ピッチ情報を援用した圧縮類似度による方言の自動分類

Automatic classification of Japanese dialects using compression similarity distance assisted by pitch information

谷研究室

齋藤 岳丸　山本 峻　鴫原 克幸

目次

目次

1．はじめに - 先行研究

『圧縮情報距離に基づく方言の自動分類』

・方言ももたろう

監修・著：杉藤　美代子

昔話「桃太郎」を全国56地方の各方言による語り（音声データ）を収録してあるソフトウェア

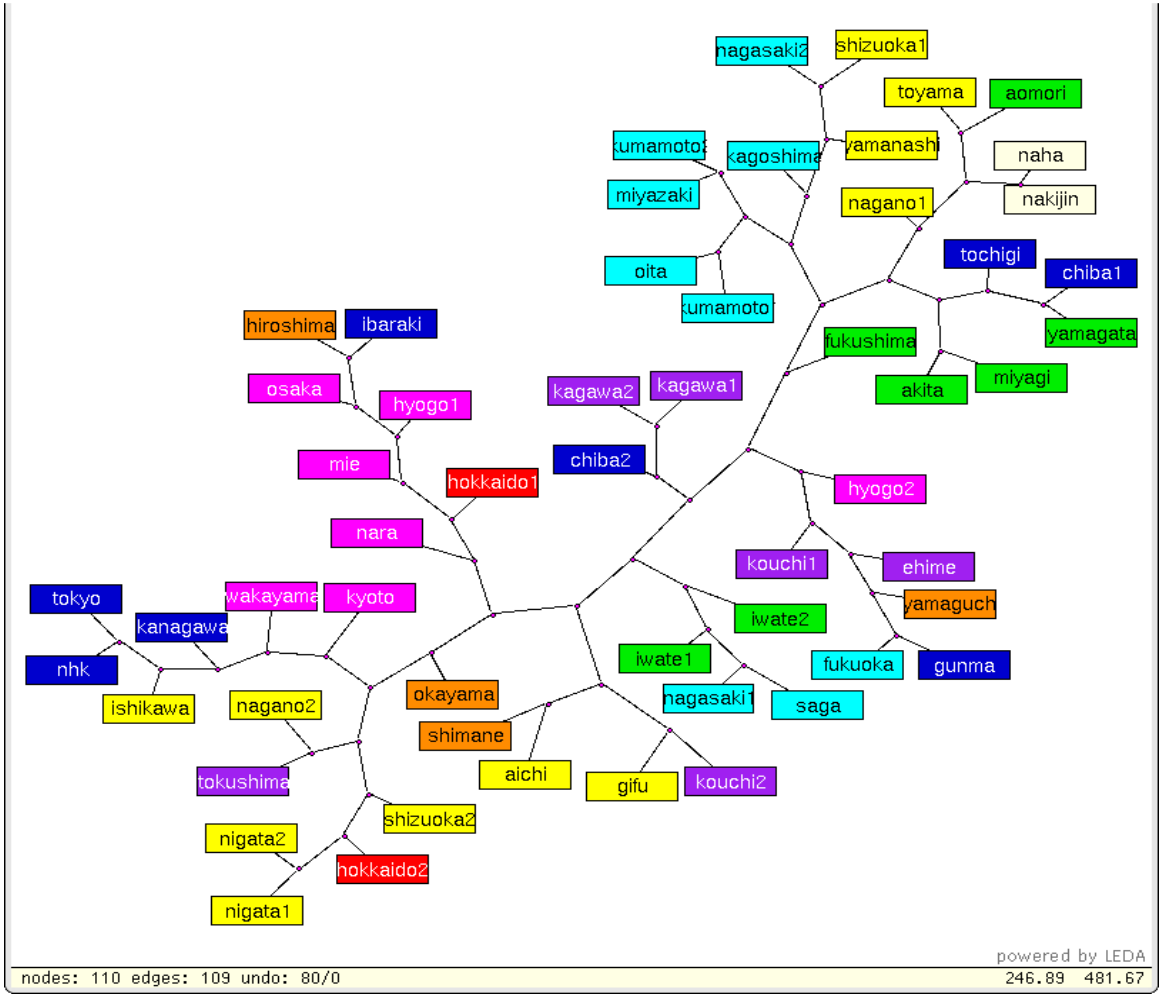
引用元
2008年　圧縮チーム

1．はじめに - 先行研究

入力例:

NHK	むかしむかしあるところにおじいさんとおばあさんがありました。
岩手	むがすむがすあるどころにおずうさんとおばあさんがいだあど。
静岡	ずうっとみゃあにあるとこでじいじいとばあばあみて。
大阪	むかしむかしあることろにおじいさんとおばあさんがすんではってん。
那覇	むかしむかしあるところんかいたんめえとんめえがめっせえびいたん。

自動分類



１．はじめに－先行研究

『圧縮情報距離に基づく方言の自動分類』

テキストファイルをデータ間の類似度を用いて自動的に分類

過去の入力

2007年　音声ファイルを人が聞いてそれを手作業でテキスト化したもの
　　とてもよい結果が出たが、以下の２点の問題があった
　　・テキスト化に人間が介入している
　　・方言で重要なはずの音声情報を全て捨てている

2008年　テキスト化に「ドラゴンスピーチ」ソフトを使い自動的にテキスト化したもの
　　2007年研究における問題点の１つを解決するために行った

１．はじめに−目的

・テキスト化に人間が介入している
2008年：自動化により改善

・方言で重要な音声情報を全て捨てている
本年度：音声情報であるピッチを援用する

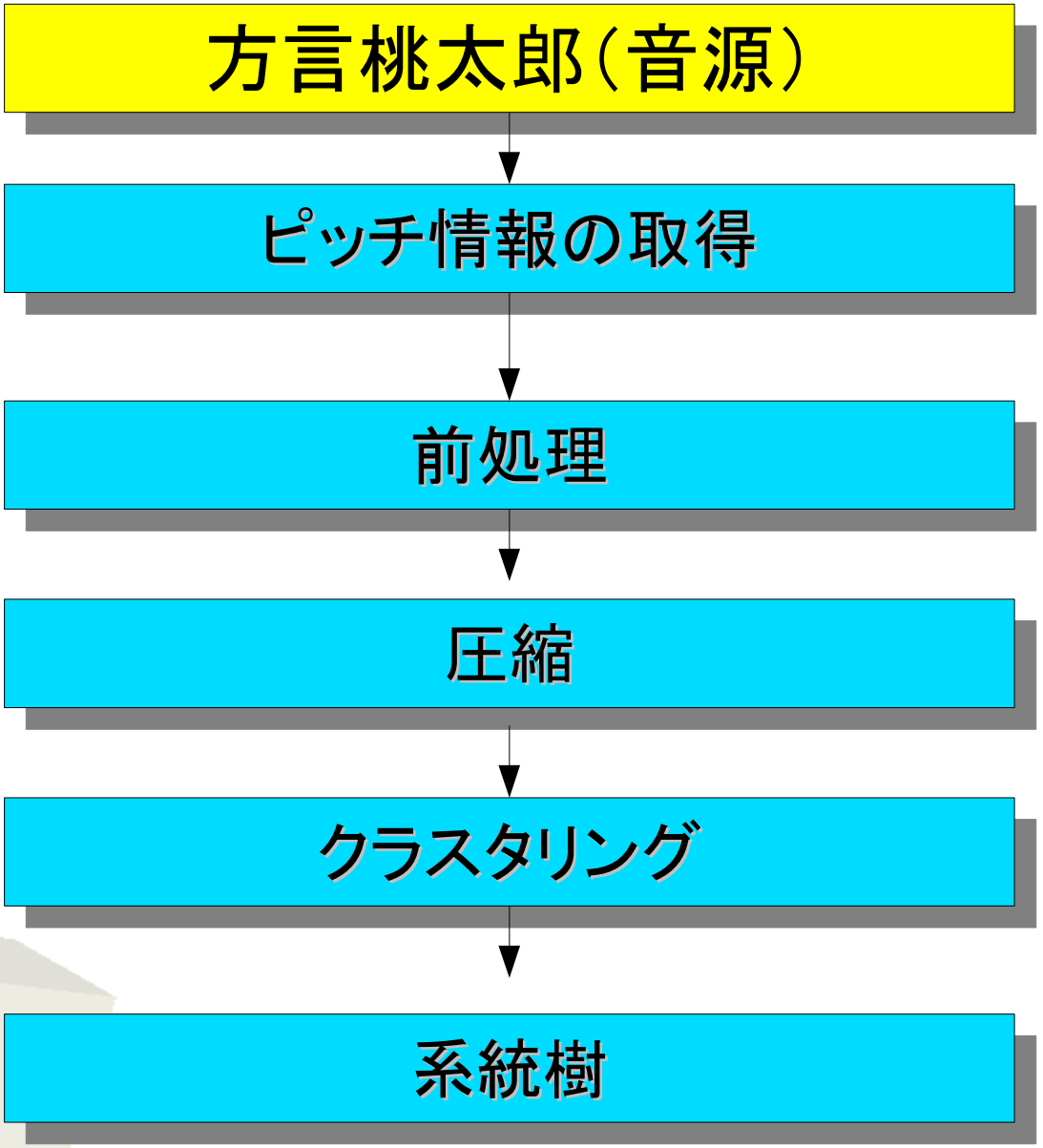
本研究

前年度までのテキストのみのデータに加え
音声情報を付加することで分類の精度をあげる。

目次

１．はじめに−研究概要

研究の手順



１．はじめに−研究概要

『方言桃太郎』

監修・著：杉藤　美代子

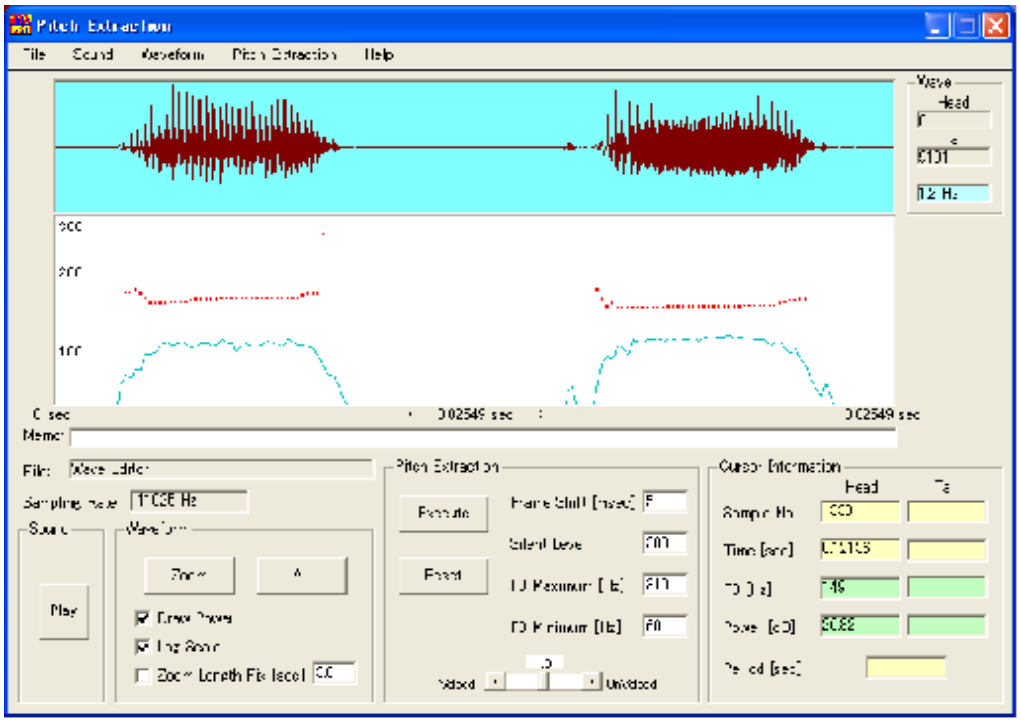
昔話「桃太郎」を全国56地方の各方言による語り
（音声データ）を収録してあるソフトウェア。

・録音時期は1989-1992に録音された。
・現在、その時期と同じように方言を喋れる話者はいない。
・地域によって録音環境が悪いところがある。

ピッチ抽出

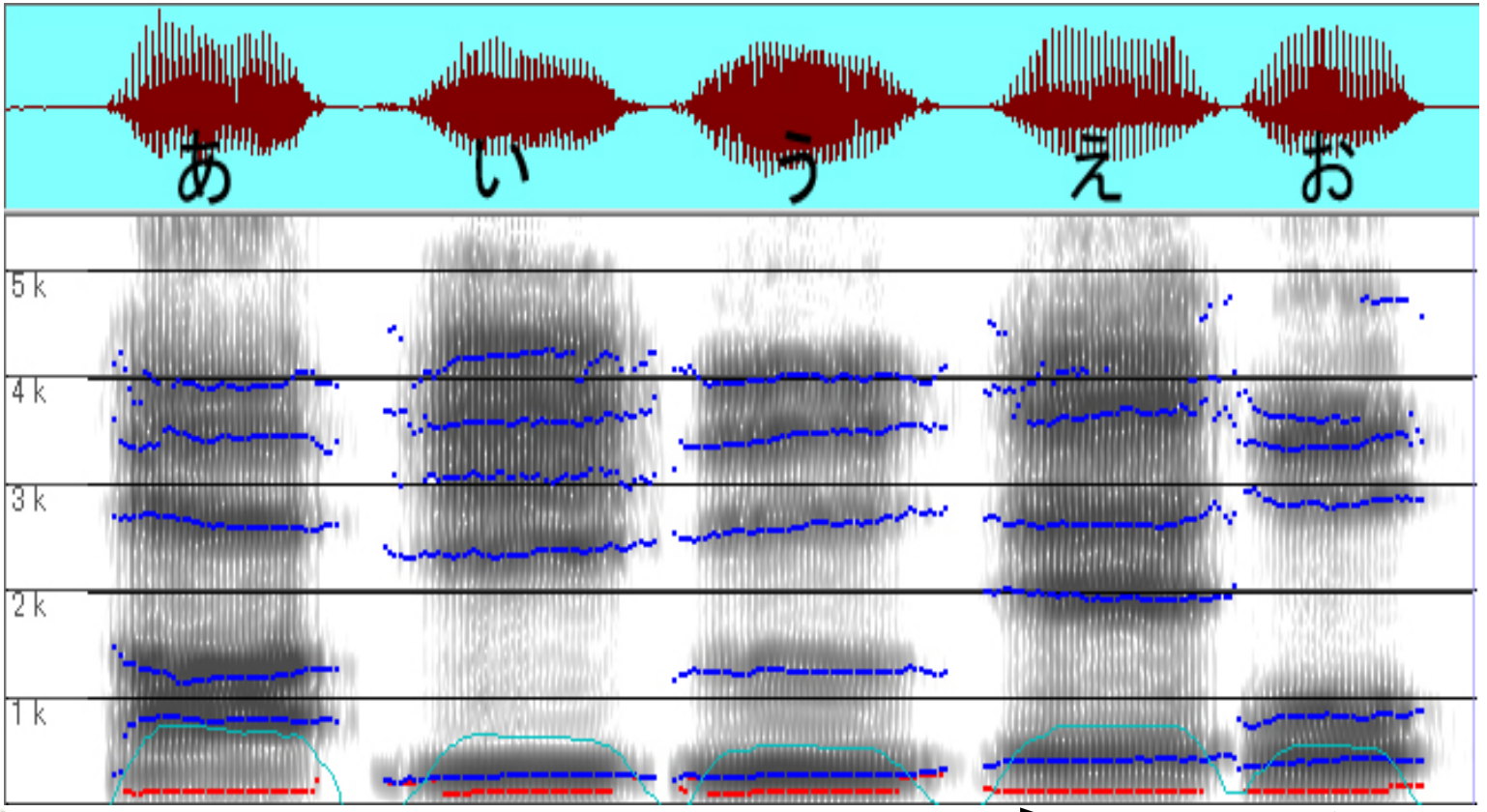
『音声録聞見』

『音声研究入門』（編　今石 元久）に付属
２００５年発行



東京大学 大学院 医学系研究科 認知・言語医学講座で
開発されたソフトウェア。

ピッチ情報とは



ピッチ ： 声帯の振動数

ピッチについて

基本周波数

音声解析においては、声帯の振動数を指す。

・基本周波数は性差、個人差によって変わる

・基本周波数は声帯が長いほど低くなる

・声帯が振動するのは母音の発音時

平均的には
男性：125Hz
女性：200Hz
子供：300Hz

ピッチについて

基本周波数

音声解析においては、声帯の振動数を指す。

・基本周波数は性差、個人差によって変わる

・基本周波数は声帯が長いほど低くなる

・声帯が振動するのは母音の発音時

平均的には
男性：125Hz
女性：200Hz
子供：300Hz

なぜピッチか

ピッチ情報に方言の地域差による
特徴が反映されていると考えたから

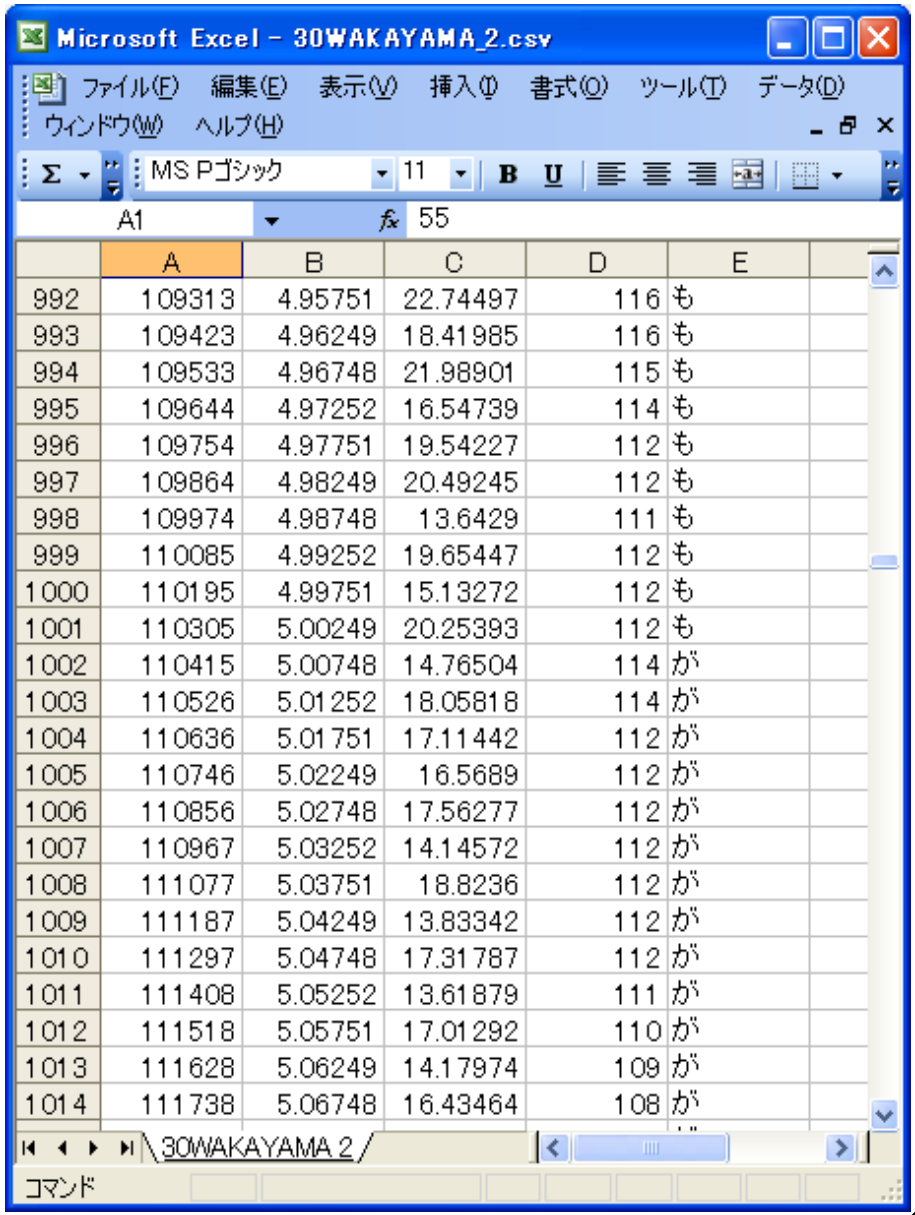
ピッチ情報の取得

音声録見聞というソフトウェアにより

１秒間を２００個に分割した

それぞれの地点でのピッチ周波数の

値が得られる



	A	B	C	D	E
992	109313	4.95751	22.74497	116	も
993	109423	4.96249	18.41985	116	も
994	109533	4.96748	21.98901	115	も
995	109644	4.97252	16.54739	114	も
996	109754	4.97751	19.54227	112	も
997	109864	4.98249	20.49245	112	も
998	109974	4.98748	13.6429	111	も
999	110085	4.99252	19.65447	112	も
1000	110195	4.99751	15.13272	112	も
1001	110305	5.00249	20.25393	112	も
1002	110415	5.00748	14.76504	114	が
1003	110526	5.01252	18.05818	114	が
1004	110636	5.01751	17.11442	112	が
1005	110746	5.02249	16.5689	112	が
1006	110856	5.02748	17.56277	112	が
1007	110967	5.03252	14.14572	112	が
1008	111077	5.03751	18.8236	112	が
1009	111187	5.04249	13.83342	112	が
1010	111297	5.04748	17.31787	112	が
1011	111408	5.05252	13.61879	111	が
1012	111518	5.05751	17.01292	110	が
1013	111628	5.06249	14.17974	109	が
1014	111738	5.06748	16.43464	108	が

ピッチ情報の取得

研究データ

拡張子WAVの音声ファイルを音声録聞見を用いて
CSVで書き出し、ピッチ情報を元にひらがなで
音声割り当てを行う。

	A	B	C	D	E
103	サンプル番号	再生位置(秒)	音量(dB)	ピッチ(Hz)	割り当て
104	14388	0.65	22.46	124	む
105	14498	0.66	25.94	124	む
106	14608	0.66	24.62	122	む
107	14718	0.67	18.44	119	む
108	14829	0.67	19.74	117	む
109	14939	0.68	15.47	110	む
110	15049	0.68	16.23	102	む
111	15159	0.69	12.53	7	む
112	15270	0.69	6.32	7	む
113	15380	0.7	6.01	7	む
114	15490	0.7	-3.27	7	む
115	15600	0.71	-3.14	0	む
116	15711	0.71	-8.87	0	む
117	15821	0.72	-8.64	0	む
118	15931	0.72	-2.23	0	か
119	16041	0.73	-9.46	0	か
120	16152	0.73	-8.87	7	か
121	16262	0.74	11.84	7	か

目次

音声処理 - ノイズ除去

ノイズ除去

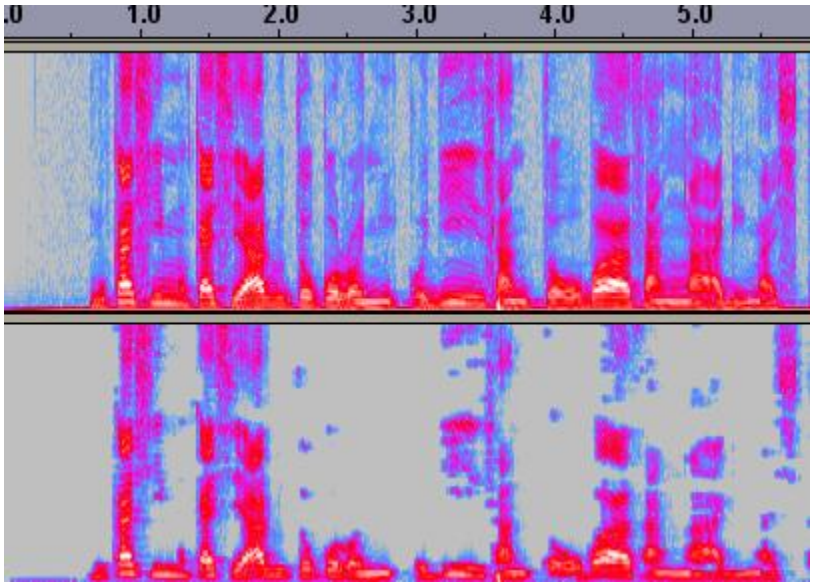
音源データの方言桃太郎は地域によってはノイズがが酷く、取得するピッチ情報に影響があるで全ファイルに対してノイズ除去を施す

使用ソフト

Audacity

ノイズ除去の他に
さまざまな効果をつけたり
編集できたりと多機能な
ソフトウェア

無処理



ノイズ除去後

23 ISHIKAWA1

目次

前処理について

■先行研究での前処理

先行研究では文字情報での比較を行っていた。
圧縮類似度を算出する際には無駄な情報を省くため
文字を数値に置き換えて単純化する。

単純化例

文字	SJIS	EUC	UTF-8	変換
あ	0x82A0	0xA4A2	0xE38182	0x02
か	0x82A9	0xA4AB	0xE3818B	0x0B
さ	0x82B3	0xA4B5	0xE38195	0x15

ひらがなに対して文字コード順に番号を割り当て
１文字1byteの情報に変換を行う。

前処理について

■本研究での前処理

本研究では文字情報にピッチ情報を組み込む必要があるため

先行研究よりも複雑かつ方法が無限に考えられるため、本研究の要点ともいえる。

今回は前処理を２種類検討し、実験を行った。

前処理1

- 文字情報
  文字情報の単純化を行い、2byteで表す
- ピッチ情報
  取得したピッチ情報の値を2byteで表す

文字情報	文字情報	単純化	ピッチ情報(Hz)	単純化
お	10	0x000A	116	0x0074
ば	48	0x0030	155	0x009B
あ	2	0x0002	145	0x0091
さ	21	0x0015	98	0x0062
ん	83	0x0053	130	0x0082
が	12	0x000C	140	0x008C

文字情報とピッチ情報を交互に並べる

完成データ(16進数表現)

000A 0074 0030 009B 0002 0091 0015 0062 0053 0082 000C 008C

前処理2

- 文字情報

  文字情報の単純化を行い、1byteで表す

- ピッチ情報

  その話者のピッチ値の平均をとり、128(1byte正整数中央値)との差をオフセット値とする。ある時点のピッチ情報とオフセット値の差をピッチ情報として扱い、1byteで表す。

ピッチ情報平均値：147
オフセット値：19

文字	文字情報	単純化	ピッチ情報	単純化
お	10	0x0A	116 - 19 = 97	0x61
ば	48	0x30	155 - 19 = 136	0x88
あ	2	0x02	145 - 19 = 126	0x7E
さ	21	0x15	98 - 19 = 79	0x4F
ん	83	0x53	130 - 19 = 111	0x6F
が	12	0x0C	140 - 19 = 121	0x79

文字情報とピッチ情報を交互に並べる

完成データ(16進数表現)

0A61 3088 027E 154F 536F 0C79

前処理1

ピッチ　→　値を1byteで出力
文字　　→　数値化して1byteで出力

Microsoft Excel - 30WAKAYAMA_2.csv

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D)
ウィンドウ(W) ヘルプ(H)

MS Pゴシック 11 B U

A1 fx 55

	A	B	C	D	E
992	109313	4.95751	22.74497	116	も
993	109423	4.96249	18.41985	116	も
994	109533	4.96748	21.98901	115	も
995	109644	4.97252	16.54739	114	も
996	109754	4.97751	19.54227	112	も
997	109864	4.98249	20.49245	112	も
998	109974	4.98748	13.6429	111	も
999	110085	4.99252	19.65447	112	も
1000	110195	4.99751	15.13272	112	も
1001	110305	5.00249	20.25393	112	も
1002	110415	5.00748	14.76504	114	が
1003	110526	5.01252	18.05818	114	が
1004	110636	5.01751	17.11442	112	が
1005	110746	5.02249	16.5689	112	が
1006	110856	5.02748	17.56277	112	が
1007	110967	5.03252	14.14572	112	が
1008	111077	5.03751	18.8236	112	が
1009	111187	5.04249	13.83342	112	が
1010	111297	5.04748	17.31787	112	が
1011	111408	5.05252	13.61879	111	が
1012	111518	5.05751	17.01292	110	が
1013	111628	5.06249	14.17974	109	が
1014	111738	5.06748	16.43464	108	が

30WAKAYAMA_2

コマンド

[aichi.data]

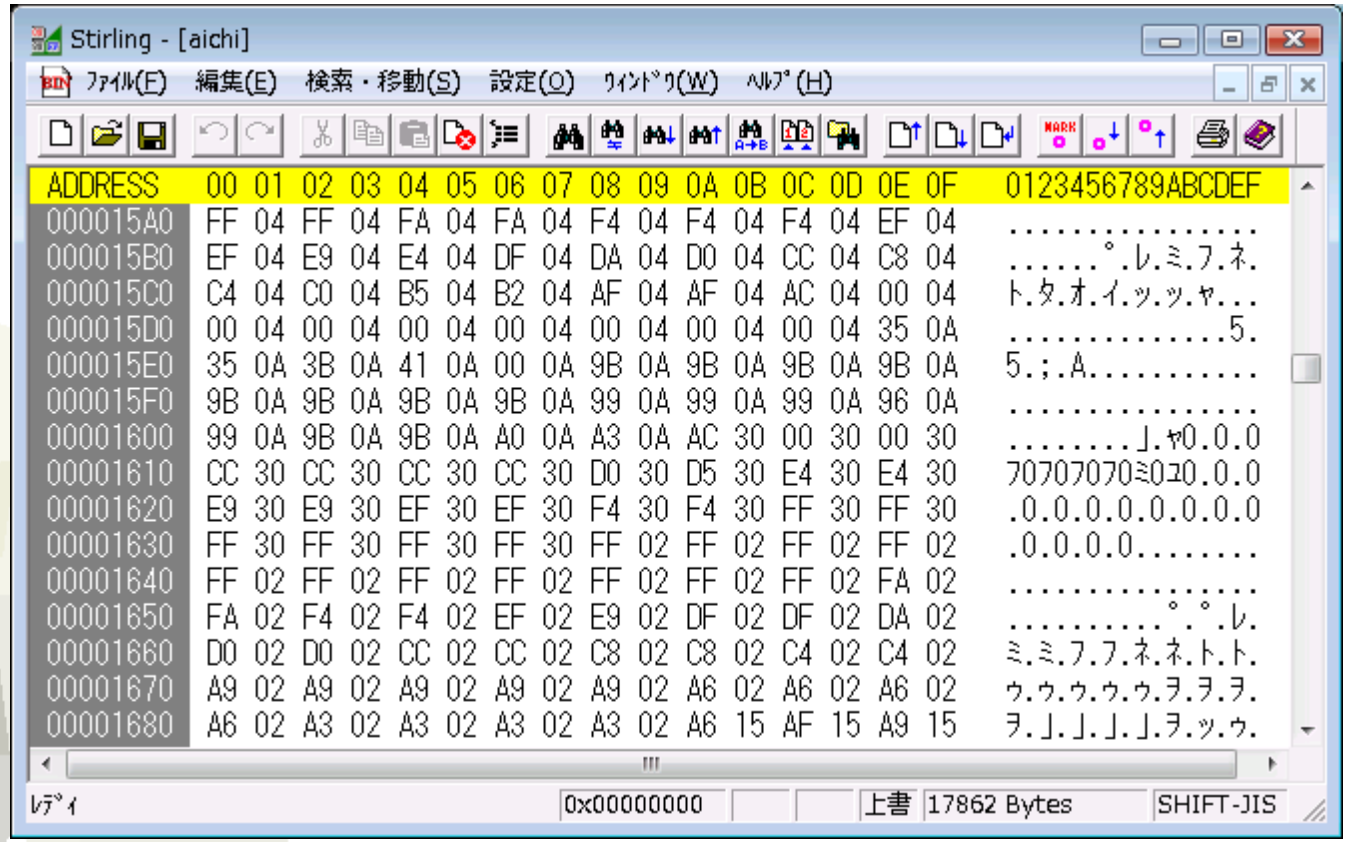
編集(E) 検索・移動(S) 設定(O) ｳｨﾝﾄﾞｳ(W) ﾍﾙﾌﾟ(H)

```
   00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   0123456789ABCDEF
00 9E 00 4F 00 A0 00 4F 00 A0 00 4F 00 A0 00 4F   ...O...O...O...O
00 A0 00 4F 00 A0 00 4F 00 A0 00 4F 00 9E 00 4F   ...O...O...O...O
00 9E 00 4F 00 9B 00 4F 00 99 00 4F 00 99 00 4F   ...O...O...O...O
00 97 00 4F 00 95 00 4F 00 C5 00 3F 00 BB 00 3F   ...O...O.ﾅ.?.ｻ.?
00 AF 00 3F 00 A5 00 3F 00 AA 00 3F 00 A7 00 3F   .ｯ.?.･.?.ｪ.?.ｧ.?
00 A7 00 3F 00 A7 00 3F 00 A7 00 3F 00 A7 00 3F   .ｧ.?.ｧ.?.ｧ.?.ｧ.?
00 A7 00 3F 00 AA 00 3F 00 AC 00 3F 00 AF 00 3F   .ｧ.?.ｪ.?.ｬ.?.ｯ.?
00 B2 00 3F 00 B2 00 3F 00 B5 00 3F 00 B8 00 3F   .ｲ.?.ｲ.?.ｵ.?.ｸ.?
00 BB 00 3F 00 BB 00 3F 00 BE 00 3F 00 C1 00 3F   .ｻ.?.ｻ.?.ｾ.?.ﾁ.?
00 C5 00 3F 00 C8 00 3F 00 C8 00 3F 00 D0 00 3F   .ﾅ.?.ﾈ.?.ﾈ.?.ﾐ.?
00 D0 00 43 00 D4 00 43 00 D4 00 43 00 D4 00 43   .ﾐ.C.ﾔ.C.ﾔ.C.ﾔ.C
00 D4 00 43 00 D4 00 43 00 6F 00 43 00 72 00 43   .ﾔ.C.ﾔ.C.o.C.r.C
00 73 00 43 00 74 00 43 00 74 00 43 00 74 00 43   .s.C.t.C.t.C.t.C
00 74 00 43 00 74 00 43 00 73 00 43 00 72 00 43   .t.C.t.C.s.C.r.C
00 70 00 43 00 70 00 43 00 72 00 43 00 72 00 43   .p.C.p.C.r.C.r.C
```

0x00000000 上書 35724 Bytes SHIFT-JIS

前処理2

ピッチ　→　値の平均を中央値として、その振れ幅を1byteで出力
文字　　→　数値化して1byteで出力



目次

圧縮情報距離

先行研究より、本研究ではBZIP2を用いて距離表を算出する。

圧縮類似度に基づく
データ間の類似度を求める式

$$d(x,y) \approx \frac{bzip2(xy) - bzip2(x)}{bzip2(y)}$$

圧縮情報距離

名前	サイズ	種類
aichi	423 KB	ファイル
akita	336 KB	ファイル
aomori	358 KB	ファイル
chiba1	201 KB	ファイル
chiba2	261 KB	ファイル
ehime	335 KB	ファイル
fukuoka	287 KB	ファイル
fukushima	246 KB	ファイル
gifu	329 KB	ファイル
gunma	268 KB	ファイル
hirosima1	341 KB	ファイル
hokkai1	391 KB	ファイル
hokkai2	283 KB	ファイル
hyogo1	307 KB	ファイル
hyogo2	328 KB	ファイル
ibaraki	267 KB	ファイル
isikawa1	234 KB	ファイル
iwate1	309 KB	ファイル
iwate2	222 KB	ファイル
kagawa1	309 KB	ファイル
kagawa2	317 KB	ファイル
kagosima	255 KB	ファイル
kanagawa	193 KB	ファイル
kouti1	391 KB	ファイル
kouti2	247 KB	ファイル

$$d(x,y) \approx \frac{bzip2(xy) - bzip2(x)}{bzip2(y)}$$

文字割り当てを行い
前処理をくわえたファイル

圧縮情報距離

名前	サイズ	種類
aichi	423 KB	ファイル
akita	336 KB	ファイル
aomori	358 KB	ファイル
chiba1	201 KB	ファイル
chiba2	261 KB	ファイル
ehime	335 KB	ファイル
fukuoka	287 KB	ファイル
fukushima	246 KB	ファイル
gifu	329 KB	ファイル
gunma	268 KB	ファイル
hirosima1	341 KB	ファイル
hokkai1	391 KB	ファイル
hokkai2	283 KB	ファイル
hyogo1	307 KB	ファイル
hyogo2	328 KB	ファイル
ibaraki	267 KB	ファイル
isikawa1	234 KB	ファイル
iwate1	309 KB	ファイル
iwate2	222 KB	ファイル
kagawa1	309 KB	ファイル
kagawa2	317 KB	ファイル
kagosima	255 KB	ファイル
kanagawa	193 KB	ファイル
kouti1	391 KB	ファイル
kouti2	247 KB	ファイル

$$d(x,y) \approx \frac{bzip2(xy)-bzip2(x)}{bzip2(y)}$$

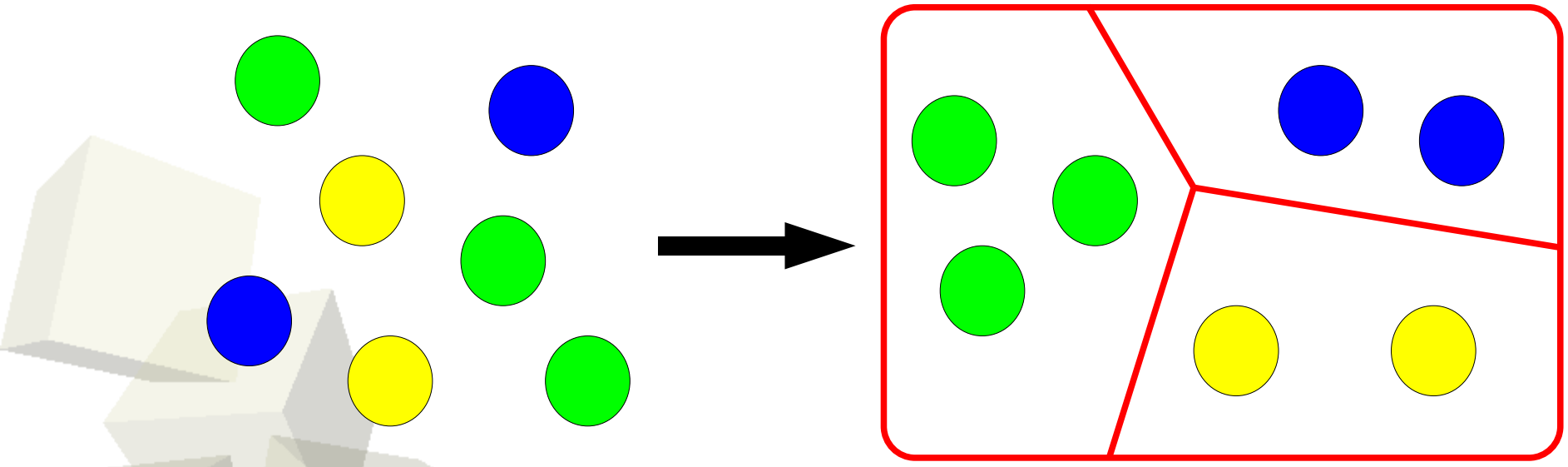
```
 1 aichi_akita.bz2 0.9483932246319455
 2 aichi_aomori.bz2 0.8390058572107013
 3 aichi_chiba1.bz2 0.8656007598543612
 4 aichi_chiba2.bz2 0.9602659490264366
 5 aichi_ehime.bz2 0.9442773468418553
 6 aichi_fukuoka.bz2 0.9442773468418553
 7 aichi_fukushima.bz2 0.9490264365996517
 8 aichi_gifu.bz2 0.8687668196928922
 9 aichi_gunma.bz2 0.8448630679119835
10 aichi_hirosima.bz2 0.921323413012506
11 aichi_hokkai1.bz2 0.9325629254392908
12 aichi_hokkai2.bz2 0.8936203894253602
13 aichi_hyogo1.bz2 0.8494538546778534
14 aichi_hyogo2.bz2 0.8717745765394966
15 aichi_ibaraki.bz2 0.8576856102580339
16 aichi_ishikawa.bz2 0.8369479183156562
17 aichi_iwate1.bz2 0.8353648883963907
18 aichi_iwate2.bz2 0.8955200253284787
19 aichi_kagawa1.bz2 0.854836156403356
20 aichi_kagawa2.bz2 0.8969447522558176
21 aichi_kagosima.bz2 0.9141997783758113
22 aichi_kanagawa.bz2 0.9154662023112237
23 aichi_kouti1.bz2 0.8797547940899089
24 aichi_kouti2.bz2 0.9428526199145164
25 aichi_kumamoto1.bz2 0.8945702073769194
26 aichi_kumamoto2.bz2 0.86290960899161
```

文字割り当てを行い
前処理をくわえたファイル

クラスタリング

3つのクラスタリングを利用
・データの類似度を用いてクラスタリングを行いたいため。
・前年度までの結果との比較。

＜分類する対象の集合＞　　＜分類した結果＞



関連したデータ同士に分類する

NJ法

近隣結合法

・全ての枝の長さの総和が最小になるように系統樹を作っていく。

・出来上がった樹が最良とは限らない。

・効率がいい。

・無根系統樹を作ることができる。
（↑本研究では関係ない）

UPGMA法

平均距離法

・最も近縁なクラスタ間の距離の平均を求めながら系統樹を作成する。

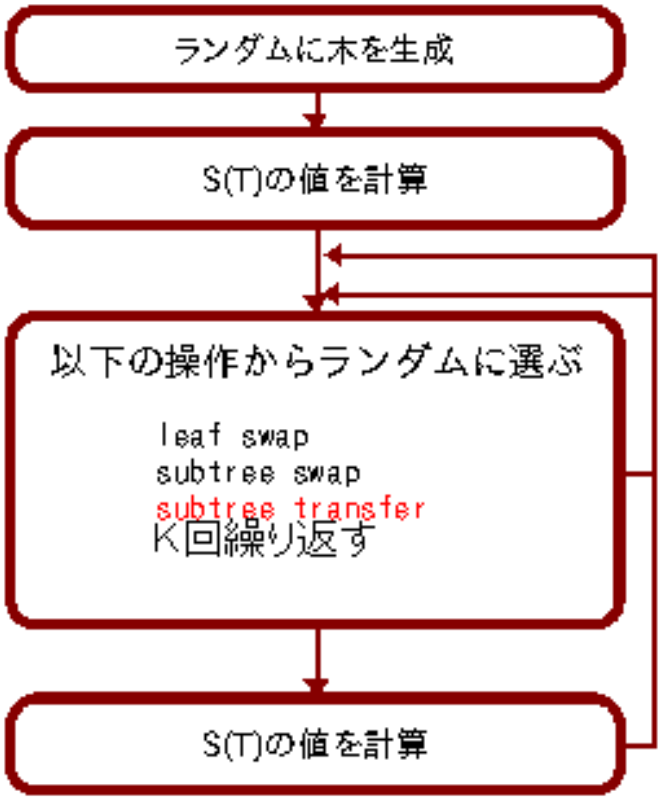
・葉から決まり、最後の根の位置が決まる。

・単純な系統樹作成法（有根系統樹の作成）。

Quartet - method

・木が距離表にどの程度沿っているか評価する基準があり、ランダムに木を変更し評価値を更新していくヒューリスティック（試行錯誤）アルゴリズム。

・NJ法、UPGMA法と比較すると大幅に処理時間を要する。



目次

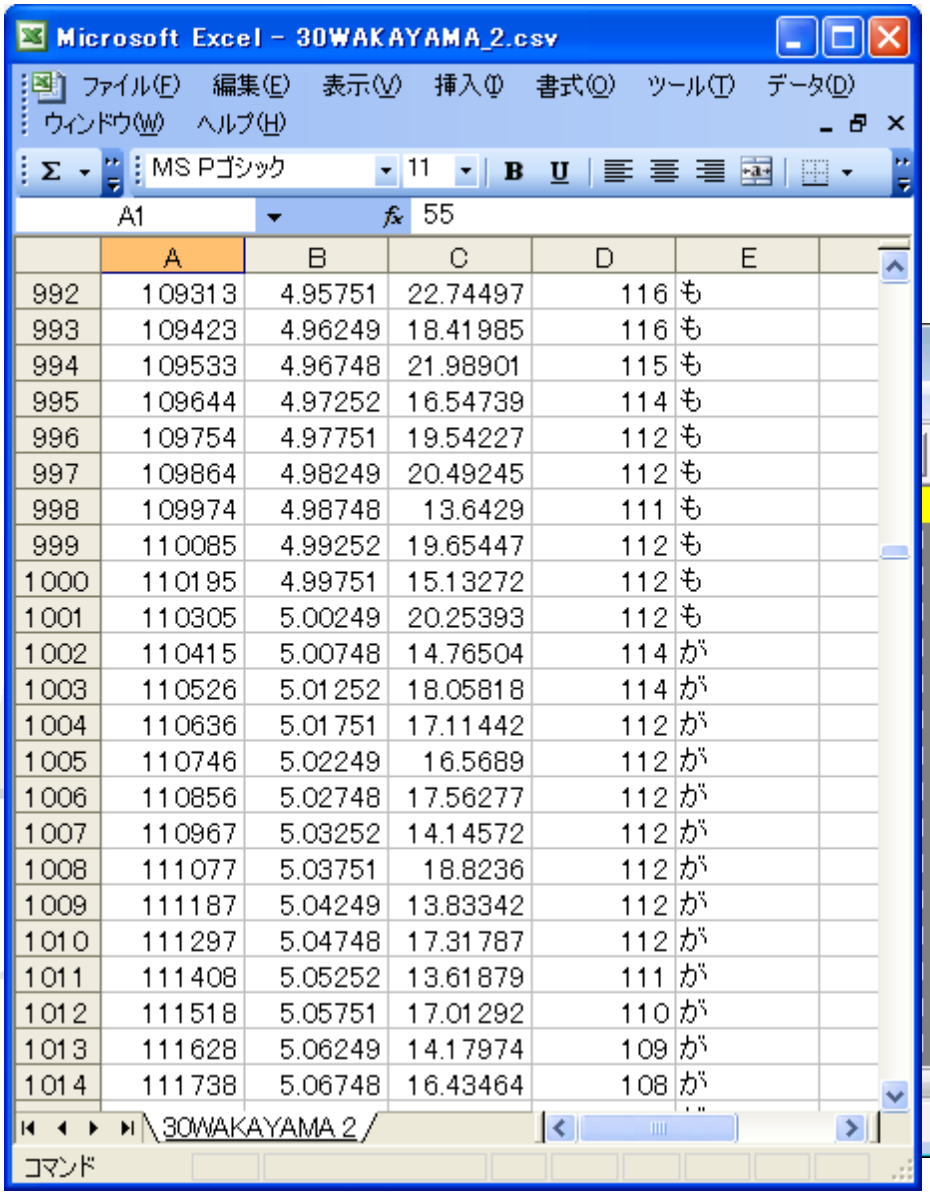
３．研究結果

ノイズカットをした音声データを用い、

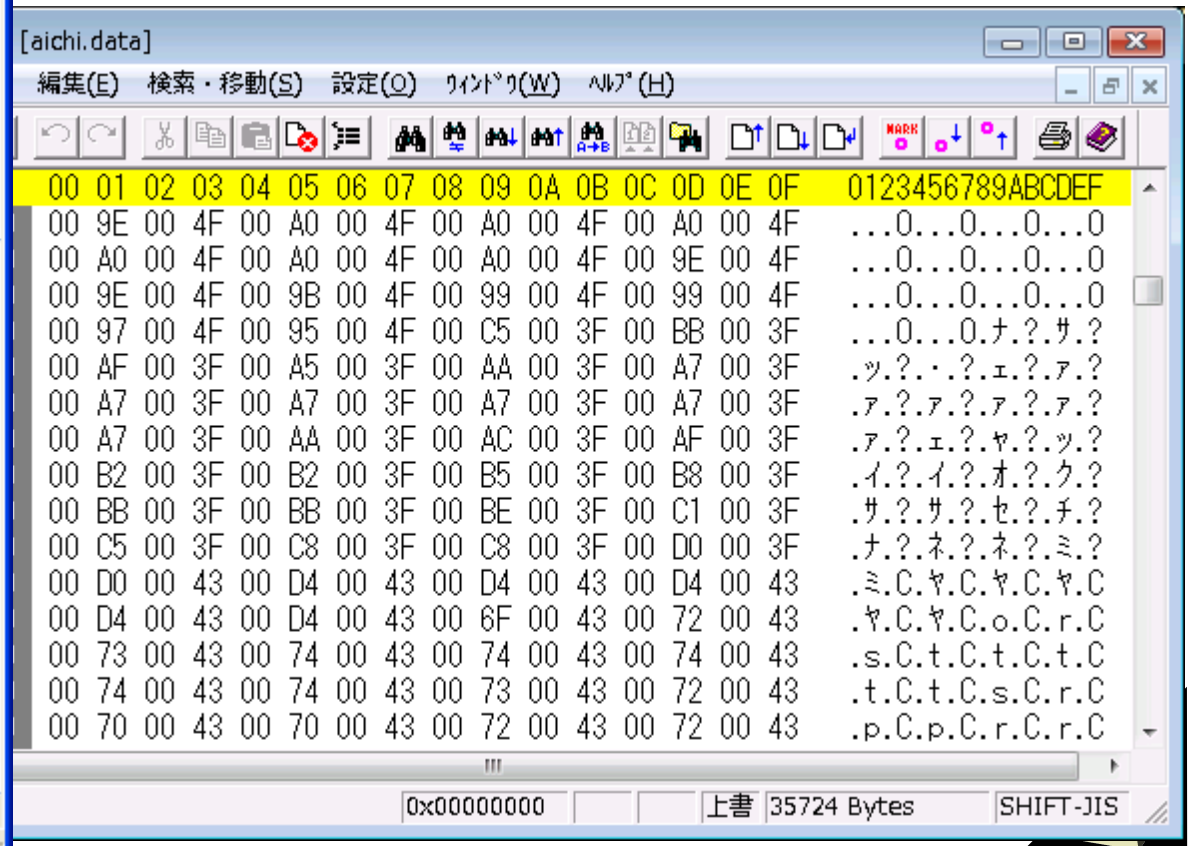
前処理１と**前処理２**で作ったグラフを、
一昨年のグラフと**視覚的**に比較する。

前処理1

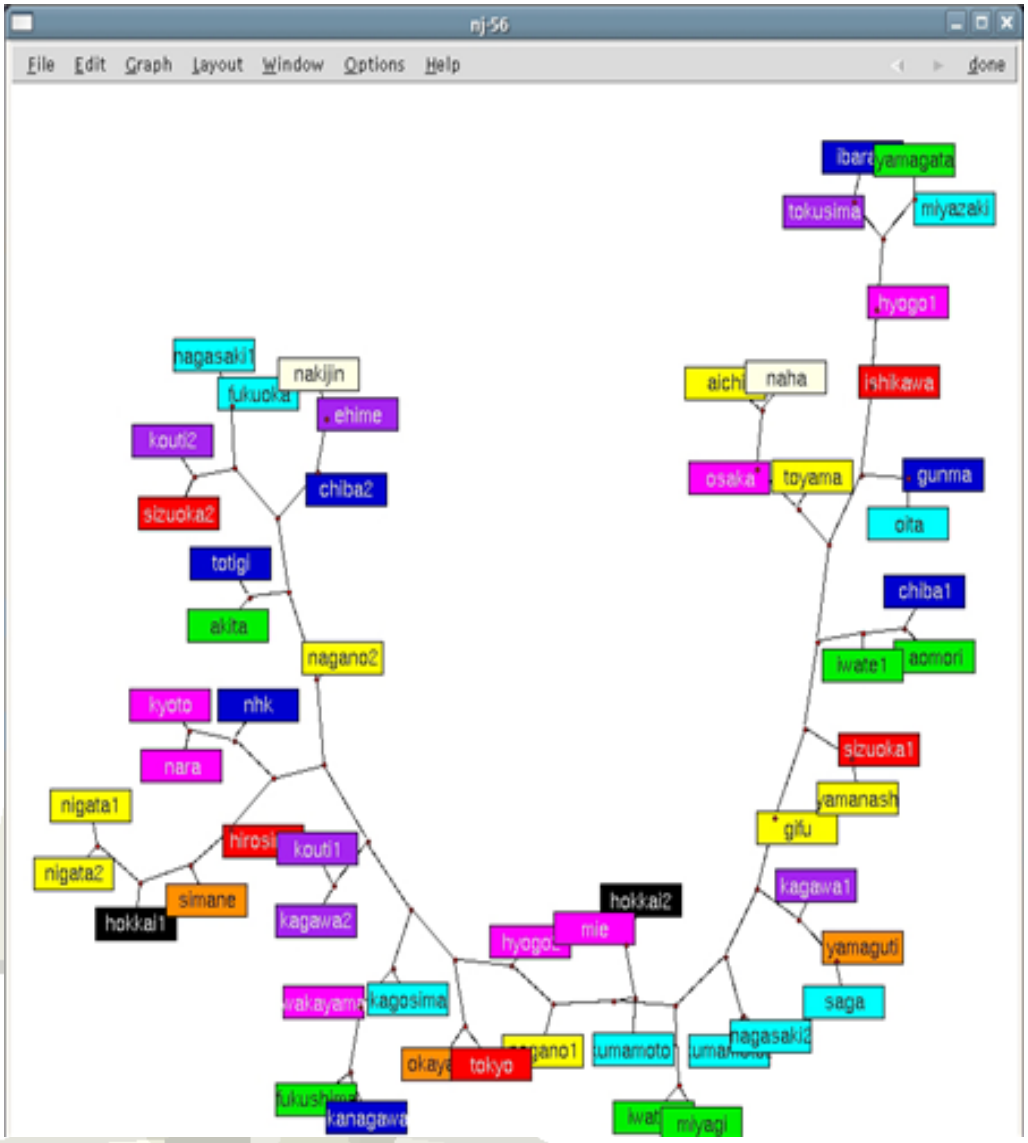
ピッチ　→　値を1byteで出力
文字　　→　数値化して1byteで出力

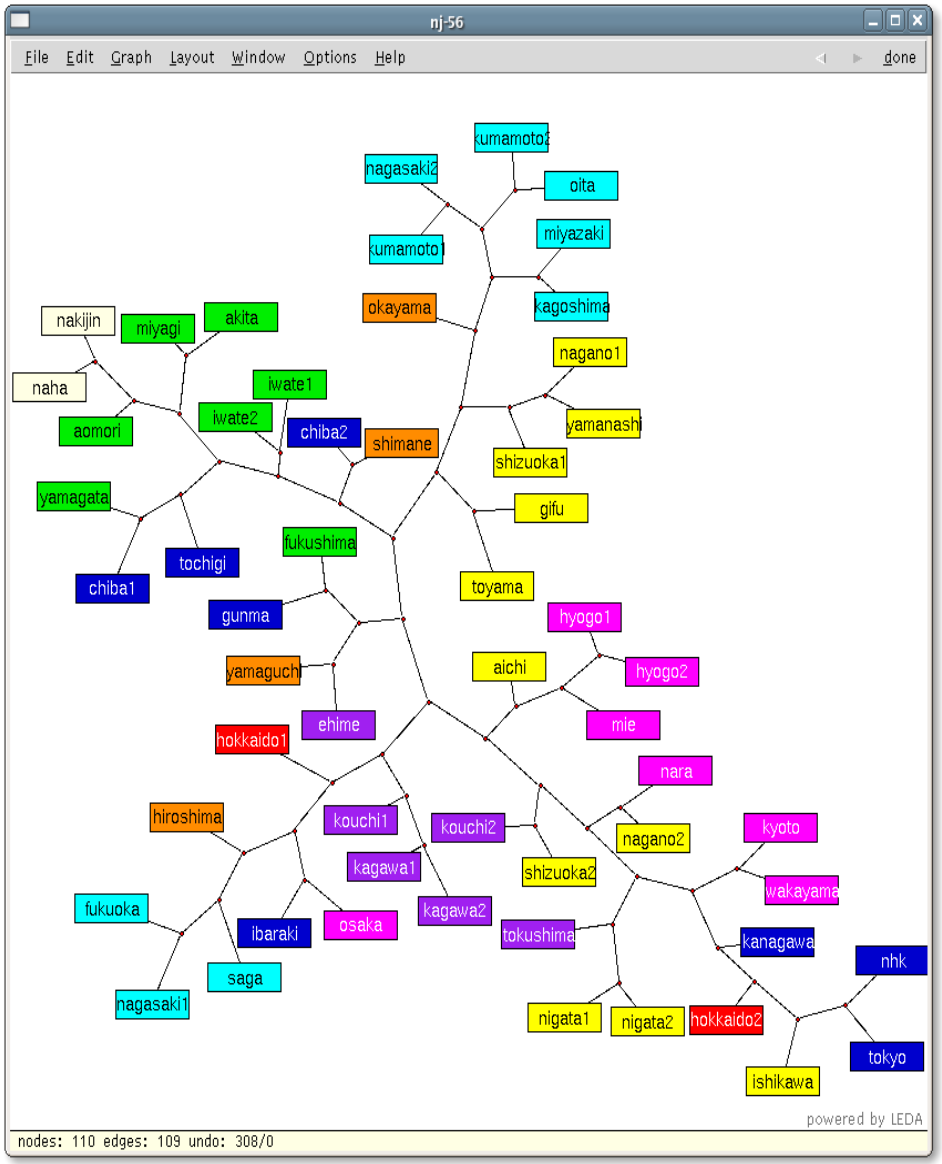


	A	B	C	D	E
992	109313	4.95751	22.74497	116	も
993	109423	4.96249	18.41985	116	も
994	109533	4.96748	21.98901	115	も
995	109644	4.97252	16.54739	114	も
996	109754	4.97751	19.54227	112	も
997	109864	4.98249	20.49245	112	も
998	109974	4.98748	13.6429	111	も
999	110085	4.99252	19.65447	112	も
1000	110195	4.99751	15.13272	112	も
1001	110305	5.00249	20.25393	112	も
1002	110415	5.00748	14.76504	114	が
1003	110526	5.01252	18.05818	114	が
1004	110636	5.01751	17.11442	112	が
1005	110746	5.02249	16.5689	112	が
1006	110856	5.02748	17.56277	112	が
1007	110967	5.03252	14.14572	112	が
1008	111077	5.03751	18.8236	112	が
1009	111187	5.04249	13.83342	112	が
1010	111297	5.04748	17.31787	112	が
1011	111408	5.05252	13.61879	111	が
1012	111518	5.05751	17.01292	110	が
1013	111628	5.06249	14.17974	109	が
1014	111738	5.06748	16.43464	108	が



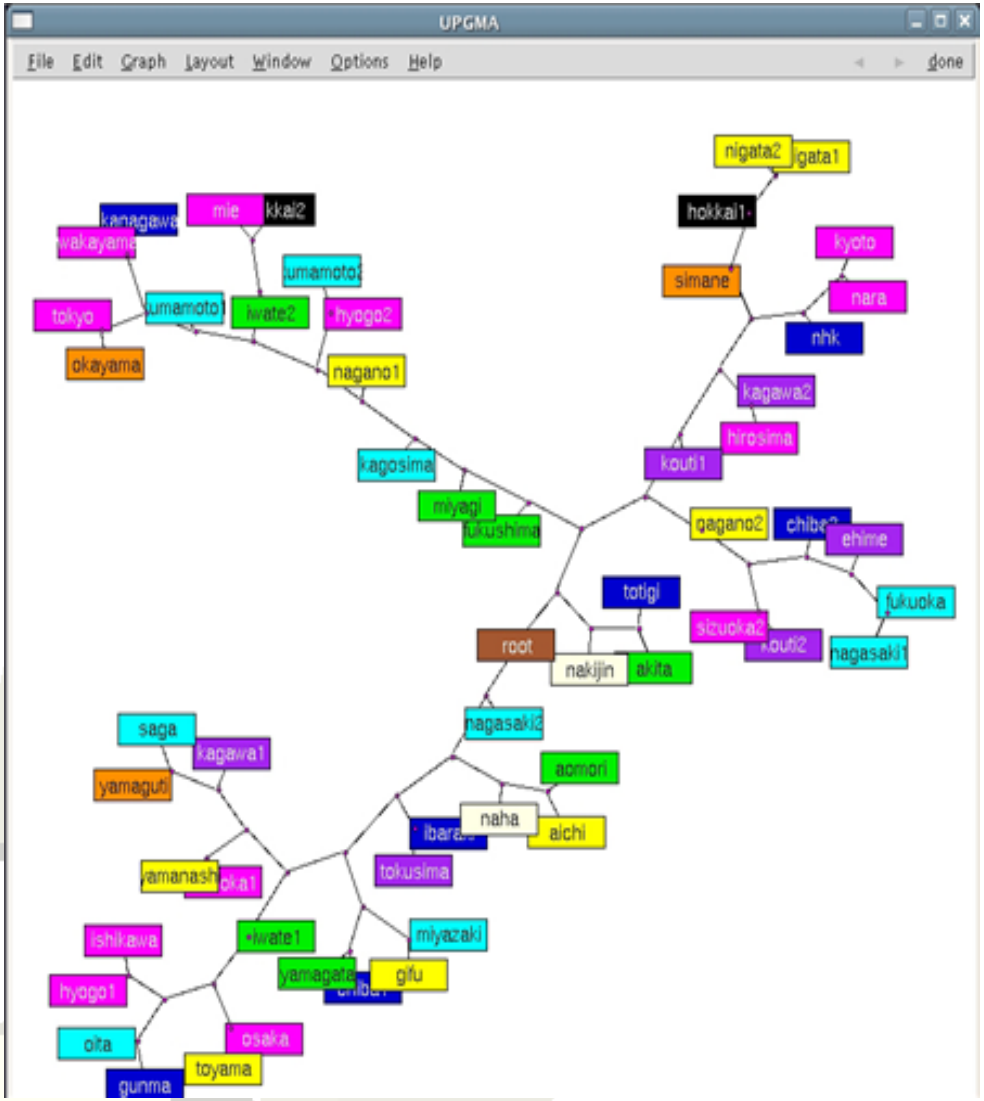
前処理 1　NJ法

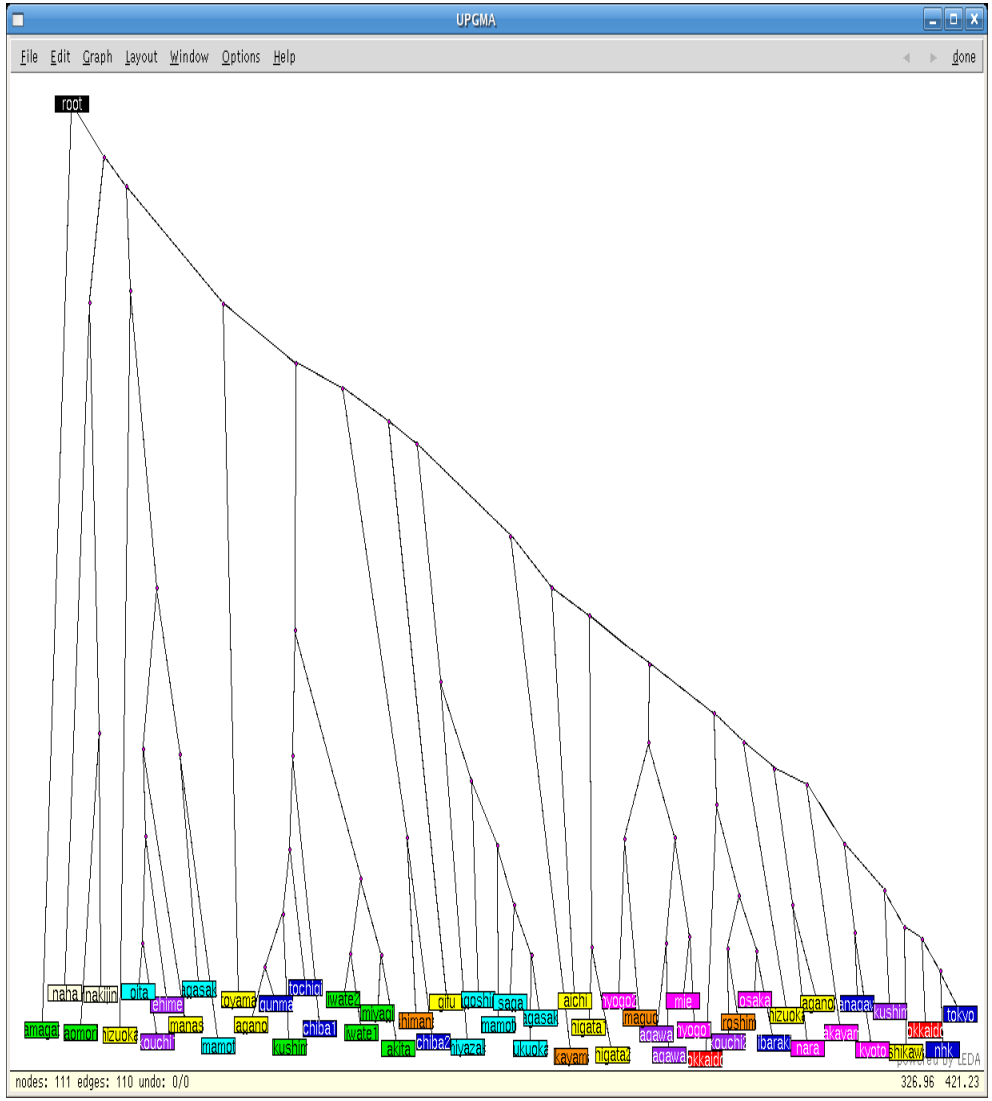




2007年度　NJ法

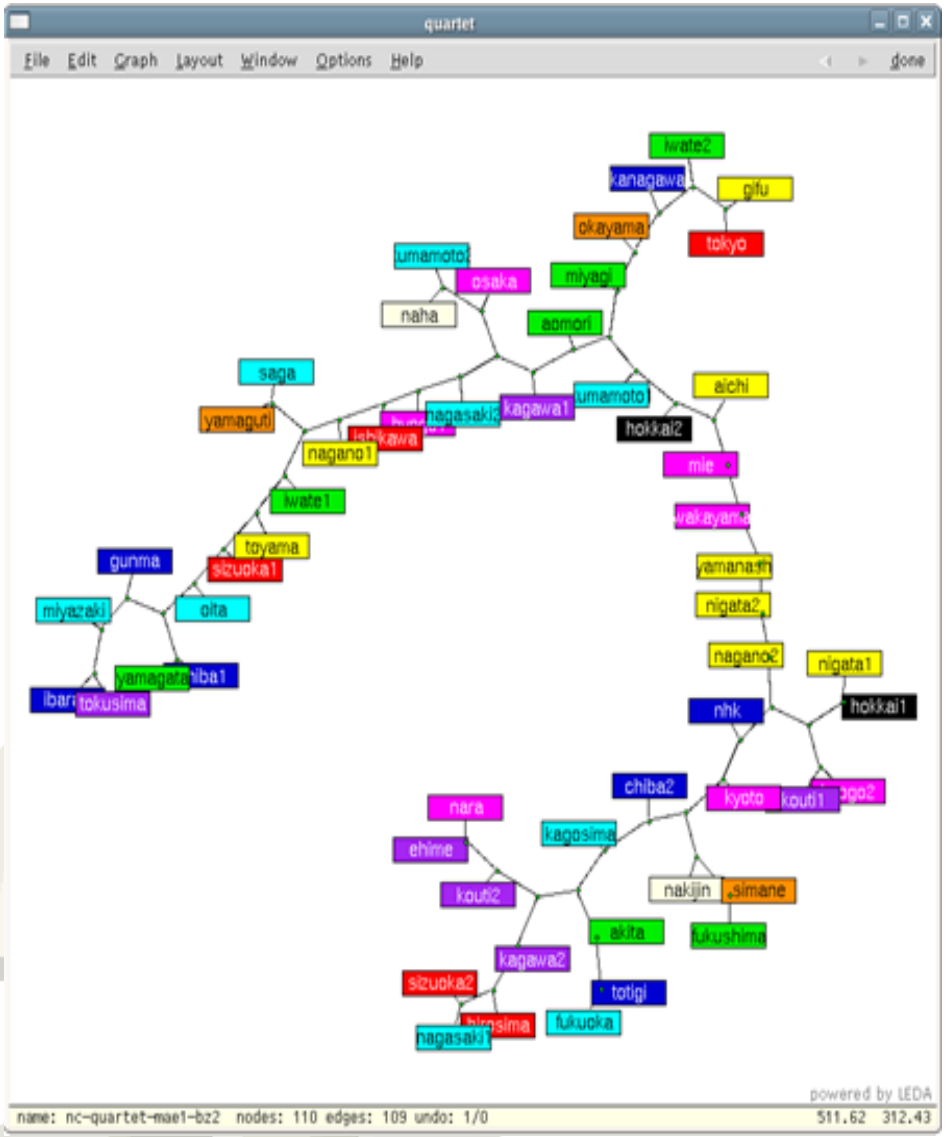
前処理 1　UPGMA法

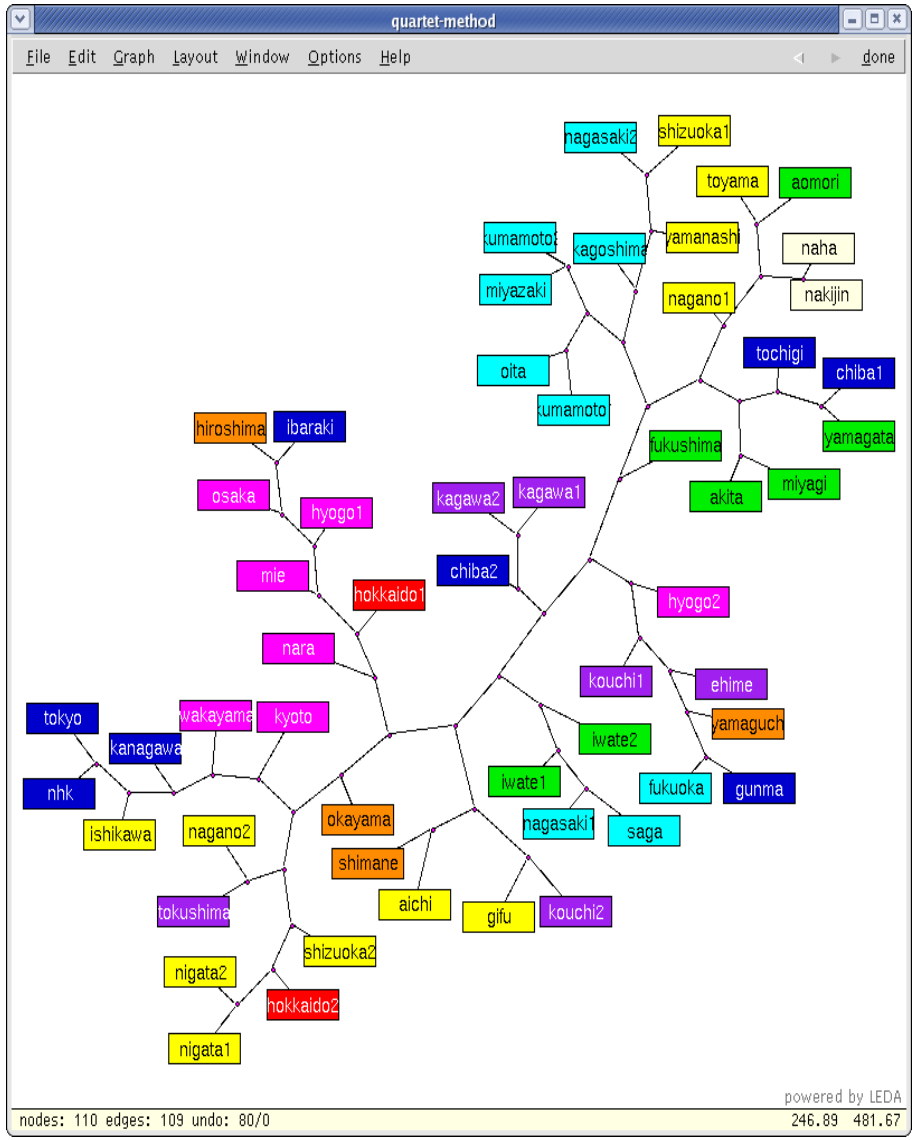




2007年度 UPGMA法

前処理 1　QUARTET-METHOD

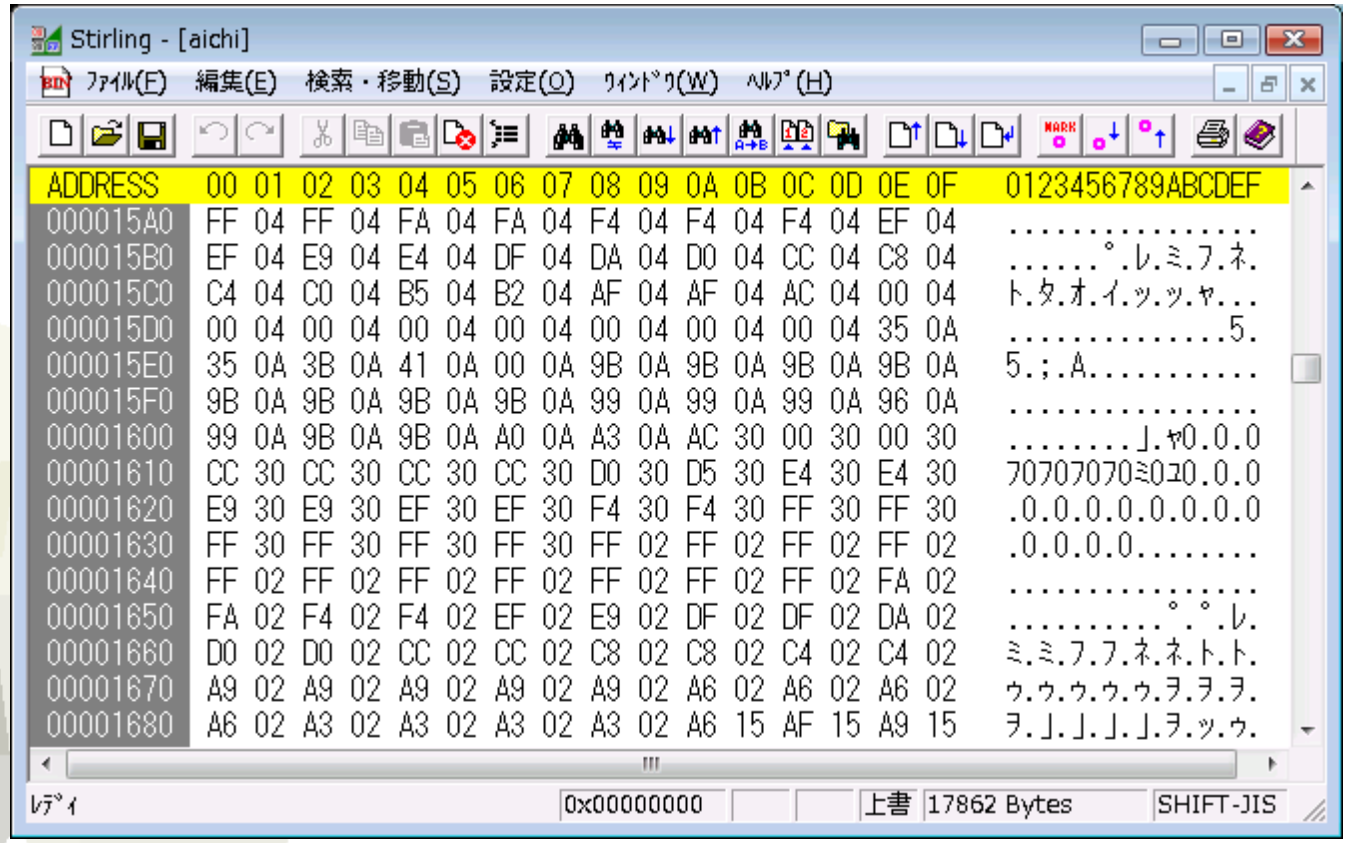




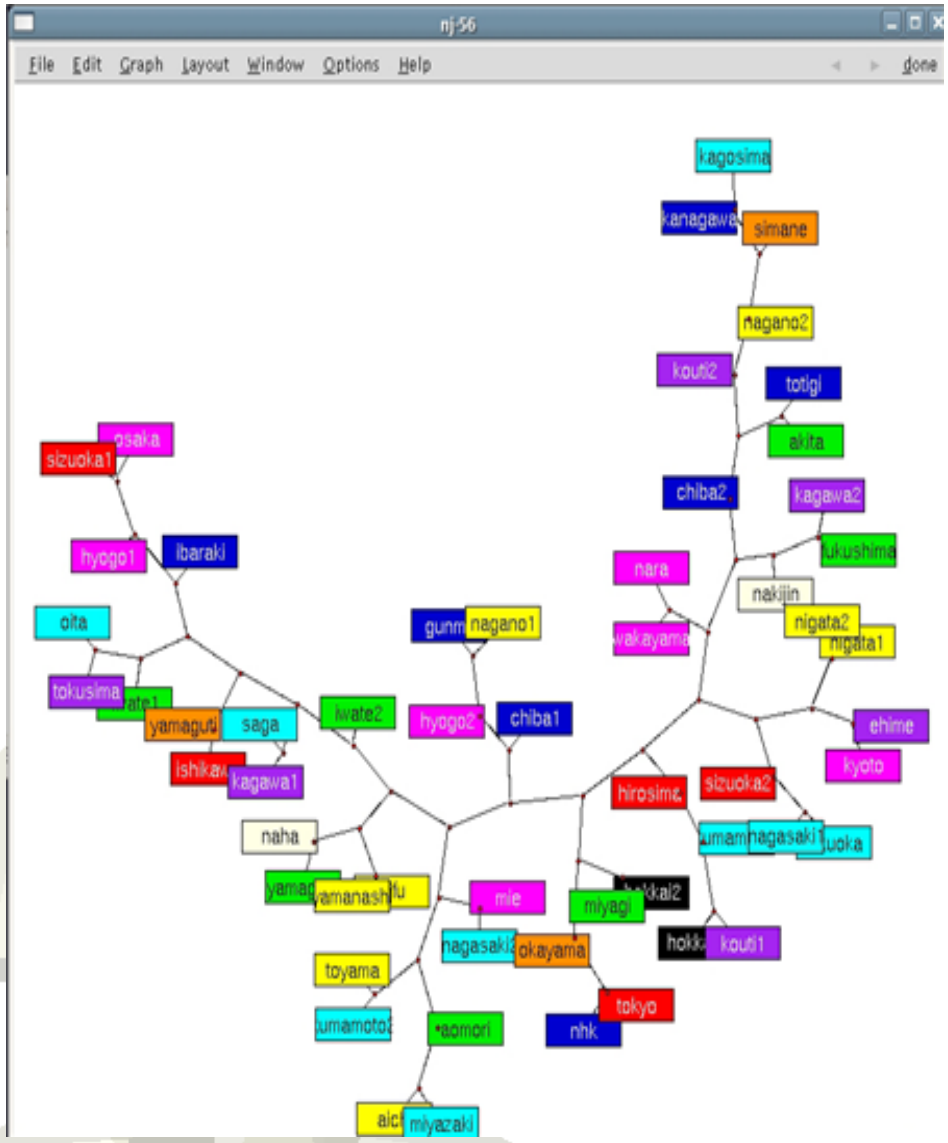
2007年度 quartet-method

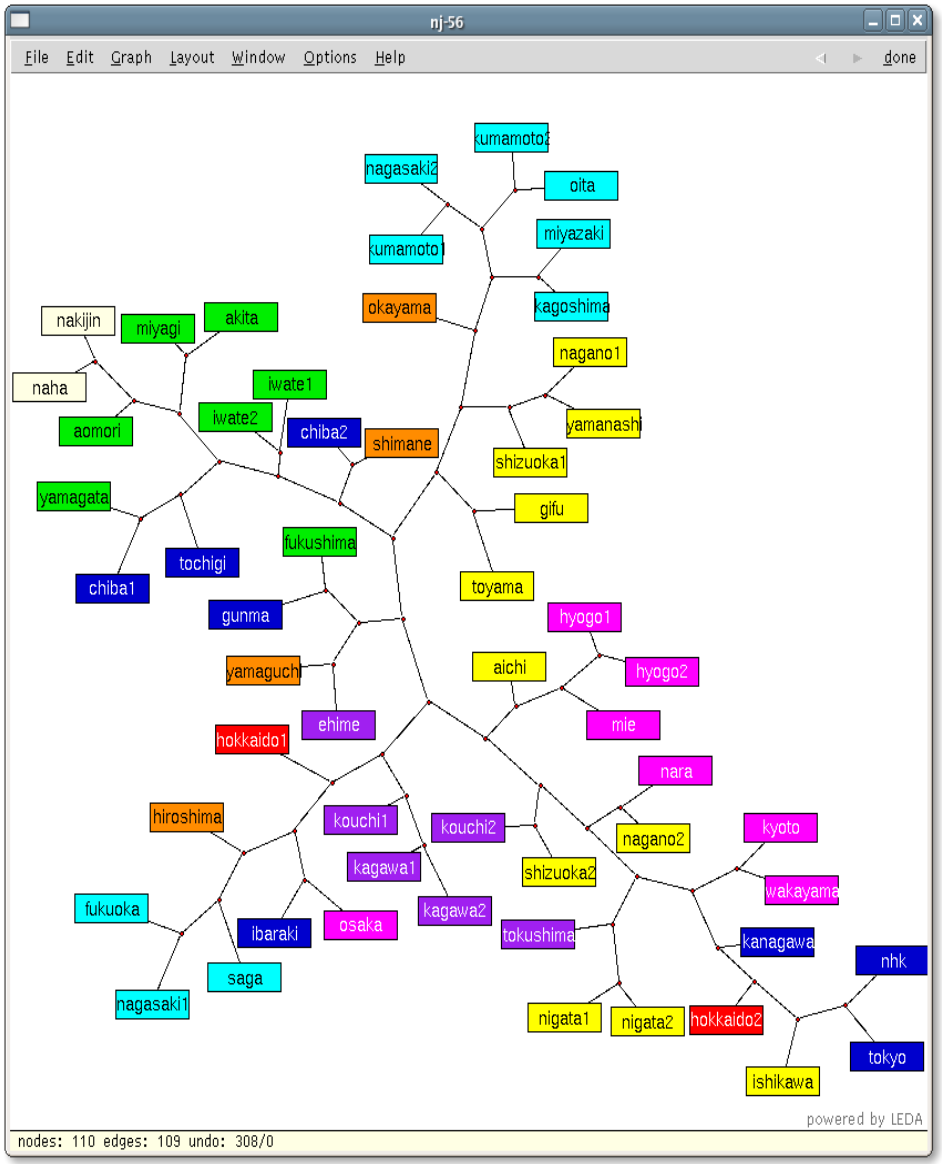
前処理2

ピッチ　→　値の平均を中央値として、その振れ幅を1byteで出力
文字　　→　数値化して1byteで出力



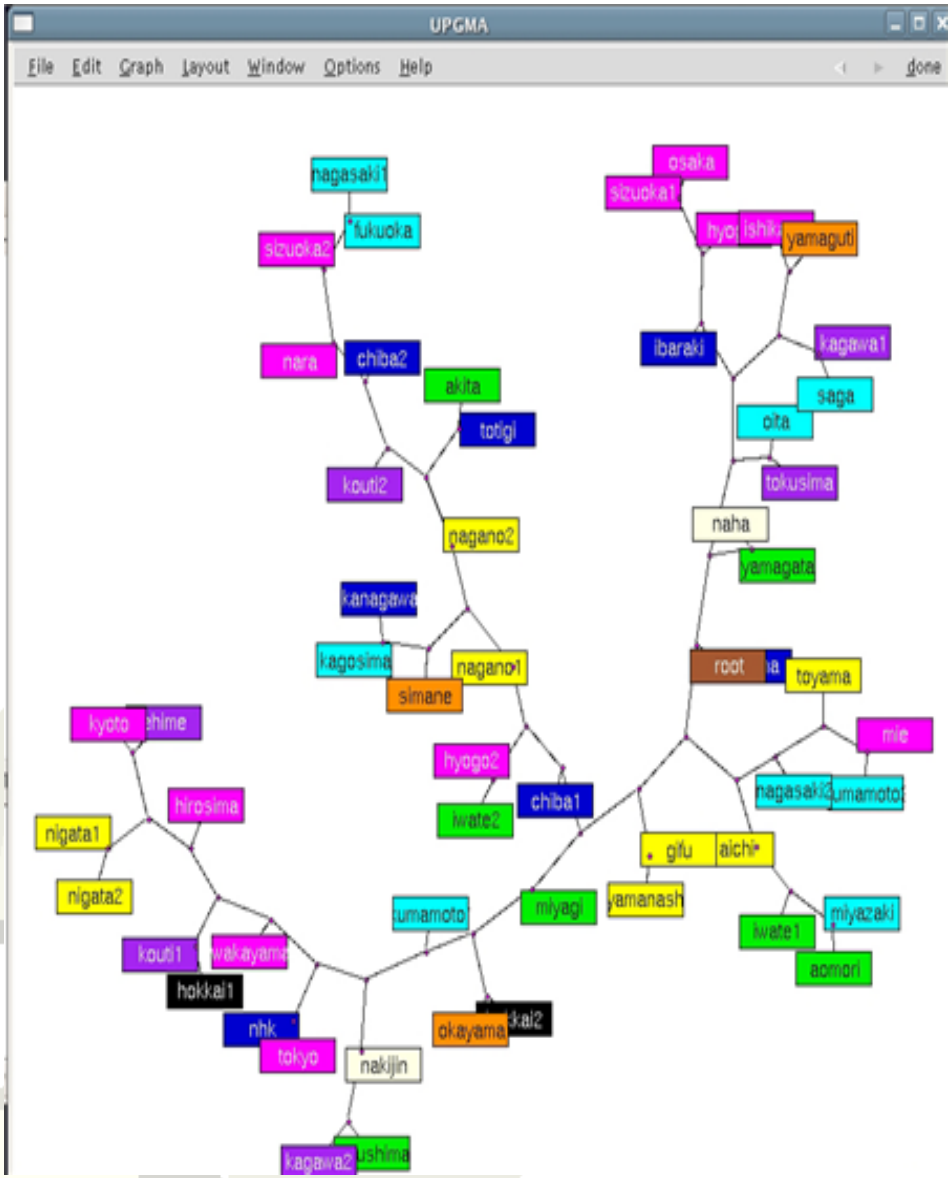
前処理２　NJ法

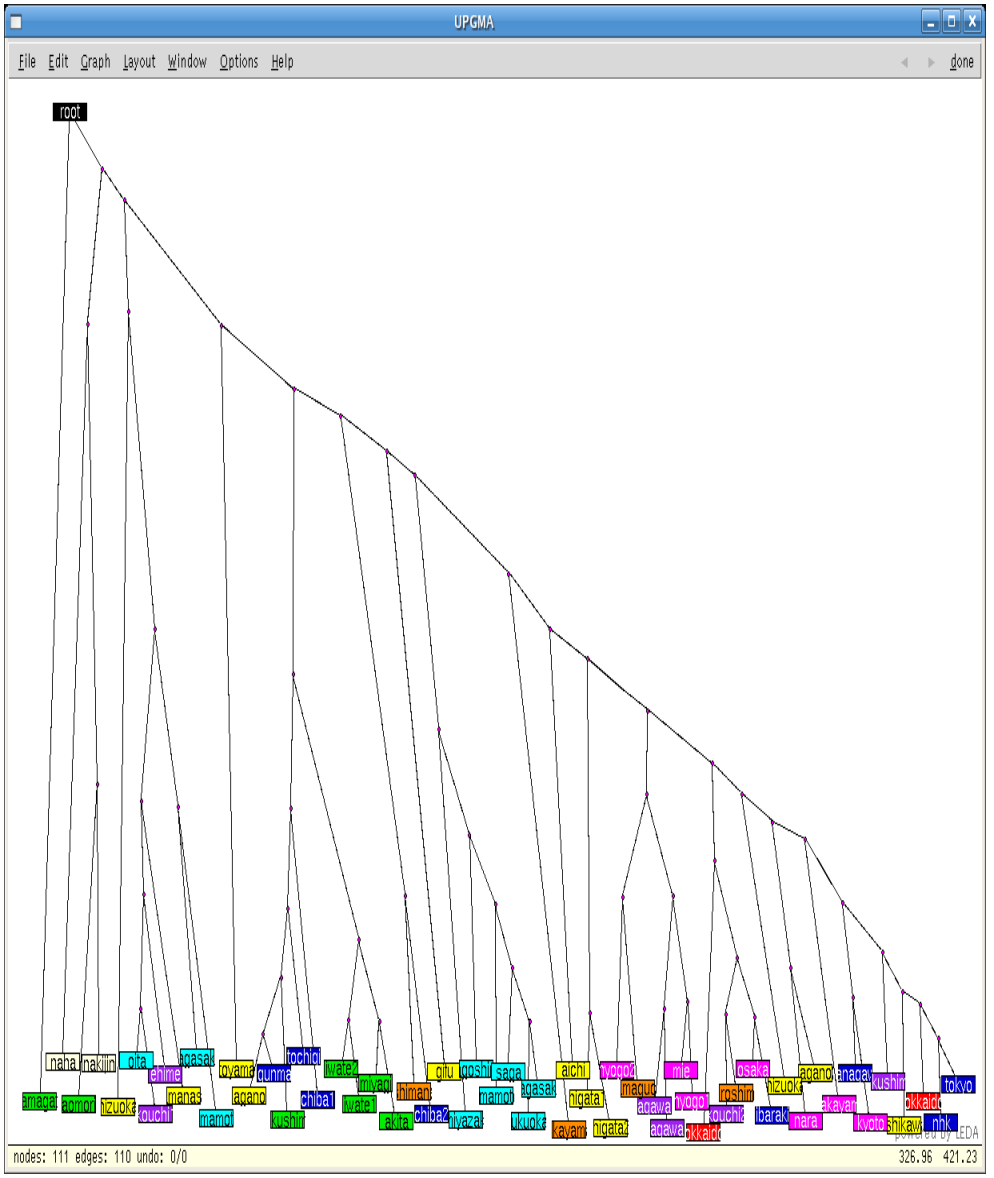




2007年度　NJ法

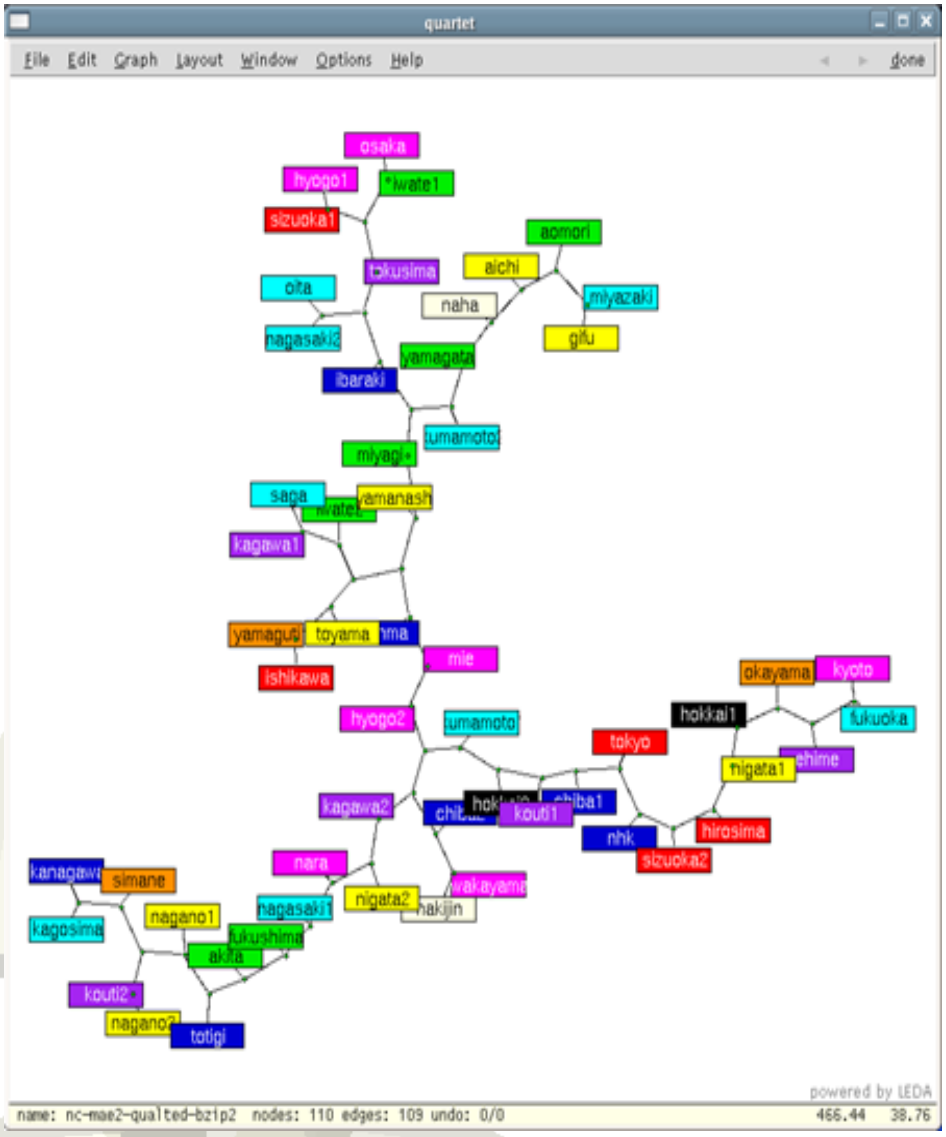
前処理２　UPGMA法

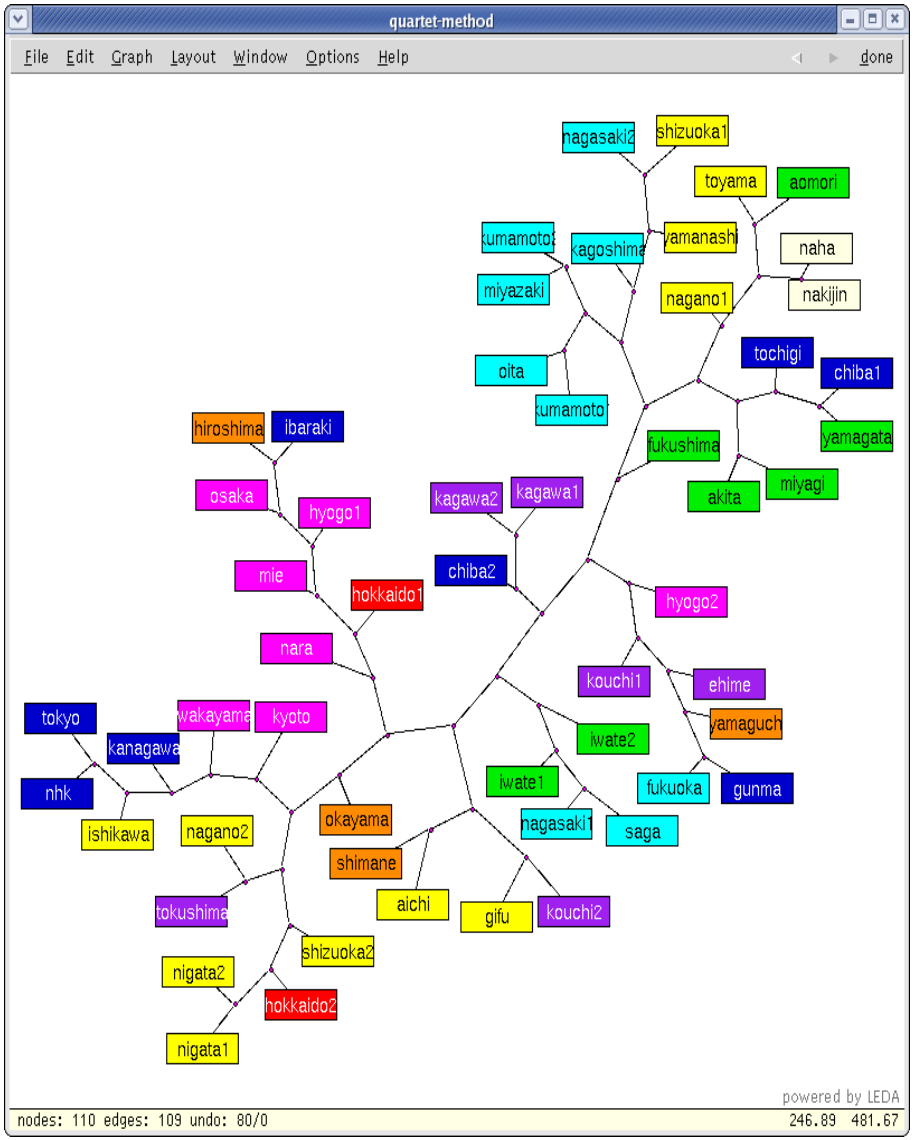




2007年度　UPGMA法

前処理２　QUARTET-METHOD





2007年度 quartet-method

目次

4. 考察、今後の課題

全体的に各地方がばらけてしまった

文字だけでの分類を行った先行研究より悪い結果。

問題点

■ピッチ情報

■音声割り当て

■前処理

方言の分類にピッチの援用が有効か判断できず

4. 考察、今後の課題

全体的に各地方がばらけてしまった

文字だけでの分類を行った先行研究より悪い結果。

問題点

■ピッチ情報

→取得できるピッチ情報が録音環境に左右されやす。

■音声割り当て

■前処理

方言の分類にピッチの援用が有効か判断できず

4. 考察、今後の課題

全体的に各地方がばらけてしまった

文字だけでの分類を行った先行研究より悪い結果。

問題点

■ピッチ情報

→取得できるピッチ情報が録音環境に左右されやす。
→ピッチ情報の組み込み方がよない可能性もあり。

■音声割り当て

■前処理

方言の分類にピッチの援用が有効か判断できず

4. 考察、今後の課題

全体的に各地方がばらけてしまった

文字だけでの分類を行った先行研究より悪い結果。

問題点

■ピッチ情報

→取得できるピッチ情報が録音環境に左右されやす。
→ピッチ情報の組み込み方がよない可能性もあり。

■音声割り当て

→ピッチに割り当てた日本語が不確実。

■前処理

方言の分類にピッチの援用が有効か判断できず

4. 考察、今後の課題

全体的に各地方がばらけてしまった

文字だけでの分類を行った先行研究より悪い結果。

問題点

■ピッチ情報

→取得できるピッチ情報が録音環境に左右されやす。
→ピッチ情報の組み込み方がよない可能性もあり。

■音声割り当て

→ピッチに割り当てた日本語が不確実。

■前処理

→有用なものを一つしか実装できなかった。

方言の分類にピッチの援用が有効か判断できず

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。

・音声の長さを一定にする。

・音声の男女差、声質差などの違いを考慮する。

・ピッチ以外の音声情報の抽出・適用。

・適切な前処理・実験データの作成方法を考案する。

・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。

・音声の男女差、声質差などの違いを考慮する。

・ピッチ以外の音声情報の抽出・適用。

・適切な前処理・実験データの作成方法を考案する。

・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。

・ピッチ以外の音声情報の抽出・適用。

・適切な前処理・実験データの作成方法を考案する。

・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。
　特に男女差はピッチへの影響が大きいため。
・ピッチ以外の音声情報の抽出・適用。

・適切な前処理・実験データの作成方法を考案する。

・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。
　特に男女差はピッチへの影響が大きいため。
・ピッチ以外の音声情報の抽出・適用。
　ピッチ以外を抽出する、
　音声ファイル自体を比べるなど。
・適切な前処理・実験データの作成方法を考案する。

・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。
　特に男女差はピッチへの影響が大きいため。
・ピッチ以外の音声情報の抽出・適用。
　ピッチ以外を抽出する、
　音声ファイル自体を比べるなど。
・適切な前処理・実験データの作成方法を考案する。
　特に前処理の考案。
・自動的に音声から日本語割り当てを行う。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。
　特に男女差はピッチへの影響が大きいため。
・ピッチ以外の音声情報の抽出・適用。
　ピッチ以外を抽出する、
　音声ファイル自体を比べるなど。
・適切な前処理・実験データの作成方法を考案する。
　特に前処理の考案。
・自動的に音声から日本語割り当てを行う。
　日本語を手で割り当てず、自動化する。

どういう図が正しいのかを調べる。

4. 考察、今後の課題

・最適なピッチ情報を取得できるような音声処理。
　ノイズカットなど音声自体に手を加える。
・音声の長さを一定にする。
　音声ファイルの長さがバラバラなので。
・音声の男女差、声質差などの違いを考慮する。
　特に男女差はピッチへの影響が大きいため。
・ピッチ以外の音声情報の抽出・適用。
　ピッチ以外を抽出する、
　音声ファイル自体を比べるなど。
・適切な前処理・実験データの作成方法を考案する。
　特に前処理の考案。
・自動的に音声から日本語割り当てを行う。
　日本語を手で割り当てず、自動化する。

どういう図が正しいのかを調べる。
・人間の声についての知識を深める。
・各地方の歴史と方言のかかわり方。

終わり

ご清聴ありがとうございました。

Similarity metric

Ming Liらの研究で**Kolmogorov記述量に基づく**類似度距離が提案された

$K(y) \geq K(x)$ のとき

$$d(x,y) = \frac{K(xy) - K(x)}{K(y)}$$

Kolmogorov 記述量　　$K(x)$

データxを生成するための最小のファイルサイズ

Similarity metric

Ming Liらの研究でKolmogorov記述量に基づく類似度距離が提案された

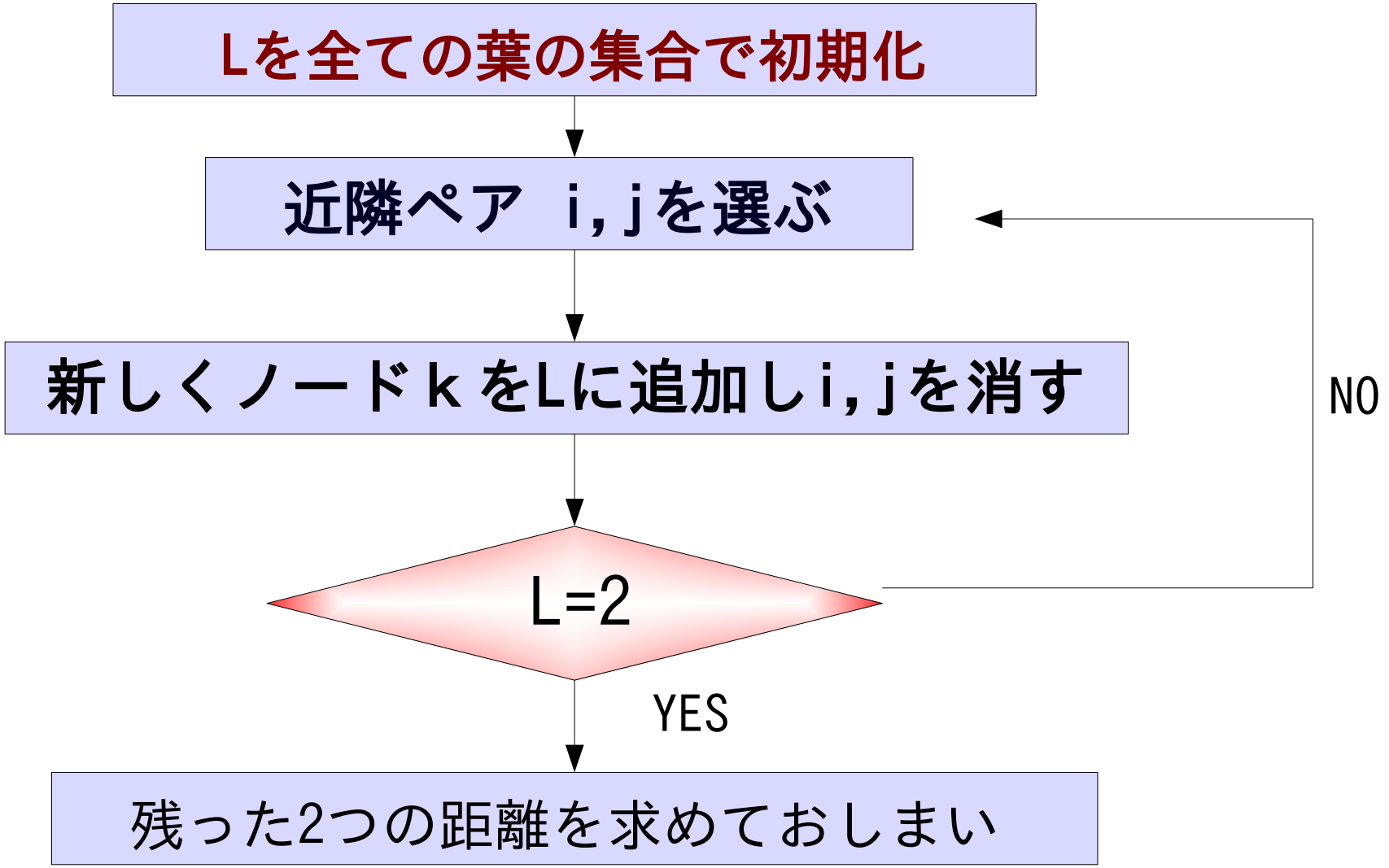
(1)類似度距離d(x,y)は文字列x,yによって得られる
(2)Kolmogorov記述量は帰納的でないため計算不可能
(3)圧縮プログラムを利用し、圧縮後のサイズをKolmogorov記述量の近似値とする。bzip2が有効

$$d(x,y)=\frac{K(xy)-K(x)}{K(y)}$$

$$d(x,y) \approx \frac{bzip2(xy)-bzip2(x)}{bzip2(y)}$$

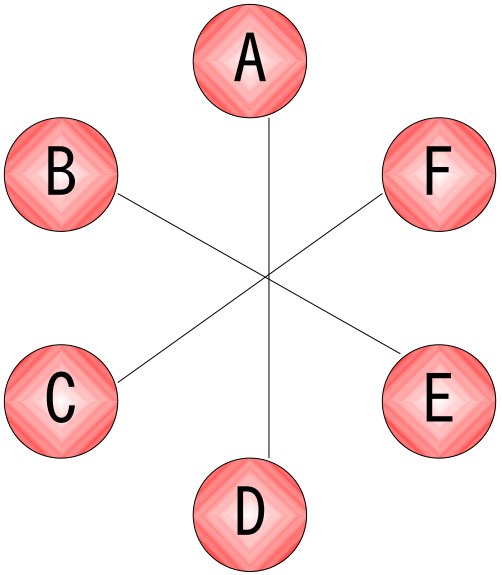
Nj法

NJ法(近隣結合法)


Lを全ての葉の集合で初期化
近隣ペア i, j を選ぶ
新しくノードkをLに追加しi, jを消す
L=2
NO
YES
残った2つの距離を求めておしまい

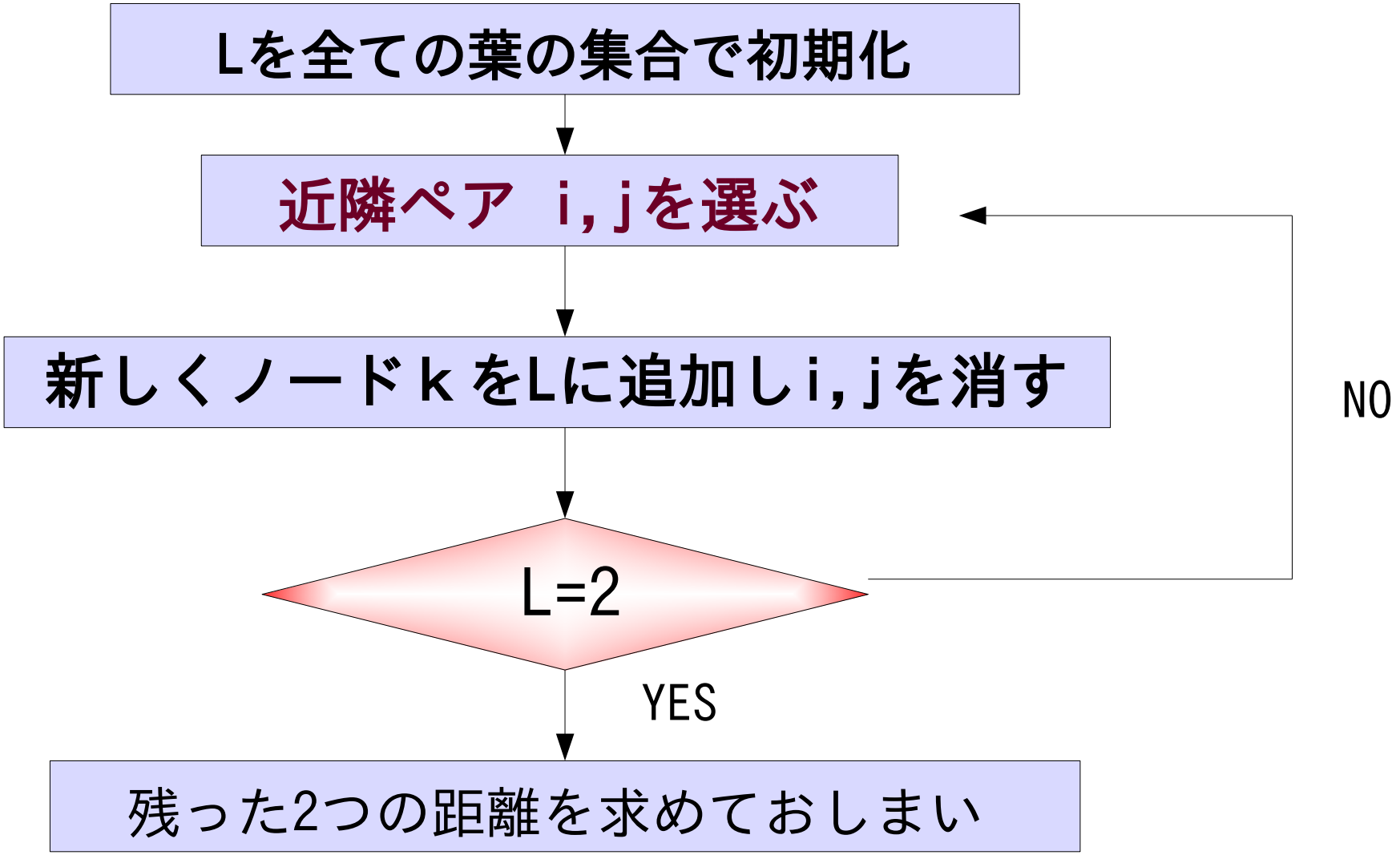
Lを全ての葉の集合で初期化

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						



L={A.B.C.D.E.F}

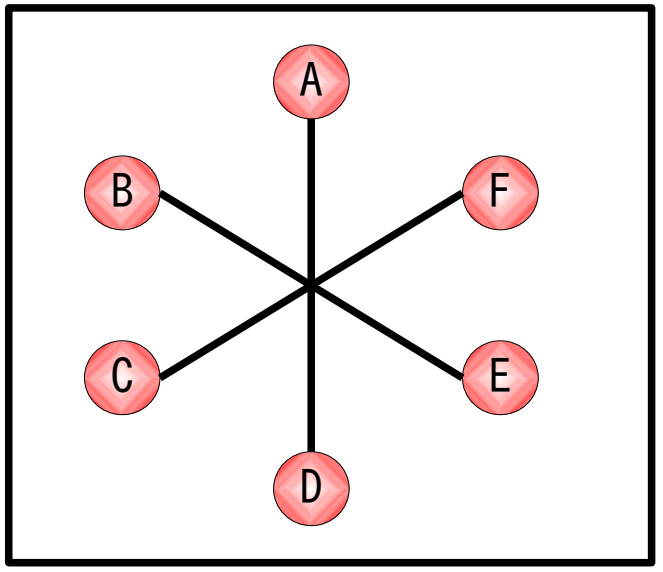
NJ法(近隣結合法)


Lを全ての葉の集合で初期化
近隣ペア i, jを選ぶ
新しくノードkをLに追加しi, jを消す
L=2
NO
YES
残った2つの距離を求めておしまい

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						

$$r_i = \frac{1}{|L|-2} \sum_{k \in L} d_{ik}$$



$$r_A = \frac{1}{|6|-2}(5+4+7+6+8) = 7.5$$

$L = \{A, B, C, D, E, F\}$

$r_A = 7.5$　　$r_B = 10.5$

$r_C = 8.0$　　$r_D = 9.5$

$r_E = 8.5$　　$r_F = 11$

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						

$$r_A = 7.5 \qquad r_B = 10.5$$

$$r_C = 8.0 \qquad r_D = 9.5$$

$$r_E = 8.5 \qquad r_F = 11$$

$$D_{ij} = d_{ij} - (r_i + r_j)$$

D(i , j)	A	B	C	D	E	F
A		-13				
B						
C						
D						
E						
F						

$$D_{AB} = d_{AB} - (r_A + r_B) = 5 - (7.5 + 10.5)$$

$$= -13$$

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						

$$r_A = 7.5 \qquad r_B = 10.5$$

$$r_C = 8.0 \qquad r_D = 9.5$$

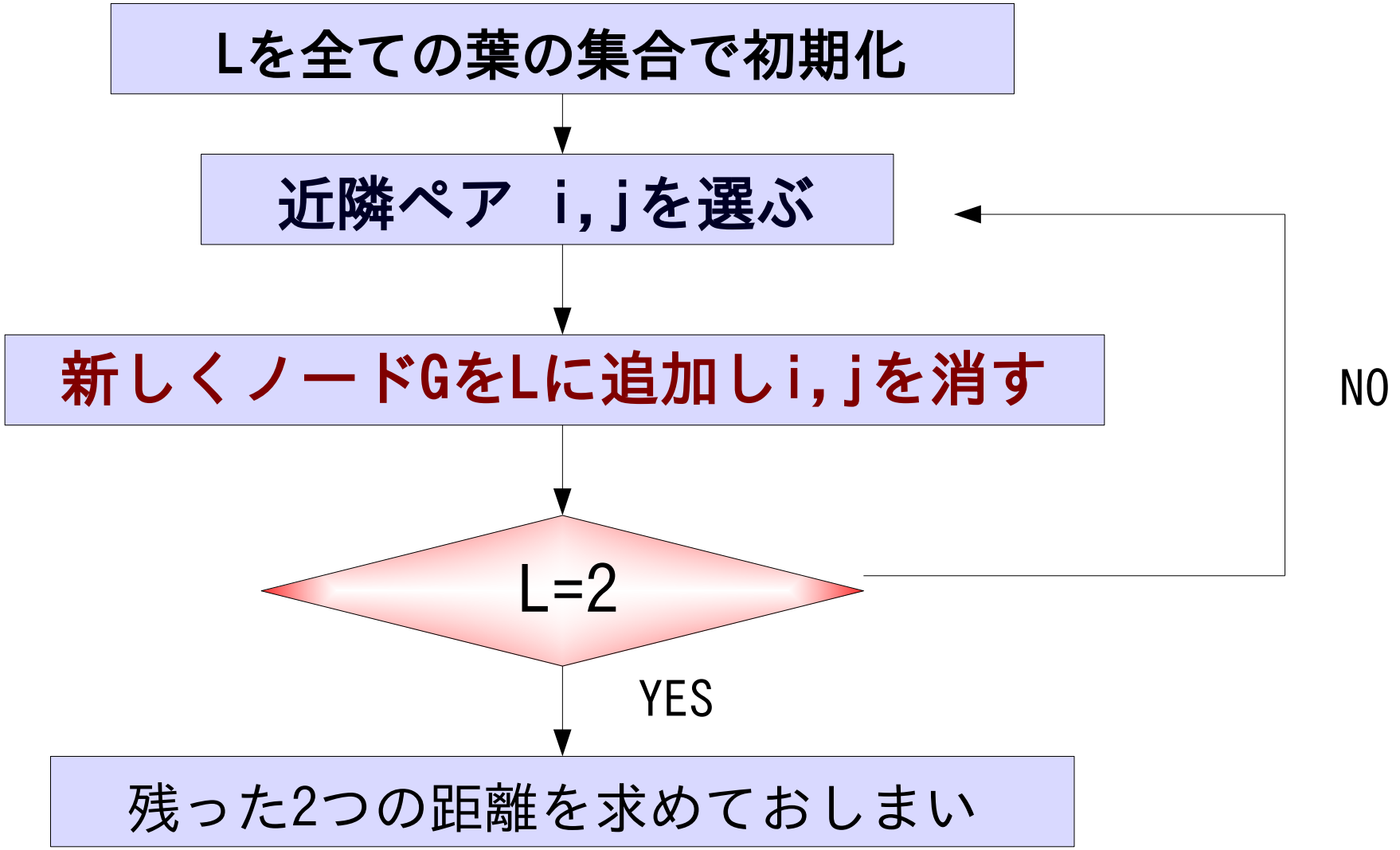
$$r_E = 8.5 \qquad r_F = 11$$

$$D_{ij} = d_{ij} - (r_i + r_j)$$

D(i,j)	A	B	C	D	E	F
A		-13	-11.5	-10	-10	-10.5
B			-11.5	-10	-10	-10.5
C				-10.5	-10.5	-11
D					-13	-11.5
E						-11.5
F						

$$D_{AB} = d_{AB} - (r_A + r_B) = 5 - (7.5 + 10.5)$$
$$= -13$$

NJ法(近隣結合法)

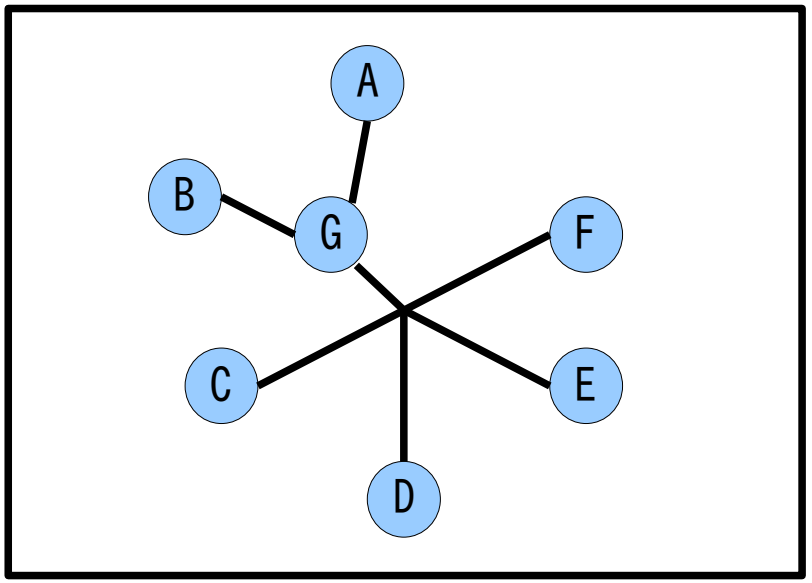

Lを全ての葉の集合で初期化
近隣ペア i,jを選ぶ
新しくノードGをLに追加しi,jを消す
L=2
NO
YES
残った2つの距離を求めておしまい

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						

new node G

$$L = \{A, B, C, D, E, F, G\}$$



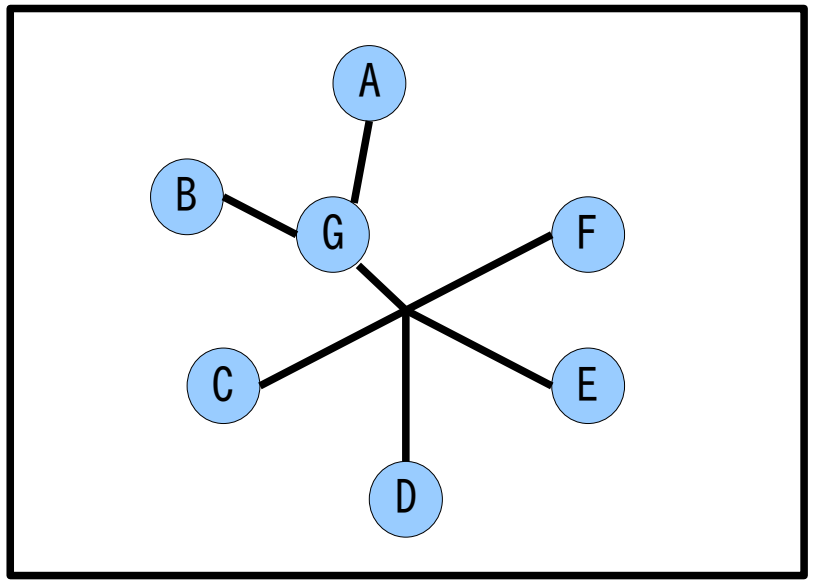
NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						

new node G

i = A
j = B

$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$$

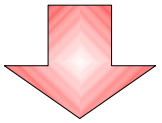


$$d_{GC} = \frac{1}{2}(d_{AC} + d_{BC} - d_{AB})$$

$$= \frac{1}{2}(4 + 7 - 5) = 3$$

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F
A		5	4	7	6	8
B			7	10	9	11
C				7	6	8
D					5	9
E						8
F						



d(i,j)	A	B	C	D	E	F	G
A		5	4	7	6	8	
B			7	10	9	11	
C				7	6	8	3
D					5	9	6
E						8	5
F							7
G							

$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$$

i = A　　j = B

$$d_{GC} = \frac{1}{2}(d_{AC} + d_{BC} - d_{AB}) = 3$$

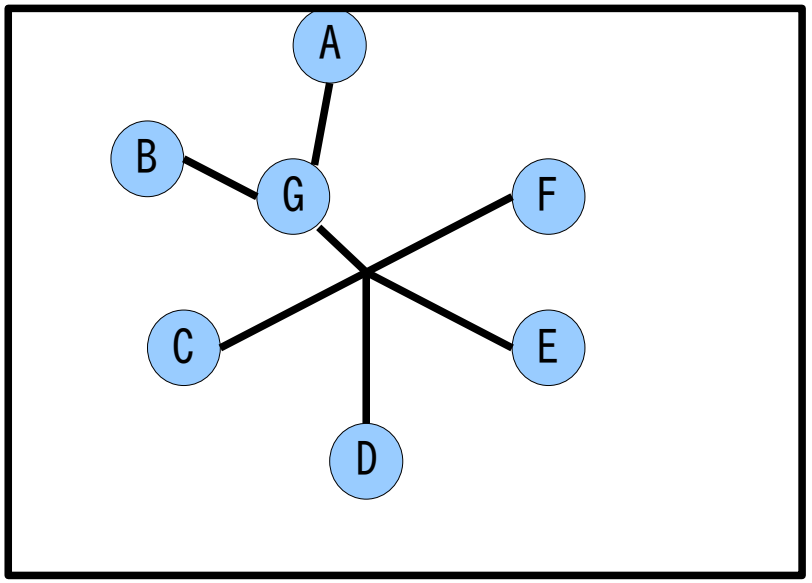
$$d_{GD} = \frac{1}{2}(d_{AD} + d_{BD} - d_{AB}) = 6$$

$$d_{GE} = \frac{1}{2}(d_{AE} + d_{BE} - d_{AB}) = 5$$

$$d_{GF} = \frac{1}{2}(d_{AF} + d_{BF} - d_{AB}) = 7$$

NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F	G
A		5	4	7	6	8	1
B			7	10	9	11	4
C				7	6	8	3
D					5	9	6
E						8	5
F							7
G							



$$d_{ik} = \frac{1}{2}(d_{ij} + r_i - r_j)$$

$$d_{jk} = d_{ij} - d_{ik}$$

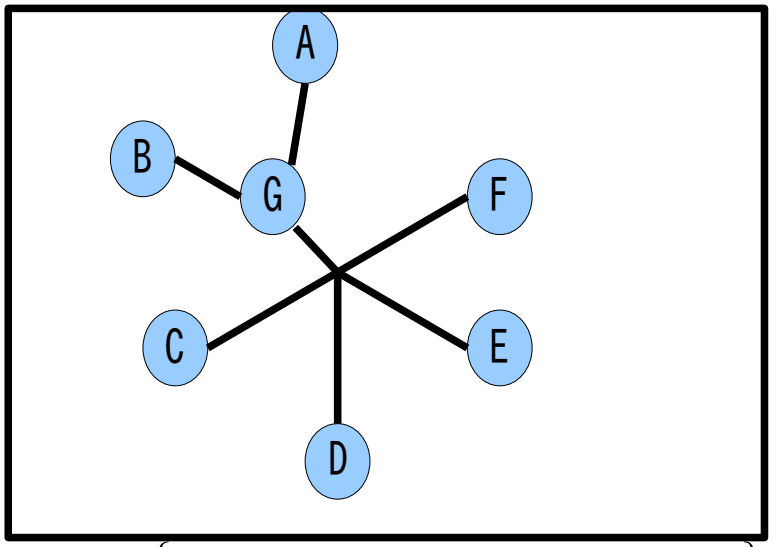
$$d_{AG} = \frac{1}{2}(d_{AB} + r_A - r_B) = 1$$

$$d_{BG} = d_{AB} - d_{AG} = 4$$

$$r_A = 7.5 \qquad r_B = 10.5$$
$$r_C = 8.0 \qquad r_D = 9.5$$
$$r_E = 8.5 \qquad r_F = 11$$

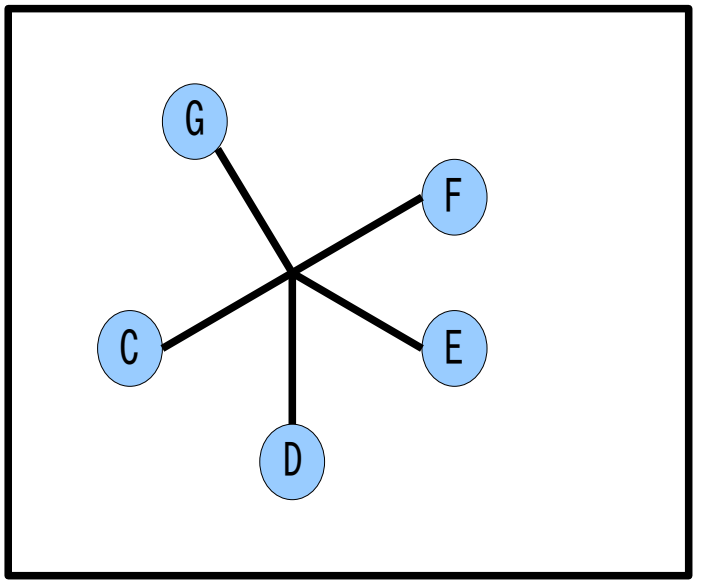
NJ法(近隣結合法) アルゴリズム

d(i,j)	A	B	C	D	E	F	G
A		5	4	7	6	8	1
B			7	10	9	11	4
C				7	6	8	3
D					5	9	6
E						8	5
F							7
G							



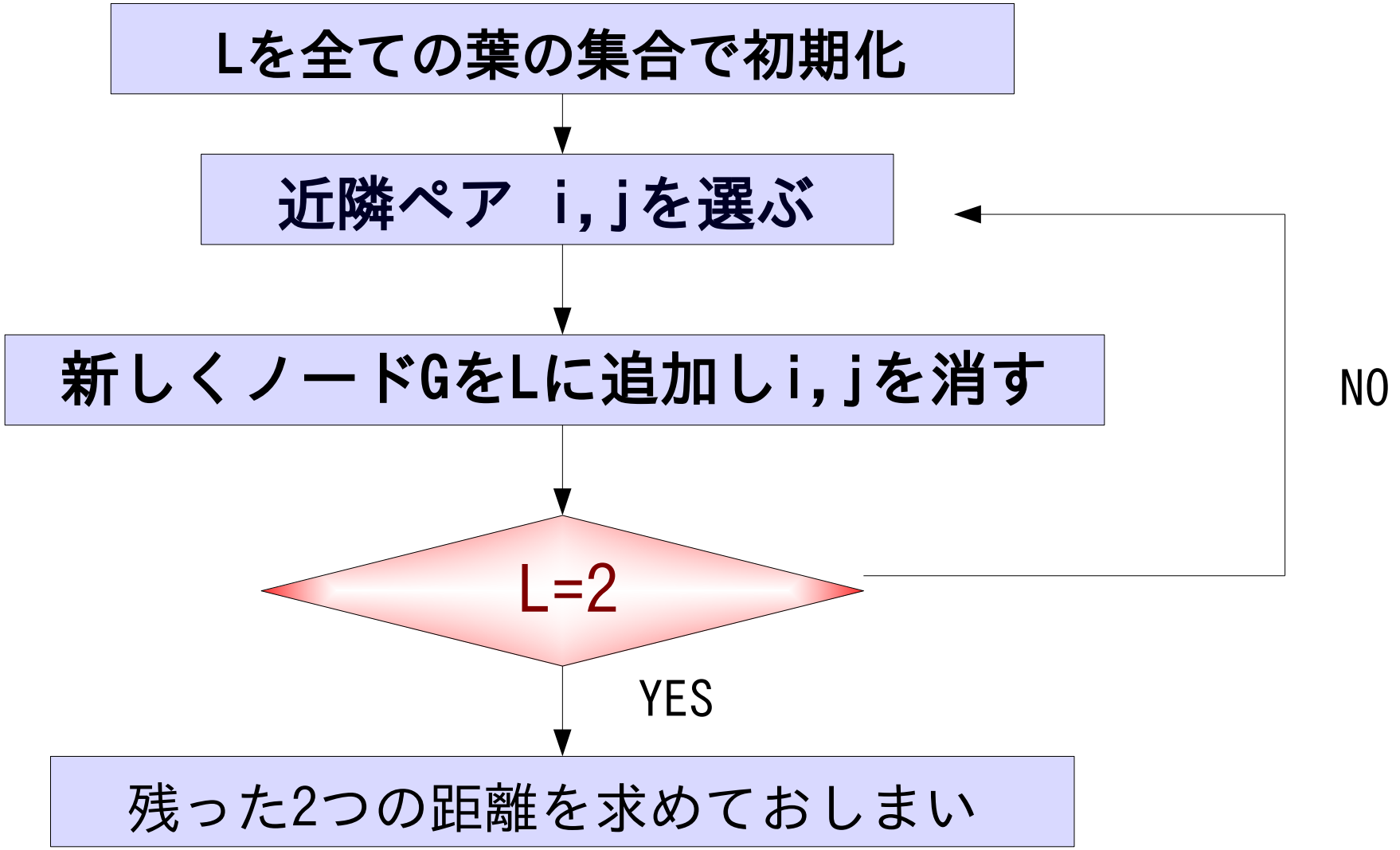
$$L=\{A,B,C,D,E,F,G\}$$

d(i,j)	C	D	E	F	G
C		7	6	8	3
D			5	9	6
E				8	5
F					7
G					



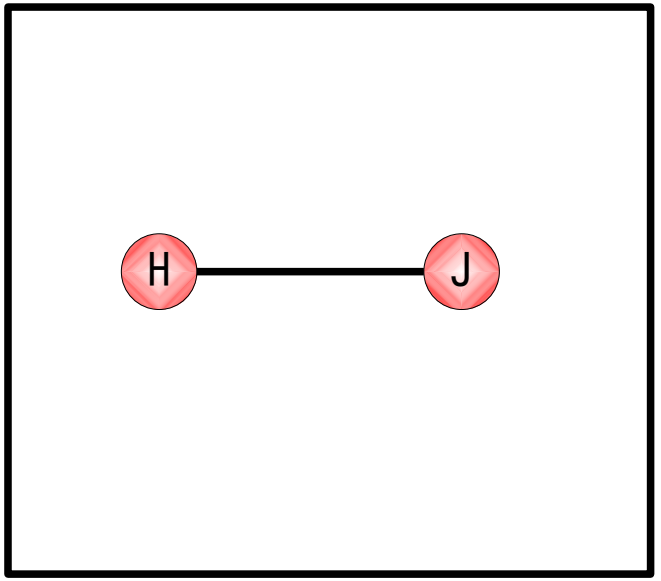
$$L=\{C,D,E,F,G\}$$

NJ法(近隣結合法)


Lを全ての葉の集合で初期化
近隣ペア i,jを選ぶ
新しくノードGをLに追加しi,jを消す
L=2
NO
YES
残った2つの距離を求めておしまい

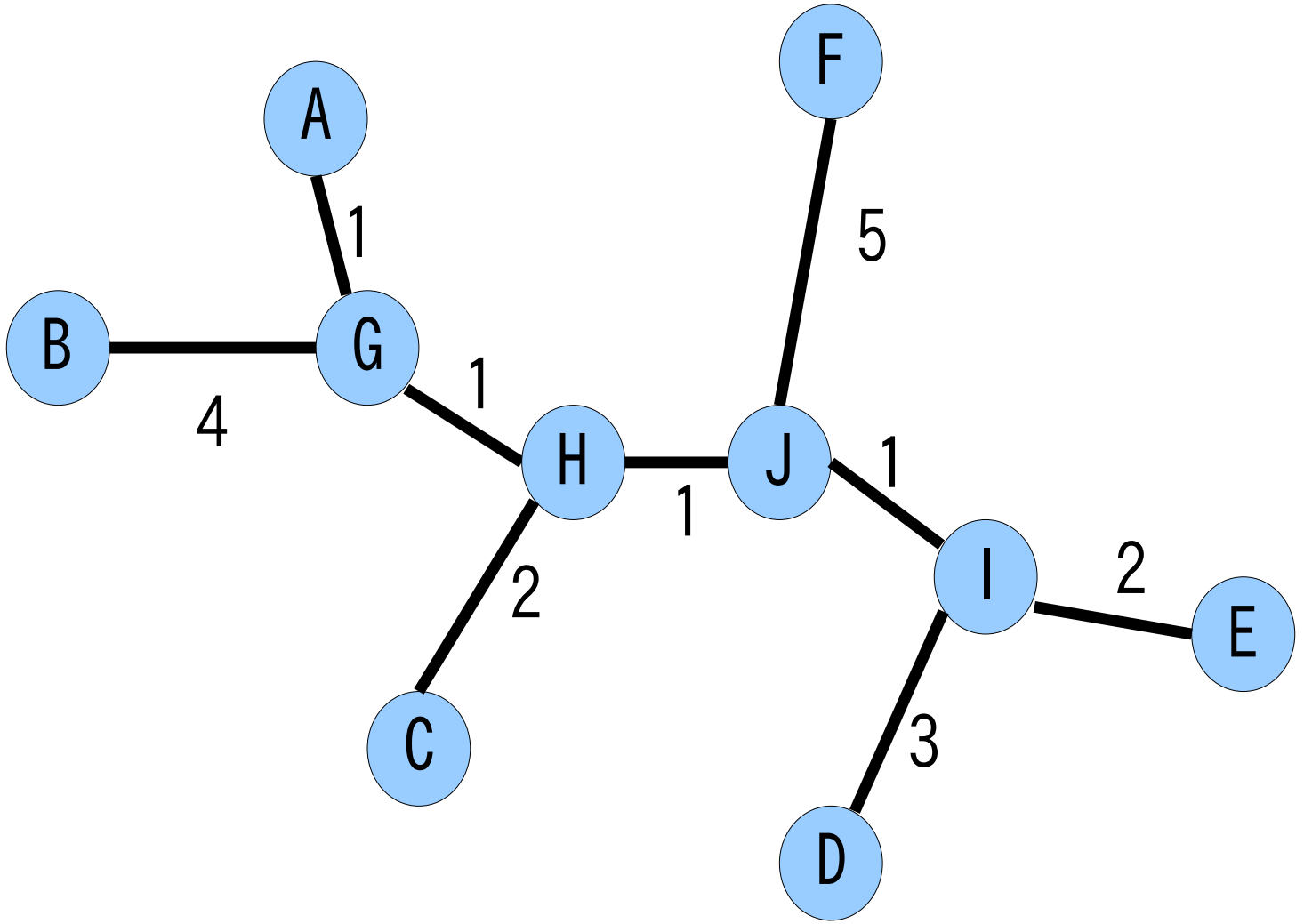
NJ法(近隣結合法)アルゴリズム

|L| = 2　　→　　Yes



$$L = \{H, J\}$$

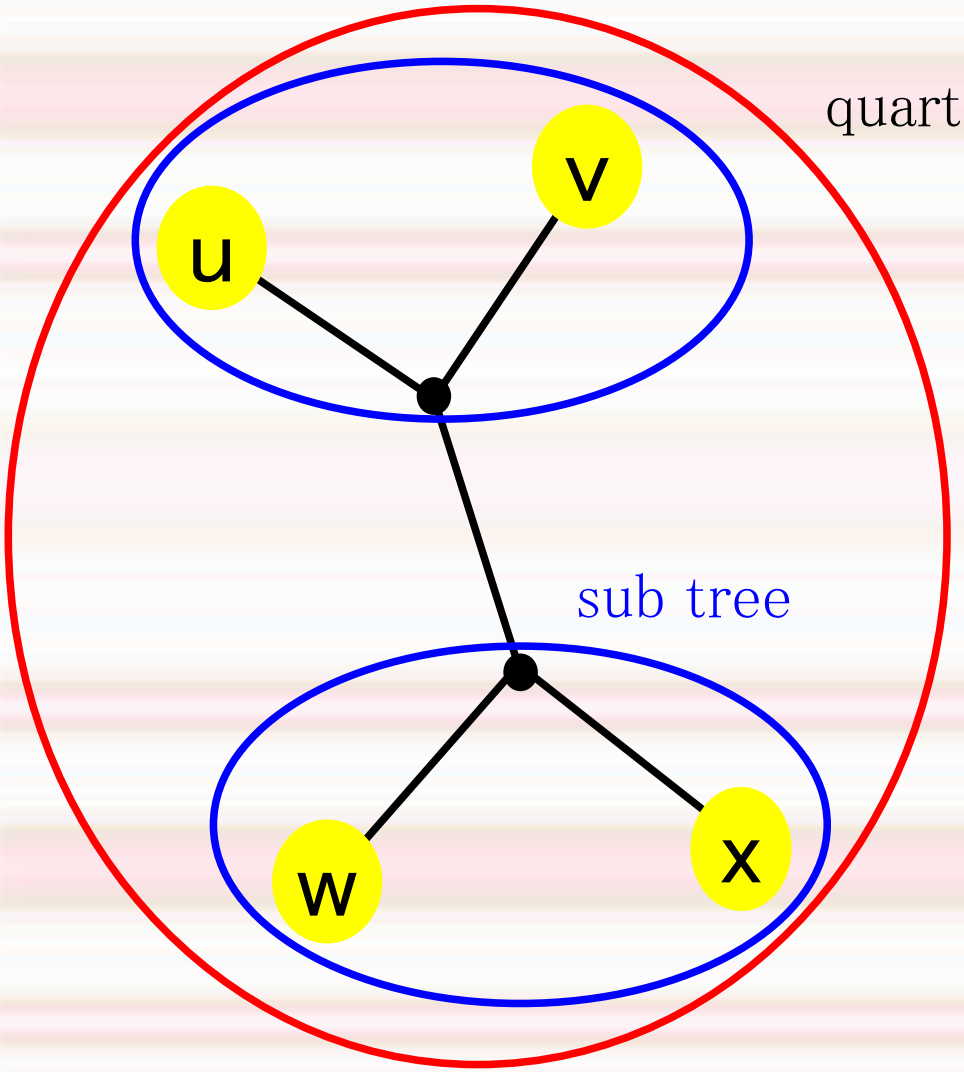
NJ法(近隣結合法) アルゴリズム



Quartet Method

- Quartet Method
  - C uv|wxの定義
  - consistentの定義
  - MaxcostとmincostとMincostの定義
  - totalcostの定義
  - S(T)の定義
  - quartet methodのアルゴリズム

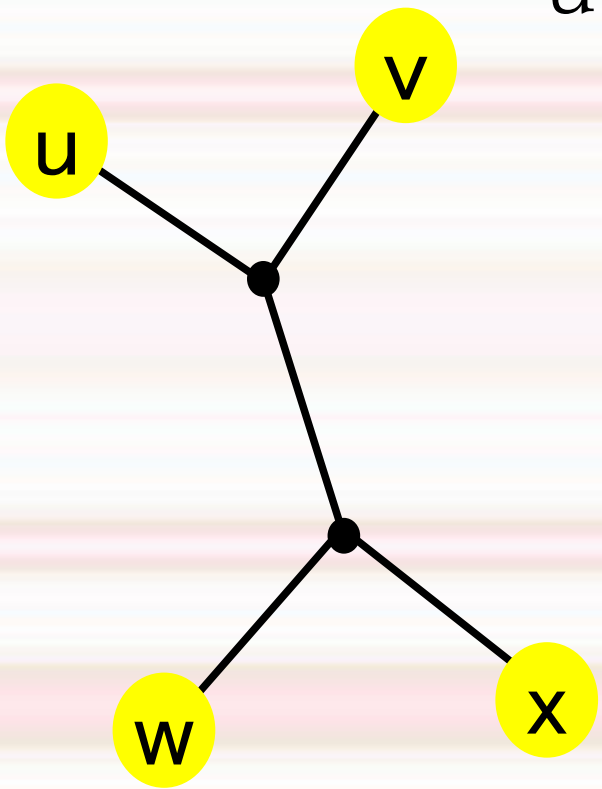
quartet tree とは



2つの葉を持つ2つの

subtreeが連結したグラフ

C uv|wx の定義



uv|wx (quartet tree)

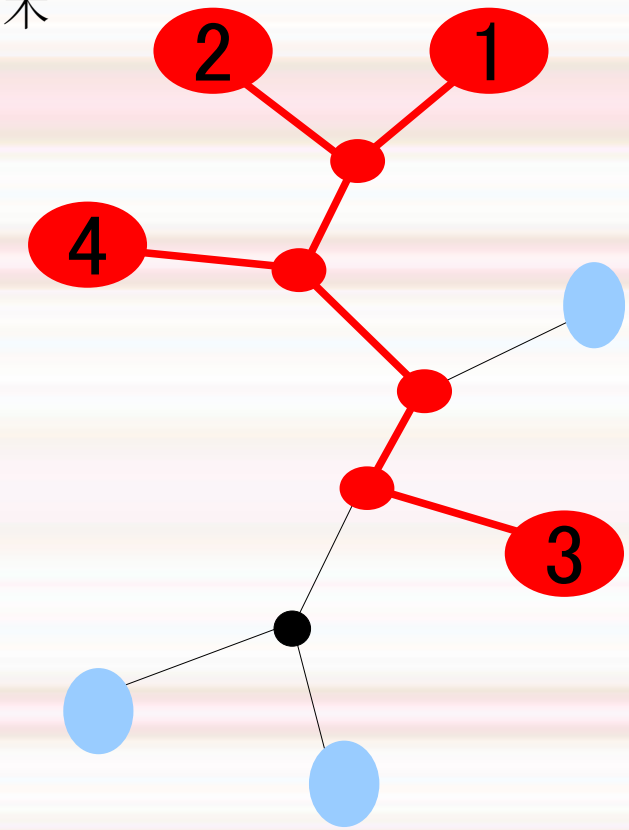
$$C_{uv|wx} = d(u,v) + d(w,x)$$

uv|wxのコスト
この値が小さいほど良い木

d(x,y) : x,y間の距離

consistentの定義

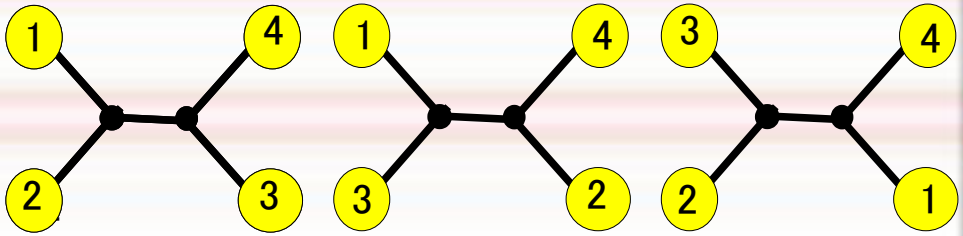
T:木



$$C_{uv|wx} = d(u,v) + d(w,x)$$

$\{1,2,3,4\}$

12|34　　13|24　　23|14



…

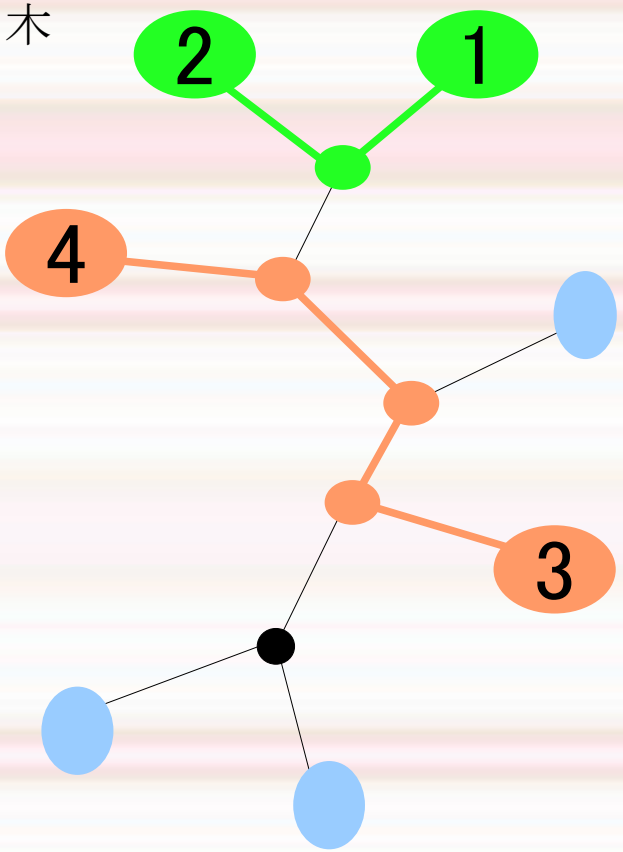
考えられる全てのquartet tree

…

1,2,3,4から3つのqarrtet tree
となる組み合わせが考えられる

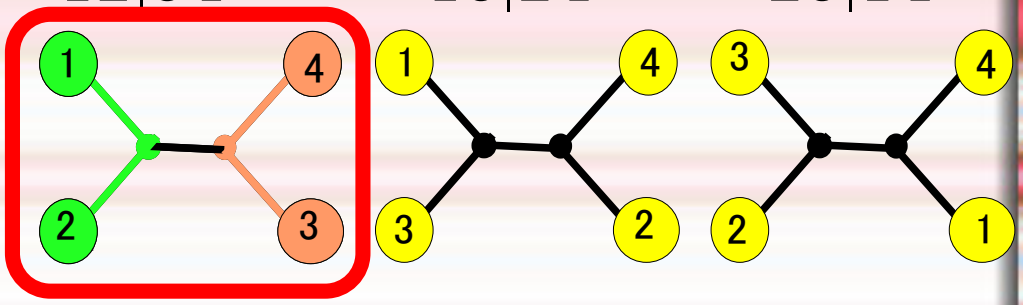
consistentの定義

T:木



{1,2,3,4}

12|34　13|24　23|14



consistent

…

考えられる全てのquartet tree

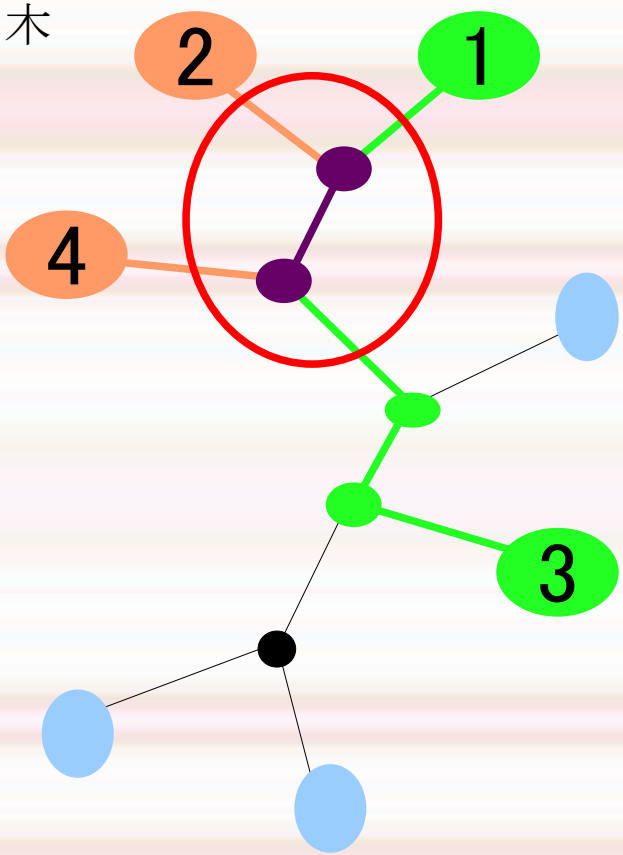
…

1,2間の道と3,4間の道は交わっていない

→consistentである

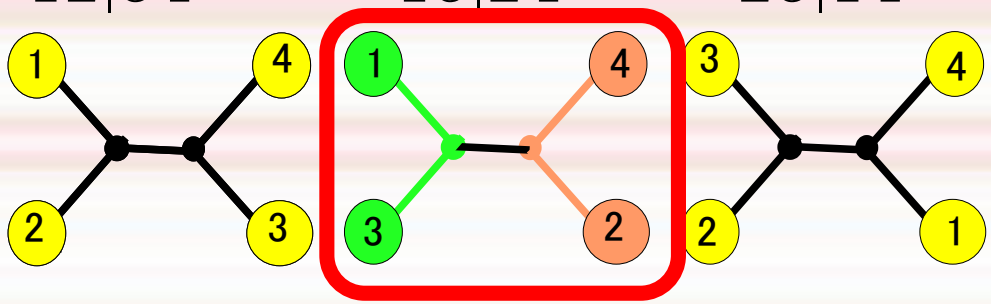
consistentの定義

T:木



{1,2,3,4}

12|34　13|24　23|14



×

…

考えられる全てのquartet tree

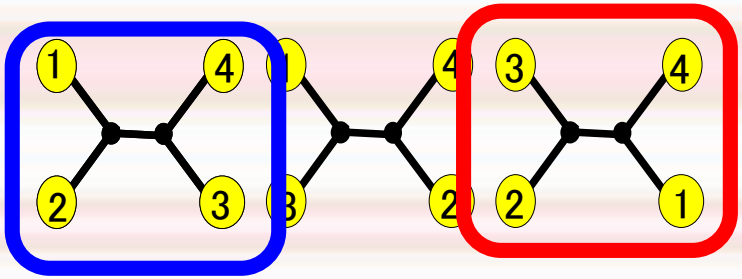
…

1,3間の道と2,4間の道は交わっている

→consistentではない

Maximum cost と minimum costの定義

$\{1,2,3,4\}$

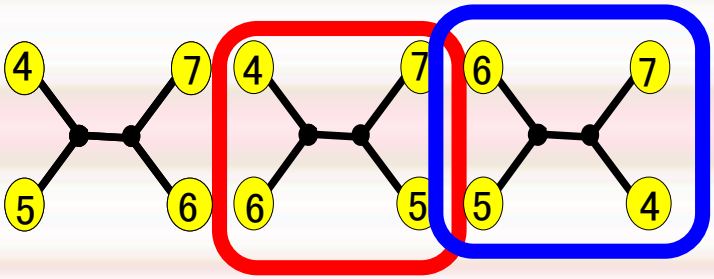


M:Maximum cost

quartetの3組の中からC uv|wx
が最大のものを選ぶ
すべてのquartetの総和を
木に対するMaximum costとする

.
.
.

$\{4,5,6,7\}$



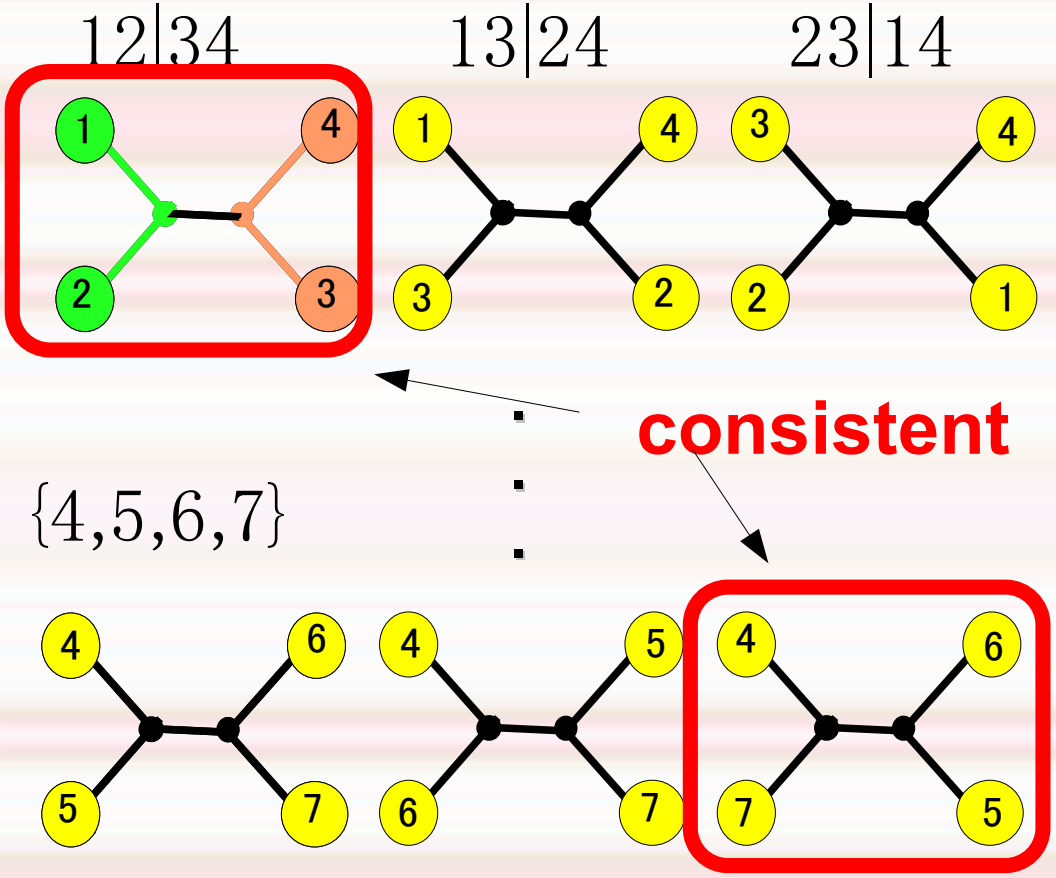
m:minimum cost

quartetの3組の中からC uv|wx
が最小のものを選ぶ
すべてのquartetの総和を
木に対するMaximum costとする

$$C_{uv|wx} = d(u,v) + d(w,x)$$

Total costの定義

{1,2,3,4}



$$C_{uv|wx} = d(u,v) + d(w,x)$$

CT : Total Cost

Consistentであるquartetの C uv|wxの総和を木のtotal costとする

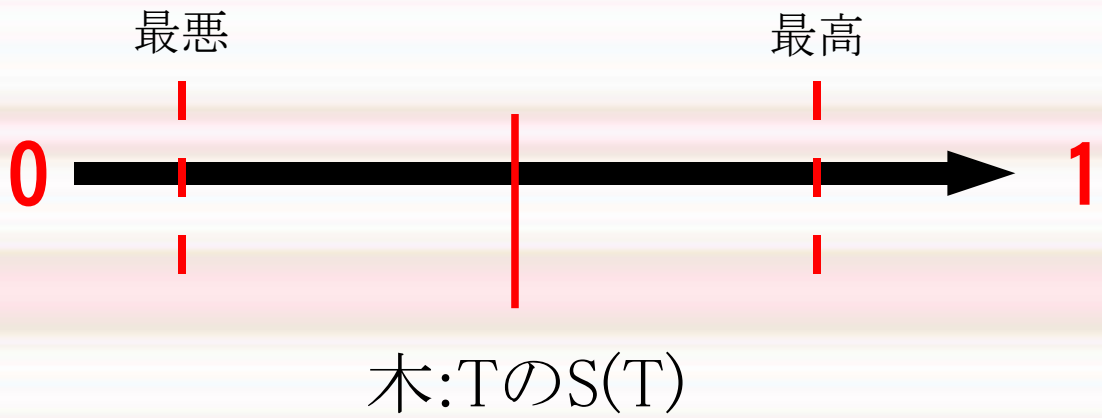
S(T) の定義

$$S(T) = \frac{(M - C_T)}{(M - m)}$$

M :maximum cost

m :minimum cost

C_T :total cost



木:TのS(T)

quartet methodのアルゴリズム

ランダムに木を生成

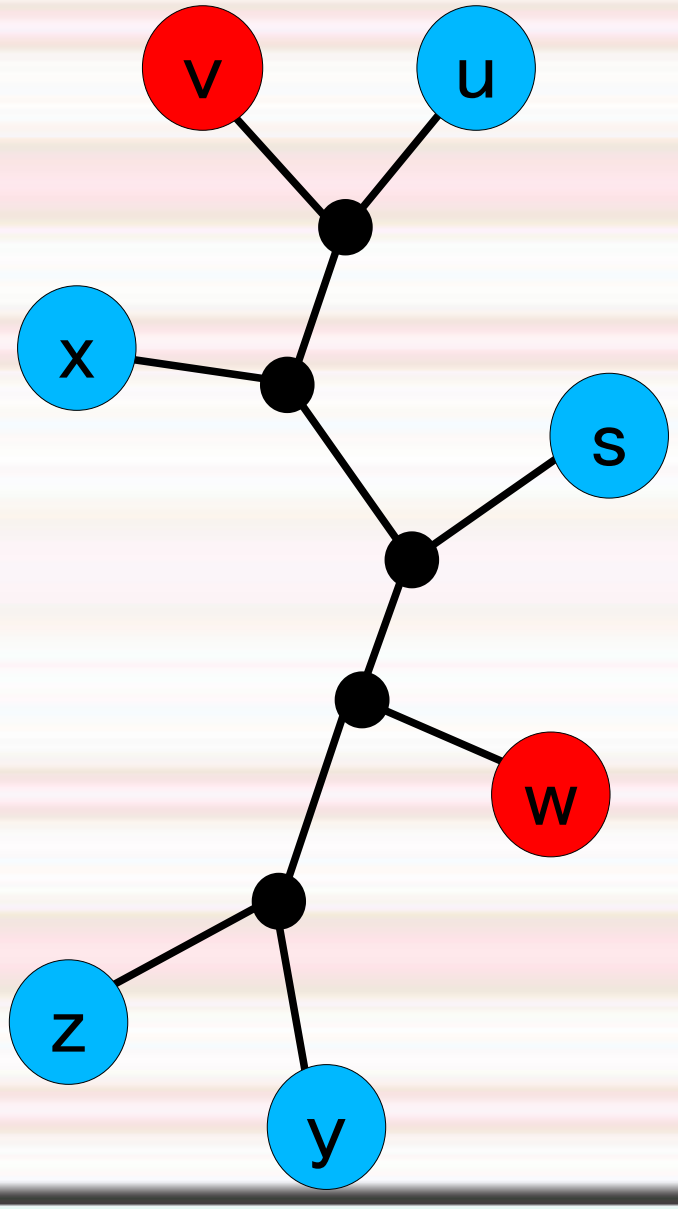
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

K回繰り返す

S(T)の値を計算



quartet methodのアルゴリズム

ランダムに木を生成

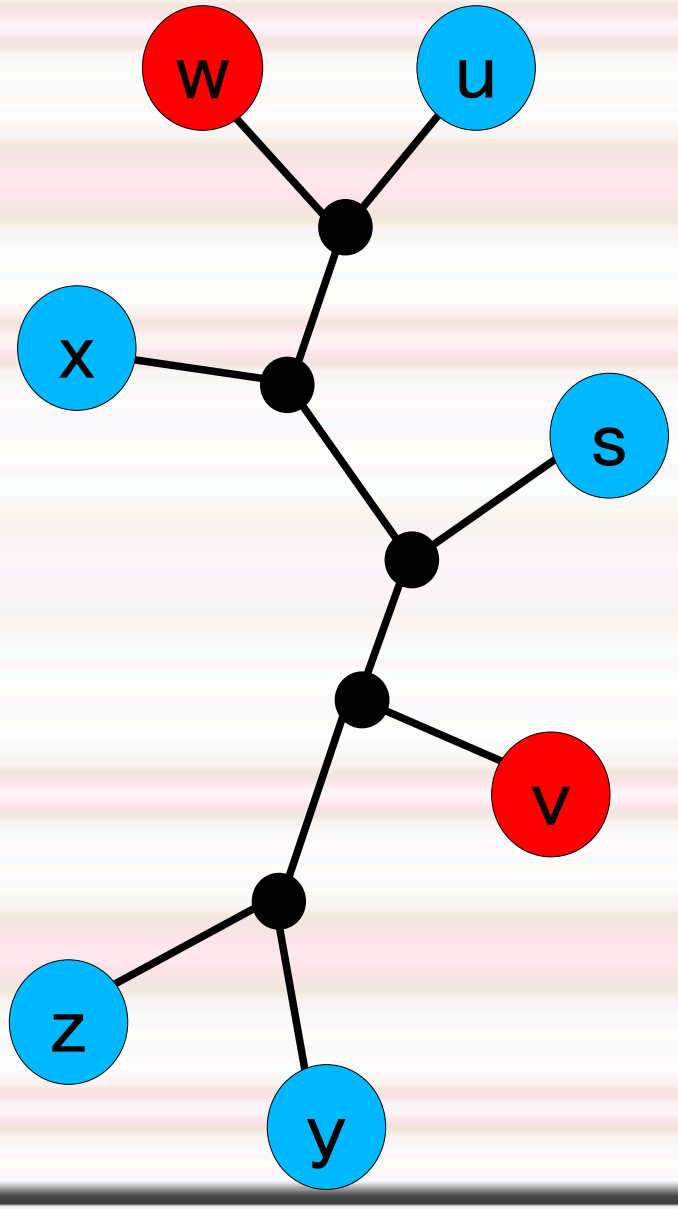
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

K回繰り返す

S(T)の値を計算



quartet methodのアルゴリズム

ランダムに木を生成

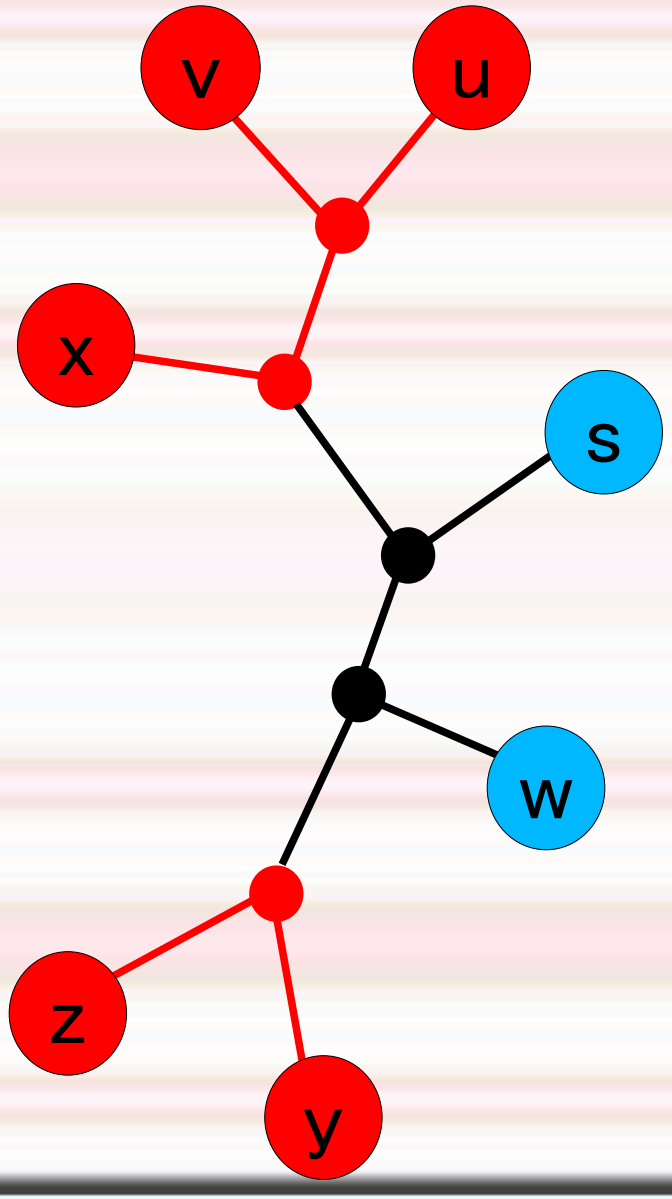
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

K回繰り返す

S(T)の値を計算



quartet methodのアルゴリズム

ランダムに木を生成

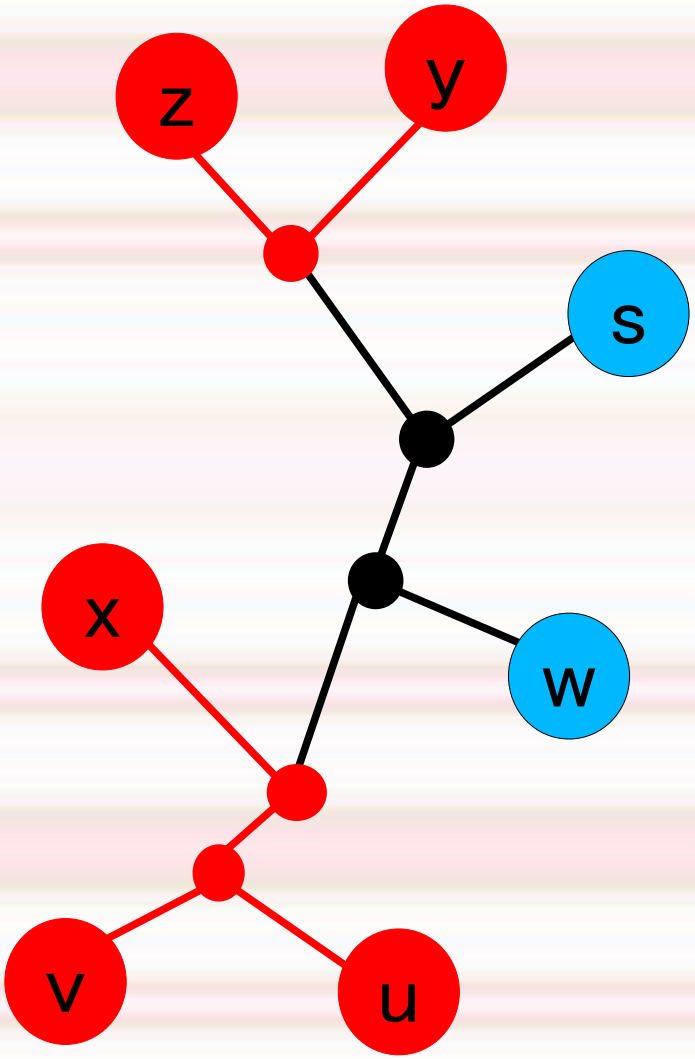
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

K回繰り返す

S(T)の値を計算



quartet methodのアルゴリズム

ランダムに木を生成

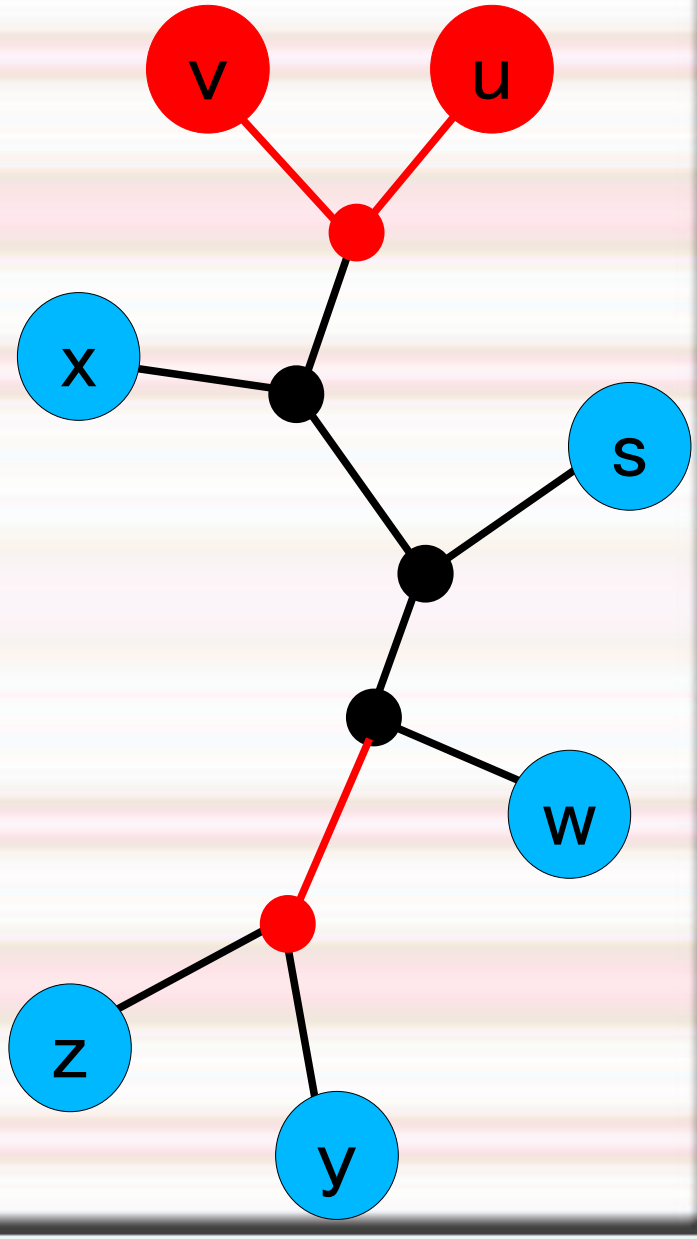
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

K回繰り返す

S(T)の値を計算



quartet methodのアルゴリズム

ランダムに木を生成

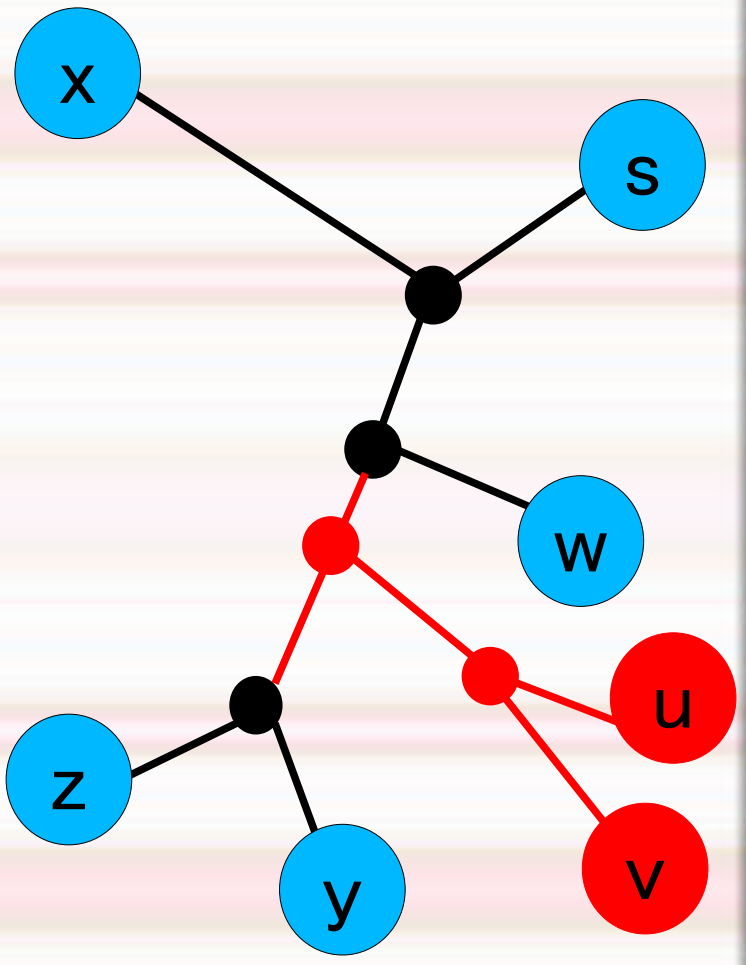
S(T)の値を計算

以下の操作からランダムに選ぶ

- leaf swap
- subtree swap
- subtree transfer

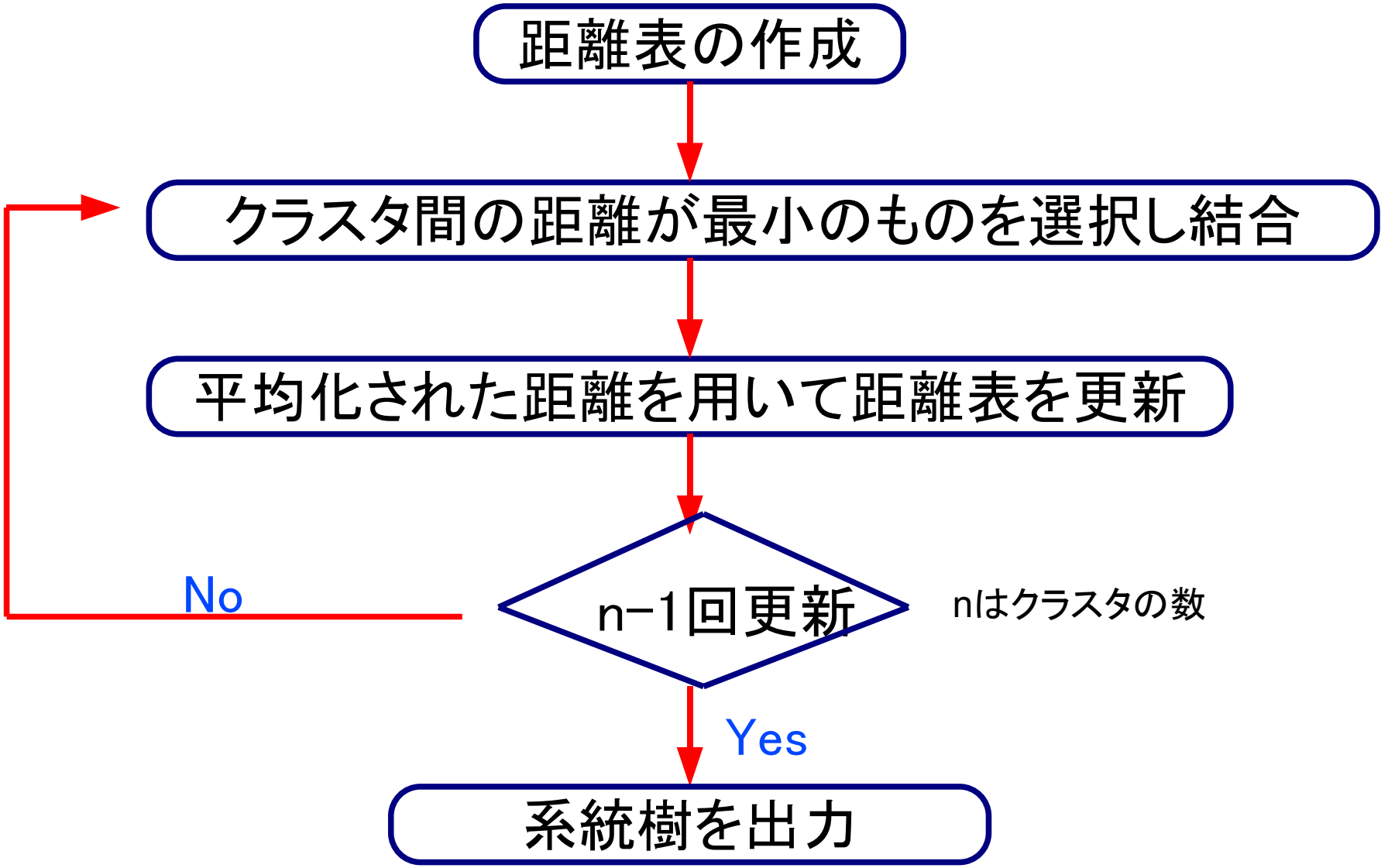
K回繰り返す

S(T)の値を計算



UPGMA

UPGMAのアルゴリズム



やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	AC	B	D
AC	0	?	?
B	−	0	10
D	−	−	0

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

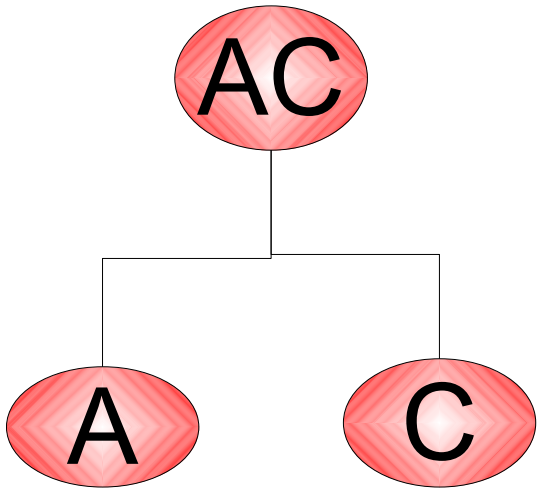
4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	AC	B	D
AC	0	?	?
B	−	0	10
D	−	−	0

やるとみせかけて

平均化された距離



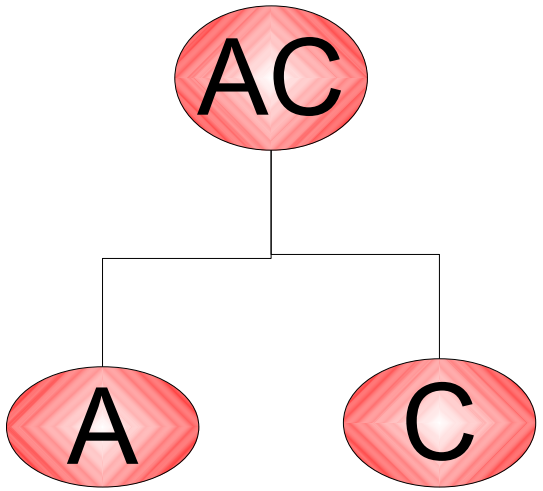
	A	B	C	D
A	0	9	4	7
B	–	0	11	10
C	–	–	0	14
D	–	–	–	0

やるとみせかけて

平均化された距離

$$d(AC, A) = \frac{\mathrm{d(A, C)}}{2}$$

$$d(AC, C) = \frac{\mathrm{d(A, C)}}{2}$$



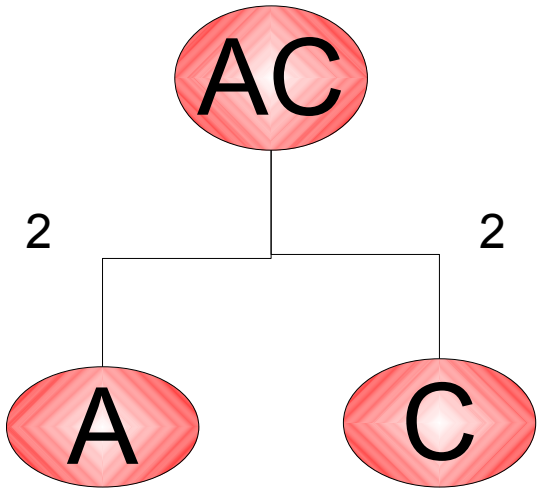
	A	B	C	D
A	0	9	4	7
B	–	0	11	10
C	–	–	0	14
D	–	–	–	0

やるとみせかけて

平均化された距離

$$d(AC, A) = \frac{\mathrm{d(A, C)}}{2} = 2$$

$$d(AC, C) = \frac{\mathrm{d(A, C)}}{2} = 2$$



	A	B	C	D
A	0	9	4	7
B	–	0	11	10
C	–	–	0	14
D	–	–	–	0

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

→

	AC	B	D
AC	0	?	?
B	−	0	10
D	−	−	0

$$d(AC,B)=\frac{d(A,B)+d(C,B)}{2}$$

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

→

	AC	B	D
AC	0	?	?
B	−	0	10
D	−	−	0

$$d(AC,B)=\frac{d(A,B)+d(C,B)}{2}=6.5$$

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

→

	AC	B	D
AC	0	6.5	?
B	−	0	10
D	−	−	0

$$d(AC,B)=\frac{d(A,B)+d(C,B)}{2}=6.5$$

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	A	B	C	D
A	0	9	4	7
B	−	0	11	10
C	−	−	0	14
D	−	−	−	0

→

	AC	B	D
AC	0	6.5	10.5
B	−	0	10
D	−	−	0

$$d(AC,D)=\frac{d(A,D)+d(C,D)}{2}=10.5$$

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	AC	B	D
AC	0	6.5	10.5
B	−	0	10
D	−	−	0

→

	ACB	D
ACB	0	10.25
D	−	0

$$d(ACB, D) = \frac{d(AC, D) + d(B, D)}{2} = 10.25$$

やってみる。

1.距離表を作成。

2.クラスタ間の距離が最小のものを選択し結合。

3.平均化された距離を用いて距離表を更新。

4.n−1回更新してなかったら２へ戻る。

5.n−1回更新してたら系統樹を出力。

	AC	B	D
AC	0	6.5	10.5
B	−	0	10
D	−	−	0

→

	ACB	D
ACB	0	10.25
D	−	0

$$d(ACB, D) = \frac{d(AC, D) + d(B, D)}{2} = 10.25$$