Multiple Sequence Alignment

BMI/CS 576
www.biostat.wisc.edu/bmi576.html
Colin Dewey
cdewey@biostat.wisc.edu

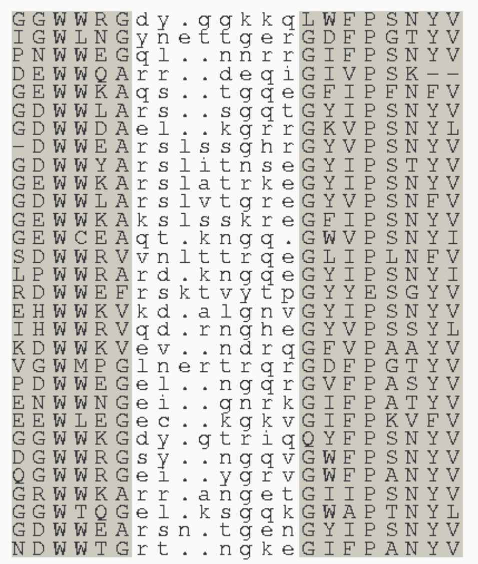
Multiple Sequence Alignment: Task Definition

- Given
  - a set of more than 2 sequences
  - a method for scoring an alignment
- Do:
  - determine the correspondences between the sequences such that the alignment score is maximized

Motivation for MSA

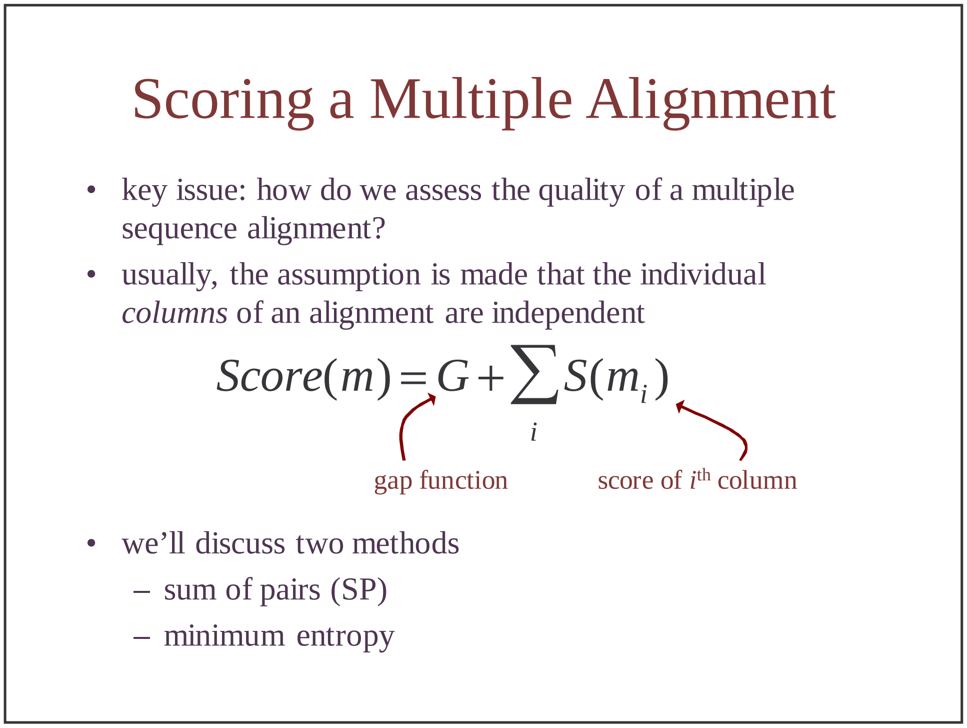
- establish input data for phylogenetic analyses
- determine evolutionary history of a set of sequences
  - At what point in history did certain mutations occur?
- discovering a common motif in a set of sequences (e.g. DNA sequences that bind the same protein)
- characterizing a set of sequences (e.g. a protein family)
- building *profiles* for sequence-database searching
  - PSI-BLAST generalizes a query sequence into a profile to search for remote relatives

Multiple Alignment of SH3 Domain



```
GGWWRGdy.ggkkqLWFPSNYV
IGWLNGynettgerGDFPGTYV
PNWWEGql..nnrrGIFPSNYV
DEWWQArr..deqiGIVPSK--
GEWWKAqs..tgqeGFIPFNFV
GDWWLArs..sgqtGYIPSNYV
GDWWDAel..kgrrGKVPSNYL
-DWWEArslssghrGYVPSNYV
GDWWYArslitnseGYIPSTYV
GEWWKArslatrkeGYIPSNYV
GDWWLArslvtgreGYVPSNFV
GEWWKAkslsskreGFIPSNYV
GEWCEAqt.kngq.GWVPSNYI
SDWWRVvnlttrqeGLIPLNFV
LPWWRArd.kngqeGYIPSNYI
RDWWEFrsktvytpGYYESGYV
EHWWKVkd.algnvGYIPSNYV
IHWWRVqd.rngheGYVPSSYL
KDWWKVev..ndrqGFVPAAYV
VGWMPGlnertrqrGDFPGTYV
PDWWEGel..ngqrGVFPASYV
ENWWNGei..gnrkGIFPATYV
EEWLEGec..kgkvGIFPKVFV
GGWWKGdy.gtriqQYFPSNYV
DGWWRGsy..ngqvGWFPSNYV
QGWWRGei..ygrvGWFPANYV
GRWWKArr.angetGIIPSNYV
GGWTQGel.ksgqkGWAPTNYL
GDWWEArsn.tgenGYIPSNYV
NDWWTGrt..ngkeGIFPANYV
```

Figure from A. Krogh, An Introduction to Hidden Markov Models for Biological Sequences



Scoring a Multiple Alignment

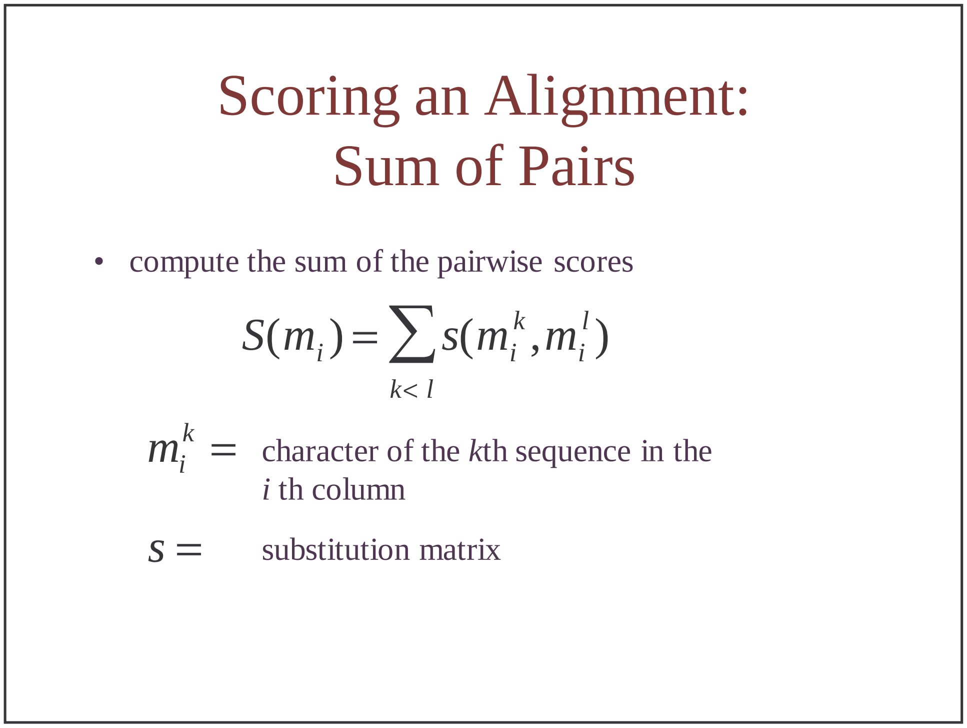
- key issue: how do we assess the quality of a multiple sequence alignment?
- usually, the assumption is made that the individual *columns* of an alignment are independent

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

gap function

score of i^{th} column

- we'll discuss two methods
  - sum of pairs (SP)
  - minimum entropy



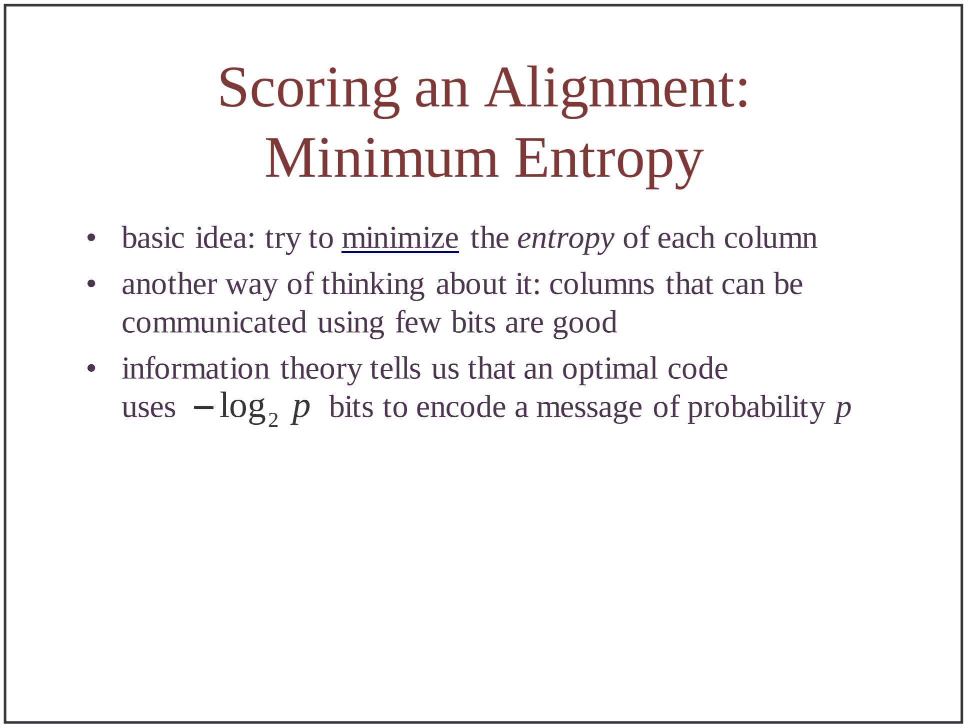
Scoring an Alignment: Sum of Pairs

- compute the sum of the pairwise scores

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

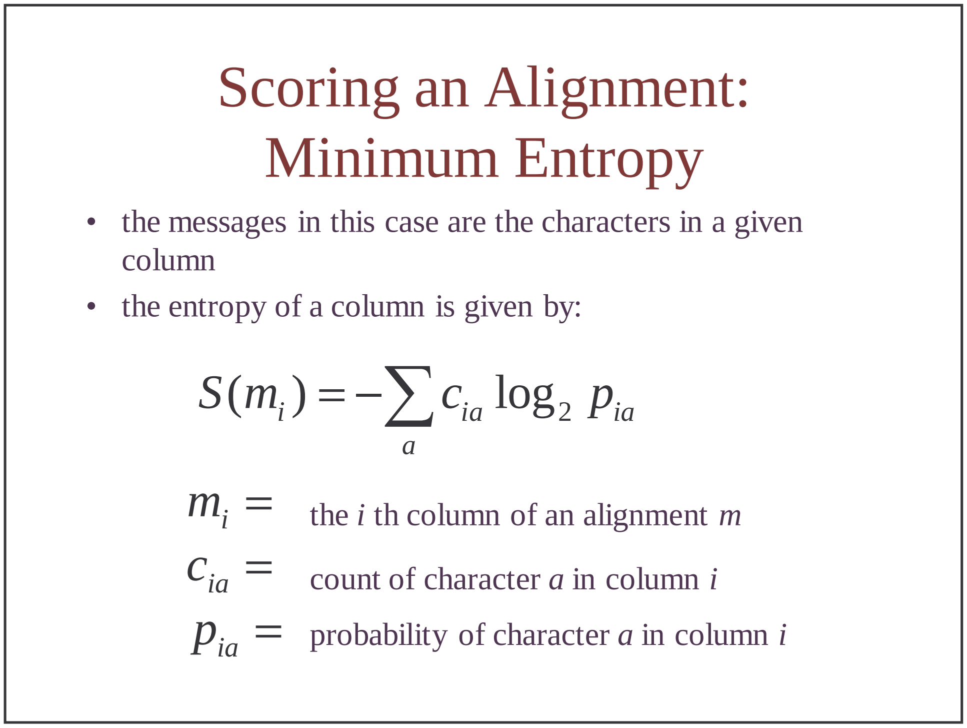
$m_i^k =$ character of the kth sequence in the i th column

$s =$ substitution matrix



Scoring an Alignment: Minimum Entropy

- basic idea: try to minimize the *entropy* of each column
- another way of thinking about it: columns that can be communicated using few bits are good
- information theory tells us that an optimal code uses $-\log_2 p$ bits to encode a message of probability *p*



Scoring an Alignment: Minimum Entropy

- the messages in this case are the characters in a given column
- the entropy of a column is given by:

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

$m_i =$ the *i* th column of an alignment *m*

$c_{ia} =$ count of character *a* in column *i*

$p_{ia} =$ probability of character *a* in column *i*

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)
- given *k* sequences of length *n*
  - space complexity is

$$O(n^k)$$

Dynamic Programming Approach

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

max score of alignment for subsequences $x^1_{i_1}, x^2_{i_2}, \ldots, x^k_{i_k}$

Dynamic Programming Approach

- given k sequences of length n
  - time complexity is

$O(k^2 2^k n^k)$ if we use sum of pairs

$O(k 2^k n^k)$ if column scores can be computed in $O(k)$, as with entropy

Heuristic Alignment Methods

- since time complexity of DP approach is exponential in the number of sequences, heuristic methods are usually used
- *progressive alignment*: construct a succession of pairwise alignments
  - star approach
  - tree approaches, like CLUSTALW
  - etc.
- 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 Approach

- given: *k* sequences to be aligned
  $x_1, \ldots, x_k$
  - pick one sequence x_c as the “center”
  - for each $x_i \neq x_c$ determine an optimal alignment between x_i and x_c
  - merge pairwise alignments
- return: multiple alignment resulting from aggregate

Star Alignments: Approaches to Picking the Center

Two possible approaches:

1. try each sequence as the center, return the best multiple alignment
2. compute all pairwise alignments and select the string x_c that maximizes:

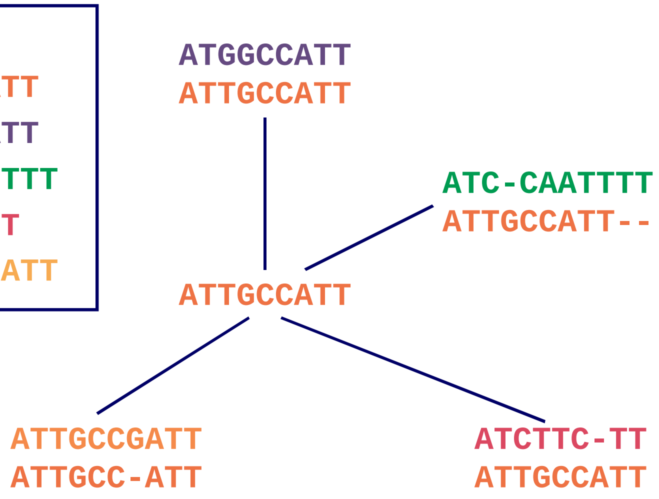
$$\sum_{i \neq c} \text{sim}(x_i, x_c)$$

Star Alignments: Aggregating Pairwise Alignments

- "once a gap, always a gap"
- shift entire columns when incorporating gaps

Star Alignment Example

Given:
ATTGCCATT
ATGGCCATT
ATCCAATTTT
ATCTTCTT
ATTGCCGATT



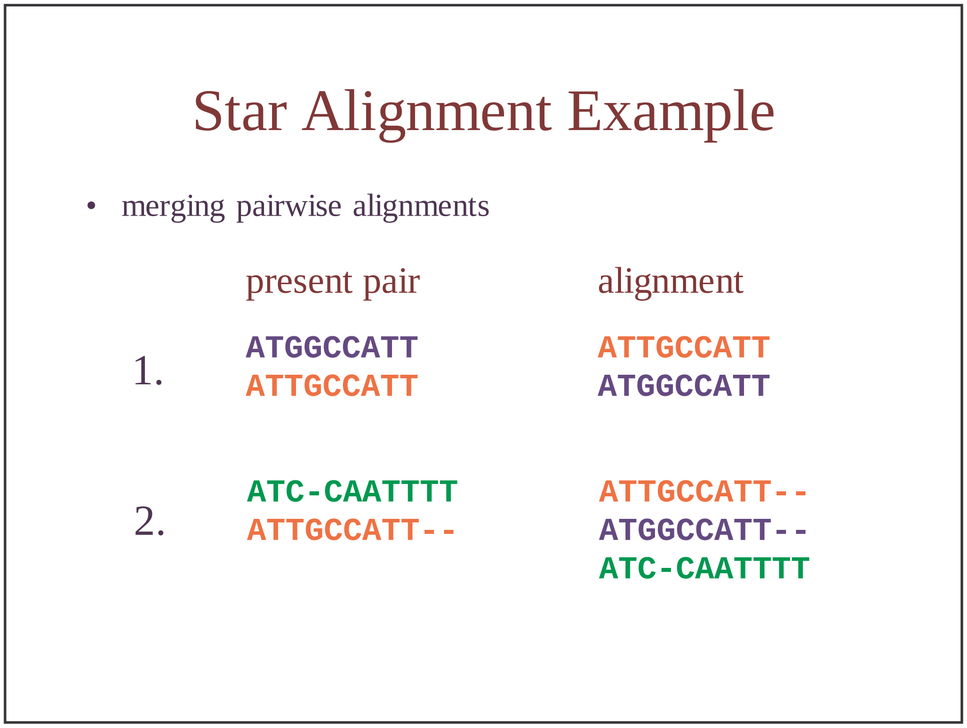
```
ATGGCCATT
ATTGCCATT
```

```
ATC-CAATTTT
ATTGCCATT--
```

```
ATTGCCATT
```

```
ATTGCCGATT
ATTGCC-ATT
```

```
ATCTTC-TT
ATTGCCATT
```



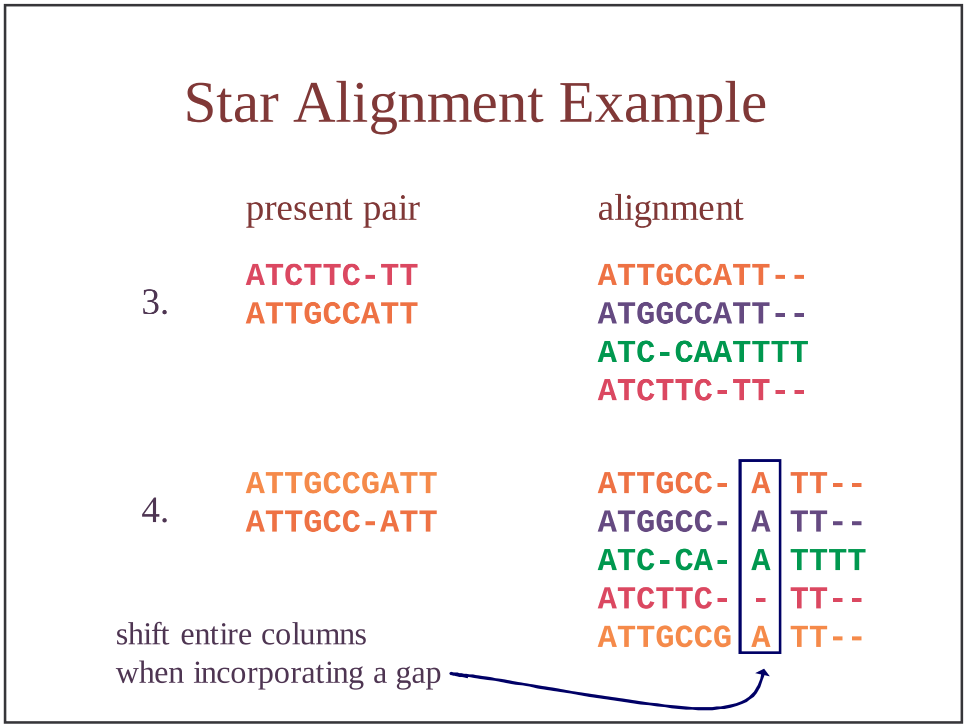
Star Alignment Example

- merging pairwise alignments

```
      present pair       alignment

1.    ATGGCCATT          ATTGCCATT
      ATTGCCATT          ATGGCCATT

2.    ATC-CAATTTT        ATTGCCATT--
      ATTGCCATT--        ATGGCCATT--
                         ATC-CAATTTT
```



Star Alignment Example

```
      present pair       alignment

3.    ATCTTC-TT          ATTGCCATT--
      ATTGCCATT          ATGGCCATT--
                         ATC-CAATTTT
                         ATCTTC-TT--

4.    ATTGCCGATT         ATTGCC- A TT--
      ATTGCC-ATT         ATGGCC- A TT--
                         ATC-CA- A TTTT
                         ATCTTC- - TT--
                         ATTGCCG A TT--
```

shift entire columns
when incorporating a gap

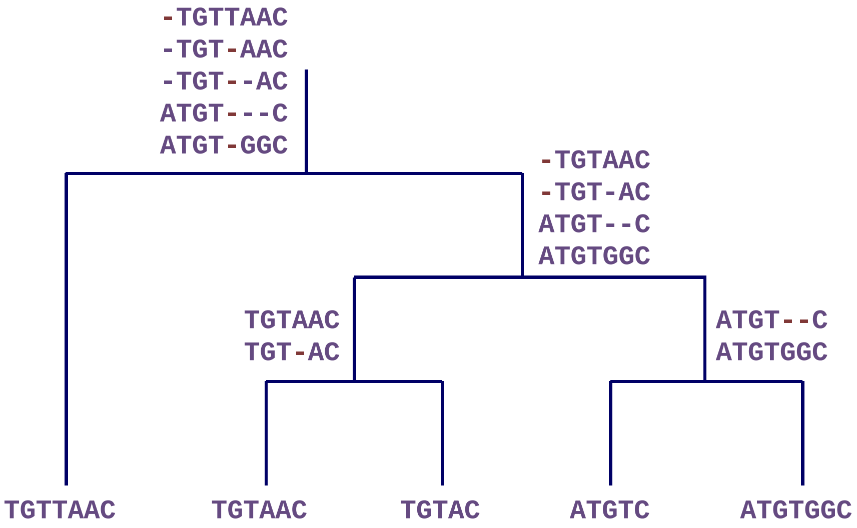
Tree Alignments

- 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: the CLUSTALW algorithm [Thompson et al. 1994]

Doing the Progressive Alignment in CLUSTALW

- depending on the internal node in the tree, we may have to align a
  - 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 SP scoring

Tree Alignment Example



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

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

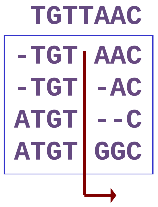
```
TGTAAC
TGT-AC
```

```
ATGT--C
ATGTGGC
```

```
TGTTAAC    TGTAAC    TGTAC    ATGTC    ATGTGGC
```

Aligning Profiles

- aligning sequences/profiles to profiles is essentially pairwise alignment
  - shift entire columns when incorporating gaps



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

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