Peplib Tutorial

Andrew White

April 3, 2014

1 Installation

To install `peplib`, simply run this command from the R command line:

```
install.packages("peplib")
```

and run `library(peplib)` to load *peplib* at the beginning of each example.

You may also install from github, which has the latest version and is more frequently updated.

```
wget https://github.com/whitead/peplib/archive/master.zip
unzip master.zip && rm master.zip
R CMD build peplib-master
sudo R CMD INSTALL peplib_*.tar.gz
```

2 Background

The manuscript describing the methods used in `peplib` is Reference 1.

3 Loading Data

The `read.sequences` will read in a text file containing sequences as ASCII strings in the first column. The text file may contain additional columns with data related to the sequences. There should be one sequence per line. The following non-standard amino acids codes may be used: ”`-`” to indicate a gap and ”`X`” to indicate an unknown amino acid. B and Z may also be used to indicate ambiguous D or N and E or Q, respectively. If the sequences are of varying length, it is recommended that the sequences are aligned. For example, using ‘clustal’ or some other online alignment too. Here is an example call to read in a file of sequences from file `sequences.txt`:

```
seq <- read.sequences("sequences.txt")

sequences.txt:

GFF-
```

```
RGD-
GRFD
-GFR
FGRD
.
.
.
```

Additionally, example datasets may be accessed using a call to `data`. There is a set of sequences from a bead library where each peptide sequence is able to be digested by the enzyme SHP-2 [2]. These may be loaded into the workspace by calling `data(SHP2Sequences)` and accessed from the variable `SHP2Sequences`. A dataset containing aligned, variable width sequences which are antimicrobial may be accessed with a call to `data(AMPSequences)`. These two sequence sets will be used as examples.

4 Motif Discovery

Motif discovery consists of two phases. The first is to cluster the sequences into different groups, typically by using a distance measure. The second phase is to discover the motifs in each group. There are as yet no simple algorithms to complete both phases simultaneously.

4.1 Clustering Sequences

In *peplib*, there are two standard ways of clustering sequences; however, many more clustering algorithms may be easily accessed in **R**. Both algorithms are accessed with a call to `aclust`. K-means, a standard clustering algorithm, may be executed by calling `aclust(SHP2Sequences, clusterNumber=3, type="kmeans")`. The second clustering algorithm is a single-linkage agglomerative clustering technique. This algorithm is often used in sequence analysis in genomics. It may be executed by calling `aclust(SHP2Sequences, clusterNumber=3, type="agglomeraitve")`. The default is `"kmeans"`.

The output of `aclust` is a list of arrays, where each array represents a cluster. The arrays contain the indices of the sequences in a given cluster. The results may be visualized:

```
data(SHP2Sequences)
theClusters <- aclust(SHP2Sequences, clusterNumber=3)
plot(SHP2Sequences, clusters=theClusters)
```

Clustering the sequences is optional, but important if there are multiple motifs present. The `plot` of the sequences helps determine this as well. Once the clusters are determined, it is possible to find the motifs present in each cluster. This also allows you to determine how different the clusters are. The clusters may be separated into new sequences sets using the following syntax:

```
data(SHP2Sequences)
```

```
clusters <- aclust(SHP2Sequences, clusterNumber=3)
cluster.1 <- SHP2Sequences[ clusters[[1]] ]
```

Finally, the motifs can be found using a motif model:

```
cluster.1.mm <- motifModel(cluster.1)
print(cluster.1.mm)
```

4.2 Motif Modeling

There are two parameters to be set when using a motif model. The first is the type of motif model. There are three types: (`fixed`) a motif model which assumes each sequence has a single motif occurrence at the same position in each sequence, (`variable`) assumes that each sequence has a single motif at different positions, and (`optional`) a motif model which assumes each sequence has one or zero motif occurrences at different positions. The first type has the lowest number of parameters but may not fit sequence sets well, especially if the clustering resulted in a diffuse cluster or if the sequences weren't aligned before being loaded. The model types may be compared using their respective Bayesian Information Criterion (BIC) by calling `BIC` on the motif model. The more complicated models should only be accepted if their BIC is lower than simpler models.

The second parameter to adjust for the motif model is the motif width. This may be adjusted by inspecting the type of motifs that occur. If a width of 6 has some variable positions, than try decreasing to 5, for example. The print method of the motif model displays each motif position as separated by '[]' brackets and if there are many amino acid letters in the bracket, the motif position is quite variable. For example, [R][GF][GFRD] indicates that only R has a greater than 15% probability in position 1. In position 2, only G and F do, with G having a higher probability than F. In position 3, G, F, R and D all have over 15% probability of occurring.

The motif models may be plotted as well; see the documentation for an example. Below is an example for testing between two motif model types.

```
data(SHP2Sequences)
clusters <- aclust(SHP2Sequences, clusterNumber=3)
cluster.1 <- SHP2Sequences[ clusters[[1]] ]

cluster.1.mm.fixed <- motifModel(SHP2Sequences, type="fixed")
cluster.1.mm.optional <- motifModel(SHP2Sequences, type="optional")

print(BIC(cluster.1.mm.fixed))
print(BIC(cluster.2.mm.fixed))

print(cluster.1.mm.fixed)
print(cluster.1.mm.fixed)
```

All of the steps seen here may be done with one function call to `motifModelSet`, which automates these steps. It is strongly recommended to perform each step since `motifModelSet` makes many guesses. Here is an example

```
data(SHP2Sequences)

mms <- motifModelSet(SHP2Sequences, motifNumber=3, width=4,
                     type="fixed")

print(mms)
```

4.3 Weighted Motif Discovery

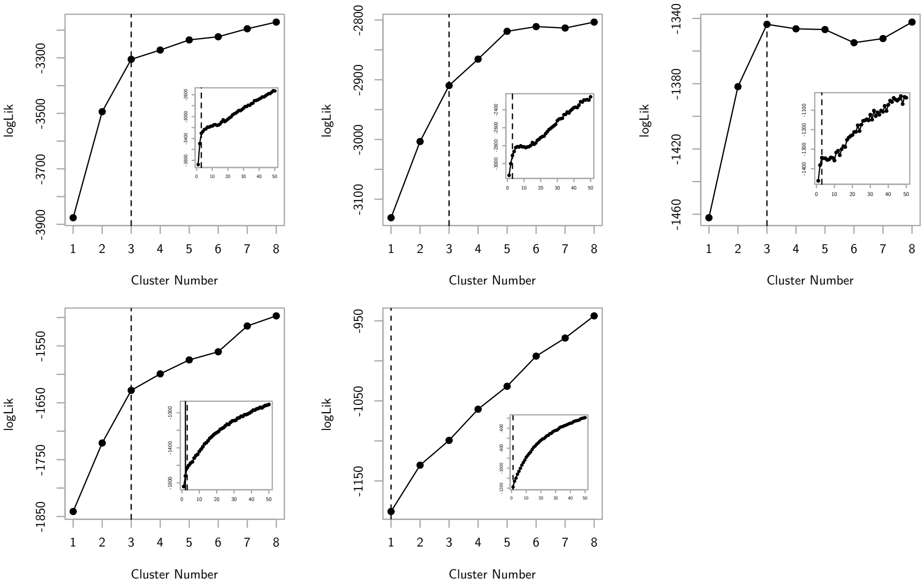
Sometimes the relative frequencies of sequences is known. For example, we may have observed one sequence many times more than another and wish to weigh motif discovery according to those relative frequencies. This can be done by using the `seq.weights` parameter in both `motifModel` and `motifModelSet` (see below). The `seq.weights` should be a vector of positive weights.

4.4 Choosing Motif Number

In addition to the methods presented above, it is possible to create an elbow plot to help choose the number of motifs present in a dataset. Using the same model type parameters, a call to `motifModelSet` can be made and it will try each motif number and plot the log-likelihood as a function of motif number. The motif number just before the first elbow in the plot is generally a good choice of motif number. The maximum number of motifs to be fit may be set as well.

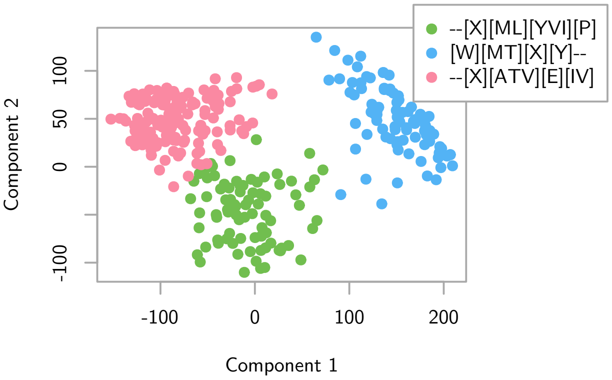
```
data(SHP2Sequences)

motifModelSet(SHP2Sequences, width=4, type="fixed", maxGuess=5)
```

Some examples of these plots along with the location of the elbow (the dashed line) are shown below. The plots without a dashed line have been determined to have only a single motif. The insets are a zoomed out view and better show the elbow.



5 Plotting

There are four main plot types in the *peplib* library. The first is the MotifModelSet plot type, an example of which is shown below.

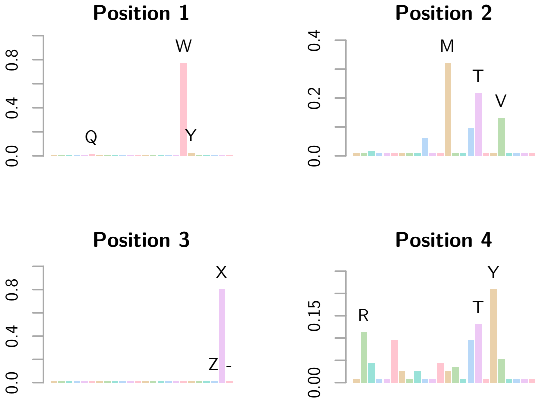


This plot may be produced with the following code:

```
data(SHP2Sequences)

ms <- motifModelSet(SHP2Sequences, width=4, type="fixed", motifNumber=3)
plot(ms)
```

This plot shows a projection of the sequences, where each point is a sequence from the dataset. The distance between points corresponds to the distance between sequences as calculated by the `dist` function. The colors corresponds to the grouping of sequences from a kmeans clustering. Finally, the legend shows a string representing the results of the motif models.

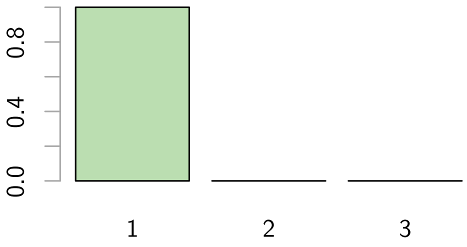
There are three types of plots for an individual motif. The first is the motif position plot, which shows the motif model at each position. An example is shown below and the code to generate one.



```
data(SHP2Sequences)

ms <- motifModelSet(SHP2Sequences, width=4, type="fixed", maxGuess=5)
motif1 <- ms@motifs[[1]]
plotPositions(motif1)
```

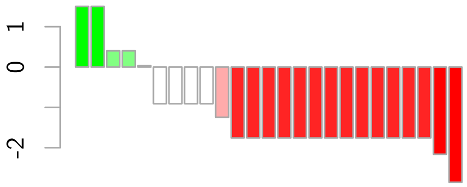
Each plot in the grid shows the likelihood of each amino acid in the motif. The three highest likelihood amino acids are annotated.

The second plot type for motifs is the starting position plot. This shows the likelihood of the motif starting at a given position. If the motif model is of type `optional` or `variable` then this plot is the average over all the sequences in the motif (the motifs of each sequence may start at different locations). An example of this plot and the code to generate it are given below.



```
data(SHP2Sequences)

ms <- motifModelSet(SHP2Sequences, width=4, type="fixed", maxGuess=5)
motif1 <- ms@motifs[[1]]
plotStartingPosition(motif1)
```

Finally, the last plot is the relative likelihood of the sequences. This barplot shows one bar per sequence in a motif model dataset and a positive number indicates the sequence fits the motif model better than a random model (multinomial) and a negative number indicates the sequence fits the random model better. A cluster with many negative numbers means the motif fits the cluster poorly. An example of this plot and the code to generate it are given below.



```
data(SHP2Sequences)

ms <- motifModelSet(SHP2Sequences, width=4, type="fixed", maxGuess=5)
motif1 <- ms@motifs[[1]]
plotFits(motif1)
```

6 Modeling

peplib also has some tools to perform traditional QSAR-type modeling on peptide libraries. This may add information about correlations between solubility and charge, for example, to activity. This type of modeling complements the motif discovery. A large descriptor matrix may be constructed using a call to `descriptors`. There are many options that control how many descriptors are created and the descriptors have unhelpful names due to their large number. For simple exploratory analysis, a call to `simpleDescriptors` provides a manageable number of intuitive descriptors. They are: the molecular weight, an estimated partition coefficient (solubility), an estimate of the surface area, an estimate of how helical the peptide is, and the number of aromatic bonds.

If there are responses (activities) these may be given as well in order to correlate activity. For example:

```
data(AMPSequences)
data(AMPSequences.response)

AMP.desc <- simpleDescriptors(AMPSequences,
                              response=AMPSequences.response)

print(corr(AMP.desc))

plot(AMP.desc)
```

The response file may be read it in using any traditional **R** `read` function, for example `read.csv`.

However, that is not always the case in peptide libraries. One often is more interested in understanding which what is common amongst the active sequences. This may be done by comparing a descriptor on the active sequences to the inactive sequences. Since inactive sequences are rarely collected in peptide libraries, we may approximate the inactive sequences as all sequences. **This assumption only holds if there is a low number of active sequences**. This is often the case but must be observed during the experiment. With this assumption, p-values may be calculated for each descriptor. These p-values do not assume normality and are a measure of the overlap between the active sequences and inactive sequences. They are calculated using a Mann-Whitney two-sample A p-value below 0.05 is considered significant and such a descriptor may be considered to be related to activity. **Remember that a descriptor may be important in connection to a motif**. Thus it is important to do both modeling and motif discovery. The p-values may be calculated like so:

```
data(SHP2Sequences)


SHP2.desc <- simpleDescriptors(SHP2Sequences, include.statistics=T)
print(SHP2.desc@pvalues)
```

When using the full method, `descriptors`, one should take care in evaluating p-values. Given that there are thousands of descriptors, it is possible by chance to have a large number of significant p-values. There should be little problem, however, when evaluating p-values from the `simpleDescriptors` method.

7 Feedback

Please send questions or provide feedback to Andrew White at `white.d.andrew@gmail.com`.

References

[1] White, A. D.; Keefe, A.-J.; Nowinski, A. K.; Shao, Q.; Caldwell, K.; Jiang, S. *J Chem. Inf. Model.* **2012**, *53(2)*, 493–499

[2] Sweeney, M. C.; Wavreille, A.-S. S.; Park, J.; Butchar, J. P.; Tridandapani, S.; Pei, D. *Biochem* **2005**, *44*, 14932–14947