

Speed, versatility, and joy - don't choose only two!

Computational biology in Julia for fun and profit

Kevin Bonham, PhD

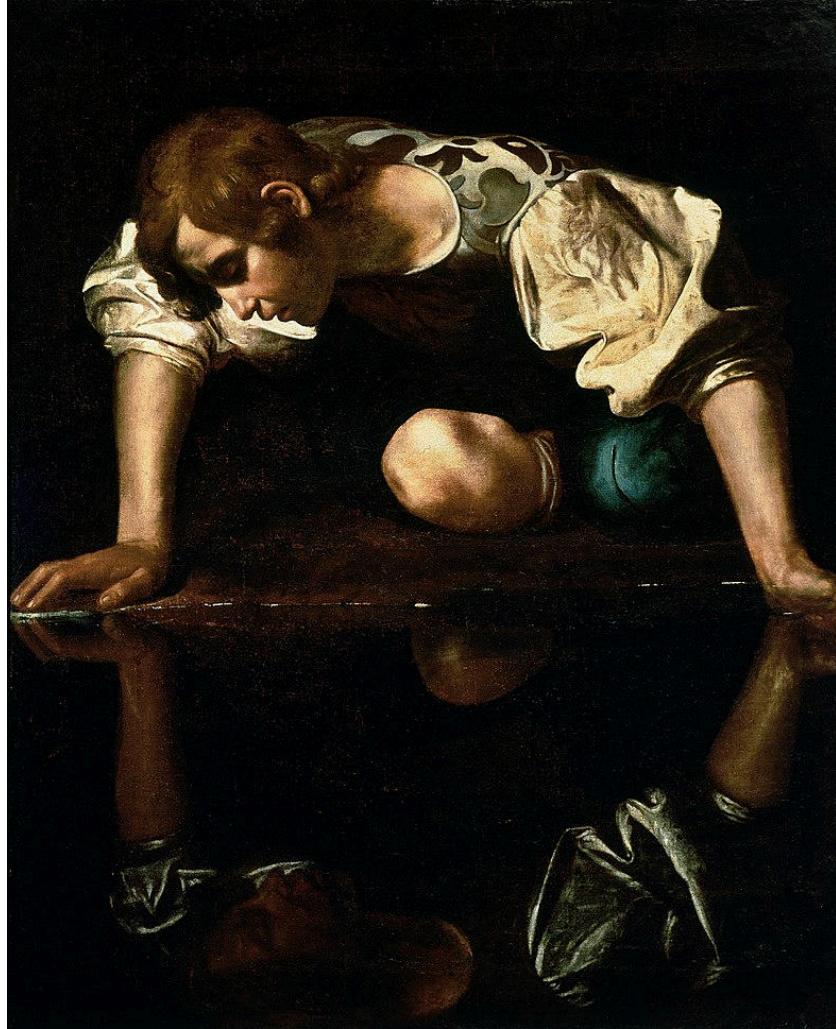
2025-05-28

Speed, versatility, and joy - don't chose only two!

Outline

- About me
- Why Julia?
- Getting started with julia

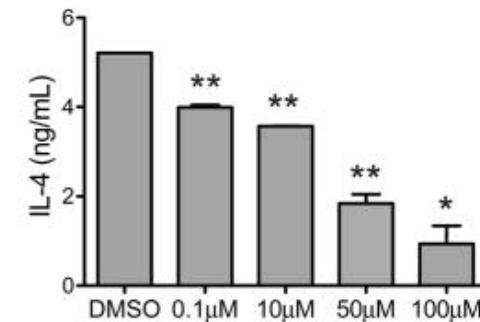
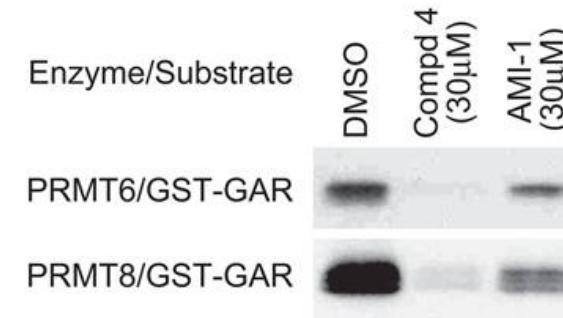
About me



[https://en.wikipedia.org/wiki/Narcissus_\(Caravaggio\)](https://en.wikipedia.org/wiki/Narcissus_(Caravaggio))

At the bench - small molecule inhibitors of T-cells

- B.S. from UCSD in Biochemistry and Cell biology
- Postbac with Kerry Mowen - chemical inhibitors of protein arginine methyl-transferases (PRMTs)

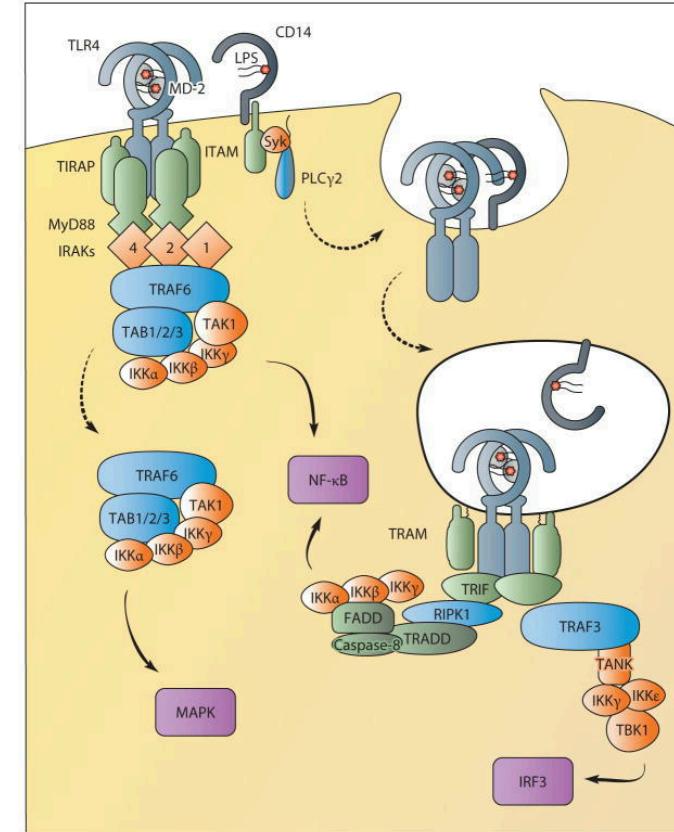


At the bench - Toll-like receptor signaling

- Investigation of the cell-biology of TIRAP, an adaptor protein - Established literature showed TIRAP was confined to the plasma membrane (eg TLR4, responding to LPS)
- TIRAP KO macrophages could still respond with endosomal TLRs (eg TLR9, responding to CpG)

Bonham, et. al., *Cell* (2014)

Brubaker, Bonham, et. al., *Ann. Rev. Immunol.* (2014)

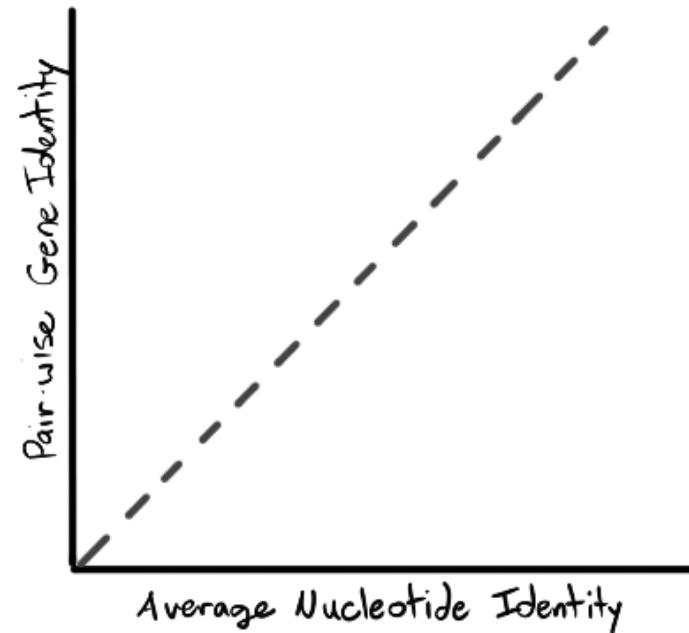


Transition to Computing - HGT in Cheese-associated bacteria

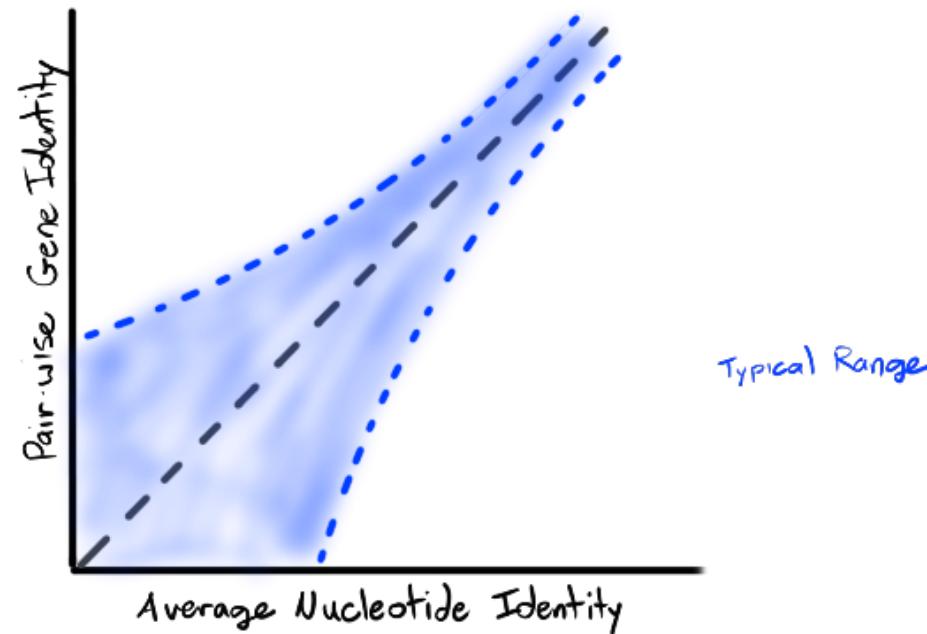


Wolfe, et. al., *Cell* (2014)

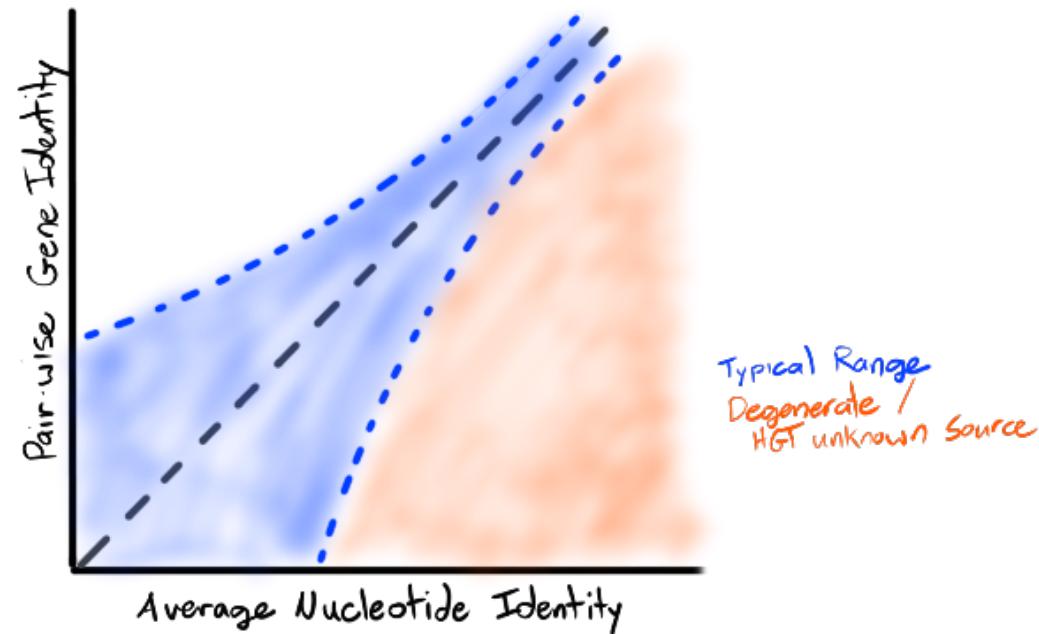
Kvasir - A python package for HGT discovery



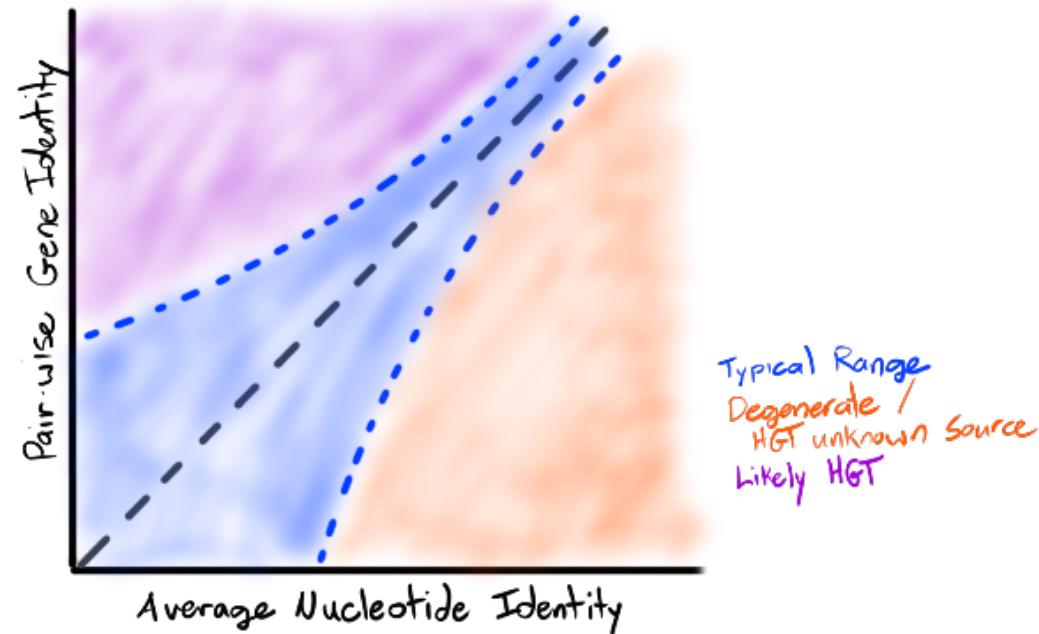
Kvasir - A python package for HGT discovery



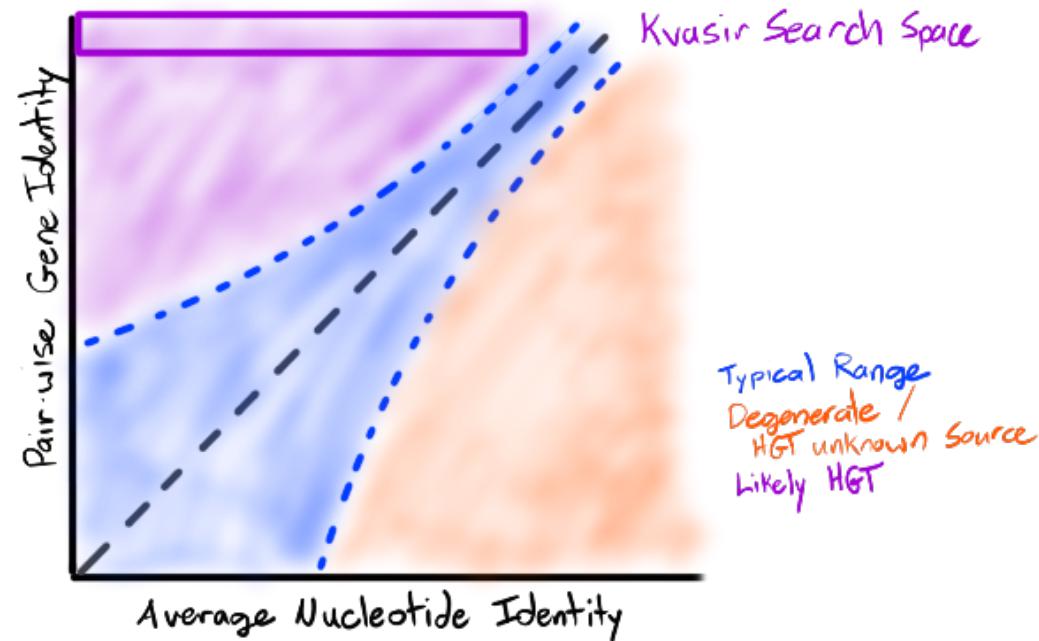
Kvasir - A python package for HGT discovery



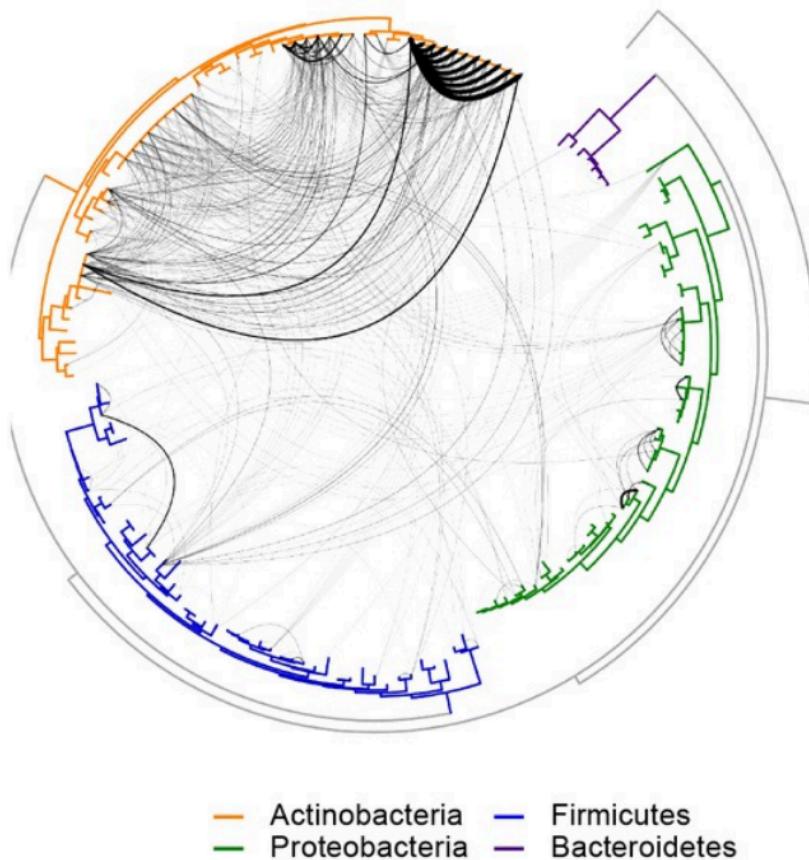
Kvasir - A python package for HGT discovery



Kvasir - A python package for HGT discovery



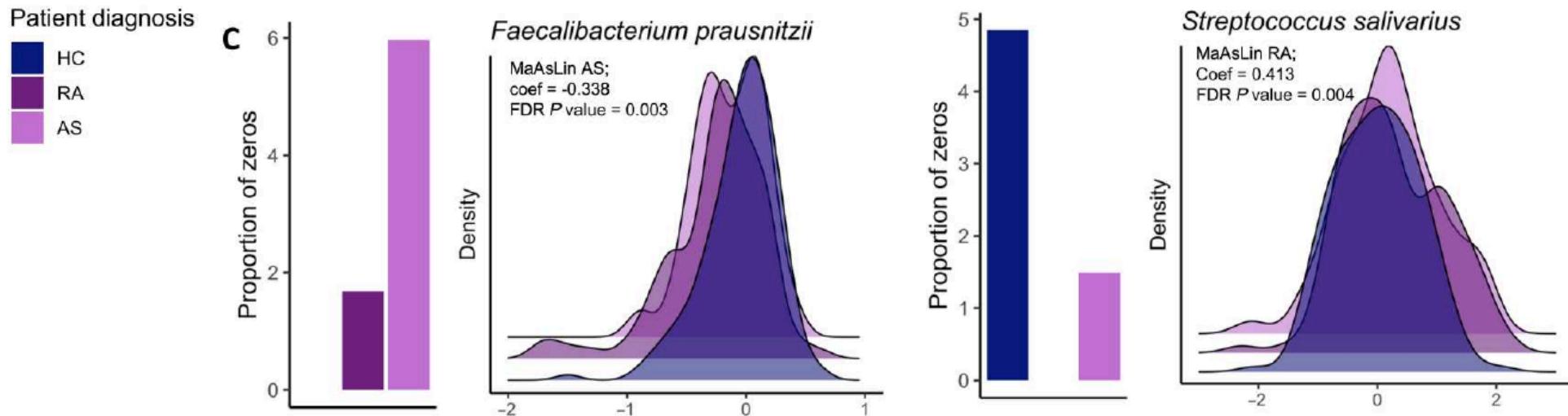
Kvasir - A python package for HGT discovery



- 200 cheese-associated isolates
- > 200 putative HGT genomic regions
- > 4700 protein coding genes
- enriched for iron transport / siderophore capture

Bonham, et. al., *eLife* (2017)

Multi-'omics and Data Science - Human microbiome epidemiology

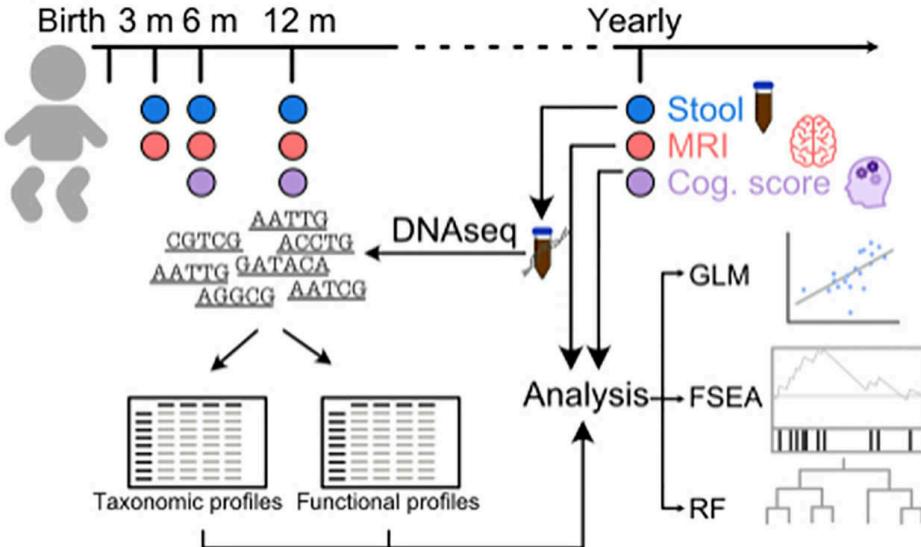


Thompson, et. al., *Sci. Transl. Med.* (2023)

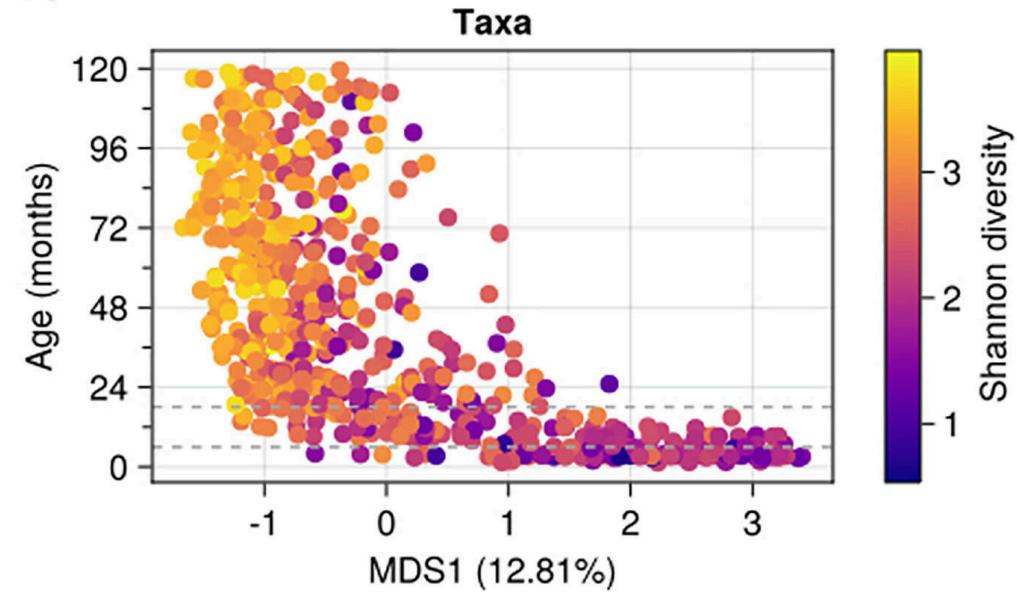
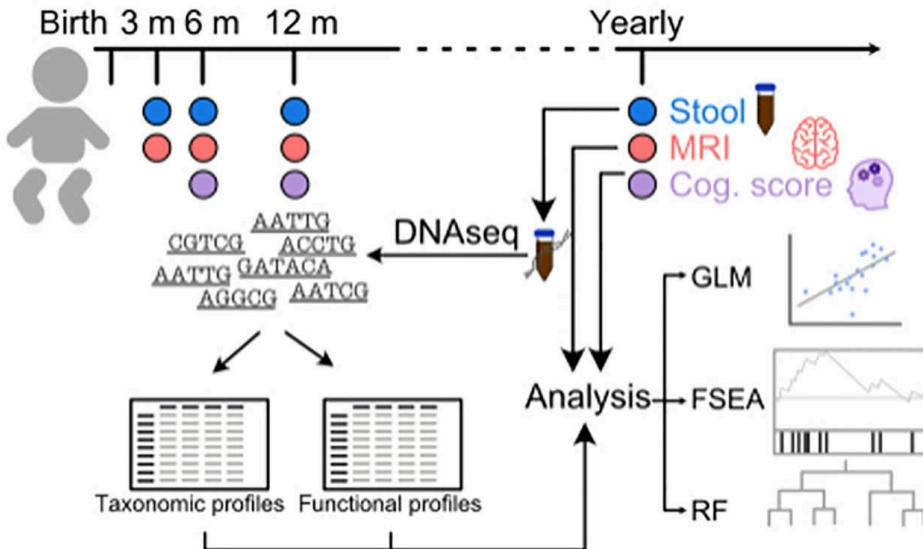
Kevin Bonham, PhD | BHCSD Seminar, May 2025

9 / 36

The gut microbiome and brain development

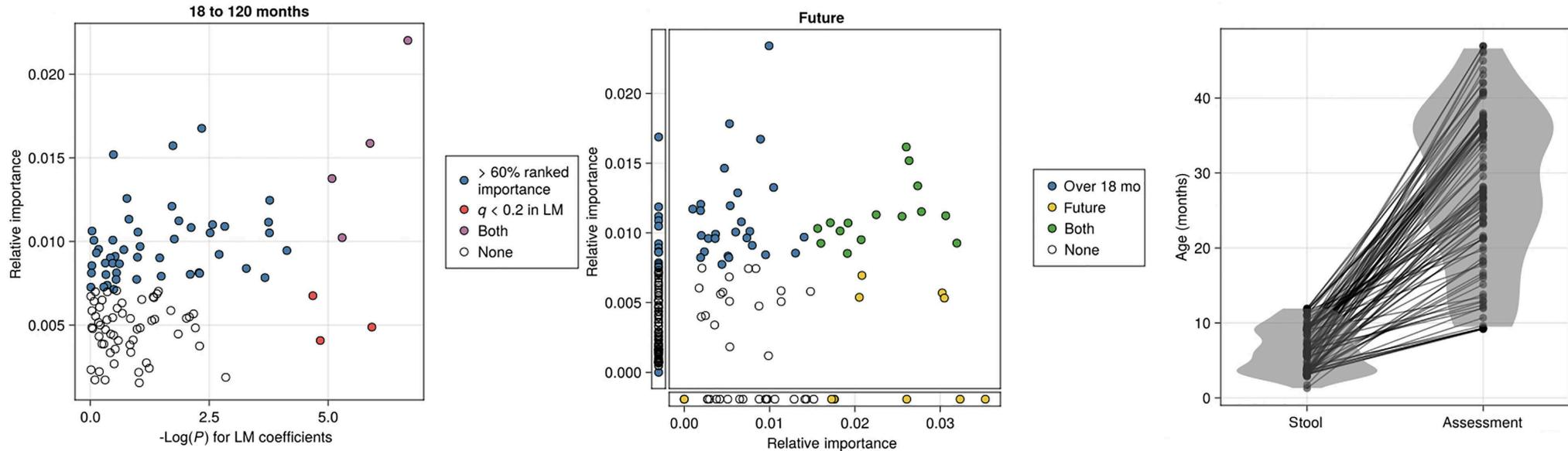


The gut microbiome and brain development



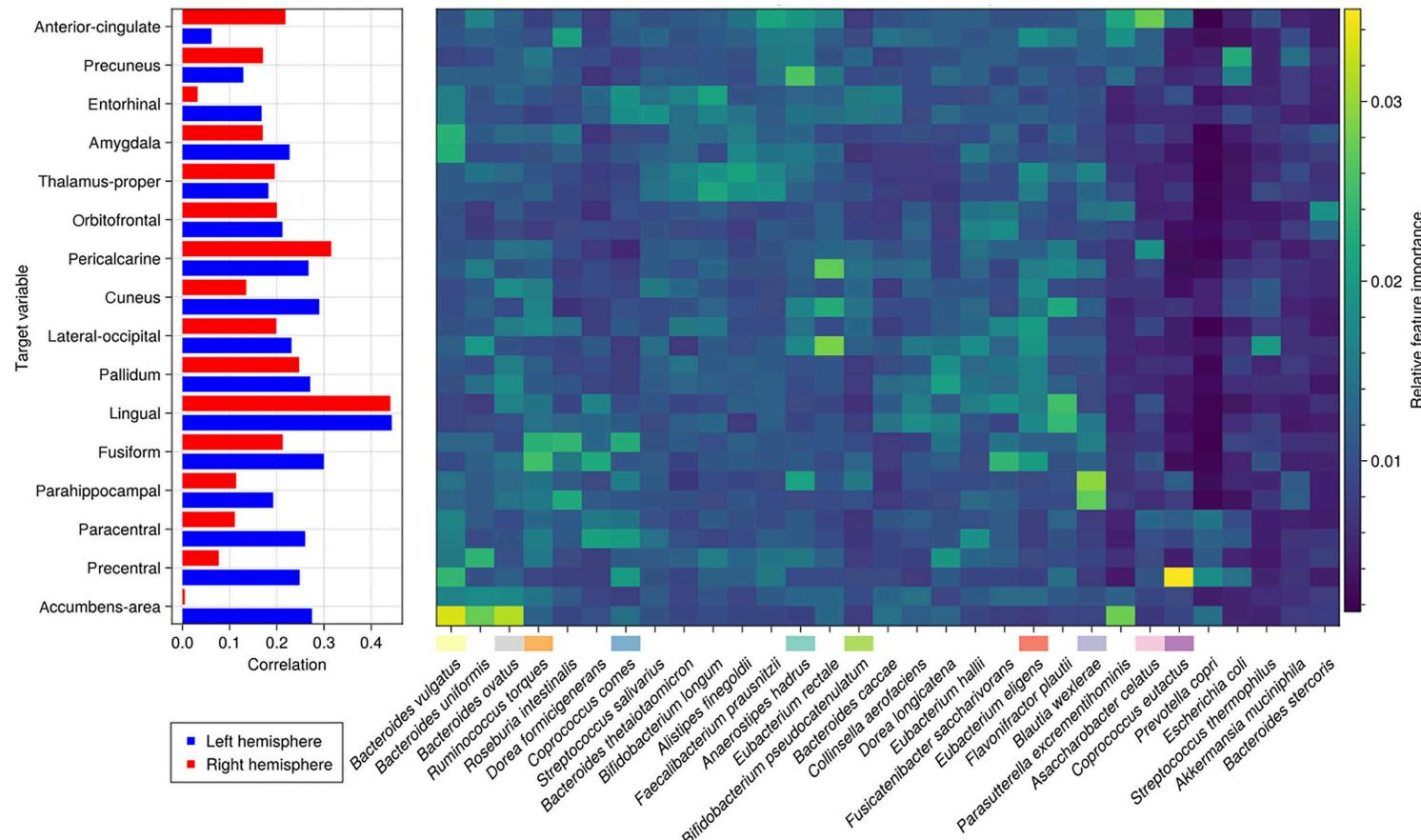
Bonham, et. al., *Sci. Adv.* (2023)

The gut microbiome and brain development



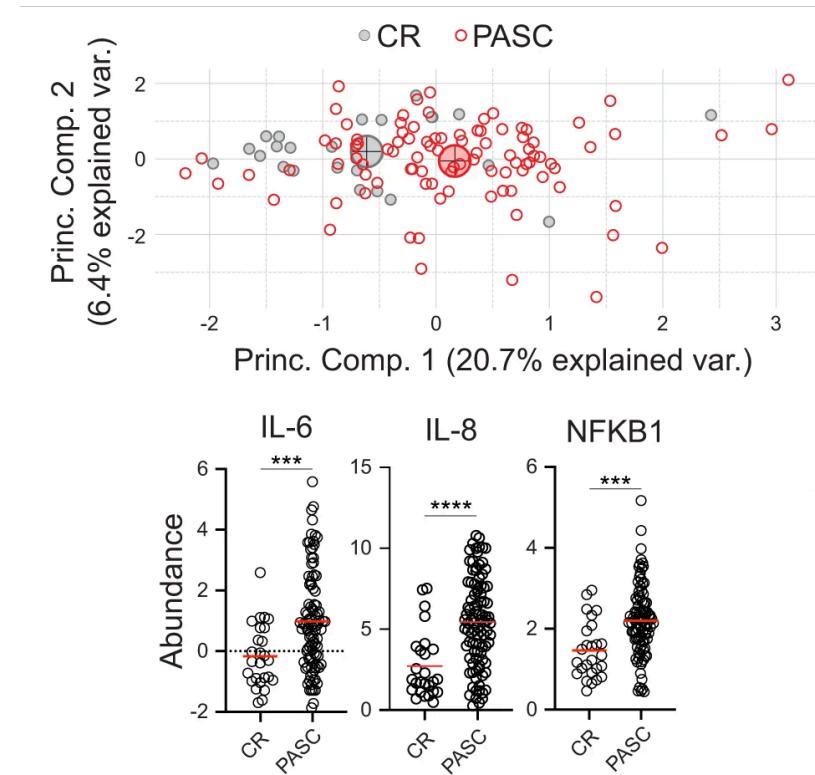
Bonham, et. al., *Sci. Adv.* (2023)

The gut microbiome and brain development



Immune dysregulation in Long-COVID

- Post-acute sequelae of COVID-19 (PASC) or “Long-COVID” is a constellation of persistent symptoms
- Affecting between 5% and 20% of acute COVID (pre-vaccine)
- Unknown etiology
- Investigated 7k blood proteins in matched PASC / COVID-recovered cohort

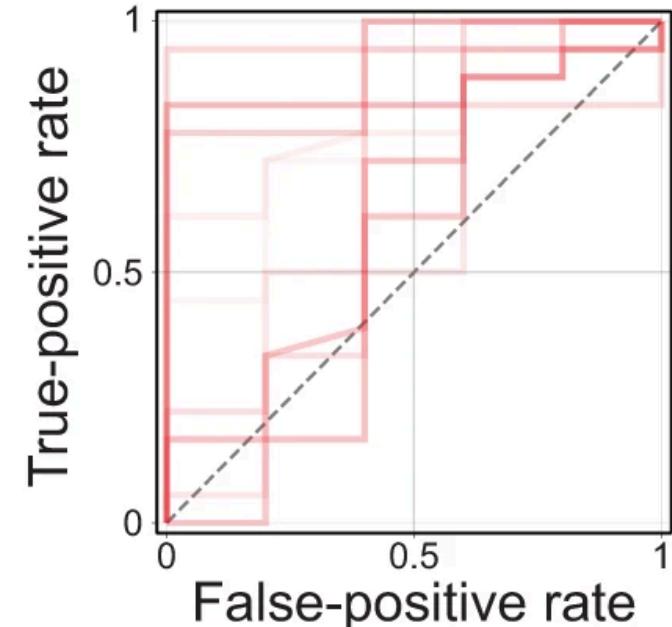
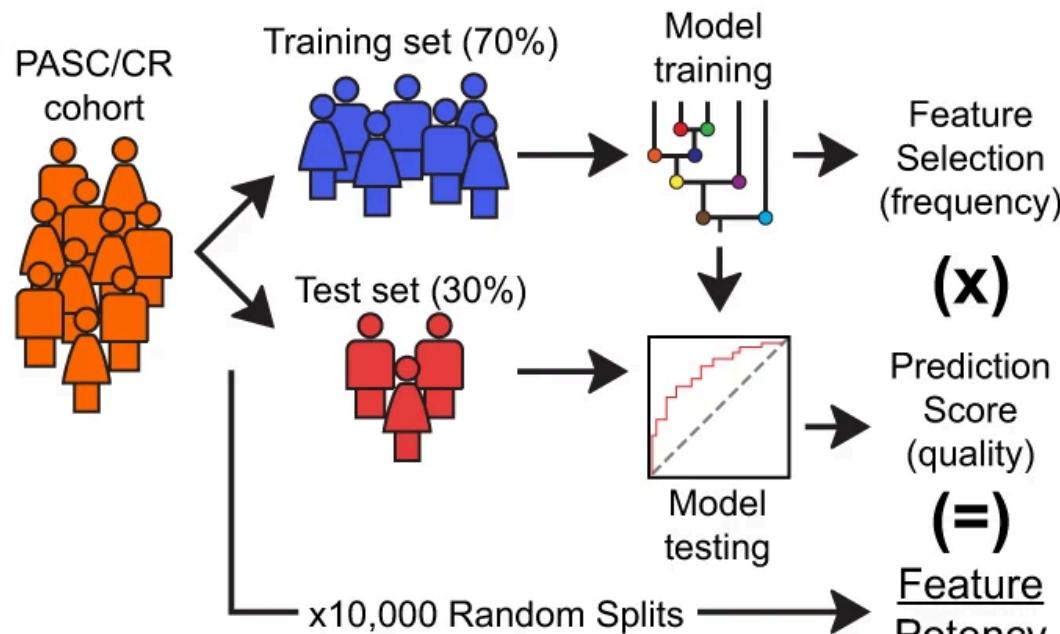


Woodruff and Bonham, *et. al.*, *Nat. Comm.* (2023)

Kevin Bonham, PhD | BHCSD Seminar, May 2025

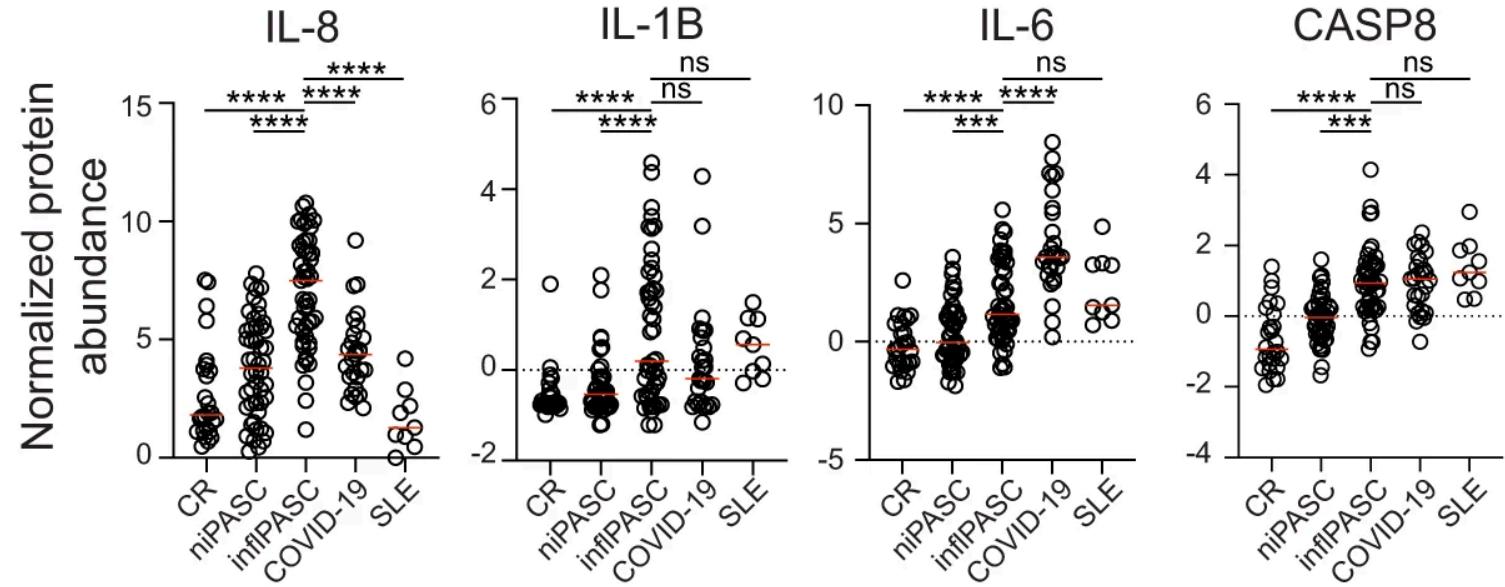
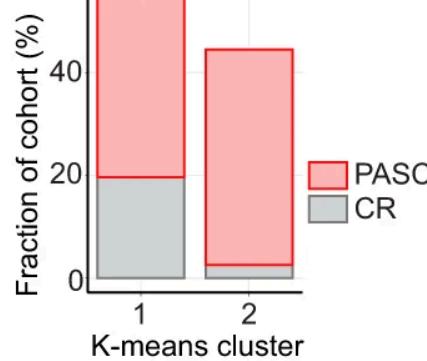
13 / 36

Immune dysregulation in Long-COVID



Woodruff and Bonham, et. al., *Nat. Comm.* (2023)

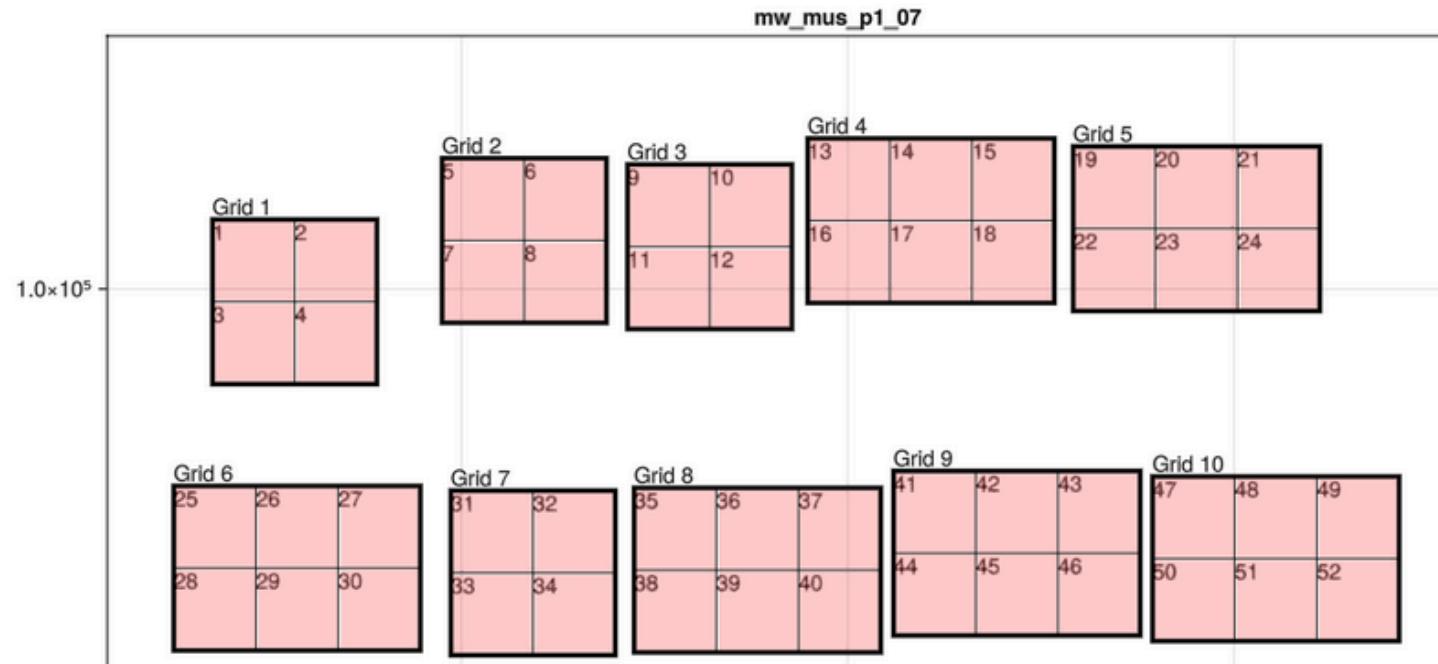
Immune dysregulation in Long-COVID



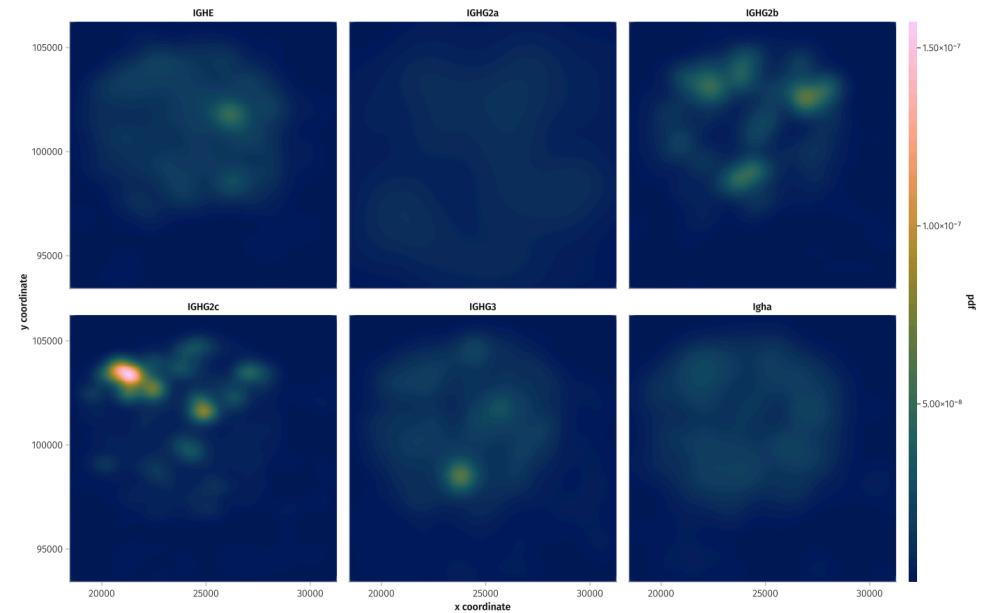
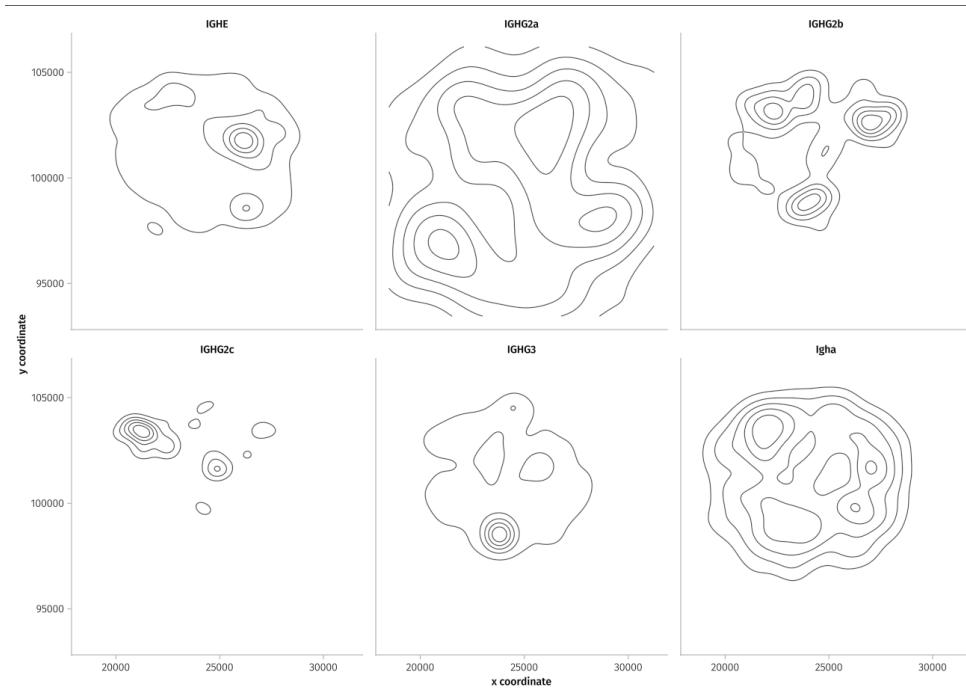
Woodruff and Bonham, et. al., *Nat. Comm.* (2023)

Current Work: SpatialOmics.jl

```
julia> using GLMakie  
  
julia> display_slide(ex["mw_mus_p1_07"])
```



Current Work: SpatialOmics.jl



Why Julia?

Why Julia?

My Julia journey

A screenshot of a Stack Overflow search results page. The search bar at the top contains the query "Dictionary help in Julia -". Below the search bar, there is a navigation menu with links for Home, Questions (which is highlighted), Staging Ground, and Tags. The main content area shows a question titled "Dictionary help in Julia -". A red box highlights the timestamp "Asked 10 years, 11 months ago".

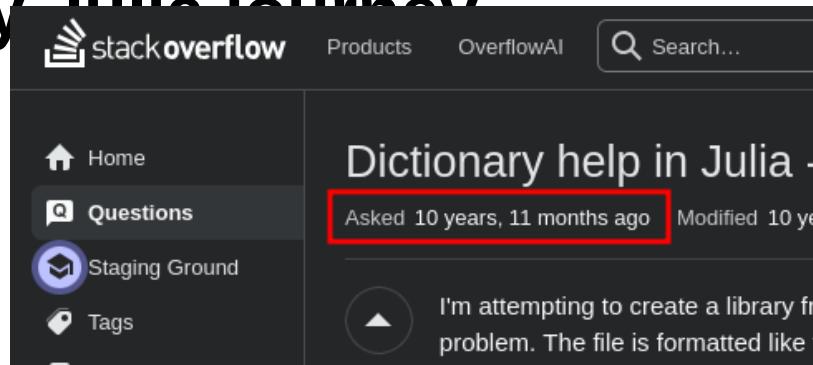
A screenshot of a GitHub repository interface. At the top, it shows "0 Open" and "3 Closed" issues. Below this, there are three pull request cards:

- add getindex(nt::NamedTuple, ::Colon) method**
#47842 by kescobo was merged on Dec 10, 2022
- Add getindex methods for NamedTuple** (collection)
#38878 by kescobo was merged on Jan 13, 2021 5 tasks done
- Add isnothing**
#29679 by kescobo was merged on Oct 17, 2018

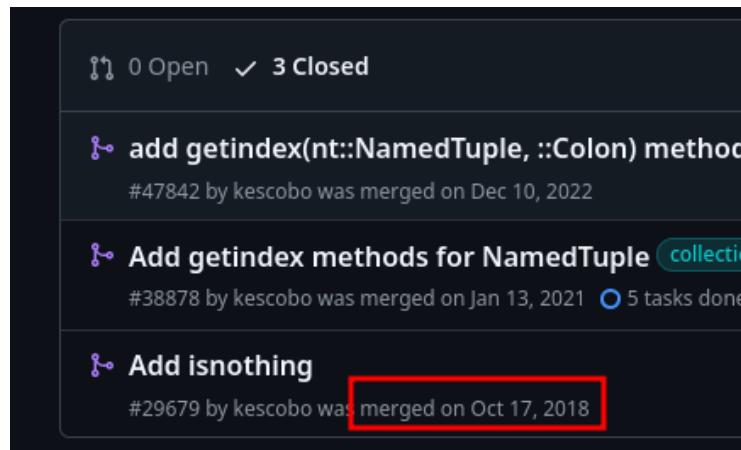
A red box highlights the merge date "Oct 17, 2018" for the third pull request.

Why Julia?

My Julia journey

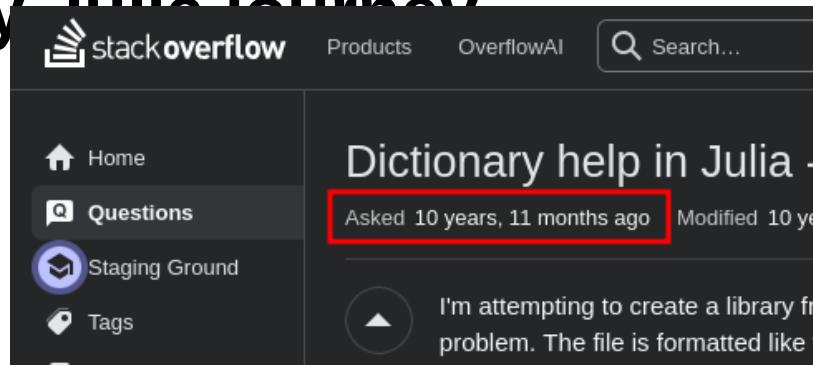


- First language I tried to learn (c. 2014, julia v0.3)

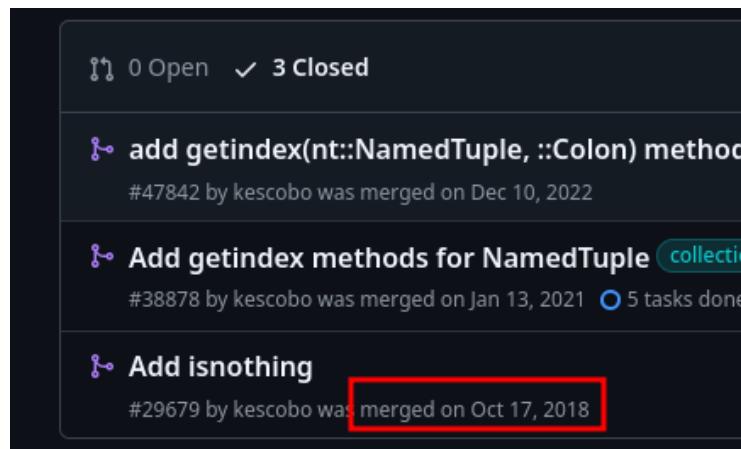


Why Julia?

My Julia journey

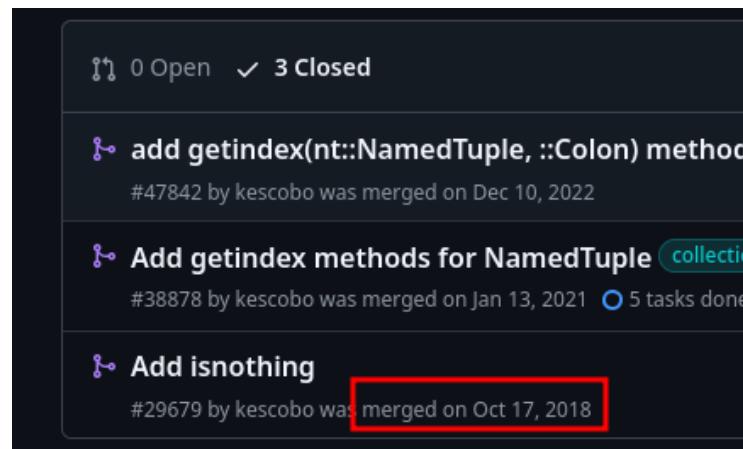
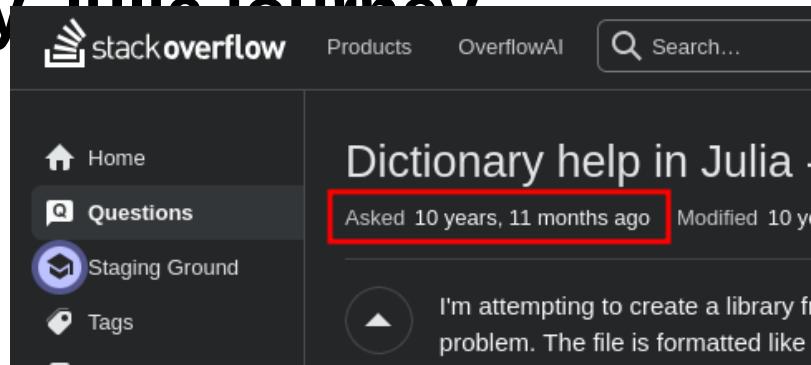


- First language I tried to learn (c. 2014, julia v0.3)
- Stopped in favor of python until May 2017



Why Julia?

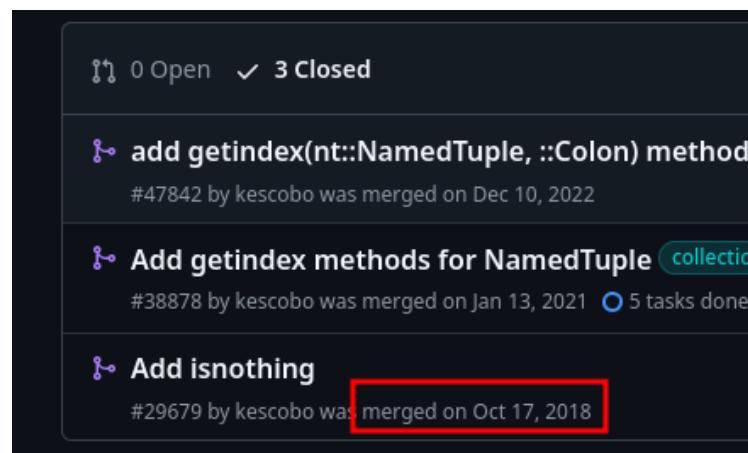
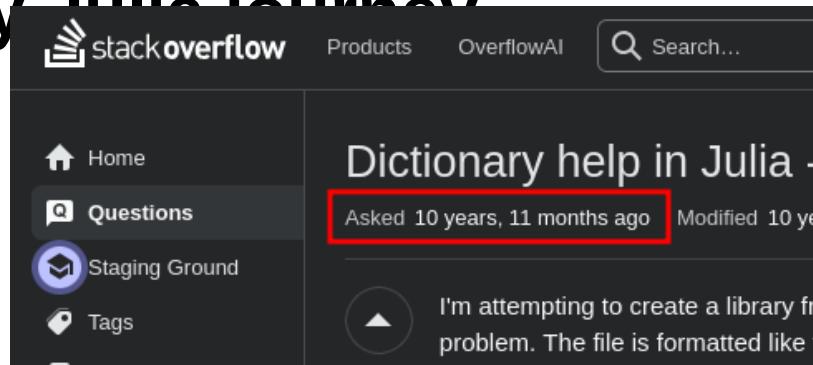
My Julia journey



- First language I tried to learn (c. 2014, julia v0.3)
- Stopped in favor of python until May 2017
- Hooked when I implemented minhash sketch just based on equation in paper, and it was *fast*

Why Julia?

My Julia journey



- First language I tried to learn (c. 2014, julia v0.3)
- Stopped in favor of python until May 2017
- Hooked when I implemented minhash sketch just based on equation in paper, and it was *fast*
- First registered package (`Microbiome.jl`) Oct 2017
- First PR to julia language in 2018

Why *not* julia

- You have everything you need in the language you're using, it's performant (enough), and you don't expect that to change

Why *not* julia

- You have everything you need in the language you're using, it's performant (enough), and you don't expect that to change
- You do not struggle with dependencies, environments, or reproducibility

Why *not* julia

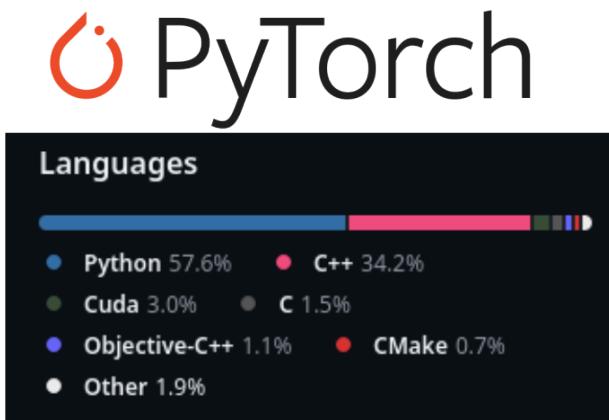
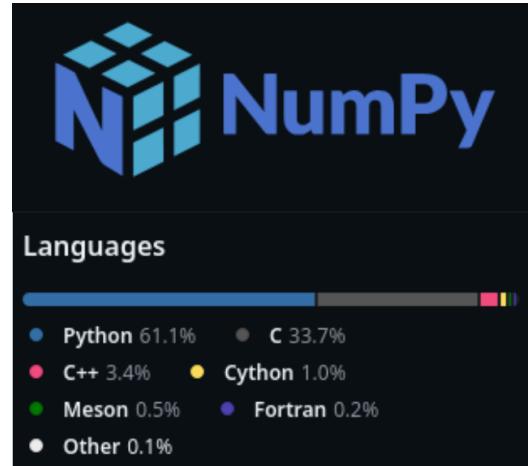
- You have everything you need in the language you're using, it's performant (enough), and you don't expect that to change
- You do not struggle with dependencies, environments, or reproducibility
- You sometimes need new functionality, but you are not interested in “rolling your own”

Why *not* julia

- You have everything you need in the language you're using, it's performant (enough), and you don't expect that to change
- You do not struggle with dependencies, environments, or reproducibility
- You sometimes need new functionality, but you are not interested in “rolling your own”
- If you hit performance bottlenecks, you'd rather wait for someone else to solve them

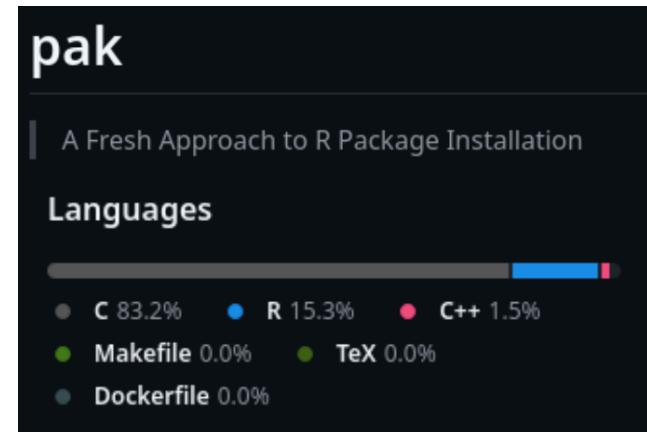
The two language problem

- In interactive languages, performance-critical code is written in compiled language
- Splits community of users and developers (the “two cultures problem”)
- Makes it difficult for domain-experts to contribute



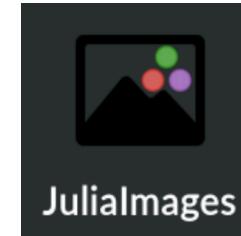
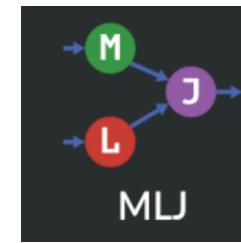
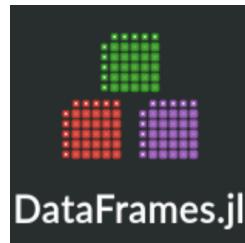
The two language problem

- In interactive languages, performance-critical code is written in compiled language
- Splits community of users and developers (the “two cultures problem”)
- Makes it difficult for domain-experts to contribute



Why Julia?

Julia (mostly) doesn't have a 2 language problem



Julia code is Just-In-Time (JIT) compiled

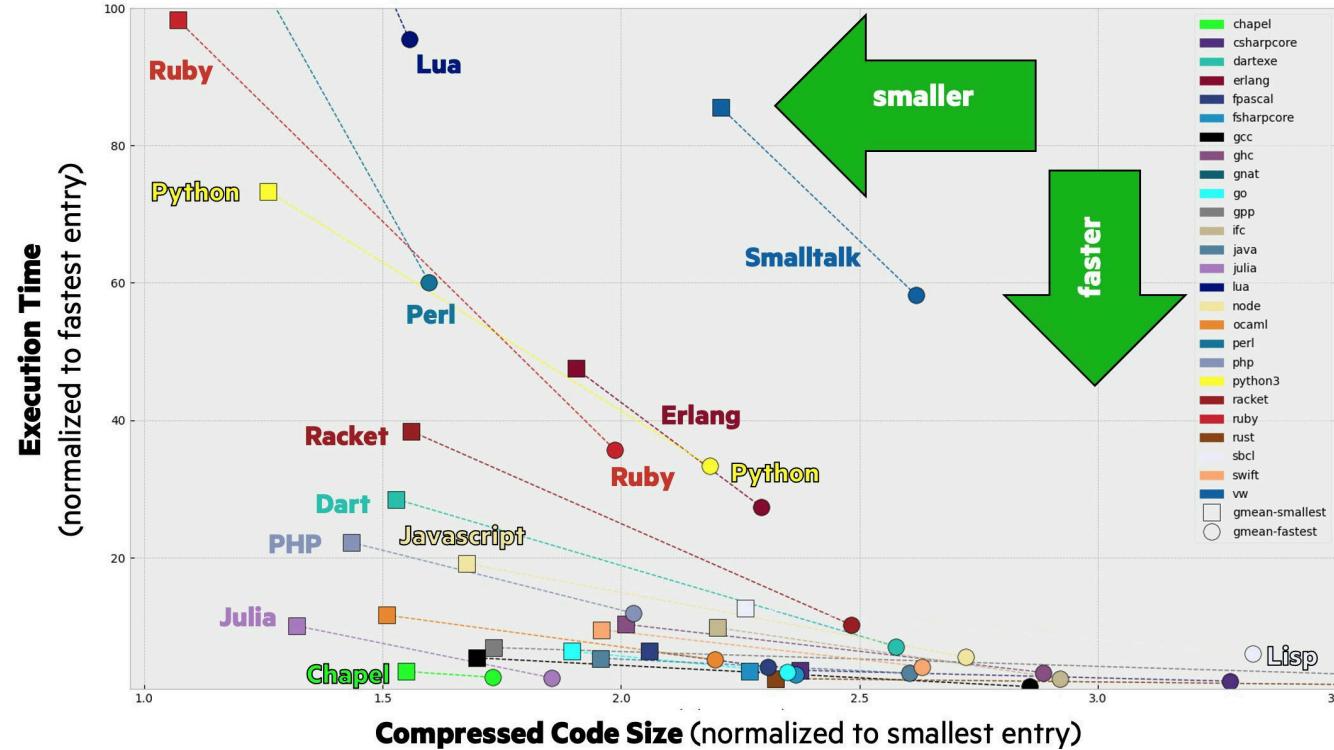
```
1 function show_compilation(n)
2     return n + 1
3 end
```

```
julia> @code_llvm show_compilation(1)
; Function Signature: show_compilation(Int64)
;  @ REPL[5]:1 within `show_compilation`
define i64 @julia_show_compilation_4487(i64 signext %"n:Int64") #0 {
top:
;  @ REPL[5]:2 within `show_compilation`
;  ↳ @ int.jl:87 within `+`
    %0 = add i64 %"n:Int64", 1
;
ret i64 %0
}
```

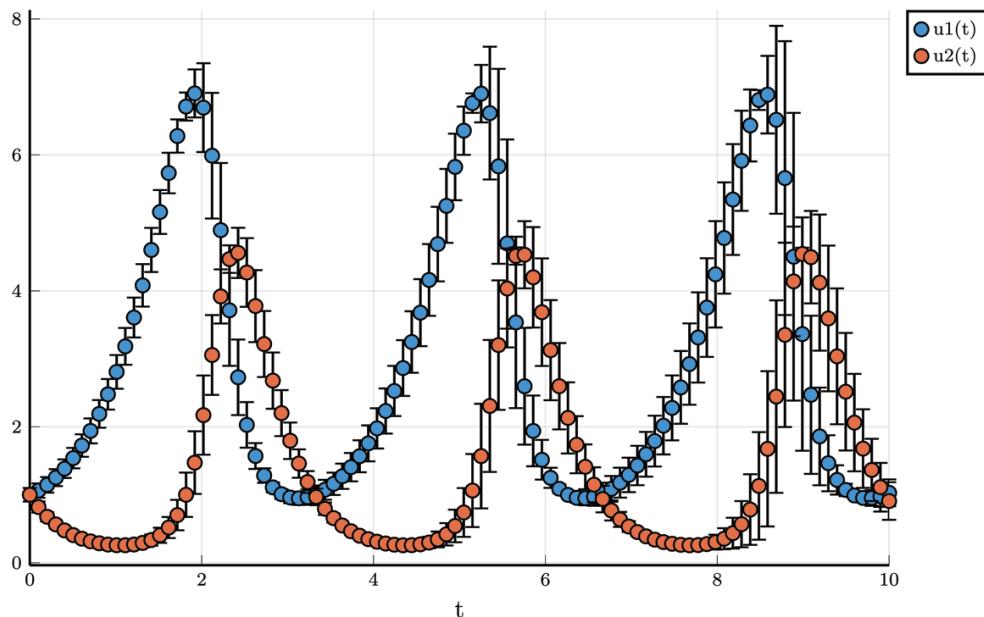
- code goes through multiple passes
- ultimately passed to LLVM for compilation
- only compiled for the types that it needs

Julia is (or can be) fast!

CLBG: ALL-LANGUAGE SUMMARY (FEB 7, 2023)



Code in julia is reusable and extensible (multiple-dispatch)



- Plotting a `ODESolution` object from `DifferentialEquations.jl`
 - containing Measurements from `Measurements.jl`
- plots solutions of the Lotka-Volterra system (lines) with correct error bounds
- Neither of these packages has code for handling types of the other package.

Plots Paper - DOI: 10.5334/jors.431

The unreasonable effectiveness of multiple-dispatch: <https://www.youtube.com/watch?v=kc9HwsxE1OY>

Package / environment manager (Pkg.jl) is built-in

- Pkg REPL mode accessed by pressing]
- Direct dependencies and compat stored in Project.toml
- Full environment with explicit versions stored in Manifest.toml
- strict adherence to SemVer
- Recreating environment as simple as] instantiate

```
(bhdsc_presentation) pkg> st
Status `~/Repos/bhdsc_presentation/Project.toml`
[31c24e10] Distributions v0.25.120

(bhdsc_presentation) pkg> add DataFrames CSV
  Updating registry at `~/.julia/registries/General.toml`
  Resolving package versions ...
    Updating `~/Repos/bhdsc_presentation/Project.toml` [336ed68f]
      + CSV v0.10.15
      + DataFrames v1.7.0
        Updating `~/Repos/bhdsc_presentation/Manifest.toml` [336ed68f]
          + CSV v0.10.15
          + CodecZlib v0.7.8
          + Crayons v4.1.1
          + DataFrames v1.7.0
          + DataValueInterfaces v1.0.0
          + FilePathsBase v0.9.24
          + InlineStrings v1.4.3
          + InvertedIndices v1.3.1
          + IteratorInterfaceExtensions v1.0.0
          + LaTeXStrings v1.4.0
```

Run Shell programs

```
julia> run(`ls -l`)
total 32
drwxr-xr-x. 1 kevin kevin  2238 May 27 11:30 assets
-rw-r--r--. 1 kevin kevin  7499 May 27 11:33 main.typ
-rw-r--r--. 1 kevin kevin 14497 May 27 11:17 Manifest.toml
-rw-r--r--. 1 kevin kevin   112 May 27 11:29 notebook_example.qmd
-rw-r--r--. 1 kevin kevin   159 May 27 11:17 Project.toml
Process(`ls -l`, ProcessExited(0))

julia> # press ;

shell> ls -l
total 32
drwxr-xr-x. 1 kevin kevin  2238 May 27 11:30 assets/
-rw-r--r--. 1 kevin kevin  7499 May 27 11:33 main.typ
-rw-r--r--. 1 kevin kevin 14497 May 27 11:17 Manifest.toml
-rw-r--r--. 1 kevin kevin   112 May 27 11:29 notebook_example.qmd
-rw-r--r--. 1 kevin kevin   159 May 27 11:17 Project.toml
```

BinaryBuilder.jl facilitates cross-platform compilation

```
julia> using bowtie_jll

julia> run(`$(bowtie2()) --help`)
Bowtie 2 version 2.5.1 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage:
  bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r> | --interleaved <i> | -b <bam>}
  [-S <sam>]

  <bt2-idx>  Index filename prefix (minus trailing .X.bt2).
              NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
  <m1>        Files with #1 mates, paired with files in <m2>.
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>        Files with #2 mates, paired with files in <m1>.
              Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
```

Run python or R code with PythonCall.jl, RCall.jl, and CondaPkg.jl

```
julia> using RCall

julia> R"x ← c(1,2,3)"
RObject{RealSxp}
[1] 1 2 3

julia> @rget x
3-element Vector{Float64}:
 1.0
 2.0
 3.0
```

Run python or R code with PythonCall.jl, RCall.jl, and CondaPkg.jl

```
julia> using RCall

julia> R"x ← c(1,2,3)"
RObject{RealSxp}
[1] 1 2 3

julia> @rget x
3-element Vector{Float64}:
 1.0
 2.0
 3.0
```

```
julia> using PythonCall

julia> const seqio = pyimport("Bio.SeqIO")
Python: <module 'Bio.SeqIO' from '/home/kevin/Repos/bhdsc_pres/default/lib/python3.12/site-packages/Bio/SeqIO/__init__.py'>

julia> for record in seqio.parse(exfastq, format="fastq")
           println(record.id)
       end
FSRRS4401BE7HA
FSRRS4401BRRTC
FSRRS4401B64ST
FSRRS4401EJ0YH
FSRRS4401BK0IB
FSRRS4401ARCCB
FSRRS4401CM938
FSRRS4401EQLIK
FSRRS4401AOV6A
FSRRS4401EG0ZW
```

Getting started with julia

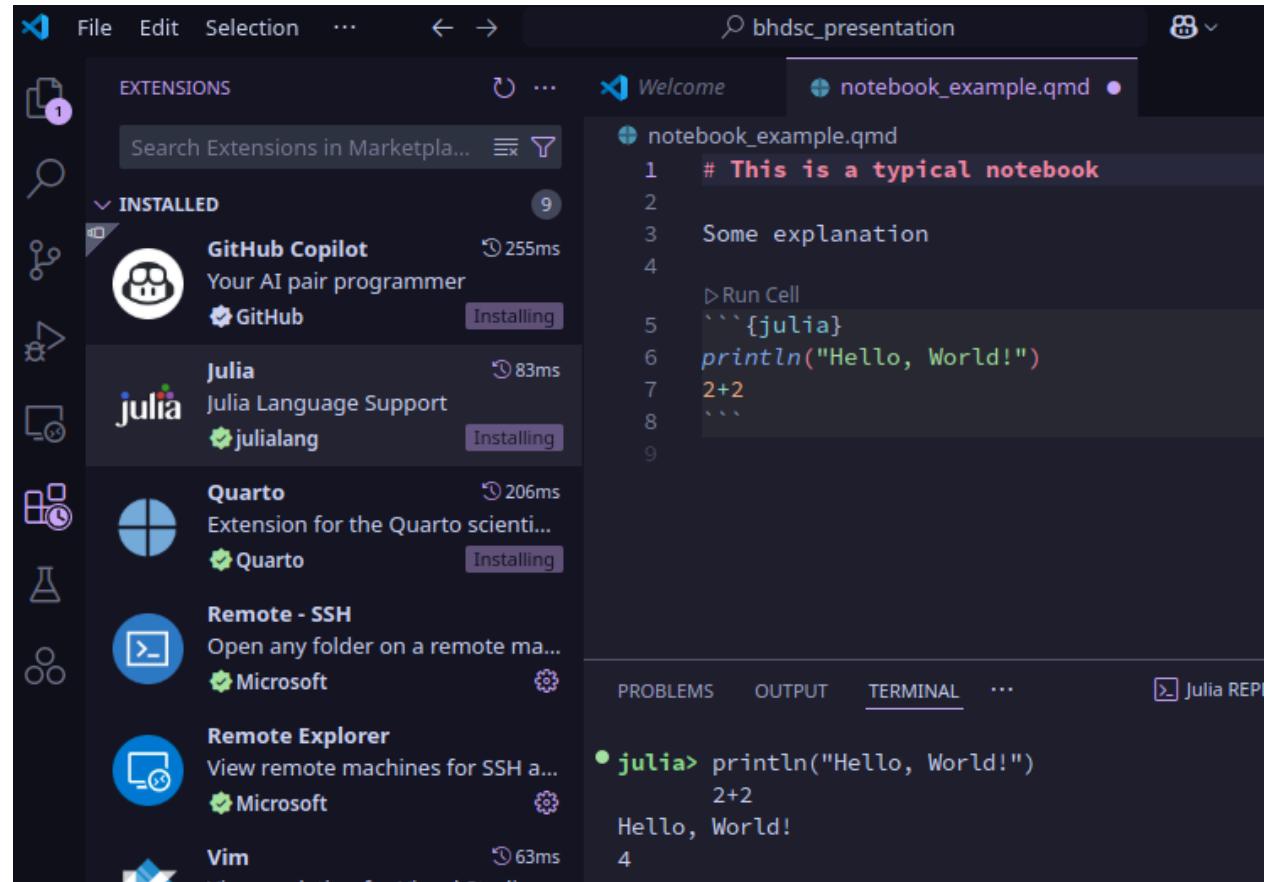
Installation with juliaup

- inspired by rustup
- curl -fsSL https://install.julialang.org | sh
- also available from the Windows store

```
~/Repos/bhdsc_presentation via ∵ v1.11.5 via t v0.13.1 took 56s
> juliaup add beta
Checking for new Julia versions
'beta' is already installed.

~/Repos/bhdsc_presentation via ∵ v1.11.5 via t v0.13.1
> juliaup update
Checking for new Julia versions
Updating channel beta
Installing Julia 1.12.0-beta3+0.x64.linux.gnu
  Downloading: [>                                ] 432.00 KiB/277.04 MiB eta:
```

Use any text editor (VS Code is well supported)



Syntax Demo

Useful Ecosystems for Bioinformatics and Data Science

- BioJulia: <https://biojulia.dev>
 - Efficient sequence types
 - parsers for common formats, Automa.jl to generate FSM-based parsers
 - SingleCellProjections.jl and BioMakie.jl

Useful Ecosystems for Bioinformatics and Data Science

- BioJulia: <https://biojulia.dev>
 - Efficient sequence types
 - parsers for common formats, Automa.jl to generate FSM-based parsers
 - SingleCellProjections.jl and BioMakie.jl
- JuliaData:
 - DataFrames.jl, CSV.jl, Tables.jl
 - Databases (DuckDB.jl, SQLite.jl, etc)

Useful Ecosystems for Bioinformatics and Data Science

- BioJulia: <https://biojulia.dev>
 - Efficient sequence types
 - parsers for common formats, Automa.jl to generate FSM-based parsers
 - SingleCellProjections.jl and BioMakie.jl
- JuliaData:
 - DataFrames.jl, CSV.jl, Tables.jl
 - Databases (DuckDB.jl, SQLite.jl, etc)
- Pluto.jl - interactive notebooks

Useful Ecosystems for Bioinformatics and Data Science

- BioJulia: <https://biojulia.dev>
 - Efficient sequence types
 - parsers for common formats, Automa.jl to generate FSM-based parsers
 - SingleCellProjections.jl and BioMakie.jl
- JuliaData:
 - DataFrames.jl, CSV.jl, Tables.jl
 - Databases (DuckDB.jl, SQLite.jl, etc)
- Pluto.jl - interactive notebooks
- JulialImages
- SciML / DifferentialEquations.jl: <https://sciml.ai>
- Turing.jl - Bayesian statistics

Resources

Note: TMC network blocks github.io domains

- Modern julia workflows: <https://modernjuliaworkflows.org>
- Why julia Manifesto: <https://github.com/Datseris/whyjulia-manifesto>
- Learn julia in Y minutes: <https://learnxinyminutes.com/julia>
- Julia Zero2Hero: <https://github.com/Datseris/Zero2Hero-JuliaWorkshop/>
- Julia documentation: <https://docs.julialang.org>
- Julia communities (slack, discourse): <https://julialang.org/community/>
- BioJulia Tutorials: <https://biojulia.dev/BioTutorials>

Contact info



[https://en.wikipedia.org/wiki/Narcissus_\(Caravaggio\)](https://en.wikipedia.org/wiki/Narcissus_(Caravaggio))

- 🐱 / ❤️: @kescobo
- 🔬 🌐 (WIP) - <https://lab.bonham.ch>
- 🧑‍🤝‍🧑 🌐 - <https://blog.bonham.ch>
- Julia slack/discourse/zulip etc - just search my name
- 🦋 <https://bsky.app/@kevinbonham.com>