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

Ryan Dale[1], Björn Grüning[*,2], Andreas Sjödin[3,4], Jillian Rowe[5], Brad A. Chapman[6], Christopher H. Tomkins-Tinch[7,8], Renan Valieris[9], Bérénice Batut[2], Adam Caprez[10], Thomas Cokelaer[11], Dilmurat Yusuf[2], Kyle A. Beauchamp[12], Karel Brinda[13], Thomas Wollmann[14], Gildas Le Corguillé[15], Devon Ryan[16], Anthony Bretaudeau[17], Youri Hoogstrate[18], Brent S Pedersen[19], Simon van Heeringen[20], Martin Raden[2], Sebastian Luna-Valero[21], Nicola Soranzo[22], Matthias De Smet[23], Greg Von Kuster[24], Rory Kirchner[25], Lorena Pantano[6], Zachary Charlop-Powers[26], Kevin Thornton[27], Marcel Martin[28], Marius van den Beek[29], Daniel Maticzka[2], Milad Miladi[2], Sebastian Will[30], Kévin Gravouil[31], Per Unneberg[32], Christian Brueffer[33], Clemens Blank[2], Vitor C. Piro[34,35], Joachim Wolff[2], Tiago Antao[36], Simon Gladman[37], Ilya Shlyakhter[8], Mattias de Hollander[38], Philip Mabon[39], Wei Shen[40], Jorrit Boekel[41], Manuel Holtgrewe[42,43], Dave Bouvier[44], Julian R. de Ruiter[45], Jennifer Cabral[39], Saket Choudhary[46], Nicholas Harding[47], Robert Kleinkauf[48], Eric Enns[39], Florian Eggenhofer[2], Joseph Brown[49], Peter J. A. Cock[50], Henning Timm[51], Cristel Thomas[52], Xiao-Ou Zhang[53], Matt Chambers[54], Nitesh Turaga[55], Enrico Seiler[34], Colin Brislawn[56], Elmar Pruesse[57], Jörg Fallmann[58], Jerome Kelleher[47], Hai Nguyen[59], Lance Parsons[60], Zhuoqing Fang[61], Endre Bakken Stovner[62], Nicholas Stoler[63], Simon Ye[64], Inken Wohlers[65], Rick Farouni[66], Mallory Freeberg[67], James E. Johnson[68], Marcel Bargull[51], Philip Reiner Kensche[69], Timothy H. Webster[70], John M Eppley[71], Christoph Stahl[72], Alexander S Rose[73], Alex Reynolds[74], Liang-Bo Wang[75,76], Xavier Garnier[17,77], Simon Dirmeier[78], Michael Knudsen[79], James Taylor[80], Avi Srivastava[81], Vivek Rai[82], Rasmus Agren[83], Alexander Junge[84], Roman Valls Guimera[85], Aziz Khan[86], Sebastian Schmeier[87], Guowei He[88], Luca Pinello[66], Emil Hägglund[89], Alexander S Mikheyev[90,91], Jens Preussner[92], Nicholas R. Waters[93], Wei Li[94], Jordi Capellades[95], Aroon T. Chande[96], Yuri Pirola[97], Saskia Hiltemann[98], Matthew L. Bendall[99,100], Sourav Singh[101], W. Augustine Dunn[102], Alexandre Drouin[103], Tomás Di Domenico[104], Ino de Bruijn[105], David E Larson[106], Davide Chicco[107], Elena Grassi[108], Giorgio Gonnella[109], Jaivarsan B[110], Liya Wang[111], Franck Giacomoni[112], Erik Clarke[113], Daniel Blankenberg[114], Camy Tran[39], Rob Patro[81], Sacha Laurent[115], Matthew Gopez[39], Bengt Sennblad[32], Jasmijn A. Baaijens[116], Philip Ewels[117], Patrick R. Wright[2], Oana M. Enache[8], Pierrick Roger[118], Will Dampier[119], David Koppstein[120], Upendra Kumar Devisetty[121], Tobias Rausch[122], MacIntosh Cornwell[123], Adrian Emanuel Salatino[124], Julien Seiler[125], Matthieu Jung[125], Etienne Kornobis[126], Fabio Cumbo[127,128,63], Bianca Katharina Stöcker[51], Oleksandr Moskalenko[129], Daniel R. Bogema[130], Matthew L. Workentine[131], Stephen J. Newhouse[132,133,134], Felipe da Veiga Leprevost[135], Kevin Arvai[136], and Johannes Köster[†,137,138]

[1]Laboratory of Cellular and Developmental Biology, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, United States
[2]Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg,

Germany
[3]Division of CBRN Security and Defence, FOI - Swedish Defence Research Agency, Umeå. Sweden
[4]Department of Chemistry, Computational Life Science Cluster (CLiC), Umeå University, Umeå, Sweden
[5]NYU Abu Dhabi, Abu Dhabi, United Arab Emirates
[6]Harvard T.H. Chan School of Public Health, Boston, United States
[7]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, United States
[8]Broad Institute of MIT and Harvard, Cambridge, United States
[9]Laboratory of Bioinformatics and Computational Biology, A. C. Camargo Cancer Center, São Paulo, Brazil
[10]Holland Computing Center, University of Nebraska, Lincoln, United States
[11]Institut Pasteur - Bioinformatics and Biostatistics Hub - C3BI, Paris, France
[12]Counsyl, South San Francisco, United States
[13]Center for Communicable Disease Dynamics, Department of Epidemiology, Harvard TH Chan School of Public Health, Boston, United States
[14]University of Heidelberg and DKFZ, Heidelberg, Germany
[15]UPMC, CNRS, FR2424, ABiMS, Station Biologique, Roscoff, France
[16]Bioinformatics core facility, Max Planck Institute for Immunobiology and Epigenetics, Freiburg, Germany
[17]INRA, UMR IGEPP, BioInformatics Platform for Agroecosystems Arthropods (BIPAA), Campus Beaulieu, Rennes, France
[18]Erasmus Medical Center, Department of Urology, Rotterdam, The Netherlands
[19]Department of Human Genetics, University of Utah, Eccles Institute of Human Genetics, Salt Lake City
[20]Radboud University, Faculty of Science, Department of Molecular Developmental Biology, Radboud Institute for Molecular Life Sciences, Nijmegen, The Netherlands
[21]MRC Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, United Kingdom
[22]Earlham Institute, Norwich Research Park, Norwich, United Kindgom
[23]Ghent University Hospital, Ghent University, Belgium
[24]Institute for CyberScience, Penn State University, University Park, United States
[25]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, United States
[26]The Laboratory for Genetically Encoded Small Molecules, The Rockefeller University, New York, United States
[27]Department of Ecology and Evolutionary Biology, University of California Irvine, Irvine, United States
[28]Department of Biochemistry and Biophysics, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Stockholm University, Sweden
[29]Stem Cells and Tissue Homeostasis, Institut Curie, Paris, France
[30]Institute for Theoretical Chemistry, University of Vienna, Vienna, Austria
[31]Université Clermont Auvergne, INRA, MEDIS, Clermont-Ferrand, France

[32]Department of Cell and Molecular Biology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Uppsala, Sweden
[33]Division of Oncology and Pathology, Department of Clinical Sciences, Lund University, Lund, Sweden
[34]Bioinformatics Unit, Robert Koch Institute, Berlin, Germany
[35]CAPES Foundation, Ministry of Education of Brazil, Brasília, Brazil
[36]Division of Biological Sciences, University of Montana, Missoula, United States of America
[37]Melbourne Bioinformatics, University of Melbourne, Melbourne, Australia
[38]Netherlands Institute of Ecology (NIOO-KNAW), Wageningen, The Netherlands
[39]National Microbiology Laboratory, Public Health Agency of Canada, Winnipeg, Canada
[40]Department of Clinical Laboratory, Chengdu Military General Hospital, Chengdu, China
[41]Department of Oncology-Pathology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Karolinska Institutet, Solna, Sweden
[42]Core Unit Bioinformatics, Berlin Institute of Health, Berlin, Germany
[43]Charité Universitätsmedizin Berlin, Berlin, Germany
[44]Department of Biochemistry Molecular Biology, Penn State University, University Park, United States
[45]Divisions of Molecular Pathology and Molecular Carcinogenesis, The Netherlands Cancer Institute, Amsterdam, The Netherlands
[46]Computational Biology and Bioinformatics, University of Southern California, Los Angeles, United States
[47]Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford, United Kingdom
[48]-
[49]Biological Sciences Division, Pacific Northwest National Laboratory, Richland, United States
[50]The James Hutton Institute, Dundee, United Kingdom
[51]Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen, Essen, Germany
[52]Northrop Grumman Corporation, Technology Services, Rockville, United States
[53]Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, Worcester, United States
[54]Department of Biochemistry, Molecular Biology and Biophysics (as contractor, not employee), University of Minnesota, Minneapolis, United States
[55]Department of Biology, Johns Hopkins University, Baltimore, United States
[56]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory, Richland, United States
[57]University of Colorado, Denver, United States
[58]Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, University of Leipzig, Leipzig, Germany
[59]Department of Chemistry Chemical Biology, Rutgers University, Piscataway, United States
[60]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, United

States
[61]Shanghai Institutes for Biological Sciences, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Shanghai China
[62]Department of Computer Science, Norwegian University of Science and Technology
[63]Department of Biochemistry and Molecular Biology, Penn State University, University Park, United States
[64]Massachusetts Institute of Technology, Cambridge, United States
[65]Lübeck Interdisciplinary Platform for Genome Analytics (LIGA), Institutes of Neurogenetics and Integrative Experimental Genomics, University of Lübeck, Lübeck, Germany
[66]Massachusetts General Hospital and Harvard Medical School, Boston, United States
[67]EMBL-EBI, Wellcome Genome Campus, Hinxton, Cambridgeshire, United Kingdom
[68]Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, United States
[69]German Cancer Research Center (DKFZ), Foundation under Public Law, Heidelberg, Germany
[70]School of Life Sciences, Arizona State University, Tempe, United States
[71]Daniel K. Inouye Center for Microbial Oceanography: Research and Education, Department of Oceanography, University of Hawaii, Honolulu, United States
[72]Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg Essen, Essen, Germany
[73]RCSB Protein Data Bank, San Diego Supercomputer Center, University of California, San Diego, United States
[74]Altius Institute for Biomedical Sciences, Seattle, United States
[75]Oncology Division, Department of Medicine, Washington University School of Medicine, St. Louis, United States
[76]McDonnell Genome Institute, Washington University School of Medicine, St. Louis, United States
[77]Dyliss - Dynamics, Logics and Inference for biological Systems and Sequences, Inria/IRISA, Campus Beaulieu, Rennes, France
[78]Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland
[79]Department of Molecular Medicine, Aarhus University Hospital, Aarhus, Denmark
[80]Departments of Biology and Computer Science, Johns Hopkins University, Baltimore, United States
[81]Department of Computer Science, Stony Brook University, Stony Brook, United States
[82]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, United States
[83]Department of Biology and Biological Engineering, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Chalmers University of Technology, Sweden
[84]Disease Systems Biology Program, Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Copenhagen, Denmark
[85]Center for Cancer Research, University of Melbourne, Melbourne, Australia
[86]Centre for Molecular Medicine Norway (NCMM), Nordic EMBL Partnership, University of Oslo, Oslo, Norway

[87]Massey University, Institute of Natural and Mathematical Sciences, North Shore City, New Zealand
[88]High Performance Computing, NYU Abu Dhabi, Abu Dhabi, United Arab Emirates
[89]Department of Molecular Evolution, Cell and Molecular Biology, Science for Life Laboratory, Biomedical Centre, Uppsala University, Uppsala, Sweden
[90]Evolutionary Genomics Lab, Research School of Biology, The Australian National University, Canberra, Australia
[91]Ecology and Evolution Unit, Okinawa Institute of Science and Technology Graduate University, Onna-son, Kunigami-gun, Okinawa, Japan
[92]ECCPS Bioinformatics Core Unit, Max Planck Institute for Heart and Lung Research, Bad Nauheim, Germany
[93]Department of Microbiology, School of Natural Sciences, National University of Ireland, Galway, Ireland Information and Computational Sciences, James Hutton Institute, Invergowrie, Scotland
[94]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Harvard T.H. Chan School of Public Health, Boston, United States
[95]Universitat Rovira i Virgili, Spanish Biomedical Research Center in Diabetes and Associated Metabolic Disorders (CIBERDEM), Reus Spain
[96]Applied Bioinformatics Laboratory, 2 Ravinia Drive, Suite 1200 Atlanta, GA 30346, United States
[97]Dip. di Informatica Sistemistica e Comunicazione, Univ. degli Studi di Milano-Bicocca, Milan, Italy
[98]Erasmus Medical Center, Rotterdam, The Netherlands
[99]Computational Biology Institute, Milken Institute School of Public Health, The George Washington University, Washington, D.C., United States
[100]Department of Microbiology, Immunology Tropical Medicine, The George Washington University School of Medicine and Health Sciences, Washington, D.C., United States
[101]Savitribai Phule Pune University, Pune, Maharashtra, India
[102]Boston Children's Hospital, Boston, United States
[103]Department of Computer Science and Software Engineering, Université Laval, Québec, Canada
[104]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, United Kingdom
[105]Department of Pathology, Memorial Sloan Kettering Cancer Center, New York, United States
[106]The McDonnell Genome Institute, Washington University, St. Louis, United States
[107]Princess Margaret Cancer Centre, Toronto, Canada
[108]Transcription and Chromatin Lab, Humanitas University, Rozzano, Italy
[109]ZBH - Center for Bioinformatics, MIN-Fakultät, Universität Hamburg, Hamburg, Germany
[110]Department of Computer Science, Computer science with specialisation in Bioinformatics, VIT University Vellore, India
[111]Cold Spring Harbor Laboratory, Cold Spring Harbor, United States
[112]Clermont Auvergne University, INRA, UNH, Human Nutrition Unit, PFEM, Metabolism Exploration Platform, MetaboHUB-Clermont, Clermont-Ferrand, France

[113]Department of Microbiology, University of Pennsylvania, United States
[114]Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic, Cleveland, United States
[115]Institute of Microbiology, Universitary Hospital of Lausanne, Switzerland
[116]Centrum Wiskunde and Informatica, Amsterdam, Netherlands
[117]Department of Biochemistry and Biophysics, Science for Life Laboratory, Stockholm University, Stockholm, Sweden
[118]CEA, LIST, Laboratory for data analysis and systems' intelligence, MetaboHUB, France
[119]Drexel University College of Medicine, Department of Microbiology and Immunology, Philadelphia, United States
[120]The Kirby Institute of Infection and Immunity, University of New South Wales, Sydney, Australia
[121]CyVerse, Bio5 institute, University of Arizona, Tucson, United States
[122]European Molecular Biology Laboratory (EMBL), Genomics Core Facility, Heidelberg, Germany
[123]New York University School of Medicine, New York City, United States
[124]Department of Molecular Genetics and Biology of Complex Diseases, Institute of Medical Research A Lanari-IDIM, University of Buenos Aires, National Scientific and Technical Research Council (CONICET), Ciudad Autónoma de Buenos Aires, Argentina.
[125]Institut de Génétique et de Biologie Moléculaire et Cellulaire, CNRS, Illkirch, France
[126]Epigenetic Regulation Unit, Pasteur Institute, Paris, France
[127]Department of Engineering, Roma Tre University, Rome, Italy
[128]Institute for Systems Analysis and Computer Science "Antonio Ruberti", National Research Council of Italy, Rome, Italy
[129]UFIT Research Computing, University of Florida, Gainesville, United States
[130]NSW Department of Primary Industries, Elizabeth Macarthur Agricultural Institute, Menangle, Australia
[131]Faculty of Veterinary Medicine, University of Calgary, Calgary, Canada
[132]Department of Biostatistics and Health Informatics, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, United Kingdom
[133]NIHR Biomedical Research Centre at South London and Maudsley NHS Foundation Trust and King's College London, United Kingdom
[134]Farr Institute of Health Informatics Research, UCL Institute of Health Informatics, University College London, London, United Kingdom
[135]Pathology Department, University of Michigan, Ann Arbor, United States
[136]GeneDx, Gaithersburg, United States
[137]Algorithms for reproducible bioinformatics, Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen
[138]Dana Farber Cancer Institute, Harvard Medical School, Boston, United States

October 21, 2017

[*]Co-first author
[†]To whom correspondence should be addressed.

Abstract

We present Bioconda (`https://bioconda.github.io`), a distribution of bioinformatics software for the lightweight, multi-platform and language-agnostic package manager, Conda. Currently, Bioconda offers a collection of over 2900 software tools, which are continuously maintained, updated, and extended by a growing global community of more than 200 contributors. Bioconda improves analysis reproducibility by allowing users to define isolated environments with defined software versions, all of which are easily installed and managed without administrative privileges.

Introduction

Thousands of new software tools have been released for bioinformatics in recent years, in a variety of programming languages. Accompanying this diversity of construction is an array of installation methods. Software written in C/C++ often has to be compiled manually for different hardware architectures and operating systems, with management left to the user or system administrator. Scripting languages usually deliver their own package management tools for installing, updating, and removing packages, though these are often limited in scope to packages written in the same scripting language, such that external dependencies (e.g., C libraries) have to be installed manually. Published scientific software often consists of simple collections of custom scripts distributed with textual descriptions of the manual steps required to use the software. New analyses often require novel combinations of multiple tools, and the heterogeneity of scientific software makes management of a software stack complicated and error-prone. Moreover, it inhibits reproducible science (Mesirov, 2010; Baker, 2016; Munafò et al., 2017) because it is hard to reproduce a software stack on different machines. System-wide deployment of software has traditionally been handled by adminstrators, but reproducibility often requires that the researcher (who is often not an expert in administration) 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. Conda normalizes software installations across language ecosystems by describing each software package with a *recipe* that defines meta-information and dependencies, as well as a *build script* that performs the steps necessary to build and install the software. Conda prepares and builds software packages within an isolated environment, transforming them into relocatable binaries. Conda packages can be built for all three major operating systems: Linux, macOS, and Windows. Importantly, installation and management of packages requires no administrative privileges, such that a researcher can control the available software tools regardless of the underlying infrastructure. Moreover, Conda obviates reliance on system-wide installation by allowing users to generate isolated software environments, within which versions and tools can be managed per-project, without generating conflicts or incompatibilies (see online methods). These environments support reproducibility, as they can can be rapidly exchanged via files that describe their installation state. Conda is tightly integrated into popular solutions for reproducible scientific data analysis like Galaxy (Afgan et al., 2016), bcbio-nextgen (`https://github.com/chapmanb/bcbio-nextgen`), and Snakemake (Köster and Rahmann, 2012). Finally, while Conda provides many commonly-used packages by default, it also allows users to optionally include additional repositories (termed *channels)* of packages that can be installed.

Results

In order to unlock the benefits of Conda for the life sciences, the Bioconda project was founded in 2015. The mission of Bioconda is to make bioinformatics software easily installable and manageable via the Conda package manager. Via its channel for the Conda package manager, Bioconda currently provides over 2500 software packages for Linux and macOS. Development is driven by an open community of over 200 international scientists. In the prior two years, package count and the number of contributors have increased linearly, on average, with no sign of saturation (Fig. 1a,b). The barrier to entry is low, requiring a willingness to participate and adherence to community guidelines. Many software developers contribute recipes for their own tools, and many Bioconda contributors are invested in the project as they are also users of Conda and Bioconda. Bioconda provides packages from various language ecosystems like Python, R (CRAN and Bioconductor), Perl, Haskell, as well as a plethora of C/C++ programs (Fig. 1c). Many of these packages have complex dependency structures that require various manual steps to install when not relying on a package manager like Conda (Fig. 2a, Online Methods). With over 5.9 million downloads, the service has become a backbone of bioinformatics infrastructure (Fig. 1d). Bioconda is complemented by the conda-forge project (`https://conda-forge.github.io`), which hosts software not specifically related to the biological sciences. The two projects collaborate closely, and the Bioconda team maintains over 500 packages hosted by conda-forge. Among all currently available distributions of bioinformatics software, Bioconda is by far the most comprehensive, while being among the youngest (Fig. 2d).

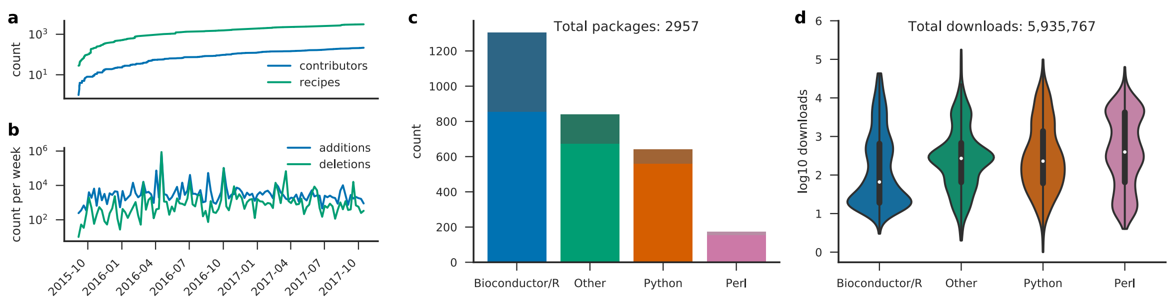

Figure 1: Bioconda development and usage since the beginning of the project (state: October 2017). (a) contributing authors and added recipes over time. (b) code line additions and deletions per week. (c) package count per language ecosystem (saturated colors on bottom represent explicitly life science related packages). (d) total downloads per language ecosystem. The term “other” entails all recipes that do not fall into one of the specific Note that a subset of packages that started in Bioconda have since been migrated to the more appropriate, general-purpose conda-forge channel. Older versions of such packages still reside in the Bioconda channel, and as such are included in the recipe count (a) and download count (d).categories.

To ensure reliable maintenance of such numbers of packages, we use a semi-automatic, agent-assisted development workflow (Fig. 2b). All Bioconda recipes are hosted in a GitHub repository (`https://github.com/bioconda/bioconda-recipes`). Both the addition of new recipes and the update of existing recipes in Bioconda is handled via *pull requests*. Thereby, a modified version of one or more recipes is compared against the current state of Bioconda. Once a pull request arrives, our infrastructure performs several automatic checks. Problems discovered in any step are reported to the contributor and further progress is blocked until they are resolved. First, the modified recipes are checked for syntactic anti-patterns, i.e., formulations that are syntactically correct but bad style (termed *linting*). Second, the modified recipes are built on Linux and macOS, via a cloud based, free-of-charge service (`https://travis-ci.org`). Successfully built recipes are tested (e.g., by running the generated executable). Since Bioconda packages must be able to run on any supported system, it is important to check that the built packages do not rely on particular elements from the build environment. Therefore, testing happens in two stages: (a) test cases are executed in the build environment (b) test cases are executed in a minimal Docker (`https://docker.com`) container which purposefully lacks all non-common system libraries (hence a dependency that is not explicitly defined will lead to a failure). Once the *build* and *test* steps have succeeded, a member of the Bioconda team reviews the proposed changes and, if acceptable, merges the modifications into the official repository. Upon merging, the recipes are built again and uploaded to the hosted Bioconda channel (`https://anaconda.org/bioconda`), where they become available via the Conda package manager. When a Bioconda package is updated to a new version, older builds are generally preserved, and recipes for multiple older versions may be maintained in the Bioconda repository. The usual turnaround time of above workflow is fast (Fig. 2d). 61% of the pull requests are merged within 5 hours. Of those, 36% are even merged within 1 hour. Only 18% of the pull requests need more than a day. Hence, publishing software in Bioconda or updating already existing packages can be accomplished typically within minutes to a few hours.

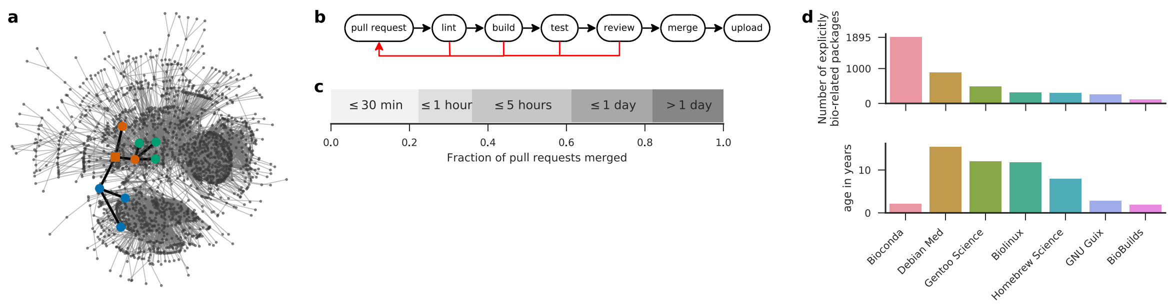

Figure 2: Dependency structure, workflow, comparison with other resources, and turnaround time. (a) largest connected component of directed acyclic graph of Bioconda packages (nodes) and dependencies (edges). Highlighted is the induced subgraph of the CNVkit (Talevich et al., 2016) package and it's dependencies (node coloring as defined in Fig. 1, squared node represents CNVkit). (b) Github based development workflow: a contributor provides a pull request that undergoes several build and test steps, followed by a human review. If any of these checks does not succeed, the contributor can update the pull request accordingly. Once all steps have passed, the changes can be merged. (c) Turnaround time from submission to merge of pull requests in Bioconda. (d) 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`), Biolinux (Field et al., 2006), 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 age since the first release or commit.

Reproducible software management and distribution is enhanced by other current technologies. Conda integrates itself well with environment modules (`http://modules.sourceforge.net/`), a technology used nearly universally across HPC systems. An administrator can use Conda to easily define software stacks for multiple labs and project-specific configurations. Popularized by Docker, containers provide another way to publish an entire software stack, down to the operating system. They provide greater isolation and control over the environment a software is executed in, at the expense of some customizability. Conda complements container-based approaches. Where flexibility is needed, Conda packages can be used and combined directly. Where the uniformity of containers is required, Conda can be used to build images without having to reproduce the nuanced installation steps that would ordinarily be required to build and install an application within an image. In fact, for each Bioconda package, the build system automatically builds a minimal Docker image containing that package, which is subsequently uploaded and made available via the Biocontainers project (da Veiga Leprevost et al., 2017). As a consequence, every built Bioconda package is available not only for installation via Conda, but also directly available as a container via Docker, Rkt (https://coreos.com/rkt), and Singularity (Kurtzer et al., 2017).

Discussion

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 and coordinate closely with other related projects such as conda-forge and Biocontainers. Bioconda offers a comprehensive resource of thousands of software libraries and tools that is maintained by hundreds of international scientists. With almost six million downloads so far, Bioconda packages have been well received by the community. We invite everybody to participate in reaching the goal of a central, comprehensive, and language agnostic collection of easily installable software by maintaining existing or publishing new software in Bioconda.

Funding

The Bioconda project has received support from Anaconda, Inc., Austin, TX, USA, in the form of expanded storage for Bioconda packages on their hosting service (`https://anaconda.org`). Further, the project has been granted extended build times from Travis CI, GmbH (https://travis-ci.com). The Bioconda community also would like to thank ELIXIR (https://www.elixir-europe.org) for their constant support.

Acknowledgments

We thank the participants of various hackathons (e.g., the GalaxyP contribution fest) for porting numerous packages to Bioconda.

Online Methods

Security Considerations

Using Bioconda as a service to obtain packages for local installation entails trusting that (a) the provided software itself is not harmful and (b) it has not been modified in a harmful way. Ensuring (a) is up to the user. In contrast, (b) is handled by our workflow. First, source code or binary files defined in recipes are checked for integrity via MD5 or SHA256 hash values. Second, all review and testing steps are enforced via the Github interface. This way, it is guaranteed that all packages have been tested automatically and reviewed by a human being. Third, all changes to the repository of recipes are publicly tracked, and all build and test steps are transparently visible to the user. Finally, the automatic parts of the development workflow are implemented in the open-source software *bioconda-utils* (`https://github.com/bioconda/bioconda-utils`). In the future, we will further explore the possibility to sign packages cryptographically.

Software management with Conda

Via the Conda package manager, installing software from Bioconda becomes very simple. In the following, we describe the basic functionality assuming that the user has access to a Linux or macOS terminal. After installing Conda, the first step is to set up the Bioconda channel via:

```
$ conda config --add channels conda-forge
$ conda config --add channels bioconda
```

Now, all Bioconda packages are visible to the Conda package manager. For example, the software CNVkit (Talevich et al., 2016), can be searched for with

```
$ conda search cnvkit
```

in order to check if and in which versions it is available. It can be installed with:

```
$ conda install cnvkit
```

CNVkit needs various dependencies from Python and R, which would otherwise have to be installed in separate manual steps (Fig. 2a). Furthermore, Conda enables updating and removing all these dependencies via one unified interface. A key value of Conda is the ability to define isolated, shareable software environments. This can happen ad-hoc, or via YAML (https://yaml.org) files. For example, the following defines an environment consisting of Salmon (Patro et al., 2017) and DESeq2 (Love et al., 2014):

```
channels:
  - bioconda
  - conda-forge
  - defaults
dependencies:
  - bioconductor-deseq2 =1.16.1
  - salmon =0.8.2
  - r-base =3.4.1
```

Given that the environment is stored in a file `env.yaml`, it can be created with the name `my-env` via the command:

```
$ conda env create --name my-env --file env.yaml
```

Then, in order to use the environment it can be activated with:

```
$ source activate my-env
```

Within the environment R, Salmon and DESeq2 are available in exactly the defined versions. For example, salmon can be executed with:

```
$ salmon --help
```

It is possible to modify an existing environment by using `conda update`, `conda install` and `conda remove`. For example, we could add a particular version of Kallisto (Bray et al., 2016) and update Salmon to the latest available version with:

```
$ conda install kallisto=0.43.1
$ conda update salmon
```

Finally, the environment can be deactivated again with:

```
$ source deactivate
```

How isolated software environments enable reproducible research

With isolated software environments as shown above, it is possible to define an exact version for each package. This increases reproducibility by eliminating differences due to implementation changes. Note that above we also pin an R version, although the latest compatible one would also be automatically installed without mentioning it. To further increase reproducibility, this pattern can be extended to all dependencies of DESeq2 and Salmon and recursively down to basic system libraries like zlib and boost (https://www.boost.org). Environments are isolated from the rest of the system, while still allowing interaction with it: e.g. tools inside the environment are preferred over system tools, while system tools that are not available from within the environment can still be used. Conda also supports the automatic creation of environment definitions from already existing environments. This allows to rapidly explore the needed combination of packages before it is finalized into an environment definition. When used with workflow management systems like Galaxy (Afgan et al., 2016), bcbio-nextgen (https://github.com/chapmanb/bcbio-nextgen), and Snakemake (Köster and Rahmann, 2012) that interact directly with Conda, a data analysis can be shipped and deployed in a fully reproducible way, from description and automatic execution of every analysis step down to the description and automatic installation of any required software.

References

E Afgan, D Baker, den Beek M van, D Blankenberg, D Bouvier, M Čech, J Chilton, D Clements, N Coraor, C Eberhard, B Grüning, A Guerler, J Hillman-Jackson, Kuster G Von, E Rasche, N Soranzo, N Turaga, J Taylor, A Nekrutenko, and J Goecks. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res*, 44:W3–W10, Jul 2016. doi: 10.1093/nar/gkw343. URL https://doi.org/10.1093/nar/gkw343.

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, may 2016. doi: 10.1038/533452a. URL https://doi.org/10.1038%2F533452a.

NL Bray, H Pimentel, P Melsted, and L Pachter. Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol*, 34:525–7, May 2016. doi: 10.1038/nbt.3519. URL https://doi.org/10.1038/nbt.3519.

F da Veiga Leprevost, BA Grüning, Aflitos S Alves, HL Röst, J Uszkoreit, H Barsnes, M Vaudel, P Moreno, L Gatto, J Weber, M Bai, RC Jimenez, T Sachsenberg, J Pfeuffer, Alvarez R Vera, J Griss, AI Nesvizhskii, and Y Perez-Riverol. BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, 33:2580–2582, Aug 2017. doi: 10.1093/bioinformatics/btx192. URL https://doi.org/10.1093/bioinformatics/btx192.

Dawn Field, Bela Tiwari, Tim Booth, Stewart Houten, Dan Swan, Nicolas Bertrand, and Milo Thurston. Open software for biologists: from famine to feast. *Nature Biotechnology*, 24(7):801–803, jul 2006. doi: 10.1038/nbt0706-801. URL https://doi.org/10.1038%2Fnbt0706-801.

GM Kurtzer, V Sochat, and MW Bauer. Singularity: Scientific containers for mobility of compute. *PLoS One*, 12:e0177459, 2017. doi: 10.1371/journal.pone.0177459. URL https://doi.org/10.1371/journal.pone.0177459.

J Köster and S Rahmann. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics*, 28:2520–2, Oct 2012. doi: 10.1093/bioinformatics/bts480. URL https://doi.org/10.1093/bioinformatics/bts480.

MI Love, W Huber, and S Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*, 15:550, 2014. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

J. P. Mesirov. Accessible Reproducible Research. *Science*, 327(5964):415–416, jan 2010. doi: 10.1126/science.1179653. URL https://doi.org/10.1126%2Fscience.1179653.

Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, 1(1):0021, jan 2017. doi: 10.1038/s41562-016-0021. URL https://doi.org/10.1038%2Fs41562-016-0021.

R Patro, G Duggal, MI Love, RA Irizarry, and C Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*, 14:417–419, Apr 2017. doi: 10.1038/nmeth.4197. URL https://doi.org/10.1038/nmeth.4197.

E Talevich, AH Shain, T Botton, and BC Bastian. CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing. *PLoS Comput Biol*, 12:e1004873, Apr 2016. doi: 10.1371/journal.pcbi.1004873. URL https://doi.org/10.1371/journal.pcbi.1004873.