

Readme Candys

Ignacio del Amo

June 2021

Summary of important classes

- **CANetwork** class: object that represents the electrical circuits. The constructor methods of this class can generate networks with different topologies. It is also possible to stitch the network by adding nodes and links by hand.
- **Vnode** class: object that represents the nodes of the circuit. It has a state and keeps track of its neighbours.
- **Mlink** class: object that represents the links (memristors) of the circuit.
- **SetBuilder** class: reads fasta files to create data sets of proteins. Is important to remember that the method `changeSets` is what actually creates the datasets in the object.
- **IDrive** interface: This interface creates a template for drivers that drive the nodes. They do so by setting its state to a certain value every time step.
- **Protein** class: object that encodes a protein into a matrix of voltages. Implements **IDrive**.
- **CodedProtein** class: object that encodes a protein into a matrix of voltages. Implements **IDrive**.
- **DegreeSort** class: auxiliary class that sorts nodes in descending order by degree.
- **GeneticSearch** class: implements a genetic algorithm to explore the space of network topologies.
- **TestWAlg** class: used for debugging. Not really important

This package also uses the Universal Java Matrix Package (<https://ujmp.org/>). For reasons of time we cannot give a detail explanation of the methods contained in each class and the use of them. For more information please contact `iamoblanco \at gmail\dot com`.