圧縮距離と作曲者判定

豊橋技術科学大学　梅村恭司

2017/9/29

話の流れ

- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

圧縮できる文字列・できない文字列

圧縮できる文字列:

パターンで構成されている文字列

ABC...ABCABC...ABC.....ABC...ABC

あまり圧縮できない文字列:

規則性のない文字列

B.C..AAB.AC..BCACA.CB

圧縮できる文字列中のパターン

圧縮できる文字列:

パターンで構成されている文字列

圧縮サイズ（30文字）

ABC...ABCABC...ABC.....ABC...

α:"ABC", β:"... "とすると

圧縮サイズ（10文字＋6文字）

αβααβαββαβ

圧縮できる文字列のコンパクトな表現

圧縮できる文字列:

パターンで構成されている文字列（30文字）

ABC...ABCABC...ABC.....ABC...

α:"ABC", β:"... "とすると

αβααβαββαβ（10文字＋6文字）

大きな部分を一つと扱うことで，辞書を使うことでコンパクトに表現できる。

話の流れ

- 圧縮プログラムの原理
- 圧縮プログラムを使った距離，
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

似ている/似ていない

ABC...ABCABC... と ...ABCABC...ABC は似ている。

ABC...ABCABC... と CBA...CBACBA... は似てない。

理由：構成しているパターンが違う。

似ている/似ていない

ABC...ABCABC... と ...ABCABC...ABC をつなげて圧縮すると

α:"ABC", β:"... "の２種類がパターン

圧縮サイズ（１０文字＋６文字）

ABC...ABCABC... と CBA...CBACBA... をつなげて圧縮すると,

α:"ABC", β:"... ", γ:"CBA"の３種類がパターン

圧縮サイズ（１０文字＋９文字）

→似ているものは一緒に圧縮すると小さい

圧縮プログラムによる文字列の距離

C（文字列）：文字列の圧縮サイズとする。

x, y：文字列で，xyはxとyつなげたものとする。

CDM（x，y）：圧縮による距離は下記で表す。

C（xy）/(C(x) + C(y))

CDMの直感的意味

別々に圧縮するの大きさ（分母）にくらべて，一緒に圧縮するとどのくらいの大きさ(分子）の比になるか。

圧縮プログラムによる文字列の距離

xとyが似ているとき

C(x) = C（"ABC...ABCABC... ") = 5+6

C(y) = C("...ABCABC...ABC") = 5+6

C(xy) = C("ABC...ABCABC.....ABCABC...ABC")=10+6

　ただし，C（文字列）: 文字列の圧縮サイズ

CDM（x，y）: 圧縮による距離

　C（xy）/(C(x) + C(y)) = 16/22

圧縮プログラムによる文字列の距離

xとyが似ていないとき

C(x) = C（"ABC...ABCABC... ") = 5+6

C(y) = C("CBA...CBACBA...") = 5+6

C(xy) = C("ABC...ABCABC...CBA...CBACBA...")=10+9

ただし，C（文字列）: 文字列の圧縮サイズ

CDM（x，y）: 圧縮による距離

C（xy）/(C(x) + C(y)) = 19/22

個xとyが似ているときより大きい

話の流れ

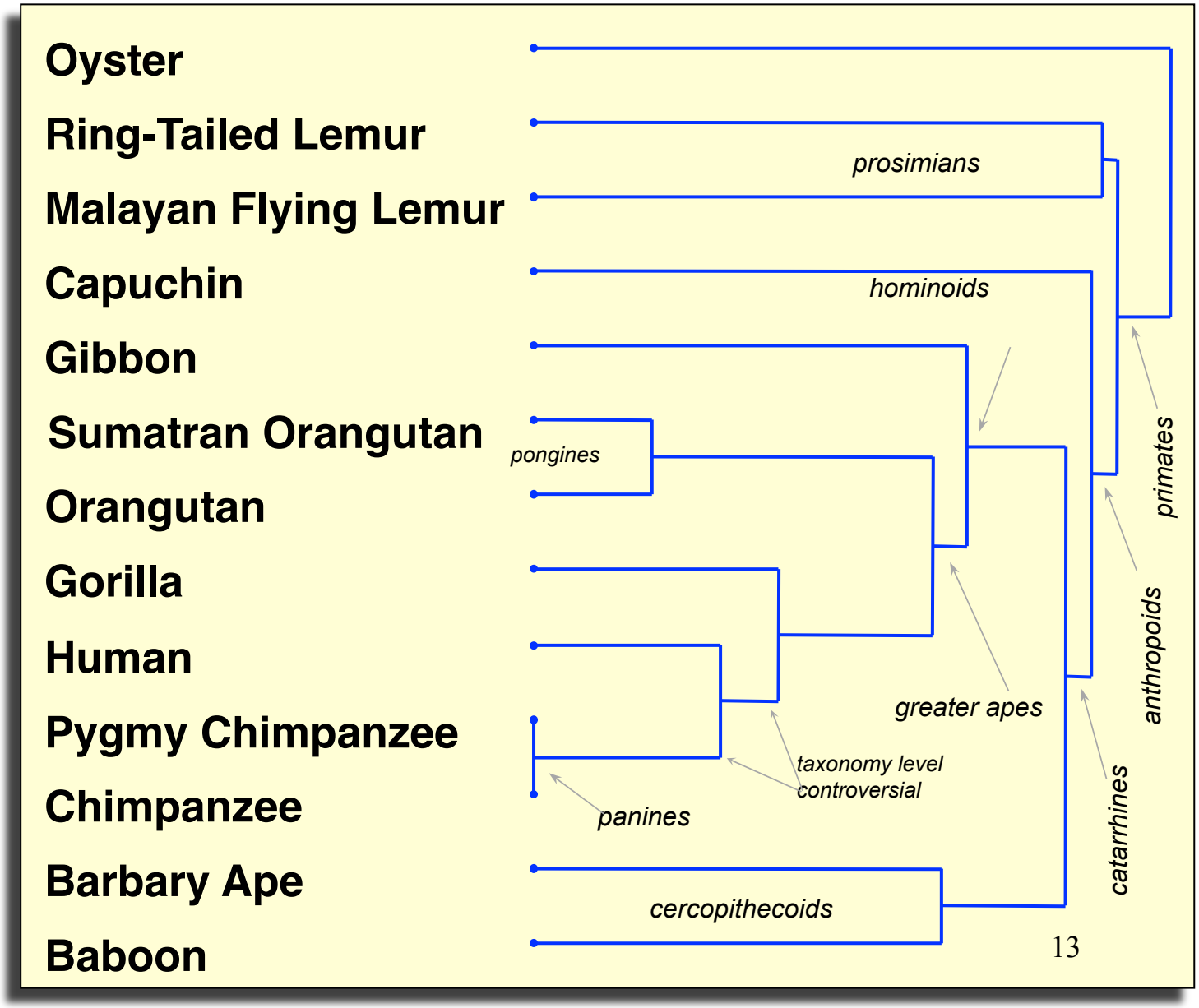
- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

The clustering achieved by CDM on 16,300 symbols from the mitochondrial DNA of 12 primates, and one "outlier" species.

This is the correct clustering for these species

Sang-Hee Lee

(Noted physical anthropologist)

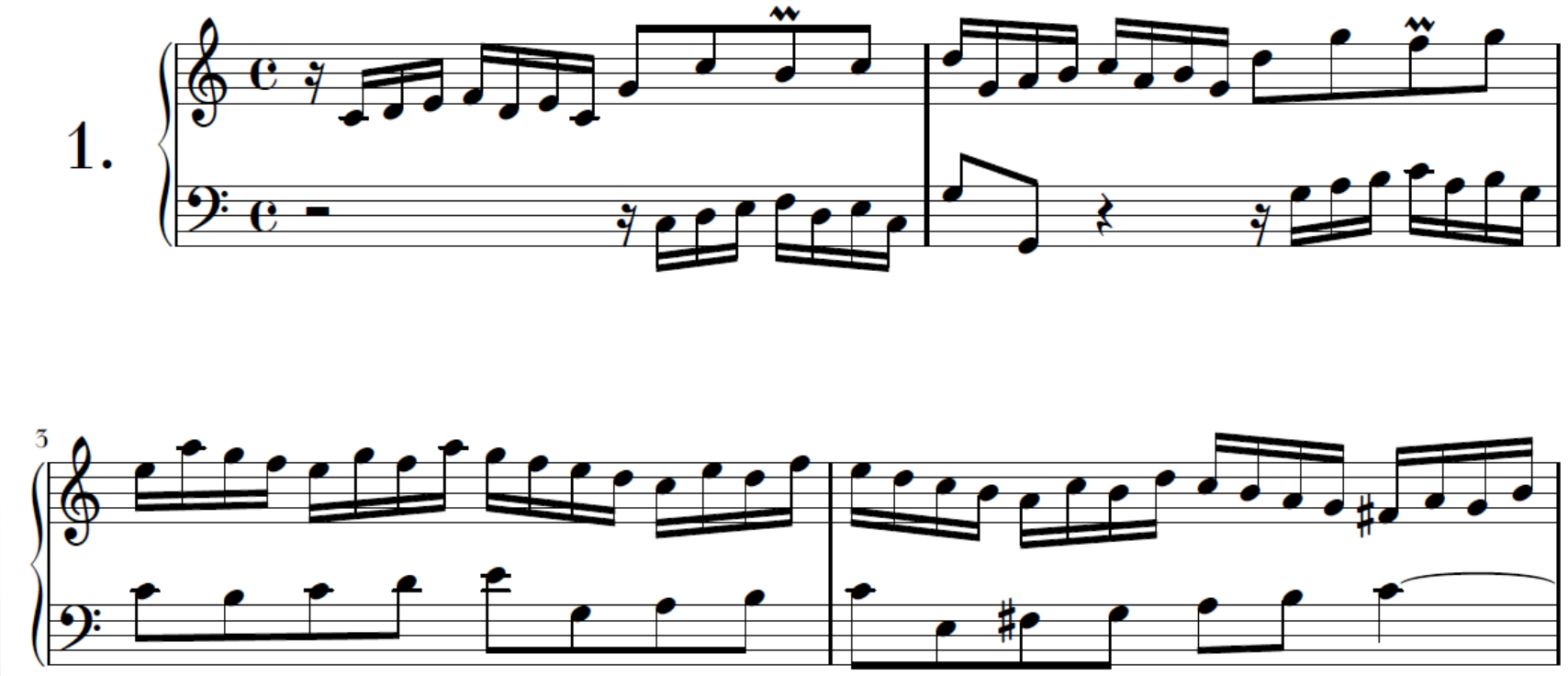


話の流れ

- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

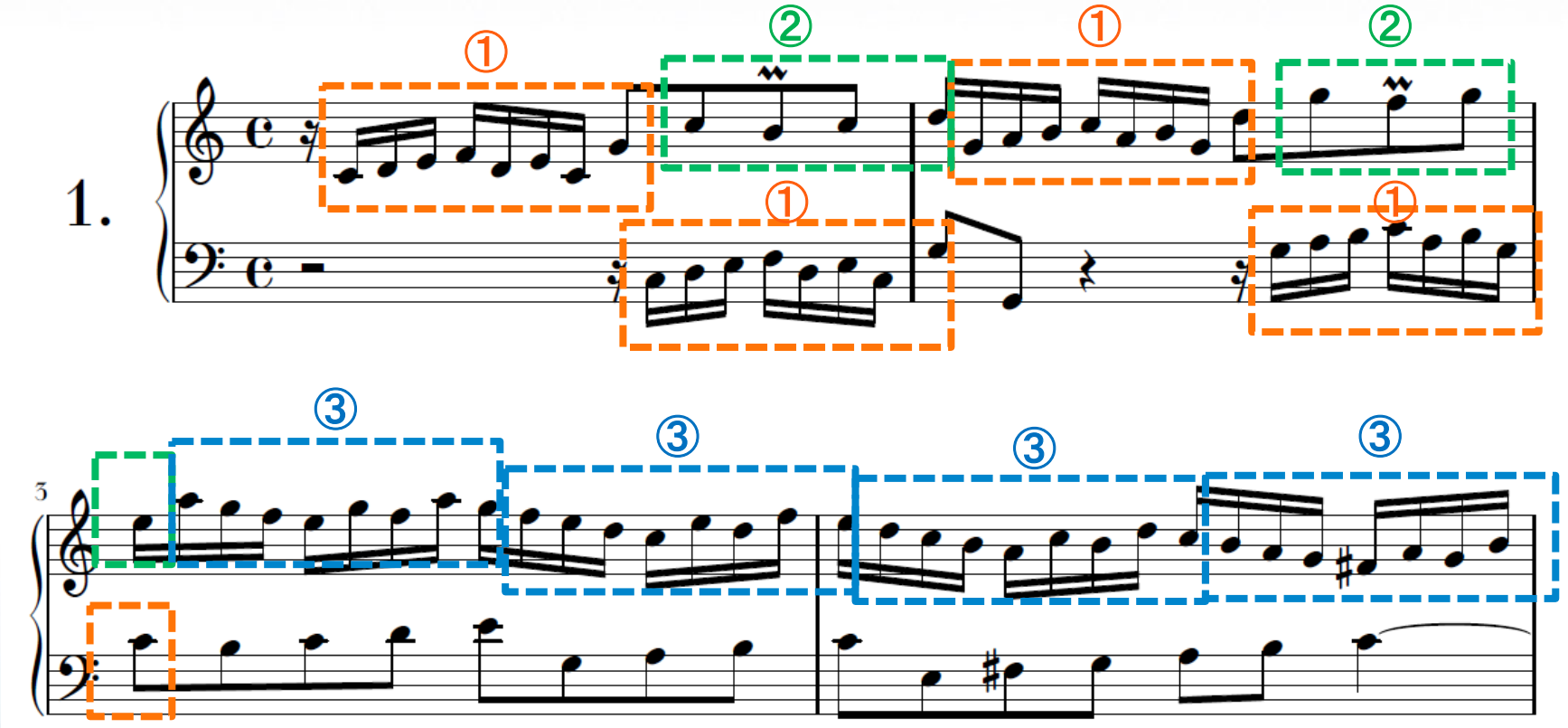
ある楽曲

例：J.S.Bach (Invention in C major, BWV 772)　冒頭4小節



楽曲の中のパターン

- 例：J.S.Bach (Invention in C major, BWV 772)　冒頭4小節



話の流れ

- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

作曲者の特徴

楽曲はいくつかのフレーズを繰り返すことで
構成されている

作曲者によって好みの「繰り返し」や
「類似したパターン」が存在するはず。

話の流れ

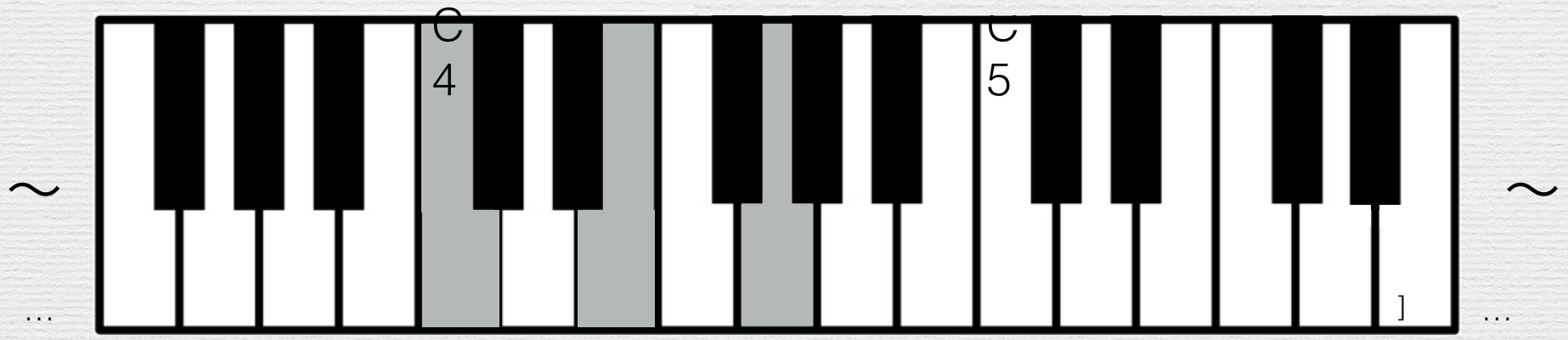
- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

楽曲の文字列化

作曲者判定を行うために楽曲を文字列化する必要がある

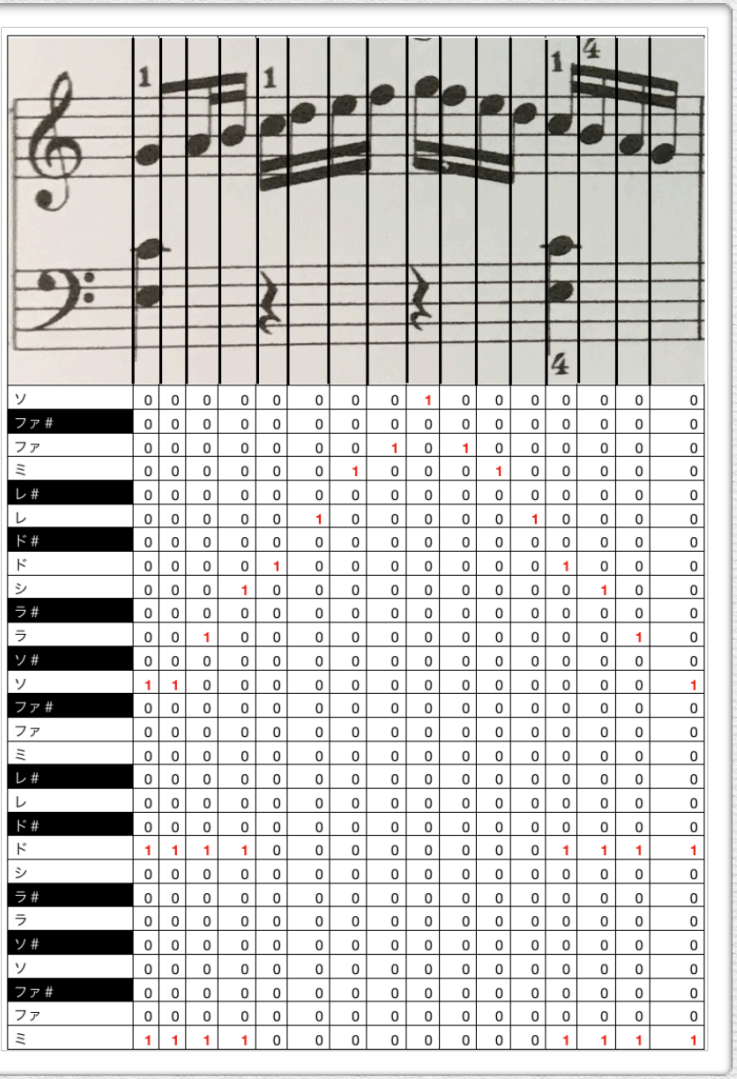
- 楽曲において鍵盤がなっているときを1，なっていない時は0として一つの長い文字列を生成し文字列化を行う
- 抽出間隔は16分音符間隔

楽曲の文字列化



```
00000000000000000000000000000000000000010001
00100000000000000000000000000000000000000000
```

楽曲の文字列化



ソ	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
ファ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ファ	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
ミ	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0
レ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
レ	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
ド＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ド	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
シ	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
ラ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ラ	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
ソ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ソ	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
ファ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ファ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ミ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
レ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
レ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ド＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ド	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
シ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ラ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ラ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ソ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ソ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ファ＃	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ファ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ミ	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1

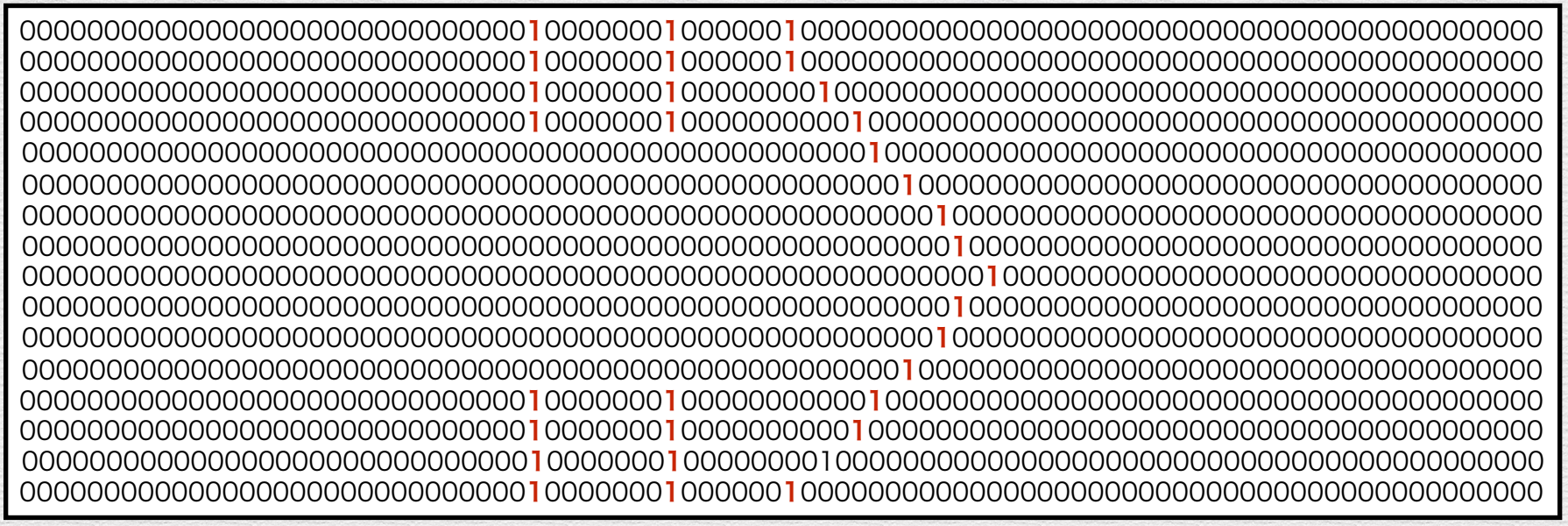
実際の楽曲を変換する場合

各タイミングごとにピッチベクトルを作成

- 一定の時間間隔で抽出
- 間隔は16分音符単位

全ての鍵盤に対し0もしくは1を割り当てる

文字列化した結果の例



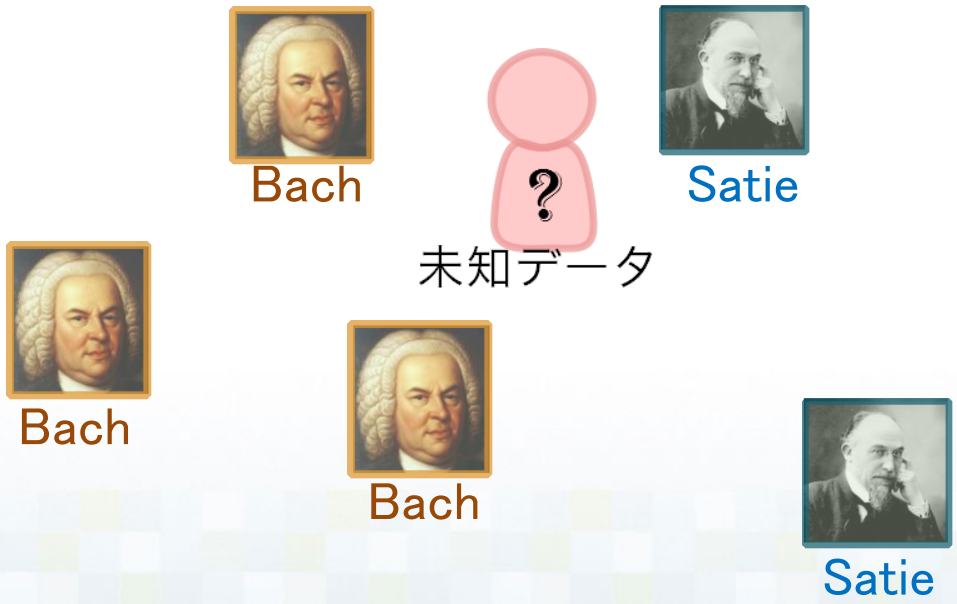
音の遷移を表現している文字列

話の流れ

- 圧縮プログラムの原理
- 圧縮プログラムを使った距離,
- 圧縮プログラムの利用例
- 楽曲の構造
- 作曲者の特徴
- 楽曲の文字列化
- 作曲判定

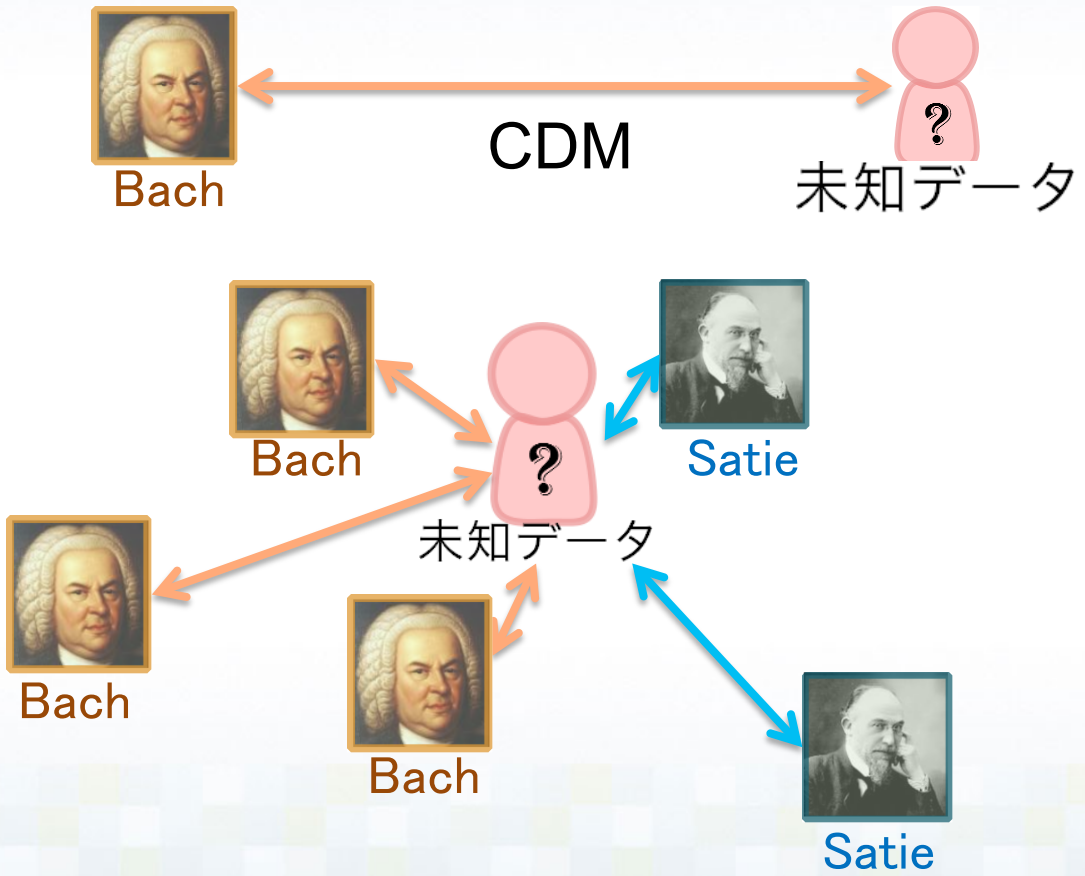
作曲者の判定方法(1/3)

- ある楽曲データの作曲者をわからないとき，2人の作曲者の他の作品から判定する



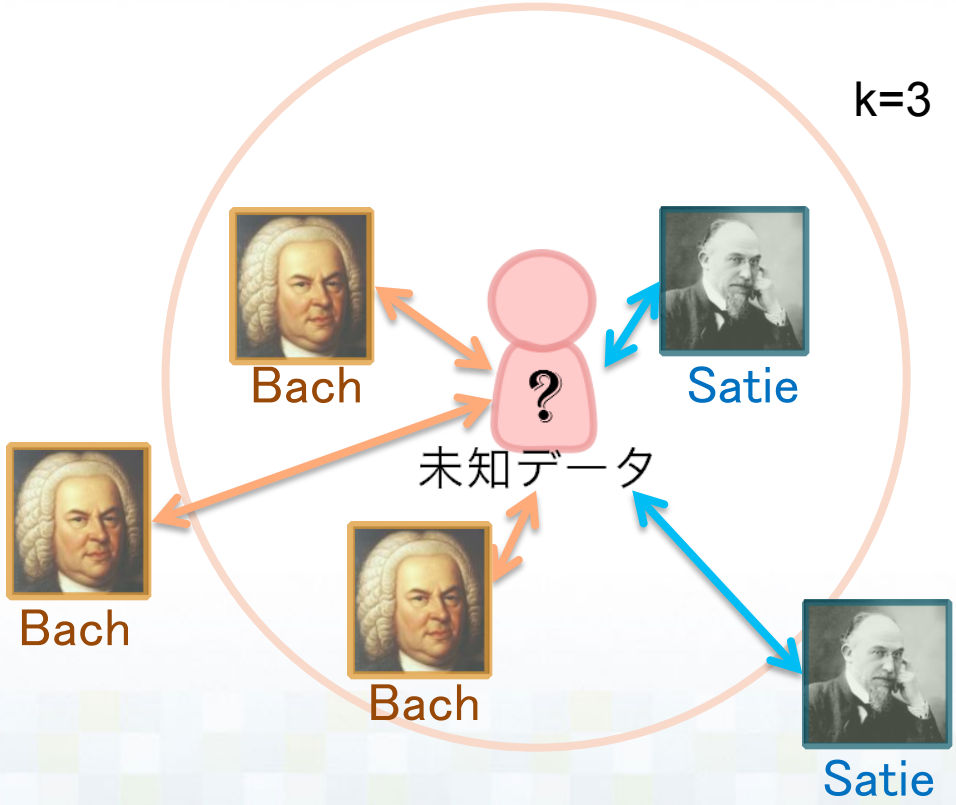
作曲者の判定方法(2/3)

- 楽曲を文字列化して，ある曲と他の曲の距離(CDM)を求める



作曲者の判定方法(3/3)

- ある曲に距離が近い順にk個選び，多数決で作曲者を決定する。kはたとえば3などがよく使われる。



実験結果
符号検定で判定ができるかどうか

網掛け：1%有意

５人の作曲者での判定の有意水準

P値	Bach	Mozart	Chopin	Debussy
Mozart	0.0081 (22/30)			
Chopin	0.0026 (23/30)	0.0007 (24/30)		
Debussy	0.0007 (24/30)	0.0000 (27/30)	0.0007 (24/30)	
Satie	0.0000 (26/30)	0.0026 (23/30)	0.0026 (23/30)	0.0007 (24/30)

- 表のなかは，

  上段：帰無仮説：「判定ができていない」に対する有意水準，　下段(正解数/全曲数）

  判定対象の曲は，両方の作曲者で同数（15曲）
- 全ての作曲者の組み合わせにおいて1%有意を達成

この方法で判定ができているといえる。＝＞作曲者ごとに好きなパターンがある