Database searching and alignment statistics

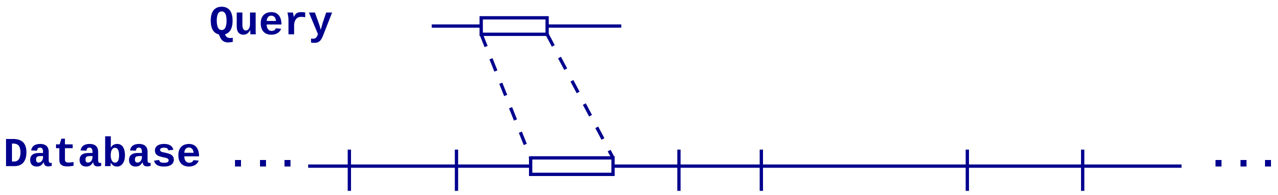
Waterman M.S. (1995) Introduction to Computational Biology.

... and some additional notes on our class home page.

The classical set-up of a molecular database search

- Given are a query sequence and a database of sequences of the same type (Protein or DNA).
- For each entry of the database a local alignment to the query is calculated and the corresponding score is stored.
- The database entries are listed in decreasing order of their scores.
- Significantly high scores indicate evolutionary relationships (Homology).

Database Search



Application

- The query is a sequence where nothing is known about.
- The database search hopefully yields a set of homologous sequences.
- If there function is known and annotated in the database, we have a clue to the function of the query as well.

Algorithms

- Database searches can be done using the Smith-Waterman algorithm. (Good choice!)

- For impatient users, there are faster heuristic algorithms, which approximate the local alignment result.

- The most famous are BLAST and FASTA.

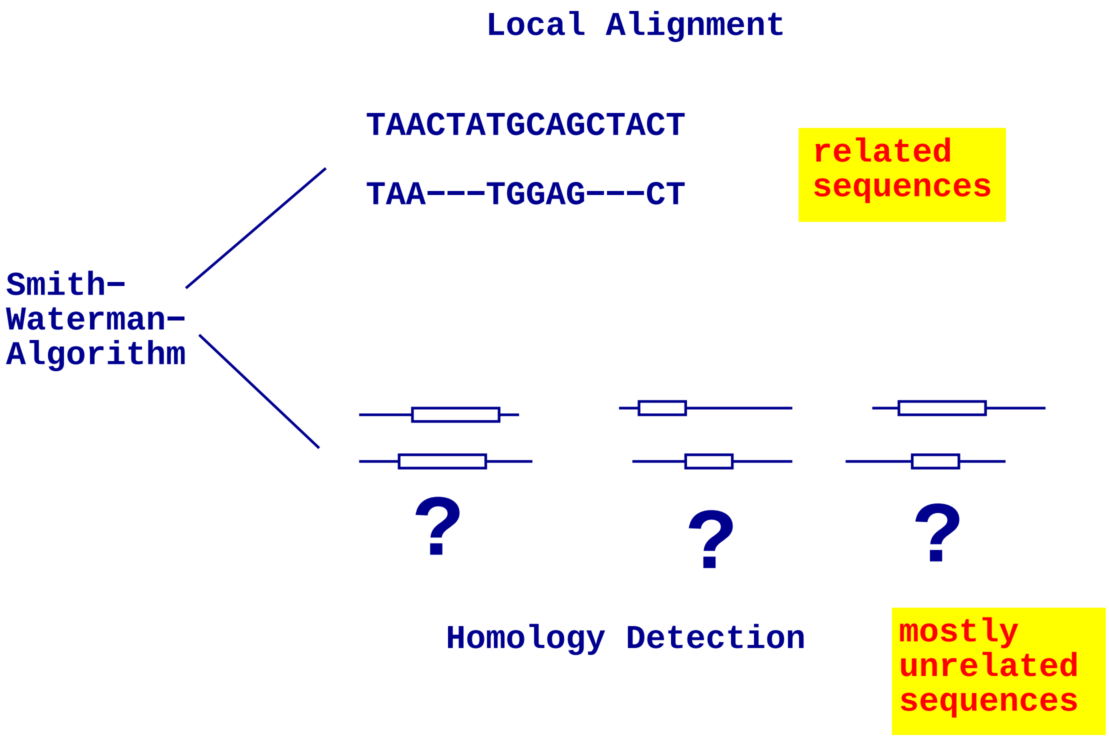
- BLAST produces a set of suboptimal ungapped local alignments and adds the scores.

- FASTA and GAP-BLAST produce local alignments with gaps, however these alignments are not always optimal.

- See Gusfield's book, for a description of the algorithms.

- PSI-Blast is different ... later.

Two completely different applications of the Smith-Waterman algorithm



Local Alignment

```
TAACTATGCAGCTACT
TAA---TGGAG---CT
```

related sequences

Smith-
Waterman-
Algorithm

? ? ?

Homology Detection

mostly unrelated sequences

- The Smith-Waterman algorithm was designed for local sequence alignments.

- Its purpose is to detect and align conserved segments in pairs of related sequences.

- In a database search most of the sequences that we align are not related at all

- Strictly speaking, this is an abuse of the method.

- What happens, if we align unrelated sequences?

```
The best scores are:                                   (len) s-w
CYC_HUMAN  CYTOCHROME C.                               (104) 532
CYC_MACMU  CYTOCHROME C.                               (104) 527
CYC_ATESP  CYTOCHROME C.                               (104) 510
CYC_BOVIN  CYTOCHROME C.                               (104) 502
CYC_RABIT  CYTOCHROME C.                               (104) 501

                                  *

CYC_CHESE  CYTOCHROME C.                               (104) 475

                                  *
                                  *
                                  *

CYC6_MONLU CYTOCHROME C6 (SOLUBLE CYTOCHROME F) (C ( 83)  70
TR2M_AGRT4 TRYPTOPHAN 2-MONOOXYGENASE (EC 1.13.12. (755)  69
C555_PROAE CYTOCHROME C555.                        ( 99)  69
CYC6_BUMFI CYTOCHROME C6 (SOLUBLE CYTOCHROME F) (C ( 86)  68
ACM5_RAT   MUSCARINIC ACETYLCHOLINE RECEPTOR M5.   (531)  68
YSXC_BACSU HYPOTHETICAL 22.0 KD PROTEIN IN LON-HEM (195)  68
VG02_BPT4  TERMINAL DNA PROTECTING PROTEIN (HEAD P (274)  67
YIF0_YEAST HYPOTHETICAL 32.0 KD PROTEIN IN SNP1-ME (285)  67
C771_SOLME CYTOCHROME P450 LXXVIIA1 (EC 1.14.14.1) (499)  67
KINC_BACSU SPORULATION KINASE C (EC 2.7.3.-).      (428)  67
CYC6_PORTE CYTOCHROME C6 (SOLUBLE CYTOCHROME F) (C ( 85)  67
CYC6_ANASQ CYTOCHROME C6 PRECURSOR (SOLUBLE CYTOCH (111)  67

                                 *
                                 *
                                 *

KEK4_MOUSE TYROSINE-PROTEIN KINASE RECEPTOR MEK4 P (983)  52
YKA8_CAEEL HYPOTHETICAL 27.3 KD PROTEIN B0303.8 IN (239)  52
IF3T_TORCA TYPE III INTERMEDIATE FILAMENT.         (458)  52
TRT1_SALTY TRAT COMPLEMENT RESISTANCE PROTEIN PREC (243)  52
C550_PSEST CYTOCHROME C551 PRECURSOR.              (104)  52
TSP4_XENLA THROMBOSPONDIN 4 PRECURSOR.             (955)  52
RPOP_AGABT PROBABLE DNA-DIRECTED RNA POLYMERASE (E (1102) 52
VG16_BPML5 PROBABLE HEAD ASSEMBLY PROTEIN (SCAFFOL (172)  52
BMP8_MOUSE BONE MORPHOGENETIC PROTEIN 8 PRECURSOR  (399)  52
104K_THEPA 104 KD MICRONEME-RHOPTRY ANTIGEN.       (924)  52
UREC_HELPY UREASE OPERON UREC PROTEIN.             (445)  67
```

Figure 1: Result of searching human *cytochrome c* against SWISs-PROT release 32.

Daylight, Twilight, Midnight

1. On top of the list are clear hits. These sequences are almost identical and their scores are far above average. Bright day light zone

2. Further down in the list we observe a twilight zone, consisting of both sequences that are distantly related to the query and sequences that are unrelated to it. However, they all share about the same amount of similarity to the query (if measured in alignment score points).

3. Finally all the other sequences of the database are listed with small scores. Midnight zone

- Although alignment algorithms are designed to detect conserved regions, they produce alignments and scores also when applied to sequences that are not related at all.

- In molecular database searches, most comparisons are comparisons between unrelated sequences.

- Random similarities are frequently observed that score higher than those arising from distant relationships.

Questions

- How different can homologous sequences be?
  Almost arbitrarily different.

- How similar do sequences need to be such that we can conclude that they are homologous?
  A very complicated biological question.

- Given an alignment, is it more likely that it is real or just chance.
  A statistical question:
  That is question we are actually interested in.
  But it soon becomes biological again, because we need a model for the similarity of related sequences
  (To complex to be modeled ?!).

- What is the probability that an alignment of random sequences results in a score higher or equal than t The complementary event is that a random score is less that t. Classical P-Values ... not what we are really interested in ... but useful in this case.

A rule of thumb ... which is questionable

- Sometimes you can read something like: Sequences with more than 30% identity are homologous and have similar folds and functions.
- However, if you take a sequence and randomly mutate 70% of its positions ... pretty sure you will end up with a sequence, that does not have the same fold and function.
- However, sequence pairs with more than 30% identity, that we find in nature mostly have a common fold and function.
- Where is the difference?

- The number of different sequences that you can get by randomly mutating a given sequence in 70% of its position is huge.
- However, almost all these sequences do not exist in nature, and hence we do not observe them. We only observe what is in the databases and that is much less.
- The 30% rule is based on experience with the data that is available.
- If you go below this value, you start observing random similarities.

Homology search is based on a statistical rationale

- If a similarity between two sequences is very unlikely to occur by chance, we conclude that it has an evolutionary basis.

Some more remarks to the 30% rule:

- We want to evaluate the evidence of homology ...
- ... and percentage identity is not a good measure for this.
- We have local alignments ... so how long does the alignment with 30% identity need to be?
- It is better to use local alignment scores.
- The rule is from experience with the data we have today (had yesterday), databases grow and random scores will become more likely ... hence, it might be well possible that we will make different experiences with more data.

The problem of random similarities between molecular sequences eventually got into the hands of mathematicians:

- A. Dembo
- S. Karlin
- O. Zeitouni
- R. Arratia
- L. Gordon
- M. S. Waterman
- ...and others.

For references see the notes on the web.

i.i.d. models

- We want to learn about the local alignment score of unrelated sequences.
- We model pairs of unrelated sequences by pairs of random sequences.
- Assume all positions in both sequences are independent and sampled from the same distribution on the alphabet A.
- Further, assume the sequences are independent from each other.
- We also assume that the sequences are infinitely long.
- i.i.d. sequences (independently, identically, distributed)

What can be found in long random sequences

- Think of the database as a infinite i.i.d. sequence over some Alphabet A.

$$X_1, X_2, \ldots$$

- Each position is distributed according to $(\mu_1, \ldots \mu_l)$. We assume that $\mu_i > 0$: every letter actually occurs.
- A may be the alphabet of amino acids, or just all ascii-characters.
- Now, think of some fixed text $s = (x_1, \ldots, x_k)$ of length k.
- s may be your first name, the human hemoglobin alpha chain, or the entire works of Shakespeare.
- Can we find this text in the random sequence ?

For all $n \in \mathbf{N}$ we have that the probability that s starts at position n is

$$\begin{aligned} p_s : &= P[X_{n+1} = x_1, \ldots, X_{n+k} = x_k] \\ &= \mu(x_1), \mu(x_2), \ldots, \mu(x_k) > 0 \end{aligned}$$

We are interested in finding an exact match of the word s at some place in the sequence $X_1, X_2, \ldots$.

$$P[\texttt{The word } s \texttt{ is included in } X_1, X_2, \ldots]$$

$$\begin{aligned} &= P[\texttt{There is an } n \texttt{ such that } X_{n+1}=x_1,\ldots,X_{n+k}=x_k] \\ &\geq P[\texttt{There is an } n \texttt{ such that } \underbrace{X_{nk+1}=x_1,\ldots,X_{(n+1)k}=x_k]}_{=:B_n} \end{aligned}$$

We have divided the sequence $X_1, X_2, \ldots$ into disjoint blocks of length k such that the events B_n are independent.

$$\underbrace{X_1, X_2, \qquad \cdots \qquad , X_k}_{k}, \underbrace{X_{k+1}, \qquad \cdots \qquad , X_{2k}}_{k}, X_{2k+1}, \quad \cdots$$

Thus

$$P[\text{There is an } n \text{ such that } \underbrace{X_{nk+1} = x_1, \ldots, X_{(n+1)k} = x_k}_{=: B_n}]$$

equals

$$\begin{aligned} P\left[\bigcup_{n\geq 0} B_n\right] &= 1 - P\left[\bigcap_{n\geq 0} B_n^c\right] \\ &= 1 - \lim_{m\to\infty} P\left[\bigcap_{0\leq n\leq m} B_n^c\right] \\ &= 1 - \lim_{m\to\infty} \prod_{0\leq n\leq m} P[B_n^c] \end{aligned}$$

Due to

$$P[B_n^c] = 1 - P[B_n] \le e^{-P[B_n]}$$

we have finally

$$\begin{aligned} & P[\text{The word } s \text{ is included in } X_1, X_2, \ldots] \\ \ge\ & 1 - \lim_{m\to\infty} \exp\left(- \sum_{0\le n\le m} P[B_n] \right) \\ =\ & 1 - \lim_{m\to\infty} \exp(-(m+1)p_s) \\ =\ & 1. \end{aligned}$$

Hence, with probability one, the sequence contains

- Your first name, my first name, the first name of any person in the world.
- The human hemoglobin sequences, and all other known protein sequences.
- The complete works of Shakespeare, in any language, including languages where no translation exists yet, and for each of those you can have as many copies as you like.

- If you started reading a random sequence, what would you expect to find first, your first name or some Shakespeare?

- Both texts are in the sequence, but the expected waiting times are very different.

- Real sequences are finite and the waiting times might be longer than there length. In this case we do not find the word at all.

- What do we expect to find at all, if the sequence is long but finite?
- If we do not find an exact match of the word s, do we find at least a similar string?
- Do we find a high scoring local alignment of s and the database?
- How high scoring do we expect it to be?
- This obviously depends on the word s.
- It also depends on the length of the random sequence.
- The longer we can search, the more chances we have for finding something.