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().`

- 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.
  ```r
  > 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(4,7,2,3,1,3,1,4,4,8,8,8),colClasses="character",comment.
  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.

> read.pdb("lamu",id="A")

<table>
<thead>
<tr>
<th>nat</th>
<th>at</th>
<th>aa</th>
<th>id</th>
<th>naa</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>GLY</td>
<td>A</td>
<td>1</td>
<td>10.929</td>
<td>62.747</td>
<td>30.169</td>
</tr>
<tr>
<td>2</td>
<td>CA</td>
<td>GLY</td>
<td>A</td>
<td>17</td>
<td>12.121</td>
<td>63.555</td>
<td>30.349</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>GLY</td>
<td>A</td>
<td>17</td>
<td>11.903</td>
<td>64.708</td>
<td>31.310</td>
</tr>
<tr>
<td>4</td>
<td>O</td>
<td>GLY</td>
<td>A</td>
<td>17</td>
<td>10.812</td>
<td>65.281</td>
<td>31.365</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>THR</td>
<td>A</td>
<td>18</td>
<td>12.968</td>
<td>65.107</td>
<td>31.999</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
<td>THR</td>
<td>A</td>
<td>18</td>
<td>12.892</td>
<td>66.160</td>
<td>33.009</td>
</tr>
<tr>
<td>7</td>
<td>C</td>
<td>THR</td>
<td>A</td>
<td>18</td>
<td>13.464</td>
<td>67.514</td>
<td>32.561</td>
</tr>
<tr>
<td>8</td>
<td>O</td>
<td>THR</td>
<td>A</td>
<td>18</td>
<td>13.206</td>
<td>68.542</td>
<td>33.189</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>nat</th>
<th>at</th>
<th>aa</th>
<th>id</th>
<th>naa</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>GLY</td>
<td>A</td>
<td>17</td>
<td>12.121</td>
<td>63.555</td>
<td>30.349</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>THR</td>
<td>A</td>
<td>18</td>
<td>12.892</td>
<td>66.160</td>
<td>33.009</td>
</tr>
<tr>
<td>13</td>
<td>C</td>
<td>HIS</td>
<td>A</td>
<td>19</td>
<td>14.765</td>
<td>68.754</td>
<td>30.893</td>
</tr>
<tr>
<td>23</td>
<td>C</td>
<td>GLU</td>
<td>A</td>
<td>20</td>
<td>17.327</td>
<td>69.446</td>
<td>33.609</td>
</tr>
<tr>
<td>32</td>
<td>C</td>
<td>GLU</td>
<td>A</td>
<td>21</td>
<td>19.913</td>
<td>71.318</td>
<td>31.511</td>
</tr>
<tr>
<td>41</td>
<td>C</td>
<td>GLU</td>
<td>A</td>
<td>22</td>
<td>17.278</td>
<td>73.664</td>
<td>30.123</td>
</tr>
<tr>
<td>50</td>
<td>C</td>
<td>GLN</td>
<td>A</td>
<td>23</td>
<td>15.880</td>
<td>74.276</td>
<td>33.602</td>
</tr>
</tbody>
</table>

...
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.