Transcriptomic analysis

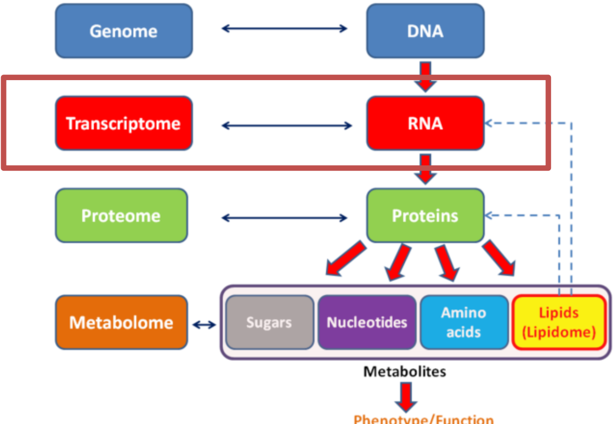
Tao Zhou

2019.1.10

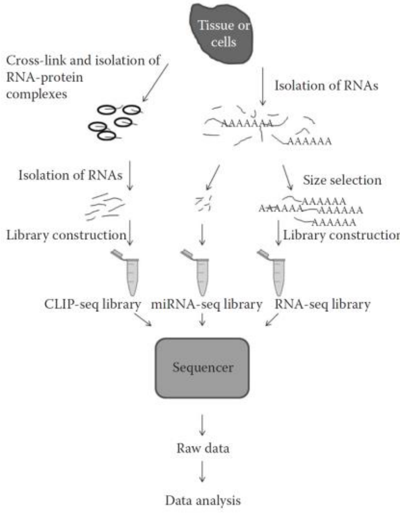
Transcriptome

The transcriptome is the set of all RNA molecules in one cell or a population of cells.

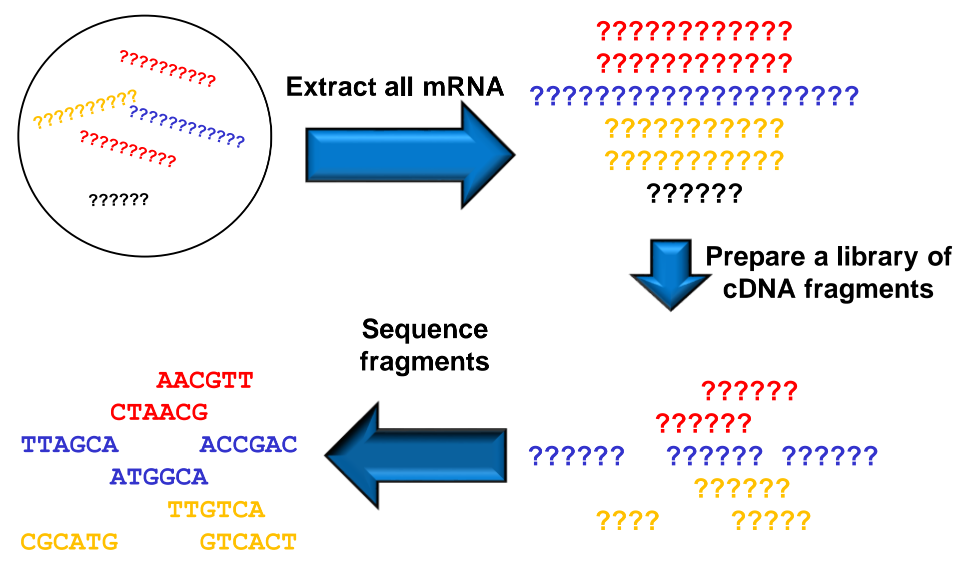
It is sometimes used to refer to all RNAs, or just mRNA, depending on the particular experiment.



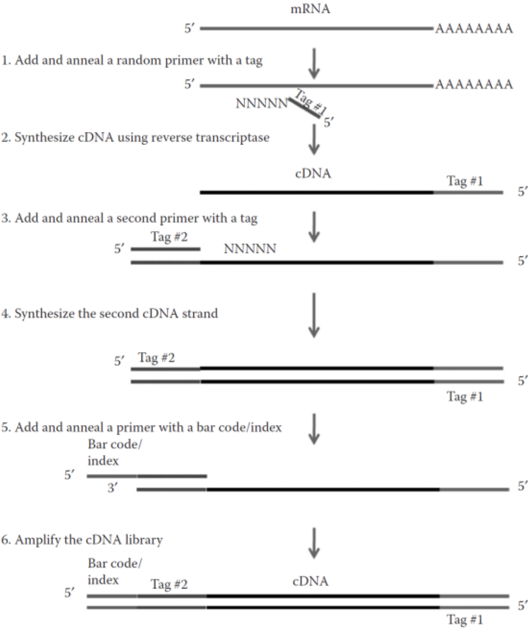
General scheme for RNA-seq experiments



Overview of RNA-seq Procedure



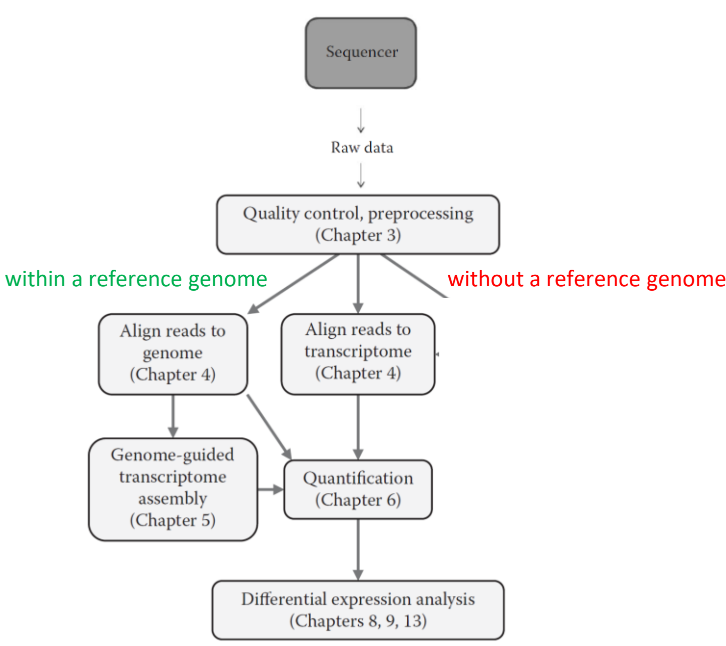
Schematic of RNA-seq library preparation



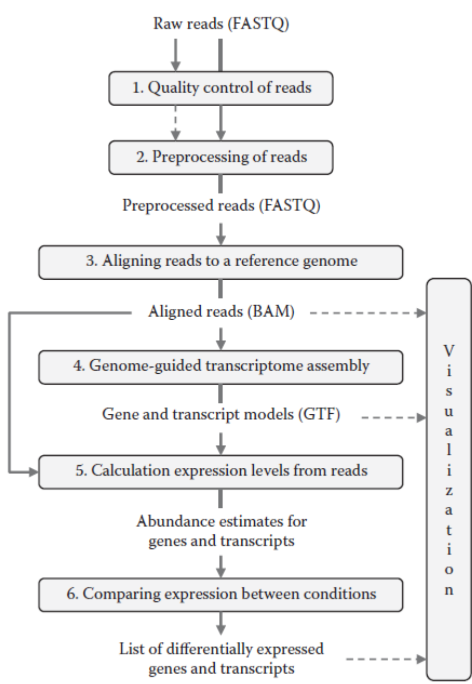
Major RNA-seq Platforms and Their General Properties

Platform	Sequencing Chemistry	Detection Chemistry	Weblink
Illumina	Sequencing by synthesis	Fluorescence	www.illumina.com
SOLID	Sequencing by ligation	Fluorescence	www.invitrogen.com
Roche 454	Sequencing by synthesis	Luminescence	www.454.com
Ion Torrent	Sequencing by synthesis	Proton release	www.iontorrent.com
Pacific biosciences	Single-molecule seq by synthesis	Real-time fluorescence	www.pacificbiosciences.com
Oxford nanopore	Single-molecule seq by synthesis	Electrical current difference per nucleotide through a pore	www.nanoporetech.com

Possible paths in RNA-seq data analysis



Workflow



Quality control and Preprocessing

1.FastQC ——a standalone Java program with a graphical user interface ,and is also easy to use on command line.

```
$ fastqc xx.fastq
```

Ouput: xx_fastqc html (fastqc reports) & xx_fastqc.zip (txt ,html)

Quality control and Preprocessing

2.Trimmomatic —— a versatile Java-based tool for preprocessing reads.(remove adapters and trim reads in different ways based on quality)

```
$ java –jar trimmomatic-0.32.jar PE SampleT12.R1.fastq SampleT12.R2.fastq
T12paired1.fastq T12unpaired1.fastq T12paired2.fastq T12unpaired2.fastq
ILLUMINACLIP:TruSeq3-PE.fa:2:30:10 CROP:107 HEADCROP:2 SLIDINGWINDOW:4:20
MINLEN:50
```

Output: T12paired1.fastq, T12unpaired1.fastq (R1)
T12paired2.fastq, T12unpaired2.fastq (R2)

ILLUMINACLIP: Cut adapter and other illumina-specific sequences from the read.

SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.

LEADING: Cut bases off the start of a read, if below a threshold quality

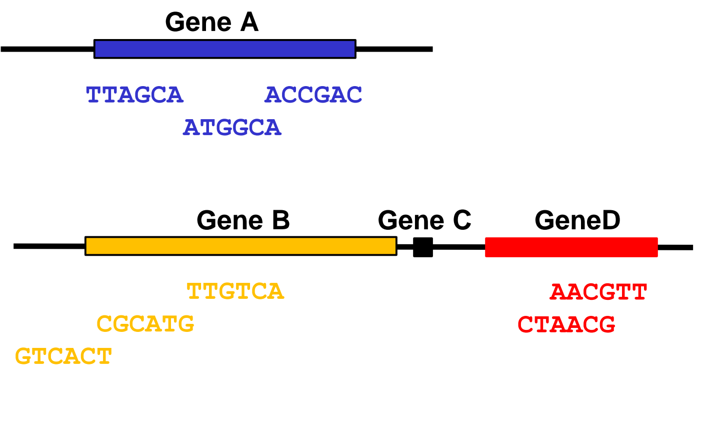
TRAILING: Cut bases off the end of a read, if below a threshold qualityInput: SampleT12.R1.fastq, SampleT12.R2.fastq

CROP: Cut the read to a specified length

HEADCROP: Cut the specified number of bases from the start of the read

MINLEN: Drop the read if it is below a specified length

Align Sequences to a reference (Genome/transcriptome)



For a given gene, the number of reads aligned to the gene measures its expression level.

Gene ID	Sample1
A	3
B	3
C	0
D	2

Aligning Reads to a Reference Genome

1.Downloading a Reference Index

Ensembl database (http://asia.ensembl.org/index.html)

a. download genome FASTA files

$ wget ftp://ftp.ensembl.org/pub/release93/fasta/oreochromis_niloticus/dna/Oreochromis_niloticus.Orenil1.0.dna.toplevel.fa.gz

b. download GTF files (annotations)

$ wget ftp://ftp.ensembl.org/pub/release-93/gtf/oreochromis_niloticus/Oreochromis_niloticus.Orenil1.0.93.gtf.gz

Aligning Reads to a Reference Genome

2. Build reference index--hisat2

$ hisat2-build genome.fa genome

$python3 extract_exons.py Oreochromis_niloticus.Orenil1.0.93.gtf > genome.exon

$ python3 extract_splice_sites.py Oreochromis_niloticus.Orenil1.0.93.gtf > genome.ss

$ python3 extract_snps.py snp142Common.txt > genome.snp (optional)

$ hisat2-build genome.fa --snp genome.snp --ss genome.ss --exon genome.exon genome_snp_tran

III. Aligning reads to reference

$ hisat2 --dta -x ./grch38_tran/genome_snp_tran -1 T12paired1.fastq -2 T12paired2.fastq –S SRR534293.sam

$ hisat2 --dta-cufflinks -x ./grch38_tran/genome_snp_tran -1 T12paired1.fastq -2 T12paired2.fastq –S SRR534293.sam (tailored for cufflinks)

Denovo assembly(transcpritome)

Software: Trinity
Trinity consists of three separate programs:

- (1) Inchworm, which constructs initial contigs,

- (2) Chrysalis, which clusters the contigs produced by Inchworm and creates a de Bruijn graph for each locus, and

- (3)Butterfly, which extracts the isoforms within each de Bruijn graph. The word “component” is used instead of locus in Trinity. During the Butterfly step, it is possible that a component produced by Chrysalis will be divided into smaller pieces if it seems that sequence reads come from more than one locus. If this happens, it is reported in the names of the output transcript sequences.

Then you can annotate the unigenes by blasting against NCBI database etc.

Trinity usage

$ Trinity.pl --seqType fq --JM 10G --left chr18_1.fq --right chr18_2.fq --CPU 4

```
-rw------- 1 somervuo    2.2M  Dec 18 15:13 Trinity.fasta
-rw------- 1 somervuo     583  Dec 18 15:13 Trinity.timing
-rw------- 1 somervuo     78M  Dec 18 14:56 both.fa
-rw------- 1 somervuo       7  Dec 18 14:56 both.fa.read_count
-rw------- 1 somervuo    159M  Dec 18 14:59 bowtie.nameSorted.sam
-rw------- 1 somervuo       0  Dec 18 14:59 bowtie.nameSorted.sam.
  finished
-rw------- 1 somervuo       0  Dec 18 14:59 bowtie.out.finished
drwx------- 3 somervuo   4.0K  Dec 18 15:04 chrysalis
-rw------- 1 somervuo    3.6M  Dec 18 14:58 inchworm.K25.L25.DS.fa
-rw------- 1 somervuo       0  Dec 18 14:58 inchworm.K25.L25.DS.fa.
  finished
-rw------- 1 somervuo       8  Dec 18 14:58 inchworm.kmer_count
-rw------- 1 somervuo   148 K  Dec 18 14:59 iworm_scaffolds.txt
-rw------- 1 somervuo       0  Dec 18 14:59 iworm_scaffolds.txt.
  finished
-rw------- 1 somervuo       0  Dec 18 14:57 jellyfish.1.finished
-rw------- 1 somervuo    125M  Dec 18 14:57 jellyfish.kmers.fa
-rw------- 1 somervuo     13M  Dec 18 14:59 scaffolding_entries.sam
-rw------- 1 somervuo    6.3M  Dec 18 14:59 target.1.ebwt
-rw------- 1 somervuo    279K  Dec 18 14:59 target.2.ebwt
-rw------- 1 somervuo    170K  Dec 18 14:58 target.3.ebwt
-rw------- 1 somervuo    557K  Dec 18 14:58 target.4.ebwt
lrwxrwxrwx 1 somervuo      73  Dec 18 14:58 target.fa ->/.../
  inchworm.K25.L25.DS.fa
-rw------- 1 somervuo       0  Dec 18 14:59 target.fa.finished
-rw------- 1 somervuo    6.3M  Dec 18 14:59 target.rev.1.ebwt
-rw------- 1 somervuo    279K  Dec 18 14:59 target.rev.2.ebwt
```

Trinity.fasta was used to blast against the database for annotation (diamond blast)

Quantitation of gene expression(count, rpkm/fpkm)

1.Counting reads per genes

HTSeq-count - - a part of the HTseq package of Python scripts for NGS data analysis, by default it expects paired-end data to be sorted by read names so that paired reads follow each other in the file

	Union	Intersection-strict	Intersection-nonempty
	✓	✓	✓
	✓	–	✓
	✓	–	✓
	✓	✓	✓
	✓	✓	✓
	?	✓	✓
	?	?	?

Black bar indicates a read,
white box indicates a gene that the read maps to,
grey box indicates another gene which partially overlaps with the white one.

Counting reads per genes

I. Convert SAM to BAM

$ samtools view –bS -o alignments.bam input.sam

II. Sort alignments in BAM by read names (-n)

$ samtools sort –n alignments.bam –o alignments.namesorted

III. htseq-count ounting reads per Genes

$ htseq-count –f bam --stranded=no alignments.namesorted Oreochromis_niloticus.Orenil1.0.93.gtf > counts.txt

-f bam indicates input format is BAM

--stranded=no so that a read is counted also when it maps to the opposite strand

Example Dataset after Aligning Reads

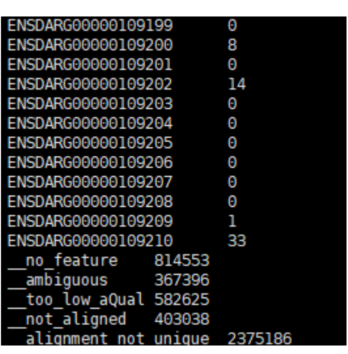
Gene	Treatment 1 (3 replicates)		Treatment 2 (3 replicates)	
1		14	18	10
2		10	3	15
3		1	0	10
4		0	0	0
5		4	3	3
.		.	.	.
.		.	.	.
.		.	.	.
53256	47	29	11	
Total	22910173	30701031	18897029	20546299

I. Combine count files from different samples to a table

$ join counts1.txt counts2.txt> count_table.txt

II.Remove the last five rows prior to stastistical testing for differential expression

$ head –n -5 count_table.txt > genecounts.txt



Quantitation of gene expression

RPKM / FPKM

RPKM, Reads per kilobase of exon per million reads mapped (single-end)

RPKM= counts of mapped reads of a gene/length of a gene

Per Kilobase means length of gene

Per Million is the depth of sequencing

FPKM, Fragments per kilobase of exon per million reads mapped (pair-end)

FPKM= counts of mapped reads of a gene/length of a gene

$$FPKM = \frac{\text{total exon Fragments}}{\text{mapped reads(Millions)} \times \text{exon length(KB)}}$$

When sequencing is single-end, RPKM =FPKM ;

When it is pair-end, we using FPKM. Therefore, FPKM is commonly used.

Counting reads per transcripts-cufflinks-fpkm

$ cufflinks –G Homo_sapiens.GRCh37.74.gtf -b GRCh37.74.
fa -u -p 8 accepted_hits.bam -o outputFolder

outputFolder: 1.genes.fpkm_tracking
2.isoforms.fpkm_tracking
3.skipped.gtf
4.transcripts.gtf

Counting Reads per Exons-DEXSeq

Differential expression can be studied at exon level using the Bioconductor package DEXSeq

- Transcript isoforms tend to have some exons in common, so an exon can appear several times in a GTF file.
- Exons can also overlap with each other if their start/end coordinates differ.
- For counting purposes, we need to construct a set of nonoverlapping exonic regions. The DEXSeq package contains a Python script dexseq_prepare_annotation.py for this task. It “flattens” a GTF file to a list of exon counting bins, which correspond to one exon or a part of an exon (in the case of overlap).

DEXseq usage

1. “flatten” a GTF file to a list of exon counting bins

$python dexseq_prepare_annotation.py Homo_sapiens.GRCh37.74.gtf GRCh37.74_DEX.gtf

2. Count reads per Exons

$python dexseq_count.py –p yes –s no –r name GRCh37.74_DEX.gtf hits_namesorted.sam exon_counts.txt

Differential Expression Analysis

Work in R (edgeR,limma,….)

Software: Rstudio

Unix-like system: Cuffdiff

Transcriptomic analysis(II)

Tao Zhou
2019.1.10

Differential Expression Analysis

1.Differential expression(DE) analysis refers to the identification of genes(or other genomic features, such as, transcripts or exons) that expressed in significantly different quantities in distinct groups of samples, be it biological conditions :

I. drug-treated vs controls

II. diseased vs healthy individuals

III. different tissues

IV. different stages of development

V. etc.

2. Differential expression analysis is typically done on one gene at a time

Gene Name	Sample1 Counts	Sample2 Counts	Sample3 Counts
A (2kb)	10	12	30
B (4kb)	20	25	60
C (1kb)	5	8	15
D (10kb)	0	0	1

RPKM-normalized counts(counts/length of gene)

Gene Name	Sample1 Counts	Sample2 Counts	Sample3 Counts
A (2kb)	1.43	1.33	1.42
B (4kb)	1.43	1.39	1.42
C (1kb)	1.43	1.78	1.42
D (10kb)	0	0	0.009

3. Choice of software

TABLE 8.1 List of (Some) Software Tools for Differential Expression Analysis

Software Tool	Type of Software	Analysis Approach	Comment
DESeq [5]	R/Bioconductor package	Count-based (negative binomial)	Considered conservative (low false-positive rate)
DESeq2 [9]	R/Bioconductor package	Count-based (negative binomial)	Recommended over DESeq by authors; less conservative than DESeq
edgeR [6]	R/Bioconductor package	Count-based (negative binomial)	Similar to DESeq in philosophy
tweeDESeq [7]	R/Bioconductor package	Count-based (Tweedie distribution family)	More general than DESeq/edgeR, but new and not widely tested
Limma [8]	R/Bioconductor package	Linear models on continuous data	Originally developed for microarray analysis, very thoroughly tested. Need to preprocess counts to continuous values
SAMSeq [10] (samr)	R package	Nonparametric test	Adapted from the SAM microarray DE analysis approach. Works better with more replicates
NOISeq [11]	R/Bioconductor package	Nonparametric test	
CuffDiff [18]	Linux command line tool	Isoform deconvolution + count-based tests	Can give differentially expressed isoforms as well as genes (also differential usage of TSS, splice sites)
BitSeq [21]	Linux command line tool and R package	Isoform deconvolution in a Bayesian framework	Can give differentially expressed isoforms. Also calculates (gene and isoform) expression estimates
ebSeq [22]	R/BioConductor package	Isoform deconvolution in a Bayesian framework	Can give differentially expressed isoforms. Can be used in a pipeline preceded by RSEM expression estimation

The DESeq(2), edgeR, and limma packages are all based on the concept of (generalized) linear models

Line Models

More generally, the linear model may be written in matrix form as

$$y = X \cdot \beta + \varepsilon$$

y is again the expression level

ε is an error term

β is a vector of parameters to be estimated from the data

X, describes the experimental factors involved, is called the *design matrix*

For instance, if you have an experiment involving different patients, treatments, and time points, the linear model for each gene could be thought of as

$$y = a + b \cdot \text{treatment} + c \cdot \text{time} + d \cdot \text{patient} + e$$

- y is the gene expression measured in some unit,
- e an error term,
- a, b, c, d are parameters to be estimated from the data. a is called the *intercept* and represents the average expression level of the gene when all the other factors (treatment, time, and patient) are in their reference state.

FACT BOX: HOW TO SELECT A SOFTWARE PACKAGE FOR DIFFERENTIAL EXPRESSION ANALYSIS

Here is a simple decision tree you might use to pick a software package depending on your needs.

Select type of feature to test differential expression for

- Differentially expressed **exons** => *DEXSeq*
- Differentially expressed **isoforms** => *BitSeq, Cuffdiff or ebSeq*
- Differentially expressed genes => **Select type of experimental design**
    - Complex design (more than one varying factor) => *DESeq, edgeR, limma*
    - Simple comparison of groups => **How many biological replicates?**
        - More than about 5 biological replicates per group => *SAMSeq*
        - Less than 5 biological replicates per group => *DESeq, edgeR, limma*

Here we choose two different packages(commonly used) for different expression analysis

1. edgeR package (count-based)

2. Cuffdiff –part of cufflinks package (Isoform deconvolution + count-based tests)

edgeR (count-based)
http://www.360doc.com/content/18/0815/22/57890290_778579575.shtml

Article: A scaling normalization method for differential expression analysis of RNA-seq data

CPM： Counts per million

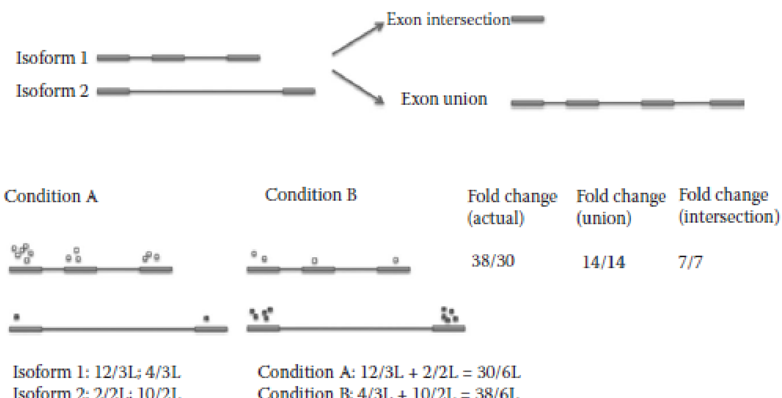
CPM=C/N*1000000

C is mapped reads of gene A (read count)， N is the number of mapped reads of total genes

CPM normalized read counts by the length of gene.

RPKM=CPM/length of gene

Cuffdiff (Isoform deconvolution + count-based tests)



Cuffdiff usage

Cuffdiff command

$cuffdiff -o chr18_hESC_vs_GM12892 -p 4 –L hESC,GM12892 --FDR 0.05 -u Homo_sapiens.GRCh37.59.chr-added.gtf hESC1_chr18.bam,hESC2_chr18.bam,hESC3_chr18.bam,hESC4_chr18.bam Gm12892_1_chr18.bam,Gm12892_2_chr18.bam,Gm12892_3_chr18.bam

Options:

-o output folder name

--FDR gives the false discovery rate cutoff for the DE analysis

−u specifies that we want to use something called “multi-read correction” which is generally recommended

Differently expressed genes(DEGs)

Foldchange(FC)=(B-A)/A

A: a gene expression under condition A

B: the same gene expression under condition B

FDR: false discovery rate

Which are DEGs:

expression of a gene $|log2FC|>1$ & $FDR<0.05$

DEGs results caculated by edgeR

A	B	C	D	E	F	G	H
ID	logFC	logCPM	PValue	FDR	Gene description	Gene name	
ENSONIG00000013737	9.750792144	2.961408366	1.64E-13	8.61E-11			
ENSONIG00000005879	9.440893206	2.626131454	3.81E-09	5.25E-07	claudin e [Source:ZFIN;Acc:ZDB-GENE-010328-	cldne (1 of many)	
ENSONIG00000019271	9.369034559	2.566932917	9.13E-14	5.37E-11	claudin 18 [Source:HGNC Symbol;Acc:HGNC:203	CLDN18 (1 of many)	
ENSONIG00000013741	8.499264141	1.745017715	2.75E-08	2.82E-06	apolipoprotein A-IV a [Source:ZFIN;Acc:ZDB-	apoa4a	
ENSONIG00000005103	8.149795286	1.386459814	7.39E-09	9.27E-07	chitinase, acidic.1 [Source:ZFIN;Acc:ZDB-GE	chia.1 (1 of many)	
ENSONIG00000020020	7.787563423	1.044568177	5.18E-07	3.47E-05	actin binding LIM protein 1 [Source:HGNC Sy	ABLIM1 (1 of many)	
ENSONIG00000013339	7.259935117	5.708696837	1.94E-09	2.97E-07			
ENSONIG00000016677	6.917385593	6.060423706	4.68E-19	9.29E-16	apolipoprotein A-Ib [Source:ZFIN;Acc:ZDB-GE	apoa1b	
ENSONIG00000020902	6.594178586	4.613553927	1.23E-22	6.29E-19	si:ch211-152c8.5 [Source:ZFIN;Acc:ZDB-GENE-	si:ch211-152c8.5 (	
ENSONIG00000009137	6.403179976	5.097637165	7.80E-07	4.87E-05	keratin 5 [Source:ZFIN;Acc:ZDB-GENE-991110-	krt5	
ENSONIG00000019632	6.270210922	4.286988405	9.85E-19	1.67E-15	SRY-box 8 [Source:HGNC Symbol;Acc:HGNC:1120	SOX8 (1 of many)	
ENSONIG00000020064	6.133519497	4.454090234	1.09E-27	1.67E-23			
ENSONIG00000024219	6.120361104	5.102674236	2.95E-07	2.15E-05			
ENSONIG00000018227	6.082469911	2.788464747	9.36E-11	2.38E-08	si:ch73-151m17.5 [Source:ZFIN;Acc:ZDB-GENE-	si:ch73-151m17.5	
ENSONIG00000022297	5.918901135	1.312615994	3.76E-09	5.23E-07			
ENSONIG00000007254	5.878643062	10.7527016	0.000509175	0.008050436	sulfotransferase family 2, cytosolic sulfot	sult2st3	
ENSONIG00000009409	5.852033077	3.480829992	2.10E-06	0.000107464	zgc:123295 [Source:ZFIN;Acc:ZDB-GENE-051127	zgc:123295	
ENSONIG00000000577	5.832665066	2.51984564	7.15E-11	1.88E-08	purine nucleoside phosphorylase 4a [Source:	pnp4a	
ENSONIG00000014454	5.732957834	3.116474052	1.67E-16	2.41E-13	prostaglandin D2 synthase b, tandem duplica	ptgdsb.1	
ENSONIG00000001556	5.68520782	1.866159419	2.94E-08	2.98E-06	si:dkey-201i6.2 [Source:ZFIN;Acc:ZDB-GENE-1	si:dkey-201i6.2	
ENSONIG00000002140	5.548043272	0.980221171	8.40E-08	7.64E-06	elastase 2 like [Source:ZFIN;Acc:ZDB-GENE-0	ela2l	
ENSONIG00000015599	5.42139856	2.491329152	1.97E-06	0.000102384	hydroxysteroid (17-beta) dehydrogenase 12a	hsd17b12a	
ENSONIG00000015922	5.359405903	1.564809198	5.60E-08	5.26E-06	hyaluronan and proteoglycan link protein 1a	hapln1a	

Functional analysis-Gene Ontology

1. About GO

Gene ontology http://www.geneontology.org/

The Gene Ontology (GO) project is a major bioinformatics initiative to develop a computational representation of our evolving knowledge of how genes encode biological functions at the molecular, cellular and tissue system levels. Biological systems are so complex that we need to rely on computers to represent this knowledge.

The project has developed formal ontologies that represent over 40,000 biological concepts, and are constantly being revised to reflect new discoveries. To date, these concepts have been used to "annotate" gene functions based on experiments reported in over 140,000 peer-reviewed scientific papers.

Functional analysis-GO

The Gene Ontology defines the universe of concepts relating to gene functions ('GO terms'), and how these functions are related to each other ('relations'). It is constantly revised and expanded as biological knowledge accumulates.

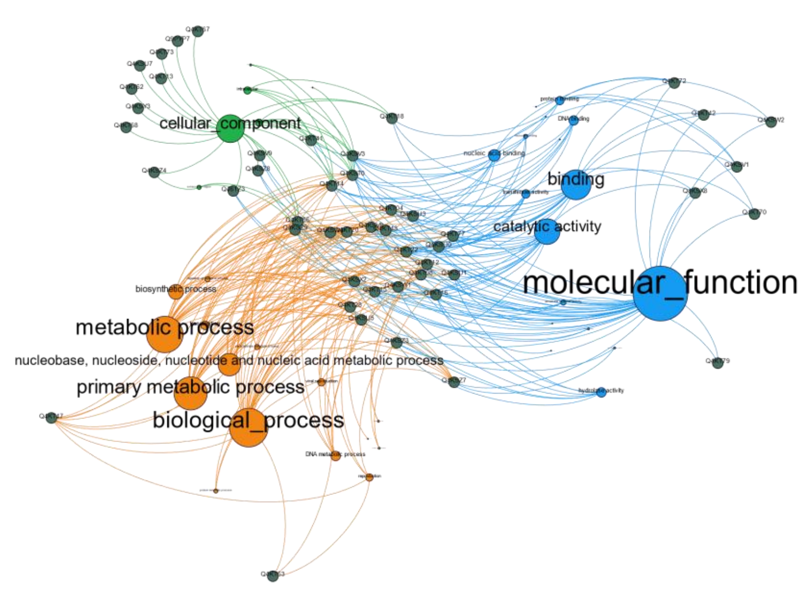
GO describes function with respect to three aspects:

molecular function (molecular-level activities performed by gene products),

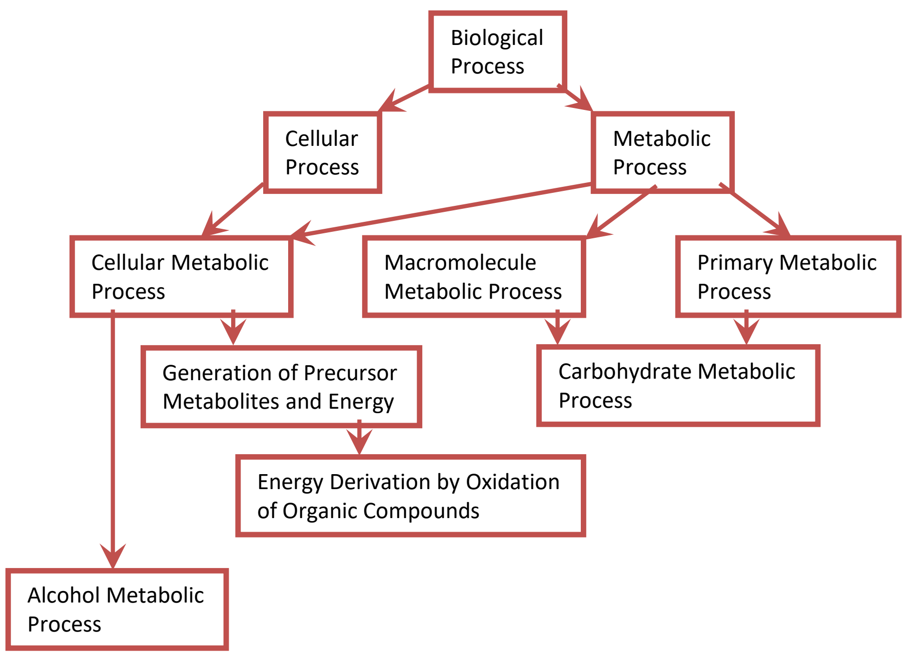
cellular component (the locations relative to cellular structures in which a gene product performs a function),

biological process (the larger processes, or 'biological programs' accomplished by multiple molecular activities).

Example of GO function(network)



Part of the GO Biological Processes DAG



2.GO annotation

I. Ensembl database –biomart

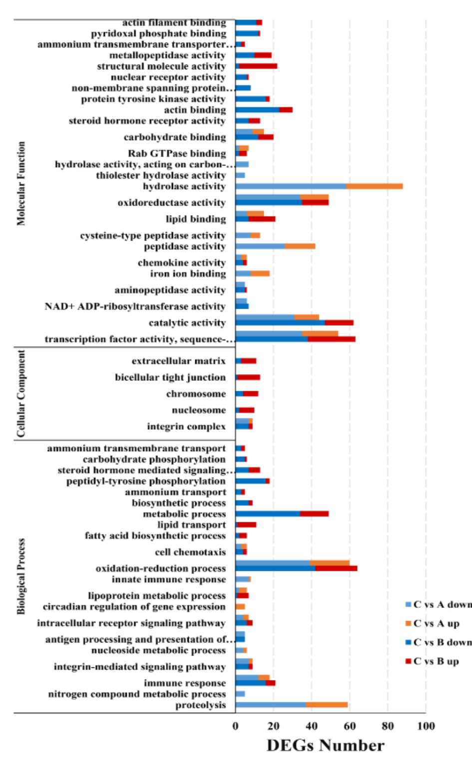
http://www.ensembl.org/biomart/martview/518c52093517bfb6c7ec24437838bded

II. blast against protein database(Swiss-port databse)

Blast2go https://www.cnblogs.com/xiaojikuaipao/p/7190779.html

Gene ontology annotation

- example



Functional analysis-KEGG

1.About KEGG

Kyoto Encyclopedia of Genes and Genomes(KEGG) https://www.kegg.jp/

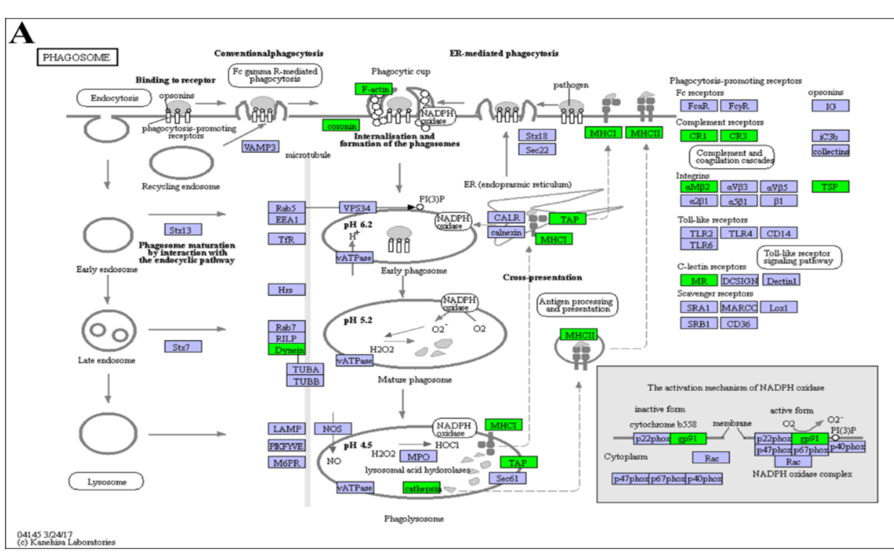
KEGG is a database resource for understanding high-level functions and utilities of the biological system, such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies.

2. KEGG annotation

KAAS - KEGG Automatic Annotation Server

https://www.genome.jp/tools/kaas/

Then mapped to kegg pathway



Function enrichment(GO/KEGG)

1. An enrichment analysis was performed via a hypergeometric test.

The P value was calculated using the following formula:

$$P = 1 - \sum_{i=0}^{m-1} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}},$$

N is the total number of genes annotated to GO term/pathway,

n is the total number of differently expressed genes annotated to GO/pathway,

M is the number of genes annotated to a certain GO term/pathway,

i is the number of differently expressed genes annotated to a certain GO term/pathway.

GO terms/pathway with P value below 0.05 were considered enriched.

All statistical calculations were performed in R.

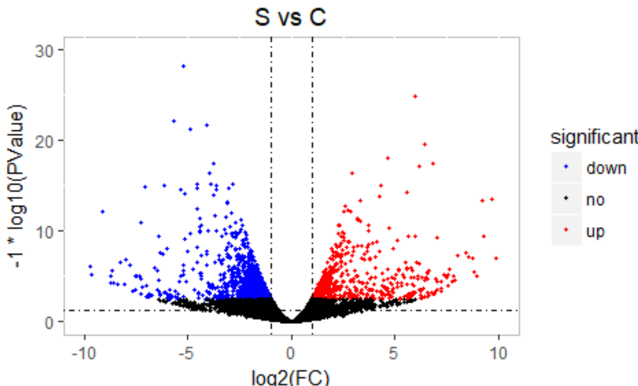
2. Use R package to perform enrichment analysis

Typical example: clusterprofiler package

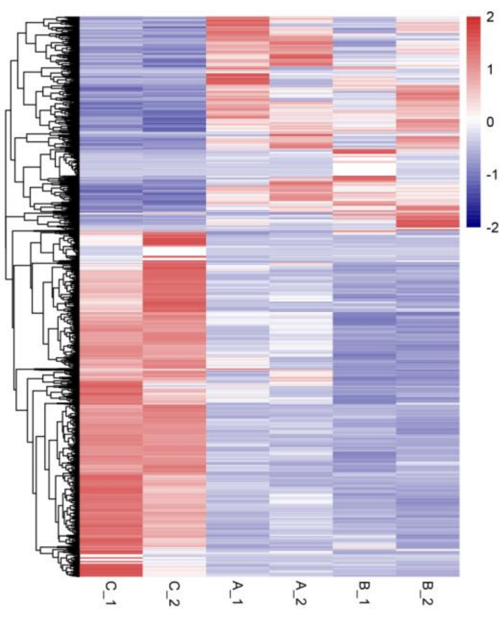
Dependencies: datasets of differently expressed genes (DEGs)

visualization

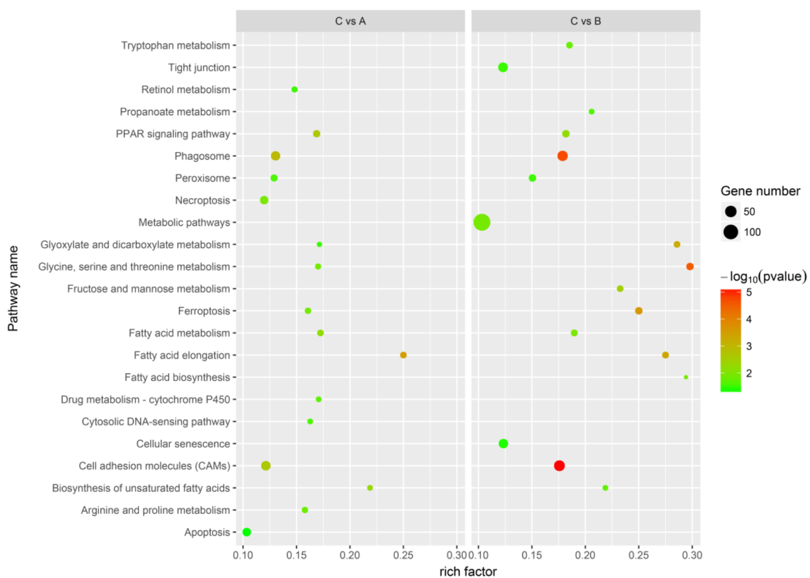
1.Volcano plot-DEGs



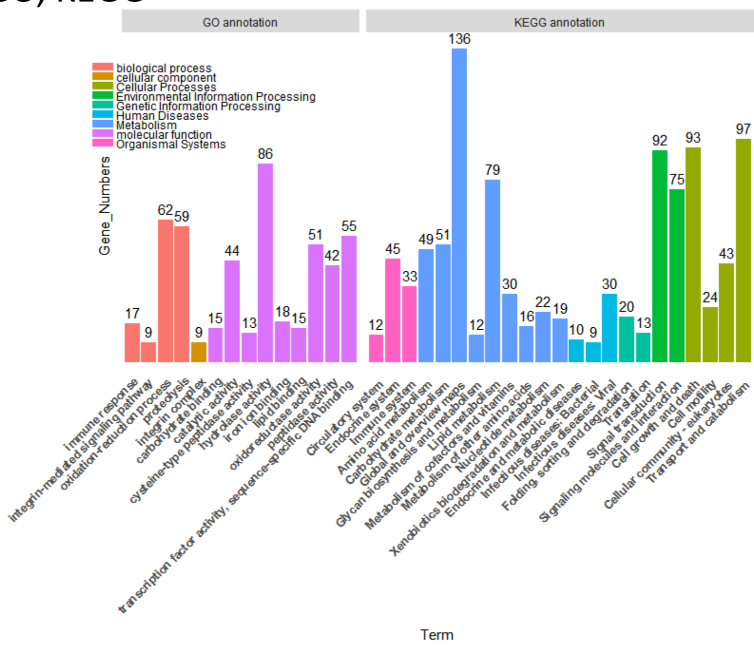
2. Heatmap plot-DEGs



3. Bubble plot-KEGG



4. Barplot - GO, KEGG



5.More…

Please read R Graph Cookbook.pdf

THANK YOU