

## Reproducibility Review

Below is a seven point reproducibility review prescribed by [Improving reproducibility and reusability in the Journal of Cheminformatics](#) of the `main` branch of repository <https://github.com/darrenjw/jax-smfsb> (commit [71a139c1](#)), accessed on 2024-10-25.

### 1. Does the repository contain a LICENSE file in its root?

Yes, Apache-2.0.

### 2. Does the repository contain a README file in its root?

Yes.

### 3. Does the repository contain an associated public issue tracker?

Yes.

### 4. Has the repository been externally archived on Zenodo, FigShare, or equivalent that is referenced in the README?

No, this repository has a README, but it does not reference Zenodo. The GitHub-Zenodo integration can be set up by following [this tutorial](#).

If your Zenodo record is XYZ, then you can use the following in your README:

`[! [DOI] (https://zenodo.org/badge/DOI/10.5281/zenodo.XYZ.svg)] (https://doi.org/10.5281/zenodo.XYZ)`

### 5. Does the README contain installation documentation?

No, this repository has a markdown README, but it does not contain a section header entitled `# Installation` (it's allowed to be any level deep). Please add a section that includes information on how the user should get the code (e.g., clone it from GitHub) and install it locally. This might read like:

```
git clone https://github.com/darrenjw/jax-smfsb
cd jax-smfsb
pip install --editable .
```

Alternatively, you can deploy your code to the [Python Package Index \(PyPI\)](#) and document how it can be installed with `pip install`. This might read like:

```
pip install jax_smfsb
```

## 6. Is the code from the repository installable in a straightforward manner?

Yes.

### Packaging Metadata

`pyroma` rating: 9/10

1. The classifiers should specify what minor versions of Python you support as well as what major version.
2. Your package does not have keywords data.
3. Specifying a development status in the classifiers gives users a hint of how stable your software is.

These results can be regenerated locally using the following shell commands:

```
git clone https://github.com/darrenjw/jax-smfsb
cd jax-smfsb
python -m pip install pyroma
pyroma .
```

## 7. Does the code conform to an external linter and formatter (e.g., `ruff` for Python)?

### Formatting

The repository does not conform to an external formatter. This is important because there is a large cognitive burden for reading code that does not conform to community standards. Formatters take care of styling code to reduce burden on readers, therefore better communicating your work to readers.

For example, `black` can be applied to auto-format Python code with the following:

```
git clone https://github.com/darrenjw/jax-smfsb
cd jax-smfsb
python -m pip install black
black .
git commit -m "Blacken code"
git push
```

### Linting

The repository does not pass linting. This means that there are obvious, and usually easy to fix issues that can be automatically detected. You can run the linter yourself with:

```
git clone https://github.com/darrenjw/jax-smfsb
cd jax-smfsb
```

```
python -m pip install ruff
ruff check .
```

## Summary

Scientific integrity depends on enabling others to understand the methodology (written as computer code) and reproduce the results generated from it. This reproducibility review reflects steps towards this goal that may be new for some researchers, but will ultimately raise standards across our community and lead to better science.

Because the repository does not pass all seven criteria of the reproducibility review, I recommend rejecting the associated article and inviting later resubmission after the criteria have all been satisfied.

## Colophon

This review was automatically generated with **autoreviewer** v0.0.6-dev-fdeabe2c with the following commands:

```
python -m pip install autoreviewer
python -m autoreviewer darrenjw/jax-smfsb
```

Please leave any feedback about the completeness and/or correctness of this review on the issue tracker for [cthoht/autoreviewer](#).