example_pairwiseGlobalAlignment

July 10, 2019

[1]:
```
import sys
sys.path.append("../src/")
from pairwiseGlobalAlignment import pairwiseGlobalAlignment
import pandas as pd
```

[2]:
```
# Import sequence alignment data
trainingDataFeatures = pd.read_csv("../data/Table S7 CS training data with all␣
 ↪features.csv")
```

[140]:
```
# Check the shape of the imported data
trainingDataFeatures.shape
```

[140]:
```
(3615, 1387)
```

"E13083.pdb" and "E13082.pdb" have the same surrounding 21mer sequences, so I'm going to exclude "E13082" to match the output from Table S7. For more information, see the "duplicate_21mers_fromS7.csv" table in this directory.

[4]:
```
# Remove E13082 (see note above)
trainingDataFeatures = trainingDataFeatures[trainingDataFeatures["Protein ID"] !
 ↪= "E13082"]

# Get the unique 21mer training data features and their associated labels
kmerSequences = list(trainingDataFeatures["Surrounding 21mer"].unique())
siteByStructureLabels = list(trainingDataFeatures["Site by Structure"].unique())
```

[5]:
```
# Just a quick informal check. These lengths should definitely be equal
len(kmerSequences)
len(kmerSequences) == len(siteByStructureLabels)
```

[5]:
```
True
```

[40]:
```
# Get pairwise global alginments of our kmer sequences
alignments = pairwiseGlobalAlignment(kmerSequences, "../matricies/")
```

[41]:
```
# Set up a dataframe to view the pairwise global alignments more easily
alignmentsdf = pd.DataFrame(alignments[0])
alignmentsdf.columns = siteByStructureLabels
alignmentsdf["Site by Structure"] = siteByStructureLabels
alignmentsdf.set_index("Site by Structure", inplace = True)
```

[144]:
```
# Take a peak at the alignments
alignmentsdf.iloc[0:3,0:3]
```

[144]:

Site by Structure	2zjr_C.pdb_K116	2zjr_C.pdb_K129	2zjr_C.pdb_K87
2zjr_C.pdb_K116	26.0	-1.6	0.4
2zjr_C.pdb_K129	-1.6	29.0	0.2
2zjr_C.pdb_K87	0.4	0.2	31.8

[113]:
```
# Grab the true alignments from the S7 table, order them in the same order as␣
 ↪the alignmentsdf above
trainingDataFeatures_ordered = pd.DataFrame()
trainingDataFeatures_unique = trainingDataFeatures.drop_duplicates("Site by␣
 ↪Structure")

for site in siteByStructureLabels:
    trainingDataFeatures_ordered[site] = trainingDataFeatures_unique[site]

trainingDataFeatures_ordered["Site by Structure"] = trainingDataFeatures["Site␣
 ↪by Structure"]
trainingDataFeatures_ordered.set_index("Site by Structure", inplace = True)
```

[145]:
```
# Take a peak at the alignments from the S7 table
trainingDataFeatures_ordered.iloc[0:3,0:3]
```

[145]:

Site by Structure	2zjr_C.pdb_K116	2zjr_C.pdb_K129	2zjr_C.pdb_K87
2zjr_C.pdb_K116	26.0	-0.8	1.4
2zjr_C.pdb_K129	-0.8	29.0	1.0
2zjr_C.pdb_K87	1.4	1.0	31.8

Right now, the alignments don't agree (except for the diagonals). I did a diff to make sure that the HIJACK matricies are the same, and they are.

The difference is that the MATLAB code uses a glocal alignment instead of a fully global alignment. I think that is actually assigning gaps to the sequence alignment, which I don't think we want. For more information, see "../scratch/testGlocalAlignment.m"