Diagram, text

Description automatically generated

Circuit (localCT)

April 5, 2022

Documentation

Here are the scripts, for use in constructing circuit topology diagrams, matrices, and relation counts given a three-dimensional protein structure. Specifically, the code allows for extracting local topological information. Users are requested to cite the below article, which is associated with this package:

Jaie Woodard, Sumaiya Iqbal, Alireza Mashaghi\*

*Circuit topology predicts pathogenicity of missense mutations*

Proteins: Structure, Function, and Bioinformatics (2022)

Comments or bug reports (to Alireza Mashaghi’s group at Leiden University) are appreciated.

Author: Jaie Woodard

For additional code and a tutorial, see the publication below. Note that some functions supersede those of version 1.

O. Schullian, J. Woodard , A. Tirandaz, A. Mashaghi, A Circuit Topology Approach to Categorizing Changes in Biomolecular Structure. Front. Phys. 8 (2020) DOI: 10.3389/fphy.2020.00005.

**Obtain diagrams:**

**circuit\_diagram**

[cmap3, s, segment, numbering] = circuit\_diagram(pdb\_file, stride\_file)

* Default values chosen for other arguments

[cmap3, s, segment, numbering] = circuit\_diagram(pdb\_file, stride\_file, ss\_elements, cutoff\_distance, cutoff\_numcontacts, plotcolor, plot\_figs, plot\_circle, plot\_ions, plot\_disulfide)

Generates the contact map and arc representation of a protein, using secondary structural elements as the contacting units.

Input:

* pdb\_file is the Protein Data Bank PDB format file, with a single chain extracted and hydrogens removed.
* stride\_file is the secondary structure file from STRIDE (<http://webclu.bio.wzw.tum.de/stride/>).
* ss\_elements are the STRIDE secondary structure letters to include as secondary structure elements
* cutoff\_distance is the atom-atom cutoff distance in Angstroms
* cutoff\_numcontacts is the number of atom-atom contacts cutoff per secondary structural element pair
* plotcolor is the color of the arc plot
* plot\_figs (0 or 1) indicates whether figures should be plotted
* plot\_circle indicates whether a circle diagram should be drawn
* plot\_ions indicates whether through-ion contacts should be indicated under the diagram
* plot\_disulfide indicates whether disulfide bonds (cysteine sulfur contacts) should be indicated separately under the diagram

Output:

* cmap3 is the segment-segment contact map
* s is the secondary structure string
* segment is a residue string with each segment assigned a consecutive number
* numbering is the residue numbering of the PDB file

**circuit\_diagram\_residue**

* [cmap3, numbering] = circuit\_diagram\_residue(pdb\_file)
* [cmap3, numbering] = circuit\_diagram\_residue(pdb\_file, cutoff\_distance, cutoff\_numcontacts, plot\_figs)
* Generate the contact map and arc representation of a protein, using individual residues as the contacting units.

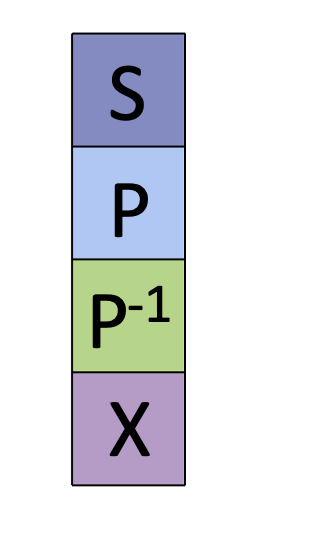
Output:

* cmap3 is the segment-segment contact map
* s is the secondary structure string
* segment is a residue string with each segment assigned a consecutive number
* numbering is the residue numbering of the PDB file

**get\_matrix**

[mat, c] = get\_matrix(cmap, plot\_figs)

Get the matrix of relations between contacts, given contact map. Numbered contacts are shown in the first figure, and the matrix is shown in the second figure. Legend is as shown below:



Input:

* cmap is the contact map generated by circuit\_diagram or circuit\_diagram\_residue (i.e., cmap3).
* plot\_figs (0 or 1) indicates whether figures should be plotted

Output:

* mat is the number coded matrix of relations
* c is the matrix of relations

**PyMOL visualization:**

**ssalter.m**

Generate PyMOL input for revising secondary structure. The sequence, s, and PDB file name must be specified. First, type

alter /,ss='L'

into PyMOL. Next, run ssalter and copy and paste the output into PyMOL. Finally, type

rebuild

into PyMOL.

**Local circuit topology and contact order**

**local\_topology.m**

Input:

* First line gives folder with contact maps

Output:

* Text files for each protein containing topology information for each residue

**local\_topology2.py**

Input:

* File with PDB ID, chain, and PDB residue number given, output of local\_topology.m

Output:

* Numbers of relations for all mutations

**Machine learning**

**pathogenicity.R**

Input:

* Excel table of features and pathogenicity.
* Decision tree and random forest results