Sequence Analysis 1

A. Concepts, conservation & substitution

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Lecture outline

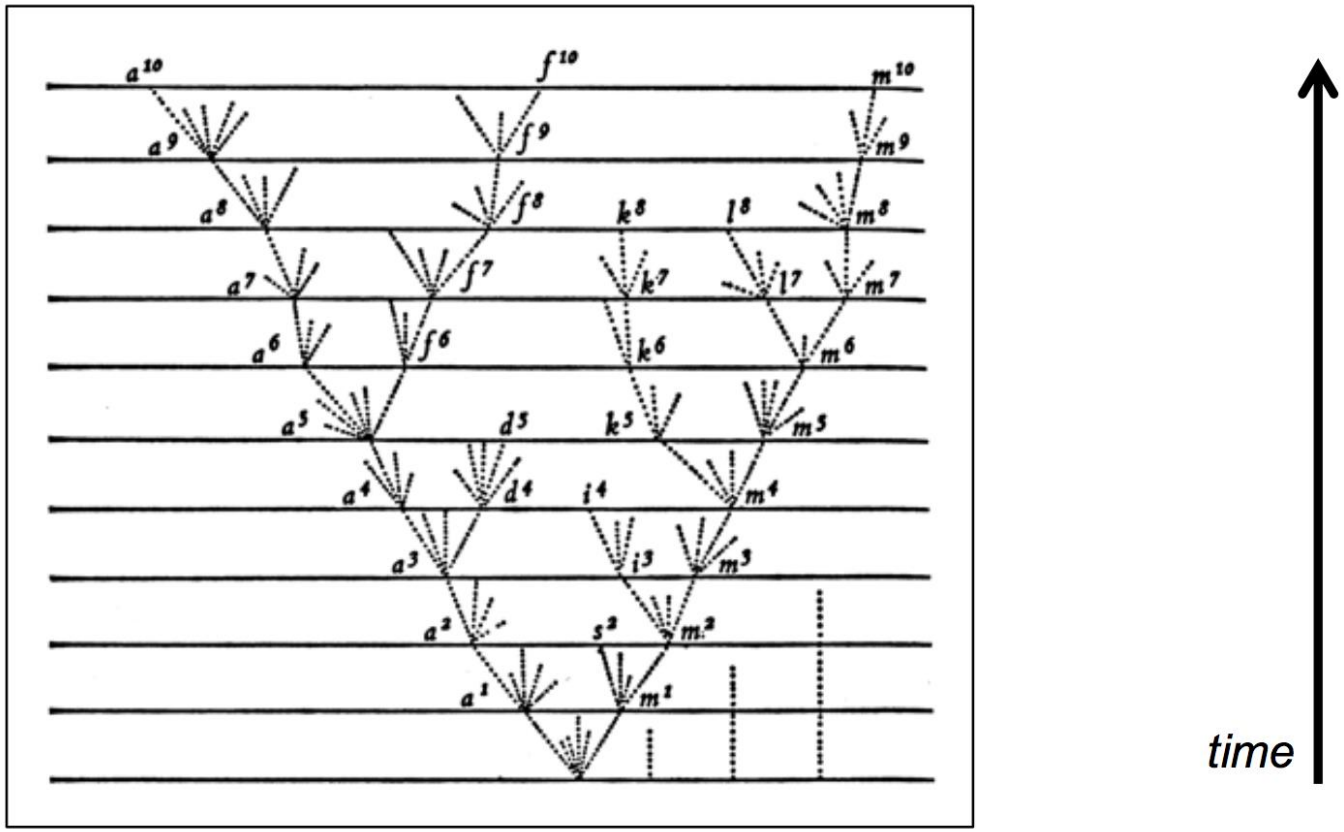
- **Concept of homology**
  - Homology and sequence similarity
- **Sequence change in evolution**
  - Random change versus biological evolution
  - Sequence change versus conservation
- **Quantifying sequence change**
  - Observed versus null probabilistic models
  - Log-odds score
  - Basic principles of PAM and BLOSUM matrices, and their differences

Concept of homology

Organisms related by genealogical descent with modification

Traits (and genes) are passed on from one generation to the next (**inheritance**)

Charles Darwin (1859). *On the Origin of Species*



Features derived from a common ancestor are said to be **homologous**. This applies to any feature – morphological or molecular (genes, RNAs, proteins).

Homology and sequence similarity

- **Sequence similarity (*or* sequence identity):** a measure of identical residues shared between two sequences, usually in an alignment (where identical/similar residues were aligned); the extent to which two sequences are **invariant**

Example:

```
Seq1  ACGTAGCTAGCTAGCTACCT
Seq2  ACGTAGCTAGCTAGCTAGCT
```

%identity between Seq1 & Seq2 **= 19/20 = 95%**

- High level of sequence similarity **usually**, but **not necessarily**, indicates evidence for homology
- Similar sequences may be homologs

Homology and sequence similarity

- **Conservation:** unchanged/invariant positions when comparing two sequences. At amino acid level, this can also refer to changes at an amino acid position that preserves the physicochemical property

Phenylalanine (**F**) Valine (**V**) are hydrophobic

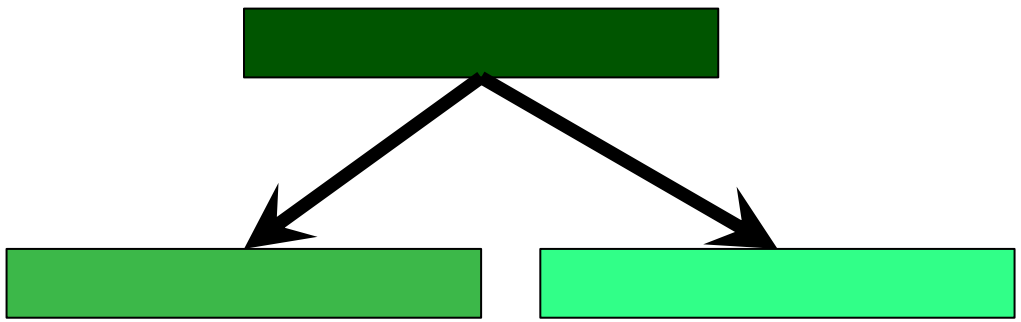
```
Seq1 ANYWQFPDGIYYEGCS
Seq2 ANYWQVPDGIHYEGCS
Seq3 ANYWQFPDGIYYEGCS
     ******:*****:******
```

Tyrosine (**Y**) and Histidine (**H**) are polar

Homology is an evolutionary relationship that either exists or not (i.e. it is **all-or-nothing**, there are no “degrees of homology”). We may be able to quantify how confident we are in believing that two molecules/sequences are homologous, but they are nonetheless either homologous or non-homologous.

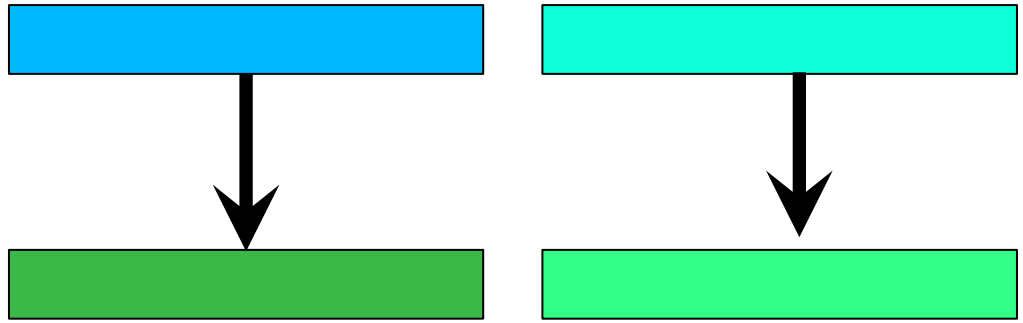
Sequence change in evolution

Two alternative explanations to observed sequence similarity:



Evolution

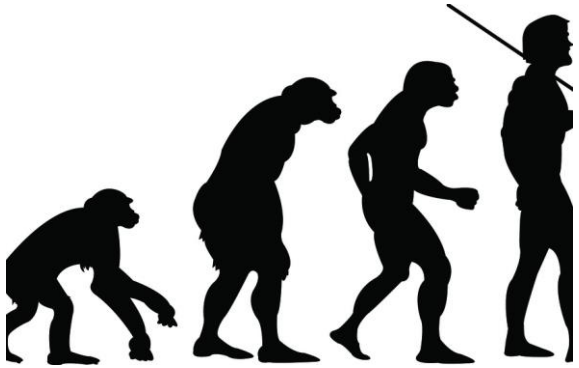
similarity is due to shared ancestry



Random

similarity is coincidental by pure chance

Compare the probabilities of the two hypotheses:



$P_{evolution}$ what we observed

versus

P_{random} the null model

Random sequence change

What are the chances of randomly matching 2 basepairs (bp) in a human-size genome (3 x 10^9 bp)?

Total genome size

$$\frac{3{,}000{,}000{,}000}{4^2} = 187{,}500{,}000$$

number of possible matches

4 possible nucleotides

Length of matched region

$$\frac{187{,}500{,}000}{3{,}000{,}000{,}000} \times 100 = 6.25\%$$

Probability of a match to occur at random

Random sequence change

*What are the chances of randomly matching **N** basepairs (bp) in a human-size genome (3 x 10^9 bp)?*

Length of match in bp (N)	**# possible matches ($3 \times 10^9 / 4^N$)**	**Probability (P_{random})**
1	750,000,000	25.0 %
2	187,500,000	6.25 %
10	2,861	9.5×10^{-5} %
20	2.73×10^{-3}	9.1×10^{-11} %
300	7.23×10^{-172}	2.4×10^{-179} %

Random change or biological evolution?

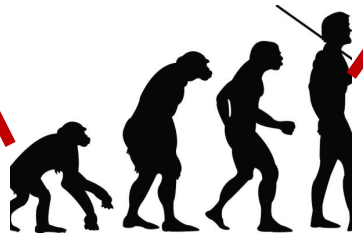
Chimpanzee genome size ≈ human genome size ≈ **3Gbp**.
Consider a **300bp**-region in chimpanzee and in human:

%identity
$= 296/300 \times 100\%$
$= \mathbf{98.67\%}$

GGCTGTCATCACTTAGACCTCACCCTGTGG	GGCTGTCATCACTTAGACCTCACCCTGTGG
AGCCACACCCTAGGGTTGGCCAATCTACTC	AGCCACACCCTAGGGTTGGCCAATCTACTC
CCAGGAGCAGGGAGGGCAGGAGCCAGGGCT	CCAGGAGCAGGGAGGGCAGGAGCCAGGGCT
GGGCATAAAAGTCAGGGCAGAGCCATCTAT	GGGCATAAAAGTCAGGGCAGAGCCATCTAT
TGCTTACATTTGCTTCTGACACAACTGTGT	TGCTTACATTTGCTTCTGACACAACTGTGT
TCACTAGCAACCTCAAACAGACACCATGGT	TCACTAGCAACCTCAAACAGACACCATGGT
GC**AT**CTGACTCCTGAGGAGAAGTCTGCCGT	**A**C**CG**CTGACTCCTGAGGAGAAGTCTGCCGT
TACTGCCCTGTGGGGCAAGGTGAACGTGGA	TACTGCCCTGTGGGGCAAGGTGAACGTGGA
TGAAGTTGGTGGTGAGGCCCTGGGCAGGTT	TGAAGTTGGTGGTGAGGCCCTGGGCAGGTT
GGTATCAAGGTTACAAGACAGG**T**TTAAGGA	GGTATCAAGGTTACAAGACAGG**C**TTAAGGA

Example

P_{random} of 296bp identical match in a 3Gbp genome
$= 6.17 \times 10^{-177}\ \%$



Do you think human and chimpanzee share a common ancestry?

Sequence change *versus* conservation

DNA/RNA sequence

↓

Protein sequence

↓

Tertiary structure

↓

Function

- Selective pressure for **divergence** e.g. genetic diversity (to increase viability), e.g. fast-evolving genes
- Substitutions, insertions, deletions, translocations, genetic transfers/exchange, etc.
- **Adaptation** due to changes in environments (biotic and abiotic stressors)
- Copy/replication errors

- Selective pressure for **preservation of critical gene function**, protein structure and function, including non-coding sequences and regulatory elements
- Especially true for critical machineries and slow-evolving genes e.g. housekeeping genes, ribosomal RNA genes (phylogenetic markers)

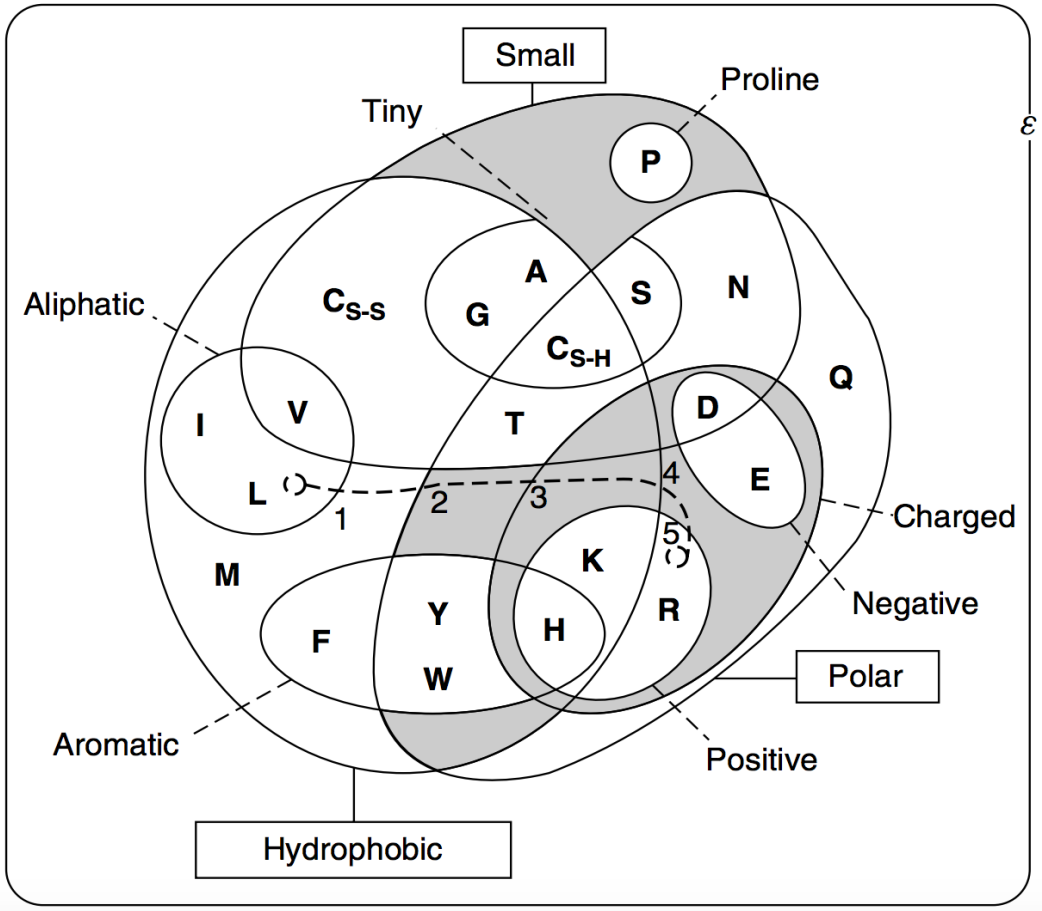
Quantifying sequence change

Homology and sequence conservation is commonly observed at the **protein** level. *Why?*

- **Codon degeneracy**: nearly one-third of the bases in coding regions are under a weak (if any) selection
- **Greater information content**: 20 amino acids versus 4 bases

Among a set of homologous sequences: **observed substitution frequency** for each amino acid can be used to quantify sequence change

- Commonly weighted based on shared **physicochemical properties** of amino acids

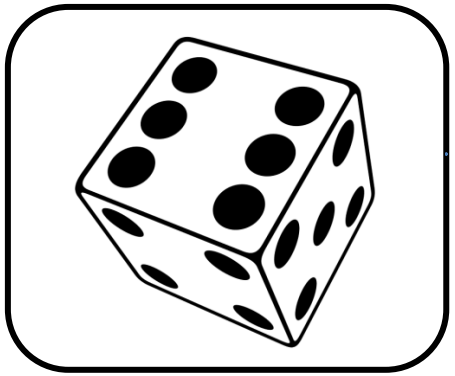


Betts & Russell (2003) Chapter 14. *Bioinformatics for Geneticists*.In: Barnes MR & Cray IC (Eds). John Wiley & Sons. Chapter 13 in the 2nd Edition (QH430 .B375 2007)

Probability in an unbiased null model

On a die:

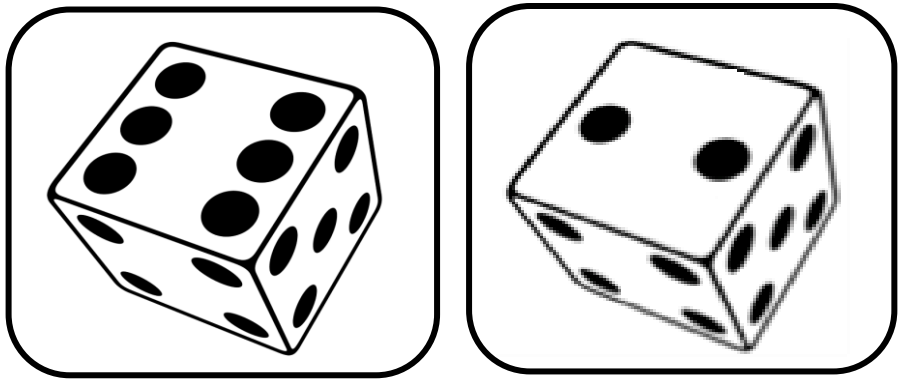
Landing a **6**?



1/6

P(‘6’) = 1/6

Landing a **6** then a **2**?



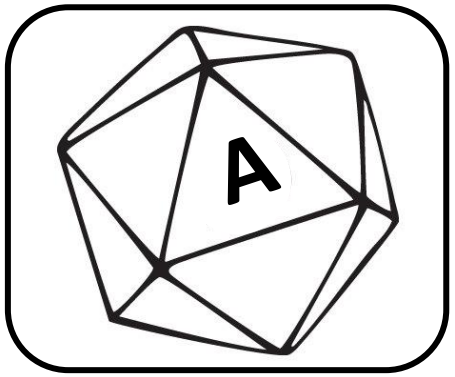
1/36

P(‘6’ ∧ ‘2’) = 1/6 × 1/6
= 1/36

Probability in an unbiased null model

Imagine a 20-face die (each face represents an amino acid)

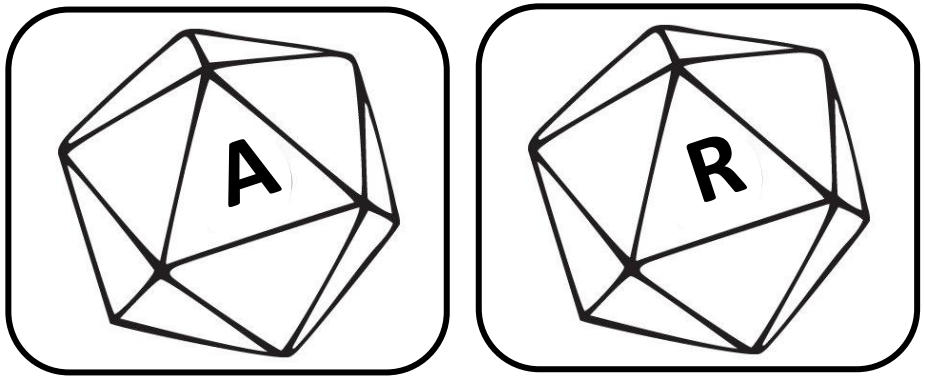
Observing an **A**?
(Alanine)



1/20

$$P(\text{'A'}) = p_A = 1/20$$

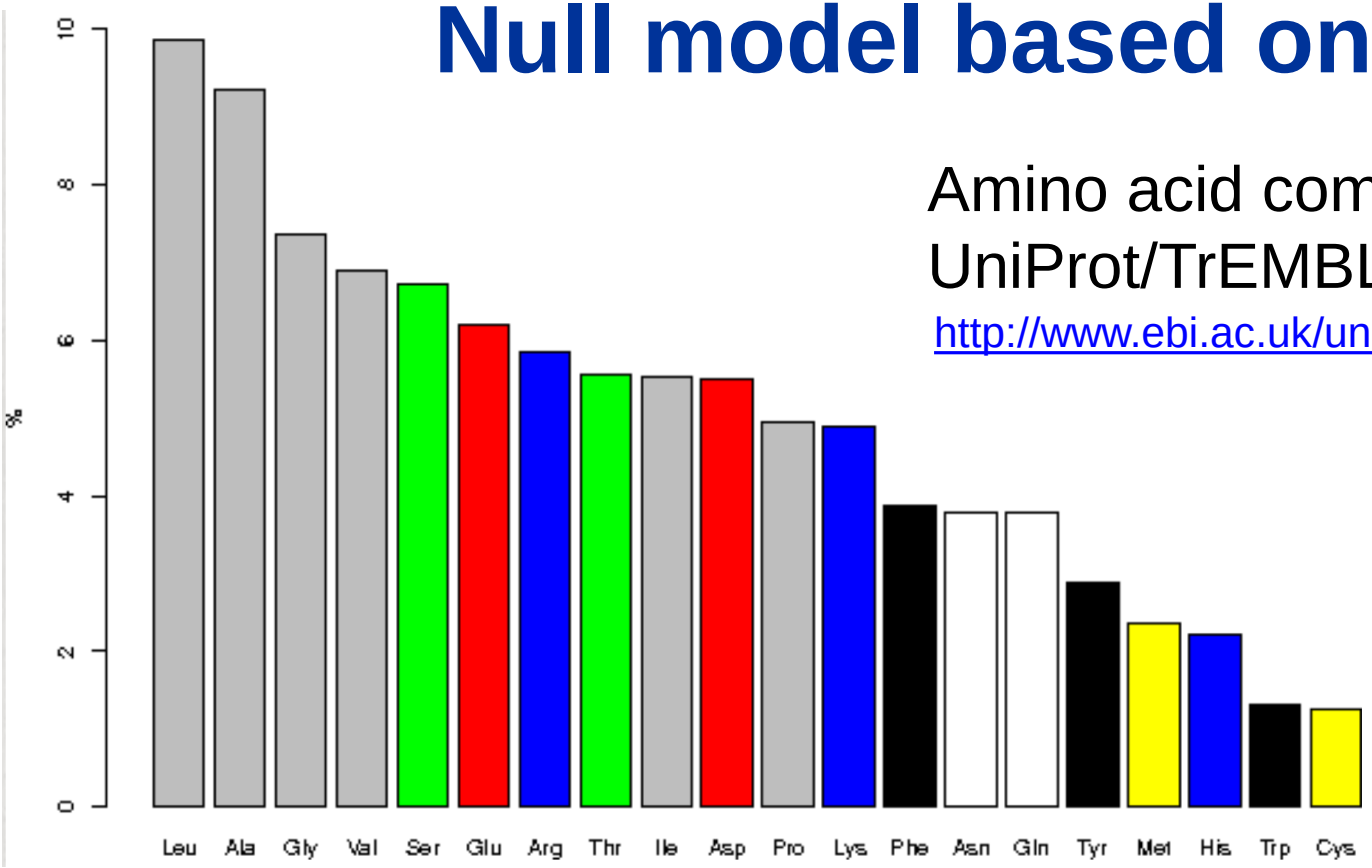
Observing **A** in one but **R** in another?
(Arginine)



1/400

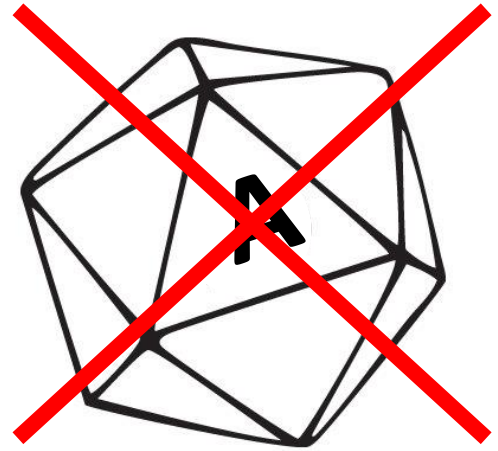
$$P(\text{'A'} \wedge \text{'R'}) = p_A \times p_R = 1/20 \times 1/20 = 1/400$$

Null model based on empirical data



Amino acid composition (%) in UniProt/TrEMBL (Feb 2021)

http://www.ebi.ac.uk/uniprot/TrEMBLstats



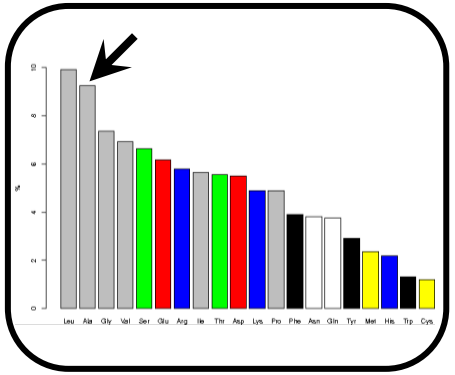
$p_S = 0.0672$

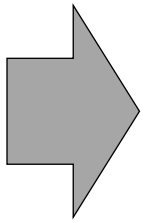
Amino acid composition (%)

Ala	(A)	9.23	Gln	(Q)	3.77	Leu	(L)	9.86	Ser	(S)	6.72
Arg	(R)	5.84	Glu	(E)	6.20	Lys	(K)	4.89	Thr	(T)	5.54
Asn	(N)	3.78	Gly	(G)	7.35	Met	(M)	2.35	Trp	(W)	1.30
Asp	(D)	5.48	His	(H)	2.20	Phe	(F)	3.88	Tyr	(Y)	2.87
Cys	(C)	1.24	Ile	(I)	5.53	Pro	(P)	4.94	Val	(V)	6.91

Probability in a little-more-realistic null model

Observing **A** in a sequence





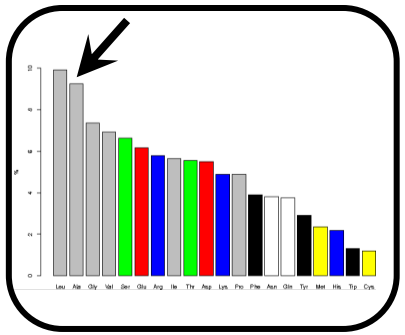
0.0923

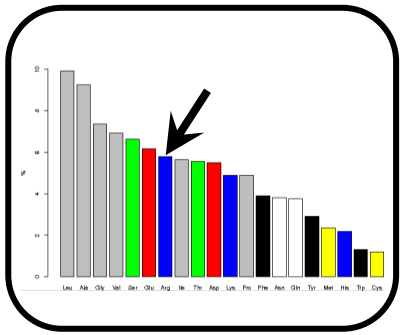
sequence **X** ...**A**...
sequence **Y** ...**R**...

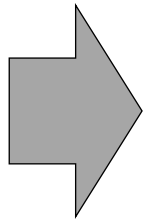
Aligned position:
X_i: position i in sequence X
Y_j: position j in sequence Y

$$P(\text{'A'}) = p_A = 0.0923$$

Observing **A** at X_i and **R** at Y_j simply by chance







0.0054

$$P(X_i=\text{'A'} \wedge Y_j=\text{'R'}) = p_A \times p_R$$
$$= 0.0923 \times 0.0584$$
$$= 0.0054$$

Log-odds score for amino acid substitution

p_a : **prior** probability of observing residue *a*
p_b : **prior** probability of observing residue *b*
q_{ab}: **joint** probability of observing residue *a* and residue *b* in the same column
γ : a scaling factor (e.g. $\gamma = 10$ if $\log_{10}$ scale is used)

Log-odds score
for ***a*** substituted by ***b***

$$S_{ab} = \gamma \log\left(\frac{q_{ab}}{p_a \cdot p_b}\right)$$

$S_{ab} > 0$ (**positive**): **more** frequent than expected ($q_{ab} > p_a \times p_b$)
$S_{ab} < 0$ (**negative**): **less** frequent than expected ($q_{ab} < p_a \times p_b$)

Consider the **A–R** substitution in these two sequences:

```
Seq1 YPSVPFSAGP
Seq2 YPVLPFSRGP
```

Example

$P_{observed}$ for **A→R** substitution = q_{AR}
P_{null} for **A→R** substitution = $p_A \times p_R$

Substitution matrix

captures the propensity of any amino acids to be substituted by another amino acid due to biological reasons

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2	-2	0	0	-2															
R	-2	6	0	-1	-4															
N	0	0	2		-4															
D	0	-1	2	4	-5															
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	-2	0	-2	-8	-5	-1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-4	-3	-3	0	7	-1
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Example

+2 indicates that the **N→N** replacement occurs 1.58 times more frequent than expected by chance

$$10 \cdot \log_{10}(x) = 2$$

$$\log_{10}(x) = 0.2$$

$$x = 10^{0.2} = 1.58$$

Log-odds scores

Why do we use log-odds score?

- Logarithms are easier to use for a scoring system
- They allow us to sum the scores of aligned residues (rather than multiplying the probabilities for independent mutations)
- The sum of log-odds is equivalent to product of probabilities

Example

```
seqA QRVYPSVPFSAGP
seqB LRKYPVLPFSAGP
```

$$S_{AB} = S_{\mathbf{QL}} + S_{\mathbf{RR}} + S_{\mathbf{VK}} + S_{\mathbf{YY}} + S_{\mathbf{PP}} + \ldots$$

Point Accepted Mutation (PAM) matrices

PAM matrices (Dayhoff, Schwartz & Orcutt, 1978)

- based on **1,572** observed mutations in **71 families** of **closely related** proteins

An **accepted point-mutation** is a single-residue mutation that was incorporated into the protein (and passed to its progeny), thus it:

(a) **did not disrupt** the protein function or

(b) was **beneficial** to the organism (e.g. in evolutionary terms, it increased the fitness of the species)

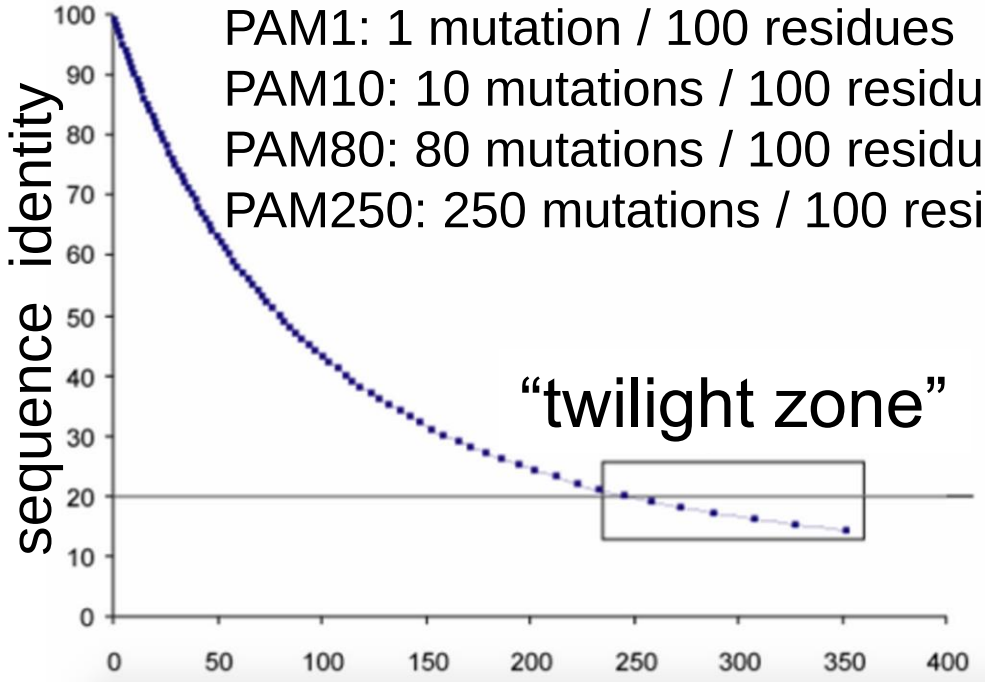
Margaret O. Dayhoff (1925–1982)

PAM units

1 PAM unit: a series of accepted point mutations (and no insertions of deletions) has converted S_1 to S_2 with an average of **one accepted point-mutation event per 100 amino acids**. It measures the **rate of divergence**, i.e. the **evolutionary distance**.

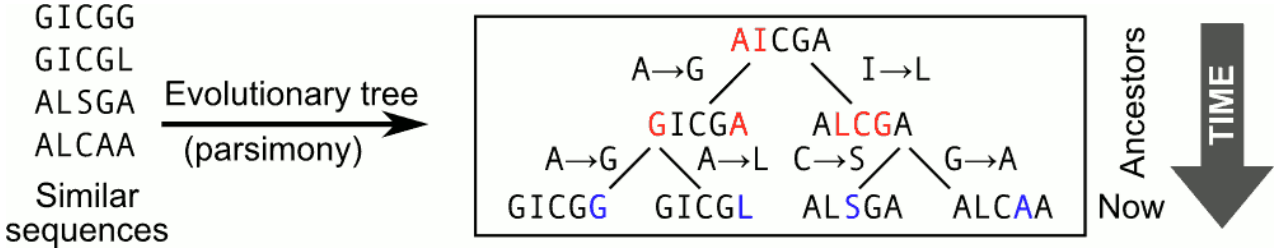
- **PAM unit** between two sequences is **not necessarily the same** as percent difference in sequence identity
- Single position may undergo > 1 mutation, which could also result in no change observed in the sequence, e.g. **Y → H → Y**

PAM0: 0 mutation / 100 residues
PAM1: 1 mutation / 100 residues
PAM10: 10 mutations / 100 residues
PAM80: 80 mutations / 100 residues
PAM250: 250 mutations / 100 residues



Construction of PAM*n* substitution matrices

1. for PAM1 matrix, protein sequences with >85% identity are used



2. count of amino acid replacements are recorded along branches of a phylogenetic tree

Count matrix

count matrix

	A	R	N	D	C
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

PAM1

3. transition probability for each pair of amino acids is calculated based on the count matrix and the occurrence of these amino acids in the dataset

Transition probability matrix

probability matrix

	A	R	N	D	C
A	13	6	9	9	5
R	3	17	4	3	2
N	4	4	6	7	2
D	5	4	8	11	1
C	2	1	1	1	52

PAM250

4. matrices of other PAM units are extrapolated from PAM1 *via* matrix multiplication: **PAM2** = $(PAM1)^2$; **PAM250** = $(PAM1)^{250}$

5. probabilities can be transformed into log-odds scores

Scoring matrix

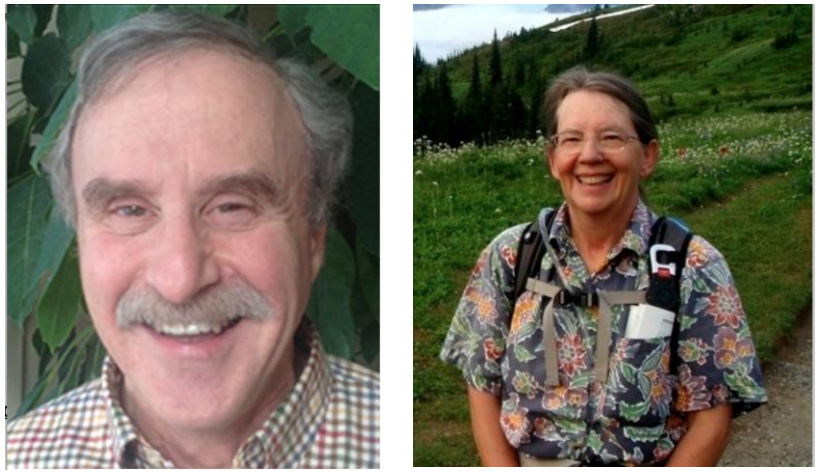
scoring matrix

A	2				
R	-2	6			
N	0	0	2		
D	0	-1	2	4	
C	-2	-4	-4	-5	12

PAM250

BLOck SUbstitution Matrix (BLOSUM)

- Based on **clustering** of **distantly related** proteins
- **Blocks** database consists of >2000 **locally aligned** (*blocks*) of conserved regions from >500 groups of distantly related proteins
- Observed amino acid frequencies derived based on aligned blocks (no phylogenetic trees)



Steven & Jorja Henikoff

- In **BLOSUM*n* matrices**, sequences with identity **> *n*%** are clustered
- Scores are derived from inter-cluster differences (among sequences sharing **< *n*% identity**).

BLOSUM: clustering based on %identity

Example

AKLGGREAVE
AKLIGREAVE
DKIGGHPAIE
DNIGGQPAIE
DKIGGQPAIE
EKLGGTTAVD
EKLGGTTAMK
EKLGGTAAVQ
EKLGGQAAVQ
YEAIGEELLS

Cluster at % identity ≥ **80**

AKLGGREAVE
AKLIGREAVE

DKIGGHPAIE
DNIGGQPAIE
DKIGGQPAIE

Identity for each possible pairwise sequences within each cluster ≥ **80%**

EKLGGTTAVD
EKLGGTTAMK
EKLGGTAAVQ
EKLGGQAAVQ

These four clusters form the basis for BLOSUM**80** matrix

YEAIGEELLS

BLOSUM: deriving transition probability

Calculate q_{QN} for BLOSUM**50**

Example

- **3** possible pairs of clusters (**1–2**, **1–3** and **2–3**)
- **5 amino acid residues** in length

A: total number of aligned pairs **= 3 × 5 = 15**

B: total **Q**⟷**N** substitution frequency **between** each cluster-pair

1

ATCK**Q**
ATCR**N**
ASCK**N**
SSCR**N**

between **1–2**

¼ is **Q** in **1** compares to ½ is **N** in **2**
¾ is **N** in **1** compares to ½ is **Q** in **2**

$$\left(\frac{1}{4}\times\frac{1}{2}\right)+\left(\frac{3}{4}\times\frac{1}{2}\right)=\frac{4}{8}=\mathbf{\frac{1}{2}}$$

2

SDCE**Q**
SECE**N**

between **2–3**

½ is **Q** in **2** compares to **0** is **N** in **3**
½ is **N** in **2** compares to **1/1** is **Q** in **3**

$$\left(\frac{1}{2}\times 0\right)+\left(\frac{1}{2}\times\frac{1}{1}\right)=\mathbf{\frac{1}{2}}$$

3

TECR**Q**

between **1–3**

¼ is **Q** in **1** compares to **0** is **N** in **3**
¾ is **N** in **1** compares to **1/1** is **Q** in **3**

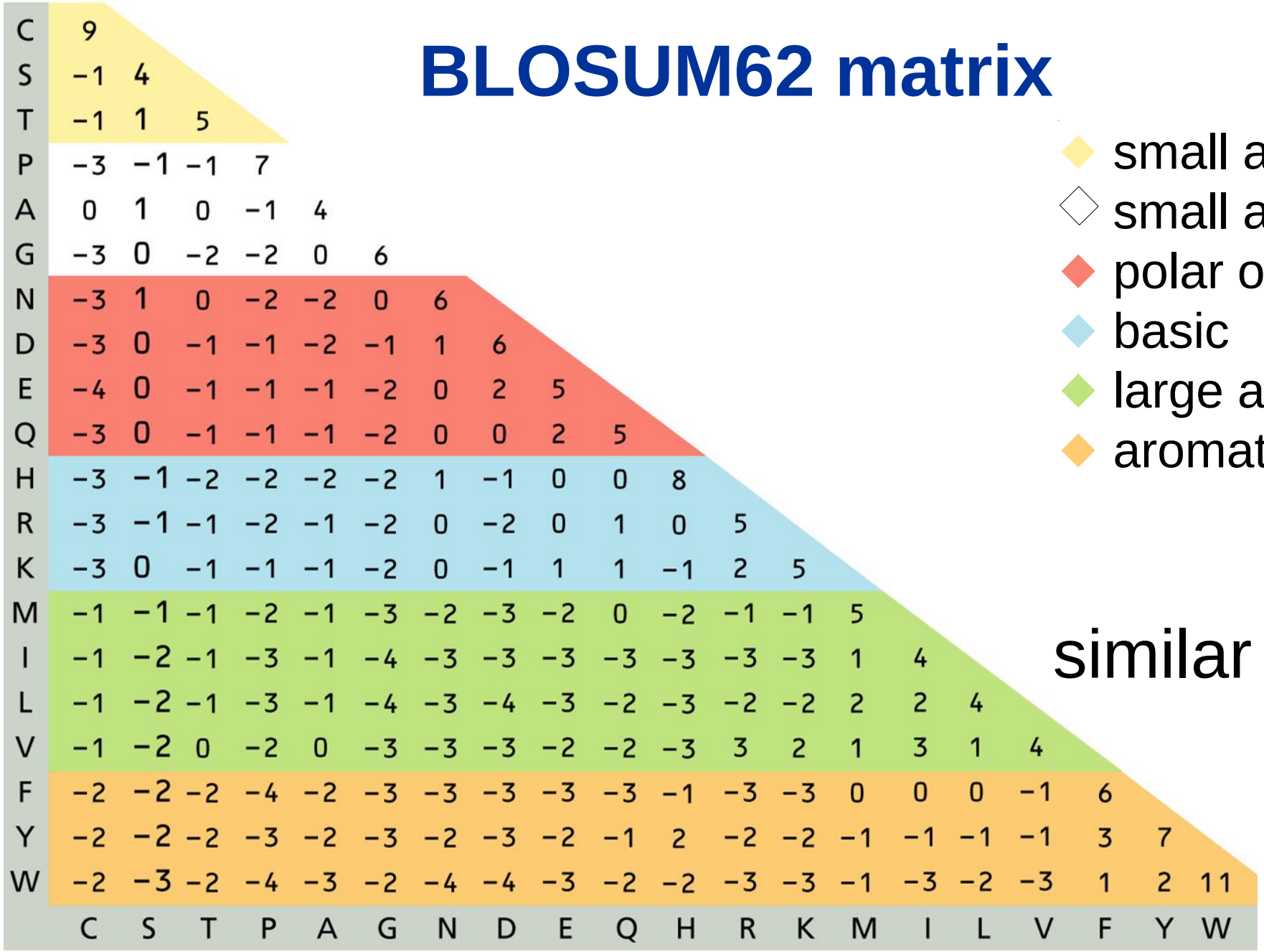
$$\left(\frac{1}{4}\times 0\right)+\left(\frac{3}{4}\times\frac{1}{1}\right)=\mathbf{\frac{3}{4}}$$

clustered at %identity ≥ 50

$$\boldsymbol{B}\ \text{Total}=\mathbf{\frac{1}{2}}+\mathbf{\frac{1}{2}}+\mathbf{\frac{3}{4}}=\frac{4+4+6}{8}=\mathbf{\frac{14}{8}}$$

$$q_{QN}=\frac{B}{A}=\frac{14}{8}\div 15=\mathbf{0.1167}$$

BLOSUM62 matrix



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	4																		
T	−1	1	5																	
P	−3	−1	−1	7																
A	0	1	0	−1	4															
G	−3	0	−2	−2	0	6														
N	−3	1	0	−2	−2	0	6													
D	−3	0	−1	−1	−2	−1	1	6												
E	−4	0	−1	−1	−1	−2	0	2	5											
Q	−3	0	−1	−1	−1	−2	0	0	2	5										
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8									
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5								
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5							
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5						
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4					
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4				
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	3	2	1	3	1	4			
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6		
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11

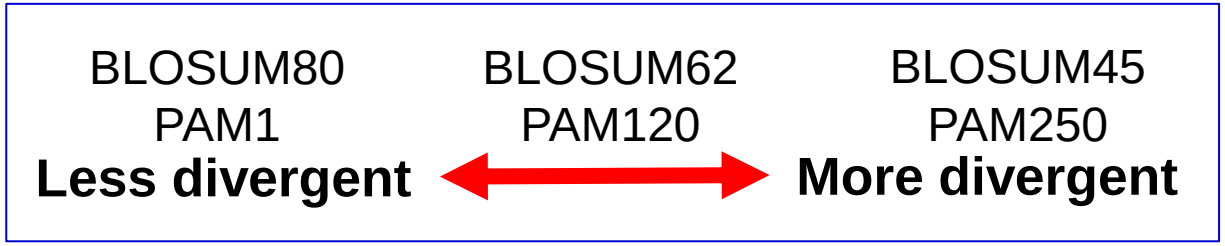
- small and polar
- small and non-polar
- polar or acidic
- basic
- large and hydrophobic
- aromatic

similar to PAM120

Figure 4.4 Textbook (pg. 83)

PAM matrices

- Uses **closely related** proteins
- Based on an **explicit** evolutionary model (i.e. replacements counted on the branches of a phylogenetic tree)
- mutations observed throughout a **global** alignment
- All mutations are **counted the same**
- **Higher** PAM units denote **larger evolutionary distance**



BLOSUM matrices

- Uses **evolutionarily divergent** proteins
- Based on an **implicit** model of evolution (no trees)
- Based only on highly conserved regions in series of **local** alignments without gaps
- Uses groups of sequences within which **not all mutations are counted the same**
- **Larger** numbers in the BLOSUM matrix naming scheme denote higher sequence similarity (and thus **smaller evolutionary distance**)

These matrices could be too simplistic. Further developments result in more-realistic substitution models based on more-extensive protein data, e.g. **JTT** and **WAG**.

Reflection

What is homology, and how does it relate to shared sequence similarity?

Biologically, why would DNA/proteins sequences be conserved, or different from one another?

How do we quantify and model sequence change?

Why do we prefer log-odds score to probability values?

How can we model substitutions based on empirical data?

What are PAM and BLOSUM matrices, and how do they differ?

Sequence Analysis 1

B. Sequence alignment

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Lecture outline

- **Sequence alignment**
  - Modelling insertions and deletions using gap
  - linear *versus* affine gap penalty
- **Dynamic programming in sequence alignment**
  - Needleman-Wunsch algorithm
  - Smith-Waterman algorithm
  - Global versus local alignment
- **Dynamic programming and computational complexity**

Sequence alignment

- The fundamental way of comparing sequences
- Aim: to **quantify similarity** among a set of (two or more) sequences, which informs shared **homology**
- locates **equivalent regions** of two or more sequences to reveal (maximise) the extent of their **similarity**

Applications:

- inference of **phylogenetic** (i.e. evolutionary) relationships (dissimilarity as a measure of evolutionary distance)
- **prediction** of functions, structures and sequence features (e.g. binding sites, splicing signals) in novel sequences based on homology evidence to known sequences

Alignment is a position-by-position hypothesis of homology

Is a substitution matrix sufficient?

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9																			
S	−1	4																		
T	−1	1	5																	
P	−3	−1	−1	7																
A	0	1	0	−1	4															
G	−3	0	−2	−2	0	6														
N	−3	1	0	−2	−2	0	6													
D	−3	0	−1	−1	−2	−1	1	6												
E	−4	0	−1	−1	−1	−2	0	2	5											
Q	−3	0	−1	−1	−1	−2	0	0	2	5										
H	−3	−1	−2	−2	−2	−2	1	−1	0	0	8									
R	−3	−1	−1	−2	−1	−2	0	−2	0	1	0	5								
K	−3	0	−1	−1	−1	−2	0	−1	1	1	−1	2	5							
M	−1	−1	−1	−2	−1	−3	−2	−3	−2	0	−2	−1	−1	5						
I	−1	−2	−1	−3	−1	−4	−3	−3	−3	−3	−3	−3	−3	1	4					
L	−1	−2	−1	−3	−1	−4	−3	−4	−3	−2	−3	−2	−2	2	2	4				
V	−1	−2	0	−2	0	−3	−3	−3	−2	−2	−3	3	2	1	3	1	4			
F	−2	−2	−2	−4	−2	−3	−3	−3	−3	−3	−1	−3	−3	0	0	0	−1	6		
Y	−2	−2	−2	−3	−2	−3	−2	−3	−2	−1	2	−2	−2	−1	−1	−1	−1	3	7	
W	−2	−3	−2	−4	−3	−2	−4	−4	−3	−2	−2	−3	−3	−1	−3	−2	−3	1	2	11

```
seqA QRVYPSVPFSAGP
seqB LRKYPVLPFSAGP
```

Example

$$S_{AB} = S_{\mathbf{QL}} + S_{\mathbf{RR}} + S_{\mathbf{VK}} + S_{\mathbf{YY}} + S_{\mathbf{PP}} + \ldots$$

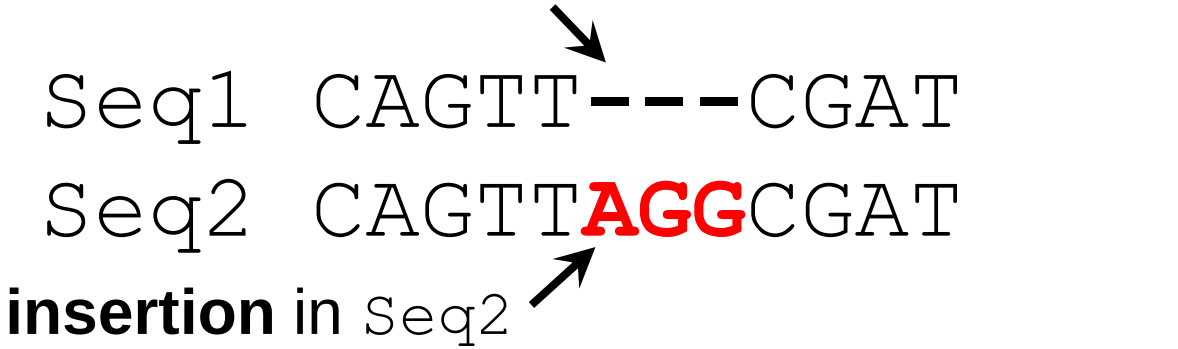
Sum of log-odds scores is used to assign a score to an alignment

Are we missing anything?

Insertions and deletions

- evolutionary events that result in the introduction of new bases into (**insertion**), or the removal of existing bases from (**deletion**), the genomic DNA
- **Indels:** mutation that includes both insertions, deletions and the combination thereof
- modelled as **gaps** in sequence alignment

Example



Modelling indels using gap penalty

Gaps in an alignment ↑ **Similarity** between two sequences ↓

Gaps are modelled as a **penalty**, i.e. assigned as a negative score

Linear gap penalty for a gap of length n, $g(n)$ is

$$g(n) = -n \times E$$ where E is a cost for a single gap

$$g(n) \propto n$$

Example

```
Seq1  CAGTT---CGAT
Seq2  CAGTTAGGCGAT
```

If $E = 1$, then $g(n) = -3$

Every gap position costs the same

Modelling indels using gap penalty

How many indel event(s) could have occurred in Seq2?

a. An insertion of **A**, an insertion of **G** and an insertion of **G** (3 events)
b. An insertion of **AG**, and an insertion of **G** (2 events)
c. An insertion of **AGG** (1 event), etc.

Example

```
Seq1  CAGTT---CGAT
Seq2  CAGTTAGGCGAT
```

Is it fair to treat every single gap position the same?

Gap open: the cost of opening/starting a gap

Gap extend: the cost of extending a gap by one position

Affine gap penalty for a gap of length n, $g(n)$ is

$$g(n) = -O - (n-1) \times E$$

where O is the gap open cost, E is the gap extend cost

If $O = 3$, $E = 1$, then $g(n) = -5$

Many small gaps cost more than one large gap

Calculating score for an alignment with gap

Alignment score $S = s + g$

s = total substitution score

g = total gap penalty (in negative)

Examples	Linear gap scheme $g(n) = -n \times E$, $E = 2$	Affine gap scheme $g(n) = -O - (n-1) \times E$, $O = 3$, $E = 1$
g(3) SVDNA---RHV SISQSAQLSHV 43001 84 −1	$S = 19 + (-6) = 13$	$S = 19 + (-5) = 14$
g(1) g(2) SVDN-A--RHV SISQSAQLSHV 4300 4 84 −1	$S = 22 + (-6) = 16$	$S = 22 + (-7) = 15$

Impact of gap penalty on an alignment

Example

Alignment A

```
seqA  LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
seqB  --WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHMETGNHYAMKILDKQKVVKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDNSNLY

seqA  QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKDDGQLFHIDFGHFLDHKKKFGYKRERVPFVLTQDF
seqB  MVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPEYLAP

seqA  LIVISKGAQECTKTREFERFQEMCYKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMKQMNDAHHGG
seqB  EIILSKGYNKAVDWWALGVLIYEMAAGYPPFFADQPIQIYEKIVSGKVRFPSHFSSDLKDLLRNLLQVDLTKRFGNLKNGVNDIKNHKWF

seqA  WTTKMDWIFHTIKQHALN----------------------------------
seqB  ATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

Alignment B

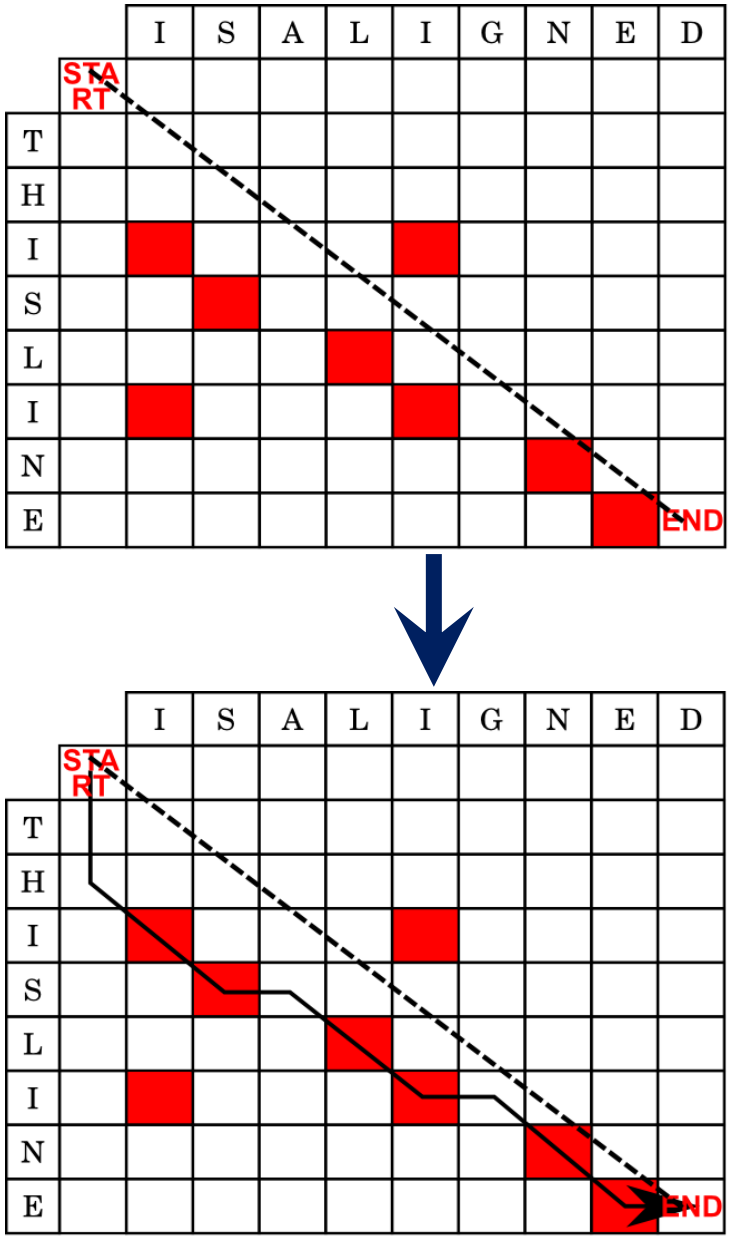
```
seqA  LNWENPDIMSELLFQNNEIIFKNGDDLRQDMLTLQIIRIMENIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQIQCKGGLKGAL
seqB  ?-WENPAQNTAHLDQFERIKTLGTGSFGRVMLVKHM--ETGNHYAMKILDKQKV-VKLKQIEHTLNEKRILQAVNFPFLVKLEFSFKDN-

seqA  QFNSHTLHQWLKDKNKGEIYDAAIDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-DGQLFHIDFGHFLDHKKKKFGYKRERVPFVL--T
seqB  -SNLYMVMEYVPGGEMFSHLRR-IGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGT

seqA  QDFL---IVISKGAQECTKTREFERF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEALEYFMK
seqB  PEYLAPEIILSKGYNKAVDWWALGVLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVRF--PSHFSSDLKDLLRNLLQVDLTKR--FGNLKN

seqA  QMNDAHHGGWTTKMDWI----------------------FHTIKQHAL----N----------
seqB  GVNDIKNHKWFATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXINEKCGKEFSEF
```

Which alignment is a result of having a higher (more costly) gap penalties?

Matrix representation of an alignment

Dynamic programming aims to find the optimal (best) path

More diagonal moves = more identical positions aligned



Example

T H I S L I N E

I S A L I G N E D

Gap in first line

Aligned positions

Gap in second line

T H I S – L I – N E –

– – I S A L I G N E D

Gap in first line

Aligned positions

Gap in second line

Dynamic programming in sequence alignment

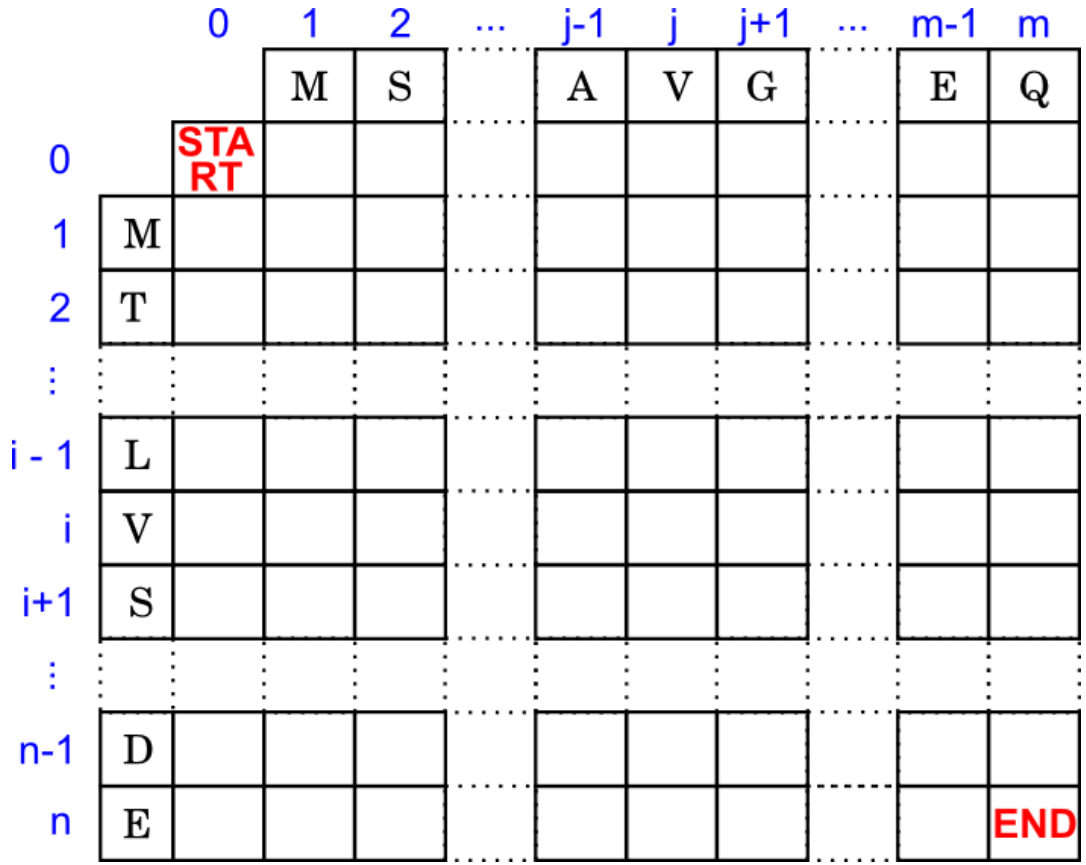
- reduces a big, hard, **optimisation problem into smaller problems** whose results can be combined: optimisation of alignment = sum of sub-alignment optimisations

$$S = S_1 + S_2 + S_3$$

- optimisation problem in time proportional to the product of two sequence lengths, m and n: $O(n \cdot m)$
- transforms a sequence into another using edit operations that **replace**, **insert**, or **remove** an element
- each operation has an associated **cost**; the goal is to find the sequence of edits with the **lowest** total cost

Dynamic programming in sequence alignment

Consider seq1 of length n, seq2 of length m. Let $S_{i,j}$ be the score for the best alignment ending at position i in seq1 and position j in seq2.



Four steps:

1. Initialisation

$S(0,0) = 0$

$S(0, j)$ and $S(i, 0)$ based on gaps

2. Recurrence

With additive costs, we can compute $S_{i,j}$ recursively from $S_{i-1,j-1}$, $S_{i-1,j}$ and $S_{i,j-1}$

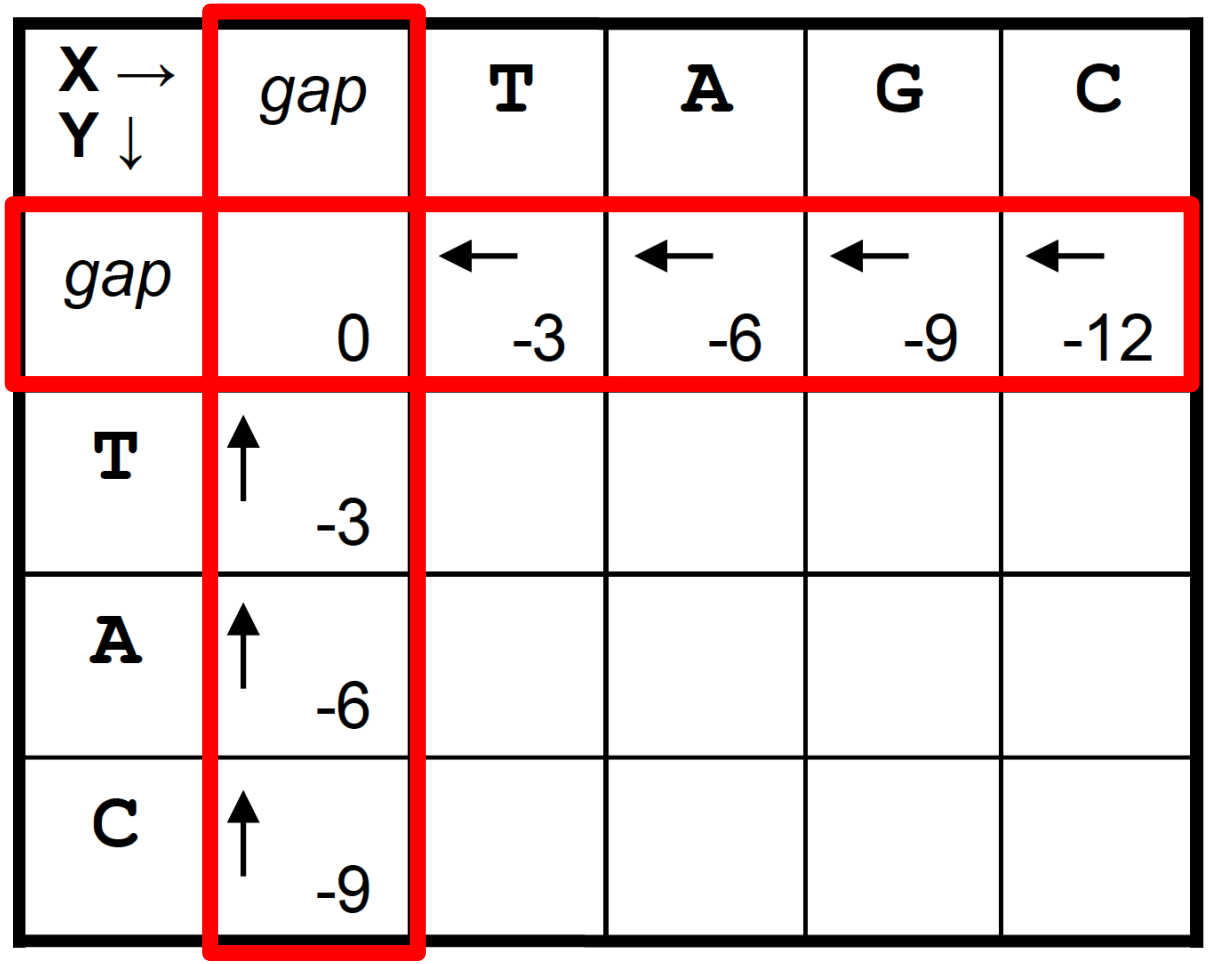
3. Termination

4. Trace backward

Needleman-Wunsch algorithm

- First description of a similarity-searching method among biological sequences (Needleman & Wunsh, 1970)
- Aims to produce an alignment by inexact string matching i.e. an alignment that incorporates **matches**, **mismatches** and **gaps** placed as required, in order to give the best possible alignment (the **optimal** alignment).
- **All** positions in a sequence are compared against **all** positions in another (i.e. in a **global** alignment)
- longest line/path is found by tracing back through the matrix (i.e. the similarity procedure)
- adopted in all commonly used alignment and similarity search programs

NW algorithm: initiation

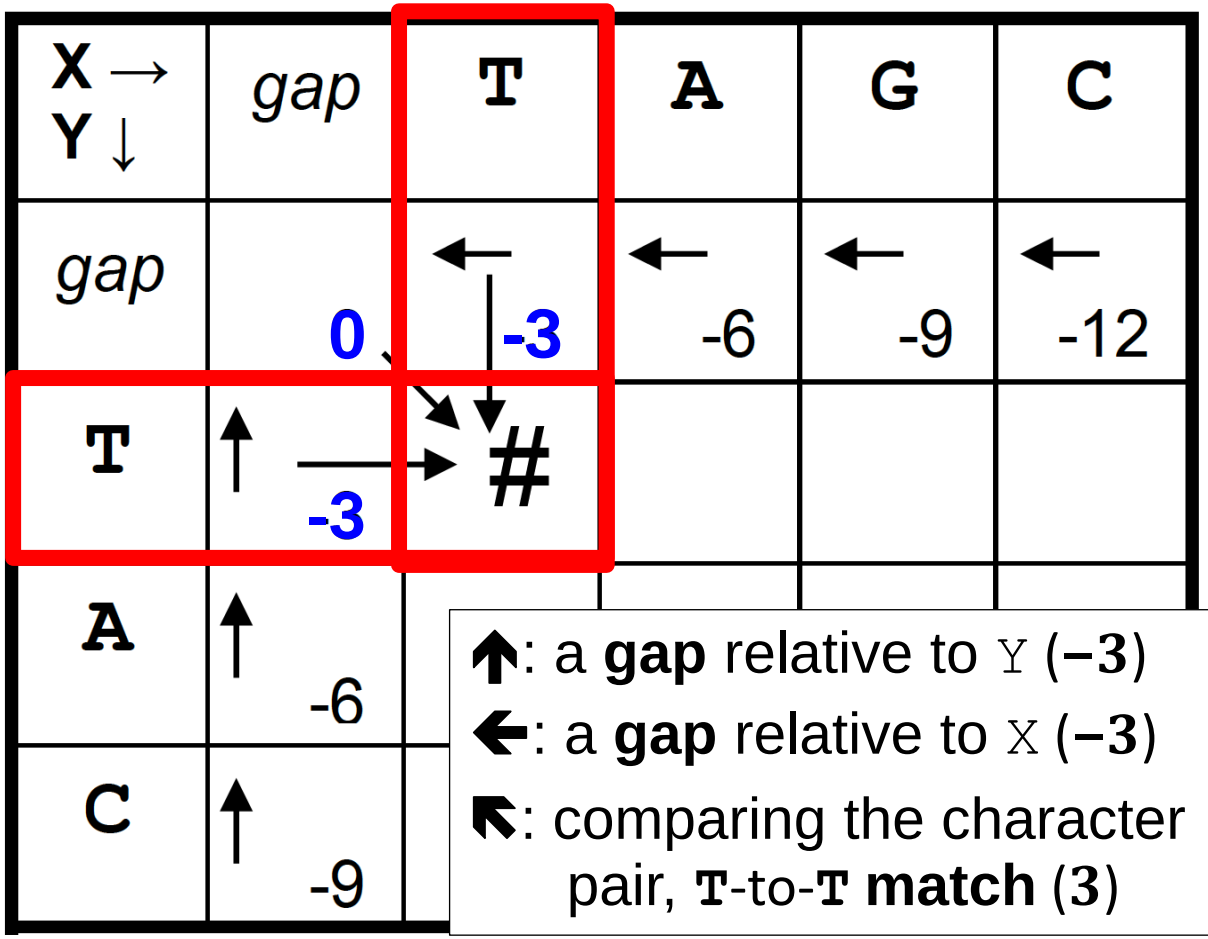


SeqX: TAGC

Scoring scheme:
- match (**+3**)
- mismatch (**+ 0**)
- gap (**–3**)

- **initialise their back-pointers** (arrows indicate where the accumulative score was obtained)
- scores are accumulative from left to right, from top to bottom

NW algorithm: recurrence



SeqX: TAGC

SeqY: TAC

↑: a **gap** relative to Y (**−3**)

←: a **gap** relative to X (**−3**)

↖: comparing the character pair, **T**-to-**T** **match** (**3**)

Scoring scheme:

- match (**+3**)
- mismatch (**+ 0**)
- gap (**−3**)

Sum each score **+** score in the box in the corresponding direction

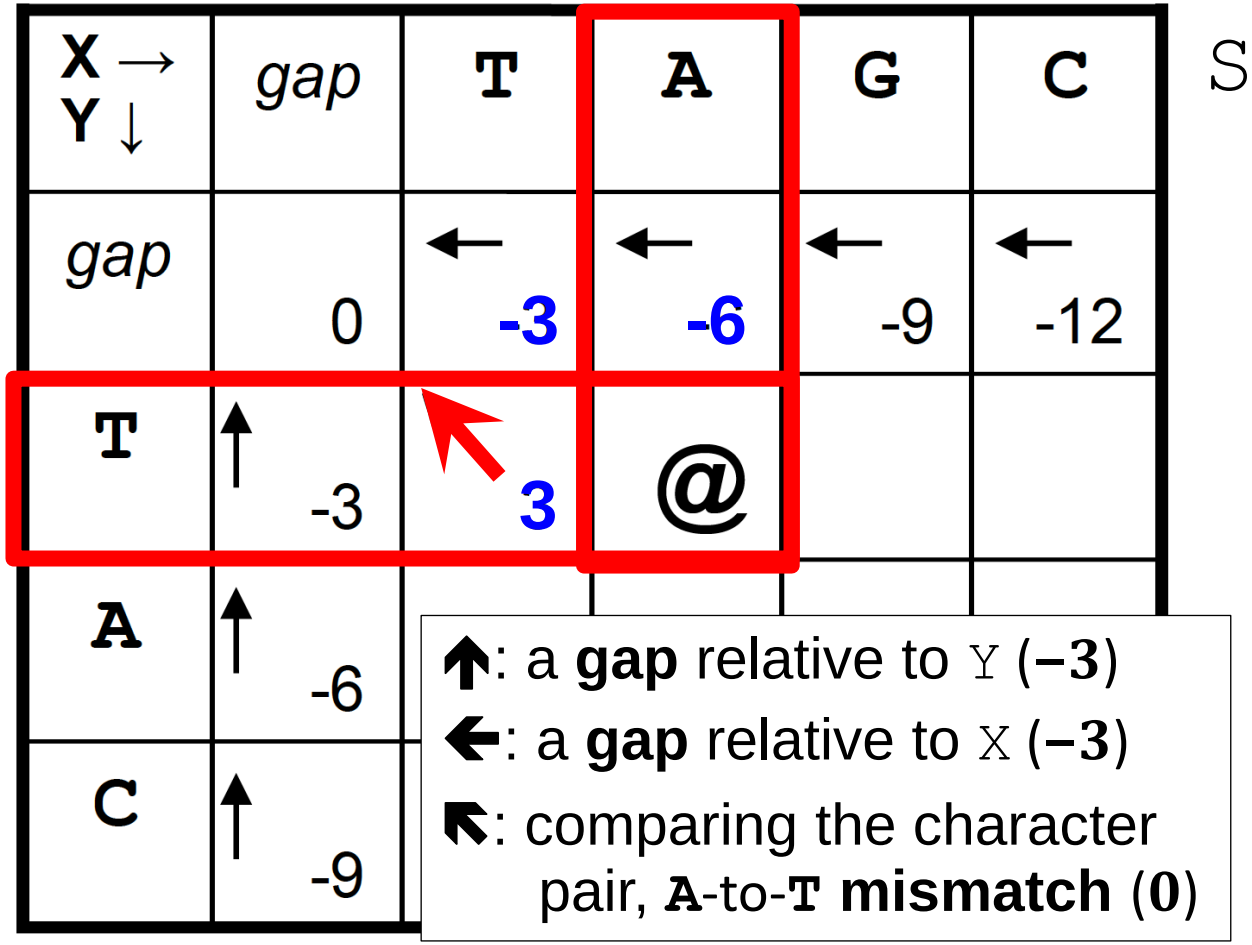
↑:(−3)+(−3) = −6

←:(−3)+(−3) = −6

↖:(3)+(0) = 3

maximum score: **3** at direction ↖

NW algorithm: recurrence



SeqX: TAGC

SeqY: TAC

↑: a **gap** relative to Y (**−3**)

←: a **gap** relative to X (**−3**)

↖: comparing the character pair, **A**-to-**T** **mismatch** (**0**)

Scoring scheme:

- match (**+3**)
- mismatch (**+ 0**)
- gap (**−3**)

Sum each score **+** score in the box in the corresponding direction

↑:**(−3)+(−6) = −9**

←:**(−3)+(3) = 0**

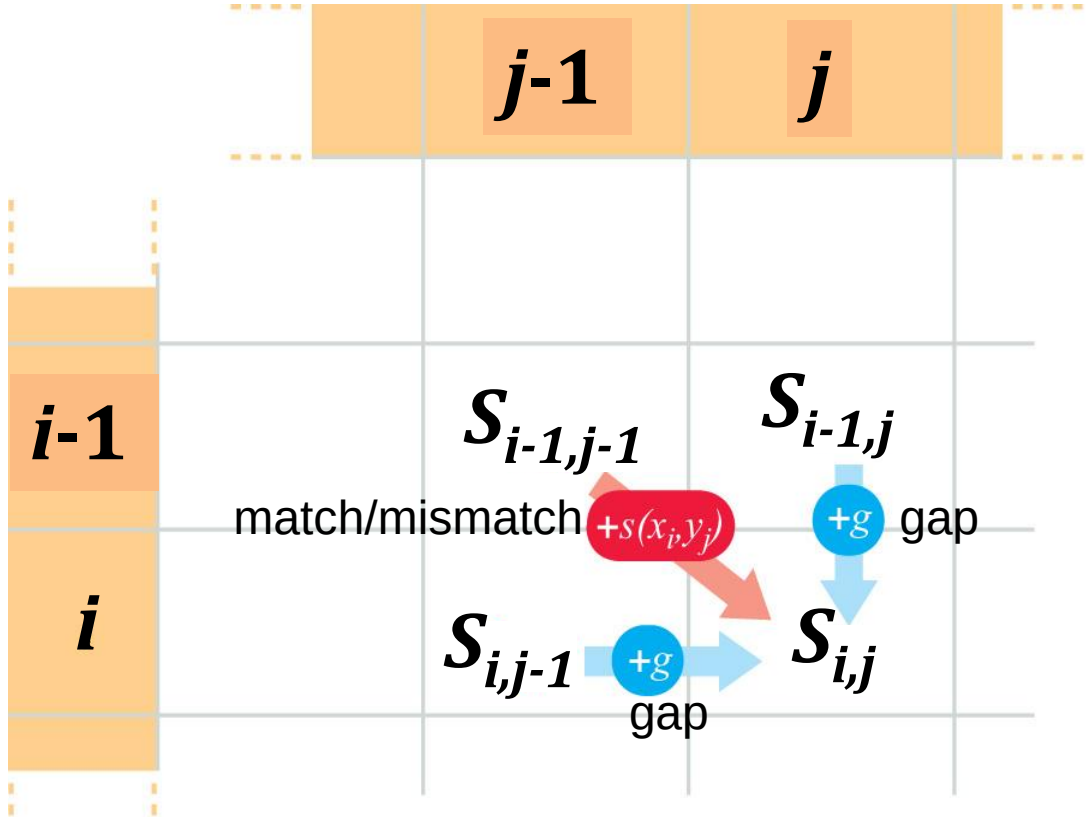
↖:**(0)+(−3) = −3**

maximum score: **0** at direction ←

and so forth, until all boxes are filled

NW algorithm: recurrence

$S_{i,j}$ is the score for the **best** alignment of the initial segments of sequence x and sequence y ending at position i and j, respectively



$$S_{i,j} = \max \begin{pmatrix} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{pmatrix}$$

NW algorithm: termination and tracing back

X → / Y ↓	*gap*	**T**	**A**	**G**	**C**
gap	0	← -3	← -6	← -9	← -12
T	↑ -3	↖ 3	← 0	← -3	← -6
A	↑ -6	↑ 0	↖ 6	← 3	← 0
C	↑ -9	↑ -3	↑ 3	↖ 6	↖ 6

SeqY: TAC

SeqX: TAGC

- Terminates at the **bottom-right** box (the end of both sequences)
- Traces from the **bottom-right** cell backward towards the **top-left** cell

```
seqX  TAGC
      : : :
seqY  TA-C
```

the **optimal** alignment
total score = **6**

See Fig. 5.11 (p131) on textbook for another example.

Smith-Waterman algorithm

modified from the NW algorithm to find **locally** matched regions between two sequences (i.e. a **local** alignment algorithm)

seqX TAGC

X → / Y ↓	gap	T	A	G	C
gap	0	← 0	← 0	← 0	← 0
A	↑ 0	↖ 0	↖ 3	← 0	↖ 0
G	↑ 0	↖ 0	↑ 0	↖ 6	← 3
T	↑ 0	↖ 3	← 0	↑ 3	↖ 5

seqY AGT

the **optimal** alignment
total score **= 6**

```
seqX AG
     ::
seqY AG
```

- **mismatches** are assigned negative scores
- negative score is set to **zero** (i.e. the minimum is non-negative)

$$S_{i,j} = \max \begin{pmatrix} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \\ 0 \end{pmatrix}$$

- traces back from the box with the **highest** score, through the alignment until a **zero** is reached

Global versus local alignment

- attempts to align **every residue in every sequence**, i.e including both highly conserved and highly variable regions
- most useful when the sequences are **similar** and of **roughly equal size**
- **less** prone to demonstrating **false homology**

- focuses only on **conserved** regions in the sequences
- useful for **dissimilar sequences** that are expected to contain **conserved** regions (e.g. protein domains) or similar sequence **motifs** (e.g. binding or active sites) within their **larger sequence context**
- **more** prone to demonstrating **false homology**

Global versus local alignment

	Needleman-Wunsch (global)	**Smith-Waterman** (local)
Initiation	$S(0,0) = 0$ $S(i,0) = g(i)$ $S(0,j) = g(j)$	$S(0,0) = 0$ $S(i,0) = 0$ $S(0,j) = 0$
Recurrence	$\boldsymbol{S_{i,j}} = \max\begin{pmatrix} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{pmatrix}$	$\boldsymbol{S_{i,j}} = \max\begin{pmatrix} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \\ 0 \end{pmatrix}$
Trace back from	Bottom-right cell	Highest scoring cell
Trace back until	Top-left cell	Cell with zero score

Dynamic programming and computational complexity

- DP determines optimal alignments by resolving optimal partial alignments; it guarantees the best alignment (solution), largely feasible for **pairwise sequence alignment**
- For ***n* individual sequences**, it requires constructing *n*-dimensional equivalent of the matrix formed in a standard pairwise alignment
- search space increases **exponentially** with ***n*** and strongly dependent on **sequence length**
- feasibility becomes an issue when aligning **three or more** sequences (i.e. **multiple sequence alignment**), prompting for heuristics (Week 4)

Reflection

- *Why would we want to align two DNA sequences (or two protein sequences)?*
- *What are the two common approaches to model insertions and deletions in sequence alignment?*
- *How does dynamic programming help in finding the best pairwise alignment?*
- *What are the key steps in dynamic programming?*
- *What are the key differences between a global alignment and a local alignment?*
- *Is dynamic programming feasible for aligning many, many sequences?*

Sequence Analysis 2

A. Multiple sequence alignment

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

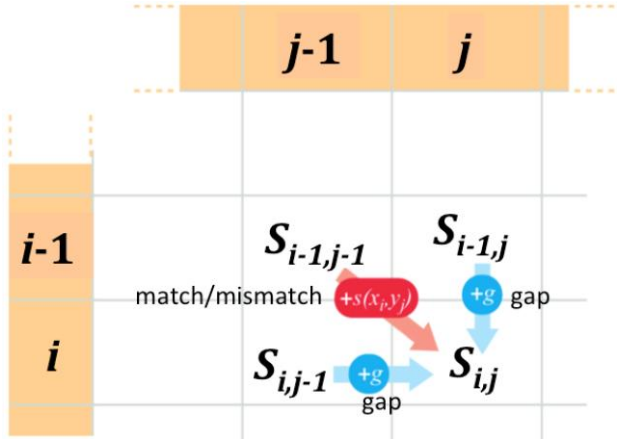
Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Lecture outline

- **Dynamic programming with affine gap penalty**
- **Multiple sequence alignment (MSA)**
  - Progressive MSA
    - Step-by-step example using Clustal, including tree-guided clustering using UPGMA and Neighbour-joining
    - Limitations
  - Iterative progressive MSA
  - Other MSA approaches
  - Measuring significance of an alignment
  - Sequence alignment versus structural alignment
- **MSA: issues and challenges**

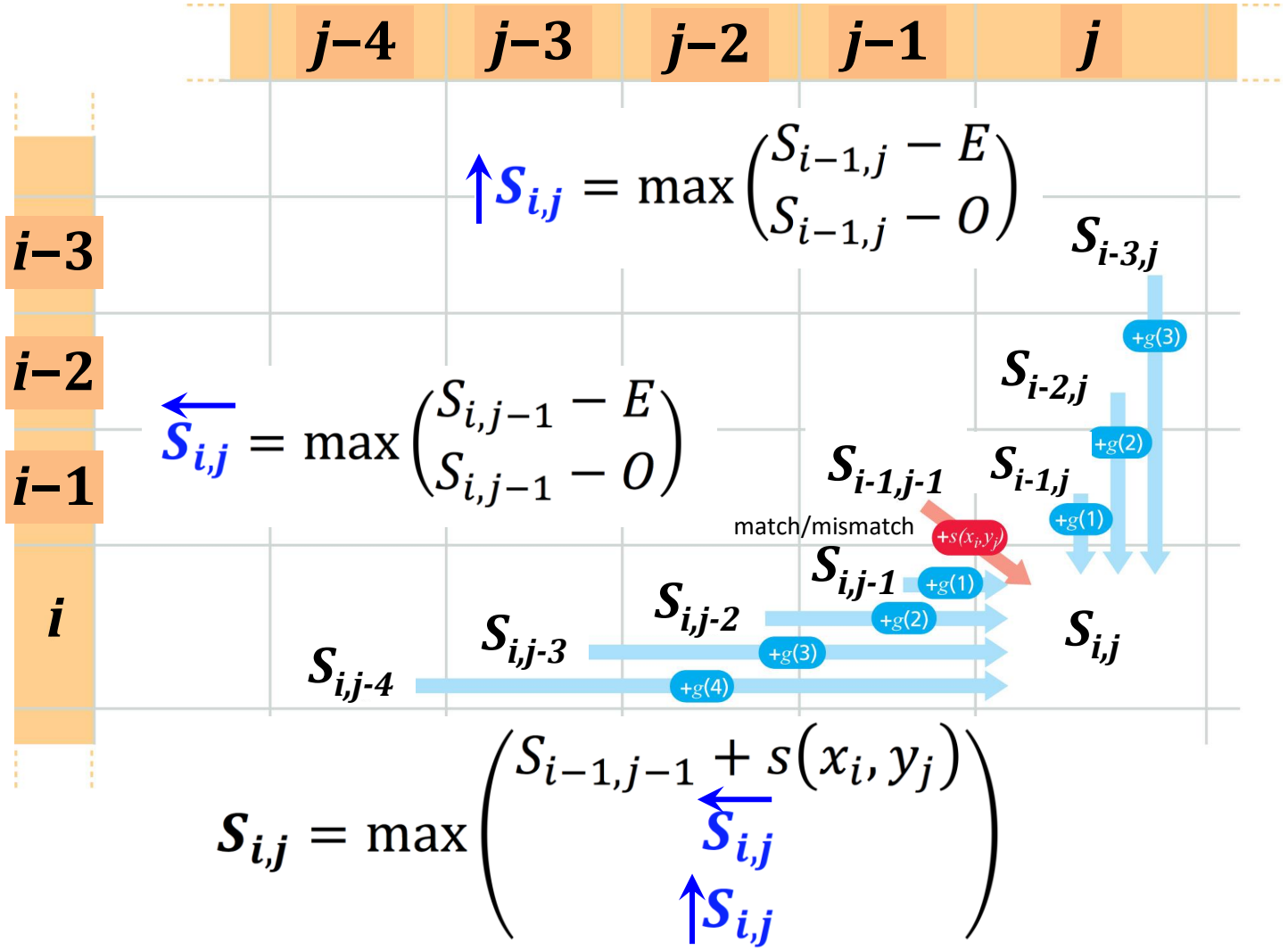
Affine gap penalty in dynamic programming



$$S_{i,j} = \max \begin{pmatrix} S_{i-1,j-1} + s(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{pmatrix}$$

Needleman-Wunsch algorithm (*Week 3*)

In that example, all gap position is treated the same, i.e. using ***linear gap penalty***



Affine gap penalty (distinction between *gap open*, ***O*** and *gap extend*, ***E***) can be applied in a more-realistic scheme

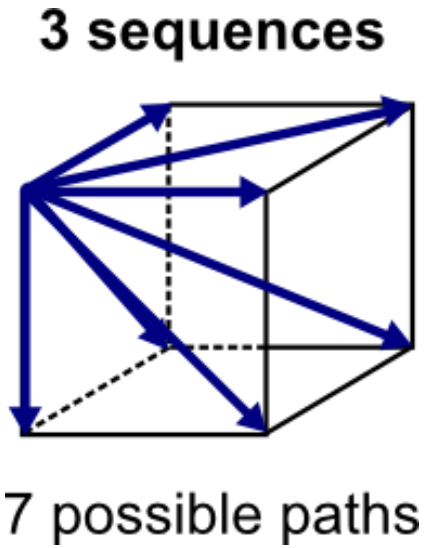
Dimensionality of scoring matrix in DP

- One dimension for each sequence in the alignment
- time and space grows **exponentially** with the number of sequences

Example: global alignment of three sequences, $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$

$\boldsymbol{S_{i,j,k}}$ is the score for the **best** alignment of the initial segments of sequences $\boldsymbol{x, y, z}$ ending at position $\boldsymbol{i, j, k}$, respectively

$$\boldsymbol{S_{i,j,k}} = \max \begin{pmatrix} S_{i-1,j-1,k-1} + s(x_i, y_j) + s(x_i, z_k) + s(y_j, z_k) \\ S_{i-1,j,k} + g + g + g \\ S_{i,j-1,k} + g + g + g \\ S_{i,j,k-1} + g + g + g \\ S_{i-1,j-1,k} + s(x_i, y_j) + g + g \\ S_{i-1,j,k-1} + s(x_i, z_k) + g + g \\ S_{i,j-1,k-1} + s(y_j, z_k) + g + g \end{pmatrix}$$



Multiple sequence alignment (MSA)

"The purpose of an MSA algorithm is to assemble alignments reflecting the **biological relationship** between several sequences. Computing exact MSAs is computationally almost impossible, and in practice **approximate algorithms (heuristics)** are used to align sequences, by maximizing their similarity."

Cédric Notredame (2007) *PLoS Computational Biology* 3(8): e123.

Multiple sequence alignment (MSA)

- alignment of three or more biological sequences
- a key step for inferring phylogenetic (evolutionary) relationships among a set of sequences
- **greater** information content (at each position) than pairwise alignment can illustrate sequence **constraints** and **integrity**, e.g. common signatures or protein domains, genetic variation etc.

Pairwise alignment

p110α	TFILGIGDRHNSNIMVKDDG-QLFHIDFGHFLDHKKKKFGYKRERVPFVLT--QDFLIVI	142
cAMP-kinase	QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCGTPEYLAPE	179

Example

Multiple sequence alignment

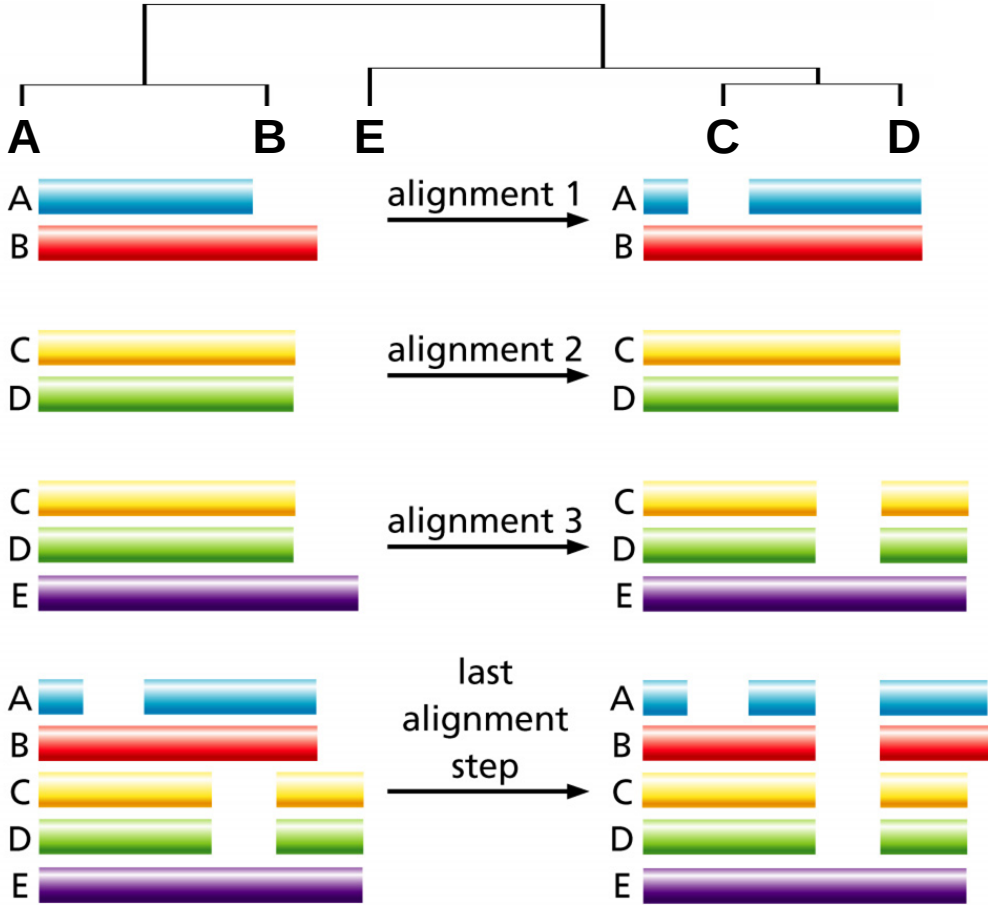
p110β	SYVLGIG----------DRHSDNINVKKTGQLFHIDFGHILGNFKSKFGIKRERVPFILT	136
p110δ	TYVLGIG----------DRHSDNIMIRESGQLFHIDFGHFLGNFKTKFGINRERVPFILT	136
p110α	TFILGIG----------DRHNSNIMVKDDGQLFHIDFGHFLDHKKKKFGYKRERVPFVLT	135
p110γ	TFVLGIG----------DRHNDNIMITETGNLFHIDFGHILGNYKSFLGINKERVPFVLT	135
p110_dicti	TYVLGIG----------DRHNDNLMVTKGGRLFHIDFGHFLGNYKKKFGFKRERAPFVFT	135
cAMP-kinase	QIVLTFEYLHSLDLIYRDLKPENLLIDQQGYIQVTDFGFAKRVKGRTWXLCG--TPEYLA	177

Progressive multiple sequence alignment

- the most widely used **heuristic** technique in MSA
- **Heuristics**: a practical method **not guaranteed** to be optimal or perfect, but sufficient for the immediate goals

Generally a three-step process:

1. Assess **pairwise sequence similarity**, e.g. build a similarity matrix
2. Build a **guide tree** based on pairwise similarity and define an order of addition of sequences to alignments (from the **most similar** sequence-pair to the **most dissimilar** pair)
3. Align sequences **progressively** based on the defined order



Align these five sequences …

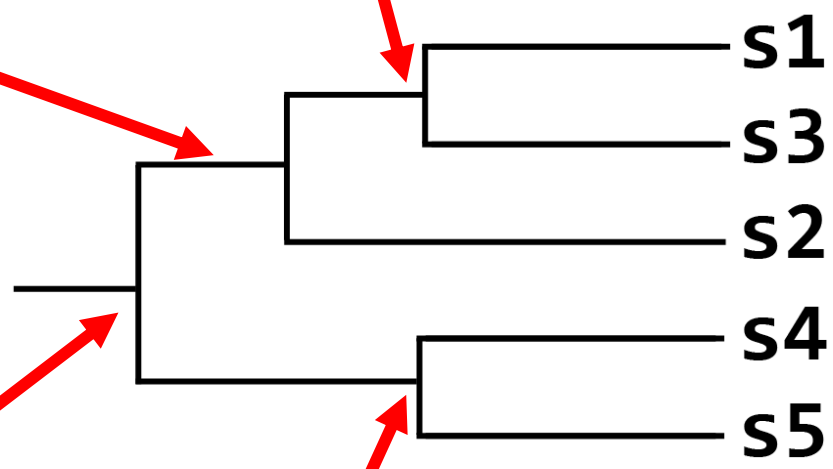
```
s1: ACCGTGAAGCCAATAC
s2: ACGTGCAACCATTAC
s3: AGCGTGCAGCCAATAC
s4: AGGGTGCCGCAATAC
s5: AGGGTGCCACAATAC
```

Alignment 2

```
s1: ACCGTGAAGCCAATAC
s3: AGCGTGCAGCCAATAC
```

Alignment 3

```
s1: ACCGTGAAGCCAATAC
s3: AGCGTGCAGCCAATAC

s2: A-CGTGCAACCATTAC
```



Alignment 4

```
s1: ACCGTGAAGCCAATAC
s3: AGCGTGCAGCCAATAC

s2: A-CGTGCAACCATTAC

s4: AGGGTGCCGC-AATAC
s5: AGGGTGCCAC-AATAC
```

Alignment 1

```
s4: AGGGTGCCGCAATAC
s5: AGGGTGCCACAATAC
```

Clustal: a progressive alignment approach

- a series of MSA tools based on the **progressive alignment**
- Clustal (Higgins & Sharp 1988), ClustalV (Higgins et al. 1992)
- **ClustalW** (Thompson *et al.* 1994) – improvement through sequence weighting, position-specific gap penalties and weight matrix choice
- **Clustal Omega** (Sievers *et al.* 2011) – more scalable

Three basic steps:

1. Assess **pairwise sequence similarity** using scores from all possible pairwise alignments
2. Establish an hierarchical order using a **guide tree** based on **UPGMA** or **Neighbour-joining (NJ)**
3. Align sequences **progressively** based on the defined order

Step 1: pairwise alignment

- given a set of sequences, pairwise alignment is performed on all possible pairs
- pairwise distance for each pairwise alignment is then determined
- $\boldsymbol{n}$ number of sequences gives $\boldsymbol{n(n\text{-}1)/2}$ pairwise alignments, i.e. "$\boldsymbol{n}$ choose $\boldsymbol{2}$": $\boldsymbol{C(n,2)}$ or ${}^{n}\boldsymbol{C}_{2}$

Example

```
Cattle   STCVLSAYWKDLNNYH
Human    STCMLGTYQDFNKFH
Pig      STCVLSAYWRNELNNFH
Rat      STCMLGTYQDLNKFH
Salmon   STCVLGKLSQELHKLQ
Sheep    STCVLSAYWKDLNNYH
```

Step 1: pairwise alignment

```
Cattle   STCVLSAYWKDLNNYH
Human    STCMLGTYQDFNKFH
Pig      STCVLSAYWRNELNNFH
Rat      STCMLGTYQDLNKFH
Salmon   STCVLGKLSQELHKLQ
Sheep    STCVLSAYWKDLNNYH
```

Example

```
Sheep    STCVLSAYWKDLNNYH
Cattle   STCVLSAYWKDLNNYH

Sheep    STCVLSAYWK-DLNNYH
Pig      STCVLSAYWRNELNNFH

Sheep    STCVLSAYWKDLNNYH
Human    STCMLGTY-QDFNKFH

Sheep    STCVLSAYWKDLNNYH
Rat      STCMLGTY-QDLNKFH

Sheep    STCVLSAYWKD-LNNYH
Salmon   STCVLGKL-SQELHKLQ

Pig      STCVLSAYWRNELNNFH
Human    STCMLGTY-QD-FNKFH

Pig      STCVLSAYWRNELNNFH
Rat      STCMLGTY-QD-LNKFH

Pig      STCVLSAYWRNELNNFH
Salmon   STCVLGKL-SQELHKLQ

Human    STCMLGTYQDFNKFH
Rat      STCMLGTYQDLNKFH

Human    STCMLGTY-QDFNKFH
Salmon   STCVLGKLSQELHKLQ

Rat      STCMLGTY-QDLNKFH
Salmon   STCVLGKLSQELHKLQ

etc.
```

Step 2: establish hierarchical order

Calculate **pairwise distance** (e.g. number of differing aligned positions)

Example

```
Cattle   STCVLSAYWKDLNNYH
Human    STCMLGTYQDFNKFH
Pig      STCVLSAYWRNELNNFH
Rat      STCMLGTYQDLNKFH
Salmon   STCVLGKLSQELHKLQ
Sheep    STCVLSAYWKDLNNYH
```

0
```
Sheep    STCVLSAYWKDLNNYH
Cattle   STCVLSAYWKDLNNYH
```

4
```
Sheep    STCVLSAYWK-DLNNYH
Pig      STCVLSAYWRNELNNFH
```

8
```
Sheep    STCVLSAYWKDLNNYH
Human    STCMLGTY-QDFNKFH
```

7
```
Sheep    STCVLSAYWKDLNNYH
Rat      STCMLGTY-QDLNKFH
```

11
```
Sheep    STCVLSAYWKD-LNNYH
Salmon   STCVLGKL-SQELHKLQ
```

9
```
Pig      STCVLSAYWRNELNNFH
Human    STCMLGTY-QD-FNKFH
```

8
```
Pig      STCVLSAYWRNELNNFH
Rat      STCMLGTY-QD-LNKFH
```

10
```
Pig      STCVLSAYWRNELNNFH
Salmon   STCVLGKL-SQELHKLQ
```

1
```
Human    STCMLGTYQDFNKFH
Rat      STCMLGTYQDLNKFH
```

9
```
Human    STCMLGTY-QDFNKFH
Salmon   STCVLGKLSQELHKLQ
```

8
```
Rat      STCMLGTY-QDLNKFH
Salmon   STCVLGKLSQELHKLQ
```

etc.

…

Step 2: establish hierarchical order

Example

	Sheep	Cattle	Pig	Human	Rat	Salmon
Sheep	0	0	4	8	7	11
Cattle	**0**	0	4	8	7	11
Pig	**4**	**4**	0	9	8	10
Human	**8**	**8**	**9**	0	1	9
Rat	**7**	**7**	**8**	**1**	0	8
Salmon	**11**	**11**	**10**	**9**	**8**	0

- the most similar sequences should be aligned first, as these are the easiest, introducing the fewest mistakes (i.e. the **greedy principle**)
- we may need to create several intermediate alignments that will later be joined

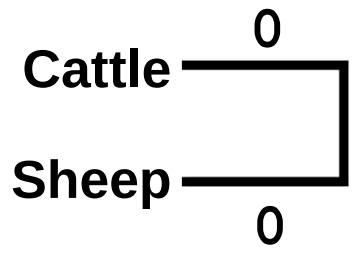
Step 2: establish hierarchical order using UPGMA

Unweighted **P**air **G**roup **M**ethod with **A**rithmetic mean

- agglomerative (“bottom up”) hierarchical clustering method
- at each step, the nearest two elements/clusters are combined (merged) into a higher-level cluster
- assumes **ultrametricity** (i.e. same root-to-tip distance for every branch tip)
- distance between clusters A & B = average distance between all element-pairs in A and in B

Example

	Sheep	Cattle	Pig	Human	Rat	Salmon
Sheep	0	0	4	8	7	11
Cattle	**0**	0	4	8	7	11
Pig	**4**	**4**	0	9	8	10
Human	**8**	**8**	**9**	0	1	9
Rat	**7**	**7**	**8**	**1**	0	8
Salmon	**11**	**11**	**10**	**9**	**8**	0

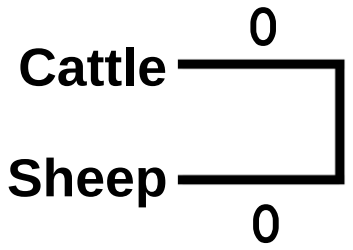


branch lengths not to scale

	Sheep	Cattle	Pig	Human	Rat	Salmon
Sheep	0					
Cattle	**0**	0				
Pig	**4**	**4**	0			
Human	**8**	**8**	**9**	0		
Rat	**7**	**7**	**8**	**1**	0	
Salmon	**11**	**11**	**10**	**9**	**8**	0

	Sheep+ Cattle	Pig	Human	Rat	Salmon
Sheep+ Cattle	0				
Pig	**4**	0			
Human	**8**	**9**	0		
Rat	**7**	**8**	**1**	0	
Salmon	**11**	**10**	**9**	**8**	0

Sheep and Cattle have the shortest distance (**0**), so they are merged first

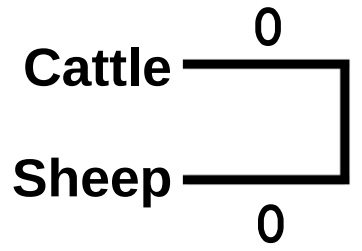


branch lengths not to scale

	Sheep+ Cattle	Pig	Human	Rat	Salmon
Sheep+ Cattle	0				
Pig	**4**	0			
Human	**8**	**9**	0		
Rat	**7**	**8**	**1**	0	
Salmon	**11**	**10**	**9**	**8**	0

	Sheep+Cattle	Pig	Human+Rat	Salmon
Sheep+Cattle	0			
Pig	**4**	0		
Human+Rat	**7.5**	**8.5**	0	
Salmon	**11**	**10**	**8.5**	0

Next, Human and Rat have the shortest distance (**1**), so they are merged



branch lengths not to scale

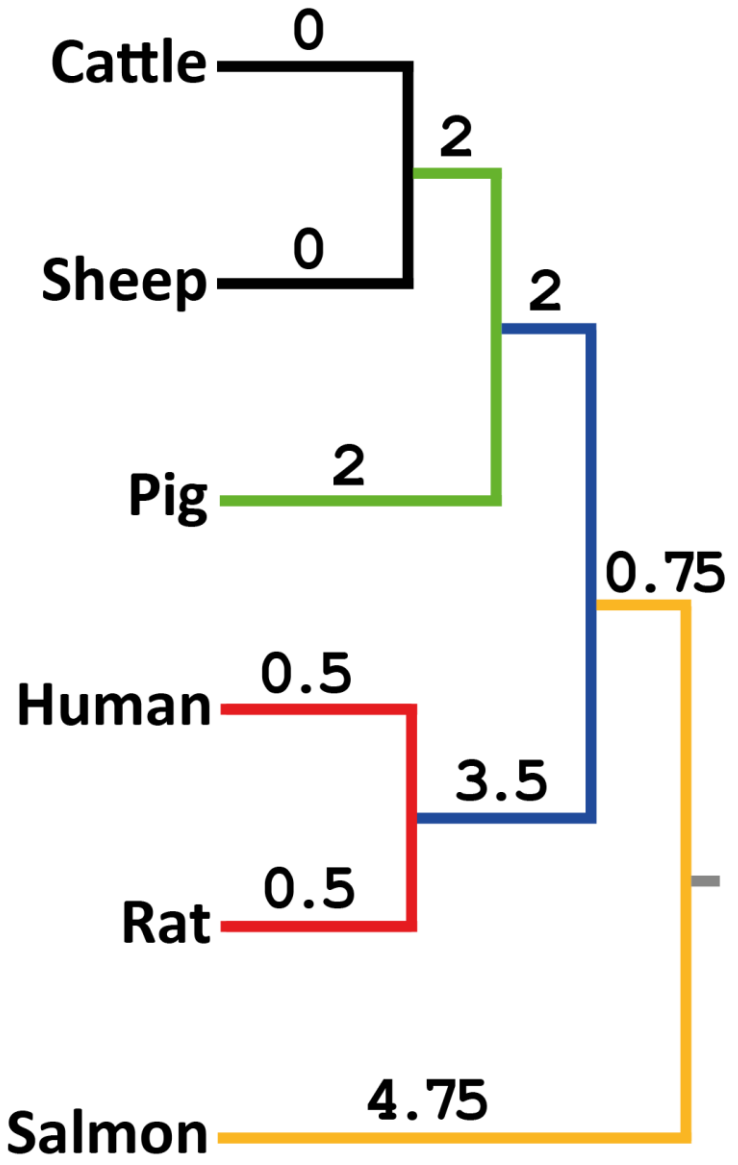
	Sheep+Cattle	Pig	Human+Rat	Salmon
Sheep+Cattle	0			
Pig	4	0		
Human+Rat	7.5	8.5	0	
Salmon	11	10	8.5	0

	Sheep+Cattle+Pig	Human+Rat	Salmon
Sheep+Cattle+Pig	0		
Human+Rat	8	0	
Salmon	10.5	8.5	0

	Sheep+Cattle+Pig+Human+Rat	Salmon
Sheep+Cattle+ Pig+Human+Rat	0	
Salmon	9.5	0

Example

and so forth ...

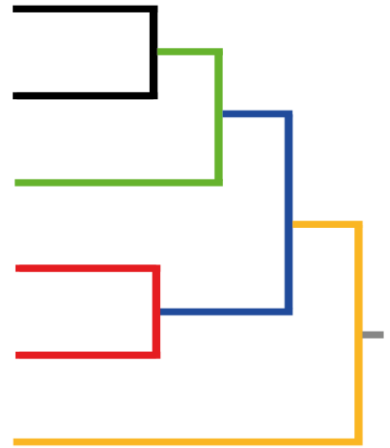


branch lengths not to scale

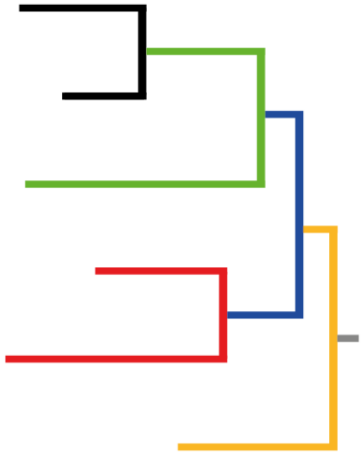
Step 2: establish hierarchical order

using Neighbour-Joining (NJ)

- proceeds in similar way as UPGMA, but based on a different distance matrix
- NJ does not assumes **ultrametricity**
- NJ is considerably more-robust to deviation from ultrametricity than UPGMA
- Progressively adding structure to the tree by joining the pair of clusters separated by the **shortest mean distance**
- default in ClustalW; slower than UPGMA



UPGMA tree



NJ tree

Step 3: progressive alignment

- pairwise alignment of alignments (**profile alignment**)
- dynamic programming can be applied:

$$S_{i,j} = \max \begin{pmatrix} S_{i-1,j-1} + m(x_i, y_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{pmatrix}$$

where $m(x_i, y_j)$ is the similarity score averaged over characters at that position, and that x_i and y_j each is a set of aligned residues from one or more sequences

Step 3: progressive alignment

Alignment 1

```
Cattle   STCVLSAYWKDLNNYH
Sheep    STCVLSAYWKDLNNYH
```

Alignment 2

```
Human    STCMLGTYQDFNKFH
Rat      STCMLGTYQDLNKFH
```

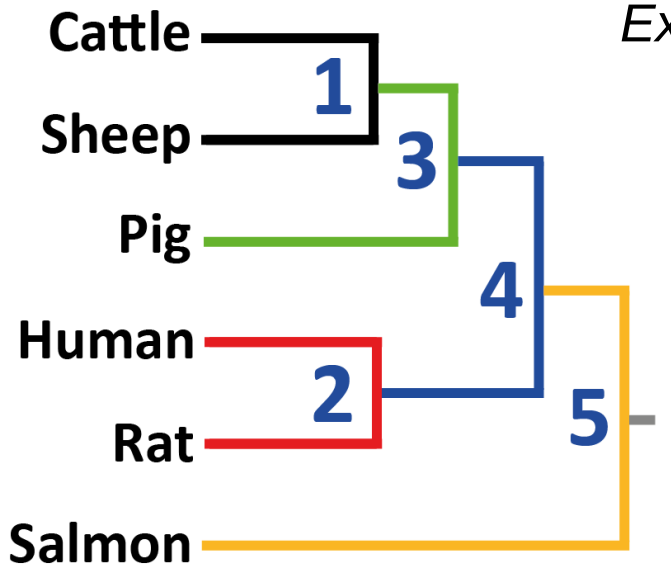
Alignment 3

```
Cattle   STCVLSAYWK-DLNNYH
Sheep    STCVLSAYWK-DLNNYH
Pig      STCVLSAYWRNELNNFH
```

Alignment 4

```
Cattle   STCVLSAYWK-DLNNYH
Sheep    STCVLSAYWK-DLNNYH
Pig      STCVLSAYWRNELNNFH
Human    STCMLGTYQD--FNKFH
Rat      STCMLGTYQD--LNKFH
```

Example



Alignment 5

```
Sheep    STCVLSAYWK-DLNNYH
Cattle   STCVLSAYWK-DLNNYH
Pig      STCVLSAYWRNELNNFH
Human    STCMLGTYQD--FNKFH
Rat      STCMLGTYQD--LNKFH
Salmon   STCVLGKLSQ-ELHKLQ
         ****:*.      :::  :
```

Final alignment – might not be the best/optimal solution

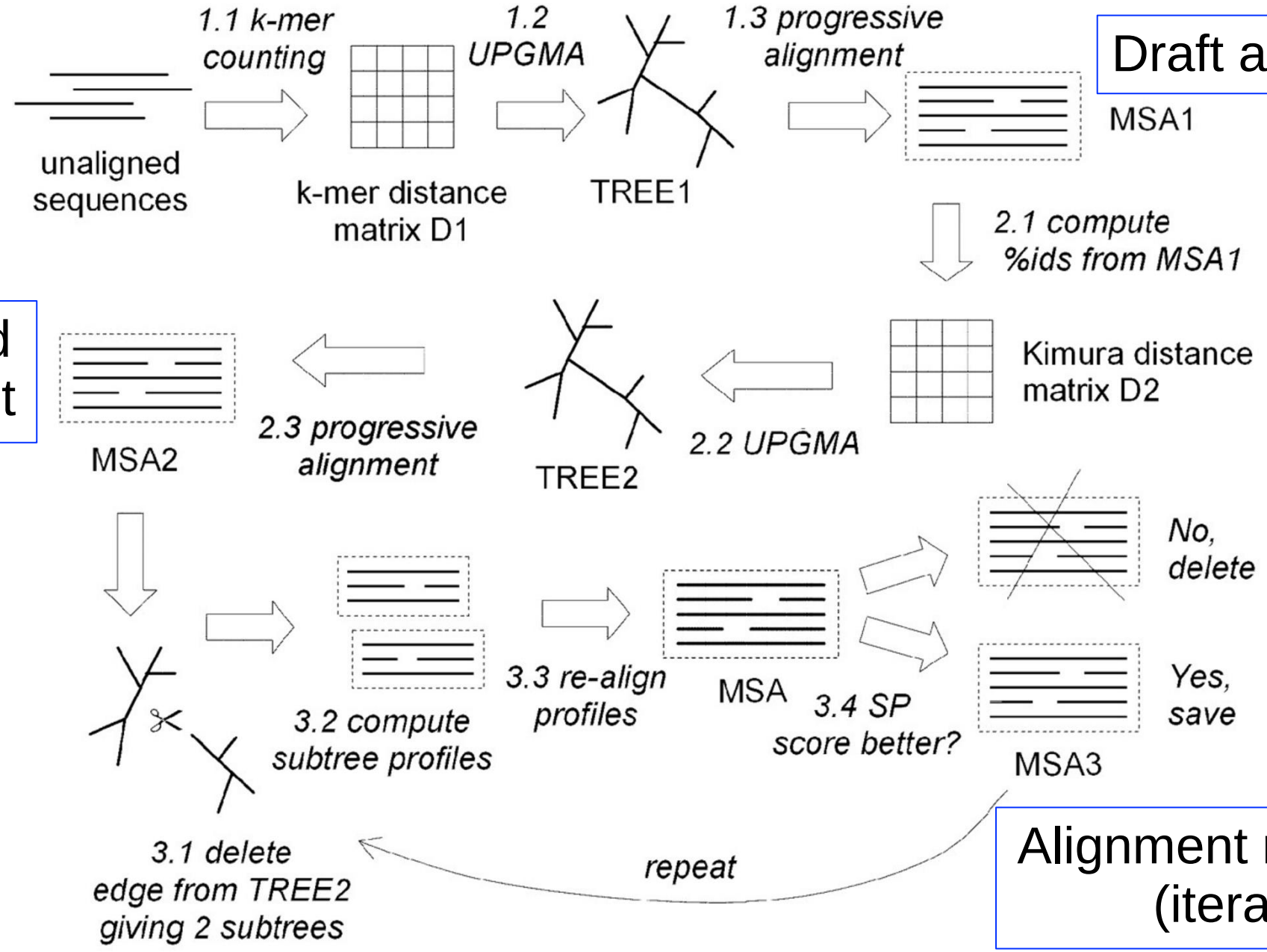
Progressive MSA: limitations

- tree might be **incorrect**, thus causing incorrect ordering of how sequences should be stacked up in the alignment
- once sequences are aligned and gaps introduced, these are **not altered**
- these early errors will be **propagated** and reflected in the final alignment, e.g. ClustalW finds a local optimum when early alignment decisions are “locked in” by the **“greedy”** algorithm
- final results **prone to errors** in alignment - some positions might be misaligned, i.e. the alignment could have a lower score than another alignment if a different ordering were used

Iterative progressive MSA

- aims to reduce the errors inherent in progressive methods
- works similarly to progressive methods (i.e., they are **iterative progressive** methods)
- repeatedly **realigns** the initial sequences as well as **adding new sequences** to the growing MSA
- can return to previously calculated pairwise alignments, or sub-alignments (subset of an alignment) incorporating the query sequence, in attempts to **optimise/refine** the overall MSA (to yield MSA with a higher score)
- MUSCLE is the most popular program: refines each tree branch independently using a draft tree and a refined tree
- other programs include DIALIGN, PRRN/PRRP

Iterative MSA: MUSCLE overview



Edgar RC (2004)
Nucl. Acids Res.
32: 1792-1797.

Other MSA approaches

Consensus methods

- attempt to find the optimal MSA given multiple different MSAs of the same set of sequences (i.e. a library of MSAs) based on **consensus**, e.g. T-COFFEE
- could adopt a meta-method approach, making use of the different MSA programs e.g. M-COFFEE

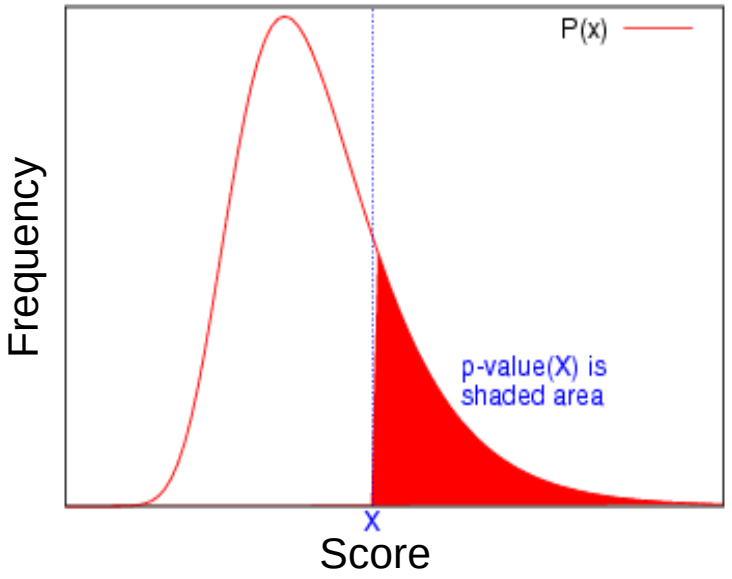
Methods based on **Hidden Markov models** (HMMs)

- uses **probabilistic models** to assign likelihoods to all possible combinations of gaps, matches, and mismatches to determine the most likely MSA or set of possible MSAs

Others e.g. **machine-learning** methods (genetic algorithms, simulated annealing) and **phylogeny-aware** methods are available; they are more computationally expensive (and less commonly used)

Measuring significance of an alignment

- statistical significance of an alignment score is used to assess whether an alignment is a result of **homology** or simply **random chance** (i.e. the biological relevance of the alignment)
- The ***p*-value** of an alignment score is the probability that a random alignment would have a an equal or higher score
- Of particular importance in database searching



Modelling score distribution

For ungapped local alignments, the distribution can be computed analytically

For gapped alignments, it must be estimated empirically

Sequence alignment *vs* structural alignment

A. Structural alignment from BAliBase (an alignment benchmark database)

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKTNGKFLIRARD--NNGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGK-KFDTLWQLVEHYSYKA------DGLLRVL-TVPCQK
1gri  EMKPHPWFFGKIPRAKAEEML-SKQRHDGAFLIRESES-APGDFSLSVKFGNDVQHFKVLRDGAGKYFL-WVV-KFNSLNELVDYHRSTS-VSRNQQIFLRDIEQVPQQ-
1aya  ---MRRWFHPNITGVEAENLLLTRG-VDGSFLARPSKS-NPGDFTLSVRRNGAVTHIKIQN--TGDYYDLYGGEKFATLAELVQYYMEHHGQLKEKNGDVIEL-KYPLN-
2pna  -LQDAEWYWGDISREEVNEKLRDT--ADGTFLVRDASTKMHGDYTLTLRKGGNNKLIKIFH-RDGKYGFSDPL-TFNSVVELINHYRNES-LAQYNPKLDVKL-LYPVS-
1bfi  HHDEKTWNVGSSNRNKAENLLRGK--RDGTFLVRESS--KQGCYACSVVVDGEVKHCVINKTATG-YGFAEPYNLYSSLKELVLHYQHTS-LVQHNDSLNVTL-AYPVYA
```

B. Multiple sequence alignment using DIALIGN (iterative method)

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKT-NGKFLIRAR-DN--NGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGKK-FDTLWQLVEHYSYKA-------DGLLRVLT-VPCQK
1gri  EMKPHPWFFGKIPRAKAEEML--SKQRHDGAFLIRESESA--PGDFSLSVKFGNDVQHFKVLRDGAGKYFLWVV-K-FNSLNELVDYHRST--SVSRNQQIFLRDIEQVPQQ-
1aya  M---RRWFHPNITGVEAENLLLTRGV--DGSFLARPSKSN--PGDFTLSVRRNGAVTHIKIQNTGDYYDLYG-GEK-FATLAELVQYYMEHHGQLKEKNGDV-IELK-YPLN-
2pna  LQDAE-WYWGDISREEVNEKL--RDTA-DGTFLVRDA-STKMHGDYTLTLRKGGNNKLIKIFHRDGKYGFSD-PLT-FNSVVELINHYRNE--SLAQYNPKLDVKLL-YPVS-
1bfi  HHDEKTWNVGSSNRNKAENLL--RGKR-DGTFLVRES-SK--QGCYACSVVVDGEVKHCVINKTATGYGFAE-PYNLYSSLKELVLHYQHT--SLVQHNDSLNVTLA-YPVYA
```

pink/red: alpha-helices
yellow: beta-sheets

C. Multiple sequence alignment using ClustalW (progressive method)

```
1csy  SHEKMPWFHGKISREESEQIVLIGSKTNGKFLIRARDN--NGSYALCLLHEGKVLHYRIDKDKTGKLSIPEGKKFD-TLWQLVEHYSYK------ADGLLRVLTVPCQK
1gri  EMKPHPWFFGKIPRAKAEE-MLSKQRHDGAFLIRESES-APGDFSLSVKFGNDVQHFKVLRDGAGKY-FLWVVKFN-SLNELVDYHRSTS-VSRNQQIFLRDIEQVPQQ
1aya  ---MRRWFHPNITGVEAEN-LLLTRGVDGSFLARPSKS-NPGDFTLSVRRNGAVTHIKIQNT-GDYYDLYGGEKFA-TLAELVQYYMEHHGQLKEKNGDVIELKYPLN-
2pna  -LQDAEWYWGDISREEVN--EKLRDTADGTFLVRDASTKMHGDYTLTLRKGGNNKLIKIFHR-DGKYGFSDPLTFN-SVVELINHYRNES-LAQYNPKLDVKLLYPVS-
1bfi  HHDEKTWNVGSSNRNKAE--NLLRGKRDGTFLVRESSK--QGCYACSVVVDGEVKHCVINKT-ATGYGFAEPYNLYSSLKELVLHYQHTS-LVQHNDSLNVTLAYPVYA
```

Which one of these alignments is the best?

"All the existing validation approaches have in common their reliance on the "one size fits all" **assumption** that structurally correct alignments are the best possible MSAs for modeling any kind of **biological signal** (**evolution**, **homology**, or **function**). … it may be reasonable to ask *whether better alignments always result in better phylogenetic trees*, and, more systematically, to *question and quantify the relationship between the accuracy of MSAs and the biological relevance of any model drawn upon them*."

Cédric Notredame (2007) *PLoS Computational Biology* 3(8): e123.

MSA: issues and challenges

```
Seq1 ATTTAAACGTCTAGATTTAA----------TAGCATGCGA
Seq2 ----------CTAGATTTAAATTTAAACGTTAGCATGCGA
```

- based on strict assumption of **whole-sequence contiguity**; largely heuristics (for feasibility)
- relevance of alignment scores to homology can be difficult to assess statistically
- loss of phylogenetic information in instances of genome rearrangements, lateral genetic transfer etc.

```
Seq1   ATTTAAACGT   CTAGATTTAA      TAGCATGCGA
Seq2   ATTTAAACGT   CTAGATTTAA      TAGCATGCGA
```

- ***Alignment-free (AF)* methods:** distances based on sub sequences of defined length (e.g., *k*-mers) – no assumption of whole-sequence contiguity
- AF methods are more scalable: on-going active field of research

Reflection

- *Is dynamic programming scalable for aligning multiple sequences?*
- *What information can we observe from an MSA? What do we commonly use MSA for?*
- *Why are heuristic approaches used for MSA?*
- *What is progressive MSA, and what are the key steps involved?*
- *What are the two common methods adopted to establish hierarchical order in progressive MSA?*
- *What is the main difference between UPGMA and NJ?*
- *What are the limitations of progressive MSA, and how can we improve it?*
- *What are some limitations of MSA, and how can we attempt to resolve these?*

Sequence Analysis 2

B. Analysis of high-throughput sequences

Cheong Xin Chan (CX)

c.chan1@uq.edu.au

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics 1 – Introduction

Lecture outline

- **High-throughput sequence data**
  - The data (sequence reads and quality)
  - Types of assemblies
- **Basic principles of genome assembly**
- **Strategies of genome assembly**
  - Overlap-layout-consensus
  - De Bruijn graph (*k*-mer-based, and examples)
  - Key terms and concepts
- **Issues and challenges**

High-throughput sequences

High-throughput sequences are short. Why?

Current sequencing technologies are not practical enough to read whole genomes in one go.

Shotgun sequencing breaks down genome into small fragments (e.g. 10^2 bases) then sequence these fragments in great depth (typically 10^6-10^8 sequence reads; hence **high throughput**).

These reads will need to be assembled to *re-assemble* the original genome sequence.

How?

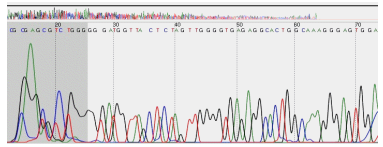
Identical regions of different reads can be collapsed into a long contiguous sequence (*à la* sequence alignment)

Regions of bad reads (with low quality score) can be down-weighted

How do we assess data quality?

A (very) simplified sequencing workflow

Sequencing



Trace data

Base calling

Sequence reads

Phred quality score (*Q*)

$$Q = -10(\log_{10}P)$$

where *P* = probability of a base-calling error

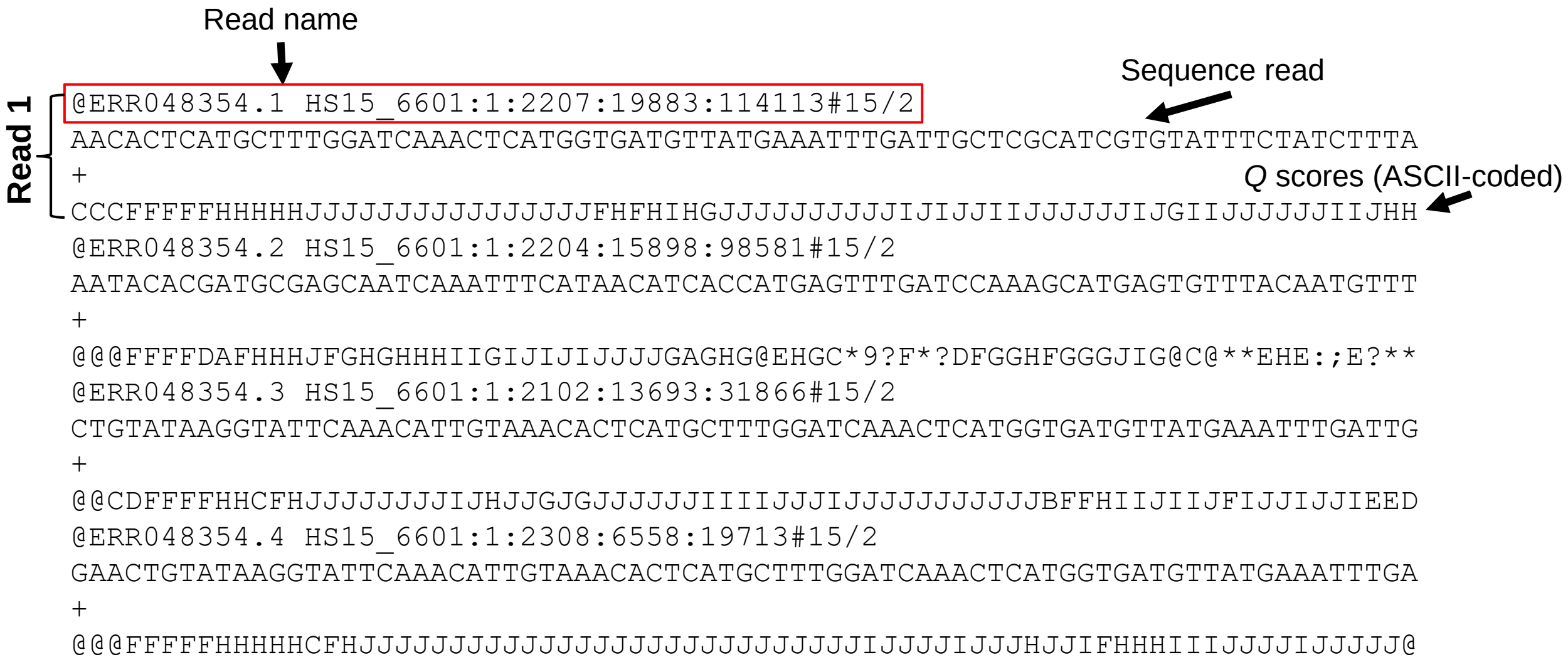
Phred Quality Score	Probability of Incorrect Base Call	Base Call Accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1,000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%

http://www.illumina.com/documents/products/technotes/technote_Q-Scores.pdf

$P = 0.001$

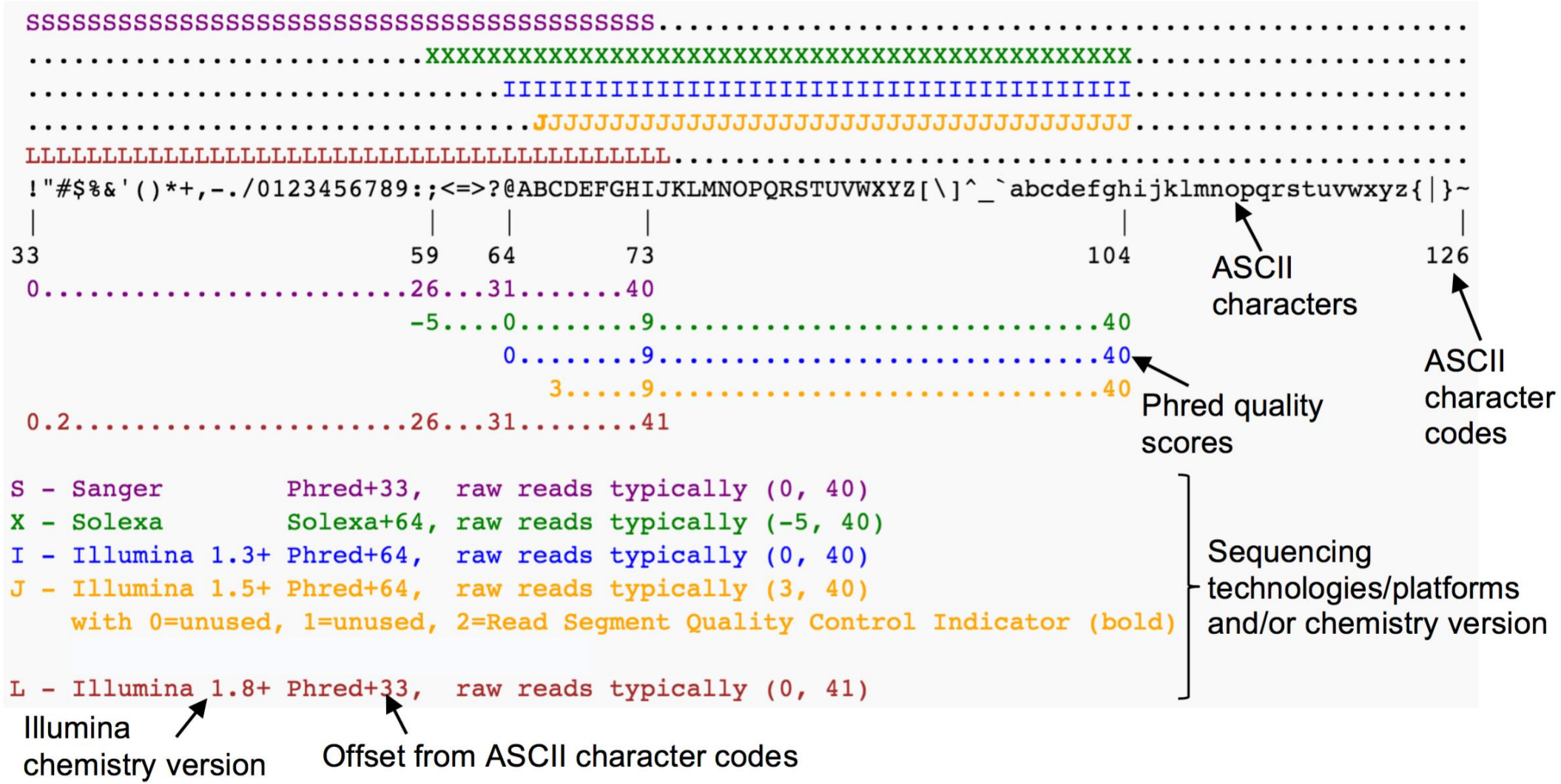
$Q = -10(\log_{10}[0.001]) = -10(-3) = 30$

High-throughput data: sequence reads



FASTQ format

Q scores in ASCII characters



Why is the offset 33?

The first 32 ASCII characters are non-printing characters (*e.g. esc, tab, backspace, ctrl, shift etc.*)

How can we tell if our data are good?

```
@ERR048354.1 HS15_6601:1:2207:19883:114113#15/2
AACACTCATGCTTTGGATCAAACTCATGGTGATGTTATGAAATTTGATTGCTCGCATCGTGTATTTCTATCTTTA
+
CCCFFFFFHHHHHJJJJJJJJJJJJJJJJJFHFHIHGJJJJJJJJJJJIJIJJIIJJJJJJIJGIIJJJJJIIJHH
@ERR048354.2 HS15_6601:1:2204:15898:98581#15/2
AATACACGATGCGAGCAATCAAATTTCATAACATCACCATGAGTTTGATCCAAAGCATGAGTGTTTACAATGTTT
+
@@@FFFFDAFHHHJFGHGHHHIIGIJIJIJJJJGAGHG@EHGC*9?F*?DFGGHFGGGJIG@C@**EHE:;E?**
@ERR048354.3 HS15_6601:1:2102:13693:31866#15/2
CTGTATAAGGTATTCAAACATTGTAAACACTCATGCTTTGGATCAAACTCATGGTGATGTTATGAAATTTGATTG
+
@@CDFFFFHHCFHJJJJJJJJIJHJJGJGJJJJJJIIIIJJJIJJJJJJJJJJJJBFFHIIJIIJFIJJIJJIEED
@ERR048354.4 HS15_6601:1:2308:6558:19713#15/2
GAACTGTATAAGGTATTCAAACATTGTAAACACTCATGCTTTGGATCAAACTCATGGTGATGTTATGAAATTTGA
+
@@@FFFFFHHHHHCFHJJJJJJJJJJJJJJJJJJJJJJJJJJJJJIJJJJIJJJHJJIFHHHIIIJJJJIJJJJJ@
...
```

Technical: base quality of the reads, presence of unwanted adapters or artefacts, sequencing errors/biases, etc.

Biological: presence of unwanted/contaminant sequence reads, adequacy of the data in addressing biological questions, etc.

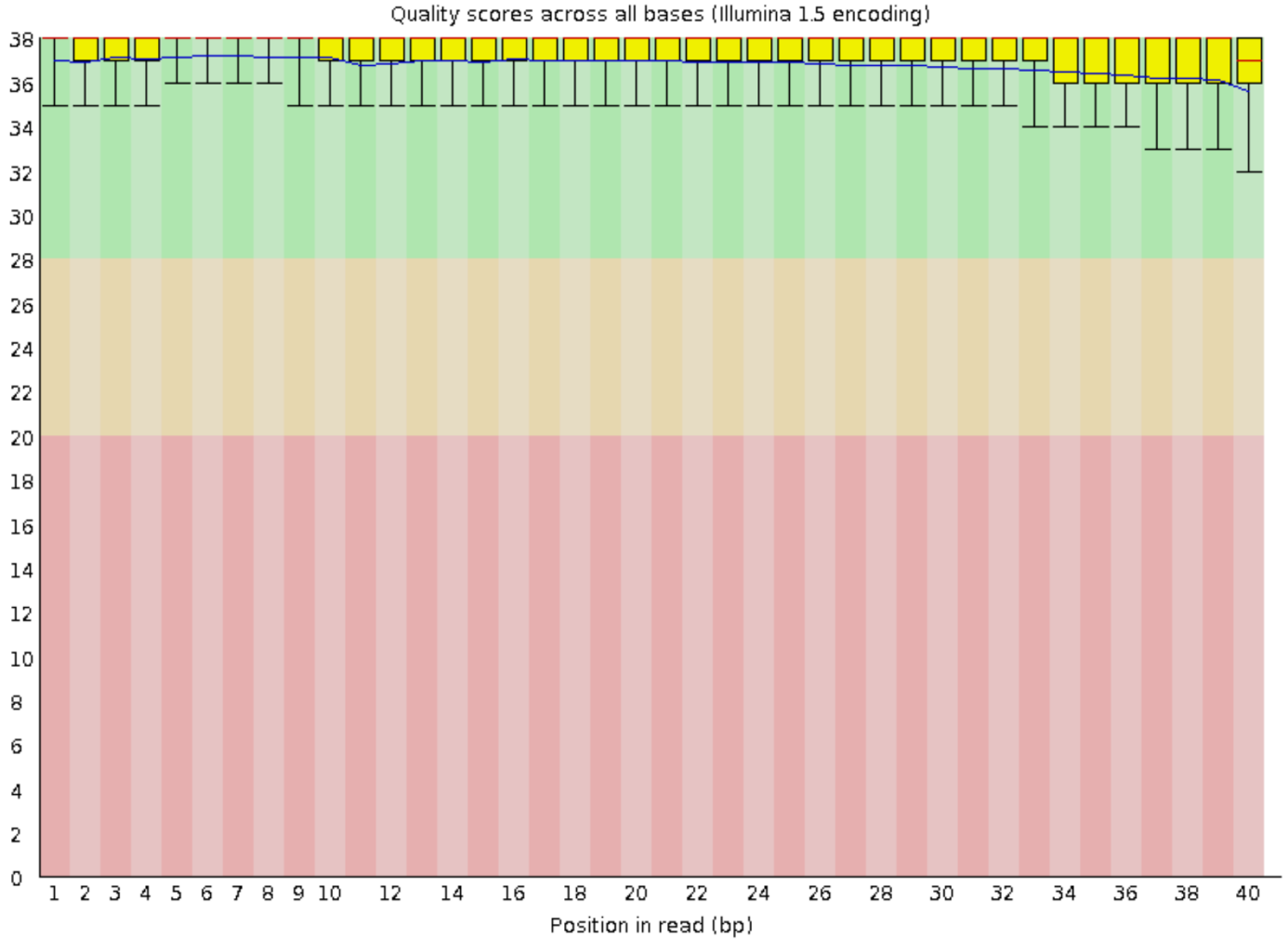
In an ideal world …

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

Per base sequence quality



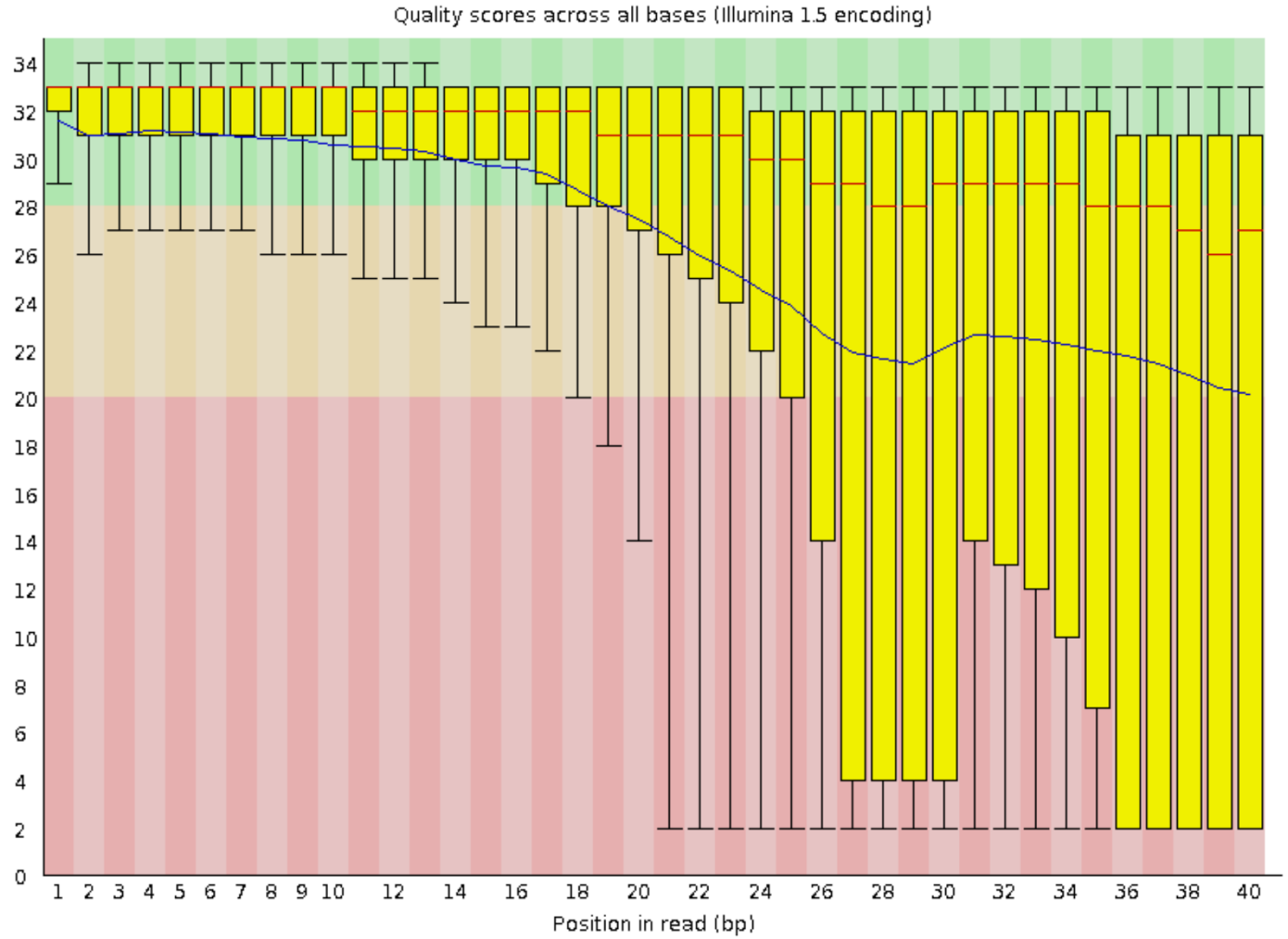
In reality …

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content

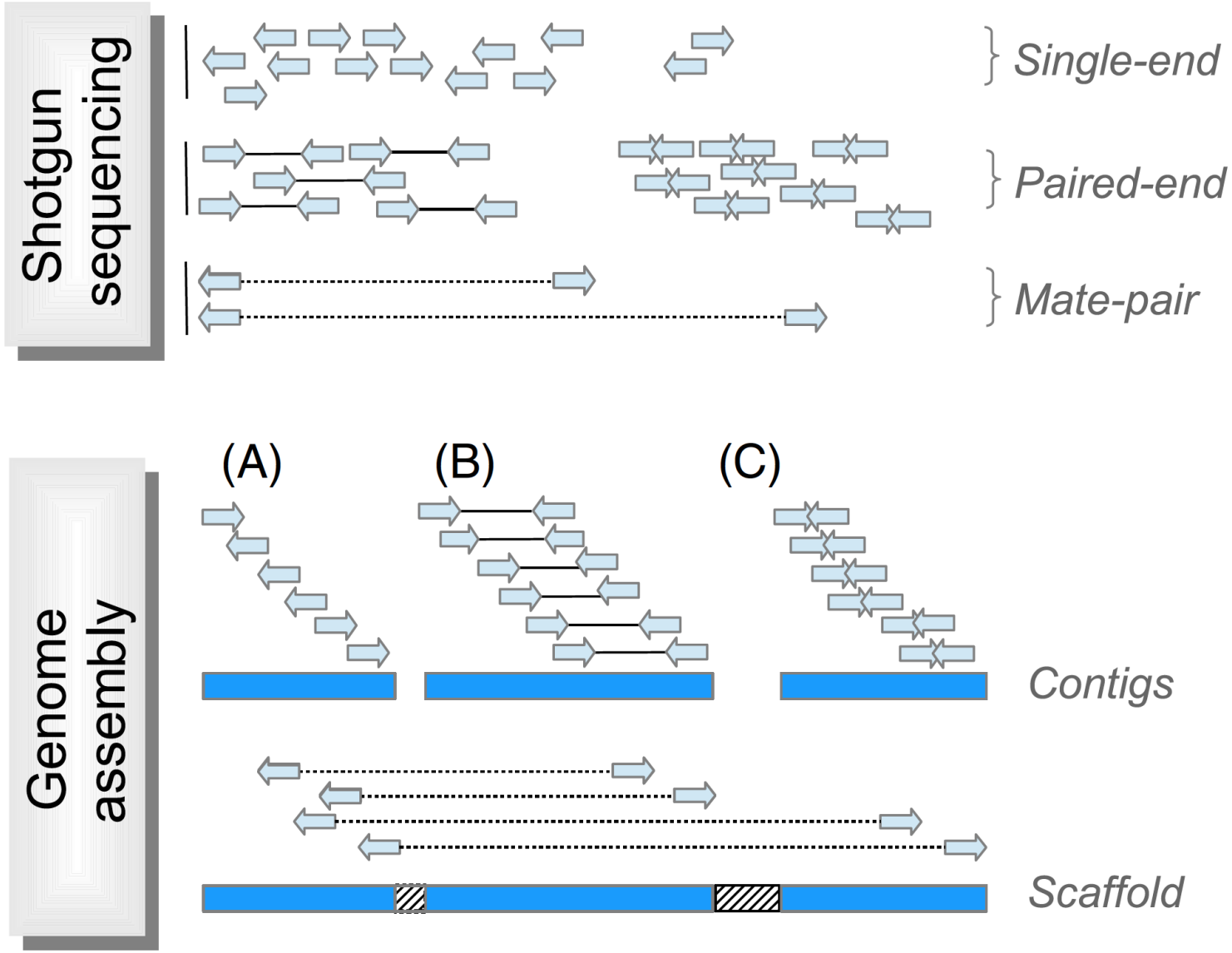
Per base sequence quality



Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CGGTTCAGCAGGAATGCCGAGA TCGGAAGAGCGGTTCAGC	12351	0.5224039	Illumina Paired End PCR Primer 2 (96% over 25bp)

Basic principles of genome assembly



Ekblom & Wolf (2014) *Evolutionary Applications* 7: 1026-42.

Genome *versus* transcriptome assemblies

Genome assembly

- Genome include genic and intergenic regions
- Often have large amounts of repetitive elements, esp. in the intergenic regions (e.g., introns)
- Long contigs are desirable

CLC Genomics Workbench
MaSuRCA
ALLPATHS-LG
SPAdes
Velvet
Celera
Newbler (for 454 data)
ABySS
SOAPdenovo
…

Transcriptome assembly

- Expressed genes - no intergenic regions
- No/less repetitive elements
- Housekeeping genes are highly expressed (higher coverage/more reads)
- Complication of alternative splicing, SNP, and post-transcriptional modification
- Long contigs might indicate multigene cluster due to over-assembly

Trinity
Velvet-Oases
Newbler (-cdna option)
CAP3
…

De novo versus mapping assemblies

De novo assembly

- Assembling a new, previously unknown sequence or genome
- More memory intensive due to computational complexity

New genomes

Mapping assembly

- Assembling reads against a reference sequence/genome
- Looking for an assembly that is similar (not necessarily identical) to the reference

- Resequencing projects
- Genomes of similar species
- Model genomes of animals/plants (e.g., humans, *Arabidopsis*)
- 1000-genome projects

…

Strategy of genome assembly

The **shortest common superstring** problem

Given a collection of strings, what is the shortest superstring that contains all these strings as substrings?

Example: *S*: BAA AAB BBA ABA ABB BBB AAA BAB

Concatenation: BAAAABBBAABAABBBBBAAABAB

24

Shortest common superstring: AAABBBABAA

10

AAA
AAB
ABB
BBB
BBA
BAB
ABA
BAA

*Given a collection of **sequence reads**, what is the **shortest sequence** that contains **all these reads** (as sub-sequences)?*

Strategy of genome assembly

Two major paradigms:

Overlap-layout-consensus (OLC)

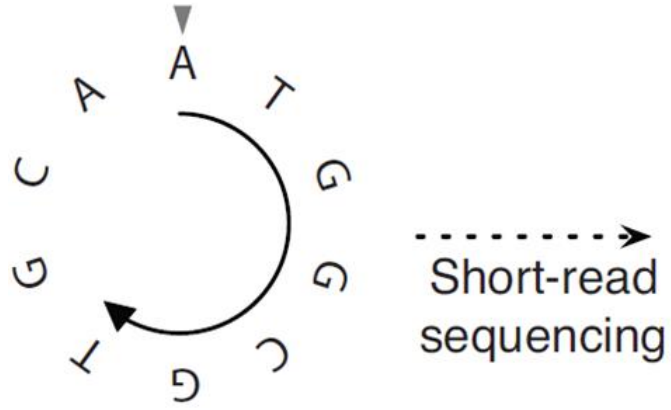
- identifies all pairs of reads that overlap sufficiently well and then organises this information into a graph
- e.g. Celera

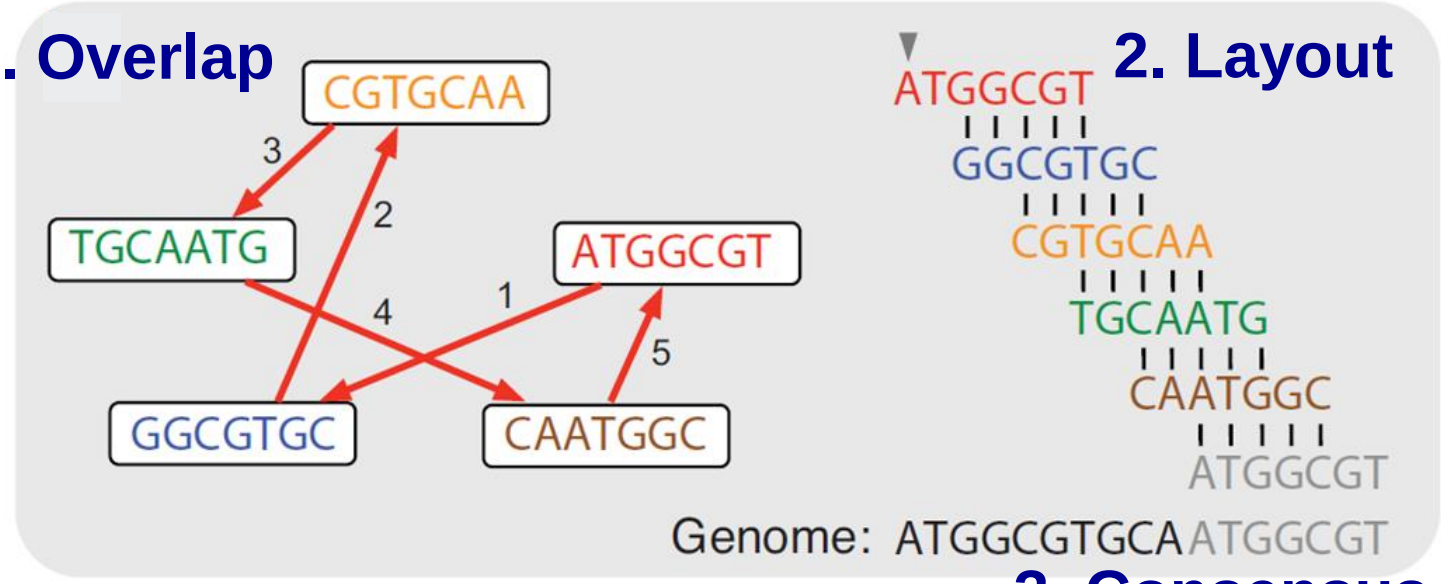
De Bruijn graph

- models the relationship (overlaps) between exact substrings of length *k* extracted from the input reads (*k*-mers)
- More popular; e.g. CLC-GW, MaSuRCA, ALLPATHS, SPAdes, Velvet, SOAPdenovo

Nagarajan & Pop (2013) *Nature Reviews Genetics* 14: 157-167.

Overlap-layout-consensus (OLC)





- Reads represented as **nodes**; alignments between reads as **edges**
- Genome reconstruction by combining alignments between successive reads
- Computationally expensive

Compeau *et al.* (2011) *Nature Biotechnology*, 29: 987-991.

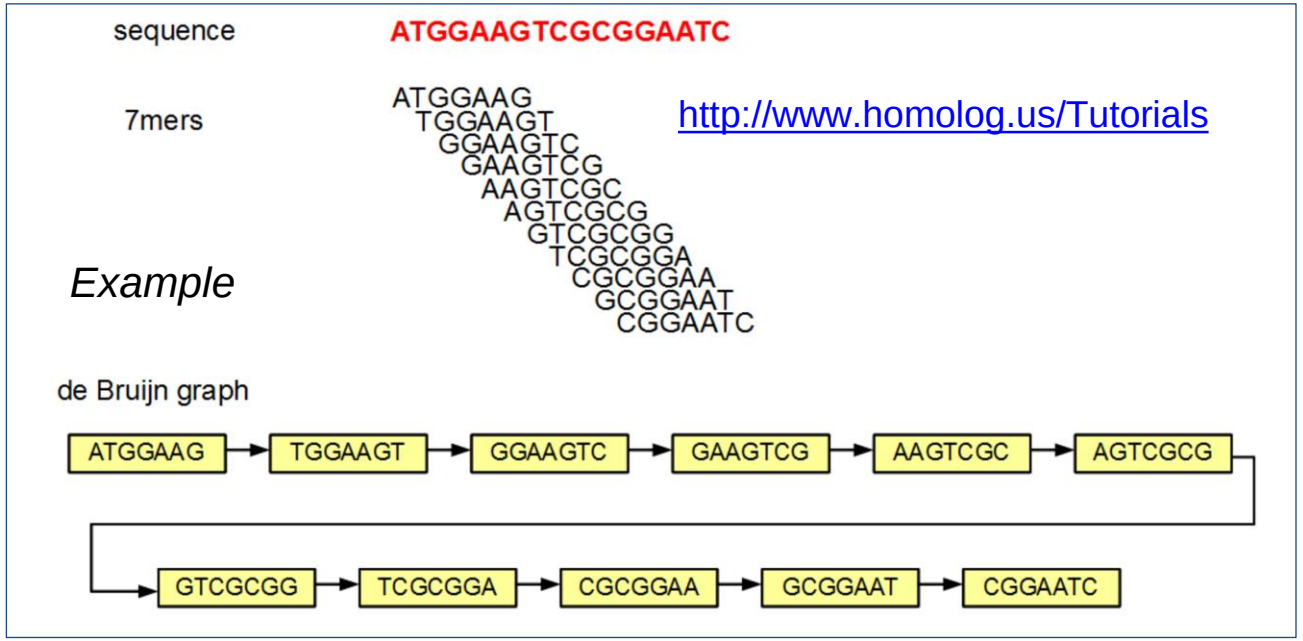
De Bruijn graph (based on *k*-mers)

What are *k*-mers?

TTGACACTTACCGA **Read**

TTGACACTTACC
TGACACTTACCG
GACACTTACCGA
k-mers for k=12

TTGAC
TGACA
GACAC
ACACT
CACTT
ACTTA
CTTAC
TTACC
TACCG
ACCGA
k-mers for k=5



- Obtain all *k*-mers at (*k* = 7 in this case) – these are **nodes**
- Construct directed graph between node pairs such that the connected nodes have (contiguous) overlaps of 6 (*k* - 1) nucleotides; these connections are **edges**.
- Find the **shortest (common) superstring**

De Bruijn graph

In a very simple example …

Example #1:

HAPPI PINE INESS APPIN

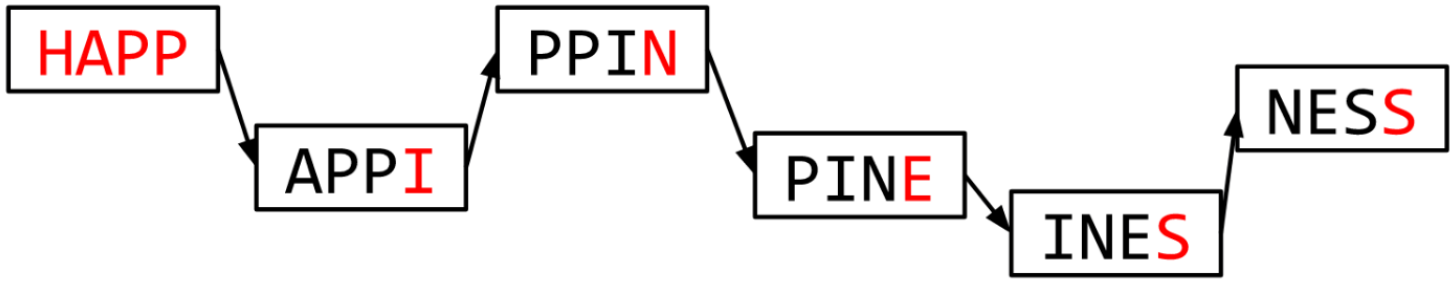
All 4-mers:

HAPP PINE INES APPI

APPI NESS PPIN

Unique 4-mers:

HAPP APPI PINE PPIN INES NESS



HAPPINESS

Identical **nodes** are merged/collapsed, reducing computational complexity.

De Bruijn graph

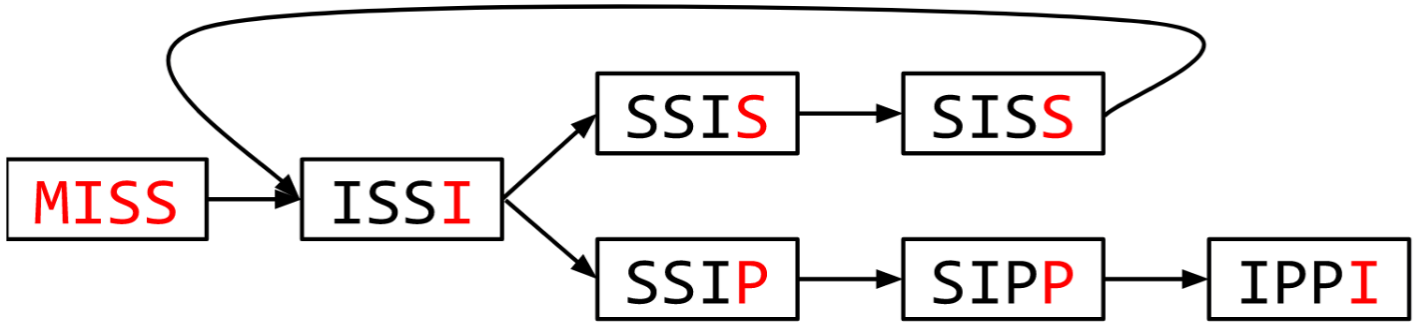
In a more-tricky example when repetitive sequence regions are present …

Example #2:

`MISSIS SSISSI SSIPPI`

All 4-mers (9):

```
MISS   SSIS   SSIP
 ISSI   SISS   SIPP
  SSIS   ISSI   IPPI
```



`MISSISSIPPI` or `MISSISSISSISSIPPI` or ...

Unique 4-mers (7):

`MISS SSIS SSIP ISSI SISS SIPP IPPI`

The same **node** can be used in assembling different sequences.

De Bruijn graph

Example #2a:

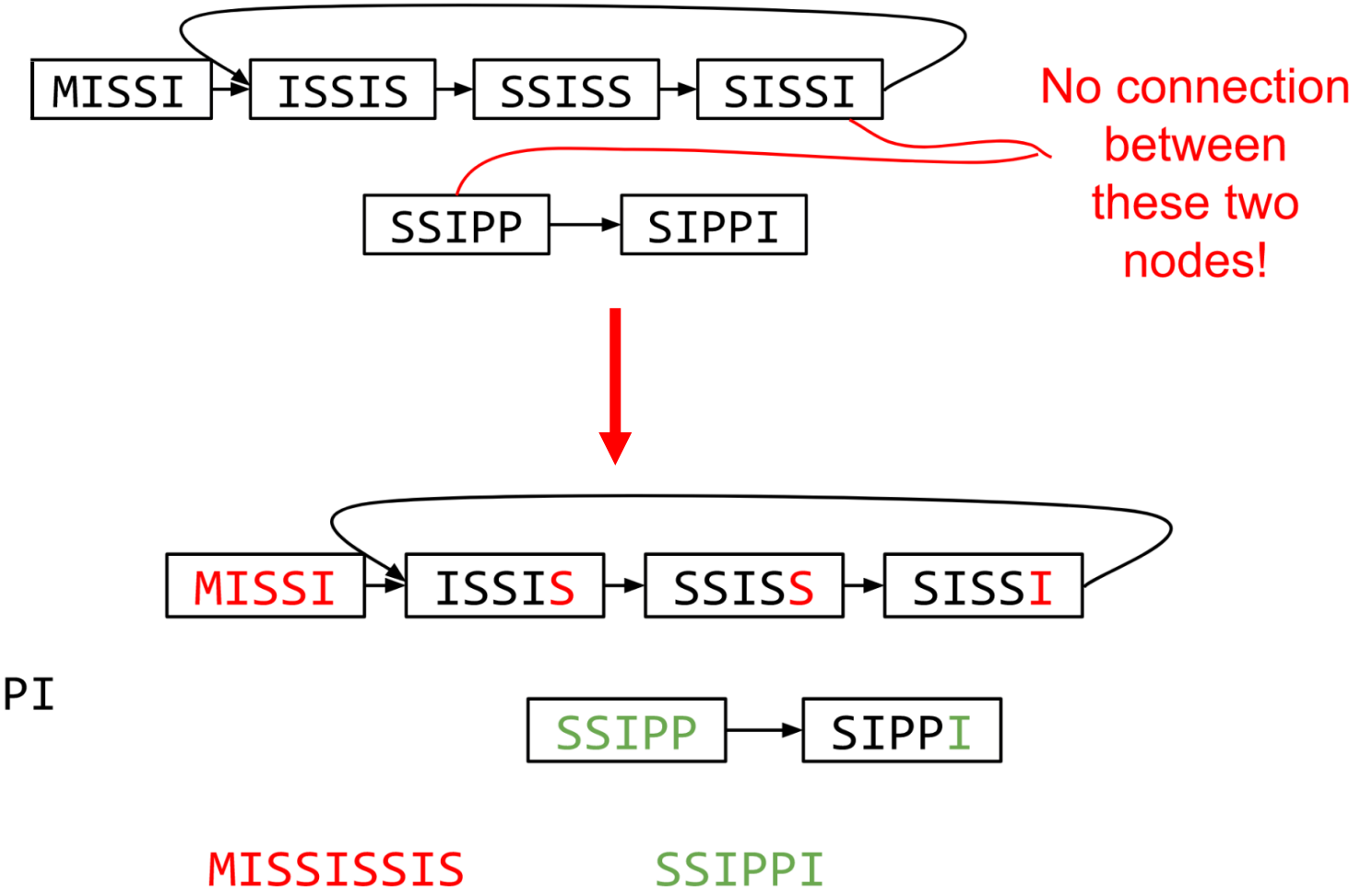
```
MISSIS SSISSI SSIPPI
```

All 5-mers (6):

```
MISSI  SSISS  SSIPP
 ISSIS  SISSI  SIPPI
```

Unique 5-mers (6, no duplicates):

```
MISSI ISSIS SSISS SISSI SSIPP SIPPI
```



Different *k* values yield different results.

k-mer length in de Bruijn graph

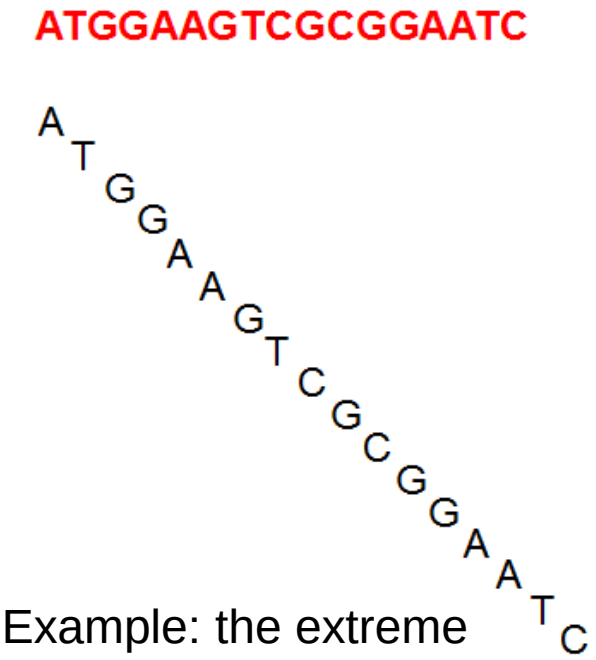
The choice of *k*-mer length (*k*) is crucial:

Short *k*:

- lowers overlap threshold (more overlaps), *k*-mers joined more readily
- generates large number of short contigs

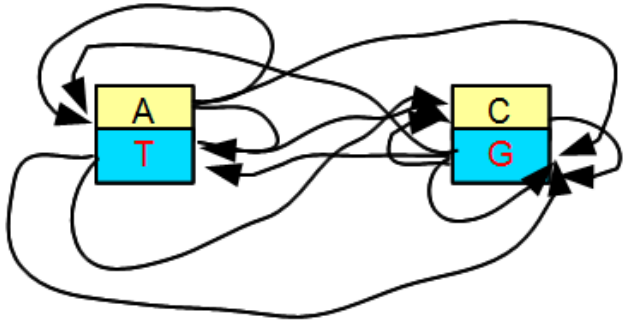
Long *k*:

- may not join contigs together when it should



Example: the extreme case of using 1-mers

It is difficult to predict the best *k* to use for each assembly, thus sometimes optimisation by trial-and-errors is necessary.



k-mer length in de Bruijn graph

7-mer (**ATATATA**):

-> ATATATA

TATATAT <-

Reverse complement of **ATATATA** is **TATATAT**

okay

6-mer (**ATATAT**):

-> ATATAT

TATATA <-

Reverse complement of **ATATAT** is **ATATAT**

confusing

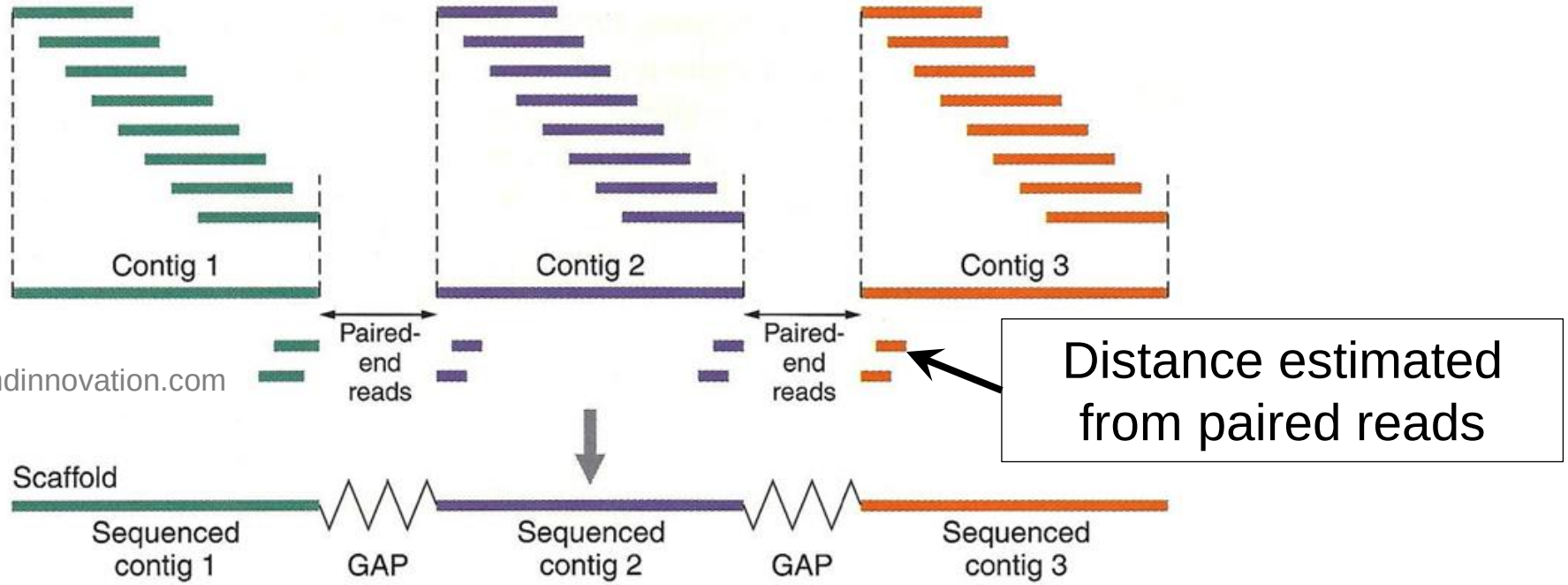
k is commonly an odd number to avoid palindromes

- If *k*-mers are of even length, some *k*-mers can be reverse complements of themselves (e.g. **ATATAT**)
- Genome assemblers commonly avoid even *k*

k must be shorter than the sequence read length, or else there will be no overlaps

Key terms and concepts

Contig: a contiguous linear stretch of consensus sequence that is constructed from a number of smaller, partially overlapping, sequence reads (fragments)



Scaffold: a sequence comprising two or more contigs that are joined together based on read-pair distance information (i.e. ordered, oriented contigs with NNNs in between)

Coverage (sequencing depth): the average number of reads representing a given nucleotide in an assembled sequence (e.g. genome)

Example: a contig of 30 bases

```
Read1       GATCTGGAATTCTCGGGCAC
Read2          CTGGAATTCTCGGGCACCAA
Read3           TGGAATTCTCGGGCACCAAG
Read4                 TCTCGGGCACCAAGGTACGC
Contig      GATCTGGAATTCTCGGGCACCAAGGTACGC
Base        11123333334444444443332111111
Coverage
```

4 reads of 20 bases = 80 bases constitute this 30-base contig

Contig coverage = **80 / 30** = **2.67**

N50: Contig length such that using equal or longer contigs produces half (50%) of the bases of the total assembled bases (sum of all contig lengths); the same applies to scaffolds

Example: a genome assembly

Contig	*Length*	*Cumulative Sum*
Contig3	295,492	295,492
Contig2	259,553	555,045
Contig6	142,866	697,911
Contig1	136,171	834,082
Contig9	135,129	969,211
Contig7	117,473	1,086,684
Contig10	115,625	1,202,309
Contig4	102,105	1,304,414
Contig8	77,713	1,382,127
Contig5	76,819	**1,458,946**

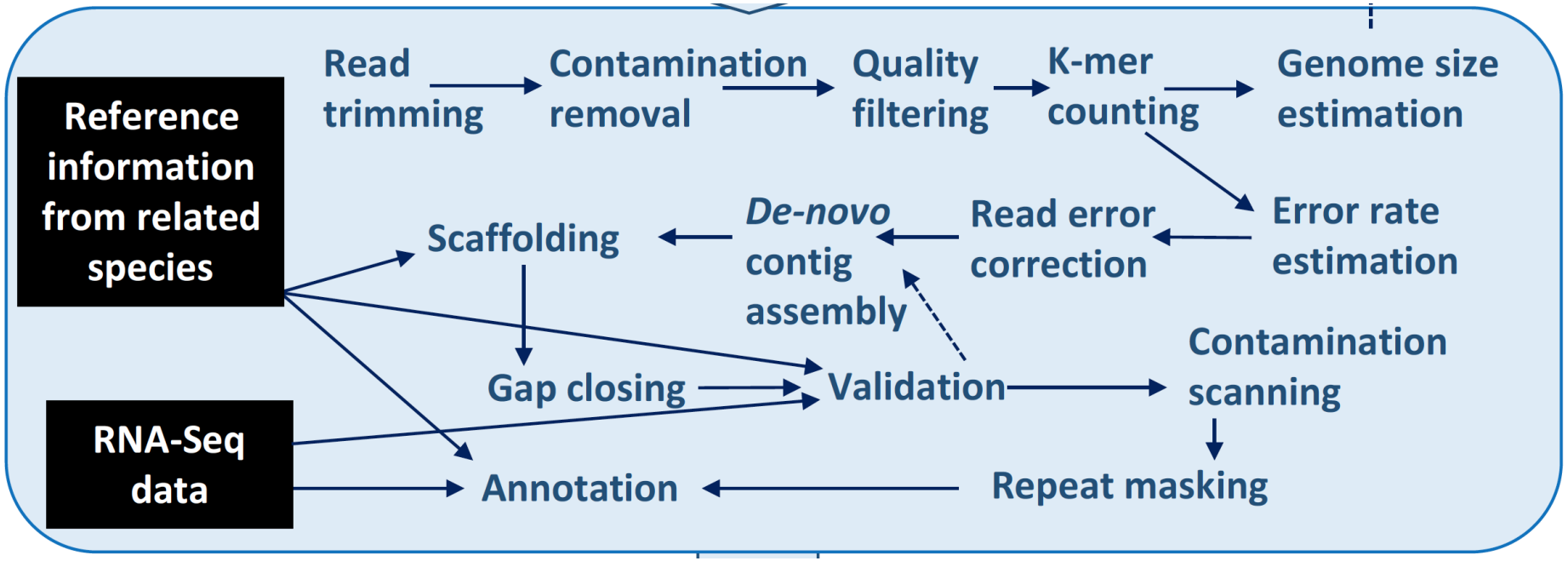
50% of bases = 1,458,946 / 2 **= 729,473**

The **first four** contigs make up ≥ 729473 bases therefore **N50 = 136,171**

How about N75?

25% of bases = 1,458,946 / 4 **= 364,736.50**

The first **two** contigs make up ≥ 364,736.50 bases. **N25** = **259,553**



A typical process of *de novo* genome assembly

Ekblom & Wolf (2014) *Evolutionary Applications* 7: 1026-42.

Read trimming: the removal of adapter sequences and low-quality/ambiguous bases from the sequence reads

Read mapping: the alignment of short sequence reads to a longer sequence (e.g. contig, scaffold, reference genome)

Gap filling/closing: the process of replacing the Ns in scaffolds with nucleotide bases based on read mapping

Issues and challenges

- Peculiarity (little-understood features) in genomes, prokaryotes versus eukaryotes
- Repetitive elements in genomes (cause error in assembly, increase time/space complexity)
- Computational tractability, memory and storage space, esp. when dealing with huge amount of data
- Sequencing error (e.g., assignment of incorrect base, under-/over-estimation of base quality scores
- All these intensified with *de novo* assemblies

Reflection

- *What are high-throughput sequence data?*
- *How do we assess the quality of high-throughput sequence data?*
- *What are the basic principles of genome assembly? Why do we need to assemble a genome (or transcriptome)?*
- *What are the differences between a genome assembly and a transcriptome assembly?*
- *What are the differences between a de novo assembly and a mapping assembly?*
- *What are the two major paradigms of genome assembly?*
- *What is a De Bruijn graph assembly?*
- *How do we choose the k-mer length in De Bruijn graph assembly?*
- *What are the key properties in genome assembly? How do we calculate N50 length?*
- *What are the key issues and challenges facing genome assembly?*

Sequence Analysis 3A:

Introduction to sequence motifs

Katherine Dougan, PhD

Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics I - Introduction

Outline

- **Introduction to sequence motifs**
  - What are they?
  - What makes them difficult to identify?
- **Discrete representations**
  - Consensus sequences
  - Degenerate consensus sequences
  - Regular expressions
- **Examples of sequence motifs**

What is a *sequence motif*?

A sequence motif is a ***short, conserved*** nucleotide or amino acid sequence that is known, or predicted, to have a ***specific biological function***.

Sequence motif ≠ Structural motif

(At least not necessarily)

Sequence motifs can be difficult to find

```
 Seq 1: TCATTGGTCCTCAGGATCACGCGACAGGAAGTGTGGCGTAACCGGTTGACTGCCACATGCGCATTGGCTTCCAGGGCC
 Seq 2: TCATTGATGCGCATTGGCTTCCAGGGGTCCTCAGGATCACACCACAGCCCGCTGGAATAAGTGTGGCGTAAGCCACCC
 Seq 3: CACACCTTTAATTGTTGCAGGATGAATCAGAGGAGGTGTGGCAGTAAACAAGAATGAACCCCACAGCTTCACACTTCC
 Seq 4: TACTGGCGCCAGAGCCAATTTGCGTCATCTAACTAAAGATTTCAACAGCAGTGATATATCTTACTCAAGTGTGGCTAG
 Seq 5: CAAGGAGTGTGGATACAAAATTGCGCAACAGAGAGCCATCACTCACCACAGCCCGCTGGAATTCTCATGCTGGAGCAA
 Seq 6: CATTGTTGCAGGACCACAGCTTCGAGGTGTGGCAACACCTTTAGTAAACACTTCCTGAATCAGAGACAAGAATGAACC
 Seq 7: ACACATCCGTGTGGCGATTGGGCGGCGTAACCTCGCTTATTTGCATAGGCCGATTGCACAACCGGGCGGCGACCTCAG
 Seq 8: ACACGGCCGATTGCACAACCACCGACCTCAGTGGGAACCGGGCGGGATCCGTGTGGCGATTGAGGCCGATTGCACCTC
 Seq 9: TTAAGAGAATGTCATTGCGGTGTGGCTGAGGGGGAGGGAGAGGTGAGGGTGCAACTTGGGTAAAGGTTGTGGAGGCAT
Seq 10: TACCACTAGCTGCCCTAACTCTTACTAATTAGCGCCAGAGACAAAGTGTGGTTCAAAGATGCAGTGATATATTGCGTC
```

There's a DNA-binding motif here…. can you find it?

Sequence motifs can be difficult to find

```
 Seq 1: TCATTGGTCCTCAGGATCACGCGACAGGAAATGTGGCGTAACCGGTTGACTGCCACATGCGCATTGGCTTCCAGGGCC
 Seq 2: TCATTGATGCGCATTGGCTTCCAGGGGTCCTCAGGATCACACCACAGCCCGCTGGAATAAGTGTGGCGTAAGCCACCC
 Seq 3: CACACCTTTAATTGTTGCAGGATGAATCAGAGGAGGTGTGGCAGTAAACAAGAATGAACCCCACAGCTTCACACTTCC
 Seq 4: TACTGGCGCCAGAGCCAATTTGCGTCATCTAACTAAAGATTTCAACAGCAGTGATATATCTTACTCAAGTGTGGCTAG
 Seq 5: CAAGGAGTGTGGATACAAAATTGCGCAACAGAGAGCCATCACTCACCACAGCCCGCTGGAATTCTCATGCTGGAGCAA
 Seq 6: CATTGTTGCAGGACCACAGCTTCGAGGTGTGGCAACACCTTTAGTAAACACTTCCTGAATCAGAGACAAGAATGAACC
 Seq 7: ACACATCCGTGTGGCGATTGGGCGGCGTAACCTCGCTTATTTGCATAGGCCGATTGCACAACCGGGCGGCGACCTCAG
 Seq 8: ACACGGCCGATTGCACAACCACCGACCTCAGTGGGAACCGGGCGGGATCCGTGTGGCGATTGAGGCCGATTGCACCTC
 Seq 9: TTAAGAGAATGTCATTGCGGTGTGGCTGAGGGGGAGGGAGAGGTGAGGGTGCAACTTGGGTAAAGGTTGTGGAGGCAT
Seq 10: TACCACTAGCTGCCCTAACTCTTACTAATTAGCGCCAGAGACAAAGTGTGGTTCAAAGATGCAGTGATATATTGCGTC
```

Their locations can *vary* and are *not obvious*…
This makes our jobs of finding them more difficult

Sequence motifs can be difficult to find

```
 Seq 1: TCATTGGTCCTCAGGATCACGCGACAGGAAATGTGGCGTAACCGGTTGACTGCCACATGCGCATTGGCTTCCAGGGCC
 Seq 2: TCATTGATGCGCATTGGCTTCCAGGGGTCCTCAGGATCACACCACAGCCCGCTGGAATAAGTGAGGCGTAAGCCACCC
 Seq 3: CACACCTTTAATTGTTGCAGGATGAATCAGAGGAGGTCTGGCAGTAAACAAGAATGAACCCCACAGCTTCACACTTCC
 Seq 4: TACTGGCGCCAGAGCCAATTTGCGTCATCTAACTAAAGATTTCAACAGCAGTGATATATCTTACTCAAGTGTCGCTAG
 Seq 5: CAAGGAGTGTGGATACAAAATTGCGCAACAGAGAGCCATCACTCACCACAGCCCGCTGGAATTCTCATGCTGGAGCAA
 Seq 6: CATTGTTGCAGGACCACAGCTTCGAGGTGTAGCAACACCTTTAGTAAACACTTCCTGAATCAGAGACAAGAATGAACC
 Seq 7: ACACATCCGTGTGACGATTGGGCGGCGTAACCTCGCTTATTTGCATAGGCCGATTGCACAACCGGGCGGCGACCTCAG
 Seq 8: ACACGGCCGATTGCACAACCACCGACCTCAGTGGGAACCGGGCGGGATCCGTGTGGCGATTGAGGCCGATTGCACCTC
 Seq 9: TTAAGAGAATGTCATTGCGGTGTGGGTGAGGGGAGGGAGAGGTGAGGGTGCAACTTGGGTAAAGGTTGTGGAGGCAT
Seq 10: TACCACTAGCTGCCCTAACTCTTACTAATTAGCGCCAGAGACAAATTGTGGTTCAAAGATGCAGTGATATATTGCGTC
```

The sequences of the motifs can also vary on less important bases.

This makes identifying them bioinformatically even more challenging.

Sequence motifs can be difficult to find

```
 Seq 1: TCATTGGTCCTCAGGATCACGCGACAGGAAATGTGGCGTAACCGGTTGACTGCCACATGCGCATTGGCTTCCAGGGCC
 Seq 2: TCATTGATGCGCATTGGCTTCCAGGGGTCCTCAGGATCACACCACAGCCCGCTGGAATAAGTGAGGCGTAAGCCACCC
 Seq 3: CACACCTTTAATTGTTGCAGGATGAATCAGAGGAGGTCTGGCAGTAAACAAGAATGAACCCCACAGCTTCACACTTCC
 Seq 4: TACTGGCGCCAGAGCCAATTTGCGTCATCTAACTAAAGATTTCAACAGCAGTGATATATCTTACTCAAGTGTCGCTAG
 Seq 5: CAAGGAGTGTGGATACAAAATTGCGCAACAGAGAGCCATCACTCACCACAGCCCGCTGGAATTCTCATGCTGGAGCAA
 Seq 6: CATTGTTGCAGGACCACAGCTTCGAGGTGTAGCAACACCTTTAGTAAACACTTCCTGAATCAGAGACAAGAATGAACC
 Seq 7: ACACATCCGTGTGACGATTGGGCGGCGTAACCTCGCTTATTTGCATAGGCCGATTGCACAACCGGGCGGCGACCTCAG
 Seq 8: ACACGGCCGATTGCACAACCACCGACCTCAGTGGGAACCGGGCGGGATCCGTGTGGCGATTGAGGCCGATTGCACCTC
 Seq 9: TTAAGAGAATGTCATTGCGGTGTGGGTGAGGGGGAGGGAGAGGTGAGGGTGCAACTTGGGTAAAGGTTGTGGAGGCAT
Seq 10: TACCACTAGCTGCCCTAACTCTTACTAATTAGCGCCAGAGACAAATTGTGGTTCAAAGATGCAGTGATATATTGCGTC
```

We can describe a motif ***qualitatively***…

(Only *presence* versus *absence* of letters at this point…
no numerical information)

Discrete representations of motifs

- Consensus sequences
- Degenerate consensus sequences
- Rule / regular expressions

We can describe a motif ***qualitatively***…

(Only *presence* versus *absence* of letters at this point…
no numerical information)

Consensus sequences

A simple way of representing a motif, is by using a **consensus sequence**, or the most common letter at each position

However, this **does not allow for variability** in identifying positions where there are multiple options for the motif

Allowing for **degeneracy** in the consensus sequence does allow for some additional flexibility…

Alignment:

```
AATGCGGA
AATGTGGC
ACTGTGGC
CGTGTGGC
CGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGG
GGTGTGGG
```

Consensus: **GG**TG**T**GG**C**

Degenerate consensus sequences

This is still limiting, however, as there is no quantitative measure for variability at the different positions

R	A | G	puRine
Y	C | T	pYrimidine
S	G | C	Weak (weaker basepairs, fewer hydrogen bonds)
W	A | T	Strong (stronger basepairs, more hydrogen bonds)
K	G | T	Keto (both have a keto group)
M	A | C	aMine (both have an amine group)
B	C | G | T	not A (B comes after A)
D	A | G | T	not C (D comes after C)
H	A | C | T	not G (H comes after G)
V	A | C | G	not T or U (V comes after T and U)
N	A | C | G | T	aNy base

Alignment:

```
AATGCGGA
AATGTGGC
ACTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGA
GGTGTGGA
```

Consensus: GGTGTGGC

Degenerate Consensus: RVTGYGGM

Describing motifs using regular expressions

We can also describe a motif using a ***rule*** or ***regular expression…***

A – matches A
[AT] – matches A or T
{ AT } – matches neither A or T (i.e. G or C)
x – matches any symbol
x(3) – any 3 symbols

```
AATGCGGA
AATGTGGC
ACTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGA
GGTGTGGA
```

[AG]{T}TG[CT]GG[AC]

Describing motifs using regular expressions

We can also describe a motif using a ***rule*** or ***regular expression…***

A – matches A
[AT] – matches A or T
{ AT } – matches neither A or T (i.e. G or C)
x – matches any symbol
x(3) – any 3 symbols

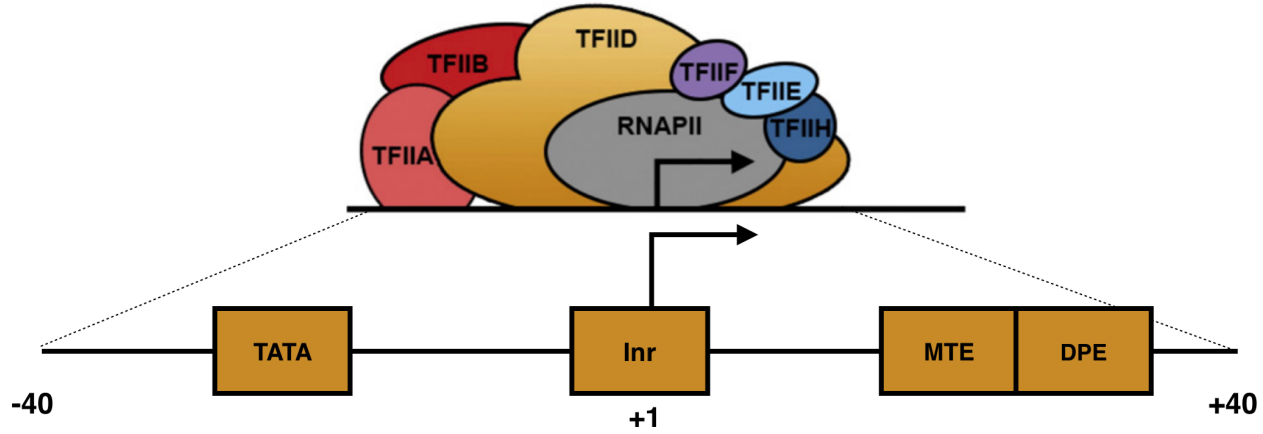
```
LVIEMLY
LVIECLY
LVIECLF
LVIEMLF
LVIEMLF
LVIEMLF
LVIEMLF
LVLEMLF
LVVEMLF
LVIEMLY
```

Like before, we are still limited in how much information this conveys. However, this can be informative in certain cases…

LV[ILV]E[MC]L[FY]

DNA: Transcription factor DNA-binding motifs

- Inr – Initiator Element
- MTE – Motif Ten Element
- DPE – Downstream Core Promoter Element



- MTE promotes transcription by RNA polymerase II
- MTE requires Inr, but independent of TATA and DPE
- MTE can compensate for mutations in TATA and DPE

Sequence Element	Approximate Position	Consensus Sequence
TATA Box	-30 to -23	TATAWAW
Inr	overlaps the transcription start site (+1)	BBCA$_{(+1)}$BW
MTE	+18 to +29	CSARCSSAACGS
DPE	+28 to +33	DSWYVY

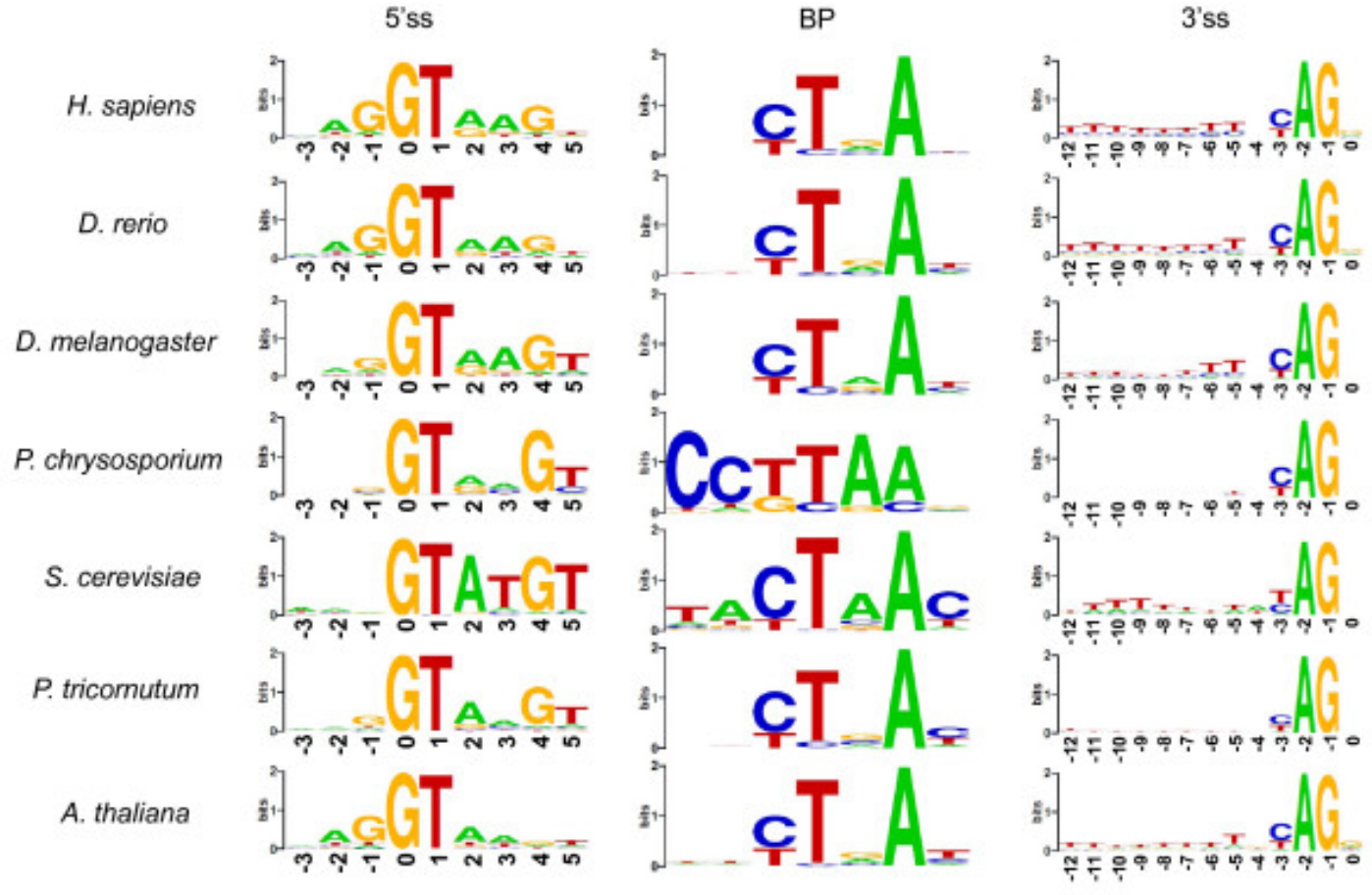
Nucleotide positions (column 2) are all relative to the A (+1) of the Inr.
IUPAC codes: D=A/G/T, S=G/C, R=A/G, V=A/C/G, W=A/T, Y=C/T, B= C/G/T

Iwata and Gotoh et al. 2011 in *BMC Genomics* 12:45

RNA: mRNA splice site motifs

Motifs for RNA splicing are generally very conserved across eukaryotes

If you're curious about exemptions to this.. read about dinoflagellate splice sites



Iwata and Gotoh et al. 2011 in *BMC Genomics* 12:45

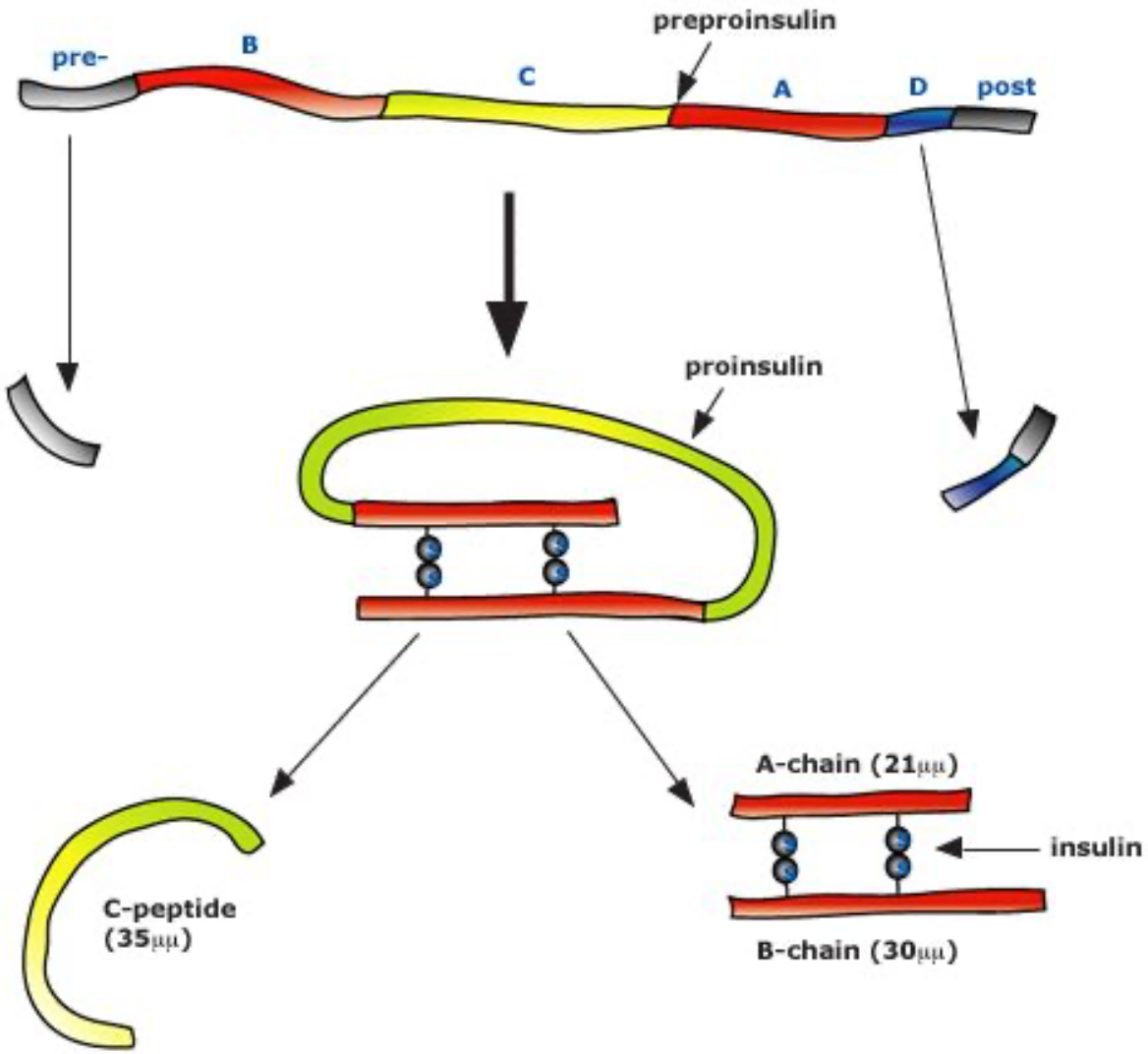
Protein: the insulin motif

Most of the insulin peptide gene sequence is not present in the final peptide product.

Even of the lingering sequence, only 8 cysteine residues are highly conserved as they are the location of the disulfide bonds that link the two chains

Distribution in cysteine residues is how they are classified in invertebrates

Chain A motif: x(5)-CC-x(3)-C-x(8)-C-x
Chain B motif: x(6)-C-x(11)-C-x(13)



Reflection

- *What are sequence motifs?*
- *What aspects of sequence motifs make them difficult to accurately locate and describe?*
- *What are the limitations of a discrete representation for a motif and why?*
- *How are exact consensus sequences, degenerate consensus sequences and regular expressions alike and dissimilar?*

Sequence Analysis 3B:

Sequence motif profiles

Katherine Dougan, PhD

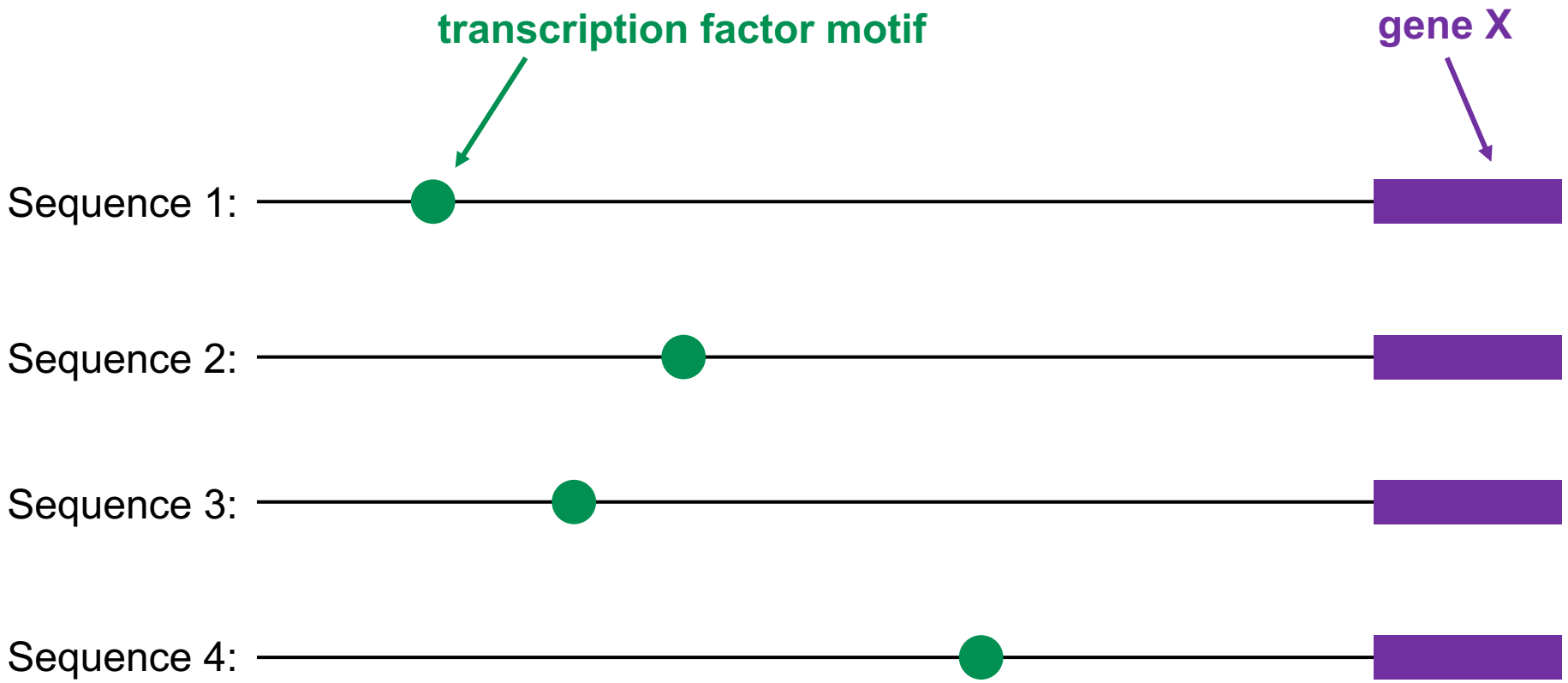
Australian Centre for Ecogenomics
School of Chemistry & Molecular Biosciences
The University of Queensland

SCIE2100 | BINF6000 | Bioinformatics I - Introduction

Outline

- **Identifying motifs in a sequence**
  - Position Specific Scoring Matrix
  - Scoring sequences with profiles
- **Constructing motif profiles**
  - Position Frequency Matrix
  - Position Probability Matrix
  - Position Weight Matrix
  - Finding the most likely start
- **Visualizing motifs with sequence logos**
  - Representing uncertainty with entropy
  - Determining Information Content

Identifying motifs in a sequence



Let's say we want to find a transcription factor motif for gene X, but its location and sequence is highly variable... ***how could we identify the motif better accounting for variability?***

Position Specific Scoring Matrix

We can use a numeric representation of a motif called a ***profile*** to identify the best starting position by calculating a similarity score between the profile and a sequence

For example, we could describe motif M of length K by the probability of encountering different nucleotides at each position of the sequence in profile f …

$$f = \begin{bmatrix} f_{1A} & f_{2A} & f_{3A} & f_{4A} & \cdots & f_{KA} \\ f_{1C} & f_{2C} & f_{3C} & f_{4C} & \cdots & f_{KC} \\ f_{1G} & f_{2G} & f_{3G} & f_{4G} & \cdots & f_{KG} \\ f_{1T} & f_{2T} & f_{3T} & f_{4T} & \cdots & f_{KT} \end{bmatrix}$$

and calculate the ***probability*** of a sequence x being an instance of motif M:

$$P(x|M) = \prod_{i=1}^{K} f_{i,x}$$

Scoring a sequence using a profile

The profile f below was generated from an alignment of motif M to the right

$$P(x|M) = \prod_{i=1}^{K} f_{i,x}$$

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

Alignment:

AATGCGGA
AATGTGGC
ACTGTGGC
CGTGTGGC
CGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGG
GGTGTGGG

What is the probability of Sequence X being an instance of M?

Sequence X: CCTGCGGC

$$P(X|M) = 0.21 * 0.14 * 0.79 * 0.07 * 0.14 * 0.79 * 0.79 * 0.57 = 0.000081$$

Usually if $P(x|M) \geq threshold$ when compared to the highest possible likelihood (i.e. consensus sequence) then it is a match

What if we do not know where the motif starts?

Finding the most likely start for a motif

We can use a sliding window and calculate the score for each set to find the one with the greatest score

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

S(1) ATGCGATGACCTGC

S(2) ATGCGATGACCTGC

S(3) ATGCGATGACCTGC

$$P(X|M) = f_{1x} * f_{2x} * f_{3x} * f_{4x} * f_{5x} * f_{6x} * f_{7x} * f_{8x}$$

Let's get into how you construct a Position Specific Scoring Matrix…

Finding the most likely start for a motif

We can use a sliding window and calculate the score for each set to find the one with the greatest score

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

S(1) ATGCGATGACCTGC

S(2) ATGCGATGACCTGC

S(3) ATGCGATGACCTGC

$$P(X|M) = f_{1x} * f_{2x} * f_{3x} * f_{4x} * f_{5x} * f_{6x} * f_{7x} * f_{8x}$$

Let's get into how you construct a Position Specific Scoring Matrix…

Finding the most likely start for a motif

We can use a sliding window and calculate the score for each set to find the one with the greatest score

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

S(1) ATGCGATGACCTGC

S(2) ATGCGATGACCTGC

S(3) ATGCGATGACCTGC

$$P(X|M) = f_{1x} * f_{2x} * f_{3x} * f_{4x} * f_{5x} * f_{6x} * f_{7x} * f_{8x}$$

Let's get into how you construct a Position Specific Scoring Matrix…

The different kinds of PSSMs or profile matrices

Let's first discuss the different kinds of profile matrices:

Position Frequency Matrix (PFM) – the position-dependent frequency f or how often each letter (i.e. nucleotide or amino acid) occurs at a given position in N sequences

Position Probability Matrix (PPM) – the probability of each letter at a given position by normalizing the PFM values by the total number of sequences, N

Position Weight Matrix (PWM) – the log likelihood ratios of the PPM

PSSM – Step 1: Position Frequency Matrix (PFM)

$n_{i,a}$ is the count of residue a in column i

PFM

$n_{i,a}$

A	3	2	0	0	0	0	0	1
C	2	1	0	0	1	0	0	7
G	5	7	0	0	0	10	10	2
T	0	0	10	10	9	0	0	0

```
AATGCGGA
AATGTGGC
ACTGTGGC
CGTGTGGC
CGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGC
GGTGTGGG
GGTGTGGG
```

PSSM – Step 2: Add pseudocounts to PFM

$n_{i,a}$ is the count of residue a in column i

PFM

$n_{i,a}$

A	3	2	0	0	0	0	0	1
C	2	1	0	0	1	0	0	7
G	5	7	0	0	0	10	10	2
T	0	0	10	10	9	0	0	0

If we have zeros in our profile, then we need to add **pseudocounts**.

Later on, we will be doing log calculations, and you can't calculate the log of 0…

Pseudocounts

(if zeros in PFM)

A	4	3	1	1	1	1	1	2
C	3	2	1	1	2	1	1	8
G	6	8	1	1	1	11	11	3
T	1	1	11	11	10	1	1	1

And it lets us account for scenarios not currently captured in our model by representing them at very low frequencies

PSSM – Step 3: Position Probability Matrix (PPM)

$n_{i,a}$ is the count of residue a in column i

Pseudocounts
(if zeros in PFM)
$n_{i,a}$

A	4	3	1	1	1	1	1	2
C	3	2	1	1	2	1	1	8
G	6	8	1	1	1	11	11	3
T	1	1	11	11	10	1	1	1

PPM

$$f_{i,a} = \frac{n_{i,a}}{N_{seq}}$$

A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

Using our PFM that has been adjusted with pseudocounts, we now calculate the probability of each character at each position

PSSM – Step 4: Adjust PPM by background probability

$n_{i,a}$ is the count of residue a in column i

PPM

$$f_{i,a} = \frac{n_{i,a}}{N_{seq}}$$

A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

Human p_a

```
A: 0.233
C: 0.268
G: 0.267
T: 0.231
```

$$\frac{f_{i,a}}{p_a}$$

A	1.24	0.90	0.30	0.30	0.30	0.30	0.30	0.60
C	0.78	0.52	0.26	0.26	0.52	0.26	0.26	2.13
G	1.61	2.13	0.26	0.26	0.26	2.96	2.96	0.79
T	0.30	0.30	3.42	3.42	3.07	0.30	0.30	0.30

Some organisms do not have the four bases occurring at similar frequencies.

For example, GC content is 38% in yeasts and 19% in Plasmodium falciparum.

PSSM – Step 5: Position Weight Matrix (PWM)

$n_{i,a}$ is the count of residue a in column i

PPM

$$\frac{f_{i,a}}{p_a}$$

A	1.24	0.90	0.30	0.30	0.30	0.30	0.30	0.60
C	0.78	0.52	0.26	0.26	0.52	0.26	0.26	2.13
G	1.61	2.13	0.26	0.26	0.26	2.96	2.96	0.79
T	0.30	0.30	3.42	3.42	3.07	0.30	0.30	0.30

PWM

$$W_{i,a} = \log_2\left(\frac{f_{i,a}}{p_a}\right)$$

A	0.32	-0.15	-1.73	-1.73	-1.73	-1.73	-1.73	-0.73
C	-0.35	-0.94	-1.94	-1.94	-0.94	-1.94	-1.94	1.09
G	0.69	1.09	-1.93	-1.93	-1.93	1.57	1.57	-0.35
T	-1.72	-1.72	1.77	1.77	1.62	-1.72	-1.72	-1.72

Finally, we can calculate the log-likelihood ratios

Finding the most likely start for a motif

Going back to our original example…

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

ATGCGATGACCTGCGGC

ATGCGATGACCTGCGGC

ATGCGATGACCTGCGGC

Alignment:

AATGCGGA

AATGTGGC

ACTGTGGC

CGTGTGGC

CGTGTGGC

GGTGTGGC

GGTGTGGC

GGTGTGGC

GGTGTGGG

GGTGTGGG

Here we used the **Position Probability Matrix**, and calculated the **PRODUCT** of the values with:

$$P(x|M) = \prod_{i=1}^{K} f_{i,x}$$

Finding the most likely start for a motif

We can also use the *Position Weight Matrix* for this…

	1	2	3	4	5	6	7	8
A	0.32	-0.15	-1.73	-1.73	-1.73	-1.73	-1.73	-0.73
C	-0.35	-0.94	-1.94	-1.94	-0.94	-1.94	-1.94	1.09
G	0.69	1.09	-1.93	-1.93	-1.93	1.57	1.57	-0.35
T	-1.72	-1.72	1.77	1.77	1.62	-1.72	-1.72	-1.72

ATGCGATGACCTGCGGC

ATGCGATGACCTGCGGC

ATGCGATGACCTGCGGC

Alignment:

AATGCGGA

AATGTGGC

ACTGTGGC

CGTGTGGC

CGTGTGGC

GGTGTGGC

GGTGTGGC

GGTGTGGC

GGTGTGGG

GGTGTGGG

If we use the **Position Weight Matrix**, then we instead calculate the **SUMS** of the values with:

$$P(x|M) = \sum_{i=1}^{K} w_{i,x}$$

Finding the most likely start for a motif

When comparing a sequence to the profile for a motif…

If we use the **Position Probability Matrix**, then we calculate the **PRODUCT** of the values with:

$$P(x|M) = \prod_{i=1}^{K} f_{i,x}$$

If we use the **Position Weight Matrix**, then we instead calculate the **SUM** of the values with:

$$P(x|M) = \sum_{i=1}^{K} w_{i,x}$$

Finding the most likely start for a motif

We can also use the *Position Weight Matrix* for this…

	1	2	3	4	5	6	7	8
A	0.32	-0.15	-1.73	-1.73	-1.73	-1.73	-1.73	-0.73
C	-0.35	-0.94	-1.94	-1.94	-0.94	-1.94	-1.94	1.09
G	0.69	1.09	-1.93	-1.93	-1.93	1.57	1.57	-0.35
T	-1.72	-1.72	1.77	1.77	1.62	-1.72	-1.72	-1.72

$$S(x|M) = \sum_{i=1}^{K} w_{i,x}$$

S(1)= GATGACCTGCGGC = -6.95
S(2)= GATGACCTGCGGC = -10.01
S(3)= GATGACCTGCGGC = -4.3
S(4)= GATGACCTGCGGC = -2.44
S(5)= GATGACCTGCGGC = -3.44
S(6)= GATGACCTGCGGC = 1.84

The last option is the most likely start for the motif

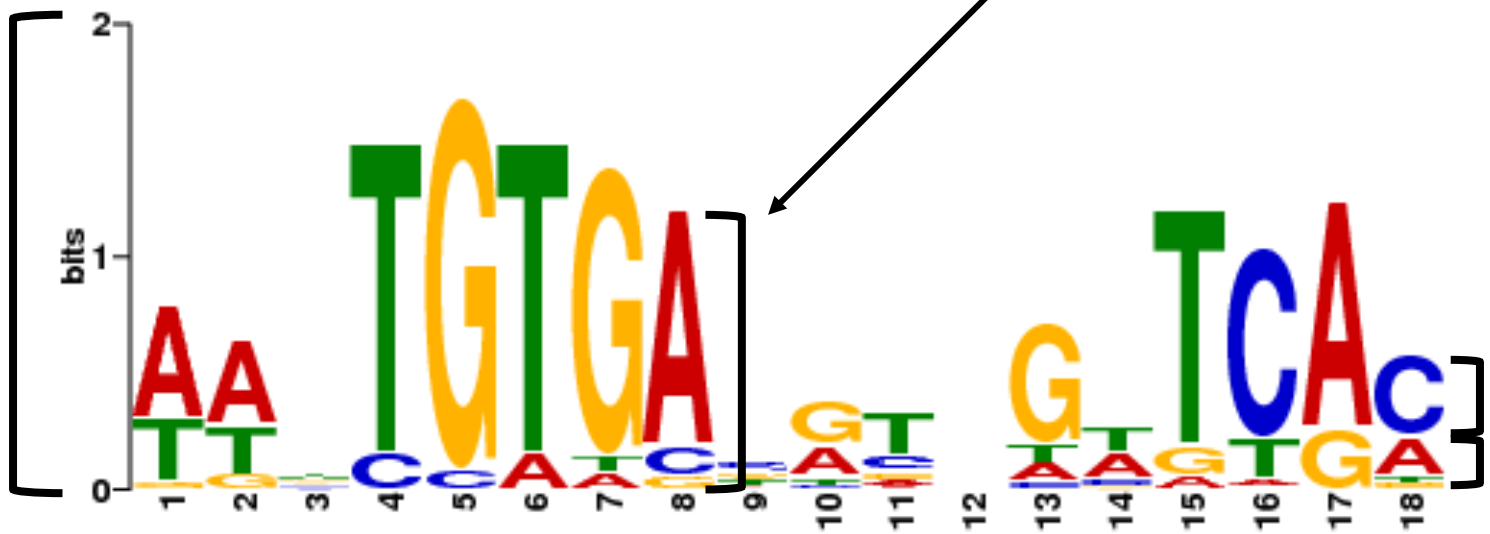
Visualizing motifs with Sequence Logos

Sequence logos are a great way to visualize not only the probability of different characters at each position in a motif, but also the ***uncertainty*** in the motif and the ***information content*** for each position

Entropy is represented by the total height of a column in *bits.*

The total height of characters in a column is the **Information Content.**

The height of an individual character is the **frequency of the residue.**



Let's explore these parameters and how they're calculated…

Representing *uncertainty* in our model with Entropy

Problem: Different positions in a motif will exhibit ***varying levels of conservation.***

We can use **Shannon entropy** to represent uncertainty, or how unpredictable sequences generated from the profile can be.

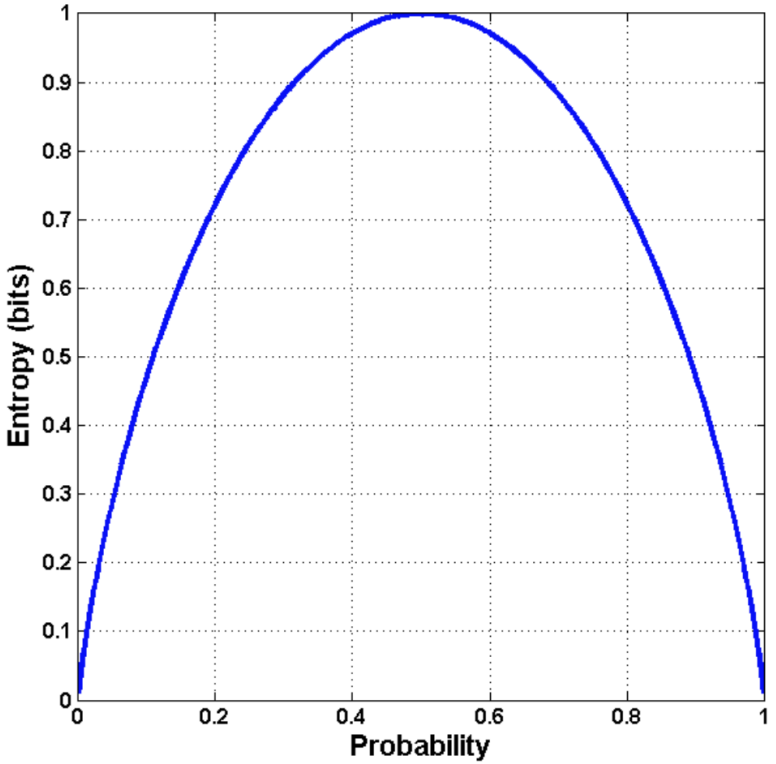
$$H_i = -\sum f_{i,a} \log_2 f_{i,a}$$

$$H = -\left(\frac{1}{100}\log_2\left(\frac{1}{100}\right) + \frac{1}{100}\log_2\left(\frac{1}{100}\right) + \frac{1}{100}\log_2\left(\frac{1}{100}\right) + \frac{97}{100}\log_2\left(\frac{97}{100}\right)\right)$$

$$\boldsymbol{H = 0.25}$$

$$H = -\left(\frac{1}{10}\log_2\left(\frac{1}{10}\right) + \frac{1}{10}\log_2\left(\frac{1}{10}\right) + \frac{1}{10}\log_2\left(\frac{1}{10}\right) + \frac{7}{10}\log_2\left(\frac{7}{10}\right)\right)$$

$$\boldsymbol{H = 1.36}$$



The less equal the probability of different outcomes are , the lower the entropy

Representing *uncertainty* in our model with Entropy

Problem: Different positions in a motif will exhibit ***varying levels of conservation.***

We can use **Shannon entropy** to represent uncertainty, or how unpredictable sequences generated from the profile can be.

$$H_i = -\sum f_{i,a} \log_2 f_{i,a}$$

If each nucleotide is equally likely, then that is the scenario of the highest uncertainty…

$f_{i,a} = \frac{1}{4}$ for all $a \in \{A, C, G, T\}$

$$H_{max} = -\left(\frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right)$$

$\log_2 \frac{1}{4} = -2$

$\boldsymbol{H_{max} = 2\ bits}$ ⟵ The maximum information content of any position is 2 bits

Determining the *Information Content* of a site

The **Information Content** (I) at each site is the ***reduction in entropy***, or how much it reduces uncertainty compared to the background model.

The difference in what we know now compared to what we knew before

$$H_i = -\sum f_{i,a} \log_2 f_{i,a}$$

$$I_i = H_{max} - H_i$$

	1	2	3	4	5	6	7	8
A	0.29	0.21	0.07	0.07	0.07	0.07	0.07	0.14
C	0.21	0.14	0.07	0.07	0.14	0.07	0.07	0.57
G	0.43	0.57	0.07	0.07	0.07	0.79	0.79	0.21
T	0.07	0.07	0.79	0.79	0.71	0.07	0.07	0.07

Let's calculate the *entropy* and *information content* for position 3 in our PPM…

$$H_{max} = -(0.233 * \log_2(0.233) + 0.268 * \log_2(0.268) + 0.267 * \log_2(0.267) + 0.231 \log_2(0.231))$$

$$H_3 = -(0.07 * \log_2(0.07) + 0.07 * \log_2(0.07) + 0.07 * \log_2(0.07) + 0.79 \log_2(0.79))$$

$$I_3 = H_{max} - H_3 = 1.996 - 1.074 = 0.992$$

```
A: 0.233
C: 0.268
G: 0.267
T: 0.231
```

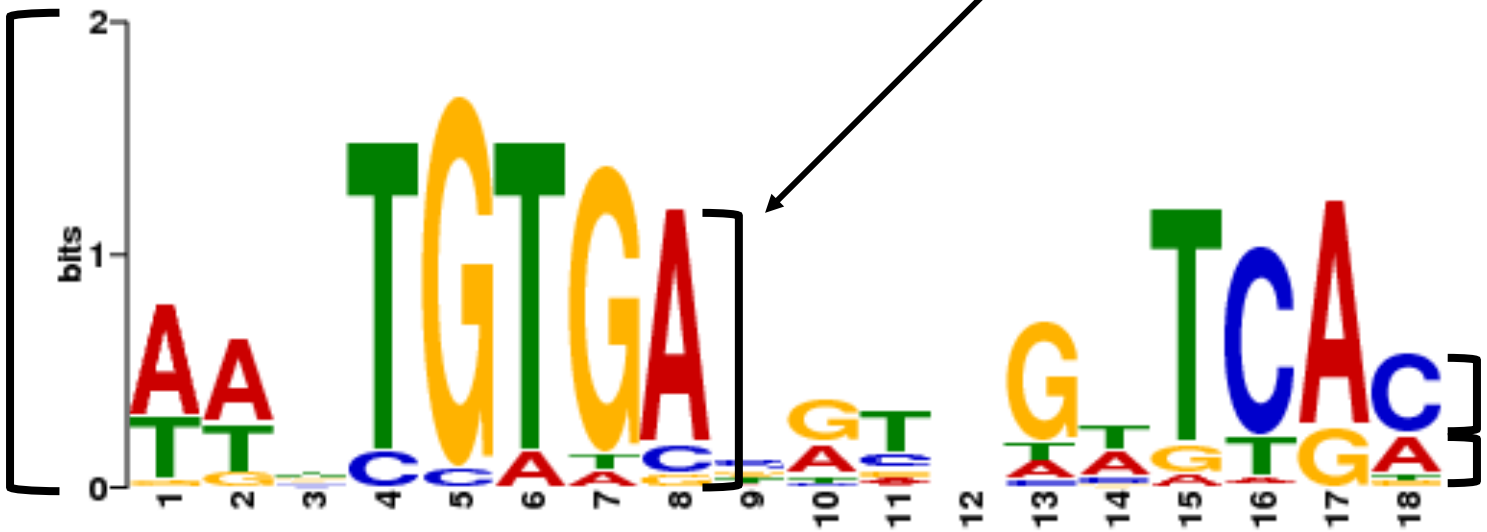
Visualizing motifs with Sequence Logos

Sequence logos are a great way to visualize not only the probability of different characters at each position in a motif, but also the ***uncertainty*** in the motif and the ***information content*** for each position

Entropy is represented by the total height of a column in *bits.*

The total height of characters in a column is the **Information Content.**

The height of an individual character is the **frequency of the residue.**



Reflection

- *What are the strengths of profiles in describing motifs compared to discrete representations?*
- *What two types of PSSM can you apply to a sequence to determine if it is an instance of a motif and how is it calculated?*
- *What are the calculations and steps involved in converting a sequence alignment to a Position Weight Matrix?*
- *How can you represent uncertainty in a profile?*
- *What is the Information Content of a profile and how is it calculated?*
- *What is a sequence logo and what parameters are needed to construct one?*