Ulana Documentation

Overview:

Unicellular Long-read Assembly aNd Annotation

Ulana is a bacterial genome assembly and annotation pipeline using Fast, HAC or SUP ONT basecalled data from MinION and Flongle flow cells. The pipeline will 1) filter reads using NanoFilt 2) create a draft assembly using Flye 3) polish the draft assembly using Medaka 4) annotate the polished assembly using Prokka 5) summarize the Prokka output and provide a tentative 16S + rpoB + dnaA identification using ropro.

ulana:
vt. To plait, weave, knit, braid; plaiting, weaving. Also unala, nala, unana. Mea ulana ʻia, plaited or woven material, textile. Mea ulana lole, weaver (Isa. 38.12), loom. (PPN langa.)

##

Resources

Github:
ulana: https://github.com/ehill-iolani/ulana.git

Docker:
ulana: https://hub.docker.com/repository/docker/ethill/ulana
##

*****ANYTHING MARKED IN** `COURIER` **SHOULD BE ENTERED INTO THE TERMINAL*****

General setup instructions:

Ulana installation:

1) Open the terminal on your computer
    a) Search "terminal" on the search bar of your computer
2) Paste this line into your terminal and wait for your download to finish:

```
docker pull ethill/ulana:latest
```

Running an ulana container:

1) Go to docker desktop and do the following:
    a) Select images from the left hand sidebar
    b) Find the image labeled "ethill/ulana:latest"
    c) Select the "play" triangle on the far right of the image
    d) Drop down additional settings
        i) Give the container a name
        ii) Select the ". . ." under "host path" to specify where the data is stored
        iii) Input the following under "container path"

```
/home/data
```

2) Select "run"
3) In your terminal paste this line which has been MODIFIED to match your container name

```
docker exec -it yourcontainername bash
```

Pipeline specific instructions:

##

*****IMPORTANT*****

Make sure to adjust the switches in the code ACCORDINGLY for YOUR analysis

`-h` help (prints this message)
`-v` version
`-q` minimum quality score; default is 10
`-l` minimum read length; default is 1000
`-c` number of cores to use; default is 4
`-i` name of input fastq file containing reads
`-b` type of basecalling used; the options are: r941_min_fast_g507, r941_min_hac_g507, r941_min_sup_g507

##

Executing a Ulana analysis:

1) Once you are in the container navigate to the data folder
   ```
   /home/data
   ```

2) Once you are in the data folder, create a new directories for the analysis
   ```
   mkdir ulana_analysis
   mkdir ulana_analysis/fastq
   ```

3) Move the relevant fastq's into `ulana_analysis/fastq`
   ```
   cp data/barcodexx/*.fastq.gz ulana_analysis/fastq
   ```

4) Change directory into ulana_analysis
   ```
   cd /home/data/ulana_analysis
   ```

5) Unzip the the fastq files if they have a .gz file ending
   ```
   gunzip ./fastq/*.gz
   ```

6) Concatenate the fastq files into 1 fastq file
   ```
   cat ./fastq/*.fastq > ./samplename_combined.fastq
   ```

7) Once you have the concatenated fastq file chose one of the following to run based on the basecalling algorithm used:
   a) Fast:
```
ulana -q 8 -l 1000 -c 56 -i samplename_combined.fastq -b r941_min_fast_g507
```
   b) High-accuracy (HAC):
```
ulana -q 9 -l 1000 -c 56 -i samplename_combined.fastq -b r941_min_hac_g507
```
   c) Super accuracy (SUP):
```
ulana -q 10 -l 1000 -c 56 -i samplename_combined.fastq -b r941_min_sup_g507
```

8) Once the pipeline completes completes, your working directory should contain the following new directories:

```
├── flye_assembly
│   ├── 00-assembly
│   ├── 10-consensus
│   ├── 20-repeat
│   ├── 30-contigger
│   └── 40-polishing
├── medaka
├── prokka_out
└── ropro_out
    └── seqs_species_identifiers
```

9) The polished assembly is in the `medaka` directory and is marked:

```
medaka
├── calls_to_draft.bam
├── calls_to_draft.bam.bai
├── consensus.fasta
├── consensus.fasta.gaps_in_draft_coords.bed
└── consensus_probs.hdf
```

10) The tabulated version of the annotation is in the `prokka_out` directory and is marked:

```
prokka_out
├── samplename.err
├── samplename.faa
├── samplename.ffn
├── samplename.ffn.fai
├── samplename.fna
├── samplename.fsa
├── samplename.gbk
├── samplename.gff
├── samplename.log
├── samplename.sqn
├── samplename.tbl
├── samplename.tsv
└── samplename.txt
```

11) A quick summary of the identification is in `ropro_out` and is marked

```
ropro_out
├── report_out.txt
├── ropro.log
└── seqs_species_identifiers
    ├── HMABPGHC_00407_16S.fa
    ├── HMABPGHC_00926_dnaA.fa
    ├── HMABPGHC_01115_16S.fa
    ├── HMABPGHC_03244_rpoB.fa
    └── HMABPGHC_05326_dnaA.fa
```

12) The extracted identifier genes are in `ropro_out/seqs_species_identifiers` and are marked:

```
ropro_out
├── report_out.txt
├── ropro.log
└── seqs_species_identifiers
    ├── HMABPGHC_00407_16S.fa
    ├── HMABPGHC_00926_dnaA.fa
    ├── HMABPGHC_01115_16S.fa
    ├── HMABPGHC_03244_rpoB.fa
    └── HMABPGHC_05326_dnaA.fa
```