Single alignment: Substitution Matrix

16 march 2017

BLOSUM Matrix

BLOSUM Matrix [2] **(Bl**ocks Amino Acid **Su**bstitution **M**atrices **)**

- It is based on the amino acids substitutions observed in ~2000 conserved block of sequences.
- Such blocks have been extracted from BLOCKS data bank composed by 500 family of proteins.

[2] Henikoff S and Henikoff JG (1992) Amino acid substitution matrices from protein blocks. Proc. Natl. Acad. Sci USA 89, 10915-10919

BLOSUM MATRIX

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Matrice

BLOSUM 62

BLOCKS and Blocks DB

http://blocks.fhcrc.org/blocks/help/

Blocks are multiply aligned ungapped segments corresponding to the most highly conserved regions of proteins.

Block Searcher, Get Blocks and Block Maker are aids to the detection and verification of protein sequence homology. They compare a protein or DNA sequence to a database of protein blocks, retrieve blocks, and create new blocks, respectively.

The Blocks Database

The blocks for the Blocks Database are made automatically by looking for the most highly conserved regions in groups of proteins documented in InterPro.

The blocks created by Block Maker are created in the same manner as the blocks in the Blocks Database but with sequences provided by the user.

BLOCKS Format

http://blocks.fhcrc.org/blocks/help/blocks_format.html

http://blocks.fhcrc.org/blocks-bin/getblock.sh?IPB001525#IPB001525A

```
ID   short_identifier; BLOCK
AC   block_number; distance from previous block = (min,max)
DE   description
BL   xxx motif; width=w; seqs=s; 99.5%=n1; strength=n2
sequence_id  (offset) sequence_segment  sequence_weight
.
.
.
//
```

ID line starts a block entry and contains a short identifier for the group of sequences from which the block was made. If the block was taken from InterPro, it will be the InterPro group ID. The identifier is terminated by a semi-colon, and the word "BLOCK" indicates the entry type.

AC line contains the block number, a seven-character group number for sequences from which the block was made, followed by a letter (A-Z) indicating the order of the block in the sequences. If the group has only one block, the letter is omitted. If the block was made from InterPro group IPRnnnnnn, the block number is IPBnnnnnna. If the block was converted from Terri Attwood's Prints Database the block number is PRnnnnna. *min,max* = minimum,maximum number of amino acids from previous block for sequences in this block. For the first block in the group, the distance from the beginning of the sequences.

DE line contains a description of the group of sequences from which the block was made. If the block was taken from InterPro, it will be a slightly edited version of the InterPro description.

BL line contains information about the block:
xxx = the amino acids in the spaced triplet found by MOTIF upon which the block is based.
w = width of the sequence segments (columns) in the block.
s = number of sequence segments (rows) in the block.
n1 = raw calibration score; 99.5th percentile score of true negative sequences. Raw search scores are normalized by dividing by this score and multiplying by 1000.
n2 = median normalized score of known true positive sequences as documented in InterPro.

Following the BL line are lines for each sequence with a segment in the block. The segments may be clustered with clusters separated by blank lines. Each segment line contains a sequence identifier, the offset from the beginning of the sequence to the block in parentheses, the sequence segment, and a weight for the segment. The weights are normalized so that the most distant segment has a weight of 100.

// line terminates a block entry.

BLOSUM vs PAM

PAM-BLOSUM differences:

- PAM matrixes derive evolutionary distances from the **assumption** that mutations are successive and independent and therefore can be added. BLOSUM matrixes, instead, do not make assumptions but are **based on the observation of accurate and real alignments**.

- All PAM are obtained by multiplying an initial matrix (the PAM 1), while the BLOSUM are calculated empirically, one by one, taking the blocks genetic sequences at a fixed distance (eg. In the BLOSUM 80 are taken only sequences identical to 80% with 20 mutations per 100 bp).

BLOSUM vs PAM

PAM-BLOSUM differences:

- An increasing index indicates PAM scores to more distant proteins among them (expressing an evolutionary distance), the growing BLOSUM index indicates the most similar proteins among themselves, expressing conservation minimum value of the BLOCK.

- The PAM tend to reward amino acid substitutions resulting from single base mutations rather than structural reasons of amino acids, as do instead BLOSUM.

BLOSUM 50

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5

Substitution score schema for gap

- *Linear score*
  - f(g)= -gd

  Where

  d represents the *gap-open penalty* and *g* is the length of gap

  In this schema the gap penalty depends only from the length of gap. This means that two isolated gaps has the same cost as a block of two consecutive gaps.
- *Affine score*
  - f(g)= -d –(g-1)

  where

  d represents *gap-open penalty*, and g represents the gap length.

  This schema is more biologically significant, since insertions and deletions of more than one residue are common events between homologous protein sequences.

Global alignment with indels

- Insertions and deletions (indels) are necessary to accurately align very similar sequences
- The naive approach to finding the optimal alignment of two sequences with indels is to generate all possible alignments, add the scores for each pair of corresponding symbols in each alignment and choosing the alignment with the maximum score.
- This approach is also impossible to short sequences (as little as 100 characters).

EXHAUSTIVE algorithms for pairwise global alignments

- The most trivial exhaustive method is to slide a sequence over one another, using the identity matrix to calculate the score:

```
AAKKQV →
  AAKWQ
Score: 0
```

```
AAKKQV →
   AAKWQ
Score: 0
```

```
AAKKQV →
    AAKWQ
Score: 0
```

```
AAKKQV →
     AAKWQ
Score: 0
```

```
AAKKQV →
 AAKWQ
Score: 1
```

```
AAKKQV →
AAKWQ
Score: 4
```

```
AAKKQV →
 AAKWQ
Score: 1
```

```
AAKKQV →
  AAKWQ
Score: 0
```

```
AAKKQV →
AAKWQ
Score: 0
```

```
AAKKQV →
   AAKWQ
Score: 0
```

Possible combinations 6*5=30

EXHAUSTIVE algorithms for pairwise global alignments

Complexity of the naive approach:

Given two sequences **s1** and **s2** having length **|s1| = m , |s2| = n**

Let us assume that *m=n*, the complexity of the exhaustive algorithm is $\mathbf{O(kn^2)}$, where *k* represents the DB size.

It is clear that such an algorithm can be used only for short sequences (i.e., with small n) and for few sequences (i.e., small k).

EXHAUSTIVE algorithms for pairwise global alignments

GAP: When aligning the proteins, or even nucleic acids, it must take into account the fact that the best alignment can be achieved with the insertion of one or more gaps, which correspond to the insertion or deletion evolutionary phenomena :

```
PLMTRWDQEQESDFGHKLPIYTREWCTRG
      || | |||    |                    ||||| |
CHKIPLMTRWPQQESDFGHKLPVIYTREW
```

Score without gap = 13

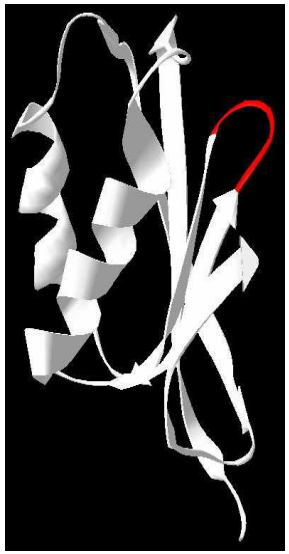
```
    IPLMTRWDQEQESDFGHKLP - IYTREWCTRG
     || | |||    |   ||||||||||  |||||||
CHKIPLMTRWPQ - QESDFGHKLPV IYTREW
```

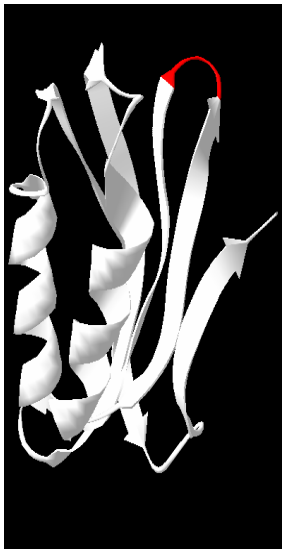
Score with gap = 24

EXHAUSTIVE algorithms for pairwise global alignments

biological meaning of GAPS:

Although the insertion of gaps can appear arbitrary, in reality there is a biological consideration that permits it; evolution can insert or remove debris (i.e., residues) from sequences to shrink or expand certain structures or to change the relationships between domains, however the overall shape of protein structure is maintained.





Global and local alignments

Global Alignment:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
 ||.  | |  |  . |    .|  ||  ||   ||  = 13 + 3
 TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

Local Alignement:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
        ||||||||.|||||                 = 13 + 1
      TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

Which is the best alignement?
Looking at the score it seems the first one, however which is the most significant from a biological viewpoint?

Global and local alignments

Global Alignment:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
 ||.  | |  |  . |    .|  ||  ||   ||  = 13 + 3
 TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

Local Alignement:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
        ||||||||.|||||                 = 13 + 1
      TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

The local alignment has found a protein domain that is not evident in the global alignment

In general, in a database, it is convenient to look for local similarity rather than global ones.

Global and local alignments

Global Alignment:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
 ||.  | |  |  . |    .|  ||  ||   ||  = 13 + 3
 TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

Local Alignement:

```
LTGARDWEDIPLWTDWDIEQESDFKTRAFGTANCHK
        ||||||||.|||||                 = 13 + 1
      TGIPLWTDWDLEQESDNSCNTDHYTREWGTMNAHKA
```

Whenever the sequences are very similar, it is very likely that they have similar local regions so extensive that the global alignment will give better results than the local one, as the former will minimize the gaps.

Global Pairwise Alignment: problem formalization

INPUT:

two sequences S and T defined on Σ alphabet and a score matrix $d: (\Sigma \cup \{-\})^2 \rightarrow R$

OUTPUT:

an alignment (S', T') between S and T whose score *A* is minimum (or maximum)

If d(a,b) represents a cost, A must be minimized.
If d(a,b) represents a benefit/reward, A must be maximized

Exhaustive Algorithms

1970:**Needleman & Wunsch** [1]: first algorithm for global alignment

1981:**Smith & Waterman** [2]: first agorithm for local alignment

They use the dynamic programming

They find the optimal alignment

But they are slow

[1] A general method applicable to the search for similarities in the amino acid sequence of two proteins
Saul B. Needleman, Christian D. Wunsch

[2] Identification of Common Molecular Subsequences
Temple F. Smith and Michael S. Waterman

Heuristic Algorithms for Sequence Alignment

FASTA - Fast A [1]**:** It performs fast search on any alphabet-

BLAST - Basic **L**ocal **A**lignment **S**earch **T**ool [2]: A BLAST search allows to compare a sequence of interest with an already known sequence database, and to identify, among the latter those which have some similarities with the sequence of interest.

[1] **Rapid and sensitive protein similarity searches**
Lipman DJ, Pearson WR.
[2] Basic local alignment search tool
Altschul, S; Gish, W; Miller, W; Myers, E; Lipman, D

First Homework

Implement the naive algorithm for global pairwise alignment using a programming language you prefer. If you already know Python, use it.

The program takes in input the sequence A, a score matrix M and a DB of sequences D and returns in output the sequence S in D that is most similar to A, the execution time of the whole algorithm and the processing time required to allign the A with S.

The DB D can be simple a text file containing a sequence per each line. Run the code using always the same matrix M and the same sequence A, but with different DB size (from 1 to 201 with step of 20) so to have 11 runs to compare.

Send by email the code and the report for the 11 runs by Monday, March 27th

FASTA

Homology vs Similarity

When comparing two sequences or structures, the terms of similarity or homology are often used interchangeably to indicate that there is a "close" relationship between the comparison objects.

However, the two terms refer to different aspects of the comparison:

The **homology** is a **qualitative** property of the comparison: you say that sequences are homologous if they have a common ancestral gene, or that share an ancestor. So the term **homology** indicates that the two entities share a common phylogenetic origin, from which they have evolved differentiated from each other.

The **similarity** and a quantitative property of a comparison. The term similarity has a more general meaning indicating a similarity regardless of the reasons that led it. The **similarity** is often due to homology but it can also be generated by the case or by adaptive convergence phenomena at both morphological and molecular level.

FASTA

FASTA is a DNA and protein sequence alignment software package first described (as FASTP) by David J. Lipman and William R. Pearson in 1985.[1]

FASTA is pronounced "fast A", and stands for "FAST-All", because it works with any alphabet, and it is an extension of "FAST-P" (protein) and "FAST-N" (nucleotide) alignment.

The current FASTA package contains programs for protein:protein, DNA:DNA, protein:translated DNA (with frameshifts), and ordered or unordered peptide searches.

Recent versions of the FASTA package include special translated search algorithms that correctly handle frameshift errors (which six-frame-translated searches do not handle very well) when comparing nucleotide to protein sequence data.

[1] Lipman, DJ; Pearson, WR (1985). "Rapid and sensitive protein similarity searches". Science. 227 (4693): 1435–41.

FASTA

In addition to rapid heuristic search methods, the FASTA package provides SSEARCH, an implementation of the optimal Smith-Waterman algorithm.

A major focus of the package is the calculation of accurate similarity statistics, so that biologists can judge whether an alignment is likely to have occurred by chance, or whether it can be used to infer homology.

The FASTA program in its classic version, is a heuristic program able to search the **global sequence similarity**. Two variants thereof, and LFASTA PLFASTA, are able to search for **local sequence similarity**.

The FASTA package is available from fasta.bioch.virginia.edu.

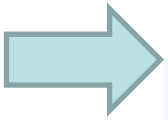
FASTA

FASTA is a four steps algorithm.

First Step: OFFSET Definition

Initially, a table is created and that contains all the positions for each type of amino acid (or nucleotide) within each of the sequences present in the database. For example, if it exists in the database a sequence like the one below, it is created the table at the right side:

Position	1	2	3	4	5	6	7
Sequence A	F	L	W	R	T	W	S

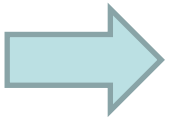


F	1
L	2
W	3, 6
R	4
T	5
S	7

FASTA

First Step: OFFSET Definition

Position	1	2	3	4	5	6
Sequence B	S	W	R	T	W	T



S	1
W	2, 5
R	3
T	4, 6

The positional table can be constructed taking into account the position of the amino acids taken individually (ktup = 1) (as is the case above) or as taken in pairs (ktup = 2). Using this second mode will result in a speeding up of the process at the expense of the accuracy of the final data.
However, the approximation is still valid because it can be assumed that the homology between two sequences are meaningful only if it can be considered pairs of amino acids and not individual amino acids. In the case of nucleotide sequences ktup is 4 or 6.

FASTA

First Step: OFFSET Definition

At this point you have to run the mathematical difference of positional values of the amino acids of the same type in the sequence that is being compared (sequence query) and the sequences in the database. This difference is also called OFFSET.

In the example below, the amino acids common to the two sequences are S, W (present in two positions), R and T.

The comparison of the positional values of these amino acids in the two sequences gives rise to these values of OFFSET:

F	**1**
L	**2**
W	**3, 6**
R	**4**
T	**5**
S	**7**

S	**1**
W	**2, 5**
R	**3**
T	**4, 6**

AA	DELTA	OFFSET
S	7 - 1	6
W	3 – 2	1
W	3 – 5	-2
W	6 – 2	4
W	6 – 5	1
R	4 – 3	1
T	5 – 4	1
T	5 – 6	-1

FASTA

First Step: OFFSET Definition

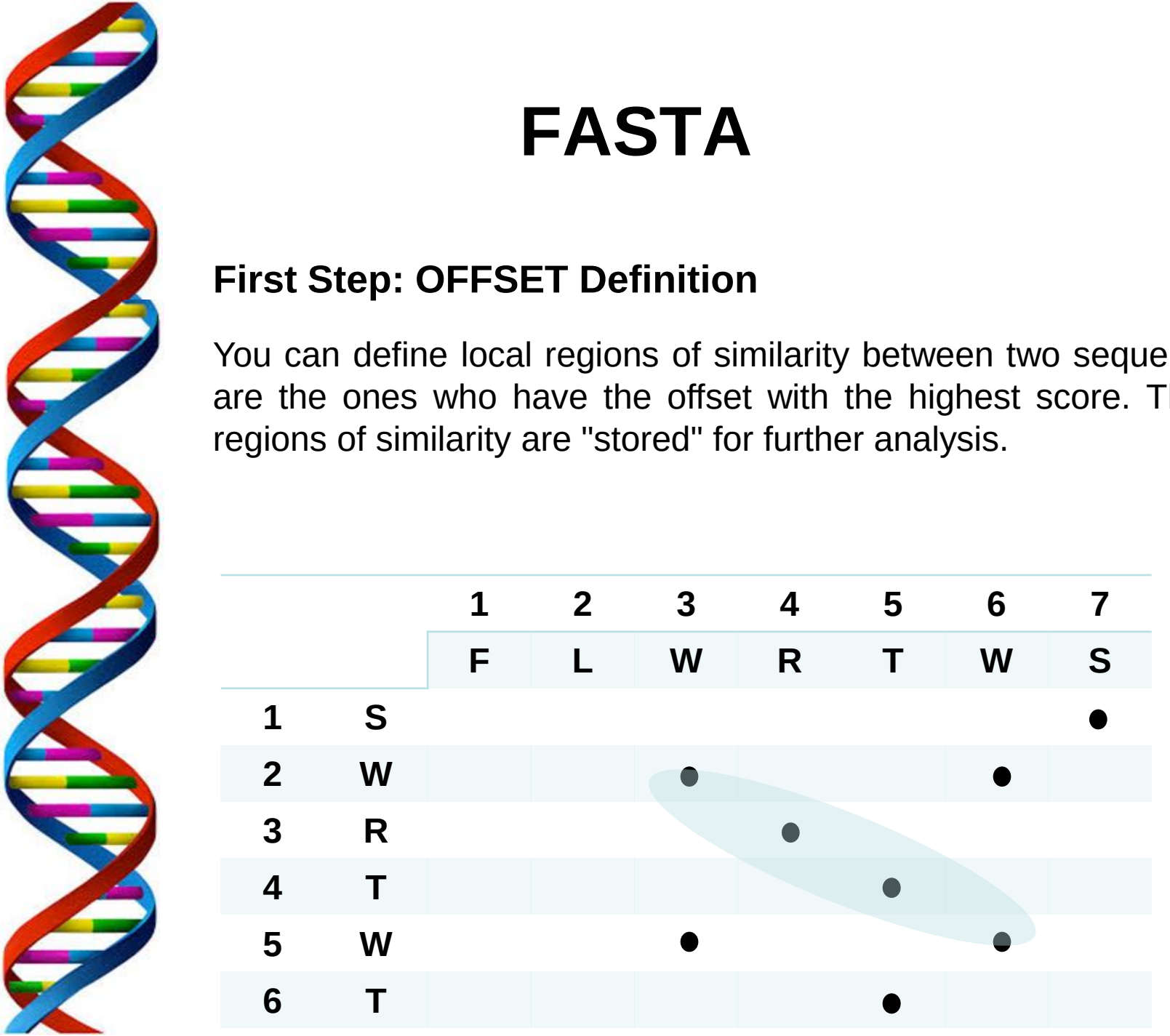
In the example, the best offset is the one with the value 1 which allows alignment of 4 amino acids. The other offsets allow the alignment of a single amino acid or no amino acid. The score in an offset is increased for each identity and reduced for each misalignment (mismatch). The latter is allowed, while insertions / deletions that are forbidden.

		1	2	3	4	5	6	7
		F	L	W	R	T	W	S
1	S							●
2	W			●			●	
3	R				●			
4	T					●		
5	W			●			●	
6	T					●		

FASTA

First Step: OFFSET Definition

You can define local regions of similarity between two sequences. They are the ones who have the offset with the highest score. The 10 best regions of similarity are "stored" for further analysis.



		1	2	3	4	5	6	7
		F	L	W	R	T	W	S
1	S							●
2	W			●			●	
3	R				●			
4	T					●		
5	W			●			●	
6	T					●		