An add-on R package for Rosetta

Ingo Ruczinski

Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health

Why Such a Package?

The R package `rosetta` is not supposed to be a front-end for the folding algorithm Rosetta. It serves two other main purposes:

- Make the development of scoring functions to improve the Rosetta folding algorithm and the development of decoy filters a lot easier (rmsd-versus-score galore).

- Deal with the aftermath (such as the decoys and prediction summaries) in a much more convenient way.

Note that R is open source, but not in the public domain!

Some Benefits of R

- Borrow from a gazillion already existing functions and algorithms: `sweep()` and `svd()` versus `calc_rmsd.f` !
- Easily call functions written in C, Fortran, and many other low level programming languages.
- Save a lot of time on the HPU for example by taking advantage of vectorized operations.

```
> sqrt(mean(apply((y-x)^2,1,sum)))
```

- Lots of stuff (statistical modeling, clustering, plotting for publications) is done in R anyway.
- R supports object oriented programming to taylor your own classes/methods for protein structures.
- File exchanges are facilitated through OmegaHat.

Some Benefits of R Packages

- Almost 100% portable.
- Nice documentation and support through manuals, help files (`?pdb.read`), examples (`example(pdb.read)`), and demos (`demo(rosetta)`).
- People who write R packages are good citizens, make a difference in the community, improve their name recognition factor, get invited to fun places, and have their beer paid for.

Functions

```
read.pdb=function(fl,id,atms,dr,ext=".pdb"){
  if(missing(dr)){fl2=paste(fl,ext,sep="")}
  else{fl2=paste(dr,fl,ext,sep="")}
  zz=read.fwf(fl2,width=c(4,7,2,3,1,3,1,1,4,4,8,8,8),colClasses="character",comment.c
  zz=subset(zz,zz[,1]=="ATOM")
  zz=subset(zz,zz[,10]=="    ")
  zz=zz[,c(2,4,6,8,9,11,12,13)]
  names(zz)=c("nat","at","aa","id","naa","x","y","z")
  zz$id[is.na(zz$id)]=" "
  zz$nat=as.numeric(zz$nat)
  zz$naa=as.numeric(zz$naa)
  zz$x=as.numeric(zz$x)
  zz$y=as.numeric(zz$y)
  zz$z=as.numeric(zz$z)
  if(missing(atms)){
    atms=c(
      "C  ","CA ","CB ","CD ","CD1","CD2","CE ","CE1","CE2","CE3",
      "CG ","CG1","CG2","CH2","CZ ","CZ2","CZ3","N  ","ND1","ND2",
      "NE ","NE1","NE2","NH1","NH2","NZ ","O  ","OD1","OD2","OE1",
      "OE2","OG ","OG1","OH ","SD ","SG ")
  }

# cont.
```

Functions

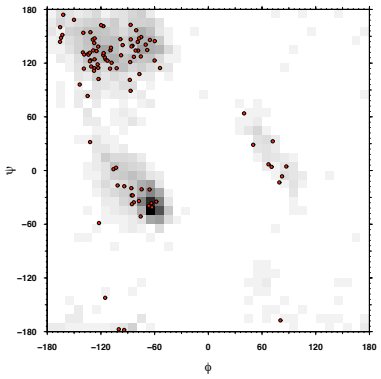
```
> read.pdb("1amu",id="A")
   nat  at  aa id naa      x      y      z
1    1 N   GLY  A  17 10.929 62.747 30.169
2    2 CA  GLY  A  17 12.121 63.555 30.349
3    3 C   GLY  A  17 11.903 64.708 31.310
4    4 O   GLY  A  17 10.812 65.281 31.365
5    5 N   THR  A  18 12.968 65.107 31.999
6    6 CA  THR  A  18 12.892 66.160 33.009
7    7 C   THR  A  18 13.464 67.514 32.561
8    8 O   THR  A  18 13.206 68.542 33.189
...

> read.pdb("1amu",id="A",atms="CA")

   nat  at  aa id naa      x      y      z
2    2 CA  GLY  A  17 12.121 63.555 30.349
6    6 CA  THR  A  18 12.892 66.160 33.009
13  13 CA  HIS  A  19 14.765 68.754 30.893
23  23 CA  GLU  A  20 17.327 69.446 33.609
32  32 CA  GLU  A  21 19.913 71.318 31.511
41  41 CA  GLU  A  22 17.278 73.664 30.123
50  50 CA  GLN  A  23 15.880 74.276 33.602
...
```

Graphics

- One of the great strengths of R are the graphics. Many functions to generate diagnostic plots or figures suitable for publication can easily be implemented.



- Visualizing protein structure is a lot harder, but it is supported through the CRAN library `rgl`. Rich already came up with a preliminary viewer. Ideally, however, we would piggy-back on a visualization tool such as RASMOL.

- X-Gobi flavored tools are on the way.