Chewbacca Documentation

Release 0.1

Mahdi Belcaid Greg Burgess

Mar 12, 2017

Contents

Chewbacca is a bioinformatics toolkit for COI analysis, meshing commonly used programs together to create a framework for automated analysis. Chewbacca currently supports the cleaning, assembly, demultiplexing, clustering, aligning, and Identification of COI data sequences. Chewbacca also allows users to build OTU tables and offer some basic functionality to visualize their data.

CHAPTER 1

Quick Start

1. Grab the Docker image version x.x.x.
2. In your docker shell:

```
$ docker load -it chewbacca_vx.x.x
$ docker run -it chewbacca_vx.x.x /bin/bash
# cd ~/ARMS/testARMS
# python ~/ARMS/src/ARMS/chewbacca.py --help
```

A complete example is available in:

```
$~/ARMS/test/commands.sh
```

This file describes the typical analysis steps.

CHAPTER 2

FAQ

What is Chewbacca?

Chewbacca is a command line bioinformatics toolkit for COI analysis, meshing commonly used programs together to create a framework for automated analysis. Chewbacca currently supports the cleaning, assembly, demultiplexing, clustering, Aligning, and Identification of COI data. Chewbacca also allows users to build OTU tables, as well as basic functionality to visualize their data. Chewbacca is under active development.

What does it do?

Chewbacca is wrapper for a number functions required for handling COI data (see available commands). It is conceptually similar to mothur and Qiime, but focuses on modularity and parallelism.

Who should use Chewbacca?

Anyone who need to analyze COI data (at any stage of processing) for abundance/distribution questions.

I have some fasta files. I need to clean them. Can you help?

Yes! Chewbacca comes with a 'default' set of steps that will take in raw reads (or reads at varying levels of assembly/cleaning) and give you back an OTU table and some nice graphs.

What makes Chewbacca Different?

1. Chewbacca is a toolkit designed with run_parallel processing in mind. Chewbacca's operations are as run_parallel as possible.

2. Chewbacca is modular. Chewbacca's subprograms are each designed to tackle one small problem. Odds are good that you'll find some parts of the toolkit useful.

3. Chewbacca remembers. Did you mess up in one of your steps? Chewbacca saves the output of each step to a directory, meaning you don't have to start from scratch if you change part of your pipeline.

4. With the constant addition of tools to hadle short read data, chewbacca is easily extensible. In fact, most commands are preconfigured with multiple options. For instance, clustering can be carried out using **'crop https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3042185/'_**, **'cd-hit http://weizhongli-lab.org/cd-hit/'_**, v-search or **'SWARM http://weizhongli-lab.org/cd-hit/'_**.

Don't like what you see?

Adding new processes/programs to Chewbacca is easy! Take a look at our dev_guide. Alternatively, please contact us for any functionality suggestions.

CHAPTER 3

Reference

- file_types
- commands
- API
- dev_guide