CIRCUIT TOPOLOGY PYTHON TOOL

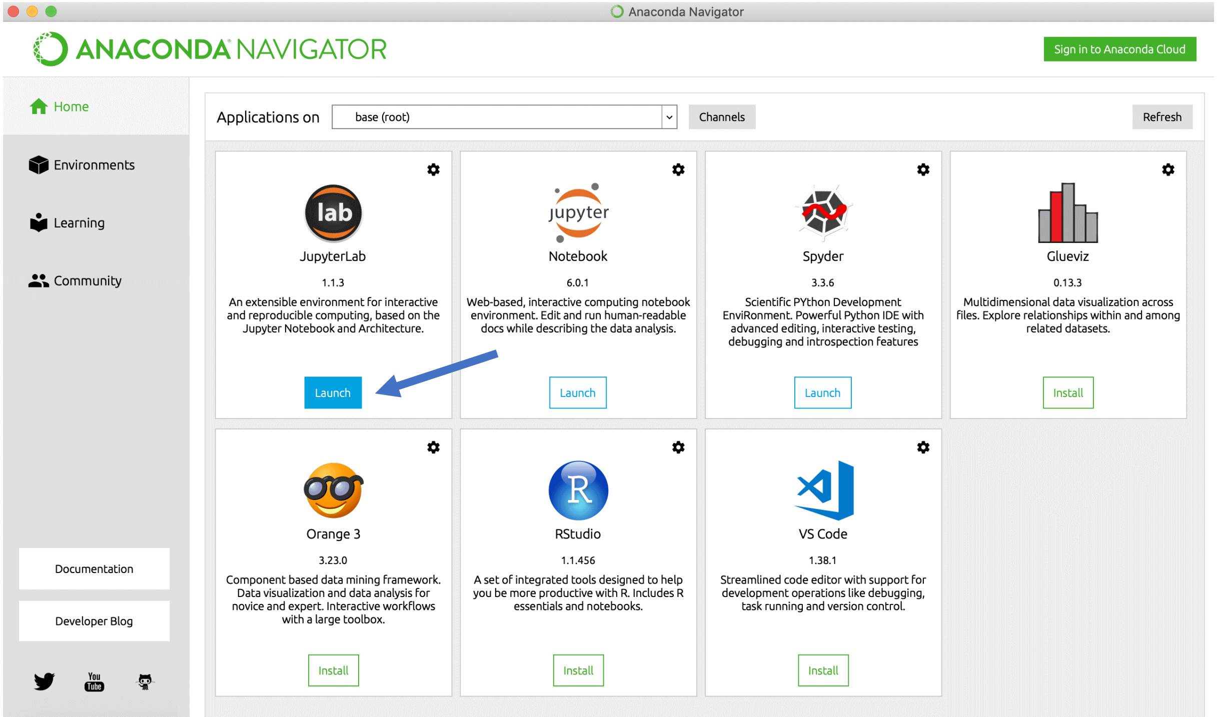
This is the documentation file for the circuit topology tool.
For any further questions regarding the script and/or suggestions:
moesduane@gmail.com

INSTALLATION GUIDE

1.Download and install anaconda from: https://www.anaconda.com/products/individual
Anaconda is an open-source python environment that allows for a very easy user experience.

2. Make sure you have downloaded the latest version of the CT tool from:
https://github.com/Duanetech/circuit_topology.git

3. Open anaconda and scroll down to find and launch Jupyter Lab.



An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

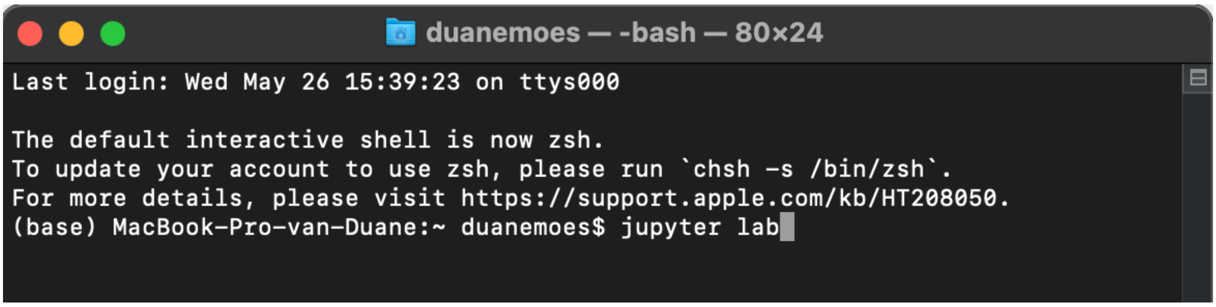
Multidimensional data visualization across files. Explore relationships within and among related datasets.

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Streamlined code editor with support for development operations like debugging, task running and version control.

Note! After launching the first time you can also start Jupyter without starting anaconda.
Just type Jupyter lab into your terminal and press enter.



```
Last login: Wed May 26 15:39:23 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) MacBook-Pro-van-Duane:~ duanemoes$ jupyter lab
```

4. Unpack the downloaded script and navigate to the map using jupyter lab
5. Open the main script called: MAIN.ipynb

FUNCTIONS

- **`retrieve_cif(prot_id)`**

  Downloads a mmCIF file from RCSB containing the specified protein.

  Parameters: *prot_id : string*

  4 letter protein code (not case sensitive)

  Returns: *none*

  No returns, it downloads a file to the `input_files/cif folder`

- **`retrieve_cif_list()`**

  Retrieves multiple proteins from a list of protein codes specified in `input_files/protlist.txt`

  Parameters: *none*

  Returns: *none*

  No returns, it downloads a file to the `input_files/cif folder`

- **`retrieve_chain(input_file, chainid = 0)`**

  Imports a pdb and creates a chain object used by BioPython to analyze structures and removes heteroatoms.

  Parameters: *input_file : string*

  Name of the PDB/mmCIF file.

  chainid : string & int

  Specify which chain to import. Either chain ID can be used or the number. Default is set to the first chain in the PDB file.

  Returns: *chain : object*

  Chain object used in the rest of the program.

  protid : string

  ID of the protein currently in the chain object, combined with the chain identifier.

- **`stride_secondary_struc(stride_file)`**

  Function that reads STRIDE files, that are downloaded externally and generate a readable secondary structure.

  # H - Alpha-Helix
  # b - Isolated Beta-Bridge
  # I - Pi helix
  # C - Coil
  # B - Isolated Beta-Bridge
  # G - 3-10 Helix
  # T - Turn

  Parameters: *stride_file : string*

  Name of the stride_file. NOTE: put STRIDE files in the `input_files/STRIDE` folder.

  Returns: *structure : string*

Secondary structure of the protein using one letter codes formatted in string

sequence : string

aminoacid sequence of the protein using one letter codes.

- **`Get_cmap`**`(chain, level = 'chain', cutoff_distance = 4.5, cutoff_numcontacts = 5,exclude_neighbours = 3)`

  Function that generates a Residue contact map for a single chain or a whole model. Note! returns of the indices of nonzero values in the contact map.

  Parameters: *chain : object*

  Chain object created in the `retrieve_chain()` function.

  level : string

  specify level of contact map. 'chain' for single chain analysis and 'model' for multi-chain analysis.

  cutoff_distance : float

  maximum distance (Ångström) between 2 atoms that will count as an atom-atom contact

  cutoff_numcontacts : int

  Minimum number of contacts between two residues to count as a residue contact.

  exclude_neighbours : int

  Number of residue neighbors to exclude from contacts.

  Returns: *index : array*

  Array consisting out of the indices of the res-res contacts

  numbering : array

  Original ID's of each of the residues

  protid : string

  ID of the protein + the chain used. When 'model' option activated, no chain identifier in the protid.

  resnames : array

  Three letter name code for each residue in the chain/model.

- **`Get_matrix`**`(index,protid)`

  Uses the index/cmap to create a topological relations matrix. Both chain index and model index are valid inputs

  Parameters: *index : array*

  Consisting out of indices for res-res contacts.

  protid : string

  Protein ID

  Returns: *mat : array*

Numerical topological relation matrix

Chain legend

0	-	diagonal	4	-	Cross contact
1	-	Serie contact	5	-	Concerted parallel contact
2	-	Parallel contact	6	-	Concerted parallel -1 contact
3	-	Parallel -1 contact	7	-	Concerted series contact

Model legend

0	-	diagonal	4	-	Independent intrachain
1	-	Parallel	5	-	T intrachain contact
2	-	Series	6	-	Loop intrachain contact
3	-	Cross			

stats : list

Either a list of the number of P,S and C contacts or a list of the number Of P,S,C,I,T,L contacts

- **`get_stats(mat)`**

  Calculates the amount of entangled contacts across the diagonal

  Parameters: *mat : array*

  Topological relations matrix retrieved from get_matrix

  Returns: *entangled : array*

  Percentage of entangled contacts (P & X) across the diagonal. Measure of globularity

- **`secondary_struc_cmap(chain,sequence, structure, cutoff_distance = 4.5, cutoff_numcontacts = 10, exclude_neighbour = 3, ss_elements)`**

  Function for creating a segment-segment contact map. Uses STRIDE secondary structure.

  Parameters: *chain : object*

  Chain object created in the `retrieve_chain()` function.

  cutoff_distance : float

  maximum distance (Ångström) between 2 atoms that will count as an atom-atom contact

  cutoff_numcontacts : int

  Minimum number of contacts between two segments to count as a segments contact.

  exclude_neighbours : int

  Number of residue neighbors to exclude from contacts.

  structure : string

  Secondary structure of the protein using one letter codes formatted in string

sequence : string

aminoacid sequence of the protein using one letter codes.

ss_elements : list

one letter codes of the types of secondary structures that are going to be counted as a segment.

Returns:

index : array

Indices of the segment-segment contacts. The different secondary structures are numbered and can be found in segment.

segment: array

Legend for the ID's of the secondary structures in the protein.

- **`Secondary_struc_filter`**`(index,structure, filtered_structure)`

Residue contact map filter that filters out res-res contacts located within the same secondary structure.

Parameters:

index : array

Residue contact map

structure : string

Secondary structure of the protein using one letter codes formatted in string

filtered_structures : list

One letter codes containing the secondary structures that are filtered out.

Returns:

index_filtered : array

Filtered index with specified secondary structures excluded

Struc_id : array

String containing the numerical ID's of the secondary structures in the proteins.

- **`energy_cmap`**`(index,numbering,res_names,protid, potential_sign = '+')`

Res contact map filter that filters out contacts based on positive/negative energy values retrieved from `input_files/matrix_potential.txt`

Parameters:

index : array

Residue contact map

numbering : array

Residue ID's of all the residues in chain

Res_names : array

Residue names of present aminoacids

protid : string
Protein ID

potential_sign: string
Option for defining positive ('+') or negative ('+') filtering

Returns:

energy_index : array
Energy filtered index

protid: string
Altered protein ID containing the filter identifier

- **`length_filtering`**`(index,distance,mode)`

  Function that filters out either long range contacts or long range contacts out of cmap

  Parameters:

  index : array
  Residue contact map

  mode : string
  Setting for only including short range contacts ('<') and excluding long range contacts or including long range contacts ('>') and excluding Short range contacts

  distance : array
  threshold for length filtering in the number of residues between contacts.

  Returns:

  entangled : array
  Percentage of entangled contacts (P & X) across the diagonal. Measure of globularity

- **`string_pdb`**`(index,numbering,threshold)`

  Function that uses Anatoly's circuit theory to define circuits within the CT and returns the amount of circuits

  Parameters:

  index : array
  Residue contact map

  numbering : array
  Numerical ID's for the residues in the chain

  threshold : int
  threshold for minimal length in residues a circuit can be

  Returns:

  segnums: int
  amount of circuits present in chain

meanlength: float
Meanlength of the circuits
segends: list
location of the circuits in the chain

- **`circuit_plot`**`(index, protid, numbering)`

  Function for creating a circuit topology diagram

  Parameters: *index : array*
  Residue contact map

  numbering : array
  Numerical ID's for the residues in the chain

  protid : string
  Protein ID

  Returns: *none*

- **`matrix_plot`**`(mat,protid)`

  Function for plotting a circuit topology relations matrix

  Parameters: *mat : array*
  Topological relations matrix retrieved from get_matrix

  protid : string
  Protein ID

  Returns: *none*

- **`matrix_plot_model`**`(mat,protid)`

  Function for plotting a circuit topology relations matrix. **Note!** This is specifically for matrices derived from a whole model/multi chain CT.

  Parameters: *mat : array*
  Topological relations matrix retrieved from get_matrix

  protid : string
  Protein ID

  Returns: *none*

- **`stats_plot`**`(entangled, psc ,protid)`

  Function for plotting a of the fractions of different types of CT relations and entanglement across the cmap diagonal

  Parameters: *entangled : array*
  Percentage of entangled contacts (P & X) across the diagonal.
  Measure of globularity

  protid : string
  Protein ID

psc : list
amount of parallel, series and cross contacts in chain

Returns: *none*

- **`export_cmap3`**`(index,protid,numbering)`

  Function for transforming the index to a residue contact map matrix and exporting it to a csv in `results/circuit_diagram`

  Parameters: *index : array*
  Residue contact map

  protid : string
  Protein ID

  numbering : array
  Numerical ID's for the residues in the chain

  Returns: *none*

- **`export_cmap4`**`(index,segment,structure,protid)`

  Function for transforming the index to a contact map matrix and exporting it to a csv in `results/circuit_diagram`

  Parameters: *index : array*
  segment-segment contact map

  segment: array
  Legend for the ID's of the secondary structures in the protein.

  protid : string
  Protein ID

  structure : string
  Secondary structure of the protein using one letter codes formatted in string

  Returns: *none*

- **`export_mat`**`(index,mat,protid)`

  Function for exporting topological relations matrix to csv located in `results/matrix`.

  Parameters: *index : array*
  Residue contact map

  mat : array
  Topological relations matrix retrieved from get_matrix

  protid : string
  Protein ID

  Returns: *none*

- **`export_psc`**`(psclist)`

  Function for exporting psc statistics to a csv located in `results/statistics`

  Parameters: *psclist : list*

  nested list of combined results containing protid,p,s,c

  Returns: *none*

- **`export_circuit`**`(circuitlist)`

  Function for exporting circuit statistics to a csv located in `results/circuit`

  Parameters: *circuitlist : list*

  nested list of combined results containing protid, segnums, meanlength, segends

  Returns: *none*