Packages, ecosystems, and services: Software resources and strategies for multimodal single-cell genomics

Vince Carey
BIRS Meeting June 2020

Road map

- "Chat race" game to warm up a little
- A triangular schema defining context of relevant work
- Brief review of scalability, divide and conquer, and the software ecosystem concept
- Some aspects of the Bioconductor ecosystem
- Some prospects for productive evolution of ecosystem management - an illustration with S. Davis' GHA workflow-4-workshop infrastructure

Chat races

Given: a "sub-quote" - a notorious phrase to which certain substitutions are applied

To win points, be the first to chat the name of the author of the original phrase

Example:

Notorious phrase: To consult the statistician after an **experiment is finished** is often merely to ask him to conduct a post mortem examination. He can perhaps say what the **experiment** died of.

Sub-quote for game: To consult computer scientists after a program has been coded is often merely to provoke them to propose that a different language be used. They can perhaps say why the original choice was a fatal one.

The answer here would be: Ronald A. Fisher

First race

Sub-quote: All <u>programs</u> are wrong; some are more useful than others

To whom is the original remark on which this is based attributed? Be the first to put their name in the chat

Second race

Sub-quote: Bioconductor is the most successful genomic analysis platform. Those who ignore it are condemned to reinvent it.

To whom is the original remark on which this is based attributed?

Third race

It takes a <u>software ecosystem</u> to raise a <u>compelling analysis in multimodal single cell genomics</u>

Who popularized the proverb from which this is derived?

Fourth race

I shall not today attempt to define "compelling analysis in multimodal single-cell genomics" but I know it when I see it.

What is the name of the Supreme Court justice who wrote the original quote in the context of an obscenity trial?

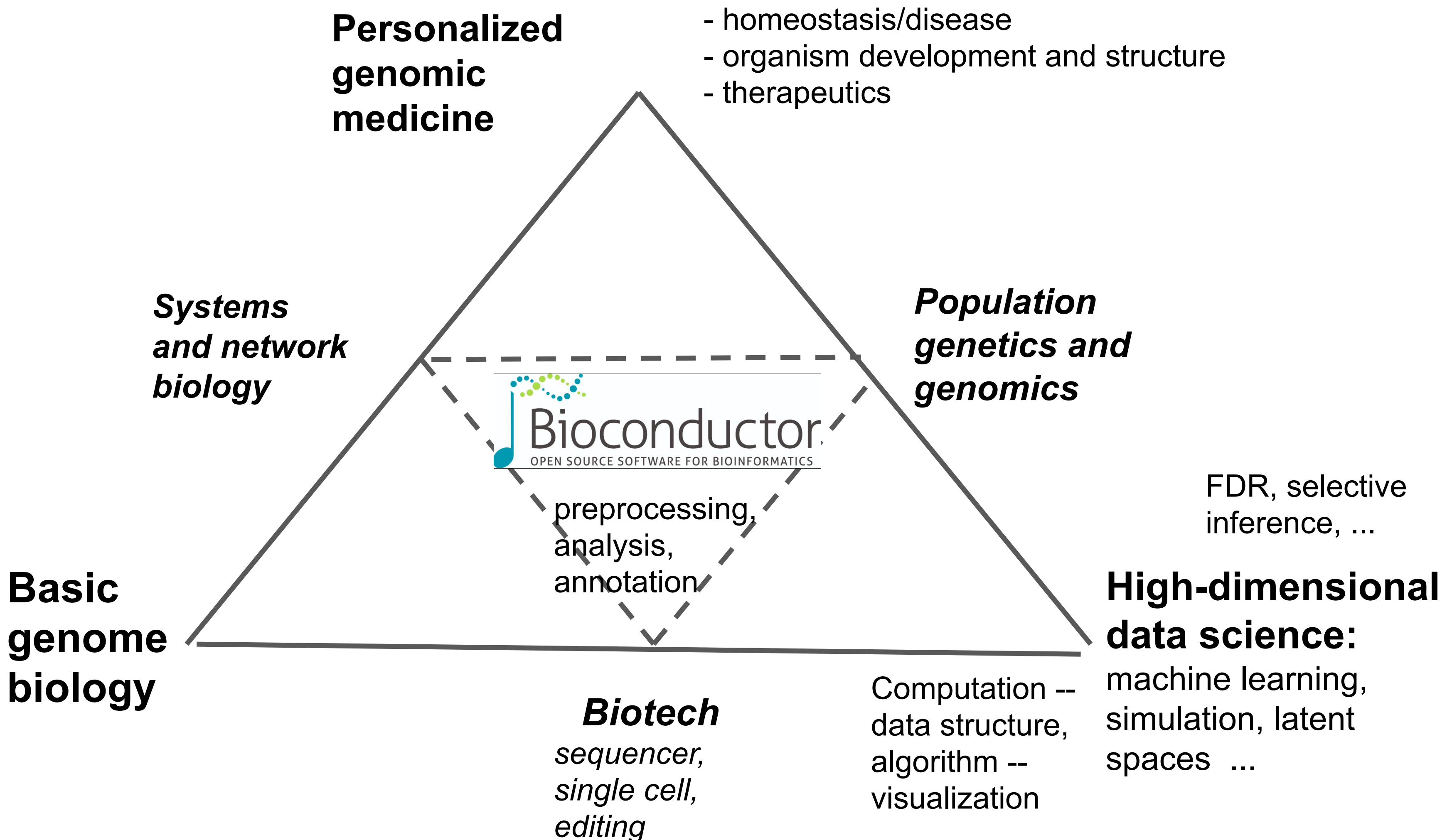
Fifth race

To create a Bioconductor package
is to battle with trolls in the vaults of WRE and the build system

To run BiocCheck over one's package
is to sit in judgment over oneself

THE GAME … I HOPE YOU ENJOYED THE GAME … I HOPE

It was not pure frivolity. Is "all models are wrong" true, but "all programs are wrong" false? Should Bioconductor be reinvented? What is a software ecosystem? What makes a multimodal single cell analysis compelling/reliable -- can we define it?   And what about BiocCheck? Where did its criteria for warning or error come from?


Personalized genomic medicine
- homeostasis/disease
- organism development and structure
- therapeutics
Systems and network biology
Population genetics and genomics
Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS
preprocessing, analysis, annotation
FDR, selective inference, ...
Basic genome biology
High-dimensional data science: machine learning, simulation, latent spaces ...
Biotech
sequencer, single cell, editing
Computation -- data structure, algorithm -- visualization

Some connections to ongoing discussions

- From benchmarking section:
  - Leverage single cell eQTL? (e.g. doi: 10.1038/s41467-020-14457-z) **Through GTEx**, we have a large set of known tissue and cell-type-specific eQTL to benchmark methods. We should expect to find relationships at single cell level when we see them in bulk.
    - **What's a good representation/API for GTEx?**
- From spatial/scProteomics:
  - Reuse concepts and structures in atmospheric modeling/GIS
    - **How do we coordinate global efforts on integrative data structure design and analysis methods?**
- **Can we plan first and then act, or do we always need to experiment first and then refine/synthesize?**
  - Example of TreeSummarizedExperiment

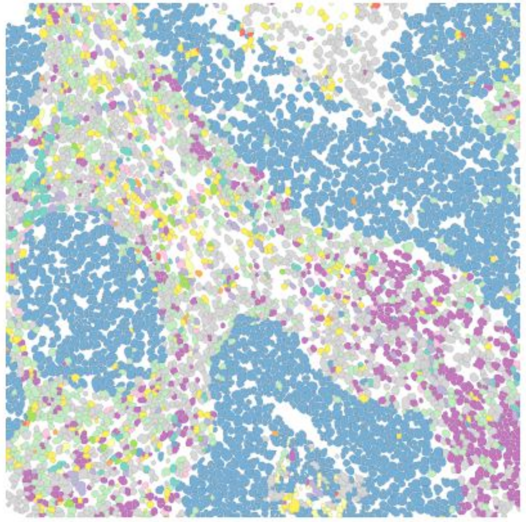
Interesting innovation in dynamic visualization: Kris

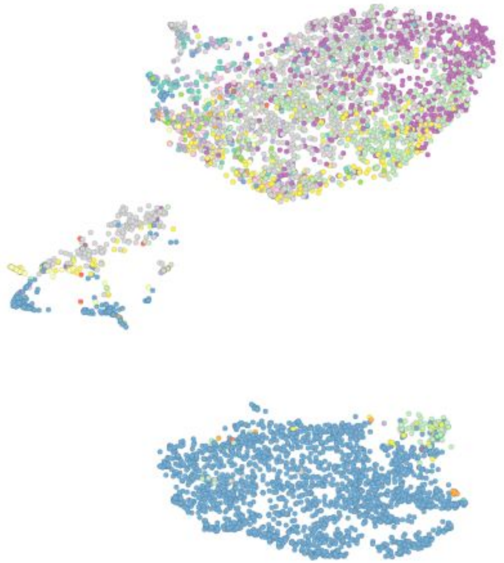
- Neighboring cells can have relatively different U-Map projections, at least with the current transformation / channels
- Immune cells at the boundary between tumor and immune have noticeably different expression (and lay elsewhere on the map)

The code to prepare the data is here.

The tumor clusters (from the original rda object) are 4, 7, 10, 17.

The immune clusters are 1, 2, 3, 4, 8, 10, 11, 12.



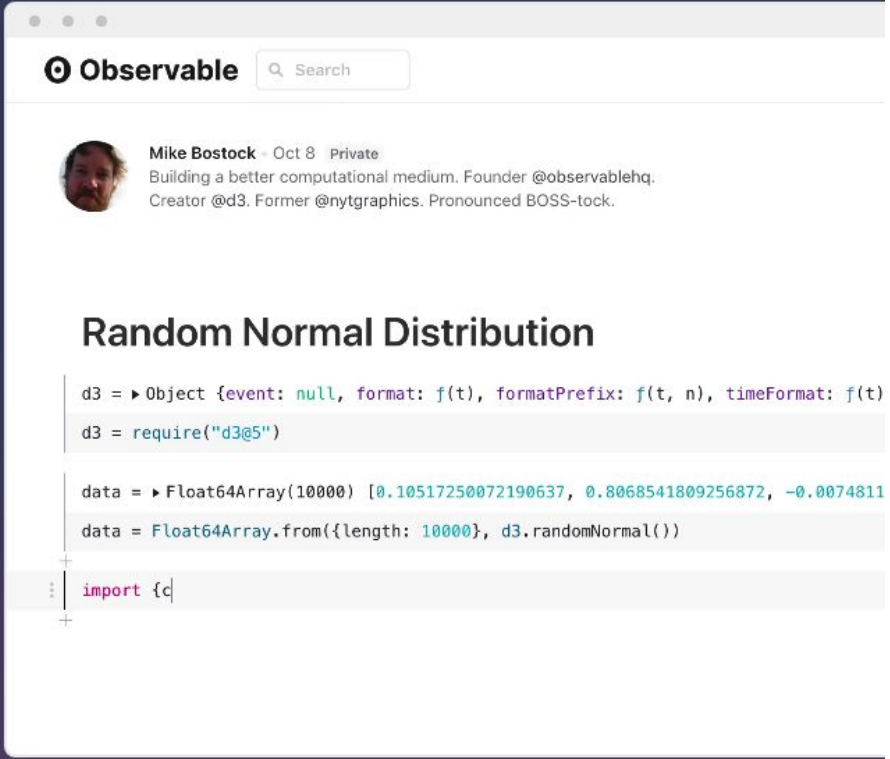


uses observablehq.com

Feature

Learn and reuse

- Designed from the ground up to **share ideas**.
- Code has never been easier to **find and reuse**.
- **Thousands of notebooks** to learn from and fork.
- Turn any notebook into a little library by **importing cells**.



Mike Bostock · Oct 8 Private
Building a better computational medium. Founder @observablehq.
Creator @d3. Former @nytgraphics. Pronounced BOSS-tock.

Random Normal Distribution

```
d3 = ▸Object {event: null, format: ƒ(t), formatPrefix: ƒ(t, n), timeFormat: ƒ(t)
d3 = require("d3@5")

data = ▸Float64Array(10000) [0.10517250072190637, 0.8068541809256872, -0.0074811
data = Float64Array.from({length: 10000}, d3.randomNormal())

import {c
```

Basic facts

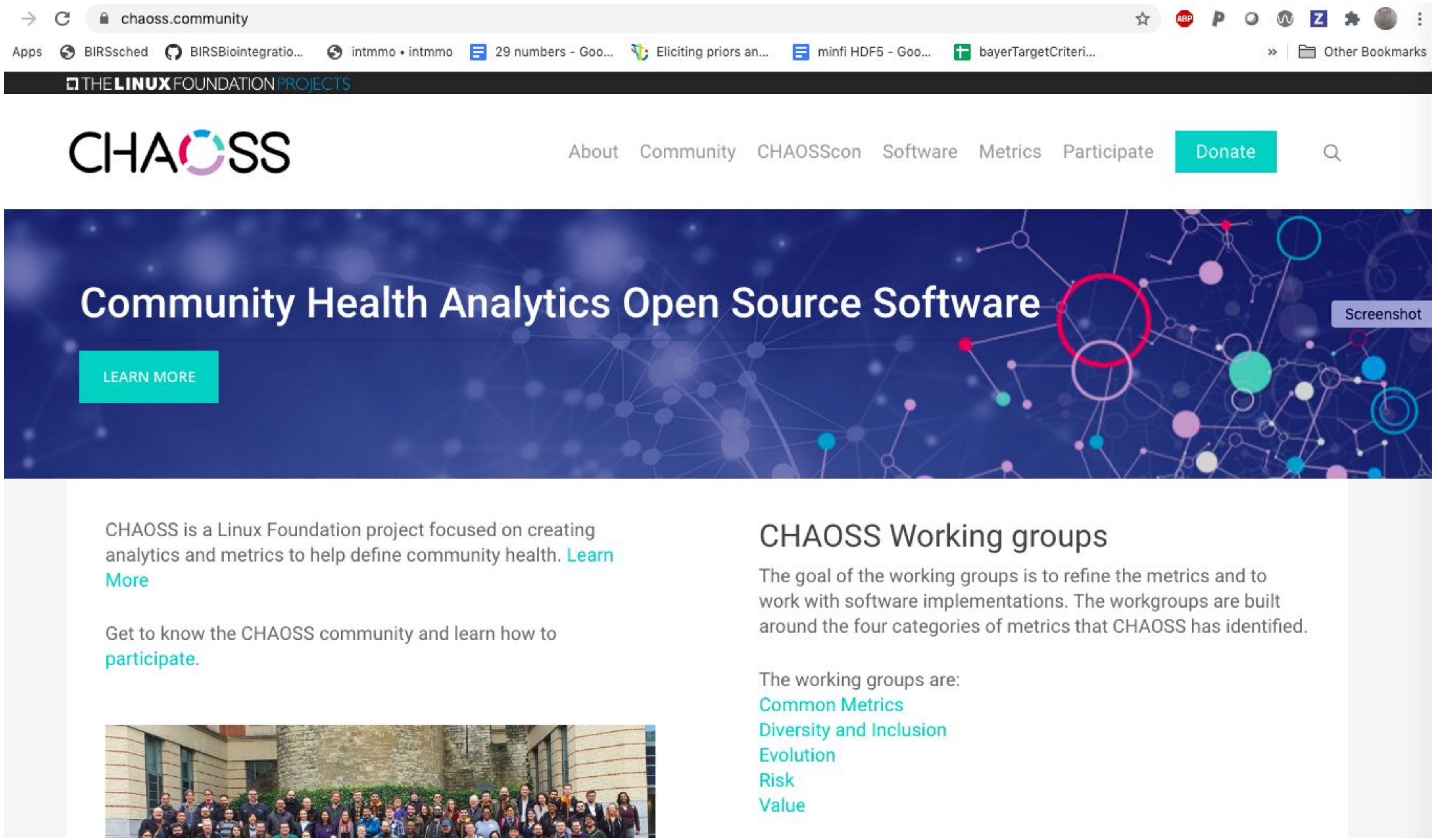
- It takes an ecosystem to make an analysis
- It takes an ecosystem to make an ecosystem
- Where do you begin?
  - Functions, packages, data structures, … ?
- Claim: Two elements are fundamental
  - Ecosystem builder
  - Evolutionary strategy
- Fun fact: software ecosystem health assessment is a thing!

Reviewing the Health of Software Ecosystems – A Conceptual Framework Proposal

Konstantinos Manikas and Klaus Marius Hansen

Department of Computer Science (DIKU)
University of Copenhagen
Njalsgade 128
2300 Copenhagen S
Denmark
{kmanikas,klausmh}@diku.dk

Abstract. The health of a software ecosystem is an indication of how well the ecosystem is functioning. The measurement of health can point to issues that need to be addressed in the ecosystem and areas for the ecosystem to improve. However, the software ecosystem field lacks an applicable way to measure and evaluate health. In this work, we review the literature related to the concept of software ecosystem health and the literature that inspired the software ecosystem health literature (a total of 23 papers) and (i) identify that the main source of inspiration is the health of business ecosystems while also influenced by theories



CHAOSS

Community Health Analytics Open Source Software

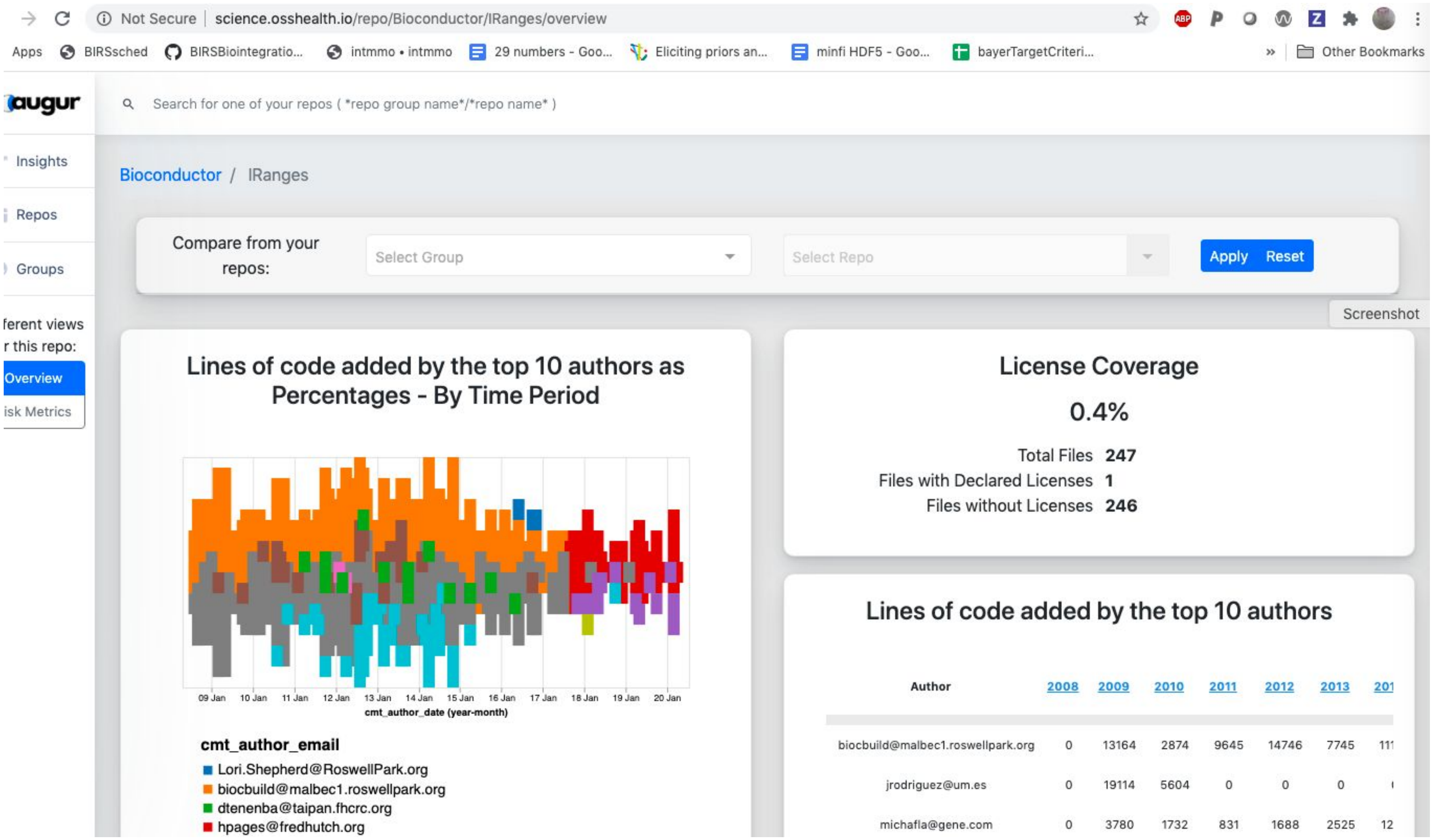
CHAOSS is a Linux Foundation project focused on creating analytics and metrics to help define community health. Learn More

Get to know the CHAOSS community and learn how to participate.

CHAOSS Working groups

The goal of the working groups is to refine the metrics and to work with software implementations. The workgroups are built around the four categories of metrics that CHAOSS has identified.

The working groups are:
Common Metrics
Diversity and Inclusion
Evolution
Risk
Value



Bioconductor / IRanges

Compare from your repos:

Lines of code added by the top 10 authors as Percentages - By Time Period

cmt_author_email

- 🟦 Lori.Shepherd@RoswellPark.org
- 🟧 biocbuild@malbec1.roswellpark.org
- 🟩 dtenenba@taipan.fhcrc.org
- 🟥 hpages@fredhutch.org

License Coverage

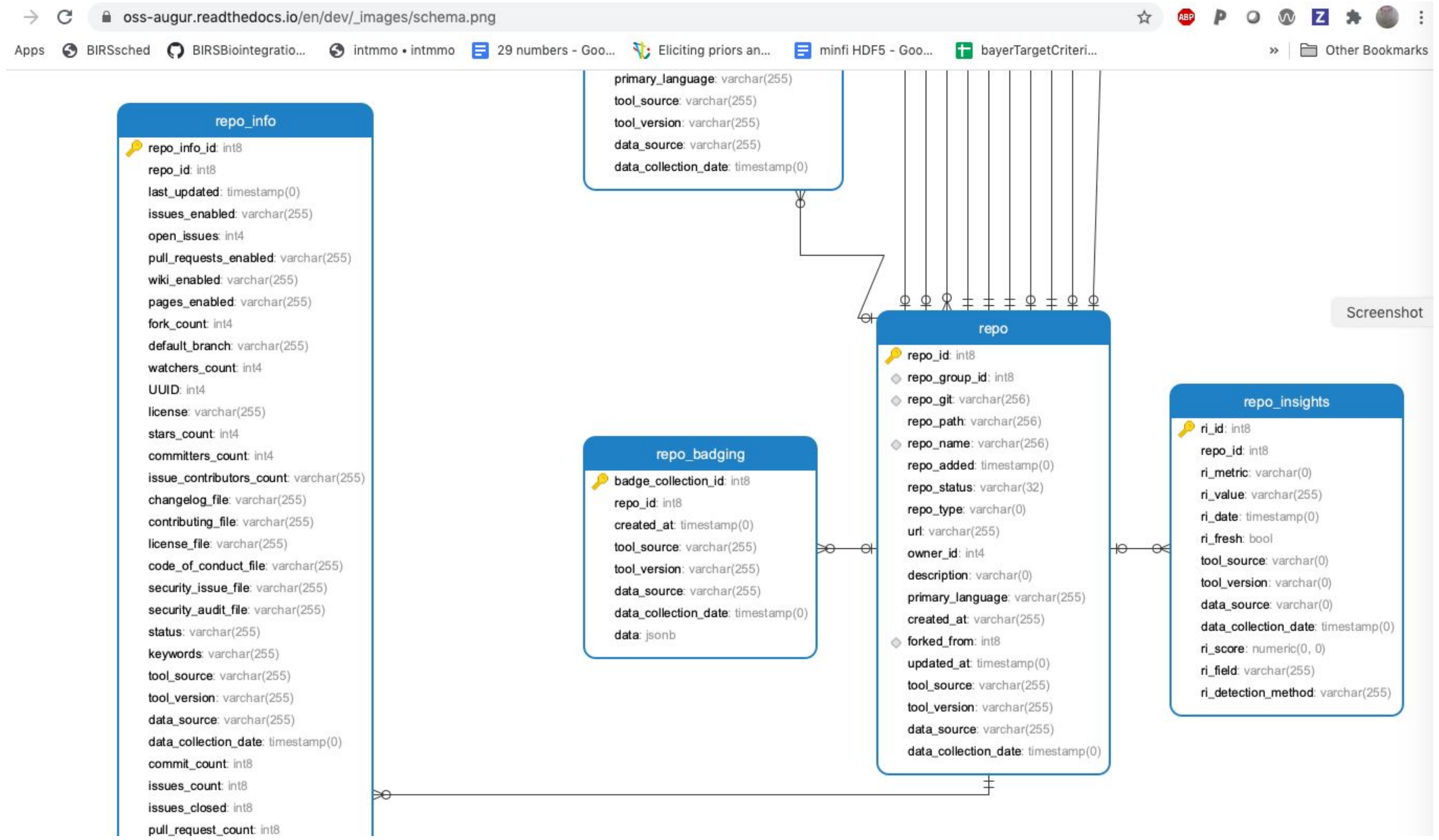
0.4%

Total Files	**247**
Files with Declared Licenses	**1**
Files without Licenses	**246**

Lines of code added by the top 10 authors

Author	2008	2009	2010	2011	2012	2013	201
biocbuild@malbec1.roswellpark.org	0	13164	2874	9645	14746	7745	111
jrodriguez@um.es	0	19114	5604	0	0	0	(
michafla@gene.com	0	3780	1732	831	1688	2525	12

CHAOSS obtains metadata about project management such as licensing, issue longevity, CoC, ...



primary_language: varchar(255)
tool_source: varchar(255)
tool_version: varchar(255)
data_source: varchar(255)
data_collection_date: timestamp(0)

repo_info

repo_info_id: int8
repo_id: int8
last_updated: timestamp(0)
issues_enabled: varchar(255)
open_issues: int4
pull_requests_enabled: varchar(255)
wiki_enabled: varchar(255)
pages_enabled: varchar(255)
fork_count: int4
default_branch: varchar(255)
watchers_count: int4
UUID: int4
license: varchar(255)
stars_count: int4
committers_count: int4
issue_contributors_count: varchar(255)
changelog_file: varchar(255)
contributing_file: varchar(255)
license_file: varchar(255)
code_of_conduct_file: varchar(255)
security_issue_file: varchar(255)
security_audit_file: varchar(255)
status: varchar(255)
keywords: varchar(255)
tool_source: varchar(255)
tool_version: varchar(255)
data_source: varchar(255)
data_collection_date: timestamp(0)
commit_count: int8
issues_count: int8
issues_closed: int8
pull_request_count: int8

repo_badging

badge_collection_id: int8
repo_id: int8
created_at: timestamp(0)
tool_source: varchar(255)
tool_version: varchar(255)
data_source: varchar(255)
data_collection_date: timestamp(0)
data: jsonb

repo

repo_id: int8
repo_group_id: int8
repo_git: varchar(256)
repo_path: varchar(256)
repo_name: varchar(256)
repo_added: timestamp(0)
repo_status: varchar(32)
repo_type: varchar(0)
url: varchar(255)
owner_id: int4
description: varchar(0)
primary_language: varchar(255)
created_at: varchar(255)
forked_from: int8
updated_at: timestamp(0)
tool_source: varchar(255)
tool_version: varchar(255)
data_source: varchar(255)
data_collection_date: timestamp(0)

repo_insights

ri_id: int8
repo_id: int8
ri_metric: varchar(0)
ri_value: varchar(255)
ri_date: timestamp(0)
ri_fresh: bool
tool_source: varchar(0)
tool_version: varchar(0)
data_source: varchar(0)
data_collection_date: timestamp(0)
ri_score: numeric(0, 0)
ri_field: varchar(255)
ri_detection_method: varchar(255)

What does Bioconductor need to be a superb platform for multimodal single-cell genomics?

- Support development of efficient data containers/data services
- Support software developers in achievement of scalability in their tools
- Support end-users in adoption and successful use

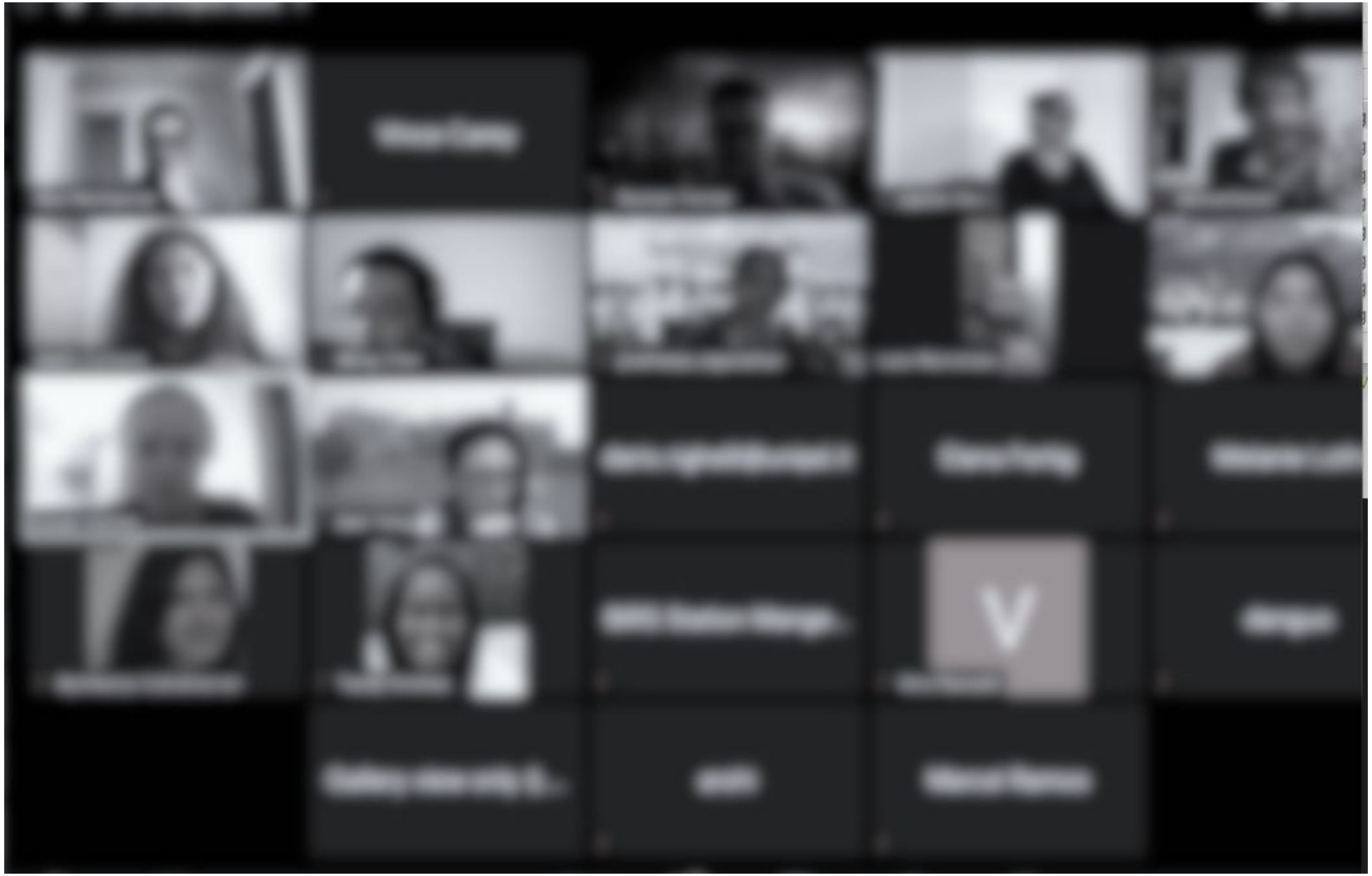
Some themes

- A system is scalable if its throughput can be increased through cost-effective additions to its "capacity"
    - Adding cores or RAM **is preferred to** reformatting data **is preferred to** refactoring/ rewriting code
- Designing new methods for scalability is intrinsically hard and unstable IT and rapid change in biotechnology makes it harder
- All the components of a software ecosystem need to be jointly tested in realistic settings
    - We are all engaged in continuous integration/continuous distribution(CI/CD), **whether we like it or not**, and components and underlying platforms at developer, tester/builder, or user end can **change at any time**

Personal thoughts

- "Divide and conquer" is a reasonable strategy for achieving scalability for various common tasks in genomic data science
    - Decomposable (random-access) data representations; programming for data access, sparsity, fault tolerance, ...
    - "Divide and conquer" seems hard to implement in social coding and organic software ecosystem development
- "Mileage" is a useful concept: use "high-mileage" tools that have been found reliable and robust in practice
- For very new biotechnologies, with emphasis on speed of deployment, these principles can be hard to adopt
- Containerization and automated CI/CD are helpful but have their own costs

Socializing divide and conquer is challenging: Brainstorm comment: Everyone thinks their problem/method is unique, so tools proliferate, even though they may implement very similar algorithms ...



"Bioconductor" centrality: bio-biotech-data science-...

- ~20 years of R-based solutions to problems arising in genome-scale assays
  - Preprocessing: parsing idiosyncratic formats, deriving quantifications, bias assessment, single-sample and multisample transformations for comparability
  - Object designs: increasing reliability through tight binding of metadata to assay data: **`X[G,S]`** endomorphisms
  - Annotation: platforms, genomes, pathways, ontologies, easily joined to assays
  - Visualization and analysis: capitalizing on R/CRAN, and extending the package and repository concepts

Package

A package is a set of functions and documents that includes a collection of tests of function correctness and adequacy, and passes these tests

The tests should exercise all the functions in realistic ways. Tests should include random inputs and requests -- repeating the same tests over and over again should be avoided, although this has some value when the underlying platform is changing over time

A package may have the capacity to be usable on different technological platforms. When this is achieved, the package is called "portable".

A package may require the existence of other packages

Virtues and costs of software packages

- Virtues
  - Modularity: address a limited class of tasks
  - Protocols for documentation and testing; issue tracking
  - Management, maintenance, portability systematized
- Costs
  - Dilemmas:
    - dependency on other packages *vs.* self-sufficiency
    - between maintaining stability for users *vs.* rapid introduction of new features and improvements
  - Management, maintenance, portability become obligations requiring effort in addition to research progress

Ecosystem

A software ecosystem is a collection of packages that includes a collection of package interoperability tests, and passes the tests

An ecosystem can be deployed on one or more technological platforms (e.g., operating systems or cloud computing systems) and co-evolves with different platforms that are changing at different rates

Giotto's implicit cross-language support (user consents)

```
> VC_small <- createGiottoObject(raw_exprs = expr_path, spatial_locs = loc_path)
Consider to install these (optional) packages to run all possible Giotto
commands:  MAST trendsceek multinet RTriangle FactoMiner
 Giotto does not automatically install all these packages as they are not
absolutely required and this reduces the number of dependenciesInstall a
miniconda Python environment for Giotto? (Yes/no/cancel) yes

 |---- install giotto environment ----|
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /Users/stvjc/Library/r-miniconda/envs/giotto_env

  added / updated specs:
    - python
```

Two strategic approaches to consider for R+python

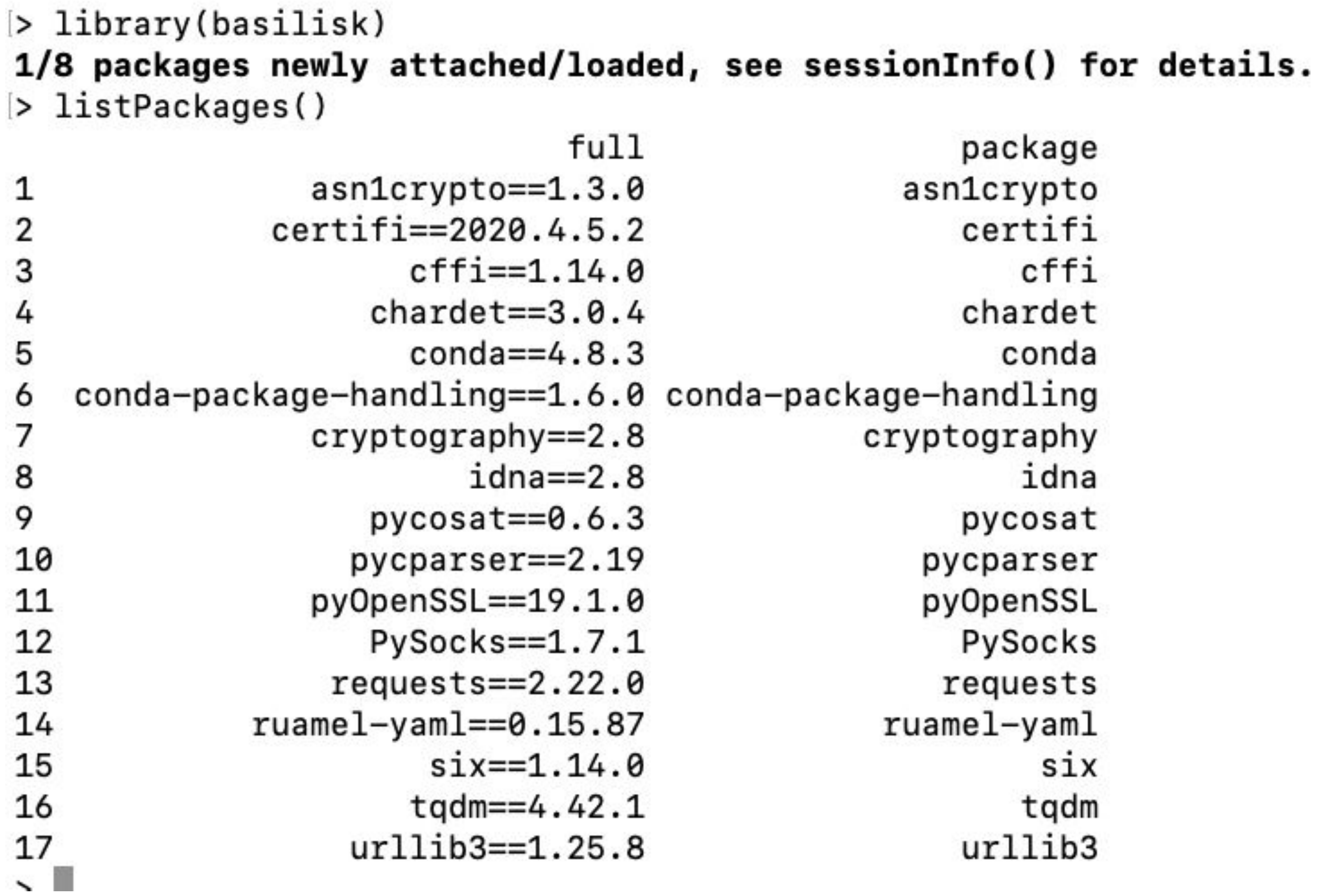
1) basilisk (Aaron Lun): Infrastructure support for Bioc-python interoperation -- a base conda environment is stored in user cache area, keyed to current Bioc/basilisk version; *client packages* specify **exact name and version** of python modules desired, which are acquired and cached as needed

2) BiocSklearn and other basilisk-client packages to expose numerical and statistical components of interest

-> Be precise about versions

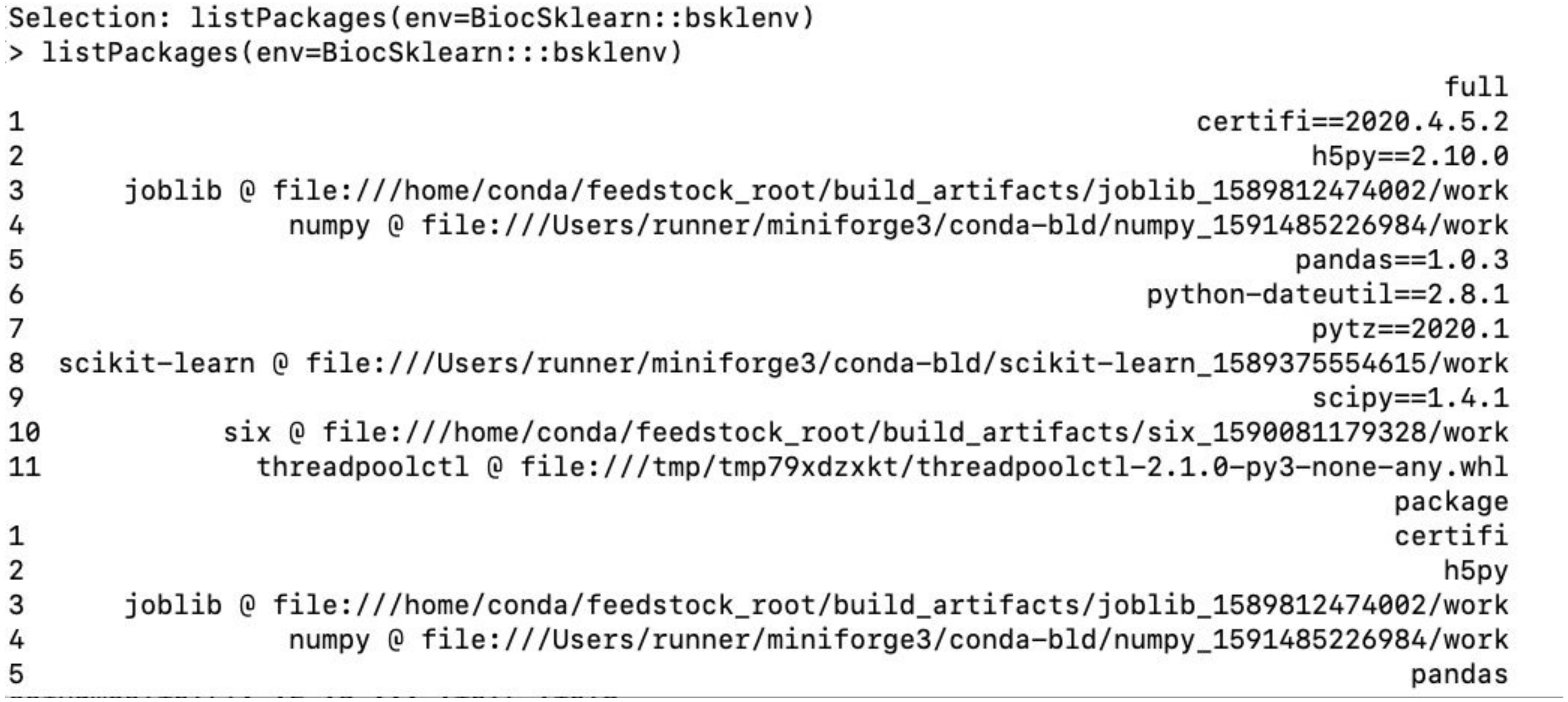
-> Don't go it alone!

Core packages installed with basilisk



```
> library(basilisk)
1/8 packages newly attached/loaded, see sessionInfo() for details.
> listPackages()
                              full                     package
1               asn1crypto==1.3.0                  asn1crypto
2              certifi==2020.4.5.2                     certifi
3                     cffi==1.14.0                        cffi
4                   chardet==3.0.4                     chardet
5                     conda==4.8.3                       conda
6  conda-package-handling==1.6.0 conda-package-handling
7               cryptography==2.8                cryptography
8                       idna==2.8                        idna
9                   pycosat==0.6.3                     pycosat
10                pycparser==2.19                   pycparser
11              pyOpenSSL==19.1.0                   pyOpenSSL
12                  PySocks==1.7.1                     PySocks
13                requests==2.22.0                    requests
14            ruamel-yaml==0.15.87                 ruamel-yaml
15                     six==1.14.0                         six
16                    tqdm==4.42.1                        tqdm
17                 urllib3==1.25.8                     urllib3
>
```

Upon first use of BiocSklearn::skPCA, 11 python packages are added to conda env for BiocSklearn

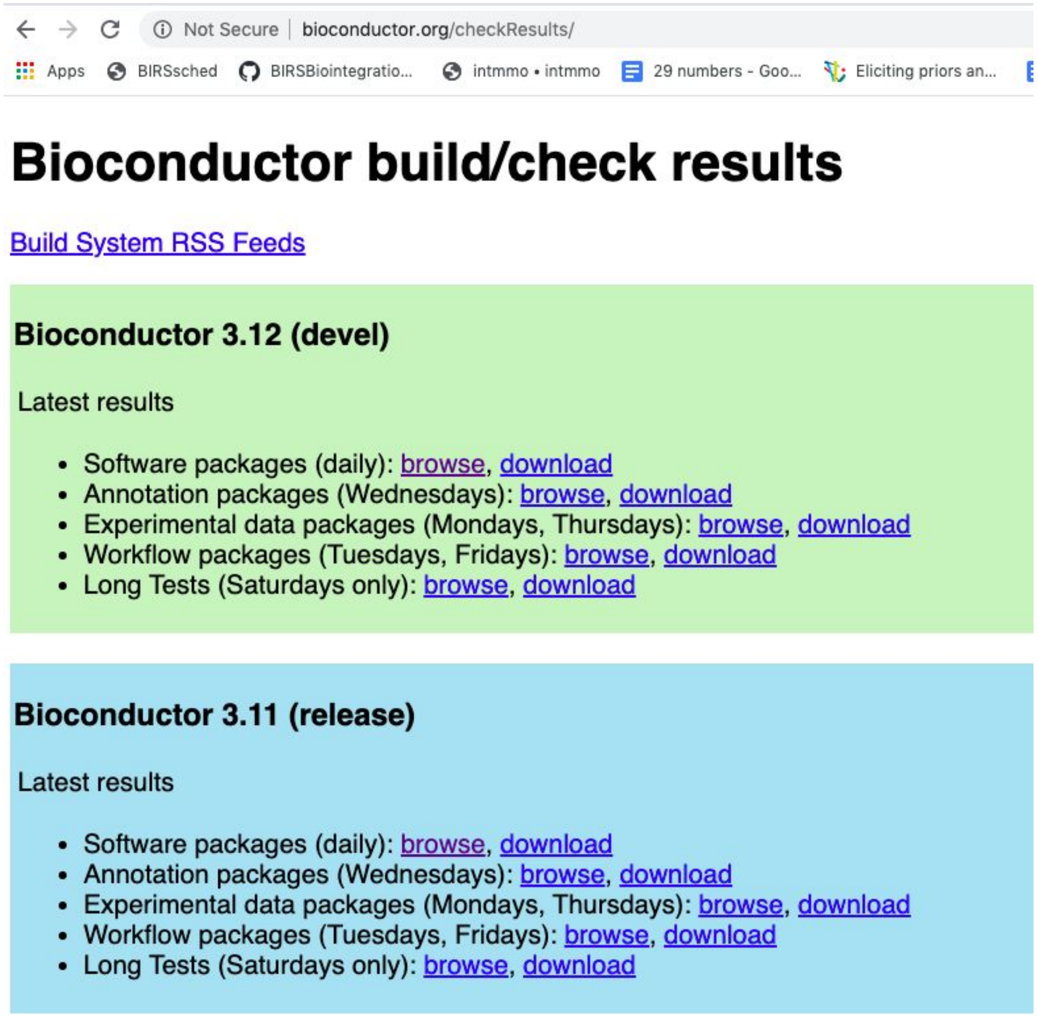


```
Selection: listPackages(env=BiocSklearn::bsklenv)
> listPackages(env=BiocSklearn:::bsklenv)
                                                                                   full
1                                                                    certifi==2020.4.5.2
2                                                                           h5py==2.10.0
3        joblib @ file:///home/conda/feedstock_root/build_artifacts/joblib_1589812474002/work
4                   numpy @ file:///Users/runner/miniforge3/conda-bld/numpy_1591485226984/work
5                                                                          pandas==1.0.3
6                                                                 python-dateutil==2.8.1
7                                                                           pytz==2020.1
8   scikit-learn @ file:///Users/runner/miniforge3/conda-bld/scikit-learn_1589375554615/work
9                                                                           scipy==1.4.1
10             six @ file:///home/conda/feedstock_root/build_artifacts/six_1590081179328/work
11               threadpoolctl @ file:///tmp/tmp79xdzxkt/threadpoolctl-2.1.0-py3-none-any.whl
                                                                                package
1                                                                               certifi
2                                                                                  h5py
3        joblib @ file:///home/conda/feedstock_root/build_artifacts/joblib_1589812474002/work
4                   numpy @ file:///Users/runner/miniforge3/conda-bld/numpy_1591485226984/work
5                                                                                pandas
```

Claim: the heart of the ecosystem is its build system/build governance

Bioconductor's principles:

- Three platforms (linux, windows, mac)
- Two streams: release, devel
- Track R's versioning (devel Oct-March, release April-Sept)
- Fast enough for biotech (point release every 6 months)
- Limited back-compatibility (to drop a component, deprecate for one release, then defunct)
- Software, annotation, and experiment packages are all first-class citizens
- Package dependencies limited to CRAN and Bioconductor



Bioconductor build/check results

Build System RSS Feeds

Bioconductor 3.12 (devel)

Latest results

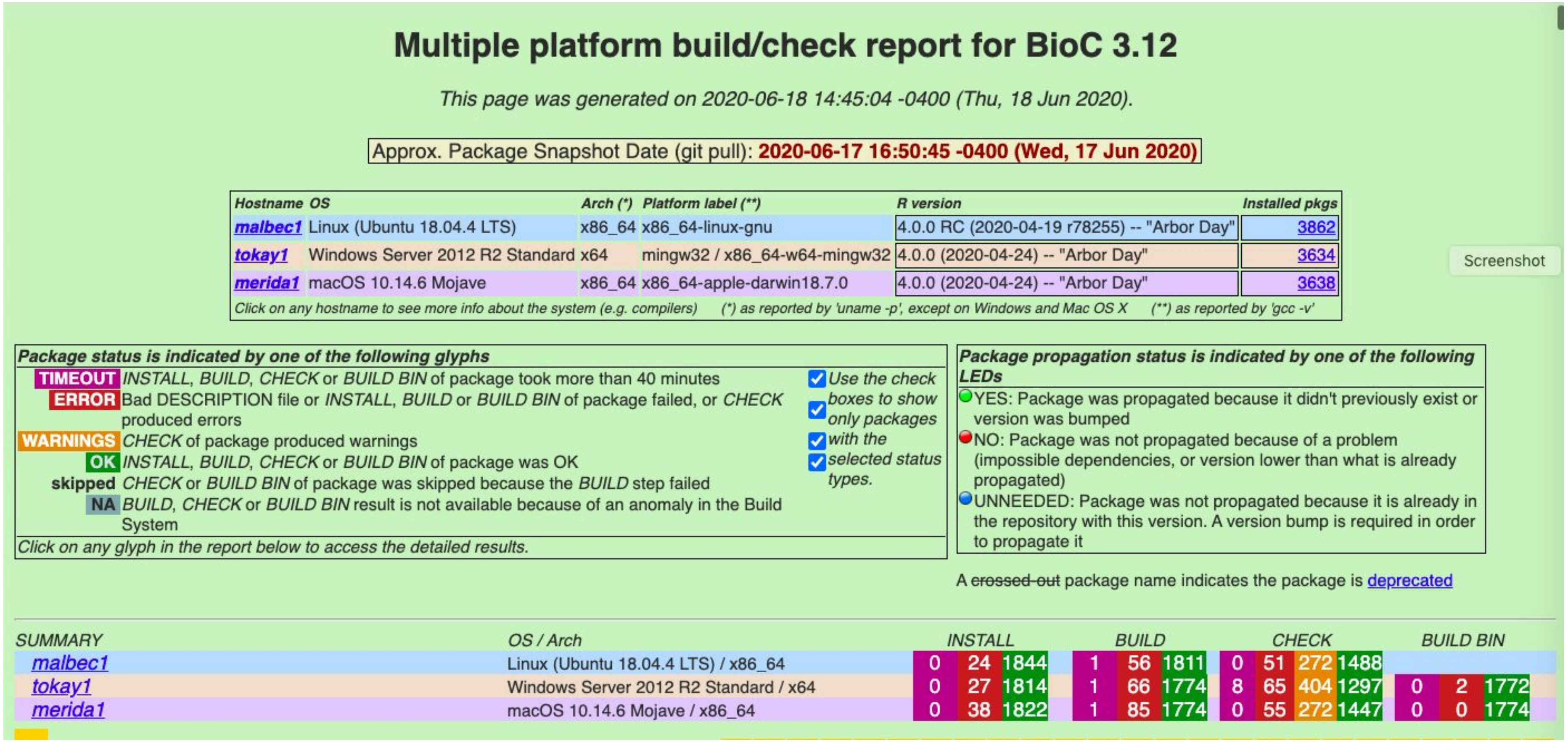
- Software packages (daily): browse, download
- Annotation packages (Wednesdays): browse, download
- Experimental data packages (Mondays, Thursdays): browse, download
- Workflow packages (Tuesdays, Fridays): browse, download
- Long Tests (Saturdays only): browse, download

Bioconductor 3.11 (release)

Latest results

- Software packages (daily): browse, download
- Annotation packages (Wednesdays): browse, download
- Experimental data packages (Mondays, Thursdays): browse, download
- Workflow packages (Tuesdays, Fridays): browse, download
- Long Tests (Saturdays only): browse, download

To support 1844 Bioc software packages, need 3862 total packages (including CRAN)



Multiple platform build/check report for BioC 3.12

This page was generated on 2020-06-18 14:45:04 -0400 (Thu, 18 Jun 2020).

Approx. Package Snapshot Date (git pull): **2020-06-17 16:50:45 -0400 (Wed, 17 Jun 2020)**

Hostname	***OS***	***Arch (*)***	***Platform label (**)***	***R version***	***Installed pkgs***
malbec1	Linux (Ubuntu 18.04.4 LTS)	x86_64	x86_64-linux-gnu	4.0.0 RC (2020-04-19 r78255) -- "Arbor Day"	3862
tokay1	Windows Server 2012 R2 Standard	x64	mingw32 / x86_64-w64-mingw32	4.0.0 (2020-04-24) -- "Arbor Day"	3634
merida1	macOS 10.14.6 Mojave	x86_64	x86_64-apple-darwin18.7.0	4.0.0 (2020-04-24) -- "Arbor Day"	3638

Click on any hostname to see more info about the system (e.g. compilers) (*) as reported by 'uname -p', except on Windows and Mac OS X (**) as reported by 'gcc -v'

Screenshot

Package status is indicated by one of the following glyphs

TIMEOUT	*INSTALL*, *BUILD*, *CHECK* or *BUILD BIN* of package took more than 40 minutes
ERROR	Bad DESCRIPTION file or *INSTALL*, *BUILD* or *BUILD BIN* of package failed, or *CHECK* produced errors
WARNINGS	*CHECK* of package produced warnings
OK	*INSTALL*, *BUILD*, *CHECK* or *BUILD BIN* of package was OK
skipped	*CHECK* or *BUILD BIN* of package was skipped because the *BUILD* step failed
NA	*BUILD*, *CHECK* or *BUILD BIN* result is not available because of an anomaly in the Build System

☑ ☑ ☑ ☑ *Use the check boxes to show only packages with the selected status types.*

Click on any glyph in the report below to access the detailed results.

Package propagation status is indicated by one of the following LEDs

- 🟢 YES: Package was propagated because it didn't previously exist or version was bumped
- 🔴 NO: Package was not propagated because of a problem (impossible dependencies, or version lower than what is already propagated)
- 🔵 UNNEEDED: Package was not propagated because it is already in the repository with this version. A version bump is required in order to propagate it

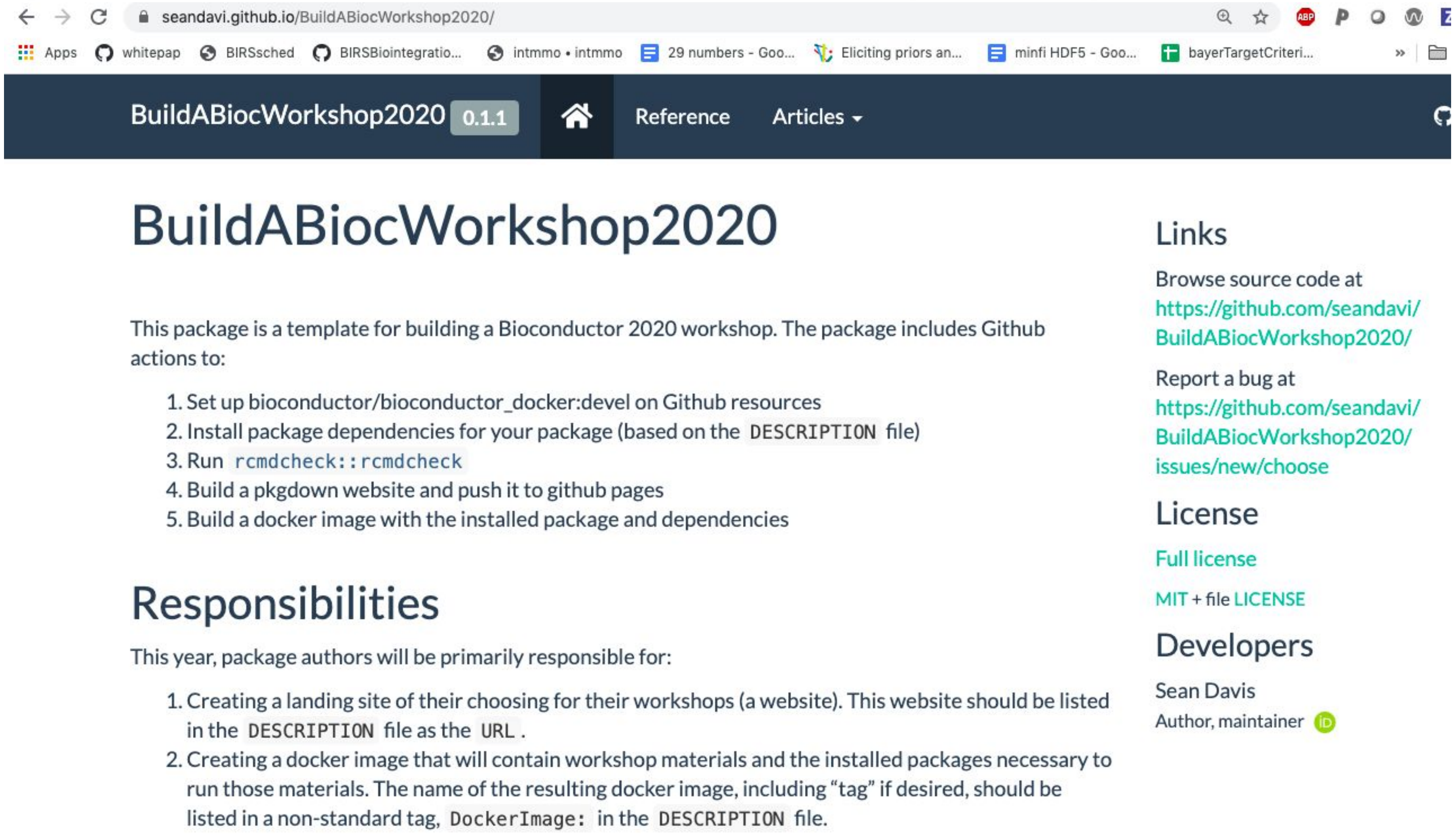
A ~~crossed-out~~ package name indicates the package is deprecated

SUMMARY	*OS / Arch*	*INSTALL*			*BUILD*			*CHECK*				*BUILD BIN*		
malbec1	Linux (Ubuntu 18.04.4 LTS) / x86_64	0	24	1844	1	56	1811	0	51	272	1488			
tokay1	Windows Server 2012 R2 Standard / x64	0	27	1814	1	66	1774	8	65	404	1297	0	2	1772
merida1	macOS 10.14.6 Mojave / x86_64	0	38	1822	1	85	1774	0	55	272	1447	0	0	1774

Opportunities for build system/ecosystem/users

- Containerization of builder infrastructure
- Binary package repositories for container users
- Continuous integration/delivery via github actions
- Data services
- Analysis services: GA4GH Tool Registry Service, and Workflow Execution Service are standards to watch for large genomic workflows

Example from Sean Davis: workshop components



BuildABiocWorkshop2020

This package is a template for building a Bioconductor 2020 workshop. The package includes Github actions to:

1. Set up bioconductor/bioconductor_docker:devel on Github resources
2. Install package dependencies for your package (based on the `DESCRIPTION` file)
3. Run `rcmdcheck::rcmdcheck`
4. Build a pkgdown website and push it to github pages
5. Build a docker image with the installed package and dependencies

Responsibilities

This year, package authors will be primarily responsible for:

1. Creating a landing site of their choosing for their workshops (a website). This website should be listed in the `DESCRIPTION` file as the `URL`.
2. Creating a docker image that will contain workshop materials and the installed packages necessary to run those materials. The name of the resulting docker image, including "tag" if desired, should be listed in a non-standard tag, `DockerImage:` in the `DESCRIPTION` file.

Links

Browse source code at https://github.com/seandavi/BuildABiocWorkshop2020/

Report a bug at https://github.com/seandavi/BuildABiocWorkshop2020/issues/new/choose

License

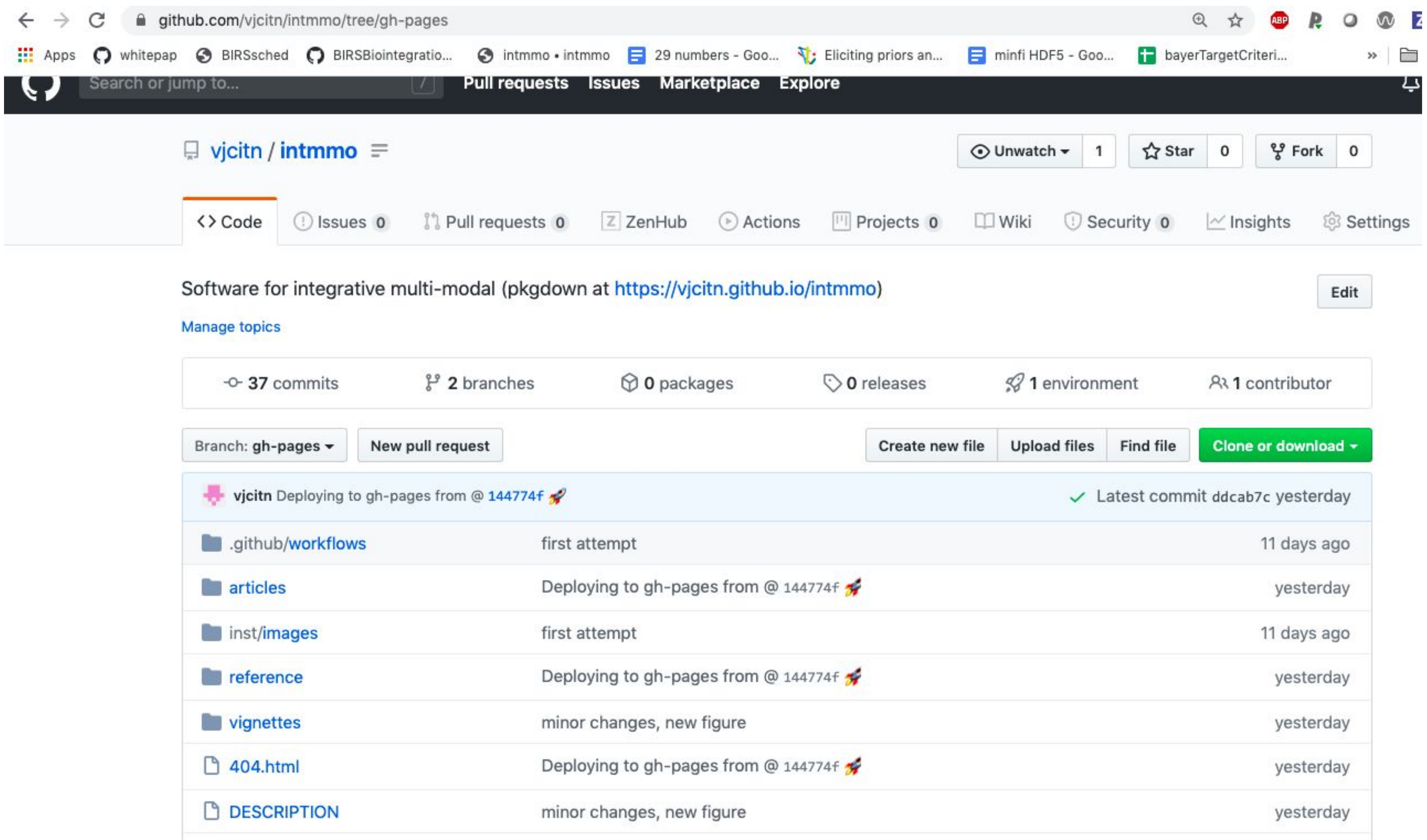
Full license

MIT + file LICENSE

Developers

Sean Davis
Author, maintainer

"workshop" concept deployed in support of this talk



github.com/vjcitn/intmmo/tree/gh-pages

Apps | whitepap | BIRSsched | BIRSBiointegratio... | intmmo • intmmo | 29 numbers - Goo... | Eliciting priors an... | minfi HDF5 - Goo... | bayerTargetCriteri...

Search or jump to... | Pull requests | Issues | Marketplace | Explore

vjcitn / intmmo

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | ZenHub | Actions | Projects 0 | Wiki | Security 0 | Insights | Settings

Software for integrative multi-modal (pkgdown at https://vjcitn.github.io/intmmo)

Edit

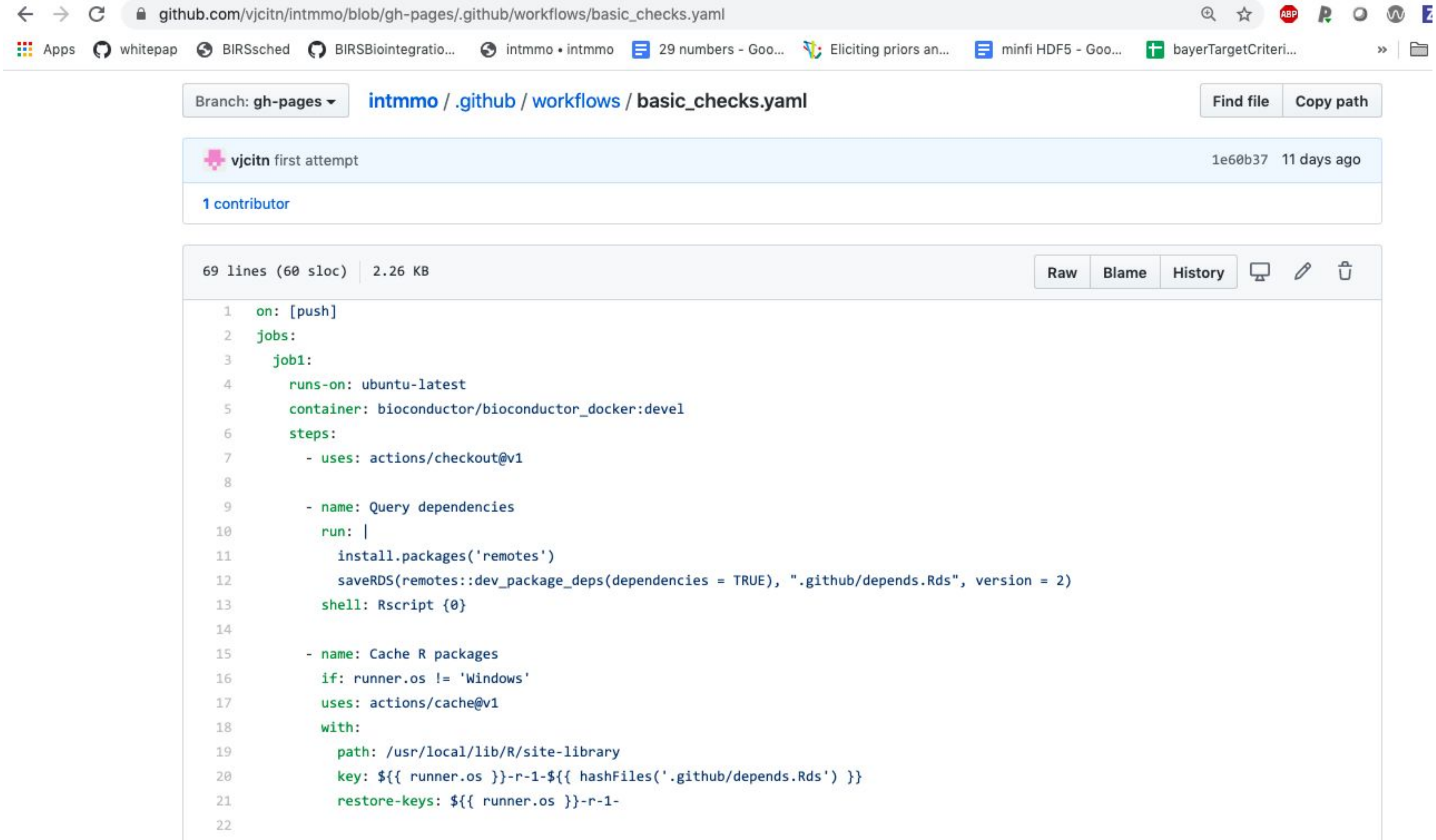
Manage topics

37 commits | 2 branches | 0 packages | 0 releases | 1 environment | 1 contributor

Branch: gh-pages | New pull request | Create new file | Upload files | Find file | Clone or download

vjcitn Deploying to gh-pages from @ 144774f 🚀 ✓ Latest commit ddcab7c yesterday

.github/workflows	first attempt	11 days ago
articles	Deploying to gh-pages from @ 144774f 🚀	yesterday
inst/images	first attempt	11 days ago
reference	Deploying to gh-pages from @ 144774f 🚀	yesterday
vignettes	minor changes, new figure	yesterday
404.html	Deploying to gh-pages from @ 144774f 🚀	yesterday
DESCRIPTION	minor changes, new figure	yesterday



intmmo / .github / workflows / basic_checks.yaml

vjcitn first attempt 1e60b37 11 days ago

1 contributor

69 lines (60 sloc) | 2.26 KB

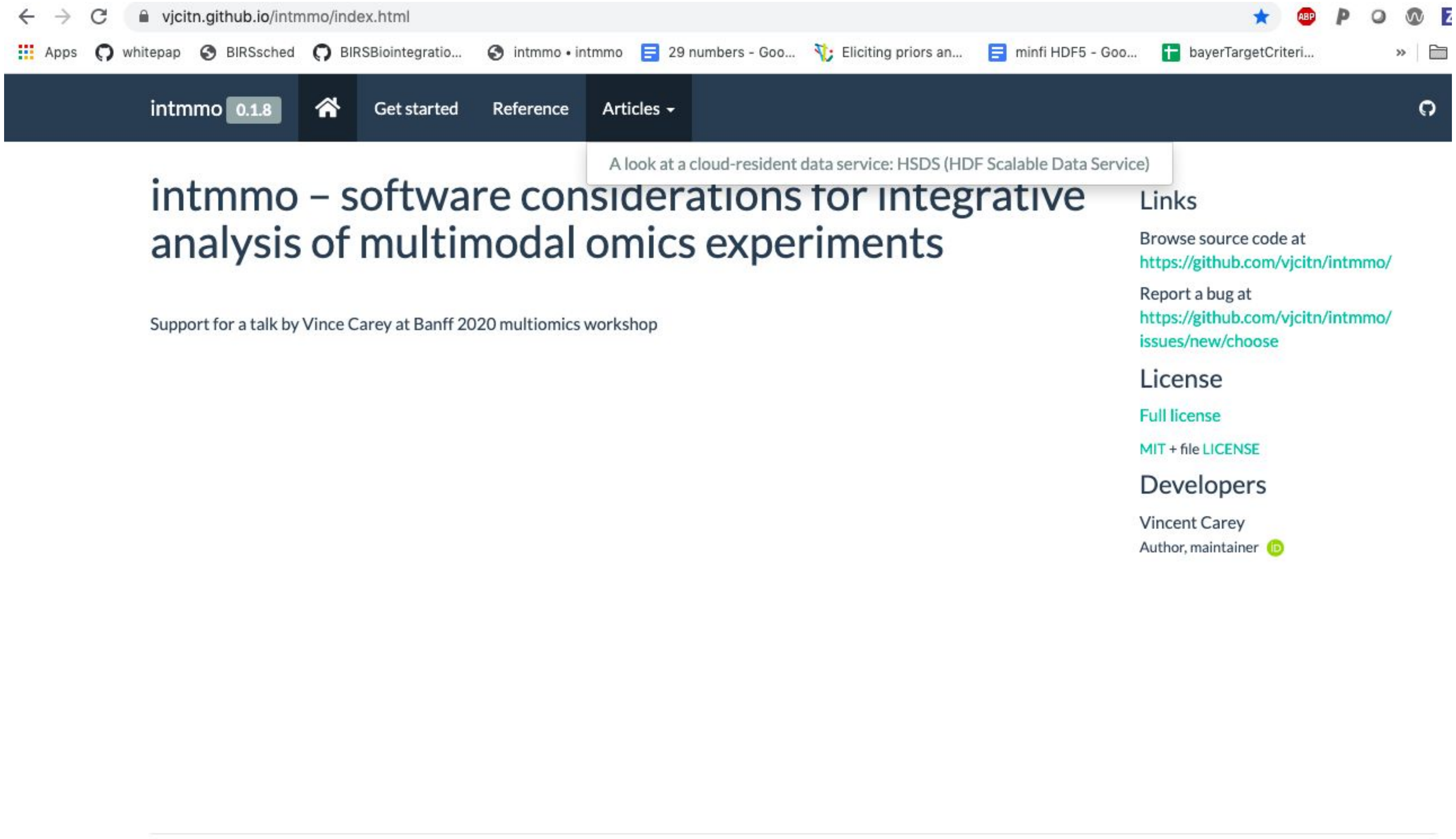
```
on: [push]
jobs:
  job1:
    runs-on: ubuntu-latest
    container: bioconductor/bioconductor_docker:devel
    steps:
      - uses: actions/checkout@v1

      - name: Query dependencies
        run: |
          install.packages('remotes')
          saveRDS(remotes::dev_package_deps(dependencies = TRUE), ".github/depends.Rds", version = 2)
        shell: Rscript {0}

      - name: Cache R packages
        if: runner.os != 'Windows'
        uses: actions/cache@v1
        with:
          path: /usr/local/lib/R/site-library
          key: ${{ runner.os }}-r-1-${{ hashFiles('.github/depends.Rds') }}
          restore-keys: ${{ runner.os }}-r-1-

```

github actions builds a formatted site



vjcitn.github.io/intmmo/index.html

A look at a cloud-resident data service: HSDS (HDF Scalable Data Service)

intmmo – software considerations for integrative analysis of multimodal omics experiments

Support for a talk by Vince Carey at Banff 2020 multiomics workshop

Links

Browse source code at https://github.com/vjcitn/intmmo/

Report a bug at https://github.com/vjcitn/intmmo/issues/new/choose

License

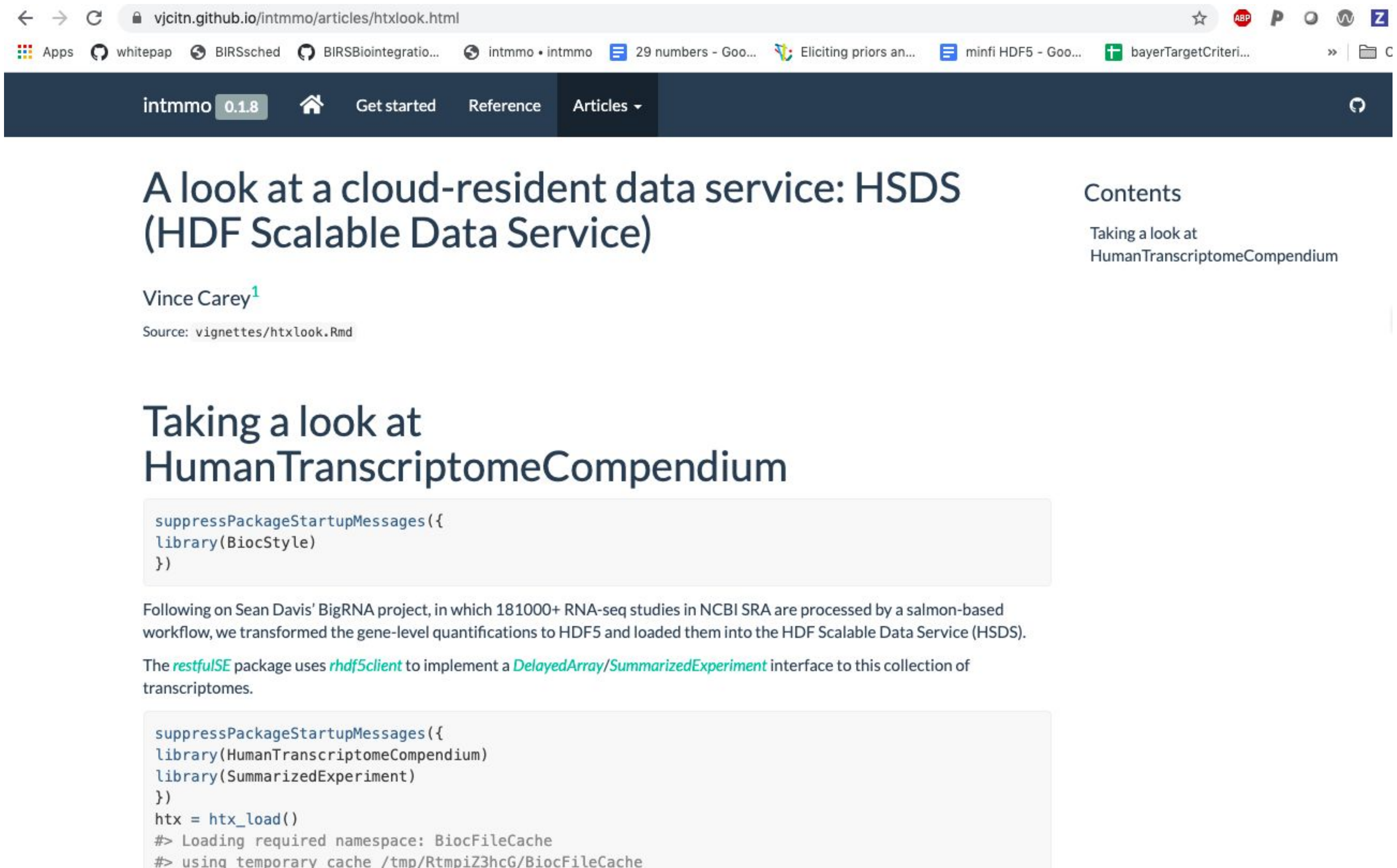
Full license

MIT + file LICENSE

Developers

Vincent Carey
Author, maintainer

The "article" illustrates use of HSDS/restfulSE



A look at a cloud-resident data service: HSDS (HDF Scalable Data Service)

Vince Carey[1]

Source: `vignettes/htxlook.Rmd`

Contents

Taking a look at HumanTranscriptomeCompendium

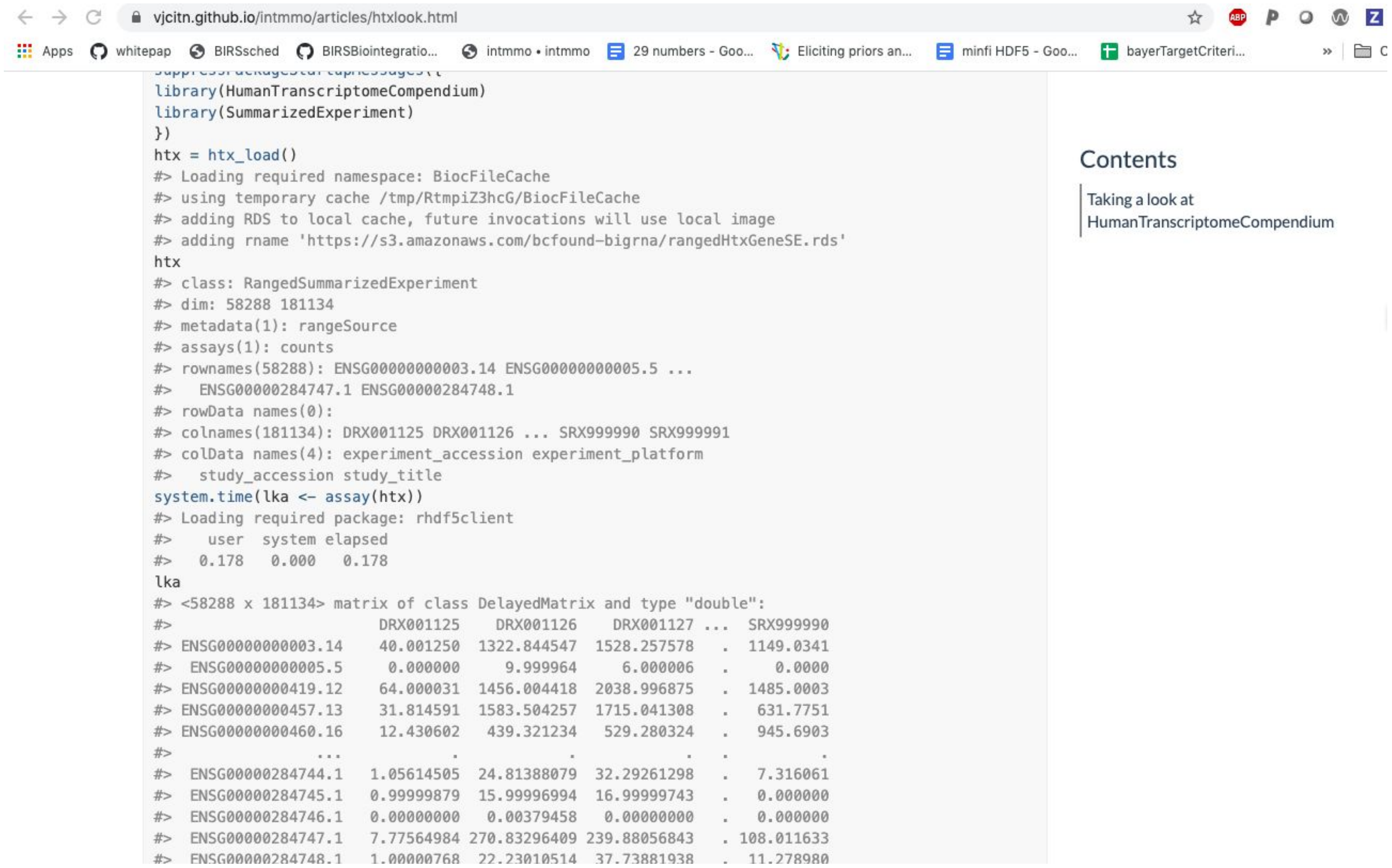
Taking a look at HumanTranscriptomeCompendium

```
suppressPackageStartupMessages({
library(BiocStyle)
})
```

Following on Sean Davis' BigRNA project, in which 181000+ RNA-seq studies in NCBI SRA are processed by a salmon-based workflow, we transformed the gene-level quantifications to HDF5 and loaded them into the HDF Scalable Data Service (HSDS).

The *restfulSE* package uses *rhdf5client* to implement a *DelayedArray*/*SummarizedExperiment* interface to this collection of transcriptomes.

```
suppressPackageStartupMessages({
library(HumanTranscriptomeCompendium)
library(SummarizedExperiment)
})
htx = htx_load()
#> Loading required namespace: BiocFileCache
#> using temporary cache /tmp/RtmpiZ3hcG/BiocFileCache
```



```
suppressPackageStartupMessages({
library(HumanTranscriptomeCompendium)
library(SummarizedExperiment)
})
htx = htx_load()
#> Loading required namespace: BiocFileCache
#> using temporary cache /tmp/RtmpiZ3hcG/BiocFileCache
#> adding RDS to local cache, future invocations will use local image
#> adding rname 'https://s3.amazonaws.com/bcfound-bigrna/rangedHtxGeneSE.rds'
htx
#> class: RangedSummarizedExperiment
#> dim: 58288 181134
#> metadata(1): rangeSource
#> assays(1): counts
#> rownames(58288): ENSG00000000003.14 ENSG00000000005.5 ...
#>   ENSG00000284747.1 ENSG00000284748.1
#> rowData names(0):
#> colnames(181134): DRX001125 DRX001126 ... SRX999990 SRX999991
#> colData names(4): experiment_accession experiment_platform
#>   study_accession study_title
system.time(lka <- assay(htx))
#> Loading required package: rhdf5client
#>    user  system elapsed
#>   0.178   0.000   0.178
lka
#> <58288 x 181134> matrix of class DelayedMatrix and type "double":
#>                      DRX001125    DRX001126    DRX001127 ...  SRX999990
#> ENSG00000000003.14    40.001250  1322.844547  1528.257578   .  1149.0341
#>  ENSG00000000005.5     0.000000     9.999964     6.000006   .     0.0000
#> ENSG00000000419.12    64.000031  1456.004418  2038.996875   .  1485.0003
#> ENSG00000000457.13    31.814591  1583.504257  1715.041308   .   631.7751
#> ENSG00000000460.16    12.430602   439.321234   529.280324   .   945.6903
#>                ...            .            .            .   .          .
#>  ENSG00000284744.1   1.05614505  24.81388079  32.29261298   .   7.316061
#>  ENSG00000284745.1   0.99999879  15.99996994  16.99999743   .   0.000000
#>  ENSG00000284746.1   0.00000000   0.00379458   0.00000000   .   0.000000
#>  ENSG00000284747.1   7.77564984 270.83296409 239.88056843   . 108.011633
#>  ENSG00000284748.1   1.00000768  22.23010514  37.73881938   .  11.278980
```

Contents

Taking a look at HumanTranscriptomeCompendium

Conclusions

- Rapid evolution in biotech and information technology lead to unstable software development situations
- Team coordination is crucial to avoid needless redundancy and gain mileage on all components
- Pain points for developers should be eased with continuous integration enhancements
    - Github actions should play a useful role
- GA4GH standards (TRS, WES) should be revisited frequently