Class06_Extension

Hailey Wheeler (A13312713)

```
# Can you improve this analysis code?
library(bio3d)

# Inserting PDB structure file for a specific protein "" - generating vector
s1 <- read.pdb("4AKE")  # kinase with drug
```

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

```
s2 <- read.pdb("1AKE")  # kinase no drug
```

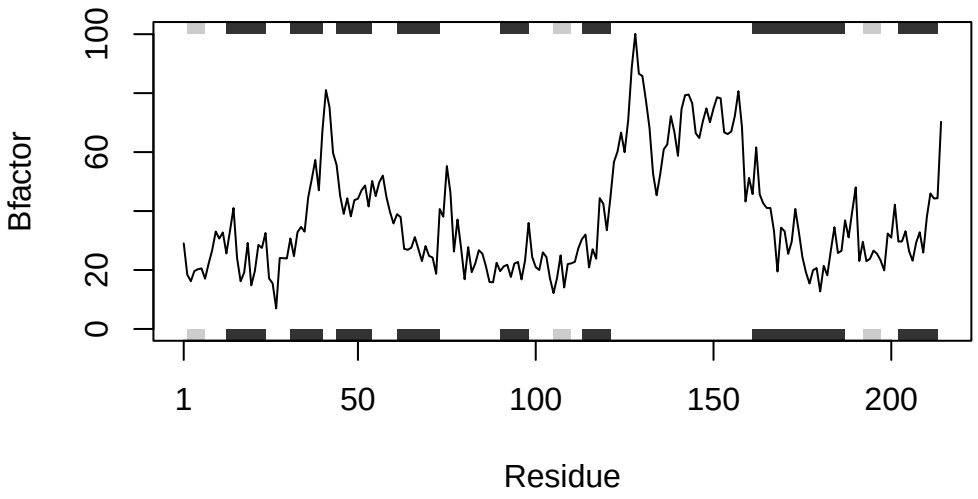
```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```

```
s3 <- read.pdb("1E4Y")  # kinase with drug
```

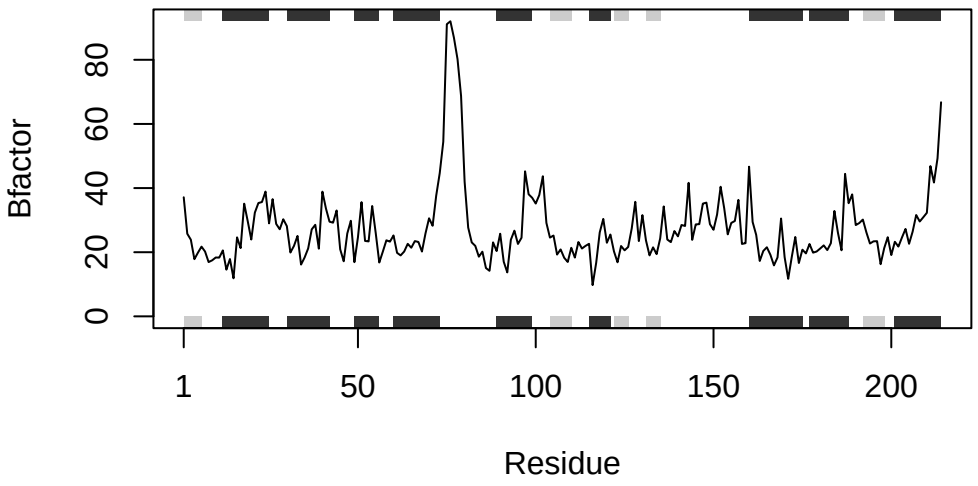
```
Note: Accessing on-line PDB file
```

```
# Create data.frame based on selected file: chain identifier, and atom type = elety
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")

# select character matrix containing all atomic coordinate data - create row per atom and
s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b
```

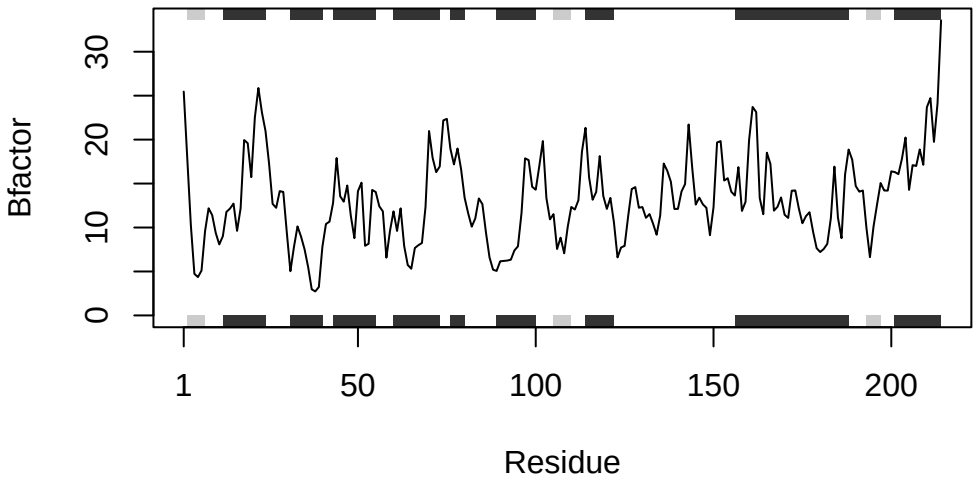
```
# Plot data with x = l and y = Bfactor
plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
```



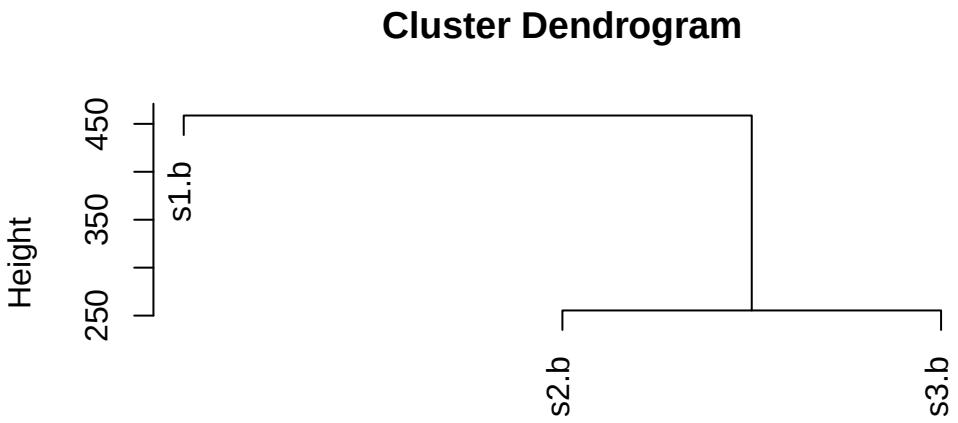

```
plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor")
```




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



```
# hierarchical cluster analysis on the dissimilarities between the chain associated with t
hc <- hclust( dist( rbind(s1.b, s2.b, s3.b) ) )
plot(hc)
```




```
lbio3d()
```

```
 [1] "aa.index"             "aa.table"             "aa123"
 [4] "aa2index"             "aa2mass"              "aa321"
 [7] "aanma"                "aanma.pdb"            "aanma.pdbs"
[10] "aln2html"             "amsm.xyz"             "angle.xyz"
[13] "as.fasta"             "as.pdb"               "as.pdb.default"
[16] "as.pdb.mol2"          "as.pdb.prmtop"        "as.select"
[19] "as.xyz"               "atom.index"           "atom.select"
[22] "atom.select.mol2"     "atom.select.pdb"      "atom.select.pdbs"
[25] "atom.select.prmtop"   "atom2ele"            "atom2ele.default"
[28] "atom2ele.pdb"         "atom2mass"            "atom2mass.default"
[31] "atom2mass.pdb"        "atom2xyz"             "basename.pdb"
[34] "bhattacharyya"        "bhattacharyya.array"  "bhattacharyya.enma"
[37] "bhattacharyya.matrix" "bhattacharyya.nma"    "bhattacharyya.pca"
[40] "binding.site"         "biounit"              "blast.pdb"
[43] "bounds"               "bounds.sse"           "build.hessian"
```

```
 [46] "bwr.colors"           "cat.pdb"              "chain.pdb"
 [49] "check.utility"        "clean.pdb"            "cmap"
 [52] "cmap.default"         "cmap.pdb"             "cmap.pdbs"
 [55] "cmap.xyz"             "cna"                  "cna.dccm"
 [58] "cna.ensmb"            "cnapath"              "com"
 [61] "com.pdb"              "com.xyz"              "combine.select"
 [64] "community.aln"        "community.tree"       "consensus"
 [67] "conserv"              "convert.pdb"          "core.cmap"
 [70] "core.find"            "core.find.default"    "core.find.pdb"
 [73] "core.find.pdbs"       "cov.enma"             "cov.nma"
 [76] "covsoverlap"          "covsoverlap.enma"     "covsoverlap.nma"
 [79] "dccm"                 "dccm.egnm"            "dccm.enma"
 [82] "dccm.gnm"             "dccm.nma"             "dccm.pca"
 [85] "dccm.xyz"             "deformation.nma"      "diag.ind"
 [88] "difference.vector"    "dist.xyz"             "dm"
 [91] "dm.pdb"               "dm.pdbs"              "dm.xyz"
 [94] "dssp"                 "dssp.pdb"             "dssp.pdbs"
 [97] "dssp.xyz"             "elements"             "entropy"
[100] "ff.aaenm"             "ff.aaenm2"            "ff.anm"
[103] "ff.calpha"            "ff.pfanm"             "ff.reach"
[106] "ff.sdenm"             "filter.cmap"          "filter.dccm"
[109] "filter.identity"      "filter.rmsd"          "fit.xyz"
[112] "fluct.nma"            "formula2mass"         "gap.inspect"
[115] "geostas"              "geostas.default"      "geostas.enma"
[118] "geostas.nma"          "geostas.pdb"          "geostas.pdbs"
[121] "geostas.xyz"          "get.blast"            "get.pdb"
[124] "get.seq"              "gnm"                  "gnm.pdb"
[127] "gnm.pdbs"             "hclustplot"           "hivp"
[130] "hmmer"                "identify.cna"         "inner.prod"
[133] "inspect.connectivity" "is.gap"               "is.mol2"
[136] "is.pdb"               "is.pdbs"              "is.select"
[139] "is.xyz"               "kinesin"              "layout.cna"
[142] "lbio3d"               "load.enmff"           "mask"
[145] "mask.dccm"            "mktrj"                "mktrj.enma"
[148] "mktrj.nma"            "mktrj.pca"            "mono.colors"
[151] "motif.find"           "mustang"              "network.amendment"
[154] "nma"                  "nma.pdb"              "nma.pdbs"
[157] "normalize.vector"     "orient.pdb"           "overlap"
[160] "pairwise"             "pca"                  "pca.array"
[163] "pca.pdbs"             "pca.tor"              "pca.xyz"
[166] "pdb.annotate"         "pdb.pfam"             "pdb2aln"
[169] "pdb2aln.ind"          "pdb2sse"              "pdbaln"
[172] "pdbfit"               "pdbfit.pdb"           "pdbfit.pdbs"
```

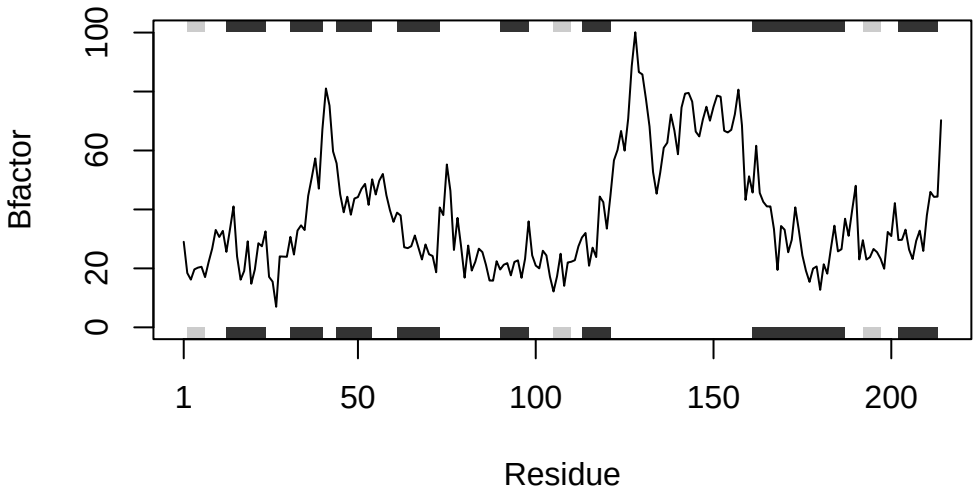
```
[175] "pdbs2pdb"           "pdbs2sse"           "pdbseq"
[178] "pdbsplit"           "pfam"               "plot.bio3d"
[181] "plot.blast"         "plot.cmap"          "plot.cna"
[184] "plot.cnapath"       "plot.core"          "plot.dccm"
[187] "plot.dmat"          "plot.ecna"          "plot.ecnapath"
[190] "plot.enma"          "plot.fasta"         "plot.fluct"
[193] "plot.geostas"       "plot.hmmer"         "plot.matrix.loadings"
[196] "plot.nma"           "plot.pca"           "plot.pca.loadings"
[199] "plot.pca.score"     "plot.pca.scree"     "plot.rmsip"
[202] "plotb3"             "print.cna"          "print.cnapath"
[205] "print.core"         "print.enma"         "print.fasta"
[208] "print.geostas"      "print.mol2"         "print.nma"
[211] "print.pca"          "print.pdb"          "print.prmtop"
[214] "print.rle2"         "print.select"       "print.sse"
[217] "print.xyz"          "project.pca"        "prune.cna"
[220] "pymol"              "pymol.dccm"         "pymol.modes"
[223] "pymol.nma"          "pymol.pca"          "pymol.pdbs"
[226] "read.all"           "read.cif"           "read.crd"
[229] "read.crd.amber"     "read.crd.charmm"    "read.dcd"
[232] "read.fasta"         "read.fasta.pdb"     "read.mol2"
[235] "read.ncdf"          "read.pdb"           "read.pdb2"
[238] "read.pdcBD"         "read.pqr"           "read.prmtop"
[241] "rgyr"               "rle2"               "rmsd"
[244] "rmsf"               "rmsip"              "rmsip.default"
[247] "rmsip.enma"         "rot.lsq"            "rtb"
[250] "sdENM"              "seq2aln"            "seqaln"
[253] "seqaln.pair"        "seqbind"            "seqidentity"
[256] "setup.ncore"        "sip"                "sip.default"
[259] "sip.enma"           "sip.nma"            "sse.bridges"
[262] "store.atom"         "stride"             "struct.aln"
[265] "summary.cna"        "summary.cnapath"    "summary.pdb"
[268] "torsion.pdb"        "torsion.xyz"        "transducin"
[271] "trim"               "trim.mol2"          "trim.pdb"
[274] "trim.pdbs"          "trim.xyz"           "unbound"
[277] "uniprot"            "var.pdbs"           "var.xyz"
[280] "vec2resno"          "vmd"                "vmd_colors"
[283] "vmd.cna"            "vmd.cnapath"        "vmd.ecna"
[286] "vmd.ecnapath"       "wrap.tor"           "write.crd"
[289] "write.fasta"        "write.mol2"         "write.ncdf"
[292] "write.pdb"          "write.pir"          "write.pqr"
[295] "xyz2atom"           "xyz2z.pca"          "z2xyz.pca"
```

```
help(bio3d)
```

```
## Install Bio3D package on Quarto
library(bio3d)

## Call your function(x)
protein <- function(x){

  # call protein file and establish as "x"
  x <- read.pdb(x)

  # create a new PDB object based on pdb selection of backbone atoms (selecting for Chain
  x.chainA <- trim.pdb(x, chain = "A", elety = "CA")

  # select character matrix containing all atomic coordinate data (create row per atom and
  # Plot residue temperature factor by secondary structure element of Chain A, using x = l
  plot.bio3d(x.chainA$atom$b, sse=x.chainA, type="l", ylab ="Bfactor")
}


# sse = secondary structure element
# Bfactor = residue temperature factors
# trim.pdb() = create a new PDB object based on our selection of backbone atoms


# Call created function 'protein' and apply that to a pdb object to analyze B-factor vs. R
protein("4AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/cs/673kvc2j4yn7dpx3wvpdlh8c0000gn/T//RtmpWvwj9S/4AKE.pdb exists.
Skipping download
```

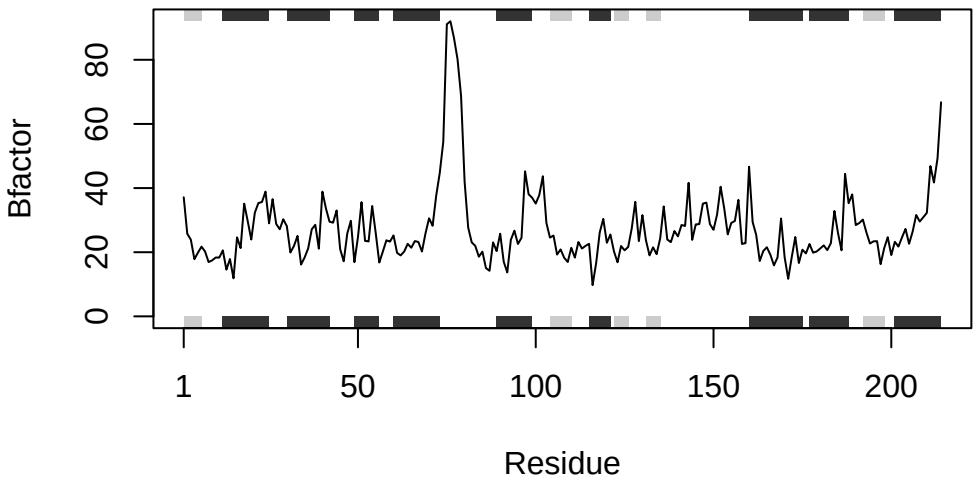



```
protein("1AKE")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/cs/673kvc2j4yn7dpx3wvpdlh8c0000gn/T//RtmpWvwj9S/1AKE.pdb exists.
Skipping download
```

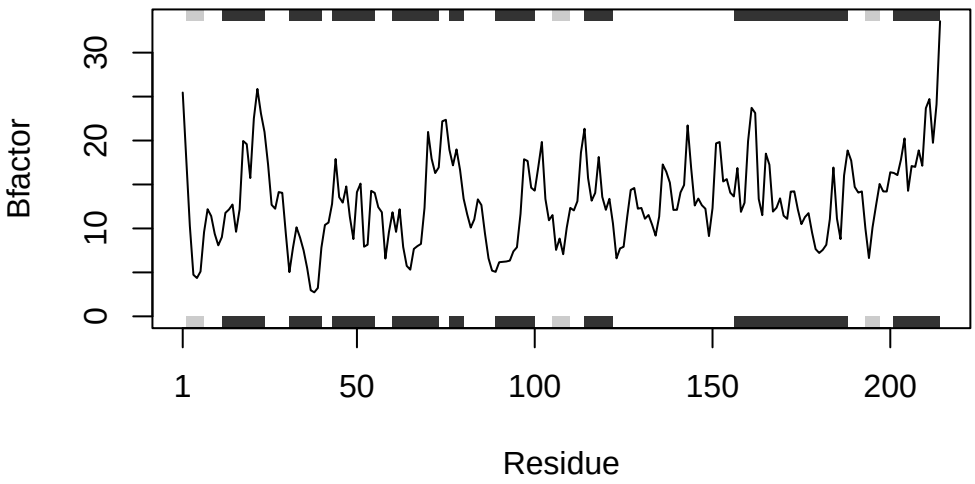
```
   PDB has ALT records, taking A only, rm.alt=TRUE
```




```
protein("1E4Y")
```

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/cs/673kvc2j4yn7dpx3wvpdlh8c0000gn/T//RtmpWvwj9S/1E4Y.pdb exists.
Skipping download
```


Bfactor
0
10
20
30
1
50
100
150
200
Residue