Bioinformatics
&
(Computational) Molecular Biology

Introduction

Alexander (Sacha) Gultyaev
a.p.goultiaev@liacs.leidenuniv.nl
**Biopolymer sequences**

**DNA**: double-helical nucleic acid. Monomers: nucleotides C, A, T, G.

![DNA structure](image1)

from “Molecular Cell Biology”, Lodish et al. (2000)

**RNA**: (single-stranded) nucleic acid. Monomers: nucleotides C, A, U, G.

![RNA structure](image2)

from “Molecular Biology of the Cell”, Alberts et al. (2002)

**Proteins**: polypeptide chains. Monomers: amino acids (20 types).

![Protein structure](image3)

from “Molecular Cell Biology”, Lodish et al. (2000)
Biopolymer sequences

Watson-Crick complementarity: AT and GC pairs in double helix

DNA
5′- ATGGCGCAGGG...-3′
3′-TACCGCGTCCC...-5′

transcription (RNA polymerase)

RNA
5′- AUGGCGCAGGG...-3′

translation (ribosome)

protein MetAlaGlnGly...

Genetic code is redundant: $4 \times 4 \times 4 = 64$ nucleotide triplets encode for 20 amino acids. Only two amino acids (Met and Trp) are encoded by a single codon. Other amino acids can be encoded by 2, 4 or 6 codons. Usually the last “wobble” position of a triplet determines “silent” substitutions. ATG (Met) is start codon (usually). TAA, TAG, TGA - stop codons.
Basic signals in gene expression

- **transcription start**
- **transcription termination**
- **genome (dsDNA)**
- **transcript (mRNA)**
- **coding sequence (ORF)**
  - start-codon: ATG
  - stop-codon: TAA or TAG or TGA
- **untranslated region (5’-UTR)**
- **untranslated region (3’-UTR)**
Basic signals in gene expression

Prokaryotes (bacteria): polycistronic mRNA

genome (dsDNA)

transcription start

ORF1

ORF2

ORF3

transcript (mRNA)

transcription termination
Basic signals in gene expression

Eukaryotes: precursor-mRNA (pre-mRNA) processing (splicing)

Alternative splicing, such as exon skipping or intron retention, leads to diverse isoforms of mRNAs and proteins encoded by the same gene. Due to frameshifts the sequences of proteins could be different.
NCBI database resources / Entrez retrieval system
Nucleotide sequence databases

Initially three main databases: GenBank (USA),
   EMBL (Europe),
   DDBJ (Japan).

Later the three databases became parts of
the **International Nucleotide Sequence Database Collaboration**.

The three organizations exchange data on a daily basis.

Each record is assigned a unique identifier, **Accession number**, that is
shared by three databases.

A single flat file format of database entries is used.
Datafields of sequence database entries:

LOCUS       NM_000518                626 bp    mRNA    linear   PRI 24-MAY-2014
DEFINITION  Homo sapiens hemoglobin, beta (HBB), mRNA.
ACCESSION   NM_000518
VERSION     NM_000518.4  GI:28302128
KEYWORDS    RefSeq.
SOURCE      Homo sapiens (human)
ORGANISM    Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
            Mammalia; Eutheria; Euarchontoglires; Primates; Haplorrhini;
            Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 626)
AUTHORS     Mei Y, Yin N, Jin X, He J and Yin Z.
TITLE       The regulatory role of the adrenergic agonists phenylephrine and isoproterenol on fetal hemoglobin expression and erythroid differentiation
JOURNAL     Endocrinology 154 (12), 4640-4649 (2013)
PUBMED      24080366

FEATURES             Location/Qualifiers
source          1..626
            /organism="Homo sapiens"
            /mol_type="mRNA"
            /db_xref="taxon:9606"
            /chromosome="11"
            /map="11p15.5"
            gene 1..626
            /gene="HBB"
            /gene_synonym="beta-globin; CD113t-C"
            /note="hemoglobin, beta"
            /db_xref="GeneID:3043"
            /db_xref="HGNC:HGNC:4827"
            /db_xref="HPRD:00786"
            /db_xref="MIM:141900"
            exon 1..142
            /gene="HBB"
            /gene_synonym="beta-globin; CD113t-C"
            /inference="alignment:Splign:1.39.8"
            CDS 51..494
            /gene="HBB"
            /gene_synonym="beta-globin; CD113t-C"
            /note="beta globin chain; hemoglobin beta chain"
            /codon_start=1
            /product="hemoglobin subunit beta"
            /protein_id="NP_000509.1"
            /db_xref="GI:4504349"
            /db_xref="CCDS:CCDS7753.1"
            /db_xref="GeneID:3043"
            /db_xref="HGNC:HGNC:4827"
            /db_xref="HPRD:00786"
            /db_xref="MIM:141900"
            /translation="MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFE
            SFGDLSSTPDAVMPNGKPVKAGKRVLGAFSGDLALDNLKTGTFASLSLEDHNDIVDPE
            NFRLGMVLYCVLAIHIFGKEETTFPVQAYQKYVAGYVANALAHKYII"

ORIGIN
1 acatttgcct ctgcactaagt tgtgttcatct agcaacacta aacagacacc atgtgcagtc
61 tgacctctga ggagaggtct ccctgttacgc cctctgtggc caaggtgaaatgtggatgaag
Examples of features annotated:

- **gene**: 1..7898
  
  /gene="SRSF7"

- **mRNA**: join(1..266,1301..1481,1790..1966,2842..2916,3338..3448,4748..4801,6308..7898)
  
  /gene="SRSF7"

- **CDS**: join(239..266,1301..1481,1790..1966,2842..2916,3338..3448,4748..4801,6308..6362)
  
  /gene="SRSF7"

- **CDS complement**: (46224..48638)
  
  /locus_tag="KPHS_00400"
  /codon_start=1
  /transl_table=11
  /product="formate dehydrogenase-O alpha subunit"
  /protein_id="YP_005224340.1"
  /db_xref="GeneID:11845018"

- **exon**: 697..832

- **regulatory**: 7153..7158
  
  /regulatory_class="polyA_signal_sequence"
GenBank and WGS Statistics

<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
<th>GenBank Bases</th>
<th>GenBank Sequences</th>
<th>WGS Bases</th>
<th>WGS Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Dec 1982</td>
<td>680338</td>
<td>606</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Nov 1983</td>
<td>2274029</td>
<td>2427</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>May 1984</td>
<td>3002088</td>
<td>3665</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Release</th>
<th>Date</th>
<th>GenBank Bases</th>
<th>GenBank Sequences</th>
<th>WGS Bases</th>
<th>WGS Sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>225</td>
<td>Apr 2018</td>
<td>260189141631</td>
<td>208452303</td>
<td>2784740996536</td>
<td>621379029</td>
</tr>
<tr>
<td>226</td>
<td>Jun 2018</td>
<td>263957884539</td>
<td>209775348</td>
<td>2944617324086</td>
<td>639804105</td>
</tr>
<tr>
<td>227</td>
<td>Aug 2018</td>
<td>260806936411</td>
<td>208831050</td>
<td>3204855013281</td>
<td>665309765</td>
</tr>
</tbody>
</table>

Databases of amino acid sequences

Historically: the first databases.

Nowadays amino acid sequences are predominantly determined by translation of massively sequenced nucleic acids. Thus a database of amino acid sequences is secondary or curated database. (In contrast to e.g. primary GenBank with records obtained from submitters.)

ENTREZ protein database: a collection of entries from several databases such as SWISS-PROT (one of the oldest and popular databases) and translations of nucleotide sequences in GenBank.

UniProtKB (uniprot.org): Knowledgebase, contains both amino acid sequences and functional annotation.
- includes SWISS-PROT (manually annotated and reviewed) and
- TrEMBL (suggested coding regions in the nucleotide database entries, automatically annotated and not reviewed).

Note that a number of coding regions (ORFs) in the nucleotide database entries may remain unannotated.
The Reference Sequence (RefSeq) Database

Non-redundant, richly annotated records of nucleotide and amino acid sequences.

RefSeq entries are similar to those of GenBank, but they have some distinct features, in particular, specific Accession prefixes.

<table>
<thead>
<tr>
<th>Accession prefix</th>
<th>Molecule type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC_</td>
<td>Genomic</td>
<td>Complete genomic molecule, usually alternate assembly</td>
</tr>
<tr>
<td>NC_</td>
<td>Genomic</td>
<td>Complete genomic molecule, usually reference assembly</td>
</tr>
<tr>
<td>NG_</td>
<td>Genomic</td>
<td>Incomplete genomic region</td>
</tr>
<tr>
<td>NT_</td>
<td>Genomic</td>
<td>Contig or scaffold, clone-based or WGS&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>NW_</td>
<td>Genomic</td>
<td>Contig or scaffold, primarily WGS&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>NS_</td>
<td>Genomic</td>
<td>Environmental sequence</td>
</tr>
<tr>
<td>NZ&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Genomic</td>
<td>Unfinished WGS</td>
</tr>
<tr>
<td>NM_</td>
<td>mRNA</td>
<td></td>
</tr>
<tr>
<td>NR_</td>
<td>RNA</td>
<td></td>
</tr>
<tr>
<td>XM&lt;sup&gt;c&lt;/sup&gt;</td>
<td>mRNA</td>
<td>Predicted model</td>
</tr>
<tr>
<td>XR&lt;sup&gt;c&lt;/sup&gt;</td>
<td>RNA</td>
<td>Predicted model</td>
</tr>
<tr>
<td>AP_</td>
<td>Protein</td>
<td>Annotated on AC_ alternate assembly</td>
</tr>
<tr>
<td>NP_</td>
<td>Protein</td>
<td>Associated with an NM_ or NC_ accession</td>
</tr>
<tr>
<td>YP&lt;sup&gt;c&lt;/sup&gt;</td>
<td>Protein</td>
<td></td>
</tr>
<tr>
<td>XP&lt;sup&gt;c&lt;/sup&gt;</td>
<td>Protein</td>
<td>Predicted model, associated with an XM_ accession</td>
</tr>
<tr>
<td>ZP&lt;sup&gt;c&lt;/sup&gt;</td>
<td>Protein</td>
<td>Predicted model, annotated on NZ_ genomic records</td>
</tr>
</tbody>
</table>

From: Chapter 18, The Reference Sequence (RefSeq) Database

The NCBI Handbook [Internet].
McEntyre J, Ostell J, editors.
Bethesda (MD): National Center for Biotechnology Information (US); 2002-.
Entrez Gene database

Gene-centered database. Integrates info from multiple databases. Typically an entry follows the annotation of RefSeq entries. The records have multiple links to other databases.

A fragment of an entry of the Gene database:

Transcript accessions

Intron

Exon

Coding region

Genomic Sequence:

Go to reference sequence details

Go to nucleotide:

Genomic regions, transcripts, and products

Genomes, NCBI Homo sapiens Annotation Release 108, 2016-06-07

Genomes, Ensembl release 85

Genome browser transcript data

Database of known mutations (dbSNP)
Sequence alignment

- Sequence alignment is used for identification of homology and/or similarity of sequences.

  Homology (evolutionary history) is not equivalent to similarity (e.g. % identity).

- However, identification of sequence similarity helps to reveal the homology.

- Similarity of 1D sequences (primary structures) can be seen in sequence monomers of two sequences mapped against each other:

  Sequence A: GCTTA----GCTATTGGCTTCTCTAAT--CACCAAGGGATATGCATACAAAAAACATTCT

  Sequence B: GATTATTTAGCTATTGGCTTCTTTAATAATAACCATTGATATG------GAAAAATTTCT

  Sequences can be aligned in many different ways

  Sequence A: GC----TTAGCTATTGGCTTCTCTAATC--ACCAAGGGATATGCATACAAAAAACATTCT

  Sequence B: GATTATTTAGCTATTGGCTTCTTTAATAATAACCATTGATATG------TGGAAAAATTTCT

  Insertion A->B or deletion B->A (alignment “gap”)

  Substitutions

Alignment algorithms attempt to identify most likely alignment, trying to follow the molecular mechanisms of sequence evolution: substitutions, deletions, insertions.
Searching for optimal alignment

Alternative alignments can be viewed as alternative paths in 2D sequence space.

Most likely alignment should contain “as-large-as-possible” number of most likely events (conservation of monomers -> matched positions) and “as-small-as-possible” number of less likely events (substitutions, gaps). It is possible to assign some scores to all alignment elements according to their probabilities in biologically relevant model.

An alignment can be scored, and so can be the corresponding path.
Searching for optimal alignment

Two separate issues in finding the optimal alignment:

1. Scoring system.

2. Algorithm to find the alignment with the best score
   (optimal alignment = optimal path).

Scoring may be relatively simple, for instance, the default parameters of the BLASTN program for alignment of nucleotide sequences:

(+2) for match;
(-3) for mismatch;
(-5) for the first gap nucleotide and (-2) for each of the nucleotides in gap extension
(negative “penalties”)
Scoring system should be derived from observed substitution frequencies in homologous sequences

During DNA replication, transitions (A ↔ G, C ↔ T) are more frequent as compared to transversions (A ↔ C, A ↔ T, G ↔ T, G ↔ C). This can be taken into account by a substitution matrix, e.g. as shown below:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>91</td>
<td>-114</td>
<td>-31</td>
<td>-123</td>
</tr>
<tr>
<td>C</td>
<td>-114</td>
<td>100</td>
<td>-125</td>
<td>-31</td>
</tr>
<tr>
<td>G</td>
<td>-31</td>
<td>-125</td>
<td>100</td>
<td>-114</td>
</tr>
<tr>
<td>T</td>
<td>-123</td>
<td>-31</td>
<td>-114</td>
<td>91</td>
</tr>
</tbody>
</table>

Identities have positive scores, substitutions - negative penalties.

Diagonal elements should not be equal, reflecting differences in occurrence of various nucleotides.
Amino acid substitution matrices

Amino acid substitution matrices take into account so-called “conservative” substitutions between residues with similar properties (e.g. Arg ↔ Lys).

The scores for a $20 \times 20$ matrix can be derived from frequencies observed in the datasets of related proteins.

These frequencies should be computed as log-odds: logarithms of ratios of the frequencies to the background ones that are determined by chance.

$$s(a,b) = \frac{1}{\lambda} \log \frac{p_{ab}}{f_a f_b}$$

Here $a$ and $b$ are two residue types, $p_{ab}$ is observed frequency, $f_a$ and $f_b$ are occurrences of $a$ and $b$, respectively, in all proteins. $\lambda$ is a scaling factor to make the scores convenient integers.
Gap penalties

Insertions or deletions (indels) are less frequent than point substitutions, and are therefore penalized in alignments by negative scores.

There is no reliable theoretical basis for gap statistics. Usually a linear function for gap penalty $S(gap)$ for a gap of $n$ monomers:

$$S(gap) = G + n \times L$$

Parameters $G$ (gap opening penalty) and $L$ (gap length or extension) are chosen empirically. The optimal choice is dependent on substitution matrices and expected similarity of aligned sequences.

For instance, $G = 10$ and $L = 1$ can be used in combination with BLOSUM62.

In alignments of nucleotide sequences the following parameters are chosen as default in BLASTN program for sequence database similarity search:

$G = 3$ and $L = 2$ in combination with match = 2 and mismatch = −3.
Searching for optimal alignment

Given a scoring system, the score of any alignment can be computed.

The problem is, however, to find the **optimal** alignment with the best score.

Even for alignment of two sequences of 300 monomers, about \(10^{179}\) alignments are possible...

Various types of alignment:

Pairwise alignment: global (full length) or local (finding the best aligned regions).

Sequence database similarity search: given a query sequence, find the best aligned sequences in the database.

Multiple sequence alignment: alignment of some number (>2) of sequences.

Different algorithms are designed for each of these problems.
Searching for optimal alignment

Alternative alignments can be viewed as alternative paths in 2D sequence space.

An alignment can be **scored**, and so can be the corresponding path.

- **Diagonal move**: match or mismatch.
- **Vertical or horizontal move**: gap.
Searching for optimal alignment by a dynamic programming algorithm

Recursive calculation of the optimal alignment score $S(i,j)$:

$$S(i,j) = \max \begin{cases} 
S(i,j) + M_{\text{subst}}[A(i),B(j)], \\
S(i,j-1) + G, \\
S(i-1,j) + G.
\end{cases}$$

$M_{\text{subst}}$ - substitution matrix of the scoring system;

$G$ - gap penalty.
Searching for optimal alignment by a dynamic programming algorithm

Recursive calculation of the optimal alignment score $S(i,j)$:

$$S(i,j) = \max \begin{cases} S(i,j) + M_{\text{subst}}[A(i),B(j)], \\ S(i,j-1) + G, \\ S(i-1,j) + G. \end{cases}$$

- $M_{\text{subst}}$ - substitution matrix of the scoring system;
- $G$ - gap penalty.

The recursive formula allows the calculation of dynamic programming matrix starting from smaller subalignments.

All elements of the matrix $S(i,j)$ correspond to the optimal scores of partial alignments.

The score $S(m,n)$, where $m$ and $n$ are two sequence lengths, is the optimal global alignment score.

The optimal alignment can be retrieved by backtracking of all moves that have led to $S(m,n)$. 

Diagram:

- SeqA
- SeqB
- Sequence A
- Sequence B
- $S(i-1,j)$
- $S(i,j)$
- $S(i,j-1)$
- $S(i-1,j-1)$
- $S(m,n)$
- Backtracking moves
A toy example (here match is +5; mismatch is -2 and insertion/deletion is -6):
**Needleman - Wunsch** algorithm:
optimal **global** alignment using dynamic programming.

**Waterman - Smith** algorithm:
optimal **local** alignment using dynamic programming.

*Optimal local alignment is defined as the alignment of subregions of two sequences with the maximum score.*

*NB. It does not mean that such subregions are aligned separately, they are identified by the dynamic programming matrix constructed for full-length sequences.*

The programs exploiting these algorithms are available via the ENTREZ system and in the EMBL-EBI tools.
Needleman - Wunsch algorithm: optimal \textit{global} alignment using dynamic programming.

Waterman - Smith algorithm: optimal \textit{local} alignment using dynamic programming.

\textit{Optimal local alignment is defined as the alignment of subregions of two sequences with the maximum score.}
\newline\textit{NB. It does not mean that such subregions are aligned separately, they are identified by the dynamic programming matrix constructed for full-length sequences.}

The programs exploiting these algorithms are available via the ENTREZ system and in the EMBL-EBI tools.

\textit{“Glocal” alignment: global on one of the sequences, local on the other.}

\textit{(Reasonable in some cases, e.g. when one of the sequences is expected to be homologous to a domain within the other).}

Can be computed e.g. with zero gap penalties at the ends of one of the sequences.
Estimates of alignment significance:

**Global alignments:** no accurate statistical theory. The reliability of an alignment can be estimated using multiple alignments of permutations of aligned sequences. If the score of alignment of interest is significantly higher than the average score obtained from pairs of sequences of the same lengths and compositions (permutations), it is judged to be significant rather than determined by chance alone.

Say, the optimal global alignment of seqA and seqB has the score $S_{AB}$.

seqA = {GAGCTAA...}
seqB = {GCAAGCC...}

\[
\text{permutations of sequences, like e.g.} \\
\text{perm\{12345\} -> \{32451\} or \{43152\} etc.}
\]

Alignment score [ perm(seqA), perm(seqB) ] --> $S_1$

\[
\text{repeat e.g. 100 times}
\]

Average ($S_1, S_2, ... S_{100}$) = $S_{avg}$

If $S_{AB}$ is significantly higher than $S_{avg}$, the alignment is significant.
Estimates of alignment significance:

**Local alignments:** expected number (E-value) of ungapped local alignments with score at least $S$ in the alignment of sequences with sufficiently large lengths $m$ and $n$:

$$E = K \cdot m \cdot n \cdot \exp(-\lambda S),$$

where $K$ and $\lambda$ depend on scoring system and monomer frequencies.

No general theory for gapped alignments. Statistics can be estimated using quasi-random sequences.
Sequence database similarity search

- Input: sequence query

- Output: list of similar sequences ("hits") found in the database
- Sequence database similarity search implies pairwise alignments of the query to all entries in the database.

- A straightforward dynamic programming algorithm is not efficient in this case (slow).

- A faster search can be realized using search for “words”: stretches of similar oligomers in two sequences (the query and a subject sequence from the database).
BLAST: Basic Local Alignment Search Tool

BLAST is the most popular program for sequence database similarity search.

First publication: Altschul et al. (1990).

Main strategy:

- Searching for “words” in a subject sequence from the database satisfying a criterion of a word size at least $W$ and a score ($S$) at least $T$ compared to a word in the query.

- If a word is found, BLAST algorithm attempts to extend it and improve the score $S$.

- The algorithm is designed for local alignments: if further extension does not improve $S$, the alignment region between the query and the subject sequence (“sequence hit”) with the maximal $S$ is returned to the user.

- The result of BLAST is a list of hits, ordered according to their significance (E-values).
Say, searching with a query: ...FDRIGDGETKLVTPVPT...

“w-mers”: words that score at least T when compared to some word (e.g. VTP) in the query.

With W=3; T=11 and BLOSUM62 matrix, w-mer scores calculated for VTP:

<table>
<thead>
<tr>
<th>w-mer</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTP</td>
<td>16</td>
</tr>
<tr>
<td>MTP</td>
<td>13</td>
</tr>
<tr>
<td>CTP</td>
<td>11</td>
</tr>
<tr>
<td>VSP</td>
<td>12</td>
</tr>
<tr>
<td>VVP</td>
<td>11</td>
</tr>
<tr>
<td>ITP</td>
<td>15</td>
</tr>
<tr>
<td>ATP</td>
<td>12</td>
</tr>
<tr>
<td>FTP</td>
<td>11</td>
</tr>
<tr>
<td>VAP</td>
<td>11</td>
</tr>
<tr>
<td>LTP</td>
<td>13</td>
</tr>
<tr>
<td>TTP</td>
<td>12</td>
</tr>
<tr>
<td>YTP</td>
<td>11</td>
</tr>
<tr>
<td>VNP</td>
<td>11</td>
</tr>
</tbody>
</table>

Subject ...VDQHGAPPEQRITPQOQ...

contains ITP (S=15) => the algorithm proceeds with the extension phase (e.g. alignment by dynamic programming)

Query ...FDRIGDGETKLVTPVPT...
Sbjct ...VDQHGAPPEQRITPQOQ...

Score improved?
Word extension search in the original BLAST algorithm

(Altschul et al., 1990)

- **HSP**, high-scoring segment pair
- **S**: minimum score to return a hit in the output
- **X**: significance decay
- **T**: word threshold
The statistics of pairwise alignments

Expected number (E-value) of ungapped HSPs with score at least \( S \) in the alignment of sequences with sufficiently large lengths \( m \) and \( n \):

\[
E = K m n \exp(-\lambda S),
\]

where \( K \) and \( \lambda \) depend on scoring system and monomer frequencies.

Normalized raw score

\[
S' = (\lambda S - \ln K) / \ln 2
\]

is a “bit score” characterizing HSP significance: \( E = m n 2^{-S'} \)
(not dependent on scoring system).

For gapped local alignments the statistics can be determined from large-scale comparisons of quasi-random sequences.
**Two-hit approach**: initial search for two non-overlapping hits of score at least $T$, within a distance $A$ of one another on a diagonal in sequence space.

*Advantage: faster search without losing significant sequence similarities.*

Two-hit approach: initial search for two non-overlapping hits of score at least $T$, within a distance $A$ of one another on a diagonal in sequence space:

- $S \geq T$
- $S' \geq S_g$

Ungapped extension:

If ungapped extension is better than some threshold $S_g$. E.g. chosen so that not more than one gapped extension is invoked per 50 database sequences, corresponding to $S_g = 22$ bits:

Gapped extension is triggered.
BLAST webpage
Multiple sequence alignment

Multiple: $N > 2$

$N=5:$

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Alignment with respect to Hfq_Bsubtilis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hfq_Bsubtilis</td>
<td>--MKPINIQDQFLNQIRKENTYVTVFLLNGFQRQVKGFDNFTVLLESEGKQQLIYKHA</td>
</tr>
<tr>
<td>Hfq_Lpneumophila</td>
<td>-MSKNLQDQFNLNELRKEKVPVSVFLVNGIKLHGIIDSFDQYVVMLKN-SITQMVYKHA</td>
</tr>
<tr>
<td>Hfq_Ecoli</td>
<td>-MAKGQSLQDPFLNALRRERVVPVISYLVNGIKLQGQIESFDQVFVILLKN-TVSQMVYKHA</td>
</tr>
<tr>
<td>Hfq_Ngonorrhoeae</td>
<td>MTAKQMLQDPFLNALRKEHVPVISYLVNGIKLQGQVESFDQYVVLLRNTSVTQMVYKHA</td>
</tr>
<tr>
<td>Hfq_Neuropaea</td>
<td>MGVKQQLQDPFLNILRKERIPVISYLVNGIKLQGQIDSFDQYVVLLKN-SVTQMVYKHA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Alignment with respect to Hfq_Bsubtilis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hfq_Bsubtilis</td>
<td>ISTFAPQKNVQLE--------------------------</td>
</tr>
<tr>
<td>Hfq_Lpneumophila</td>
<td>ISTVVPVMVKEAESSGEEGTVAD------------------</td>
</tr>
<tr>
<td>Hfq_Ecoli</td>
<td>ISTVVPVRVSHPHNSNAGGTSSNYHGSAQNTSQAQDSTE</td>
</tr>
<tr>
<td>Hfq_Ngonorrhoeae</td>
<td>ISTIVPARSVNLQHENKPQAPASTL----VQVETVQOAPAE-</td>
</tr>
<tr>
<td>Hfq_Neuropaea</td>
<td>ISTIVPAKASIPIPADTQTEQDEP-----------------</td>
</tr>
</tbody>
</table>

- An accurate alignment of multiple sequences by direct application of dynamic programming is not feasible (computationally demanding, could be applied only for small datasets of relatively short sequences).
- Various MSA strategies are used for faster algorithms. One of the most straightforward ones: progressive multiple sequence alignment.
Progressive multiple sequence alignment

Progressive MSA: an algorithm starts from aligning the closely related sequences, with following iterations consisting of aligning the previously built alignments. At every iteration, a pairwise alignment of two clusters of sequences is carried out.

ClustalW
(Thompson et al., 1994):

Pairwise alignment: Calculate distance matrix

Unrooted Neighbor-Joining tree

Rooted NJ tree (guide tree) and sequence weights

Progressive alignment: Align following the guide tree
Using sequence weights in progressive multiple alignment

- E.g. ClustalW algorithm:

![Rooted tree with branch lengths](image)

\[ W(1) = 0.1 + \frac{0.2}{2} + \frac{0.1}{4} = 0.225 \]
\[ W(2) = 0.2 + \frac{0.2}{2} + \frac{0.1}{4} = 0.325 \]
\[ W(3) = 0.3 + \frac{0.1}{2} + \frac{0.1}{4} = 0.375 \]
\[ W(4) = 0.25 + \frac{0.1}{2} + \frac{0.1}{4} = 0.325 \]
\[ W(5) = \frac{0.5}{1} = 0.5 \]

The contributions of branch lengths to the weights are divided by cluster sizes.

- Sequences are aligned according to the tree order. At each step, dynamic programming is used for pairwise alignment of (clusters of) sequences. The substitution scores are calculated as weighted averages of scores for substitutions between clusters.

\[ S(i,j) = S(E,K) \times W(1) \times W(3) + \]
\[ + S(E,R) \times W(1) \times W(4) + \]
\[ + S(K,K) \times W(2) \times W(3) + \]
\[ + S(K,R) \times W(2) \times W(4) \]

For instance, alignment of two clusters \{seq1,seq2\} vs \{seq3,seq4\}:

...VLLESEGKQL... seq1
...VMLKN-SITQM... seq2
.....(i)..........
Multiple sequence alignment: various strategies

e.g. MUSCLE (R.C. Edgar, 2004):

1.1 k-mer counting

1.2 UPGMA

1.3 progressive alignment

2.1 compute %ids from MSA1

Kimura distance matrix D2

2.2 UPGMA

2.3 progressive alignment

3.1 delete edge from TREE2 giving 2 subtrees

3.2 compute subtree profiles

3.3 re-align profiles

3.4 SP score better?

No, delete

Yes, save

repeat

MSA1

TREE1

k-mer distance matrix D1

MSA2

MSA3
Multiple sequence alignment: various strategies

E.g., T-Coffee
(Notredame et al., 2004):

- Weighting
  - Signal Addition

  PRIMARY LIBRARY

  EXTENSION

  EXTENDED LIBRARY

  PROGRESSIVE ALIGNMENT

  \[
  \begin{array}{c}
  \text{ClustalW Primary Library} \\
  \text{(Global Pairwise Alignment)} \\
  \end{array}
  \]

  \[
  \begin{array}{c}
  \text{Lalign Primary Library (Local Pairwise Alignment)} \\
  \end{array}
  \]

  Lalign local: 10 top-scoring non-intersecting local alignments

- Monomer \(x(A)\) aligned to \(y(B)\): constraint.
  - \(W(\text{constraint}) = \%\) similarity in the alignment.
  - \(W(x,y) = W(x,y, \text{global}) + W(x,y, \text{local})\).
  - \(W = 0\) if \(x\) and \(y\) are not aligned.

- If \(x(A)\) is aligned to \(z(C)\) and \(y(B)\) is aligned to \(z(C)\) as well: \(x\) and \(y\) are aligned through sequence \(C\), thus additional constraint weight:
  \(W(x,y) = W(x,y) + \min [W(x,z) + W(y,z)]\).

- Progressive alignment according to the NJ tree from pairwise alignments.
  - Dynamic programming is carried out with account of weights.
  - No gap penalties (indirectly they are already taken into account).
Alignment-free sequence comparisons (k-mer analysis)

Similar sequences have similar “word” compositions.

*Words: L-tuples or k-mers.*

Comparisons of these compositions can be done faster than those based on alignments.

*Important for large datasets of sequences.*

Say, sequences $X = \text{AAACTGGT...}$ -> 6-mers: $\text{AAACTG, AACTGG, ACTGGT, ...}$

$Y = \text{AGAAGCTGG...}$ -> $\text{AGAACT, GAACTG, AACTGG, ...}$

$Z = \text{AAATTGGT...}$ -> $\text{AAATTG, AATTGG, AATTGG, ...}$

$\Rightarrow$ In this region $X$ and $Y$ share one 6-mer, $Z$ is different.

For a given $k$, a sequence $X$ can be converted into vector $C(X,k) = (c_{X,k,1}, c_{X,k,2}, \ldots c_{X,k,N})$

($c_{X,k,i}$ is word count for the $i$-th k-mer and $N$ is number of all possible k-mers.)

Different metrics for a distance $d(X,Y)$ are possible, for instance:

$$d(X,Y) = \sum_{i=1}^{N} (c_{X,k,i} - c_{Y,k,i})^2$$

(Euclidean distance)

or fractional common $k$-mer count $F$:

$$F(X,Y) = \sum_{i=1}^{N} \frac{\min(c_{X,k,i}, c_{Y,k,i})}{[ \min (\text{lengthX, lengthY}) - k + 1 ]}$$

(Here an upper limit of homologous $i$-th $k$-mers in two sequences is normalised by the maximum number of homologous $k$-mers. This value decreases with increasing evolutionary distance).