Plasmid backbone gene/protein naming project: workflow

Zac Lindsey (June 2016)

We are trying to find out which plasmid backbone genes have different names, but are actually the same or similar according to homology. Each protein has numerous distinct names due to a lack of standard naming conventions. We will use usearch to perform clustering on amino acid sequences of a large list of plasmid backbone genes, and then see which 4-letter names are in each cluster. We will be looking primarily at 4-letter names, and will hopefully be able to come up with a better naming convention so that the proteins that have many names can be consolidated.

1. **Download all plasmids from NCBI** which have sequence length between 20 and 200 kbp. Make sure they have the tags “plasmid” and “complete.” Download all of them as a Genbank file. We will call this “Plasmids.gb.”
2. **In ProtAnalysis.py, run extract_plnames_from_file** using Plasmids.gb as the input parameter. This function generates a list of names called “Plasmids.csv.”
3. **In ProtAnalysis.py, run StripCDSAddInc** using Plasmids.gb and Plasmids.csv as the input arguments.
    a. There may be errors depending on how the Genbank file was formatted. This function is supposed to add the incompatibility group to each name, but sometimes the inc-group is not available, and this might throw an index error in Python if it can not find the inc-group when the file is parsed.
    b. StripCDSAddInc generates a file called PlasmidsAA.fa. It represents all of the protein sequences within the plasmids’ genomes.
    c. I have modified the function so that the name of each protein in PlasmidsAA is the protein name followed by the plasmid name, with an underscore separating the two.
    d. I made a version which does not include the incompatibility group because the vast majority of these were “IncUnknown.”
4. **Run usearch** on PlasmidsAA.fa with 70-70 arguments.
    a. In ProtAnalysis.py, this step will generate a tab file (which actually has the format of a .uc file) called PlasmidsClusters.tab or something similar.
        i. http://www.drive5.com/usearch/manual/opt_uc.html
    b. It will also generate a fasta file containing the sequences of the centroid proteins.
5. We suspect that many of the proteins in each cluster have different names but are actually the same or similar in homology. We will look at a list of about 100 plasmid backbone protein 4-letter names. We will try to find these 4-letter names in the cluster file that we made. If one of these 4-letter names if found in any cluster, we will add all of the other proteins in this cluster to the 4-letter name’s corresponding “Group.”

Grouping Algorithm

1. **Run gather_clusters_R_ZL() using a Backbone.csv** of the primary 4-letter names we want to look at, as well as the cluster file from usearch, as input arguments.
    a. The program adds all of the rows from the USEARCH cluster file to a list -- sRows.
    b. The program loops through each of these rows.
        i. If a 4-letter protein name from Backbone.csv is found in any row, the cluster ID number is recorded for this row.
        ii. All protein names from rows with this cluster ID are added to the list of names for this Backbone protein name. This means that each list contains not only names which contained the 4-letter name but also other names that were clustered with these names.
    c. **You will end up with a list of N dictionaries called Names**, where N is the number of 4-letter proteins imported from Backbones.csv. Each dictionary contains a certain number of cluster IDs. The values corresponding to each cluster ID are a “set” of different protein names [all of the names associated with this ID].
    d. You will also have a large dictionary of protein names called ProtInfo. Each key is a protein name, and each value is a list of two elements: sequence length and percent similarity to the centroid.
2. The final output will be contained in a file - **ClusterGroups.csv by default.**
    a. Column 1: 4-letter reference protein names from Backbones.csv
    b. Column 2: Cluster ID numbers
    c. Column 3: Different protein names from clusters
    d. Column 4: Sequence Length for protein corresponding to col. 3
    e. Column 5: Percent similarity of this protein with respect to its cluster centroid

Working in R - generate plots

1. Get the code on Github (https://github.com/ZacLindsey/ProteinNaming).
2. Run the Shiny App.
3. I used ggplot2, shiny, seqinr, and Rmisc packages.