

# MUS4831 - Lydanalyse

31 oktober 2013

# Denne forelesningen:

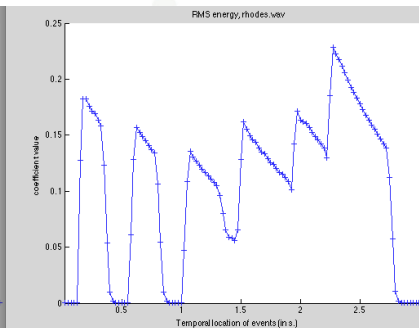
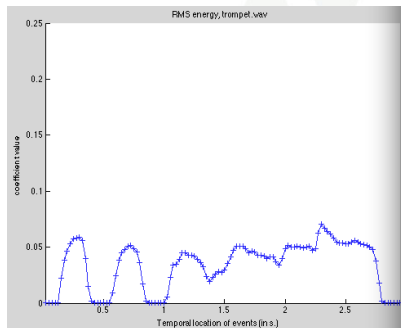
- Gjennomgang av Oppgave 3
- Tips til semesteroppgaven
- Music Information Retrieval del 2: Sound, mønstre, likhet, osv. Teoretisk og praktisk

# Obligatorisk innleveringsoppgave 3

# Oppgave 3

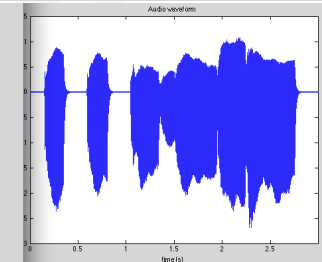
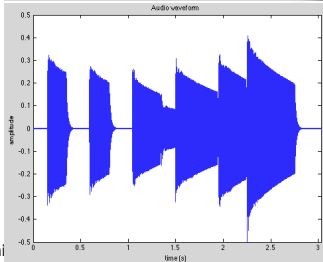
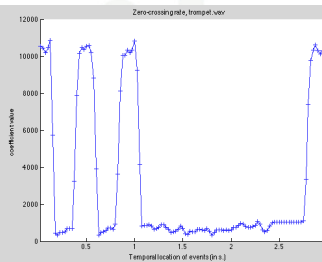
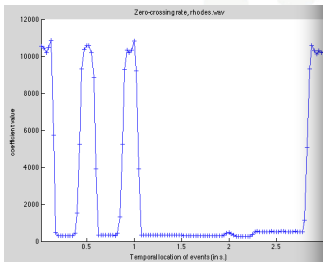
Lydfiler: **rhodes.wav** og **trompet.wav**

- Lag plot av lydenergien i hver av de to filene, for eksempel ved å bruke root-mean-square funksjonen i MIR toolbox. Hvordan ser plottene ut i forhold til hva du hører?



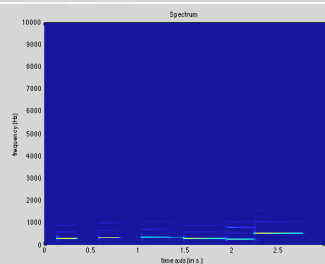
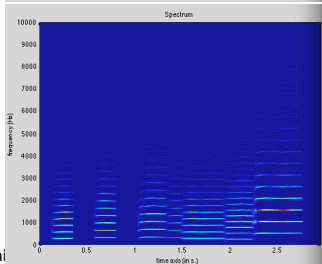
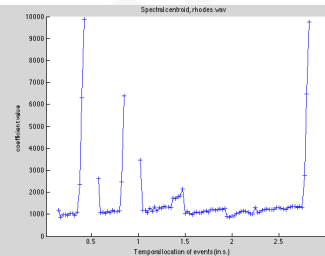
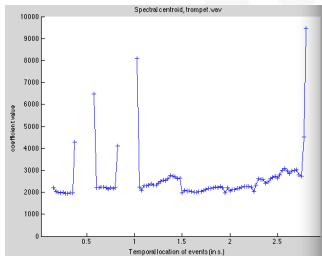
# Oppgave 3

- Lag plot av zerocross rate. Denne blir til tider svært høy, hva tror du dette kommer av?



# Oppgave 3

- Lag plot av spektralsentroiden til de to lydfilene og sammenlikn av dem. Ser du en likhet med zerocross-rate her?



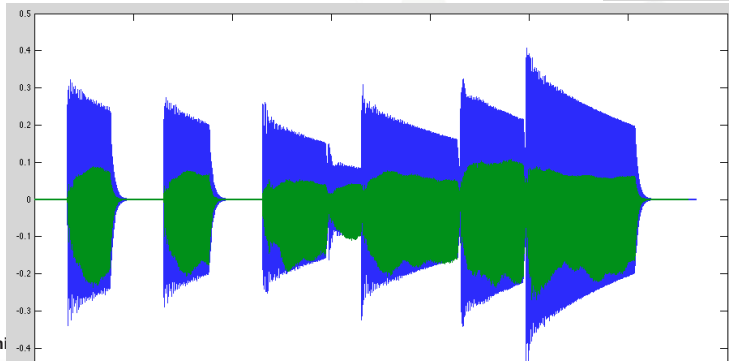
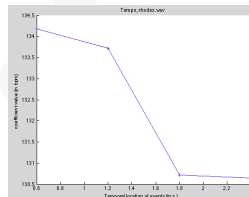
# Oppgave 3

- Hvilket tempo (bpm) har de to filene?

The Tempo related to file rhodes.wav is 132.4165 bpm

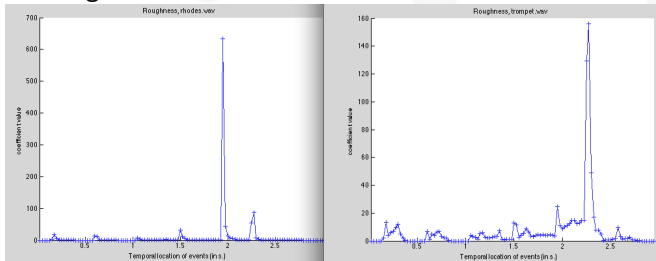
```
>> mirtempo(b)
```

The Tempo related to file trumpet.wav is 131.5569 bpm

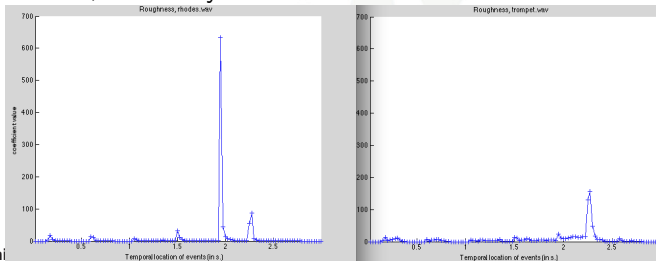


# Oppgave 3: roughness

mirroughness, med standardverdier:



samme, med lik y-akse:





## Oppgave 3: roughness

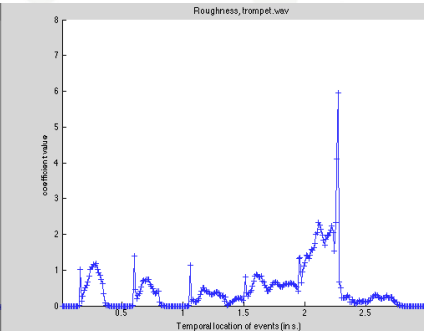
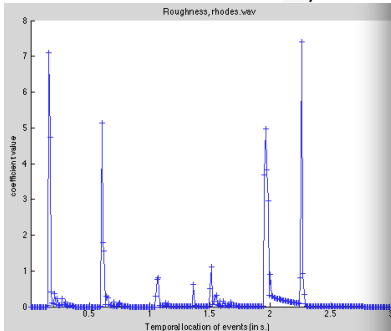
Fra hjelpefilen til mirroughness:

$r = \text{mirroughness}(x)$  calculates the roughness, or sensory dissonance, due to beating phenomenon between close frequency peaks. The frequency components are supposed to remain sufficiently constant throughout each frame of each audio file

- Hvis du skulle svare ut ifra hva du hører, hvilket av de to instrumentene har størst "roughness" / røffhet / ugjevnhet?

# Oppgave 3: roughness

20 millisekunder vindustørrelse:



```
>> mirroughness(b, 'frame', 0.02)
Computing mirspectrum related to trompet.wav...
Computing mirpeaks related to trompet.wav...
Computing mirroughness related to trompet.wav...

The Roughness related to file trompet.wav is displayed in Figure 3.

>> mirroughness(a, 'frame', 0.02)
Computing mirspectrum related to rhodes.wav...
Computing mirpeaks related to rhodes.wav...
Computing mirroughness related to rhodes.wav...

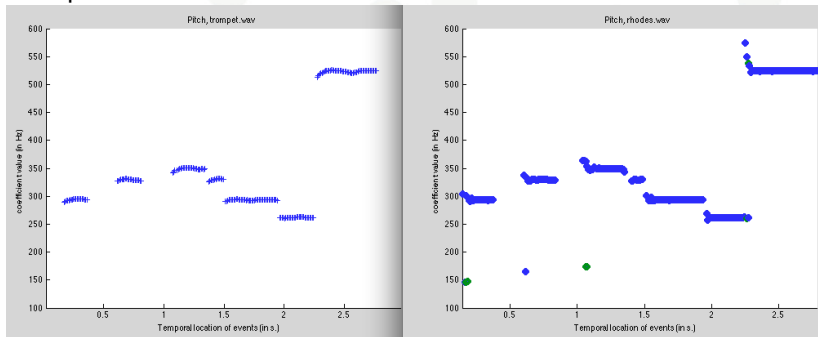
The Roughness related to file rhodes.wav is displayed in Figure 4.

>> ylim([0 8])
```

```
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
mirrough
```

# Oppgave 3

## Plot pitchen i de to filene



- Hva ser vi her?

# Oppgave 3

Oppsummering: Litt lærdom fra oblig 3.

- Tolkning.... ikke stol 100% på automatiske analyser!  
eksempler fra oblig 3:
  - 2 pitcher
  - tempoforskjeller
  - pitchvariasjoner. Er det faktiske variasjoner eller svakheter i analysemetoden?
- Roughness: Viser viktigheten av vindusstørrelse og tolkning av resultat
- Aksebenevning og skalering.

# Semesteroppgaven

## 1. Dokumentasjon.

- Det du har gjort bør være så godt beskrevet at noen andre kan gjenta eksperimentet.
- Utvetydig
- Gjerne med bilder / video dersom dere gjør egne opptak.
- Hva slags utstyr er brukt?
- Hva slags programvare er brukt?
- Har du brukt tillegg til standarddistribusjonen av programmene? (f.eks. MIR toolbox til matlab)
- Hvilke innstillinger bruker du? (f.eks. vinduslengde)
- Hvorfor bruker du disse programmene/tilleggene/innstillingene?
- Forklar forkortelser første gang de brukes (RMS,ASDR,...)

# Litt veiledning for semesteroppgaver i MUS4831

## 2. Referanser.

Tenk som i masteroppgaven. Vær nøye med referanser. La det være helt tydelig hva som er ditt eget arbeid, og hvilken informasjon du bygger på fra andre steder. Sammenlikn helst ditt eget arbeid (både resultater og metode) med det andre har gjort. Legg ved referanser til programvare.

I semesteroppgaven kreves det ikke referanser til programvare som er en del av pensum, men det oppfordres likevel til å inkludere det. (Og uansett om dere har med referanser eller ikke til programvaren må det selvsagt nevnes hvilken programvare som er brukt).

- Boersma, Paul and Weenink, David (2013). *Praat: doing phonetics by computer* [Computer program]. Version 5.3.56, retrieved 17 September 2013 from <http://www.praat.org/>
- Klingbeil, Michael (2005). "Software for Spectral Analysis, Editing, and Synthesis" in *Proceedings of the International Computer Music Conference*, pp. 1–4.

## 3. Skrivefeil

En og annen sporadisk skrivefeil trekker ikke ned, men mange gjennomgående skrivefeil kan trekke en oppgave ned med en hel karakter. Bruk stavekontroll, eller eventuelt få noen andre til å lese over oppgaven din og se etter skrivefeil.

## 4. Tolkning av plot

Følg med på verdien på aksene på plottene når du sammenlikner to eller flere plot. Ikke stol blindt på automatiske analyser. (MIR toolbox gjør "feil". For eksempel vil vindusstørrelsen som brukes i de underliggende funksjonene påvirke resultatet fra tempoanalyser)



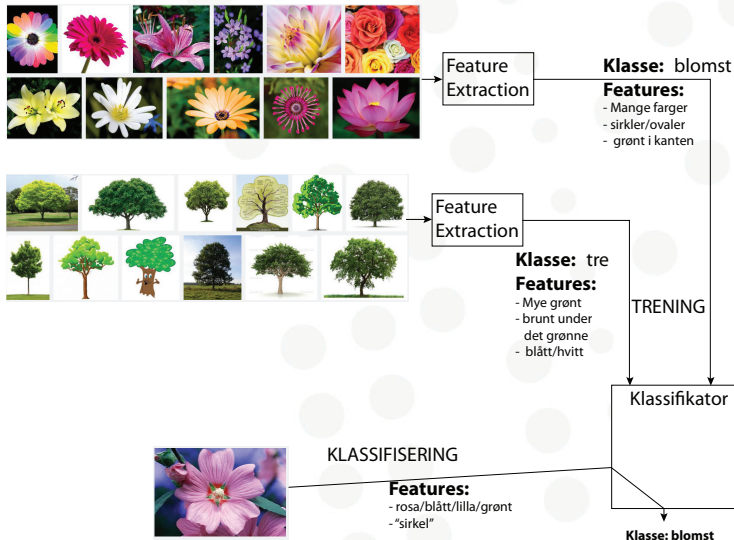
# Music Information Retrieval - del 2

- Maskinlæring / Mønstergjenkjenning
- MIR i lengre musikksekvenser: Form, likhet, harmoni...

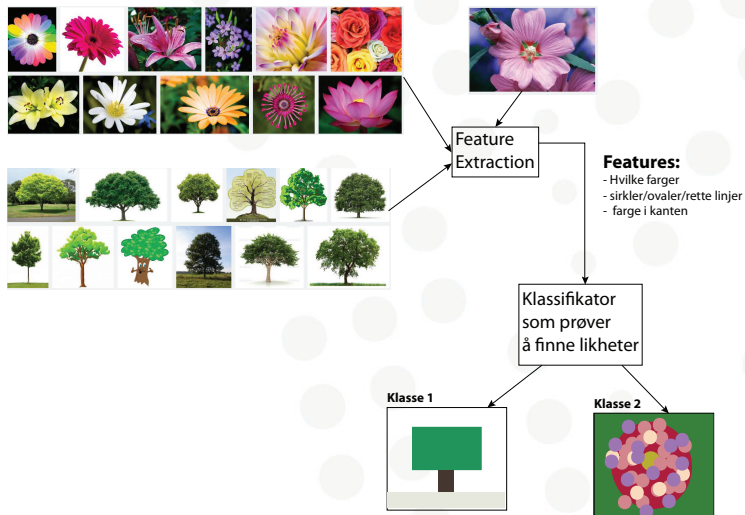
Noen hovedkategorier:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning (ikke så relevant for MIR)

# Supervised learning



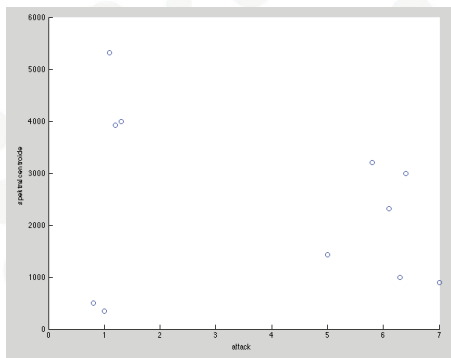
# Unsupervised learning



# Mønstergjenkjenning / "Pattern Recognition" og MIR

Enkelt eksempel med 4 lydklasser og to features:

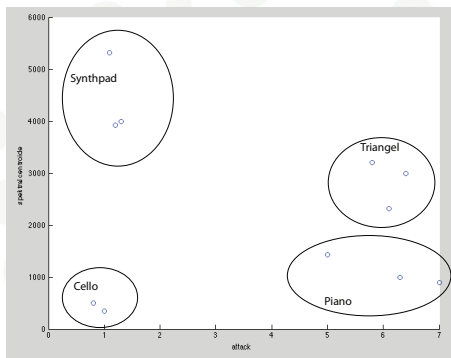
KLASSE	ATTACK	SPECTRAL CENTROID
Triangel	6.4000	3000
Piano	6.3000	1000
Triangel	5.8000	3200
SynthPad	1.3000	4000
Cello	0.8000	500
Piano	5	1423
Cello	1	345
SynthPad	1.1000	5310
SynthPad	1.2000	3923
Triangel	6.1000	2313
Piano	7	900



# Mønstergjenkjenning / "Pattern Recognition" og MIR

Enkelt eksempel med 4 lydklasser og to features:

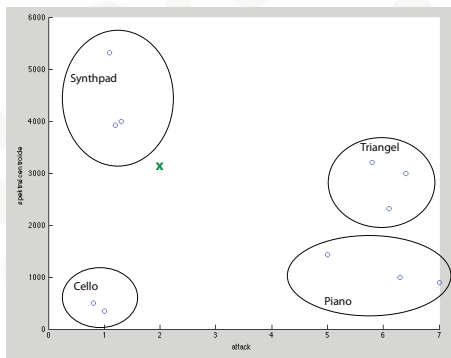
KLASSE	ATTACK	SPECTRAL CENTROID
Triangel	6.4000	3000
Piano	6.3000	1000
Triangel	5.8000	3200
SynthPad	1.3000	4000
Cello	0.8000	500
Piano	5	1423
Cello	1	345
SynthPad	1.1000	5310
SynthPad	1.2000	3923
Triangel	6.1000	2313
Piano	7	900



# Mønstergjenkjenning / "Pattern Recognition" og MIR

Enkelt eksempel med 4 lydklasser og to features:

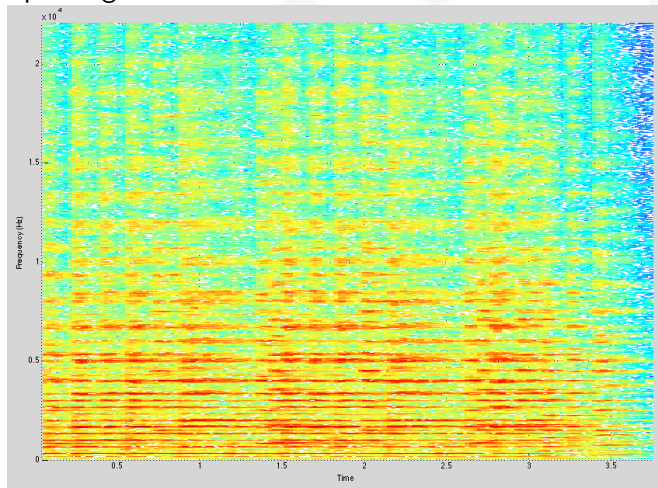
KLASSE	ATTACK	SPECTRAL CENTROID
Triangel	6.4000	3000
Piano	6.3000	1000
Triangel	5.8000	3200
SynthPad	1.3000	4000
Cello	0.8000	500
Piano	5	1423
Cello	1	345
SynthPad	1.1000	5310
SynthPad	1.2000	3923
Triangel	6.1000	2313
Piano	7	900





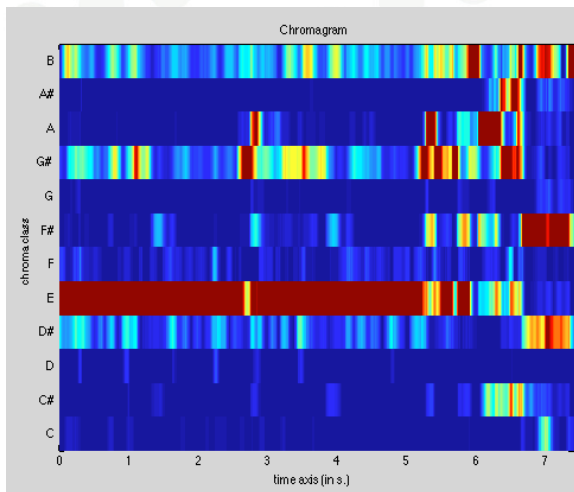
# Kjenne igjen ulike deler i et musikkstykke

Spektrogram:



Kanskje vi kan kjenne igjen noe her, men det er vanskelig...

# Kjenne igjen ulike deler i et musikkstykke

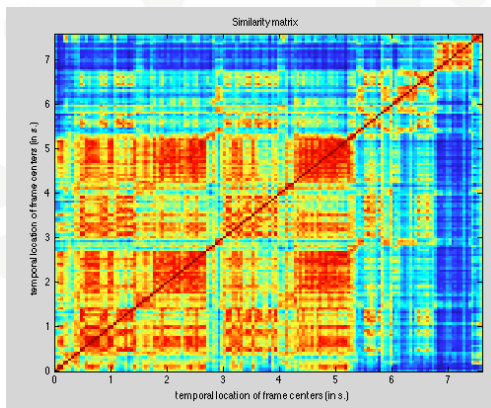


Chromagram:

Hvilke kromatiske toner er mest representert i hver frame?

- her ser vi en stor forandring etter ca 5 sek

# Kjenne igjen ulike deler i et musikkstykke



Self-similarity matrix:

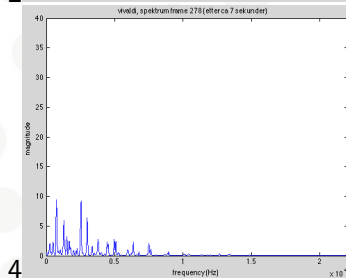
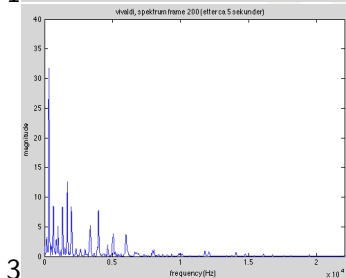
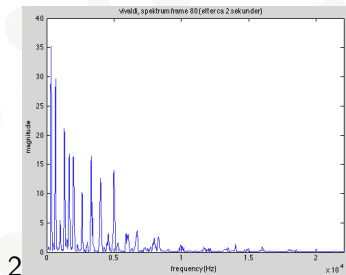
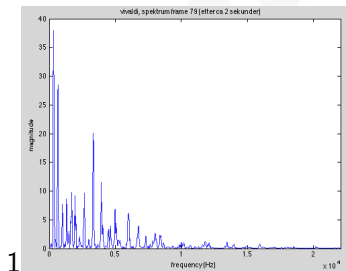
Hvor likt er spektrumet i hver frame på X-aksen sammenliknet med hver frame på Y-aksen? (rød = stor grad av likhet, blå = ulik)

# Self-similarity matrix

Slik virker mirsimatrix (konseptuelt sett)

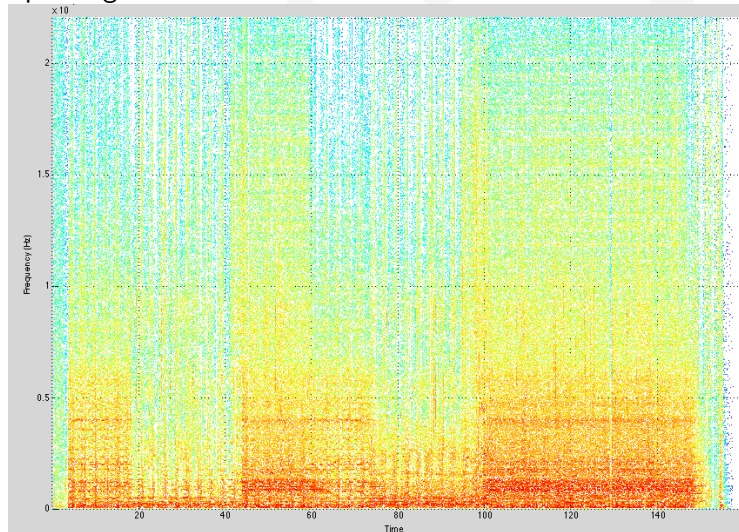
- Finn spektrumet i hver frame i lydfilen
- Ta først utgangspunkt i frame 1
  - Sammenlikn spektrumet i frame 1 med spektrumet i frame 1
  - Sammenlikn spektrumet i frame 1 med spektrumet i frame 2
  - Sammenlikn spektrumet i frame 1 med spektrumet i frame 3
  - osv
- Ta så utgangspunkt i frame 2, frame 3, 4, 5...
- I eksempelet på forrige side er det en tydelig sort diagonal. Den illustrerer at hver frame er lik seg selv.
- Vi ser på forrige side at lydfilen etter 2 sekunder er ganske lik seg selv etter 5 sekunder, men ulik seg selv etter 7 sekunder. Se plot av spektrumene for disse øyeblikkene på neste side.

# Self-similarity matrix



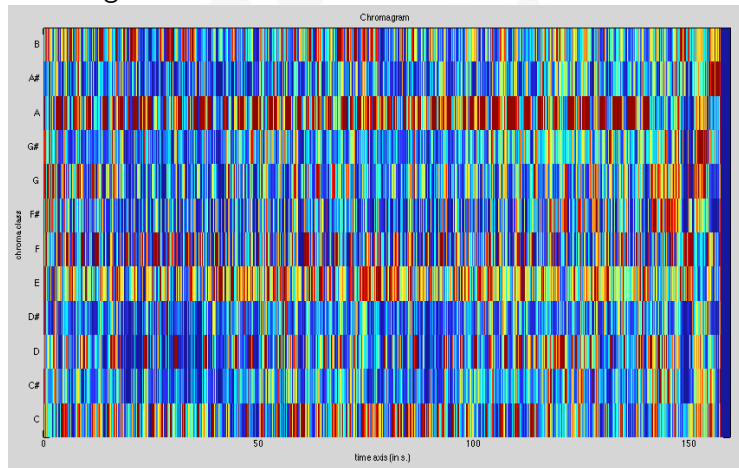
# En litt lengre låt:

Spektrogram:



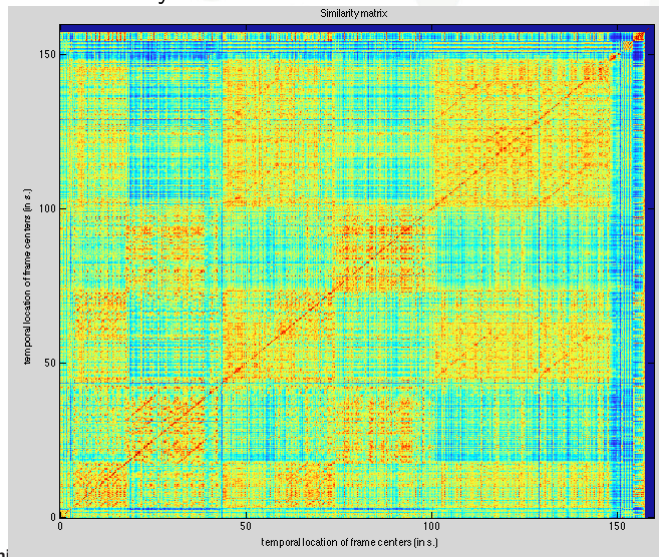
# En litt lengre låt:

Chromagram:



# En litt lengre låt:

Self-similarity matrix:





# Noen MIR-toolbox funksjoner

**mirchromagram** Viser et chromagram

**mirflux** Hvor mye noe forandrer seg

- `mirflux(mirchromagram(lyd,'frame'))`

- `mirflux(mirspectrum(lyd,'frame'))`

- (sistnevnte er det samme som `mirflux(lyd)`)

**mirkey** Toneart

**mirfeatures** Som før: alt mulig rart....

- Når vi har beregnet features for en låt kan vi trene opp en mønstergjenkjenningsalgoritme til å kjenne igjen f.eks. ulike sjangre.
- MEN: som regel kreves en maskinlæringsmetode som kan ta inn tidsvarierende features. f.eks:
  - Hidden markov models (HMM)
  - Hierarchical temporal memory (HTM)
  - Dynamic Time Warping

## Programmer for maskinl ring:

- FTM (for Max)  
<http://ftm.ircam.fr>
- RapidMiner  
<http://rapid-i.com/content/view/181/>

## Mer om Music Information Retrieval:

- Nicola Orio: Music Retrieval: A Tutorial and Review
- Meinhard M ller: Information Retrieval for Music and Motion  
(begge er tilgjengelige som e-bok ved UiO)