SBGrid RELION Workshop 2017

Software installation
Intro to Computing on Linux

Jason Key PhD & Peter Meyer PhD

BCMP Harvard Medical School

SBGrid Consortium

key@hkl.hms.harvard.edu
meyer@hkl.hms.harvard.edu

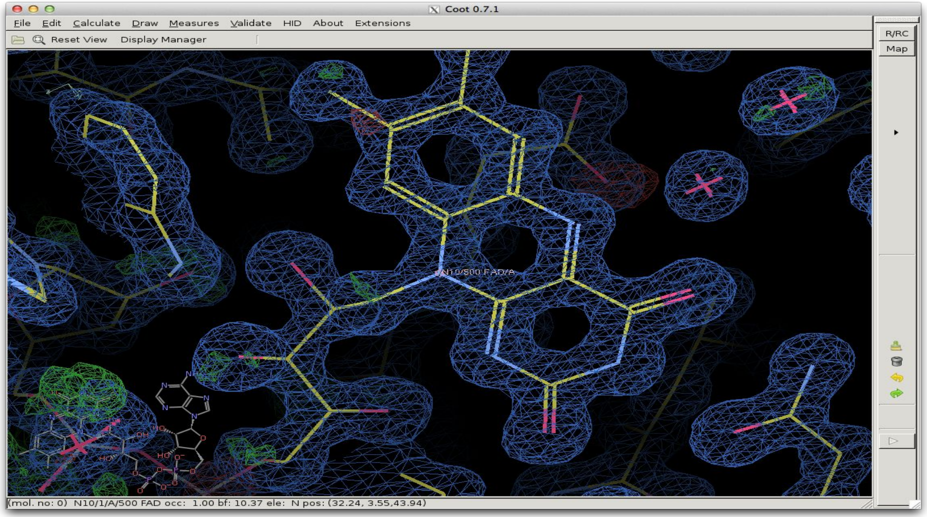
Welcome!

SBGrid : Structural Biology Research Computing

Who we are:

Non-profit Consortium based in BCMP @HMS focused on Structural Biology computing

Structural Biologists, IT pros, software engineers, programmers, software policy advocates, postdocs, students



(mol. no: 0) N10/1/A/500 FAD occ: 1.00 bf: 10.37 ele: N pos: (32.24, 3.55,43.94)

Welcome!

SBGrid : Structural Biology Research Computing

Pete
Meyer

Mick
Timony

Dimitry
Filonov

Justin
O’Connor

Piotr
Sliz

James
Vincent

Michelle
Ottaviano

Saythyda
Corrado

Carol Herre
Rob DiCamillo

SBGrid RELION Workshop 2017

Software Installation

Intro to Linux Computing

SBGrid RELION Workshop 2017

We will be using Amazon EC2.
You may want to install RELION and associated applications:

- RELION

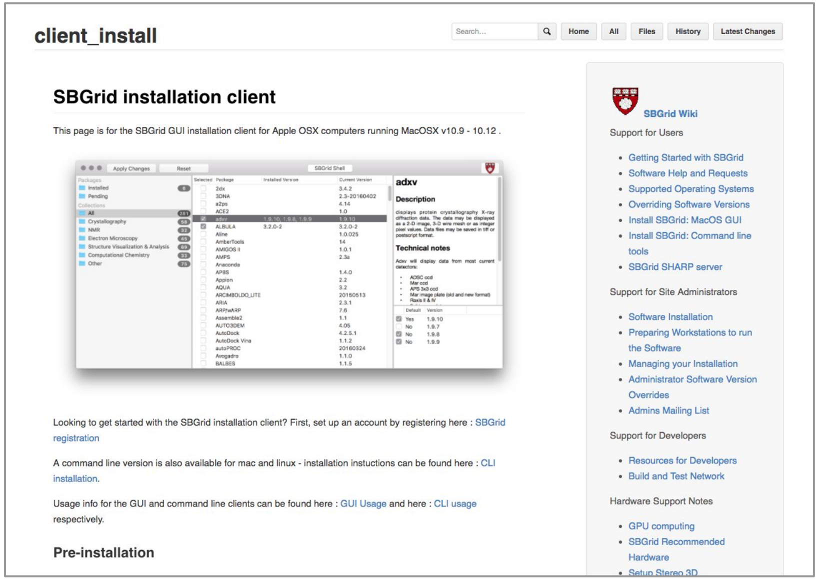
  gctf ctf motioncorr motioncor2 unblur summovie

- Chimera

SBGrid RELION Workshop 2017

___

https://sbgrid.org/wiki/client_install



client_install

SBGrid installation client

This page is for the SBGrid GUI installation client for Apple OSX computers running MacOSX v10.9 - 10.12 .

Selected	Package	Installed Version	Current Version
□	2dx		3.4.2
□	3DNA		2.3-20160402
□	a2ps		4.14
□	ACE2		1.0
☑	adxv	1.9.10, 1.9.8, 1.9.9	1.9.10
☑	ALBULA	3.2.0-2	3.2.0-2
□	Aline		1.0.025
□	AmberTools		14
□	AMIGOS II		1.0.1
□	AMPS		2.3a
□	Anaconda		
□	APBS		1.4.0
□	Appion		2.2
□	AQUA		3.2
□	ARCIMBOLDO_LITE		20150513
□	ARIA		2.3.1
□	ARP/wARP		7.6
□	Assemble2		1.1
□	AUTO3DEM		4.05
□	AutoDock		4.2.5.1
□	AutoDock Vina		1.1.2
□	autoPROC		20160324
□	Avogadro		1.1.0
□	BALBES		1.1.5

adxv

Description

displays protein crystallography X-ray diffraction data. The data may be displayed as a 2-D image, 3-D wire mesh or as integer pixel values. Data files may be saved in tiff or postscript format.

Technical notes

Adxv will display data from most current detectors:

- ADSC ccd
- Mar ccd
- APS 3x3 ccd
- Mar image plate (old and new format)
- Raxis II & IV

	Default	Version
☑	Yes	1.9.10
□	No	1.9.7
☑	No	1.9.8
☑	No	1.9.9

Looking to get started with the SBGrid installation client? First, set up an account by registering here : SBGrid registration

A command line version is also available for mac and linux - installation instuctions can be found here : CLI installation.

Usage info for the GUI and command line clients can be found here : GUI Usage and here : CLI usage respectively.

Pre-installation

SBGrid Wiki

Support for Users

- Getting Started with SBGrid
- Software Help and Requests
- Supported Operating Systems
- Overriding Software Versions
- Install SBGrid: MacOS GUI
- Install SBGrid: Command line tools
- SBGrid SHARP server

Support for Site Administrators

- Software Installation
- Preparing Workstations to run the Software
- Managing your Installation
- Administrator Software Version Overrides
- Admins Mailing List

Support for Developers

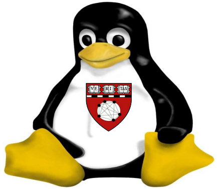
- Resources for Developers
- Build and Test Network

Hardware Support Notes

- GPU computing
- SBGrid Recommended Hardware
- Setup Stereo 3D

SBGrid RELION Workshop 2017

Intro to Linux Computing



- Introduction to Linux

  Why Linux?

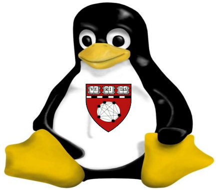
- The Linux interface

  (Understanding the Shell, scripting)

- Scientific Computing on linux -

  Tips and tools for computing and research

Linux

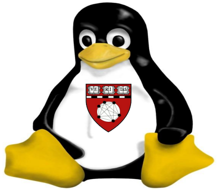


Linux is an open-source operating systems modeled on UNIX developed by Linux Torvalds in 1991.

Comp.os.minix

Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. …

Linux

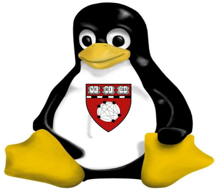


The GNU (Gnu’s Not Unix) was an effort to develop free and open source OS and applications.

Torvalds developed ‘kernel’ and combined it with software from Richard Stallman @ MIT.

GNU’s Not Unix

Linux

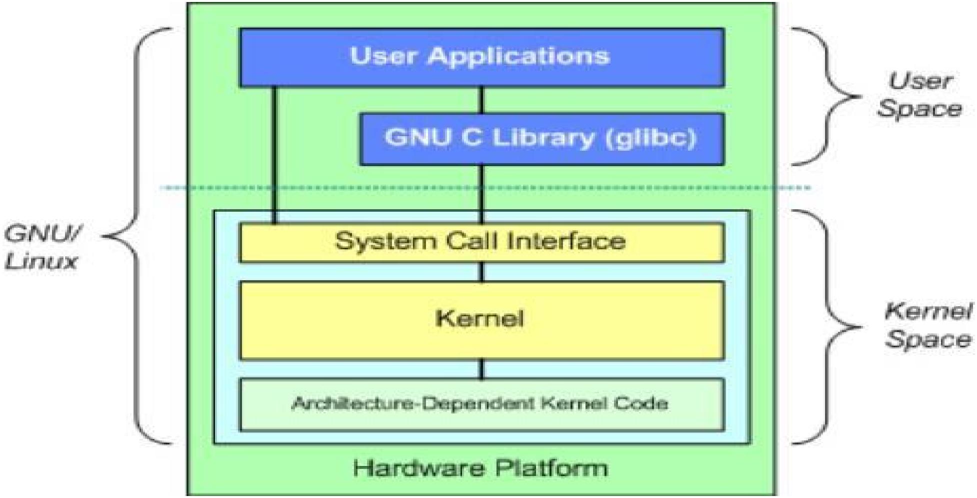


Multi-user, multi-tasking

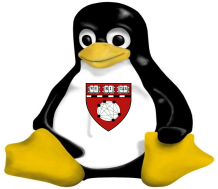
Many users on the same machine at once, running many programs

Multi-platform

runs on many different processor types



Linux



Multi-user, multi-tasking

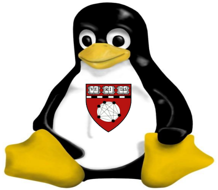
Many users on the same machine at once, running many programs

Multi-platform

runs on many different processor types

Linux is a Unix-like system free of proprietary software for which source code is available and freely distributed

Linux



Multi-user, multi-tasking

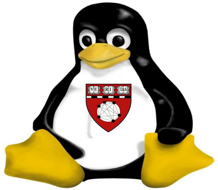
Many users on the same machine at once, running many programs

Multi-platform

runs on many different processor types

Linux is a Unix-like system free of proprietary software for which source code is available and freely distributed

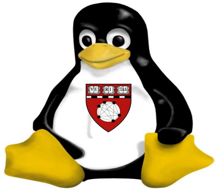
Linux



But why did Linux succeed?
(and not HURD, BSD, Minux, etc ...)

- Decentralized Development
- Pragmatic (Not an academic or ideological exercise)
- Technological Superiority
- Luck?

Linux



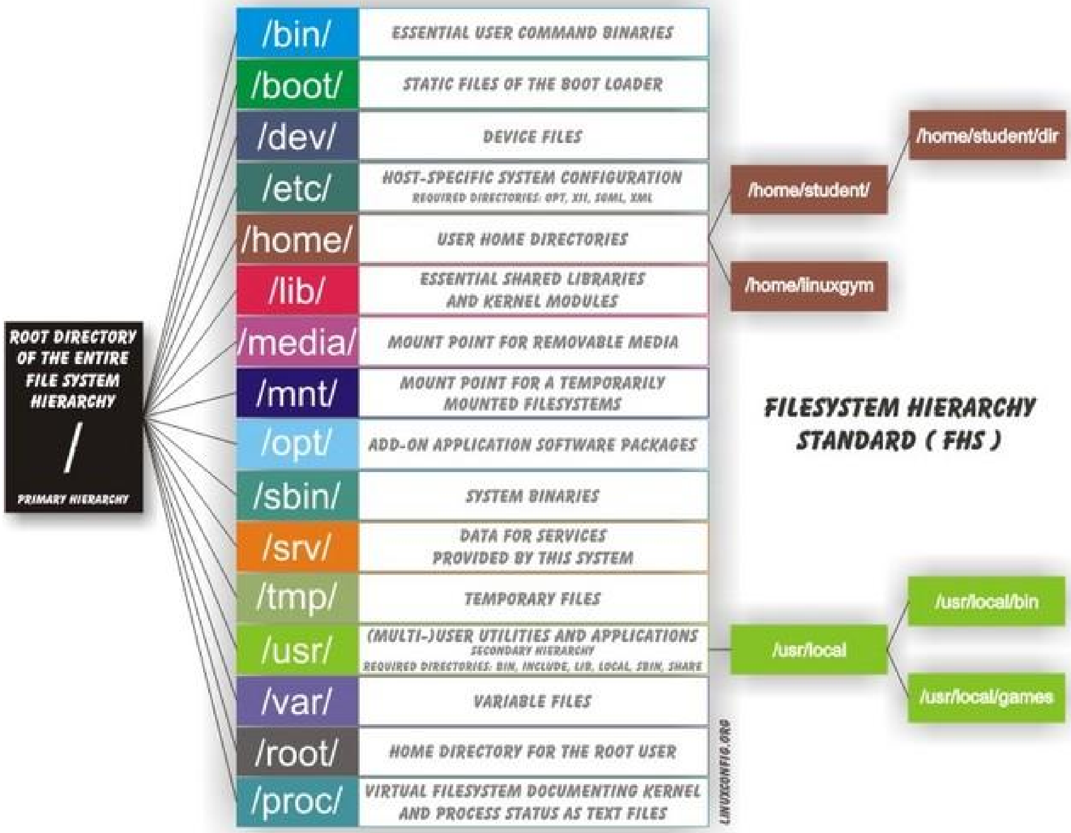
The Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents.

"Everything is a file"

Defining features of *nix

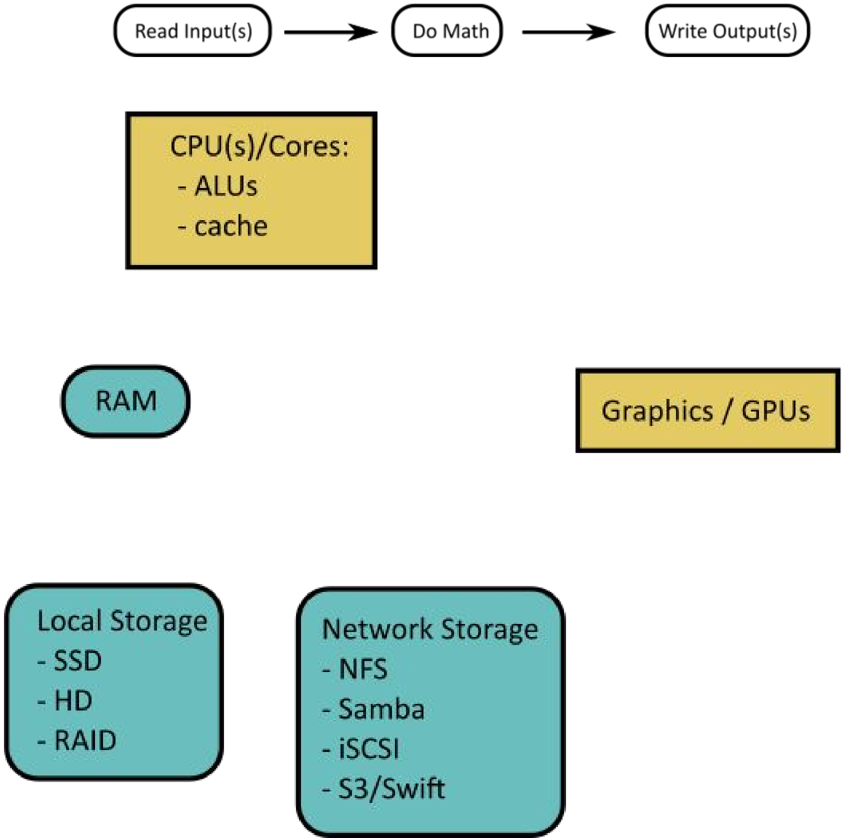
Resources (documents, directories, keyboards, printers, storage, network communications, etc)

Are simple streams of bytes exposed through the filesystem name space



/bin/	ESSENTIAL USER COMMAND BINARIES
/boot/	STATIC FILES OF THE BOOT LOADER
/dev/	DEVICE FILES
/etc/	HOST-SPECIFIC SYSTEM CONFIGURATION REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/	USER HOME DIRECTORIES
/lib/	ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/	MOUNT POINT FOR REMOVABLE MEDIA
/mnt/	MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/	ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/	SYSTEM BINARIES
/srv/	DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/	TEMPORARY FILES
/usr/	(MULTI-)USER UTILITIES AND APPLICATIONS SECONDARY HIERARCHY REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/	VARIABLE FILES
/root/	HOME DIRECTORY FOR THE ROOT USER
/proc/	VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES

Hardware and Workflows


Read Input(s) → Do Math → Write Output(s)

CPU(s)/Cores:
- ALUs
- cache

RAM

Graphics / GPUs

Local Storage
- SSD
- HD
- RAID

Network Storage
- NFS
- Samba
- iSCSI
- S3/Swift

The Shell

The ‘shell’ is the Command Line Interface for Linux

This is an program that interprets what you type, keeps track of programs on the system, etc.

Common Shells:

tcsh : exTended C SHell
bash : Bourne Again SHell
ksh : Korn SHell
csh : C SHell (early popular shell)
sh : the original shell, often a synonym for bash now

The Shell : Commands

Import / include (source)

Navigation (cd, pwd, ls)

Manipulating Files (cp, rm, mv, mkdir)

Search (grep, find)

Permissions (chmod, chown, chgrp)

Job Control (jobs, ps, fg, bg, nohup)

http://linuxcommand.org/lc3_learning_the_shell.php

The Shell : The Environment

The shell environment is configured globally per user in files and startup scripts

- Settings for variables
- Function definitions
- Aliases

Except for the reserved Shell special parameters variable names can be set by the user

Quotes remove special meaning from one or multiple characters

The Shell : The Environment

The ‘printenv’ command

The Shell : The Environment

The ‘printenv’ command

Shell variables:

PATH

Where executables can be found

HOME

User’s home directory

USER

User’s username

SHELL

Default shell setting

The Shell : The Environment

The ‘printenv’ command

Shell variables:

PS1

Shell prompt settings

LD_LIBRARY_PATH

Primary search path for library directories

...

The Shell : The Environment

The ‘alias’ command

An alias is a shortcut or abbreviation.

Great for avoiding typing a long command sequences

Aliases do NOT get passed to scripts (sub-shells)

The Shell : The Environment

Functions: The ‘declare -f’ command

Functions are subroutines :
a code block (list of commands) that implements a set of operations.

The Shell : stdout stdin stderr

stdin :

- Input for commands
  usually come from the keyboard

stdout :

- Output from commands
  written to the screen

stderr :

- Error messages from processes
  usually written to the screen

The Shell : stdout stdin stderr

Pipe (|):

stdout of one command to stdin of another command

Output Redirection (>):

stdout of a command to a file

Output Appending (>>):

stdout of a command appending to a file

Input Redirection (<):

stdin of a command from a file

Use “-” to read this from standard input

The Shell : stdout stdin stderr

Stderr redirection

For tcsh

```
    &> filename
```

For bash

```
    2>&1 filename
```

The Shell : stdout stdin stderr

Most Linux (*nix) commands can be strung together

Example:

How many image files do I have?

```
    ls -l *img | wc
```

How many image files do I have that are not have ‘native’ in the name?

```
    ls -l *img | grep -v “native” | wc
```

The Shell : stdout stdin stderr

Most Linux (*nix) commands can be strung together

Example:

A list of all my image files :

```
ls *img > my_images.txt
```

A list of all my images sorted in reverse numerical order?

```
ls -l *img | sort -rn -k 9 > sorted_files.txt
```

The Shell script

Shell scripts are text files of variables, functions, and commands

A ‘shebang’ (#!/bin/bash, …) is required to indicate which interpreter the OS programs loader should use

Conditional expressions: if/else, case
Loops: for, while, until

http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html

Scripting and Scientific Computing

Why *wouldn’t* you want to script data processing?

- Dealing with intrinsically visual data (density interpretation, particle picking, etc)
- Using programs that are GUI only

Scripting and Scientific Computing

Why would you want to script data processing?

- Documenting *how* a dataset was processed, and *why* particular options were used
- Easier to process several datasets identically (e.g. - comparing apo structure and complex structure)
- Easier to explore alternative ways to process (e.g. - MR/density fitting for 50 models

Scripting and Scientific Computing

```
#!/bin/bash



process_my_data input*.mrc output.mrc
```

Scripting and Scientific Computing

```
#!/usr/bin/env bash

# first stage of my data processing

process_my_data input*.mrc output.mrc > processing.log
```

Scripting and Scientific Computing

```
#!/usr/bin/env bash
# first stage of my data processing
job=ProcessingStage01
inp="input*.mrc"
opf="${job}-output.mrc"
process_my_data $inp $opf > ${job}.log
```

Scripting and Scientific Computing

```
#!/usr/bin/env bash
# first stage of my data processing
#tuning parameter
tune_param="0.2"
# pipeline flag
flag="EvalReconstruct"
job=ProcessingStage01
inp="input*.mrc"
opf="${job}-output.mrc"
process_my_data --input $inp --output $opf --tune $tune_param << eof > ${job}.log
PIPELINE_FLAG $flag
eof
```

Scripting and Scientific Computing

```
#!/usr/bin/env bash

job=TestScan

input_dir="models/"
output_dir="results/"
map=input.mrc

for model in `ls $input_dir/*.pdb | awk -F. '{print $1}'`
do
     search_density_for_model --map $map --search_model ${input_dir}/${model}.pdb --output
${output_dir}/${model}_results.out
done
```

Scientific Computing: tools, tips and tricks

- Getting there and moving data
- What resources does this computer have (and what is it doing)?
- Reproducibility and collaboration

Scientific Computing: tools, tips and tricks

SSH

provides a secure channel (encrypted) over an unsecured network in a client-server architecture

- remote command-line login
- remote command execution
- any network service can be secured with SSH.

Scientific Computing: SSH

SSH public-key authentication allows login and command execution without passwords based on a public/private key pair

Setup: create keys, set a password

```
ssh-keygen -t rsa
```

Public key goes on the remote server in your .ssh directory in the file $HOME/.ssh/authorized_keys

Private key stays in $HOME/.ssh

Ssh-agent manages keys - typically running by default on most Linuxes

Use ssh-add command to add key, **No more passwords**!

Scientific Computing: SSH

SSH public-key authentication allows login and command execution without passwords based on a public/private key pair

Use ssh -X to forward X11 for graphics access

Execute code remotely with a single command

Scientific Computing: Moving data with rsync

rsync

- Transfers only changes in a file tree
- Local and remote synchronization of data file and directories

```
rsync -rv /my/files/here/ /my/files/over_there/
```

- Can be run over ssh for secure transfer

```
rsync -rv /my/files/here/ remote.server.org:/my/files/over_there/
```

- Ideal for data backup

```
man rsync
```

For more info

Scientific Computing: Hardware

CPU, Storage,memory,usb and pci

- CPU : cpuinfo or cat /proc/cpu
- DISK : df or lsblk
- MEMORY : free
- USB : lsusb
- PCI (internal cards, etc) : lspci

Scientific Computing: What is running

CPU and memory use, jobs, IO

- top, uptime
- ps
- sar

Scientific Computing: history

history

The shell records all commands.
This record can be accessed with the ‘history’ command.

Some relevant variables:

```
    HISTSIZE
        Define number of commands

    HISTFILE
        Define file

    HISTCONTROL=ignoredups
        Ignore duplicates
```

Scientific Computing: Terminal multiplexer

Tmux (or screen)

A terminal multiplexer is terminal-based program that gives the user

- Ability to detach and reattach sessions from a terminal
  - Sessions persist on the remote machine
  - A terminal session can be accessed from multiple machines
  - Persist through network disconnection

- Multiple separate login sessions inside a single terminal window

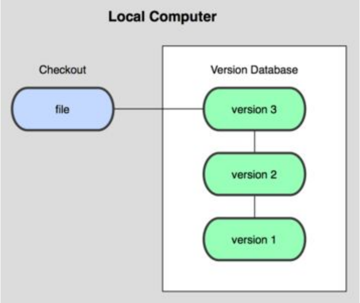
Scientific Computing: Version Control

VCS

Version control systems are designed for software development, are are great for scientific computing projects

Version control software keeps track of every modification to the code

Earlier versions of code are retained and can be accessed

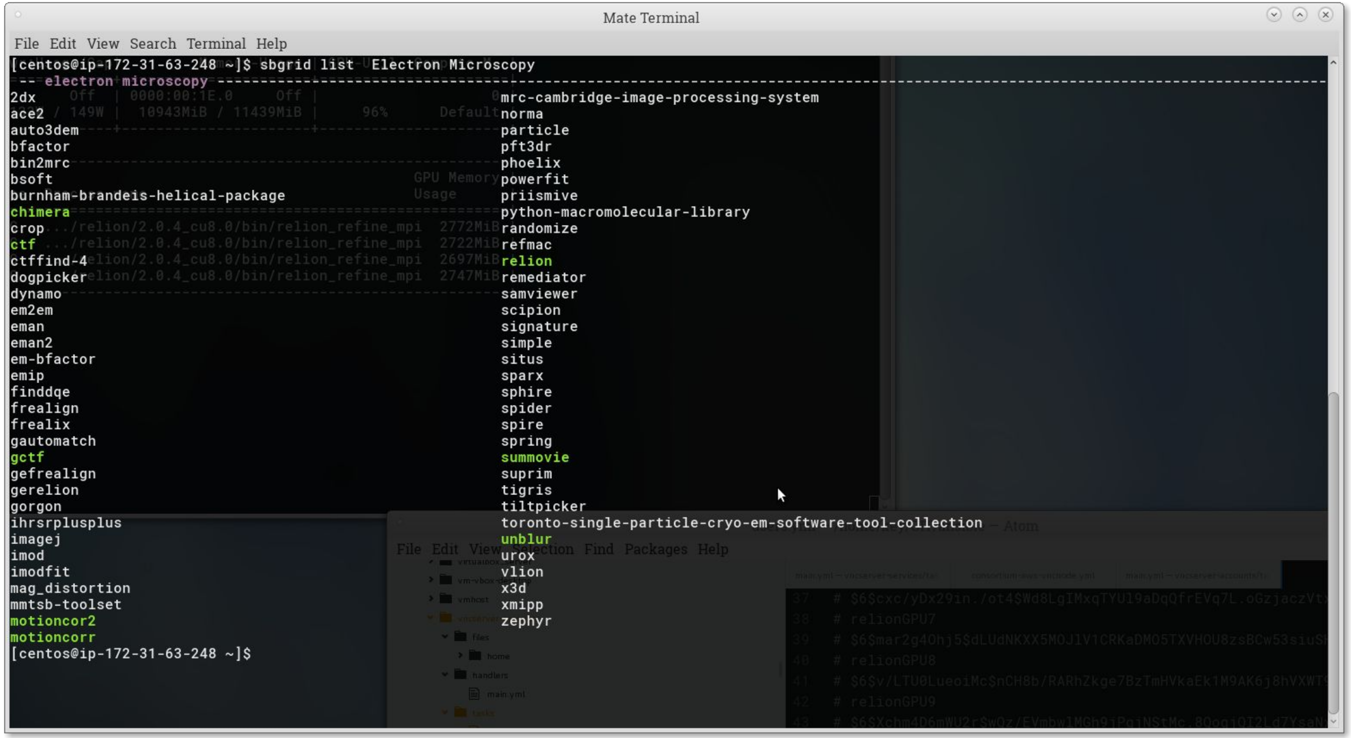


Scientific Computing: Getting started with git

More GIT
Gitlab, github bitbucket, RELION is in GIT

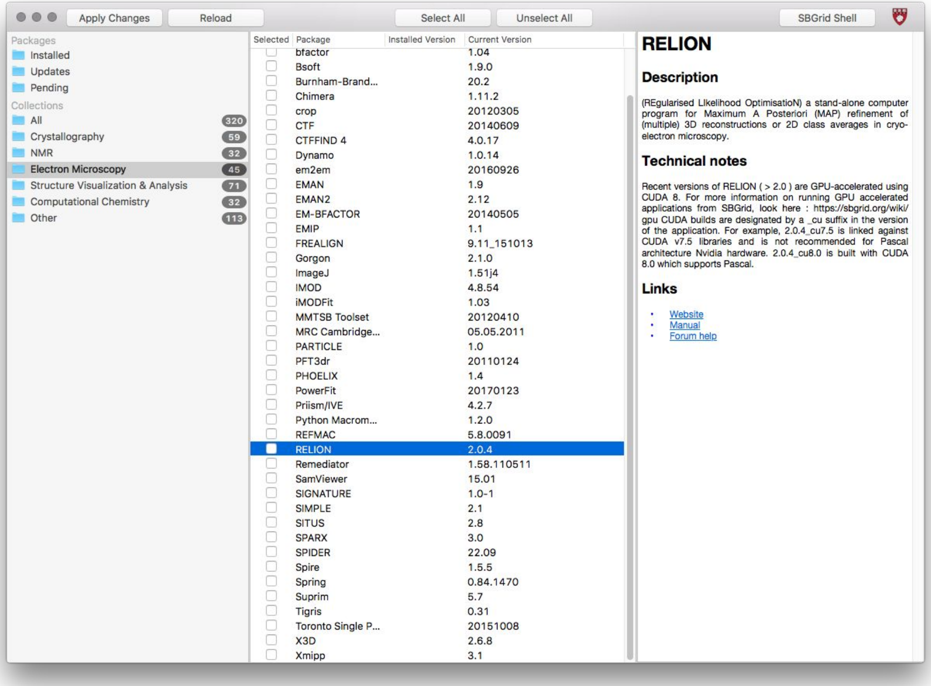
https://git-scm.com/book/en/v1/Getting-Started

Scientific Computing: Install Client CLI



```
[centos@ip-172-31-63-248 ~]$ sbgrid list  Electron Microscopy
-- electron microscopy ------------------------------------------------------------------------------------------------------
2dx                                                      mrc-cambridge-image-processing-system
ace2                                                     norma
auto3dem                                                 particle
bfactor                                                  pft3dr
bin2mrc                                                  phoelix
bsoft                                                    powerfit
burnham-brandeis-helical-package                         priismive
chimera                                                  python-macromolecular-library
crop                                                     randomize
ctf                                                      refmac
ctffind-4                                                relion
dogpicker                                                remediator
dynamo                                                   samviewer
em2em                                                    scipion
eman                                                     signature
eman2                                                    simple
em-bfactor                                               situs
emip                                                     sparx
finddqe                                                  sphire
frealign                                                 spider
frealix                                                  spire
gautomatch                                               spring
gctf                                                     summovie
gefrealign                                               suprim
gerelion                                                 tigris
gorgon                                                   tiltpicker
ihrsrplusplus                                            toronto-single-particle-cryo-em-software-tool-collection
imagej                                                   unblur
imod                                                     urox
imodfit                                                  vlion
mag_distortion                                           x3d
mmtsb-toolset                                            xmipp
motioncor2                                               zephyr
motioncorr
[centos@ip-172-31-63-248 ~]$
```

Scientific Computing: Install client GUI



Packages

- Installed
- Updates
- Pending

Collections

All	320
Crystallography	59
NMR	32
Electron Microscopy	45
Structure Visualization & Analysis	71
Computational Chemistry	32
Other	113

Selected	Package	Installed Version	Current Version
□	bfactor		1.04
□	Bsoft		1.9.0
□	Burnham-Brand...		20.2
□	Chimera		1.11.2
□	crop		20120305
□	CTF		20140609
□	CTFFIND 4		4.0.17
□	Dynamo		1.0.14
□	em2em		20160926
□	EMAN		1.9
□	EMAN2		2.12
□	EM-BFACTOR		20140505
□	EMIP		1.1
□	FREALIGN		9.11_151013
□	Gorgon		2.1.0
□	ImageJ		1.51j4
□	IMOD		4.8.54
□	iMODFit		1.03
□	MMTSB Toolset		20120410
□	MRC Cambridge...		05.05.2011
□	PARTICLE		1.0
□	PFT3dr		20110124
□	PHOELIX		1.4
□	PowerFit		20170123
□	Priism/IVE		4.2.7
□	Python Macrom...		1.2.0
□	REFMAC		5.8.0091
□	RELION		2.0.4
□	Remediator		1.58.110511
□	SamViewer		15.01
□	SIGNATURE		1.0-1
□	SIMPLE		2.1
□	SITUS		2.8
□	SPARX		3.0
□	SPIDER		22.09
□	Spire		1.5.5
□	Spring		0.84.1470
□	Suprim		5.7
□	Tigris		0.31
□	Toronto Single P...		20151008
□	X3D		2.6.8
□	Xmipp		3.1

RELION

Description

(REgularised LIkelihood OptimisatioN) a stand-alone computer program for Maximum A Posteriori (MAP) refinement of (multiple) 3D reconstructions or 2D class averages in cryo-electron microscopy.

Technical notes

Recent versions of RELION (> 2.0) are GPU-accelerated using CUDA 8. For more information on running GPU accelerated applications from SBGrid, look here : https://sbgrid.org/wiki/gpu CUDA builds are designated by a _cu suffix in the version of the application. For example, 2.0.4_cu7.5 is linked against CUDA v7.5 libraries and is not recommended for Pascal architecture Nvidia hardware. 2.0.4_cu8.0 is built with CUDA 8.0 which supports Pascal.

Links

- Website
- Manual
- Forum help