seqmagick Documentation

Release 0.6.0

Matsen Group

November 17, 2014

Contents

Contents

CHAPTER 1

Changelog

1.1 0.6.0

- Map `.nex` extension to NEXUS-format (–alphabet must be specified if writing)
- Use reservoir sampling in `--sample` selector (lower memory use)
- Support specifying negative indices to `--cut` [GH-33]
- Optionally allow invalid codons in `backtrans-align` [GH-34]
- Map `.fq` extension to FASTQ format
- Optional multithreaded I/O in `info` [GH-36]
- Print sequence name on length mismatch in `backtrans-align` [GH-37]
- Support for + and − in head and tail to mimick Linux head and tail commands.
- Fix scoring for mixed-case sequences in `primer-trim`.
- Fix bug in `primer-trim` - failed when sequence had multiple 5' gaps compared to the primer.
- Clarify documentation and fix bug in convert/mogrify `--pattern-replace` [GH-39]
- Support for gzip files in `seqmagick convert --sort`

1.2 0.5.0

- Change `seqmagick extract-ids --source-format` to `--input-format` to match other commands (GH-29)
- Support gzip- and bzip2-compressed inputs and outputs for most commands (GH-30)
- Change default input format for `sff` to `sff-trim`, which respects the clipping locations embedded in each sequence record.
- Add `--details-out` option to `seqmagick quality-filter`, which writes details on each read processed.
- Match barcode/primer `seqmagick quality-filter` against a trie; allows per-specimen barcodes.
- Remove `--failure-out` option from `seqmagick quality-filter`. See `--details-out`
- Raise an error if number of codons does not match number of amino acids in `seqmagick backtrans-align`

- Add `--sample` subcommand (GH-31)

1.3 0.4.0

- Fix bug in `--squeeze`
- More informative messages in `seqmagick primer-trim`
- Added `--alphabet` flag to allow writing NEXUS (GH-23)
- Exiting without error on SIGPIPE in extract-ids, info (GH-17)
- Ambiguities are translated as 'X' in –translate (GH-16)
- Allowing '.' or '-' as gap character (GH-18)
- `--name-prefix` and `--name-suffix` no longer create a mangled description (GH-19)
- Files owned by another user can be mogrified, as long as they are group writeable (GH-14)
- Add `backtrans-align` subcommand, which maps unaligned nucleotides onto a protein alignment (GH-20)
- Allow FASTQ as input to quality-filter
- Significantly expand functionality of quality-filter: identify and trim barcodes/primers; report detailed failure information.
- Cleanup, additional tests
- Add `--drop` filter to convert and mogrify (GH-24)
- Apply current umask when creating files (GH-26)
- Support stdin in `seqmagick info` (GH-27)
- Support translating ambiguous nucleotides, if codon translation is unambiguous

1.4 0.3.1

- Fix bug in `quality-filter` MinLengthFilter
- Case consistency in seqmagick

1.5 0.3.0

- Internal reorganization - transformations are converted to partial functions, then applied.
- Argument order now affects order of tranformation application.
- Change default output format to 'align' for TTYs in seqmagick info
- Add BioPython as dependency (closes GH-7)
- Add `primer-trim` subcommand
- Add option to apply custom function(s) to sequences
- Add new filtering options: `--squeeze-threshold`, `--min-ungapped-length` `--include-from-file` `--exclude-from-file`

- Removed seqmagick muscle
- Added new subcommand `quality-filter`
- Added new subcommand `extract-ids` (closes GH-13)
- Allow use of '-' to indicate stdin / stdout (closes GH-11)
- Add mapping from .phyx to `phylip-relaxed` (targeted for BioPython 1.58)

1.6 0.2.0

- Refactoring
- Added hyphenation to multi-word command line options (e.g. `--deduplicatetaxa` -> `--deduplicate-taxa`)
- Add support for `.needle`, `.sff` formats
- Close GH-4

1.7 0.1.0

Initial release

CHAPTER 2

Motivation

We often have to convert between sequence formats and do little tasks on them, and it's not worth writing scripts for that. Seqmagick is a kickass little utility built in the spirit of imagemagick to expose the file format conversion in Biopython in a convenient way. Instead of having a big mess of scripts, there is one that takes arguments:

```
seqmagick convert a.fasta b.phy    # convert from fasta to phylip
seqmagick mogrify --ungap a.fasta  # remove all gaps from a.fasta, in place
seqmagick info *.fasta             # describe all FASTA files in the current directory
```

And more.

CHAPTER 3

Installation

First, you'll need to install BioPython. NumPy (which parts of BioPython depend on) is not required for `seqmagick` to function. Once done, install the latest release with:

```
pip install seqmagick
```

Or install the bleeding edge version:

```
pip install git+git://github.com/fhcrc/seqmagick.git@master#egg-info=seqmagick
```

CHAPTER 4

Use

Seqmagick can be used to query information about sequence files, convert between types, and modify sequence files. All functions are accessed through subcommands:

```
seqmagick <subcommand> [options] arguments
```

CHAPTER 5

List of Subcommands

5.1 `convert` and `mogrify`

Convert and mogrify achieve similar goals. `convert` performs some operation on a file (from changing format to something more complicated) and writes to a new file. `mogrify` modifies a file in place, and would not normally be used to convert formats.

The two have similar signatures:

```
seqmagick convert [options] infile outfile
```

vs:

```
seqmagick mogrify [options] infile
```

Options are shared between convert and mogrify.

5.1.1 Examples

Basic Conversion

`convert` can be used to convert between any file types BioPython supports (which is many). For a full list of supported types, see the BioPython SeqIO wiki page.

By default, file type is inferred from file extension, so:

```
seqmagick convert a.fasta a.sto
```

converts an existing file `a.fasta` from FASTA to Stockholm format. **Neat!** But there's more.

Sequence Modification

A wealth of options await you when you're ready to do something slightly more complicated with your sequences.

Let's say I just want a few of my sequences:

```
$ seqmagick convert --head 5 examples/test.fasta examples/test.head.fasta
$ seqmagick info examples/test*.fasta
name                      alignment  min_len  max_len  avg_len  num_seqs
examples/test.fasta       FALSE      972      9719     1573.67  15
examples/test.head.fasta  FALSE      978      990      984.00   5
```

Or I want to remove any gaps, reverse complement, select the last 5 sequences, and remove any duplicates from an alignment in place:

```
seqmagick mogrify --tail 5 --reverse-complement --ungap --deduplicate-sequences examples/test.fasta e
```

You can even define your own functions in python and use them via `--apply-function`.

Note: To maximize flexibility, most transformations passed as options to `mogrify` and `convert` are processed *in order*, so:

```
seqmagick convert --min-length 50 --cut 1:5 a.fasta b.fasta
```

will work fine, but:

```
seqmagick convert --cut 1:5 --min-length 50 a.fasta b.fasta
```

will never return records, since the cutting transformation happens before the minimum length predicate is applied.

Command-line Arguments

```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

5.2 backtrans-align

Given a protein alignment and unaligned nucleotides, align the nucleotides using the protein alignment. Protein and nucleotide sequence files must contain the same number of sequences, in the same order, with the same IDs.


```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

5.3 extract-ids

`seqmagick extract-ids` is extremely simple - all the IDs from a sequence file are printed to stdout (by default) or the file of your choosing:

```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

5.4 info

`seqmagick info` describes one or more sequence files

5.4.1 Example

```
seqmagick info examples/*.fasta
```

```
name                        alignment  min_len  max_len  avg_len  num_seqs
examples/aligned.fasta      TRUE       9797     9797     9797.00  15
examples/dewrapped.fasta    TRUE       240      240      240.00   148
examples/range.fasta        TRUE       119      119      119.00   2
examples/test.fasta         FALSE      972      9719     1573.67  15
examples/wrapped.fasta      FALSE      120      237      178.50   2
```

Output can be in comma-separated, tab-separated, or aligned formats. See `seqmagick info -h` for details.

Usage:

```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

5.5 quality-filter

`quality-filter` truncates and removes sequences that don't match a set of quality criteria. The subcommand takes a FASTA and quality score file, and writes the results to an output file:

```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

5.6 primer-trim

`primer-trim` trims an alignment to a region defined by a set of forward and reverse primers. Usage is as follows:

```
Traceback (most recent call last):
  File "../seqmagick.py", line 7, in <module>
    sys.exit(cli.main(sys.argv[1:]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 12, in main
    action, arguments = parse_arguments(argv)
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/scripts/cli.py", line 58, in parse
    for name, mod in subcommands.itermodules():
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/__init__.py", line 7,
    __import__('%s.%s' % (root, command), fromlist=[command]))
  File "/var/build/user_builds/seqmagick/checkouts/0.6.0/seqmagick/subcommands/convert.py", line 8, i
    from Bio import Alphabet, SeqIO
ImportError: No module named Bio
```

CHAPTER 6

Supported File Extensions

By default, `seqmagick` infers the file type from extension. Currently mapped extensions are:

Extension	Format
.afa	fasta
.aln	clustal
.fa	fasta
.faa	fasta
.fas	fasta
.fasta	fasta
.fastq	fastq
.ffn	fasta
.fna	fasta
.fq	fastq
.frn	fasta
.gb	genbank
.gbk	genbank
.needle	emboss
.nex	nexus
.phy	phylip
.phylip	phylip
.phyx	phylip-relaxed
.qual	qual
.sff	sff-trim
.sth	stockholm
.sto	stockholm

Note: NEXUS-format output requires the `--alphabet` flag.

6.1 Default Format

When reading from stdin or writing to stdout, `seqmagick` defaults to fasta format. This behavior may be overridden with the `--input-format` and `--output-format` flags.

If an extension is not listed, you can either rename the file to a supported extension, or specify it manually via `--input-format` or `--output-format`.

6.2 Compressed file support

most commands support gzip (files ending in `.gz`) and bzip (files ending in `.bz2` or `.bz`) compressed inputs and outputs. File types for these files are inferred using the extension of the file after stripping the file extension indicating that the file is compressed, so `input.fasta.gz` would be inferred to be in FASTA format.

CHAPTER 7

Acknowledgements

seqmagick is written and maintained by the Matsen Group at the Fred Hutchinson Cancer Research Center.

CHAPTER 8

Contributing

We welcome contributions! Simply fork the repository on GitHub and send a pull request.