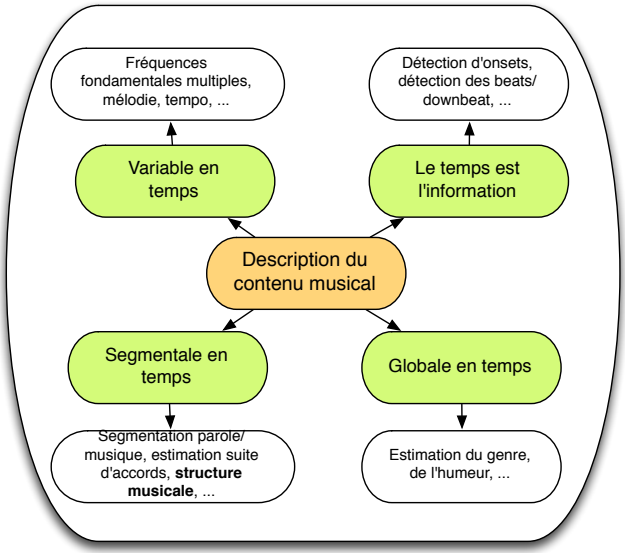
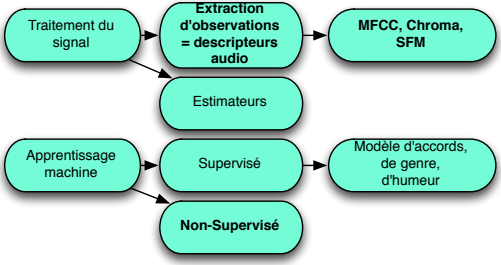
Master 2 ATIAM (2017-2018)
Extraction automatique d'une structure musicale et génération d'un résumé audio

Geoffroy.Peeters@ircam.fr
UMR STMS 9912 (IRCAM CNRS UPMC)

1. Introduction
1.1 Différents types de description du contenu musical
1.2 Détection d'une structures musicale d'un morceau de musique
1.3 Méthodes d'estimation de la structure
2. Descripteurs audio
2.1 Introduction
2.2 Taux de passage par zéro
2.3 Enveloppe ADSR
2.4 Description du spectre (barycentre, étendue spectral)
2.5 Mel Frequency Cepstral Coefficients (MFCCs)
2.6 Chroma - Pitch Class Profile (PCP)
2.7 Spectral Flatness Measure (SFM)
2.8 Intégration temporelle
3. Représentation visuelle de la structure temporelle de la musique
3.1 La matrice d'auto-similarité
3.2 Hypothèses concernant la macro-structure d'un morceau
3.3 Matrice d'auto-similarité/distance (temps,temps)
3.4 Matrice d'auto-similarité/distance (temps,lag)
3.5 Génération de résumé audio par méthode du "summary score"
4. Segmentation temporelle d'un flux de descripteurs
4.1 Segmentation trame-à-trame
4.2 Critère BIC (Bayes Information Criteria)
4.3 Convolution de la matrice d'auto-similarité par un noyau en damier
5. Algorithmes de clustering
5.1 Introduction
5.2 Algorithmes de partitionnement : K-Means (nuées dynamiques)
5.3 Algorithmes de partitionnement : Fuzzy-K-Means
5.4 Algorithmes de partitionnement : Gaussian Mixture Model
5.5 Algorithmes hiérarchiques : par agglomération
5.6 Algorithmes hiérarchiques : par divisions
5.7 Autres : Clustering spectral : Singular Value Decomposition
5.8 Autres : Clustering par NMF
5.9 Conclusion
6. Génération de résumé audio par estimation de structure
7. Estimation d'une structure musicale - approche par "séquence"
7.1 Segmentation : méthode des "Structural features"
7.2 Segmentation : méthode des "Structural features" avec probabilité a-priori
7.3 Regroupement par Dynamic Time Warping


Fréquences fondamentales multiples, mélodie, tempo, ...
Détection d'onsets, détection des beats/ downbeat, ...
Variable en temps
Le temps est l'information
Description du contenu musical
Segmentale en temps
Globale en temps
Segmentation parole/ musique, estimation suite d'accords, structure musicale, ...
Estimation du genre, de l'humeur, ...


Traitement du signal
Extraction d'observations = descripteurs audio
MFCC, Chroma, SFM
Estimateurs
Apprentissage machine
Supervisé
Modèle d'accords, de genre, d'humeur
Non-Supervisé

Objectifs

- Estimer une **structure** d'un morceau de musique
- Créer automatiquement un **résumé audio** représentatif du contenu du morceau

Applications

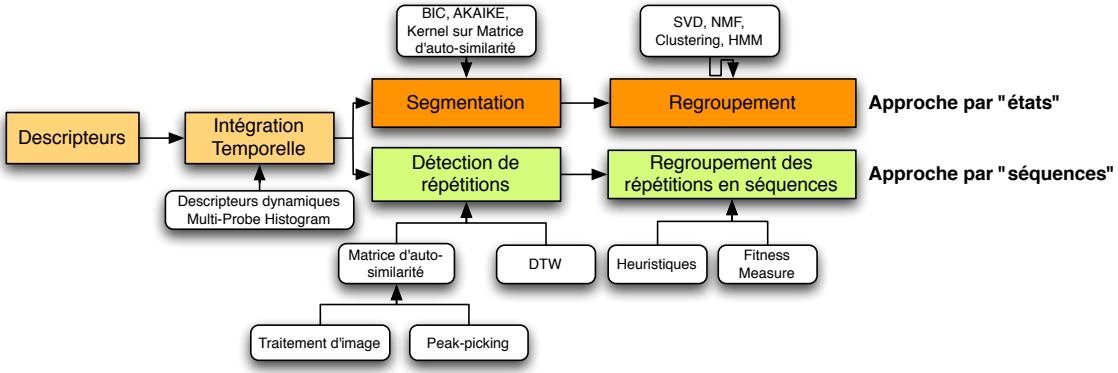
- Ecoute inter-active : création d'interface de lecture (player) interactif,
- Pré-écoute rapide d'un morceau
- **Exemples audio et vidéo**

Méthode

- Extraction de descripteurs audio
- Visualisation de la structure
- Estimation de la structure
  - Apprentissage non-supervisé (pas de pré-apprentissage possible)

source : Quaero project, MSSE-Orange interface

- 1) Extraction d'observations pertinentes du signal audio
  - **Descripteurs audio** : mise en évidence de différents contenus (timbre, harmonique, bruité, ...)
- 2) Analyse des observations afin de détecter une structure
  - Approche par **états**
    - **segmentation** temporelle et
    - **regroupement** des segments homogènes identiques
  - Approche par **séquences**
    - **détection des répétitions** non-homogènes et
    - regroupement des segments répétés en séquences



2- Descripteurs audio

Les descripteurs audio

[G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Cuidado project report, Ircam, 2004.]

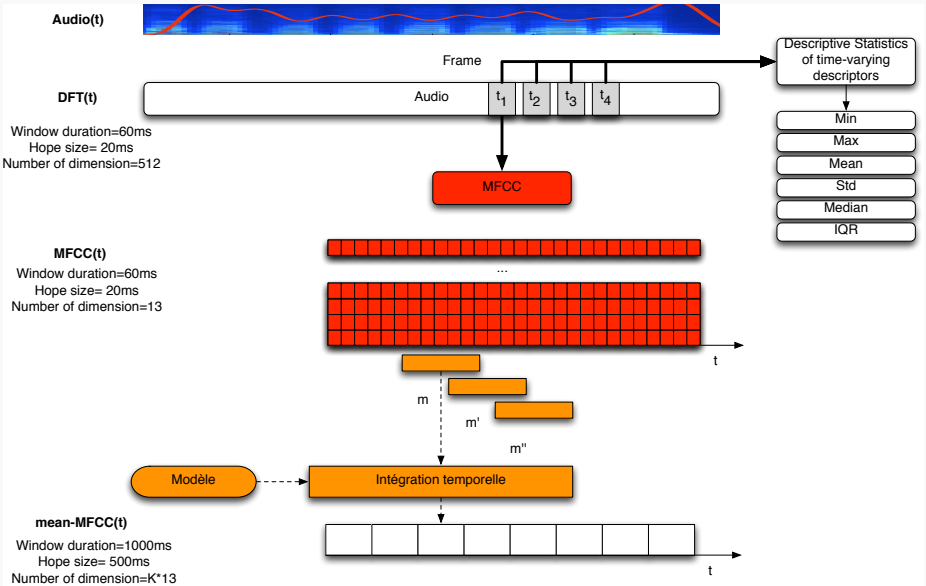
- Valeurs numériques extraites du signal audio dont le but est de représenter une propriété particulière de son contenu
  - Tout est dans la forme d'onde, dans la TFCT, difficile à lire, trop grande dimension
- Contrainte :
  - on veut le même nombre de dimensions pour toutes les données
- Extraction ?
  - Algorithme d'estimation
  - Opérateurs mathématique

Les descripteurs audio

[G. Peeters. A large set of audio features for sound description (similarity and classification) in the cuidado project. Cuidado project report, Ircam, 2004.]

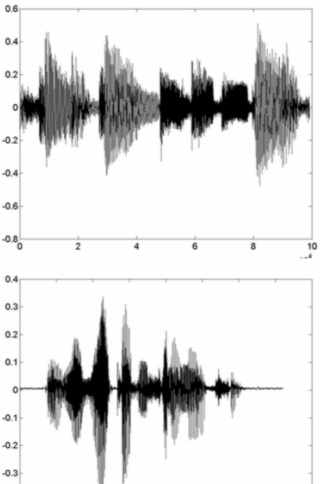
- Différentes **formes** :
  - **scalaire** : Centroide spectral, étendue spectrale, fréquence fondamentale, spectral roll-off, spectral flux, zero-crossing rate, RMS, ...
  - **vecteur** : Mel Frequency Cesptral Coefficients, coefficients LPC, coefficients PLP ...
- Différentes **temporalité** :
  - représente une **trame** du signal audio → descripteurs "instantanés"
  - représente le résumé du contenu d'un **ensemble local de trame** → texture windows
  - représente **globalement** le signal audio
- Mise en évidence de différents **contenus** (, harmonique, bruité, ...)
  - contenu **timbral** : Mel Frequency Cesptral Coefficients, coefficients LPC, coefficients PLP ...
  - contenu **harmonique** : Pitch Class Profiles/ Chroma ...
  - contenu **bruité** : Spectral Flatness Measure
  - contenu **rythmique** : ...

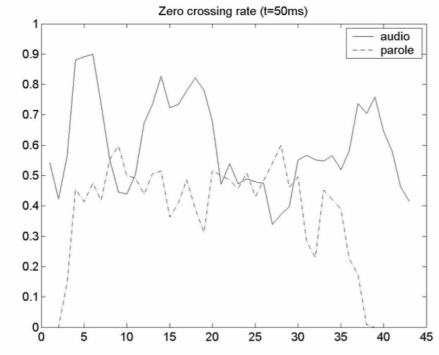
Les descripteurs audio



Taux de passage par zéro / zero-crossing rate (zcr)

- Mesure le nombre de fois que la forme d'onde croise l'axe zéro
  - $zcr = 0.5 \sum_{n=1}^{N} |sign(x(n)) - sign(x(n-1))|$
- Utilisation :
  - permet de distinguer les signaux bruités → zcr élevé
  - permet de distinguer les signaux harmoniques → zcr bas

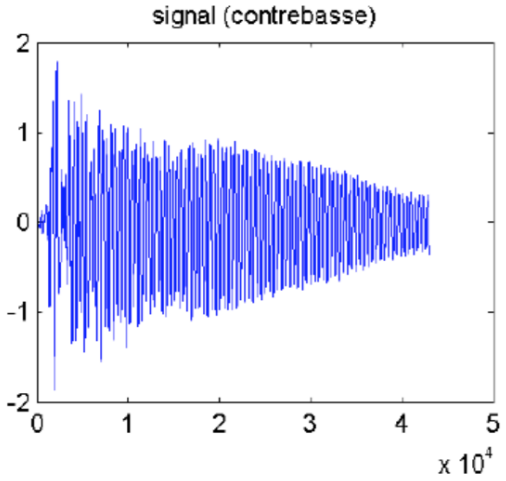


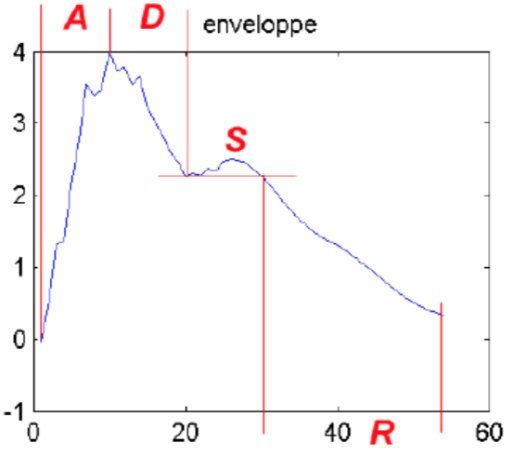


source : Gaël Richard

Enveloppe ADSR (Attack, Decay, Sustain, Release)

- Modèle réprésentant l'évolution (l'enveloppe) d'énergie d'une note de musique
- Utilisation :
  - permet de distinguer les attaques rapides (sons percussifs) / lentes
  - permet de distinguer les décroissances rapides (sons non-tenus) / lentes (sons tenus)

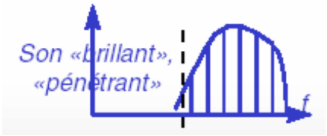




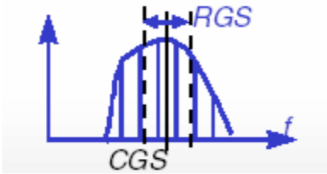
Description du spectre (barycentre, étendue spectral)

- **Centroid spectral**
  - $cs = \frac{\sum_k f_k A_k}{\sum_k A_k}$
  - Utilisation :
    - permet de distinguer les sons terne des sons brilliant
- **Etendue spectral**
  - $es = \sqrt{\frac{\sum_k (f_k - cs)^2 A_k}{\sum_k A_k}}$
  - Utilisation :
    - permet de distinguer les sons pauvres des sons riches
- **Flux spectral**
  - Mesure la variation temporel du spectre
  - $fs = \sum_k (A_k(t) - A_k(t-1))^2$
  - Utilisation :
    - permet de distinguer les sons pauvres des sons riches

source : Gaël Richard



source : Gaël Richard



source : Gaël Richard

Objectif

- décrire la forme du spectre (du timbre) d'un signal à l'aide d'un nombre réduit de coefficients

Cepstre complexe

- **Cepstre complexe** $c(\tau)$:

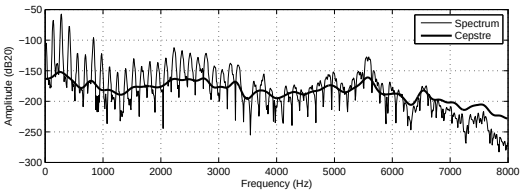
$$\begin{aligned} c(\tau) &= TF^{-1}\left[\log(X(\omega))\right] \\ &= \frac{1}{2\pi}\int_{\omega}\log(X(\omega))e^{j\omega\tau}d\omega \end{aligned} \tag{1}$$

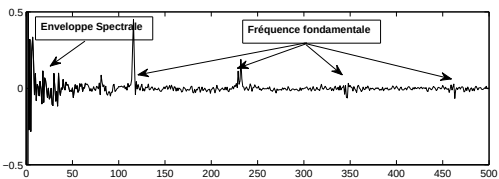
  - τ est appelé "céfrence"
  - $x(t) \xrightarrow{TF} X(\omega) \xrightarrow{\log} \log(X(\omega)) \xrightarrow{TF^{-1}} c(\tau)$

Cepstre complexe

- Modèle source/ filtre :
  - Source : signal périodique
  - Filtre : résonant/ anti-résonant

$$x(t) = e(t) \circledast g(t) \xrightarrow{TF} X(\omega) = E(\omega) \cdot G(\omega) \quad (2)$$





$$\xrightarrow{\log} \log(X(\omega)) = \underbrace{\log(E(\omega))}_{\text{variation rapide à travers } \omega} + \underbrace{\log(G(\omega))}_{\text{variation lente à travers } \omega}$$

$$\xrightarrow{TF^{-1}} TF^{-1}\left[\log(X(\omega))\right] = \underbrace{TF^{-1}\left[\log(E(\omega))\right]}_{\text{énergie aux céfrences } \tau >>} + \underbrace{TF^{-1}\left[\log(G(\omega))\right]}_{\text{énergie aux céfrences } \tau <<} \quad (3)$$

Cepstre réel

- **Cepstre réel** :
  - Cepstre calculé sur la partie réelle du log-spectrum

$$\begin{aligned} X(\omega) &= A(\omega) \cdot e^{j\phi(\omega)} \\ \log(X(\omega)) &= \log(A(\omega)) + j\phi(\omega) \\ \Re(\log(X(\omega))) &= \log(A(\omega)) \end{aligned} \tag{4}$$

$$\begin{aligned} \text{cepstre réel} &= TF^{-1}\left[\Re(\log(X(\omega)))\right] \\ &= TF^{-1}\left[\log(A(\omega))\right] \\ c(\tau) &= \frac{1}{2\pi}\int_{\omega} \log(A(\omega)e^{j\omega\tau}d\omega \end{aligned} \tag{5}$$

- Le spectre d'amplitude étant réel etsymétrique
  - sa TF se réduit à sa partie réelle
    - donc à la projection de $\log(A(\omega))$ sur un ensemble de cosinus $\rightarrow$ DCT

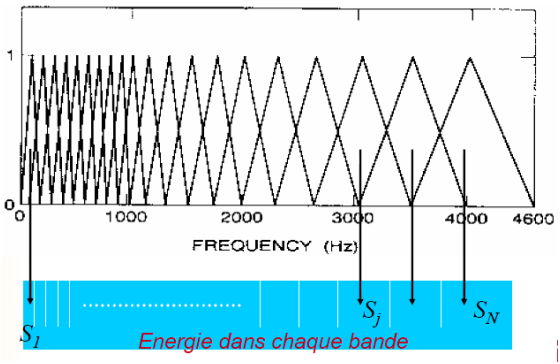
Mel Frequency Cepstral Coefficients (MFCCs)

- **Mel Frequency Cepstral Coefficient** :
  - Cepstre réel calculé sur un spectre d'énergie exprimé en convertissant l'énergie $|X(\omega)|^2$ en échelle perceptive (échelle de Mel)
- Pouquoi ?
  - La transformée de Fourier :
    - décomposition sur une série de sinusoides linéairement espacées (10Hz, 20Hz, 30Hz, ... Hz)
  - L'oreille :
    - décomposition sur une série de filtres de fréquences logarithmiquement espacé (10, 20, 40, 80, ... Hz).
    - meilleure résolution en basses fréquences que en hautes fréquences.
    - résonances de l'enveloppe spectrale sont plus rapprochées en basse fréquence.
  - MFCCs permet une représentation plus compacte que le cepstre réel
- Comment ?
  - On utilise des échelles dites perceptives : échelles de Mel, de Bark, filtres ERB, Gamma tone
- Utilisation ?
  - Les coefficients les plus utilisés dans le monde de la reconnaissance audio : parole, musique, sons environnementaux, ...

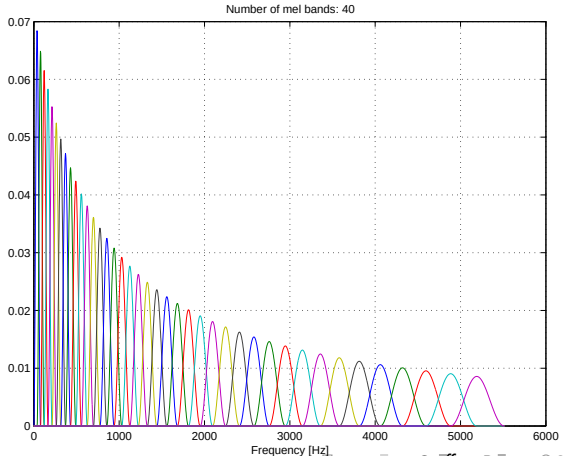
Mel Frequency Cepstral Coefficients (MFCCs)

- Echelle de Mel :

$$M = f \quad \text{pour} \quad f < 1000Hz$$
$$M = f_c \left(1 + \log_{10}\left(\frac{f}{f_c}\right)\right) \quad \text{pour} \quad f \geqslant 1000Hz \tag{6}$$

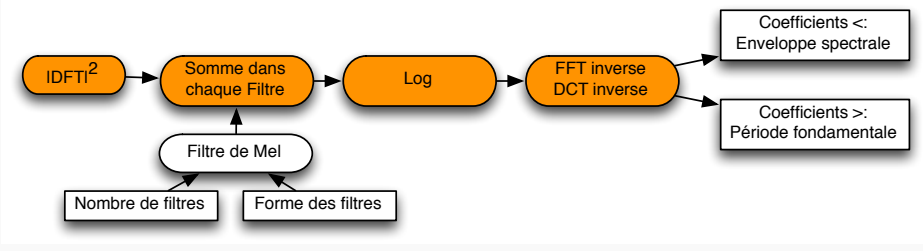


source : Gaël Richard

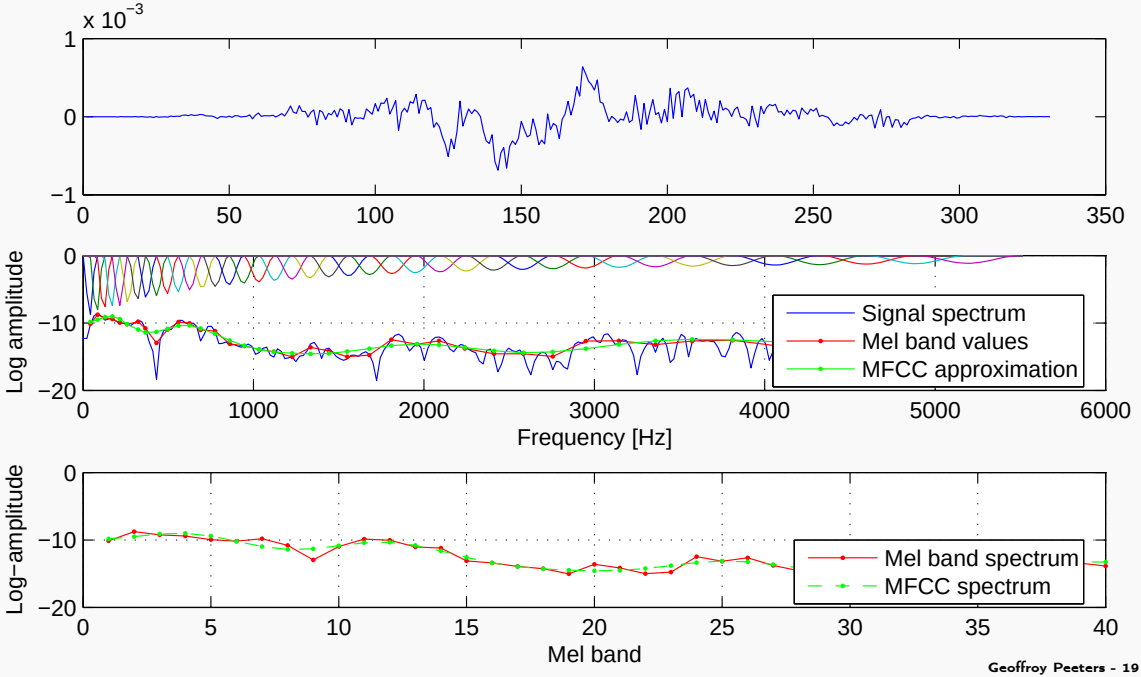


Mel Frequency Cepstral Coefficients (MFCCs)

- Calcul du spectre de puissance : $|X(\omega)|^2$
- Calcul des filtres de Mel : $H_b(\omega)$ avec $b \in [1, B]$
  - choix du nombre de filtres B : 40
  - choix de la forme des filtres : triangulaire, hanning, tanh, ...
- Conversion du spectre de puissance en bandes de Mel : $S(b) = \sum_\omega |X(\omega)|^2 \cdot H_b(\omega)$
- Passage en échelle logarithmique : $\log(S(b))$
- Calcul de la IFFT (ou de la IDCT) :
- Sélection des coefficients de la IDCT proches de zéro (jusqu'à 13)
  - les coefficients proches de zéro représentent la décomposition du spectre en échelle de Mel sur un ensemble de cosinus à variation lente

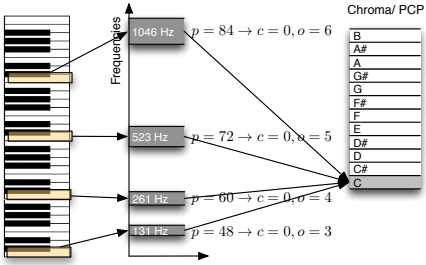


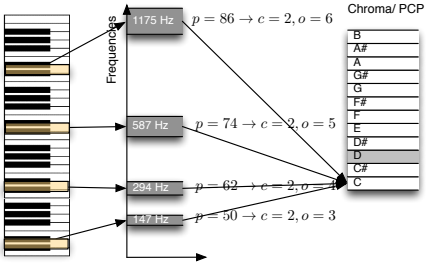
Exemple de calcul de MFCCs



Définition des Chroma - Pitch Class Profile (PCP)

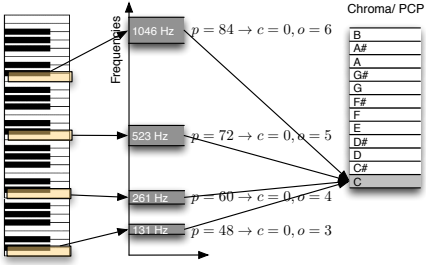
- **Objectif :**
  - le spectre à l'instant n : $X(k, n)$
  - représenter son contenu harmonique sous forme d'un vecteur : $C(c, n)$ $c \in [0, 12[$
- Utilisations :
  - reconnaissance de tonalité,
  - reconnaissance de suite d'accords,
  - détection de "cover versions"
- Shepard-1964 :
  - représenter la hauteur d'une note p comme une structure bi-dimensionelles :
  - $p = c + o \cdot 12$
    - le chroma c (classe de hauteur).
    - la hauteur tonale o (numéro d'octave),

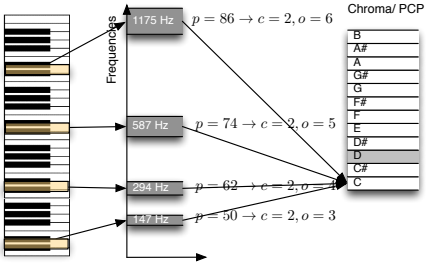




Calcul des Chromas - Pitch Class Profile (PCP)

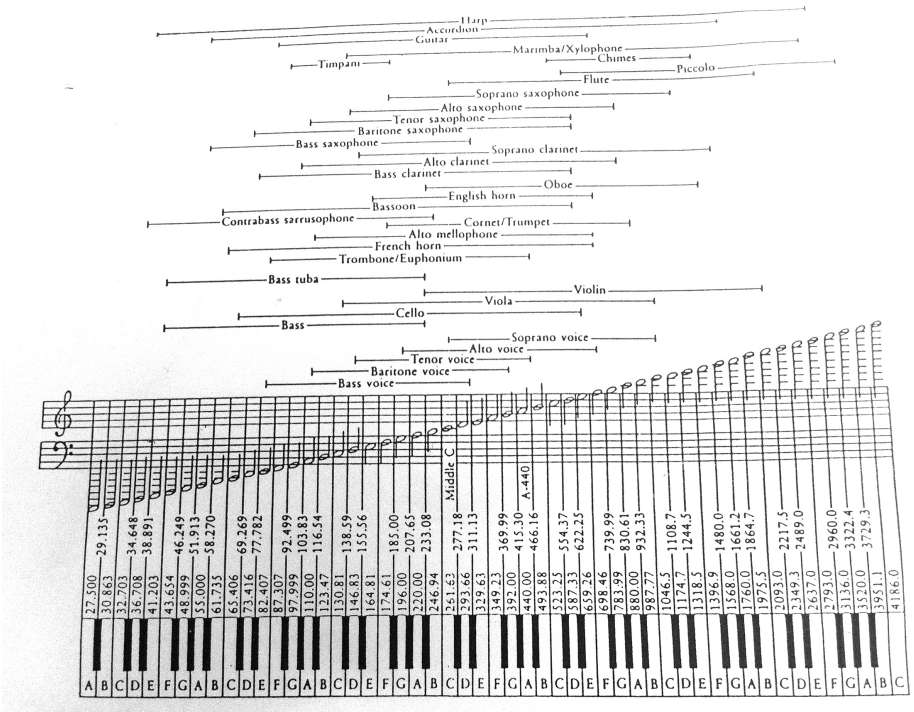
- Relation entre les fréquences f_k de la DFT et les hauteurs de note p (hauteurs de demi-tons en échelle de notes MIDI)
  - $p(f_k) = 12 \log_2 \left(\frac{f_k}{440}\right) + 69, \quad p \in \mathbb{R}^+$
  - $f(p) = 440 \cdot 2^{\frac{p-69}{12}}$
- Calcul des chromas $C(c, n)$
  - On additionne toutes les valeurs du spectre $X(k, n)$ tel que f_k correspondent à un c donné
  - Hard-mapping
  - Soft-mapping





Calcul des Chromas - Pitch Class Profile (PCP)

- Résolution fréquentielle ?
  - Elle doit permettre la séparation des notes voisines
  - On définit la largeur (à -6 dB) : $Bw = \frac{Cw}{L_{sec}}$
  - Si $f_{\min}$ (la fréquence la plus basse considérée dans le sectre) est 50 Hz
    - on veut séparer G#1 (51.91Hz) et A1 (55Hz) $\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{3.0869Hz} = 0.7613s$
  - Si $f_{\min}$ est 100 Hz
    - on veut séparer G#2 (103.82Hz)de A2 (110Hz) $\rightarrow L_{sec} = \frac{Cw}{Bw} = \frac{2.35}{6.1738Hz} = 0.3806s$
- Deux possibilités :
  - Choisir L_{sec} en fonction $f_{\min}$
  - Choisir $f_{\min}$ en fonction de L_{sec}

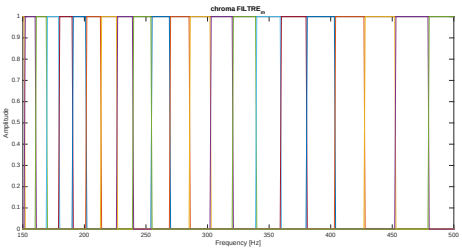

Harp
Accordion
Guitar
Marimba/Xylophone
Timpani
Chimes
Piccolo
Flute
Soprano saxophone
Alto saxophone
Tenor saxophone
Baritone saxophone
Bass saxophone
Soprano clarinet
Alto clarinet
Bass clarinet
Oboe
English horn
Bassoon
Contrabass sarrusophone
Cornet/Trumpet
Alto mellophone
French horn
Trombone/Euphonium
Bass tuba
Violin
Viola
Cello
Bass
Soprano voice
Alto voice
Tenor voice
Baritone voice
Bass voice
Middle C
A-440
A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C D E F G A B C
27.500 29.135 30.863 32.703 34.648 36.708 38.891 41.203 43.654 46.249 48.999 51.913 55.000 58.270 61.735 65.406 69.269 73.416 77.782 82.407 87.307 92.499 97.999 103.83 110.00 116.54 123.47 130.81 138.59 155.56 164.81 174.61 185.00 196.00 207.65 220.00 233.08 246.94 261.63 277.18 293.66 311.13 329.63 349.23 369.99 392.00 415.30 440.00 466.16 493.88 523.25 554.37 587.33 622.25 659.26 698.46 739.99 783.99 830.61 880.00 932.33 987.77 1046.5 1108.7 1174.7 1244.5 1318.5 1396.9 1480.0 1568.0 1661.2 1760.0 1864.7 1975.5 2093.0 2217.5 2349.3 2489.0 2637.0 2793.0 2960.0 3136.0 3322.4 3520.0 3729.3 3951.1 4186.0

Calcul des Chromas - Pitch Class Profile (PCP)

- Calcul des chromas $C(c, n)$
  - On additionne toutes les valeurs du spectre $X(k, n)$ tel que f_k correspondent à un c donné

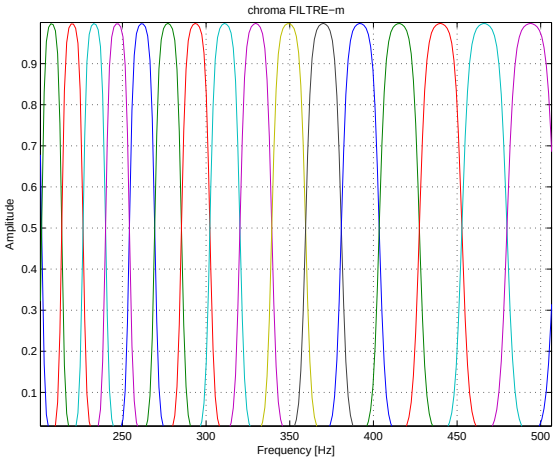
Hard-mapping

- Hard-mapping ?
  - Une fréquence f_k de la DFT contribue uniquement à la note la plus proche
  - Par exemple,
    - l'énergie à f_k=452 Hz ($p(f_k)$=69.4658) contribue entièrement à la note p=69 (c=10)
    - alors que f_k=453 Hz ($p(f_k)$=69.5041) à p=70 (c=11).
- Création d'un banc de filtres $H_{p'}$ centrés sur les hauteurs de demi-tons $p' \in [43, 44, \ldots, 95]$:



Soft-mapping

- Soft-mapping ?
  - Une fréquence f_k de la DFT contribue à différents chroma avec un poids inversement proportionnel à la distance entre $p(f_k)$ et les p les plus proches
  - Par exemple,
    - l'énergie à f_k=452 Hz ($p(f_k)$=69.4658) contribuera de manière presque égale à p=69 (c=10) et p=70 (c=11).
- Création d'un banc de filtres $H_{p'}$ centrés sur les hauteurs de demi-tons $p' \in [43, 44, \ldots, 95]$:

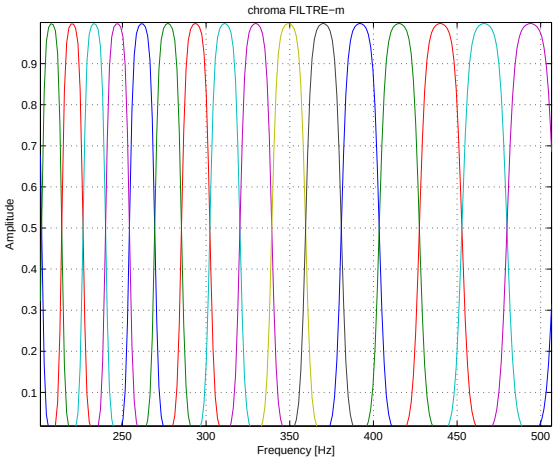


Soft-mapping

- Création d'un banc de filtres $H_{p'}$ centrés sur les hauteurs de demi-tons $p' \in [43, 44, \ldots, 95]$:
  - Chaque filtre est défini par la fonction

$$H_{p'}(f_k) = \frac{1}{2}\tanh\left(\pi(1-2x)\right) + \frac{1}{2}$$

    dans lequel x= distance relative entre centre du filtre et fréquences de la TF $x = R\,|p' - p(f_k)|$.
  - Les filtres sont équi-répartis et symétriques sur l'échelle logarithmique des hauteurs de demi-tons, non-nulles entre $p'-1$ et $p'+1$ et à valeur maximale en p'.



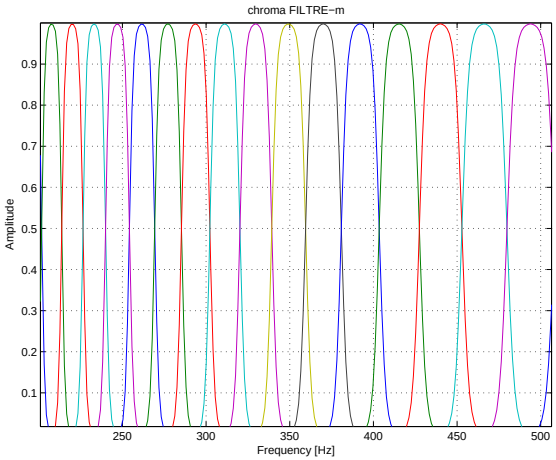
Calcul des Chromas - Pitch Class Profile (PCP)

- La valeur du spectre de hauteur de demi-ton $N(n')$ est obtenue en multipliant les valeurs de la transformée de Fourier $A(f_k)$ par l'ensemble des filtres $H_{n'}$:

$$P(p') = \sum_{f_k} H_{p'}(f_k) A(f_k)$$

- Le mapping entre les hauteurs de demi-tons n et les classes de hauteurs de demi-ton (chroma) c est défini par $c(p) = mod\,(p, 12)$.
- La valeur du vecteur de chroma est obtenue en additionnant les valeurs de classes de hauteur équivalentes

$$C(c) = \sum_{p' \text{ tel que } c(p')=I} P(n') \quad c \in [0, 12[$$



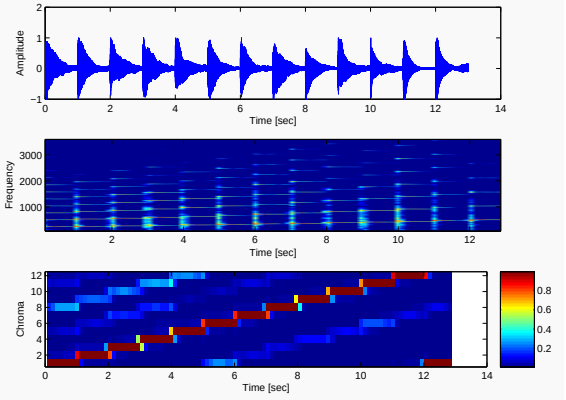
Limitations des Chromas - Pitch Class Profile (PCP)

- Présence des harmoniques supérieures de chaque note
  - En pratique pour une note C on a pas $[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
  - mais plutôt $[a_1 + a_2 + a_4, 0, 0, 0, a_5, 0, 0, a_4, 0, 0, 0, 0]$
- Influence de l'enveloppe spectrale

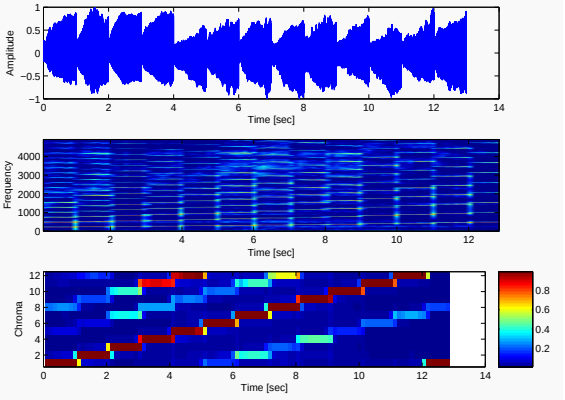
Pitch	Harmonic	Frequency f_μ	MIDI-scale m_μ	Chroma/PCP p
c3	f_0	130.81	48	1 (=c)
	$2f_0$	261.62	60	1 (=c)
	$3f_0$	392.43	67.01	8.01 ($\simeq$ g)
	$4f_0$	523.25	72	1 (=c)
	$5f_0$	654.06	75.86	4.86 ($\simeq$ e)
	...	...	...	...

Limitations des Chromas - Pitch Class Profile (PCP)

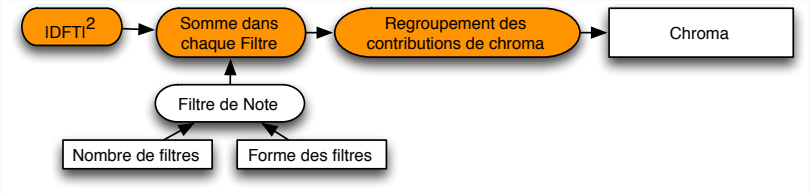
Exemple piano

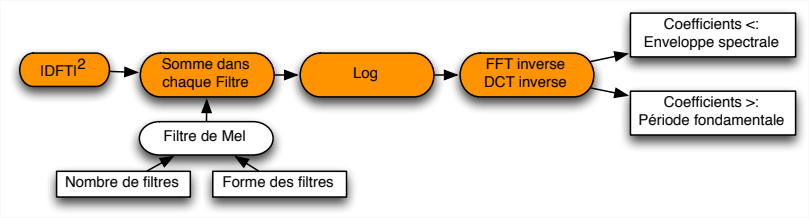


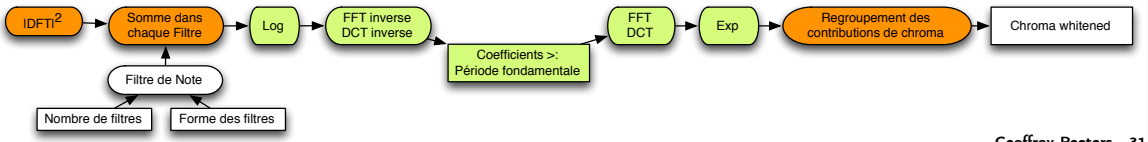
Exemple violon



Variante du calcul des Chromas - Pitch Class Profile (PCP) : blanchissement/ whitening







Spectral Flatness Measure (SFM)

- **Objectif :** distinguer la présence de contenu harmonique ou bruité dans chaque bande
  - avec les MFCCs/PCP même valeur si le contenu est harmonique ou bruité dans une bande du spectre

- **Spectral Flatness Measure** : mesure de la platitude d'une bande du spectre
  - Si la bande du spectre contient du bruit → spectre plat (flat)
  - Si la bande du spectre contient des sinusoides → spectre avec des piques (peaky)
  - Calcul [**?**] : rapport moyenne géométrique / moyenne arithmétique

$$SFM = \frac{\left(\prod_{k\in K} a(k)\right)^{1/K}}{\frac{1}{K}\sum_{k\in K} a(k)} \tag{7}$$

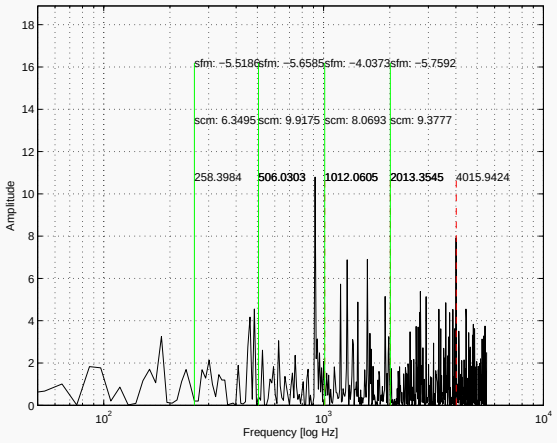
  - SFM$\simeq$0 pour signaux tonaux, SFM$\simeq$1 pour signaux bruités
  - Calcul effectué dans plusieurs bandes de fréquence :
    - $[250-500]$, $[500-1000]$, $[1000-2000]$, $[2000-4000]$ Hz (MPEG-7)

- **Mesure de tonalité** :

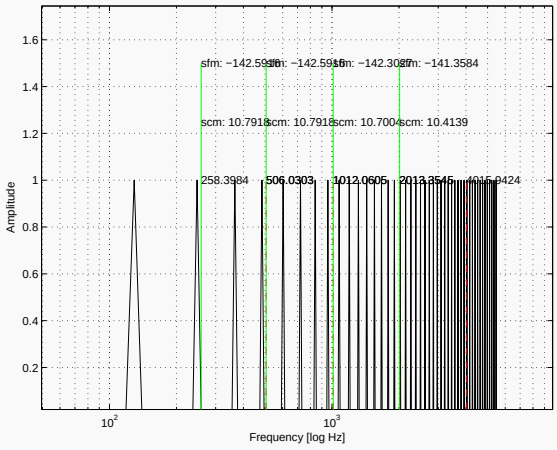
$$SFM_{dB} = 10\log_{10}(SFM) \qquad Tonality = \min\left(\frac{SFM_{dB}}{-60}, 1\right) \tag{8}$$

  - Tonality$\simeq$0 pour signaux bruités, Tonality$\simeq$1 pour signaux tonaux

Exemple cas bruité

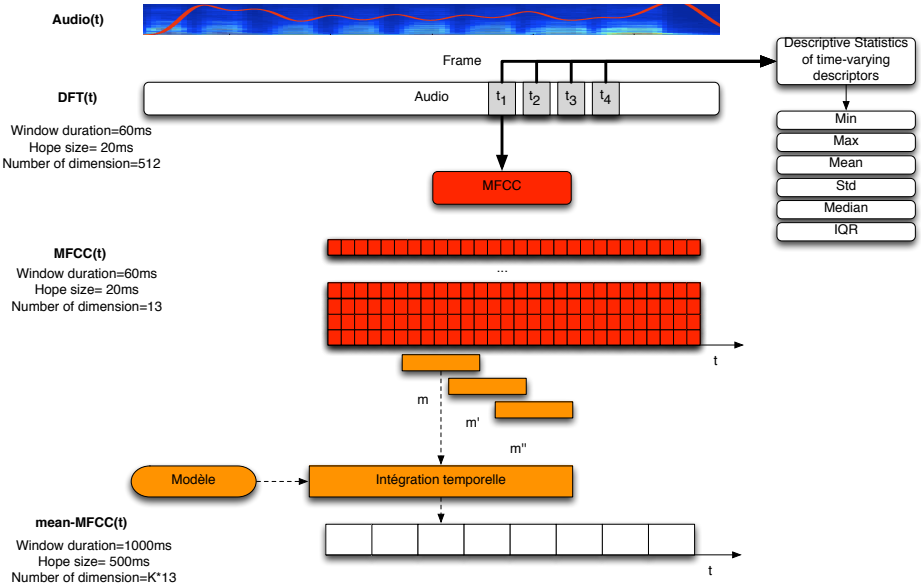


Exemple cas non-bruité



Intégration temporelle

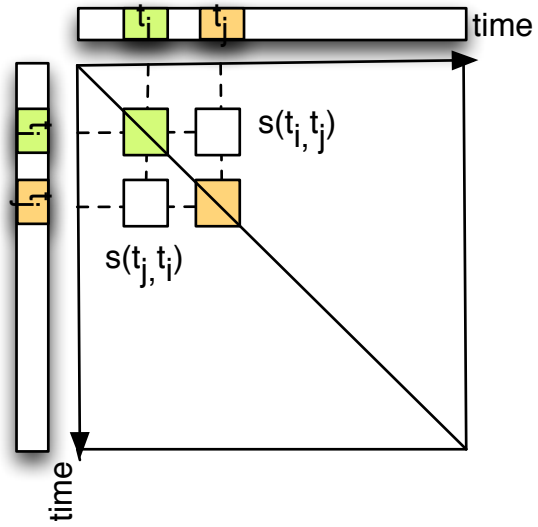
- **Objectifs :**
  - Représenter le comportement temporel des observations
    - Calcul des **dérivées et accélérations** des observations (Δ-MFCC, Δ-Δ-MFCC) $\rightarrow$ permet de représenter le comportement temporel du descripteur au cours du temps
  - Réduire la quantité de données à traiter
    - Si le pas d'avancement = 20 ms, un morceau de 4 m. = 12.000 trames
    - $\rightarrow$ matrice d'auto-similarité = 12.000 x 12.000 $\rightarrow$ c'est beaucoup !
- **Intégration sans-modèles**
  - Analyse des descripteurs sur une fenêtre de durée plus longue (0.5 s., 1 s., ...)
    - Calcul des **moments statistiques** (μ, σ) de chaque dimension k d'un descripteur (chaque coefficient MFCC, PCP, SFM, ...)
    - modulation spectrum, scattering transform
    - modèles AR
- **Intégration avec modèles**
  - Multi-prob histogram
  - Universal Background Model, iVector


Audio(t)
Frame
Descriptive Statistics of time-varying descriptors
DFT(t)
Audio
t1
t2
t3
t4
Min
Max
Mean
Std
Median
IQR
Window duration=60ms
Hope size= 20ms
Number of dimension=512
MFCC
MFCC(t)
Window duration=60ms
Hope size= 20ms
Number of dimension=13
...
t
m
m'
m"
Modèle
Intégration temporelle
mean-MFCC(t)
Window duration=1000ms
Hope size= 500ms
Number of dimension=K*13
t

Représentation visuelle de la structure temporelle de la musique

La matrice d'auto-similarité

- Similarité entre deux instants t_i et t_j
- Similarité entre les observations du signal aux trames i et j : $s(t_i, t_j) = s(\underline{d}^i, \underline{d}^j)$
- Matrice d'auto-similarité= les valeurs $s(t_i, t_j)$ sont représentées sous forme d'une matrice $\underline{\underline{S}} = s(t_i, t_j) \quad \forall i, j$

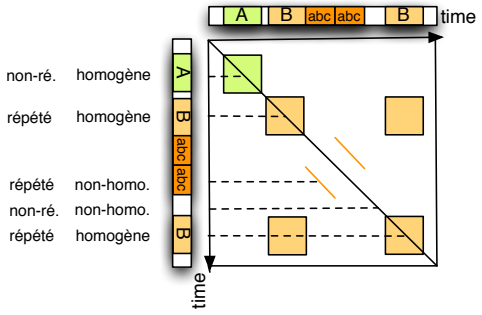


Lecture/ interprétation

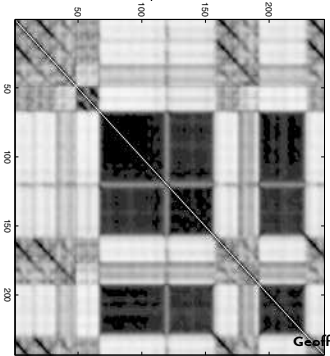
- Une valeur élevée dans $S(t_i, t_j)$ représente une similarité importante entre les instants t_i et t_j.
- Si $t_i \simeq t_{i+1} \simeq t_{i+2}$ nous observons un **bloque homogène**
- Si une **séquence de temps** $t_i, t_{i+1}, t_{i+2}, ...$ est similaire à une séquence de temps $t_j, t_{j+1}, t_{j+2}, ...$ nous observons une diagonale supérieure/ inférieure dans S.

Hypothèse 1 : homogénéité

- Hypothèse : le morceau est formé d'une succession de segments temporels **homogènes** $t_i \simeq t_{i+1} \simeq t_{i+2}, \ldots$ et de segments non homogènes
  - homogène ? contenant une information similaire au sens d'un critère d'observation)
  - "A" et "B" sur la Figure
- Exemple : arrangements d'un couplet ou d'un refrain
- Méthode : **approche par "état"**

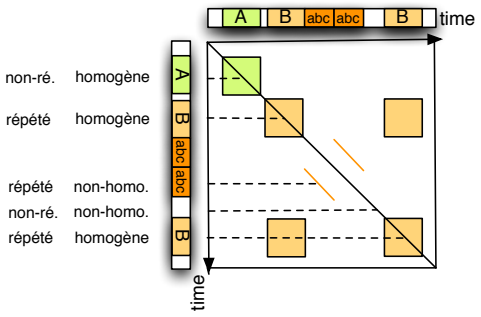


	homogène	non-homogène
répété	état	séquence
non-répété	état	état

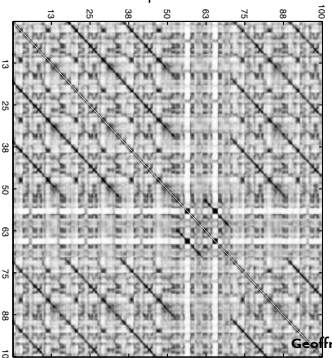


Hypothèse 2 : répétition

- Hypothèse : le morceau renferme des **répétitions** temporelles.
  - elles peuvent correspondre à des répétitions de segments **homogènes**
    - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ et $t_i \simeq t_{i+1} \simeq t_{i+2}$
    - "B" dans la figure
    - Méthode : **approche par "état"**
  - elles peuvent correspondre à des répétitions de segments **non homogènes**
    - $\{t_j, t_{j+1}, t_{j+2}\} \simeq \{t_i, t_{i+1}, t_{i+2}\}$ et $t_i \neq t_{i+1} \neq t_{i+2}$
    - séquence "abc" dans la Figure
    - Méthode : **approche par "séquence"**



	homogène	non-homogène
répété	état	séquence
non-répété	état	état



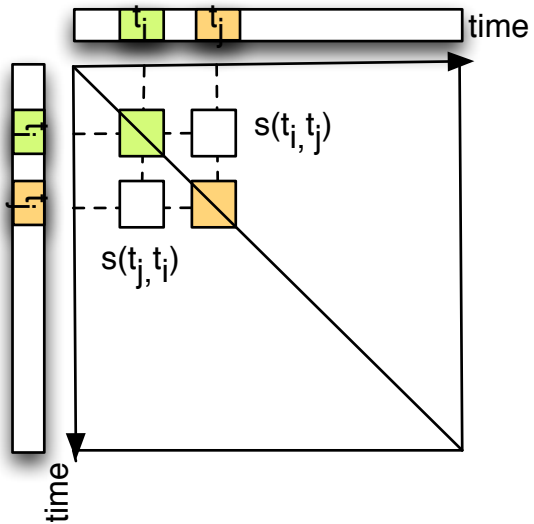
3- Représentation visuelle de la structure temporelle de la musique

Matrice d'auto-similarité/distance (temps,temps)

- Similarité entre deux instants t_i et t_j
- Similarité entre les observations du signal à deux trames i et j : $s(t_i, t_j) = s(\underline{d}^i, \underline{d}^j)$
- Descripteurs audio multi-dimensionnels $\underline{d} = d_k \ k \in K$

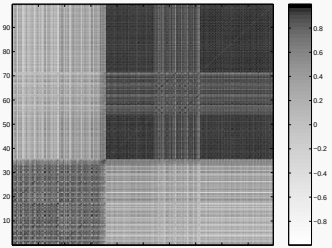
Choix d'une distance

- Distance euclidéenne : $\sqrt{\sum_k (d_k^i - d_k^j)^2}$
- Corrélation : $\sum_k (d_k^i \cdot d_k^j)$
- Distance cosinusoidale : $\frac{\sum_k (d_k^i \cdot d_k^j)}{\sqrt{\sum_k (d_k^i)^2}\sqrt{\sum_k (d_k^j)^2}}$
- Correlation Pearson : $\frac{\sum_k (d_k^i - \mu^i)\cdot(d_k^j - \mu^j)}{\sqrt{\sum_k (d_k^i - \mu^i)^2}\sqrt{\sum_k (d_k^j - \mu^j)^2}}$
- ...

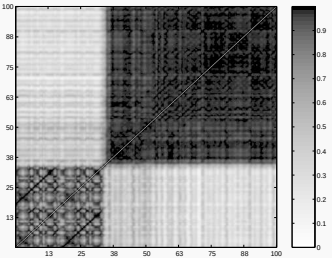


3- Représentation visuelle de la structure temporelle de la musique
Matrice d'auto-similarité/distance (temps,temps)

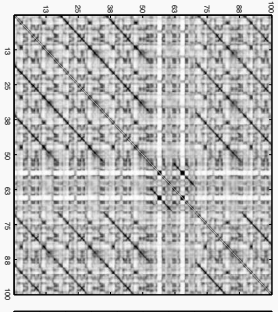
MFCC



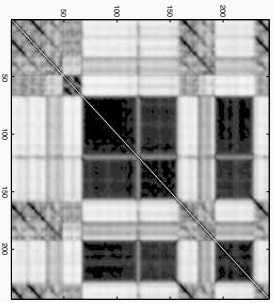
Modulation spectrum 1



Chroma

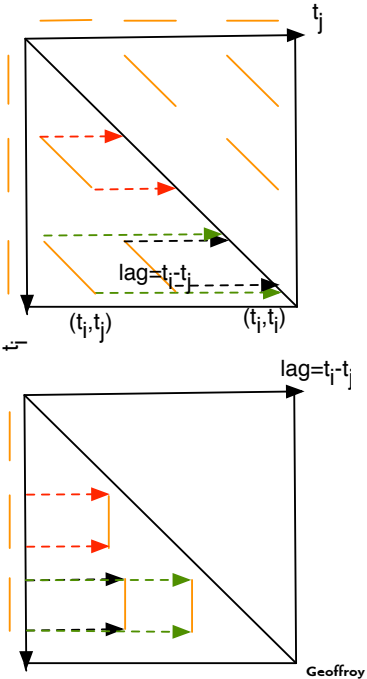


Modulation spectrum 2



Matrice d'auto-similarité/distance (temps,lag)

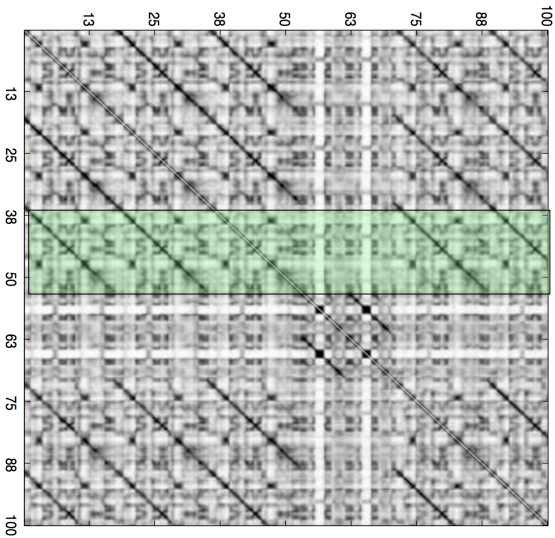
- Une valeur élevée dans $S(t_i, t_j)$ représente une similarité importante entre les instants t_i et t_j.
- Si une séquence de temps $t_i, t_{i+1}, t_{i+2}, ...$ est similaire à une séquence de temps $t_j, t_{j+1}, t_{j+2}, ...$ nous observons une diagonale supérieure/ inférieure dans S.
- **Lag** = distance entre la répétition (démarrant en t_i) et la séquence originale (démarrant en t_j)
  - cette distance est donnée par la projection de t_i sur la diagonale principale de la matrice : $t_i - t_j$
  - souvent constante
- Matrice de lag :
  $L = L(t_i, lag_{ij}) = S(t_i, t_i - t_j)$
  - les diagonales dans une matrice (temps,temps)
  - deviennent des lignes verticales dans une matrice (temps,lag)



Génération de résumé audio par méthode du “summary score"

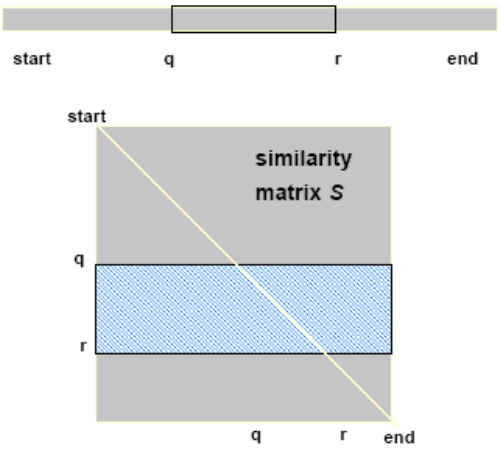
[M. Cooper and J. Foote. Automatic music summarization via similarity analysis. In Proc. of ISMIR, Paris, France, 2002.]

- Recherche du segment temporel continu représentant au mieux le contenu d'un morceau de musique selon un critère de similarité → création de “previews” musicaux

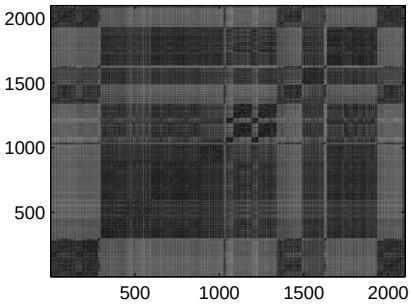


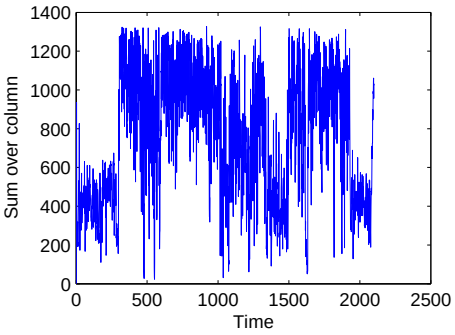
Génération de résumé audio par méthode du "summary score"

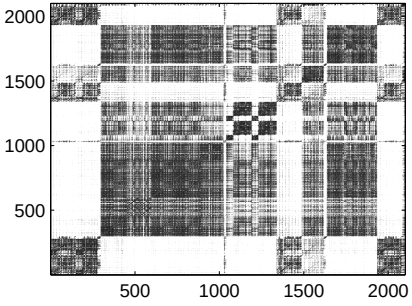
- Recherche du segment démarrant en q de durée $L = r - q$ expliquant le maximum de répétitions
- Similarité moyenne **de l'instant** q avec tous les temps du morceau
  - $\frac{1}{N} \sum_{n=1}^{N} S(q, n)$
- Similarité moyenne **du segment** $[q, r]$ (de longueur $L = r - q$) avec tous les temps du morceau
  - $s(q, L) = \frac{1}{LN} \sum_{m=q}^{r} \sum_{n=1}^{N} S(m, n)$
- Pour un L donné, nous cherchons q maximisant $s(q, L)$
  - $q_L = \text{argmax}_{1 \leqslant i \leqslant N-L} s(i, L)$
- Variante : pour favoriser la détection de résumés en début de morceau,
  - ajout d'une pondération $w(n)$ fonction décroissante du temps
  - $s(q, L) = \frac{1}{LN} \sum_{m=q}^{r} \sum_{n=1}^{N} w(n) S(m, n)$

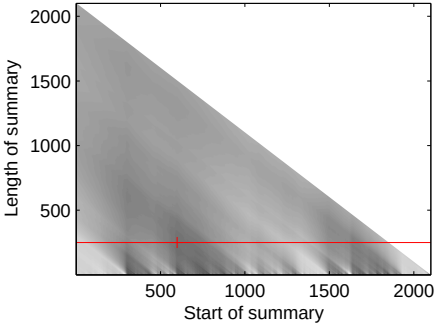


source : [Cooper and Foote, 2002, ISMIR]


2000
1500
1000
500
500
1000
1500
2000


1400
1200
1000
800
600
400
200
0
Sum over column
0
500
1000
1500
2000
2500
Time


2000
1500
1000
500
500
1000
1500
2000


2000
1500
1000
500
Length of summary
500
1000
1500
2000
Start of summary

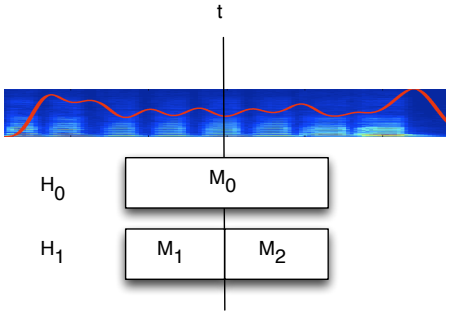
4- Segmentation temporelle d'un flux de descripteurs

Segmentation trame-à-trame

- Variation trame-à-trame de $\underline{d}^t$

Critère BIC (Bayes Information Criteria)

- Pour chaque temps t (potentiellement instant de rupture) on compare deux hypothèses
  - H0 : le signal obéit au même modèle probabiliste de part et d'autre de t, modèle noté $M_0(\mu_0, \Sigma_0)$
  - H1 : il y a un changement de modèle en t, deux modèles différents $M_1(\mu_1, \Sigma_1)$ et $M_2(\mu_2, \Sigma_2)$



- Critère Delta BIC

$$\Delta BIC = R(t) - \lambda P$$

$$R(t) = \frac{1}{2}(N \log(|\Sigma_0|) - t \log(|\Sigma_1|) - (N-t) \log(|\Sigma_2|)$$

- si $\Delta BIC > 0$, H1 est vérifiée
- paramètres :
  - P : proportionnel à la différence des nombres de paramètres estimés pour chaque hypothèse
  - λ : facteur de pénalité choisi tel que $\Delta BIC > 0$ si H1 est vérifiée

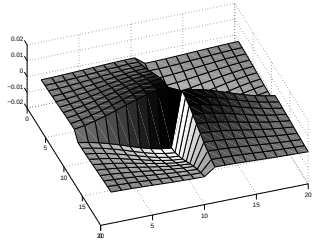
Convolution de la matrice d'auto-similarité par un noyau en damier

[J. Foote. Automatic audio segmentation using a measure of audio novelty. In Proc. of IEEE ICME, New York City, NY, USA, 2000.]

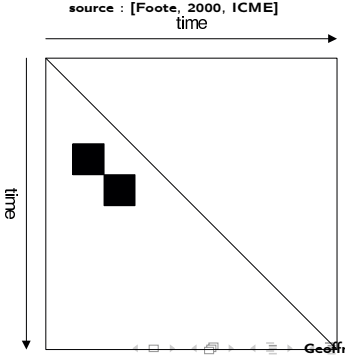
- Méthode "Novelty Curve" [Foote, 2000, ICME]
- Approche plus robuste
- Convolution de la matrice de similarité $\underline{\underline{S}}$ par un noyau prenant en compte
  - la similarité inter-segment (homogénéité) et
  - la dis-similarité entre **segments** gauches et droites
  - **"checker-board" kernel** :

$$C = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

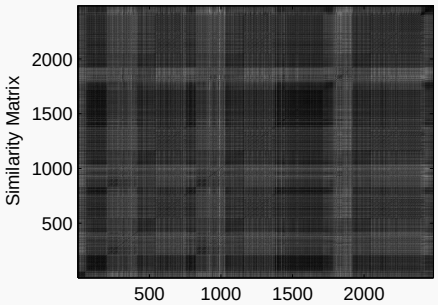
  - La valeur de la diagonale de la matrice "filtrée" mesure la similarité/ dis-similarité des **régions** gauches et droites

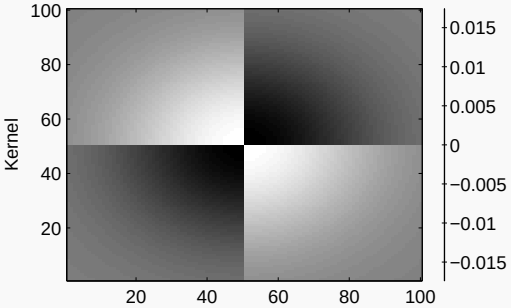


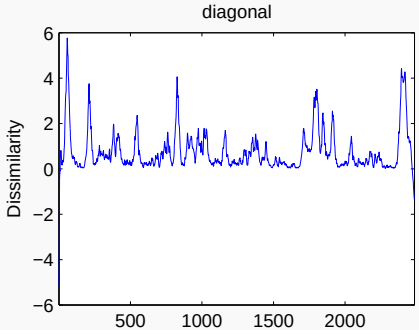
source : [Foote, 2000, ICME]

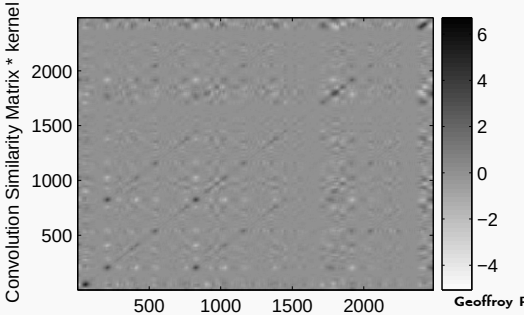


Exemple


Similarity Matrix
2000
1500
1000
500
500
1000
1500
2000


Kernel
100
80
60
40
20
20
40
60
80
100
0.015
0.01
0.005
0
−0.005
−0.01
−0.015


diagonal
Dissimilarity
6
4
2
0
−2
−4
−6
500
1000
1500
2000


Convolution Similarity Matrix * kernel
2000
1500
1000
500
500
1000
1500
2000
6
4
2
0
−2
−4

Clustering ?

- Processus qui partitionne un ensemble de données en sous-classes (clusters) ayant du sens
- Algorithme permettant de trouver la structure sous-jacente à un ensemble de données
- Apprentissage non-supervisé (par opposition à l'apprentissage supervisé : Bayes, ...)

- Deux grandes classes d'algorithmes :

Algorithmes de **partitionnement** :

divise un ensemble de N items en K clusters, tous les clusters sont considérés simultanément

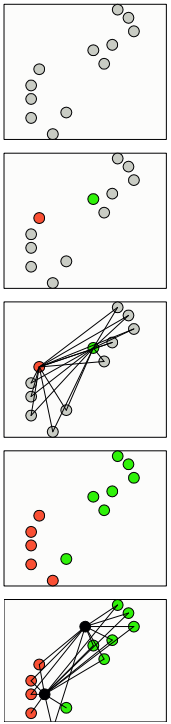
- K-means
- Fuzzy-K-Means
- GMM

Algorithme **hiérarchiques** :

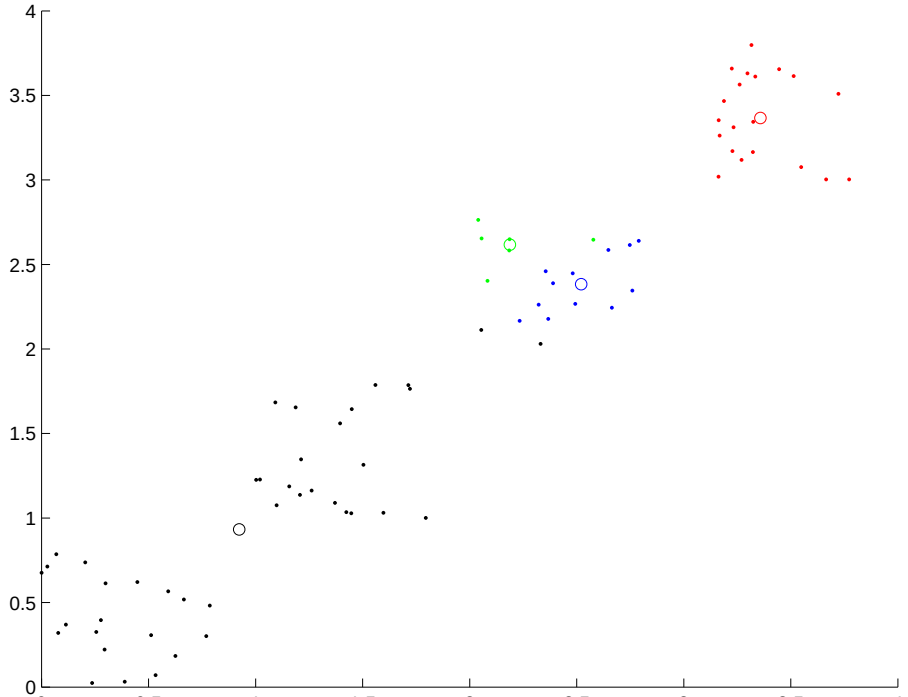
- Par **agglomération** :
  - les paires d'objets ou de clusters sont successivement liées pour produire des clusters plus grand (bottom-up)
- Par **division** :
  - les clusters sont successivement divisés en de plus petits clusters (top-down)

Algorithmes de partitionnement : K-Means (nuées dynamiques)

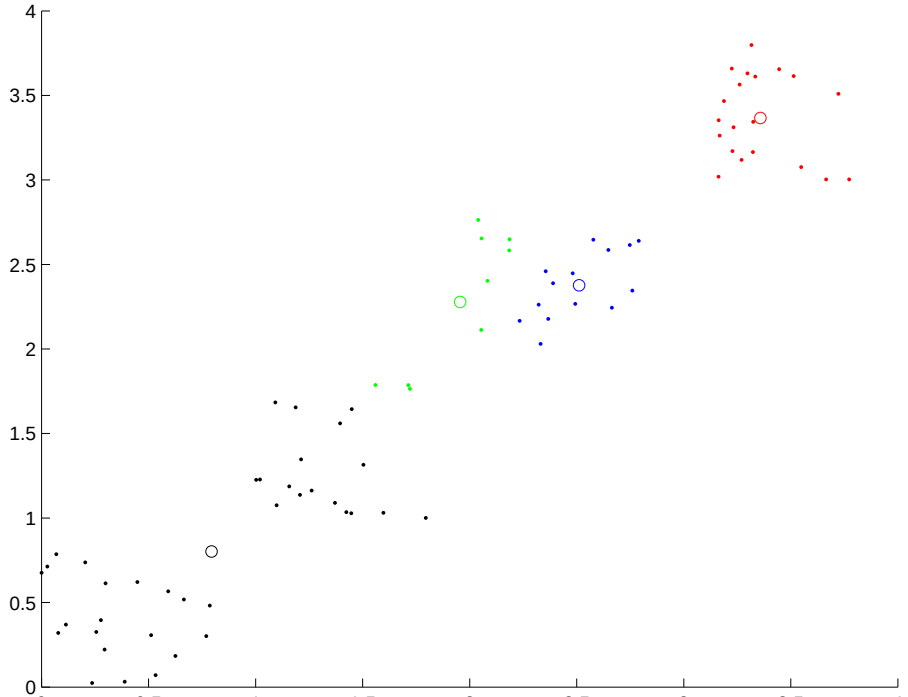
- **Initialisation :**
  - choisir K subsets (clusters) initiaux
  - calculer le centroide de chaque subset (cluster) à partir des objets attribués à ce subset (cluster)
  - différentes méthodes pour l'initialisation : random, KD-tree ...
- **Etape E (Expectation) :** attribuer chaque objet au subset (cluster) dont il est le plus proche (distance euclidienne)
- **Etape M (Maximization) :** étant donné la nouvelle attribution des objets aux subsets (clusters) recalculer les centroides (moyenne arithmétique)
- **Itération :** réitérer jusqu'à ce que les objets ne bougent plus (ou que la valeur des centroides ne bougent plus)



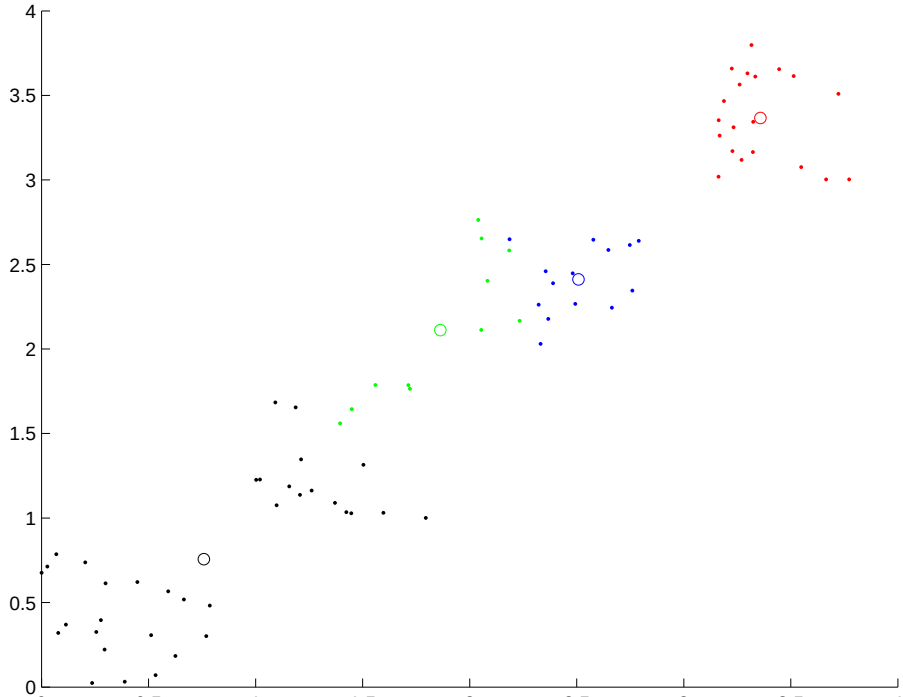
1er iteration



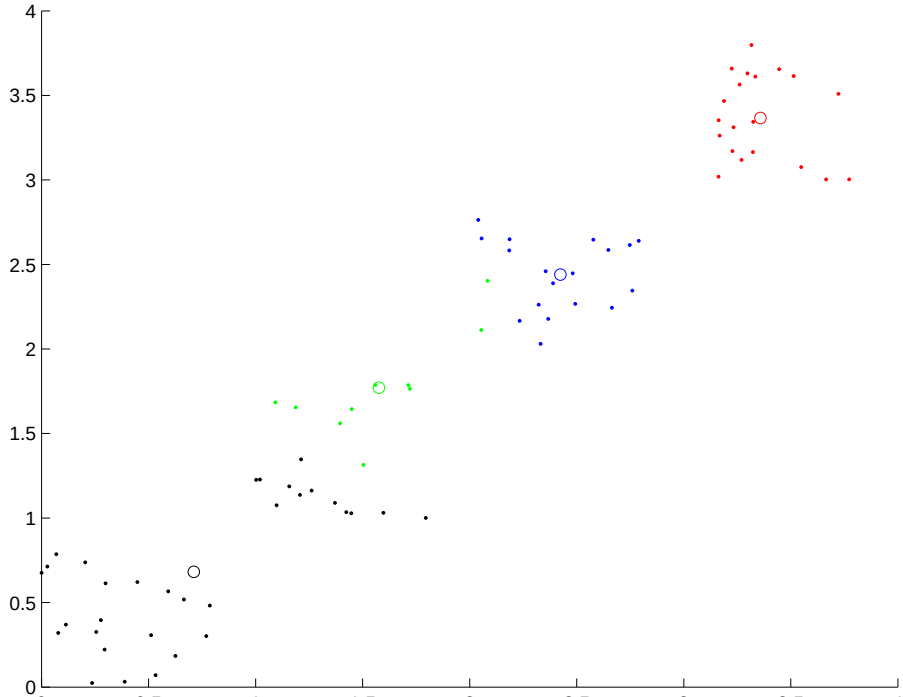
2em iteration



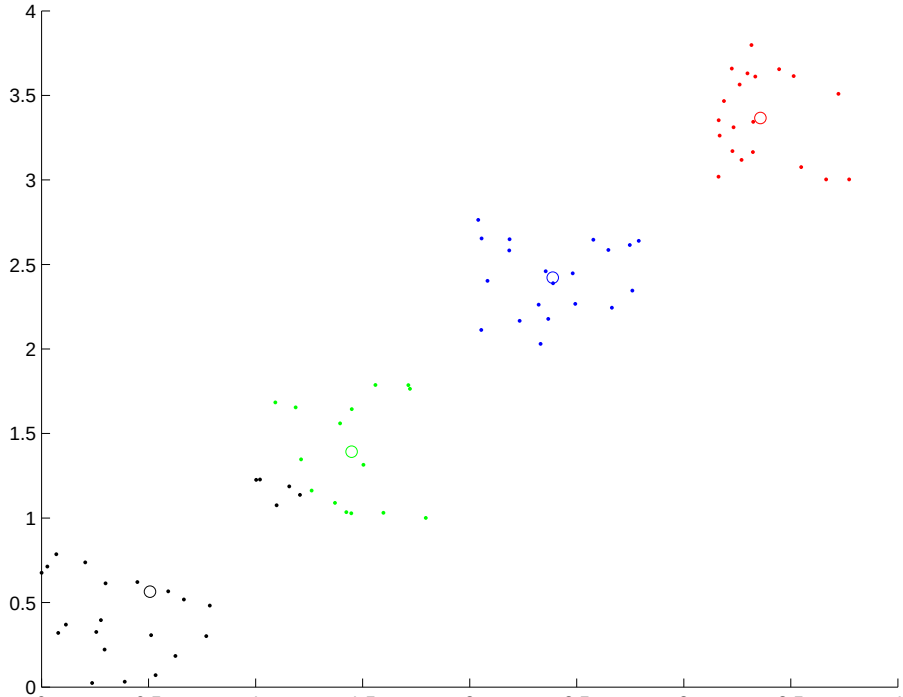
3em iteration



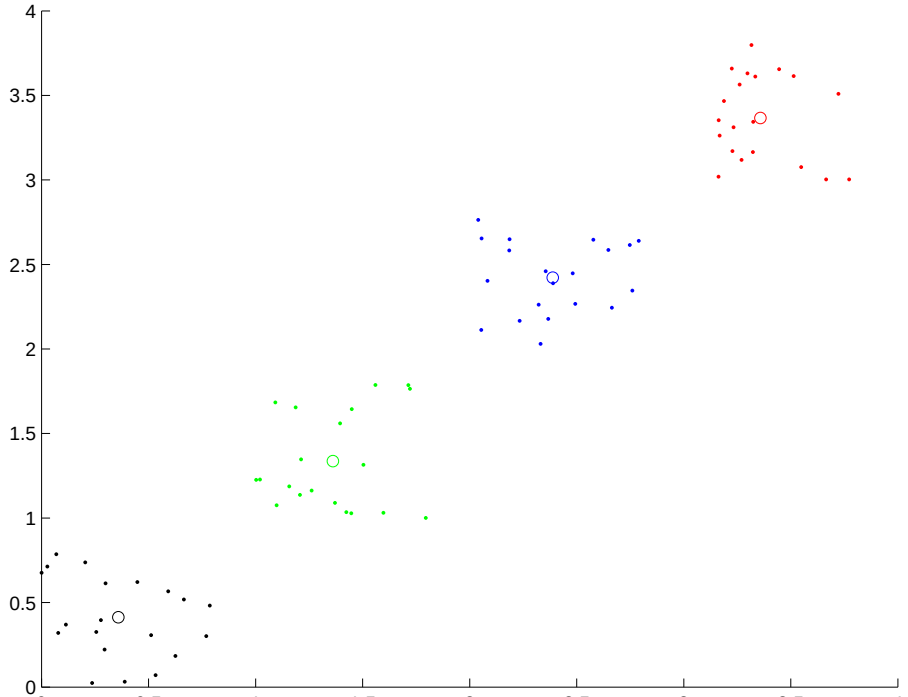
4em iteration



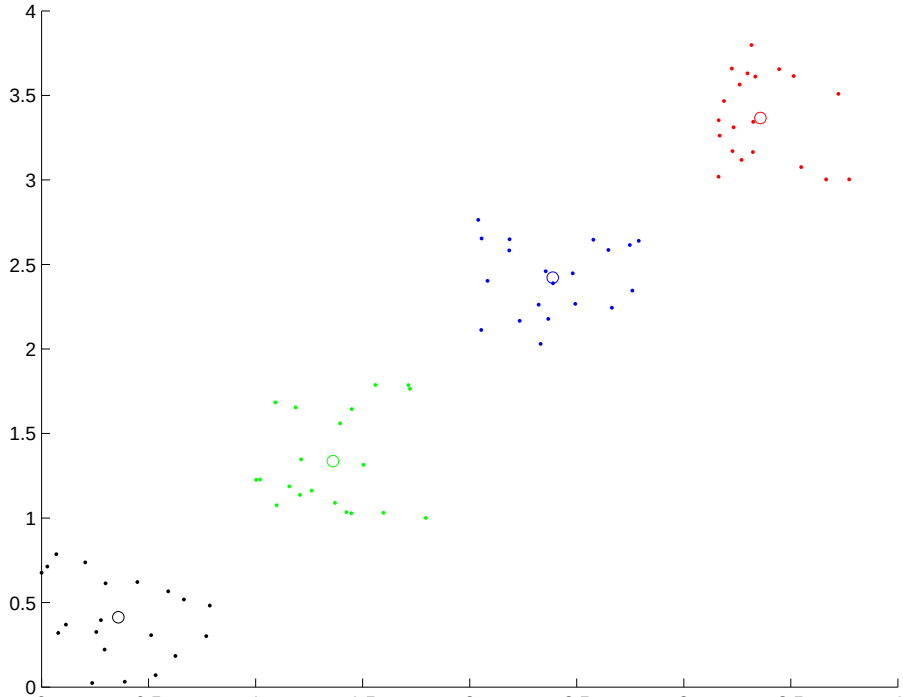
5em iteration



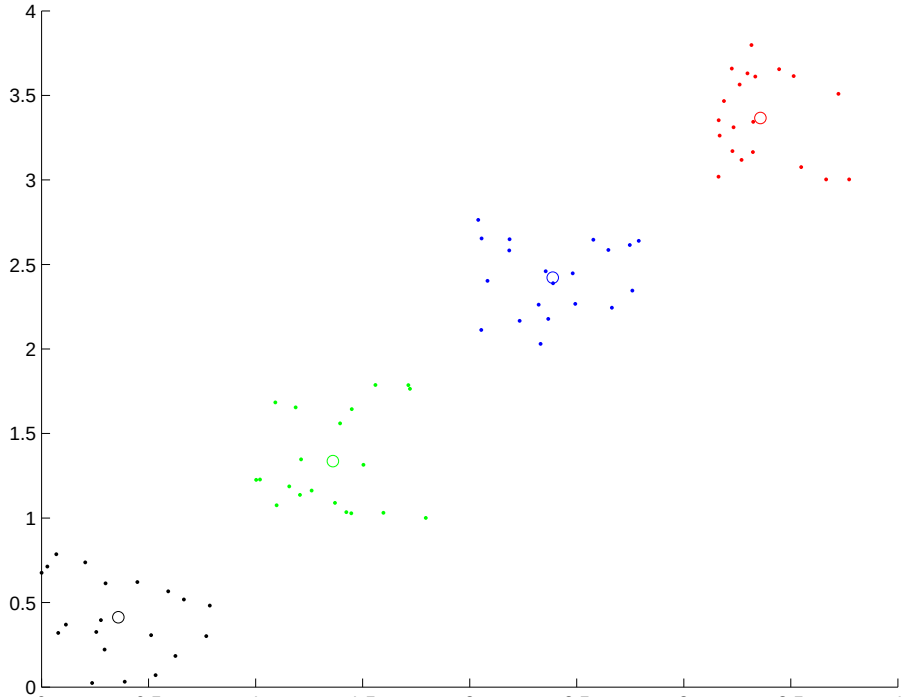
6em iteration



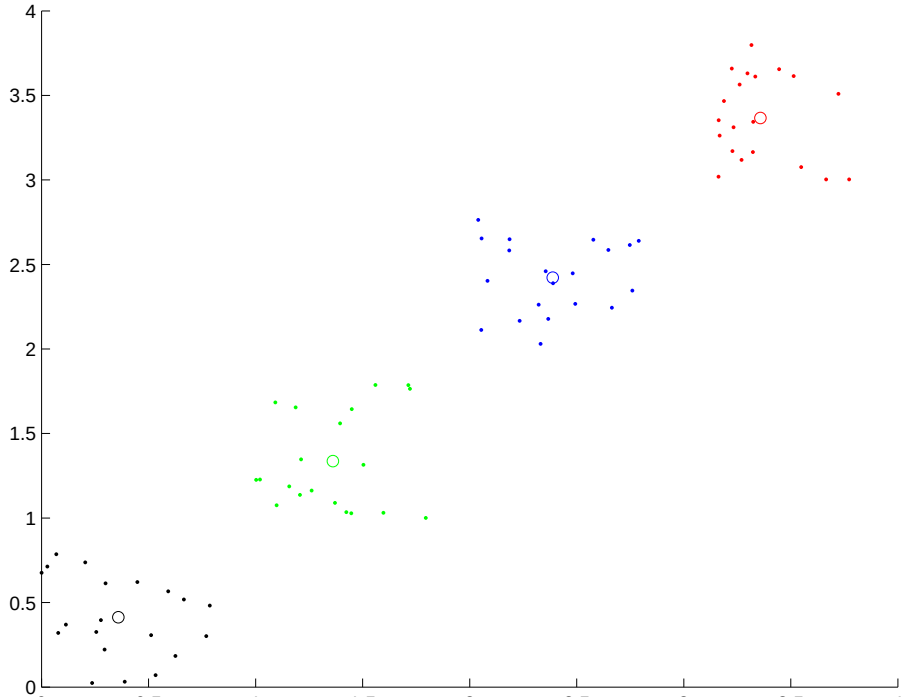
7em iteration



8em iteration



9em iteration



Algorithmes de partitionnement : Fuzzy-K-Means

- Variante de l'algorithme K-means :
  - chaque objet appartient, avec une certaine probabilité (distance), à tous les clusters
  - proche de l'algorithme EM pour les GMMs
- **Initialisation :**
  - choisir K subsets (clusters) initiaux,
  - calculer le centroide de chaque subset (cluster) à partir des objets attribués à ce subset
- **Etape E (Expectation) :**
  - chaque objet à une certaine probabilité d'appartenance à chaque cluster, on calcul l'appartenance de chaque objet x à chaque cluster k comme

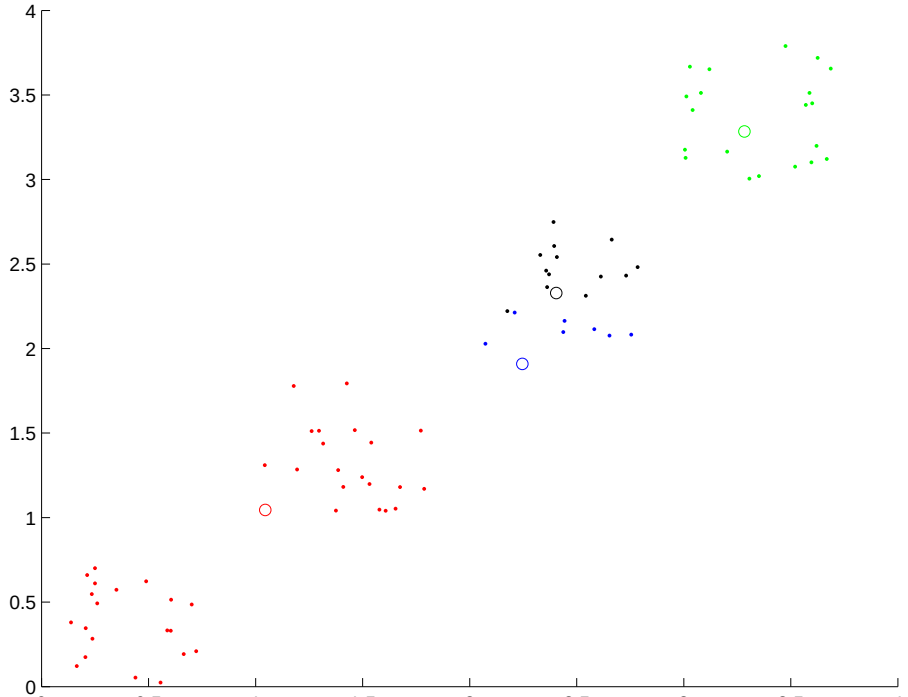
$$invd(k,x) = \left(\frac{1}{d(k,x)}\right)^{1/(b-1)}$$

$$P(k,x) = \frac{invd(k,x)}{\sum_k invd(k,x)}$$

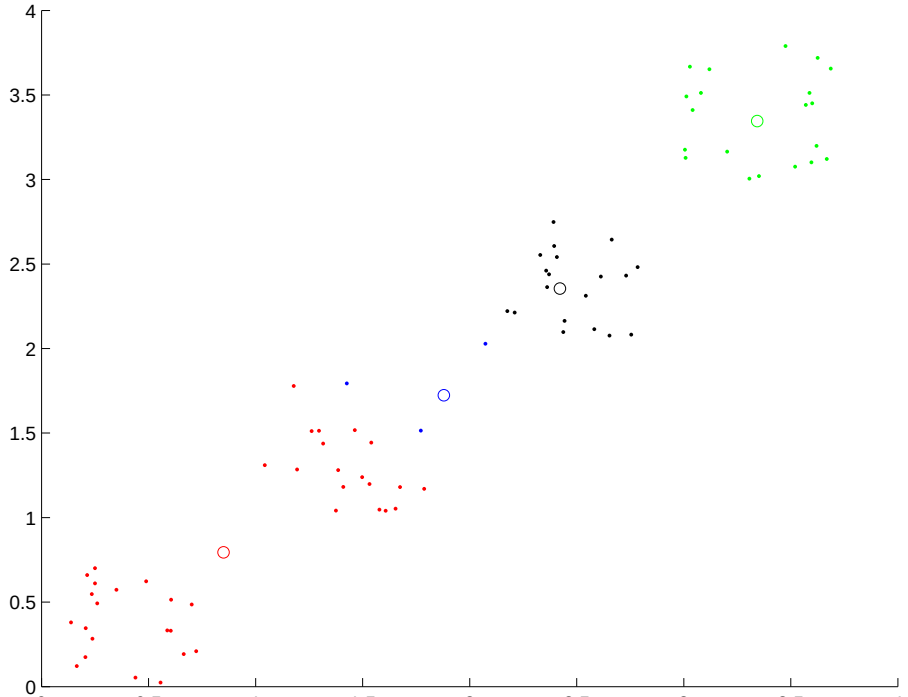
- **Etape M (Maximization) :**
  - le centroide de chaque subset (cluster) est calculé comme une moyenne pondérée de tous les objets, la pondération est fonction de l'appartenance des objets x à un cluster k donné

$$G_k = \frac{\sum_x P(k,x)^b f(x)}{\sum_x P(k,x)^b}$$

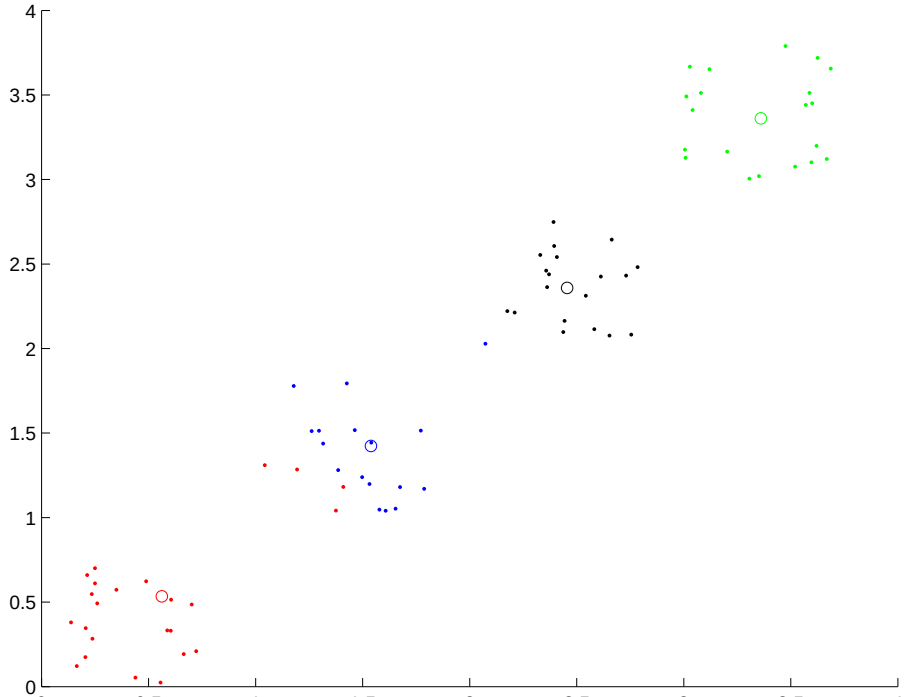
1er iteration



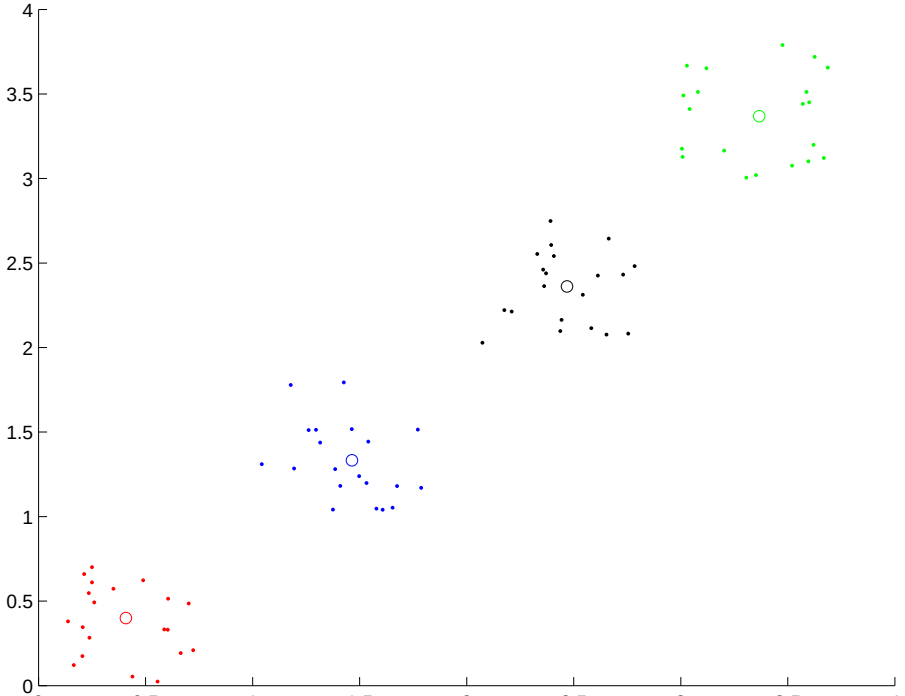
2em iteration



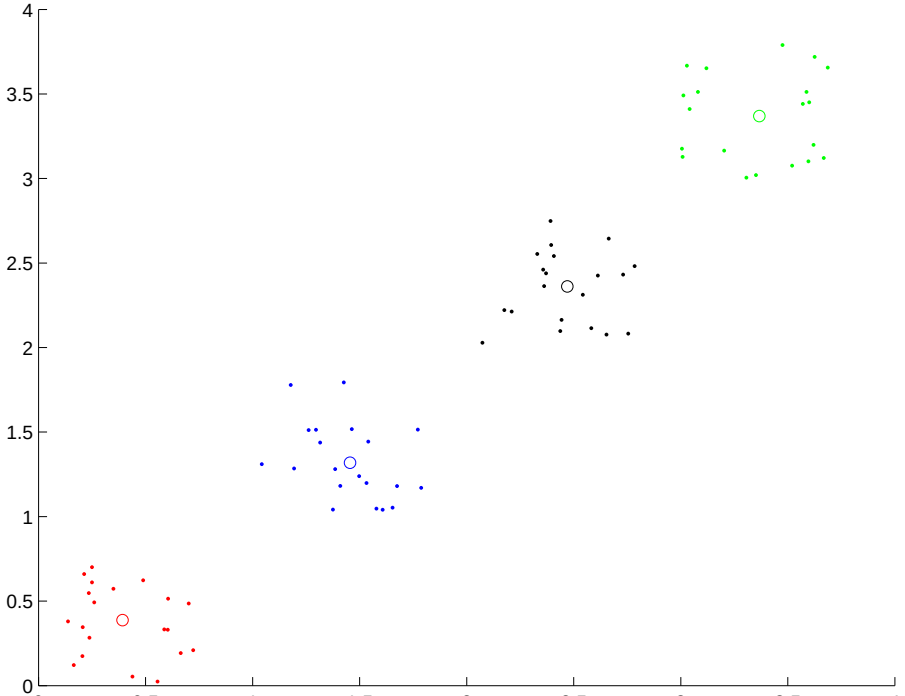
3em iteration



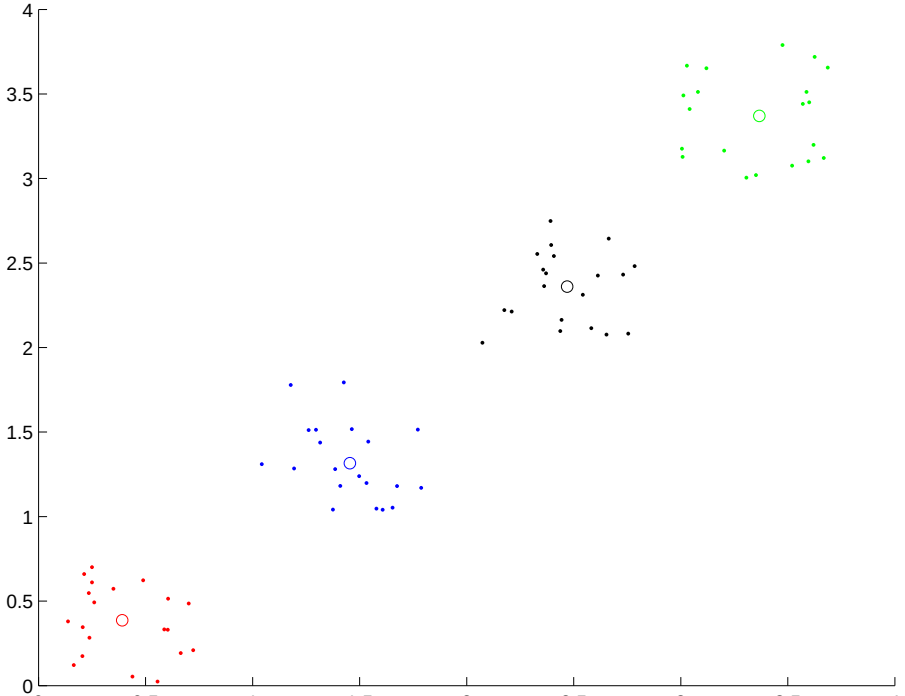
4em iteration



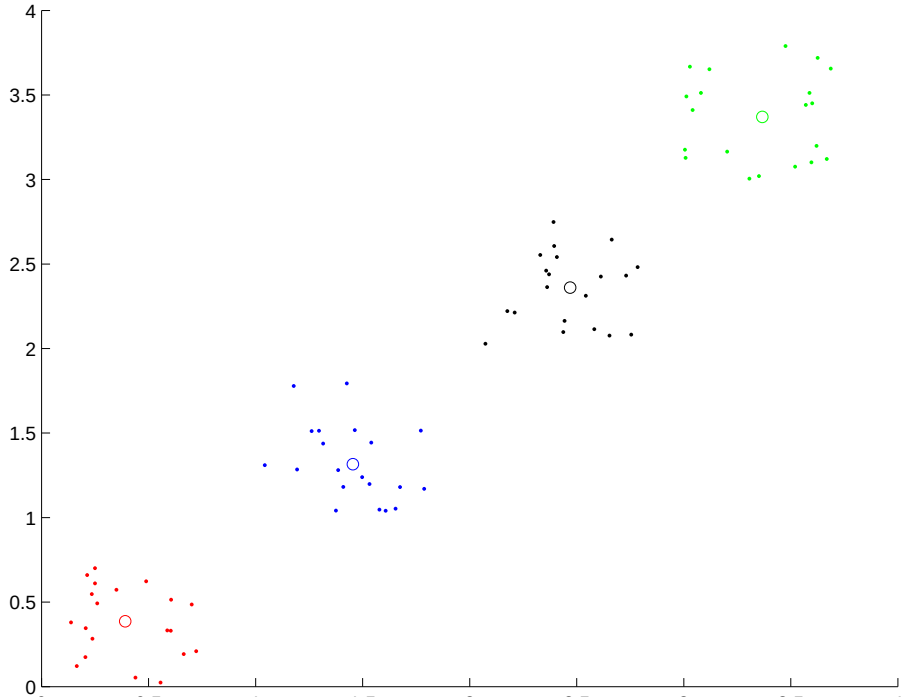
5em iteration



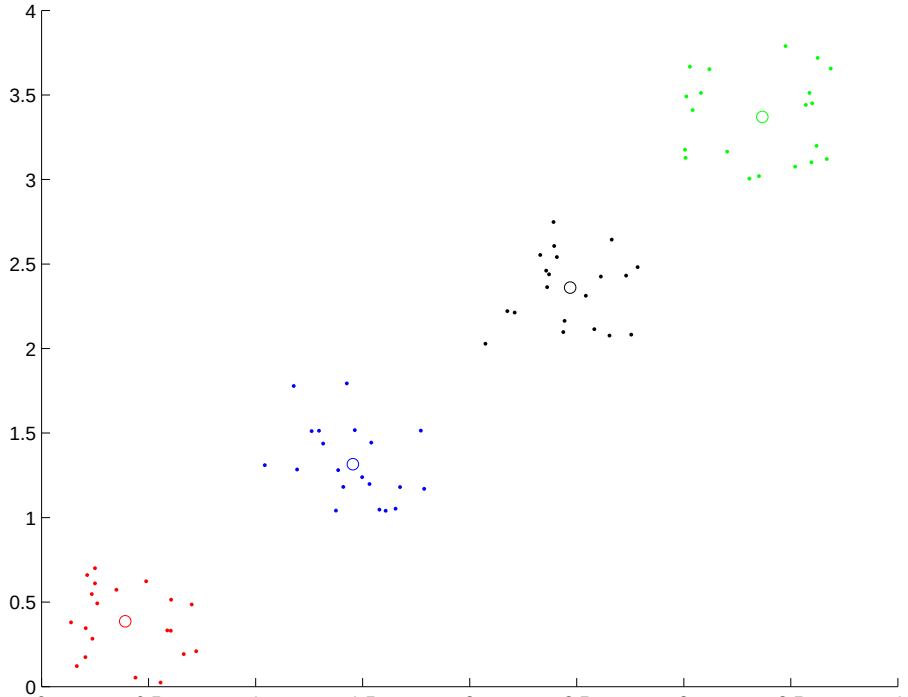
6em iteration



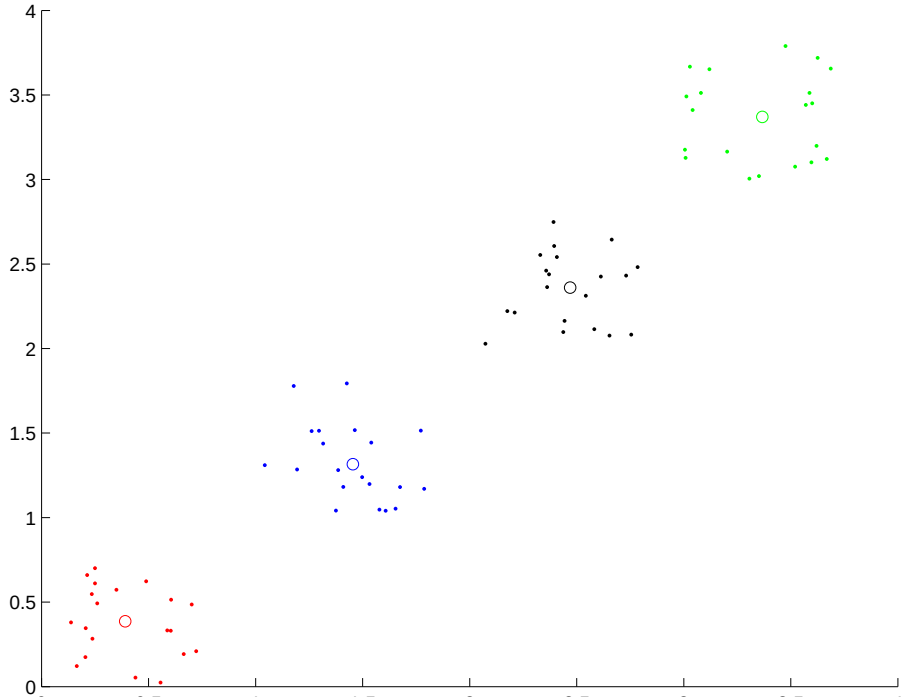
7em iteration



8em iteration



9em iteration



Algorithmes de partitionnement : Gaussian Mixture Model

- Modèle :
  - $p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$
    - avec $\sum_{k=1}^{K} \pi_k = 1$
- On peut ré-écrire ce modèle comme
  - $p(x) = \sum_{k=1}^{K} p(k)p(x|k)$
- Estimation
  - K mélanges
  - $n \in [1, N]$ données x_n

Expectation

- Probabilité a-posteriori :

$$\gamma_k(x) = p(k|x) = \frac{p(k)p(x|k)}{p(x)} = \frac{\pi_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

Maximization

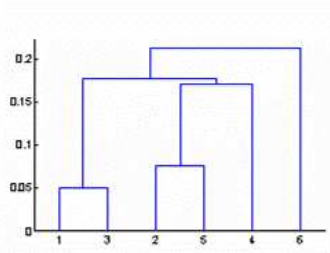
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

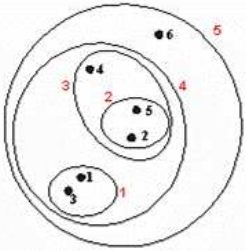
$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

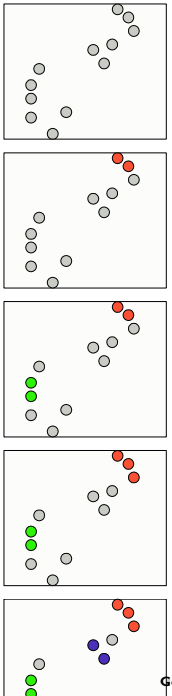
$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

Algorithmes hiérarchiques : par agglomération

- **Initialisation :** chaque objet constitue un cluster
- **Itération :** regroupement des objets ou clusters les plus proches
- **Condition d'arrêt :** on arrive au sommet de l'arbre, ou bien on a obtenu K clusters







- Choix de **distance entre objets** x et y
  - $dist$ = distance Euclidienne, Minkowski, cosine, ...
- Choix de **distance entre entre un objet et un cluster, entre deux clusters** R **et** S
  - **Single :** plus petite distance ...

$$d(R,S) = \min_{x,y}(dist(x,y)) \ \forall x \in R, \ \forall y \in S$$

  - **Complete :** plus grande ...

$$d(R,S) = \max_{x,y}(dist(x,y)) \ \forall x \in R, \ \forall y \in S$$

  - **Average :** moyenne des distances entre les paires ...

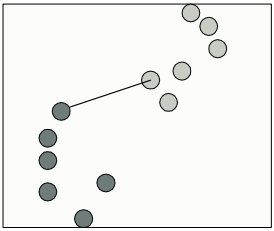
$$d(R,S) = \text{mean}_{x,y}(dist(x,y)) \ \forall x \in R, \ \forall y \in S$$

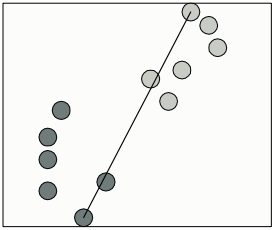
  - **Centroid :** distance entre les centroides de R st S

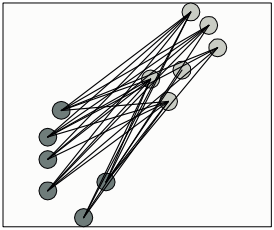
$$d(R,S) = dist(\overline{R},\overline{S})$$

  - **Ward :** représente l'augmentation de l'inertie intra-groupe due à la réunion des groupes R et S

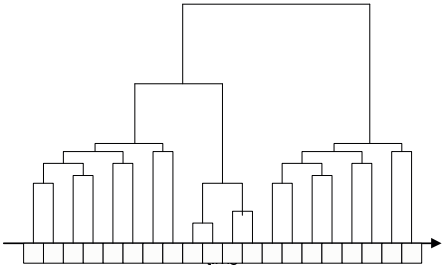
$$d(r,s) = \frac{n_r n_s}{n_r + n_s} dist(\overline{R},\overline{S})$$







- Représentations et coefficients
  - **Dendrogramme :**
    - Représente graphiquement sous forme d'arbre binaire les différentes connections entre objets/clusters.
    - La hauteur de la connexion (distance Cophenetic) entre deux objets/clusters représente la distance entre les deux objets/ clusters connectés.



- **Coefficient de corrélation Cophenetic :**
  - Compare la distance entre les objets entre eux à la distance entre les objets tels que connectés dans l'arbre ; indique l'adéquation de l'arbre à représenter les données
- **Coefficient d'inconsistance :**
  - Compare la hauteur d'une connexion à la hauteur moyenne des connections (précédantes) ayant amené ces objets.
  - Connexion **consistante :** a approximativement la même hauteur que les connections inférieures, indique qu'il n'y a pas de division distinctes entre les objets réunis à ce niveau de la hiérarchie et les objets qu'ils contiennent
  - Connexion **inconsistante :** a une hauteur différentes des connections inférieures, indiquent que les objets réunis à ce niveau sont beaucoup plus éloignés que les objets qu'ils contiennent.

Algorithmes hiérarchiques : par divisions

- Initialisation : un seul cluster contenant tous les objets
- Itération : séparer les objets ou clusters les plus dis-similaires
- Condition d'arrêt : tous les objets sont séparés ou K clusters

Autres : Clustering spectral : Singular Value Decomposition

[M. Cooper and J. Foote. Summarizing popular music via structural similarity analysis. In Proc. of IEEE WASPAA, New Paltz, NY, USA, 2003.]

- Pour chaque segment j, calcul de μ_j et Σ_j pour chaque segment j
- Construction d'une matrice de similarité $\underline{\underline{S}}$ entre segments
  - utilisation de la divergence symétrisée de Kullback-Leibler (cas Gaussien) :

$$d_{KL}(G(\mu_i,\Sigma_i),G(\mu_j,\Sigma_j)) = \frac{1}{2}\left[Tr(\Sigma_i\Sigma_j^{-1}) + Tr(\Sigma_j\Sigma_i^{-1}) + (\mu_i-\mu_j)^t(\Sigma_i^{-1}+\Sigma_j^{-1})(\mu_i-\mu_j)\right] - B$$

  - B est le nombre de dimensions
- Décomposition de la matrice de similarité $\underline{\underline{S}}$ par SVD :

$$S = U\Lambda V^T$$
$$S(i,j) = \sum_p \lambda_p U(i,p)V(j,p) = \sum_p B_p(i,j)$$

  - pour chaque valeur singulière λ_p, on a la décomposition du morceau sur une sous-matrice $B_p(i,j)$
- Pour une sous-matrice donnée $B_p(i,j)$, la somme d'une colonne $b_p(j) = \sum_i B_p(i,j)$
  - fournit la similarité entre un segment j et tous les autres segments i selon ce cluster p.
  - le segment j est assigné au cluster p (à la sous-matrice $B_p(i,j)$) pour lequel il est le plus similaire aux autres segments.

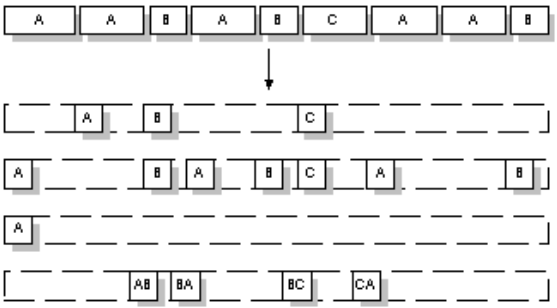
Autres : Clustering par NMF

- Problèmes des algorithmes de clustering :
  - Choix du nombre de clusters
  - Choix d'un critère de distance/ similarité pour le regroupement
    - distance euclidienne, distance de Minkowski, de Manhattan, ... ?
  - Choix d'un critère de qualité de chaque cluster
  - Problème d'homogénéité des unités des dimensions
    - normalisation des dimensions par leur variance ?

Génération de résumé audio par estimation de structure

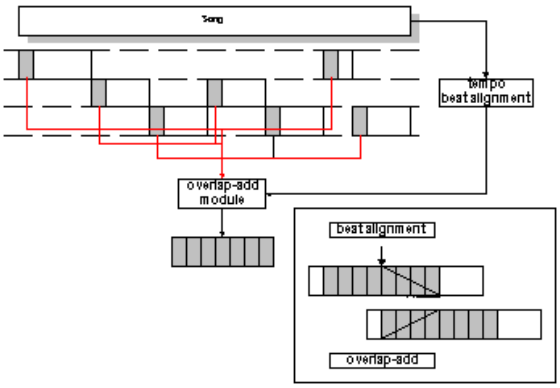
[G. Peeters, A. Laburthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In Proc. of ISMIR, Paris, France, 2002.]Peeters, Laburthe and Rodet 2002, ISMIR]

- Stratégie proposée
  - choisir des extraits audio spécifiques en fonction du contenu dérivé de l'approche par séquence/ par état
- Construction du résumé
  - Le signal est représenté comme une succession de séquences/ états A A B A B C A A B
  - Quels séquences/ états pour le résumé ?
    - un exemple unique de chaque séquence/ état
    - reproduire la succession temporelle des séquences/ états
    - la séquence/ état le plus important (en terme de nombre de répétition, en terme d'extension temporelle)
    - exemple audio des transitions entre états



Génération de résumé audio par estimation de structure

- Construction du signal audio :
  - Extraits courts de signal audio correspondant aux séquences/ états choisis
  - Doit fournir une construction "cohérente" et "intelligente"
  - Continuité de l'information : Addition/ Recouvrement (Overlap-add), respect du tempo/ beat, taille des segments = kx4 or kx3 bars, synchronisation aux positions des beats

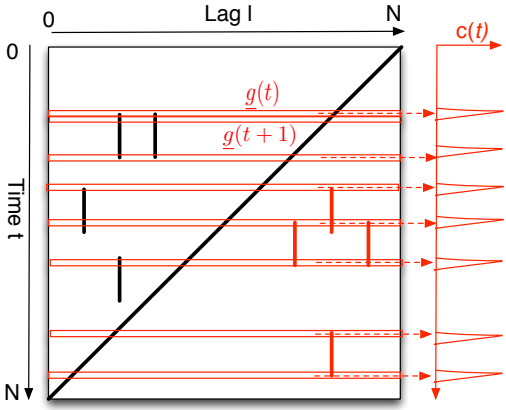


7- Estimation d'une structure musicale - approche par "séquence"

Segmentation : méthode des "Structural features"

[J. Serra, M. Muller, P. Grosche, and J. L. Arcos. Unsupervised detection of music boundaries by time series structure features. In Proc. of AAAI Conference on Artificial Intelligence, 2012.]

- Calcul de la matrice d'auto-similarité en (temps, lag)Lag-matrix
- On considère chaque ligne (les lags pour un temps donné) comme une "structural feature" $\underline{g}^t$
- On calcul la différence trame à trame de $\underline{g}^t$: $||\underline{g}^{t+1} - \underline{g}^t||^2$

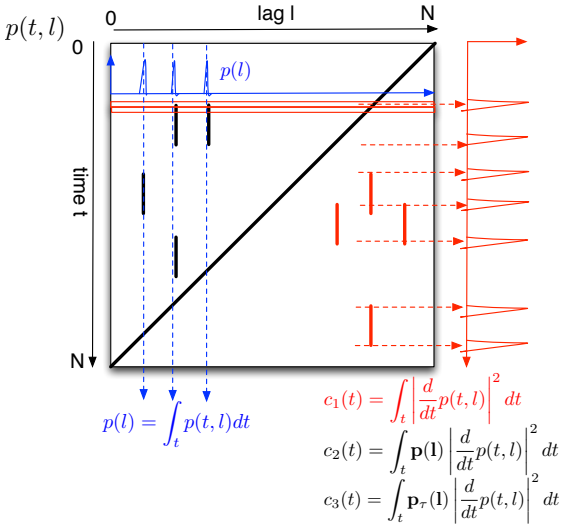


Serra: $c(t) = ||\underline{g}(t+1) - \underline{g}(t)||^2$

Segmentation : méthode des "Structural features" avec probabilité a-priori

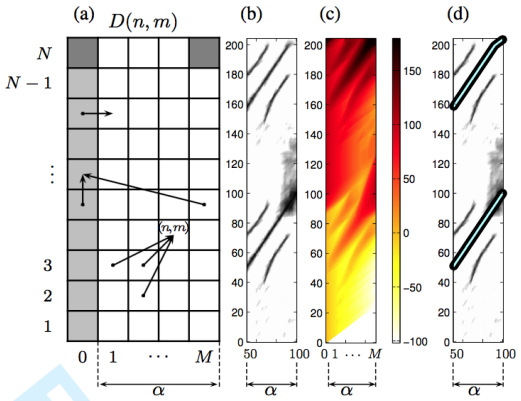
[G. Peeters and V. Bisot. Improving music structure segmentation using lag-priors. In Proc. of ISMIR, Taipei, Taiwan, 2014.]

- Pondération des "structural feature" par la probabilité a prior d'observer une répétition à un lag donné
- Calcul de cette probabilité par la méthode de Goto 2003
- On calcul la différence trame à trame

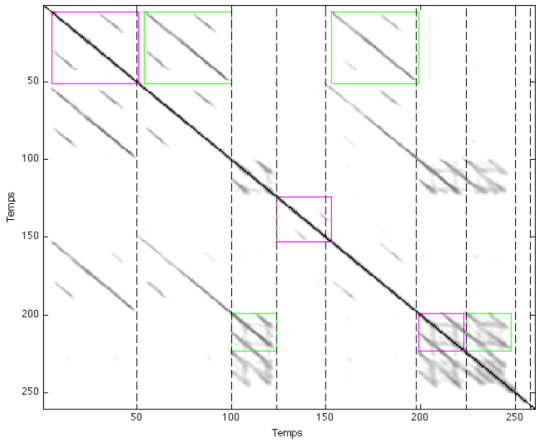


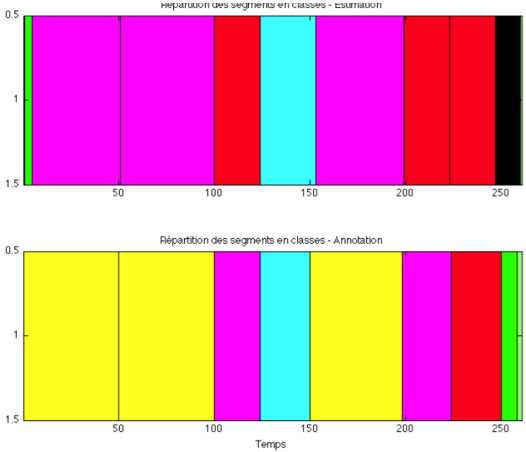
Regroupement par Dynamic Time Warping

- Pour chaque segment détecté, on cherche quelles séquences temporelles sont expliquées
- Utilisation d'une version modifiée du Dynamic Time Warping



source : Mueller





source : Bisot

Questions ?