(Non-) linear sparse component analysis: theory and applications in medical imaging, chemo- and bioinformatics

Ivica Kopriva

Ruđer Bošković Institute

e-mail: ikopriva@irb.hr ikopriva@gmail.com
Web: http://www.lair.irb.hr/ikopriva/

Acknowledgment:

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

Rudjer Boskovich Institute, Zagreb, Croatia
https://www.irb.hr/eng

The Ruđer Bošković Institute is regarded as Croatia's leading scientific institute in the natural and biomedical sciences as well as marine and environmental research, owing to its size, scientific productivity, international reputation in research, and the quality of its scientific personnel and research facilities.

The Institute is the leading and internationally most competitive Croatian institute by virtue of its participation in international research projects, such as the IAEA and EC FP5-7 programs funded by the European Commission, NATO, NSF, SNSF, DAAD and other international scientific foundations.

Today, the Ruđer Bošković Institute has over 550 scientists and researchers in more than 80 laboratories pursuing research in theoretical and experimental physics, physics and materials chemistry, electronics, physical chemistry, organic chemistry and biochemistry, molecular biology and medicine, the sea and the environment, informational and computer sciences, laser and nuclear research and development.

Roger Joseph Boskovich

http://en.wikipedia.org/wiki/Roger_Joseph_Boscovich

Ruđer Bošković (18 May 1711 – 13 February 1787) was a physicist, astronomer, mathematician, philosopher, diplomat, poet, theologian, Jesuit priest, and a polymath from the city of Dubrovnik in the Republic of Ragusa (today Croatia), who studied and lived in Italy and France where he also published many of his works.

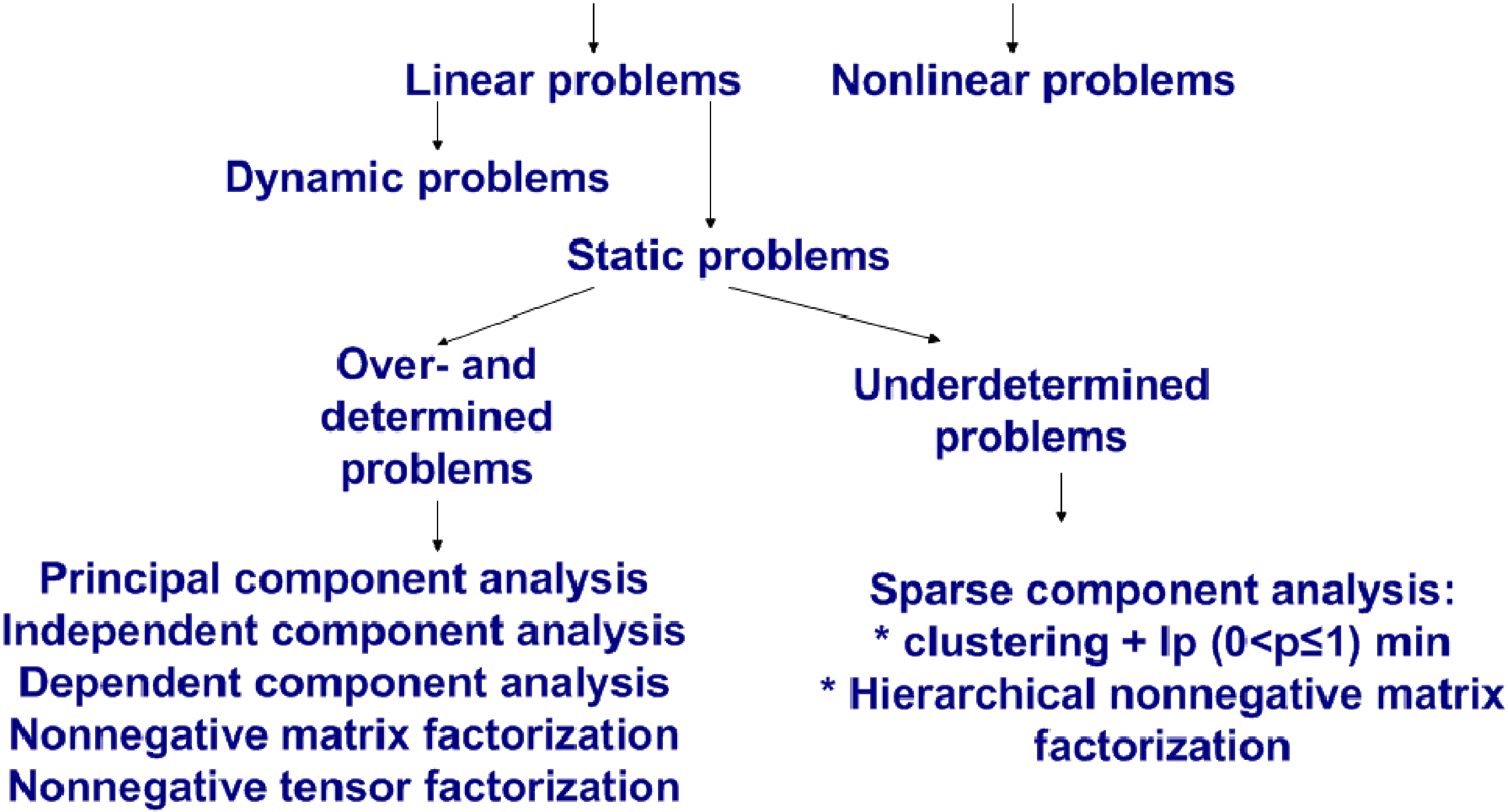
Among his many achievements he was the first to suggest least absolute deviation based regression (1757). That was studied by Laplace (1793) and predated the least square technique originally developed by Legendre (1805) and Gauss (1823):

P. Bloomfield and W. L. Steiger. *Least Absolute Deviations: Theory, Applications, and Algorithms.* Birkhauser, Boston, MA, 1983.

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)
  - SCA/NMF in reproducible kernel Hilbert spaces (RKHS)
- Applications in multispectral and magnetic resonance image decomposition, chemo- and bioinformatics

Blind separation of sources



Linear problems

Nonlinear problems

Dynamic problems

Static problems

Over- and determined problems

Underdetermined problems

Principal component analysis
Independent component analysis
Dependent component analysis
Nonnegative matrix factorization
Nonnegative tensor factorization

Sparse component analysis:
* clustering + lp (0<p≤1) min
* Hierarchical nonnegative matrix factorization

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:

$\mathbf{X}=\mathbf{AS}$ $\mathbf{X}\in R^{NxT}$, $\mathbf{A}\in R^{NxM}$, $\mathbf{S}\in R^{Mx\mathit{T}}$

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 **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, $\mathbf{T}=\mathbf{P\Lambda}$, are meaningful. For such solutions constraints must be imposed on **A** and/or **S**.

Independent component analysis (ICA) solves BSS problem provided that: source signals **S** are statistically independent and non-Gaussian; mixing matrix **A** is full column rank i.e. M≤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 **X**: $L(\mathbf{X})=\mathbf{A}L(\mathbf{S})$. ICA is applied to $L(\mathbf{X})$ to estimate **A** and $L(\mathbf{S})$. **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 **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 **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**

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.

Underdetermined BSS

- SCA-based solution of the uBSS problem is obtained in two stages:

 1) estimate basis or mixing matrix **A** using data clustering.

 2) estimating sources, with estimated **A**, one at a time $\mathbf{s}_t$, t=1,…,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≤ p ≤1.

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

- NMF-based solution yields **A** and **S** simulatneously through sparseness and nonnegativity constrained factorization of **X**.

When uBSS problems can(not) be solved?

Let us focus on undeterdermined linear system:

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

Let **s** be K-sparse i.e. $K=||\mathbf{s}||_{\mathbf{0}}$.

Provided that **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 **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 **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, $\lim_{p\to 0} N \geq C_1(0)K$. 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 ℓ^0-constraints. *Neurocomputing*. 2012; **80**: 38-46.

c) Mohimani H, Babie-Zadeh B, Jutten C. A Fast Approach for Overcomplete Sparse Decomposition Based on ℓ_0 Smoothed Norm. *IEEE Trans. Sig. Proc*. 2009; **57**(1): 289-301.

When uBSS problems can(not) be solved?

However in BSS problems **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 **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?

One result for deterministic **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 **A** it applies $N=O(K^2)$. That is significantly worse than $N \approx K\log(M/K)$ for random **A**.

K correponds 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*$_{\mathbf{p}}$ norm minimization: 0< *p* ≤1

- Signal **s** is K-sparse if it has K non-zero components, i.e. K=||**s**||$_{\mathbf{0}}$. Thereby,

$$\|\mathbf{s}\|_0 = \sum_{m=1}^{M} |s_m|^0 \quad \text{By definition}: 0^0 = 0.$$

- If uBSS problem is not sparse in original domain <u>it ought to be transformed</u> in domain where enough level of sparseness can be achieved: *T*(**x**)=**A***T*(**s**).

- Time-frequency and time-scale (wavelet) bases are employed for this purpose quite often.

- In addition to sparseness requirement on **s** certain degree of incoherence of the mixing matrix **A** is required as well. Mutual coherence is defined as the largest absolute and normalized inner product between different columns in **A**, what reads as

$$\mu \ \mathbf{A} \ = \max_{1\le i,j\le M \ \text{ and } i\ne j} \frac{\left|\mathbf{a}_i^T \mathbf{a}_j\right|}{\left\|\mathbf{a}_i\right\|\left\|\mathbf{a}_j\right\|}$$

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 **s** relates sparseness requirement on **s** and coherence of **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 **A** has unique representation property, that is if all $N\times N$ sub-matrices are full rank, the unique solution of **x**=**As** 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 **A** can not be predefined i.e. they are problem dependent. Yet, **A** dictates a level of sparseness of **s** that is necessary to obtain possibly unique solution of the uBSS problem: **x**=**As**. To obtain such solution it is necessary to:

- estimate **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 **x=As** amounts to solving:

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

or for problems with noise or approximation error:

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

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

Direct minimization of L_0–norm of **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:

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

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

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

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

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

a) S..J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing* **1**, 606-617 (2007), **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. Figuiredo, 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 **s**(*t*) is Laplacian, maximum likelihood approach to maximization of posterior probability *P*(**s**|**x**,**A**) yields minimum L_1-norm as the solution:

$$\hat{\mathbf{s}}(t) = \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\ \mathbf{s}(t)\big|\mathbf{x}(t),\hat{\mathbf{A}}$$

$$= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\ \mathbf{x}(t)\big|\mathbf{s}(t),\hat{\mathbf{A}}\ \ P\ \mathbf{s}(t)$$

$$\propto \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\ \mathbf{s}(t)$$

$$= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \exp - \ \left|\mathbf{s}_1(t)\right| + \ldots + \left|\mathbf{s}_M(t)\right|$$

$$= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left|\mathbf{s}_1(t)\right| + \ldots + \left|\mathbf{s}_M(t)\right|$$

$$= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left\|\mathbf{s}(t)\right\|_1$$

uBSS – L_1 norm minimization

Sequence of MATLAB commands for solution of the problem **x**=**As** using command `linprog`:

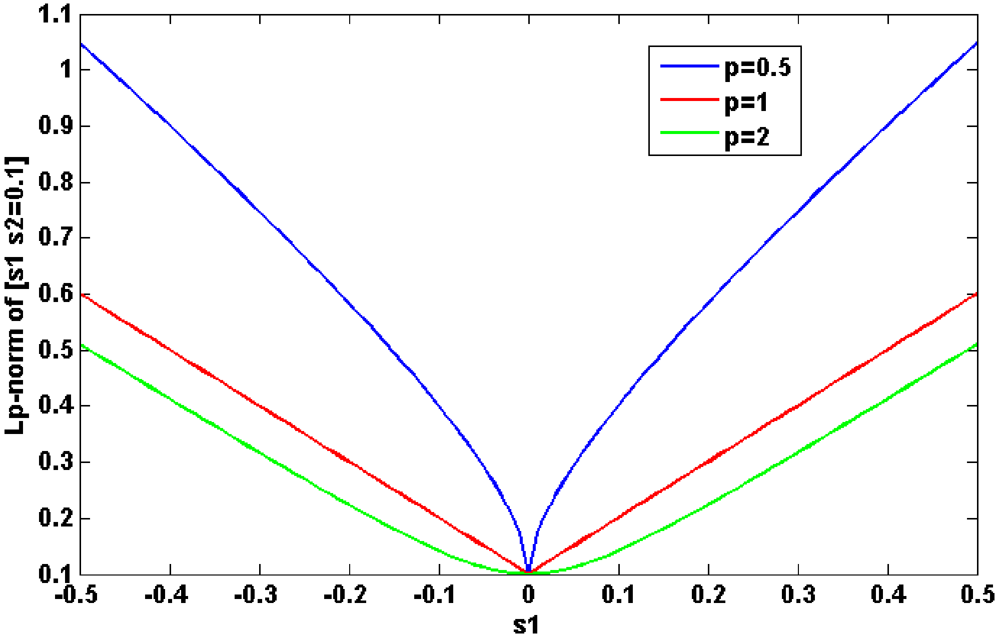
```
% Linear progamming solution
% solves linear program min(x) f'*x s.t. Ax=b, lb<=x<=ub.
f = ones(M,1);
lb = zeros(M,1);
ub = 1000*ones(M,1);

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 Lalpacian? 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 ≤1

Minimizing L_p-norm, 0<p<1, of **s** yields better performance when solving underdetermined system **x**=**As** 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-norm of [s_1 0.1] : $\|\mathbf{s}\|_p = \left(\sum_{m=1}^{M} |s_m|^p \right)^{1/p}$

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

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

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

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 σ brings us closer to $L_0(\mathbf{s})$, while larger σ 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 σ and than use obtained solution as initial value for next maximization of $F_\sigma(\mathbf{s})$ for smaller σ.

After each iteration computed approximation of **s** is projected back onto the constraining set **As**=**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

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

```
**Initialize**: ε=1, $\mathbf{s}^{(0)}$ = pinv(A)x, k=1.
**do**
    **repeat**
        $w_m = \quad s_m^{(k-1)} \quad {}^{2} + \varepsilon \quad {}^{p/2-1}$
        $\mathbf{Q}_k = diag \ \ 1/w_m$
        $\mathbf{s}^{(k)} = \mathbf{Q}_k\mathbf{A}^T \ \ \mathbf{AQ}_k\mathbf{A}^T \ \ {}^{-1}\mathbf{x}$
        $k = k+1$
    **until** $\left\|\mathbf{s}^{(k)} - \mathbf{s}^{(k-1)}\right\|_2 < \sqrt{\varepsilon}/100$
    ε=ε/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:

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

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

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

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

where λ=σ^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:

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

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

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

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

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) T. Blumensath, M.E. Davies, Iterative Thresholding for Sparse Approximations, *J Fourier Anal Appl* (2008) **14**: 629-654.

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, SNR [dB] $\in\{10, 20, 30, \text{inf}\}$, 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×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:

$$\varepsilon \;\; \mathbf{S} \;\; = 10\log_{10} \;\; \left\| diag \;\; \mathbf{R}_{\mathrm{ss}} \;\; \right\|_2^2 \Big/ \left\| diag \;\; \mathbf{R}_{\mathrm{ss}} - \mathbf{R}_{\mathrm{s}\hat{\mathrm{s}}} \;\; \right\|_2^2$$

has been averaged after 100 runs, where $\mathbf{R}_{\mathbf{ss}}$ and $\mathbf{R}_{\mathrm{s}\hat{\mathrm{s}}}$ 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 were noise is significant:

M. Hayder, K. Mahata, "An Improved Smoothed Approximation Algorithm for Sparse Representation," *IEEE Trans. Sig. Proc*., vol. **5**8, No. 4, pp. 2194-2205, 2010.

The ISLO 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. Rezaiifar, 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. Sc*i., 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):

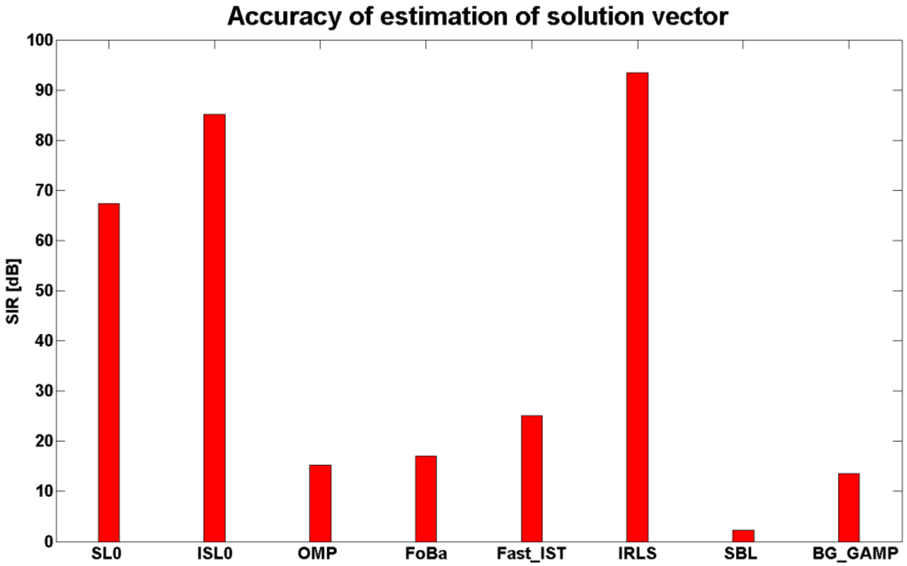
J. Vila, P. Schniter, “Expectation-maximization Bernoulli-Gaussian approximate message passing”, *Proc. Asilomar Conf. on Signals, Systems, and Computers* (Pacific Grove, CA), Nov. 2011.

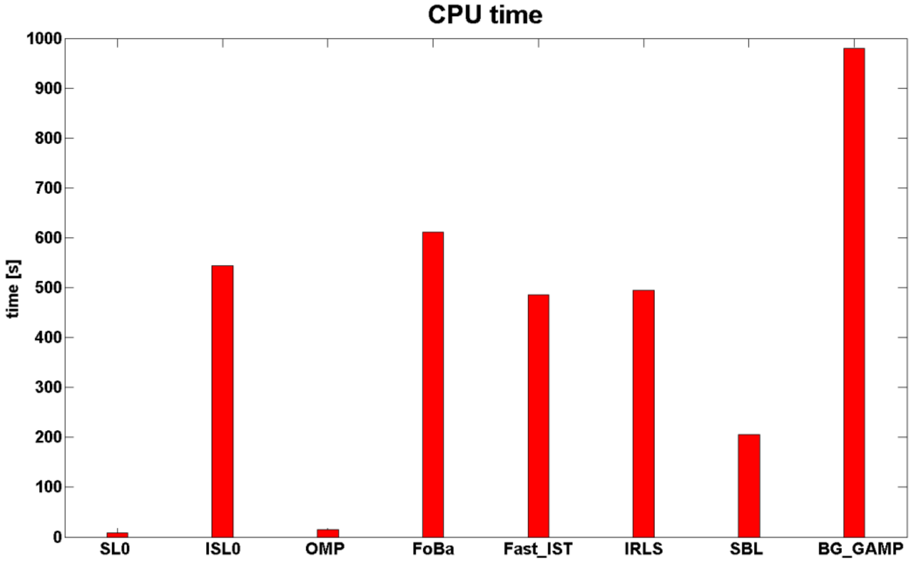
Source signal is modeled as Bernoulli-Gaussian with unknown sparisty, mean and variance, and noise is modeled as Gaussian with unknown variance. Hence, the BG-GAMP algorithm. Expectation maximization is used to learn the signal and noise parameters as algorithm iterates.

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

Comparative performance analysis

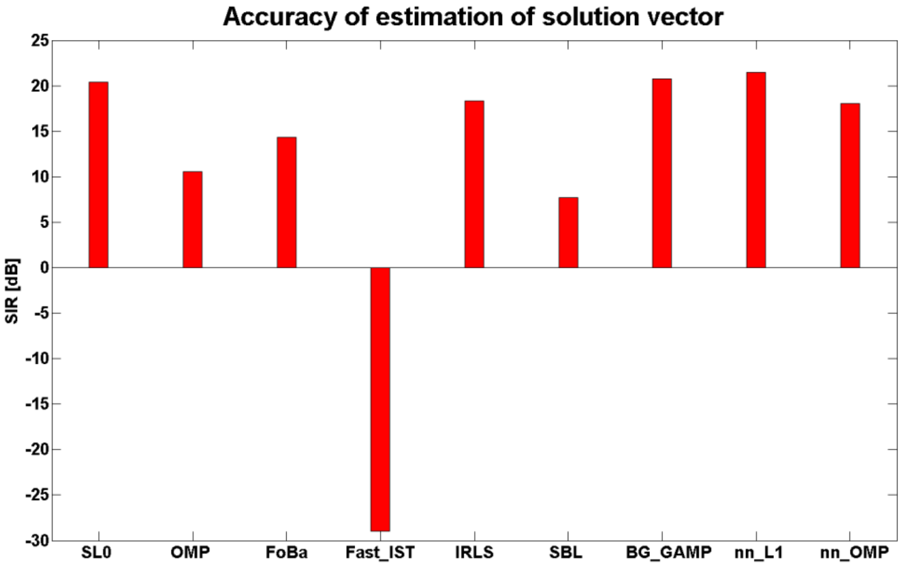
SNR=inf

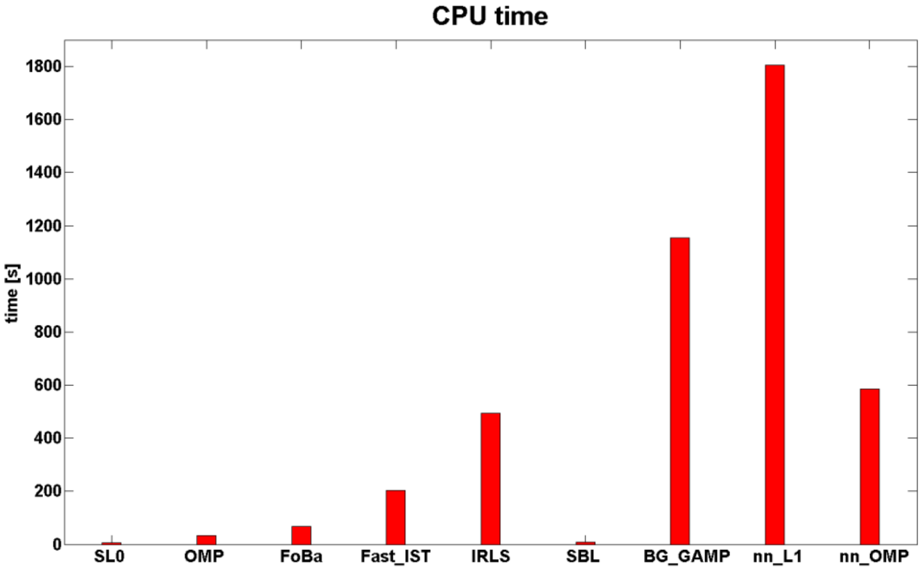




Comparative performance analysis

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 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 dominantely 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:

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

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 π radians.

At SSPs mixture vectors $\mathbf{x}_t$ and 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 **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 $\boldsymbol{a}_\mathbf{m}$ and N=2 we can parameterize column vectors in a plane by one angle

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

Assuming that **s** is *1*-sparse in representation domain estimation of **A** and *M* is obtained by means of data clustering algorithm.

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

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

Estimation of the mixing matrix: clustering

Calculate function *f*(**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 σ is called dispersion. If σ is set to sufficiently small value the value of the function *f*(**a**) will approximately equal the number of data points close to **a**. Thus by varying mixing angle φ we effectively cluster data.

- Number of peaks of the function *f*(**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 hypespectral 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 encompases 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

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

Unlike may 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, 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.

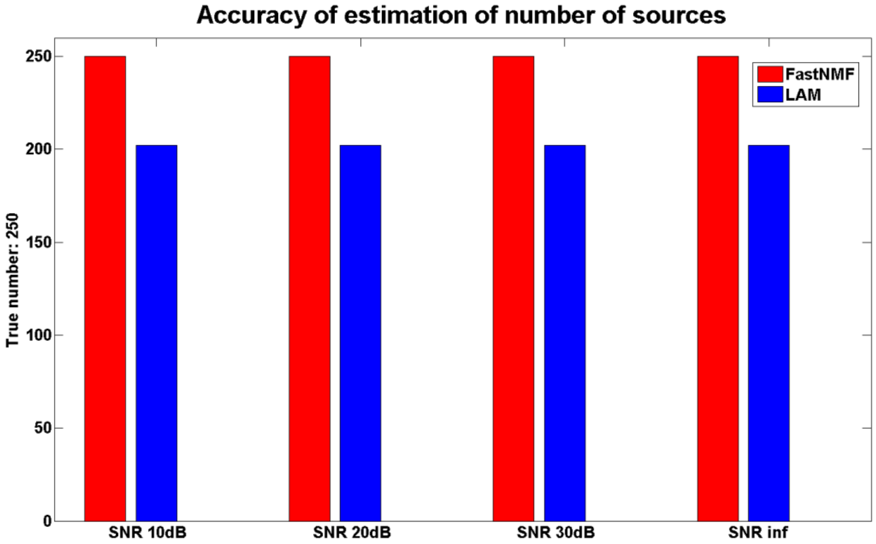
Comparative performance analysis

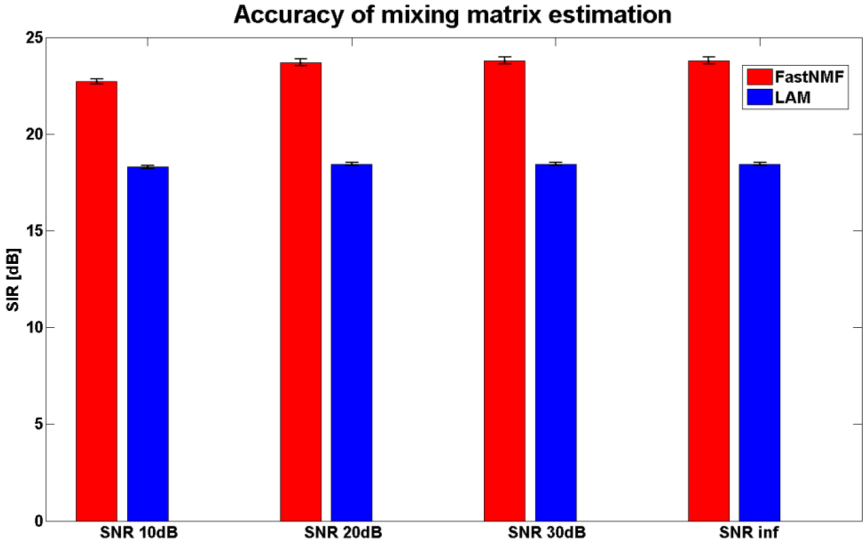
Performance measure:

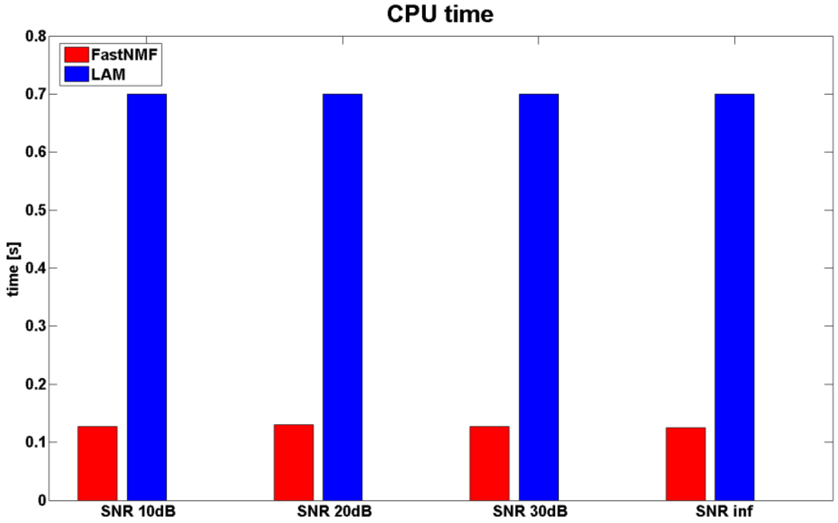
$$\varepsilon\left(\mathbf{A}\right)=10\log_{10}\left(\left\|diag\left(\mathbf{R}_{\mathbf{AA}}\right)\right\|_2^2\Big/\left\|diag\left(\mathbf{R}_{\mathbf{AA}}-\mathbf{R}_{\mathbf{A\hat{A}}}\right)\right\|_2^2\right)$$

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

Comparative performance analysis

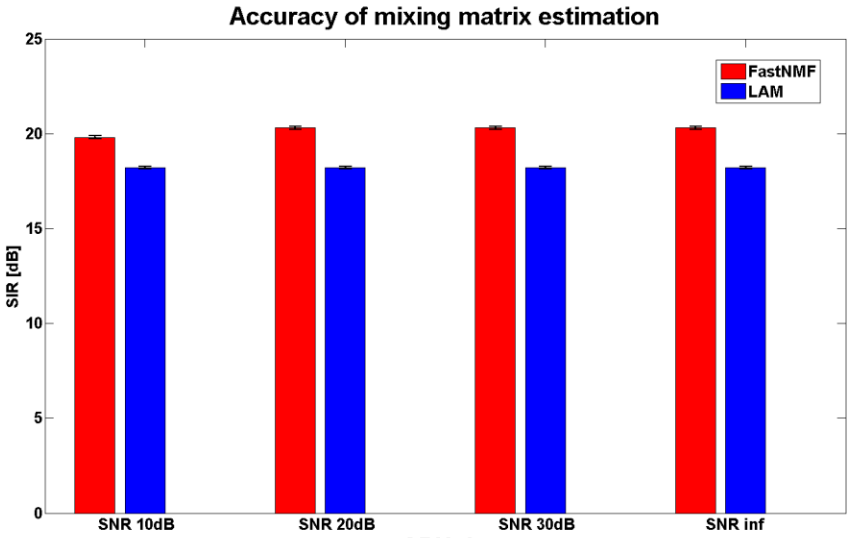

Accuracy of estimation of number of sources

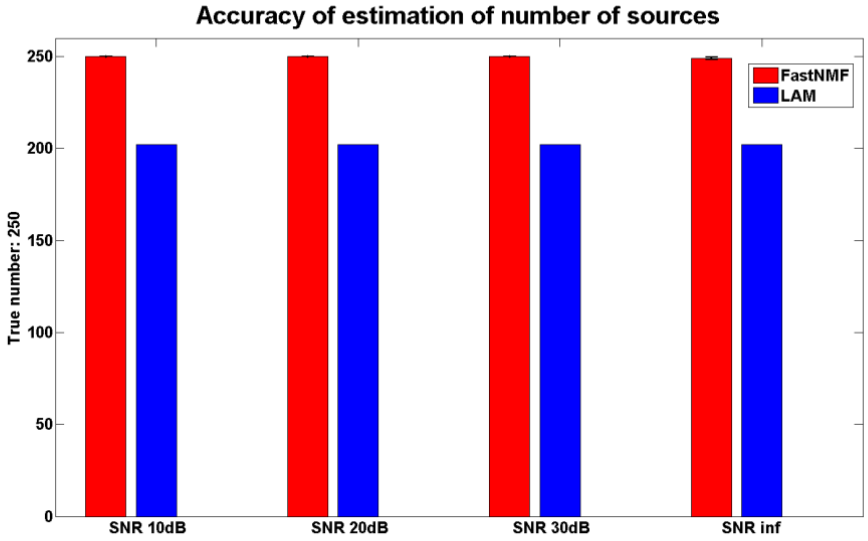

Accuracy of mixing matrix estimation

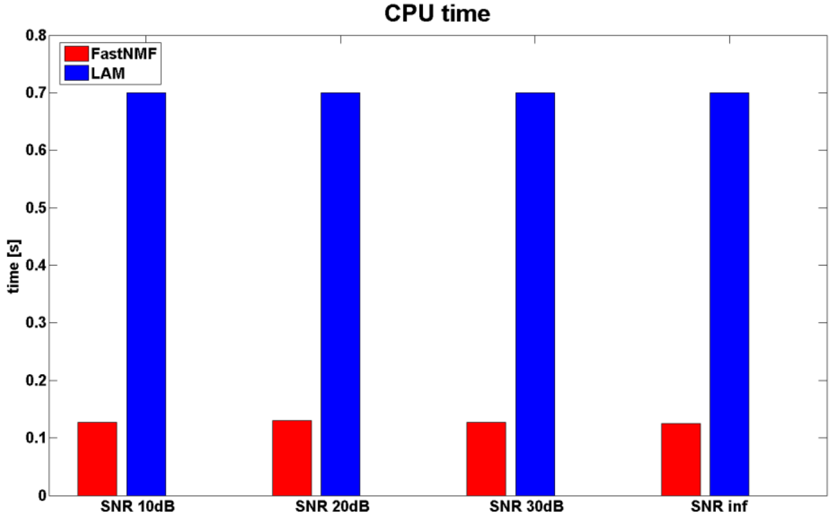

CPU time

K=10

Comparative performance analysis

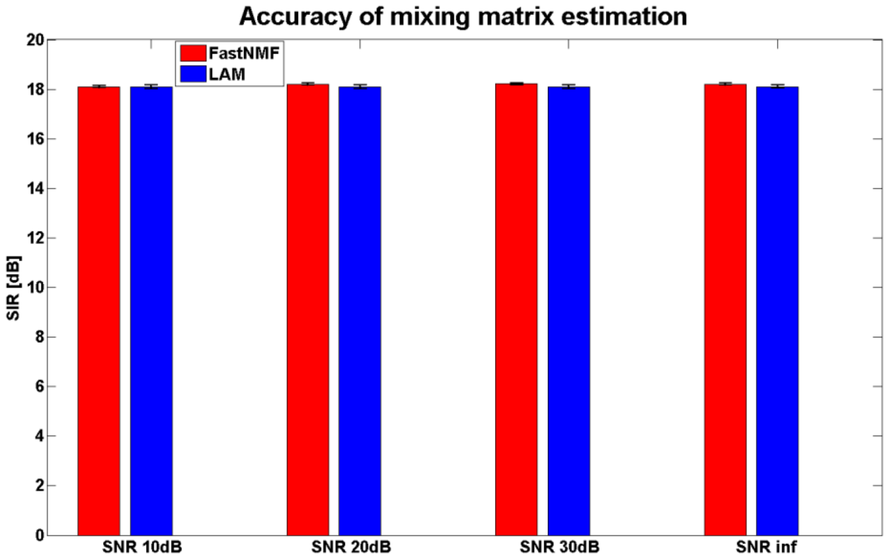


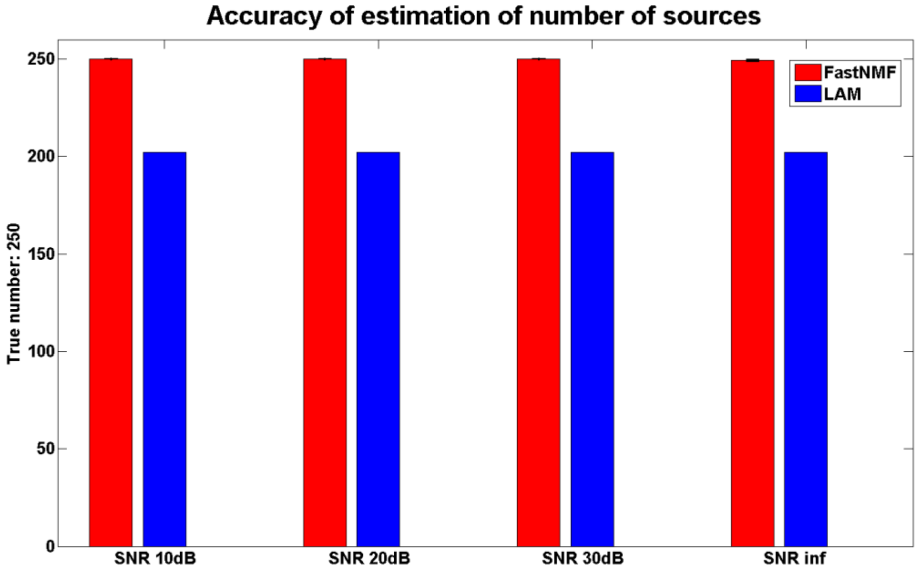


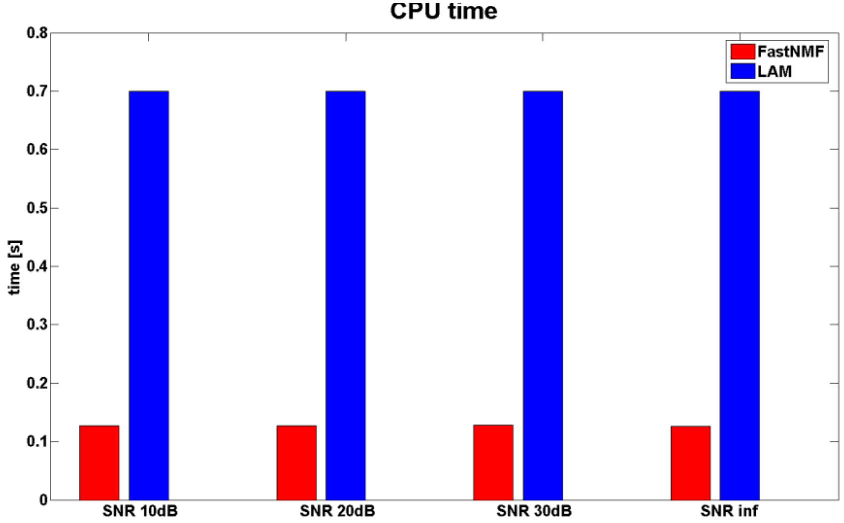


K=30

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 severly violated.

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

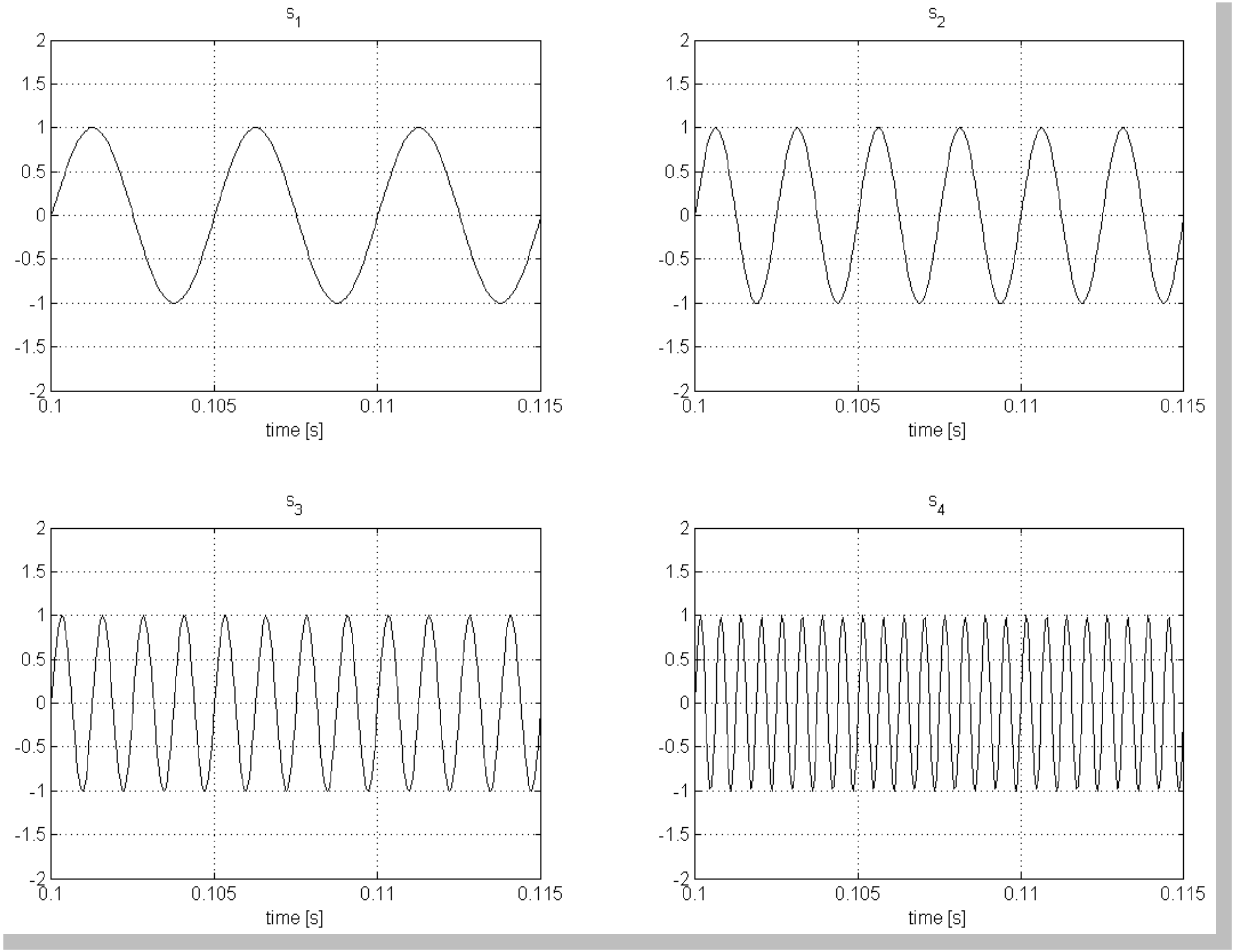
a) number of sources is greater than number of mixtures.
b) pure pixel assumption is violated significantly.

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.

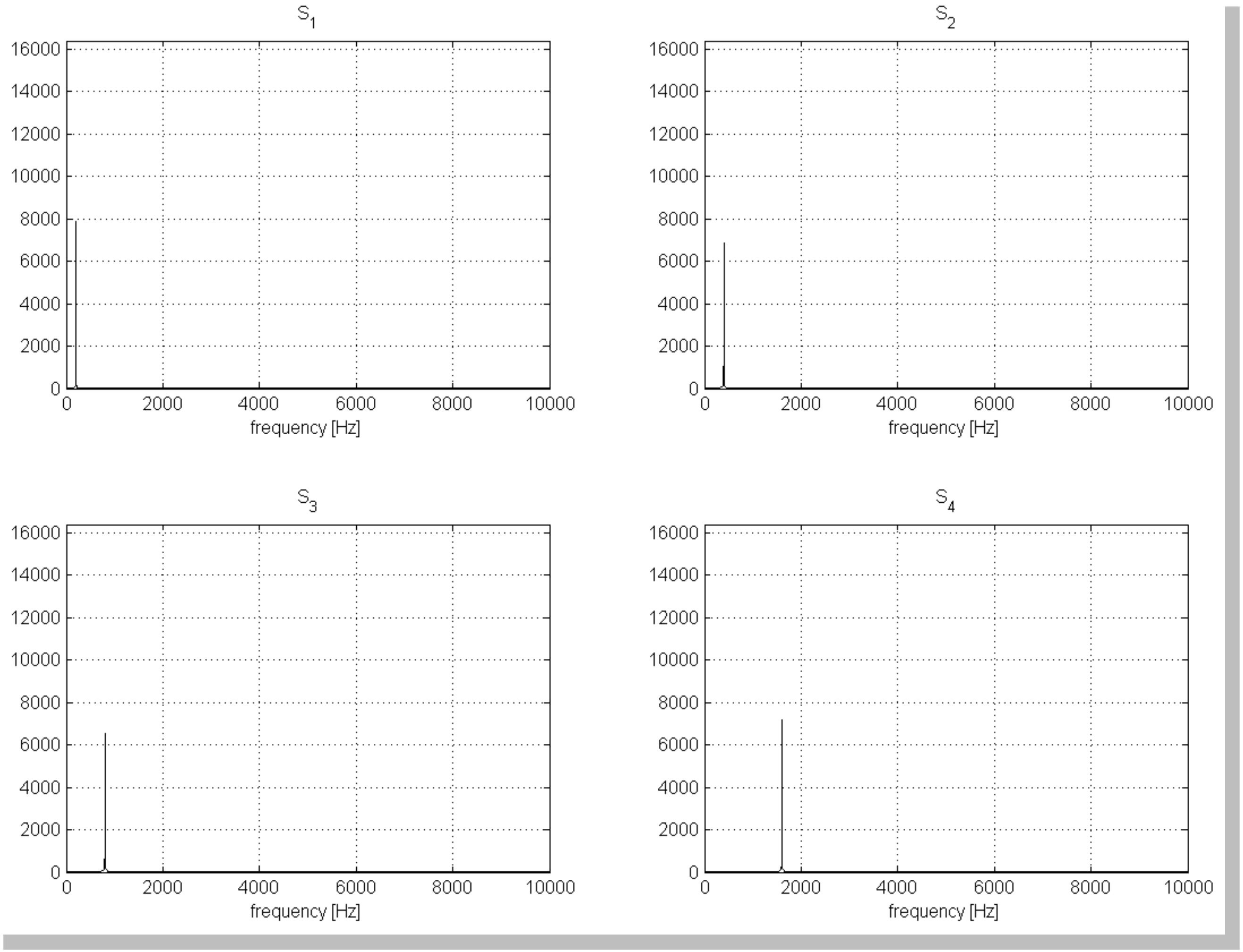
TIME DOMAIN



Blind separation of four sine signals from two mixtures

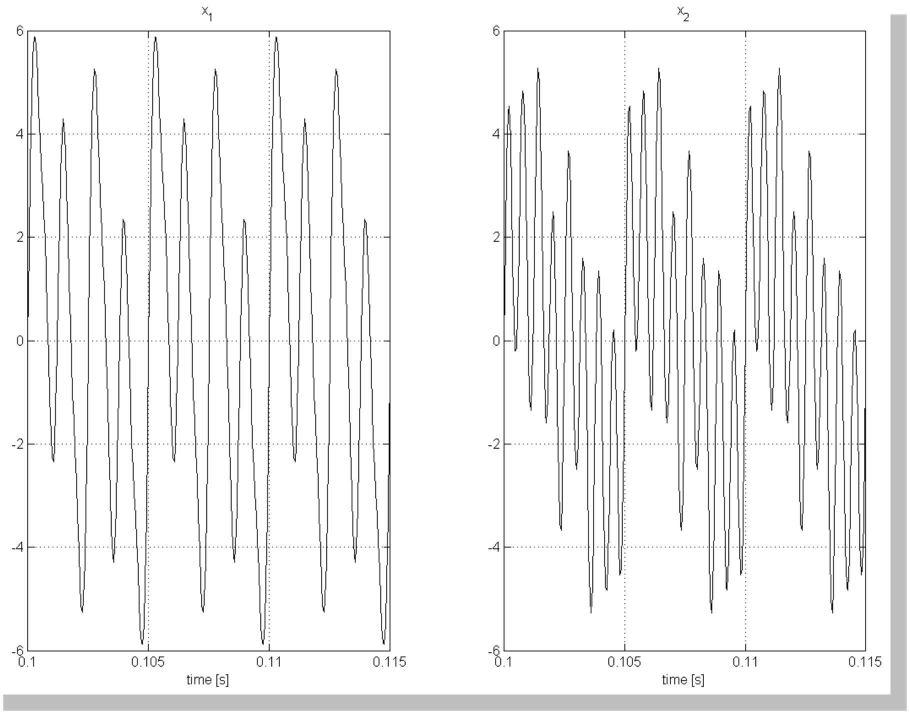
Four sinusoidal signals with frequencies 200Hz, 400Hz, 800Hz and 1600Hz.

FREQUENCY DOMAIN

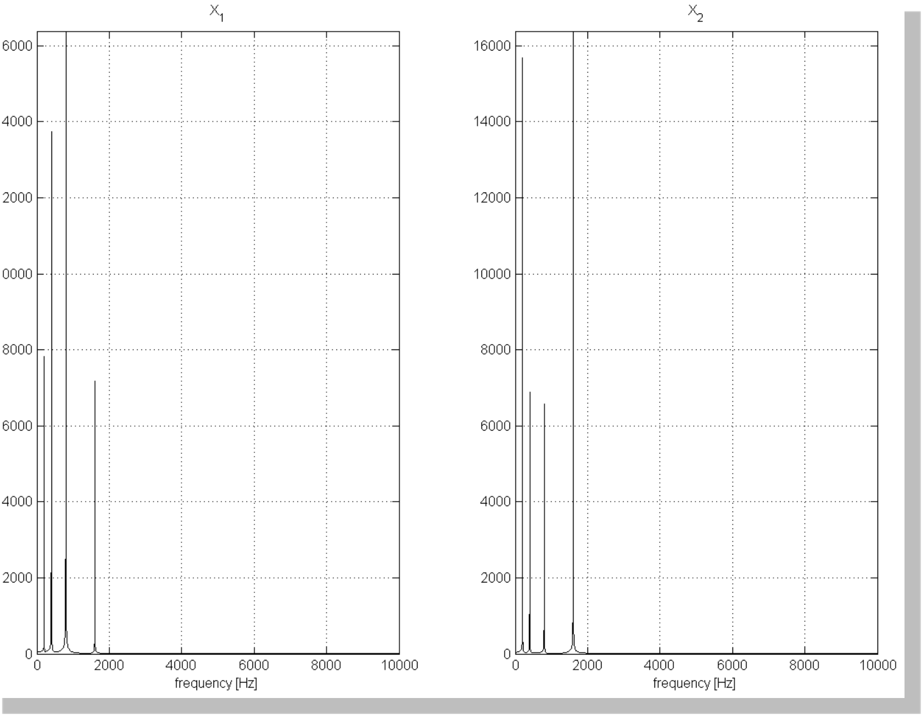


Blind separation of four sine signals from two mixtures

Two mixed signals



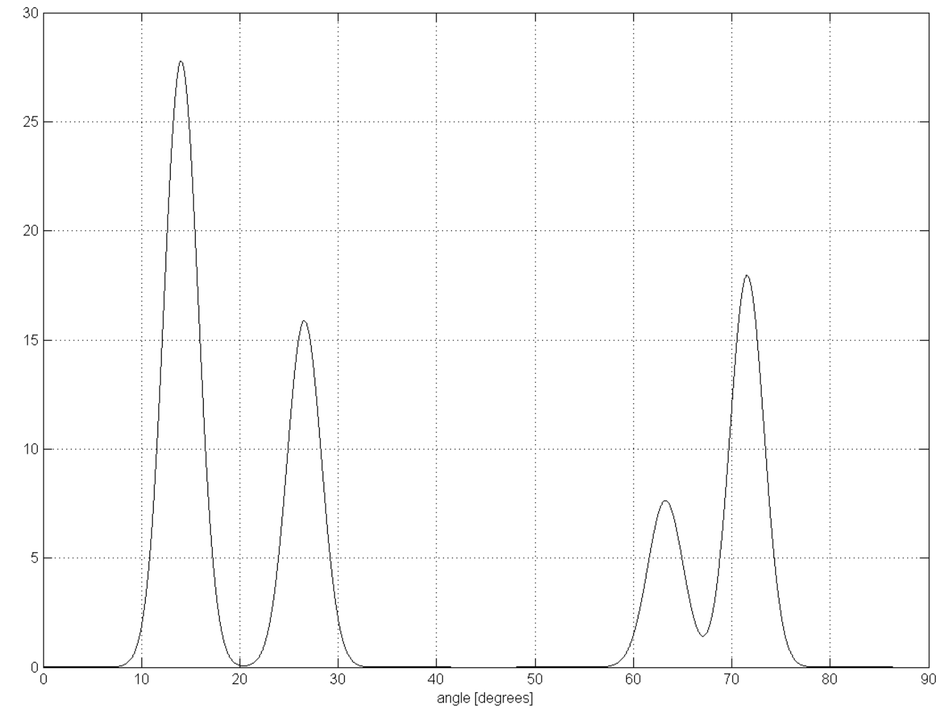
TIME DOMAIN



FREQUENCY DOMAIN

Blind separation of four sine signals from two mixtures

Clustering function



A=[63.44^0 26.57^0 14.04^0 71.57^0]

AH=[14.03^0 26.55^0 63.26^0 71.55^0]

Blind separation of four sine signals from two mixtures

Linear programming based estimation of the sources based on estimated mixing matrix **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:

$$\bar{\mathbf{x}} \quad \omega \quad = \bar{\mathbf{A}}\bar{\mathbf{s}} \quad \omega$$

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

Blind separation of four sine signals from two mixtures

Introducing:

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

yields:

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

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

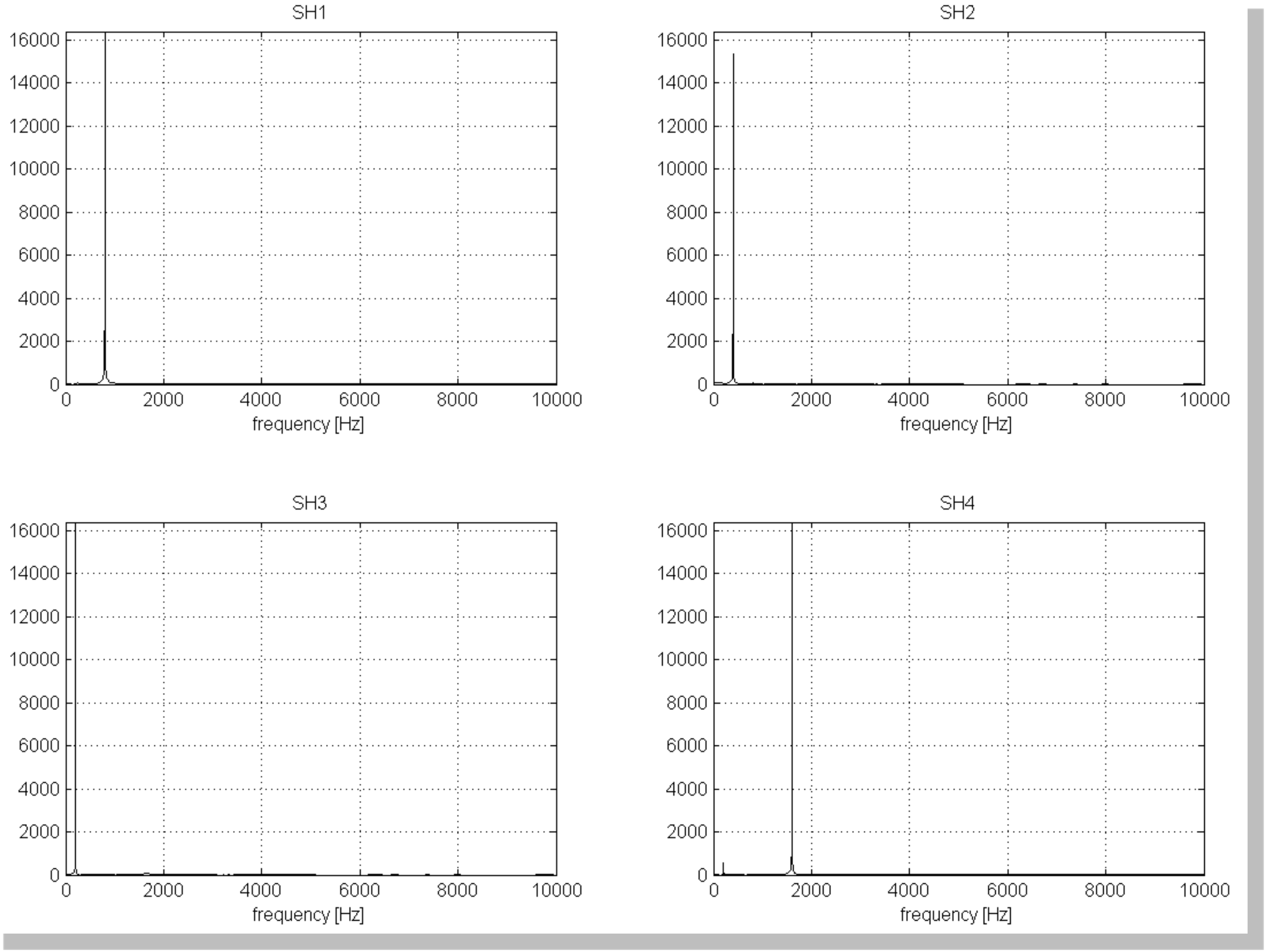
We obtain $\bar{\mathbf{s}}\quad\omega$ from $\hat{\mathbf{z}}(\omega)$ as:

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

and *s*(t) as:

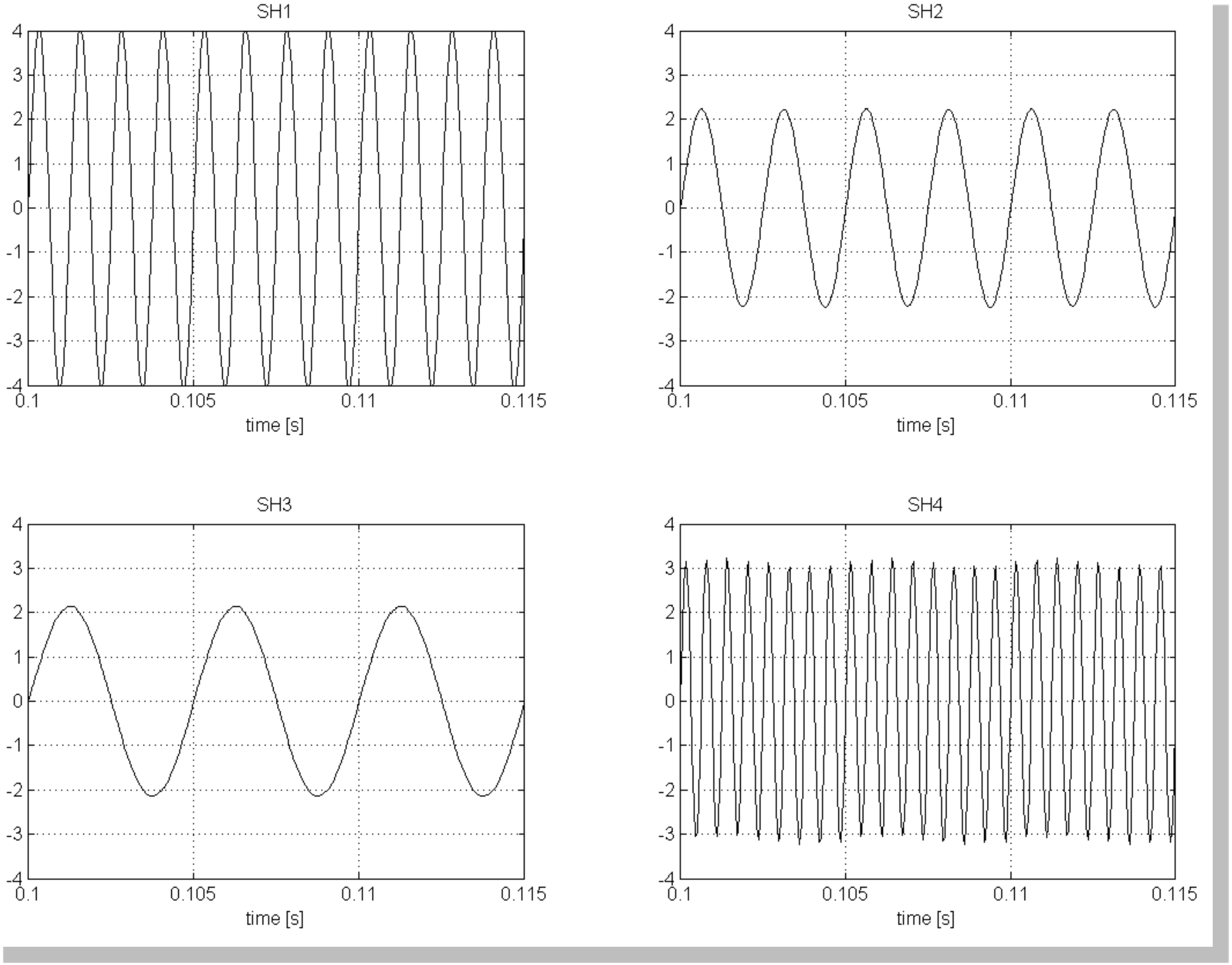
$$s_m(t) = IDFT\ \ \bar{s}_m(\omega)$$

Blind separation of four sine signals from two mixtures



Magnitudes of the estimated sources in FREQUENCY DOMAIN

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

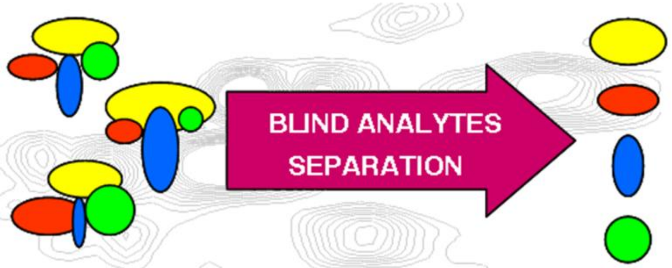
I. Kopriva**,** I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1920.
I. Kopriva, I. Jerić (2009). Multi-component Analysis: Blind Extraction of Pure Components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

Linear mixing model

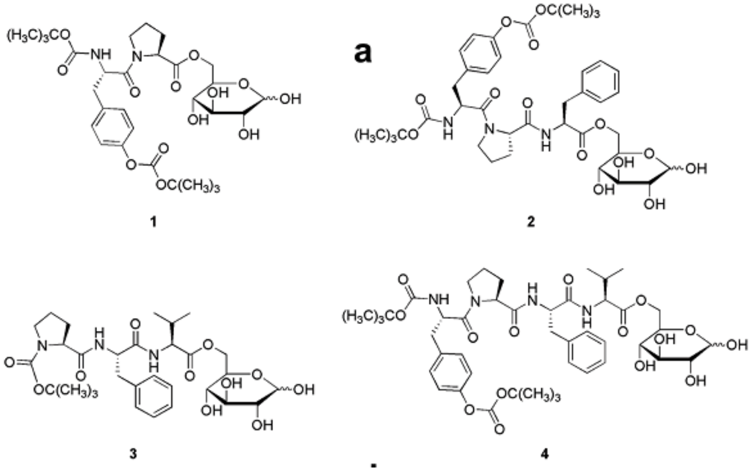
X=AS $$\mathbf{X}\in\mathbb{R}_{0+}^{N\times T},\ \mathbf{S}\in\mathbb{R}_{0+}^{M\times T},\ \mathbf{A}\in\mathbb{R}_{0+}^{N\times M} \qquad (1)$$

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

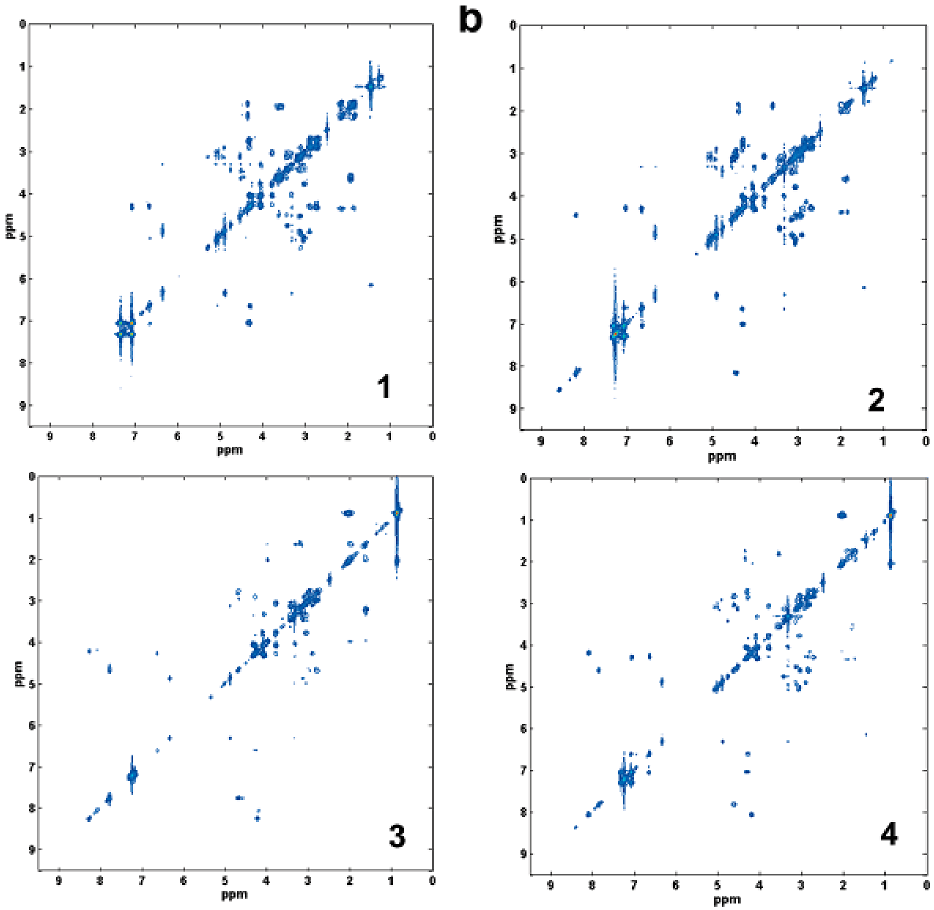
The (u)BSS problem relates to extraction of anlytes (and their concentratios) using mixture spectra **X** only:



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).

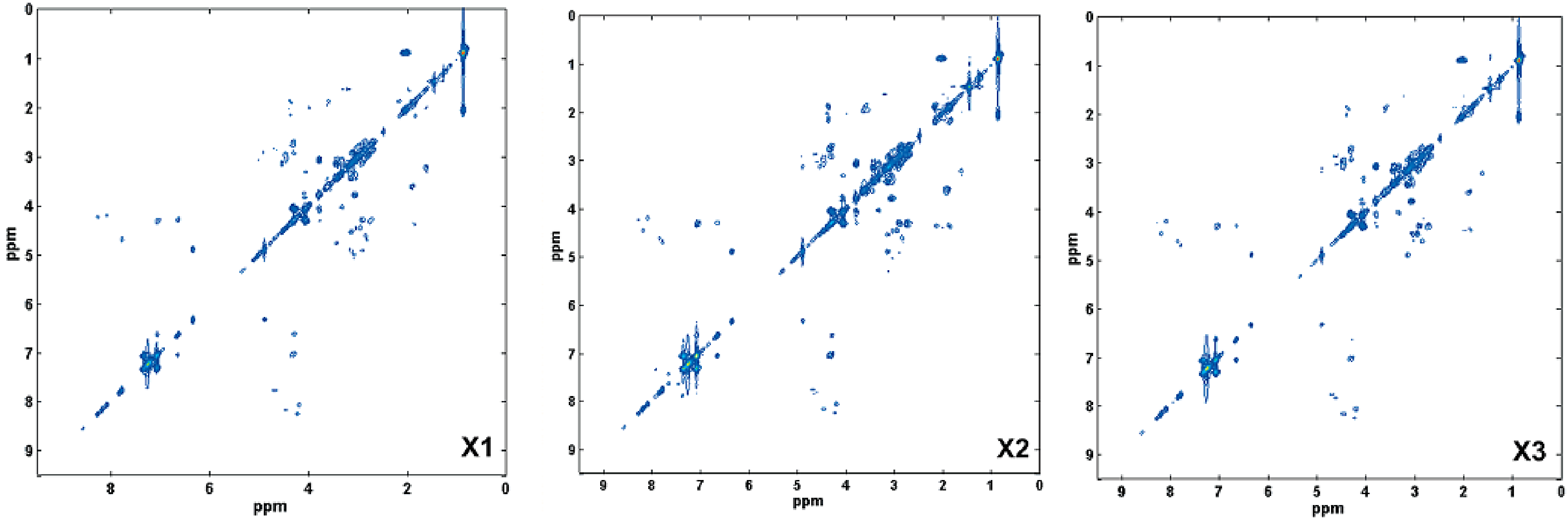


Structure of four analytes (glycopeptides)

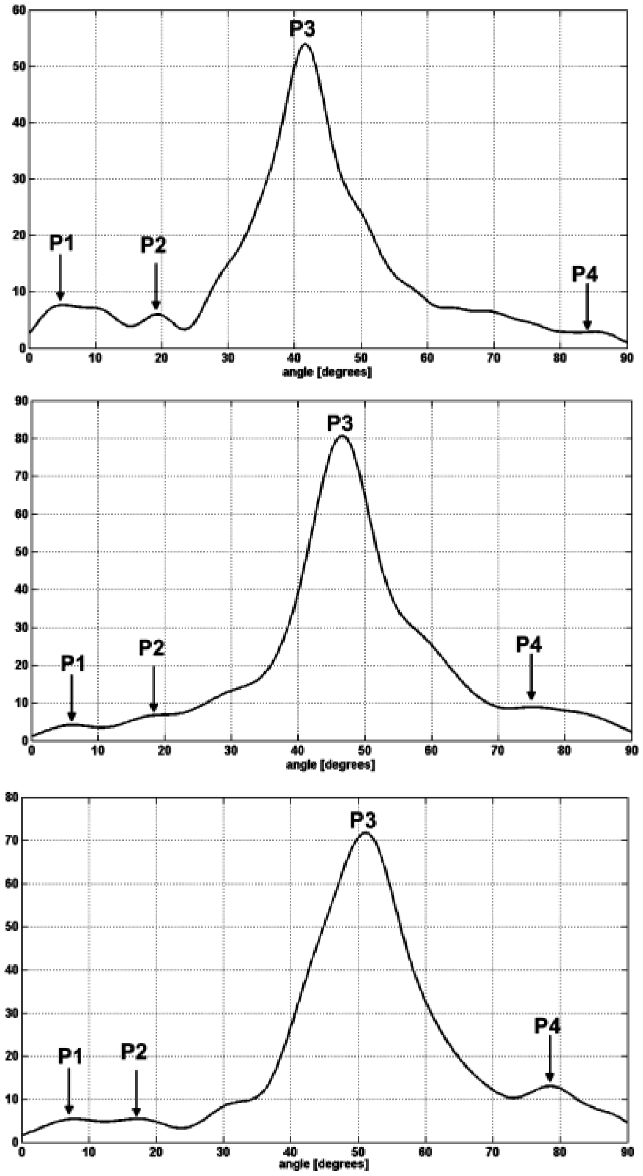


COSY NMR spectra of four analytes

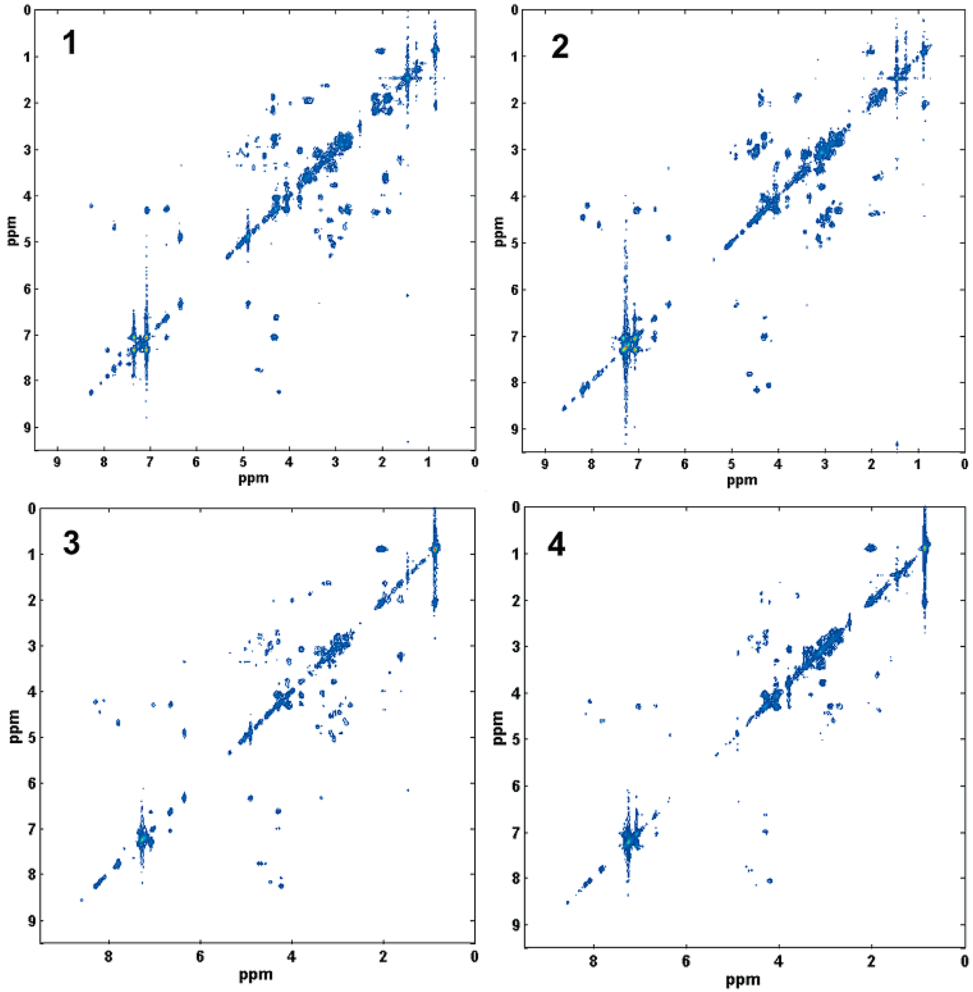
I. Kopriva**,** I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1920.



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.



Estimated COSY NMR spectra of analytes in 2D Fourier domain

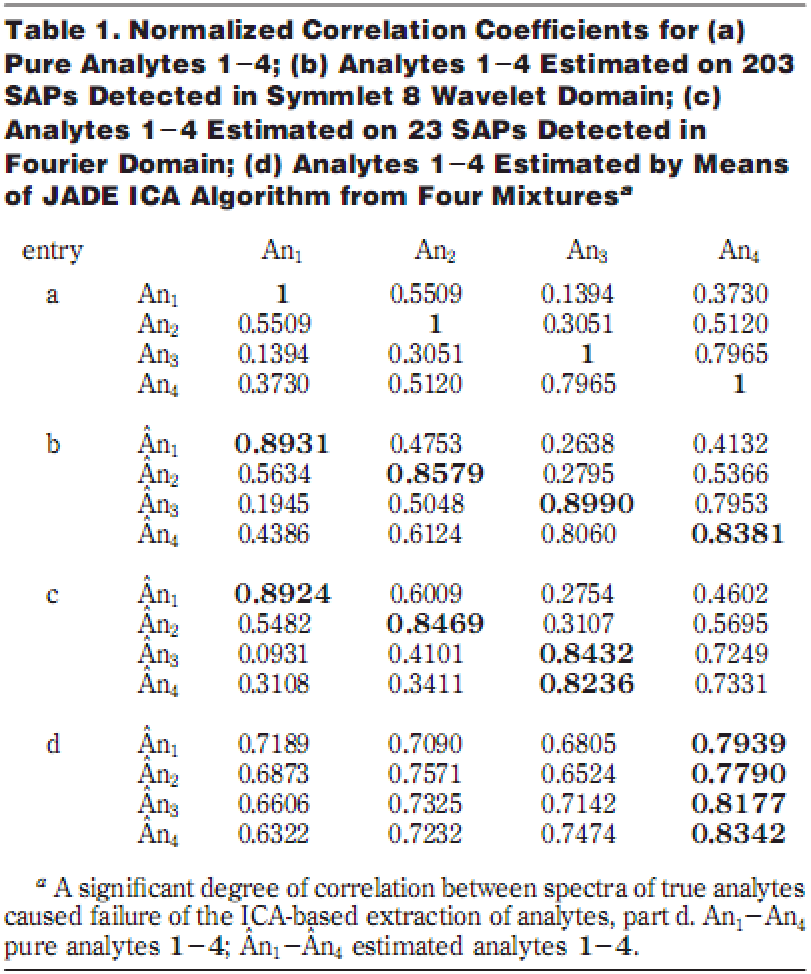
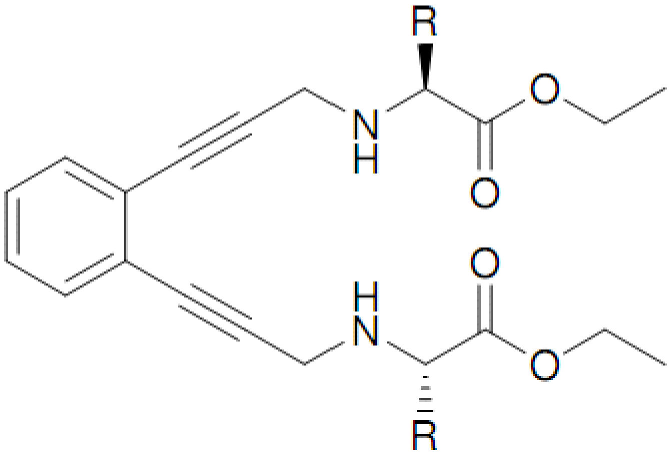


Table 1. Normalized Correlation Coefficients for (a) Pure Analytes 1−4; (b) Analytes 1−4 Estimated on 203 SAPs Detected in Symmlet 8 Wavelet Domain; (c) Analytes 1−4 Estimated on 23 SAPs Detected in Fourier Domain; (d) Analytes 1−4 Estimated by Means of JADE ICA Algorithm from Four Mixtures[a]

entry		An_1	An_2	An_3	An_4
a	An_1	1	0.5509	0.1394	0.3730
	An_2	0.5509	1	0.3051	0.5120
	An_3	0.1394	0.3051	1	0.7965
	An_4	0.3730	0.5120	0.7965	1
b	$\hat{\mathrm{An}}_1$	**0.8931**	0.4753	0.2638	0.4132
	$\hat{\mathrm{An}}_2$	0.5634	**0.8579**	0.2795	0.5366
	$\hat{\mathrm{An}}_3$	0.1945	0.5048	**0.8990**	0.7953
	$\hat{\mathrm{An}}_4$	0.4386	0.6124	0.8060	**0.8381**
c	$\hat{\mathrm{An}}_1$	**0.8924**	0.6009	0.2754	0.4602
	$\hat{\mathrm{An}}_2$	0.5482	**0.8469**	0.3107	0.5695
	$\hat{\mathrm{An}}_3$	0.0931	0.4101	**0.8432**	0.7249
	$\hat{\mathrm{An}}_4$	0.3108	0.3411	**0.8236**	0.7331
d	$\hat{\mathrm{An}}_1$	0.7189	0.7090	0.6805	**0.7939**
	$\hat{\mathrm{An}}_2$	0.6873	0.7571	0.6524	**0.7790**
	$\hat{\mathrm{An}}_3$	0.6606	0.7325	0.7142	**0.8177**
	$\hat{\mathrm{An}}_4$	0.6322	0.7232	0.7474	**0.8342**

[a] A significant degree of correlation between spectra of true analytes caused failure of the ICA-based extraction of analytes, part d. An_1−An_4 pure analytes 1−4; $\hat{\mathrm{An}}_1$−$\hat{\mathrm{An}}_4$ estimated analytes 1−4.



5 $R=H$

6 $R=CH_3$

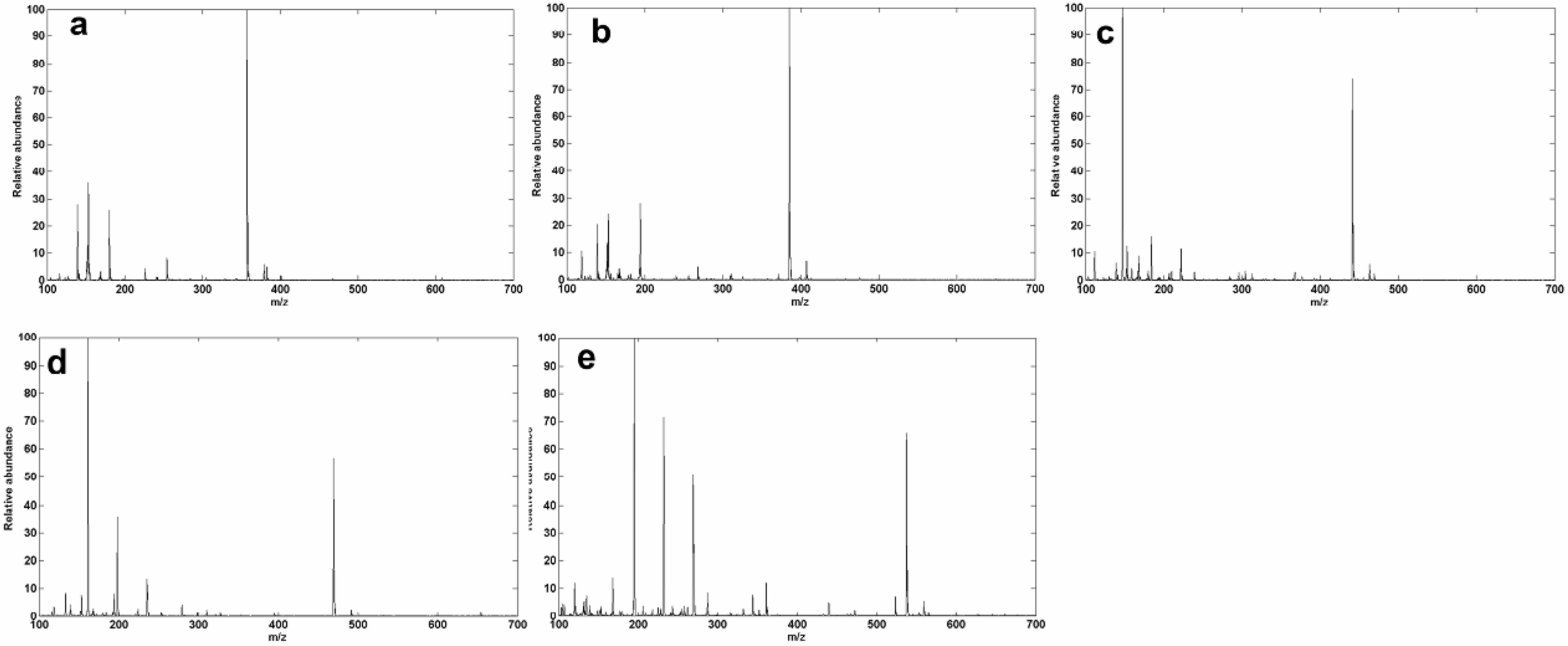
7 $R=CH(CH_3)_2$

8 $R=CH_2CH(CH_3)_3$

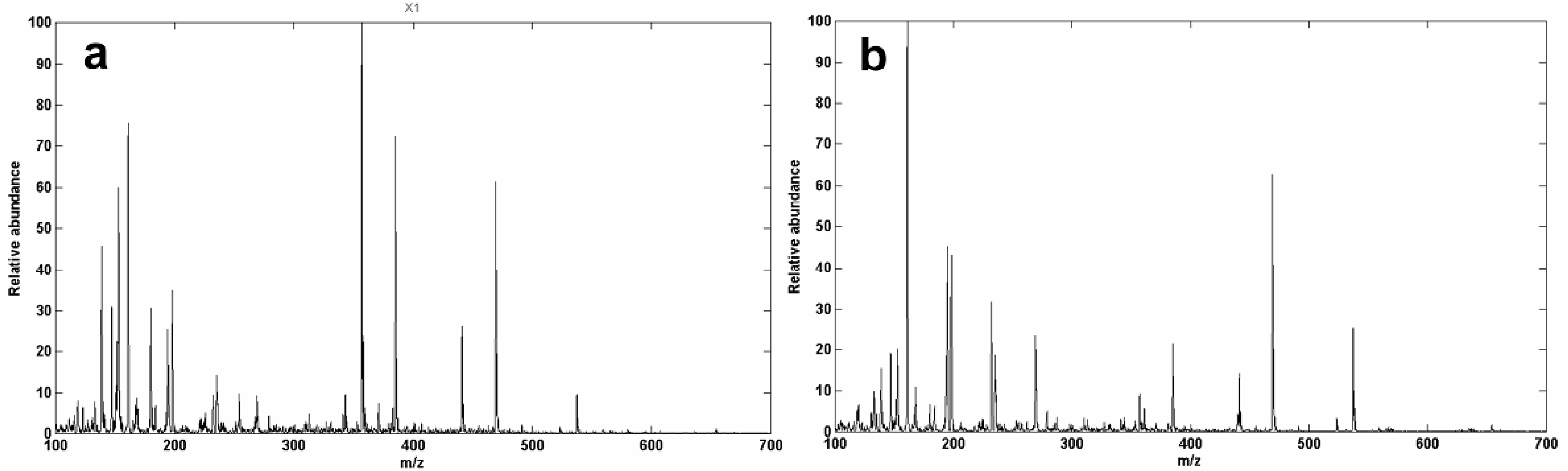
9 $R=CH_2C_6H_5$

Chemical structure of five pure components.

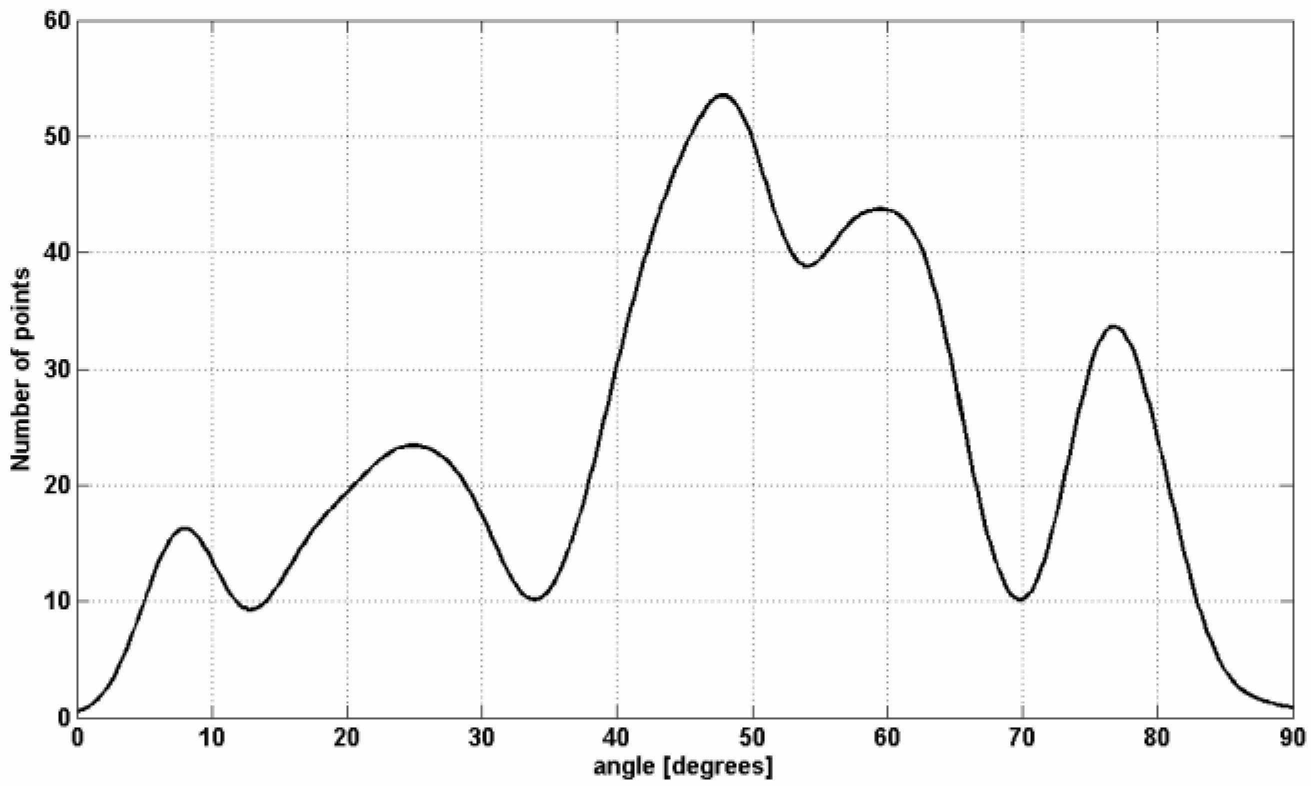
I. Kopriva, I. Jerić (2010). Blind separation of analytes in nuclear magnetic resonance spectroscopy and mass spectrometry: sparseness-based robust multicomponent analysis, *Analytical Chemistry* **82**:1911-1920.



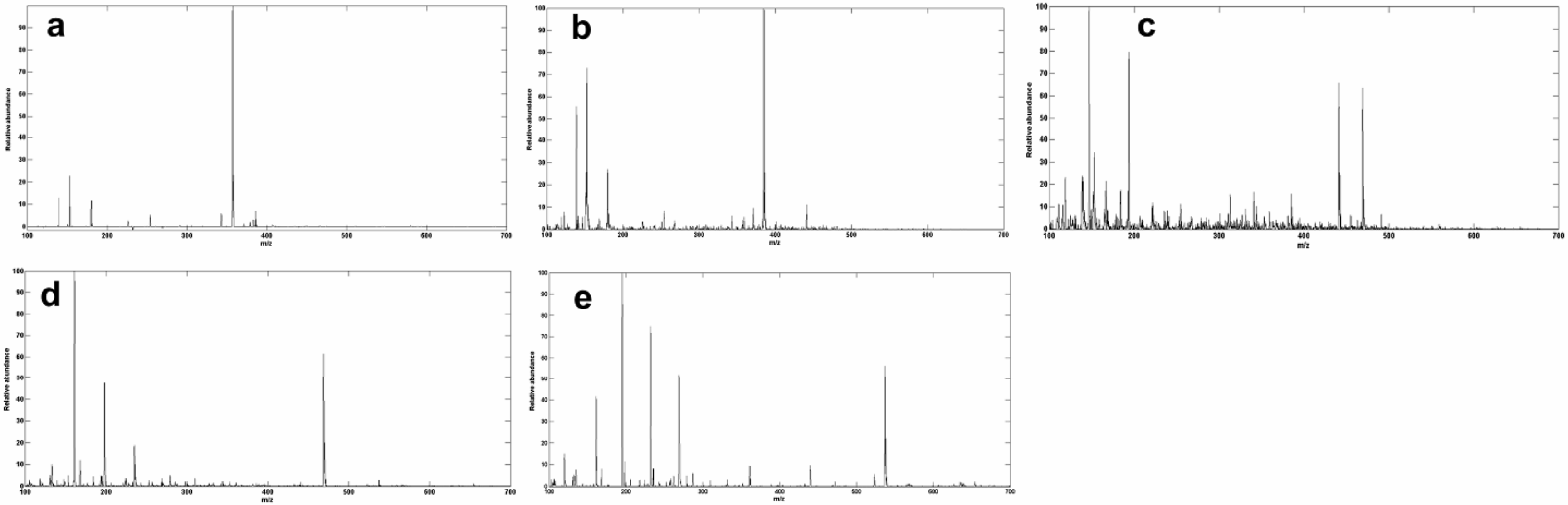
Mass spectra of five pure components.



Mass spectra of two mixtures



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



Estimated mass spectra of five pure components.

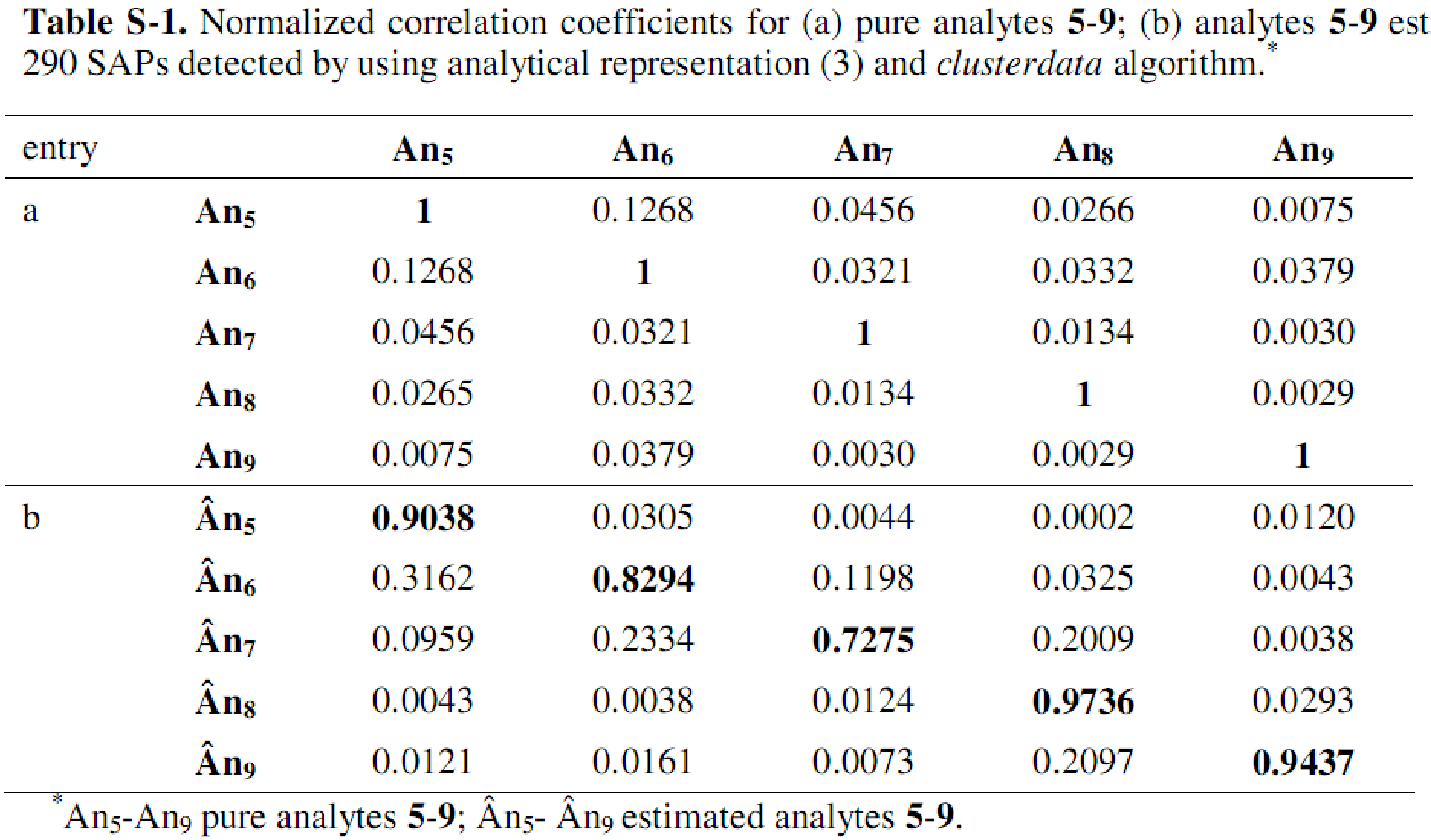


Table S-1. Normalized correlation coefficients for (a) pure analytes **5-9**; (b) analytes **5-9** estimated on 290 SAPs detected by using analytical representation (3) and *clusterdata* algorithm.[*]

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

[*] An$_5$-An$_9$ pure analytes **5-9**; $\hat{A}$n$_5$- $\hat{A}$n$_9$ 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):

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

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 **A** and **S**, to narrow down number of possible decomposition of **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 **A** and **S** through alternative minimization procedure of the two possibly different cost functions:

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

For** k=1,2,…, until convergence **do

Step 1: $\mathbf{S}^{(k+1)} = \underset{s_{mt} \geq 0}{\arg\min}\, D_{\mathbf{S}} \;\; \mathbf{X} \| \mathbf{A}^{(k)}\mathbf{S} \;\; {}_{\mathbf{S}^{(k)}}$

Step 2: $\mathbf{A}^{(k+1)} = \underset{a_{nm} \geq 0}{\arg\min}\, D_{\mathbf{A}} \;\; \mathbf{X} \| \mathbf{A}\mathbf{S}^{(k+1)} \;\; {}_{\mathbf{A}^{(k)}}$

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:

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

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

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

- In many instances non-negativity constraints imposed on **A** and **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]:

$$D\;\mathbf{X}\|\mathbf{AS}\; = \frac{1}{2}\left\|\mathbf{X}-\mathbf{AS}\right\|_F^2 + \alpha_{\mathbf{S}} J_{\mathbf{S}}(\mathbf{S}) + \alpha_{\mathbf{A}} J_{\mathbf{A}}(\mathbf{A})$$

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

$$\frac{\partial D\;\mathbf{A},\mathbf{S}}{\partial a_{nm}} = \left[-\mathbf{X}\mathbf{S}^{\mathrm{T}} + \mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}\right]_{nm} + \alpha_{\mathbf{A}} \frac{\partial J_{\mathbf{A}}(\mathbf{A})}{\partial a_{nm}}$$

$$\frac{\partial D\;\mathbf{A},\mathbf{S}}{\partial s_{mt}} = \left[-\mathbf{A}^{\mathrm{T}}\mathbf{X} + \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}\right]_{mt} + \alpha_{\mathbf{S}} \frac{\partial J_{\mathbf{S}}(\mathbf{S})}{\partial s_{mt}}$$

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*(**A**,**S**I**X**) yields:

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

Above formulation is equivalent to maximizing likelihood P**(X**I**A**,**S)** and maximizing prior probabilities P(**A**) and P(**S**). Assuming normal distribution of approximation error **E=X-AS** this yields:

$$\mathbf{A}^{*},\mathbf{S}^{*} \;= \arg\min_{\mathbf{A},\mathbf{S}} \frac{1}{2}\left\|\mathbf{X}-\mathbf{AS}\right\|_{F}^{2} + \alpha_{\mathbf{S}} J_{\mathbf{S}}(\mathbf{S}) + \alpha_{\mathbf{A}} J_{\mathbf{A}}(\mathbf{A}) \;\; s.t.\,\mathbf{A}\geq\mathbf{0},\mathbf{S}\geq\mathbf{0}.$$

Maximum a posteriori probability BSS/NMF

Assuming non-informative prior on **A**: *P*(**A**)=const and Laplacian (sparse) prior on **S**: $P(\mathbf{S}) = \exp - \left|\mathbf{s}_1\right| + \ldots + \left|\mathbf{s}_M\right|$ yields:

$$\mathbf{A}^*, \mathbf{S}^* = \underset{\mathbf{A},\mathbf{S}}{\arg\min} \frac{1}{2}\left\|\mathbf{X} - \mathbf{AS}\right\|_F^2 + \alpha_{\mathbf{S}}\left\|\mathbf{S}\right\|_1 \; s.t. \, \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}.$$

It is possible to select for *P*(**S**) prior other than Laplacian. That leads to general sparseness constrained factorization:

$$\mathbf{A}^*, \mathbf{S}^* = \underset{\mathbf{A},\mathbf{S}}{\arg\min} \frac{1}{2}\left\|\mathbf{X} - \mathbf{AS}\right\|_F^2 + \alpha_{\mathbf{S}}\left\|\mathbf{S}\right\|_p \; s.t. \, 0 < p \leq 1, \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}.$$

Nonnegative matrix factorization

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

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

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:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\left[\mathbf{X}\mathbf{S}^{\mathrm{T}} - \alpha_{\mathbf{A}} \frac{\partial J_{\mathbf{A}}(\mathbf{A})}{\partial \mathbf{A}}\right]_{+}}{\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}} + \varepsilon \mathbf{1}_{NM}} \qquad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\left[\mathbf{A}^{\mathrm{T}}\mathbf{X} - \alpha_{\mathbf{S}} \frac{\partial J_{\mathbf{S}}(\mathbf{S})}{\partial \mathbf{S}}\right]_{+}}{\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S} + \varepsilon \mathbf{1}_{MT}}$$

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

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

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]:

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

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. a_{nm} and s_{mt} the following learning rules in MATLAB notation are obtained

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

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

where ⊗ denotes component-wise multiplication, and ⊘ denotes component-wise division. Relaxation parameter $\omega \in (0,2]$ provides improvement of the convergence, while $\alpha_{\mathbf{S}} \geq 0$ and $\alpha_{\mathbf{A}} \geq 0$ 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 α-divergence cost function can be used

$$D\ \ \mathbf{X}\|\mathbf{AS}\ = \frac{1}{\alpha(\alpha-1)}\sum_{nt} x_{nt}^{\alpha}\ \ \mathbf{AS}_{nt}^{1-\alpha} - \alpha x_{nt} + (\alpha-1)\ \ \mathbf{AS}_{nt}$$

I-divergence is obtained in the limit when α→1, and dual Kullback-Leibler divergence when α→0. Using MATLAB notation the following update rules are obtained for α≠0,1.

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

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

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

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:

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

Local cost functions can be also squared Euclidean norms

$$D^{(m)}\ \ \mathbf{X}^{(m)}\|\mathbf{a}_m\underline{\mathbf{s}}_m\ \ =\frac{1}{2}\left\|\mathbf{X}^{(m)}-\mathbf{a}_m\underline{\mathbf{s}}_m\right\|_F^2+\alpha_{\mathbf{s}}^{(m)}J_{\mathbf{s}}(\underline{\mathbf{s}}_m)+\alpha_{\mathbf{a}}^{(m)}J_{\mathbf{a}}(\mathbf{a}_m)\quad m=1,\ldots,M$$

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

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 **A** and/or **S** yields

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

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

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

where $\mathbf{I}_{1\times T}$ is an M×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 λ 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: $\mathbf{X} \cong \mathbf{A}^{(1)}\mathbf{A}^{(2)}\cdots\mathbf{A}^{(L)}\mathbf{S}^{(L)}$.

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 $D(\mathbf{X}||\mathbf{AS})$ with respect to **A** and **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,…,R do
    Initialize randomly $\mathbf{A}^{(0)}$ and $\mathbf{S}^{(0)}$
    $\{\mathbf{A}^{(r)},\mathbf{S}^{(r)}\} \leftarrow nmf\_algorithm(\mathbf{X},\mathbf{A}^{(0)},\mathbf{S}^{(0)},K_i);$
    compute $d=D(\mathbf{X}||\mathbf{A}^{(r)}\mathbf{S}^{(r)});$
end
$r_{min}=argmin_{1\le n\le R} d_r;$
$\{\mathbf{A},\mathbf{S}\} \leftarrow nmf\_algorithm(\mathbf{X},\mathbf{A}^{(rmin)},\mathbf{S}^{(rmin)},K_f);$
```

Non-negative matrix under-approximation (NMU)

NMF algorithms outlined befor 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 $\mathbf{a}_m\mathbf{s}_\mathbf{m}$ one at a time. Each time $\mathbf{a}_m\mathbf{s}_\mathbf{m}$ is estimated it is removed from $\mathbf{X}\rightarrow \mathbf{X}-\mathbf{a}_m\mathbf{s}_\mathbf{m}$. To prevent subtraction from being negative the under-approximation constraint is imposed on $\mathbf{a}_m\mathbf{s}_\mathbf{m}$: $\mathbf{a}_m\mathbf{s}_\mathbf{m}\leq\mathbf{X}$.

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

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

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 **A** and **S** is less than in **X**. Thus, the underapproximation constraint ensures sparse (parts based) factorization of **X**. This, however, does not imply that **A** and **S** obtained by enforcing underapproximation constrain yields the sparseset decomposition of **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 **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 prior*i. Quite often that is hard to achieve in practice.

MATLAB code for NMF_L0 algorithm is available at: **http://www3.spsc.tugraz.at/people/robert-peharz**.

a) R. Peharz, F. Pernkopf, "Sparse nonnegative matrix factorization with ℓ^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, SNR [dB]$\in\{$10, 20, 30, inf$\}$, 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×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

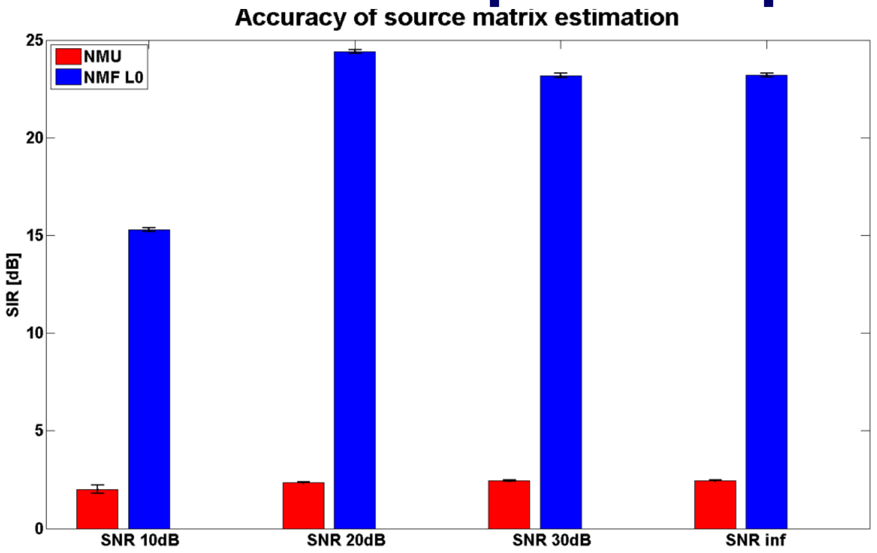
Performance measures:

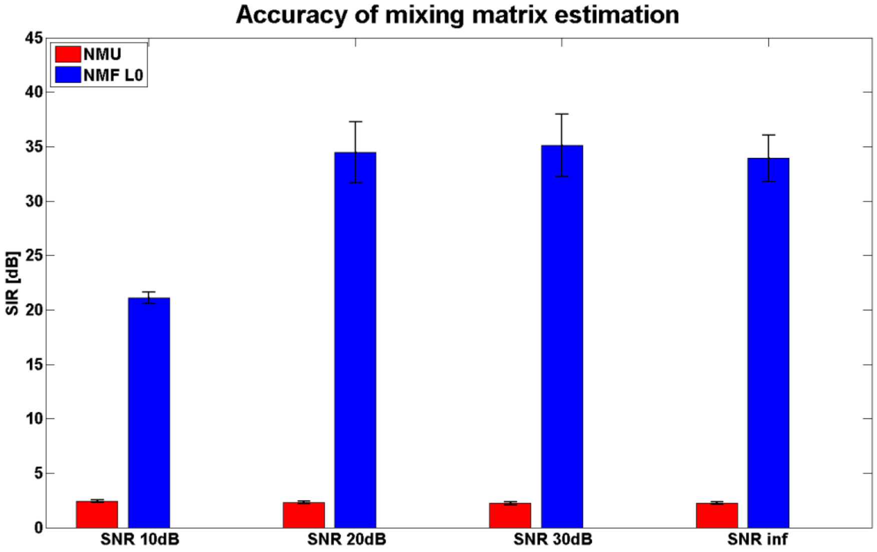
$$\varepsilon(\mathbf{S}) = 10\log_{10}\left(\left\|diag(\mathbf{R}_{\mathbf{SS}})\right\|_2^2 \Big/ \left\|diag(\mathbf{R}_{\mathbf{SS}} - \mathbf{R}_{\mathbf{S\hat{S}}})\right\|_2^2\right)$$

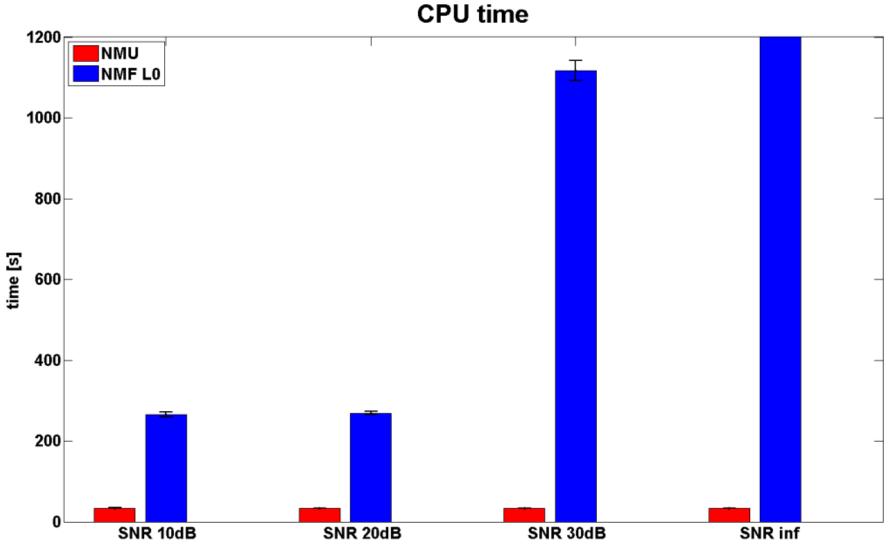
$$\varepsilon(\mathbf{A}) = 10\log_{10}\left(\left\|diag(\mathbf{R}_{\mathbf{AA}})\right\|_2^2 \Big/ \left\|diag(\mathbf{R}_{\mathbf{AA}} - \mathbf{R}_{\mathbf{A\hat{A}}})\right\|_2^2\right)$$

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

Comparative performance analvsis



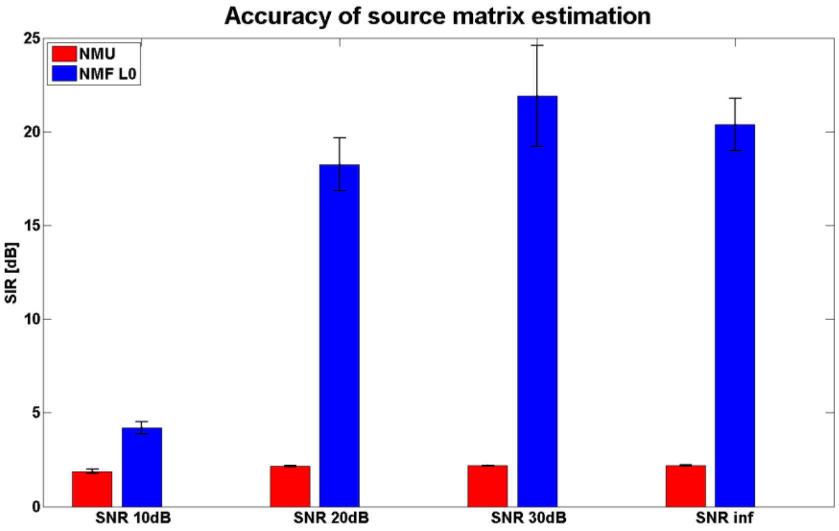


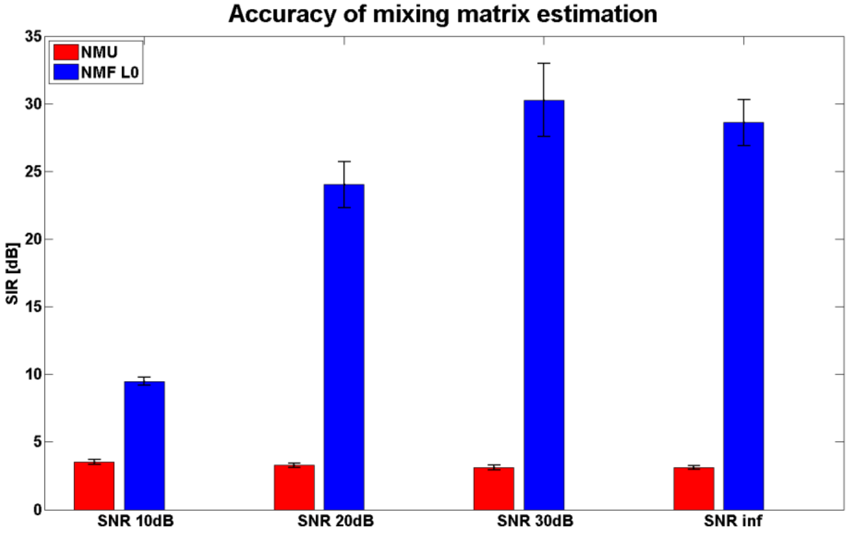


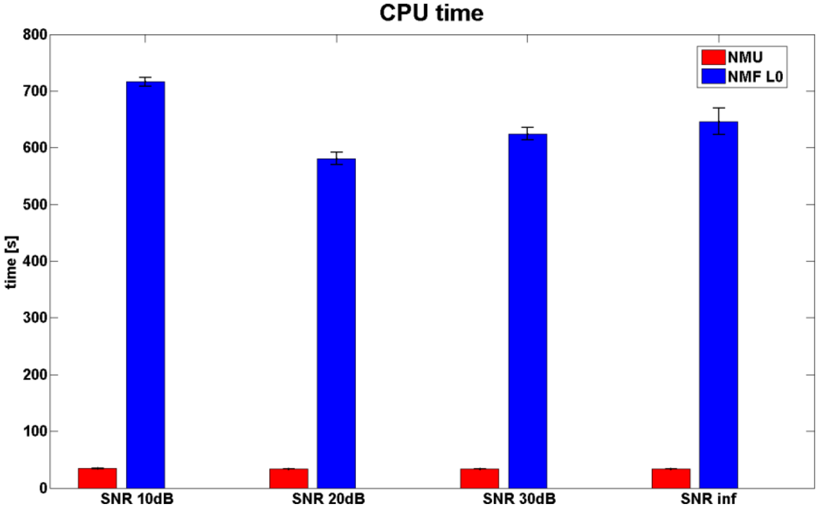
K=10

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

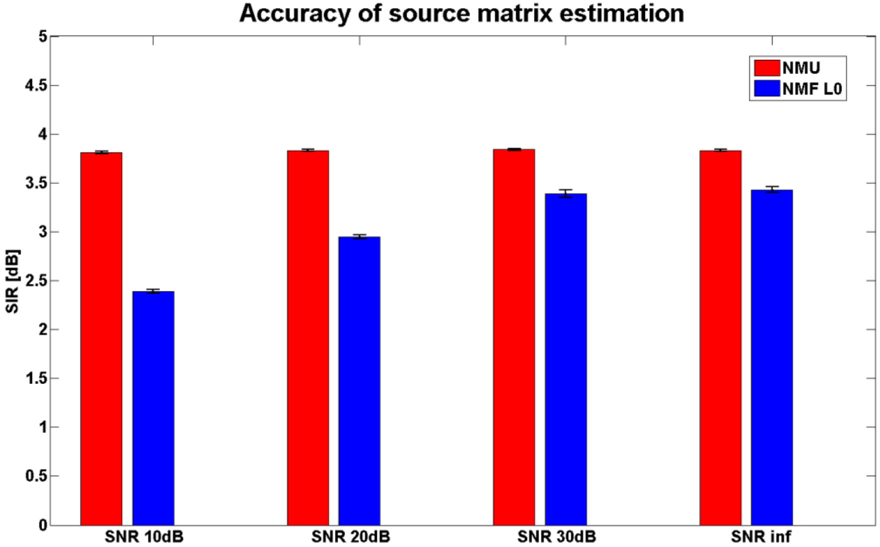

Accuracy of source matrix estimation

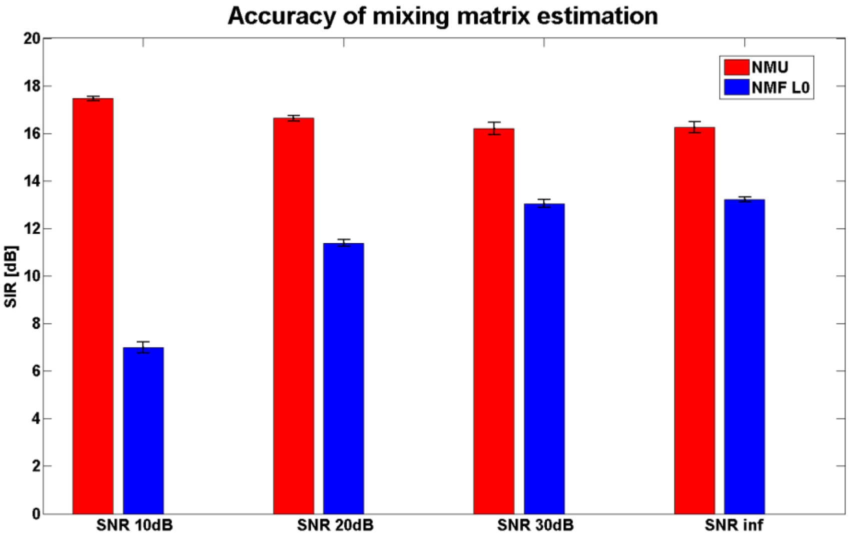

Accuracy of mixing matrix estimation

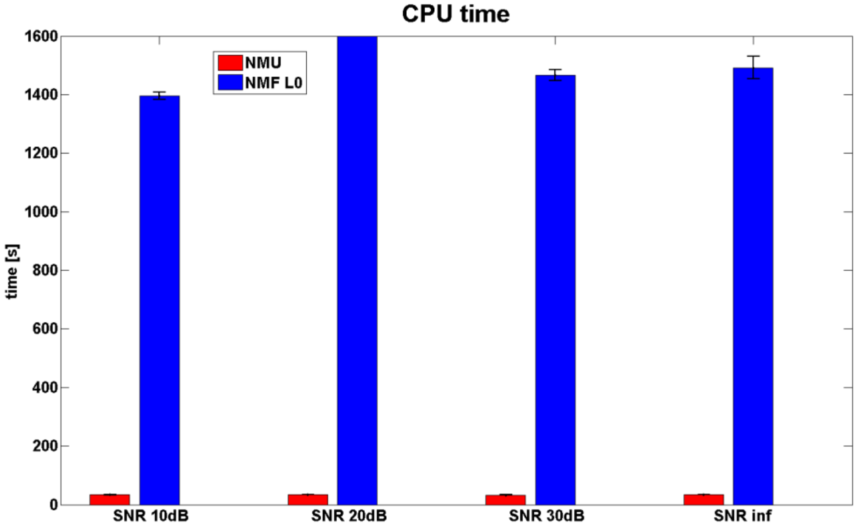

CPU time

K=30

Comparative performance analysis







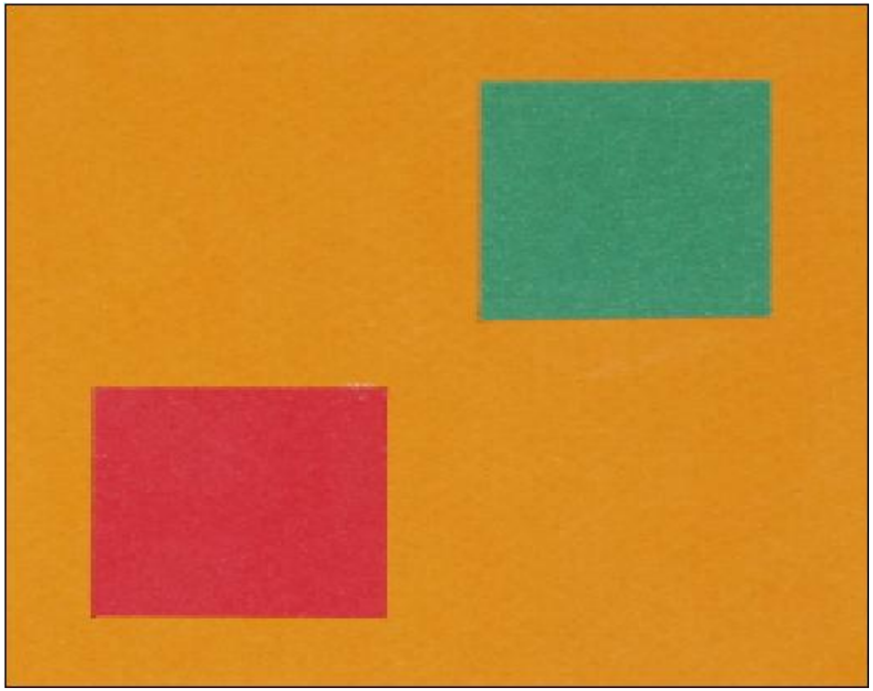
K=50

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 decomposoition



Original RGB image

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).

Multichannel image and linearan mixing model

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

In imaging spectroscopy (multispectral/RGB image) rows of **X** are vectorized channel images (eg. red, green or blue color), columns of **A** are *spectral profiles* of objects (tissues, organs) present in image **X**, and rows of **S** are distributions of intensities of objects (tissues, organs) present in image **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 **X** into anatomically meaningful components: distributions of intensities of objects present in the image **X**.

Unsupervised decomposition of multispectral images

When degree of overlap between objects in spatial domain is very small i.e. $s_{\mathrm{m}}(t)*s_n(t)\approx\delta_{nm}$, 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 **A** and number of objects *M* is achieved by clustering.

Unsupervised decomposition of multispectral images

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

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

•Due to nonnegativity constraints both angles are confined in [0,π/2]. Now estimation of **A** and *M* is obtained by means of data clustering algorithm:

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

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

Unsupervised decomposition of multispectral images

- Calculate function *f*(**a**):

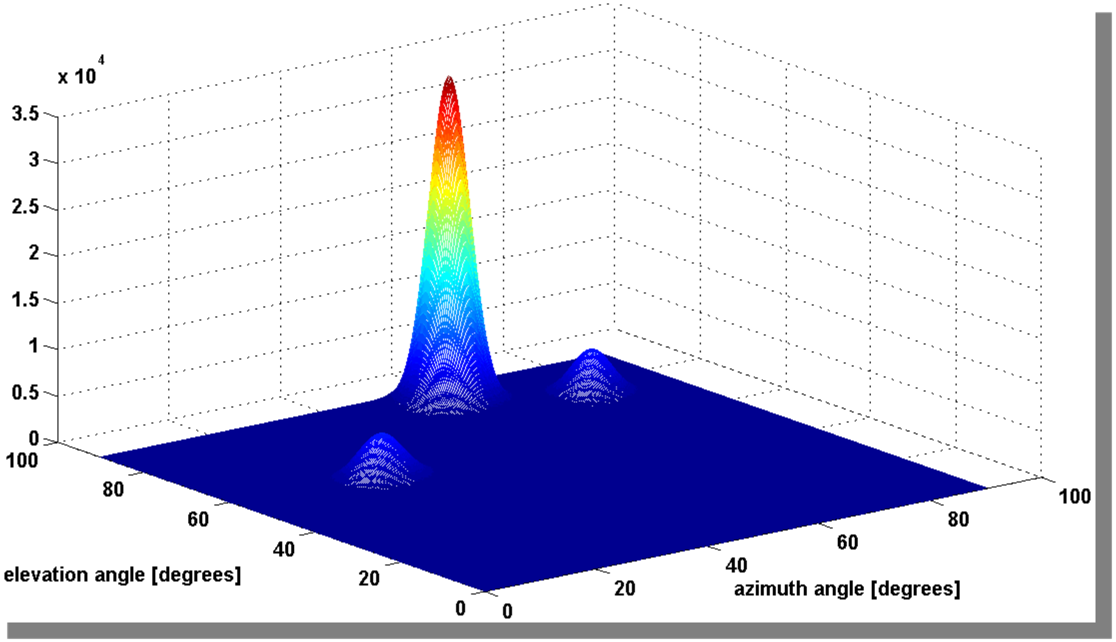
$$f\quad\mathbf{a}\quad=\sum_{t=1}^{\bar{T}}\exp\left(-\frac{d^{2}\quad\mathbf{x}(t),\mathbf{a}}{2\sigma^{2}}\right)$$

where $d\quad\mathbf{x}(t),\mathbf{a}\quad=\sqrt{1-\quad\mathbf{x}(t)\cdot\mathbf{a}\quad^{2}}$ and $\mathbf{x}(t)\cdot\mathbf{a}$ denotes inner product. Parameter σ 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*(**a**) will approximately equal the number of data points close to **a**. Thus by varying mixing angles 0≤φ,θ≤π/2 we effectively cluster data.

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

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**.

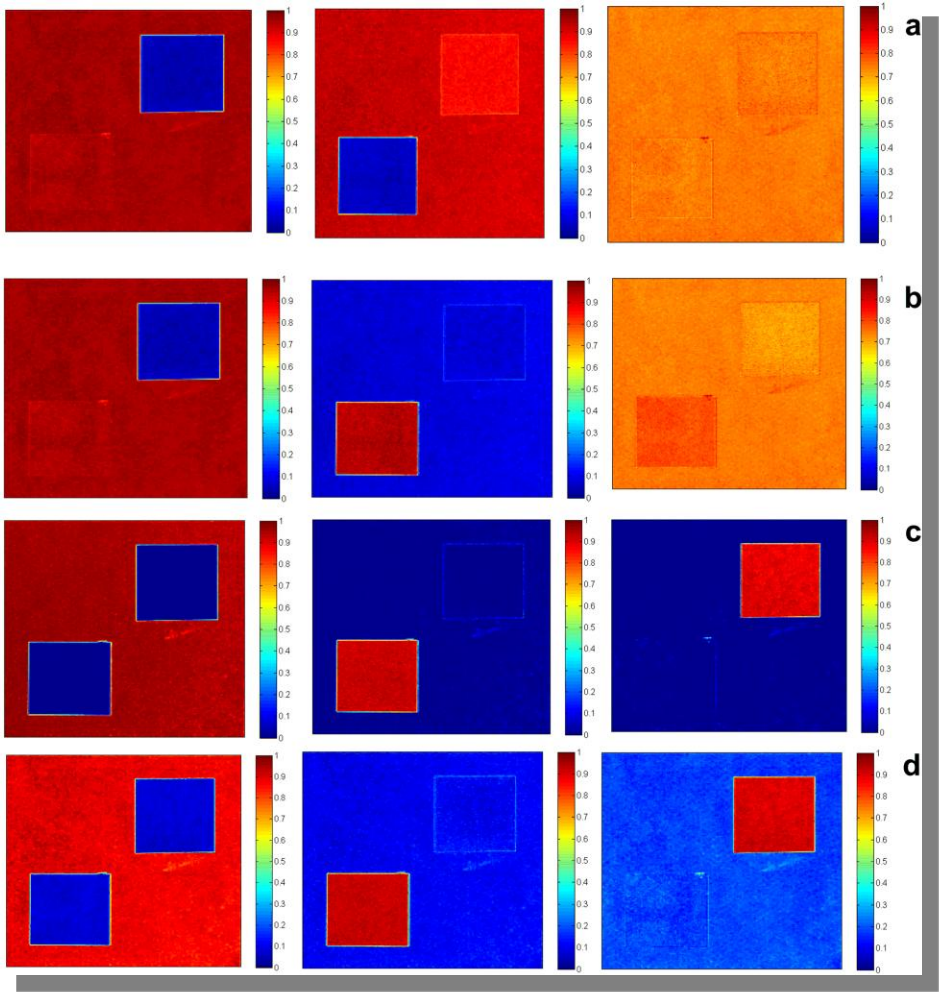
Unsupervised decomposition of multispectral images

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.

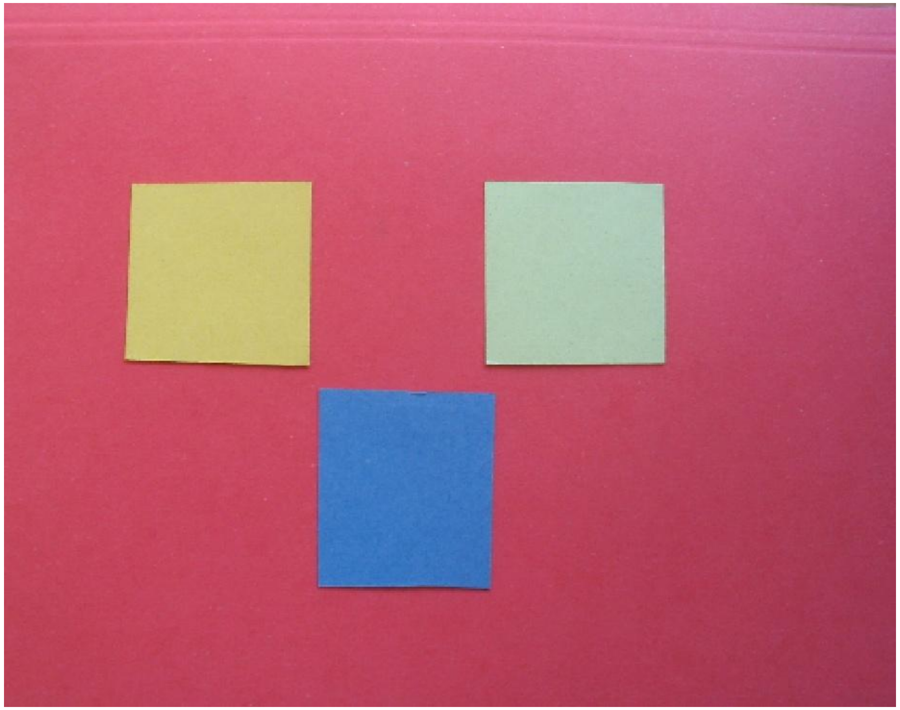
Unsupervised decomposition of multispectral images



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

Unsupervised decomposition of multispectral images

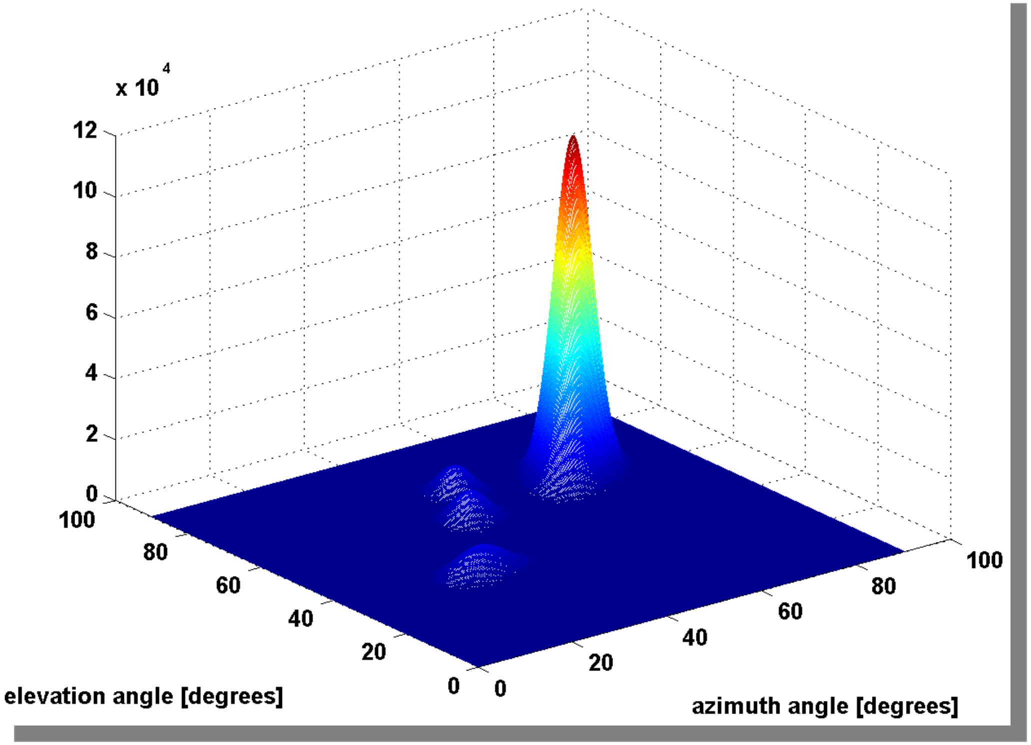
Consider blind decomposition of the RGB image (*N*=3) composed of four materials (*M*=4):



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

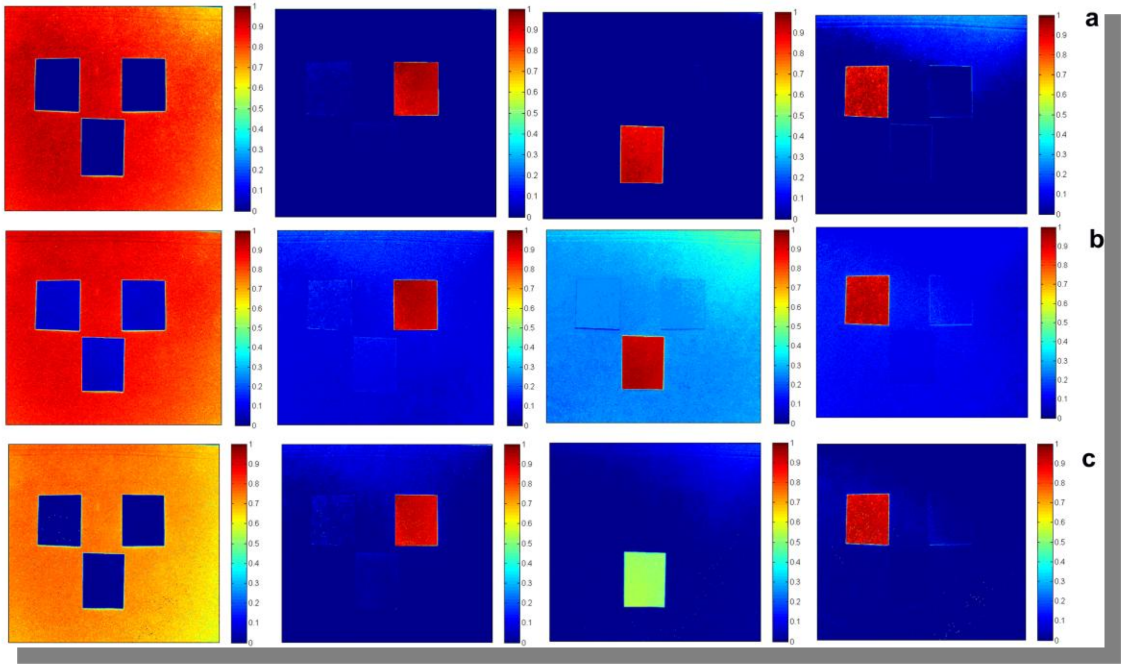
For shown experimental RGB image clustering function is obtained as:



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/.**

Unsupervised decomposition of multispectral images

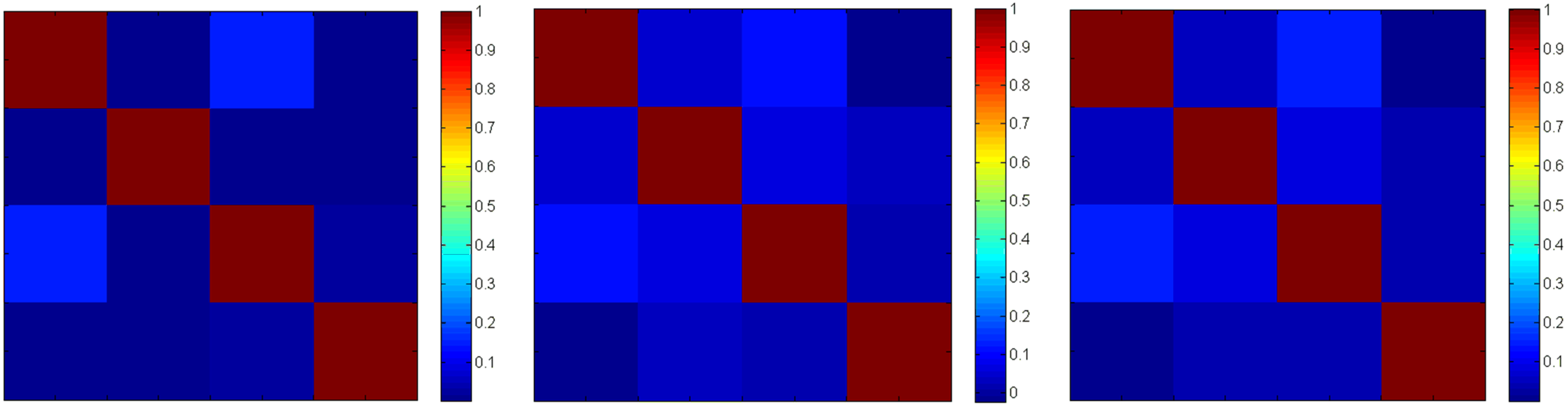
Since materials in the experimental RGB image are orthogonal (they do not overlap in spatial domain) we can evaluate performance of the employed blind image decomposition methods via the correlation matrix defined as **G=SS**^{T}. For perfect estimation the correlation matrix will be diagonal and performance 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 **G** rows of **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.

Sparse component analysis and non-negative matrix factorization in reproducible kernel Hilbert spaces

Preprocessing transforms based on empirical kernel- and explicit feature maps

I. Kopriva, I. Jerić, L. Brkljačić, „Nonlinear mixture-wise expansion approach to underdetermined blind separation of nonnegative dependent sources", *Journal of Chemometrics,* vol. 27, pp. 189-197, 2013.
http://www.lair.irb.hr/ikopriva/publikacije.html

When uBSS problems can(not) be solved?

Let us focus on undeterdermined linear system:

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

Let **s** be *K*-sparse i.e. $K=||\mathbf{s}||_{\mathbf{0}}$.

Provided that **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 **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?

However in BSS problems **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 **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?

One known result for deterministic **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 **A** it applies $N=O(K^2)$. That is significantly worse than $N \approx K\log(M/K)$ for random **A**.

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

When number of sources grows, *K* will grow as well. But *N* is usually fixed for some imaging modality????

Nonlinear mapping of linear uBSS problem?

In ref. [a] a new concept was proposed by mapping original uBSS problem **X=AS** nonlinearly into new one:

$$\mathbf{x}(t) \rightarrow \phi \;\; \mathbf{x}(t) \;\; {}_{t=1}^{T} \quad s.t.\ \mathbf{x}(t) \in \mathbb{R}_{0+}^{N},\ \phi \;\; \mathbf{x}(t) \;\; \in \mathbb{R}_{0+}^{\bar{N}} \ \ and \ \ \bar{N} \gg N$$

since mapping $\phi(\mathbf{x}(t))$ is nonlinear new measurements are linearly independent.

The nonlinear mapping has the following algebraic structure:

$$\phi \;\; \mathbf{x}(t) \;\; = \Big[\;\; c_{q_1 \ldots q_N} x_1^{q_1}(t) \ldots x_N^{q_N}(t) \;\; {}_{q_1,\ldots,q_N=0}^{\bar{N}} \Big]^T \ \ such\ that \ \sum_{n=1}^{N} q_n \leq \bar{N}, \quad \forall t = 1,\ldots,T.$$

a) Kopriva I, Jerić I, Brkljačić, L. Nonlinear mixtures-wise expansion approach to underdetermined blind separation of nonnegative depedent sources. *J. Chemometrics* 2013; **27**: 189-197.

Nonlinear mapping of linear uBSS problem?

The mapped problem becomes:

$$\phi\ \mathbf{x}(t)\ = c_0\mathbf{e}_1 + \mathbf{B}\begin{bmatrix}0\\ \mathbf{s}(t)\end{bmatrix} + \mathbf{B}_{HOT}\begin{bmatrix}0\\ \mathbf{0}_{M\times 1}\\ \mathbf{s}(t)_{HOT}\end{bmatrix} \quad \forall t = 1,\dots,T$$

where **s**(*t*)$_{HOT}$ is $\bar{N} - M - 1$ column vector comprised of: $s_1^{q_1}(t)\times..\times s_M^{q_M}(t)\ {}_{q_1,\dots,q_M=2}^{\bar{N}}$

such that: $\sum_{m=1}^{M} q_m \le \bar{N}$.

If only one source is not present at location *t*, corresponding cross-product is zero. Also, by assuming 0≤s_{mt}≤1 it follows that $s_m^{q_m}(t) \rightarrow 0$ when q_m grows.

Nonlinear mapping of linear uBSS problem?

Thus, by *hard* or *soft thresholding of* $\phi(\mathbf{x}(t))$ higher-order terms can be suppressed. That yields:

$$\phi\ \mathbf{X}\ _{\tau} \approx \left[\underbrace{c_0\mathbf{e}_1 \ldots c_0\mathbf{e}_1}_{\times T\, times}\right] + \bar{\mathbf{B}}\begin{bmatrix}0\\ \bar{\mathbf{S}}\end{bmatrix}$$

where $\phi\ \mathbf{X}\ _{\tau} \in \mathbb{R}_{0+}^{\bar{N}\times T}$, $\bar{\mathbf{B}} \in \mathbb{R}_{0+}^{\bar{N}\times P+1}$, $\bar{\mathbf{S}} \in \mathbb{R}_{0+}^{P\times T}$.

Thus, linear uBSS problem characterized by (*N*,*M*,*K*) is converted into new one characterized by $(\bar{N}, P, Q)$, where *Q* denotes maximal number of overlapping sources in mapped domain. If sources do not overlapp heavily and higher-order terms are suppressed we have:

$$(\bar{N}/N) \gg (P/M) \quad and \quad (\bar{N}/N) \gg (Q/K)$$

Nonlinear mapping of linear uBSS problem?

How to choose nonlinear mapping ? One, smart (?), way is to select ϕ by factorizing positive semi-definite symmetric kernel function *k*(**x**,**y**) on the basis of reproducibility condition:

$$k(\mathbf{x},\mathbf{y}) = \left\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \right\rangle$$

That enables implicit calculations of nonlinear mappings. For polynomial kernel *k*(**x**,**y**)= (<**x**,**y**>+1)^{d} we obtain:

$$\phi\ \ \mathbf{x} = \left\{ \sqrt{\begin{pmatrix} d \\ \boldsymbol{\alpha} \end{pmatrix}} \mathbf{x}^{\boldsymbol{\alpha}} \right\}_{|\boldsymbol{\alpha}| \leq d}$$

$$\begin{pmatrix} d \\ \boldsymbol{\alpha} \end{pmatrix} = \frac{d!}{d - |\boldsymbol{\alpha}|\ \ ! \times \boldsymbol{\alpha}!} \qquad \boldsymbol{\alpha} \in \mathbb{N}_0^N \qquad |\boldsymbol{\alpha}| = \sum_{n=1}^{N} \alpha_n \qquad \boldsymbol{\alpha}! = \prod_{n=1}^{N} \alpha_n! \qquad \mathbf{x}^{\boldsymbol{\alpha}} = \prod_{n=1}^{N} x_n^{\alpha_n}$$

Nonlinear mapping of linear uBSS problem?

Example. For $\mathbf{x} \in \mathbb{R}^3$ nonlinear associated with polynomial kernel of degree *d*=3 yields:

$$\phi \ \mathbf{x} \ = \begin{bmatrix} 1\ \sqrt{3}x_1\ \sqrt{3}x_2\ \sqrt{3}x_3\ \sqrt{3}x_1^2\ \sqrt{3}x_2^2\ \sqrt{3}x_3^2\ x_1^3\ x_2^3\ x_3^3\ \sqrt{3}x_1^2x_2\ \sqrt{3}x_1^2x_3\ \ldots \\ \ldots\sqrt{3}x_1x_2^2\ \sqrt{3}x_1x_3^2\ \sqrt{3}x_2^2x_3\ \sqrt{3}x_2x_3^2\ \sqrt{6}x_1x_2x_3\ \sqrt{6}x_1x_2\ \sqrt{6}x_1x_3\ \sqrt{6}x_2x_3 \end{bmatrix}^{\mathrm{T}}$$

Thus, 3D vector (pixel in RGB image for example) is mapped into 20D vector. Second order mapping yields 10D vector.

This mapping enables better discrimination of objects with close spectral profiles (in case of MSI) or close density profiles (in case of CT imaging).

Nonlinear mapping of linear uBSS problem?

The problem with using explicit feature maps $\phi(\mathbf{x}(t))$ is that $\bar{N}$ can be very large or even infinite. Thus, factorization problem:

$$\phi \quad \mathbf{X} \quad {}_{\tau} \approx \begin{bmatrix} \underbrace{c_0\mathbf{e}_1 \ldots c_0\mathbf{e}_1}_{\times T\, times} \end{bmatrix} + \bar{\mathbf{B}} \begin{bmatrix} 0 \\ \bar{\mathbf{S}} \end{bmatrix}$$

becomes computationally intractable. That is fixed by projecting $\phi(\mathbf{x}(t))$ onto $\phi(\mathbf{V})$ where $\mathbf{V} = \quad \mathbf{v}_d \in \mathbb{R}^{N\times 1} \quad {}_{d=1}^{D}$ stands for basis such that:

$$span \quad \mathbf{v}_d \quad {}_{d=1}^{D} \approx span \quad \mathbf{x}_t \quad {}_{t=1}^{T}$$

Then:

$$span \quad \phi \quad \mathbf{v}_d \quad {}_{d=1}^{D} \approx span \quad \phi \quad \mathbf{x}_t \quad {}_{t=1}^{T}$$

Nonlinear mapping of linear uBSS problem?

Projection yields:

$$\phi\ \mathbf{V}^{T}\phi\ \mathbf{x}_t\ =\psi\ \mathbf{x}_t\ {}_{\mathbf{V}}=\left[\left\langle\phi\ \mathbf{v}_1\ ,\phi\ \mathbf{x}_t\ \right\rangle\ldots\left\langle\phi\ \mathbf{v}_D\ ,\phi\ \mathbf{x}_t\ \right\rangle\right]^T$$

When ϕ(**x**)=*k*(°,**x**) it follows: <ϕ(**v**), ϕ(**x**)>=*k*(**v,x**). It is shown in ref. a) that:

$$\psi\ \mathbf{X}\ {}_{\tau}\approx\left[\underbrace{c_0\mathbf{e}_1\ldots c_0\mathbf{e}_1}_{\times T\,times}\right]+\bar{\mathbf{B}}\begin{bmatrix}0\\ \bar{\mathbf{S}}\end{bmatrix}$$

$$\psi\ \mathbf{X}\ {}_{\tau}\in\mathbb{R}_{0+}^{D\times T}\qquad\bar{\mathbf{B}}\in\mathbb{R}_{0+}^{D\times P+1}\qquad\bar{\mathbf{S}}\in\mathbb{R}_{0+}^{P\times T}$$

a) Kopriva I, Jerić I, Brkljačić, L. Nonlinear mixtures-wise expansion approach to underdetermined blind separation of nonnegative depedent sources. *J. Chemometrics* 2013; **27**: 189-197.

Nonlinear mapping of linear uBSS problem?

Linear uBSS problem characterized by (*N*,*M*,*K*) is converted into new one characterized by (*D*,*P*,*Q*) , where *Q* denotes maximal number of overlapping sources in mapped domain.

If sources do not overlap heavily and higher-order terms are suppressed we have:

$$(D/N)\gg(P/M)\quad and\quad (D/N)\gg(Q/K)$$

Thus, new BSS problem is easier to solve than the original one.

Numerical experiment

Linear uBSS problem characterized by N=5,M=10,T=1000 and $K\in\{2,3,4\}$ is simulated.

Each source signal is according to p.d.f. based on mixed state random variable model [a, b]:

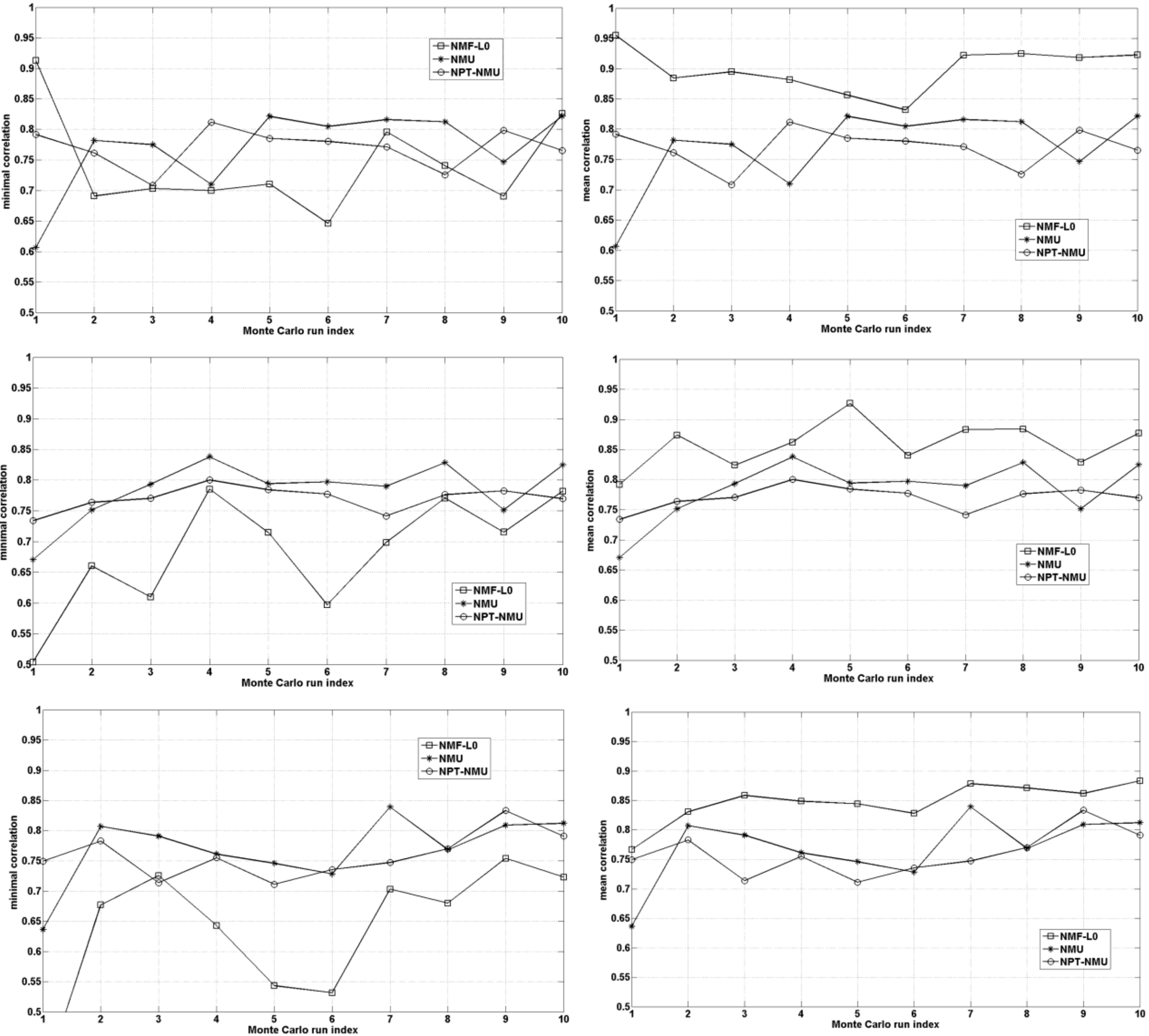
$$p(s_{mt}) = \rho\delta(s_{mt}) + (1-\rho)\delta^{*}(s_{mt})f(s_{mt}) \quad \forall m = 1,\dots,M \ \forall t = 1,\dots,T$$

where $\delta(s_{mt})$ is an indicator function and $\delta^{*}(s_{mt})$ =1- $\delta(s_{mt})$ is its complementary function. $\rho = \{P(\mathbf{s}_{mt} = 0)\}_{t=1}^{T}$ Thus, $\{P(s_{mt} > 0) = 1-\rho\}_{t=1}^{T}$.

a) Bouthemy P, Piriou C H G, Yao J. Mixed-state auto-models and motion texture modeling. *J. Math Imaging Vision* 2006; 25: 387-402.

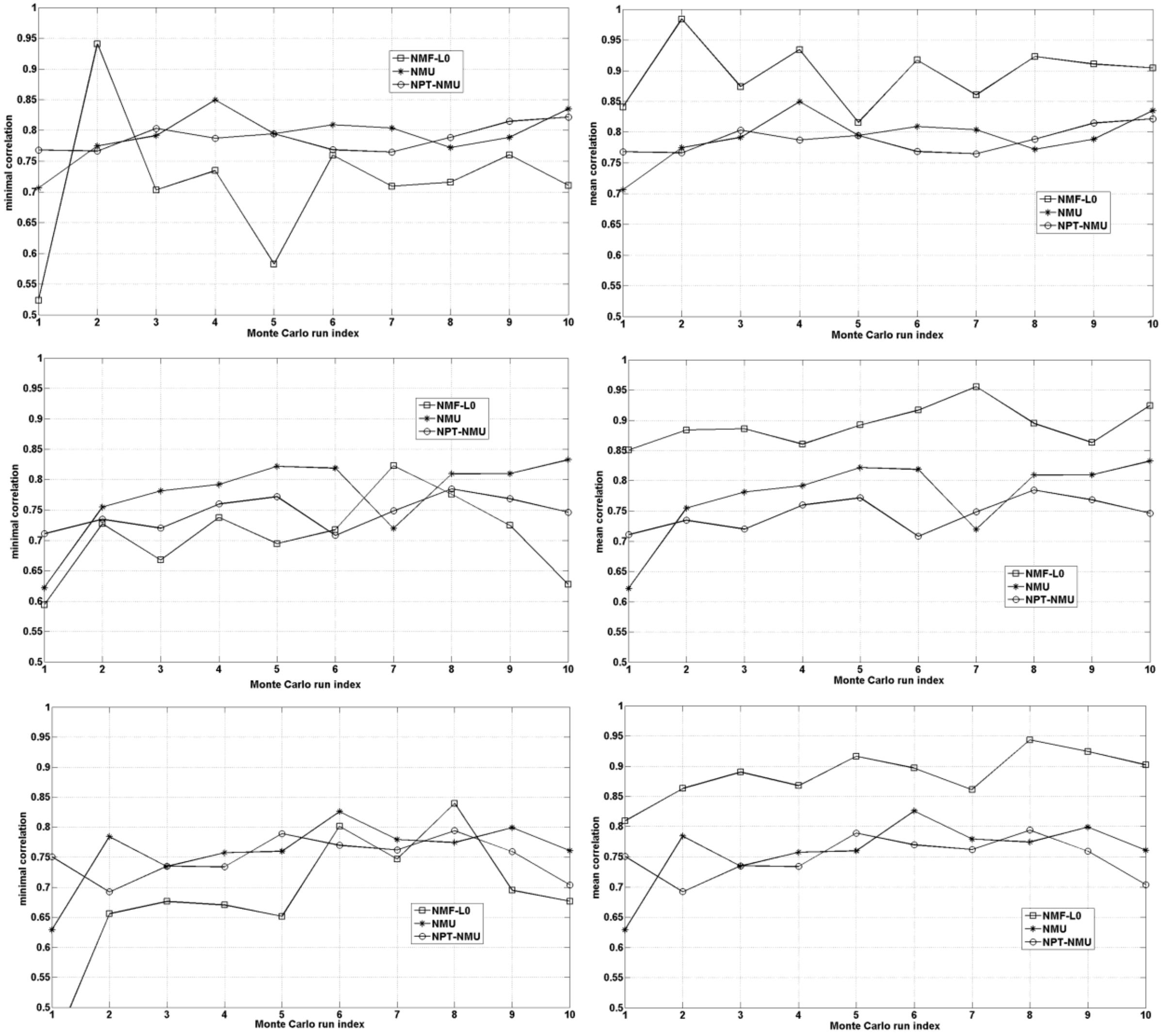
b) Caifa C, Cichocki A. Estimation of Sparse Nonnegative Sources from Noisy Overcomplete Mixtures Using MAP. *Neural Comput.* 2009; 21: 3487-3518.

Numerical experiment



Normalized correlation coefficient vs. Monte Carlo run index between true and extracted sources by algorithms: NMF_L0 (squares), NMU (stars) and NPT-NMU (circles).
Left: minimal (worst) value;
(right) mean value for ten sources.
From top to bottom –
number of overlapping sources K :
2, 3, and 4. Most probable value of the nonzero state, generated according to exponential distribution, equal to 0.1.

NPT-NMU algorithm yields better accuracy than NMU algorithm in 30% of the runs, while NMU is better in 60% of the runs. While average performance of the NMF_L0 algorithm is the best it yields the worse value of the minimal correlation coefficient.



Normalized correlation coefficient vs. Monte Carlo run index between true and extracted sources by algorithms: NMF_L0 (squares), NMU (stars) and NPT-NMU (circles). Left: minimal (worst) value; (right) mean value for ten sources. From top to bottom - number of overlapping sources K : 2, 3, and 4. Most probable value of the nonzero state, generated according to exponential distribution, equal to 0.01.

NPT-NMU algorithm yields better accuracy than NMU algorithm in 30% of the runs, while NMU is better in 60% of the runs. While average performance of the NMF_L0 algorithm is the best it yields the worse value of the minimal correlation coefficient.

Nonlinear decomposition of RGB image of skin cancer

I. Kopriva**,** A. Peršin (2009) Unsupervised decomposition of low-intensity low-dimensional multi-spectral fluorescent images for tumour demarcation, *Medical Image Analysis* , vol.**13**, pp.507-518.

I. Kopriva, Method for real time tumour visualisation and demarcation by means of photodynamic diagnosis, *US Patent* 8,224,427, 17. 7. 2012.

Nonlinear decomposition of RGB image of skin cancer

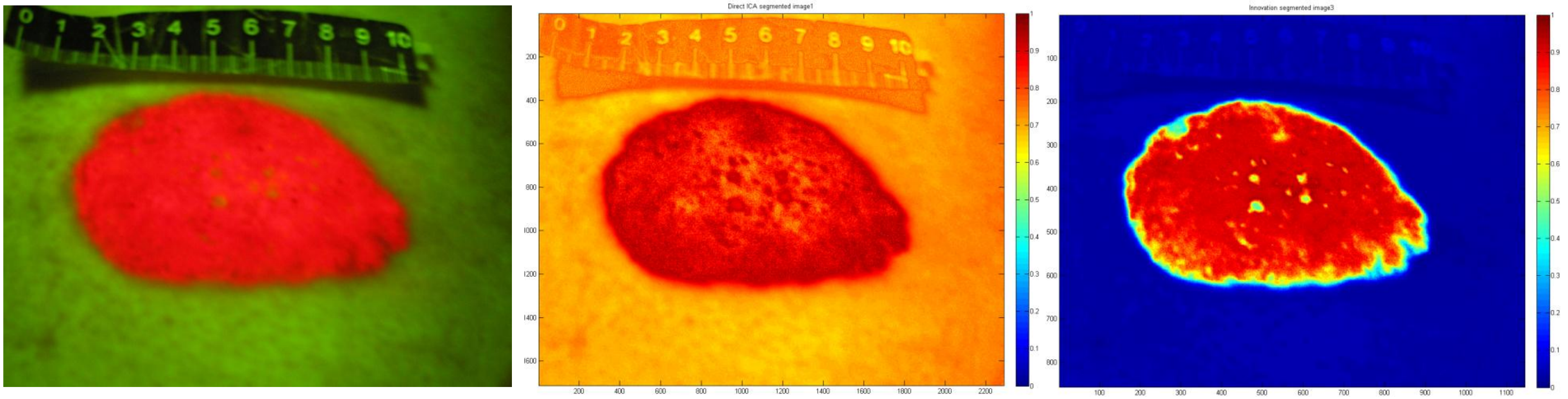
Decomposition of objects with spectrally similar profiles is hard problem. That occurs due to poor spectral resolution or due to physiological reasons.

In fluorescent imaging that occurs when intensity of fluorescence is weak.

As an example that may happen when intensity of illuminating (laser) light is low (we do not to cause damage).

RGB image is first mapped nonlinearly by means of 2nd order explicit feature map (EFM). Afterwards, dependent component analysis (DCA) is executed in induced space.

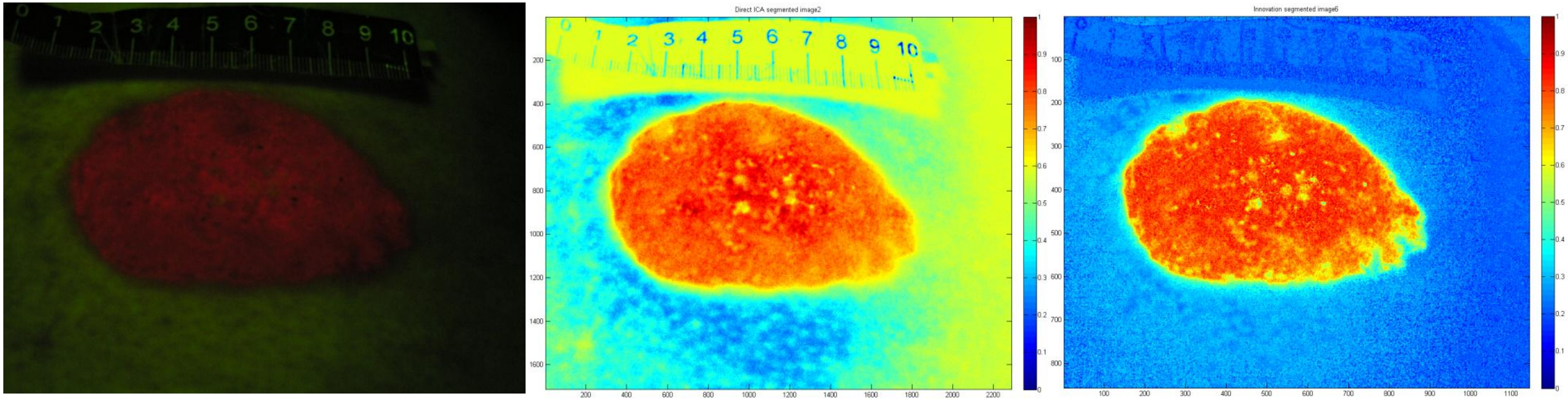
2nd order EFM and dependent component analysis



Left: Experimental high-intensity fluorescent RGB image of the skin tumour (basal cell carcinoma).

Mid: linear ICA algorithm; **Right:** Nonlinear DCA algorithm.

2nd order EFM and dependent component analysis



Left: Experimental low-intensity fluorescent RGB image of the skin tumour (basal cell carcinoma).

Mid: linear ICA algorithm; **Right:** Nonlinear DCA algorithm.

Nonlinear projection on orthonormal basis and ICA?

If centered basis in RKHS is orthogonalized:

$$\psi_c\ \mathbf{V} = \mathbf{I}_D - 1/D\mathbf{1}_D\mathbf{1}_D^T\ \psi\ \mathbf{V}\ \ \mathbf{I}_D - 1/D\mathbf{1}_D\mathbf{1}_D^T\ = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$$

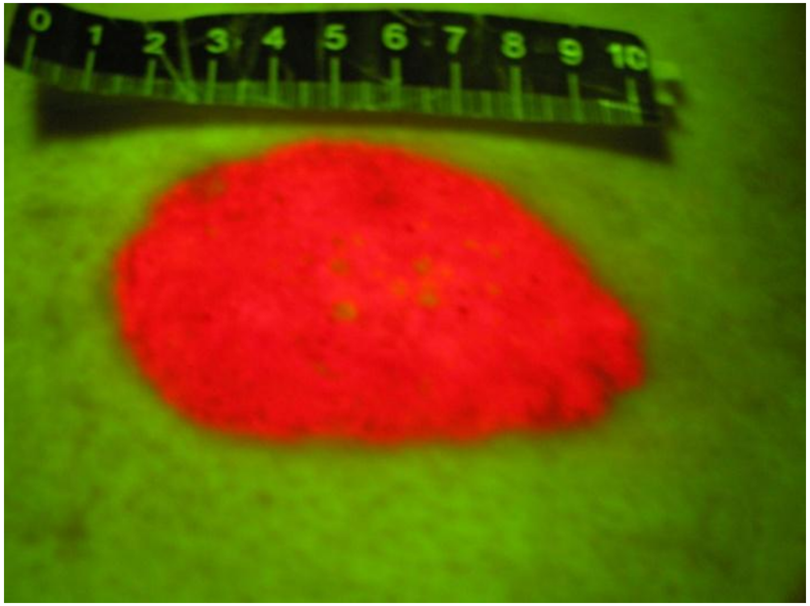
we can project centered data in RKHS onto $\psi_c\ \mathbf{V}$:

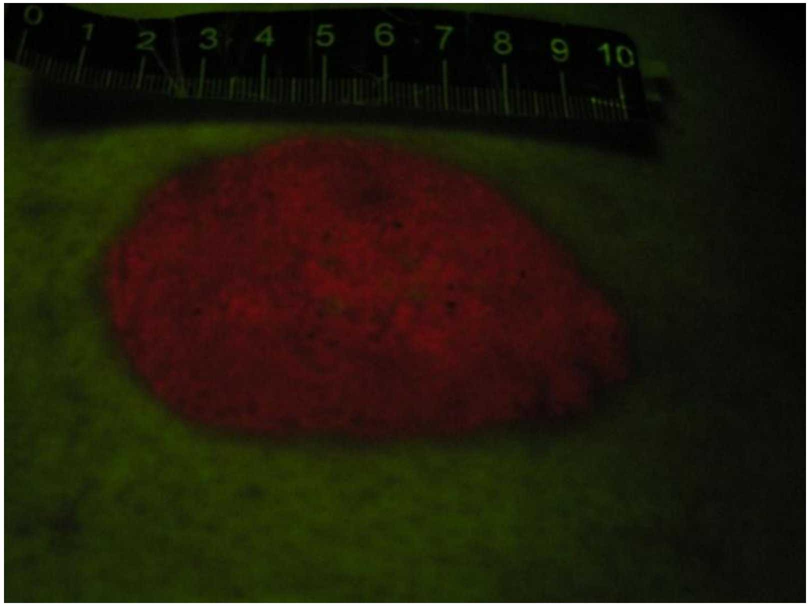
$$\mathbf{Y} = \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^T\ \mathbf{I}_D - 1/D\ \mathbf{1}_D\mathbf{1}_D^T\ \ \psi\ \mathbf{X} - 1/D\ \psi\ \mathbf{V}\ \mathbf{1}_D\mathbf{1}_T^T$$

Thus, **Y** contains decorrelated components. By applying some linear ICA algorithm on Y we obtain nonlinear ICA of **X**.

Nonlinear projection on orthonormal basis and ICA?

We have applied “temporal predictability” ICA algorithm, ref. a, on RKHS-deccorelated version of the low-intensity fluorescent RGB image of basal cell carcinoma.

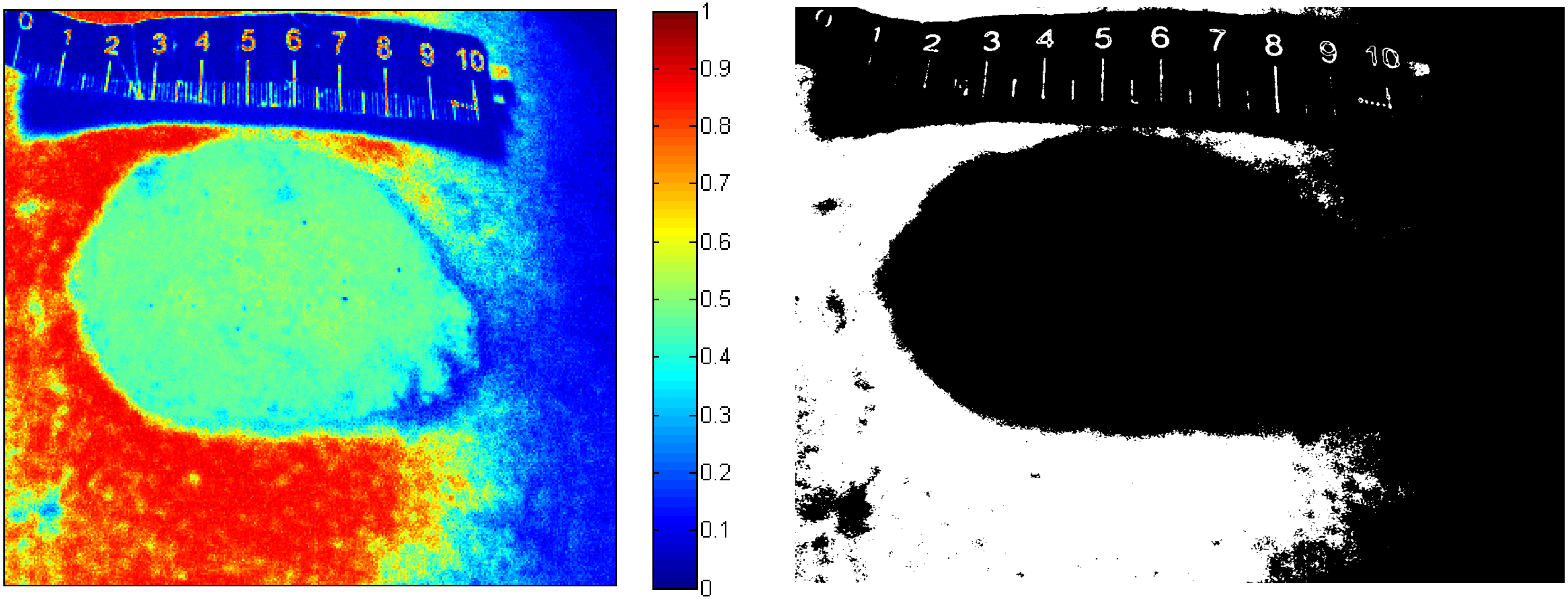




Left: high-intensity fluorescent RGB image of BCC. **Right**: low-intensity fluorescent image of BCC.

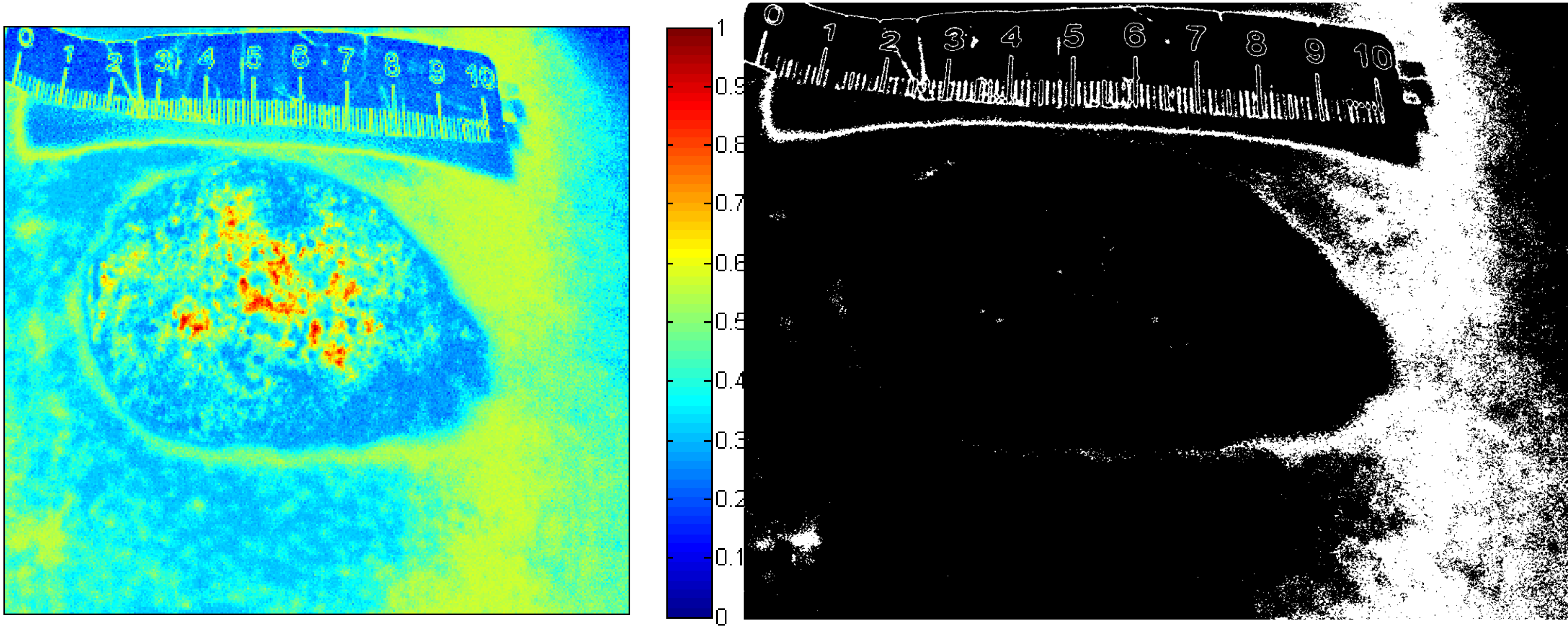
a) Stone, J.V., 2001. Blind source separation using temporal predictability. Neural Comput. 13, 1559-1574.

Nonlinear projection on orthonormal basis and ICA?



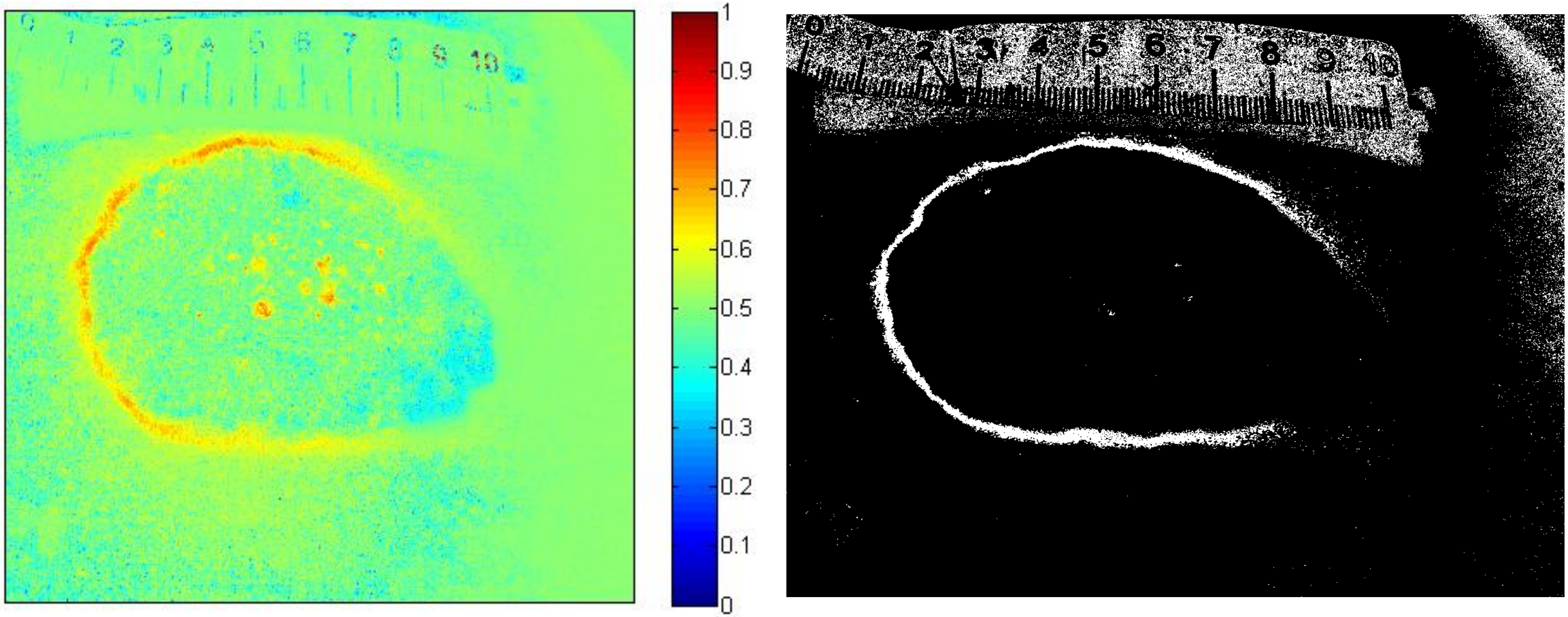
Background and ruler numbers. Left: Nolninear ICA extarcted component; Right: binarized version.

Nonlinear projection on orthonormal basis and ICA?



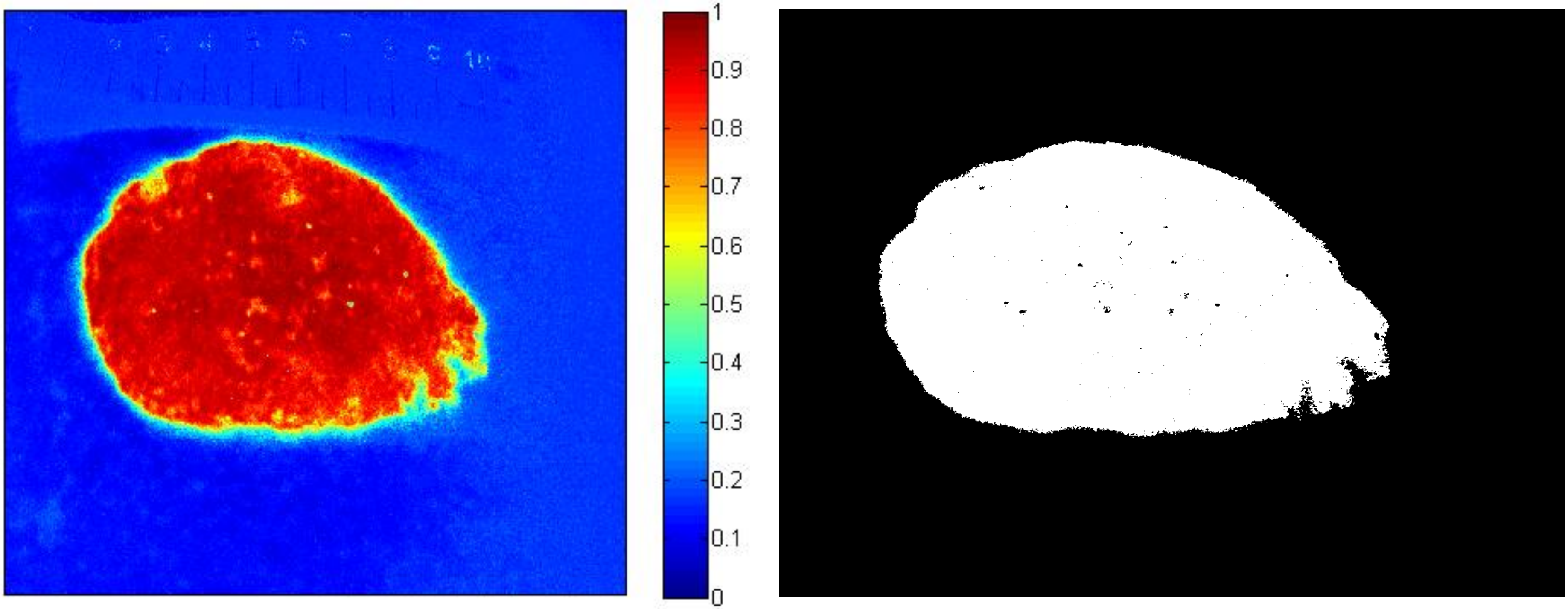
Background and ruler numbers. Left: Nolninear ICA extarcted component; Right: binarized version.

Nonlinear projection on orthonormal basis and ICA?



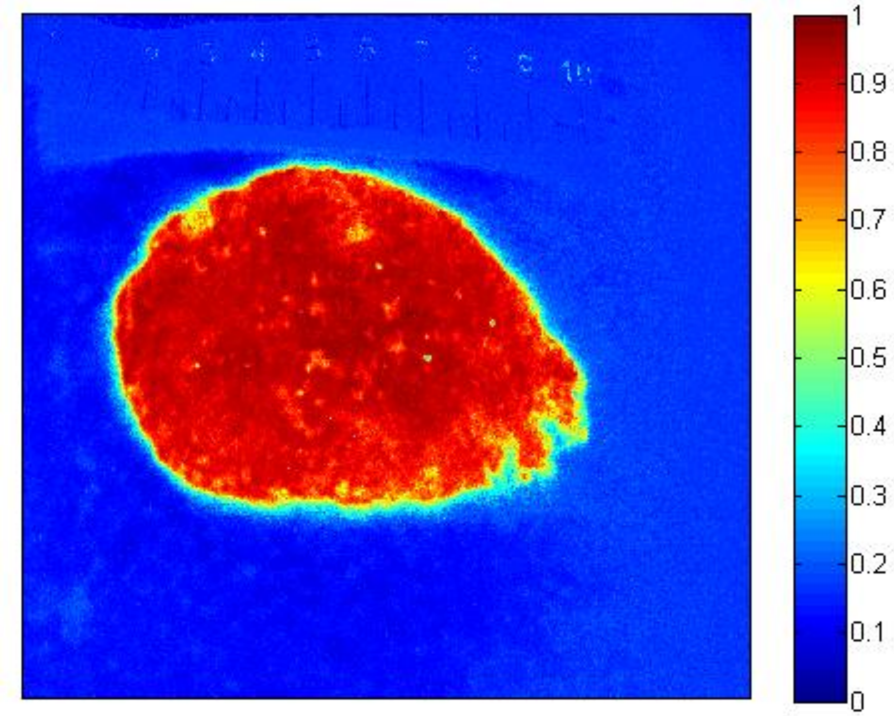
Tumor demarcation line and ruler body. Left: Nolninear ICA extarcted component; Right: binarized version.

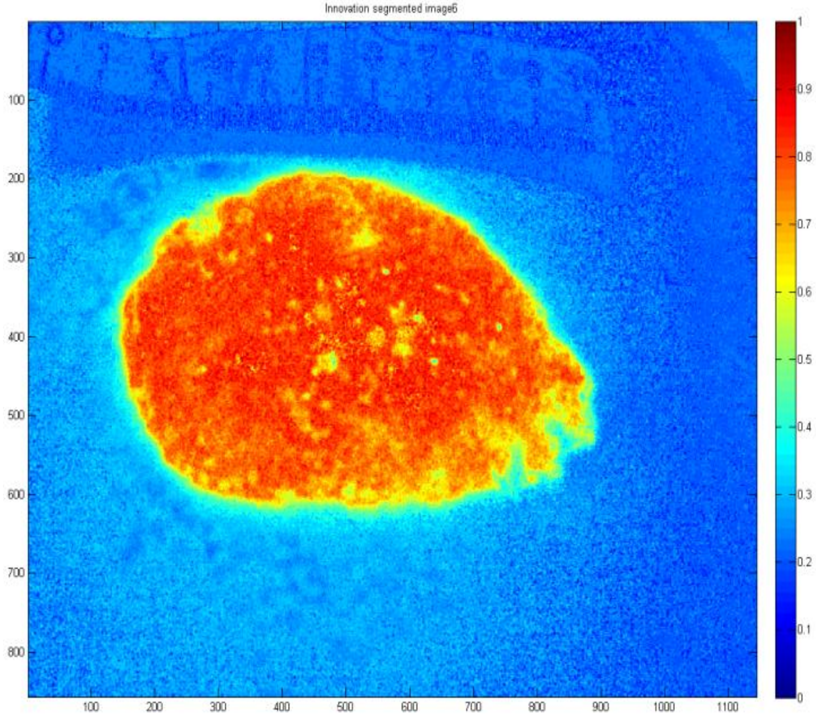
Nonlinear projection on orthonormal basis and ICA?



Basal cell carcinoma. Left: Nolninear ICA extarcted component. Right: binarized version.

Nonlinear projection on orthonormal basis and ICA?



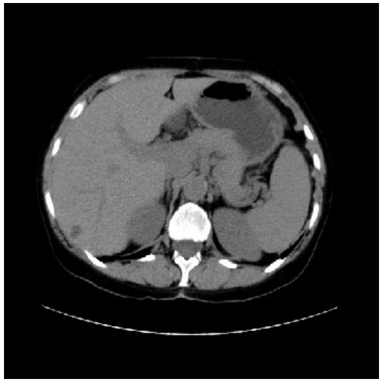


I. Kopriva**,** A. Peršin (2009) *Medical Image Analysis* , vol.**13**, pp.507-518.

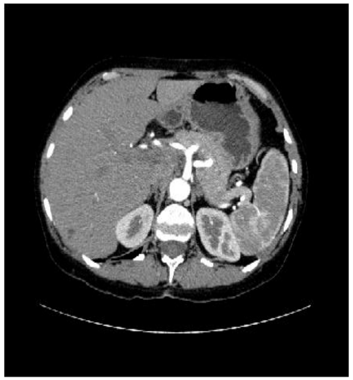
Nonlinear projection on orthonormal basis and ICA?

We have applied “AMUSE” ICA algorithm, ref. a, on RKHS-deccorelated version of the multi-phase CT of abdomen. (D=100, Gaussian kernel, $\sigma^2=10^3$.

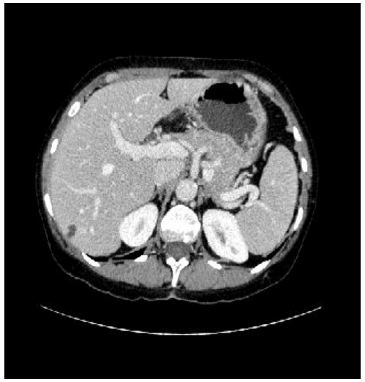
Slide 114 displayed in window [-100 200] Hounsfield unit.



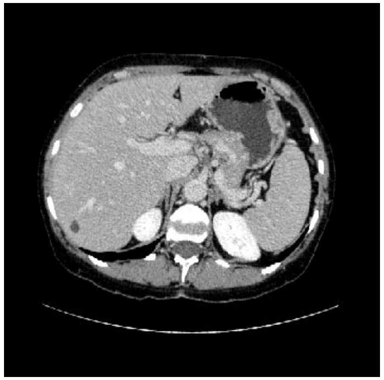
Non-contrast



Arterial



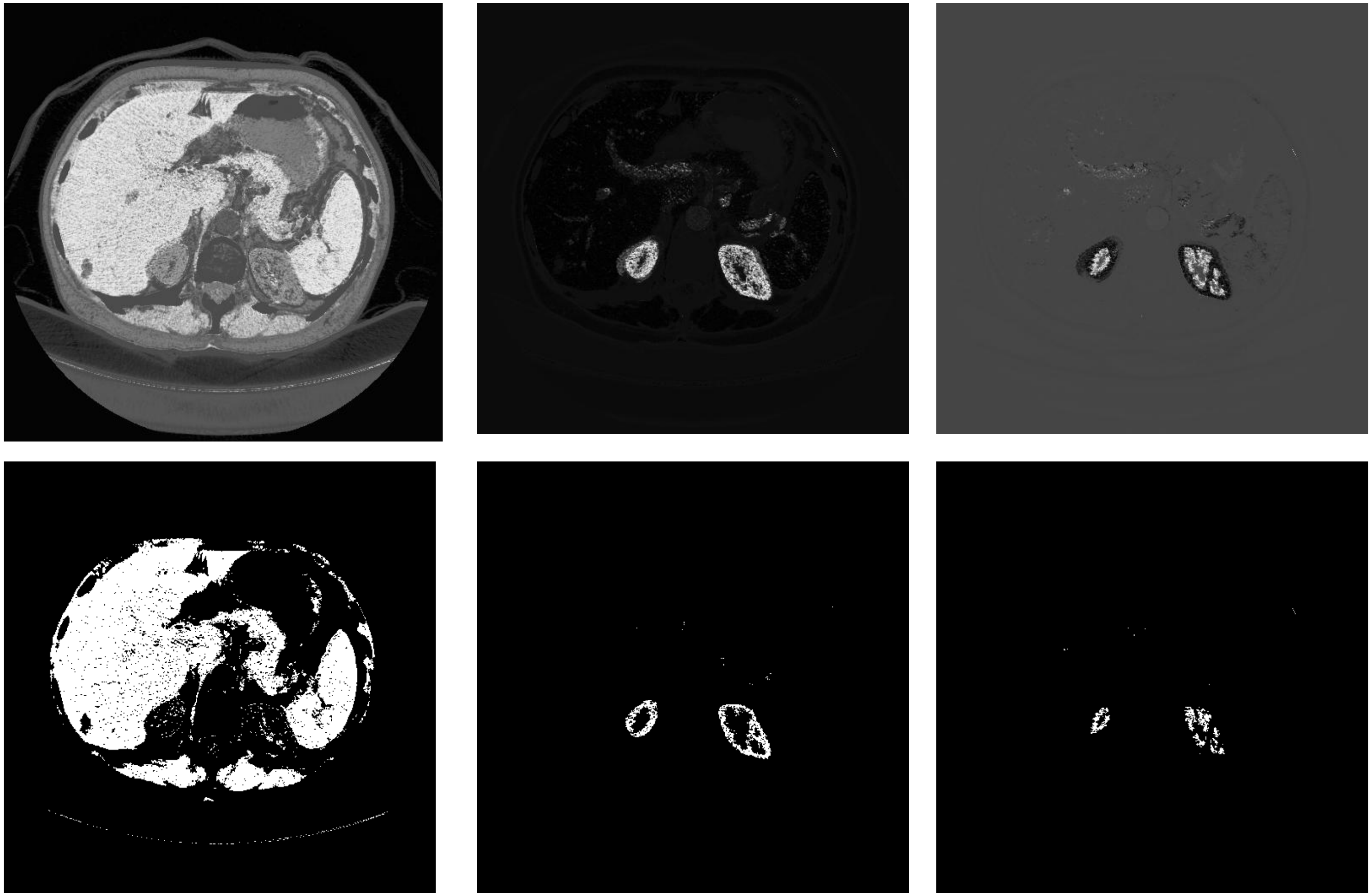
Venous 1



Venous 2

a) L. Tong, R.W. Liu, V.C. Soon, and Y. F. Huang, “Indeterminacy and identifiability of blind identification,” *IEEE Trans. on Circuits and Systems,* 38:499-509, 1991.

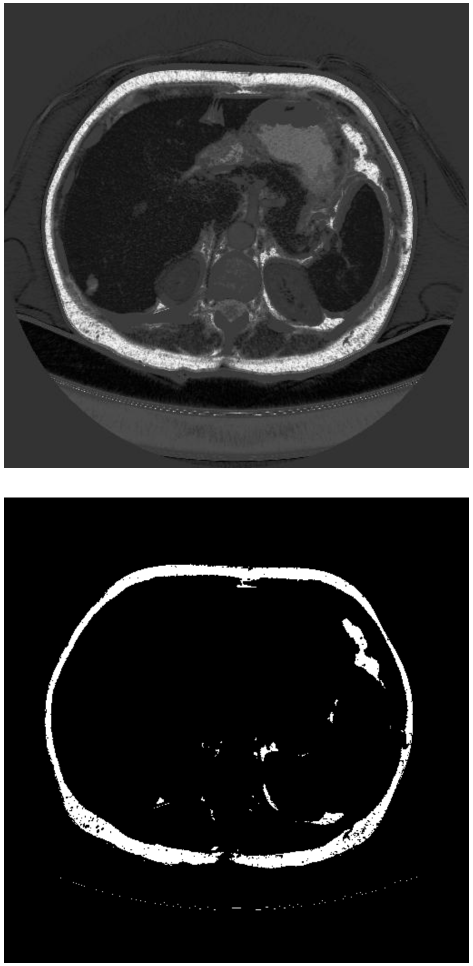
Top row:components decomposed by nonlinear ICA algorithm.
Bottom row: assignment according to maximal value criterion (occlusions).



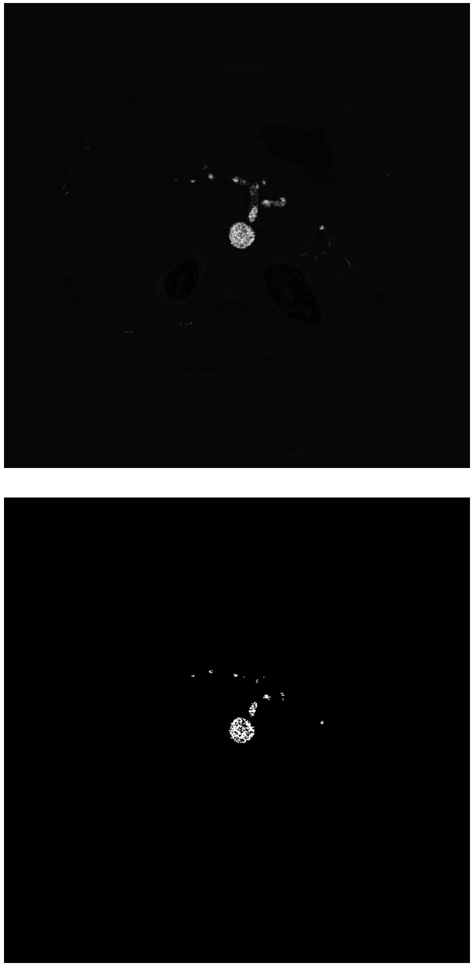
Liver

Kidneys – renal cortex

Kidneys – renal medula



skin



aorta

Nonlinear projection on orthonormal basis and ICA?

Nonlinear ICA can, **possibly**, be applied on unsupervised decomposition (segmentation) of multichannel medical images with **good spatial resolution and low-sensitivity** (contrast).

One imaging modaility of immediate relevance is multi-phase computed tomography (CT) imaging, where soft tissues (liver, kidneys, …) have low contrast.

In particular, nonlinear ICA can be used **to reduce number of phase-contrast** images!!!

Nonlinear ICA can also be used on **multimodal images** (e.g. CT and PET)**?**

Nonlinear decomposition of RGB image of unstained specimen in histopatology

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 histopatology

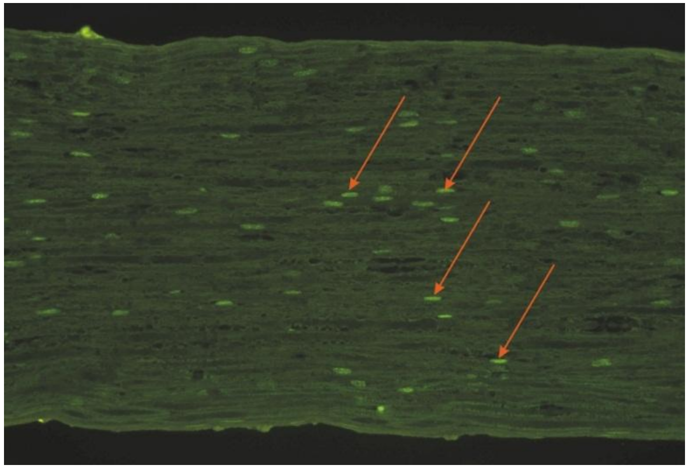
Decomposition of objects with spectrally similar profiles is hard problem. That occurs due to poor spectral resolution or due to physiological reasons.

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

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

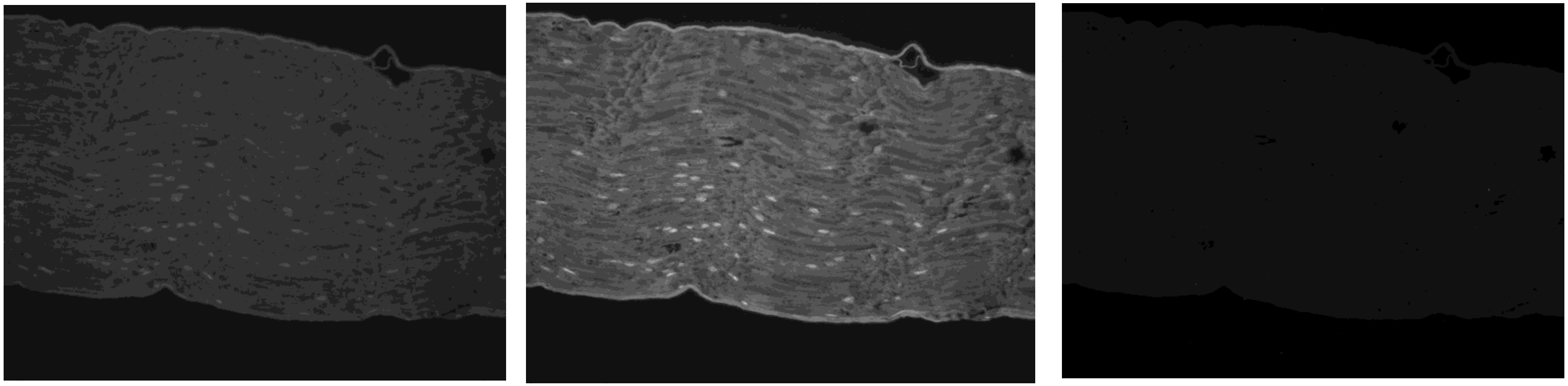
RGB image is first mapped nonlinearly (by means of explicit feature maps, a.k.a. rational variety mapping - RVM). Afterwards, sparseness constrained NMF is executed in induced space.

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

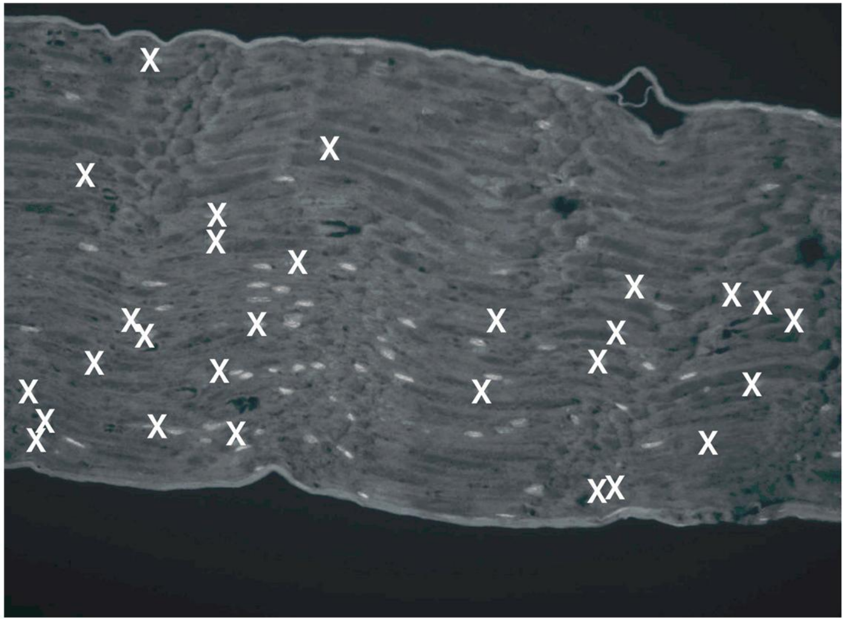
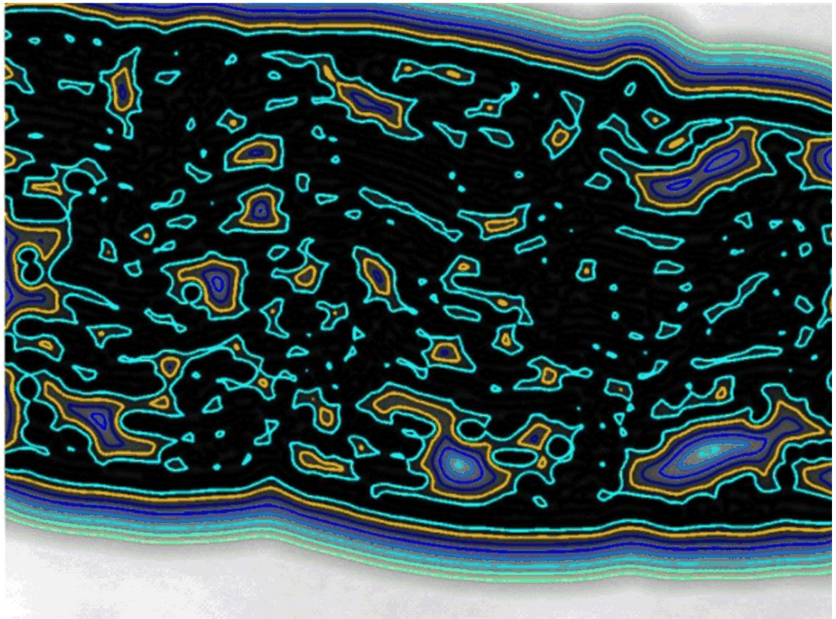
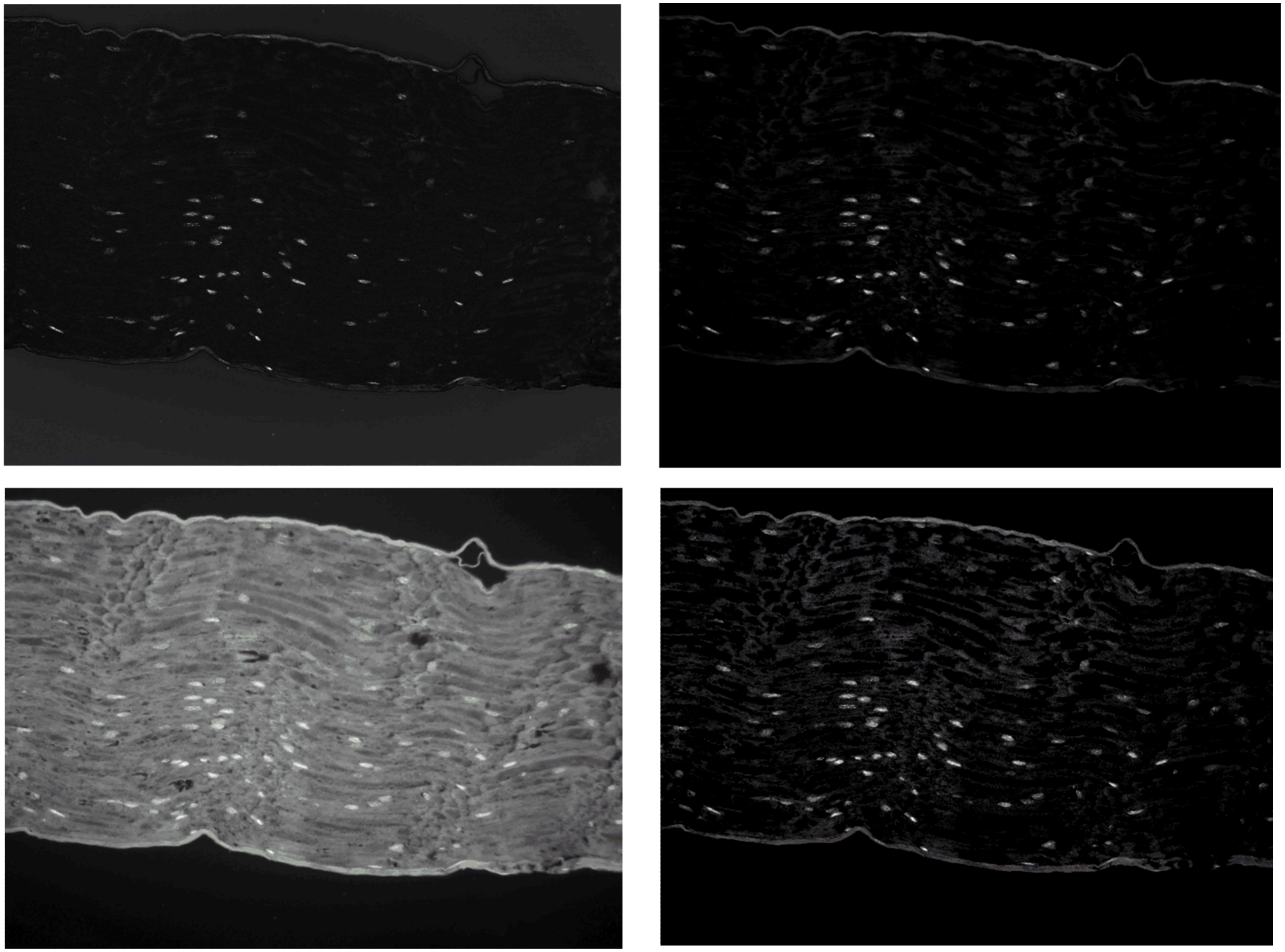


Image of unstained specimen at 510 nm wavelength (green color). White crosses denote false positive spots.



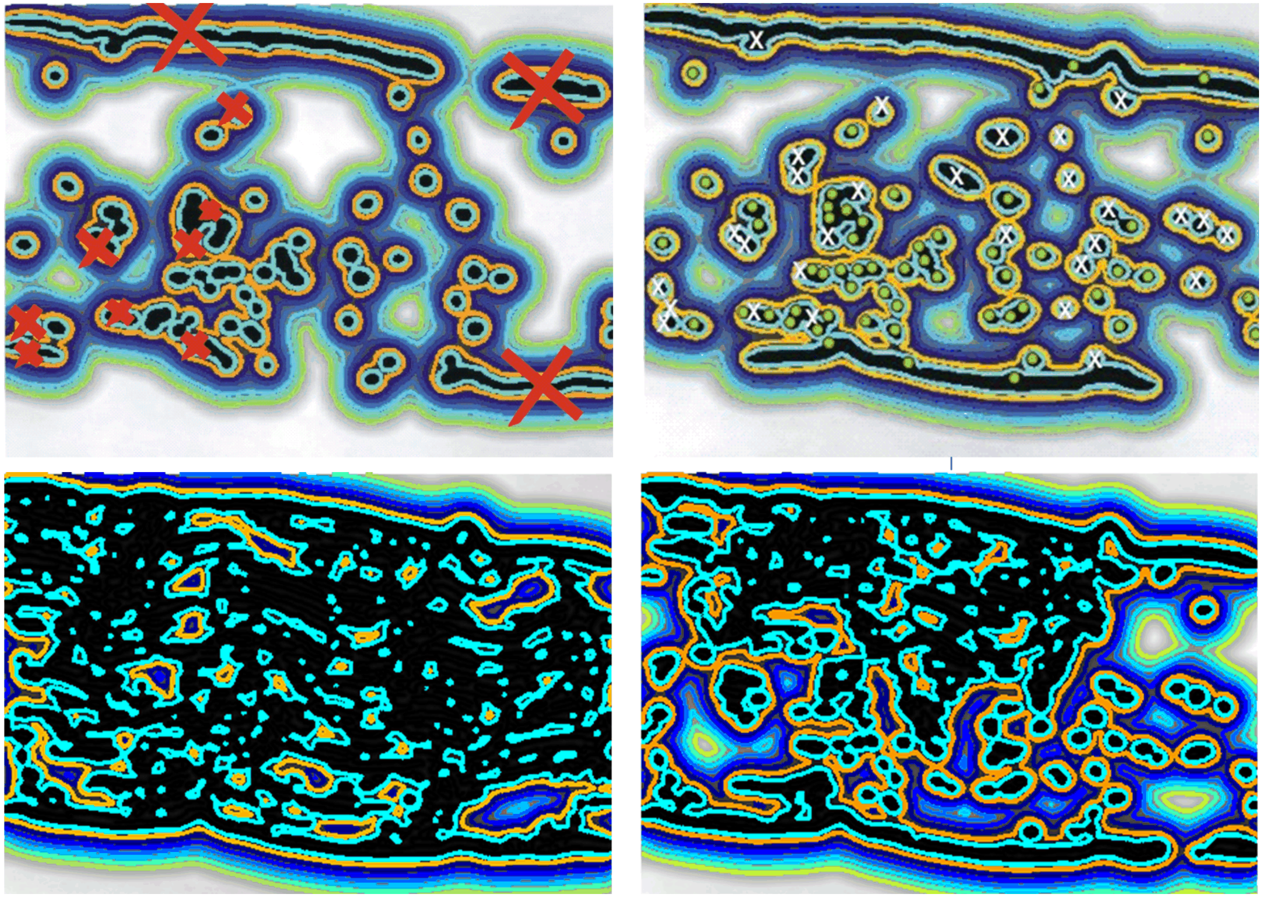
Active contours



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

Top right: RVM3 and NMU decompsoition; 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 decompsoition; 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:

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

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

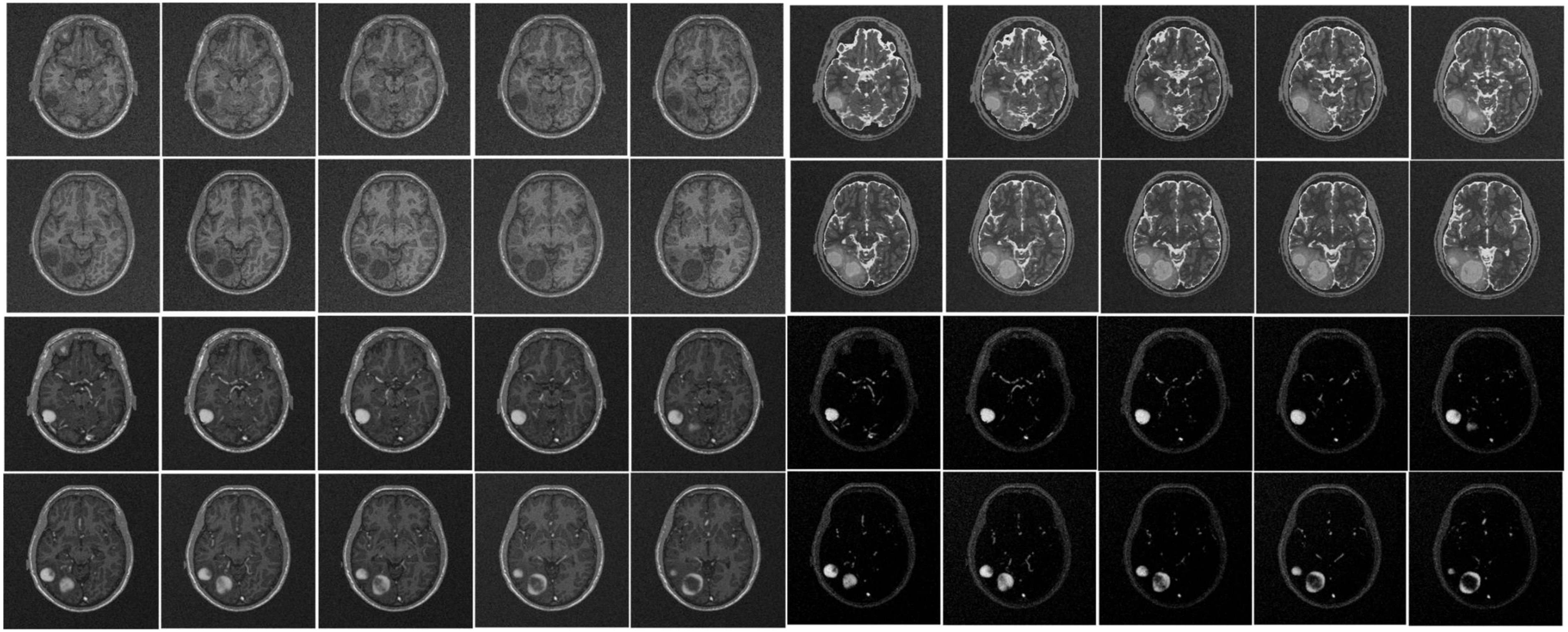
3D decomposition method is demonstrated on extraction of brain tumor from synthetic mMR image. The image is obtained from TumorSim database of the Utah Center for Neuroimage Analysis, [a].

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 256×256 pixels.

a) The TumorSim database of Utah Center for Neuroimage Analysis: **http://www.nitrc.org/projects/tumorsim/**.

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

Slice number	50	51	52	53	54	55	56	57	58	59	60
3D Segmentation	**0.7278**	**0.7679**	**0.8387**	**0.8669**	**0.8634**	**0.8512**	**0.8748**	**0.8875**	**0.8876**	**0.8938**	**0.7811**
T1_GAD image	0.1942	0.2280	0.2565	0.2836	0.2940	0.3193	0.3388	0.3626	0.3754	0.3583	0.3536

Slice number	61	62	63	64	65	66	67	68	69	70
3D Segmentation	**0.7436**	**0.7587**	**0.7061**	**0.7699**	**0.7223**	**0.5672**	**0.5635**	**0.4799**	**0.4060**	**0.4113**
T1_GAD image	0.3810	0.4137	0.4415	0.4343	0.4221	0.3619	0.3287	0.2851	0.2431	2158

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 (ovefitting).

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 confimed 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 confimed 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.

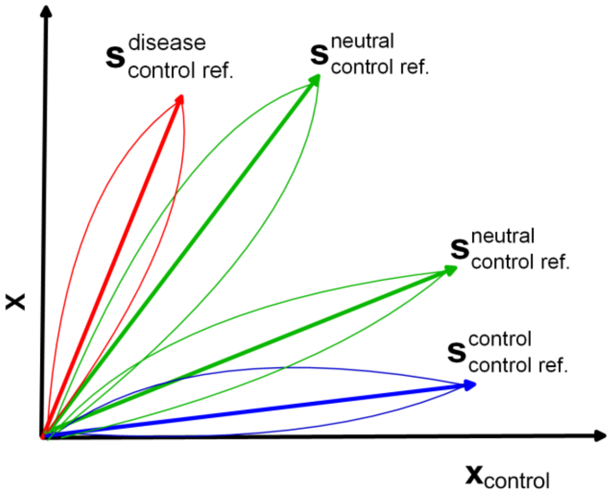
$$\begin{bmatrix} \mathbf{x}_{\text{control}} \\ \mathbf{x} \end{bmatrix} = \mathbf{A}_{\text{control}}\mathbf{S}_{\text{control}} : \ \mathbf{X} \in R^{2\times T}, \mathbf{A}_{\text{control}} \in R^{2\times M}, \mathbf{S}_{\text{control}} \in R^{M\times T} \ \ M \geq 2$$

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

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.

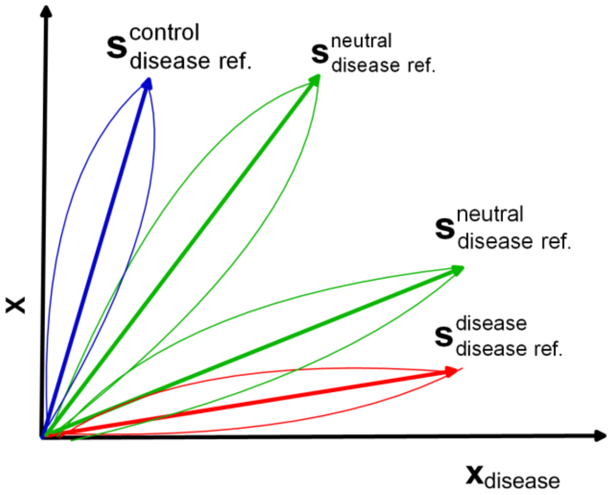
Linear mixture model

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


$s^{disease}_{control\ ref.}$
$s^{neutral}_{control\ ref.}$
$s^{neutral}_{control\ ref.}$
$s^{control}_{control\ ref.}$
x
$x_{control}$

Geometry of linear mixture model

with control reference

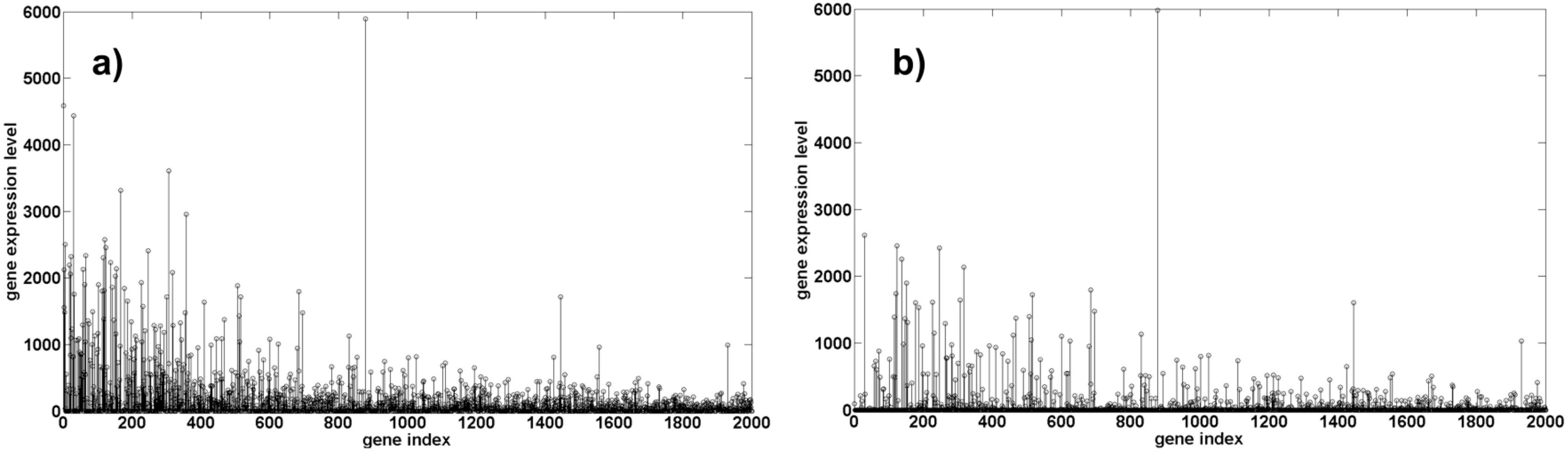

$s^{control}_{disease\ ref.}$
$s^{neutral}_{disease\ ref.}$
$s^{neutral}_{disease\ ref.}$
$s^{disease}_{disease\ ref.}$
x
$x_{disease}$

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 with2 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.

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

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 **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 **S** that are influenced by noise.

Sparse component analysis algorithm

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

In this case optimization problems become quadratic programms. Hence gradient descent with projection onto non-negative orthant: max(**0,S**) .

Regularization parameter λ is chosen by cross-validation as $\lambda=c\times\lambda_{max}$ where λ_{max} denotes value at which **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 **A**) is estimated by cross-validation as well.

Sparse component analysis algorithm - outline

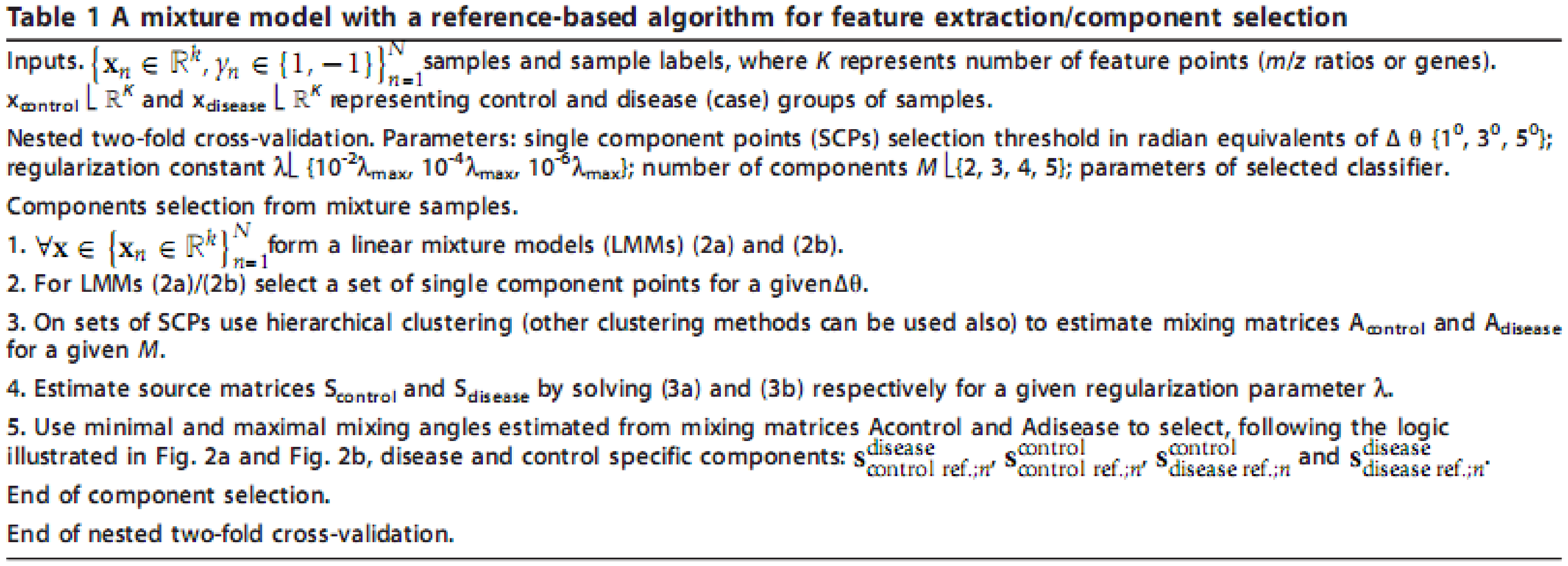


Table 1 A mixture model with a reference-based algorithm for feature extraction/component selection

Inputs. $\left\{\mathbf{x}_n \in \mathbb{R}^k, y_n \in \{1,-1\}\right\}_{n=1}^{N}$ samples and sample labels, where *K* represents number of feature points (*m/z* ratios or genes). $\mathrm{x}_{\mathrm{control}} \lfloor \mathbb{R}^K$ and $\mathrm{x}_{\mathrm{disease}} \lfloor \mathbb{R}^K$ representing control and disease (case) groups of samples.

Nested two-fold cross-validation. Parameters: single component points (SCPs) selection threshold in radian equivalents of $\Delta\,\theta$ $\{1^0, 3^0, 5^0\}$; regularization constant $\lambda \lfloor \{10^{-2}\lambda_{\max}, 10^{-4}\lambda_{\max}, 10^{-6}\lambda_{\max}\}$; number of components *M* $\lfloor\{2, 3, 4, 5\}$; parameters of selected classifier.

Components selection from mixture samples.

1. $\forall \mathbf{x} \in \left\{\mathbf{x}_n \in \mathbb{R}^k\right\}_{n=1}^{N}$ 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 $\mathrm{A}_{\mathrm{control}}$ and $\mathrm{A}_{\mathrm{disease}}$ for a given *M*.

4. Estimate source matrices $\mathrm{S}_{\mathrm{control}}$ and $\mathrm{S}_{\mathrm{disease}}$ by solving (3a) and (3b) respectively for a given regularization parameter λ.

5. Use minimal and maximal mixing angles estimated from mixing matrices Acontrol and Adisease to select, following the logic illustrated in Fig. 2a and Fig. 2b, disease and control specific components: $\mathbf{s}^{\text{disease}}_{\text{control ref.};n}$, $\mathbf{s}^{\text{control}}_{\text{control ref.};n}$, $\mathbf{s}^{\text{control}}_{\text{disease ref.};n}$ and $\mathbf{s}^{\text{disease}}_{\text{disease ref.};n}$.

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

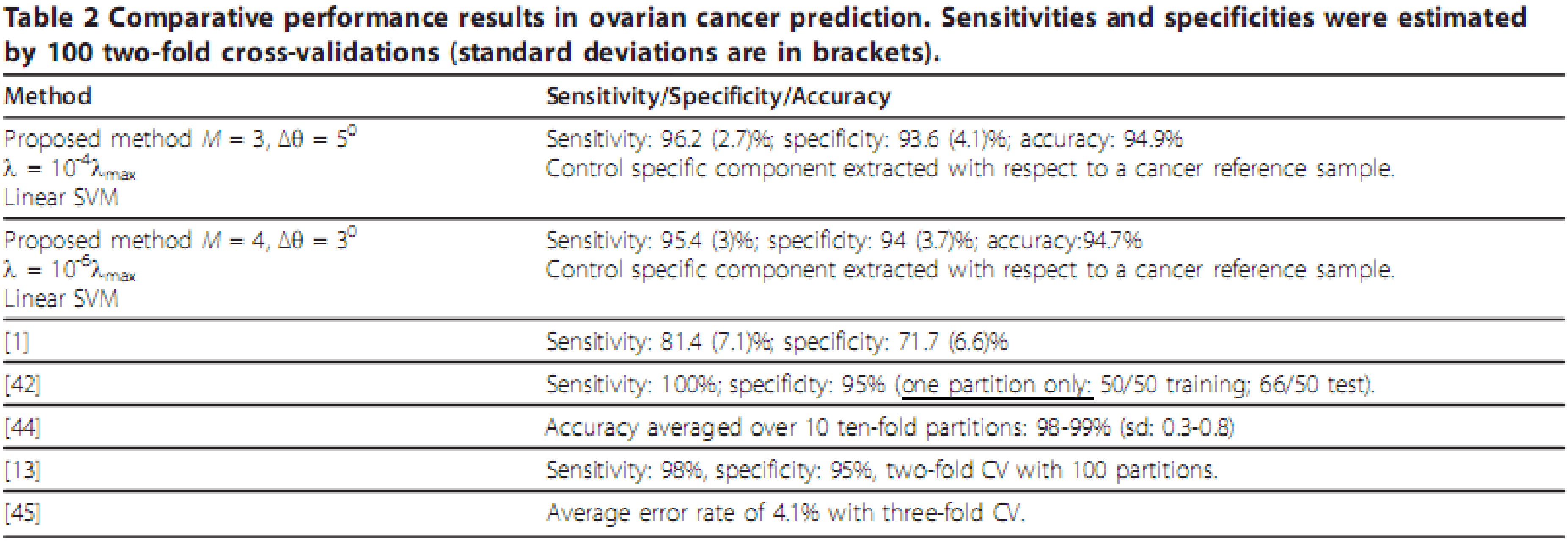


Table 2 Comparative performance results in ovarian cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).

Method	Sensitivity/Specificity/Accuracy
Proposed method $M = 3$, $\Delta\theta = 5^0$ $\lambda = 10^{-4}\lambda_{max}$ 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 $M = 4$, $\Delta\theta = 3^0$ $\lambda = 10^{-6}\lambda_{max}$ 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% (one partition only: 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<PSA<10 and 43 with PSA>10, have been used:

Petricoin EF, Ornstein DK, Paweletz CP, Ardekani A, Hackett PS, Hitt BA, Velassco 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

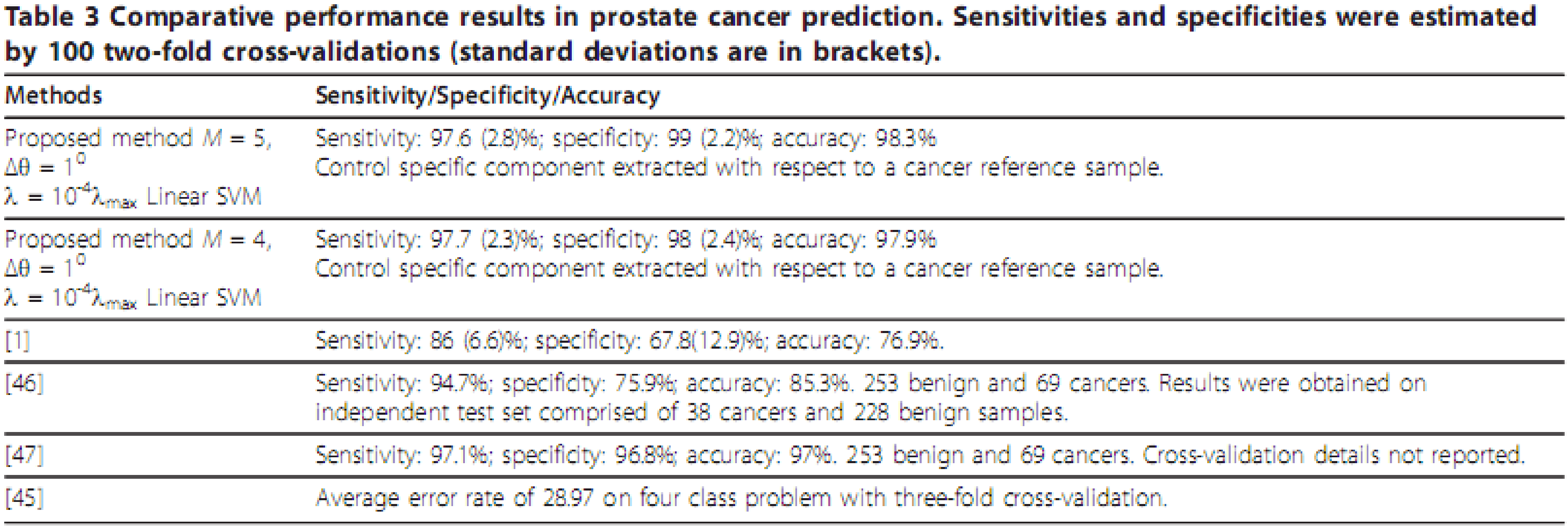


Table 3 Comparative performance results in prostate cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).

Methods	Sensitivity/Specificity/Accuracy
Proposed method *M* = 5, $\Delta\theta = 1^{0}$ $\lambda = 10^{-4}\lambda_{max}$ 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, $\Delta\theta = 1^{0}$ $\lambda = 10^{-4}\lambda_{max}$ 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 obatined 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 reatined 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

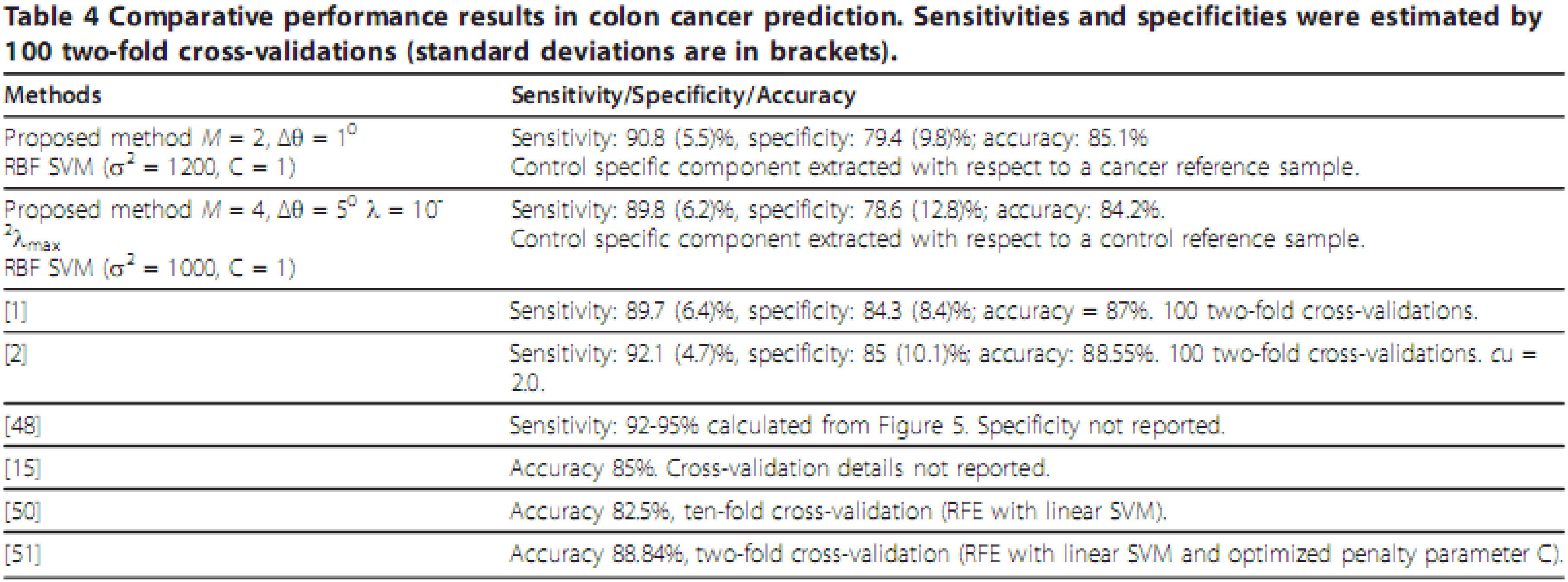


Table 4 Comparative performance results in colon cancer prediction. Sensitivities and specificities were estimated by 100 two-fold cross-validations (standard deviations are in brackets).

Methods	Sensitivity/Specificity/Accuracy
Proposed method $M = 2$, $\Delta\theta = 1^{0}$ 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 $M = 4$, $\Delta\theta = 5^{0}$ $\lambda = 10^{-2}\lambda_{max}$ 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. cu = 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).

THANK YOU !!!!!!!!