Python Project Outline

What is the field of operation of your program?

It is a combination between a data processing and visualization tool. It will be in the form of a pipeline and will consist of modules that the user can decide to call or not to call.

What is the name of your project?

find_best_genomebins.py

What is your project going to do? Why is it going to be useful? Why would someone want to use it?

My project is to create a script (with associated modules) that takes input genome bins, finds ones with good statistics, does some basic processing of the bins and outputs summary/descriptive information about them in a nice graphical/tabular output.

There is a large issue in genomics of huge datasets that are difficult to work with manually. Automation of data processing steps where a "human eye" is not essential can greatly increase effective use of data generated. There are a number of methods for generating genome bins from metagenomic assemblies (ex. PhymmBL, ESOM, CONCOCT). Some of these tools require manual delimitation of putative genome bins (re. using ESOM maps), however others automatically assign contigs from metagenomic assemblies to such putative bins (PhymmBL, CONCOCT). Hundreds and even thousands of potential genome bins are generated in this way. But how to effectively manage and explore this data?

Many researchers only extract genome bins of interest, but what about the rest of the data? It would be useful to have a tool to explore genome bins generated without predefining what you are looking for. This script will extract any genome bins above a certain quality threshold (which the user will be able to specify (or not include if they would like to process everything), for example: completeness > 70% and redundancy < 1.2). The script will then generate the summary information for each bin including number of contigs, GC content, rRNA, ORFs and putative taxonomy. Files containing the contigs for each bin, ORFs and rRNA will also be generated.

This data will then be displayed graphically to give a general overview of the high quality bins. A summary data table across all the bins of good quality will also be generated.

Possible future expansions:

- Make the graphical output interactive
- Estimate total genome length and gene number
- Find tRNA genes
- Estimate gene density
- Coverage estimation from Kallisto

What are the inputs and what are the outputs of your program?

Main Goal: by the end of this project to have a script that takes an input assembly and file containing contig assignments (from binning software), and generates bins. The script will then filter bins based on user-specified completeness and redundancy. Basic descriptive data for each bin will then be generated and displayed in both a table and graphically.

*Anything with a star will only be completed if there is time. The main focus will be to build a script with associated modules that can later be expanded with more time.

Inputs:

- Metagenomic assembly (fasta file of contigs)
- Contig assignment file (contig anme and bin # assigned to – comma OR tab seperated)
- Database of rRNA (SILVA SSU/LSU database: https://www.arb-silva.de)*

Outputs:

- Bins
  - fasta file containing contigs
  - faa file containing ORFs*
  - fasta file containing rRNA sequences*
- Summary Information
  - Plot showing distribution of genomes extracted across phyla*
  - Plot showing distribution of genome completeness by genome redundancy
  - Plot showing distribution of length by GC
  - file with all summary data in both tabular and html format
    - bin number
    - taxonomic classification *
    - number of contigs
    - N50
    - Genome bin length
    - GC content
    - Number of ORFs*
    - Number of 16S rRNA*
    - Number of 23S rRNA*
    - Completeness
    - Redundancy

Programs

- Rnammer or Barrnap*
- Prodigal (http://prodigal.ornl.gov)*
- CheckM (http://ecogenomics.github.io/CheckM/)
- CONCOCT – RPSBLAST* (https://concoct.readthedocs.io/en/latest/complete_example.html#validation-using-single-copy-core-genes)

What are the main objects? Write the name of the main classes you are going to use in your project, as well as the name of the attributes and methods they will have.

*Only completed for the most basic form of the script

Classes:
    Attributes:
        Methods:

- Assembly
    - Self
    - Fasta_file
    - Binning file
        - Extract bins
- Bin
    - Self
    - Name (Bin Number)
    - Contigs
    - GC
    - Length
    - N50
        - Calculate GC
        - Calculate Length
        - Calculate N50
        - Calculate Completeness and Redundancy
        - Generate Descriptive Data
        - Generate Distribution Plots
- Contig
    - Self
    - Name (Contig Number)
    - Length
    - GC content
        - Calculate GC
        - Calculate Length

Once you have the name of the classes and methods, write some pseudo code before starting with the real implementation.

*Work in progress

```
import sys
import os
import rufus #will maybe use
import argparser as ar
import matplotlib.pyplot as py

class Assembly(object):

class Bin(object):
```

class Contig(object):

Describe a user case. You have to make up a fictional character named "Bob" who wants to use your project. Describe his ideas and his goals, and tell us how he is going to use your project in a few sentences. Also describe the results he will obtain.

***Only completed for the most basic form of the program

Bob is having problem's (well more than normal…). He is exploring novel microbial genome diversity, and is interesting in extracting high-quality genome bins for this.
BUT he's realized that his recent run of CONCOCT has resulting in over 1000 genome bins! To go through all of them one by one and by hand would take him a number of weeks and his supervisor wants to know which bins could be useful by their next meeting in two days! What shall poor Bob do? Well he will use find_best_genomebins.py of course!

Bob will only need to provide his initial metagenome assembly and the output file from CONCOCT which contains assignments of each contig to a bin.

Bob inputs his data into the script and behold within an hour he has the contigs associated with each bin that is high quality (> 70% completeness and < 1.2 re) extracted and in separate folders. In the parent folder with all of the bins he also finds a summary html file with plots showing the distribution of some main characteristics of the bins (size, GC content, redundancy, completeness) and a table outlining information for all of the bins (which also finds in a tab-separated txt format in another file). He now has 114 high-quality bins extracted from his dataset that he can show to his supervisor at his meeting in two days. And now he can go take a nap instead of pulling a few all-nighters to get the data processed…