Strengthening NumPy's foundations – growing beyond code

1 Goals

The goal of this proposal is to improve NumPy's sustainability and grow its core team and contributor community beyond the currently dominant focus on code. We aim to achieve this through:

1. improvements in NumPy's governance and organizational structure,
2. improvements in NumPy's website and documentation aimed at new users and contributors,
3. community outreach and mentoring of new core team members,
4. addressing key technical issues in OpenBLAS which regularly impact the NumPy release process.

NumPy [1] is a numerical library for Python that provides the key data structure - an N-dimensional array - and numerical algorithms on which much of the scientific Python ecosystem is built. NumPy is used in virtually every area of science and engineering, including in many biomedical tools and applications like QIIME [2], Qiita [3], MNE [4], scikit-bio [5], and GALA [6]. It is used in the data analysis pipelines of large collaborations like the Human Cell Atlas [7] and the International Brain Laboratory [8]. It is used directly by millions of scientists analyzing their own data, and is a key dependency of statistical, image analysis and visualization tools they use like Matplotlib [9], SciPy [10], Scikit-image [11], Scikit-learn [12], Pandas [13], Statsmodels [14], PyMC3 [15], Tensorflow [16], and PyTorch [17].

Relative to this massive user base, NumPy has a quite modestly sized team of maintainers. Of the 15 current maintainers, half are only occasionally active and the bulk of the code maintenance and review activities is performed by about 5-7 people [18]. Other important activities like writing high-level documentation, are done by even fewer of those maintainers.

OpenBLAS [19] is a library that provides accelerated versions of the BLAS [20] and LAPACK [21] linear algebra algorithms. It is a key dependency of NumPy and SciPy, as well as of R and Julia. So while most users will not realize they are using it, it is essential for the three leading scientific open source programming ecosystems. Moreover, there are few alternatives for OpenBLAS today – ATLAS [22] is significantly slower and dormant, BLIS/Flame [23] isn't yet ready as a replacement, and Intel MKL cannot be used by default for licensing reasons. While the OpenBLAS repository lists 122 contributors (over a timespan of 9 years and covering several CPU architectures), the project effectively only has a single maintainer and less than ten semi-regular participants. This presents a challenge. OpenBLAS issues regularly impact the NumPy release process, and most of the NumPy maintainers do not have the specific technical expertise needed to address such issues in the OpenBLAS code base. Therefore we aim to use part of the funding requested in this proposal to improve the sustainability and reduce technical debt of OpenBLAS.

This proposal aims to provide part-time funding for Ralf Gommers and Martin Kroeker, as well as for a technical writer and for a website redesign. Gommers has been a NumPy maintainer for 11 years and is driving NumPy's governance, website and documentation activities. He is employed by Quansight (US), where he leads the Labs division that is aiming to sustain and grow the larger scientific Python ecosystem. Kroeker has been the lead OpenBLAS maintainer for 4 years, and runs his own software business in Germany. The complete proposed team and their responsibilities are:

- Ralf Gommers (PI, 12 hrs/wk for 1 year); governance organizational work, mentoring of new maintainers, and managing documentation and website development work.
- Martin Kroeker (10 hrs/wk for 1 year); OpenBLAS technical improvements.
- Technical writer (to be hired at Quansight Labs, 32 hrs/wk for 1 year); high-level documentation and website content, and engaging with volunteer writers and educators.
- Web developer (150 hrs); website redesign.
- Graphic designer (150 hrs); website redesign.
- Student intern (3 months); OpenBLAS technical improvements.

1.1 Success evaluation

The proposed activities have, with the exception of mentoring, a lot of granular deliverables. More important though than determining whether those deliverables have been met, is whether they had the intended *impact*. That intended impact is to (a) significantly grow the NumPy team, thereby improving sustainability and accelerating the development of the project, (b) better serve new users and contributors, and (c) make technical improvements to OpenBLAS that also benefit NumPy.

These impacts are measurable, much more so than the impact of, e.g., adding new features to NumPy. In addition to a qualitative evaluation, we propose the following metrics:

- A >10x improvement in unique visitors to numpy.org (now 300,000/month).
- A >30% growth in number of people maintaining NumPy, as measured by people with commit rights to at least one NumPy repository.
- Dedicated website and documentation teams on GitHub (now non-existent) with at least two people per team.
- At least one new Institutional Partner or corporate sponsor to NumPy.
- A decrease in OpenBLAS issues reported on the NumPy issue tracker for new NumPy releases.

2 Work plan

2.1 Governance and organizational structure

NumPy has a formal, consensus-based governance document [24] which specifies how decisions get made. It defines a Steering Council and the process of people joining and leaving that council. This governance method has fulfilled its primary role, specifying decision making, well so far. However, it has not helped much in making clearer how the project is run on a daily basis, what the principles and rules for any given topic are and how and where people new to the project can best get involved.

We see an opportunity for significantly improving this, by writing a set of documents, guiding them through the community review and acceptance process, and then using those as tools for outreach, mentoring and empowering new and current team members. Concretely, we propose to:

1. Define explicit roles in the project structure and adding a mechanism for devolved decision making on code, documentation, website, and fundraising.
2. Write a new, aspirational scope and vision document.
   This will replace the current scope document [25], which only documents the status-quo.
3. Write a sponsorship and corporate engagement policy
4. Write an informational NumPy Enhancement Proposal (NEP) on principles and priorities of how to spend project funds.
   NumPy will always be a project with many times more volunteer contributors than funded people. Therefore having those funded people operate in ways that attract more volunteers and

enhance their participation experience is critical.

5. Write a backwards compatibility and deprecation policy NEP.
   This is the single most contentious issue in NumPy development, and therefore needs a formal policy.
6. Implement a policy on maintaining translations of high-level documentation and website content to other languages.
   The majority of users are not native English speakers. Some content (limited scope, so it remains maintainable) in their native language will be a major improvement in them feeling welcome and able to engage with the project.

Work will be performed by Ralf Gommers, using the standard collaboration methods of the project: GitHub, mailing list, and a weekly video call that is open to the whole community.

2.2 Website and high-level documentation

NumPy's documentation is contained largely in its user and reference guides. The user guide contains tutorials and explanations of concepts, while the reference guide contains detailed documentation of individual modules and functions. The reference guide is largely complete, while the user guide needs major improvements - in particular to cater to beginning users. Additionally the numpy.org website, the entry point to the project, needs to be completely rebuilt to guide users – as well as potential contributors, educators and other stakeholders – to the documentation they need.

We propose to address these content gaps through:

1. A full numpy.org website redesign, delivering a site with content and look-and-feel of similar quality to jupyter.org (which we consider as best-in-class for scientific Python project websites).
2. Rewriting the "What is NumPy" and "Installing NumPy" sections of the user guide.
3. Writing a set of engaging, beginner-friendly tutorials. At least five separate ones, for example:
   - Conway's game of life with NumPy
   - Build a simple neural net with only NumPy
   - Loading experimental data and analyzing it with NumPy's Fourier transforms and linear algebra functions.
4. Restructuring the user guide
5. Adding graphical navigation guides per chapter/topic in the reference guide.
6. Writing reviewer guidelines, focusing on engaging new contributors.
7. Writing documentation about non-coding contributions (e.g. triaging role, doc team, website contributions, showcases/testimonials, etc.)

Work will be performed by the technical writer, who is to be hired before the start of the grant, and the web development and design contractors. The writer will spend part of her time starting to build a dedicated documentation team with volunteers from the community.

2.3 Growing the team and mentoring

To make the NumPy project thrive, we need to significantly grow the core team. This includes attracting more maintainers, building new documentation and website teams, and defining and filling roles focused on marketing, funding and other organizational and management tasks.

The concrete deliverables to enable this are discussed in section 2.1 of this proposal. In addition

to those, we recognize that we need to spend significant time on outreach to attract new volunteers, and on mentoring them once those volunteers show up. Activities will include:

- Attending weekly community video calls.
- Setting up a new real-time communication method (e.g. Gitter or Slack) and being available there to help new contributors.
- Giving talks about NumPy at 3-4 conferences (SciPy, PyData and PyCon events), organizing sprints at those events, and advertising volunteer opportunities there.
- Connecting volunteers with relevant resources and people on the team, and empowering them to make decisions and take ownership of parts of the project.

We will pay specific attention to mentoring volunteers from under-represented groups.

Work will be performed by Ralf Gommers and the technical writer. Gommers' employer Quansight will fund the travel to the conferences.

2.4 OpenBLAS technical improvements

We propose to tackle the three categories of technical issues that together cause the majority of issues in upstream libraries:

1. Identify and remove any remaining thread safety problems. This may require a redesign of fairly fundamental code in OpenBLAS.
2. Fix all the known bugs in the 512-bit Advanced Vector Extensions (AVX-512) code, and add more comprehensive AVX-512 support.
3. Identify and fix any remaining bugs in the recently added Thread Local Storage (TLS) code.

Furthermore we propose to implement improvements (in particular to the time complexity and stack usage of some of its algorithms) that OpenBLAS needs in the Recursive LAPACK Collection (ReLAPACK [26]). ReLAPACK development is stagnant and OpenBLAS is its main user, therefore these improvements will likely only materialize if they are done within the context of the OpenBLAS project.

Work will be done by Martin Kroeker and the student intern, and delivered via pull requests on GitHub. Kroeker will recruit and mentor the intern.

3 Existing support

The Berkeley Institute for Data Science has received two grants for the project "Improving NumPy for Better Data Science".

- $659,359 from the Alfred P. Sloan Foundation, Apr 2018 - Oct 2020.
- $645,020 from the Gordon and Betty Moore Foundation, Apr 2018 - Oct 2020.

The main focus of those grants is to reduce technical debt and implement architectures improvements that make NumPy easier to extend. It supports two full-time NumPy developers at Berkeley. These are the only grants NumPy has received to date.

Beginning May 2019 NumPy signed up with Tidelift. In exchange for maintenance commitments around NumPy releases and security vulnerabilities, Tidelift pays NumPy $1,000 per month until October 2020.

OpenBLAS has not had any financial support since September 2014. During 2009-2014 the creation of OpenBLAS was supported by several grants of Chinese institutional funders to the

University of Chinese Academy of Sciences in Beijing, as listed on the homepage [27] of OpenBLAS's original developer Zhang Xianyi (who based OpenBLAS on GotoBLAS, which had funding before 2008).

References

[1] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, p. 22, 2011.

[2] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Pena, J. K. Goodrich, J. I. Gordon, *et al.*, "QIIME allows analysis of high-throughput community sequencing data," *Nature methods*, vol. 7, no. 5, p. 335, 2010.

[3] A. Gonzalez, J. A. Navas-Molina, T. Kosciolek, D. McDonald, Y. Vázquez-Baeza, G. Ackermann, J. DeReus, S. Janssen, A. D. Swafford, S. B. Orchanian, *et al.*, "Qiita: rapid, web-enabled microbiome meta-analysis," *Nature methods*, vol. 15, no. 10, p. 796, 2018.

[4] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "Mne software for processing MEG and EEG data," *Neuroimage*, vol. 86, pp. 446–460, 2014.

[5] "Scikit-bio, http://scikit-bio.org." Accessed: 2019-07-26.

[6] J. Nunez-Iglesias, R. Kennedy, S. M. Plaza, A. Chakraborty, and W. T. Katz, "Graph-based active learning of agglomeration (gala): a python library to segment 2D and 3D neuroimages," *Frontiers in neuroinformatics*, vol. 8, p. 34, 2014.

[7] A. Regev, S. A. Teichmann, E. S. Lander, I. Amit, C. Benoist, E. Birney, B. Bodenmiller, P. Campbell, P. Carninci, M. Clatworthy, *et al.*, "Science forum: the human cell atlas," *Elife*, vol. 6, p. e27041, 2017.

[8] L. F. Abbott, D. E. Angelaki, M. Carandini, A. K. Churchland, Y. Dan, P. Dayan, S. Deneve, I. Fiete, S. Ganguli, K. D. Harris, *et al.*, "An international laboratory for systems and computational neuroscience," *Neuron*, vol. 96, no. 6, pp. 1213–1218, 2017.

[9] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing in science & engineering*, vol. 9, no. 3, p. 90, 2007.

[10] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İlhan Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . Contributors, "SciPy 1.0 – fundamental algorithms for scientific computing in python," 2019.

[11] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[13] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445, pp. 51–56, Austin, TX, 2010.

[14] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, vol. 57, p. 61, Scipy, 2010.

[15] J. Salvatier, T. V. Wiecki, and C. Fonnesbeck, "Probabilistic programming in python using PyMC3," *PeerJ Computer Science*, vol. 2, p. e55, 2016.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[18] R. Gommers, "Inside NumPy: preparing for the next decade," 2019. https://www.slideshare.net/RalfGommers/inside-numpy-preparing-for-the-next-decade.

[19] Q. Wang, X. Zhang, Y. Zhang, and Q. Yi, "AUGEM: automatically generate high performance dense linear algebra kernels on x86 CPUs," in *SC'13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pp. 1–12, IEEE, 2013.

[20] J. J. Dongarra, J. Du Croz, S. Hammarling, and R. J. Hanson, "An extended set of FORTRAN basic linear algebra subprograms," *ACM Transactions on Mathematical Software (TOMS)*, vol. 14, no. 1, pp. 1–17, 1988.

[21] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen, "Lapack: A portable linear algebra library for high-performance computers," in *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, pp. 2–11, IEEE Computer Society Press, 1990.

[22] R. C. Whaley and J. J. Dongarra, "Automatically tuned linear algebra software," in *SC'98: Proceedings of the 1998 ACM/IEEE conference on Supercomputing*, pp. 38–38, IEEE, 1998.

[23] F. G. Van Zee and R. A. Van De Geijn, "BLIS: A framework for rapidly instantiating BLAS functionality," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, no. 3, p. 14, 2015.

[24] "NumPy governance, https://numpy.org/devdocs/dev/governance/index.html." Accessed: 2019-07-26.

[25] "NumPy scope, http://numpy.org/neps/scope.html." Accessed: 2019-07-26.

[26] E. Peise and P. Bientinesi, "Recursive algorithms for dense linear algebra: The ReLAPACK collection," *CoRR*, vol. abs/1602.06763, 2016.

[27] "Zhang Xianyi's homepage, https://xianyi.github.io." Accessed: 2019-07-26.