

---

# **INFBIOrx121 - Algorithms module**

## Fastq and alignment

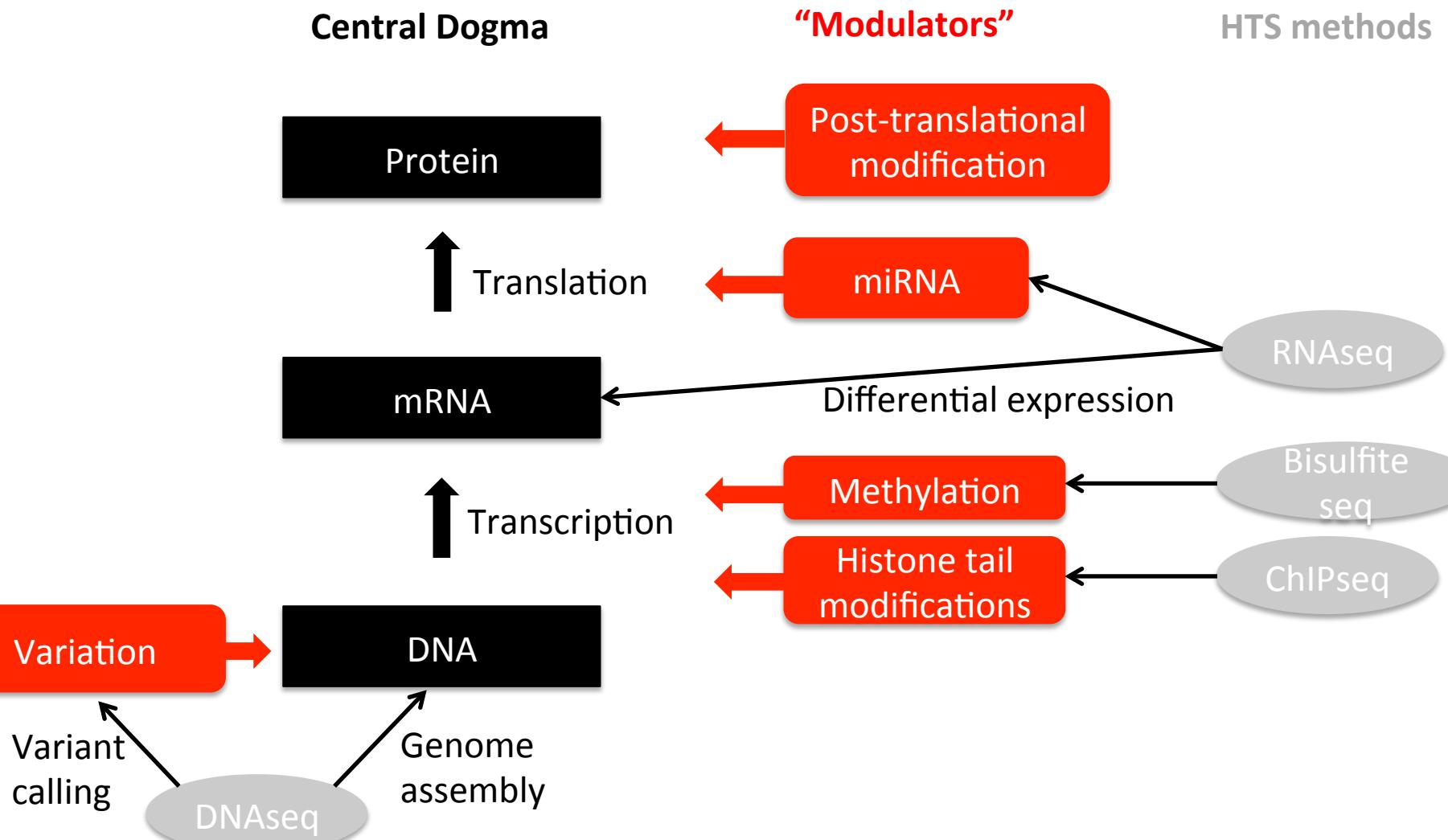
Autumn 2017



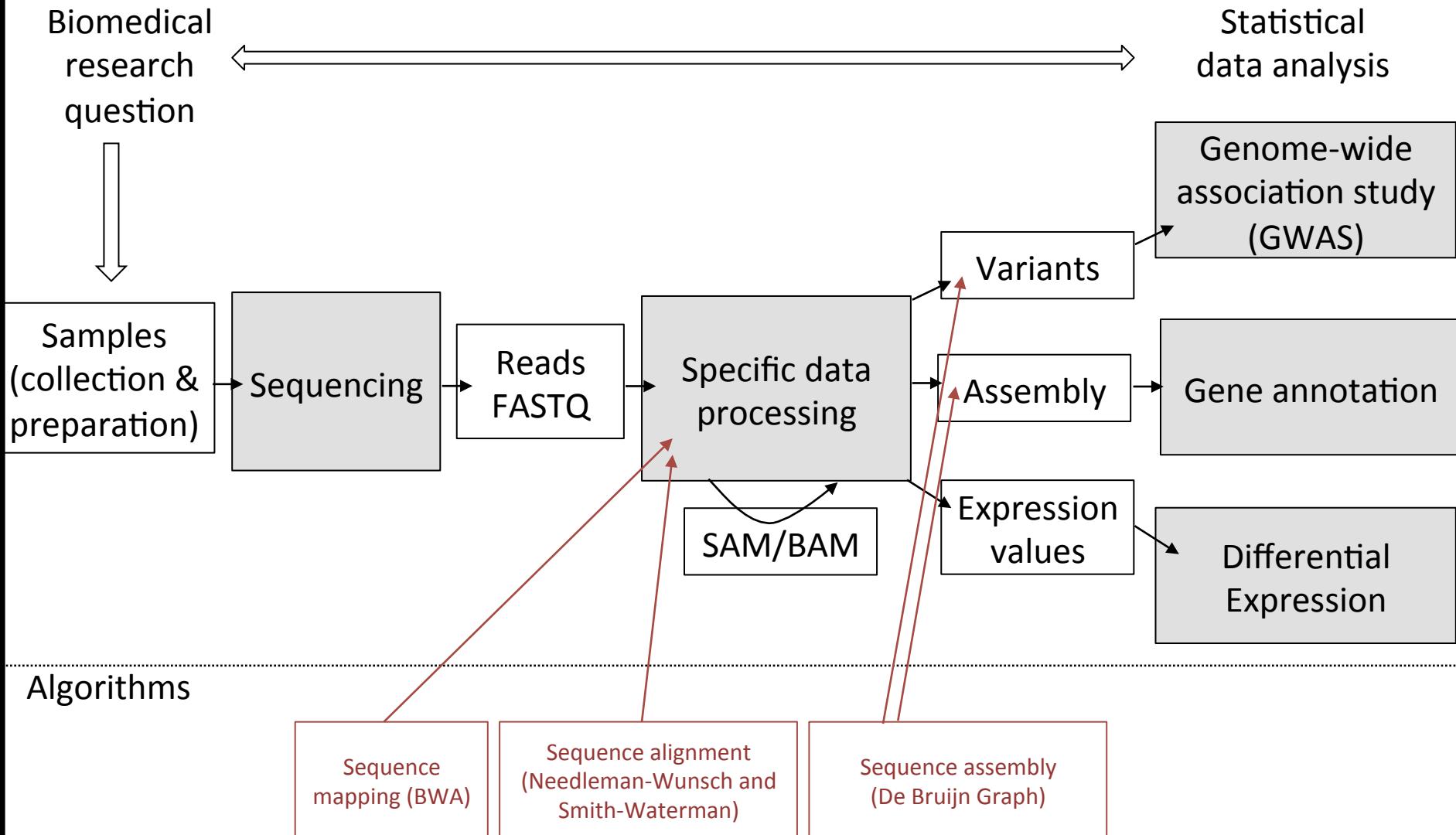
NORWEGIAN SEQUENCING CENTRE



# Central dogma and HTS



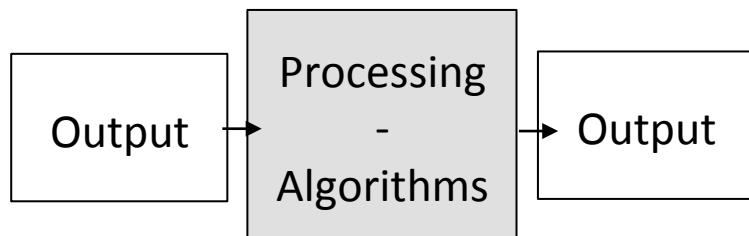
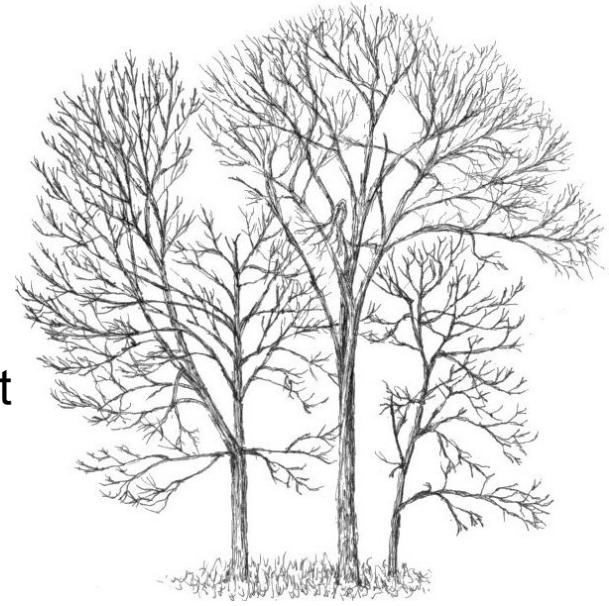
# Key algorithms in the big picture?



# Good practices for computational data processing and analysis

---

- Make sure you always know where you are!
- Be careful and structured where you put things:
  - file naming
  - directory structure
- Checking that the computer has done what you expect it to do, verifying outputs:
  - file size
  - timestamps
  - file contents
- A computer is a very complex device.
  - You don't need to understand all the details all the way down the stack
  - But you do need to understand what is happening at the level that is relevant for your work.

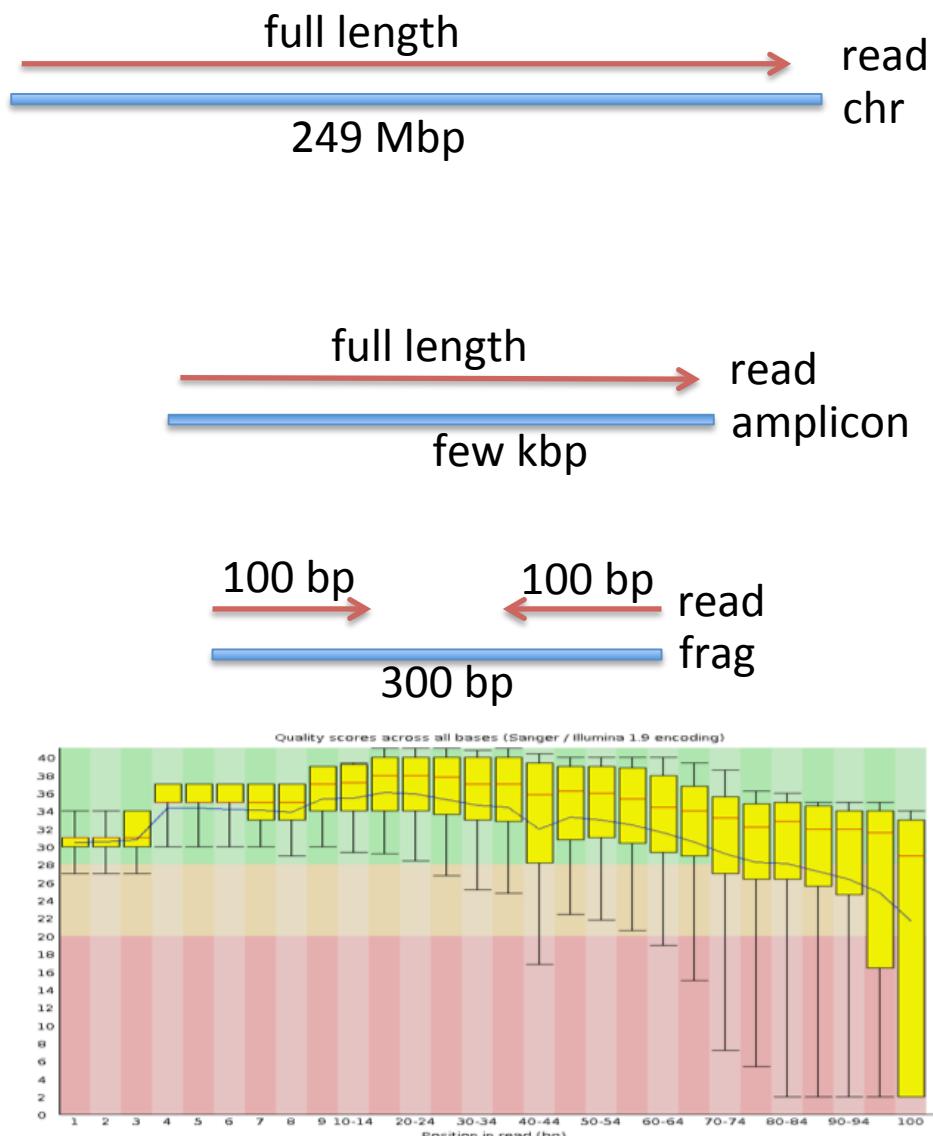


---

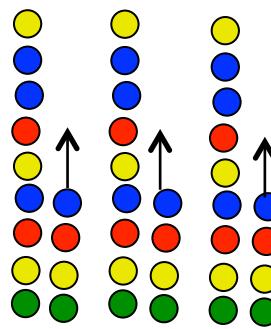
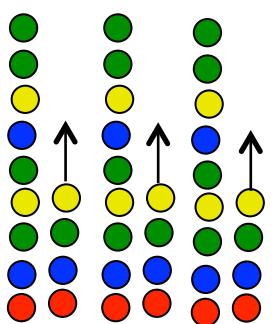
# **FASTQ: ERROR PROBABILITY, QUALITY SCORES AND ENCODINGS**

# In a perfect world – Perfect sequencing

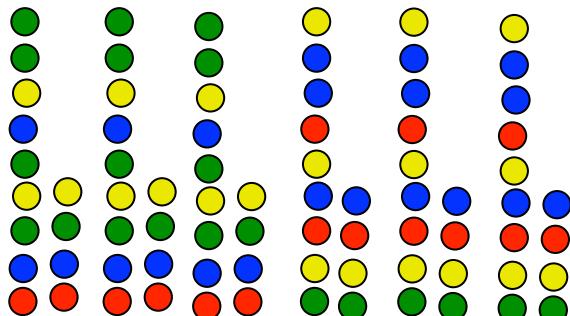
- Perfect sequencing:
  - single molecule (no PCR)
  - **full** length
  - no deterioration of quality
- While we are waiting:
  - Sanger
    - PCR
    - length: some kb
    - limited number of reads
    - high quality
  - HTS (Illumina)
    - PCR
    - 100 bp PE
    - billions of reads
    - high quality, but deteriorating along read



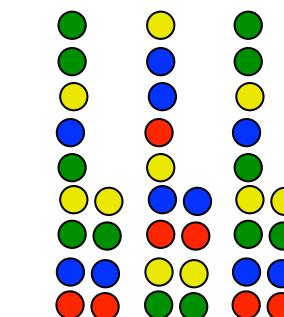
# Explaining how/why sequence quality varies



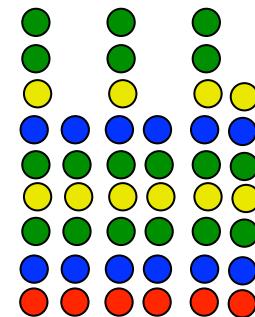
Normal situation: Well defined “pure” clusters



Neighbouring clusters



“Mixed” cluster



Unphased



The identity of the base in a given cluster  
in a given cycle is not known with certainty

# Fastq format – fasta with qualities

```
@J00146:31:HJF5NBBXX:7:1101:2483:1121 1:N:0:NTTCAGAA+NTTCGCCT
NTTGTGAGGGAAAGGATTAGGAAGTTGAGTGTCCTATTGAGTTGGATTGAAATGAGGGCAATTAAGAGTGGGA'
+
#AAAFAFJJJJJA-JJAFJJ<7FJJJ---7-<F-7FA-<FJJ-<<FJ-F-AA7J-FFJ7A<JJJ--7FAJF7A<<A
```

- $p$  = the probability that the corresponding base call is wrong
- Qualities  $Q_{\text{sanger}} = -10 \log_{10} p$
- Rule of thumb
  - $p = 0.1 \rightarrow Q = 10$
  - $p = 0.01 \rightarrow Q = 20$
  - $P = 0.001 \rightarrow Q = 30$

# Quality scores

---

- Why use a quality score
  - More intuitive that a “better” characteristic gets a higher score
  - More usable representation: integer number of 1 or 2 digits rather than a decimal number
- Where are quality scores used for:
  - base quality scores
  - mapping quality scores
  - variant quality scores

# ASCII conversion

@J00146:31:HJF~~5~~NBBXX:7:1~~1~~01:2483:1121 1:N:0:NTTCAGAA+NTTCGCCT  
NTTGTGAGGGAAAGGATTAGGAAGTTGAGTGTCCTATTGAGTTGGATTGAAATGAGGGCAATTAAGAGTGGGA'  
+  
#AAAFAFJJJJJA-JJAFJJ<7FJJJ---7-<F-7FA-<FJJ-<<FJ-F-AA7J-FFJ7A<JJJ--7FAJF7A<<A

↑      ↑

ASCII encoding: Sanger/Phred format can encode a quality score from 0 to 93 using ASCII 33 to 126: Q + 33

## Exercise:

- Why do you think we add 33?
- What would be the ASCII code for a base that had a probability of error of 0.15
- What is the probability of error of the first base in the above sequence?
- What is the probability of error of the 10<sup>th</sup> base in the above sequence?

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
32	20	040	&#32;	Space	64	40	100	&#64;	@
33	21	041	&#33;	!	65	41	101	&#65;	A
34	22	042	&#34;	"	66	42	102	&#66;	B
35	23	043	&#35;	#	67	43	103	&#67;	C
36	24	044	&#36;	\$	68	44	104	&#68;	D
37	25	045	&#37;	%	69	45	105	&#69;	E
38	26	046	&#38;	&	70	46	106	&#70;	F
39	27	047	&#39;	'	71	47	107	&#71;	G
40	28	050	&#40;	(	72	48	110	&#72;	H
41	29	051	&#41;	)	73	49	111	&#73;	I
42	2A	052	&#42;	*	74	4A	112	&#74;	J
43	2B	053	&#43;	+	75	4B	113	&#75;	K
44	2C	054	&#44;	,	76	4C	114	&#76;	L
45	2D	055	&#45;	-	77	4D	115	&#77;	M
46	2E	056	&#46;	.	78	4E	116	&#78;	N
47	2F	057	&#47;	/	79	4F	117	&#79;	O
48	30	060	&#48;	0	80	50	120	&#80;	P
49	31	061	&#49;	1	81	51	121	&#81;	Q
50	32	062	&#50;	2	82	52	122	&#82;	R
51	33	063	&#51;	3	83	53	123	&#83;	S
52	34	064	&#52;	4	84	54	124	&#84;	T
53	35	065	&#53;	5	85	55	125	&#85;	U
54	36	066	&#54;	6	86	56	126	&#86;	V
55	37	067	&#55;	7	87	57	127	&#87;	W
56	38	070	&#56;	8	88	58	130	&#88;	X
57	39	071	&#57;	9	89	59	131	&#89;	Y
58	3A	072	&#58;	:	90	5A	132	&#90;	Z
59	3B	073	&#59;	:	91	5B	133	&#91;	[
60	3C	074	&#60;	<	92	5C	134	&#92;	\
61	3D	075	&#61;	=	93	5D	135	&#93;	]
62	3E	076	&#62;	>	94	5E	136	&#94;	^
63	3F	077	&#63;	?	95	5F	137	&#95;	-

# Different quality formulas and encodings

- Beware of different versions

- With sequence data recently produced, you do not need to worry: **Phred + 33**
  - For older data, be careful

# Illumina sequence identifiers

```
@SEQ_ID
GATTTGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTGTTCAACTCACAGTTT
+
! ' * ( ( ( ****+ ) %%++ ) %%%.1***-+* ' ) **55CCF>>>>CCCCCCC65
```

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

<b>EAS139</b>	the unique instrument name
<b>136</b>	the run id
<b>FC706VJ</b>	the flowcell id
<b>2</b>	flowcell lane
<b>2104</b>	tile number within the flowcell lane
<b>15343</b>	'x'-coordinate of the cluster within the tile
<b>197393</b>	'y'-coordinate of the cluster within the tile
<b>1</b>	the member of a pair, 1 or 2 ( <i>paired-end or mate-pair reads only</i> )
<b>Y</b>	Y if the read is filtered, N otherwise
<b>18</b>	0 when none of the control bits are on, otherwise it is an even number
<b>ATCACG</b>	index sequence

---

# **SEQUENCE MAPPING**

# Mapping and alignment

---

1                  10                  20                  30                  40                  50  
AGCTGTAGTAGCTGTAGCGTAGTTGCCGTACGTCTATGGTGAGGAGAGAGTTCCCCTA

**Human genome is 3G bases spread over 23 chromosomes**

1. Can you locate GTTGCCGTA?

**>> sequence mapping**

2. Assume that you now have a slightly different sequence GTTGCCTT,  
how does this compare to the sequence in 1?

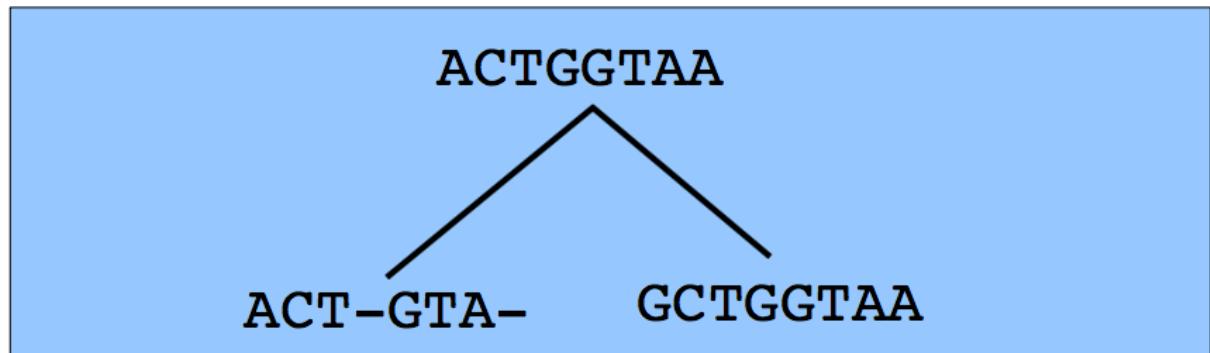
**>> sequence alignment**

---

# **SEQUENCE ALIGNMENT**

# Sequence divergence

Evolution



Observation

ACTGTA  
GCTGGTAA

Goal

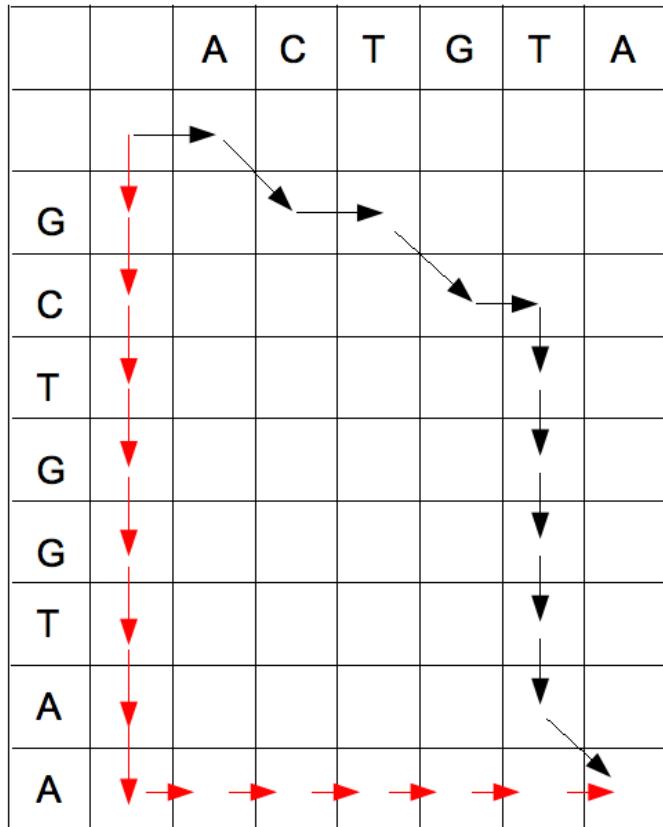
ACT-GTA-  
GCTGGTAA

# What is a global alignment?

---

- A global alignment of 2 sequences q and d must satisfy:
  - All symbols in q and d have to be in the alignment, and **in the same order as they appear in q and d**
  - We can align one symbol from q with one from d
  - A symbol can be aligned with a blank (gap)
  - Two blanks cannot be aligned
- In the case of DNA, the symbols are A, G, C, and T

# A global alignment is a path in a matrix



ACTGT-----A  
-G-C-TGGTAA

-----ACTGTA  
GCTGGTAA-----

There are a large number of paths through the matrix

We need:

- a way of scoring different paths
- a way of finding the best scoring path

**Important:** Paths through the matrix have to incorporate a new base from at least one of the sequences ie arrows moving “back up” the matrix do not correspond to an alignment.

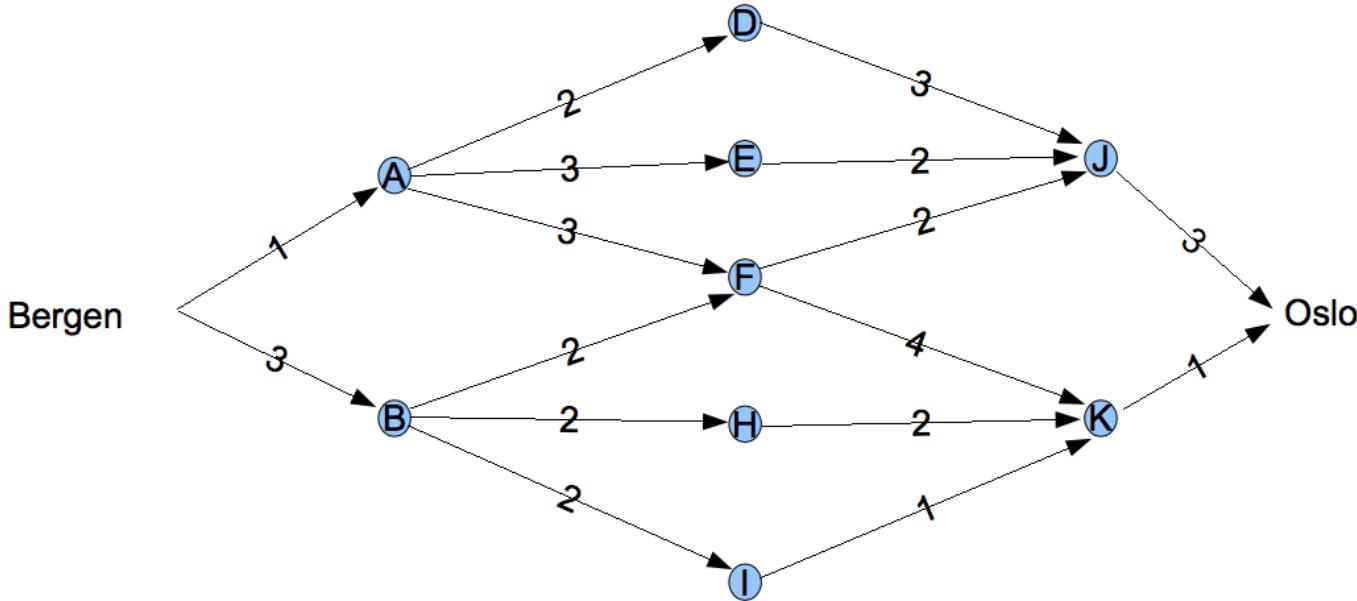
# Scoring schemes

---

- A scoring scheme
  - each column can be given a score, **independently** of the other columns, meaning that mutations are single mutations
    - need score for alignment of two residues
    - need scoring for alignment of residue with gap
  - The score of the alignment can be found as the sum of the score of all columns
- Scoring scheme is **critical** to the alignment produced
- Computation time **intensive** to enumerate all alignments and score them all to find the best scoring
- **Dynamic programming** to find the best scoring alignments  
(Needleman and Wunsch 1970)

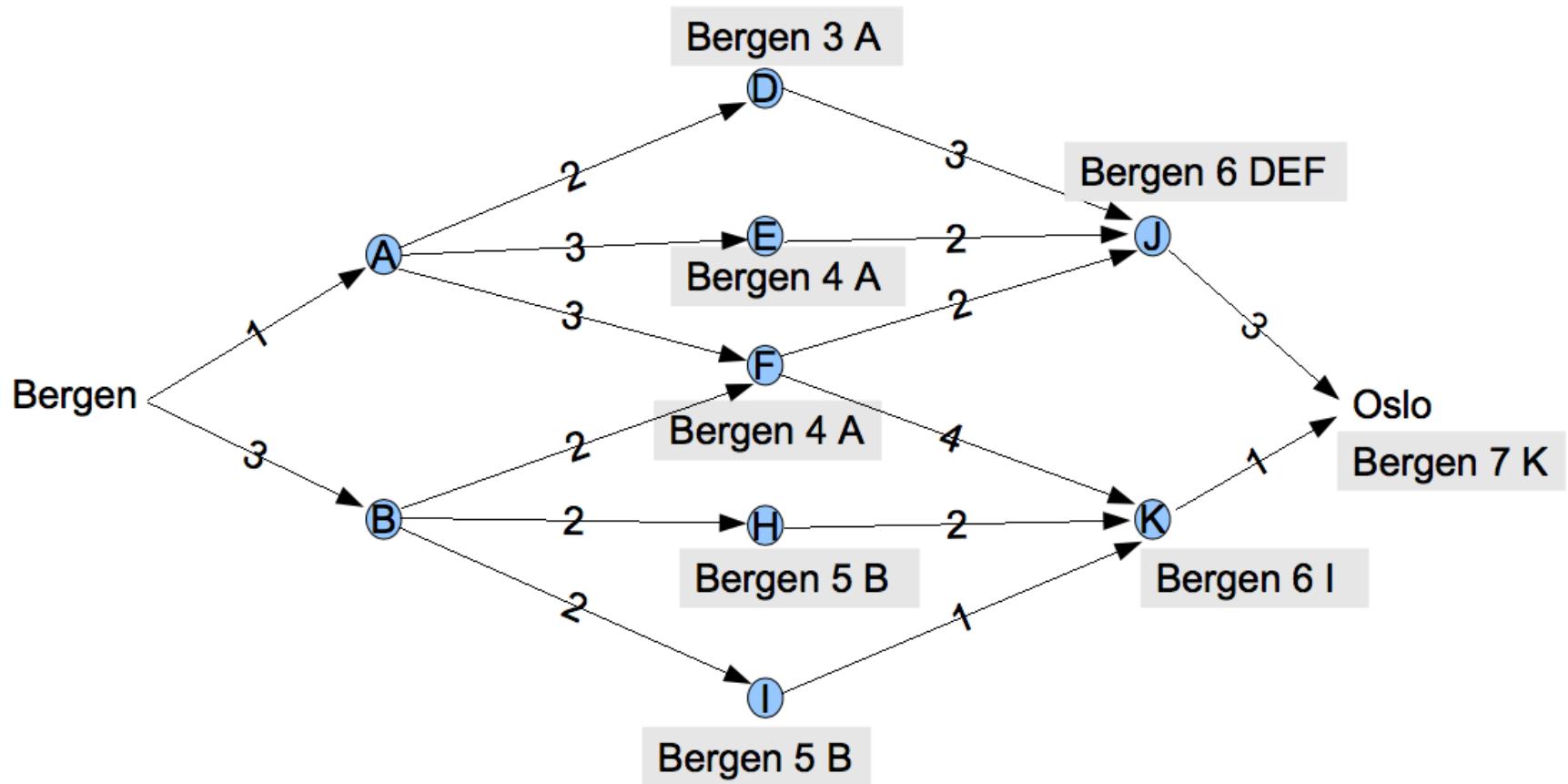
# An analogy

---



- The shortest path is not obvious
- Enumerating all paths is not a viable solution for anything but a trivial case (like the above)
- Greedy algorithm is inappropriate
- Solution: dynamic programming

# Illustration of dynamic programming



# The best alignment

---

Example scoring scheme

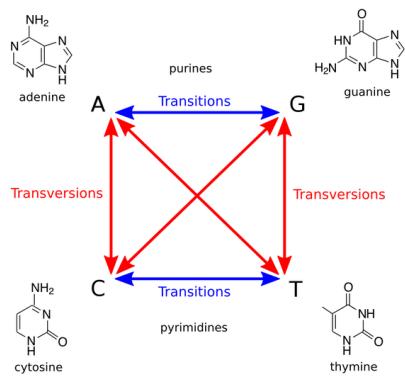
- gap: -1
- match: 2
- transition: 1
- transversion: -2

		A	C	T	G	T	A	
		0	-1	-2	-3	-4	-5	-6
G	-1							
C	-2							
T	-3							
G	-4							
G	-5							
T	-6							
A	-7							
A	-8							

# Needleman-Wunsch 1970

Example scoring scheme

- gap:  $W_1 = -1$
- match:  $a_i = b_j, s(a_i, b_j) = 2$
- mismatch:  $a_i \neq b_j$ 
  - transition:  $s(a_i, b_j) = 1$
  - transversion:  $s(a_i, b_j) = -2$



	A	C	T	G	T	A
0	-1	-2	-3	-4	-5	-6
G	-1	1	0	-1	-1	-2
C	-2	0	3	2	1	0
T	-3	-1	1	5	0	3
G	-4	-2	0	4	7	6
G	-5	-3	-1	3	6	5
T	-6	-4	-2	2	5	8
A	-7	-4	-3	1	4	7
A	-8	-5	-4	0	3	6
						9

$H_{i-1,j-1}$	$H_{i-1,j}$
$H_{i,j-1}$	$H_{i,j}$

$$H_{ij} = \max \begin{aligned} & H_{i-1,j-1} + s(a_i, b_j), \\ & H_{i-1,j} + W_1, \\ & H_{i,j-1} + W_1, \end{aligned}$$

# The best alignments

	A	C	T	G	T	A	
0	-1	-2	-3	-4	-5	-6	
G	-1	1	0	-1	-1	-2	-3
C	-2	0	3	2	1	0	-1
T	-3	-1	1	5	0	3	2
G	-4	-2	0	4	7	6	5
G	-5	-3	-1	3	6	5	7
T	-6	-4	-2	2	5	8	7
A	-7	-4	-3	1	4	7	10
A	-8	-5	-4	0	3	6	9

ACT-GTA-  
GCTGGTAA

Truth

ACT-GT-A  
GCTGGTAA

$$H_{ij} = \max$$

$$\begin{aligned} & H_{i-1,j-1} + s(a_i, b_j), \\ & H_{i-1,j} + W_1, \\ & H_{i,j-1} + W_1, \end{aligned}$$

ACTG-TA-  
GCTGGTAA

ACTG-T-A  
GCTGGTAA

Time complexity is  $O(mn)$

Space complexity is same, but can be made linear

# Scoring matrix

---

- We chose a simple scoring matrix
- More generally:
  - the score of the alignment of two residues should **reflect the probability that they are homologous**
  - or in other words that one residue is the result of one or several mutations of the other
- Gap penalties
  - we used a linear gap penalty
  - a more general form would be an affine gap penalty, this means that all blanks in the alignment do **NOT** carry the same penalty

$$\text{gap cost} = \text{opening cost} + \text{length} \times \text{extension cost}$$



Affine gap penalty leads to a minor modification to the scoring of the matrix, but principle remains the same

# The importance of scoring

---

[http://www.ebi.ac.uk/Tools/psa/emboss\\_needle/nucleotide.html](http://www.ebi.ac.uk/Tools/psa/emboss_needle/nucleotide.html)

Two homologous sequences

AGTAAAATTATATATGTA  
GGTAAAAA-----ATATGTT



Needleman-Wunsch  
Gap open 5 and ext 1 (typical defaults)  
Notice the affine gap



AGTAAAATTATATATGTA	18
.                  . .	
GGTAAAAA-----ATATGTT	14

# The importance of scoring

[http://www.ebi.ac.uk/Tools/psa/emboss\\_needle/nucleotide.html](http://www.ebi.ac.uk/Tools/psa/emboss_needle/nucleotide.html)

AGTAAAATTATATATGTA  
GGTAAAA----ATATGTT



Gap open 1 and ext 0.001  
("cheap" gap)

A-GTAAAATTATATATG-TA      18  
  |||||        ||||| |  
-GGTAAAA----ATATGTT-      14

Gap open 50 and ext 1  
("expensive" gap)

AGTAAAATTATATATGTA      18  
.|||||.||||.|.  
GGTAAAAATATGTT----      14

# Practical with Needleman-Wunsch

$H_{i-1,j-1}$	$H_{i-1,j}$
$H_{i,j-1}$	$H_{i,j}$

# Example scoring scheme

- gap:  $W_1 = -1$
  - match:  $a_i = b_j, s(a_i, b_j) = 2$
  - mismatch:  $a_i \neq b_j, s(a_i, b_j) = -1$

## 1. Initialise

## 2. Score

$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ H_{i-1,j} + W_1, \\ H_{i,j-1} + W_1, \end{cases}$$

### 3. Traceback

# Practical with Needleman-Wunsch

$H_{i-1,j-1}$	$H_{i-1,j}$
$H_{i,j-1}$	$H_{i,j}$

Example scoring scheme

- gap:  $W_1 = -1$
- match:  $a_i = b_j, s(a_i, b_j) = 2$
- mismatch:  $a_i \neq b_j, s(a_i, b_j) = -1$

## 1. Initialise

## 2. Score

$$H_{i-1,j-1} + s(a_i, b_j),$$

$$H_{i-1,j} + W_1,$$

$$H_{i,j-1} + W_1,$$

## 3. Traceback

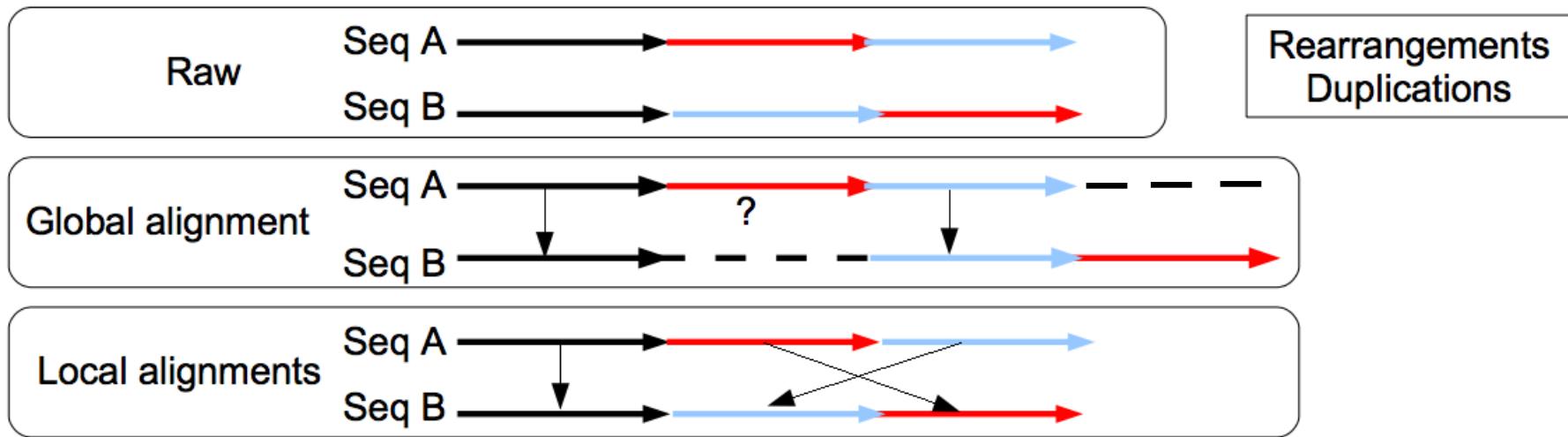
	A	G	T	C	T	G	A	T
O	-1	-2	-3	-4	-5	-6	-7	-8
A	-1	-2	-2	-3	-4	-5	-6	-7
T	-2	-3	-1	0	-1	-2	-3	-4
T	-3	-4	0	1	0	3	2	1
C	-4	-5	-1	-2	-1	2	1	0
G	-5	-6	-2	-3	-1	4	3	2
A	-6	-7	-3	-4	0	1	0	1
T	-7	-8	-4	-5	-1	2	1	0

①

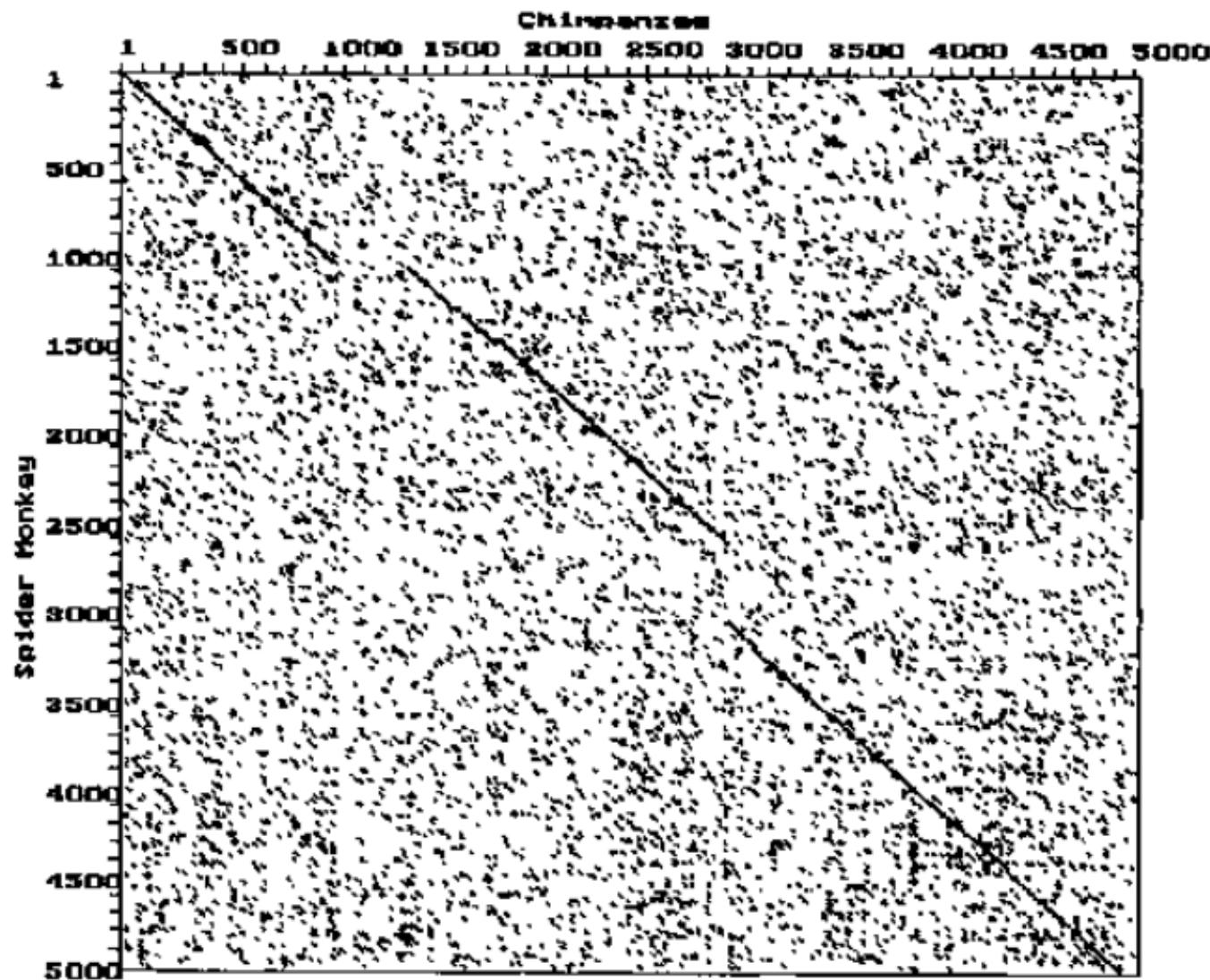
②

① A	G	T	C	T	G	A	T
② A	T	T	C	T	G	A	T

# Pairwise local alignment



# A visual technique – The dot-plot



Identities of length 6bp- Chimpanzee hemoglobin intergenic DNA  
against spider monkey. From [helix.biology.mcmaster.ca](http://helix.biology.mcmaster.ca)

# DP – Smith Waterman 1981

Protein sequences

Needleman-Wunsch scoring

q\d		R	E	D	C	H	D	K	L	
	i\j	0	1	2	3	4	5	6	7	8
	0	0	-0.5	-1	-1.5	-2	-2.5	-3		
A	1	-0.5	-0.3	-0.8	-1.3	-1.8	-2.3	-2.8		
C	2	-1	-0.8	-0.6	-1.1	-0.8	-1.3	-1.8		
E	3	-1.5	-1.3	-0.3	-0.8	-1.3	-0.3	-0.8		
D	4	-2	-1.8	-0.8	0.2	-0.3	-0.8	0.2		
E	5	-2.5	-2.3	-1.3	-0.3	-0.1	-0.2	-0.7		
C	6	-3	-2.8	-1.8	-0.8	0.2	-0.3	-0.5		
A	7	-3.5	-3.3	-2.3	-1.3	-0.3	-0.1	-0.6		
D	8	-4	-3.8	-2.8	-1.8	-0.8	-0.6	0.4		
E	9									

Smith-Waterman scoring

q\d		R	E	D	C	H	D	K	L	
	i\j	0	1	2	3	4	5	6	7	8
	0	0	0	0	0	0	0	0	0	0
A	1	0	0	0	0	0	0	0	0	0
C	2	0	0	0	0	0.5	0	0	0	0
E	3	0	0	0.5	0	0	1	0.5	0	0
D	4	0	0	0	1	0.5	0.5	1.5	1	0.5
E	5	0	0	0.5	0.5	0.7	1.0	1.0	1.2	0.7
C	6	0	0	0	0.2	1	0.5	0.7	0.7	0.9
A	7	0	0	0	0	0.5	0.7	0.2	0.4	0.4
D	8	0	0	0	0.5	0	0.2	1.2	0.7	0.2
E	9	0	0	0.5	0	0.2	0.5	0.7	0.9	0.4

From Eidhammer et al. 2004

Smith-Waterman is an adaptation of Needleman-Wunsch:

	Needleman-Wunsch	Smith-Waterman
Initialisation	First row and first column are subject to gap penalty	First row and first column set to 0
Scoring	Score can be negative	Negative score is set to 0
Traceback	Begin with cell at bottom right and end at the top left	Begin with the highest score and end when reach score 0

---

# **APPENDIX**

# Local alignment in practice – BLAST 1990

- Smith-Waterman is time expensive, especially for searching large databases >> use **heuristics**
- **BLAST** is the most popular local alignment heuristic
  - find **short segments** of **equal length** that score > T (preprocessing and scanning)
  - **extend** the matches formed by such segment pairs **without introducing gaps** as long as the score does not fall below threshold (~90% execution time)

Example with segment length 2  
(in practice longer)

2    4    6  
GCTGGTA  
1    3    5    7

$4^2$  words

GA	4:3	7:3
GC	1:4	5:3
GG		
GT		
CA		
CC		
CG		
.		
.		

- match: 2
- transition: 1
- transversion: -2

This basic BLAST method was further refined in 1997 with the two hit method