

Julia in Pharma

Dr. Viral B. Shah (Julia Computing)

Dr. Vijay Ivaturi (Pumas-AI)

R/Pharma 2020

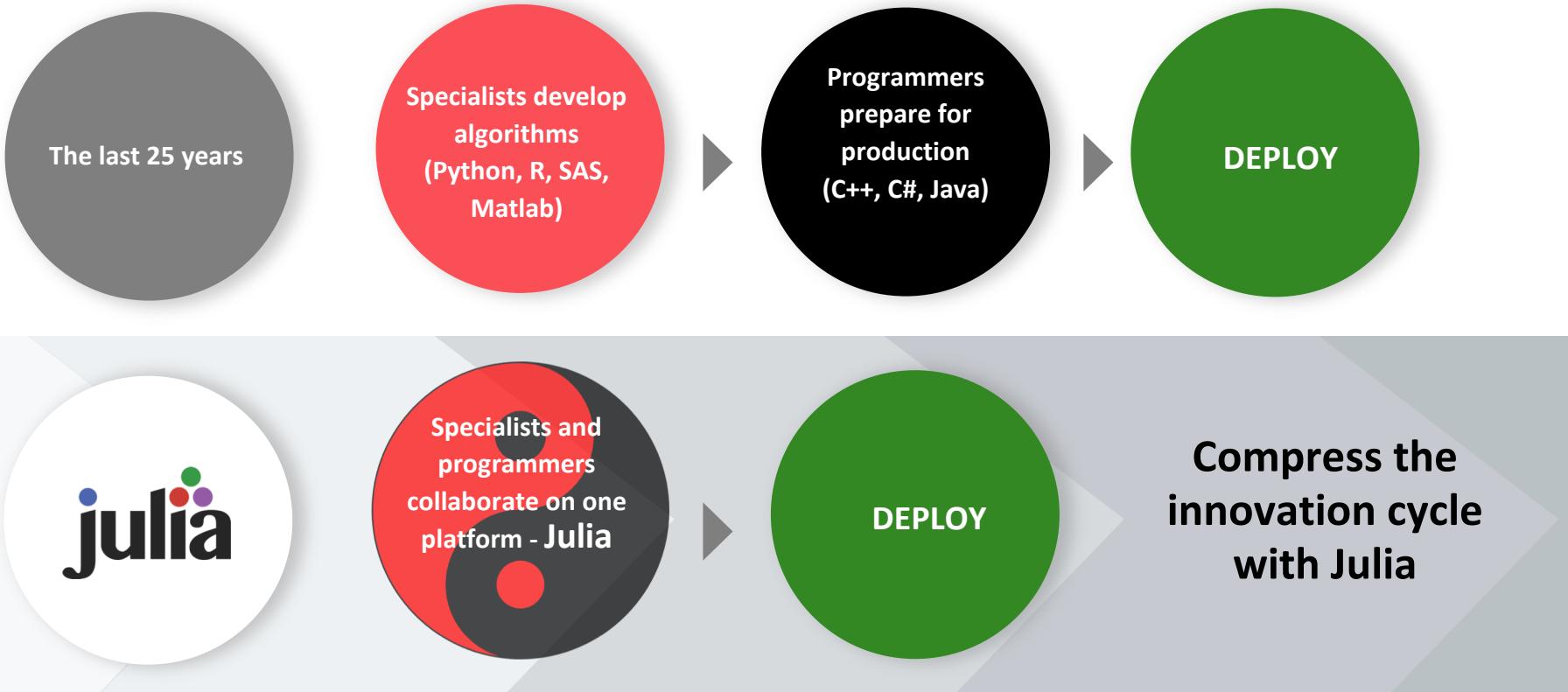


Email: info@juliacomputing.com



Email: info@pumas.ai

We created Julia to solve the two language problem



Origin and Journey

1.	2009	Discussion about a new language
2.	2012	<u>Why we created Julia</u>
3.	2013	Julia becomes the “Ju” in Jupyter
4.	2015	<u>Julia co-creators found Julia Computing, Inc.</u>
5.	2017	<u>1M Julia downloads</u>
5.	2018	<u>Julia 1.0.</u>
6.	2019	10M Julia downloads. <u>Wilkinson Prize</u> . <u>Sidney Fernbach Prize</u> .
7.	2020	Pumas.ai founded. 20M Julia downloads. <u>Julia 1.5 released</u>

Dr. Viral Shah Prof. Alan Edelman



Dr. Jeff Bezanson Stefan Karpinski

Over 10,000 organizations are using Julia



LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY



BLACKROCK®



WESTERN ASSET

abbvie



ExxonMobil



JPMORGAN
CHASE & CO.



moderna
messenger therapeutics

SANOFI

STATE STREET.



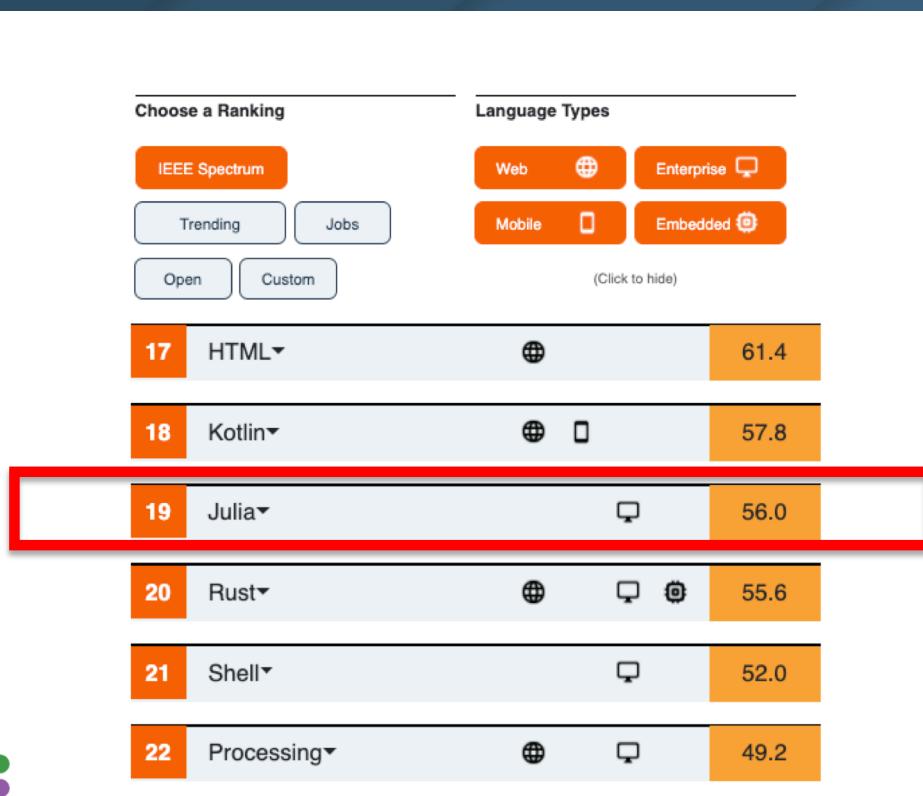
EY
Building a better working world

Over 1,500 Universities are using and teaching Julia



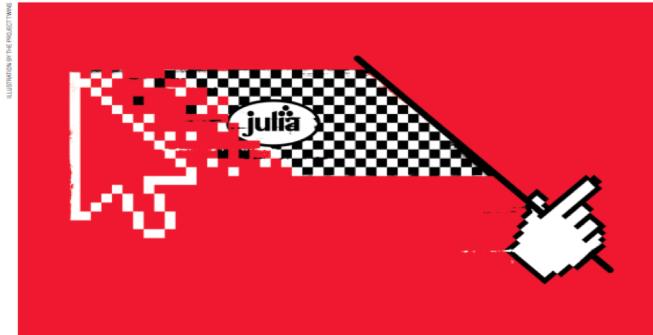
Rapid adoption for a young language

20M downloads; 4,000 packages; 10,000 companies; 1,500 universities



TOOLBOX JULIA: COME FOR THE SYNTAX, STAY FOR THE SPEED

Researchers often find themselves coding algorithms in one programming language, only to have to rewrite them in a faster one. An up-and-coming language could be the answer.



BY JEFFREY M. PERKEL

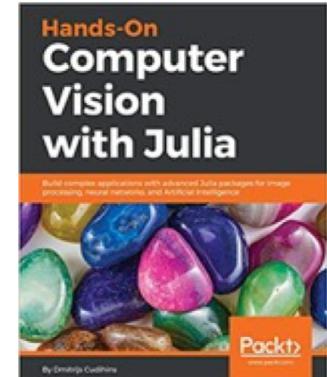
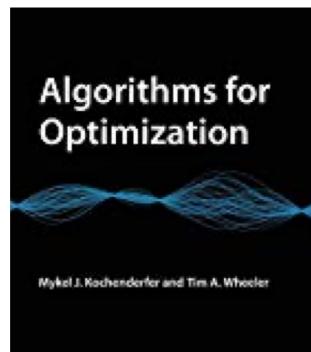
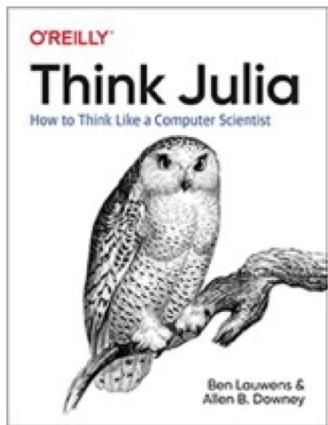
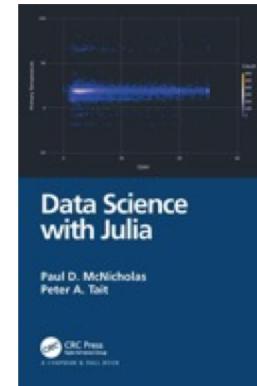
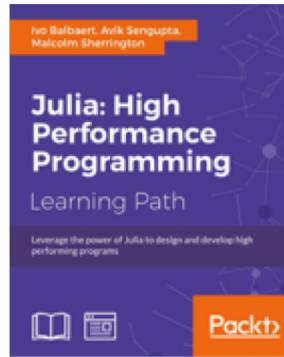
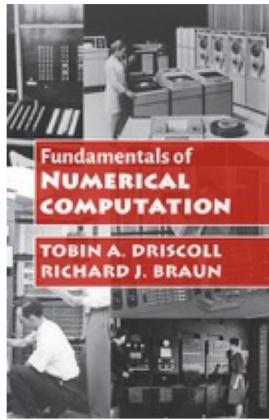
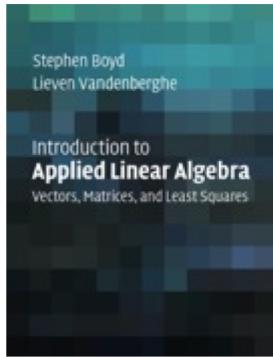
When it comes to climate modeling, even the simplest model must link air, land, sun and sea, and the complicated physics that links them; these models can run to millions of lines of code, which are executed on the world's most powerful computers. So when the Climate-Modeling Center of the Climate Modeling Alliance (CLIMA) — a coalition of US-based scientists, engineers and mathematicians — set out to build a model from the ground up, they opted for a language that could handle their needs. They opted for Julia.

Launched in 2012, Julia is an open-source language that combines the interactivity and syntax of 'scripting' languages, such as Python, Matlab and R, with the speed of compiled languages like Fortran and C.

nature
International journal of science

Giraldo says, and since adopting Julia he has seen an uptick in interest. 'Some of them are really interested in climate modelling, but others are intrigued by the idea of using Julia for other applications,' he says. Jane Herriman, who is studying materials science at the California Institute of Technology in Pasadena, says that she has seen ten-fold improvements in her calculations since switching to Julia.

Books on Julia



Google.ai Head Jeff Dean and Fast.ai co-author Jeremy Howard on on Julia

 **Jeff Dean** 
@JeffDean

Following

Julia + TPUs = fast and easily expressible ML computations!

Keno Fischer @KenoFischer
Our new paper today: arxiv.org/abs/1810.09868. Compile your #juliaLang code straight to @Google's #CloudTPU. Must go faster! We'll have an (alpha quality) repo up soon for people to start playing with this.

6:23 AM - 24 Oct 2018

240 Retweets 617 Likes



6 240 617 



 **Jeremy Howard** @jeremyphoward · Sep 3
Replies to @jeremyphoward @HamelHusain and 2 others
Python is highly dynamic. We pay the price of that with performance (which can't be avoided whilst writing python code that's handled by cpython), and static analysis (which can only be avoided by skipping the dynamism pretty much entirely)

2 5 

 **Jeremy Howard** @jeremyphoward · Sep 3
Yes, it has a rich ecosystem. But I'm talking about the *language*. I'm talking about what we see if we take a 5-10 year view.

If the community focuses on turning Python into a static language, then we lose its raison d'être.

1 7 

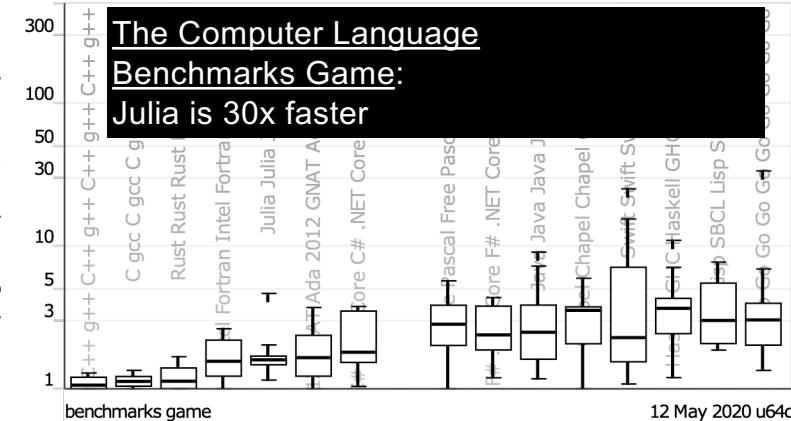
 **Jeremy Howard** @jeremyphoward · Sep 3
I have no interest in building a ecosystem around the limited subset of Python that supports static analysis.

For that, I'd much rather work on Julia or Swift, which have far stronger type systems and far stronger static analysis support.

2 8 

The need for speed: Common Benchmarks

How many times slower?

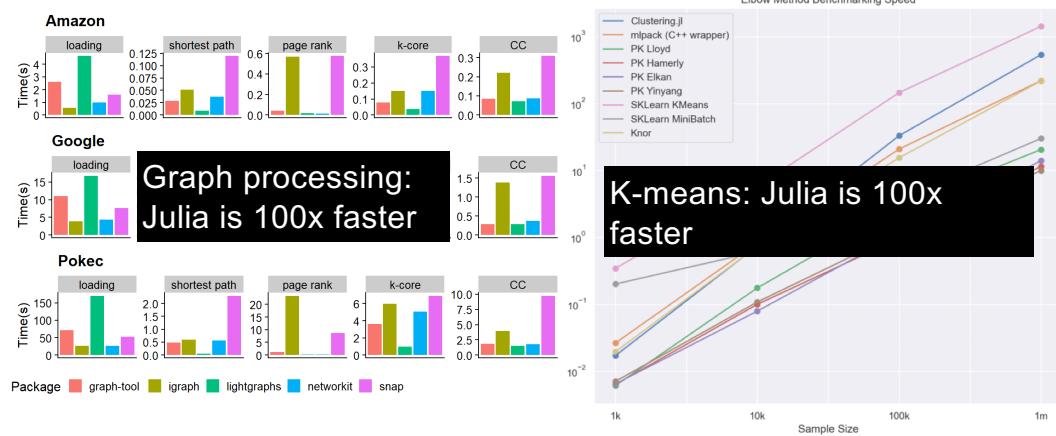


Input table: 100,000,000 rows x 9 columns (5 GB)

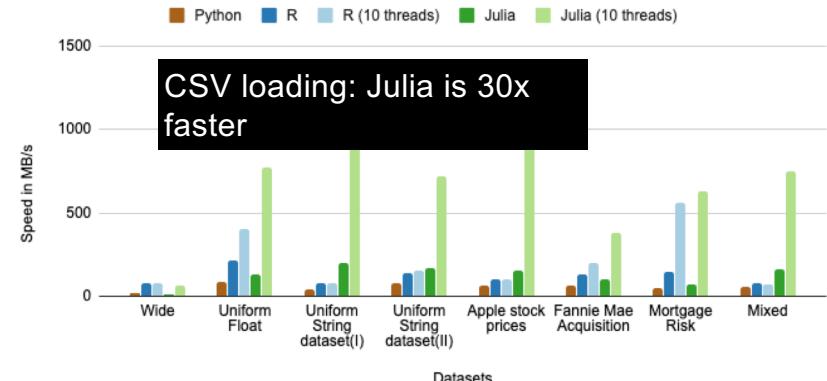
ClickHouse	20.3.8.53	2020-04-30
data.table	1.12.9	2020-04-22
spark	2.4.5	2020-03-04
DataFrames.jl	0.21.1	2020-05-24
(py)datatable	0.11.0a0	2020-05-24
pandas	1.0.3	2020-05-19
dplyr	0.8.5	2020-03-08
data.table	1.12.9	2020-04-22
ctc	0.1.0	2020-05-24
M	0.1.0	2020-05-24

H2O DataFrames benchmark: GroupBy
Julia is 2x faster

7s
13s
36s
42s
68s
92s
191s
305s
of memory pending



CSV reading benchmarks: Python, R, and Julia





Pluto Notebooks

Writing a notebook is not just about writing the final document — Pluto empowers the experiments and discoveries that are essential to getting there.

Explore models and share results in a notebook that is:

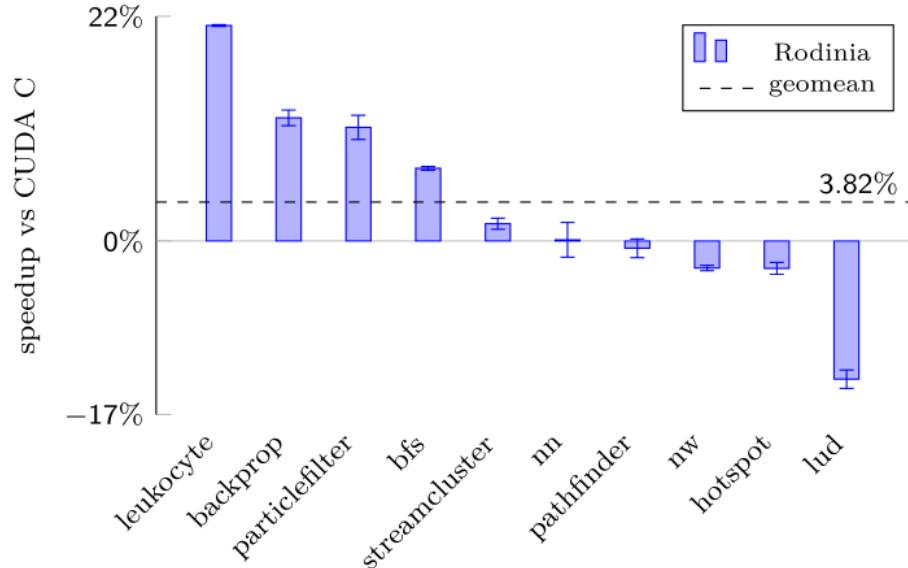
- **Reactive** - when changing a function or variable, Pluto automatically updates all affected cells.
- **Lightweight** - Pluto is written in pure Julia and is easy to install.
- **Simple** - no hidden workspace state; friendly UI.

[JuliaCon 2020 talk](#):

<https://www.youtube.com/watch?v=IAF8DjrQSSk>

Julia on GPUs: <https://juliagpu.org>

Supports NVIDIA GPUs. Nascent support for AMD and Intel GPUs.



Benchmarks compared to CUDA C

Noteworthy new capabilities

- Multi-GPU programming
- Support for CUDA 11 (and CUDA 10 also)
- CUDNN support
- Multi-tasking and multi-threading

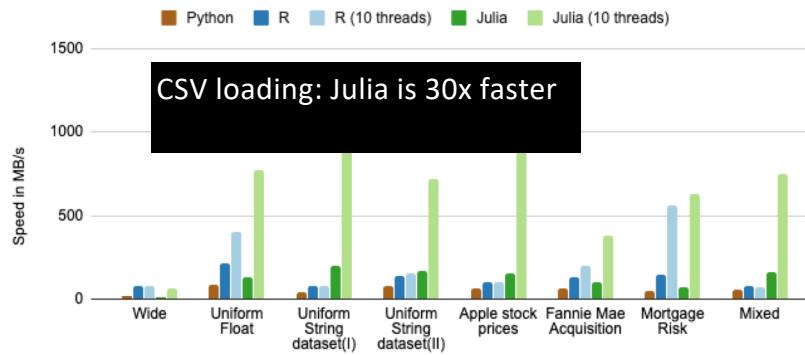
Noteworthy applications

- 300x improvement in pharmaceutical workloads
- 1,000 GPU parallel deployment at CSCS (Switzerland)
- Clima Project – Oceananigans.jl
- Multi-physics simulations
- Reinforcement learning – AlphaZero.jl



Data Science with DataFrames.jl and CSV.jl

CSV reading benchmarks: Python, R, and Julia



Input table: 100,000,000 rows x 9 columns (5 GB)

ClickHouse	20.3.8.53	2020-04-30	7s
data.table	1.12.9	2020-04-22	13s
spark	2.4.5	2020-03-04	36s
DataFrames.jl	0.21.1	2020-05-24	42s
(py)datatable	0.11.0a0	2020-05-24	68s
pandas	1.0.3	2020-05-19	92s
dplyr	0.8.5	2020-03-08	191s
data.table	1.12.9	2020-03-11	305s
H2O DataFrames	0.12.0	2020-03-11	305s
clouddataframe	0.12.0	2020-03-11	305s
MemoryDataFrames	0.12.0	2020-03-11	305s

H2O DataFrames benchmark: GroupBy
Julia is 2x faster

DataFrames.jl

- High performance native Julia package for data manipulation
- GroupBy operations 2x faster than Pandas and R
- [DataFrames.jl status and roadmap](#)

CSV.jl

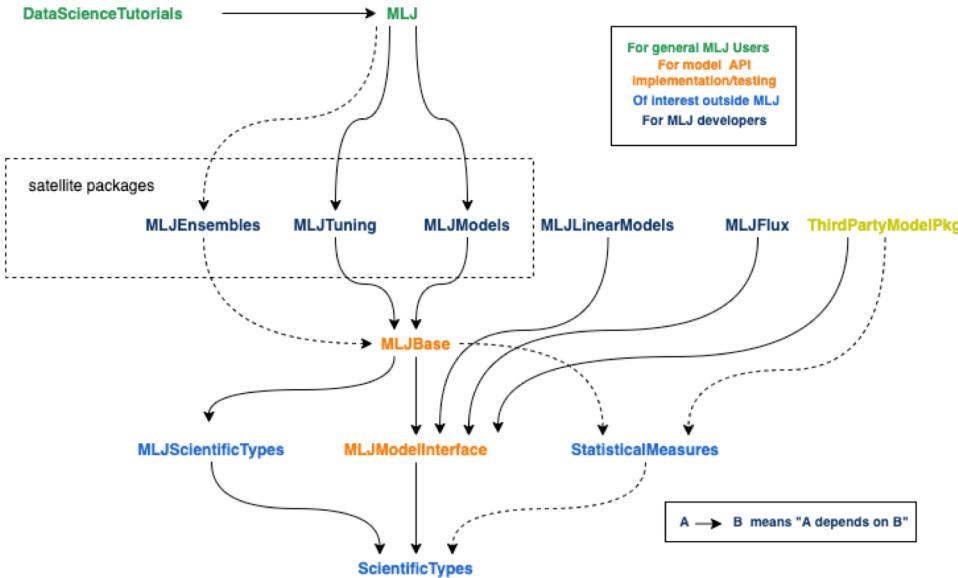
- Native Julia package for loading CSV files
- Significantly faster than Python (pandas) and R
- Multi-threaded
- [The great CSV showdown \(TowardsDataScience\)](#)

of memory pending



Machine Learning with MLJ.jl

The scikit-learn like package in Julia



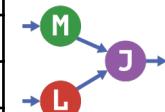
Dependency chart for MLJ repositories. Repositories with dashed connections do not currently exist but are planned/proposed.



[MLJ.jl: <https://github.com/alan-turing-institute/MLJ.jl>](https://github.com/alan-turing-institute/MLJ.jl)

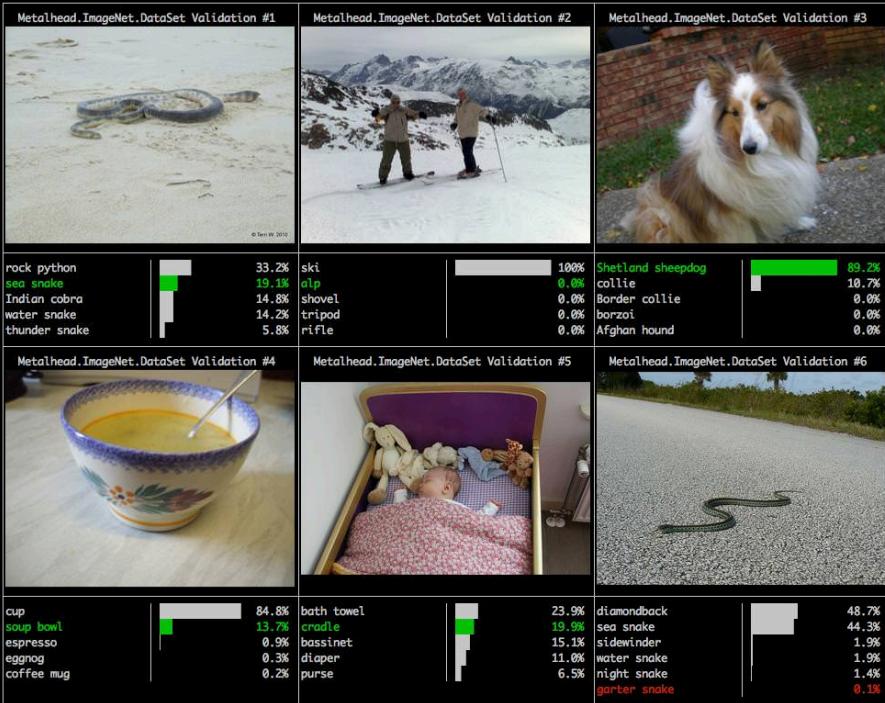
Packages supported by MLJ.jl

Package	Maturity
Clustering.jl	high
DecisionTree.jl	high
EvoTrees.jl	medium
GLM.jl	medium
LIBSVM.jl	high
LightGBM.jl	high
MLJFlux.jl	experimental
MLJLinearModels.jl	experimental
MLJModels.jl (builtins)	medium
MultivariateStats.jl	high
NaiveBayes.jl	experimental
NearestNeighbors.jl	high
ParallelKMeans.jl	experimental
ScikitLearn.jl	high
XGBoost.jl	high



Deep Learning with Flux.jl

<https://fluxml.ai>



using Flux

```
W = rand(2, 5)
b = rand(2)

predict(x) = (W * x) .+ b
loss(x, y) = sum((predict(x) .- y).^2)

x, y = rand(5), rand(2) # Dummy data
l = loss(x, y) # ~ 3

θ = params([W, b])
grads = gradient(() -> loss(x, y), θ)
```

Flux.jl

- Simple Kera—like syntax
- Native Julia implementation
- Easy to look under the hood and modify
- Model Zoo to get started

Pre-trained models

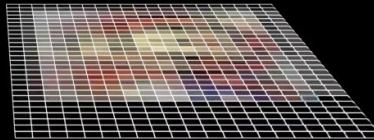
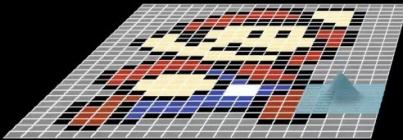
- MetalHead.jl
- YOLO.jl
- Darknet.jl
- ObjectDetector.jl
- Transformers.jl
- TextAnalysis.jl
- GeometricFlux.jl



Image Processing with Images.jl

<https://juliaimages.org>

Convolutions



in image processing

Images.jl

JuliaImages is focussed on a clean architecture and hopes to unify "machine vision" and "biomedical 3d image processing" communities.

- Native Julia based image processing package
- Being used at MIT in a class taught in collaboration with 3blue1brown
- Native Julia datatypes such as RGB.
- Extensive documentation

<https://www.youtube.com/watch?v=DGojl9xcCfg>
<https://www.youtube.com/watch?v=8rrHTtUzyZA>



Optimization

INFORMS Computing Society Prize

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4		8		3				1
7			2					6
	6				2	8		
		4	1	9				5
			8			7	9	

JuMP.jl, Optim.jl

- JuMP is a DSL for mathematical optimization
- Received [INFORMS Computing Society Prize](#)
- Supports over 25 backend solvers
- Optim.jl provides univariate and multivariate optimization in Julia.
- Jump-Dev 2020 is in progress



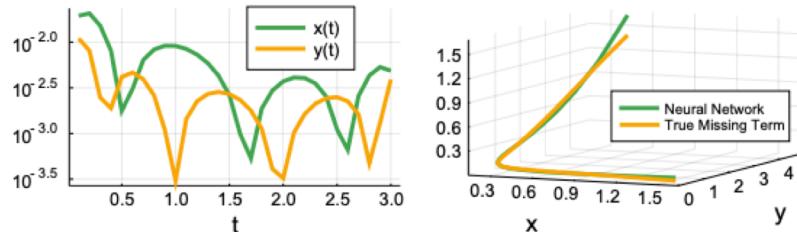
JuMP.jl: <https://github.com/IainNZ/SudokuService>



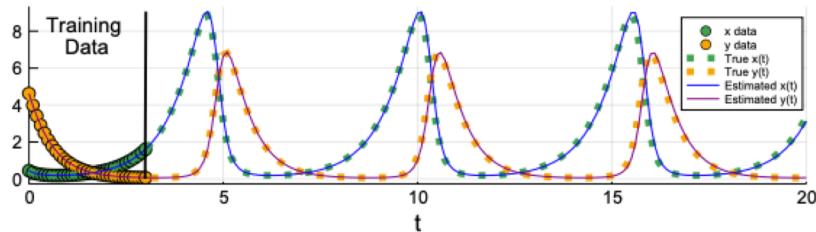
Scientific Machine Learning

<https://sciml.ai>

A Timeseries of UODE Error B Neural Network Fit of U2(t)



C Extrapolated Fit From Short Training Data



Benchmarks compared to CUDA C

Noteworthy new capabilities

- Combine Science and Machine Learning
- Comprehensive Differential Equation Solvers
- GPU acceleration
- MTK.jl: A DSL for modeling and simulation
- Automatic-differentiation

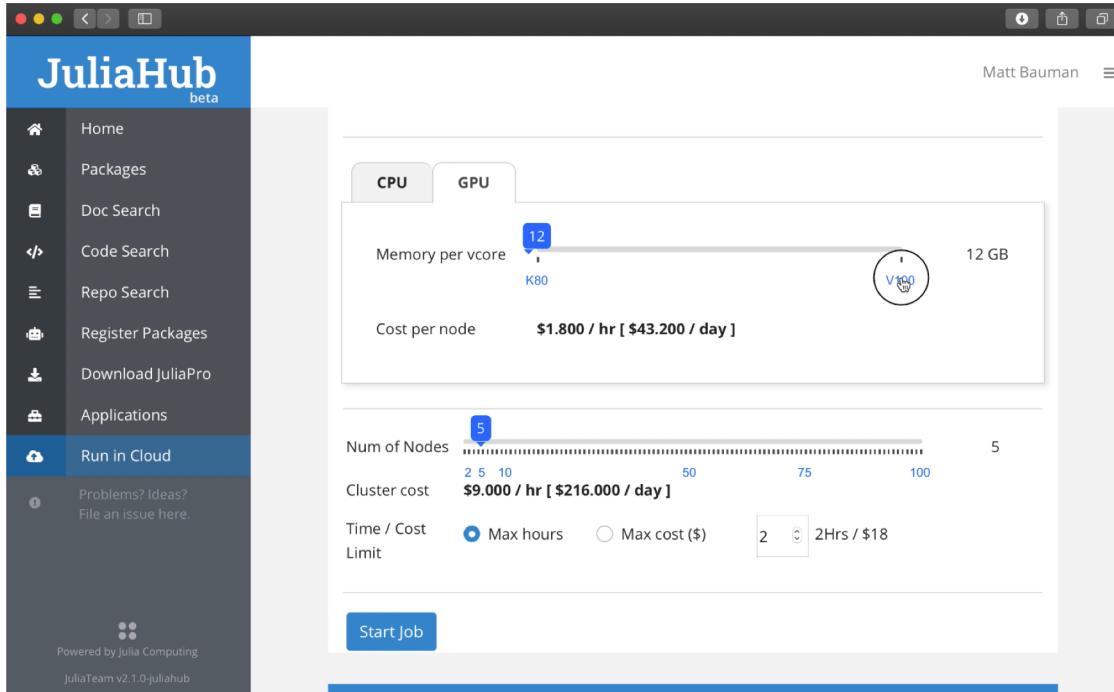
Noteworthy applications

- ACED: Computational Electrochemical Systems
- Clima: Climate Modelling
- NYFed DSGE modelling
- Pumas.jl: Pharma simulations
- MADS: Decision Support System
- QuantumOptics
- Robotics



Pumas + JuliaHub

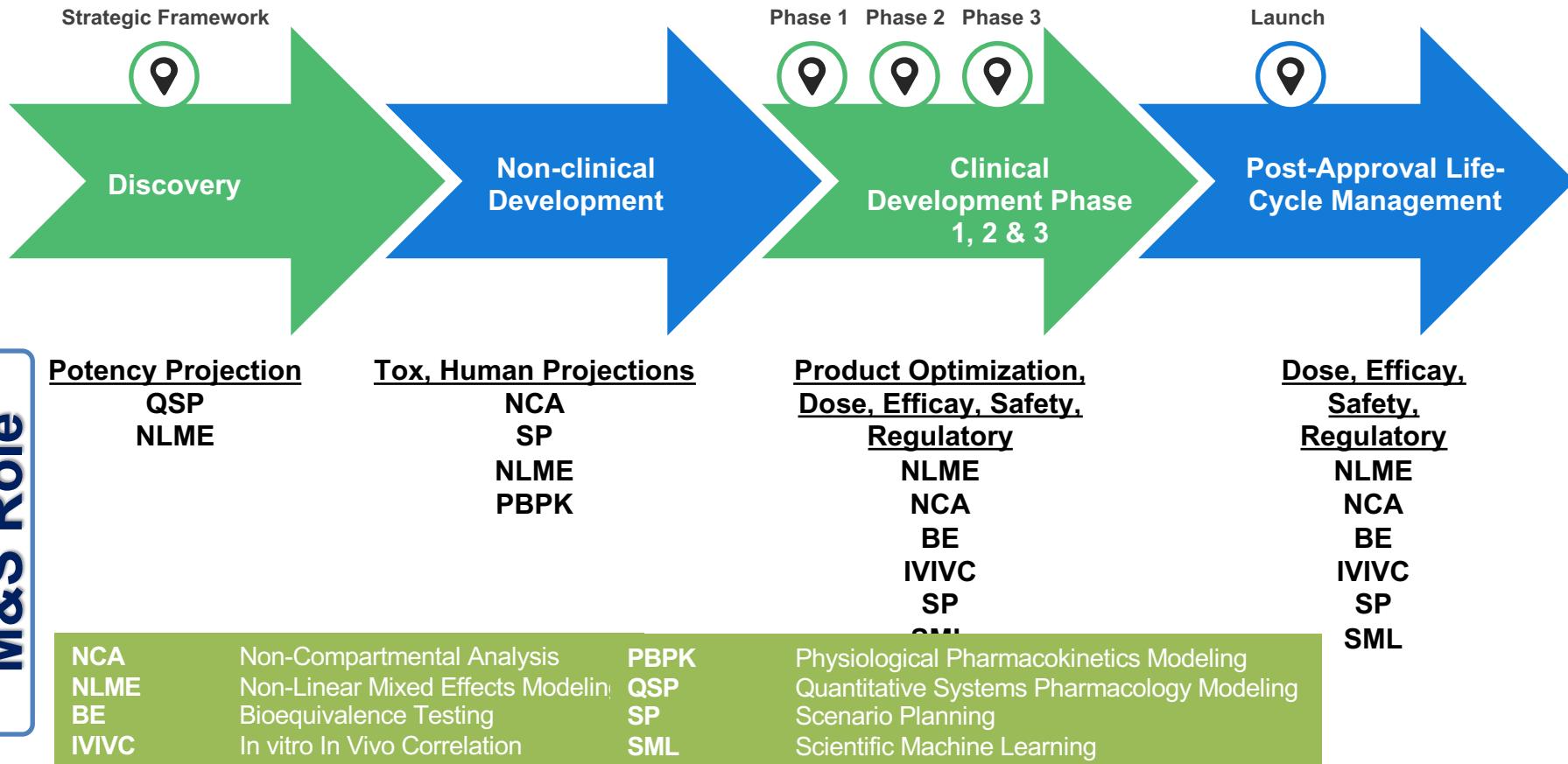
A Cloud Platform for Julia-based Pharmacometric workflows



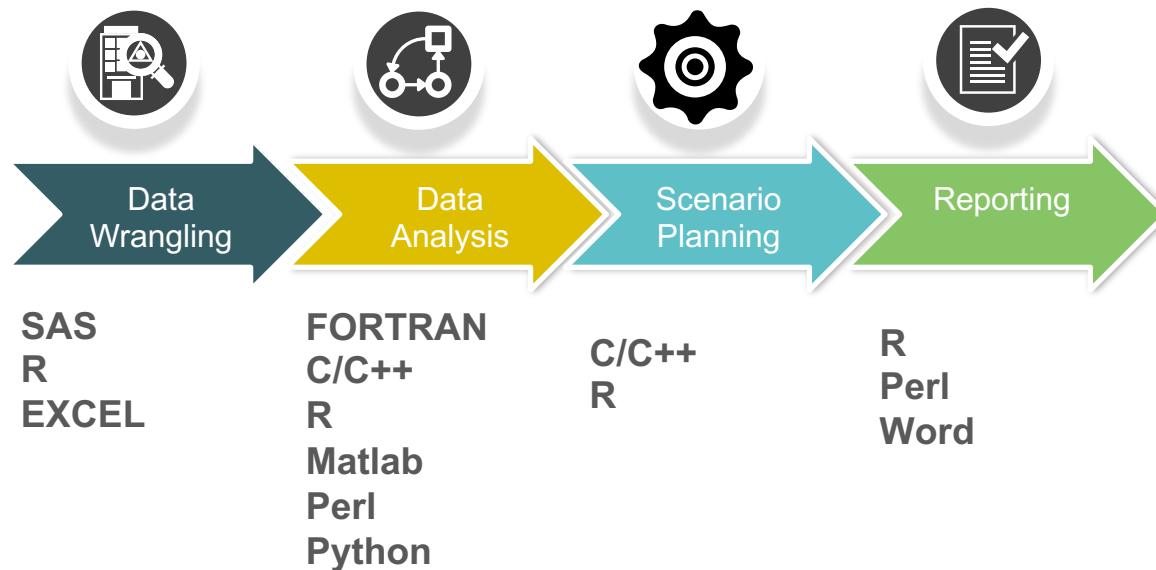
- Single click parallelism
- Integrated right into the IDE (Visual Studio Code)
- Complete browser-based workflow
- Support for polyglot multi-language workflows
- GPU acceleration
- Regulatory compliance

Varied analytics required along the drug development pipeline

M&S Role



High costs for interoperability



Efficient Modeling and Simulation → Accelerated Drug Development



Pumas

100 % Julia



**Julia has excellent interoperability
with other languages, especially R**

Statistics in Julia

- I/O
- CSV.jl
 - DataFrame.jl
 - Tables.jl
 - Query.jl
 - JSON.jl
 - HD5.jl
 - Feather/Arrow

- Core
- StatsBase.jl
 - Distributions.jl
 - Distances.jl

- Classical Inference
- HypothesisTest.jl
 - MixedModels.jl
 - MultivariateStats.jl
 - KernelDensity.jl
 - Bootstrap.jl

- Bayesian/PPL
- Turing.jl
 - Soss.jl
 - AdvancedMH.jl
 - ADvancedHMC.jl

Back to the Future: Lisp as a Base for a Statistical Computing System by Ihaka and Lang (2008)

Julia is actually a Lisp in disguise

Key Highlights

Download Pumas at <https://pumas.ai>

- 450 budding pharmacists trained in Julia and Pumas in 2020 so far
- > 40 pharma have taken up Julia and Pumas for their workflows
- Julia/Pumas has sped up multiscale models by 10-300x
- Julia/Pumas has allowed scientists to run on GPU
- 6 academic groups have shifted their educational programs and use Julia/Pumas along with R in their curriculum

Thank you

Contact us: info@juliacomputing.com



Julia
SURE

Enterprise support and indemnity



Julia
TEAM

Enterprise governance and collaboration



Julia
RUN

Deployment and scalability

Discover
The Power
of Julia



Our products make Julia easy to use,
easy to deploy and easy to scale



Julia
PRO

Seamless development



Pumas

Pharmaceutical modelling and simulation