Short Narrative Bio: Matt Haberland is an Assistant Professor in the BioResource and Agricultural Engineering Department at Cal Poly, San Luis Obispo. He earned his Ph.D. in Mechanical Engineering at MIT in 2014 for his thesis "Extracting Principles from Biology for Application to Running Robots", and previously, he created the Contact Sensor / Stabilizer for the rock drill of the Mars rover Curiosity. Dr. Haberland has received several honors for teaching, including the "Distinguished Teaching Award" from the UCLA Math department in 2017, and he has published research in academic journals including IEEE Robotics and Automation Letters, Bioinspiration and Biomimetics, Robotica, and Nature Methods.

1. Proposal Title: SciPy: Fundamental Tools for Biomedical Research

2. Did you previously apply for funding for this or a related proposal under the CZI EOSS program?

No.

SciPy has applied for several CZI EOSS opportunities, but the proposals were not closely related.

EOSS-0000000432 was a successful proposal for improving SciPy's statistics capabilities.
EOSS3-0000000269 unsuccessfully proposed the addition of sparse linear algebra solvers and optimizers.
EOSS4-0000000460 unsuccessfully proposed the addition of sparse linear algebra solvers.
EOSS-DI-0000000031 was a successful proposal for advancing inclusive culture in the scientific Python ecosystem.

4. Proposal Purpose

Limit to one sentence (maximum of 255 characters, including spaces)

To better serve biomedical applications, SciPy will add important new features, perform essential maintenance, and disseminate the work to biomedical researchers and software developers.

5. Amount Requested - Enter total budget amount requested in USD, including indirect costs; this number should be between $100k and $400k total costs over a two-year period.

Year One: $195,717
Year Two: $199,589
Total All Years: $395,306

6. Proposal Summary/Scope of Work: A short summary of the application (maximum of 500 words)

This proposal has four goals: 1a) improvement of SciPy functions used by biomedical software tools, 1b) enhancement of SciPy functionality used directly by biomedical researchers, 2) SciPy general maintenance, and 3) dissemination of results.

1a) To determine what improvements would be most valuable to biomedical packages, we searched the source code and issue trackers of CZI-supported projects for uses of SciPy. In addition to our study of project source code, we surveyed biomedical Python project maintainers to better understand what is needed from SciPy.

1b) To determine what new features are needed by biomedical researchers, we reviewed all open access articles in the March issues of Nature BioMedical Engineering [1], Nature Biotechnology [2], and the New England Journal of Medicine [3] to characterize the types of statistical analysis being used in modern research. We also searched the 10,000+ citations of SciPy's 2020 *Nature Methods* article [4] for direct uses of SciPy in biomedicine [5], and we surveyed the corresponding authors of SciPy-citing papers to better understand their computing needs, their current uses of SciPy, and improvements they expect from SciPy.

In this research, we found that `scipy.stats` is the subpackage most used by biomedical researchers. Accordingly, the work plan describes several new statistics features to support their work, an overhaul of the statistical distribution infrastructure, the addition of an infrastructure for hypothesis testing, and a more consistent interface for statistics functions to improve usability. The project will also support a variety of improvements in other subpackages as listed in [6].

2) We will perform essential maintenance throughout the SciPy codebase, fixing bugs so that dependent projects and researchers can use SciPy with confidence, and making other enhancements under the guidance of the aforementioned survey results.

3) As we perform the work, we will disseminate the results to biomedical researchers by a) adding to the SciPy documentation biomedical examples and tutorials; b) presenting the improvements at conferences and CZI meetings; c) hosting "office hours" during which researchers can get live help from SciPy maintainers; and d) direct communication with dependent-project maintainers and biomedical researchers.

7. *Work Plan: A description of the proposed work for which funding is being requested, including resources the applicants will provide that are not part of the requested funding. For software development-related work (e.g., engineering, product design, user research), specify how the work fits into the existing software project roadmap. For community outreach related activities (e.g., sprints, training), specify how these activities will be organized, the target audience, and expected outcomes (maximum of 750 words)*

To achieve the goals above, we request support for three work components: 1) improvements to SciPy to support biomedical research, 2) general maintenance, and 3) dissemination of results.

1) Based on our research and surveys, the most commonly-used SciPy subpackage for biomedical research is `scipy.stats`. Building on the success of our previous CZI grant (EOSS-0000000432), we propose the following, all of which are consistent with the project roadmap [7] [8].

- New features for biomedical researchers. Specifically, we will add features for
    - false discovery rate controlling procedures (Benjamini–Hochberg, Benjamini–Yekutieli),
    - commonly used post hoc tests (e.g. Dunnett's test, Student–Newman–Keuls), and
    - fundamental survival analysis tools (Kaplan–Meier estimator, log-rank test, and hazard functions for all statistical distributions).
- Overhaul of the univariate distribution infrastructure and test suite. This is the most important statistics item on the SciPy roadmap, fixing several deeply-rooted bugs and enabling major enhancements that have been desired for the past decade. It is also the most challenging and time-consuming part of the proposal, as it may require multiple revisions to address all user needs, hence the request to fund this major undertaking.
- Consistent treatment of multidimensional input arrays, NaNs (Not a Number values), and NumPy masked arrays. As noted in [9], at least thirty issues of this type have been reported, and many others exist. Implementing these behaviors and verifying their correctness throughout scipy.stats it will greatly improve the usability of the subpackage.
- An infrastructure for hypothesis tests. Traditionally, every function that performs a hypothesis test in SciPy is implemented from scratch. We will create a common infrastructure for hypothesis test to resolve the bugs in legacy hypothesis tests and facilitate the implementation of consistent, reliable new tests.

Other uses of SciPy in biomedicine is spread among many other subpackages. Specific tasks based on findings of our literature review and surveys are summarized in [6]. We will address these items as part of the project.

2) For each hour spent on the tasks above, we will spend one hour on maintenance, taming SciPy's growing backlog of pull requests in need of review and issues to be addressed. Resolving these maintenance issues frees volunteer time for projects they find more exciting, improves project morale and health, and fixes bugs that have plagued SciPy users – and users of downstream libraries – for years.

3) Dissemination of results:

- Documentation – We will set aside 5% of our time for improvements to documentation. All functions need practical examples, so we will add examples from the biomedical literature.

- Office Hours – To provide a more direct link between SciPy learners and experts, the applicants will organize SciPy "Office Hours". Once per month, we will arrange for a SciPy expert to host a live stream on Twitch focused on answering questions from users. The applicants will act as moderators, presenting questions to the host for live video response. Office hours will be advertised on the SciPy website, mailing list, and social media, and we will specifically invite biomedical researchers and software maintainers that have responded to our surveys.
- Sprints, Tutorials, and Talks – Using funding separate from this grant, we will host in-person sprints, tutorials, and talks at SciPy 2023 and other conferences to support users and reinforce the relationships between SciPy and the greater open-source scientific software community.
- Direct Communication – In preparation for this proposal, we have developed relationships with dozens of biomedical researchers and software developers. We will continue to discuss our work with them and invite them to be part of the development process (e.g. interface design, technical review).

Dr. Haberland and Dr. Roy are two of SciPy's most qualified maintainers for these tasks. For instance, since the beginning of 2022, they have authored over 100 merged PRs and have made hundreds of other contributions in the form of commits to PRs authored by others, code review, and issue comments. Over the past two years, they have been the #1 and #3 contributors by number of commits [10], and they are the #1 and #4 "first responders" to new issues [11].

The applicants and their organizations will cover the cost of travel for outreach and will provide all other required resources (e.g. computers, reference materials). Both applicants are core developers of SciPy with commit rights, so they will tag-team to author and review/merge the proposed work.

References

[1] Nature, "Research articles | Nature Biomedical Engineering," [Online]. Available: https://www.nature.com/natbiomedeng/research-articles?year=2022. [Accessed 20 May 2022].

[2] Nature, "Research articles | Nature Biotechnology," [Online]. Available: https://www.nature.com/nbt/research-articles?type=article&year=2022. [Accessed 20 May 2022].

[3] NEJM Group, "The New England Journal of Medicine: Table of Contents," [Online]. Available: https://www.nejm.org/toc/nejm/386/. [Accessed 5 May 2022].

[4] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland and e. al, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods,* pp. 261-272, 3 February 2020.

[5] Google Scholar, "Biomedical Articles Citing SciPy 1.0: fundamental algorithms for scientific computing in Python," [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=2005&sciodt=0%2C5&cites=13879940319533982630&scipsc=1&q=biomedical&btnG=. [Accessed 20 May 2022].

[6] M. Haberland, "SciPy for Biomedicine · Issue #16191 · scipy/scipy," 15 May 2022. [Online]. Available: https://github.com/scipy/scipy/issues/16191. [Accessed 20 May 2022].

[7] The SciPy Community, "SciPy Roadmap," [Online]. Available: http://scipy.github.io/devdocs/dev/roadmap.html.

[8] The SciPy Community, "Detailed SciPy Roadmap," [Online]. Available: http://scipy.github.io/devdocs/dev/roadmap-detailed.html.

[9] M. Haberland, "ENH: stats: consistent nan_policy, axis, and masked array support · Issue #14651 · scipy/scipy," 26 August 2021. [Online]. Available: https://github.com/scipy/scipy/issues/14651. [Accessed 20 May 2022].

[10] GitHub, "Contributors to scipy/scipy," [Online]. Available: https://github.com/scipy/scipy/graphs/contributors?from=2020-05-23&to=2022-05-23&type=c. [Accessed 23 05 2022].

[11] The SciPy Project, "Scientific Python Developer Statistics," [Online]. Available: https://github.com/scipy/scipy/graphs/contributors?from=2020-05-23&to=2022-05-23&type=c. [Accessed 23 05 2022].

[12] M. Johnson, "Statistical Analysis Software Programs in Biomedical Research," *Materials and Methods,* vol. 1282, no. 4, 2022.

[13] M. Mendonça, "Newcomers meetings," 4 May 2022. [Online]. Available: https://mail.python.org/archives/list/scipy-dev@python.org/thread/NMYTU4W4LCU2VFK4LITGYVWGHDPM3BON/.

[14] M. Mendonça, "DOC: Revamp contributor setup guides by melissawm · Pull Request #15947 · scipy/scipy," 6 April 2022. [Online]. Available: https://github.com/scipy/scipy/pull/15947. [Accessed 21 May 2022].

[15] M. Haberland, "ENH: stats: univariate distribution meta-issue · Issue #15928 · scipy/scipy," 2 April 2022. [Online]. Available: https://github.com/scipy/scipy/issues/15928. [Accessed 20 May 2022].

8. *List expected milestones and deliverables, and their expected timeline. Be specific and include where possible any goals for metrics the software project(s) are expected to reach upon completion of the grant. Please use a third-person voice (maximum of 500 words).*

Dates assume a project start of November 1, 2022. PRs will be developed during the months indicated; a two-week opportunity for community review will begin in the following month. Dr. Haberland and Dr. Roy are both SciPy core developers with commit rights; they will review the work of one another and the student assistants, and they will merge PRs after all feedback from the community review period is resolved.

The number of hours required for each activity was estimated by each author independently, then discussed until consensus was reached. The number of months required for each deliverable assumes 50% level of effort (LoE) on average for Dr. Roy and associates and, for Dr. Haberland and students, 25% LoE during months October – March and 100% LoE April – September.

November 2022 – October 2024 (ongoing): general maintenance, documentation, outreach, and mentoring
November 2022: false discovery rate controlling procedures
December 2022: adding examples to documentation (dedicated time in addition to ongoing efforts)
January 2023: fundamental survival analysis tools
February 2023: post hoc tests
March 2023 – September 2023: univariate distribution infrastructure overhaul
October 2023 – March 2024: stats function API consistency (other than distribution infrastructure)
April 2024: hypothesis testing infrastructure
May 2024 – October 2024: enhancements to other subpackages suggested by biomedical researchers and software developers [6]

The metric for completion of most milestones listed above is PRs merged into the main branch of SciPy. There will be a one-to-one correspondence between the identified features and PRs except for the univariate distribution infrastructure overhaul (one per distribution, ~125 PRs), stats function API consistency (one per function, ~50 PRs), and documentation (one per function, ~50 PRs).

The primary metric for maintenance work is the number of issues closed (assuming that each PR closes at least one open issue). Based on the time allocated to maintenance and an estimate of 10 hrs / issue, our goal is to be significantly involved in the resolution of at least one issue per week October – March and four issues per week April – September each year for a total of ~260 issues over the course of the project. In addition to closing issues, we will enable progress toward many more.

The metrics for dissemination of results are adding biomedical examples for at least 50 SciPy functions, hosting 24 office hours, and participating in SciPy Conferences (2023 and 2024) and all CZI meetings.

Progress toward these deliverables will be tracked using GitHub Projects and shared in the reports.

9. List active and recently completed (previous two calendar years) financial or in-kind support for the software project(s), including duration, total costs in USD, and source of funding. Include any previous funding for these software projects received from CZI outside of the EOSS program (maximum of 250 words).

SciPy was awarded $200k in CZI EOSS Cycle 1 for the project "A Solid Foundation for Statistics in Python with SciPy" (EOSS-0000000432). The grant concluded in 2021 after ~1.5 years (including a no-cost extension).

SciPy and collaborating projects NumPy, Matplotlib, and pandas were awarded $400k for the CZI EOSS D&I project “Advancing an Inclusive Culture in the Scientific Python Ecosystem” (EOSS-DI-0000000031). The duration of the project is two years, and the project is still active.

SciPy and collaborating projects NumPy, pandas, and scikit-learn were awarded $1.383M for the NASA ROSES-2020 project “Reinforcing the Foundations of Scientific Python”. The project timeline is from January 2022 – January 2025. Only a fraction of the grant supports SciPy, and the objectives of the grant (e.g. multiprocessing and distributed memory support) are different from the present proposal.

SciPy has been awarded several short-term (~6 month) Small Development Grants from NumFOCUS:

Introducing Users to Powerful New Features of SciPy – $4003 – 2022
A Mixed Integer Programming Solver for SciPy – $4985 – 2021
Add PROPACK Sparse SVD to SciPy – $3550 – 2021
Improving Boundary Handling and Data Type Support in scipy.ndimage – $5000 – 2000
Enhanced LAPACK Support in SciPy – $4978 – 2019
Complete the SciPy Special Functions Documentation – $2500 – 2019
SciPy Development Documentation Overhaul – $4274 – 2019
An Efficient, High-Level Implementation of Linear Programming – $2000 – 2018
Maturing a Sparse Array Implementation for SciPy – $3000 – 2018

SciPy has received $2500 per month from Tidelift since June 2019. This funding is slated for relatively small maintenance projects.

10. *Describe the other software tools (either proprietary or open source) that the audience for this proposal primarily uses. How do the software project(s) in this proposal compare to these other tools in terms of user base size, usage, and maturity? How do existing tools and the project(s) in this proposal interact? (maximum of 250 words).*

Many biomedical researchers use software tools for statistical analysis. According to [12] and our own review of the March issues of several biomedical research journals, GraphPad Prism is (by far) the most popular tool for this purpose. Prism was first released many years before the SciPy project began, so Prism 9.3.1 is a very mature, commercially developed product. SciPy maintainers refer to Prism documentation for inspiration regarding statistical functionality, interface, and documentation enhancements. Since Prism is proprietary, relies on a graphical user interface, and is only available on Windows and macOS, advantages of SciPy include cross-platform compatibility, easy interfacing with other code, different features, and a permissive open source license. Of the other tools used for statistical analysis, almost all are proprietary. Compared to the only other popular open-source alternative (tools written in R), SciPy has a more permissive license and is more self-contained. Other popular statistical tools for Python (e.g. statsmodels, pingouin, pymc3) are complementary: all of them use SciPy for the basics and add more advanced functionality.

The landscape of scientific software tools for other biomedical research needs (e.g. optimization and curve fitting, interpolation, integration, clustering, image processing, and linear algebra) is similar: biomedical researchers also use more mature, commercially developed, proprietary tools (e.g. Matlab, Mathematica, Excel), but SciPy offers a comprehensive, permissively licensed, cross-platform, free, and open-source alternative. Other tools in the scientific Python ecosystem depend on SciPy and extend its capabilities for more specific applications (e.g. DeepLabCut, MDAnalysis, MNE-Python).

11. Describe the expected value of the proposed work to the biomedical research community (maximum of 250 words). (auto-filled from LOI; update if neeeded)

The biomedical research community depends on Python tools, and Python tools depend on SciPy. Out of 55 CZI EOSS grants for Python software projects, 40% involve projects that list SciPy as a dependency. Accordingly, Goal 1a will impact the biomedical research community via faster, more accurate, and more reliable performance of the software tools they already rely on. These enhancements will also allow biomedical research tools to develop more quickly because the developers can increase their reliance on SciPy to perform lower-level tasks.

By adding new features requested by the biomedical community, Goal 1b will benefit biomedical researchers who use SciPy firsthand. These improvements will enable the use of SciPy for more (if not all) of their analysis pipelines, and in some cases, enable work that would have been impractical before.

Goal 2 will make use of SciPy and its dependent packages more seamless. After upgrading to the latest version of SciPy, downstream developers and users will enjoy improved user interfaces and fewer unexpected behaviors, allowing them to spend less time debugging code and more time on the valuable parts of their work.

Finally, while some of the proposed work will benefit the community in the background (e.g. bug fixes, enhancements for downstream developers), the outreach efforts of Goal 3 will ensure that the other improvements (e.g. new functions) are easy for biomedical researchers to learn about and use.

12. *Advancing DEI is a core value for CZI, and we are requesting information on your efforts in this area. Describe any efforts the software project(s) named in this proposal have undertaken to increase diversity, equity, and inclusion with respect to their contributors and audience. Please see examples from applications funded in previous cycles (maximum of 250 words)*

As an open-source project, anyone and everyone is welcome to contribute to SciPy. Accordingly, the SciPy project adopted the following diversity statement shortly before the release of SciPy 1.0 in late 2017:

"The SciPy project welcomes and encourages participation by everyone. We are committed to being a community that everyone enjoys being part of. Although we may not always be able to accommodate each individual's preferences, we try our best to treat everyone kindly.

No matter how you identify yourself or how others perceive you: we welcome you. Though no list can hope to be comprehensive, we explicitly honor diversity in age, culture, ethnicity, genotype, gender identity or expression, language, national origin, neurotype, phenotype, political beliefs, profession, race, religion, sexual orientation, socioeconomic status, subculture, and technical ability, to the extent that these do not conflict with this code of conduct."

It is not enough to be welcoming; we need to take an active approach to improve DEI and representation in our contributor and maintainer base. As part of the EOSS D&I joint project with NumPy, matplotlib, and pandas, we host mentored sprints at conferences (including PyCon and SciPy 2022) and monthly meetings dedicated to newcomers [13]. Other activities include focused guidance for new contributors and improvements to the developer documentation (e.g. [14]).

9. Open Source Project #1 Details:

1. Project Name: SciPy (Library)

2. Homepage URL: https://www.scipy.org/

3. Hosting Platform: GitHub

4. Main Repo: https://github.com/scipy/scipy

5. Short Description of Project:

SciPy is a library of numerical routines for the Python programming language that provides fundamental building blocks for modeling and solving scientific problems. SciPy includes algorithms for optimization, statistical analysis, integration, interpolation, eigenvalue problems, differential equations, fast Fourier transforms and many other classes of problems; it also provides specialized data structures, such as sparse matrices and k-dimensional trees. SciPy is built on top of NumPy, which provides array data structures and related fast numerical routines, and SciPy is itself the foundation on which higher level scientific libraries, including scikit-learn and scikit-image, are built.