BurritoLy

Jon Ngo
Pablo Saucedo
Kyra Scott

Assembling parts can lead to unintentional combinations where, if created, the resulting protein would not function properly or would be a pathogen. BurritoLy‘s goal is to be used as a background tool, similar to a spell check tool that runs on a word processor application, that checks builds for bad combinations of parts. The user imports a custom parts library as a .JSON file, and then any parts that are flagged as bad are removed. When parts are added in combination, the tool iteratively checks through every part and removes them as necessary, preemptively preventing the user from creating bad parts. Parts are bad when they match our mock IGSC database that is based off of the database. Their current assembly is saved as a query, which is the sequence of amino acids.

The majority of the UI is on BurritoLy.py, which is how the main application is run. The fake IGSC databases are created in a script in Fake Database, and demo / test cases are created in Demo Libraries. The other python scripts (‘functions.py, interface.py, state_class.py’) make up the majority of the algorithm. These run BLAST on the list of parts being made and save user history.