

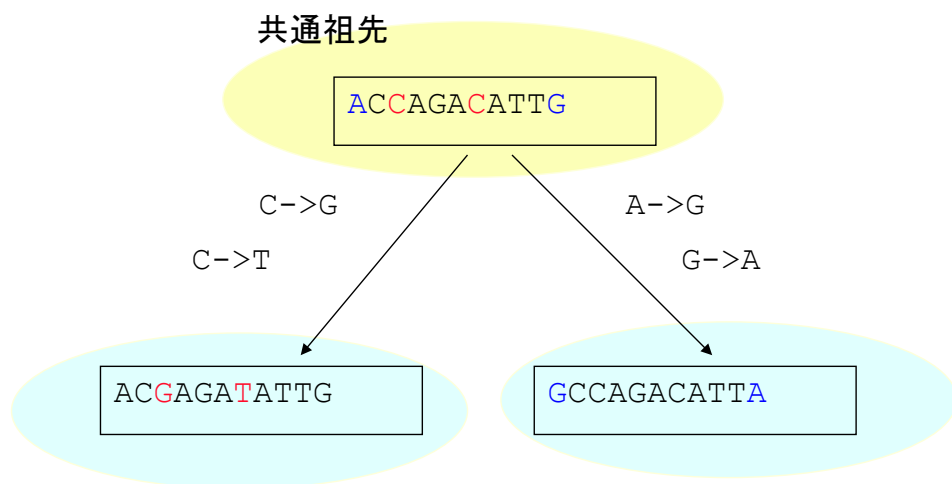
# BLAST Inside

## 配列検索の理論的背景

内山郁夫

## ホモロジー(相同性)

共通祖先の同一の構造から派生した構造であること



相同性(homology)は類似性(similarity)に基づいて推定する

十分高い類似性がある → 相同性がある

```
ACGAGATATTG
| | | | |
GCCAGACATTG
```

# アライメントによる配列比較

2本の配列を、類似性スコアが最大になるように、ギャップを適当に挿入して並べる

GCATGAGGA  
GTATGGATAAGA



スコアの定義 : 一致 **+2**、不一致 **-1**、  
ギャップ **-2**

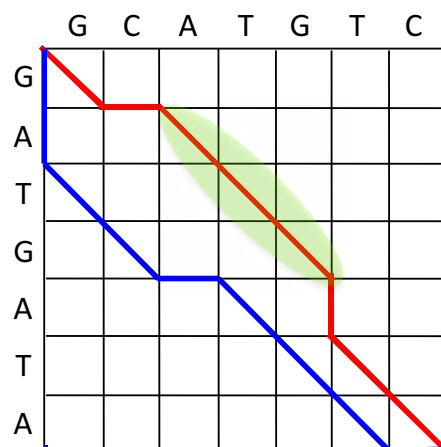
GCATG---AGGA

GTATGGATAAGA

**+2****-1****+2****+2****+2****-2****-2****-2****+2****-1****+2****+2** = +6

## 配列アライメントとマトリクス

あらゆるアライメントは、2つの配列を縦横に配置したマトリクス上のパスとして表される



アライメント1

GCATG-TC

G-ATGATA

アライメント2

--GCATGTC

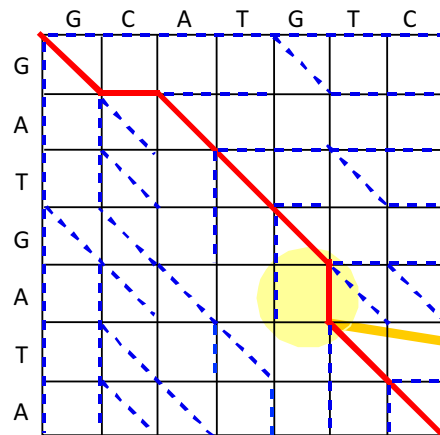
GATG-ATA-

斜め(対角線)方向 → 2つの配列をギャップを入れずに並べる

縦・横方向 → 一方の配列にギャップを挿入する

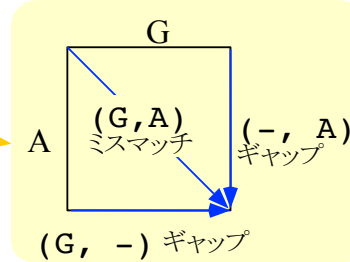
# 動的計画法による最適配列アライメント

## Needleman-Wunsch 法



最適アライメント

GCATG-TC  
G-ATGATA



$$S(i, j) = \max \begin{cases} S(i-1, j-1) + s(i, j) \\ S(i-1, j) + g \\ S(i, j-1) + g \end{cases}$$

2本の配列の長さを2辺とする長方形の升目を埋めるだけの計算量が必要

## グローバルアライメントとローカルアライメント

### ・ グローバル(大域)アライメント

- 配列全長どうしを並べる。
- 両者ともに全長にわたって相同性がある場合のみ意味を持つ。



### ・ ローカル(局所)アライメント

- 部分配列間のアライメントで類似度が最大となるものを求める。
- 配列の一部の領域(ドメイン)のみに相同性がある場合、もしくは相同性があっても類似性が低く、一部の領域のみしか類似性が認識できないような場合に効果的。
- 類似度が低い場合にスコアがマイナスになるようなスコア体系を用いる必要がある。

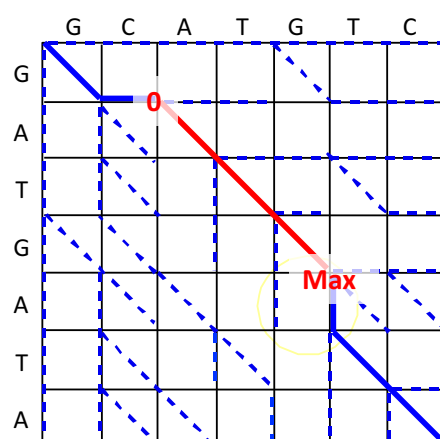


相同性検索において、通常は全長に渡る相同性があることは確証できないので、ローカルアライメントがよりよい選択肢となる。

# ローカルアライメントとマトリクス

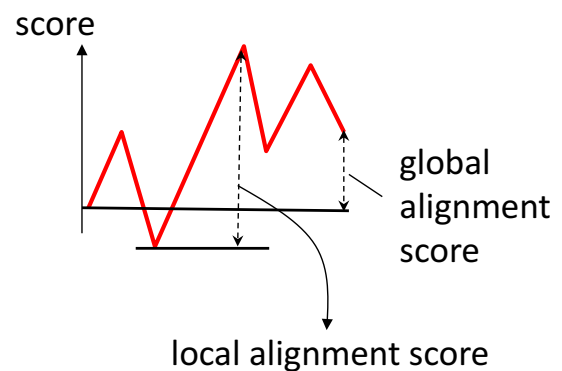


## 動的計画法による局所アライメント Smith-Waterman法



局所最適アライメント

GCATG-TC  
G-ATGATA



$$S(i, j) = \max \begin{pmatrix} S(i-1, j-1) + s(i, j) \\ S(i-1, j) + g \\ S(i, j-1) + g \\ 0 \end{pmatrix}$$

局所アライメントのスコアはつねに0以上になる

# 高速相同性検索

- データベース中からクエリ配列と十分に高い類似性(=相同性)を持つ配列領域を検索する
  - そのような配列領域をいかにして高速に検索するか(アルゴリズムの問題)
  - 「十分に高い類似性」をどのようにして定義するか(類似性の定義と統計的検定の問題)

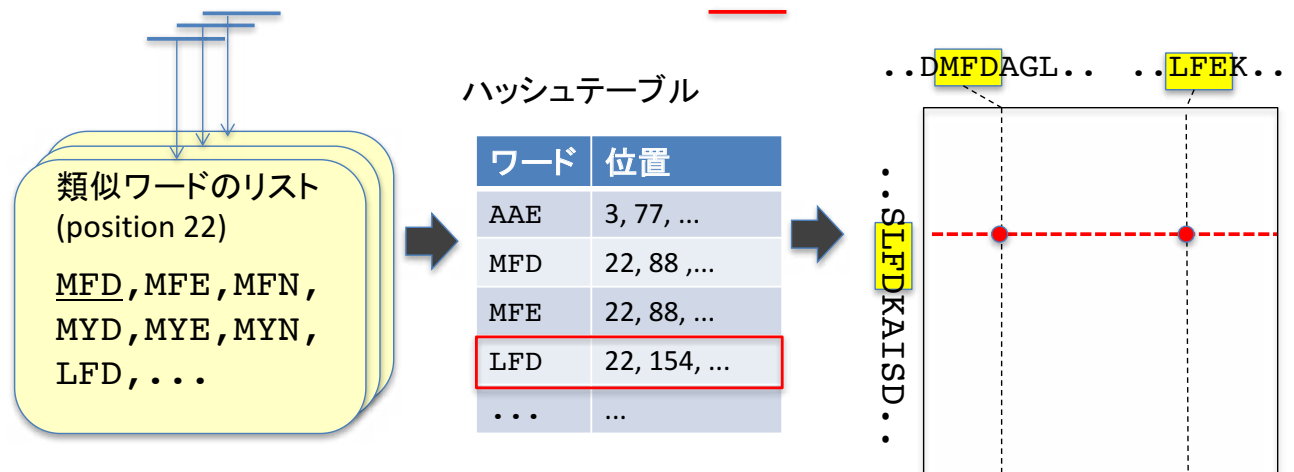
## BLAST のアルゴリズム 初期検索

問い合わせ(Query)配列

...MFDAGLNDGE...

データベース(Subject)配列

...SLFDKAISDGD...



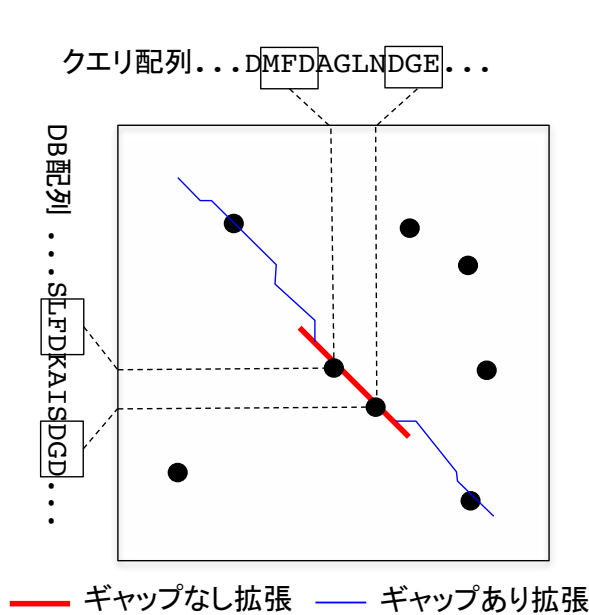
関連するオプション

-word\_size 初期検索で使うワードのサイズ

-threshold 類似ワードに加える際の類似性スコアの閾値

# BLAST のアルゴリズム

## 類似領域の拡張(アライメント)

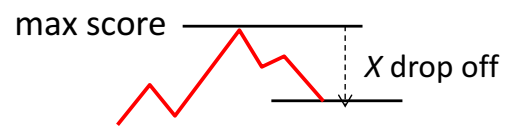


1) 一定のウィンドウ内に2回ヒットが見つかる領域をとる

...DMFDAGLNDGE...  
...SLFDKAISDGD...

2) ギャップなしで拡張して、閾値以上のスコアをとる領域について、ギャップありでさらにアライメントを拡張

3) アライメントの拡張は、最高スコアから  $X$  低くなったら打ち切る

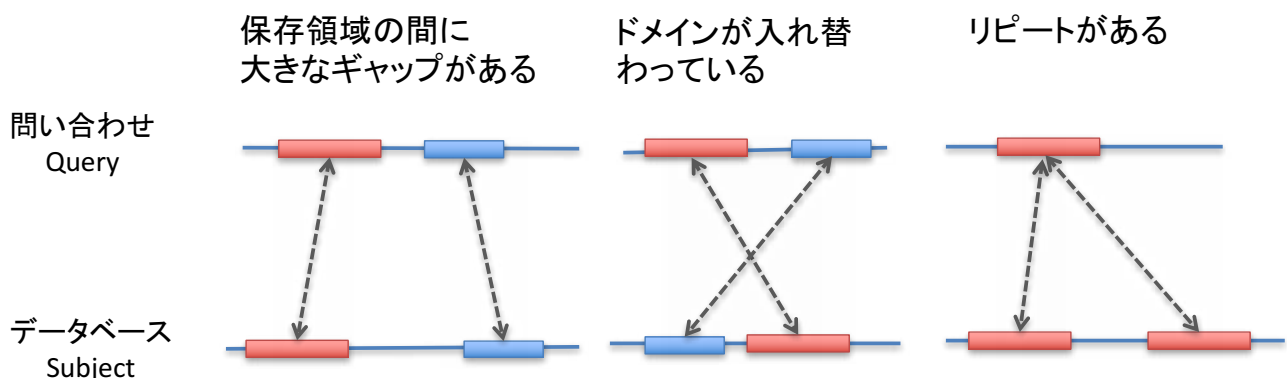


4) 統計的に有意なスコアを持つヒットを残す

関連するオプション

- window\_size 2つのヒットを見つけるウィンドウのサイズを設定
- xdrop\_ungap/gap Xdrop パラメータの設定

## 一つのエントリに対して 複数のアライメント(HSP)が含まれる場合



デフォルトでは最高スコアのHSPを全体のスコアとして評価する。  
consistentなアライメントについてスコアの和をとって評価するオプションもある

関連するオプション

- sum\_stats consistentなHSPが複数あるとき、スコアの和でE-valueを計算する

# スコアの定義

## ギャップペナルティ

スコアの定義(1) : 一致+2、不一致-1、ギャップ-2

GCATG-A-G-GA

GTATGGATAAGA

+2-1+2+2+2-2+2-2-1-2+2+2 = +6

スコアの定義(2) : 一致+2、不一致-1、  
ギャップ開始-2、ギャップ延長-1

GCATG---AGGA

GTATGGATAAGA

+2-1+2+2+2-2-1-1+2-1+2+2 = +8

注) BLASTでは、最初のギャップペナルティを  
ギャップ開始+ギャップ延長  
となるように定義している。従って、  
この例では開始、延長ともペナルティは-1となる。

### 関連するオプション

- reward 一致のスコア (≥0; DNAのみ)
- penalty 不一致のペナルティ (≤0; DNAのみ)
- gapopen ギャップ開始に対するペナルティ(≥0)
- gapextend ギャップ延長に対するペナルティ(≥0)

# スコアの定義

## アミノ酸置換行列(PAM250)

P	6																	形や大きさ、性質が似たアミノ酸間には高いスコア																																		
A	1	2	small																		似ていないアミノ酸間には低いスコア																															
G	0	1	5																																																	
N	0	0	0	2	amide																																															
Q	0	0	-1	1	4																																															
D	-1	0	1	2	2	4	acidic																																													
E	-1	0	0	1	2	3	4																																													
T	0	1	0	0	-1	0	0	3	hydroxyl																																											
S	1	1	1	1	-1	0	0	1	2																																											
C	-3	-2	-3	-4	-5	-5	-5	-2	0	12																																										
V	-1	0	-1	-2	-2	-2	-2	0	-1	-2	4																																									
I	-2	-1	-3	-2	-2	-2	-2	0	-1	-2	4	5	aliphatic (V,I,L)																																							
M	-2	-1	-3	-2	-1	-3	-2	-1	-2	-5	2	2	6																																							
L	-3	-2	-4	-3	-2	-4	-3	-2	-3	-6	2	2	4	6																																						
K	-1	-1	-2	1	1	0	0	0	0	-5	-2	-2	0	-3	5	basic																																				
R	0	-2	-3	0	1	-1	-1	-1	0	-4	-2	-2	0	-3	3	6																																				
H	0	-1	-2	2	3	1	1	-1	-1	-3	-2	-2	-2	-2	0	2	6																																			
F	-5	-3	-5	-3	-5	-6	-5	-3	-3	-4	-1	1	0	2	-5	-4	-2	9	aromatic																																	
Y	-5	-3	-5	-2	-4	-4	-4	-3	-3	0	-2	-1	-2	-1	-4	-4	0	7	10																																	
W	-6	-6	-7	-4	-5	-7	-7	-5	-2	-8	-6	-5	-4	-2	-3	2	-3	0	0	17																																
	P	A	G	N	Q	D	E	T	S	C	V	I	M	L	K	R	H	F	Y	W																																

### 関連するオプション

- matrix スコア行列を指定(タンパク質のみ)

## アミノ酸置換行列(BLOSUM62)

P	A	G	N	Q	D	E	T	S	C	V	I	M	L	K	R	H	F	Y	W
-1	-2	-2	-1	-1	-2	-1	0	-1	-3	-2	-3	-2	-3	-2	-2	-4	-4	-3	-2
4	0	0	0	0	0	2	-1	0	-3	-3	-3	-1	-3	5	0	0	3	2	1
small			amide		acidic		hydroxyl			aliphatic (V,I,L)				basic			aromatic		
6	5	6	5	4	9	4	5	8	6	7	11								

**-matrix** スコア行列を指定(タンパク質のみ)

## 置換頻度の統計に基づく アミノ酸置換行列の定義

相同配列間でアミノ酸a,bが置換する確率

$$S(a,b) = \alpha \log \frac{q(a,b)}{p(a)p(b)}$$

非相同配列間でアミノ酸  $i, j$  が偶然揃う確率

配列をランダムに  
並べ替えたもの

実際のアライメント中でaとbが並ぶ頻度

PDSTIQMINRYLAKHPQTNRFRIILVCGGDG  
| | | | |  
PIKAVQLCTLL...PYYS..ARVLVCGGDG

TIDKANLPVLPPITAVLPLGTGNDLARCLRWG  
| | | | |  
CIDKANFTKHPPVAVLPLGTGNDLARCLRWG

配列間の近さによって置換頻度は異なる

## アライメントをどう計算するか？ (スコア行列を使わずに)

対数オッズスコア

$S > 0$  相同である  
 $S < 0$  相同でない

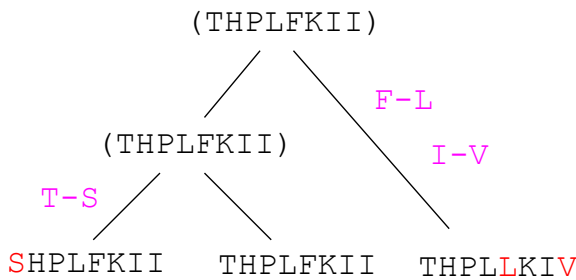


# PAM行列 (Dayhoff 1978)

PAM (accepted point mutation)

100残基あたりで受容された点突然変異の回数。  
配列間の進化的な距離(時間)の単位。

1) 近縁配列間の最節約系統樹に基づいて、置換頻度をカウント



2) 置換頻度 ( $f(a,b)$ ) を規格化して 1PAMあたりの置換確率行列  $M$  を計算

$$M(a,b) = \frac{f(a,b)}{\sum_{x \neq a} f(a,x)} m(a)$$

$M(a,b)$ : あるアミノ酸  $a$  が、1PAMの期間内に  $b$  に置換する確率

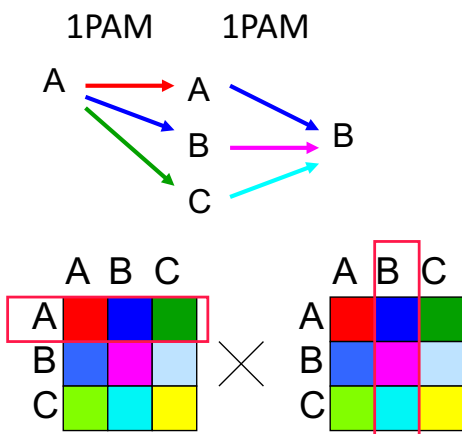
$m(a)$ : アミノ酸  $a$  の相対的な置換しやすさ

# PAM行列 (Dayhoff 1978)

PAM (accepted point mutation)

受容された点突然変異の100残基あたりの回数。  
配列間の進化的な距離(時間)の単位。

3) 置換確率行列  $M$  を  $n$  回掛け合わせることで、 $n$ PAMの置換確率行列  $M_n$  を計算



4) 対数オッズをとって、 $n$ PAMのスコア行列を計算

$$S_n(a,b) = \log \frac{M_n(a,b)}{p(b)}$$

$$= \log \frac{q_n(a,b)}{p(a)p(b)}$$

# BLOSUM行列

## Blocksデータベース

FA12_HUMAN ( 217)	CYDGRGLSYRGLARTTSLGAPCQPWAS
HGFA_HUMAN ( 286)	CFLGNGTCYRGVASTSASGLSCLAWNS
UROK_CHICK ( 79)	CYSGNGEDYRGMAEDPGCLYWDHPSVI
UROT_HUMAN ( 215)	CYFGNGSAYRGTHSLTESGASCLPWNS
APOA_HUMAN ( 28)	CYHGDGQSYRGTYSTTVTGRTCAWSS
PLMN_BOVIN ( 358)	CYHGNGQSYRGTSSTTITGRKCQSWSS
PLMN_HUMAN ( 377)	CYHGDGQSYRGTSSTTTTGKKCQSWSS
PLMN_MACMU ( 377)	CYHGDGQSYRGTSSTTTTGKKCQSWSS
PLMN_MOUSE ( 377)	CYQSDGQSYRGTSSTTITGKKCQSWAA
PLMN_PIG ( 358)	CYRGNGESYRGTSSTTITGRKCQSWVS
HGFL_HUMAN ( 283)	CYRGKGECYRGTTANTTTAGVPCQRWDA
HGFL_MOUSE ( 292)	CYRGKGEDYRGTTNTTSAGVPCQRWDA

$n\%$ 以上一致する  
配列をグループ化  
(クラスタリング)

グループの和が1とな  
るよう配列を重みづけし  
て頻度をカウント

Y:4, F:2

S:2, G:1.5, D:1.5, A:1

Y  $\leftrightarrow$  F :  $4 \times 2 = 8$   
F  $\leftrightarrow$  F :  $2 / 2 = 1$   
Y  $\leftrightarrow$  Y :  $4 \times 3 / 2 = 6$

カラムごとにアミノ酸置換数を数え  
上げて、置換頻度 $q(a,b)$ を計算

対数オッズ

$$M(a,b) = \log \frac{q(a,b)}{p(a)p(b)}$$

## BLOSUM $n$ 行列

	A	R	N	D	C
A	7	-3	-3	-3	-1
R	-3	9	-1	-3	-6
N	-3	-1	9	2	-5
D	-3	-3	2	10	-7
C	-1	-6	-5	-7	13

# スコア行列の使い分け

## PAM (Dayhoff et al. 1978)

近縁蛋白質配列間のマ  
ニュアルアライメント  
をもとに作成

120  
160  
210  
250

強くて短い類似領域の検出

## BLOSUM (Henikoff et al. 1992)

自動生成された、保存  
部位周辺のマルチプル  
アライメントから作成

80  
62  
50  
45

弱くて長い類似領域の検出

BLAST 標準

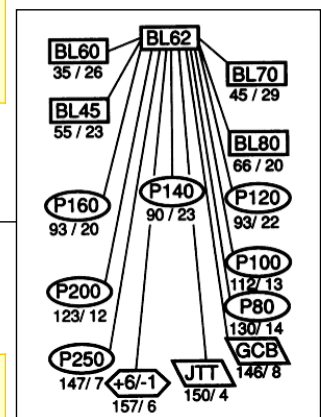
FASTA 標準

## BLOSUM 80

	A	R	N	D	C
A	7	-3	-3	-3	-1
R	-3	9	-1	-3	-6
N	-3	-1	9	2	-5
D	-3	-3	2	10	-7
C	-1	-6	-5	-7	13

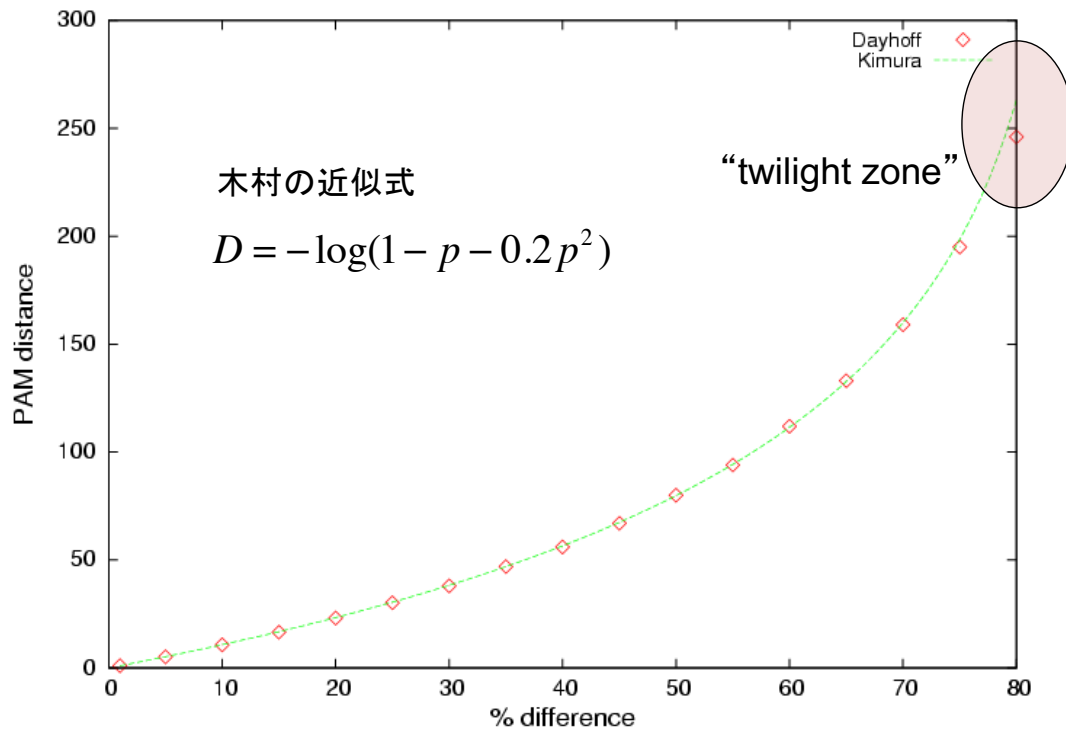
## BLOSUM 45

	A	R	N	D	C
A	5	-2	-1	-2	-1
R	-2	7	0	-1	-3
N	-1	0	6	2	-2
D	-2	-1	2	7	-3
C	-1	-3	-2	-3	12



BLASTパフォーマンス  
試験による比較  
(Henikoff et al. 1992)

## 観察される配列間の違い(%difference)と 実際に起きた変異数(PAM)との関係



## 統計的評価(E-value)

- E-value: 同じ大きさのランダム配列データベースを検索したとき、スコアS以上のヒットが偶然に得られる個数の期待値

$$E = Kmne^{-\lambda S}$$

$m$ : データベース全体の長さ

$n$ : 問い合わせ配列の長さ

$K, \lambda$ : スコア行列とアミノ酸組成に依存するパラメータ  
(BLAST内部で計算される)

### 関連するオプション

-evalue E-valueの閾値を設定

-dbsize データベースサイズを設定

-searchsp 検索空間(データベースサイズ×クエリ配列サイズ)を設定

# 統計的評価(p-value)

- p-value: 同じ大きさのランダム配列データベースを検索したとき、スコア $S$ 以上のヒットが(少なくとも一つ)見つかる確率

$$p = 1 - e^{-E} = 1 - e^{-Kmn e^{-\lambda S}}$$

期待値 $E$ のポアソン分布の式  $P(k) = \frac{E^k e^{-E}}{k!}$  から、 $1-P(0)$ として求められる

$E \rightarrow 0$ のとき、 $p = 1 - e^{-E} \rightarrow E$ となるので、  
 $E$ が小さければp-value は E-valueと同じと考えてよい

## 標準化されたスコア (bit-score)

$$\text{bit-score: } S' = \frac{\lambda S - \log K}{\log 2}$$

このとき、E-valueは  $E = mn2^{-S'}$  で計算できる。

ビットスコアは、スコア行列に固有のパラメータ $\lambda$ ,  $K$ に依存せず、  
統計的評価と直接結びついている

例) ビットスコア  $S'=30$ , データベース長  $m=5,000,000$ , クエリ長  $n=200$ のとき、  
 $E = 5 \times 10^6 \times 200 \times 2^{-30} = 0.93$

# 統計的検定についての注意

- E-valueが低い→帰無仮説を否定
- ここでの帰無仮説は、「得られたスコアが、同じアミノ酸組成を持つランダムなアミノ酸配列から得られるスコアと変わらないこと」
- 大抵はE-valueが低いことから2つの配列が相同であると結論づけられるが、そこには若干の飛躍があり、必ずしもそうはいえないこともある。

## 低複雑性領域のフィルタリング

SEG - 低複雑性(=アミノ酸組成の偏った)領域を除く

### クエリ配列

```
>SOS_DROME son-of-sevenless
MFSGPSGHAHTISYGGGIGLGTGGGGGSGG
SGSGSQGGGGGIGIGGGVAGLQDCDGYDF
TKCENAARWRGLFTPSLKKVLEQVHPRVTA
KEDALLYVEKLCRLLLAMLCAPLPHSVQD
```

### 検索結果

SOS_DROME	SON OF SEVENLESS PROTEIN.	0.0
GNRP_RAT	GUANINE NUCLEOTIDE RELEAS	7.5e-43
GNRP_MOUSE	GUANINE NUCLEOTIDE RELEAS	5.7e-42
CC25_SACKL	CELL DIVISION CONTROL PRO	3.4e-32
CC25_YEAST	CELL DIVISION CONTROL PRO	3.9e-22
CC25_CANAL	CELL DIVISION CONTROL PRO	4.0e-21
STE6_SCHPO	STE6 PROTEIN.	2.0e-17
SC25_YEAST	SCD25 PROTEIN.	3.1e-16
GNDS_MOUSE	GUANINE NUCLEOTIDE DISSOC	5.8e-14
GNDS_RAT	GUANINE NUCLEOTIDE DISSOC	1.2e-13
BRN1_HUMAN	BRAIN-SPECIFIC HOMEBOX/P	1.1e-10
DISC_DROME	DISCONNECTED PROTEIN.	3.0e-10

生物学的に意味のない(相同でない)ヒット

```
>BRN1_HUMAN BRAIN-SPECIFIC HOMEBOX/POU DOMAIN PRO

Query: 9 AHTISYGGGIGLGTGGGGGSGSGSQGGGGGIGIGGGV 49
      A +I +      G G GGGG G G G+ GGGG+  G  V
Sbjct:18 AGSIVHSDAAGAGGGGGGGGGGGGAGGGGGGMPGSAAV 58
```

### フィルタリング後のクエリ配列

```
>SOS_DROME son-of-sevenless
MFSGPSGHAHTISYXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXVAGLQDCDGYDF
TKCENAARWRGLFTPSLKKVLEQVHPRVTA
KEDALLYVEKLCRLLLAMLCAPLPHSVQD
```

### 検索結果

SOS_DROME	SON OF SEVENLESS PROTEIN.	0.0
GNRP_RAT	GUANINE NUCLEOTIDE RELEAS	3.3e-37
GNRP_MOUSE	GUANINE NUCLEOTIDE RELEAS	2.5e-36
CC25_SACKL	CELL DIVISION CONTROL PRO	6.0e-26
CC25_CANAL	CELL DIVISION CONTROL PRO	2.1e-16
CC25_YEAST	CELL DIVISION CONTROL PRO	1.1e-15
SC25_YEAST	SCD25 PROTEIN.	8.0e-11
GNDS_MOUSE	GUANINE NUCLEOTIDE DISSOC	2.5e-10
STE6_SCHPO	STE6 PROTEIN.	3.2e-10
GNDS_RAT	GUANINE NUCLEOTIDE DISSOC	5.3e-09
H2A1_MOUSE	HISTONE H2A.1.	5.3e-05
H2A2_HUMAN	HISTONE H2A.2.	5.3e-05

生物学的に意味のあるヒットが上位に出現する

# BLASTで利用可能な フィルタリングプログラム

- **-seg**: for protein; default OFF
  - アミノ酸組成の偏った(複雑度が小さい)領域をマスクする。(プログラムsegmasker)
- **-dust**: for DNA; default ON
  - 3塩基ワードの出現が偏った領域をマスクする。(プログラムdustmasker)
- **-window\_masker\_db <file>**: for DNA; optional
  - ゲノム内で多数出現するワードをマスクする。ワード数を数える前処理が必要。(プログラムwindowmasker)
- **-filtering\_db <db>**: for DNA; optional
  - 指定したデータベース(例えばRepBase)とヒットする領域をマスクする。
- **-lcasemasking**: for protein and DNA; optional
  - 小文字の部分をマスクする(クエリ、データベースとも)。

## ハードマスクとソフトマスク

### • ハードマスク

- クエリ配列中でマスクする領域を、"X"(DNAの場合"N")で置き換えることにより、完全に検索対象から外す。

(問題点)クエリ配列をハードマスクしてしまうと、配列間の正確なアライメントや類似性スコアが計算できなくなる。

### • ソフトマスク

- クエリ配列中でマスクする領域は、初期検索における類似ワードリストの作成の対象からは外すが、その後のアライメント拡張フェイズでは通常の配列と同様に扱う。  
→アライメントや類似性スコアはマスクせずに計算される。

関連するオプション

-soft\_masking フィルタをソフトマスクとして使用

# Composition-based statistics

## アミノ酸スコア行列

$$S(a,b) = \log \frac{q(a,b)}{p(a)p(b)}$$

相同配列間で観察される置換の頻度

平均的なアミノ酸の出現頻度

配列のアミノ酸組成が平均の組成と比べて大きく偏っている場合、このスコア行列は最適ではない

## アミノ酸組成に基づくスコアの補正

$$S'(a,b) = \log \frac{q'(a,b)}{p'(a)p''(b)}$$

$q'(a,b)$ :  $p', p''$  に合わせて補正した置換頻度

$p'(a)$ : クエリ配列のアミノ酸頻度

$p''(b)$ : データベース配列のアミノ酸頻度

関連するオプション

`-comp_based_stats` 組成に基づくスコア統計を使用

## オプションのまとめ

- 検索の速度を上げる(感度は下がる)
  - `word_size`を大きく、`threshold`を大きくする。
  - `window_size`を小さく、`xdrop_{ungap/gap}`を小さくする。
- 高い類似性での一致にフォーカスする
  - タンパク質の場合、`matrix`としてBLOSUMの大きいものかPAMの小さいものを指定する。
  - DNAの場合、`reward`に対して`penalty`の絶対値を大きくする。
- 出力を類似性スコアが高いものに絞る(速度も若干向上)
  - `evaluate`(閾値)を小さくする。
  - `max_target_seqs`(最大出力数)を小さくする(`outfmt>=5`の場合)。
- 繰り返し配列のフィルタリング
  - 組成が偏った領域を除くには、タンパク質では`seg`、DNAでは`dust`を使う。タンパク質では`comp_base_stats`によっても緩和される。
  - ゲノム中に散在する反復配列を除くには、`window_masker`か`filtering_db`を使う。

# デフォルトのオプション設定 (blastp/blastx/tblastn)

program/task	blastp	blastp-short	blastx	tblastn
word_size	3	2	3	3
threshold	11	16	12	13
window_size	40	15	40	40
gapopen	11	9	11	11
gapextend	1	1	1	1
matrix	BLOSUM62	PAM30	BLOSUM62	BLOSUM62
seg	no	no	yes	yes
soft_masking	false	false	false	false
comp_based_stats	2	0 (false)	2	2
purpose	general	to find short & strong match	translate nucl query	translate nucl DB

# デフォルトのオプション設定(blastn)

task	blastn	blastn-short	megablast	dc-megablast
word_size	11	7	28	11
window_size	0	0	0	40
gapopen	5	5	0	5
gapextend	2	2	2.5	2
reward (match)	2	1	1	2
penalty (mismatch)	-3	-3	-2	-3
dust	yes	yes	yes	yes
soft_masking	true	true	true	true
purpose	general	to find short & strong match	to find strong match for large query	to find moderate match for large query



# Discontiguous seed (dc-megablast)

Contiguous seed: 11111111

Discontiguous seed: 11011010111

0=don't care

CAGTGTAGGACGTGATCAC

GAGTGTACGACGTGATCAG

contiguous

11111111

11111111

11111111

discontiguous

11011010111

11011010111

ヒットする個数の期待値は1の個数が同じならほぼ同じ

Discontiguous seedの方が置換を許したヒットを効率よく見つけられる