Workshop Format

We’ll be going through these topics today:

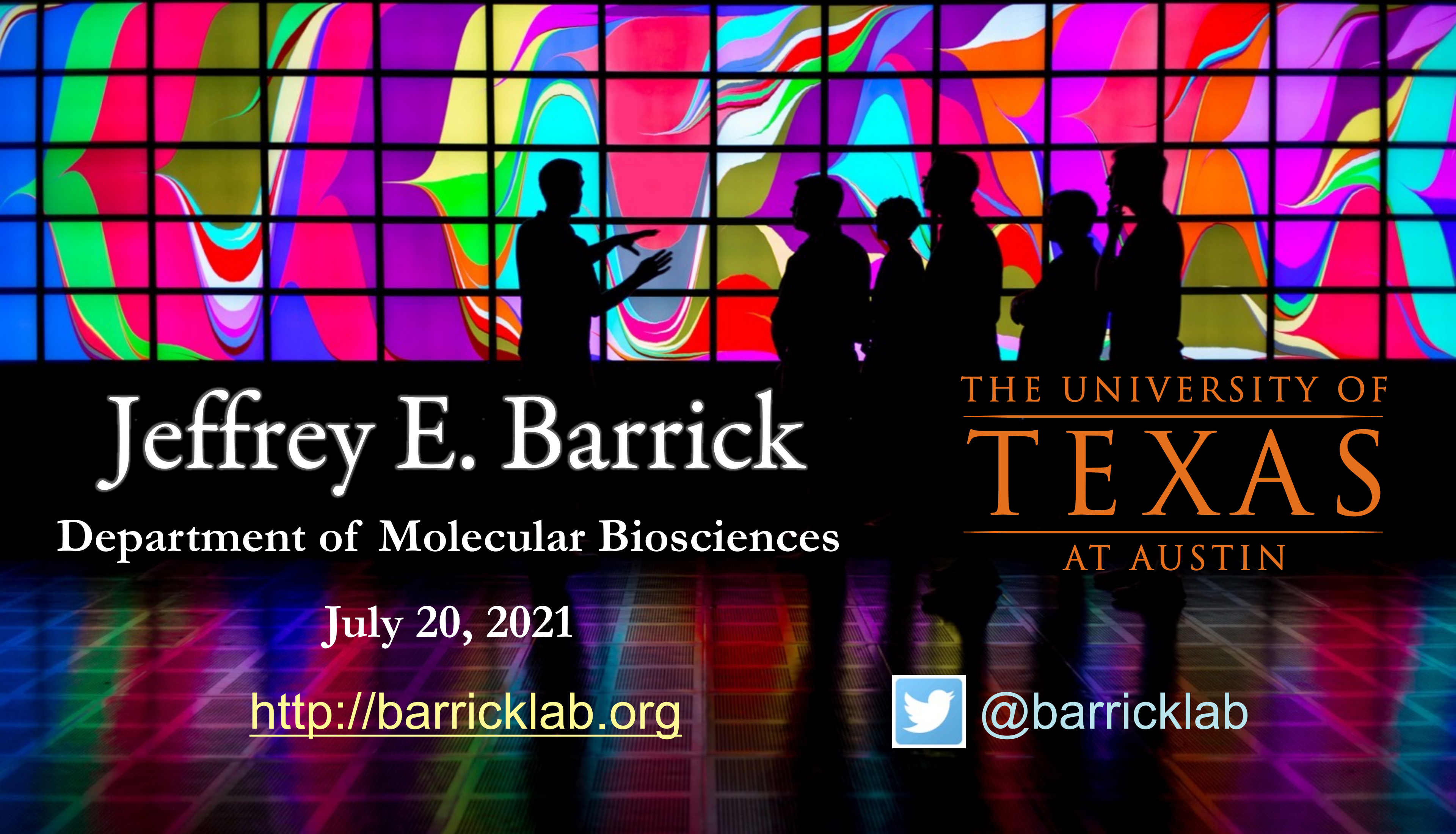
- Installation
- Basic usage: common options/modes
- Interpreting HTML output

Feel free to jump in with questions or ask them the chat!

I recommend you open a browser to this page:
https://barricklab.org/breseq

You can click through the same results I will be viewing there.

breseq :: Introductory Topics



Jeffrey E. Barrick

Department of Molecular Biosciences

THE UNIVERSITY OF TEXAS AT AUSTIN

July 20, 2021

http://barricklab.org

@barricklab

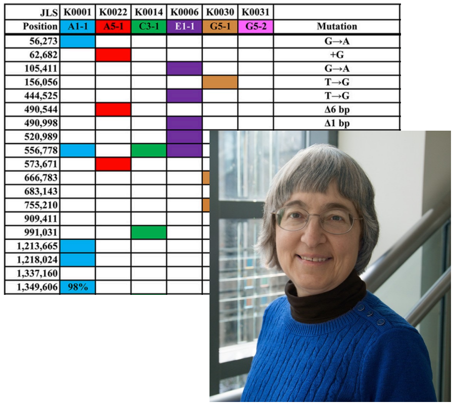
When is *breseq* the right tool?

Deatherage, D. E., Barrick, J. E. (2014) **Identification of mutations in laboratory-evolved microbes from next-generation sequencing data using *breseq*.** *Methods Mol. Biol.* **1151**: 165–188. https://doi.org10.1007/978-1-4939-0554-6_12

https://barricklab.org/breseq https://github.com/barricklab/breseq

- You have short-read NGS resequencing data.
- Your reference genome is ***haploid.***
  - Bacteria, Archaea, Phages, Plasmids, Haploid yeast
- You expect few genetic differences from the reference (a few to <1,000) in each sample.
- It's important that you identify all mutations.
- You are comfortable with using the terminal a little.
  - Changing directories, copying files, running a command

Workshop Presentations

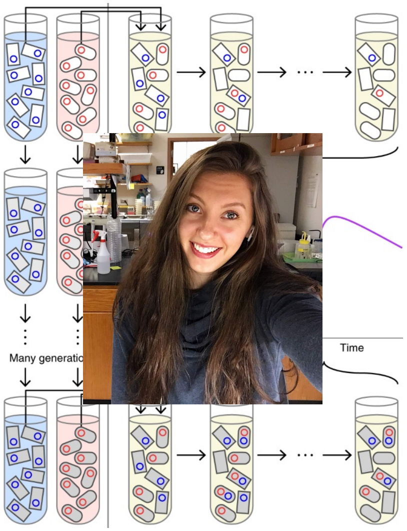


JLS	K0001	K0022	K0014	K0006	K0030	K0031	
Position	A1-1	A5-1	C3-1	E1-1	G5-1	G5-2	Mutation
56,273	🟦						G→A
62,682		🟥					+G
105,411				🟪			G→A
156,056					🟧		T→G
444,525				🟪			T→G
490,544		🟥					Δ6 bp
490,998				🟪			Δ1 bp
520,989				🟪	[illegible]	[illegible]	[illegible]
556,778	🟦		🟩	🟪	[illegible]	[illegible]	[illegible]
573,671		🟥			[illegible]	[illegible]	[illegible]
666,783					🟧	[illegible]	[illegible]
683,143					[illegible]	[illegible]	[illegible]
755,210					🟧	[illegible]	[illegible]
909,411					[illegible]	[illegible]	[illegible]
991,031			🟩		[illegible]	[illegible]	[illegible]
1,213,665	🟦				[illegible]	[illegible]	[illegible]
1,218,024	🟦				[illegible]	[illegible]	[illegible]
1,337,160					[illegible]	[illegible]	[illegible]
1,349,606	98%				[illegible]	[illegible]	[illegible]

Antibiotic Resistance Reversal: *breseq* Analysis of Experimental Evolution, Compared with FACS Competition Assays of Relative Fitness

Joan Slonczewski

Kenyon College



Identifying Adaptive Paths in Host-Plasmid Coevolution Using *breseq*

Olivia Kosterlitz

University of Washington

Workshop Presentations

EVOLVINGSTEM

Decoding Evolution-In-Action in Classroom Experiments That Simulate Infection Biology Using *breseq*

Vaughn Cooper

University of Pittsburgh

ALEdb

ALEdb: A Living High-Quality Database of Mutations from Adaptive Evolution Experiments Powered by *breseq*

Adam Feist

University of California, San Diego

Introductory Topics

- Installation
  - Different methods
  - Common problems
- Basic usage
  - Most important options/modes
  - Different types of reference files
- Interpreting HTML output
  - Compare output generated using different options
  - Understanding and evaluating predictions

breseq

Installing *breseq*

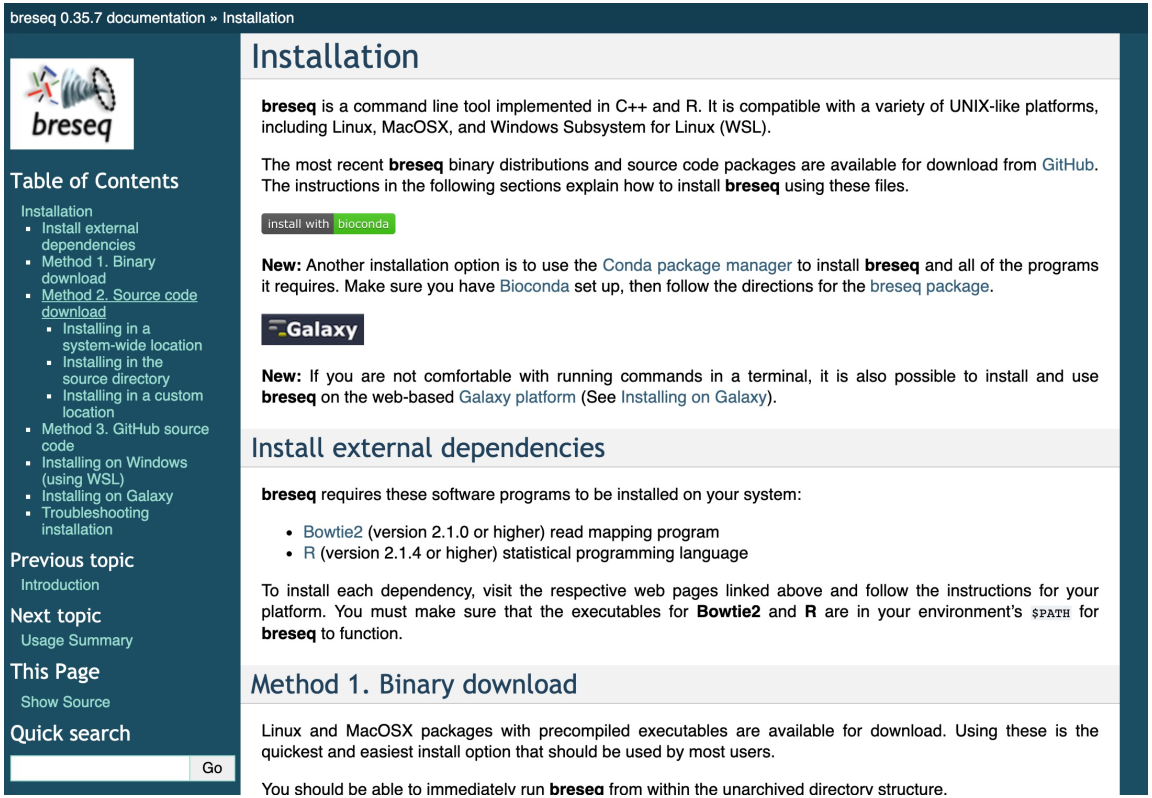
Like many bioinformatics tools, *breseq* requires a Unix-like environment with a command line.

Linux and MacOS

- Open the terminal

Windows

- Install **WSL** (Windows Subsystem for Linux). Then you should be able to follow the Linux instructions!
- Also possible to use **Cygwin** (but not as straightforward)



breseq 0.35.7 documentation » Installation

breseq

Table of Contents

Previous topic

Introduction

Next topic

Usage Summary

This Page

Show Source

Quick search

Go

Installation

breseq is a command line tool implemented in C++ and R. It is compatible with a variety of UNIX-like platforms, including Linux, MacOSX, and Windows Subsystem for Linux (WSL).

The most recent **breseq** binary distributions and source code packages are available for download from GitHub. The instructions in the following sections explain how to install **breseq** using these files.

install with bioconda

New: Another installation option is to use the Conda package manager to install **breseq** and all of the programs it requires. Make sure you have Bioconda set up, then follow the directions for the breseq package.

Galaxy

New: If you are not comfortable with running commands in a terminal, it is also possible to install and use **breseq** on the web-based Galaxy platform (See Installing on Galaxy).

Install external dependencies

breseq requires these software programs to be installed on your system:

- Bowtie2 (version 2.1.0 or higher) read mapping program
- R (version 2.1.4 or higher) statistical programming language

To install each dependency, visit the respective web pages linked above and follow the instructions for your platform. You must make sure that the executables for **Bowtie2** and **R** are in your environment's `$PATH` for **breseq** to function.

Method 1. Binary download

Linux and MacOSX packages with precompiled executables are available for download. Using these is the quickest and easiest install option that should be used by most users.

You should be able to immediately run **breseq** from within the unarchived directory structure.

Web-Base Galaxy Platform

- Available from the Galaxy Toolshed

Installing *breseq*

CONDA

https://docs.conda.io/en/latest/miniconda.html

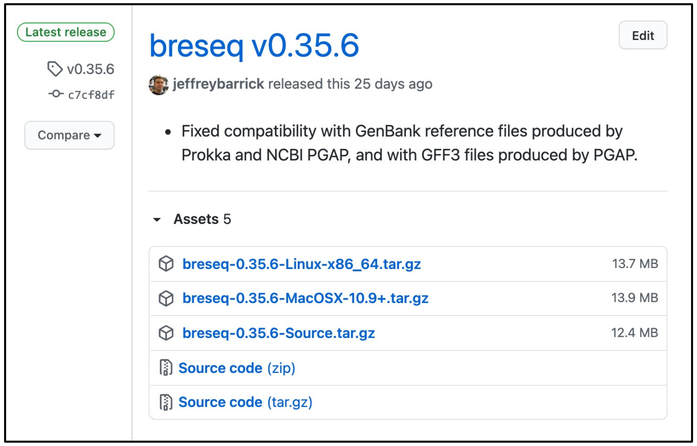
BIOCONDA®

http://bioconda.github.io/index.html

- Easiest way
  - Install **miniconda** then

```
$ conda -c bioconda breseq
```

- Harder way
  - Download binary for Linux or MacOS X
  - You must also install **bowtie2** and **R**
  - You need to set your **`$PATH`**



Latest release

v0.35.6

c7cf8df

breseq v0.35.6

jeffreybarrick released this 25 days ago

- Fixed compatibility with GenBank reference files produced by Prokka and NCBI PGAP, and with GFF3 files produced by PGAP.

▾ **Assets** 5

breseq-0.35.6-Linux-x86_64.tar.gz	13.7 MB
breseq-0.35.6-MacOSX-10.9+.tar.gz	13.9 MB
breseq-0.35.6-Source.tar.gz	12.4 MB
Source code (zip)	
Source code (tar.gz)	

- Hardest way
  - Download source code or clone GitHub repository
  - Follow the instructions in the DEVELOPER text file
  - Requires you to have a C++ compiler, dev version of libz with headers, autoconf, automake, etc., installed.

Most Common Install Problems

- You get a message like this, or *breseq* has an error because it can’t find an installed bowtie2 or R:

```
zsh: command not found: breseq
```

  - One of these commands is not in your **`$PATH`**
  - Great explanation if you don’t understand: https://astrobiomike.github.io/unix/modifying_your_path

- You are on a computer cluster and get errors when generating output plots or empty plots
  - Installed R does not have graphics capability
  - Try installing your own version (using miniconda, for example) or ask the system administrator.

Crafting your *breseq* command

Basic *breseq* command

```
$ breseq -r reference.gbk reads_1.fastq reads_2.fastq
```

References (`-r`) can be in GenBank, GFF3, or FASTA format.

Multiple read files can be used. Paired/unpaired are treated the same.

Multiple reference files can be used:

```
-r genome.fasta -r plasmid.gff3
```

Read files can be gzipped: `reads_1.fastq.gz`

Speed up execution by using multiple threads: `-j 8`

View common options

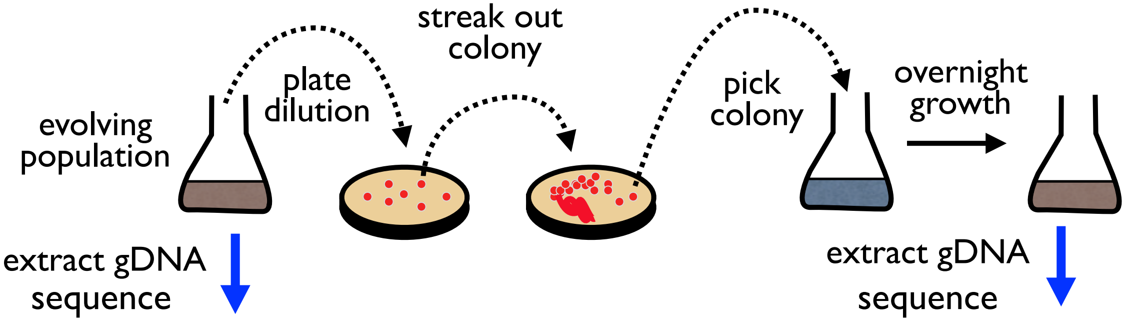
```
$ breseq
```

View all options

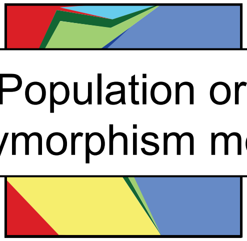
```
$ breseq -h  or $ breseq --help
```

Analysis modes

There are three overall modes for running *breseq*…



Every read could be from any individual.

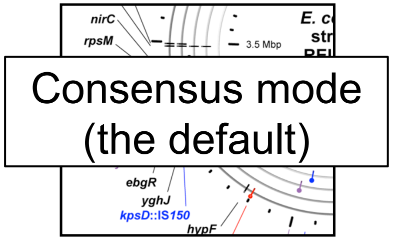


Population or Polymorphism mode

```
$ breseq -p
```

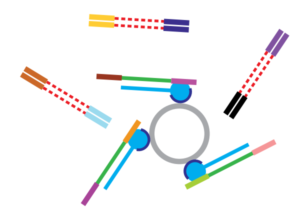
breseq

All reads are from a single clone.



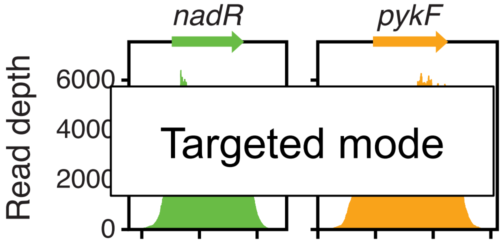
Consensus mode (the default)

```
$ breseq ...
```



PCR amplicons or pull-down enrichment

Reads are from only a subset of the genome



Targeted mode

```
$ breseq -t
```

Important! Each mode has different assumptions/options.

Reference file considerations

- **Microbes (<20Mb)**: download GenBank or GFF3 files with both DNA sequence and features.
- **Important:** having transposable elements annotated leads to better predictions!
- **What do I do if there is no reference?**
  - *de novo* assemble and annotate your own
  - **Recommendation:** Unicycler PROKKA
  - You may need to iteratively improve the assembly and annotation to get the best results. You could use **`gdtools APPLY`** (see advanced workshop).

Specifying reference sequences

You can have three types of references:

1. **Normal (-r, --reference)**
   - Call all kinds of mutations. Each sequence is a different episome.
2. **Contig (-c, --contig-reference)**
   - This is a de novo assembly. Treat all of the sequences in this file as if they are from the same episome (e.g., one chromosome)
   - This improves calling deletions by uniformly assigning a read-depth of coverage across
3. **Junction-only (-s, --junction-only-reference)**
   - I am searching for where part of this sequence was inserted into my genome. Don't want breseq call mutations in this sequence.
   - Example: integration cassette / transposon

Read file considerations

Sequencing technology

- Can work with any FASTQ
- Best results with short-read data (< 1000 bases)
- Not appropriate for **long-read** data (Nanopore, PacBio, etc.) In this case, you should *de novo* assemble and then compare assemblies.

Recommended depth of coverage

>40x for clonal samples

>120x for population samples

More coverage is unlikely to give improvements without error correction (ex: molecular barcodes).

Adaptor and Barcode Removal

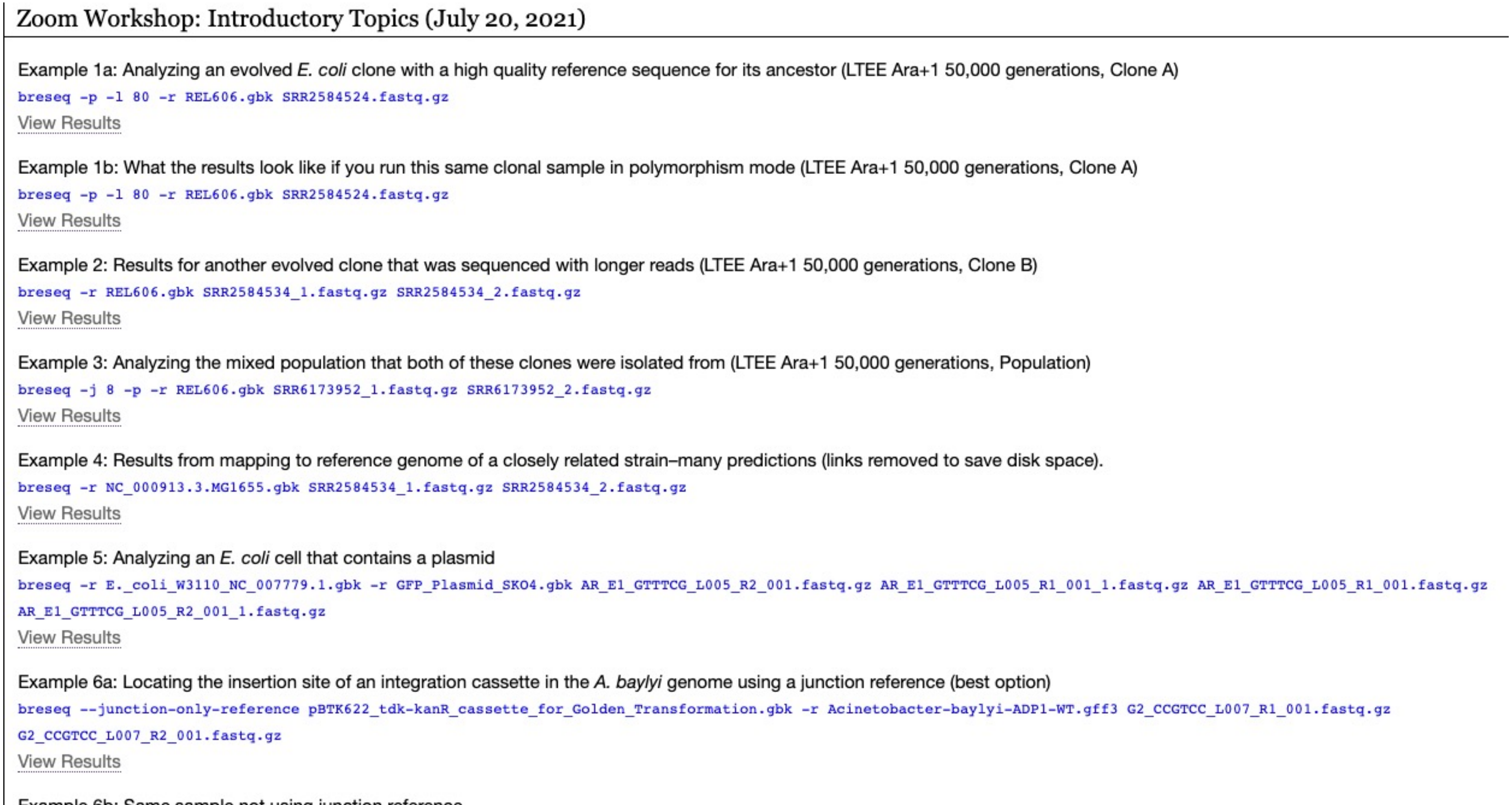
You must trim your reads to remove these!

Use `fastp`, `trimmomatic`, etc. You can evaluate reads with `fastqc`.

If you don’t clean this up, then they may result in reads not mapping (90% of length must be covered by the read alignment by default).

Example *breseq* input/output

Let's look at some results! https://barricklab.org/breseq



Zoom Workshop: Introductory Topics (July 20, 2021)

Example 1a: Analyzing an evolved *E. coli* clone with a high quality reference sequence for its ancestor (LTEE Ara+1 50,000 generations, Clone A)

```
breseq -p -l 80 -r REL606.gbk SRR2584524.fastq.gz
```

View Results

Example 1b: What the results look like if you run this same clonal sample in polymorphism mode (LTEE Ara+1 50,000 generations, Clone A)

```
breseq -p -l 80 -r REL606.gbk SRR2584524.fastq.gz
```

View Results

Example 2: Results for another evolved clone that was sequenced with longer reads (LTEE Ara+1 50,000 generations, Clone B)

```
breseq -r REL606.gbk SRR2584534_1.fastq.gz SRR2584534_2.fastq.gz
```

View Results

Example 3: Analyzing the mixed population that both of these clones were isolated from (LTEE Ara+1 50,000 generations, Population)

```
breseq -j 8 -p -r REL606.gbk SRR6173952_1.fastq.gz SRR6173952_2.fastq.gz
```

View Results

Example 4: Results from mapping to reference genome of a closely related strain–many predictions (links removed to save disk space).

```
breseq -r NC_000913.3.MG1655.gbk SRR2584534_1.fastq.gz SRR2584534_2.fastq.gz
```

View Results

Example 5: Analyzing an *E. coli* cell that contains a plasmid

```
breseq -r E._coli_W3110_NC_007779.1.gbk -r GFP_Plasmid_SKO4.gbk AR_E1_GTTTCG_L005_R2_001.fastq.gz AR_E1_GTTTCG_L005_R1_001_1.fastq.gz AR_E1_GTTTCG_L005_R1_001.fastq.gz
AR_E1_GTTTCG_L005_R2_001_1.fastq.gz
```

View Results

Example 6a: Locating the insertion site of an integration cassette in the *A. baylyi* genome using a junction reference (best option)

```
breseq --junction-only-reference pBTK622_tdk-kanR_cassette_for_Golden_Transformation.gbk -r Acinetobacter-baylyi-ADP1-WT.gff3 G2_CCGTCC_L007_R1_001.fastq.gz
G2_CCGTCC_L007_R2_001.fastq.gz
```

View Results

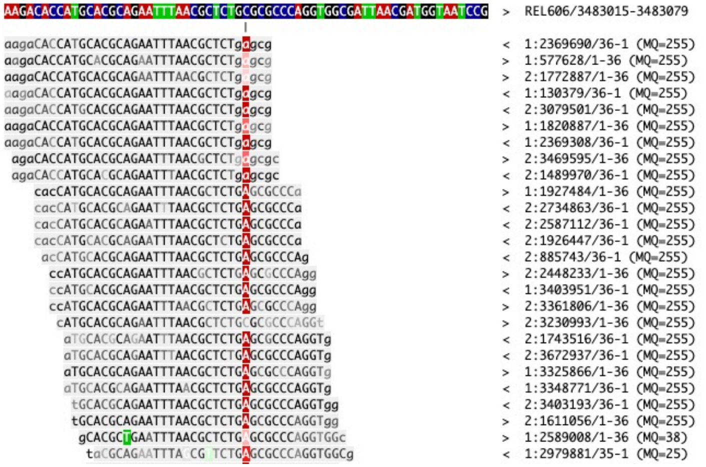
Example 6b: Same sample not using junction reference

Utilities to explore output

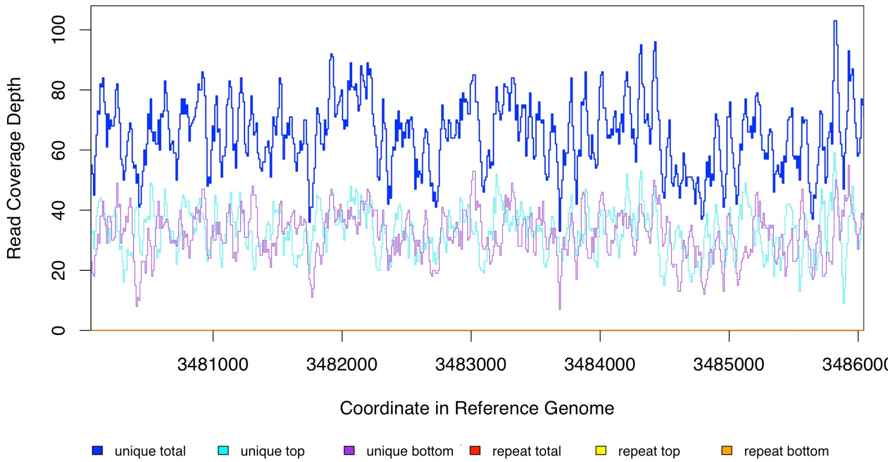
You can run utility subcommands from inside the main output directory of a *breseq* run. `$ breseq --help` to see others.

```
$ breseq BAM2ALN
  -o alignment.html
  REL606:3483047-3483047
```

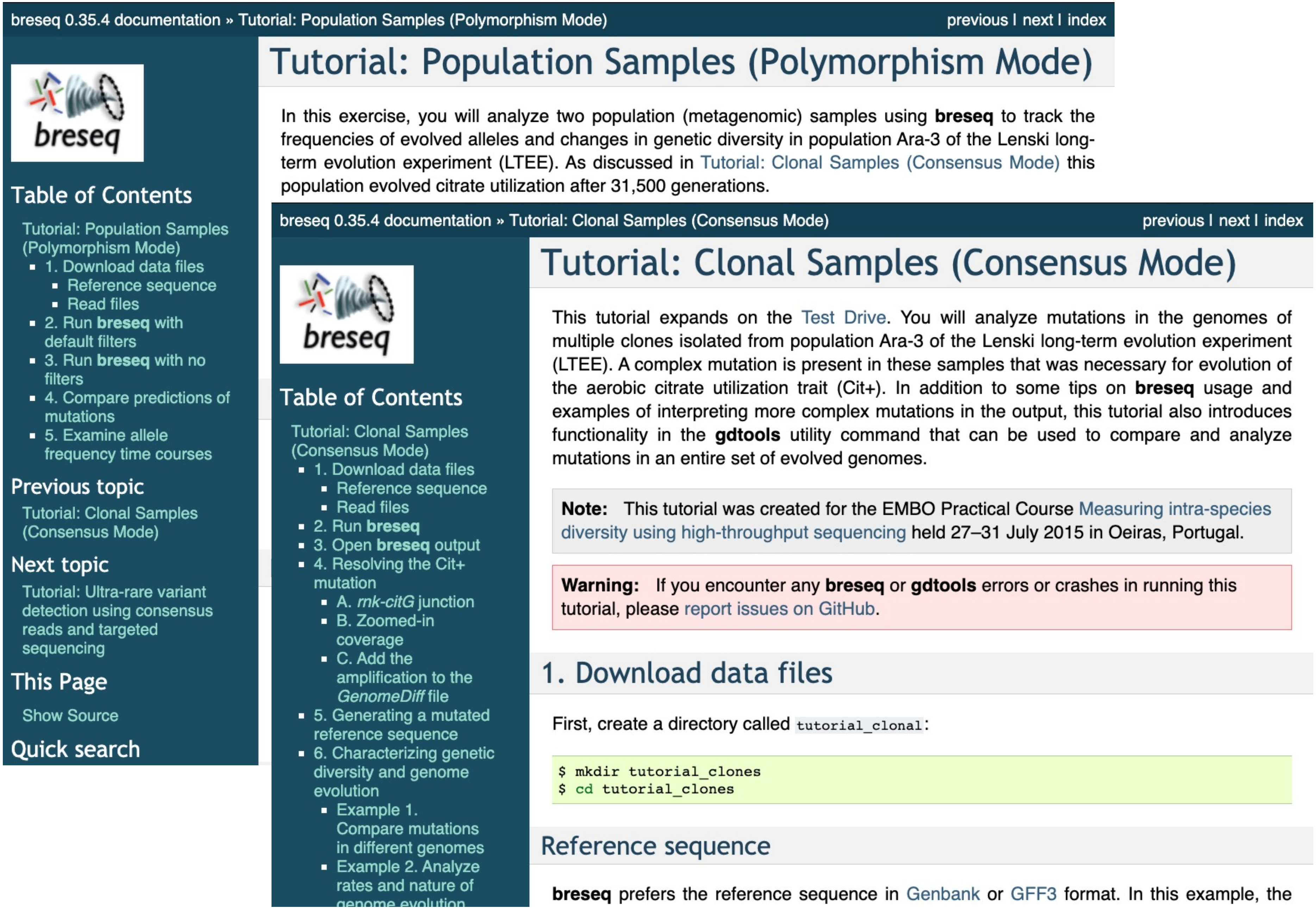
```
$ breseq BAM2COV
  -o coverage.png
  REL606:3480047-3486047
```



```
AAGACACCATGCACGCAGAATTTAACGCTCTGCGCGCCCAGGTGGCGATTAACGATGGTAATCCG  >  REL606/3483015-3483079
                                |
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               <  1:2369690/36-1 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               >  1:577628/1-36 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               >  2:1772887/1-36 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               <  1:130379/36-1 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               <  2:3079501/36-1 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               >  1:1820887/1-36 (MQ=255)
aagaCACCATGCACGCAGAATTTAACGCTCTgagcg                               <  1:2369308/36-1 (MQ=255)
 agaCACCATGCACGCAGAATTTAACGCTCTgagcgc                              >  2:3469595/1-36 (MQ=255)
 agaCACCATGCACGCAGAATTTAACGCTCTgagcgc                              <  2:1489970/36-1 (MQ=255)
    cacCATGCACGCAGAATTTAACGCTCTGAGCGCCCa                           >  1:1927484/1-36 (MQ=255)
    cacCATGCACGCAGAATTTAACGCTCTGAGCGCCCa                           <  2:2734863/36-1 (MQ=255)
    cacCATGCACGCAGAATTTAACGCTCTGAGCGCCCa                           <  2:2587112/36-1 (MQ=255)
    cacCATGCACGCAGAATTTAACGCTCTGAGCGCCCa                           <  2:1926447/36-1 (MQ=255)
     acCATGCACGCAGAATTTAACGCTCTGAGCGCCCAg                          <  2:885743/36-1 (MQ=255)
      ccATGCACGCAGAATTTAACGCTCTGAGCGCCCAgg                         >  2:2448233/1-36 (MQ=255)
      ccATGCACGCAGAATTTAACGCTCTGAGCGCCCAgg                         <  1:3403951/36-1 (MQ=255)
      ccATGCACGCAGAATTTAACGCTCTGAGCGCCCAgg                         >  2:3361806/1-36 (MQ=255)
       cATGCACGCAGAATTTAACGCTCTGCGCGCCCAGGt                        >  2:3230993/1-36 (MQ=255)
        aTGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTg                       <  2:1743516/36-1 (MQ=255)
        aTGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTg                       <  2:3672937/36-1 (MQ=255)
        aTGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTg                       >  1:3325866/1-36 (MQ=255)
        aTGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTg                       <  1:3348771/36-1 (MQ=255)
         tGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTgg                      <  2:3403193/36-1 (MQ=255)
         tGCACGCAGAATTTAACGCTCTGAGCGCCCAGGTgg                      >  2:1611056/1-36 (MQ=255)
          gCACGCTGAATTTAACGCTCTGAGCGCCCAGGTGGc                     >  1:2589008/1-36 (MQ=38)
           taCGCAGAATTTAACGTTCTGAGCGCCCAGGTGGCg                    <  1:2979881/35-1 (MQ=25)
```



These can help with identifying copy number changes (e.g, duplications) and understanding complex structural variation.



breseq 0.35.4 documentation » Tutorial: Population Samples (Polymorphism Mode) previous | next | index

Tutorial: Population Samples (Polymorphism Mode)

In this exercise, you will analyze two population (metagenomic) samples using **breseq** to track the frequencies of evolved alleles and changes in genetic diversity in population Ara-3 of the Lenski long-term evolution experiment (LTEE). As discussed in Tutorial: Clonal Samples (Consensus Mode) this population evolved citrate utilization after 31,500 generations.

Table of Contents

Previous topic

Tutorial: Clonal Samples (Consensus Mode)

Next topic

Tutorial: Ultra-rare variant detection using consensus reads and targeted sequencing

This Page

Show Source

Quick search

breseq 0.35.4 documentation » Tutorial: Clonal Samples (Consensus Mode) previous | next | index

Tutorial: Clonal Samples (Consensus Mode)

This tutorial expands on the Test Drive. You will analyze mutations in the genomes of multiple clones isolated from population Ara-3 of the Lenski long-term evolution experiment (LTEE). A complex mutation is present in these samples that was necessary for evolution of the aerobic citrate utilization trait (Cit+). In addition to some tips on **breseq** usage and examples of interpreting more complex mutations in the output, this tutorial also introduces functionality in the **gdtools** utility command that can be used to compare and analyze mutations in an entire set of evolved genomes.

Note: This tutorial was created for the EMBO Practical Course Measuring intra-species diversity using high-throughput sequencing held 27–31 July 2015 in Oeiras, Portugal.

Warning: If you encounter any **breseq** or **gdtools** errors or crashes in running this tutorial, please report issues on GitHub.

Table of Contents

1. Download data files

First, create a directory called `tutorial_clonal`:

```
$ mkdir tutorial_clones
$ cd tutorial_clones
```

Reference sequence

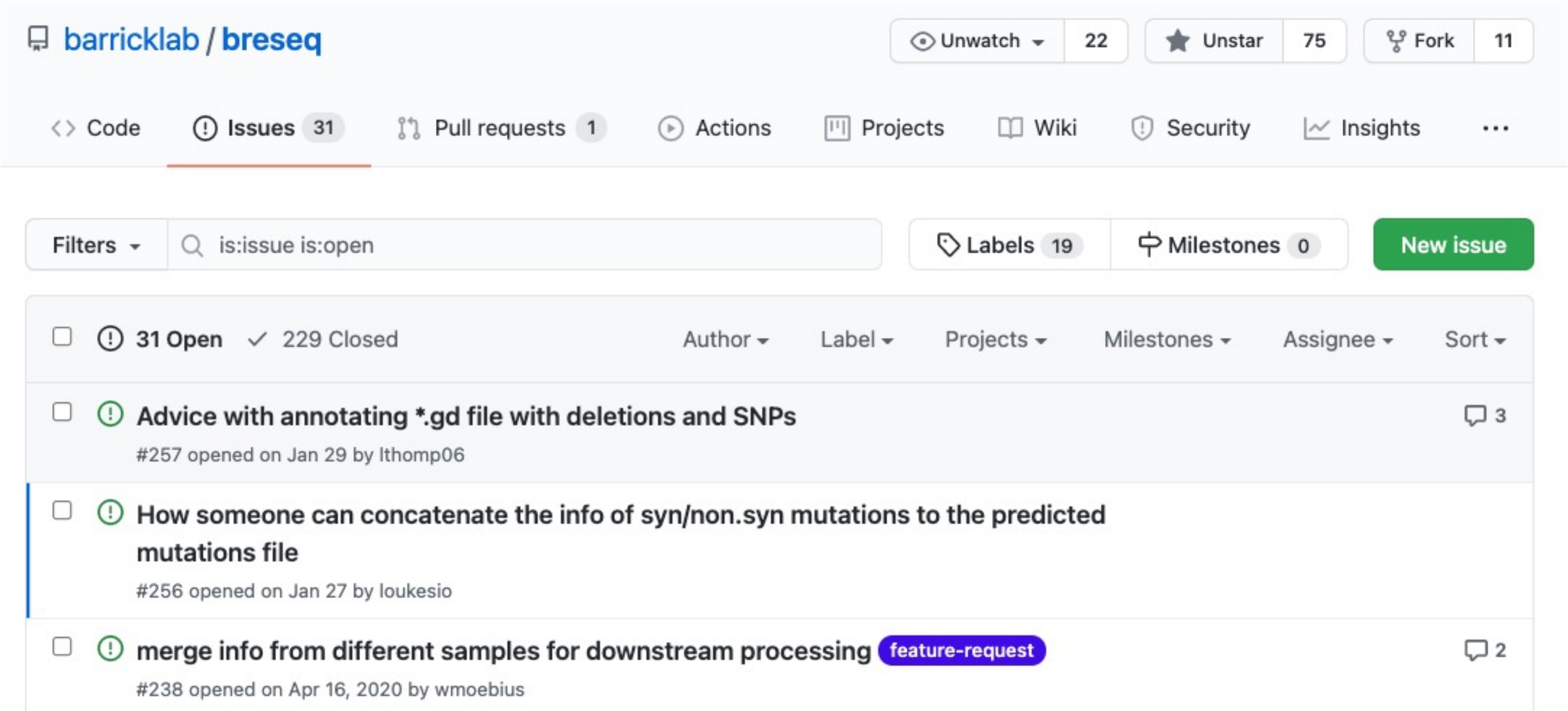
breseq prefers the reference sequence in Genbank or GFF3 format. In this example, the

Let us know how we can help!

These slides can be downloaded at http://barricklab.org/breseq

Post bug reports and issues on GitHub

Please check that you are using the newest *breseq* version first!



□ Advice with annotating *.gd file with deletions and SNPs — 3
#257 opened on Jan 29 by lthomp06

□ How someone can concatenate the info of syn/non.syn mutations to the predicted mutations file
#256 opened on Jan 27 by loukesio

□ merge info from different samples for downstream processing feature-request — 2
#238 opened on Apr 16, 2020 by wmoebius

Acknowledgments

Breseq Developers

Dan Deatherage
David Knoester
Geoffrey Colburn
Matt Strand
Jordan Borges
Aaron Reba

Funding

NIH K99/R00
(GM087550)

NSF CAREER
(CBET-1554179)

NSF BEACON Center
(DBI-0939454)

Thanks to many *breseq* users and research collaborators who have given feedback over the past decade!

Including Richard Lenski, Dominique Schneider, Olivier Tenaillon, Vaughn Cooper, Michael Desai, Yousif Shamoo, Zachary Blount, Genoscope, the Gulbenkian Institute, and members of these and many other research groups and communities.