



 @ChasingMicrobes

Your Missing Step in Reproducible R Programming: Continuous Deployment

SIMPLY EXPLAINED

geek & poke



Chase Clark
PhD Candidate
Center for Biomolecular Sciences
Department of Pharmaceutical Sciences
University of Illinois at Chicago



Outline

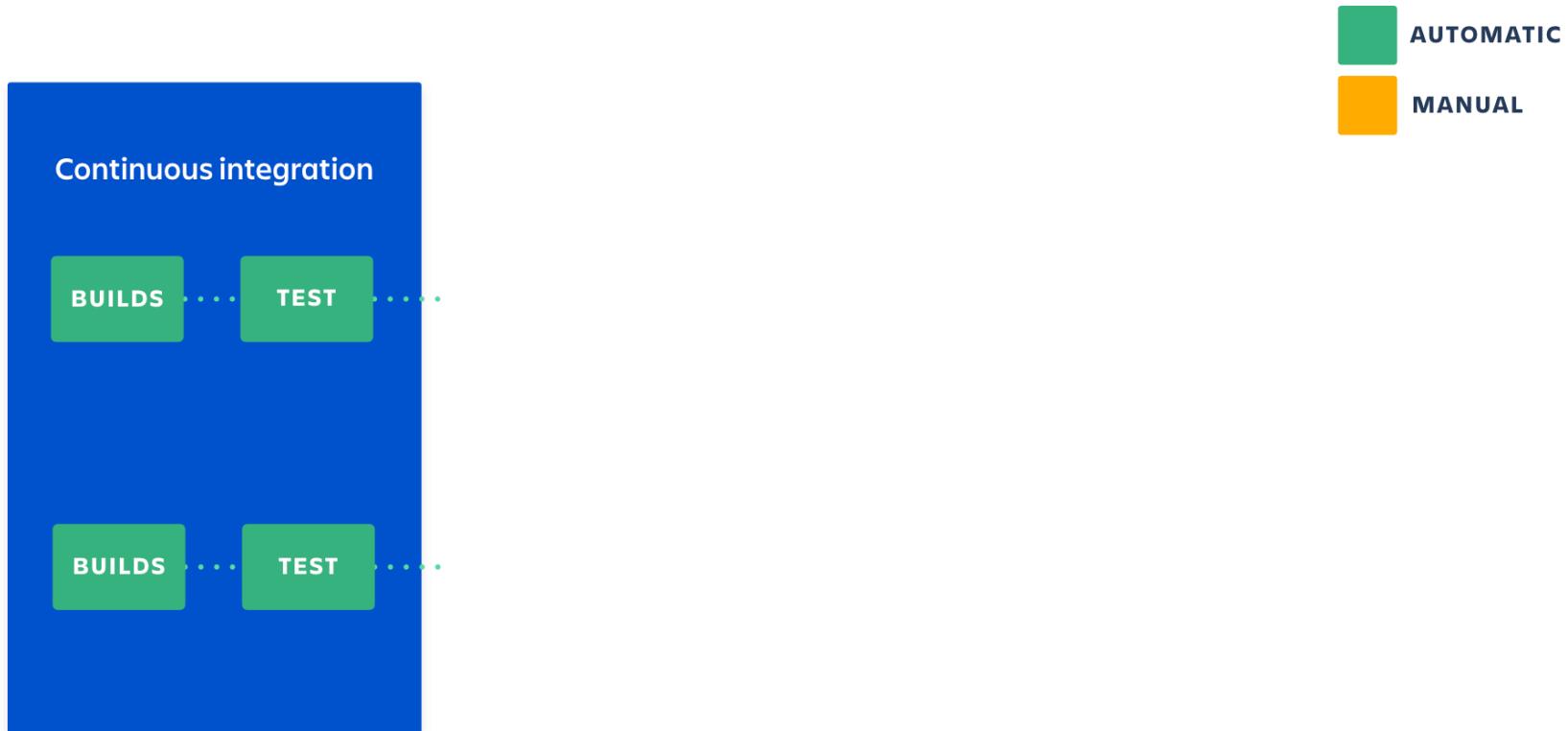
 @ChasingMicrobes

Not a tutorial but providing examples:

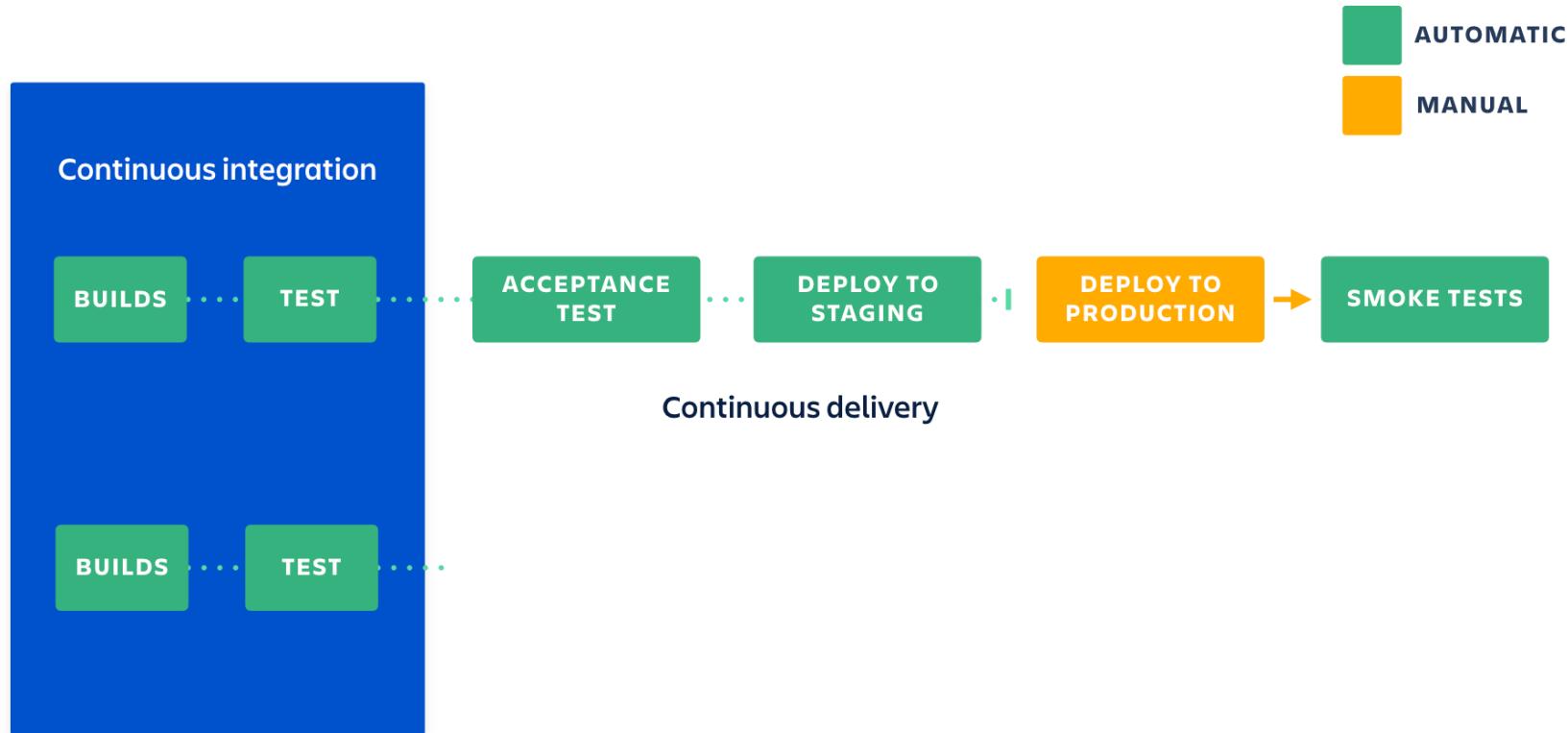
- General Overview of CD
- Automating Rmarkdown (brief)
- Automating Drake (brief)
- Automating Package Versioning (brief)
- Automating Shiny Software



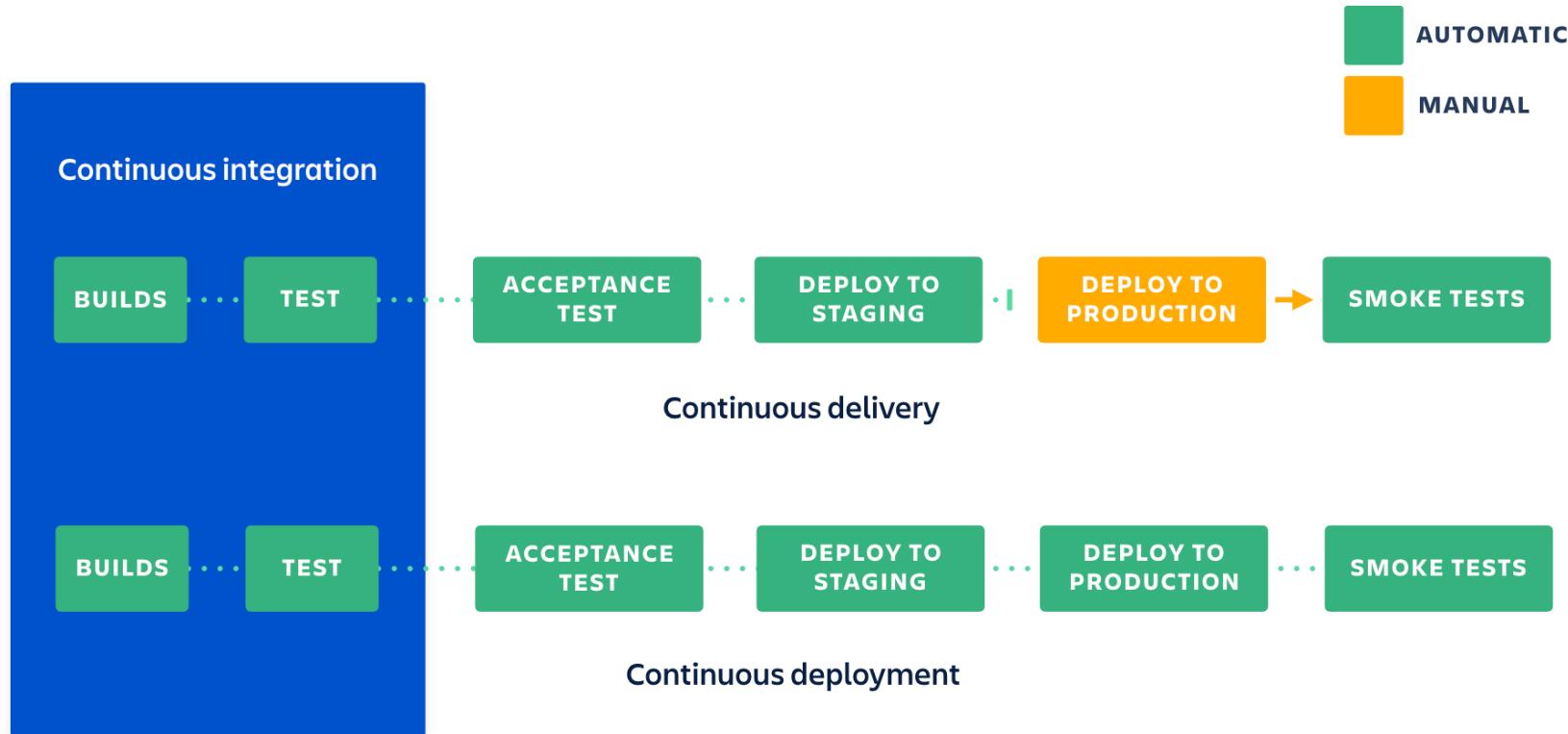
What is CI/CD?



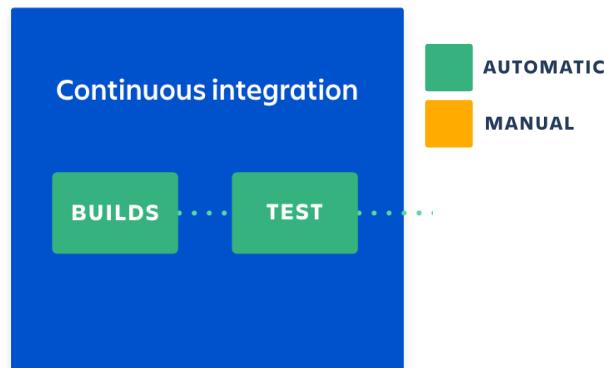
What is CI/CD?



What is CI/CD?



Who Uses Continuous Integration?



Who Uses Continuous Integration?

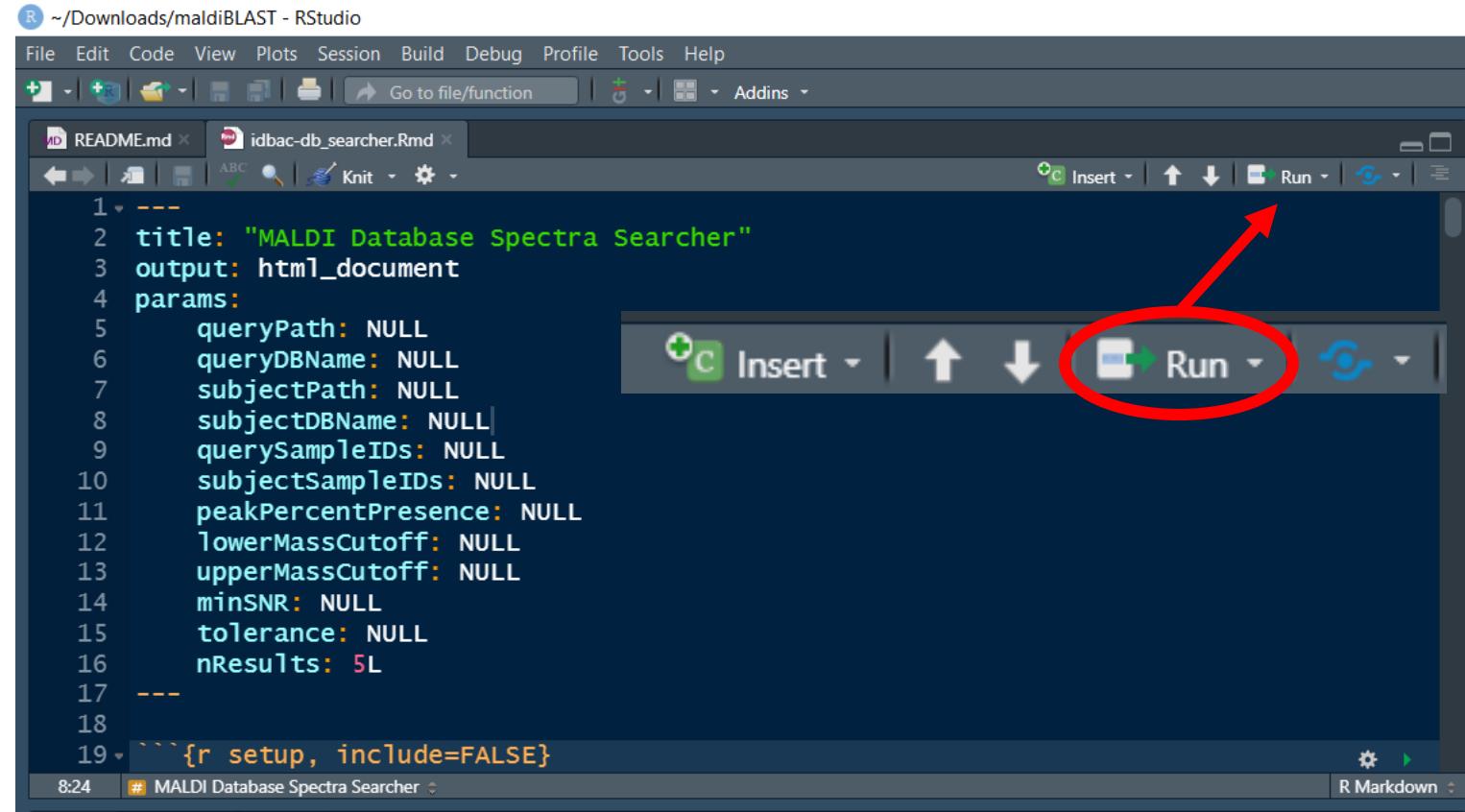
Who Uses Continuous Deployment?



Why Should We Care?

The goal is not reproducible code; it's reproducible results.

- Missing step... how do we know an output came from the latest version of code?



A screenshot of the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with various icons. The main workspace shows two files: README.md and idbac-db_searcher.Rmd. The idbac-db_searcher.Rmd file contains R Markdown code:

```
1 ---  
2 title: "MALDI Database Spectra Searcher"  
3 output: html_document  
4 params:  
5   queryPath: NULL  
6   queryDBName: NULL  
7   subjectPath: NULL  
8   subjectDBName: NULL  
9   querySampleIDs: NULL  
10  subjectSampleIDs: NULL  
11  peakPercentPresence: NULL  
12  lowerMassCutoff: NULL  
13  upperMassCutoff: NULL  
14  minSNR: NULL  
15  tolerance: NULL  
16  nResults: 5L  
17 ---  
18  
19 {r setup, include=FALSE}  
8:24 # MALDI Database Spectra Searcher R Markdown
```

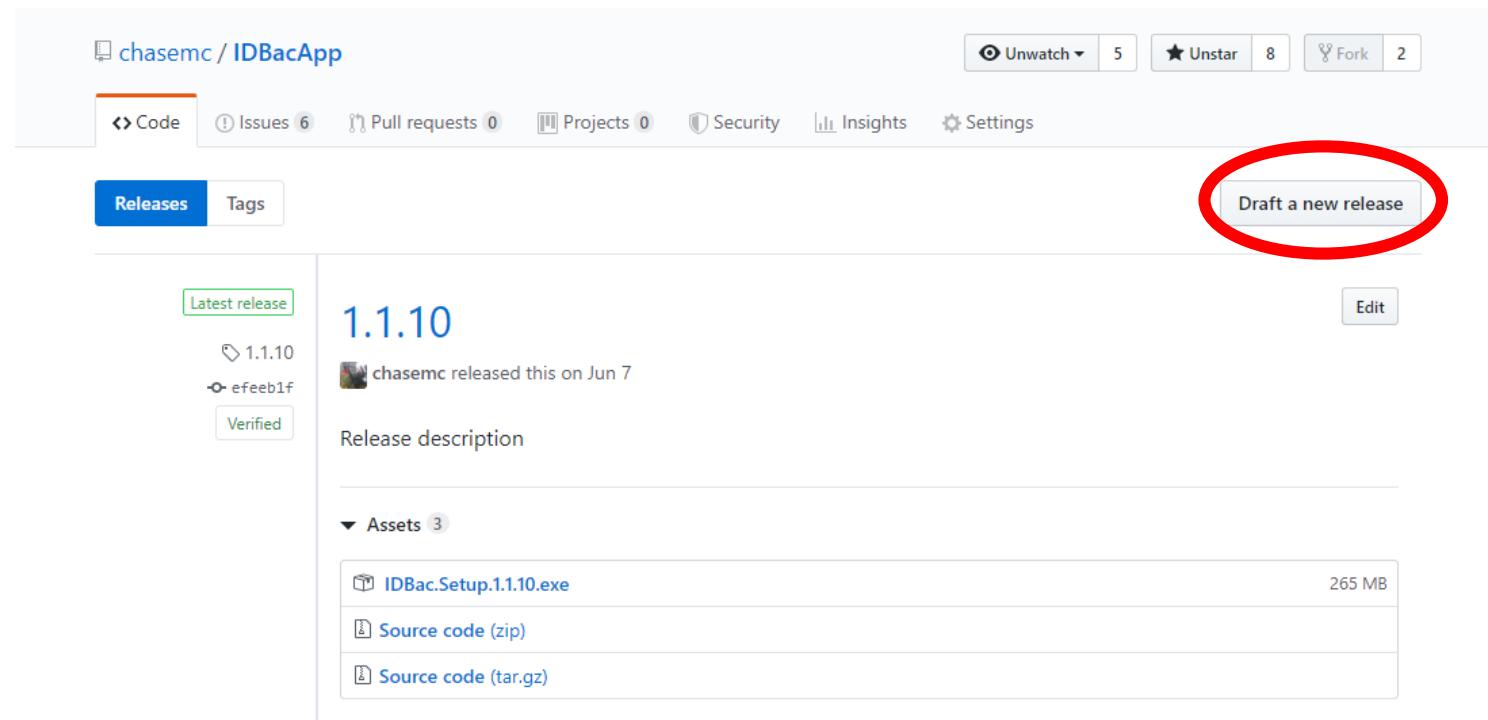
A red arrow points from the bottom right towards the 'Run' button in the toolbar, which is highlighted with a red circle. The 'Run' button icon consists of a blue square with a white arrow pointing right.



Why Should We Care?

The main goal is not reproducible code; it's reproducible results.

- Missing step... how do we know an output came from the latest version of code?



Why Should We Care?

The main goal is not reproducible code; it's reproducible results.

- Missing step... how do we know an output came from the latest version of code?

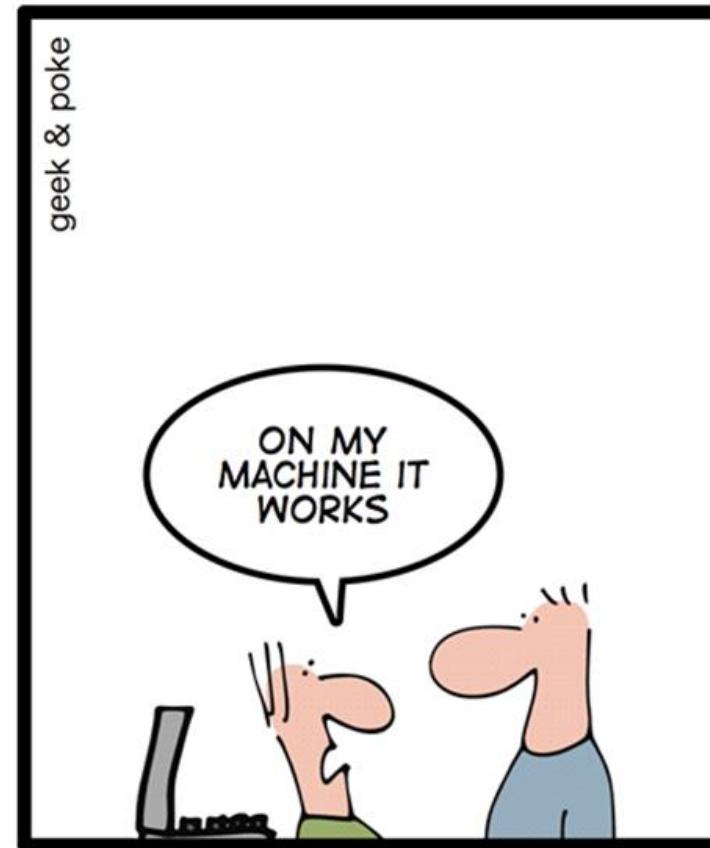
A screenshot of a GitHub repository page for 'chasemc / IDBacApp'. The top navigation bar shows 'Code' as the active tab, along with 'Issues 6', 'Pull requests 0', 'Projects 0', 'Security', 'Insights', and 'Settings'. The main content area is titled 'Releases' and shows a form for creating a new release. The form includes fields for 'Tag version' (set to 'master'), 'Release title', and a large text area for 'Describe this release'. Below these is a file upload section with the instruction 'Attach files by dragging & dropping, selecting or pasting them.' A red oval highlights a second file upload section below it with the instruction 'Attach binaries by dropping them here or selecting them.' To the right of the form, there is a sidebar with 'Tagging suggestions' (common practices for version names like v1.0 or v2.3.4) and 'Semantic versioning' (a link to a guide). At the bottom of the release form are 'Publish release' and 'Save draft' buttons.



Why Should We Care?

The main goal is not reproducible code; it's reproducible results.

- Missing step... how do we know an output came from the latest version of code?
- Consistency in deploying
 - R Packages
 - Reports/Graphs
 - Shiny Apps
 - Plumber API's



CI is Still Part of the Process

Testing

- Makes you think about your code
- Prevent releasing bugs
- Prevent errors now, in X years



```
context("test-binnr")
set.seed(42)

p <- c(MALDIquant::createMassPeaks(mass = c(2000, 3000, 4000),
                                      intensity = c(1, 1, 1)),
       MALDIquant::createMassPeaks(mass = c(2000, 3001, 4005),
                                   intensity = rep(1, 2, 3)),
       MALDIquant::createMassPeaks(mass = c(2000.01, 3002, 10000),
                                   intensity = rep(1, 2, 3)))

binned <- IDBacApp::peakBinner(peakList = p,
                                 ppm = 300,
                                 massStart = 1000,
                                 massEnd = 15000)

test_that("binnR works", {
  expect_known_hash(binned, "f5d2c83a3f")
  expect_warning(IDBacApp::peakBinner(peakList = list("A"),
                                       ppm = 100000))
  expect_identical(suppressWarnings(IDBacApp::peakBinner(peakList = list("A"),
                                                       ppm = 100000)),
                  list())
})
```

Make (nearly) Everything an R Package

- You can write a package!
- Nearly everything should be a package.
 - DESCRIPTION provides a summary of why the code exists, etc.
 - Handles dependencies
 - Allows for standardized relative paths (e.g. “inst/data/mytable.csv”)
 - Promotes documentation of code functions
 - Provides method of creating vignettes
 - Promotes unit testing, CRAN check, etc.
 - Etc.
- {devtools}, {usethis}, {attachment}, {golem}, {desc} all help with this and make the process nearly-trivial.



RStudio Connect Pricing

RStudio Connect is the place to publish all the work your teams create in R. Publish Shiny applications, R Markdown reports, APIs, dashboards, plots, and more to one convenient place with a click of a button in your RStudio IDE. Get your developers and publishers started with Connect Base or Standard. Share their work with Named User Packs for only \$5 per user per month.

Contact Sales for Named User volume pricing.

RStudio Connect
Base

\$14,995 per year

\$62 per user/month

RStudio Connect
Standard

\$24,995 per year

\$21 per user/month

RStudio Connect
Enterprise*

\$74,995 per year

\$6.25 per user/month

General Overview of (Free) CD for Open-Source

- The following can be setup as online or offline workflows.
- I depend heavily on GitHub, Appveyor, and Travis CI.
- There are many other options: Jenkins, Bamboo, etc.



Organize analysis as an R package



Output attached to GitHub release

Version Control with Git



Merge/push to Master/branch

Push to Dev-Branch

Continuous Deployment



Travis CI



Continuous Integration



Travis CI



Organize analysis as an R package



Output attached to GitHub release



Merge/push to Master/branch

Push to Dev-Branch

Continuous Integration
AppVeyor

Continuous Integration
Travis CI

Continuous Integration
AppVeyor

Organize analysis as an R package



Output attached to GitHub release



Merge/push to Master/branch

Push to Dev-Branch

Continuous Deployment



Travis CI



Continuous Integration



Travis CI



Organize analysis as an R package



Output attached to GitHub release



Merge/push to Master/branch

Push to Dev-Branch

Continuous Deployment



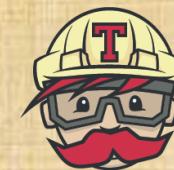
Travis CI



AppVeyor



Continuous Integration



Travis CI



AppVeyor

Organize analysis as an R package



Output attached to GitHub release



Merge/Push to Master-branch

Push to Dev-Branch

Continuous Deployment



Continuous Integration



Organize analysis as an R package



Output attached to GitHub release



Merge/Push to Master-branch

Push to Dev-Branch

Continuous Deployment



Continuous Integration



Organize analysis as an R package



Output attached to GitHub release



Merge/Push to Master-branch

Push to Dev-Branch

Continuous Deployment



Continuous Integration



Organize analysis as an R package



Output attached to GitHub release



Merge/Push to Master-branch

Push to Dev-Branch

Continuous Deployment



Travis CI

AppVeyor



Continuous Integration



Travis CI

AppVeyor



@ChasingMicrobes

Organize analysis as an R package



Output attached to GitHub release



Merge/Push to Master-branch

Push to Dev-Branch

Continuous Deployment



Travis CI

AppVeyor



Continuous Integration



Travis CI

AppVeyor

Outline

Not a tutorial but providing examples:

- ~~General Overview of CD~~
- Automating Rmarkdown (brief)
- Automating Drake (brief)
- Automating Package Versioning (brief)
- Automating Shiny Software



Reproducible Report

Chase Clark

August 12, 2019

```
## Built from commit: abc
```

Preferably use :: to specify namespace, rather than calling library(). This is will make it easier for others to understand what functions come from which package (and easier for you in 6-months). For some packages I still use library() because it's much eassier (eg ggplot2)

More info on the data here: <https://www.glerl.noaa.gov/metdata/chi/metadata.html>

Retrieve results using tested functions

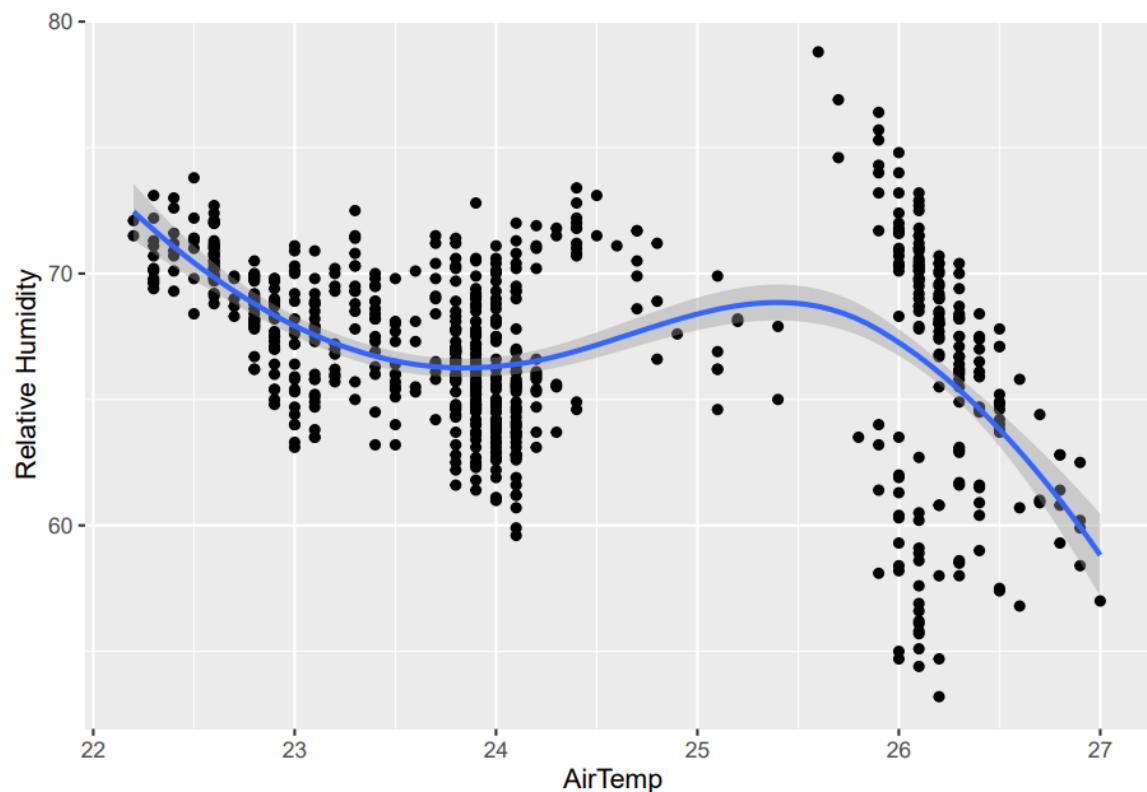
```
# Get url:  
  
yesterday_url <- autoReportR::urlGenerator()  
  
yesterday_data <- autoReportR::readNoaa(yesterday_url)
```

Quick peek:

```
##   ID Year DOY UTC      C m.s m.s.1 deg    X.  
## 1  4 2019 222 2356 24.1 6.7    7.0  94 62.6  
## 2  4 2019 222 2358 24.1 6.7    7.1  91 63.5  
## 3  4 2019 223     0 24.1 6.3    6.8  91 64.6  
## 4  4 2019 223     2 24.1 6.4    6.9  91 64.3  
## 5  4 2019 223     4 24.2 6.3    7.0  92 63.1  
## 6  4 2019 223     6 24.1 6.4    7.0  92 62.8
```

Simple plot

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



@ChasingMicrobes

The image shows two side-by-side browser windows. The left window is a GitHub release page for the repository "chasemc/autoReportR". It displays two releases: "1.0.1: Update .travis.yml" (the latest) and "1.0.0". The "1.0.1" release was made by "chasemc" 5 days ago, containing a commit "travis encrypt 2XXXXXXXXXX8 --pro" and three assets: "reproducible_report.pdf" (171 KB), "Source code (zip)", and "Source code (tar.gz)". The "1.0.0" release was made by "chasemc" 5 days ago, containing an "Initial commit" and two assets: "reproducible_report.pdf" (171 KB) and "Source code (zip)". The right window is a Travis CI build history for the same repository. It shows two builds: build #4 (passed) and build #3 (canceled). Build #4 was made by "Chase Clark" 4 days ago, taking 2 min 52 sec. Build #3 was made by "Chase Clark" 4 days ago, but was canceled. Both builds were triggered by commits "65d4aa5".

Releases - chasemc/autoReportR

GitHub, Inc. [US] | https://github.com/chasemc/autoReportR/releases

chasemc / autoReportR

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Community

Releases Tags

Draft a new release

Latest release

1.0.1 chasemc released this 5 days ago

travis encrypt 2XXXXXXXXXX8 --pro

Assets 3

- reproducible_report.pdf 171 KB
- Source code (zip)
- Source code (tar.gz)

1.0.0

chasemc released this 5 days ago · 1 commit to master since this release

Initial commit

Assets 2

Builds - chasemc/autoReportR

Travis CI

chasemc / autoReportR

build canceled

More options

Current Branches Build History Pull Requests

#4 passed

1.0.1

Update .travis.yml

Chase Clark

2 min 52 sec

4 days ago

#3 canceled

master

Update .travis.yml

Chase Clark

51 sec

4 days ago

Type here to search

8:17 AM 4/26/2019



Outline

Not a tutorial but providing examples:

- ~~General Overview of CD~~
- ~~Automating Rmarkdown (brief)~~
- Automating Drake (brief)
- Automating Package Versioning (brief)
- Automating Shiny Software





Drake:

- Launch the parts that changed since last time.
- Skip the rest.

History and provenance

As of version 7.5.2, `drake` tracks the history and provenance of your targets: what you built, when you built it, how you built it, the arguments you used in your function calls, and how to get the data back. (Disable with `make(history = FALSE)`)





Drake:

- Launch the parts that changed since last time.
- Skip the rest.

Consider automating `drake::make(...)` using CD on pushes/PR to master branch.

History and provenance

As of version 7.5.2, `drake` tracks the history and provenance of your targets: what you built, when you built it, how you built it, the arguments you used in your function calls, and how to get the data back. (Disable with `make(history = FALSE)`)



Outline

Not a tutorial but providing examples:

- ~~General Overview of CD~~
- ~~Automating Rmarkdown (brief)~~
- ~~Automating Drake (brief)~~
- Automating Package Versioning (brief)
- Automating Shiny Software



Have you ever messed up a package version number?

The screenshot shows a GitHub repository interface in RStudio. The left sidebar displays the repository structure, including branches (local: automation, bin, dev, displayNews, master; remote: origin, dev, displayNews, Fastjsonparse, master, newUpdateFunction, qc), pull requests (0), tags (0.0.15.2, 1.1.10, 1.0.100, 1.0.0), and submodules (0). The main area shows a commit history for the 'newUpdateFunction' branch. A specific commit, 'Version: 1.1.10', is highlighted. Two source code editor windows are open: one showing the DESCRIPTION file with the package metadata, and another showing the app_ui.R file containing R Markdown code. The bottom right corner features a commit message editor.

repository IDBacApp > branch newUpdateFunction

Viewing 17/17 Show All

Filter (Ctrl + Alt + f)

LOCAL 7/7

- automation
- bin
- dev
- displayNews
- master
- newUpdateFunction**
- qc

REMOTE 6/6

- origin
- dev
- displayNews
- Fastjsonparse
- master
- newUpdateFunction**
- qc

PULL REQUESTS 0

TAGS 4/4

- 0.0.15.2
- 1.1.10
- 1.0.100
- 1.0.0

SUBMODULES 0

branch newUpdateFunction

undo redo pull push branch stash pop boards

R ~/GitHub/IDBacApp - newUpdateFunction - RStudio Source Editor

DESCRIPTION

```
1 Package: IDBacApp
2 Type: Package
3 Title: Protein and small molecule analysis of bacteria by MALDI-TOF MS
4 Version: 1.1.10
5 Author: Chase Clark
6 Maintainer: Chase Clark <cclark42@uic.edu>
7 Description: This app allows users to take raw dat MALDI-TOF MS files (Tested on Bruker autoFlex)
8 License: GPL-3
9 Encoding: UTF-8
10 LazyData: true
11 biocviews:
12 Depends:
13   dendextend (>= 1.12.0),
14   shiny (>= 1.3.2)
```

R ~/GitHub/IDBacApp - newUpdateFunction - RStudio Source Editor

app_ui.R

```
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
```

Staged Files (2)

- R 1
- log.txt

Stage all changes

Unstaged Files (0)

Amend

Commit Message

Summary

Description

Stage files/changes to commit

Auto-Bump Package Version Number

- Read version from DESCRIPTION
- Increment version
- Update DESCRIPTION
- Commit/Push change

```
1  Package: desc
2  Title: Manipulate DESCRIPTION Files
3  Version: 1.2.0
4  Authors@R:
5    c(person(given = "Gábor",
6              family = "Csárdi",
7              role = c("aut", "cre"),
8              email = "csardi.gabor@gmail.com"),
9    person(given = "Kirill",
10           family = "Müller",
11           role = "aut"),
12    person(given = "Jim",
13           family = "Hester",
14           role = "aut",
15           email = "james.f.hester@gmail.com"),
16    person(given = "Maëlle",
17           family = "Salmon",
18           role = "ctb",
19           comment = c(ORCID = "0000-0002-2815-0399")))
20  Maintainer: Gábor Csárdi <csardi.gabor@gmail.com>
21  Description: Tools to read, write, create, and manipulate DESCRIPTION files.
22  It is intended for packages that create or manipulate other packages.
```



Auto-Bump Package Version Number

- Read version from DESCRIPTION
- Increment version
- Update DESCRIPTION
- Commit/Push change

```
1  Package: desc
2  Title: Manipulate DESCRIPTION Files
3  Version: 1.2.0
4  Authors@R:
5    c(person(given = "Gábor",
6              family = "Csárdi",
7              role = c("aut", "cre"),
8              email = "csardi.gabor@gmail.com"),
9    person(given = "Kirill",
10           family = "Müller",
11           role = "aut"),
12    person(given = "Jim",
13           family = "Hester",
14           role = "aut",
15           email = "james.f.hester@gmail.com"),
16    person(given = "Maëlle",
17           family = "Salmon",
18           role = "ctb",
19           comment = c(ORCID = "0000-0002-2815-0399")))
20  Maintainer: Gábor Csárdi <csardi.gabor@gmail.com>
21  Description: Tools to read, write, create, and manipulate DESCRIPTION files.
22  It is intended for packages that create or manipulate other packages.
```

Infinite Loop



Auto-Bump Package Version Number



```
before_test:
  - git checkout master
  - Rscript -e "desc::desc_bump_version('patch',
    list.files(pattern = 'DESCRIPTION',
    ignore.case = T,
    recursive = F))"
  - Rscript -e "writeChar(as.character(desc::desc_get_version(list.files(pattern = 'DESCRIPTION',
    ignore.case = T,
    recursive = F))),
    'version_temp.txt')"
  - ps: $env:appVersion = Get-Content version_temp.txt
  - git config --global credential.helper store
  - ps: Add-Content "$HOME\.git-credentials" "https://$(($env:access_token)):x-oauth-basic@github.com`n"
  - git config --global user.email "cclark42@uic.edu"
  - git config --global user.name "Chase Clark"
  - git add DESCRIPTION
  - git commit -m "AppVeyor version bump; [skip appveyor]"
  - git push origin master
```



Outline

Not a tutorial but providing examples:

- ~~General Overview of CD~~
- ~~Automating Rmarkdown (brief)~~
- ~~Automating Drake (brief)~~
- ~~Automating Package Versioning (brief)~~
- Automating Shiny Software



CD and Shiny

- Requires an opinionated way to structure an app for production:
 - Mostly the same as ThinkR's {golem}
 - Structure Shiny app as a package



CD and Shiny

- Requires an opinionated way to structure an app for production:
 - Mostly the same as ThinkR's {golem}
 - Structure Shiny app as a package
- New R package for deploying Shiny apps as Electron applications:
 - github.com/chasemc/electricShine



CD and Shiny

- Requires an opinionated way to structure an app for production:
 - Mostly the same as ThinkR's {golem}
 - Structure Shiny app as a package
- New R package for deploying Shiny apps as Electron applications:
 - github.com/chasemc/electricShine
- How to get started/minimal examples of combining Shiny and CD:
 - github.com/chasemc/demoApp



CD and Shiny

- Requires an opinionated way to structure an app for production:
 - Mostly the same as ThinkR's {golem}
 - Structure Shiny app as a package
- New R package for deploying Shiny apps as Electron applications:
 - github.com/chasemc/electricShine
- How to get started/minimal examples of combining Shiny and CD:
 - github.com/chasemc/demoApp
- How this looks in practice:
 - github.com/chasemc/IDBacApp



Using Shiny To Profile Bacteria and Find New Drug Leads



ANTIBIOTIC HUNTERS

Words & Photos by Jennifer Yang

<http://projects.thestar.com/antibiotics-resistance-and-the-race-for-new-bacteria/#one>



@ChasingMicrobes

Pharmacognosy

The study of the physical, chemical, biochemical and biological properties of drugs, drug substances or potential drugs of natural origin as well as the search for new drugs from natural sources.

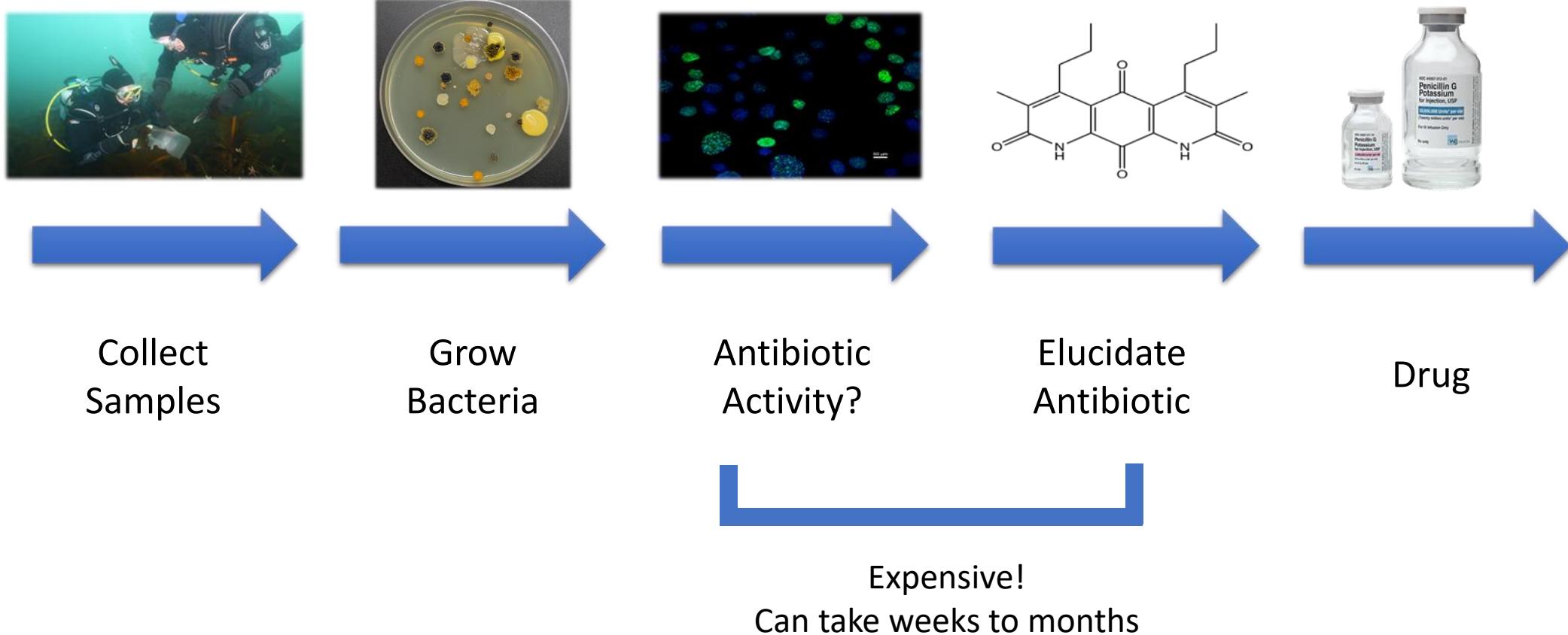


Pharmacognosy

The study of the physical, chemical, biochemical and biological properties of drugs, drug substances or potential drugs of natural origin as well as the search for new drugs from natural sources.



Discovery Pipeline (Simplified)



We didn't have an easy
way to prioritize bacteria





IDBac

A MALDI Protein and Small Molecule Bioinformatics Platform

GITHUB

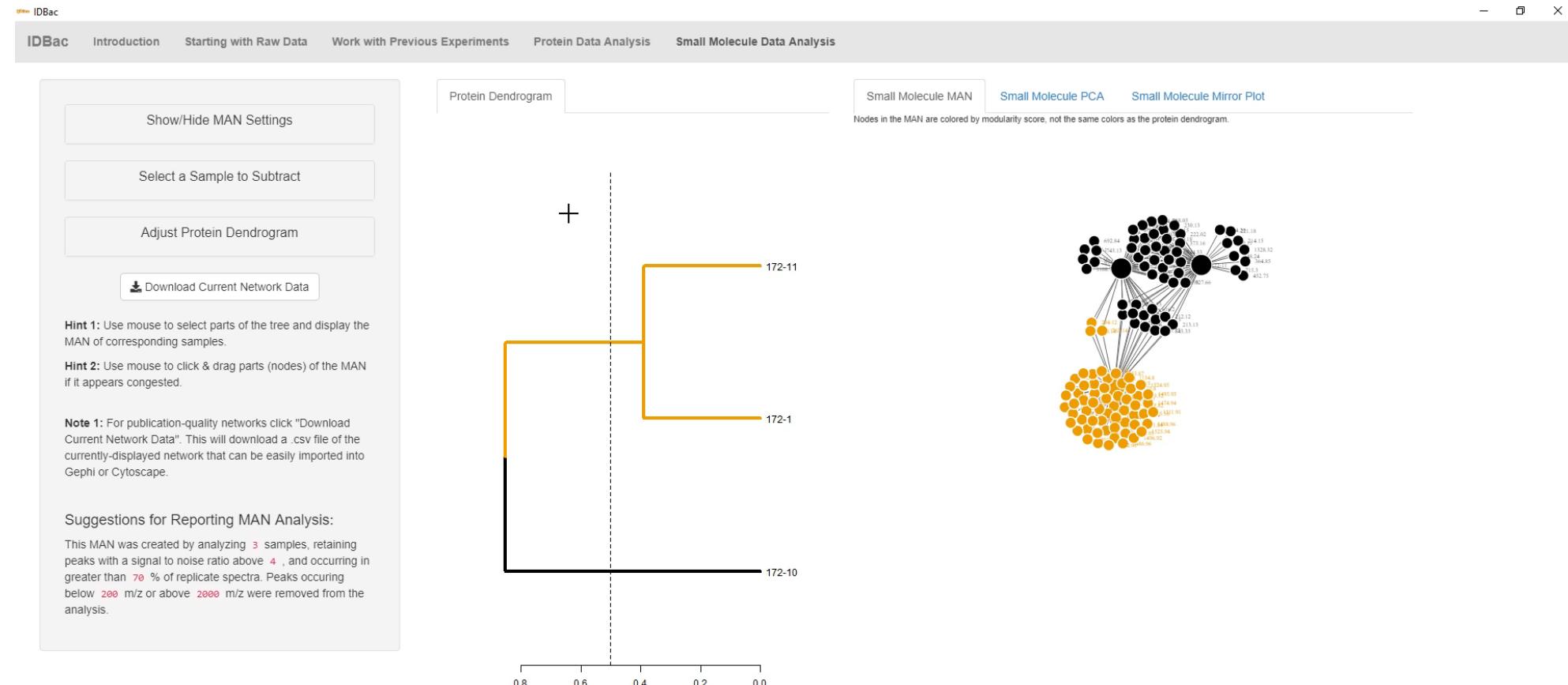
TWITTER



@ChasingMicrobes



Interactively Work with Disparate Data Sets



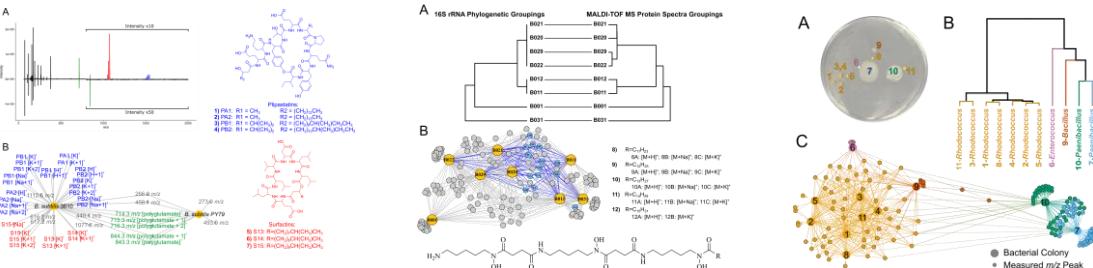
@ChasingMicrobes

Coupling MALDI-TOF mass spectrometry protein and specialized metabolite analyses to rapidly discriminate bacterial function

Chase M. Clark^{a,1}, Maria S. Costa^{b,1}, Laura M. Sanchez^{a,2}, and Brian T. Murphy^{a,2,3}

^aDepartment of Medicinal Chemistry and Pharmacognosy, College of Pharmacy, University of Illinois at Chicago, Chicago, IL; and ^bFaculty of Pharmaceutical Sciences, University of Iceland, Hagi, IS-107 Reykjavik, Iceland

Edited by Jerryold Meinwald, Cornell University, Ithaca, NY, and approved April 5, 2018 (received for review January 22, 2018)



<https://doi.org/10.1073/pnas.1801247115>

Detailed description: The screenshot shows a video player with a play button at 0:19, a progress bar at 0:29, and a 1x speed indicator. The video title is 'Using the Open-Source MALDI TOF-MS IDBac Pipeline for Analysis of Microbial Protein and Specialized Metabolite Data'. Below the video are sections for 'CITE THIS', 'SHARE', and 'CHAPTERS'. The 'CHAPTERS' section lists 14 numbered steps from 'Title' to 'Detailed Analysis of Microbial Protein and Specialized Metabolite Data'.

<https://www.jove.com/video/59219/using-open-source-maldi-tof-ms-idbac-pipeline-for-analysis-microbial>

JOURNAL OF NATURAL PRODUCTS

Cite This: J. Nat. Prod. XXXX, XXX, XXX–XXX

pubs.acs.org/jnp

Minimizing Taxonomic and Natural Product Redundancy in Microbial Libraries Using MALDI-TOF MS and the Bioinformatics Pipeline IDBac

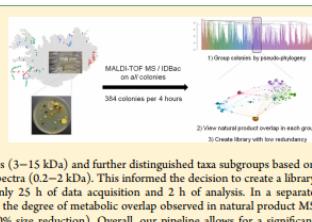
Maria S. Costa,^{†,‡,§} Chase M. Clark,^{‡,§} Sessela Ómarsdóttir,[†] Laura M. Sanchez,^{‡,§} and Brian T. Murphy^{‡,§,¶}

[†]Faculty of Pharmaceutical Sciences, University of Iceland, Hagi, Hofsvallagata 53, IS-107 Reykjavik, Iceland

[‡]Department of Pharmaceutical Sciences, College of Pharmacy, University of Illinois at Chicago, 833 South Wood Street (MC 781), Room 539, Chicago, Illinois 60607, United States

Supporting Information

ABSTRACT: Libraries of microorganisms have been a cornerstone of drug discovery efforts since the mid-1950s, but strain duplication in some libraries has resulted in unwanted natural product redundancy. In the current study, we implemented a workflow that minimizes both the natural product overlap and the total number of bacterial isolates in a library. Using a collection expedition to Iceland as an example, we purified very distinct bacterial colony off isolation plates derived from 86 environmental samples. We employed our mass spectrometry (MS)-based IDBac workflow on these isolates to form groups of taxa based on protein MS fingerprints (3–15 kDa) and further distinguished tax subgroups based on their degree of overlap within corresponding natural product spectra (0.2–2 kDa). This informed the decision to create a library of 301 isolates spanning 54 genera. This process required only 25 h of data acquisition and 2 h of analysis. In a separate experiment, we reduced the size of an existing library based on the degree of metabolic overlap observed in natural product MS spectra of bacterial colonies (from 833 to 233 isolates, a 72.0% size reduction). Overall, our pipeline allows for a significant reduction in costs associated with library generation and minimizes natural product redundancy entering into downstream biological screening efforts.



<https://pubs.acs.org/doi/pdf/10.1021/acs.jnatprod.9b00168?rand=dnd7220z>



Some Shiny Apps Work With Large Quantities of User-Input Data

MassIVE Dataset Summary + C massive.ucsd.edu/ProteoSAFe/dataset.jsp?task=ad311cd913c347578f1f1c29fabcea83

Computer Science and Engineering
University of California, San Diego

Center for Computational Mass Spectrometry

MassIVE Datasets | General Info | UCSD Proteomics | Future Tools | Demo | Contact

User: chasemc Pass: Sign in
Don't have an account? [Register!](#)

MassIVE MSV000083461

Partial Public

A new paradigm to maximize taxonomic and natural product space in microbial libraries using MALDI-TOF MS/IDBac.

Subscribe Comment Reanalyze Spectra Add Reanalysis

Description

This bacterial MALDI-TOF dataset was used for the study "A new paradigm to maximize taxonomic and natural product space in microbial libraries using MALDI-TOF MS/IDBac". It consists of triplicate MALDI-TOF MS profiles (both reflectron and linear datasets) for >1,600 bacteria cultivated from a collection in Iceland. [doi:10.25345/C5261K] [dataset license: CC0 1.0 Universal (CC0 1.0)]

Keywords: MALDI ; Bacteria ; IDBac

Contact

Principal Investigators: Brian Murphy, University of Illinois at Chicago, United States
Laura Sanchez, University of Illinois at Chicago, United States

Submitting User: chasemc

Number of Files: 9,705
Total Size: 33.83 GB
Spectra: 0
Subscribers: 0

Owner	Reanalyses
Protein Accessions (reported)	0
Peptides:	0
Variant Peptides:	0
PSMs:	0

[FTP Download](#)

FTP Download Link (click to copy):
<ftp://massive.ucsd.edu/MSV000083461/>

JOURNAL OF
NATURAL
PRODUCTS

Cite This: J. Nat. Prod. XXXX, XXX, XXX-XXX
pubs.acs.org/jnp

Article

Minimizing Taxonomic and Natural Product Redundancy in Microbial Libraries Using MALDI-TOF MS and the Bioinformatics Pipeline IDBac

Maria S. Costa,^{†,‡,§,¶} Chase M. Clark,^{§,¶} Sesselja Ómarsdóttir,[†] Laura M. Sanchez,^{§,¶} and Brian T. Murphy,^{*‡,§,¶}

[†]Faculty of Pharmaceutical Sciences, University of Iceland, Hagi, Höfsvallagata 53, IS-107 Reykjavík, Iceland
[‡]Department of Pharmaceutical Sciences, College of Pharmacy, University of Illinois at Chicago, 833 South Wood Street (MC 781), Room 539, Chicago, Illinois 60607, United States
[§]Supporting Information

ABSTRACT: Libraries of microorganisms have been a cornerstone of drug discovery efforts since the mid-1950s, but strain duplication in some libraries has resulted in unwanted natural product redundancy. In the current study, we implemented a workflow that minimizes both the natural product overlap and the total number of bacterial isolates in a library. Using a collection expedition to Iceland as an example, we purified every distinct bacterial colony off isolation plates derived from 80 environmental samples. We employed our mass spectrometry (MS)-based IDBac workflow on these isolates to form groups of taxa based on protein MS fingerprints (3–15 kDa) and further distinguished taxa subgroups based on their degree of overlap within corresponding natural product spectra (0.2–2 kDa). This informed the decision to create a library of 301 isolates spanning 54 genera. This process required only 25 h of data acquisition and 2 h of analysis. In a separate experiment, we reduced the size of an existing library based on the degree of metabolic overlap observed in natural product MS spectra of bacterial colonies (from 833 to 233 isolates, a 72.0% size reduction). Overall, our pipeline allows for a significant reduction in costs associated with library generation and minimizes natural product redundancy entering into downstream biological screening efforts.

(1) MALDI-TOF MS / IDBac on all colonies
(2) View natural product overlap in each group
(3) Create library with low redundancy





Recycle Bin



SpaceSniffer

Docker
Desktop

Python_3-7



RStudio



IDBac

How do users interact with your results?

- Technical users (use R)
- Technical users (don't use R)
- Non-technical users (definitely don't use R)





Recycle Bin



SpaceSniffer



Docker
Desktop



Python_3-7



RStudio



IDBac



Type here to search

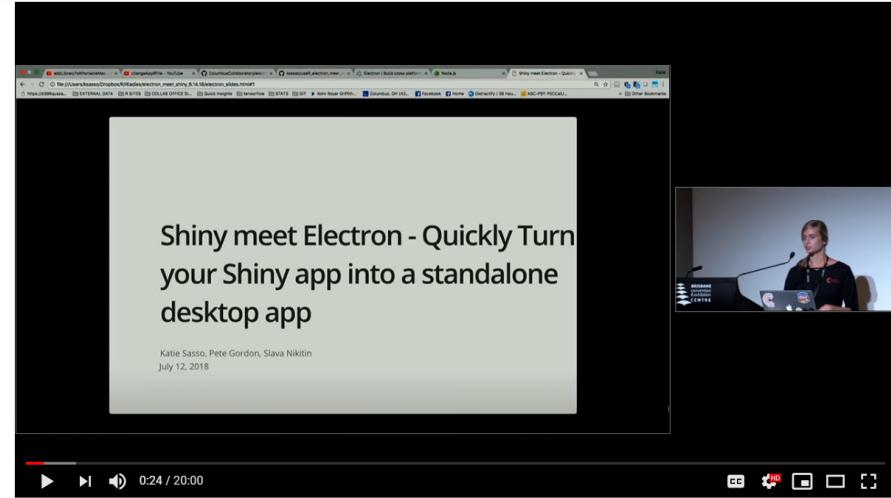
12:10 PM
3/25/2019

Creating/Distributing Local Shiny Applications



github.com/ficonsulting/RInno

Only Windows
Can be Complicated
Not Self-Sufficient



Shiny meets Electron: Turn your Shiny app into a standalone desktop app in no time

4,057 views

81 likes 1 dislike SHARE SAVE ...

<https://github.com/ColumbusCollaboratory/electron-quick-start>
<https://github.com/ColumbusCollaboratory/photon>

Only preliminary template- still manual work to make app
Relies on Rportable- no provenance, Chinese characters



@ChasingMicrobes



electricShine R Package

Windows CI:

- build passing

Mac and Linux CI

- build passing

Is easy! One meta-function:

```
electricShine::buildElectricApp(  
  app_name = "My_App",  
  description = "My demo application",  
  package_name = "demoApp",  
  semantic_version = "1.0.0",  
  build_path = buildPath,  
  mran_date = MRANdate,  
  function_name = "run_app",  
  github_repo = "chasemc/demoApp",  
  local_path = NULL  
)
```

Self-Sufficient

The electricShine package:

- Installs nodejs
- Sets up Electron build
- Compiles your app
- When complete, will work for Windows/Mac/Linux
- Compatible with continuous-deployment

An electricShine app contains:

- R- from specified MRAN date
- R packages- from specified MRAN date
- Your Shiny app!

Create distributable Shiny Electro X +

← → C chasemc.github.io/electricShine/ 🔍 ⭐ 🌐 ABP C ⋮

electricShine 0.0.0.9000  Reference Articles ▾

electricShine

Purpose

Sometimes an R Shiny app is too resource-intensive, or otherwise restricted, to be deployed into the cloud. Along with this, it can be non-trivial for someone inexperienced with R, or programming in general, to install R and open your Shiny app.

For these reasons it is desireable to be able to create a Shiny app that can be opened like a “regular” computer application, preferably from a Desktop shortcut. This is the purpose of `{electricShine}` and what it will do with your Shiny app.

High-Level Overview

`{electricShine}` is based on [Electron](#) which is a well-used and supported framework for creating desktop applications, usually using just javascript, html and css.

Repeatability of creating these desktop apps is a priority of `{electricShine}`, and to help with this it installs both R and R packages from a single MRAN date.

It currently only builds windows apps, but I’m investigating adding support for Mac and Linux as well. That is quite a bit harder because the R installation for Mac and Linux hard-codes paths into the installation and part of the benefit of `{electricShine}` is not relying on the system’s version of R <https://community.rstudio.com/t/is-r-on-mac-portable/36642/8>



Links

Browse source code at
<https://github.com/chasemc/electricShine>

Report a bug at
<https://github.com/chasemc/electricShine/issues>

License

[Full license](#)

[MIT + file LICENSE](#)

Developers

Chase Clark
Author, maintainer 

[All authors...](#)

Dev status

 lifecycle experimental

 build passing

 build passing

```
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\appveyor> C:/R/bin/R.exe

R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> .libPaths("C:/RLibrary")
> -
```



Windows Server 2012 R2

FROM THE LAB

SCIENCE
MUSEUM

Funding:

National Center For
Complementary & Integrative
Health of the National Institutes
of Health Award Number:
F31AT010419

 NATIONAL
GEOGRAPHIC
Grant CP-044R-17

Murphy Lab Current Grad and Post-Doc Members:

Dr. Brian Murphy

Chase Clark

Antonio Hernandez

Dr. Jeongho Lee

Dr. Tuan Anh Tran

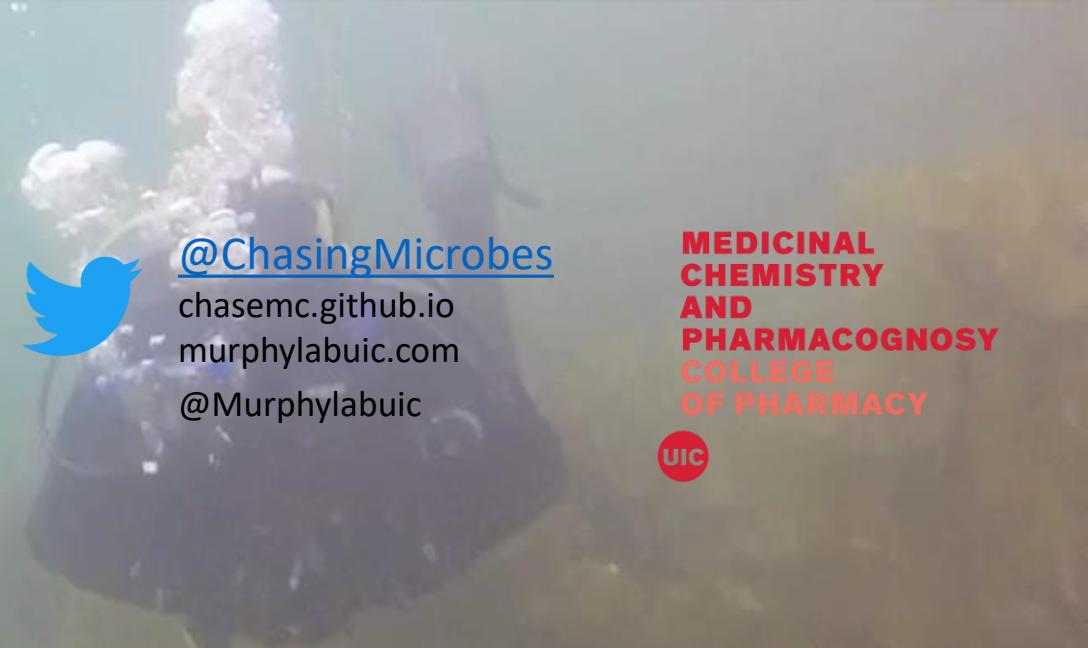
Dr. Linh Nguyen

Maryam Elfeki

Vanessa Nepomuceno

Erin Conley

Christie Kang



[@ChasingMicrobes](https://twitter.com/ChasingMicrobes)

chasemc.github.io

murphylabuic.com

@Murphylabuic

MEDICINAL
CHEMISTRY
AND
PHARMACOGNOSY
COLLEGE
OF PHARMACY

