How *Bioconductor* advances science and contributes to *R*

Martin Morgan
Roswell Park Comprehensive Cancer Center
Buffalo, New York, USA

slides

When I talk about *R*...

```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

Expressive, vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

Domain-specific understanding

Personality

Domain-specific understanding

```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

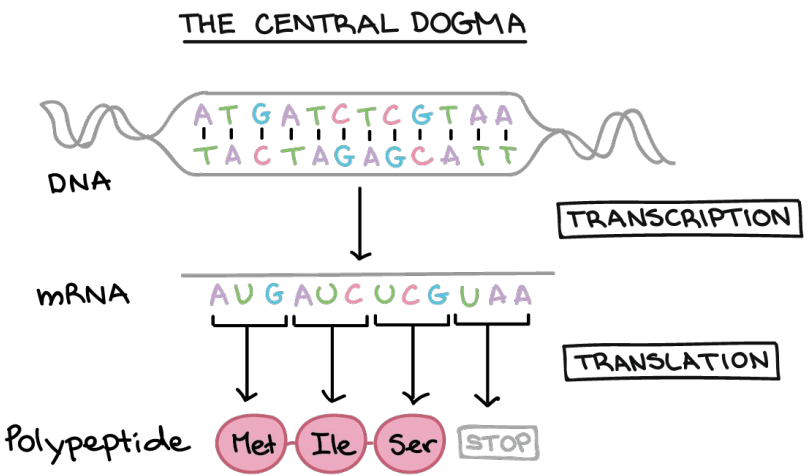
Expressive, vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

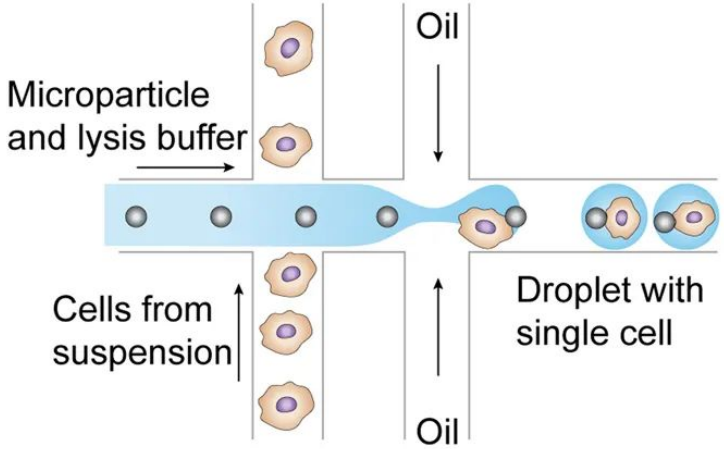
Domain-specific understanding

Personality

High-throughput genomics, e.g., single-cell RNA-seq



https://cdn.kastatic.org/ka-perseus-images/2b597889d05bc601803a3b4d9ec5ccd5e7b8d3af.png All Khan Academy content is available for free at www.khanacademy.org

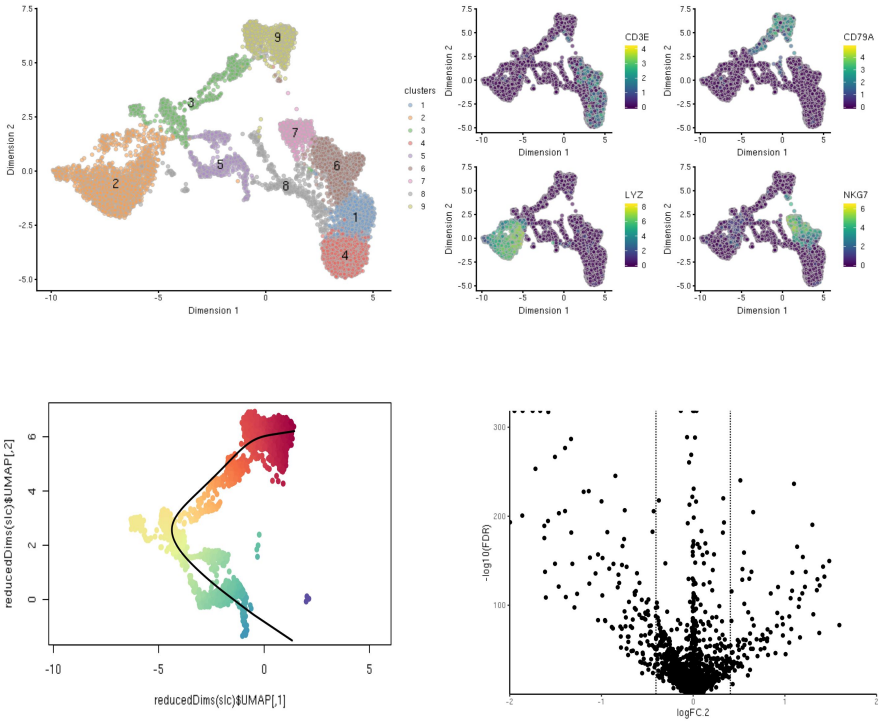


- Isolate individual cells
- Associate each cell with bar-coded beads
- Sequence bar-coded cDNA
- Hwang et al., 2018
  https://doi.org/10.1038/s12276-018-0071-8

Many statistical challenges & research questions

```
> counts      # 20k gene x 50k+ cell matrix
       GATAGGAG-1 GCGGCTTC-1 GGAATCGC-1 ...
ENSG001       679        448        873 ...
ENGS002         0          0          0 ...
ENGS003       467        515        621 ...
...
```

- Clustering & cell type classification*
- Gene expression*
- Cell trajectories*
- Data integration
- Differential expression*
- Annotation, gene set analysis, ...



Hicks, Amezquita, et al., https://osca.bioconductor.org

Data

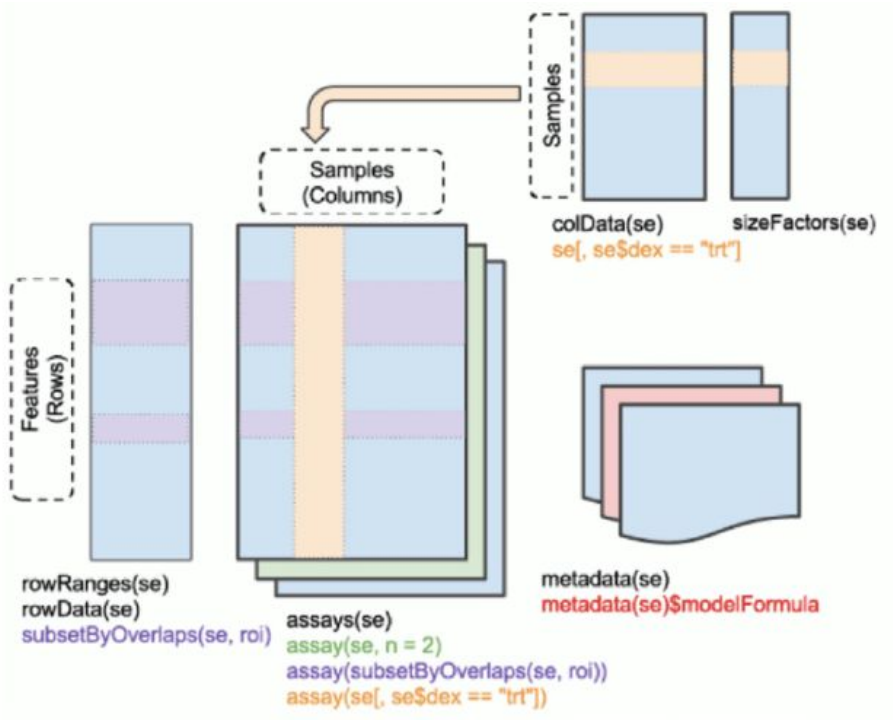
Very large primary data

- DNA sequence 'reads' (from mRNA)
- Reads aligned to a reference genome

Reduced for analysis

- E.g., matrix of read counts
- 'Genes' (e.g., 20k) x 'Samples' (e.g., 50k)
- Largest to date: ~28,000 genes x 1.3 million cells

Annotations on both genes and samples



Analysis

Inherently statistical

- Designed experiments
- Technology & artifacts -- batch effects, library size, sparsity, distribution, ...
- Usually: large P (genes) small N (samples)
    - E.g., 28k genes, 1.3 million cells, but only two (!) mice

Comparatively ‘big’

- Primary data is very large
- Even after reduction, e.g., single cell experiments, data can be 'big'
- Strategies for managing memory and throughput

Comprehension

Reproducibility

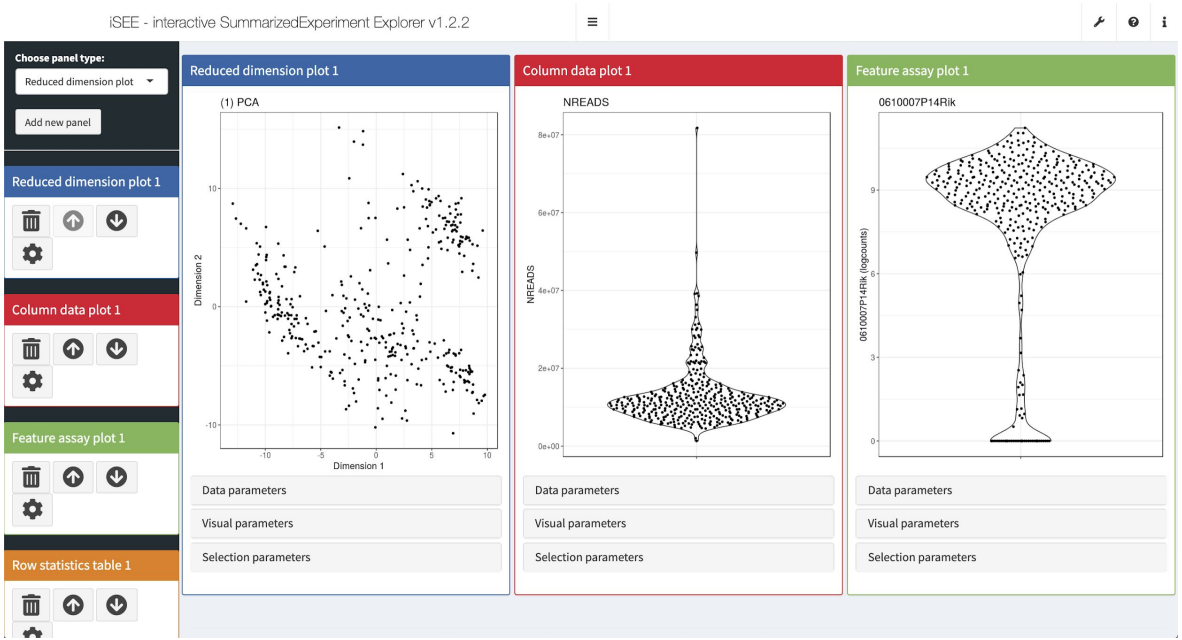
- During the analysis

Annotation

- Gene ids,pathways, drug targets, ...

Communication

- E.g., visual summaries, reports, slides, interactive apps, ...



iSEE package: https://bioconductor.org/packages/iSEE
blog: https://bit.ly/2UhNTP4

Bioconductor -- Statistical analysis and comprehension of high-throughput genomic data

Single-cell and bulk sequence data; expression, methylation, SNP and other microarrays; flow cytometry; proteomics; …

Funding from the US NIH, EU, Chan-Zuckerberg and other sources support a small professional team of developers

Real scientific progress from the broader developer & user community!

[Stickers!]

Personality

```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

Expressive, vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

Domain-specific understanding

Personality

About *Bioconductor* -- https://bioconductor.org

Established 2001 by R. Gentleman & colleagues

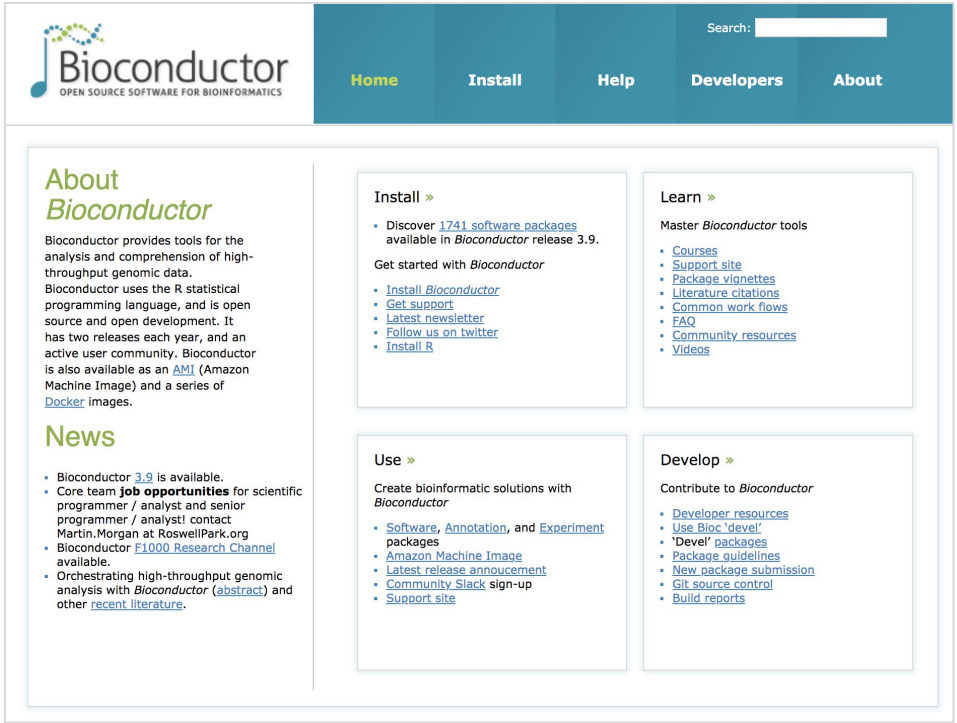
- Now 1750 *R* software packages

Users

- ½ million unique IP downloads last year
- 30,000 PubMedCentral citations

Contributors & developers

- More than 1200 maintainers worldwide

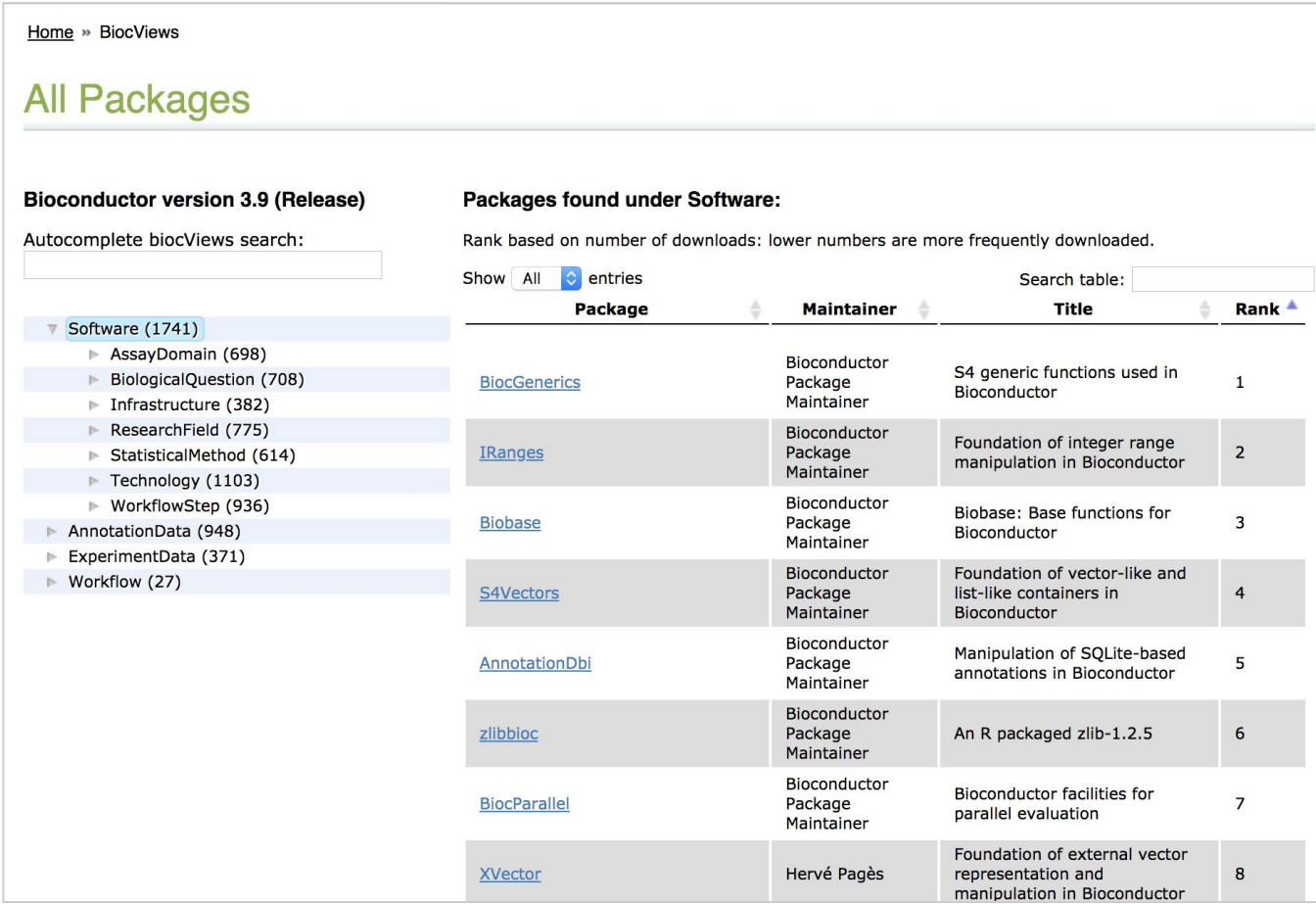


Packages

Analytic software packages

Additional types of packages

- Annotation: biological context
- Experiment data: reproducible examples & case studies
- Workflow: end-to-end documentation



Home » BiocViews

All Packages

Bioconductor version 3.9 (Release)

Autocomplete biocViews search:

- Software (1741)
  - AssayDomain (698)
  - BiologicalQuestion (708)
  - Infrastructure (382)
  - ResearchField (775)
  - StatisticalMethod (614)
  - Technology (1103)
  - WorkflowStep (936)
- AnnotationData (948)
- ExperimentData (371)
- Workflow (27)

Packages found under Software:

Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries Search table:

Package	Maintainer	Title	Rank
BiocGenerics	Bioconductor Package Maintainer	S4 generic functions used in Bioconductor	1
IRanges	Bioconductor Package Maintainer	Foundation of integer range manipulation in Bioconductor	2
Biobase	Bioconductor Package Maintainer	Biobase: Base functions for Bioconductor	3
S4Vectors	Bioconductor Package Maintainer	Foundation of vector-like and list-like containers in Bioconductor	4
AnnotationDbi	Bioconductor Package Maintainer	Manipulation of SQLite-based annotations in Bioconductor	5
zlibbioc	Bioconductor Package Maintainer	An R packaged zlib-1.2.5	6
BiocParallel	Bioconductor Package Maintainer	Bioconductor facilities for parallel evaluation	7
XVector	Hervé Pagès	Foundation of external vector representation and manipulation in Bioconductor	8

Package landing pages

Home » Bioconductor 3.9 » Software Packages » DESeq2

DESeq2

platforms all | rank 25 / 1741 | posts 360 / 1 / 2 / 50 | in Bioc 6 years
build ok | updated before release | dependencies 109

DOI: 10.18129/B9.bioc.DESeq2

Differential gene expression analysis based on the negative binomial distribution

- Badges, citations, DOI, …
- Installation instructions
- Vignettes

Bioconductor version: Release (3.9)

Estimate variance-mean dependence in count data from high-throughput sequencing assays and test for differential expression based on a model using the negative binomial distribution.

Author: Michael Love, Simon Anders, Wolfgang Huber

Maintainer: Michael Love <michaelisaiahlove at gmail.com>

Citation (from within R, enter `citation("DESeq2")`):

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

Installation

To install this package, start R (version "3.6") and enter:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("DESeq2")
```

For older versions of R, please refer to the appropriate Bioconductor release.

Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("DESeq2")
```

HTML	R Script	Analyzing RNA-seq data with DESeq2
PDF		Reference Manual
Text		NEWS

Support

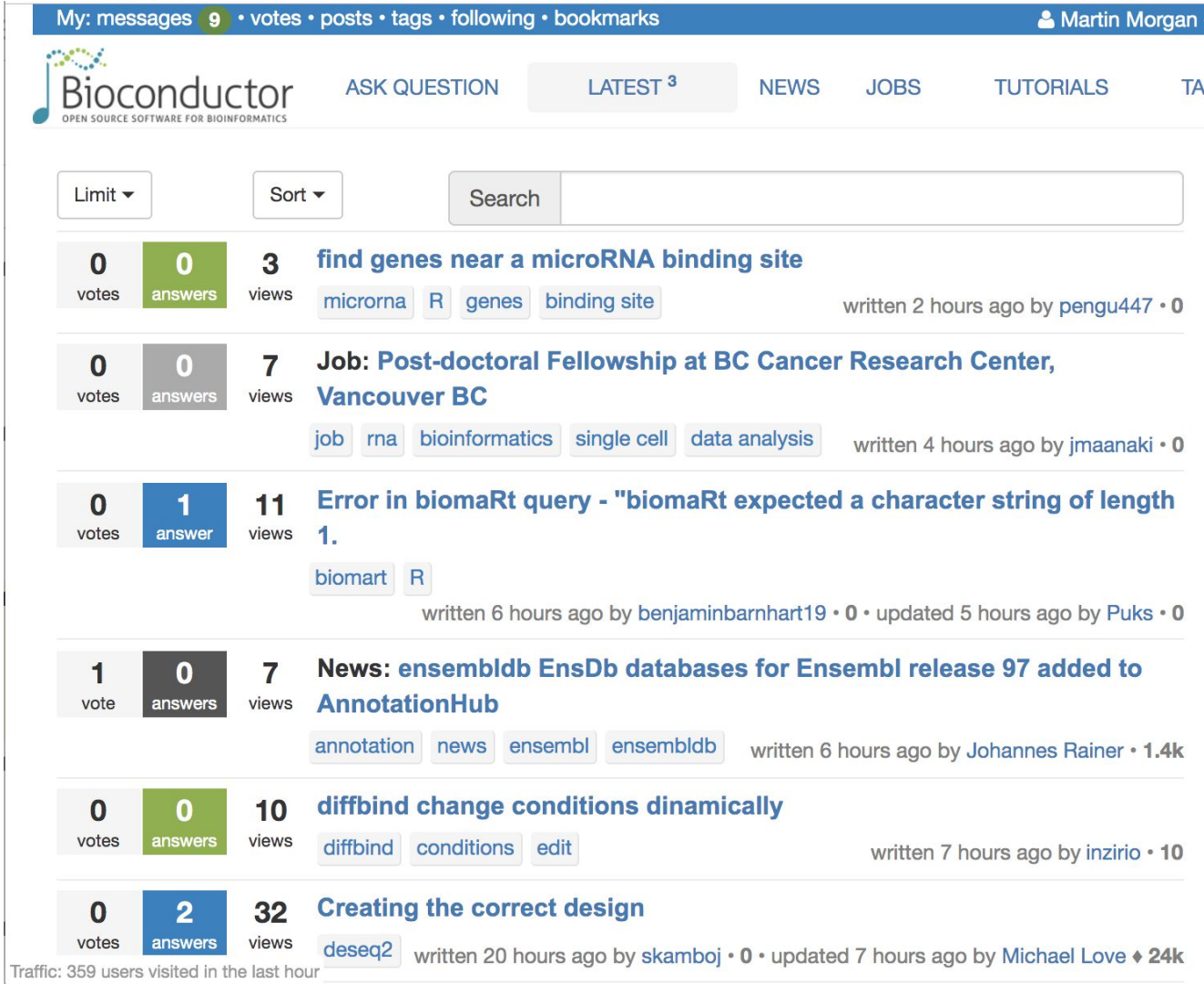
Users

- StackOverflow-style [support site]

Users / Developers

- Community [slack]

Maintainer [mailing list]



Support

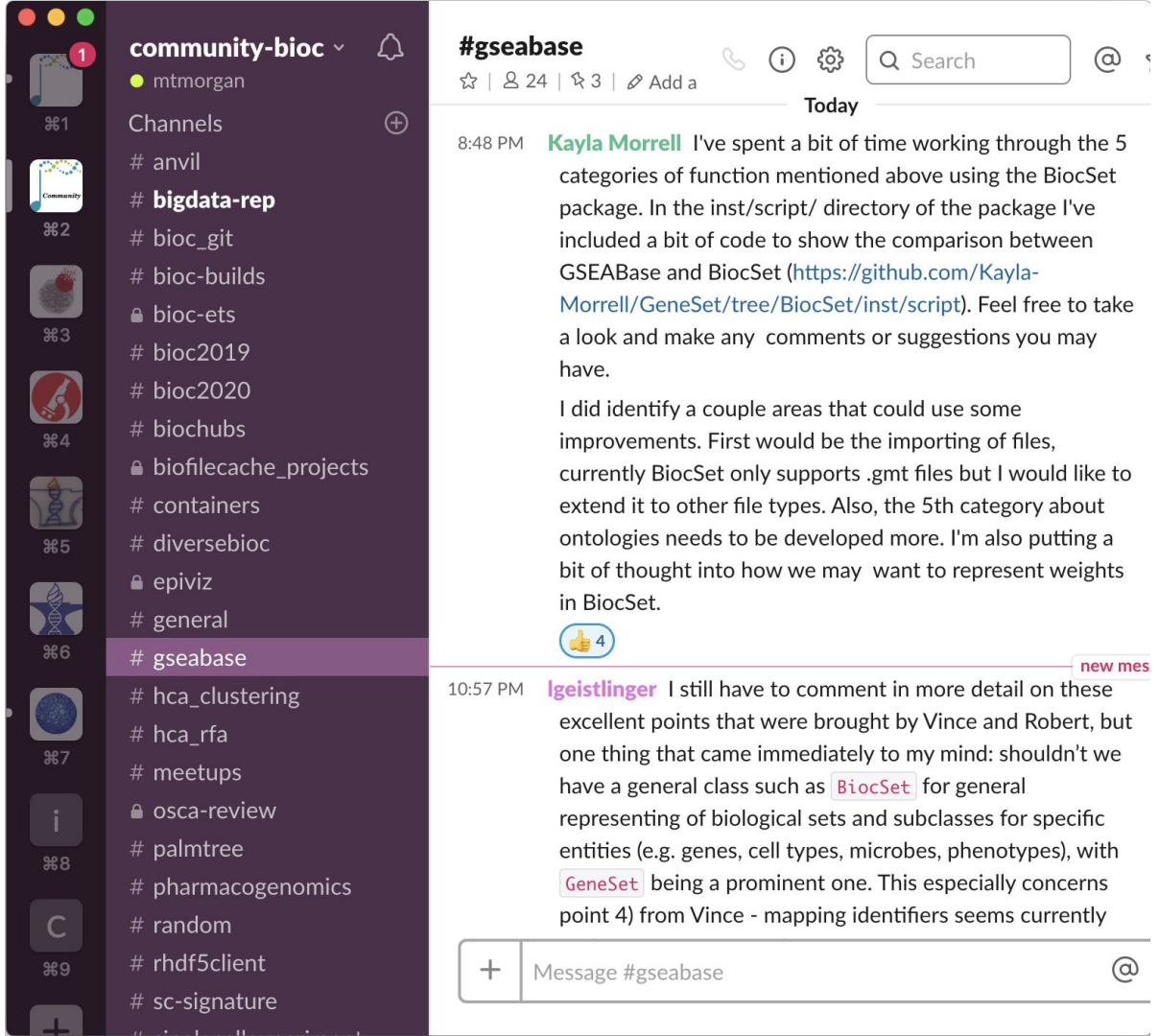
Users

- StackOverflow-style [support site]

Users / Developers

- Community [slack]

Maintainer [mailing list]



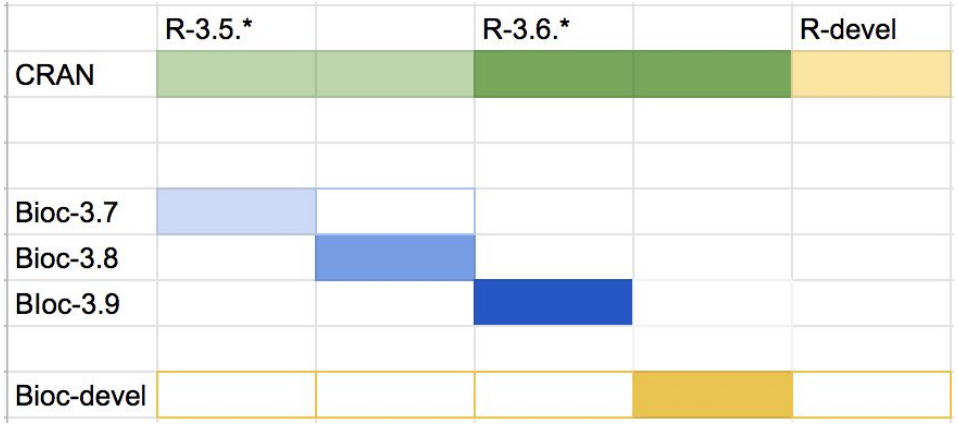
Releases & repositories

Releases

- A stable ‘release’ branch for users
- A ‘devel’ branch for new packages & features
- Twice-yearly: devel becomes release

BiocManager

- CRAN package for installing correct versions of *Bioconductor* packages
- Also installs CRAN, github packages




```
## install.packages("BiocManager") # CRAN
BiocManager::install(pkgs) # Bioc / CRAN / github
```

Packages

Version control

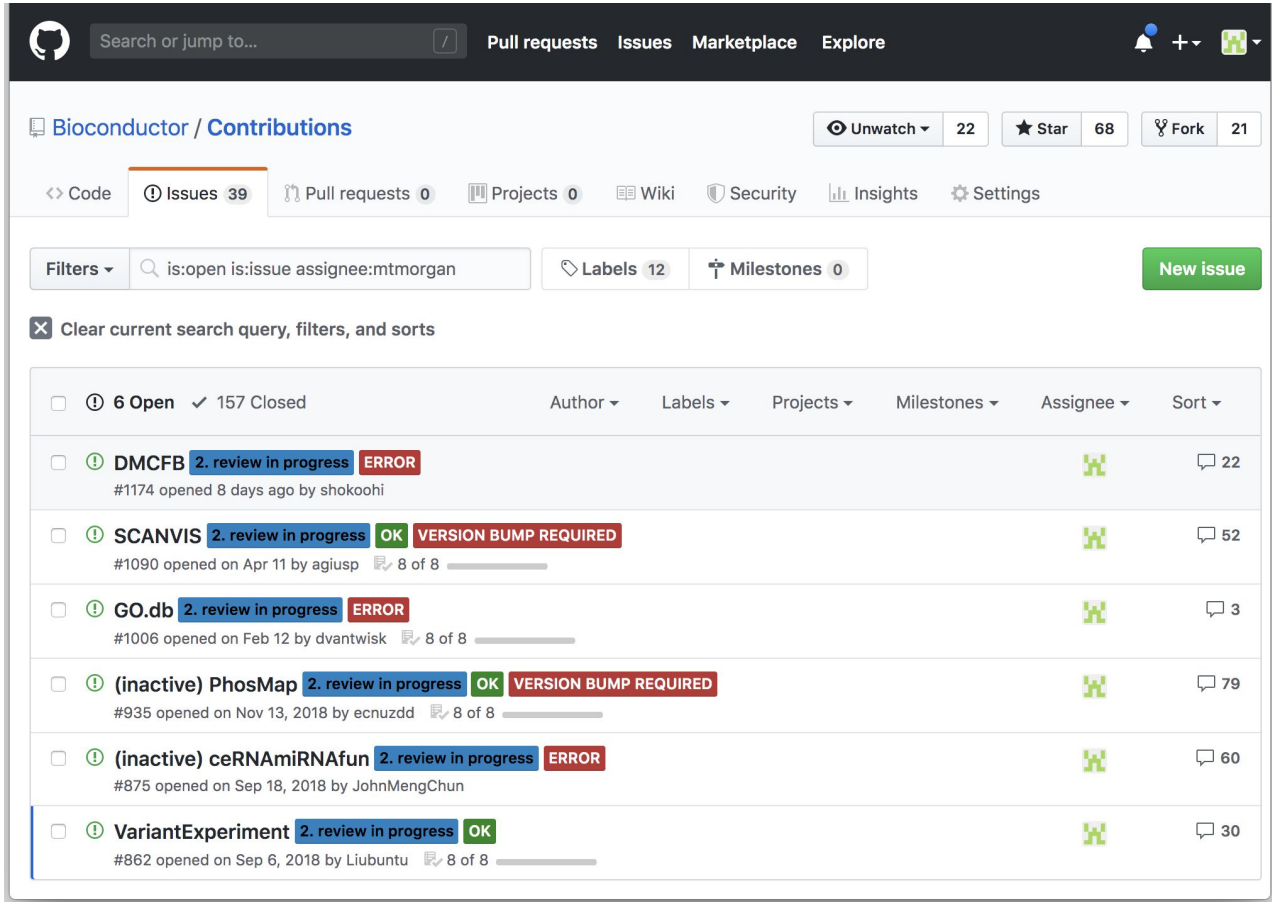
- Our own git

Nightly builds

- Cross-platform ‘integration’ tests

New packages

- Public review via github issues
- help technical quality



Personality

Broad global user base

Supportive community

- For users and developers

Extensive, tested, maintained, and supported software

Stable release branch / flexible & innovative devel branch

Large scientific and professional impact

Objects


```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

Expressive, vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

Domain-specific understanding

Personality

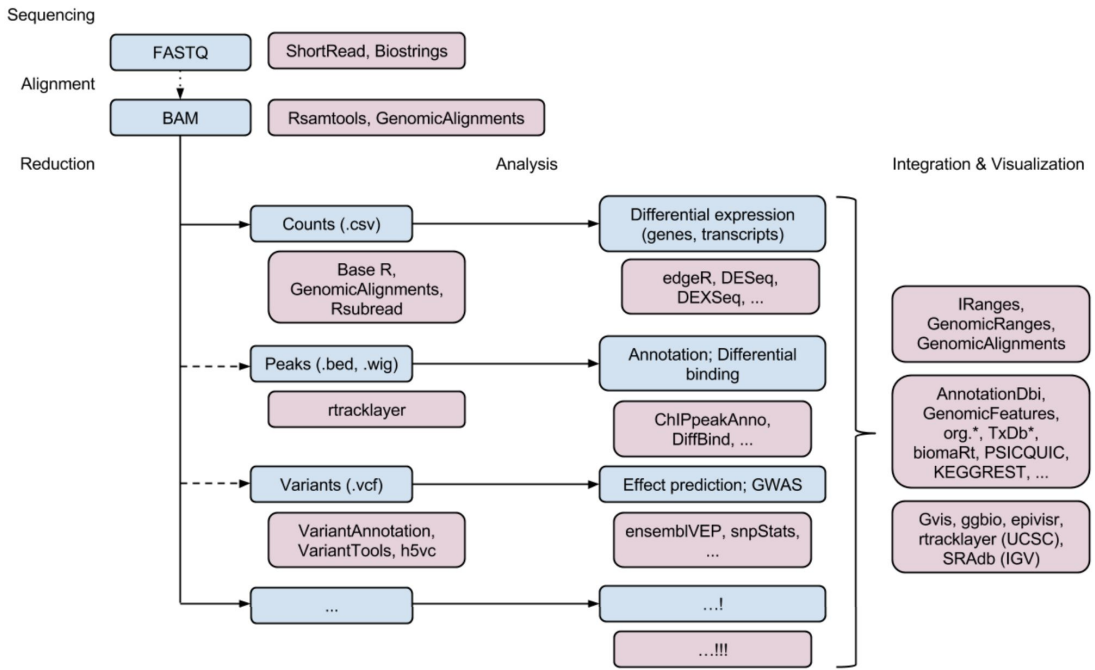
We want interoperable & robust software

Interoperable

- Exploratory: apply different approaches to the same data
- Mature: apply the same approach to different data

Robust

- Data validation
- Easy management of complicated data
- Tested through use



Why (S4) classes & methods?

```
> upstream2k
A DNAStringSet instance of length 26454
      width seq
  [1]  2000 GTTGGTGGCCCACCAGTGC...AGTTTACCGGTTGCACGGT
  [2]  2000 TTATTTATGTAGGCGCCCG...ACGGAAAGTCATCCTCGAT
  [3]  2000 TTATTTATGTAGGCGCCCG...ACGGAAAGTCATCCTCGAT
  ...

> reverseComplement(upstream2k)
A DNAStringSet instance of length 26454
      width seq
  [1]  2000 ACCGTGCAACCGGTAAACT...GCACTGGTGGGCCACCAAC
  [2]  2000 ATCGAGGATGACTTTCCGT...CGGGCGCCTACATAAATAA
  [3]  2000 ATCGAGGATGACTTTCCGT...CGGGCGCCTACATAAATAA
  ...
```

For the user

- Validation, e.g., specific nucleotide 'letters' in the DNA alphabet
- Domain-specific methods, e.g., `reverseComplement()`
- Friendly 'interface'

For the developer

- Efficient 'implementation' to store many large strings

Major *Bioconductor* classes

Vector-like representations

- `DNAStringSet`
- **`GRanges`** (genomic ranges)

Experimental results -- relational constraints

- `SummarizedExperiment`

S4Vectors

- `Vector`
- `List`
- `DataFrame`
- `...`

```
> exons
GRanges object with 515283 ranges and 1 metadata column:
    seqnames            ranges strand |        gene
       <Rle>         <IRanges>  <Rle> | <character>
[1]    chr12 56118262-56118318      + |       84872
[2]     chr7 29126856-29126939      + |   101928168
[3]    chr15 60479178-60479398      + |   101928784
...
-------
seqinfo: 25 sequences (1 circular) from hg38 genome
```

- Required and optional information
- One-based, closed-intervals
- Provenance -- what genome is this?
- Looks two-dimensional, but actually a `Vector`

Major *Bioconductor* classes

Vector-like representations

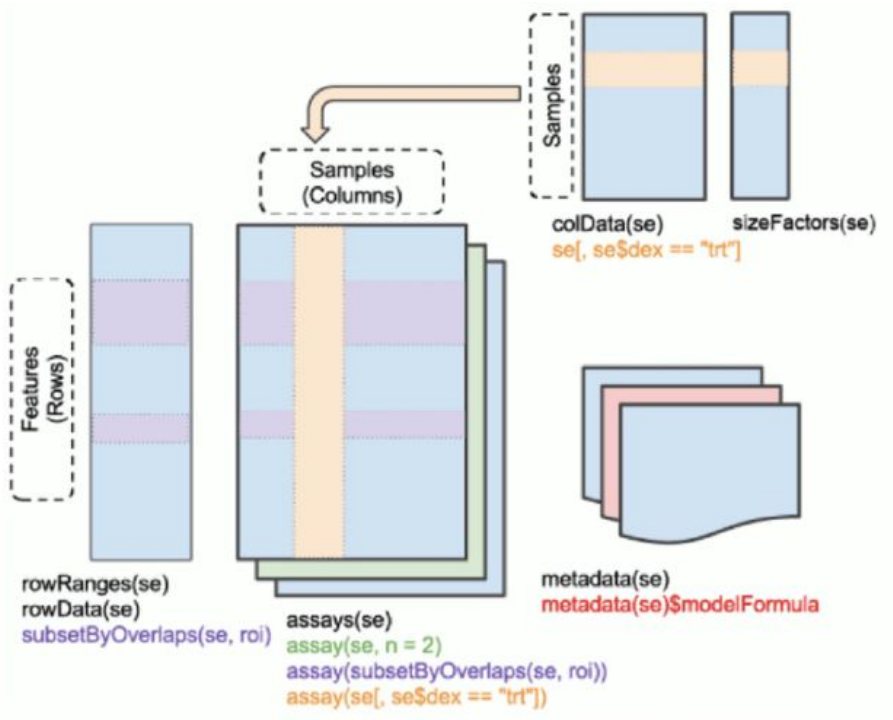
- `DNAStringSet`
- `GRanges` (genomic ranges)

Experimental results -- relational constraints

- **`SummarizedExperiment`**

S4Vectors

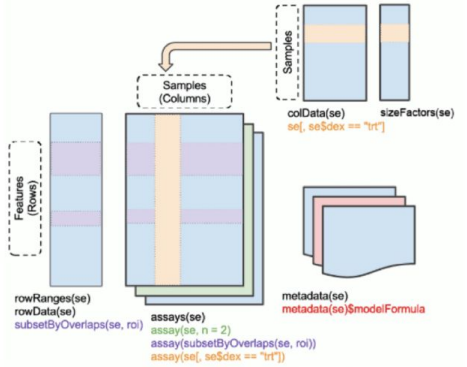
- `Vector`
- `List`
- `DataFrame`
- `...`



SummarizedExperiment

```
> counts     # gene x sample matrix
      SRR1039508 SRR1039509 SRR1039512 ...
ENSG001      679        448        873 ...
ENGS002        0          0          0 ...
ENGS003      467        515        621 ...
...

> samples    # data.frame/tibble/DataFrame
             cell     dex   ...
            <fct>   <fct>
SRR1039508  N61311  untrt  ...
SRR1039509  N61311  trt    ...
SRR1039512  N052611 untrt  ...
...
```




```
se = SummarizedExperiment(
    counts,
    colData = samples
)

## ‘library size’ of each sample
colSums(assay(se))

## subset counts and samples to contain
## only “untrt” samples
untrt = se[, se$dex == “untrt”]
```

Major *Bioconductor* classes

Vector-like representations

- `DNAStringSet`
- `GRanges` (genomic ranges)

Experimental results -- relational constraints

- `SummarizedExperiment`

S4Vectors

- `Vector`
- `List`
- `DataFrame`
- `...`

What is a `Vector`? (approximately...)

- Anything that implements `length()`, `` `[` ``, `` `[<-` `` and `names()`

What is a `List`?

- Anything that implements `Vector` plus `lengths()`, `` `[[` ``, `` `[[<-` ``

What is a `DataFrame` *column*?

- Anything that implements `Vector`, or an atomic `vector` or `list`

Major *Bioconductor* classes

Vector-like representations

- `DNAStringSet`
- `GRanges` (genomic ranges)

Experimental results -- relational constraints

- `SummarizedExperiment`

S4Vectors

- `Vector`
- `List`
- `DataFrame`
- `...`

```
> dna = DNAStringSet(c("AACTG", "CCCATG"))
> gr = GRanges(c("chr1:21-30", "chr2:55-59"))

> is(dna, "Vector")
[1] TRUE
> is(gr, "Vector")
[1] TRUE

> df = DataFrame(id = 1:2, gr, dna)
> df
DataFrame with 2 rows and 3 columns
         id         gr            dna
  <integer>  <GRanges> <DNAStringSet>
1         1 chr1:21-30          AACTG
2         2 chr2:55-59         CCCATG
```

Objects

Encourage interoperability between packages

Provide robust validation

Enable easy management of complex relational data

Separate

- 'interface' experienced by the user
- 'implementation' available to the developer

Reproduce & communicate

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

Vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

Domain-specific understanding

Personality

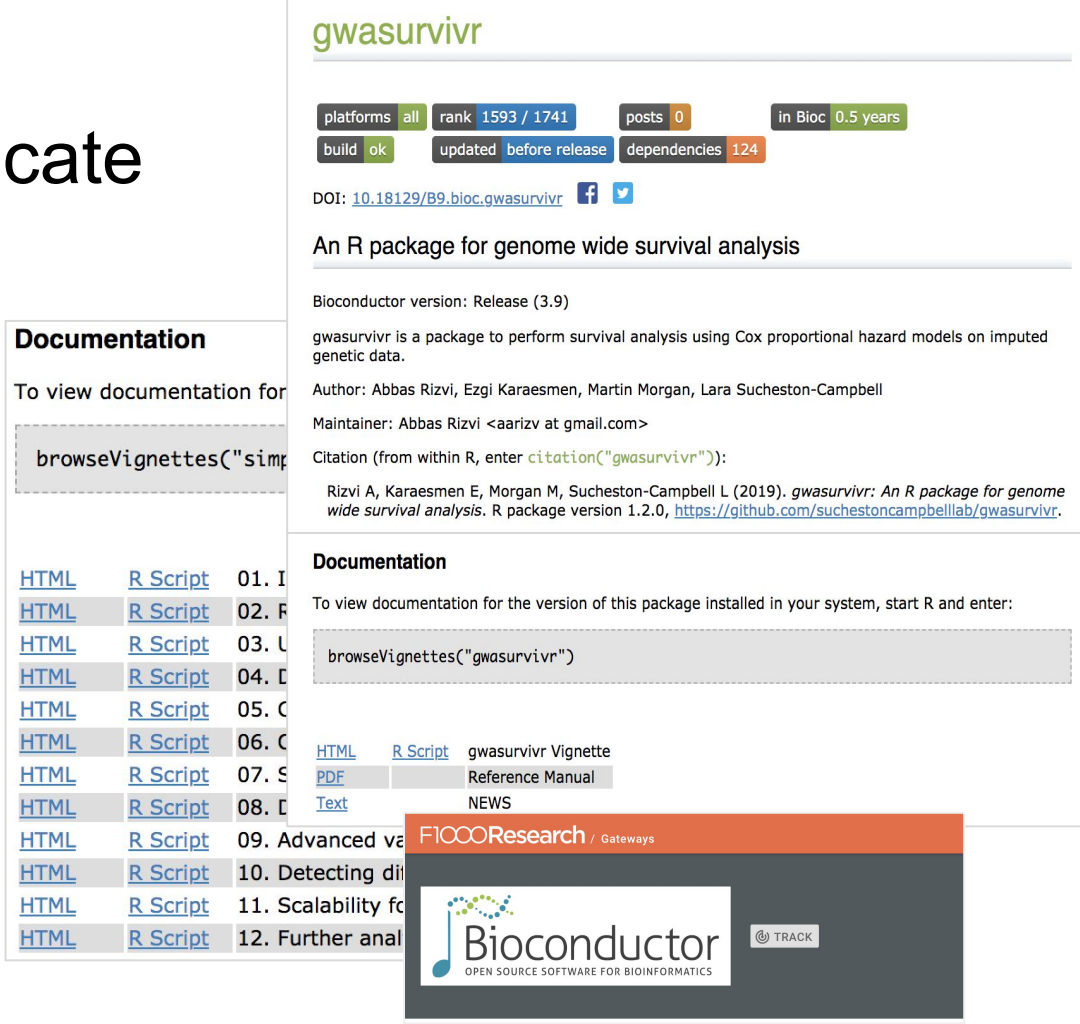
Reproduce & communicate

Integrative documentation

- Extensive, computable vignettes
  - Required of all *Bioconductor* packages
- Comprehensive workflows
- Workshop compendia (e.g., BioC2019) & community-developed material (e.g., osca)

Paths to academic recognition

- Individual packages, DOI, and citations
- F1000 Research / *Bioconductor* channel



Challenges & opportunities

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

Old-fashioned calculator
Statistical programming language
Interactive (immediate) visualization

Expressive, vectorized

Objects for coordinating data
Computable results
Interface (not implementation)

Domain-specific understanding

Personality

Large data representation, e.g., single-cell data

Illusions

- Show and update ‘corners’ of data
- Delay full update until needed

On-disk representation

- Only row & column indexes in memory

Chunk-wise parallel processing

- Read blocks of memory, perform calculation
- Often a reduction, e.g., `colSums()`, that fits in memory

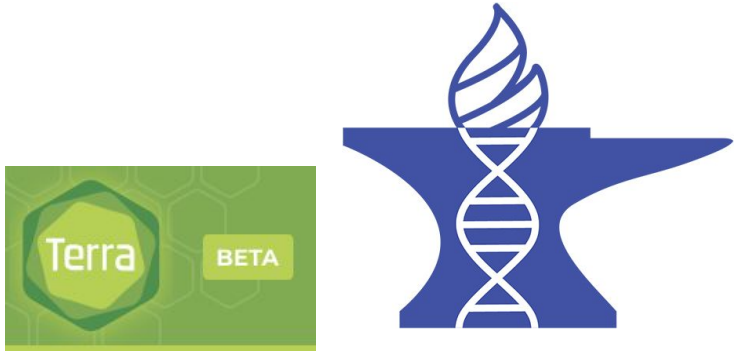
```
> tenx = TENxBrainData::TENxBrainData()
> dim(tenx) # on-disk SummarizedExperiment
[1]   27998 1306127

> log(1 + assay(tenx))          # illusion

> cidx = sample(ncol(tenx), 200)
> tenx_subset = tenx[, cidx]
> sum(assay(tenx_subset) == 0) /
    prod(dim(tenx_subset))
[1] 0.9289764                   # sparse!

> colSums(assay(tenx_subset)))  # realized
```

The AnVIL cloud https://anvilproject.org



US NHGRI initiative

- Genomic Data Commons -- large ‘consortium’ data
- Terra -- Cloud-based computation
- *Python*, *R* / *Bioconductor*, *Galaxy*, and other tools

Benefits

- Simple -- e.g., containerized *Bioconductor*
- Fast -- cloud-based data access
- Scalable, e.g., configured *BiocParallel*
- Secure

Alternative paradigms in *R*

Tidy data

- Long-form `tibble` / `data.frame`

Bioconductor data

- Generalized `Vector`
  - `DNAStringSet`, `GRanges`
- Structured, efficient matrix
  - `assay()` data of `SummarizedExperiment`
- Relational
  - `SummarizedExperiment`
  - Row & column names serve as unique keys for joins

```
> assay(airway) %>% ... %>%
    ## ...: matrix -> tidy tibble
    group_by(sample) %>%
    summarize(lib_size = sum(value))

> assay(airway) %>%
    colSums() %>%
    enframe("sample", "lib_size")

# A tibble: 8 x 2
  sample     lib_size
  <fct>         <int>
1 SRR1039508 20637971
2 SRR1039509 18809481
...
```

Challenges & opportunities

Scalability

- Memory management & delayed evaluation

Cloud-based computation

- Containerization, e.g., reliable binary installations
- Fast data access
- Scalable

Working well with other parts of the evolving *R* & broader ecosystems

- tidyverse, python, cloud, ...

I talked about...


```
1 + 2
x = rnorm(100)
hist(x)

y = x + rnorm(100)

df = data.frame(x, y); plot(y ~ x, df)
fit = lm(y ~ x, df)
anova(fit); abline(fit)

library(ggplot2)

ggplot(df, aes(x, y)) + geom_point()
```

4. Reproduce & communicate

5. Challenges & opportunities

3. S4 objects & interdependent packages

1. High-throughput genomics

2. The *Bioconductor* ecosystem

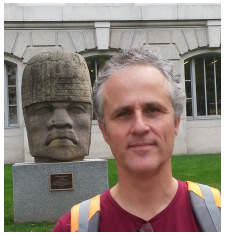
Conclusions and acknowledgements

Bioconductor core team & close collaborators

- Funded by US National Institutes of Health, European Union, Chan-Zuckerberg Initiative ...

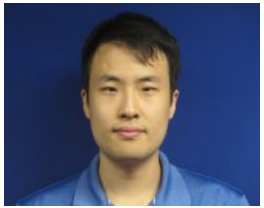
World-wide community of users & developers

Technical and scientific advisory boards







Acknowledgements

Research reported in this presentation was supported by the NHGRI and NCI of the National Institutes of Health under award numbers U41HG004059, U24CA180996, and U24CA232979. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

This work was performed on behalf of the SOUND Consortium and funded under the EU H2020 Personalizing Health and Care Program, Action contract number 633974.

A portion of this work is supported by the Chan Zuckerberg Initiative DAF, an advised fund of Silicon Valley Community Foundation.