

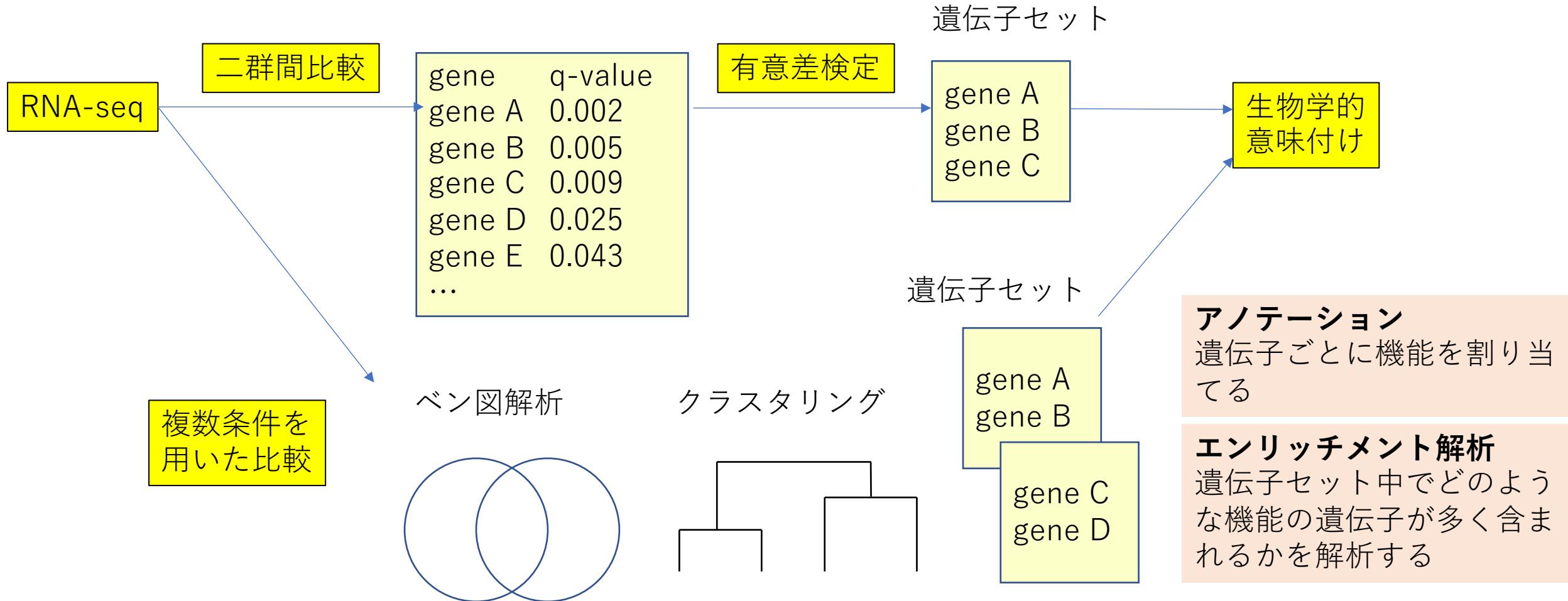
# 機能アノテーションと GO解析

基礎生物学研究所

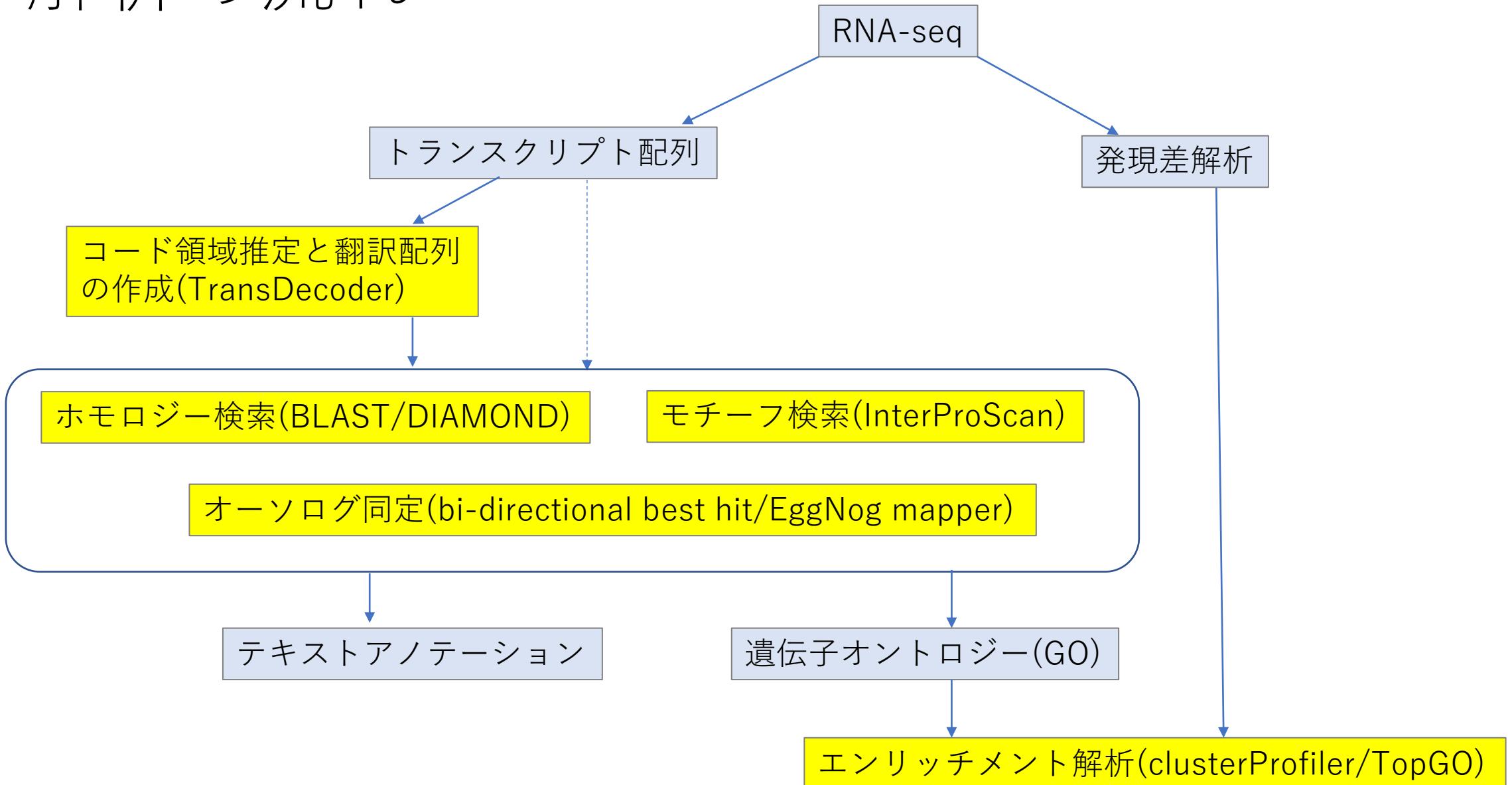
情報管理解析室

内山 郁夫

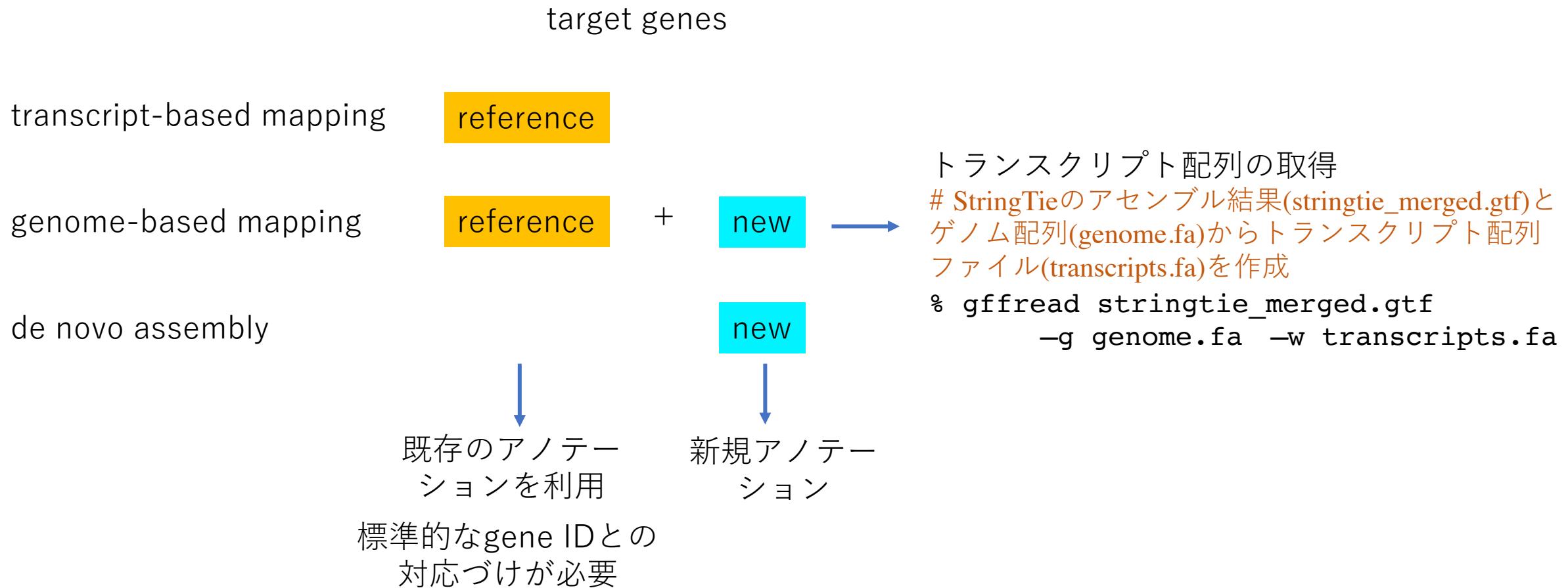
# RNA-seq解析結果の解釈



# 解析の流れ



# 遺伝子アノテーション 基本戦略



# 遺伝子アノテーション

- 遺伝子構造の予測
  - 通常の遺伝子構造予測では、インtron-エクソン構造の予測が必要だが、RNA-seq解析では、すでにmRNA配列が得られているので、これは不要。ただし、配列中のタンパク質をコードする領域(CDS)を予測する必要はある。
  - blastxなどのツールを用いると、CDS予測をせずに、全読枠を翻訳してホモロジー検索を実行することも可能。ただし時間がかかる。
- 遺伝子機能の予測
  - ホモロジー検索、もしくはモチーフ検索を行い、ヒットした類似配列もしくはモチーフの機能に基づいて機能を推定する。

# TransDecoder

- 転写配列中のコード領域を予測するツール。もともとTrinityに付属のツールとして開発されたが、現在は独立のツールとして公開されている。

Home Brian Haas edited this page on 3 May · 12 revisions Edit New Page

## TransDecoder (Find Coding Regions Within Transcripts)

TransDecoder identifies candidate coding regions within transcript sequences, such as those generated by de novo RNA-Seq transcript assembly using Trinity, or constructed based on RNA-Seq alignments to the genome using Tophat and Cufflinks.

TransDecoder identifies likely coding sequences based on the following criteria:

- a minimum length open reading frame (ORF) is found in a transcript sequence
- a log-likelihood score similar to what is computed by the [GeneID](#) software is > 0.
- the above coding score is greatest when the ORF is scored in the 1st reading frame as compared to scores in the other 2 forward reading frames.
- if a candidate ORF is found fully encapsulated by the coordinates of another candidate ORF, the longer one is reported. However, a single transcript can report multiple ORFs (allowing for operons, chimeras, etc).
- a PSSM is built/trained/used to refine the start codon prediction.
- optional the putative peptide has a match to a Pfam domain above the noise cutoff score.

### Obtaining TransDecoder

The latest release of TransDecoder can be found [here](#).

### Running TransDecoder

Note, TransDecoder is now available on [Galaxy](#) [usegalaxy.eu](#)

#### Predicting coding regions from a transcript fasta file

The 'TransDecoder' utility is run on a fasta file containing the target transcript sequences. The simplest usage is as follows:

Step 1: extract the long open reading frames

```
TransDecoder.LongOrfs -t target_transcripts.fasta
```

Pages 1 Find a Page...  
Home  
TransDecoder (Find Coding Regions Within Transcripts)  
Obtaining TransDecoder  
Running TransDecoder  
Predicting coding regions from a transcript fasta file  
Step 1: extract the long open reading frames  
Step 2: (optional)  
Step 3: predict the likely coding regions  
Starting from a genome-based transcript structure GTF file (e.g. cufflinks or stringtie)  
Sample data and execution  
Output files explained  
Including homology searches as ORF retention criteria  
BlastP Search  
Pfam Search  
Integrating the Blast and Pfam search results into coding region selection  
Viewing the ORF predictions in a genome browser  
Viewing ORFs on target transcripts  
Viewing ORFs in the context of the transcript structures on the genome  
Technical Support and Project Announcements

# TransDecoderの実行

クエリ配列： seqfile.fa

- 長いORFを抽出する( $\geq 100$  aa)

% TransDecoder.LongOrfs -t seqfile.fa

- 長さ上位500のORFを使って、コード領域の確率モデルパラメータを推定し、それを用いてコード領域を予測する

% TransDecoder.Predicts -t seqfile.fa

出力ファイル： seqfile.fa.transdecoder.???

pep: アミノ酸配列、 cds: 塩基配列、 gff3: GFF形式の遺伝子座標

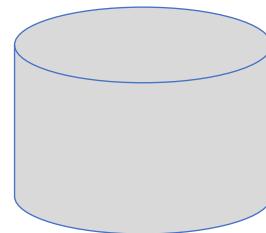
# ホモロジーに基づく機能推定

## ホモロジー検索

問い合わせ

Gene X

タンパク質配列DB



類似配列検索

Gene A: 機能

Gene B: 機能

Gene C: 機能

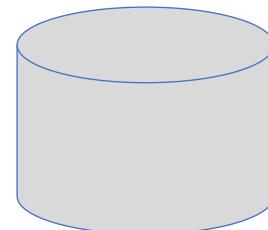
機能  
アノテーション

## モチーフ／プロファイル検索

問い合わせ

Gene X

遺伝子ファミリー  
(ドメイン)DB



類似ファミリー  
検索

ファミリーA  
機能

Gene A Gene C  
Gene B Gene D

機能  
アノテーション

# ホモロジー検索とアノテーション

ヘモグロビン  
β鎖

||  
オーソログ

他の  
グロビン族  
(パラログ)

非相同  
蛋白質

Sequences producing significant alignments:		
	Score (bits)	E Val
sp:HBB_HUMAN HEMOGLOBIN BETA CHAIN.	306	1e-83
sp:HBB_GORGO HEMOGLOBIN BETA CHAIN.	305	3e-83
sp:HBB2_PANLE HEMOGLOBIN BETA-2 CHAIN.	302	2e-82
sp:HBB_HYLLA HEMOGLOBIN BETA CHAIN.	300	6e-82
sp:HBB_PREEN HEMOGLOBIN BETA CHAIN.	298	4e-81
sp:HBB_COLPO HEMOGLOBIN BETA CHAIN.	295	2e-80
sp:HBB_CERAE HEMOGLOBIN BETA CHAIN.	295	3e-80
sp:HBB_MACFU HEMOGLOBIN BETA CHAIN.	293	1e-79
sp:HBB_COLBA HEMOGLOBIN BETA CHAIN.	293	1e-79
sp:MYG_BALAC MYOGLOBIN.	49	5e-06
sp:MYG_MEGNO MYOGLOBIN.	48	8e-06
sp:MYG_ESCGI MYOGLOBIN.	48	1e-05
sp:MYG_BALPH MYOGLOBIN.	47	2e-05
sp:MYG_ZIPCA MYOGLOBIN.	46	4e-05
sp:GLB1_ARTSX GLOBIN E1, EXTRACELLULAR.	45	9e-05
sp:GLP2_GLYDI GLOBIN, POLYMERIC COMPONENT P2.	42	6e-04
sp:GLP1_GLYDI GLOBIN, MAJOR POLYMERIC COMPONENT P1.	41	8e-04
sp:HBAZ_MACEU HEMOGLOBIN ZETA CHAIN (FRAGMENTS).	39	0.005
sp:GLP3_GLYDI GLOBIN, POLYMERIC COMPONENT P3.	38	0.009
sp:LGB2_PEA LEGHEMOGLOBIN II.	36	0.035
sp:LGB1_PEA LEGHEMOGLOBIN I.	35	0.079
sp:LGB2_SESRO LEGHEMOGLOBIN 2.	34	0.18
sp:HBP_CANLI LEGHEMOGLOBIN.	32	0.40
sp:LGB1_VICFA LEGHEMOGLOBIN I.	32	0.53
sp:LACG_LACCA 6-PHOSPHO-BETA-GALACTOSIDASE (EC 3.2.1.85) (BETA-...)	32	0.53
sp:LGB3_SESRO LEGHEMOGLOBIN 3.	31	0.90
sp:LGBA_PHAVU LEGHEMOGLOBIN A.	31	0.90
sp:HMPA_BACSU FLAVOHEMOPROTEIN (HAEMOGLOBIN-LIKE PROTEIN) (FLAV...	31	1.2

ベストヒット  
E-valueが低い

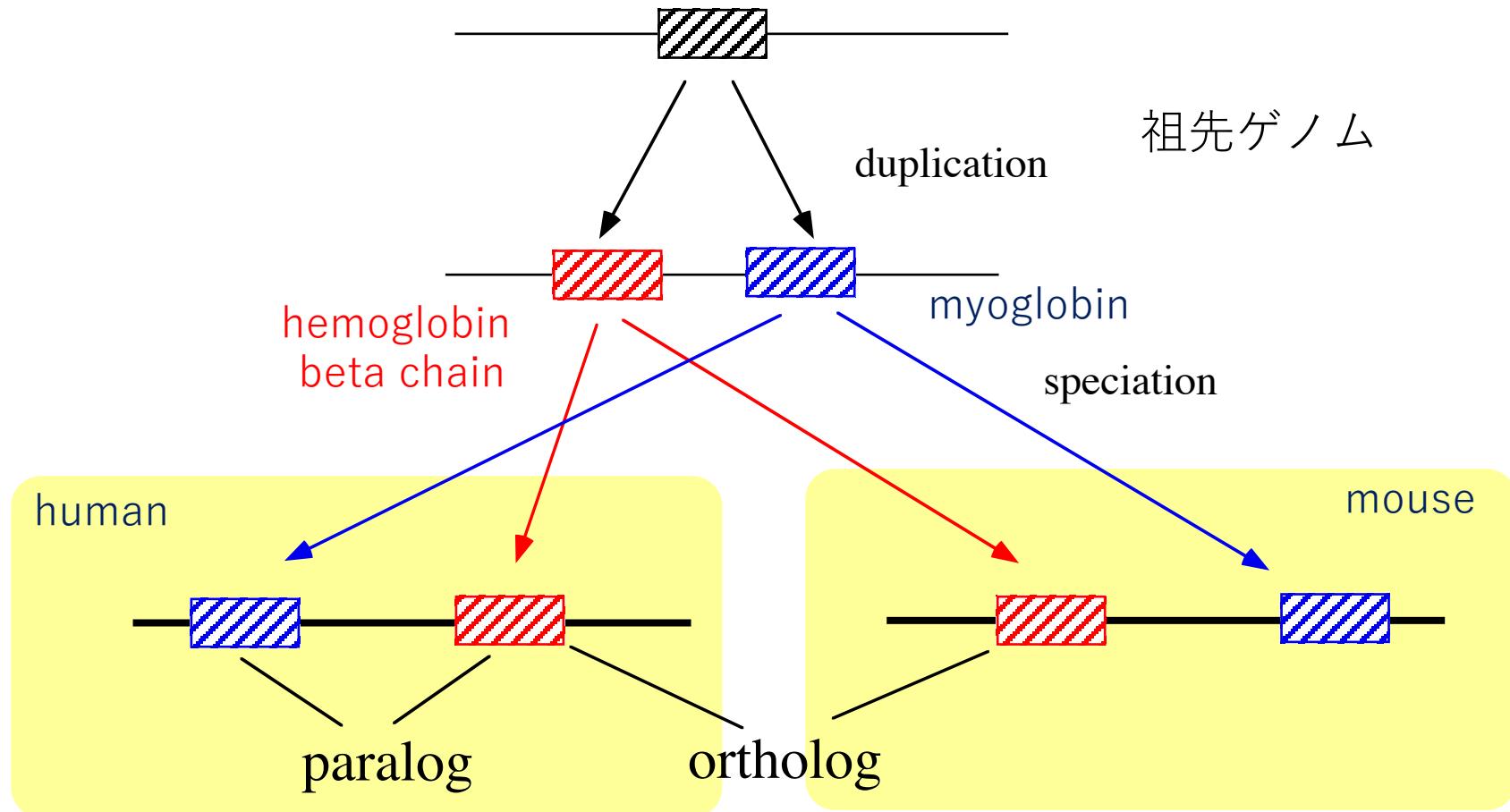
E-value  
非相同配列を  
間違って拾って  
しまう個数の  
期待値

統計的に  
有意な類似性

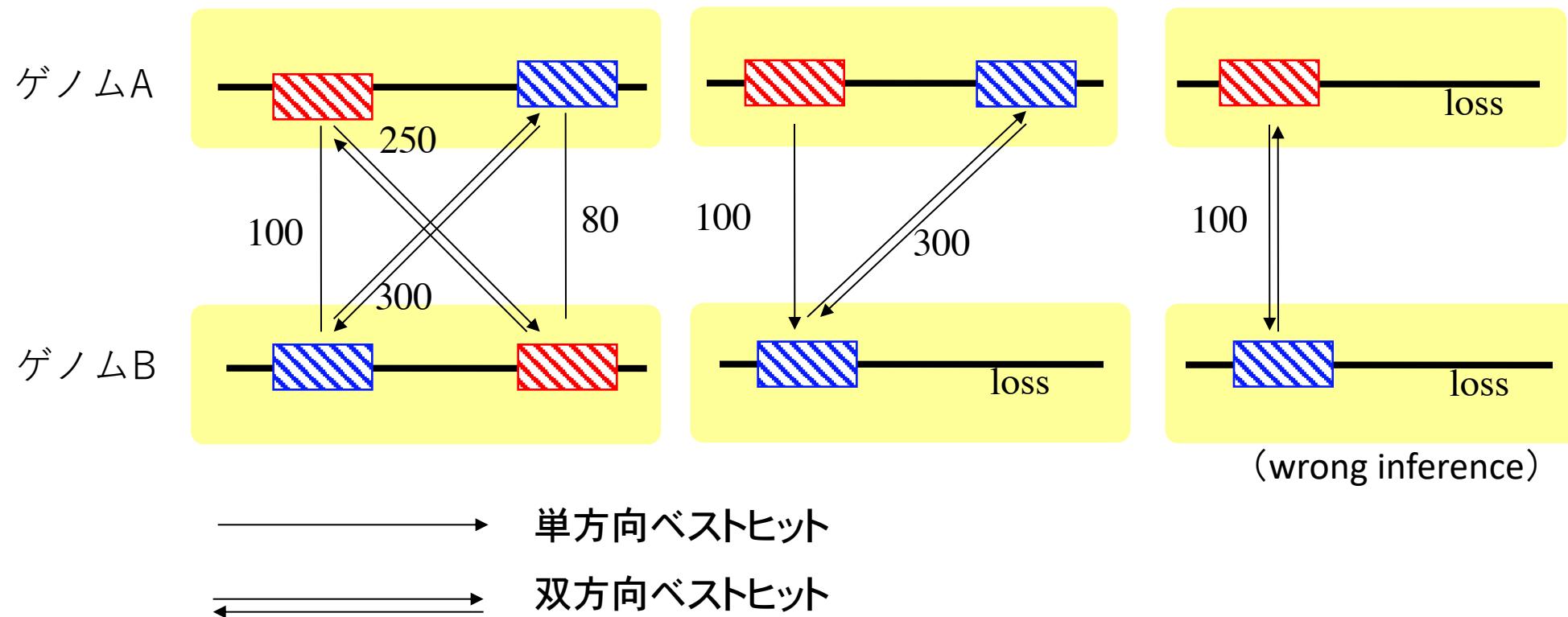
微妙な類似性

統計的に  
有意でない

# オーソログとパラログ



# 双方向ベストヒットによるオーソログの推定



# BLAST 検索の実行

```
# 配列ファイルdb.faに検索用のインデックスを作成。データベース名 dbnameで出力。
% makeblastdb -in db.fa -dbtype prot -parse_seqids -out dbname
# 作成したデータベースに対して、query.faをクエリとした検索の実行。
# タブ区切り形式 (outfmt 6)で、標準の形式にタイトル行を付加して、上位10ヒットを出力。
% blastp -query query.fa -db dbname -evalue 0.001
    -outfmt "6 std stitle" -max_target_seqs 10 -num_threads 8 > blastout.tab
```

1. query	2. subject (database)	3. %identity	5. mismatch		6. gap_open		9. s_start		10. s_end		12. bit-score		13. title
			4. align-len		7. q_start	8. q_end			11. eval				
DI49_1142	NP_009669.1	93.421	228	15	0	8	235	4	231	9.83e-162	444	ADP-ribose diphosphatase	
DI49_2764	NP_012252.1	88.604	1404	156	3	1	1401	1	1403	0.0	2560	ATP-binding cassette multid	
DI49_2764	NP_014654.1	67.521	1404	426	8	1	1393	1	1385	0.0	1943	ATP-binding cassette sterol	
DI49_2764	NP_014468.3	37.097	1364	721	26	51	1356	48	1332	0.0	873	ATP-binding cassette multid	
DI49_2764	NP_014468.3	22.911	694	426	27	762	1402	43	680	2.40e-27	119	ATP-binding cassette multid	
DI49_2764	NP_014468.3	20.819	562	377	20	111	639	807	1333	3.47e-17	85.9	ATP-binding cassette multid	
MSTRG.1000.1	NP_013901.1	68.702	1425	423	7	1	1416	1	1411	0.0	2005	Ecm5p [Saccharomyces cerevi	
MSTRG.1000.1	NP_012653.1	34.177	79	51	1	485	562	384	462	5.68e-05	45.4	histone demethylase [Saccha	
MSTRG.1000.1	NP_012653.1	27.523	109	73	3	637	743	487	591	5.54e-04	42.4	histone demethylase [Saccha	
MSTRG.1000.3	NP_013901.1	68.702	1425	423	7	1	1416	1	1411	0.0	2005	Ecm5p [Saccharomyces cerevi	
MSTRG.1000.3	NP_012653.1	34.177	79	51	1	485	562	384	462	5.68e-05	45.4	histone demethylase [Saccha	
MSTRG.1000.3	NP_012653.1	27.523	109	73	3	637	743	487	591	5.54e-04	42.4	histone demethylase [Saccha	

# DIAMOND 超高速ホモジ一検索

# インデックスの作成

```
% diamond makedb --in db.fa --db db
```

# 検索の実行

```
% diamond blastp --query query.fa --db db  
--eval 0.001 --max-target-seqs 10  
--outfmt 6 qseqid sseqid pident eval bitscore stitle  
--threads 4 --out diamondout.tab
```

1. query (database)	2. subject	3. %identity	4. eval	5. bit-score	6. title
------------------------	------------	--------------	---------	--------------	----------

DI49_1142	NP_009669.1	93.4	2.5e-122	433.3	NP_009669.1 ADP-ribose diphosphatase [Saccharomyces cerevisiae S288C]
DI49_2764	NP_012252.1	88.7	0.0e+00	2499.2	NP_012252.1 ATP-binding cassette multidrug transporter PDR11 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_014654.1	67.9	0.0e+00	1876.3	NP_014654.1 ATP-binding cassette sterol transporter AUS1 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_014468.3	37.4	1.1e-236	815.8	NP_014468.3 ATP-binding cassette multidrug transporter PDR18 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_010294.1	35.5	6.0e-232	800.0	NP_010294.1 ATP-binding cassette transporter SNQ2 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_015267.1	33.4	8.2e-205	709.9	NP_015267.1 ATP-binding cassette multidrug transporter PDR12 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_014796.3	31.8	8.2e-181	630.2	NP_014796.3 ATP-binding cassette multidrug transporter PDR5 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_014973.1	31.9	2.4e-180	628.6	NP_014973.1 ATP-binding cassette multidrug transporter PDR10 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_010694.1	31.0	1.6e-168	589.3	NP_010694.1 ATP-binding cassette multidrug transporter PDR15 [Saccharomyces cerevisiae S288C]
DI49_2764	NP_014567.2	23.5	2.0e-49	193.7	NP_014567.2 uncharacterized protein YOL075C [Saccharomyces cerevisiae S288C]

# BLAST結果からベストヒットを抽出する

```
% sort -k 1,1 -u -s blastout.tab > blasttop.tab
```

第1フィールド（クエリ配列）をキーとしてソートし（-k 1,1）、キーが重複した場合は最初の行のみを出力（-u ユニーク）。元の並びがクエリ配列ごとにE-value(スコア)の順になっており、-s（安定ソート）を指定することで、その順を維持したままソート・ユニークが実行されるため、ベストヒット1つのみが出力される。

ソートキー（クエリ）										E-value	Score	
DI49_1142	NP_009669.1	93.421	228	15	0	8	235	4	231	9.83e-162	444	ADP-ribose diphosphatase
DI49_2764	NP_012252.1	88.604	1404	156	3	1	1401	1	1403	0.0	2560	ATP-binding cassette multid
DI49_2764	NP_014654.1	67.521	1404	426	8	1	1393	1	1385	0.0	1943	ATP-binding cassette sterol
DI49_2764	NP_014468.3	37.097	1364	721	26	51	1356	48	1332	0.0	873	ATP-binding cassette multid
DI49_2764	NP_014468.3	22.911	694	426	27	762	1402	43	680	2.40e-27	119	ATP-binding cassette multid
DI49_2764	NP_014468.3	20.819	562	377	20	111	639	807	1333	3.47e-17	85.9	ATP-binding cassette multid
MSTRG.1000.1	NP_013901.1	68.702	1425	423	7	1	1416	1	1411	0.0	2005	Ecm5p [Saccharomyces cerevi
MSTRG.1000.1	NP_012653.1	34.177	79	51	1	485	562	384	462	5.68e-05	45.4	histone demethylase [Saccha
MSTRG.1000.1	NP_012653.1	27.523	109	73	3	637	743	487	591	5.54e-04	42.4	histone demethylase [Saccha
MSTRG.1000.3	NP_013901.1	68.702	1425	423	7	1	1416	1	1411	0.0	2005	Ecm5p [Saccharomyces cerevi
MSTRG.1000.3	NP_012653.1	34.177	79	51	1	485	562	384	462	5.68e-05	45.4	histone demethylase [Saccha
MSTRG.1000.3	NP_012653.1	27.523	109	73	3	637	743	487	591	5.54e-04	42.4	histone demethylase [Saccha

# 双方向ベストヒットの抽出

# 各クエリ配列に対するベストヒットの出力（前出）

```
% sort -k 1,1 -u -s blastout.tab > blast_top.tab
```

# 逆方向（データベース配列）から見たベストヒットの出力

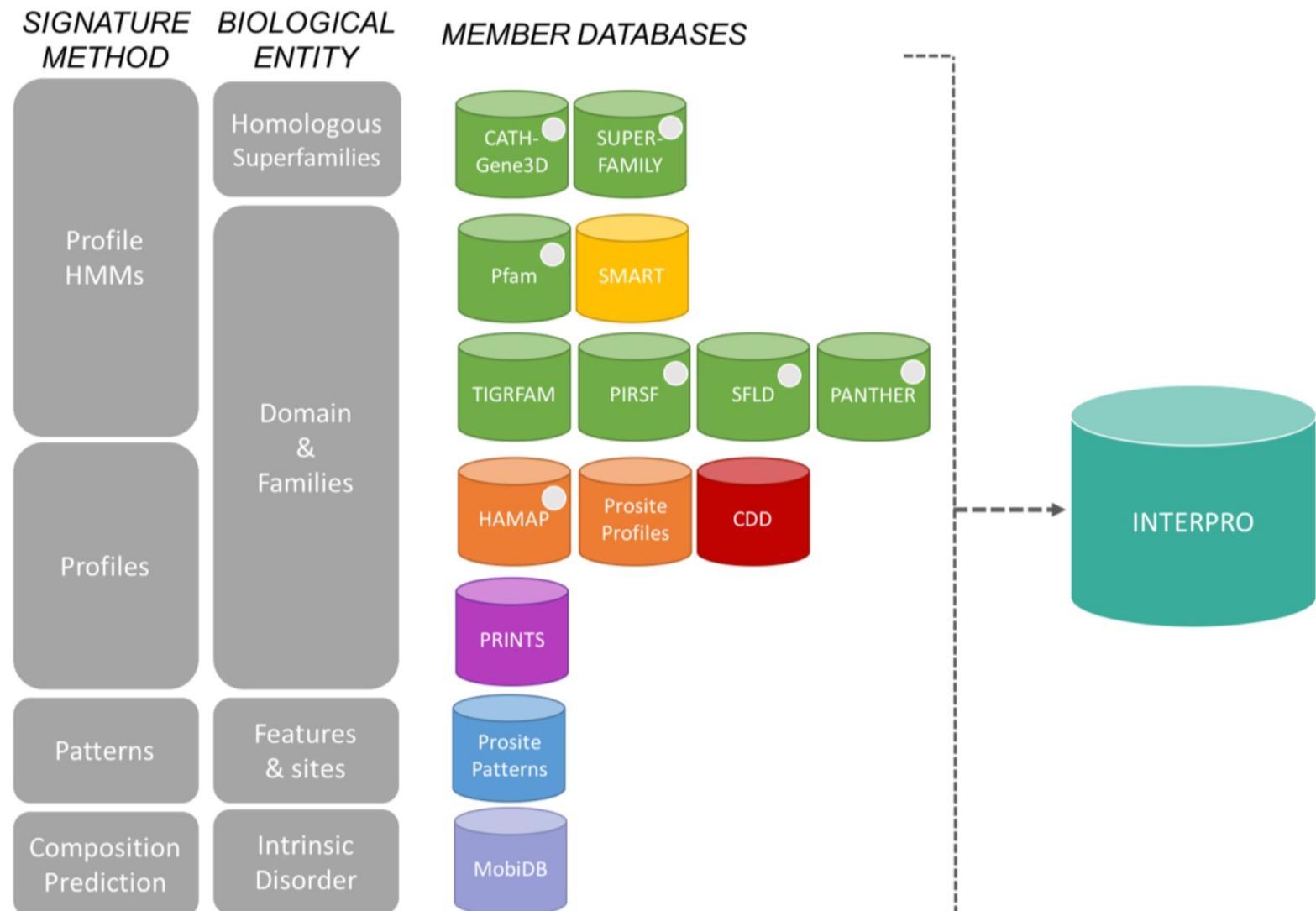
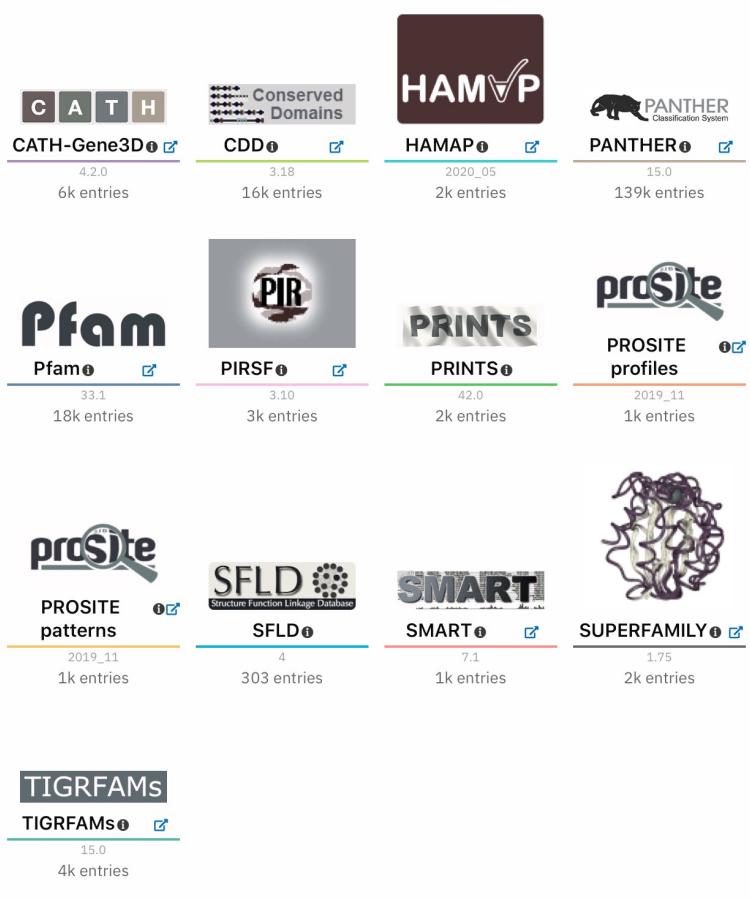
```
% sort -k 2,2 -k 11,11g -k 12,12nr blastout.tab | sort -k 2,2 -u -s  
        > blast_top_rev.tab
```

# 双方向ベストヒット（オーソログの推定）の出力

```
% sort blast_top.tab blast_top_rev.tab | uniq -d > blast_bbh.tab
```

（両方向のベストヒットを合わせてソートし、重複した行を出力する(`uniq -d`)。どちらの方向からみたベストヒットにも含まれるペアが出力される）

# InterProScan モチーフ／ドメイン検索



# InterProScan の実行

```
# query.faをクエリとして、InterProのデータベース全てを対象として検索。  
# 検索結果にはGO termも含める。
```

```
% interproscan.sh -i query.fa -b iprout -goterms --cpu 4
```

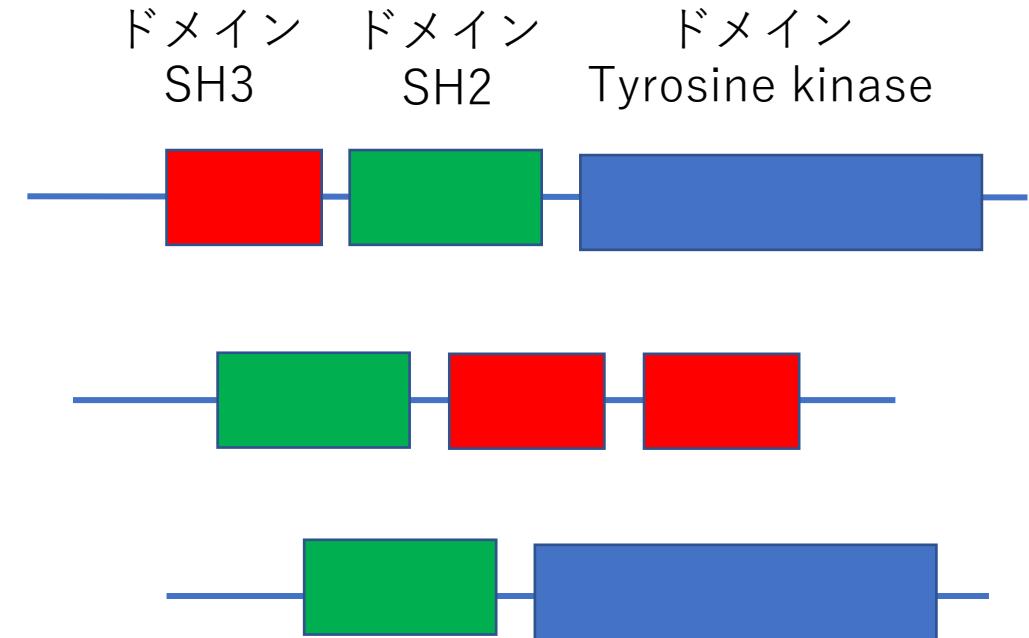
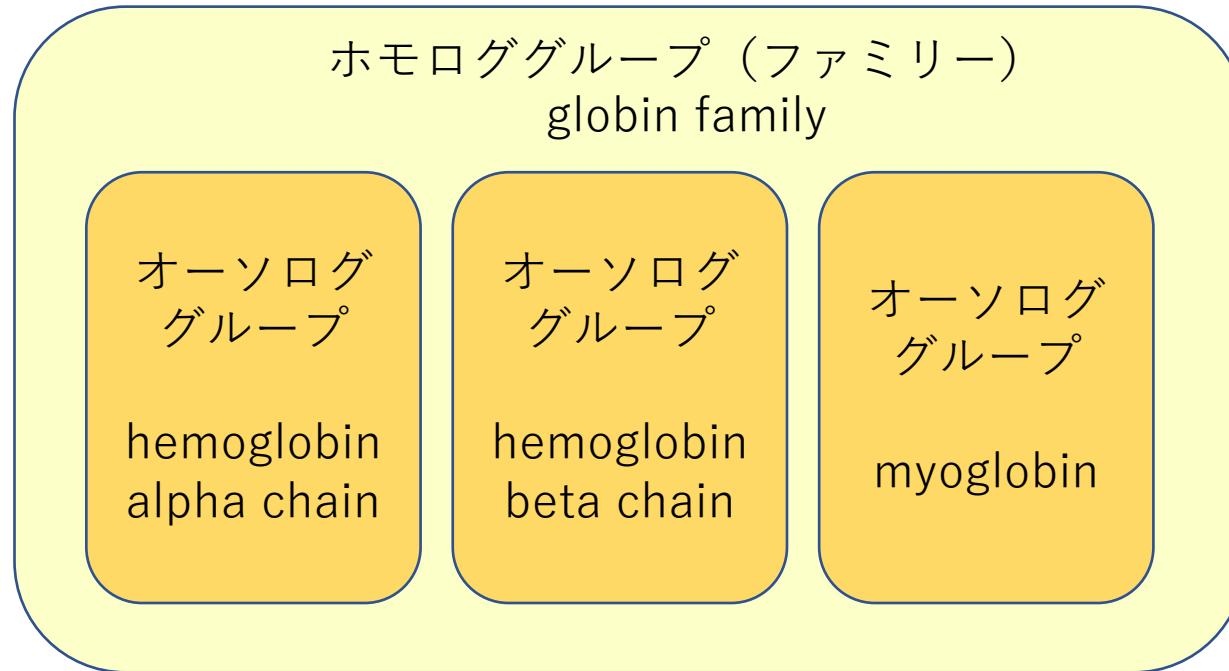
```
XM_018368617.1 b190671767a6618abe8a5898271389c4 298 Pfam PF01680 SOR/SNZ family 4 211 3.5E-103 ..  
XM_018368617.1 b190671767a6618abe8a5898271389c4 298 ProSiteProfiles PS51129 PdxS/SNZ family profile. 6 298 150 ...  
XM_018368617.1 b190671767a6618abe8a5898271389c4 298 Gene3D G3DSA:3.20.20.70 1 298 7.8E-137 ..  
XM_018368617.1 b190671767a6618abe8a5898271389c4 298 TIGRFAM TIGR00343 TIGR00343: pyridoxal 5'-phosphate synthase..  
XM_018368617.1 b190671767a6618abe8a5898271389c4 298 PIRSF PIRSF029271 1 298 1.9E-143 ..
```

```
# 検索結果からGO termの抽出
```

```
% cut -f1,14 seub_genes6.iprscan.tsv
```

```
XM_018368617.1  
XM_018368617.1 GO:0042819|GO:0042823  
XM_018368617.1 GO:0003824  
XM_018368617.1 GO:0042819|GO:0042823  
XM_018368617.1 GO:0042819|GO:0042823
```

# ファミリー／ドメインアノテーションの複雑さ



検索結果は各配列に一つではなく、ドメイン単位でつけられるもの、ドメインはオーバーラップしているが、ファミリーの大きさが異なるものなどが含まれる。

# ホモロジーに基づくアノテーションの戦略

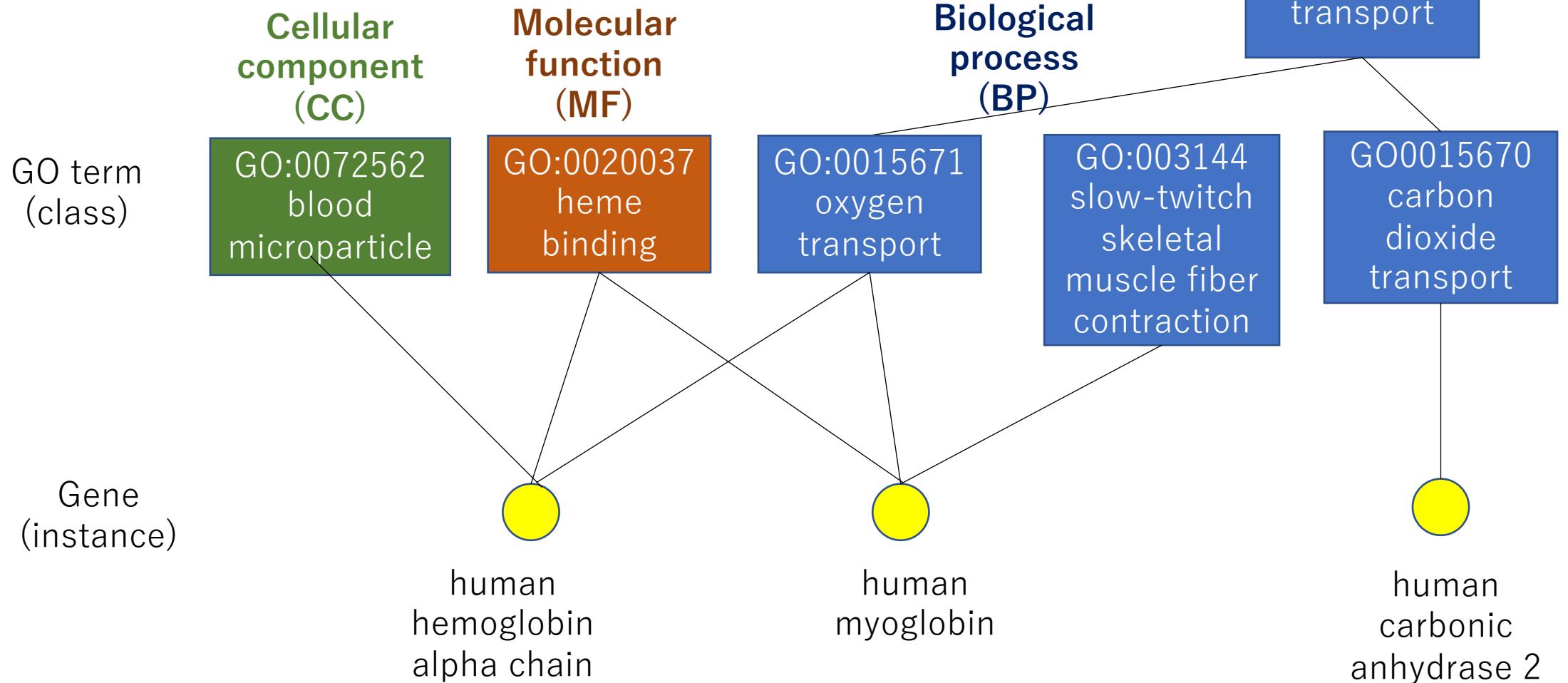
- 近縁種で、高品質なアノテーションがついたモデル生物ゲノムが利用可能な場合——そのゲノムに対するオーソログを同定し、アノテーションをコピーする。
- 特にターゲットとする生物種を絞らず、幅広い生物種から情報を集めたい場合——nrなどの網羅的なデータベースを検索して上位のヒットのアノテーションをコピーする。
- ホモロジー検索だけでは類似性が低い配列しかヒットせず、信頼性が低い場合——InterProなどのモチーフデータベースの検索を併用する。

## テキスト記述によるアノテーションの問題点

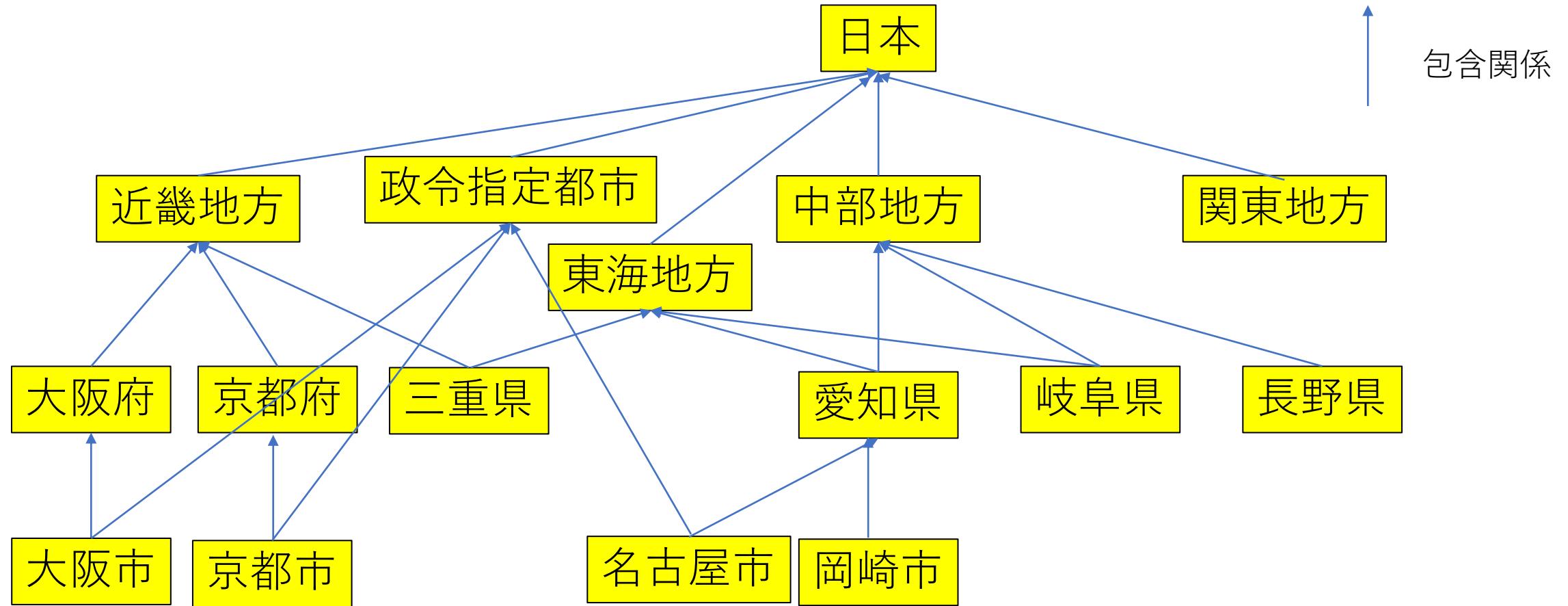
- 基本的に遺伝子（タンパク質）の名前を記載しただけで、具体的な機能について記載しているわけではない。
- 生物学的な解釈を考えるには、各遺伝子の機能に関する知識が別途必要になる。
- 大規模なRNA-seq解析結果を解釈するには、この部分についても計算機のサポートが必要。

# Gene Ontology (GO)

## 機械可読な遺伝子機能の表現

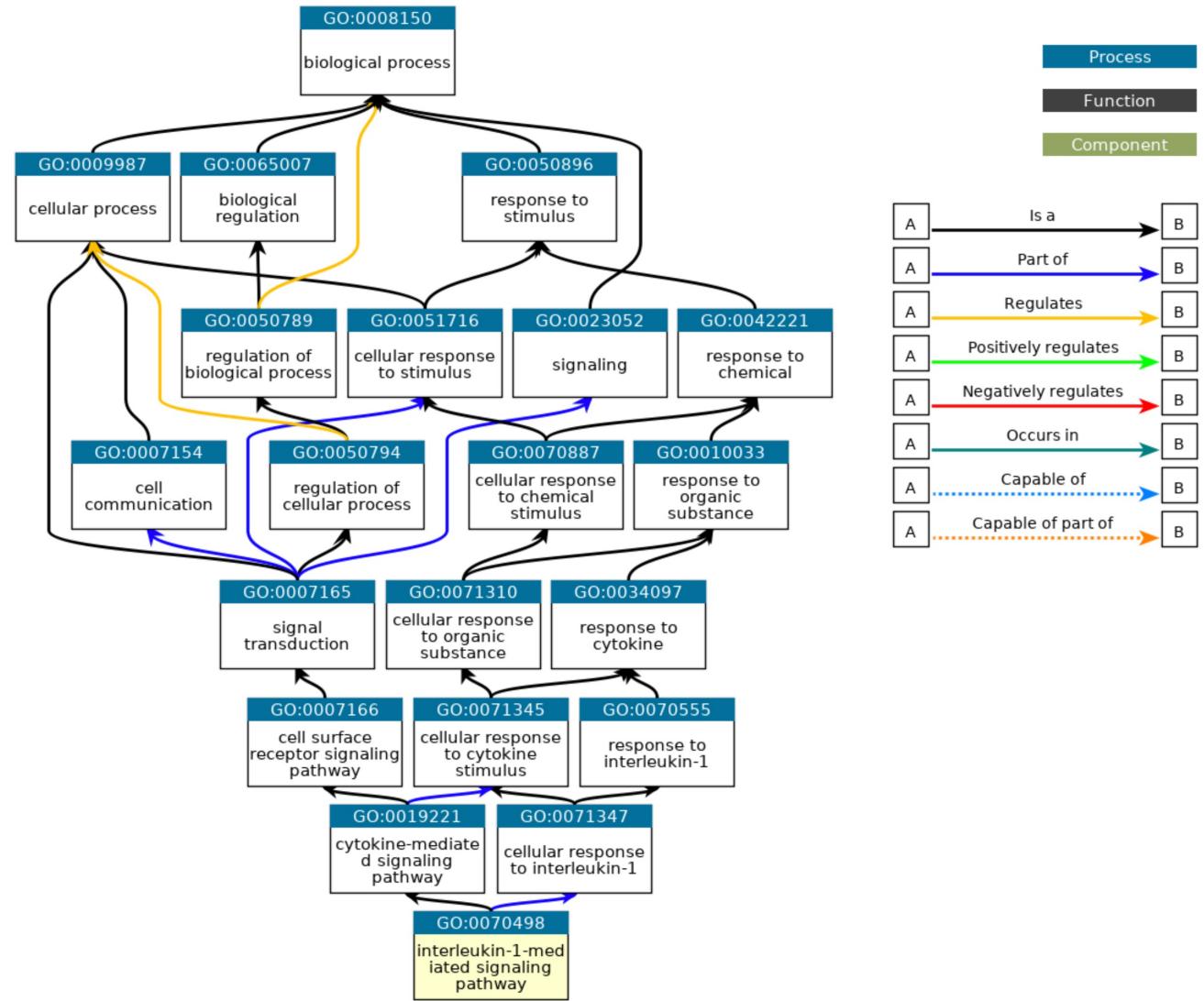


# 包含関係を表すグラフ：有向非巡回グラフ Directed Acyclic Graph (DAG)



# GO階層のグラフ

- 各ノードはGO termを表す
  - 最上位ノードは以下の3種類
    - biological process
    - molecular function
    - cellular component
- 各矢印（エッジ）はGO term間の関係を表す
  - 包含関係を表す矢印は2種類
    - is\_a AはBの一種である
    - part\_of AはBの一部である
  - その他の関係を表す矢印
    - regulates
    - occurs\_in など



# GO annotation

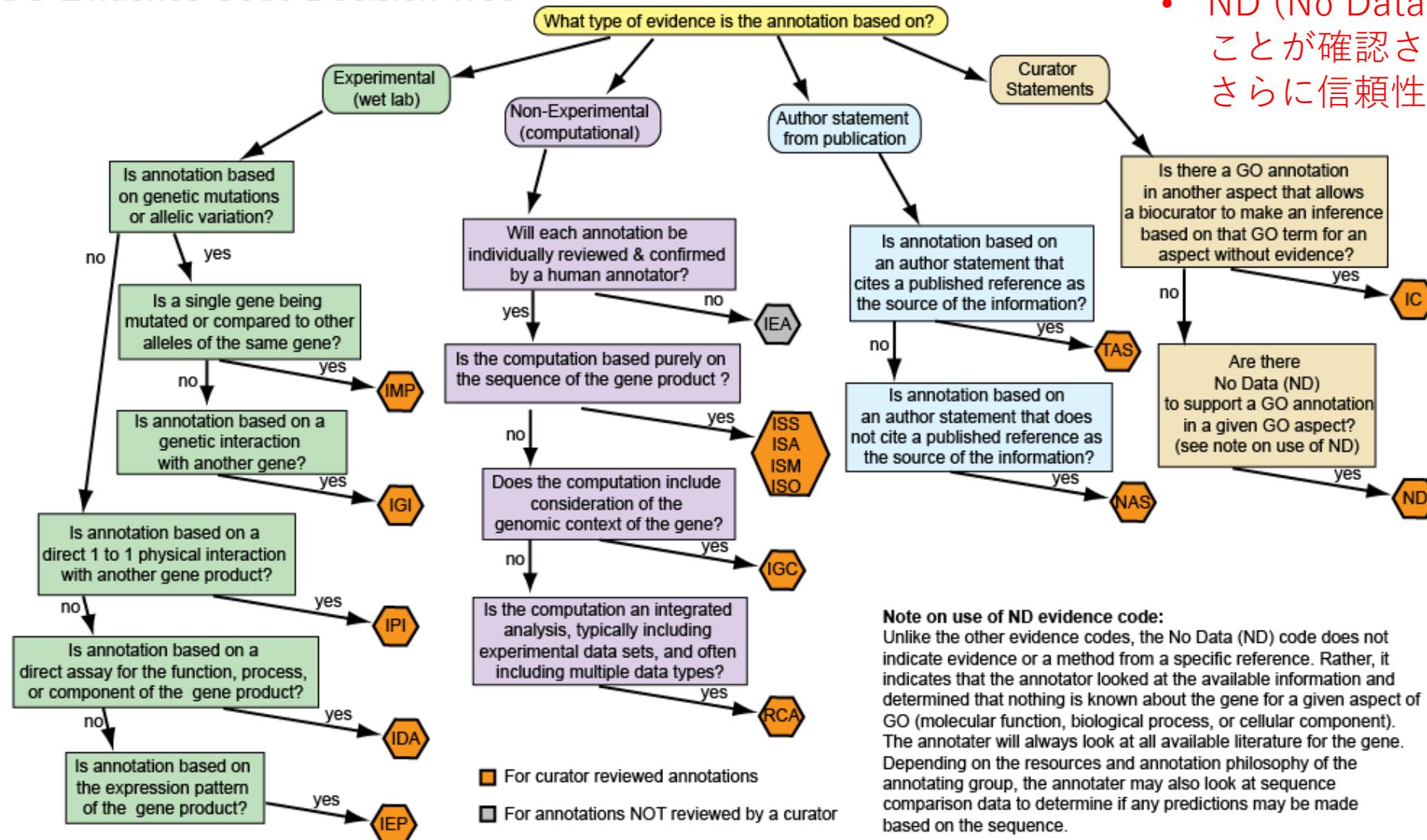
## 既知遺伝子へのGO termの割り当て

- GOアノテーションデータベース
  - モデル生物ゲノムデータベース
    - MGI (マウス)
    - FlyBase (ショウジョウバエ)
    - TAIR (シロイヌナズナ)
    - SGD (酵母) など
  - 網羅的なデータベース
    - タンパク質配列データベースUniProt
    - モチーフドメインデータベース InterPro
    - パスウェイデータベースReactome など
- アノテーションの根拠を Evidence Codeで示す

Human IKBKB gene に対するGO アノテーション													
Gene/product	Gene/product name	Annotation qualifier	GO class (direct)	Annotation extension	Contributor	Organism	Evidence	Evidence with	PANTHER family	Type	Isoform	Reference	Date
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		stimulatory C-type lectin receptor signaling pathway		Reactome	Homo sapiens	TAS		ikb kinase pthr22969	protein		Reactome:R-HSA-5621481	20181121
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent		Reactome	Homo sapiens	TAS		ikb kinase pthr22969	protein		Reactome:R-HSA-1236974	20180419
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		MyD88-independent toll-like receptor signaling pathway		Reactome	Homo sapiens	TAS		ikb kinase pthr22969	protein		Reactome:R-HSA-168927	20171201
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein kinase activity		UniProt	Homo sapiens	IDA		ikb kinase pthr22969	protein		PMID:20434986	20100804
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		UniProt	Homo sapiens	IDA		ikb kinase pthr22969	protein		PMID:15084260	20151001
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		Reactome	Homo sapiens	EXP		ikb kinase pthr22969	protein		PMID:18692471	20170505
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		Reactome	Homo sapiens	EXP		ikb kinase pthr22969	protein		PMID:23613522	20170505
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		UniProt	Homo sapiens	IDA		ikb kinase pthr22969	protein		PMID:25326418	20200702
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity	CACAO	Homo sapiens	IDA			ikb kinase pthr22969	protein		PMID:25636800	20151001
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		Reactome	Homo sapiens	TAS		ikb kinase pthr22969	protein		Reactome:R-HSA-168140	20170811
<input type="checkbox"/> IKBKB	Inhibitor of nuclear factor kappa-B kinase subunit beta		protein serine/threonine kinase activity		Reactome	Homo sapiens	TAS		ikb kinase pthr22969	protein		Reactome:R-HSA-202541	20150207

# Evidence Code

## GO Evidence Code Decision Tree



- IEA (Inferred from Electronic Annotation) は計算機による予測のみなので要注意！
- ND (No Data) は、証拠がないことが確認されているので、さらに信頼性が低い

### Note on use of ND evidence code:

Unlike the other evidence codes, the No Data (ND) code does not indicate evidence or a method from a specific reference. Rather, it indicates that the annotator looked at the available information and determined that nothing is known about the gene for a given aspect of GO (molecular function, biological process, or cellular component). The annotator will always look at all available literature for the gene. Depending on the resources and annotation philosophy of the annotating group, the annotator may also look at sequence comparison data to determine if any predictions may be made based on the sequence.

# ホモロジーに基づくアノテーションの戦略（再掲）

- 近縁種で、高品質なアノテーションがついたモデル生物ゲノムが利用可能な場合——そのゲノムに対するオーソログを同定し、アノテーションをコピーする。
- 特にターゲットとする生物種を絞らず、幅広い生物種から情報を集めたい場合——nrなどの網羅的なデータベースを検索して上位のヒットのアノテーションをコピーする。
- ホモロジー検索だけでは類似性が低い配列しかヒットせず、信頼性が低い場合——InterProなどのモチーフデータベースの検索を併用する。

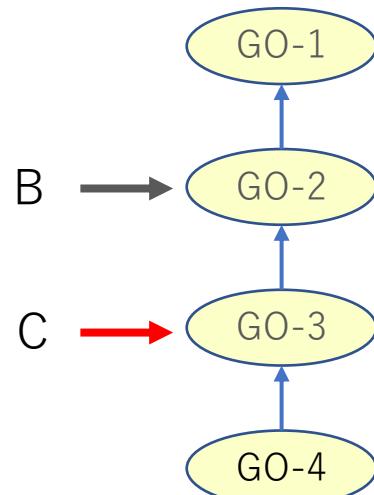
GOを用いてアノテーションする場合、GOアノテーションのついたデータベースを検索して、その結果に基づいてGO termをコピーすればよい。ただし、アノテーションのクオリティを考慮すると、2番目のケースは特に注意が必要。

# ホモロジーに基づくGOアノテーション

Hit	Score	GO	Evidence
Gene-A	90	GO-2	IEA
Gene-B	85	GO-2	IDA
Gene-C	80	GO-3	IDA
Gene-D	60	GO-4	IEA

- 類似性検索でトップヒットのGOを採用すれば良いとは限らない。

→スコアが同程度なら、エビデンスコードを考慮して、より信頼性の高いアノテーションを採用した方がよい。



- あるGO termがアサインされる場合、その上位のGO termも必ずアサインされる。

→アノテーションとしては、その遺伝子に当たる最も下位のGO termを記載する。

# BLAST2GO

- Annotation Score:  $AS = DT + AT$
- $DT = \max(\text{similarity} \times ECw)$   
 $ECw$ : Evidence Code weight
- $AT = (\#GO - 1) \times GOw$   
 $\#GO$ : number of child GOs assigned  
 $GOw$ : GO weight (user defined parameter)
- $AS \geq \text{threshold}$  を満たす最下層のGOをアサインする

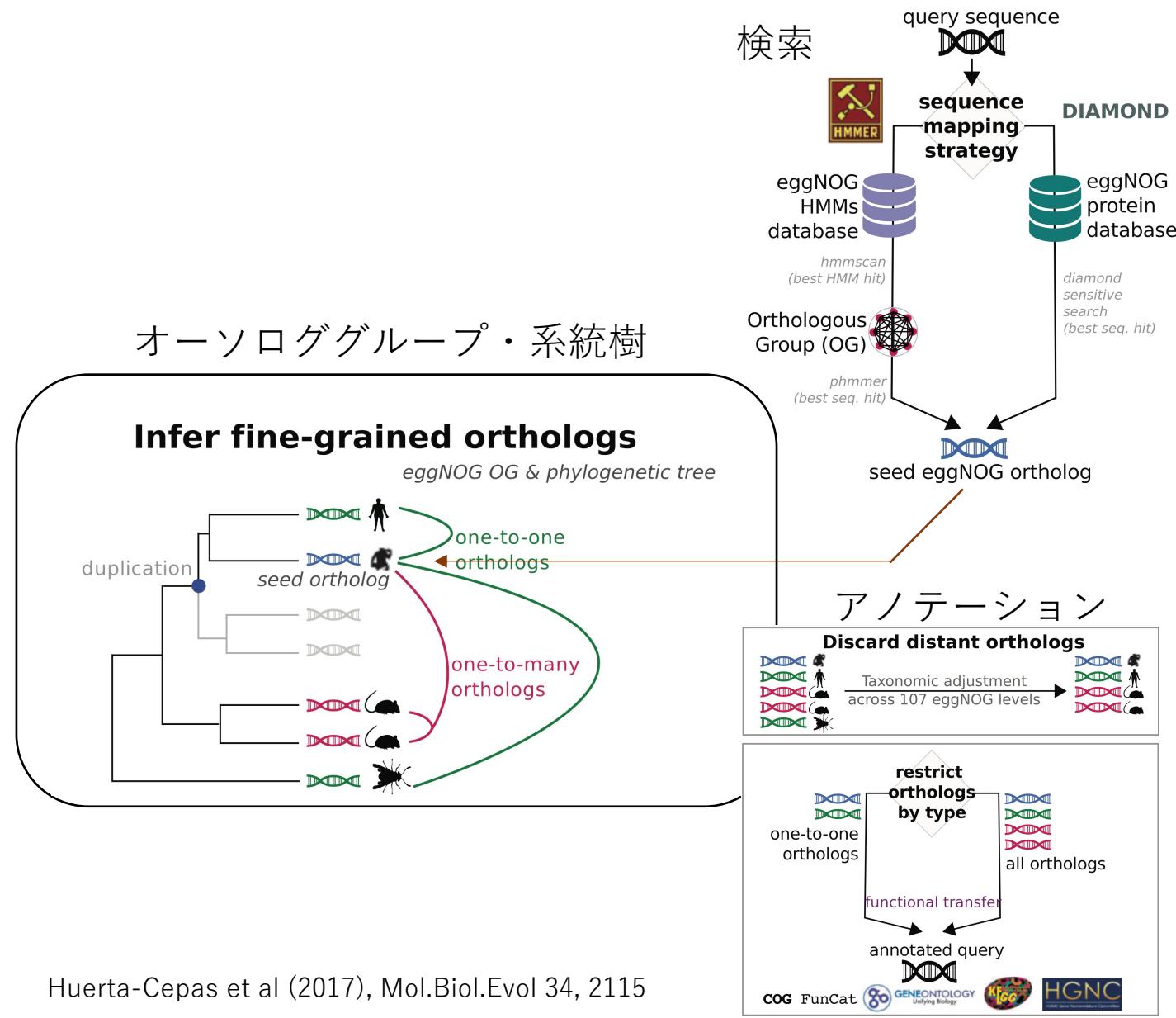
Evidence code weight

EC	Description	Default
IDA	Inferred from direct assay	1
IMP	Inferred from mutant phenotype	1
IGI	Inferred from genetic interaction	1
IPI	Inferred from physical interaction	1
IEP	Inferred from expression pattern	1
TAS	Traceable author statement	0.9
NAS	Non-traceable author statement	0.9
IC	Inferred by curator	0.9
ISS	Inferred from sequence or structural similarity	0.9
RCA	Inferred from reviewed computational analysis	0.9
IEA	Inferred from electronic annotation	0.7
ND	No biological data available	0.5
NR	Not recorded	0.5

Götz et al. (2008) Nucl.Acids.Res. 36, 3420

→OmixBoxという有償ソフトへ統合化

# EggNOG Mapper オーソログベースのアノテーション



- あらかじめデータベース中の配列をオーソロググループに分類し、各グループの系統樹を作成。
- クエリ配列に対するホモロジー検索によりseed ortholog配列を同定し、さらに系統樹を用いて他生物種におけるオーソログを同定する。
- seed orthologに対する類縁関係を考慮し、近縁種のオーソログに付けられた機能アノテーションをコピーする。

# EggNOG mapper の実行

# DIAMONDを用いて、EggNOG mapperの実行

```
% emapper.py -i query.fa -o outname -m diamond -cpu 6
```

# アノテーション結果から、テキスト記述を抽出

```
% cut -f1,8 outname.emapper.annotations
```

#query_name	narr_og_desc
XM_018368616.1	expression is induced before the diauxic shift and also in the absence of thiamin
XM_018368617.1	Belongs to the PdxS SNZ family
XM_018368618.1	member of a subtelomeric gene family including THI5 THI11 THI12 and THI13
XM_018368619.1	siderophore-bound iron prior to uptake by transporters
XM_018368620.1	involved in the retention of siderophore-iron in the cell wall

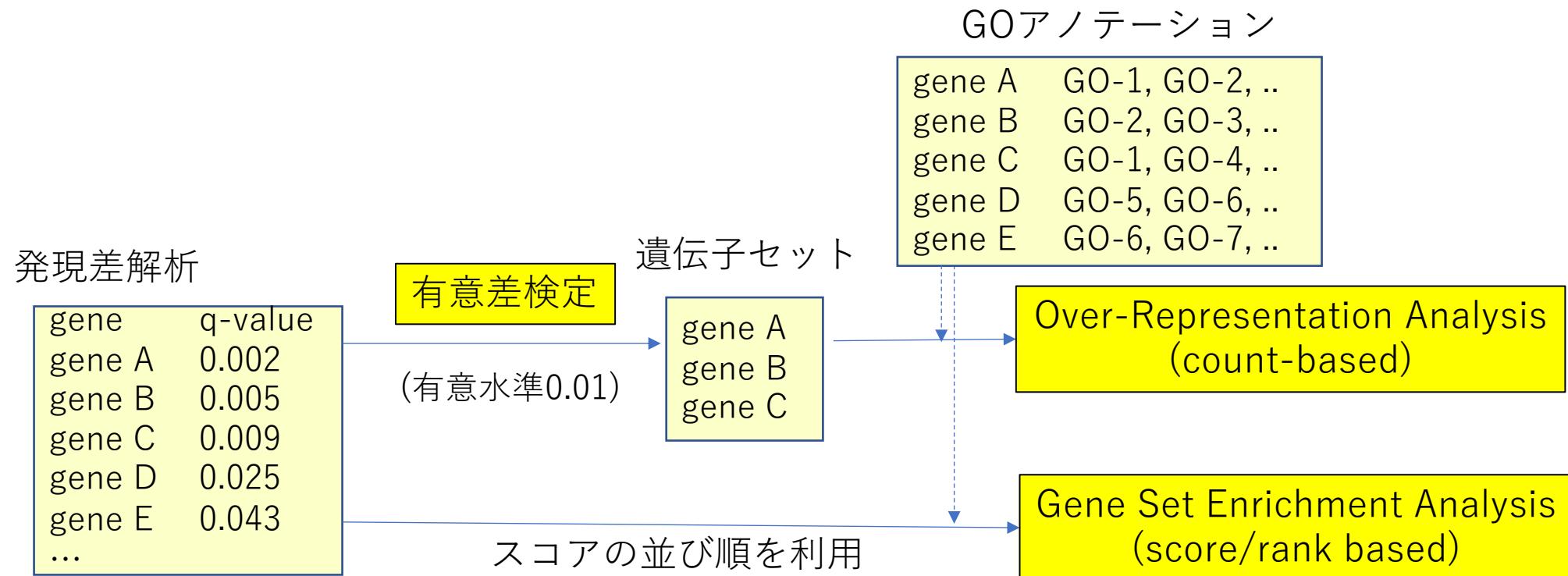
# アノテーション結果から、アサインされたGOのリストを抽出

```
% cut -f1,13 outname.emapper.annotations
```

#query_name	GOs
XM_018368616.1	GO:0003674, GO:0003824, GO:0004359, GO:0005575, GO:0005622, GO:0005623, GO:0005737, GO:0005829, GO:0006081, ..
XM_018368617.1	GO:0000096, GO:0000097, GO:0003674, GO:0003824, GO:0003922, GO:0005575, GO:0005576, GO:0006082, GO:0006520, ..
XM_018368618.1	GO:0003674, GO:0003824, GO:0006725, GO:0006766, GO:0006767, GO:0006772, GO:0006790, GO:0006807, GO:0008150, ..
XM_018368619.1	GO:0000041, GO:0000293, GO:0000322, GO:0000323, GO:0000324, GO:0000329, GO:0003674, GO:0003824, GO:0005575, ..
XM_018368620.1	GO:0000322, GO:0000323, GO:0000324, GO:0005575, GO:0005618, GO:0005622, GO:0005623, GO:0005737, GO:0005773, ..

# GO enrichment 解析

- ・発現量が増加／減少した遺伝子群において、より多く出現する（エンリッチしている）機能(GO term)を抽出する。
- ・まず設定した閾値（有意水準）によって遺伝子セットを抽出し、その中でエンリッチメント解析を行うアプローチと、スコアの並び順を用いてエンリッチメント解析を行うアプローチがある。



# Overrepresentation Analysis (Fisher's exact test)

一組のトランプから10枚のカードを抜き出したとき、赤札（ハート、ダイヤ）が5枚、絵札（JQK）が4枚含まれていた。このとき、赤札と絵札のどちらがよりエンリッチ（濃縮）していると言えるか？



赤札 5枚

全体  $26/52 = 0.5$

手札  $5/10 = 0.5$

$p=0.6368$

絵札4枚

全体  $12/52 = 0.23$

手札  $4/10 = 0.4$

$p=0.2492$

ダイヤの絵札3枚

全体  $3/52 = 0.058$

手札  $3/10 = 0.3$

$p=0.00543$

分割表

	絵札	数札	計
手札	4	6	10
残り	8	34	42
計	12	40	52

Rでの計算

```
> tab <- matrix( c(4,8,6,34), 2, 2 )
> fisher.test(tab,
               alternative='greater')
```

## Fisher's exact test (正確確率検定)

N枚のカード中にn枚の「当たり」が含まれるとき、そこからm枚を抜き出した中に「当たり」がk枚以上(または以下)含まれる確率を求める。

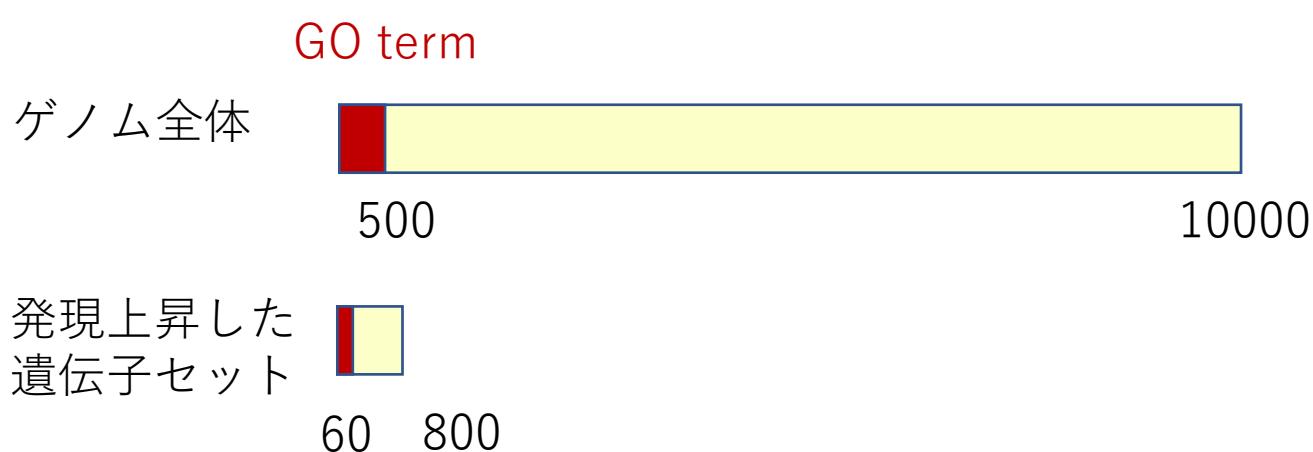
→超幾何分布(Hypergeometric distribution)  
で計算できる

$$P(k; n, m, N) = \frac{\binom{n}{k} \binom{N-n}{m-k}}{\binom{N}{m}}$$

# Overrepresentation Analysis (Fisher's exact test)

発現上昇した遺伝子セット800個のうち、60個にあるGO termが付けられていた。ゲノム全体10000遺伝子のなかで、そのGO termが付けられた遺伝子は500個であった。このGO termは発現上昇した遺伝子セット中で過剰出現していると言えるか？

発現上昇した遺伝子数が1000個であった場合はどうか？



分割表

	機能を持つ	機能を持たない	計
発現上昇あり	60	740	800
発現上昇なし	440	8760	9200
計	500	9500	10000

→ Fisherの正確確率検定 (Fisher's exact test)

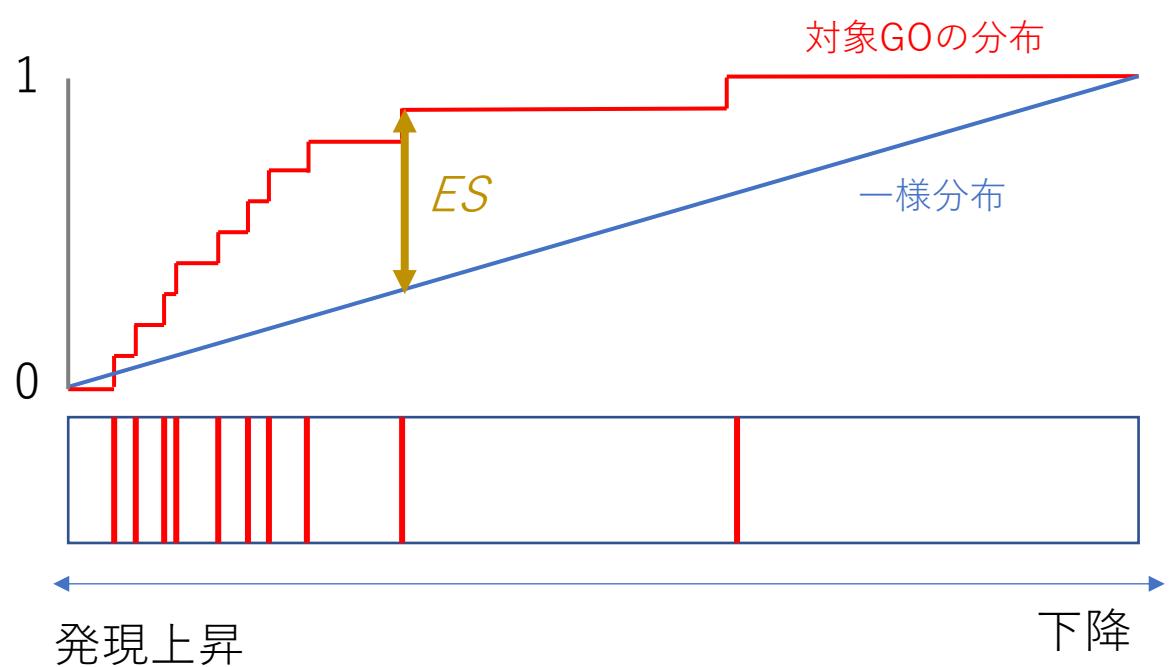
```
fisher.test( matrix( c(60, 440, 740, 8760), 2, 2), alternative='greater' ) p=0.00089
```

(発現上昇した遺伝子が1000個の場合)

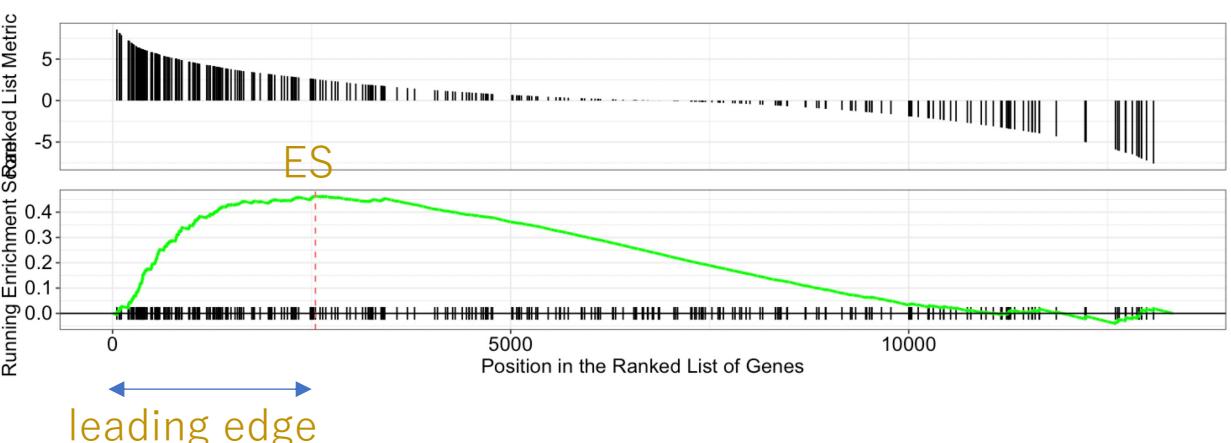
```
fisher.test( matrix( c(60, 440, 940, 8560), 2, 2), alternative='greater' ) p=0.076
```

結果が発現差  
解析の閾値に  
依存

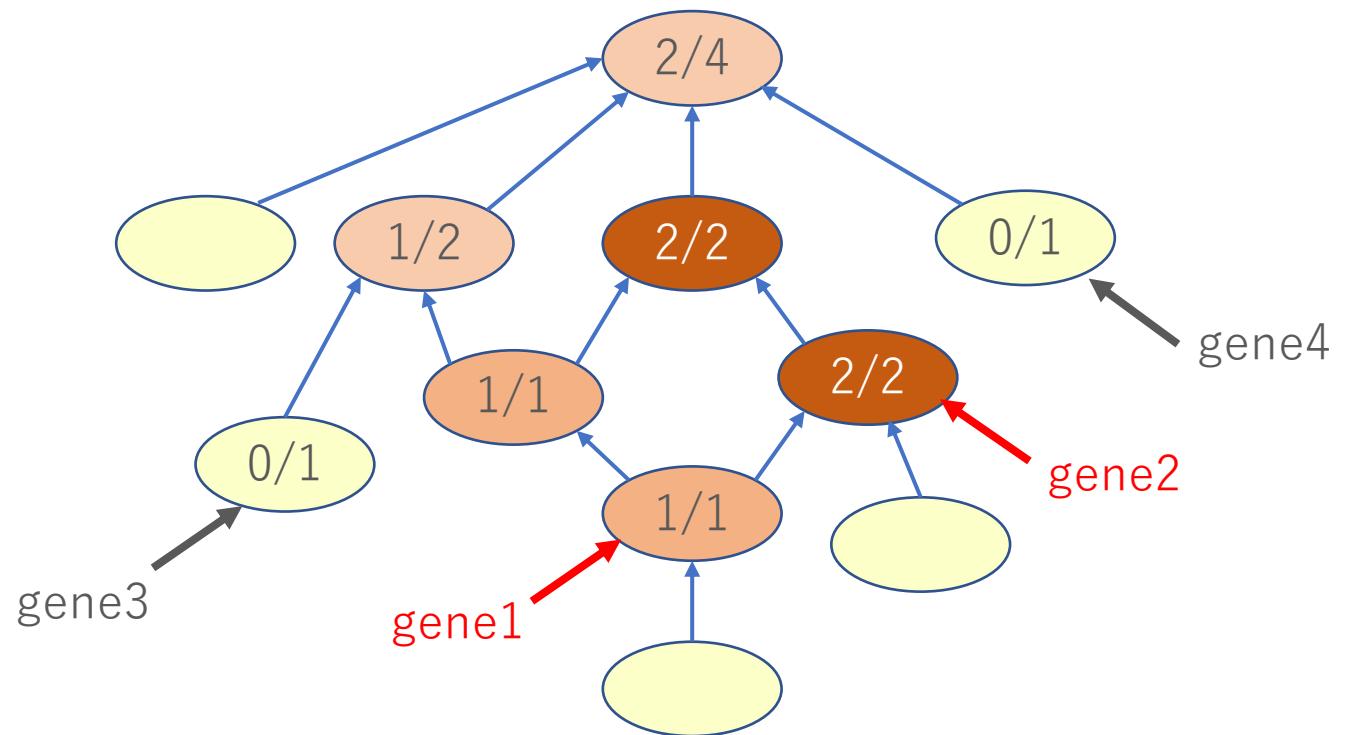
# Gene Set Enrichment Analysis (GSEA)



- 全遺伝子を発現解析の結果に基づいてソートし、その並び順の中で、対象とする遺伝子セットの出現が、偏りのない分布（一様分布）からずれているかを判定する（Kolmogorov-Smirnov検定）。
- カウントを、ソートに用いた指標で重み付けすることにより、上位/下位のヒットにより大きな重みを与えることもできる。



# GO enrichment 解析とGO階層



- GO enrichment解析では、全てのGO termについてenrichment testを繰り返す。
- GO階層の下位ノードにアサインされた遺伝子は上位ノードにも自動的にアサインされるため、上下ノードに依存関係が生じる。

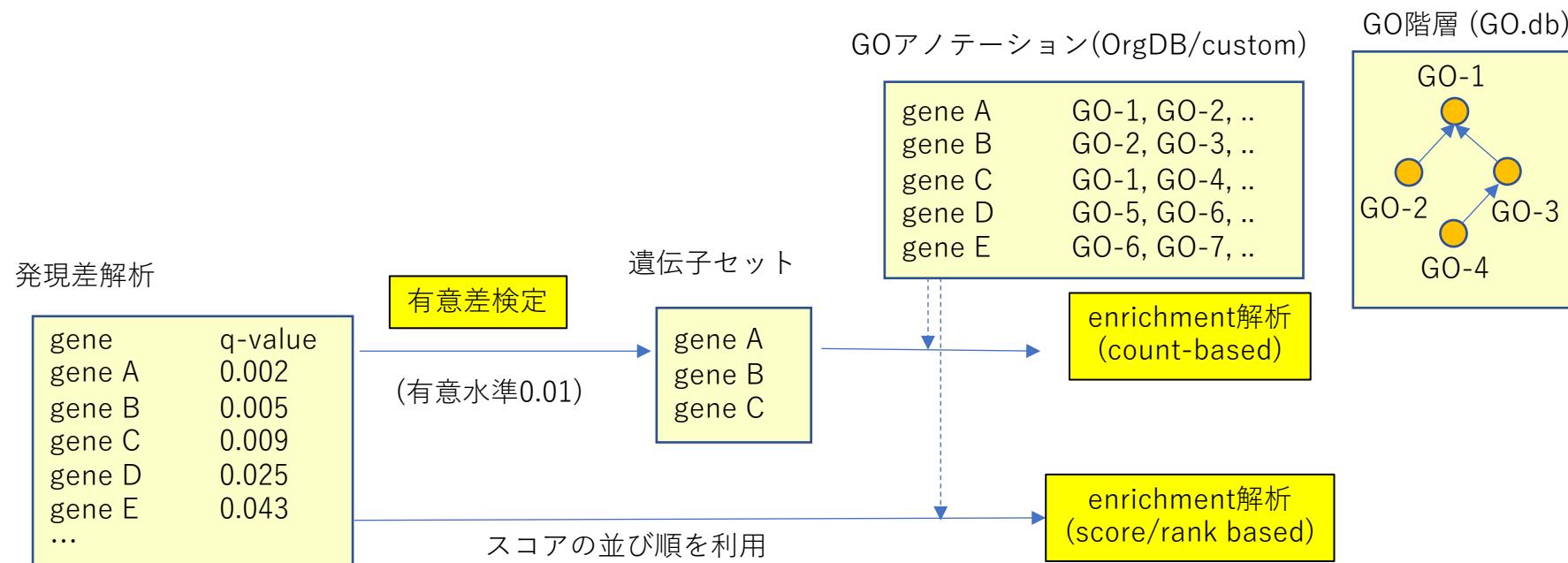
→有意なGO termは、GO階層上近傍に集まって出現することが多い。

→上位ノードは分母が大きくなるため、「濃縮度」は低くなる。

# Rを用いたGO enrichment 解析

- clusterProfiler
  - GOのほか、KEGGやDisease Ontology(DO)など、様々なデータベースに対応
  - 結果を可視化するツールが充実している

- TopGO
  - 隣接するGO階層間で生じる冗長性を排除するアルゴリズムを実装
  - 様々な統計手法とアルゴリズムを組み合わせてenrichment解析を行う汎用的な枠組みを提供



# 入力データの準備

```
# edgeRのデータオブジェクト(DGEList)から解析結果全体をテーブルとして抜き出す  
etab <- topTags(DGEList, n=999999)$table
```

## ## Overrepresentation analysis (Fisher's exact test)用の遺伝子リストの作成

```
# FDRが0.01以下で、かつ2倍以上(logFC>1)発現上昇した遺伝子を抽出  
upreg <- subset(etab, logFC > 1.0 & etab$FDR < 0.01)  
# 発現上昇した遺伝子と全体の遺伝子名リストを作成  
upregGenes <- rownames(upreg)  
allGenes <- rownames(etab)
```

## ## Gene set enrichment analysis (GSEA)用のソートされたスコアリストの作成

```
# -log(FDR)にlogFCの符号をかけてup/downを区別した指標を算出  
expScore <- sign(etab$logFC) * -log(etab$FDR)  
# expScoreの各値にnameとして遺伝子名を対応づけ、昇順にソートする  
names(expScore) <- rownames(etab)  
expScore <- sort(expScore, decreasing=TRUE )
```

# GOアノテーションの準備

## 公開されたアノテーションデータベースの利用1

### OrgDb

- 代表的モデル生物種の遺伝子データベースでGOのアノテーションを含む。
- org.<Sp>.<id>.db という名称のパッケージとして公開。ヒトorg.Hs.eg.db、マウスorg.Mm.eg.dbなど、<id>=eg (Entrez Gene)のものを中心に現在約20種類。
- enrichment解析に用いるには、発現解析に用いた遺伝子IDが、NCBI GenelDなど、このデータベースに登録されたIDのいずれかと一致する必要がある。
- キー（遺伝子ID）とカラム(抽出数r遺伝子属性)を指定して情報を取り出す。

```
> library(org.Mm.eg.db)
# マウス用データベース。ロードすると、org.Mm.eg.db という名前のオブジェクトを介してデータベースにアクセスできる
# 利用可能な属性の一覧を表示
> columns(org.Mm.eg.db)
[1] "ACCCNUM"        "ALIAS"          "ENSEMBL"         "ENSEMLPROT"      "ENSEMLTRANS"
[6] "ENTREZID"       "ENZYME"         "EVIDENCE"        "EVIDENCEALL"    "GENENAME"
[11] "GO"              "GOALL"          "IPI"             "MGI"            "ONTOLOGY"
[16] "ONTOLOGYALL"   "PATH"           "PFAM"           "PMID"          "PROSITE"
[21] "REFSEQ"         "SYMBOL"         "UNIGENE"        "UNIPROT"
# SYMBOLが "KRAS" である遺伝子のGOを表示
> select(org.Mm.eg.db, keys="Kras", keytype="SYMBOL", columns="GO")
  SYMBOL          GO EVIDENCE ONTOLOGY
1  Kras GO:0000166     IEA      MF
2  Kras GO:0001934     ISO      BP
3  Kras GO:0003924     IGI      MF
```

# GOアノテーションの準備

## 公開されたアノテーションデータベースの利用2

### AnnotationHub

- より多くの生物種のアノテーションを集積。必要なものをダウンロードして使う。"OrgDb"として登録されたものは前項のOrgDbデータベースと同様に使える。

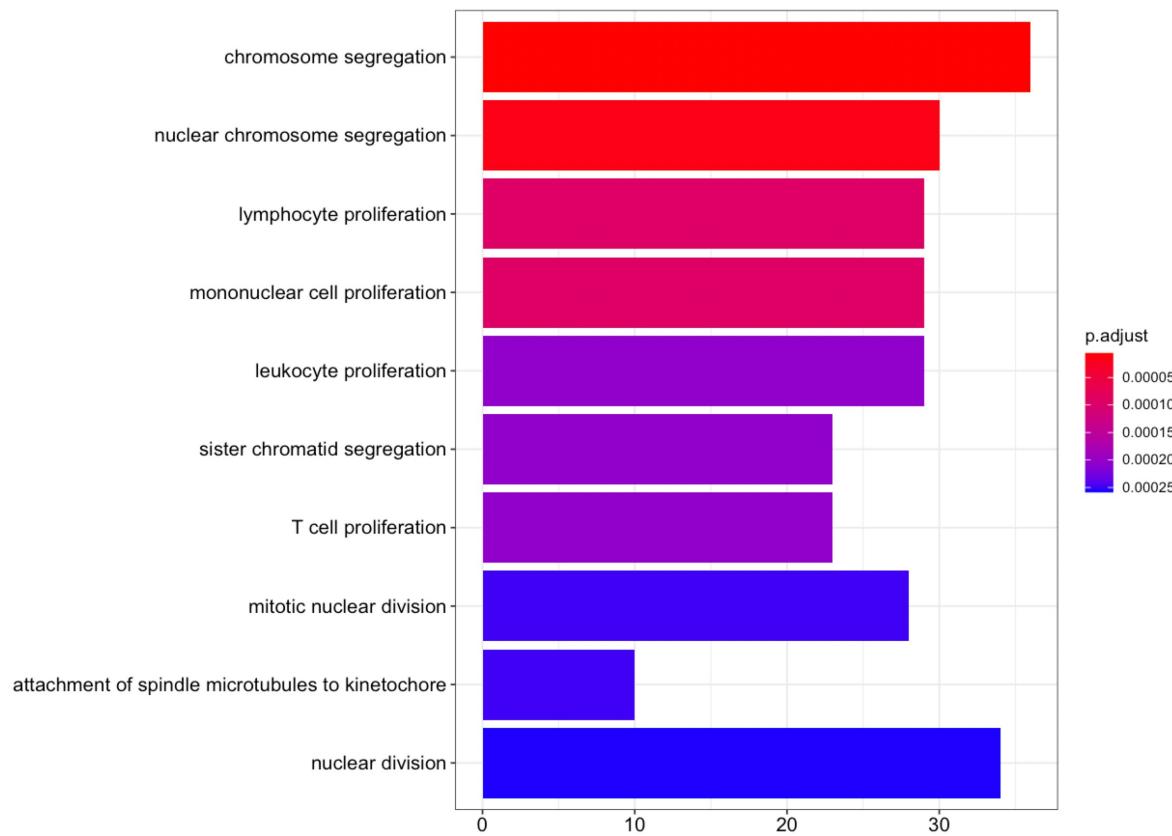
```
> library(AnnotationHub)
# AnnotationHubへのアクセスインターフェイスを変数ahに格納
> ah <- AnnotationHub()
# イネ(Oryza sativa)を検索。Queryコマンドの2番目の引数は、任意の数のキーワードをベクトルとして与える。
> query(ah, c("OrgDb", "oryza sativa"))
AnnotationHub with 3 records # <- ヒットが3つ見つかっている
# snapshotDate(): 2020-10-27
...
# retrieve records with, e.g., 'object[["AH85565"]]' # <- データへのアクセス方法。objectは変数名ahに置き換える
      title           # 以下、ヒットしたデータの一覧
AH85565 | org.Oryza_sativa_(japonica_cultivar-group).eg.sqlite
AH85566 | org.Oryza_sativa_Japonica_Group.eg.sqlite
AH85567 | org.Oryza_sativa_subsp._japonica.eg.sqlite
# データをダウンロードし、org.os.dbという変数に格納
> org.os.db <- ah[["AH85565"]]
> columns(org.os.db)
"ACCCNUM"    "ALIAS"    "CHR"      "ENTREZID"   "EVIDENCE"   "EVIDENCEALL" "GENENAME"   "GID"        "GO"
"GOALL"      "ONTOLOGY"  "ONTOLOGYALL" "PMID"       "REFSEQ"     "SYMBOL"
```

# clusterProfilerによる解析(Fisher's exact test)

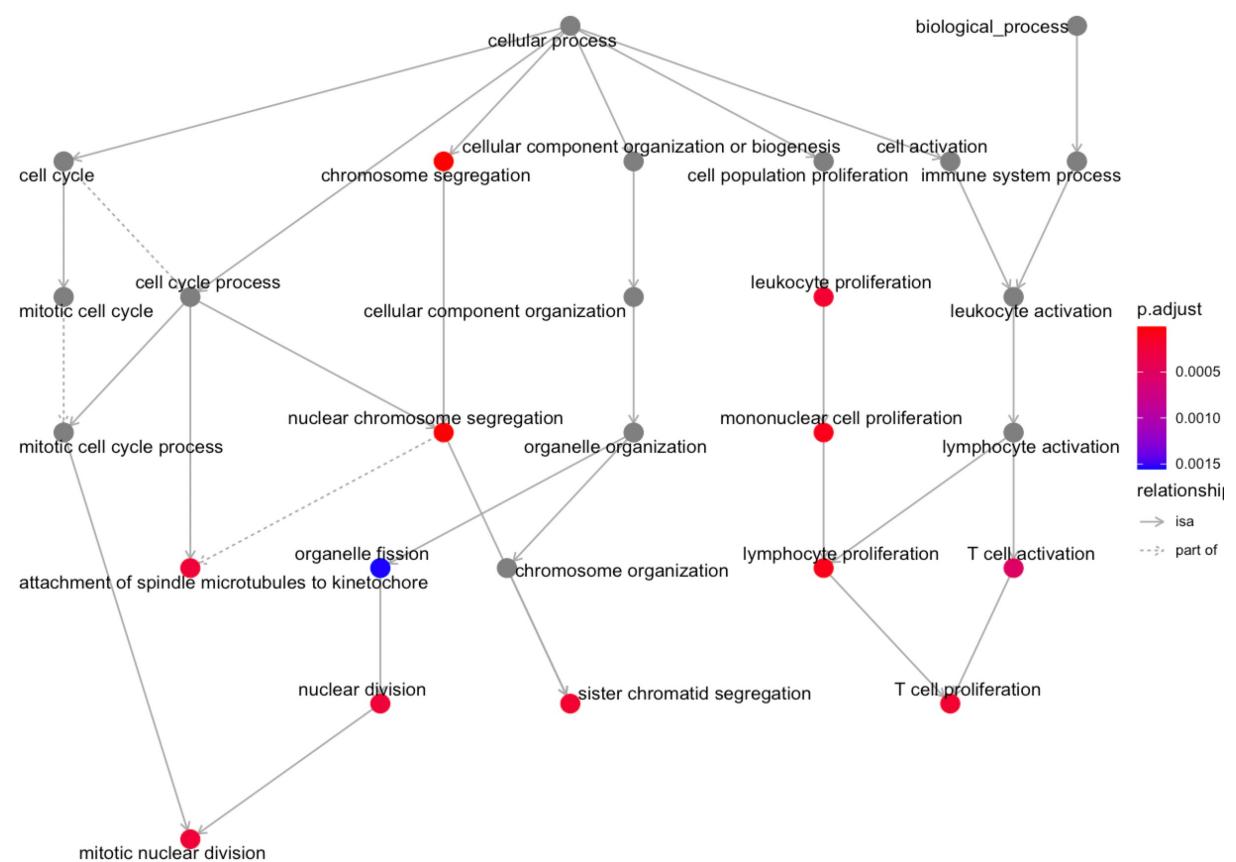
```
> library(clusterProfiler)
# Biological Process (BP)に対してFisher's exact testの実施。遺伝子の検索キーはENSEMBL
> cprof.up.bp.fisher <- enrichGO(gene=upregGenes, universe=allGenes,
  OrgDb=org.Mm.eg.db, ont="BP", keyType="ENSEMBL",
  pvalueCutoff=0.01)
# 上位20個のGO termを表示
> head(cprof.up.bp.fisher, 20)
```

# 解析結果の可視化(Fisher's exact test)

```
> library(enrichplot)  
> barplot(cprof.up.bp.fisher, showCategory=10)
```

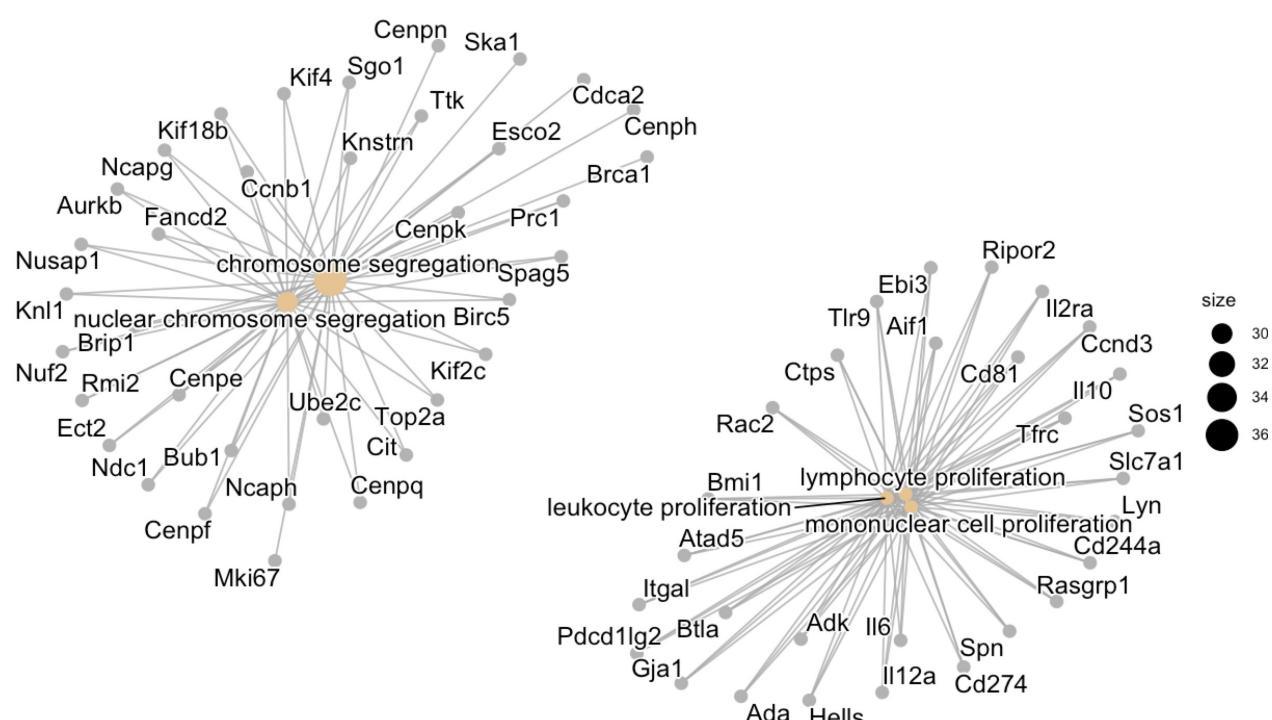


```
> goplot(cprof.up.bp.fisher, showCategory=10)
```

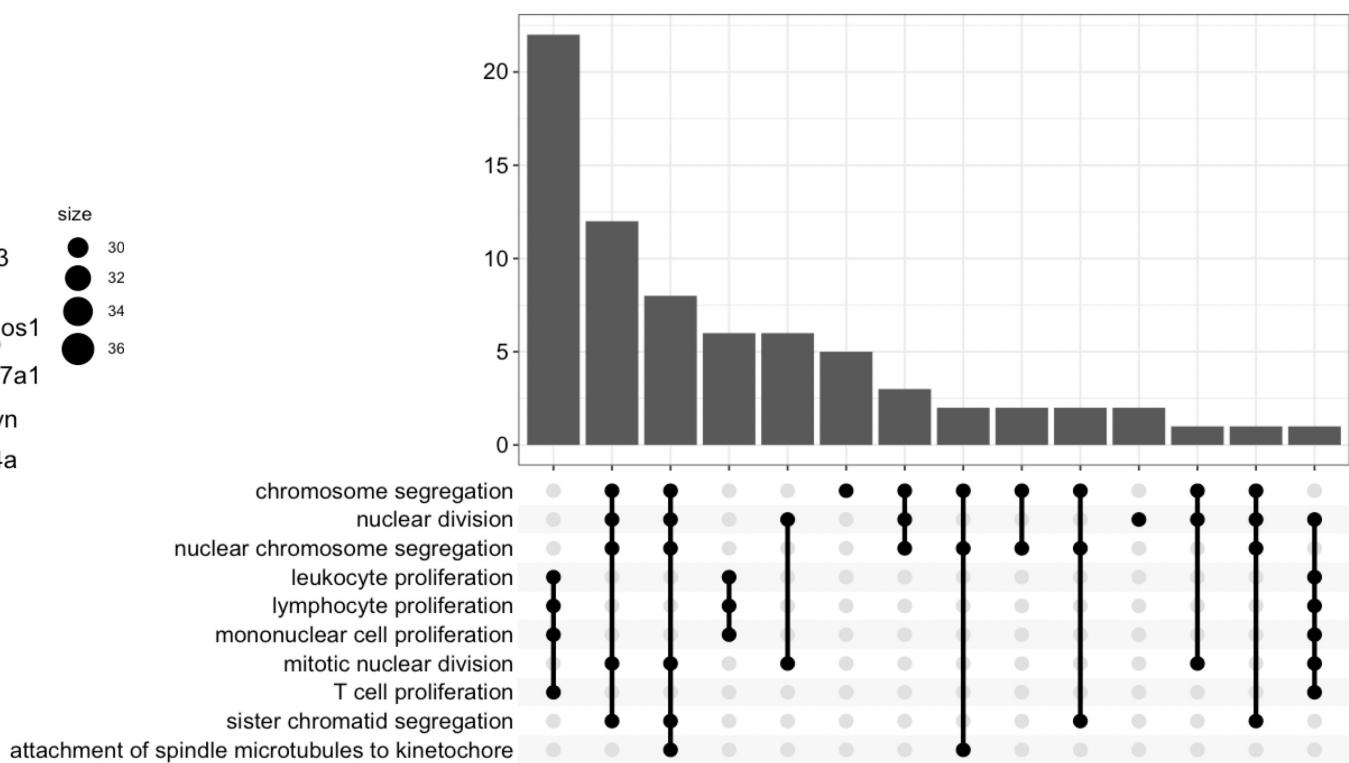


# 解析結果の可視化(Fisher's exact test)

```
# org.Mm.eg.db データベースを使って、ENSEMBL IDから可読性の高い名前に変換する  
> cprof.up.bp.fisher <- setReadable(cprof.up.bp.fisher, 'org.Mm.eg.db', "ENSEMBL")  
> cnetplot(cprof.up.bp.fisher)
```



```
> upsetplot(cprof.up.bp.fisher)
```



# clusterProfilerによる解析(Gene Set Enrichment Analysis)

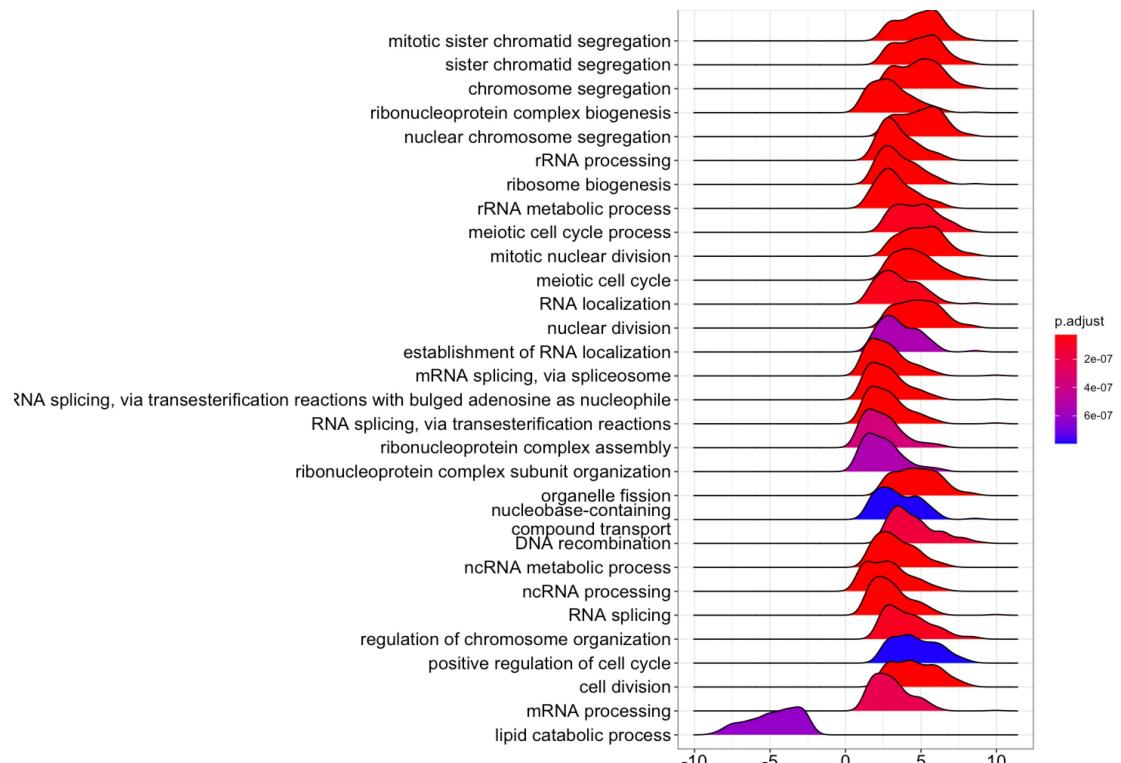
```
# Gene Set Enrichment Analysisの実施
```

```
> cprof.bp.gsea <- gseGO(geneList=expScore, OrgDb=org.Mm.eg.db, ont="BP",
  keyType="ENSEMBL", pvalueCutoff=0.1)
```

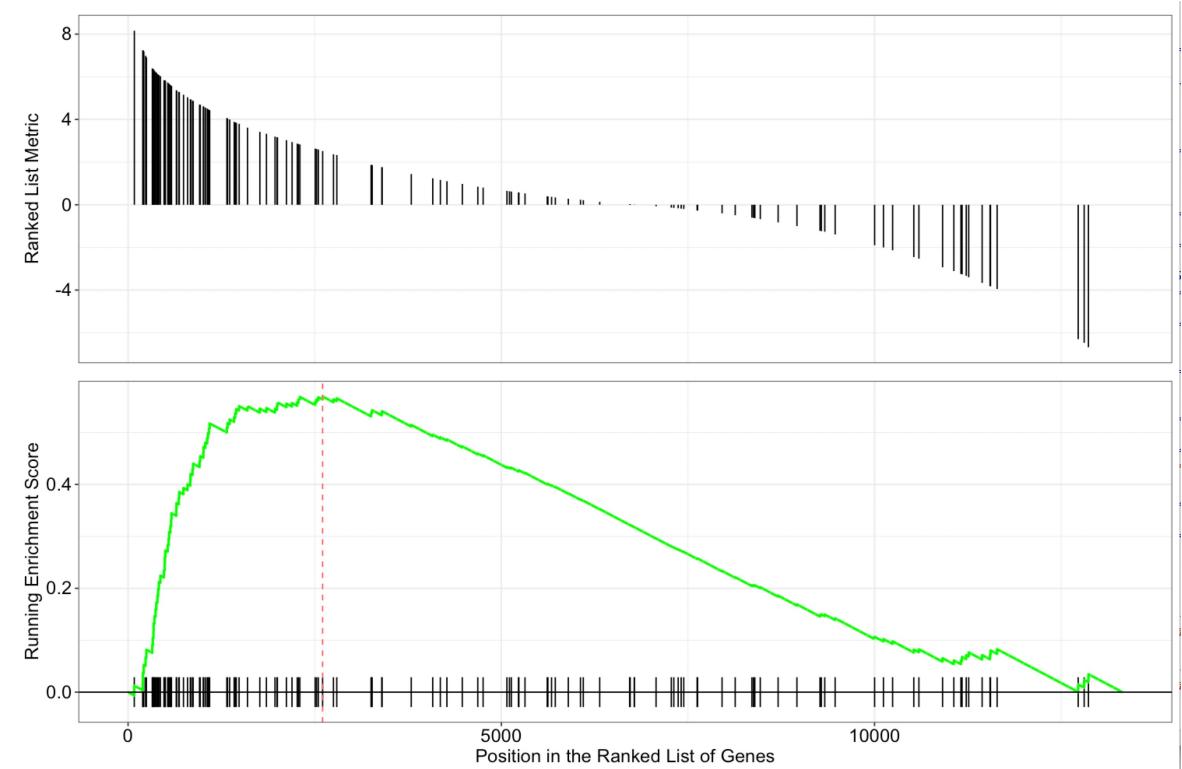
# 上位20個のGO termを表示

```
> head(cprof.up.bp.fisher, 20)
```

```
> ridgeplot(cprof.bp.gsea)
```



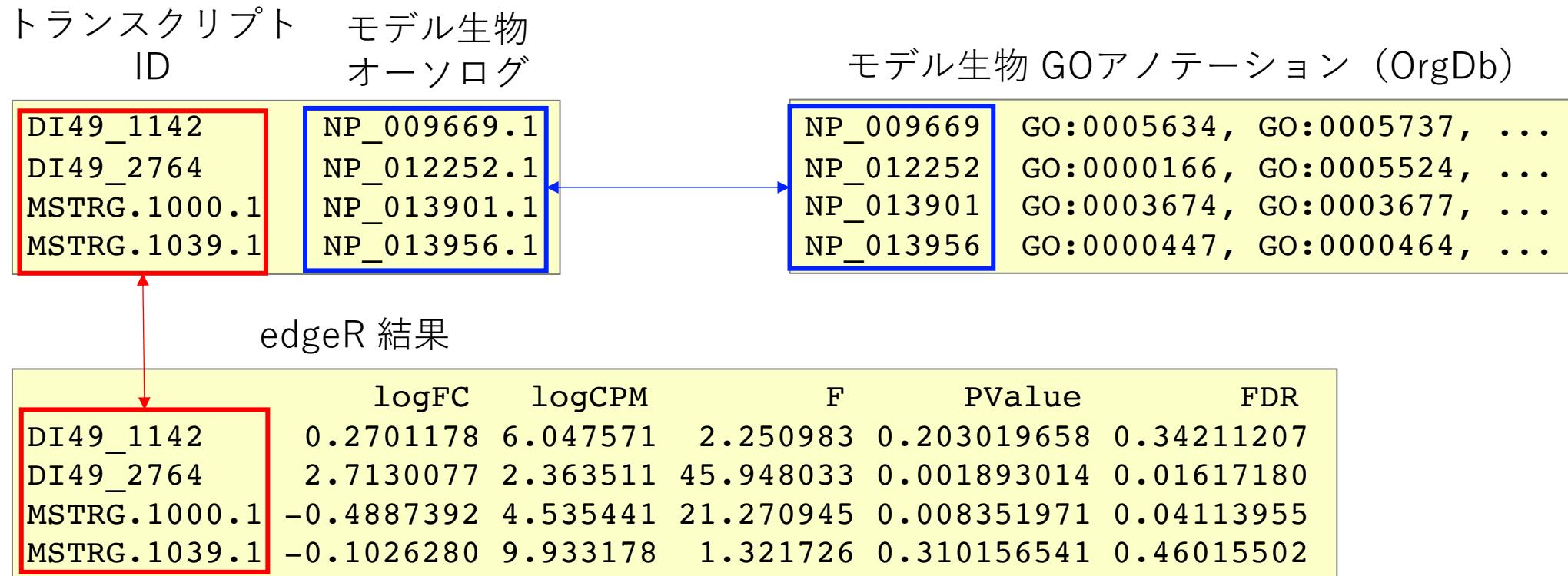
```
> gseaplot(cprof.bp.gsea, geneSetID=1)
```



# ユーザーアノテーションの利用 1

## オーソログ解析に基づくアノテーション

- 近縁のモデル生物の遺伝子に対するオーソログ対応づけができている場合、それを基にモデル生物のアノテーションを対応づけてGO解析を行うことができる。



# ユーザーアノテーションの利用 2

## 遺伝子-GO対応表の取り込み

- 遺伝子ごとのGO termのアサインができている場合、それを直接読み込んで解析する。
- clusterProfilerでは、以下のいずれかの形式で遺伝子とGOの対応表を準備する。

GMT 形式

GO-1	GOname1	gene1	gene2	gene3
GO-2	GOname2	gene4	gene5	

`read.gmt(gmfile)` で読み込む。  
GOnameを読み飛ばして、右の形式と同じ  
データフレームを生成する。

または、以下の形式

GO-1	gene1
GO-1	gene2
GO-1	gene3
GO-2	gene4
GO-2	gene5

`read.delim(file)` で読み込む

- clusterProfilerでは、GO ID とその名前の対応表(TERM2NAME)も用意する必要がある。

GO-1	GOname1
GO-2	GOname2

# clusterProfilerによる解析(対応表を読み込む)

```
# 遺伝子とGOID対応表の読み込み  
> go2gene <- read.gmt("go2gene.gmt")  
# 祖先ノードに遡ってGOをアサインする  
> go2gene.expand <- biuldG0map(go2gene)  
# TERM2NAMEの対応表をGO.dbから作成する  
> gonames <- select(GO.db, keys=keys(GO.db), columns="TERM")  
# Fisher's exact testの実施  
> cprof.fisher2 <- enricher(gene=upregGenes, TERM2GENE=go2gene.expand,  
    TERM2NAME=gonames)  
# Gene Set Enrichment Analysisの実施  
> cprof.gsea2 <- GSEA(geneList=expScore, TERM2GENE=go2gene.expand  
    TERM2NAME=gonames)
```

# TopGO

- ・多様なGOエンリッチメント解析に対応したRパッケージ。
- ・様々な「統計手法」と、それをGO階層に適用する際の「アルゴリズム」の組み合わせを選択できる。

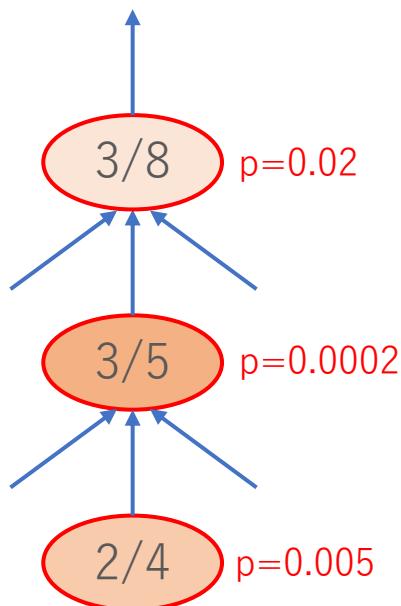
algorithms	statistics				
	fisher	ks	t	globaltest	sum
classic	✓	✓	✓	✓	✓
elim	✓	✓	✓	✓	✓
weight	✓	-	-	-	-
weight01	✓	✓	✓	✓	✓
lea	✓	✓	✓	✓	✓
parentchild	✓	-	-	-	-



ks: Kolmogorov-Smirnov test = GSEA

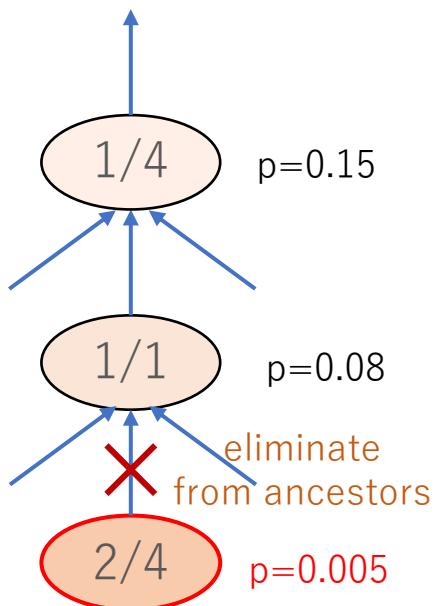
# Algorithms in TopGO

classic



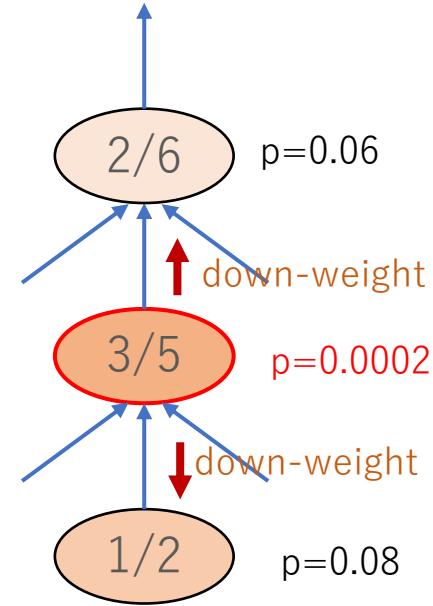
何もしない

eilm



子ノードを優先

weight



p-valueが低い  
ノードを優先

- ボトムアップに検定を行い、p-valueを計算。
- 有意性が確認されたノード中の遺伝子の影響が隣接するノードに広がらないように、親ノードや子ノードから遺伝子を除いたり、重みを下げたりする。

# TopGO の実行

```
# Fisher test の対象とする遺伝子セットを抽出するための関数を定義
# 遺伝子ごとのスコアexpScoreが-log(FDR)で定義されているとして、FDR<0.01という条件を指定
> selfun <- function(x) {return(x>-log(0.01))}

# expScoreとそのアノテーションなど、解析に必要なデータ群をまとめたオブジェクトを作成
> topgoData = new("topGOdata", ontology="BP", allGenes=expScore,
   geneSel=selfun, annot=annFUN.org, maping="org.Mm.eg.db", ID="ENSEMBL")

# エンリッチメント解析の実行。同じrunTest関数で様々な解析を実行できる。
> resultFisher <- runTest(topgoData, algorithm="classic", statistic="fisher")
> resultElimKS <- runTest(topgoData, algorithm="elim", statistic="ks")

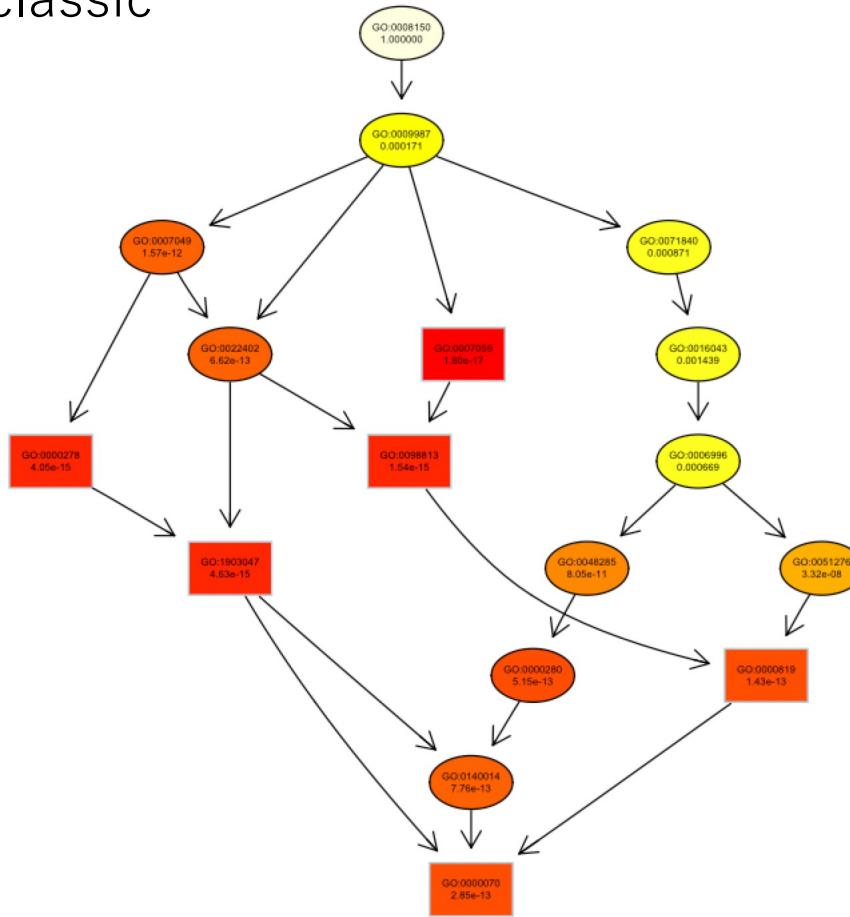
# 複数の解析結果をテーブルにまとめて表示する。
> GenTable(topgoData, classicFisher=resultFisher, elimKS=resultElimKS)
```

	GO.ID	Term	Annotated	Significant	Expected	Rank	in elimKS	classicFisher	elimKS
1	GO:0007059	chromosome segregation	270	67	21.67	13649	1.8e-17	0.99	
2	GO:0098813	nuclear chromosome segregation	211	55	16.93	13343	1.5e-15	0.98	
3	GO:0000278	mitotic cell cycle	699	117	56.09	12719	4.1e-15	0.96	
4	GO:1903047	mitotic cell cycle process	581	103	46.63	13032	4.6e-15	0.97	
5	GO:0000819	sister chromatid segregation	160	44	12.84	13469	1.4e-13	0.98	
6	GO:0000070	mitotic sister chromatid segregation	138	40	11.07	13325	2.9e-13	0.98	
7	GO:0000280	nuclear division	313	65	25.12	13575	5.1e-13	0.99	
8	GO:0022402	cell cycle process	906	134	72.71	12268	6.6e-13	0.94	
9	GO:0140014	mitotic nuclear division	228	53	18.30	13716	7.8e-13	0.99	
10	GO:0007049	cell cycle	1303	174	104.57	11362	1.6e-12	0.89	

# TopGO 解析結果の可視化

```
> showSigOfNodes(topgoData, score(resultFisher), firstSigNodes=6, useInfo='pval')
```

## classic



weight

