# SEISMIC-RNA Developer Manual

Version 0.22.3
2024-12-30

# Contents

# List of Figures

# Chapter 1

# Releases

This chapter explains how to release a new version of SEISMIC-RNA.

## 1.1 Versioning

### 1.1.1 Why and when to release a new version

Release a new version when you have made a significant change to the software that you want to share with other users. Do not release a new version if any unit tests failed on the most recent commit, which is indicated by either a green checkmark (all successful) or red X (one or more failures) beside the name of the most recent commit on the main page of the repository (`https://github.com/rouskinlab/seismic-rna`). You can also navigate to the "Actions" tab and check that all workflow runs with the name of the most recent commit have a green checkmark (not a red X) next to them (there will be more than one run per commit, so make sure to check ALL runs for the most recent commit).

### 1.1.2 How to number a new version

SEISMIC-RNA uses semantic versioning in which each version comprises three non-negative integers separated by periods and an optional pre-release tag:

**Major version** Increase when the new version introduces a major change that fundamentally changes how SEISMIC-RNA works and/or is mostly incompatible with previous versions. These types of changes should happen rarely, less than once annually. A major version of 0 indicates that the version is a pre-release and not yet considered stable, so that increases in the minor version may also indicate major backwards-incompatible changes.

**Minor version** Increase when the new version introduces new features, removes minor features, and/or makes minor backwards-incompatible changes for which there are workarounds. Ideally, any backwards-incompatible changes will be forewarned by deprecation notices. Reset the minor version to 0 whenever the major version increases.

**Patch** Increase when the new version makes a small improvement (such as fixing a bug or optimizing an algorithm) that causes little to no change to the user interface and would not be noticed by users who had not encountered the issue that this release fixes. Reset the patch to 0 whenever the minor version increases.

**Pre-release tag** Append to the version number when the current version is a pre-release (i.e. not considered stable or definitive). This tag can contain any letters and numbers (but cannot start with a number) and should briefly describe the nature of the pre-release, such as "alpha", "beta", or "dev1". Do not include a pre-release tag on a production version of the code.

### 1.1.3 Where to define the version number

The version number is defined in `src/seismicrna/core/version.py` as `__version__`.

## 1.2 Releasing on each platform

SEISMIC-RNA is released on three platforms:

- GitHub, as source code
- Python Package Index (PyPI), for pip install
- Bioconda, for conda install

### 1.2.1 Releasing on GitHub

#### The first time you release on GitHub

1. Create an account on GitHub and obtain permission to modify the SEISMIC-RNA repository (`https://github.com/rouskinlab/seismic-rna`).

#### Every time you release on GitHub

1. Make sure you have committed and pushed everything you want to release to the GitHub repository; importantly, make sure that the version number defined in the source code (`src/seismicrna/core/version.py`) is up-to-date.

2. Make sure the test suite ran successfully. The test suite runs automatically each time updates to the source code are pushed to the main branch. On GitHub, under the "Actions" tab, click on "Install SEISMIC-RNA and run its test suite" in the sidebar and make sure that this workflow ran after the most recent modification to the code and that it completed successfully (green checkmark). If any errors occurred, fix them before releasing.

3. Open the repository in a web browser, find the "Releases" panel on the right sidebar, and click "Create a new release".

4. On the new page that opens, you will need to write information about the release.

5. Select "Choose a tag" and in the box labeled "Find or create a new tag" type the version number of the new release prefaced by a "v", e.g. "v2.7.18". Make sure the new tag is exactly a "v" followed by the version number, or else releasing on Conda will not work.

6. Give this release a "Release title" of up to 10 words that succinctly describes the main point of this release.

7. Write a more detailed description in "Describe this release" that begins with "What's new in x.y.z" where "x.y.z" is the version. Describe new features, removed features, bug fixes, changes to the user interface, etc. in bullet-point format, ideally with sub-headings for each category of changes.

8. Move the cursor to the bottom of the text box and click "Generate release notes" to add a summary of all commits, pull requests, etc. associated with this release.

9. If this is a pre-release, check the box "Set as a pre-release" at the bottom.

10. Click "Publish release" when you are finished.

11. Publishing the release will trigger the GitHub Actions script `build-publish.yml`, which will automatically release the code on PyPI. See below for details.

### 1.2.2 Releasing on PyPI

The SEISMIC-RNA repository on GitHub includes a GitHub Actions script `build-publish.yml` that builds and publishes a new version of SEISMIC-RNA on PyPI every time a new release is created on GitHub. This script exists to ensure that releases on GitHub are synced with those on PyPI, and to reduce the amount of manual effort (and potential for error) required to release a new version. The latter aim became more important when SEISMIC-RNA began including C extension modules, which require compilation and thus a separate wheel file for each platform and version of Python.

#### The first time you release on PyPI

1. On the GitHub repository for SEISMIC-RNA (`https://github.com/rouskinlab/seismic-rna`), navigate to `.github/workflows`.

2. Confirm that there is a workflow script called `build-publish.yml`, which runs every time a new release is created on GitHub (or when run manually) and executes all steps needed to publish on PyPI.

3. Create an account on PyPI and obtain permission to modify the SEISMIC-RNA project (`https://pypi.org/project/seismic-rna`).

4. Log into your account and open SEISMIC-RNA in PyPI.

5. Click on "Manage project" and then on "Publishing".

6. SEISMIC-RNA should be set up to use Trusted Publisher Management (for details, see `https://docs.pypi.org/trusted-publishers`). Under "Manage current publishers" check that GitHub is listed as a publisher with the Repository set to `rouskinlab/seismic-rna`, the Workflow set to `build-publish.yml`, and the Environment name set to `publish-pypi`. If not, add a new publisher with this information.

**Every time you release on PyPI**

1. Release a new version on GitHub following the steps above. Make sure that all tests pass by checking that the workflow "Install SEISMIC-RNA and run its test suite" completes with no errors (green checkmark).

2. Once a new version is released on GitHub, the workflow "Build and Publish Project on PyPI" (in the script `build-publish.yml`) should run automatically. This workflow builds the source distribution and builds wheels for Linux and macOS.

3. If any build fails (red X), then fix the errors and re-run this workflow manually by clicking "Re-run all jobs".

4. If all builds succeed, then (as a security measure) the final release must be approved by one of the repository owners (currently `matthewfallan` or `justinaruda`). The repository owners should approve the release only after confirming that all tests have passed and that the code appears functional.

5. Upon approval, the release will be published on PyPI using Trusted Publisher Management. The new version can then be installed using pip.

### 1.2.3 Releasing on Bioconda

These steps are adapted from the Bioconda contribution instructions: `https://bioconda.github.io/contributor/index.html`. Refer to these instructions (and their links) if you need to troubleshoot this workflow.

**The first time you release on Bioconda**

1. Create an account on GitHub.

2. Fork the Bioconda Recipes repository by navigating to `https://github.com/bioconda/bioconda-recipes` and clicking "Create a new fork" in the drop-down menu underneath "Fork" in the upper right panel.

3. Clone your fork of the repository (so you can modify it on your computer) by typing `git clone https://github.com/<USERNAME>/bioconda-recipes.git`, while replacing `<USERNAME>` with the username you used to fork the repository.

4. Create a new Conda environment and install the Bioconda developer tools by typing `conda create -n bioconda -c conda-forge -c bioconda bioconda-utils`.

**Every time you release on Bioconda**

1. Release this version of the software on GitHub (see above). Make sure to give it the tag "v" followed by the version number, otherwise building the metadata file for Conda will fail.

2. Navigate to the directory into which you cloned the Bioconda Recipes repository and bring it up-to-date with the `master` branch by typing

```
git checkout master
git pull upstream master
git push origin master
```

3. Create a new branch for the new version, e.g. called `seismic-rna_v2.7.18`, by typing `git checkout -b seismic-rna_v2.7.18`.

4. Create the Conda recipe files by navigating to the main directory `seismic-rna` and typing `python make_conda_recipe.py`. This will create or update the Conda recipe files `build.sh` and `meta.yaml` in the directory `seismic-rna/conda`. It will also copy them automatically to `recipes/seismic-rna` (in your Bioconda Recipes repository).

5. Navigate back to your Bioconda Recipes repository and confirm that the latest versions of `build.sh` and `meta.yaml` were copied into it.

6. Activate your `bioconda` environment by typing `conda activate bioconda`.

7. Lint your updates by typing `bioconda-utils lint --packages seismic-rna` and confirm that it prints an OK message, not "Errors were found".

8. Test building the package typing `bioconda-utils build --packages seismic-rna` and confirm that it prints an OK message, not "Errors were found".

9. Once both linting and building succeed, push the updates to your fork of Bioconda Reipces by typing

```
git add recipes/seismic-rna/*
git commit -m "Update seismic-rna"
git push
```

10. Bioconda is set up to detect version updates and create pull requests automatically. To check if a pull request was created for you, navigate to the Bioconda Recipes GitHub repository Pull Requests tab (`https://github.com/bioconda/bioconda-recipes/pulls`) and check if one of the most recent pull requests matches the name of your commit.

11. If not, then create a pull request manually by navigating to your fork of Bioconda Recipes on GitHub. If you see a message near the top that says a recent branch had changes, click "Compare and Pull Request"; otherwise, go to the Pull Requests tab and click "New pull request" in the upper right corner to create your pull request.

12. Navigate back to the Bioconda Recipes GitHub repository Pull Requests tab (`https://github.com/bioconda/bioconda-recipes/pulls`) and confirm that your pull request has been opened. Monitor the status badge beside the commit message and wait until turns from an orange circle (pending) into a green checkmark (success) or red X (failure).

13. If it fails, then check the failure messages and revise the `build.sh` and `meta.yaml` files. If you need help, then you can ask the Bioconda maintainers.

14. Once it succeeds, then the Bioconda maintainers should eventually (within several days) merge your pull request, and the new version will be ready to install with the command `conda install -c bioconda seismic-rna`. Before it is merged, check back periodically for any messages from the Bioconda maintainers.

15. Delete your local branch by typing `git branch -d seismic-rna_v2.7.18` and also delete the branch on your fork of the Bioconda Recipes GitHub repository, as they are no longer needed.