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

10 April 2024

Why sequencing?

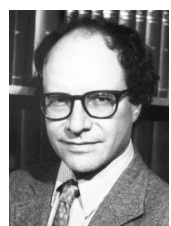
- Assemble the genome and transcriptome of a species
- Find genomic variation in a population
- Find genomic and transcriptomic associations with diseases and phenotypes
- Find organisms in environmental sample $\rightarrow$ metagenomics and -transcriptomics
- Identify potential drug targets $\rightarrow$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

$\Rightarrow$ One of the biggest hammers in the tool box right now – always ask could this experiment be done using sequencing instead?

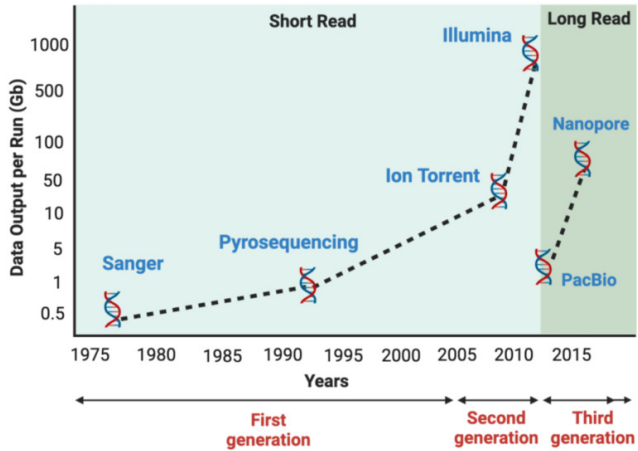
What is your application?

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



Evolution of sequencing technologies



Satam et al, Biology (Basel) 2023

Today's programme

		Alignment and mapping
0815-0900	*Lecture*	Alignment methods
0900-1000	*Exercise*	Dynamical programming of pairwise alignment *(on paper)*
1000-1030	*Lecture*	Read mapping
		Feature abundance and differential expression
1030-1100	*Lecture*	Normalization + Transformation
11-12	*Lunch*	
1200-1300	*Lecture*	Unsupervised + Supervised data exploration
1300-1445	*Exercise*	Differential expression analysis *(in R)*
1445-1500		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

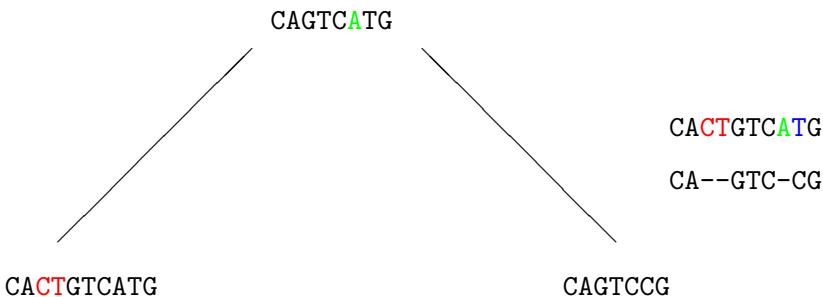
- DNA sequences change in time.
- Find evolutionary related sequences.
- Evolutionary events:

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

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

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:

```
CACTGTCATG
CA--GTC-CG
```

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

- 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

- 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				

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

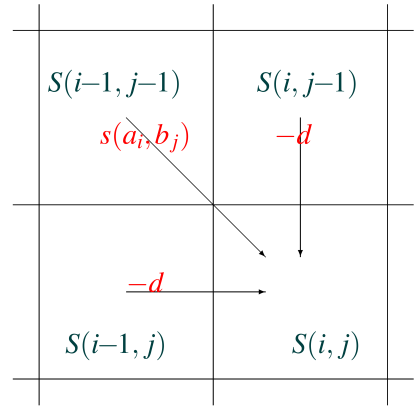
- Find the alignment between `CACTGTCATG` and `CAGTCTG` that has the maximal score?
- What would be a trivial way?
- Basic idea: Use sub sequences! $\rightarrow$ **Dynamic Programming**

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

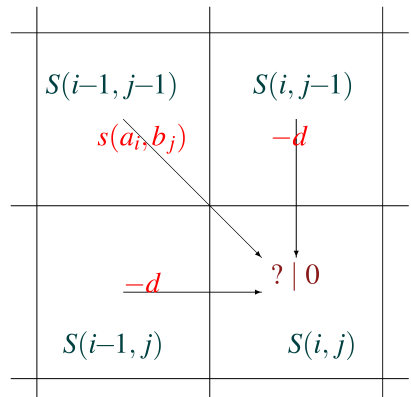
		C	A	C	T	G	T	C	A	T	G
	0	−10	−20	−30	−40	−50	−60	−70	−80	−90	−100
C	−10	**8**	−2	−12	−22	−32	−42	−52	−62	−72	−82
A	−20	−2	**16**	**6**	**-4**	−14	−24	−34	−44	−54	−64
G	−30	−12	6	13	3	**4**	−6	−16	−26	−36	−46
T	−40	−22	−4	8	21	11	**12**	2	−8	−18	−28
C	−50	−32	−14	4	11	18	13	**20**	**10**	0	−10
T	−60	−42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

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



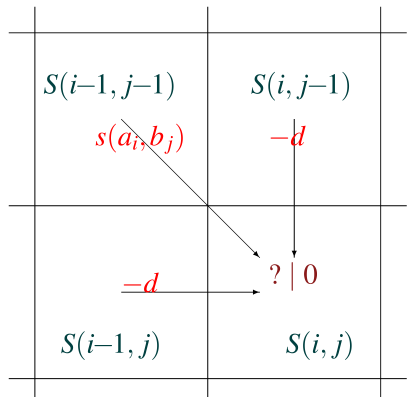
Note only positive numbers!
$i = 1, \ldots, N$; $j = 1, \ldots, M$

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:

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



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

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

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

Time is important

- Dynamic programming: exact in $O(NM)$.

When becomes time an issue?

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).

What are the default parameter settings of NCBI blastn and megablast?

Which differences in the alignments do you expect based on their parameters?

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/

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, *e.g.*, use only diagonals in dynamic programming

Exercise: Dynamical programming of pairwise alignment

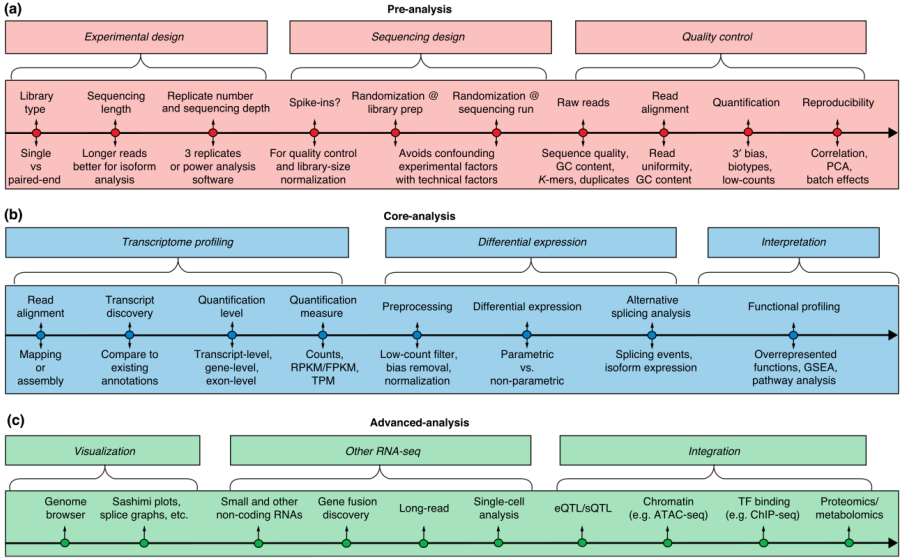
Complete the dynamic programming matrix of a global alignment:

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						

Read mapping

Roadmap for RNAseq analysis



Conesa, Ana, *et al.* "A survey of best practices for RNA-seq data analysis" Genome Biol. 2016

When becomes time an issue?

- Target is entire genome
- Target is all observed sequences (*e.g.* RefSeq non-redundant database)
- **Query are millions of reads**

How many reads do we get from modern Illumina sequencing?

	MiSeq	HiSeq 4000	NovaSeq 6000 S4
Run Time	4-56 hours	2-4 days	36-44 hours
Maximum Output	15 Gb	1500 Gb	2400-3000 Gb
Average Read Output	22 - 25 million	250 - 400 million	2,000 - 2,500 million
Maximum Read Length	2 $\times$ 300 bp	2 $\times$ 150 bp	2 $\times$ 150 bp

https://med.stanford.edu/gssc/services/sequencing1.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.

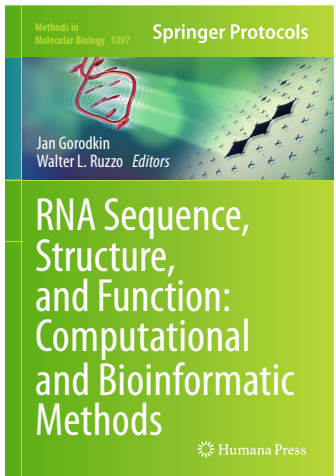
$\rightarrow$ Are optimal alignments based on quality scores still feasible?

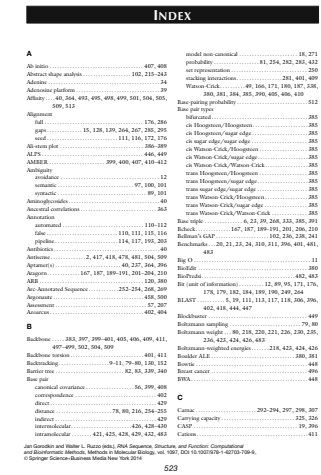
Principles of mapping reads

- Most computational time is spent on alignment.
- **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**.





INDEX

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,
© Springer Science+Business Media New York 2014

Gorodkin et al. Methods Mol Biol. 2014

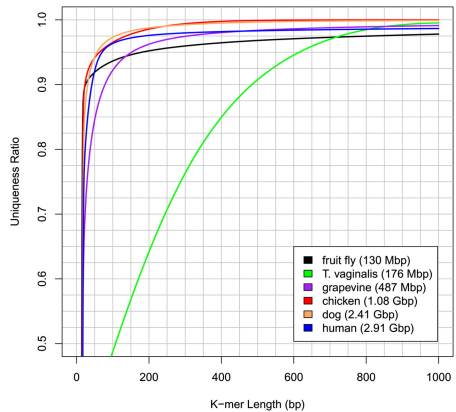
k-mer

"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

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

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.

Why might an exact mapping not always be what we want?

Which concerns might you have when mapping genomic sequence?

Which concerns might you have when mapping transcribed sequence?

Indexing problems

Flexibility and constraints:

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

Indexing method choice is crucial!

- **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

Indexing is often used for seed matching

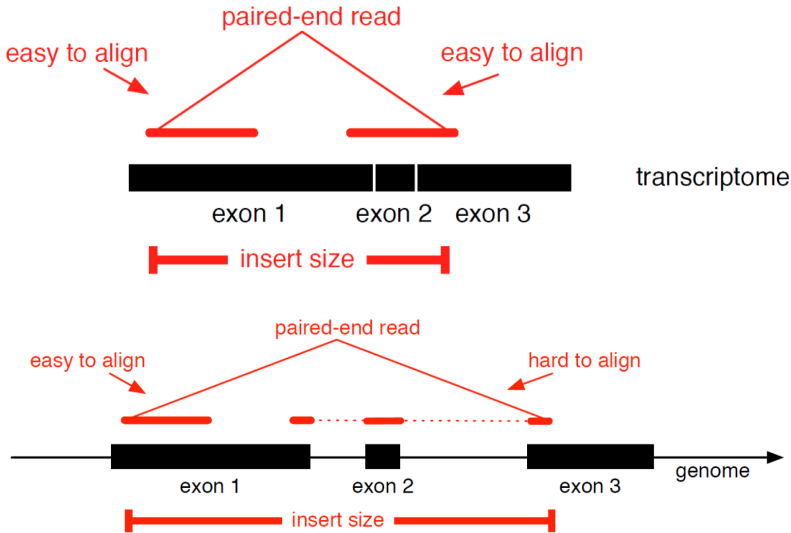
"Seed-and-extend" approach

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

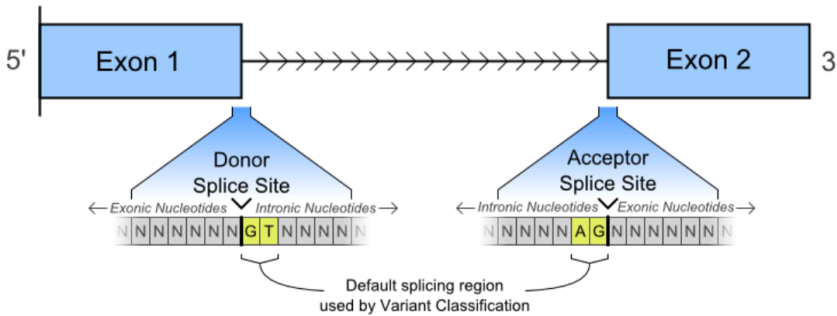
STAR, HISAT2, BLAST

Transcriptome versus genome mapping



Splice-aware genome mapping

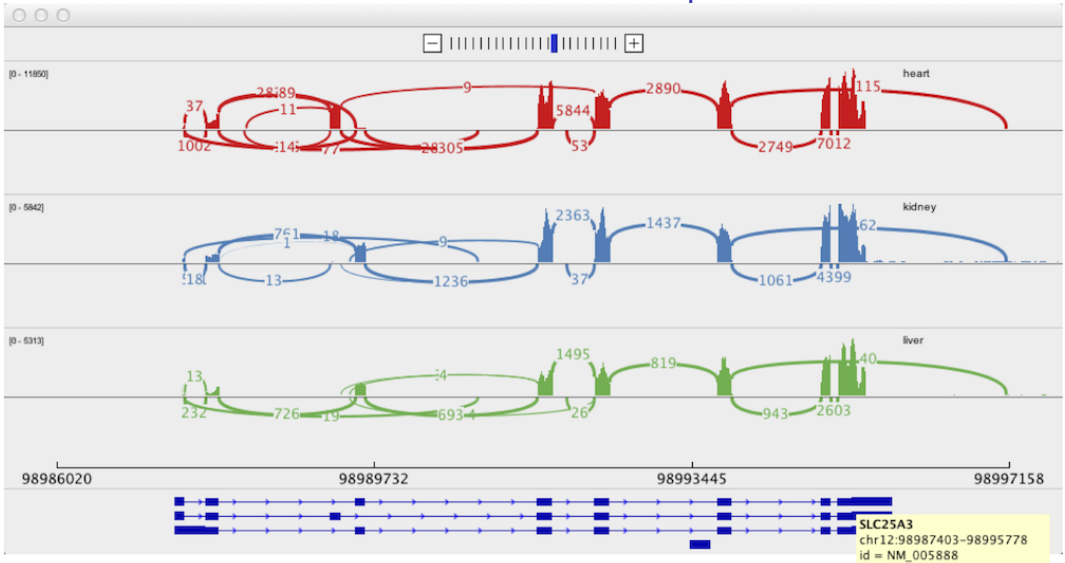
There are many mappers, but they must be able to detect splice junctions to be used in transcriptome assembly and quantification.



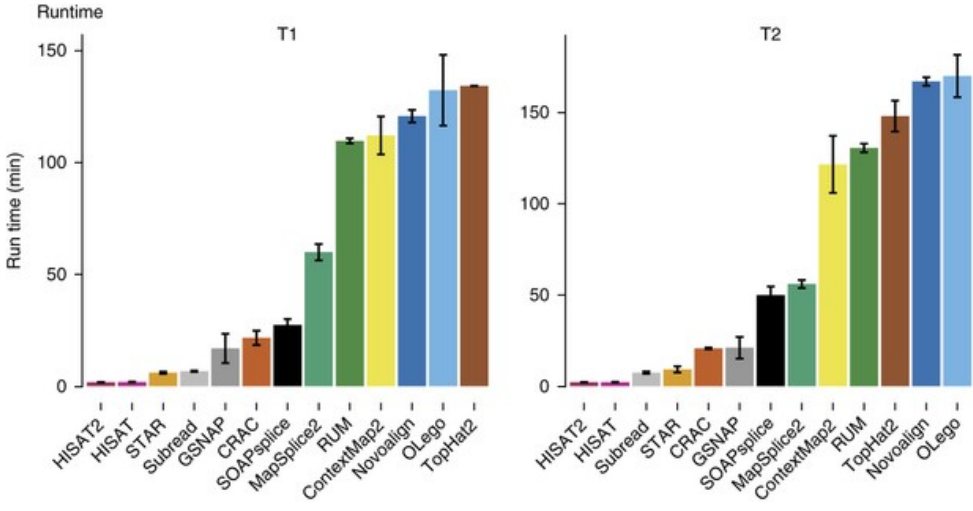
Common splice-aware alignment software

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

Sashimi Plot: Visualization of spliced reads

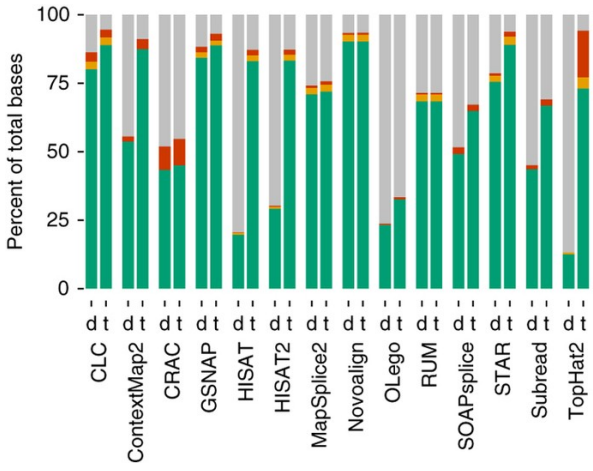


Aligner's speed



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

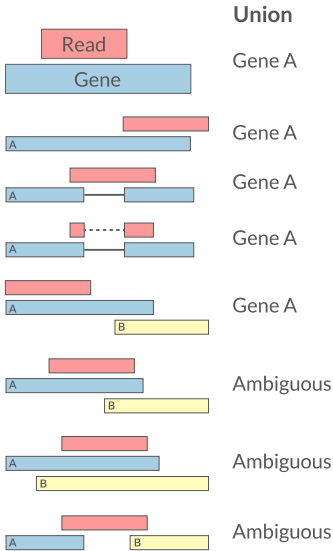
Quantification – estimation of expression

Quantification:
Count reads mapping to a genomic feature (gene, transcript, exon etc)

$\rightarrow$ genomic features can be annotated (e.g., NCBI RefSeq genome annotation) or be predicted (e.g., transcriptome assembly)

Assumption:
Number of reads produced from a feature $\sim$ feature's **relative** abundance in the sample

Gene-level quantification



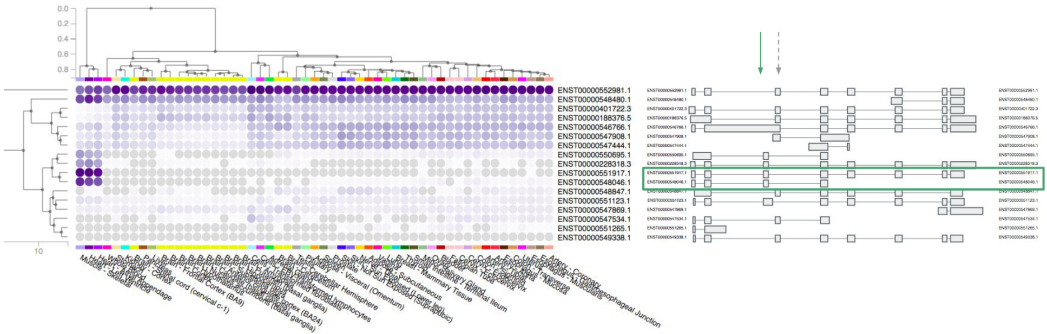
- By default ambiguous reads are not counted
- By default multi-mapping reads (reads aligning with multiple locations) are not counted

Common read counting software

featureCounts, HTSeq

Transcript-level quantification

Quantify per transcript (not just exon or gene)



Why is it important for transcript-level quantification to consider ambiguous reads?

Transcript-level quantification

- Statistical methods are used to find the probability that a read originates from a specific transcript.
- ! 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

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

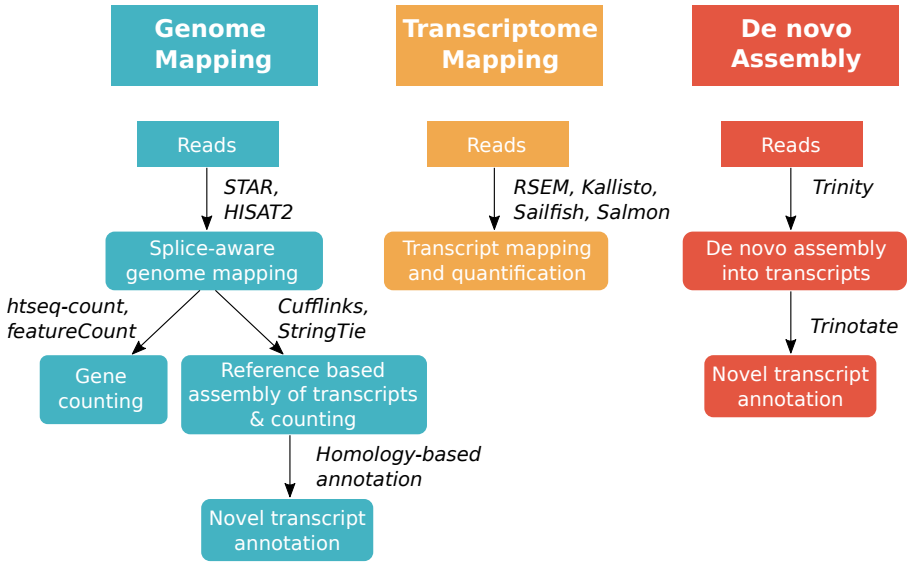
Cons

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

Common software

Salmon, Kallisto, Sailfish

Summary: From reads to count matrix



Normalization + Transformation

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

   $\rightarrow$ RNA-seq data informs about the relative abundance BUT NOT about the absolute abundance.

b. **gene length**,
   the longer the gene, more reads will be mapped

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

   $\rightarrow$ RNA-seq data informs about the relative abundance BUT NOT about the absolute abundance.

b. **gene length**,
   the longer the gene, more reads will be mapped

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/Fragments Per Kilobase transcript per Million mapped reads (RPKM/FPKM):

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

RPKM is used for single end reads
FPKM is used for paired end reads.

- First metric used in the old times (e.g. cufflinks).

Counts Per Million (CPM):

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

- Sum of all CPMs is constant (1 million).
- Does not consider gene lengths.

Simple normalization could fail

Genes	Control	Treated
Gene A	10	30
Gene B	30	90
Gene C	5	15
Gene D	1	3
Gene N	1000	240
Total	1046	378

DESeq2's median of ratios (sample-wise "size factor")

Assumption: most genes are not differential expressed

1. For each gene, calculate geometric mean

Genes	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6	Geomean
Gene 1	34	56	23	12	10	30	23
Gene 2	10	6	7	11	12	8	9
Gene n	65	78	67	34	56	23	50

2. For each gene, calculate ratio to geometric mean

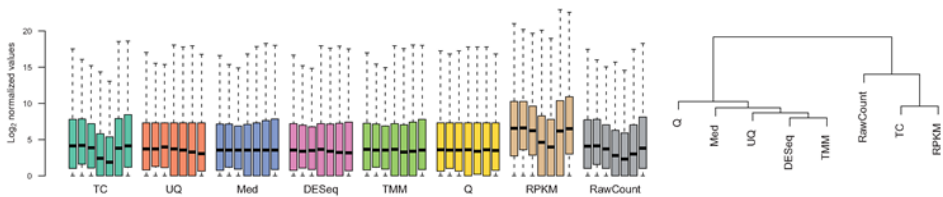
Genes	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
Gene 1	1.5	2.4	1.0	0.5	0.4	1.3
Gene 2	1.1	0.7	0.8	1.3	1.4	0.9
Gene n	1.3	1.6	1.4	0.7	1.1	0.5

3. Take median of these ratios as sample normalization factor ("size factor")

1.3	1.6	1	0.7	1.1	0.9

Summary: Normalization

- Use **raw counts** when using DGE packages, *e.g.*, DESeq2 and edgeR, as normalization is done internally
- RPKM/FPKM/CPM are not recommended for DGE analysis
- CPM is usable for visual data exploration (heatmap, abundance comparison, PCA)

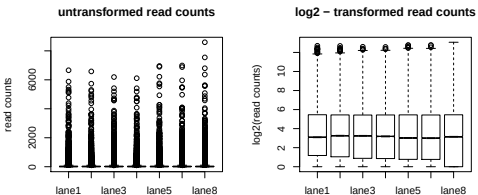


- Median of Ratios (DESeq2) and Trimmed Mean of M-values TMM (edgeR) perform the best for DGE analysis
- other solutions: spike-ins/house-keeping genes

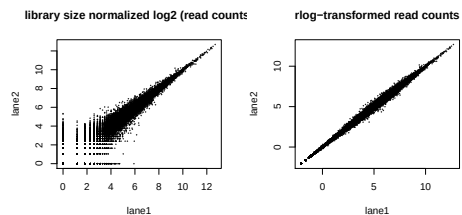
Dillies *et al.* "A comprehensive evaluation of normalization methods for Illumina high-throughput RNAseq data analysis." Brief Bioinform. 2013

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

Alternative: Compositional data analysis

DESeq's median of ratios and edgeR's TMM are less suitable in highly **asymmetrical** or **sparse** datasets → unacceptably high false positive DEGs

- ratio transformations capture the relationships between the features in the dataset
- centered log-ratio (clr) transformation:

  $\mathrm{x}_{clr} = [log(x_1/G(\mathrm{x})), log(x_2/G(\mathrm{x})) \ldots log(x_D/G(\mathrm{x}))]\,,$
  $G(\mathrm{x}) = \sqrt[D]{x_1 \cdot x_2 \cdot \ldots \cdot x_D}$
- clr-transformed values are scale-invariant (same ratio with few as well as many read counts)
- to calculate G(x) we have to delete, replace or **estimate** the 0 count values
- estimate technical variation within each sample using Monte-Carlo instances drawn from the Dirichlet distribution → probability vector prior to clr transformation

Gloor *et al.* "Microbiome Datasets Are Compositional: And This Is Not Optional" Front Microbiol 2017

Unsupervised + Supervised data exploration

Explore global and local read count patterns

1. **unsupervised**

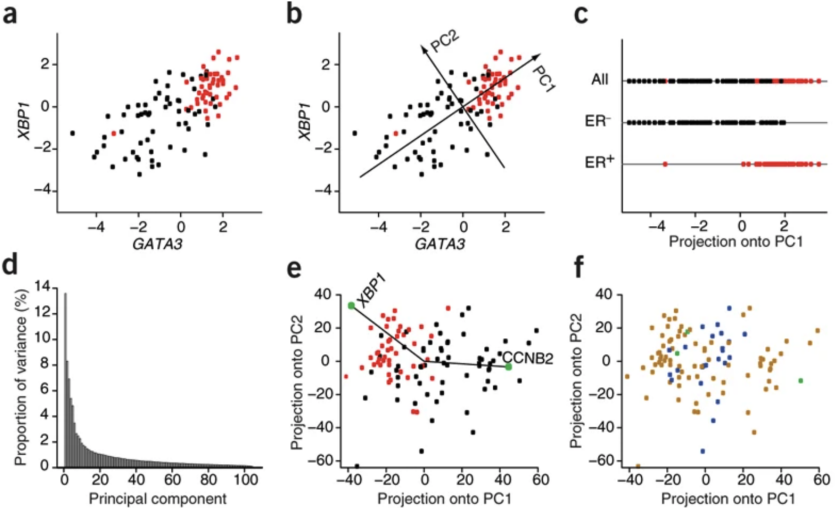
   no *a priori* information is needed
   → to detect technical noise and batch effects
   - Dimensional reduction → Principle Component Analysis (PCA)
   - Clustering → hierarchical, k-means

   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
   → differential expression

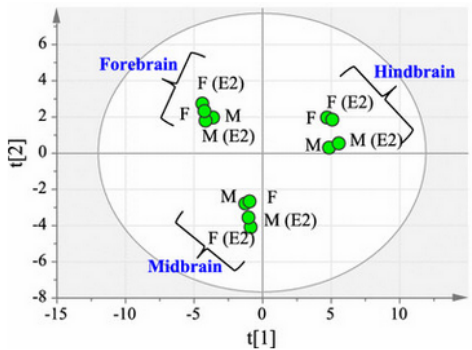
Principle component analysis – PCA



Ringnér *et al.* "What is principal component analysis?" Nature Biotechnology. 2008

Principle component analysis – PCA

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

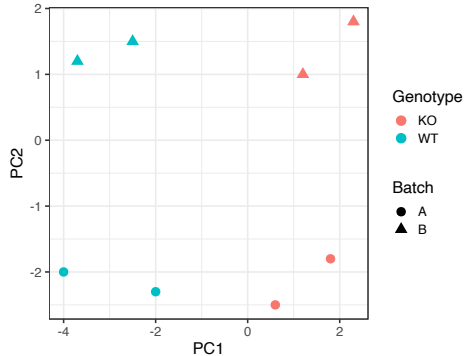


Applications:

- 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

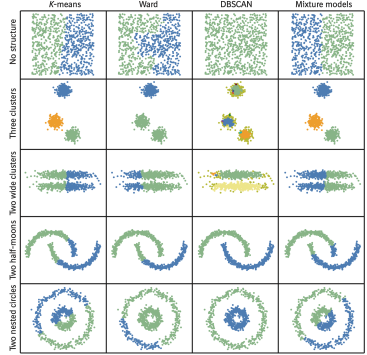
There are several clustering algorithms and more are being developed. Why?

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?

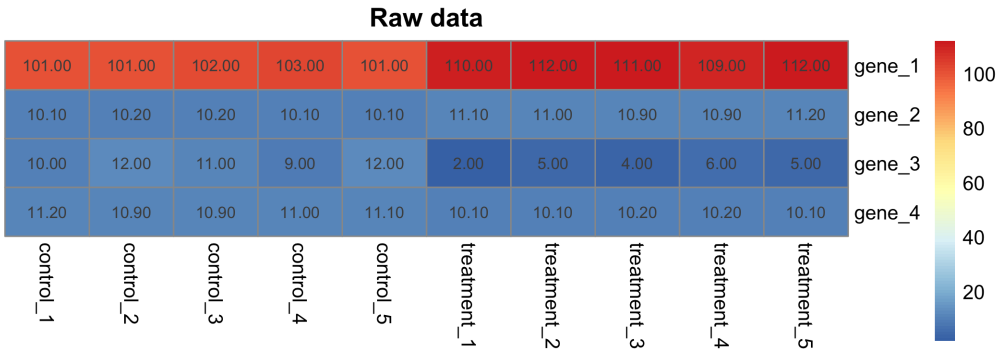
- No clustering algorithm is perfect
- Remember! Always "see" your data and judge if the clusters make sense
- The performance of the clustering algorithm depends on the structure of your data



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

Finding patterns in your data

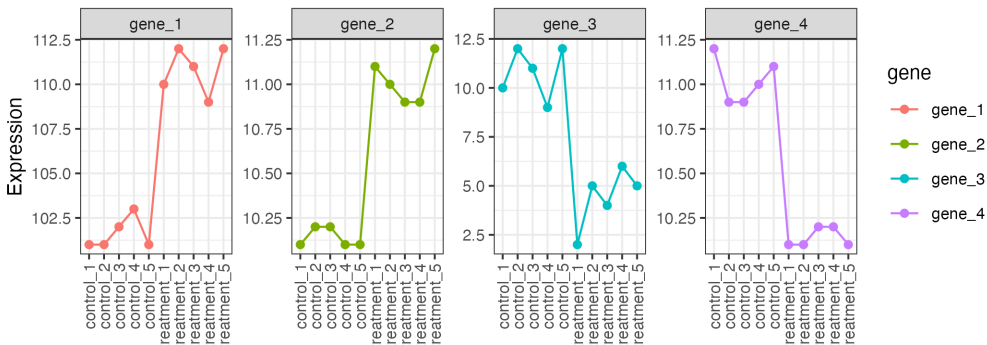
Let's assume that you got these 4 genes differentially expressed in your dataset



Can you identify a pattern?

Finding patterns in your data

Perhaps this gives you a better hint

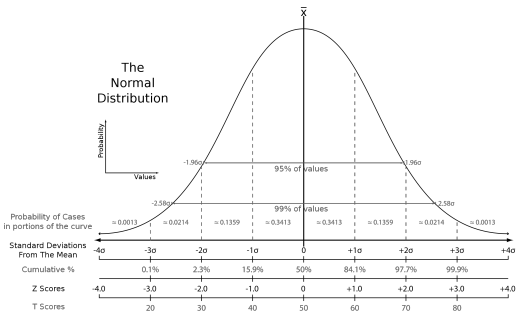


Z-score

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

μ = mean

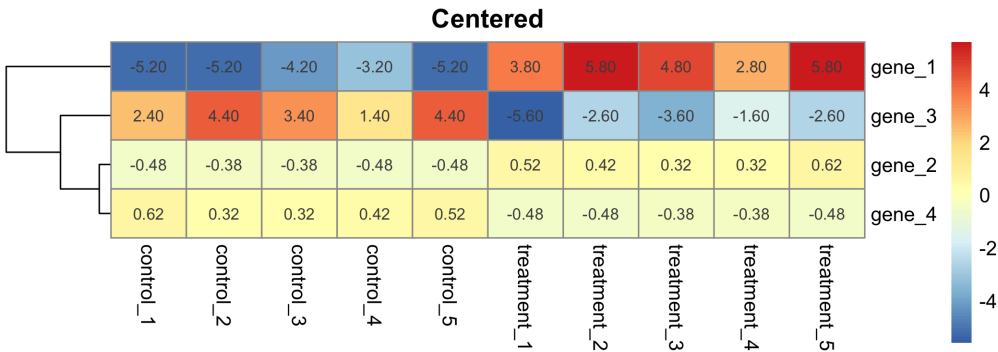
σ = standard deviation



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

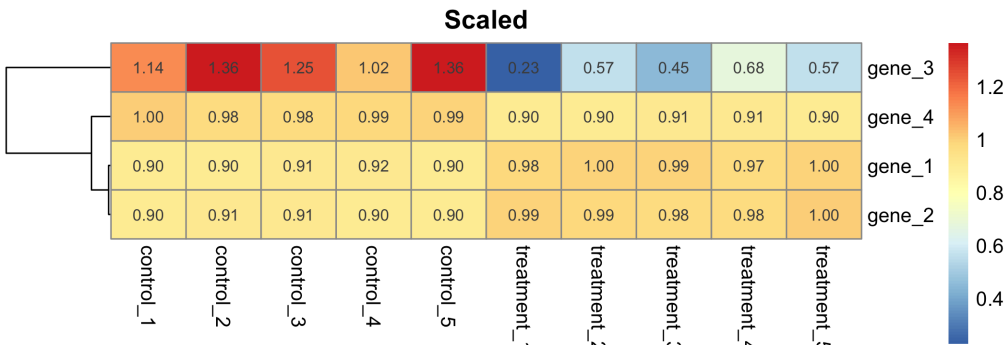
Centering data

$$x - \mu$$



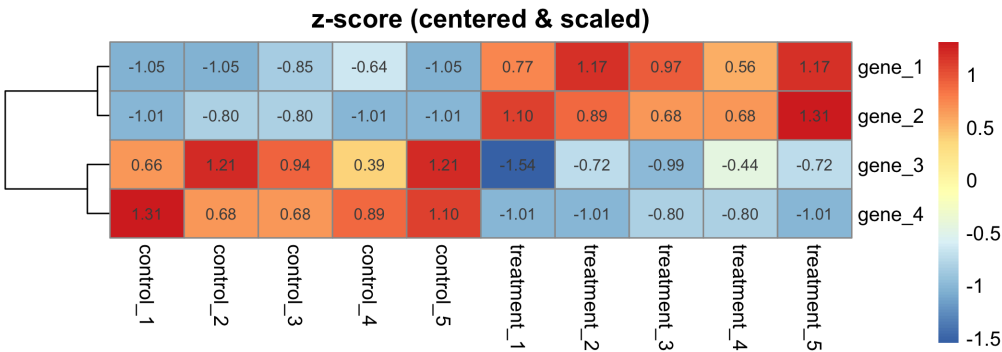
Scaling data

$\frac{x}{\sigma}$



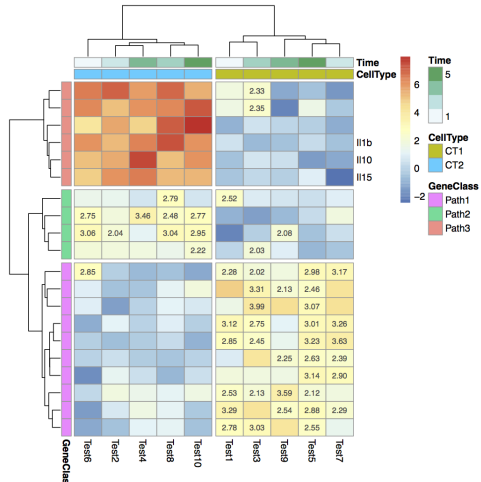
Z-score

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



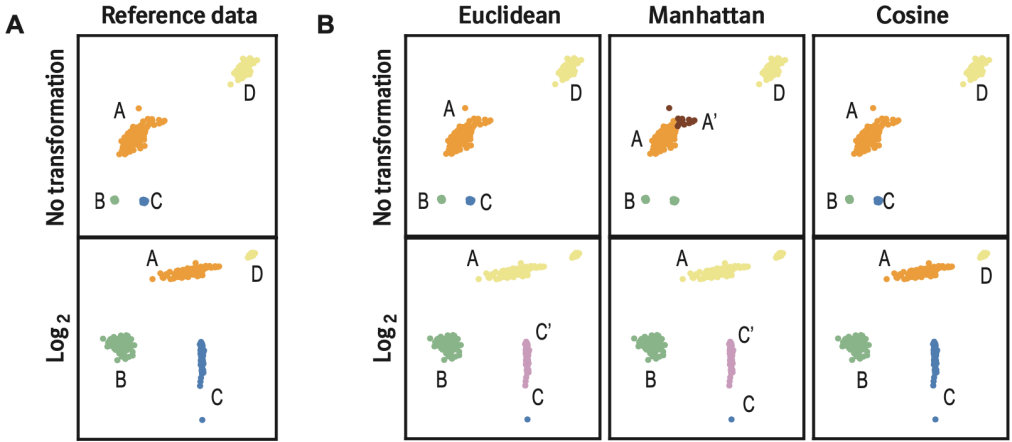
Hierarchical clustering

- No need for pre-defining the number of c830007lusters
- 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



Effect of transformations and distance metrics on clustering

- use normalized read counts (rlog, CPM etc.)
- transform your data for better scaling (log2, z-score transformation etc.)



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

Differentially expressed genes

Expression level of gene 1

Control:		**Treated:**	
Replicate 1	24	*Replicate 1*	23
Replicate 2	25	*Replicate 2*	26
Replicate 3	27	*Replicate 3*	102

Is this a differentially expressed gene?

You might get different answers depending of which software you run.

Differential gene expression

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!

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 hard to show statistical significance with non-parametric methods if only few replicates are available (less than 8).

Issues with DGE analysis 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 ...

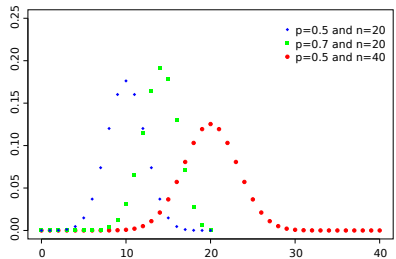
Binomial distribution

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)



As RNA-seq experiments produce large number of reads (n is large) the Gaussian distribution can replace the binomial.

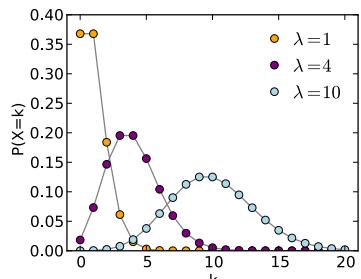
Poisson distribution

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:

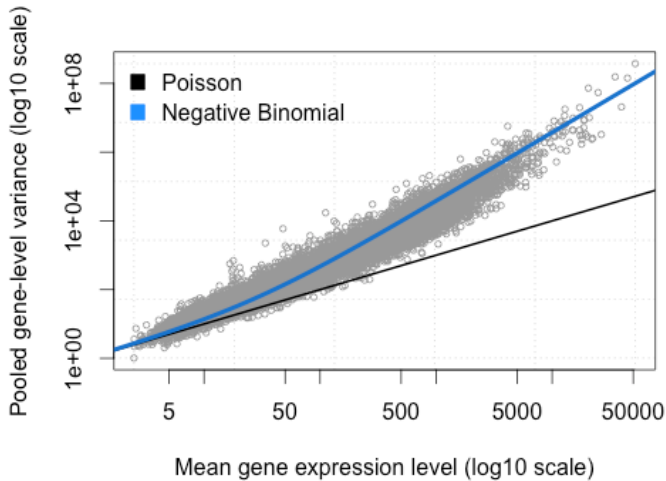
$$\lambda = np$$

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**.



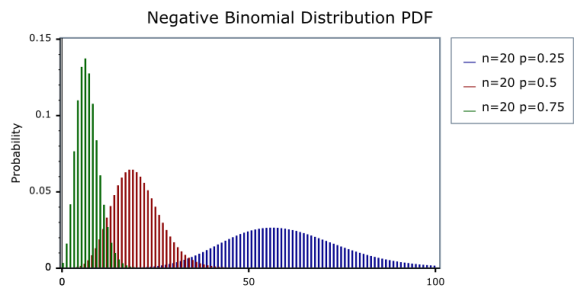
Negative binomial distribution

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 r successes (and k failures) if the probability of a single success is p:

$$NB(X = k) = \binom{k + r - 1}{k} p^r (1 - p)^k$$

both the mean and variance can be calculated from r and p



Fitting a negative binomial GLM

Raw count for gene *i* in sample *j*

Controls the variance

$$K_{ij} \sim \mathrm{NB}\left(\text{mean} = \mu_{ij}, \text{dispersion} = \alpha_i\right)$$

$$\mu_{ij} = s_j q_{ij}$$

Normalization ("size") factor

Normalized count

Design matrix

-- Control or Treatment?

-- Batch (e.g., flow cell, plate, lab)

-- Other co-factors (e.g., gender)

Linear Model:

$$\log_2 q_{ij} = \sum_r x_{jr} \beta_{ir}$$

Coefficient r

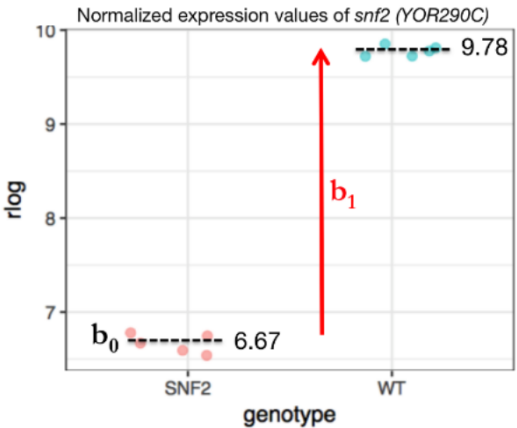
Generalized Linear Model (GLM) coefficients

-- One for each Design matrix column (factor)

= strength of effect

= **log2 fold change** for each gene

Interpretation of the negative binomial GLM



Linear regression model (LM) is evaluated for every gene: $Y = b_0 + b_1 * x_1 + e$

Y . . . describes all read counts for a gene
b_0 . . . average of baseline group, *e.g.*, control
x_1 . . . design factor, *e.g.*, condition (often 0 or 1)
b_1 . . . coefficent that captures the difference between different conditions
e . . . error or uncertainty

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

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

Design & contrast matrix

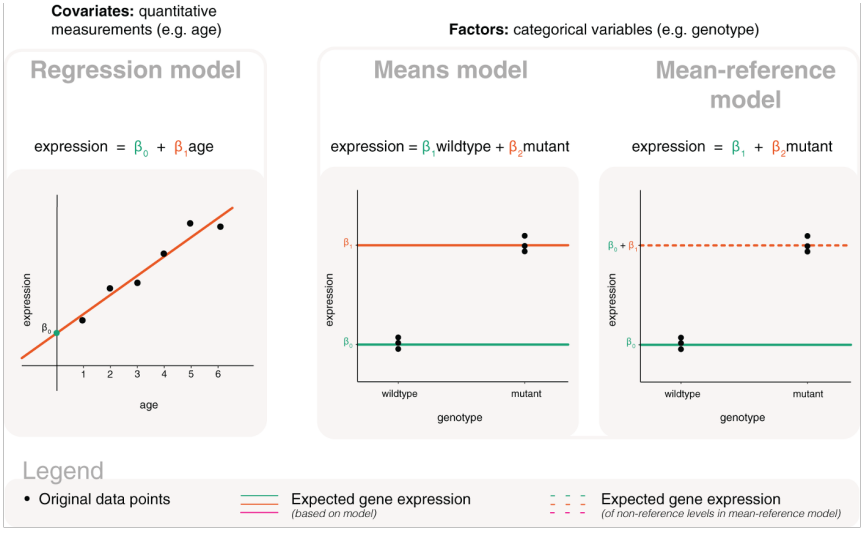
Design matrix (a.k.a. model matrix) has 2 main roles:

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 $group_2$

Basic regression models



Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix with intercept

x is an indicator variable for **sick** mice:

- x = 1 for sick mice
- x = 0 otherwise

Model

E(y) = 2.95 + 1.62x

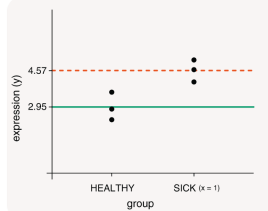
E(y) = 2.95	= 2.95	*(for healthy group)*
E(y) = 2.95 + 1.62	= 4.57	*(for sick group)*

Matrix

```
> model.matrix(~group)
```

	(Intercept)	groupSICK
1	1	0
2	1	0
3	1	0
4	1	1
5	1	1
6	1	1

Plot



contrast = c(0,1)

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

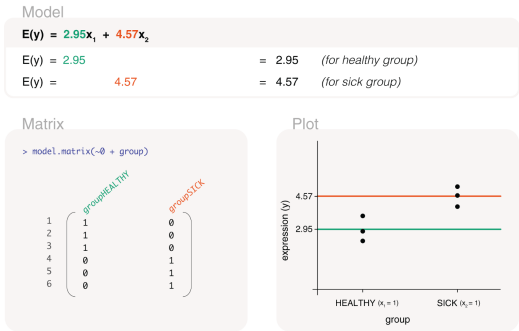
Design matrix without intercept

x1 is an indicator variable for **healthy** mice:

- x1 = 1 for healthy
- x1 = 0 otherwise

x2 is an indicator variable for **sick** mice:

- x2 = 1 for sick
- x2 = 0 otherwise



contrast = c(-1,1)

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix with intercept

Model

$$E(y) = 1.03 + 1.09x_1 + 1.97x_2 + 3.87x_3$$

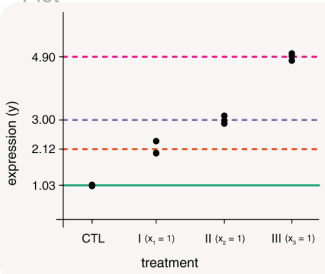
$E(y) = 1.03$	$= 1.03$	*(for control)*
$E(y) = 1.03 + 1.09$	$= 2.12$	*(for treatment I)*
$E(y) = 1.03 + 1.97$	$= 3.00$	*(for treatment II)*
$E(y) = 1.03 + 3.87$	$= 4.90$	*(for treatment III)*

Matrix

```
> model.matrix(~treatment)
```

	(Intercept)	treatmentI	treatmentII	treatmentIII
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	1	0	0
5	1	1	0	0
6	1	1	0	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	0
10	1	0	0	1
11	1	0	0	1
12	1	0	0	1

Plot



$$contrast_{treatment_{III}\ vs\ control} = c(0, 0, 0, 1)$$

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix without intercept

Model

$E(y) = 1.03x_0 + 2.12x_1 + 3.00x_2 + 4.90x_3$

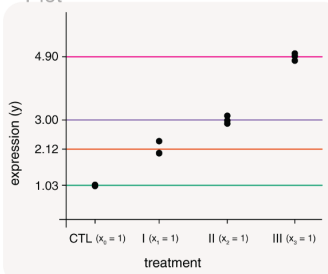
E(y) =	1.03				= 1.03	*(for control)*
E(y) =		2.12			= 2.12	*(for treatment I)*
E(y) =			3.00		= 3.00	*(for treatment II)*
E(y) =				4.90	= 4.90	*(for treatment III)*

Matrix

```
> model.matrix(~0 + treatment)
```

	treatmentCTL	treatmentI	treatmentII	treatmentIII
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	0	1	0	0
5	0	1	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	1	0
9	0	0	1	0
10	0	0	0	1
11	0	0	0	1
12	0	0	0	1

Plot

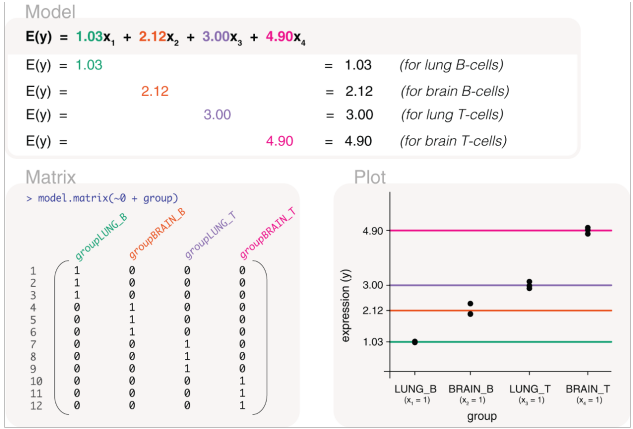


$$contrast_{treatment_{III}\ vs\ control} = c(-1, 0, 0, 1)$$

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix for multiple covariates

group factor is converted from two factors representing **tissue** samples and **cell types**



$$contrast_{BRAIN\_B \ vs \ BRAIN\_T} = c(0, -1, 0, 1)$$

$$contrast_{LUNG \ vs \ BRAIN} = c(0.5, -0.5, 0.5, -0.5)$$

Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix for cyclic time series

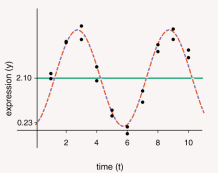
Model

$$E(y) = 2.10 + 0.53\sin(\pi/3\ t) + -1.87\cos(\pi/3\ t)$$

Matrix

```
> model.matrix(~sinphase + cosphase)
   (Intercept)        time       time2
1            1        0.87         0.5
2            1        0.87         0.5
3            1        0.87        -0.5
4            1        0.87        -0.5
5            1     1.2e-16        -1.0
6            1     1.2e-16        -1.0
7            1       -0.87        -0.5
8            1       -0.87        -0.5
9            1       -0.87         0.5
10           1       -0.87         0.5
11           1     -2.4e-16         1.0
12           1     -2.4e-16         1.0
13           1        0.87         0.5
14           1        0.87         0.5
15           1        0.87        -0.5
16           1        0.87        -0.5
17           1     3.7e-16        -1.0
18           1     3.7e-16        -1.0
19           1       -0.87        -0.5
20           1       -0.87        -0.5
```

Plot



Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Design matrix for cyclic time series

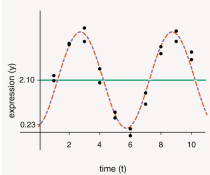
Model

$$E(y) = 2.10 + 0.53\sin(\pi/3\ t) + -1.87\cos(\pi/3\ t)$$

Matrix

```
> model.matrix(~sinphase + cosphase)
   (Intercept)    time         timeZ
1            1    0.87           0.5
2            1    0.87           0.5
3            1    0.87          -0.5
4            1    0.87          -0.5
5            1 1.2e-16          -1.0
6            1 1.2e-16          -1.0
7            1   -0.87          -0.5
8            1   -0.87          -0.5
9            1   -0.87           0.5
10           1   -0.87           0.5
11           1 -2.4e-16           1.0
12           1 -2.4e-16           1.0
13           1    0.87           0.5
14           1    0.87           0.5
15           1    0.87          -0.5
16           1    0.87          -0.5
17           1 3.7e-16          -1.0
18           1 3.7e-16          -1.0
19           1   -0.87          -0.5
20           1   -0.87          -0.5
```

Plot



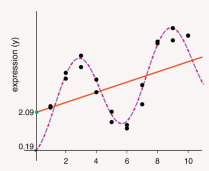
Model

$$E(y) = 2.09 + 0.25t + 0.45\sin(\pi/3\ t) + -1.90\cos(\pi/3\ t)$$

Matrix

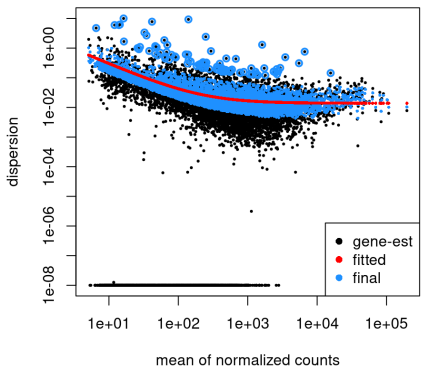
```
> model.matrix(~time+ sinphase + cosphase)
   (Intercept)    time sinphase cosphase
1            1    1      0.87       0.5
2            1    1      0.87       0.5
3            1    2      0.87      -0.5
4            1    2      0.87      -0.5
5            1    3   1.2e-16      -1.0
6            1    3   1.2e-16      -1.0
7            1    4     -0.87      -0.5
8            1    4     -0.87      -0.5
9            1    5     -0.87       0.5
10           1    5     -0.87       0.5
11           1    6   -2.4e-16       1.0
12           1    6   -2.4e-16       1.0
13           1    7      0.87       0.5
14           1    7      0.87       0.5
15           1    8      0.87      -0.5
16           1    8      0.87      -0.5
17           1    9   3.7e-16      -1.0
18           1    9   3.7e-16      -1.0
19           1   10     -0.87      -0.5
20           1   10     -0.87      -0.5
```

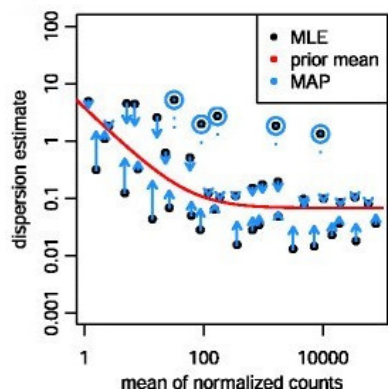
Plot



Law CW, Zeglinski K, Dong X *et al.*A guide to creating design matrices for gene expression experiments. F1000Research 2020, 9:1444

Dispersion estimate





- Not enough replicates to estimate dispersion for individual genes
- Borrow information from genes of similar expression strength among the replicates
- Genes with very high dispersion left as is

Implementation of DGE 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.
- *Statistical test to examine if the changes are statistically significant:* `DESeq2` provides the Wald test or the likelihood ratio test; `edgeR` uses quasi-likelihood (QL) F-test or likelihood ratio test
- *Multiple testing issue:* All current packages report false discovery rate FDR (most often Benjamini-Hochberg corrected p values).

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: $\alpha_{multiple} = 1 - (1 - \alpha)^n = 0.64$

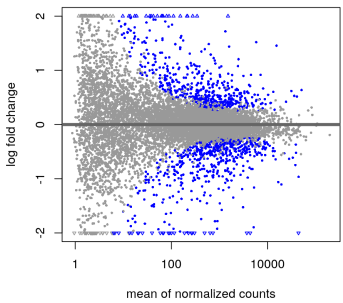
Multiple testing issue

- 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: $\alpha_{multiple} = 1 - (1 - \alpha)^n = 0.64$

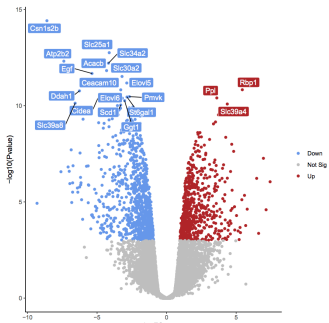
- If the number of tests (m) increases, the type-I error rate $\alpha_{multiple}$ will reach to 1
- This inflation of α has to be handled by multiple testing correction for p-values
- Most applied method for omics studies is Benjamini-Hochberg method (a.k.a. False Discovery Rate, FDR)

Visualization of DE analysis

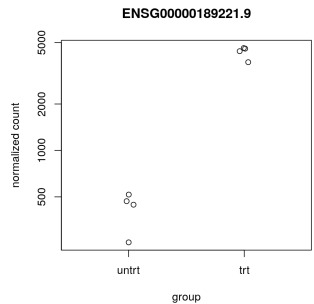
MA-plot (Bland-Altman)



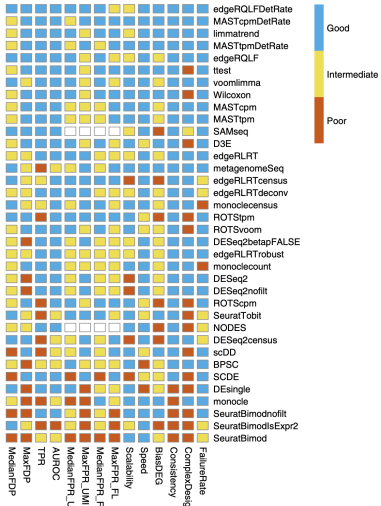
Volcano plot



Normalised counts



An overview of statistical tests for DGE analysis

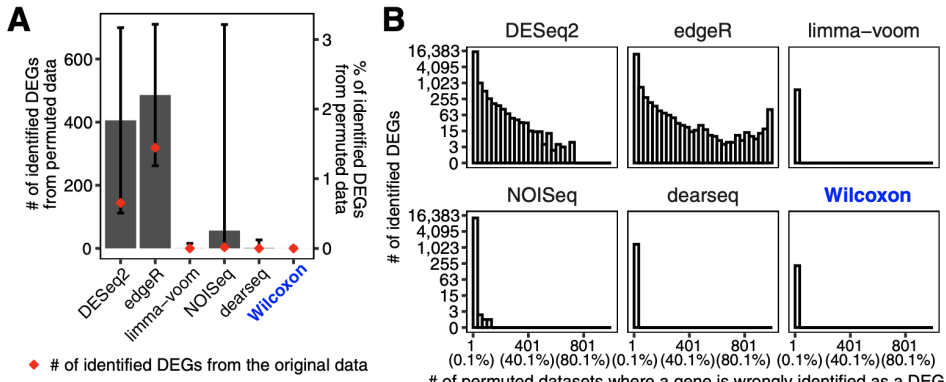


- Classical tests (t-test, Wilcoxon) still perform well
- However, they cannot handle complex designs
- Advanced methods (e.g. DESeq2, edgeR, limma) can handle complex experimental designs
- Choose your method carefully based on your needs
- If you don't know what to do, advanced methods are still the way to go

Soneson & Robinson. Bias, robustness and scalability in single-cell differential expression analysis. Nature Methods (2018)

Very large sample sizes

- population-level RNA-seq studies
- single cell RNA-seq



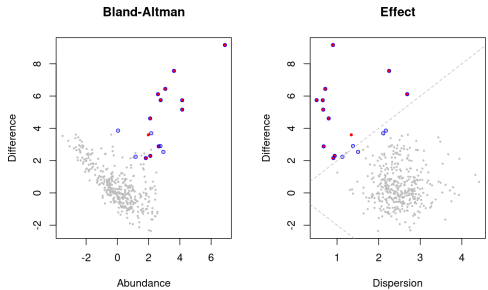
→ Non-parametric approaches, *e.g.*, Wilcoxon rank-sum test, perform best.

Li *et al.* Exaggerated false positives by popular differential expression methods when analyzing human population samples. Genome Biol (2022)

Alternative: Compositional data analysis

ALDex for differential expression analysis:

1. Add a small prior count to the observed counts for taxa j across all samples
2. Draw Monte Carlo samples using the Dirichlet distribution
3. Transform the samples using the Centered log ratio (CLR) transform
4. Hypothesis testing, *e.g.*, Welch's t-test or Wilcoxon rank test
5. Report expected values from statistical tests and effect-size estimate



https://bioconductor.org/packages/release/bioc/vignettes/ALDEx2/inst/doc/ALDEx2_vignette.html

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 2025; BAGED-single October 2024
6. Sign up early due to limited number of seats