Improving analysis code by writing functions

Andrew Kapinos

10/23/2021

Can you improve this analysis code?

library(bio3d) s1 <- read.pdb("4AKE") # kinase with drug s2 <- read.pdb("1AKE") # kinase no drug s3 <- read.pdb("1E4Y") # kinase with drug

s1.chainA <- trim.pdb(s1, chain="A", elety="CA") s2.chainA <- trim.pdb(s2, chain="A", elety="CA") s3.chainA <- trim.pdb(s3, chain="A", elety="CA")

s1.b <- s1.chainA$*atom*b s2.b <- s2.chainA$*atom*b s3.b <- s3.chainA$*atom*b

plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor") plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor") plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")

> Q6. How would you generalize the original code above to work with any set of input protein structures?

We need to break down the original code into its various steps.

Step 1: Load bio3d packages (not neccessary to include in function). Step 2: Access online PDB file and assign to an object. Step 3: Trim PDB file to only include chain A. Step 4: Describe b for atoms in chain A. Step 5: Plot data.

Starting at step 2, we can simplify the code given for accessing the PDB files and assigning them to an object. At each step, we'll run the original code and our new version to ensure that the same output is generated.

```
# Original code
library(bio3d)
s1 <- read.pdb("4AKE") # kinase with drug
```

```
##   Note: Accessing on-line PDB file
```

```
s1
```

```
##
##  Call:  read.pdb(file = "4AKE")
##
##    Total Models#: 1
##      Total Atoms#: 3459,  XYZs#: 10377  Chains#: 2  (values: A B)
##
##      Protein Atoms#: 3312  (residues/Calpha atoms#: 428)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
```

```
##
##      Non-protein/nucleic Atoms#: 147  (residues: 147)
##      Non-protein/nucleic resid values: [ HOH (147) ]
##
##    Protein sequence:
##       MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##       DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##       VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##       YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILGMRIILLGAPGA...<cut>...KILG
##
## + attr: atom, xyz, seqres, helix, sheet,
##         calpha, remark, call
```

```
# Simplified version
library(bio3d)
x <- "4AKE"
x <- read.pdb(x)
```

```
##   Note: Accessing on-line PDB file

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/08/
## v95p5lpj0c1dymxdpt1292pw0000gn/T//Rtmp3Jtmrh/4AKE.pdb exists. Skipping download
```

```
x
```

```
##
##  Call:  read.pdb(file = x)
##
##    Total Models#: 1
##      Total Atoms#: 3459,  XYZs#: 10377  Chains#: 2  (values: A B)
##
##      Protein Atoms#: 3312  (residues/Calpha atoms#: 428)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
##
##      Non-protein/nucleic Atoms#: 147  (residues: 147)
##      Non-protein/nucleic resid values: [ HOH (147) ]
##
##    Protein sequence:
##       MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##       DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##       VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##       YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILGMRIILLGAPGA...<cut>...KILG
##
## + attr: atom, xyz, seqres, helix, sheet,
##         calpha, remark, call
```

Moving onto step 3, let's simplify the code used to trim the PDB file to only chain A.

```
# Original code
s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s1.chainA
```

```
##
##  Call:  trim.pdb(pdb = s1, chain = "A", elety = "CA")
##
##    Total Models#: 1
##      Total Atoms#: 214,  XYZs#: 642  Chains#: 1  (values: A)
##
##      Protein Atoms#: 214  (residues/Calpha atoms#: 214)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
##
##      Non-protein/nucleic Atoms#: 0  (residues: 0)
##      Non-protein/nucleic resid values: [ none ]
##
##    Protein sequence:
##       MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##       DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##       VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##       YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
##
## + attr: atom, helix, sheet, seqres, xyz,
##         calpha, call
```

```
# Simplified version
chainA <- trim.pdb(x, chain="A", elety="CA")
chainA
```

```
##
##  Call:  trim.pdb(pdb = x, chain = "A", elety = "CA")
##
##    Total Models#: 1
##      Total Atoms#: 214,  XYZs#: 642  Chains#: 1  (values: A)
##
##      Protein Atoms#: 214  (residues/Calpha atoms#: 214)
##      Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)
##
##      Non-protein/nucleic Atoms#: 0  (residues: 0)
##      Non-protein/nucleic resid values: [ none ]
##
##    Protein sequence:
##       MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
##       DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
##       VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
##       YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
##
## + attr: atom, helix, sheet, seqres, xyz,
##         calpha, call
```

Moving onto step 4, let's simplify the code used to describe b for atoms in chain A.

```
# Original code
s1.b <- s1.chainA$atom$b
s1.b
```

```
##   [1]  29.02  18.44  16.20  19.67  20.26  20.55  17.05  22.13  26.71  33.05
```

```
##  [11]  30.66  32.73  25.61  33.19  41.03  24.09  16.18  19.14  29.19  14.79
##  [21]  19.63  28.54  27.49  32.56  17.13  15.50   6.98  24.07  24.00  23.94
##  [31]  30.70  24.70  32.84  34.60  33.01  44.60  50.74  57.32  47.04  67.13
##  [41]  81.04  75.20  59.68  55.63  45.12  39.04  44.31  38.21  43.70  44.19
##  [51]  47.00  48.67  41.54  50.22  45.07  49.77  52.04  44.82  39.75  35.79
##  [61]  38.92  37.93  27.18  26.86  27.53  31.16  27.08  23.03  28.12  24.78
##  [71]  24.22  18.69  40.67  38.08  55.26  46.29  26.25  37.14  27.50  16.86
##  [81]  27.76  19.27  22.22  26.70  25.52  21.22  15.90  15.84  22.44  19.61
##  [91]  21.23  21.79  17.64  22.19  22.73  16.80  23.25  35.95  24.42  20.96
## [101]  20.00  25.99  24.39  17.19  12.16  17.35  24.97  14.08  22.01  22.26
## [111]  22.78  27.47  30.49  32.02  20.90  27.03  23.84  44.37  42.47  33.48
## [121]  44.56  56.67  60.18  66.62  59.95  70.81  88.63 100.11  86.60  85.80
## [131]  77.48  68.13  52.66  45.34  52.43  60.90  62.64  72.19  66.75  58.73
## [141]  74.57  79.29  79.53  76.58  66.40  64.76  70.48  74.84  70.11  74.82
## [151]  78.61  78.24  66.70  66.10  67.01  72.28  80.64  68.54  43.23  51.24
## [161]  45.72  61.60  45.61  42.57  41.03  41.02  33.34  19.48  34.38  33.11
## [171]  25.48  29.68  40.71  32.91  24.41  19.20  15.43  19.93  20.66  12.72
## [181]  21.40  18.21  26.68  34.50  25.77  26.52  36.85  31.05  39.84  48.03
## [191]  23.04  29.57  23.00  23.80  26.59  25.49  23.25  19.89  32.37  30.97
## [201]  42.16  29.64  29.69  33.15  26.38  23.17  29.35  32.80  25.92  38.01
## [211]  45.95  44.26  44.35  70.26
```

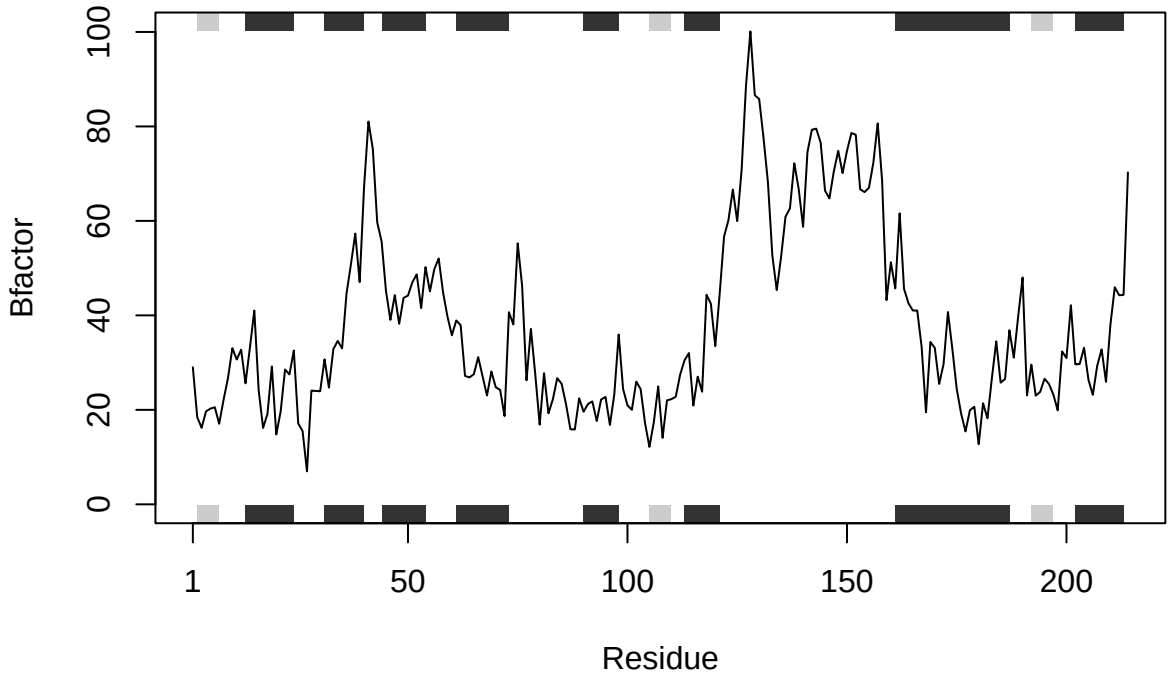
```
# Simplified version
b <- chainA$atom$b
b
```

```
##   [1]  29.02  18.44  16.20  19.67  20.26  20.55  17.05  22.13  26.71  33.05
##  [11]  30.66  32.73  25.61  33.19  41.03  24.09  16.18  19.14  29.19  14.79
##  [21]  19.63  28.54  27.49  32.56  17.13  15.50   6.98  24.07  24.00  23.94
##  [31]  30.70  24.70  32.84  34.60  33.01  44.60  50.74  57.32  47.04  67.13
##  [41]  81.04  75.20  59.68  55.63  45.12  39.04  44.31  38.21  43.70  44.19
##  [51]  47.00  48.67  41.54  50.22  45.07  49.77  52.04  44.82  39.75  35.79
##  [61]  38.92  37.93  27.18  26.86  27.53  31.16  27.08  23.03  28.12  24.78
##  [71]  24.22  18.69  40.67  38.08  55.26  46.29  26.25  37.14  27.50  16.86
##  [81]  27.76  19.27  22.22  26.70  25.52  21.22  15.90  15.84  22.44  19.61
##  [91]  21.23  21.79  17.64  22.19  22.73  16.80  23.25  35.95  24.42  20.96
## [101]  20.00  25.99  24.39  17.19  12.16  17.35  24.97  14.08  22.01  22.26
## [111]  22.78  27.47  30.49  32.02  20.90  27.03  23.84  44.37  42.47  33.48
## [121]  44.56  56.67  60.18  66.62  59.95  70.81  88.63 100.11  86.60  85.80
## [131]  77.48  68.13  52.66  45.34  52.43  60.90  62.64  72.19  66.75  58.73
## [141]  74.57  79.29  79.53  76.58  66.40  64.76  70.48  74.84  70.11  74.82
## [151]  78.61  78.24  66.70  66.10  67.01  72.28  80.64  68.54  43.23  51.24
## [161]  45.72  61.60  45.61  42.57  41.03  41.02  33.34  19.48  34.38  33.11
## [171]  25.48  29.68  40.71  32.91  24.41  19.20  15.43  19.93  20.66  12.72
## [181]  21.40  18.21  26.68  34.50  25.77  26.52  36.85  31.05  39.84  48.03
## [191]  23.04  29.57  23.00  23.80  26.59  25.49  23.25  19.89  32.37  30.97
## [201]  42.16  29.64  29.69  33.15  26.38  23.17  29.35  32.80  25.92  38.01
## [211]  45.95  44.26  44.35  70.26
```

Finally, in step 5, we simplify the code used to plot the data.

```
# Original code
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```

```
# Simplified version
plotb3(b, sse=chainA, typ="l", ylab="Bfactor")
```



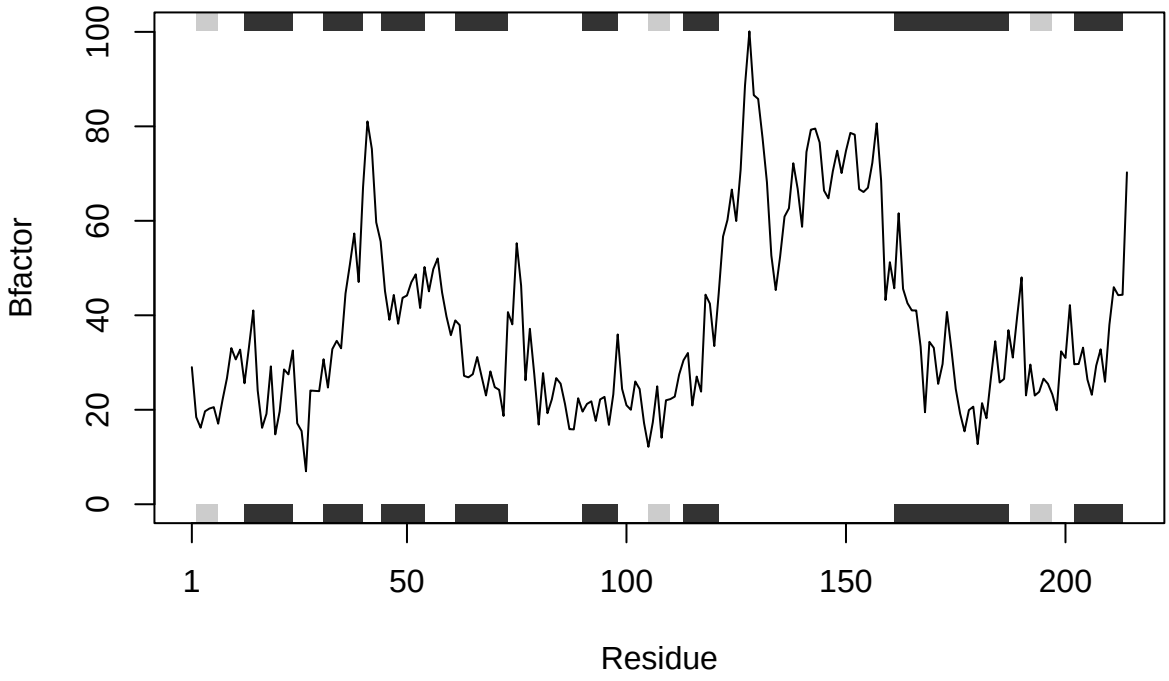
Let's put it all together!

```
x <- "4AKE"
library(bio3d)
x <- read.pdb(x)
```

```
##   Note: Accessing on-line PDB file

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/08/
## v95p5lpj0c1dymxdpt1292pw0000gn/T//Rtmp3Jtmrh/4AKE.pdb exists. Skipping download
```

```
chainA <- trim.pdb(x, chain="A", elety="CA")
b <- chainA$atom$b
plotb3(b, sse=chainA, typ="l", ylab="Bfactor")
```



Now, let's write our code into a function.

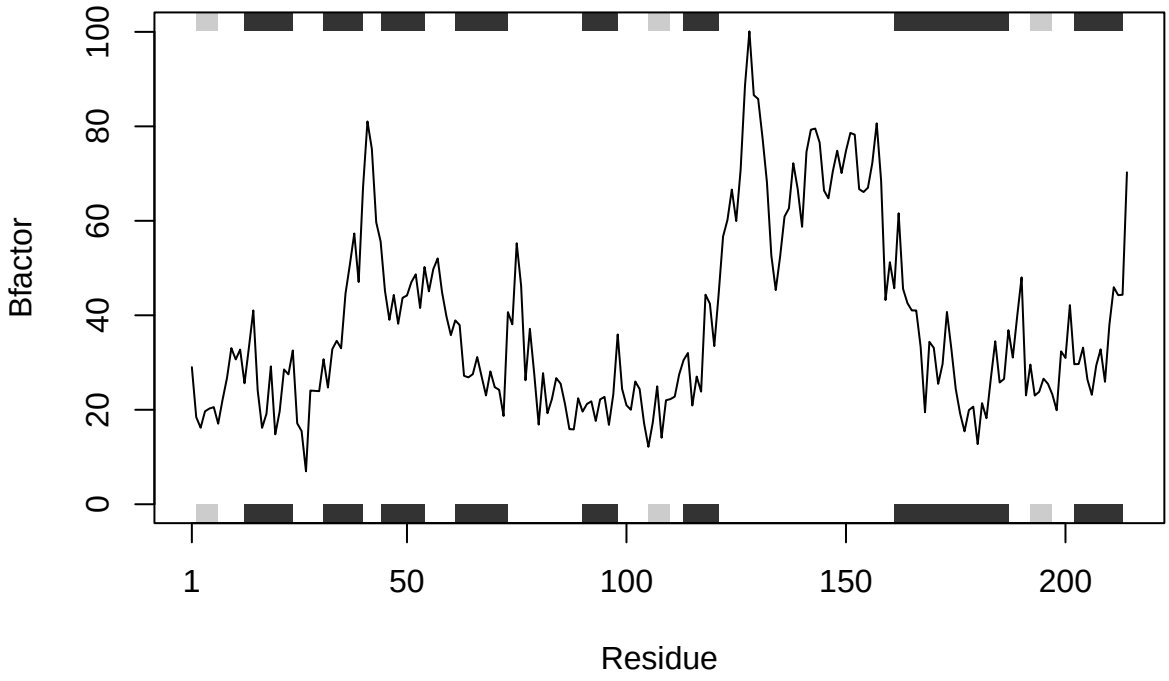
```
prot_drug_interaction <- function(x) {
  library(bio3d)
  x <- read.pdb(x)
  chainA <- trim.pdb(x, chain="A", elety="CA")
  b <- chainA$atom$b
  plotb3(b, sse=chainA, typ="l", ylab="Bfactor")
}
```

And let's verify that it works for of our sample input.

```
prot_drug_interaction("4AKE")
```

```
##   Note: Accessing on-line PDB file

## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/08/
## v95p5lpj0c1dymxdpt1292pw0000gn/T//Rtmp3Jtmrh/4AKE.pdb exists. Skipping download
```

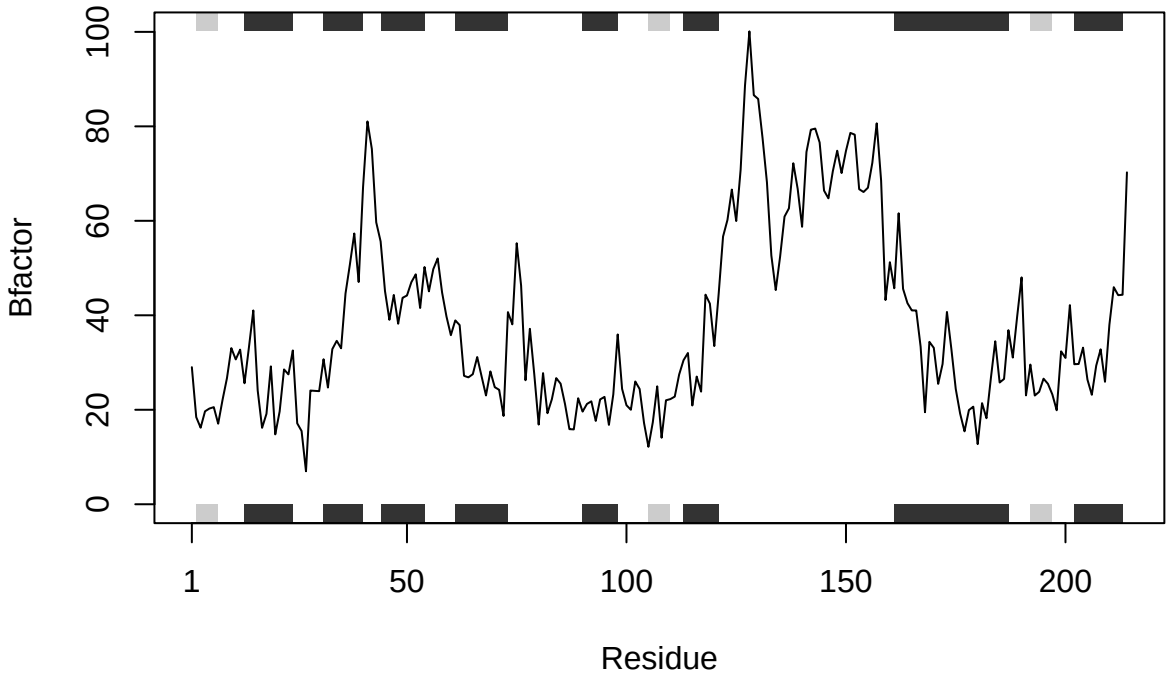


Let's create a data frame that contains the PDB codes provided as example inputs above. Then, we can use *apply()* to run the function on each of the original input PDB codes, by checking each row in the data frame for the input, x (ie. the PDB code).

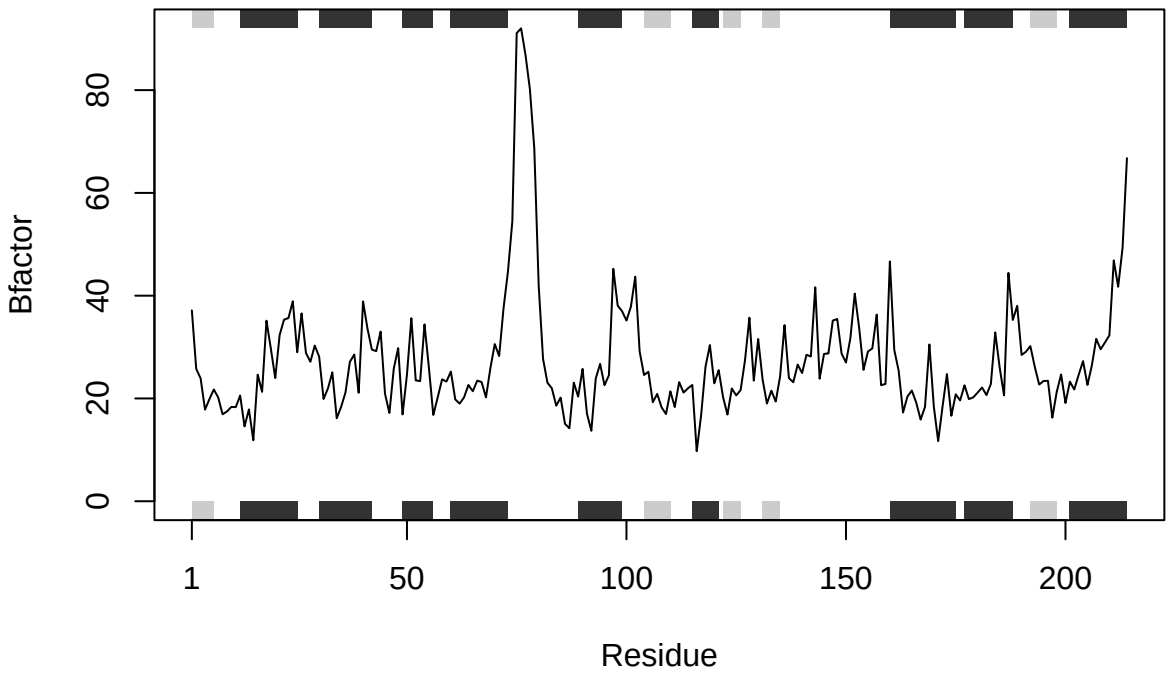
```
PDB_codes <- data.frame(c("4AKE","1AKE","1E4Y"))
apply(PDB_codes,1,prot_drug_interaction)
```

```
##   Note: Accessing on-line PDB file
```

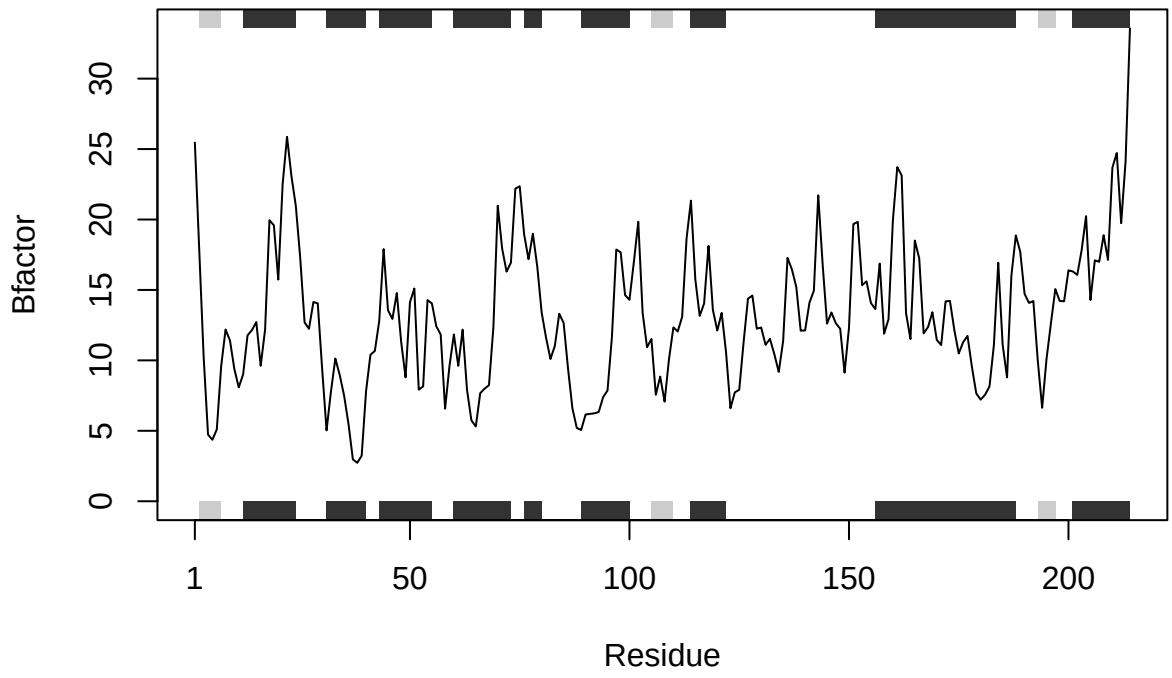
```
## Warning in get.pdb(file, path = tempdir(), verbose = FALSE): /var/folders/08/
## v95p5lpj0c1dymxdpt1292pw0000gn/T//Rtmp3Jtmrh/4AKE.pdb exists. Skipping download
```




```
##   Note: Accessing on-line PDB file
##    PDB has ALT records, taking A only, rm.alt=TRUE
```




```
##   Note: Accessing on-line PDB file
```




```
## NULL
```