Enabling collaborative software development with inner and open-source

Michael Lawrence

November 6, 2019

Genentech
A Member of the Roche Group

R has many strengths

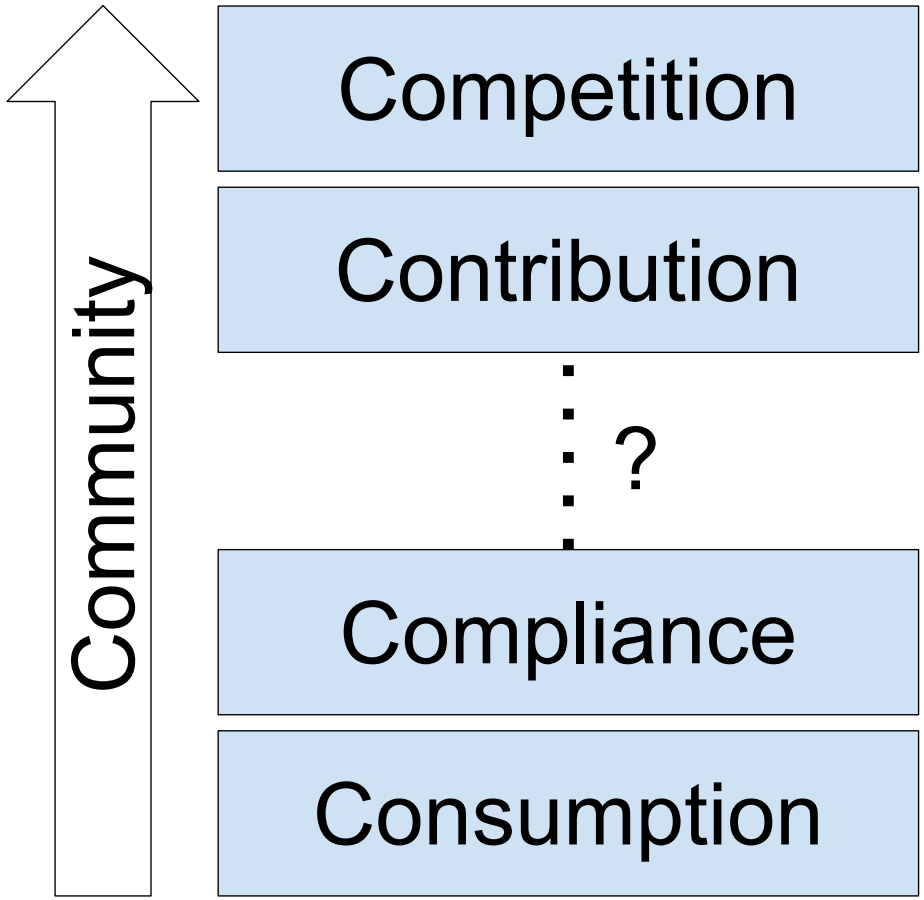
- R is free and open
- R is extensible and programmable
- R has thousands of high quality packages

R's strengths are also challenges

- ~~R is free~~ R is an investment
- ~~R is extensible and programmable~~ R needs to be programmed
  - How do we develop software for clinical science?
    - That enables collaboration across the enterprise and the industry?
- ~~R has thousands of high quality packages~~ R has too many packages
  - How do we manage shared analytical computing environments?
  - Which packages are validated?
  - How do we manage license compliance?

Open-source adoption follows a pattern

Lee Calcote, Open Source Leadership Summit 2019



From paying a software company, to becoming a software company.

Contributing to open-source has clear business benefits

Late-stage development case

Enables pre-competitive collaboration and standardization on cross-industry problems

- Clinical reporting infrastructure
- Interaction with regulatory authorities

Early-stage research case

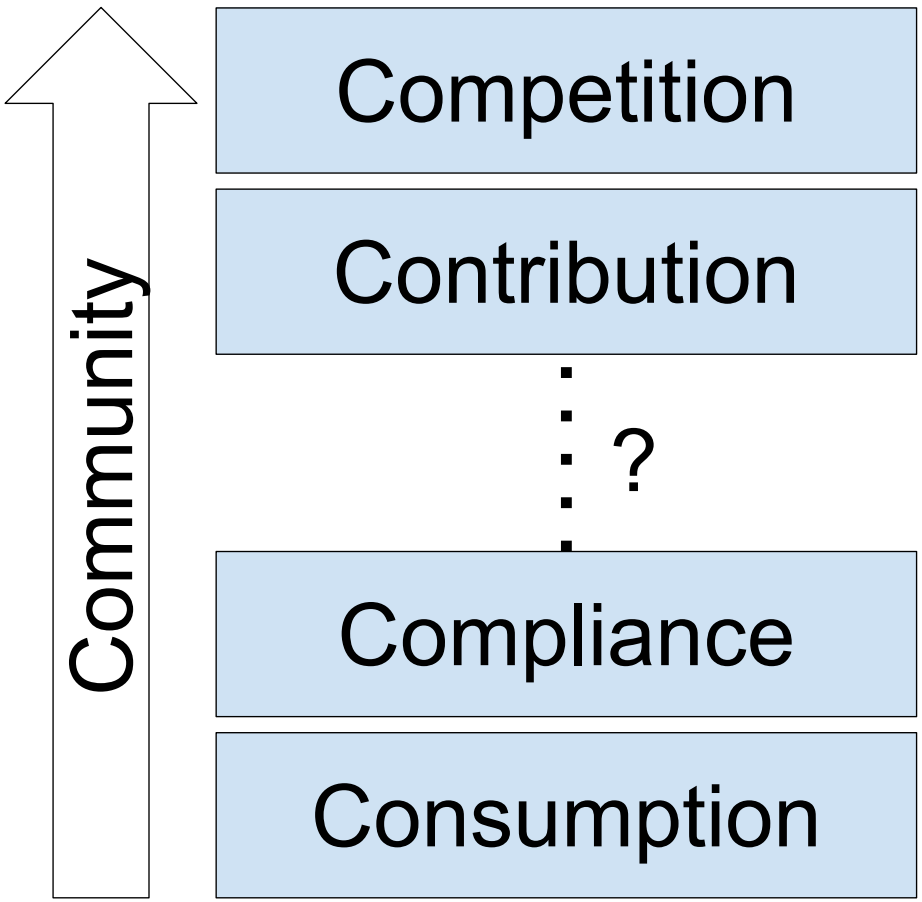
- Open-source software publication is an intellectual contribution to science, equivalent to a paper
- Supports reproducibility of scientific results

Engineering case

Setting out to be open-source puts the right perspective on development and leads to better software

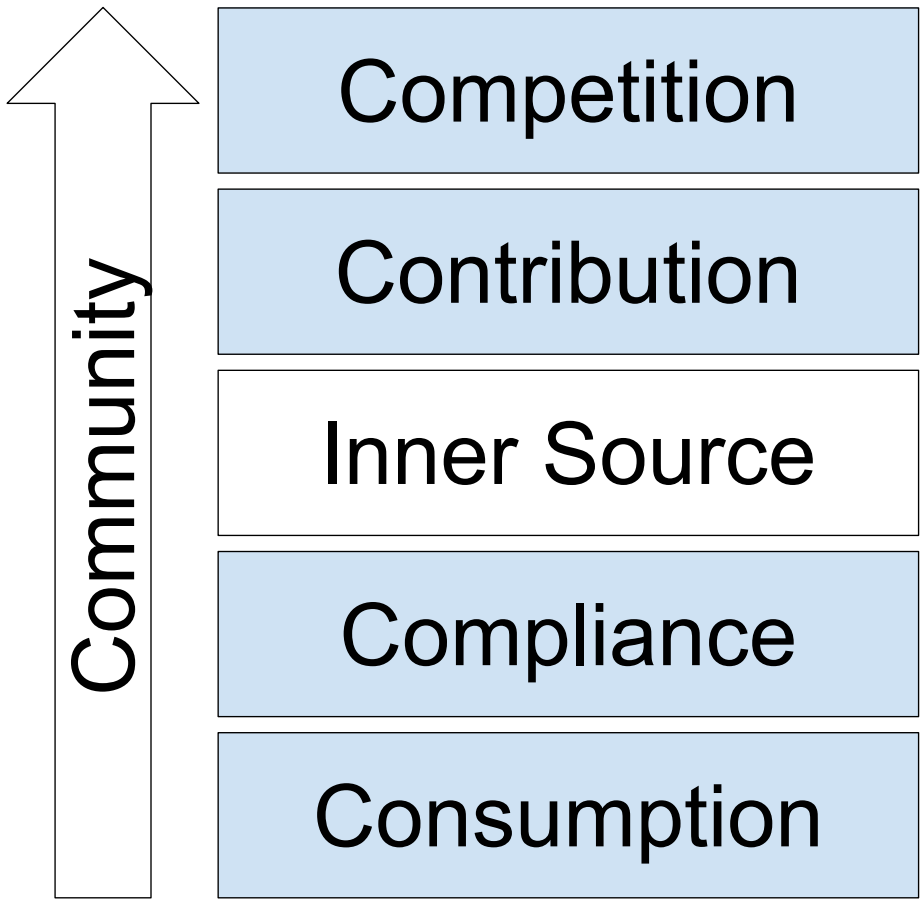
Inner source is a bridge towards open-source

Lee Calcote, Open Source Leadership Summit 2019



Inner source is a bridge towards open-source

Lee Calcote, Open Source Leadership Summit 2019

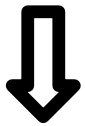


Inner source is based on open source

Enabling collaborative development across an enterprise by applying open-source *principles*, cultural *values* and *practices*.

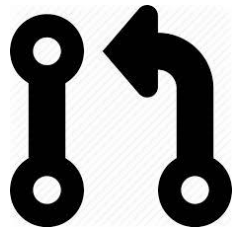
Three principles underlie the Open philosophy

Open **communication**



Open **code**, **documentation**

Open **contribution**



Three cultural values guide the Open community

Egalitarian	anyone, in principle, can contribute
Meritocratic	contributions judged on merit
Self-organizing	ownership implied through contribution

Three practices drive Open software development

Collaborative development contributors submit patches, reviewed, accepted and merged by project owners

Participatory reuse collaborative development of shared dependencies

Self-selection of tasks contributors decide what and when to contribute

Inner source benefits the organization

Agility Allows resource sharing, preempting fixed development schedules

Reduced cost Encourages code reuse, standardization

Innovation Breaks down barriers and spurs innovation

Knowledge management Encourages good documentation, spreads knowledge through shared support, code reading

Motivated workforce Volunteers are intrinsically motivated

Quality assurance "Given enough eyeballs, all bugs are shallow" - Linus's Law

Primed for open-source Readies organization for open-source engagement

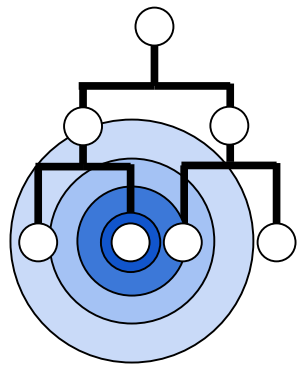
Inner source benefits the individual

Development Exposure to new types of projects drives individual development

Influence Contributing across organizations more broadly impacts the business

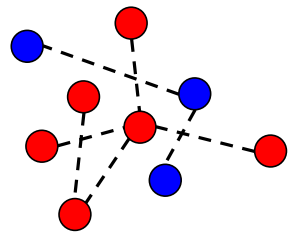
Broader outlook Reveals the bigger picture

Primed for open-source Experience with inner-source communities carries over to open-source



Inner source has many challenges

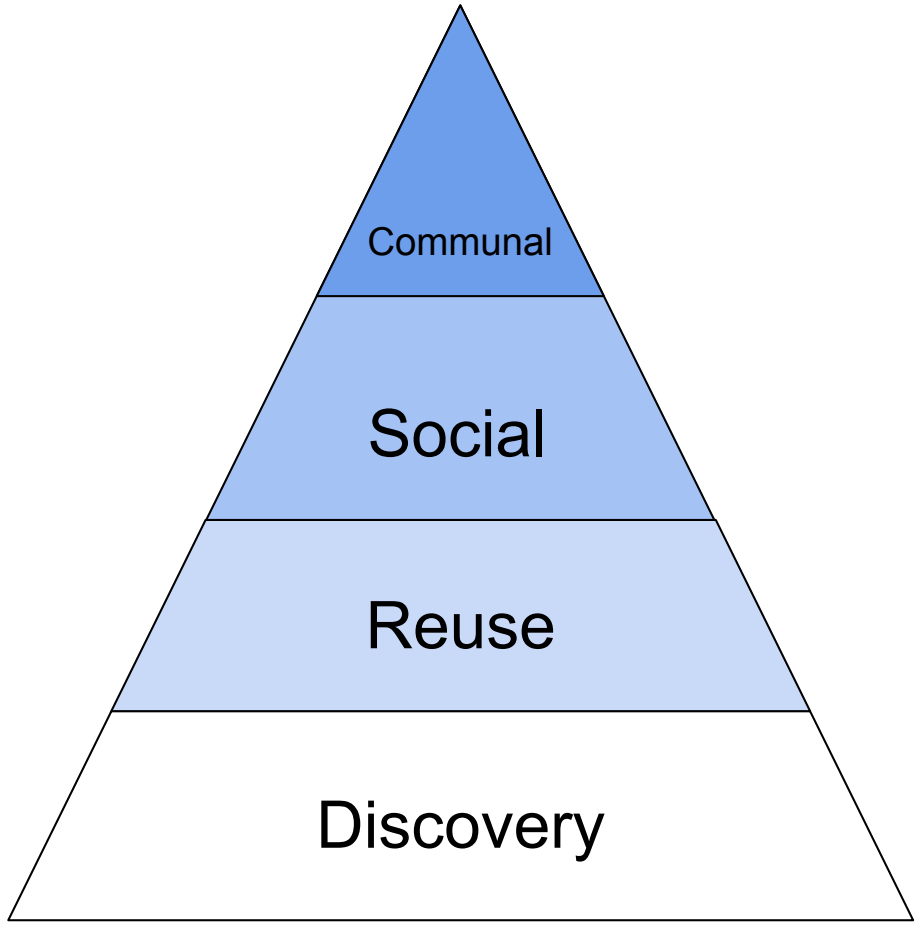
- Large, complex ecosystem, discoverability issues
- Changes in working style and culture
- Management challenges in balancing incentives, and allowing for self-organization and self-selection of tasks
- Fear of code scrutiny
- Conflicts around issues, contributions, forking
- Competition between similar projects
- Competition between local and global interests
- Knowing what to inner source



Our strategy reconciles competing concerns

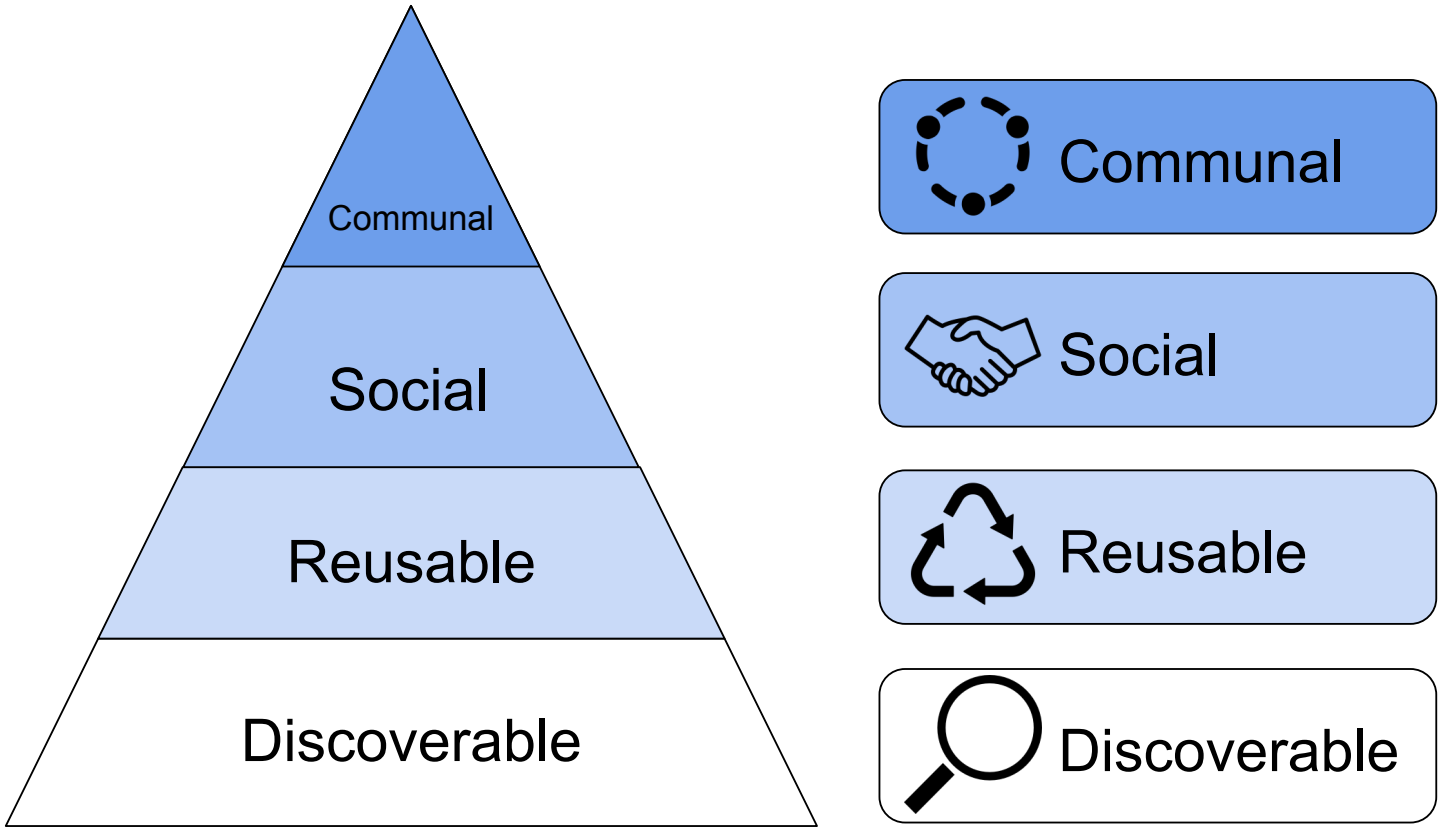
Enable collaborative development while mitigating risk and controlling cost through incremental strategies towards right-sized inner sourcing policies and processes.

The inner source pyramid is an incremental strategy



Two-way inner-source contracts formalize expectations

Licenses are to open-source as contracts are to inner-source.



By clarifying expectations, contracts facilitate decision-making and integrate with goals and performance evaluation

Contracts pertain to both developers and participants

	Developer obligations	Participant obligations
 Communal	Establish formal community structure, standardize, automate	Adhere to formal standards, respec community decisions
 Social	Design and document software for extensibility, on-board participants, respond to pull requests	Follow on-boarding procedures, ensure pull requests are in scope, iterate on contributions
 Reusable	Design and document for reuse, respond to support requests	Read documentation, submit actionable and in scope support requests
 Discoverable	Annotates project with sufficient metadata for discovery	Communicate feedback to maintainers, do not reinvent

There are many archetypes of open-source projects

Open Source Archetypes: A Framework For Purposeful Open Source (Mozilla)

Business-to-business

Drives platform adoption with modifiable base

Wide open

Contributions at every level, democratic governance

Multi-vendor infrastructure

Vendors share cost and differentiate on deployments

Mass market

Fully open, but separates non-technical users and developers

Trusted vendor

Single vendor offers insurance against lock-in

Upstream dependency

Contributions from dependent projects

Rocket ship to Mars

Less collaborative, but transparent, drive towards singular goal

Specialized library

Contributions require specialized knowledge

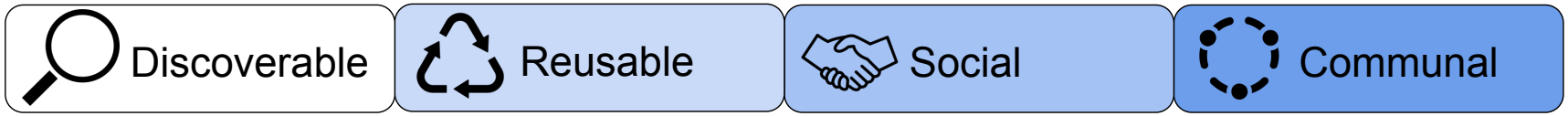
Controlled ecosystem

Core team provides infrastructure, contributors extend with plugins

Bathwater

Thrown out with no promise of maintenance or support

Projects climb the pyramid by evolving through archetypes



Bathwater

Rocket ship to Mars

Trusted vendor

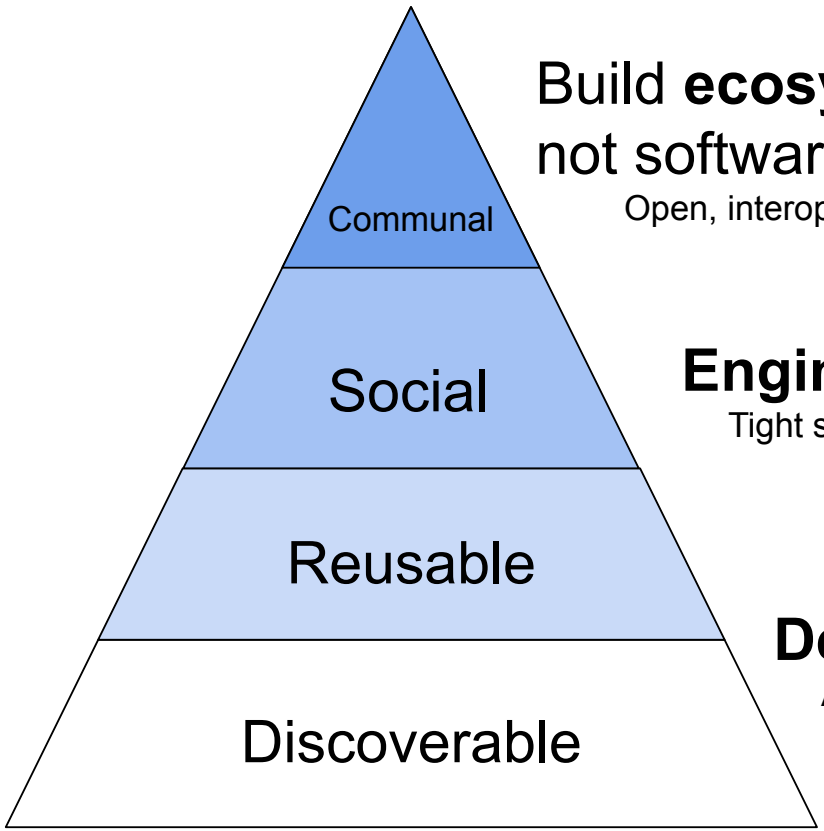
Multi-vendor infrastructure

Upstream dependency

Controlled ecosystem

Three principles for sustained scientific software innovation

Synergistic with inner sourcing



Build **ecosystems**,
not software suites

Open, interoperable abstractions

Engineer early

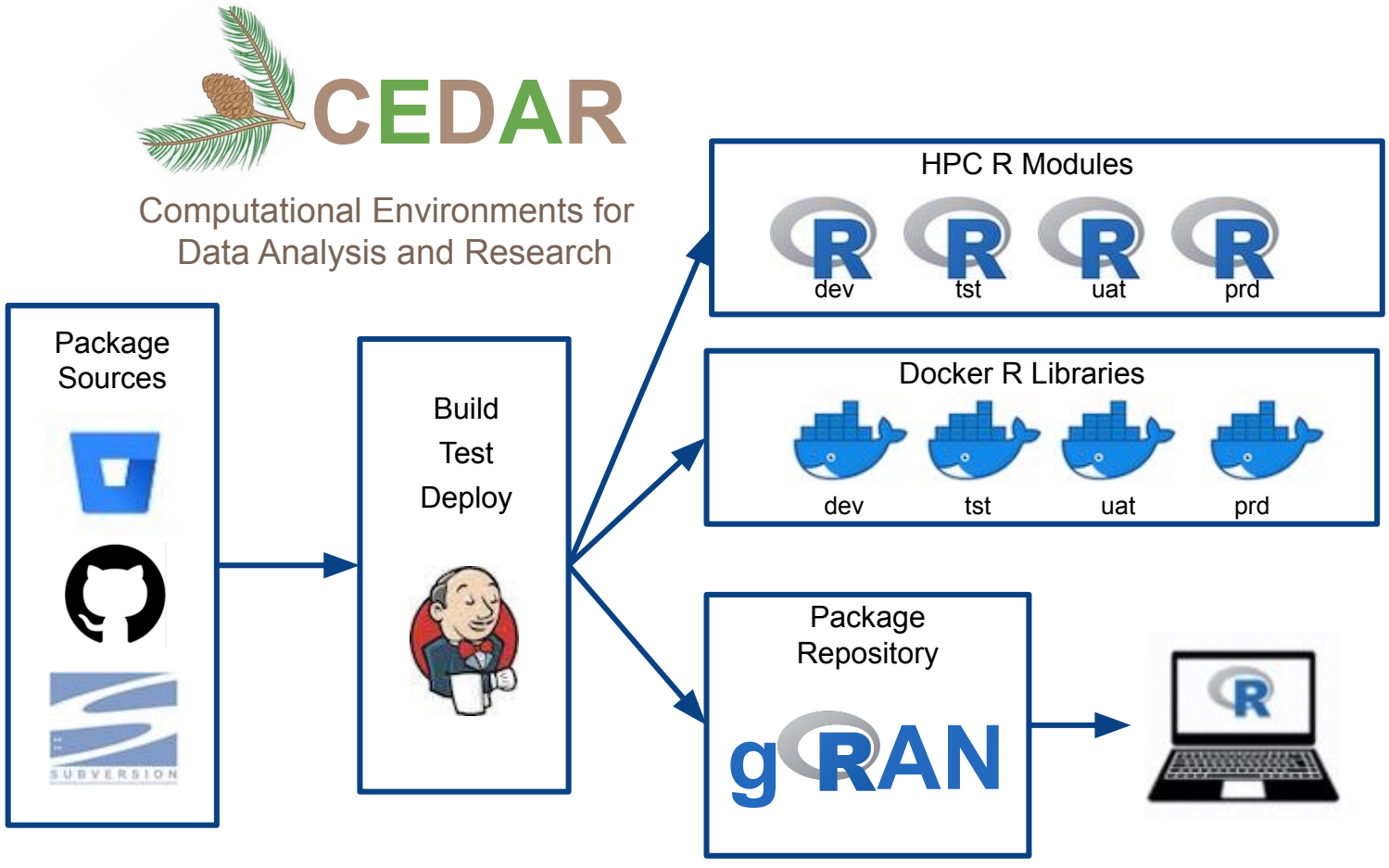
Tight science/engineering partnership

Democratize contribution

Anyone can be a data scientist

Anyone can extend our analytical computing environment

while following best practices



Package splash pages make packages findable

gp.sa.diff: Testing for Differential Events over Lots of Features

Provides methods to test for differential events between samples in 'omics datasets containing many features. This includes differential expression for RNA-seq or microarray data, differential binding for ChIP-seq data, differential abundance for mass cytometry, differential interactions for Hi-C data, etc.

GRAN Release: GRANtst

Build status: check note(s)

Test Coverage: 93.65 %

Authors: Aaron Lun [cre, aut]

Maintainer: Aaron Lun luna@gene.com

Installation

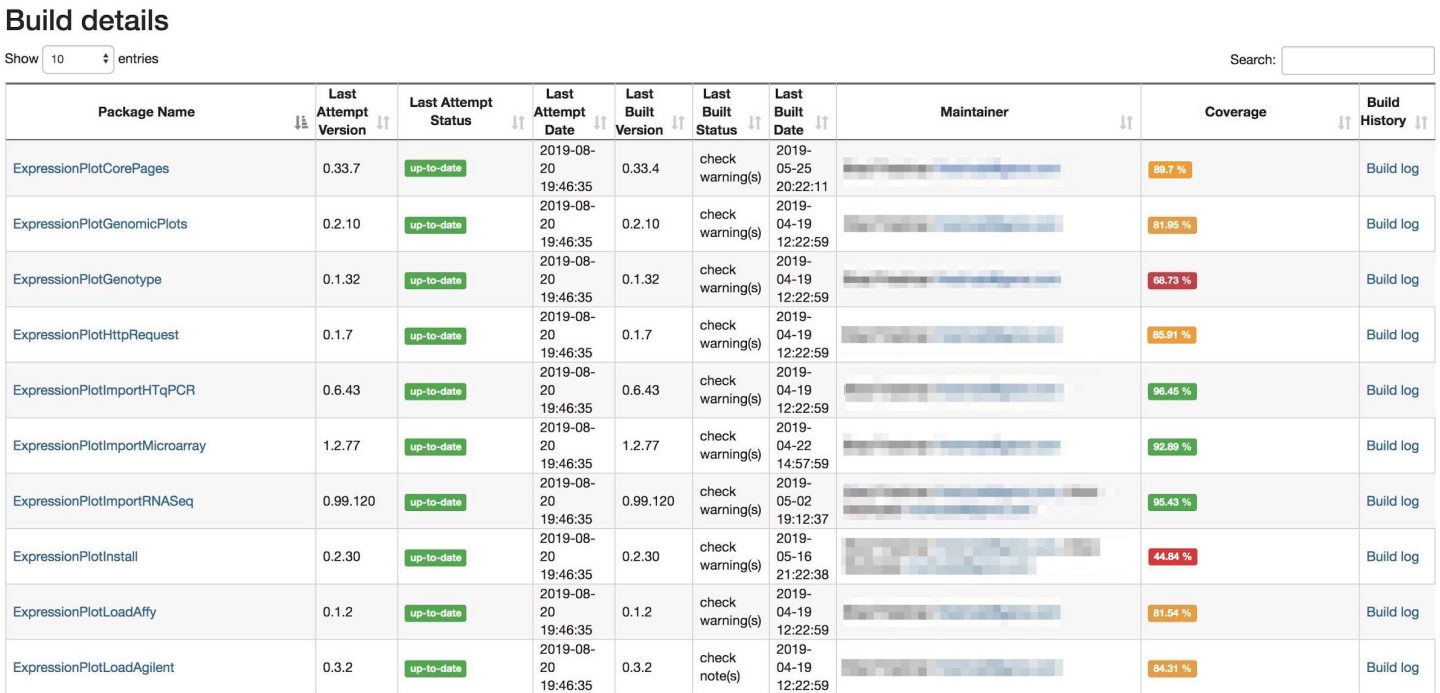
To install this package, start R and enter:

```
source("http://reztst.gene.com/gran/getGRAN-tst.R")
library(GRANtst)
install_packages("gp.sa.diff", type="source")
```

Details

Package	gp.sa.diff
Version	0.99.27
Date	2019-08-13
Title	Testing for Differential Events over Lots of Features
Description	Provides methods to test for differential events between samples in 'omics datasets containing many features. This includes differential expression for RNA-seq or microarray data, differential binding for ChIP-seq data, differential abundance for mass cytometry, differential interactions for Hi-C data, etc.

Package build reports bring transparency



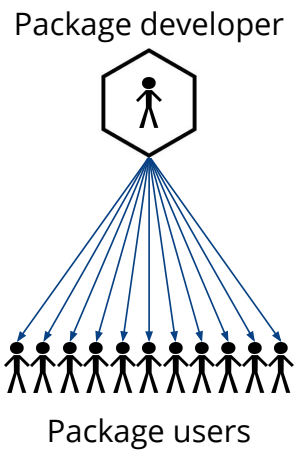
Build details

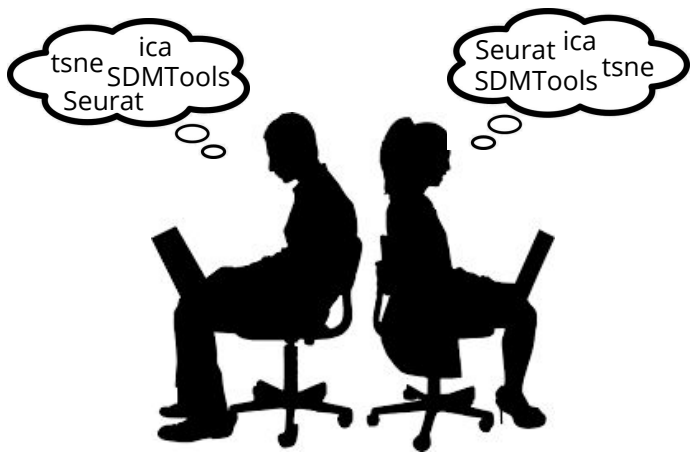
Show 10 entries Search:

Package Name	Last Attempt Version	Last Attempt Status	Last Attempt Date	Last Built Version	Last Built Status	Last Built Date	Maintainer	Coverage	Build History
ExpressionPlotCorePages	0.33.7	up-to-date	2019-08-20 19:46:35	0.33.4	check warning(s)	2019-05-25 20:22:11		89.7 %	Build log
ExpressionPlotGenomicPlots	0.2.10	up-to-date	2019-08-20 19:46:35	0.2.10	check warning(s)	2019-04-19 12:22:59		81.95 %	Build log
ExpressionPlotGenotype	0.1.32	up-to-date	2019-08-20 19:46:35	0.1.32	check warning(s)	2019-04-19 12:22:59		68.73 %	Build log
ExpressionPlotHttpRequest	0.1.7	up-to-date	2019-08-20 19:46:35	0.1.7	check warning(s)	2019-04-19 12:22:59		85.91 %	Build log
ExpressionPlotImportHTqPCR	0.6.43	up-to-date	2019-08-20 19:46:35	0.6.43	check warning(s)	2019-04-19 12:22:59		96.45 %	Build log
ExpressionPlotImportMicroarray	1.2.77	up-to-date	2019-08-20 19:46:35	1.2.77	check warning(s)	2019-04-22 14:57:59		92.89 %	Build log
ExpressionPlotImportRNASeq	0.99.120	up-to-date	2019-08-20 19:46:35	0.99.120	check warning(s)	2019-05-02 19:12:37		95.43 %	Build log
ExpressionPlotInstall	0.2.30	up-to-date	2019-08-20 19:46:35	0.2.30	check warning(s)	2019-05-16 21:22:38		44.84 %	Build log
ExpressionPlotLoadAffy	0.1.2	up-to-date	2019-08-20 19:46:35	0.1.2	check warning(s)	2019-04-19 12:22:59		81.54 %	Build log
ExpressionPlotLoadAgilent	0.3.2	up-to-date	2019-08-20 19:46:35	0.3.2	check note(s)	2019-04-19 12:22:59		84.31 %	Build log

Spyware makes software (and people) discoverable

datetime	user	host	R version	package	package version
2019-08-08 16:14:58	user1	host1	R-3.6.0-Bioc-3.9-prd	package1	1.1.11
2019-08-08 16:12:01	user	host1	R-3.6.0-Bioc-3.9-prd	package2	0.18.5
2019-08-06 07:29:33	user2	host2	R-3.6.0-Bioc-3.10-dev	package3	0.2.4





Jaccard-based clustering reveals many package cliques

Package suggesting enhances discoverability

Frequently bought together

Total price: $17.93

Add all three to Cart

Add all three to List

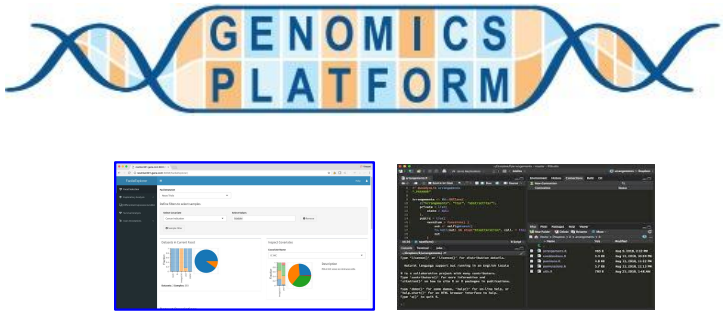
Frequently loaded together

Total deps: 118

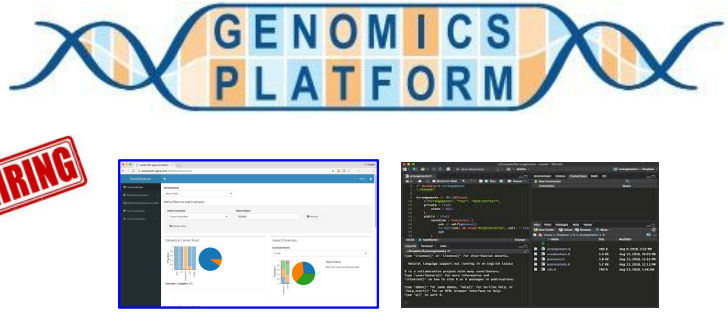
Load all three

Add all three to container

Abstractions enable decoupled software development

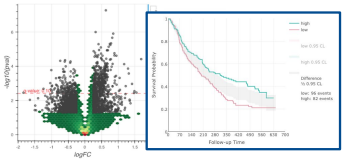


Abstractions enable decoupled software development



Common, interoperable data abstraction

Interfaces for interactive data **analysis**

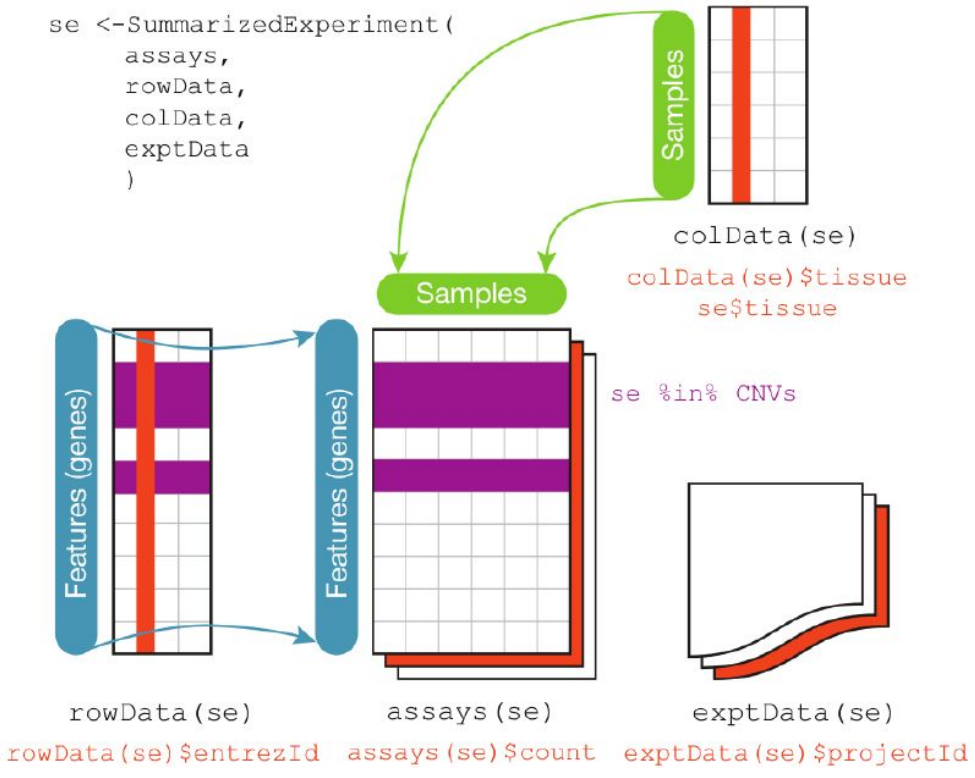


Analytical pipelines

Data import and **integration** pipelines

Any summarized genomic data

We adopted a common, third-party abstraction



Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

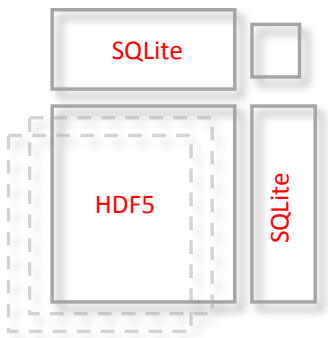
LOOMPY

hail

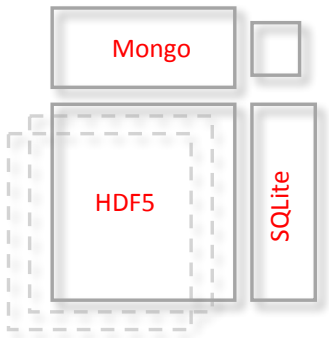
Why is an abstraction important?

- Bridges implementation to the user's mental model
- Decouples components to minimize system viscosity, allowing
  - Adapting implementation as datasets grow in size or as new scientific questions arise
  - Decentralized development, independent evolution of components

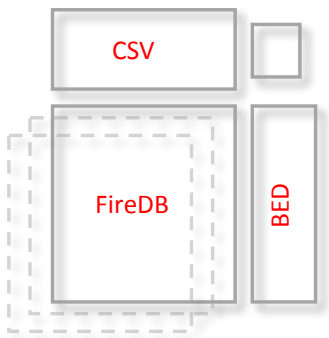
RNASeq (Facile)



Germline Mutation



ChIPSeq + ATACSeq



Why is a public standard important?

- Enables interfacing with externally developed systems
  - 292 Bioconductor packages use the standard
    - Interfaces to Python (Loom) and Spark (Hail)
    - Importers from public databases
  - DelayedArray for transparently keeping data out of memory
- Likely to be independently adopted by other parts of the enterprise

Inner sourcing enables entrepreneurial leadership

Acknowledgement: Hillary Parker

The **Harvard Business Review** proposes that there are **three tiers** of leadership in an innovative organization:

***Entrepreneurial Leaders*: Innovate** by applying direct knowledge of business needs to meet strategic goals from above

Inner sourcing enables entrepreneurial leadership

Acknowledgement: Hillary Parker

The **Harvard Business Review** proposes that there are **three tiers** of leadership in an innovative organization:

***Enabling Leaders*: Develop** people, **connect** people, establish **communication** channels to directly enable entrepreneurialism
***Entrepreneurial Leaders*: Innovate** by applying direct knowledge of business needs to meet strategic goals from above

Inner sourcing enables entrepreneurial leadership

Acknowledgement: Hillary Parker

The **Harvard Business Review** proposes that there are **three tiers** of leadership in an innovative organization:

***Architecting Leaders*: Setup the game board, define overarching strategy, establish conditions for entrepreneurialism**
***Enabling Leaders*: Develop people, connect people, establish communication channels to directly enable entrepreneurialism**
***Entrepreneurial Leaders*: Innovate by applying direct knowledge of business needs to meet strategic goals from above**

Acknowledgements and Resources

Fruitful Discussions Sally Madden, Kiran Mukhyala

Open Source Program Office guides (TODO group)

Review paper Capraro, M., & Riehle, D. (2017, February). Inner Source Definition, Benefits, and Challenges. ACM Computing Surveys, vol. 49, no. 4, article 67.

InnerSource Commons website (Paypal)

Open-source Archetypes white paper (Mozilla)

CEDAR Rena Yang