Title: emcee: The MCMC Hammer

Name: Pierce Hanlon

Date: 5/14/2025

1. Name of the package, and what is the basic aim of what the package does or solve?

The name of the package is emcee. It aims to provide an intuitive, robust, and thoroughly tested Python implementation of the Affine Invariant Markov chain Monte Carlo, or MCMC, Ensemble sampler. In other words it helps explore probability distributions and estimate uncertainties on model parameters given by your data, as well as handle complex high-dimensional parameter spaces with more efficiency than previous MCMC methods.

2. Why/how did you select this package?

I selected this package because it is very popular and used frequently, meaning I am more likely to come across it later in my career/life and with greater frequency than other packages, so having a better understanding of it as a baseline will be helpful.

3. How old is the package? does it have a genealogy, i.e. what related codes came before or after. Are there other codes that solve the same problem?

emcee originally released on February 15, 2012, making it around 13 years old. The Affine Invariant Ensemble Sampler algorithm was developed by Goodman & Weare in 2010, but emcee was more widely adopted and intuitive for users. MCMC packages that followed emcee like zeus and more recent editions of TensorFlow Probability and PyMC3 that include ensemble samplers can be viewed as responses or descendents to the ideas popularized by the emcee package. I installed version 3.1.6.

4. Is it still maintained, and by the original author(s)? Are there instructions on how to contribute to this project?

The emcee package is still actively maintained, and one of the most active contributors and maintainers of the package is the original author, Daniel Foreman-Mackey, as can be seen in the official emcee GitHub repository: https://github.com/dfm/emcee. There are instructions on how to contribute, also in the GitHub repository.

5. Evaluate how easy it was to install and use. What commands did you use to install?

It was very easy to install. I opened my terminal and ran “pip install emcee.”

6. Does it install via the "standard" pip/conda, or is it more complex?

The package installs via the “standard” pip.

7. Is the source code available? For example, "pip install galpy" may get it to you, but where can you inspect the code?

The source code is available on the GitHub repository: https://github.com/dfm/emcee.

8. Is the code used by other packages (if so, give one or two examples)

Yes, it is used by other packages within the scientific Python ecosystem that need the robust MCMC sampling emcee offers. Examples include corner (https://ascl.net/1702.002), a widely used Python package for making corner/triangle plots, and Astropy (https://ascl.net/1304.002), the core package for a lot of astronomy work within Python.

9. Give an example how you use the code. Is it commandline, or jupyter notebook, or a web interface?

Example has been submitted. It is a jupyter notebook. In the example, I used emcee for a linear regression problem. I used artificial data without specific units for simplicity. First, I imported the libraries. Then, I defined the probability model using log_likelihood(theta, x, y, yerr), log_prior(theta), and log_probability(theta, x, y, yerr). Then I set up the MCMC sampler with ndim, nwalkers, p0, and emcee.EnsembleSampler(). Next, I run the MCMC sampler for an initial number of steps defined as n_burn and then reset the sampler. For the production run I run the sampler for the main sampling phase defined by n_steps to collect the posterior samples. I used the progress=True argument so a progress bar would be displayed. Finally, I analyze the results using sampler.get_chain(flat=True) which returns all of the samples from all walkers as one flattened array, and I get summary statistics, specifically the median, the 16th, and the 84th percentiles of the posterior samples for every parameter to receive an estimate of the parameter values and their respective uncertainties. I made trace plots to visualize how each parameter "evolves" over the MCMC steps for each walker, which helps show if the chains converge. I installed corner as well to make a corner plot that shows the 1-D marginalized posterior distributions for each parameter, as well as the 2-D joint posterior distributions for every pair of parameters, which visualizes uncertainties and correlations. I also plotted the original data with the true underlying line for comparison, multiple samples from the posterior distribution, and error bars to show the uncertainty.

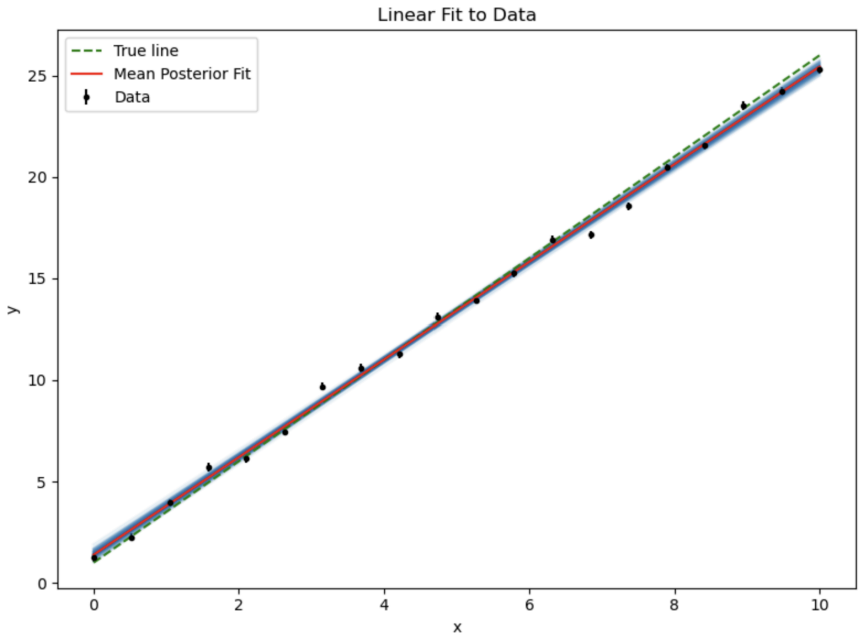
10. Provide examples using the code. if you prefer to use a jupyter notebook instead of a python script, that's ok.

I used a jupyter notebook, which has been submitted.

11. Does the package produce figures, or are you on your own? Is matplotlib used?

The emcee package does not produce figures. In the example I made, I used both matplotlib and corner.

12. Your code and report should show at least one figure, and create a nice figure caption explaining what it shows. Your notebook should show how the figure was made (reproducible).



As stated in my description of the code example, I plotted the original data with the true underlying line for comparison, multiple samples from the posterior distribution, and error bars to show the uncertainty. This is one of 3 different groups of plots I made, the others being the posterior distributions and the trace plots.

13. Is the package pure python? or does it need accompanying C/C++/Fortran code?

The emcee package is primarily pure Python, and it does not generally need accompanying C/C++/Fortran code for most use cases.

14. What is the input to the package? Just parameters, or dataset(s), or can they be generated from scratch?

The main inputs are generally what I discussed in my example. You need the log-probability function that takes a set of parameters as its first argument, while subsequent arguments are necessary data. You also need to provide initial positions of the walkers, the number of walkers, and the number of dimensions/parameters. The package does not generate datasets from scratch, but like in my example, you can make artificial datasets using NumPy prior to giving those data sets to emcee.

15. What is the output of the package? Just parameters, or dataset(s)?

The main output of the emcee package is a set of samples (parameters) taken from the posterior probability distribution of a model's parameters organized into "chains," with there being one per walker. Usually this is accessed through the sampler object after the MCMC has been run. The package does not really modify input data or output new data.

16. Does the code provide any unit tests, regression or benchmarking?

The unit tests and integration tests used during development and maintenance can be found in the emcee GitHub repository, both of which I believe can serve as a sort of regression test, although not necessarily. From what I can find there is no benchmarking in the documentation.

17. How can you feel confident the code produces a reliable result? (see also previous question)

I feel confident the code produces reliable results based on the fact that there are publically available tests confirming its functionality and it is actively being maintained and developed. Furthermore, the fact that it is open-source and very popular and thus is subject to all necessary scrutiny from the scientific community that uses it, so flaws and bugs are more likely to be found, reported, and addressed.

18. What main python package(s) does it use or depend on (e.g. numpy, curve_fit, solve_ivp) - how did you find this out?

The emcee package heavily depends on NumPy. Throughout the source code of emcee on the GitHub repository, NumPy is imported multiple times throughout the codebase, and the documentation frequently references NumPy arrays and concepts. Furthermore, when you install emcee using pip, pip automatically installs NumPy as a prerequisite if you don't already have it installed since pip handles dependencies on its own.

19. What kind of documentation does the package provide? Was it sufficient for you?

The emcee package has an official documentation website (https://emcee.readthedocs.io/en/stable/) that includes a thorough user guide, tutorials, and a changelog. There are also multiline docstrings within the code itself that explain what different classes, modules, functions, etc. do and what arguments they require. The emcee GitHub, which I have referenced multiple times already, has plenty of specific information. There is also their original published paper which you can view as a PDF. It was more than sufficient.

20. If you use this code in a paper, do they give a preferred citation method?

If you find emcee helpful for research and use it in a paper, they say to cite Foreman-Mackey, Hogg, Lang & Goodman (2012) and link it (https://arxiv.org/abs/1202.3665) on the GitHub page. They also provide the BibTeX entry for the paper.

21. Provide any other references you used in your report.

Official emcee Documentation Website: https://emcee.readthedocs.io/en/stable/

emcee GitHub Repository: https://github.com/dfm/emcee

Original Publication: https://arxiv.org/abs/1202.3665

PyPI page for emcee: https://pypi.org/project/emcee/

22. Can you find two other papers that used this package? E.g. use ADS citations for ASCL based code.

The Milky Way's circular velocity curve between 4 and 14 kpc from APOGEE data

(https://ui.adsabs.harvard.edu/abs/2012ApJ...759..131B/abstract)

Binary Companions of Evolved Stars in APOGEE DR14: Search Method and Catalog of ~5000 Companions

(https://ui.adsabs.harvard.edu/abs/2018AJ....156...18P/abstract)

23. Did you have to learn new python methods to use this package? Or was the class good enough to get you through this project.

I didn't really have to learn new Python methods to use emcee, I mainly had to re-familiarize myself with Markov chains and other related math concepts to understand what emcee was actually doing and otherwise applied what I learned in this class to the specifics of both emcee and corner. I stuck with the style I broadly prefer.

24. Final Disclaimer: you need to state if you have prior experience in using the package or the data, or this is all new to you. In addition, if you collaborated in a group, as long as this is your work.

I do not have prior experience using emcee, I tried to be as thorough with my understanding as possible since I imagine I will encounter it frequently in the future.