基生研ゲノムインフォマティクストレーニングコース

BLAST 自由自在

配列解析の極意をマスターする

December 1, 2016

基礎生物学研究所　生物機能解析センター

BLAST 基礎

Shuji Shigenobu /重信秀治

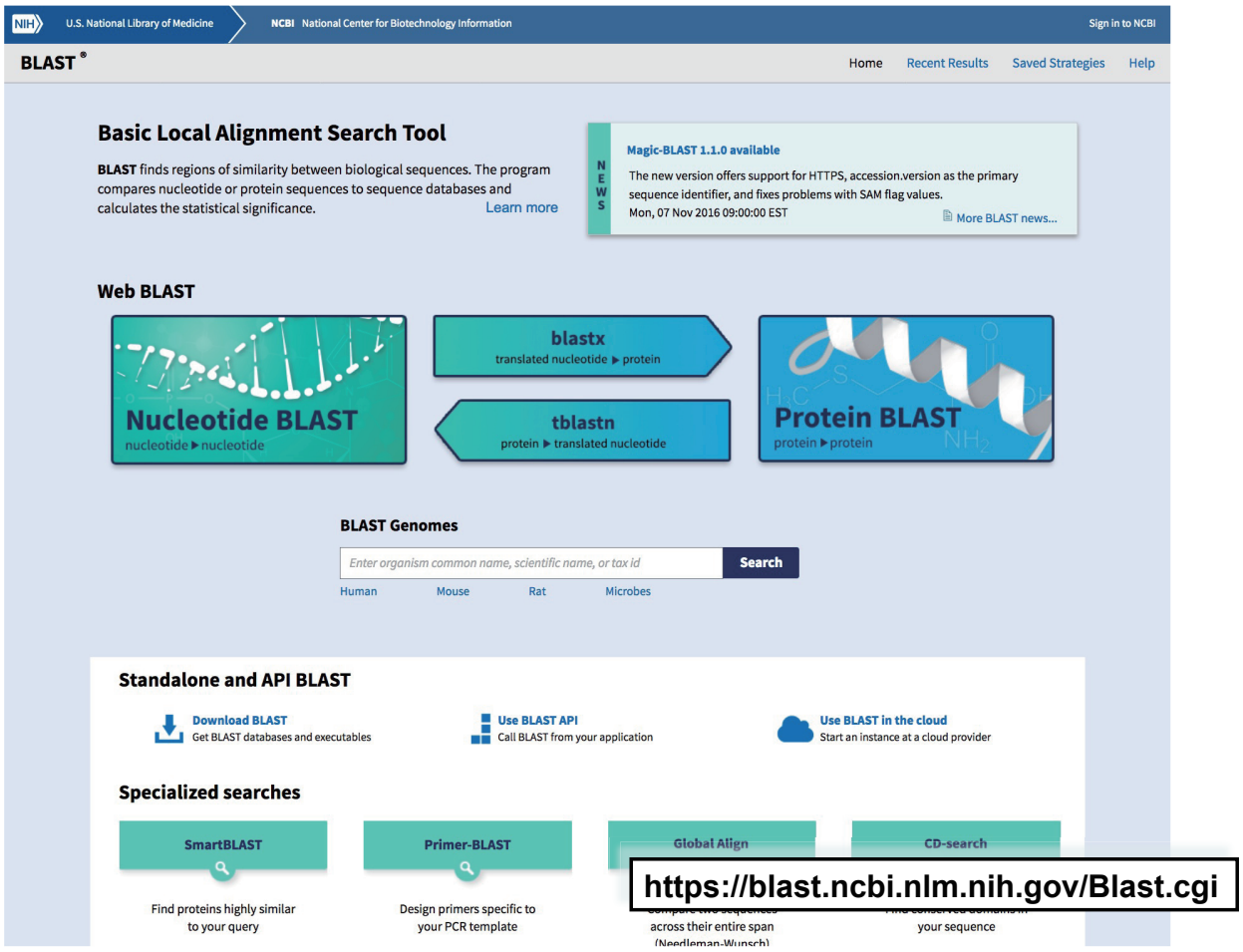
Aim
- BLASTの基本を学ぶ
- CUI環境でBLASTを実行する技術を習得する
- BLASTパッケージに含まれる便利な諸ツールを習得する(blastdbcmdなど)

What's **BLAST**

Basic Local Alignment Search Tool

- BLAST is the tool most frequently used for calculating sequence similarity
- Finds best local alignments
- Heuristic approach based on Smith Waterman algorithm
- Provides statistical significance
- BLAST can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families

http://blast.ncbi.nlm.nih.gov/Blast.cgi



Standard BLAST Programs

Query	DB	Program
DNA *vs*	DNA	**BLASTN**
Protein *vs*	Protein	**BLASTP**
DNA translation *vs*	Protein	**BLASTX**
Protein *vs*	DNA translation	**TBLASTN**
DNA translation *vs*	DNA translation	**TBLASTX**

Quick start

ウサギのあるcDNA断片データを得た。これが何の遺伝子をコードしているのかを、マウスタンパク質データベースに対してsimilarity searchする事により明らかにする。

- Query: XM_002716705.3.fasta
- DB: mouse_proteins.pep.fasta

1. Format DB

```
$makeblastdb -in mouse_proteins.pep.fasta -dbtype prot -parse_seqids
```

2. Search

```
$blastx -db mouse_proteins.pep.fasta –query XM_002716705.3.fasta
```

Anatomy of BLAST output (format 0)

```
out.bl0.txt — Edited
BLASTX 2.5.0+


Reference: Stephen F. Altschul, Thomas L. Madden, Alejandro A.
Schaffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J.
Lipman (1997), "Gapped BLAST and PSI-BLAST: a new generation of
protein database search programs", Nucleic Acids Res. 25:3389-3402.


Database: mouse_proteins.pep.fasta
           49,870 sequences; 21,467,191 total letters


Query= XM_002716705.3 PREDICTED: Oryctolagus cuniculus autophagy related 3
(ATG3), mRNA

Length=1516
                                                                      Score     E
Sequences producing significant alignments:                          (Bits)  Value

Q9CPX6  Ubiquitin-like-conjugating enzyme ATG3 OS=Mus musculus GN...  598     0.0
Q8R1P4  Ubiquitin-like-conjugating enzyme ATG10 OS=Mus musculus G...  35.0    0.13
A0A0R4J029  Autophagy-related 10 (Yeast), isoform CRA_a OS=Mus mu...  34.7    0.14
A0A0X1KG62  Negative elongation factor B OS=Mus musculus GN=Nelfb...  30.4    5.0
Q0VF22  Coiled-coil domain-containing protein 138 OS=Mus musculus...  30.0    6.9
Q8C4Y3  Negative elongation factor B OS=Mus musculus GN=Nelfb PE=...  30.0    7.1
Q9JK38  Glucosamine 6-phosphate N-acetyltransferase OS=Mus muscul...  29.3    8.3
Q8C5K5  Uncharacterized protein CXorf38 homolog OS=Mus musculus P...  29.6    8.8


>Q9CPX6 Ubiquitin-like-conjugating enzyme ATG3 OS=Mus musculus GN=Atg3
PE=1 SV=1
Length=314

 Score = 598 bits (1542),  Expect = 0.0, Method: Compositional matrix adjust.
 Identities = 309/314 (98%), Positives = 313/314 (99%), Gaps = 0/314 (0%)
 Frame = +1

Query  418   MQNVINTVKGKALEVAEYLTPVLKESKFKETGVITPEEFVAAGDHLVHHCPTWQWATGEE  597
             MQNVINTVKGKALEVAEYLTPVLKESKFKETGVITPEEFVAAGDHLVHHCPTWQWATGEE
Sbjct  1     MQNVINTVKGKALEVAEYLTPVLKESKFKETGVITPEEFVAAGDHLVHHCPTWQWATGEE  60

Query  598   LKVKAYLPTGKQFLVTKNVPCYKRCKQMEYSDELEAIIEEDDGDGGWVDTYHNTGITGIT  777
             LKVKAYLPT KQFLVTKNVPCYKRCKQMEYSDELEAIIEEDDGDGGWVDTYHNTGITGIT
Sbjct  61    LKVKAYLPTDKQFLVTKNVPCYKRCKQMEYSDELEAIIEEDDGDGGWVDTYHNTGITGIT  120

Query  778   EAVKEITLENKDSIKLQDCSALCeeeeeedegeaadmeeyeeSGLLETDEATLDTRKLVE  957
             EAVKEITLE+KDSIKLQDCSALC+EE+EEDEGEAADMEEYEESGLLETDEATLDTRK+VE
Sbjct  121   EAVKEITLESKDSIKLQDCSALCDEEDEEDEGEAADMEEYEESGLLETDEATLDTRKIVE  180

Query  958   ACKAKADAGGEDAILQTRTYDLYITYDKYYQTPRLWLFGYDEQRQPLTVEHMYEDISQDH  1137
             ACKAKADAGGEDAILQTRTYDLYITYDKYYQTPRLWLFGYDEQRQPLTVEHMYEDISQDH
Sbjct  181   ACKAKADAGGEDAILQTRTYDLYITYDKYYQTPRLWLFGYDEQRQPLTVEHMYEDISQDH  240

Query  1138  VKKTVTIENHPHLPPPPMCSVHPCRHAEVMKKIIETVAEGGGELGVHMYLLIFLKFVQAV  1317
             VKKTVTIENHPHLPPPPMCSVHPCRHAEVMKKIIETVAEGGGELGVHMYLLIFLKFVQAV
Sbjct  241   VKKTVTIENHPHLPPPPMCSVHPCRHAEVMKKIIETVAEGGGELGVHMYLLIFLKFVQAV  300

Query  1318  IPTIEYDYTRHFTM  1359
             IPTIEYDYTRHFTM
Sbjct  301   IPTIEYDYTRHFTM  314


>Q8R1P4 Ubiquitin-like-conjugating enzyme ATG10 OS=Mus musculus GN=Atg10
PE=1 SV=1
Length=215

 Score = 35.0 bits (79),  Expect = 0.13, Method: Compositional matrix adjust.
 Identities = 25/116 (22%), Positives = 52/116 (45%), Gaps = 11/116 (9%)
 Frame = +1

Query  994   AILQTRTYDLYITYDKYYQTPRLWLFGYDEQRQPLTVEHMYEDISQDHVKK-------TV  1152
             A+ +   ++ ++ Y   YQ P L+         +PL +E ++E + + +  +       T+
Sbjct  88    AVAEVIKHEYHVLYSCSYOVPVLYFRASFLDGRPLALEDIWEGVHECYKPRLLOGPWDTI  147
```

header

one-line summaries

alignments

Anatomy of BLAST output (format 0)

```
out.bl0.txt — Edited
Sbjct  184  KKQGLL-PHEINQIYDELYHIHMKLQYETTAQKKFAEELQKREQFLAER-------EQL  234

Query  682  EYSDE-----LEAIIEEDDGDGGWVDTYHNTGITGITEAVKEITLENK  810
             +S E     ++ + EE        +   H T I  +TEA+KE   ENK
Sbjct  235  LFSHETALSKIKGVKEEVLTRFQILKEQHGTEIEHLTEALKEKNKENK  282


>Q8C4Y3 Negative elongation factor B OS=Mus musculus GN=Nelfb PE=1 SV=2
Length=580

 Score = 30.0 bits (66),  Expect = 7.1, Method: Compositional matrix adjust.
 Identities = 10/25 (40%), Positives = 16/25 (64%), Gaps = 1/25 (4%)
 Frame = -2

Query  549  ISSCHKLFWCYDTCFLKLRFLEDRR  475
            +  CHK  WC D C ++ RF++ +R
Sbjct  256  VDPCHKFTWCLDAC-IRERFVDSKR  279


>Q9JK38 Glucosamine 6-phosphate N-acetyltransferase OS=Mus musculus GN=Gnpnat1
PE=1 SV=1
Length=184

 Score = 29.3 bits (64),  Expect = 8.3, Method: Compositional matrix adjust.
 Identities = 20/73 (27%), Positives = 31/73 (42%), Gaps = 13/73 (18%)
 Frame = +1

Query  337  WGRPTAQLLPGLPSPRRRGAGRCSGPRMQNVINTVKGKALEVAEYLTPVLKESKFKETGV  516
            W + TA   P + SP   G G   P     +N    K L             +  ETGV
Sbjct  19   WSQNTAIFSPAI-SPTHPGEGLVLRPLCTADLNKGFFKVL------------GQLTETGV  65

Query  517  ITPEEFVAAGDHL  555
            ++PE+F+ + +H+
Sbjct  66   VSPEQFMKSFEHM  78


>Q8C5K5 Uncharacterized protein CXorf38 homolog OS=Mus musculus PE=1
SV=1
Length=320

 Score = 29.6 bits (65),  Expect = 8.8, Method: Compositional matrix adjust.
 Identities = 22/83 (27%), Positives = 37/83 (45%), Gaps = 2/83 (2%)
 Frame = +1

Query  580  WATGEELKVKAYLPTGKQFLVTKNVPCYKRCKQMEYSDELEAIIEEDDGDGGWVDTYHNT  759
            W    ++K++ +L   K   + + V  Y R +Q+  SD    I EED+ DG   +
Sbjct  177  WLRDFQIKIQNFLNEFKN--IPEIVAVYSRIEQLLTSDWAVHIPEEDERDGCEFEIGSYL  234

Query  760  GITGITEAVKEITLENKDSIKLQ  828
             ++ I E   E+  E    + LQ
Sbjct  235  SVSQIHEIEIELLKEKLQEMYLQ  257


Lambda     K      H      a      alpha
   0.318   0.134  0.401  0.792  4.96

Gapped
Lambda     K      H      a      alpha   sigma
   0.267   0.0410 0.140  1.90   42.6    43.6

Effective search space used: 6420178198


  Database: mouse_proteins.pep.fasta
    Posted date:  Nov 18, 2016  4:07 PM
  Number of letters in database: 21,467,191
  Number of sequences in database:  49,870


Matrix: BLOSUM62
Gap Penalties: Existence: 11, Extension: 1
Neighboring words threshold: 12
Window for multiple hits: 40
```

alignments

Hierarchical structure of alignment part

- alignment #1
  - HSP #1
  - HSP #2
  - …
- alignment #2
- alignment #3
- …

footer

Let's try

- help を表示させる
- -evalue オプションを使ってみる。evalue cutoff = 1.0e-8 を指定してみよう
- 他のフォーマットで出力してみよう
  - format 6: table
  - format 7: table with comment lines
  - format 5: XML

Anatomy of BLAST output (format 7)

out.bl7.txt — Edited

```
# BLASTX 2.5.0+
# Query: XM_002716705.3 PREDICTED: Oryctolagus cuniculus autophagy related 3 (ATG3), mRNA
# Database: mouse_proteins.pep.fasta
# Fields: query acc., subject acc., % identity, alignment length, mismatches, gap opens, q. start,
q. end, s. start, s. end, evalue, bit score
# 8 hits found
XM_002716705.3	Q9CPX6	98.408	314	5	0	418	1359	1	314	0.0	598
XM_002716705.3	Q8R1P4	21.552	116	80	3	994	1320	88	199	0.13	35.0
XM_002716705.3	Q0VF22	27.778	108	64	3	502	810	184	282	6.9	30.0
XM_002716705.3	Q8C4Y3	40.000	25	14	1	549	475	256	279	7.1	30.0
XM_002716705.3	Q9JK38	27.397	73	40	2	337	555	19	78	8.3	29.3
XM_002716705.3	Q8C5K5	26.506	83	59	1	580	828	177	257	8.8	29.6
# BLAST processed 1 queries
```

Concise.
Useful for large scale analysis.
Tab-delimited text is easy to parse.
You can modified the format. (explain later)

Anatomy of BLAST output (format 6)

query	subject	%identity	align-len	mismatch	gap_open	q_start	q_end	s_start	s_end	evalue	bit-score
spo:SPAC212.11	sce:YMR190C	27.297	370	238	12	1169	1525	653	1004	7.08e-23	105
spo:SPAC212.11	sce:YDR021W	38.053	113	64	3	1417	1529	267	373	1.42e-14	75.9
spo:SPAC212.11	sce:YNL112W	25.076	331	225	12	1204	1525	148	464	1.06e-12	70.9
spo:SPAC212.11	sce:YBR237W	24.403	377	237	17	1190	1532	276	638	1.68e-12	70.5
spo:SPAC212.11	sce:YOR204W	28.505	214	128	8	1352	1559	339	533	2.80e-12	69.7
spo:SPAC212.11	sce:YOR046C	28.205	234	140	8	1310	1530	227	445	1.65e-11	66.6
spo:SPAC212.11	sce:YDR243C	23.512	336	227	10	1208	1524	216	540	7.40e-11	65.1
spo:SPAC212.11	sce:YGL078C	22.689	357	239	11	1188	1527	130	466	7.62e-11	64.7
spo:SPAC212.11	sce:YPL119C	27.619	210	129	8	1352	1556	351	542	2.84e-10	63.2
spo:SPAC212.11	sce:YGL064C	22.195	410	216	16	1215	1541	166	555	8.25e-10	61.6
spo:SPAC212.11	sce:YDL084W	24.424	217	156	5	1308	1524	201	409	7.84e-07	51.6
spo:SPAC212.11	sce:YHR169W	27.273	143	93	4	1418	1555	257	393	1.09e-06	51.2
spo:SPAC212.11	sce:YLR276C	26.667	210	120	10	1189	1377	38	234	2.85e-06	50.1
spo:SPAC212.11	sce:YLR276C	34.146	41	27	0	1484	1524	386	426	0.12	35.0
spo:SPAC212.11	sce:YLL008W	21.965	346	233	13	1195	1524	258	582	4.71e-06	49.3
spo:SPAC212.11	sce:YPL082C	24.074	108	81	1	1417	1524	1648	1754	0.002	40.8
spo:SPAC212.11	sce:YGL070C	37.838	37	23	0	1583	1619	52	88	2.3	29.3
spo:SPBPB10D8.05C	sce:YPL092W	30.730	397	224	9	10	356	11	406	3.04e-47	165
spo:SPBPB10D8.05C	sce:YGL195W	31.034	58	37	2	140	195	2014	2070	0.37	30.8
spo:SPBPB10D8.05C	sce:YDR283C	51.613	31	14	1	221	250	589	619	0.66	30.0
spo:SPBPB10D8.05C	sce:YBR028C	35.135	37	24	0	229	265	127	163	0.84	29.6
spo:SPBPB10D8.05C	sce:YPL027W	27.957	93	54	3	138	221	53	141	3.1	27.3
spo:SPBPB10D8.05C	sce:YDR161W	28.571	49	35	0	318	366	102	150	3.2	27.7
spo:SPBPB10D8.05C	sce:YER166W	27.451	51	35	1	50	98	1273	1323	7.0	26.9
spo:SPBPB10D8.05C	sce:YGR040W	23.256	86	55	3	214	299	3	77	7.8	26.6
spo:SPBPB10D8.05C	sce:YAR019C	44.444	18	10	0	235	252	30	47	9.5	26.2
spo:SPBC359.02	sce:YOR368W	29.885	87	56	2	53	138	7	89	0.88	29.3
spo:SPCC330.07C	sce:YHR197W	26.761	71	39	3	65	130	287	349	2.0	28.9
spo:SPCC330.07C	sce:YGR224W	23.881	67	50	1	181	247	152	217	5.1	27.7

Multiple queries

ウサギのあるcDNA断片データを100個得た。これが何の遺伝子をコードしているのかを、マウスタンパク質データベースに対してsimilarity searchする事により明らかにする。多数の配列を解析したいときにこそCUIの本領発揮。

- Query: rabbit_100mRNAs.nuc.fastaa
- DB: mouse_proteins.pep.fasta
- options:
  - evalue < 1.0e-5; format 7; 高速化のためにCPUを4コア使う。

1. Format DB

新たに作成する必要はない。

2. Search

自分で考えてみよう

Why and when command-line BLAST is useful?

- Automated
- Many queries
- Use local database (unpublished)

Exercise

- (詳細はwiki)
- ex 4-1: mapping cDNA/EST to a genome
- ex 4-2: mapping a PacBio read to a genome
- ex 4-3: compare two closely related genome
- ex 4-4: BLASTP homolog candidate search
- ex 4-5: BLASTX: annotating RNA-seq de novo assembled contigs
- ex 4-6: tblastn: mapping proteins to a genome
- ex 4-7: blastn: CRISPR off-target check

Installation and configuration

- Installation
  - follow manual
    https://www.ncbi.nlm.nih.gov/books/NBK279671/
    https://www.ncbi.nlm.nih.gov/books/NBK52640/

- Functionality offered by BLAST+ applications
  - 1) Search tools
  - 2) BLAST database tools
  - 3) sequence filtering tools

- Setup and configuration
  - blast programs　のパスを通す
  - 環境変数 BLASTDB を設定する

```
export BLASTDB=$HOME/myblastdb
```

Resources

- User manual: https://www.ncbi.nlm.nih.gov/books/NBK279670/

- help

Beyond standard blast commands

- pairwise alignment by using blast commands
- blastdbcmdを使った配列データのmanipulation
- advanced blast search tools

Pairwise BLAST

BLASTプログラムを使って、2つの配列のローカルアライメントをとることも可能である。

- Query 1: OcAtg3.aa.fa
- Query 2: MmAtg3.aa.fa

Alignment

```
$blastp -query OcAtg3.aa.fa -subject MmAtg3.aa.fa
```

出力フォーマットは柔軟に変更できる。例えば、標準的なformat 6にアライメント結果を追加

```
$blastp -query OcAtg3.aa.fa -subject MmAtg3.aa.fa \
 -outfmt "6 std qseq sseq"
```

BLAST DB as a sequence DB – **blastdbcmd (1)**

makeblastdbで作成したblast dbは、配列データベースとしても、とても便利である。
- IDをキーに配列を取得できる。部分配列や相補鎖の取得も可能。
- アノテーションが付されていれば、その情報も取得できる。
- NCBI純正のnrやntデータベースであれば、taxonomy 情報も取得できる。

blastdbcmd を使う。

Format DB
mouse_proteins.pep.fasta のアミノ酸配列データベースを作成

```
$makeblastdb -in mouse_proteins.pep.fasta -dbtype prot -parse_seqids
(一度作成すればよい)
```

Retrieve sequence
Quick startのBLAST検索でトップヒットだった、「Q9CPX6」の配列を取得する。

```
$blastdbcmd -db mouse_proteins.pep.fasta -entry Q9CPX6
```

blastdbcmd (2)

複数の配列をまとめて取得する。
Quick startのBLAST検索でヒットしたすべての配列を取得する。

Prepare ID list
Quick startのBLAST検索でヒットしたすべての配列のIDを1行1ID、テキストファイルに保存。数が少ない場合は人間が入力してもよいが、BLAST結果のタブ区切りフォーマット(format6)ファイルからIDの列のみを抽出する方がミスも防げて良い。
(ex) BLAST結果がformat6で、out.bl6.txtの名前で保存されていた場合。

```
$cut -f 2 out.bl6.txt  > idlist.txt
```

Retrieve sequence
-entry_batch オプションの引数にidlist.txtを与える

```
$blastdbcmd -entry_batch idlist.txt -db mouse_proteins.pep.fasta
```

blastdbcmd (3)

ゲノム上の一部の領域の配列のみを取得する。相補鎖の配列を取得する。

例）buchnera.genome.fasta　はバクテリア*Buchnera aphidicola* の全ゲノム配列である。dnaK遺伝子はマイナス鎖の162206-164119にコードされている事がわかっている。この領域の配列を取得したい。

Build DB

```
$makeblastdb -in buchnera.genome.fasta  -dbtype nucl -parse_seqids
```

Retrieve sequence

```
blastdbcmd -db buchnera.genome.fasta -entry buc \
 -range 162206-164119 -strand minus
```

blastdbcmd (4)

BLASTのformat6/7の結果テーブルにはヒットした遺伝子のIDのみが記録され、descriptionの記載がないのでどのような遺伝子かわからない。blastdbcmdを使ってblastdbからdescription情報を引っ張ってくることができる。

Retrieve description by ID
Quick startのBLAST検索でトップヒットだった、「Q9CPX6」のdescriptionを取得する。

```
blastdbcmd -db mouse_proteins.pep.fasta -entry Q9CPX6 -outfmt "%t"
```

-outfmt オプションは柔軟に書式をアレンジできる。例えば、IDもあわせて取得しセミコロンで区切って表示する場合。

```
blastdbcmd -db mouse_proteins.pep.fasta -entry Q9CPX6 \
 -outfmt "%i; %t"
```

(タブ区切りにしたい場合は、ctrl+v のあとtabを入力する)

応用問題　queryが複数の場合。例えば、blast format6の出力すべての行にdescriptionをつけるような場合。

blastdbcmd (6)

NCBI純正のnrやntデータベースであれば、taxonomy 情報も取得できる。ただしデータベースの準備が必要。

Quick startのBLAST検索でトップヒットだった、「Q9CPX6」のtaxonomy情報を取得する。（Q9CPX6はNCBI nrに含まれる）

Setup nr/nt database and taxonomy

1. NCBIのFTPサイト (ftp://ftp.ncbi.nlm.nih.gov/blast/db/) からpre-formatted nr（もしくはnt）DBの圧縮ファイルをダウンロードし解凍する。
2. 同じくNCBIのFTPサイトから、taxdb.tar.gzをダウンロードし解凍する。nr (or nt)と同じディレクトリに保存。
3. [optional] 環境変数 BLASTDB に上記ディレクトリを指定する。
4. 基生研のbias4をはじめ共用計算機にはサーバーの管理者によって1-3の設定が完了している事が多いので、通常ユーザーがこの作業をする必要はない。

Retrieve taxonomy information

「Q9CPX6」のtaxonomy informationを取得する。

```
$blastdbcmd -db nr -entry Q9CPX6 -outfmt "%i; %T; %S; %L; %K" \
 -target_only

Q9CPX6.1; 10090; Mus musculus; house mouse; Eukaryota
```

Advanced BLAST search tools

- PSI-BLAST: Position Specific Iterative BLAST
  - Automatically generates a position specific score matrix (PSSM)
  - PSI-BLAST finds sequences significantly similar to the query in a database search and uses the resulting alignments to build a PSSM for the query. With this PSSM the database is scanned again to eventually pull in more significant hits, and further refine the scoring model.
  - More sensitive than standard BLAST
- RPS-BLAST: Reverse Position-Specific BLAST
  - RPS-BLAST uses the query sequence to search a database of pre-calculated PSSMs, and report significant hits in a single pass.
  - Used in CD-search (Conserved Domain search) at NCBI website.
- DELTA-BLAST
  - DELTA-BLAST searches a protein sequence database using a PSSM constructed from conserved domains matching a query. It first searches the NCBI CDD database to construct the PSSM.

落ち穂拾い

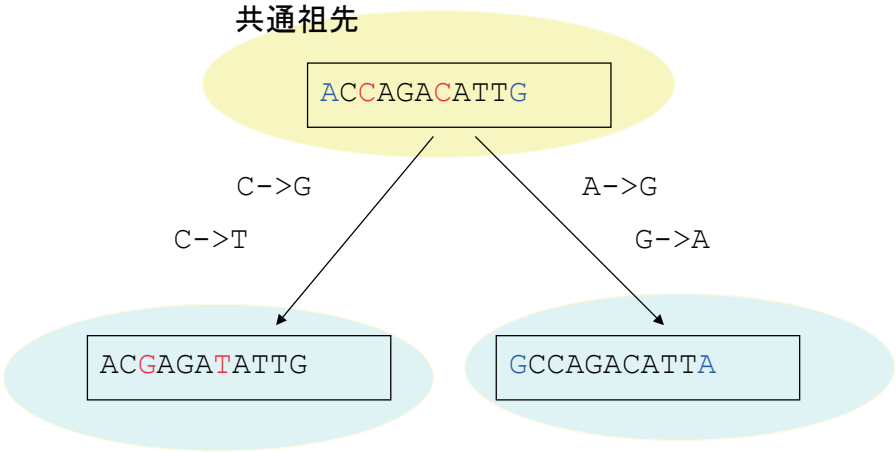
- BLASTN: -task option
- 繰り返し配列のマスキング (SEG, DUST)
- combine multiple databases or use a subset of built DB
- remote blast
- BLAST+ and legacy BLAST

BLAST Inside
配列検索の理論的背景

内山郁夫
基礎生物学研究所

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
 |  |||  |||
GCCAGACATTA
```

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

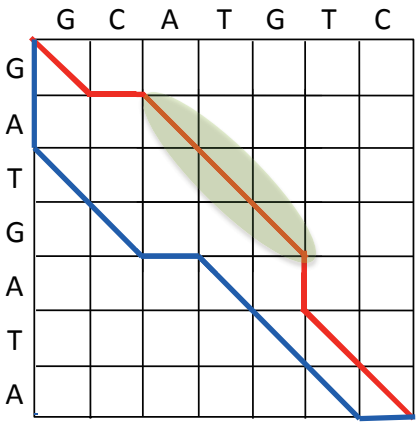
GCATG---AGGA

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

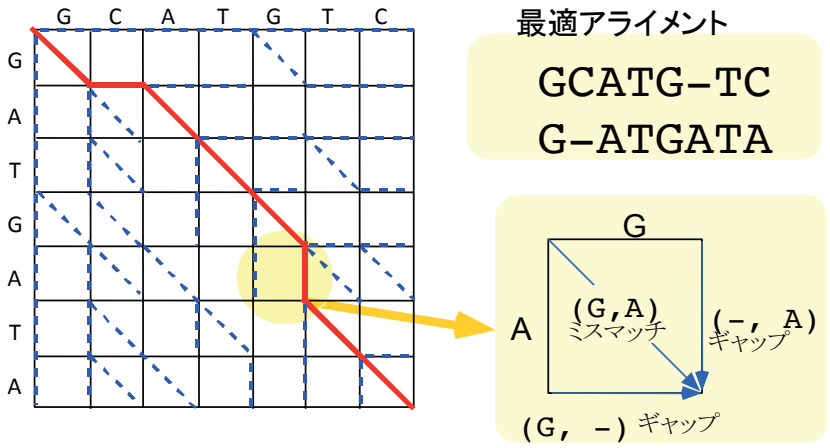
アライメント2

--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



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

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。

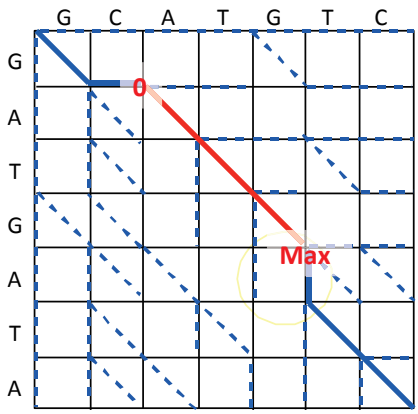
- ローカル（局所）アライメント
  - 部分配列間のアライメントで類似度が最大となるものを求める。
  - 配列の一部の領域（ドメイン）のみに相同性がある場合、もしくは相同性があっても類似性が低く、一部の領域のみしか類似性が認識できないような場合に効果的。
  - 類似度が低い場合にスコアがマイナスになるようなスコア体系を用いる必要がある。

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

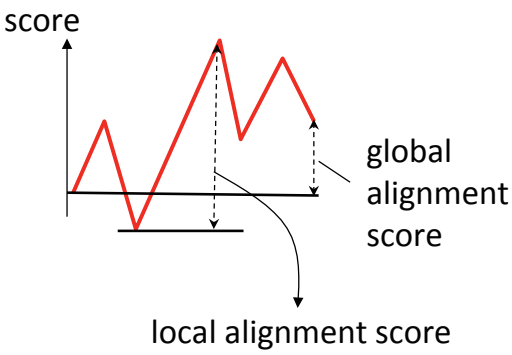
動的計画法による局所アライメント

Smith-Waterman法



局所最適アライメント

```
GCATG-TC
G-ATGATA
```



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

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

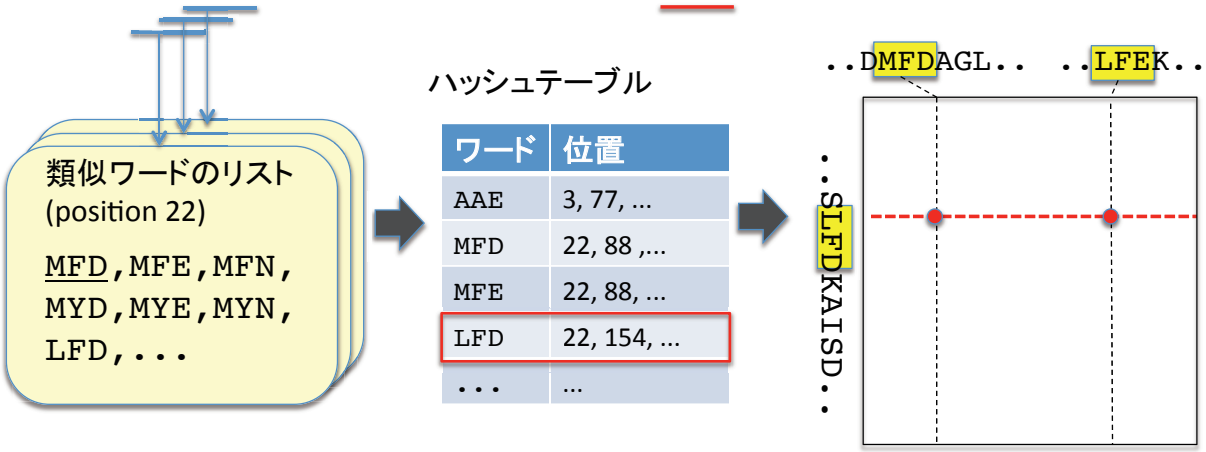
初期検索

問い合わせ(Query)配列

...MFDAGLNDGE...

データベース(Subject)配列

...SLFDKAISDGD...

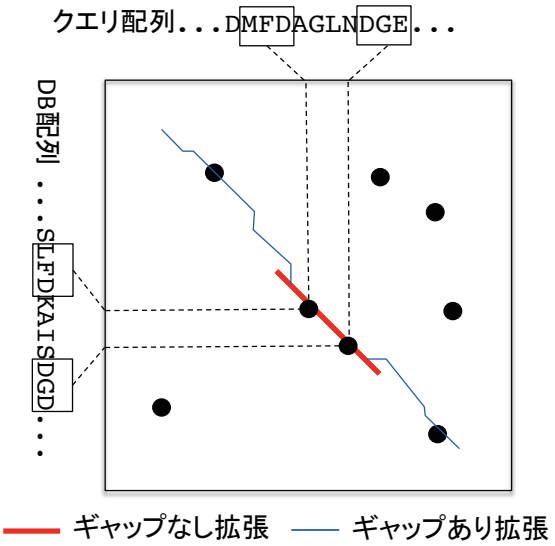


関連するオプション

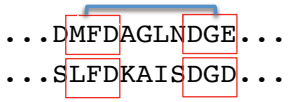
-word_size　初期検索で使うワードのサイズ

-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）

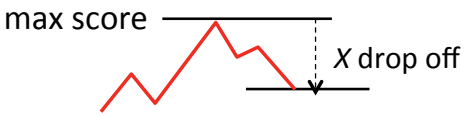


1) 一定のウィンドウ内に2回ヒットが見つかる領域をとる



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



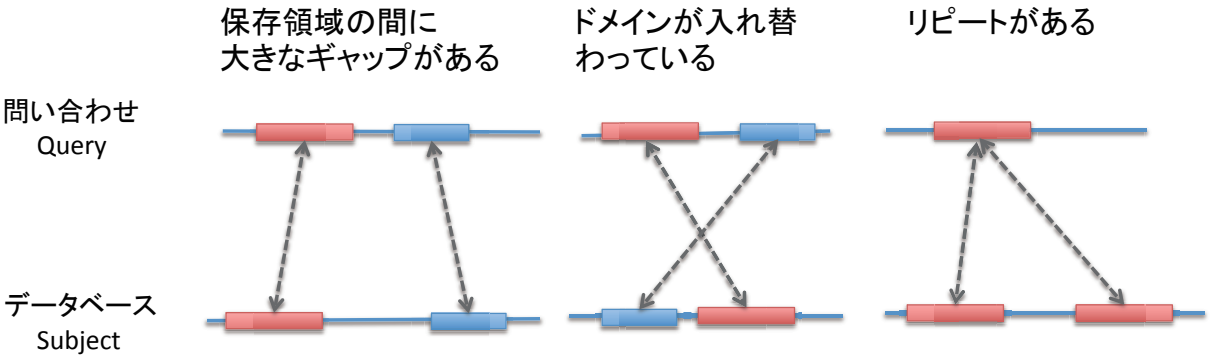
4)統計的に有意なスコアを持つヒットを残す

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

形や大きさ、性質が似たアミノ酸間には高いスコア
似ていないアミノ酸間には低いスコア

	P	A	G	N	Q	D	E	T	S	C	V	I	M	L	K	R	H	F	Y	W
P	6																			
A	1	2																		
G	0	1	5																	
N	0	0	0	2																
Q	0	0	-1	1	4															
D	-1	0	1	2	2	4														
E	-1	0	0	1	2	3	4													
T	0	1	0	0	-1	0	0	3												
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								
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					
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		
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

small　amide　acidic　hydroxyl　aliphatic (V,I,L)　basic　aromatic

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)

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

```
PDSTIQMINRYLAKHPEQTNRFRILVCGGDG
|    |         |      |||||||
PIKAVQLCTLL....PYYS..ARVLVCGGDG

TIDKANLPVLPPIAVLPLGTGNDLARCLRWG
  |||||    ||||||||||||||||||||
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）

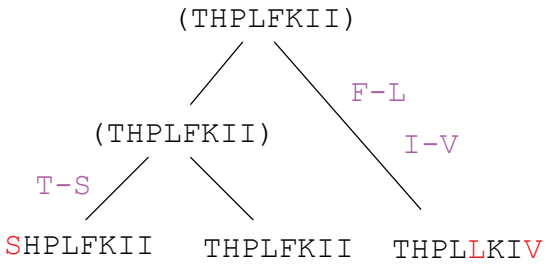
対数オッズスコア

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

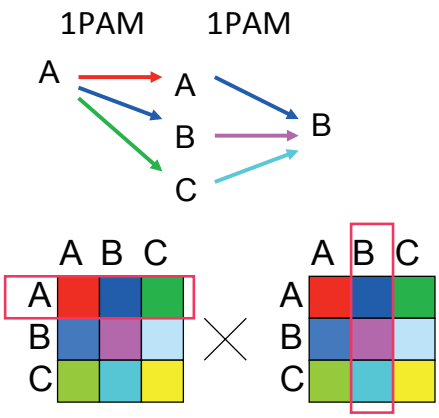
$$M(a,b) = \frac{f(a,b)}{\sum_{x \neq a} f(a,x)} m(a)$$

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により、nPAMの置換確率行列 M_n を計算



4）対数オッズをとって、PAMn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) CYDGRGLSYRGLARTTLSGAPCQPWAS

HGFA_HUMAN ( 286) CFLGNGTGYRGVASTSASGLSCLAWNS

UROK_CHICK (  79) CYSGNGEDYRGMAEDPGCLYWDHPSVI

UROT_HUMAN ( 215) CYFGNGSAYRGTHSLTESGASCLPWNS

APOA_HUMAN (  28) CYHGDGQSYRGTYSTTVTGRTCQAWSS
PLMN_BOVIN ( 358) CYHGNGQSYRGTSSTTITGRKCQSWSS
PLMN_HUMAN ( 377) CYHGDGQSYRGTSSTTTTGKKCQSWSS
PLMN_MACMU ( 377) CYHGDGQSYRGTSSTTTTGKKCQSWSS
PLMN_MOUSE ( 377) CYQSDGQSYRGTSSTTITGKKCQSWAA
  PLMN_PIG ( 358) CYRGNGESYRGTSSTTITGRKCQSWVS

HGFL_HUMAN ( 283) CFRGKGEGYRGTANTTTAGVPCQRWDA
HGFL_MOUSE ( 292) CFRGKGEDYRGTTNTTSAGVPCQRWDA
```

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
(Da a. 1978)

120
160
210
250

BLOSUM
(H a. 1992)

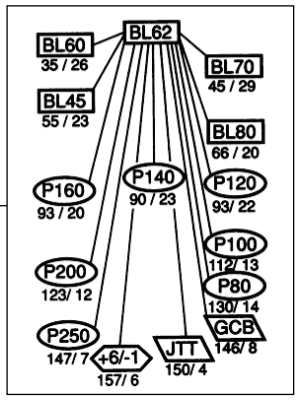
80
62 → BLAST
50 → FASTA
45

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

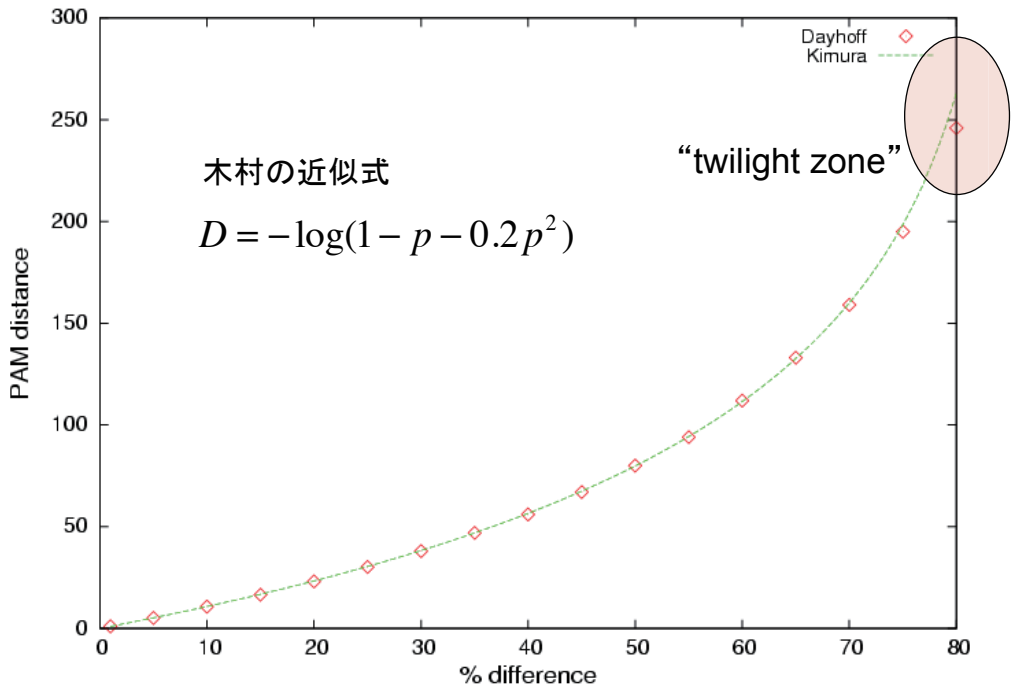
BLOSUM 45

	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, λ：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x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

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

[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→0のとき、p=1-e^{-E}→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)

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 λ, 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、
$E=5\times10^{6}\times 200\times 2^{-30} = 0.93$

フィルタープログラム

SEG - 単純な（アミノ酸組成の偏った）領域を除く

使用前

```
>SOS_DROME son-of-sevenless
MFSGPSGHAHTISYGGGIGLGTGGGGGSGG
SGSGSQGGGGGIGIGGGGVAGLQDCDGYDF
TKCENAARWRGLFTPSLKKVLEQVHPRVTA
KEDALLYVEKLCLRLLAMLCAKPLPHSVQD
```

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 HOMEOBOX/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 HOMEOBOX/POU DOMAIN PRO

Query: 9 AHTISYGGGIGLGTGGGGGSGGSGSGSQGGGGGIGIGGGGV 49
         A +I +    G G GGGGG GG G G+ GGGGG+  G   V
Sbjct:18 AGSIVHSDAAGAGGGGGGGGGGGGGGAGGGGGGMQPGSAAV 58
```

使用後

```
>SOS_DROME son-of-sevenless
MFSGPSGHAHTISYXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXVAGLQDCDGYDF
TKCENAARWRGLFTPSLKKVLEQVHPRVTA
KEDALLYVEKLCLRLLAMLCAKPLPHSVQD
```

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、その後の拡張フェイズでは通常の配列と同様に使う 。

    →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

相同配列間で観察される置換の頻度

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

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

p', p'' に合わせて補正した置換頻度

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

クエリ配列の
アミノ酸頻度

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）
  - evalue（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

大規模なBLAST検索の効率化

内山郁夫
基礎生物学研究所

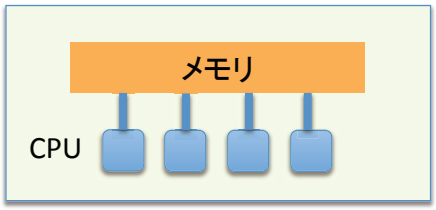
大規模なBLAST検索の例

- 遺伝子予測
  - 発現遺伝子全体 x ゲノムDNA配列
  - ゲノムDNA配列 x タンパク質DB（blastx）
- 機能アノテーション
  - ゲノム内の遺伝子全体 x タンパク質DB
- 比較ゲノム解析
  - ゲノム内の遺伝子全体 x 別のゲノムの遺伝子全体

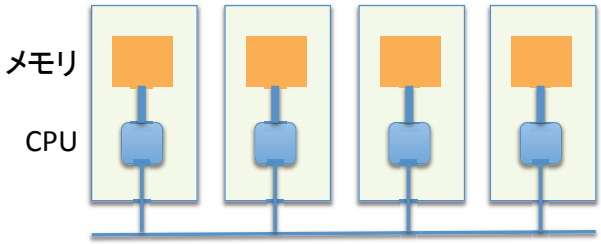
大規模検索は並列化によって高速化できる

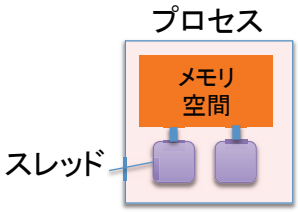
並列計算の基礎

共有メモリ型

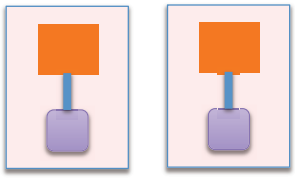


分散メモリ型





マルチスレッド



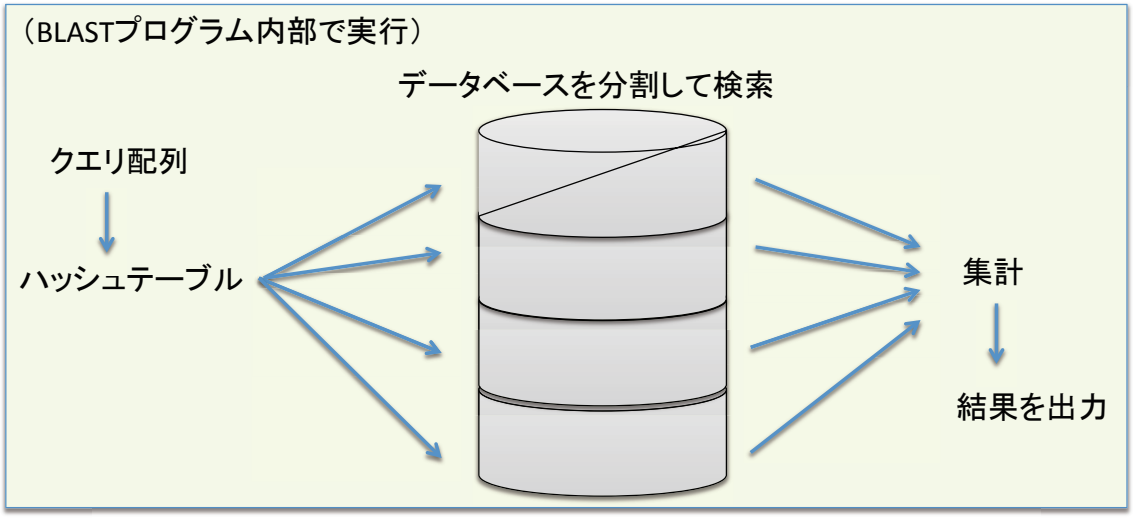
マルチプロセス

スレッド数及びメモリ空間の合計は、実CPUコア数、実メモリ数を超えて使用することができるが、一般に効率は落ちる（特にメモリが超えた場合は深刻な遅延を生じることがある）

BLASTの並列実行

マルチスレッドによる並列化

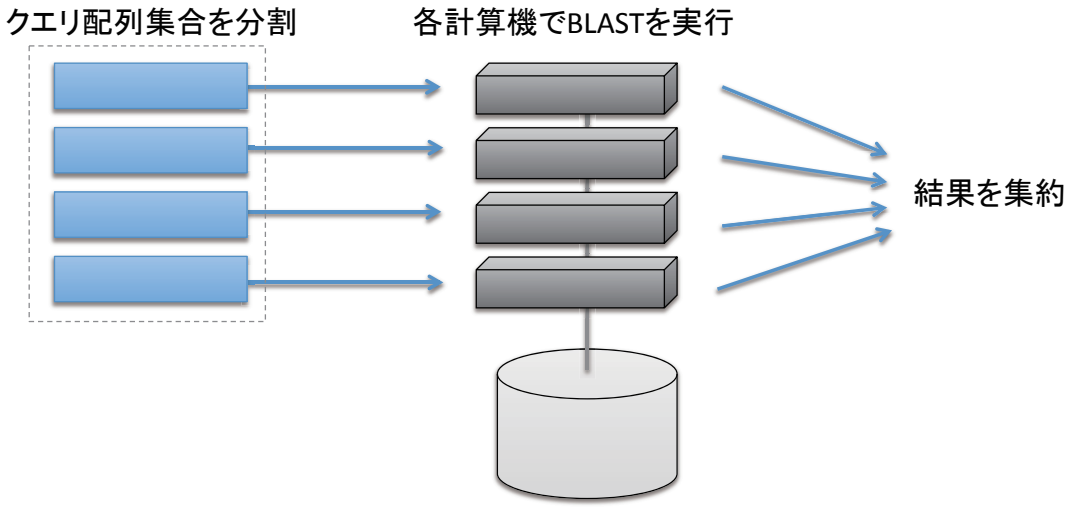
- BLASTの-num_threads オプションを使う
- スレッド数は、実行する計算機に搭載されたコア数を超えないようにする
- クエリひとつでも高速化が可能



BLASTの並列実行

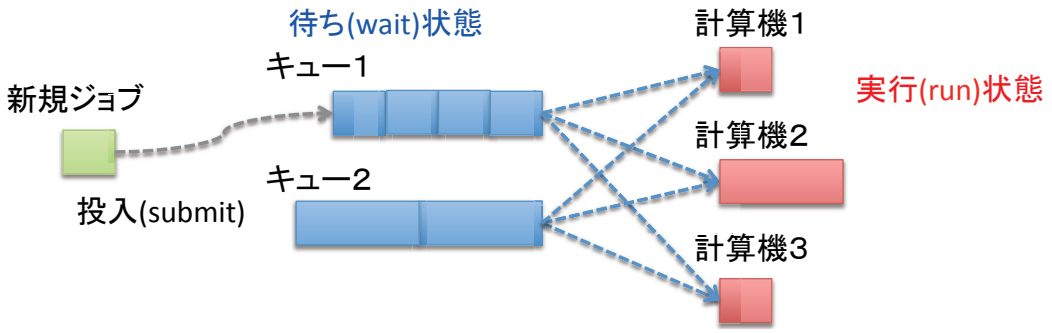
分散処理による並列化

- クエリ配列が大量にある場合、クエリ配列を複数のファイルに分割して、分散処理環境を用いて並列にBLASTを走らせるのが効果的



ジョブ管理システム

- 「キュー（待ち行列）」にたいして「ジョブ」を投入する形で、プログラムの実行をシステムに委託する。
- 「キュー」に入ったジョブは、複数の計算機に分配され並列に実行される。
- 既定の処理量を超えるジョブが投入された場合、後から投入されたジョブは先行するジョブが終了するまで「待ち状態」となる。
- 対象とするマシンや、想定されるジョブの大きさによって、複数のキューが用意されている。



Sun Grid Engineでのジョブの実行

- クラスター計算機でプログラムを走らせるには、走らせるコマンドを記述したジョブファイルを作成してから、qsub コマンドを使う

ジョブの投入　`qsub [-q キュー名] ジョブファイル`

ジョブの状態の確認（自分のジョブ）　`qstat`

（全員のジョブ）　`qstat -u '*'`

（キューごとのサマリ表示）　`qstat -g c`

ジョブの削除　（特定のジョブ）　`qdel ジョブ番号`

（全部キャンセル）　`qdel '*'`

NIBB 生物情報解析システムにおけるキューの構成

	分散並列処理型			共有メモリ型		
キュー名	**small**	**medium**	**large**	**smps**	**smpm**	**smpl**
ジョブの特徴	短時間 並列多	中時間	長時間	中メモリ	大メモリ	最大メモリ
利用ノード	node01 - node40			ldas-smp		
最大実行時間/job	6時間	72時間	no limit	no limit		no limit
最大メモリ/job	4.8GB	4.8GB	4.8GB	500GB	1TB	4TB
最大ジョブ数/キュー	580	200	20	8	4	1

BLASTは、クエリを細かく分割して並列度を上げてsmallキューで実行するのがよい

今回の実習では、講習会用に特別に作ったキュー blast を使う

実習：2ゲノム間の遺伝子比較

- 出芽酵母 *Saccharomyces cerevisiae*
  ファイル sce
- 分裂酵母 *Schizosaccharomyces pombe*
  ファイル spo

NCBIで配布されたファイルのエントリ名に生物種名のコードを付加

```
% sed 's/^>/>sce:/' GCF_000146045.2_R64_protein.faa > sce
% sed 's/^>/>spo:/' GCF_000002945.1_ASM294v2_protein.faa > spo
```

```
>sce:NP_001018029.1 hypothetical protein YBR230W-A
MRNELYQLWCVASAARGVAKSSFVRANSAMCEYVRTSNVLSRWTRDRQWE
>sce:NP_001018030.1 L-serine/L-threonine ammonia-l
MSIVYNKTPLLRQFFPGKASAQFFLKYECLQPSGSFKSRGIGNLIMKSAI
LSLPCTVVVPTATKKRMVDKIRNTGAQVIVSGAYWKEADTFLKTNVMNKI
QDLKSQHISVNKVKGIVCSVGGGGLYNGIIQGLERYGLADRIPIVGVETN
VISNQTFEYARKYNTRSVVIEDKDVIETCLKYTHQFNMVIEPACGAALHL
NTIKDLEEALDSMRKKDTPVIEVADNFIFPEKNIVNLKSA
>sce:NP_001018031.2 Adf1p [Saccharomyces cerevisia
MGKCSMKKKGVGKNVGVGKKVQKKRSISTAERKRTKLQVEKLNKSSETMI
EKDSKVREQIRTEKSKTNDSMLKQIEMISGFSL
```

```
>spo:NP_001018179.1 hydroxymethylbilane synthase (
MPSCTSFPIGTRKSKLAVIQSEIIREELEKHYPHLEFPIISRDTIGDEIL
ILVHSLKDLPSEMPDGMVIACIPKRSCPLDAIVFKAGSHYKTVADLPPGS
TRLAKLDAPDSQFDCLVLAAAGLFRLGLKDRIAQMLTAPFVYYAVGQGAL
RALMKRLQGGCAIPIGVQTDVLAISNSSYRISLLGTVLSADGLRAAFGNA
EEHQRSSDSEESLKNY
>spo:NP_001018181.1 poly(A) polymerase Cid14 [Schi
MGKKSVSFNRNNYKKRKNERTEPLPRRIFKNDKPSKFKSKRKEKDKNSDA
NDSEGIRDKGGVEISNKNDPYIQFGKADPLEPLEKPDLPEEAIKRGEPTI
WNSDEDEDSVSNDKSKNNESLKKSSKNEIPGFMRQRGRFFHEANEKSDSN
FHQDILHFIDYITPTPEEHAVRKTLVSRINQAVLQKWPDVSLYVFGSFET
AHHLKKLKLASEVQVITTANVPIIKFVDPLTKVHVDISFNQPGGLKTCLV
```

生物情報解析システム bias4.nibb.ac.jpにログインする。

```
% ssh bias4.nibb.ac.jp
```

実習用ディレクトリに移動

```
% makeblastdb –in spo –dbtyhpe prot –parse_seqids
```

配列集合を分割する

```
# FASTA形式の配列を、長さの和がBLOCK_SIZEを超えるごとに分割

BLOCK_SIZE=100000

foreach line in Lines do
    if (line が '>' で始まる) then
        # FASTA形式のタイトル行
        if (savedLen >= BLOCK_SIZE) then
            新しいファイルをオープンし、そこに
                SavedLines を出力する
            SavedLines を空にする
            savedLen = 0
        fi
    else
        # 配列行
        savedLen += length(line)
    fi
    SavedLines に line を加える
done
# まだ出力されていない行
新しいファイルをオープンし、そこに SavedLines を出力する
```

配列を分割する

- サンプルスクリプト split_seq.pl

```
split_seq.pl [–BLOCK_SIZE=block_size]
             [-QUERY_OUT_DIR=query_dir]
             [-QSUB_SCRIPT_FILE=[script_file]] query_file
```

FASTA形式のファイル *query_file* 中の配列を長さの和がblock_sizeを超えるごとに別のファイルに分割して、ディレクトリ *query_dir* 以下に *query_file.fileNo*（*fileNo*は1から*split_num*までの数字）という名前で格納する。合わせて、qsub_blast.shというqsub 用のスクリプトを作成する。

```
% ls
sce  spo
% makeblastdb –in spo –dbtype prot
% split_seq.pl sce
% ls
qsub_blast.sh  query_sce sce  spo  spo.phr  spo.pin  spo.psq
% ls query_sce
spo.1  spo.2  spo.3  ...
```

アレイジョブ

- 同じコマンドを、入力（及び出力）ファイルを変えて複数回並行して実行させる。
- qsub のオプションに –t 開始番号–終了番号を指定し、スクリプトファイルに変数 ${SGE_TASK_ID}を埋め込むと ${SGE_TASK_ID} を開始番号から終了番号まで順次置き換えたジョブとしてサブミットされる。
- サブミットするジョブ数が大きいときは、最大の同時実行ジョブ数を-tcオプションで指定する。

```
#!/bin/sh
#$ -t 1-200
#$ -tc 50
#$ -cwd
command input.${SGE_TASK_ID} > output.${SGE_TASK_ID}
```

input.1 からinput.200までのファイルを入力とし、結果を対応するoutput.1からoutput.200までのファイルに出力するジョブ200個を生成し、最大同時実行数50で実行する

qsub による実行

- スクリプト qsub_blast.sh を編集

```
% emacs qsub_blast.sh
```

```
#!/bin/sh
#$ -cwd
#$ -t 1-30
#$ -tc 32
#$ -N blastjob
#$ -S /bin/sh
DB=spo
SEQDIR=query_sce
OUTPUTDIR=blastout_sce

OPTIONS=(-outfmt 6 -evalue 0.001)

if [ ! -d $OUTPUTDIR ]; then
        mkdir $OUTPUTDIR
fi
blastp -db $DB -query query_sce/sce.$SGE_TASK_ID -out $OUTPUTDIR/sce.blast.$SGE_TASK_ID "${OPTIONS[@]}"
```

検索対象DB名を変更

必要に応じてオプションやコマンド行を変更

カーソルキーで移動
デリートキーで文字を消去して書き換え
保存するには Control-X Control-S を順に押す
終了するには Control-X Control-C を順に押す

qsub による実行

- ジョブをサブミット（blast キューを使う）

```
% qsub –q blast qsub_blast.sh
```

- 実行状況の確認

```
% qstat
```

（何も表示されなくなったら終了）

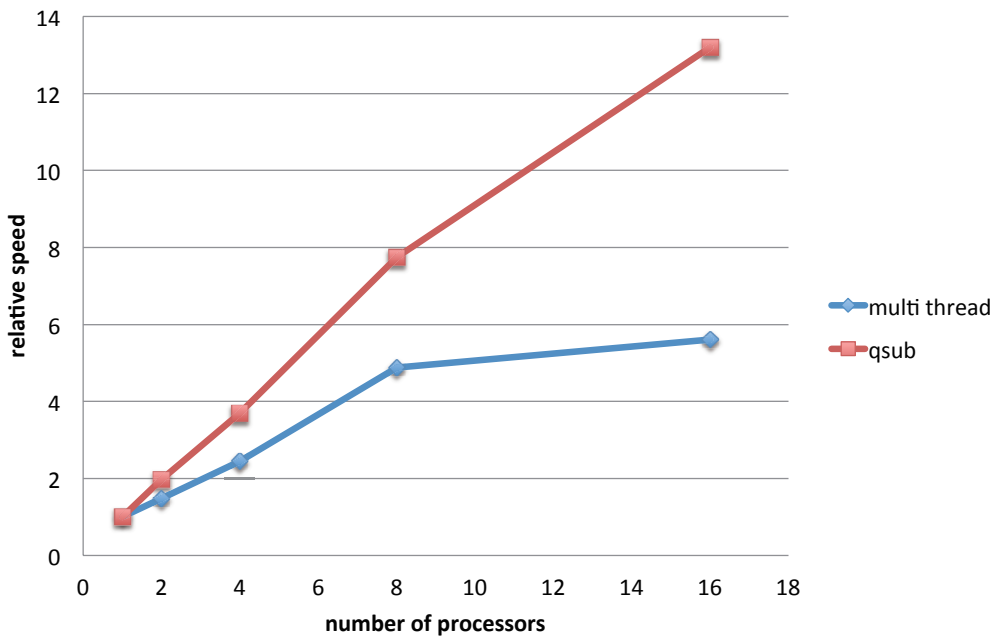
- 結果を一つにまとめる

```
% cat blastout_sce/sce.blast.* > sce-spo.blast
```

（参考） 元の順番通りにつなげたい場合、bashだと以下のようにして行える

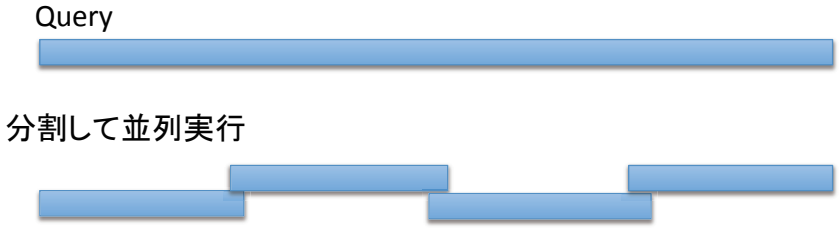
```
for ((i=1; i<=30; i++)); do
  cat blastout_sce/sce.blast.$i >>sce-spo.blast
done
```

並列度と検索速度の関係



クエリ配列の分割

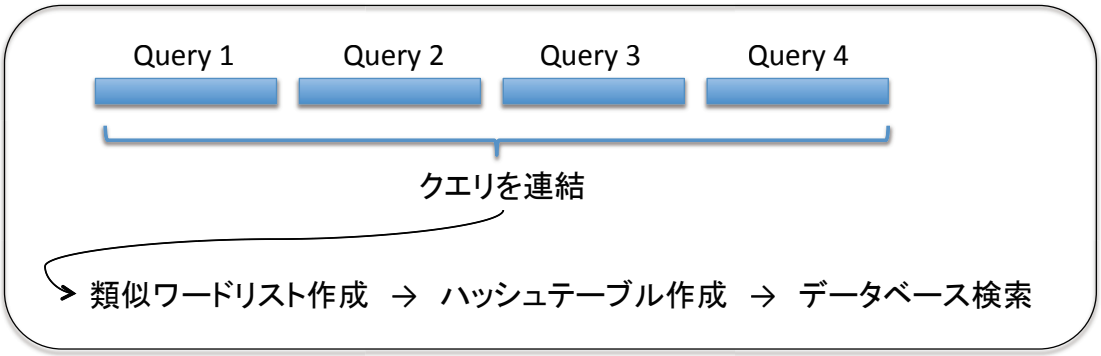
長いクエリ配列が一つある場合は、配列を分割することにより並列に実行できる



`-query_loc` 開始位置-終了位置
クエリ配列の開始位置から終了位置までをクエリとして使う

クエリ配列の連結

BLASTでは、多数のクエリを検索する際は、クエリを連結することによって検索回数を少なくしている。従って、クエリを分割しすぎるとかえって効率が悪くなる。



一方、クエリの連結によってメモリの使用量は増大する

連結する際の長さは、環境変数 BATCH_SIZE で調節できる

BLAST 応用例 1

BLASTによる遺伝子モデルの簡易機能アノテーション

重信 秀治

Aim
- 新規に非モデル生物の網羅的遺伝子モデルを構築し、それらの遺伝子配列が手元に有るとする。これらの機能アノテーションをBLASTによる配列類似性解析によって行いたい、という状況を考える。
- 1) 近縁のモデル生物に対するBLAST検索。
- 2) GeneOntology termをアサインする。

Case study 1: ソラマメアブラムシの遺伝子アノテーション

大目的：非モデル昆虫「ソラマメアブラムシ」*Megoura crassicauda* の網羅的遺伝子レパートリーを構築しそれらの機能アノテーションをしたい。

実験・解析の経過：ソラマメアブラムシから抽出したRNAで、Illumina MiSeqによるRNA-seqを行った。Trinityによるde novo assembly を行い、その結果 21,730のcontigsが得られた。その中からORFを推定したところ、16,968のcoding genesが推定された。(Shigenobu et al., unpublished)

実戦演習課題：上記の解析で推定されたcoding genesの中から、本コースの練習用に400遺伝子を抽出した。これらをアミノ酸配列に翻訳した配列を

`MEGCR_proteins400.pep.fasta`

として用意した。それぞれのタンパク質がどのような機能を持っているか、BLAST検索を駆使してアノテーションせよ。

簡易アノテーションの戦略 1

基本戦略：機能アノテーションが充実しているモデル生物を軸にアノテーションを行う。

- 1) モデル昆虫 Drosophila melanogaster のタンパク質データベースに対するBLAST検索を行う。トップヒットの遺伝子をホモログと見なす。
- 2) D. melanogaster ホモログのGene Ontology (GO) termをソラマメアブラムシにアサインする。

Data

- Target: MEGCR_proteins400.pep.fasta

- References:
  - D. melanogaster 全タンパク質配列（flybase）: dmel-all-translation-r6.13.fasta
  - D. melanogaster GO term annotation file: gene_association.fb

Workflow

- 1. BLAST against Dmel
  - 1.1 BLAST search against Dmel proteins
    - output table format
  - 1.2 Retrieve description line
- 2. GO term assignment
  - 2.1 Build FBgn ⇔ FBpp ⇔ GO table
  - 2.2 Assign GO terms

簡易アノテーションの戦略 2

基本戦略：NCBI nr (non-redundant protein database) に対するBLAST検索を行う。

- 1) NCBI nrに対するBLAST検索を行う。ただし、今回は計算済みのタブ区切りテキストを提供する。
- 2) description line を取得する。
- 3) トップヒットした配列の生物種とtaxonomy information を得る。

Data

- **`MEGCR_proteins400.pep.fasta_ALL.fasta.vs.nr.blast.fmt7.txt --`** query=MEGCR_proteins400.pep.fasta; db=nr のBLAST検索結果 in format7

Workflow

- 1. Retrieve description line
- 2. Retrieve taxonomy information
- 3. Merge this annotation table with Dmel-based annotation.

advanced

簡易アノテーションの戦略 3

基本戦略：タンパク質「モチーフ」の有無を調べる

- 1) RPS-BLAST

- Notes:
  - 実践上はInterProScanを使うのが便利。
  - モチーフ検索はhmmerを使ったPfam検索などほかにも多数の手法がある。

BLAST 応用例 2

BLASTによるオーソログ解析

内山 郁夫

BLAST tabular 形式

query	subject	%identity	align-len	mismatch	gap_open	q_start	q_end	s_start	s_end	evalue	bit-score
spo:SPAC212.11	sce:YMR190C	27.297	370	238	12	1169	1525	653	1004	7.08e-23	105
spo:SPAC212.11	sce:YDR021W	38.053	113	64	3	1417	1529	267	373	1.42e-14	75.9
spo:SPAC212.11	sce:YNL112W	25.076	331	225	12	1204	1525	148	464	1.06e-12	70.9
spo:SPAC212.11	sce:YBR237W	24.403	377	237	17	1190	1532	276	638	1.68e-12	70.5
spo:SPAC212.11	sce:YOR204W	28.505	214	128	8	1352	1559	339	533	2.80e-12	69.7
spo:SPAC212.11	sce:YOR046C	28.205	234	140	8	1310	1530	227	445	1.65e-11	66.6
spo:SPAC212.11	sce:YDR243C	23.512	336	227	10	1208	1524	216	540	7.40e-11	65.1
spo:SPAC212.11	sce:YGL078C	22.689	357	239	11	1188	1527	130	466	7.62e-11	64.7
spo:SPAC212.11	sce:YPL119C	27.619	210	129	8	1352	1556	351	542	2.84e-10	63.2
spo:SPAC212.11	sce:YGL064C	22.195	410	216	16	1215	1541	166	555	8.25e-10	61.6
spo:SPAC212.11	sce:YDL084W	24.424	217	156	5	1308	1524	201	409	7.84e-07	51.6
spo:SPAC212.11	sce:YHR169W	27.273	143	93	4	1418	1555	257	393	1.09e-06	51.2
spo:SPAC212.11	sce:YLR276C	26.667	210	120	10	1189	1377	38	234	2.85e-06	50.1
spo:SPAC212.11	sce:YLR276C	34.146	41	27	0	1484	1524	386	426	0.12	35.0
spo:SPAC212.11	sce:YLL008W	21.965	346	233	13	1195	1524	258	582	4.71e-06	49.3
spo:SPAC212.11	sce:YPL082C	24.074	108	81	1	1417	1524	1648	1754	0.002	40.8
spo:SPAC212.11	sce:YGL070C	37.838	37	23	0	1583	1619	52	88	2.3	29.3
spo:SPBPB10D8.05C	sce:YPL092W	30.730	397	224	9	10	356	11	406	3.04e-47	165
spo:SPBPB10D8.05C	sce:YGL195W	31.034	58	37	2	140	195	2014	2070	0.37	30.8
spo:SPBPB10D8.05C	sce:YDR283C	51.613	31	14	1	221	250	589	619	0.66	30.0
spo:SPBPB10D8.05C	sce:YBR028C	35.135	37	24	0	229	265	127	163	0.84	29.6
spo:SPBPB10D8.05C	sce:YPL027W	27.957	93	54	3	138	221	53	141	3.1	27.3
spo:SPBPB10D8.05C	sce:YDR161W	28.571	49	35	0	318	366	102	150	3.2	27.7
spo:SPBPB10D8.05C	sce:YER166W	27.451	51	35	1	50	98	1273	1323	7.0	26.9
spo:SPBPB10D8.05C	sce:YGR040W	23.256	86	55	3	214	299	3	77	7.8	26.6
spo:SPBPB10D8.05C	sce:YAR019C	44.444	18	10	0	235	252	30	47	9.5	26.2
spo:SPBC359.02	sce:YOR368W	29.885	87	56	2	53	138	7	89	0.88	29.3
spo:SPCC330.07C	sce:YHR197W	26.761	71	39	3	65	130	287	349	2.0	28.9
spo:SPCC330.07C	sce:YGR224W	23.881	67	50	1	181	247	152	217	5.1	27.7

BLAST tabular 形式からトップヒットを抜き出す

- awkコマンドを用いて、BLAST tabular ファイルからクエリ配列ごとに、最大スコアをとるヒット（トップヒット）を出力する

```
awk 'prev!=$1{print; prev=$1}' sce-spo.blastout
```

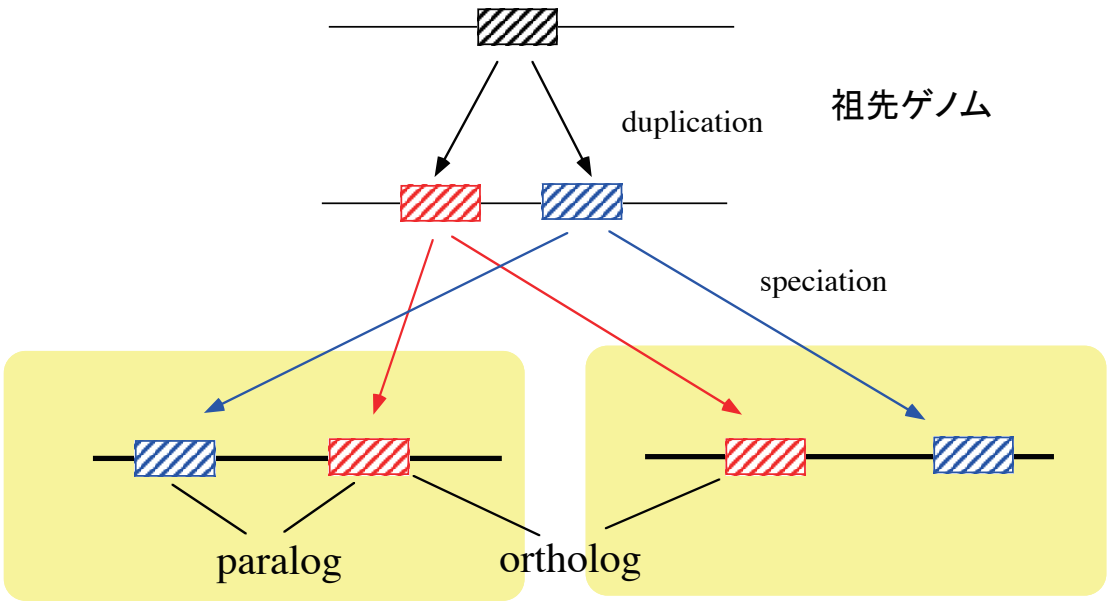
変数prevに一つ前の行のクエリ配列名（1カラム目）が入っている。
クエリ名が変わるごとに、その行を出力して、prevにそのクエリ名を代入する。

spo:SPAC212.11	sce:YMR190C	27.297	370	238	12	1169	1525	653	1004	7.08e-23	105
spo:SPAC212.11	sce:YDR021W	38.053	113	64	3	1417	1529	267	373	1.42e-14	75.9
spo:SPAC212.11	sce:YNL112W	25.076	331	225	12	1204	1525	148	464	1.06e-12	70.9
spo:SPAC212.11	sce:YBR237W	24.403	377	237	17	1190	1532	276	638	1.68e-12	70.5
spo:SPAC212.11	sce:YOR204W	28.505	214	128	8	1352	1559	339	533	2.80e-12	69.7
spo:SPAC212.11	sce:YOR046C	28.205	234	140	8	1310	1530	227	445	1.65e-11	66.6
spo:SPAC212.11	sce:YDR243C	23.512	336	227	10	1208	1524	216	540	7.40e-11	65.1
spo:SPAC212.11	sce:YGL078C	22.689	357	239	11	1188	1527	130	466	7.62e-11	64.7
spo:SPAC212.11	sce:YPL119C	27.619	210	129	8	1352	1556	351	542	2.84e-10	63.2
spo:SPAC212.11	sce:YGL070C	37.838	37	23	0	1583	1619	52	88	2.3	29.3
spo:SPBPB10D8.05C	sce:YPL092W	30.730	397	224	9	10	356	11	406	3.04e-47	165
spo:SPBPB10D8.05C	sce:YGL195W	31.034	58	37	2	140	195	2014	2070	0.37	30.8
spo:SPBPB10D8.05C	sce:YDR283C	51.613	31	14	1	221	250	589	619	0.66	30.0
spo:SPBPB10D8.05C	sce:YBR028C	35.135	37	24	0	229	265	127	163	0.84	29.6

（参考）E-value 0.001 以下、% identity 50% 以上のものを抜き出す

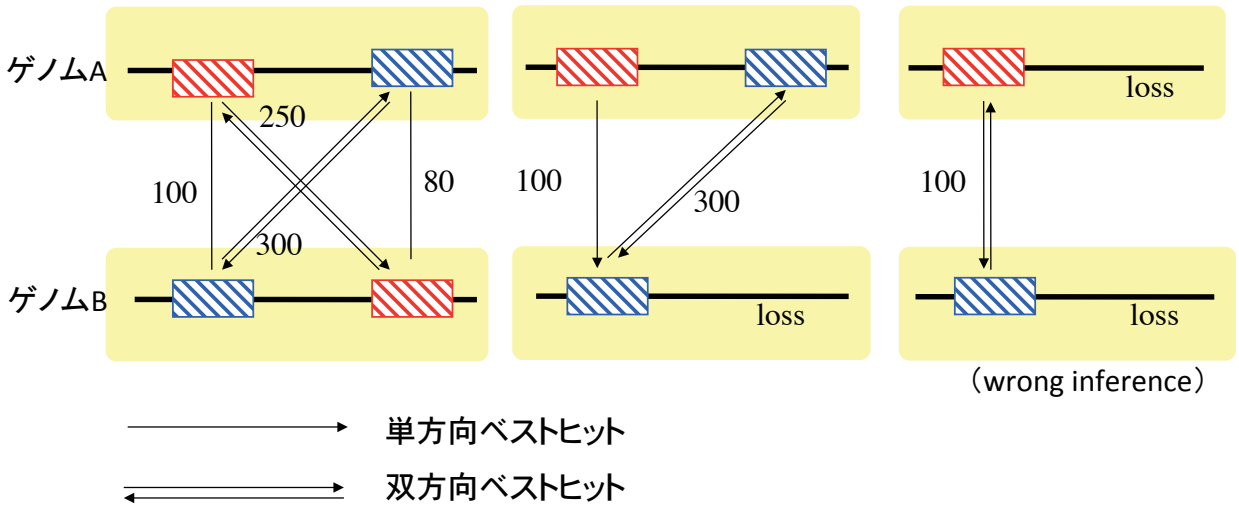
```
awk '$11<=0.001 && $3>=50 {print}'
```

オーソログとパラログ



オーソログの操作的定義

双方向ベストヒット (bi-directional best hit/reciprocal best hit)



双方向ベストヒットの検出

ゲノム1の遺伝子　ゲノム2の遺伝子　類似性スコア

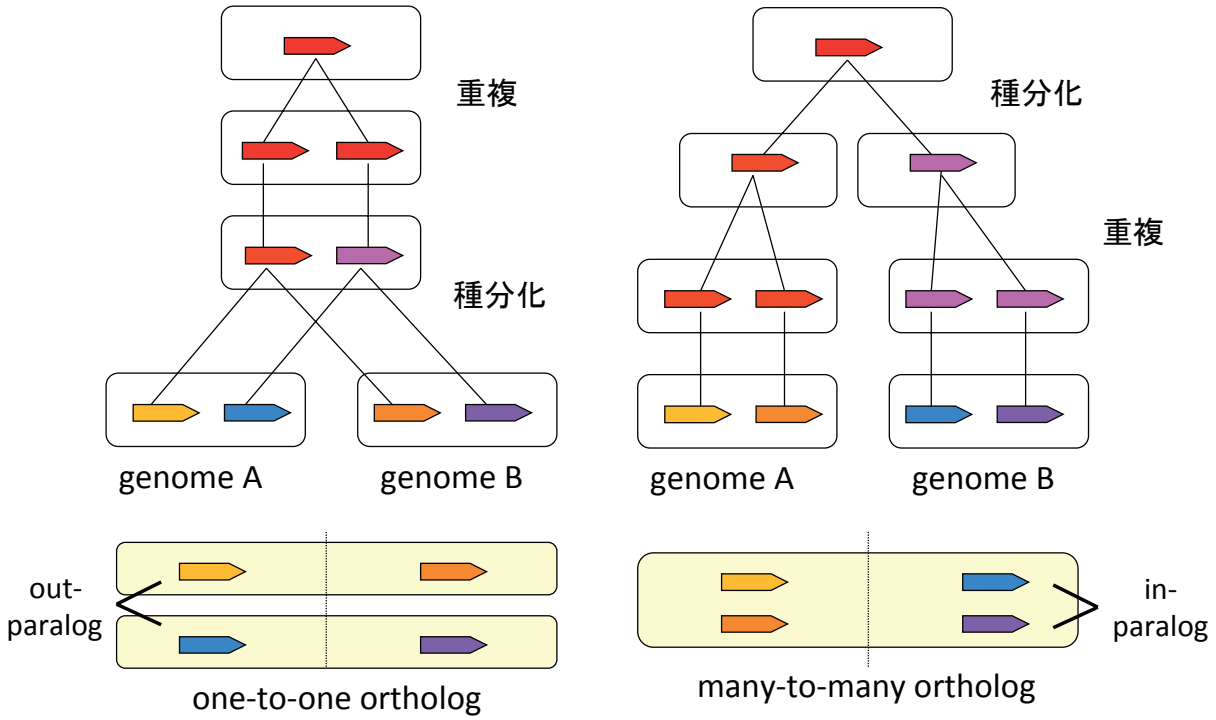
```
spo:SPAC4F8.12C   sce:YHR165C   3060
spo:SPAC22G7.06C  sce:YJL130C   2939
spo:SPAC56E4.04C  sce:YNR016C   2714
spo:SPAPB1E7.07   sce:YDL171C   2568
spo:SPBC216.07C   sce:YKL203C   2296
spo:SPBC216.07C   sce:YJR066W   2276
spo:SPAC4A8.11C   sce:YPL231W   2247
```

Lines はスコア順にソートされたファイルの行 のリスト

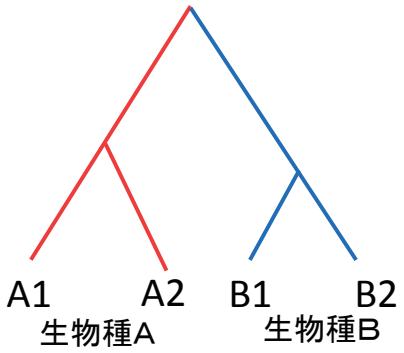
```
foreach line in Lines do
    (name1, name2, score) = split(line)
    Rank1[name1]++
    Rank2[name2]++
    if (Rank1[name1] == 1
            && Rank2[name2] == 1) then
        print  line
    fi
done
```

入力ファイル：　ゲノム間の総当りの
類似性スコアのリスト。
スコアの大きい順にソートされているとする。

種内パラログ in-paralog が存在する場合



多対多の関係を考慮した拡張



A1-B1, A1-B2, A2-B1, A2-B2は
類似度がほぼ同じ
→いずれもオーソログの関係

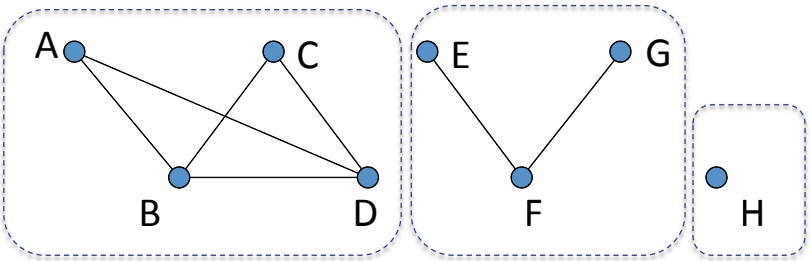
RATIO = 0.9; #類似度を同じと見なす許容範囲
Lines はスコア順にソートされたファイルの行のリスト

```
foreach line in Lines do
    (name1, name2, score) = split(line)

    if (Best1[name1]が未定義) then
        Best1[name1] = score
    fi
    if (Best2[name2]が未定義) then
        Best2[name2] = score
    fi

    if (score >= Best1[name1] * RATIO
        && score >= Best2[name2] * RATIO) then
        print line
    fi
done
```

実習：出芽酵母と分裂酵母のオーソログ解析

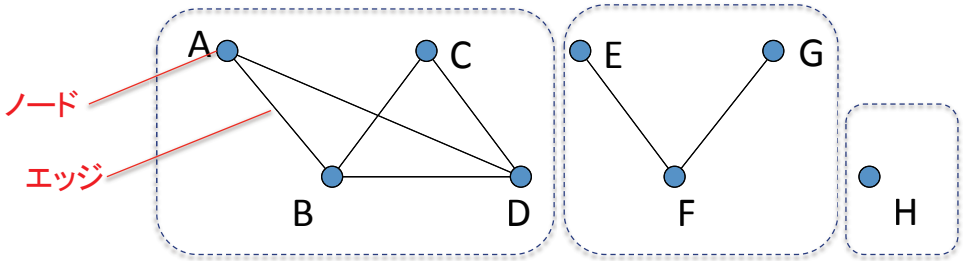
```
# bit-score の順にソートする
% sort -k 12,12nr > sce-spo.blast.sorted

# 双方向ベストヒットをとる
% bbh.pl sce-spo.blast.sorted > sce-spo.bbh

# 双方向ベストヒットをとる（条件を緩めたバージョン）
% bbh2.pl sce-spo.blast.sorted > sce-spo.bbh2
```

単連結クラスタリング



- 関係で結ばれた遺伝子対をすべてつなぐ
- 一般に、ホモロジー検索結果の整理に有効
  - 相同関係の推移性に基づく
    「A と B が相同で B と C も相同なら、A と C も相同である」
  - ホモロジー検索では、類似性が低い相同関係を取りこぼす可能性がある
    →単連結クラスタリングは検索のとりこぼしを補ってくれる
- アルゴリズムはシンプル（グラフの連結成分connected componentをとる）

2項関係のグラフによる表現



A	B
A	D
B	C
B	D
E	F
F	G

入力ファイル： 関連を持つ遺伝子対のリスト

2つのノードがエッジでつながっていることを2次元ハッシュを用いて表す

Link["A"]["B"] = *Link*["B"]["A"] = 1
Link["A"]["D"] = *Link*["D"]["A"] = 1

keys(*Link*["A"]) (ハッシュ *Link*["A"] におけるキーの集合)
== ノード "A"とつながっているノードの集合
→ "B"と"D"

単連結クラスタリング

```
#データを読み込んでグラフを構築
foreach line in Lines do
    (node1, node2) = split(line)
    # node1とnode2がつながっていることを2次元ハッシュで表す
    Link[node1][node2] = Link[node2][node1] = 1;
done

nodeSet = keys(Link)
foreach node in nodeSet do
    if (Mark[node]==0) then
        Cluster（配列）を空にする
        Traverse(node)
        Cluster を出力する
    fi
done
```

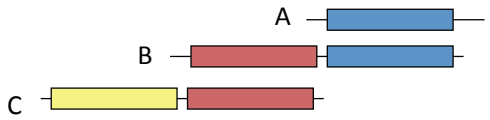
```
# サブルーチン Traverse
# node1につながるノードを再帰的に
# たどって Cluster に加える
Traverse (node1) {
    if (Mark[node1] > 0) then
        # マークされたノードはスキップ
            return
    fi

    Cluster に node1 を加える
    Mark[node1] = 1  #出力済みマーク
    nodeSet = keys(Link[node1])
    foreach node2 in nodeSet do
        Traverse(node2);
    done
}
```

```
RPOB SA0500 4628
RPOC SA0501 4507
POLC SA1107 4390
NARG SA2185 3997
GLTA SA0430 3898
PYCA SA0963 3830
PYRAB SA1046 3712
UVRA SA0714 3397
VALS SA1488 3350
```

入力ファイル： 類似性スコアのリスト。ソートされている必要なし。

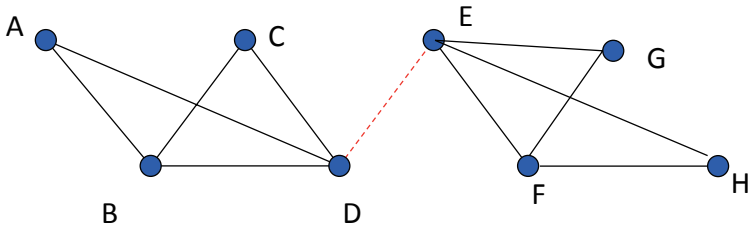
単連結クラスタリングの問題点

- マルチドメイン蛋白質の場合、推移律が満たされないことがある



→アライメントのカバレッジを上げる

- ひとつでも間違った関係があると、分類を大きく間違える可能性がある



→類似性スコアの閾値を上げる

実習：オーソログ結果のクラスタリング

```
# 双方向ベストヒット（条件を緩めたバージョン）のクラスタリング
% slink.pl sce-spo.bbh2 > sce-spo.oclust

# タイトルをつける
% add_title.pl –fasta_files=sce,spo sce-spo.oclust >
    sce-spo.oclust_title
```

類似性に関する指標

1. bit score
2. E-value　　統計的評価
3. percent identity
4. percent positive score (ppos)
5. score/length
6. query coverage ((qend-qstart+1)/qlen)
7. subject coverage ((send-sstart+1)/slen)

長さ当たりの類似性
→進化距離を反映

全長が
マッチ
するか?

4,6,7は –outfmt 6 で追加のカラム指定が必要
例） `-outfmt "6 std qlen slen"`

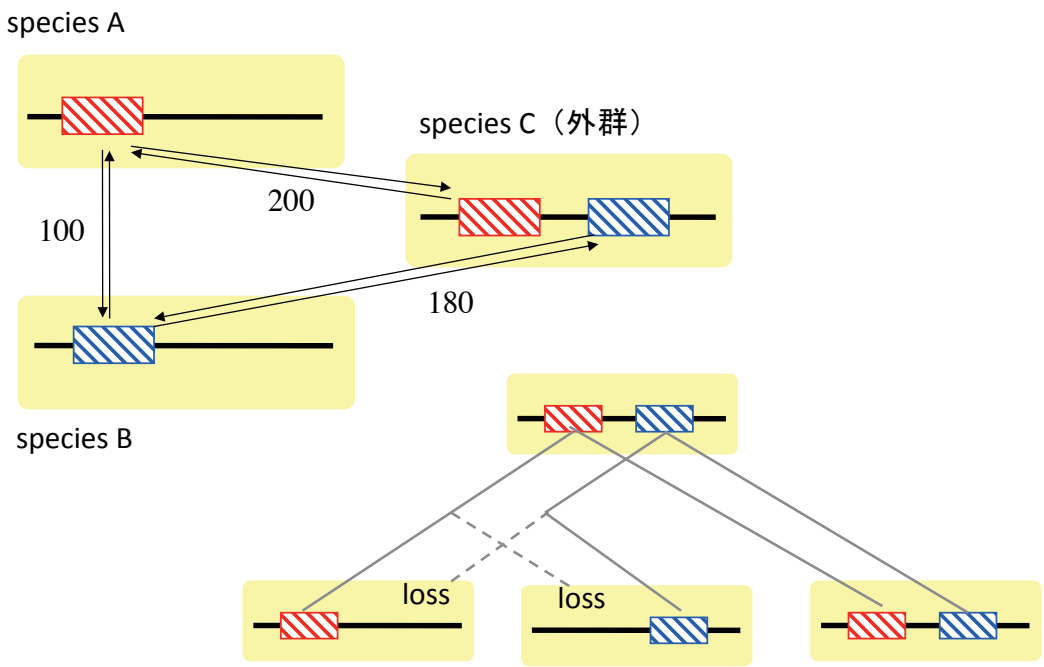
多対多オーソログをより正確にとるには

1. 種間比較だけでなく、種内比較の結果も考慮する
   – specA-specB に加えて、specA-specA、specB-specBの比較も行う（→2つのファイルを連結して自分自身に対して相同性検索を行う）

```
% cat sce spo > sce+spo
% blastp –db sce+spo –query sce+spo
```

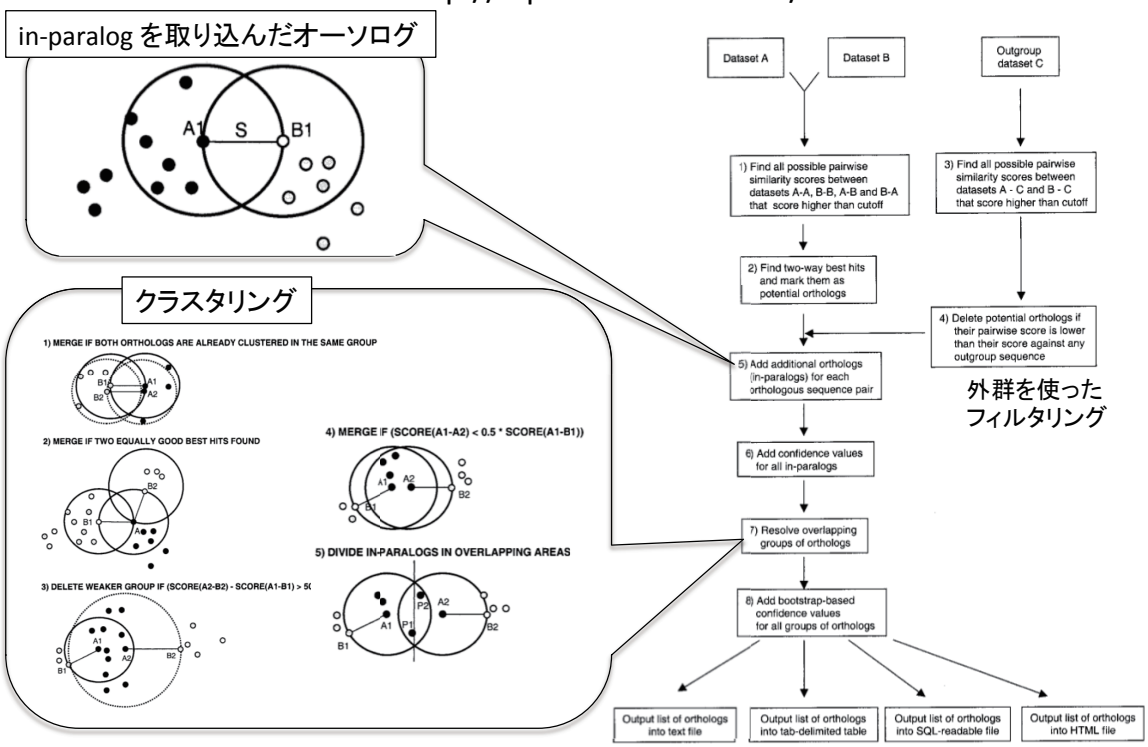
2. オーソログの同定基準やクラスタリング手順を工夫する

外群を加えたオーソログ解析



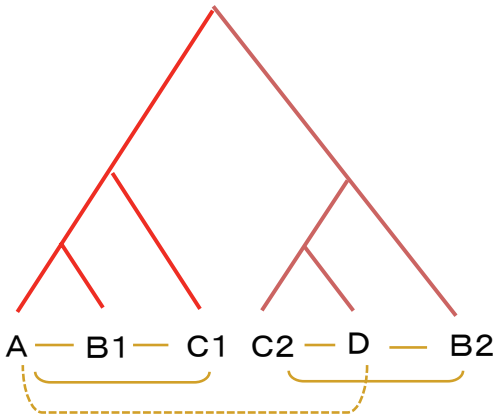
Inparanoid

http://inparanoid.sbc.su.se/

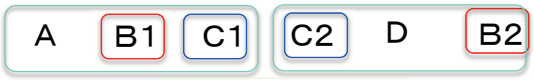


3種以上のオーソログ解析

遺伝子系統樹

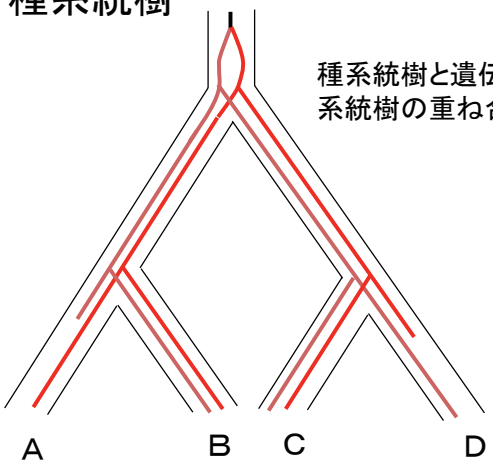


双方向ベストヒットのクラスタリング



種の重複度チェック

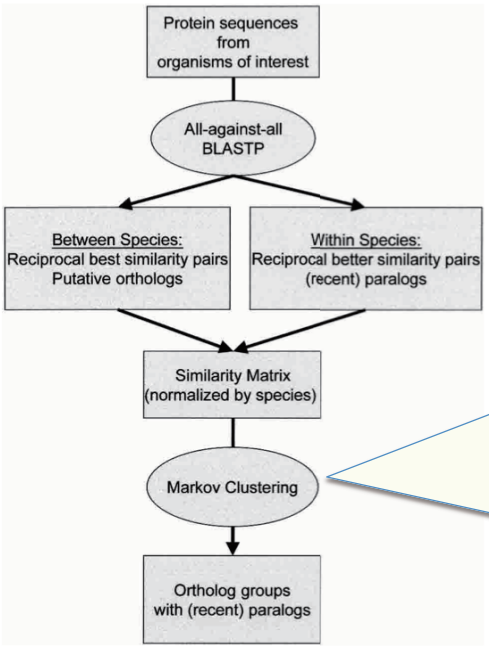
種系統樹

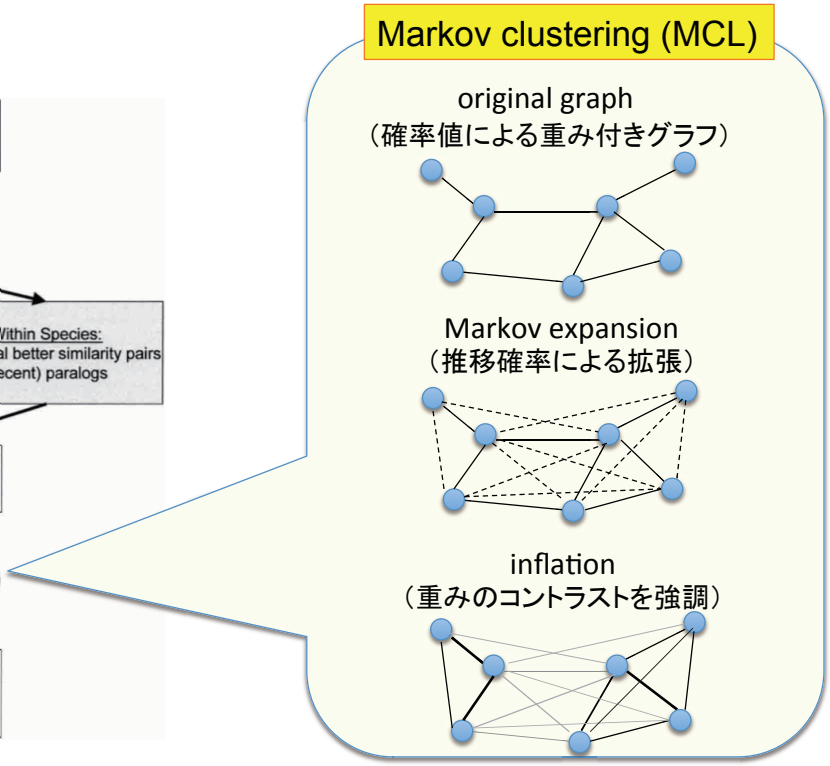


種系統樹と遺伝子系統樹の重ね合わせ

OrthoMCL

http://orthomcl.org/





DomClust

http://mbgd.genome.ad.jp/domclust/

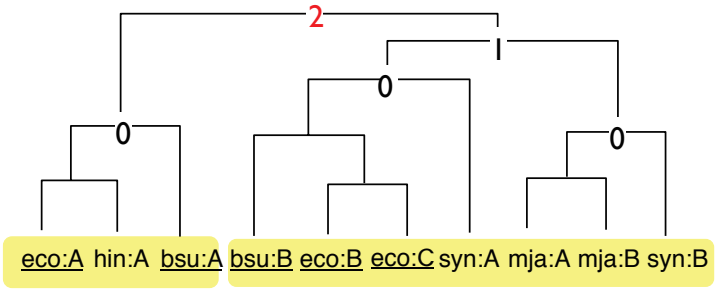
総当たりの
ホモロジー検索

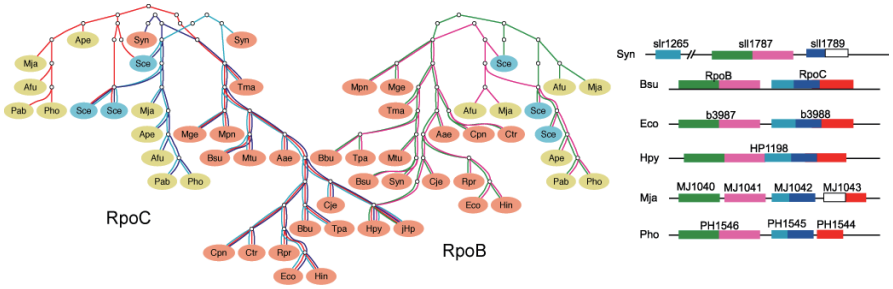
↓

階層的クラスタリング
（DomClust）

↓

ドメイン単位の
オーソログループ



Cut if $\frac{|Spec(A) \cap Spec(B)|}{\min(|Spec(A)|, |Spec(B)|)} > p$

（種の重複度チェック）