San Francisco | 5 – 7 June, 2017

Enhancing reproducibility, comparability and discoverability of results in multi-analyst settings

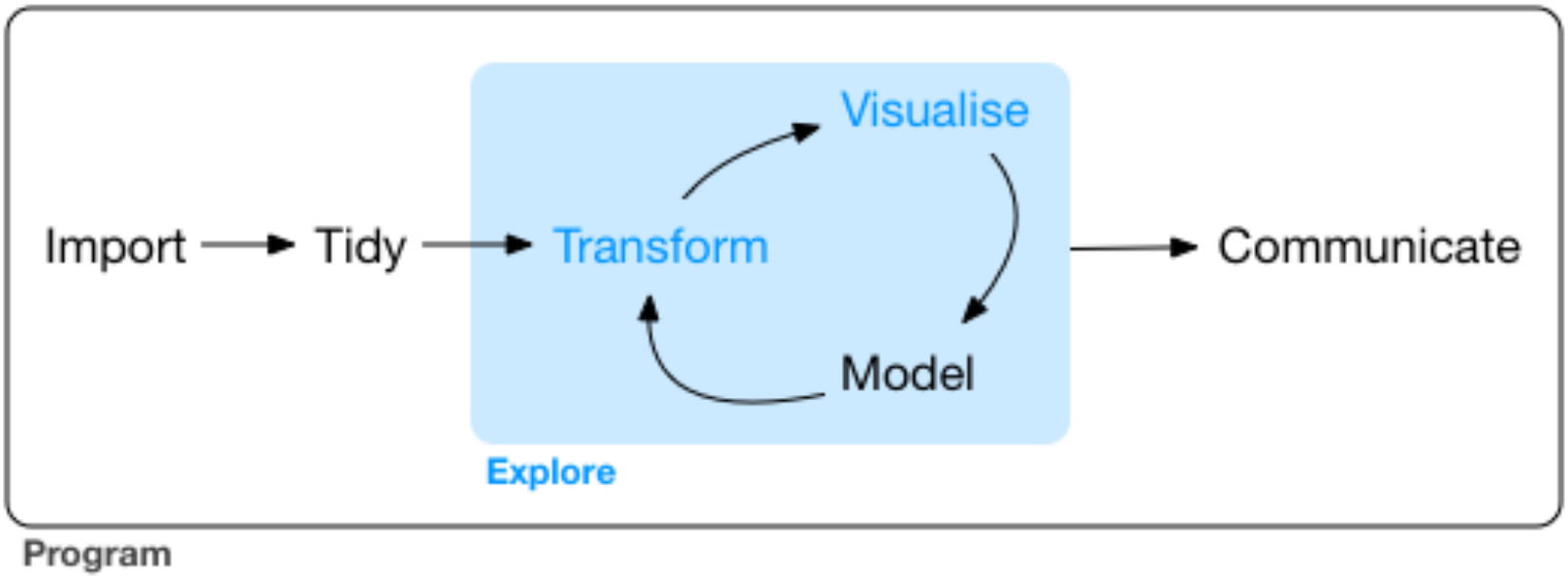
Gabriel Becker

@groundwalkergmb

The Setting

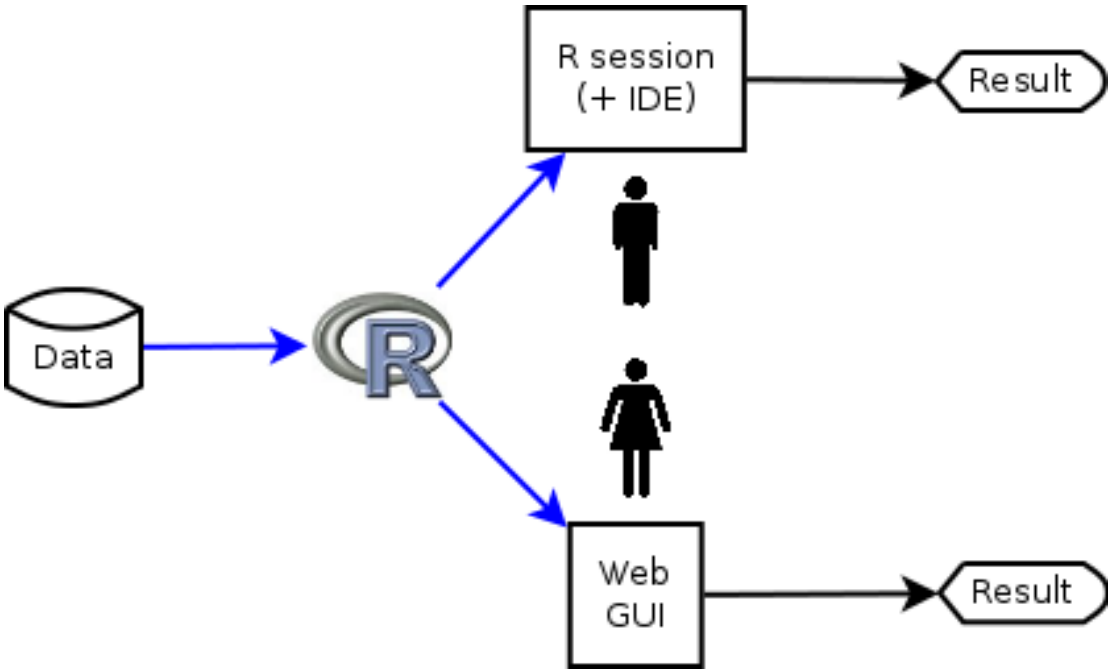
- ~30 PhD bioinformaticians
- R + Bioconductor shop
- Shared “big-ish” data
- We publish
  - Come back to analysis months/years later
- We write packages
  - Used in analyses

The individual data scientist

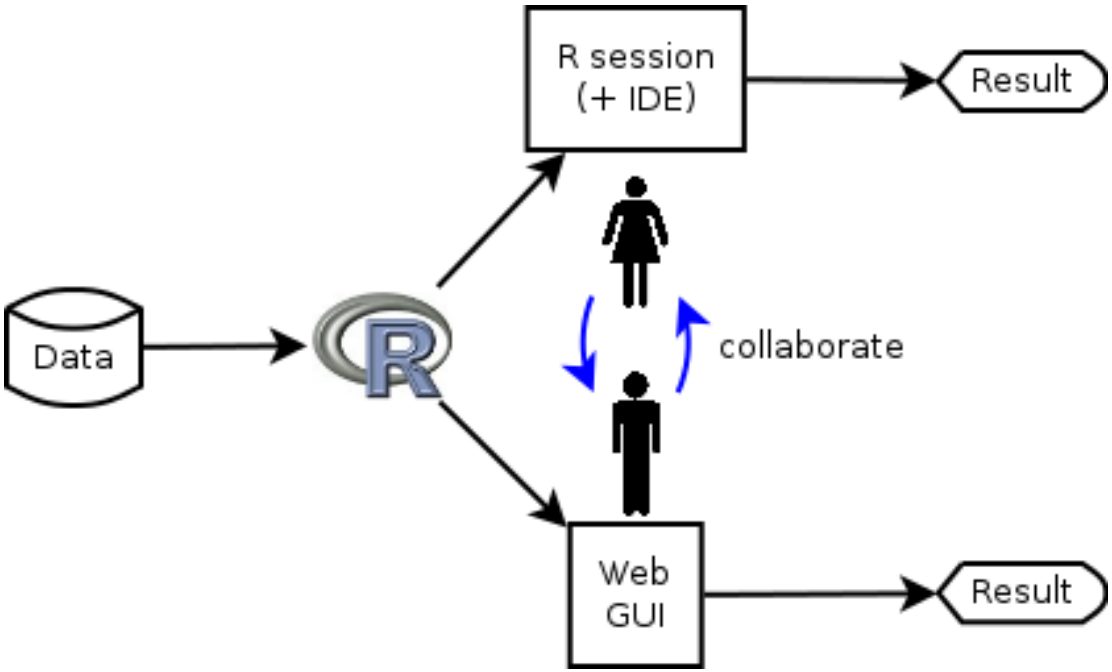


COLLABORATIVE SCIENCE

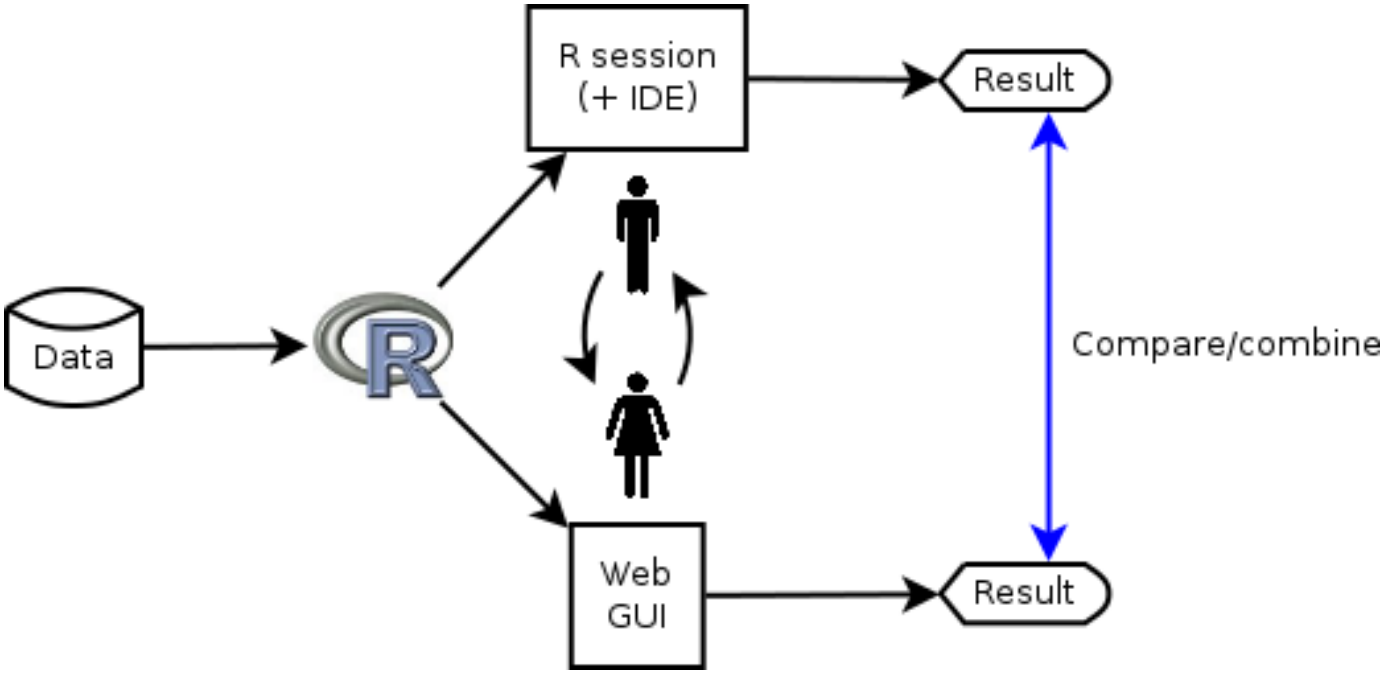
Multiple interfaces to data



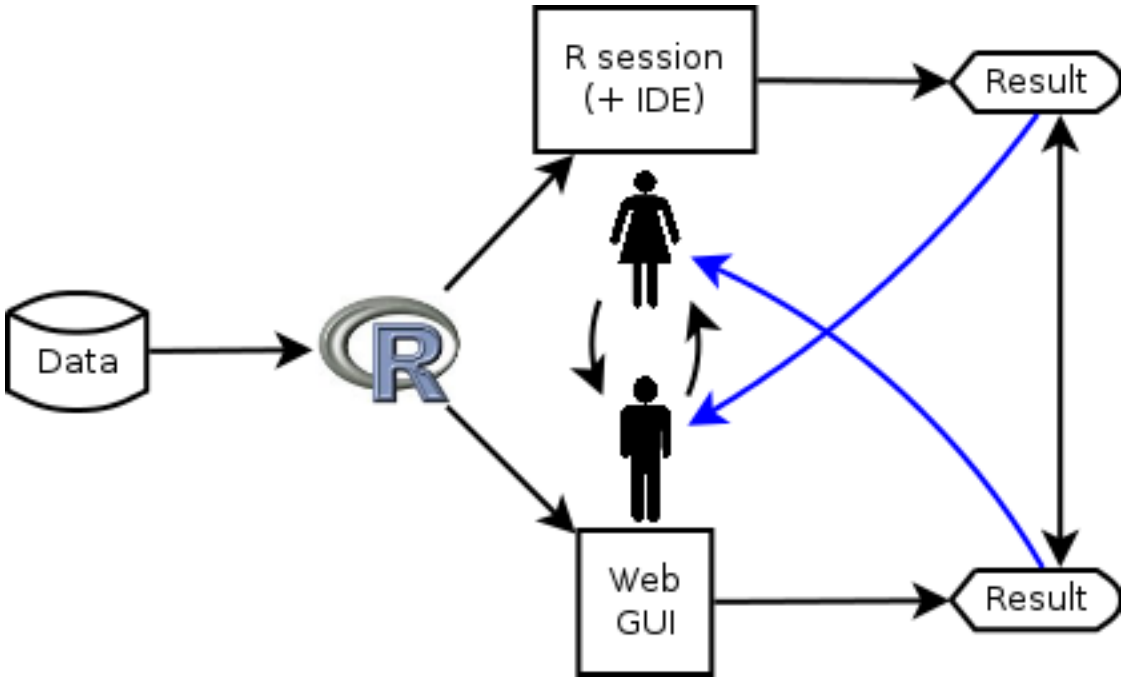
Collaborating across interfaces to data



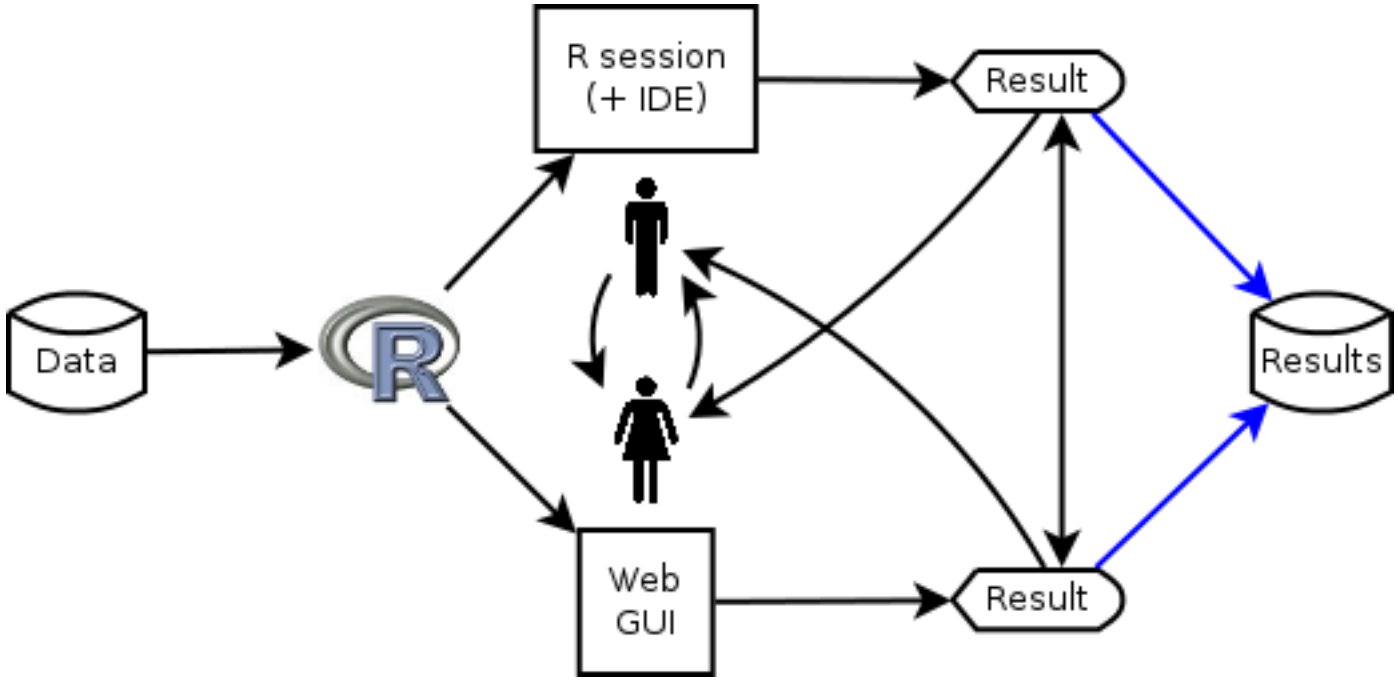
Comparing results



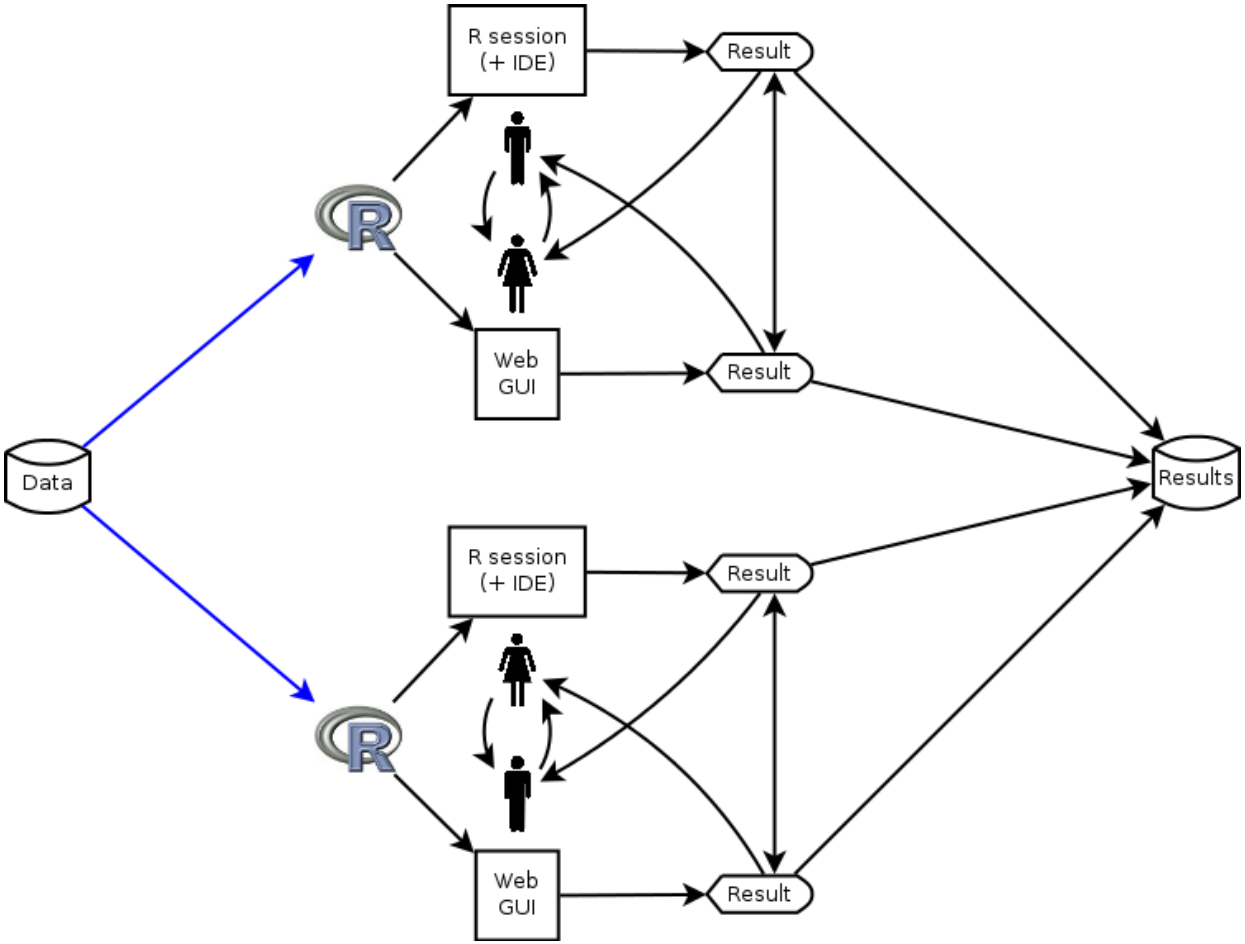
Collaborative iteration



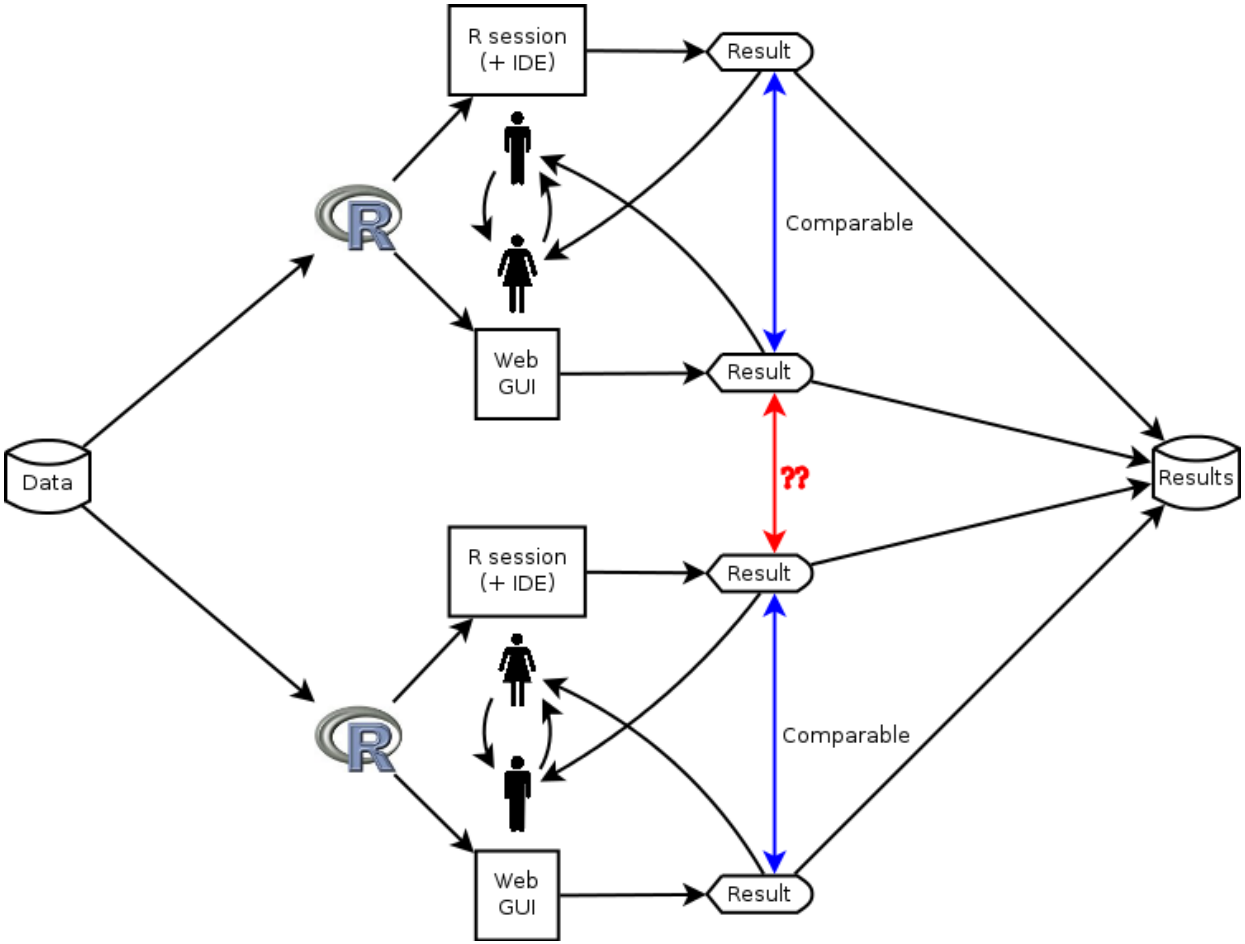
Results are *data*



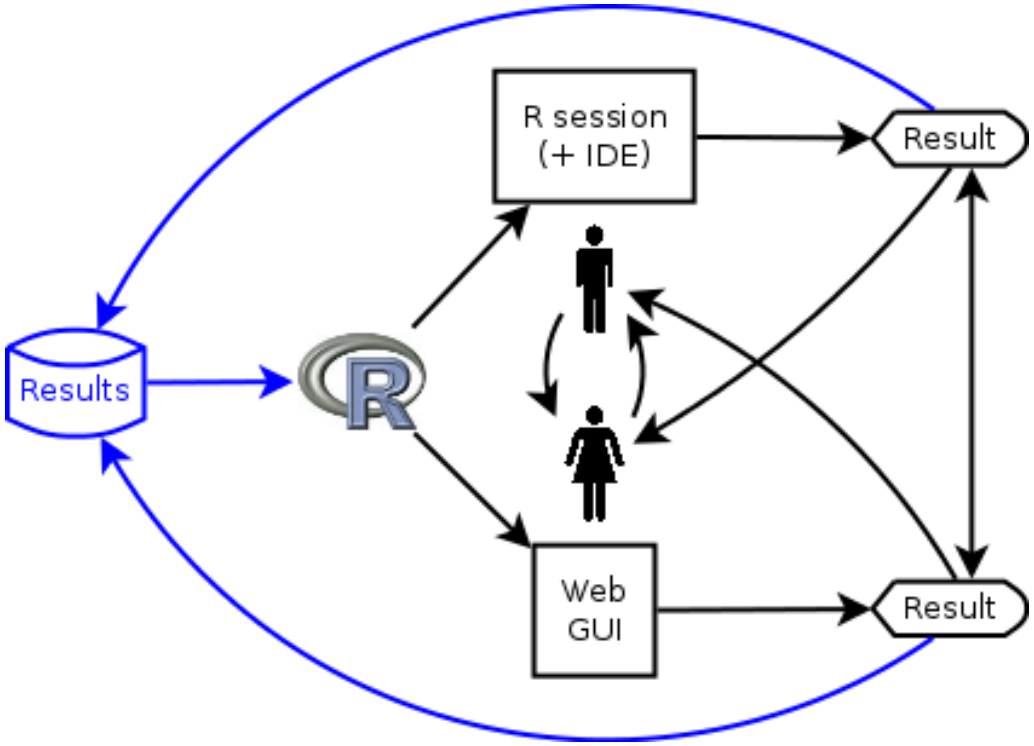
Differing needs



Comparing results redux



Results as *inputs*



Organization-level concerns

- Reproducibilty
  - Can we regenerate and confirm results?
- Compatibility
  - Is it safe/valid to compare and combine results?
- Discoverability
  - Can we discover and leverage existing results?
- Empowerment
  - Do scientists have the (computational) tools we need to answer our questions?

THE GENENTECH WAY

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

PROVIDING R ON THE CLUSTER

Projects have different needs

- Long running projects require stability
- Package development requires bleeding edge versions of dependencies
- Standard analyses should emphasize compatibility
- Custom analyses may requrie new/updated pkgs and methods

Empowerment ⬆
Reproducibility ⬇
Compatibility ⬇

Reproducibility ⬆
Compatibility ⬆
Empowerment ⬇

Empowerment ↑
Compatibility ↓
Reproducibility ↓

Compatibility ↑
Reproducibility ↑
Empowerment ↓

ONE R IS NOT ENOUGH

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

“Stable” R module

- Default R module
- New module every 6 months
  - Lagged after Bioconductor release
- Not updated in place
  - Narrow updates for absolutely critical bugfixes only
- Retained for reproducibility

“Current” R module

- Updated in-place nightly
  - CRAN, Bioc, and passing internal pkgs
- New module every 6 months
  - Not lagged
- Not retained

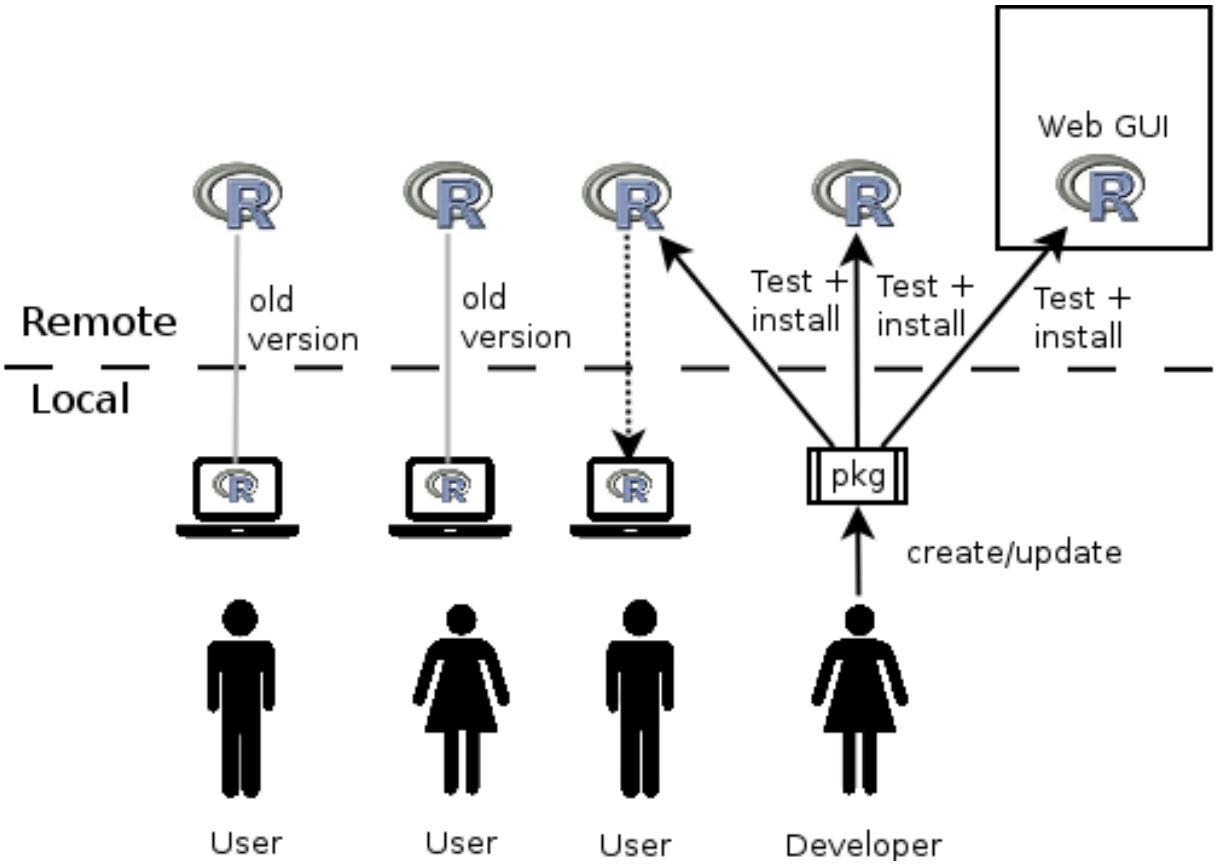
“Devel” R module

- Devel branch of bioconductor
- Updated in-place nightly
  - CRAN, Bioc, and passing internal pkgs
- New module every 6 months
  - Not lagged
- Not retained

Our analyses use our packages!

- Need tested versions of internal packages delivered to analysts hands
  - No SCM checkout on their part

Package testing



GRANBase

- Test cohorts of packages
  - Full build-install-check test coverage
  - Incremental – tests only run if package is updated
  - Tests run in appropriate R modules (current/devel)
- Build system installs passing pkgs to site libraries nightly

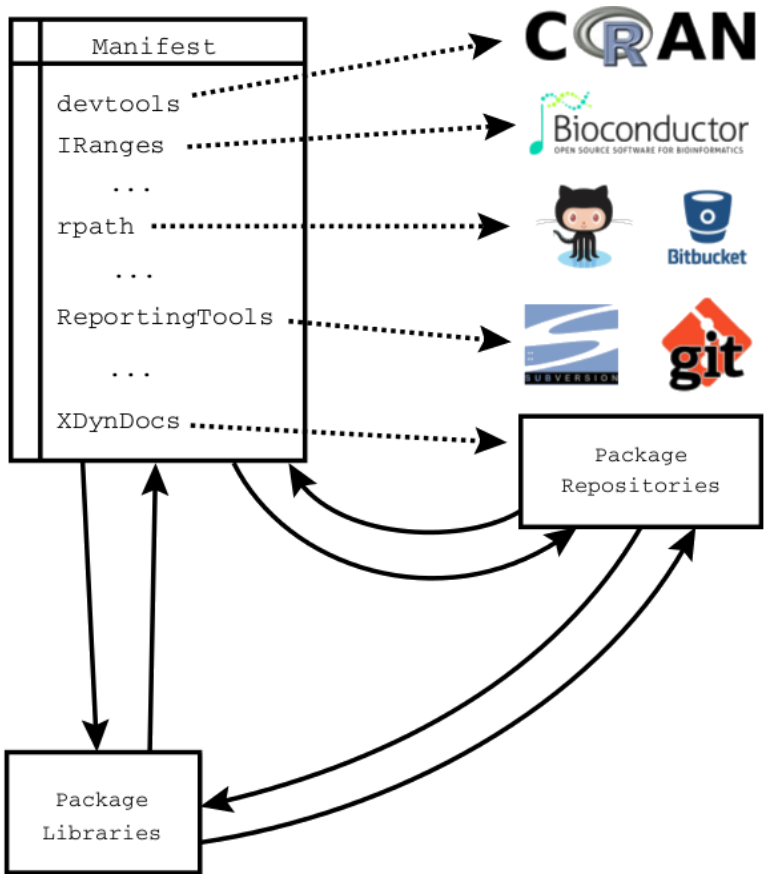
Best practices and lessons learned

- Automate which parts of CRAN you include when rolling a new R
  - Taskviews can provide a good baseline to build from
  - Check the list into SCM, automate adding and pulling from it
- User libraries
  - Allow but discourage
  - Ensure they are different for each R installation
- Provide a release candidate before cutting over default R version

R ENVIRONMENT RECREATION AND SANDBOXING

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

Dealing in package cohorts with switchr



- *Package cohorts are everywhere*
  - Repositories
  - Package libraries
  - SessionInfo
  - Package + dependencies

Switchr

- Install packages from repo and non-repo sources
  - Non-repo dependencies
- Manage and switch between multiple package libraries
  - Recreate/deploy pkg libraries
  - “Sandbox” R-based computations

DEPLOYING SHINY APPS TO THE CLUSTER

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

Shiny apps as packages

- Installable
- Self-describing
  - Dependencies
  - Title/description/authors
- Define software other code can use
- Testable
- *Can include arbitrary files*

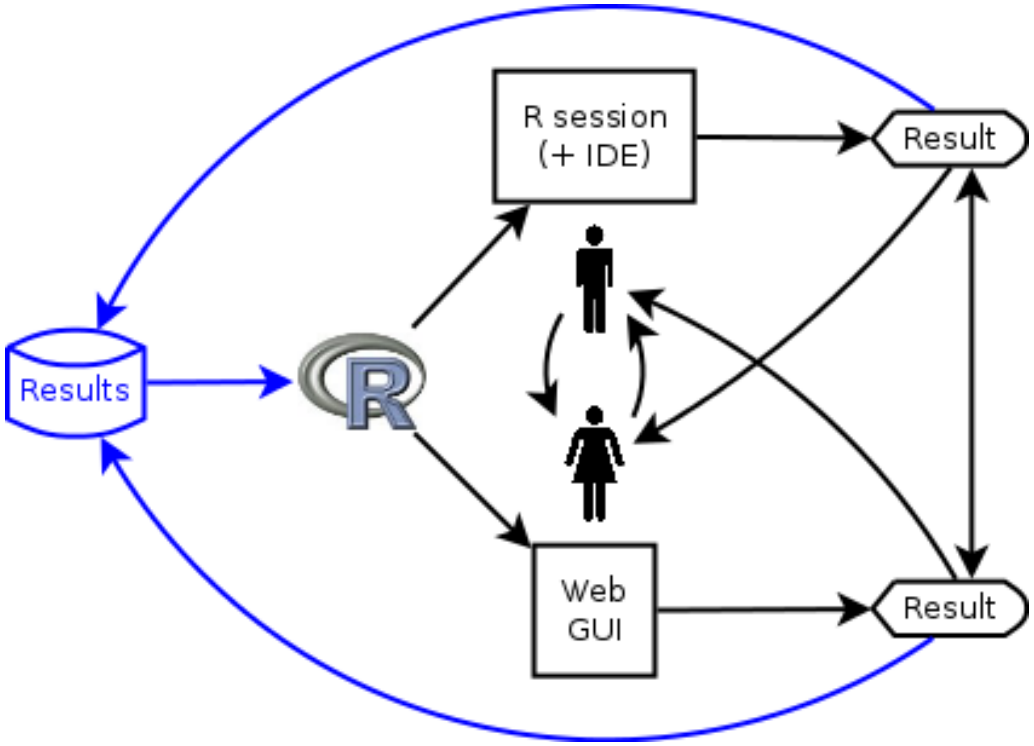
Shiny app deployment

- Provide template of deployable app package
- Switchr
  - Installation directly from SCM
  - Sandboxing
    - Created during installation
    - global.R invokes switchr to activate sandbox
- Structural testing of app pkg
- Symlink to “hot” Shiny-server Pro directory

DISCOVERABILITY

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

Discoverability of results

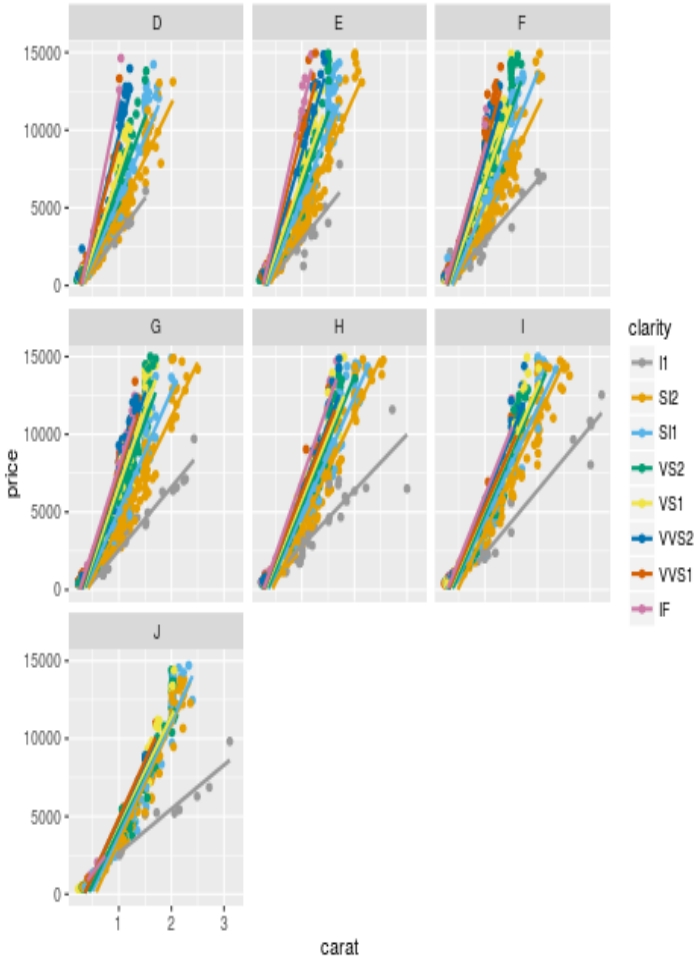


Trackr R package

- Automatically annotate and index results passed it
  - Focus on annotations useful for discovery or reproduction
    - Inferred descriptive info
    - Code
    - Dependencies/provenance

Live trackr demo

Selected captured metadata



FIELD	KNOWN INFO
geom.type	point, smooth
titles	null (bad Gabe!)
varlabels.x	carat
varlabels.y	price
varlabels.group.color	clarity
varlabels.group.panel	color
sessioninfo	<the sessionInfo>
code	<the code>
analysisfile	<path>/diamondplot.R
rstudioproject	<path>/useR2016

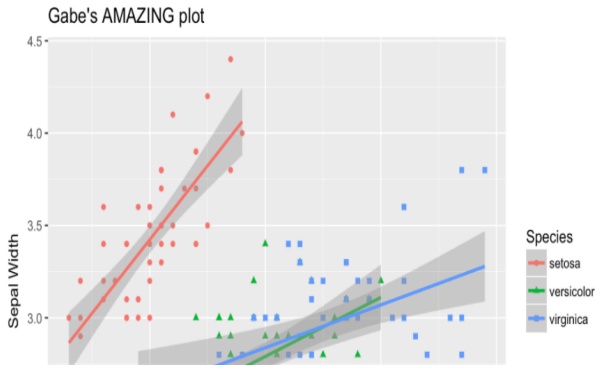
Selected captured metadata redux

Here we go

This is a test

Super special text.

```
library(ggplot2)
gg <- ggplot(data=iris, aes(Sepal.Length, y=Sepal.Width, color=Species)) +
  geom_point(aes(shape=Species), size=1.5) + geom_smooth(method="lm") +
  xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Gabe's AMAZING plot")
gg
```



FIELD	KNOWN INFO
chunks	<all text from Rmd file>
fullcode	<all code in Rmd file>
numplots	1
Rmdfile	demo.Rmd
Rmdfileid	<hash of demo.Rmd contents>
outfile	Demo.html
outputids	<id of the plot contained in report>
analysisfile	<path>/demo.R
rstudioproject	<path>/plotcon2017

Acknowledgements

- Michael Lawrence
- Sara Moore
- Cory Barr
- Robert Gentleman
- Matt Brauer
- B&CB
- Mango Solutions
- You

SLIDE GARAGE

EARL
ENTERPRISE APPLICATIONS OF THE R LANGUAGE

On the difference between flexibility and agility

- Flexibility is everyone getting to choose between 10 different R installations
- Agility doing nightly updates to the 1 R installation everyone uses

Containerization

- Benefits
  - If original work is done in containers, we get **Reproducibility** for free
  - Containers give us **Agility** and **Flexibility** to provide new software at whatever rate and granularity we want, maximizing **Empowerment**
- Challenges
  - Often need a *shared computing environment* to achieve **Compatibility**
    - Ability to do distributed computing on cluster with consistent computing environment on compute nodes
    - Shared file system
    - Shared/identical analysis platform (R, R packages, command line Bioinformatics tools)
  - Risk of *too much* **Flexibility**, destroying **Compatibility**

Collaboration

- Results from multiple sources must be **Compatible** so we can compare/combine them
  - Self-service analysis portals (e.g., shiny)
  - Local and/or remote work by multiple bioinformaticians
- Results must be **Discoverable** and **Reproducible** so that others can find and extend them

Packages

- Must **Empower** analysts to use new versions of internal and external packages
  - Painless use of appropriate versions, ideally with minimal manual action
- Packages need to be tested *as cohorts* to ensure they will work together

Flexibility  Unification

- **Flexibility** *empowers* analysts or projects to
  - Customize tools/ environment to the job
  - Set up things exactly how they like it

- **Unification** provides *shared computing environment*
  - Analysts can *compare*, *combine*, and *collaborate* on results
  - Centralized maintenance

Agility  Stability

- **Agility** provides *new/updated software*
  - Scientific methods and best practices evolve rapidly
  - Improvements over existing software/versions
- **Stability** *provides assurances* that
  - Code and applications will continue to run
  - Results will not unexpectedly change

Switchr

- Manage, describe, and recreate R package libraries
  - Ensure teammates are using same versions of R packages
  - Encapsulate analysis/shiny app with it's own package library
- Flexibly install packages
  - including specific historical versions
  - Retrieve and install non-repository dependencies when isntalling packages
    - i.e., github package depends on other github package
- On CRAN + github
  - https://github.com/gmbecker/switchr
- Paper preprint (accepted in JSS)
  - https://arxiv.org/abs/1501.02284

Switchr – installation

- Supports installation of packages from non-repo sources
  - Including non-repo dependencies (github pkg depending on github pkg)
    - **Without** modifying DESCRIPTION file
- Supports installation of exact, historical versions of CRAN and scm-tracked packages
- provides heuristics to determine correct dependency versions for old packages
  - Via Csardi's crandb

Switchr – sandboxing

- Manage multiple package libraries by name
  - `switchTo("mylib")`
  - Include or exclude site library (ie sandbox)