



Kolmogorov 記述量に基づく類似度を 用いた方言の自動分類

谷研究室 大江碧

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

研究動機・目的

動機

方言に対して興味があった



自分の住む地域と全国の方言がどれ位似ているかを単純に知りたいと思った

目的

全国の方言の類似性を調べて可視化

手法

Ming Li らにより定義された Kolmogorov 記述量に基づく類似度を表す距離

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

研究の背景

単語単位での類似度は、アンケート等で統計的に調べられている。



文章データがなく、あったとしても、統計的に調べるのは難しい。
文章を対象とした研究をするのは、初めてだと思われる。



類似性を調べるのに、Kolmogorov 記述量を使う!!!

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化



研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化



研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化



研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化



研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化



研究の流れ

text ファイル

WAV ファイル

mp3 ファイル

圧縮 → ファイルサイズを調べる

Kolmogorov 記述量に基づく距離を算出

グラフの生成

可視化

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

クラスタリングとは

データ解析手法の一つ

分類対象集合

大阪

長崎

香川

高知

北海道

青森

熊本

東京

分割した結果

本州

九州

青森

東京

長崎

大阪

熊本

四国

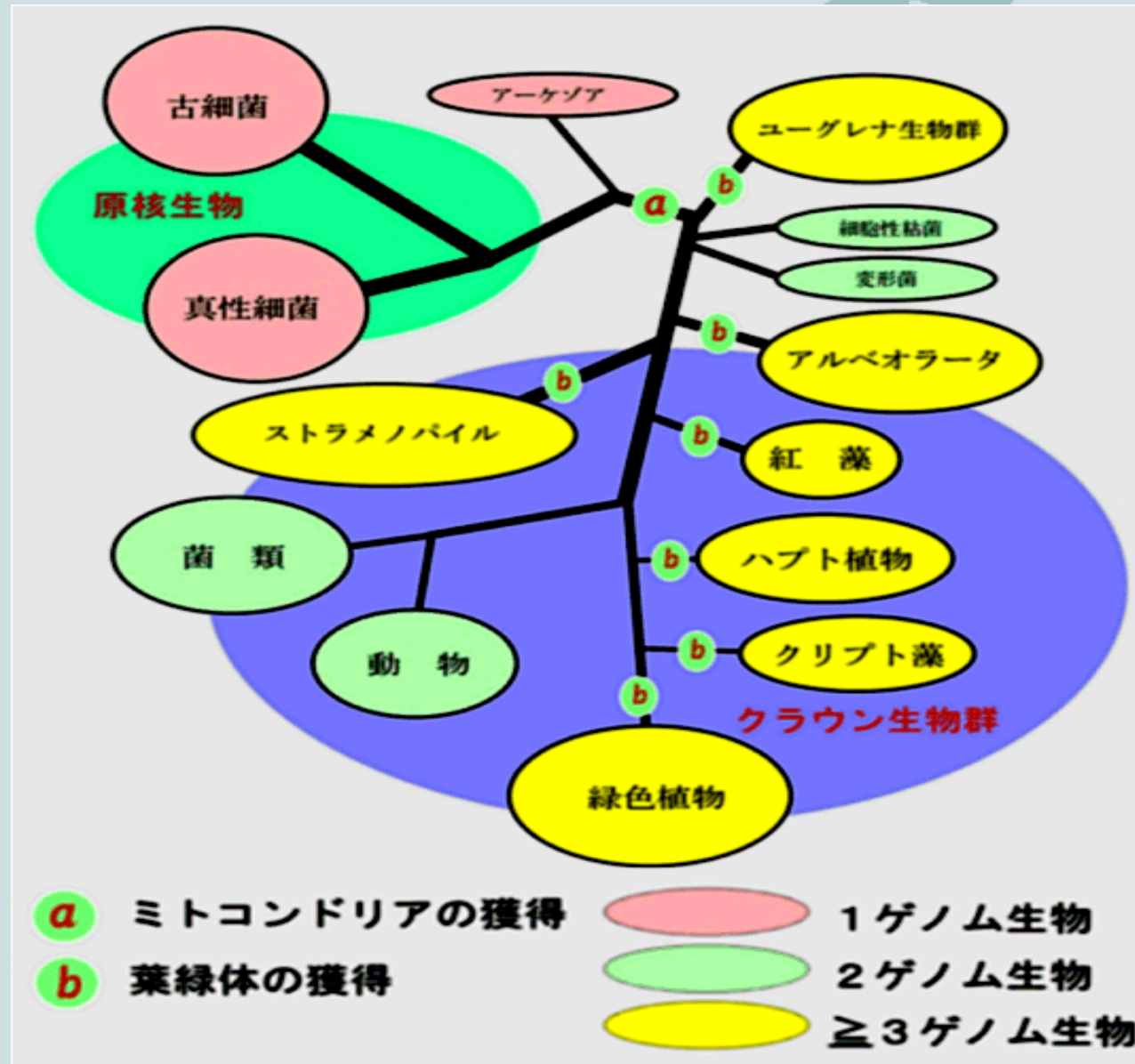
北海道

香川

北海道

高知

系統樹



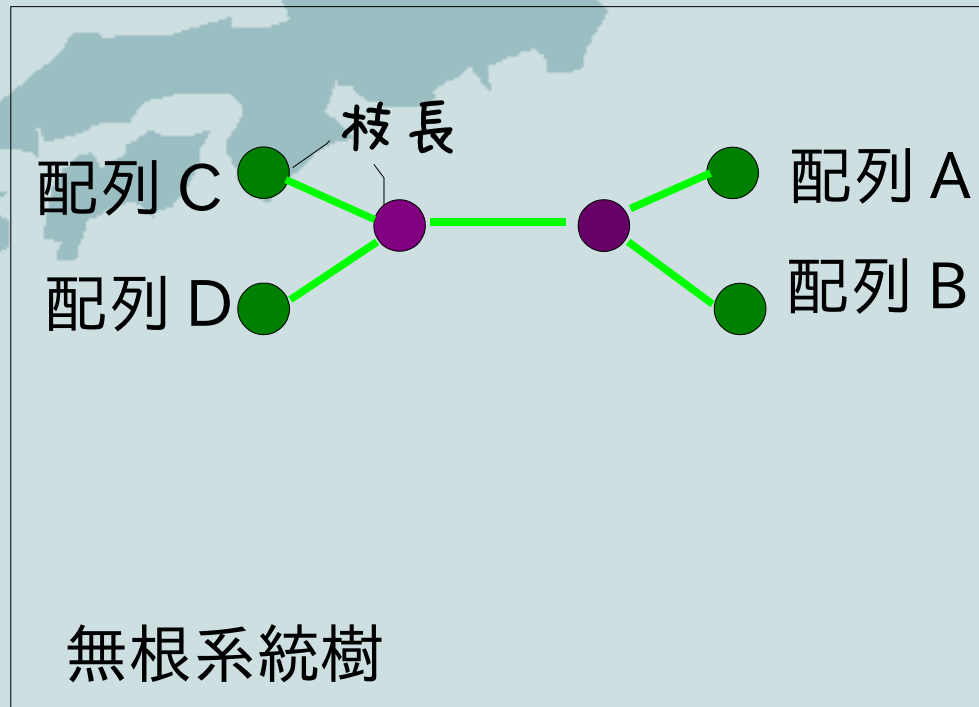
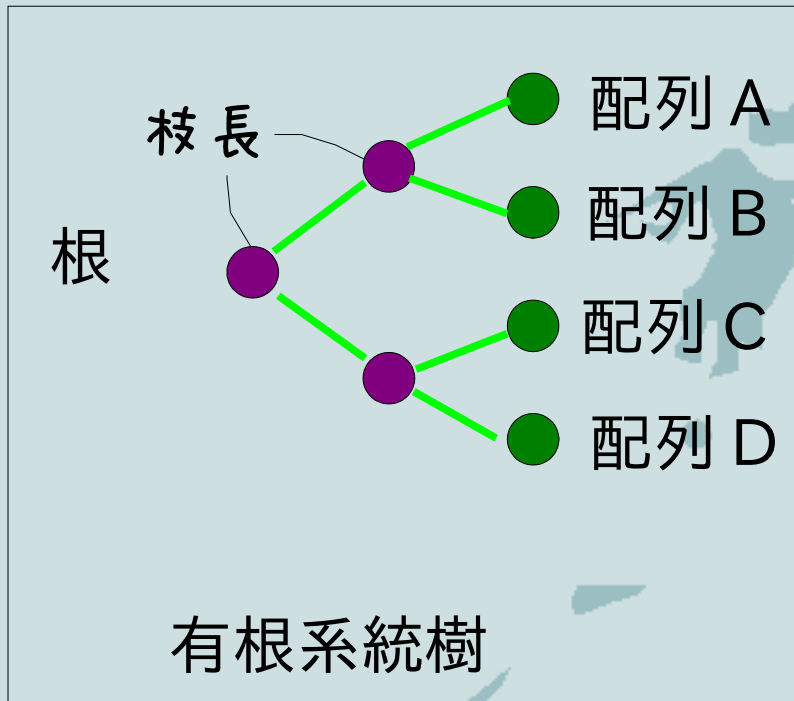
系統樹

分類単位を表す葉，

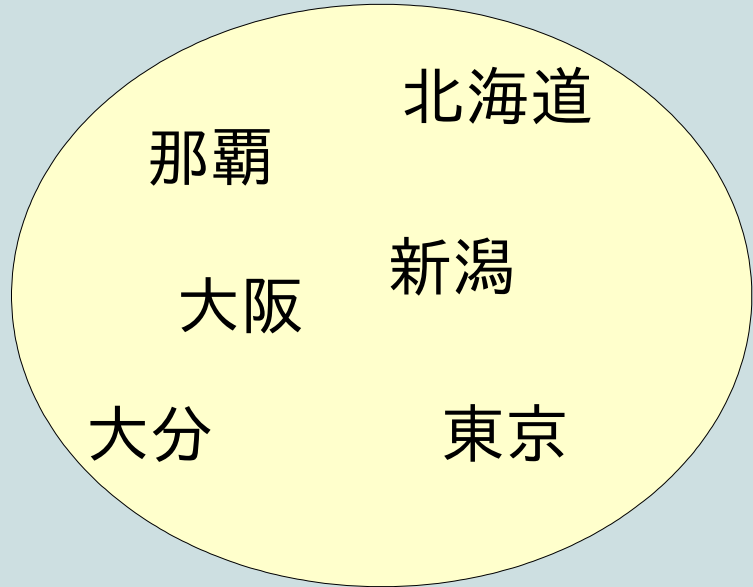
下図では配列 A から D として

示されている分類単位間の関係を表す

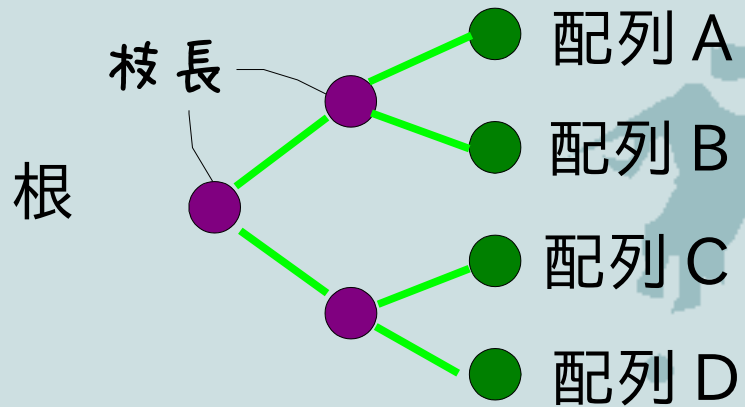
節と枝からなっている



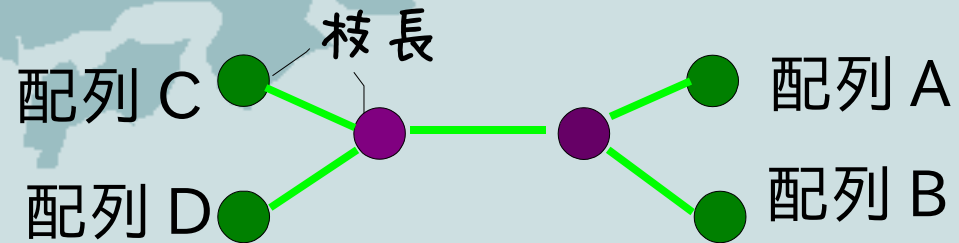
系統樹



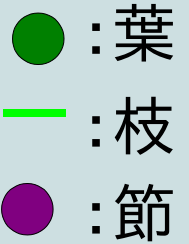
どこから発生したか不明



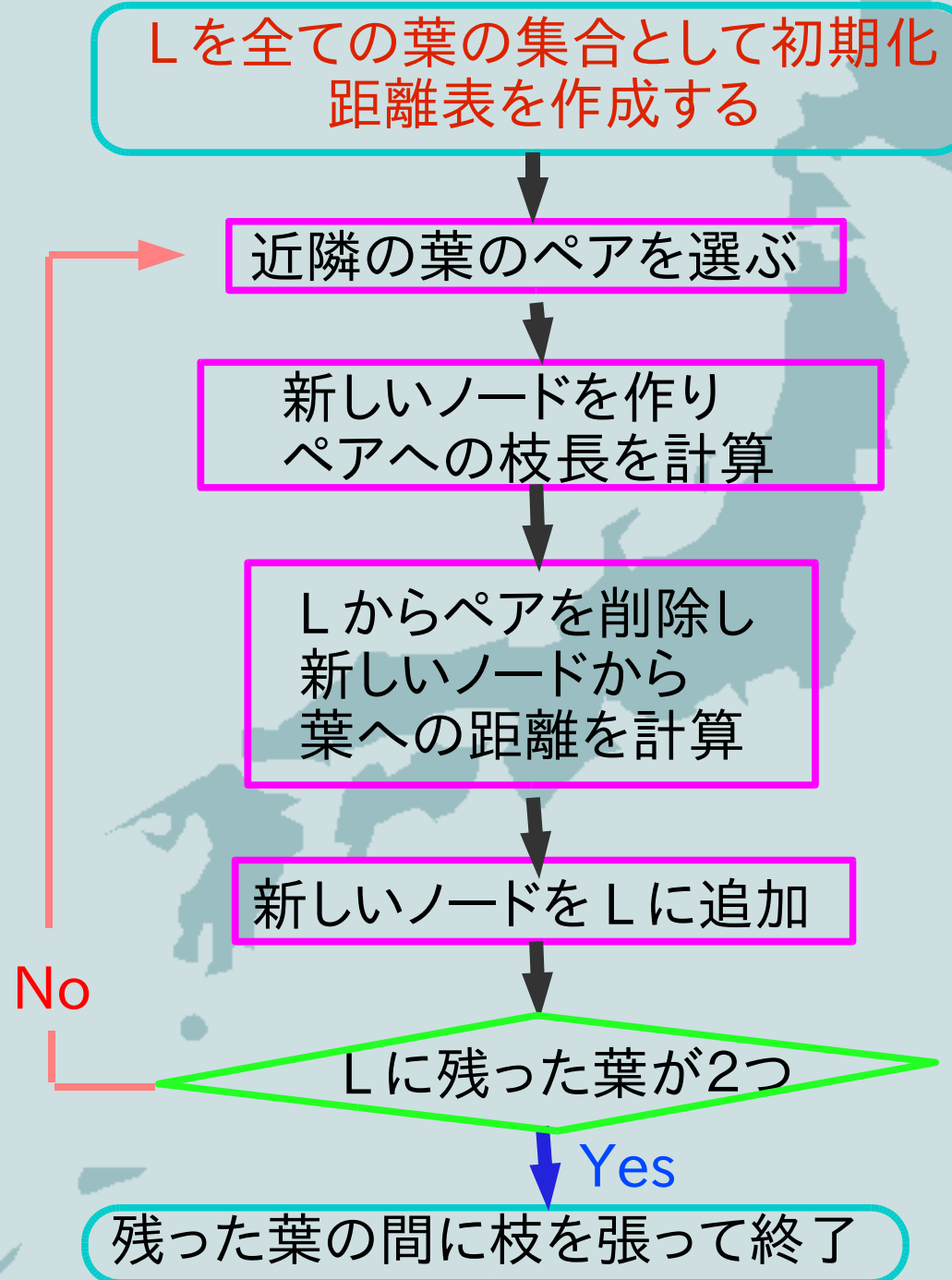
有根系統樹



無根系統樹



NJ 法



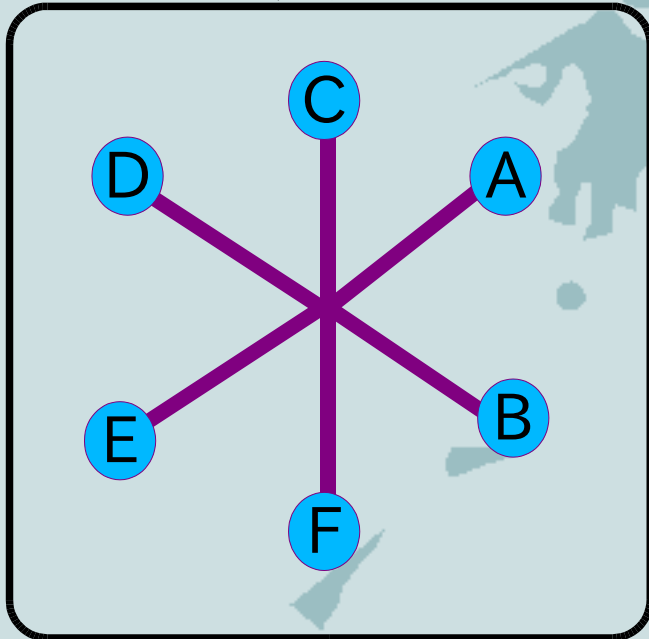
NJ 法 STEP 1

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	-	-	-	-	-	-

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

$N = 6$

初期状態



L を全ての葉の集合として初期化
距離表を作成する

近隣の葉のペアを選ぶ

新しいノードを作り
ペアへの枝長を計算

L からペアを削除し
新しいノードから
葉への距離を計算

新しいノードを L に追加

L に残った葉が2つ

Yes

残った葉の間に枝を張って終了

No

NJ 法 STEP 1

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	-	-	-	-	-	-

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

$$N = 6$$

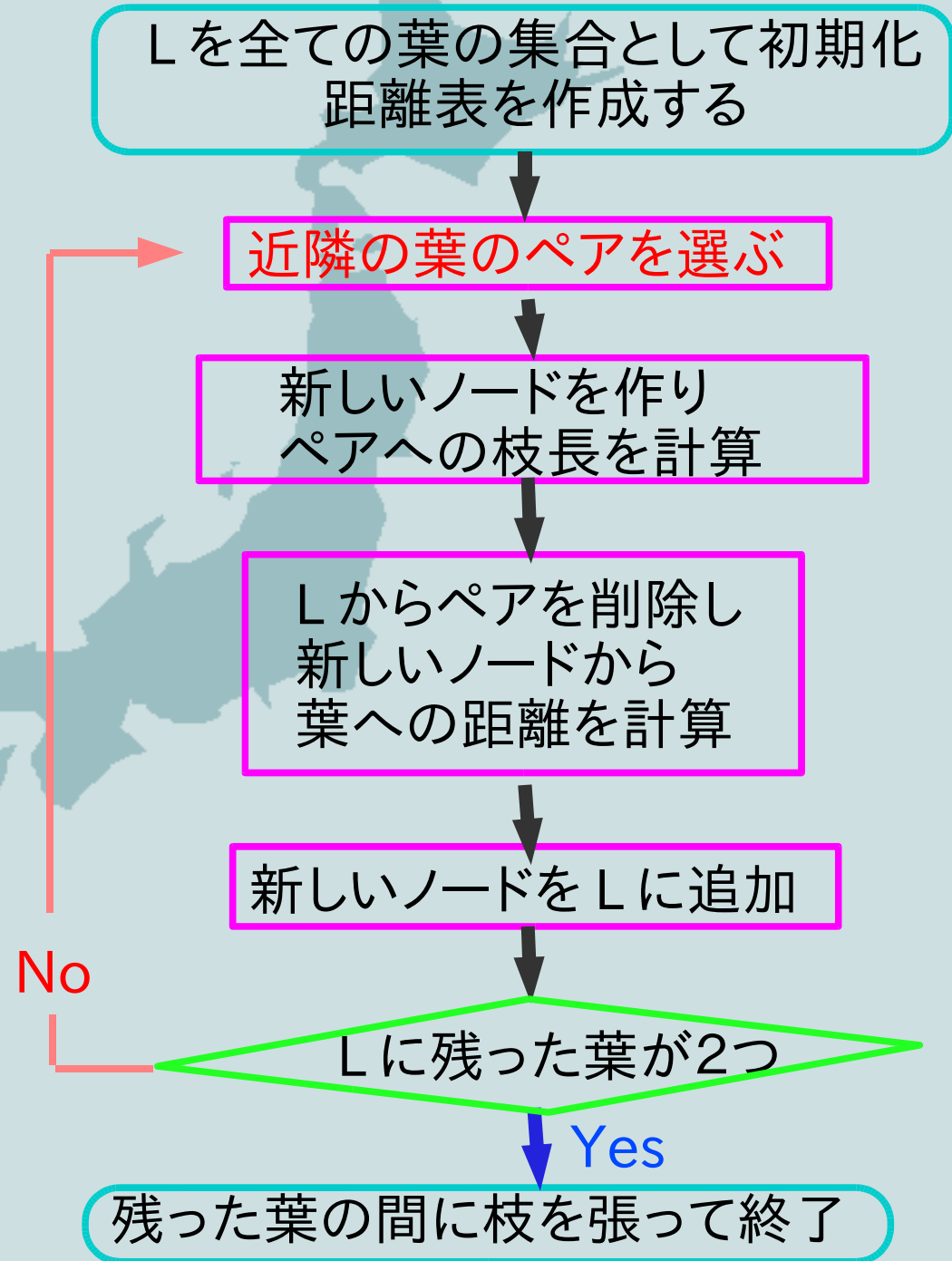
$$r(i) = \sum_{k \in L, k \neq i}^{N-1} d(i, k)$$

$$r(A) = d(AB) + d(AC) + d(AD) + d(AE) + d(AF) = 5 + 4 + 7 + 6 + 8 = 30$$

NJ 法 STEP 2

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	-	-	-	-	-	-

$$M(i, j) = d(i, j) - \frac{r(i) + r(j)}{N - 2}$$



NJ 法 STEP 2

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	-	-	-	-	-	-



M(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	-	-	-	-	-	-

$$M(i, j) = d(i, j) - \frac{r(i) + r(j)}{N - 2}$$

$$r(A) = 30 \quad r(D) = 38$$

$$r(B) = 42 \quad r(E) = 34$$

$$r(C) = 32 \quad r(F) = 44$$

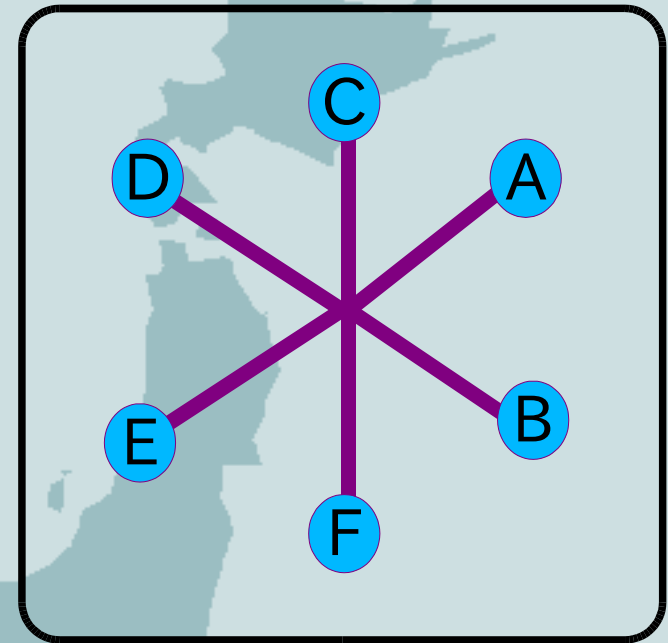
AB のペアの場合 ...

$$M(A, B) = d(A, B) - \frac{r(A) + r(B)}{N - 2}$$

$$= -13$$

NJ 法 STEP 2

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	-	-	-	-	-	-



ここでは AB を選ぶ



AB 近隣

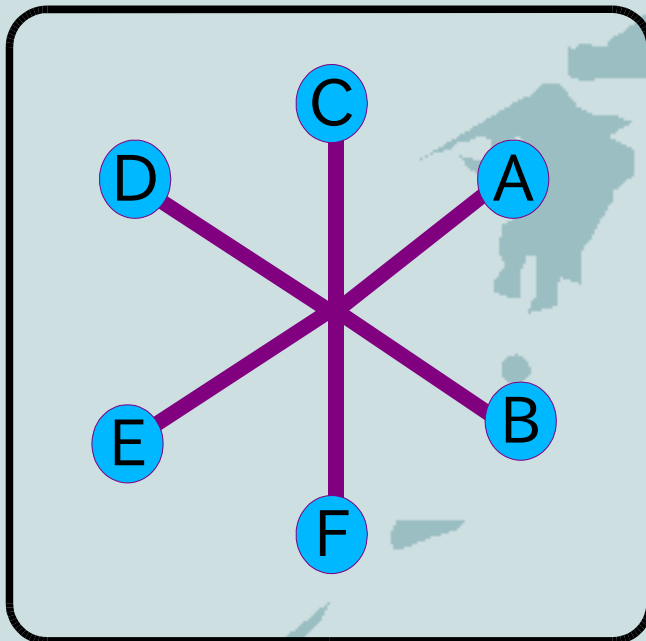
M(i,j)	A	B	C	D	E	F
A	-	-13	-11.5	-10	-10	-10.5
B	-	-	-11.5	-10	-10	-10.5
C	-	-	-	-10.5	-10.5	-11
D	-	-	-	-	-13	-11.5
E	-	-	-	-	-	-11.5
F	-	-	-	-	-	-

NJ 法 STEP 3

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	-	-	-	-	-	-

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

$N = 6$



L を全ての葉の集合として初期化
距離表を作成する

近隣の葉のペアを選ぶ

新しいノードを作り
ペアへの枝長を計算

L からペアを削除し
新しいノードから
葉への距離を計算

新しいノードを L に追加

L に残った葉が2つ

Yes

残った葉の間に枝を張って終了

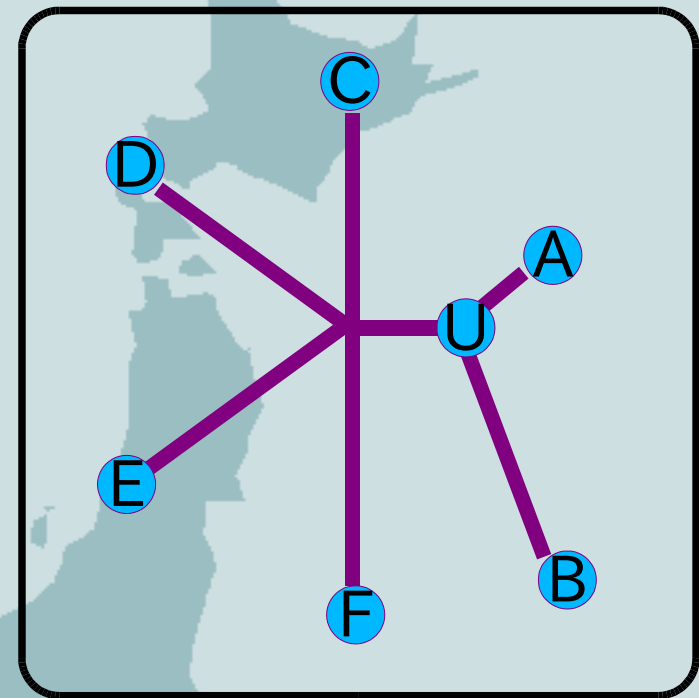
No

NJ 法 STEP 3

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	-	-	-	-	-	-



M(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	-	-	-	-	-	-



ここでは AB を選ぶ



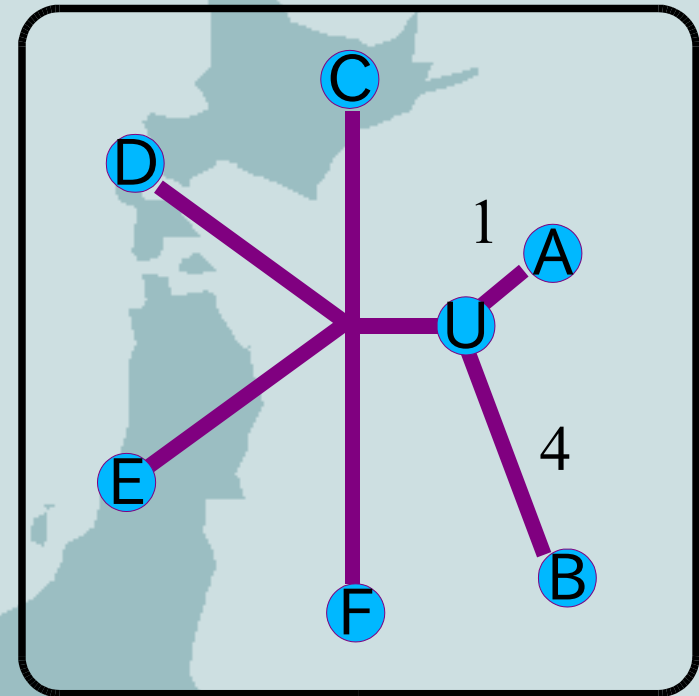
AB 近隣



“U” と呼ばれる新しい node を作る

NJ 法 STEP 3

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	-	-	-	-	-	-



node U から A や B への枝長を計算する

$$S(A, U) = \frac{d(A, B)}{2} - \frac{r(A) - r(B)}{2(N-2)} = 1$$

$$S(B, U) = d(A, B) - S(A, U) = 4$$

$$\begin{aligned} r(A) &= 30 & r(D) &= 38 \\ r(B) &= 42 & r(E) &= 34 \\ r(C) &= 32 & r(F) &= 44 \end{aligned}$$

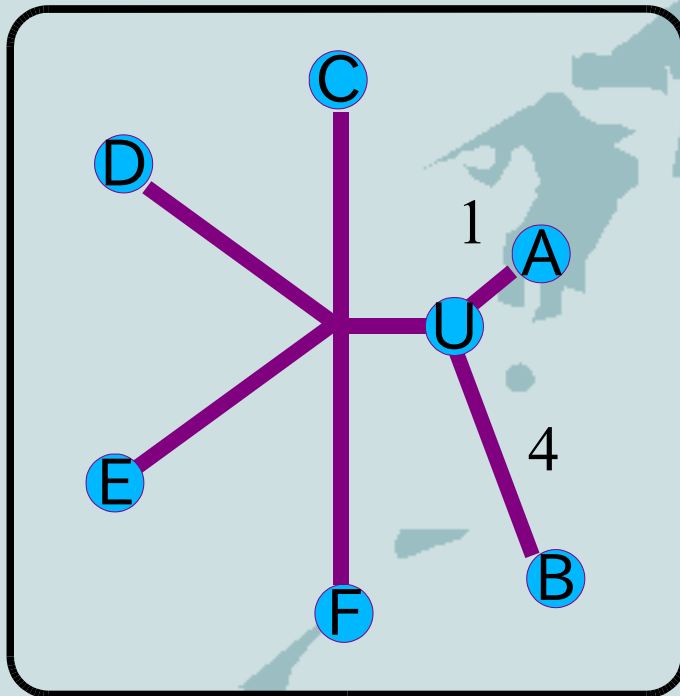
node U からそれぞれの末端の node までの新しい距離を定義する

NJ 法 STEP 4

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	-	-	-	-	-	-

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

$N = 6$



L を全ての葉の集合として初期化
距離表を作成する

近隣の葉のペアを選ぶ

新しいノードを作り
ペアへの枝長を計算

L からペアを削除し
新しいノードから
葉への距離を計算

新しいノードを L に追加

L に残った葉が2つ

Yes

残った葉の間に枝を張って終了

No

NJ 法 STEP 4

L から A,B を除く $L = \{C, D, E, F\}$

U から L に含まれる葉 q への距離 d を計算する

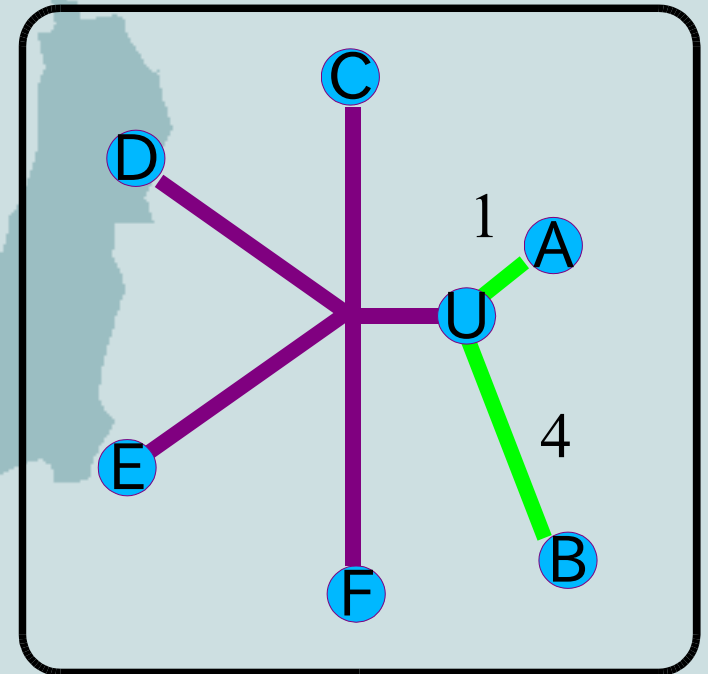
d(i,j)	A	B	C	D	E	F
A	-	5	4	7	6	8
B	-	-	7	10	9	11
C	-	-	-	7	6	8
D	-	-	-	-	5	9
E	-	-	-	-	-	8
F	-	-	-	-	-	-

$$d(C, U) = \frac{d(A, C) + d(B, C) - d(A, B)}{2} = 3$$

$$d(D, U) = \frac{d(A, D) + d(B, D) - d(A, B)}{2} = 6$$

$$d(E, U) = \frac{d(A, E) + d(B, E) - d(A, B)}{2} = 5$$

$$d(F, U) = \frac{d(A, F) + d(B, F) - d(A, B)}{2} = 7$$

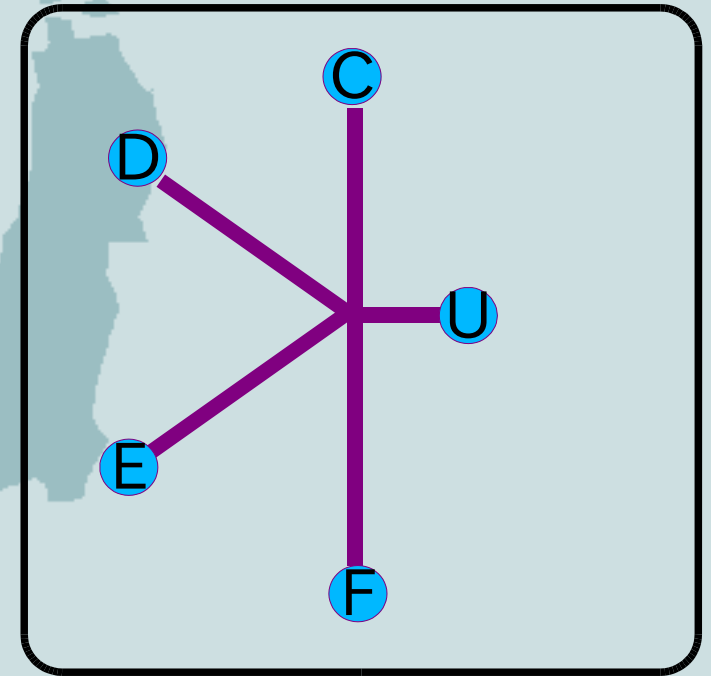


NJ 法 STEP 4

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	-	-	-	-	-	-



U を一つの葉とみなす



d(ij)	U	C	D	E	F
U	-	3	6	5	7
C	-	-	7	6	8
D	-	-	-	5	9
E	-	-	-	-	8
F	-	-	-	-	-

$$d(C, U) = 3$$

$$d(D, U) = 6$$

$$d(E, U) = 5$$

$$d(F, U) = 7$$

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

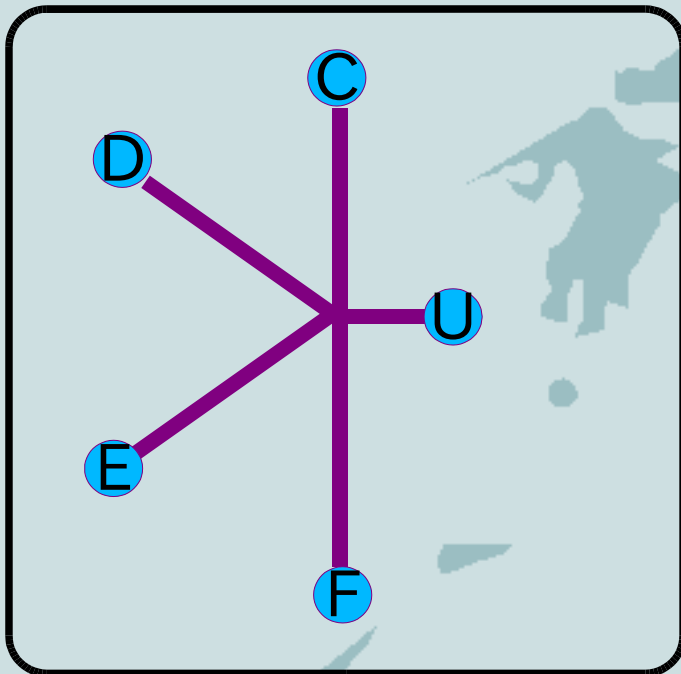
$$N = N - 1 = 5$$

NJ 法 STEP 5

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	-	-	-	-	-	-

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

$N = 5$



Lを全ての葉の集合として初期化
距離表を作成する

近隣の葉のペアを選ぶ

新しいノードを作り
ペアへの枝長を計算

Lからペアを削除し
新しいノードから
葉への距離を計算

新しいノードをLに追加

Lに残った葉が2つ

Yes

残った葉の間に枝を張って終了

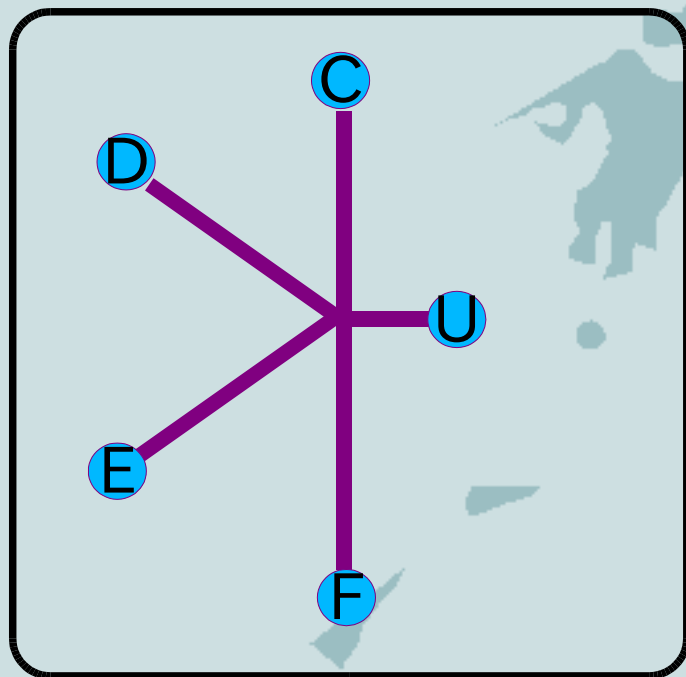
No

NJ 法

d(ij)	U	C	D	E	F
U	-	3	6	5	7
C	-	-	7	6	8
D	-	-	-	5	9
E	-	-	-	-	8
F	-	-	-	-	-

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

$$N = 5$$



L を全ての葉の集合として初期化
距離表を作成する

近隣の葉のペアを選ぶ

新しいノードを作り
ペアへの枝長を計算

L からペアを削除し
新しいノードから
葉への距離を計算

新しいノードを L に追加

L に残った葉が2つ

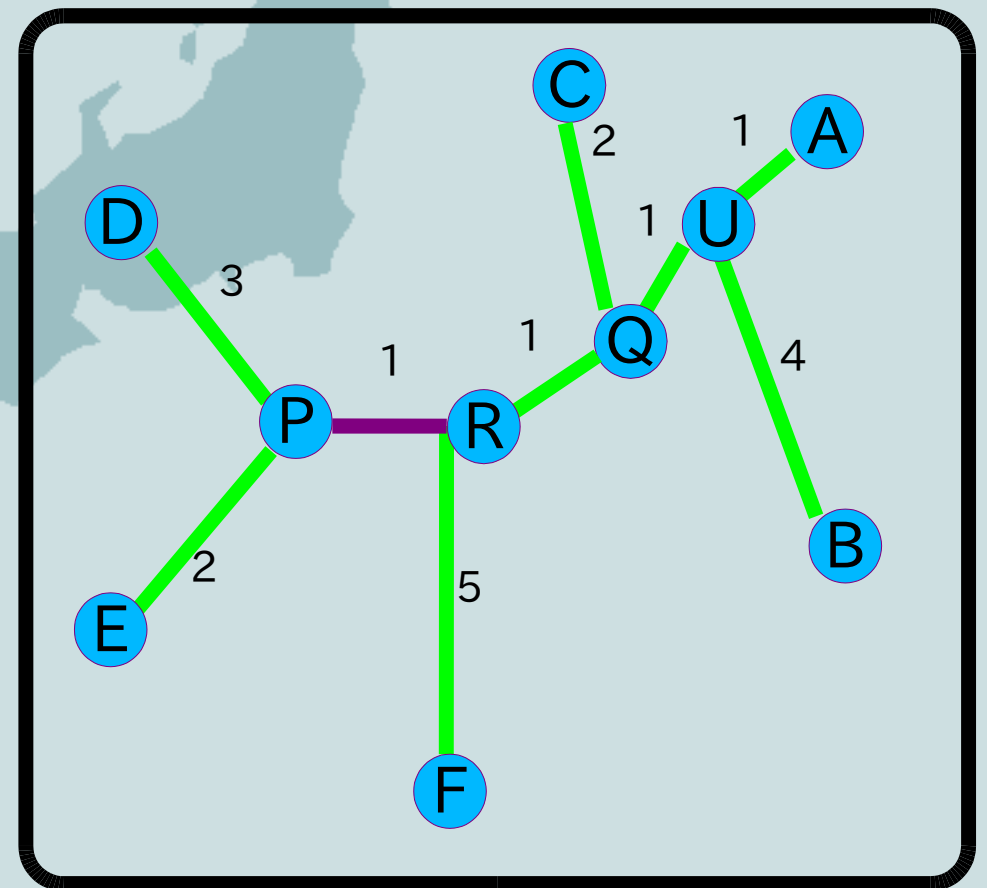
Yes

残った葉の間に枝を張って終了

No

系統樹完成

d(ij)	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	-	-	-	-	-	-



系統樹

niigata 1

むかしむかしあるところにおじじとおばばがあったと。おじじはやまへしばかりに、おばばはかわへせんたくにいったと。

おばばがせんたくをしていると、かわかみからおおきなももがどんぶりどんぶりことながれてきたと。おばばはそのももをひろっていえへかえったと。

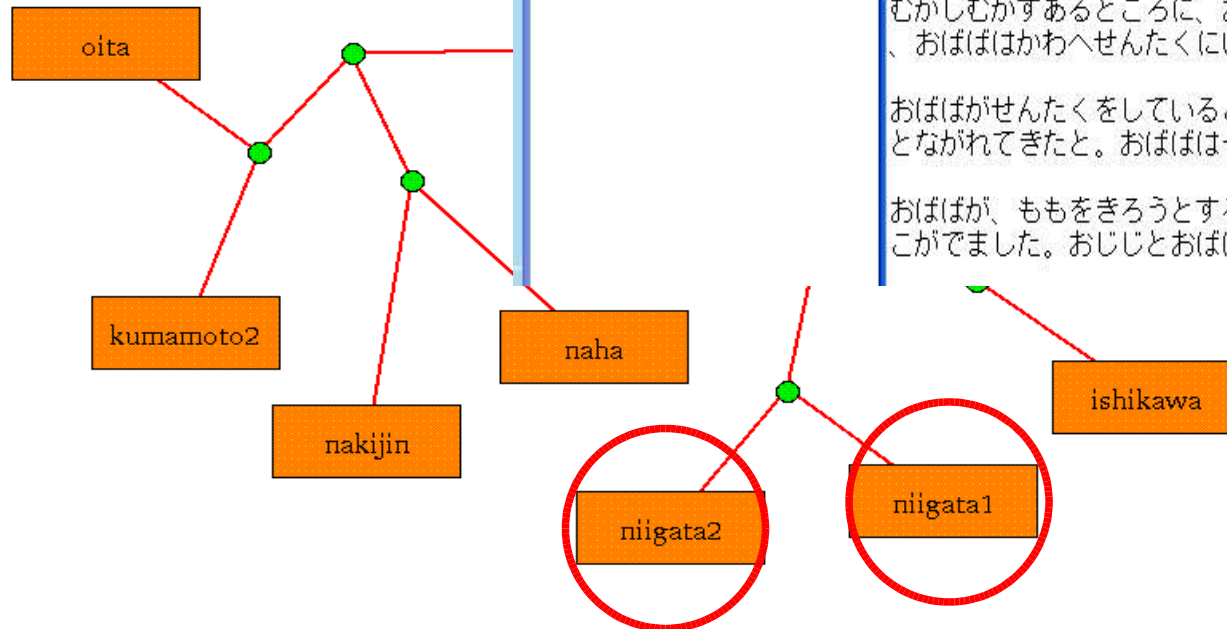
おばばがももをきろうとするとおももがふたつにわれ、なかからおおきなおとこのこがでました。おじじとおばばはそのこにももたろうというなまえをつけたとさ。

niigata 2

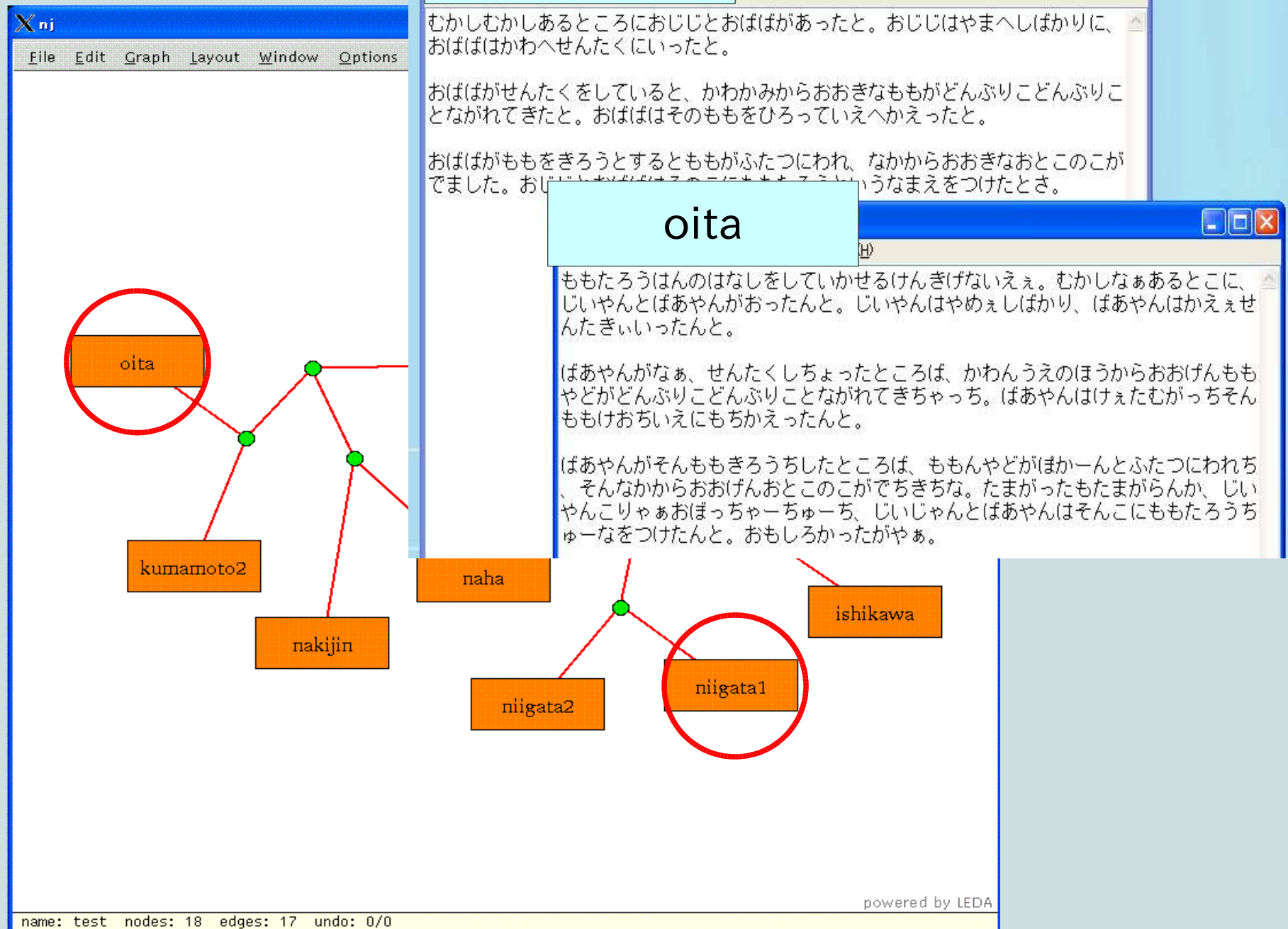
むかしむかしあるところに、おじじとおばばがあったと。おじじはやまへしばかりに、おばばはかわへせんたくにいったと。

おばばがせんたくをしていると、かわかみからおおきなももがどんぶりどんぶりことながれてきたと。おばばはそのももをひろっていえへかえったと。

おばばが、ももをきろうとするとおももがふたつにわれて、なかからおおきなおとこのこがでました。おじじとおばばはそのこにももたろうというなまえをつけたとさ。



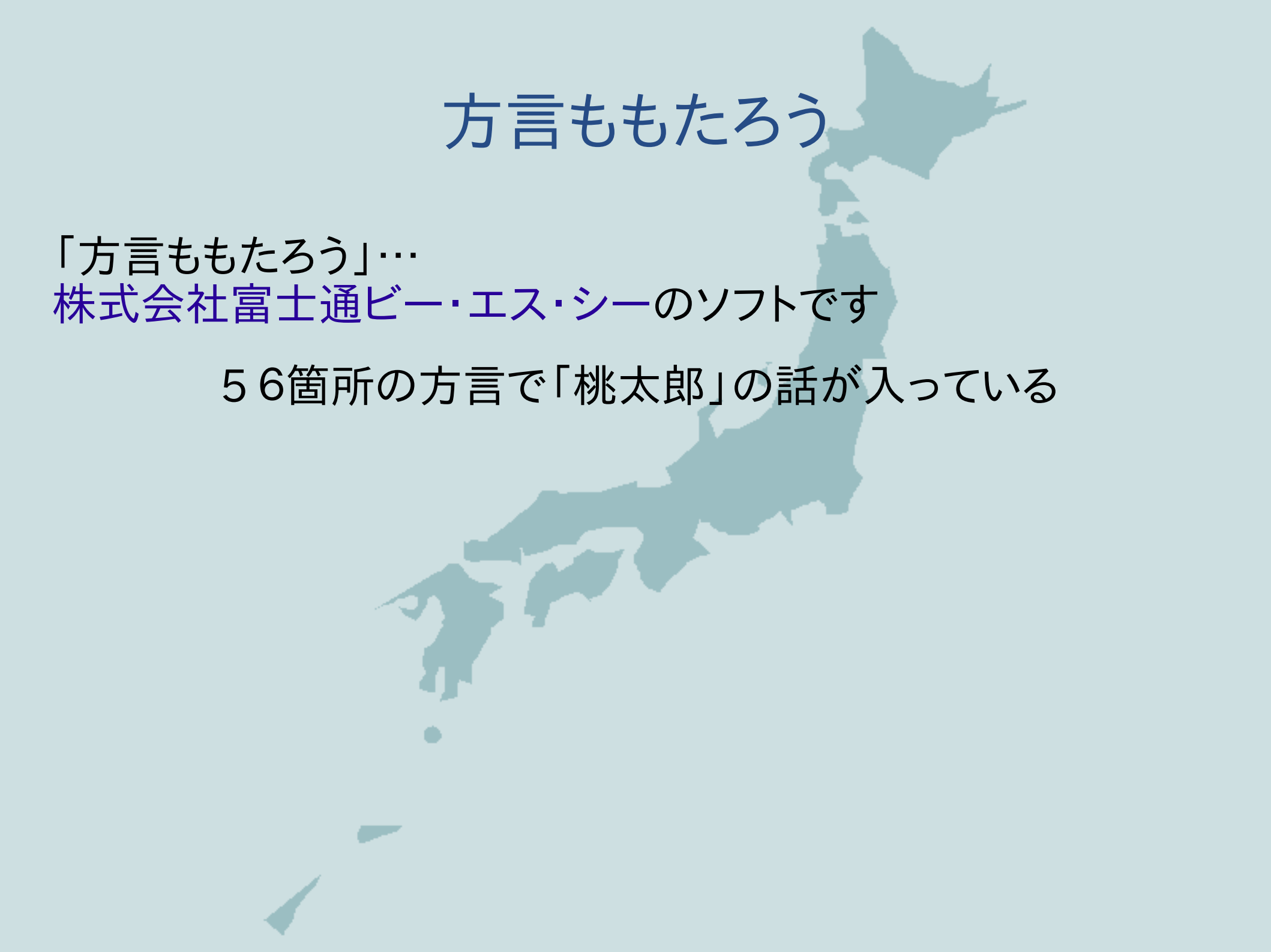
系統樹



目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

方言ももたろう



「方言ももたろう」…

株式会社富士通ビー・エス・シーのソフトです

56箇所の方言で「桃太郎」の話が入っている

方言ももたろう

標準語

むかしむかし あるところに おじいさんとおばあさんがいました

青森

むがしむがし あるところにおじいさんとあつていたと

岩手

むがすむがす さんがいたあと

宮城

むがしむがし あるところにおじいさんとおばんちゃんが
あつたとっさあ

那覇

むかしむかし あるところんかい たんめいとみめいが
めんせいにいたん

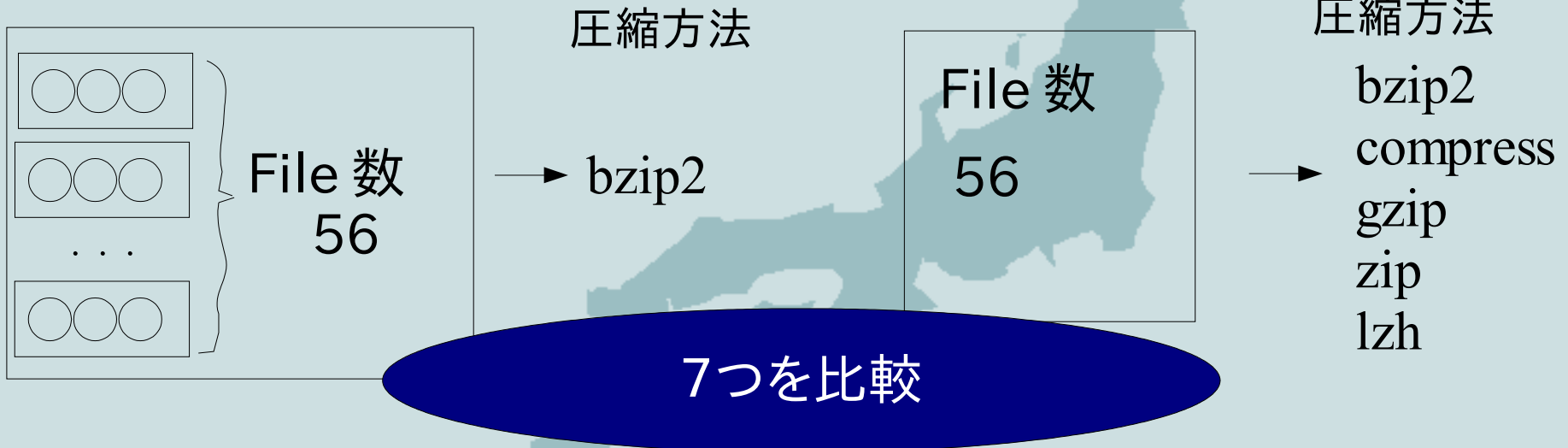
Kolmogorov 記述量で
類似性を調べよう!

方言ももたろう

対象データ

音声ファイル (WAV、mp3)

テキストファイル



系統樹作成法

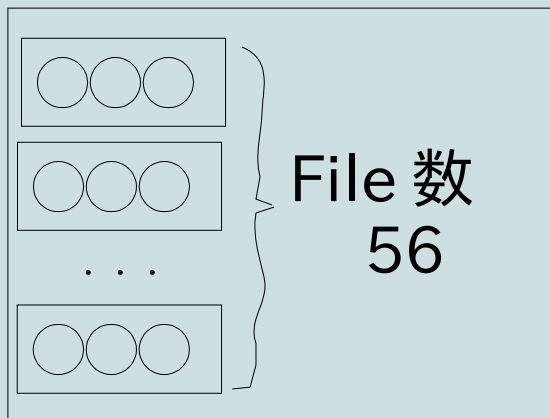
Nj法を用いて可視化

方言ももたろう

対象データ

音声ファイル (WAV、mp3)

テキストファイル



圧縮方法

→ bzip2



圧縮方法

→
bzip2
compress
gzip
zip
lzh

Kolmogorov 記述量による類似度に関する距離 d は

$$0 \leq d \leq 1$$

方言ももたろう

対象データ

音声ファイル (WAV、mp3)

テキストファイル



Kolmogorov 記述量による類似度に関する距離 d は

$$0 \leq d \leq 1$$

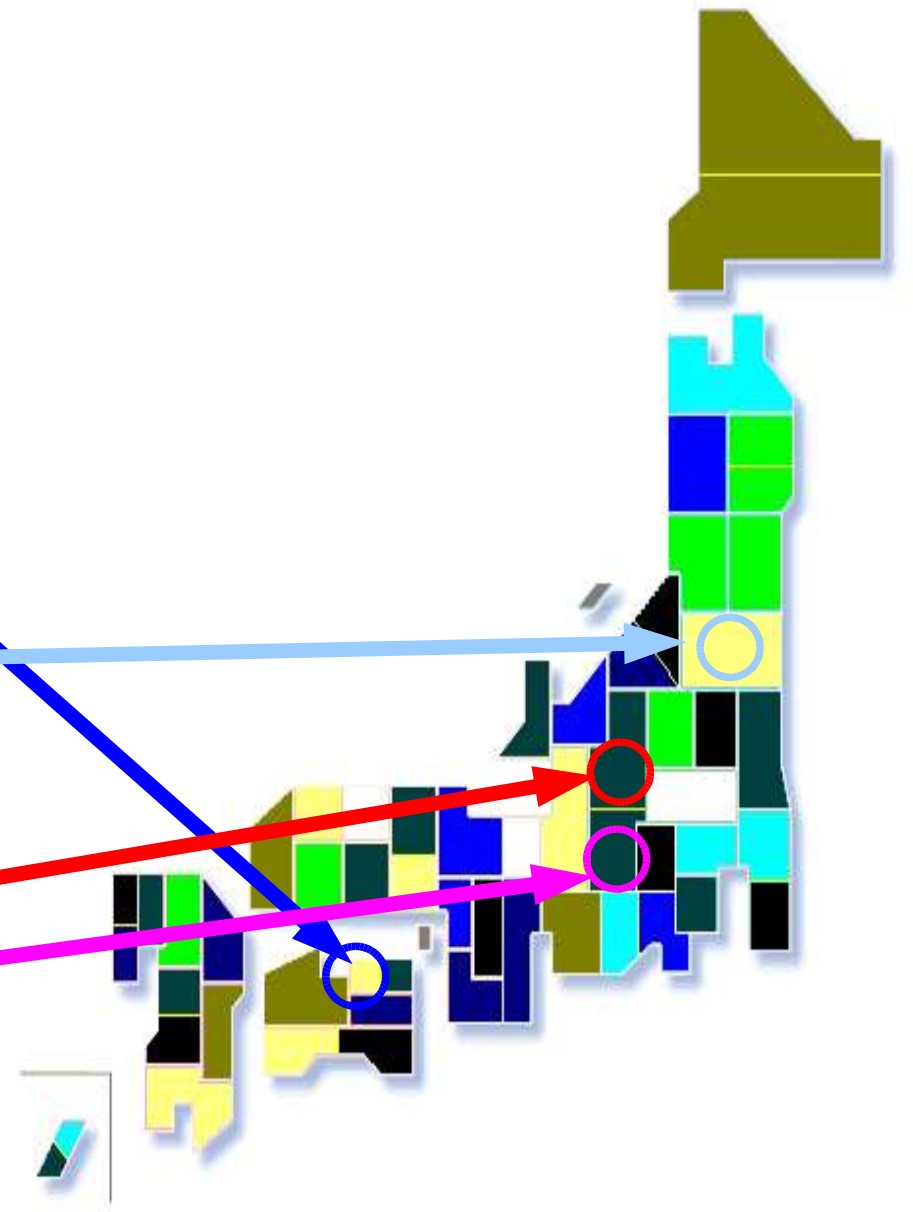
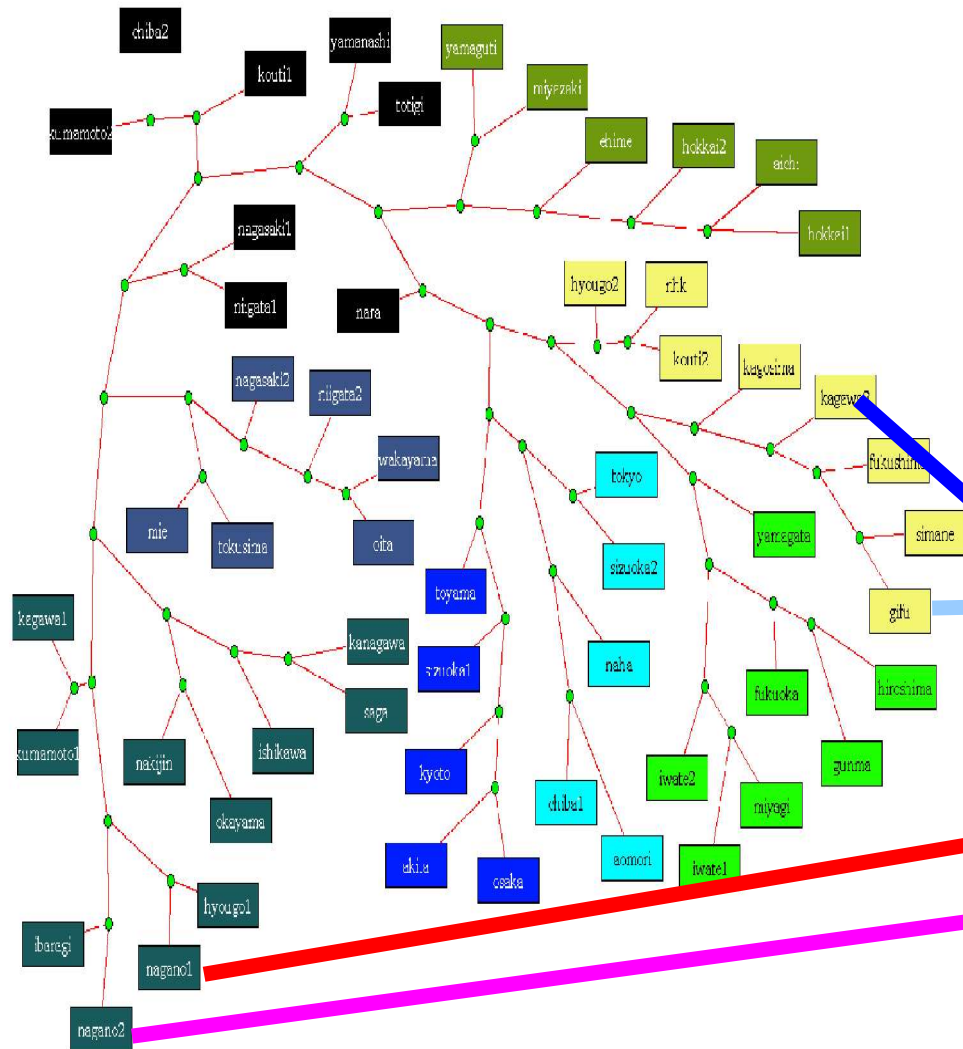
WAV, gzip, zip, lzh は d の範囲に当てはまらない

目次

- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

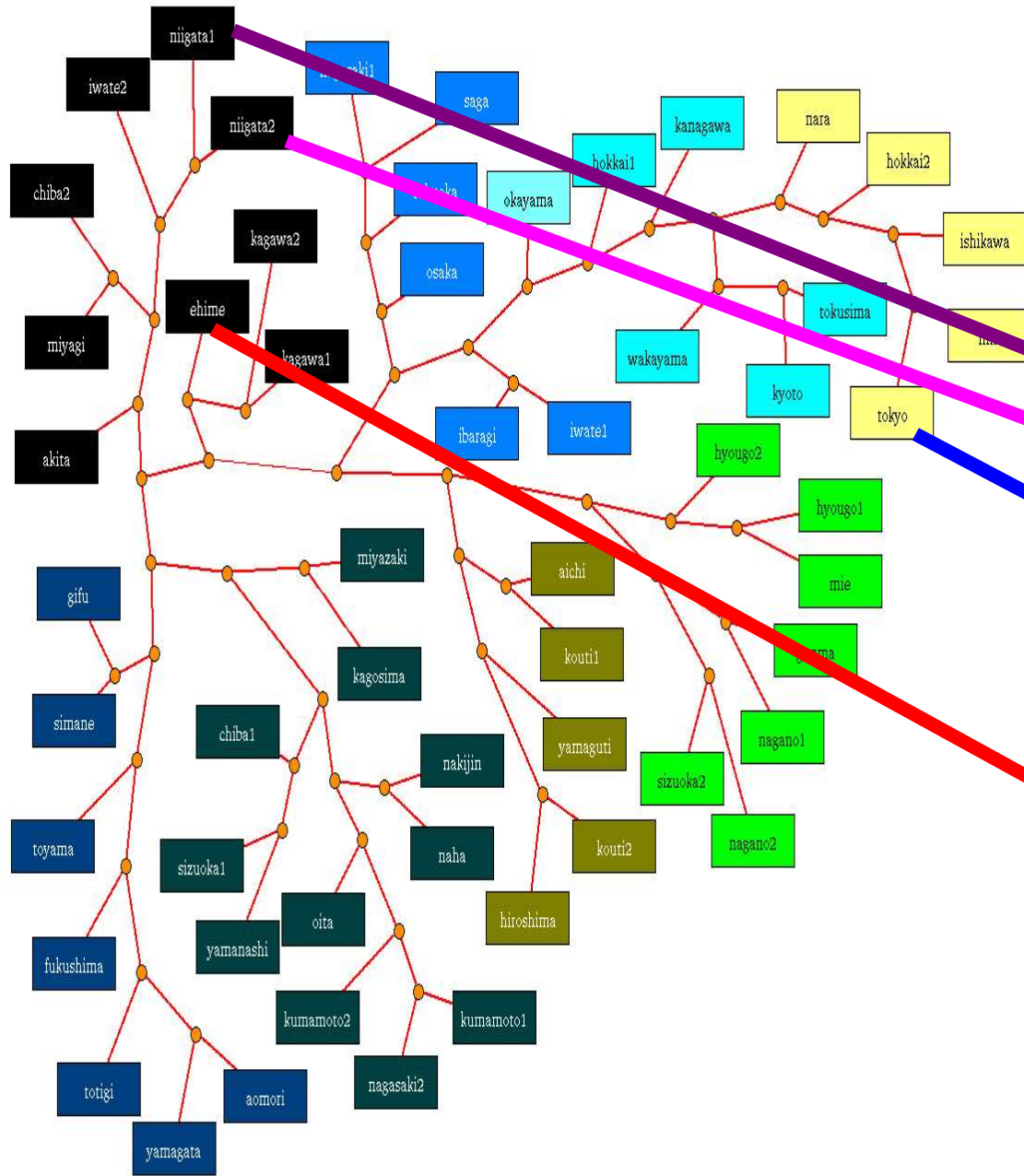
実験結果 音声ファイル (mp 3)

mp3



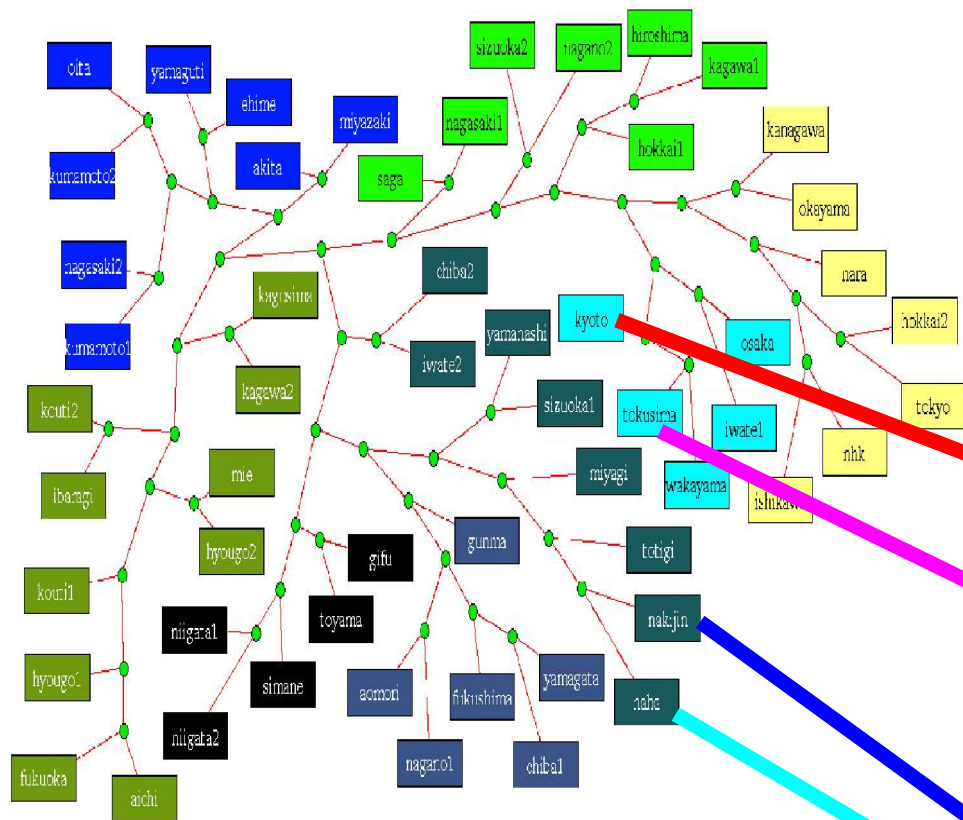
実験結果 テキストファイル (bzip2 , compress)

bzip2



実験結果 テキストファイル (bzip2 , compress)

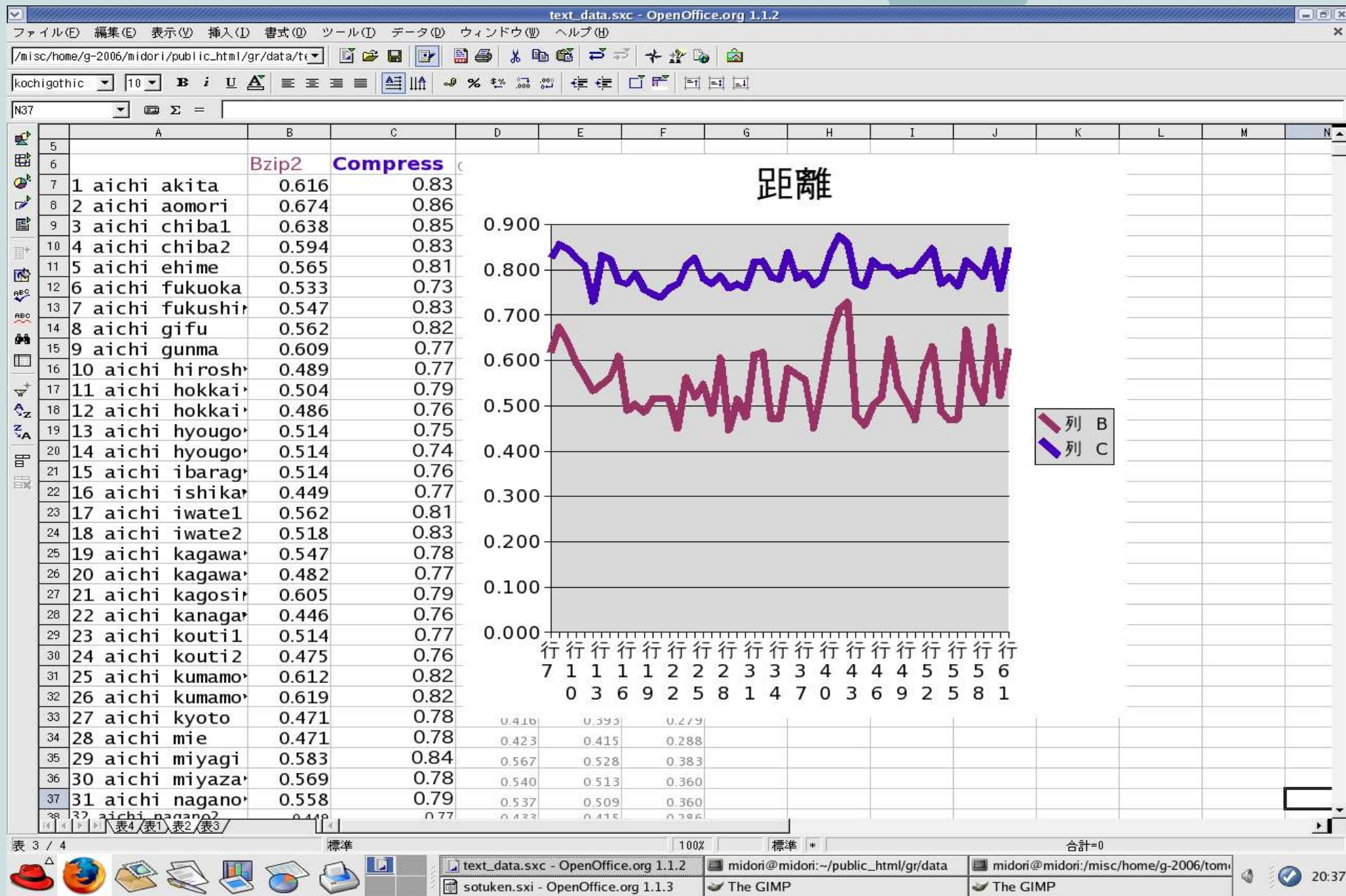
compress



bzip2 と compress の圧縮率の違い

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
4													圧縮率	
5		Txt	Bzip2	圧縮率	Compress*	圧縮率								Compress*
6	Aichi	538	276	48.70%	333	38.10%							0.49	0.
7	Akita	427	224	47.54%	258	39.58%							0.48	0.
8	Aomori	458	245	46.51%	284	37.99%							0.47	0.
9	Chiba1	396	224	43.43%	251	36.62%							0.43	0.
10	Chiba2	434	227	47.70%	261	39.86%							0.48	0.
11	Ehime	430	230	46.51%	268	37.67%							0.47	0.
12	Fukuoka	532	248	53.38%	313	41.17%								0.
13	Fukushima	412	227	44.90%	258	37.38%	55.00%							0.
14	Gifu	448	247	44.87%	276	38.39%								0.
15	Gunma	526	260	50.57%	325	38.21%	53.00%							0.
16	Hiroshima	472	242	48.73%	288	38.98%								0.
17	Hokkai1	458	232	49.34%	276	39.74%	51.00%							0.
18	Hokkai2	440	223	49.32%	267	39.32%								0.
19	Hyougo1	590	269	54.41%	336	43.05%	49.00%							0.
20	Hyougo2	554	255	53.97%	323	41.70%								0.
21	Ibaragi	487	243	50.10%	296	39.22%	47.00%							0.
22	Ishikawa	448	230	48.66%	269	39.96%								0.
23	Iwate1	470	235	50.00%	280	40.43%	45.00%							0.
24	Iwate2	439	225	48.75%	264	39.86%								0.
25	Kagawa1	460	243	47.17%	288	37.39%	43.00%							0.
26	Kagawa2	534	250	53.18%	308	42.32%								0.
27	Kagosima	496	248	50.00%	301	39.31%	41.00%							0.
28	Kanagawa	452	236	47.79%	276	38.94%								0.
29	Kouti1	536	275	48.69%	329	38.62%	39.00%							0.
30	Kouti2	482	246	48.96%	296	38.59%								0.
31	Kumamoto*	510	246	51.76%	296	41.96%	37.00%							0.
32	Kumamoto*	611	291	52.37%	355	41.90%								0.
33	Kyouto	448	235	47.54%	278	37.95%	35.00%							0.
34	Mie	536	259	51.68%	316	41.04%								0.
35	Miyagi	490	234	52.24%	282	42.45%								0.
36	Miyazaki	454	236	48.02%	279	38.55%								0.
37	Nagano1	508	253	50.20%	306	39.76%	行 行 行 行 行 行 行 行 行 行 行 行 行 行 行 行 行							0.
38	Nagano2	458	239	47.82%	284	37.99%	6 9 1 1 1 2 2 2 3 3 3 3 4 4 4 5 5 6 2 5 8 1 4 7 0 3 6 9 2 5 8 1 4 8 7 0							0.

実験結果 bzip2 と compress の距離の違い



考察

- bzip2 と compress の違いはみられるが、どちらが有用であるかはわからなかった



Kolmogorov に基づく類似度を表す距離は bzip2 で圧縮したものが良いといわれている

目次

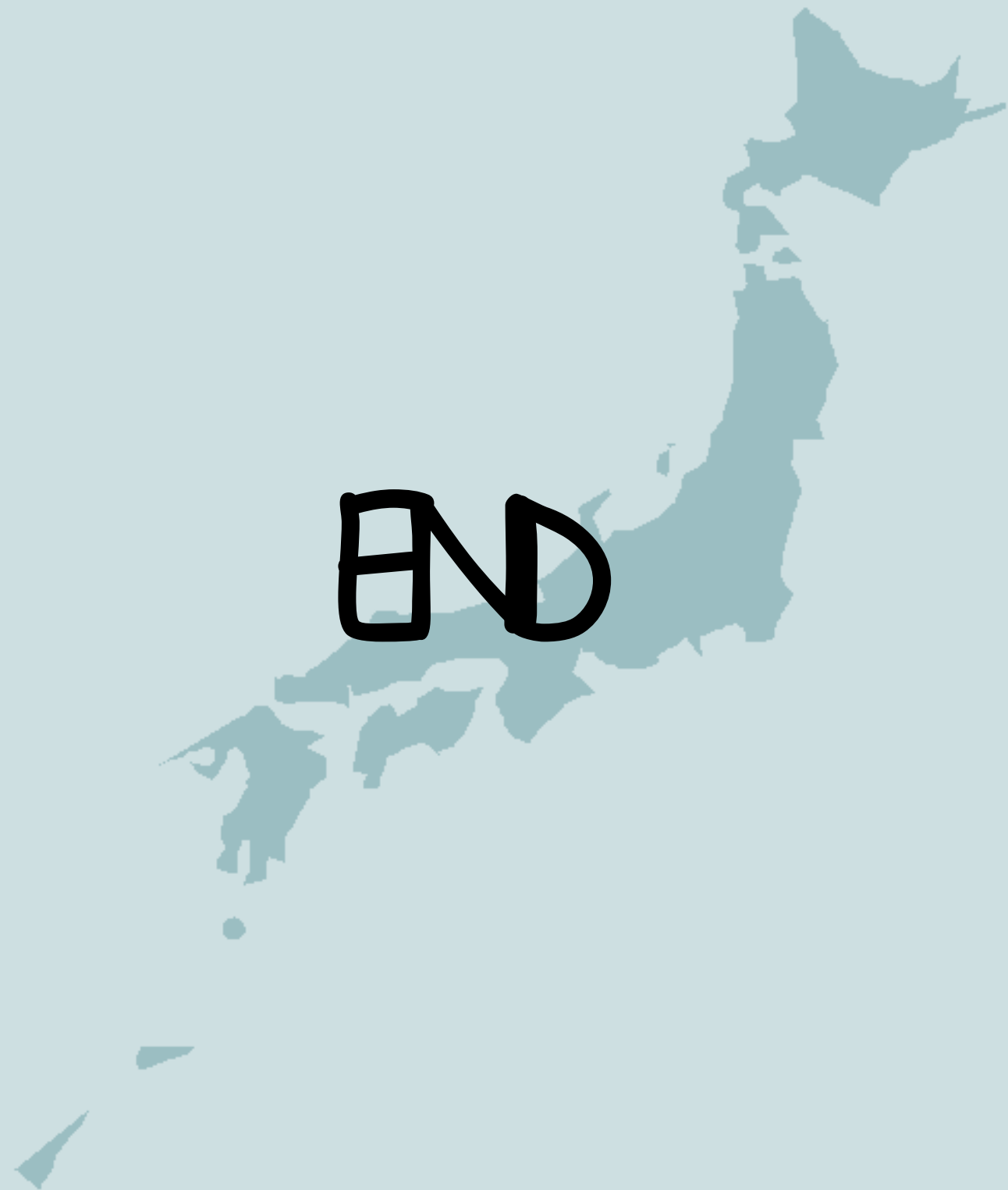
- 研究動機・目的
- 研究の背景
- 研究の流れ
 - × クラスタリング (NJ 法)
 - × 方言ももたろう
- 実験結果・考察
- 今後の目標

今後の目標

音声ファイルでの類似度がうまくでなかった



- WAV,mp3 以外の音声ファイルでも試してみる
- bzip2 以外の圧縮方法でも試してみる



Kolmogorov 記述量

データ x の Kolmogorov 記述量は
あるプログラミング言語で
 x を生成する最小のプログラムサイズ

Kolmogorov 記述量

データ x の Kolmogorov 記述量は
あるプログラミング言語で
 x を生成する最小のプログラムサイズ

どんなプログラミング言語を選んでもそれが妥当であれば、
情報量は定数分の差しかないことが証明されている

目次

- 研究動機・目的
- 研究の背景
- Kolmogorov 記述量
 - × 定義
 - × 距離
- 方言ももたろう
- 現状
- 目標

Kolmogorov 記述量の定義

S プログラムミング言語
 $|p|$... プログラムミングのサイズ

このとき、Kolmogorov 記述量は $K_s(x)$ を

$$K_s(x) = \min \{ |p| : S(p) = x \}$$

と定義する

Kolmogorov 記述量の定義

計算不能

$$K_s(x) = \min \{ |p| : S(p) = x \}$$

ただし x を生成する p が存在しないときは

$$K_s(x) = \infty$$

と定義される

目次

- 研究動機・目的
 - 研究の背景
 - Kolmogorov 記述量
 - × 定義
 - × 距離
 - 方言ももたろう
 - 現状
 - 目標
- 

距離の定義

X : 集合

D : 距離関数

D が X 上の距離であるとは、任意の X の元 x 、 y 、 z に対して以下が成り立つときをいう。

$$x = y \text{ ならば、} D(x, y) = 0$$

$$D(x, y) + D(y, z) \geq D(x, z)$$

$$D(x, y) = D(y, x)$$

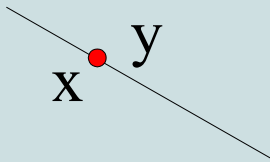
距離の定義

X : 集合

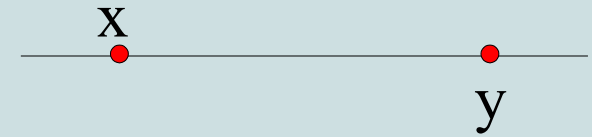
$x, y, z : X$ の元

D : 距離関数

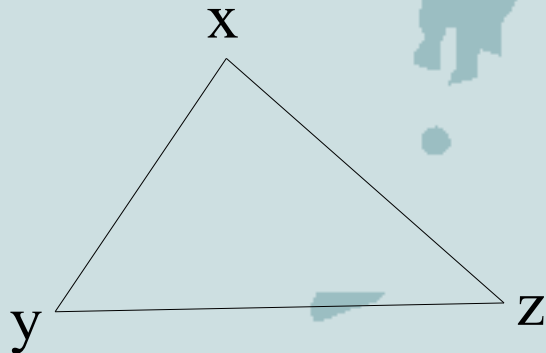
$x = y$ ならば、 $D(x, y) = 0$



$D(x, y) = D(y, x)$



$D(x, y) + D(y, z) \geq D(x, z)$



条件付き Kolmogorov 記述量

U : 固定した万能言語

$|p|$: プログラムミングのサイズ

補助情報 y に対するデータ x の Kolmogorov 記述量 $K(x|y)$

$$K(x|y) = \min \{ |p| : U(p, y) = x \}$$

y が x に関する情報を含んでいれば、

$$K(x|y) < K(x)$$

条件付き Kolmogorov 記述量

$K(x)$: x の情報量

$K(x|y)$: x を用いた y の記述量

c : ある定数

y に含まれる x の情報量を $I(x : y)$ とする。

$$I(x : y) = K(x) - K(x|y)$$

$$I(x : y) - I(y : x) \leq c \longrightarrow I(x : y) = I(y : x)$$

Kolmogorov 記述量に基づく距離

$K(x)$: x の情報量

$K(xy)$: x と y を結合した情報量

$K(y) > K(x)$ のとき、

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

このままでは計算できない!
bzip2 に置きかえる。

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

Kolmogorov 記述量のメリット

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

圧縮したファイルのサイズによって類似度を表す距離が簡単に求められる。

現状

作ってあるもの

- ××.txt
○○.WAV

- ××.txt
○○.txt

→

××.txt.bz2
○○.WAV.bz2

1、ファイルサイズを書き出す

2、ファイルサイズを書き出す

1.txt.bz2

1~3080

1.txt.bz2 ××.txt ○○.txt

3、リスト作成

- 類似度を表す距離を計算するプログラム