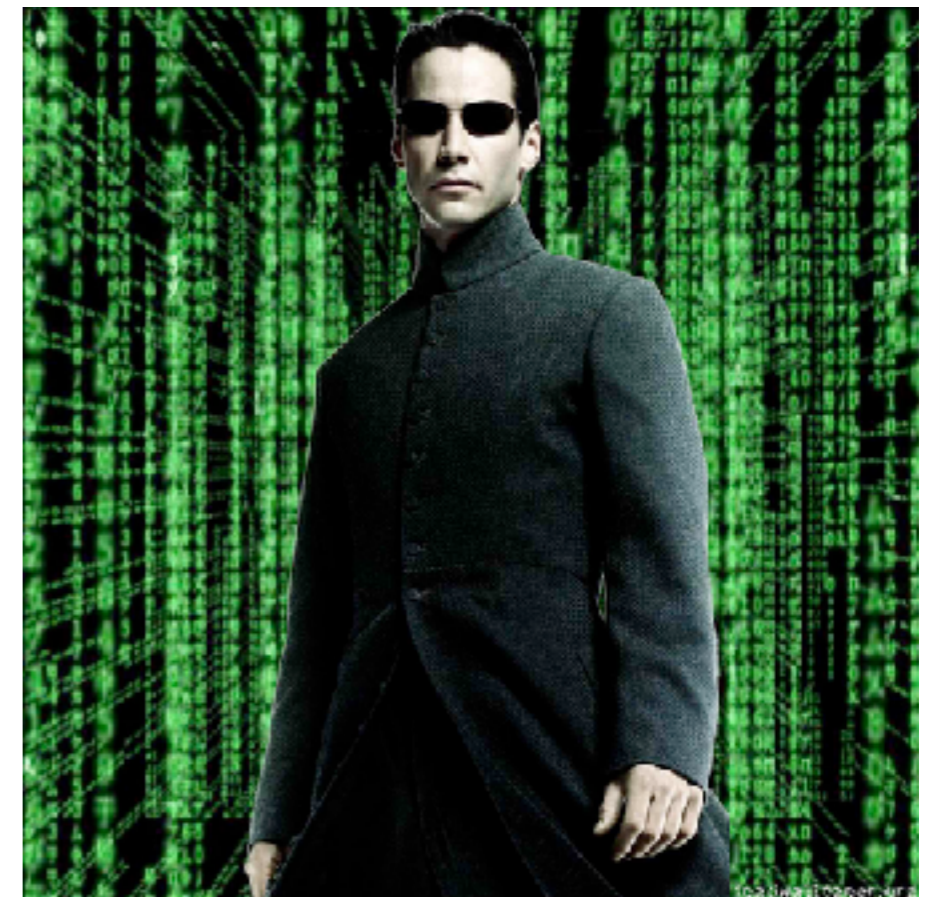


Command Line

Tips & Tricks



- Reduce mouse movement and clicks
- Reduce the number of key strokes
- Get more work done with less effort
Let the computer work for you!
- Prevent mouse-arm and other repetitive strain injuries
- Be a hacker :D



The default terminal

```
1. bash
bash #1 ssh #2
Last login: Sat May  8 12:40:03 on ttys001
Jorans-MacBook-Pro:~ joran$ ls
Applications      Google Drive      Pictures           edirect           repositories
Desktop           Library           Public            miniconda3       sandbox
Documents         Movies            Zotero            perl5             tools
Downloads         Music             db                remotes           warehouse
Jorans-MacBook-Pro:~ joran$ ls repositories/ergo-genome/
README.md
Jorans-MacBook-Pro:~ joran$ ls Documents/Projects/ergobibamus/
00_seq              Snakefile_working_20210320
01_rnaseq           ergo_project_env.yml
02_nanopore         jobgraph.png
Snakefile_test      rulegraph.png
Jorans-MacBook-Pro:~ joran$ cd Documents/Projects/ergobibamus/02_nanopore/
Jorans-MacBook-Pro:02_nanopore joran$ ls -la
total 16
drwxr-xr-x  8 joran  staff   256 Apr 13 21:49 .
drwxr-xr-x 11 joran  staff   352 Apr  5 19:50 ..
-rw-r--r--@ 1 joran  staff  6148 Apr 13 21:49 .DS_Store
drwxr-xr-x  7 joran  staff   224 Apr  8 11:54 20210331_1915_MN33143_AGG378_20c133e5
drwxr-xr-x  4 joran  staff   128 Apr  9 00:29 20210406_1946_MN33143_AGG363_5187cc54
drwxrwxrwx 12 joran  staff   384 Apr 13 22:31 20210412_1817_MN33143_AGG615_76af424f
drwxr-xr-x  5 joran  staff   160 Apr  9 15:50 combined_run1_run2
drwxr-xr-x  8 joran  staff   256 Apr 27 15:16 combined_runs1_to_3
Jorans-MacBook-Pro:02_nanopore joran$
```

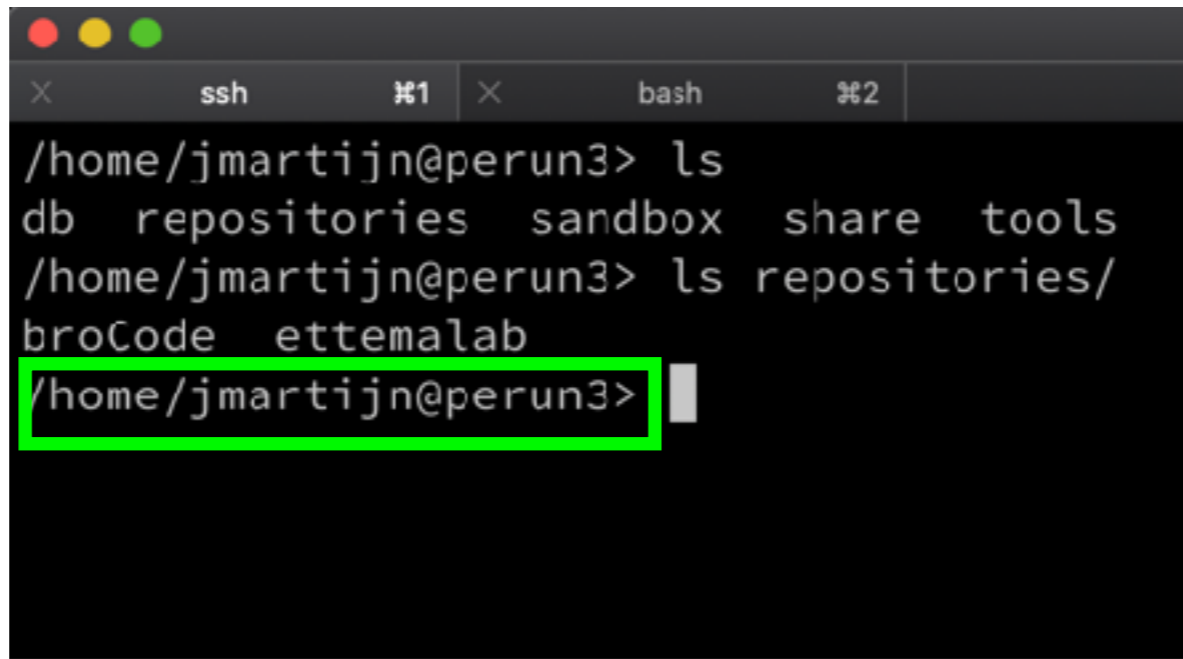
- Very monotone and boring
- What are the files, what are the directories, what are the scripts?
- Difficult to visually find your previous commands
- Which directory are we in?

The .bashrc or .profile

```
1. ssh
bash  %1  ssh  %2
/home/jmartijn@perun2> ls -la
total 150
drwxr-xr-x  19 jmartijn roger   30 May  8 12:54 .
drwxr-xr-x 123 root      root  126 May  8 12:50 ..
-rw-r--r--   1 jmartijn roger  766 May  7 21:24 .alias
-rw-----   1 jmartijn roger 8174 May  8 02:19 .bash_history
-rw-r--r--   1 jmartijn roger  807 May  7 21:24 .bashrc
drwx-----   6 jmartijn roger    7 May  2 12:48 .cache
drwxrwsr-x   3 jmartijn roger    4 Jan 28 22:36 .conda
drwxr-xr-x   5 jmartijn roger    6 Apr  6 11:31 .config
drwxr-xr-x   3 jmartijn roger    3 Jan  9 11:44 db
drwx-----   6 jmartijn roger    9 May  6 16:24 .emacs.d
-rw-r--r--   1 jmartijn roger  788 Apr 23 13:35 .functions
-rw-r--r--   1 jmartijn roger  178 Mar 10 2020 .gitconfig
-rw-r--r--   1 jmartijn roger   91 Apr  7 20:29 .gm_key
drwx-----   3 jmartijn roger    3 May  5 21:47 .gnupg
drwxr-xr-x   3 jmartijn roger    3 Dec 21 14:08 .java
drwxr-xr-x   2 jmartijn roger    3 Sep 25 2019 .keras
-rw-----   1 jmartijn roger  314 Apr 20 14:51 .lessht
drwxr-xr-x   3 jmartijn roger    3 May  5 21:50 .local
drwxr-xr-x   3 jmartijn roger    3 Oct 14 2020 .parallel
drwx-----   3 jmartijn roger    3 Apr  6 11:31 .pki
-rw-----   1 jmartijn roger  100 May  2 14:19 .python_history
drwxr-xr-x   4 jmartijn roger    4 May  7 21:26 repositories
drwxr-xr-x   2 jmartijn roger    3 Nov 23 15:12 sandbox
drwxrwxrwx   3 jmartijn roger    3 Jan  9 11:35 share
drwx-----   2 jmartijn roger    9 May  7 21:45 .ssh
drwxr-xr-x   4 jmartijn roger    4 Oct 24 2020 tools
-rw-----   1 jmartijn roger 8091 Apr 20 23:28 .viminfo
drwxr-xr-x   5 jmartijn roger    8 May  4 15:26 .vscode-server
-rw-r--r--   1 jmartijn roger  183 May  1 17:32 .wget-hsts
-rw-----   1 jmartijn roger 2766 May  8 12:42 .Xauthority
/home/jmartijn@perun2> █
```

- A hidden file
- Located in your home
- Contains the configuration settings for the terminal session
- Loaded each time you start a new session
- You can force load it with `$ source ~/.bashrc` or `$. ~/.bashrc`
- We will edit this to improve our terminal experience!

The prompt



```
ssh #1 bash #2
/home/jmartijn@perun3> ls
db repositories sandbox share tools
/home/jmartijn@perun3> ls repositories/
broCode ettemalab
/home/jmartijn@perun3>
```

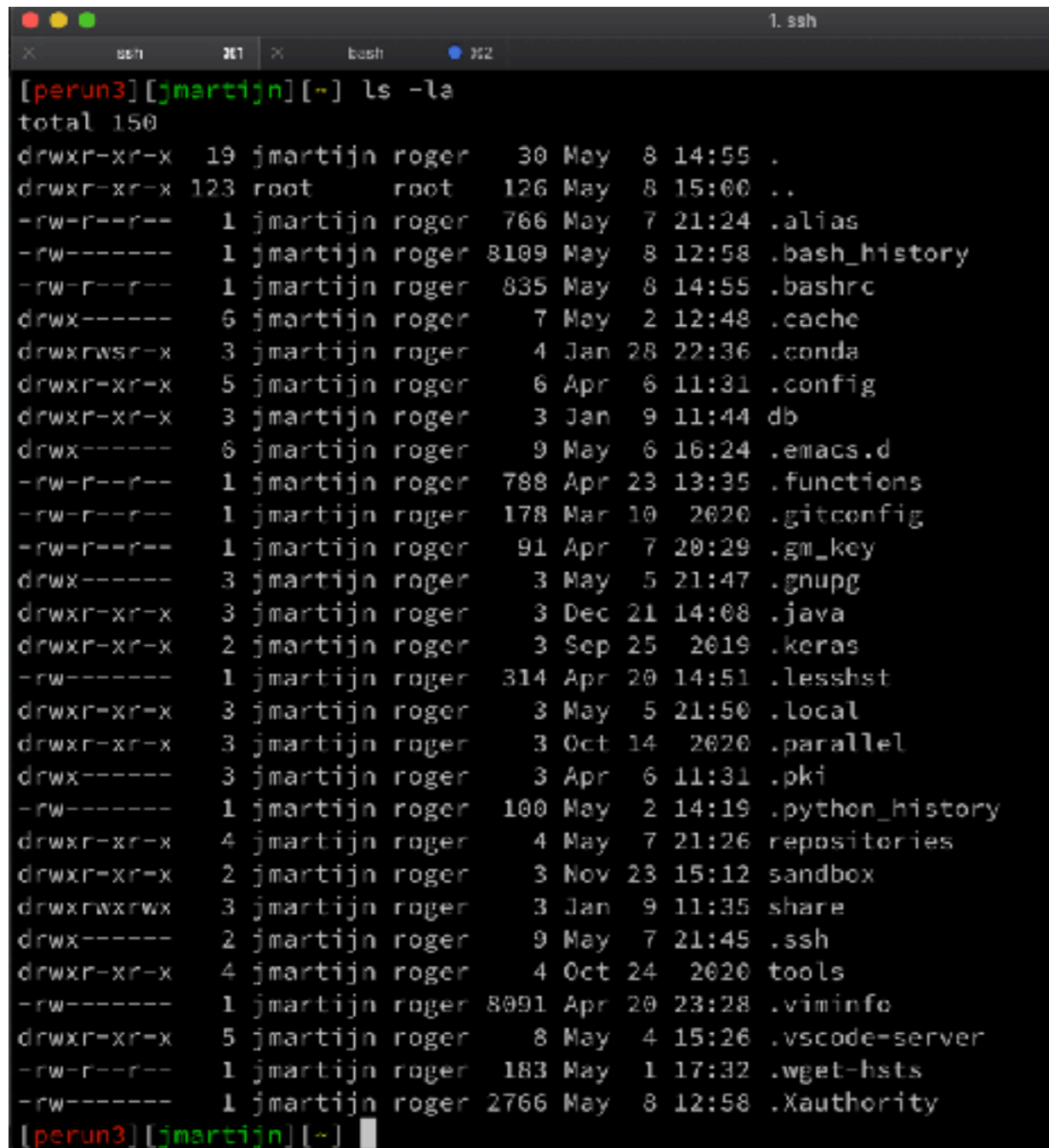
- The prompt typically contains your username, the hostname and the current directory
- The environment variable `$PS1` controls how the prompt looks like
- Add this to your `.bashrc` (it is a single line):

```
#Jimmy Saw's prompt
export PS1="\[\e[00;37m\][\[\e[0m\][\[\e[00;31m\]\h\[\e[0m\]\
[\e[00;37m\]][\[\e[0m\][\[\e[00;32m\]\u\[\e[0m\][\[\e[00;37m\]][\
[\e[0m\][\[\e[00;33m\]\W\[\e[0m\][\[\e[00;37m\]] \[\e[0m\]"
```

- Then reload the `.bashrc` with

```
$ . ~/.bashrc
```

Coloring ls output



```
[perun3][jmartijn][~] ls -la
total 150
drwxr-xr-x  19 jmartijn roger   30 May  8 14:55 .
drwxr-xr-x 123 root      root  126 May  8 15:00 ..
-rw-r--r--   1 jmartijn roger   766 May  7 21:24 .alias
-rw-----   1 jmartijn roger  8109 May  8 12:58 .bash_history
-rw-r--r--   1 jmartijn roger   835 May  8 14:55 .bashrc
drwx-----   6 jmartijn roger    7 May  2 12:48 .cache
drwxrwsr-x   3 jmartijn roger    4 Jan 28 22:36 .conda
drwxr-xr-x   5 jmartijn roger    6 Apr  6 11:31 .config
drwxr-xr-x   3 jmartijn roger    3 Jan  9 11:44 db
drwx-----   6 jmartijn roger    9 May  6 16:24 .emacs.d
-rw-r--r--   1 jmartijn roger   788 Apr 23 13:35 .functions
-rw-r--r--   1 jmartijn roger   178 Mar 10 2020 .gitconfig
-rw-r--r--   1 jmartijn roger    91 Apr  7 20:29 .gm_key
drwx-----   3 jmartijn roger    3 May  5 21:47 .gnupg
drwxr-xr-x   3 jmartijn roger    3 Dec 21 14:08 .java
drwxr-xr-x   2 jmartijn roger    3 Sep 25 2019 .keras
-rw-----   1 jmartijn roger  314 Apr 20 14:51 .lessht
drwxr-xr-x   3 jmartijn roger    3 May  5 21:50 .local
drwxr-xr-x   3 jmartijn roger    3 Oct 14 2020 .parallel
drwx-----   3 jmartijn roger    3 Apr  6 11:31 .pki
-rw-----   1 jmartijn roger   100 May  2 14:19 .python_history
drwxr-xr-x   4 jmartijn roger    4 May  7 21:26 repositories
drwxr-xr-x   2 jmartijn roger    3 Nov 23 15:12 sandbox
drwxrwxrwx   3 jmartijn roger    3 Jan  9 11:35 share
drwx-----   2 jmartijn roger    9 May  7 21:45 .ssh
drwxr-xr-x   4 jmartijn roger    4 Oct 24 2020 tools
-rw-----   1 jmartijn roger  8091 Apr 20 23:28 .viminfo
drwxr-xr-x   5 jmartijn roger    8 May  4 15:26 .vscode-server
-rw-r--r--   1 jmartijn roger   183 May  1 17:32 .wget-hsts
-rw-----   1 jmartijn roger  2766 May  8 12:58 .Xauthority
[perun3][jmartijn][~]
```

- By default files and directories have the same color

- `$ ls -lap --color=auto`

-l detailed list
-a hidden files
-p trailing slash

- It's a pretty long command.
Let's use `alias` to create a shortcut

Add

```
alias l='ls -lap --color=auto'
alias ll='ls -lp --color=auto'
alias ls='ls -p --color=auto'
```

to your `.bashrc`, reload and try
`ls`, `ll`, and `l`

Moving around efficiently

- `$ cd`
Return to your home
- `$ cd -`
Return to the previous directory you were in
- Use `alias` to create cd shortcuts

Add to your `.bashrc`

```
## moving around quickly
alias ..='cd ..'
alias ...='cd ../..'
alias cdseq='cd /scratch4/jmartijn/nanopore_data/'
alias cdrick='cd /scratch3/jmartijn/marinerick/'
alias cdmgiv='cd /scratch3/jmartijn/mgiv/'
alias cdfan='cd /scratch3/jmartijn/lufan_response/'
alias cdergo='cd /scratch3/jmartijn/ergobibamus/'
alias cdexc='cd /scratch3/jmartijn/excavata_monophyly/'
```

Other useful shortcuts

```
## editor shortcuts
```

```
alias e='emacs -nw'
```

```
## common commands
```

```
alias a='source activate'
```

```
alias d='conda deactivate'
```

```
alias editb='emacs -nw ~/.bashrc'
```

```
alias loadb='source ~/.bashrc'
```

Command line hotkeys

Ctrl-a	Jump to START of line
Ctrl-e	Jump to END of line
M-f	Jump FORWARD one word or until special character. For example ' / \ _ . , - '
M-b	Jump BACK one word or until special character
Ctrl-k	Remove (KILL) everything after the cursor
Ctrl-u	Remove everything before the cursor
Ctrl-w	Remove the (previous) word
Ctrl-y	Paste (YANK) the last thing you removed with Ctrl-k, Ctrl-u or Ctrl-w
Ctrl-l	Clear the screen (same as <code>clear</code>)
Ctrl-d	Exit terminal session (same as <code>exit</code>)
M-Bcksp	Remove everything before the cursor until a special character
Ctrl-r	Search for commands in the commands history (also see <code>history</code>)

The Meta key

The Meta key is a special key that is commonly used in command line work. It used to be an actual physical key on keyboards from the 1970s, when computer standard protocols were being developed. Nowadays its more like a virtual key which is activated by different physical keys depending on the system you are using

Meta = Alt on standard keyboards

Meta = Esc on Mac keyboards

If you use iTerm2 on a Mac, you can make the Option key act as the Meta key



Viewing common files

- `$ less <blast_file>`
Meh
- `$ less -S <blast_file>`
Better
- `$ column -t -s $'\t' <blast_file> | less -S`
Even better
- `$ coltab <blast_file>`
Perfect
- Add to your `.bashrc`

```
# easy viewing of tab separated files
function coltab {
    column -t -s $'\t' $1 | less -S
}
```

Viewing alignment files

- Let's say you are logged in to Perun and you just wanted to check if an alignment you just ran looks sensible

You could

- `$ less <alignment_file>`

But if in FASTA format you won't actually see the alignment

- Transfer it to your own PC

```
$ rsync <user>@perun.biochem.dal.ca:<alignment_file> .
```

And then open it with Seaview or Aliview

But it would be better to use `alan`

- `$ alan <alignment_file>`

Installing alan

- Go to your home on Perun
- `$ mkdir tools; cd tools`
- `$ git clone https://github.com/mpdunne/alan.git`
- Add alan to your \$PATH by adding

```
# alan - a command line alignment viewer  
export PATH=~/.tools/alan:$PATH
```

to your `.bashrc`

- -Optionally- replace the variable `default_p` in `~/tools/alan/alan` to make the colors less bright:

```
# the original value of default_p with BRIGHT COLORS  
default_p="KkRr\t01;41\nSsTtQqNn\t01;37;42\nAaIiLlMmFfWwVvCc\t01;44\nCc\t01;105\nEeDd\t01;45\nGg\t01;90;103\nHhYy\t01;46\nPp\t01;90;43"
```

```
# the new value of default_p with TEMPERED COLORS  
default_p="KkRr\t01;31\nSsTtQqNn\t01;32\nAaIiLlMmFfWwVvCc\t01;34\nCc\t01;37\nEeDd\t01;35\nGg\t01;33\nHhYy\t01;36\nPp\t01;33"
```

Other useful custom functions

```
# fish out a particular sequence in a FASTA file
# depends on seqtk
function grabseq {
    fasta=$3
    to_grab=$2
    case "$1" in
        -s) seqtk subseq $fasta <(grep "$to_grab" $fasta | sed -e 's/> //' -e 's/ .* //');;
        -l) seqtk subseq $fasta <(grep -f "$to_grab" $fasta | sed -e 's/> //' -e 's/ .* //');;
        *) echo -e "grabseq -s <seqname> <fasta> to grab a single sequence\ngrab_seq -l
<seqlist> <fasta> to grab a list of sequences"
    esac
}
```

```
# remove a particular sequence from a FASTA file
# depends on seqtk
function rmseq {
    fasta=$3
    to_rmv=$2
    case "$1" in
        -s) seqtk subseq $fasta <( grep ">" $fasta | grep -E -v "$to_rmv" | sed "s/> //" );;
        -l) seqtk subseq $fasta <( grep ">" $fasta | grep -v -f "$to_rmv" | sed "s/> //" );;
        *) echo -e "rmseq -s <seqname> <fasta> to remove a single sequence\nremove_seq -l
<seqlist> <fasta> to remove a list of sequences"
    esac
}
```

Other useful custom functions

```
# format NCBI headers to something readable
```

```
function reformat_ncbi_headers {  
    newfile=${1%.fasta}.hdfmt.fasta  
    cp $1 $newfile  
    sed -i -r -e '/^>/ s/ >.*//' -e 's/>([^\ ]*).*\[(.*)\]/>\2_i_\1/' -e 's/ /_/g' $newfile  
    sed -i -r -e '/^>/ s/gi.*ref\|(.*)\|/\1/' $newfile  
}
```

```
# replace taxanames in trees, fasta, etc
```

```
function replace_names {  
    input=$1  
    mappingfile=$2  
    cp $input $input.nms  
    cat $mappingfile | while read SEARCH REPLACE; do  
        sed -i -r "s/$SEARCH/$REPLACE/" $1.nms  
    done  
}
```

Other useful custom functions

```
# reverse complement function
function revcomp {
    tr "[ATGCatgcNn]" "[TACGtacgNn]" | rev
}

# fasta to phylip
# depends on trimal
function fa2phy {
    trimal -in $1 -out ${1%.*}.phylip -phylip
}

# sum up all numbers in a list
function total {
    tr '\n' '+' $1 | sed "s/\+$/\n/" | bc
}
```

SSH Keys

- To login to a remote server like Perun or Graham, you would typically open your terminal and type

```
$ ssh -XY <user>@<HostName>
```

And then proceed to type in your password
Altogether A LOT of keystrokes

- It is however much much more convenient and safer to switch from a password authentication system to an **SSH Key based authentication system**

ACCOUNT CREATION



PASSWORD AUTHENTICATION



PASSWORD AUTHENTICATION

LET'S SEE ..

ARMISURPLUS = SIGNUM

VITRIOLIX = DIGNUSEST

GIVUSABONUS = SESTERTIUS

ASTERISCLUS = AVECAESAR

GLUTEUSMAXIMUS = OLYMPUS



**PASSWORD
AUTHENTICATION
PASSED**

THESE ROMANS
ARE CRAZY



YOU MAY
ENTER



A HACKER LISTENING IN



AHA!
ACC: ASTERISCLUS
PWD: AVECAESAR

I AM
ASTERISCLUS

PASSWORD IS
'AVE CAESAR'

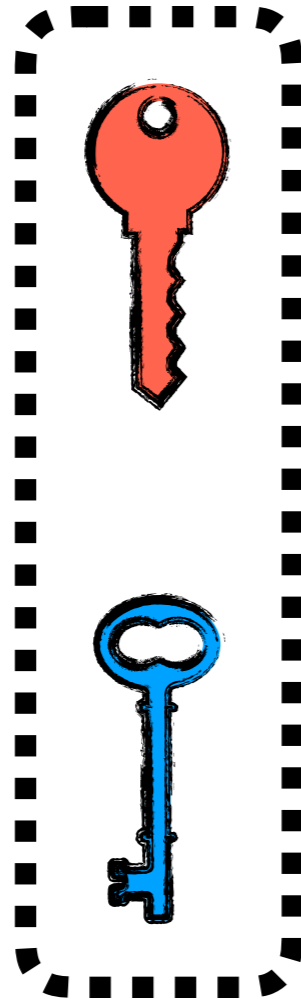
WHO ARE
YOU?

WHAT'S THE
PASSWORD?



SSH KEY AUTHENTICATION

A **public private**
key pair



A **public key** *encrypts* data

A public key can be freely distributed to anyone

A **private key** *decrypts* data

It can only decrypt data that was encrypted with
the associated public key

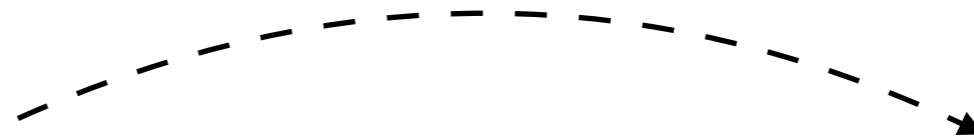
A private key must NEVER be shared with anyone!

SSH KEY KEY PAIR CREATION AND PUBLIC KEY TRANSFER

Create
key pair



Copy **public key**
and transfer



Authenticate
with your
password



SSH KEY AUTHENTICATION

ASTERISCLUS'
PRIVATE KEY



ASTERISCLUS'
PUBLIC KEY



I AM
ASTERISCLUS

AND YES,
OF COURSE!

WHO ARE
YOU?

CAN YOU
DECRYPT
THIS RANDOM
ENCRYPTED
NUMBER?



SSH KEY AUTHENTICATION



AHA!

KEY ID:
ASTERISCLUS

ENCR #:
76103L7X987CJ
2457CBJNDVMC
DFG87KJV82

I AM
ASTERISCLUS

YES,
OF COURSE!

WHO ARE
YOU?

CAN YOU
DECRYPT
THIS RANDOM
ENCRYPTED
NUMBER?



SSH KEY
AUTHENTICATION
PASSED

THESE ROMANS
ARE CRAZY



YOU MAY
ENTER



- **No more password** required
- **Safer** because an external listener will
 - (a) have an EXTREMELY hard time to decrypt the number without the private key
 - (b) the number used is different each time you log in
- **RISK:** if an attacker gains access to your system in any way, they can enter any remote that you have an SSH Key set up for without having to enter a password

You can offset that risk by **setting a passphrase**, but this means you have to type the passphrase each time you login

The SSH Key + passphrase combination is the safest option, by far

1. `$ ssh-keygen -t rsa`
2. Save private key as `~/.ssh/<id>_rsa`
Public key auto saved as
`~/.ssh/<id>_rsa.pub`
3. Optionally choose a passphrase
4. `$ ssh-copy-id <user>@<HostName>`
5. Enter regular password
6. The public key is now stored in
`~/.ssh/authorized_keys` server side

You can now login without typing the password with

```
$ ssh -i ~/.ssh/<id>_rsa <user>@<HostName>
```

SSH Config

- `~/.ssh/config`
- If it doesn't exist yet, you can create it

```
$ touch ~/.ssh/config
```

- A configuration file that remembers for you which ssh parameters you want to run when you want to connect to a particular server
- `$ ssh -i ~/.ssh/<id>_rsa -XY <user>@<HostName>`

equals

```
$ ssh <host>
```

when in your `~/.ssh/config` you have :

- Saves you a TON of keystrokes!!
- Also add to your `~/.bashrc` :

```
Host <host>
HostName <HostName>
User <user>
IdentityFile ~/.ssh/id_rsa
ForwardX11 yes
ForwardX11Trusted yes
```

```
# add tab completion for remote hosts in ssh config file
complete -o default -o nospace -W "$(grep '^Host' ~/.ssh/config | cut -d' ' -f2)" ssh
```