

Sequence Alignment Sample Design

CSCI 134

This is one way to structure the program, but certainly not the only way. It covers the main properties of the program, but is not a complete implementation --- there may be additional useful methods, instance variables, constants, etc. to include in the code.

Feel free to use ideas from this design in addition to your original designs while working on the code.

class Organism

Instance Vars:

- `String binomialName` the organism's binomial name
- `String commonName` its common name
- `String aminoAcidSequence` the particular sequence to be aligned

Constructor:

```
Organism(String bName, String cName, String sequence)
```

Remember the binomial name, common name, and sequence for later

Methods:

- `public String getBinomialName()`
return the binomial name
 - `public String getCommonName()`
return the common name
 - `public String getSequence()`
return the sequence
-

class OrganismCollection

Instance Vars:

- `Organism[] organisms` Array of organisms

- JButton simpleButton, blossomButton To select similarity metric
- Text alignmentDisplay1, alignmentDisplay2 To display on canvas
- OrganismCollection organisms A collection of organisms
- SimilaritySchemeInterface similarityScheme Similarity metric

begin():

Set up the GUI:

canvas for displaying alignments in the CENTER

JPanel in the WEST for buttons that control canvas display

JPanel in the NORTH to manually enter sequences; button to align them

JPanel in the SOUTH for button to select a file; menus and buttons

JPanel in the EAST for buttons to select similarity metric

Initialize similarityScheme to be a SimpleSimilarityScheme

Methods:

- private void loadOrgFile(String orgFile)
construct a new organism collection
try
 create a BufferedReader (or Scanner)
 while (the file still has data)
 read it
 add the organism to each menu
 add it to the collection
catch
- private void alignAndDisplay(String seq1, String seq2)
align the given sequences
display the alignment on the canvas
- private void displayBestMatches(OrganismScoreMatrix m)
display all the ranked matches on the canvas
- public void actionPerformed(ActionEvent e)
determine which button was clicked
react appropriately

class SimpleSimilarityScheme implements SimilaritySchemeInterface

Instance Vars:

- `private final int MATCH_SCORE = 2`
- `private final int MISMATCH_SCORE = -1`
- `private final int GAP = -1` the particular sequence to be aligned

Constructor:

```
SimpleSimilarityScheme()  
Nothing to do here
```

Methods:

- `public int getSimilarity(char c1, char c2)`
return `MATCH_SCORE` if `c1` and `c2` match
return `MISMATCH_SCORE` otherwise
- `public int getGapScore()`
return the gap score

class OrganismScoreMatrix

Instance Vars:

- `OrganismCollection organisms` organisms for which to compute scores
- `int[][] scoreMatrix` store scores for all pairs of organisms
- `SimilaritySchemeInterface similarityScheme` the similarity metric to be used during alignment

Constructor:

```
OrganismScoreMatrix(OrganismCollection collection,  
                    SimilaritySchemeInterface scheme)
```

Construct the score matrix
Remember the collection and scheme
Compute the score matrix

Methods:

- `private void computeScoreMatrix()`
for each pair of organisms
 if both organisms in the pair are the same, set their match score to a constant
 else construct a new `Aligner` for the two organisms' sequences,

compute the alignment,
get the score for the alignment and store it in the proper place in the matrix

- `public int[][] getMatrix()`
return the score matrix
-

class Aligner

Instance Vars:

- `private static final int DIAG = 1`
- `private static final int LEFT = 2`
- `private static final int UP = 3`
- `int[][] h` the alignment score matrix
- `int[][] direction` the array of directions
- `String seq1, seq2` the strings to align
- `SimilaritySchemeInterface similarityScheme` similarity metric to use
- `String alignment = ""` the final alignment; none at the start
- `int score = -999` the alignment score; -999 if no alignment yet

Constructor:

```
Aligner(String sequence1, String sequence2  
        SimilaritySchemeInterface scheme)
```

Construct the matrices; initialize first row and column of direction matrix
Remember the sequences and similarity scheme
Compute the alignment

Methods:

- `private void computeAlignment()`
for x = 1 to the number of columns
 for y = 1 to the number of rows
 compute `h[x][y]` and `direction[x][y]` by determining the maximum of
 the four values as specified in the algorithm
find the highest value in `h`; that is the score of the best alignment
remember the row and column of the highest value in order to trace back
construct a 1-d array of `int`, to hold the directions of the trace
trace back through the direction array, starting at the location of the highest value
 while not yet at `direction[0][0]`
 record the directions traversed
declare local variables for the two sequences as they're aligned:
 `s1 = ""`
 `s2 = ""`

for loop to walk through 1-d trace array backward
 if trace[i] is diagonal, add one character from each original sequence to
 the corresponding aligned sequence
 else if trace[i] is left, add a char from seq1 to s1, add a "-" to s2
 else if trace[i] is up, add a char from seq2 to s2, add a "-" to s1
if the high score in h was not in the lower right corner, add the missing sequence
suffixes to s1 and s2
alignment = s1 + "\n" + s2

- `public int getScore()`
 return the score
- `public String getAlignment()`
 return the alignment