PSU-Alpha Documentation

Protein Structure Utility Version Alpha

Rachel Alcraft

Doc creation 6 Feb 2020
Doc update 1-Mar-20

Summary

PSU is a student first attempt at creating some protein structure analysis and creation tools based on the MSc Bioinformatics course at Birkbeck College 2019/2020.

The initial tools are to load pdb files and examine structure.

The project is written in C++ using Visual Studio, but I have put most of the classes in a shared library that should be easily portable into a CMake project. Data is output from the project in either a pdb format or a data frame format for loading into R.

The project is run simply from a light commandline exe using a config file for its run information and relying on configuration data.

Input Files

The config file lives next to the exe.

Within the config file 2 paths are set, for the inputs and for the outputs.

```
CONFIGPATH=F:\\PSUA\\Config\\
```

Within these locations the format is fixed as:

```
CONFIGPATH\\data_aminoinfo.csv    //r data frame format amino acid info REQUIRED
CONFIGPATH\\PDB\*.pdb             //where it looks for pdb files
```

Output Files

Within the config file 2 paths are set, for the inputs and for the outputs.

```
OUTPUTPATH=F:\\PSU\\Data\\
RUNID=TST1
```

Each time the utility is run an id is created based on the time. It can be overwritten if you don't want to create lots of directories (but then it overwrites results).

```
OUTPUTPATH\\RunId\\Logger.txt     //the log file is here
OUTPUTPATH\\RunId\\Reports\\       //All the reports and data are output here
```

Functionality

RAMA=TRUE
Ramachandran Plots, an R compatible data file is saved, with corresponding R script available for creating a Ramachandran plot, or any Chi plots and variations of. 3D psi/chi/omega available but not very interesting. Also has the secondary structure information based on the phi/psi angles which can be viewed in R.

CONTACT = TRUE
Contact map report based on pdb1 file. Currently not configurable because more data makes better plots in R, the distance is set quite high as the corresponding R report can filter to desired distance for display. High distance is useful for a heat map. The R report will show blobs coloured on distance or any other property of the amino acid of interest. As above, also has the secondary structure information based on the phi/psi angles which can be viewed in R.

CONTACTCHAIN1=A
CONTACTCHAIN2=B
In this case the contact map will be selected on each given chain. It will not look anything like a regular contact map, it won't be symmetrical for example. It is intended to show closeness of chains for Protein-Protein-Interaction where a complex is given. I have no idea if it means anything yet.

RMSDFIX=TRUE
An RMSD report will be run without moving the structures at all, so will just calculate on C-alphas in order up to the minimum possible – unless an alignment is either given or requested (not implemented that yet).

RMSDOPT=TRUE
ALIGNMENT=TRUE
Using the BTL library from Birkbeck (http://people.cryst.bbk.ac.uk/~classlib/bioinf/BTL05.html) (with permission), the structures are aligned and the rmsd calculated using Kearsley's method. PDB structure 1 is moved onto structure 2. If the Alignment=FALSE then it will take the minimum c-alphas and match on those, if it is true it will do a needleman-wunsch alignment from the BTL library, just using a simple gap penalty of 1.

RMSDCONTACT=TRUE
A contact map is done for the 2 structures against each other, not sure if it means anything.

Algorithms

RMSD
Uses BTL library implementation of Kearsley's matrix transformation. Optionally also with the Needleman-Wunsch algorithm for alignment.

Development Plans

Split space up into connected segments into which atoms are registered. Node types of both segments and atoms allows traversal. This could enable both structural creation and optimisation by notifying space, and possibly density checks for inside/outside/surface.

Probability distributions: could make distributions based on geometrical features as per the Top1000 prototype. Not all will be normally distributed. Could ALSO look at distributions for gaps between residues, eg find the probability distribution of the distance between CYS and MET when they are 11

residues apart. Then use a sort of dynamic programming grid kind of approach to find the likely distances between all residues in a sequence, with standard deviation. Then starting with the lowest sd build it up. This would mean some sort of SphereProximity class for every residue pair defining which would be a connected graph that could enable the structure to rejig with each new placement.

Visual interface is needed, consider https://www.qt.io/developers

Add a calculation for solvent accessible surface area (SASA). Code found here:
https://www.cgl.ucsf.edu/chimera/data/sasa-nov2013/sas_area.cpp
https://www.cgl.ucsf.edu/chimera/data/sasa-nov2013/sasa.html

Enable the rotation of 1 protein around another.

Add an algorithm to find an area of similar closeness when moving proteins around each other.

How do I distinguish atoms on the surface and core? How do I know for any point in space if it is inside or outside of a structure?

RMSD taking a FASTA format alignment or implement Needleman-Wunsch

Add in hydrogen bonds – will it depend on the quality/resolution of the structure?

Conversion between angles and coordinates of structure descriptions so I can move between 3d description and a creation from start in distance and angle.

Secondary structure prediction from sequence

Conformation optimised by energy calculations

Add a visual interface to watch conformations change (http://www.gnuplot.info/?). Include live updating of gnuplot:
http://hxcaine.com/blog/2013/02/28/running-gnuplot-as-a-live-graph-with-automatic-updates/

Generate my own 'forcefield' taking the top 1000(?) best structures from pdb, calculating all angles and bond lengths, and assuming those are the best structures so calculating parameters to fit them. Or is that a statistical model?

Geometric unravelling of sequence using SS prediction and most likely geometry from above.

Ball library has CHARMM energy model:
https://github.com/BALL-Project/ball/blob/master/data/CHARMM/param22.ini

References

Dihedral angles: http://www.ccp14.ac.uk/ccp/web-mirrors/garlic/garlic/commands/dihedrals.html

Amino Acid properties: http://www.imgt.org/IMGTeducation/Aide-memoire/_UK/aminoacids/IMGTclasses.html

Amino Acid structures: http://www.imgt.org/IMGTeducation/Aide-memoire/_UK/aminoacids/formuleAA/

SASA calculation (not implemented): https://www.cgl.ucsf.edu/chimera/data/sasa-nov2013/sasa.html

BTL Library: http://people.cryst.bbk.ac.uk/~classlib/bioinf/BTL05.html