平成19年 2月 10日 (土)

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　　関根　康人
高野　浩明
堀中　幸司

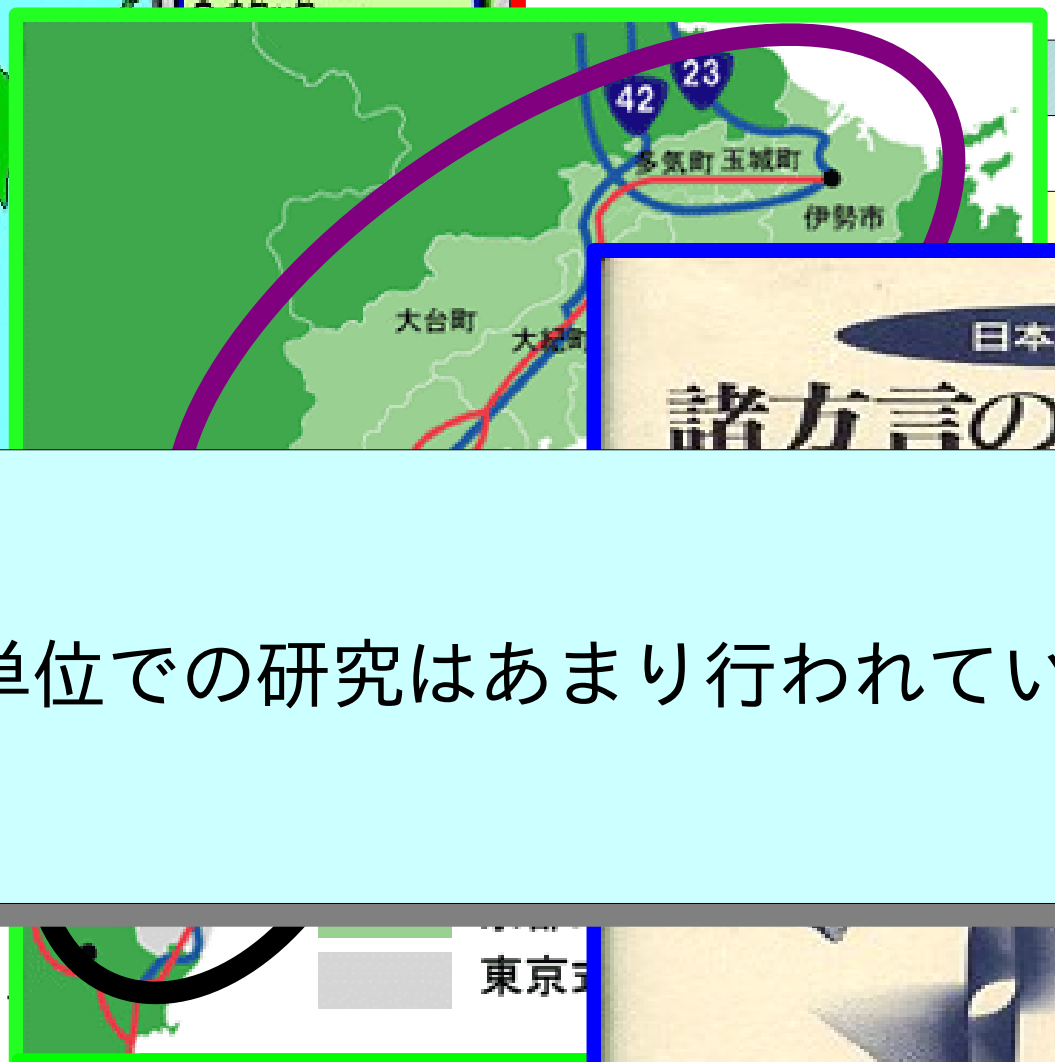
目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

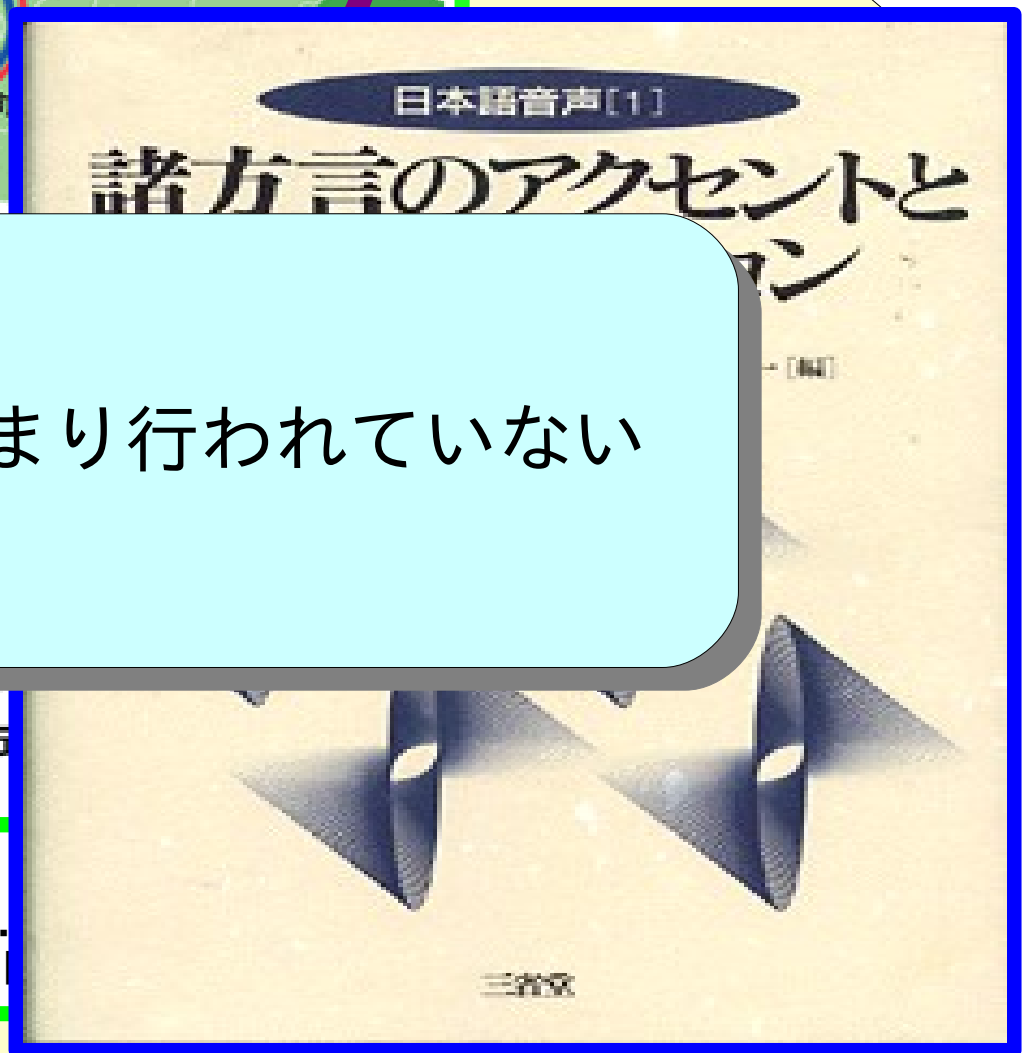
目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

引用元
http://www.bsc.fuji
n/alacarte.html



引用元
http://www.cbr.mlit.go.
mano_hougen/hogen2.html



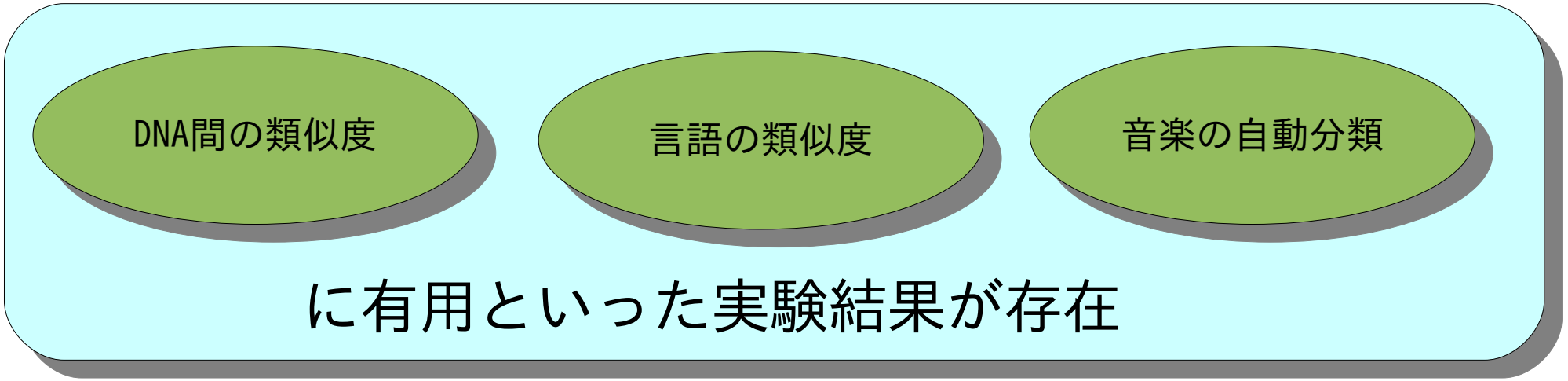
引用元http://www.sanseido-
publ.co.jp/publ/jap_1oto.html

文章単位での研究はあまり行われていない

背景

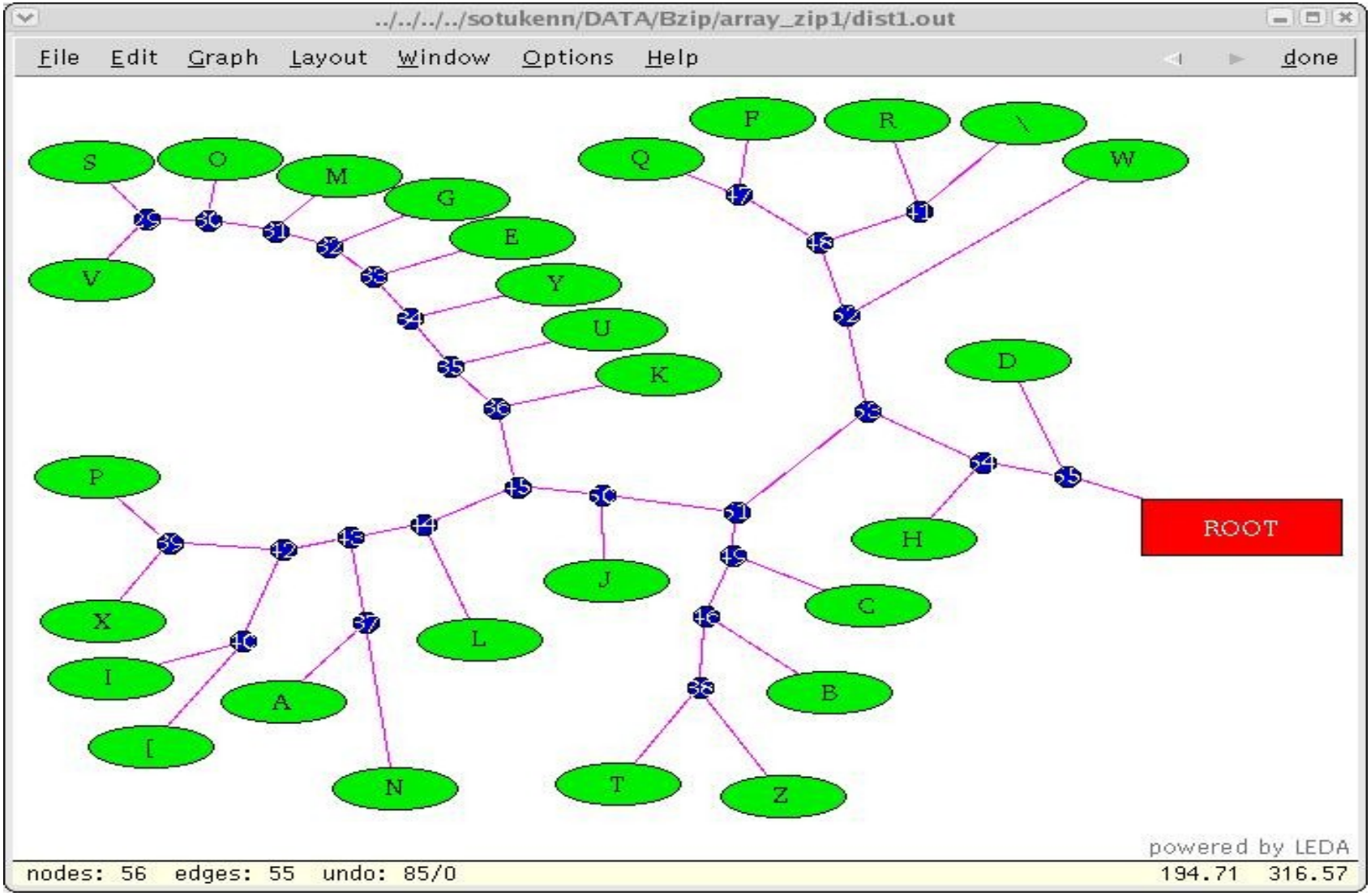
similarity metric

- Ming Liらによって提案(2003)
- Kolmogorov 記述量を用いて文字列間の類似度距離を表す



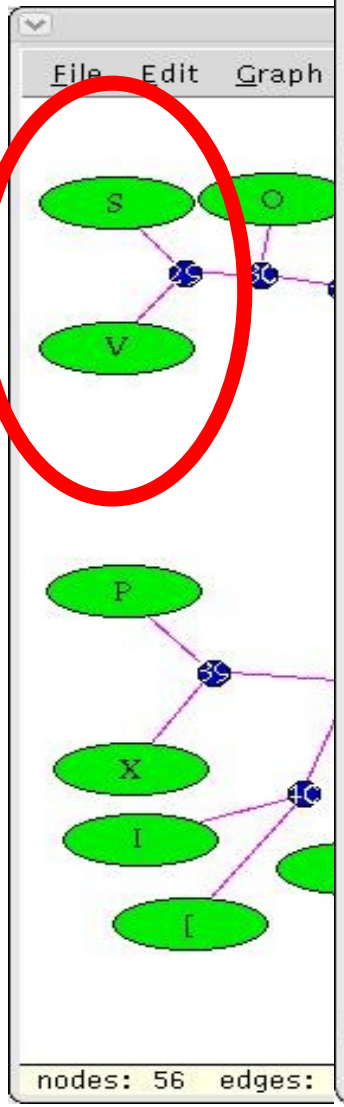
背景

利用例：生徒のプログラム課題の中から複製、提出されたものを見つけ出す

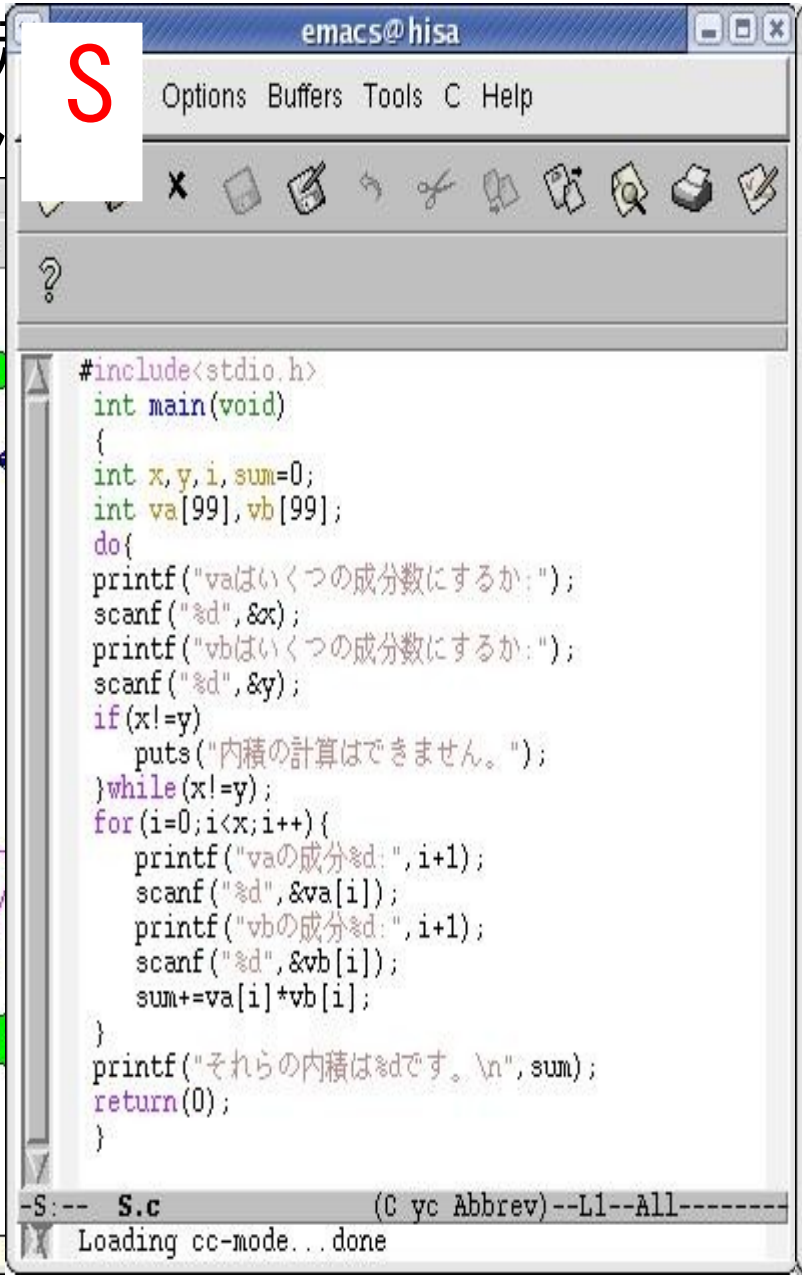


背景

利用例：生徒
ない



S



V

背景

similarity metric

- Ming Liらによって提案(2003)

昨年、大江氏による文字列間の類似度を用いた方言の分類

に有用といった実験結果が存在

目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

先行研究

大江氏による先行研究…

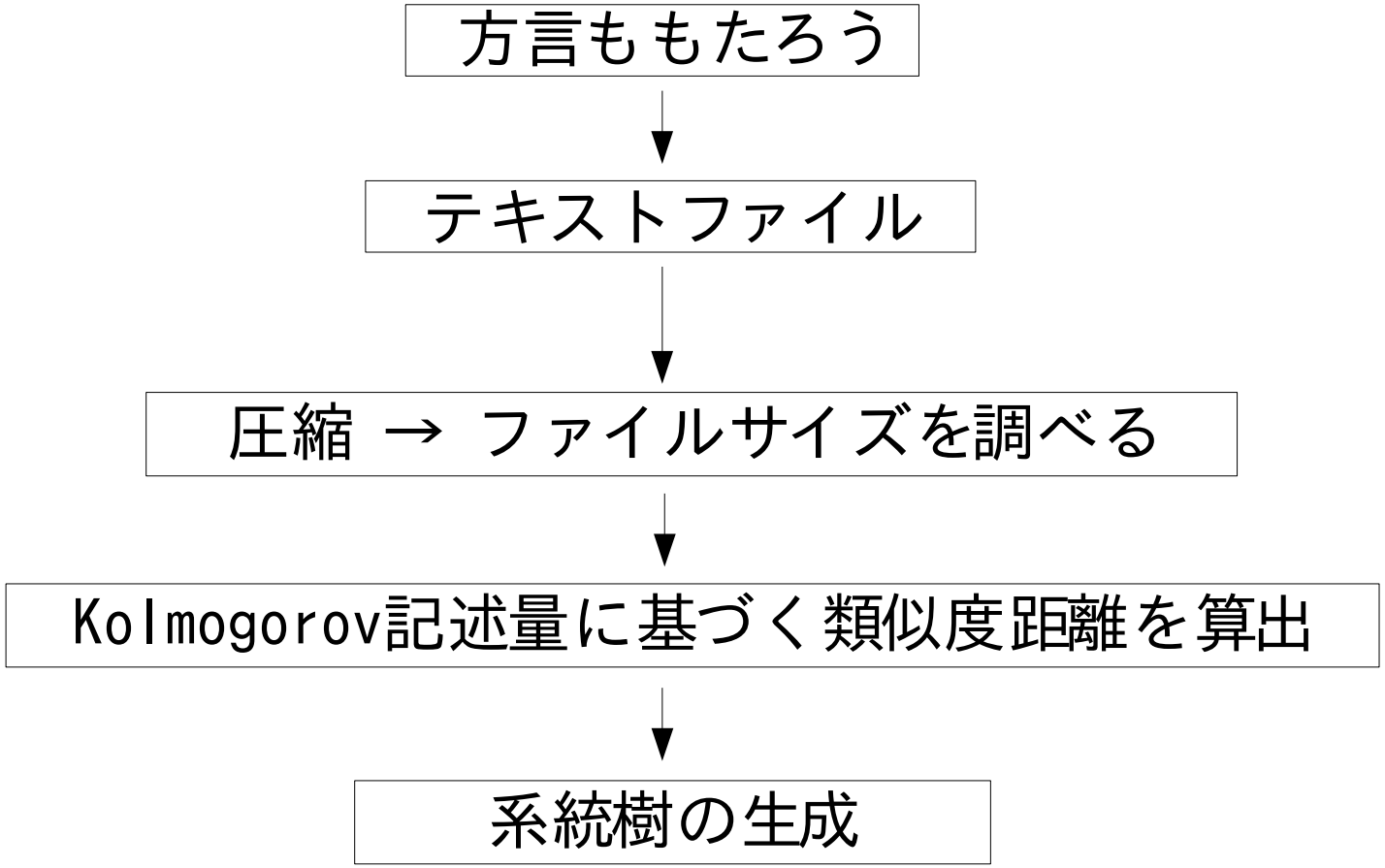
・方言ももたろう

監修・著：杉藤　美代子

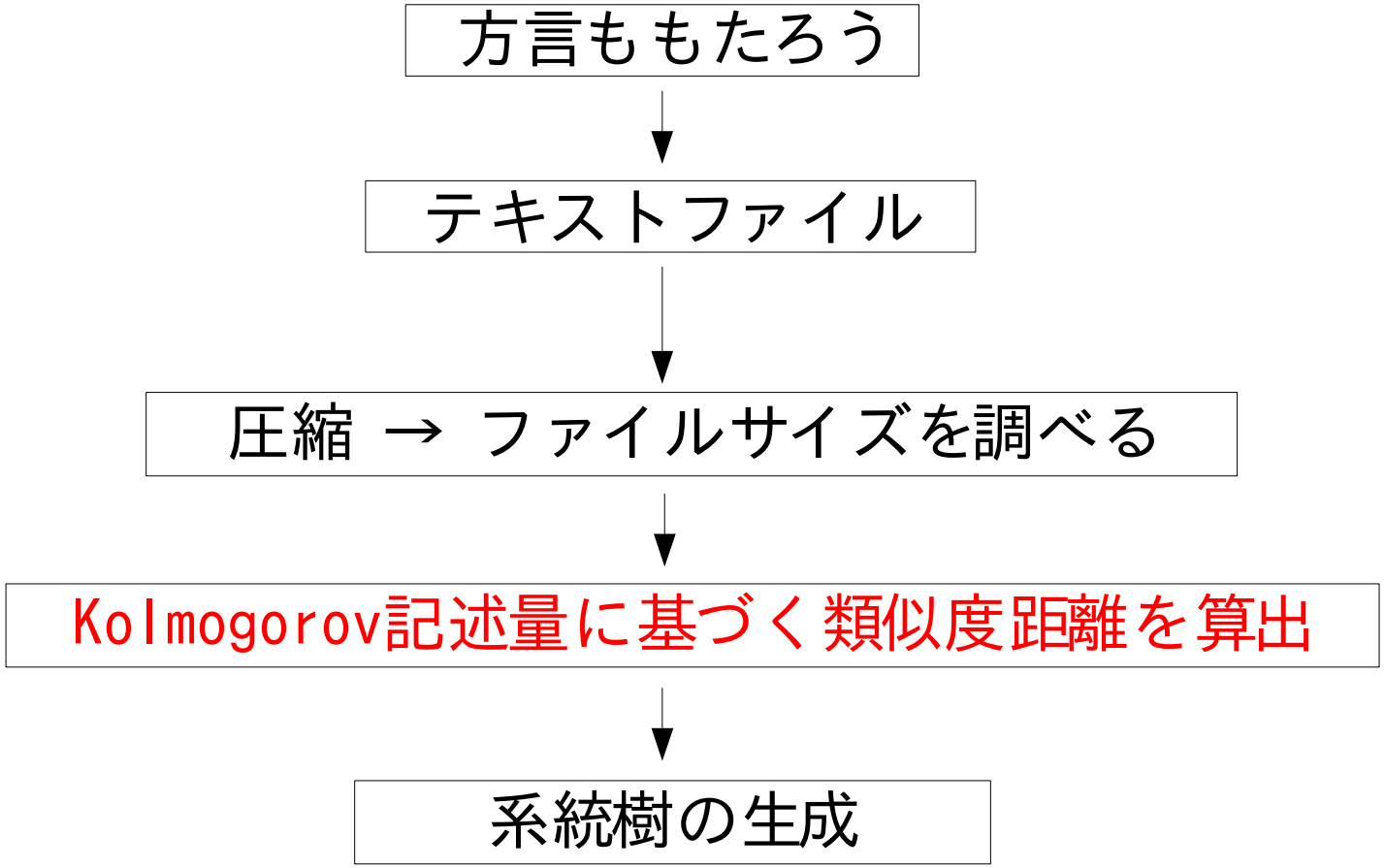
昔話「桃太郎」を全国56地方の各方言による語り（音声データ）を収録してあるソフトウェア

引用元
http://www.bsc.co.jp/nihongo/momotaro/

先行研究


方言ももたろう
テキストファイル
圧縮 → ファイルサイズを調べる
Kolmogorov記述量に基づく類似度距離を算出
系統樹の生成

先行研究


方言ももたろう
テキストファイル
圧縮 → ファイルサイズを調べる
Kolmogorov記述量に基づく類似度距離を算出
系統樹の生成

Similarity metric

Ming Liらの研究で類似度距離が提案された

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

Kolmogorov 記述量　$K(x)$

データxを究極の方法で圧縮した時のサイズ

Similarity metric

Ming Liらの研究で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

$$d(x,y)=\frac{K(xy)-K(x)}{K(y)} \quad \Rightarrow \quad d(x,y) \approx \frac{bzip2(xy)-bzip2(x)}{bzip2(y)}$$

先行研究

対象データ（方言ももたろう）

テキストファイル

北海道
青森
…
沖縄

File数
56

圧縮方法

bzip2
compress
gzip
zip
lzh

5つの圧縮サイズを比較

目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

研究動機

先行研究　対象：方言ももたろう

大江氏の聞き取りにより作成されたテキストファイル
をデータ間の類似度を用いて自動的に分類

本研究　対象：方言ももたろう　　方言の読本

1.音声データを文字におこす際の揺らぎ
2.文字コードによる違い
3.系統樹作成法の選択
4.「方言の読本」
5.木の評価

目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

TXT化の時点での揺らぎ

方言ももたろう → TXTファイル

音声データをカナでTXT化する時点で方言のアクセントやイントネーション、半濁点等の情報が落ちてしまう

今回のメンバー（堀中・関根・高野）で再度TXT化

TXT化の時点での揺らぎ

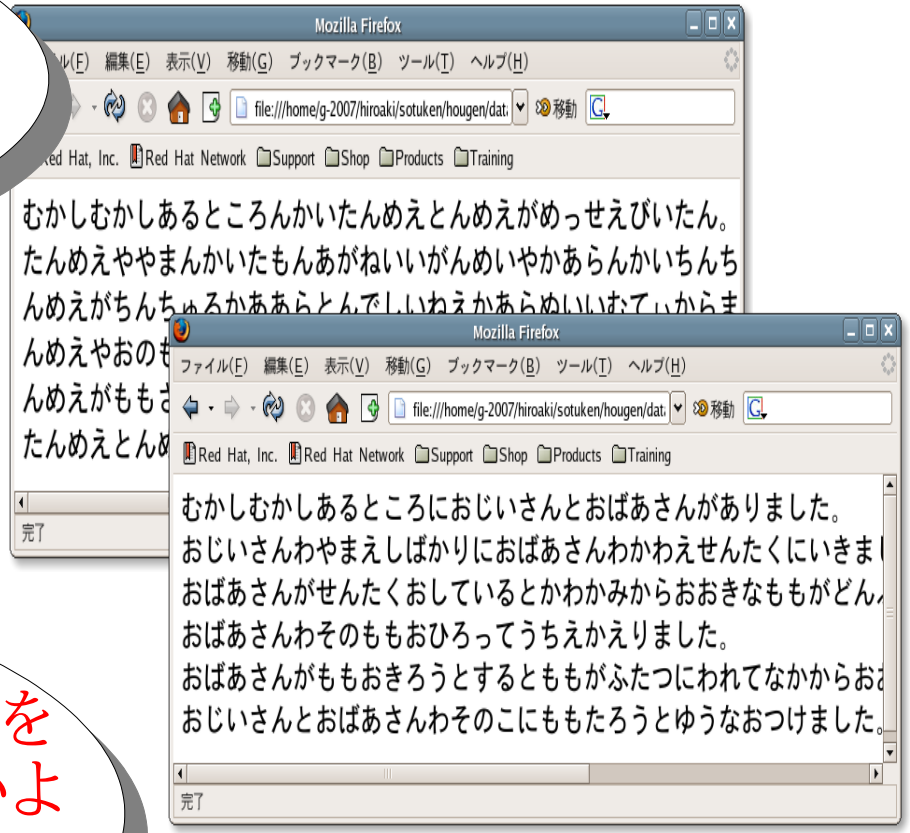
ファイルサイズ	Hori	Sekine	Takano	Midori
aichi	782	789	782	782
akita	615	633	599	612
aomori	653	654	653	654
chiba1	606	618	575	581
chiba2	624	648	647	636
ehime	648	648	624	627
fukuoka	789	792	788	783
fukushima	593	606	611	603
gifu	622	663	608	645
gunma	735	765	737	774

同じ音声を聞いて打ち込んでいるはずなのに聞き取り者によって大きくファイルサイズが違っている地域がある

TXT化の時点での揺らぎ

音としてのデータとして打込む。
「は」→「わ」「を」→「お」
「へ」→「え」

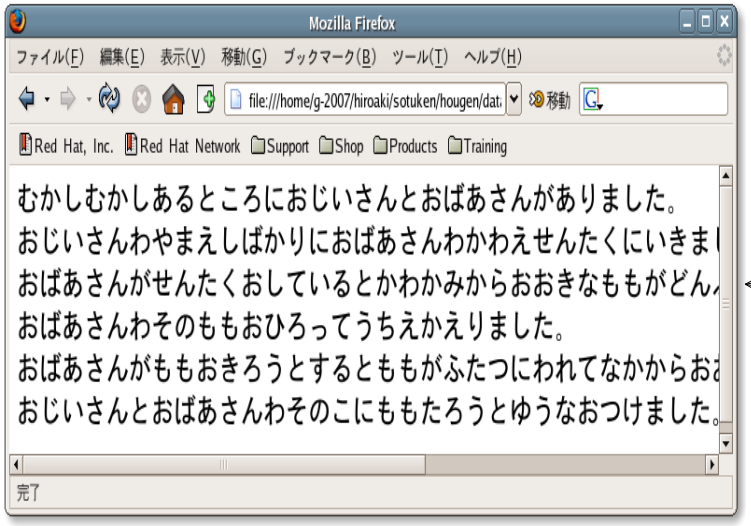
方言ももたろう

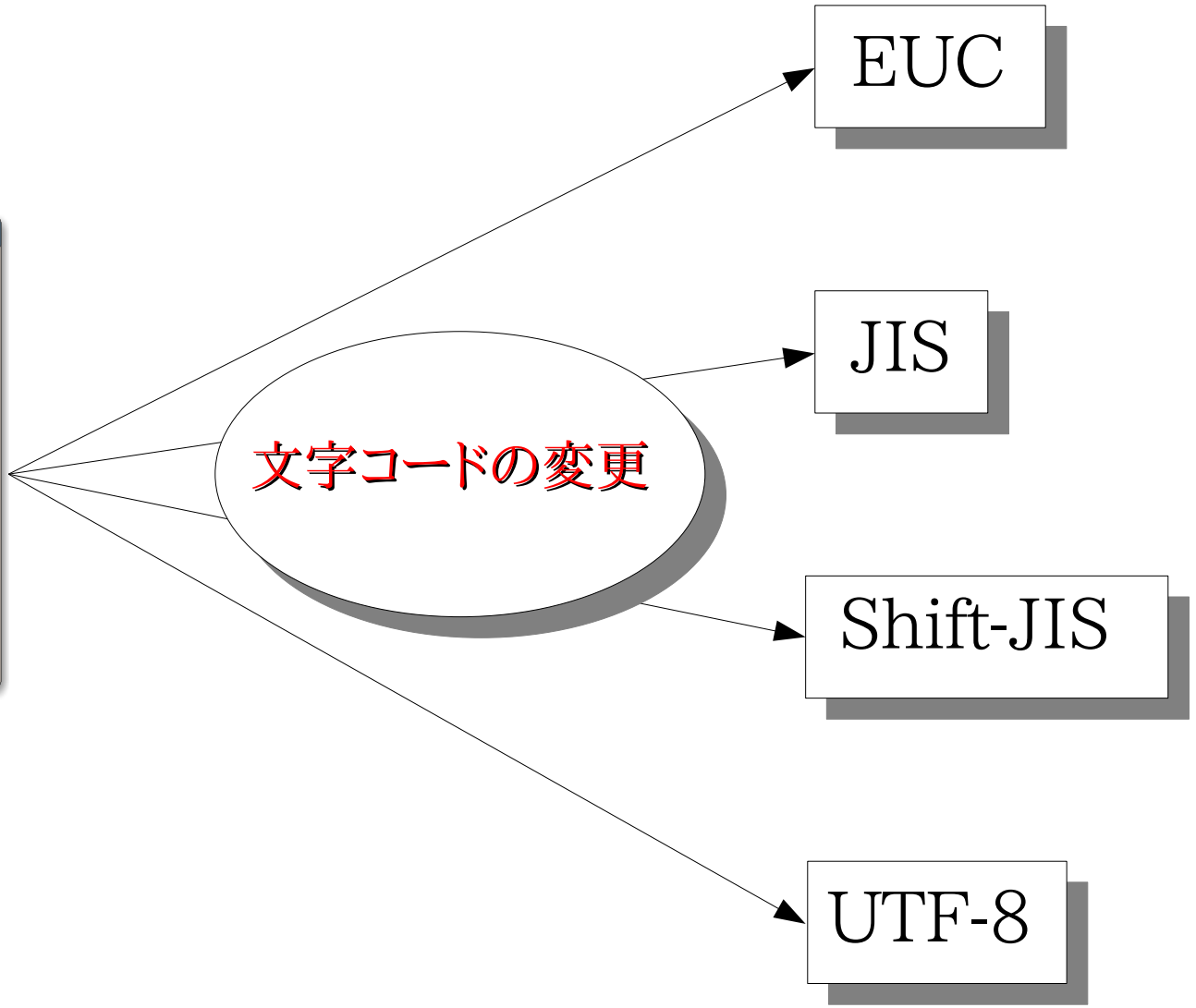


再度、国文学科の先生のアドバイスを受け、なるべく元のデータを逃さないように全員で再度聞き直し

このデータを使って研究を進めていく

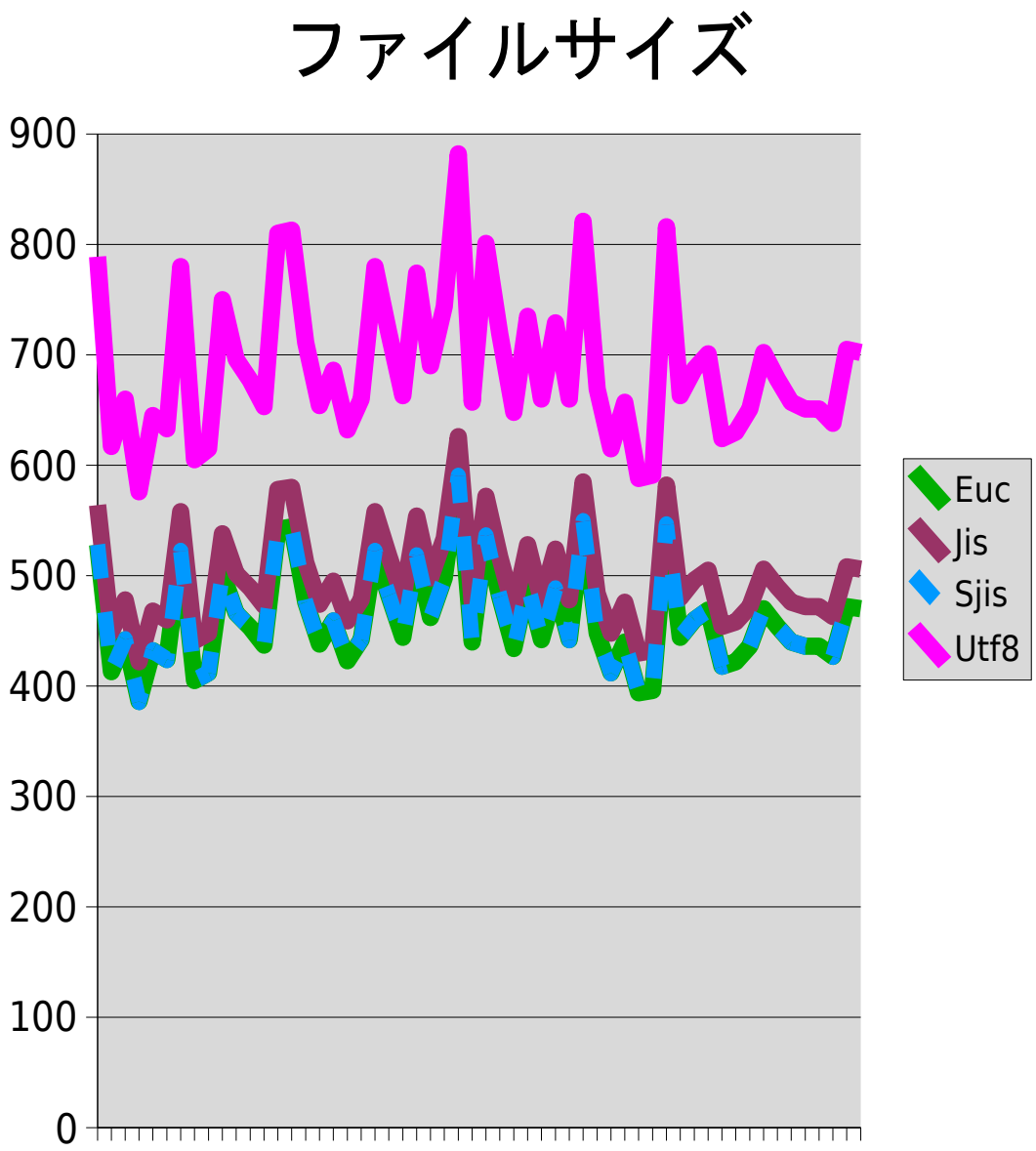
文字コードによる違い


Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)
file:///home/g-2007/hiroaki/sotuken/hougen/dat
移動
Red Hat, Inc. Red Hat Network Support Shop Products Training
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。
おじいさんわやまえしばかりにおばあさんわかわえせんたくにいきま
おばあさんがせんたくおしているとかわかみからおおきなももがどん
おばあさんわそのももおひろってうちえかえりました。
おばあさんがももおきろうとするとももがふたつにわれてなかからお
おじいさんとおばあさんわそのこにももたろうとゆうなおつけました。
完了


文字コードの変更
EUC
JIS
Shift-JIS
UTF-8

文字コードによる違い

Filesize	Euc	Jis	Sjis	Utf8
aichi	528	564	528	789
akita	413	449	413	617
aomori	442	478	442	660
chiba1	386	422	386	576
chiba2	432	468	432	645
ehime	424	460	424	633
fukuoka	522	558	522	780
fukushima	405	441	405	605
gifu	412	448	412	615
gunma	502	538	502	750
hiroshima	466	502	466	696
hokkaido1	453	489	453	677
hokkaido2	437	473	437	653
hyogo1	542	578	542	810
hyogo2	544	580	544	813
ibaraki	476	512	476	711
ishikawa	438	474	438	654



文字コードによる違い

NHKからの距離	Euc	Jis	Sjis	Utf8
aichi	0.43939	0.44689	0.41791	0.42537
akita	0.55263	0.55230	0.52361	0.50862
aomori	0.63025	0.61200	0.62979	0.57563
chiba1	0.60092	0.58091	0.57870	0.53017
chiba2	0.47664	0.43172	0.45833	0.43103
ehime	0.47964	0.47458	0.49770	0.44397
fukuoka	0.49138	0.47200	0.51852	0.44841
fuku				
gifu				
gunma				

どれが有用かを判断出来ない
余分な情報を取り除きたい→前処理

前処理

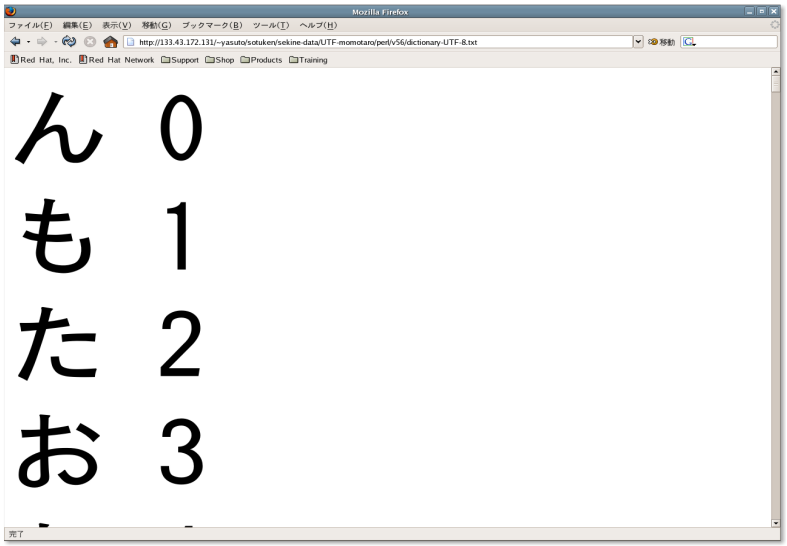
前処理1

あいうえお順に番号付け

前処理2

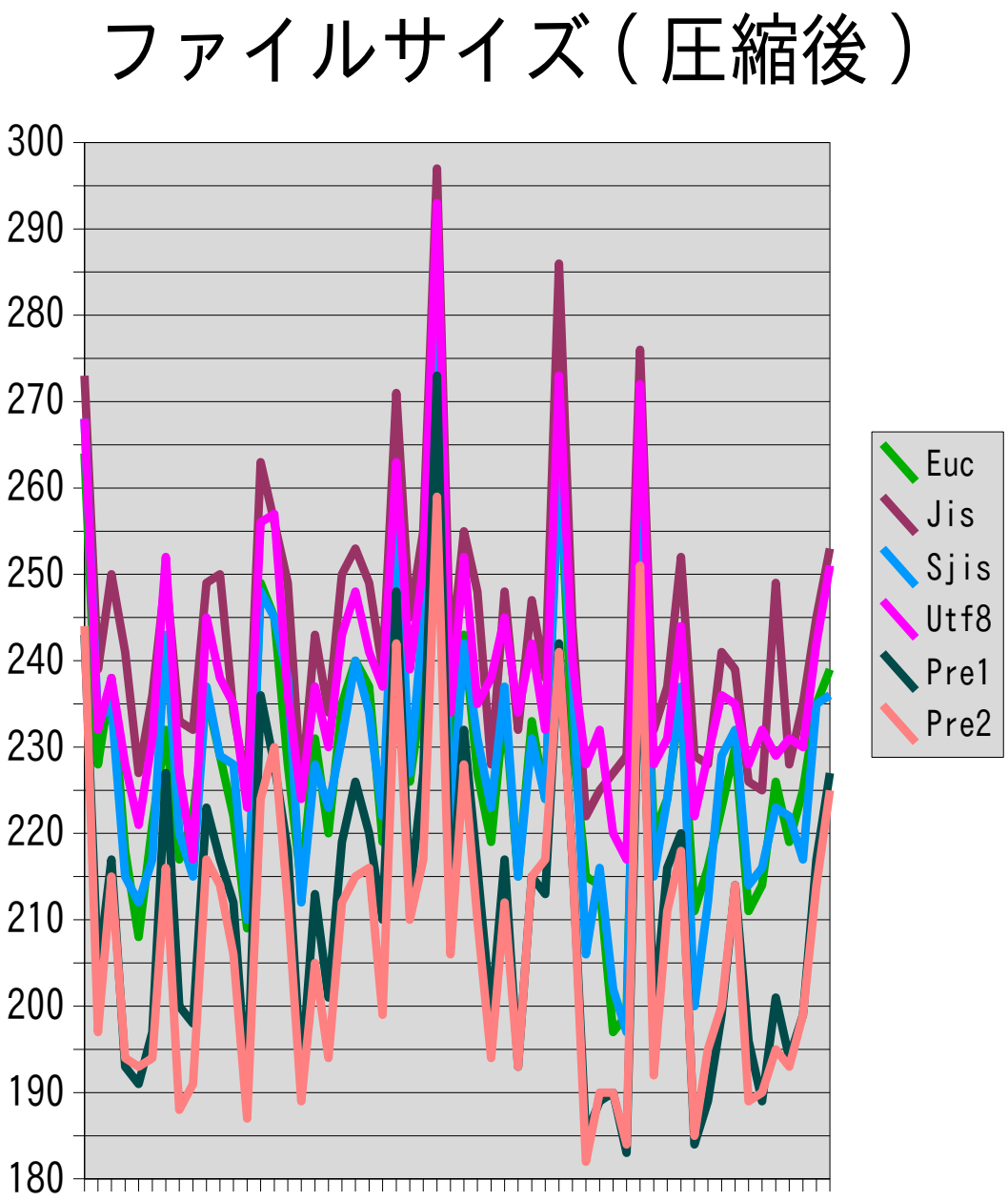
全てのデータの文字の出現数のカウント、ソート、番号付け

作成した1バイトのコードに変換



文字コードによる違い

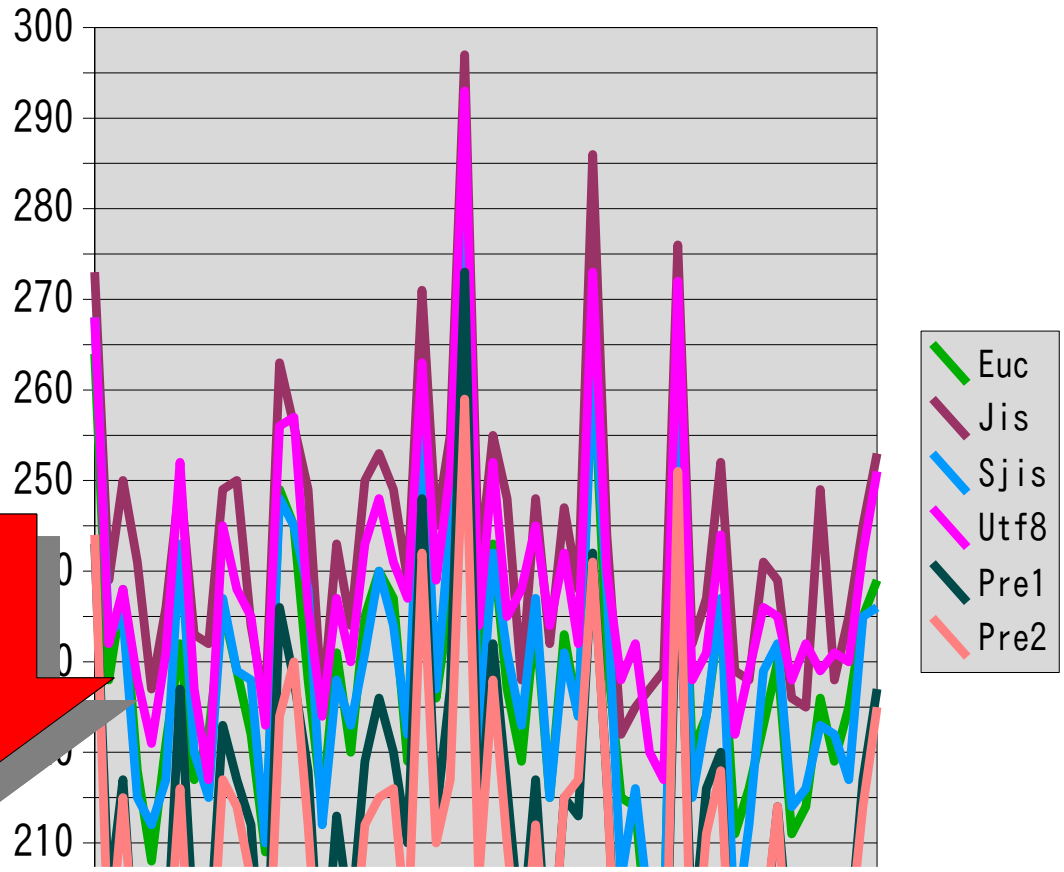
	Euc	Jis	Sjis	Utf8	Pre1	Pre2
aichi	264	273	268	268	243	244
akita	228	239	233	232	204	197
aomori	238	250	235	238	217	215
chiba1	218	241	215	228	193	194
chiba2	208	227	212	221	191	193
ehime	221	236	217	231	197	194
fukuoka	232	250	243	252	227	216
fukushima	217	233	220	227	200	188
gifu	222	232	215	217	198	191
gunma	236	249	237	245	223	217
hiroshima	229	250	229	238	217	214
hokkaido1	222	234	228	235	212	206
hokkaido2	209	225	210	223	190	187
hyogo1	249	263	248	256	236	224
hyogo2	245	256	245	257	228	230
ibaraki	228	249	238	236	218	212
ishikawa	214	225	212	224	192	189



文字コードによる違い

	Euc	Jis	Sjis	Utf8	Pre1	Pre2
aichi	264	273	268	268	243	244
akita	228	239	233	232	204	197
aomori	238	250	235	238	217	215
chiba1	218	241	215	228	193	194
chiba2	208	227	212	221	191	193
ehime	221	236	217	231	197	194
fukuoka	232	250	243	252	227	216
fukushima	217	233	220	227	200	188
gifu	222	232	215	217	198	191
gunma	236	249	237	245	223	217
hiroshima	229	250	229	238	217	2[illegible]
hokkaido1	222	234	228	235	212	206
hokkaido2	209	225	210	223	190	187
hyo						
hyo						
ibar						
ishikawa	214	229	212	224	192	189

ファイルサイズ（圧縮後）



前処理2を用い研究をすすめる

目次

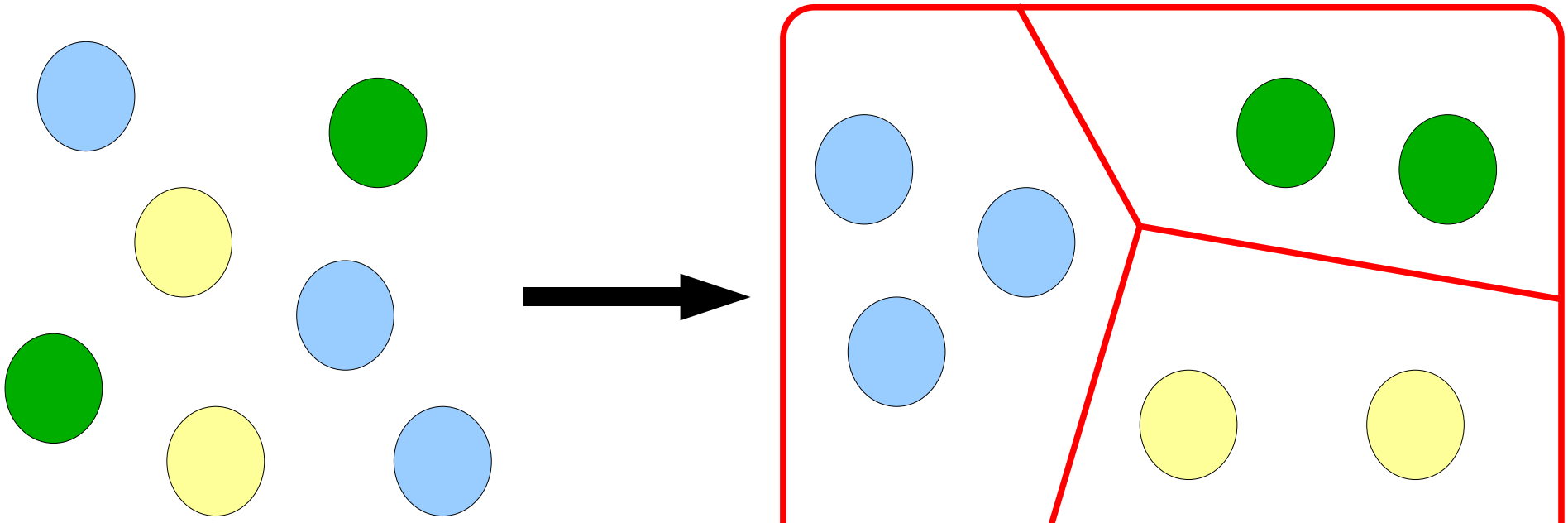
- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

系統樹作成法の選択

クラスタリング…

分類する対象の集合

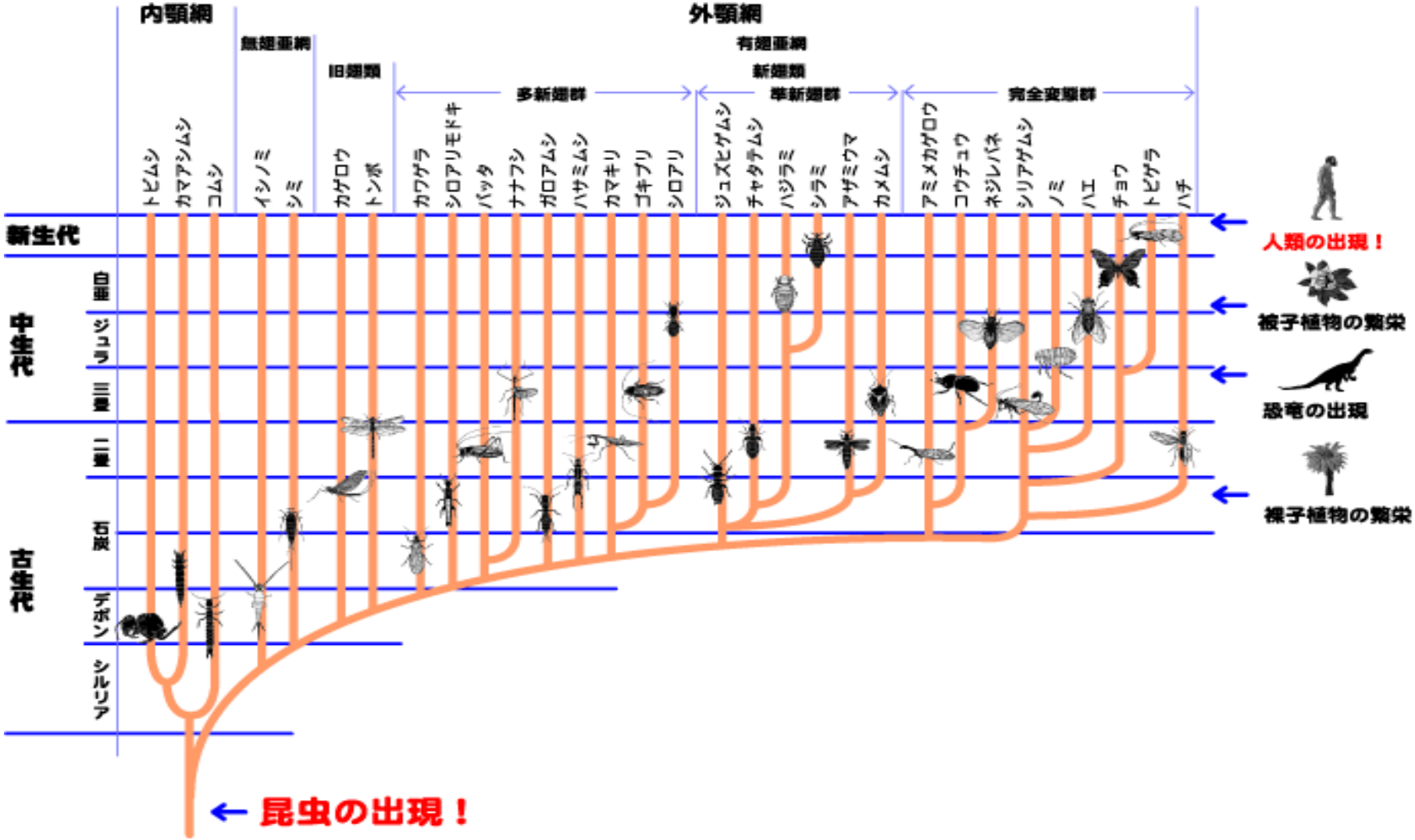
分類した結果



関連したデータ同士に分類する

系統樹作成法の選択

系統樹とは…生物の進化の道筋等を枝分かれした図として示したもの



引用元http://www.museum.kyushu-u.ac.jp/INSECT/02/02.html

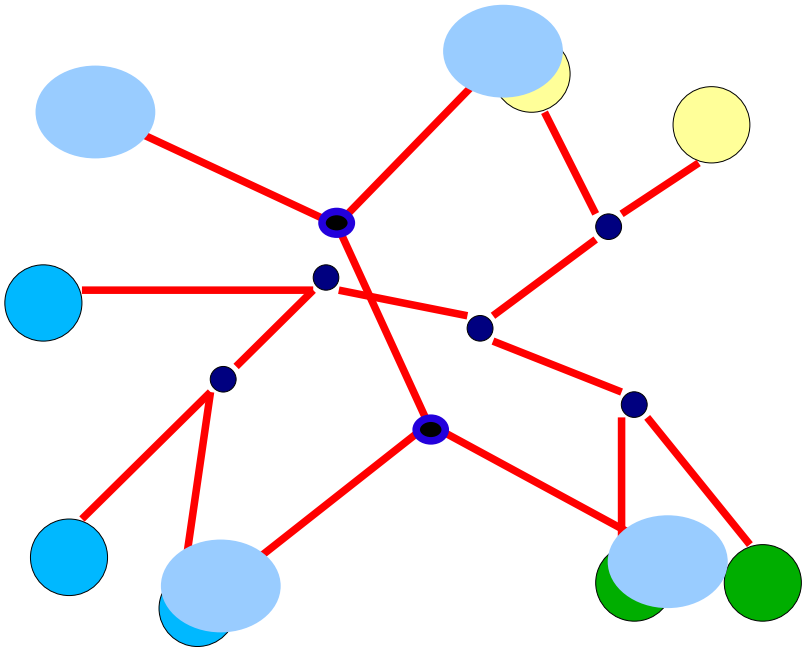
系統樹作成法の選択

nj法(近隣結合法)

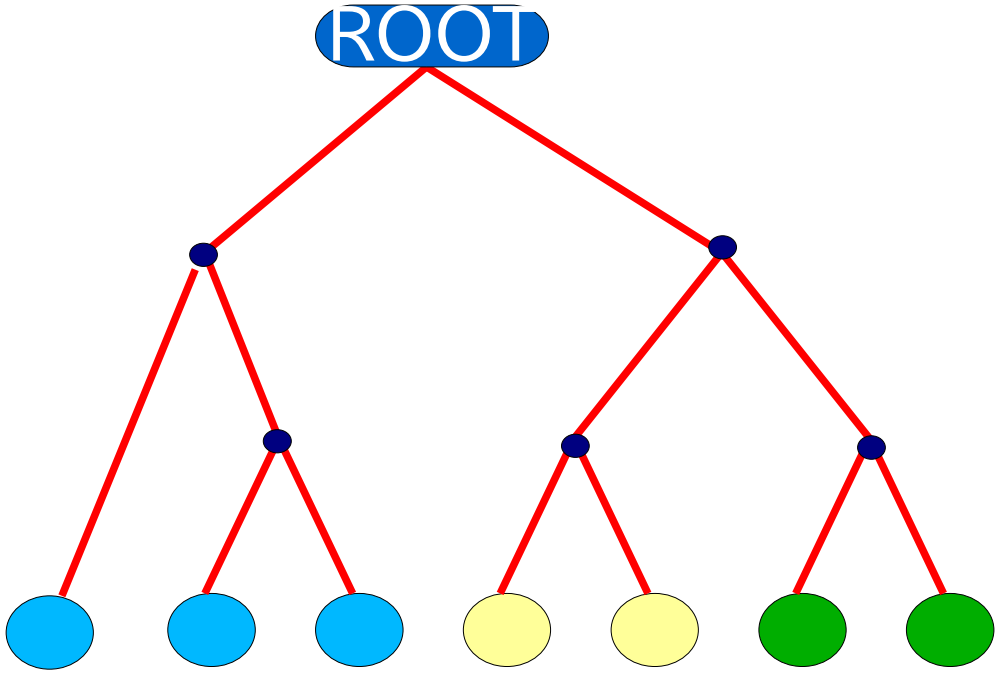
quartet method

upgma法(平均距離法)

無根系統樹

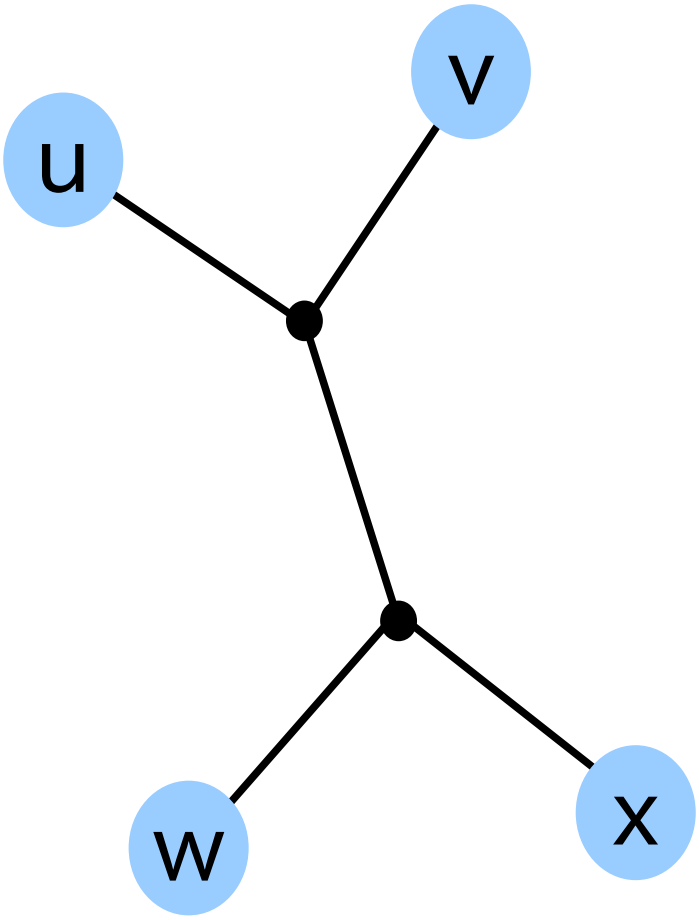


有根系統樹



系統樹作成法の選択

quartet method



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

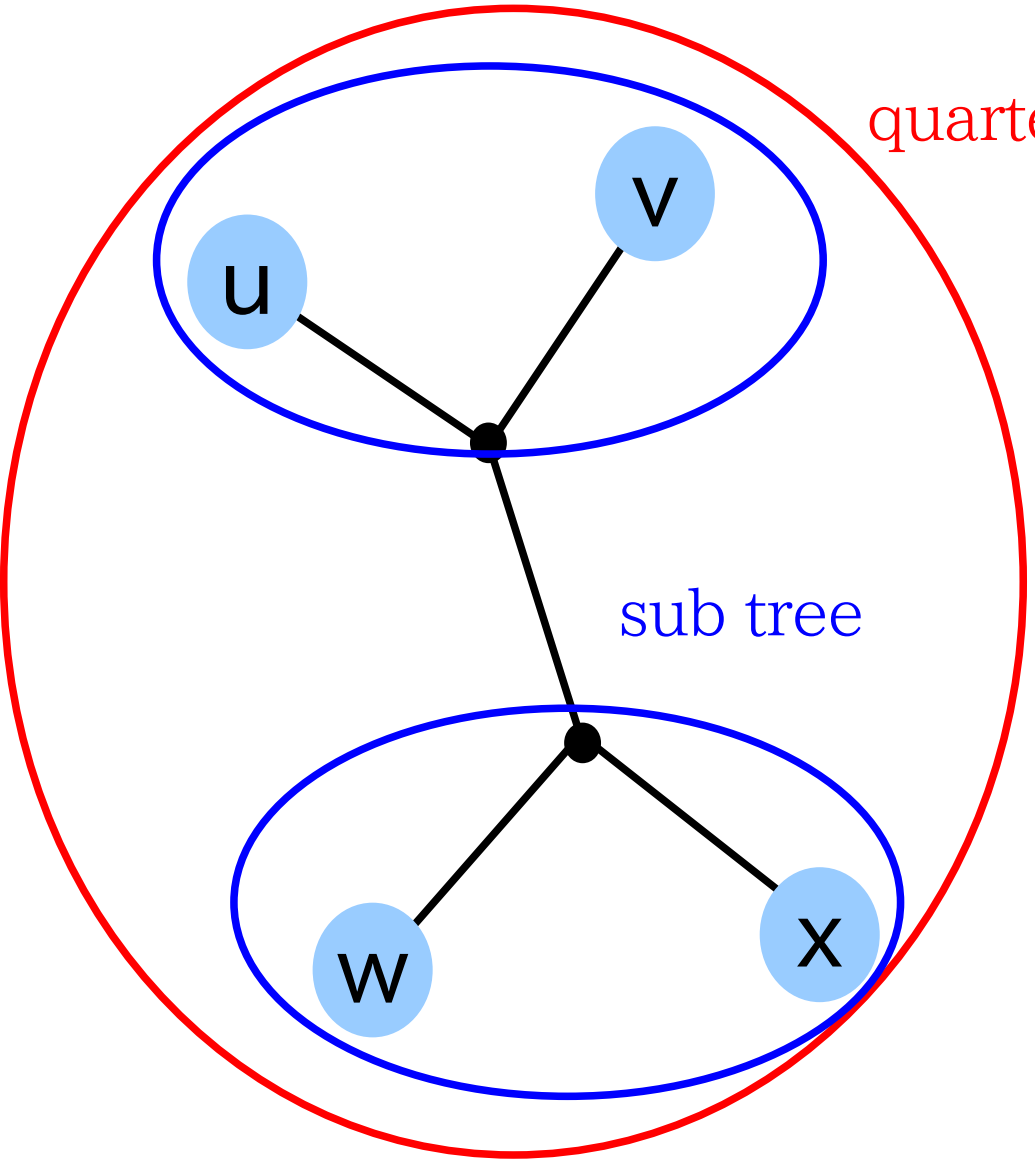
- leaf swap
- subtree swap
- subtree transfer

K回繰り返す 又は S(T) ≥ x

S(T)の値を計算

系統樹作成法の選択

quartet…2つの葉を持つ2つのsubtreeが連結したグラフ

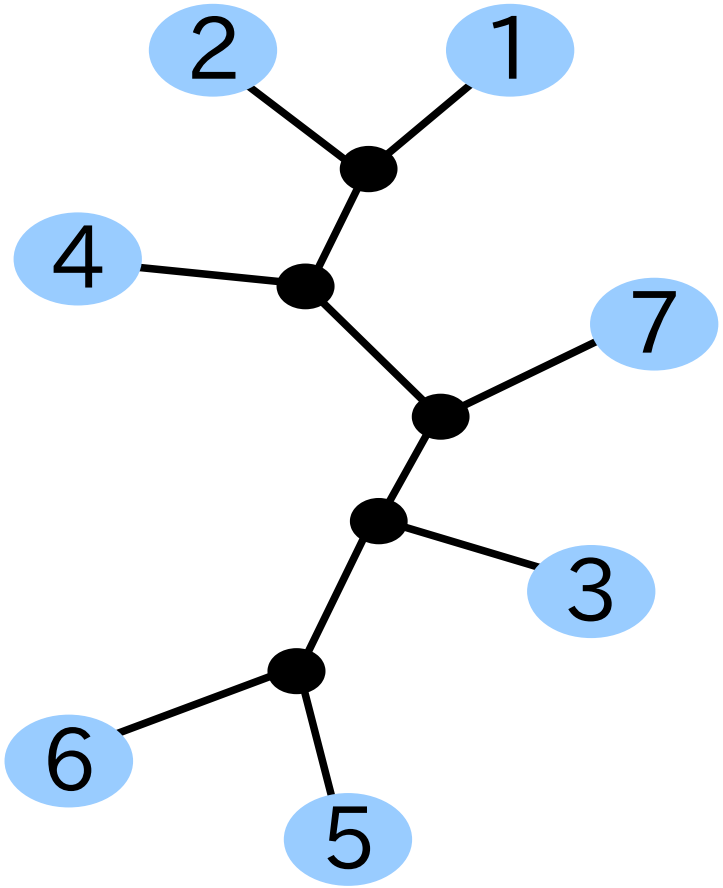


$$uv|wx$$

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

近隣結合法の選択

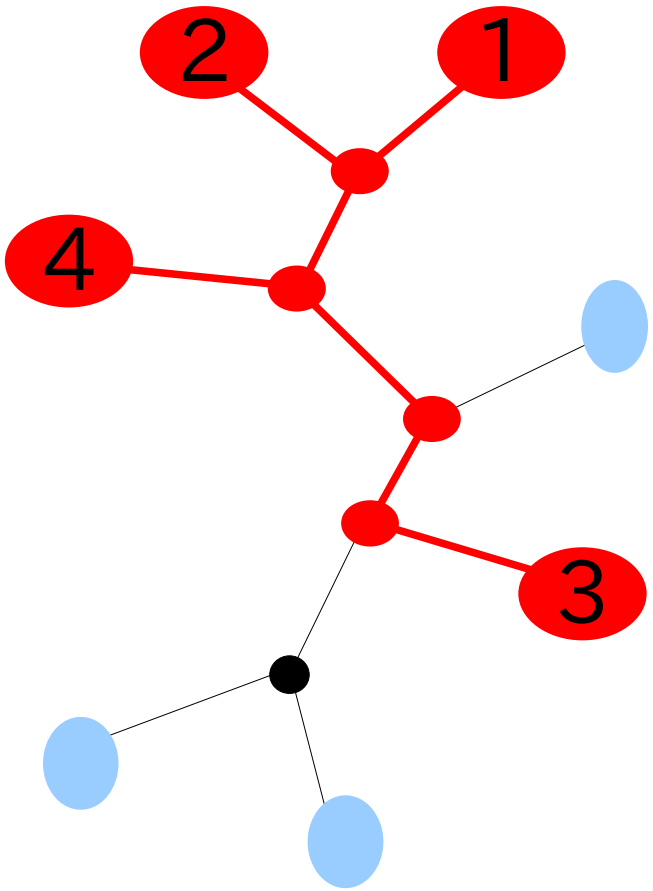
T:木



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

隣接結合法の選択

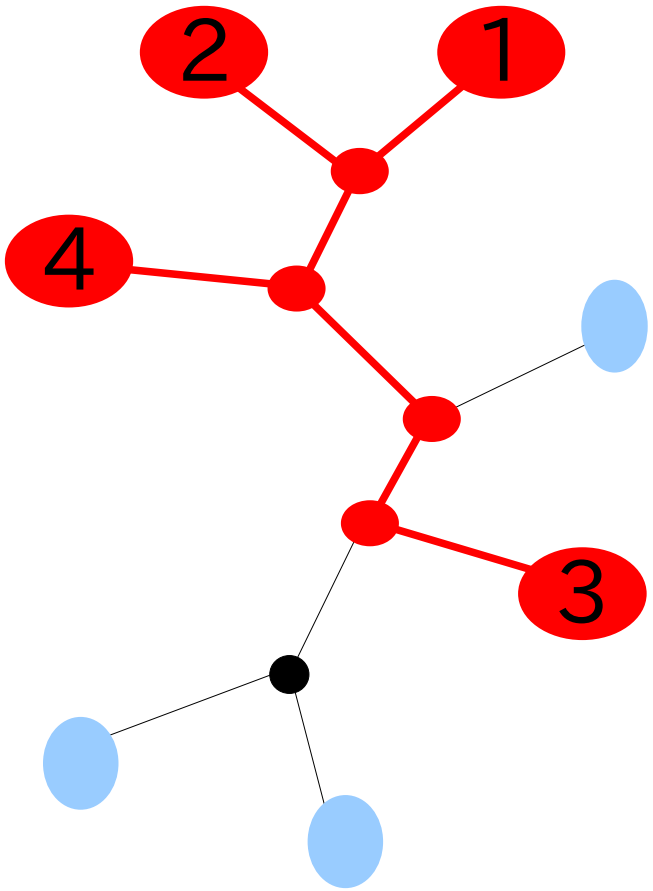
T:木



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

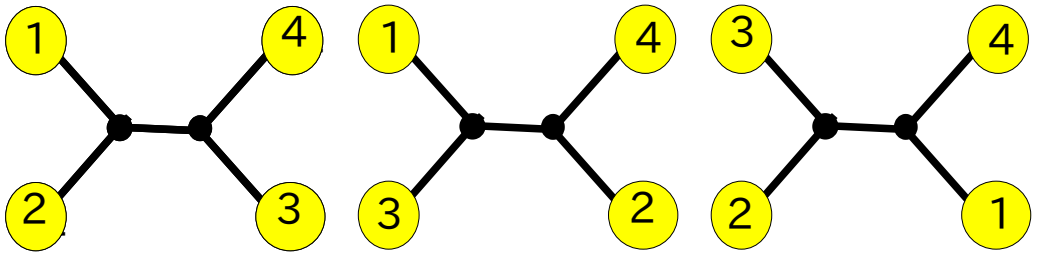
系統樹作成法の選択

T:木



{1,2,3,4}

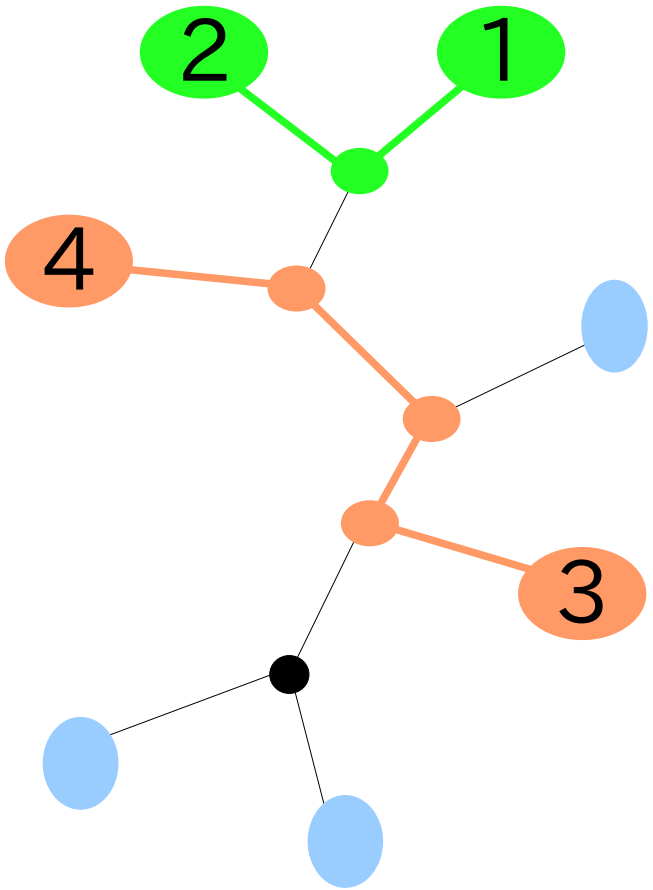
12|34　　13|24　　23|14



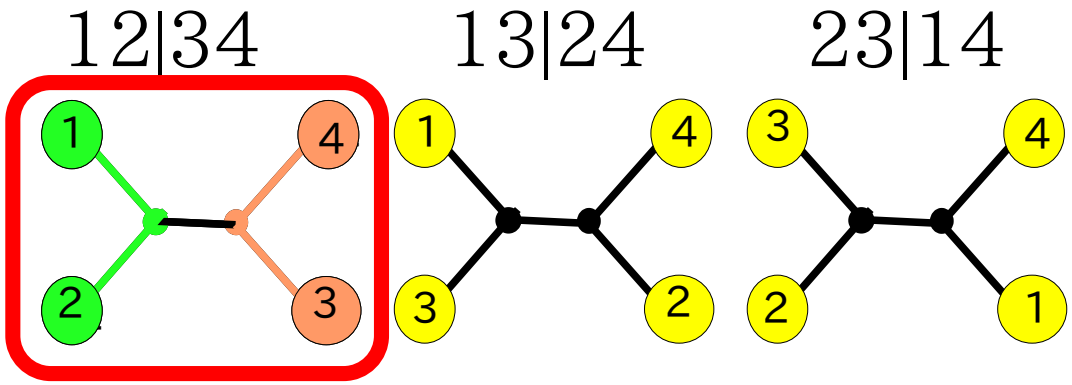
$$C_{uv|wx} = d(u,v) + d(w,x)$$

近隣結合法の選択

T:木



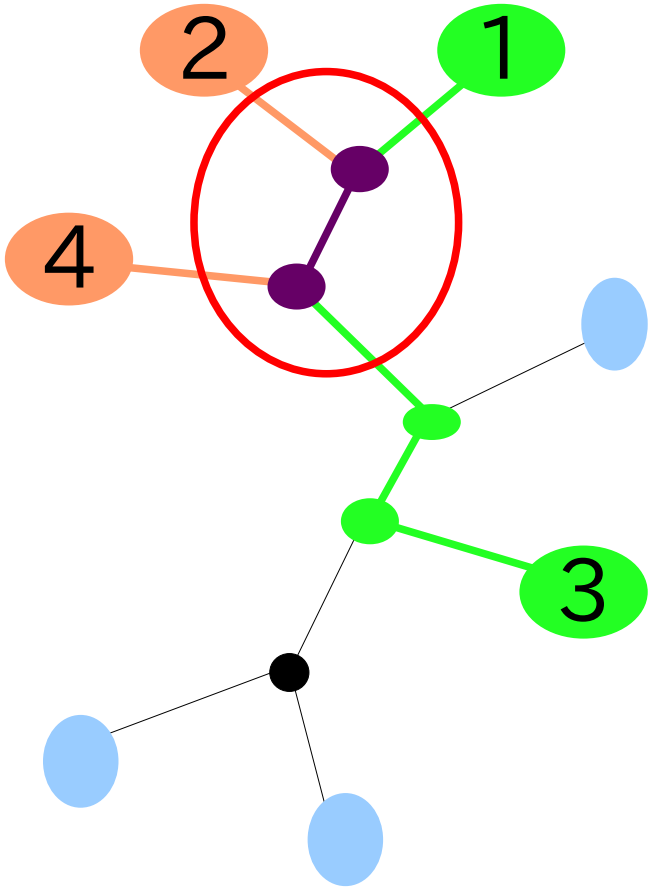
{1,2,3,4}



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

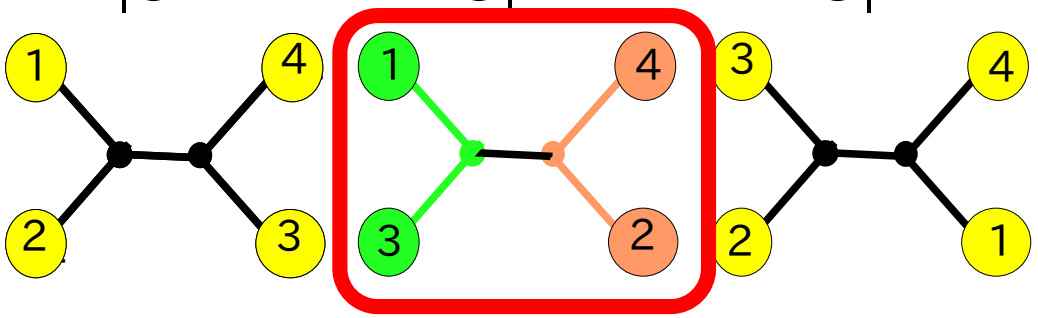
系統樹作成法の選択

T:木



{1,2,3,4}

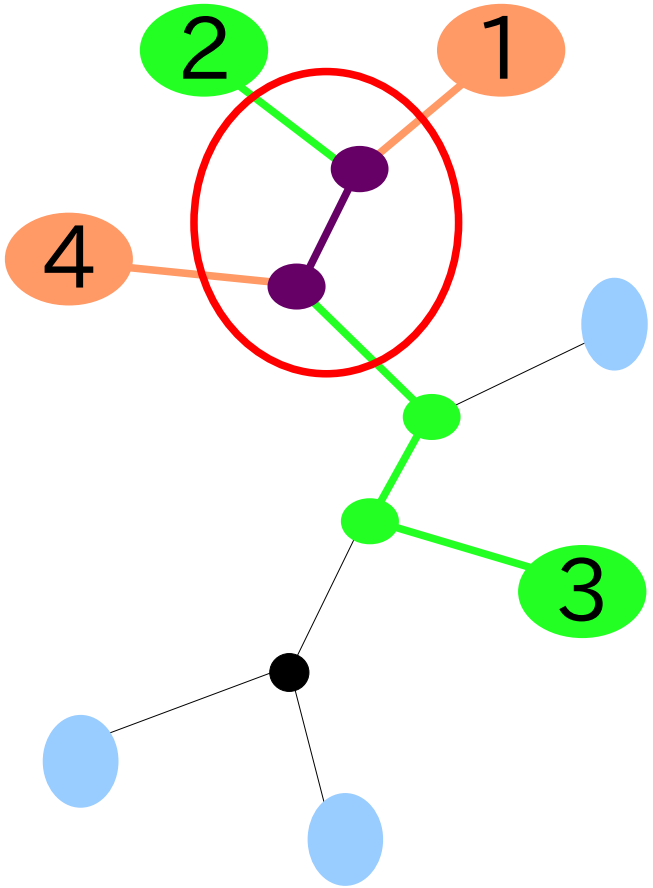
12|34　13|24　23|14



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

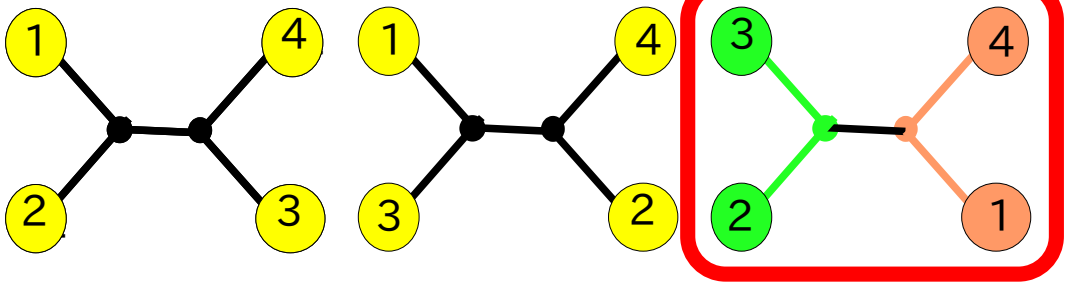
近隣結合法の選択

T:木



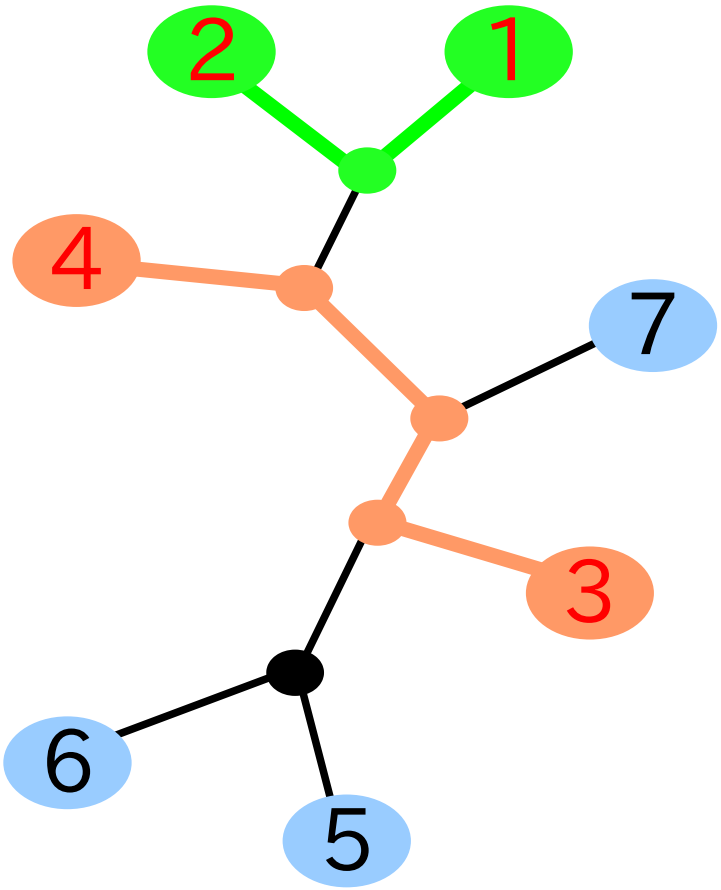
{1,2,3,4}

12|34　13|24　23|14



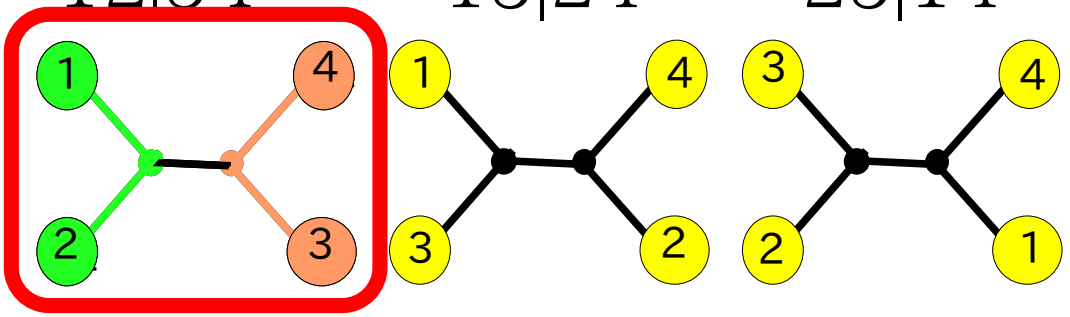
$$C_{uv|wx} = d(u,v) + d(w,x)$$

系統樹作成法の選択



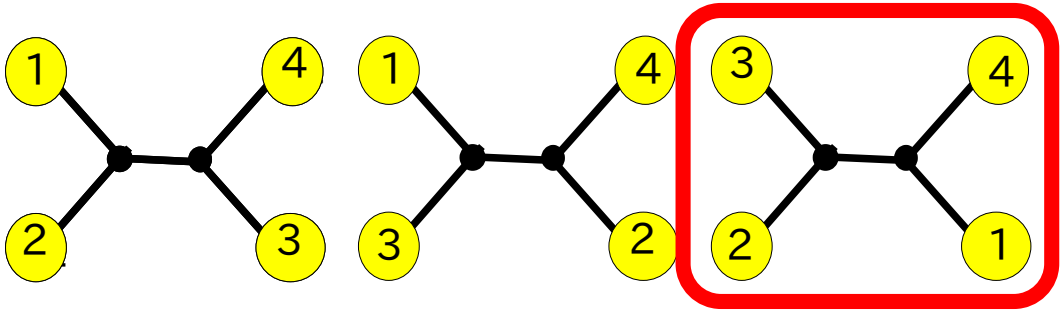
{1,2,3,4}

12|34　　13|24　　23|14



consistent

{4,5,6,7}

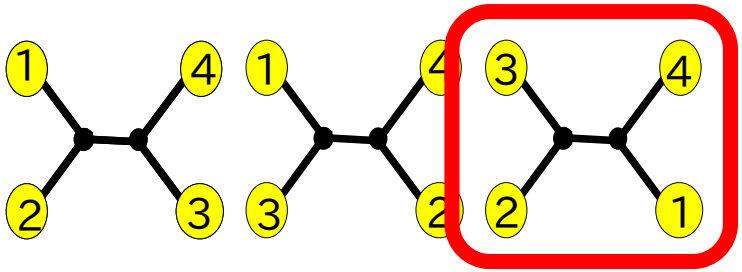


$$C_{uv|wx} = d(u,v) + d(w,x)$$

total cost: C_T

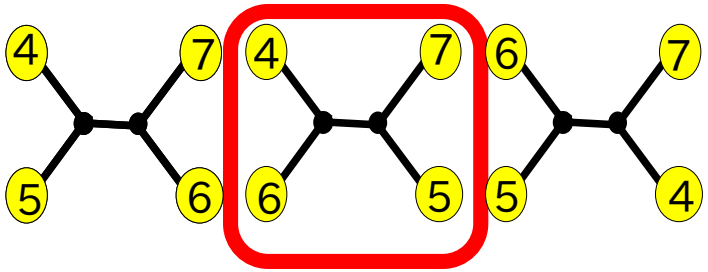
系統樹作成法の選択

{1,2,3,4}

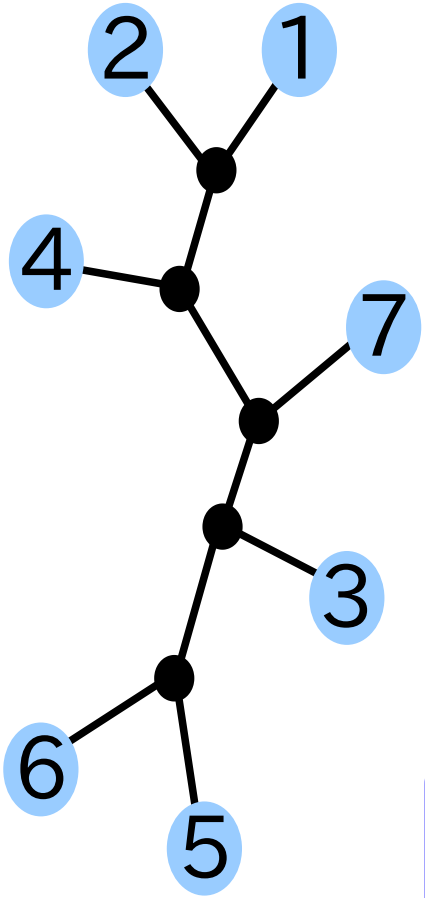


minimum cost

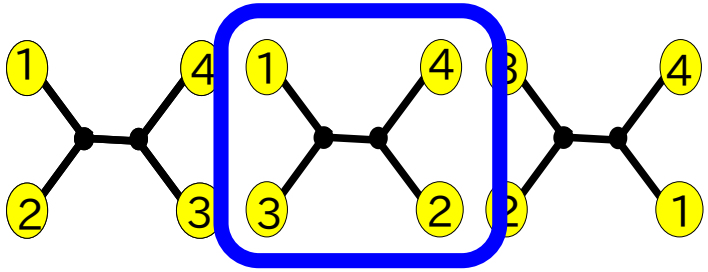
{4,5,6,7}



minimum cost: m

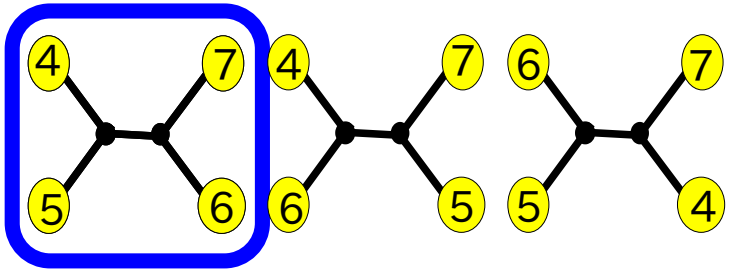


{1,2,3,4}



maximum cost

{4,5,6,7}



maximum cost: M

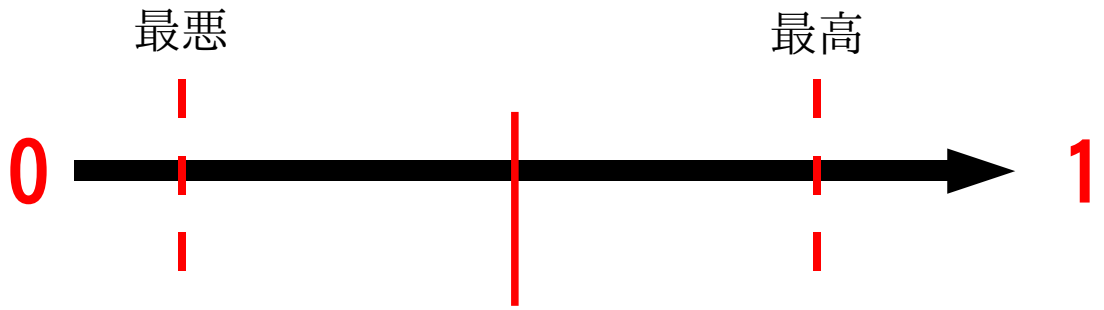
疑似木生成機構の選択

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

M :maximum cost

m :minimum cost

C_T :*total cost*



木:TのS(T)

疑似木作成法の選択

ランダムに木を生成

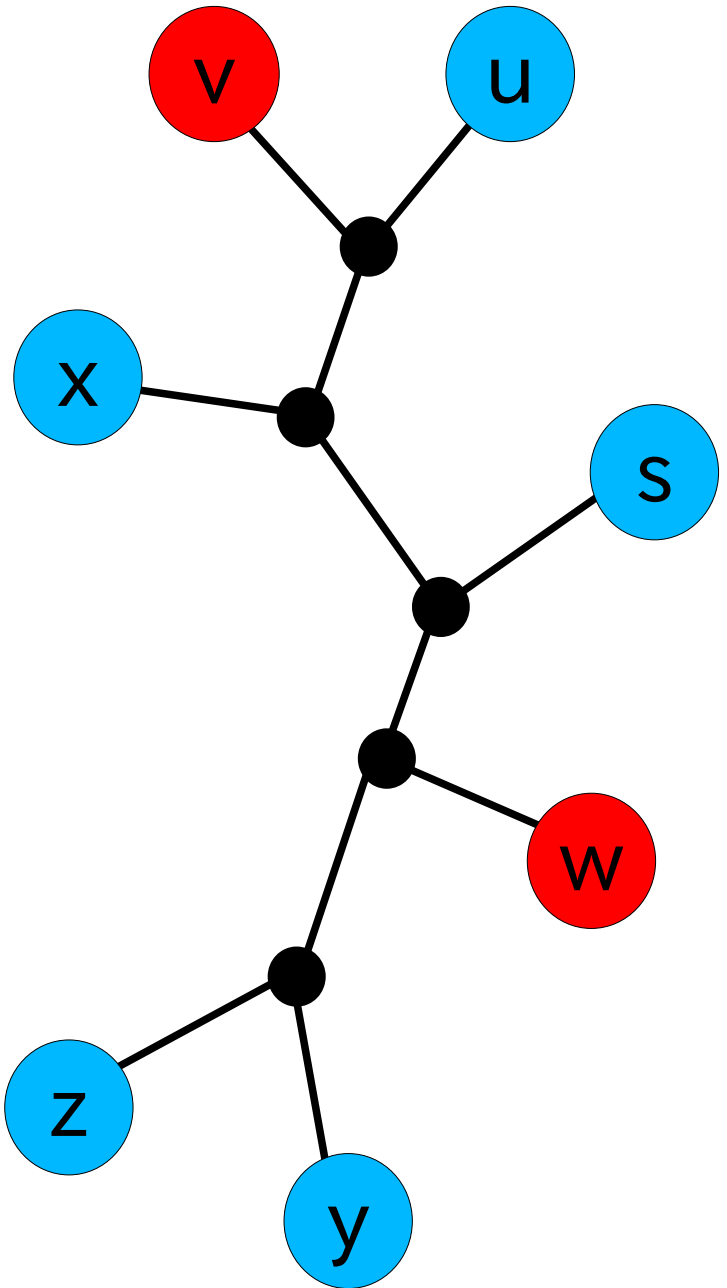
S(T)の値を計算

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 又は S(T) ≥ x

S(T)の値を計算



系統樹作成法の選択

ランダムに木を生成

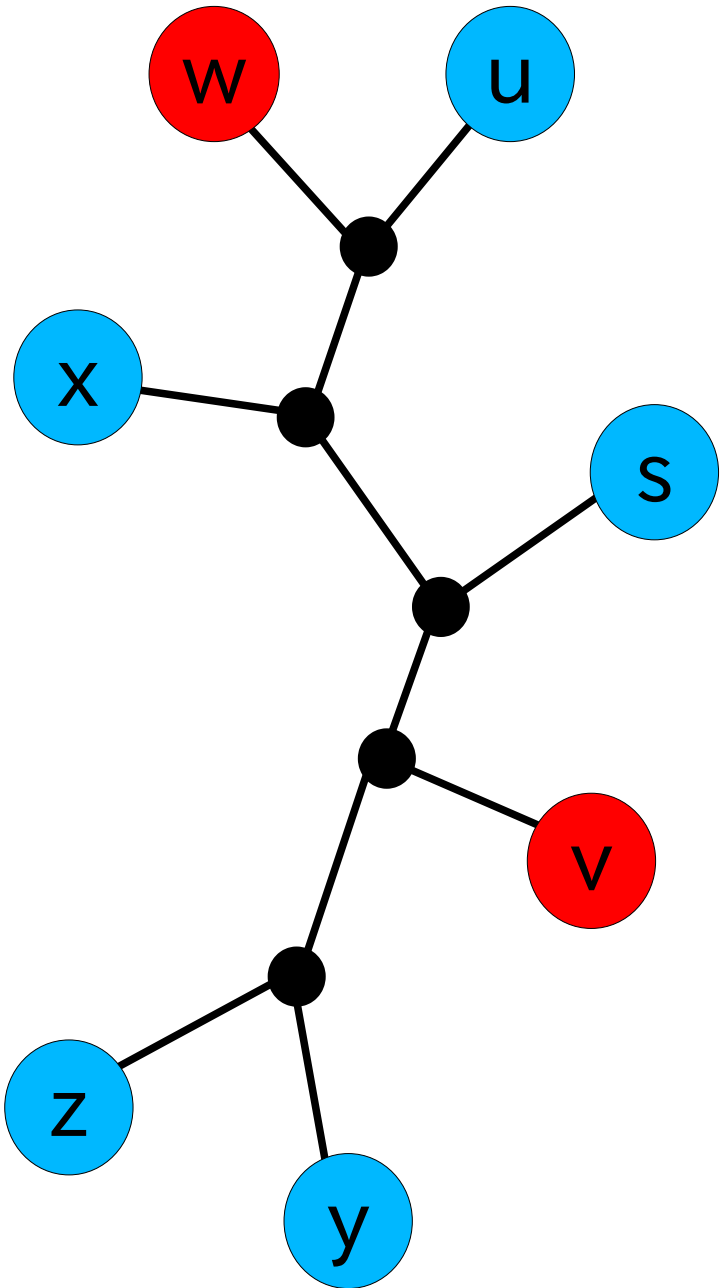
S(T)の値を計算

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 又は S(T) ≥ x

S(T)の値を計算



疑似木作成法の選択

ランダムに木を生成

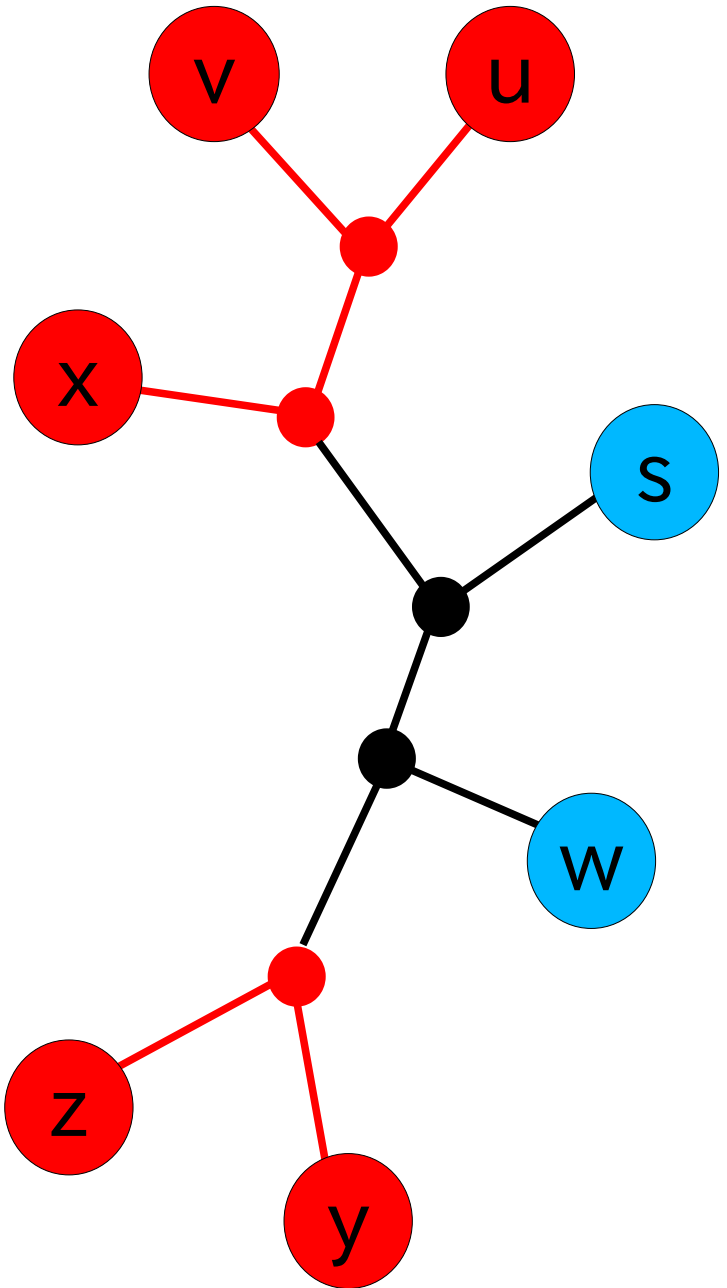
S(T)の値を計算

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 又は S(T) ≥ x

S(T)の値を計算



疑似木作成法の選択

ランダムに木を生成

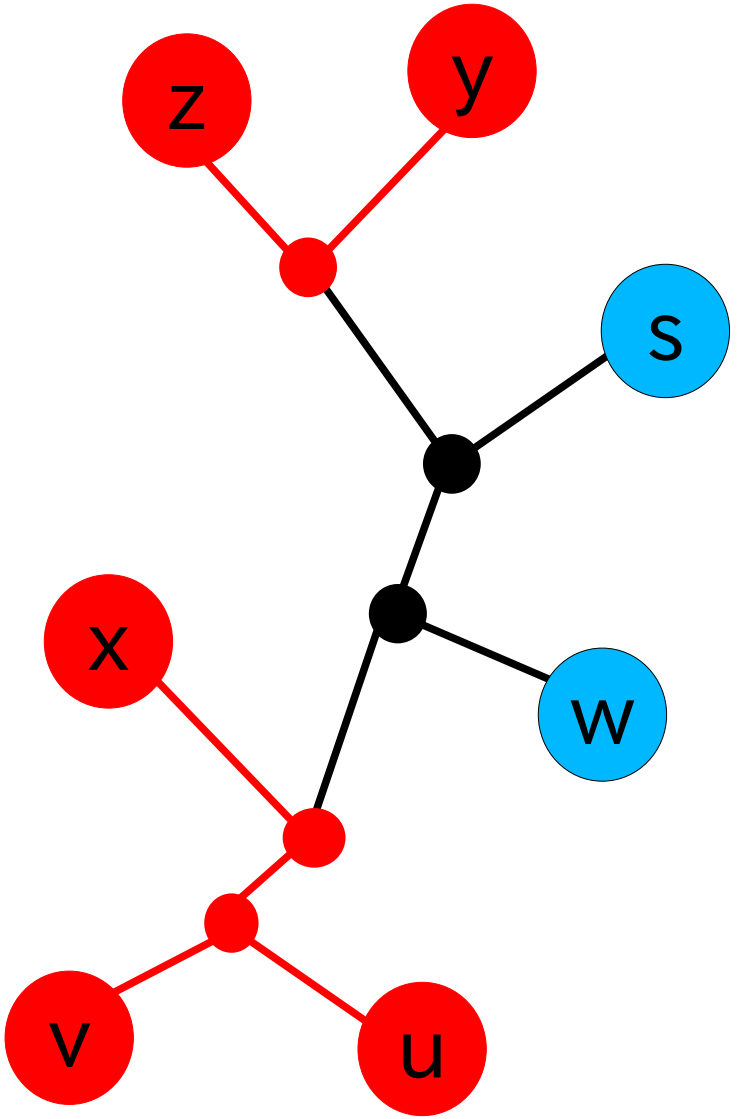
S(T)の値を計算

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 又は S(T) ≥ x

S(T)の値を計算



疑似木作成法の選択

ランダムに木を生成

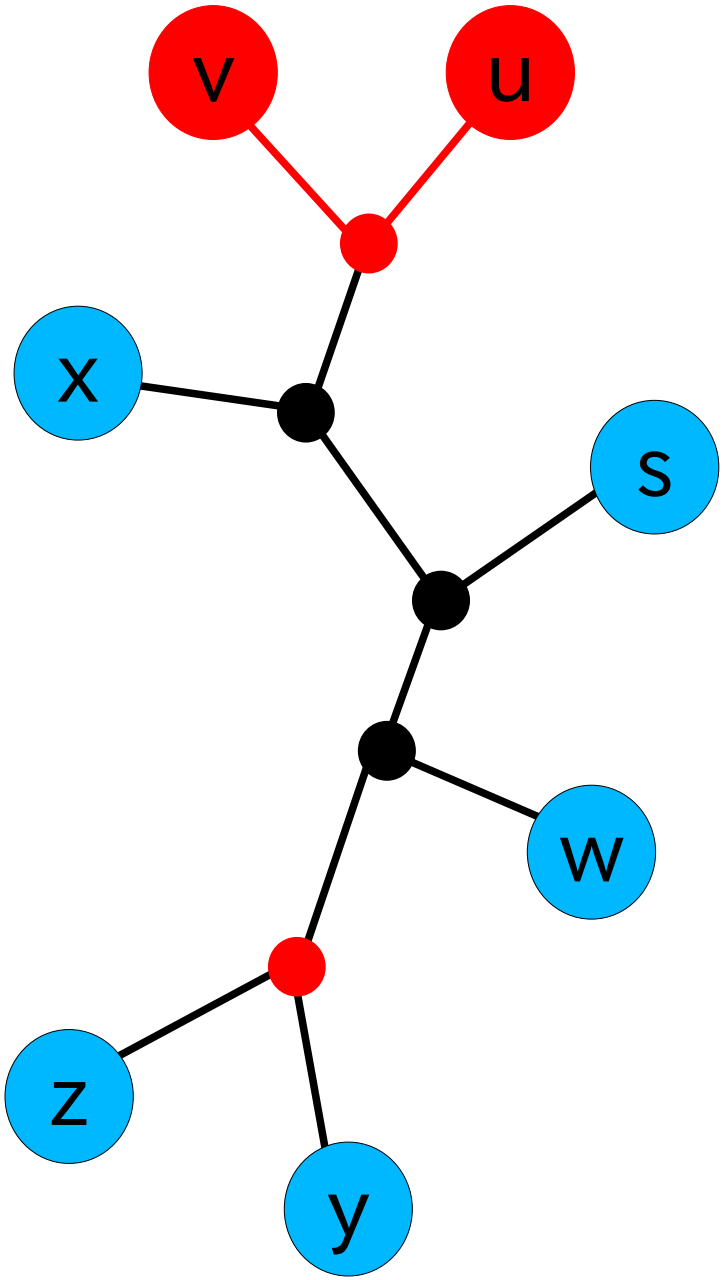
S(T)の値を計算

以下の操作からランダムに選ぶ

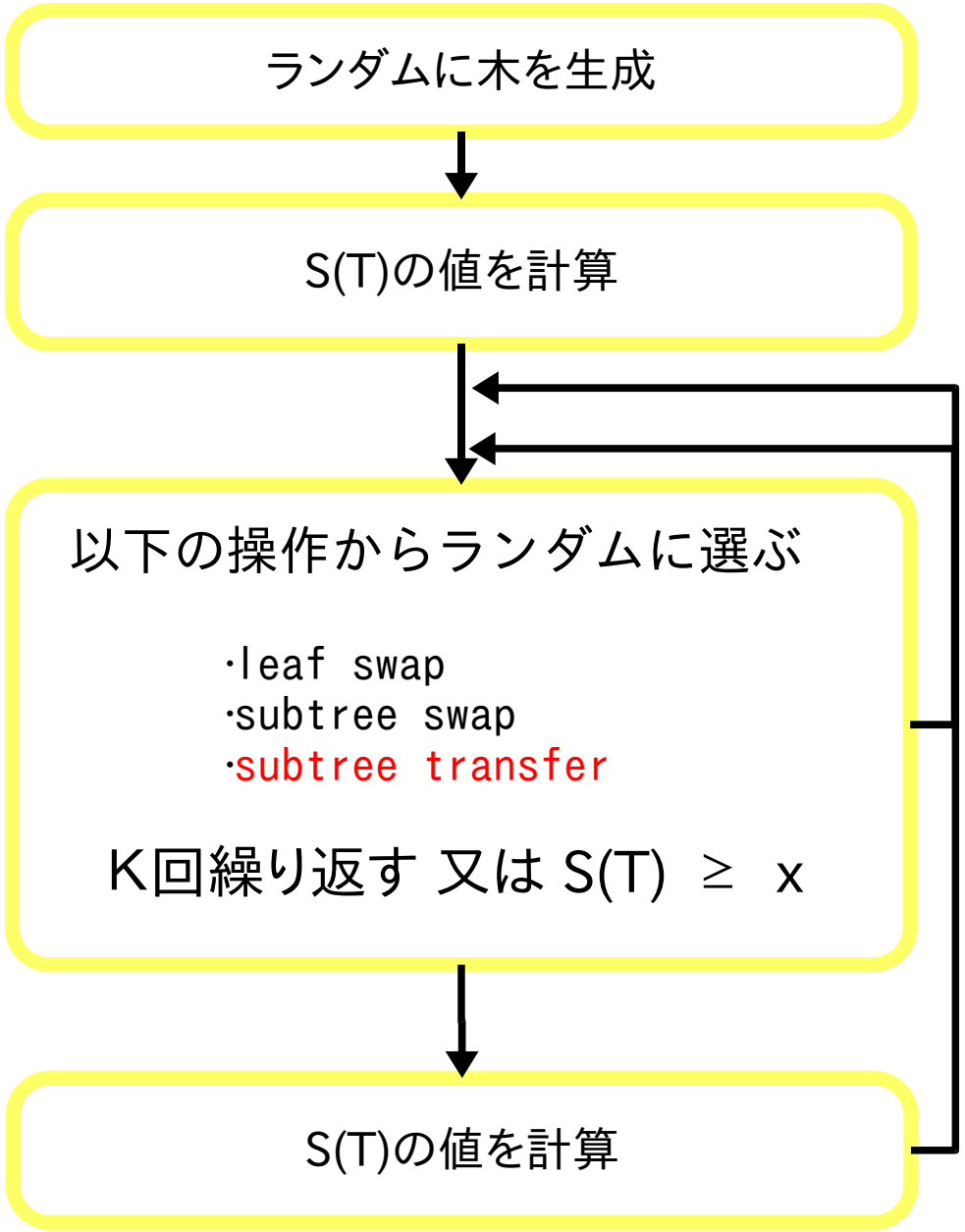
- leaf swap
- subtree swap
- subtree transfer

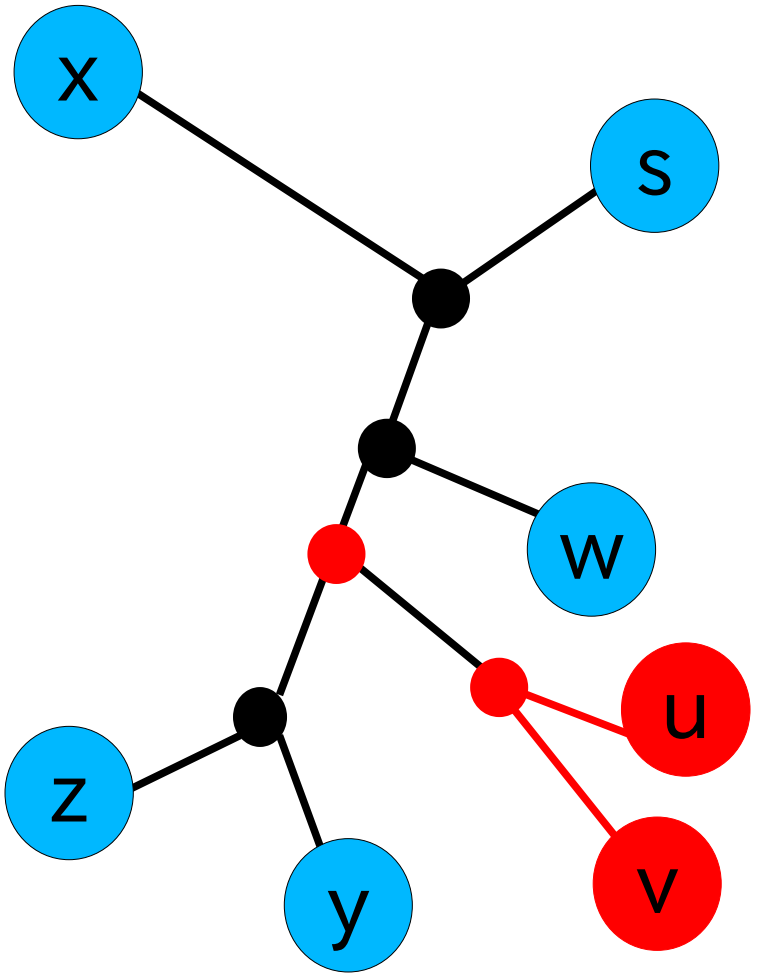
K回繰り返す 又は S(T) ≥ x

S(T)の値を計算

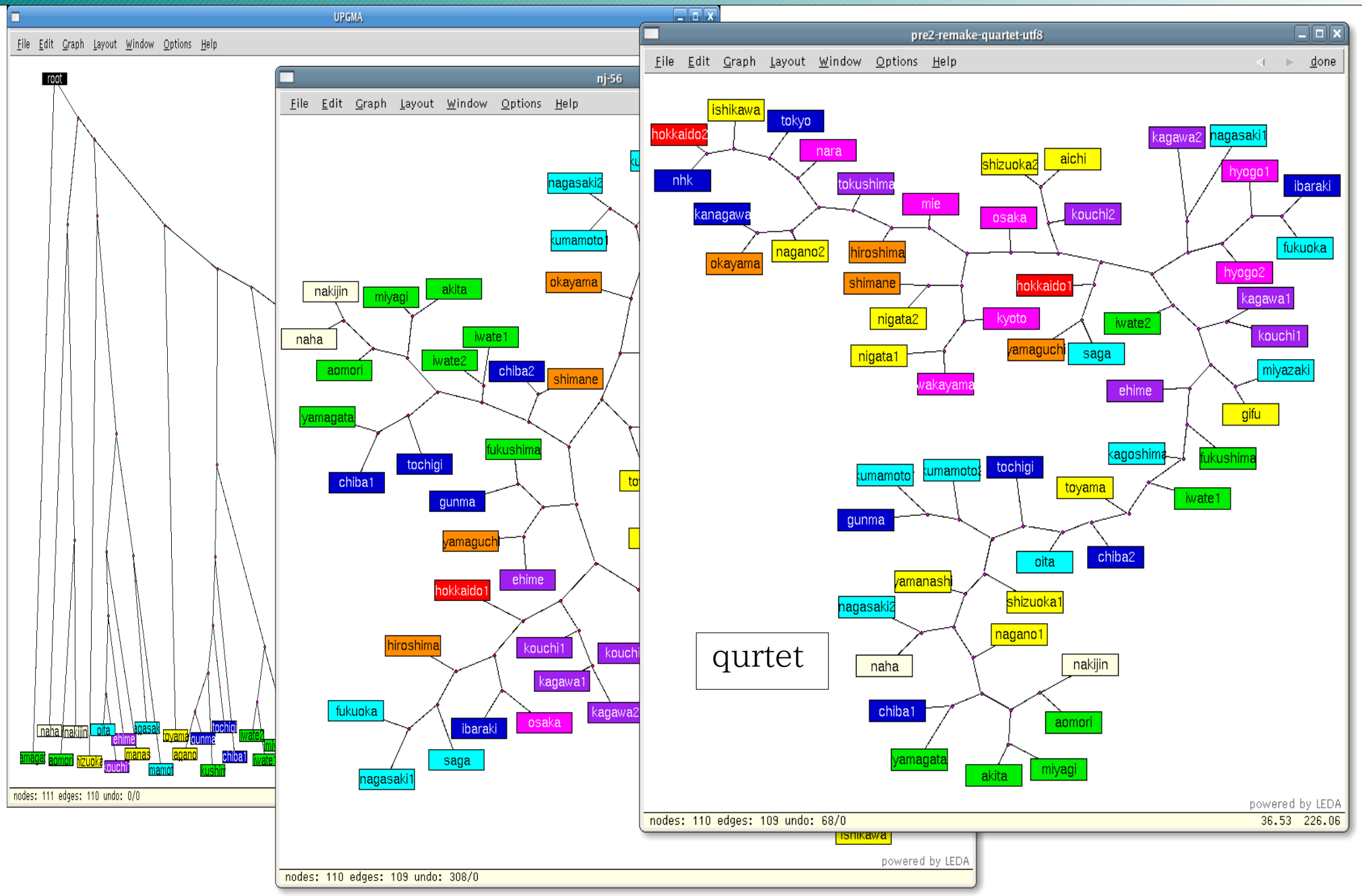


系統樹作成法の選択


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 又は S(T) ≥ x
S(T)の値を計算


x
s
w
u
z
y
v

系統樹作成法の選択

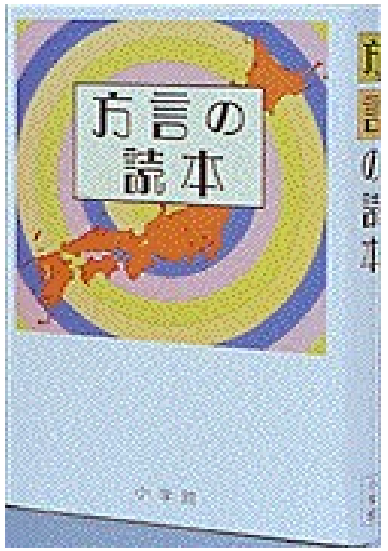

UPGMA
File Edit Graph Layout Window Options Help
root
nodes: 111 edges: 110 undo: 0/0
nj-56
File Edit Graph Layout Window Options Help
nagasaki2
kumamoto1
okayama
nakijin
miyagi
akita
naha
iwate1
aomori
iwate2
chiba2
shimane
yamagata
fukushima
tochigi
chiba1
gunma
yamaguchi
ehime
hokkaido1
hiroshima
kouchi1
kagawa1
fukuoka
ibaraki
osaka
saga
nagasaki1
powered by LEDA
nodes: 110 edges: 109 undo: 308/0
pre2-remake-quartet-utf8
File Edit Graph Layout Window Options Help done
ishikawa
tokyo
hokkaido2
nara
nhk
tokushima
kanagawa
mie
okayama
nagano2
hiroshima
shimane
nigata2
kyoto
nigata1
wakayama
shizuoka2
aichi
osaka
kouchi2
kagawa2
nagasaki1
hyogo1
ibaraki
fukuoka
hyogo2
hokkaido1
kagawa1
iwate2
kouchi1
yamaguchi
saga
miyazaki
ehime
gifu
kagoshima
fukushima
kumamoto1
kumamoto2
tochigi
toyama
iwate1
gunma
oita
chiba2
yamanashi
shizuoka1
nagasaki2
nagano1
naha
nakijin
chiba1
aomori
yamagata
akita
miyagi
qurtet
powered by LEDA
nodes: 110 edges: 109 undo: 68/0
36.53 226.06

目次

- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の揺らぎ
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

『方言の読本』(小学館/著・佐藤亮一)

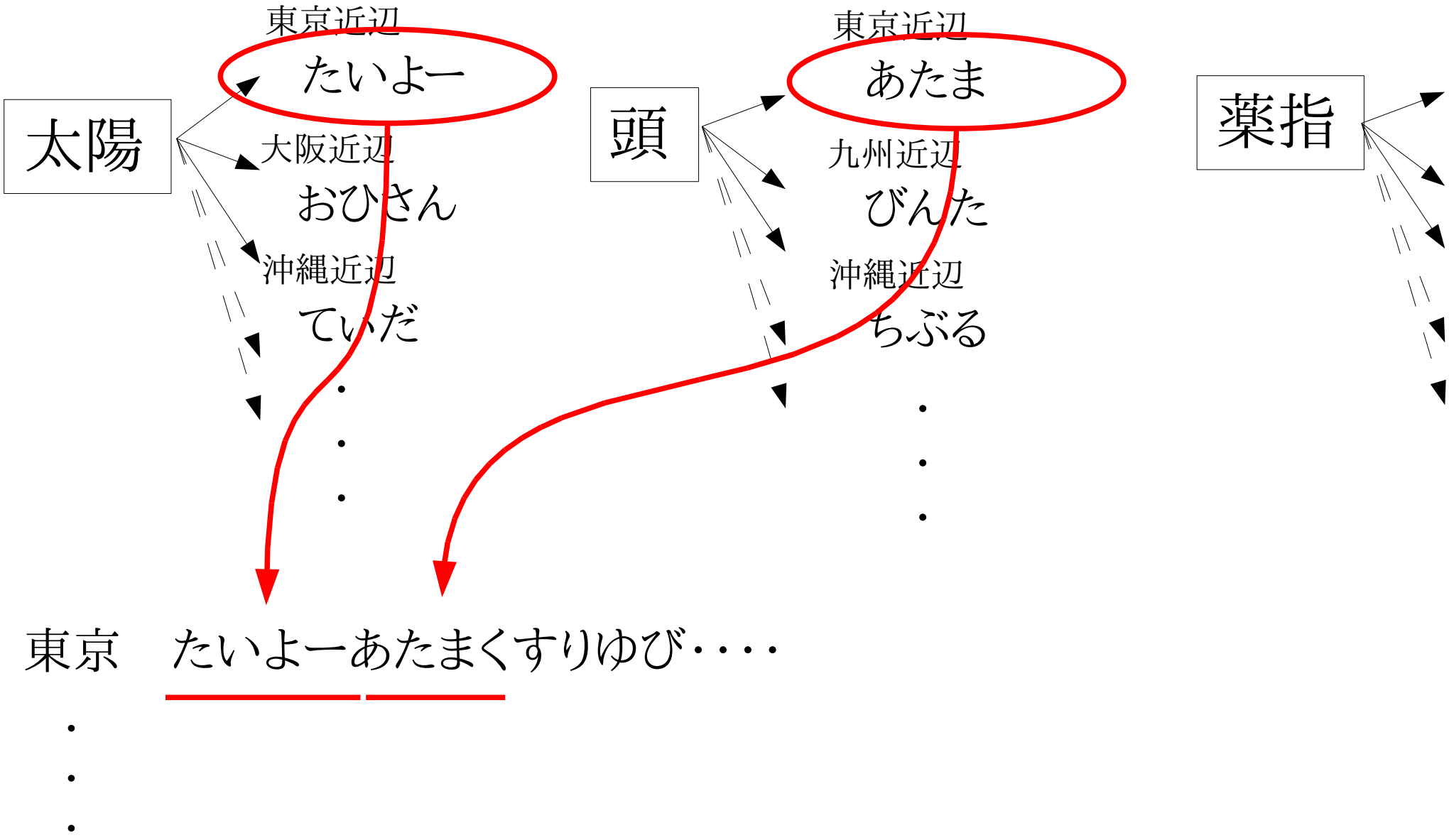
『日本方言大辞典』(小学館)所収の
方言地図から単語130点を選定・分類
し解説をつけまとめたもの



http://www.s-book.com/plsql/com2_detail?isbn=409504151X

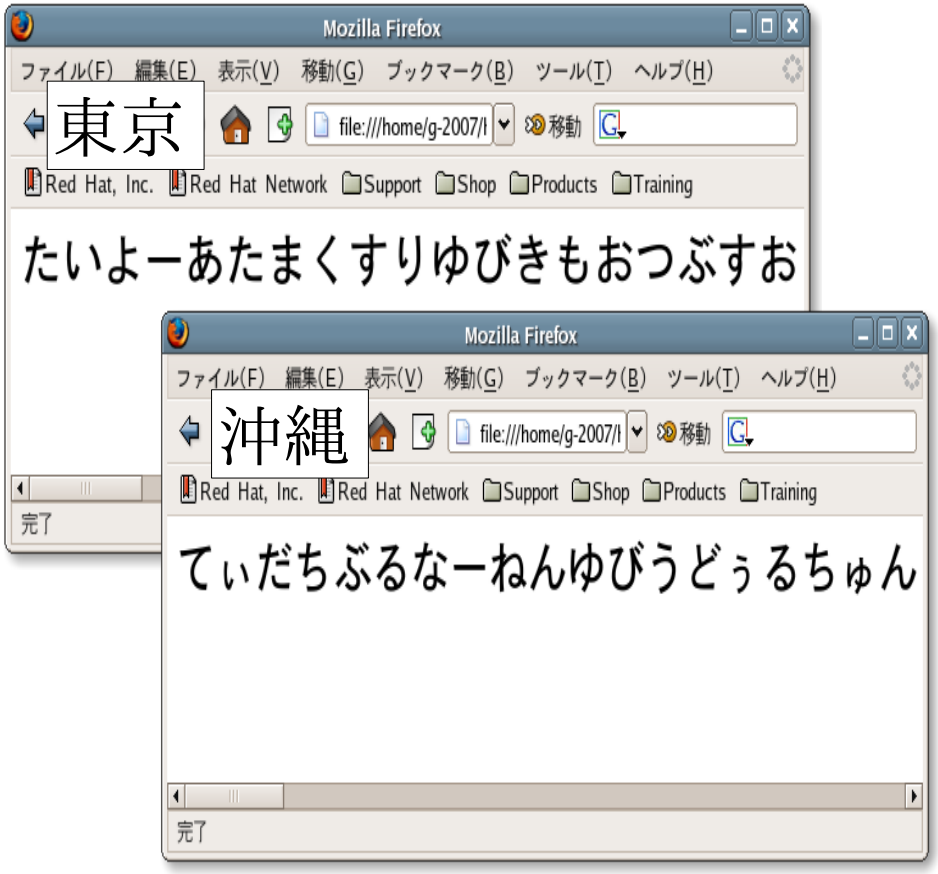
単語毎の分類がされているものを文章にするとどのような分布になるのか

データ作り


太陽
東京近辺
たいよー
大阪近辺
おひさん
沖縄近辺
ていだ
頭
東京近辺
あたま
九州近辺
びんた
沖縄近辺
ちぶる
薬指
東京　たいよーあたまくすりゆび・・・・

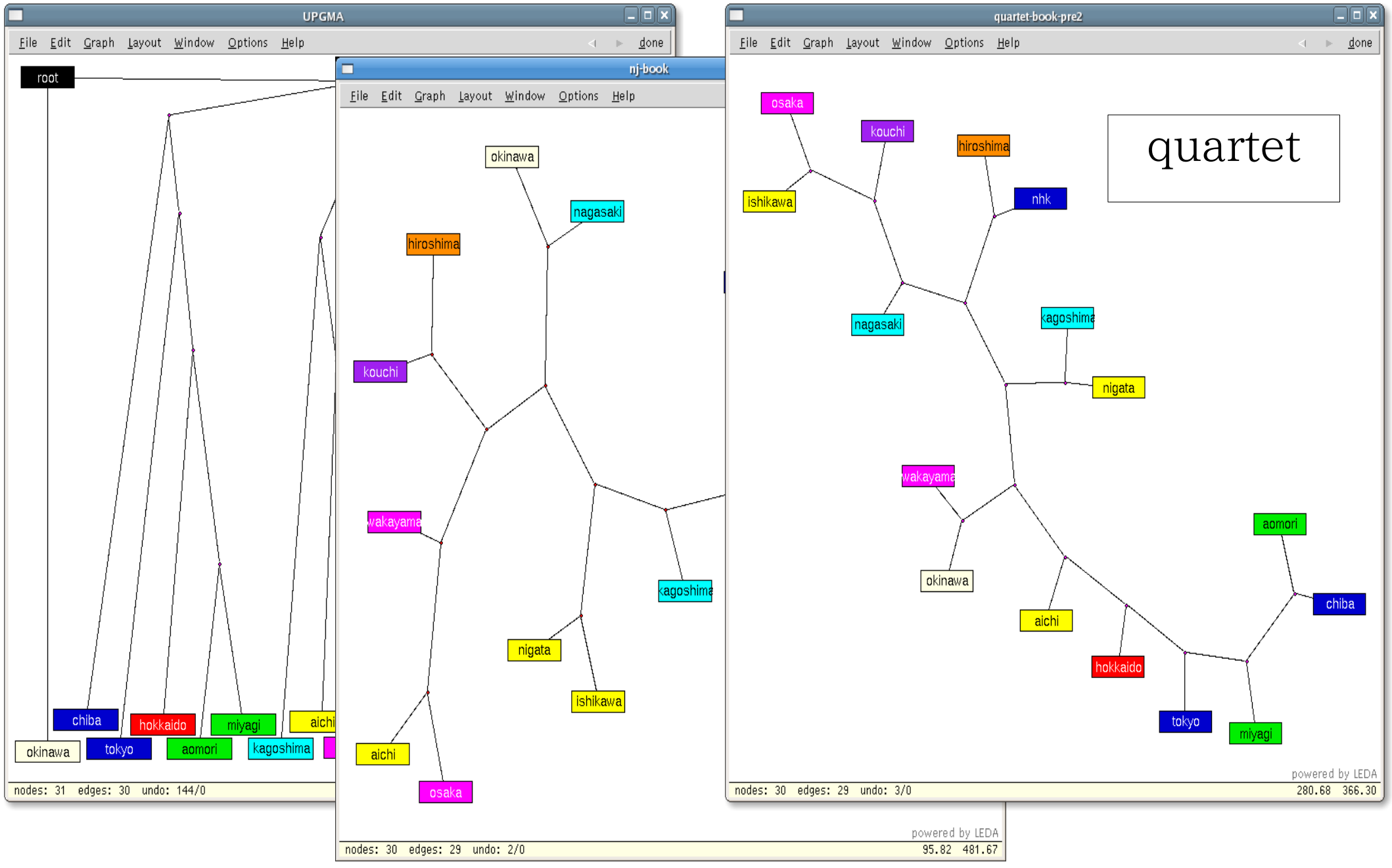
データ作り

方言の読本



方言の特徴が出やすいであろう
全国16箇所を選定し、単語毎の
方言をつなげ合わせ一文にする

視覚的な比較


UPGMA
File Edit Graph Layout Window Options Help done
root
chiba
hokkaido
miyagi
aichi
okinawa
tokyo
aomori
kagoshima
nodes: 31 edges: 30 undo: 144/0
nj-book
File Edit Graph Layout Window Options Help
okinawa
nagasaki
hiroshima
kouchi
wakayama
kagoshima
nigata
ishikawa
aichi
osaka
powered by LEDA
nodes: 30 edges: 29 undo: 2/0
95.82 481.67
quartet-book-pre2
File Edit Graph Layout Window Options Help done
quartet
osaka
kouchi
hiroshima
ishikawa
nhk
nagasaki
kagoshima
nigata
wakayama
aomori
okinawa
chiba
aichi
hokkaido
tokyo
miyagi
powered by LEDA
nodes: 30 edges: 29 undo: 3/0
280.68 366.30

目次

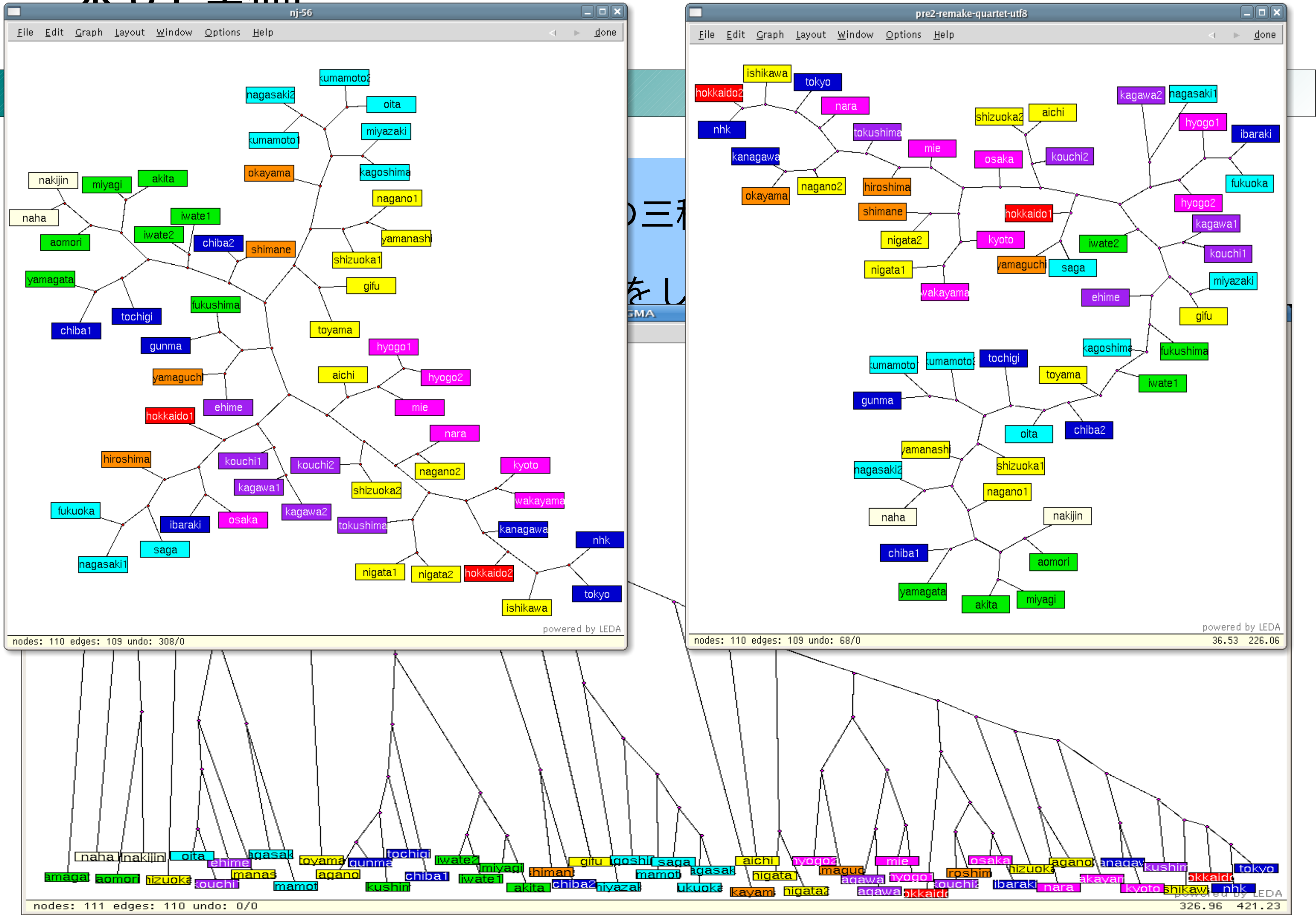
- 背景
- 先行研究
- 研究動機
- 研究項目
  - ・音声データを文字におこす際の信憑性
  - ・文字コードによる違い
  - ・系統樹作成法の選択
  - ・「方言の読本」
  - ・木の評価
- 今後の課題

木の評価

NJ、UPGMA、Quartet Methodの三種類でグラフを作成

三種類のどれがより良い分類をしているのだろうか??

木の評価

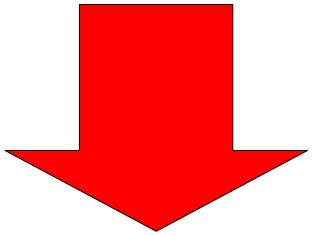


nj-56
File Edit Graph Layout Window Options Help done
powered by LEDA
nodes: 110 edges: 109 undo: 308/0
pre2-remake-quartet-utf8
File Edit Graph Layout Window Options Help done
powered by LEDA
nodes: 110 edges: 109 undo: 68/0 36.53 226.06
nodes: 111 edges: 110 undo: 0/0 326.96 421.23

木の評価

NJ、UPGMA、Quartet Methodの三種類でグラフを作成

三種類のどれがより良い分類をしているか分からない。



Quartet Methodで使用したS(T)に着目

NJ、UPGMAで作成したグラフのS(T)を求める。

木の評価

S(T)

		Remake
Preprocess2	Quartet Method	0.850286
	NJ	0.455565
	UPGMA	0.455565

NJ、UPGMAのグラフはS(T)が低い

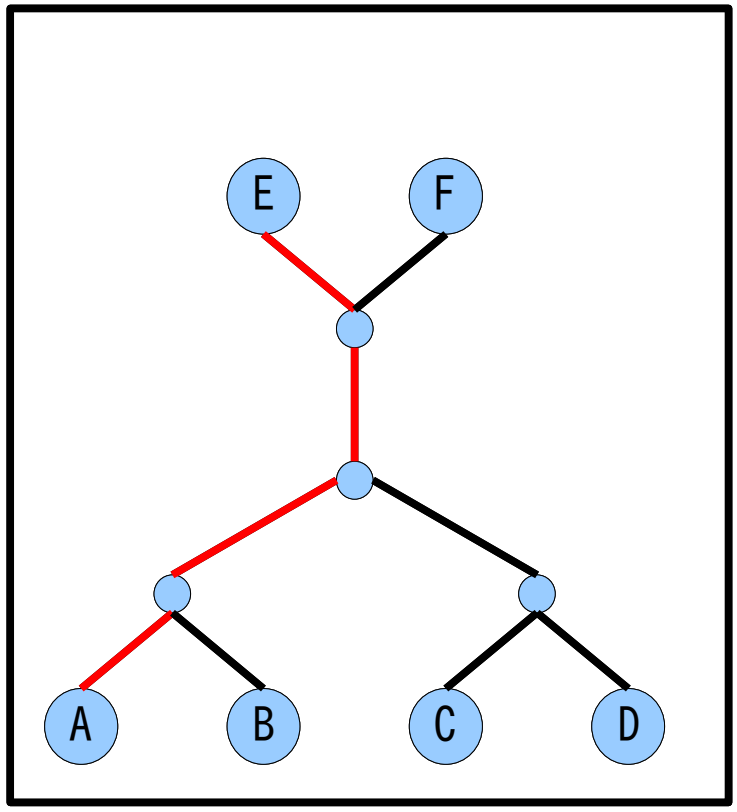
木の評価

グラフを見た場合に類似度の距離が
より反映しているグラフは3つでどれか??

反映しているか評価する値を定義

木の評価

グラフを見た場合の距離



$td(u,v)$：ノード間の辺の数を距離としたもの

$td(A,E)=4$

木の評価

$td(u,v)$ を類似度の距離に対応させた距離を $ntd(u,v)$ とする

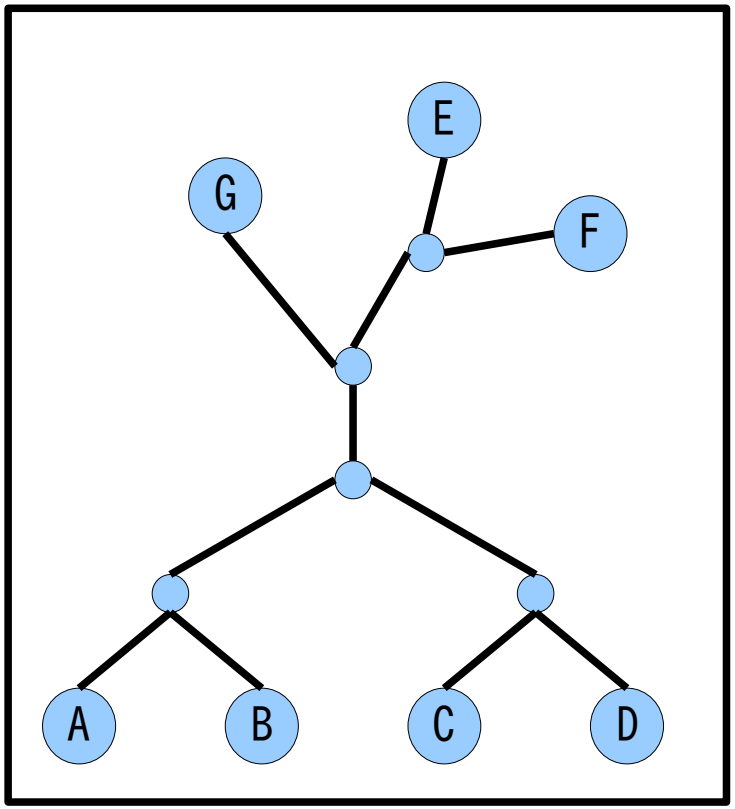
$sd(u,v)$ ： 類似度の距離

$max(sd)$ ：$sd(u,v)$の最大値
$min(sd)$ ：$sd(u,v)$の最小値
$max(td)$ ：$td(u,v)$の最大値
$min(td)$ ：$td(u,v)$の最小値

$$ntd(u,v)=min(sd)+\frac{max(sd)-min(sd)}{max(td)-min(td)}(td(u,v)-min(td))$$

木の評価

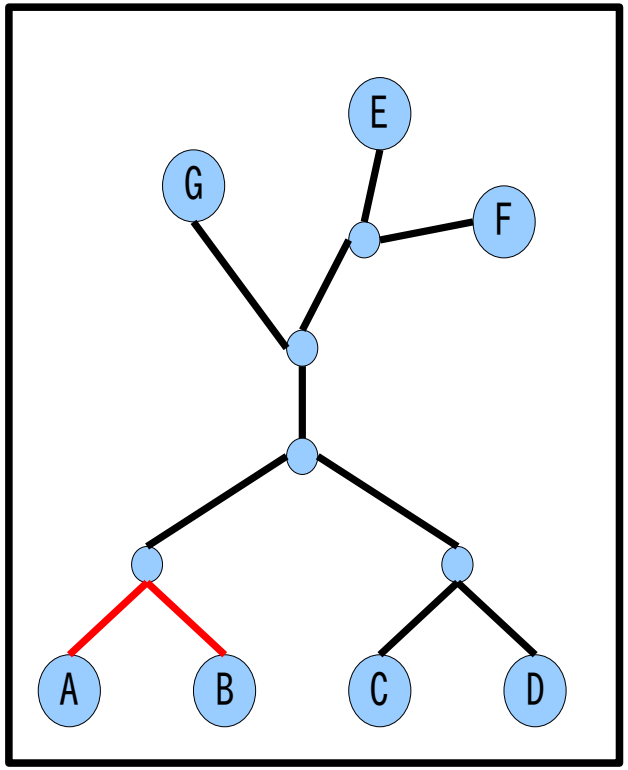
$td(u,v)$を類似度の距離に対応させた距離を$ntd(u,v)$とする

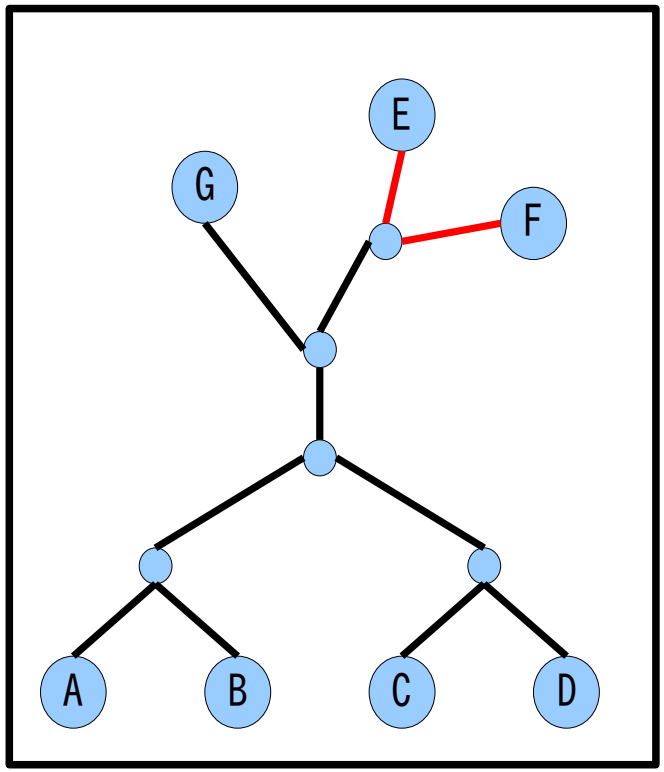


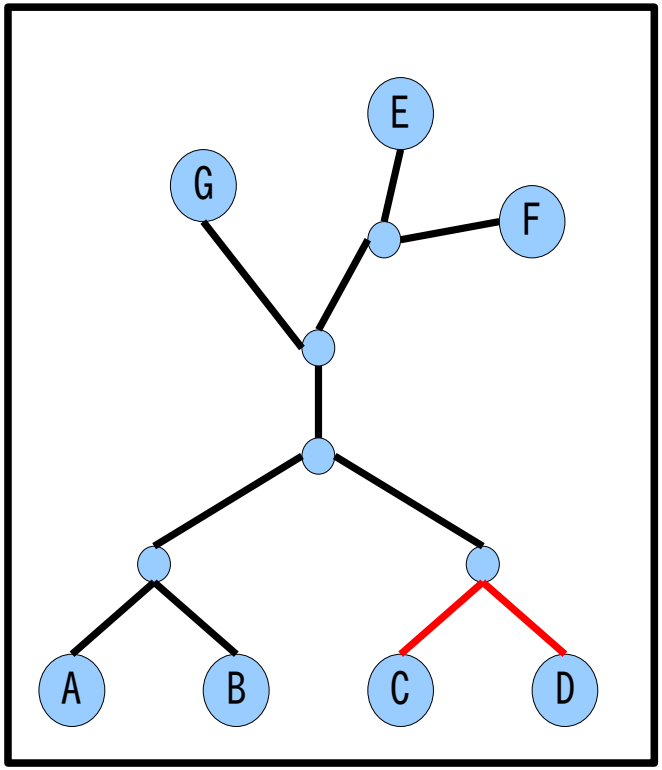
木の評価

$td(u,v)$を類似度の距離に対応させた距離を $ntd(u,v)$とする

$sd(u,v)$の最大値 ＝ 17
$sd(u,v)$の最小値 ＝ 5

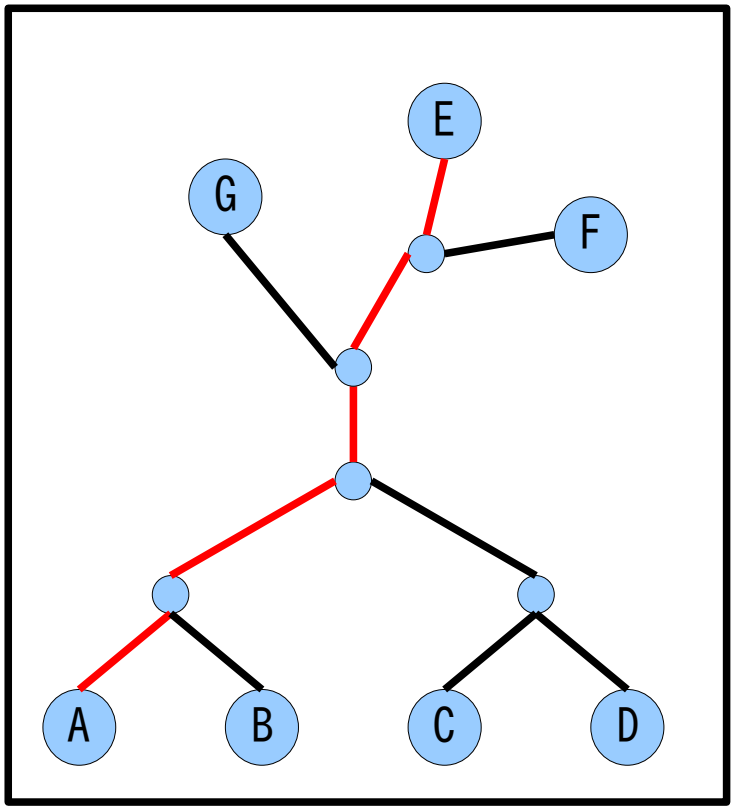






木の評価

$td(u,v)$を類似度の距離に対応させた距離を $ntd(u,v)$とする



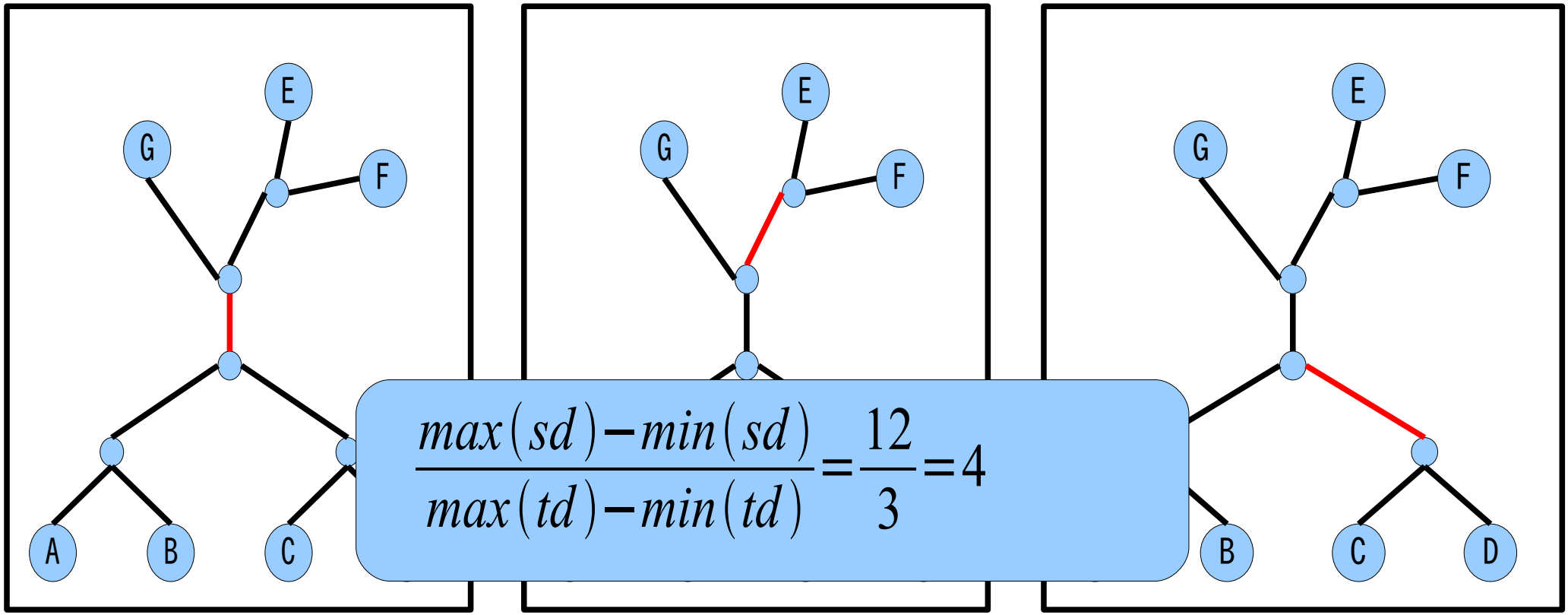
$sd(u,v)$の最大値 ＝ 17

$sd(u,v)$の最小値 ＝ 5

木の評価

$td(u,v)$を類似度の距離に対応させた距離を $ntd(u,v)$とする

$sd(u,v)$の最大値 ＝ 17
$sd(u,v)$の最小値 ＝ 5

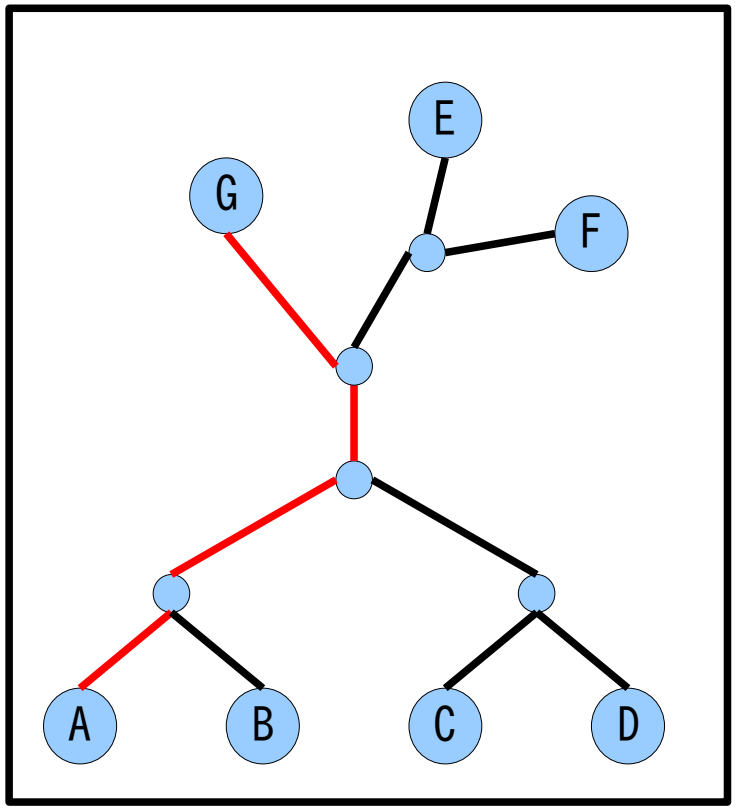


$$\frac{max(sd)-min(sd)}{max(td)-min(td)}=\frac{12}{3}=4$$

木の評価

$td(u,v)$を類似度の距離に対応させた距離を$ntd(u,v)$とする

$sd(u,v)$の最大値 ＝ 17
$sd(u,v)$の最小値 ＝ 5



$$ntd(A,G)=5+4\times2=13$$

木の評価

グラフを評価する値を TV_1, TV_2 と定義

$$TV_1 = \frac{2}{N(N-1)} \sum |sd(u,v) - ntd(u,v)|$$

平均値

$$TV_2 = \sqrt{\sum (sd(u,v) - ntd(u,v))^2}$$

2乗和の平方根

値が小さければ $sd(u,v)$ からより反映したグラフである

木の評価

$$TV_1 = \frac{2}{N(N-1)} \sum |sd(u,v) - ntd(u,v)|$$

平均

		Remake
Preprocess2	NJ	0.130395
	UPGMA	0.162626
	Quartet Method	0.177194

＊ 赤は同じ距離表から作った中で、評価が良いもの。青は2番目。

木の評価

$$TV_2=\sqrt{\sum\left(sd(u,v)-ntd(u,v)\right)^2}$$

2乗和の平方根

		Remake
Preprocess2	NJ	6.352560
	UPGMA	7.745910
	Quartet Method	8.268810

＊ 赤は同じ距離表から作った中で、評価が良いもの。青は2番目。

木の評価

S(T)はQuarutet Method が良かった

		Remake
Preprocess2	Quartet Method	0.850286
	NJ	0.455565
	UPGMA	0.455565

TV は NJ法 が良かった

		Remake
Preprocess2	NJ	6.352560
	UPGMA	7.745910
	Quartet Method	8.268810

Quartet method は $sd(u,v)$ をあまり反映していない

木の評価

Quarutet Methodについて着目

Quartet methodの改良案

TVの評価を上げて行くプログラムを追加

TVはTV_2を使用する

木の評価

Quartet method(改)

S(T) ＞ 0.85

TV_2をより小さくしていく

実験結果

実行時間　　約85時間　　(2/8 15：00 継続中)

S(T) ＝ 0.780169

TV_2 ＝ 4.5571

TV2が小さくなり過ぎて

S(T)が更新できない

木の評価

Quartet method(改)

S(T) > 0.85

TV_2 < 6.5　　緩和

実験結果

実行時間　　14時間程度

S(T) ＝　　0.850452

TV_2 ＝　　6.46065

木の

Quar

S(T)

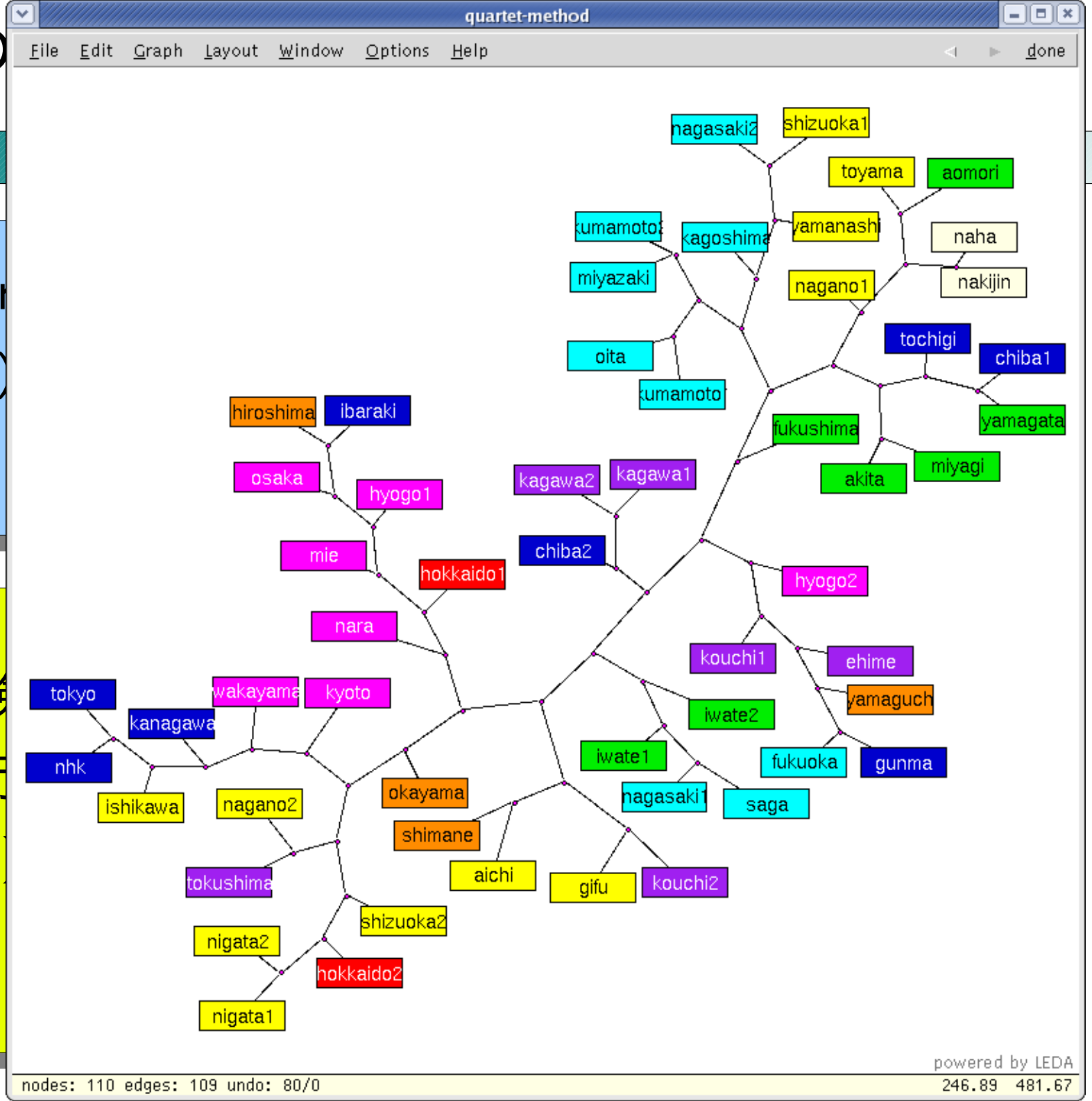
TV_2

実験

実行

S(T)

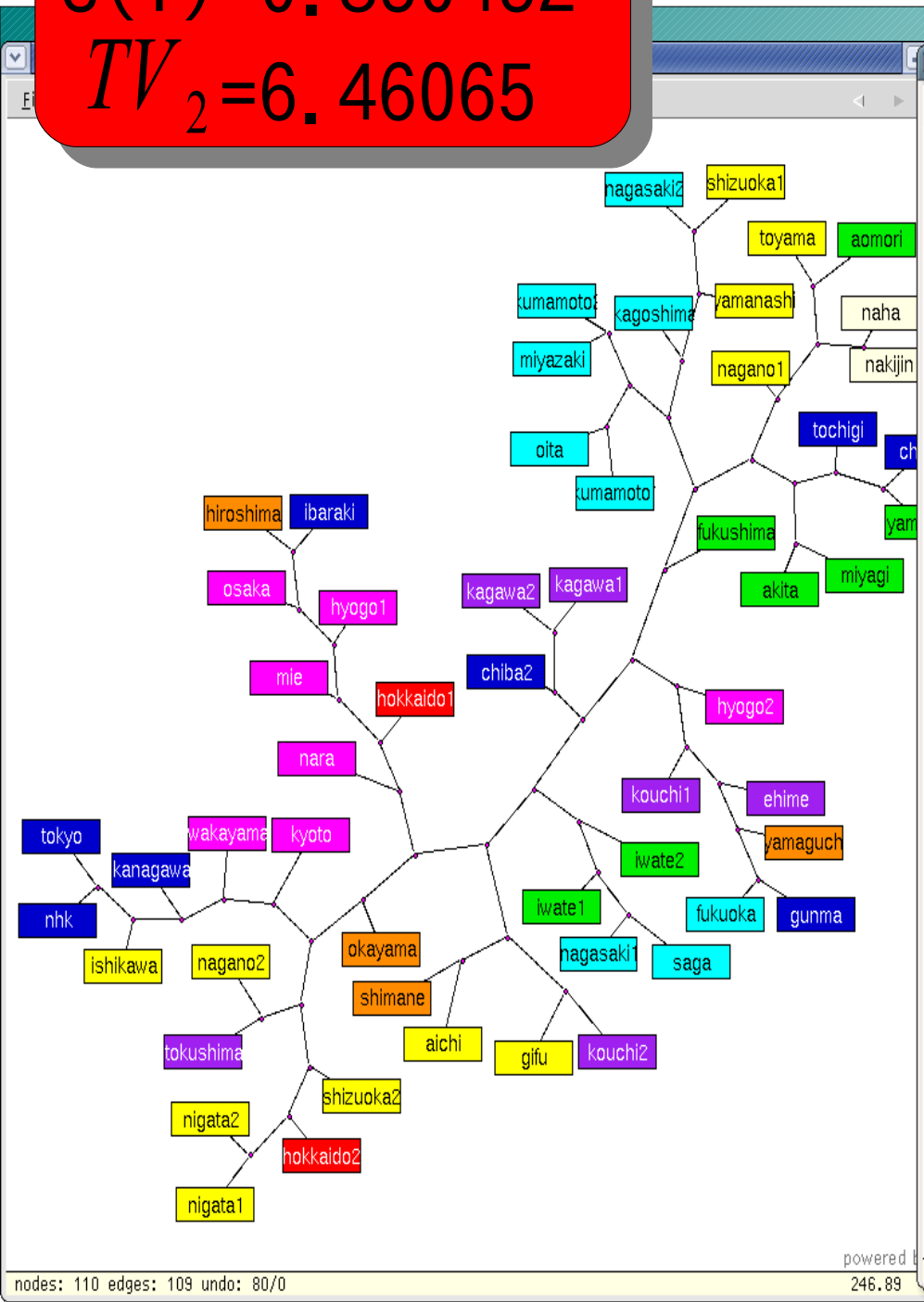
TV_2



木の評価

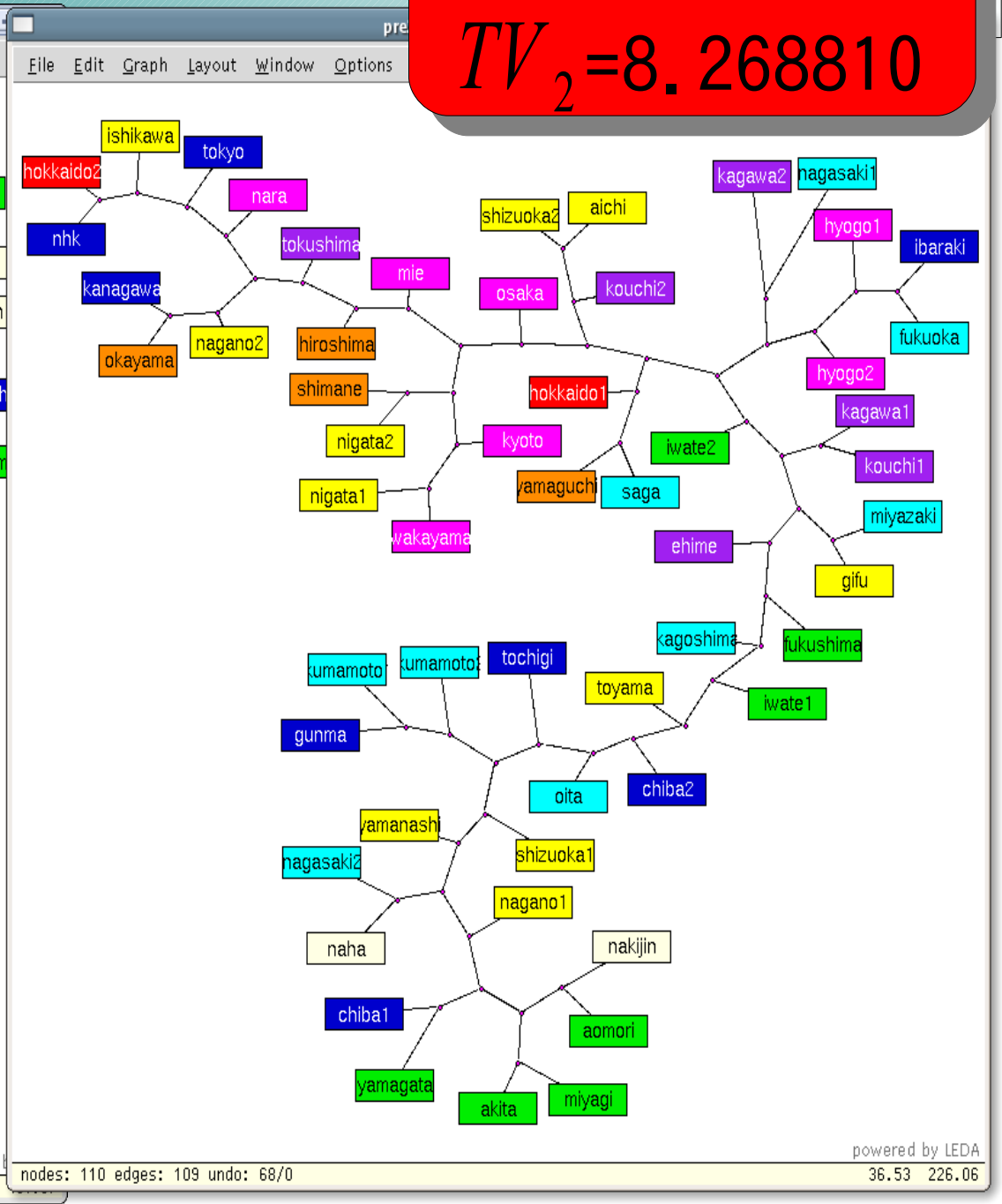
$S(T)=0.850452$

$TV_2=6.46065$



$S(T)=0.850286$

$TV_2=8.268810$

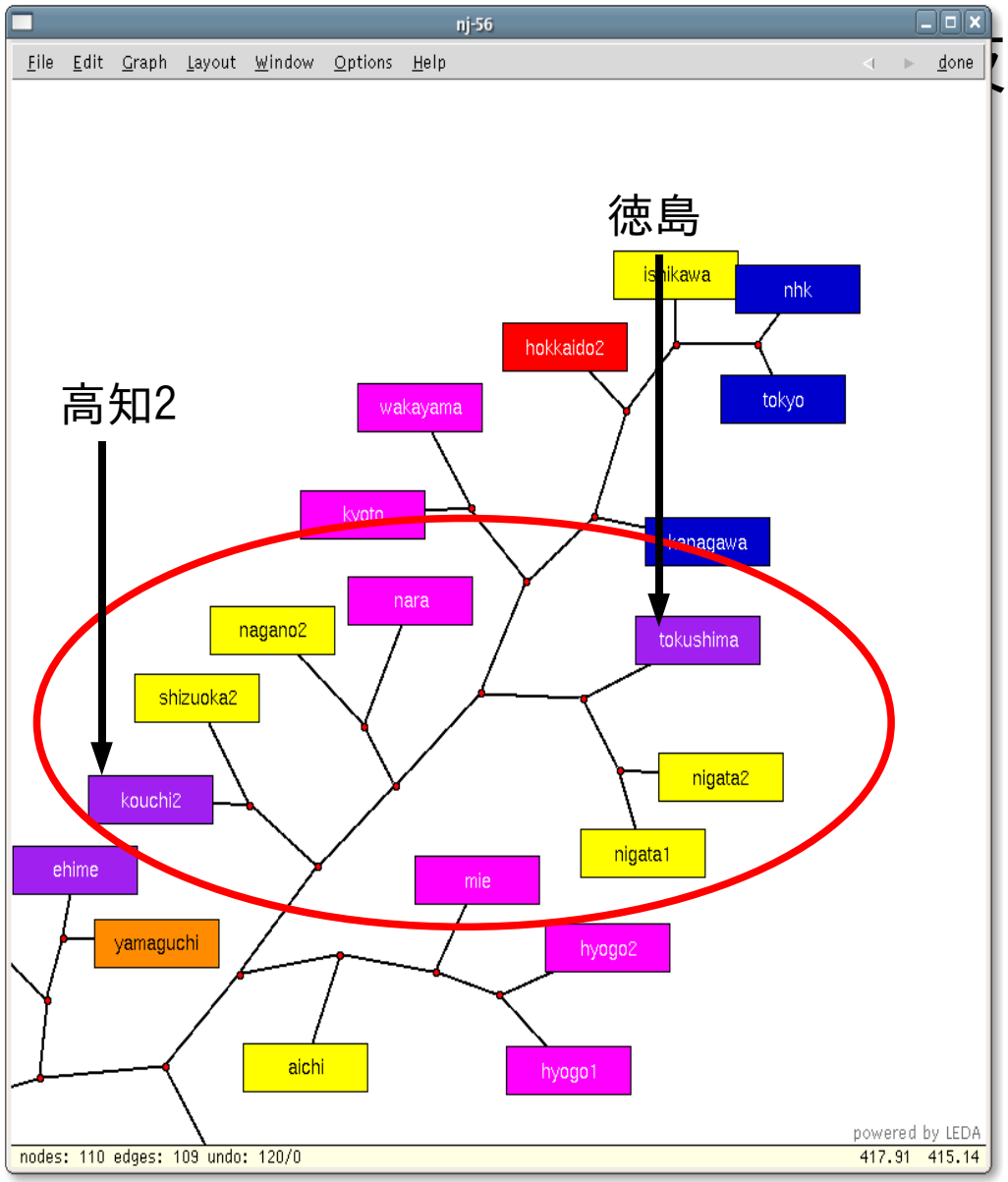


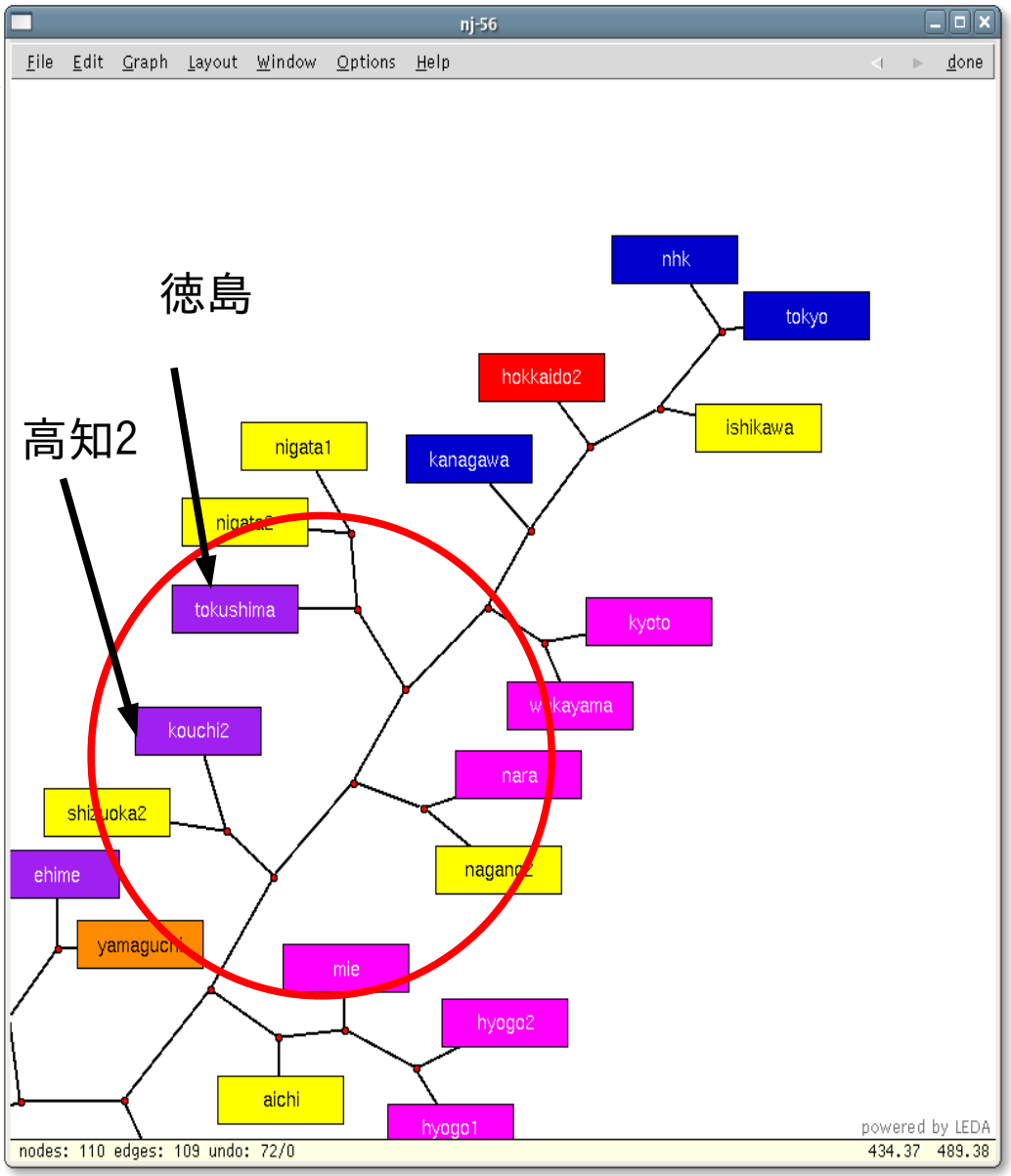
まとめサイト

今後の課題

- 対象データの改良
- 他の階層型クラスタリングを試す
- 考案されているクラスタリング評価する指標を試す
- 描画の問題

今後の課題


nj-56
File Edit Graph Layout Window Options Help done
徳島
高知2
ishikawa
nhk
hokkaido2
tokyo
wakayama
kyoto
kanagawa
nara
nagano2
tokushima
shizuoka2
nigata2
kouchi2
nigata1
ehime
mie
yamaguchi
hyogo2
aichi
hyogo1
powered by LEDA
nodes: 110 edges: 109 undo: 120/0
417.91 415.14


nj-56
File Edit Graph Layout Window Options Help done
徳島
高知2
nhk
tokyo
hokkaido2
ishikawa
nigata1
kanagawa
tokushima
kyoto
kouchi2
nara
ehime
mie
hyogo2
aichi
hyogo1
powered by LEDA
nodes: 110 edges: 109 undo: 72/0
434.37 489.38

今後の課題

- 対象データの改良
- 他の階層型クラスタリングを試す
- 考案されているクラスタリング評価する指標を試す
- 描画の問題

終わり

御静聴ありがとうございました

木の評価

Quartet method(改)

S(T) > 0.85

TV_2をより小さく || TV_2< 5　緩和

実験結果

実行時間　10時間程度

S(T) ＝　0.790128

TV_2　＝　4.98911

距離の比較

nhkからの距離(方言の読本)				ももたろう		
順位	都道府県	距離		地域	距離	順位
1	nhk_nigata	0.576923	←差→	nigata2	0.422460	10
2	nhk_osaka	0.576923	← →	osaka	0.453202	13
3	nhk_hiroshima	0.576923	← →	hiroshima	0.436620	11
4	nhk_tokyo	0.607407	← →	tokyo	0.208556	1
5	nhk_aichi	0.615385	← →	aichi	0.516529	23
6	nhk_nagasaki	0.616541	← →	nagasaki1	0.525114	25
7	nhk_aomori	0.623077	← →	aomori	0.741935	52
8	nhk_kouchi	0.623077	← →	kouchi2	0.466019	16
9	nhk_wakayama	0.651852	← →	wakayama	0.368421	7
10	nhk_hokkaido	0.661654	← →	hokkaido2	0.264550	2
11	nhk_miyagi	0.669118	← →	miyagi	0.686916	49
12	nhk_ishikawa	0.669231	← →	ishikawa	0.272727	3
13	nhk_okinawa	0.689189	← →	naha	0.752033	54

距離の比較

nhkからの距離				標準語形の全国分布		
順位	都道府県	距離		順位	都道府県	平均分布率
1	nhk_nigata	0.576923	←差→	31	新潟	3 %
2	nhk_osaka	0.576923	← →	21	大阪	40
3	nhk_hiroshima	0.576923	← →	33	広島	34.4
4	nhk_tokyo	0.607407	← →	1	東京	61.6%
5	nhk_aichi	0.615385	← →	12	愛知	47.5%
6	nhk_nagasaki	0.616541	← →	41	長崎	25.5%
7	nhk_aomori	0.623077	← →	44	青森	22.1%
8	nhk_kouchi	0.623077	← →	24	高知	38.2%
9	nhk_wakayama	0.651852	← →	16	和歌山	43.6%
10	nhk_hokkaido	0.661654	← →	7	北海道	53.8%
11	nhk_miyagi	0.669118	← →	38	宮城	31.0%
12	nhk_ishikawa	0.669231	← →	35	石川	31.7%
13	nhk_okinawa	0.689189	← →	48	沖縄	3.3%
14	nhk_kagoshima	0.692308	← →	47	鹿児島	16.1%
15	nhk_chiba	0.693431	← →	10	千葉	52.5%

標準語形の全国的分布(河西秀早子)

『日本言語地図』全6巻300枚の中から地域の方言の特色が出ていると思われる計82枚の地図から標準語形の分布率の順位をつけたもの

木の評価

Quartet method(改)

S(T) > 0.85

TV_2をより小さく || TV_2< 5　緩和

実験結果

実行時間　10時間程度

S(T) ＝　0.790128

TV_2　＝　4.98911

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。

$x = y$　ならば　$D(x, y) = 0$

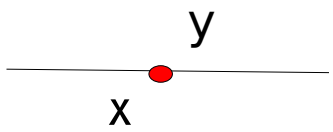
$D(x, y) + D(y, z) \geq D(x, y)$　（三角不等式）

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

距離の定義

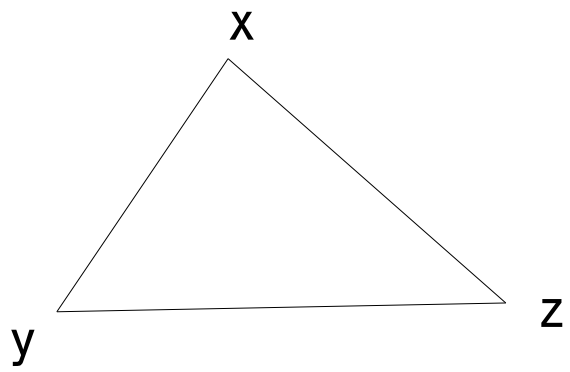
X：集合　　x、y、z：Xの元
D：距離関数

$x=y$ ならば $D(x,y)=0$



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

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



Kolmogorov記述量の定義

S.....プログラムミング言語

$|p|$ ...プログラムのサイズ

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

yに含まれる xの情報量

$K(x)$ ： xの記述量

$K(y|x)$ ： xを用いたyの記述量

$K(xy)$ ： xとyの記述量

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

$$I(x:y)-I(y:x)<c \quad \Rightarrow \quad I(x:y)=I(y:x)$$

条件付Kolmogorov記述量

U ： データを記述する言語

$|p|$ ： 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 $K(x|y)$

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

と定義する

Kolmogorov 記述量

83	86	77	15	93	35	86	92	49	21
62	27	90	59	63	26	40	26	72	36
11	68	67	29	82	30	62	23	67	35
29	2	22	58	69	67	93	56	11	42
29	73	21	19	84	37	98	24	15	70
13	26	91	80	56	73	62	70	96	81
5	25	84	27	36	5	46	29	13	57
24									
87									
32									

```
#include<stdio.h>
int main(){
  int i,j;
  for(j=0;j<10;j++){
    for(i=0;i<10;i++){
    printf(" %2d ",random()%100);
    }
    printf("\n");
```

どのプログラム言語を選んでも情報量は
定数分の差しかないことが証明されている

データ x のKolmogorov 記述量

最小のプログラムサイズ(ビット数)と定義 $K(x)$

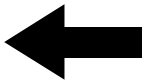
Kolmogorov 記述量

補助情報yに対するデータxのKolmogorov 記述量

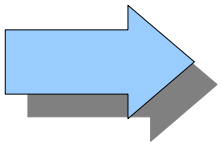
x:1111100000111111000001111110000011111100000111111000000

y:1111100000

```
#include<stdio.h>
int main(){
  int i,y=1111100000;
   for(i=0;i<5;i++){
    printf("%d",y);
   }
  return(0);
}
```



補助情報

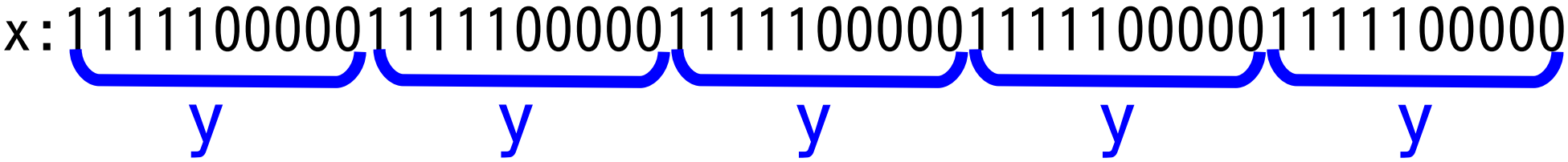


$$K(x|y)$$

条件付Kolmogorov 記述量

Kolmogorov 記述量

補助情報yに対するデータxのKolmogorov 記述量



y:1111100000

$$K(x|y)$$

条件付Kolmogorov 記述量

Similarity metric

Ming Liらの研究で類似度距離が提案された

$$d(x,y)=\frac{max\{K(x|y),K(y|x)\}}{max\{K(x),K(y)\}}$$

$K(x)$ ：xの記述量

$K(y|x)$ ：xを用いたyの記述量

$K(xy)$ ：xとyの記述量

$K(y)\geq K(x)$のとき

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

$O(logK(xy))$の誤差範囲

$$K(xy)=K(x)+K(y|x)$$

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