

转录组数据分析

实验操作

目 录

转录组数据的产生

转录组数据的存储格式——FASTQ

 知识点1——什么是phred值

 知识点2——序列存储格式FASTA

转录组数据的质量检测——FastQC

转录组数据的质量控制——FASTX-Toolkit

转录组数据分析流程——Tophat、Cufflinks、Cuffdiff、CummeRbund

 知识点3——什么是gtf格式

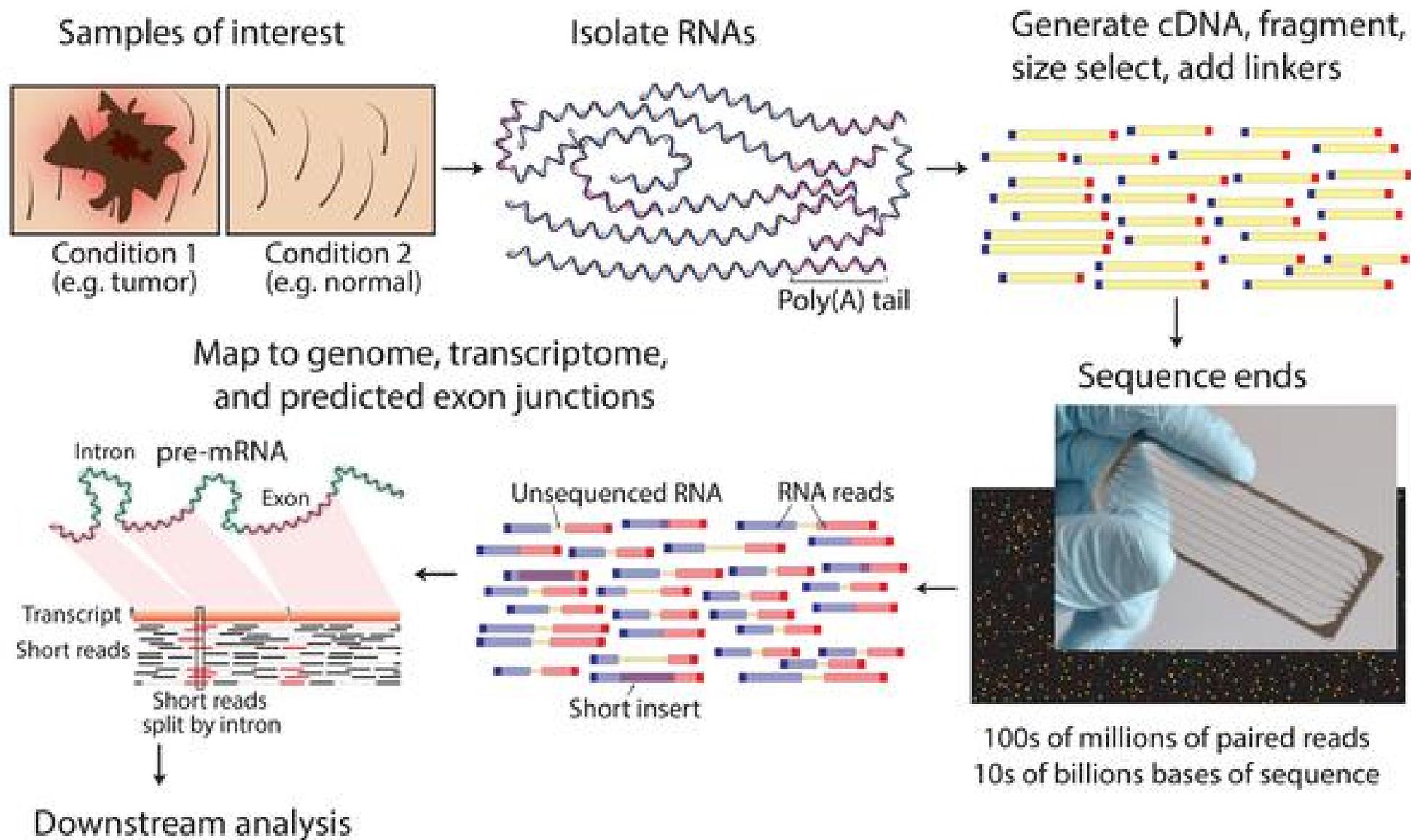
 知识点4——什么是bam格式

其他转录组数据分析流程（简要介绍HISAT、STRINGTIE、BALLGOWN）

GO功能分类

KEGG分析——KOBAS软件

转录组数据的产生



转录组数据的存储格式——FASTQ

```
@ST-E00126:128:HJFLHCCXX:2:1101:7405:1133
GATTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!' '*(((('*'+))%%%++) (%%%) .1***-+*'') **55CCF>>>>>CCCCCCC65
```

第1行主要储存序列测序时的坐标等信息；

第2行就是测序得到的序列信息，一般用ATCGN来表示，其中N用于荧光信号干扰无法判断到底是哪个碱基时的代表符号；

第3行以“+”开始，可以储存一些附加信息，但目前的测序fastq文件这一行一般是空的。

第4行储存的是质量信息，与第2行的碱基序列是一一对应的，其中的每一个符号对应的ASCII值是经过换算的phred值，可以简单理解为对应位置碱基的测序质量值，越大说明测序的质量越好。不同的版本对应的phred值范围不同。

知识点1——什么是phred值

Phred值是评估这个bp测序质量的值，测序仪通过判断荧光信号的颜色来判断碱基的种类，ATCG分别对应红黄蓝绿，信号强弱不同，在这种情况下对每个结果的判断的正确性都存在一个概率值，这个值被储存为ASCII码形式，转化方式如下：

- 将该碱基判断错误概率值P取log10之后再乘以-10，得到的结果为Q。

若P=1%，那么对应的

$$Q = -10 * \log_{10}(0.01) = 20$$

(这个计算公式illumina平台使用，

Solexa系列测序仪使用不同的公示来计算质量值： $Q = -10 \log(P/1-P)$)

- 把这个Q加上33或者64转成一个新的数值，称为Phred，最后把Phred对应的ASCII字符对应到这个碱基。

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII									
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 \$	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

ASCII_BASE=64 Old Illumina

Q	P_error	ASCII									
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

知识点2——序列存储格式FASTA

蛋白质fasta文件

```
>sp|P69905|HBA_HUMAN Hemoglobin subunit alpha OS=Homo sapiens GN=HBA1
MVLSPADKTNVKAAWGKVGHAHAGEYGAELERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDD
MPNALSALSSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKLASVSTVLTISKYR
```

核酸序列文件 (mRNA序列中的U均用T来代替)

```
>gi|13650073|gb|AF349571.1| Homo sapiens hemoglobin alpha-1 globin chain
(HBA1) mRNA, complete cds
CCCACAGACTCAGAGAGAACCCACCATGGTGCTGTCTCCTGACGACAAGACCAACGTCAAGGCCGCCTGGGGTAAG
GTCGGCGCGCACGCTGGCGAGTATGGTGCGGAGGCCCTGGAGAGGATGTTCCCTGTCCTTCCCCACCACCAAGACCT
ACTTCCC GCACTTCGACCTGAGCCACGGCTCTGCCCAGGTTAAGGGCCACGGCAAGAAGGTGGCCGACGCGCTGAC
CAACGCCCGTGGCGCACGTGGACGACATGCCCAACGCGCTGTCCGCCCTGAGCGACCTGCACGCGCACAAAGCTTCGG
GTGGACCCGGTCAACTTCAAGCTCCTAAGCCACTGCCTGCTGGTGACCCTGGCCGCCACCTCCCCGCCGAGTTCA
CCCCTGCGGTGCACGCCTCCCTGGACAAGTTCCTGGCTTCTGTGAGCACCGTGCTGACCTCCAAATACCGTTAAGC
TGGAGCCTCGGTGGCCATGCTTCTTGCCCCCTTTG
```

转录组数据的质量检测——FastQC

运行命令

```
fastqc -t 8 -o path/fastqc  
sample1_R1.fq sample1_R2.fq
```

参数

-o --outdir: 输出路径
--extract: 结果文件解压缩
--noextract: 结果文件压缩
-f --format: 输入文件格式
-t --threads: 线程数
-c --contaminants: 制定污染序列
-a --adapters: 指定接头序列
-k --kmers: 指定kmers长度 (2-10bp, 默认7bp)
-q --quiet: 安静模式

运行结果 (html和zip)

FastQC Report

Summary

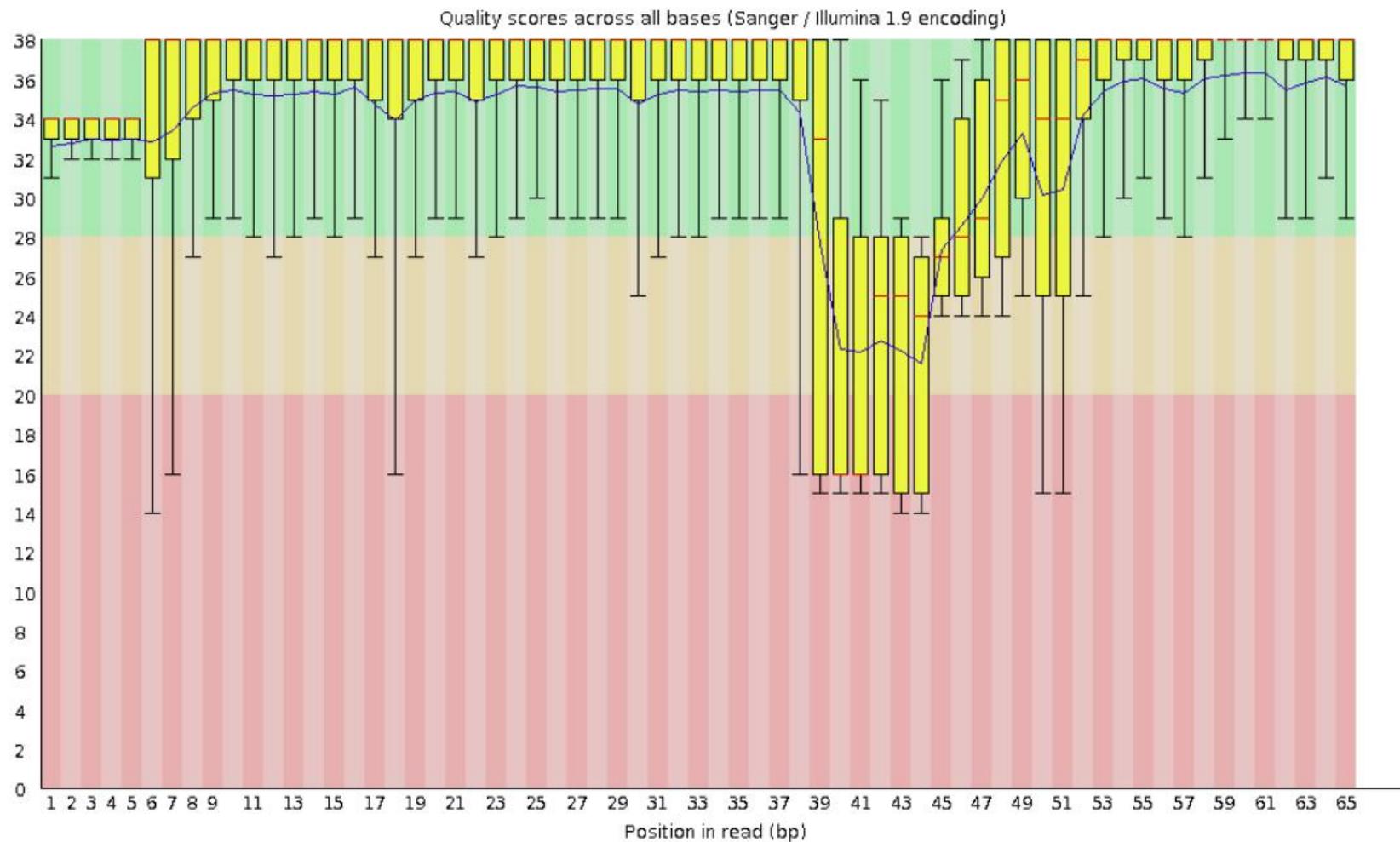
-  [Basic Statistics](#)
-  [Per base sequence quality](#)
-  [Per tile sequence quality](#)
-  [Per sequence quality scores](#)
-  [Per base sequence content](#)
-  [Per sequence GC content](#)
-  [Per base N content](#)
-  [Sequence Length Distribution](#)
-  [Sequence Duplication Levels](#)
-  [Overrepresented sequences](#)
-  [Adapter Content](#)
-  [Kmer Content](#)

转录组数据的质量检测——FastQC结果解读

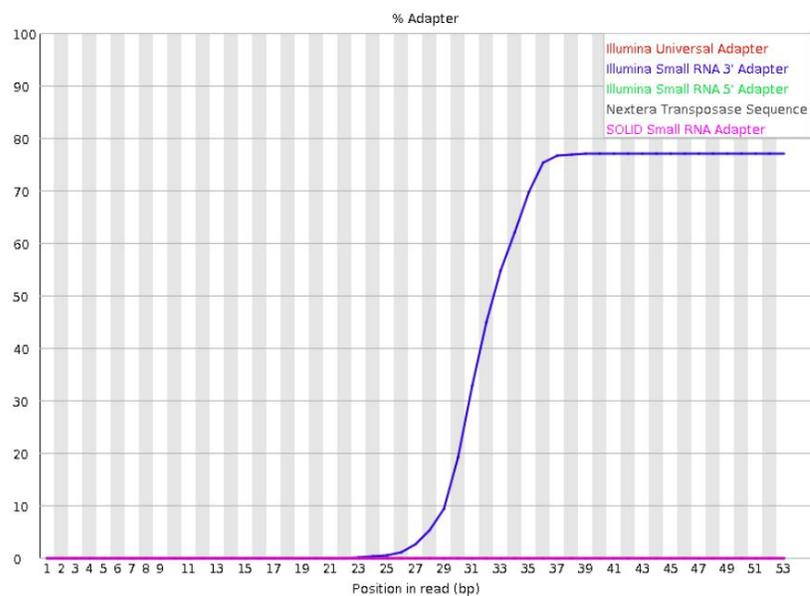
Basic Statistics

Measure	Value
Filename	SRR5345622.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	47613077
Sequences flagged as poor quality	0
Sequence length	65
%GC	64

Per base sequence quality



Adapter Content



转录组数据的质量控制——FASTX-Toolkit

(1) fastx_clipper去接头

运行命令

```
fastx_clipper [-h] [-a ADAPTER] [-D] [-l N] [-n] [-d N] [-c] [-C] [-o] [-v] [-z] [-i INFILE] [-o OUTFILE]
```

去接头例子

Clipping Example:

```
>1
ATGTAATGTTTATATATATATCGTAAATCCAACACAAT
>2
TATTTTGGAAATTCCACGACCCTGTAGGCACCATCAA
>3
ACGTTGTTTCGGTGCCTGAACTGTAGGCACCATC
>4
TTTCTTCTTATCTCTTCGAGTCTGTAGGCACCATCA
>5
TGGAACTTGCTGTAGGCACCATCATTATTTATATAA
>6
TTTACCGGAAGCATAACTCTTCTGTAGGCACCATCA
>7
TGTATTAGCGGTGGGGCCCGACTGTAGGCACCATCA
```



```
>2
TATTTTGGAAATTCCACGACC
>3
ACGTTGTTTCGGTGCCTGAA
>4
TTTCTTCTTATCTCTTCGAGT
>6
TTTACCGGAAGCATAACTCTT
>7
TGTATTAGCGGTGGGGCCCGA
```

In the above example:

- Sequence no. 1 was discarded since it wasn't clipped (i.e. didn't contain the adapter sequence). (**Output** parameter).
- Sequence no. 5 was discarded --- its length (after clipping) was shorter than 15 nt (**Minimum Sequence Length** parameter).

fastx_clipper参数

[-a ADAPTER] = 接头序列 (默认为CCTTAAGG)

[-l N] = 忽略那些碱基数目少于N的reads, 默认为5

[-d N] = 保留接头序列后的N个碱基默认 -d 0

[-c] = 只保留包含接头的序列

[-C] = 只保留没有接头的序列

[-k] = 报告只有接头的序列

[-n] = 保留有N多序列, 默认不保留

[-v] = 详细-报告序列编号

[-z] = 压缩输出

[-D] = 输出调试结果

[-M N] = 要求最小能匹配到接头的长度N

[-i INFILE] = 输入文件

[-o OUTFILE] = 输出文件

转录组数据的质量控制——FASTX-Toolkit

(2) fastq_quality_filter过滤低质量序列

运行命令 `fastq_quality_filter [-h] [-v] [-q N] [-p N] [-z] [-i INFILE] [-o OUTFILE]`

参数

- `[-q N]` = 最小的需要留下的质量值
- `[-p N]` = 每个reads中最少有百分之多少的碱基需要有`-q`的质量值
- `[-z]` = 压缩输出
- `[-v]` = 详细-报告序列编号, 如果使用了`-o`则报告会直接在STDOUT, 如果没有则输入到STDERR

例子

Quality score distribution (of all cycles) is calculated for each read. If it is lower than the quality cut-off value - the read is discarded.

Example:

```
@CSHL_4_FC042AG00II:1:2:214:584
GACAATAAAC
+CSHL_4_FC042AG00II:1:2:214:584
30 30 30 30 30 30 30 30 20 10
```

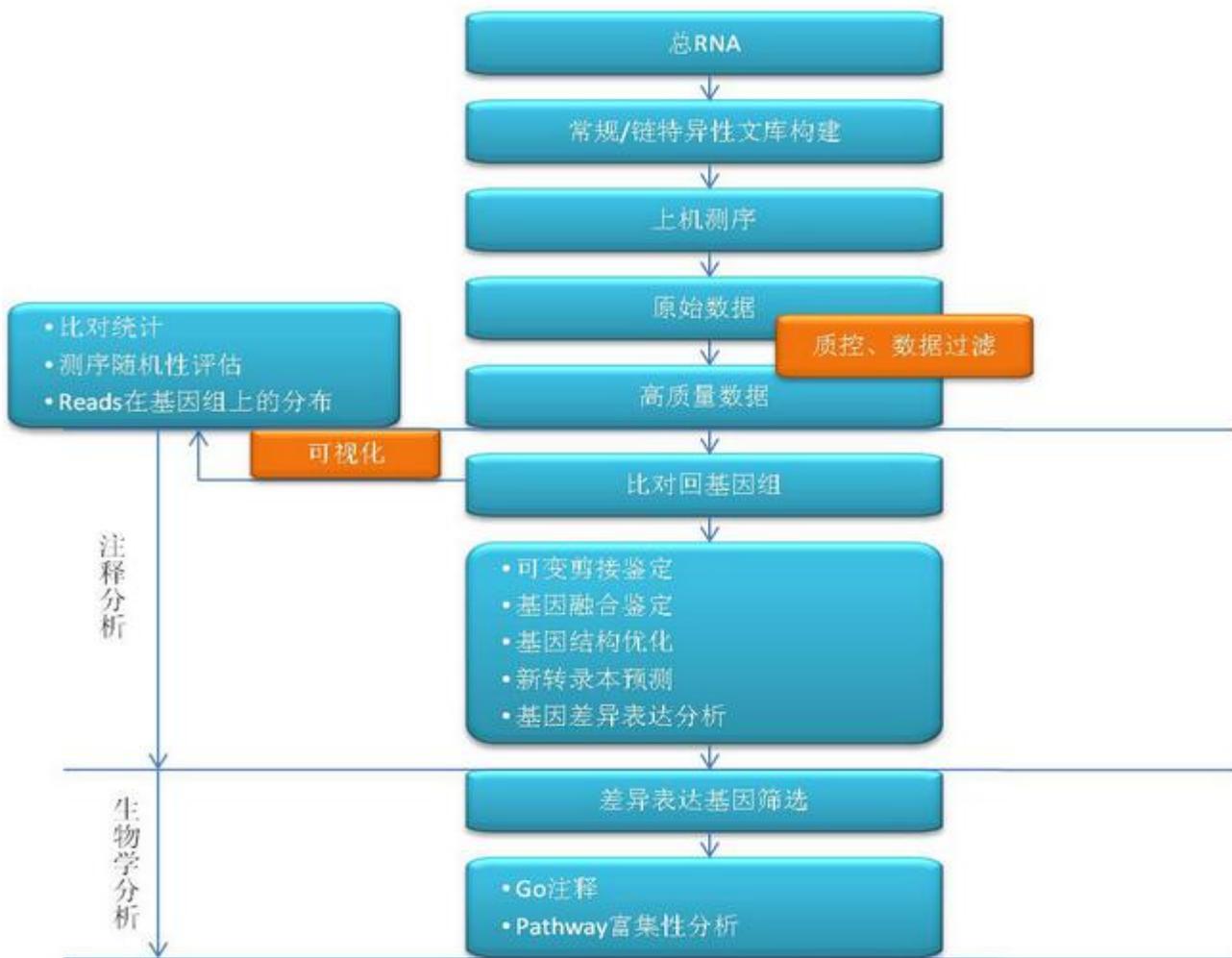
Using **percent = 50** and **cut-off = 30** - This read will not be discarded (the median quality is higher than 30).

Using **percent = 90** and **cut-off = 30** - This read will be discarded (90% of the cycles do not have quality equal to / higher than 30).

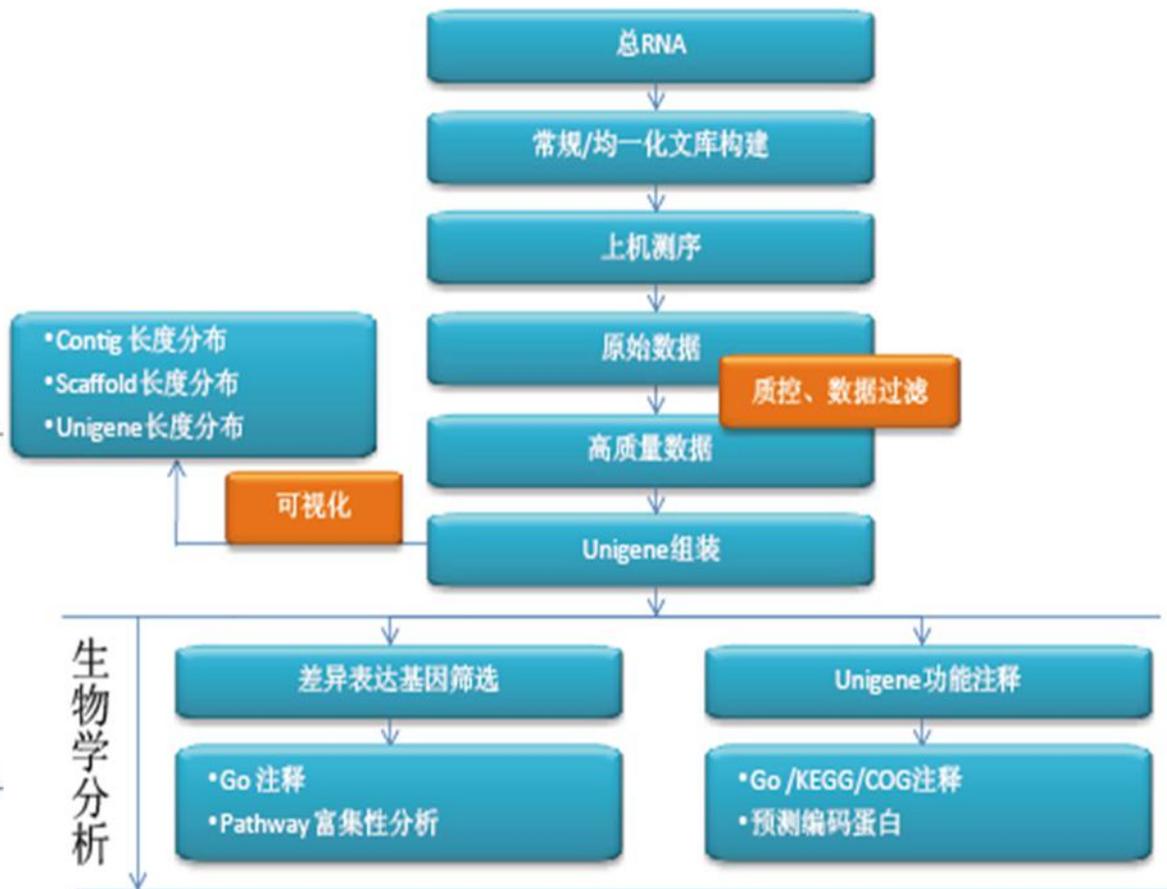
Using **percent = 100** and **cut-off = 20** - This read will be discarded (not all cycles have quality equal to / higher than 20).

转录组数据分析——有参和无参

有参



无参



转录组数据常规流程用到的软件——Bowtie、Tophat、Cufflinks、CummeRbund



Bowtie

Extremely fast, general purpose short read aligner



TopHat

Aligns RNA-Seq reads to the genome using Bowtie
Discovers splice sites



CummeRbund

Plots abundance and differential expression results from Cuffdiff



Cufflinks package

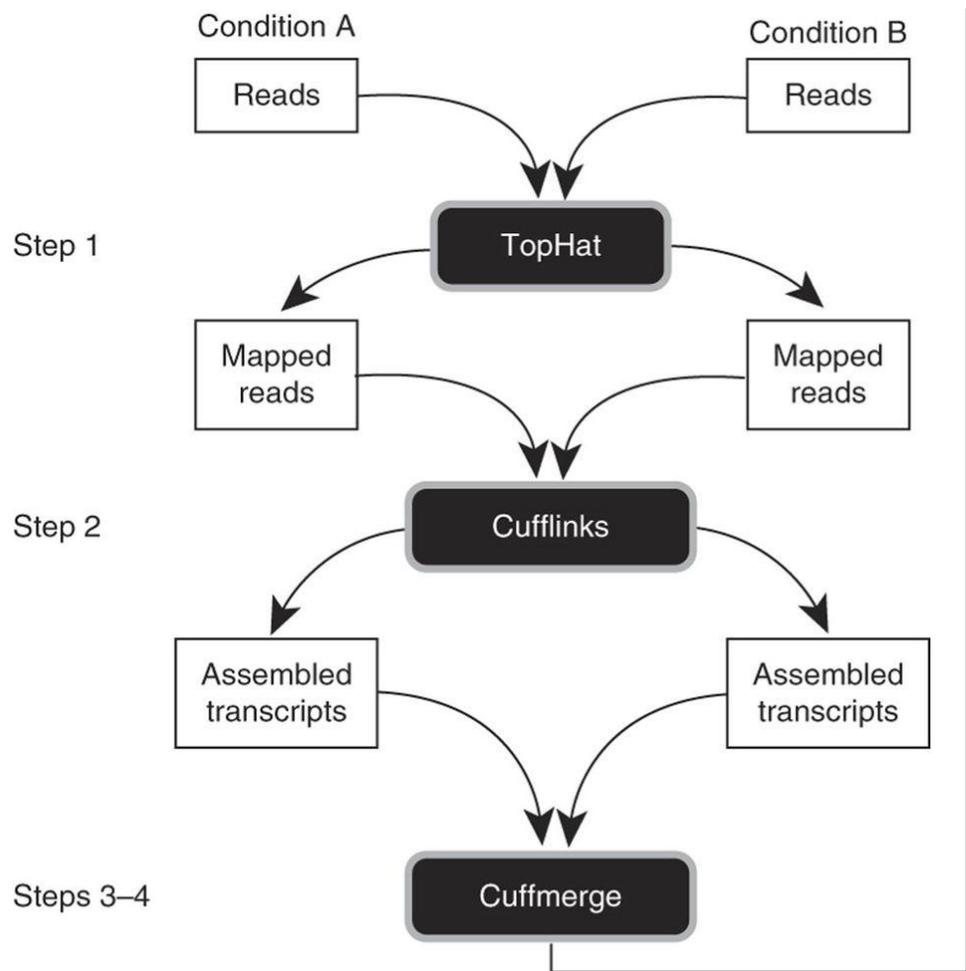
Cufflinks
Assembles transcripts

Cuffcompare
Compares transcript assemblies to annotation

Cuffmerge
Merges two or more transcript assemblies

Cuffdiff
Finds differentially expressed genes and transcripts
Detects differential splicing and promoter use

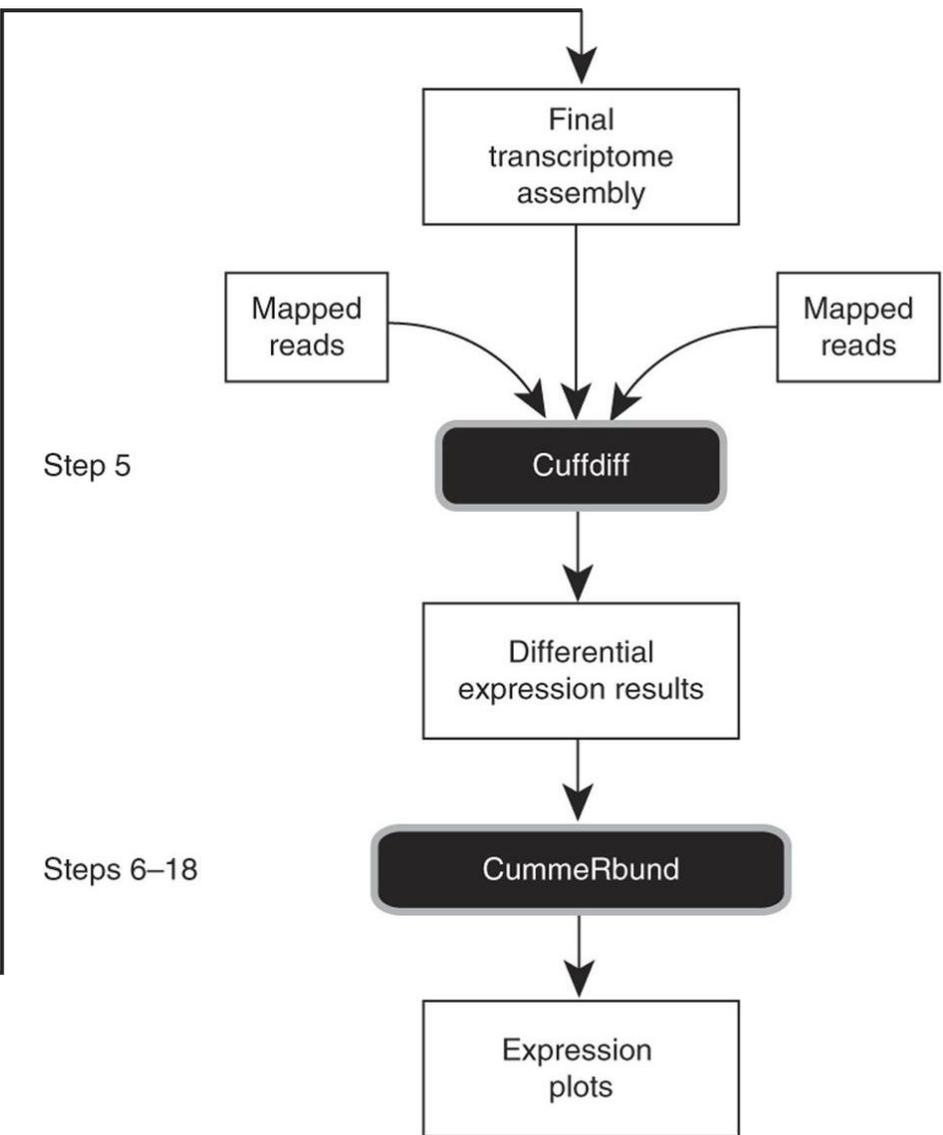
转录组数据分析流程——Tophat、Cufflinks、Cuffdiff、CummeRbund



Step 1

Step 2

Steps 3-4



Step 5

Steps 6-18

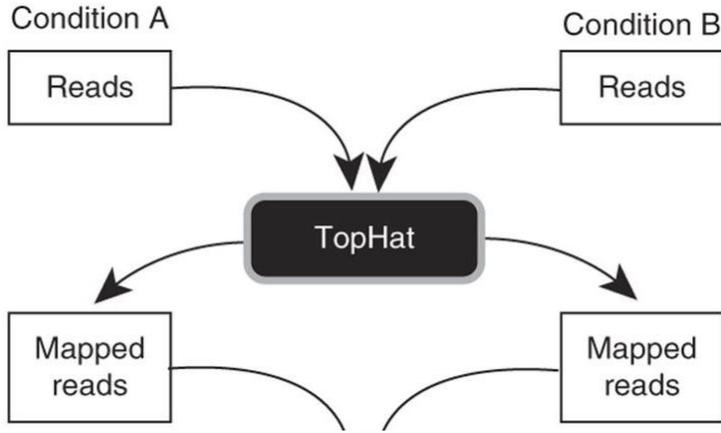
1. Read alignment with **TopHat**

2. Transcript assembly with **Cufflinks**

3. Differential analysis with **Cuffdiff**

4. Visualization with **CummeRbund**

序列比对——Tophat



输出文件

accepted_hits.bam
align_summary.txt
deletions.bed
insertions.bed
junctions.bed
logs
prep_reads.info
unmapped.bam

genes.gtf为基因组注释文件
.fq为测序文件

建bowtie index库
bowtie-build genome.fa
genome

1 | Map the reads for each sample to the reference genome:

```
$ tophat -p 8 -G genes.gtf -o C1_R1_thout genome C1_R1_1.fq C1_R1_2.fq
$ tophat -p 8 -G genes.gtf -o C1_R2_thout genome C1_R2_1.fq C1_R2_2.fq
$ tophat -p 8 -G genes.gtf -o C1_R3_thout genome C1_R3_1.fq C1_R3_2.fq
$ tophat -p 8 -G genes.gtf -o C2_R1_thout genome C2_R1_1.fq C2_R1_2.fq
$ tophat -p 8 -G genes.gtf -o C2_R2_thout genome C2_R2_1.fq C2_R2_2.fq
$ tophat -p 8 -G genes.gtf -o C2_R3_thout genome C2_R3_1.fq C2_R3_2.fq
```

知识点3——什么是gtf格式

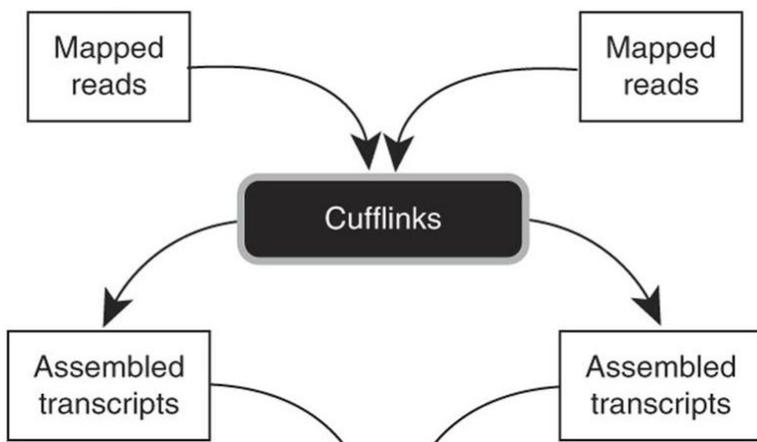
#gtf(General Transfer Format) format

```
1 Ensembl gene 11869 14409 . + . gene_id "ENSG00000223972";
```

#Details

1. seqname - name of the chromosome or scaffold.
2. source - name of the program that generated this feature, or the data source
3. feature - feature type name, e.g. Gene, Variation, Similarity
4. start - Start position of the feature, with sequence numbering starting at 1.
5. end - End position of the feature, with sequence numbering starting at 1.
6. score - A floating point value.
7. strand - defined as + (forward) or - (reverse).
8. frame - One of '0', '1' or '2'.
9. attribute - A semicolon-separated list of tag-value pairs, providing additional information about each feature.

转录本组装——Cufflinks



2 | Assemble transcripts for each sample:

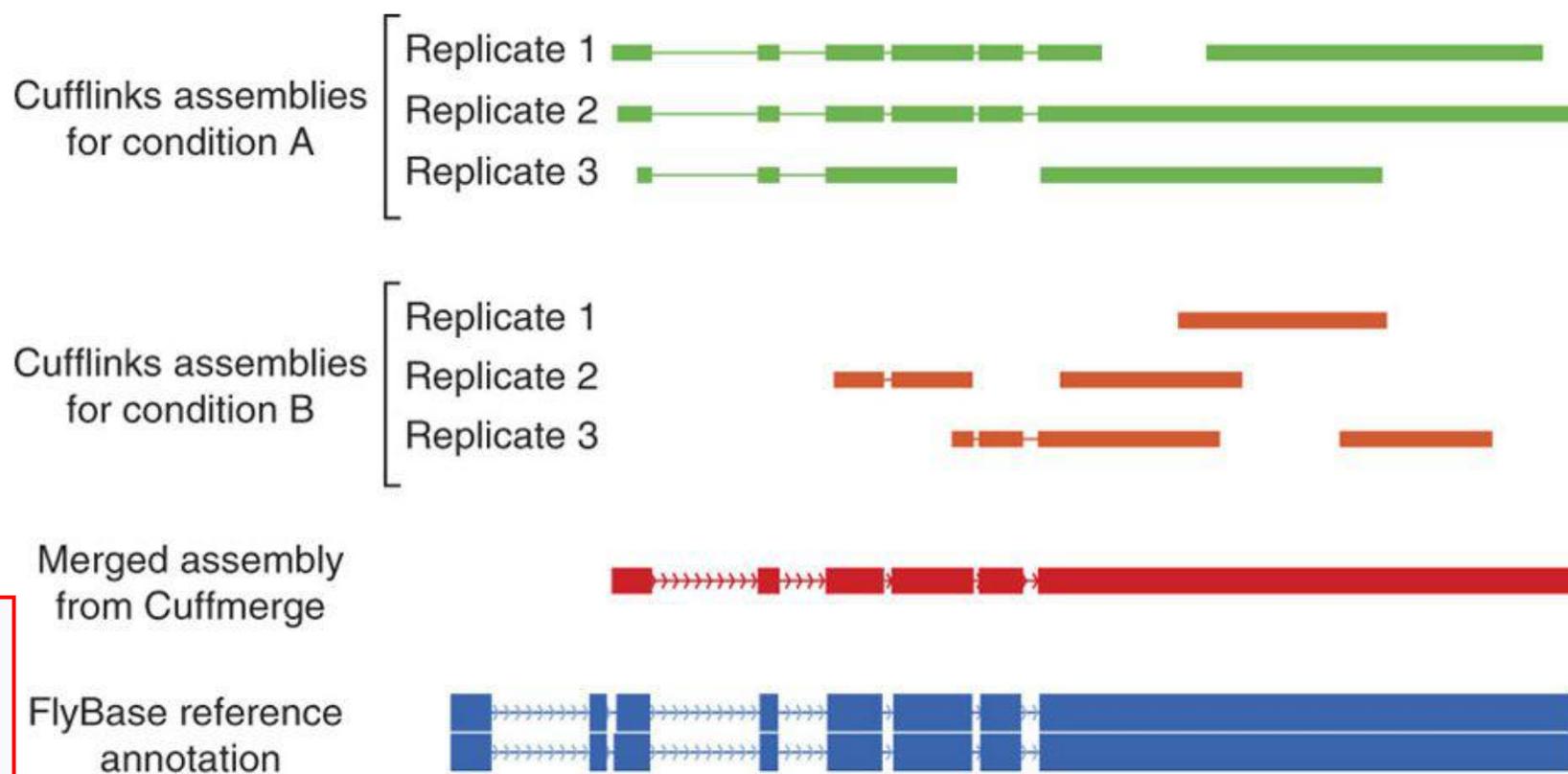
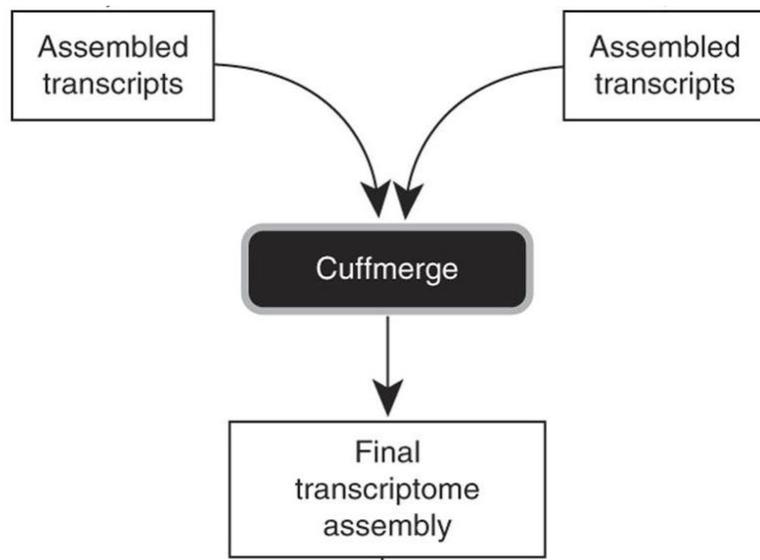
```
$ cufflinks -p 8 -o C1_R1_clout C1_R1_thout/accepted_hits.bam  
$ cufflinks -p 8 -o C1_R2_clout C1_R2_thout/accepted_hits.bam  
$ cufflinks -p 8 -o C1_R3_clout C1_R3_thout/accepted_hits.bam  
$ cufflinks -p 8 -o C2_R1_clout C2_R1_thout/accepted_hits.bam  
$ cufflinks -p 8 -o C2_R2_clout C2_R2_thout/accepted_hits.bam  
$ cufflinks -p 8 -o C2_R3_clout C2_R3_thout/accepted_hits.bam
```

输出文件

genes.fpk_tracking
isoforms.fpk_tracking
skipped.gtf
transcripts.gtf

```
2L> Cufflinks exon 73485 73692 1000 + . gene_id "CUFF.7"; transcript_id "CUFF.7.2"; exon_number "1";  
2L> Cufflinks exon 74903 75018 1000 + . gene_id "CUFF.7"; transcript_id "CUFF.7.2"; exon_number "2";  
2L> Cufflinks exon 75078 76210 1000 + . gene_id "CUFF.7"; transcript_id "CUFF.7.2"; exon_number "3";  
2L> Cufflinks transcript 74485 76210 310 + . gene_id "CUFF.7"; transcript_id "CUFF.7.3"; FPKM "9.24272";  
2L> Cufflinks exon 74485 74572 310 + . gene_id "CUFF.7"; transcript_id "CUFF.7.3"; exon_number "1"; FPKM "9.24272";  
2L> Cufflinks exon 74903 75018 310 + . gene_id "CUFF.7"; transcript_id "CUFF.7.3"; exon_number "2"; FPKM "9.24272";  
2L> Cufflinks exon 75078 76210 310 + . gene_id "CUFF.7"; transcript_id "CUFF.7.3"; exon_number "3"; FPKM "9.24272";  
2L> Cufflinks transcript 102382 105332 1000 + . gene_id "CUFF.8"; transcript_id "CUFF.8.1"; FPKM "1";  
2L> Cufflinks exon 102382 102906 1000 + . gene_id "CUFF.8"; transcript_id "CUFF.8.1"; exon_number "1";  
2L> Cufflinks exon 103006 103434 1000 + . gene_id "CUFF.8"; transcript_id "CUFF.8.1"; exon_number "2";  
2L> Cufflinks exon 103516 105332 1000 + . gene_id "CUFF.8"; transcript_id "CUFF.8.1"; exon_number "3";  
2L> Cufflinks transcript 102382 106710 1000 - . gene_id "CUFF.9"; transcript_id "CUFF.9.1"; FPKM "8";  
2L> Cufflinks exon 102382 104947 1000 - . gene_id "CUFF.9"; transcript_id "CUFF.9.1"; exon_number "1";  
3L> Cufflinks exon 105005 105332 1000 - . gene_id "CUFF.9"; transcript_id "CUFF.9.1"; exon_number "2";
```

转录本组装——Cuffmerge整合Cufflinks组装的转录本

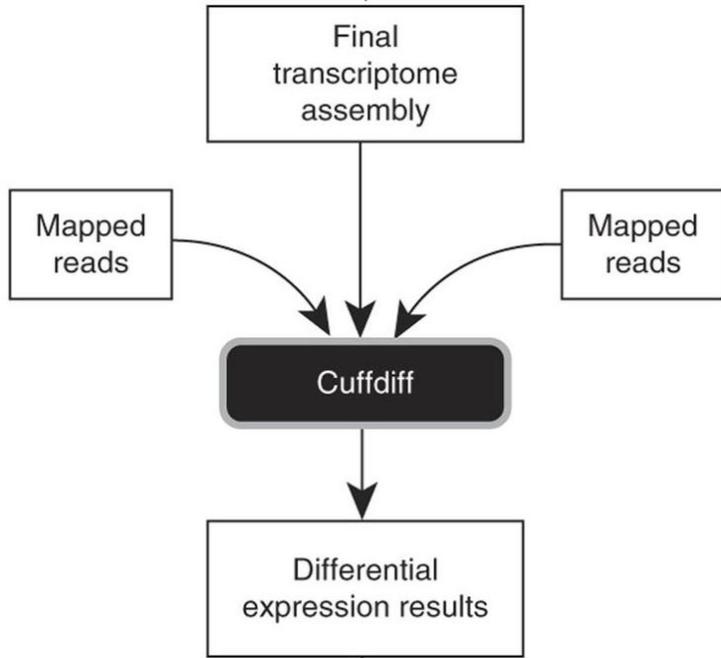


```
./C1_R1_clout/transcripts.gtf  
./C2_R2_clout/transcripts.gtf  
./C1_R2_clout/transcripts.gtf  
./C2_R1_clout/transcripts.gtf  
./C1_R3_clout/transcripts.gtf  
./C2_R3_clout/transcripts.gtf
```

输出结果 merged_asm/merged.gtf

- 3| Create a file called `assemblies.txt` that lists the assembly file for each sample.
- 4| Run Cuffmerge on all your assemblies to create a single merged transcriptome annotation:
`cuffmerge -g genes.gtf -s genome.fa -p 8 assemblies.txt`

差异分析——Cuffdiff



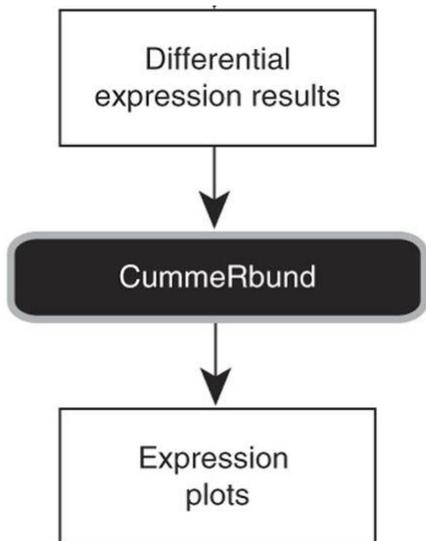
输出文件

```
diff_out/bias_params.info
diff_out/cds.count_tracking
diff_out/cds.diff
diff_out/cds.fpkm_tracking
diff_out/cds.read_group_tracking
diff_out/cds_exp.diff
diff_out/gene_exp.diff
diff_out/genes.count_tracking
diff_out/genes.fpkm_tracking
diff_out/genes.read_group_tracking
diff_out/isoform_exp.diff
diff_out/isoforms.count_tracking
diff_out/isoforms.fpkm_tracking
diff_out/isoforms.read_group_tracking
diff_out/promoters.diff
diff_out/read_groups.info
diff_out/run.info
diff_out/splicing.diff
diff_out/tss_group_exp.diff
diff_out/tss_groups.count_tracking
diff_out/tss_groups.fpkm_tracking
diff_out/tss_groups.read_group_tracking
diff_out/var_model.info
```

5| Run Cuffdiff by using the merged transcriptome assembly along with the BAM files from TopHat for each replicate:

```
$ cuffdiff -o diff_out -b genome.fa -p 8 -L C1,C2 -u merged_asm/merged.gtf
./C1_R1_thout/accepted_hits.bam, ./C1_R2_thout/accepted_hits.bam, ./C1_R3_thout/accepted_hits.bam
./C2_R1_thout/accepted_hits.bam, ./C2_R3_thout/accepted_hits.bam, ./C2_R2_thout/accepted_hits.bam
```

可视化——CummeRbund

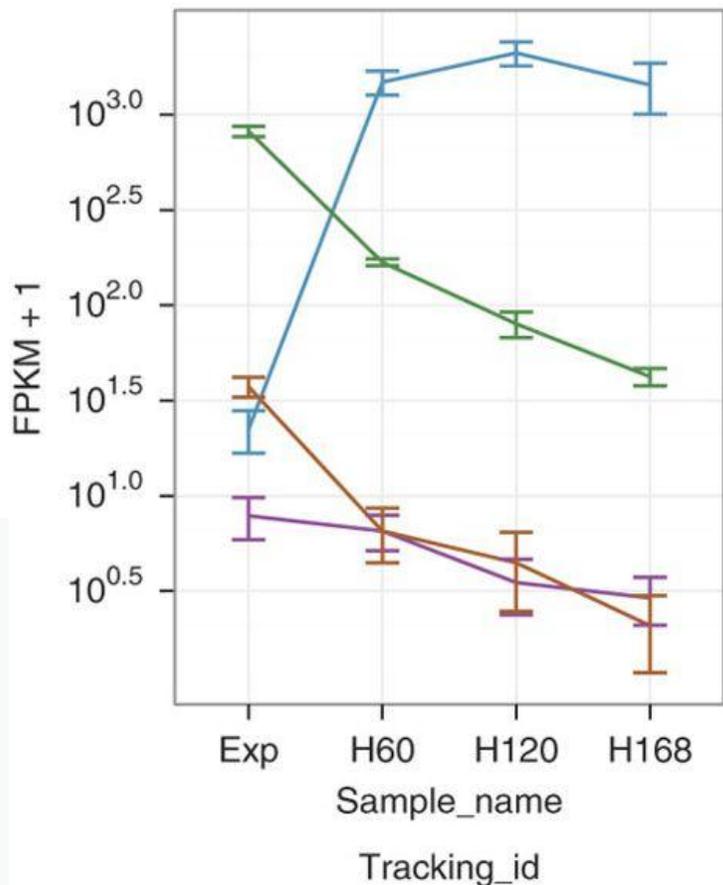


```
library(cummeRbund)
cuff_data <- readCufflinks('diff_out')
csDensity(genes(cuff_data))
mygene <- getGene(cuff_data, 'regucalcin')
expressionBarplot(mygene)
```

```
expressionPlot(isoforms(tpn1), logMode=T)
```

```
sig_genes <- getGenes(cd, geneIdList)
csHeatmap(sig_genes, clustering="row", labRow=F)
```

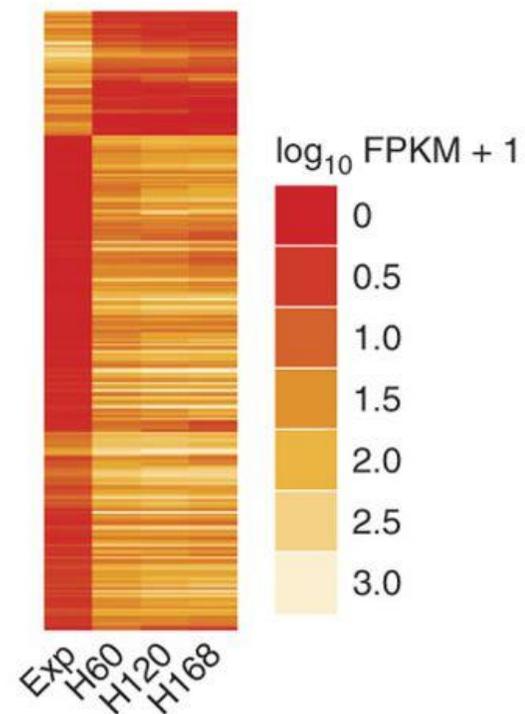
a `expressionPlot(isoforms(tpn1), logMode=T)`



— AK002271.1 — AK077713.1
— AK032942.1 — NM_024427.4

b

```
sig_genes <- getGenes(cd, geneIdList)
csHeatmap(sig_genes,
           clustering="row",
           labRow=F)
```



GO功能分类



GO (Gene Ontology , 基因本体论) 数据库将基因功能分为三大类 :

分子功能(Molecular Function)

生物学过程 (Biological Process)

细胞组分 (Cellular Component)。

在数据分析中，研究者可以找出哪些变化基因属于一个共同的**GO**功能分支，并用统计学方法检定结果是否具有统计学意义，从而得出一组基因主要参与了哪些生物功能。

1. Select analysis tool:

Singular Enrichment Analysis (SEA) ✓

Parametric Analysis of Gene Set Enrichment (PAGE)

Transfer IDs by BLAST (BLAST4ID)

Cross comparison of SEA (SEACOMPARE)

Customized comparison

Reduce + Visual Gene Ontology (REVIGO)

SEA is a traditional and widely used method. User only needs to prepare a list of gene/probe names, and enrichment GO terms will be found out after statistical test from pre-calculated background or customized one.

2. Select the species:

Supported species

Customized annotation

Arabidopsis thaliana

Query list [Example]

```
<allowed ID types in Arabidopsis>
TAIR locus ID: AT3G34250
TAIR alias ID: ACL3
TAIR transcript ID: AT3G34250.1
GenBank ID: AAP50233.1
DBJ ID: BAB11514.1
EMBL ID: CAA18188.1
UniProt ID: Q9LYA9
RefSeq Peptide ID: NP_564434
PDB ID: 2zsi_B
Affymetrix ATH1 Genome Array (GPL198): 267636_at
Affymetrix 8K Genome Array (GPL71): 16876_at
Operon Array v3, Meyerowitz (GPL2810): A021028_01
Agilent 3 Oligo Microarray (GPL2871): A_84_P69894
OAR27K array, Yale University (GPL988): 3243009
```

No your ID? Try **BLAST4ID**

3. Select reference:

Suggested backgrounds

Customized reference [Example]

Customized annotated reference

Arabidopsis genemodel (TAIR9)

For each species, suggested backgrounds are provided. These backgrounds are all pre-computed, and are available to download. To those species without a relatively completed GO profile, backgrounds from near organisms are used as suggestion. If you don't like these backgrounds, then you may submit your customized with/without GO annotation.

4. Advanced options (optional):

Submit

Reset

GO功能分类——AGRIGO结果

Analysis Brief Summary

Job ID: 286731400 [Useful within 7 days]

Job Name: 286731400

Species: *Arabidopsis thaliana*

GO type: Completed GO

Background/Reference: Affymetrix ATH1 Genome Array (GPL198)

Annotated number in query list: 168 [Download]

Annotated number in background/reference: 22479

Significant GO terms: 20 [Details]

Graphical Results

Select Category

Biological Process Cellular Component Molecular Function

Advanced Parameter Settings

Graphic result format: PNG PDF JPEG GIF SVG

Graph rank direction: Top to Bottom Left to Right Bottom to Top Right to Left

Graph font size (pt): 7 8 9 10 11 12

Generate Image

GO flash Chart

Select Category

Biological Process Cellular Component Molecular Function

Advanced Parameter Settings

Bar style: Glass Bar Filled Bar 3D Bar Cylinder Bar

Query bar color: #1A8CFF Bg/Ref bar color: #66FF33 [HEX format only] [default]

X legend content: GO annotation GO accession font: 14

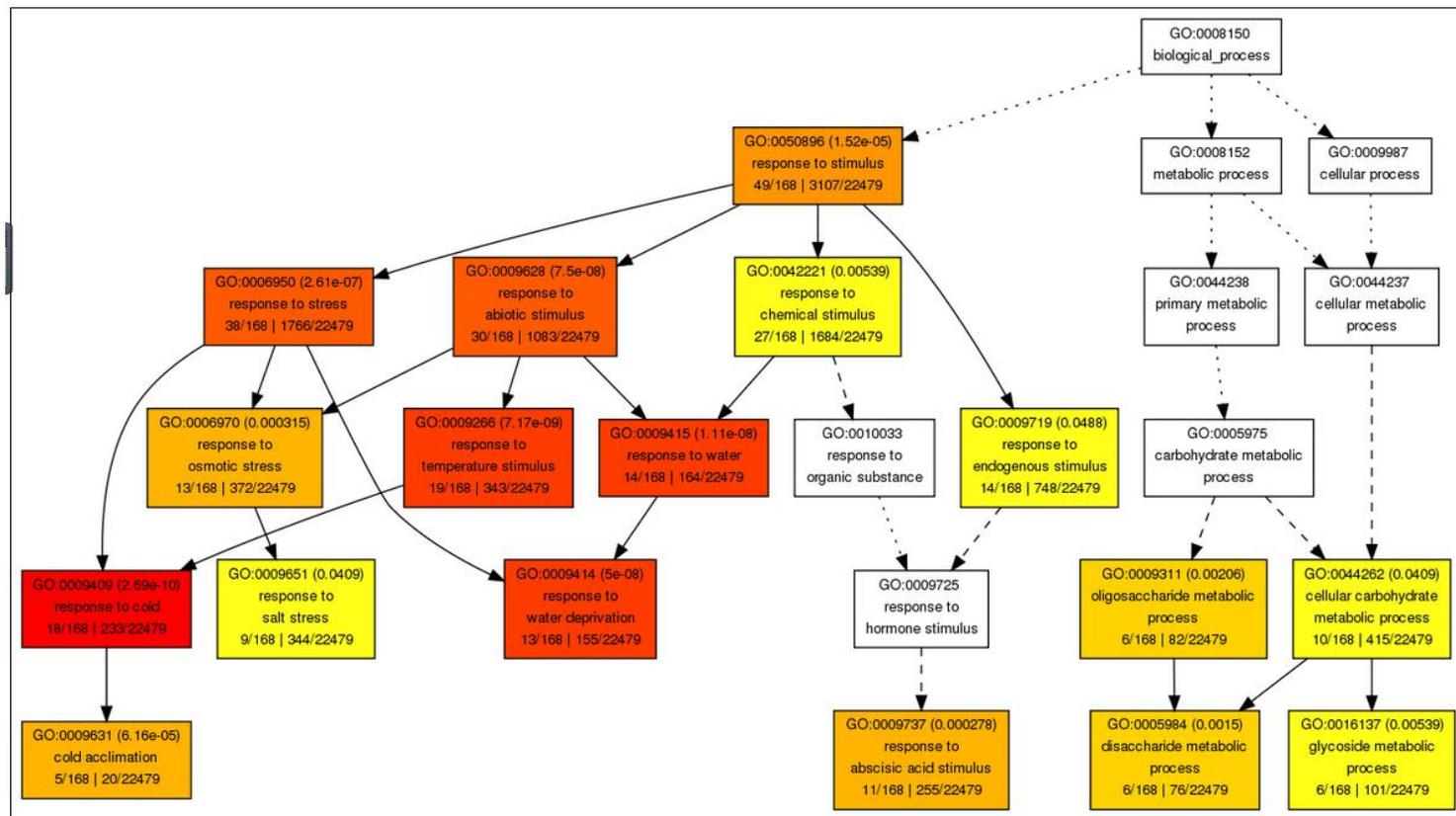
X legend rotation: 300 [270 to 315 is suggested]

Generate Chart

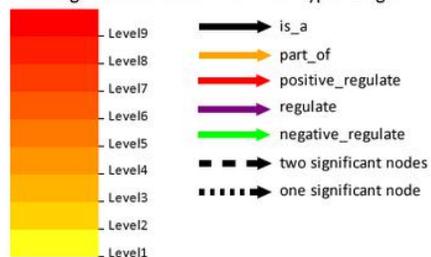
Detail information

You can [Browse in tree traversing mode] [Browse all GO terms] [Download] [REVIGO]

Or select from following significant terms to [Draw graphical results] [Create bar chart]



Significance levels and Arrow types Diagram



Detail information

You can [Browse in tree traversing mode] [Browse all GO terms] [Download] [REVIGO]

Or select from following significant terms to [Draw graphical results] [Create bar chart]



KEGG

Search

Help

» Japanese

全称京都基因与基因组百科全书。它是关于基因、蛋白质、生化反应以及通路的综合生物信息数据库。由多个子库构成。

这些子库中，KEGG PATHWAY 数据库包含了大量物种的代谢与生物信号传导通路信息。

Pathway 数据库下又分为7个部分：1)

Metabolism，2) Genetic Information

Processing，3) Environmental Information

Processing，4) Cellular Processes，5)

Organismal Systems，6) Human Diseases，7)

Drug Development。

KEGG Home
Release notes
Current statistics
Plea from KEGG

KEGG Database
KEGG overview
Searching KEGG
KEGG mapping
Color codes

KEGG Objects
Pathway maps
Brite hierarchies
KEGG DB links

KEGG Software
KEGG API
KGML

KEGG FTP
Subscription

GenomeNet

DBGET/LinkDB

Feedback
Copyright request

Kanehisa Labs

KEGG: Kyoto Encyclopedia of Genes and Genomes

KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies. See Release notes (October 15, 2018) for new and updated features.

Article in 2019 NAR DB Issue
New approach for understanding genome variations in KEGG

Main entry point to the KEGG web service
KEGG2 KEGG Table of Contents [Update notes | Release history]

Data-oriented entry points

KEGG PATHWAY	KEGG pathway maps
KEGG BRITE	BRITE hierarchies and tables
KEGG MODULE	KEGG modules
KEGG ORTHOLOGY	KO functional orthologs [Annotation]
KEGG GENOME	Genomes [Pathogen Virus Plant]
KEGG GENES	Genes and proteins [SeqData]
KEGG COMPOUND	Small molecules
KEGG GLYCAN	Glycans
KEGG REACTION	Biochemical reactions [RModule]
KEGG ENZYME	Enzyme nomenclature
KEGG NETWORK	Disease-related network elements
KEGG DISEASE	Human diseases [Cancer]
KEGG DRUG	Drugs [New drug approvals]

KEGG MEDICUS Health information resource [Drug labels search]

Organism-specific entry points
KEGG Organisms Enter org code(s) hsa hsa eco

Analysis tools

KEGG Mapper	KEGG PATHWAY/BRITE/MODULE mapping tools
BlastKOALA	Genome annotation and KEGG mapping
GhostKOALA	Metagenome annotation and KEGG mapping
BLAST/FASTA	Sequence similarity search
SIMCOMP	Chemical structure similarity search

Classification
Pathway
Brite
Brite table
Module
KO (Function)
Organism
Compound
Network
Disease (ICD)
Drug (ATC)
Drug (Target)

KEGG分析——KOBAS界面

KOBAS 3.0 [Home](#) [Annotate](#) [Gene-list Enrichment](#) [Exp-data Enrichment](#) [Download](#) [Help](#)

Input

Type: [?](#)

Species: [?](#)

Fasta Protein Sequence Fasta Nucleotide Sequence Tabular BLAST Output Entrez Gene ID
 UniProtKB AC Refseq Protein ID Ensembl Gene ID

Input key words here, and choose the species below.(Default:human)

Homo sapiens (human) Mus musculus (mouse) Drosophila melanogaster (fruit fly) Arabidopsis thaliana (thale cress) Saccharomyces cerevisiae (budding yeast)
Escherichia coli K-12 MG1655 Caenorhabditis elegans (nematode)

Input **Ensembl Gene ID**, one id in one line:

Or upload a file:
 未选择文件。

Use KO to do orthologous analysis between different species? (This may take a long time.)

Database

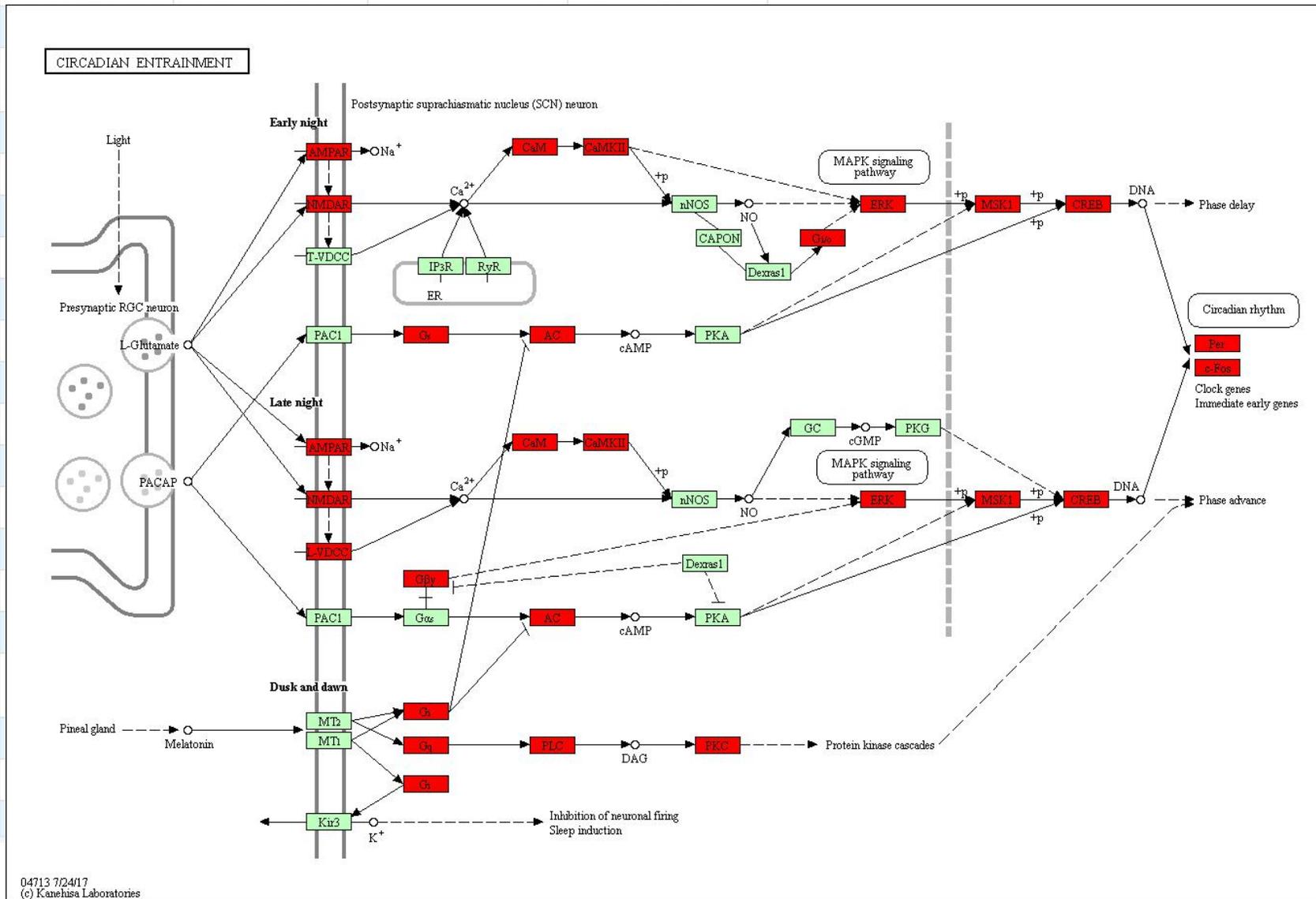
Select your desired databases (Note: the Corrected P-Values will be affected by the number of selected databases)

Pathway	Disease	GO
<input checked="" type="checkbox"/> KEGG Pathway	<input checked="" type="checkbox"/> OMM	<input checked="" type="checkbox"/> GO
<input checked="" type="checkbox"/> Reactome	<input checked="" type="checkbox"/> KEGG Disease	<input checked="" type="checkbox"/> GO Slim
<input checked="" type="checkbox"/> BioCyc	<input checked="" type="checkbox"/> NHGRI GWAS Catalog	
<input checked="" type="checkbox"/> PANTHER		

Copyright © 2005-2017, Center for Bioinformatics, Peking University. All rights reserved.

KEGG分析——KOBAS结果

Term	Database	ID	Input number	Background number	P-Value	Corrected P-Value
Circadian entrainment	KEGG PATHWAY	hsa04713	23	95	1.27808899232e-19	3.15687981103e-17
Dopaminergic synapse	KEGG PATHWAY	hsa04728				
Retrograde endocannabinoid signaling	KEGG PATHWAY	hsa04723				
Pathways in cancer	KEGG PATHWAY	hsa05200				
Amphetamine addiction	KEGG PATHWAY	hsa05031				
Estrogen signaling pathway	KEGG PATHWAY	hsa04915				
cAMP signaling pathway	KEGG PATHWAY	hsa04024				
Serotonergic synapse	KEGG PATHWAY	hsa04726				
Glutamatergic synapse	KEGG PATHWAY	hsa04724				
MAPK signaling pathway	KEGG PATHWAY	hsa04010				
GABAergic synapse	KEGG PATHWAY	hsa04727				
Gap junction	KEGG PATHWAY	hsa04540				
Alzheimer's disease	KEGG PATHWAY	hsa05010				
Cholinergic synapse	KEGG PATHWAY	hsa04725				
HIF-1 signaling pathway	KEGG PATHWAY	hsa04066				
Cocaine addiction	KEGG PATHWAY	hsa05030				
Neuroactive ligand-receptor interaction	KEGG PATHWAY	hsa04080				
HTLV-1 infection	KEGG PATHWAY	hsa05166				
Alcoholism	KEGG PATHWAY	hsa05034				
Oxytocin signaling pathway	KEGG PATHWAY	hsa04921				



谢谢！