mitoGenome Assembly

Ian G. Brennan

22 April 2018

Contents

Install the software **2**

Multiple mtGenome assembly and alignment using MITObim in R **2**

File Preparation 2

Executing the Code 3

Install the software

1. MITObim[1]: https://github.com/chrishah/MITObim
   - make sure to follow install instructions, including dependecies: MIRA, PERL, etc.
2. MUSCLE[2]: https://www.drive5.com/muscle/

Multiple mtGenome assembly and alignment using MITObim in R

File Preparation

We'll use the MITObim pipeline, run from R to assemble our mtGenomes against a reference. We'll then pull out the final assembly for each sample and align them against each other using MUSCLE.
Start by creating a directory to hold:

- the MITObim pipeline ('MITObim.pl')
- the MIRA sequence assembler and mapper directory ('mira.4.0.2...')
- the MUSCLE executable ('muscle3.8.31...')
- a project-specific directory holding all your mtDNA reads for each sample in subdirectories ('Frogs'), and your reference mtGenome as a fasta file, labelled '[taxon]_mtGenome.fasta'

Here's a quick schematic of what the structure should look like:

```
/PATH_TO_PARENT_DIRECTORY/mtGenomes
|-- MITObim.pl
|-- mira_4.0.2_darwin13.1.0_x86_64_static
|   |-- bin
|   |-- etc.
|-- muscle3.8.31_i86darwin64
|-- Frogs
    |-- [taxon]_mtGenome.fasta (reference genome)
    |-- Taxon1
    |   |-- Taxon1_R1.fastq.gz
    |   |-- Taxon1_R2.fastq.gz
    |-- Taxon2
        |-- Taxon2_R1.fastq.gz
        |-- Taxon2_R2.fastq.gz
```

[1]Hahn, C., Bachmann, L., Chevreux, B. Reconstructing mitochondrial genomes directly from genomic next-generation sequencing - a baiting and iterative mapping approach. Nucleic Acids Research 41:13. doi:10.1093/nar/gkt371 (2013).

[2]Edgar, R.C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Research 32(5). doi:10.1093/nar/gkh340 (2004).

Executing the Code

Open a new R script, and source the functions we'll need

```
source("/PATH_TO_CODE/mtGenome_Assembly.R")
```

Then set your working directory to the directory holding MITObim.pl, MUSCLE, MIRA, and your project-specific directory (I've called this directory 'Frogs')

```
setwd("/PATH_TO_DIR/Frogs")
```

The first function (*mitoAssemble*) assumes that in the directory 'Frogs', there are one or more subdirectories (1 per sample), with one or more gzipped fastq files of the raw reads. For each sample (subdirectory), it will:

- merge (concatenate) any *fastq.gz files together.
- copy your reference genome (fasta file)
- call MITObim and attempt to assemble the mtGenome
- copy the assembled mtGenome to a directory of all assemblies

The function requires as input, three things:

```
mitoAssemble(num.iter, reference.name, project.name)
```

- *num.iter* is the number of assembly iterations MITObim should try before it times out and moves on to the next sample. I usually leave this set to 100, sometimes it takes 4 iterations, sometimes 60, hard to know.
- *reference.name* is the unique name of your reference genome fasta file which precedes '_mtGenome.fasta'
- *project.name* is what you'd like the generated directories to be called (where the assemblies are stored)

The function *mitoAssemble* will spit out all the assemblies to a new directory, and tell you where it is:

```
Assembly(s) completed and saved to: /Users/Ian/MITObim/Assa/Assa_mtGenomes
```

The second function *mitoAlign* will:

- combine all assembled mtGenomes into a single fasta alignment ([project.name]_Assembly_Alignment.fasta)
- align the assemblies using MUSCLE, into a single final alignment ([project.name_Aligned_Assemblies.fasta]) The function *mitoAlign* will spit out the final alignment into your new directory, and tell you where it is:

```
your alignment of mtGenome assemblies is called:
 /Users/Ian/MITObim/Assa/Assa_mtGenomes/Assa_Aligned_Assemblies.fasta
```

The function requires as input just one thing:

```
mitoAlign(project.name)
```

- *project.name* is simply what you'd like the alignment to be named

---

There's probably a bunch of other slick things I could do after this, like have it plot the individual assembly lengths, but I'm not sure it's worth it at the moment. Let me know if there's something specific you're looking for though.
Good luck!
