

Bioinformatics とソフトウェア

(株)数理システム 田辺隆人(Takahito Tanabe)
Mathematical Systems Inc.

Cypripedium



Lady's slipper orchid

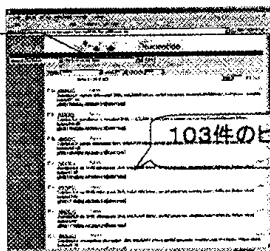
主要な配列データベース

- GenBank (<http://www.ncbi.nlm.nih.gov/>)
- EMBL (<http://www.ebi.ac.uk/emb/>)
- DDBJ (<http://www.ddbj.nig.ac.jp/>)
- PIR (<http://www-nbrf.georgetown.edu/>)
- EXPASY (<http://www.expasy.ch/>)
- SRS (<http://srs6.ebi.ac.uk/>)

データベース検索

- NCBI, ヌクレオチドデータベース

"Cypripedium
Japan"
で検索



データベース検索

- 個々のアイテムの概要

country:Canada,Prince Edward Island, isolate:AC-02
gi|13517201|gb|AB056315.1|[13517201]

4. AB056314 Reports
Cypripedium sp. U-03 chloroplast DNA, trnL(UAA) intron, partial sequence,
country:Japan,Hokkaido, Rebun Island, isolate:U-03
gi|13517200|gb|AB056314.1|[13517200]

5. AB056313 Reports
Cypripedium sp. U-02 chloroplast DNA, trnL(UAA) intron, partial sequence,

データベース検索

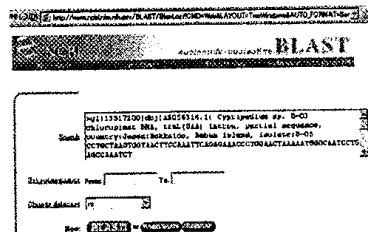
- 配列 (FASTA形式)

4. AB056314 Reports Cypripedium sp. U.
gi|13517200

```
>gi|13517200|gb|AB056314.1| Cypripedium sp. U-03 chloroplast DNA, trnL(UAA) intron, partial
CCTGCTAAGTGGTAACCTCCAAATTCAGAGAAACCCCTGGAACTAAAATGGGCAATCCCGAGCCAAATGT
TTGTTTTATAAAATGCAAAATAGATAAAAAGGGAAGGTCCAGAGACTCAATGGAACTGTCTA
ACGAATCAAAATTAAGTAAATGGAAGATTTATCTCAATCCATTCGAAATTTGAAGGAA
TAGAATCAAAATTAAGTAAATGGAAGATTTATCTCAATCCATTCGAAATTTGAAGGAA
AAAGTTAATGGCAGAGATAAAGAGAGACTCCCATTTTACATGCAATATCGACAAATGAAA
```

アラインメント

- BLAST (類似の配列の検索)



漸化式

Needleman-Wunschアルゴリズム

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

$F(i, 0) = -id$
 $F(0, j) = -jd$

一致するとき
のスコア


どちらかを
飛ばしたときの
ペナルティ

開始点をずらした
ときのペナルティ

インパクト

- 進化の経路の解析

シロイヌナズナ
Arabidopsis thaliana



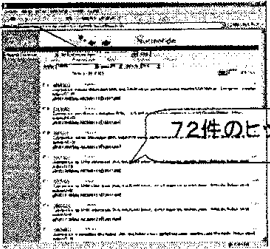
シロイヌナズナは、世代経路が短くゲノムがコンパクトなため遺伝率のモデル植物として広く使われ、多数の突然変異、クマクローン、形質転換系統の資源が利用可能です。シロイヌナズナのゲノムやRNAの正確な配列情報を得ることで、ゲノムの全体的な変異が明らかになります。ゲノムがコンパクトであるため、遺伝的変異の検出が容易で、ゲノム全体の解析が容易です。シロイヌナズナは、ゲノムがコンパクトであるため、ゲノム全体の解析が容易です。ゲノム全体の解析が容易です。ゲノム全体の解析が容易です。

アラインメント解析で69%の遺伝子機能が判明
(実験的には9%)

データベース検索

- NCBI, 構造データベース

Chloroplast
で検索



72件のヒット

データベース検索

- 個々の結果

Crystal Structure Of Mutant S188a Of Photosynthetic Glyceraldehyde-3-Phosphate Dehydrogenase A4 Isoform, Complexed With NADp

Description: Crystal Structure Of Mutant S188a Of Photosynthetic Glyceraldehyde-3-Phosphate Dehydrogenase A4 Isoform, Complexed With NADp.

Deposition: F. Sparta, S. Ferman, G. Fakri, A. Ripamonti, P. Sabatini, P. Pupillo & P. Trost, 27-Nov-03

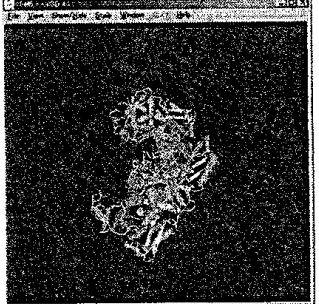
Taxonomy: [Slu03321.0193038](#)

Reference: [PubMed](#) [MMDB: 28549](#) [PDB: 1RLH5](#)

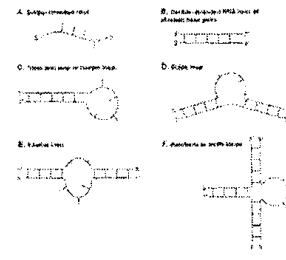
ほうれん草

データベース検索

- 三次構造



二次構造特定 RNAfolding



二次構造特定 入力データ

- RNA塩基配列
- 二塩基間結合のエネルギー
- 各種ループ構造のエネルギー

Table 1. Free energy parameters for RNA secondary structure prediction.

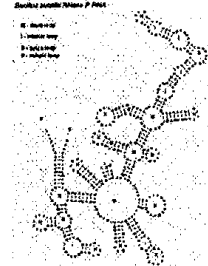
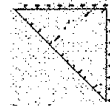
A. Stacking energy per base pair		B. Loop initiation energy	
Base pair	Energy (kcal/mol)	Loop size	Energy (kcal/mol)
AA	-1.0	3	1.2
AG	-1.0	4	1.4
AA	-1.0	5	1.6
AG	-1.0	6	1.8
AA	-1.0	7	2.0
AG	-1.0	8	2.2
AA	-1.0	9	2.4
AG	-1.0	10	2.6
AA	-1.0	11	2.8
AG	-1.0	12	3.0
AA	-1.0	13	3.2
AG	-1.0	14	3.4
AA	-1.0	15	3.6
AG	-1.0	16	3.8
AA	-1.0	17	4.0
AG	-1.0	18	4.2
AA	-1.0	19	4.4
AG	-1.0	20	4.6
AA	-1.0	21	4.8
AG	-1.0	22	5.0
AA	-1.0	23	5.2
AG	-1.0	24	5.4
AA	-1.0	25	5.6
AG	-1.0	26	5.8
AA	-1.0	27	6.0
AG	-1.0	28	6.2
AA	-1.0	29	6.4
AG	-1.0	30	6.6

B. Destabilizing energies for loops

Number of bases	1	5	10	20	30
Loop	0.0	0.5	1.0	1.5	2.0
Stack	0.0	0.0	0.0	0.0	0.0

二次構造特定 アルゴリズム

- エネルギー最小化



二次構造特定 アルゴリズム

$$W(k, j) = \min\{W(k+1, j), W(k, j-1), V(k, j), \min_{1 \leq k < j} \{W(k, k) + W(k+1, j)\}\} \quad (1)$$

$$\text{and } V(k, j) = \min\{W(k, j) + V(k+1, j-1), VM(k, j), VM(k, j)\} \quad (2)$$

where

$$VM(k, j) = \min_{\substack{1 < i < j < k \\ i+k+j-j > 2}} \{VM(k, j, i, j) + V(i, j)\} \quad (3)$$

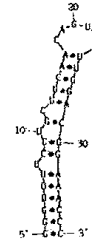
and

$$VM(k, j) = \min_{1 < k < j-1} \{W(k+1, k) + W(k+1, j-1)\} \quad (4)$$

The minimum folding energy, E_{min} , is given by $W(1, n)$.

○ エネルギー最小の値
2-loopまで考慮

二次構造の特定 mfoldサーバー



"GGGUUUUCCUGCUCAACAG
UGCUUGGACGGAAACCC"に對
する mfoldサーバーの応答

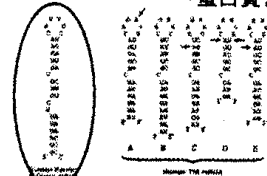
<http://www.bielefeld.uni-leipzig.de/applications/mfold/si/rna/fercal.cgi>

$\Delta G = -10.1$ (initially -10.1) ferritin

インパクト RNA機能調整の解明

- 血中フェリチン
- IRE(28ゲノム)にIRE結合蛋白が結合
→ mRNA安定化

Examples of IREs



→蛋白質を大量に合成

特徴的な構造

三次構造特定 入出力

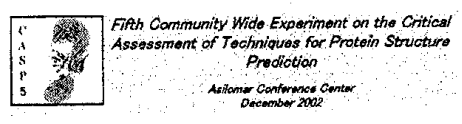
```

RIGIN
1 mntskdskps ydfdlilias szgflaako aakfdkxv ldfvtlplg twnglgtov
81 nvgcipkka hqaal lcaol kdsnygkkl edtkvhdok atesvohig slngprval
121 rakovvyna ygfifpshk matnreklak vsyaerfla tserprylai sedkayciss
181 ddflyoyoo gktlvgsay valocagfla gfgldvtvw rasilrgfdq dmnkligahh
241 ehhkifira fvothoale aglgrfkt skotnsaal edefovilla gsdactrl
301 giatvyrkin aktkispvd aactvoviy aladi lekl eltpvalag zllarlygg
381 slvkcdydv pttvtlplv acaalsska vektgsenie vyhaffwle atvsrdnak
421 cyakvlenik dnervgthv lgnagevlg gfaalilogl tkqaldstg ihpucseift
481 tlvsvkrsg dliesgoc
    
```



三次構造特定

- コンペティション



三次構造特定 アルゴリズム

- 断片(3-9残基)の検索
- MonteCarlo法
- Superfamily に特徴的な構造情報利用
- ポテンシャル関数(疎水性・free-energy. . .)
- 相同解析の結果を利用
- Knowledge base 検索

帰納的な方法が不可欠

機械学習 ソフトウェア

- ソフトウェア
HMMpro(隠れマルコフモデル解析ソフト)

開発元: Net-ID Inc.

- 応用
 - 遺伝子発見
 - 構造情報の取得
 - 相同解析

隠れマルコフモデル

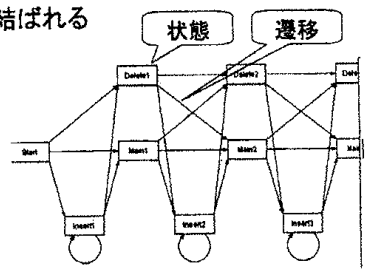
- 記号列
 - DNA/RNAアルファベット(4種)
 - A = Adenine C = Cytosine
 - G = Guanine T = Thymine/U = Uracil
 - アミノ酸(20種)

A = Alanine (Ala)	R = Arginine (Arg)
D = Aspartic Acid (Asp)	C = Cysteine [Cys]
Q = Glutamine (Gln)	E = Glutamic Acid (Glu)
N = Asparagine (Asn)	G = Glycine (Gly)
H = Histidine (His)	I = Isoleucine (Ile)
L = Leucine (Leu)	K = Lysine (Lys)
M = Methionine (Met)	F = Phenylalanine (Phe)
P = Proline (Pro)	S = Serine (Ser)
T = Threonine (Thr)	W = Tryptophan (Trp)
Y = Tyrosine (Tyr)	V = Valine (Val)

状態遷移確率 + 出力確率

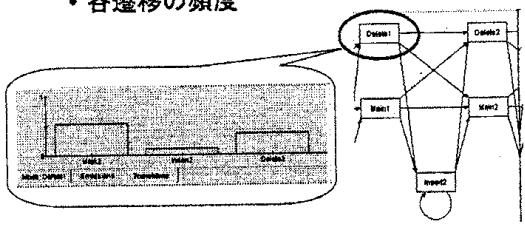
状態列

- 遷移で結ばれる



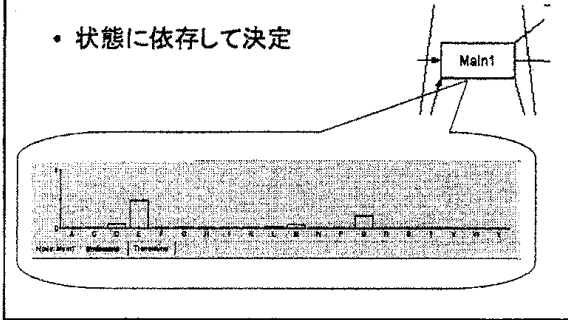
状態遷移確率

- 各遷移の頻度



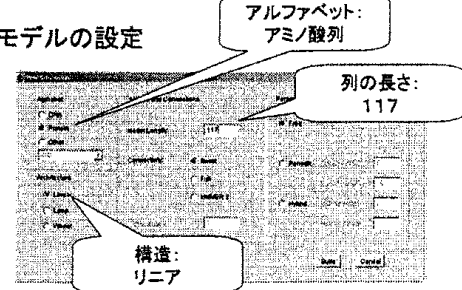
出力確率

- 状態に依存して決定



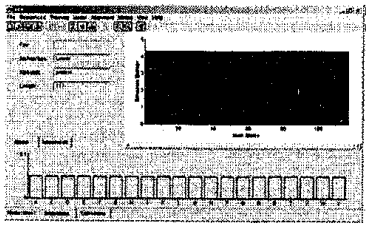
マルコフモデルの学習

- モデルの設定



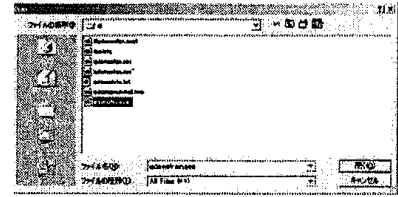
学習前

- 出力分布



学習元ファイルのインポート

- ファイル選択



学習の実施

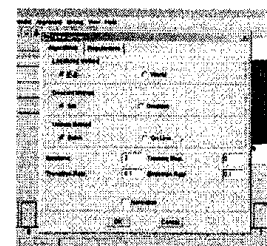
- 教師データ

```

ObjectData:
DAT: string
MKLPVRLVLMFWIPASSSDVVMTOITPLSLVSLGDQASISCRSSOSLVHSHGNTYLNWYLO
KAGOSPKLLYKVNRFSGVDFRFSGSGSOTDFTLKISKVBAEDLGIYFCSQITTHVPTFGGGT
KLEIKR
LOOPGAEIVKPGASVILSCKASGYTFYTWYVWVYKQKRGLEWIGRIDPNSGGTRYNEKF
KQKATLTKNSNTAYMQLSSLISDDSAVYYCARGDYSYAMDYWGQGTIVTSS
ESGGGLVQPGSMKLCVASCFTFSNYWVWVYRQSPKGLWVAERLKSQYATHYAESV
KGRFTSRDSDSSVYLOMNNLKAEDTGIYCTRPGVDPYWGQGTILVSS
MEFGLSWELVALKQVQCEVRLVSGGDLVEPGSLKVSCEVSGPFSKAWMNWVYRQAFG
KOLQVVGQIKNYGQGTIDYAAVYVGGREKRDSDKSTVYLOMNNLKAEDTAVYYCYQNYT
GTYDYWGQGTILVTVSS
ISCKASGYTFYTWYVWVYKQKRGLEWIGRIDPNSGGTRYNEKF
YLOMNNLKAEDTAVYYCYQNYTAAVYVGGREKRDSDKSTVYLOMNNLKAEDTAVYYCYQNYT
LVLQGSQPLVVKPFTSMKISCKTSGYSFTGYTMSWVROSHGKSLWIGLIPNSGGTINYQ
KPKDKASLTVDKSSSTAYMELLSLTSDSAVYYCARPSYVYGRNTYVWVYKQKRGLEWIGRIDPNSGGTIVTSS
SAK
    
```

学習の実施

- アルゴリズム・パラメータ設定



全文検索 索引作成

- サフィックスとインデクスポイント

Text	a	b	r	a	c	a	d	a	b	r	a
Index	0	1	2	3	4	5	6	7	8	9	10

Suffix	Index
a b r a c a d a b r a	0
b r a c a d a b r a	1
r a c a d a b r a	2
a c a d a b r a	3
d a b r a	4
a b r a	5
c a d a b r a	6
a d a b r a	7
a b r a	8
r a	9
a	10

全文検索 索引作成

- ソート

Sorted Suffix	Index
a	10
a b r a	7
a b r a c a d a b r a	0
a c a d a b r a	3
a d a b r a	5
b r a	8
b r a c a d a b r a	1
c a d a b r a	4
d a b r a	6
r a	9
r a c a d a b r a	2

全文検索 索引の利用

- 二分探索

Sorted Suffix	Index
a	10
a b r a	7
a b r a c a d a b r a	0
a c a d a b r a	3
a d a b r a	5
b r a	8 ← 1
b r a c a d a b r a	1
c a d a b r a	4
d a b r a	6 ← 2
r a	9 ← 3
r a c a d a b r a	2

全文検索 索引作成アルゴリズム

- ブロックソートに帰着

入力文字列 + \$	ソート	Suffix array (索引)
- \$abcabc		- \$abcabc
0 abcabc\$		3 abc\$abc
1 bcabc\$a		0 abcabc\$
2 cabcb\$bc		4 bc\$abca
3 abc\$abc		1 bcabc\$a
4 bc\$abca		5 c\$abcab
5 c\$abcab		2 cabcb\$bc

\$: いずれの文字よりも小

ブロックソート

入力文字列

0 AbcABC	3 ABCAbc
1 bcABCA	0 AbcABC
2 cABCAb	4 BCAbcA
3 ABCAbc	5 CAbcAB
4 BCAbcA	1 bcABCA
5 CAbcAB	2 cABCAb

ソート

入力が特殊なソート

ブロックソートによる変換 Burrows-Wheeler 変換

- 着眼

入力文字列

0 AbcABC	3 ABCAbc
1 bcABCA	0 AbcABC
2 cABCAb	4 BCAbcA
3 ABCAbc	5 CAbcAB
4 BCAbcA	1 bcABCA
5 CAbcAB	2 cABCAb

最終列と1番行の場所で全情報再現可能

Burrows-Wheeler 変換 逆変換

- 入力

Burrows-Wheeler 変換 逆変換

- ソートによって最初の行を知る

Burrows-Wheeler 変換 逆変換

- 行番号特定

$(i+1)$ 番行の最終文字
= i 番行の先頭文字

Burrows-Wheeler 変換 逆変換

- Suffix array 及び 元の文字列判明

suffix array ・元の文字列判明

Burrows-Wheeler 変換 メリット

```

t: hat acts like this:<13><10><1
t: hat buffer to the constructor
t: hat corrupted the heap, or wo
W: hat goes up must come down<13
t: hat happens, it isn't likely
w: hat if you want to dynamical
t: hat indicates an error.<13><1
t: hat it removes arguments from
t: hat looks like this:<13><10><
t: hat looks something like this
t: hat looks something like this
t: hat once I detect the mangled
    
```

繰り返しが見れやすい→符号化効率良好

圧縮と検索の両立

ブロックソート

↓

BWTによる高性能圧縮
+
suffix arrayによる高速全文検索

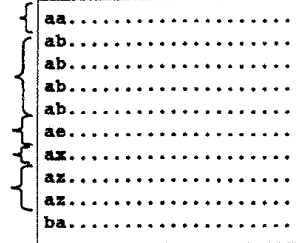
ブロックソート アルゴリズム

- quick sort
一般アルゴリズムなので向かない
- ternary partitioning[Bentley, Sedgewick 97]
無駄な文字列比較が少ない
- doubling algorithm
多くの場合最速
- copy method
対象の性質を利用
- layer method
copy method の改良

’04 田辺・小林

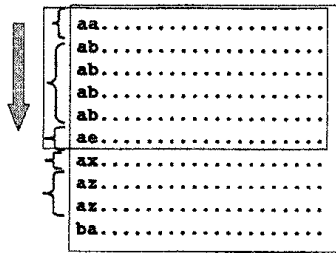
copy method 原理

- 先頭二文字のみでソートし、「区間」決定



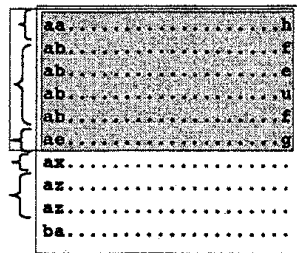
copy method 原理

- 先頭文字が同一の「区間」内を順にソート



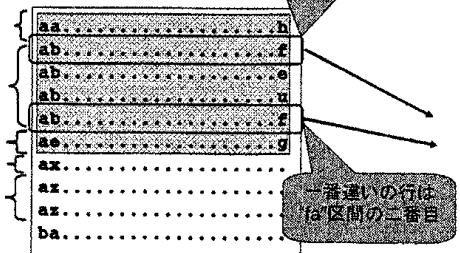
copy method 原理

- ソートされた区間内の末尾文字を調べる



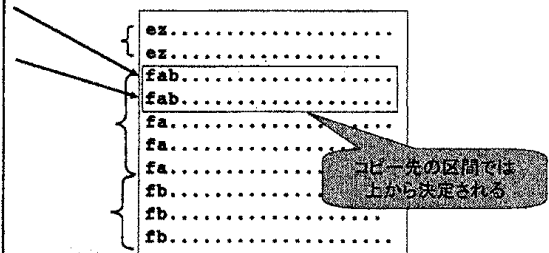
copy method 原理

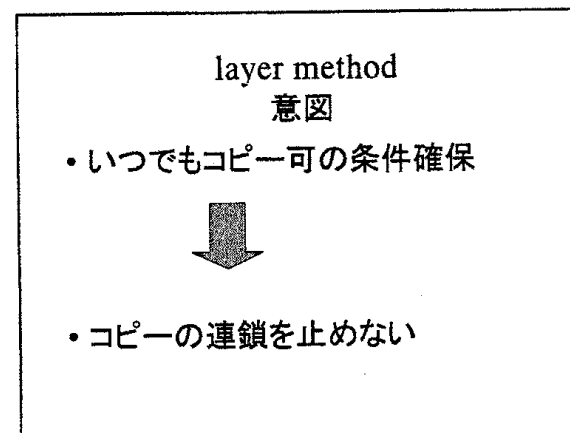
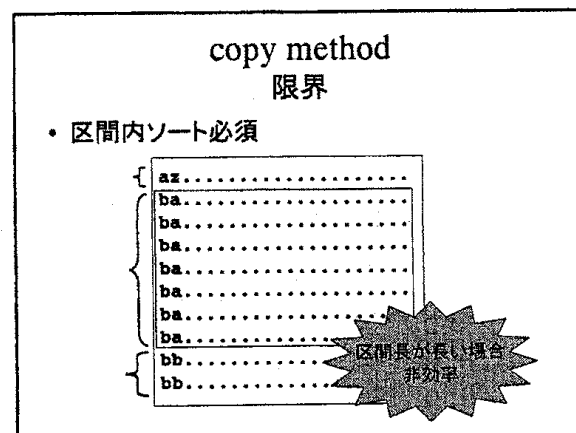
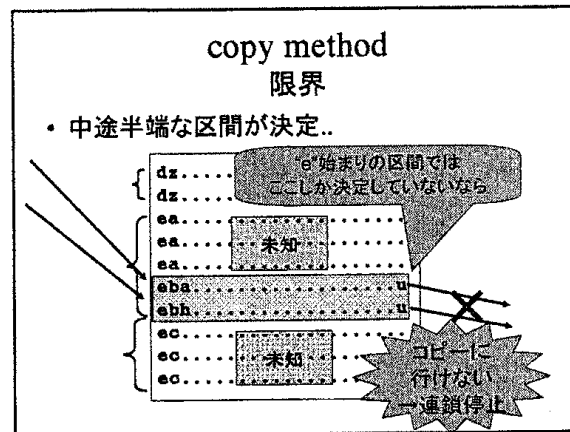
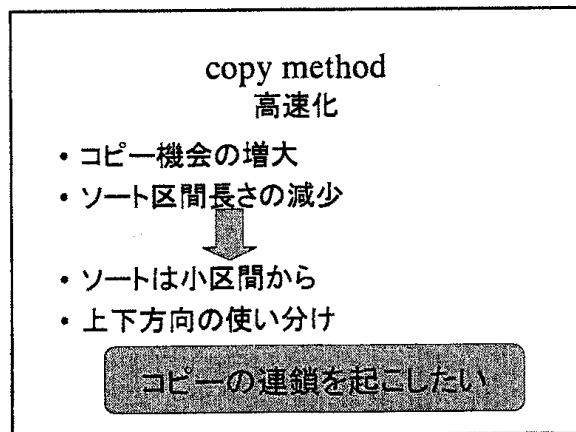
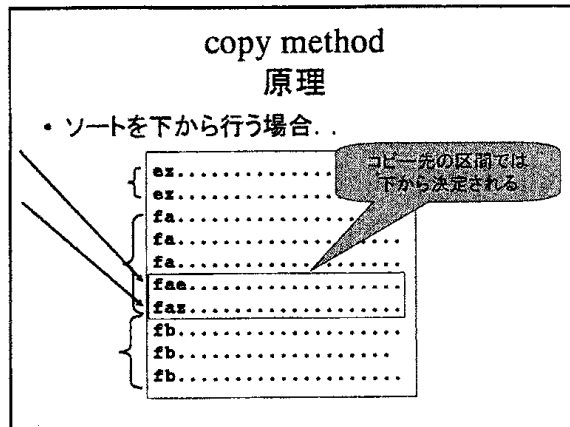
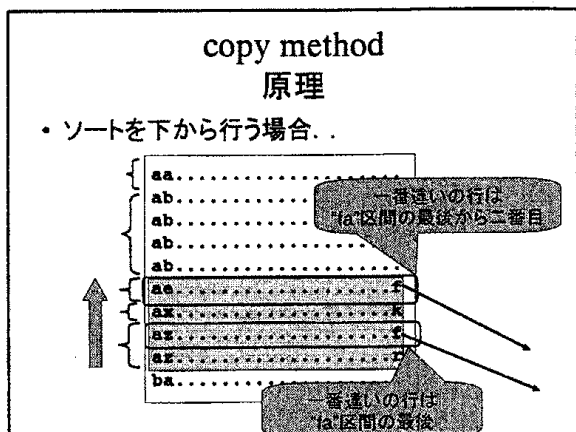
- 判明箇所を決定(コピー)



copy method 原理

- 判明箇所を決定(コピー)





layer method 着眼

- 区間内文字数カウントがあれば..

layer method 着眼

- 区間内分割ができれば軸はコピー可能

layer method 着眼

- 軸による区間分割
→新たな軸に対する区間分割コスト削減

layer method アルゴリズム

0. 文字数カウント設定, 初期区間と軸の決定
1. 軸に対する区間分割と軸のコピー
2. 文字数カウント更新
3. コピー回数 = 文字数なら終了
さもなければ
軸をコピー先区間でも軸に設定
1. に戻る

初期区間設定以外ソート不要

layer method 実装と今後

課題:
文字数カウントの効率的更新
アルゴリズムの理論的解析

変更余地:
軸を複数行に
区間分割機会
最小化のための軸設定

参考文献

- Suffix Array 説明
<http://namazu.org/~satoru/sary/docs/suffix-array.html>
- BW解説記事
<http://www.dogma.net/marky/articles/bwt/bwt.htm>
- K. Sadakane: A Modified Burrows-Wheeler Transformation for Case-insensitive Search with Application to Suffix Array Compression, To appear in Proc. of Data Compression Conference '99. (poster session) [600dpi ps.gz file](#) (36612 bytes), [PowerPoint file](#) (65536 bytes).
<http://naomi.is.s.u-tokyo.ac.jp/~sada/papers/index.html>

上記の記事の図表を使用させていただきました。