



Check for updates

SOFTWARE TOOL ARTICLE

BiocPkgTools: Toolkit for mining the *Bioconductor* package ecosystem

[version 1; peer review: 2 approved, 1 approved with reservations]

Shian Su ¹, Vincent J. Carey ², Lori Shepherd³, Matthew E. Ritchie ¹, Martin T. Morgan³, Sean Davis⁴

¹Molecular Medicine Division, The Walter and Eliza Hall Institute of Medical Research, Parkville, Australia²Channing Division of Network Medicine, Brigham and Women's Hospital, Boston, MA, USA³Roswell Park Cancer Institute, Buffalo, NY, USA⁴Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, MD, USA

V1 First published: 29 May 2019, 8:752
<https://doi.org/10.12688/f1000research.19410.1>

Latest published: 29 May 2019, 8:752
<https://doi.org/10.12688/f1000research.19410.1>

Abstract

Motivation: The Bioconductor project, a large collection of open source software for the comprehension of large-scale biological data, continues to grow with new packages added each week, motivating the development of software tools focused on exposing package metadata to developers and users. The resulting BiocPkgTools package facilitates access to extensive metadata in computable form covering the Bioconductor package ecosystem, facilitating downstream applications such as custom reporting, data and text mining of Bioconductor package text descriptions, graph analytics over package dependencies, and custom search approaches.

Results: The BiocPkgTools package has been incorporated into the Bioconductor project, installs using standard procedures, and runs on any system supporting R. It provides functions to load detailed package metadata, longitudinal package download statistics, package dependencies, and Bioconductor build reports, all in "tidy data" form. BiocPkgTools can convert from tidy data structures to graph structures, enabling graph-based analytics and visualization. An end-user-friendly graphical package explorer aids in task-centric package discovery. Full documentation and example use cases are included.

Availability: The BiocPkgTools software and complete documentation are available from Bioconductor (

<https://bioconductor.org/packages/BiocPkgTools>).

Keywords

bioinformatics, r, bioconductor, software, reproducible research

Open Peer Review

Approval Status

	1	2	3
version 1			
29 May 2019	view	view	view

1. **Simina M. Boca** , Georgetown University Medical Center, Washington, USA

2. **Mike Smith** , European Molecular Biology Laboratory, Heidelberg, Germany

3. **Henrik Bengtsson**, University of California, San Francisco, San Francisco, USA

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the **Bioinformatics** gateway.



This article is included in the **Bioconductor** gateway.



This article is included in the **RPackage** gateway.

Corresponding author: Sean Davis (seandavi@gmail.com)

Author roles: **Su S:** Conceptualization, Data Curation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Carey VJ:** Conceptualization, Funding Acquisition, Methodology, Software, Supervision, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Shepherd L:** Software, Writing – Review & Editing; **Ritchie ME:** Funding Acquisition, Methodology, Project Administration, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Morgan MT:** Funding Acquisition, Methodology, Project Administration, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Davis S:** Conceptualization, Data Curation, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under award number U41HG004059, the National Cancer Institute of the National Institutes of Health under award numbers U24CA180996 and U01CA214846-02, and the Center for Cancer Research, part of the Intramural Research Program at the National Cancer Institute at the National Institutes of Health. Part of this work was performed on behalf of the SOUND Consortium and funded under the EU H2020 Personalizing Health and Care Program, Action contract number 633974.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2019 Su S *et al.* This is an open access article distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The author(s) is/are employees of the US Government and therefore domestic copyright protection in USA does not apply to this work. The work may be protected under the copyright laws of other jurisdictions when used in those jurisdictions.

How to cite this article: Su S, Carey VJ, Shepherd L *et al.* **BiocPkgTools: Toolkit for mining the Bioconductor package ecosystem [version 1; peer review: 2 approved, 1 approved with reservations]** F1000Research 2019, 8:752
<https://doi.org/10.12688/f1000research.19410.1>

First published: 29 May 2019, 8:752 <https://doi.org/10.12688/f1000research.19410.1>

Introduction

Bioconductor is an open source software project (comprising 1741 individual analysis packages) and community for the analysis and comprehension of large-scale biological data. Newly submitted software packages undergo a technical review to ensure that best practices and *Bioconductor* coding conventions are followed. The project maintains an automated build system that ensures that packages in the *Bioconductor* project are compiled and built successfully and pass basic checks. Package downloads are tracked and aggregated by package and month, longitudinally. Finally, package details such as title, description, version, author, and dependencies on other R packages are compiled based on package metadata.

The current size and growth of the *Bioconductor* project suggests that there is merit in exposing computable forms of the metadata describing the *Bioconductor* package ecosystem. To that end, we developed a small suite of tools, BiocPkgTools, to provide easy access to project details such as download statistics, bulk package metadata, and package build status. Developers, project leaders, open source software researchers, and *Bioconductor* end users can build on the availability of these data for applications such as custom reporting, dependency graph analytics, package filtering, and text mining.

Features and usage

The core functionality of BiocPkgTools is to expose *Bioconductor* project and package metadata as tidy data¹ objects (Figure 1). The data presented by the package are accessed directly from online resources available from *Bioconductor*. As such, the package relies on web connectivity and collects the most recent data. Installation instructions are detailed on the package website.

Package functionality can be roughly divided into data access, data presentation, and graph/network functionality. See Table 1 for an overview.

After installing BiocPkgTools, the biocDownloadStats function can generate a tidy data structure summarizing monthly download statistics (both total and unique IP addresses) for all *Bioconductor* packages.

```
library(BiocPkgTools)
dlstats = biocDownloadStats()
head(dlstats, 3)
```

```
## # A tibble: 3 x 6
##   Package Year Month Nb_of_distinct_IPs Nb_of_downloads repo
##   <fct>   <int> <fct>          <int>          <int> <chr>
## 1 ABarray 2018 Jan          117            150 Software
## 2 ABarray 2018 Feb           97            125 Software
## 3 ABarray 2018 Mar          102            121 Software
```

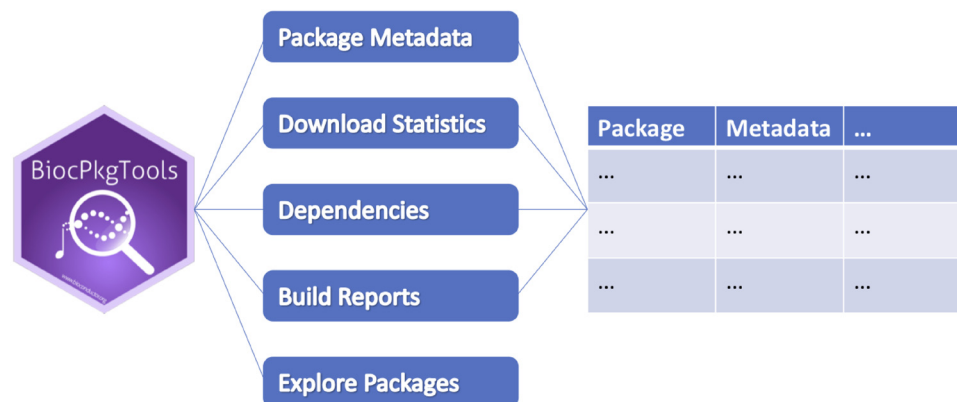


Figure 1. Schematic overview of the BiocPkgtools package. BiocPkgTools can access and transform web-accessible resources including package metadata, download statistics, dependencies between packages, and updated *Bioconductor* build report status to “tidy data” reports that can be manipulated using standard R tools. Interactive package exploration is also available.

Table 1. Main package functions and descriptions.

Name	Functionality
biocPkgList	Package details including description, author and maintainer, dependencies, URLs, bug report mechanism
biocDownloadStats	Monthly download statistics for all packages
biocbuildReport	<i>Bioconductor</i> build report for all packages and systems
biocExplore	Interactive, browsable “bubble plot” of <i>Bioconductor</i> packages and details
problemPage	Interactive, customized build report for an individual package author
buildPkgDependencyDataFrame	Package dependencies as data frame
buildPkgDependencyIgraph	Package dependencies as a graph ²
inducedSubgraphByPkgs	Create a minimal subgraph of <i>Bioconductor</i> dependencies based on specific packages
subgraphByDegree	Create a subgraph of all packages within a given degree of a single package

The `biocBuildReport` function gathers information from the *Bioconductor* [build report site](#) and can be used, for example, to summarize the “build status” for all *Bioconductor* packages.

```
buildrep = biocBuildReport(version = "3.9")
table(buildrep$stage, buildrep$result)

##
##          ERROR    OK skipped TIMEOUT WARNINGS
##  buildbin      2 3352      70         0         0
##  buildsrc     93 5057         0         5         0
##  checksrc     57 4181      98         8       811
##  install      39 5116         0         0         0
```

These data are useful to developers to track the health of their software either programmatically or via a searchable, sortable table from the `problemPage` function.

As an alternative to basic web browser search and the *Bioconductor* online software list, the `biocExplore` function offers interactive and graphical approach to package browsing (see [Figure 2](#)). The `biocExplore` widget allows browsing packages under different `biocViews`, *Bioconductor*’s software category tags. This interactively visualises the relative number of downloads each package has under different `biocViews`, allowing users to quickly determine which packages are most commonly used for different analysis tasks.

The *Bioconductor* package ecosystem is, by design, highly interconnected via package dependencies. Several functions in the `BiocPkgTools` package provide examples of package dependency graph creation and visualization. [Figure 3](#) displays packages within one degree of dependency relationship of the `GEOquery` package.

Implementation

`BiocPkgTools` is implemented as a standard R package and hosted in the *Bioconductor* repository. All functions are documented and include examples. An included tutorial (vignette) demonstrates features and capabilities.

Discussion

The `BiocPkgTools` package comprises a set of functions for accessing software metadata from the growing collection of *Bioconductor* packages. For software developers, this metadata can be useful for tracking package build status and the health of package dependencies. Easy access to descriptive package metadata for all *Bioconductor* software resources can empower researchers or users interested in text mining, custom package search, or analysis of the existing software ecosystem. `BiocPkgTools` can provide easy access to metrics of *Bioconductor* software usage that are increasingly being incorporated into funding and promotion decisions.

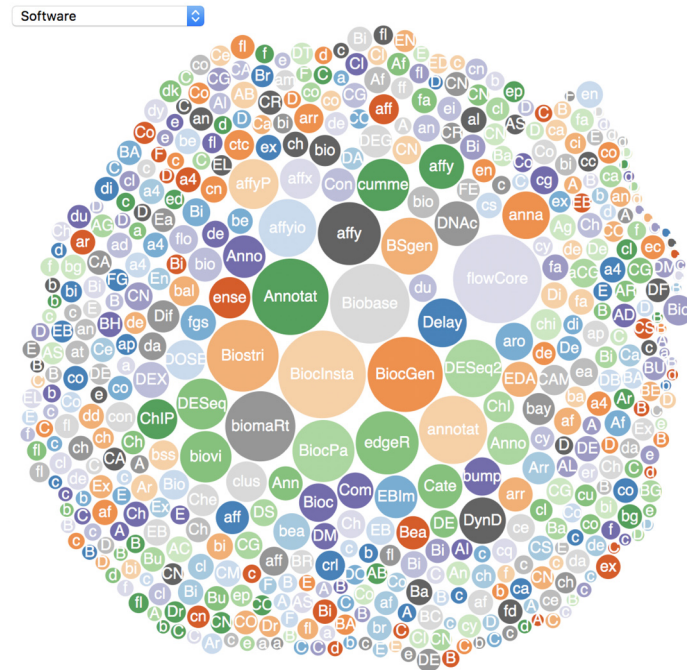


Figure 2. The `biocExplore` function opens an interactive web application that allows users to select focused groups of *Bioconductor* packages to view as a bubble plot. Bubbles are sized based on download statistics. Hovering over a bubble will give download number while clicking on a bubble will pop up a package details page, including a link to the package landing page.

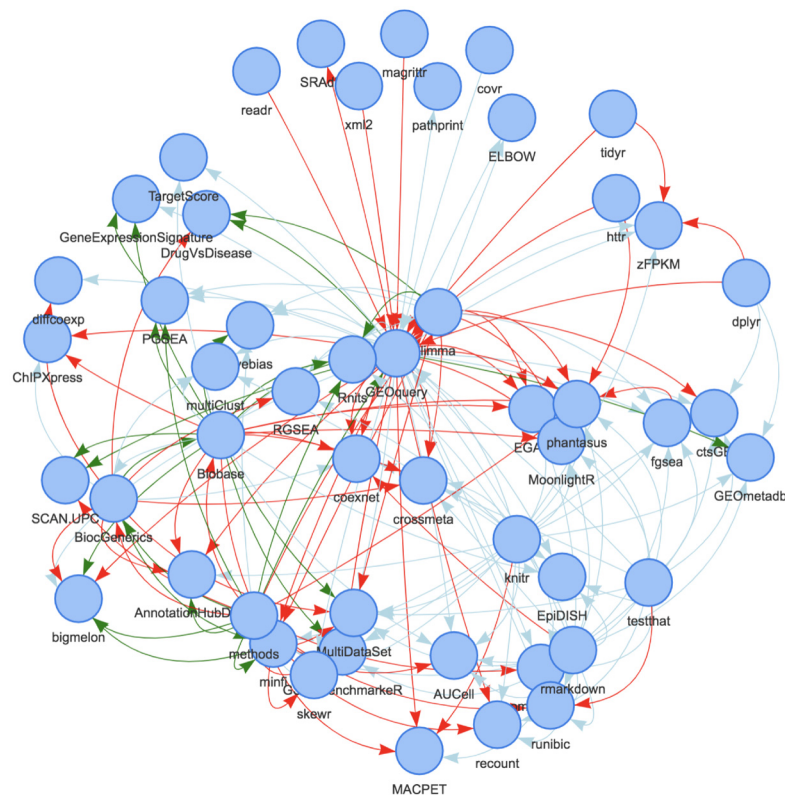


Figure 3. The `subgraphByDegree` function builds a data visualization of dependencies between all packages within one degree of the `GEOquery` package³ using the `visNetwork` package³. Links are colored based on type (Suggests [light blue], Depends [green], and Imports [red]) and arrows point to the “dependent” package.

Data availability

All data accessed and used by the BiocPkgTools package are publicly available and are updated regularly at the *Bioconductor* project.

Software availability

- Software available from: <https://bioconductor.org/packages/BiocPkgTools>
- Source code available from: <https://github.com/seandavi/BiocPkgTools>
- Archived source code as at time of publication: <https://doi.org/doi:10.18129/B9.bioc.BiocPkgTools>⁴
- License: MIT License

Grant information

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under award number U41HG004059, the National Cancer Institute of the National Institutes of Health under award numbers U24CA180996 and U01CA214846-02, and the Center for Cancer Research, part of the Intramural Research Program at the National Cancer Institute at the National Institutes of Health. Part of this work was performed on behalf of the SOUND Consortium and funded under the EU H2020 Personalizing Health and Care Program, Action contract number 633974.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Wickham H: Tidy data. <i>Journal of Statistical Software, Articles</i>. 2014; 59(10): 1–23.
Publisher Full Text 2. Csardi G, Nepusz T: The igraph software package for complex network research. <i>InterJournal</i>. 2006; Complex Systems, 1695.
Reference Source | <ol style="list-style-type: none"> 3. Almende B, Thieurmél B, Robert T: visNetwork: Network Visualization using 'vis.js' Library. R package version 2.0.4. 2018. 4. Su S, Davis S: BiocPkgTools: Collection of simple tools for learning about Bioc Packages. R package version 1.2.0, 2019.
Reference Source |
|--|---|

Open Peer Review

Current Peer Review Status:   

Version 1

Reviewer Report 21 June 2019

<https://doi.org/10.5256/f1000research.21277.r49217>

© 2019 Bengtsson H. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Henrik Bengtsson

Division of Bioinformatics, Department of Epidemiology and Biostatistics, University of California, San Francisco, San Francisco, CA, USA

This article presents the BiocPkgTools package, which provides an R API to the various package metadata that is available mostly in human-readable formats on the <https://bioconductor.org/> website. By providing an R API for accessing package metadata, the authors argue that "data mining and value-added functionality such as package searching, text mining, and analytics on packages" may follow.

I believe that this package will add value to the Bioconductor community and beyond. It is likely that the package will lower the threshold for doing "data mining" on Bioconductor packages. The reports that are based on these metadata and produced by this package are likely to inspire others to produce other types of reports and interactive tools.

Having said, I do think there is room for some immediate improvements to the article, which might also spill over to the package itself.

Major:

The role of the package:

- It is not clear whether this package is to be considered a Bioconductor core package or a user-contributed package. That can only be inferred/guessed from authors list and possible from the package name.
- If it is an official core Bioconductor package (similar to BiocManager) supported and maintained by the Bioconductor Team, then I think it would be of value to make that explicit. This would change the expectation on the package and its long-term support, e.g. will it break or not when the Bioconductor website changes, what should be documented as part of the package and what should be documented on the Bioconductor website., etc. The questions in the following section illustrate this.

Description on the metadata:

- There is no explicit reference to the source data, e.g. is this package using a public or a private Bioconductor Online API to gather the data. Is this API, or URLs official and stable, or is this package meant to play that role?
- Is there another reference from where one can learn more about how the data is collected? Are data from other Bioconductor mirrors included in the download stats?
- The download stats do not contain information on the package version, which is mentioned in the package vignette. However, it's not clear whether it is only downloads from the current release branch that are counted or not. For instance, if I download a package for a legacy Bioconductor release version or the current developer version, will that add to the data?

Minor/trivial:

API:

- Given the package title 'Collection of simple tools for learning about Bioc Packages' and description, the `biocBuildEmail()` function seems to be an odd-one-out.

Typos/spelling:

- Section 'Introduction': "a open source software" -> "an open-source software".
- Section 'Features and usage': missing "the" in "to expose [the] Bioconductor project".
- Section 'Features and usage': "software cate[r]gory tags".
- Section 'Discussion': ... Bioconductor sof[t]ware...
- Table 1: `biocbuildReport` -> `biocBuildReport`.
- Table 1 caption: "all Bioconductor pac[a]kages".
- Grammar in Section 'Introduction': Should "suggest" be used instead of "suggests" in "The current size and growth of the Bioconductor project suggests"?
- In addition, ``spelling::spell_check_package()`` on the package itself reveals several mistakes.
- Nomenclature: Bioconductor is sometimes referred to as 'bioconductor' (lower case) or just 'bioc' and 'Bioc' (the latter is even used in the package title).

Figures and tables:

- Figure 1: It's not clear what the different parts of this figure are. As I understand it, the left part (the hex logo) represents the package itself, and the middle part (five blocks) represents the "web-accessible resources" that the package queries. What does the table and the edges in the right part represent? Is it meant to illustrate that the package pulls data from 5 different sources and joins them into a single table? The caption also tries to mention "Interactive package exploration is also available" which makes it harder to

understand the figure. Maybe it's the ordering of these three parts are confusing. Maybe it would be more clear if the edges from the hex logo to the web resources would be dropped. Maybe it would help if the empty table could be populated with something else than just "..." entries. Alternative, maybe something like [this](#) would clarify the flow of the data?

Package/examples:

- Section 'Implementation': "All functions are documented and include examples." Maybe rephrase so it doesn't sound that there are examples for **all** functions, which is not the case.
- Coding Convention: The package vignette and examples, and the example code snippets in the article, use `a = b`` for assignments rather than `a <- b`` that is the recommended coding style for Bioconductor, cf. <https://www.bioconductor.org/developers/how-to/coding-style/>.
- EXAMPLES: The presented output of the `dIstats = biocDownloadStats()` example is outdated; it shows other fields/columns than what the current BiocPkgTools 1.2.0 outputs.
- In addition to the above, I've identified other package-specific issues that I reported directly to the package's issue tracker (<https://github.com/seandavi/BiocPkgTools/issues/34>).

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Statistic, Bioinformatics, Computer Science, Scientific Computations

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 19 June 2019

<https://doi.org/10.5256/f1000research.21277.r49235>

© 2019 Smith M. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Mike Smith 

Genome Biology Unit, European Molecular Biology Laboratory, Heidelberg, Germany

In this paper Su et al present BiocPkgTools, an R package that provides programmatic access to metadata about software in the Bioconductor project. The package is available from Bioconductor and the source code can be easily viewed on Github. The metadata it can access includes download numbers, package 'health' status, topic categories, and software dependencies.

Since the software forms at least as large a part of this publication as the article, I have tried to review both.

Paper:

The paper is succinct and to the point. In some instances there is an implied understanding of R package development and the Bioconductor build system. For example, in the code block presented to demonstrate the biocBuildReport function, the row names e.g. buildbin, checksrc etc are probably fairly meaningless to someone unfamiliar with Bioconductor's continuous integration platform. It's fair to say most interested readers will already be Bioconductor package developers, but a caption linking this back to the build system discussion in the introduction would add clarity.

Similarly the biocViews concept is not fully described, but forms a key part of one of the packages main functions: biocExplore(). Expanding a little on how these terms are assigned to packages (i.e. mostly but not always by the package authors) might make it clearer to users why some values return unexpected results e.g. 'Bioinformatics' only shows me 15 packages presumably because most package authors see this as redundant, although 'Software' feels similarly obvious yet returns a huge number of packages as this is assigned by Bioconductor itself.

The motivation behind providing programmatic access to build reports and download statistics is presented in the text. However, it would be nice for the authors to expand upon what they feel the use cases for the package dependency graphs may be. They look cool, but based on the paper content it's not immediately obvious to me where they might be used. The discussion highlights the fact that download metrics etc are gaining traction as a measure factored into funding decisions, and one thing that springs to mind is that a similar point can be made visually for software infrastructure that has many downstream dependents. My efforts to create a ``subgraphByDegree()`` with degree greater than one (to view a larger software stack) creates a graph too large to render, so an example of how to visualise this (i.e. show all downstream dependsOnMe packages in a tree) would be a great addition to either the paper or the package vignette.

Minor points:

I recommend providing a link to the BiocPkgTools landing page after the sentence "Installation instructions are detailed on the package website."

Indicate that the code to produce Figure 3 can be found in the package vignette (or include it as supplementary material here). Since it is quite a bit more than a one-liner to produce Figure 3 I think it would be beneficial to point readers to the code.

Perhaps I am mistaken, but I can find no mention in the paper or package of what the colours represent in the bubble plots produced by biocExplore(). They definitely feel like they should mean something.

I would consider rephrasing the sentence "This interactively visualises the relative number of downloads each package has under different biocViews". To me this reads like the plot is somehow visualising how relevant the package is to the selected view, when really it is just filtering the package list based on the biocView category.

Typos (given in context, specific [word] highlighted):

- for all Bioconductor [pacakages]
- software [catergory] tags
- Bioconductor [software] usage
- bioc[b]uildReport (Table 1)

Software:

In general the software is well written with a comprehensive vignette providing more extensive examples than seen in the paper. However I did encounter a few issues when testing aspects of the package:

- The function firstInBioc() fails with 'Error in desc(Date) : could not find function "desc"'. Similarly, the function inducedSubgraphByPkgs() fails with 'Error in V(g2) <- `*vtmp*` : could not find function "V<-"'.
 - I think in both cases the relevant functions need to be added as imports in the NAMESPACE. These work in the package examples as additional libraries are loaded first.
- The manual page for subgraphByDegree() is slightly confusing. The argument pkg takes a vector of length one, but the description field reads like multiple packages can be supplied.
- Why is the function get_bioc_data() in snake_case, while all other functions are named with camelCase?
- I wonder if the function biocBuildEmail() should really be exported. This seems very specific to the Bioconductor core team, but is very similarly named to biocBuildReport() which is one of the primary data getters and will be widely used. Also, the 'Usage' section of the biocBuildEmail function references the function .getTemplatePath() which is not exported, perhaps reflecting the niche nature of this function.
- It's not clear to me what the difference is between an 'induced subgraph' and a 'regular subgraph' in the functions inducedSubgraphByPkgs() vs subgraphByDegree() and I couldn't find an explanation in the package documentation. Perhaps this is just a lack of knowledge about graphs on my part, but it's information I looked for based on the different function names.

- Are all of the packages listed in the Suggests field of the DESCRIPTION file used? I can't find mention of tm, SnowballC

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: I am a co-author with VJC and MTM on <https://doi.org/10.1101/590562>

Reviewer Expertise: Software development, Bioinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 10 June 2019

<https://doi.org/10.5256/f1000research.21277.r49219>

© 2019 Boca S. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Simina M. Boca 

Innovation Center for Biomedical Informatics, Department of Oncology, Georgetown University Medical Center, Washington, DC, USA

This article presents the new Bioconductor package, BiocPkgTools, which allows users to obtain various statistics about Bioconductor packages, including the number of downloads by month, the build status, and the package dependencies.

Overall, this seems to be a very useful package! I'm excited about using it as a way of measuring the use of my own package and perhaps doing some other exploratory analyses on the

Bioconductor ecosystem. I was able to load and run everything and obtain similar results to those presented (slight differences due to running it at a different time) after reading this article and the package vignette.

I believe the following minor edits could improve the presentation:

1. Provide the code for generating Figures 2 & 3, even though it is in the vignette. This would be helpful from a reproducibility perspective.
2. Specify how often the data presented are updated. Presumably this is done almost immediately, given that the build status is included. It would also be interesting to give an idea of how this is done.
3. Specify in the output/help page that the filtering in the bubble chart obtained by running `biocExplore()` is done by `biocViews` (this can be a bit confusing since, for example, under "Bioinformatics" only a few packages are selected, but that is not something you can fix, since it's up to the authors to include the appropriate `biocViews`).
4. Specify that the stats downloaded from `biocDownloadStats()` are the total number of downloads, not the number of downloads from distinct IPs, as the Bioconductor package page provides both.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: biostatistics, bioinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research