Bioconda: A sustainable and comprehensive software distribution for the life sciences

Björn Grüning

Ryan Dale[[1]](#footnote-21),

Andreas Sjödin

Brad A. Chapman

Jillian Rowe

Christopher H. Tomkins-Tinch

Renan Valieris

Adam Caprez

Bérénice Batut

Mathias Haudgaard

Thomas Cokelaer

Kyle A. Beauchamp

Brent S Pedersen

Youri Hoogstrate

Devon Ryan

Anthony Bretaudeau

Gildas Le Corguillé

Christian Brueffer

Dilmurat Yusuf

Sebastian Luna-Valero

Rory Kirchner

Karel Brinda

Martin Raden

Thomas Wollmann

Simon J. van Heeringen

Lorena Pantano

Nicola Soranzo

Zachary Charlop-Powers

Matthias De Smet

Per Unneberg

Marcel Martin

Greg Von Kuster

Milad Miladi

Tiago Antao

Marius van den Beek

Kevin Thornton

Daniel Maticzka

Clemens Blank

Sebastian Will

Kévin Gravouil

Joachim Wolff

Manuel Holtgrewe

Vitor C. Piro

Marcel Bargull

Jörg Fallmann

Ilya Shlyakhter

Philip Mabon

Ayman Yousif

Xiao-Ou Zhang

Wei Shen

Jennifer Cabral

Eric Enns

Cristel Thomas

Joseph Brown

Jerome Kelleher

Dave Bouvier

Mattias de Hollander

Jorrit Boekel

Nitesh Turaga

Julian R. de Ruiter

Simon Gladman

Nicholas Harding

Saket Choudhary

Florian Eggenhofer

Aziz Khan

Zhuoqing Fang

Peter J. A. Cock

Arne Kratz

Henning Timm

Robert Kleinkauf

Endre Bakken Stovner

Hai Nguyen

Enrico Seiler

Matt Chambers

Philip Ewels

Colin Brislawn

Elmar Pruesse

James E. Johnson

Emil Hägglund

Roman Valls Guimera

Lance Parsons

Simon Ye

Rob Patro

Alex Reynolds

W. Augustine Dunn

Nicholas Stoler

Elena Grassi

MacIntosh Cornwell

David Koppstein

Inken Wohlers

Daniel Blankenberg

Guowei He

Alexander Junge

Rick Farouni

Mallory Freeberg

Sahil Seth

Sourav Singh

Daniel R. Bogema

Nicholas R. Waters

David E Larson

Sacha Laurent

Liang-Bo Wang

Fabio Cumbo

David Laehnemann

Matthew L. Workentine

Upendra Kumar Devisetty

James Abbott

Aroon T. Chande

Pierrick Roger

Tobias Rausch

Rasmus Agren

John M Eppley

Xavier Garnier

Wei Li

James Taylor

Bianca Katharina Stöcker

Davide Chicco

Jasmijn A. Baaijens

Iain Milne

Adam P. Taranto

Nezar Abdennur

Jens Preussner

Patrick R. Wright

Bengt Sennblad

Oana M. Enache

Matthew Gopez

Michael Knudsen

Philip Reiner Kensche

Ino de Bruijn

Kai Blin

Avi Srivastava

Yuri Pirola

Luca Pinello

Giorgio Gonnella

Camy Tran

Christoph Stahl

Saskia Hiltemann

Alistair Miles

Vivek Rai

Yannick Boursin

Matthew L. Bendall

Sebastian Schmeier

Kevin Arvai

Yasset Perez-Riverol

Simon Dirmeier

Matthieu Jung

Erik Clarke

Gianvito Urgese

Julien Seiler

Alexander S Mikheyev

Luis Pedro Coelho

Daniela Beisser

Jordi Capellades

Oleksandr Moskalenko

Sven Rahmann

Josh Moore

Gordon Stephen

Timothy H. Webster

Christopher Schröder

Adrian Emanuel Salatino

Tomás Di Domenico

Etienne Kornobis

Eric Rasche

Johannes Köster[[2]](#footnote-22),

**To the Editor:** Bioinformatics software comes in a variety of programming languages and diverse installation methods, which are frequently not straightforward. This heterogeneity makes management of a software stack complicated and error-prone while consuming inordinate amounts of time. Moreover, it inhibits reproducible science 1,2,3, because it is hard to recreate a software stack on different machines. While deployment of software has traditionally been handled by administrators, reproducibility of data analyses requires that the researcher is able to maintain full control of the software environment and rapidly modify it without administrative privileges.

The Conda package manager (<https://conda.io>) has become an increasingly popular approach to overcome these challenges for all major operating systems. Conda normalizes software installations across language ecosystems by describing each software with a human readable *recipe* that defines meta-information and dependencies, as well as a simple *build script* that performs the steps necessary to build and install the software. Conda builds software packages within an isolated environment, transforming them into relocatable binaries. Importantly, it obviates reliance on system-wide administation privileges by allowing users to generate isolated software environments, within which software versions can be managed per-project, without generating incompatibilities and side-effects (see Supplementary Results S3). These environments support reproducibility, as they can be rapidly exchanged via files that describe their installation state. Conda is tightly integrated into popular solutions for reproducible data analysis like Galaxy 4, bcbio-nextgen (<https://github.com/chapmanb/bcbio-nextgen>), and Snakemake 5. To further enhance reproducibility guarantees, Conda can be combined with container or virtual machine based approaches (see Supplementary Results S2) and archive facilities such as Zenodo (see Supplementary Results S3.1). Finally, while Conda provides many commonly-used packages by default, it also allows users to optionally include additional, community-managed, repositories of packages (termed *channels)*.

In order to unlock the benefits of Conda for the life sciences, we present the Bioconda project (<https://bioconda.github.io>). The Bioconda project provides over 3000 Conda software packages for Linux and macOS. Rapid turnaround times (Supplementary Results S1, Fig. S1e) and extensive documentation (<https://bioconda.github.io/contributing.html>), have led to a growing community of over 200 international scientists working in the project (Supplementary Results S1, Fig. S1a,b). The project is led by a core team, complemented by interest groups for particular language ecosystems. Unlimited (in time and space) storage for generated packages is donated by Anaconda Inc. All other used infrastructure is free of charge. Bioconda provides packages from various language ecosystems like Python, R (CRAN and Bioconductor), Perl, Haskell, Java,  and C/C++ (Fig. ???a). Many of these packages have complex dependency structures that require various manual steps to install when not relying on a package manager like Conda (Supplementary Fig. S1c). With over 6.3 million downloads, Bioconda has become a backbone of bioinformatics infrastructure that is used heavily across all language ecosystems (Fig. ???b). Bioconda is complemented by the conda-forge project (<https://conda-forge.github.io>), which hosts software not specifically related to the biological sciences. This separation has proven to be beneficial, because the focused nature of the Bioconda community allows for fast turnaround times and support when contributing packages or fixing problems. Nevertheless, the two projects collaborate closely, and the Bioconda team maintains over 500 packages hosted by conda-forge.

Bioconda is not the only effort to distribute bioinformatics software (Fig. ???c). The alternatives can be categorized into system-wide (Debian-Med, Genotoo Science, Biolinux, Homebrew) and per user (EasyBuild, GNU Guix, BioBuilds) installation mechanisms. The system-wide approaches lack the ability to put the scientist in control of the installed software stack, thus not solving the requirements for reproducibility outlined above. All per-user based approaches provide a similar feature set (BioBuilds is even also using the Conda package manager). However, among all available approaches, Bioconda is by far the most comprehensive, while being among the youngest (Fig. ???c).

Package numbers and usage. (a) Package count per language ecosystem (saturated colors on bottom represent explicitly life science related packages). (b) Distribution of per-package downloads separated by language ecosystem. The term other entails all packages that do not fall into one of the specific categories. White dots represent the mean, dark bars represent the interval between upper and lower quartile. (c) Comparison of explicitly life science related packages in Bioconda with Debian Med (https://www.debian.org/devel/debian-med), Gentoo Science Overlay (category sci-biology,  https://github.com/gentoo/sci), EasyBuild (module bio, https://easybuilders.github.io/easybuild), Biolinux 6, Homebrew Science (tag bioinformatics, https://brew.sh), GNU Guix (category bioinformatics, https://www.gnu.org/s/guix), and BioBuilds (https://biobuilds.org). The lower panel shows the project age since the first release or commit. Note that various non-bio-related packages that started in Bioconda have since been migrated to the general-purpose conda-forge channel. Older versions of such packages still reside in the Bioconda channel, and as such are included in the download count (b). Statistics obtained Oct. 25, 2017. 

Package numbers and usage. (a) Package count per language ecosystem (saturated colors on bottom represent explicitly life science related packages). (b) Distribution of per-package downloads separated by language ecosystem. The term “other” entails all packages that do not fall into one of the specific categories. White dots represent the mean, dark bars represent the interval between upper and lower quartile. (c) Comparison of explicitly life science related packages in Bioconda with Debian Med ([https://www.debian.org/devel/debian-med](https://www.debian.org/devel/debian-med/)), Gentoo Science Overlay (category sci-biology,  <https://github.com/gentoo/sci>), EasyBuild (module bio, [https://easybuilders.github.io/easybuild](https://easybuilders.github.io/easybuild/)), Biolinux 6, Homebrew Science (tag bioinformatics, [https://brew.sh](https://brew.sh/)), GNU Guix (category bioinformatics, [https://www.gnu.org/s/guix)](https://www.gnu.org/s/guix/), and BioBuilds (<https://biobuilds.org>). The lower panel shows the project age since the first release or commit. Note that various non-bio-related packages that started in Bioconda have since been migrated to the general-purpose conda-forge channel. Older versions of such packages still reside in the Bioconda channel, and as such are included in the download count (b). Statistics obtained Oct. 25, 2017.

For reproducible data science, it is crucial that software libraries and tools are provided via an easy to use, unified interface, such that they can be easily deployed and sustainably managed. With its ability to maintain isolated software environments, the integration into major workflow management systems and the fact that no administration privileges are needed, the Conda package manager is the ideal tool to ensure sustainable and reproducible software management. With Bioconda, we unlock Conda for the life sciences. Bioconda offers a comprehensive resource of thousands of software libraries and tools that is maintained by hundreds of international contributors. Although it is among the youngest, it outperforms all competing projects by far in the number of available packages. With over six million downloads so far, Bioconda packages have been well received by the community. We invite everybody to join the Bioconda community, participate in maintaining or publishing new software, and reach the goal of a central, comprehensive, and language-agnostic collection of easily installable software for the life sciences.

# Data and code availability

Data and code underlying the presented results are enclosed in a Snakemake workflow archive available at <https://doi.org/10.5281/zenodo.1068297>. The archive can also be used to automatically reproduce all results and figures presented in this work. Also see the “Life Sciences Reporting Summary”.

# Acknowledgments

We thank the participants of various hackathons (e.g., the GalaxyP and IUC contribution fest, ELIXIR BioContainers and NETTAB hackathon) and all contributors that joined us after writing this manuscript for porting numerous packages to Bioconda. We thank the Anaconda Inc. team and the conda-forge team for the excellent cooperation and support. The Bioconda project receives funding from Anaconda, Inc., Austin, TX, USA, in the form of unlimited (in time and space) storage for Bioconda packages on their hosting service (<https://anaconda.org>). Further, the project has been granted extended build times from Travis CI ([https://travis-ci.com](https://travis-ci.com/)) and a free open source Linux and MacOS builds plan from Circle CI (<https://circleci.com>). The Bioconda community would like to thank ELIXIR ([https://www.elixir-europe.org](https://www.elixir-europe.org/)) for their constant support and donating staff. Ryan Dale was supported by the Intramural Program of the National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, USA. Johannes Köster was supported by the The Netherlands Organisation for Scientific Research (NWO) via a VENI grant and (in part) by German Research Foundation SFB 876.

# Author contributions

Kyle Beauchamp, Christian Brueffer, Brad Chapman, Ryan Dale, Florian Eggenhofer, Björn Grüning, Johannes Köster, Elmar Pruesse, Martin Raden, Jillian Rowe, Devon Ryan, Ilya Shlyakter, Andreas Sjödin, Christopher Tomkins-Tinch, and Renan Valieris (in alphabetical order) have written the manuscript. Johannes Köster has drafted and finalized the manuscript. Johannes Köster and Ryan Dale have conducted the data analysis. Dan Ariel Søndergaard contributed by supervising student programmers on writing Conda package recipes and maintaining the connection with ELIXIR. All other authors have contributed or maintained recipes.

# Competing Financial Interests

No competing interests.

# References

1 Mesirov JP. Computer science. Accessible reproducible research. *Science* 2010; **327**: 415–6.

2 Baker M. 1,500 scientists lift the lid on reproducibility. *Nature* 2016; **533**: 452–4.

3 Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, Sert NP du *et al.*. A manifesto for reproducible science. *Nature Human Behaviour* 2017; **1**: 0021.

4 Afgan E, Baker D, van den BM, Blankenberg D, Bouvier D, Čech M *et al.*. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res* 2016; **44**: W3–W10.

5 Köster J, Rahmann S. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics* 2012; **28**: 2520–2522.

6 Field D, Tiwari B, Booth T, Houten S, Swan D, Bertrand N *et al.*. Open software for biologists: from famine to feast. *Nat Biotechnol* 2006; **24**: 801–3.

1. Co-first author [↑](#footnote-ref-21)
2. To whom correspondence should be addressed. [↑](#footnote-ref-22)