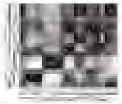
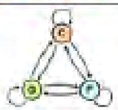
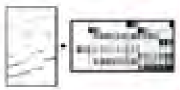
10420CS 573100
音樂資訊檢索
Music Information Retrieval

Lecture 9
Source Separation

Yi-Hsuan Yang Ph.D.
http://www.citi.sinica.edu.tw/pages/yang/
yang@citi.sinica.edu.tw

Music & Audio Computing Lab,
Research Center for IT Innovation,
Academia Sinica

Reference

Chapter		Music Processing Scenario
1		Music Represenations
2		Fourier Analysis of Signals
3		Music Synchronization
4		Music Structure Analysis
5		Chord Recognition
6		Tempo and Beat Tracking
7		Content-Based Audio Retrieval
8		Musically Informed Audio Decomposition

Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

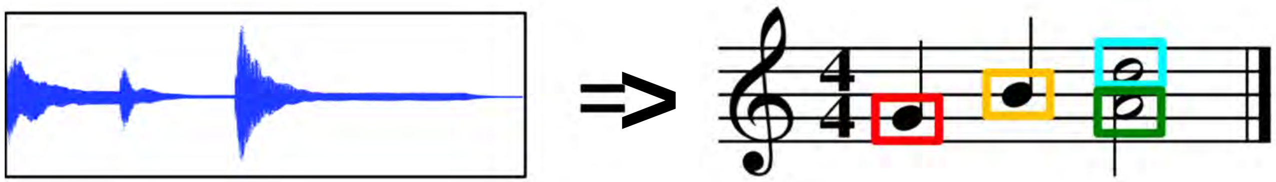
Accompanying website:
www.music-processing.de

Why Source Separation

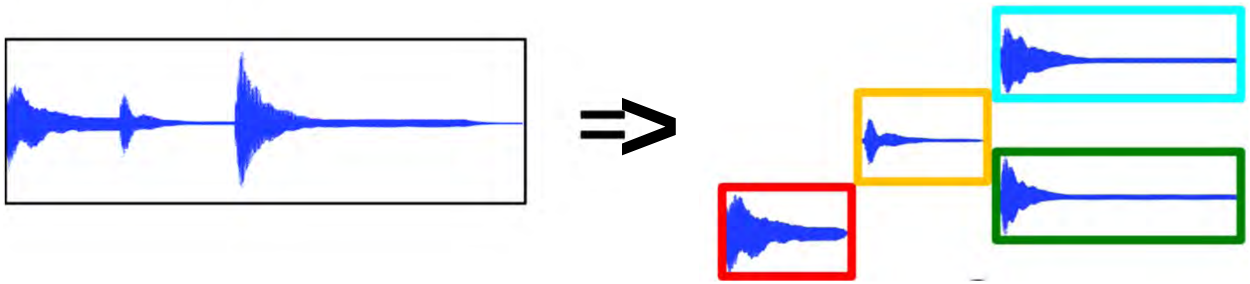
- Because we are obsessed with this topic …
  - “Complex and quaternionic principal component pursuit and its application to audio separation,” SPL 2016
  - “Informed monaural source separation of music based on convolutional sparse coding,” ICASSP 2015
  - “Vocal activity informed singing voice separation with the IKALA dataset,” ICASSP 2015
  - “Sparse modeling for artist identification: Exploiting phase information and vocal separation ,” ISMIR 2013
  - “Low-rank representation of both singing voice and music accompaniment via learned dictionaries,” ISMIR 2013
  - “On sparse and low-rank matrix decomposition for singing voice separation,” ACM MM 2012

Why Source Separation

- The “two” holy grails in MIR
  - automatic transcription



  - source separation



Figures from [Mueller, FPM, Chapter 8, Springer 2015]

Application: Instrument Equalization

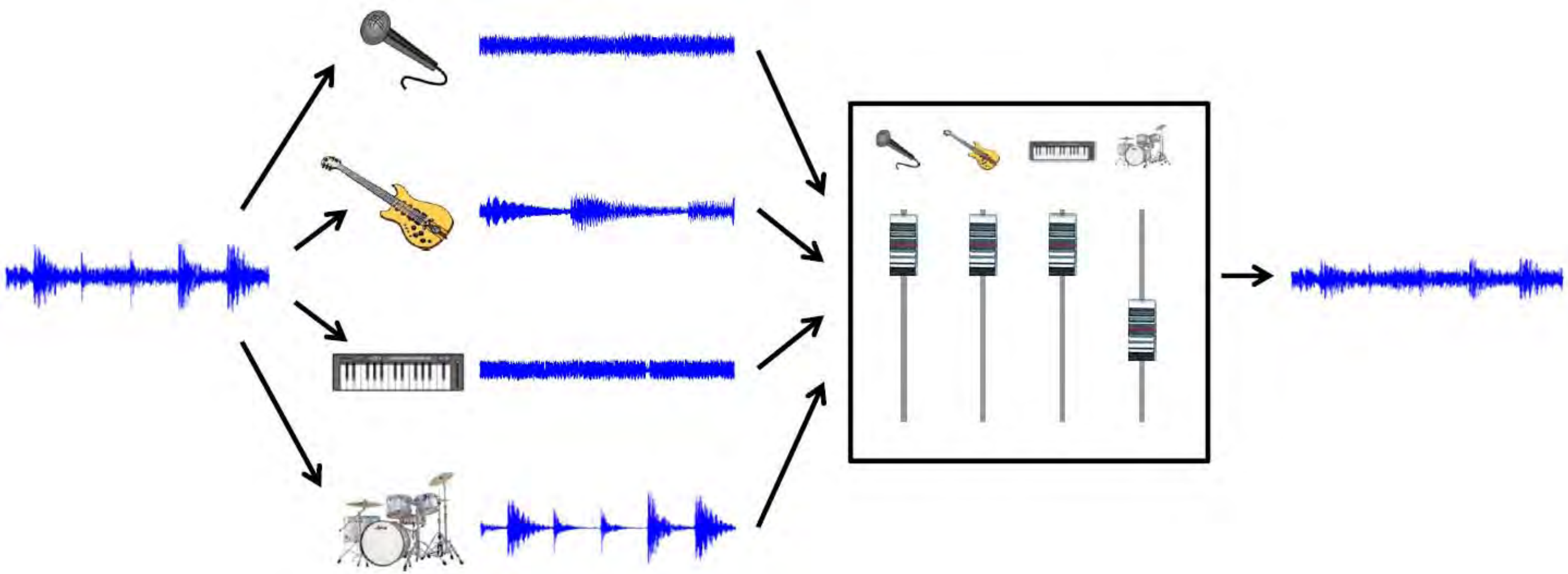


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Application: Instrument Equalization

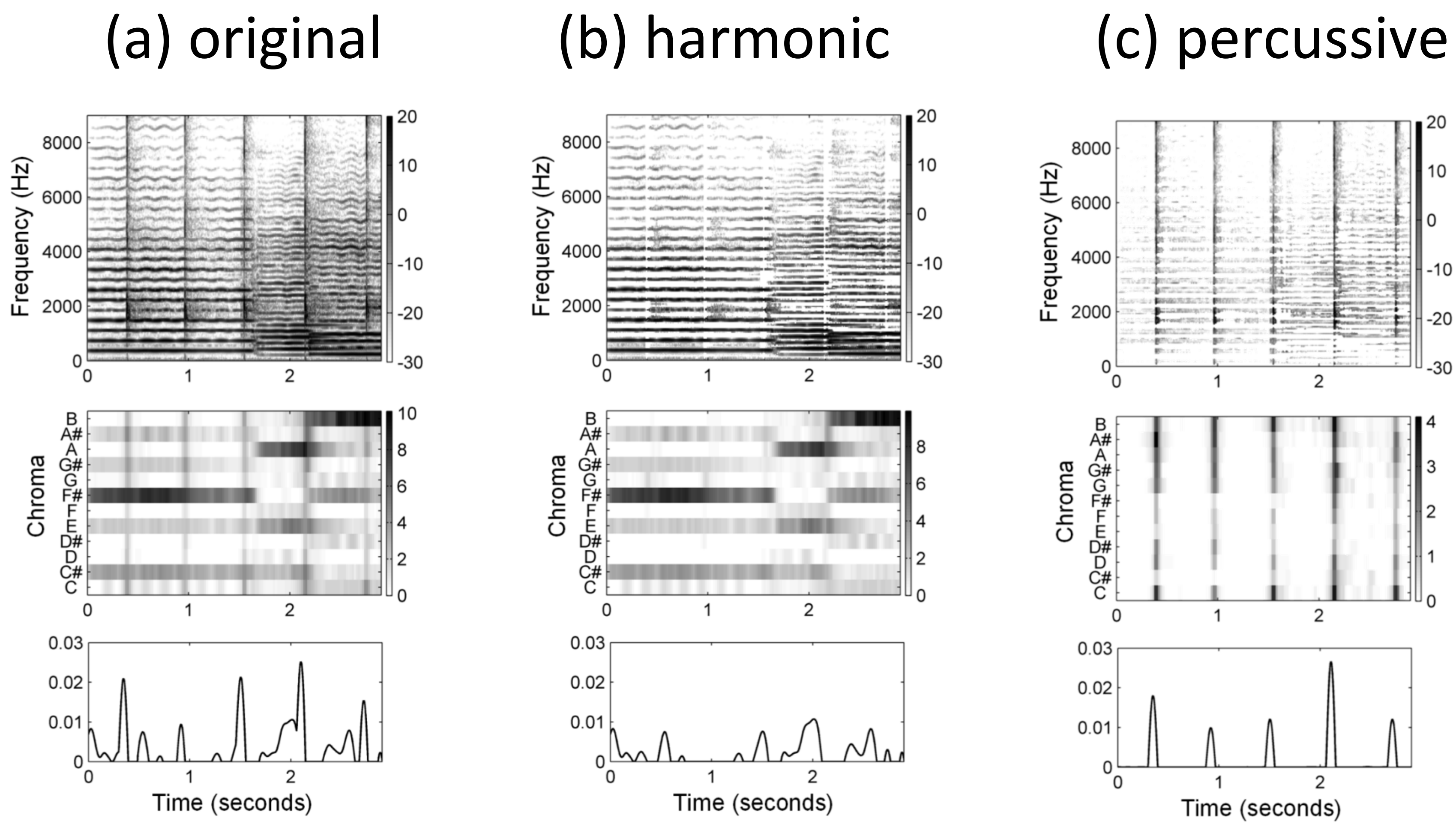


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Application: Audio Editing

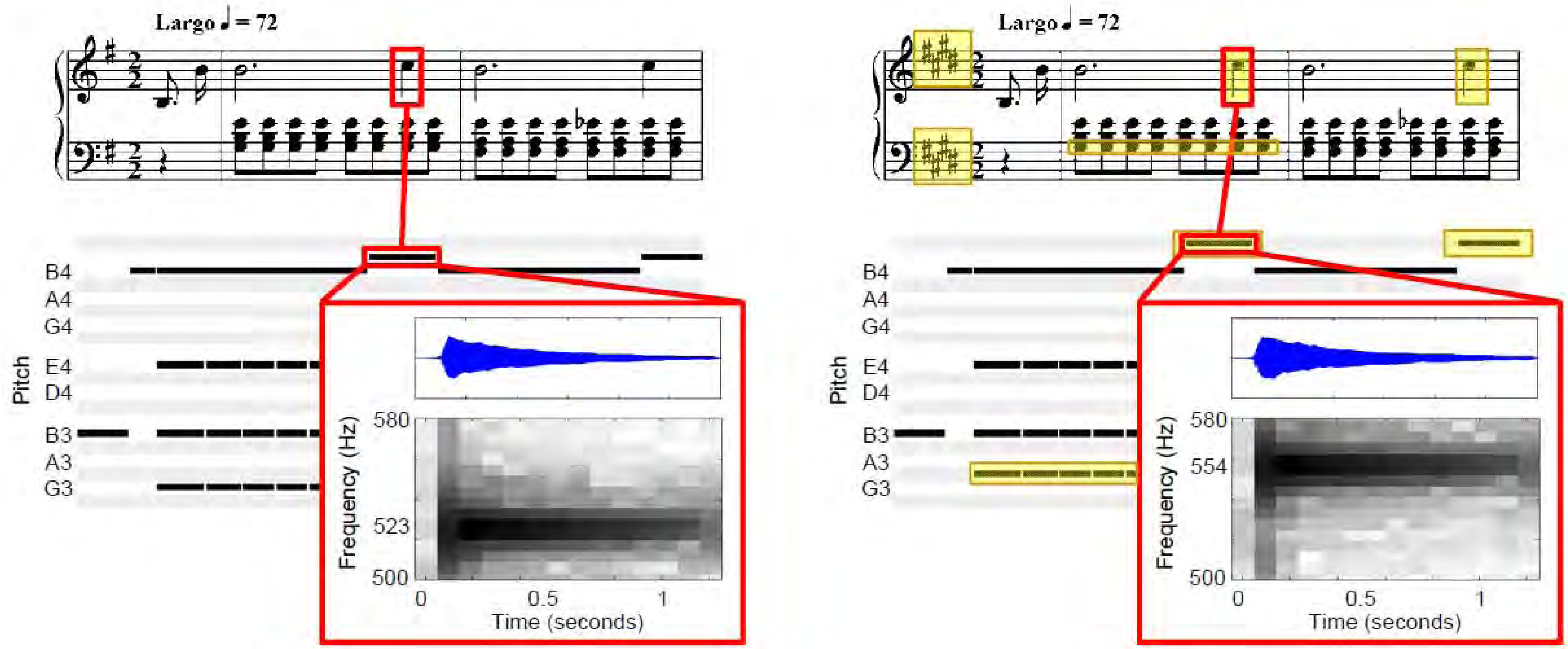
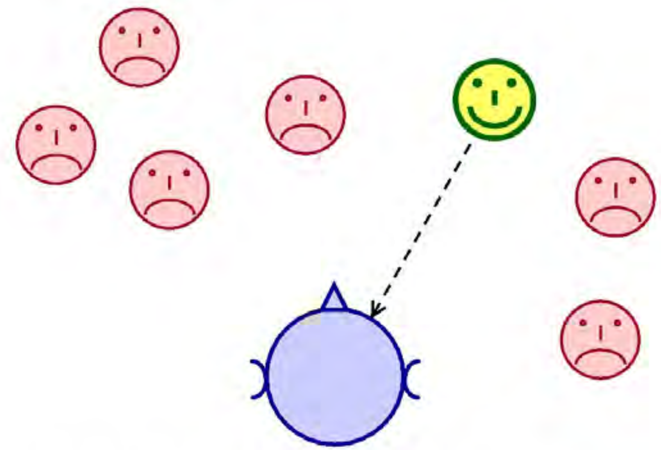


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Types of Separation Problems

- Type of sources
  - separating multiple speakers (a.k.a. *cocktail party effect*)

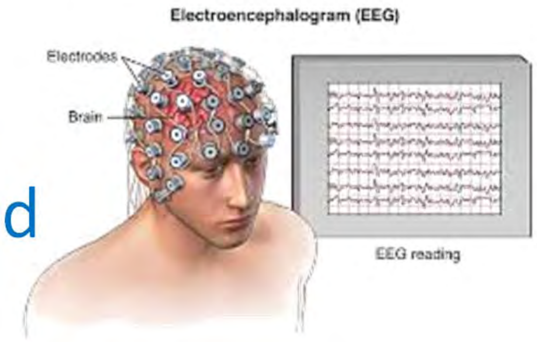


Cocktail-Party-Effekt:
Herausfiltern einer Schallquelle bei Anwesenheit mehrere Schallquellen

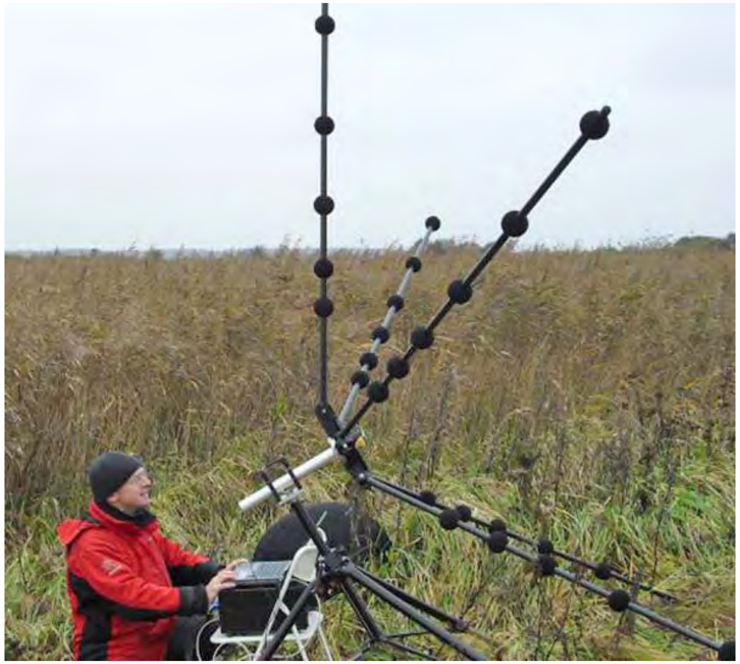
  - **W9**: separating **multiple instruments** (e.g., piano, violin)
  - **W10**: separating **harmonic/percussive** components
  - **W11**: separating **singing voice** from the accompaniments

Types of Separation Problems

- #sources vs. #channels
  - overdetermined vs underdetermined
  - single-channel vs. multi-channel

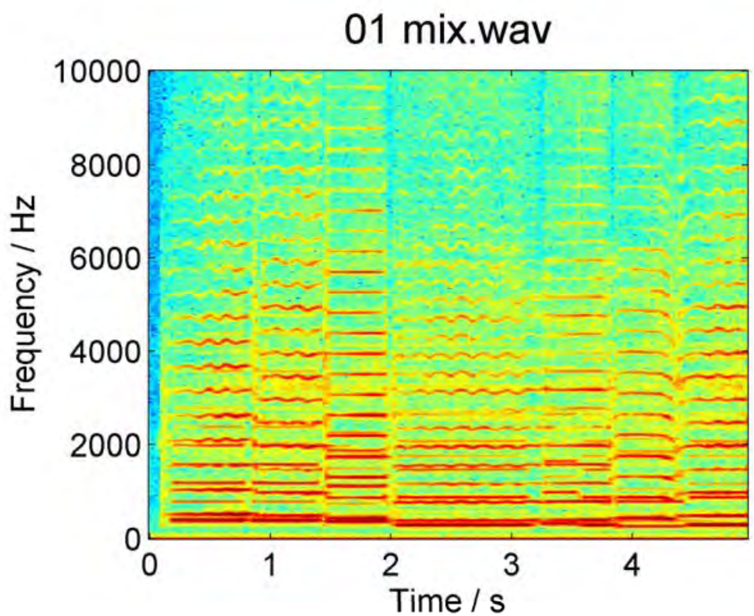


- Amount of side information
  - blind source separation vs. "guided" source separation
- Online or offline



Why Source Separation is Difficult?

- Harmonic overlaps + underdetermined

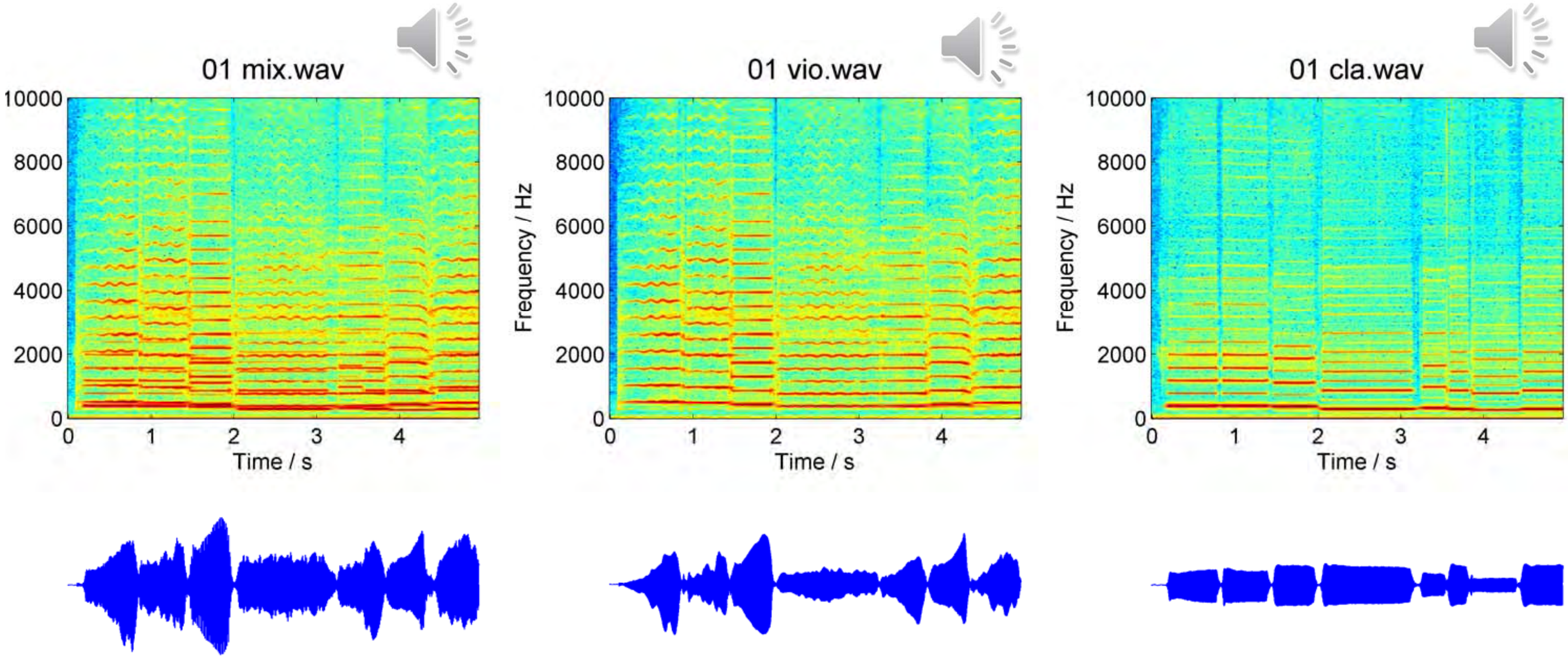


violin

clarinet

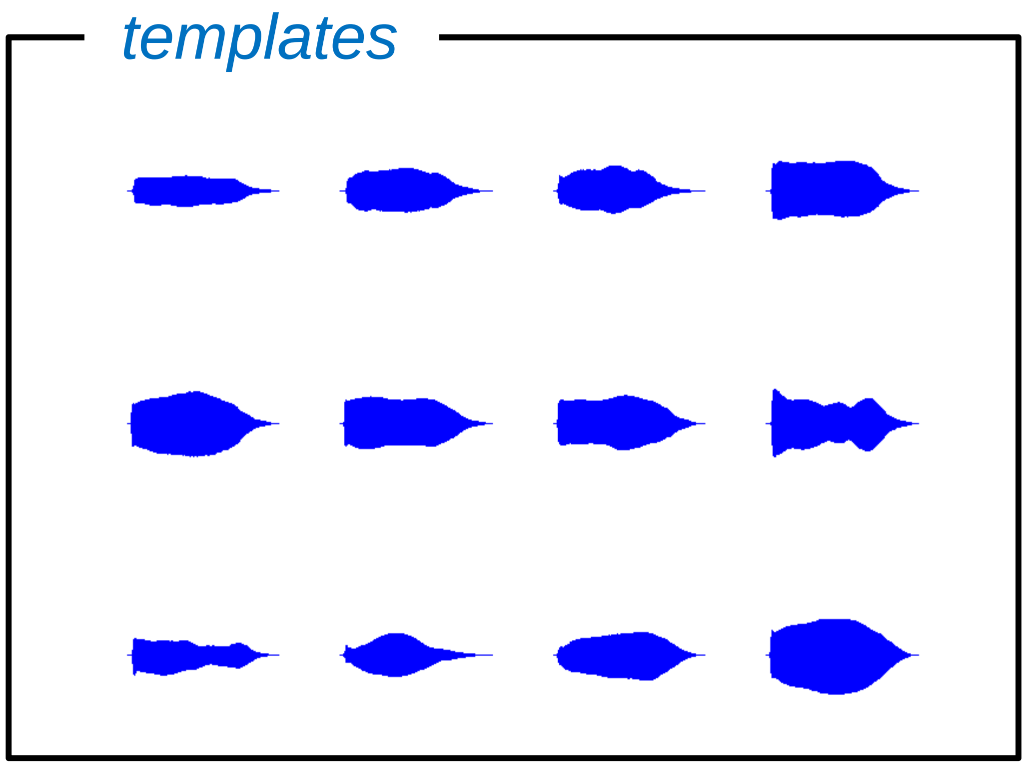
Why Source Separation is Difficult?

- Harmonic overlaps + underdetermined



Approach

- **Unsupervised**: rule-based
- **Supervised**: learn from “clean sources”



Approach

- **W9: multiple instruments separation**
  => *dictionary based* methods: nonnegative matrix factorization (NMF) and friends

- **W10: harmonic/percussive separation**
  => median filtering and friends

- **W11: singing voice separation**
  => *low-rank based* methods: robust principal component analysis (RPCA) and friends

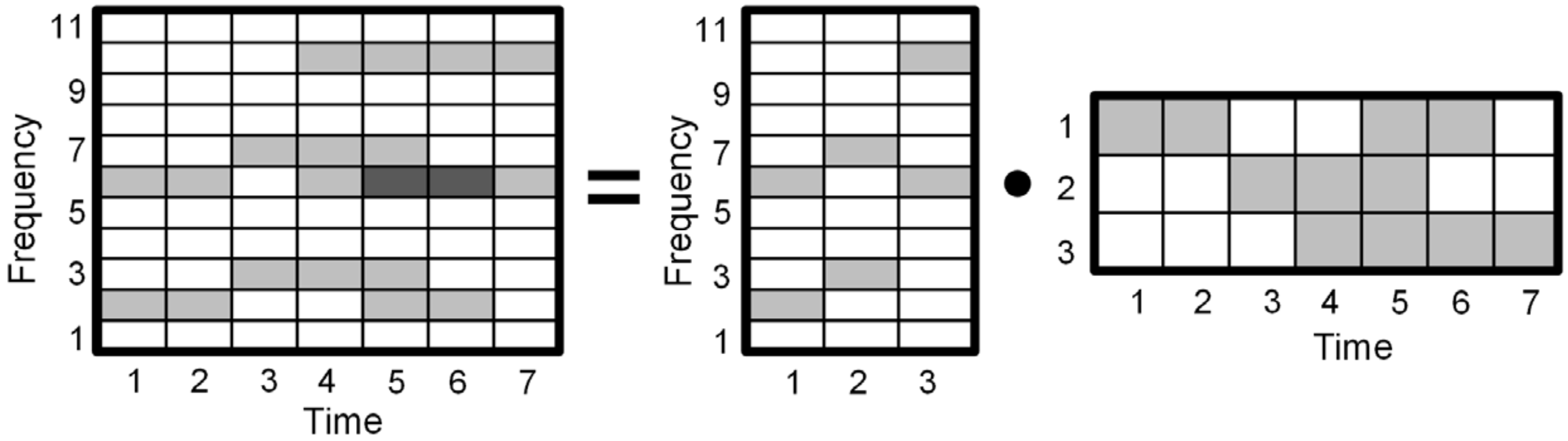
Nonnegative Matrix Factorization (NMF)

- Factorize (decompose) a matrix into two

Learning the parts of objects by non-negative matrix ...
www.nature.com › ... › Archive › Letters to Nature ▼ 翻譯這個網頁
由 DD Lee 著作 - 1999 - 被引用 6453 次 - 相關文章
Learning the parts of objects by non-negative matrix factorization. Daniel D. Lee & H. Sebastian Seung. Bell Laboratories, Lucent Technologies , Murray Hill, ...

[PDF] Algorithms for Non-negative Matrix Factorization - NIPS ...
papers.nips.cc/.../1861-algorithms-for-non-negative-matri... ▼ 翻譯這個網頁
由 DD Lee 著作 - 2001 - 被引用 4917 次 - 相關文章
Two different multi- plicative algorithms for NMF are analyzed. They differ only slightly in the multiplicative factor used in the update rules. One algorithm can be.

NMF: Basic Idea



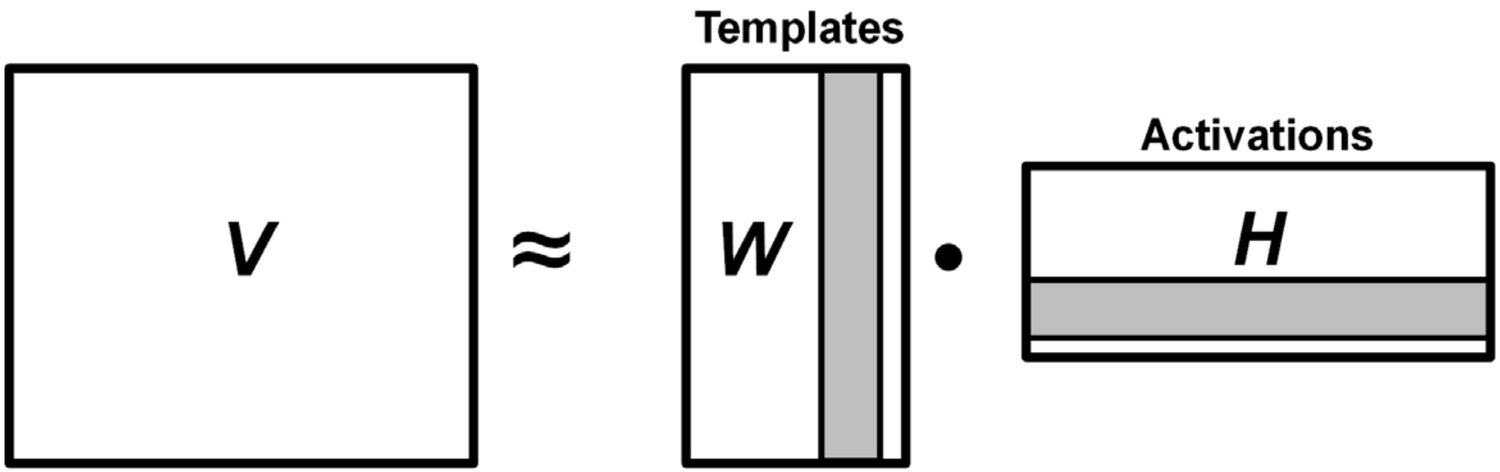


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF: Basic Idea

Given a *nonnegative* matrix $\mathbf{V}$ of dimensions $F \times N$, NMF is the problem of finding a factorization

$$\mathbf{V} \approx \mathbf{WH}$$

where $\mathbf{W}$ and $\mathbf{H}$ are *nonnegative* matrices of dimensions $F \times K$ and $K \times N$, respectively.

K is usually chosen such that $F\,K + K\,N << F\,N$, hence reducing the data dimension, but not always.

From Cédric Févotte's slides

NMF: Basic Idea

Along VQ, PCA or ICA, NMF provides an **unsupervised linear representation** of data

$\mathbf{v}_n$	$\approx$	$\mathbf{W}$	$\mathbf{h}_n$
data vector		"explanatory variables"	"regressors"
		"basis", "dictionary"	"expansion coefficients"
		"patterns"	"activation coefficients"

and $\mathbf{W}$ is learnt from the set of data vectors $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_N]$.

- **nonneg. of W** ensures *interpretability* of the dictionary (features $\mathbf{w}_k$ and data $\mathbf{v}_n$ belong to same space).
- **nonneg. of H** tends to produce *part-based* representations because subtractive combinations are forbidden.

From Cédric Févotte's slides

NMF for Music Audio

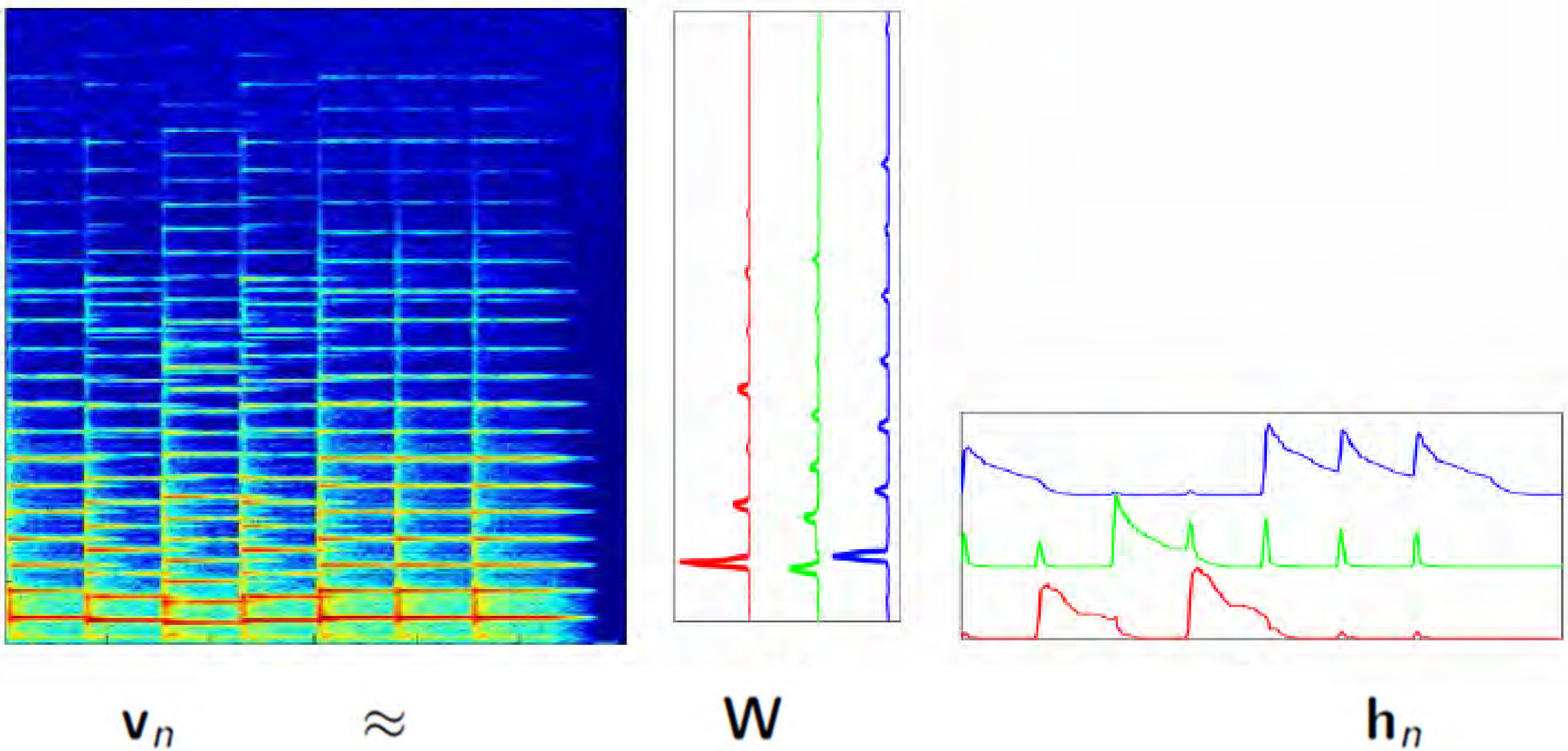


Figure from Byran & Sun's slides

NMF for Music Audio

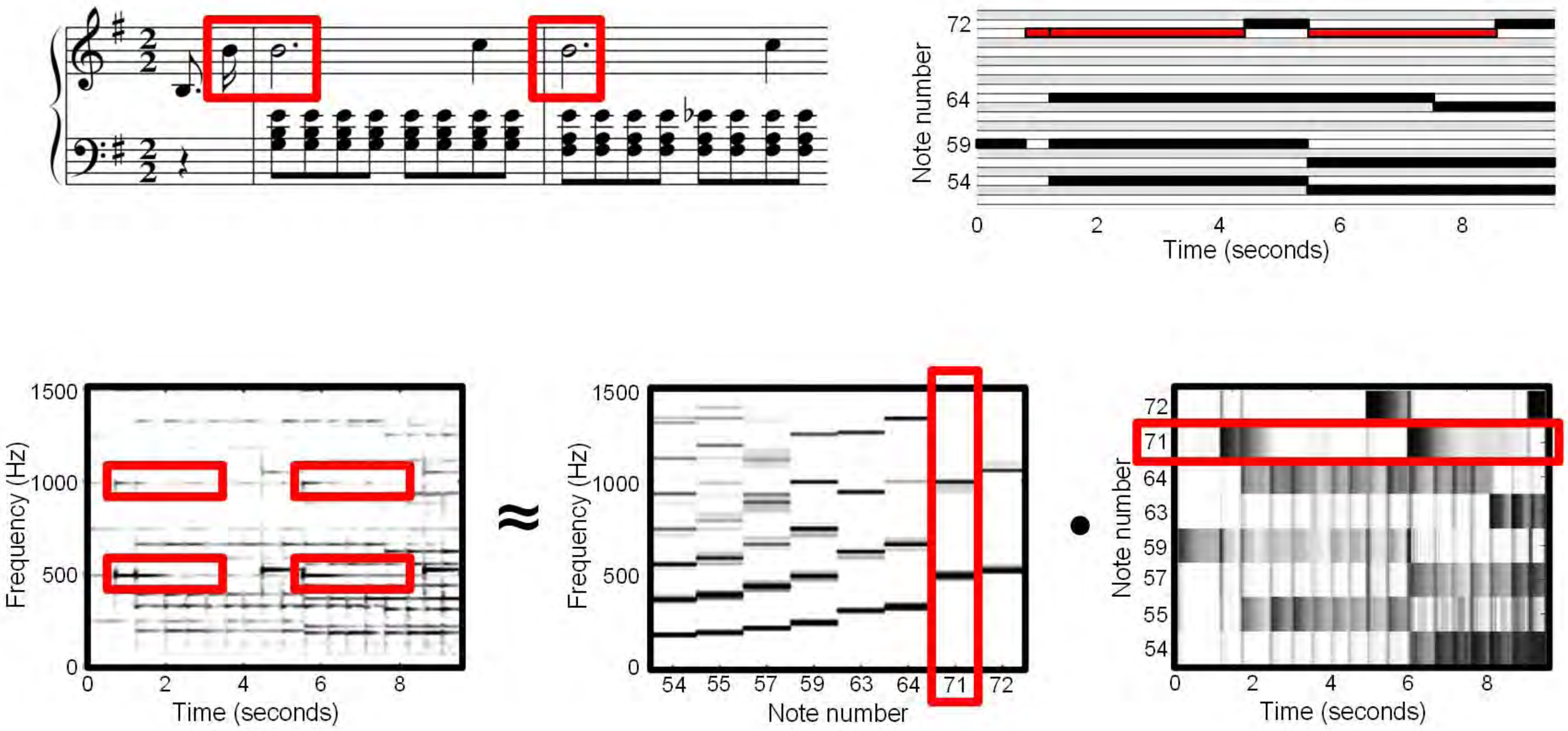
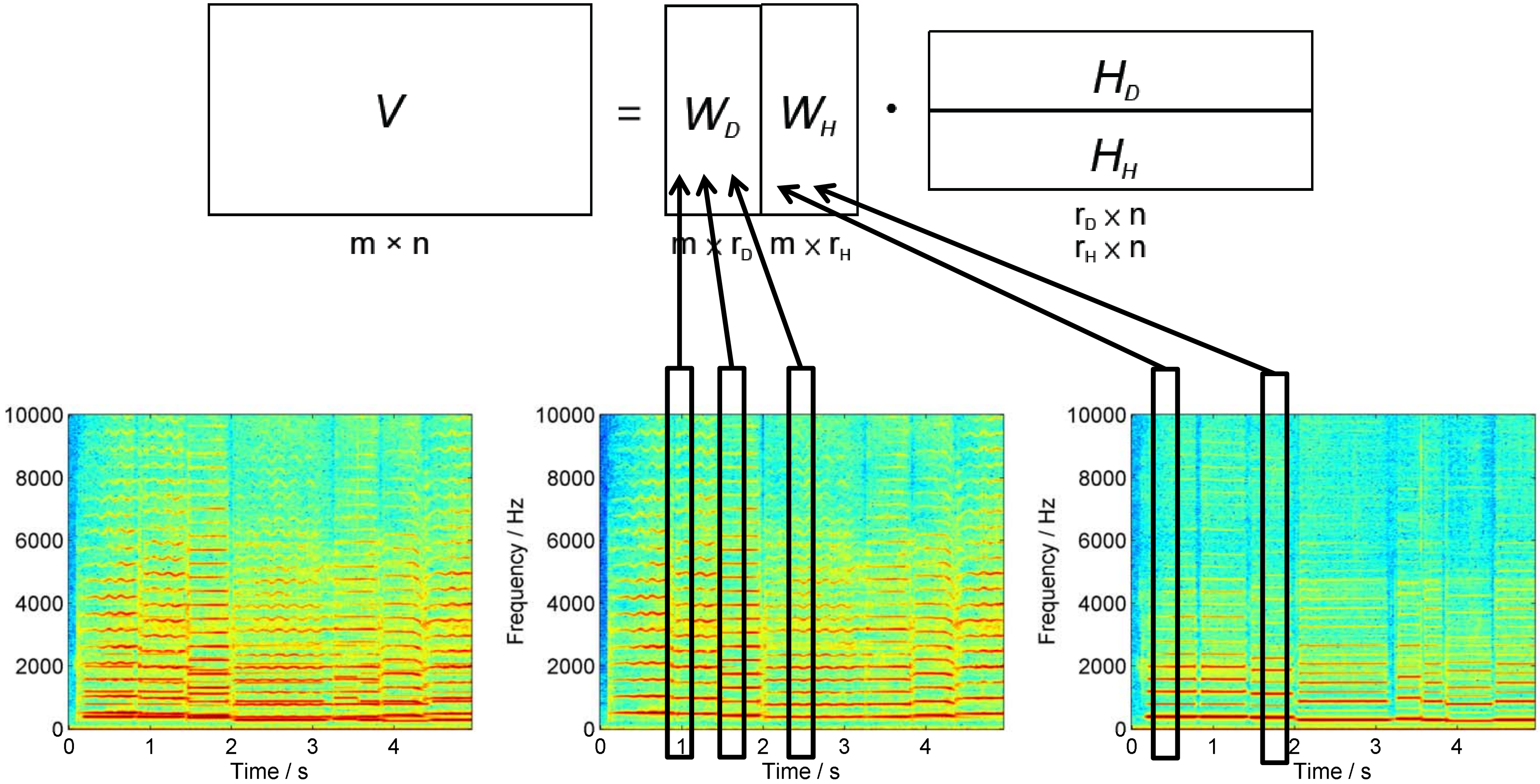
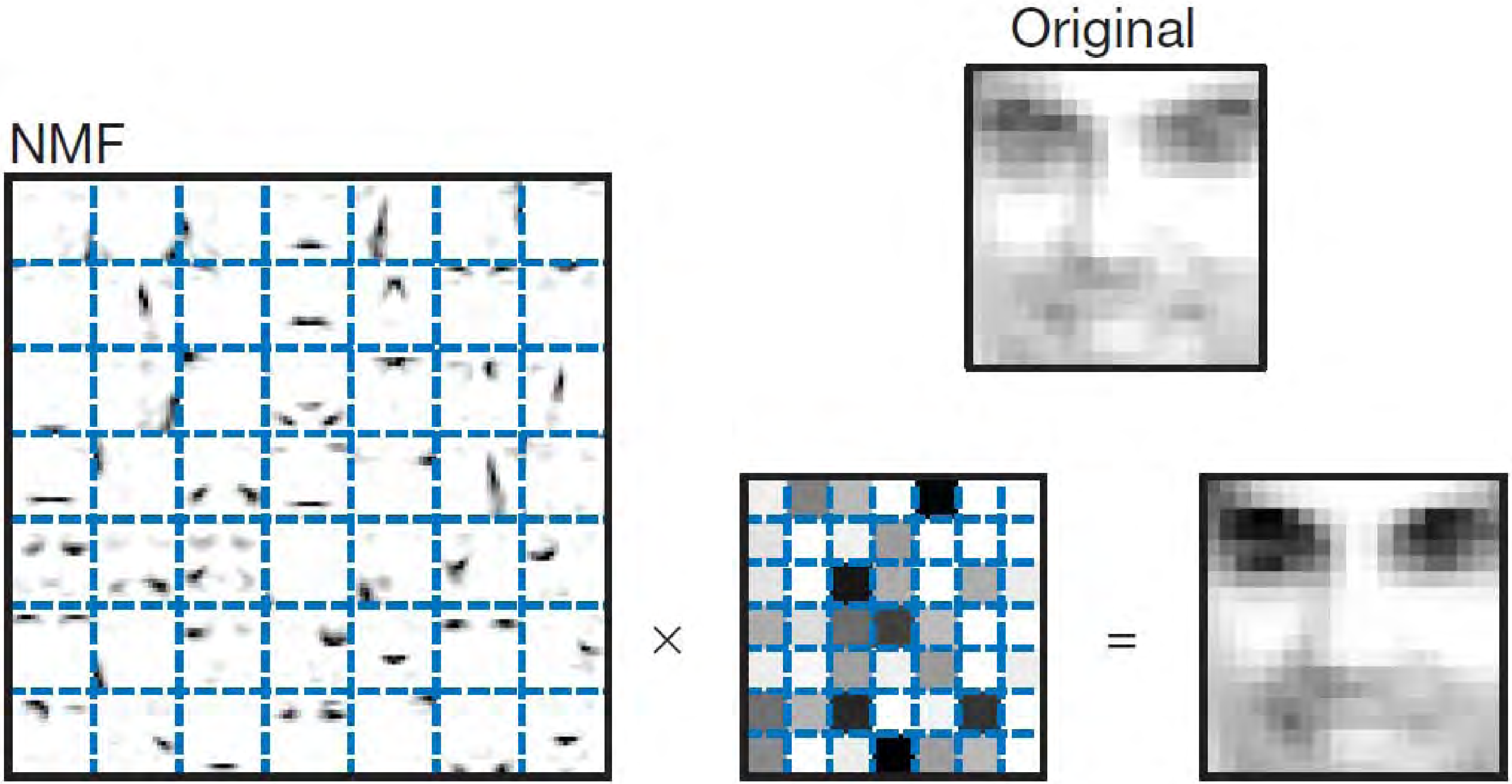


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF for Music Audio



NMF for Face Images



NMF: Algorithm

We seek to minimize a measure of fit between data $\mathbf{V}$ and model $\mathbf{WH}$, subject to nonnegativity of $\mathbf{W}$ and $\mathbf{H}$:

$$\min_{\mathbf{W},\mathbf{H}\geq \mathbf{0}} D(\mathbf{V}|\mathbf{WH}) = \sum_{fn} d([\mathbf{V}]_{fn}|[\mathbf{WH}]_{fn})$$

where $d(x|y)$ is a scalar cost function.

Regularization terms are often added to $D(\mathbf{V}|\mathbf{WH})$ to favor sparsity or smoothness of $\mathbf{W}$ or $\mathbf{H}$.

From Cédric Févotte's slides

NMF: Algorithm

- Block-coordinate update of $\mathbf{H}$ given $\mathbf{W}^{(i-1)}$ and $\mathbf{W}$ given $\mathbf{H}^{(i)}$

$$\min_{\mathbf{H}\geq\mathbf{0}} D(\mathbf{V}|\mathbf{W}^{(i-1)}\mathbf{H}), \quad \min_{\mathbf{W}\geq\mathbf{0}} D(\mathbf{V}|\mathbf{W}\mathbf{H}^{(i)})$$

- The updates of $\mathbf{W}$ and $\mathbf{H}$ are equivalent by symmetry:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \iff \mathbf{V}^T \approx \mathbf{H}^T\mathbf{W}^T$$

- The objective function is separable in the columns of $\mathbf{H}$ or the rows of $\mathbf{W}$:

$$D(\mathbf{V}|\mathbf{W}\mathbf{H}) = \sum_n D(\mathbf{v}_n|\mathbf{W}\mathbf{h}_n)$$

From Cédric Févotte's slides

NMF: Algorithm

- **Cost function**: Euclidean distance

$$||V - WH||^2 = \sum_{ij}(V_{ij} - WH_{ij})^2$$

- **Fix *W*, update *H***: ***additive*** update

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}\left[(W^T V)_{a\mu} - (W^T W H)_{a\mu}\right].$$

  - hard to set the learning rate $\eta_{a\mu}$
  - hard to ensure nonnegativity

NMF: Algorithm

- **Cost function**: Euclidean distance

$$||V - WH||^2 = \sum_{ij} (V_{ij} - WH_{ij})^2$$

- **Fix *W*, update *H***: ***multiplicative*** update

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu} \left[(W^T V)_{a\mu} - (W^T W H)_{a\mu} \right].$$

$$\eta_{a\mu} = \frac{H_{a\mu}}{(W^T W H)_{a\mu}},$$

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

NMF: Algorithm

- **Fix *W*, update *H***: ***multiplicaitve*** update

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

  - easily preserver nonnegativity
  - easy to implement
  - fast (of complexity O(FKN) per iteration)
  - zeros remain zeros!

NMF: Algorithm

Algorithm: NMF ($V \approx WH$)

Input: Nonnegative matrix V of size $K \times N$
Rank parameter $R \in \mathbb{N}$
Threshold ε used as stop criterion

Output: Nonnegative template matrix W of size $K \times R$
Nonnegative activation matrix H of size $R \times N$

Procedure: Define nonnegative matrices $W^{(0)}$ and $H^{(0)}$ by some random or informed initialization. Furthermore set $\ell = 0$. Apply the following update rules (written in matrix notation):

(1) $H^{(\ell+1)} = H^{(\ell)} \odot \left(((W^{(\ell)})^\top V) \oslash ((W^{(\ell)})^\top W^{(\ell)} H^{(\ell)}) \right)$

(2) $W^{(\ell+1)} = W^{(\ell)} \odot \left((V(H^{(\ell+1)})^\top) \oslash (W^{(\ell)} H^{(\ell+1)} (H^{(\ell+1)})^\top) \right)$

(3) Increase ℓ by one.

Repeat the steps (1) to (3) until $\|H^{(\ell)} - H^{(\ell-1)}\| \leq \varepsilon$ and $\|W^{(\ell)} - W^{(\ell-1)}\| \leq \varepsilon$ (or until some other stop criterion is fulfilled). Finally, set $H = H^{(\ell)}$ and $W = W^{(\ell)}$.

Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF for Music Audio Decomposition

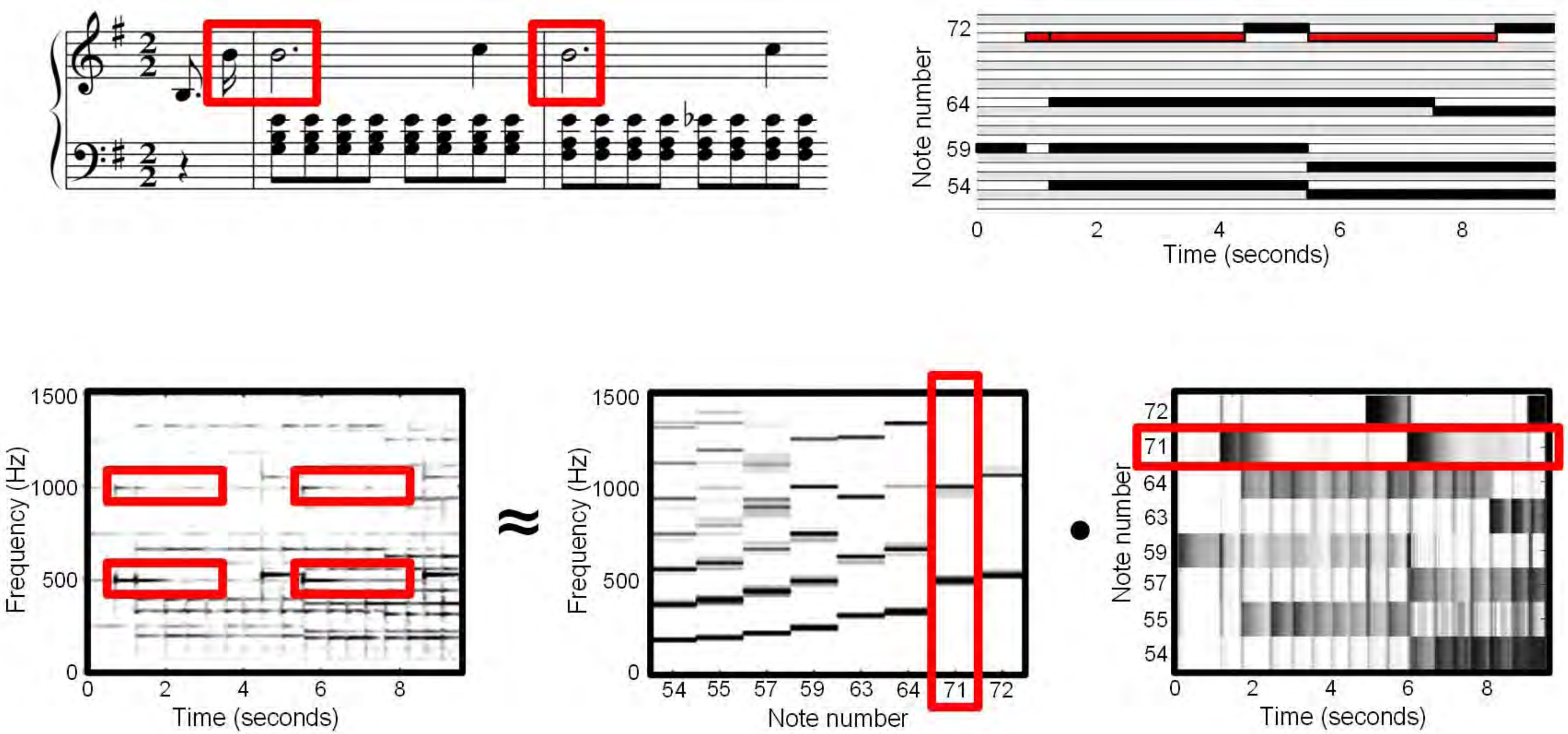
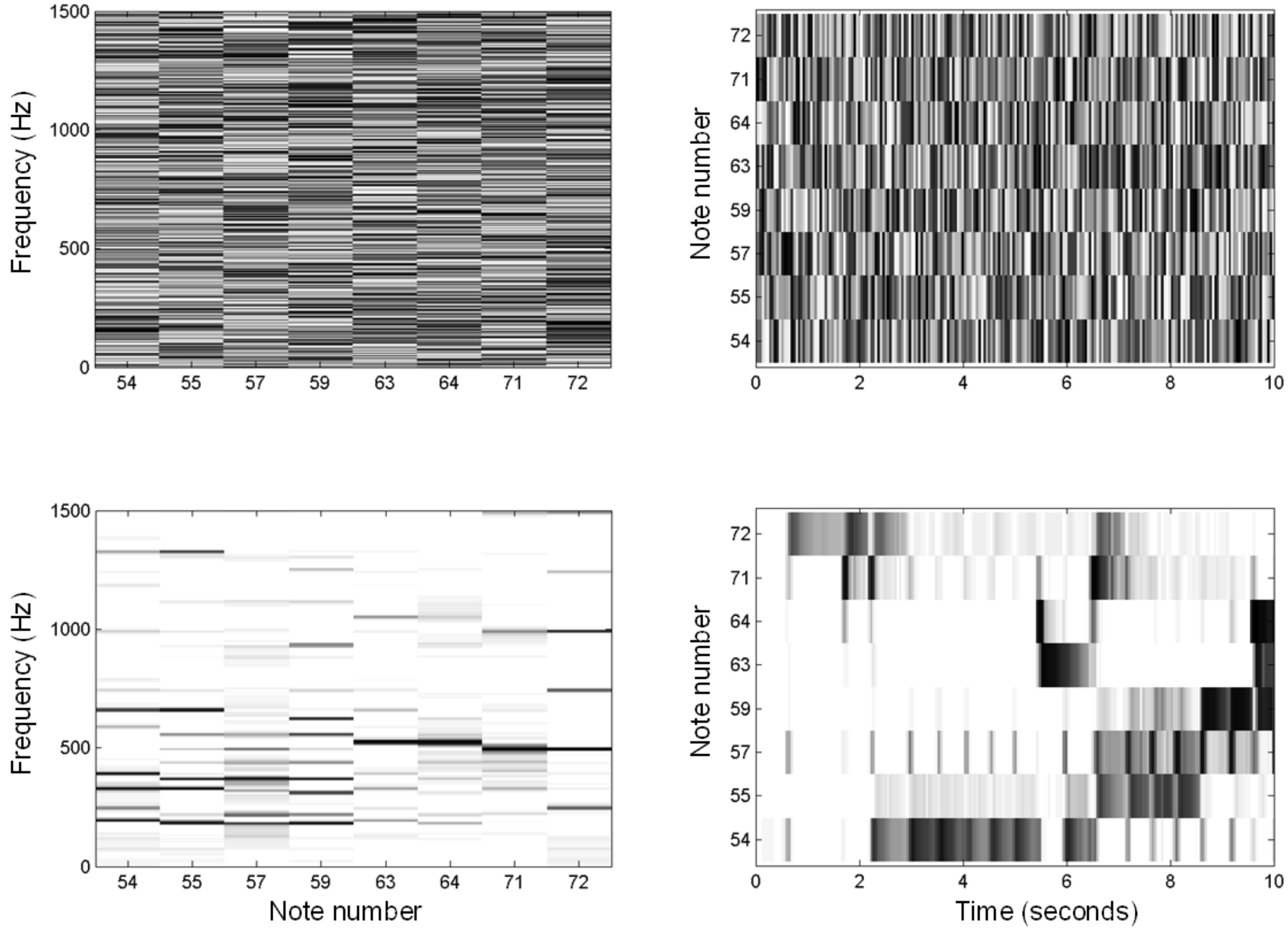


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF: Random Initialization

initial W

initial H



learned W

learned H

Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF: Harmonic Template Initialization

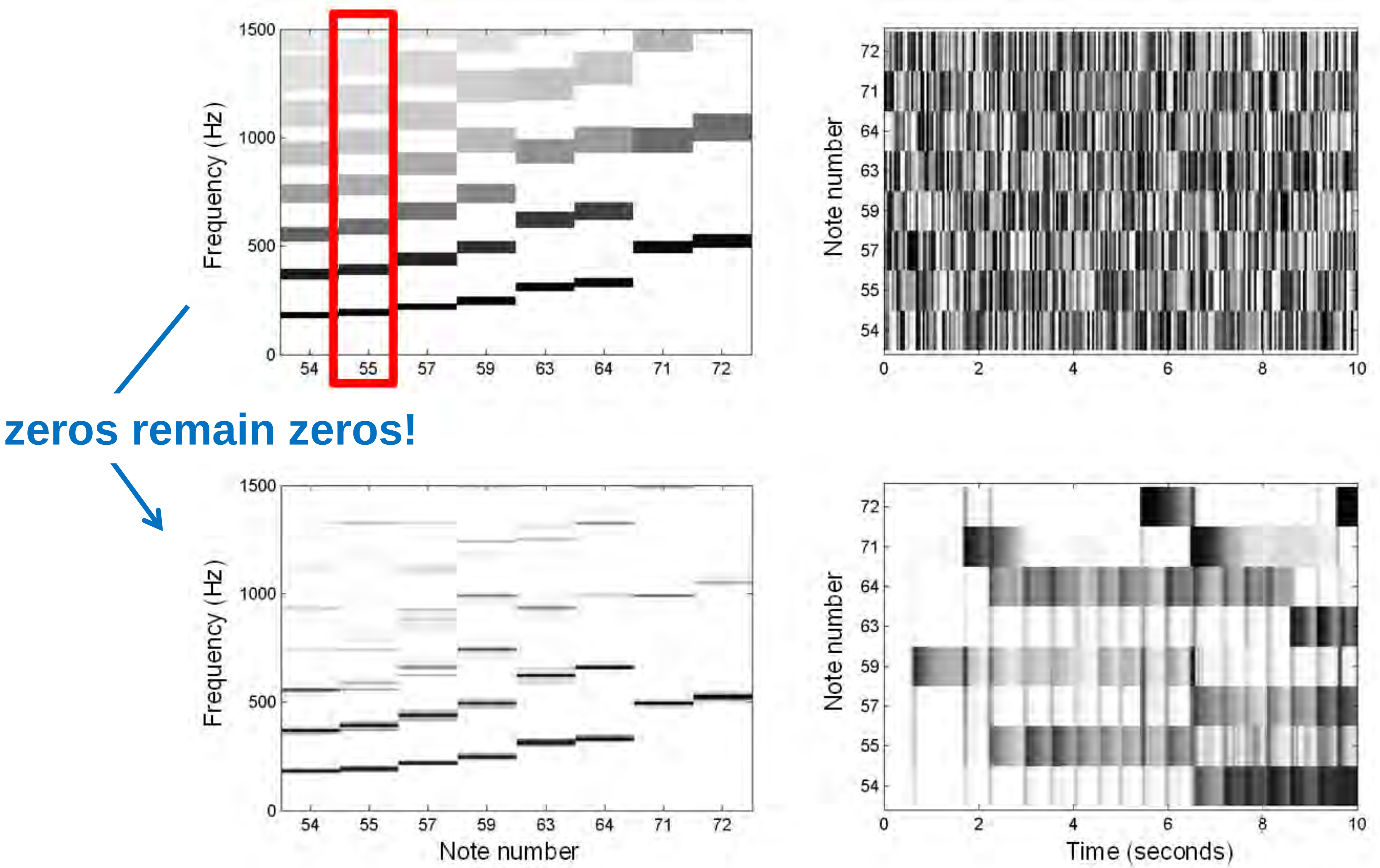


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

NMF: Score-Informed Initialization

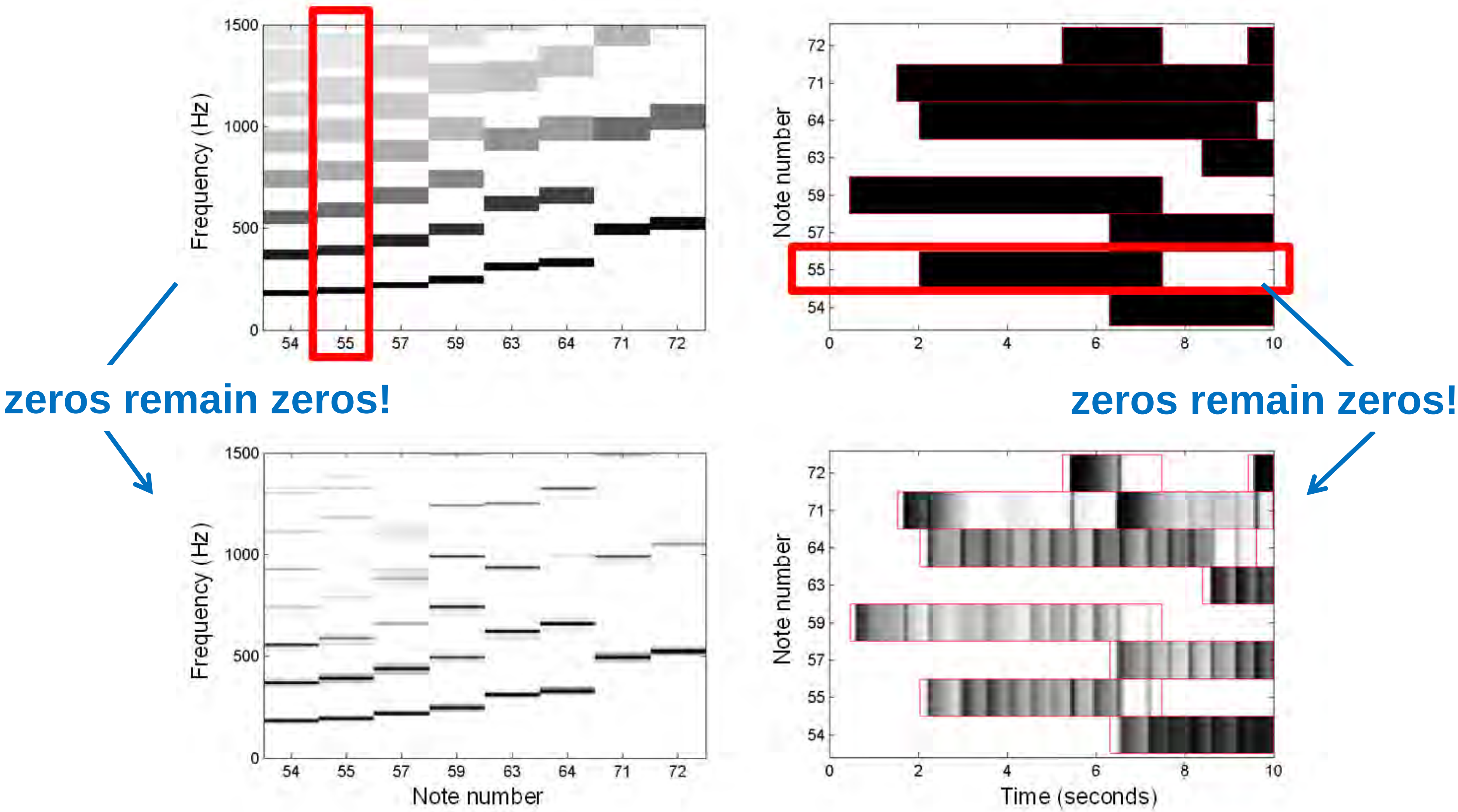
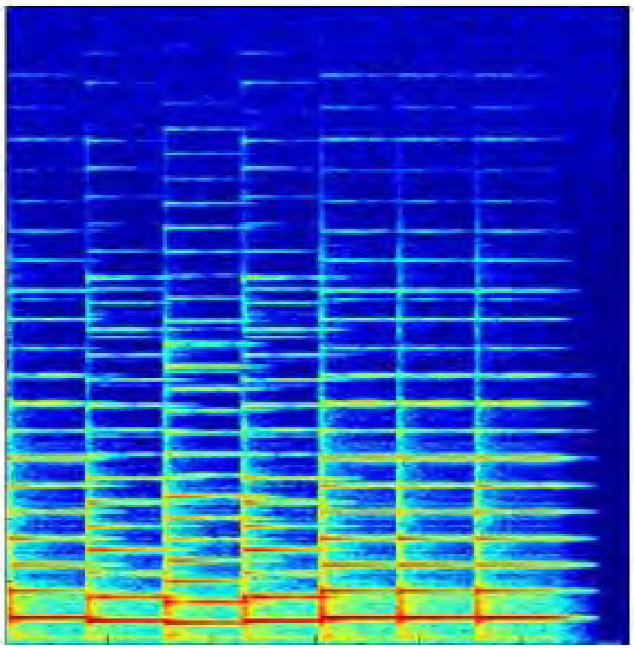
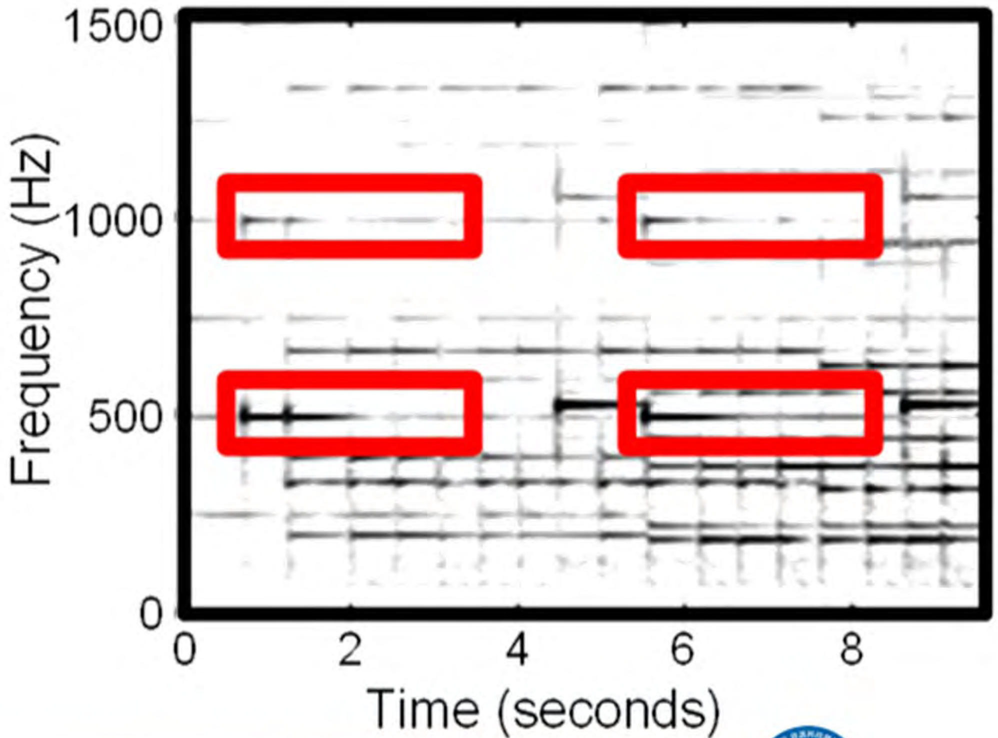


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Dealing with Transients

- In acoustics and audio, a transient is a high amplitude, short-duration sound at the beginning of a waveform that occurs in phenomena such as musical sounds





NMF: Score-Informed Initialization + Onset

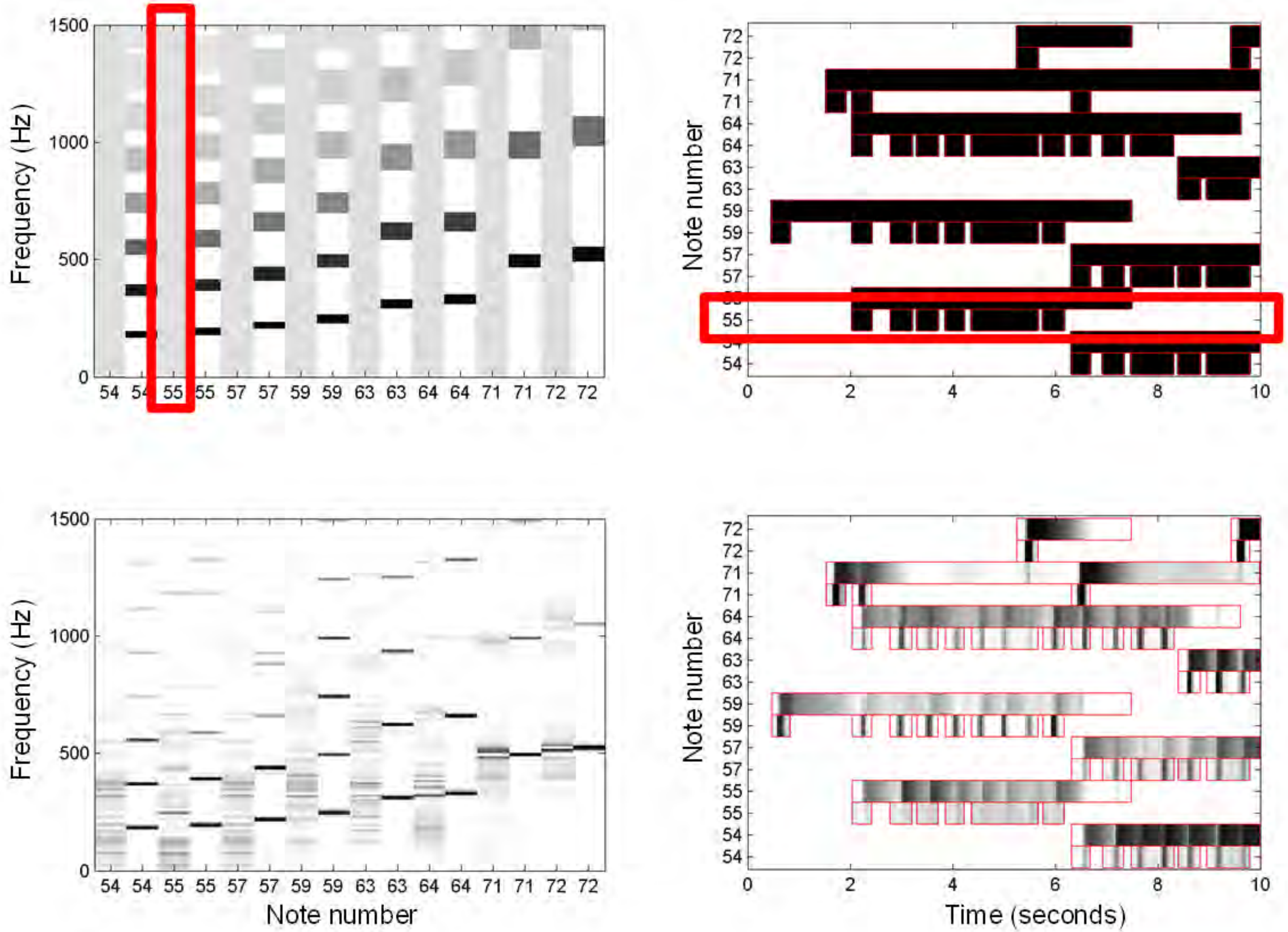


Figure from [Mueller, FPM, Chapter 8, Springer 2015]

Unsupervised vs Supervised NMF

- **Unsupervised**: decompose the matrix itself

$$\min_{W,H} \| V - WH \|_F$$

- **Supervised**: use pre-trained templates

Training phase

$$\min_{W_A, H_A} \| V_A - W_A H_A \|_F$$

$$\min_{W_B, H_B} \| V_B - W_B H_B \|_F$$

Testing phase

$$\min_{H} \| V_{\text{mix}} - [W_A, W_B] H \|_F$$

NMF: Implementation

- Matlab
- Python
  - http://bmcfee.github.io/librosa/generated/librosa.decompose.decompose.html#librosa.decompose.decompose
  - http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html#sklearn.decomposition.NMF
- Or,
  - https://www.csie.ntu.edu.tw/~cjlin/nmf/

Toolboxes for NMF-based Separation

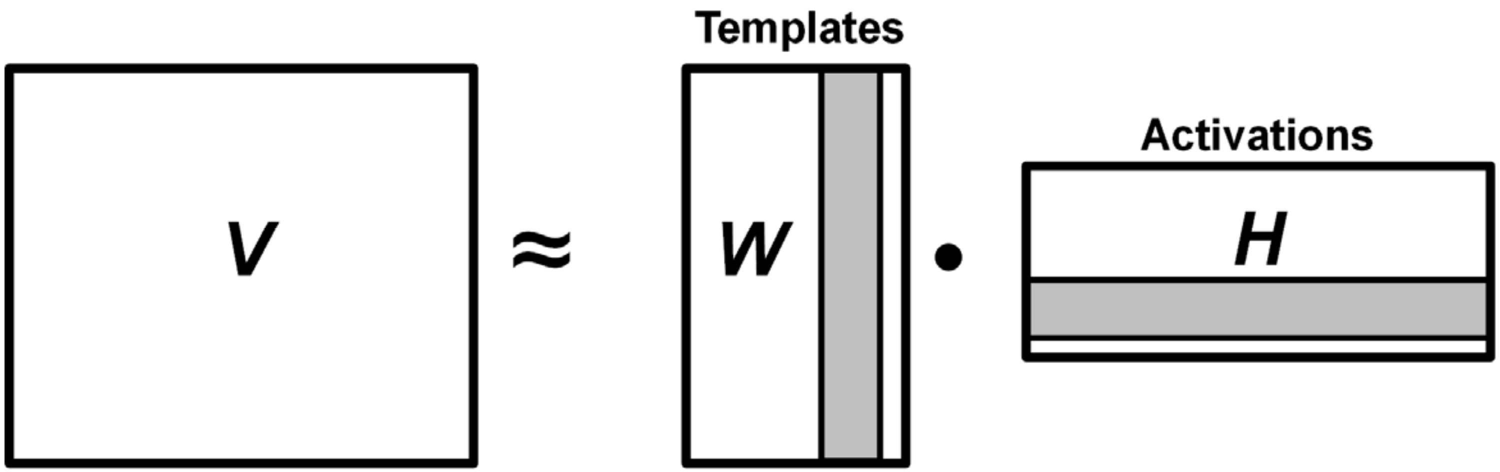
- Flexible Audio Source Separation Toolkit (**FASST**)
  - http://bass-db.gforge.inria.fr/fasst/
  - implemented in C++, Matlab and python
  - more sophisticated

$$\mathbf{V}_j = (\mathbf{W}_j^{\text{ex}}\,\mathbf{U}_j^{\text{ex}}\,\mathbf{G}_j^{\text{ex}}\,\mathbf{H}_j^{\text{ex}}) \odot (\mathbf{W}_j^{\text{ft}}\,\mathbf{U}_j^{\text{ft}}\,\mathbf{G}_j^{\text{ft}}\,\mathbf{H}_j^{\text{ft}})$$

- **OpenBliSSART**
  - http://openblissart.github.io/openBliSSART/
  - implemented in C++, can be run on GPUs

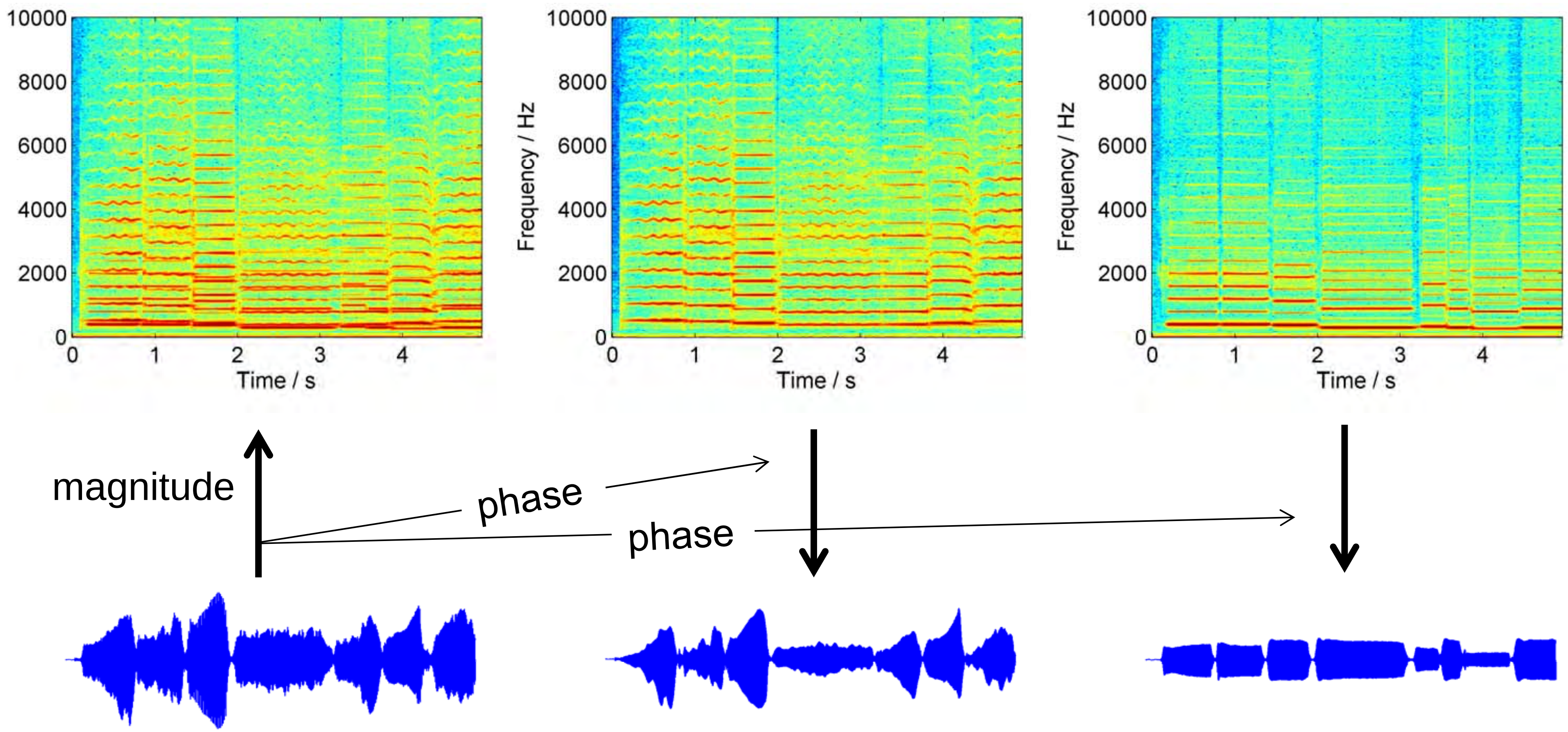
Parameters

- Window size, hop size
- Number of templates
- Normalization of the templates
- Cost function of NMF
- Reconstruction method



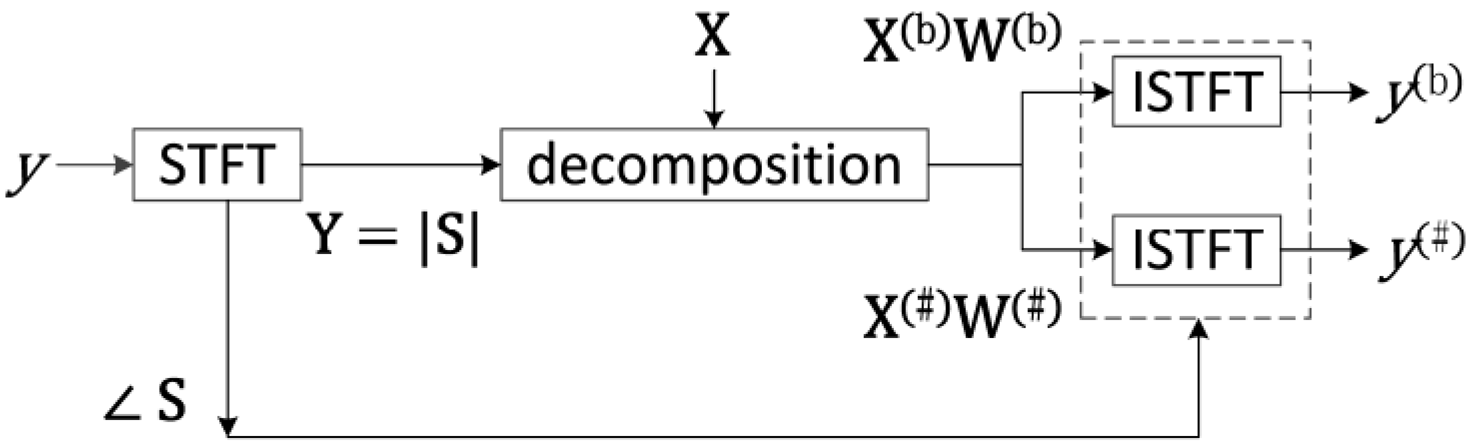
Reconstruction

- Need to recover the time-domain signals



Reconstruction

1. Given a mixture *y*, compute the STFT ***Y***
2. Decompose the magnitude |***Y***| into two matrices **A** and **B** (which are also real values)
3. Make **A** (or **B**) complex by adding the phase ∢***Y*** back
4. Do inverse STFT (ISTFT)



Reconstruction

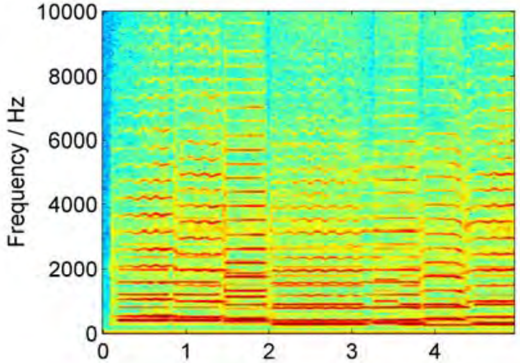
1. Given a mixture *y*, compute the STFT ***Y***
2. Decompose |***Y***| into **A** and **B**
3. Make **A** (or **B**) complex by adding the phase ∢***Y*** back
4. Do ISTFT

- https://www.ee.columbia.edu/~dpwe/resources/matlab/sgram/
- myspecgram
- abs, angle
- ispecgram

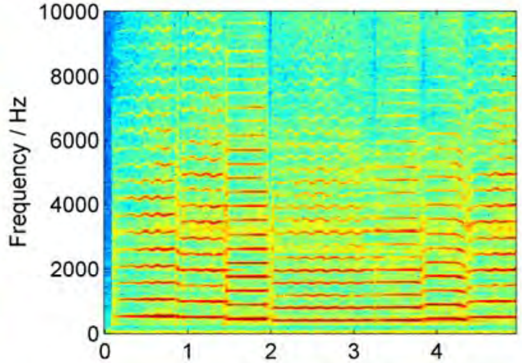
- |***Y***|=abs(***Y***), ∢***Y*** =angle(***Y***)
- ***Y*** = |***Y***|.*cos(∢***Y***) + i*|***Y***|.*sin(∢***Y***);

Reconstruction: Wiener Filter (Binary)

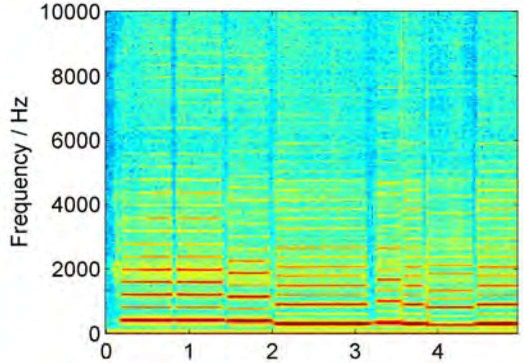
$|Y|$



A

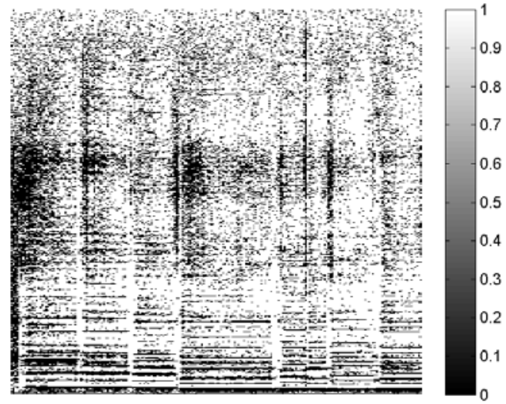


B

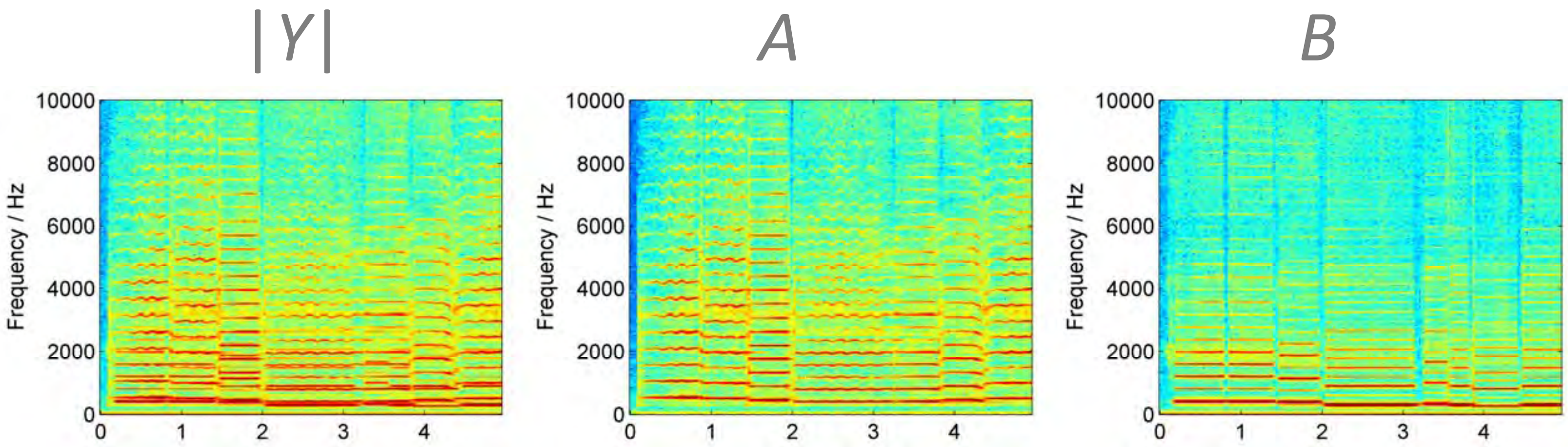


- $M_A[t,f] = \begin{cases} 1, & \text{if } A[t,f] > B[t,f] \\ 0, & \text{otherwise} \end{cases}$
- $\hat{A} = |Y| \odot M_A$
- Use $\hat{A}$ instead of A in the ISTFT
- M_A is referred to as a *binary mask*

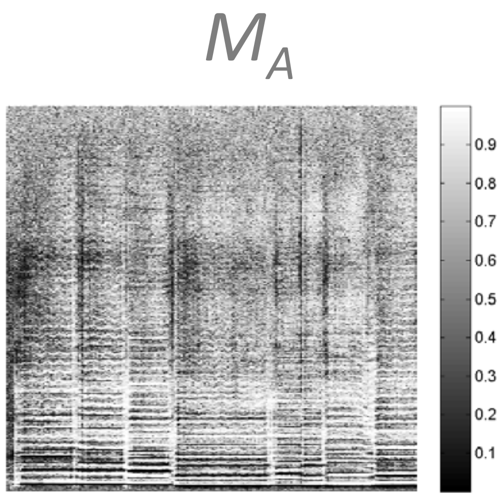
M_A



Reconstruction: Wiener Filter (Soft)



- $M_A[t,f] = \frac{A[t,f]^c}{A[t,f]^c + B[t,f]^c}$
- $\hat{A} = |Y| \odot M_A$
- Use $\hat{A}$ instead of A in the ISTFT
- M_A is referred to as a *soft mask*
- c = 1 or 2



Evaluation

- Source-to-distortion ratio (SDR)
- Source-to-interference ratio (SIR)
- Source-to-artifact ratio (SAR)
  - true sources: **a**, **b**
  - estimated sources: **ae**, **be**
  - SDR(a): how **ae** is similar to **a**
  - SIR(a): how **ae** is similar to **b**
  - SAR(a): how **ae** is not similar to either **a or b**
  - we can also compute SDR(b), SIR(b), SAR(b)

Evaluation

- **BSS_Eval** (Matlab)
  - http://bass-db.gforge.inria.fr/bss_eval/bss_eval_sources.m

```
% [SDR,SIR,SAR,perm]=bss_eval_sources(se,s)
%
% Inputs:
% se: nsrc x nsampl matrix containing estimated sources
% s: nsrc x nsampl matrix containing true sources
%
% Outputs:
% SDR: nsrc x 1 vector of Signal to Distortion Ratios
% SIR: nsrc x 1 vector of Source to Interference Ratios
% SAR: nsrc x 1 vector of Sources to Artifacts Ratios
% perm: nsrc x 1 vector containing the best ordering of estimated sources
% in the mean SIR sense (estimated source number perm(j) corresponds to
% true source number j)
```

Evaluation

- **mir_eval** (python)
  - http://labrosa.ee.columbia.edu/mir_eval/
  - http://craffel.github.io/mir_eval/#module-mir_eval.separation

```
>>> # reference_sources[n] should be an ndarray of samples of the
>>> # n'th reference source
>>> # estimated_sources[n] should be the same for the n'th estimated
>>> # source
>>> (sdr, sir, sar,
...  perm) = mir_eval.separation.bss_eval_sources(reference_sources,
...                                                estimated_sources)
```

- mir_eval can be used in most MIR tasks (chord recognition, onset detection, segmentation, etc)

Evaluation

- Source-to-distortion ratio (SDR)
- Source-to-interference ratio (SIR)
- Source-to-artifact ratio (SAR)
  - true sources: **a**, **b**
  - estimated sources: **ae**, **be**

- **ae** can be slightly shorter than **a** due to the windowing => *chop off* the end of **a** such that the length of **a** and **ae** are the same

Extension: Different Cost Functions*

- β-divergence

$$d_\beta(x|y) = \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta-1}$$

- $\beta = 2$ (Euclidean): $d(x|y) = \frac{1}{2}(x-y)^2$
- $\beta = 1$ (Kullback-Leibler): $d(x|y) = x \log \frac{x}{y} - x + y$
- $\beta = 0$ (Itakura-Saito): $d(x|y) = \frac{x}{y} - \log \frac{x}{y} - 1.$

– Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence, ICASSP 2014
– Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis, Neural Computing 2009

Extension: Different Cost Functions*

- Euclidean distance

Theorem 1 *The Euclidean distance $||V - WH||$ is nonincreasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}} \qquad W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}} \tag{4}$$

- KL divergence

Theorem 2 *The divergence $D(V||WH)$ is nonincreasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu} \frac{\sum_i W_{ia} V_{i\mu}/(WH)_{i\mu}}{\sum_k W_{ka}} \qquad W_{ia} \leftarrow W_{ia} \frac{\sum_\mu H_{a\mu} V_{i\mu}/(WH)_{i\mu}}{\sum_\nu H_{a\nu}} \tag{5}$$

Algorithms for non-negative matrix factorization, NIPS 2000

Extension: Temporal Continuity & Sparsity

$$D(\mathbf{X} \| \mathbf{BG}) = \sum_{k,t} \underline{[\mathbf{X}]_{k,t} \log \frac{[\mathbf{X}]_{k,t}}{[\mathbf{BG}]_{k,t}} - [\mathbf{X}]_{k,t} + [\mathbf{BG}]_{k,t}}.$$

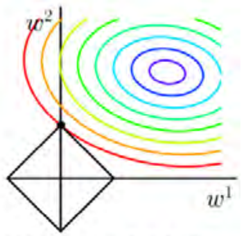
$$c_{\mathrm{t}}(\mathbf{G}) = \sum_{j=1}^{J} \frac{1}{\sigma_j^2} \sum_{t=2}^{T} \underline{(g_{t,j} - g_{t-1,j})^2}.$$

squared difference

$$c_{\mathrm{s}}(\mathbf{G}) = \sum_{j=1}^{J} \sum_{t=1}^{T} \underline{f(g_{j,t}/\sigma_j)}$$

usually implemented by the L1 norm

$$\nabla c(\mathbf{B}, \mathbf{G}) = \nabla c_{\mathrm{r}}(\mathbf{B}, \mathbf{G}) + \alpha \nabla c_{\mathrm{t}}(\mathbf{G}) + \beta \nabla c_{\mathrm{s}}(\mathbf{G}).$$



(a) ℓ_1-ball meets quadratic function. ℓ_1-ball has corners. It's very likely that the meet-point is at one of the corners.

Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria, TASLP 2007

Extension: More Regularizers

- http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html#sklearn.decomposition.NMF

The objective function is:

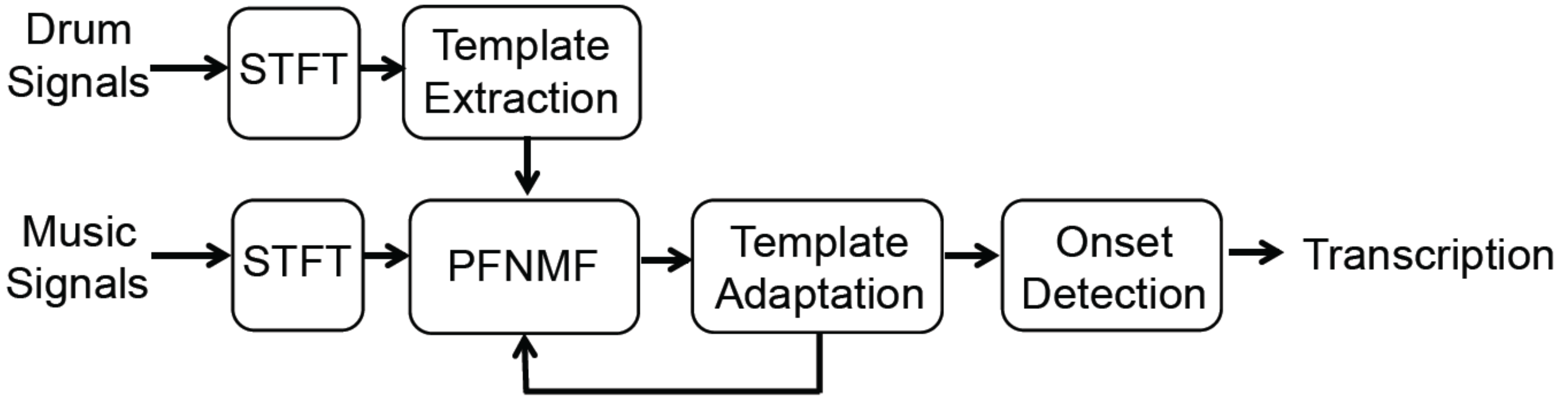
```
0.5 * ||X - WH||_Fro^2
+ alpha * l1_ratio * ||vec(W)||_1
+ alpha * l1_ratio * ||vec(H)||_1
+ 0.5 * alpha * (1 - l1_ratio) * ||W||_Fro^2
+ 0.5 * alpha * (1 - l1_ratio) * ||H||_Fro^2
```

$$\underset{\mathbf{H}}{\operatorname{argmin}}\, D(\mathbf{M} \mid \mathbf{WH}) + \mu|\mathbf{H}|_1$$

```
>>> import numpy as np
>>> X = np.array([[1,1], [2, 1], [3, 1.2], [4, 1], [5, 0.8], [6, 1]])
>>> from sklearn.decomposition import NMF
>>> model = NMF(n_components=2, init='random', random_state=0)
>>> model.fit(X)
NMF(alpha=0.0, beta=1, eta=0.1, init='random', l1_ratio=0.0, max_iter=200,
  n_components=2, nls_max_iter=2000, random_state=0, shuffle=False,
  solver='cd', sparseness=None, tol=0.0001, verbose=0)
```

Extension: Template Adaptation

- Pre-train the templates offline, but update them online according to the target signal



Drum transcription using partially fixed non-negative matrix factorization with template adaptation, ISMIR 2015

Extension: Adding a Noise Dictionary

- To account for the possible noises in the signal

W^p	W^v	W^g	W^d	W^n
piano	violin	guitar	drum	noise

Extension: Discriminative NMF

- Instead of training the dictionaries (templates) for different instruments separately; training them "jointly" to reduce the "cross-talk"

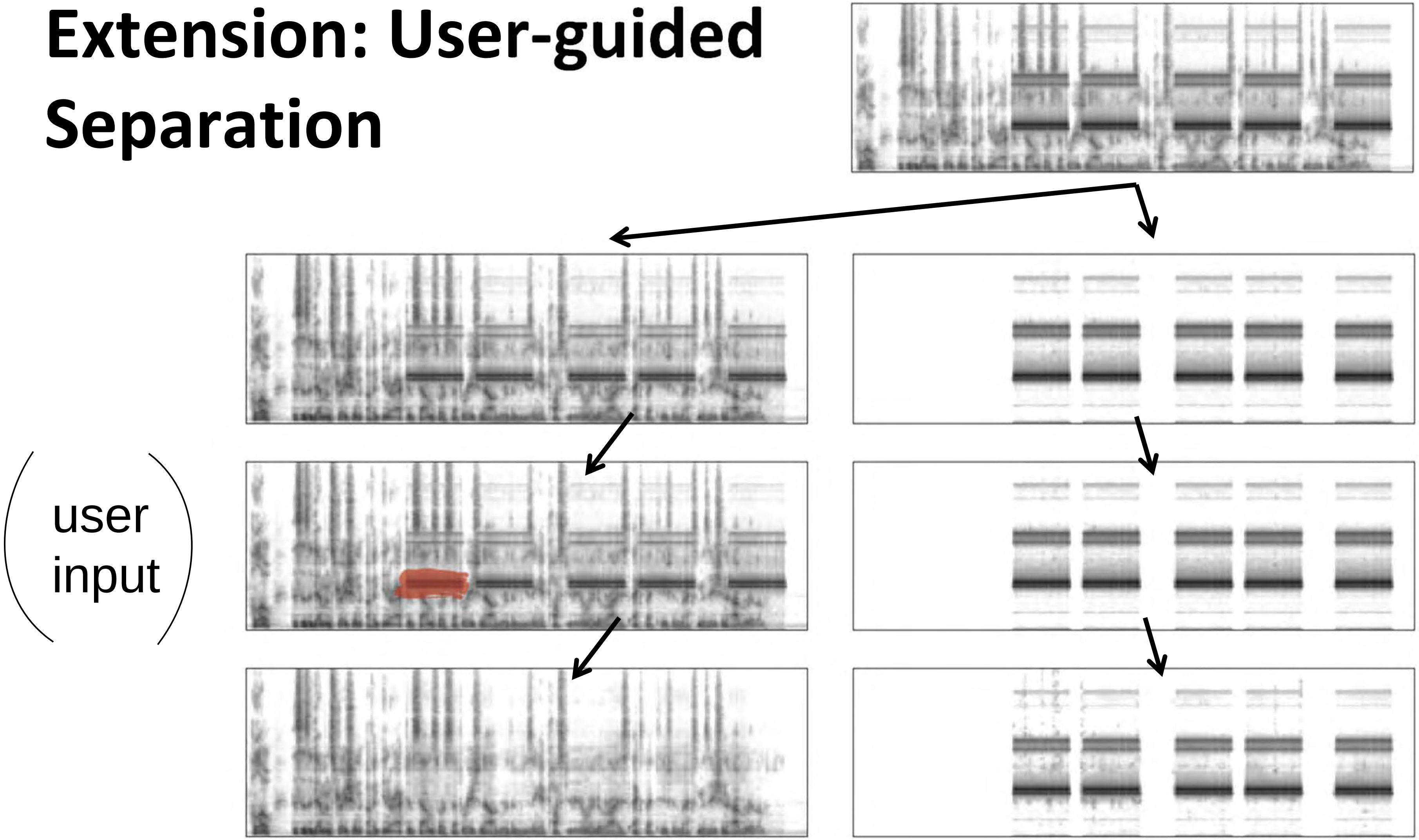
$$\mathbf{M} \approx \mathbf{W}\mathbf{H} = [\mathbf{W}^1 \cdots \mathbf{W}^S][\mathbf{H}^1; \cdots ; \mathbf{H}^S]$$

$$\hat{\mathbf{W}} = \operatorname*{argmin}_{\mathbf{W}} \sum_l \gamma_l D_\beta \left(\mathbf{S}^l \mid \mathbf{W}^l \hat{\mathbf{H}}^l(\mathbf{M}, \mathbf{W}) \right),$$

$$\text{where } \hat{\mathbf{H}}(\mathbf{M}, \mathbf{W}) = \operatorname*{argmin}_{\mathbf{H}} D_\beta(\mathbf{M} \mid \widetilde{\mathbf{W}}\mathbf{H}) + \mu |\mathbf{H}|_1,$$

Discriminative NMF and its application to single-channel source separation, ICASSP 2014

Extension: User-guided Separation



Interactive refinement of supervised and semi-supervised sound source separation estimates, ICASSP 2013

Extension: Complex NMF and Friends

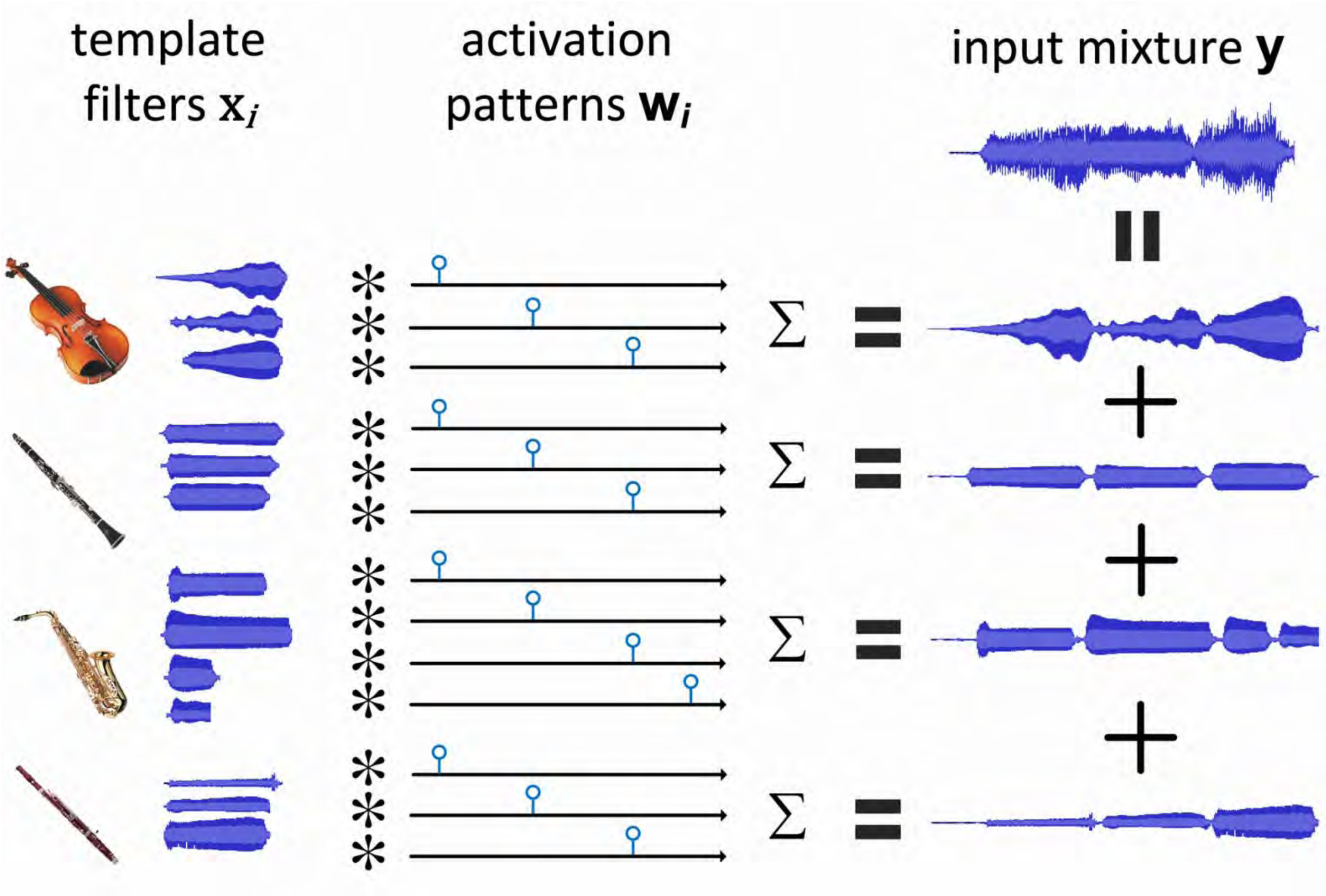
- Explicitly take phase into account

$$\mathcal{S} \approx \sum_{k=1}^{K} W(m,k)H(k,n)e^{i\phi_k(m,n)}.$$

- Or, do things directly in the time-domain

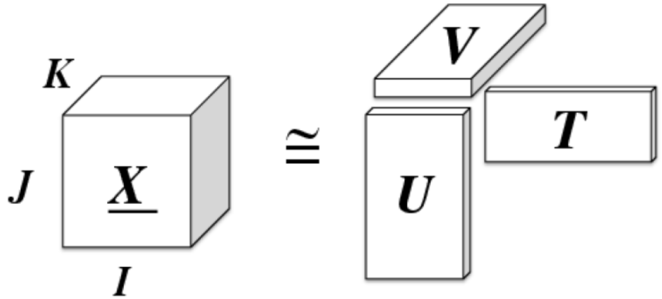
  – Complex NMF: A new sparse representation for acoustic signals, ICASSP 2009
  – Beyond NMF- time-domain audio source separation without phase reconstruction, ISMIR 2013
  – Informed monaural source separation of music based on convolutional sparse coding, ICASSP 2015
  – Multi-resolution signal decomposition with time-domain spectrogram factorization, ICASSP 2015
  – A score-informed shift-invariant extension of complex matrix factorization for improving the separation of overlapped partials in music recordings, ICASSP 2016

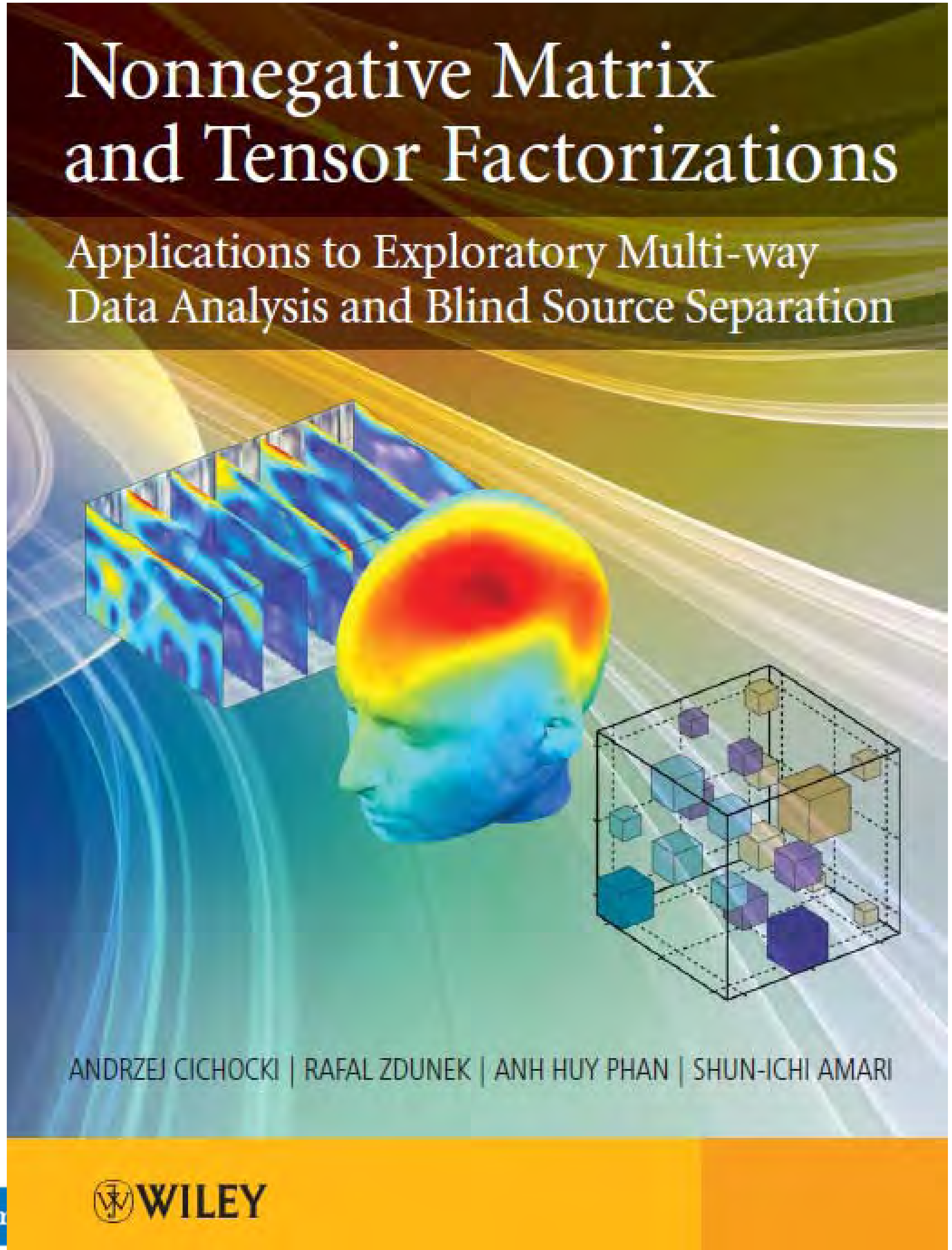
Extension: Time-domain Separation



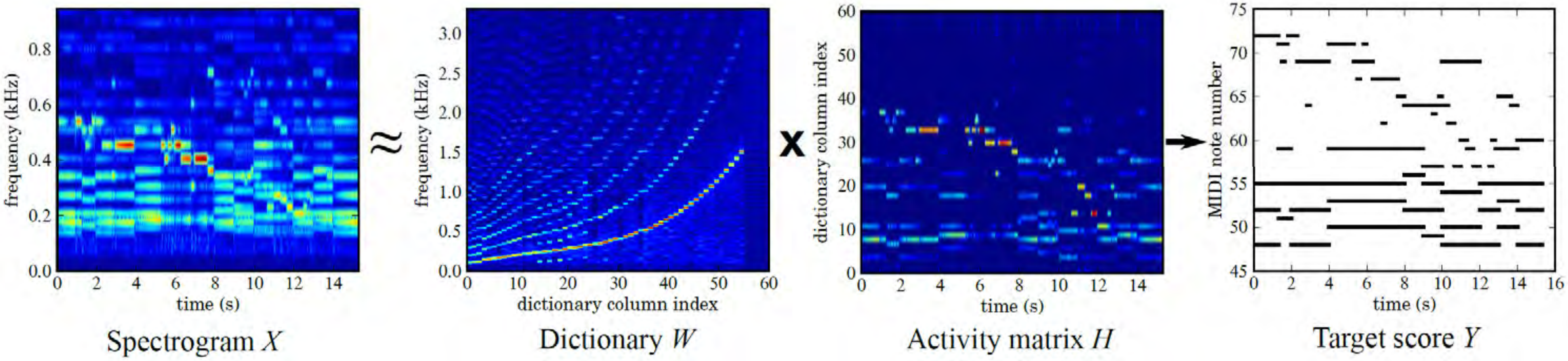
Informed monaural source separation of music based on convolutional sparse coding, ICASSP 2015

Extension: Tensor Decomposition





Extension: Dictionaries for Pitch Estimation



- Decompose the input as a linear combination of individual components
  - ➢ templates of instruments => source separation
  - ➢ templates of notes => multi-pitch estimation
  - ➢ templates of chords => chord recognition

Discriminative non-negative matrix factorization for multiple pitch estimation, ISMIR 2012

Extension: Voice Conversion

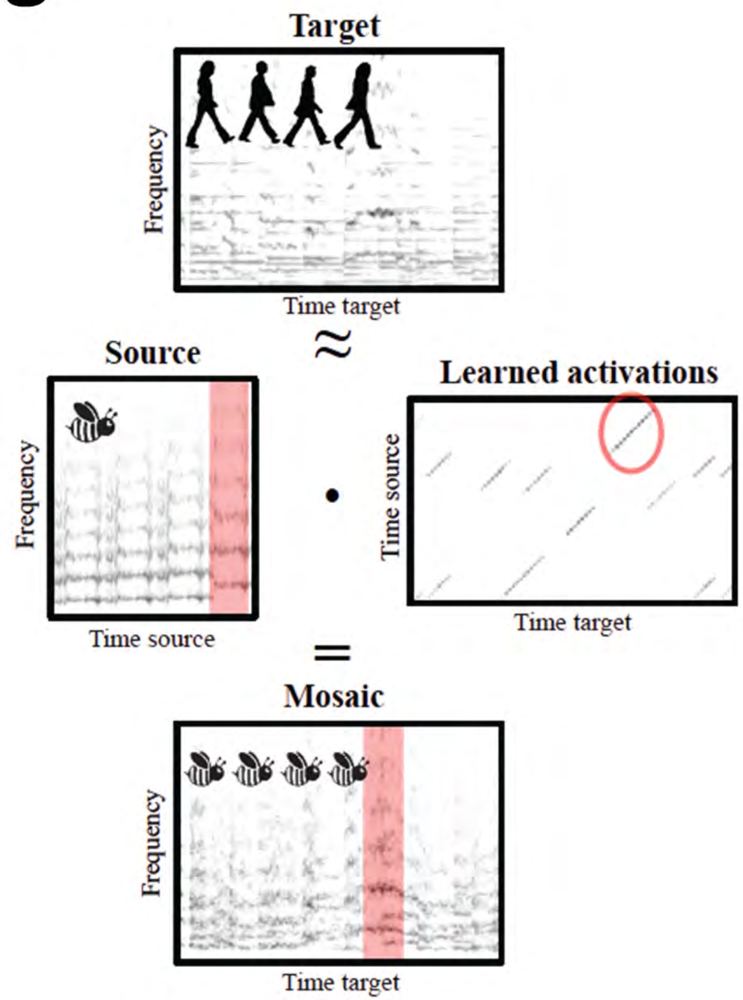
Quiz

- we have two dictionaries $\mathbf{X}_{\text{vio}} \in \mathbb{R}^{n \times m}$ and $\mathbf{X}_{\text{flu}} \in \mathbb{R}^{n \times m}$ for violin and flute, satisfying
  - each dictionary contains spectral templates of different pitches;
  - the two dictionaries have one-to-one correspondence (i.e. $\mathbf{X}_{\text{vio}}^{(j)}$ and $\mathbf{X}_{\text{flu}}^{(j)}$ correspond to the same pitch, $\forall j$);

  for a violin recording $\mathbf{Y}_*$, we compute $\mathbf{W}_*$ s.t. $\mathbf{Y}_* \simeq \mathbf{X}_{\text{vio}}\mathbf{W}_*$; then, what would happen if we take $\mathbf{X}_{\text{flu}}\mathbf{W}_*$?

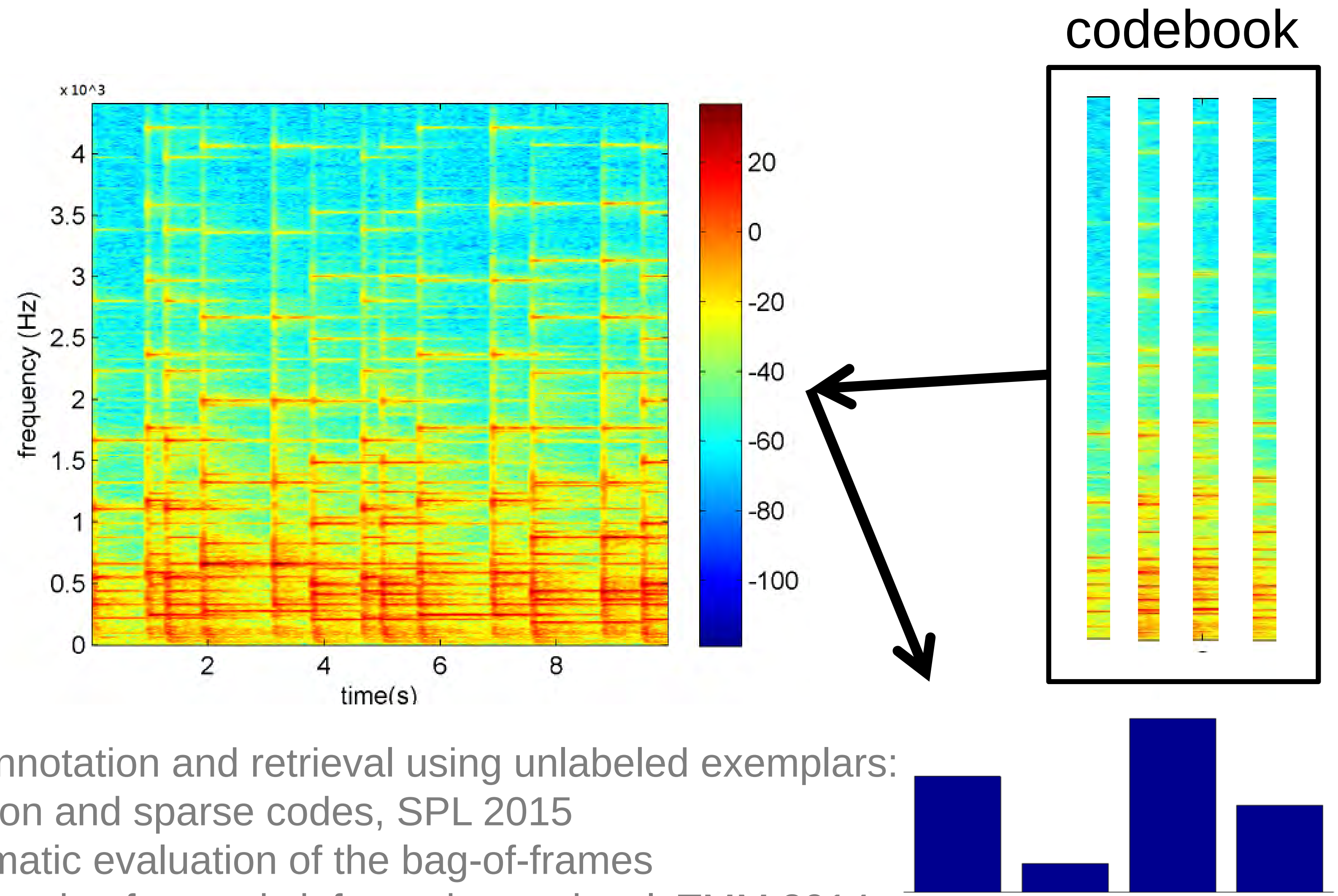
Extension: Audio Mosaicing

- Given a *target* and a *source* recording, the goal of *audio mosaicing* is to generate a mosaic recording that conveys musical aspects (like melody and rhythm) of the target, using sound components taken from the source
- https://www.audiolabs-erlangen.de/resources/MIR/2015-ISMIR-LetItBee/



Let it Bee - Towards NMF-Inspired Audio Mosaicing, ISMIR 2015

Extension: Dictionaries for Classification



- Music annotation and retrieval using unlabeled exemplars: correlation and sparse codes, SPL 2015
- A systematic evaluation of the bag-of-frames representation for music information retrieval, TMM 2014