BPQuant Example

Lisa Bramer & Bobbie-Jo Webb-Robertson

February 17, 2016

This file is intended to demonstrate using the `bpquant()` function in R.

The function is written to take input from one protein at a time and requires two inputs:
1. `protein_sig` is a matrix or data.frame with p rows and n columns, where p is the number of peptides mapped to the protein of interest and n is the number of tests conducted to generate signatures made up of values 0, 1, and -1.
2. `pi_not` is a numeric value between 0 and 1 indicating the background probability/frequency of a zero signature.

Example Data

The file "protein_signatures.csv" contains some example signature data for a few different proteins and will be used to demonstrate the use of the `bpquant` function.

```
source("bpquant.R")
protein_sig_data = read.csv("protein_signatures.csv")

head(protein_sig_data)
```

```
##   Protein Peptide Sig1 Sig2 Sig3
## 1       A     PA1    0    0    0
## 2       A     PA2    0    0    1
## 3       A     PA3    0    0    0
## 4       A     PA4    0    0    1
## 5       A     PA5    0    0    1
## 6       A     PA6    0   -1    0
```

Using the Function

After reading in the data, we can see that there are signatures for peptides associated with three different proteins. We will start by pulling the signatures for the first protein:

```
# number of proteins in the data.frame #
length(unique(protein_sig_data$Protein))
```

```
## [1] 3
```

```
cur_protein = subset(protein_sig_data, Protein == "A")

# make sure to remove any character columns before inputting into bpquant #
cur_protein_sigs = cur_protein[,-(1:2)]
```

Here's what our signature data looks like for all peptides mapping to the current protein of interest:

```
##   Sig1 Sig2 Sig3
## 1    0    0    0
## 2    0    0    1
## 3    0    0    0
## 4    0    0    1
## 5    0    0    1
## 6    0   -1    0
```

We then set our background frequency for a zero signature and call `bpquant`

```
pi_zero = 0.9

results = bpquant(protein_sig = cur_protein_sigs, pi_not = pi_zero)
```

Output

`bpquant` returns a list of five items:
1. num_proteoforms - the number of proteoforms as identified by bpquant
2. unique_sigs - matrix of unique signatures observed
3. proteoform_configs - matrix of 0/1 values indicating scenarios of proteoform absence/presence scenarios
4. post_prob - vector of posterior probabilities corresponding to each proteoform configuration in "proteoform_configs"
5. peptide_idx - vector of 0, 1, 2, . . . values indicating which proteoform each peptide belongs to

```
# bpquant identifies 1 proteoform #
results$num_proteoforms
```

```
## [1] 1
```

```
# unique signatures observed #
results$unique_sigs
```

```
##   Sig1 Sig2 Sig3
## 1    0    0    0
## 2    0    0    1
## 6    0   -1    0
```

```
# possible proteoform configurations #
results$proteoform_configs
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    1    0    0
## [3,]    0    1    0
## [4,]    0    0    1
## [5,]    1    1    0
## [6,]    1    0    1
## [7,]    0    1    1
## [8,]    1    1    1
```

```
# posterior probability of each proteoform configuration #
results$post_prob
```

```
## [1] 3.552385e-02 1.758527e-05 8.366089e-01 5.188153e-03 4.141442e-04
## [6] 2.568277e-06 1.221843e-01 6.048454e-05
```

```
# 0/1 vector indicating which peptides follow the predominant signature #
results$peptide_idx
```

```
## [1] 0 1 0 1 1 0
```

```
# here we would use the following peptides for protein quantitation #
cur_protein[which(results$peptide_idx == 1),]
```

```
##   Protein Peptide Sig1 Sig2 Sig3
## 2       A     PA2    0    0    1
## 4       A     PA4    0    0    1
## 5       A     PA5    0    0    1
```

Example – Two Proteoforms

Move onto the second protein in the data

```
cur_protein = subset(protein_sig_data, Protein == "B")

# make sure to remove any character columns before inputting into bpquant #
cur_protein_sigs = cur_protein[,-(1:2)]
```

Here's what our signature data looks like for all peptides mapping to the current protein of interest:

```
##    Sig1 Sig2 Sig3
## 7     0    0    0
## 8     0    0    1
## 9     0   -1   -1
## 10    0    0    1
## 11    0    0    1
## 12    0   -1   -1
## 13    0   -1   -1
```

We then set our background frequency for a zero signature and call `bpquant`

```
pi_zero = 0.9

results = bpquant(protein_sig = cur_protein_sigs, pi_not = pi_zero)
```

Output

```
# bpquant identifies 2 proteoform #
results$num_proteoforms
```

```
## [1] 2
```

```
# unique signatures observed #
results$unique_sigs
```

```
##   Sig1 Sig2 Sig3
## 7    0    0    0
## 8    0    0    1
## 9    0   -1   -1
```

```
# possible proteoform configurations #
results$proteoform_configs
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    0
## [2,]    1    0    0
## [3,]    0    1    0
## [4,]    0    0    1
## [5,]    1    1    0
## [6,]    1    0    1
## [7,]    0    1    1
## [8,]    1    1    1
```

```
# posterior probability of each proteoform configuration #
results$post_prob
```

```
## [1] 4.470542e-03 4.023489e-09 6.239154e-02 6.239154e-02 5.615239e-08
## [6] 5.615239e-08 8.707455e-01 7.836710e-07
```

```
# 0/1/2 vector indicating which peptides follow the which predominant signature #
results$peptide_idx
```

```
## [1] 0 1 2 1 1 2 2
```

```
# proteoform 1 #
cur_protein[which(results$peptide_idx == 1),]
```

```
##    Protein Peptide Sig1 Sig2 Sig3
## 8        B     PB2    0    0    1
## 10       B     PB4    0    0    1
## 11       B     PB5    0    0    1
```

```
# proteoform 2 #
cur_protein[which(results$peptide_idx == 2),]
```

```
##    Protein Peptide Sig1 Sig2 Sig3
## 9        B     PB3    0   -1   -1
## 12       B     PB6    0   -1   -1
## 13       B     PB7    0   -1   -1
```

```
# peptides which don't follow either proteoform signature #
cur_protein[which(results$peptide_idx == 0),]
```

```
##   Protein Peptide Sig1 Sig2 Sig3
## 7       B     PB1    0    0    0
```