PHS 398 Research Plan

OMB Number: 0925-0001
Expiration Date: 03/31/2020

Introduction	
1. Introduction to Application (for Resubmission and Revision applications)	
Research Plan Section	
2. Specific Aims	Specific_Aims_FF.pdf
3. Research Strategy*	Research_Strategy_FF.pdf
4. Progress Report Publication List	
Other Research Plan Section	
5. Vertebrate Animals	
6. Select Agent Research	
7. Multiple PD/PI Leadership Plan	Multiple_PI_Plan_FINAL.pdf
8. Consortium/Contractual Arrangements	Consortium_contractual_agreement_text.pdf
9. Letters of Support	Combined_LOS_NOBOOK.pdf
10. Resource Sharing Plan(s)	Resource_Sharing_Plan_FINAL.pdf
11. Authentication of Key Biological and/or Chemical Resources	
Appendix	
12. Appendix	

Tracking Number: GRANT13000578
Funding Opportunity Number: PAR-17-273. Received Date: 2020-01-24T10:13:23.000-05:00

Introduction and specific aims of the renewal application

The Bioconductor project is dedicated to the analysis and comprehension of high throughput genomic data, including sequencing, microarray, flow cytometetry, proteomics, and imaging data. For two decades, the project has supported collaborative efforts between biologists and data scientists to develop, document, and distribute high-quality scientific software that solves pressing problems in genome-scale biology. Throughout the lifetime of the project, software, technical and pedagogic documentation, and exemplary datasets have been made available following findable, accessible, interoperable, and reusable (FAIR) principles, for cutting-edge assays and analysis methods in genome biology.

Bioconductor is based on the R statistical programming language. It consists of 1,823 software, 953 annotation, and 385 data packages that are portable across Linux, Windows, and macOS systems, supported by a team of core investigators and software engineers who have defined and implemented standards of documentation and code reliability followed by more than 1,200 contributing developers to date. Bioconductor is highly respected and core team members have been invited to collaborate as principal investigators on large genome informatics consortia such as the Human Cell Atlas and the NHGRI Genomic Analysis and Visualization Informatics Lab Space (AnVIL).

The overall goal of this proposal is to maintain and extend the Bioconductor project so that it continues to offer the premier data science platforms for genomics. Specifically, we will achieve this goal with the following aims:

Specific Aim 1) Maintain availability and growth of the premier open source/open development software/data ecosystem for modern genome biology and genomic medicine. Since its inception around 2001, bioconductor.org has been available 24/7 to provide biologists with tools to analyze state-of-the-art genome-scale assays, and to provide statisticians and data scientists with raw and processed genome-scale data to foster the creation of algorithms that identify and reduce systematic bias and unwanted variability, and to perform rigorous statistical inference on compelling biological hypotheses. We will continue to maintain and modernize the highly available resource delivery system that has attracted over 1,000 contributing developers and responds to over 500,000 download requests per year.

Specific Aim 2) Enhance reliability and performance of core genome analysis infrastructure components through improved formal testing disciplines and modernization of continuous integration/continuous delivery methods of the project. Authors of Bioconductor packages may declare dependencies on other Bioconductor packages, and on packages distributed by the Comprehensive R Archive Network (CRAN). Package interdependencies are desirable insofar as they represent efficient component reuse but are undesirable to the extent that the effects of errors in one package can propagate to all packages that depend upon it. We will introduce analysis of package interdependencies in Bioconductor and metrics for coverage of package code by unit tests, define goals for unit testing within packages, and targets for interoperability testing. We will also address improvements of multiplatform continuous integration/continuous delivery methodologies for software ingestion, change management, and testing so that developers can work directly with runtime images of the bioconductor build system.

Specific Aim 3) Advance development and deployment of data science genomic methods that leverage emerging cloud-scale computing and data service paradigms. New methodologies of software containerization, cluster orchestration, API integration, and workflow specification and execution are having strong and favorable impacts on large scale projects in population genetics and genome biology. We will build bridges between familiar Bioconductor programming paradigms/data structures and cloud-oriented methods for scalable batch and interactive analyses with distributed cloud-scale data. We will also continue to enhance our highly successful methods of data quality assessment, data element harmonization, curation and distribution for experiments in genome biology by supporting contributing developers in adoption of best practices for design and deployment of performant, reliable, and reproducible analytical methods.

Specific Aim 4) Enhance the education and community engagement processes that have been intrinsic to the project since its inception. We will continue our commitment to managing regular scientific conferences and meetups for the Bioconductor developer and user communities. We will maintain our engagement with our Scientific Advisory Board and Technical Advisory Board to receive input on emerging requirements, opportunities, and obstacles, and will introduce a Community Advisory Board to widen avenues of input to the project. We will continue the enhancement of documentation processes for training at novice and expert levels, taking advantage of cloud-based interactive notebook environments, automated code walkthroughs, contributions to MOOCs and other video-based methods of instruction.

Research Strategy

Justification

Overview. Rapid advances in genome biology and biotechnology have sparked enormous ambitions in the directions of comprehensive mechanistic theories of biological processes, with personalized genomic medicine as a primary target. Readouts of genomic assays and the sample sizes of genome-scale experiments have grown to the scale of billions of features on millions of samples. In their historical review of medical genetics, Claussnitzer and colleagues cite massively parallel reporter assays, CRISPR editing, knockout screens, regulatory element detection, high-content imaging, and single-cell transcriptomics and epigenomics as developments that can help identify specific disease-causing variants [1]. These authors conjecture that *in silico* models integrating over the outputs of these assay modalities can help ”predict causal variants, effector transcripts and cellular effects” and argue that ”[i]n due course, such models should reduce the need for exhaustive experimental characterization of function for all variants across all cell types.”

Method developers have worked hard to keep pace with the variety and volume of genome-scale data resources and data analysis problems emerging through biotechnological innovation. We expect this pace to continue or increase during the next several years. For example, in the domain of immuno-oncology alone, Finotello and colleagues describe in a review how new sequencing technologies and applications, such as HLA typing, neoantigen prediction, cell-type deconvolution, sc-RNA-seq analysis, TCR/BCR repertory analysis, multiplex image analysis, single-cell visualization, and general data repository management, require over 100 computational tools to manage and analyze [2]. Navarro and colleagues note that accelerating research into *relationships between* new data resources based on dynamic imaging, health records analysis, and proteomics will lead to data management and analysis requirements of staggering scope [3].

The Bioconductor project (https://bioconductor.org, main funding to date through NHGRI U41 HG 004059) has a two-decade history of coordinating collaborative effort between biologists and data scientists to develop, document, and distribute high-quality scientific software that solves pressing problems in genome-scale biology. Throughout the lifetime of the project, software and exemplary datasets have been made available following FAIR (findable, accessible, interoperable, and reusable) principles, for cutting-edge assays and analysis methods in genome biology, including the majority of assay modalities noted in the reviews cited above. Thus Bioconductor has served the scientific community as a go-to resource for researchers and trainees who want to get acquainted with new data types and analysis methods. This results from Bioconductor’s commitment to creating and managing an ecosystem of **easily acquired, formally tested, and richly documented software and data packages** in the R framework for statistical computing. Figure 1 is a concise view of key stakeholder activities and aims of the project.

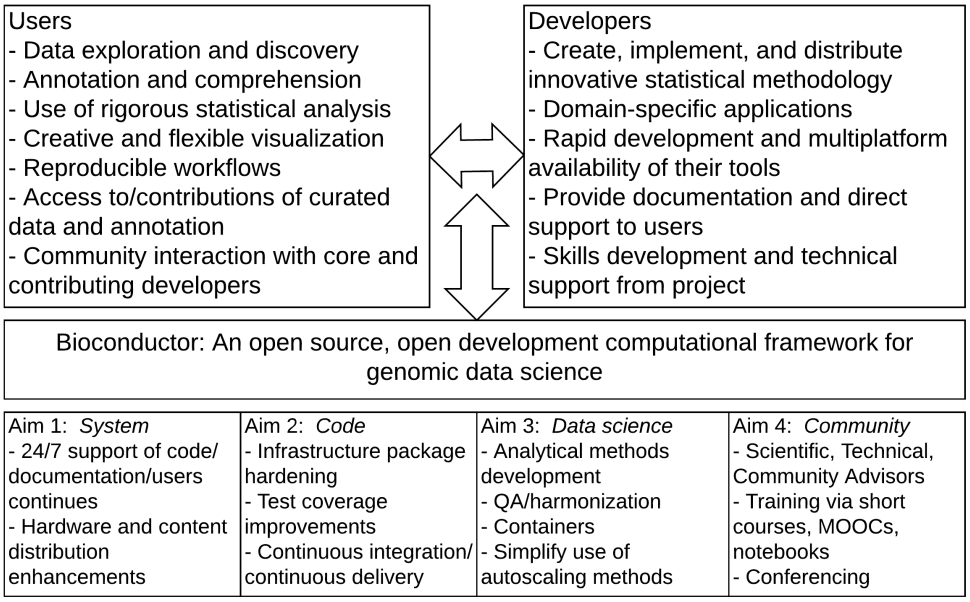


Figure 1: Project overview.

To place project aims in the context of justification of this renewal proposal, we provide a brief historical sketch.

- 2000-2010: Bioconductor provides state-of-the-art methods for preprocessing, normalization, and analysis of DNA microarray data addressing gene expression, genetic variation, and epigenetic modifications. Biologists and bioinformaticians are aided by a rich collection of array and functional genome annotations that are integrated seamlessly with the R programming language. In this phase, project members contribute key concepts of R data structure, package design, Web API strategies, and repository architecture. Below are some examples:
  - We introduced *S4 class systems* to formally specify collections of data types with built-in validity checks. This system greatly improved reproducibility as it automated consistency between assay data, sample information, and feature annotation.
  - To solve concrete problems with instructive narrative we introduced the concepts of *vignettes*, computable documents that demonstrate integrated use of a family of functions, possibly drawn from multiple packages. We required packages to include at least one vignette.
  - The *GEOquery* and *ArrayExpress* packages were developed to help users directly import archived experimental data from NCBI GEO and EBI ArrayExpress as S4 instances representing self-documenting objects compatible with Bioconductor packages. By streamlining the process and removing data formatting steps, this greatly facilitated the analysis of publicly available data.
  - The `biocLite()` user-level function was introduced to resolve package and language-version interdependencies and provide consistent user installations.
  - A robust build system was developed. This system was an early version of multiplatform continuous integration that tested and built all packages in two streams: *release*, where APIs are locked except to fix bugs, and *devel*, in which new features are introduced until frozen in the next release.
  - First Bioconductor paper published in 2004 in Genome Biology. According to Google Scholar this paper has been cited 11,394 times, as of January 20, 2020.
- 2010-2015: The focus changes to sequence-based technologies, necessitating new approaches to preprocessing, normalization, annotation, and organization of readouts from multiple assays. Highlights during this period included the following:
  - The *GenomicRanges* package and extensive supporting infrastructure is introduced to provide a way to efficiently represent and manipulate genomic annotations and alignments and becomes one of the most downloaded packages (currently 10th with over 150,000 downloads per year).
  - The *DESeq2* package for differential expression analysis of RNA-Seq data in contributed to Bioconductor and quickly enters into the top five packages (by citation count) with over 1,000 citations by 2015.
  - Batch effects are identified as one of the major challenges to genomics studies [4]. Two solutions, surrogate variable analysis and ComBat [5] are highlighted in this paper, and are made available through the contributed Bioconductor package *sva*.
  - Packages developed from 2000 - 2010 continue to be widely used, with citation records identifying *limma*, *vsn*, and *affy* as garnering, cumulatively, > 700 citations each during this period.
  - Annual reports describe 6-10 short courses delivered internationally each year.
  - Nature Methods overview of project is published in 2015 [6]. According to Google Scholar this paper has been cited 1,449 times, as of January 20, 2020.
- 2016-present: Advent of widespread use of single-cell assays introduces concerns with much larger data volumes, increased requirements for high-resolution annotation, and significant methodological attention to bias modeling. Highlights and important accomplishments during this period included the following:

  - The *GenomicDataCommons* package, providing R functions and classes to use RESTful services with token-based authentication, is developed to handle Cloud-native genomic data methods handled for NCI's data commons. Dr. Sean Davis of NCI, a member of Bioconductor's Technical Advisory Board is the contributor.
  - The support site support.bioconductor.org receives 297,467 visits in 2016, growing to 492,422 in 2019. Annual posts to the site are approximately 3,000, annual reply counts are around 11,000 per year. The developer mailing list has 140 posts/month, from about 40 authors/month.
  - Collaborations with Human Cell Atlas and NHGRI AnVIL begin in 2018, leading to extensive development of new infrastructure for high-performance numerical formats and novel approaches to API integration.
  - Current PI Dr. Morgan is recruited to Roswell Park Cancer Institute (RPCI) and main project assets are successfully relocated from Fred Hutchinson (Seattle WA) to Buffalo, NY in 2016.

This brief summary of events of 20 years of the project demonstrates that:

- The project has a large and devoted *user community*. Given the statistics noted above on support site visits and software downloads, at present, it seems likely that every working bioinformatician in the world has *used* components of this system, explicitly or implicitly, as packages and functions are deployed in comprehensive analytic frameworks in many different domains. Beyond this, we acknowledge thousands of contributing developers, authors, and support site contributions, whose efforts are directed at the newest challenges arising in genomic data science, and who view the Bioconductor project as the best vehicle for transparent distribution of their work and ideas.

- *Uptake and support* of the project are also well-documented through the events noted in the historical summary above, which show that the tools are described and used in highly cited publications, and that the general methodology is sufficient to warrant invitation of core project members to collaborate on cutting edge solutions in genomic data science.

- *Anticipated impact* of the renewal of the project is predicted to be extremely favorable, given continually increasing need for statistically rigorous data analysis methodology and innovative software design to achieve performant and reproducible analyses in modern genomic data science. The proposed development work on improving reliability of the software ecosystem, the research aims connected with bias reduction, data harmonization, and integrative analysis are expected to have strong positive impacts as single-cell and multiomic experiments proliferate. The extensive new work on fostering interactive cloud-resident analysis, and developing new materials and methods for training the workforce in genomic data science are also expected to be highly impactful.

To conclude this discussion of justification for the renewal proposal, we describe Bioconductor's *complementarity to other resources*. Briefly, we consider the NHGRI Genomic Data Science Analysis and Visualization Informatics Laboratory (AnVIL), the Chan-Zuckerberg Initiative (CZI) Human Cell Atlas (HCA), the NCI Information Technology for Cancer Research program, and the NCI TCGA (The Cancer Genome Atlas) and the NCBI resources Gene Expression Omnibus and Sequence Read Archive.

- Along with Dr Levi Waldron of City University of New York, Drs Carey and Morgan are PIs of a component of the NHGRI AnVIL and have contributed infrastructure software and pedagogical materials to help bring the visions of federated genomic data and analysis in the cloud [7] to reality. The value of a smooth entrance to this resource, for thousands of Bioconductor users, cannot be overestimated, both for the return on investments into the AnVIL platform, and for genomic data science in general.

- Dr Morgan leads an 8-member group of Bioconductor contributors who are funded by the Chan Zuckerberg Initiative (CZI) to create interfaces to tools and data emerging in the Human Cell Atlas (HCA). *HCAExplorer* is part of Bioconductor 3.10 and currently provides interactive access to over 30Tb of data obtained on over 4.5 million cells. Other members contribute to HCA developments in multiomics and single cell analysis methodology.

- Along with Dr Levi Waldron, Drs Morgan and Carey are funded in the NCI Information Technology for Cancer Research program through a U24. Aims of their project involve methods for analysis of multiomic assays with single cells, comprehensive functional annotation of cancer-associated noncoding variants, development of interfaces to other ITCR projects like UCSC Xena and CiVIC, and outreach and training.

- Drs Jack Zhu and Sean Davis of NCI contributed packages *GEOquery*, *GEOmetadb*, and *SRAdb* to address discovery and retrieval of assay data from the NCBI genomic data archives. *GEOquery* captures assay quantifications from a GEO series, along with metadata about assay features and assayed samples, and assembles an `ExpressionSet` instance to coordinate the molecular, annotation, and phenotype information for experimental samples. The 2007 paper [8] has over 600 Google scholar citations.

In summary, given the relevance of project resources to a large and growing user community, the evidence of scientific utility through the record of direct use and citation in scientific literature, and direct collaborative interaction with significant independent genomic research projects, we find strong justification for renewal of project funding.

Production of the resource

In our description of production activities planned in this proposal, we enumerate continuing features to be maintained, and new features to be introduced, in the project plan. These features are enumerated linearly to simplify cross-referencing, but those that are new in this renewal are specifically noted.

Aim 1) Maintain availability and growth of bioconductor.org.

Feature 1 *Maintain the CRAN package BiocManager, the bioconductor.org web site, and git and GitHub repositories to provide immediate access to current and legacy Bioconductor resources within interactive R sessions.*

- **BiocManager:** Wide usage of R in science and industry entails that, among scientific users worldwide, there will be a mix of versions of R on various platforms that will be used to request installation of Bioconductor packages. *The* "BiocManager" *package* includes infrastructure that interrogates the version of R in use to determine the version of Bioconductor from which requests for software should be satisfied. Function `install` arranges transfer and installation of the requested package, and all necessary dependencies. Function `valid` assesses the versions of all installed packages in a given session and reports on whether upgrades or downgrades are necessary to achieve a consistent collection. Function `repositories` defines the web locations of repositories that should be interrogated to resolve requests for software packages. The `cranlogs` service reports that this package is downloaded 71,000 times per month. Maintenance of this package to ensure continued availability at CRAN is necessary to ensure that changes to R and to distributed repository infrastructure do not introduce warnings and errors.

- **The website bioconductor.org** is the main public-facing asset of the project. Visitors have convenient access to documentation on project motivations and activities, sections on installation, learning, usage, and development and contribution, a twitter feed, an events calendar, and excerpts from recent activity on the support site. Package landing pages (see Figure 2) are available for each package (including legacy versions) and provide rich information on availability, popularity, frequency of mentions in the support site, duration of presence in the project, build and test status. Links to package vignettes and other web-accessible resources are also available on landing pages.

- **Git and GitHub source code management.** Many current contributors use GitHub to publicize and manage their software. Bioconductor uses pure **git** to manage definitive package images, but mirrors sources to GitHub for many key packages, to facilitate general use of the web-based user interface. Bioconductor technical staff have write privileges over all git repositories of contributed and project-based packages, so that critical inconsistencies can be resolved and users can be assisted in the event of infrastructure changes that have systemic adverse effects.

▷ **Tasks of feature 1** are ongoing and will be continued throughout the renewal period.

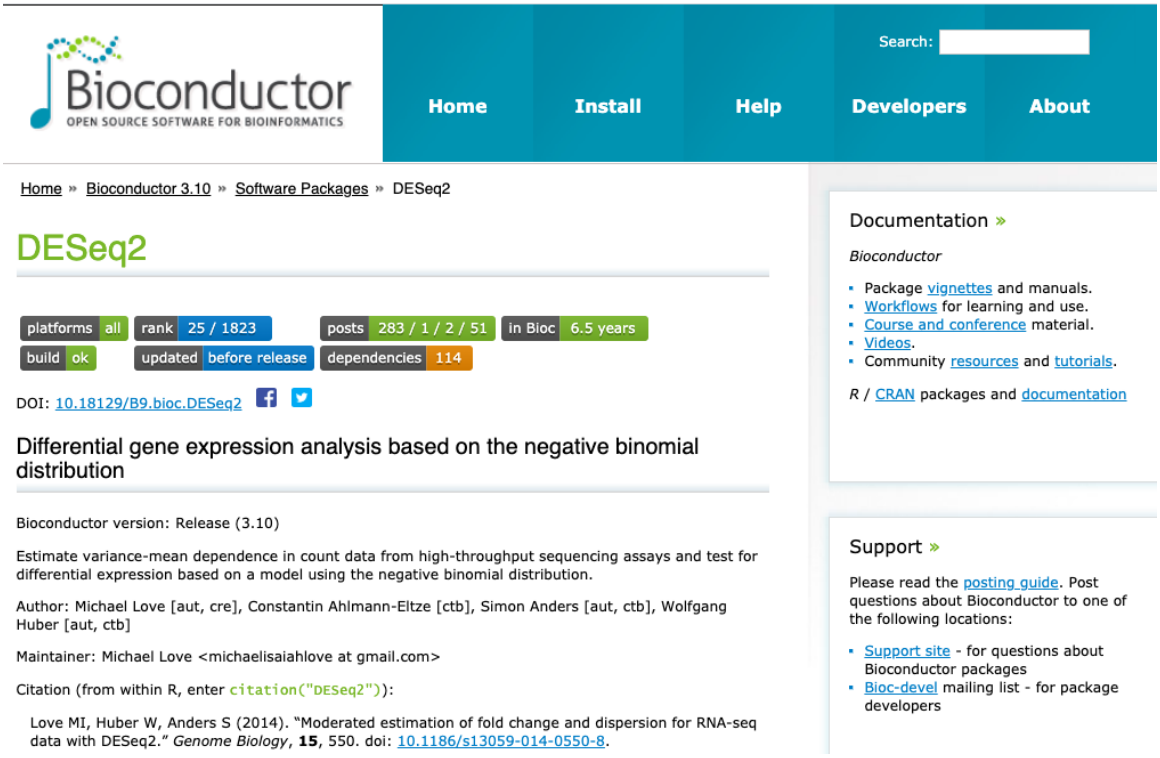


Figure 2: An example of a bioconductor.org software landing page.

Feature 2 *Maintain semiannual release cycle, synchronized with updates to the R language as necessary.* Exclusive of the incorporation of brand new contributions (see Feature 4 below), the release cycle discipline includes a) deadline definitions, coordinated with R core, and communications to staff and contributors, b) package updating (conducted by staff and all contributors) to address changes to R and to upstream dependencies, with conflict resolution as necessary, c) verification that changes to package functionality do not have adverse effects on reverse dependencies, d) general API testing and improvement of documentation.

▷ **Tasks of feature 2** are ongoing and will be continued throughout the renewal period.

Feature 3 *Maintain legacy resources (software, data and annotation packages tagged* `RELEASE_X_Y`*) through git and Docker container repositories.* When a new release is established, the associated sources are immediately copied into the next *devel* version for all packages, and the prior release is frozen. No changes whatsoever are permitted within the project for packages in past releases. `git fetch` and `checkout` commands for any Bioconductor package repository provide access to the full history of code versions. This process is well-established over the history of the project.

▷ **Tasks of feature 3** are ongoing and will be continued throughout the renewal period.

Feature 4 *Continue systematic acquisition and review of newly contributed packages through a github-based protocol.* Guidance for prospective contributors is clearly identified in the "Develop" pane of bioconductor.org. The issues tracker at `https://github.com/Bioconductor/Contributions/issues` includes one issue per active contribution that is under review. Developers agree to a number of conditions concerning maintenance commitments and the review dialogue takes place in the open in the issue tracker. The dedicated single-package builder infrastructure builds and checks the package on every commit with version bump.

▷ **Tasks of feature 4** are ongoing and will be continued throughout the renewal period.

Feature 5 *Maintain and improve general Bioconductor Build System accessibility and reporting.* The Bioconductor Build System generates reports on a daily basis that are available at the "Develop" pane at `bioconductor.org`. Separate build processes are conducted for Linux, macOS, and Windows, and results of builds are compiled for review by the community. See Figure 3 for a small excerpt. The report indicates a) the general scope of building, with 1,847 package install attempts on Linux (with 11 failures), b) the diversity of results for different platforms, c) platform-specific outcomes for each package. The "CHECK" column shows warnings or error on all platforms for two packages, and the tags (OK, WARNINGS, ERROR) are all hyperlinked. The developer can get access to the logs recording the events leading to warning or error by clicking on the tags. This build reporting process will be continued.

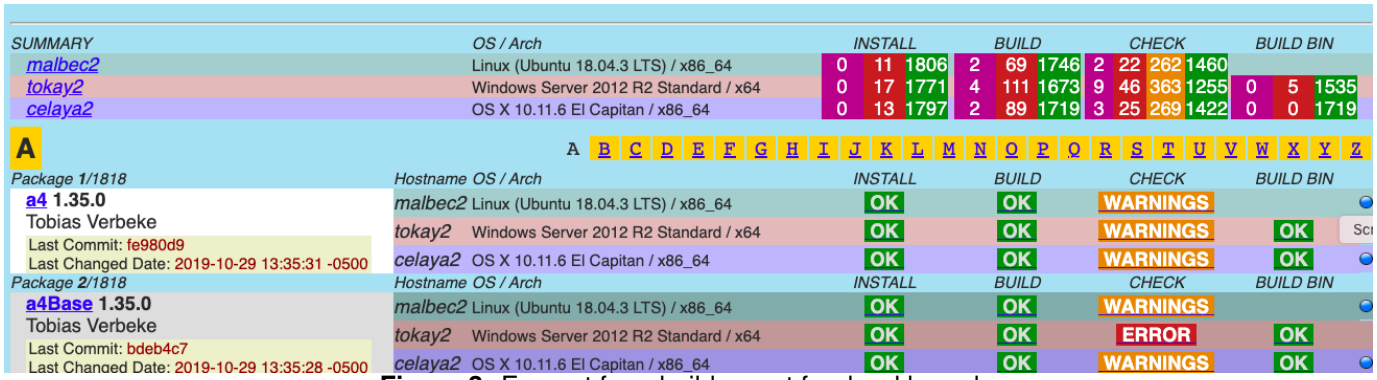


Figure 3: Excerpt from build report for devel branch.

▷ **Tasks of feature 5** are ongoing and will be continued throughout the renewal period.

Feature 6 *Maintain package landing pages for documentation and code access.* Package landing pages are extremely convenient summaries of package functionality and include links to vignettes and developer-provided external resources when available. See Figure 2 for a limited excerpt. We will maintain the autogeneration process for generating and publishing package landing pages throughout the project.

▷ **Tasks of feature 6** are ongoing and will be continued throughout the renewal period.

Feature 7 *Continue support of innovative approaches to documentation.* The *BiocStyle* package was introduced in 2013 to facilitate markdown-based multiform vignette, manual page, and workflow production. It is easy for package maintainers to achieve uniform high-quality formatting of their vignette text by including references to styling resources that are installed with *BiocStyle*. R's infrastructure for vignette building originates with Bioconductor, and now includes specification of vignette building processes in package DESCRIPTION and vignette metadata elements. These practices are now used in CRAN packages and constitute a major and enduring contribution from the Bioconductor project back to the R community.

Maintenance of the *BiocStyle* package is significant and involves careful tracking of changes to *knitr*, LaTeX and pandoc projects, unit testing of styling support components, and responding to user feedback. In addition to vignette-oriented support in this task, the growth of interest in composition of interactive notebooks for demonstrative training in a read/write context has led to progress in technology for linking software repositories to notebook environments. Translation from R markdown to jupyter notebook syntax is possible, and Rstudio has its own notebook concept. Feature 19 below addresses new initiatives in project activities related to notebook creation and maintenance.

▷ **Tasks of feature 7** are ongoing and will be continued throughout the renewal period. Research activities related to notebook composition and maintenance are a part of Aim 3 and are slated to begin in quarter 6 (Q6) of the project.

Feature 8 *Continue support of the BioStars-derived user/developer forum* `support.bioconductor.org`, *and the developer-focused mailing list bioc-devel at bioconductor.org.*

We have included a letter of support from Dr Istvan Albert, the leader of the Biostars.org project [9], who generously supported the team in adapting the Biostars source to create support.bioconductor.org. We are using the Amazon Web Services Simple Email Service to manage large-scale mail operations for the developer community. As community engagement and use grows, the server load from support.bioconductor.org has increased and additional hardware will be devoted to this asset.

▷ **Tasks of feature 8** are ongoing and will be continued throughout the renewal period.

Feature 9 (new) *Transition main production assets to Dana-Farber Cancer Institute (DFCI).* The purchase of hardware, transfer of cloud service configuration and accounting items, and transfer of git/github management processes to DFCI will occur during the first two quarters of the project. Formal conditions of non-regression of the relocated system will be agreed upon by Roswell Park Cancer Institute (RPCI) co-investigator Dr Martin Morgan and Drs Irizarry and Carey, with consultation of the Technical Advisory Board. After verification of successful transition, services at RPCI will be shut down.

▷ **Tasks of feature 9** will commence in the first quarter of the project (Q1) and are planned to conclude at the end of Q2.

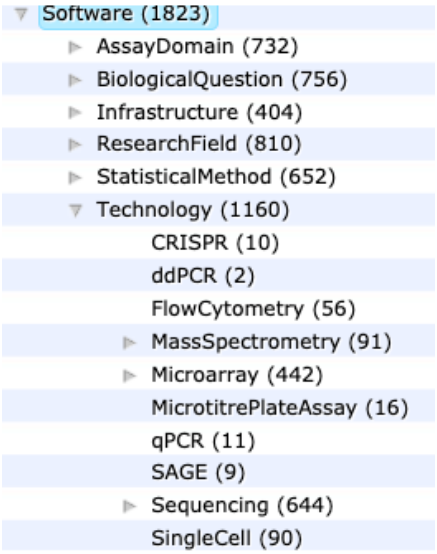
Feature 10 (new) *Improve resource discovery support at bioconductor.org.* The main components for resource discovery in the project are a) the top level search facility at the web site `bioconductor.org`, for very general inquiries, b) the package annotation system governed by the *biocViews* package, for package-oriented

inquiries, and c) the *AnnotationHub* and *ExperimentHub* metadata elements, for inquiries about genomic reference data and exemplary experiments.

The search facility is currently fairly primitive. A search for ”spatial”, for example, returns 4573 results. Though many of the ”top hits” are of evident interest, the reported count is exaggerated. The list of hits has a repetitious appearance, and does not distinguish current from legacy resources. We will introduce improved indexing and filtering of search results using standard methodology of search engine optimization.

Figure 4 is an excerpt of the biocViews hierarchical vocabulary showing main subtopics of ”Software”, and the expansion of the term ”Technology”. The topic set will be revised and publicized through the GitHub repository for the biocViews package. An ongoing effort will be made to obtain community input on adequacy of the vocabulary prior to each release Package reviews will include assessment of accuracy of package tagging in DESCRIPTION. Reports on package downloading and citation will include sections organizing according to biocViews terms, to aid in substantive evaluation of community interest and usage trends.

Figure 4: Expansion of ”Technology” biocViews topic. Numbers in parentheses are counts of packages using the associated tag in DESCRIPTION.



▷ **Tasks of feature 10** will commence in Q3 of the project and will be a component of each subsequent release through the processes of biocViews vocabulary assessment and revision, evaluation of package tagging, and reporting on resource usage organized topically using view terms.

Feature 11 (new) *Enhance developer support resources at bioconductor.org.* The current ”Developer resources” section of the main site has excellent material on package initiation, submission, and maintenance processes in Bioconductor. The document ”Troubleshooting Build Reports” specifically addresses changes to the R language and package validity criteria that may necessitate changes to contributed package code. These resources will be kept current throughout the project period.

Areas of software development that will be carefully reviewed in support material for developers are

- strategies for parallelization, typically to be mediated through *BiocParallel*, which may involve multicore or cluster orchestration;
- strategies for analysis of out-of-memory data resources, for which the most familiar case at present is local HDF5;
- reliability enhancement through effective testing and instrumentation; static analysis with *BiocCheck* identifies opportunities to harden code and documentation, and this procedure will be maintained.

We include a letter of support from Dr Mike Smith of EMBL who has initiated a monthly live Bioconductor Developer Forum conducted via BlueJeans. The November 2019 forum is available on YouTube at `https://youtu.be/NXztWuJSItk`. We will digest topics covered in the Forum series and publicize these through the support site and the main page.

▷ **New Tasks of feature 11** including renovation of developer resource content will commence in Q3. The developer forum meetings will be continued monthly throughout the renewal period.

Aim 2) Enhance reliability and performance of core infrastructure packages; modernize continuous integration and continuous delivery strategies.

Feature 12 *Maintain the collection of infrastructure packages for data structures, annotation resources, and interoperation*. Key user-facing components such as `SummarizedExperiment`, `GenomicRanges`, `GenomicFeatures` (for, e.g., gene and transcript catalogs) are built upon a system of (sometimes virtual) classes managed in packages that users seldom have direct involvement with (e.g, *S4Vectors*, *XVector*). These infrastructure packages are sometimes dependent on software defined in CRAN packages. Continuous testing and maintenance of these packages is essential for smooth functioning of user-facing functions. Package lifetime and change plans will be transparently publicized using GitHub site components.

▷ **Tasks of feature 12** are ongoing and will be continued throughout the renewal period.

Feature 13 (new) *Formalize the governance of key infrastructure packages.* In consultation with core staff developers and the Technical Advisory Board, we will enumerate the key infrastructure packages. Proposals to modify these packages must take the form of a GitHub ”pull request” defined and tested on a new branch of a fork of the package repository. Governance materials will specify a procedure for pull request review and a voting discipline to authorize changes to key infrastructure components.

▷ **Tasks of feature 13** will begin in Q3. The governance materials related to pull request evaluation and voting will be ratified by Q6. The procedures adopted by Q6 will be operative for release 3.16 in Q8.

Feature 14 (new) *Implement test coverage measurement and enhancement plans for key infrastructure packages.*

The CRAN/Rstudio *covr* package measures the quantity of package code that is exercised in unit tests, man page examples, or vignettes. Use of unit tests is optional, and measures of test coverage are not collected in the Bioconductor build system. Figure 5 is the report for the devel version of *SummarizedExperiment* as of January 17, 2020. Many functions are exercised in unit tests or documentation examples or vignettes, but we note that the `RangedSummarizedExperiment` code is not fully tested; over 40% of the code is not exercised. Because `RangedSummarizedExperiment` is an important class for unifying ranges of assayed features in genomic coordinates with assay readouts and sample characteristics, it is desirable that unit testing for this class and its methods be as comprehensive as possible.

We will introduce test coverage measurement for the key infrastructure packages identified in Feature 13 and will consult with the Technical Advisory Board on a plan to optimize coverage measurement in the build system. The use of *covr* as part of the build system, producing coverage reports for all packages, introducing elements related to test coverage method to *BiocCheck* as appropriate. We note that test coverage measurement can be time-consuming, and reporting tasks may need to be spread out over the week, testing subgroups of packages at each build event.

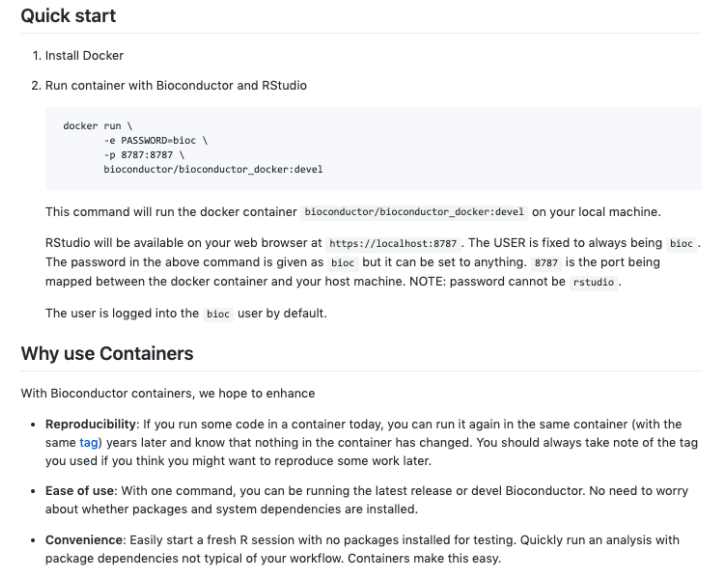
Figure 5: Report from `package_coverage(type="all")` for SummarizedExperiment package.

```
SummarizedExperiment Coverage: 80.54%
R/RangedSummarizedExperiment-class.R: 57.22%
R/Assays-class.R: 67.88%
R/readKallisto.R: 74.63%
R/makeSummarizedExperimentFromLoom.R: 84.62%
R/SummarizedExperiment-class.R: 88.72%
R/makeSummarizedExperimentFromExpressionSet.R: 93.37%
R/coverage-methods.R: 100.00%
R/findOverlaps-methods.R: 100.00%
R/inter-range-methods.R: 100.00%
R/intra-range-methods.R: 100.00%
R/makeSummarizedExperimentFromDataFrame.R: 100.00%
R/nearest-methods.R: 100.00%
R/zzz.R: 100.00%
>
```

▷ **Tasks of feature 14** will begin in Q5. We will aim to have an approach to comprehensive test coverage measurement by release 4.0 in Q10.

Feature 15 (new) *Introduce developer-accessible continuous integration discipline for Bioconductor packages.* A frequent complaint encountered on the `bioc-devel` mailing list concerns events in which the project build system flags warnings or errors for a given package, which cannot be replicated by the developer, even when the developer's machine type and operating system closely match those of the build system. Typically the problem arises from incompatible installed package sets, but more subtle incompatibilities can involve version discrepancies in operating system runtime libraries or hard-to-manage assets like LaTeX. Containerization can be used at both the project build system and the developer's local system to ensure consistency of system operations and reproducibility of adverse events. New developments in the GitHub Actions system provide GitHub users with access to Linux, macOS and Windows builders. We will provide documentation and examples of working with containers and GitHub Actions to achieve consistency of the build system and user development environments.

Figure 6: Excerpt from README.md at `github.com/Bioconductor/bioconductor_docker`.



Quick start

1. Install Docker
2. Run container with Bioconductor and RStudio

```
docker run \
       -e PASSWORD=bioc \
       -p 8787:8787 \
       bioconductor/bioconductor_docker:devel
```

This command will run the docker container `bioconductor/bioconductor_docker:devel` on your local machine.

RStudio will be available on your web browser at `https://localhost:8787`. The USER is fixed to always being `bioc`. The password in the above command is given as `bioc` but it can be set to anything. `8787` is the port being mapped between the docker container and your host machine. NOTE: password cannot be `rstudio`.

The user is logged into the `bioc` user by default.

Why use Containers

With Bioconductor containers, we hope to enhance

- **Reproducibility**: If you run some code in a container today, you can run it again in the same container (with the same tag) years later and know that nothing in the container has changed. You should always take note of the tag you used if you think you might want to reproduce some work later.
- **Ease of use**: With one command, you can be running the latest release or devel Bioconductor. No need to worry about whether packages and system dependencies are installed.
- **Convenience**: Easily start a fresh R session with no packages installed for testing. Quickly run an analysis with package dependencies not typical of your workflow. Containers make this easy.

▷ **Tasks of feature 15** will begin in Q6. A report on methods of continuous integration in the project should be submitted Q12.

Aim 3) Advance development and deployment of methods of genomic data science that take advantage of emerging cloud-scale computing and data service paradigms. Support contributing developers in adoption of best practices for design and deployment of performant, reliable, and reproducible analytical methods.

Feature 16 *Continue the development of containers encapsulating infrastructure required to use all aspects of R/Bioconductor.*

Figure 6 provides the exact one-line command sufficient for a Docker user to obtain a container image that includes R, RStudio (Community Edition), *BiocManager*, and sufficient runtime library and compilation infrastructure to allow installation of 99.9% of all Bioconductor packages on demand. (The exceptional packages have special prerequisites that expert users can address with additional configuration.)

Container images will be available through Dockerhub, Google Cloud Registry, and other relevant distribution mechanisms. Dockerfiles for major container types developed in the project will be managed in public GitHub repositories.

▷ **Tasks of feature 16** are ongoing and will be continued throughout the renewal period.

Feature 17 *Continue enhancing Bioconductor methods for scalable data analysis.* Main directions of work on enhancing scalability of genomic data analysis processes include the following.

- Memory-sparing data representation and access: The *DelayedArray* package defines methods for lazy programming with local or remote data stores whose contents cannot be ingested in RAM. Interfaces have been defined to allow use of R programming idioms with Bioconductor data classes with local HDF5 (*HDF5Array*), remote HDF5 in the HDF Scalable Data Service (*rhdf5client*), Google BigQuery (*restfulSE*), and the zarr format (under development in `github.com/Bioconductor/ZarrExperiment`.)
- Configurable programming for parallelization. Package *BiocParallel* streamlines the selection and use of diverse methods of parallel computation, and is used by over 100 other Bioconductor packages.
- Characterization and instrumentation. As defined by Weinstock and Goodenough [10], a process is scalable when throughput can be increased in a cost-effective way by adding available resources. Tradeoffs exist when considering how to produce scalable workflows in genomics. When comparing solutions that use shared memory multicore computing against solutions that use clusters of possibly heterogeneous processors, costs of interprocess communication, vulnerability to faults arising owing to contention for resources, and increases to software complexity, must be weighed together. Methods for measuring system throughput and assessing costs and risks of specific strategies are in short supply, and we will introduce documentation, executable examples, and research publications illustrating profiling for measurement of task timing, memory consumption management, and fault-tolerant programming methods.

▷ **Tasks of feature 17** are ongoing, and extension of the family of back ends for DelayedArray processing will proceed as new back ends are identified as useful for genomic computing. A research publication on strategies for achieving scalable analysis workflows will be produced by Q12.

Feature 18 (new) *Introduce new methods for managing and analyzing data that take advantage of the ALTREP developments by R Core*. ALTREP allows ALTernative REPresentations of base *R* data structures such as numeric vectors. Instead of *R*'s fully realized in-memory representation of data, ALTREP implementations might use memory-mapped, compressed, or remote data represenations. ALTREP is completely transparent to end-users, so that no new classes or programming paradigms are required. The approach places considerable burden on the developer to implement sufficiently rich alternative representations for useful gains in memory management and performance, and ALTREP support in base *R* is still being developed. Preliminary *Bioconductor* examples include packages that enable read-only disk-based object sharing across parallel processes (avoiding memory and CPU use associated with copying data) and support for standard statistical calculations on ALTREP objects; the core team has also explored use of ALTREP objects in *Bioconductor* data representations such as `GenomicRanges`. Lessons learned from these explorations emphasize the need for carefully developed programming guidelines for effective use of ALTREP.

▷ **Tasks of feature 18** will commence in Q6 of the project.

Feature 19 (new) *Introduce support for interactive notebook-based analytic programming in dedicated cloud environments.*

Dr Carey has contributed Jupyter notebooks that demonstrate advanced methods in Bioconductor to NHGRI AnVIL, HDF Kita Lab, and Google Colaboratory. Two examples for the Colaboratory are at `https://tinyurl.`

com/rwlr9dd (a simple genetics application using 1000 genomes VCF in Amazon Web Services (AWS) S3) and https://tinyurl.com/tzm8dkw (demonstrates access to 181,000 human RNA-seq studies in HDF Scalable Data Service). Both make use of an archive of precompiled binary packages, matched to the notebook runtime environment, so that long package retrieval and compilation efforts can be avoided.

Best practices for provisioning cloud-based analysis environments with resources from a dynamic software and data ecosystem like Bioconductor are not known at present. Tradeoffs exist between user convenience, reproducibility, and managerial simplicity. A new concept of ”repository” will be needed, that extends the concepts of ”platform” and ”package” now central to distribution and installation concepts for R and Bioconductor. An endorsed container image will be treated as a ”platform” is now – it will be self-identifying and identifiable to the Bioconductor package distribution system. When installation requests are made, precompiled binaries in a secure, hosted repository are checked for compatibility with the requesting container image and shipped and installed when appropriate. Further discussion of these concepts is provided below under ”Contributions to the theory and practice of design and operation of data/analysis commons” in the ”Research to improve the resource” section.

▷ **Tasks of feature 19** will commence in Q6 of the project. Work on the extended repository concept will include consultations with colleagues at Rstudio and with the Bioconductor Technical Advisory Board.

Feature 20 (new) *Provide methods for natural programming of genomic workflows with Kubernetes cluster orchestration.*

Kubernetes is an environment-agnostic (cloud-agnostic) framework for automating use of containers across clusters of hosts. Dr Morgan has begun design of Kubernetes orchestration interfaces at https://github.com/Bioconductor/k8sredis. These will ultimately allow programs using *BiocParallel* to execute on Kubernetes clusters.

▷ **Tasks of feature 20** are ongoing. A research report on the relationships between dynamic software ecosystems, data commons, and federated analysis environments will be begun in Q6. Additional material on this topic is provided in the Research component 3 below.

Aim 4) Enhance the community engagement and education processes that have been intrinsic to the project since its inception.

Note that conference “developer days” have been eliminated, in order to broaden inclusiveness of all conference events. More fine-grained activities such as package writing, data resource contributions, documentation authoring will have allocated periods in future conferences.

Feature 21 *Maintain pedagogic documentation resources at the project, package, and workflow levels.* See Dissemination section below for more details.

▷ **Tasks of feature 21** are ongoing and will be continued throughout the renewal period.

Feature 22 *Continue the conference and meetup series in North America, Europe, and Asia, uniting developers and users on a regular basis.* Conference organizing committees communicate regularly using Slack channels managed by project staff.

▷ **Tasks of feature 22** are ongoing and will be continued throughout the renewal period.

Feature 23 *Refine and publicize the governance documents and code of conduct recently developed for the project as a whole.* See the Technical Advisory Board governance document [11].

▷ **Tasks of feature 23** are ongoing and will be continued throughout the renewal period.

Feature 24 *Continue to convene the Scientific Advisory Board, Technical Advisory Board, and Community Advisory Board on specified schedules.* The Scientific Advisory Board convenes on an annual basis in conjunction with the North American conference. The Technical Advisory Board meets monthly, for discussion of new development techniques and priorities. The newly proposed Community Advisory Board will be chaired by Dr Matt Richie of Walter and Eliza Hall Institute of Australia; membership nomination process will begin in 2020.

▷ **Tasks of feature 24** are ongoing and will be continued throughout the renewal period.

Feature 25 (new) *Introduce support for videos and podcasts addressing elementary and advanced aspects of project resources.* We include a letter of support from Dr Stephanie Hicks, who is a creator of the ”Corresponding Author” podcast series, and who has proposed creation of podcasts on modern genomic analysis concepts confronted by Bioconductor. Dr Irizarry is an architect of the edX https://www.edx.org/xseries/data-analysis-life-sciences courses on Data Analysis for the Life Sciences Series and the Genomics Data Analysis Series. We will, in conjunction with the Technical and Community Advisory Boards, define a series of topics to be ad-

dressed in video and podcast presentations, to increase accessibility of project developments to the scientific community.

▷ **Tasks of feature 25** will begin in Q1 and will be continued throughout the renewal period.

Feature 26 (new) *Define communication and action channels connecting the Technical and Community Advisory boards along with governance and regularly updated agenda documents to clarify plans and expectations of the two bodies.* Dr Matt Ritchie is a member of the Technical Advisory Board and will be the inaugural chair of the Community Advisory Board.

Board agendas will be structured so that on a quarterly basis, a liaison from the Technical Advisory Board will report to the Community Advisory Board on topics of concern, and vice versa. Dr Aedin Culhane (DFCI co-investigator) will be the primary interface between the PIs and the Community Advisory Board. Project principal investigators will have executive authority to seek support for and to implement processes advocated by the boards.

▷ **Tasks of feature 26** will begin in Q1 and will be continued throughout the renewal period.

Figure 7 provides a schematic year-by-year overview of major undertakings, stratified by specific aim.

Technologies for establishing and maintaining the resource

We consider technological elements of project execution in three main areas: Source code management and package distribution, operations and project management, and methods for dissemination and socialization of project values.

Source code management and package distribution. The computational environment for the proposed work is made up of a mix of dedicated on-premises servers and commercial cloud computing.

Source code control and package distribution (continuous distribution). We will establish a Linux (Ubuntu) server system at Dana-Farber Cancer Institute that will host a secure git repository for all packages and package commit histories. Bioconductor technical staff will have write privileges over all git repositories of contributed and project-based packages, so that critical inconsistencies can be resolved and users can be assisted in the event of infrastructure changes that have systemic adverse effects. As an example, documentation written in LaTeX can become stale as formatting packages and practices evolve, leading to failures of package compilation. Project staff can perform global search and replace over all vignette folders to mitigate such conditions.

Source code ingestion, management and review of newly contributed packages. GitHub.com and a locally installed instance of the single package builder are used to handle the introduction of new packages. Prospective contributors are instructed to create public GitHub repositories for their packages and to add a webhook that triggers an attempt by the single package builder system to compile the package whenever a push is made that includes an increment to the package version number.

Web site management. Standard Apache software and configuration components are used to host the site https://bioconductor.org.

Virtualization/containerization. Docker images for Bioconductor are based on Dockerfiles that are managed at https://github.com/Bioconductor/bioconductor_full. We also maintain easily deployed AWS EC2 Amazon Machine Images for each Bioconductor release, catalogued at https://bioconductor.org/help/bioconductor-cloud-ami/.

Hub packages for reference genomic annotation and exemplary experiments. The *ExperimentHub* and *AnnotationHub* packages implement a novel approach to efficiently conveying current information to users. SQLite databases are maintained that include metadata and are (individually) updated every time a resource is added to the Annotation or Experiment hubs. The Relational Data Service (RDS) of Amazon Web Services is used to manage the Mariadb database of logs of transactions with the hub databases. Hub resources are typically serialized RDA objects representing curated experimental or reference/annotation data, and these are stored in AWS S3 buckets. Users of *AnnotationHub* or *ExperimentHub* receive, upon evaluating the eponymous functions in these packages, a reference to a locally cached SQLite database containing metadata about all annotation or experiment resources currently available. The cache-resident metadata will be refreshed if resources have been added to the hubs; this happens continuously throughout each release. When users request particular resources from these hubs, downloads from AWS S3 are conducted, the downloaded entity is locally cached, and its value is conveyed to the requestor's session. Future requests for the resource will be resolved against the local cache, unless this particular resource has been modified in the cloud-resident instance in AWS S3, in which case the user will be informed and queried on whether the cached instance should be updated.

Figure 7: Summary of deliverables by year and aim.

	Year 1	Year 2	Year 3	Year 4	Year 5
Aim 1: System	Q1: hiring, hardware acquisition Q2: test build nodes/services at DFCI Q3: Go live with DFCI; shut down RPCI distribution	Q5-Q8: standard operations; ingestion of new packages with build/test on single-package builder at DFCI	Q10: hardware enhancement	Q13-Q16: standard operations	Q17-Q20: standard operations
Releases/ size	Q2: Release 3.13, ~2000 packages Q4: Release 3.14, ~2100 packages	Q6: Release 3.15, ~2200 packages Q8: Release 3.16, ~2300 packages	Q10: Release 4.0, ~2400 packages Q12: Release 4.1, ~2500 packages	Q14: Release 4.2, ~2600 packages Q16: Release 4.3, ~2700 packages	Q14: Release 4.4, ~2800 packages Q16: Release 4.5, ~2900 packages
Aim 2: Code	Q1: core infrastructure package priorities set: e.g., GenomicRanges, SummarizedExperiment, rhdf5 Q2: begin change management governance policy formulation; measurement of test coverage for core infrastructure; introduce public issue tracking system; formalize container maintenance protocol	Q5: test coverage metrics computed and reported for all packages; guidance on improving coverage Q6: ratify change management procedures and long term plans for core infrastructure packages; begin design of open continuous integration/delivery system for developers; container-linked binary package set distribution	Q9: begin publication on continuous integration strategies for open software ecosystems Q10: deploy open continuous integration/continuous delivery for all developers; collect data on utilization, performance, and impact of this approach for publication Q12: submit publication	Q13-Q16: maintenance and standard operations	Q17-Q20: maintenance and standard operations
Aim 3: Data science	Q1: ongoing work on bias reduction and clustering for single-cell sequencing; large-scale metadata integration based on omicidx; documentation on the role of containerized software and cluster orchestration for genomic workflows; Containers4Bioc package development begins Q2: evaluation of AnnotationHub and ExperimentHub maintenance and contribution patterns	Q5: cross-language interoperability protocols come to maturity; specific targets are python machine learning and semantic analysis tooling Q6: begin report on scalable genomic data science with Bioconductor; begin investigations of ALTREP methods; beginner and intermediate Rstudio notebooks in Rstudio cloud; conversions to jupyter notebooks at HDF KitaLab, Google Colaboratory	Q9: demonstrations of autoscaling cluster orchestration with Kubernetes driven by BiocParallel; complete report on scalable genomic data science	Q13-Q16: continued research on scalable bias reduction and improved inference in single-cell genomics and multiomics	Q17-Q20: continued research on scalable bias reduction and improved inference in single-cell genomics and multiomics
Aim 4: Community	Q1: Execution and/or archiving of European Bioc Conference 2021; plans for Bioc2021 North American Conference begin; video training for highlights of Bioconductor release 3.13 to start a series of release-based videos; manage standard board meeting schedule; Q2: NA conf.	Q5: Manage materials for European Bioconductor conference 2022 Q6: Bioc NA 2022	Q9: Manage materials for European Bioconductor conference 2023 Q10: Bioc NA 2023	Q13: Manage materials for European Bioconductor conference 2024 Q14: Bioc NA 2024	Q17: Manage materials for European Bioconductor conference 2025 Q18: Bioc NA 2025

System for building and testing all packages. The Bioconductor Build System is an open source collection of python and bash scripts that drive the multiplatform package testing and reporting system, managed at https://github.com/Bioconductor/BBS. Builds for Ubuntu and Windows occur on locally managed hardware, while macOS builds occur in hosted Apple Hardware in MacStadium.

Operations and project management.

Monthly meetings of the Technical Advisory Board, and weekly technical lab meetings of the principal investigators define the primary managerial processes for the project.

Project-wide e-mail is managed through AWS SES (Simple Email Services) using gmail as the basic service for handling internal mail.

Project management for the core staff is carried out with trello. We plan to adopt Zendesk or a comparable system to handle issue tracking for the project as a whole.

Measurement of web resource usage is conducted using Google Analytics. See Figure 8 for an illustration of page entry and exit tracking over a recent six-week period. Analyses of visitor traffic and download patterns will be used to help guide improvements to web site structure and to formulate agendas for package reliability and performance assessment and improvement.

Dissemination and socialization of project values. We have accounts at twitter.com, youtube.com, and facebook.com. All are used to communicate basic facts about project release cycles, conference dates and real-time conference activity news, and training and job opportunities.

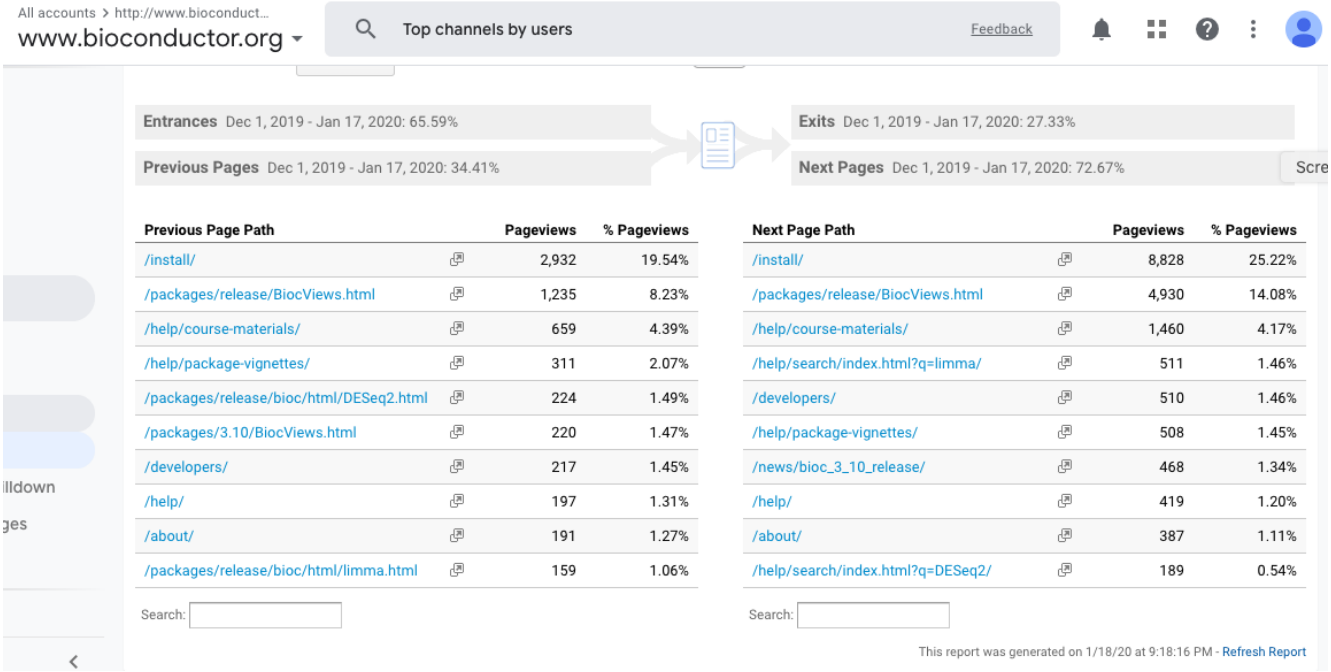
Plans for updating production resource. The updating process for Bioconductor has been integral to the project's function and existence since its inception. To keep pace with evolution in biotechnology, it was agreed upon that the software ecosystem should be updated every six months, with the current release branch frozen (with the exception of bug fixes) to foster stability for users with ongoing analyses. This policy has been followed for the entire history of the project going back to 2002, and appears sound, given anticipated resources, for the duration of the project.

Plans for support and distribution after period of award. Because the main project assets in the software ecosystem are in world-readable git repositories, full development histories can be made available in perpetuity, upon conversion of project-managed git repositories to github.com repositories, should support end. Maintenance of code (e.g., updates to address changes to R) would become a community responsibility. Curated data and annotation would be donated to a suitable scientific open data repository such as Dryad or Harvard Dataverse.

Accomplishments

We review the accomplishments of the Bioconductor project in three subsections. We begin with a quantitative overview, discuss general developments in approaches to administration and community support that enhance

Figure 8: Google Analytics report on user interaction with bioconductor.org.



robustness and impact of the project, and then summarize events in the context of the details of the specific aims of the prior proposal (2015 submission for U41.)

Accomplishments part 1: Quantitative overview Details on developments discussed here can be found in the series of annual progress reports published at https://bioconductor.org/about/annual-reports/.

1. *Growth in user base.* Figure 9 (left panel) depicts numbers of unique IP addresses downloading Bioconductor packages over time. The annual figure now exceeds one-half million unique IP addresses initiating more than 23 million package downloading events per year.

2. *Growth in software availability.* Release 2.7 in 2010 included 419 user-contributed packages. Release 3.2 in 2015 included 1,104. The current 3.10 release has 1,823 packages contributed by more than 1,200 maintainers.

3. *Relevance to and satisfaction of the genome biology and genomic data science communities.*

   (a) *Responsiveness of ecosystem to technological innovations.* Figure 9 (right panel) shows that newly contributed packages are more likely to address sequencing, single cell, or mass spectrometry applications than applications to other technologies.

   (b) *Citations in scientific literature:* We obtain an approximate citation count by searching PubMed title and abstract content, and, for years 2012 and beyond, PubMedCentral full text for "Bioconductor". There were 52 citations per year in 2010, 3,138 in 2015, and 4,610 in 2018; cumulatively there are more than 34,500 total citations.

   (c) In the Justification section above, we noted *formally supported large-scale collaborations* with NHGRI AnVIL, Chan Zuckerberg Initiative Human Cell Atlas, and NCI Information Technology for Cancer Research

   (d) Bioconductor concepts and packages are used directly in large-scale and high-profile training. Examples include the edX Genomics for Data Analysis series (https://www.edx.org/course/introduction-to-bioconductor-annotation-and-analys), the Coursera modules on genomic data science (https://www.coursera.org/learn/bioconductor), and Cold Spring Harbor Laboratory's course (https://meetings.cshl.edu/courses.aspx?course=c-data&year=19).

4. *Growth in annotation and experiment data resources.* Bioconductor has emphasized web-based distribution of annotation and experiment data resources through AnnotationHub (since 2013) and ExperimentHub (since 2016) software, respectively. In 2019, 5,500 AnnotationHub resources were used 124,380 times by 2,088 distinct IP addresses, and 2,680 ExperimentHub resources were used 1.4 million times by 14,160 distinct IP addresses.

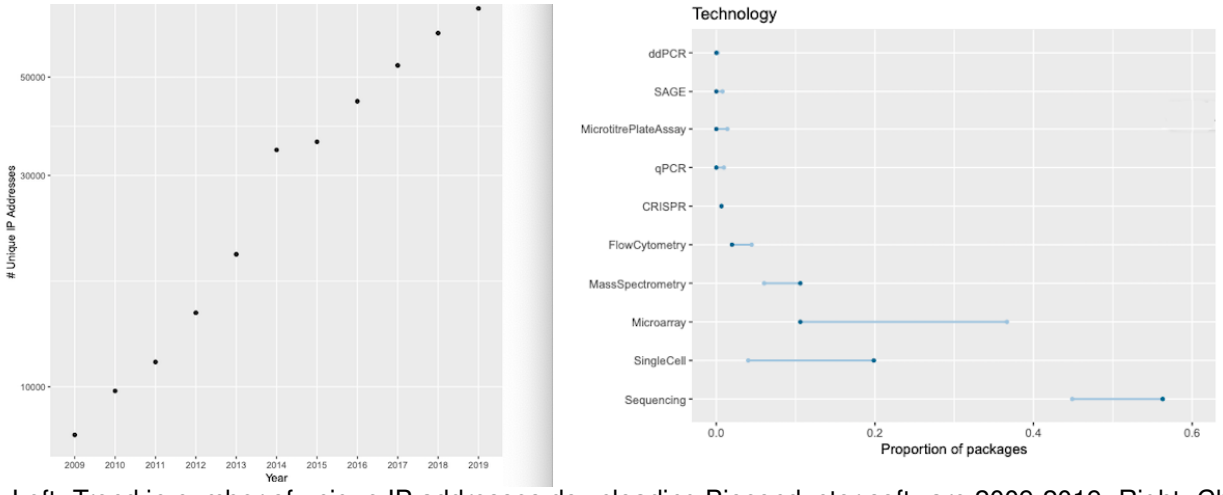


Figure 9: Left: Trend in number of unique IP addresses downloading Bioconductor software 2009-2019. Right: Changes in proportion of contributed packages related to given technologies. Dots are 2019 proportion, opposite end of line segment is proportion of packages related to given technology prior to 2019.

Accomplishments part 2: Enhancements to administration and community support

Governance. The importance of formal governance procedures for the project became apparent in the Scientific Advisory Board meeting of 2018, in which Dr James Taylor of the Galaxy project discussed Galaxy's approaches to management and contributor engagement. Three major steps have been taken in the past year to improve transparency of project decision-making, ensure appropriateness of conduct in collaborations and community events, and to formalize avenues for community involvement.

- A governance document for the Technical Advisory Board (TAB) is available at https://bioconductor.org/about/technical-advisory-board/TAB-Governance.pdf. The charge to the TAB is stated as:

  > Developing strategies to ensure long-term technical suitability of core infrastructure for the Bioconductor mission. Core infrastructure includes: all aspects of package addition, management, and distribution; end-user engagement (e.g., web, support site, and slack); developer support; and development of packages for use by the broader developer community; identifying and pursuing technical and scientific aspects of funding strategies for long-term viability of Bioconductor.

  The governance document prescribes membership election and term limits. Current TAB membership, and minutes of TAB meetings are published under the "About" tab at bioconductor.org.

- A conference Code of Conduct was implemented for Bioc 2019, text at https://bioc2019.bioconductor.org/code_of_conduct. Work proceeds on the creation of a project Code of Conduct, with discussion at the #diversebioc slack channel at community-bioc.slack.com.

- Dr Matt Ritchie of the TAB conducted an initial community survey in Summer 2019. Results were reviewed at the November TABmeeting. There were 107 respondents (as at 25th October 2019), mostly male (78%),post-docs (33%), from Europe (52%), dry lab researchers (80%), based in academia/university (88%), using R for 5-10 years and Bioconductor for 2-5 years (37%). The top 3 reported uses of Bioconductor reported were for bulk RNA-seq (82 responses), single cell genomics (52 responses) and DNA-seq data (41 responses). Satisfaction levels seemed high across 15 areas surveyed, including core data structures, annotation resources, training material, and website. Responses "Very Satisfied" and "Satisfied" combined dominated all other levels of satisfaction reported. The survey probed on barriers to contributing, and the use of S4 class discipline was mentioned a number of times. Freeform commentary indicated community interest in increasing ways to increase female participation, to reach more wet lab researchers, and to increase training material on S4 to help new developers.

- The creation of a Community Advisory Board (CAB) was objectives of enabling productive and respectful participation by Bioconductor users and developers at all levels of experience, and approved at the November 2019 TAB meeting. A draft CAB governance document is in development, and includes empowering user and developer communities by coordinating training and outreach activities. Implementation will begin with appointment of up to 8 members by the Chair of the Technical Advisory Board, in consultation with the Technical Advisory Board and solicitation of nominees from the Bioconductor and broader community. Ultimately the CAB should consist of 8 to 15 members, nominated to serve fixed terms by the current CAB. The CAB membership should provide a broad representation of the Bioconductor scientific community, including emerging and established researchers; drawing from individuals with interests spanning biological, statistical, and computational science, including representatives from relevant software and other communities. The CAB will strive for diverse representation of gender, ethnicity, geography, and other aspects of the Bioconductor community.

Enhanced opportunities for training and outreach. This grant has enabled annual national (BioC 2005–2019; funding supporting speaker, advisory board, and core team travel and accommodation) and international (BiocEurope, 2005–2019, BiocAsia, 2015–2019; funding supporting travel of key individuals) conferences, as well as regional workshops; extensive training resources developed for these events are available at https://bioconductor.org/help/course-materials/. Additional outreach activities include the monthly developer forum, and the ongoing review process for proposed contributed packages.

Accomplishments part 3: Review in the context of prior specific aims

The previous U41 proposal (funded 2015-2021) had several cores devoted to production, management and dissemination, and training. The production core specified five specific aims broken into 37 distinct features. We briefly review accomplishments and limitations encountered in executing these tasks.

Prior Aim 1: Maintain stable repository and archive of Bioconductor packages. In 2017, the main source code control system transitioned to git from subversion. For all packages, git branch `master` is the current devel image, while branches named `RELEASE_X_Y` are frozen images of prior release sources. The master branch for each package is transitioned by project staff to its `RELEASE` version every six month. Developers can make changes to `RELEASE` versions to enhance documentation of fix bugs, but otherwise API changes such as enhancement are not allowed for `RELEASE` versions. Efficient international distribution is accomplished using AWS CloudFront. *BiocManager* was contributed to CRAN to allow trusted installation in any R session via `install.packages("BiocManager")` using base R. The version of BiocManager will match the version of R in use, and will install packages suitable for that version of R. BiocManager includes code to assess consistency of the currently installed package set. *BiocPkgTools* was introduced to support user-level analysis of package interdependencies.

Aim 1 additionally addressed management of web presence, enhancements to persistent resource discoverability, and enhancement of documentation. The web site has undergone revisions to include a twitter feed and to present live excerpts from the support site. Automated minting of digital object identifiers (DOI) for all packages was introduced in 2017. Documentation enhancements include the extensive renovation of the *BiocStyle* package which supplies infrastructure for taking advantage of markdown composition of vignettes and workflows. The 2015 proposal included a plan to increase availability of training videos for the project. The YouTube channel `https://www.youtube.com/user/bioconductor` is available, but content is currently sharply limited.

Prior Aim 2: Enable access to curated reference data resources. Versioned genome, gene, and functional reference annotation are of immense importance for interpreting readouts from genome-scale assays. Through prior Aim 2, the *AnnotationHub* package has matured to include, as of 1/9/2020, 47,241 resources on 2,262 species. A sense of the convenience and scope of the resources provided with this package can be gleaned from Figure 10. When *AnnotationHub* is installed, a SQLite database of 100MB or so of metadata is placed in a dedicated cache. A `query()` method is available to search the metadata for species or data types available. The user obtains the cDNA sequences for *Z. alibicollis* with the expression `rtracklayer::import(ah[['AH78365']])`.

Additional work accomplished under previous proposal Aim 2 includes a) an *AnnotationHub* vignette detailing end-user creation of annotation packages for model organisms or data types not present in the current Hub, b) a protocol for end-user contribution to the AnnotationHub, c) improvement of facilities for passing outputs of external web services to users in familiar self-describing structures, d) progress in comprehensive support for new reference genomes such as GRCh38, and e) general robustification of hub and file cache management.

Prior Aim 3: Enable user support and user/developer interaction through accessible channels. The primary user support site, support.bioconductor.org, repurposes BioStars.org infrastructure, thanks to collaborative effort contributed by Istvan Albert, the head of the BioStars team. The support site includes a search facility, includes a web-email interface so that contributors are notified of responses to questions and comments, and computes reputation metrics for contributors. Issue tracking for newly contributed packages is achieved with a GitHub-based protocol, and more comprehensive issue tracking methods will be introduced in the next project period

Prior Aim 4: Provide robust developer support throughout the software life cycle. We maintained the `bioc-devel` mailing list for technical questions concerning package development and maintenance. The number of subscribers to the list has grown from 900 to 1,500 over the past six years. A new system, the ”single package builder” for building and testing newly contributed packages on Linux, macOS, and Windows platforms, has been introduced for the releases 3.8 and beyond. The general build system has been enhanced to support comprehensive nightly testing of over 1,800 software packages. We have added an additional layer of static analysis of contributed packages with the *BiocCheck* package, which functions like the native static analysis tool, `R CMD check`, but adds additional criteria for package evaluation including conditions on code structure, thoroughness of testing, and resource consumption for build and installation processes. The ”for developers” section of the web site (https://bioconductor.org/developers/how-to/git/) includes guidance on best practices for working with git and GitHub in the context of the Bioconductor package ecosystem. A protocol has been established to manage package ”end-of-life” practices, so that the developer and user communities are made aware of removal of a resource, and given an opportunity to adopt it for maintenance. Packages are deprecated (usable, but with warnings of

Figure 10: Transcript of acquisition of AnnotationHub metadata, and import of cDNA sequences for *Z. albicollis*.

```
> ah = AnnotationHub::AnnotationHub()
> ah
AnnotationHub with 47241 records
# snapshotDate(): 2020-01-09
# $dataprovider: BroadInstitute, Ensembl, UCSC, ftp://ftp.ncbi.nlm.nih.gov/g...
# $species: Homo sapiens, Mus musculus, Drosophila melanogaster, Bos taurus,...
# $rdataclass: GRanges, BigWigFile, TwoBitFile, Rle, OrgDb, EnsDb, ChainFile...
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., object[["AH5012"]]

            title
  AH5012  | Chromosome Band
  AH5013  | STS Markers
  AH5014  | FISH Clones
  ...       ...
  AH78365 | Zonotrichia_albicollis.Zonotrichia_albicollis-1.0.1.cdna.all.2bit
  AH78366 | Zonotrichia_albicollis.Zonotrichia_albicollis-1.0.1.dna_rm.tople...
  AH78367 | Zonotrichia_albicollis.Zonotrichia_albicollis-1.0.1.dna_sm.tople...
  AH78368 | Zonotrichia_albicollis.Zonotrichia_albicollis-1.0.1.ncrna.2bit
  AH78369 | Genome wide annotation for Myxococcus xanthus DK 1622

 > rtracklayer::import(ah[['AH78365']])
loading from cache
DNAStringSet object of length 23661:
         width seq                                               names
     [1]  5097 CTCCCCTCCTCCGCTCCCTCTC...ACCATTATGTGTGAGTGTGCTT ENSZALT00000000002.1
     [2]  5034 CTCCCCTCCTCCGCTCCCTCTC...ACCATTATGTGTGAGTGTGCTT ENSZALT00000000003.1
     ...   ... ...
[23660]  4350 ATGGCCAGAACTTGCCAGAGTG...TGGGATGGGGAGGTTTAAGTGA ENSZALT00000028827.1
[23661]   651 ATGGCGACGCGCGACGGCACCG...CCCCCGGCCTCAGTGGAACTGA ENSZALT00000028891.1
```

imminent removal) for one release and then removed. In this aim it was proposed that an image of the main build system would be made available for developers so that adverse events in package testing could be exactly reproduced by developers on demand. There has been progress on this development but it is not yet available.

Prior Aim 5. Facilitate cloud-based and reproducible analysis through virtualization. We maintain publicly accessible AWS Amazon Machine Instances pre-loaded with runtime support and CRAN and Bioconductor package sets, along with Dockerfiles and Docker container images registered in Dockerhub. Thanks to our archiving of sources for all releases in git repositories, users of these virtual resources can employ the matched versions of the BiocManager package to acquire compatible versions of any desired packages. It has become feasible to use the docker images to avoid tasks of manually provisioning and building R and package sets, but widely prevalent security-related restrictions on using Docker in the scientific and commercial communities prevent very wide adoption of this approach.

Research to improve the resource

Overview

We propose that research activities directed at improving Bioconductor's achievement of its core aim of promoting analysis and comprehension of high-throughput genomic data in the context of a five-year plan should have the following objectives:

- **Grow the workforce of individuals capable of reliably analyzing genomic data**. The main impetus to

the project some 20 years ago was reduction of barriers to entry for collaborative work between genome biologists and data scientists. Barriers still exist, mostly because of the rapid pace of biotechnological innovation. We also recognize the speed with which the technical milieus for training, communication, resource development and distribution are themselves changing. We must take steps to attract new users and developers through incorporation of new ideas and methods of bioinformatic and general information science, statistical science for high-dimensional data, and computational innovations for scalability.

- **Maintain high satisfaction of the thriving user and developer base already assembled in Bioconductor.** New research in the project must address real needs of practicing biologists and data scientists who already appreciate and rely upon the resource. Two directions of new work in this area are increased effort towards formal assurance of software reliability and implementation of improved approaches to continuous integration and continuous delivery discipline, to meet needs of hardworking contributing developers.
- **Improve community readiness to tackle new challenges in genomic data science as the vision of biodata analysis enhanced by artificial intelligence and machine learning (BioData AI/ML) comes to fruition.** Two key areas in which research progress is needed are statistical interpretation of massive data resources generated in single cell genomics, and efficient methods for reliably using and growing analysis ecosystems for federated data commons.

Project investigators and subcontractors work actively in all three of these areas. Our research plan for improving Bioconductor resources has four main components: formalizing governance and change management for key infrastructure classes and packages, hardening core infrastructure with increased unit test coverage, contributing to the theory and practice of data/analysis commons design and operation, and contributing to genomic data science methodology.

Research component 1: Governance of core infrastructure classes and packages for the Bioconductor software ecosystem

Research on principles and metrics for software ecosystem health and effectiveness is very limited to date. One of the most basic considerations developers must make in contributing code to a larger project is whether they will be free to introduce design and functionality changes that are incompatible with past operating and use patterns. Such changes are called ”breaking”, or ”API breaking”. In a symposium paper on Foundations of Software Engineering, Bogart and colleagues [12] studied the developer groups for the Eclipse IDE, CRAN, and Node.js and considered the questions: How do developers make decisions about whether and when to perform breaking changes and how do they mitigate or delay costs for other developers? How do developers react to and manage change in their dependencies? How do policies, tooling, and community values influence decision making? They concluded that

> community values play an essential role in shaping a software ecosystem, yet they can be somewhat difficult to distill from the outside .... Making community values and the involved tradeoffs explicit and transparent can help to ensure that all stakeholders understand the tradeoffs of decisions made by the platform and the accepted consequences, such as higher costs for certain stakeholders or reduced attractiveness to newcomers.

Furthermore, the authors distinguished the three software projects' approaches to ”breaking APIs” as: in Eclipse, you don't, in R/CRAN, you reach out to affected downstream developers, and in Node.js/npm you increase the major version number.” In this section we describe conditions in Bioconductor that propel us to formalize code change management governance procedures for core infrastructure.

Package dependency structure. Figure 11 depicts interdependencies among packages that define key infrastructure, annotation, and analytical capabilities of Bioconductor. Graph nodes are packages, and directed edges represent directed “dependencies”. For example, the *SingleCellExperiment* package declares an explicit dependency upon *SummarizedExperiment*.

R package interrelationships have three major forms: If package A declares a *dependency* upon package B, then when A is loaded, so is B, and all functions provided by B are visible to the user of A. If package A declares that it *imports* package B, then functions defined in A may use functions defined in B to do their work, but functions in B are not visible to the user unless B is explicitly loaded. This approach helps reduce “namespace pollution”

	Package	purpose	sdeps	gdeps
1	Biobase	basic infrastructure	510	568
2	S4Vectors	improved abstractions	441	449
3	GenomicRanges	genomic coordinates	427	453
4	BiocGenerics	shared methods	412	614
5	IRanges	interval algebra	403	424
6	SummarizedExperiment	coordination	302	328
7	Biostrings	sequence analysis	235	257
8	GenomeInfoDb	reference nomenclature	232	244
9	limma	differential expression	209	269
10	BiocParallel	parallelization	200	213

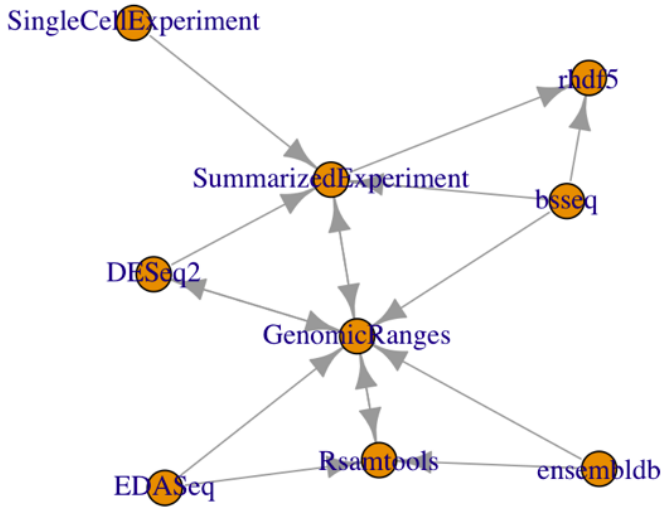
	Package	purpose	sdeps	gdeps
1	VariantAnnotation	VCF, SNPs	60	75
2	DESeq2	RNA-seq	57	90
3	SingleCellExperiment	scRNA-seq	54	65
4	flowCore	flow cytometry	47	51
5	annotate	PubMed, MIAME, GO, ...	43	58

Table 1: Top: Top 10 Bioconductor packages, ordered by number of dependent packages. Bottom: 5 Bioconductor packages with moderate numbers of dependent packages. Column 'sdeps' gives the number of packages for which software functionality is dependent on the listed package, 'gdeps' gives the number of packages for which software functionality or documentation production is dependent on the listed package.

that occurs when too many packages are loaded. If package A declares that it *suggests* package B, this means that documentation and examples of package A can make use of functions in package B. The documents and examples cannot function unless package B is installed, but functions in package A do not rely on availability of B for their operation.

Edges in Figure 11 do not distinguish the various relationship types. The bidirectional edge linking *Rsamtools* and *GenomicRanges* represents the fact that *Rsamtools depends* upon *GenomicRanges*, while *GenomicRanges* only *suggests Rsamtools*. This means that *GenomicRanges* documentation makes calls to functions in *Rsamtools*, but none of the functions of *GenomicRanges* rely on *Rsamtools* to do their work. Note that circular dependencies are not permitted.

Figure 11: Interdependencies in a subset of Bioconductor packages.



The message of Figure 11 is clear. From the perspective of general maintenance of all packages and their documentation, packages like *GenomicRanges* that have high indegree can, when modified, have substantial impacts on many other packages in the software collection. It is therefore essential to have the capacity to do comprehensive testing of change effects before changes are pushed into the ecosystem.

Scope of package interdependencies. As of January 14 2020, there are 1,813 software packages, 955 annotation packages, and 381 experiment packages in Bioconductor 3.11 (under development). Table 1 lists the top 15 packages, ordered by number of packages naming the package in the 'provider' column as a 'Depends' or 'Imports' dependency. Column 'sdeps' gives the count of the number of packages with depends/imports dependence on provider, 'gdeps' gives the count of the number of packages with depends/imports/suggests or enhances dependency. See the recent work of Su and colleague [13] for details on how these tables are computed.

Most of the packages in Table 1(top) are actively managed by Bioconductor core members and programmers. Table 1(bottom) provides a view of packages with moderate dependency profiles. There are 4 different maintainers for packages listed in Table 1(bottom).

The significance of change management for components of the Bioconductor ecosystem is fairly self-evident, but the open development nature of the project imposes limits on formal control of change processes. Unit testing is encouraged, but test coverage is variable from package to package.

Research plan. Bioconductor version 3.14 will be released in Q1 of the project. In Q2-Q3 we will enumerate, in conjunction with the Technical Advisory Board (TAB), essential core packages in this release to be managed through formal governance procedures. These will include a set for which change management *must* include TAB signoff, such as *GenomicRanges*, *SummarizedExperiment*, *SingleCellExperiment*, *Biostrings*, *AnnotationHub*. Changes to these packages can be proposed through demonstration of functional improvement and non-regression on a new github code branch. An additional set of highly used packages will be identified for which change management oversight by TAB will be offered on a voluntary basis. This set would likely include *DESeq2*, *edgeR*, *iSEE*, *tximport*. Processes for API change oversight by the TAB will be added to the TAB governance document.

Research activities to be conducted in this element of the project include analysis of the history of changes to package code through github logs, assessment of frequency of warning and error conditions for provider and dependent packages, and qualitative evaluation of interactions between package maintainer/developers and TAB when change oversight events occur. Longer term work in this domain will address the large-scale structure of package dependencies and API-breaking change events within the Bioconductor ecosystem as a whole, and between CRAN and Bioconductor packages.

Research component 2: Monitoring and increasing test coverage to enhance reliability of core infrastructure packages

Figure 5 is a brief report of ”test coverage” for a key Bioconductor infrastructure package, *SummarizedExperiment*. The report shows that, for example, every line of code in the ‘findOverlaps-methods‘ module is executed in one or more of the unit test, manual page examples, or vignette components of the package, while only about 60% of the lines dealing with the ‘RangedSummarizedExperiment‘ class are executed. The purpose of aiming for 100% coverage is to ensure that opportunities to detect errors in routine daily quality checking of package code, before users encounter these errors in actual work, are not missed.

While integrated unit testing for R packages is not new (*RUnit* dates back to 2004, *testthat* to 2009), no systematic approach to measuring scope or quality of testing of packages in the Bioconductor ecosystem has been undertaken. The *BiocCheck* utility static analysis does inform if there is no evidence of unit testing anywhere in a package, but no assessment of test coverage is routinely undertaken.

Research plan. In Q4 of the project we will prepare a comprehensive report on test coverage and vulnerabilities of key infrastructure packages, and will create a process for mapping risks and planning mitigation of vulnerabilities discovered. Because high test coverage can increase time consumed in build system operations for a package, estimates of impact of increased test coverage will be needed. Avoidance of redundancy (e.g., situations in which examples and unit tests exercise the same code in the same way) will be important for efficiency and analysis of *covr* reports can help identify such redundancies. We will state goals for increased coverage and quality of tests in Q4-Q5, and in Q6-Q8 we will encourage staff and contributors to meet these goals. Subsequent to Q8 new procedures for measuring and achieving minimal acceptable test coverage conditions will be implemented for new contributions to the Bioconductor project.

Research component 3: Contributions to the theory and practice of design and operation of data/analysis commons

Championing a vision of federated data architecture for health sciences research that spans four decades, Grossman [7] defines a data commons (DC) in terms of governance agreements for data access and system usage, data models describing DC contents, a community-based process for data contribution, affordance of pipeline-based analysis, and pursuit of interoperability across multiple independent DCs.

Research underlying methods for implementing and optimizing DCs is intrinsically interdisciplinary. Development and operation of DCs introduce cutting-edge problems in law and sociology, data structure and algorithm design, and systems analysis. It seems fair to say that research literature on DC design and use is very limited. Some relevant prior work includes that of Bar-Sinai and colleagues on privacy profile formalisms [14], Siu et al. on responsible data sharing [15], and Karp et al. on ethics of data aggregation [16]. In this section we will describe research to be undertaken to better understand how a dynamic and open software ecosystem can interact with data/analysis commons systems in mutually beneficial ways.

Assumptions. We assume that we are working with commercial cloud infrastructure, possibly bound to a specific vendor to achieve particular efficiencies. Cloud-agnostic solutions are preferred, other things being equal. Users will need to be aware of costs of performing certain tasks, although charges to be incurred for new tasks will be at best partly predictable. The basic framework for users and developers involves the notion of a shareable workspace with authentication processes for access to data, code and/or compute resources. A versioned compute environment can be considered ”containerized”, and data and software resources can be mapped to cluster nodes where containers are run on commercial cloud hardware assets.

Data commons. We will use DC to denote a data commons. Let (X,Y) denote a pair of data commons resources, with X to be regarded as information supporting prediction of Y. We use $a \to B \to c$ to indicate that B operates on a to produce or contribute to c. Let I denote the process by which data resources are ingested into DC, Q denote a query on DC that yields a data resource, by filtering and combining previously ingested resources, and f denote a computational analysis of data resources, culminating in an analysis output denoted Ω. The three fundamental processes involved in populating and using a data commons are ingestion $[(X,Y) \to I \to \mathrm{DC}]$, query resolution $[\mathrm{DC} \to Q \to (X,Y)]$, and analysis via application of a function f $[(X,Y) \to f \to \Omega]$.

Analysis ecosystem. An analysis ecosystem (AE) is a family of software components $f_1, \ldots, f_N$ satisfying conditions of adequacy and interoperability. A candidate component f_{cand} is accompanied by a suite of self-tests t; when these are passed, the component is denoted f_{elig}. The eligible components are jointly subjected to a set of interoperability tests T; when these are passed, the f_{elig} is added to ecosystem $\mathfrak{F}$. Symbolically,

$$f_{\mathrm{cand},i} \to t_i \to f_{\mathrm{elig},i}$$
$$(f_{\mathrm{elig},1}, \ldots f_{\mathrm{elig},N}) \to T \to \mathfrak{F}$$

In the pursuit of federated computational environments for genomics, standards and metrics for evaluating combinations and operations of DC and AE will be essential for decisionmaking and resource planning. We will devote significant effort to a) fleshing out the concept of ”analysis ecosystem” and b) evaluating approaches to implementing operations Q and f, the fundamental components of an analysis ecosystem. Several observations motivate these concerns.

- The schematics presented ignore the aging of DC/AE contents and components. Time is a fundamental parameter of the process of managing a DC/AE. Versions of a DC/AE *per se* need to be distinguished from the versions of its constituent data elements.
- The environment in which $f(X,Y)$ is computed has finite capacity and will likely be shared by multiple analytical workflows. A process is needed for ensuring that the resources allocated to the computation of $f(X,Y)$ are minimally sufficient *for its entire duration*.
- Computation of Q and f will incur costs dependent on their detailed design in relation to the DC and analysis environments. Tradeoffs between software design costs and execution costs will be present, and cost profiles in both design and execution domains are likely to change over time. Facilities for accurately estimating and strictly bounding costs of DC/AE usage are essential.

The Bioconductor project has a fundamental interest in ensuring that emerging approaches to federated genomic analysis benefit from the approaches that underlie the project’s durability and impact to date. We propose the following research projects for advancing this interest.

Research Component 3 Plan, part 1: Extension of the R package repository concept to container-based environments. Users of R/Bioconductor are accustomed to rapid revisions of their active package set. Expectation of real-time updating is supported by the *BiocManager* functions for checking for package updates, resolving dependencies, and by the rolling updating practices of the CRAN-style repositories from which packages are retrieved. For Linux users, packages are usually distributed in source form, and it is assumed that the receiving system has all necessary compilation and library runtime infrastructure to accomplish installation from source. For macOS and Windows users, statically linked binary versions of packages are assembled on a regular basis at repositories, for distribution to those platforms. The user invokes the installer program which selects from the repository the package appropriate to the requestor’s system. Currently binary package distribution is fairly coarse-grained, with macOS 10.11.6 and Windows Server 2012 R2 the basic environments that are supported. These limitations arise from the necessity of compatibility with binary packages built at CRAN, where these specific platform versions are used. As containerization becomes more prevalent, it will be attractive for users to have access to archives of compiled versions of compatible packages, because installation ”from source” can

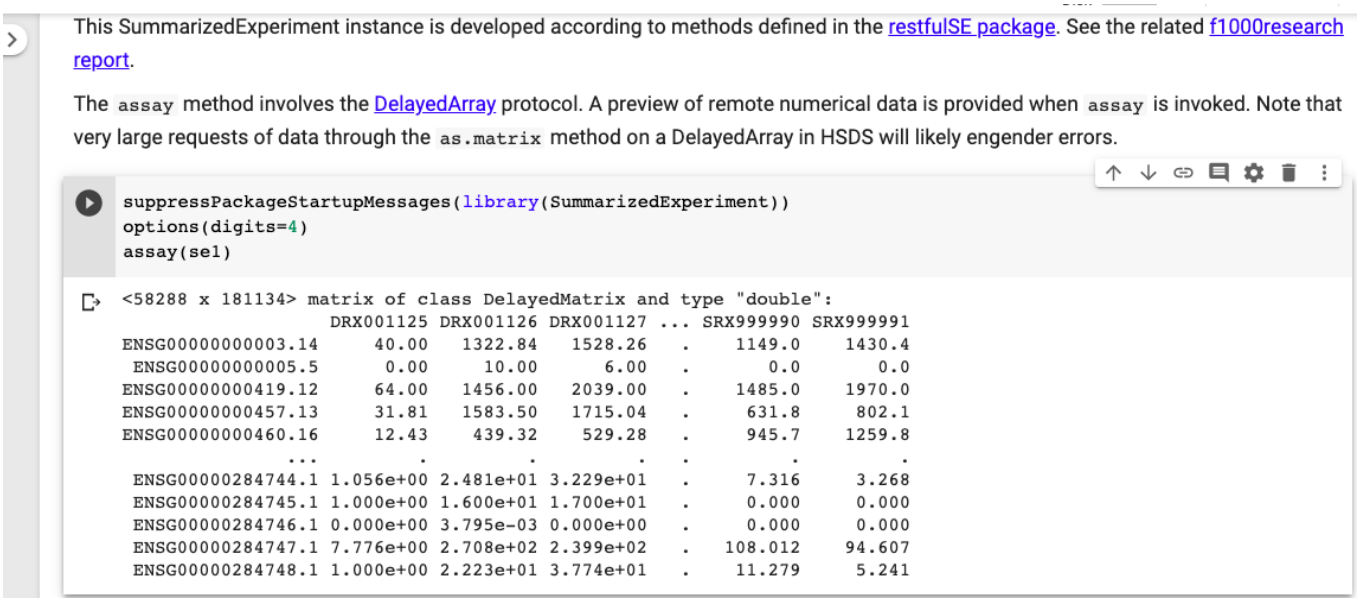


Figure 12: Screenshot of material in the Google Colaboratory notebook at https://tinyurl.com/vr9zgre.

be fairly time-consuming, and, for a given package version and container type, should not have to happen more than once for the community of users of that container type. We have seen, in the discussion of Production Feature 19 above, how archives of compiled packages, matched to the runtime environment used for Google Colaboratory notebooks, can be retrieved from cloud storage and used to accelerate notebook execution. We will create an add-on package for use with instances of R in containers that will allow installation procedures to efficiently identify, verify, and install binary-compatible compiled packages in container-aware repositories. This work will be carried out in Q5-Q8 of the project timeline. It is inevitable that in some situations, users will need to compile packages from source to run their custom workflows, and this will always be possible, but our objective is to eliminate the need for compilation from source whenever possible.

Research Component 3 Plan, part 2: Lazy data access concepts for cloud-scale resources. The Colaboratory notebooks discussed in Feature 19 provide proofs of the concept that rich software stacks can be acquired and used on demand in interactive analysis workspaces. Figure 12 is a screenshot of a segment of the notebook that provides a preview of assay quantifications for a compendium of 181,000 uniformly processed human RNA-seq studies derived from the NCBI Sequence Read Archive. The data are managed in the HDF Scalable Data Service HSDS and code in the *rhdf5client* and *restfulSE* packages mediates between the data service and users, so that the remote resource can be interrogated with familiar programming patterns. In this research component, we will interface Data Commons data model frameworks such as the Gen3 system (https://gen3.org) to cloud-scale array and metadata storage frameworks like HSDS and Zarr, mediated by Bioconductor's SummarizedExperiment class and its extensions. An important step forward will be to enhance client-side configuration of array access functions to take advantage of multiplexed query handling in HSDS.

Research component 4: synergy from data science method development

An important aspect of Bioconductor that makes it particularly successful is that members of the core team, as well as most package contributors, are actively engaged in scientific collaborations. By working directly with laboratory scientists developing and using the most cutting edge technologies to investigate important biological questions or clinical applications, our team has an in-depth understanding of the genomic research community's most urgent computational needs. Furthermore, since our team is submerged in data analysis, computing with data, and statistical method development, they also have a deep understanding of the needs and idiosyncrasies of the genomic data scientist community. Here we provide an example of a broad data science challenge we will tackle and incorporate into the project.

During the last decade, high throughput technology applications have expanded to include measurement of dynamic outcomes underlying genomic function in development and disease. Measurements related to functional elements that act at the protein and RNA levels, and regulatory elements that control gene activity, are at the core of studies undertaken by large consortia and individual labs alike. More recently, technologies capable of providing insights into the relevance of spatial organization to genomic function are being developed. To understand

the mechanisms of these dynamic processes, we need unbiased, high-throughput technologies which capture the spatial context of single cells along with their molecular identities. These new measurements introduce levels of variability that give rise to data analytic challenges related to distinguishing unwanted or uninterested sources of variability, from biologically relevant signals. Our group has extensive experience developing statistical tools for these applications and have authored some of the most widely used software in this area. We will leverage our experience to continue developing tools in response to urgent computational needs in genome science and help keep Bioconductor relevant to the research community. In particular, we will expand our work to include applications related to single cell technologies including new approaches that permit the study of spatial effects and interactions.

The advent of single cell technologies has markedly changed the nature of genomics data. As a result, we find that the modular approach of current pipelines, developed and successfully deployed for protocols based on bulk tissue, do not extend well to their single-cell counterparts. A particular shortcoming of many existing methods is the reliance on distributions associated with large counts, while single cell data is extremely sparse. A specific challenge is that current approaches do not properly account for high across-cell variability of total read counts and proportions of undetected genes. To overcome this and improve statistical inference rigor, we are developing unified approaches that leverage statistical models to account for technical bias, sparse data, and provide intuitive parameterizations that facilitate downstream analysis, such as clustering, differential analysis, and spatial effect estimation.

We note that widely used and valuable tools are available for the analysis of scRNA-seq data, for example Seurat [17] and ScanPy [18] provide tools for preprocessing, quality control, batch effect removal, clustering, visualization, and differential analysis. However, while existing tools have provided effective ways to process and analyze single cell data, new technologies, more complex biological questions, and the availability of increasingly complete datasets are posing new challenges. We also point out that we intend to collaborate with the authors of these tools to assure harmonization between software suites (See letter of support from Rahul Satija).

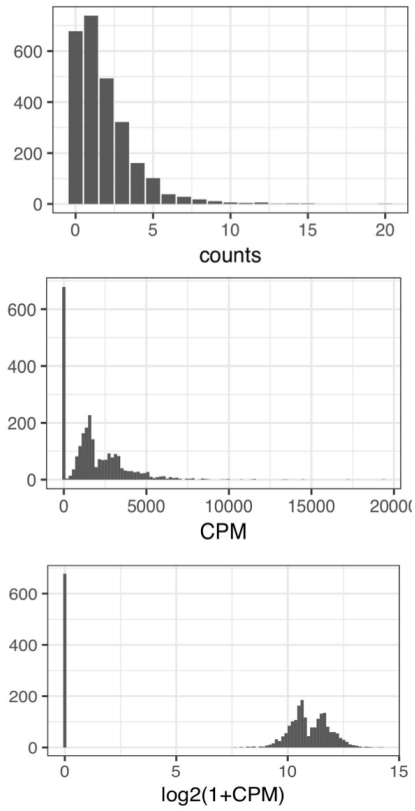


Figure 13: These three histograms show the artifact introduced by the current approach: UMI count data starts out following a Poisson distribution (top) and only after normalization (middle) and taking the log after adding a pseudo-count does the data appear to follow a zero inflated distribution.

As an example of the need for new methods for sparse data we show an example related to a common task in studies based on single cell data: using clustering algorithms to classify cells into biologically meaningful groups such as cell types. Because differences in experimental conditions between individual cells can lead to distortions that bias distance calculations, and as a result, impair clustering results, data analysis pipelines include a normalization step which seeks to adjust for these differences. Currently, the most commonly used normalization approach is to compute log(CPM+1). Specifically, raw counts (Fig 13) are normalized using sample-specific size factors: the total counts are divided by 10^6 (counts per million or CPM). These are then log-transformed in an attempt to reduce skewness. To avoid logging zeros, a small pseudocount is added. Current pipelines implement normalization in a modular way, meaning that they normalize first and then input the normalized results into downstream analysis such as clustering. To reduce the computational burden and to reduce noise, clustering algorithms include a dimension reduction step which is usually principal component analysis (PCA). As we summarize below, we have previously shown that, when applied to scRNA-seq data, this approach often leads to distorted clustering results and false discoveries [19].

A general framework we intend to use, models that use distributions appropriate for count data and account for level of variability within thins model. Specifically, we let y_{ij} be the observed UMI counts for cell or droplet i and gene j, and let $n_i = \sum_j y_{ij}$ be the total UMIs in the sample. We assume a negative binomial model with

$$y_{ij} \sim \text{NB}\left(n_i \exp\{\mu_{ij}\};\ \phi_j\right)$$

with μ_{ij} further modeled depending on the application and the ϕ_j are gene specific overdispersion parameters.

Note that including the n_i as an offset helps account for technical variation. We can now use this model for different applications. For example, for dimension reduction we assume that

$$\mu_{ij} = \alpha_j + \mathbf{u_i}^\top \mathbf{v_j}$$

with $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^L$ for $i = 1,\ldots,I, j = 1,\ldots,J$ representing principal components and loadings respectively, L a lower dimension relative to the original data matrix and α_j gene-specific mean levels. The number of latent dimensions L controls the complexity of the model. We have previously shown that a simple version of this generalized version of PCA (GLM_PCA), using the Poisson distribution, greatly outperforms the current default approaches [20] (Figure 14) by removing false clusters introduced by technical variability.

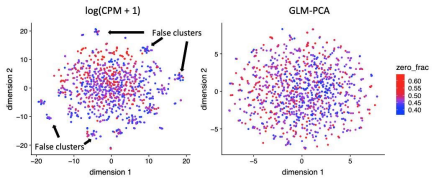


Figure 14: GLM-PCA can improve clustering results. The left pain shows a tSNE plot for negative control data normalized using current default. The right pane shows the result of applying our GLM-PCA method.

For differential expression analysis we adapt the model to

$$\mu_{ij} = \alpha_j + \mathbf{x_i}\beta_\mathbf{j}$$

with $\mathbf{X_i}$ a vector of covariates for cell i and $\beta_\mathbf{j}$ parameters for gene j. An example of a covariate is an indicator variable distinguishing the different groups. Note this is a very similar model to that used by edgeR and DESeq2. However, both these methods were motivated by bulk RNA-seq experiments and preliminary data suggest that the approach needs to be adjusted for single cell data, which has substantially different characteristics. Specifically, the way in which edgeR and DESeq estimate the gene-specific dispersion parameter ϕ_j was motivated by cases in which the number of samples (cells in this case) is small: a typical example was 3 cases and 3 controls. As a result the methods include computationally expensive multi-level statistical procedures to ensure that the estimates of ϕ_j are stable. In contrast, a typical single cell experiment will have hundreds or thousands of cells for each group. With this number of data points maximum likelihood estimation should provide a useful approach. However, we expect to need to implement multilevel models to avoid unstable estimates due the observed mean level being exactly 0 for entire groups.

Finally for spatial applications we will need to adapt the model to

$$\mu_{ij} = \alpha_j + f(x_i, y_i)$$

with f a smooth two-dimensional function and x_i and y_i the location of cell i. More complex versions of this model will be required to account for the fact that each cell i may be a different type. We will work with our collaborators to develop these models.

Incorporating this general approach into Bioconductor will require the implementation of new optimization algorithms that take into account and leverage the sparsity of the data. Methodological and computational innovation here will likely be applicable with other application producing sparse data. We expect this to increase as the single cell approach extends other molecular events, such as that provided by ATAC-Seq technology [21].

Additional Funding

The Bioconductor project has several sources of additional funding. These sources enable project development and impact beyond the scope of the current application. NIH / NCI / ITCR award U24CA180996 *Cancer Genomics: Integrative and Scalable Solutions in R/Bioconductor* enables cancer-related software development, providing: data representations for multi-omic single-cell analysis; curation and dissemination of cancer-related annotation and experiment data resources; and development of methods for prioritization of non-coding somatic variants. Our participation in NIH / NHGRI award U24HG010263 *Implementing the Genomic Data Science Analysis, Visualization, and Informatics Lab-space (AnVIL)* motivates Bioconductor development of strategies for cloud-based computation on large-scale data resources in a secure environment. Our Specific Aim 3 enhances the impact of this funding by making cloud-based strategies for scalable computation available to all R / Bioconductor users and developers; the transition to cloud-based computation is essential over the course of the proposed funding period. Bioconductor core team members receive seed funding from the Chan Zuckerberg Initiative (CZI). Dr. Morgan leads an international group of 8 researchers developing methods for accessing data and performing scalable single cell exploratory analysis on Human Cell Atlas data. CZI funding amplifies the impact of investments in the current proposal by enabling broader use of Bioconductor infrastructure, and by engaging members of the broader Bioconductor community in funded software development. Dr. Carey leads a short-term

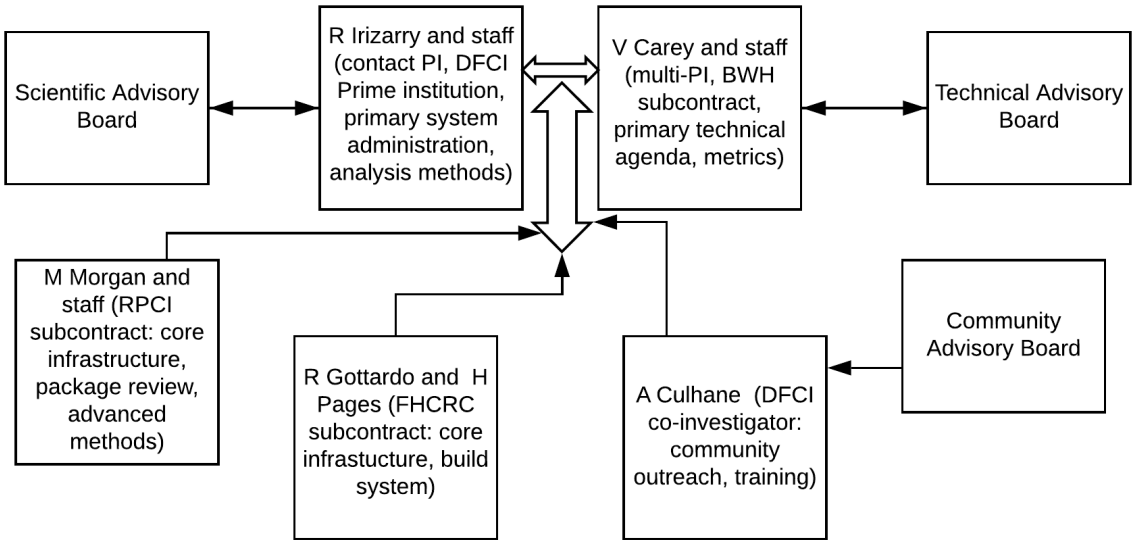


Figure 15: Organizational chart.

CZI sponsored Essential Open Source Software for Science project to explore innovative approaches to our build system; we envision this as an opportunity for innovative exploration of strategies for long-term evolution of the build system infrastructure that will ultimately be supported as the production environment of the current proposal.

Management

The basic organizational chart of the project is displayed in Figure 15. Drs Irizarry and Carey provide executive guidance to project staff and subcontractors, with input from Scientific and Technical Advisory Boards. Dr Culhane of DFCI has provided extensive effort in training and community outreach and will be primary liaison to the Community Advisory Board.

We have appointed a Scientific Advisory Board for the previous funding period. The composition of the Board included established scientists in genome sequencing, statistical methodology, industry leaders in scientific computing, and senior researchers in computer science. Important guidance has been offered by board members on numerous aspects of project performance and governance. We will continue to renew appointments to the Scientific Advisory Board to maintain coverage in these areas. Scientific Advisory Board members are asked to review annual reports and to meet with investigators at the time of the North American conference, in person if possible.

Dissemination

Our commitment and approach to dissemination of high quality software, pedagogic documentation in static and interactive forms, and curated, easily accessible and experimental data is described throughout the proposal, and is indicated particularly in production features 21 (maintain existing approaches to disseminating project resources through web-accessible package and workflow documents) and 25 (introduce support for podcast and video content presenting elementary and advanced genomic analysis methods). Pedagogic material is available on the ”Learn” pane of bioconductor.org, is present in every vignette of every package, and may be studied without installing any software. Packages of software, data, and annotation meet quality criteria and are accessible via ‘BiocManager::install‘, for immediate installation in a current R session. Public distribution of all scientifically relevant administrative materials (governance documents, Technical Advisory Board minutes) is facilitated through posts on the main web site. Source code for all software packages is disseminated in the open through git and github repositories.

Production feature 22 (continue conference and meetup series) is an approach to disseminating project methods and tools in an active way, and most presentation materials from all conferences (back to 2010) are made publicly accessible at https://bioconductor.org/help/events/.

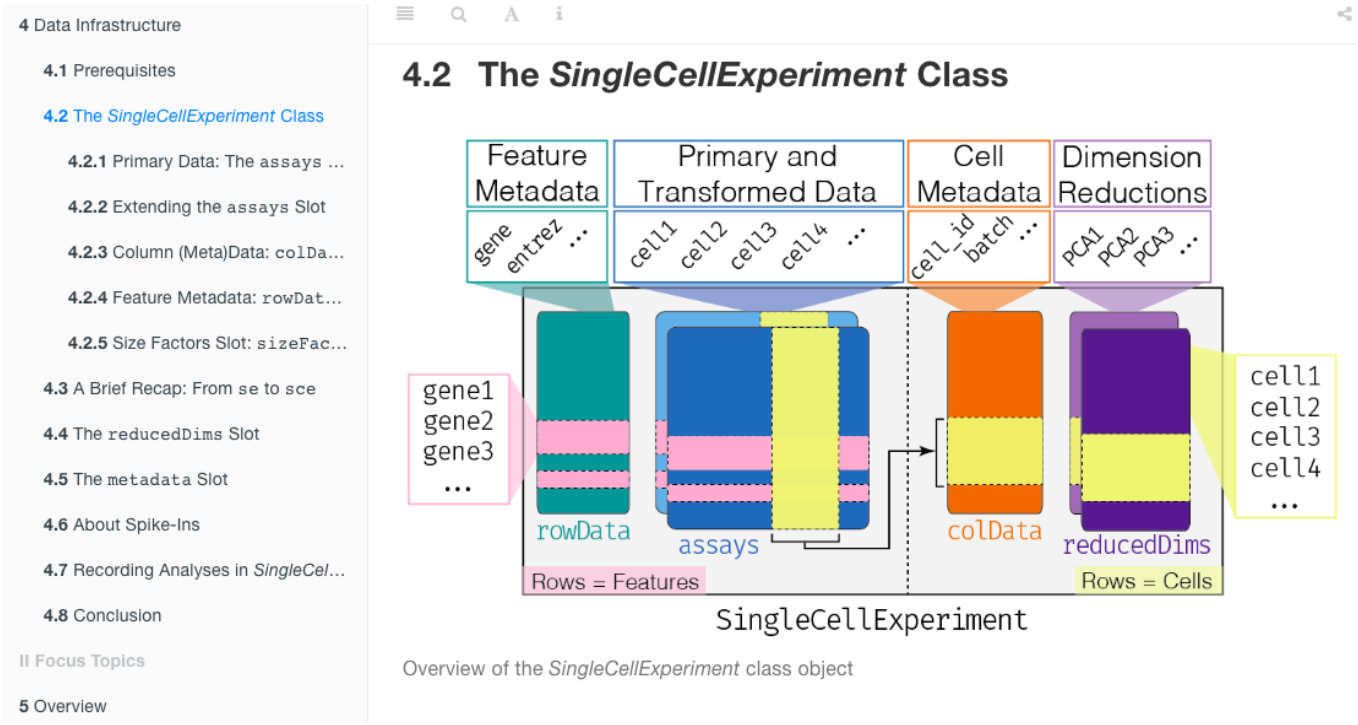


Figure 16: Screenshot of material in the online resource https://osca.bioconductor.org/, a continuously revised compendium on data analysis for single cell genomics with Bioconductor.

User training

The Bioconductor project has always placed a high priority on providing training for both practicing biologists seeking tools for analysis and comprehension of their experimental data and data scientists seeking to improve methods for processing and interpreting genome-scale assays. Bioconductor introduced the concept of package vignette as an integrated component of R package structure and installation. Vignettes are obligatory for all Bioconductor software packages, and the practice of including vignettes has grown in use outside the project. Vignettes are an efficient approach to transmission of detailed information about methods and tool operation, and serve readily as materials for course segments in genomic data science.

Online tutorial "monographs". A more recent approach to user training in the domain of single cell transcriptomics is depicted in Figure 16. The screenshot is a segment of a tutorial resource provided as an accompaniment to the recently published Nature Methods paper "Orchestrating single-cell analysis with Bioconductor" [22]. The sidebar is a slice of the table of contents, which currently includes 22 sections addressing methods concepts including normalization, clustering, cell type annotation, and cell cycle assignment, along with a series of workflows demonstrating processing and analysis with data from a variety of single-cell protocols including 10X, SMART-seq, and CEL-seq. This resource grew out of materials presented at the 2019 Bioconductor conference workshop (workshop archived at https://tinyurl.com/soqbpry). Project resources will be allocated to production and maintenance of resources of this type.

Hosted ready-to-run, ready-to-modify notebooks. We have discussed the use of jupyter notebooks in cloud-hosted analysis environments such as Google Colaboratory; see, e.g, Figure 12. Bioconductor workflows written in rmarkdown can be automatically translated to jupyter markup and hosted, providing capacity for immediate student interaction and modification to explore effects of changes to workflow data or parameters of workflow analysis processes.

Contributions to MOOCs. We will continue to add and update offerings to the edX MOOC platform so that students all over the world can take advantage of the array of documents and tools available in Bioconductor, to learn about and work with new concepts and experiments emerging in genome biology.