STATISTICAL METHODS IN BIOINFORMATICS:

Analysis of RNA sequencing data

Stefan E Seemann
ses@sund.ku.dk

Center for non-coding RNA in Technology and Heath (http://rth.dk)
Dep. of Veterinary and Animal Sciences, SUND,
University of Copenhagen DK

31 March 2023

Why sequencing?

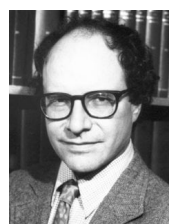
Why sequencing?

- Assemble the genome and transcriptome of a species
- Find genomic variation in a population
- Find genomic and transcriptomic associations with diseases and phenotypes
- Find organisms in environmental sample $\rightarrow$ metagenomics
- Identify potential drug targets $\rightarrow$ personalized medicine
- Tracking of virus variants and mutations $\rightarrow$ vaccine development

$\Rightarrow$ **Understand Molecular Biology**

The beginning

- 1968 — The first 12 bases
- 1973 — 24 bases of the lactose-repressor binding site
  $\rightarrow$ two years of work: one base per month
- 1977 — Sanger sequencing and Gilbert sequencing
  $\rightarrow$ The Nobel Prize in Chemistry 1980 for Frederick Sanger and Walter Gilbert



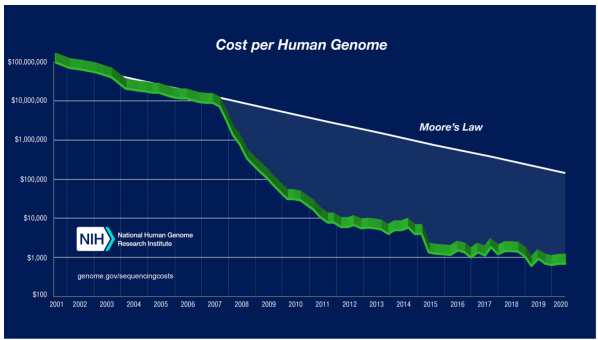
Next generation sequencing (NGS)

(synonym: high-throughput sequencing)

Massively parallelize the sequencing process

2000: Lynx Therapeutics Company launched first NGS technologies → later bought by Illumina

Cost of sequencing a single human genome from 2001 to 2020:



Today's programme

		Alignment, mapping and assembly
0815-0900	Lecture	Alignment methods
0900-1000	Exercise	Dynamical programming of pairwise alignment *on paper*
1000-1030	Lecture	Read mapping
1030-1100	Lecture	Transcriptome assembly
11-12	Lunch	
		Estimation of gene expression and differential expression
1200-1300	Lecture	Expression analysis
1300-1430	Exercise	Expression analysis *in R*
1430-1500		Summary and Discussion

Alignment methods

Where do we need sequence alignments?

Where do we need sequence alignments?

- Sequence similarity
- Gene finding by similarity
- Protein structure by similarity
- RNA structure by similarity
- Motif finder
- Genome and **transcriptome assembly**
- **Gene expression estimation**

Evolutionary events

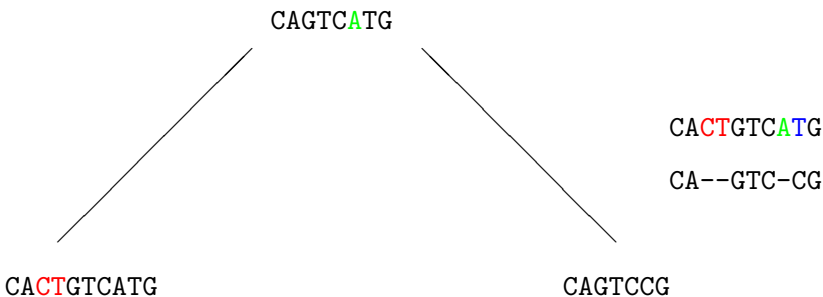
- DNA sequences change in time.
- Find evolutionary related sequences.
- Evolutionary events:

$$\texttt{CAGTCATG} \xrightarrow{\text{Insertion}} \texttt{CACTGTCATG} \xrightarrow{\text{Deletion}} \texttt{CACTGTCTG} \xrightarrow{\text{Substitution}} \texttt{CACTATCTG}$$

$$\texttt{CAGTCATG} \xrightarrow{\text{Duplication}} \texttt{CAGTGTCATG} \xrightarrow{\text{Translocation}} \texttt{CATCAGTGTG} \xrightarrow{\text{Inversion}} \texttt{CATTGACGTG}$$

Evolutionary tree

- Finding common ancestors.
- Parsimony principle: Evolution uses minimum number of operations.



- Probabilistic approaches (max. likelihood or sampling).

Alignments: optimize a score

Score of a given alignment:

```
ACTGTCATG
A--GTC-CG
```

$$S_{\text{tot}} = S\begin{bmatrix}\texttt{A}\\\texttt{A}\end{bmatrix} + S\begin{bmatrix}\texttt{C}\\\texttt{-}\end{bmatrix} + S\begin{bmatrix}\texttt{T}\\\texttt{-}\end{bmatrix} + S\begin{bmatrix}\texttt{G}\\\texttt{G}\end{bmatrix} + S\begin{bmatrix}\texttt{T}\\\texttt{T}\end{bmatrix} + S\begin{bmatrix}\texttt{C}\\\texttt{C}\end{bmatrix} + S\begin{bmatrix}\texttt{A}\\\texttt{-}\end{bmatrix} + S\begin{bmatrix}\texttt{T}\\\texttt{C}\end{bmatrix} + S\begin{bmatrix}\texttt{G}\\\texttt{G}\end{bmatrix}$$

- Score: substituting a residue in one seq. with a residue in another.
- Find the alignment that have the highest score.
- Try out all alignment combinations? (we deal with this soon)
- So speed of sequence comparisons **matters**!

Score matrices for DNA

- Standard DNA matrices look like

	A	C	G	T
A	5	-4	-4	-4
C	-4	5	-4	-4
G	-4	-4	5	-4
T	-4	-4	-4	5

- Exercise construct our own matrix:
  Identity: 8
  Transition: 2 (eg. {A,G} →: {A,G}; purine to purine)
  Transversion: -3 (eg. A→ {C,T}; purine to pyrimidine).

	A	C	G	T
A				
C				
G				
T				

Score matrices for DNA

- Standard DNA matrices look like

	A	C	G	T
A	5	-4	-4	-4
C	-4	5	-4	-4
G	-4	-4	5	-4
T	-4	-4	-4	5

- Exercise construct our own matrix:
  Identity: 8
  Transition: 2 (eg. {A,G} →: {A,G}; purine to purine)
  Transversion: -3 (eg. A→ {C,T}; purine to pyrimidine).

	A	C	G	T
A	8	-3	2	-3
C	-3	8	-3	2
G	2	-3	8	-3
T	-3	2	-3	8

What about gaps?

- Gap cost. Cost of indel (Eg. $d = 10$).
- Initiation and elongation.

Dynamical programming

- Find the alignment between `CACTGTCATG` and `CAGTCTG` that has the maximal score?
- What would be a trivial way?

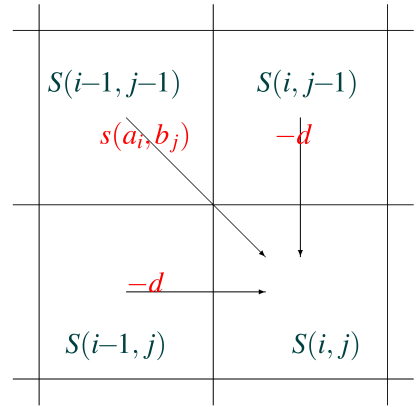
Dynamical programming

- Find the alignment between `CACTGTCATG` and `CAGTCTG` that has the maximal score?
- What would be a trivial way?
- Basic idea: Use sub sequences! $\rightarrow$ **Dynamic Programming**

Pairwise global alignments (Needleman–Wunsch)

Comparing sequences a and b. Given a substitution score $s(x, y)$ of replacing letter x with letter y, the highest scoring alignment can be found by the following recursion:

$$S(i,j) = \max \begin{cases} S(i\text{–}1, j\text{–}1) + s(a_i, b_j) \\ S(i\text{–}1, j) - d \\ S(i, j\text{–}1) - d \end{cases}$$



a_i residue at position i in seq. a
b_j residue at position j in seq. b
$i = 1, \ldots, N$; $j = 1, \ldots, M$

Initialization: $S(0,0) = 0$. Hence: $S(i,0) = -id$, $S(0,j) = -jd$.
Note: the alignment takes time $O(NM)$.

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0										
C											
A											
G											
T											
C											
T											
G											

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10										
A	−20										
G	−30										
T	−40										
C	−50										
T	−60										
G	−70										

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2									
G	−30	−12									
T	−40	−22									
C	−50	−32									
T	−60	−42									
G	−70	−52									

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	-10	-20	-30	-40	-50	-60	-70	-80	-90	-100
C	-10	8	-2	-12	-22	-32	-42	-52	-62	-72	-82
A	-20	-2	16	6	-4	-14	-24	-34	-44	-54	-64
G	-30	-12	6	13	3	4	-6	-16	-26	-36	-46
T	-40	-22	-4	8							
C	-50	-32	-14	4							
T	-60	-42	-24	-6							
G	-70	-52	-34	-16							

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	12	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	20	10	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	18	8
G	−70	−52	−34	−16	2	20	10	23	18	14	26

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	12	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	20	10	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	18	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	12	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	20	10	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	12	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	20	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	12	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	4	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	−4	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	6	**-4**	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	16	**6**	**-4**	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	8	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	**16**	**6**	**-4**	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Example of global alignment:

Align the two sequences CACTGTCATG and CAGTCTG

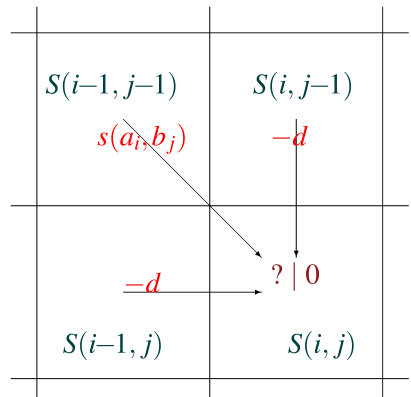
		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	**8**	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	**16**	**6**	**-4**	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42	−24	−6	12	8	26	16	17	**18**	8
G	−70	−52	−34	−16	2	20	10	23	18	14	**26**

Back-tracking ...

Pairwise local alignments (Smith–Waterman)

Comparing sequences a and b. Given a substitution score $s(x, y)$ of replacing letter x with letter y, the highest scoring alignment can be found by the following recursion:

$$S(i,j) = \max \begin{cases} S(i\text{–}1, j\text{–}1) + s(a_i, b_j) \\ S(i\text{–}1, j) - d \\ S(i, j\text{–}1) - d \\ 0 \end{cases}$$



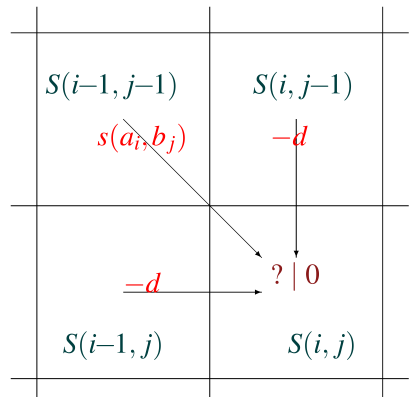
Note only positive numbers!
$i = 1, \ldots, N$; $j = 1, \ldots, M$

Initialization: $S(0,0) = 0$. Hence: $S(i,0) = ?$, $S(0,j) = ?$.

Pairwise local alignments (Smith–Waterman)

Comparing sequences a and b. Given a substitution score $s(x, y)$ of replacing letter x with letter y, the highest scoring alignment can be found by the following recursion:

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



Note only positive numbers!
$i = 1, \ldots, N$; $j = 1, \ldots, M$

Initialization: $S(0,0) = 0$. Hence: $S(i,0) = 0$, $S(0,j) = 0$.

Example of local alignment:

Align the two sequences AAACTGTTTAACAG and AACAGGGGAAACTG.

		A	A	A	C	T	G	T	T	T	A	A	C	A	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0														
A	0														
C	0														
A	0														
G	0														
G	0														
G	0														
G	0														
A	0														
A	0														
A	0														
C	0														
T	0														
G	0														

Example of local alignment:

Align the two sequences AAACTGTTTAACAG and AACAGGGGAAACTG.

		A	A	A	C	T	G	T	T	T	A	A	C	A	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	8	8	8	0	0	2	0	0	0	8	8	0	8	2
A	0	8	16	16	6	0	2	0	0	0	8	16	6	8	10
C	0	0	6	13	24	14	4	4	2	2	0	6	24	14	5
A	0	8	8	14	14	21	16	6	1	0	10	8	14	32	22
G	0	2	10	10	11	11	29	19	9	0	2	12	5	22	40
G	0	2	4	12	7	8	19	26	16	6	2	4	9	12	30
G	0	2	4	6	9	4	16	16	23	13	8	4	1	11	20
G	0	2	4	6	3	6	12	13	13	20	15	10	1	3	19
A	0	8	10	12	3	0	8	9	10	10	28	23	13	9	9
A	0	8	16	18	9	0	2	5	6	7	18	36	26	21	11
A	0	8	16	24	15	6	2	0	2	3	15	26	33	34	24
C	0	0	6	14	32	22	12	4	2	4	5	16	34	30	31
T	0	0	0	4	22	40	30	20	12	10	1	6	24	31	27
G	0	2	2	2	12	30	48	38	28	18	12	3	14	26	39

Example of local alignment:

Align the two sequences AAACTGTTTAACAG and AACAGGGGAAACTG.

		A	A	A	C	T	G	T	T	T	A	A	C	A	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	8	8	8	0	0	2	0	0	0	**8**	8	0	8	2
A	0	8	16	16	6	0	2	0	0	0	8	**16**	6	8	10
C	0	0	6	13	24	14	4	4	2	2	0	6	**24**	14	5
A	0	8	8	14	14	21	16	6	1	0	10	8	14	**32**	22
G	0	2	10	10	11	11	29	19	9	0	2	12	5	22	**40**
G	0	2	4	12	7	8	19	26	16	6	2	4	9	12	30
G	0	2	4	6	9	4	16	16	23	13	8	4	1	11	20
G	0	2	4	6	3	6	12	13	13	20	15	10	1	3	19
A	0	**8**	10	12	3	0	8	9	10	10	28	23	13	9	9
A	0	8	**16**	18	9	0	2	5	6	7	18	36	26	21	11
A	0	8	16	**24**	15	6	2	0	2	3	15	26	33	34	24
C	0	0	6	14	**32**	22	12	4	2	4	5	16	34	30	31
T	0	0	0	4	22	**40**	30	20	12	10	1	6	24	31	27
G	0	2	2	2	12	30	**48**	38	28	18	12	3	14	26	39

Example of local alignment:

Align the two sequences AAACTGTTTAACAG and AACAGGGGAAACTG.

		A	A	A	C	T	G	T	T	T	A	A	C	A	G
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	8	8	8	0	0	2	0	0	0	**8**	8	0	8	2
A	0	8	16	16	6	0	2	0	0	0	8	**16**	6	8	10
C	0	0	6	13	24	14	4	4	2	2	0	6	**24**	14	5
A	0	8	8	14	14	21	16	6	1	0	10	8	14	**32**	22
G	0	2	10	10	11	11	29	19	9	0	2	12	5	22	**40**
G	0	2	4	12	7	8	19	26	16	6	2	4	9	12	30
G	0	2	4	6	9	4	16	16	23	13	8	4	1	11	20
G	0	2	4	6	3	6	12	13	13	20	15	10	1	3	19
A	0	**8**	10	12	3	0	8	9	10	10	28	23	13	9	9
A	0	8	**16**	18	9	0	2	5	6	7	18	36	26	21	11
A	0	8	16	**24**	15	6	2	0	2	3	15	26	33	34	24
C	0	0	6	14	**32**	22	12	4	2	4	5	16	34	30	31
T	0	0	0	4	22	**40**	30	20	12	10	1	6	24	31	27
G	0	2	2	2	12	30	**48**	38	28	18	12	3	14	26	39

Alignment Substitution Matrices

- Alignments based on the cost of substitions and indels (insertion/deletions).
- List the cost of replacing a residue in one sequence with a residue in another.
- Basic idea: comparing log-odds:

$$s(a, b) = \log \frac{p_{ab}}{q_a q_b} = \log \frac{\text{observed frequency}}{\text{expected frequency}}$$

- The 'log' makes the scores additive.

Question: What is the optimal way to choose observed frequencies?

Alignment Substitution Matrices

- Alignments based on the cost of substitions and indels (insertion/deletions).
- List the cost of replacing a residue in one sequence with a residue in another.
- Basic idea: comparing log-odds:

$$s(a,b) = \log \frac{p_{ab}}{q_a q_b} = \log \frac{\text{observed frequency}}{\text{expected frequency}}$$

- The 'log' makes the scores additive.

Question: What is the optimal way to choose observed frequencies?
$\rightarrow$ large database of true alignments

Commonly used scoring matrices:

- BLOSUM: BLOck SUbstitution Matrix
- PAM: Point Accepted Mutation

Alignment score statistics

Question: Given a particular scoring system, how many distinct local alignments with score $\geq$S can one expect to find by chance from the comparison of two random sequences of lengths m and n?

Or in other words, when can a local alignment be considered statistically significant?

E-values and P-values

The expected number of local alignments with a score of at least S is given by the E-value for the score S:

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

1. Doubling the length of the query sequence (m) or the size of the database (n) should double the number of local alignments.
2. E-value decreases exponentially as score S increases.

The probability of observing *at least* one alignment with score $\geq S$

$$p = 1 - e^{-E}$$

$\Rightarrow$ Sequence similarity score S is *extreme value distributed*

---

https://www.ncbi.nlm.nih.gov/BLAST/tutorial/

Time is important

- Dynamic programming: exact in $O(NM)$.
- When becomes time an issue?

Time is important

- Dynamic programming: exact in $O(NM)$.
- When becomes time an issue?
  $\rightarrow$ Target is entire genome
  $\rightarrow$ Alignment of two genomes
  $\rightarrow$ Target is all observed sequences (*e.g.* NCBI BLAST non-redundant database)
- Heuristics (*e.g.* use only diagonals in dynamic programming)

BLAST (Basic Local Alignment Search Tool)

Less accurate than Smith-Waterman, **BUT** 50 times faster.

Idea: true matches are likely to have short stretches of identity (high score).

1. List of short words of fixed length that will match the query sequence (word length: 3 for protein; 11 for nucleic acids).
2. Scan database for these words. Extend matches in both directions in an attempt to find an alignment with a score exceeding S.

Segment pairs whose scores cannot be improved by extending or trimming are called high scoring pairs (HSPs).

Summary

- Dynamic programming (DP) saves time in sequence comparisons
- Some assumptions in DP, mention some
- In many applications, heuristics are needed to further speed up the comparison

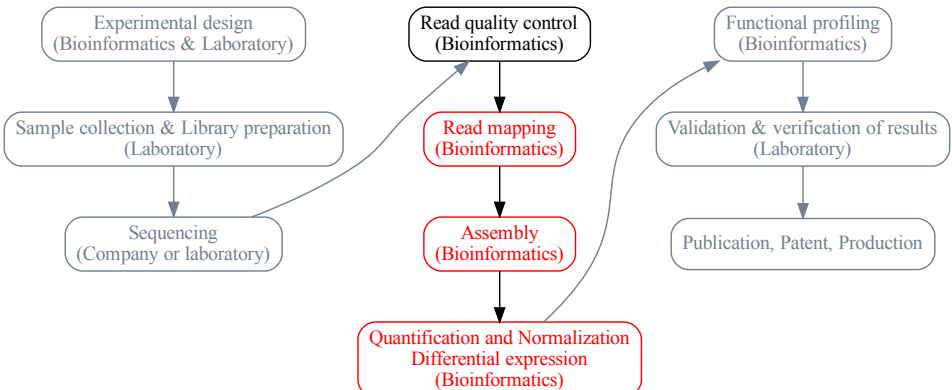
Exercise: Dynamical programming of pairwise alignment

Complete the dynamic programming matrix of a global alignment:

Align the sequences ACGTG and AACGGTG using a match score of 1, a mismatch of -4 and a gap cost of -10.

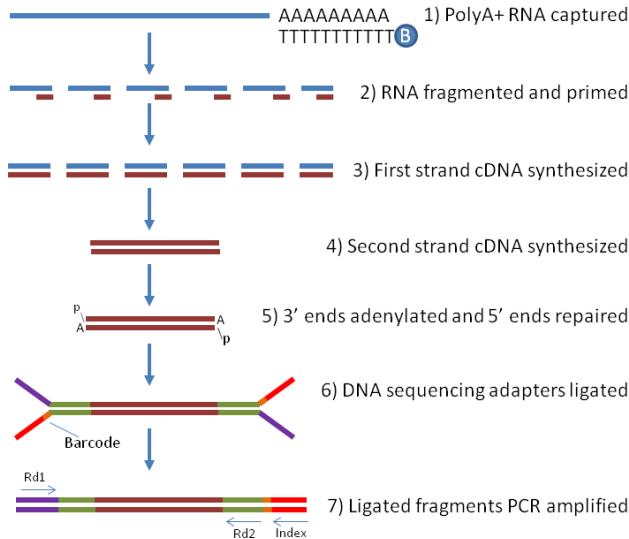
		A	C	G	T	G
A						
A						
C						
G						
G						
T						
G						

RNA–seq workflow



Read mapping

Full-length RNA library preparation



https://www.labome.com/method/RNA-seq.html

Raw data (Sequencing reads)

The FASTQ format:

```
@ERR459145.1 DHKW5DQ1:219:D0PT7ACXX:2:1101:1590:2149/1
GATCGGAAGAGCGGTTCAGCAGGAATGCCGAGATCGGAAGAGCGGTTCAGC
+
@7<DBADDDBH?DHHI@DH>HHHEGHIIIGGIFFGIBFAAGAFHA'5?B@D
```

- @: begin header
- 2:1101 flowcell lane 2, tile 1101
- x and y coordinates: 1590:2149
- /1 single-end reads;
  /1 and /2 paired-end (mate-paired) reads
- read sequence
- quality encoded ASCII characters

What is read mapping?

Determine position of a short read on the reference genome or transcriptome.

```
Reference:  ...AA-CGCCTT...       | = match
|:-:|||||          : = mismatch
Read:           AGGGGCCTT         - = gap
```

Naive mapping

Search for query at each position in reference genome

```
ACGTTACCGAATCGATCAAAGTCGA
GTTA
```

m = query length, n = genome length

Naive mapping

Search for query at each position in reference genome

```
ACGTTACCGAATCGATCAAAGTCGA
 GTTA
```

m = query length, n = genome length

Naive mapping

Search for query at each position in reference genome

```
ACGTTACCGAATCGATCAAAGTCGA
  GTTA                    :)
```

m = query length, n = genome length $\rightarrow$ Time: $O(mn)$

Naive mapping

- Human Genome (queries) would take far too long:
  - Illumina/Solexa sequencing technology produces **50 – 200 million, 32 – 100 bp short reads**
  - Mapping these reads to a **3.2 billion bp** human genome is a challenge
- Far worse when we allow for Indels and mismatches.

$\rightarrow$ Are optimal alignments (based on quality scores) still feasible?

Principles of mapping reads

Many sequenced reads are redundant!!!
We do not need to search the entire genome each time again.

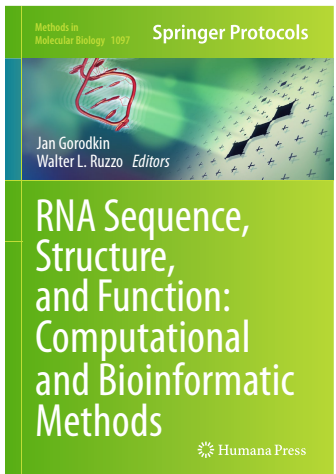
Book analog

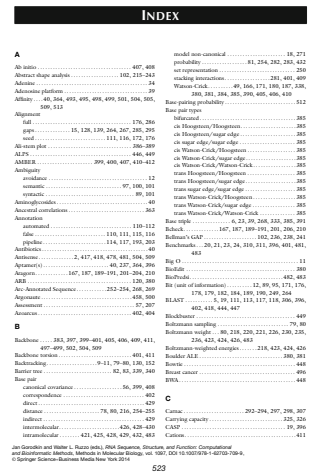
Do not search the entire book, instead search the book **index**.

Gorodkin et al. Methods Mol Biol. 2014

Book analog

Do not search the entire book, instead search the book **index**.





INDEX

A
Ab initio 407, 408
Abstract shape analysis 102, 215–243
Adenine 34
Advanota platform 29
Affinity 40, 364, 493, 495, 498, 499, 501, 504, 505, 509, 512
Alignment
  full 176, 286
  gaps 15, 128, 139, 264, 267, 285, 295
  seed 111, 116, 172, 176
All-train plot 386–389
ALPS 446, 449
AMBER 399, 400, 407, 410–412
Ambiguity
  avoidance 12
  semantic 97, 100, 101
  syntactic 89, 101
Aminoglycosides 40
Ancestral correlations 363
Annotation
  automated 110–112
  false 110, 111, 115, 116
  pipeline 114, 117, 193, 203
Antibiotics 40
Antisense 2, 417, 418, 478, 481, 504, 509
Aptamer(s) 40, 237, 264, 396
Aragorn 167, 187, 189–191, 201–204, 210
ARB 120, 380
Arc-Annotated Sequence 252–254, 268, 269
Argonaute 458, 500
Assessment 87, 207
Asteracea 402, 404

B
Backbone 383, 397, 399–401, 405, 406, 409, 411, 497–499, 502, 504, 509
Backbone torsion 401, 411
Backtracking 9–11, 79–80, 130, 152
Barrier tree 82, 83, 239, 340
Base pair
  canonical covariance 56, 399, 408
  correspondence 402
  direct 429
  distance 78, 80, 216, 254–255
  indirect 429
  intermolecular 420, 428–430
  intramolecular 421, 425, 428, 429, 432, 483
  model non-canonical 18, 271
  probability 81, 254, 282, 283, 432
  set representation 250
  stacking interactions 281, 401, 409
  Watson-Crick 49, 166, 171, 180, 187, 238, 280, 281, 284, 385, 390, 405, 406, 410
Base-pairing probability 512
Basic pair types
  bifurcated 285
  cis Hoogsteen/Hoogsteen 285
  cis Hoogsteen/sugar edge 285
  cis sugar edge/sugar edge 285
  cis Watson-Crick/Hoogsteen 285
  cis Watson-Crick/sugar edge 285
  cis Watson-Crick/Watson-Crick 285
  trans Hoogsteen/Hoogsteen 285
  trans Hoogsteen/sugar edge 285
  trans sugar edge/sugar edge 285
  trans Watson-Crick/Hoogsteen 285
  trans Watson-Crick/sugar edge 285
  trans Watson-Crick/Watson-Crick 285
Base triple 6, 23, 39, 268, 233, 285, 291
Bcheck 167, 187, 189–191, 201, 206, 210
Bellman's GAP 102, 236, 238, 241
Benchmarks 20, 21, 23, 24, 310, 311, 396, 401, 481, 483
Big O 11
Biofilm 380
BioProdict 482, 483
Bit (unit of information) 12, 89, 95, 171, 176, 178, 179, 182, 184, 189, 190, 249, 264
BLAST 5, 59, 111, 113, 117, 118, 206, 396, 402, 418, 444, 447
Blockbuster 449
Boltzmann sampling 79, 80
Boltzmann weight 80, 218, 220, 221, 226, 230, 235, 236, 423, 424, 426, 483
Boltzmann-weighted energies 218, 423, 424, 426
Boulder ALE 380, 381
Boxtic 448
Breast cancer 496
BWA 448

C
Camac 292–294, 297, 298, 307
Carrying capacity 325, 326
CASP 59, 396
Cations 411

Jan Gorodkin and Walter L. Ruzzo (eds.), *RNA Sequence, Structure, and Function: Computational and Bioinformatic Methods*, Methods in Molecular Biology, vol. 1097, DOI 10.1007/978-1-62703-709-9,

Gorodkin et al. Methods Mol Biol. 2014

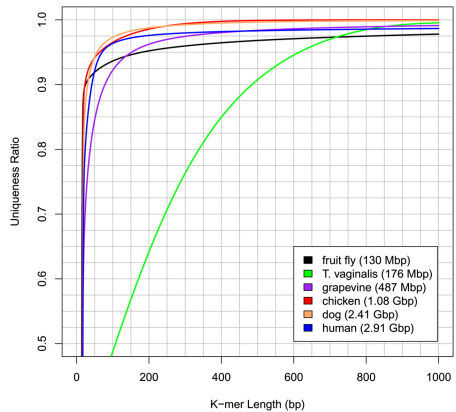
k-mer

"k-mer" is a substring of length k

For example sequence **GGCGATTCATCG**:

4-mer GGCG, GCGA CGAT

3-mer GGC, GCG, CGA, GAT



Schatz et al. Genome Res 2010

Read mapping through indexing

Most computational time is spent on alignment.

Solution

An index is a data structure that improves the speed of data retrieval operations at the cost of additional storage space to maintain the index data structure.

Pros

Quick search for matches in an entire genome.

Cons

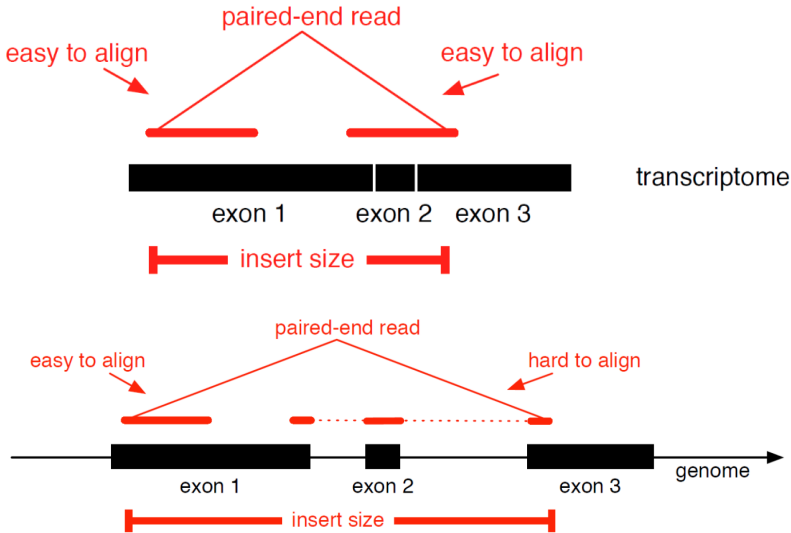
Index structure of the entire genome takes a lot of memory.

Indexing problems

In principle read mapping is to map an exact piece of sequence to the genome.

- Why an exact mapping might not always be what we want.
- Which concerns might you have when mapping genomic sequence?
- Which concerns might you have when mapping transcribed sequence?

Transcriptome versus genome mapping



Indexing problems

Flexibility and constraints:

- *Errors versus natural variation:*
  trade-off in error threshold.
- *Computational efficiency (time / memory):*
  allowed mismatches / unique mappings.
- Balance between speed, memory and reported mappings.

$\rightarrow$ Indexing is often used for seed matching.

$\Rightarrow$ Indexing method choice is crucial!

Commonly used indexing methods

- **Hash-based** (BLAST, Salmon, Kallisto)
- **Suffix arrays** (Salmon, STAR)

  A sorted table of all suffixes (substrings) of a given string
- **Burrows-Wheeler Transform** (BWA, SOAP2, Bowtie2, Hisat)

  A compressed form of suffix arrays

"Seed-and-extend" approach

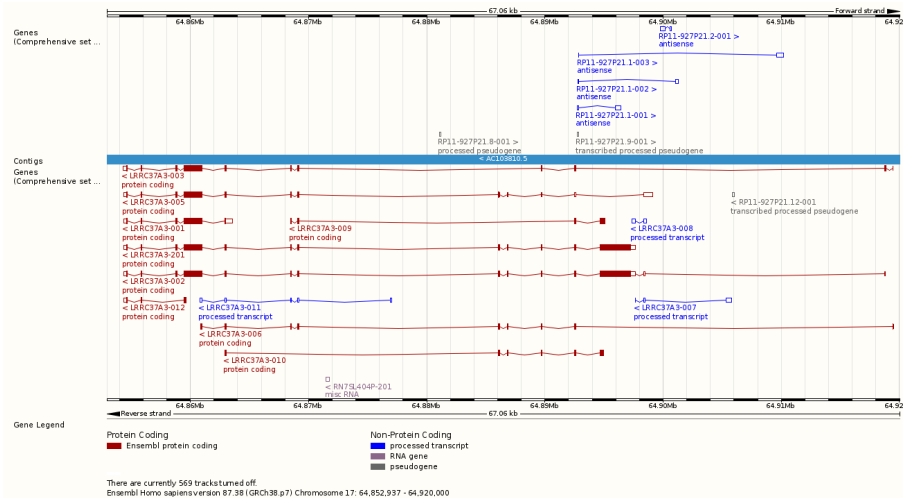
1. find the best possible match of a seed in an index made up from the reference genome
2. every matched seed is extended on both sides by optimal local alignment

Common software

STAR, HISAT2, BLAT

What are the characteristics of eukaryotic transcripts?

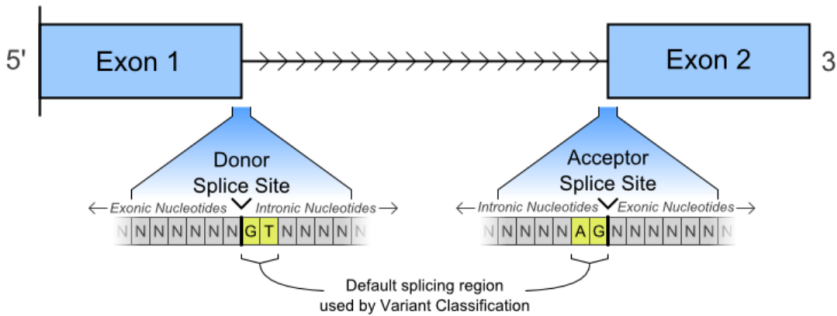
Hurdles for a transcriptome analysis?



- **Splicing (Introns, Exons)**
- Poly-A
- Cap-complex
- ... many more

Splice-aware genome mapping

There are many mappers, but they must be able to detect splice junctions to be used in transcriptome assembly and quantification.

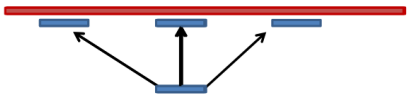


Common splice-aware alignment software

STAR, HISAT2, BLAT, TopHat (based on Bowtie2), Segemehl

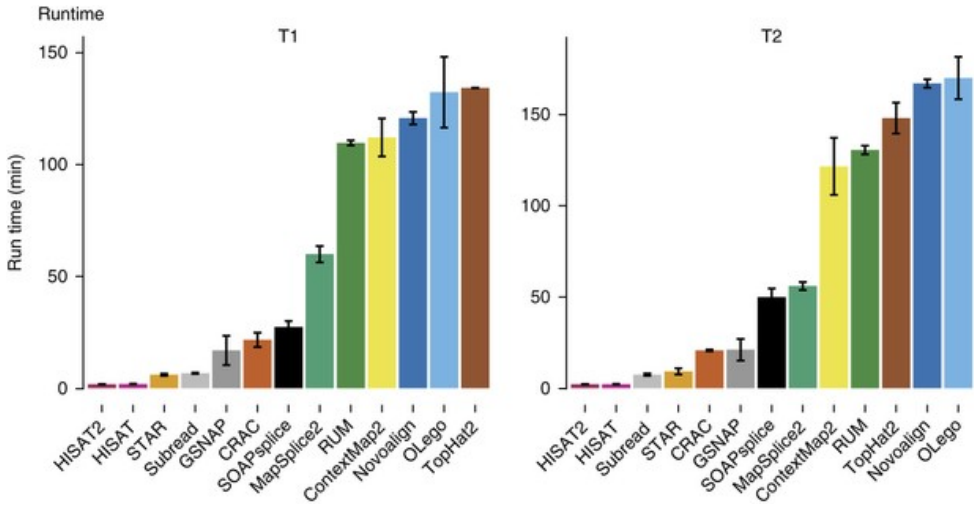
Multimappers

- Reads that align to multiple locations



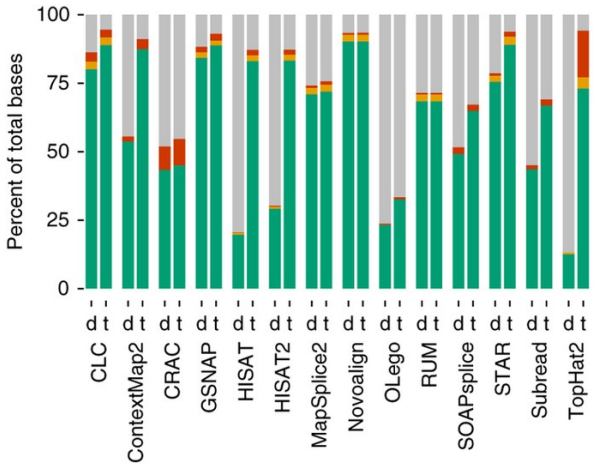
- How to handle multi-mapped reads? Depends on tool:
  - Map to best region (but what is "best"? And what about ties?)
  - Map to all regions
  - Map to one region randomly
  - Discard read
- How do we determine best region (primary alignment)?
  - Assign alignment score to every mapping

Aligner's speed



Baruzzo *et al.* "Simulation-based comprehensive benchmarking of RNA-seq aligners." Nature Methods 2017

Aligner's accuracy



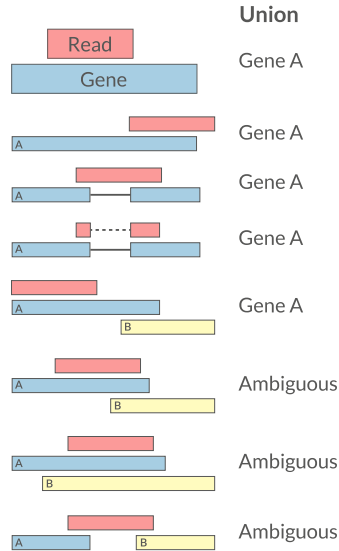
Increasing Accuracy

- Novel variants / RNA editing
- Allele-specific expression
- Genome annotation
- Gene and transcript discovery
- Differential expression

Baruzzo *et al.* "Simulation-based comprehensive benchmarking of RNA-seq aligners." Nature Methods 2017

Gene-level/Exon-level quantification

- Count reads mapping to a genomic feature
- Read counts = gene expression
- What about transcripts?



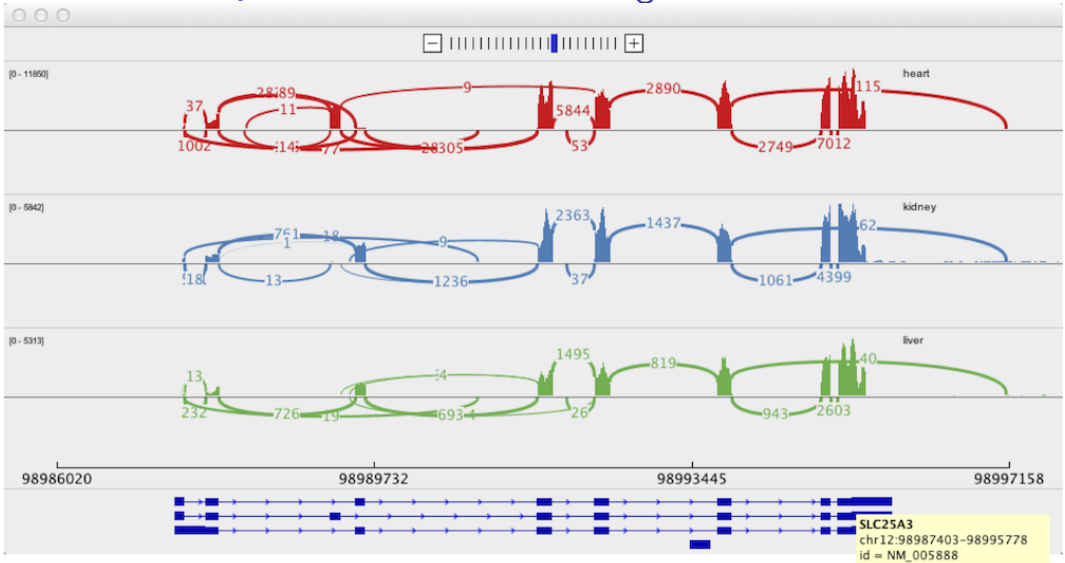
Common read counting software

featureCounts, HTSeq

Transcriptome assembly

Why assembling?

Quantification of different gene isoforms



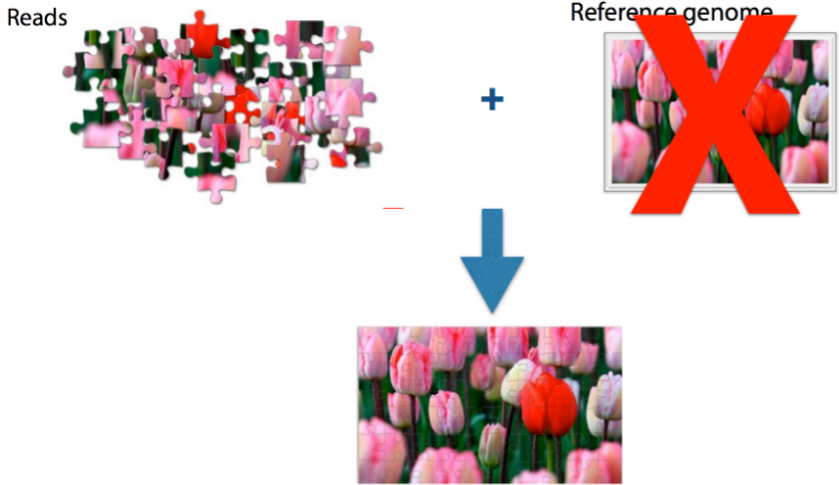
Uncertainties in transcript-level quantification

- we do not expect unique regions but sets of transcripts
- a read can match several transcripts (short reads)
- read abundance from different genes and transcripts spans several orders of magnitude
- experiments can be without information about strand
- reads originate from mature RNA (mRNA and ncRNA) and from incompletely spliced precursor RNA

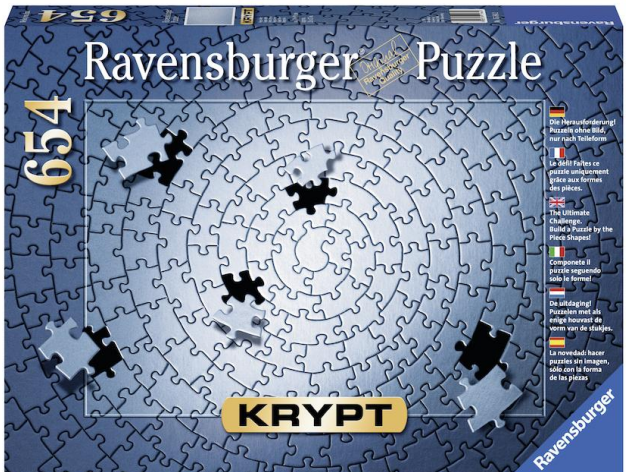
$\Rightarrow$ Assembly

The assembly process

Reference based (comparative) *OR* De novo



Messy biology



- Missing pieces
- Sequencing errors
- Experimental biases
- . . .

Brainstorming: Transcript assembly

Given reads:

1. AAAACCCC
2. CCCCGG
3. GGUUUUUUUU
4. UUCUUUAAAA

Brainstorming: Transcript assembly

Given reads:

1. AAAACCCC
2. CCCCGG
3. GGUUUUUUUU
4. UUCUUUAAAA

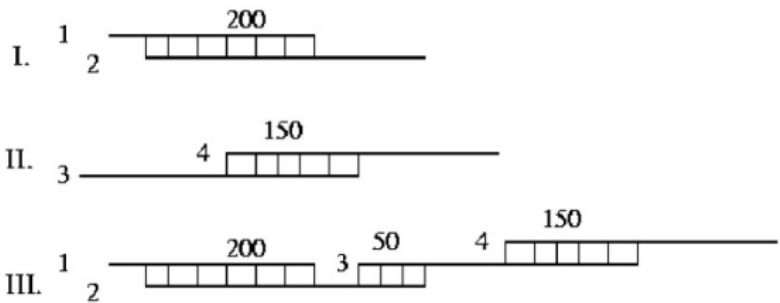
```
AAAACCCC   GGUUUUUUUU
    CCCCGGGG       UUCUUUUAAAA
```

Or with error tolerance:

```
AAAACCCC   GGUUUUUUUUU
    CCCCGGGG   UUCUUUUAAAA
```

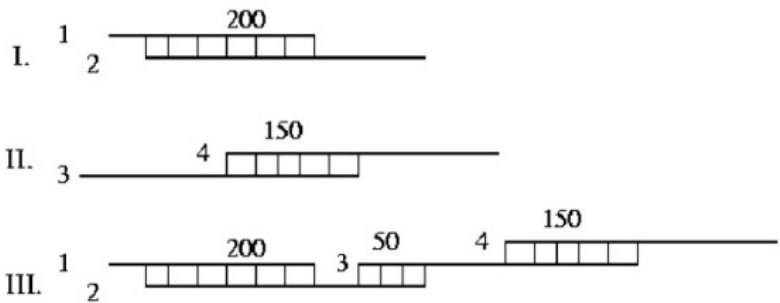
Greedy algorithm

We pick two strings s_i and s_j with largest overlap from the set of all reads (breaking ties arbitrarily) and replace them with their merge. Stop when there is only one string left.



Greedy algorithm

We pick two strings s_i and s_j with largest overlap from the set of all reads (breaking ties arbitrarily) and replace them with their merge. Stop when there is only one string left.



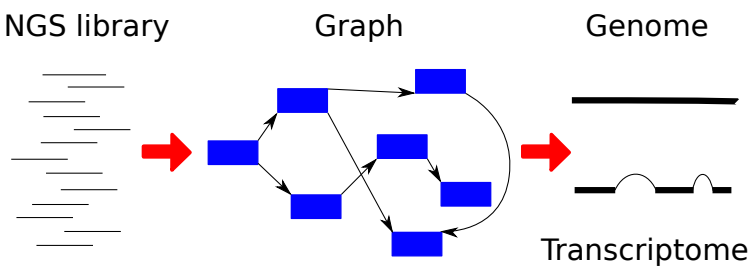
Terrible idea for transcriptome assembly! Why?

Why to use graphs for assembly?

Say a sequencer produces **d** reads of length **n** in one sequencing run from a genome of length **m**:

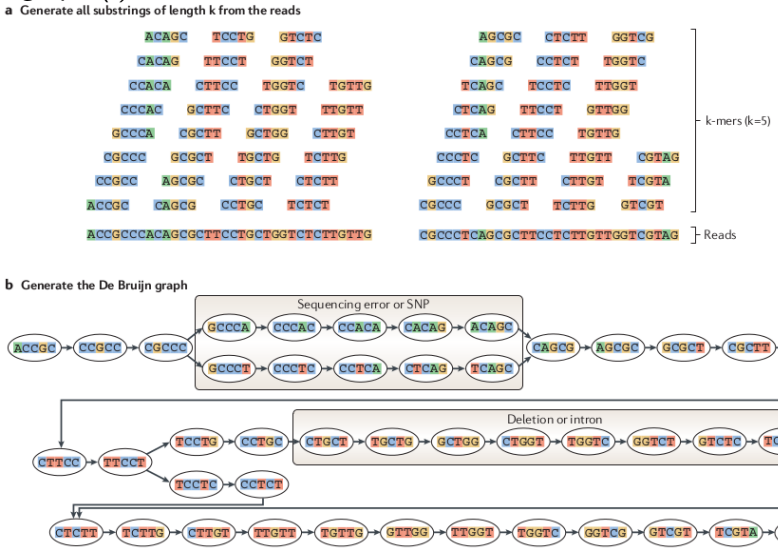
- d 6×10^9 reads
- n 100 nt
- m 3×10^9 nt $\sim$ human

Task: Glue overlapping reads together to recover biology.
$\rightarrow$ Combinatorical problem best solved by graph theory!

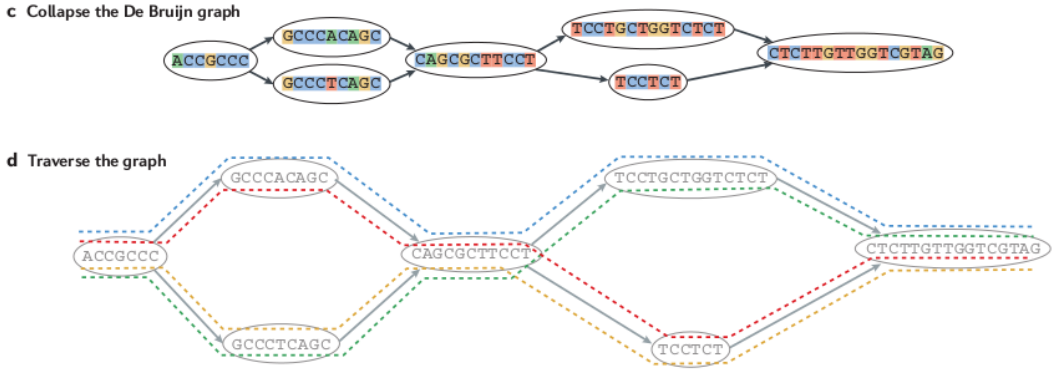


De novo assembly

De Bruijn graph (I):



De Bruijn graph (II):



De novo based methods

Applications

Organisms with poor reference genomes or no reference genome; independent of correct splice sites or intron length

Pros

Identification of novel transcripts; no reference needed

Cons

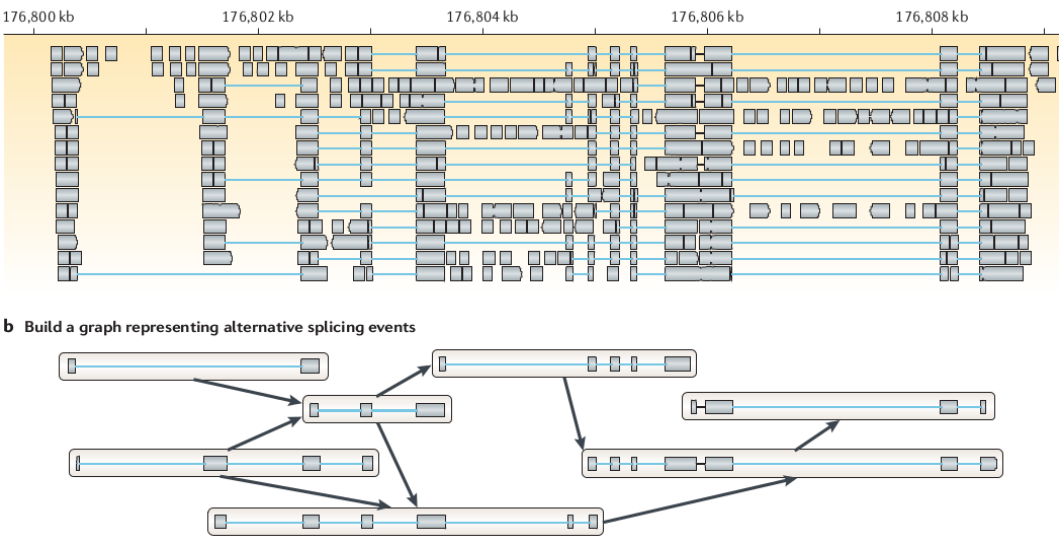
Computationally intensive; requires high read depth; sensitive to sequencing errors, chimeric molecules and contamination; Sensitive to transcripts from highly similar genes (*e.g.* paralogues)

Common software

Trinity

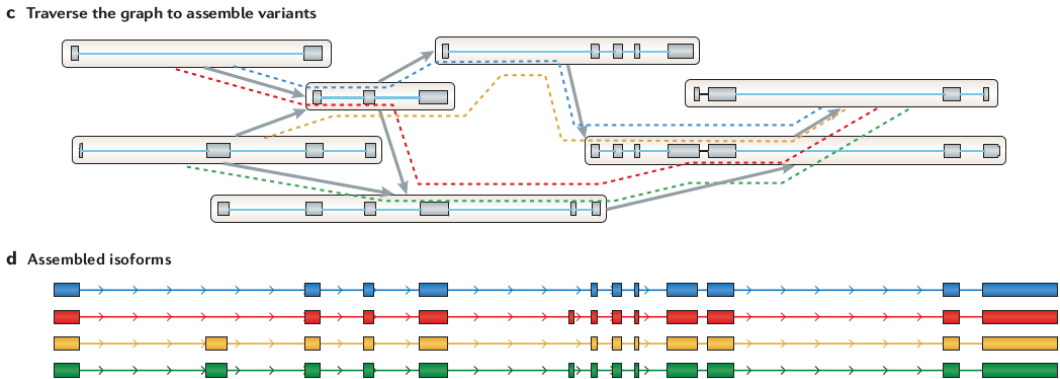
Reference based assembly

Overlap graph (I):



Reference based assembly

Overlap graph (II):



Reference based assembly

Applications

Organisms with good reference genomes, except perhaps polyploid organisms

Pros

Low computational resources (can be run in parallel, not much memory); Contamination not a major issue; Very sensitive even with low abundant transcripts (less coverage needed)

Cons

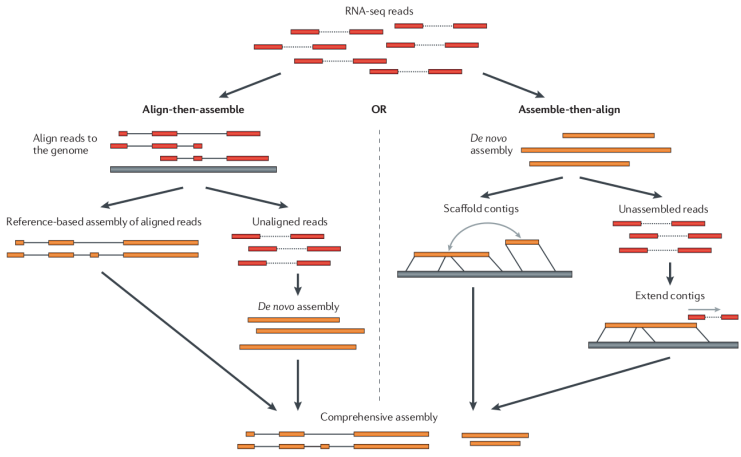
Reference dependant (quality of reference assembly); known splice site dependant; long introns may be predicted; mismatch between reference used and our sample (e.g. cancer)

Common software

Cufflinks, StringTie

Combined methods

Combining both *de novo* (Assemble) and *reference based* (Align) strategies is sometimes the best method:



Martin & Wang "Next-generation transcriptome assembly" Nat Rev Genet 2011

Assembly QC

Evaluating assembly quality is as important as the assembly itself!

- Assembly quality depends on
  1. Coverage: low coverage is mathematically hopeless
  2. Repeat composition: high repeat content is challenging
  3. Read length: longer reads help resolve repeats
  4. Sequencing errors: reduced coverage and more false positives

Alignment-free methods ("pseudoaligners")

1. Focus: Only annotated transcripts (not entire genome!)
2. Pseudoalign: K-mer composition of reads/transcripts
3. Abundance *estimation*
4. GC-content, transcript position correction included (*e.g.* 3' end degragation)

Pros

dramatic increase in speed; improvements in accuracy for gene-level quantification

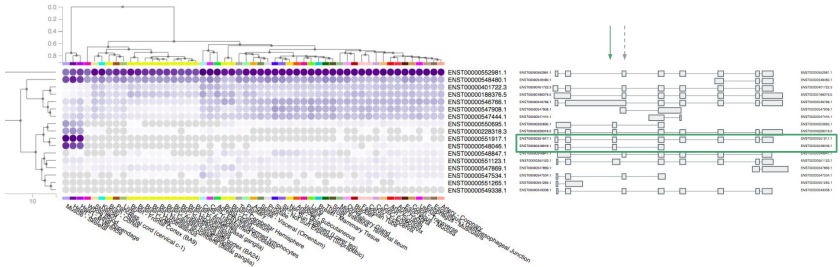
Cons

absolute reliance on a precise and comprehensive transcript annotation; no information on where each read is mapping

Common software

Salmon, Kallisto, Sailfish

Transcript-level quantification



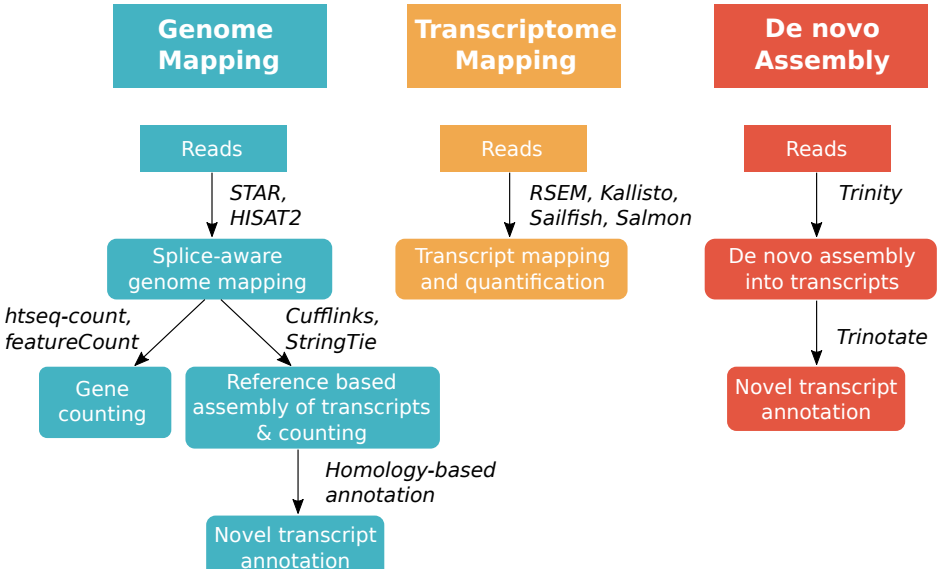
- Quantify per transcript (not just exon or gene)
- ! Beware of annotation quality
- Suggested reading: CSAMA tutorial, Zhang *et al.* "Evaluation and comparison of computational tools for RNA-seq isoform quantification." BMC Genomics 2017

Common software

RSEM, Cufflinks, Kallisto, Salmon, Sailfish

Fig: https://www.gtexportal.org/home/gene/SLC25A3

Method overview for mapping and assembly and quantification



Summary

- Tradeoff between accuracy and speed of read alignment algorithm
- Be aware how your choice of mapping tools effects your results
  - How are mismatches handled?
  - How are multi-mapped reads handled?
- Gene-level abundance estimates and statistical inference offer advantages over transcript-level analyses in terms of performance and interpretability
- If isoform detection is the research goal then consider long-read sequencing technologies (*e.g.* PacBio, Oxford Nanopore)

Expression analysis

Count matrix

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516	SRR1039517	SRR1039520	SRR1039521
ENSG00000000003	679	448	873	408	1138	1047	770	572
ENSG00000000005	0	0	0	0	0	0	0	0
ENSG00000000419	467	515	621	365	587	799	417	508
ENSG00000000457	260	211	263	164	245	331	233	229
ENSG00000000460	60	55	40	35	78	63	76	60
ENSG00000000938	0	0	2	0	1	0	0	0
ENSG00000000971	3251	3679	6177	4252	6721	11027	5176	7995
ENSG00000001036	1433	1062	1733	881	1424	1439	1359	1109
ENSG00000001084	519	380	595	493	820	714	696	704
ENSG00000001167	394	236	464	175	658	584	360	269
ENSG00000001460	172	168	264	118	241	210	155	177
ENSG00000001461	2112	1867	5137	2657	2735	2751	2467	2905
ENSG00000001497	524	488	638	357	676	806	493	475
ENSG00000001561	71	51	211	156	23	38	134	172
ENSG00000001617	555	394	905	415	727	697	618	599
ENSG00000001626	10	2	9	2	10	6	5	5
ENSG00000001629	1660	1251	2259	1079	2462	2514	1888	1660
ENSG00000001630	59	54	66	23	84	87	31	59
ENSG00000001631	729	692	943	475	1034	1163	731	744
ENSG00000002016	201	161	256	99	268	257	160	137
ENSG00000002079	3	0	3	1	4	0	0	1

Estimation of gene expression

You are analyzing 2 genes (gene A and B) in two conditions (condition 1 and 2) on the bases of an RNA-seq experiment that resulted in the following number of reads:

	Condition 1	Condition 2
Gene A	1000	3000
Gene B	2000	4000

Are the following statements correct:

a. Both genes A and B are more expressed in condition 2.

b. Gene B is more expressed than gene A.

Estimation of gene expression

We cannot state any such thing since we do not know

a. **sequencing depth (library size)**,
   expression of all other genes within the sample
b. **gene length**,
   the longer the gene, more reads will be mapped
c. GC content

Estimation of gene expression

We cannot state any such thing since we do not know

a. **sequencing depth (library size)**,
   expression of all other genes within the sample
b. **gene length**,
   the longer the gene, more reads will be mapped
c. GC content

Solution: Control (Normalize) for

1. sequencing depth
2. compositional bias

Normalization

R = Reads count for the gene
G = Gene length in kilobases
T = Total number of mapped reads in a sample

Reads Per Kilobase transcript per Million mapped reads (RPKM):

$$RPKM = \frac{\left(\frac{R}{T/10^6}\right)}{G}$$

$RPKM$ is used for single end reads

FPKM is similar measure used for paired end reads.

Tags Per Million (TPM):

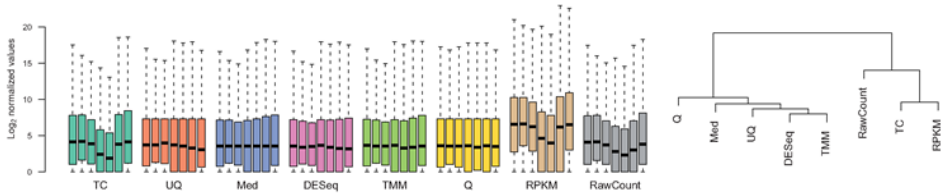
$$TPM = \frac{\frac{R}{G}}{\left(\frac{\sum_{all\ \ genes} \frac{R}{G}}{10^6}\right)} = \frac{RPKM}{\sum RPKM}$$

Counts Per Million (CPM):

$$CPM = \frac{R*10^6}{T}$$

- **RPKM/FPKM:** First metric used in the old times (e.g. cufflinks).
- **TPM:** Alternative to RPKM/FPKM. Total TPM counts are equal to 1 M.
- **CPM:** Similar to TPM as it adds up to 1 M, but does not consider gene lengths. The most applied metric in the field.

Normalization



- Use **raw counts** for DE analysis using DGE packages (*e.g.* DESeq2, edgeR) as normalization is done internally
- RPKM/FPKM/TPM/CPM are not recommended for DE analysis.
- TPM/CPM are great for visual exploration of data (heatmaps, abundance comparisons, PCA/MDS plots).
  - Sum of all TPMs/CPMs is constant (1 million)
- Median of Ratios (DESeq2) and TMM (edgeR) methods perform the best for DGE
- other solutions: spike-ins/house-keeping genes

Dillies *et al.* "A comprehensive evaluation of normalization methods for Illumina high-throughput RNAseq data analysis." Brief Bioinform. 2013

Normalization for DE

Assumption: most genes are not differential expressed

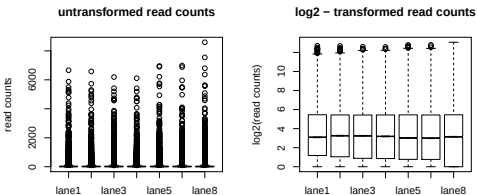
DESeq2's sample-wise sizefactor:
`DESeq2` scaling factor for a given sample is computed as the median of the ratio, for each gene, of its read count over its geometric mean across all libraries.

DESeq Bioconductor package > estimateSizeFactors > sizeFactors

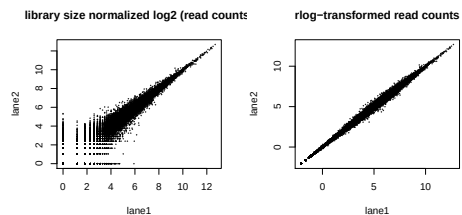
Trimmed Mean of M-values (TMM):
edgeR Bioconductor > calcNormFactors > estimateCommonDisp > estimateTagwiseDisp

Transformation of sequencing-depth-normalized read counts

Log_2 transformation:



Transformation incl variance shrinkage:



For clustering, heatmaps etc use VST (DESeq2), VOOM (limma) or RLOG (DESeq2)

Challenge your data by different normalization methods → robustness of DGE analysis

---

Eder *et al.* "NVT: a fast and simple tool for the assessment of RNA-Seq normalization strategies" Bioinformatics 2016

Explore global and local read count patterns (feature selection)

1. unsupervised

   no a priori information is needed
   $\rightarrow$ to detect technical noise and batch effects
   - Dimensional reduction $\rightarrow$ Principle Component Analysis (PCA)
   - Clustering $\rightarrow$ hierarchical, k-means

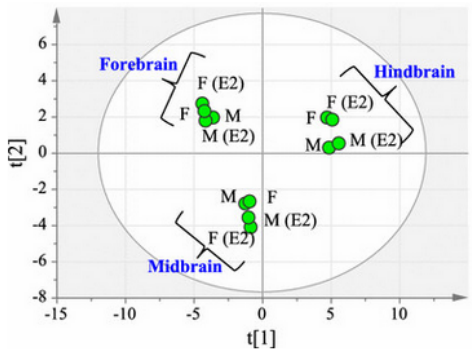
   PCA and clustering should be done on normalized and transformed read counts so that high variability of low read counts does not occlude potentially informative data trends.

2. supervised

   usage of known biological labels
   $\rightarrow$ differential expression

Principle component analysis – PCA

- transforms measurements into new variables that are truly independent
- new variables of most variance are the principal components
- dimensionality reduction

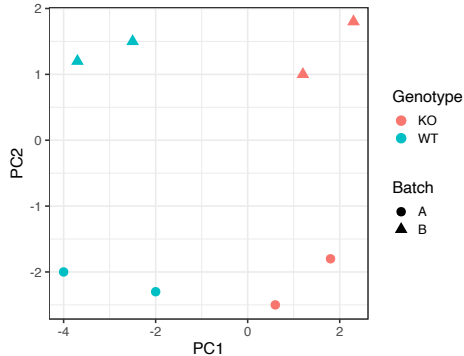


Applications:

- visualization of your data in lower dimensions (2D, 3D)
- find patterns in numeric data
- identify batch effects or other possible covariates (*e.g.* male and female) by labeling them

Pradhan *et al.* Behav Brain Funct 2015

PCA to detect batch effects



- PC1 separates the *genotype* (group of interest); however
- PC2 separates the *batch* effect (or other covariates)
  - *e.g.* experiment date, sex, experimenter, different RNA isolation kit (you name it!)
- When batch effect is observed in PCA plots, add it as covariate to your GLM
  - $\sim$ Batch + Genotype

Clustering

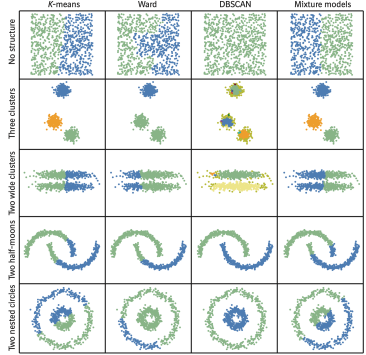
There are several clustering algorithms and more are being developed. Why?

Ronan *et al.* "Avoiding common pitfalls when clustering biological data" Science Signalling 2016

Clustering

There are several clustering algorithms and more are being developed. Why?

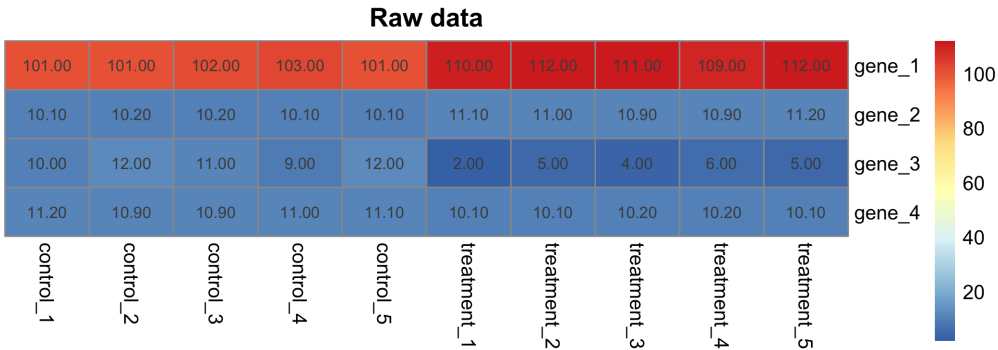
- No clustering algorithm is perfect
- Remember! Always "see" your data and judge if the clusters make sense
- The performance of the clustering algorithm depends on the structure of your data



Ronan *et al.* "Avoiding common pitfalls when clustering biological data" Science Signalling 2016

Finding patterns in your data

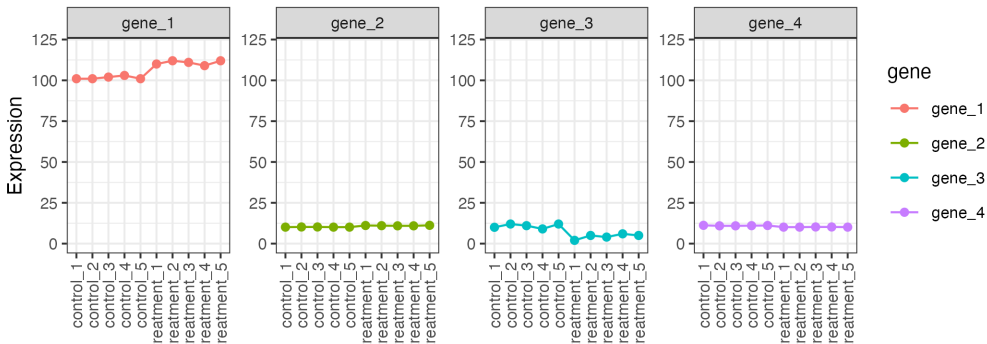
Let's assume that you got these 4 genes differentially expressed in your dataset



Can you identify a pattern?

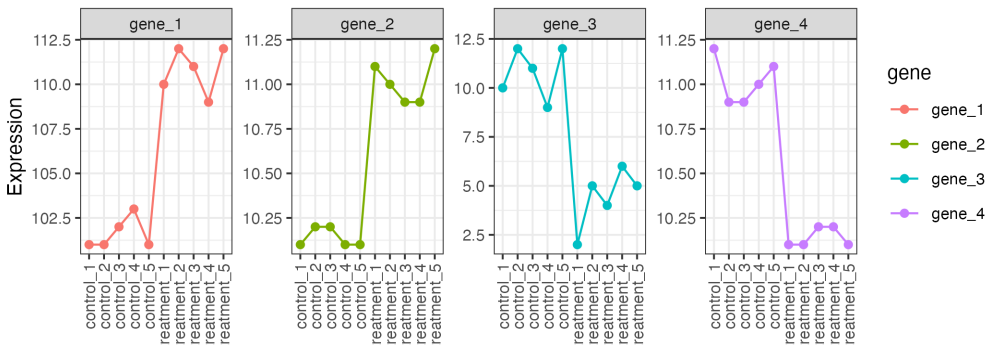
Finding patterns in your data

What about now?



Finding patterns in your data

Perhaps this gives you a better hint

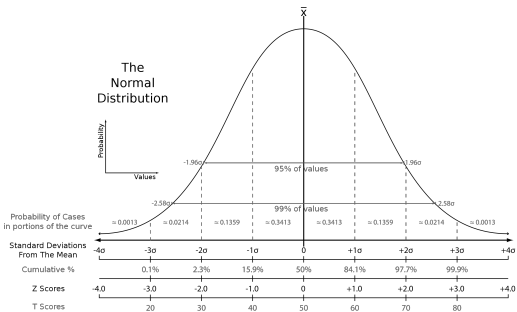


Z-score

$$z = \frac{x-\mu}{\sigma}$$

μ = mean

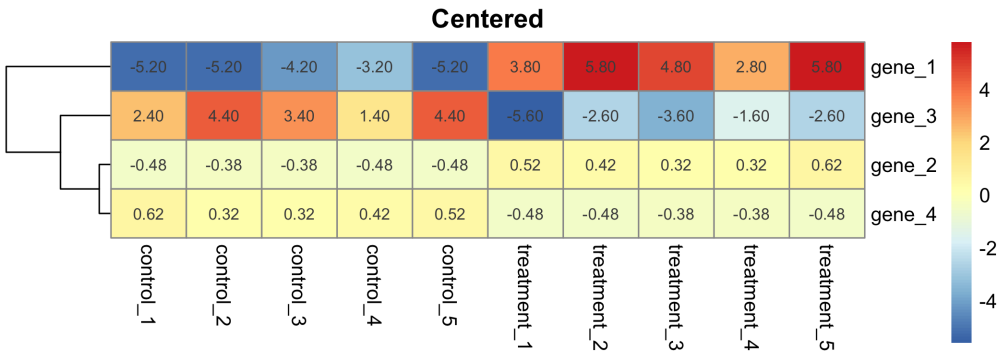
σ = standard deviation



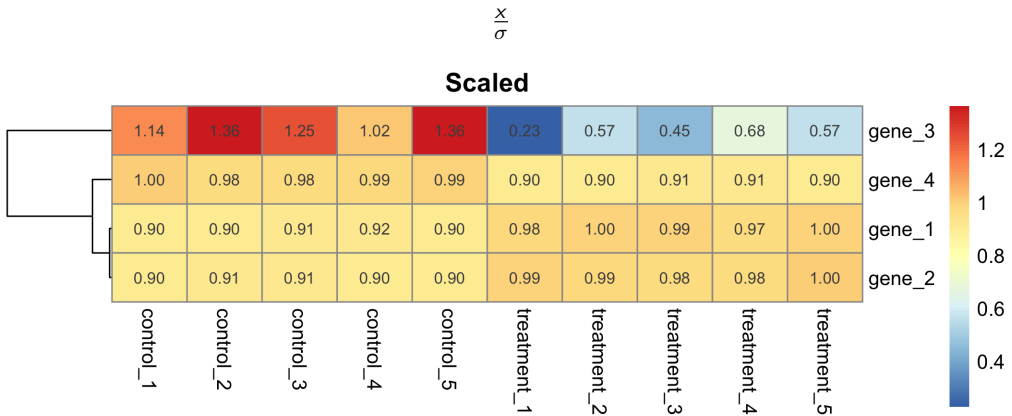
https://en.wikipedia.org/wiki/Standard_score

Centering data

$$x - \mu$$

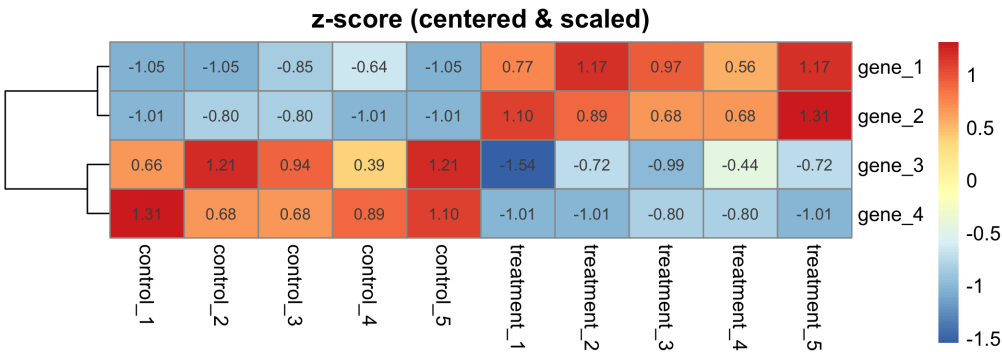


Scaling data



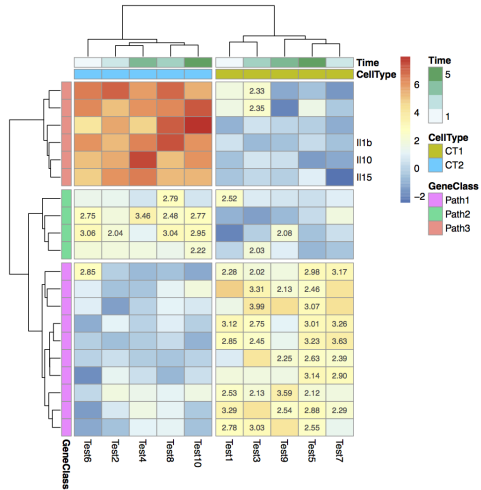
Z-score

$$z = \frac{x - \mu}{\sigma}$$



Hierarchical clustering

- No need for pre-defining the number of clusters
- Results dependent on
  1. *distance metric:* euclidean, manhattan, Pearson correlation etc.
  2. *clustering method:* Ward, complete, average etc.
- Used with heatmaps, thus allows better visual inspection of data



Differential expression (DE)

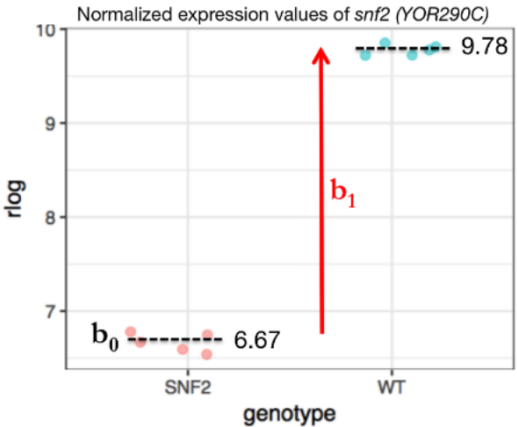
Common scientific question:
Quantification and statistical inference of systematic changes between conditions.

Principles are the same as for all other significance tests:

1. Use the replicates (samples of the same conditions) to estimate the within-condition variability.(variance) of the expression
2. Use the expression and variance to test whether the difference between conditions is random or not

The test's statistical power increases with more biological (and technical) replicates!

Differential expression (DE)



Linear regression model (LM) is evaluated for every gene: $Y = b_0 + b_1 * x + e$

Y . . . describes all read counts for a gene
b_0 . . . average of the baseline group
x . . . condition (for RNA-seq often 0 or 1)
e . . . error or uncertainty
b_1 . . . coefficent that captures the difference between different conditions

$\rightarrow$ the closeness of b_1 to zero will be evaluated during statistical testing steps

$\rightarrow$ DESeq2 and edgeR use a generalized linear model (GLM)

Parametric vs. non-parametric methods

It would be nice to not have to assume anything about the expression value distributions but only use rank-order statistics.

However, it is (typically) harder to show statistical significance with non-parametric methods with few replicates (less than 10?).

Issues with DE testing for RNA-seq

Couldn't we just use a Student's t-test for each gene?

1. Distribution is not **normal**. Which parametric distribution should I use?
2. **Variance** across groups may not be **homogeneous** e.g. unequal group size
3. The number of replicates is often too small to estimate the variance.
4. If we test each gene for DE, we have to account for **multiple testing**!

Which parametric distribution should I use?

Models for read counts originated from the idea that each read is sampled independently from a pool of reads and hence the number of reads for a given gene follows a ...

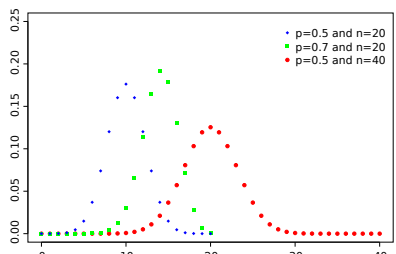
Binomial distribution

The binomial distribution works when we have a fixed number of trials n, each with a constant probability of success p.

The random variable X is the number of k successes:

$$p(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Event: An RNA-seq read "lands" in a given gene (success) or not (failure)



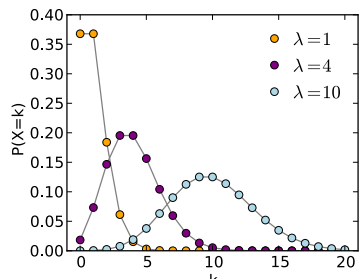
Poisson distribution

RNA-seq experiments produce large number of reads (n is large) and probabilities of success are small (p is small) which can be modelled by the poisson distribution which is an approximation of the binomial.

Instead we know the average number of successes per intervall:

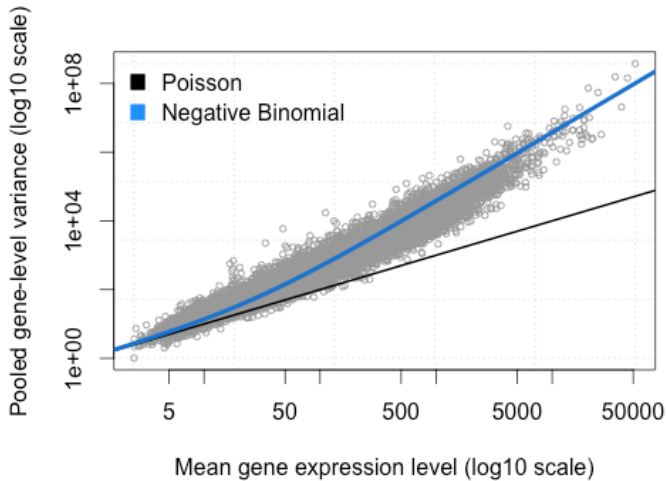
$$\lambda = np$$

For $X \sim \text{Poisson}(\lambda)$, both the mean and the variance are equal to λ.



Poisson versus negative binomial distribution

Many studies have shown that the variance grows faster than the mean in RNAseq data. This is known as **overdispersion**.

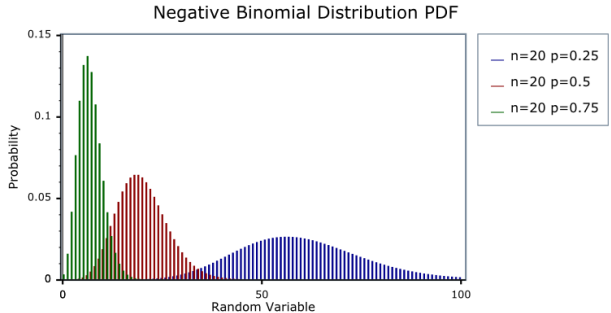


Negative binomial distribution

The negative binomial distribution works for discrete data over an unbounded positive range whose sample variance exceeds the sample mean.

The random variable X is the number of trials needed to make n successes (read counts) if the probability of a single success is p:

$$NB(X = x) = \binom{x-1}{n-1} p^n (1-p)^{x-n}$$



Design & contrast matrix

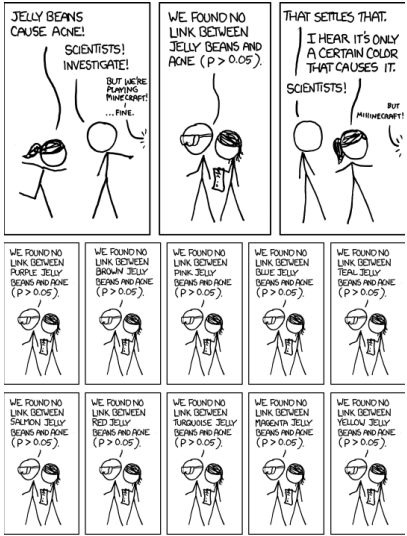
Design matrix (a.k.a. model matrix) has 2 main roles:

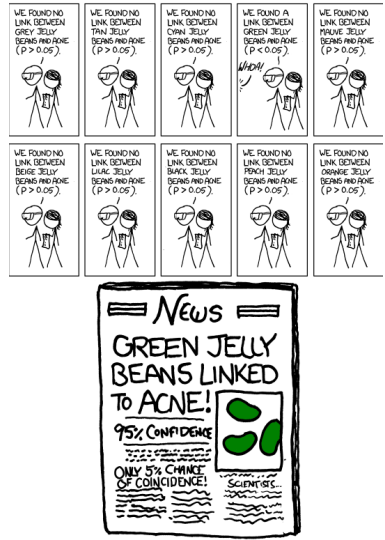
1. defines the form of the model, or structure of the relationship between genes and explanatory variables
2. is used to store values of the explanatory variable(s)

Contrast matrix is used for:

1. identifying the differences (contrast) between explanatory variables
   e.g. $group_1$ vs $group_2$

Multiple testing issue





https://xkcd.com/882/

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with $p < 0.05$ level assuming independence of tests

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with $p < 0.05$ level assuming independence of tests
- Probability of **observing** a type-I error (false positive) in a single test: $\alpha_{single} = 0.05$

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with p < 0.05 level assuming independence of tests
- Probability of **observing** a type-I error (false positive) in a single test: $\alpha_{single} = 0.05$
- Probability of **not observing** a type-I error (false positive) in a single test: $\beta_{single} = 1 - \alpha = 0.95$

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with p < 0.05 level assuming independence of tests
- Probability of **observing** a type-I error (false positive) in a single test: $\alpha_{single} = 0.05$
- Probability of **not observing** a type-I error (false positive) in a single test: $\beta_{single} = 1 - \alpha = 0.95$
- If you run a test with 20 (n) genes: $\beta_{multiple} = (1 - \alpha)^n = 0.95^{20} = 0.36$

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with p < 0.05 level assuming independence of tests
- Probability of **observing** a type-I error (false positive) in a single test: $\alpha_{single} = 0.05$
- Probability of **not observing** a type-I error (false positive) in a single test: $\beta_{single} = 1 - \alpha = 0.95$
- If you run a test with 20 (n) genes: $\beta_{multiple} = (1 - \alpha)^n = 0.95^{20} = 0.36$
- Likewise, type-I error for multiple comparisons become: $\alpha_{multiple} = 1 - (1 - \alpha)^n = 0.64$

Multiple testing issue

- Assume that you are comparing genes between condition A and B
- You would expect 1 in 20 (5/100) genes to be significant with $p < 0.05$ level assuming independence of tests
- Probability of **observing** a type-I error (false positive) in a single test: $\alpha_{single} = 0.05$
- Probability of **not observing** a type-I error (false positive) in a single test: $\beta_{single} = 1 - \alpha = 0.95$
- If you run a test with 20 (n) genes: $\beta_{multiple} = (1 - \alpha)^n = 0.95^{20} = 0.36$
- Likewise, type-I error for multiple comparisons become: $\alpha_{multiple} = 1 - (1 - \alpha)^n = 0.64$

- If the number of tests (m) increases, the type-I error rate $\alpha_{multiple}$ will reach to 1
- This inflation of α has to be handled by multiple testing correction for p-values
- Most applied method for omics studies is Benjamini-Hochberg method (a.k.a. False Discovery Rate, FDR)

Implementation of DE testing for RNA-seq

- *Seq. depth normalization:* `DESeq2` uses sample-wise size factor, `edgeR` and `Limma-Voom` use TMM
- *Assumed distribution:* `edgeR` and `DESeq` model the count data using a negative binomial distribution and use their own modified statistical tests based on that. `Limma-Voom` uses log-normal distribution and t-test.
- *Dispersion estimate:* `edgeR`, `DESeq2`, `Limma-Voom` (in slightly different ways) "borrow" information across genes to get a better variance estimate.
- *Multiple testing issue:* All current packages report false discovery rate FDR (most often Benjamini-Hochberg corrected p values).

DESeq2

0. Find confounding effects by PCA, sva package or RUVSeq package
1. Model known batches or other possible covariates in the LM/GLM model $>$ design formula, *e.g.* $\sim$sex+condition
2. Estimate size factors $>$ estimateSizeFactors()
3. Estimate gene-wise dispersion $>$ estimateDispersions()
4. Fit curve to gene-wise dispersion estimates
5. Shrink gene-wise dispersion estimates
6. GLM fit for each gene
7. Wald test $>$ nbinomWaldTest()
8. Results log2 fold changes (LFC) and adjusted p-values for each gene $>$ results()
9. Summary of DE genes, *e.g.* LFC > 0 (up) and adj. pvalue $< 0.1 >$ summary()

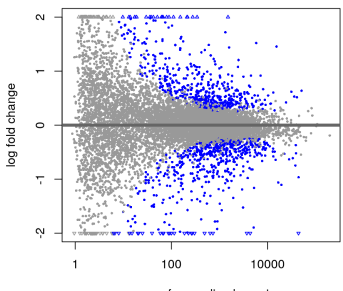
https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#theory

http://bioconductor.org/packages/release/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html

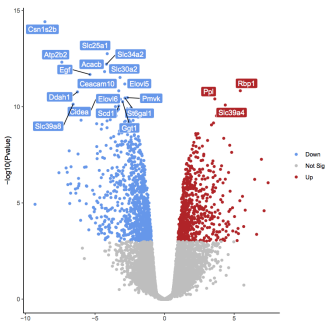
Visualization of DE analysis

MA-plot

> plotMA()

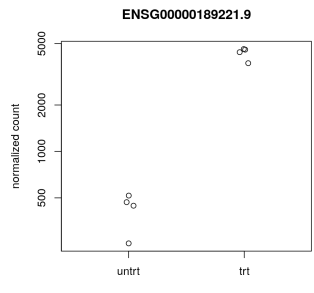


Volcano plot



Normalised counts

> plotCounts()



Summary

- Always challenge your data, think of plausible technical explanation first

"I'm a scientist and I know what constitutes proof. But the reason I call myself by my childhood name is to remind myself that a scientist must also be absolutely like a child. If [they] see a thing, [they] must say that [they] see it, whether it was what [they] thought [they] were going to see or not. See first, think later, then test. But always see first. Otherwise you will only see what you were expecting. Most scientists forget that. "

– adapted from The Ultimate Hitchhiker's Guide to the Galaxy by Douglas Adams

PhD course "Bioinformatics analysis of gene expression data (BAGED)"

1. Do you have your own bulk or single-cell RNA-seq data of a vertebrate?
2. Do you want to learn how to analyze your data yourself?
3. What? 2 to 3 weeks of lectures, tutorials and most importantly student projects
4. Hardware and Software? UCloud, (Galaxy), R, Cytoscape
5. When? BAGED-bulk January 2024; BAGED-single March 2024
6. Sign up early due to limited number of seats (Oct-Nov)