

Reconnaissance musicale



I/Principe de la reconnaissance musicale

II/Algorithme d'empreinte

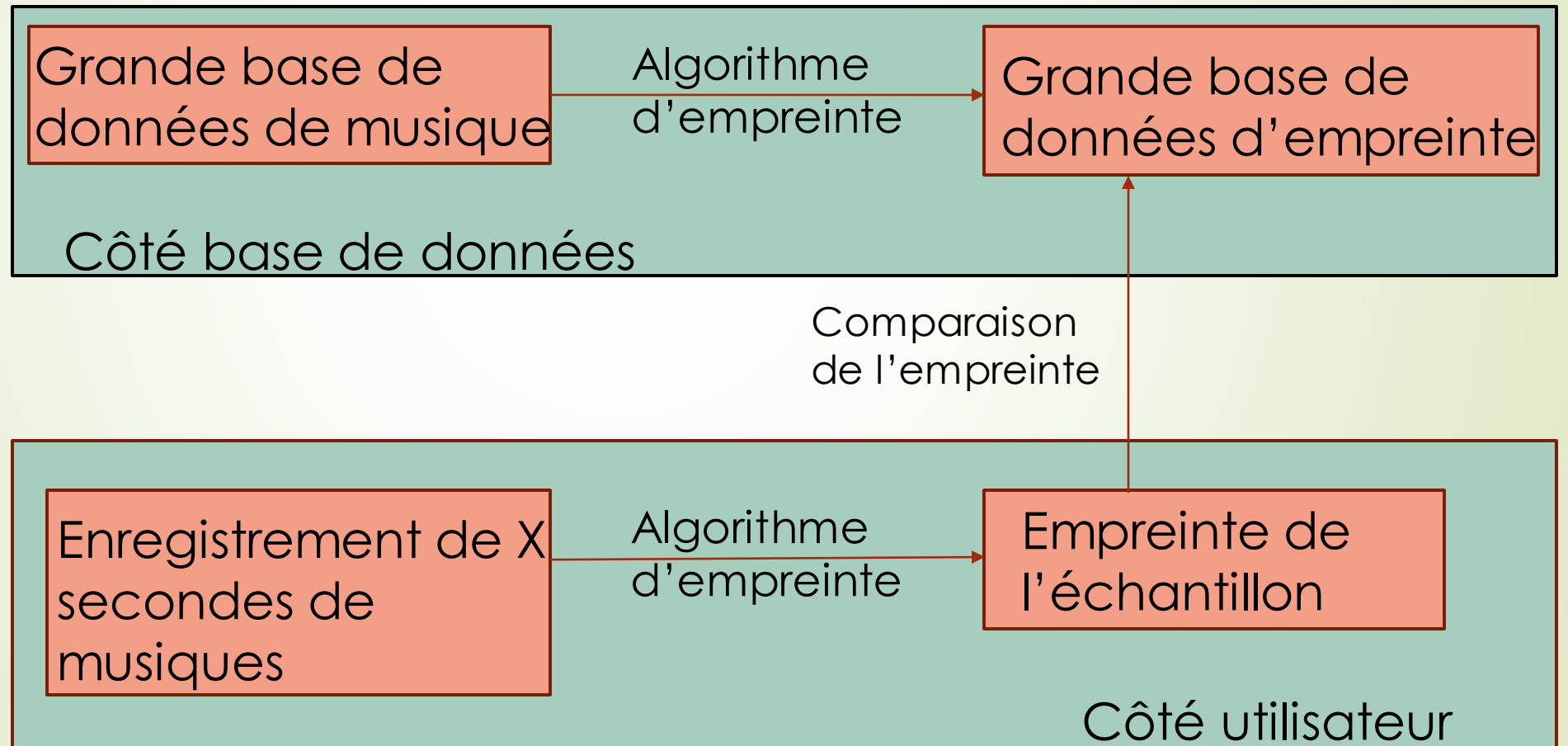
- Comment trouver des maxima locaux d'amplitudes?
- Former une empreinte et l'encoder

III/Base de données et résultats expérimentaux

- Reconnaissance d'une musique
- Fonctionnement avec la Base de données
- Résultats et efficacité

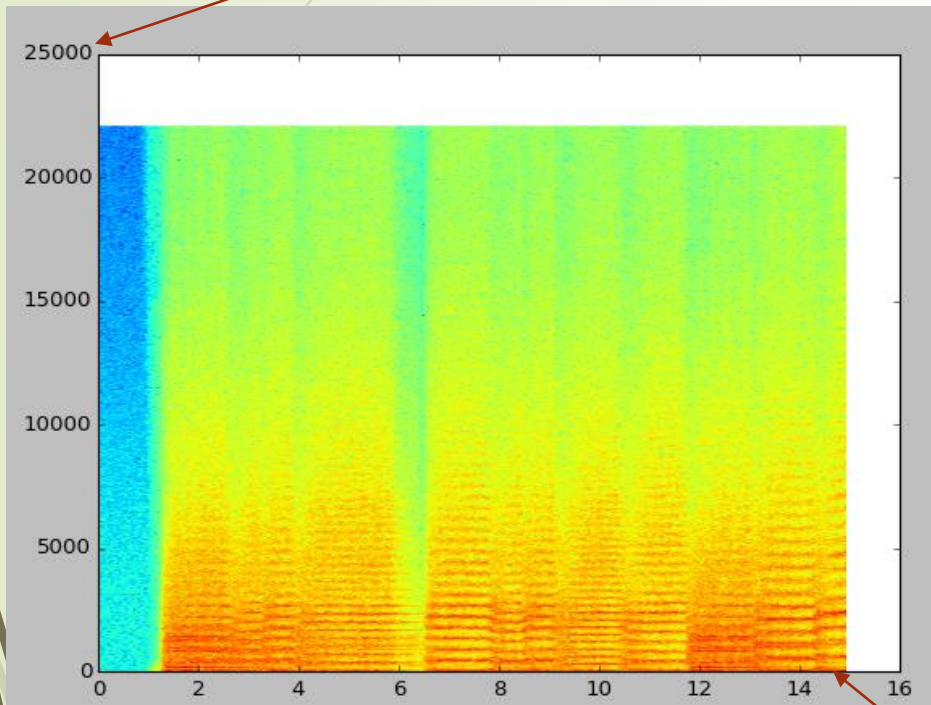


I/ Principe de la reconnaissance musicale



Ce qu'on utilise pour l'algorithme

Fréquence



Spectrogramme

Temps

Temps

Fréquence

Amplitude	
Amplitude	

Tableau FFT

II/Algorithme d'empreinte, trouver de maxima d'amplitude: Un exemple

Fréquence	Temps				
	4	35	25	10	6
	5	32	18	10	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

```
[[False, True, False],  
 [ True, True, True],  
 [False, True, False]]
```

Structure de
carré

1ère boucle

	Temps				
Fréquence	4	35	25	10	6
	5	32	18	10	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

	Temps				
Fréquence	4	4	25	10	6
	4	32	18	9	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

2ème boucle

	Temps				
Fréquence	4	35	25	10	6
	5	32	18	10	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

	Temps				
Fréquence	35	35	35	10	6
	4	35	18	9	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

3ème boucle

	Temps				
Fréquence	4	35	25	10	6
	5	32	18	10	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

	Temps				
Fréquence	35	35	35	25	6
	4	35	25	9	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

Résultat finale :

	Temps				
Fréquence	35	35	35	25	39
	37	35	51	52	39
	66	56	52	52	52
	66	66	56	52	38
	66	58	58	40	38
	57	58	41	36	38

Tableau de base :

	Temps				
Fréquence	4	35	25	10	6
	5	32	18	10	39
	37	9	51	52	34
	66	56	40	20	24
	57	58	40	14	38
	39	3	28	36	25

COMPARAISON

Amplitude	Temps	Fréquence
35	1	0
52	3	2
58	1	4
39	4	1
66	0	3
38	4	4
36	3	5

Filtre des amplitudes
les plus basses

Amplitude	Temps	Fréquence
35	1	0
52	3	2
58	1	4
39	4	1
66	0	3
38	4	4
36	3	5

II/Algorithme d'empreinte : Formation d'empreinte

Temps	Fréquence
1	0
3	2
1	4
4	1
0	3
4	4
3	5

Fréquence 1 Fréquence 2 Différence de temps

Une sous-empreinte

Sous -Empreintes possibles (42)

0 2 2	2 0 -2	4 0 0	1 0 -3	3 0 1	4 0 -3	5 0 -2
0 4 0	2 4 -2	4 2 2	1 2 -1	3 2 3	4 2 -1	5 2 0
0 1 3	2 1 1	4 1 3	1 4 -3	3 4 1	4 4 -3	5 4 -2
0 3 -1	2 3 -3	4 3 1	1 3 -4	3 1 4	4 1 0	5 1 1
0 4 3	2 4 1	4 4 3	1 4 0	3 4 4	4 3 -4	5 3 -3
0 5 2	2 5 0	4 5 2	1 5 -1	3 5 3	4 5 -1	5 4 1

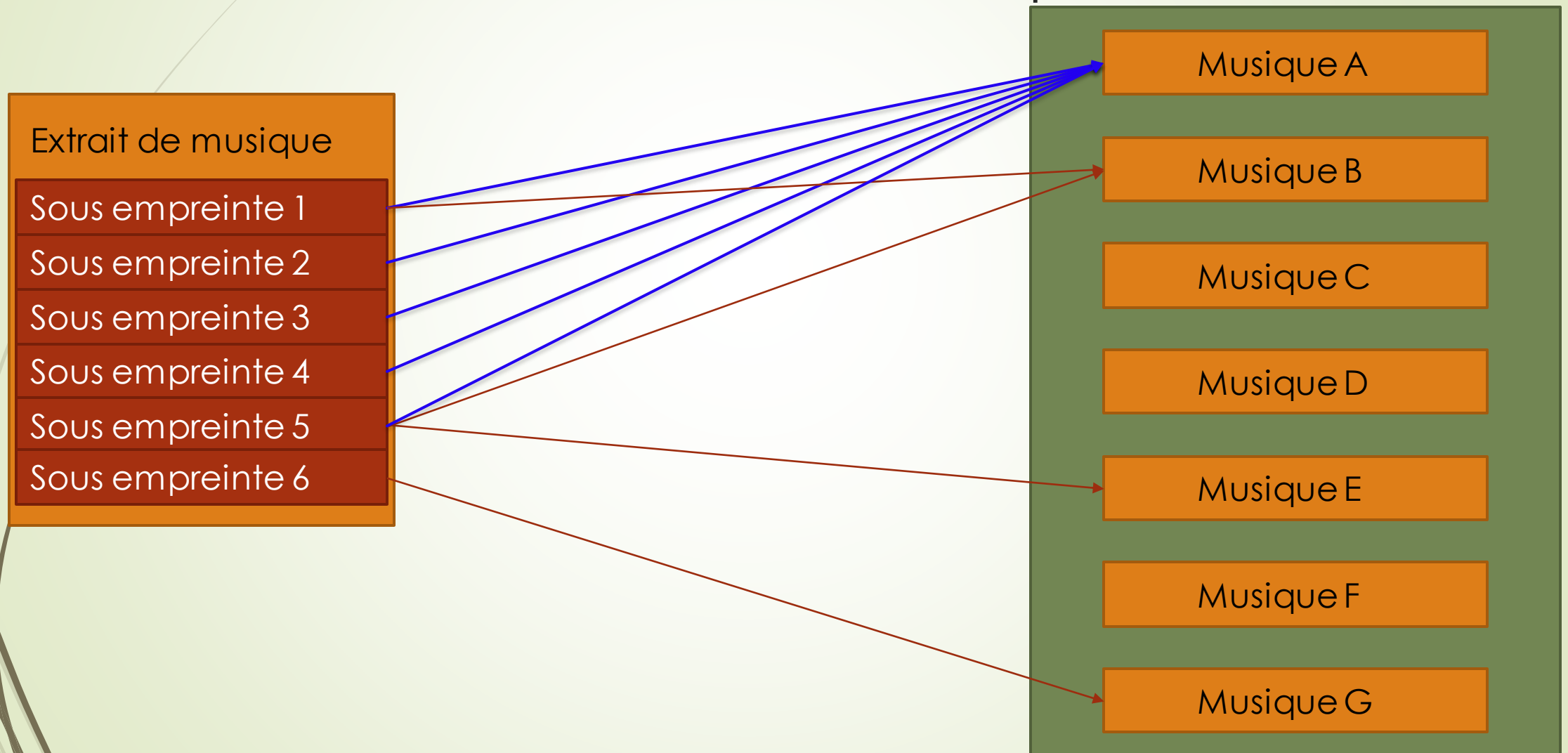
L'empreinte de l'extrait

Tri des sous-empreintes (degré = 1)

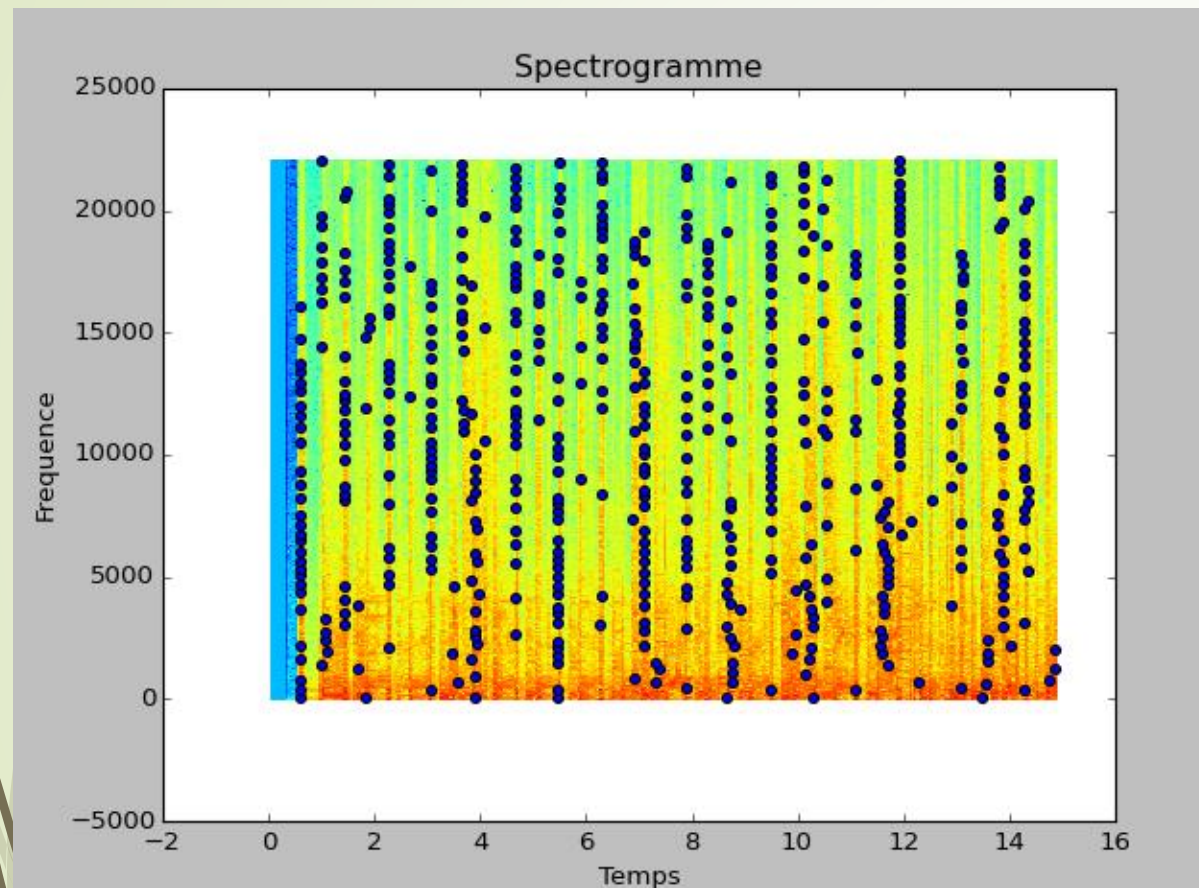
		4 0 0		3 0 1		
0 4 0			1 2 -1		4 2 -1	5 2 0
	2 1 1			3 4 1		
0 3 -1		4 3 1			4 1 0	5 1 1
	2 4 1		1 4 0			
	2 5 0		1 5 -1		4 5 -1	5 4 1

III/Base de données et résultats expérimentaux

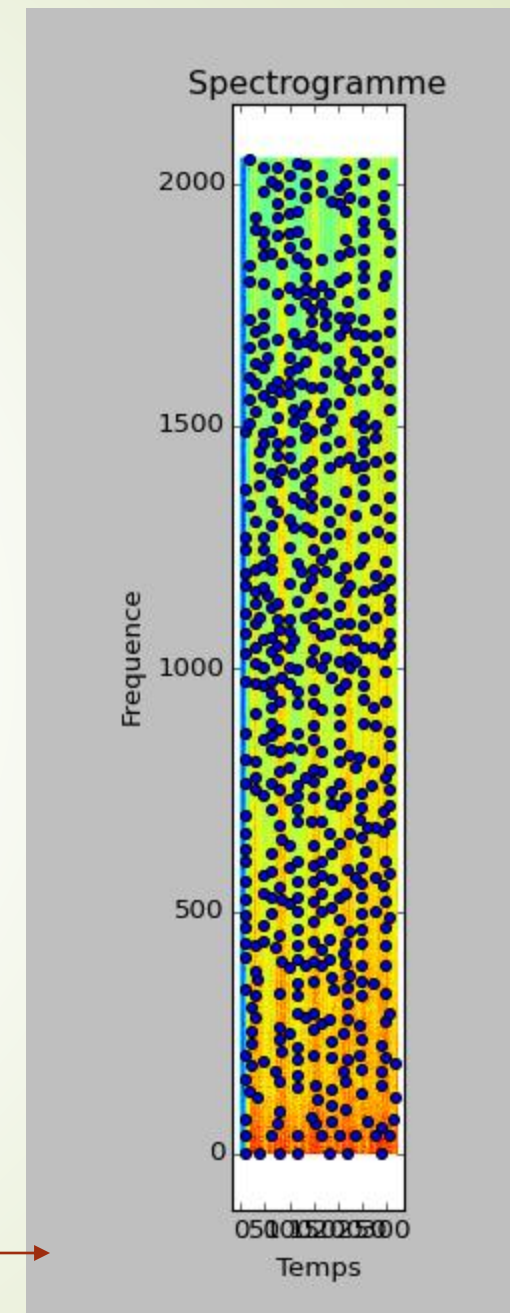
-Reconnaissance d'une musique



Algorithme d'empreinte: (Pour un morceau de 15 sec)



Spectre réel
Spectre, unité arbitraire



Efficacité

	id	Nom_du_chanteur	Album
	Filter	Filter	Filter
1	Essai	SSY	ost 1
2	Essai saturation 84%	SSY	ost 1
3	SSY mt	SSY	ost 1
4	Sultans Of Swing	Dire Strait	Sultans of swi...
5	Even Flow	Pearl Jam	Ten
6	Why Go	Pearl Jam	Ten
7	Jamming	Bob Marley	Exodus

Morceau de durée 3:43

```
empreinte_enregistre_base("15 Jamming.wav","Jamming")  
59.71375219622951
```

Morceau de durée 15 secondes

```
recognize("Essai4.wav")  
(('SSY mt',), 0.9061154177433247, 12.795865423975101)
```

Morceau de durée 5 secondes

```
recognize("Essai 12.wav")  
(('SSY mt',), 0.7429390094146541, 1.8416176565210662)
```

Morceau de durée 2 secondes

```
recognize("Essai 13.wav")  
(('SSY mt',), 0.04338070306656694, 1.02176788631823)
```



Conclusion

Ce qui a marché :

Reconnaissance quasi-parfaite (but recherché)

Ce qui est améliorable

Le temps de reconnaissance

Incrémentation de la base moins longue

