

Length filter: Keeps FASTA records whose sequence length lies within [minLen, maxLen]

Jens Allmer [ORCID](#)

unreviewed

Jens Allmer (2026). Length filter: Keeps FASTA records whose sequence length lies within [minLen, maxLen]. Version 1. Internet of Science. local:article/component-fasta-len-filter. <https://ioscience.org/a/component-fasta-len-filter/v1/>

Produced in the Internet of Science; reproducible: nothing embedded to re-run.

A component publication of the Internet of Science: FASTALenFilter 0.2.0, an atom. Generated from the manifest and the class comment on 2026-10-26; the author completes it and saves new versions. Lines marked FILL IN are the author's.

1 What it does and why

Keeps the FASTA records whose sequence length lies within [minLen, maxLen] (inclusive).

FILL IN: why this component exists — the problem it solves and what the IoS lacked without it.

2 Algorithm

FILL IN: the algorithm reference (a citation tag from docs/vision/README.md or a paper); the class comment names none yet.

3 Inputs and outputs

- input in: FASTA.Entry — One FASTA record: { header: string (with leading ">"), seq: string (no whitespace) }.
- output out: FASTA.Entry — One FASTA record: { header: string (with leading ">"), seq: string (no whitespace) }.

4 Parameters

- minLen (integer, default 1, min 1): Minimum length
- maxLen (integer, default -1, min 1, -1 = unlimited): Maximum length (-1: unlimited)

5 How it is tested

- tests/unit/apps/graph-to-workflow.test.js
- tests/unit/apps/presentation-publications.test.js

- tests/unit/apps/suggest.test.js
- tests/unit/apps/wfms-server.test.js
- tests/unit/atomic/fasta/FASTALenFilter.test.js
- tests/unit/bin/ios-batch.test.js
- tests/unit/bin/ios-bundle.test.js
- tests/unit/bin/ios-new.test.js
- tests/unit/bin/ios-publish-component.test.js
- tests/unit/runtime/pipeline.test.js
- tests/unit/runtime/qc.test.js
- tests/unit/runtime/registry-roots.test.js
- tests/unit/runtime/registry-workflows.test.js
- tests/unit/runtime/registry.test.js
- tests/unit/runtime/workflow-failure.test.js
- tests/unit/runtime/workflow-subworkflows.test.js
- tests/unit/runtime/workflow.test.js
- tests/unit/runtime/WorkflowStage.test.js
- tests/unit/tools/build-browser.test.js
- tests/unit/tools/bundler.test.js
- the conformance suite (tests/conformance/components.test.js, tests/conformance/credits.test.js, tests/conformance/dead-ends.test.js, tests/conformance/maturity.test.js, tests/conformance/tasks.test.js): the component contract for every registered component

FILL IN: the positive cases (synthetic data of known truth), the negative controls and the experimental data the tests use.

6 Maturity

Maturity 1 of 3 (ADR 0012); a reviewer raises it, the credits earn it. Maturity 2 needs a test or test-validation credit by a person who is not an author: independent tests (CONTRIBUTING rules 17, 21); from then on this publication must exist (rule components/publication).

7 Credits

- Software: Jens Allmer (implementation)

8 How to use

```
node bin/ios.js describe FASTALenFilter
node bin/ios.js run FASTALenFilter < input > output
```

Other parameters as `--<name> <value>; node bin/ios.js describe FASTALenFilter` lists them with their defaults.

As a node of a workflow document (`*.workflow.json`, ADR 0011):

```
{ "id": "filter", "type": "FASTALenFilter", "params": { "minLen": 1, "maxLen": 1000 }
```

Example workflow: `src/workflows/examples/fasta-length-filter.workflow.json`.

9 Results

FILL IN: run the example workflow (`node bin/ios.js run-workflow src/workflows/examples/fasta-length-filter.workflow.json --artefacts <dir>`) on a registered dataset, open the result in the builder and drag the figure or table into this article (Manuscript pane); write one paragraph on what it shows.

Contributions

Authors are the contributors whose shares, largest first, reach 80 % (ADR 0029): Jens Allmer 100 % (writing 492 of 492 words).

Writing – original draft Jens Allmer