

BioType: A crash course in bioinformatics and custom pipelines

Ethan Gniot

May 2018

Todo list

Talk about the biology concepts they should be familiar with before starting.	5
Still need to add information about the inflammatory bowel disease dataset	7
Make URLs footnotes instead of appendix entries	7
Fix bibliography entries in this paragraph and in general. They're not correctly referencing even though bib file has entries.	7
(Include information about the untested inflammatory bowel disease dataset that we will analyze using the completed pipeline)	7
Add further software that you end up using (e.g., USEARCH). ALSO, make sure to update the section that mentions installing packages if you do so.	12
add 3rd column to table with link to documentation/download source	12
Write a blurb explaining the benefits of using a virtual environment. .	13
Link to resource for further reading on virtual environments	13
Write blurb about what packages are.	14
Link to resource for further reading about packages.	14
Solve the issue with failing to center figures when they're on their own page	15
Talk about setting up the sra-tools workspace (https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=std)	15
PYPIPER IS ONLY COMPATIBLE WITH MAC AND LINUX. Start by coding with just subprocess module commands. If the scripts work on Windows computers, then forget about using pypiper. But if the subprocess scripts don't work on windows, then we'll be developing exclusively for Mac anyways, so you could use pypiper without any worries. In that case, talk about installing pypiper here. OTHERWISE, delete this section.	15
show image of what the prompt looks like when it's done initializing. .	15
Reference the figure pypiper-already-installed	17
Figure: pypiper-already-installed	17
reference figure pypiper-wrong-location	17
Figure: pypiper-wrong-location	17
Talk about choosing between PyCharm and other options	18
(Talk about putting all the tools in the same path/directory)	18

Contents

1	Foreword	4
1.1	Goal	4
1.2	How to Use this Manual	4
1.3	Pre-requisites	5
2	Source Code	6
3	Sample Information	7
3.1	Test Dataset: Relative Abundance Analysis	7
4	Setting the scene	8
5	Microbiome Analysis	9
5.1	The Gut Microbiome	9
5.2	Relative Abundance Analysis	9
5.3	Metagenomics	9
5.4	Python	9
6	How to Find Tools	10
6.1	Finding Data	10
6.2	Finding Software	10
7	Plan: 8 Main Sections	11
8	Software and Set-up	12
8.1	Software	12
8.2	Set-up and Install Dependencies	12
8.2.1	Install Anaconda	12
8.2.2	Create a New Virtual Environment	13
8.2.3	Install packages	14
8.2.4	Integrated Development Environment (IDE)	18
8.2.5	PATH	18
8.3	Analysis Pipeline	18

9 The Dataset	19
9.1 Find Dataset	19
9.2 Download Dataset	19
9.2.1 BioPype Workflow	19
9.2.2 Creating the code	19
9.3 Perform Quality Control on Dataset	19
10 Section 3: Relative Abundance Analysis	20
11 Section 4: Predict ORFs	21
12 Section 5: Create Non-redundant Gene Sets	22
13 Section 6: Align Genes	23
14 Section 7: Get GenBank Accession Numbers	24
15 Section 8: Find COG Functional Classes	25
16 New Analysis	26
17 TODO	27
A Web Links	28
B Referenced Studies	29

1 Foreword

1.1 Goal

This tutorial aims to improve your general understanding of bioinformatics through several methods:

- Define technical terms commonly used in bioinformatics methods and found in the literature.
- Provide a collection of various useful resources, including...
 1. Resources for finding tools, data, and background information that can help answer your research questions.
 2. Resources that explain details about bioinformatics concepts and techniques in beginner-friendly language.
- Demonstrate how Python can be used to answer your research questions by combining existing bioinformatics tools and automating repetitive or time-consuming tasks.

1.2 How to Use this Manual

The main text of this book is written in the right-hand margin, while the left-hand margin contains special markers and important notes to the reader. All of the resources referenced in the following chapters can be found in the left-hand margin, the appendices, or both. The book can be used as a self-paced tutorial with the help of the markers described below.

This is a margin label. I will write things here to further explain the main text, define jargon, etc.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas eu felis sodales, interdum purus nec, interdum ex. Integer at nunc ultricies, tempus nibh eget, egestas risus.

! → This is an “Attention” marker. It will indicate key pieces of information that you should pay special attention to. *Curabitur egestas aliquam nisl, pharetra finibus mauris placerat nec.*

→ This kind of annotation will reference appendix entries that you can consult for more-detailed information about the main text.

Ut ultrices eros velit, at faucibus ante rutrum eget. Pellentesque a molestie diam. Curabitur mattis dui a risus lacinia fringilla. Phasellus porttitor elit nec neque euismod, id ultrices elit lobortis.

Chapters 2 - 4 cover supplementary information about this project. Chapter 5 and Chapter 6 are educational in nature. Chapter 5 gives background information about microbiome analyses and references helpful resources for learning why these analyses are useful. Chapter 6 talks about various

databases for finding public data, along with places where people can look for existing software programs that could help them answer their research questions. Chapter 7 breaks down the steps required to complete the microbiome analyses in this tutorial.

Finally, Chapters 8 - 16 are about creating and using the BioPype pipeline. These chapters are hyper-specific to the BioPype commands, rather than bioinformatics in general, though they still provide resources for reading more about the topics that are discussed. Each chapter covers a different step of the microbiome analysis process, and each chapter is divided into two sections: a "How to Build" section and a "How to Use" section.

The **How to Build** section teaches you how to build your own pipeline to answer your research questions. This is accomplished by explaining *how* and *why* the BioPype commands were created the way they were. By seeing the thought process behind BioPype's development process, you can adapt the process to suit your own development needs.

The **How to Use** section explains how to use BioPype's functionality to answer questions about the gut microbiome. The section provides a workflow that explains which commands should be used in which order to complete the step that the chapter focuses on.

1.3 Pre-requisites

Things to know

Talk about the biology concepts they should be familiar with before starting.

Computer requirements

- At least 4GB RAM bare minimum, 16GB RAM if possible (basically, the more RAM, the better)
- Must be able to run macOS 10.12 Sierra, preferably macOS 10.13 High Sierra
- At least 500GB storage (ideally several TB)

2 Source Code

(Source code can be found at github.com/EthanGniot/BioPype)

3 Sample Information

Still need to add information about the inflammatory bowel disease dataset

While we are building the BioType pipeline, we will need a dataset that we can use to test our pipeline throughout the process and make sure it is working as intended. In order to do this, we must use a dataset that's already been analyzed so that we know what the results should look like. When we test our pipeline, if our results match those of the original analysis, then we will know that our tool is working correctly.

3.1 Test Dataset: Relative Abundance Analysis

There are several public datasets that can be used to test our code while we develop the microbial relative abundance analysis.

Make URLs footnotes instead of appendix entries

→ Caporaso et al, 2011 [?]

The first is the dataset used in both the QIIME “Illumina Overview Tutorial” (A.2) and the QIIME 2 “Moving Pictures” tutorial (A.3) derived from the *Moving Pictures of the Human Microbiome* study, where two human subjects collected daily samples from four body sites: the tongue, the palm of the left hand, the palm of the right hand, and the gut (via fecal samples obtained by swapping used toilet paper). These data were sequenced using the barcoded amplicon sequencing protocol described in *Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample*.

→ THIS CITATION NEEDS TO BE FIXED [?]

Fix bibliography entries in this paragraph and in general. They're not correctly referencing even though bib file has entries.

(Include information about the untested inflammatory bowel disease dataset that we will analyze using the completed pipeline)

4 Setting the scene

("Here" is a hypothetical situation/research question that a student may have. This is the research question that will be answered by the pipeline we are creating.)

5 Microbiome Analysis

(This chapter will give some general background information about the topics listed below. More-detailed information will be provided in later chapters when we are actually creating the pipeline)

5.1 The Gut Microbiome

5.2 Relative Abundance Analysis

5.3 Metagenomics

5.4 Python

6 How to Find Tools

6.1 Finding Data

(Here is where we talk about various databases that users can use to find general information, data files, study results, public datasets, etc.)

6.2 Finding Software

(Here is where we talk about ways/places that people can look for software programs that can help answer their research question.)

7 Plan: 8 Main Sections

(Break down the sub-tasks required to accomplish the two main tasks: Relative abundance analysis and metagenomic analysis)

1. What is my research question?
2. Is there an existing tool that I can use to directly answer my research question? If not, proceed to Step 3.
3. What is the step-by-step process required to answer my research question?
4. What existing tools are available that can help me accomplish each of these steps?
5. How do I write code that uses these tools to accomplish the steps?

8 Software and Set-up

8.1 Software

Add further software that you end up using (e.g., USEARCH). ALSO, make sure to update the section that mentions installing packages if you do so.

(Table of software name, name in PATH, version number, function of the software for each one we're gonna use)

add 3rd column to table with link to documentation/download source

<i>Software name</i>	<i>Version number</i>
Anaconda	5.1
Biopython	1.70
BLAT	35
matplotlib	2.2.2
pandas	0.22.0
sra-tools	2.8.2
trim-galore	0.4.5

Table 8.1: **Software used to create the tutorial pipeline.**

8.2 Set-up and Install Dependencies

Before we write any code, there are several steps that must be completed to prep your machine for the tasks we will be performing in this tutorial. Without these prerequisites, the code you write during this tutorial will not work correctly:

1. Install and open Anaconda
2. Create a new virtual environment
3. Install packages

8.2.1 Install Anaconda

→ RESOURCE FOR LEARNING ABOUT PYTHON PACKAGES

The Anaconda program will play a key role in this tutorial. Anaconda is essentially Python and a lot of scientific computing tools bundled together, along with many popular add-ons to Python called packages. Downloading all of these tools individually can be difficult, as the quirks of one package may conflict with another when they're installed manually; using Anaconda to install packages greatly simplifies the process because Anaconda can smoothly handle all of the minute details that cause manual installations to fail.

Install and Open:

1. Go to the download page for the Anaconda distribution at <https://www.anaconda.com/download>.

2. Select your preferred operating system from the Windows, macOS, or Linux tabs, then select the Download option for the **Python 3.6 version** (Figure 8.1) and follow the installation instructions.

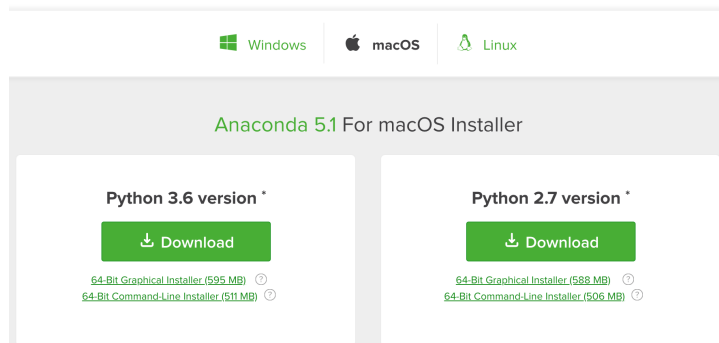



Figure 8.1: The Anaconda download options provided on the Anaconda distribution website at <https://www.anaconda.com/download>

3. After installation is complete, open the application named “Anaconda-Navigator” (the icon looks like ) . After a brief start-up period, you should see the following window (Figure 8.2):

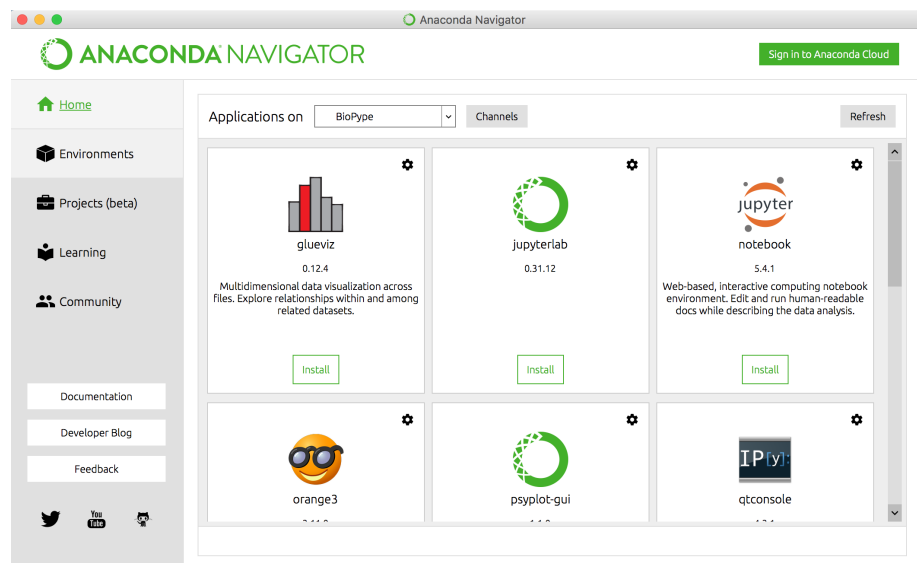


Figure 8.2: The window displayed to the user upon opening Anaconda-Navigator.

8.2.2 Create a New Virtual Environment

Write a blurb explaining the benefits of using a virtual environment.

Link to resource for further reading on virtual environments

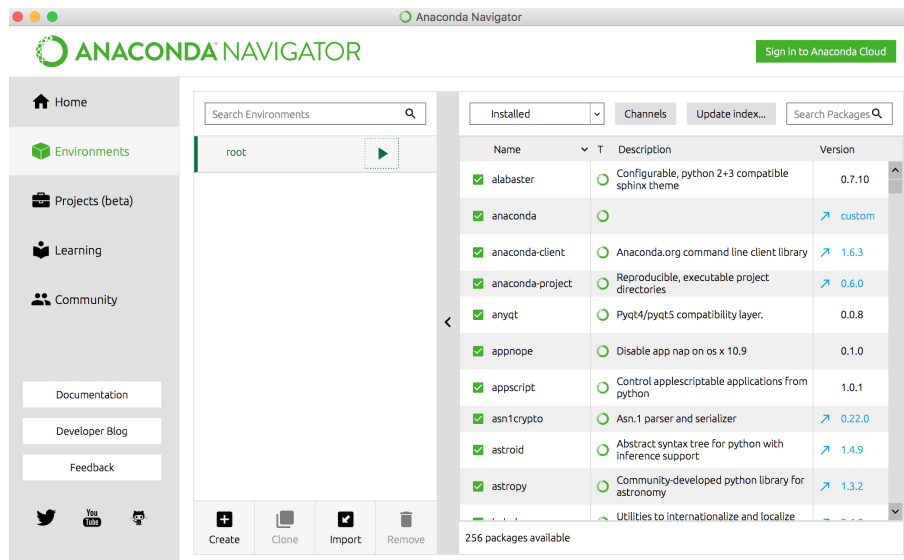


Figure 8.3: The Environments window of the Anaconda-Navigator.

Make sure the computer has an internet connection while completing this section, otherwise Anaconda will not let you create a virtual environment.

1. On the left side of the Anaconda-Navigator window, click on the tab labeled **Environments**. (Figure 8.3)
2. Click the **Create** button on the bottom of the center panel. A new window titled “Create new environment” will appear. (Figure 8.4)
3. Enter a **Name** for the environment. You may choose any name you want, but for the sake of this tutorial we will name the new environment “BioPype”.
4. Select the box labeled **Python** next to the **Packages** heading.
5. Choose the latest version of Python from the adjacent drop-down menu (Python 3.6 is the most current version at the time of this writing, so we choose **3.6**).
6. Click the **Create** button within the “Create new environment window”.

8.2.3 Install packages

Write blurb about what packages are.

Link to resource for further reading about packages.

1. Change Anaconda’s current environment from the **root** environment by selecting the **BioPype** tab in the middle panel of the Environments window.
2. Click on the drop-down menu in the right-hand panel that says “Installed” and change it to “All”.

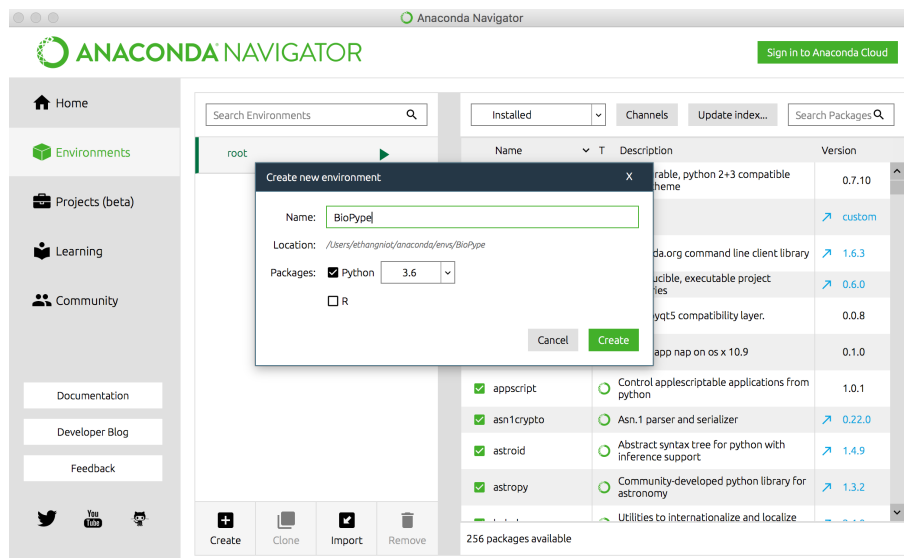


Figure 8.4: The “Create new environment” window.

3. In the “Search Packages” box, enter “biopython”. The search should return a package named “biopython”. Select the checkbox to the left of the name. (Figure 8.5)

- A pair of green and red boxes (reading “Apply” and “Clear”, respectively) will appear in the bottom-right of the window once the package is selected. Do not click these just yet.

Solve the issue with failing to center figures when they're on their own page

4. Use the search bar to find and select the other packages listed in Table 8.1. Once all packages have been selected, click the green “Apply” button in the bottom right corner of the window, then select “Apply” again within the “Install Packages” window that appears. (Figure 8.6) Anaconda will now install the selected packages.

5. [Talk about setting up the sra-tools workspace \(https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_dock&f=std\)](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_dock&f=std)

6. **PYPYPIPER IS ONLY COMPATIBLE WITH MAC AND LINUX.** Start by coding with just subprocess module commands. If the scripts work on Windows computers, then forget about using pypiper. But if the subprocess scripts don't work on windows, then we'll be developing exclusively for Mac anyways, so you could use pypiper without any worries. In that case, talk about installing pypiper here. OTHERWISE, delete this section.

- (a) Open a terminal window in the BioPype environment by clicking the “play” button on the BioPype environment tab and then selecting “Open Terminal”.

- (b) Wait for the terminal window to finish opening. You'll know it's finished when you see

show image of what the prompt looks like when it's done initializing.

- (c) Install **pypiper** by typing the following at the command prompt, followed by pressing return/enter:

```
pip install —user pypiper
```

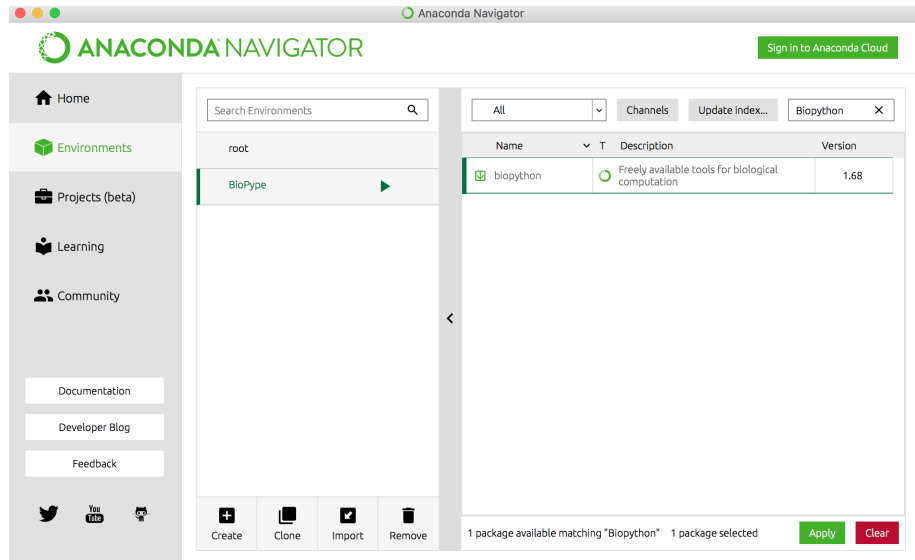



Figure 8.5: Searching for a package. When a package is selected, the check-box next to the package’s name will be green.

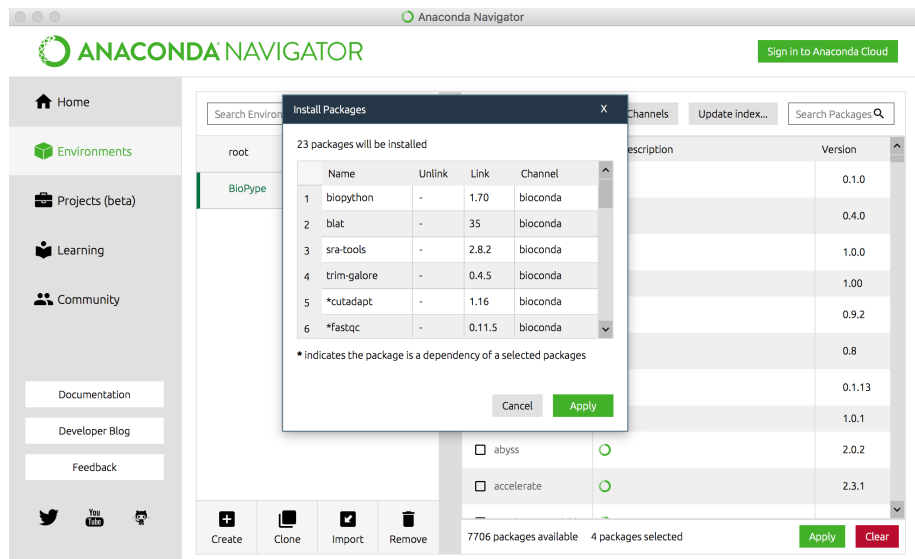


Figure 8.6: The window displaying the packages and dependencies that will be installed.

- (d) Check that the package was installed correctly by executing the following in the command prompt:

```
conda list
```

This will generate a list of all the packages installed in the current environment. If you see the **pypiper** package listed, the installation was successful and you may skip the rest of this section. If not, proceed with the following steps.

- (e) Execute the install command from step (c) again. This time, the Terminal should return a message similar to the one displayed in

Reference the figure `pypiper-already-installed`

. The line that reads “Requirement already satisfied: pypiper in” tells us the location where the package was (incorrectly) installed.



- (f) Open a Terminal window, and navigate to the location indicated by the message from the previous step. For my example, I need to start at my home directory and walk through the following folders: `.local` — `lib` — `python3.6` — `site-packages`.

- The folders along the path to the pypiper installation may be hidden. On a Mac, these hidden folders are preceded by a “.” If the path to the pypiper installation includes hidden locations, reveal them by pressing “Cmd + Shift + .” in the Finder window.
- Once you find the site-packages folder containing two pypiper folders

reference figure `pypiper-wrong-location`

, copy those folders and their contents and paste them into the `/anaconda/envs/BioPype/lib/python3.6/site-packages` directory. The package should now be installed correctly.



8.2.4 Integrated Development Environment (IDE)

Talk about choosing between PyCharm and other options

8.2.5 PATH

(Talk about putting all the tools in the same path/directory)

8.3 Analysis Pipeline

(Use figures to illustrate the stages of the pipeline)

9 The Dataset

In the previous chapter, we set up our machine so that it has all of the software BioPyne needs in order to function.

In this chapter, we will use BioPyne to download experimental data, and then prepare them for analysis via a process called “quality control”.

9.1 The Data

What experiment are the data from? What was the study investigating? Where did we find the data?

(Find dataset)

(Create script to automate download of data)

(Quality control protocols and how to make script for automated QC)

9.2 Find Dataset

1. NCBI
2. SRA database
3. Look for WGS data

9.3 Download Dataset

1. Download RunInfo Table
2. Create RunTable object from RunInfo Table file
3. Use filtering methods of RunTable object to select data

9.3.1 BioPyne Workflow

9.3.2 Creating the code

9.4 Perform Quality Control on Dataset

10 Section 3: Relative Abundance Analysis

(How to compare relative abundance of bacterial taxa between experimental conditions using QWRAP.)

11 Section 4: Predict ORFs

(how to predict the ORFs of the sequencing reads)

12 Section 5: Create Non-redundant Gene Sets

(How to create non-redundant gene sets using the predicted ORFs and what kind of information they provide) (Align reads using BLAT)

13 Section 6: Align Genes

14 Section 7: Get GenBank Accession Numbers

15 Section 8: Find COG Functional Classes

16 New Analysis

(Walk user through analysis of new, untested dataset looking for age-related differences in patients with Inflammatory Bowel Disease.)

17 TODO

- Talk about the IBD dataset in Chapter 3: Sample Information
- Fix the bibliography citations. Some work, some don't, and I'm not sure why.
- Put URLs in footnotes instead of appendix entries?

Appendix A Web Links

A.1 this is a test section

A.2 http://nbviewer.jupyter.org/github/biocore/qiime/blob/1.9.1/examples/ipynb/illumina_overview_tutorial.ipynb

A.3 <https://docs.qiime2.org/2018.2/tutorials/moving-pictures/>

Link 3

Appendix B Referenced Studies
