Authors:

Brian Muegge and Jeremiah Faith (v0.2/0.3)
Patrick Degnan (v1)

1. Rationale 1
2. Installing Microbialomics 2
3. Running Microbialomics 3

1. Rationale

The initial impetus for this series of program tools was to aggregate Human Gut Microbial Genomes sequenced by various groups and organize them in a MySQL database using a common set of functional annotations. This database could then be used as the backend of a graphical genome browser (e.g., LWGV, GMOD) or other tools to analyze and understand genomes. These scripts have been adapted and used successfully on multiple campus clusters and large multiprocessor servers. These scripts are provided as-is and will require familiarity with PERL coding to be run on new platforms. Further, given changes to GenBank default formats since the initial coding of these programs, the continued use of these programs is questionable. However, they are provided here for legacy purposes.

2. Installing Microbialomics

Microbialomics is a PERL program that can take a NCBI RefSeq genome. Furthermore, additional data types can be utilized to filter the results including expression differences, known binding sites, operon predictions and window size of possible binding sites.

As written the program can run on any Unix/Linux based system, however it has a number of dependencies.

First, download and install the following software tools and all of their dependencies according the authors' instructions:

- HMMER3
- Prodigal
- tRNAscan-SE
- Infernal
- RNAmmer
- CELLO
- RNIE

Second, download the relevant databases:

- String Protein DB
- KEGG Protein DB
- TIGRFAMs
- PFAM
- RFAM

Microbialomics can work with local genome assemblies. However, to access genomes from NCBI install the `efetch` program from the Entrez Direct (`edirect`) toolkit.

- edirect

Retrieve and decompress the **Microbialomics** directory from GitHub containing core pipeline script and its additional required support PERL scripts.

- Microbialomics

Make sure **Microbialomics** and all of the programs are in your user path. Modify the core pipeline script with the absolute path locations for all of the external tools, databases and other support PERL scripts.

3. Running Microbialomics

1A. Download and Check data from NCBI RefSeq [`pre_process_genomes.pl`]

Download RefSeq genomes from NCBI using FTP (https://www.ncbi.nlm.nih.gov/genome/browse/#!/overview/), eftech or other platform.

Note that this pipeline was developed prior to changes in GenBank's formatting changes to default files provided and the change to their FTP addresses and still requires *ptt, *faa, *rnt, *fna, *gbk input files for RefSeq genomes. Current *gbff files can suffice, however some modifications will be required to the scripts (particularly glob and regex commands).

Check the completeness of the data files

```
pre_process_genomes.pl
usage: pre_process_genomes.pl <list of folder names>
```

Purpose: For incomplete GenBank / scaffolded genomes, confirm proper file structure & compression

Expects: *ptt, *faa, *rnt, *fna, *gbk
Make missing *ptt files from *gbk files
Concatenate *ptt files and *faa files into single files

1B. Prep multisequence genome scaffolds or contigs file

Concatenate scaffolds/contigs into a multisequence fasta file in its own subdirectory.

2. Run initial annotations on scaffolds [process_new_genome.pl]. This script was initially developed by J. Faith and B. Muegge while at Wash U. Several parts have been updated or changed. Currently it uses:

- tRNAscan to search for tRNAs
- rnammer to search for 5S, 16S and 23S rRNA
- Infernal to search for known riboswitch and conserved sRNA models
- Prodigal protein coding sequence (CDS/ORF) prediction
- Predicted proteins are then searched against using hmmer3:
  - PFAM
  - TIGRFAM
  - String database
  - KEGG

Note that String and KEGG searches take the longest.

Implementation:

```
$ process_new_genome.pl

Usage: perl process_new_genome.pl
      -i refseq_accession or fasta_file_with_contigs
      -o outfile basename (all outfiles will be called
            basename.some_suffix)
      -n genome_name
      -g gene_prefix (e.g. BAC will have gene names BAC0001, BAC0002,
            etc...)
      -t trim_size (i.e. discard contigs shorter than this; default
            300bp)
      -k kingdom (bac, arc); for bacteria also use bacn for gram- and
            bacp for gram+ if you want to run CELLO too
      -a Find genes only? or Rename GenBank files only? [Yy]es (default
            = No)
      -m Is this a metagenomic sample? [Yy]es (default = No, single
            genome)
```

Example run on metagenomic contigs:

```
$ process_new_genome.pl -i J11b_unknown.fna -o J11b -n "J11b water
      sample metagenome" -k bac -g J11b
```

Example for scaffolded genome:

```
$ process_new_genome.pl -i C09_genome.fna -o C09 -n "C09 Bacteroides
      thetaiotaomicron str. VPI 12345, draft genome" -k bacn -g C09
```

Example for RefSeq file:

```
$ process_new_genome.pl -i NC_004663 -o NC_004663 -k bacn
```

Notes on program:
Folder must contain for each Genbank genome the .ptt, .rnt, .fna, .gbk and .faa files.

Use 'NC_' or 'NZ_' as prefix to convince script that it is a genbank file. *gbk, *rnt and *fna files largely ignored by program
"ann_XXXXX" directory made with results of searches

Matches GI numbers in genbank *faa file to GI numbers in *ptt file and labels *faa with Locus IDs (no underscores)

*faa

```
>gi|384895179|ref|YP_005769168.1| hypothetical protein [Helicobacter pylori
35A]
MRRSLKNKGSIFSASNPNKEEQRHAEKIKNIKQLIASGF
>gi|384895180|ref|YP_005769169.1| hypothetical protein [Helicobacter pylori
35A]
MGFQNENQLKVGALVKATINDKVVEAKVISIGFNRVTLRSEKGNEVSYAFNSEKFLKWFNHTPLSEVAKN
HAESGNKDILDGVKIVTSGPTIKEMTTTPKEKEDRFKLAFGFRGVEEEGVSVSEVMISDYTLTERKSRLG
```

*ptt

```
Helicobacter pylori 35A chromosome, complete genome - 1..1566655
1470 proteins
Location Strand   Length    PID      Gene     Synonym  Code     COG      Product
183..302 +        39        384895179-        HMPREF4655_20002  -        -        hypothetical protein
326..1225+        299       384895180-        HMPREF4655_20003  -        -        hypothetical protein
```

New *faa file (no suffix) put in the "ann_XXX" directory

```
>HMPREF465520002
MRRSLKNKGSIFSASNPNKEEQRHAEKIKNIKQLIASGF
>HMPREF465520003
MGFQNENQLKVGALVKATINDKVVEAKVISIGFNRVTLRSEKGNEVSYAFNSEKFLKWFNHTPLSEVAKN
HAESGNKDILDGVKIVTSGPTIKEMTTTPKEKEDRFKLAFGFRGVEEEGVSVSEVMISDYTLTERKSRLG
VLLSPMLYSGNGSQISALIITALANAKGFNKHSDAEWFKMIEARNEDECEVDTFDNLDRVVLTLYCNVIK
AYAEWREEFQNDFNDFSPSGFWAQVLPKNKNEALFVAQLLCDGGINKYGLSCAGLTENLLKDVELTFGLA
TPSEIDEYLANLDKEGEVE
```

Separate GenBank accessions such as an organism that has a genome + a plasmid, they should be run in separate folders. Otherwise, plasmid will be concatenated with genome.

3A. Clean up and zip RefSeq files

```
post_process_genomes.pl
usage: post_process_genomes.pl <list of folder names>
```

Purpose: Clean up output of run for incomplete GenBank / scaffolded genomes

1. Compress extraneous files,
2. make *contig file
3. Make single *ptt and *rnt files for each genome

4. MOVE ALL FILES to one place & double check contents.

With one genome per directory the files after running steps 1-3 should be as follows:

```
NZ_ACTM00000000/
NZ_ACTM00000000.ptt
NZ_ACTM00000000.contig
NZ_ACTM00000000.fna
NZ_ACTM00000000.faa
NZ_ACTM00000000.gbk
NZ_ACTM00000000.rnt
      ann_*/
      NZ_ACTM00000000
      NZ_ACTM00000000.CELLO
      NZ_ACTM00000000.KEGG
      NZ_ACTM00000000.PFAM
      NZ_ACTM00000000.TIGRFAM
      NZ_ACTM00000000.stringCOG
```

For purposes of uploading the data to MYSQL all files are expected to be in the same directory.

The use of the following script is mainly if you have lots of genomes you have annotated and want to upload. If you do not have a lot of genomes, you can just do it manually and forgo this script.

```
$ mkdir /data/new_genomes/ANNO_GENOMES
```

Make text file:

```
$ nedit list.txt
NZ_ACTM00000000
NC_04663
...
```

Script has to be modified based on your naming conventions:

```
$ condense_files.pl list.txt
```

Copy script and modify as necessary.

5. SETUP MYSQL database.

Format database:

```
$ mysql -u user -p DATABASENAME < create_microbialomics_schema1.sql
Enter password:
```

6. Steps are for uploading genomes into MySQL:

If you have more than one genome to add I suggest first test loading a single genome before using the multi upload script below.

```
$ load_completed_genome3.pl
      usage: load_completed_genome.pl -a <accession(s) comma separated> -DB
[database] -c [clear database] -dir [annotation file directory]

$ load_completed_genome3.pl -a C09 -dir C09 -DB DATABASENAME
```

Program needs to be run from local directory containing all of the *faa, *fna, *ptt, *rnt, and annotation files

If problems arise, overwrite database and return to load_complete_genome3.pl:

```
$ mysql -u user -p  DATABASENAME < create_microbialomics_schema1.sql
```

7. MULTI Upload to MYSQL database

This takes a little while. If working remotely - start a screen job:

```
$ screen

$ multi_load.pl
usage: multi_load.pl <list of folder names> <DB>

$ multi_load.pl list.txt DATABASENAME &> load.log &

ctrl-a d
 logout
```

Check load.log when complete for possible errors

Update build table

```
$ mysql -u user -p DATABASENAME
```

Enter password:

```
> use DATABASENAME;

> insert into build (build_id,build_name,build_date,description) values (1,
DATABASENAME ','2014-07-30','Initial annotations of gut microbe genomes');

> quit;
```