Practical Bioinformatics

Mark Voorhies

6/22/2010

Review

- Sets
- Dictionaries
- list.sort

Review

- Sets
- Dictionaries
- list.sort
- string.strip, string.join
- Things that work for both sets and lists:
  - Concatenation with + or +=
  - Indexing and slicing
  - For loops

Review

- Sets
- Dictionaries
- list.sort
- string.strip, string.join
- Things that work for both sets and lists:
    - Concatenation with + or +=
    - Indexing and slicing
    - For loops
- The re module (re.search and re.finditer)
- (string.find and string.replace)

Why compare sequences?

Why compare sequences?

- To find genes with a common ancestor
- To infer conserved molecular mechanism and biological function
- To find short functional motifs
- To find repetitive elements within a sequence
- To predict cross-hybridizing sequences (e.g. in microarray design)
- To predict nucleotide secondary structure

Nomenclature

Homologs	heritable elements with a common evolutionary origin.

Nomenclature

Homologs	heritable elements with a common evolutionary origin.
Orthologs	homologs arising from speciation.
Paralogs	homologs arising from duplication and divergence within a single genome.

Nomenclature

Homologs	heritable elements with a common evolutionary origin.
Orthologs	homologs arising from speciation.
Paralogs	homologs arising from duplication and divergence within a single genome.
Xenologs	homologs arising from horizontal transfer.
Onologs	homologs arising from whole genome duplication.

Exercise: Scoring an ungapped alignment

```
s  ={"A" :{"A" :  1.0 ,"T" : -1.0 ,"G" : -1.0 ,"C" : -1.0} ,
     "T" :{"A" : -1.0 ,"T" :  1.0 ,"G" : -1.0 ,"C" : -1.0} ,
     "G" :{"A" : -1.0 ,"T" : -1.0 ,"G" :  1.0 ,"C" : -1.0} ,
     "C" :{"A" : -1.0 ,"T" : -1.0 ,"G" : -1.0 ,"C" :  1.0}}
```

Exercise: Scoring an ungapped alignment

$$
\begin{aligned}
s = \{ &"A":\{"A": \ 1.0, "T": -1.0, "G": -1.0, "C": -1.0\},\\
&"T":\{"A": -1.0, "T": \ 1.0, "G": -1.0, "C": -1.0\},\\
&"G":\{"A": -1.0, "T": -1.0, "G": \ 1.0, "C": -1.0\},\\
&"C":\{"A": -1.0, "T": -1.0, "G": -1.0, "C": \ 1.0\}\}
\end{aligned}
$$

$$S(x,y) = \sum_{i}^{N} s(x_i, y_i)$$

Exercise: Scoring an ungapped alignment

$$
\begin{aligned}
s = \{ & "A" : \{ "A" : \ 1.0, "T" : -1.0, "G" : -1.0, "C" : -1.0\}, \\
& "T" : \{ "A" : -1.0, "T" : \ 1.0, "G" : -1.0, "C" : -1.0\}, \\
& "G" : \{ "A" : -1.0, "T" : -1.0, "G" : \ 1.0, "C" : -1.0\}, \\
& "C" : \{ "A" : -1.0, "T" : -1.0, "G" : -1.0, "C" : \ 1.0\}\}
\end{aligned}
$$

$$S(x, y) = \sum_{i}^{N} s(x_i, y_i)$$

1. Given two equal length sequences and a scoring matrix, return the alignment score for a full length, ungapped alignment.

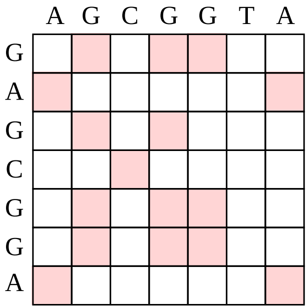
Exercise: Scoring an ungapped alignment

```
s  ={"A" : {"A" :  1.0 ,"T" : -1.0 ,"G" : -1.0 ,"C" : -1.0} ,
     "T" : {"A" : -1.0 ,"T" :  1.0 ,"G" : -1.0 ,"C" : -1.0} ,
     "G" : {"A" : -1.0 ,"T" : -1.0 ,"G" :  1.0 ,"C" : -1.0} ,
     "C" : {"A" : -1.0 ,"T" : -1.0 ,"G" : -1.0 ,"C" :  1.0}}
```

$$S(x, y) = \sum_{i}^{N} s(x_i, y_i)$$

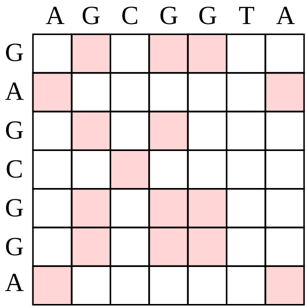
1. Given two equal length sequences and a scoring matrix, return the alignment score for a full length, ungapped alignment.
2. Given two sequences and a scoring matrix, find the offset that yields the best scoring ungapped alignment.

Dotplots



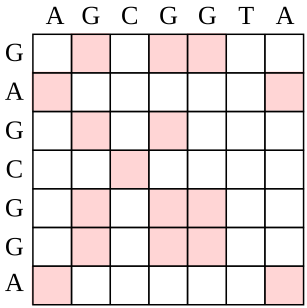
	A	G	C	G	G	T	A
G		🟥		🟥	🟥		
A	🟥						🟥
G		🟥		🟥			
C			🟥				
G		🟥		🟥	🟥		
G		🟥		🟥	🟥		
A	🟥						🟥

Dotplots



1. Given two sequences, write a dotplot in CDT format for JavaTreeView

Dotplots



1. Given two sequences, write a dotplot in CDT format for JavaTreeView
2. Add a windowing function to smooth the dotplot

Exercise: Scoring a gapped alignment

1. Given two equal length gapped sequences (where "-" represents a gap) and a scoring matrix, calculate an alignment score with a -1 penalty for each base aligned to a gap.

Exercise: Scoring a gapped alignment

1. Given two equal length gapped sequences (where "-" represents a gap) and a scoring matrix, calculate an alignment score with a -1 penalty for each base aligned to a gap.
2. Write a new scoring function with separate penalties for opening a zero length gap (*e.g.*, G = -11) and extending an open gap by one base (*e.g.*, E = -1).

$$S_{gapped}(x,y) = S(x,y) + \sum_{i}^{gaps}(G + E * len(i))$$

Types of alignments

Global Alignment
: Each letter of each sequence is aligned to a letter or a gap (*e.g.*, Needleman-Wunsch)

Local Alignment
: An optimal pair of subsequences is taken from the two sequences and globally aligned (*e.g.*, Smith-Waterman)

Homework

1. Read chapter 3 of the BLAST book (Sequence Alignment).
2. Try initializing and filling in a dynamic programming matrix by hand (*e..g*, try reproducing one of the examples from the BLAST book on paper).