Summary of steps to creating the model:

Step 0 - Preprocessing of training set

Step 1 - Generate input features:

- 1a. Run the PARTIE tool
- 1b. Estimate the human DNA percentage
- 1c. Run the FOCUS tool & calculate the percentages of bacterial reads in the PARTIE coming from each source category
- 1d. Parse the outputs of steps 1A-1C to include only the metagenomes from step 0. (those in the training set) and combine into a single table.

Step 2 - Create Random Forest Model:

- 2a. Scale the data from step 1D
- 2b. Generate a random forest classifier using the scaled input data

Step 3 - Summarize model (visualization):

- 3a. Obtain the error rates for each level of the model (sources) and create a plot of these

Step 4 - optional PCA analysis:

- 4a. Scale the data from step 1D and perform PCA
- 4b. Isolate single levels of the model (sources) and plot these

Detailed steps to creating the model (protocol)

Step 0. Preprocessing of training set:

To begin the creation of the model, a simple CSV file that contains two columns of information is required. The columns should be as follows: 1) Unique Identifier of the genome (SRA ID), can be single entry or comma-separated list 2) Ground truth value (source location)

To do this: Download the entire set of classified genomes here: https://docs.google.com/spreadsheets/d/1-8Luo0nnqH7XrKwXKNLXAVIx494VLLzkk0F7c_Vcrx4/edit#gid=1463266187 (Note: You may need to request permission in order to view or edit this spreadsheet.)

Retain only the 4th and 6th column, the ID's and the "ground truth values" from this spreadsheet. This can be visually using excel or buy using:

```
 awk '{print $4,$6}' myfile > outfile
```

You will also want to remove the extra rows from the file that have no ground truth value. Again, this can be done visually in Excel or using the command:

```
awk '$6!=""' myfile > outfile
```

Step 1 - Generate input features:

- 1a. Run the PARTIE tool:

  A spreadsheet containing the PARTIE output for many metagenomes from the SRA database is located here: https://raw.githubusercontent.com/linsalrob/partie/master/SRA_PARTIE_DATA.txt

  See the PARTIE documentation manual for details on how to run this tool.

- 1b. Estimate of the human DNA percentage

  This was done using Bowtie2 to map the reads from the metagenomes to the human reference genome (obtained from https://www.ncbi.nlm.nih.gov/grc/human).

  A spreadsheet containing the both the PARTIE output (described in the previous step) as well as the estimated percentage of human DNA is found here:
  https://github.com/linsalrob/partie/blob/working/SRA_PARTIE_DATA.txt
  (note that this is a "working" branch rather than a master branch and is subject to change)
- 1c. Run the FOCUS tool & calculate the percentages of bacterial reads in the PARTIE coming from each source category
  - For details on running the FOCUS tool in this context see the FOCUS manual or check with Rob Edwards.
  - To calculate the percentages of bacterial reads and parse the FOCUS output after having been run using custom databases containing the relevant bacteria from PATRIC:
    - Use "cd" command to enter the directory containing the tar archives.
    - Unpack the files: for f in *.tgz; do tar -xzvf "$f" -C **/path/to/your/directory/focus/**; done
    - Use cd to enter your working directory (the one you unpacked the files to in the previous step)
    - Rename the tar files to that they are compatible with the parser script: for i in *.focus/output_Strain_tabular.xls;do for j in $(echo $i | cut -d '.' -f1);do cp $i **/path/to/your/directory/focus/**$j.focus.tsv ;done;done
    - To remove the full output files (optional): rm -rf *.focus/
    - Parse the "strain level" files to obtain a table of all the results by running the *focus_parser.py* script. Both arguments are required: Python focus_parser.py -d focus/ -o output_file_focus_data.tsv
      - Arguments: -d → the directory containing the focus output (results of step 2). It must be empty except for the results and the filenames must contain the genome ID followed by a file extension (examples:, genome1.tsv, 123545345345.sra.genome.tsv, etc.) -o output filename, this may also be a full path if you do not want to write the output to the current directory, which is default. (Examples: output_file.tsv, /myhomedir/focus_folder/output_file.tsv, etc.)

        - Convert this to “environmental” counts by modifying line 47 of the script *parsed_focus_to_environment.py* (must download from GitHub at https://github.com/linsalrob/partie_hat/blob/master/parsed_focus_to_environment.py). The modified line looks like this: thisenvironment = envs.get('PATRIC|'+str(p[0]), "UNKNOWN") The script requires an additional input file, a tsv file called patric_genomes_metadata.tsv (https://raw.githubusercontent.com/linsalrob/partie_hat/master/patric_data/patric_metadata_isolation_host_env.tsv) Run like so:

          ```
          Python3.7  parsed_focus_to_environment.py -f focus_round2_data.tsv -e ~/focus_data/patric_genomes_metadata.tsv -o metagenomes_envs.json
          ```

- 1d. Parse the outputs of steps 1A-1C to include only the metagenomes from step 0. (those in the training set) and combine into a single table by running the script *create_rf_training_data.py.* This requires 3 input files (generated from step0, step 1a/b, and step 1c) and is run like so:

  ```
  Python create_rf_training_data.py -ids training_set_ids_from_step0.csv -focus output_focus_from_step_1d.json -partie output_partie_from_step1a_b.csv -runs 50 -out outfile.tsv
  ```

Step 2 - Create Random Forest Model and visualize results:

- Scale the data from step 1D
- Generate a random forest classifier using the scaled input data
- Obtain the error rates for each level of the model (sources) and create a plot of these

All three of these steps are performed using the script *random_forest_unbalanced.R*.

You will need to change the output and input file paths manually in this code before running it.

The input file here is the output of step 1d, the output is a picture file (png) of the bar graph of the within group error rates.

Step 3 – (optional) PCA analysis:

- 3a. Scale the data from step 1D and perform PCA
- 3b. Isolate single levels of the model (sources) and plot these

These steps are performed by running the code: pca_by_environment.R

You will need to change the output and input file paths as well as the name of the source environment that you wish to plot manually in the code. The input file here is the output of Step 1d. The output is a picture file (png) of the PCA scatter plot.