RSPH HPC Cheat Sheet

Allison Codi

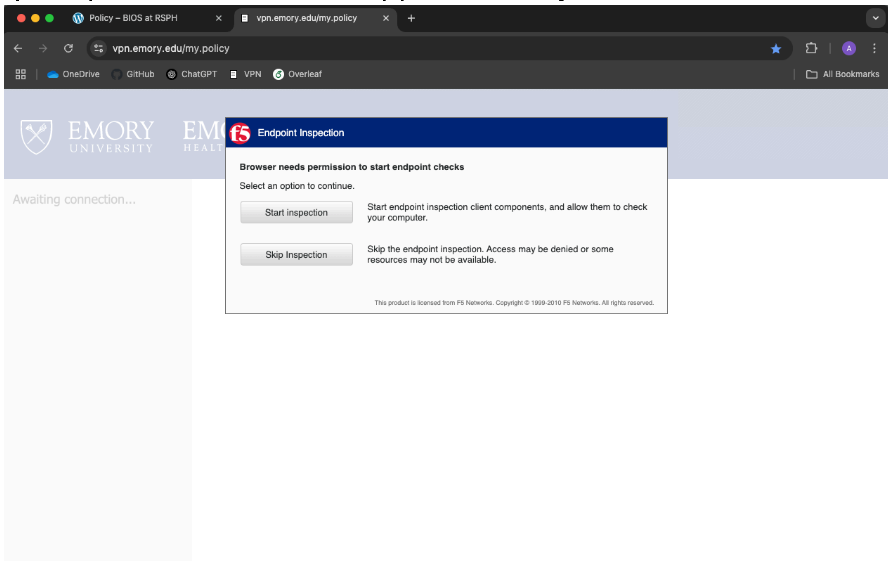
5/22/25

Official website: https://scholarblogs.emory.edu/rsph-hpc/

Official policy documentation: https://github.com/RSPH-HPC/Documentation/blob/master/HPC%20Getting%20Started%20Guide%20V.2.0.pdf

I. Connecting to HPC

1. Connect to VPN https://vpn.emory.edu/
    a. The first time you connect, it will prompt you to download the VPN client
    b. After the client is installed, you can connect to the VPN by opening the application directly or logging in through the VPN website. Follow the prompts until the F5 VPN application says 'Connected'



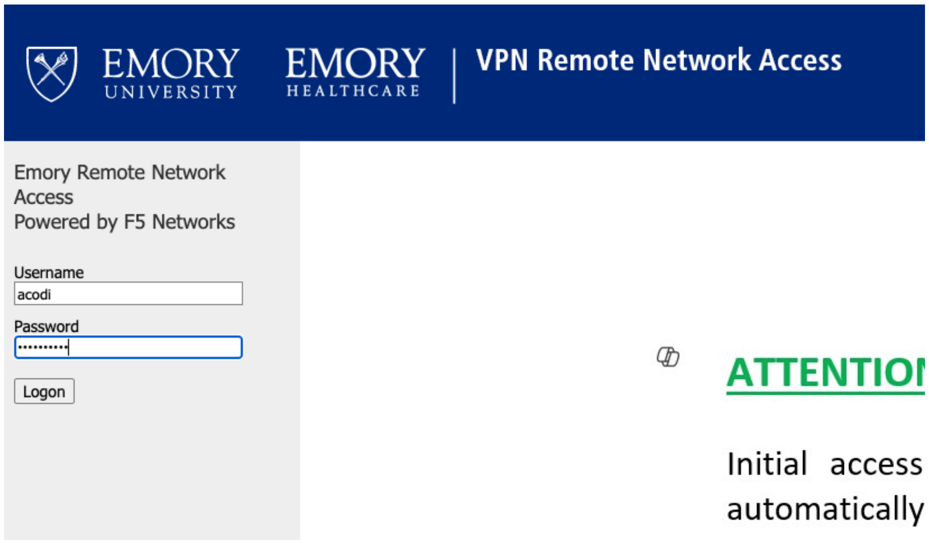
Browser needs permission to start endpoint checks

Select an option to continue.

Start inspection — Start endpoint inspection client components, and allow them to check your computer.

Skip Inspection — Skip the endpoint inspection. Access may be denied or some resources may not be available.

This product is licensed from F5 Networks. Copyright © 1999-2010 F5 Networks. All rights reserved.



EMORY UNIVERSITY EMORY HEALTHCARE | VPN Remote Network Access

Emory Remote Network Access
Powered by F5 Networks

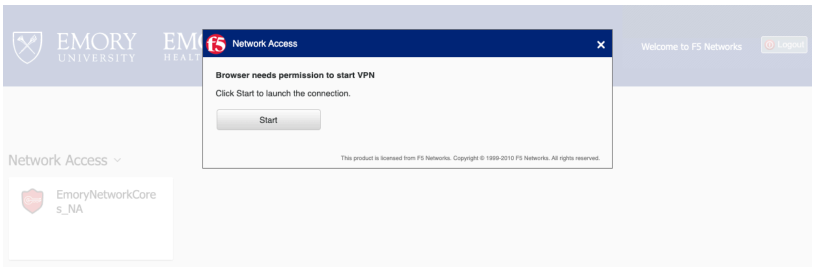
Username
acodi

Password
••••••••••

Logon

ATTENTIOI

Initial access
automatically



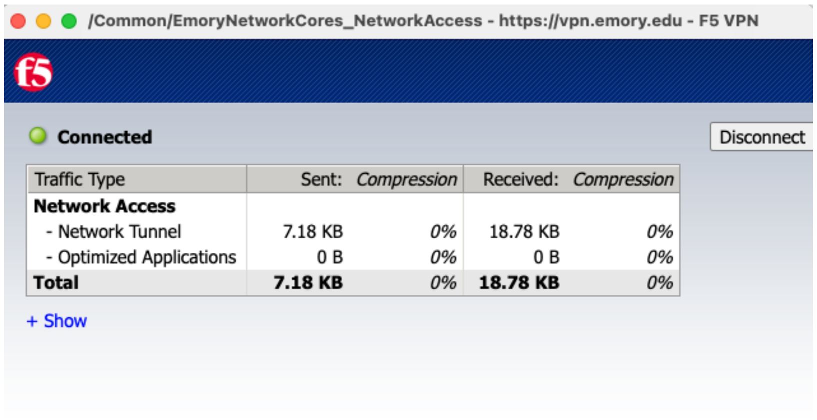
Network Access

Browser needs permission to start VPN

Click Start to launch the connection.

Start

This product is licensed from F5 Networks. Copyright © 1999-2010 F5 Networks. All rights reserved.



/Common/EmoryNetworkCores_NetworkAccess - https://vpn.emory.edu - F5 VPN

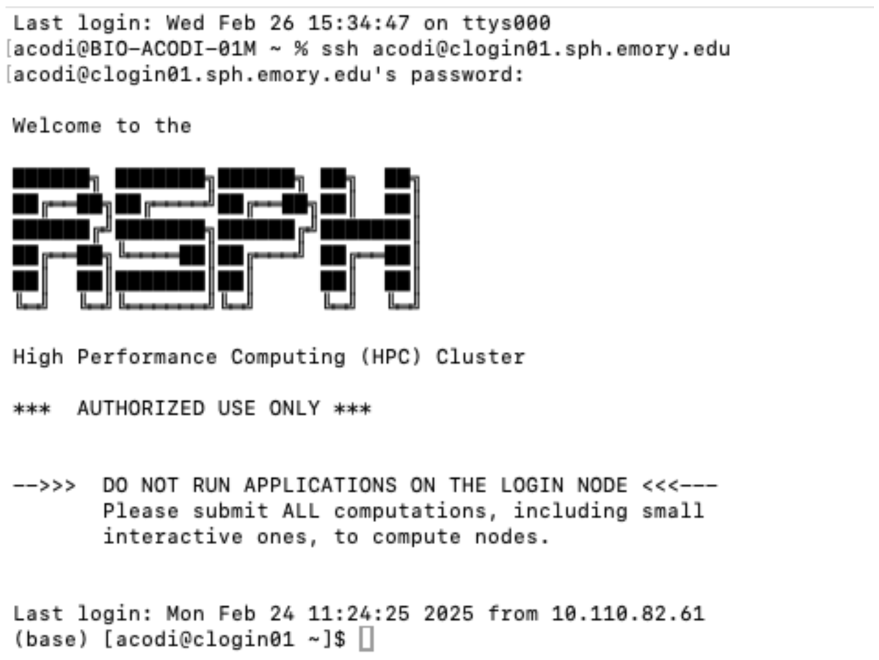
Connected

Disconnect

Traffic Type	Sent:	*Compression*	Received:	*Compression*
Network Access				
- Network Tunnel	7.18 KB	*0%*	18.78 KB	*0%*
- Optimized Applications	0 B	*0%*	0 B	*0%*
Total	**7.18 KB**	*0%*	**18.78 KB**	*0%*

+ Show

2. SSH into HPC via terminal
    a. Macbook
        i. Open terminal
        ii. Type `ssh <userid>@clogin01.sph.emory.edu` and hit enter
        iii. Enter password (you will not see characters appear, but type anyway; hit enter when done)



```
Last login: Wed Feb 26 15:34:47 on ttys000
acodi@BIO-ACODI-01M ~ % ssh acodi@clogin01.sph.emory.edu
acodi@clogin01.sph.emory.edu's password:

Welcome to the

High Performance Computing (HPC) Cluster

***  AUTHORIZED USE ONLY ***


-->>>  DO NOT RUN APPLICATIONS ON THE LOGIN NODE <<<---
       Please submit ALL computations, including small
       interactive ones, to compute nodes.


Last login: Mon Feb 24 11:24:25 2025 from 10.110.82.61
(base) [acodi@clogin01 ~]$ 
```

b. Windows
   i. Download a program with a terminal emulator (ex. git, which comes with git bash) or use the already-installed OpenSSH program
   ii. Open git bash or OpenSSH program
   iii. Type `ssh <userid>@clogin01.sph.emory.edu` and hit enter
   iv. Click open, enter password when prompted (you will not see characters appear, but type anyway; hit enter when done)

II. Navigating HPC

RSPH HPC uses a Linux operating system with SLURM for cluster management

1. Useful Linux commands

Command	**Description**
`cd <dir_name>`	Navigates into the folder/directory "dir_name"
`ls <dir_name>`	Lists all files and directories in the current directory
`ls -latr`	List with options. -l is list in long form (includes timestamps, permissions, size), -a is all files (including hidden files beginning with '.'), -t is sort by time with newest first, -r is reverse order while sorting. See man page for additional settings
`mkdir <dir_name>`	create new directory
`pwd`	Print path of current directory
`cp <file_to_copy> <copy_to_here>`	Copy file to new directory and/or rename. Include name of file in path
`mv <file_to_move> <move_to_here>`	Move file to another location. Include name of file in path
`rm <file_name>`	permanently delete/remove file
`rm -r <folder_name>`	permanently delete/remove folder (-r means recursive, will go through entire directory and delete)
`cat <file_name>`	print file
`touch <file_name>`	create new file with name <file_name>
`vi <file_name>`	open existing file in VIM text editor or create new file/open in text editor
`nano <file_name>`	open existing file in Nano text editor or create new file/open in text editor
`du -sh <folder_name>`	check how much storage is used by folder
`Ctrl + C`	kill current command
`man <command_name>`	open manual page for command
`clear`	clear screen
`chmod <###> <file_or_folder_name>`	change permissions on file or folder. Commonly needed for changing execution permissions of scripts. `chmod 711` will give you (owner) read, write, and execute permissions while other cluster users will only be able to execute. See here for additional details

Additional resource for learning Linux: https://www.linuxcommand.org/index.php

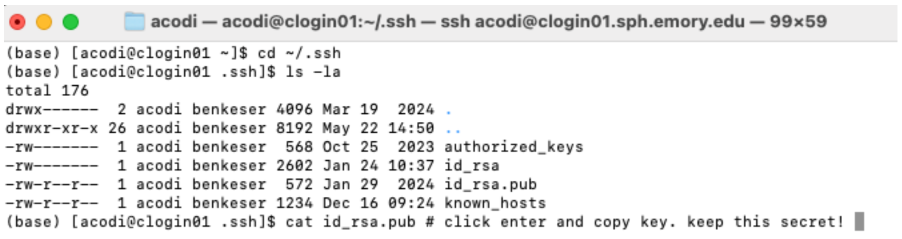
2. Useful SLURM specific commands

Command	**Description**
`sbatch <file_name>`	submit job to SLURM with default settings. See section V for additional details on sbatch
`squeue -u <userid>`	list jobs currently running from user (use your userid to list your own jobs)
`squeue --me`	list your own jobs
`squeue -p <partition_name>`	list jobs currently running on specific partition
`scancel<job_id>`	cancel job currently running
`scancel -u <userid>`	cancel all jobs from userid
`sinfo`	View available partitions on cluster

III. Setting up ssh for GitHub

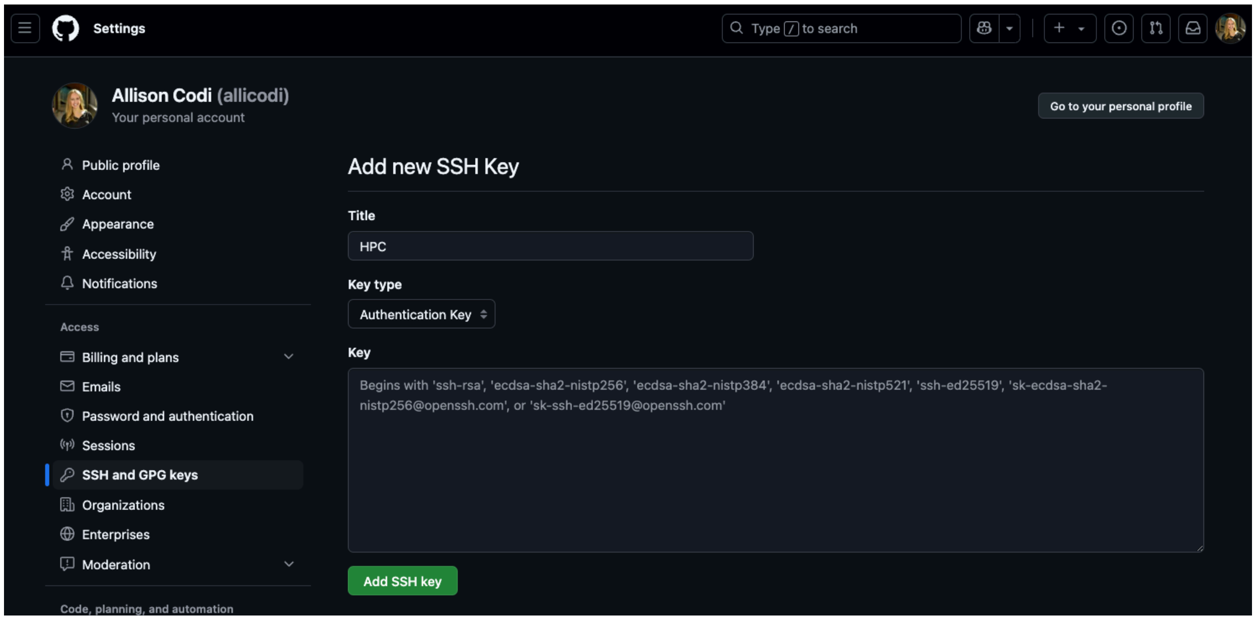
GitHub can be a helpful tool for project management and syncing files between your local computer and HPC. Upon initial setup, you must generate a ssh key and add it to your GitHub account. Detailed instructions can be found here. In summary:

1. Check for existing SSH keys.
    a. If none exist, create one using ssh-keygen (Additional instructions here)
    b. Otherwise, copy public key


acodi — acodi@clogin01:~/.ssh — ssh acodi@clogin01.sph.emory.edu — 99×59

```
(base) [acodi@clogin01 ~]$ cd ~/.ssh
(base) [acodi@clogin01 .ssh]$ ls -la
total 176
drwx------  2 acodi benkeser 4096 Mar 19  2024 .
drwxr-xr-x 26 acodi benkeser 8192 May 22 14:50 ..
-rw-------  1 acodi benkeser  568 Oct 25  2023 authorized_keys
-rw-------  1 acodi benkeser 2602 Jan 24 10:37 id_rsa
-rw-r--r--  1 acodi benkeser  572 Jan 29  2024 id_rsa.pub
-rw-r--r--  1 acodi benkeser 1234 Dec 16 09:24 known_hosts
(base) [acodi@clogin01 .ssh]$ cat id_rsa.pub # click enter and copy key. keep this secret!
```

2. Log into your GitHub account. Click icon in top right corner to enter settings → SSH and GPG keys → Add new SSH key. Paste the key you copied above into the Key box, and give it an informative title.



Add new SSH Key

Title: HPC

Key type: Authentication Key

Key: Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

3. Test that ssh access is working by entering `ssh -T git@github.com`. You should receive the following message:

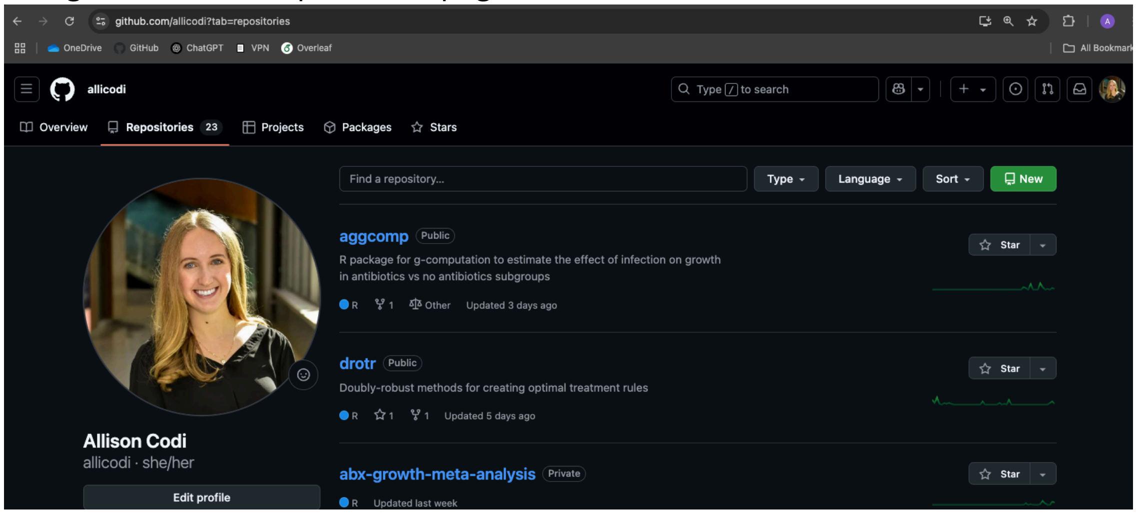
```
(base) [acodi@clogin01 drotr_example_analysis]$ ssh -T git@github.com
Hi allicodi! You've successfully authenticated, but GitHub does not provide shell access.
(base) [acodi@clogin01 drotr_example_analysis]$
```

IV. File transfer

a. File transfer via GitHub

Useful for version control and quickly syncing code back and forth from local computer to HPC

1. Set up remote repository on GitHub
   a. Navigate to Home/Repositories page and click “New”



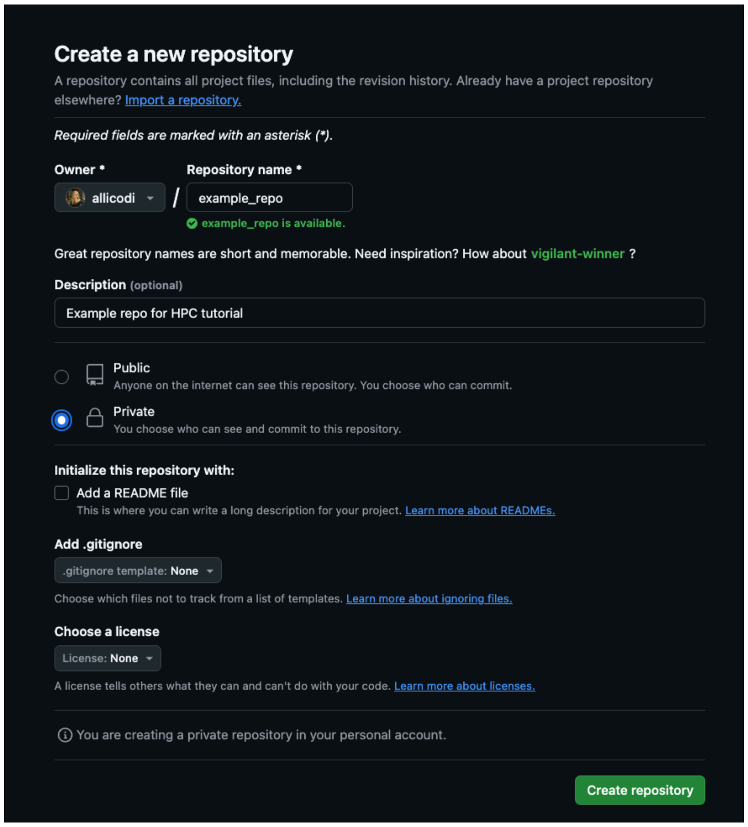
Allison Codi
allicodi · she/her

aggcomp Public
R package for g-computation to estimate the effect of infection on growth in antibiotics vs no antibiotics subgroups
R 1 Other Updated 3 days ago

drotr Public
Doubly-robust methods for creating optimal treatment rules
R 1 1 Updated 5 days ago

abx-growth-meta-analysis Private
R Updated last week

   b. Enter settings for remote repository



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner *: allicodi / Repository name *: example_repo

example_repo is available.

Great repository names are short and memorable. Need inspiration? How about vigilant-winner ?

Description (optional): Example repo for HPC tutorial

Public — Anyone on the internet can see this repository. You choose who can commit.

[✓ Private] — You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file — This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

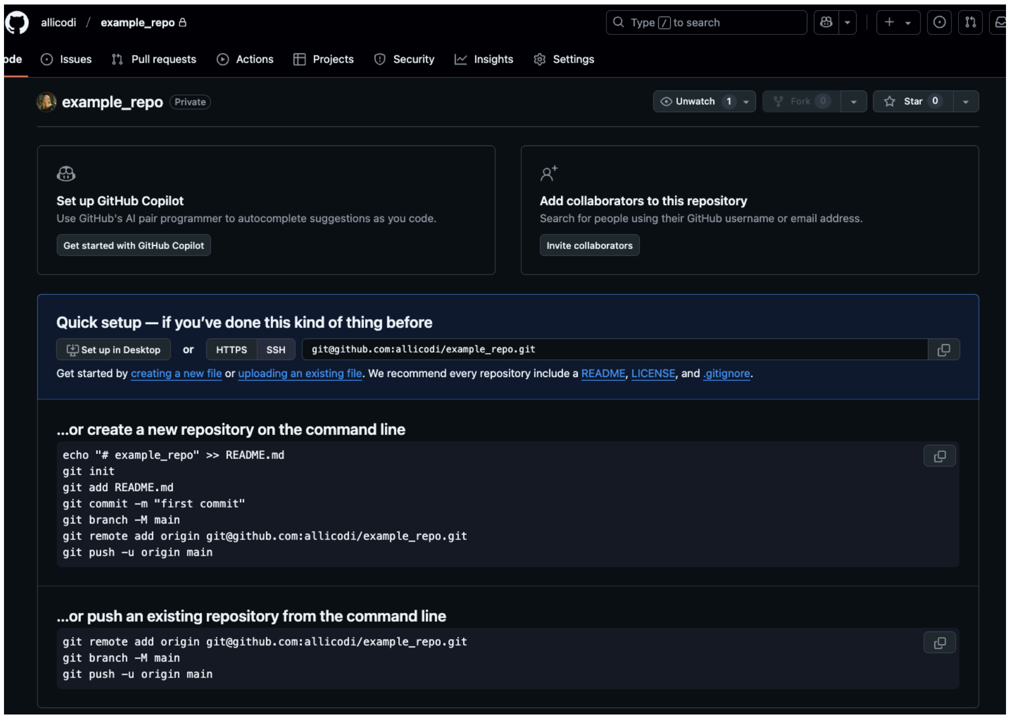
Choose which files not to track from a list of templates. Learn more about ignoring files.

Choose a license

License: None

A license tells others what they can and can't do with your code. Learn more about licenses.

You are creating a private repository in your personal account.



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS [✓ SSH] `git@github.com:allicodi/example_repo.git`

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

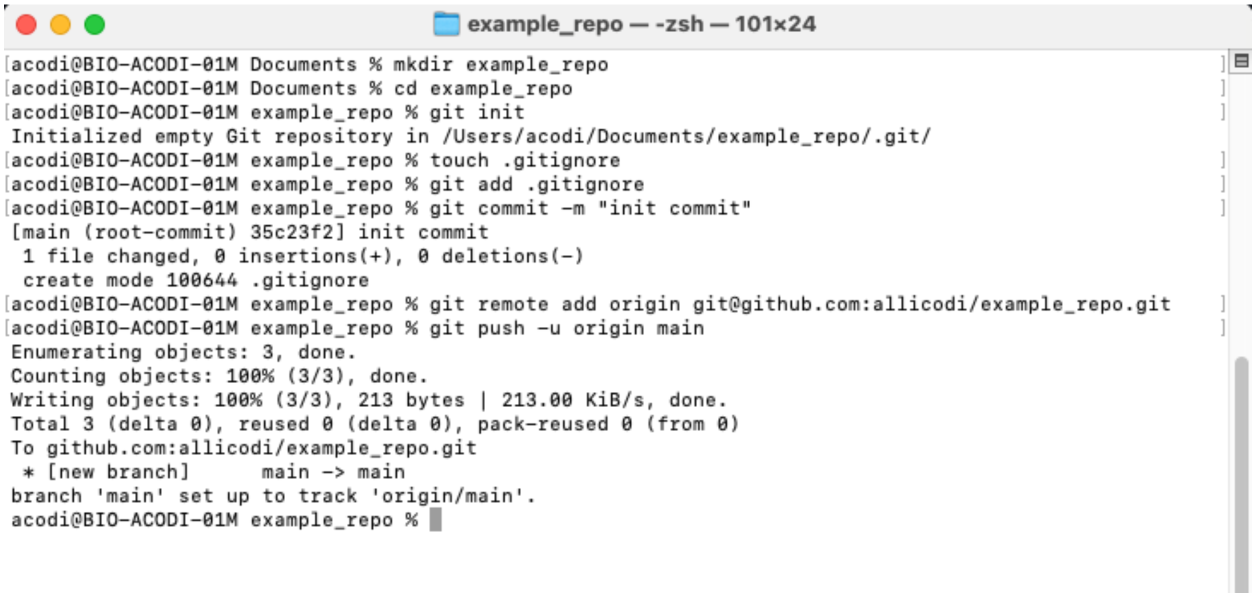
...or create a new repository on the command line

```
echo "# example_repo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:allicodi/example_repo.git
git push -u origin main
```

...or push an existing repository from the command line

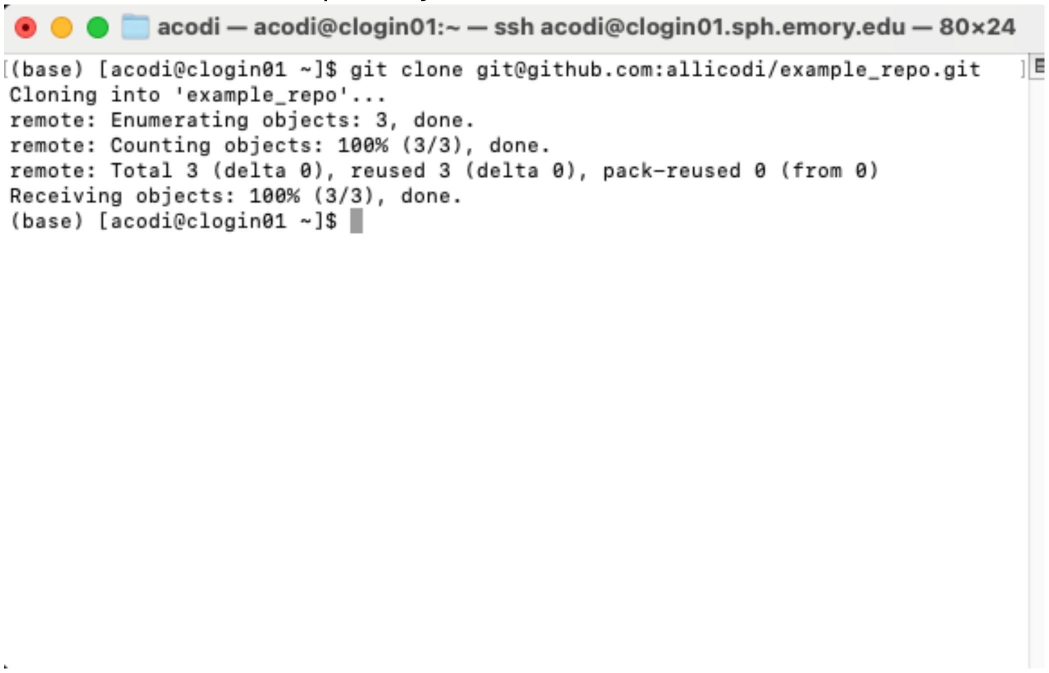
```
git remote add origin git@github.com:allicodi/example_repo.git
git branch -M main
git push -u origin main
```

2. Create and/or link local repository to remote repository. Note this example assumes you already have set up an ssh key for GitHub to your local machine



```
acodi@BIO-ACODI-01M Documents % mkdir example_repo
acodi@BIO-ACODI-01M Documents % cd example_repo
acodi@BIO-ACODI-01M example_repo % git init
Initialized empty Git repository in /Users/acodi/Documents/example_repo/.git/
acodi@BIO-ACODI-01M example_repo % touch .gitignore
acodi@BIO-ACODI-01M example_repo % git add .gitignore
acodi@BIO-ACODI-01M example_repo % git commit -m "init commit"
[main (root-commit) 35c23f2] init commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 .gitignore
acodi@BIO-ACODI-01M example_repo % git remote add origin git@github.com:allicodi/example_repo.git
acodi@BIO-ACODI-01M example_repo % git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 213 bytes | 213.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:allicodi/example_repo.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
acodi@BIO-ACODI-01M example_repo %
```

3. Clone remote repository on HPC

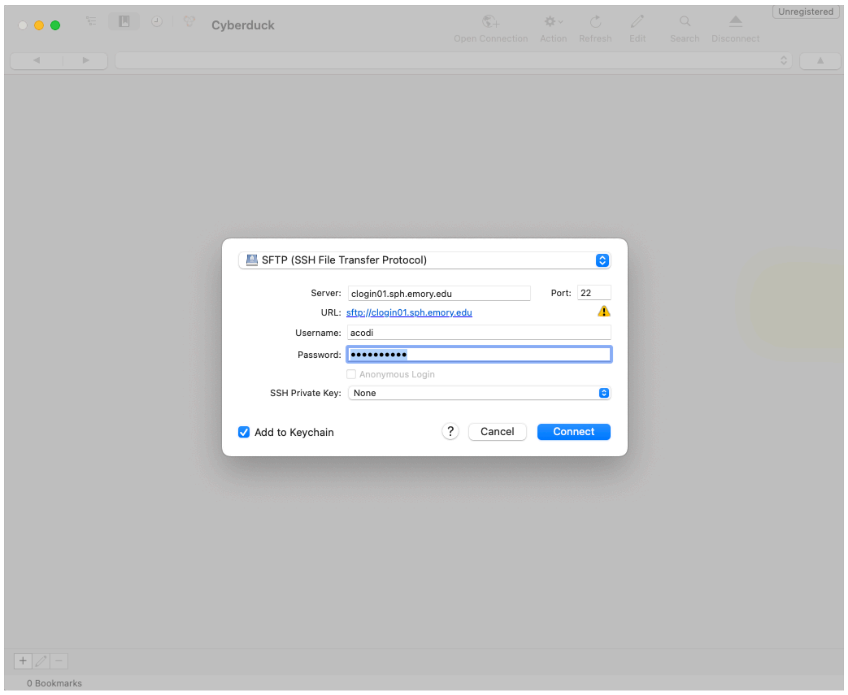

acodi — acodi@clogin01:~ — ssh acodi@clogin01.sph.emory.edu — 80×24

```
(base) [acodi@clogin01 ~]$ git clone git@github.com:allicodi/example_repo.git
Cloning into 'example_repo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
(base) [acodi@clogin01 ~]$
```

b. File transfer via SFTP Client

Useful for transferring large data/results files you do not want to upload to GitHub

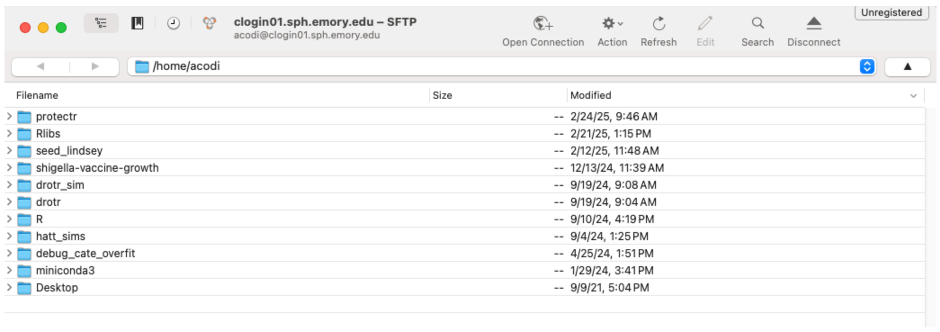
1. Download and/or open SFTP client (ex. Cyberduck)
2. Click ‘Open Connection’
3. Enter ssh settings
    a. Server = clogin01.sph.emory.edu
    b. Port = 22
    c. Username = <userid>
    d. Password = password



SFTP (SSH File Transfer Protocol)

Server:	clogin01.sph.emory.edu
Port:	22
URL:	sftp://clogin01.sph.emory.edu
Username:	acodi
Password:	••••••••••
	Anonymous Login
SSH Private Key:	None

[✓] Add to Keychain   ?   Cancel   Connect

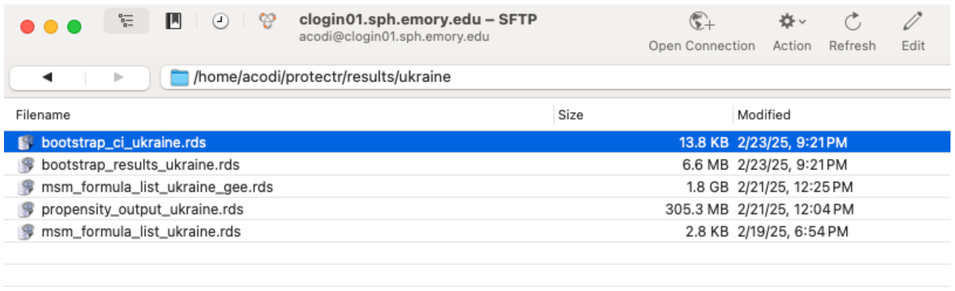


clogin01.sph.emory.edu – SFTP
acodi@clogin01.sph.emory.edu

/home/acodi

Filename	Size	Modified
protectr	--	2/24/25, 9:46 AM
Rlibs	--	2/21/25, 1:15 PM
seed_lindsey	--	2/12/25, 11:48 AM
shigella-vaccine-growth	--	12/13/24, 11:39 AM
drotr_sim	--	9/19/24, 9:08 AM
drotr	--	9/19/24, 9:04 AM
R	--	9/10/24, 4:19 PM
hatt_sims	--	9/4/24, 1:25 PM
debug_cate_overfit	--	4/25/24, 1:51 PM
miniconda3	--	1/29/24, 3:41 PM
Desktop	--	9/9/21, 5:04 PM

4. Downloading files
    a. Click into folder with file to download

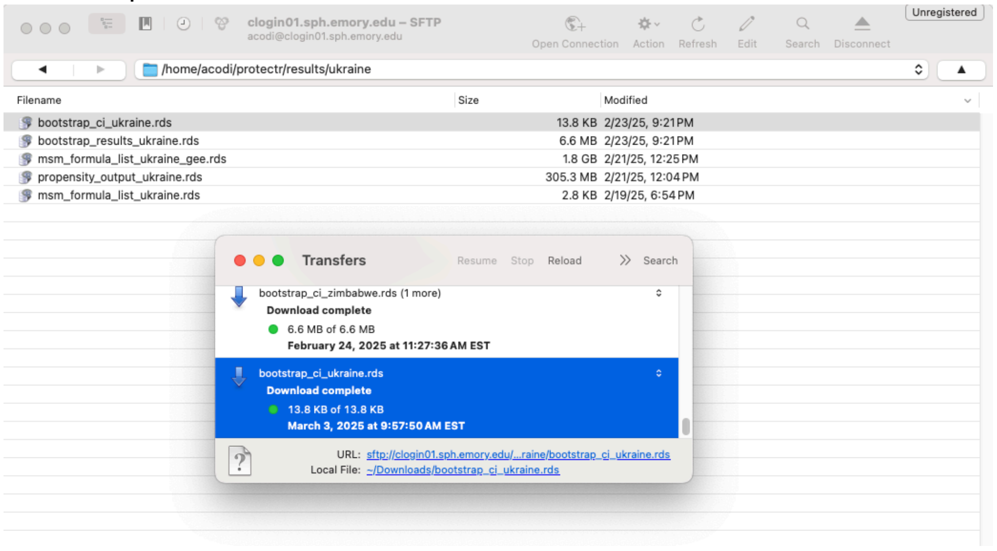


clogin01.sph.emory.edu – SFTP
acodi@clogin01.sph.emory.edu

/home/acodi/protectr/results/ukraine

Filename	Size	Modified
bootstrap_ci_ukraine.rds	13.8 KB	2/23/25, 9:21 PM
bootstrap_results_ukraine.rds	6.6 MB	2/23/25, 9:21 PM
msm_formula_list_ukraine_gee.rds	1.8 GB	2/21/25, 12:25 PM
propensity_output_ukraine.rds	305.3 MB	2/21/25, 12:04 PM
msm_formula_list_ukraine.rds	2.8 KB	2/19/25, 6:54 PM

b. Double click file to automatically download into Downloads folder on local computer



clogin01.sph.emory.edu – SFTP
acodi@clogin01.sph.emory.edu

/home/acodi/protectr/results/ukraine

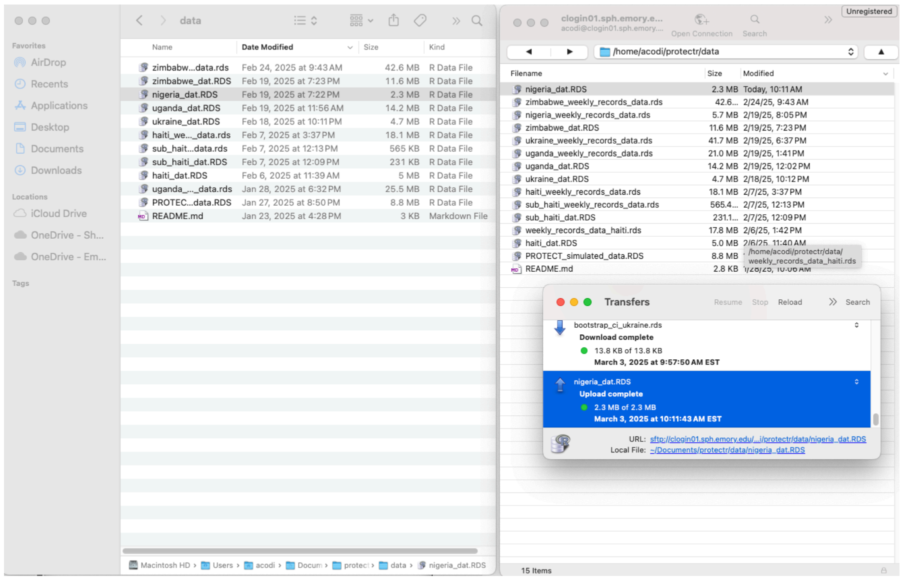
Filename	Size	Modified
bootstrap_ci_ukraine.rds	13.8 KB	2/23/25, 9:21 PM
bootstrap_results_ukraine.rds	6.6 MB	2/23/25, 9:21 PM
msm_formula_list_ukraine_gee.rds	1.8 GB	2/21/25, 12:25 PM
propensity_output_ukraine.rds	305.3 MB	2/21/25, 12:04 PM
msm_formula_list_ukraine.rds	2.8 KB	2/19/25, 6:54 PM

bootstrap_ci_zimbabwe.rds (1 more)
Download complete
6.6 MB of 6.6 MB
February 24, 2025 at 11:27:36 AM EST

bootstrap_ci_ukraine.rds
Download complete
13.8 KB of 13.8 KB
March 3, 2025 at 9:57:50 AM EST

URL: sftp://clogin01.sph.emory.edu/...raine/bootstrap_ci_ukraine.rds
Local File: ~/Downloads/bootstrap_ci_ukraine.rds

5. Uploading files
   a. Drag and drop file of interest from local computer into folder of interest via SFTP client



Name	Date Modified	Size	Kind
zimbabw...data.rds	Feb 24, 2025 at 9:43 AM	42.6 MB	R Data File
zimbabwe_dat.RDS	Feb 19, 2025 at 7:23 PM	11.6 MB	R Data File
nigeria_dat.RDS	Feb 19, 2025 at 7:22 PM	2.3 MB	R Data File
uganda_dat.RDS	Feb 19, 2025 at 11:56 AM	14.2 MB	R Data File
ukraine_dat.RDS	Feb 18, 2025 at 10:11 PM	4.7 MB	R Data File
haiti_we..._data.rds	Feb 7, 2025 at 3:37 PM	18.1 MB	R Data File
sub_hait...data.rds	Feb 7, 2025 at 12:13 PM	565 KB	R Data File
sub_haiti_dat.RDS	Feb 7, 2025 at 12:09 PM	231 KB	R Data File
haiti_dat.RDS	Feb 6, 2025 at 11:39 AM	5 MB	R Data File
uganda_..._data.rds	Jan 28, 2025 at 6:32 PM	25.5 MB	R Data File
PROTEC...data.RDS	Jan 27, 2025 at 8:50 PM	8.8 MB	R Data File
README.md	Jan 23, 2025 at 4:28 PM	3 KB	Markdown File

Macintosh HD › Users › acodi › Docum › protect › data › nigeria_dat.RDS

/home/acodi/protectr/data

Filename	Size	Modified
nigeria_dat.RDS	2.3 MB	Today, 10:11 AM
zimbabwe_weekly_records_data.rds	42.6...	2/24/25, 9:43 AM
nigeria_weekly_records_data.rds	5.7 MB	2/19/25, 8:05 PM
zimbabwe_dat.RDS	11.6 MB	2/19/25, 7:23 PM
ukraine_weekly_records_data.rds	41.7 MB	2/19/25, 6:37 PM
uganda_weekly_records_data.rds	21.0 MB	2/19/25, 1:41 PM
uganda_dat.RDS	14.2 MB	2/19/25, 12:02 PM
ukraine_dat.RDS	4.7 MB	2/18/25, 10:12 PM
haiti_weekly_records_data.rds	18.1 MB	2/7/25, 3:37 PM
sub_haiti_weekly_records_data.rds	565.4...	2/7/25, 12:13 PM
sub_haiti_dat.RDS	231.1...	2/7/25, 12:09 PM
weekly_records_data_haiti.rds	17.8 MB	2/6/25, 1:42 PM
haiti_dat.RDS	5.0 MB	2/6/25, 11:40 AM
PROTECT_simulated_data.RDS	8.8 MB	[illegible]
README.md	2.8 KB	1/28/25, 10:06 AM

/home/acodi/protectr/data/
weekly_records_data_haiti.rds

bootstrap_ci_ukraine.rds
Download complete
13.8 KB of 13.8 KB
March 3, 2025 at 9:57:50 AM EST

nigeria_dat.RDS
Upload complete
2.3 MB of 2.3 MB
March 3, 2025 at 10:11:43 AM EST

URL: sftp://clogin01.sph.emory.edu/...i/protectr/data/nigeria_dat.RDS
Local File: ~/Documents/protectr/data/nigeria_dat.RDS

15 Items

Note you can also use `scp` to transfer files via command line.

- Remote to local:
  - `scp -r <your_user_name>@clogin01.sph.emory.edu:/<path_to_file>`
- Local to remote:
  - `scp <file_name> <your_user_name>@clogin01.sph.emory.edu:/<path_to_file>`

V. Modules

Modules refer to the software that is installed on HPC and available to use

Command	**Description**
`module available`	list all software installed on cluster
`module load <module_name>`	load a module. (ex. `module load R/4.4.0`)
`module list`	list currently loaded modules
`module unload <module_name>`	unload module
`module purge`	unload all modules at once

VI. Writing sbatch scripts in bash

Bash scripts can be used to submit jobs and tailor settings using the SBATCH command

Example script 1:

```
#!/bin/bash


PARTITION=$1

SETTING=$2

NSEEDS=$3


sbatch --array=1-$NSEEDS \

         --partition=$PARTITION \

         --nodes=1 \

         --ntasks-per-node=1 \

         --cpus-per-task=32 \

         --job-name=my_job \

         --output=/projects/dbenkes/allison/my_project/scratch/${SETTING}_boot_${SLURM_ARRAY_TASK_ID}_%J.out \

         --export=SLURM_ARRAY_TASK_ID=$SLURM_ARRAY_TASK_ID,SETTING=$SETTING \

         --wrap "Rscript run_simulation.R"
```

This script runs run_simulation.R NSEED times on the specified partition. It will be run on a single node and use 32 CPUs on that node. Any messages/errors from the code will be saved in a scratch file in the specified folder. The setting specified in SETTING will be available to the R script as an environment variable.

Example script 2 (from RSPH HPC documentation):

```
#!/bin/bash


#SBATCH --nodes=1

#SBATCH --ntasks-per-node=4

#SBATCH --mem-per-cpu=1G

#SBATCH --time=01:00:00

#SBATCH --partition=week-long-cpu

#SBATCH --job-name=HelloWorld

#SBATCH --error=job.%J.err

#SBATCH --output=job.%J.out


echo "HelloWorld"

sleep 300
```

The script above will request for 1 node, 4 cores, 4Gb memory (1G per core), 1 hour run time on the partition named 'week-long-cpu'. The job name is 'HelloWorld'. This job will print 'HelloWorld' to the output file then stay idle for another 300 seconds.

Useful sbatch arguments include (but are not limited to):

Argument	**Description**
--nodes	**Number of nodes requested**
--ntasks-per-node	**Number of cores per node. For example, if –nodes=2 and –ntasks-per-node=2, then a total of 4 cores will be requested**
--ntasks	**Total number of cores regardless of node number**
--mem-per-cpu	**Memory allocated per core for the job**
--time	**Maximum time job can run**
--job-name	**Name of job, will be listed in queue**
--partition	**The partition to run the job on**
--output	**Location of stdout for the job**
--array	**Submit jobs over array**

See here for additional arguments

VII. Interactive computing

DO NOT RUN ANY LOCAL CODE ON THE LOGIN NODE. If you want to run a script or test code interactively, start an interactive job on the interactive-cpu node.

```
srun --pty --partition=interactive-cpu --nodes=1 --ntasks-per-node=1 --
mem-per-cpu=8G --time=02:00:00 bash
```

Exit interactive session with ‘exit’

VIII. Personal libraries on HPC

Some R packages are not automatically installed on HPC and may not download to the main Rlibs folder. To get around this, you can create a personal library and install R packages into it.

1. Make directory to install packages into `mkdir ~/Rlibs`
2. Set path to library at top of R script to be run on HPC using `.libPaths(“~/Rlibs”)`