Workshop Format

We'll be going through these topics today:

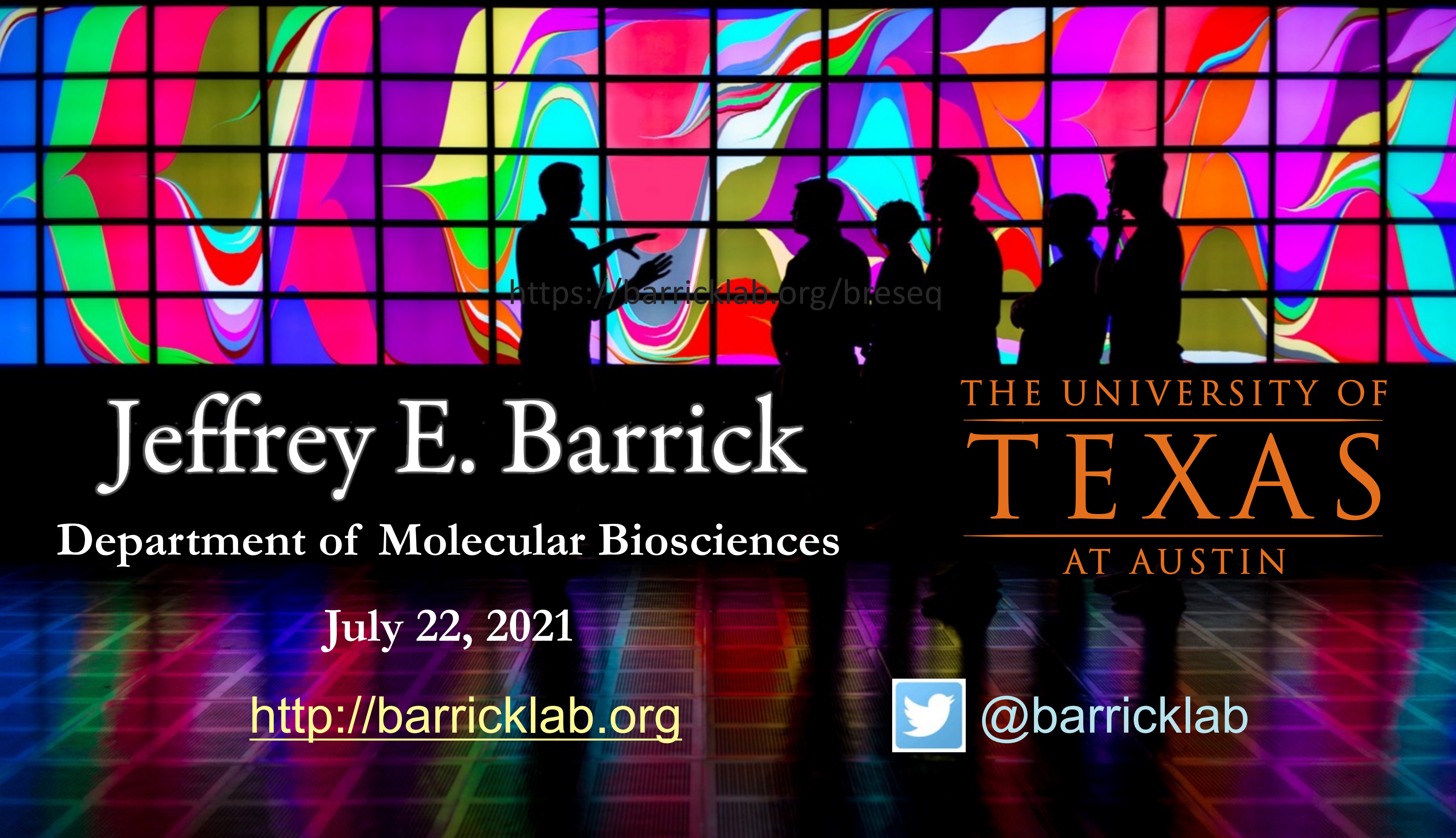
- Genome Diff format / manual curation of results
- Applying/subtracting mutations to update a reference seq
- Comparing many samples, counting mutations
- Export for analysis in other programs
- Phylogenetic trees, Muller plots

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

It will have links to some of the files we are looking at.

breseq :: Advanced Topics



https://barricklab.org/breseq

Jeffrey E. Barrick

Department of Molecular Biosciences

THE UNIVERSITY OF TEXAS AT AUSTIN

July 22, 2021

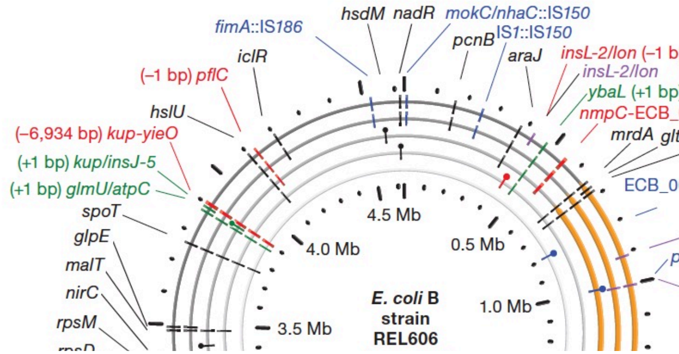
http://barricklab.org

@barricklab

Some uses of *breseq*

Genetics

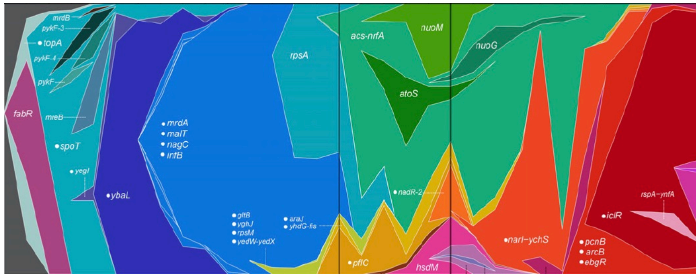
- Mechanisms of antibiotic resistance
- Mapping suppressor mutations



Barrick *et al.* (2009) *Nature*

Experimental evolution

- Rates/nature of genome evolution
- Genetic diversity in populations



Maddamsetti *et al.* (2015) *Genetics*

Biotechnology

- Verifying engineered plasmids/genomes
- Understanding beneficial mutations that arise during adaptive laboratory evolution

Last Time: Running *breseq*…

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

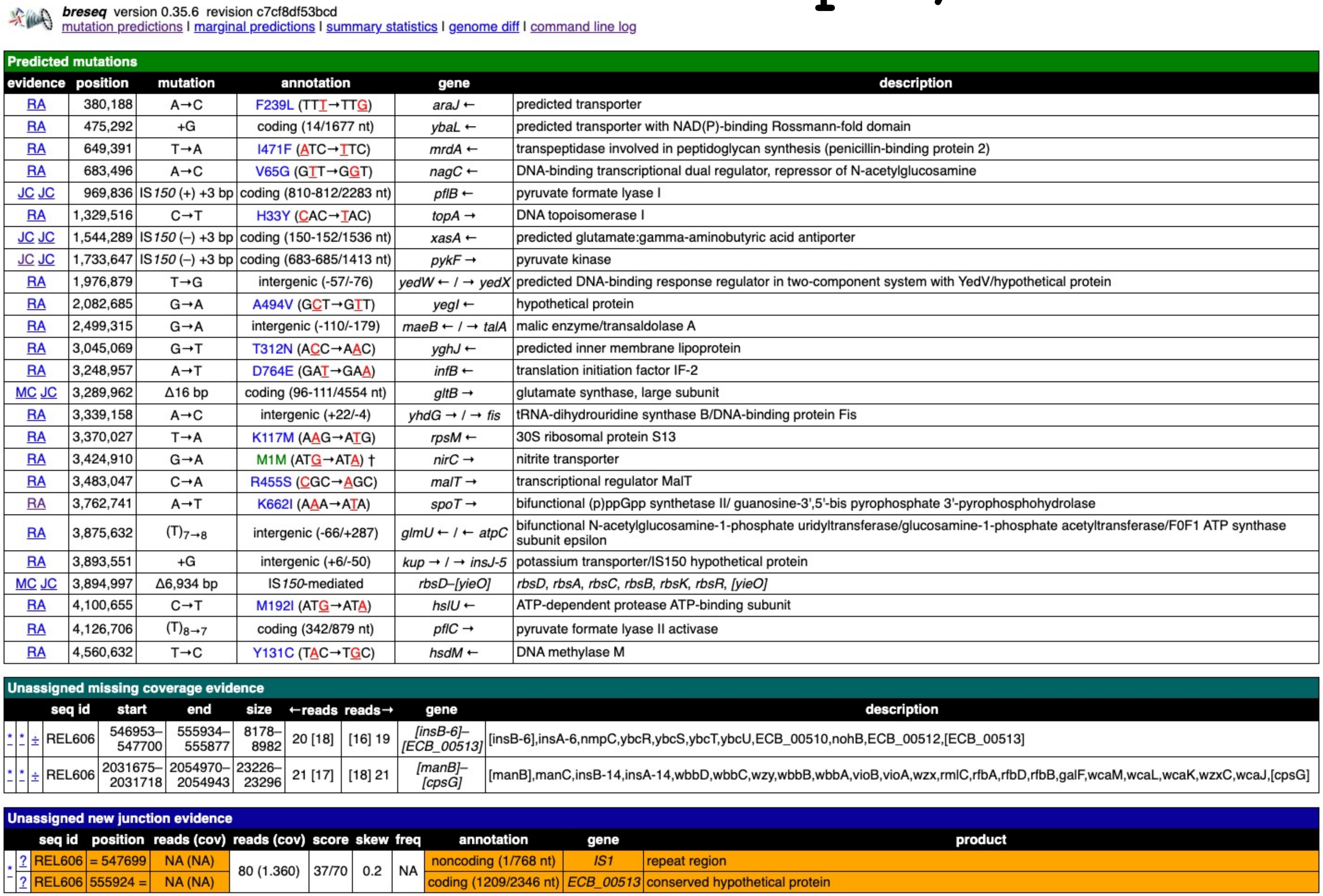
```
$ breseq
```

View all options

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

...and Viewing HTML Output

`output/index.html`



breseq version 0.35.6 revision c7cf8df53bcd
mutation predictions | marginal predictions | summary statistics | genome diff | command line log

Predicted mutations

evidence	position	mutation	annotation	gene	description
RA	380,188	A→C	F239L (TTT→TTG)	*araJ* ←	predicted transporter
RA	475,292	+G	coding (14/1677 nt)	*ybaL* ←	predicted transporter with NAD(P)-binding Rossmann-fold domain
RA	649,391	T→A	I471F (ATC→TTC)	*mrdA* ←	transpeptidase involved in peptidoglycan synthesis (penicillin-binding protein 2)
RA	683,496	A→C	V65G (GTT→GGT)	*nagC* ←	DNA-binding transcriptional dual regulator, repressor of N-acetylglucosamine
JC JC	969,836	IS*150* (+) +3 bp	coding (810-812/2283 nt)	*pflB* ←	pyruvate formate lyase I
RA	1,329,516	C→T	H33Y (CAC→TAC)	*topA* →	DNA topoisomerase I
JC JC	1,544,289	IS*150* (–) +3 bp	coding (150-152/1536 nt)	*xasA* ←	predicted glutamate:gamma-aminobutyric acid antiporter
JC JC	1,733,647	IS*150* (–) +3 bp	coding (683-685/1413 nt)	*pykF* →	pyruvate kinase
RA	1,976,879	T→G	intergenic (-57/-76)	*yedW* ← / → *yedX*	predicted DNA-binding response regulator in two-component system with YedV/hypothetical protein
RA	2,082,685	G→A	A494V (GCT→GTT)	*yegI* ←	hypothetical protein
RA	2,499,315	G→A	intergenic (-110/-179)	*maeB* ← / → *talA*	malic enzyme/transaldolase A
RA	3,045,069	G→T	T312N (ACC→AAC)	*yghJ* ←	predicted inner membrane lipoprotein
RA	3,248,957	A→T	D764E (GAT→GAA)	*infB* ←	translation initiation factor IF-2
MC JC	3,289,962	Δ16 bp	coding (96-111/4554 nt)	*gltB* →	glutamate synthase, large subunit
RA	3,339,158	A→C	intergenic (+22/-4)	*yhdG* → / → *fis*	tRNA-dihydrouridine synthase B/DNA-binding protein Fis
RA	3,370,027	T→A	K117M (AAG→ATG)	*rpsM* ←	30S ribosomal protein S13
RA	3,424,910	G→A	M1M (ATG→ATA) †	*nirC* →	nitrite transporter
RA	3,483,047	C→A	R455S (CGC→AGC)	*malT* →	transcriptional regulator MalT
RA	3,762,741	A→T	K662I (AAA→ATA)	*spoT* →	bifunctional (p)ppGpp synthetase II/ guanosine-3',5'-bis pyrophosphate 3'-pyrophosphohydrolase
RA	3,875,632	$(\mathrm{T})_{7\rightarrow 8}$	intergenic (-66/+287)	*glmU* ← / ← *atpC*	bifunctional N-acetylglucosamine-1-phosphate uridyltransferase/glucosamine-1-phosphate acetyltransferase/F0F1 ATP synthase subunit epsilon
RA	3,893,551	+G	intergenic (+6/-50)	*kup* → / → *insJ-5*	potassium transporter/IS150 hypothetical protein
MC JC	3,894,997	Δ6,934 bp	IS*150*-mediated	*rbsD*–*[yieO]*	*rbsD*, *rbsA*, *rbsC*, *rbsB*, *rbsK*, *rbsR*, *[yieO]*
RA	4,100,655	C→T	M192I (ATG→ATA)	*hslU* ←	ATP-dependent protease ATP-binding subunit
RA	4,126,706	$(\mathrm{T})_{8\rightarrow 7}$	coding (342/879 nt)	*pflC* →	pyruvate formate lyase II activase
RA	4,560,632	T→C	Y131C (TAC→TGC)	*hsdM* ←	DNA methylase M

Unassigned missing coverage evidence

			seq id	start	end	size	←reads	reads→	gene	description
*	*	÷	REL606	546953– 547700	555934– 555877	8178– 8982	20 [18]	[16] 19	*[insB-6]–* *[ECB_00513]*	[insB-6],insA-6,nmpC,ybcR,ybcS,ybcT,ybcU,ECB_00510,nohB,ECB_00512,[ECB_00513]
*	*	÷	REL606	2031675– 2031718	2054970– 2054943	23226– 23296	21 [17]	[18] 21	*[manB]–* *[cpsG]*	[manB],manC,insB-14,insA-14,wbbD,wbbC,wzy,wbbB,wbbA,vioB,vioA,wzx,rmlC,rfbA,rfbD,rfbB,galF,wcaM,wcaL,wcaK,wzxC,wcaJ,[cpsG]

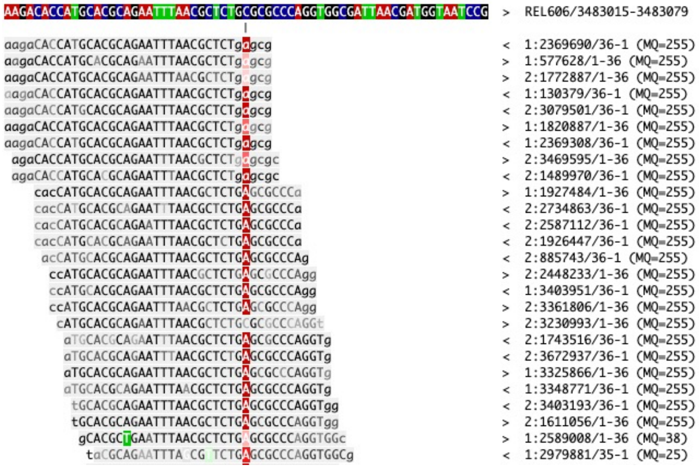
Unassigned new junction evidence

		seq id	position	reads (cov)	reads (cov)	score	skew	freq	annotation	gene	product
*	?	REL606	= 547699	NA (NA)	80 (1.360)	37/70	0.2	NA	noncoding (1/768 nt)	*IS1*	repeat region
	?	REL606	555924 =	NA (NA)					coding (1209/2346 nt)	*ECB_00513*	conserved hypothetical protein

Utilities to explore output

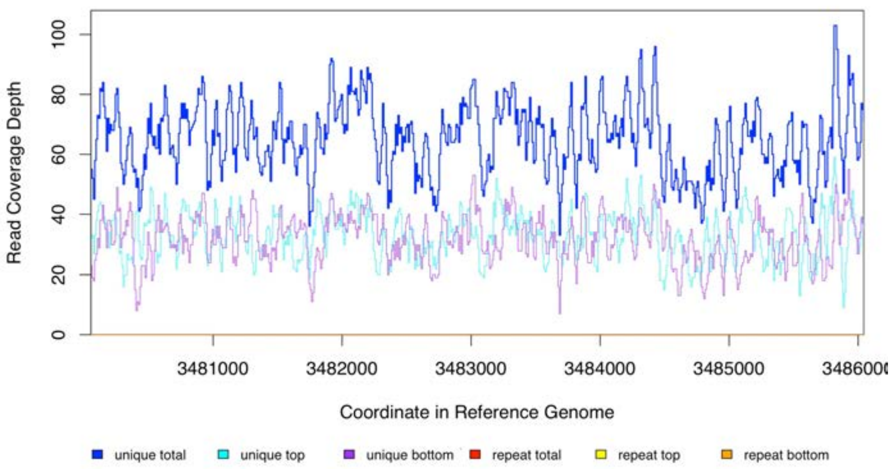
You can run utility subcommands from inside the main output directory of a *breseq* run. `$ breseq --help` to see others.

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



```
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)
           taCGCAGAATTTA CG TCTGAGCGCCCAGGTGGCg                     < 1:2979881/35-1 (MQ=25)
```

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

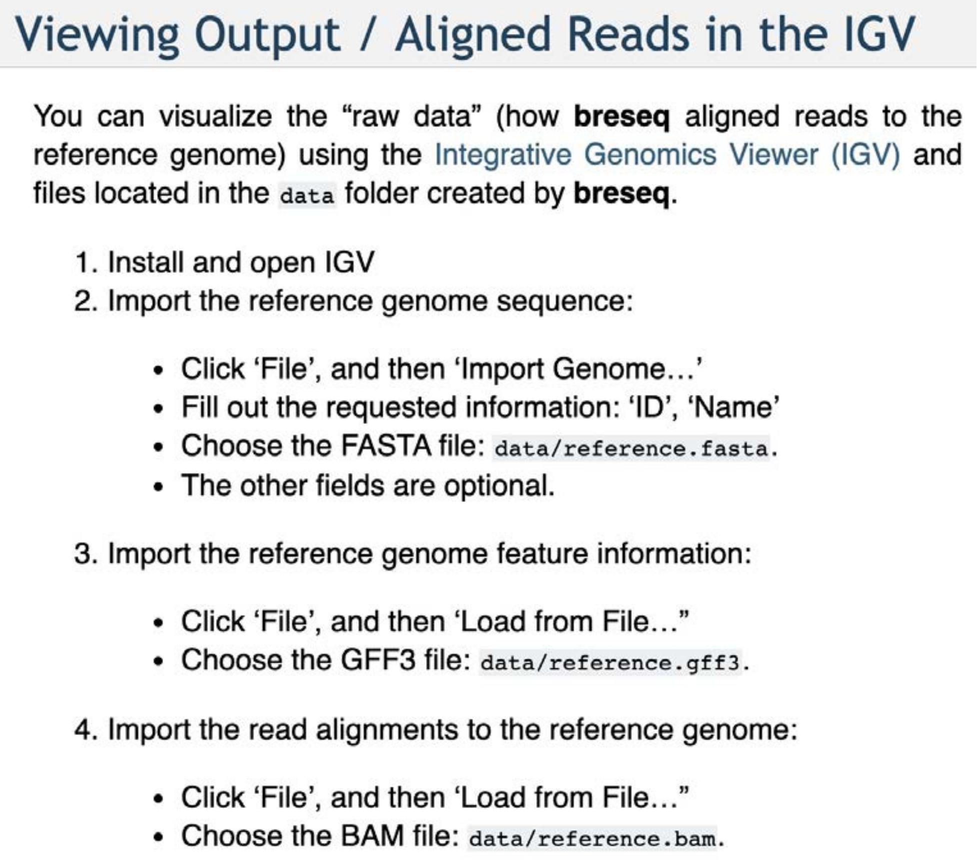


These can help with identifying copy number changes (e.g, duplications) and understanding complex structural variation.

Explore aligned reads using IGV

Integrative Genomics Viewer

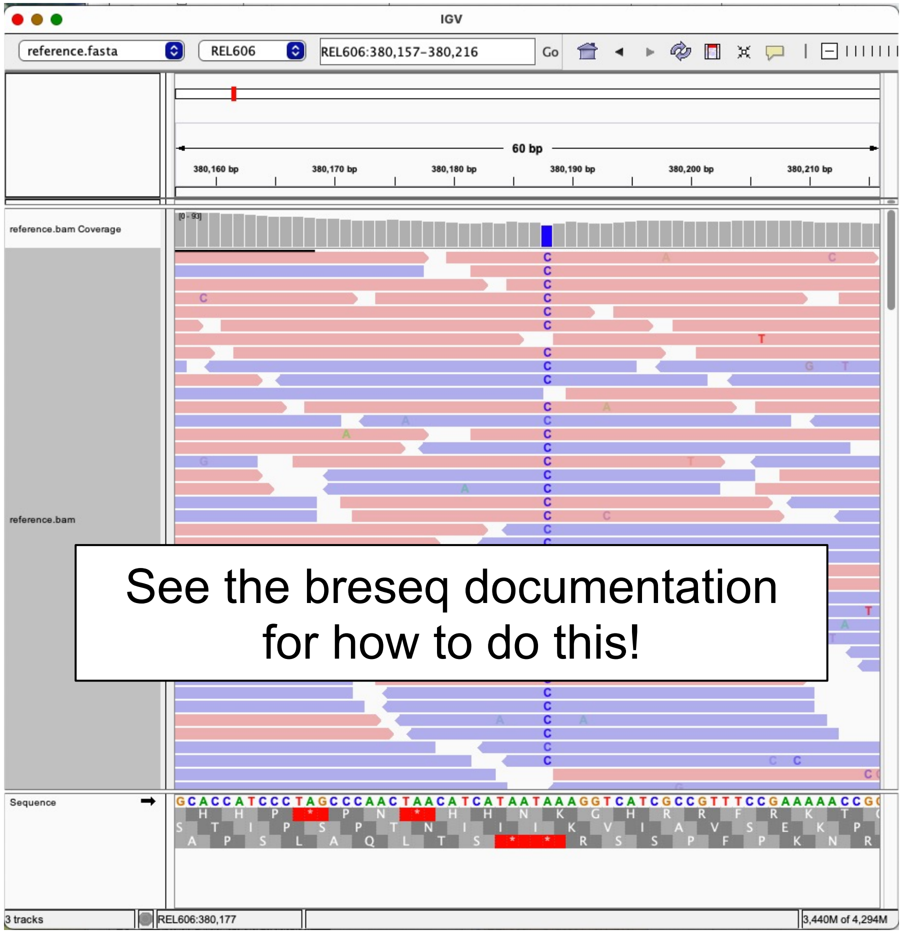
https://software.broadinstitute.org/software/igv/



Viewing Output / Aligned Reads in the IGV

You can visualize the “raw data” (how **breseq** aligned reads to the reference genome) using the Integrative Genomics Viewer (IGV) and files located in the `data` folder created by **breseq**.

1. Install and open IGV
2. Import the reference genome sequence:
   - Click ‘File’, and then ‘Import Genome…’
   - Fill out the requested information: ‘ID’, ‘Name’
   - Choose the FASTA file: `data/reference.fasta`.
   - The other fields are optional.
3. Import the reference genome feature information:
   - Click ‘File’, and then ‘Load from File…”
   - Choose the GFF3 file: `data/reference.gff3`.
4. Import the read alignments to the reference genome:
   - Click ‘File’, and then ‘Load from File…”
   - Choose the BAM file: `data/reference.bam`.



See the breseq documentation for how to do this!

Load mutations in VCF format

Today: GenomeDiff output

Machine-readable text file describing mutations

```
#=GENOME_DIFF    1.0
#=CREATED   15:16:00 24 May 2021
#=PROGRAM   breseq 0.35.6 revision c7cf8df53bcd
#=COMMAND   breseq -j 8 -o tests/long_Ara-1_10000gen_4536A ...
#=REFSEQ    tests/long_Ara-1_10000gen_4536A/../data/long_tests/REL606.gbk
#=READSEQ   tests/long_Ara-1_10000gen_4536A/../data/long_tests/SRR030255_1.fastq.gz
#=READSEQ   tests/long_Ara-1_10000gen_4536A/../data/long_tests/SRR030255_2.fastq.gz
#=CONVERTED-BASES 295047936
#=CONVERTED-READS 8195776
#=INPUT-BASES    298701576
#=INPUT-READS    8297266
#=MAPPED-BASES   277772336
#=MAPPED-READS   7750270
SNP   1     29    REL606      380188      C
INS   2     32    REL606      475292      G
SNP   3     36    REL606      649391      A
SNP   4     37    REL606      683496      C
MOB   5     101,102     REL606      969836      IS150 1     3
SNP   6     41    REL606      1329516     T
MOB   7     103,109     REL606      1544289     IS150 -1    3
MOB   8     110,111     REL606      1733647     IS150 -1    3
SNP   9     46    REL606      1976879     G
SNP   10    49    REL606      2082685     A
...
```

GenomeDiff format

`output/output.gd`

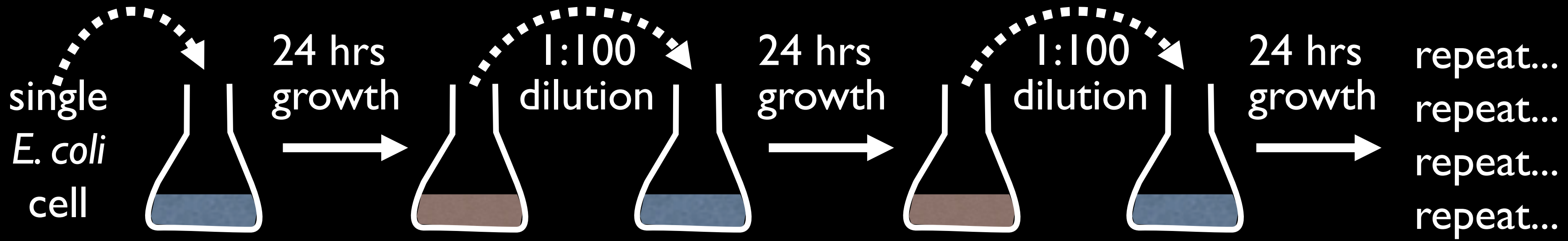
Format specification provided in the *breseq* manual...
Let’s look at that now!

Today: `gdtools` utility commands

Installed/included with *breseq*

```
Usage: gdtools [COMMAND] [OPTIONS]

Manipulate Genome Diff (*.gd) files using the following commands.

General:
    VALIDATE                check formatting of input files
    APPLY                   apply mutations to a sequence
    ANNOTATE (or COMPARE)   annotate the effects of mutations and compare multiple samples
    MUTATIONS               (re)predict mutations from evidence
    NORMALIZE               normalize mutation positions and annotations


Set and Filtering Operations:
    SUBTRACT                remove mutations in one file from another
    INTERSECT               keep shared mutations in two files
    UNION/MERGE             combine mutations, removing duplicates
    FILTER/REMOVE           remove mutations matching specified conditions
    MASK                    remove mutation predictions in masked regions


Format Conversions:
    GD2VCF                  GD to Variant Call Format (VCF)
    VCF2GD                  Variant Call Format(VCF) to GD
    GD2GVF                  GD to Genome Variation Format (GVF)

Analysis:
    COUNT                   count statistics for different types of mutations
    PHYLOGENY               create maximum parsimony tree from mutations (requires PHYLIP)

TACC Utilities:
    DOWNLOAD                download reference and read files from GD header info
    RUNFILE                 create a commands file and launcher script for use on TACC
```

Lenski Long-Term Evolution Experiment



single *E. coli* cell

24 hrs growth

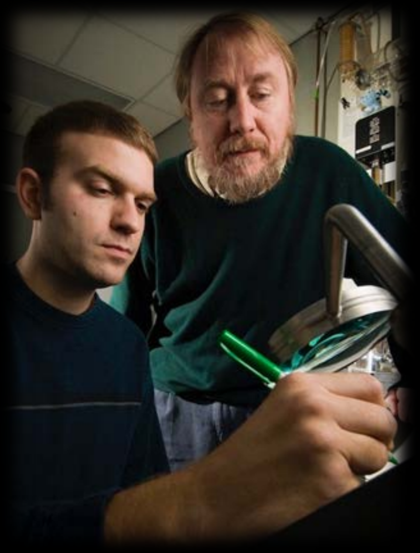
1:100 dilution

24 hrs growth

1:100 dilution

24 hrs growth

repeat...
repeat...
repeat...
repeat...



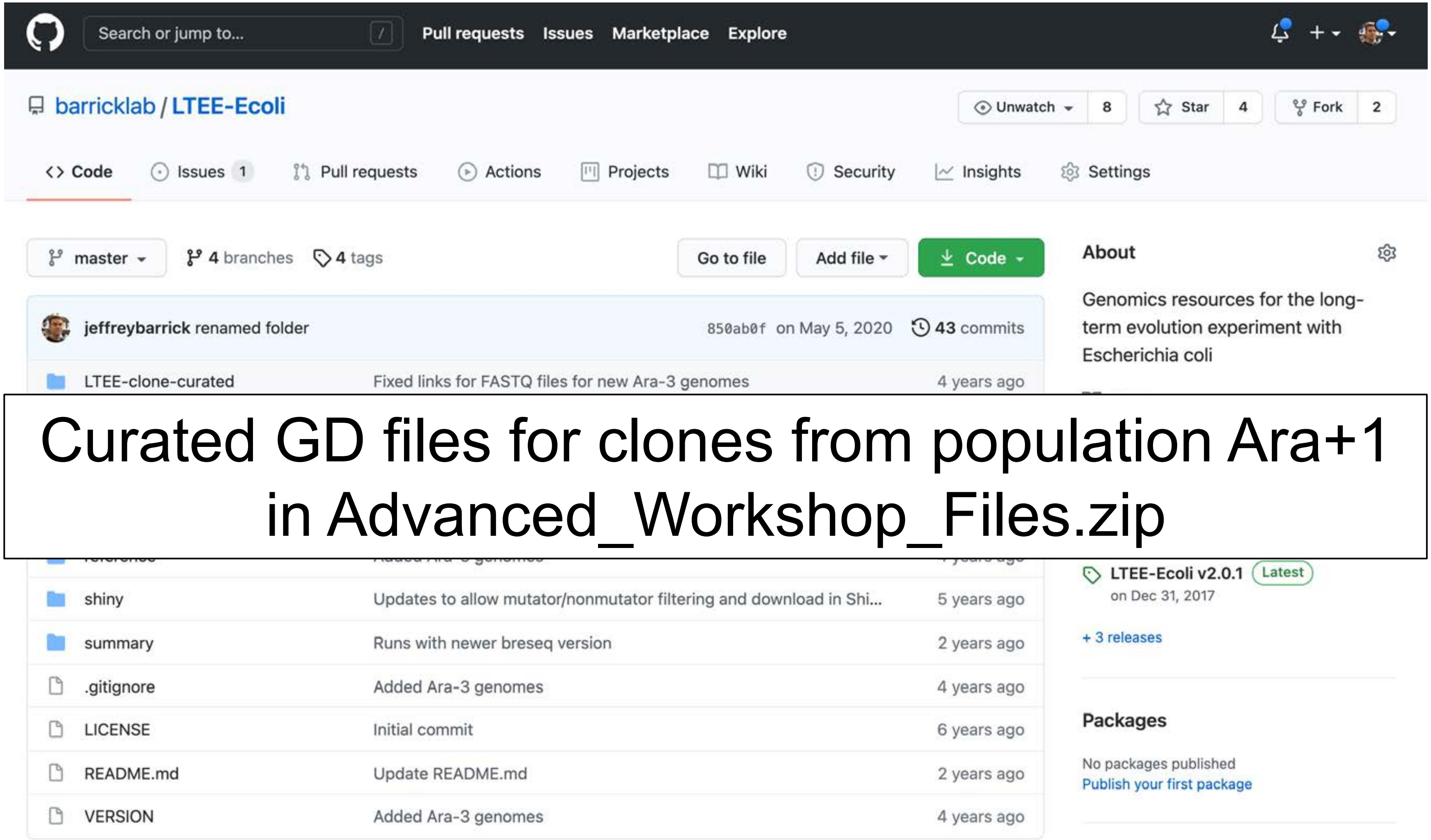
Richard Lenski
Michigan State

- ❖ 12 independent populations
- ❖ Deep evolutionary history
- ❖ Viable frozen "fossil record"

>73,000 generations of
E. coli growth (>30 years)!

LTEE-Ecoli Repository

https://github.com/barricklab/LTEE-Ecoli



jeffreybarrick renamed folder 850ab0f on May 5, 2020 43 commits

LTEE-clone-curated	Fixed links for FASTQ files for new Ara-3 genomes	4 years ago
reference	Added Ara-3 genomes	4 years ago
shiny	Updates to allow mutator/nonmutator filtering and download in Shi...	5 years ago
summary	Runs with newer breseq version	2 years ago
.gitignore	Added Ara-3 genomes	4 years ago
LICENSE	Initial commit	6 years ago
README.md	Update README.md	2 years ago
VERSION	Added Ara-3 genomes	4 years ago

About

Genomics resources for the long-term evolution experiment with Escherichia coli

LTEE-Ecoli v2.0.1 Latest
on Dec 31, 2017

+ 3 releases

Packages

No packages published
Publish your first package

Curated GD files for clones from population Ara+1 in Advanced_Workshop_Files.zip

gdtools APPLY

Uses the specified mutations to update the reference file (ex: from ancestor to evolved)

```
$ gdtools APPLY –f GFF3 –o updated.gff3
      -r reference.gbk input.gd
```

If the output of running *breseq* on the updated reference comes back with no changes, you (most likely) found all mutations!

```
$ breseq –r updated.gff3 reads.fastq
```

Manual curation of GenomeDiffs

For a complete, high-quality set of mutations, you will likely have to add and subtract some mutations manually from **`output.gd`**.

Use an industrial strength text editor!

- Turn off line wrapping
- Show invisible characters (tabs vs spaces)

Check the formatting of your edited file:

```
$ gdtools VALIDATE -r reference.gbk input.gd
```

Options for dealing with mutations in the ancestor of your experiment

Often your strain has a few differences from the database sequence due to errors/evolution.

1. Generate an updated reference using **`gdtools APPLY`**. Run your samples against it.

   ****Downside:** sequence and gene positions in updated reference are shifted after indels. Comparing is harder.

2. Run the ancestor and samples against the database reference. Use **`gdtools SUBTRACT`** to remove ancestral “mutations” before analysis.

```
$ gdtools SUBTRACT –o new.gd sample.gd ref.gd
```

Export to R / Excel

Use COMPARE to create a tab-separated values (TSV) file for loading into R or Excel.

```
$ gdtools COMPARE –f TSV clone1.gd clone2.gd ...
```

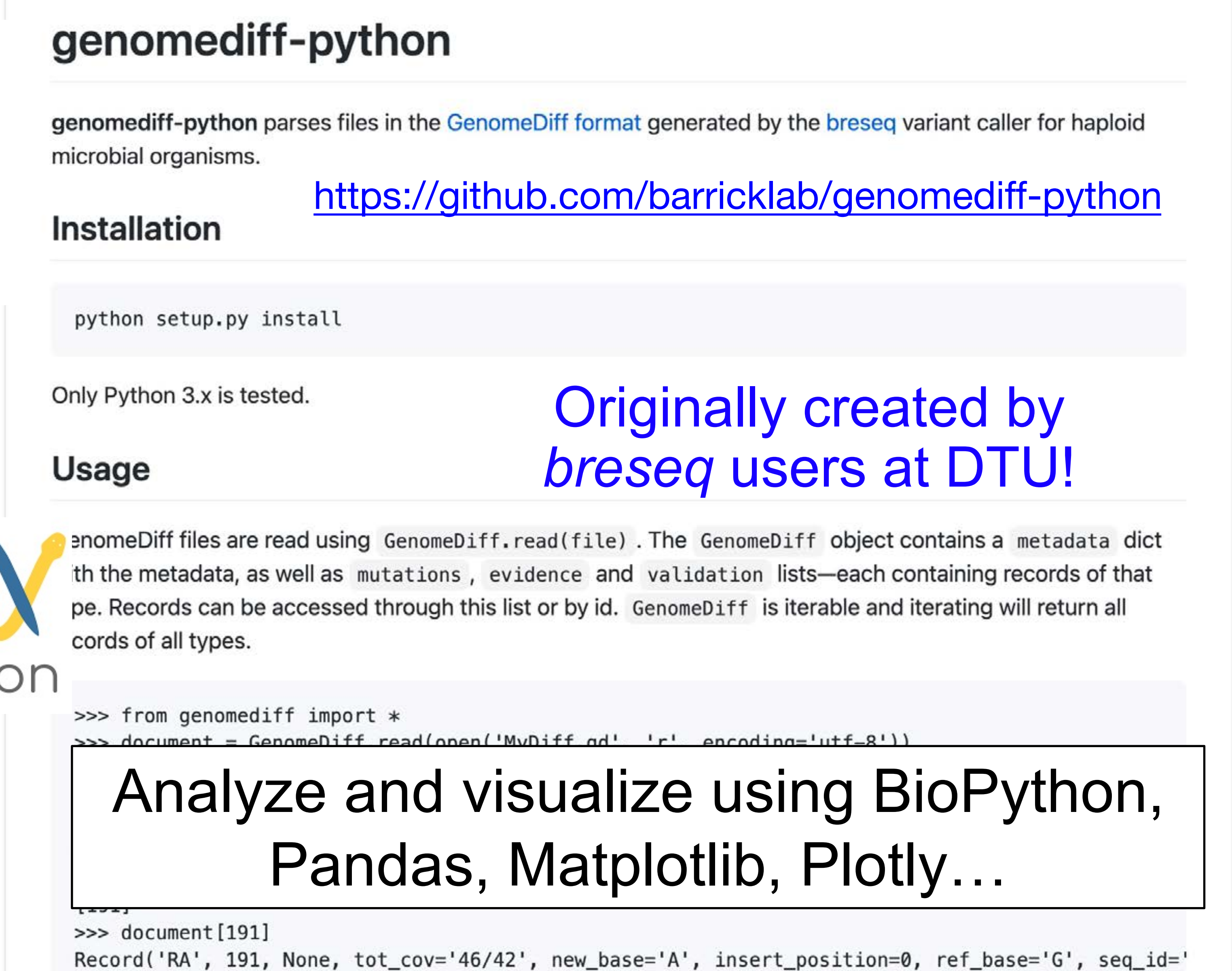
Typical workflow:

Load into R as a table

Wrangle using **tidyverse**

Plot using **ggplot2**

Export to Python



genomediff-python

genomediff-python parses files in the GenomeDiff format generated by the breseq variant caller for haploid microbial organisms.

https://github.com/barricklab/genomediff-python

Installation

```
python setup.py install
```

Only Python 3.x is tested.

Usage

Originally created by *breseq* users at DTU!

enomeDiff files are read using `GenomeDiff.read(file)`. The `GenomeDiff` object contains a `metadata` dict th the metadata, as well as `mutations`, `evidence` and `validation` lists—each containing records of that pe. Records can be accessed through this list or by id. `GenomeDiff` is iterable and iterating will return all cords of all types.

```
>>> from genomediff import *
>>> document = GenomeDiff.read(open('MyDiff.gd', 'r', encoding='utf-8'))
[191]
>>> document[191]
Record('RA', 191, None, tot_cov='46/42', new_base='A', insert_position=0, ref_base='G', seq_id='
```

Analyze and visualize using BioPython, Pandas, Matplotlib, Plotly…

Other ways to export

VCF (Variant call format)

Useful for other NGS tools (e.g., IGV).

GVF (Genome Variation format)

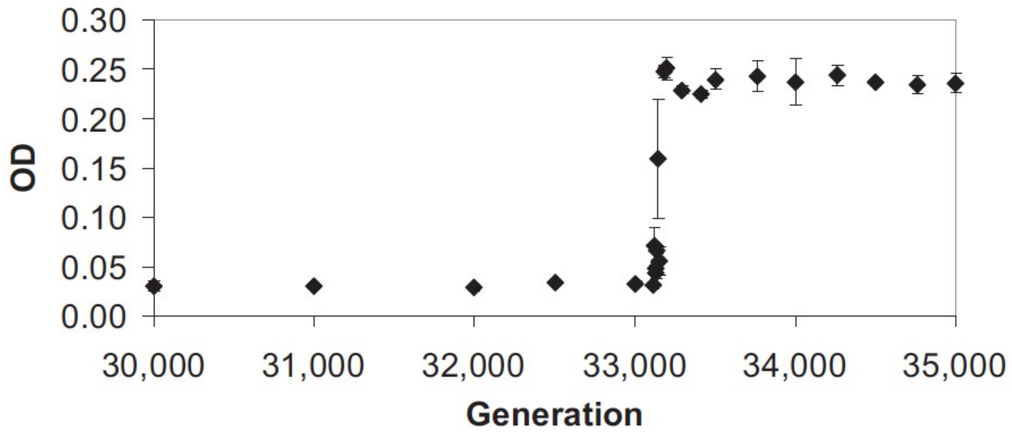
JSON (JavaScript Object Notation)

Useful for import into other scripting or programming languages (e.g., Javascript).

Analysis: Causative Mutations

How?

Backcross and sequence: Only **two mutations** required for strong Cit+ phenotype



Citrate utilization evolved after 33,000 generations in one LTEE population

Blount *et al.* (2008) *PNAS*

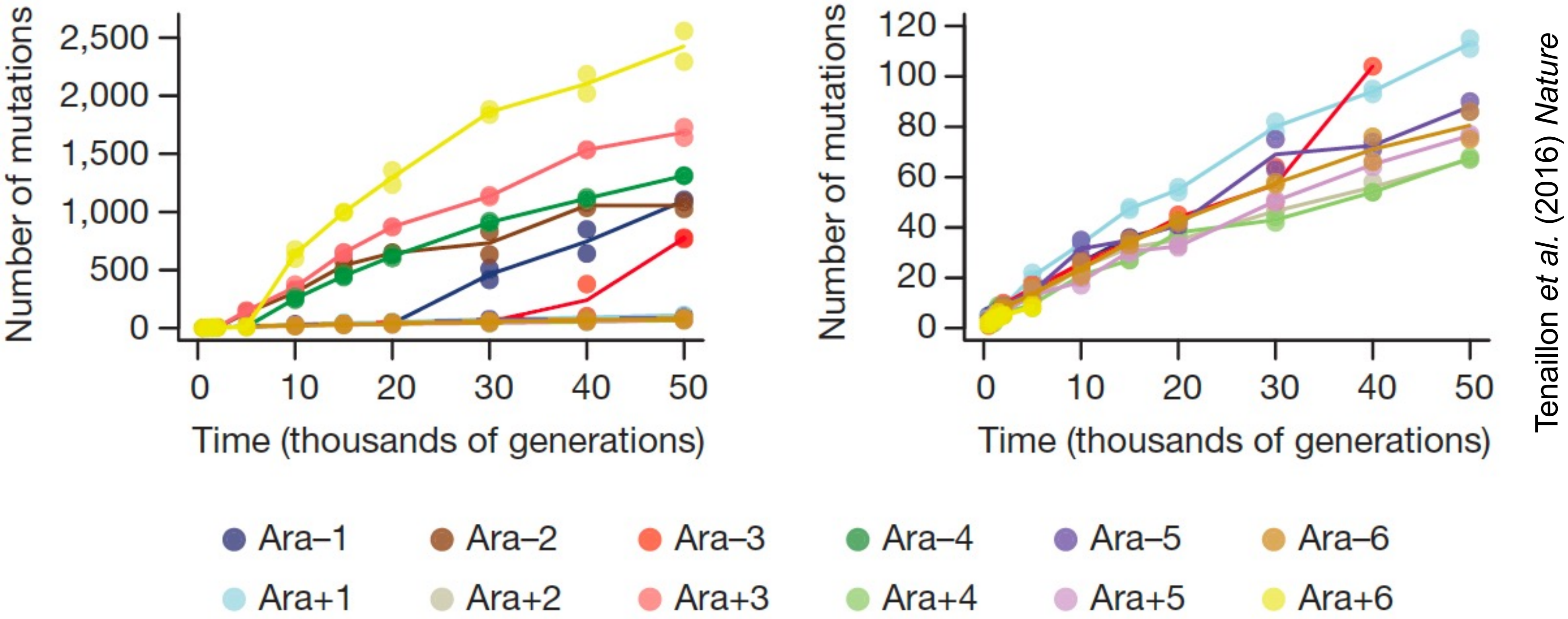
	Strain							
CZB 154	R1	R2.1	R2.2	R2.3	R2.4	R3	R4	**Allele**
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*yaaH*
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*yaaH/yaaW*
🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	*fadE*
🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟥	*ykgK/ykgL*
🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	*secF*
🟩	🟩	🟩	🟩	🟩	🟥	🟥	🟥	*tesB/ybaY*
🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	*ybaL*
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	*dsbG–insA-9*
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	***rnk–citG (citT)***
🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	*ybeF*
🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	*mrdB*
🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	*gltA*
🟥	🟦	🟦	🟦	🟥	🟦	🟦	🟦	*gltA/sdhC*
🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟥	*clpA/serW*
🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟥	*infA*
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*topA*
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*acpD*
🟩	🟩	🟩	🟥	🟩	🟥	🟥	🟥	*mokB/trg*
🟥	🟦	🟦	🟥	🟦	🟥	🟥	🟥	*yddK*
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*pykF*
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*ECB_02200/nuoN*
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*amiC*
🟥	🟦	🟥	🟥	🟥	🟥	🟥	🟥	*yqeB* (Δ)
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*yqeB*
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*arcB*
🟩	🟩	🟥	🟥	🟩	🟩	🟥	🟥	*fis*
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟥	*rpsG*
🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	*envZ*
🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟦	*ompR/greB*
🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟥	*malT*
🟥	🟥	🟦	🟥	🟥	🟦	🟥	🟥	*rhsB–rhsA*
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	***dctA/yhjK (dctA*)***
🟩	🟥	🟥	🟥	🟥	🟥	🟥	🟥	*waaT*
🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟥	*rbsD–yieO*
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*hslU*
🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	*metL*
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*iclR*
🟩	🟩	🟥	🟩	🟥	🟩	🟥	🟥	*yjeM*
🟩	🟩	🟥	🟩	🟥	🟥	🟥	🟥	*cycA*
🟩	🟩	🟥	🟥	🟥	🟥	🟥	🟥	*nadR*
79	50	28	32	23	15	18	7	**Evolved alleles**

Allele present: 🟩 CZB154 🟥 REL607 🟦 New

```
$ gdtools COMPARE...
```

Quandt *et al.* (2014) *PNAS*

Analysis: Rates of Evolution

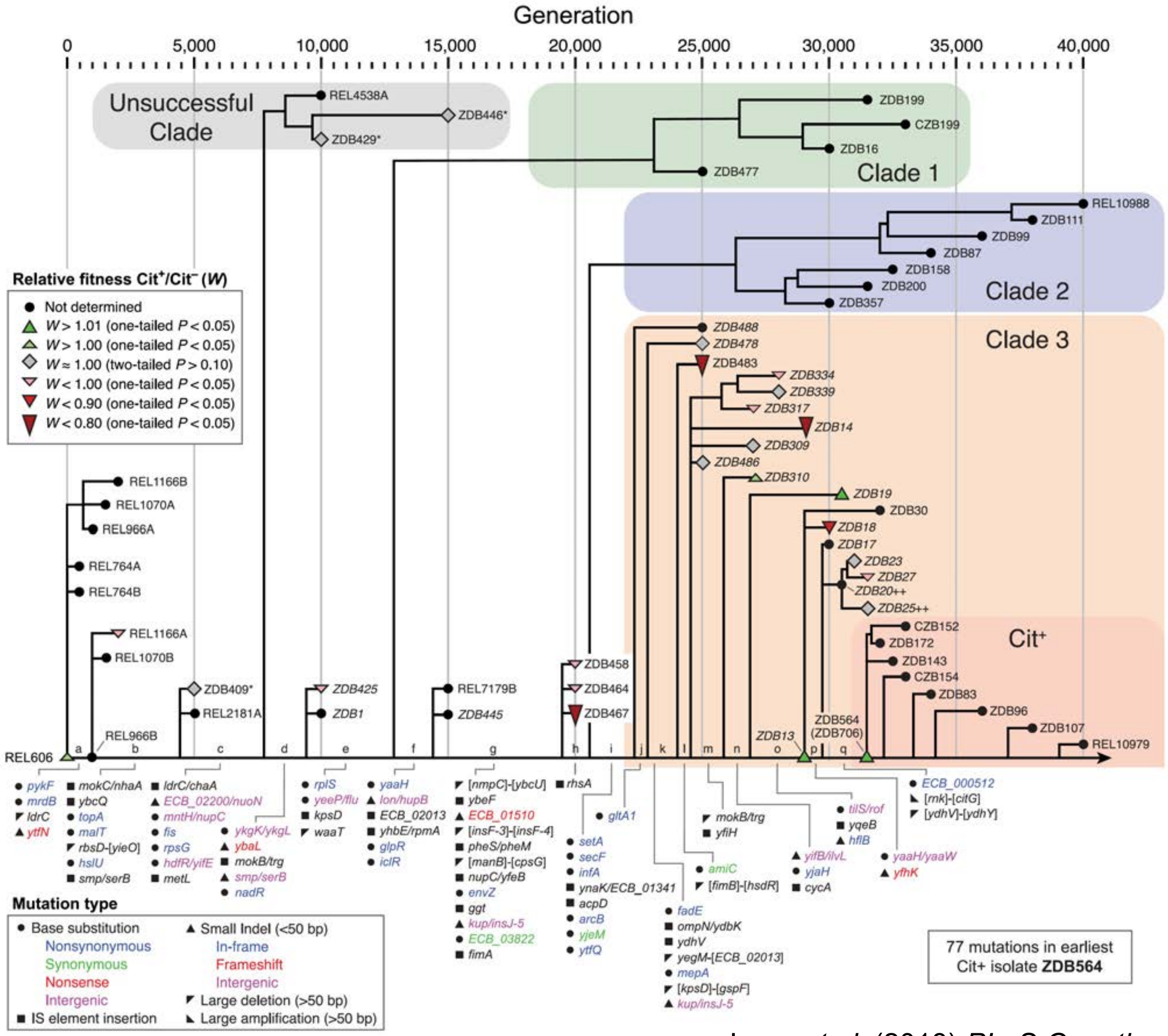


Tenaillon *et al.* (2016) *Nature*

```
$ gdtools COUNT ...
```

Outputs a table with different counts and genome statistics. Totals and divided out into specific types.

Analysis: Phylogenetic trees



Leon *et al.* (2018) *PLoS Genetics*

What mutations led to Cit+ evolution?

Generate an alignment of genomic changes

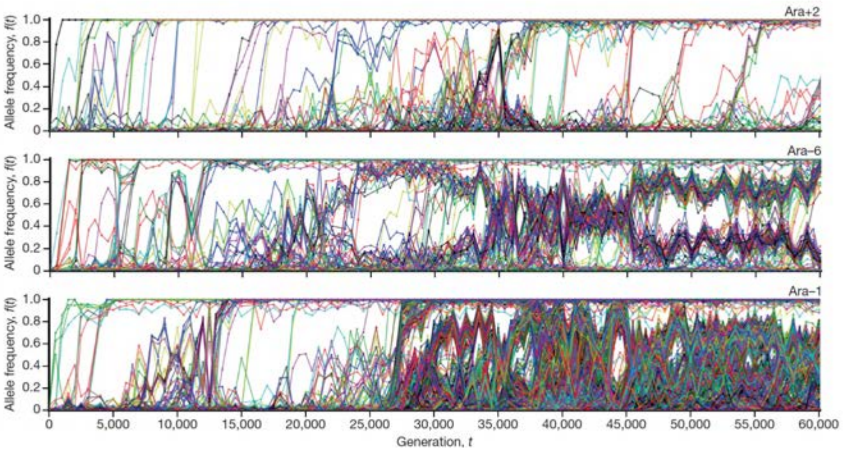
```
$ gdtools COMPARE
  -f PHYLIP clone1.gd
  clone2.gd ....
```

Build and visualize a maximum parsimony tree using PHYLIP, MEGAX, etc.

```
$ gdtools PHYLOGENY
  -r reference.gbk
  clone1.gd clone2.gd
  ....
```

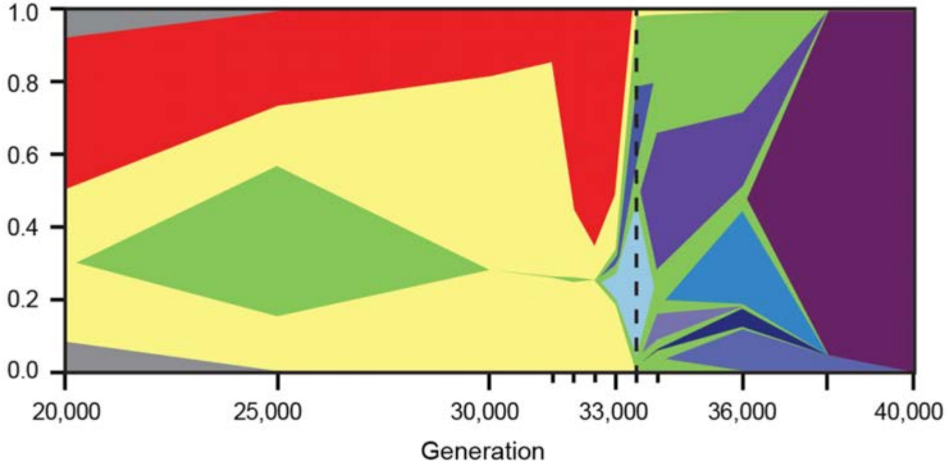
Analysis: Allele/Genotype Frequencies

Allele Frequency



Good *et al.* (2017) *Nature*

Muller Plot (Genotype Frequency)



Quandt *et al.* (2015) *eLife*

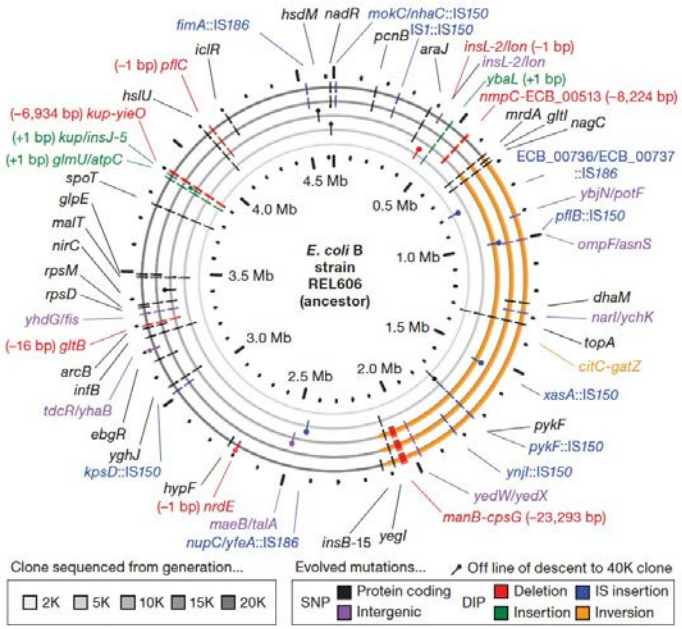
For tracking how genetic diversity evolves within populations, visualizing dynamics, selective sweeps, and stable coexistence.

```
gdtools COMPARE -f TSV pop1.gd pop2.gd ....
```

Programs/packages that can help:

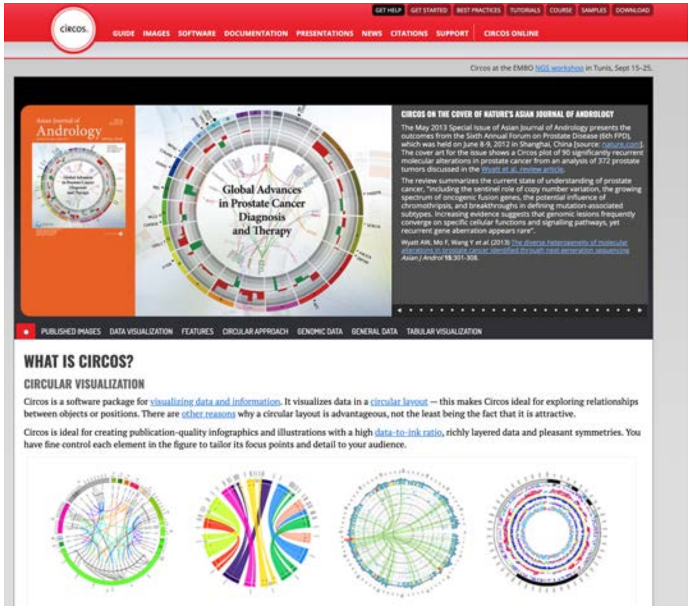
R, ggplot, ggMuller, EvoFreq, MullerPlot

Analysis: Genome Plots



Barrick *et al.* (2009) *Nature*

CIRCOS Plot



Circos at the EMBO NGS workshop in Tunis, Sept 15-25.

CIRCOS ON THE COVER OF NATURE'S ASIAN JOURNAL OF ANDROLOGY

The May 2013 Special issue of Asian Journal of Andrology presents the outcomes from the Sixth Annual Forum on Prostate Disease (6th FPD), which was held on June 8-9, 2012 in Shanghai, China [source: nature.com]. The cover art for the issue shows a Circos plot of 90 significantly recurrent molecular alterations in prostate cancer from an analysis of 372 prostate tumors discussed in the Wyatt et al. review article.

The review summarizes the current state of understanding of prostate cancer, "including the sentinel role of copy number variation, the growing spectrum of oncogenic fusion genes, the potential influence of chromothripsis, and breakthroughs in defining mutation-associated subtypes. Increasing evidence suggests that genomic lesions frequently converge on specific cellular functions and signalling pathways, yet recurrent gene aberration appears rare".

Wyatt AW, Mo F, Wang Y et al. (2013) The diverse heterogeneity of molecular alterations in prostate cancer identified through next generation sequencing *Asian J Androl* 15:301-308.

WHAT IS CIRCOS?

CIRCULAR VISUALIZATION

Circos is a software package for visualizing data and information. It visualizes data in a circular layout — this makes Circos ideal for exploring relationships between objects or positions. There are other reasons why a circular layout is advantageous, not the least being the fact that it is attractive.

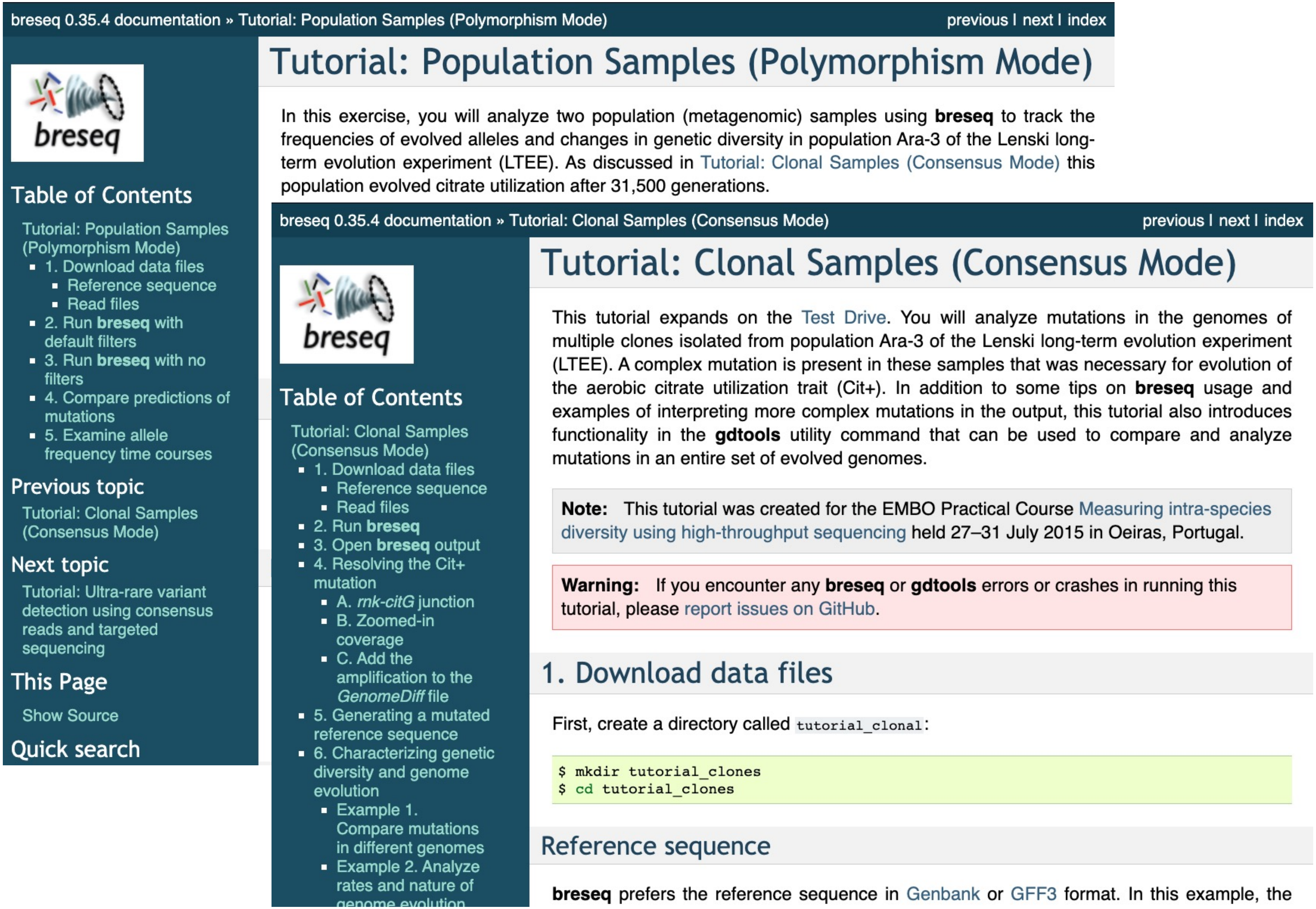
Circos is ideal for creating publication-quality infographics and illustrations with a high data-to-ink ratio, richly layered data and pleasant symmetries. You have fine control each element in the figure to tailor its focus points and detail to your audience.

```
$ gdtools GD2CIRCOS ...
```

+ Adobe Illustrator!

Do not use this. It is *BROKEN!*

We would love to have someone adopt this or come up with alternative visualizations!



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

Tutorial: Population Samples (Polymorphism Mode)

- 1. Download data files
  - Reference sequence
  - Read files
- 2. Run **breseq** with default filters
- 3. Run **breseq** with no filters
- 4. Compare predictions of mutations
- 5. Examine allele frequency time courses

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

Table of Contents

Tutorial: Clonal Samples (Consensus Mode)

- 1. Download data files
  - Reference sequence
  - Read files
- 2. Run **breseq**
- 3. Open **breseq** output
- 4. Resolving the Cit+ mutation
  - A. *rnk-citG* junction
  - B. Zoomed-in coverage
  - C. Add the amplification to the *GenomeDiff* file
- 5. Generating a mutated reference sequence
- 6. Characterizing genetic diversity and genome evolution
  - Example 1. Compare mutations in different genomes
  - Example 2. Analyze rates and nature of genome evolution

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.

1. Download data files

First, create a directory called `tutorial_clonal`:

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

Reference sequence

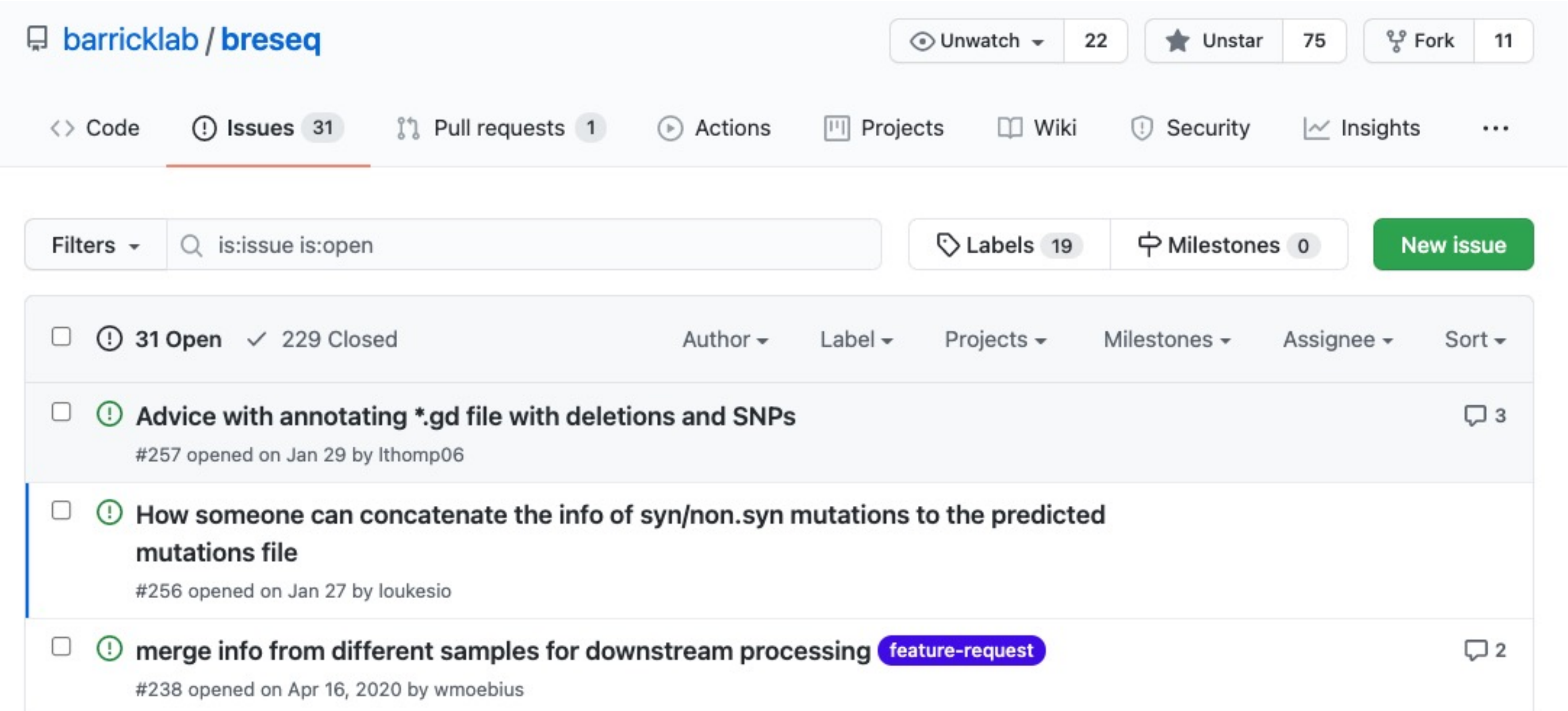
breseq prefers the reference sequence in Genbank or GFF3 format. In this example, the

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!



barricklab / breseq

is:issue is:open

□ 31 Open ✓ 229 Closed

□ 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.