

Homework 10

due Tue Dec 13 at 11:59pm

(there will be no Homework 11)

Help Sessions

Mon Dec 12 at 3:45pm in Wilson 113
(note room change)

quick tour of high-performance computing

[Vanderbilt's Return to Campus website](#)

[CDC's Coronavirus Disease website](#)

Home About Us ▾ Getting Started ▾ Documentation ▾ Visualization Portal Jupyter Other Services ▾ For PIs ▾ Support ▾

ACCRE

Advanced Computing Center
for Research and Education



Storage issues with /scratch and networked storage

Nov. 20, 2020—At around 10am this morning we received alerts for the /scratch storage sub-system and subsequently for the networked storage sub-system. /scratch unmounted on 57 compute nodes and 20 GPU nodes as well as the gateways. An investigation of all three sub-systems (/data, /scratch, networked storage) showed that a few LUNs were unavailable due to three...

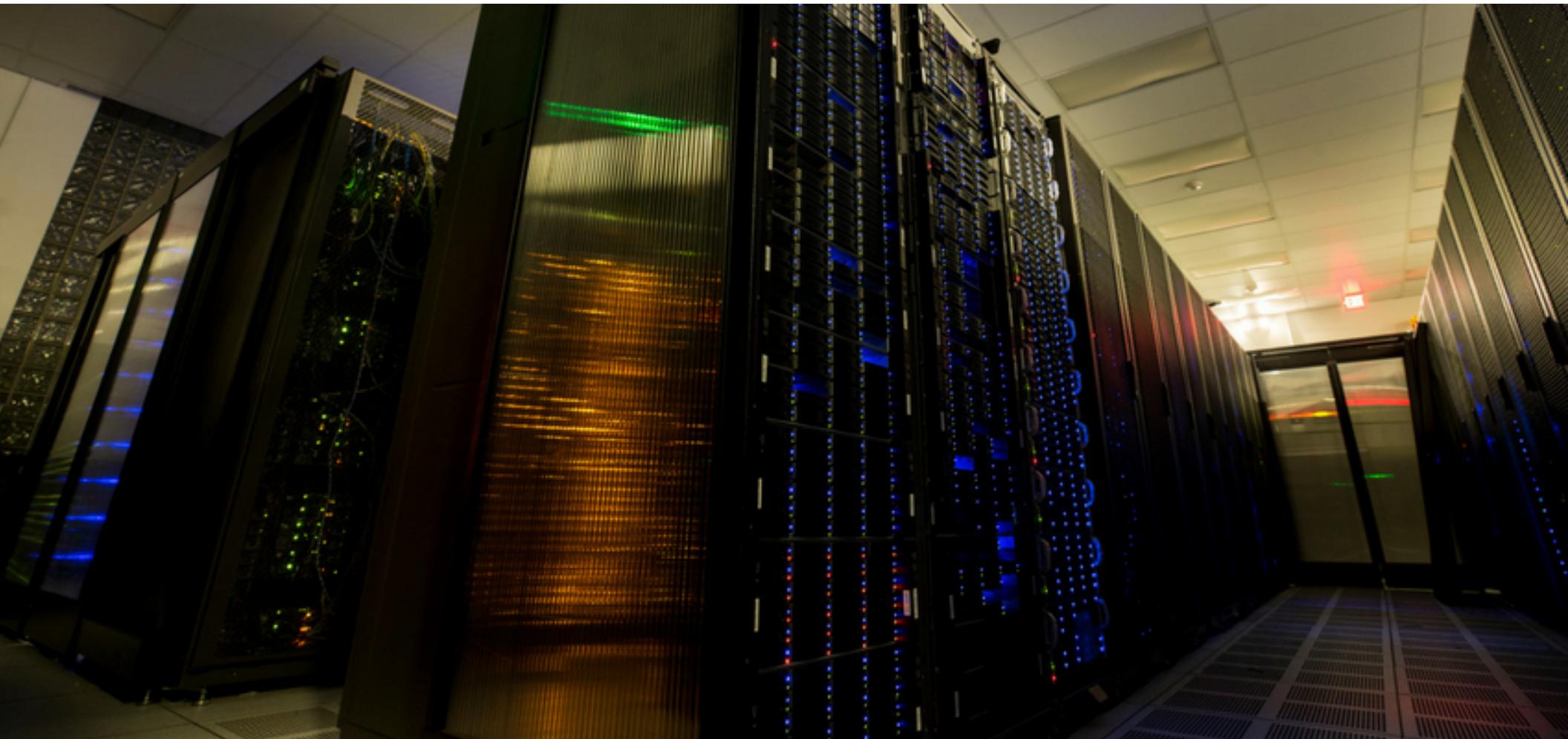
[Read more](#)

Welcome to ACCRE



ACCRE is the premier resource for the high-performance computing needs of research throughout Vanderbilt University. With over 600 multi-core systems in a 4,000 square foot facility, the ACCRE cluster is used for research in a wide range of fields including:

ACCRE

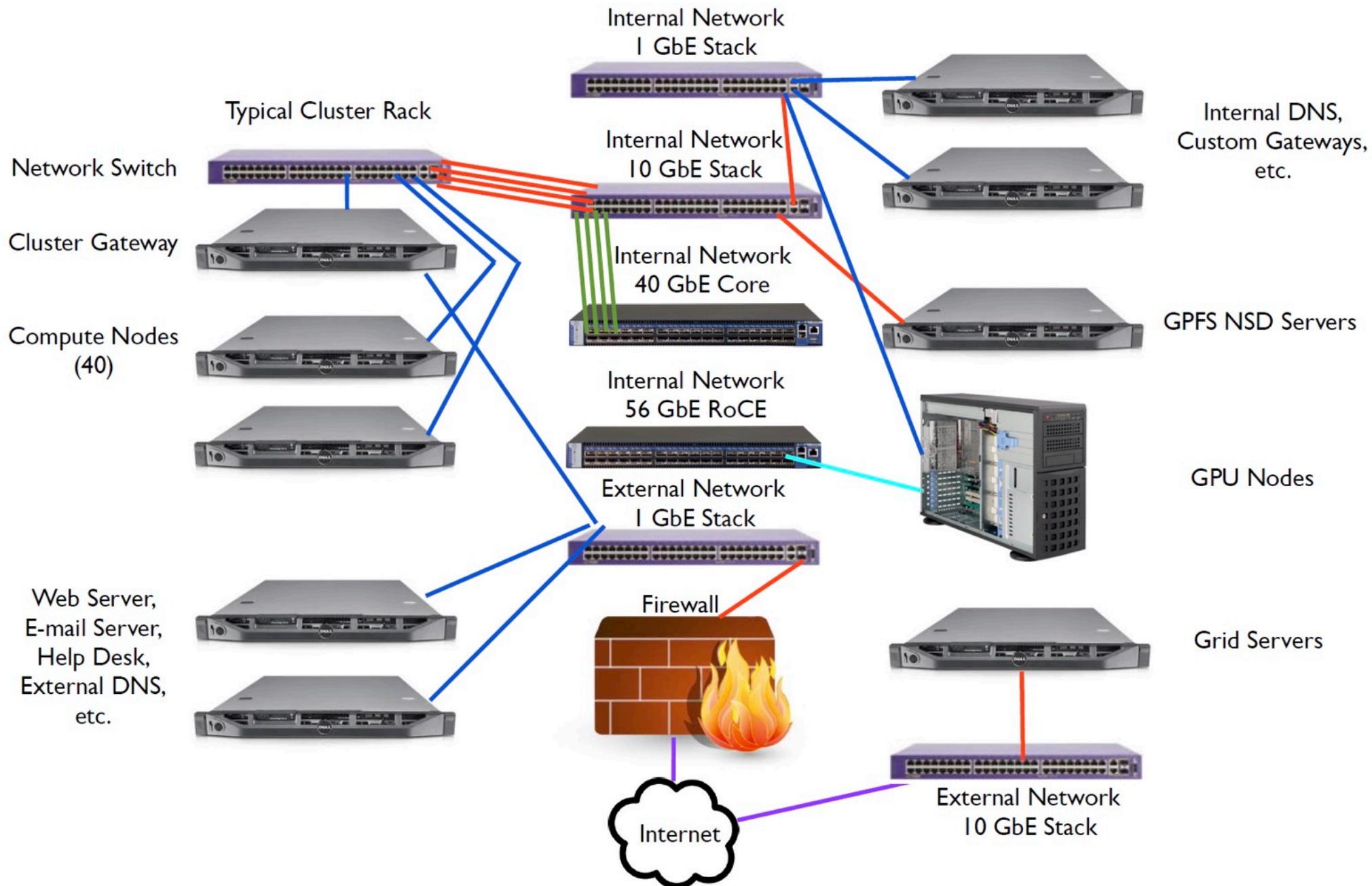


"beowulf" cluster (of commodity-grade computers)

<http://accre.vanderbilt.edu>

ACCRE Cluster Diagram

Blue = Gigabit Ethernet, Red = 10 Gigabit Ethernet, Green = 40 Gigabit Ethernet,
Purple = 100 Gigabit Ethernet, Turquoise = 56 Gigabit RoCE



ACCRe

Last updated February 2022.

| | |
|-----------|--|
| 3,100 | Number of researchers |
| 50 | Number of campus departments and centers |
| 5 | Number of schools |
| 844 | Number of compute nodes |
| ~14,400 | Total number of cores |
| 24-1028GB | Memory available per node |
| 3.8PB | Disk space available through GPFS |
| 130 Gb/s | I/O bandwidth for GPFS |
| 16PB | Disk space available through LStore |



CPU



GPU

ACCRE

GPU cluster

We currently have roughly 54 GPU nodes each equipped with four NVIDIA GPUs for accelerating CUDA and OpenCL enabled applications. The research groups who helped pay for the hardware and support are given the highest priority on these nodes; however, we also allow guest access. Please open a [helpdesk ticket](#) with us to request access.

| Feature | CPUs | Slurm Memory (GB) | GPUs | Count | |
|--------------|------------|-------------------|------------|-----------|--|
| maxwell | 12 | 120 | 4 | 12 | |
| pascal | 8 | 246 | 4 | 22 | |
| turing | 12 | 371 | 4 | 20 | |
| Total | 560 | 14,272 | 216 | 54 | |



CPU



GPU

High-Performance Computing Milestones (1960–2019)**Frontier (Oak Ridge)→**


 1×10^{18}

2019: exaFLOP Barrier To Be Reached?

Summit (Oak Ridge)→**Sunway (China)→****Titan (Oak Ridge)→**


2009: Cray XT5-HE Goes Live

2008: PetaFLOP Barrier Broken

ACCRE (GPU)→**ACCRE (CPU)→**

 1×10^{12}

2005: Millennium Run Simulation

1999: ASCI Blue Pacific Goes Live

2003: Human Genome Mapped

iPhone 8 Plus →

 1×10^9

1984: M-13 Supercomputer

1993: CM-5/1024 Supercomputer


 1×10^6

1976: Cray 1 Goes Live

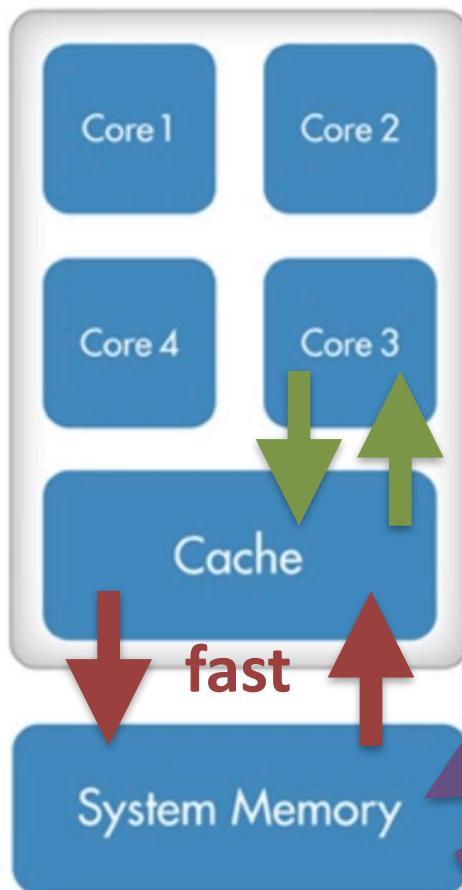

 1×10^3

1960: Univac LARC Goes Live

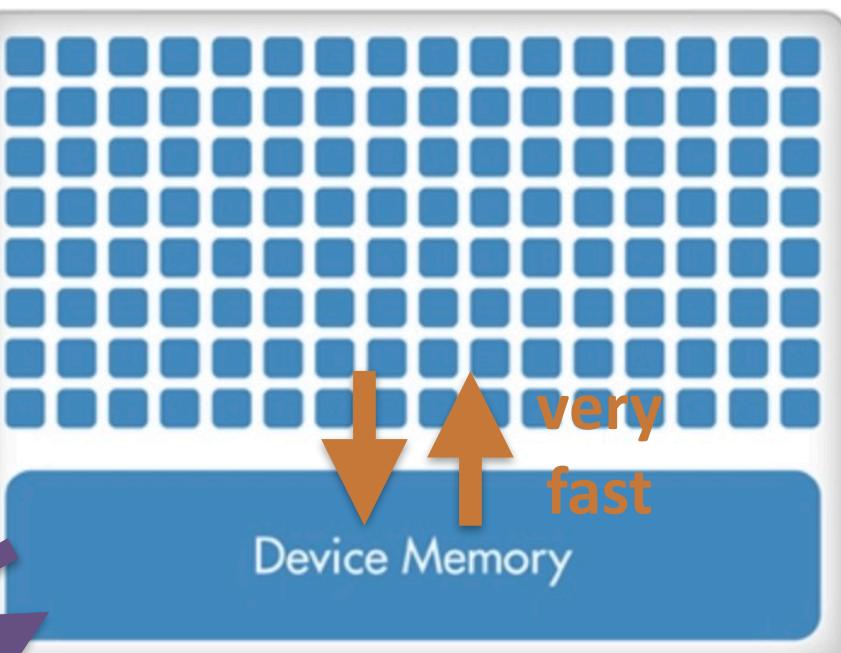
1960 1970 1980 1990 2000 2010 2020

CPU (Multiple Cores)

GPU (Hundreds of Cores)



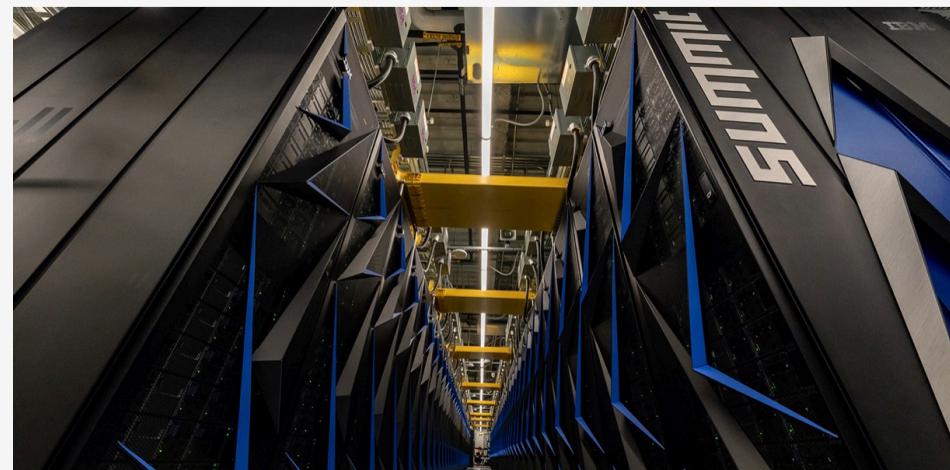
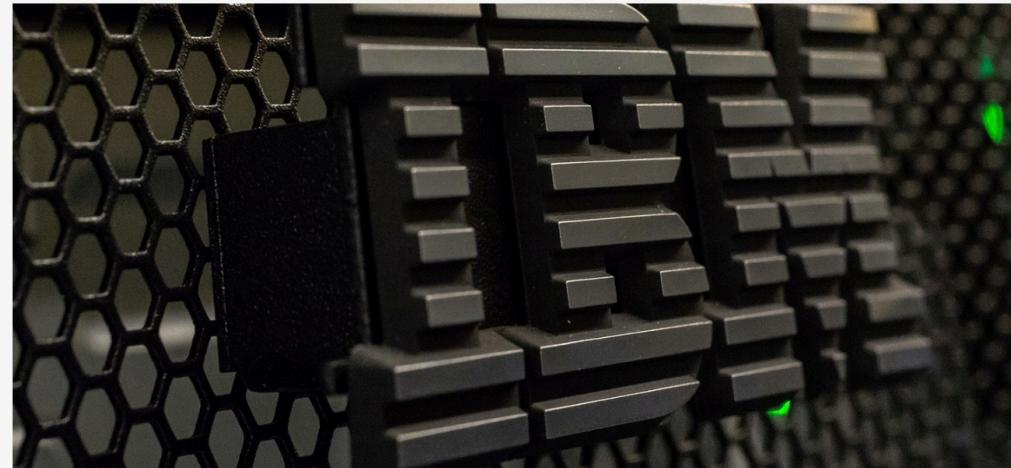
CPU



GPU

SUMMIT @ Oak Ridge
fastest supercomputer in the world (as of November 2018)

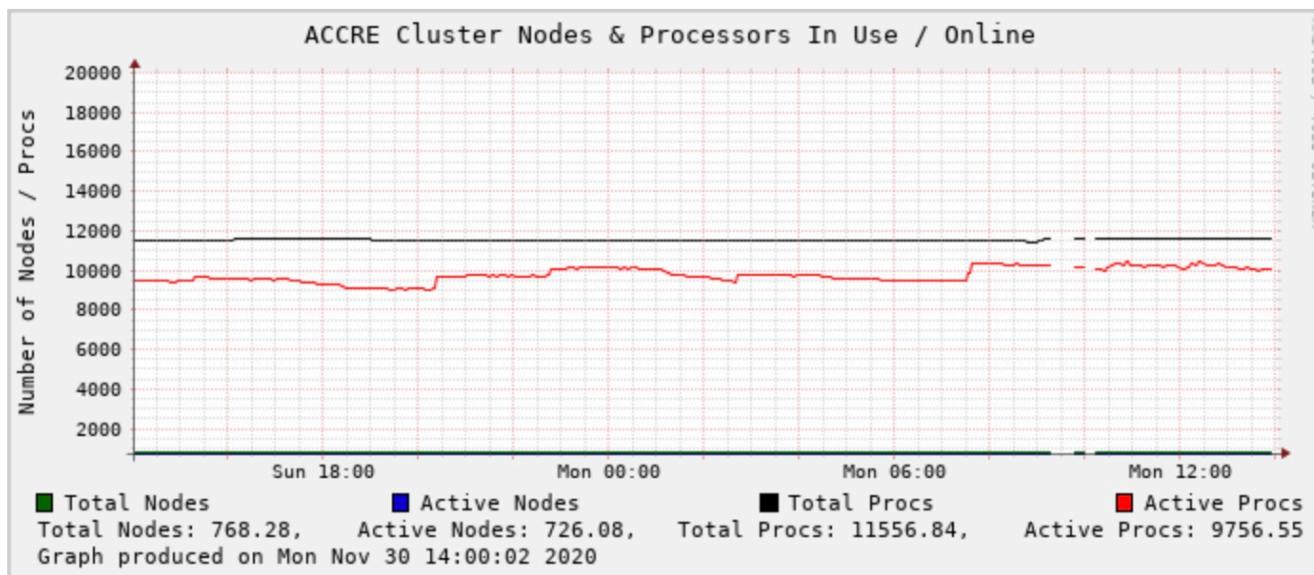
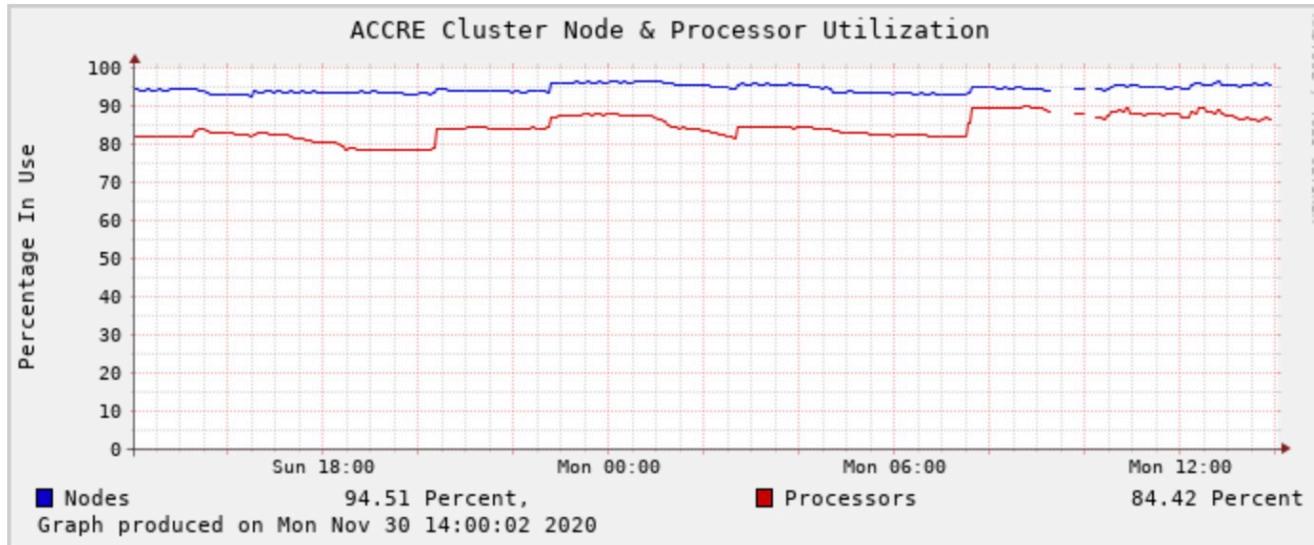
(now #5)



27,000 NVIDIA Volta GPUs
9,000 IBM Power9 CPUs
4608 nodes
1600GB per node (10PB total)
200 PetaFLOPs

ACCRE

80-90% usage

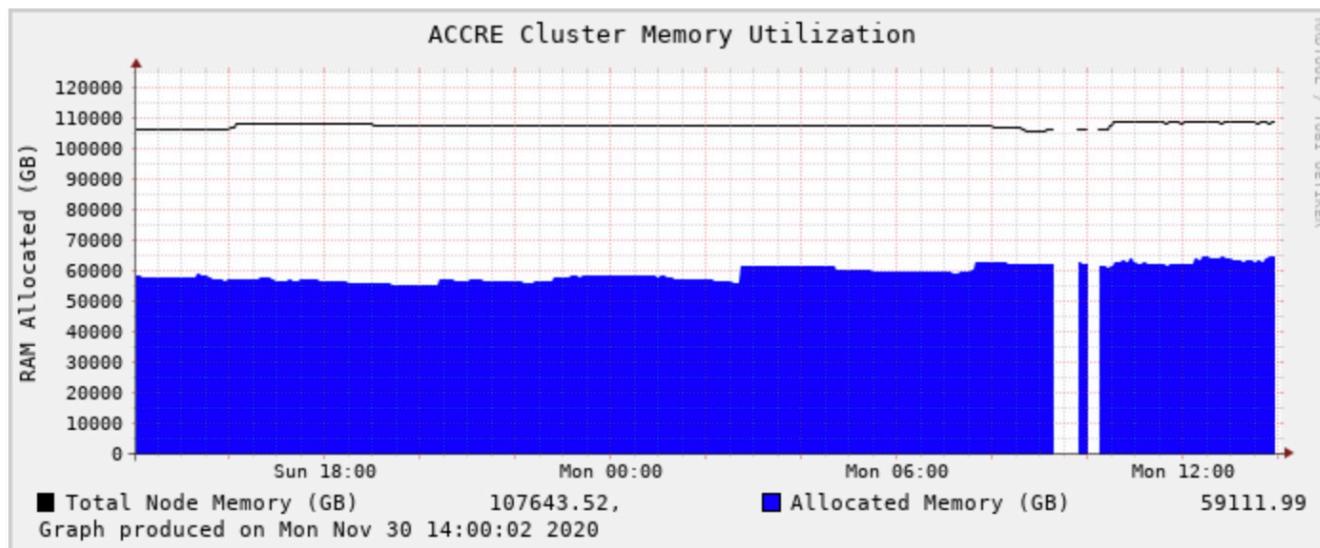
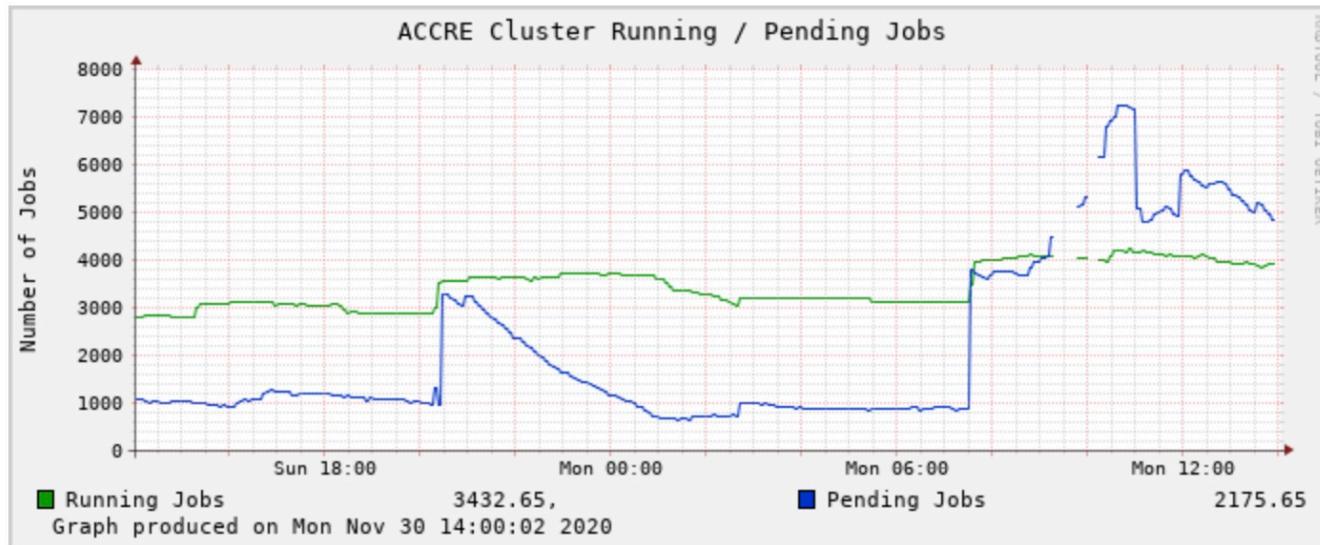


>750 nodes (computers)

12000 computer cores

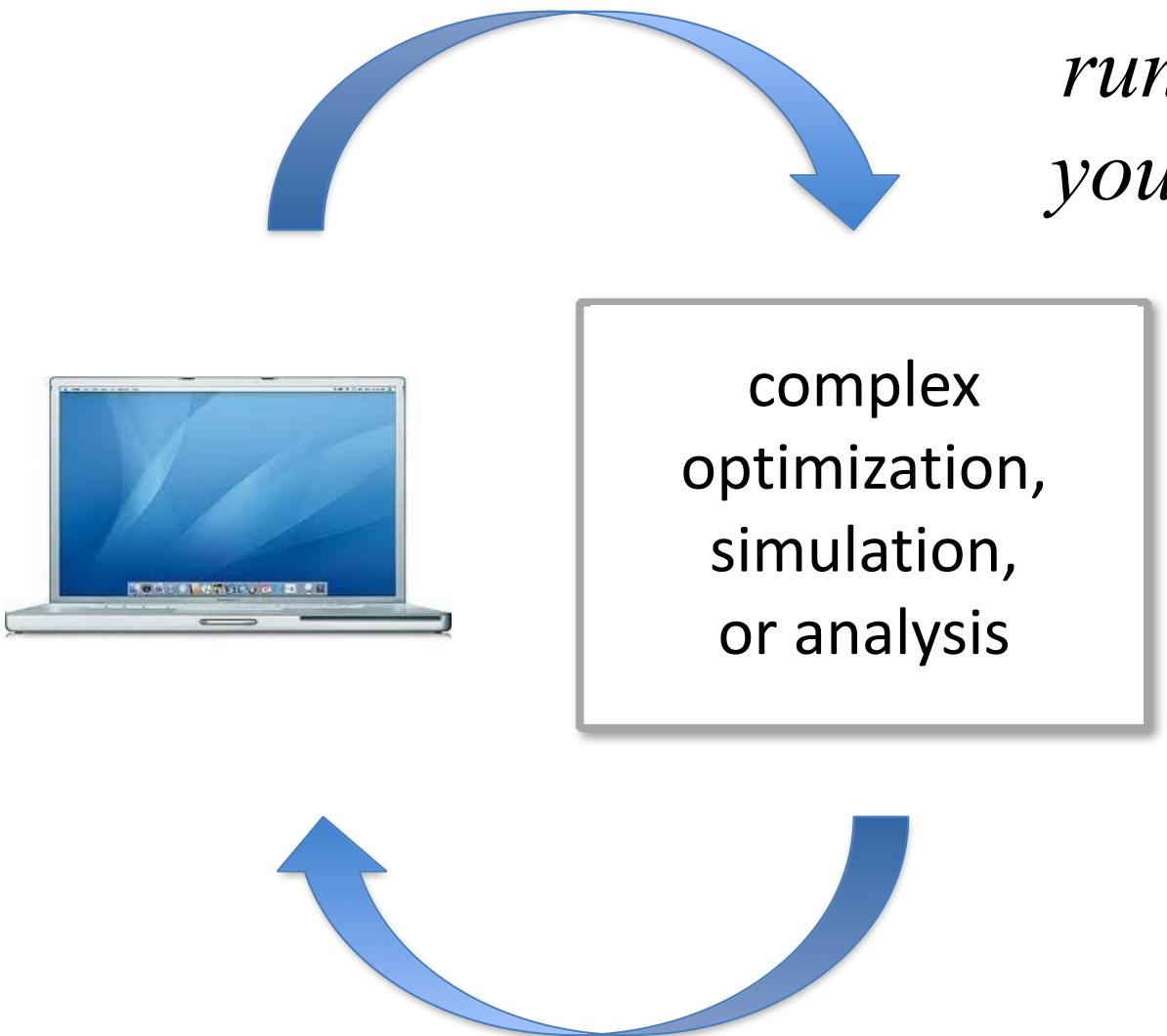
ACCRE

thousands of jobs running and pending



> 100 Petabytes of memory

Could take several weeks



*run locally on
your computer*

**laptop
computer**

*submit as a batch job
to a queue*

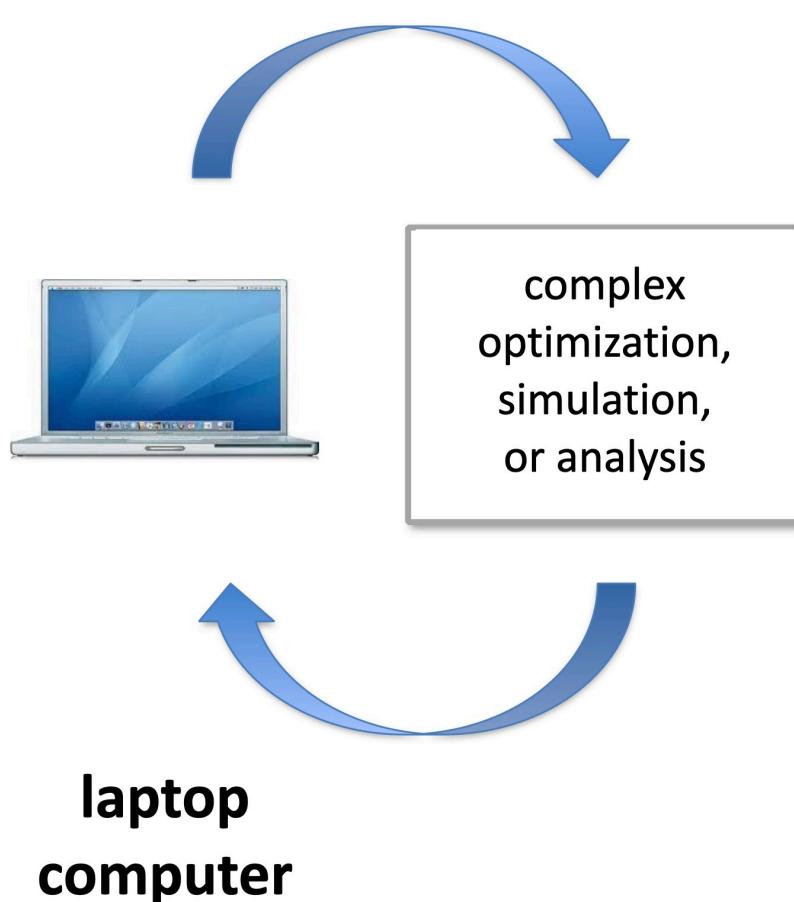
Could take several hours



complex
optimization,
simulation,
or analysis

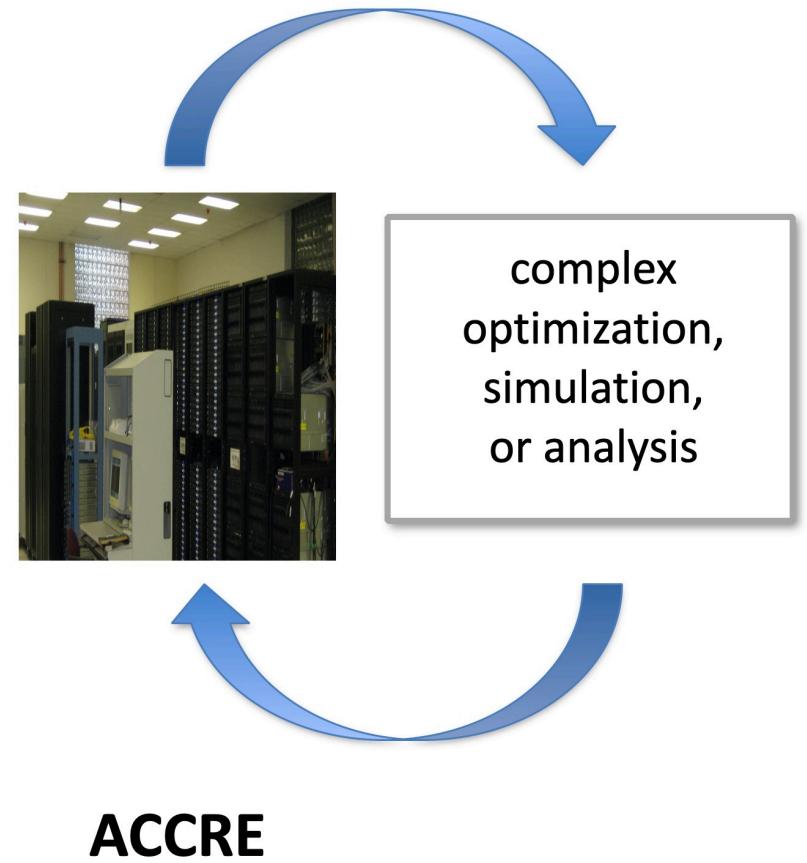
ACCRE

personal computer vs. ACCRE



run one job 1000 times (serial)

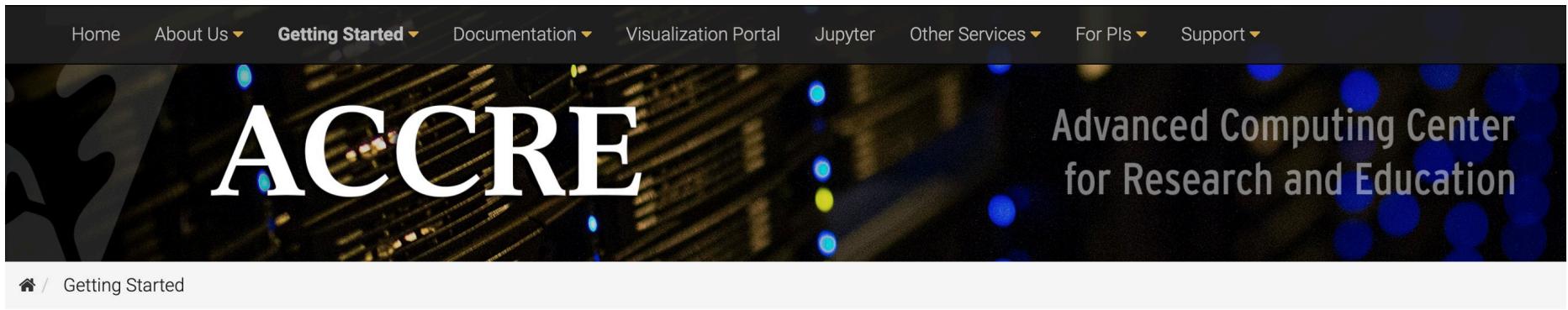
use one GPU



run on 1000 cores at once (parallel)
(truly parallel or stupidly parallel)

use four GPUs

accessing ACCRE



Getting Started



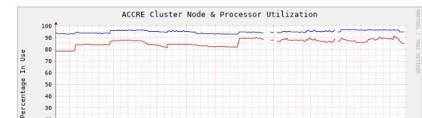
ACCRE offers two computing services for the Vanderbilt community:

- ① **The traditional compute cluster**, which lets users schedule intensive computing jobs to run on ACCRE's compute nodes. The cluster provides access to 200+ software modules.

Getting Started

- Getting Started
- Request a Cluster Account
- First-Time Account Setup
- Required Training
- Submitting Your First Job: One Million Digits of Pi

Cluster utilization, past 24h



<https://www.vanderbilt.edu/accre/getting-started/>

accessing ACCRE

- faculty advisor or lab group (or guest account)
- faculty purchase hardware and pay service fee
or
lease hardware (which includes service fee)
- required to take training courses
<https://www.vanderbilt.edu/accre/getting-started/training/>
- ACCRE used in some courses

<https://www.vanderbilt.edu/accre/getting-started/>

what can run on ACCRE?

- Jupyter notebooks
- Python, R, Matlab, Python, C++, Fortran

<https://www.vanderbilt.edu/accre/getting-started/>

cloud computing

- monthly cost (based on CPU, GPU, memory, disk usage)
or
- institutional subscription



<https://aws.amazon.com>



<https://azure.microsoft.com>



<https://cloud.google.com>



<https://www.ibm.com>

robust, reliable, reproducible
research

Why Economics Is Having a Replication Crisis

Recreating research by gathering data from the real world and analyzing it statistically often fails to produce the same result.

By Noah Smith

September 17, 2018, 8:00 AM CDT

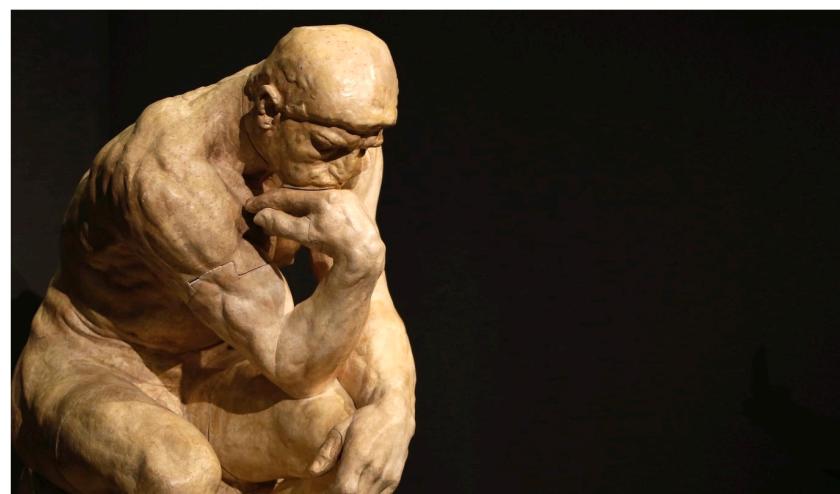


SCIENCE

Psychology's Replication Crisis Is Running Out of Excuses

Another big project has found that only half of studies can be repeated. And this time, the usual explanations fall flat.

ED YONG NOVEMBER 19, 2018



AUGUST 19, 2020

We rely on science. Why is it letting us down when we need it most?

by Stuart Ritchie, Los Angeles Times



future tense

Cancer Research Is Broken

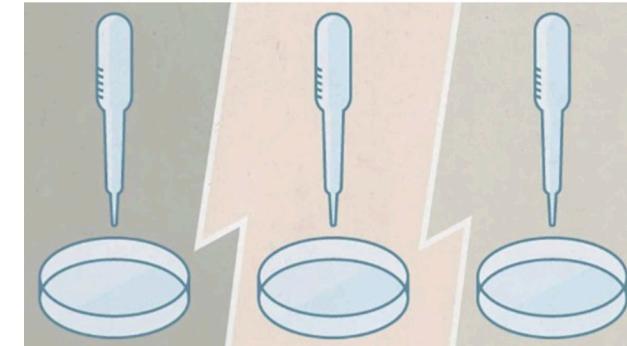
There's a replication crisis in biomedicine—and no one even knows how deep it runs.

By DANIEL ENGBER

SPECIAL | 18 OCTOBER 2018

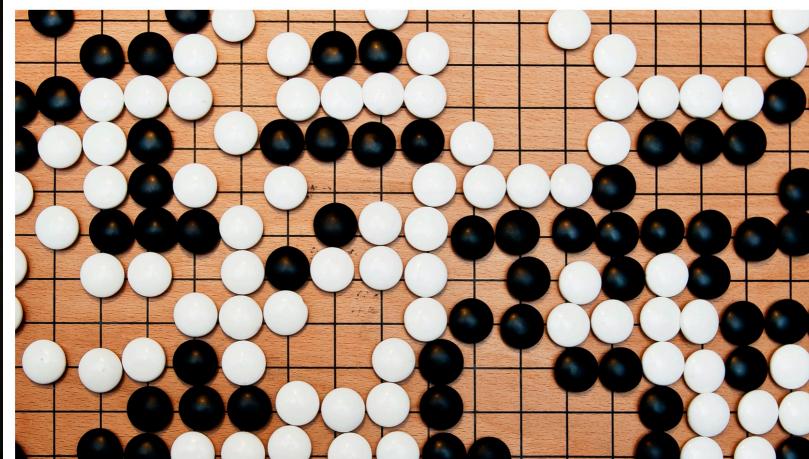
Challenges in irreproducible research

Science moves forward by corroboration – when researchers verify others' results. Science advances faster when people waste less time and money. But as many that do not



Artificial Intelligence Confronts a 'Reproducibility' Crisis

Machine-learning systems are black boxes even to the researchers that build them. That makes it hard for others to assess the results.



Facebook researchers said they found it "very difficult, if not impossible" to reproduce DeepMind's AlphaGo program. PHOTOGRAPH: GETTY IMAGES

role of computing in addressing the crisis

lots of potential reasons for this crisis of confidence,* but better computing practices can play a role in combating it

* including flawed reward structures that favor quantity over quality, lack of statistical training in some disciplines, journals that favor surprising results over theoretically-grounded results, traditionally little weight given to replication, traditionally little incentive or requirement to share details of methods or share data with others

towards robust, reliable, reproducible research

- **Using Notebooks**

a simple step we've talked about: using Jupyter Notebooks (or R Markdown) to document to full data processing, cleaning, analysis, visualization pipeline

(not doing analyses piecemeal, by hand, in spreadsheets, with missing documentation and justification)

towards robust, reliable, reproducible research

- **Literate Programming**

"The idea that you do not document programs (after the fact), but write documents that contain programs.

– John Max Skaller

"The main idea is to regard a program as a communication to human beings rather than as a set of instructions to a computer."

– Donald Knuth

towards robust, reliable, reproducible research

- **Robust Programming Methods**

`try:`

some code here

`except:`

exception handling code here

towards robust, reliable, reproducible research

– Robust Programming Methods

using docstrings

The screenshot shows a PyCharm code editor with the following Python code:

```
# demo docstring in PyCharm

def myfun(a, b, c):
    """
    This is a comment

    :param int a: the first parameter is an integer
    :param float b: the second parameter is an float
    :param str c: the third parameter is a string
    :return float d: the first returned value
    :return str e: the second returned value
    """

    d = a+b
    e = 'This is a string: ' + c
    return (d, e)
```

Below the code, there is a call to `myfun` with parameters `par1`, `par2`, and `par3`. A tooltip window is open over the call, displaying the function's docstring and its parameters:

DocStringDemo
def myfun(a: int,
 b: float,
 c: str) -> Tuple[float, str]

This is a comment

Params: a – the first parameter is an integer
 b – the second parameter is an float
 c – the third parameter is a string

The tooltip also shows parts of the system path and environment variables at the bottom.

<https://www.python.org/dev/peps/pep-0257/>

towards robust, reliable, reproducible research

- **Robust Programming Methods**

using type hints

```
from typing import Tuple
```

```
def myfun(a: int, b: float, c: str) -> Tuple[float, str]:  
    function code here  
    return(d, e)
```

<https://www.python.org/dev/peps/pep-0484/>

<https://docs.python.org/3/library/typing.html>

towards robust, reliable, reproducible research

– Robust Programming Methods

using asserts

7.3. The assert statement

Assert statements are a convenient way to insert debugging assertions into a program:

```
assert_stmt ::= "assert" expression [ "," expression ]
```

The simple form, `assert expression`, is equivalent to

```
if __debug__:  
    if not expression: raise AssertionError
```

The extended form, `assert expression1, expression2`, is equivalent to

```
if __debug__:  
    if not expression1: raise AssertionError(expression2)
```

These equivalences assume that `__debug__` and `AssertionError` refer to the built-in variables with those names. In the current implementation, the built-in variable `__debug__` is `True` under normal circumstances, `False` when optimization is requested (command line option `-O`). The current code generator emits no code for an assert statement when optimization is requested at compile time. Note that it is unnecessary to include the source code for the expression that failed in the error message; it will be displayed as part of the stack trace.

Assignments to `__debug__` are illegal. The value for the built-in variable is determined when the interpreter starts.

https://docs.python.org/3/reference/simple_stmts.html#grammar-token-assert-stmt

towards robust, reliable, reproducible research

- **Robust Programming Methods**

using asserts

```
x = 1
```

use for debugging purposes (raise AssertionError)

```
assert x == 2
```

```
assert x == 2  
AssertionError
```

```
assert type(x) == str
```

```
assert type(x) == str  
AssertionError
```

towards robust, reliable, reproducible research

- Version Control

using Git/GitHub (other other versioning platforms)

The screenshot shows a GitHub repository page for 'VandyCatLab / Stepping-Ramping'. The top navigation bar includes 'Search or jump to...', 'Pull requests', 'Issues', 'Marketplace', 'Explore', and user icons. Below the header, a banner reads 'Learn Git and GitHub without any code!' with a 'Read the guide' button. The main repository area shows a list of commits from 'hansonci':

| Author | Commit Message | Date |
|----------|--|-------------------|
| hansonci | added visualizations to diffusion and starting diffusion to thinning ... | ba99cae on Aug 23 |
| hansonci | add chapter about simulations with thinning | 5 months ago |
| hansonci | Delete filler | 6 months ago |
| hansonci | added visualizations to diffusion and starting diffusion to thinning ... | 3 months ago |
| hansonci | Create Link to Hugh_Wilson_Spikes.pdf | 6 months ago |
| hansonci | Create Link to PillowLabStepRampMCMC | 6 months ago |
| hansonci | updated from 7-9 group meeting | 5 months ago |

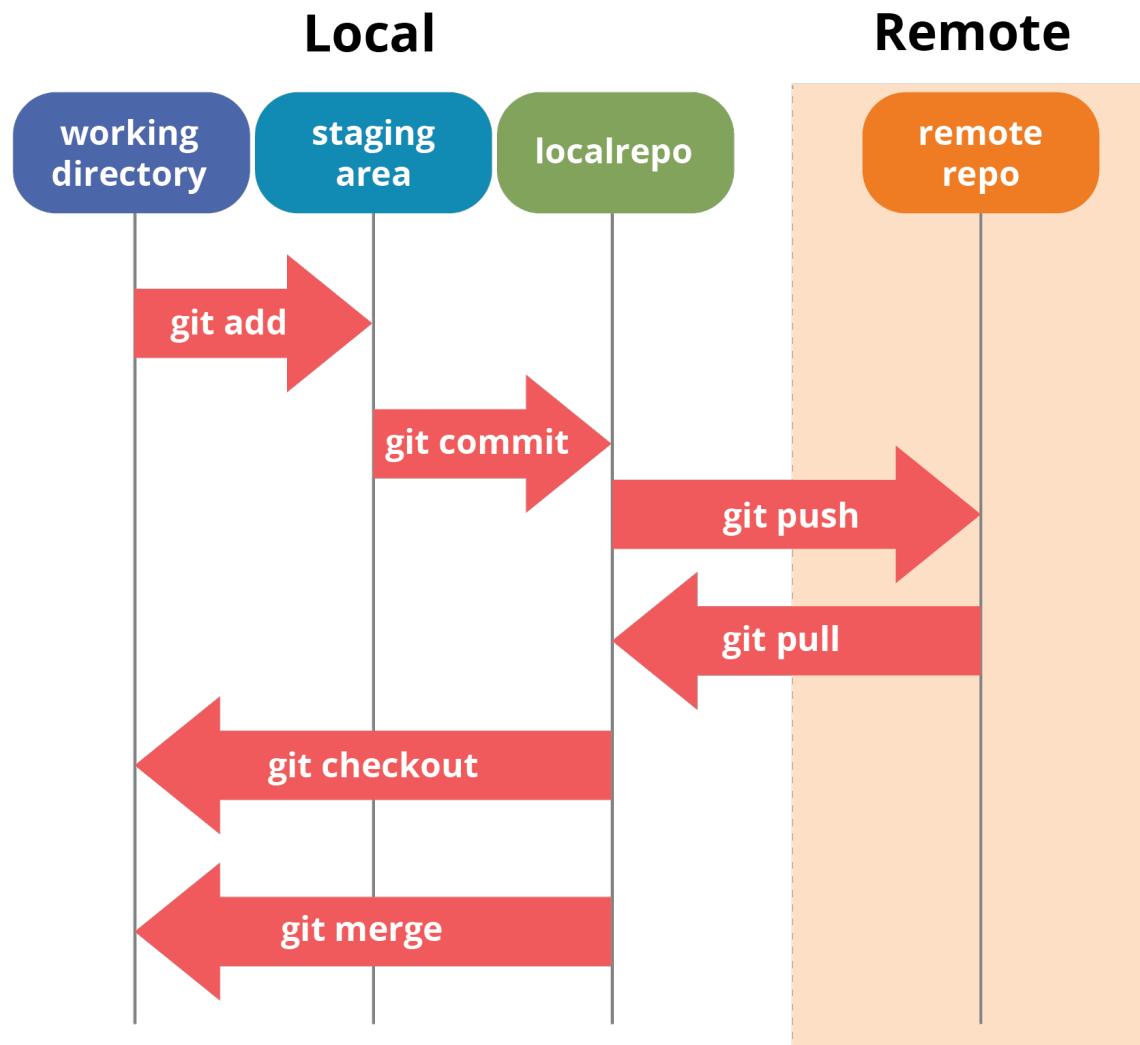
On the right side, there are sections for 'About', 'Modeling neurons', 'Custom notification settings' (with a 'Got it' button), 'Releases' (no releases published), and 'Packages' (no packages published). A call-to-action at the bottom encourages adding a README.

<https://github.com>

towards robust, reliable, reproducible research

- **Version Control**

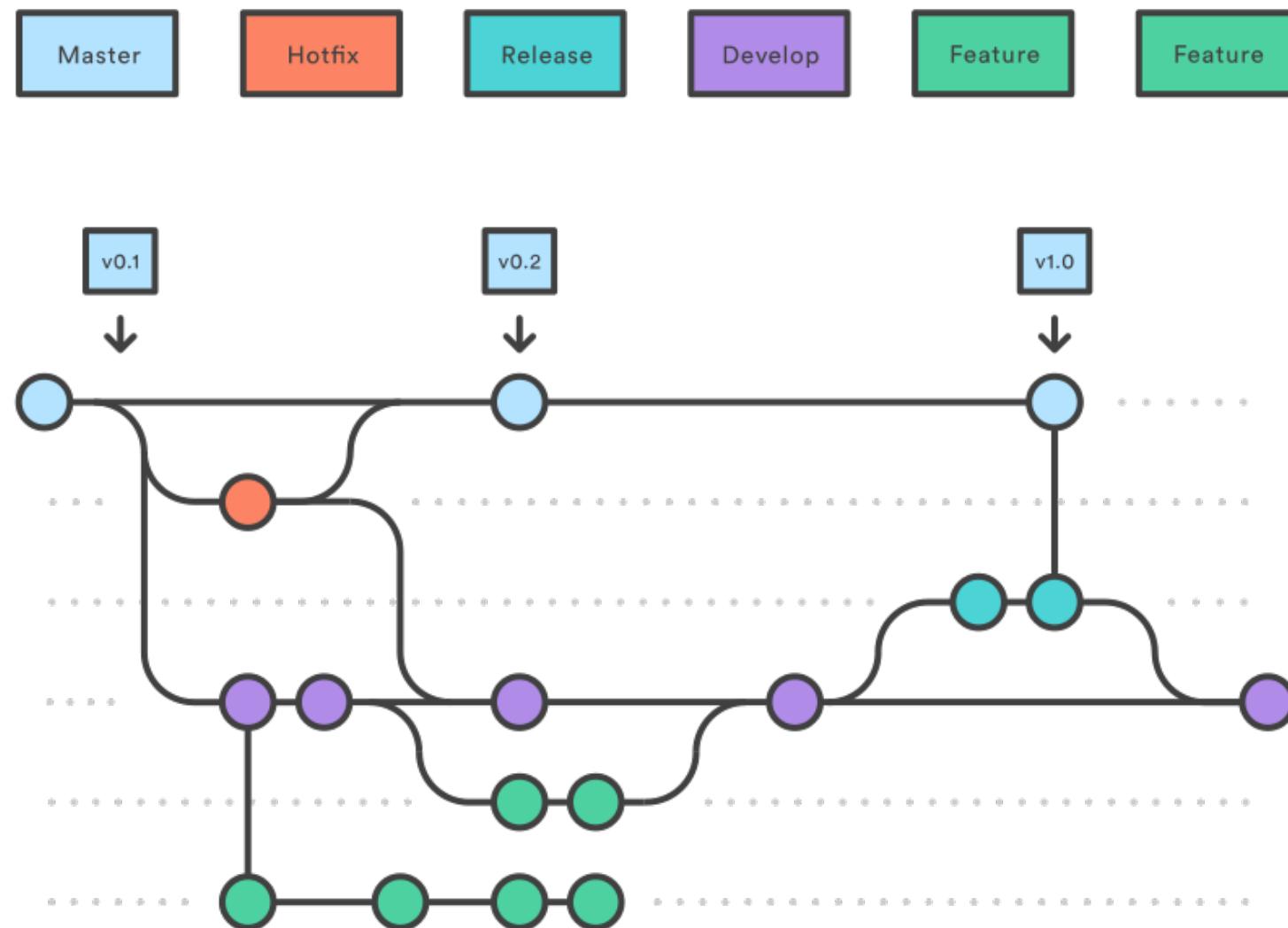
using Git/GitHub (other other versioning platforms)



towards robust, reliable, reproducible research

- **Version Control**

using Git/GitHub (other other versioning platforms)

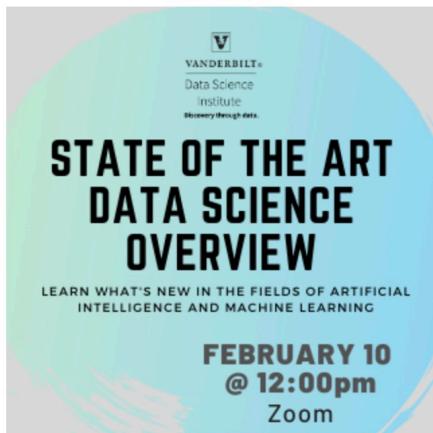


towards robust, reliable, reproducible research

- Version Control

learning about Git/GitHub

covered in summer DSi-SRP



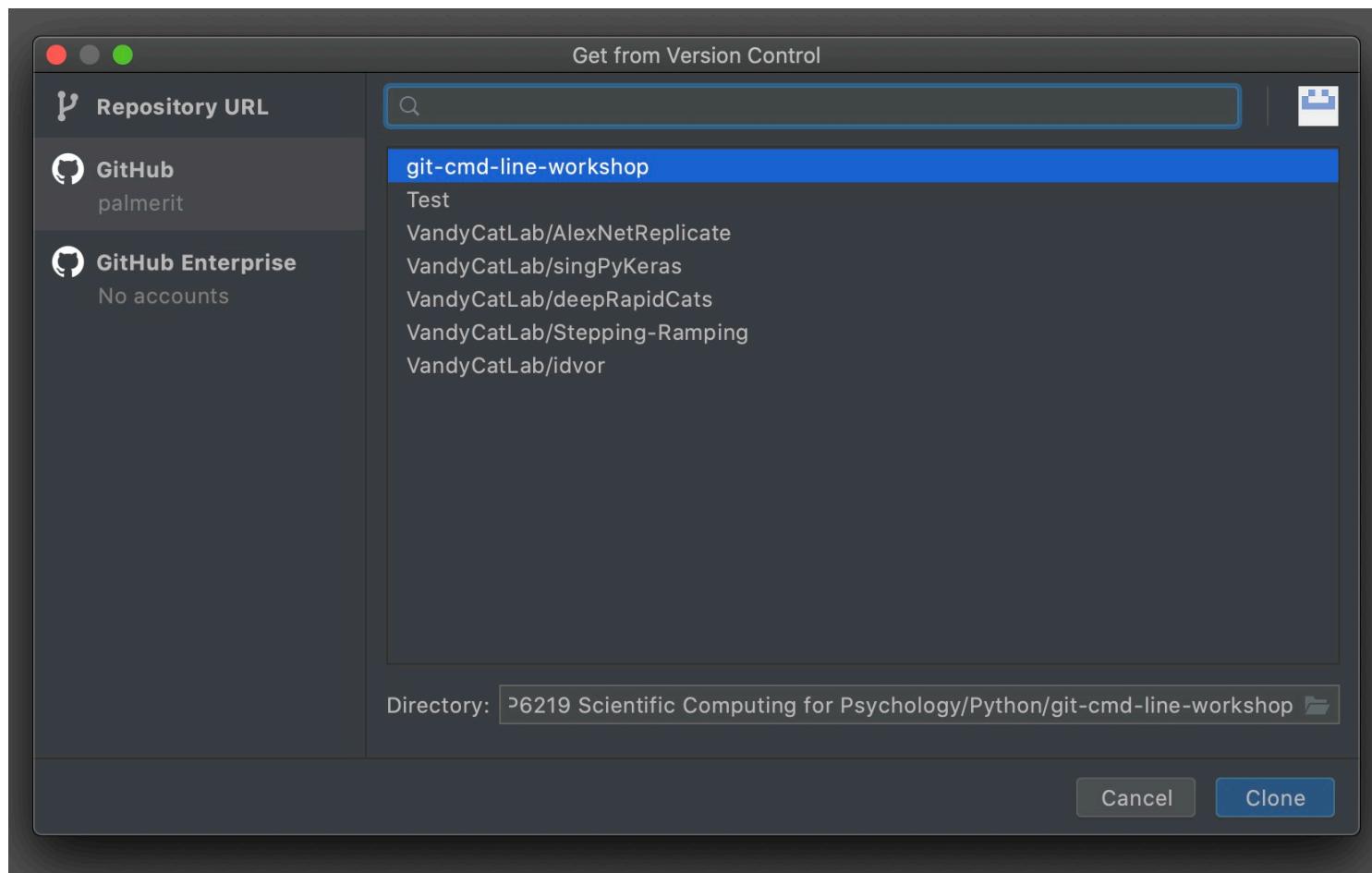
access recorded YouTube videos from workshops

<https://www.vanderbilt.edu/datascience/events/data-science-workshops/>

towards robust, reliable, reproducible research

- Version Control

can access GitHub from within PyCharm



<https://www.jetbrains.com/help/pycharm/set-up-a-git-repository.html>

towards robust, reliable, reproducible research

– **Code Sharing (Open Science)**

public repositories (on Git)

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

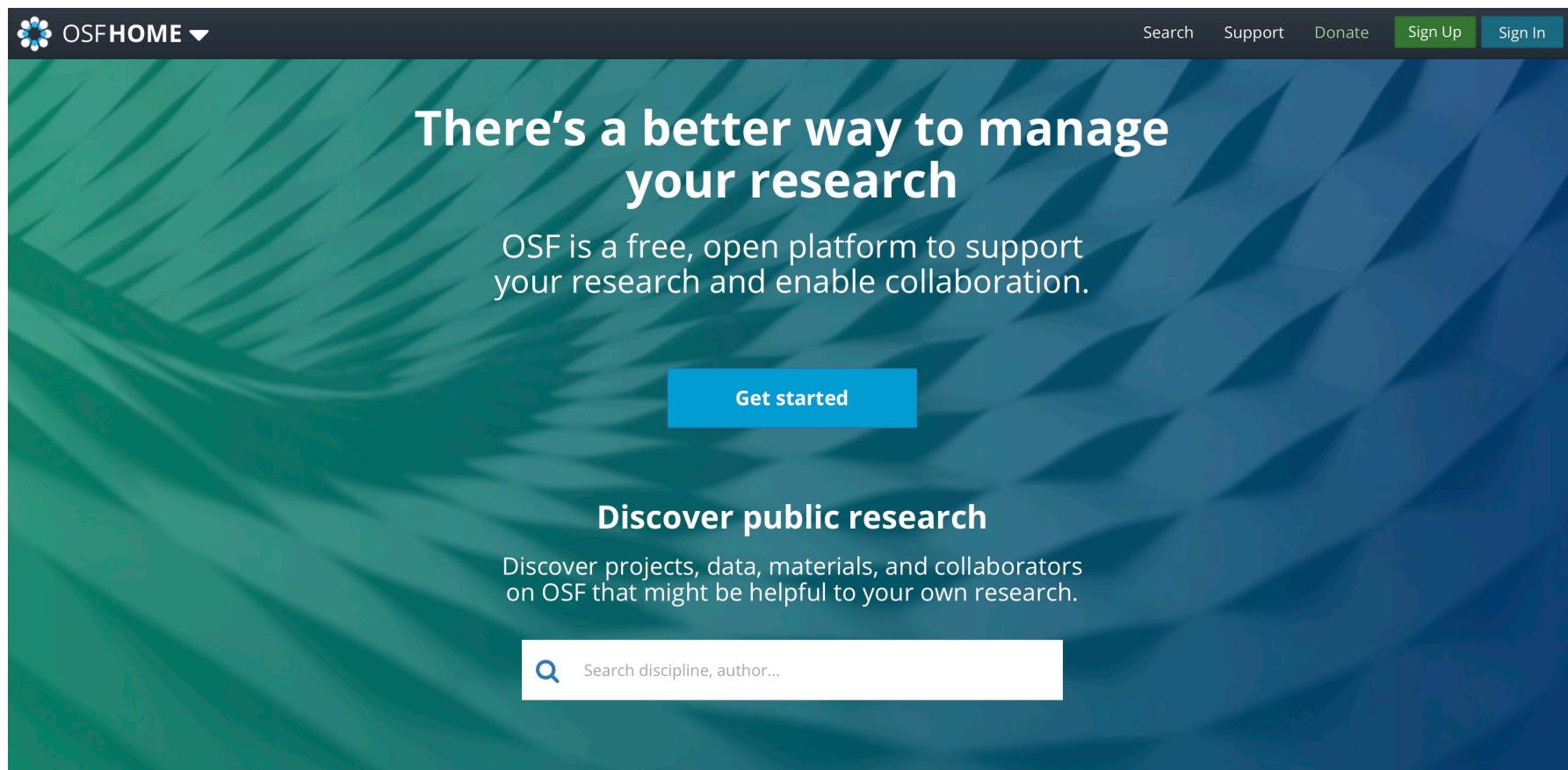
[Read the guide](#)

The screenshot shows the GitHub repository page for `scikit-learn / scikit-learn`. At the top, there's a green banner with the text "Learn Git and GitHub without any code!" and a "Read the guide" button. Below the banner, the repository header includes icons for Sponsor, Watch, Star, Fork, and the count of 2.2k, 43.3k, and 20.7k respectively. The main content area shows the "Code" tab selected, with a dropdown menu showing "master". It displays 23 branches and 113 tags. A list of recent commits is shown, starting with a commit from `ogrissel` that removes a GitHub workflow file. Other commits include updates to .binder, .circleci, and .github directories, as well as benchmarks and build tools. On the right side, there's a sidebar with "Custom notification settings" (with a "Got it" button), a link to `scikit-learn.org`, and sections for "machine-learning", "python", "statistics", "data-science", and "data-analysis". There are also links for "Readme", "BSD-3-Clause License", and "Releases" (113).

towards robust, reliable, reproducible research

- **Code Sharing (Open Science)**

public repositories (on OSF - Open Science Foundation)



The screenshot shows the homepage of the Open Science Foundation (OSF). At the top, there is a dark navigation bar with the OSF logo, "OSF HOME" with a dropdown arrow, and links for "Search", "Support", "Donate", "Sign Up" (in a green button), and "Sign In". The main background features a gradient from teal to blue with a subtle leaf pattern. The central message is "There's a better way to manage your research" in white text. Below it, a sub-message reads: "OSF is a free, open platform to support your research and enable collaboration." A blue "Get started" button is positioned below the sub-message. Further down, there is a section titled "Discover public research" with the subtext: "Discover projects, data, materials, and collaborators on OSF that might be helpful to your own research." At the bottom, there is a search bar with a magnifying glass icon and the placeholder text "Search discipline, author...".

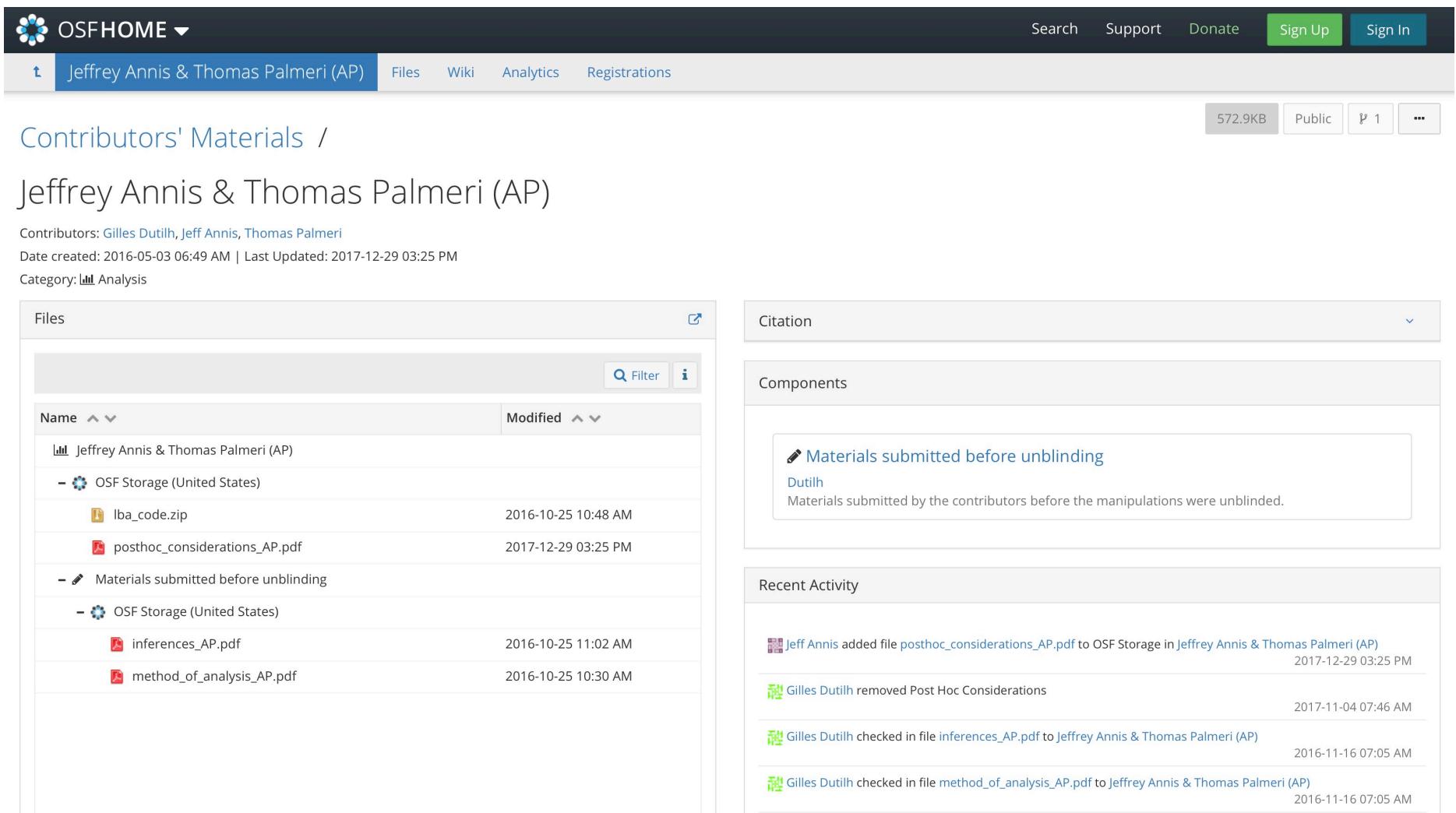
How OSF supports your research

<https://osf.io>

towards robust, reliable, reproducible research

- **Code Sharing (Open Science)**

public repositories (on OSF - Open Science Foundation)



The screenshot shows a project page on the OSF platform. At the top, there's a navigation bar with links for 'Search', 'Support', 'Donate', 'Sign Up', and 'Sign In'. Below the navigation, the project title 'Jeffrey Annis & Thomas Palmeri (AP)' is displayed, along with links for 'Files', 'Wiki', 'Analytics', and 'Registrations'. The main content area shows the project's contributors' materials. On the left, under 'Files', there's a table listing various files: 'lba_code.zip' (modified 2016-10-25 10:48 AM), 'posthoc_considerations_AP.pdf' (modified 2017-12-29 03:25 PM), and two files from 'OSF Storage (United States)': 'inferences_AP.pdf' (modified 2016-10-25 11:02 AM) and 'method_of_analysis_AP.pdf' (modified 2016-10-25 10:30 AM). On the right, there are sections for 'Citation' (with a link to 'Materials submitted before unblinding' by Dutilh), 'Components' (listing 'Materials submitted before unblinding' by Dutilh), and 'Recent Activity' (listing actions like adding and checking in files by users Jeff Annis and Gilles Dutilh).

Contributors: [Gilles Dutilh](#), [Jeff Annis](#), [Thomas Palmeri](#)

Date created: 2016-05-03 06:49 AM | Last Updated: 2017-12-29 03:25 PM

Category: [Analysis](#)

| Name | Modified |
|-------------------------------|---------------------|
| lba_code.zip | 2016-10-25 10:48 AM |
| posthoc_considerations_AP.pdf | 2017-12-29 03:25 PM |
| OSF Storage (United States) | |
| inferences_AP.pdf | 2016-10-25 11:02 AM |
| method_of_analysis_AP.pdf | 2016-10-25 10:30 AM |

Citation

Components

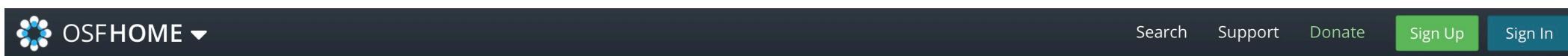
Recent Activity

- Jeff Annis added file [posthoc_considerations_AP.pdf](#) to OSF Storage in [Jeffrey Annis & Thomas Palmeri \(AP\)](#) 2017-12-29 03:25 PM
- Gilles Dutilh removed Post Hoc Considerations 2017-11-04 07:46 AM
- Gilles Dutilh checked in file [inferences_AP.pdf](#) to [Jeffrey Annis & Thomas Palmeri \(AP\)](#) 2016-11-16 07:05 AM
- Gilles Dutilh checked in file [method_of_analysis_AP.pdf](#) to [Jeffrey Annis & Thomas Palmeri \(AP\)](#) 2016-11-16 07:05 AM

towards robust, reliable, reproducible research

– **Code Sharing (Open Science)**

pre-registered research



Improve your research with [preregistration](#). By writing out specific details such as data collection methods, analysis plans, and rules for data exclusion, you can make important decisions early on and have a clear record of these choices. This can help reduce biases that occur once the data are in front of you.

Use [OSF Registries](#) to discover previously registered work.

Preregister

This website relies on cookies to help provide a better user experience. By clicking Accept or continuing to use the site, you agree. For more information, see our [Privacy Policy](#) and information on [cookie use](#).

Accept

Copyright © 2011-2020 Center for Open Science | [Terms of Use](#) | [Privacy Policy](#) | [Status](#) | [API](#)
[TOP Guidelines](#) | [Reproducibility Project: Psychology](#) | [Reproducibility Project: Cancer Biology](#)



