Multiple Sequence Alignment

Jiří Kléma

Department of Computer Science,
Czech Technical University in Prague

Lecture based on Mark Craven's class at University of Wisconsin

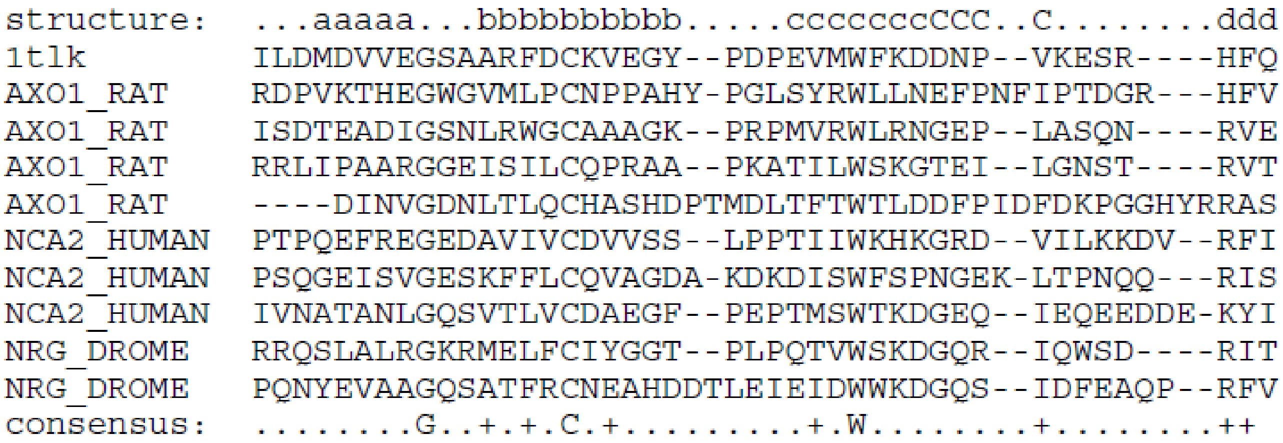
http://cw.felk.cvut.cz/wiki/courses/b4m36bin/start

Overview

- Multiple sequence alignment (MSA)
  - the algorithmic task,
  - biological motivation
    - why pairwise alignment is sometimes not enough,
- what is needed to score an alignment of multiple sequences?
  - can be done in a similar way as in the pairwise case,
  - othe options abvailable too,
- optimal solution
  - dynamic programming,
  - not truly applicable for larger sequence sets.
- heuristic solutions
  - progressive alignment,
  - statistical approaches (hidden Markov models, fast Fourier transform).

Multiple sequence alignment: task definition

- Given
  - a set of k sequences, $k > 2$,
  - a method for scoring an alignment,
- Do
  - determine the correspondences between the sequences such that the alignment score is maximized,
- Example



```
structure:   ...aaaaa...bbbbbbbbbb.....cccccccCCC..C........ddd
1tlk         ILDMDVVEGSAARFDCKVEGY--PDPEVMWFKDDNP--VKESR----HFQ
AXO1_RAT     RDPVKTHEGWGVMLPCNPPAHY-PGLSYRWLLNEFPNFIPTDGR---HFV
AXO1_RAT     ISDTEADIGSNLRWGCAAAGK--PRPMVRWLRNGEP--LASQN----RVE
AXO1_RAT     RRLIPAARGGEISILCQPRAA--PKATILWSKGTEI--LGNST----RVT
AXO1_RAT     ----DINVGDNLTLQCHASHDPTMDLTFTWTLDDFPIDFDKPGGHYRRAS
NCA2_HUMAN   PTPQEFREGEDAVIVCDVVSS--LPPTIIWKHKGRD--VILKKDV--RFI
NCA2_HUMAN   PSQGEISVGESKFFLCQVAGDA-KDKDISWFSPNGEK-LTPNQQ---RIS
NCA2_HUMAN   IVNATANLGQSVTLVCDAEGF--PEPTMSWTKDGEQ--IEQEEDDE-KYI
NRG_DROME    RRQSLALRGKRMELFCIYGGT--PLPQTVWSKDGQR--IQWSD----RIT
NRG_DROME    PQNYEVAAGQSATFRCNEAHDDTLEIEIDWWKDGQS--IDFEAQP--RFV
consensus:   ........G..+.+.C.+.........+.W........+........++
```

Durbin et al.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.

Motivation for MSA

- determine evolutionary history of a set of sequences
  - at what point in history did certain mutations occur?
  - establish input data for phylogenetic analyses,
- discover a common motif in a set of sequences
  - e.g. DNA sequences that bind the same protein,
- characterize a set of sequences
  - e.g. a protein family,
  - build a (simplifying) profile model for such a set,
  - establish input data e.g. for a profile HMM,
- build profiles for sequence-database searching
  - PSI-BLAST generalizes a query sequence into a profile to search for remote relatives.

Scoring a multiple alignment

- ideally it should
  - be position-specific (some positions more conserved than others),
  - consider evolutionary relationships among sequences (a phylogenetic tree),
- in practice, many simplifying assumptions made
  - usually, the individual columns of an alignment considered independent

$$Score(m) = G + \sum_i S(m_i)$$

  - where G is a gap function and $S(m_i)$ the score of the i-th column,
- we will discuss two methods to estimate $S(m_i)$
  - sum of pairs (SP),
  - minimum entropy.

Scoring a multiple alignment: sum of pairs

- compute the sum of the pairwise scores

$$S(m_i) = \sum_{l<m} s(m_i^l, m_i^m)$$

  - m_i^l = character of the l-th sequence in the i-th column,
  - $s(a,b)$ = substitution score for a and b,
- seems to be perfectly natural, however
  - each sequence is scored as if it descended from the $k-1$ other sequences instead of a single ancestor,
  - does not perfectly fit with log-odds pairwise substitution scores

$$\log \frac{p_{abc}}{p_a p_b p_c} \neq \log \frac{p_{ab}}{p_a p_b} + \log \frac{p_{ac}}{p_a p_c} + \log \frac{p_{bc}}{p_b p_c} = \log \frac{p_{ab} p_{ac} p_{bc}}{p_a^2 p_b^2 p_c^2}$$

Scoring a multiple alignment: minimum entropy

- basic idea
  - characters in each column can be seen as a message,
  - columns that can be communicated using few bits are good,
  - try to minimize the entropy of each column,

$$S(m_i) = -\sum_a c_{ia} \log_2 p_{ia}$$

  - c_{ia} = count of character a in column i,
  - p_{ia} = probability of character a in column i,
- analogically, stems from a model that assumes
  - independent residues within the column as well as between columns, then the probability of a column m_i is

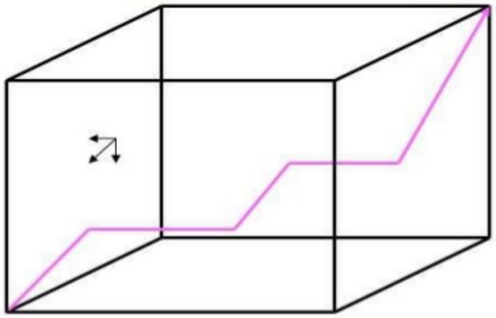
$$P(m_i) = \prod_a p_{ia}^{c_{ia}}$$

Dynamic programming approach

- can find optimal alignments using dynamic programming,
- generalization of methods for pairwise alignment
  - consider k-dimension matrix for k sequences (instead of 2-dimensional matrix),
  - each matrix element represents alignment score for k subsequences (instead of 2 subsequences)
    $\alpha_{i_1,i_2,\ldots,i_k}$ = the maximum score of an alignment up to the subsequences ending with $x^1_{i_1}, x^2_{i_2}, \ldots, x^k_{i_k}$,
- given k sequences of length n
  - space complexity is $\mathcal{O}(n^k)$.

Dynamic programming approach

$$\alpha_{i_1,i_2,\ldots,i_k} = max \begin{cases} \alpha_{i_1-1,i_2-1,\ldots,i_k-1} + s(x^1_{i_1}, x^2_{i_2}, \ldots, x^k_{i_k}) \\ \alpha_{i_1,i_2-1,\ldots,i_k-1} + s(-, x^2_{i_2}, \ldots, x^k_{i_k}) \\ \alpha_{i_1-1,i_2,\ldots,i_k-1} + s(x^1_{i_1}, -, \ldots, x^k_{i_k}) \\ \ldots \\ \alpha_{i_1,i_2,\ldots,i_k-1} + s(-, -, \ldots, x^k_{i_k}) \\ \ldots \end{cases}$$



Find a path through k-dimensional matrix

- Time complexity is
  - $\mathcal{O}(k^2 2^k n^k)$ if we use sum of pairs,
  - $\mathcal{O}(k 2^k n^k)$ if column scores can be computed in $\mathcal{O}(k)$ as with entropy.

Heuristic alignment methods

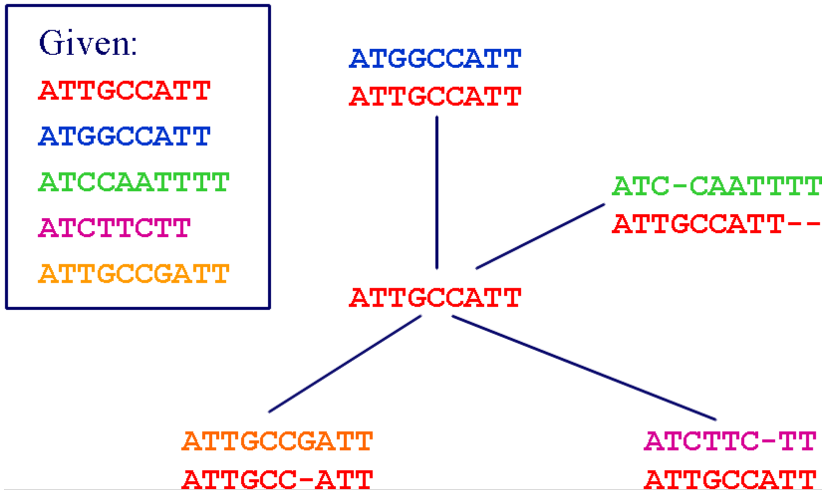
- DP approach is exponential in the number of sequences
  - heuristic methods used for larger k,
- **progressive alignment**
  - construct a succession of pairwise alignments,
  - star approach,
  - tree approaches, like CLUSTALW,
- iterative refinement
  - given a multiple alignment (say from a progressive method),
  - remove a sequence, realign it to profile of other sequences,
  - repeat until convergence.

Star alignment

- given: k sequences x_1, ..., x_k to be aligned
- do:
  - pick one sequence x_c as the "center"
    - either try each sequence as the center, pick the one with the best multiple alignment,
    - or compute all pairwise alignments and select x_c that maximizes $\sum_{i \neq c} sim(x_i, x_c)$,
  - for each $x_i \neq x_c$ determine an optimal alignment between x_i and x_c,
  - merge pairwise alignments
    - "once a gap, always a gap",
    - shift entire columns when incorporating gaps,
- return: multiple alignment resulting from aggregate.

Star alignment: example

- Pick the center and align against it



Given:
ATTGCCATT
ATGGCCATT
ATCCAATTTT
ATCTTCTT
ATTGCCGATT

ATGGCCATT
ATTGCCATT

ATC-CAATTTT
ATTGCCATT--

ATTGCCATT

ATTGCCGATT
ATTGCC-ATT

ATCTTC-TT
ATTGCCATT

- Merge pairwise alignments

	present pair	alignment
1.	ATGGCCATT ATTGCCATT	ATTGCCATT ATGGCCATT
2.	ATC-CAATTTT ATTGCCATT--	ATTGCCATT-- ATGGCCATT-- ATC-CAATTTT
3.	ATCTTC-TT ATTGCCATT	ATTGCCATT-- ATGGCCATT-- ATC-CAATTTT ATCTTC-TT--
4.	ATTGCCGATT ATTGCC-ATT	ATTGCC-ATT-- ATGGCC-ATT-- ATC-CA-ATTTT ATCTTC--TT-- ATTGCCGATT--

shift entire columns
when incorporating a gap

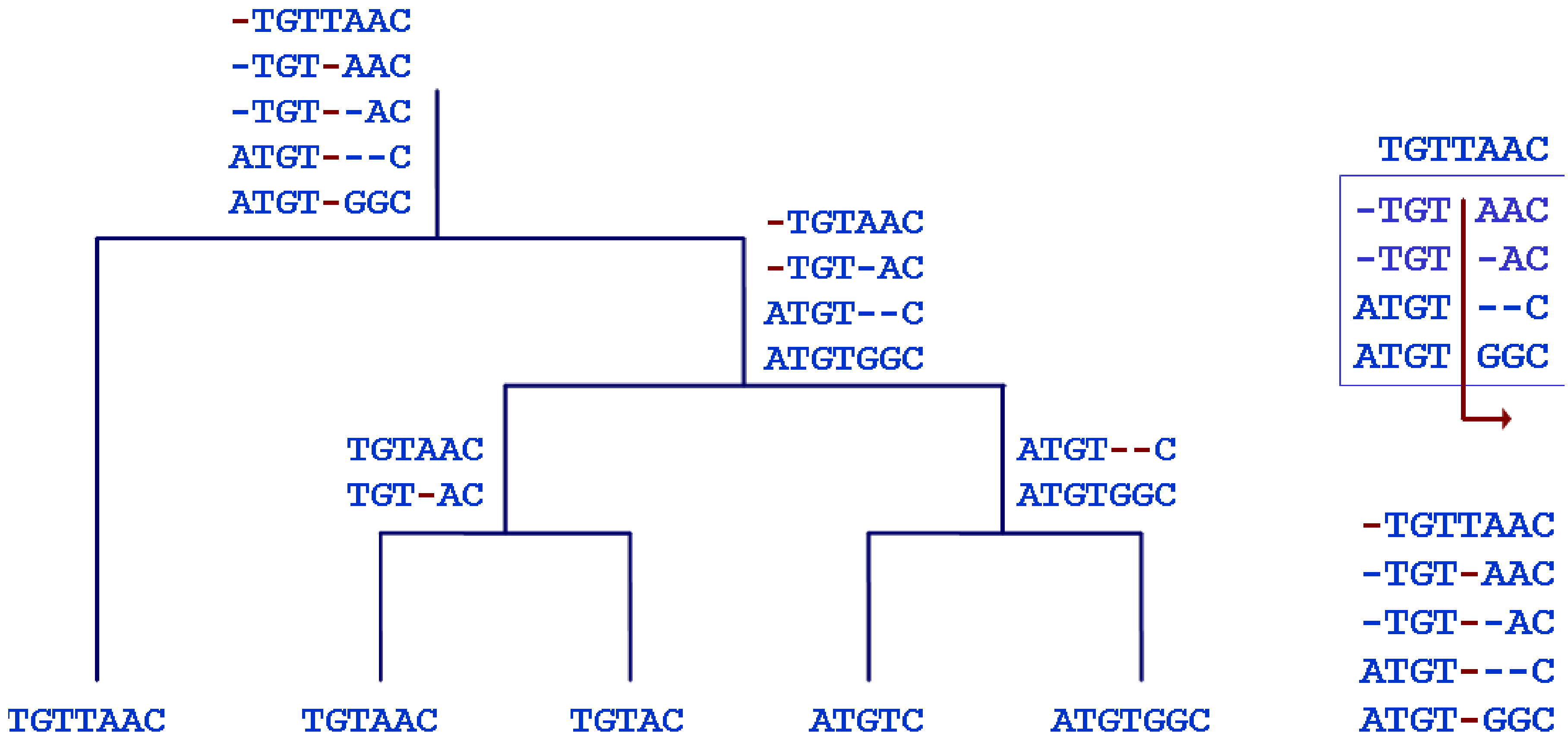
Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Tree alignments

- basic idea
  - organize multiple sequence alignment using a **guide tree**
    - leaves represent sequences,
    - internal nodes represent alignments,
- determine alignments from bottom of tree upward
  - return multiple alignment represented at the root of the tree,
- one common variant is the CLUSTALW algorithm [Thompson et al. 1994]
- progressive alignment in CLUSTALW
  - depending on the internal node in the tree, we may have to align
    - a sequence with a sequence,
    - a sequence with a profile (partial alignment),
    - a profile with a profile,
  - in all cases we can use dynamic programming
    - for the profile cases, use sum of pairs scoring.

Tree alignment: example

- The whole guide tree and one particular profile alignment
  - pairwise alignment, always shift entire columns.



```
-TGTTAAC
-TGT-AAC
-TGT--AC
ATGT---C
ATGT-GGC
```

```
-TGTAAC
-TGT-AC
ATGT--C
ATGTGGC
```

```
TGTAAC
TGT-AC
```

```
ATGT--C
ATGTGGC
```

```
TGTTAAC   TGTAAC   TGTAC   ATGTC   ATGTGGC
```

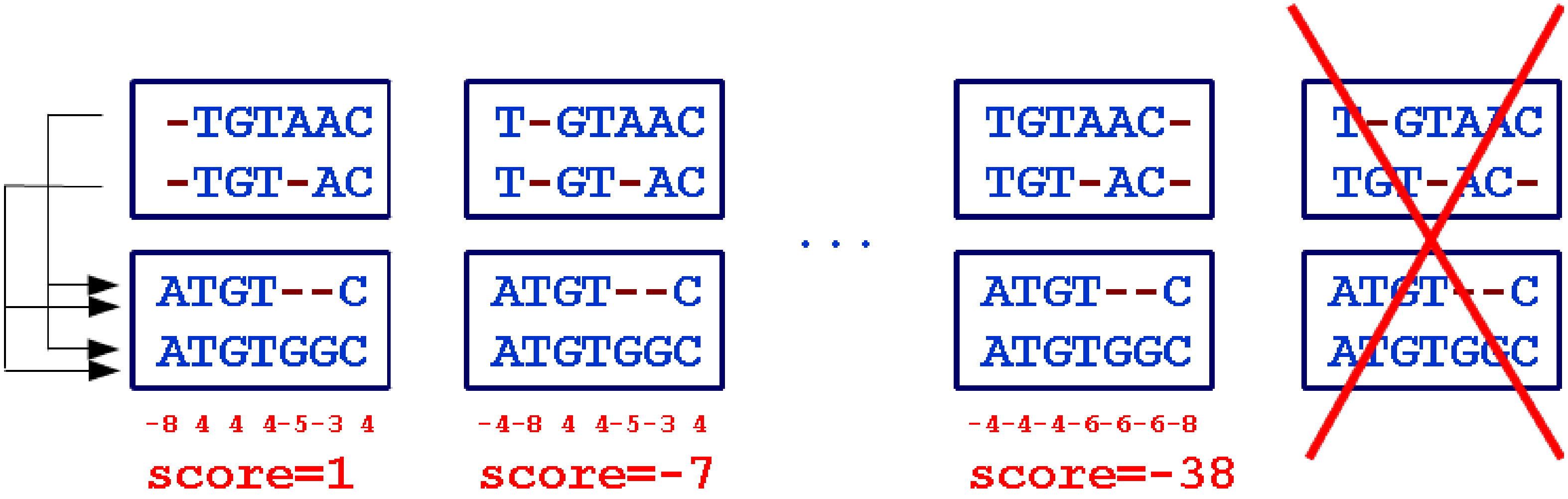
```
     TGTTAAC
-TGT AAC
-TGT -AC
ATGT --C
ATGT GGC
```

```
-TGTTAAC
-TGT-AAC
-TGT--AC
ATGT---C
ATGT-GGC
```

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Profile alignment: example

- Scoring scheme
  - if $x_i = y_j$ then $s(i,j)$=1 otherwise $s(i,j)$=-1, gap penalty = 2,
  - sum of pairs method,
- profiles never brake (no shifts inside of them)
  - the last alignment below is not allowed,
  - score only between profiles, within-profile scores remain constant.



-TGTAAC
-TGT-AC
ATGT--C
ATGTGGC
-8 4 4 4-5-3 4
score=1

T-GTAAC
T-GT-AC
ATGT--C
ATGTGGC
-4-8 4 4-5-3 4
score=-7

. . .

TGTAAC-
TGT-AC-
ATGT--C
ATGTGGC
-4-4-4-6-6-6-8
score=-38

T-GTAAC
TGT-AC-
ATGT--C
ATGTGGC

Multiple sequence alignment summary

- as with pairwise alignment, can compute local and global multiple alignments,
- dynamic programming is not feasible for most cases
  - heuristic methods usually used instead,
- some frequently used tools
  - Clustal Omega – progressive alignment that uses profile HMMs to model groups of sequences,
  - MAFFT – iterative method that uses Fast Fourier Transform,
  - T-Coffee – consistency-based method suitable for small alignments,
- alignment visualization and quality control
  - heuristic alignments often contain errors,
  - smaller alignments can be visually inspected and manually curated,
  - for larger alignments e.g., remove low quality alignment blocks.