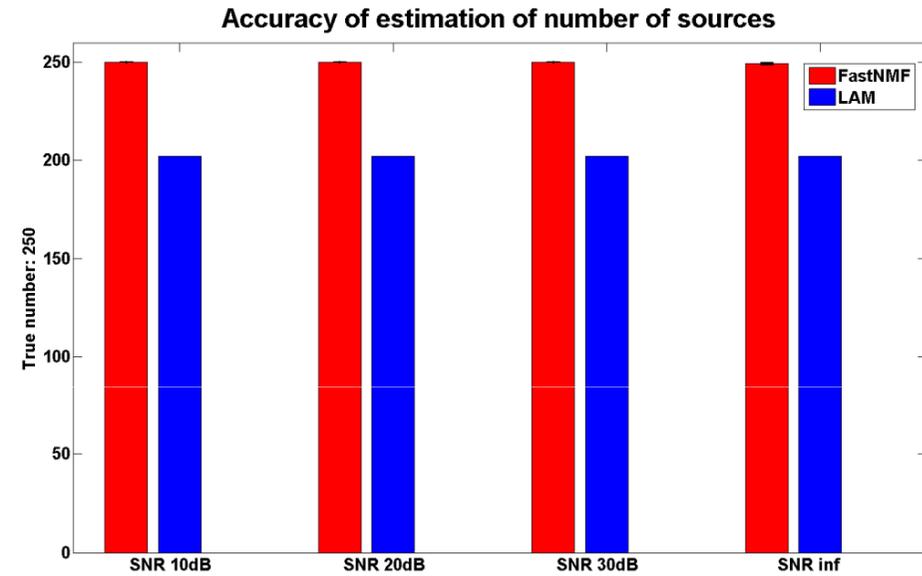
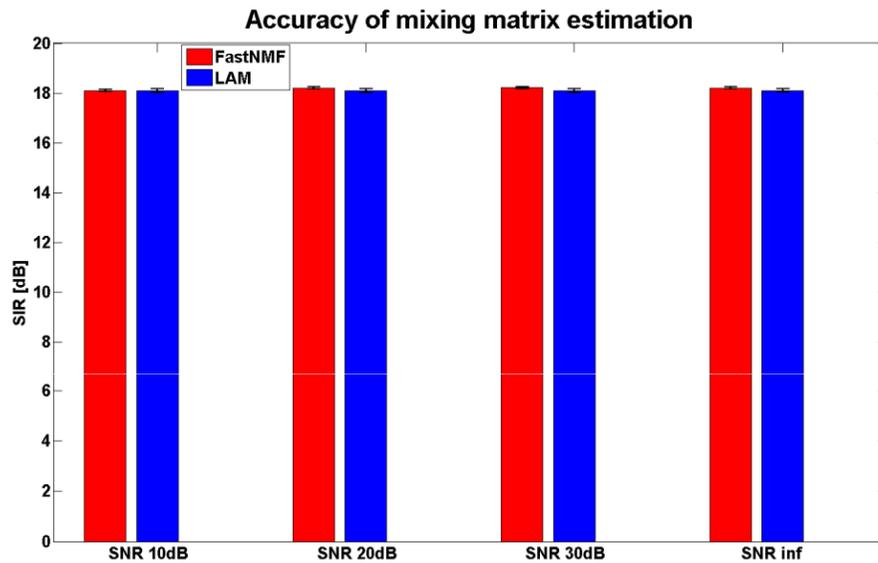
## Comparative performance analysis



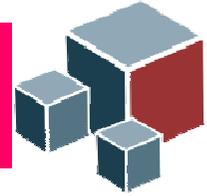
K=30



## Comparative performance analysis



K=50



## Comparative performance analysis - conclusions

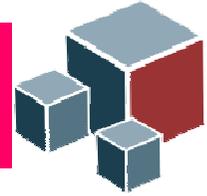
Fast\_NMF algorithm is good choice for mixing matrix estimation when:

- a) number of sources is not greater than number of mixtures
- b) pure pixel assumption is not severely violated.

The LAM algorithm is good choice for mixing matrix estimation when:

- a) number of sources is greater than number of mixtures.
- b) pure pixel assumption is violated significantly.

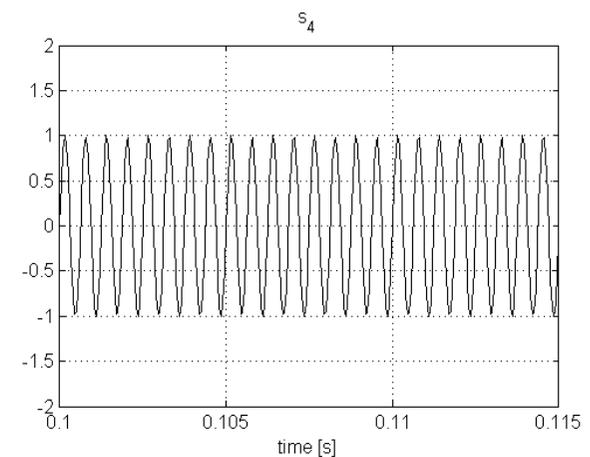
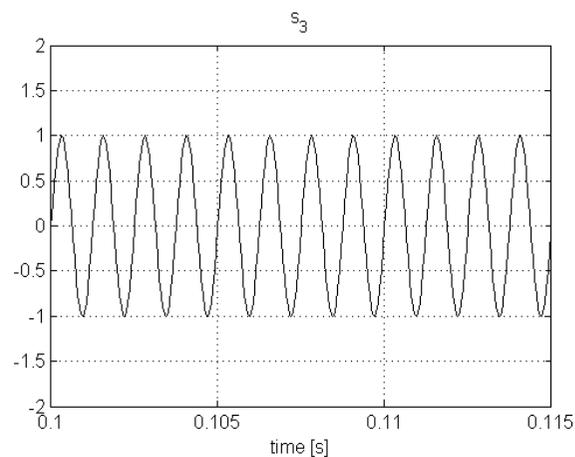
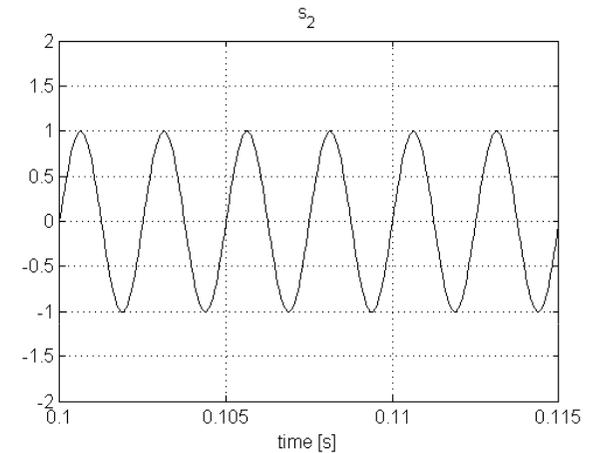
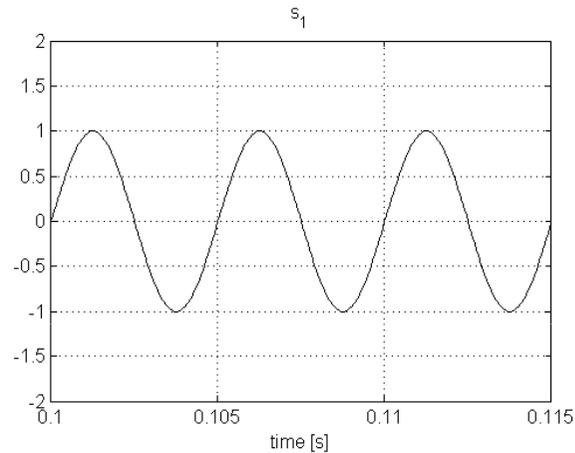
Majority of other mixing matrix estimation algorithms fall in between scenarios covered by Fast\_NMF and LAM algorithms.

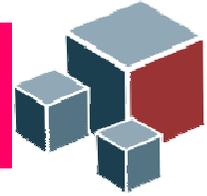


# Blind separation of four sine signals from two mixtures

Four sinusoidal signals with frequencies 200Hz, 400Hz, 800Hz and 1600Hz.

## TIME DOMAIN

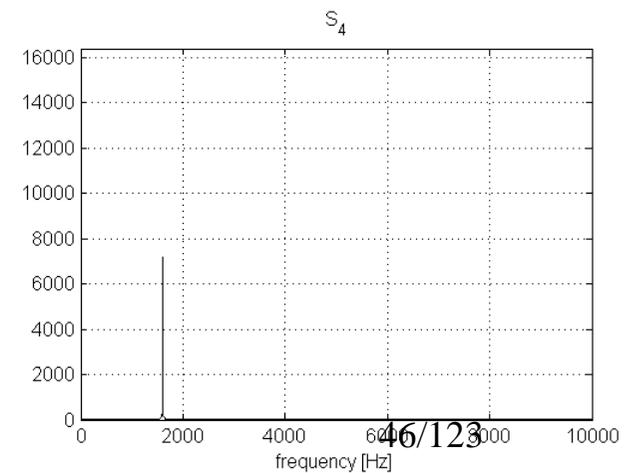
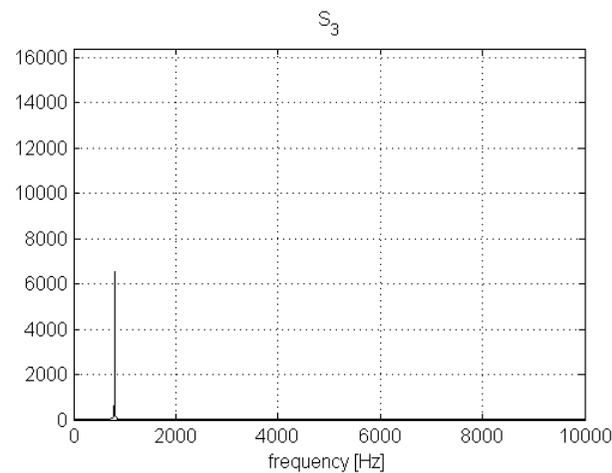
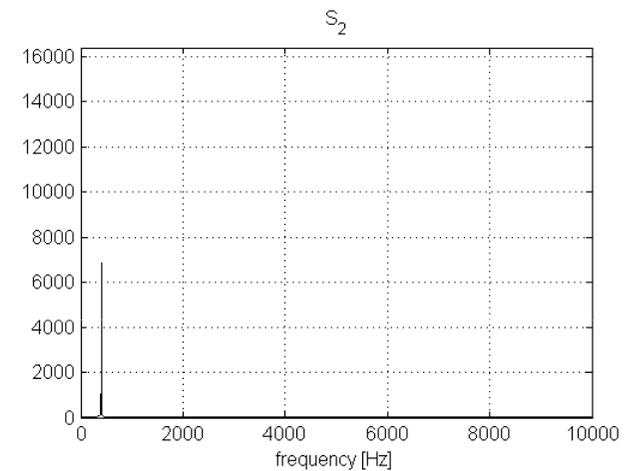
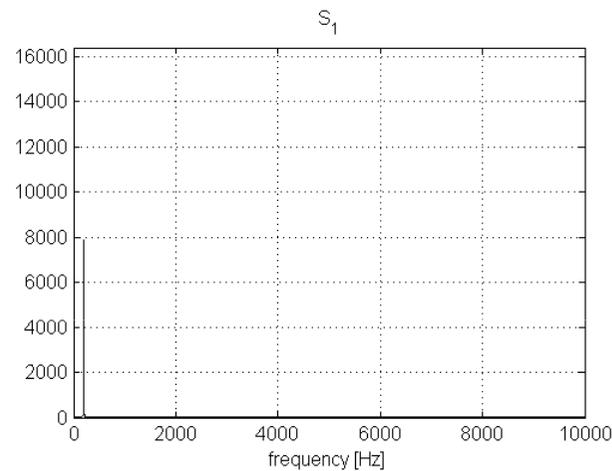


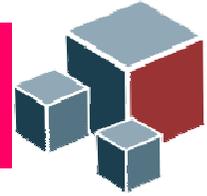


# Blind separation of four sine signals from two mixtures

Four sinusoidal signals with frequencies 200Hz, 400Hz, 800Hz and 1600Hz.

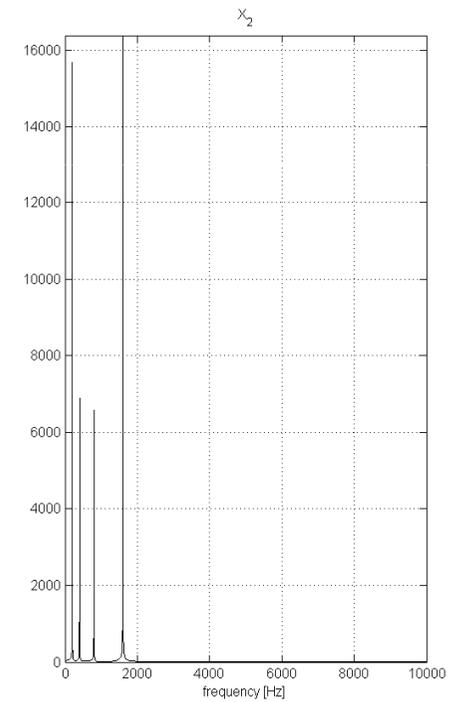
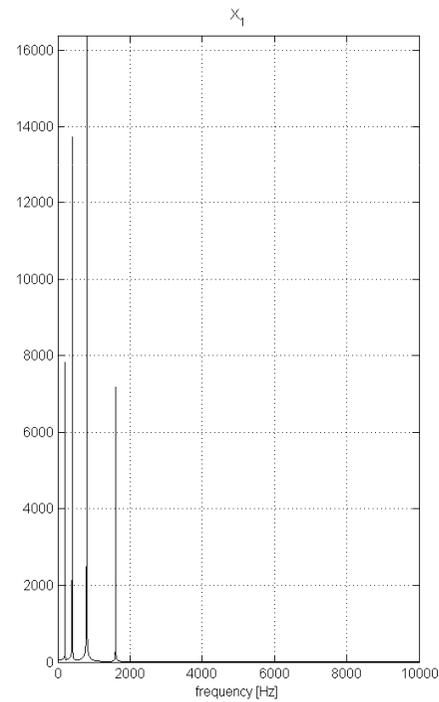
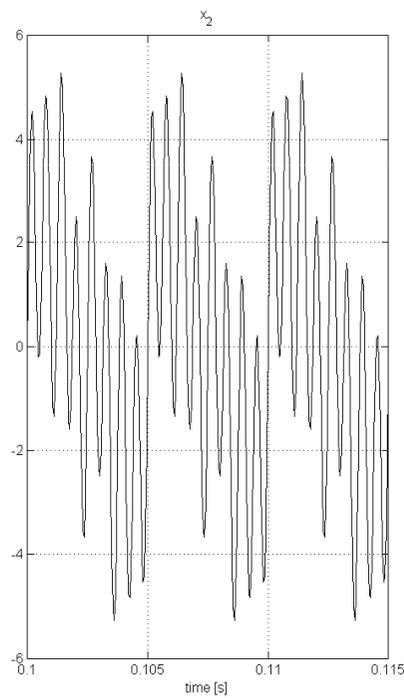
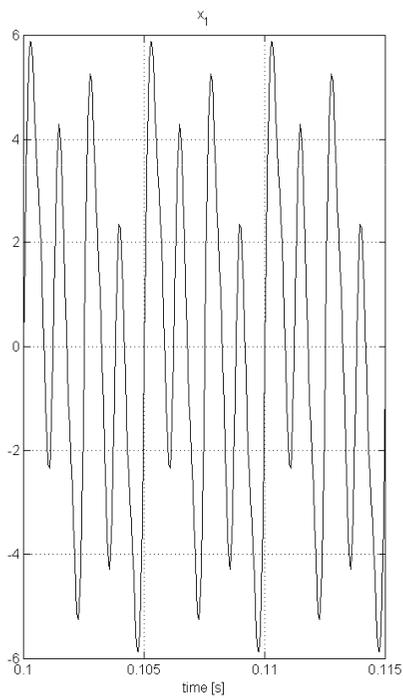
## FREQUENCY DOMAIN





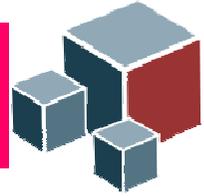
# Blind separation of four sine signals from two mixtures

## Two mixed signals



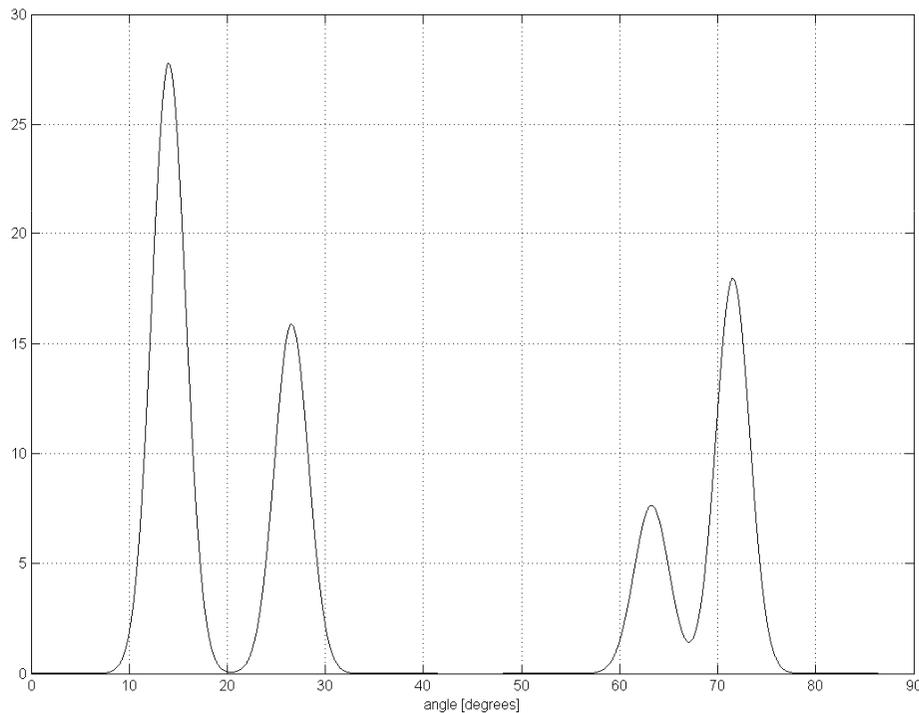
TIME DOMAIN

FREQUENCY DOMAIN<sub>47/123</sub>



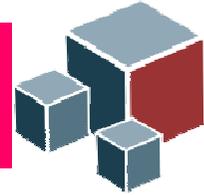
# Blind separation of four sine signals from two mixtures

## Clustering function



$$\mathbf{A}=[63.44^{\circ} \ 26.57^{\circ} \ 14.04^{\circ} \ 71.57^{\circ}]$$

$$\mathbf{AH}=[14.03^{\circ} \ 26.55^{\circ} \ 63.26^{\circ} \ 71.55^{\circ}]$$



## Blind separation of four sine signals from two mixtures

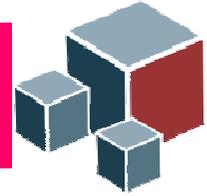
Linear programming based estimation of the sources based on estimated mixing matrix  $\mathbf{A}$

$$\begin{bmatrix} \mathbf{x}_r(\omega) \\ \mathbf{x}_i(\omega) \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r(\omega) \\ \mathbf{s}_i(\omega) \end{bmatrix}$$

or:

$$\bar{\mathbf{x}}(\omega) = \bar{\mathbf{A}}\bar{\mathbf{s}}(\omega)$$

$\mathbf{s}_r(\omega)$  and  $\mathbf{s}_i(\omega)$  are not necessarily nonnegative. Thus, constraint  $\bar{\mathbf{s}}(\omega) \geq \mathbf{0}$  required by linear program is not satisfied. In such a case it is customary to introduce dummy variables:  $\mathbf{u}, \mathbf{v} \geq \mathbf{0}$ , such that  $\bar{\mathbf{s}}(\omega) = \mathbf{u} - \mathbf{v}$ .



## Blind separation of four sine signals from two mixtures

Introducing:

$$\mathbf{z}(\omega) = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \quad \tilde{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{A}} & -\bar{\mathbf{A}} \end{bmatrix}$$

yields:

$$\hat{\mathbf{z}}(\omega) = \arg \min_{\mathbf{z}(\omega)} \sum_{m=1}^{4M} z_m(\omega) \quad \text{s.t.} \quad \tilde{\mathbf{A}}\mathbf{z}(\omega) = \bar{\mathbf{x}}$$

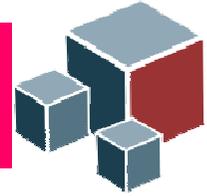
$$\mathbf{z}(\omega) \geq \mathbf{0}$$

We obtain  $\bar{\mathbf{s}}(\omega)$  from  $\hat{\mathbf{z}}(\omega)$  as:

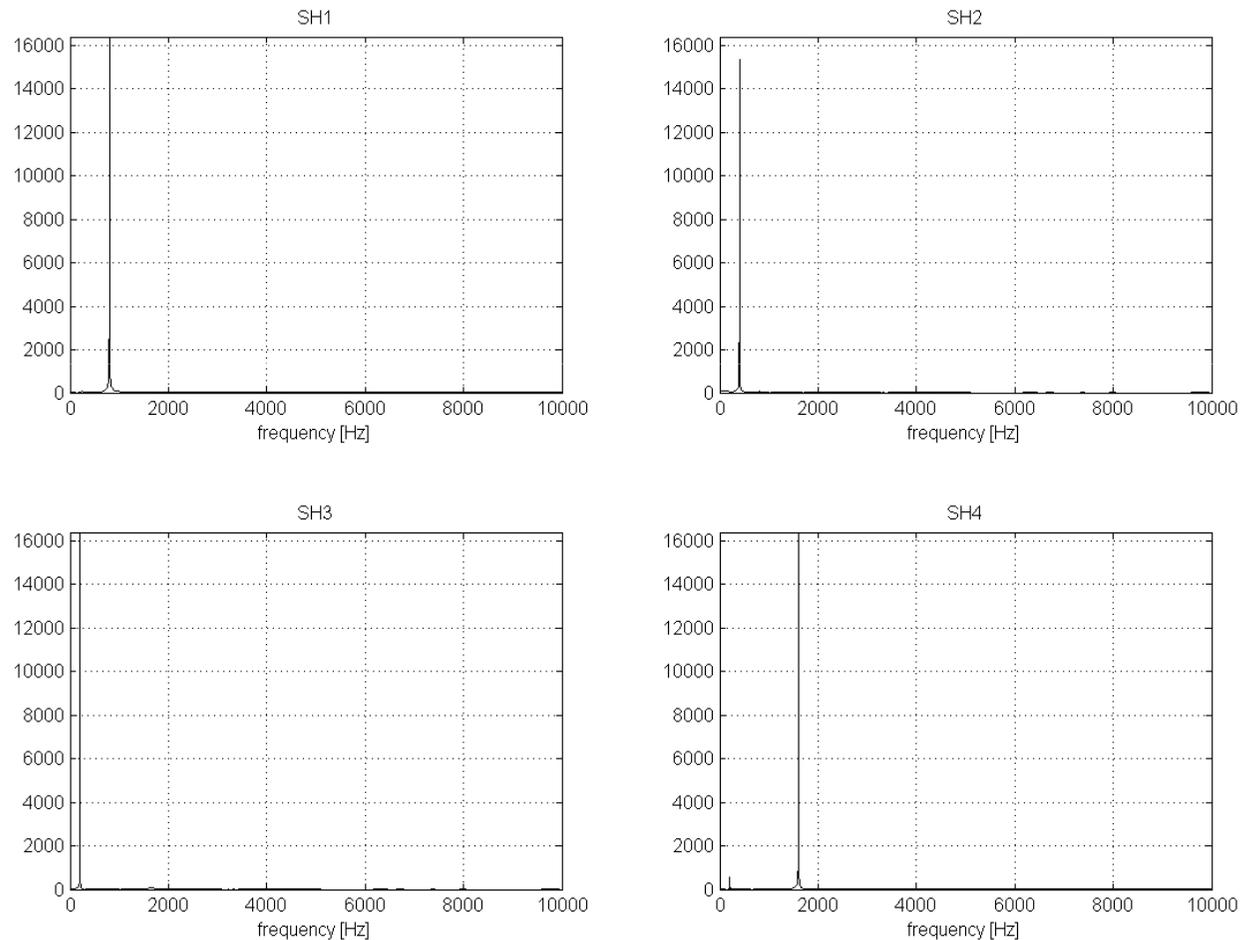
$$\bar{\mathbf{s}}(\omega) = \hat{\mathbf{u}} - \hat{\mathbf{v}}$$

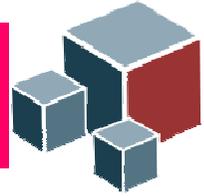
and  $s(t)$  as:

$$s_m(t) = IDFT[\bar{s}_m(\omega)]$$

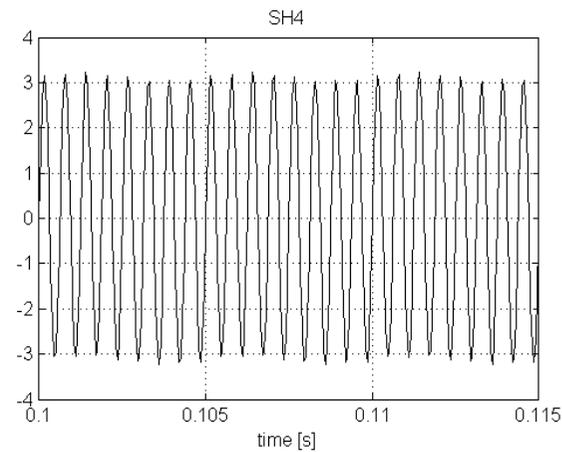
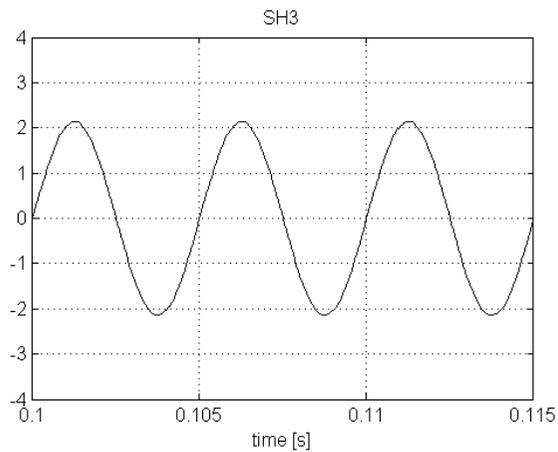
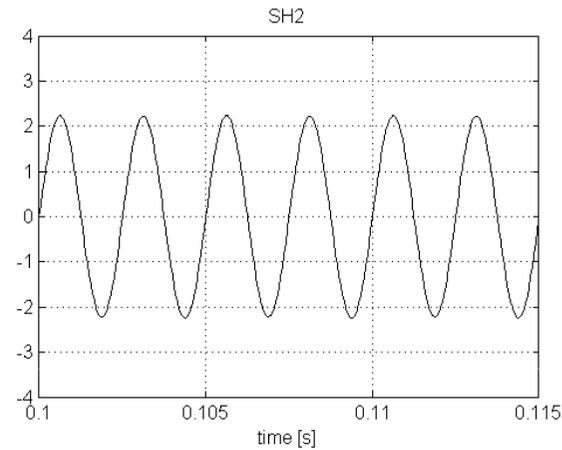
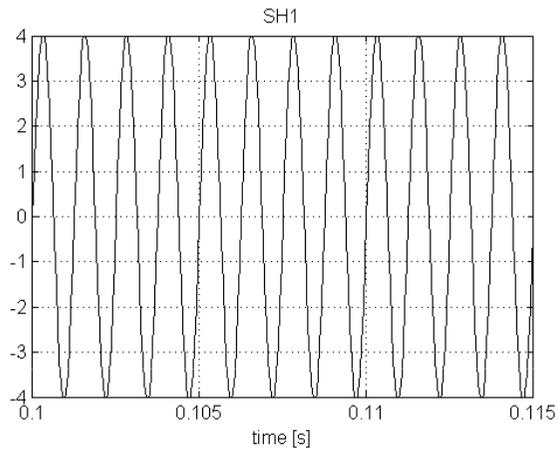


# Blind separation of four sine signals from two mixtures

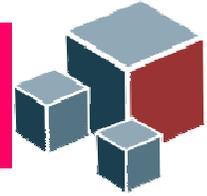




# Blind separation of four sine signals from two mixtures

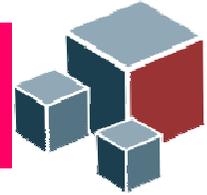


Estimated sources in TIME DOMAIN



# Blind extraction of analytes (pure components) from mixtures of chemical compounds in NMR spectroscopy and mass spectrometry

- I. Kopriva, I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1920.
- I. Kopriva, I. Jerić (2009). Multi-component Analysis: Blind Extraction of Pure Components Mass Spectra using Sparse Component Analysis, *Journal of Mass Spectrometry*, vol. **44**, issue 9, pp. 1378-1388

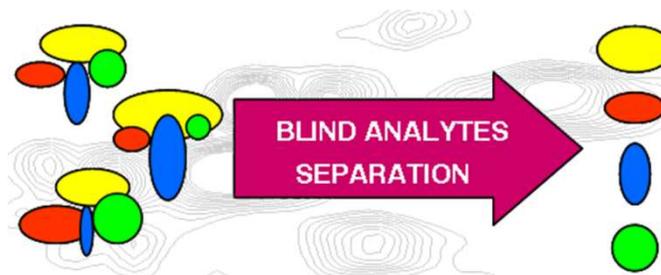


## Linear mixing model

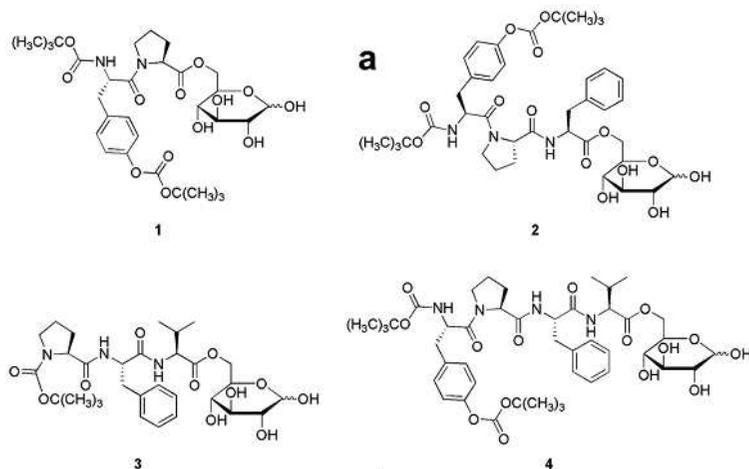
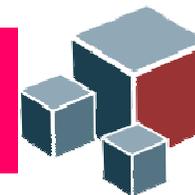
$$\mathbf{X} = \mathbf{A}\mathbf{S} \quad \mathbf{X} \in \mathbb{R}_{0+}^{N \times T}, \mathbf{S} \in \mathbb{R}_{0+}^{M \times T}, \mathbf{A} \in \mathbb{R}_{0+}^{N \times M} \quad (1)$$

In chemometrics (NMR spectroscopy or mass spectrometry) rows of  $\mathbf{X}$  represent spectra of mixture samples, columns of  $\mathbf{A}$  represent concentration profiles of analytes (a.k.a. pure components) present in mixture spectra  $\mathbf{X}$ , while rows of  $\mathbf{S}$  represent spectra of analytes present in mixture spectra  $\mathbf{X}$ .

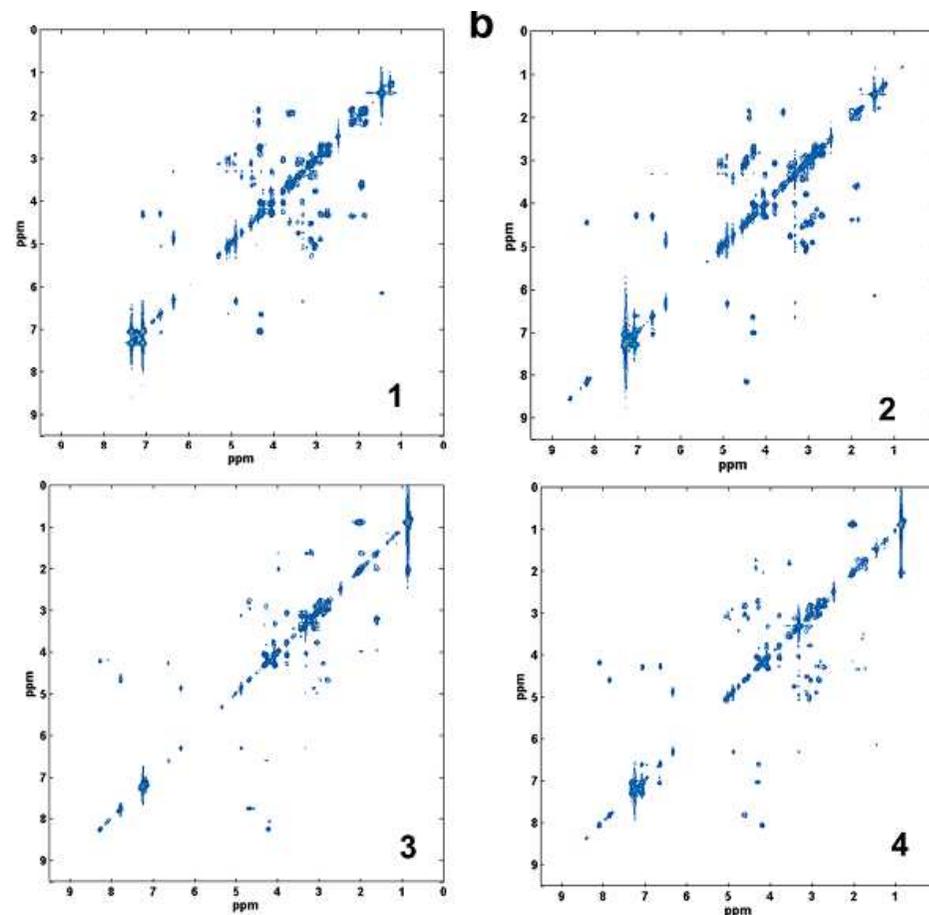
The (u)BSS problem relates to extraction of analytes (and their concentrations) using mixture spectra  $\mathbf{X}$  only:



Pure components can represent compounds indicative for disease. Thus, they can be useful for biomarker analysis. They can be isolated from spectra (NMR, mass) of biological samples (urine, blood, tissues).

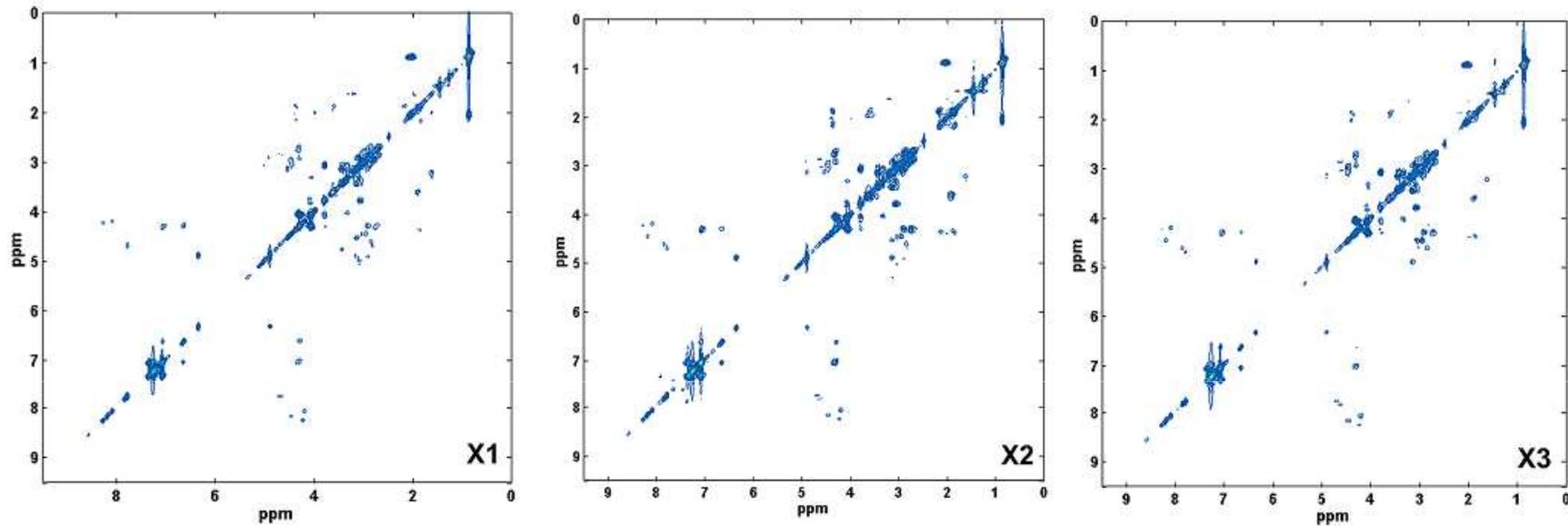
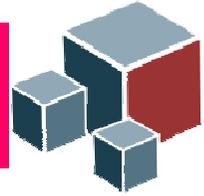


Structure of four analytes (glycopeptides)

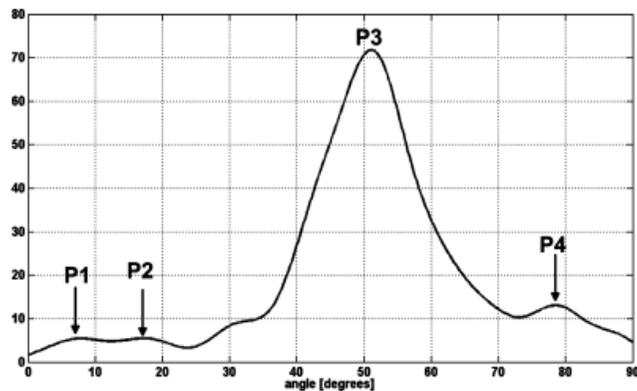
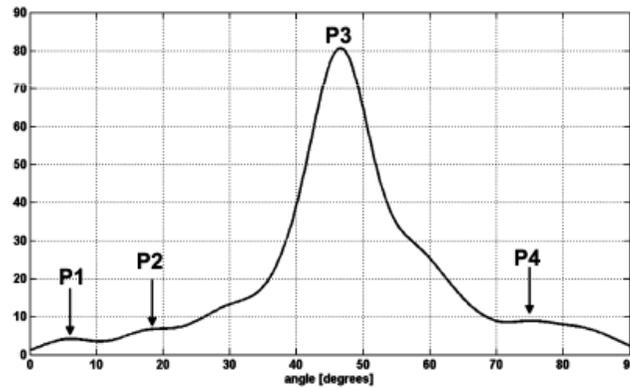
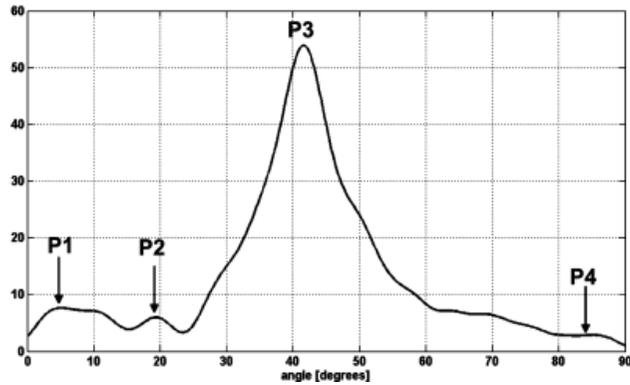
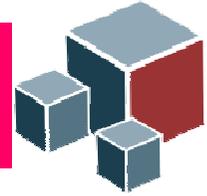


COSY NMR spectra of four analytes

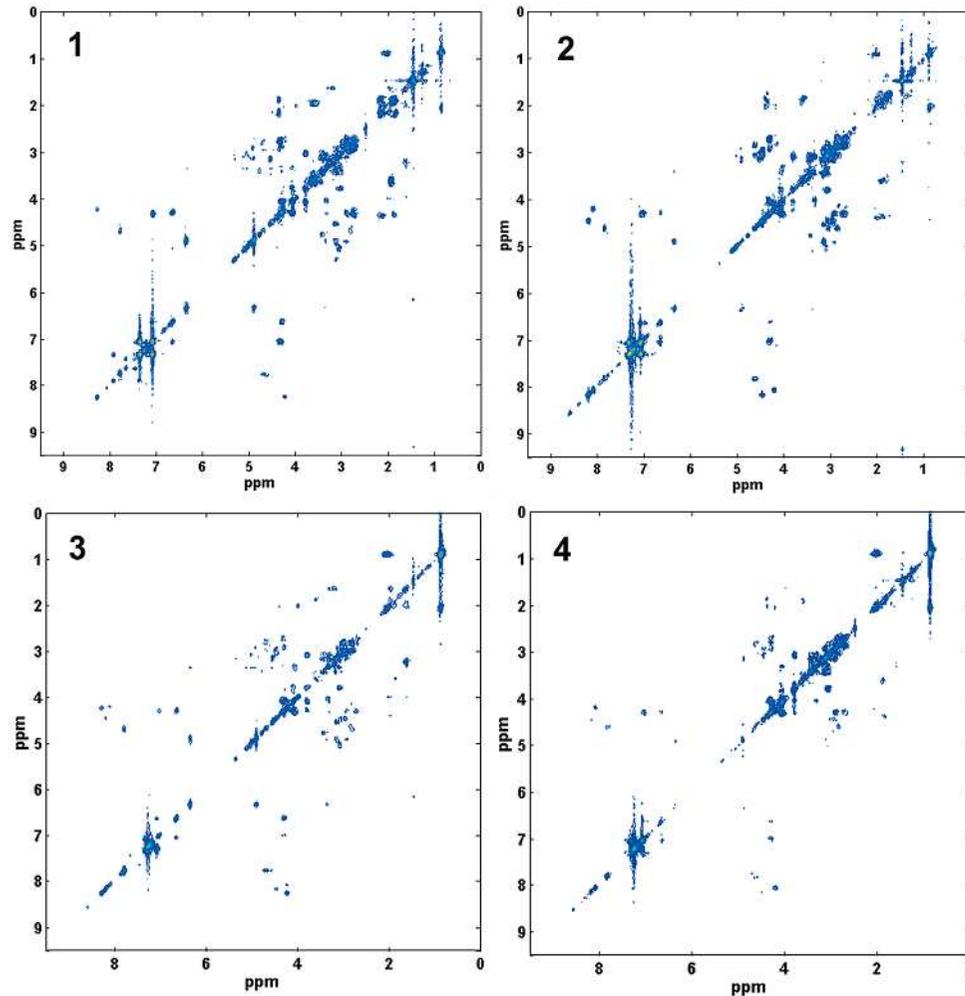
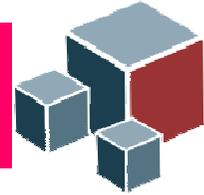
I. Kopriva, I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1926



COSY NMR spectra of three mixtures



Clustering functions calculated on a set of 203 SAPs in 2D wavelet domain in 2D subspaces:  $X_1X_2$ ,  $X_1X_3$  and  $X_2X_3$ .

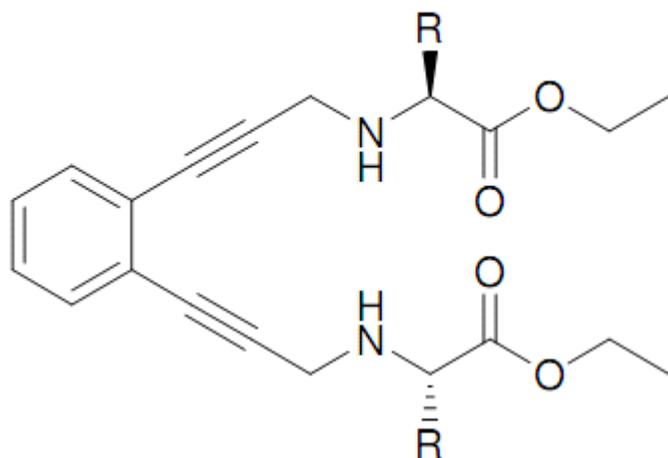
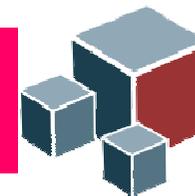


**Table 1. Normalized Correlation Coefficients for (a) Pure Analytes 1–4; (b) Analytes 1–4 Estimated on 203 SAPs Detected in Symmlet 8 Wavelet Domain; (c) Analytes 1–4 Estimated on 23 SAPs Detected in Fourier Domain; (d) Analytes 1–4 Estimated by Means of JADE ICA Algorithm from Four Mixtures<sup>a</sup>**

entry		An <sub>1</sub>	An <sub>2</sub>	An <sub>3</sub>	An <sub>4</sub>
a	An <sub>1</sub>	<b>1</b>	0.5509	0.1394	0.3730
	An <sub>2</sub>	0.5509	<b>1</b>	0.3051	0.5120
	An <sub>3</sub>	0.1394	0.3051	<b>1</b>	0.7965
	An <sub>4</sub>	0.3730	0.5120	0.7965	<b>1</b>
b	$\hat{A}n_1$	<b>0.8931</b>	0.4753	0.2638	0.4132
	$\hat{A}n_2$	0.5634	<b>0.8579</b>	0.2795	0.5366
	$\hat{A}n_3$	0.1945	0.5048	<b>0.8990</b>	0.7953
	$\hat{A}n_4$	0.4386	0.6124	0.8060	<b>0.8381</b>
c	$\hat{A}n_1$	<b>0.8924</b>	0.6009	0.2754	0.4602
	$\hat{A}n_2$	0.5482	<b>0.8469</b>	0.3107	0.5695
	$\hat{A}n_3$	0.0931	0.4101	<b>0.8432</b>	0.7249
	$\hat{A}n_4$	0.3108	0.3411	<b>0.8236</b>	0.7331
d	$\hat{A}n_1$	0.7189	0.7090	0.6805	<b>0.7939</b>
	$\hat{A}n_2$	0.6873	0.7571	0.6524	<b>0.7790</b>
	$\hat{A}n_3$	0.6606	0.7325	0.7142	<b>0.8177</b>
	$\hat{A}n_4$	0.6322	0.7232	0.7474	<b>0.8342</b>

<sup>a</sup> A significant degree of correlation between spectra of true analytes caused failure of the ICA-based extraction of analytes, part d. An<sub>1</sub>–An<sub>4</sub> pure analytes 1–4;  $\hat{A}n_1$ – $\hat{A}n_4$  estimated analytes 1–4.

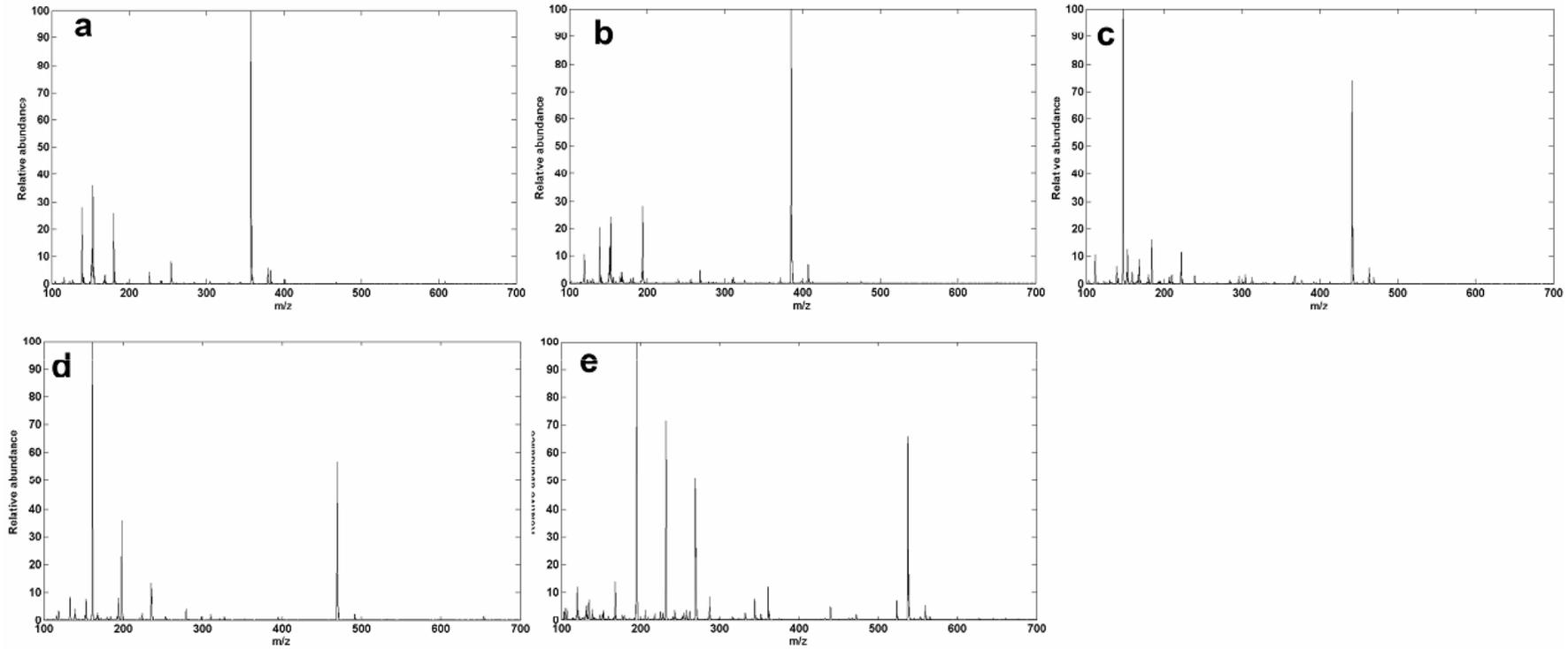
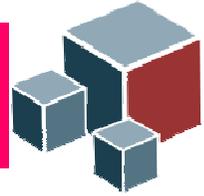
Estimated COSY NMR spectra of analytes in 2D Fourier domain



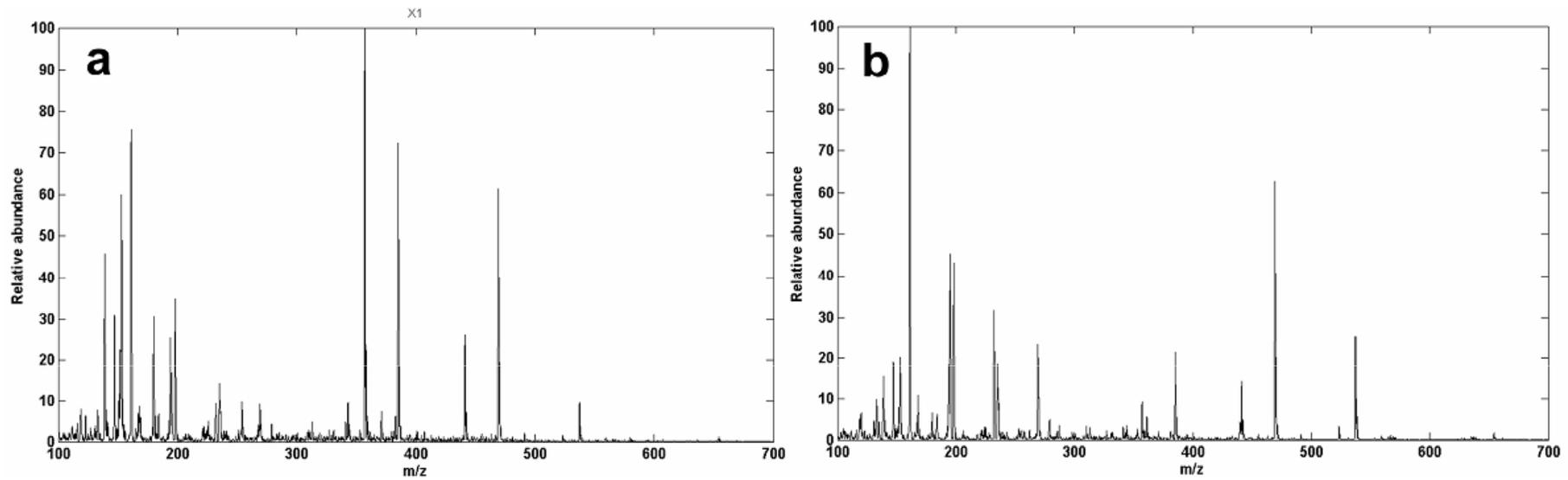
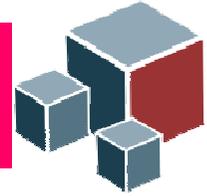
- 5 R=H
- 6 R=CH<sub>3</sub>
- 7 R=CH(CH<sub>3</sub>)<sub>2</sub>
- 8 R=CH<sub>2</sub>CH(CH<sub>3</sub>)<sub>3</sub>
- 9 R=CH<sub>2</sub>C<sub>6</sub>H<sub>5</sub>

Chemical structure of five pure components.

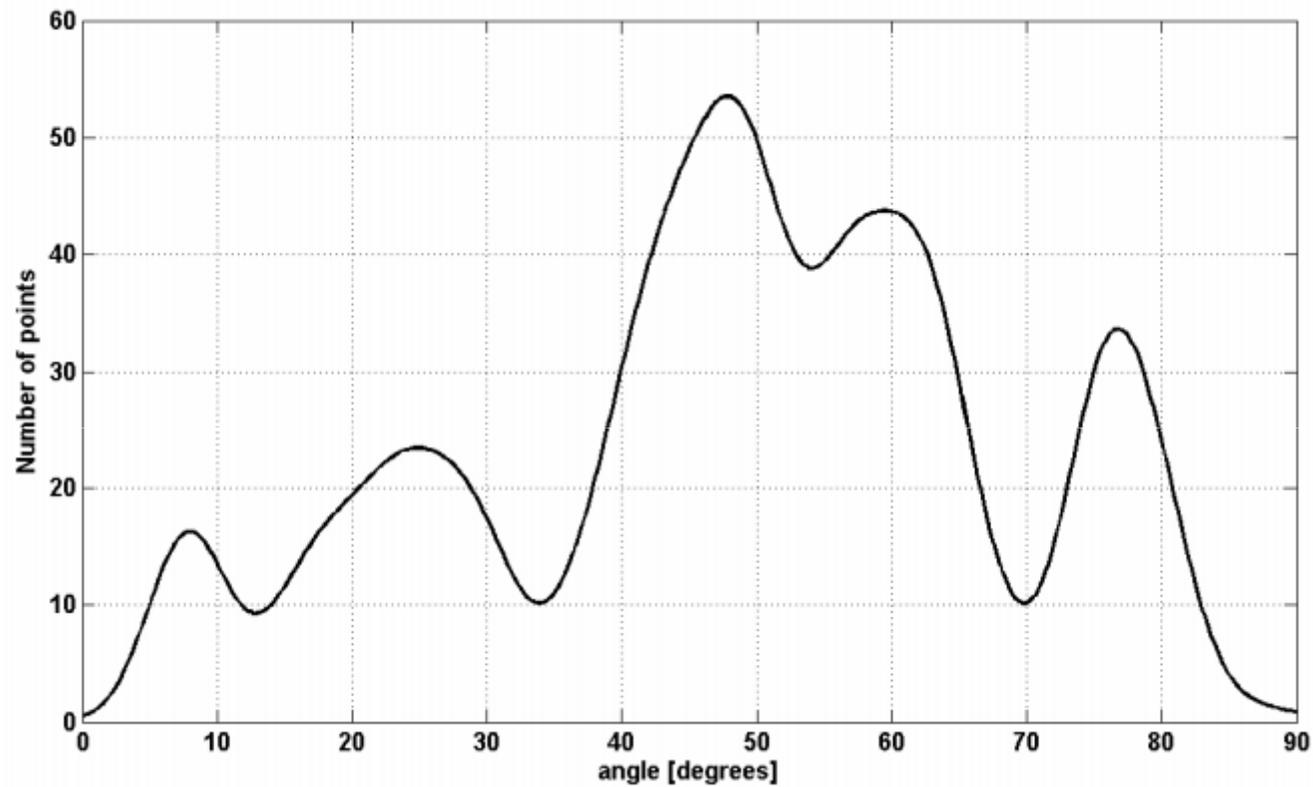
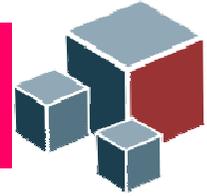
I. Kopriva, I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1920.



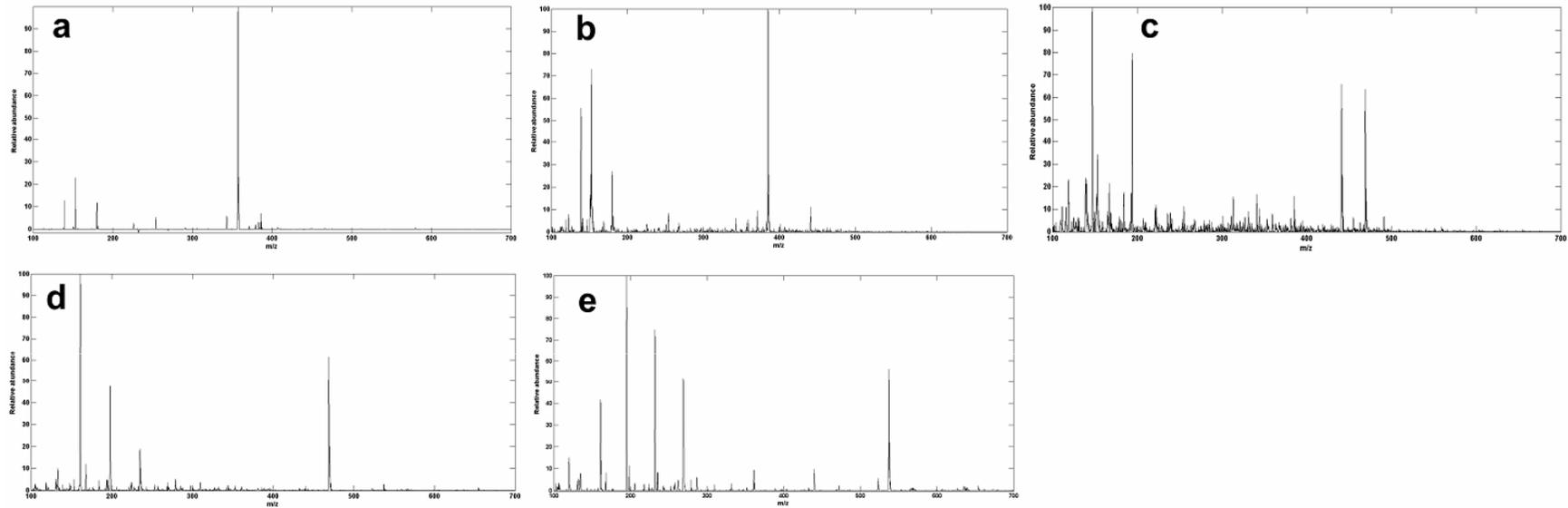
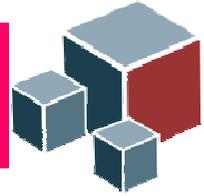
Mass spectra of five pure components.



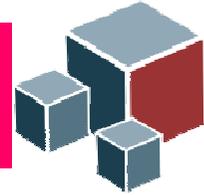
Mass spectra of two mixtures



Data clustering function in the mixing angle domain. Five peaks indicate presence of five components in the mixtures spectra.



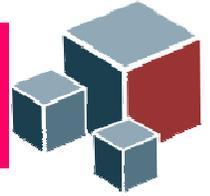
Estimated mass spectra of five pure components.



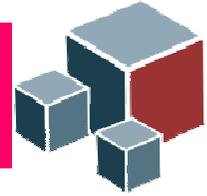
**Table S-1.** Normalized correlation coefficients for (a) pure analytes **5-9**; (b) analytes **5-9** estimated on 290 SAPs detected by using analytical representation (3) and *clusterdata* algorithm.\*

entry		$An_5$	$An_6$	$An_7$	$An_8$	$An_9$
a	$An_5$	<b>1</b>	0.1268	0.0456	0.0266	0.0075
	$An_6$	0.1268	<b>1</b>	0.0321	0.0332	0.0379
	$An_7$	0.0456	0.0321	<b>1</b>	0.0134	0.0030
	$An_8$	0.0265	0.0332	0.0134	<b>1</b>	0.0029
	$An_9$	0.0075	0.0379	0.0030	0.0029	<b>1</b>
b	$\hat{An}_5$	<b>0.9038</b>	0.0305	0.0044	0.0002	0.0120
	$\hat{An}_6$	0.3162	<b>0.8294</b>	0.1198	0.0325	0.0043
	$\hat{An}_7$	0.0959	0.2334	<b>0.7275</b>	0.2009	0.0038
	$\hat{An}_8$	0.0043	0.0038	0.0124	<b>0.9736</b>	0.0293
	$\hat{An}_9$	0.0121	0.0161	0.0073	0.2097	<b>0.9437</b>

\*  $An_5$ - $An_9$  pure analytes **5-9**;  $\hat{An}_5$ -  $\hat{An}_9$  estimated analytes **5-9**.



# Nonnegative matrix factorization (NMF)



## Nonnegative matrix factorization

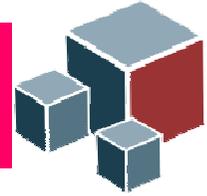
Many BSS problems arising in imaging, chemo- and/or bioinformatics are described by superposition of non-negative latent variables (sources):

$$\mathbf{X} = \mathbf{AS} \quad \mathbf{X} \in \mathbb{R}_{0+}^{N \times T}, \quad \mathbf{A} \in \mathbb{R}_{0+}^{N \times M} \quad \text{and} \quad \mathbf{S} \in \mathbb{R}_{0+}^{M \times T}$$

where  $N$  represents number of sensors,  $M$  represents number of sources and  $T$  represents number of samples.

Thus, solution of related decomposition problem can be obtained by imposing non-negativity constraints on  $\mathbf{A}$  and  $\mathbf{S}$ , to narrow down number of possible decomposition of  $\mathbf{X}$ . This leads to NMF algorithms.

Due to non-negativity constraints some other constraints (statistical independence) can be relaxed/replaced in applications where they are not fulfilled.



## Nonnegative matrix factorization

Modern approaches to NMF problems have been initiated by Lee-Seung' Nature paper, ref. [a], where it is proposed to estimate  $\mathbf{A}$  and  $\mathbf{S}$  through alternative minimization procedure of the two possibly different cost functions:

***Set Randomly initialize:  $\mathbf{A}^{(0)}, \mathbf{S}^{(0)}$ ,***

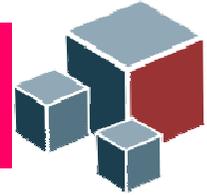
***For  $k=1,2,\dots$ , until convergence do***

$$\text{Step 1: } \mathbf{S}^{(k+1)} = \arg \min_{s_{mi} \geq 0} D_s \left( \mathbf{X} \parallel \mathbf{A}^{(k)} \mathbf{S} \right)_{\mathbf{S}^{(k)}}$$

$$\text{Step 2: } \mathbf{A}^{(k+1)} = \arg \min_{a_{nm} \geq 0} D_A \left( \mathbf{X} \parallel \mathbf{A} \mathbf{S}^{(k+1)} \right)_{\mathbf{A}^{(k)}}$$

If both cost functions represent squared Euclidean distance (Froebenius norm) we obtain alternating least square (ALS) approach to NMF.

a) D D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature* **401** (6755), 788-791 (1999).



## Nonnegative matrix factorization

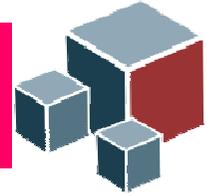
ALS-based NMF:

$$\left( \mathbf{A}^*, \mathbf{S}^* \right) = \arg \min_{\mathbf{A}, \mathbf{S}} D(\mathbf{X} \| \mathbf{AS}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 \quad s.t. \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}$$

- Minimization of the square of Euclidean norm of approximation error  $\mathbf{E} = \mathbf{X} - \mathbf{AS}$  is, from the maximum likelihood viewpoint, justified only if error distribution is Gaussian:

$$p(\mathbf{X} | \mathbf{A}, \mathbf{S}) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\|\mathbf{X} - \mathbf{AS}\|_2^2}{2\sigma^2}\right)$$

- In many instances non-negativity constraints imposed on  $\mathbf{A}$  and  $\mathbf{S}$  do not suffice to obtain solution that is unique up to standard BSS indeterminacies: permutation and scaling.



## Nonnegative matrix factorization

In relation to original Lee-Seung NMF algorithm additional constraints are necessary to obtain factorization unique up to permutation and scaling. Generalization that involves constraints is given in [a]:

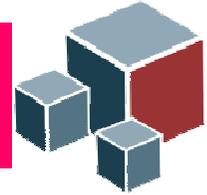
$$D(\mathbf{X} \parallel \mathbf{AS}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \alpha_S J_S(\mathbf{S}) + \alpha_A J_A(\mathbf{A})$$

where  $J_S(\mathbf{S}) = \sum_{m,t} s_{mt}$  and  $J_A(\mathbf{A}) = \sum_{n,m} a_{nm}$  are sparseness constraints that correspond with  $L_1$ -norm of  $\mathbf{S}$  and  $\mathbf{A}$  respectively.  $\alpha_S$  and  $\alpha_A$  are regularization constants. Gradient components in matrix form are:

$$\frac{\partial D(\mathbf{A}, \mathbf{S})}{\partial a_{nm}} = \left[ -\mathbf{XS}^T + \mathbf{ASS}^T \right]_{nm} + \alpha_A \frac{\partial J_A(\mathbf{A})}{\partial a_{nm}}$$

$$\frac{\partial D(\mathbf{A}, \mathbf{S})}{\partial s_{mt}} = \left[ -\mathbf{A}^T \mathbf{X} + \mathbf{A}^T \mathbf{AS} \right]_{mt} + \alpha_S \frac{\partial J_S(\mathbf{S})}{\partial s_{mt}}$$

a) A. Cichocki, R. Zdunek, and S. Amari, “Csiszár’s Divergences for Non-negative Matrix Factorization: Family of New Algorithms,” *LNCS 3889*, 32-39 (2006).



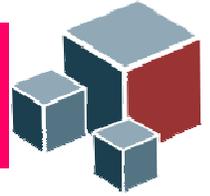
## Maximum a posteriori probability BSS/NMF

Maximization of *a-posterior* probability (MAP)  $P(\mathbf{A}, \mathbf{S} | \mathbf{X})$  yields:

$$(\mathbf{A}^*, \mathbf{S}^*) = \max_{\mathbf{A}, \mathbf{S}} P(\mathbf{A}, \mathbf{S} | \mathbf{X}) \propto \max_{\mathbf{A}, \mathbf{S}} P(\mathbf{X} | \mathbf{A}, \mathbf{S}) P(\mathbf{A}) P(\mathbf{S}) \quad s.t. \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}$$

Above formulation is equivalent to maximizing likelihood  $P(\mathbf{X} | \mathbf{A}, \mathbf{S})$  and maximizing prior probabilities  $P(\mathbf{A})$  and  $P(\mathbf{S})$ . Assuming normal distribution of approximation error  $\mathbf{E} = \mathbf{X} - \mathbf{AS}$  this yields:

$$(\mathbf{A}^*, \mathbf{S}^*) = \arg \min_{(\mathbf{A}, \mathbf{S})} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \alpha_S J_S(\mathbf{S}) + \alpha_A J_A(\mathbf{A}) \quad s.t. \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}.$$



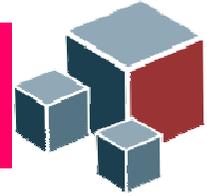
## Maximum a posteriori probability BSS/NMF

Assuming non-informative prior on  $\mathbf{A}$ :  $P(\mathbf{A})=\text{const}$  and Laplacian (sparse) prior on  $\mathbf{S}$ :  $P(\mathbf{S}) = \exp(-(|s_1| + \dots + |s_M|))$  yields:

$$(\mathbf{A}^*, \mathbf{S}^*) = \arg \min_{(\mathbf{A}, \mathbf{S})} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \alpha_S \|\mathbf{S}\|_1 \quad s.t. \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}.$$

It is possible to select for  $P(\mathbf{S})$  prior other than Laplacian. That leads to general sparseness constrained factorization:

$$(\mathbf{A}^*, \mathbf{S}^*) = \arg \min_{(\mathbf{A}, \mathbf{S})} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \alpha_S \|\mathbf{S}\|_p \quad s.t. 0 < p \leq 1, \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}.$$



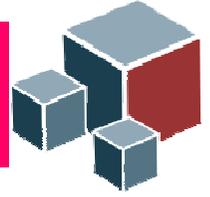
## Nonnegative matrix factorization

Since NMF problem deals with non-negative variables the idea is to automatically ensure non-negativity of  $\mathbf{A}$  and  $\mathbf{S}$  through learning. That can be achieved by multiplicative learning equations:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\nabla_{\mathbf{A}}^{-} D(\mathbf{A}, \mathbf{S})}{\nabla_{\mathbf{A}}^{+} D(\mathbf{A}, \mathbf{S})} \quad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\nabla_{\mathbf{S}}^{-} D(\mathbf{A}, \mathbf{S})}{\nabla_{\mathbf{S}}^{+} D(\mathbf{A}, \mathbf{S})}$$

where  $\otimes$  denotes entry-wise multiplication,  $\nabla_{\mathbf{A}}^{-} D(\mathbf{A}, \mathbf{S})$  and  $\nabla_{\mathbf{A}}^{+} D(\mathbf{A}, \mathbf{S})$  denote respectively negative and positive part of the gradient  $\nabla_{\mathbf{A}} D(\mathbf{A}, \mathbf{S})$ . Likewise,  $\nabla_{\mathbf{S}}^{-} D(\mathbf{A}, \mathbf{S})$  and  $\nabla_{\mathbf{S}}^{+} D(\mathbf{A}, \mathbf{S})$  are negative and positive part of the gradient  $\nabla_{\mathbf{S}} D(\mathbf{A}, \mathbf{S})$ .

When gradients converge to zero corrective terms converge to one. Since learning equations include multiplications and divisions of non-negative terms, non-negativity is ensured automatically.



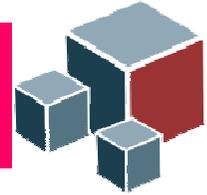
## Nonnegative matrix factorization

Multiplicative learning rules for NMF based on regularized squared  $L_2$ -norm of the approximation are obtained as:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\left[ \mathbf{X}\mathbf{S}^T - \alpha_A \frac{\partial J_A(\mathbf{A})}{\partial \mathbf{A}} \right]_+}{\mathbf{A}\mathbf{S}\mathbf{S}^T + \varepsilon \mathbf{1}_{NM}} \quad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\left[ \mathbf{A}^T \mathbf{X} - \alpha_S \frac{\partial J_S(\mathbf{S})}{\partial \mathbf{S}} \right]_+}{\mathbf{A}^T \mathbf{A}\mathbf{S} + \varepsilon \mathbf{1}_{MT}}$$

where  $[x]_+ = \max\{\varepsilon, x\}$  with small  $\varepsilon$ . For  $L_1$ -norm based regularization, derivatives of sparseness constraints in above expressions are equal to 1, i.e.:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\left[ \mathbf{X}\mathbf{S}^T - \alpha_A \mathbf{1}_{NM} \right]_+}{\mathbf{A}\mathbf{S}\mathbf{S}^T + \varepsilon \mathbf{1}_{NM}} \quad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\left[ \mathbf{A}^T \mathbf{X} - \alpha_S \mathbf{1}_{MT} \right]_+}{\mathbf{A}^T \mathbf{A}\mathbf{S} + \varepsilon \mathbf{1}_{MT}}$$



## Nonnegative matrix factorization

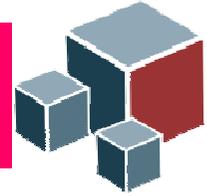
NMF through minimization of Froebenius norm is optimal when data are corrupted by additive Gaussian noise. Another cost function that is used most often for NMF is Kullback-Leibler divergence, also called I-divergence [a]:

$$D(\mathbf{X} \parallel \mathbf{AS}) = \sum_{nt} \left( x_{nt} \ln \frac{x_{nt}}{[\mathbf{AS}]_{nt}} - x_{nt} + [\mathbf{AS}]_{nt} \right)$$

It can be shown that minimization of Kullback-Leibler divergence is equivalent to the maximization of the Poisson likelihood

$$L(\mathbf{X} | \mathbf{A}, \mathbf{S}) = \prod_{nt} \left( \frac{[\mathbf{AS}]_{nt}}{x_{nt}!} \exp(-[\mathbf{AS}]_{nt}) \right)$$

a) A. Cichocki, R. Zdunek, and S. Amari, "Csiszár's Divergences for Non-negative Matrix Factorization: Family of New Algorithms," *LNCS* **3889**, 32-39 (2006).



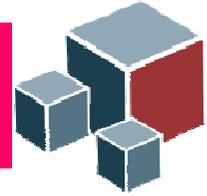
## Nonnegative matrix factorization

Calculating gradients of I-divergence cost function w.r.t.  $\mathbf{a}_{nm}$  and  $\mathbf{s}_{mt}$  the following learning rules in MATLAB notation are obtained

$$\mathbf{S}^{(k+1)} = \left( \mathbf{S}^{(k)} \otimes \left( \mathbf{A}^T \left( \mathbf{X} \oslash \left( \mathbf{A} \mathbf{S}^{(k)} \right) \right) \right)^{.[\omega]} \right)^{.[1+\alpha_S]}$$

$$\mathbf{A}^{(k+1)} = \left( \mathbf{A}^{(k)} \otimes \left( \left( \mathbf{X} \oslash \left( \mathbf{A}^{(k)} \mathbf{S} \right) \right) \mathbf{S}^T \right)^{.[\omega]} \right)^{.[1+\alpha_A]}$$

where  $\otimes$  denotes component-wise multiplication, and  $\oslash$  denotes component-wise division. Relaxation parameter  $\omega \in (0, 2]$  provides improvement of the convergence, while  $\alpha_S \geq 0$  and  $\alpha_A \geq 0$  are sparseness constraints that are typically confined in the interval  $[0.001, 0.005]$ .



## Nonnegative matrix factorization

In order to obtain NMF algorithms optimal for different statistics of data and noise the  $\alpha$ -divergence cost function can be used

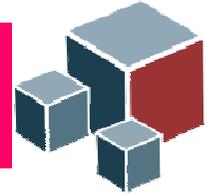
$$D(\mathbf{X} \parallel \mathbf{AS}) = \frac{1}{\alpha(\alpha-1)} \sum_{nt} \left( x_{nt}^\alpha [\mathbf{AS}]_{nt}^{1-\alpha} - \alpha x_{nt} + (\alpha-1) [\mathbf{AS}]_{nt} \right)$$

I-divergence is obtained in the limit when  $\alpha \rightarrow 1$ , and dual Kullback-Leibler divergence when  $\alpha \rightarrow 0$ . Using MATLAB notation the following update rules are obtained for  $\alpha \neq 0, 1$ .

$$\mathbf{S} \leftarrow \left( \mathbf{S} .* \left( \mathbf{A}^T .* \left( \mathbf{X} ./ [\mathbf{AS} + \varepsilon \mathbf{1}_{NT}]_+ \right)^\alpha \right)^{\omega/\alpha} \right)^{1+\alpha_S}$$

$$\mathbf{A} \leftarrow \left( \mathbf{A} .* \left( \left( \mathbf{X} ./ [\mathbf{AS} + \varepsilon \mathbf{1}_{NT}]_+ \right)^\alpha \mathbf{S}^T \right)^{\omega/\alpha} \right)^{1+\alpha_A}$$

$$\mathbf{A} \leftarrow \mathbf{A} * \text{diag} \left( 1 ./ \text{sum}(\mathbf{A}, 1) \right)$$



## Hierarchical ALS NMF

Local or hierarchical ALS NMF algorithms were derived in [a, b, c]. They employ minimization of the global cost function to learn the mixing matrix and minimization of set of local cost functions to learn the sources. Global cost function can for example be squared Euclidean norm:

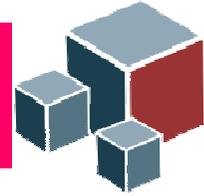
$$D(\mathbf{X} \|\mathbf{AS}) = \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 + \alpha_A J_A(\mathbf{A})$$

Local cost functions can be also squared Euclidean norms

$$D^{(m)}(\mathbf{X}^{(m)} \|\mathbf{a}_m \mathbf{s}_m) = \frac{1}{2} \|\mathbf{X}^{(m)} - \mathbf{a}_m \mathbf{s}_m\|_F^2 + \alpha_s^{(m)} J_S(\mathbf{s}_m) + \alpha_a^{(m)} J_a(\mathbf{a}_m) \quad m = 1, \dots, M$$

$$\mathbf{X}^{(m)} = \mathbf{X} - \sum_{j \neq m} \mathbf{a}_j \mathbf{s}_j$$

- a) A. Cichocki, R. Zdunek, S.I. Amari, Hierarchical ALS Algorithms for Nonnegative Matrix Factorization and 3D Tensor Factorization, *LNCS* **4666** (2007) 169-176
- b) A. Cichocki, A-H. Phan, R. Zdunek, and L.-Q. Zhang, "Flexible component analysis for sparse, smooth, nonnegative coding or representation," *LNCS* **4984**, 811-820 (2008).
- c) A. Cichocki, R. Zdunek, S. Amari, Nonnegative Matrix and Tensor Factorization, *IEEE Sig. Proc. Mag.* **25** (2008) 142-145.



## Hierarchical ALS NMF

Minimization of above cost functions in ALS manner with  $L_1$ -based sparseness constraints imposed on  $\mathbf{A}$  and/or  $\mathbf{S}$  yields

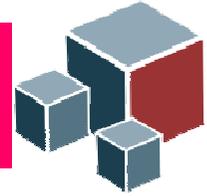
$$\left\{ \underline{\mathbf{s}}_m \leftarrow \left[ \mathbf{a}_m^T \mathbf{X}^{(m)} - \alpha_s^{(m)} \mathbf{1}_{1 \times T} \right]_+ \right\}_{m=1}^M$$

$$\mathbf{A} \leftarrow \left[ \left( \mathbf{X} \mathbf{S}^T - \alpha_A \mathbf{1}_{N \times M} \right) \left( \mathbf{S} \mathbf{S}^T + \lambda \mathbf{I}_M \right)^{-1} \right]_+$$

$$\left\{ \mathbf{a}_m \leftarrow \mathbf{a}_m / \|\mathbf{a}_m\|_2 \right\}_{m=1}^M$$

where  $\mathbf{I}_{1 \times T}$  is an  $M \times M$  identity matrix,  $\mathbf{1}_{1 \times T}$  and  $\mathbf{1}_{N \times M}$  are row vector and matrix with all entries equal to one and  $[\xi]_+ = \max\{\varepsilon, \xi\}$  (e.g.,  $\varepsilon = 10^{-16}$ ).

Regularization constant  $\lambda$  changes as a function of the iteration index as  $\lambda_k = \lambda_0 \exp(-k/\tau)$  (with  $\lambda_0 = 100$  and  $\tau = 0.02$  in the experiments).



## Multilayer NMF

Additional improvement in the performance of the NMF algorithms is obtained when they are applied in the multilayer mode [a,b], whereas sequential decomposition of the nonnegative matrices is performed as follows.

In the first layer, the basic approximation decomposition is performed:

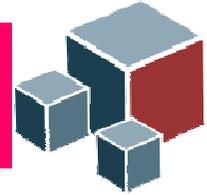
$$\mathbf{X} \cong \mathbf{A}^{(1)} \mathbf{S}^{(1)} \in \mathbb{R}_{0+}^{N \times T}$$

In the second layer result from the first layer is used to build up new input data matrix for the second layer  $\mathbf{X} \leftarrow \mathbf{S}^{(1)} \in \mathbb{R}_{0+}^{M \times T}$ . This yields  $\mathbf{X}^{(1)} \cong \mathbf{A}^{(2)} \mathbf{S}^{(2)} \in \mathbb{R}_{0+}^{M \times T}$ .

After  $L$  layers data decomposes as follows:  $\mathbf{X} \cong \mathbf{A}^{(1)} \mathbf{A}^{(2)} \dots \mathbf{A}^{(L)} \mathbf{S}^{(L)}$ .

a) A. Cichocki, and R. Zdunek, "Multilayer Nonnegative Matrix Factorization," *El. Letters* **42**, 947-948 (2006).

b) A. Cichocki, R. Zdunek, A. H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations-Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, John Wiley, 2009.



## Multi-start initialization for NMF algorithms

Combined optimization of the cost function  $D(\mathbf{X}|\mathbf{A}\mathbf{S})$  with respect to  $\mathbf{A}$  and  $\mathbf{S}$  is non-convex optimization problem. Hence, some strategy is necessary to decrease probability that optimization process will get stuck in some local minima. Such procedure is outlined with the following pseudo code: Select  $R$ -number of restarts,  $K_i$  number of alternating steps,  $K_f$  number of final alternating steps.

**for**  $r = 1, \dots, R$  **do**

*Initialize randomly  $\mathbf{A}^{(0)}$  and  $\mathbf{S}^{(0)}$*

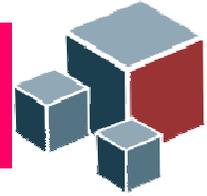
$\{\mathbf{A}^{(r)}, \mathbf{S}^{(r)}\} \leftarrow \text{nmf\_algorithm}(\mathbf{X}, \mathbf{A}^{(0)}, \mathbf{S}^{(0)}, K_i);$

*compute  $d = D(\mathbf{X}|\mathbf{A}^{(r)}\mathbf{S}^{(r)})$ ;*

**end**

$r_{min} = \text{argmin}_{1 \leq n \leq R} d_n$ ;

$\{\mathbf{A}, \mathbf{S}\} \leftarrow \text{nmf\_algorithm}(\mathbf{X}, \mathbf{A}^{(r_{min})}, \mathbf{S}^{(r_{min})}, K_f);$



## Non-negative matrix under-approximation (NMU)

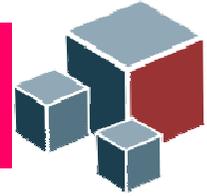
NMF algorithms outlined before require a priori knowledge of sparseness related regularization (trade off) constant.

A sequential approach to NMF has been recently proposed in [a] by estimating rank-1 one factors  $\mathbf{a}_m \mathbf{s}_m$  one at a time. Each time  $\mathbf{a}_m \mathbf{s}_m$  is estimated it is removed from  $\mathbf{X} \rightarrow \mathbf{X} - \mathbf{a}_m \mathbf{s}_m$ . To prevent subtraction from being negative the under-approximation constraint is imposed on  $\mathbf{a}_m \mathbf{s}_m$ :  $\mathbf{a}_m \mathbf{s}_m \leq \mathbf{X}$ .

Hence, the NMU algorithm is obtained as a solution of:

$$(\mathbf{A}^*, \mathbf{S}^*) = \arg \min_{(\mathbf{A}, \mathbf{S})} \frac{1}{2} \|\mathbf{X} - \mathbf{AS}\|_F^2 \quad s.t. \quad \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}, \mathbf{AS} \leq \mathbf{X}.$$

a) N. Gillis, and F. Glineur, "Using underapproximations for sparse nonnegative matrix factorization," *Patt. Recog.*, vol. 43, pp. 1676-1687, 2010.



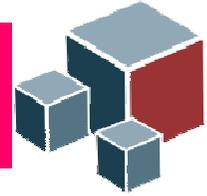
## Non-negative matrix under-approximation (NMU)

Theorem 1 in [a] proves that number of nonzero entries in  $\mathbf{A}$  and  $\mathbf{S}$  is less than in  $\mathbf{X}$ . Thus, the underapproximation constraint ensures sparse (parts based) factorization of  $\mathbf{X}$ . This, however, does not imply that  $\mathbf{A}$  and  $\mathbf{S}$  obtained by enforcing underapproximation constrain yields the sparseset decomposition of  $\mathbf{X}$ .

However, since no explicit regularization is used there are no difficulties associated with selecting values of regularization constants.

MATLAB code for NMU algorithm is available at:  
<https://sites.google.com/site/nicolasgillis/code>

a) N. Gillis, and F. Glineur, "Using underapproximations for sparse nonnegative matrix factorization," *Patt. Recog.*, vol. 43, pp. 1676-1687, 2010.



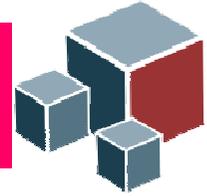
## Non-negative matrix factorization with $L_0$ -constraint (NMF\_L0)

The NMF\_L0 algorithm, [a], imposes explicit  $L_0$ -constraint on entries of  $\mathbf{S}$ , i.e. number of nonzero entries is tried to be minimized explicitly by integrating nonnegativity constraint in the OMP algorithm. That is achieved through modifications of the nonnegative least square (NNLS) algorithm, [b], called sparse NNLS and recursive sparse NNLS. The mixing matrix is updated by some of standards dictionary update methods.

The „weak” side of the NMF\_L0 algorithm is that, in addition to number of sources  $M$ , the maximal number of overlapped sources  $K$  has to be known *a priori*. Quite often that is hard to achieve in practice.

MATLAB code for NMF\_L0 algorithm is available at:  
<http://www3.spssc.tugraz.at/people/robert-peharz>.

- a) R. Peharz, F. Pernkopf, "Sparse nonnegative matrix factorization with  $\ell^0$  constraints," *Neurocomputing*, vol. 80, pp. 38-46, 2012.
- b) C. Lawson, R. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.



## Comparative performance analysis

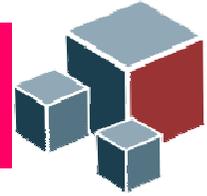
NMU and NMF\_L0 algorithms were compared on mixing matrix estimation using model:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N}$$

where  $N=100$ ,  $M=250$ ,  $T=2500$ ,  $\text{SNR [dB]} \in \{10, 20, 30, \text{inf}\}$ ,  $K \in \{10, 30, 50\}$ .

Each source was generated with probability of being zero equal to 0.8. The nonzero state was generated by uniform distribution on interval (0,1].

Entries of the  $N \times M$  mixing matrix were drawn from uniform distribution on interval (0,1]. The mixing matrix has been column normalized to unit  $L_2$  norm.



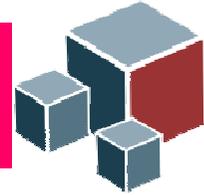
## Comparative performance analysis

Performance measures:

$$\varepsilon(\mathbf{S}) = 10 \log_{10} \left( \frac{\| \text{diag}(\mathbf{R}_{\text{SS}}) \|_2^2}{\| \text{diag}(\mathbf{R}_{\text{SS}} - \mathbf{R}_{\text{S}\hat{\text{S}}}) \|_2^2} \right)$$

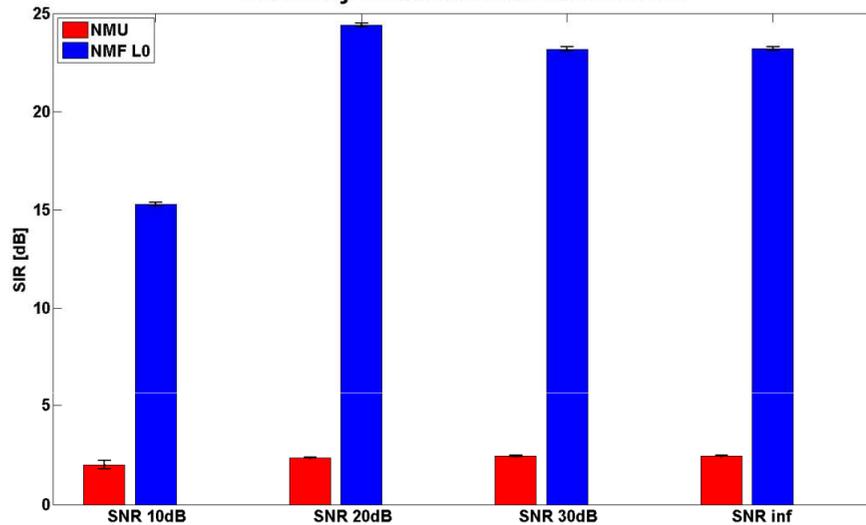
$$\varepsilon(\mathbf{A}) = 10 \log_{10} \left( \frac{\| \text{diag}(\mathbf{R}_{\text{AA}}) \|_2^2}{\| \text{diag}(\mathbf{R}_{\text{AA}} - \mathbf{R}_{\text{A}\hat{\text{A}}}) \|_2^2} \right)$$

have been averaged after 10 runs, where  $\mathbf{R}_{\text{SS}}$ , respectively  $\mathbf{R}_{\text{AA}}$ , and  $\mathbf{R}_{\text{S}\hat{\text{S}}}$ , respectively  $\mathbf{R}_{\text{A}\hat{\text{A}}}$ , are autocovariance and cross-covariance matrices calculated after permutation indeterminacy has been resolved.

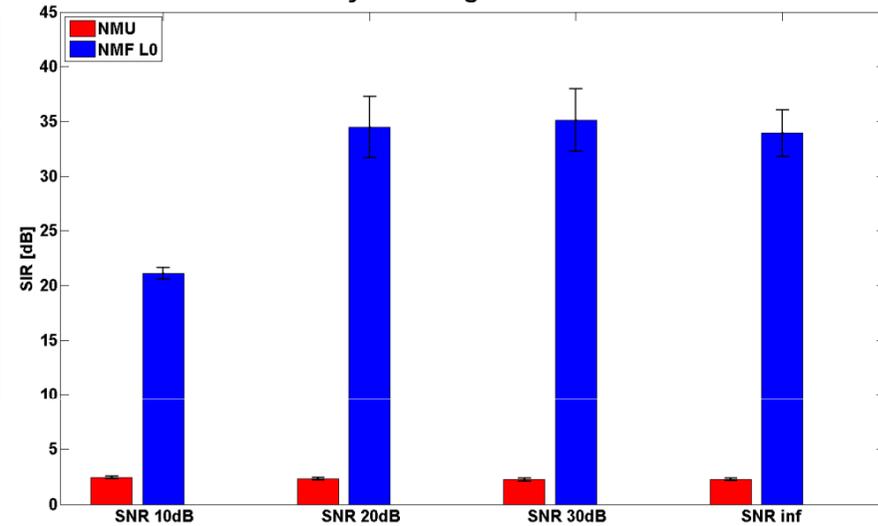


## Comparative performance analysis

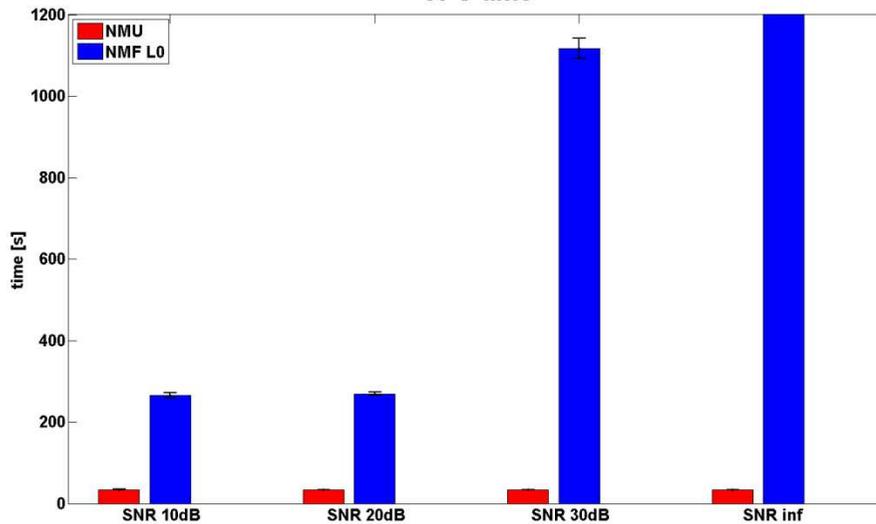
Accuracy of source matrix estimation



Accuracy of mixing matrix estimation

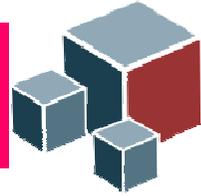


CPU time

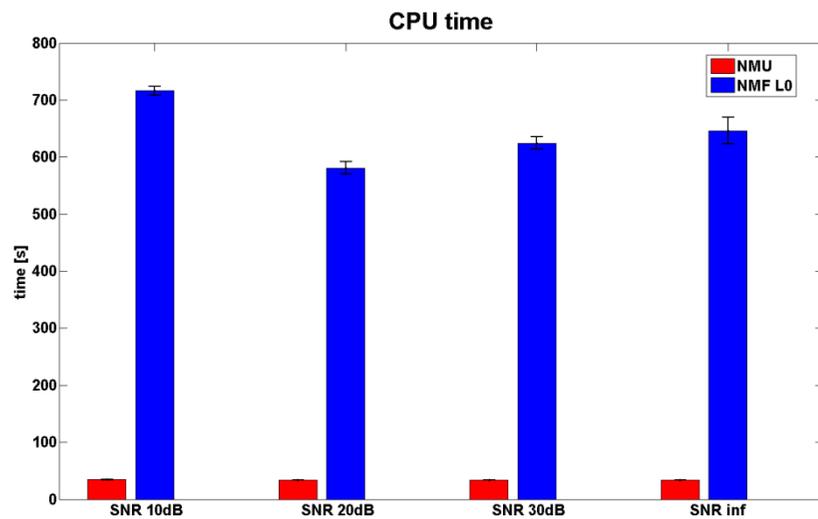
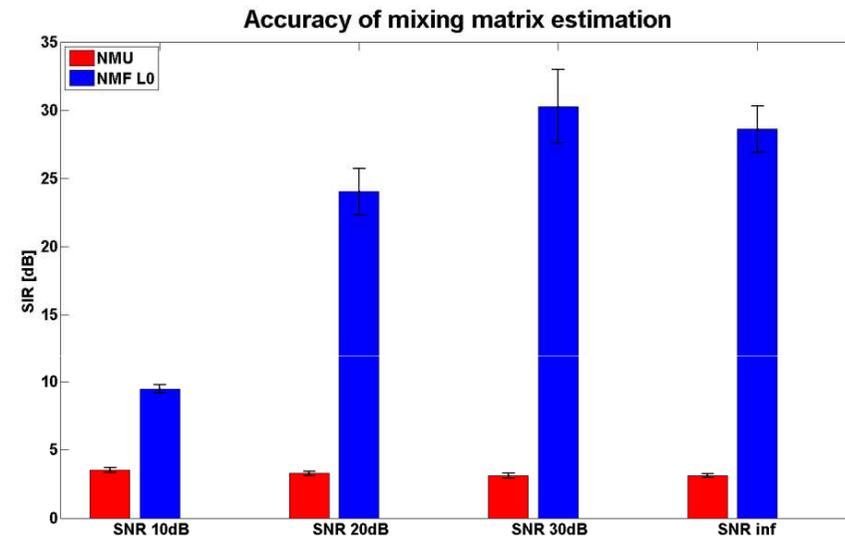
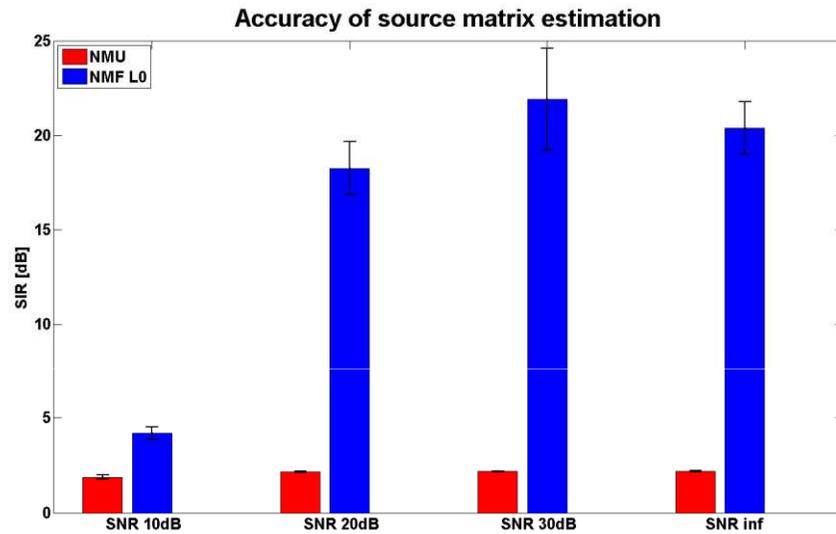


$K=10$

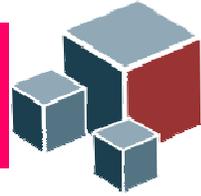
The sparse coding part of the NMF\_L0 is very slow when data matrix is very sparse. That can be „fixed” by adding (very) small amount of noise.



# Comparative performance analysis

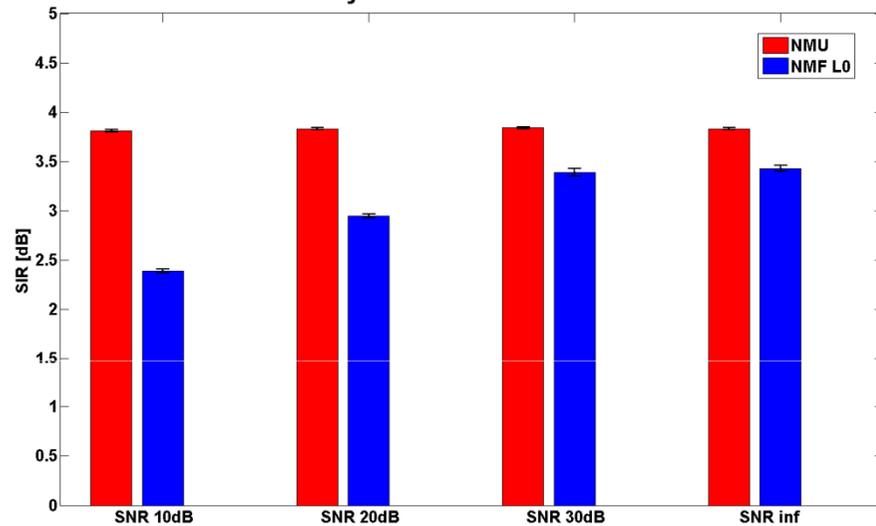


K=30

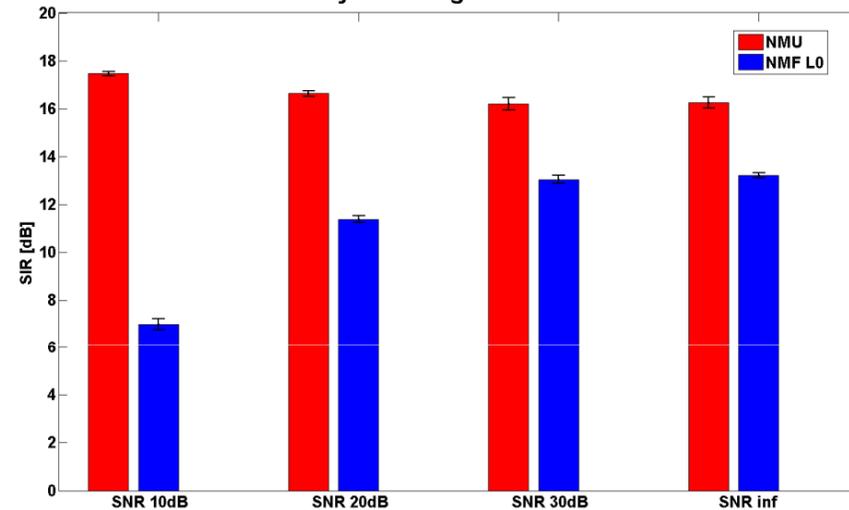


# Comparative performance analysis

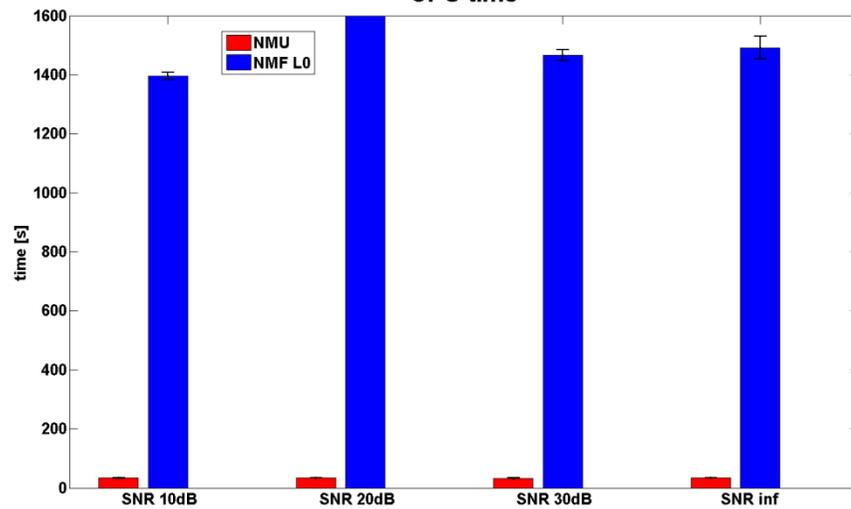
Accuracy of source matrix estimation



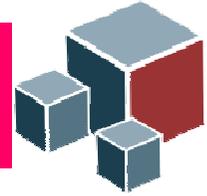
Accuracy of mixing matrix estimation



CPU time



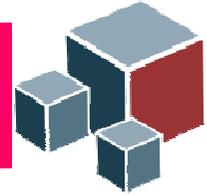
K=50



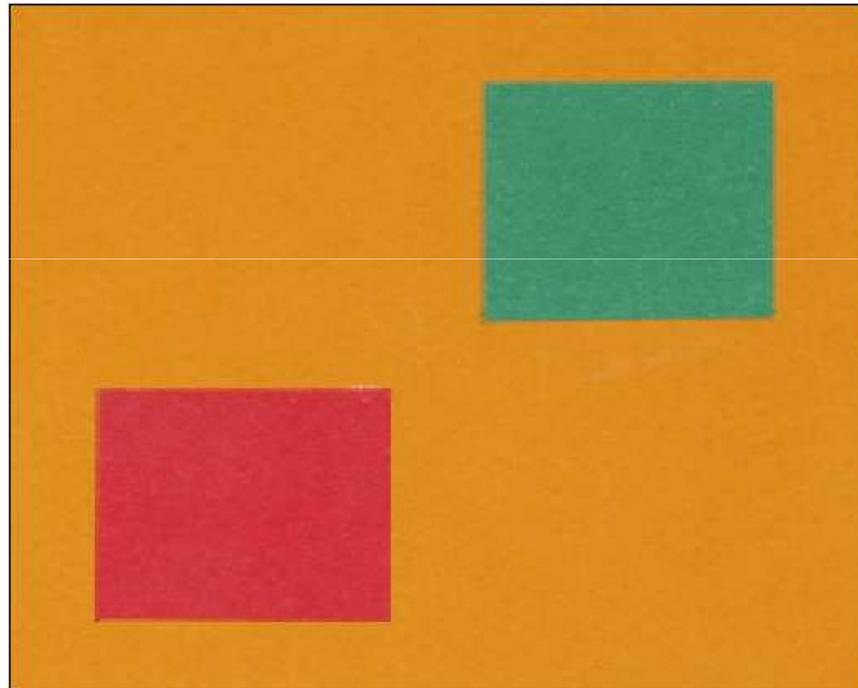
## Comparative performance analysis -conclusions

NMF\_L0 algorithm yields much better accuracy than NMU algorithm, especially in mixing matrix estimation, provided that number of overlapped sources  $K$  is known.

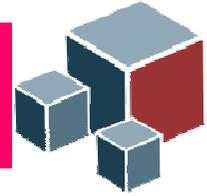
NMU can be good alternative for problems where no *a priori* information on number of overlapped sources is known in advance.



## Sparseness constrained unsupervised multichannel image decomposition



Original RGB image



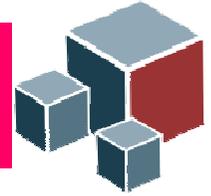
## Multichannel image and linearan mixing model

$$\mathbf{X}=\mathbf{A}\mathbf{S} \quad \mathbf{X} \in \mathbb{R}_{0+}^{N \times T}, \mathbf{S} \in \mathbb{R}_{0+}^{M \times T}, \mathbf{A} \in \mathbb{R}_{0+}^{N \times M} \quad (1)$$

In imaging spectroscopy (multispectral/RGB image) rows of  $\mathbf{X}$  are vectorized channel images (eg. red, green or blue color), columns of  $\mathbf{A}$  are spectral profiles of objects (tissues, organs) present in image  $\mathbf{X}$ , and rows of  $\mathbf{S}$  are distributions of intensities of objects (tissues, organs) present in image  $\mathbf{X}$ .

By an equivalent interpretation the model (1) is applicable to other types of co-registered multichannel images such as: hyperspectral image, multiphase CT, multispectral magnetic resonance (MR), functional MR image, imaging mass spectrometry, multimodal image obtained by image fusion (PET/CT),...

(u)BSS problem relates to unsupervised decomposition of image  $\mathbf{X}$  into anatomically meaningful components: distributions of intensities of objects present in the image  $\mathbf{X}$ .

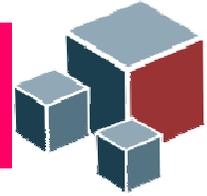


## Unsupervised decomposition of multispectral images

When degree of overlap between objects in spatial domain is very small  
i.e.  $s_m(t) * s_n(t) \approx \delta_{nm}$ , it implies  $\|\mathbf{s}(t)\|_0 \approx 1$  i.e.  $K=1$ .

RGB image decomposition problem can be solved with some SCA algorithm,  
eg. clustering and  $L_1$ -norm minimization or NMF algorithm with sparseness  
constraint.

Estimate of the mixing  $\mathbf{A}$  and number of objects  $M$  is achieved by clustering.

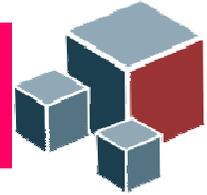


## Unsupervised decomposition of multispectral images

- Assuming unit  $L_2$ -norm of  $\mathbf{a}_m$  we can parameterize column vectors in 3D space by means of azimuth and elevation angles

$$\mathbf{a}_m = [\cos(\varphi_m) \sin(\theta_m) \quad \sin(\varphi_m) \sin(\theta_m) \quad \cos(\theta_m)]^T$$

- Due to nonnegativity constraints both angles are confined in  $[0, \pi/2]$ . Now estimation of  $\mathbf{A}$  and  $M$  is obtained by means of data clustering algorithm:
- We remove all data points close to the origin for which applies:  $\{|\mathbf{x}(t)|_2 \leq \varepsilon\}_{t=1}^T$  where  $\varepsilon$  represents some predefined threshold.
- Normalize to unit  $L_2$ -norm remaining data points  $\mathbf{x}(t)$ , i.e.,  $\{\mathbf{x}(t) \rightarrow \mathbf{x}(t)/|\mathbf{x}(t)|_2\}_{t=1}^{\bar{T}}$



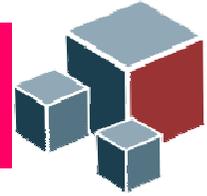
## Unsupervised decomposition of multispectral images

- Calculate function  $f(\mathbf{a})$ :

$$f(\mathbf{a}) = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2(\mathbf{x}(t), \mathbf{a})}{2\sigma^2}\right)$$

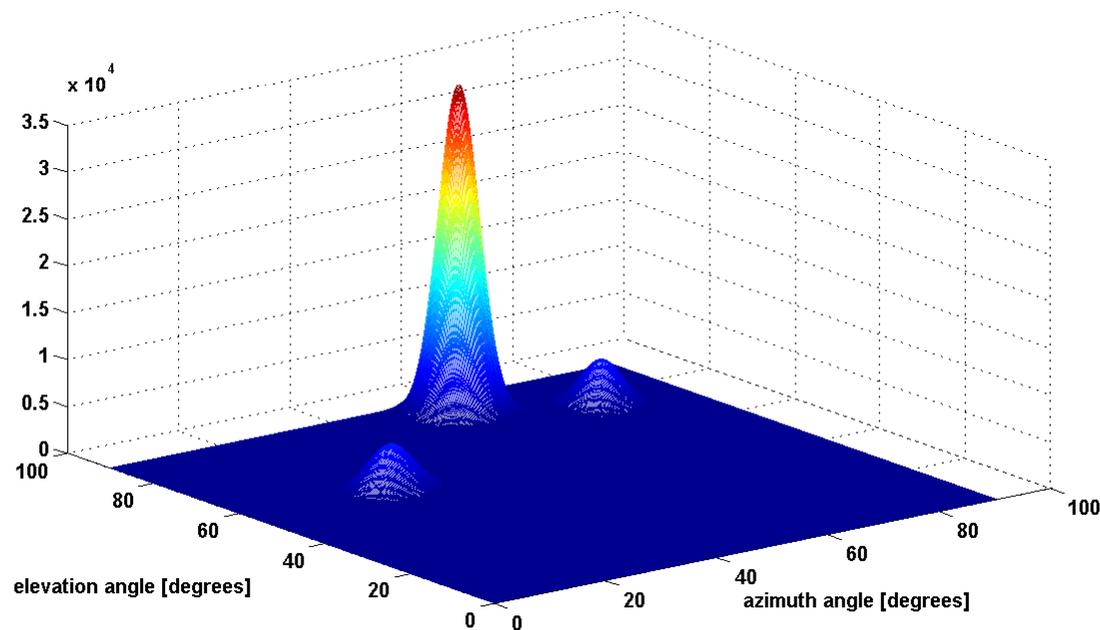
where  $d(\mathbf{x}(t), \mathbf{a}) = \sqrt{1 - (\mathbf{x}(t) \cdot \mathbf{a})^2}$  and  $(\mathbf{x}(t) \cdot \mathbf{a})$  denotes inner product. Parameter  $\sigma$  is called dispersion. If set to sufficiently small value, in our experiments this turned out to be  $\sigma \approx 0.05$ , the value of the function  $f(\mathbf{a})$  will approximately equal the number of data points close to  $\mathbf{a}$ . Thus by varying mixing angles  $0 \leq \varphi, \theta \leq \pi/2$  we effectively cluster data.

- Number of peaks of the function  $f(\mathbf{a})$  corresponds with the estimated number of materials  $M$ . Locations of the peaks correspond with the estimates of the mixing angles  $\{(\hat{\varphi}_m, \hat{\theta}_m)\}_{m=1}^M$ , i.e., mixing vectors  $\{\hat{\mathbf{a}}_m(\hat{\varphi}_m, \hat{\theta}_m)\}_{m=1}^M$ .

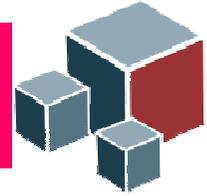


# Unsupervised decomposition of multispectral images

Clustering algorithm is used to estimate number of materials  $M$ .



Three peaks suggest existence of three materials in the RGB image i.e.  $M=3$ .

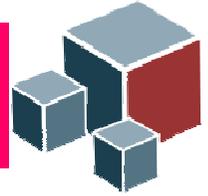


## Unsupervised decomposition of multispectral images

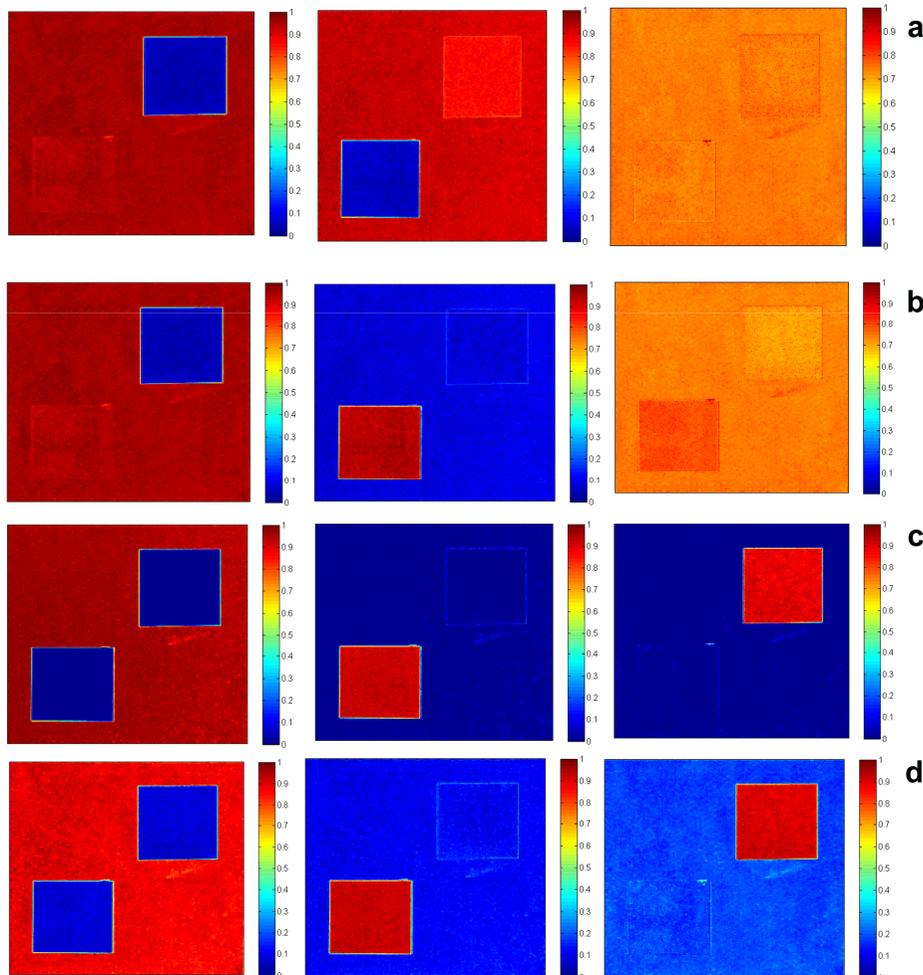
Intensity distributions of the materials were extracted by NMF with 25 layers, SCA basedd on linear programming, ICA and DCA methods.

Extracted maps were rescaled to the interval  $[0,1]$  where 0 means full absence of the material and 1 means full presence of the material.

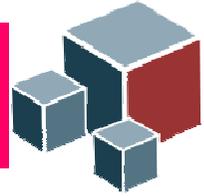
This enables visualization of the quality of decomposition process. Zero probability (absence of the material) is visualized with dark blue color and probability one (full presence of the material) is visualized with dark red color.



# Unsupervised decomposition of multispectral images

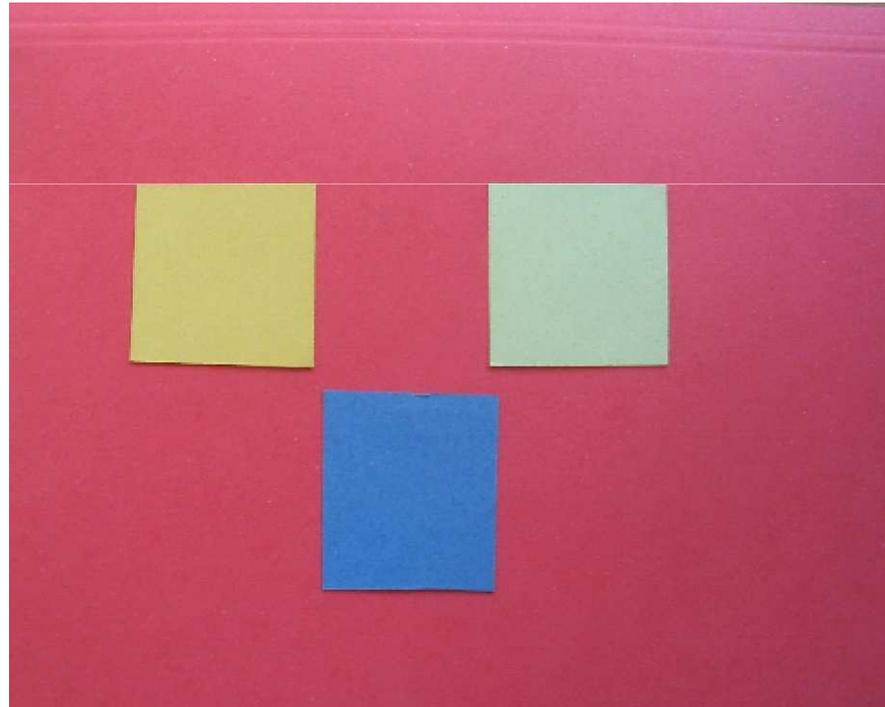


- a) DCA
- b) ICA
- c) NMF
- d) SCA- linear programming

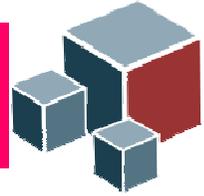


## Unsupervised decomposition of multispectral images

Consider blind decomposition of the RGB image ( $N=3$ ) composed of four materials ( $M=4$ ):

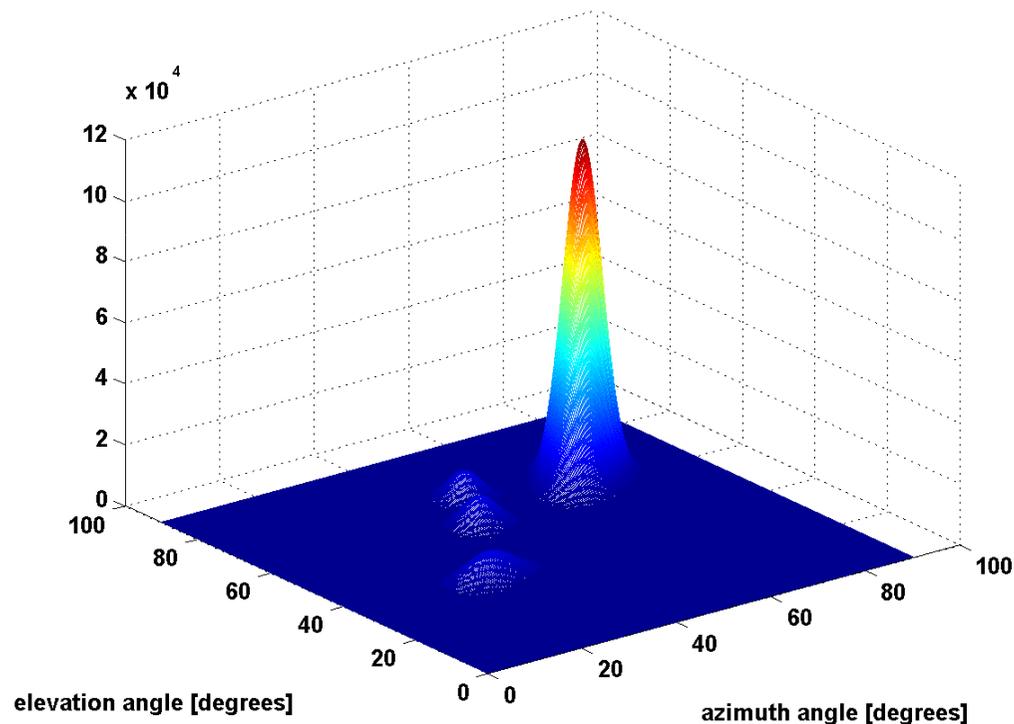


I. Kopriva and A. Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. **23**, Issue 11, pp. 590-597 (2009).

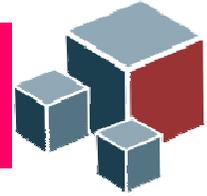


# Unsupervised decomposition of multispectral images

For shown experimental RGB image clustering function is obtained as:

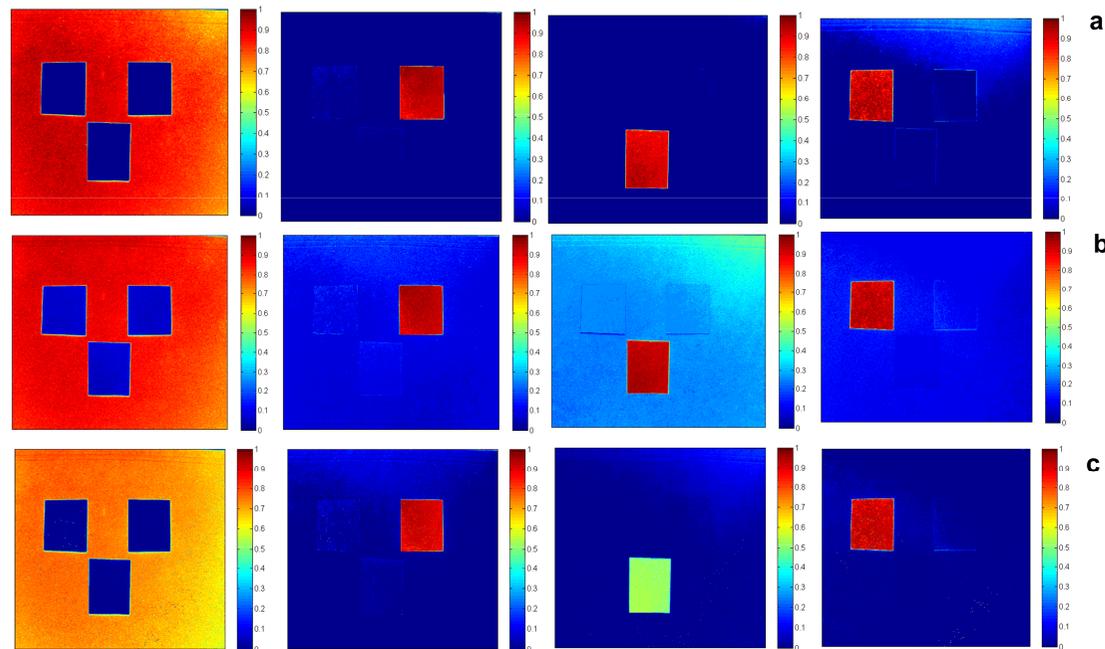


Four peaks suggest existence of four materials in the RGB image i.e.  $M=4$ .



# Unsupervised decomposition of multispectral images

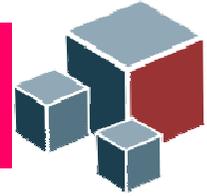
Intensity maps of the materials extracted by HALS NMF with 25 layers, linear programming and interior point method, [a], are obtained as:



a) 25 layers HALS NMF; b) Interior point method; c) Linear programming.

a) S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale  $L_1$ -Regularized Least Squares," IEEE Journal of Selected Topics in Signal Processing **1**, 606-617 (2007).

[http://www.stanford.edu/~boyd/l1\\_ls/](http://www.stanford.edu/~boyd/l1_ls/).

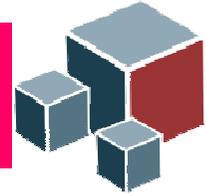


## Unsupervised decomposition of multispectral images

Since materials in the experimental RGB image are orthogonal (they do not overlap in spatial domain) we can evaluate performance of the employed blind image decomposition methods via the correlation matrix defined as  $\mathbf{G}=\mathbf{S}\mathbf{S}^T$ . For perfect estimation the correlation matrix will be diagonal and performance is visualized as deviation from diagonal matrix. To quantify decomposition quality numerically we compute the correlation index in dB scale as

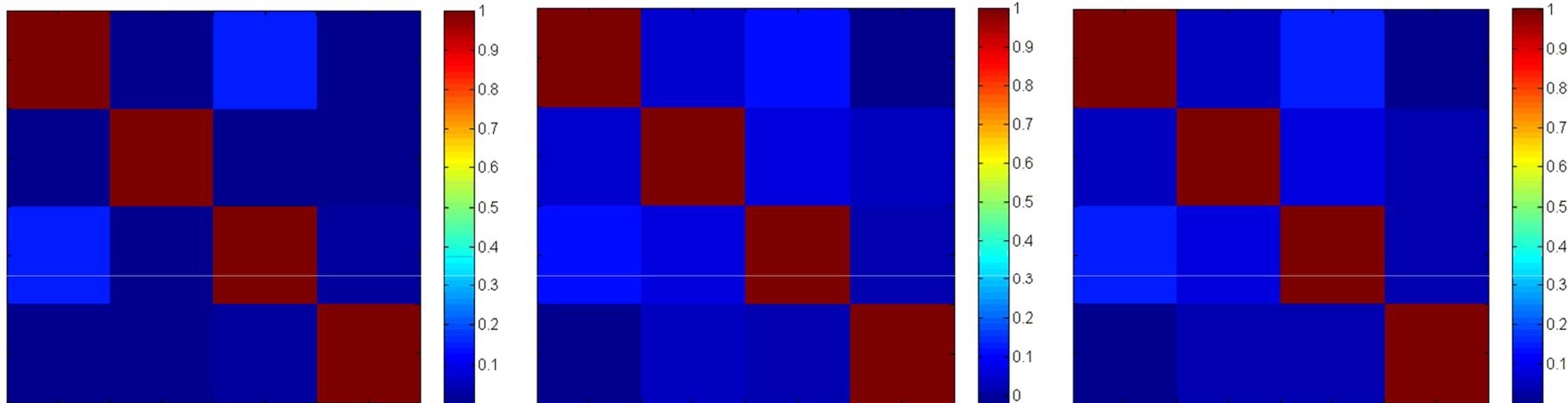
$$CR = -10\log_{10} \sum_{\substack{i,j=1 \\ j \neq i}}^M g_{ij}^2$$

where before calculating correlation matrix  $\mathbf{G}$  rows of  $\mathbf{S}$  are normalized to unit  $L_2$ -norm.



# Unsupervised decomposition of multispectral images

Correlation matrices

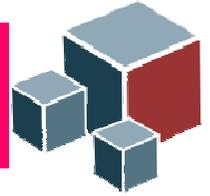


From left to right: 25 layers HALS NMF; Interior point method; c) Linear programming.

CR performance measure in dB

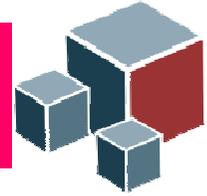
	Multilayer HALS NMF	Interior-point method	Linear program
CR [dB]	13.67	9.97	7.77
CPU time [s]*	3097	7751	3265

\* MATLAB environment on 2.4 GHz Intel Core 2 Quad Processor Q6600 desktop computer with 4GB RAM.



# Nonlinear decomposition of RGB image of unstained specimen in histopathology

I. Kopriva, M. Hadžija, M. Popović-Hadžija, M. Korolija, A. Cichocki (2011). Rational Variety Mapping for Contrast-Enhanced Nonlinear Unsupervised Segmentation of Multispectral Images of Unstained Specimen, *The American Journal of Pathology*, vol. **179**, No. 2, pp. 547-553.



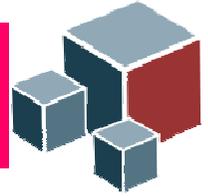
## Decomposition of RGB image in histopathology

Decomposition of objects with spectrally similar profiles is hard problem. That occurs due to poor spectral resolution or due to physiological reasons.

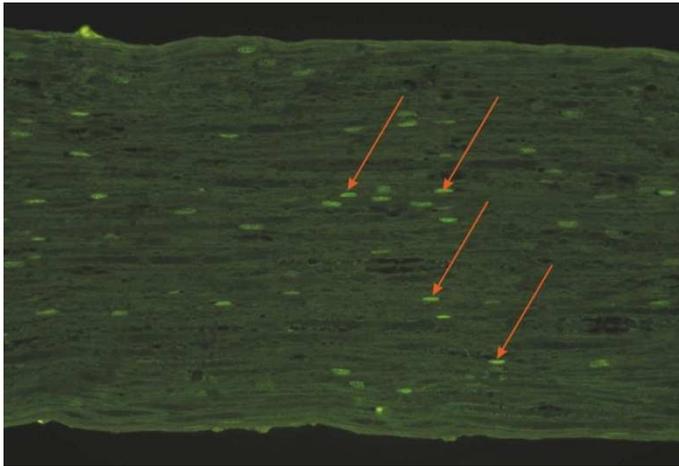
“Standard” way of enhancing visual contrast is by means of staining i.e.using contrast agents to treat a specimen.

This, possibly, can also be achieved by digital image analysis through nonlinear sparse component analysis (NSCA).

RGB image is first mapped nonlinearly (by means of explicit feature maps, a.k.a. rational variety mapping - RVM). Afterwards, sparseness constrained NMF is executed in induced space.

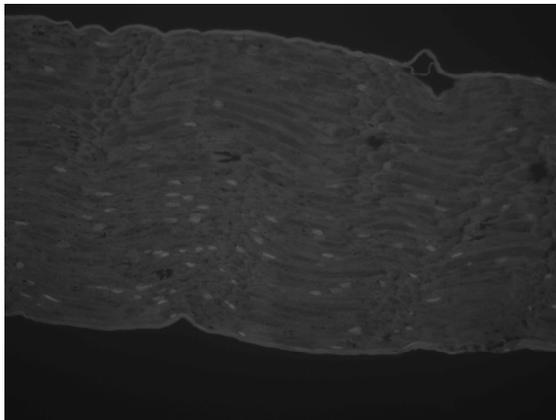


## RGB image of a nerve (*nervus ischiadicus*)

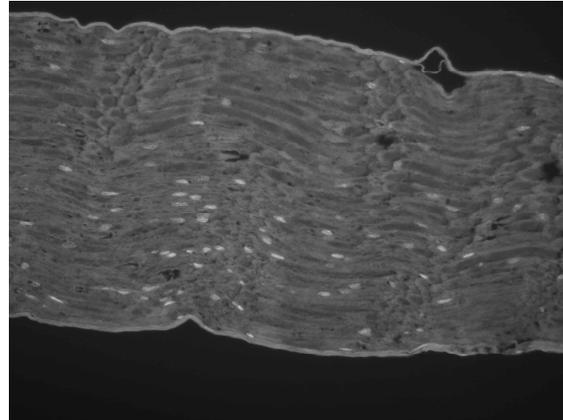


Nerves in RGB image of unstained specimen

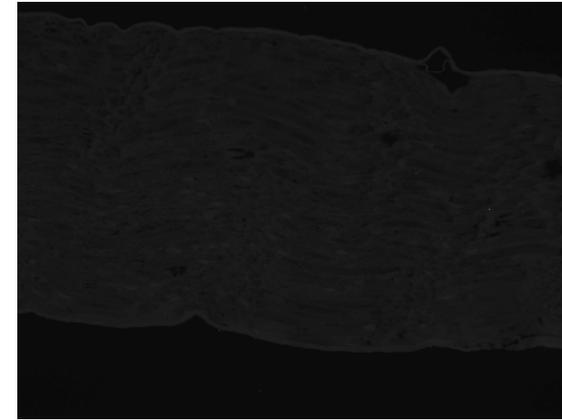
### Spectral channels of RGB image of unstained specimen



red



green



blue

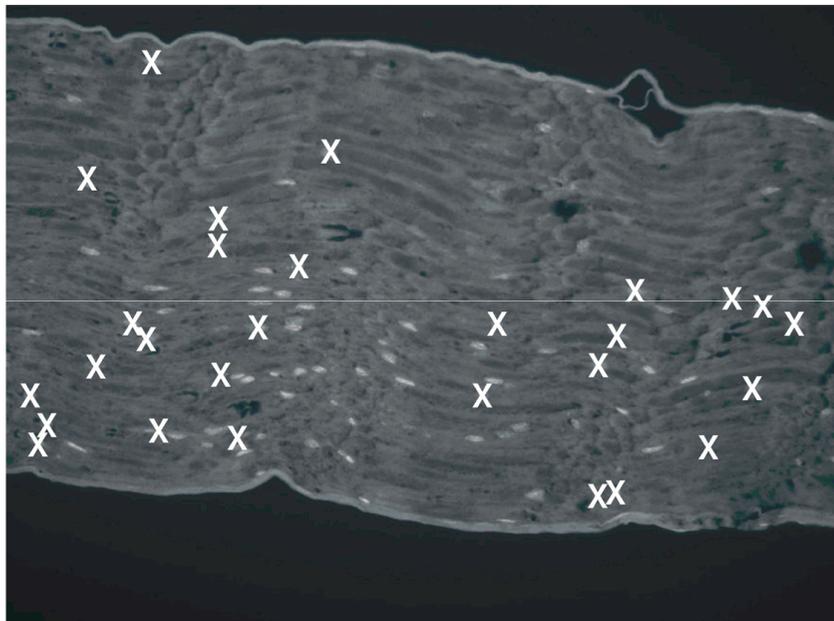
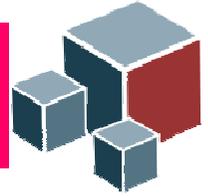
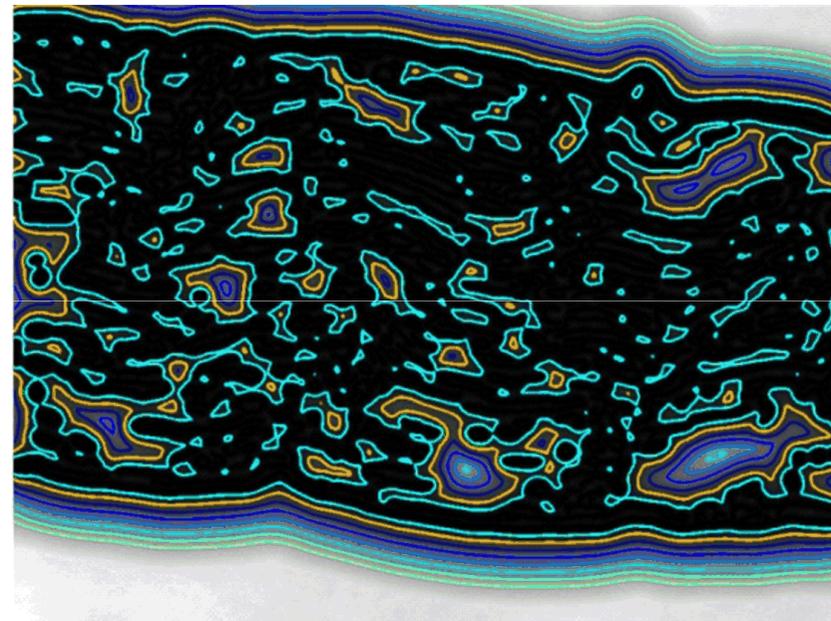
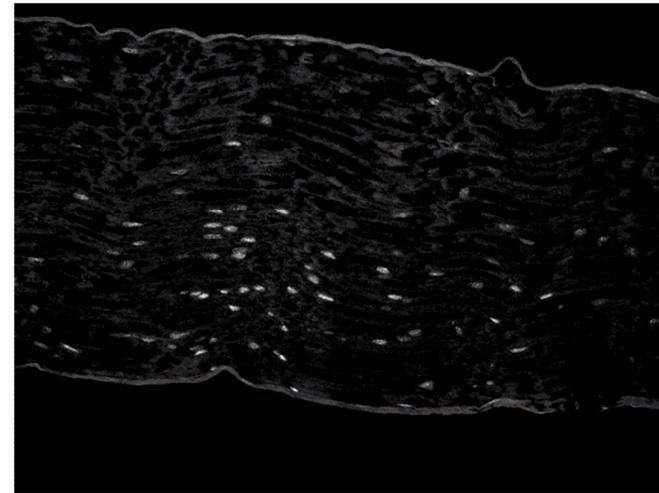
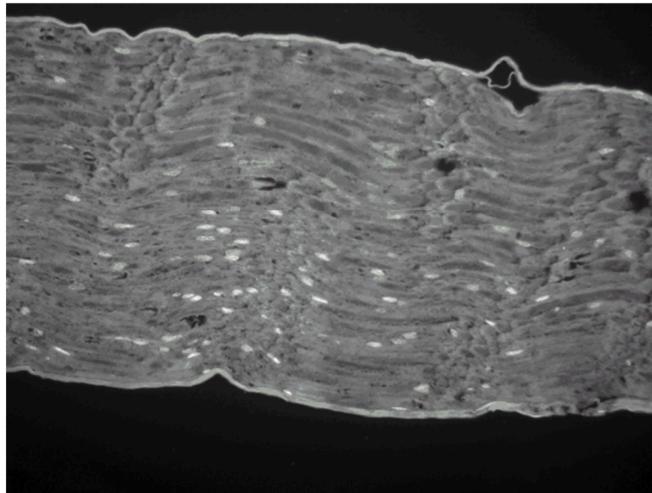
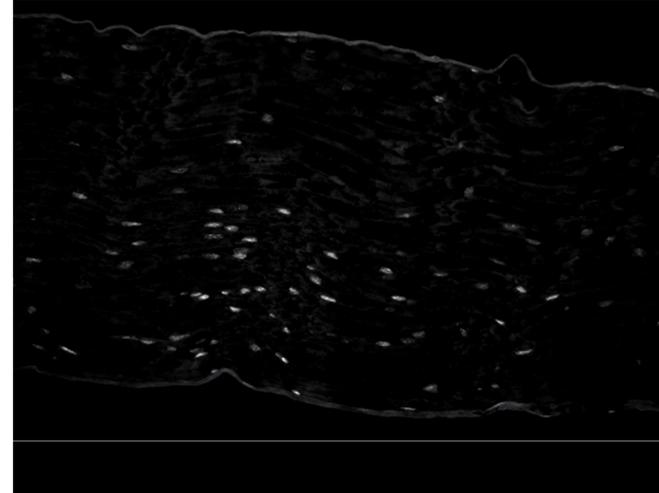
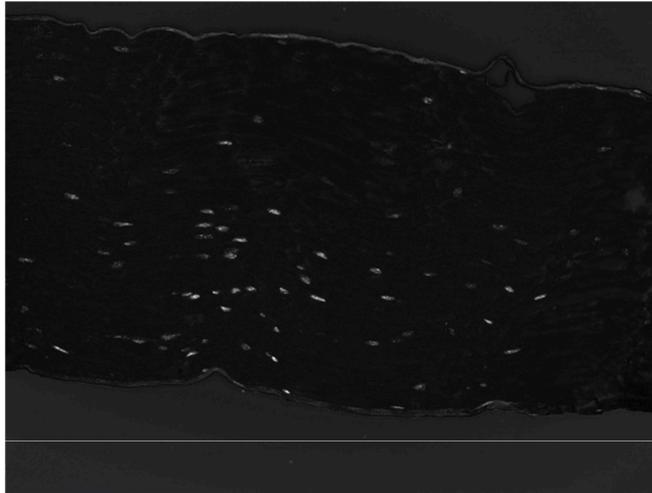
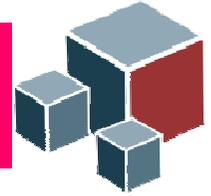


Image of unstained specimen at 510 nm wavelength (green color). White crosses denote false positive spots.

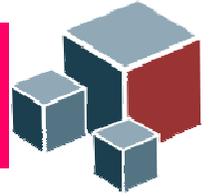


Active contours

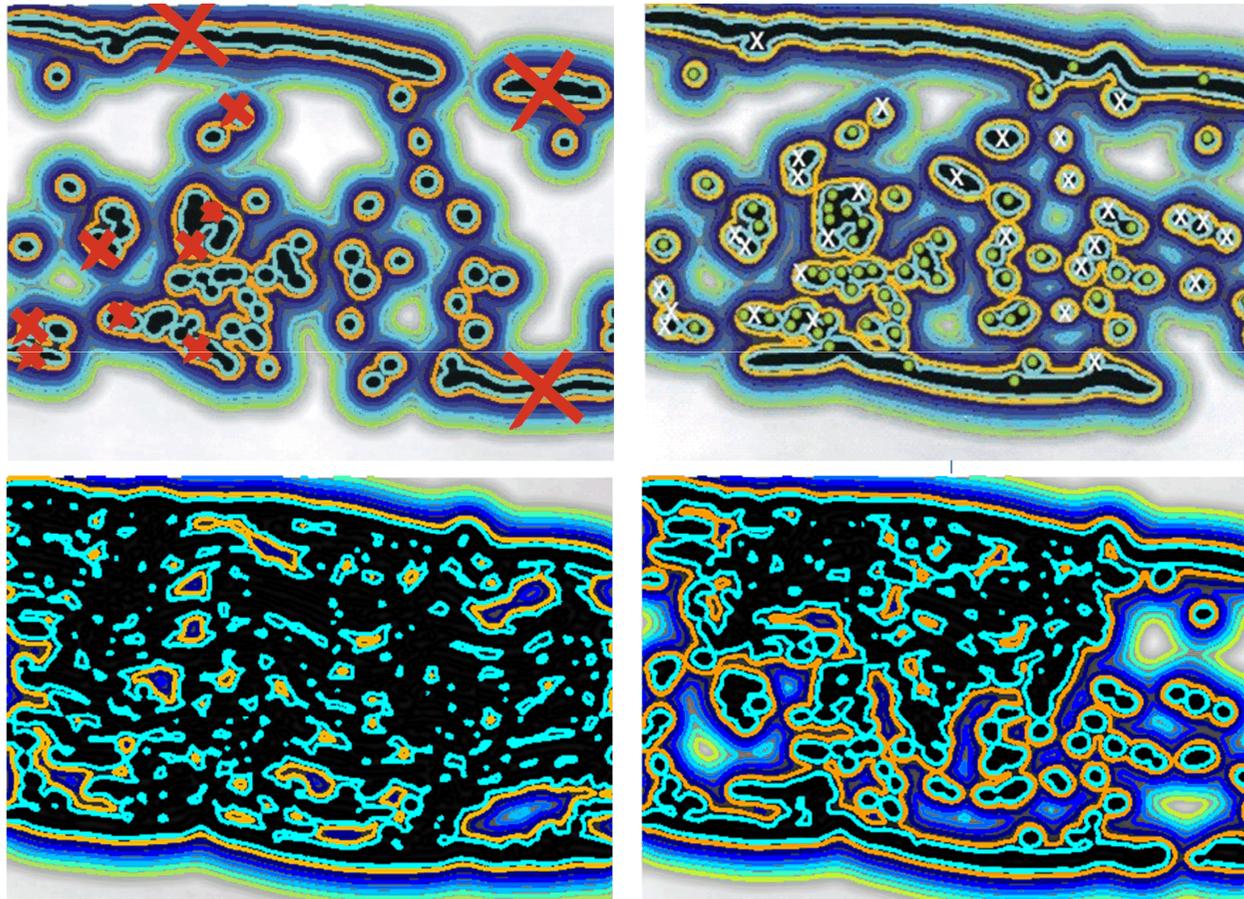


Top left: RVM2 and DCA decomposition; Bottom left: DCA decomposition only.

Top right: RVM3 and NMU decomposition; Bottom right: NMU decomposition only.

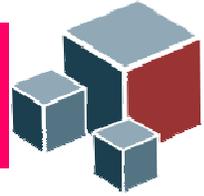


## Active contours for decomposed nerve component



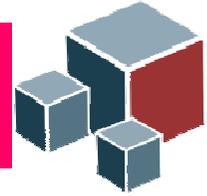
Top left: RVM2 and DCA decomposition; Bottom left: DCA decomposition only.

Top right: RVM3 and NMU decomposition; Bottom right: NMU decomposition only.



# Sparseness constrained NMF for 3D decomposition of multichannel medical images

I. Kopriva, A. Jukić, X. Chen, "Sparseness Constrained Nonnegative Matrix Factorization for Unsupervised 3D Segmentation of Multichannel Images: Demonstration on Multispectral Magnetic Resonance Image of the Brain," *SPIE Medical Imaging Symposium*, Orlando, FL, February 9-14, 2013, Proc. SPIE Vol. **8669**, paper # 119.



## 4D tensor model of multi-channel multi-slice image

For 3D decomposition multi-channel and multi-slice image is represented by multilinear mixture model:

$$\underline{\mathbf{X}} \approx \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \times_4 \mathbf{A}^{(4)}$$

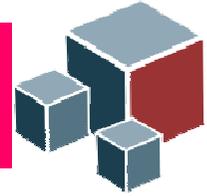
where  $\underline{\mathbf{X}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3 \times I_4}$  stands for image tensor composed of  $I_4$  channel images,  $I_3$  slices, and  $I_1 \times I_2$  pixel (voxel) elements per slice.

Above model is known as Tucker4 model, [a], where  $\underline{\mathbf{G}} \in \mathbb{R}_{0+}^{J_1 \times J_2 \times J_3 \times J_4}$  stands for core tensor and  $\{\mathbf{A}^{(n)} \in \mathbb{R}_{0+}^{I_n \times J_n}\}_{n=1}^4$  stand for factor matrices.

Factor matrices associated with first three modes represent directional basis along these modes. They can be used to model source tensor:

$$\underline{\mathbf{S}} = \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} = \underline{\mathbf{X}} \times_4 \left( \mathbf{A}^{(4)} \right)^\dagger$$

a) Tucker, L. R., "Some mathematical notes on three-mode factor analysis," *Psychometrika* 31, 279-311 (1966).



## 4D tensor model of multi-channel multi-slice image

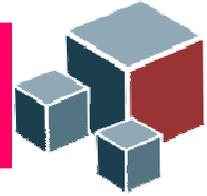
$\underline{\mathbf{S}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3 \times J}$  contains 3D intensity distributions of  $J$  organs (tissues) present in the image.

Matrix  $\mathbf{A}^{(4)}$  stands for mixing matrix that in a case of multispectral magnetic resonance image contains in its columns spectral profiles of the tissues present in the image. The image tensor  $\underline{\mathbf{X}}$  can be unfoled along mode-4 yielding:

$$\mathbf{X}_{(4)} \approx \mathbf{A}^{(4)} \mathbf{G}_{(4)} \left[ \mathbf{A}^{(3)} \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right]^T = \mathbf{A}^{(4)} \mathbf{S}_{(4)}$$

3D decomposition is performed applying sparseness constrained factorization of  $\mathbf{X}_{(4)}$ , for example using the NMU algorithm.

Afterwards,  $\mathbf{S}_{(4)}$  is tensorized to get  $\underline{\mathbf{S}}$ .

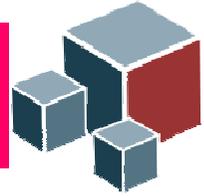


## 3D decomposition of brain tumor

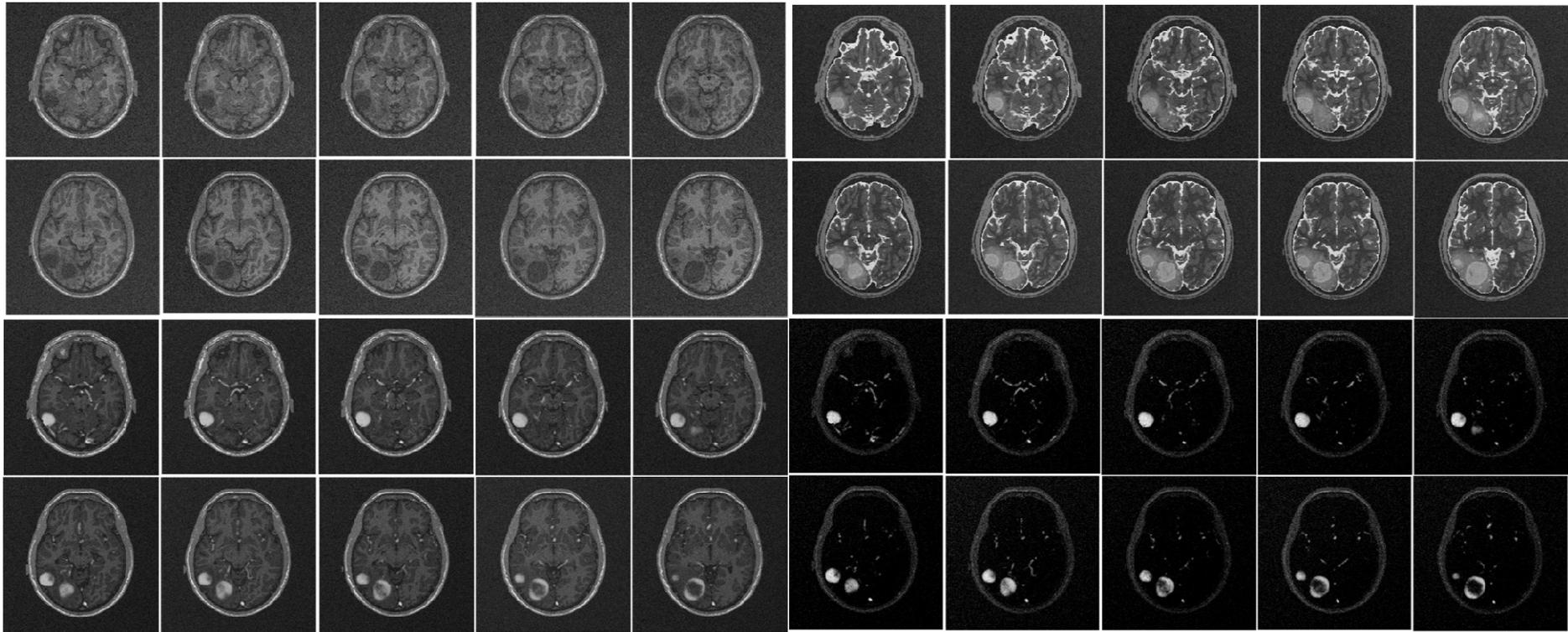
3D decomposition method is demonstrated on extraction of brain tumor from synthetic mMR image. The image is obtained from TumorSim database of the Utah Center for Neuroimage Analysis, [a].

In relation to standard mMR image comprised of T1, T2 and PD images, the PD image has been replaced by T1-weighted image obtained after administration gadolinium contrast agent.

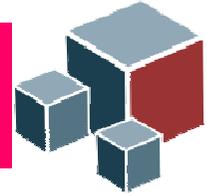
3D decomposition method is applied to slices 50 to 70 of the TumoSimData\_004 dataset. Thus,  $I_3=21$  slices were segmented jointly. Each slice has  $256 \times 256$  pixels.



## 3D decomposition of brain tumor



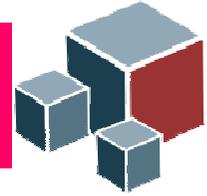
Every second slice from 52 to 70. T1 image (top left), T2 image (top right), T1\_GAD image (bottom left), NMU extracted tumor (bottom right).



## 3D decomposition of brain tumor

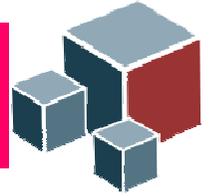
<b>Slice number</b>	<i>50</i>	<i>51</i>	<i>52</i>	<i>53</i>	<i>54</i>	<i>55</i>	<i>56</i>	<i>57</i>	<i>58</i>	<i>59</i>	<i>60</i>
<b>3D Segmentation</b>	0.532	0.575	0.627	0.637	0.647	0.639	0.600	0.514	0.457	0.423	0.367
<b>T1_GAD image</b>	0.017	0.019	0.021	0.022	0.024	0.024	0.029	0.037	0.044	0.047	0.050
<b>Slice number</b>	<i>61</i>	<i>62</i>	<i>63</i>	<i>64</i>	<i>65</i>	<i>66</i>	<i>67</i>	<i>68</i>	<i>69</i>	<i>70</i>	
<b>3D Segmentation</b>	0.427	0.451	0.473	0.480	0.439	0.343	0.359	0.305	0.224	0.218	
<b>T1_GAD image</b>	0.052	0.054	0.055	0.057	0.059	0.060	0.061	0.061	0.061	0.060	

Decomposition / segmentation results in term of Dice's coefficient for slices 50 to 70.



# Sparse component analysis approach to selection of component with disease relevant features from gene and protein expression levels

I. Kopriva, M. Filipović (2011). A mixture model with a reference-based automatic selection of components for disease classification from protein and/or gene expression levels, *BMC Bioinformatics*, vol. **12**, pp. 496 (17 pages).

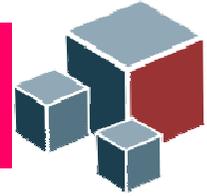


## Motivation

Disease diagnosis from protein and/or gene expression levels in contemporary proteomics and genomics is characterized by small number of samples (experiments) and large number of features (variables). This results in classical “small  $N$  large  $p$  problem” in which case classifiers and regression models are overly tuned to the training data (overfitting).

Linear mixture models often used in bioinformatics data analysis represent samples as additive mixture of components.

State-of-the-art matrix factorization methods are used to extract those components using mixture samples only.



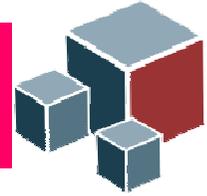
## SCA in bioinformatics

Stadtlthanner K, Theis FJ, Lang EW, Tomé AM, Puntonet CG, Górriz JM: Hybridizing Sparse Component Analysis with Genetic Algorithms for Microarray Analysis. *Neurocomputing* 2008, **71**: 2356-2376.

Sparseness constrained NMF for BSS problems with sufficiently sparse sources is applied to microarray data analysis. Sparseness constrained factorization yields components comprised of small number of genes. These, highly expressive genes are most typical for a specific cellular process:

Lee SI, Batzoglou S: Application of independent component analysis to microarrays. *Genome Biology* 2003, **4**: R76.

Thus, sparseness constraint is biologically justified.



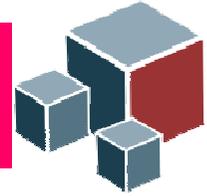
## SCA in bioinformatics

Gao Y, Church G: Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics* 2005, **21**: 3970-3975.

Sparseness constrained NMF is used to decompose set of  $N$  gene expression profiles (mixtures in BSS) into  $M$  metagenes (sources in BSS). Sparseness constrained factorization yields metagenes comprised of small number of co-expressed genes. This indicates that they can be involved in disease (cancer). Thus, sparseness constraint is biologically justified. Extracted metagenes were confirmed meaningful through subsequent biological analysis.

Kim H, Park H: Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics* 2007, **23**: 1495-1502.

Sparse NMF is formulated as  $L_1$ -constrained alternating nonnegative least square problem. Algorithm is applied to microarray datasets (leukemia, central nervous system,...). Extracted metagenes were confirmed meaningful through subsequent biological analysis.



## SCA in bioinformatics

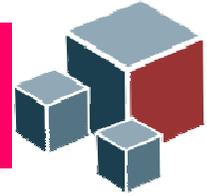
Carmona-Saez P, Pascual-Marqui RD, Tirado F, Carazo JM, Pascual-Montano A: Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics* 2006, **7**: 78.

Non-smooth NMF yields sparse factorization of microarray dataset. Extracted metagenes are shown to be biologically relevant.

In cited papers, automatic selection of extracted components to be retained for classification analysis remains an open issue.

In ref. [a] it has been proposed a novel type of additive linear mixture model of a sample that enables automatic selection of component with disease specific features on a sample-by-sample basis.

a) I. Kopriva, M. Filipović (2011). A mixture model with a reference-based automatic selection of components for disease classification from protein and/or gene expression levels, *BMC Bioinformatics*, vol. **12**, pp. 496 (17 pages).



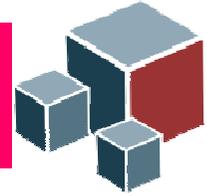
## Linear mixture model

Novel linear mixture model is comprised of actual test sample under consideration and a reference sample representing disease and/or control group. Number of additive components  $M$  is unknown and is estimated by cross-validation.

$$\begin{bmatrix} \mathbf{x}_{\text{control}} \\ \mathbf{x} \end{bmatrix} = \mathbf{A}_{\text{control}} \mathbf{S}_{\text{control}} : \mathbf{X} \in R^{2 \times T}, \mathbf{A}_{\text{control}} \in R^{2 \times M}, \mathbf{S}_{\text{control}} \in R^{M \times T} \quad M \geq 2$$

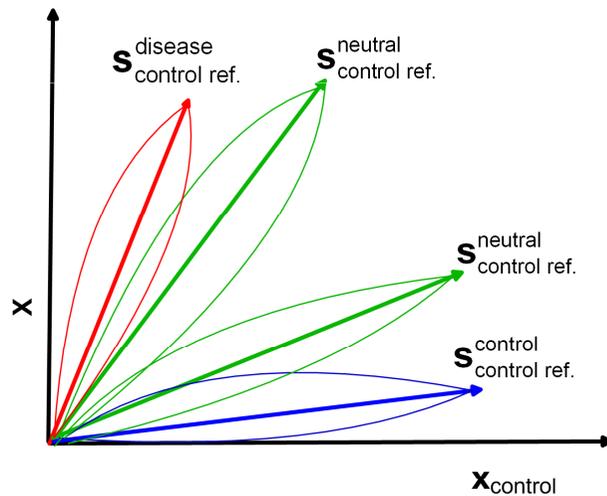
$$\begin{bmatrix} \mathbf{x}_{\text{disease}} \\ \mathbf{x} \end{bmatrix} = \mathbf{A}_{\text{disease}} \mathbf{S}_{\text{disease}} : \mathbf{X} \in R^{2 \times T}, \mathbf{A}_{\text{disease}} \in R^{2 \times M}, \mathbf{S}_{\text{disease}} \in R^{M \times T} \quad M \geq 2$$

It is assumed that disease specific features are present in prevailing concentration in disease samples. Likewise, control specific features are assumed to be present in prevailing concentration in control samples. Indifferent features are assumed to be present in similar concentration in both groups of samples.

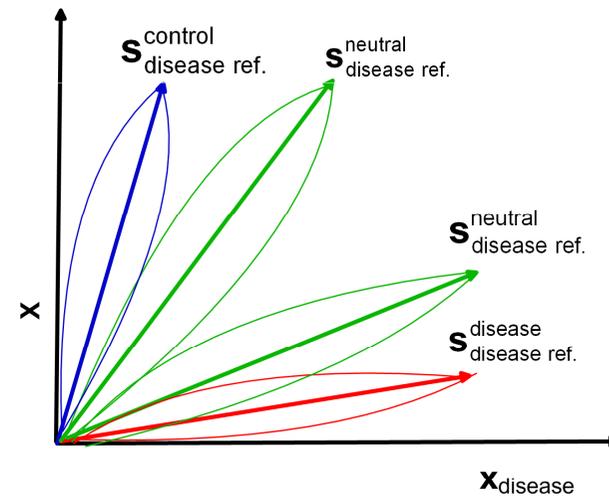


## Linear mixture model

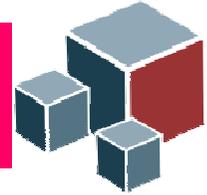
Components with disease and/or control specific features are selected automatically in mixing angles domain by exploiting geometry of linear mixture model.



Geometry of linear mixture model  
with control reference

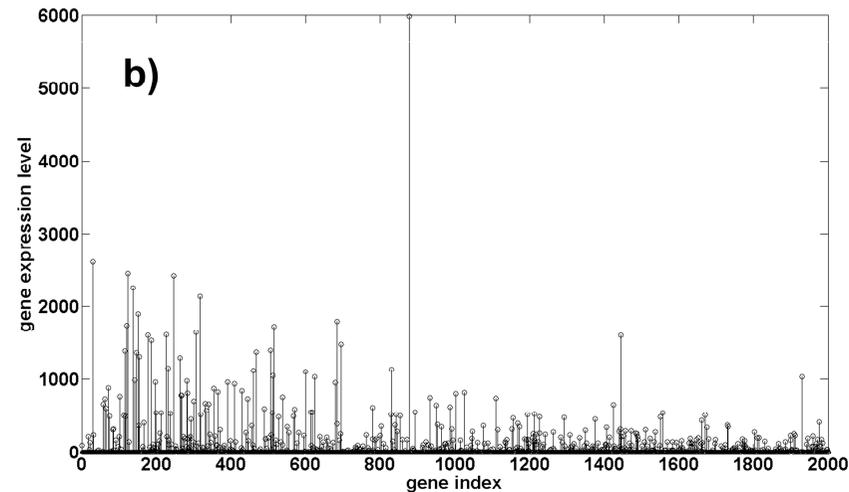
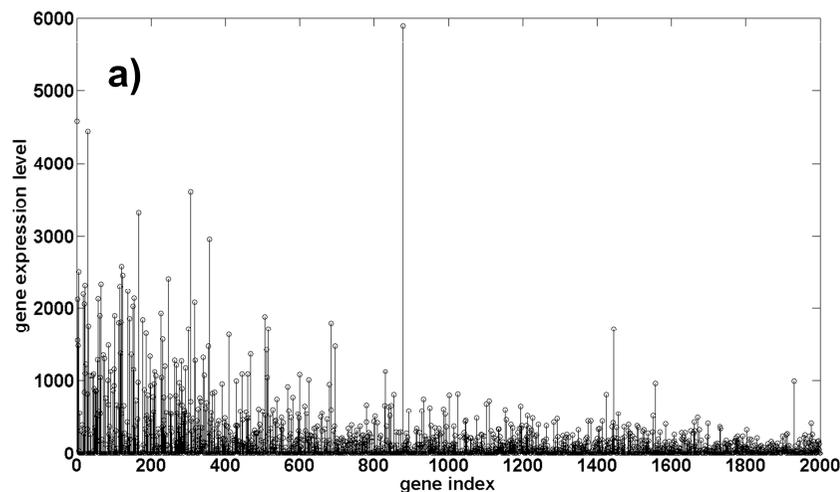


Geometry of linear mixture model  
with disease reference



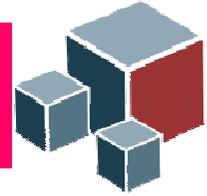
## Linear mixture model

Postulating greater number of components  $M$  will decrease complexity of component comprised of disease specific features. That is because more features that are not strongly expressed across the sample population will be picked-up into neutral components (comprised of indifferent features).



Component containing up-regulated genes extracted from a cancerous sample w.r.t. to a control reference sample:  
 a) assumed number of components  $M=2$ ; b) assumed number of components  $M=4$ .

Colon cancer dataset comprised of 40 cancerous and 22 control samples with 2000 genes available at:  
<http://genomics-pubs.princeton.edu/oncology/affydata/index.html>

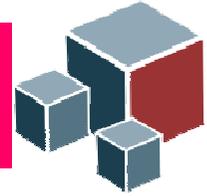


## Linear mixture model

For postulated number of components  $M > 2$  blind source separation problem implied by linear mixture models is underdetermined. To ensure unique factorization *sparseness constraint* has to be imposed on sources.

This means that *two out of  $M$*  components can be dominantly present at each feature point ( $m/z$  ratio or gene), i.e. presence in small concentrations of other components is treated as error or modeling noise.

Since decomposition is run *locally* (on a sample-by-sample basis) it further means that the same feature can be placed at different combination of components at different samples: one disease specific and one neutral; or two neutral; or one neutral and one control specific.



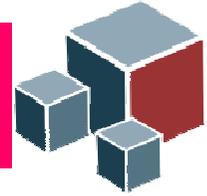
## Linear mixture model

Dataset comprised of  $N$  labeled samples is decomposed into four sets of components comprised of disease and control specific (up- and down regulated) features.

$$\left\{ \mathbf{s}_{\text{control ref.};n}^{\text{disease}}, y_n \right\}_{n=1}^N \quad \left\{ \mathbf{s}_{\text{control ref.};n}^{\text{control}}, y_n \right\}_{n=1}^N \quad \left\{ \mathbf{s}_{\text{disease ref.};n}^{\text{control}}, y_n \right\}_{n=1}^N \quad \left\{ \mathbf{s}_{\text{disease ref.};n}^{\text{disease}}, y_n \right\}_{n=1}^N \quad \left\{ y_n \in \{-1, 1\} \right\}_{n=1}^N$$

One or more classifiers can be trained on these sets and the one with the highest accuracy achieved through cross-validation can be retained for disease prediction.

Moreover, component with disease specific features can also be retained for further biomarker related analysis. Unlike standard BSS methods (ICA, NMF) that need the whole dataset to obtain such component proposed method can extract it from one sample only.



## Sparse component analysis algorithm

Mixing matrix  $\mathbf{A}$  had been estimated by clustering a set of feature points where only one component is dominantly present.

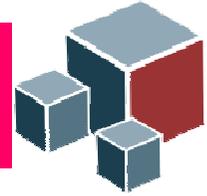
Matrix of components is obtained by solving sparseness constrained system of equations:

$$\hat{\mathbf{S}}_{\text{control}} = \min_{\mathbf{S}} \left\{ \frac{1}{2} \left\| \hat{\mathbf{A}}_{\text{control}} \mathbf{S} - \begin{bmatrix} \mathbf{x}_{\text{control}} \\ \mathbf{x} \end{bmatrix} \right\|_F^2 + \lambda \|\mathbf{S}\|_1 \right\}$$

$$\hat{\mathbf{S}}_{\text{disease}} = \min_{\mathbf{S}} \left\{ \frac{1}{2} \left\| \hat{\mathbf{A}}_{\text{disease}} \mathbf{S} - \begin{bmatrix} \mathbf{x}_{\text{disease}} \\ \mathbf{x} \end{bmatrix} \right\|_F^2 + \lambda \|\mathbf{S}\|_1 \right\}$$

Above optimization problems are solvable by the LASSO type of algorithms:

Tibshirani R: Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.* 1996, **58**: 267-288.



## Sparse component analysis algorithm

For component extraction from baseline corrected mass spectra (they can be negative) we have used iterative shrinkage thresholding (IST) type of method:

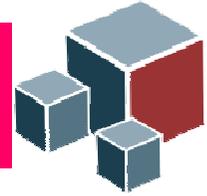
Beck A, Teboulle M: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. on Imag. Sci.* 2009, **2**: 183-202.

with a MATLAB code available at:

<http://ie.technion.ac.il/Home/Users/becka.html>

The method can be easily implemented in batch mode to solve all the  $T$  equations simultaneously.

The method also shrinks to zero small nonzero elements of  $\mathbf{S}$  that are influenced by noise.



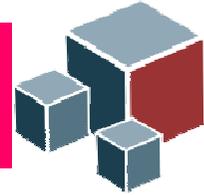
## Sparse component analysis algorithm

For component extraction from gene expression levels non-negativity constraint is imposed on  $\mathbf{S}$ :  $\mathbf{S} \geq \mathbf{0}$ .

In this case optimization problems become quadratic programs. Hence gradient descent with projection onto non-negative orthant:  $\max(\mathbf{0}, \mathbf{S})$ .

Regularization parameter  $\lambda$  is chosen by cross-validation as  $\lambda = c \times \lambda_{\max}$  where  $\lambda_{\max}$  denotes value at which  $\mathbf{S}$  is maximally sparse i.e.  $\mathbf{S} = \mathbf{0}$ .

Likewise, angular displacement  $\Delta\theta$  used in selection of single component points (used for estimation of  $\mathbf{A}$ ) is estimated by cross-validation as well.



# Sparse component analysis algorithm - outline

**Table 1 A mixture model with a reference-based algorithm for feature extraction/component selection**

Inputs.  $\{x_n \in \mathbb{R}^k, y_n \in \{1, -1\}\}_{n=1}^N$  samples and sample labels, where  $K$  represents number of feature points ( $m/z$  ratios or genes).  
 $x_{\text{control}} \in \mathbb{R}^K$  and  $x_{\text{disease}} \in \mathbb{R}^K$  representing control and disease (case) groups of samples.

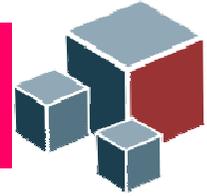
Nested two-fold cross-validation. Parameters: single component points (SCPs) selection threshold in radian equivalents of  $\Delta \theta \{1^\circ, 3^\circ, 5^\circ\}$ ;  
 regularization constant  $\lambda \{10^{-2}\lambda_{\text{max}}, 10^{-4}\lambda_{\text{max}}, 10^{-6}\lambda_{\text{max}}\}$ ; number of components  $M \{2, 3, 4, 5\}$ ; parameters of selected classifier.

Components selection from mixture samples.

1.  $\forall x \in \{x_n \in \mathbb{R}^k\}_{n=1}^N$  form a linear mixture models (LMMs) (2a) and (2b).
2. For LMMs (2a)/(2b) select a set of single component points for a given  $\Delta \theta$ .
3. On sets of SCPs use hierarchical clustering (other clustering methods can be used also) to estimate mixing matrices  $A_{\text{control}}$  and  $A_{\text{disease}}$  for a given  $M$ .
4. Estimate source matrices  $S_{\text{control}}$  and  $S_{\text{disease}}$  by solving (3a) and (3b) respectively for a given regularization parameter  $\lambda$ .
5. Use minimal and maximal mixing angles estimated from mixing matrices  $A_{\text{control}}$  and  $A_{\text{disease}}$  to select, following the logic illustrated in Fig. 2a and Fig. 2b, disease and control specific components:  $s_{\text{control ref.};n'}^{\text{disease}}$ ,  $s_{\text{control ref.};n'}^{\text{control}}$ ,  $s_{\text{disease ref.};n}^{\text{control}}$  and  $s_{\text{disease ref.};n}^{\text{disease}}$ .

End of component selection.

End of nested two-fold cross-validation.



## Ovarian cancer prediction

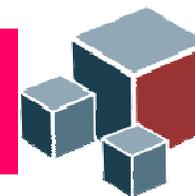
Low resolution surface-enhanced laser desorption ionization time-of-flight (SELDI-TOF) mass spectra of 100 control and 100 case samples have been used:

Petricoin EF, Ardekani AM, Hitt BA, Levine PJ, Fusaro VA, Steinberg SM, Mills GB, Simone C, Fishman DA, Kohn EC, Liotta LA: Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet* 2002, 359: 572-577.

See also the website of the National Cancer Institute (NCI) program in clinical proteomics:

<http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp>

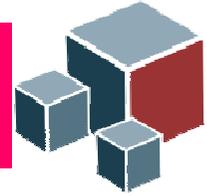
Sensitivities and specificities are estimated for linear and nonlinear (RBF and poly) SVM classifiers by 100 two-fold cross-validations.



# Ovarian cancer prediction

**Table 2 Comparative performance results in ovarian cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).**

Method	Sensitivity/Specificity/Accuracy
Proposed method $M = 3, \Delta\theta = 5^\circ$ $\lambda = 10^{-4}\lambda_{\max}$ Linear SVM	Sensitivity: 96.2 (2.7)%; specificity: 93.6 (4.1)%; accuracy: 94.9% Control specific component extracted with respect to a cancer reference sample.
Proposed method $M = 4, \Delta\theta = 3^\circ$ $\lambda = 10^{-6}\lambda_{\max}$ Linear SVM	Sensitivity: 95.4 (3)%; specificity: 94 (3.7)%; accuracy: 94.7% Control specific component extracted with respect to a cancer reference sample.
[1]	Sensitivity: 81.4 (7.1)%; specificity: 71.7 (6.6)%
[42]	Sensitivity: 100%; specificity: 95% ( <u>one partition only</u> ; 50/50 training; 66/50 test).
[44]	Accuracy averaged over 10 ten-fold partitions: 98-99% (sd: 0.3-0.8)
[13]	Sensitivity: 98%, specificity: 95%, two-fold CV with 100 partitions.
[45]	Average error rate of 4.1% with three-fold CV.



## Prostate cancer prediction

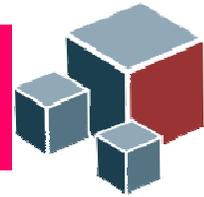
Low resolution SELDI-TOF mass spectra of 63 control samples: no evidence of cancer with prostate-specific antigen (PSA) $<1$ , and 69 prostate cancer samples : 26 with  $4 < \text{PSA} < 10$  and 43 with  $\text{PSA} > 10$ , have been used:

Petricoin EF, Ornstein DK, Paweletz CP, Ardekani A, Hackett PS, Hitt BA, Velasco A, Trucco C, Wiegand L, Wood K, Simone CB, Levine PJ, Linehan WM, Emmert-Buck MR, Steinberg SM, Kohn EC, Liotta LA: Serum proteomic patterns for detection of prostate cancer. *J. Natl. Canc. Institute* 2002, 94: 1576-1578.

See also the website of the National Cancer Institute (NCI) program in clinical proteomics:

<http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp>

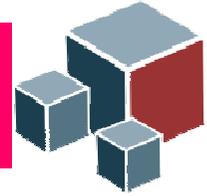
Sensitivities and specificities are estimated for linear and nonlinear (RBF and poly) SVM classifiers by 100 two-fold cross-validations.



# Prostate cancer prediction

**Table 3 Comparative performance results in prostate cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).**

Methods	Sensitivity/Specificity/Accuracy
Proposed method $M = 5$ , $\Delta\theta = 1^\circ$ $\lambda = 10^{-4}\lambda_{\max}$ Linear SVM	Sensitivity: 97.6 (2.8)%; specificity: 99 (2.2)%; accuracy: 98.3% Control specific component extracted with respect to a cancer reference sample.
Proposed method $M = 4$ , $\Delta\theta = 1^\circ$ $\lambda = 10^{-4}\lambda_{\max}$ Linear SVM	Sensitivity: 97.7 (2.3)%; specificity: 98 (2.4)%; accuracy: 97.9% Control specific component extracted with respect to a cancer reference sample.
[1]	Sensitivity: 86 (6.6)%; specificity: 67.8(12.9)%; accuracy: 76.9%.
[46]	Sensitivity: 94.7%; specificity: 75.9%; accuracy: 85.3%. 253 benign and 69 cancers. Results were obtained on independent test set comprised of 38 cancers and 228 benign samples.
[47]	Sensitivity: 97.1%; specificity: 96.8%; accuracy: 97%. 253 benign and 69 cancers. Cross-validation details not reported.
[45]	Average error rate of 28.97 on four class problem with three-fold cross-validation.



## Colon cancer prediction

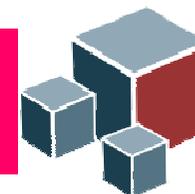
Gene expression profiles of 40 colon cancer and 20 control samples obtained by Affymetrix oligonucleotide array have been used:

Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* 1999, 96: 6745-6750.

Gene expression data originally contained 6500 genes but only 2000 high-intensity genes were retained for analysis. Data can be downloaded from:

<http://genomics-pubs.princeton.edu/oncology/affydata/index.html>

Sensitivities and specificities are estimated for linear and nonlinear (RBF and poly) SVM classifiers by 100 two-fold cross-validations.



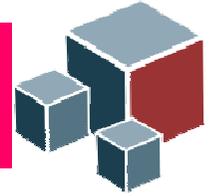
## Colon cancer prediction

**Table 4 Comparative performance results in colon cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).**

Methods	Sensitivity/Specificity/Accuracy
Proposed method $M = 2, \Delta\theta = 1^\circ$ RBF SVM ( $\sigma^2 = 1200, C = 1$ )	Sensitivity: 90.8 (5.5)%, specificity: 79.4 (9.8)%; accuracy: 85.1% Control specific component extracted with respect to a cancer reference sample.
Proposed method $M = 4, \Delta\theta = 5^\circ, \lambda = 10^{-2}\lambda_{max}$ RBF SVM ( $\sigma^2 = 1000, C = 1$ )	Sensitivity: 89.8 (6.2)%, specificity: 78.6 (12.8)%; accuracy: 84.2% Control specific component extracted with respect to a control reference sample.
[1]	Sensitivity: 89.7 (6.4)%, specificity: 84.3 (8.4)%; accuracy = 87%. 100 two-fold cross-validations.
[2]	Sensitivity: 92.1 (4.7)%, specificity: 85 (10.1)%; accuracy: 88.55%. 100 two-fold cross-validations. $c_u = 2.0$ .
[48]	Sensitivity: 92-95% calculated from Figure 5. Specificity not reported.
[15]	Accuracy 85%. Cross-validation details not reported.
[50]	Accuracy 82.5%, ten-fold cross-validation (RFE with linear SVM).
[51]	Accuracy 88.84%, two-fold cross-validation (RFE with linear SVM and optimized penalty parameter C).



2013 IEEE SPS Summer School on Biomedical Image Processing and Analysis – June 9 2013, Dubrovnik, Croatia.  
"Sparse component analysis - applications in multichannel medical imaging and bioinformatics"



---

**THANK YOU !!!!!!!!**