

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

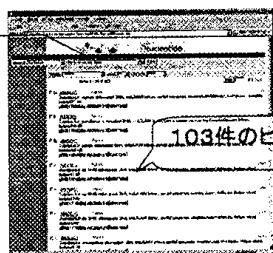
主要な配列データベース

- 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|dbj|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|dbj|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

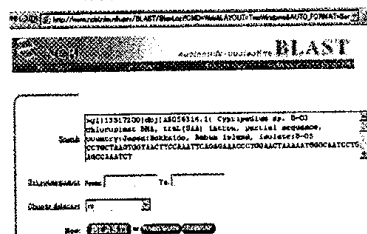
- 配列 (FASTA形式)

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

```
>gi|13517200|dbj|AB056314.1| Cypripedium sp. U-03 chloroplast DNA, trnL(UAA) intron, partial
CTGCTAAGTGGTAACCTCCAAATTCAGAGAAACCCCTGGAACTAAAAATGGGCAATCCCGAGCCAAATGT
TTGTTTTTATAAAAATGCAAAATAGATAAAAAAGGATAGGTCCAGAGACTCAATGGAACTGTTCTA
ACGAATGAATTCAGTCTTAAATGGAAGATTATTCTGCAATCCATTCCAAATGAAGTTGAAAAAGGAA
TAGAATTCAAATTCAGTCTTAAATGGAAGATTATTCTGCAATCCATTCCAGATTTGATAGATCTTTGAAAGTTAA
AAAGTTAATGGCAGGAGATTAAGAGAGAGCTCCGATTTACATGTCATATCCAGCAAGTGAAG
```

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

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) & \text{一致するときのスコア} \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

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

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

開始点をずらしたときのペナルティ

どちらかを飛ば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

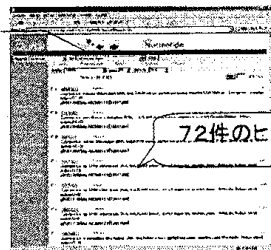
Arabidopsis thaliana

シロイヌナズナは、世代結実が早くゲノムがコンパクトなため遺伝率の高いモデル植物として広く使われ、多数の突然変異、クマクロソウ、形質転換系統等の資源が利用可能です。セントロメアやテロメアやRNAの位置配列情報を集めて、ゲノムの全塩基配列がほぼ決まっています。ゲノムデータベースの応用している基盤生物の中でも最もゲノム、全塩基配列が済んだのはシロイヌナズナです（新巻やGakopoulosではゲノムRNAメタデータが豊富に集まっています）。様々な植物生産現象の研究に使われていますが、質屋動物にまで共通する遺伝子メタデータ構築の発見にも使われています。

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

FIGURE 1
Crystal Structure Of Mutant S188a Of Photosynthetic Glyceraldehyde-3-Phosphate Dehydrogenase A4 Isoform, Complexed With NADp
[medd22548]

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

Deposition: F. Sparta, S. Ferman, G. F. F. A. R. P. Sabatini, P. Pupillo & P. Tosti, 27-Nov-03

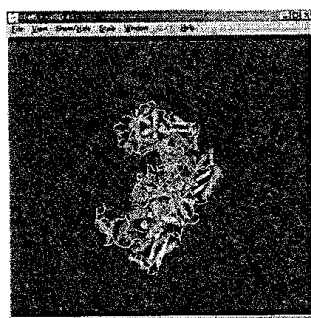
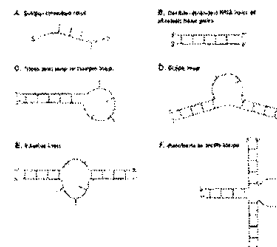
Taxonomy: [Succisa pratensis](#)

Reference: [PubMed](#) [MMDB](#) [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
RNA folding

三次構造特定

- コンペティション



Fifth Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction

Asilomar Conference Center
December 2002

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

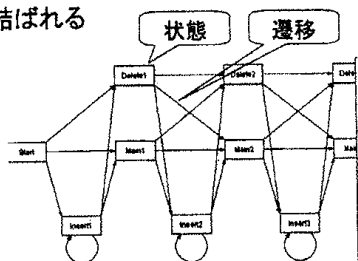
- アミノ酸(20種)

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

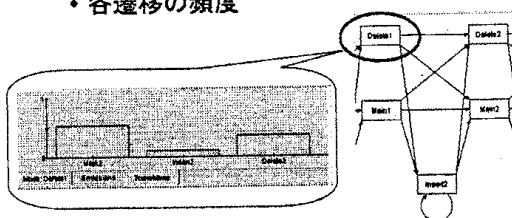
状態列

- 遷移で結ばれる



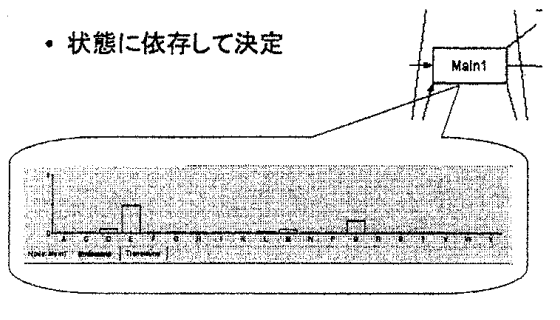
状態遷移確率

- 各遷移の頻度



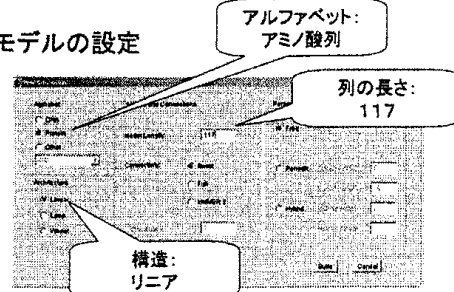
出力確率

- 状態に依存して決定



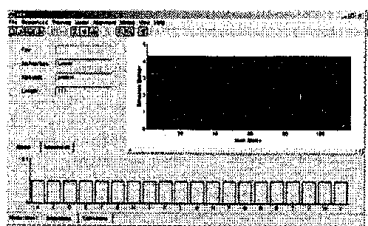
マルコフモデルの学習

- モデルの設定



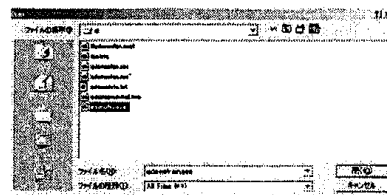
学習前

- 出力分布



学習元ファイルのインポート

- ファイル選択



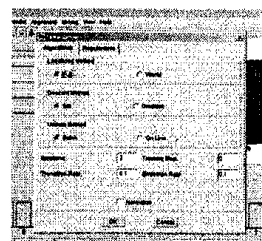
学習の実施

- 教師データ

ObjectData:
DATA: string
MKLPVRLVLMFWIPASSSDVVMTOITPLSLPVSLQDQASISCRSSOSLVHSHGNTYLNWYLO
KAGOSPKLLVYKVSNIPOVDFRPSGSGSOTDFTLKISKVBAEDLGIFCSQTHVPTFGGOT
KLEIKR
LOOPGAEVLKPGASVILCKASGYTFITNYWVWVKORPGRGLEWIGRIDPNSGGTKYNEKF
KNAATLTINKFNTAYMQLSSLTDDSAVYYCARGDYSYAMDYWGQOTSVTVSS
ESGGGLVQPGGSMKLSVCASGPTTSNYWMNVYROSPEKLEWVAERLKSGYATHYAESV
KGRFTSRDDSKSSVYLOMNNLKAEDTGIYYCTRPVDPDYWGQOTTLTVSS
MEPOLSWIELVAILKGOVCEVRLVSGGDLVEPGSLXVSCVSGPISKAWMNVYRQAPG
KOLQVVGOKNHYDQGTIDYAAVVGGRPERDDSKSTVYLQMNELKIEDTAVYYCYQNT
GTVDYWGGQTLTVTVSS
ISCKASGYTFITNYGMNVVVKQAPGKGLKWMGWINTYTGEPYADDFKGRFAFSLTSASTA
YLONNLKNBDATYPCARGSYDYIYAMDYWGQOTSVTVSS
LVOLQSGQPLVKPGTSMKISCKTSYSGFTGYTMSWVROSHGKSLWIGLIPSGGTNYNO
KPKDKASLTVDKSSSTAYMELLSLTSDDSAVYYCARPSYYSGRNYYAMDYWGQOTSVTVS
SAK

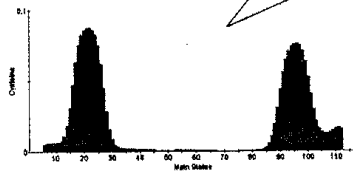
学習の実施

- アルゴリズム・パラメータ設定



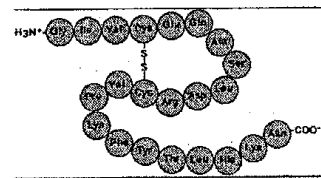
学習結果

- 学習結果 (Cysteine)



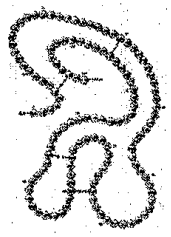
Disulfide bridge?

- 概念図



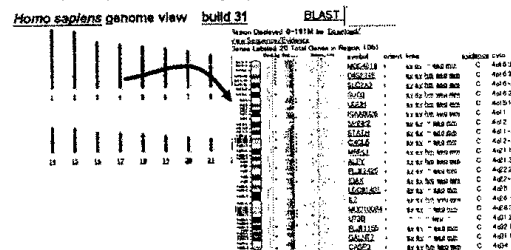
Disulfide bridge

- Lysozyme (4箇所出現)

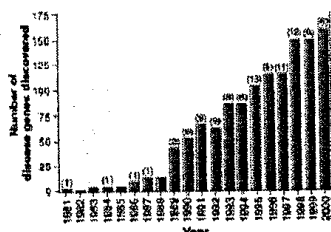


インパクト 遺伝子発見

- データベースに蓄積→遺伝子地図



インパクト 医療方面への応用



計算機科学への要請

- インタフェース
- 最適化
- 全文検索
- 圧縮

— Ugly programming instead of
deep biological insights

全文検索 索引作成

- サフィックスとインデックスポイント

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	
b	r	a	c	a	d	a	b	r	a		0
r	a	c	a	d	a	b	r	a		1	2
a	c	a	d	a	b	r	a		2	3	4
c	a	d	a	b	r	a		3	4	5	6
d	a	b	r	a		4	5	6	7	8	9
a	b	r	a		5	6	7	8	9	10	
b	r	a		6	7	8	9	10			
r	a		7	8	9	10					
a		8	9	10							
c		9	10								
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
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 索引作成アルゴリズム

- ブロックソートに帰着

入力文字列 + \$

Suffix array
- \$abcabc
0 abcabc\$
1 bcabc\$a
2 cabcb\$bc
3 abc\$abc
4 bc\$abca
5 c\$abcab

ソート

\$: 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

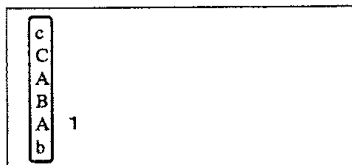
0 AbcABC	3 ABCAbc
1 bcABCA	0 AbcABC
2 cABCAB	4 BCAbcA
3 ABCAbc	5 CAbcAB
4 BCAbcA	1 bcABCA
5 CAbcAB	2 cABCAB

ソート

最終列と1番行の場所で全情報再現可能

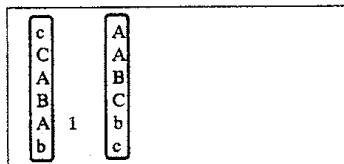
Burrows-Wheeler 変換 逆変換

- 入力



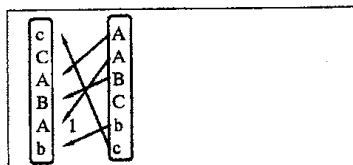
Burrows-Wheeler 変換 逆変換

- ソートによって最初の行を知る



Burrows-Wheeler 変換 逆変換

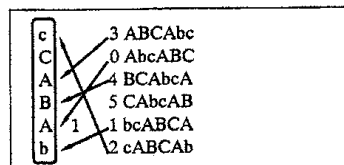
- 行番号特定



(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 dynamicall
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
```

繰り返しが見えやすい→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

```

{ aa.....
{ ab.....
{ ab.....
{ ab.....
{ ab.....
{ ae.....
{ ax.....
{ az.....
{ az.....
{ ba.....
  
```

copy method 原理

- 先頭文字が同一の「区間」内を順にソート

```

{ aa.....
{ ab.....
{ ab.....
{ ab.....
{ ab.....
{ ae.....
{ ax.....
{ az.....
{ az.....
{ ba.....
  
```

copy method 原理

- ソートされた区間内の末尾文字を調べる

```

{ aa.....h
{ ab.....f
{ ab.....e
{ ab.....u
{ ab.....f
{ ae.....g
{ ax.....
{ az.....
{ az.....
{ ba.....
  
```

copy method 原理

- 判明箇所を決定(コピー)

```

{ aa.....h
{ ab.....f
{ ab.....e
{ ab.....u
{ ab.....f
{ ae.....g
{ ax.....
{ az.....
{ az.....
{ ba.....
  
```

一番速いのは
「a」区間の最初

一番速いのは
「a」区間の二番目

copy method 原理

- 判明箇所を決定(コピー)

```

{ ex.....
{ ex.....
{ fab.....
{ fab.....
{ fa.....
{ fa.....
{ fa.....
{ fb.....
{ fb.....
{ fb.....
  
```

コピー先の区間では
上から決定される

