

Introduction to Machine Learning with the Tufts High Performance Research Cluster

Sergey Voronin

Department of Mathematics, Tufts University
In collaboration with Tufts Technology Services

16 May 2017

Thanks to

- Shawn Doughty, Sr. Research Technology Specialist
- Durwood Marshall, Fmr Sr. Research Technology Specialist
- Lionel Zupan, Dir. Research Technology
- David Lapointe, Sr. Bioinformatics Specialist

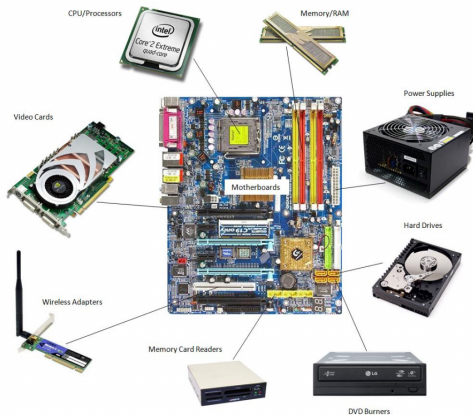
Agenda

- Layout of a cluster.
- Connection and file transfer.
- Slurm resource manager.
- Intro to machine learning.
- Using KNN and SVMs with R and Python.
- Applying deep learning to the ESC data set from scratch with Java and Python.

1. Anatomy of a computer and of a computer cluster

- Fundamental computer parts
- Layout of different nodes in a typical research cluster

Computer components

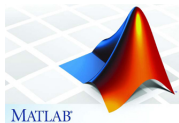


- CPU for computation
- RAM and CPU cache for intermediate process storage
- Hard drive for permanent storage

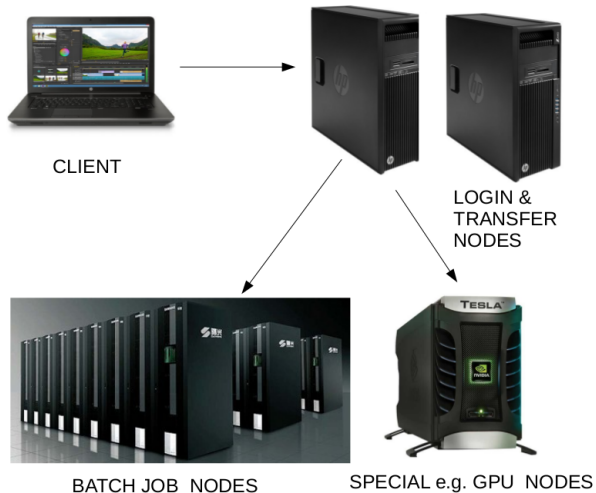
Computing systems and software



- Different problems demand various CPU, RAM, and disk resources.
- Various software maybe useful (some requires hefty licensing fees).



Layout of the computational cluster



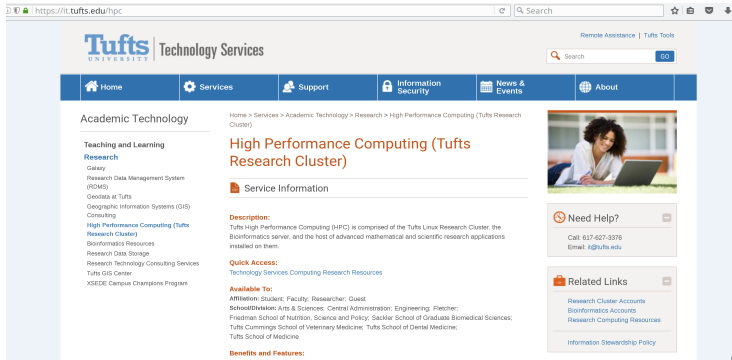
⇒ Don't run computations on login and xfer nodes. These are special nodes used to process logins and for file transfer.

2. Connection and file transfer

- Getting an account
- ssh connection fundamentals
- File transfer via scp and rsync
- X11 server
- Windows and Mac OS specifics

Getting an account on the cluster

- See <https://it.tufts.edu/hpc>.
- Email tts-research@tufts.edu.
- Your username and password will be the same as your Tufts credentials.

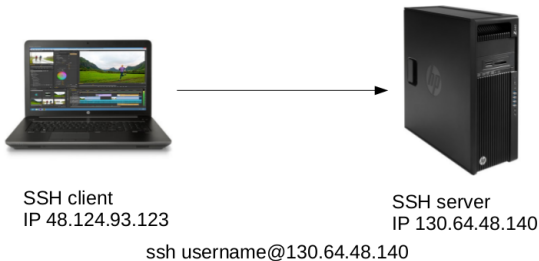


The screenshot shows a web browser window displaying the Tufts University Technology Services website. The URL in the address bar is <https://it.tufts.edu/hpc>. The page features a navigation menu with links for Home, Services, Support, Information Security, News & Events, and About. The main content area is titled "Academic Technology" and includes a sidebar with "Teaching and Learning Research" and a list of services such as Research Data Management System (RDMS), Geodata at Tufts, and Geographic Information Systems (GIS) Consulting. The main heading is "High Performance Computing (Tufts Research Cluster)", followed by "Service Information". A description states: "Tufts High Performance Computing (HPC) is comprised of the Tufts Linux Research Cluster, the Bioinformatics server, and the host of advanced mathematical and scientific research applications installed on them." A "Quick Access" section lists "Technology Services Computing Research Resources". An "Available To:" section lists various departments and schools, including the Friedman School of Nutrition, Science and Policy, Sackler School of Graduate Biological Sciences, Tufts Cummings School of Veterinary Medicine, and Tufts School of Dental Medicine. A "Benefits and Features:" section is partially visible. On the right side, there is a "Need Help?" section with contact information (Call: 617-627-3376, Email: it@tufts.edu) and a "Related Links" section with links to Research Cluster Accounts, Bioinformatics Accounts, Research Computing Resources, and Information Stewardship Policy.

Essential software

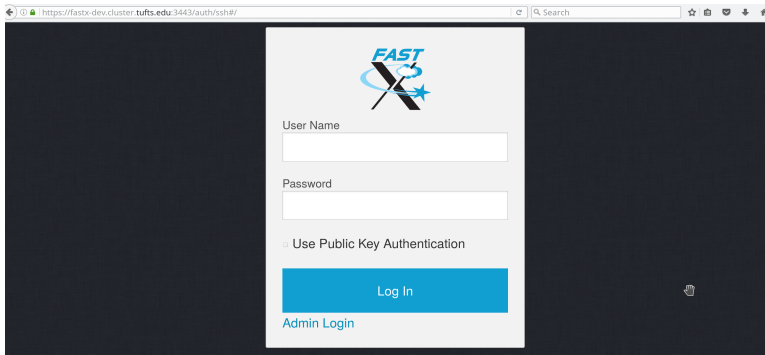
- SSH connection software.
- SCP/RSYNC file transfer software.
- X11 Xorg server software for tunneling graphical applications.

Client to server connection via SSH protocol



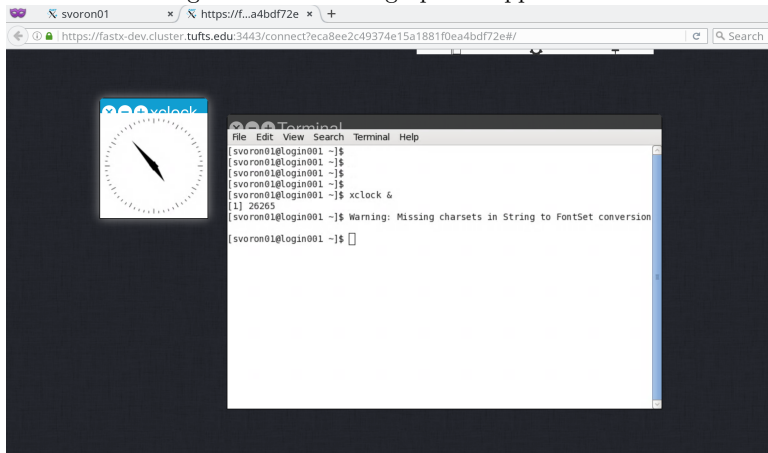
fastX web interface

- Go to `https://fastx-dev.cluster.tufts.edu:3443/`.
- Contact Shawn Doughty `shawn.doughty@tufts.edu`.
- Your username and password will be the same as your Tufts credentials.



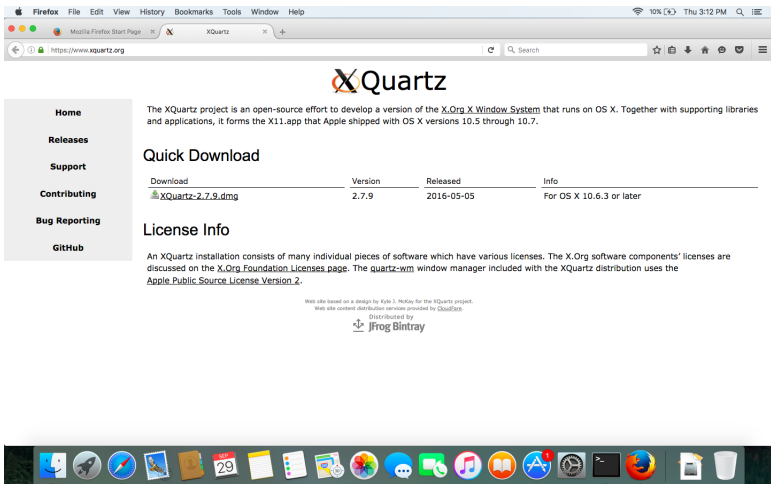
Using SSH with X11 via fastX web interface

Login and run xlock graphical application.



For general cluster use, this is sufficient. But it is useful to know how to do this on Windows, Mac, and Linux.

Using SSH with X11 on Mac OS X



The screenshot shows a Firefox browser window displaying the XQuartz website. The browser's address bar shows the URL `https://www.xquartz.org`. The website has a navigation sidebar on the left with links for Home, Releases, Support, Contributing, Bug Reporting, and GitHub. The main content area features the XQuartz logo, a description of the project as an open-source effort to develop a version of the X.Org X Window System for OS X, and a 'Quick Download' section. The download section includes a table with columns for Download, Version, Released, and Info. Below the table is a 'License Info' section. At the bottom of the page, there is a small note about the website's design and distribution, along with the JFrog Bintray logo. The browser's status bar at the top shows the system time as 3:12 PM on Thursday.

Firefox File Edit View History Bookmarks Tools Window Help 10% [X] Thu 3:12 PM


Mozilla Firefox Start Page X XQuartz

`https://www.xquartz.org` Search

XQuartz

The XQuartz project is an open-source effort to develop a version of the [X.Org X Window System](#) that runs on OS X. Together with supporting libraries and applications, it forms the X11.app that Apple shipped with OS X versions 10.5 through 10.7.


Quick Download

Download	Version	Released	Info
 XQuartz-2.7.9.dmg	2.7.9	2016-05-05	For OS X 10.6.3 or later

License Info

An XQuartz installation consists of many individual pieces of software which have various licenses. The X.Org software components' licenses are discussed on the [X.Org Foundation Licenses page](#). The `quartz-wm` window manager included with the XQuartz distribution uses the [Apple Public Source License Version 2](#).

Web site based on a design by Kyle J. McKay for the XQuartz project.
Web site content distribution services provided by [CloudFlare](#).

Distributed by
 JFrog Bintray

Mac OS X desktop dock with various application icons.

Using SSH with X11 on Mac OS X

The screenshot shows a Firefox browser window displaying the XQuartz website at <https://www.xquartz.org>. The page features a navigation sidebar on the left with links for Home, Releases, Support, Contributing, Bug Reporting, and GitHub. The main content area includes a 'Quick Download' section with a table of download links, a 'License Info' section, and a footer with attribution to Kyle J. McKay and JFrog Bintray.

The browser's address bar shows the URL <https://www.xquartz.org>. The page title is 'XQuartz'. The browser's download manager is open, showing a file named 'XQuartz-2.7.9.dmg' (74.2 MB) downloaded from 'xquartz.org' at 3:13 PM. A dialog box titled 'Opening "XQuartz-2.7.9.dmg"...' is displayed over the download manager, with a progress bar and 'Cancel' and 'Skip' buttons. Another dialog box titled 'Verifying...' is also visible.

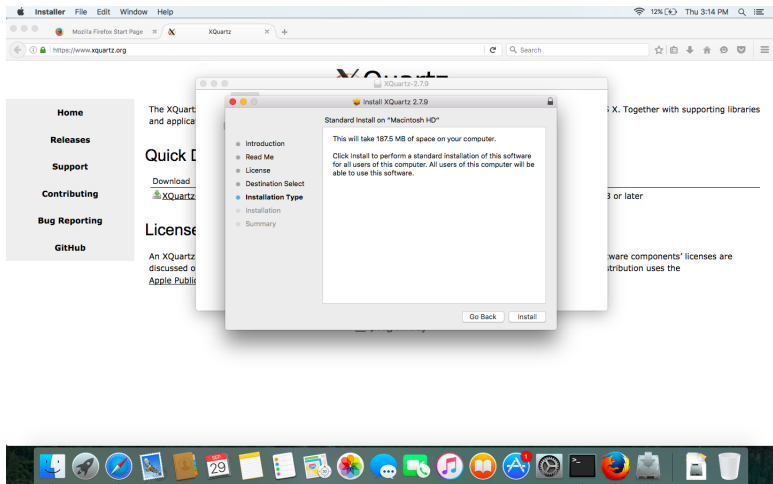
The 'Quick Download' table contains the following information:

Download	Version	Release Date	Info
XQuartz-2.7.9.dmg	2.7.9	2016-05-05	For OS X 10.6.3 or later

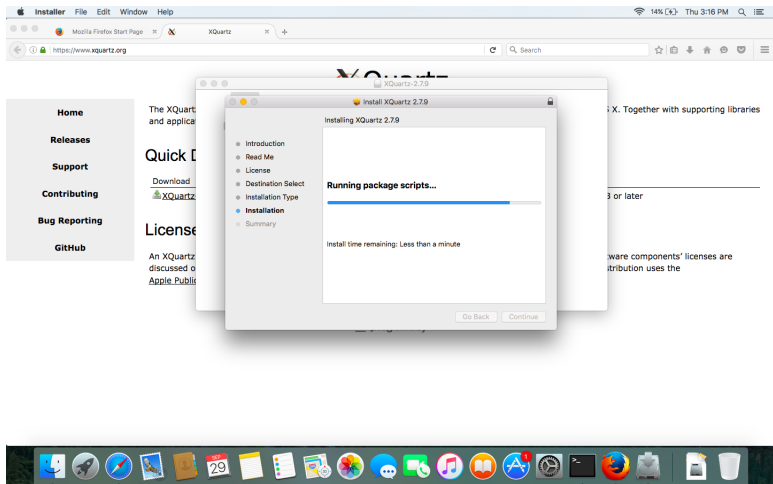
The 'License Info' section states: "An XQuartz installation consists of many individual pieces of software which have various licenses. The X.Org software components' licenses are discussed on the [X.Org Foundation Licenses page](#). The `quartz-wm` window manager included with the XQuartz distribution uses the [Apple Public Source License Version 2](#)."

The footer includes the text: "Web site based on a design by Kyle J. McKay for the XQuartz project. Web site content distribution services provided by [CloudFlare](#). Distributed by [JFrog Bintray](#)".

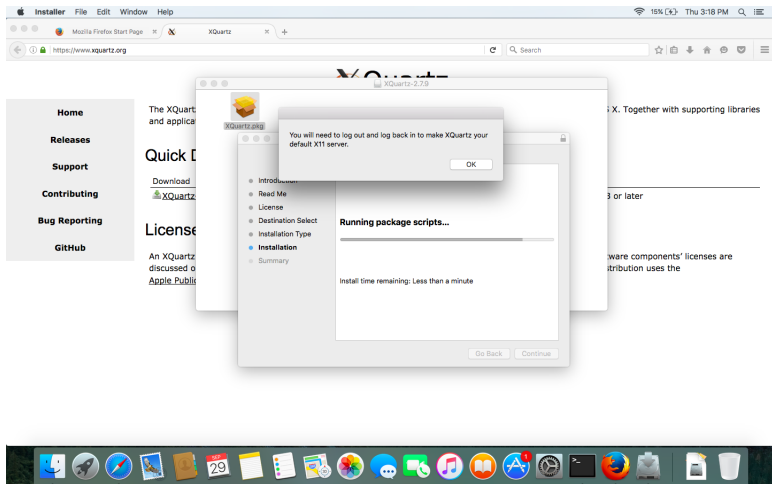
Using SSH with X11 on Mac OS X



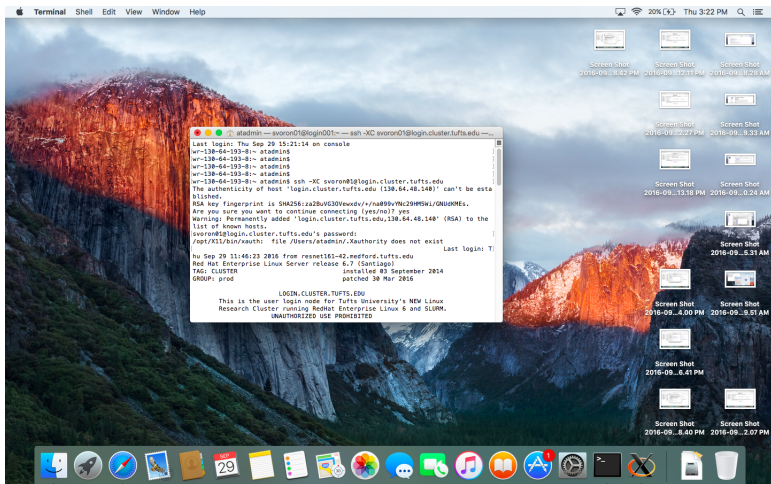
Using SSH with X11 on Mac OS X



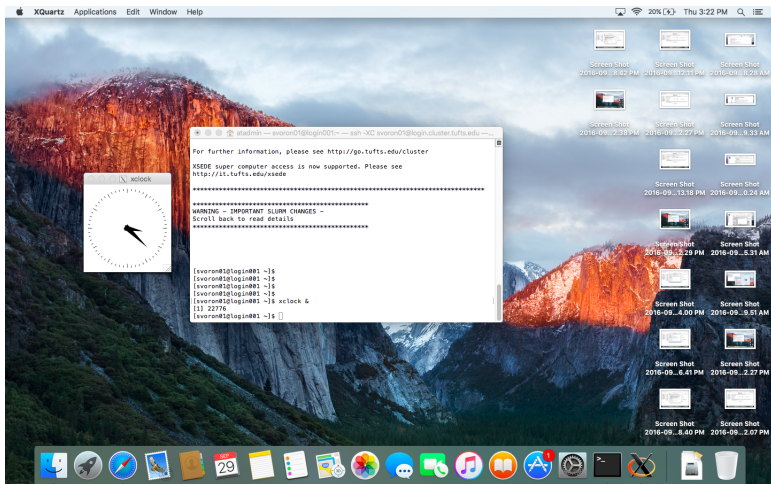
Using SSH with X11 on Mac OS X



Using SSH with X11 on Mac OS X



Using SSH with X11 on Mac OS X



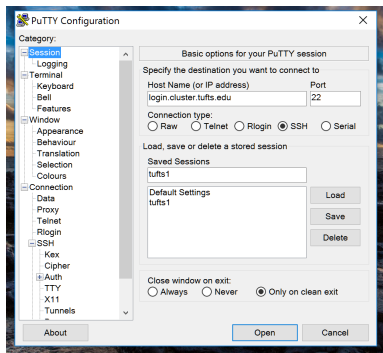
Using SSH with X11 on Windows 10



Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon T source software that is available with source code and is develop

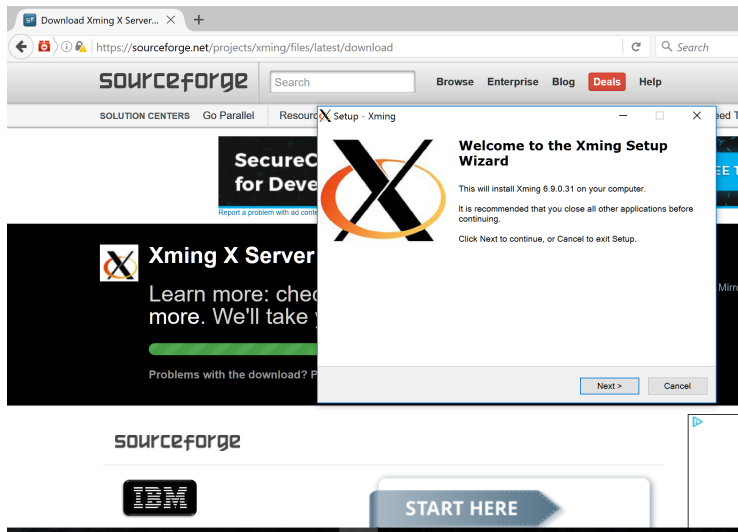
You can download PuTTY [here](#).



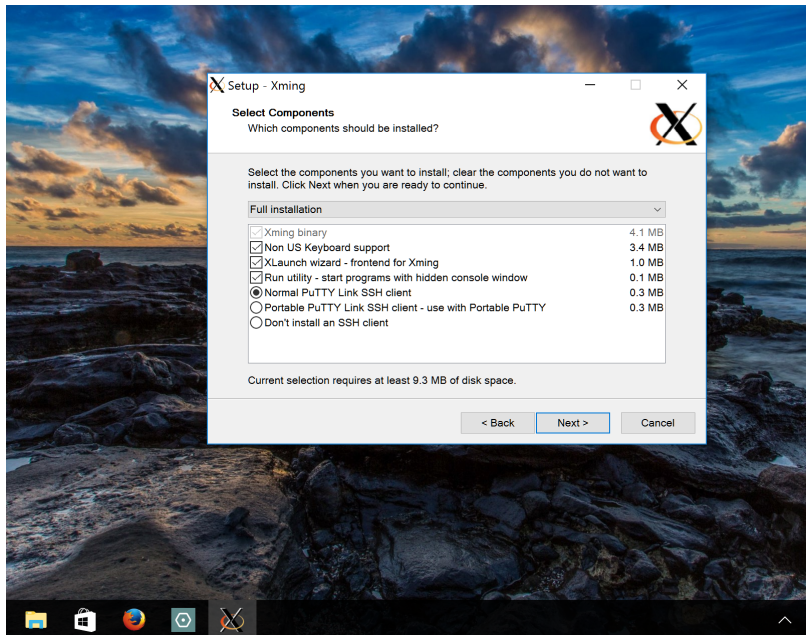
Using SSH with X11 on Windows 10

The screenshot shows a web browser window with the address bar displaying `https://sourceforge.net/projects/xming/`. The SourceForge logo is prominent at the top left, with a search bar and navigation links for 'Browse', 'Enterprise', 'Blog', 'Deals', and 'Help'. Below the navigation bar, there are links for 'SOLUTION CENTERS', 'Go Parallel', 'Resources', 'Newsletters', 'Cloud Storage Providers', 'Business VoIP Providers', and 'Internet Speed'. A large blue banner for Intel Software is visible, with the text 'IOT: FROM START TO FINISH' and 'See how Intel bridges the gap between concept and reality with this real-world example.' Below the banner, the breadcrumb trail reads 'Home / Browse / Development / Terminal Emulators/X Terminals / Xming X Server for Windows'. The main heading is 'Xming X Server for Windows' with the subtitle 'X Window System Server for Windows'. It is attributed to 'colinharrison'. The page has tabs for 'Summary', 'Files', 'Reviews', and 'Support'. The 'Summary' tab is active, showing '17,289 Downloads (This Week)' and 'Last Update: 2016-08-09'. There are social media sharing buttons for 'Tweet', 'G+', and 'Like 37'. A green 'Download' button is present, with the text 'Xming-6-9-0-31-setup.exe' below it. A 'Browse All Files' link is also visible. A partial advertisement on the right side says 'Get out center a board r'.

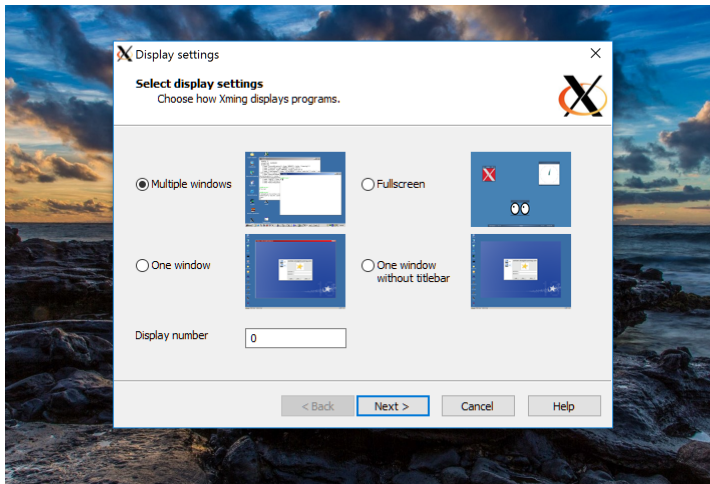
Using SSH with X11 on Windows 10



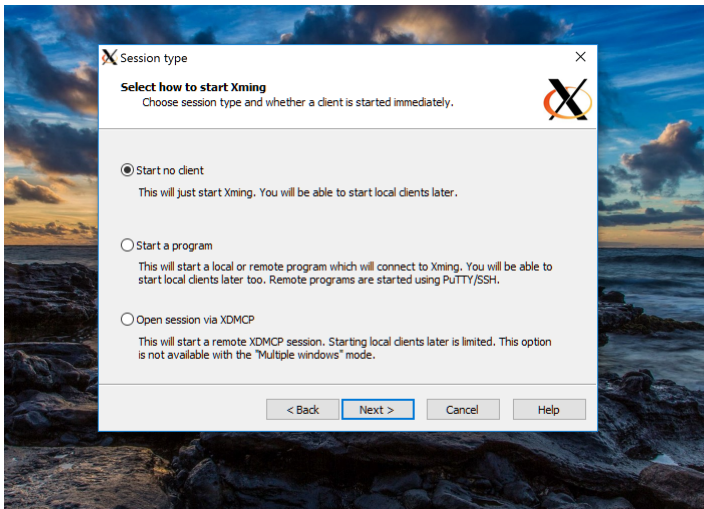
Using SSH with X11 on Windows 10



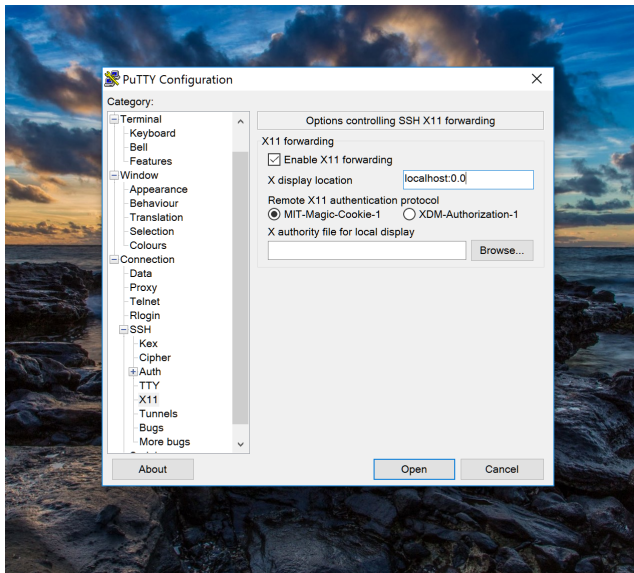
Using SSH with X11 on Windows 10



Using SSH with X11 on Windows 10

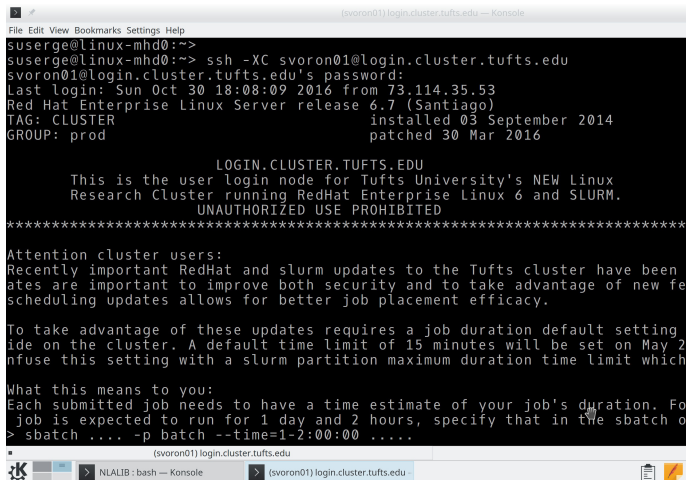


Using SSH with X11 on Windows 10



Linux and Mac

- In terminal: `ssh -CX username @ server address`
- '-C' option for compression
- '-X' option for X11 server forwarding



```
(svoron01) login.cluster.tufts.edu — Konsole
File Edit View Bookmarks Settings Help
suserge@linux-mhd0:~>
suserge@linux-mhd0:~> ssh -XC svoron01@login.cluster.tufts.edu
svoron01@login.cluster.tufts.edu's password:
Last login: Sun Oct 30 18:08:09 2016 from 73.114.35.53
Red Hat Enterprise Linux Server release 6.7 (Santiago)
TAG: CLUSTER                               installed 03 September 2014
GROUP: prod                                 patched 30 Mar 2016

      LOGIN.CLUSTER.TUFTS.EDU
      This is the user login node for Tufts University's NEW Linux
      Research Cluster running RedHat Enterprise Linux 6 and SLURM.
      UNAUTHORIZED USE PROHIBITED
      *****

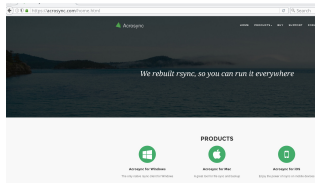
Attention cluster users:
Recently important RedHat and slurm updates to the Tufts cluster have been
ates are important to improve both security and to take advantage of new fe
scheduling updates allows for better job placement efficacy.

To take advantage of these updates requires a job duration default setting
ide on the cluster. A default time limit of 15 minutes will be set on May 2
nfuse this setting with a slurm partition maximum duration time limit which

What this means to you:
Each submitted job needs to have a time estimate of your job's duration. Fo
job is expected to run for 1 day and 2 hours, specify that in the sbatch o
> sbatch ... -p batch --time=1-2:00:00 .....
```

scp and rsync for file transfer

- Two related protocols to transfer data between client and server.
- scp to transfer large files (all at once).
- rsync to transfer incremental changes or for backups.



scp and rsync for file transfer

- Similar syntax to ssh, transfer file via ssh protocol.
- `scp username@server:/path/to/file/on/server /local/path`
- The '-r' (recursive) option is used to transfer directories.
- rsync can be used to transfer incremental changes (only transfers what has changed, useful for backups and code syncs). This can save a lot of time over scp transfer!
- Use **xfer node** for file transfers. Avoid the use of the login node for this.
- Command line examples on Linux and Mac:

```
svoronin@linux-mhd0:~> scp svoron01@xfer.cluster.tufts.edu:~/scripts/socket_script.pl .
svoronin@linux-mhd0:~> scp -r svoron01@xfer.cluster.tufts.edu:~/scripts/ .
svoronin@linux-mhd0:~> rsync -avp scripts/ svoron01@xfer.cluster.tufts.edu:~/scripts/
```

3. Slurm resource manager and examples

- Resource requests
- Interactive work and batch scripts
- Different partitions (interactive, batch, cpu)
- Examples with image denoising, matrix multiply.

SLURM

Simple Linux Utility for Resource Management (getting cluster resources).
(Command table credit: thanks to David Lapointe).

Action	Slurm
Show running jobs	squeue
Submit a batch job	sbatch
Submit with allocations	salloc
Start an interactive session	srun
List queues/partitions	squeue
List nodes	sinfo
Control running jobs	scontrol
Kill a job	scancel
User accounting	sreport or sacct
Other functions	srun

Available Partitions

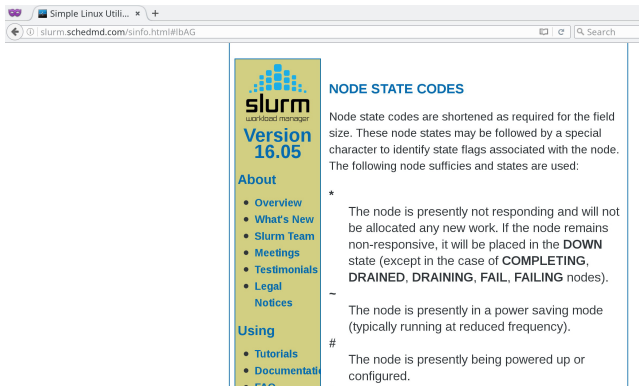
```
[svoron01@login001 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
gpu        up 3-00:00:00    2   idle alpha025,omega025
largemem   up 7-00:00:00    4   mix  alpha[022-023],omega[022-023]
interactive up 4:00:00:00    1   comp alpha001
interactive up 4:00:00:00    1   mix  omega001
interactive up 4:00:00:00    2   idle alpha025,omega025
batch*     up 3-00:00:00    1   down* alpha027
batch*     up 3-00:00:00    2   drain m3n[52,54]
batch*     up 3-00:00:00   31   mix  alpha[002-008,018,022-024,028-030],m3n[22-23,43-45,49-51],m4lmem01,omega[012,022-023,026-030]
batch*     up 3-00:00:00   26   alloc alpha[009-017,019-021,026,031],omega[002-011,024,031]
batch*     up 3-00:00:00   52   idle m3n[01-21,24-38,40-42,46-48,53],omega[013-021]
mpi        up 7-00:00:00    1   down* alpha027
mpi        up 7-00:00:00   17   mix  alpha[003-008,018,024,028-030],omega[012,026-030]
mpi        up 7-00:00:00   25   alloc alpha[009-017,019-021,026,031],omega[003-011,024,031]
mpi        up 7-00:00:00    9   idle omega[013-021]
m4         up 7-00:00:00   59   alloc m4c[01-59]
m4         up 7-00:00:00    1   idle m4c60
[svoron01@login001 ~]$
```

The largemem nodes have 396 GB of memory.

The gpu nodes have an integrated gpu computational card.

The mpi nodes have fast interconnections.

Slurm node states



The screenshot shows a web browser window with the URL `slurm.schedmd.com/sinfo.html#lbAG`. The page content includes the Slurm logo (workload manager Version 16.05) and a navigation menu with sections for 'About' and 'Using'. The main content area is titled 'NODE STATE CODES' and explains that node state codes are shortened and may be followed by a special character to identify state flags. It lists three node states: '*' (not responding), '~' (power saving mode), and '#' (being powered up or configured).

NODE STATE CODES

Node state codes are shortened as required for the field size. These node states may be followed by a special character to identify state flags associated with the node. The following node suffixes and states are used:

- * The node is presently not responding and will not be allocated any new work. If the node remains non-responsive, it will be placed in the **DOWN** state (except in the case of **COMPLETING**, **DRAINED**, **DRAINING**, **FAIL**, **FAILING** nodes).
- ~ The node is presently in a power saving mode (typically running at reduced frequency).
- # The node is presently being powered up or configured.

ALLOCATED: The node has been allocated to one or more jobs.

DRAINED: The node is unavailable for use per system administrator request.

DRAINING: The node is currently executing a job, but will not be allocated to additional jobs.

Node details

```
[svoron01@login001 ~]$ sinfo -N -l
Mon May  8 00:19:37 2017
NODELIST      NODES  PARTITION      STATE  CPUS    S:C:T  MEMORY
alpha001      1  interactive    idle   40     2:10:2 129054
alpha002      1    testing     drained 40     2:10:2 129054
alpha003      1      mpi        mixed  40     2:10:2 129054
alpha003      1    batch*     mixed  40     2:10:2 129054
alpha004      1      mpi        mixed  40     2:10:2 129054
alpha004      1    batch*     mixed  40     2:10:2 129054
```

```
[svoron01@login001 ~]$ sinfo -o "%15N %10c %10m %25f %10G"
NODELIST      CPUS    MEMORY    FEATURES      GRES
alpha025,omega0 40     128915    usnic,gpu     gpu
alpha[001-021,0 40     128858+   usnic         (null)
m3n[01-38,40-51 12     15937+   M3            (null)
alpha[022-023,0 40     258130+   usnic,bigmem  (null)
m4lmem01       16     387581    (null)        (null)
m4c[01-60]     16     32079+    M4            (null)
[svoron01@login001 ~]$
```

Available software

The cluster has many different installed libraries and software packages. Run command: `module avail` to see available packages. The command `module load [package]` sets up the environmental variables for the use of package.

```
[svoron01@login001 ~]$ module avail
```

```
openmpi/1.8.6
OpenSees/r4985(default)
OpenSees/r4985-MP
paml/4.8
pandaseq/2.5(default)
paraview/3.8.1
paraview/4.4

python/2.7.6
python/3.5.0

R/3.2.2
R/3.2.5
R/3.3.2

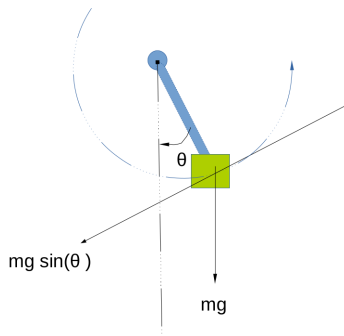
tophat/2.0.9(default)
transdecoder/2.0
trinity/10.15.15
trinotate/2.0.2
ufc/2.3.0(default)
umfpack/5.5.1
usearch/7.0.1001(default)
VASP/4.6
VASP/5.4.1
vcftools/0.1.12b(default)
velvet/1.0.19
velvet/1.2.10
ViennaRNA/2.1.6(default)
visit/2.6.2
vmd/1.9
Wavelab/850
weka/3-6-10(default)
wpp/1.1-2008.0/optimize
wpp/1.1-2008.0/optimize-min
wpp/2.0-2010.0/optimize
wpp/2.0-2010.0/optimize-min
wpp/2.1.5
yap/5.1.3
[svoron01@login001 ~]$
```

Using Maple

Maple is full featured package and CAS for analytical and numerical analysis. We will use it for modeling pendulum motion.

- (1) `salloc -N1 -c4 -t 100 -p interactive`
- (2) `squeue -u username; ssh -XC granted_node`
- (3) `module load maple`
- (4) `$ xmaple &`

Modeling pendulum motion



$$F = ma \implies -\alpha v - mg \sin(\theta) = ma \implies a = -\frac{\alpha}{m}v - g \sin(\theta) \quad (1)$$

It remains to express the acceleration a in terms of l and θ . Let us take s to be the arc length along the circular direction of motion. Then we have that:

$$\frac{s}{2\pi l} = \frac{\theta}{2\pi} \implies s = l\theta \implies v = \frac{ds}{dt} = l \frac{d\theta}{dt} \implies a = \frac{d^2 s}{dt^2} = l \frac{d^2 \theta}{dt^2} \quad (2)$$

Plugging in for v and a in terms of θ , we obtain:

$$l \frac{d^2\theta}{dt^2} = -\frac{\alpha}{m} l \frac{d\theta}{dt} - g \sin(\theta) \implies \frac{d^2\theta}{dt^2} = -\frac{\alpha}{m} \frac{d\theta}{dt} - \frac{g}{l} \sin(\theta)$$

and the IVP:

$$\frac{d^2\theta}{dt^2} + \frac{\alpha}{m} \frac{d\theta}{dt} + \frac{g}{l} \sin(\theta) = 0 \quad ; \quad \theta(0) = \theta_0 \quad \text{and} \quad \frac{d\theta}{dt}(0) = v_0 \quad (3)$$

We first rewrite the IVP as a first order system, by defining $p = \theta$ and $q = \theta'$:

$$\begin{aligned} \frac{dp}{dt} &= q \\ \frac{dq}{dt} &= -\frac{\alpha}{m} q - \frac{g}{l} \sin(p) \\ p(0) &= \theta_0 \quad \text{and} \quad q(0) = v_0 \end{aligned}$$

Next, we rewrite the system as a single first order ODE of a vector valued function. Letting:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ q(t) \end{bmatrix} \quad \text{and} \quad f(t) = \begin{bmatrix} u_2(t) \\ -\frac{\alpha}{m} u_2(t) - \frac{g}{l} \sin(u_1(t)) \end{bmatrix}$$

Modeling with maple

The screenshot shows the Maple software interface with the following content:

```
EQN1 := diff(theta(t), t, t) = -g/l * sin(theta(t));
```

$$\frac{d^2}{dt^2} \theta(t) = -9.8 \sin(\theta(t)) \quad (1)$$

```
EQN2 := diff(theta(t), t, t) = -g/l * theta(t);
```

$$\frac{d^2}{dt^2} \theta(t) = -9.8 \theta(t) \quad (2)$$

```
ics := theta(0) = theta_0, D(theta)(0) = v_0
```

$$\theta(0) = \theta_{0_0}, D(\theta)(0) = v_0 \quad (3)$$

```
sol2 := dsolve({EQN2, ics})
```

$$\theta(t) = \frac{1}{7} \sqrt{5} v_0 \sin\left(\frac{7}{5} \sqrt{5} t\right) + \theta_{0_0} \cos\left(\frac{7}{5} \sqrt{5} t\right) \quad (4)$$

```
sol2eval := eval(sol2, [v_0 = 0, theta_0 = 3.14/2, g = 9.8, l = 1]);
```

$$\theta(t) = 1.570000000 \cos\left(\frac{7}{5} \sqrt{5} t\right) \quad (5)$$

Modeling with maple

The screenshot shows the Maple software interface. On the left is a sidebar with various tool icons. The main workspace contains the following code and output:

```
plot(theta_sol, t = 0 .. 10)
```

alpha := 1; m := 1; l := 1; g := 9.8

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 9.8 \end{bmatrix} \quad (7)$$

```
SYS := array([[diff(p(t), t) = q(t)],  
              [diff(q(t), t) = -alpha/m*q(t) - g/l*sin(p(t))]]):  
print(SYS);
```

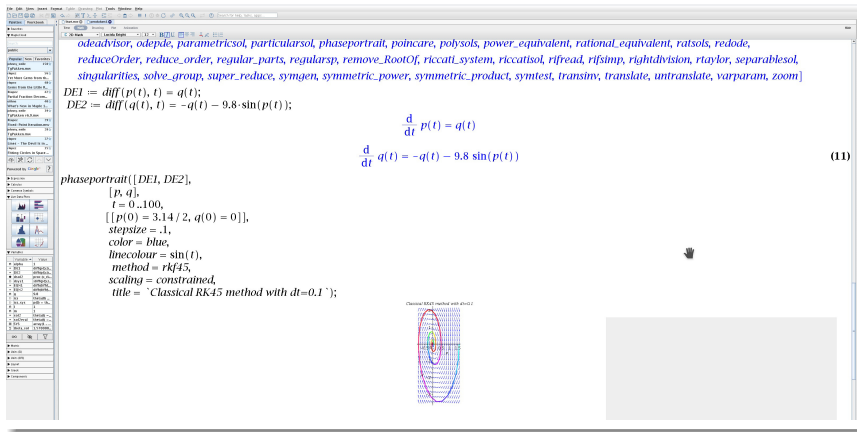
$$\begin{bmatrix} \frac{d}{dt} p(t) = q(t) \\ \frac{d}{dt} q(t) = -\frac{\alpha}{m} q(t) - \frac{g \sin(p(t))}{l} \end{bmatrix}$$

```
ics_sys := p(0) = theta_0, q(0) = v_0
```

$$p(0) = \text{theta}_0, q(0) = v_0$$

The plot shows a red sinusoidal wave oscillating between approximately -1.5 and 1.5 on the y-axis over the interval t from 0 to 10 on the x-axis.

Modeling with maple



The screenshot shows the Maple software interface. The main window contains the following code and output:

```
odeadvisor, odepde, parametricsol, particularsol, phaseportrait, poincare, polysols, power_equivalent, rational_equivalent, ratsols, redode,
reduceOrder, reduce_order, regular_parts, regularsp, remove_RootOf, riccati_system, riccatisol, rtfread, rtfsimp, righdivision, rtaylor, separablesol,
singularities, solve_group, super_reduce, symgen, symmetric_power, symmetric_product, symtest, transinv, translate, untranslate, varparam, zoom]
DE1 := diff(p(t), t) = q(t);
DE2 := diff(q(t), t) = -q(t) - 9.8*sin(p(t));
```

$$\frac{d}{dt} p(t) = q(t)$$
$$\frac{d}{dt} q(t) = -q(t) - 9.8 \sin(p(t)) \quad (11)$$

```
phaseportrait([DE1, DE2],
[p, q],
t = 0..100,
[[p(0) = 3.14/2, q(0) = 0]],
stepsize = .1,
color = blue,
linecolour = sin(t),
method = rkf45,
scaling = constrained,
title = `Classical RK45 method with step=2`);
```

The output shows a phase portrait plot titled "Classical RK45 method with step=2". The plot displays a trajectory in the (p, q) plane, starting at (0, 3.14/2) and oscillating around a point. The trajectory is colored according to the sine function of the angle p(t).

Using R, Python, Java

R is an open source package useful for statistical processing and data analysis. By installing some packages for R locally, we can use it for linear and non-linear model fitting. Example use:

- (1) `salloc -N1 -c4 -mem 30G -t 100 -p interactive` (4 cores, 30 GB total mem, for 100 min on interactive partition)
- (2) `squeue -u username; ssh -XC granted_node`
- (3) `module load R/3.3.2`
- (4) `R`
- (5) `$ > install.packages('class')`
- (6) `module load python`
- (7) `python2 test1.py`
- (8) `module load java/1.8.0_60`
- (9) `javac runAlg.java ; java runAlg`

Using R

Can use without or with X11 output (e.g. plotting).

```
svoronin@alpha001 ~]$ module load R/3.3.2
svoronin@alpha001 ~]$
svoronin@alpha001 ~]$
svoronin@alpha001 ~]$
svoronin@alpha001 ~]$ R

R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

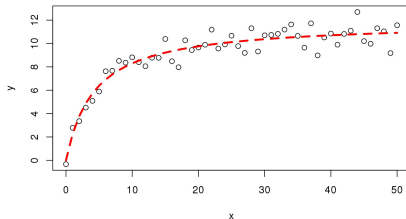
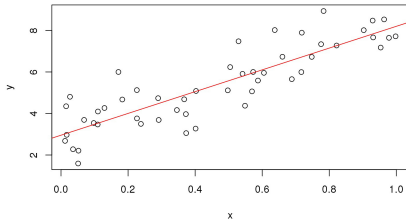
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> █
```



Using iPython

iPython is an interactive environment for testing and debugging your Python codes (similar to a Matlab/Octave prompt). First, run:

```
$ pip2 install --user ipython
```

Then, call from `.local` subdirectory:

```
[svoron01@alpha001 ipython-5.3.0]$ ~/.local/bin/ipython2
Python 2.7.6 (default, Jan 14 2014, 09:37:27)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]:

In [1]: import numpy as np

In [2]:


In [2]: np.random.uniform(-1,1,4)
Out[2]: array([-0.22273188, -0.04487004,  0.93704455,  0.90914968])

In [3]: █
```

Using Java

Can use Java with source code or (sometimes with GUI interface)

```
svoron01@login001 ~]$  
svoron01@login001 ~]$  
svoron01@login001 ~]$  
svoron01@login001 ~]$  
svoron01@login001 ~]$ ssh -XC alpha001  
Last login: Mon May  8 12:03:56 2017 from login001.lux.tufts.edu  
svoron01@alpha001 ~]$  
svoron01@alpha001 ~]$ cd storage/machine_learning_workshop/weka-3-8-1  
svoron01@alpha001 weka-3-8-1]$  
svoron01@alpha001 weka-3-8-1]$ module load java/1.8.0_60  
svoron01@alpha001 weka-3-8-1]$  
svoron01@alpha001 weka-3-8-1]$ java -jar weka.jar &  
1] 40795  
svoron01@alpha001 weka-3-8-1]$
```



Matrix multiplication in batch mode

$$\begin{array}{c} \text{row } i \hookrightarrow \end{array}
 \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{i1} & a_{i2} & a_{i3} & \dots & a_{in} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \cdot \begin{array}{c} \text{column } j \\ \downarrow \\ \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1j} & \dots & b_{1n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{i1} & b_{i2} & \dots & b_{ij} & \dots & b_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nj} & \dots & b_{nn} \end{bmatrix} \end{array} =$$

$$= \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1j} & \dots & c_{1n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & c_{i2} & \dots & c_{ij} & \dots & c_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nj} & \dots & c_{nn} \end{bmatrix} \begin{array}{c} \text{entry on row } i \\ \text{column } j \end{array}$$

Using sbatch

Often, (time-limited) interactive use is not necessary. This applies especially to developed code one wishes to run on the cluster. In this case, the batch interface is used.

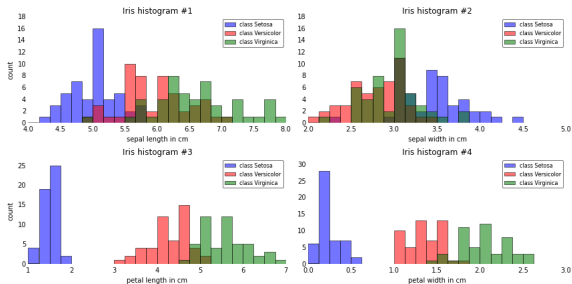
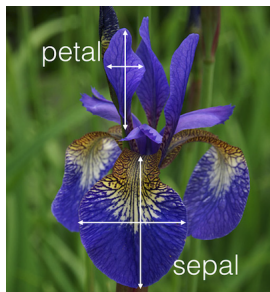
```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH -c 4
#SBATCH --mem-per-cpu=4000
#SBATCH --time 00:30:00
#SBATCH --output=mm_gpu.%N.%j.out
#SBATCH --error=mm_gpu.%N.%j.err
module load matlab
matlab -nodesktop -r "run('matrix_mult_gpu.m'); exit;"
```

Submit the above script using `sbatch batch_script.sh`. It will run the specified Matlab script for at most 30 minutes on a gpu partition, having access to 4 cores and 4 GB per core. Reported errors and output will go to the specified text files.

Using machine learning for classification

One of the main uses of machine learning is for clustering. Given, some training data, we would like to classify new instances.

Classic example: Iris flower data set



Clustering with R

```
> data(iris)
>
>
> names(iris)
[1] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
```

```
> iris
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1         5.1         3.5          1.4          0.2  setosa
2         4.9         3.0          1.4          0.2  setosa
3         4.7         3.2          1.3          0.2  setosa
4         4.6         3.1          1.5          0.2  setosa
5         5.0         3.6          1.4          0.2  setosa
6         5.4         3.9          1.7          0.4  setosa
7         4.6         3.4          1.4          0.3  setosa
8         5.0         3.4          1.5          0.2  setosa
9         4.4         2.9          1.4          0.2  setosa
10        4.9         3.1          1.5          0.1  setosa
11        5.4         3.7          1.5          0.2  setosa
12        4.8         3.4          1.6          0.2  setosa
```

```
> n <- nrow(iris);
> train <- sort(sample(1:n, floor(n/2)))
>
> train
[1] 5 12 15 16 17 19 21 22 23 24 25 27 28 29 30 36 38 41 42
[20] 43 46 47 49 50 53 55 57 58 59 61 62 63 65 67 68 70 71 82
[39] 83 84 86 87 88 91 92 94 97 98 99 103 104 105 110 111 114 115 117
[58] 118 119 120 121 122 123 124 126 128 130 132 133 135 136 140 141 142 146
>
> n
[1] 150
>
> length(train)
[1] 75
>
> iris.train <- iris[train,]
> iris.test <- iris[-train,]
```

Using KNN

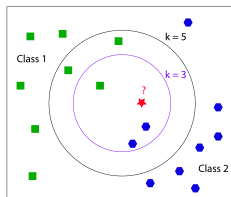
The K nearest neighbor algorithm uses the minimum distance between the test instance (the one we would like to classify) to the training samples to determine the K nearest neighbors. After the K nearest neighbors are determined, one takes the simple majority of these K nearest neighbors to be the prediction of the test instance.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$

Manhattan $\sum_{i=1}^k |x_i - y_i|$

Minkowski $\left(\sum_{i=1}^k (|x_i - y_i|^q) \right)^{1/q}$



```
1 test_pred <- knn(train = iris.train,
2   test = iris.test, cl = train_labels, k = 2);
```

```
> source('test_iris1.R')
[1] 75
[1] "results:\n"
[1] "classification accuracy:"
[1] 0.9733333
>
```

Disease detection

Important to scale features to a similar range. Goodness of high accuracy is problem dependent. Look at confusion matrix.

```
> names(wbcd)
 [1] "diagnosis"      "radius_mean"    "texture_mean"
 [4] "perimeter_mean" "area_mean"      "smoothness_mean"
 [7] "compactness_mean" "concavity_mean" "points_mean"
[10] "symmetry_mean"  "dimension_mean" "radius_se"
[13] "texture_se"     "perimeter_se"  "area_se"
[16] "smoothness_se"  "compactness_se" "concavity_se"
[19] "points_se"      "symmetry_se"   "dimension_se"
[22] "radius_worst"   "texture_worst"  "perimeter_worst"
[25] "area_worst"     "smoothness_worst" "compactness_worst"
[28] "concavity_worst" "points_worst"   "symmetry_worst"
[31] "dimension_worst"
```

```
$ perimeter_worst : num  87 78.3 79.9 76.5 104.5 ...
$ area_worst      : num  549 425 471 434 819 ...
$ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
$ compactness_worst : num  0.127 0.252 0.148 0.182 0.174 ...
$ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
$ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
$ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
$ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
[1] "confusion matrix:"
      obs
pred  0  1
      0 61  2
      1  0 37
```

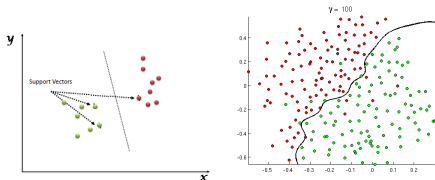
Analyzing the confusion matrix at different k

To see false positives and most importantly false negatives.

k value	False negatives	False positives	Percent classified incorrectly
1	1	3	4 percent
5	2	0	2 percent
11	3	0	3 percent
15	3	0	3 percent
21	2	0	2 percent
27	4	0	4 percent

Gender classification with support vector machines

If each point is plotted in n -dimensional space (where n is the number of features), then classification can be performed by finding a hyper-plane that differentiates two classes with respect to some cost function.



```
In [29]: run run_in_python.py
Python version: 2.7.6 (default, Jan 14 2014, 09:37:27)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)]
pandas version: 0.18.1
NumPy version: 1.12.1
SciPy version: 0.19.0
IPython version: 5.3.0
scikit-learn version: 0.18.1
loading data..

classification accuracy: 88.62

confusion matrix:
(P)      0  1
(A) 0| 461  38
(A) 1|  81 466
```

Environmental sound data set

Ten classes of sound files (we will start with the first 5).

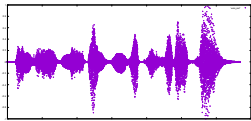
ESC-10: Dataset for Environmental Sound Classification

11 commits 1 branch 0 releases 1 contributor

Branch: **master** [New pull request](#) [Find file](#) [Clone or download](#)

karoldvl Update README.md Latest commit [td4c6f1](#) on Oct 8, 2015

001 - Dog bark	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
002 - Rain	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
003 - Sea waves	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
004 - Baby cry	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
005 - Clock tick	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
006 - Person sneeze	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
007 - Helicopter	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
008 - Chainsaw	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
009 - Rooster	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago
010 - Fire cracking	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp...	2 years ago

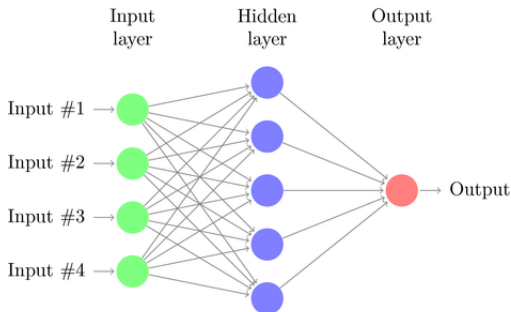


Extract features; normalize columns; split data:

`extract_features.py` `normalize_data.py` `process_test_train_split.pl`

Multi Layer Perceptron

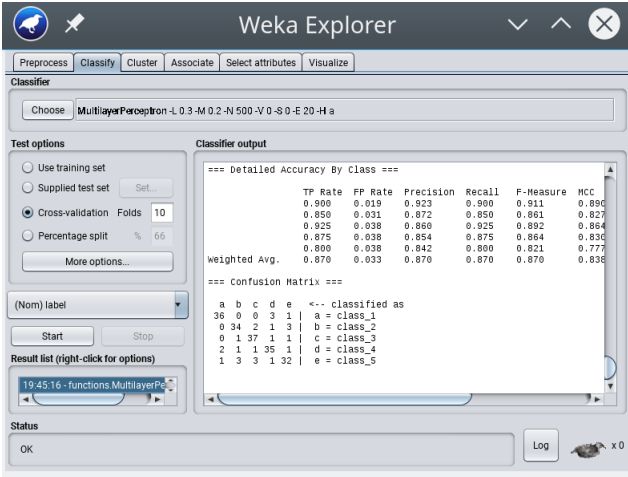
Each node in the Hidden layer is a function of the nodes in the previous layer, and the Output node is a function of the nodes in the Hidden layer.



We can test the algorithm with Java Weka engine (using either code or GUI). For this, we must convert our csv files to arff format.

Using Java Weka

- 1 `module load java`
- 2 `java -jar weka-3-8-1/weka.jar &`



The screenshot shows the Weka Explorer application window. The title bar reads "Weka Explorer". Below the title bar are tabs for "Preprocess", "Classify", "Cluster", "Associate", "Select attributes", and "Visualize". The "Classify" tab is active.

Classifier
Choose: **MultilayerPerceptron-L 0.3-M 0.2-N 500-V 0-S 0-E 20-H a**

Test options

- Use training set
- Supplied test set (Set...)
- Cross-validation Folds: **10**
- Percentage split %: **66**


More options...

(Nom) label: **a**

Start Stop

Result list (right-click for options)

19:45:16 - functions.MultilayerPe...

Status
OK Log  x0

Classifier output

```
=== Detailed Accuracy By Class ===
                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC
0.900  0.019  0.923  0.900  0.911  0.890
0.850  0.031  0.872  0.850  0.861  0.827
0.925  0.038  0.860  0.925  0.892  0.864
0.875  0.038  0.854  0.875  0.864  0.830
0.800  0.038  0.842  0.800  0.821  0.777
weighted Avg.  0.870  0.033  0.870  0.870  0.870  0.838

=== Confusion Matrix ===
  a  b  c  d  e  <-- classified as
36  0  0  3  1 | a = class_1
 0 34  2  1  3 | b = class_2
 0  1 37  1  1 | c = class_3
 2  1  1 35  1 | d = class_4
 1  3  3  1 32 | e = class_5
```

Using Python

Scikit-learn and Tensorflow have a large collection of machine learning algorithms. To install the scikit-learn library use:

- 1 `module load python/2.7.6`
- 2 `pip2 install --user scikit-learn`

Usage from Python:

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(solver='lbfgs', random_state=0, hidden_layer_sizes=[125]);
mlp.fit(X_train, y_train);
y_pred = mlp.predict(X_test)
print("mlp test set score: {:.2f}".format(np.mean(y_pred == y_test)))
```