RUM 2 (RNA-Seq Unified Mapper)

Mike DeLaurentis

University of Pennsylvania

August 15, 2012

Agenda

About me

- Software engineering background
- Working at Penn since January
- Experience in a variety of languages (Perl, Java, Clojure (lisp), Python, Ruby)

Agenda

About RUM

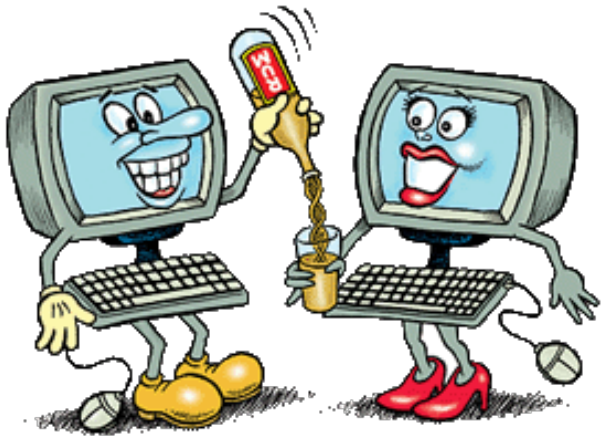
- "RUM is an alignment, junction calling, and feature quantification pipeline specifically designed for Illumina RNA-Seq data"
- Written by Gregory Grant (ggrant@grant.org)
- Runs on Linux / UNIX / Mac
- Can distribute work across multiple machines
- Written in Perl 5

The RUM Pipeline

- Align all reads against genome using Bowtie
- Align all reads against transcriptome using Bowtie
- Merge genome and transcriptome alignments and identify unmapped reads
- Align unmapped reads against genome using BLAT
- Merge Bowtie and Blat alignments
- Produce some output files based on merged results

RUM output files

- Files of unique and non-unique alignments
- All alignments in SAM format
- Coverage plots
- Feature quantifications
- Junction calls
- List of novel inferred internal exons



Phases

Preprocessing Split the input into N chunks.

Phases

Preprocessing Split the input into N chunks.

Processing Run Bowtie and Blat. Each chunk can be processed independently of the others.

Phases

Preprocessing Split the input into N chunks.

Processing Run Bowtie and Blat. Each chunk can be processed independently of the others.

Postprocessing Merge alignments for chunks together. Produce coverage files, quantifications, junction files, find novel internal exons.

Work distribution

- RUM can run alignment phase in parallel
- Splits input reads into N chunks
- Alignment for each chunk is independent of other chunks
- Results are merged together for coverage, quantification, junction calling

Agenda

RUM 2 Features

- Standard installation process
- New command-line interface
- Get status of running job
- Restart a job where it left off
- More reliable kill command
- Run a chunk or postprocessing by itself
- Relocatable indexes
- SAM file is closer to conforming to standard

Installing RUM

- Uses standard Perl Makefile.PL
- Should be familiar to system administrators
- Download tarball from `https://github.com/PGFI/rum/downloads`
- Run `perl Makefile.PL`
- Then install indexes...

Installing Indexes

- RUM needs an index for each organism you want to align against
- Index includes genome, gene annotations, and binary index files for Bowtie
- Run `rum_indexes`; it will guide you through the process

Command-line interface

Usage is `rum_runner ACTION [OPTIONS]` where action is one of:

- `align` - Run an alignment
- `status` - Check the status of a job
- `stop` - Stop a job (can be restarted later)
- `kill` - Stop a job and clean it up (to restart from scratch)
- `clean` - Remove output files for a job
- `help` - Get help
- `version` - Show version number

rum_runner align

Use `rum_runner align` to run an alignment:

```
rum_runner align \
  --output dir \
  --index  ~/rum_indexes/hg19 \
  --name   TestJob \
  --chunks 25 \
  ~/samples/forward.fq ~/samples/reverse.fq
```

Job status

- Use `rum_runner status` to check on the status of a running job.

```
Processing in 25 chunks
-----------------------
XXXXXXXXXXXXXXXXXXXXXXXXX Run bowtie on genome
XXXXXXXXXXXXXXXXXXXXXXXXX Parse genome Bowtie output
X XXX  XX XX XXXX  XXXXX  Run bowtie on transcriptome
X XXX  XX XX XXXX  XXXXX  Parse transcriptome Bowtie output
X XXX  XX XX XXXX  XXXXX  Merge unique mappers together
X XXX  XX XX XXXX  XXXXX  Merge non-unique mappers together
X XXX  XX XX XXXX  XXXXX  Make unmapped reads file for blat
X XXX  XX XX XXXX  XXXXX  Run blat on unmapped reads
X XXX  XX XX XXXX  XXXXX  Run mdust on unmapped reads
X XXX  XX XX XXXX  XXXXX  Parse blat output
X XXX  XX XX XXXX  XXXXX  Merge bowtie and blat results
X XX   XX XX  XXX   XXXX  Clean up RUM files
X XX   XX XX  XXX   XXXX  Produce RUM_Unique
X XX   XX XX  XXX   XXXX  Sort RUM_Unique by location
X X    XX XX  XXX   XXXX  Sort cleaned non-unique mappers by ID
X X    XX XX  XXX   XXXX  Remove duplicates from NU
X X    XX XX  XXX   XXXX  Create SAM file
X X    XX XX  XXX   XXXX  Create non-unique stats
X X    XX XX  XXX   XXXX  Sort RUM_NU
X X    XX XX  XXX   XXXX  Generate quants
...
```

Job status

```
Postprocessing
--------------
  Merge RUM_NU files
  Make non-unique coverage
  Merge RUM_Unique files
  Compute mapping statistics
  Make unique coverage
  Finish mapping stats
  Merge SAM headers
  Concatenate SAM files
  Merge quants
  make_junctions
  Sort junctions (all, bed) by location
  Sort junctions (all, rum) by location
  Sort junctions (high-quality, bed) by location
  Get inferred internal exons
  Quantify novel exons

All the chunk error log files are empty. That's good.
Main error log file is empty. That's good.

RUM is running (job ids 815718, 815720).
```

Recovering from errors

- Sometimes bad things happen
- RUM 2 allows easier recovery from infrastructure failures
- Running "`rum_runner align`" again will restart a job from where it left off
- Can save *a lot* of time when recovering from infrastructure failure
- Determines state of job by looking at which output files exist

Killing a job

- To stop a job and remove all of its output:

```
rum_runner kill -o dir
```

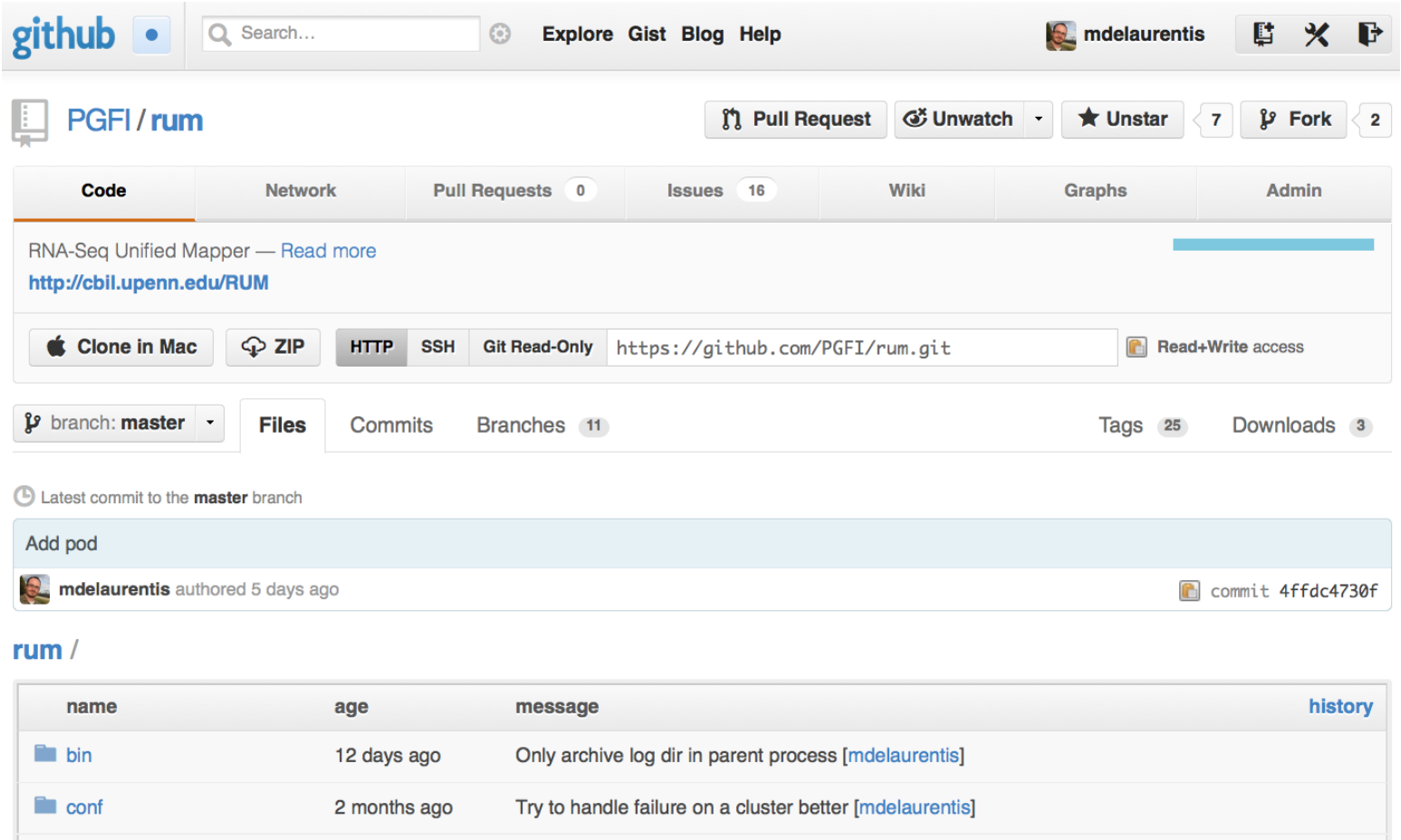
- Useful if you've run a job with incorrect parameters and need to start over

Agenda

Agenda

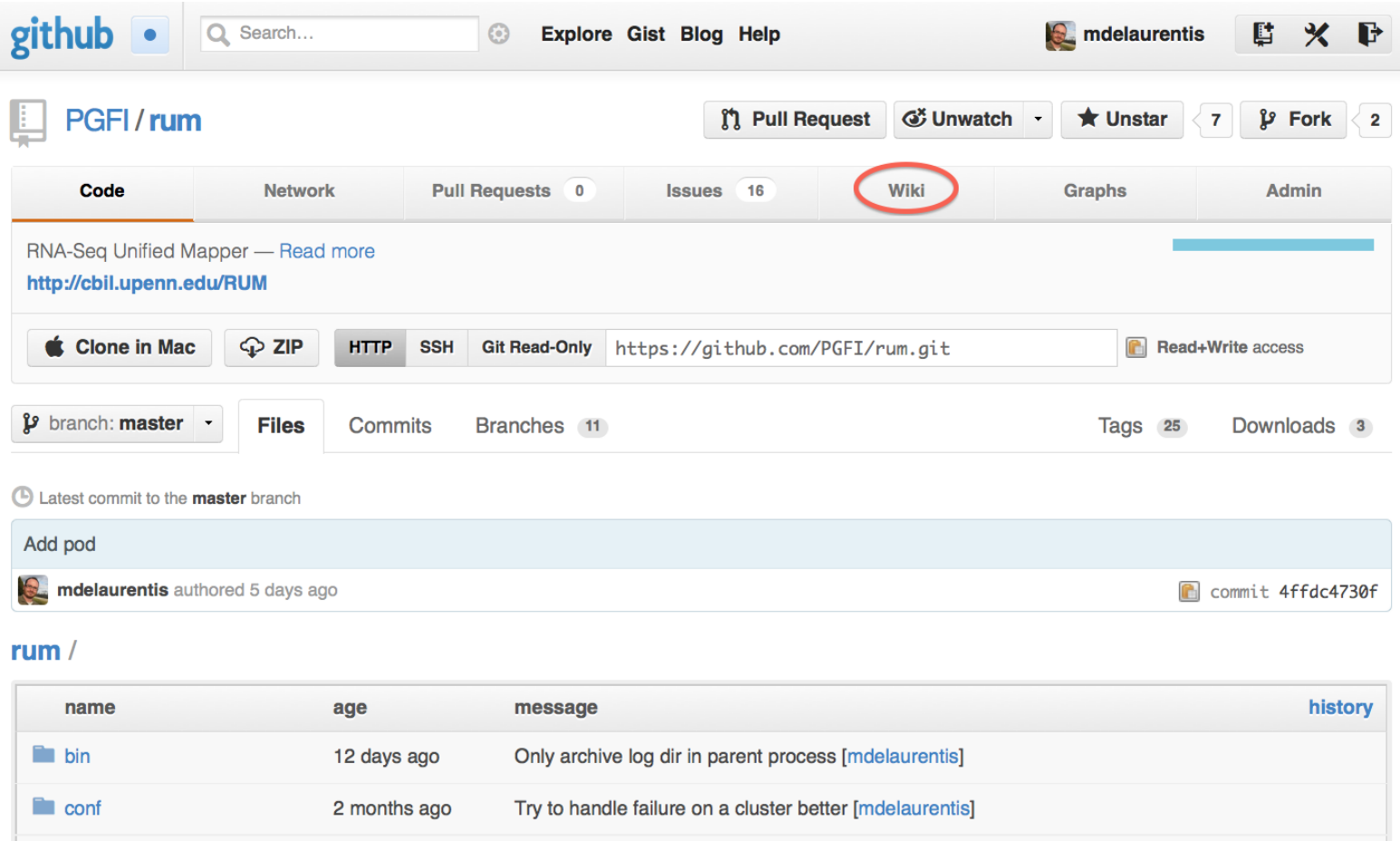
Main Github page

https://github.com/PGFI/rum



User guide

https://github.com/PGFI/rum/wiki



User guide

https://github.com/PGFI/rum/wiki

github | Search... | Explore Gist Blog Help | mdelaurentis

PGFI / rum | Pull Request | Unwatch | Unstar 7 | Fork 2

Code | Network | Pull Requests 0 | Issues 16 | Wiki | Graphs | Admin

Home | Pages | Wiki History | Git Access

Home

New Page | Edit Page | Page History

Welcome, gentle user. These pages describe the RNA-Seq Unified Mapper (RUM). We recommend reading them roughly in the order listed here.

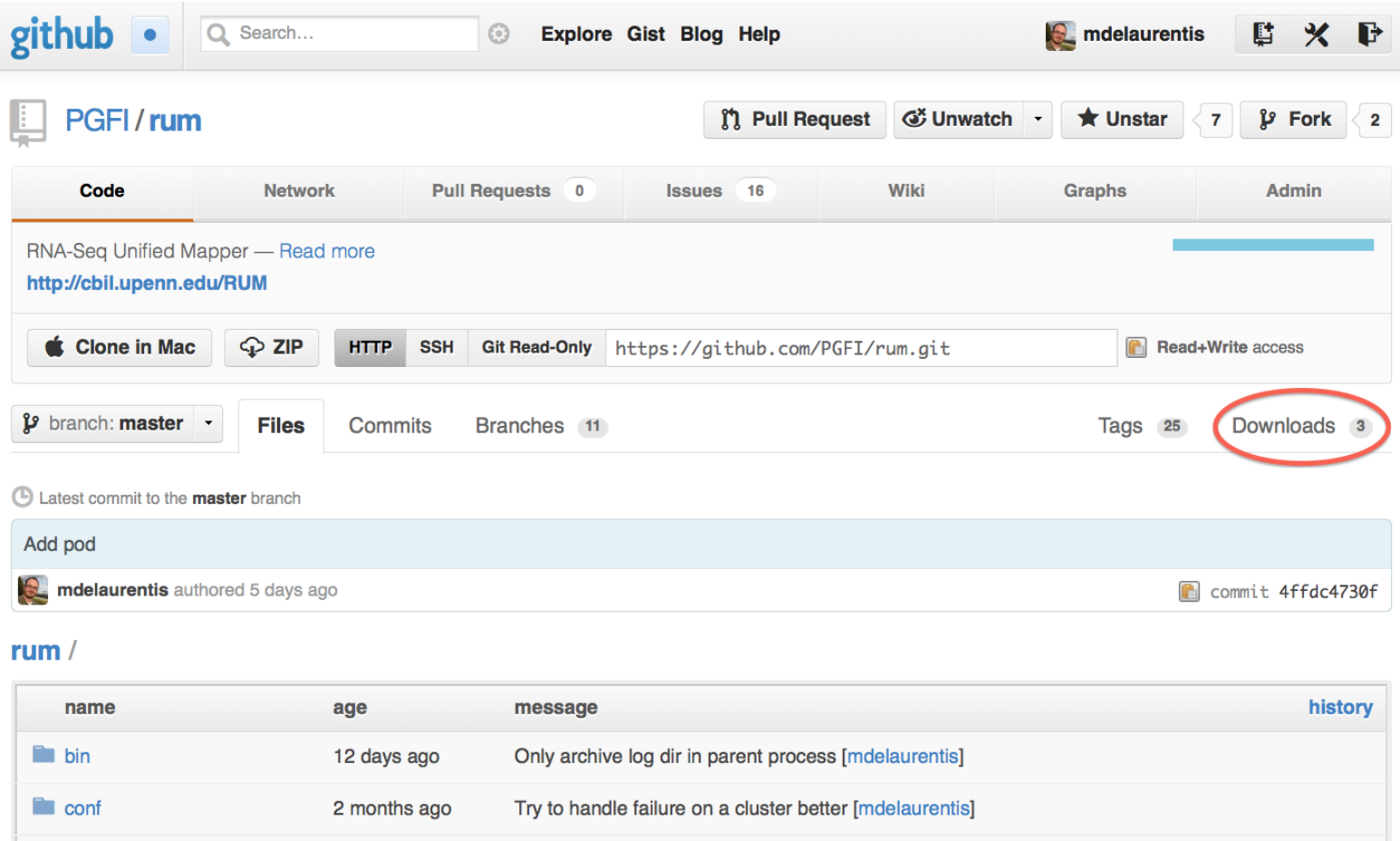
You can find the latest version of RUM at our downloads page. If you find any issues or have any feature requests, please enter them into our issues list. We would really appreciate your feedback.

User guide

1. About RUM
2. Installing RUM
3. Installing indexes
4. Creating indexes (optional)
5. Running RUM
6. RUM output files
7. Running RUM on a cluster (optional)

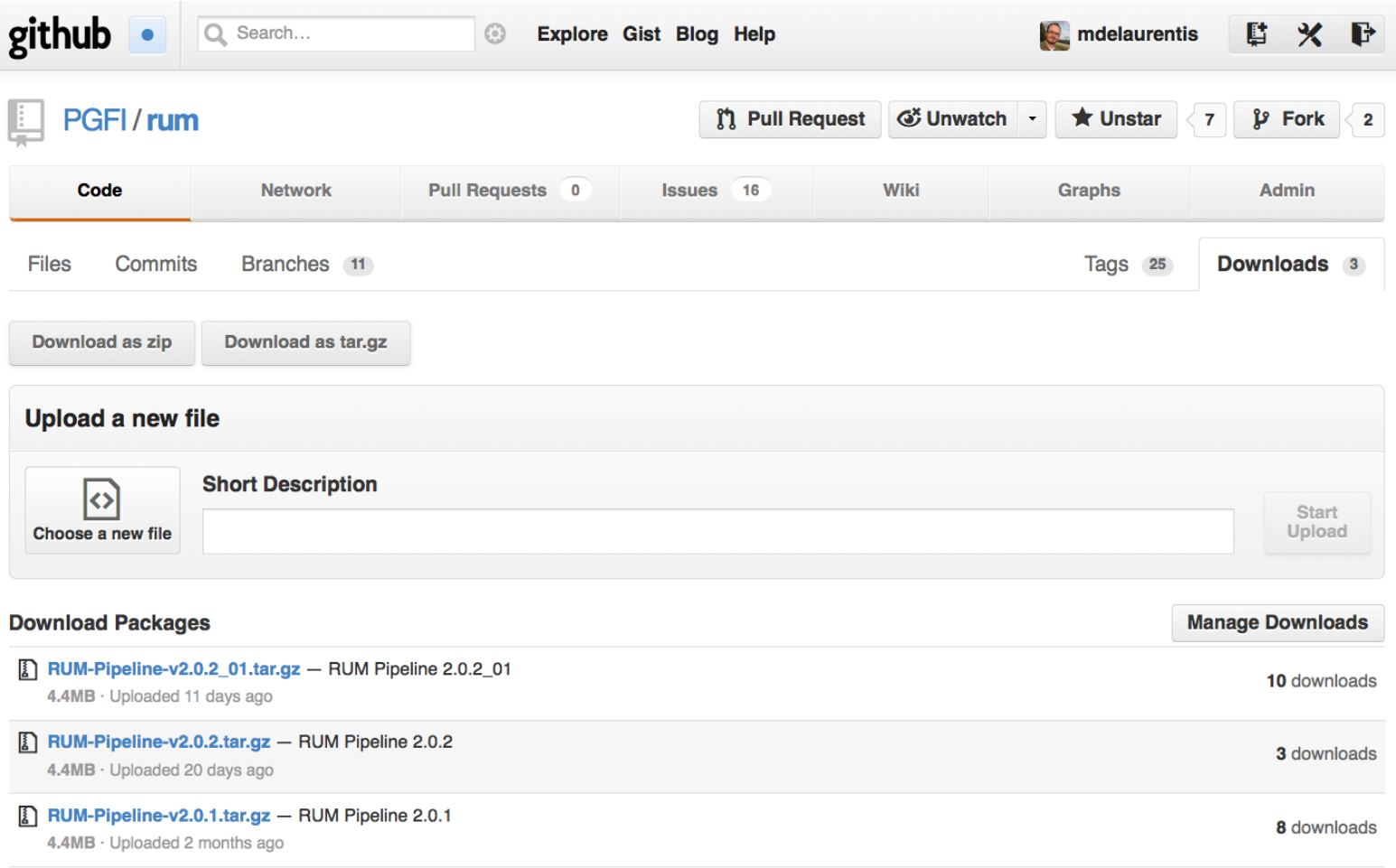
Downloads

https://github.com/PGFI/rum/downloads



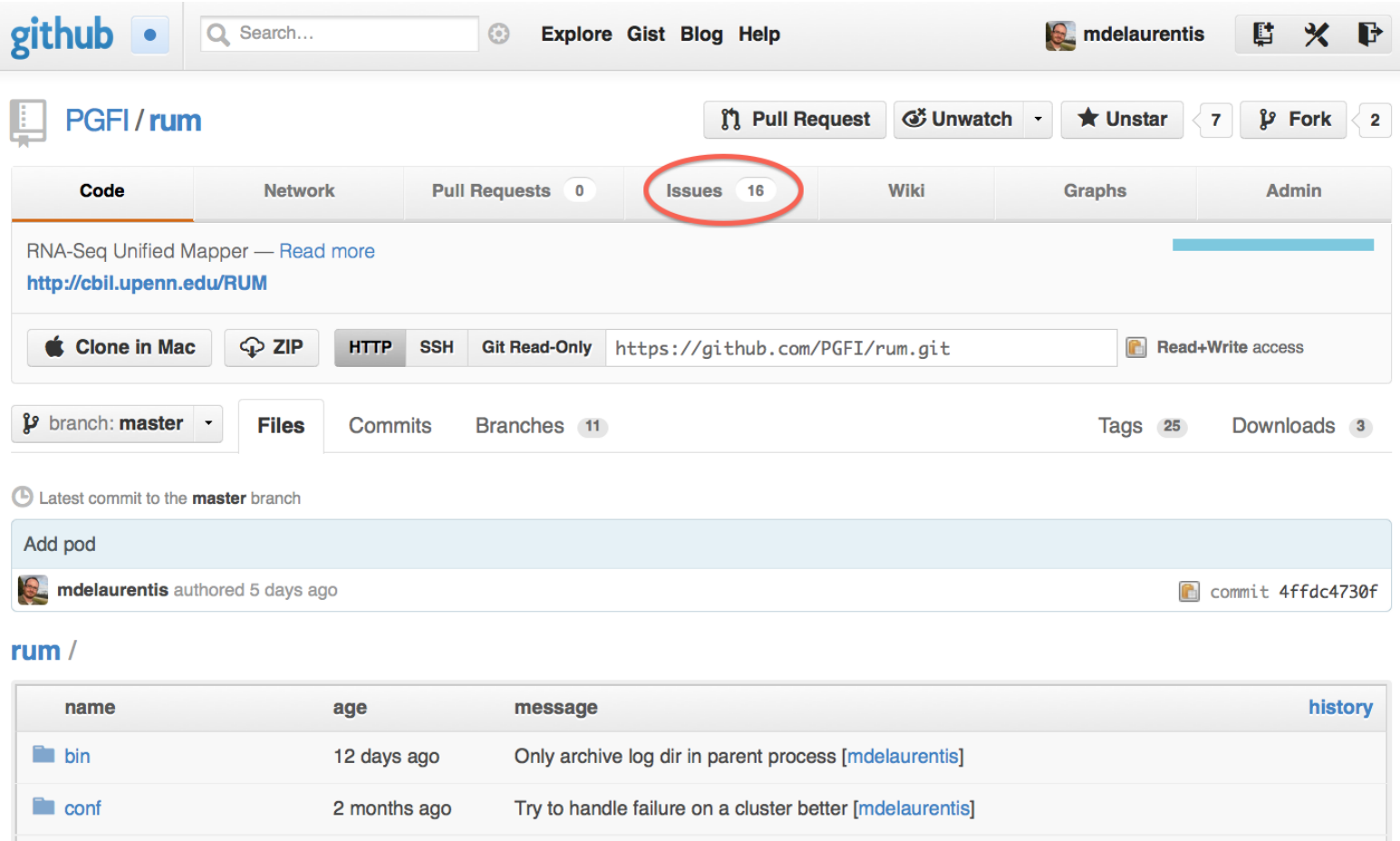
Downloads

https://github.com/PGFI/rum/downloads



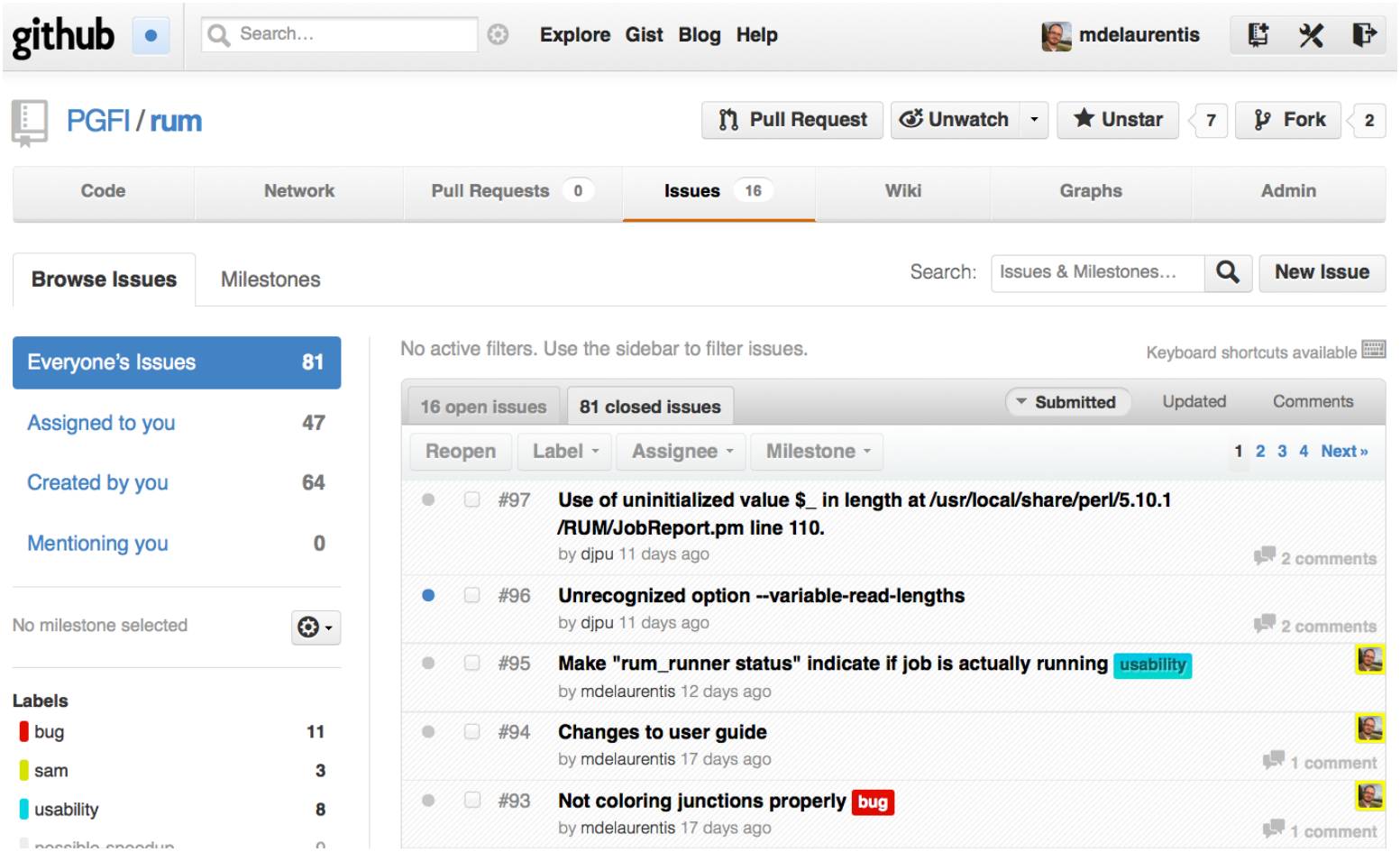
Issues

https://github.com/PGFI/rum/issues



Issues

https://github.com/PGFI/rum/issues



Agenda

Possible future enhancements

- Use original read names throughout the pipeline
- Performance improvements