

# Unix, Perl and Python

## Introduction to Unix and LSF

Bingbing Yuan, M.D., Ph.D.

WIBR Bioinformatics and Research Computing

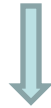
# Question

- I found 100 genes from de novo assembly, I want to quickly find out how many of them are potentially functional.
  - We can blast them against known protein databases.
  - Can we get an answer within one hour?

# Outline

UNIX

1. About files/folders
2. Commonly used UNIX commands
3. Very useful bioinformatics commands



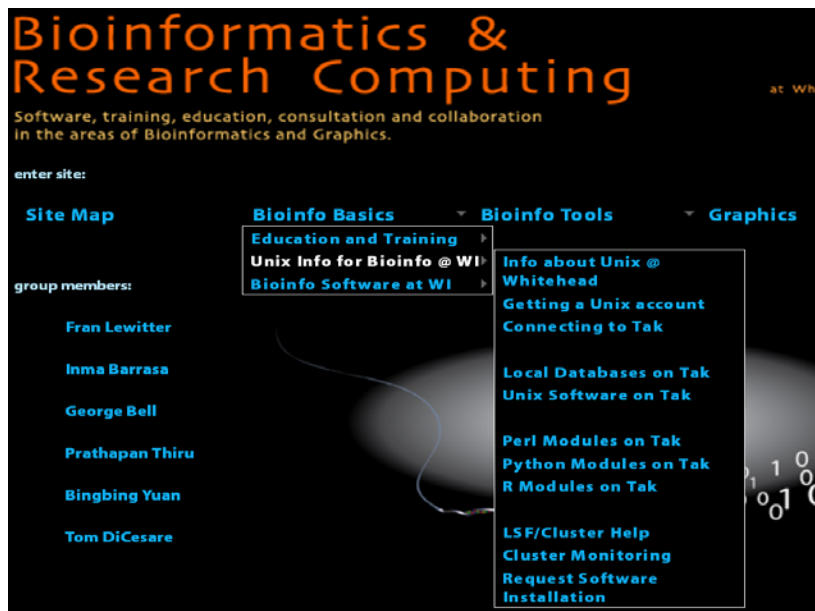
LSF (Load Sharing System)

# Why Unix?

- Many repetitive analyses or tasks can be easily automated
- Some computer programs only run on the Unix operating system.
- TAK (our Unix server): lots of software and databases already installed or downloaded.
- Multiple remote users have access to the Unix at the same time.

# Where can UNIX be used?

- Mac computers
  - Come with Unix
- Windows computers: Install Cygwin
- Dedicated Unix server
  - “tak”, the Whitehead Scientific Linux server



<http://jura.wi.mit.edu/bio>

# What is on tak?

The screenshot shows the Tak website interface. At the top left is the Tak logo and the text "The Whitehead Institute's Scientific Computing Server". A blue box highlights the URL "http://tak.wi.mit.edu/trac/wiki" in the browser's address bar. Below the logo, there are navigation links for "Home", "Timeline", "Roadmap", "View Tickets", "New Ticket", and "Search". A search bar is also present. The main content area is titled "Tak 2.0" and contains two sections: "Software installation requests or issues" with a link to "Create a new ticket", and "Installed software and modules:" with links to "Installed Packages", "Installed Perl Modules", "Installed Python Modules", and "Installed R Modules". A table lists various software packages with columns for Package, Version, Install Date, Description, and Examples and Documents. A blue arrow points from the "Installed software and modules:" section to the "batman" package in the table.

<http://tak.wi.mit.edu/trac/wiki>

logged in as byuan | [Logout](#) | [Packages](#) | [Perl](#) | [Python](#) | [R](#) | [Help/Guide](#)

[Home](#) | [Timeline](#) | [Roadmap](#) | [View Tickets](#) | [New Ticket](#) | [Search](#)

wiki: [WikiStart](#)

## Tak 2.0

**Software installation requests or issues**

- [Create a new ticket](#)

**Installed software and modules:**

- [Installed Packages](#)
- [Installed Perl Modules](#)
- [Installed Python Modules](#)
- [Installed R Modules](#)

Package	Version	Install Date	Description	Examples and Documents
<a href="#">affymetrix</a>	1.12.0	2010-10-12	Affymetrix Power Tools	<a href="#">More info...?</a>
<a href="#">aracne</a>	2006.01.10	2010-11-30	ARACNE (Algorithm for the Reconstruction of Accurate Cellular Networks),	<a href="#">More info...?</a>
<a href="#">bamtools</a>	20101215	2011-01-06	BamTools provides both a programmer's API and an end-user's toolkit for handling BAM files.	<a href="#">More info...?</a>
<a href="#">batman</a>	20090617	2010-08-17	Batman is a new tool for analysing DNA methylation data from MeDIP experiments	<a href="#">More Info...?</a>
<a href="#">bedtools</a>	2.15.0	2012-01-20	The BEDTools utilities allow one to address common genomics tasks such as finding feature overlaps and computing coverage.	<a href="#">More info...?</a>
<a href="#">bioperl</a>	1.6.1	2010-01-28	Perl tools for computational molecular biology	<a href="#">More info...?</a>
<a href="#">bismark</a>	0.6.3	2012-01-24	A tool to map bisulfite converted sequence reads and determine cytosine methylation states	<a href="#">More info...?</a>
<a href="#">blastz</a>	7.0	2010-07-16	blastz alignment program	<a href="#">More info...</a>
<a href="#">blat</a>	0.34	2010-07-16	blat	<a href="#">More info...</a>
<a href="#">bowtie</a>	0.12.7	2011-01-18	Bowtie is an ultrafast, memory-efficient short read aligner.	<a href="#">More info...?</a>
<a href="#">bowtie2</a>	2.0.0.5	2012-01-31	Bowtie 2 is an ultrafast and memory-efficient tool for aligning	<a href="#">More info...?</a>

# Connect to tak with X Window

- Macs:
  1. Access to Terminal: Go => Utilities => Terminal
  2. log in to tak:  
`ssh -Y userName@tak`  
or  
`ssh -X userName@tak`
- Windows:
  1. Launch X Window Server: Xming
  2. Connect to tak with Secure Shell client: PuTTY

# What is in the folder ?

## List all files/directories

**ls** [only show names]

**ls -l** [long listing: show other information too]

```
byuan@tak ~/unix_2012$ ls
```

```
blast_seqs.sh* seq.fa temp/
```

```
byuan@tak ~/unix_2012$ ls -l
```

```
-rwxr--r--  1 byuan barc  1148  2012-03-25 10:05 blast_seqs.sh*
```

```
-rw-r--r--  1 byuan barc 150150 2012-03-25 10:05 seq.fa
```

```
drwxrwsr-x  2 byuan barc  4096  2012-03-25 10:06 results/
```



# Who can read, edit and execute files?

*Error: permission denied*

- Mode: **r**ead, **w**rite, or **e**xecute files?
- Who: **u**ser (**u**), **g**roup (**g**), **o**thers (**o**), everybody (**a**)?

```
-rw-r--r-- byuan barc foo.pl
```

```
chmod u+x foo.pl
```

```
-rwxr--r-- byuan barc foo.pl
```

Allow user to execute script

```
-rw-r--r-- byuan barc document.txt
```

```
chmod g+w document.txt
```

```
-rw-rw-r-- byuan barc document.txt
```

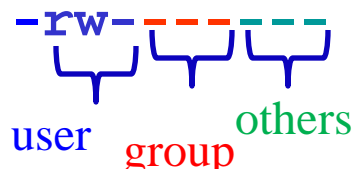
Allow group to edit file

```
-rw-r--r-- byuan barc private.txt
```

```
chmod go-r private.txt
```

```
-rw----- byuan barc private.txt
```

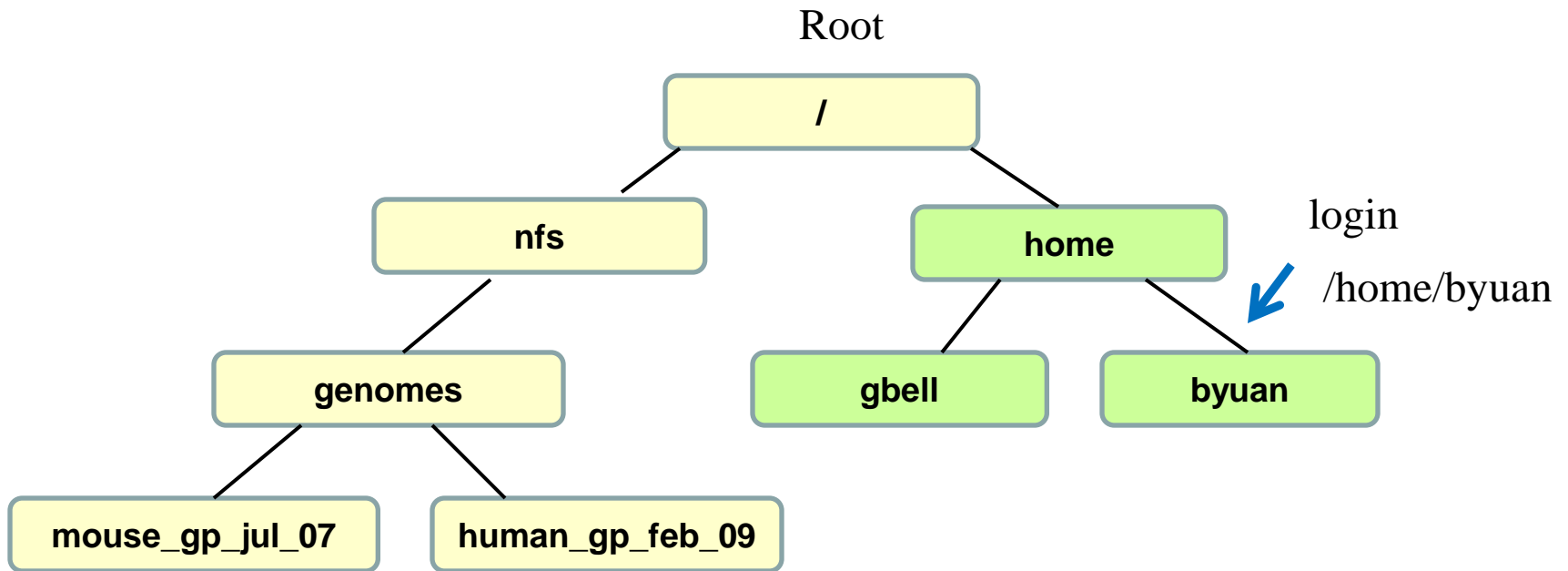
Only user can read/edit file

  
user group others

# Where do you want to go?

*Error: No such file or directory*

- Print the **w**orking **d**irectory: **pwd**
- Change **d**irectories to where you want to go:  
**cd dir**
- Going up the hierarchy: **cd ..**
- Go back home: **cd** or **cd ~**
- Root: /
- Folders:
  - Lab: /nfs/ or /lab/  
e.g. /nfs/BaRC ⇔ WI-FILES1->BaRC
  - /nfs/BaRC\_Public ⇔ WI-FILES1->BaRC\_Public



```
byuan@tak ~$ pwd  
/home/byuan  
byuan@tak ~$ cd /nfs/genomes  
byuan@tak /nfs/genomes$ cd ../../  
byuan@tak /$ cd  
byuan@tak ~$
```

# How to organize files/folders ?

- Make a directory  
`mkdir my_data`
- Remove a directory (after emptying)  
`rmdir my_data`
- Move (rename) a file or directory  
`mv oldFile newFile`
- Copy a file  
`cp oldFile newFileCopy`
- Remove (delete) a file  
`rm oldFile`

Organize computational biology projects:  
[Plos Comp Bio. Jul;5\(7\):e1000424. Epub 2009](#)

# Combining commands

- In a pipeline of commands, the output of one command is used as input for the next
- Link commands with the “pipe” symbol: |

How many fasta files in the folder:

```
ls -l *.fa | wc -l
```

wc -l: count the number of lines

How many items mapped to chr15:

```
grep "chr15" myfile | wc -l
```

grep: print lines matching a pattern

# Save files

- Defaults: stdin = keyboard; stdout = screen
- output examples

`ls > file_name` (make new file)

`ls >> file_name` (append to file)

`ls foo >| file_name` (overwrite)

# Read files

- **Display files on a page-by-page basis**

`more file_name`

↓ or ↑ move line by line      Space: next page      q: quit

- **Display first 2 lines of file:**

`head -2 file_name`

- **Display first 10 lines of file:**

`head file_name`

- **Display last 10 lines of file:**

`tail file_name`

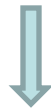
- **Display the last line of file:**

`tail -1 file_name`

# Outline

## UNIX

1. About files/folders ?
2. **Commonly used UNIX commands**
3. Very useful bioinformatics commands



LSF (Load Sharing System)



# Concatenate files

## cat

- Concatenate files

```
cat file1 file2 > bigFile
```

- Show file content at once

```
cat file
```

```
A      it
B      his
D      her
```

- Show hidden characters with `-A` option

```
cat -A file
```

```
A^Iit$
B^Ihis$
D^Iher$
```

```
cat -A file
```

```
A^Iit^M$
B^Ihis^M$
D^Iher^M$
```

<code>^I</code>	TAB ( <code>\t</code> )
<code>\$</code>	end of line ( <code>\$</code> )
<code>^M</code>	carriage return( <code>\r</code> )

From Excel 

# Print lines matching a pattern

## grep

```
byuan@tak$ cat FILE
chr19.fa 4126539 R
chr6.fa 81889764 R
Chr6.fa 77172493 R
byuan@tak$ grep -v 'chr19' FILE
chr6.fa 81889764 R
Chr6.fa 77172493 R
```

```
byuan@tak$ grep 'chr6' FILE
chr6.fa 81889764 R
byuan@tak$ grep -i 'chr6' FILE
chr6.fa 81889764 R
Chr6.fa 77172493 R
byuan@tak$ grep -n -i 'chr6' FILE
2:chr6.fa 81889764 R
3:Chr6.fa 77172493 R
```

<b>-v</b>	<b>Select non-matching lines</b>
<b>-i</b>	<b><u>I</u>gnore case</b>
<b>-n</b>	<b>Print line <u>n</u>umber</b>

# Sort lines of text files:

## sort

**cat FILE**

```
chr6 34314346 F
chr6 52151626 R
chr6 81889764 R
chr6 52151626 R
```

**sort FILE**

```
chr6 34314346 F
chr6 52151626 R
chr6 52151626 R
chr6 81889764 R
```

**sort -u FILE**

```
chr6 34314346 F
chr6 52151626 R
chr6 81889764 R
```

**cat geneFile**

```
geneA chr6 34314346 F
geneB chr8 52151626 R
geneC chr6 11889764 R
```

# sort by chromosome and by genomic location

**sort -k 2,2 -k 3,3n geneFile**

```
geneC chr6 11889764 R
geneA chr6 34314346 F
geneB chr8 52151626 R
```

-n	numerical sort
-r	reverse the result of comparisons
-k pos1,pos2	Start a key at pos1, end it at pos2
-u	unique

# cut sections from each line of files

## cut

```
cat sample.gtf
```

```
chr16 mm9_refGene exon 8513522 8621658 0.000000 + . gene_id "Abat"; transcript_id "NM_172961"  
chr16 mm9_refGene exon 8513522 8621658 0.000000 + . gene_id "Abat"; transcript_id "NM_001170978"  
chr1 mm9_refGene exon 134212715 134230065 0.000000 + . gene_id "Nuak2"; transcript_id "NM_028778"
```

```
# show hidden characters
```

```
cat -A sample.gtf
```

```
chr16^Imm9_refGene^Iexon^I8513522^I8621658^I0.000000^I+^I.^Igene_id "Abat"; transcript_id "NM_172961"$  
chr16^Imm9_refGene^Iexon^I8513522^I8621658^I0.000000^I+^I.^Igene_id "Abat"; transcript_id "NM_001170978"$  
chr1^Imm9_refGene^Iexon^I134212715^I134230065^I0.000000^I+^I.^Igene_id "Nuak2"; transcript_id "NM_028778"$
```

```
# last field separated by tab
```

```
cut -f9 sample.gtf
```

```
gene_id "Abat"; transcript_id "NM_001170978"  
gene_id "Abat"; transcript_id "NM_172961"  
gene_id "Nuak2"; transcript_id "NM_028778"
```

```
# gene names:
```

```
cut -d " " -f2 sample.gtf
```

```
"Abat";  
"Abat";  
"Nuak2";
```

```
# unique gene names
```

```
cut -d " " -f2 sample.gtf | sort -u
```

```
"Abat";  
"Nuak2";
```

-f	output only these fields
-d	field delimiter Default: TAB

# report or omit repeated lines

## uniq

```
cat genes.txt
```

```
Abat    NM_172961  
Abat    NM_001170978  
Nuak2   NM_028778
```

```
cut -f1 genes.txt
```

```
Abat  
Abat  
Nuak2
```

```
# How many transcripts each gene has ?
```

```
cut -f1 genes.txt | uniq -c
```

```
2 Abat  
1 Nuak2
```

```
# Which genes have multiple transcripts?
```

```
cut -f1 genes.txt | uniq -d
```

```
Abat
```

```
# Which genes have only one transcript?
```

```
cut -f1 genes.txt | uniq -u
```

```
Nuak2
```

Note: run sort before uniq

# Downloading files from the web

- **Directly save to disk from web:**

```
wget ftp://ftp.ncbi.nih.gov/pub/geo/...GSM537962%2ECE%2Egz
```

- **Decompress files:**

```
gunzip file.gzip
```

```
tar -xvf file.tar
```

```
tar -xzf file.tar.gz
```

```
tar -xzf /lab/solexa_public/xxx/s_6_sequence.txt.tar.gz -O > s_6_sequence
```

-x: extract files from archive.

-f: specifies filename / tarball name.

-v: Verbose (show progress while extracting files).

-z: filter the archive through gzip, use to decompress .gz files.

-O: extract files to standard output

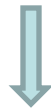
# Notes

- Use up arrow, down arrow to re-use previous commands
- CTRL-C: stop process that are running
- Auto-complete with TAB (filename)
- When reading files/documents:
  - ↓ or ↑ move line by line      space: next page      q: quit
- One-line description of command:      **whatis**  
**whatis mv**
- To get help (**man**ual) command:      **man**  
**man ls**
- Avoid filenames with spaces
  - If necessary to use, refer to with quotes:  
**"My dissertation version 1 .txt"**
- Case sensitive: directories/files, commands

# Outline

## UNIX

1. About files/folders ?
2. Commonly used UNIX commands
3. **Very useful bioinformatics commands**



LSF (Load Sharing System)



# Run blast locally

## blastall

Common Switches:

option	description	default
-p	Program Name	
-d	Database	nr
-i	Query File	stdin
-e	Expectation Value	10
-m	Alignment view options	
-o	output	stdout

Column descriptions for BLAST tabular output file with `-m 8` option

1. query id
2. subject id
3. percent id
4. alignment length
5. number of mismatches
6. number of gap openings
7. query start
8. query end
9. subject start
10. subject end
11. expect value
12. bit score

```
blastall -p blastx -e 1e-10 -m 8 -i transcripts.fa -d hs.faa -o hs_blastx.txt
```

```
TCONS_00035902 gi|154937380|ref|NP_001094348.1| 100.00 16 0 0 480 527 345 360 6e-55 36.2  
TCONS_00039543 gi|256418948|ref|NP_009084.2| 65.00 20 7 0 2726 2785 1244 1263 0.0 31.6
```

More information on blastall

[http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/blastall/blastall\\_node3.html](http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/blastall/blastall_node3.html)

List of local databases:

<http://iona.wi.mit.edu/bio/databases/list.php>

# Check the quality of Illumina reads

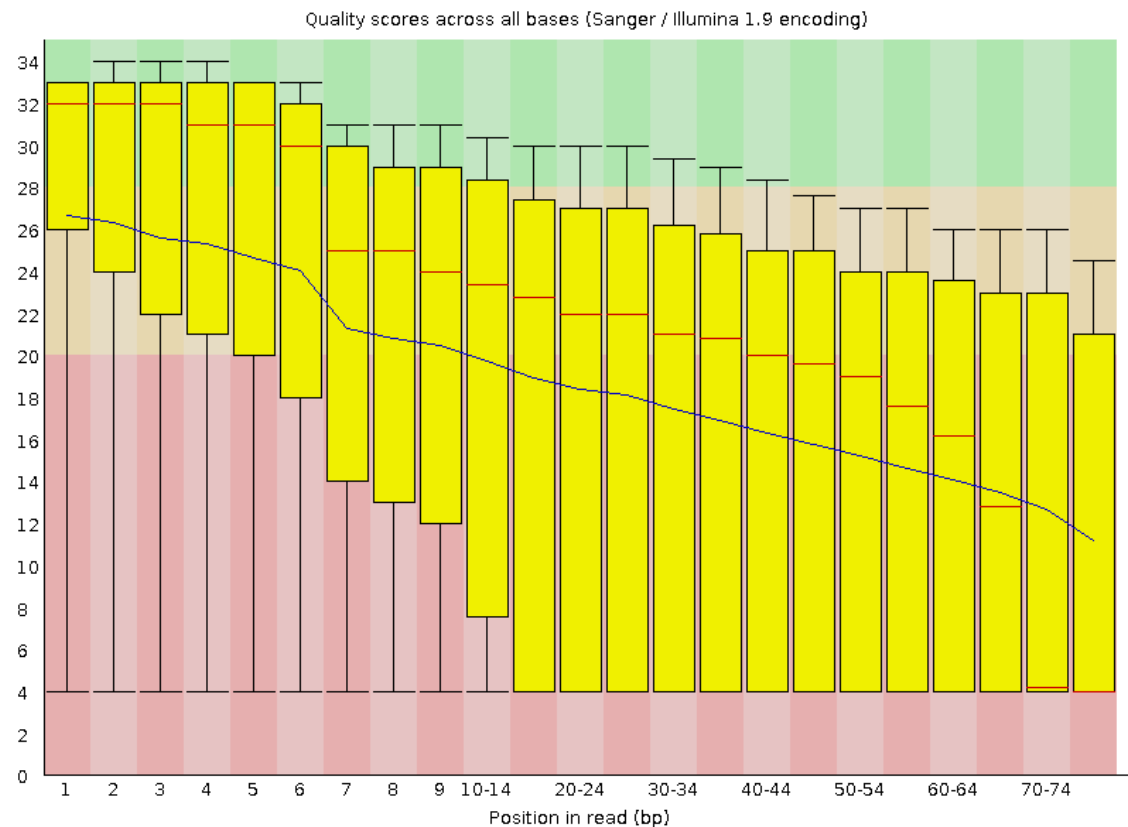
FastQC <http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/>

## FastQC Report

### Summary

- ✓ Basic Statistics
- ✗ Per base sequence quality →
- ✗ Per sequence quality scores
- ! Per base sequence content
- ✓ Per base GC content
- ! Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ! Overrepresented sequences
- ! Kmer Content

fastqc s\_2\_sequence.txt



# Remove reads with low quality

## `fastq_quality_filter`

`fastq_quality_filter -h`

`[-q N]` = Minimum quality score to keep.

`[-p N]` = Minimum percent of bases that must have `[-q]` quality

`[-i INFILE]` = FASTA/Q input file. default is STDIN.

`[-o OUTFILE]` = FASTA/Q output file. default is STDOUT.

`fastq_quality_filter -q 20 -p 75 -i myFile.fq -o myFile.trimmed.fq`

FASTX Toolkit: [http://hannonlab.cshl.edu/fastx\\_toolkit/](http://hannonlab.cshl.edu/fastx_toolkit/)

# Genomic features overlap with each other ?

## intersectBed

### intersectBed

Usage: bedtools intersect [OPTIONS] -a <bed/gff/vcf> -b <bed/gff/vcf>

Options:

- abam The A input file is in BAM format. Output will be BAM as well.
- ubam Write uncompressed BAM output. Default writes compressed BAM.
- s Require same strandedness. That is, only report hits in B that overlap A on the `_same_` strand.
  - By default, overlaps are reported without respect to strand.

**intersectBed -a genes.bed -b peaks.bed > Overlapped.bed**

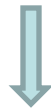


Bedtools: <http://code.google.com/p/bedtools/>

# Outline

## UNIX

1. About files/folders ?
2. Commonly used UNIX commands
3. Very useful bioinformatics commands



**LSF (Load Sharing System)**

# LSF (Load Sharing Facility)

- Jobs takes a long time to finish
- Speed: multiple jobs running at same time



# Hosts

<https://tak.wi.mit.edu/trac/>

## Software installation requests or issues:

- [Create a new ticket](#)

## Installed software and modules:

- [Installed Packages](#)
- [Installed Perl Modules](#)
- [Installed Python Modules](#)
- [Installed R Modules](#)

LSF Cluster Activity		
User Name	Jobs Running	Jobs Pending
byuan	1	0
ibarras	21	0
jwnam	1	1
kmck	1	1
orlando	1	0
sgupta	2	0
<b>Totals:</b>	27	2

Last updated: Tue Jan 17 13:30:01 EST 2012

Host Available    Host Closed    Host Unavailable

Host	Status	Max Jobs	Running	CPU Cores	CPU Speed	Memory	Uptime	Load	System Info
it-c01b04	Available	8	1	8	1.99 GHz	32 GB	179 days	1.02%	HP ProLiant BL460c G1
it-c01b05	Available	8	1	8	2.33 GHz	32 GB	213 days	1.59%	HP ProLiant BL460c G1
it-c01b06	Available	8	2	8	1.86 GHz	32 GB	213 days	2.08%	HP ProLiant BL460c G1
it-c01b07	Available	8	1	8	2.33 GHz	14 GB	66 days	1.20%	HP ProLiant BL460c G1
it-c01b08	Available	8	1	8	2.33 GHz	16 GB	213 days	1.17%	HP ProLiant BL460c G1
it-c01b09	Available	8	2	8	2.33 GHz	16 GB	213 days	2.04%	HP ProLiant BL460c G1
it-c01b10	Available	8	1	8	2.33 GHz	16 GB	213 days	1.70%	HP ProLiant BL460c G1
it-c01b11	Available	8	1	8	2.33 GHz	16 GB	213 days	1.90%	HP ProLiant BL460c G1
it-c01b12	Available	8	1	8	2.33 GHz	16 GB	213 days	1.15%	HP ProLiant BL460c G1
it-c01b13	Available	8	1	8	2.33 GHz	16 GB	213 days	1.06%	HP ProLiant BL460c G1
it-c01b14	Available	8	1	8	2.33 GHz	16 GB	213 days	1.06%	HP ProLiant BL460c G1
it-c01b15	Available	8	1	8	2.33 GHz	16 GB	213 days	1.28%	HP ProLiant BL460c G1
it-c05b02	Available	8	2	16	2.53 GHz	72 GB	209 days	2.25%	HP ProLiant BL460c G6
it-c05b03	Unavailable	1	0	16	2.53 GHz	72 GB	2 days	0.00%	HP ProLiant BL460c G6

# submit jobs: bsub

```
bsub myscript.pl
```

```
bsub "myscript.pl > result"
```

- Send error and standard output to files

```
bsub -e error_file -o std_file myscript.pl
```

- Send job to a host

```
bsub -m it-c01b08 myscript.pl
```

- Send notification to specified email

```
bsub -u foo@gmail.com myscript.pl
```





# Kill jobs: bkill

## **bjobs**

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
103889	byuan	RUN	normal	tak	it-c05b10	*med.fastq	Mar 16 10:35
103890	byuan	RUN	normal	tak	it-c05b12	*leaned.fq	Mar 16 10:37
103891	byuan	RUN	normal	tak	it-c05b12	*eained2.fq	Mar 16 10:37

**bkill 103889**

Job <103889> is being terminated

## **bjobs**

JOBID	USER	STAT	QUEUE	FROM_HOST	EXEC_HOST	JOB_NAME	SUBMIT_TIME
103890	byuan	RUN	normal	tak	it-c05b12	*leaned.fq	Mar 16 10:37
103891	byuan	RUN	normal	tak	it-c05b12	*eained2.fq	Mar 16 10:37

**bkill 0**

Job <103890> is being terminated

Job <103891> is being terminated

## **bjobs**

No unfinished job found

# commands

ls	pwd	chmod	ln -s
cp	mv	rm	mkdir
rmdir	more	head	tail
cat	cut	gunzip	tar
wget	sort	uniq	wc -l
grep	whatis	man	
bsub	bjobs	bkill	
blastall		fastQC	
fastq_quality_filter		intersectBed	

# Further Reading

- BaRC: Getting Started with UNIX
  - [http://iona.wi.mit.edu/bio/education/unix\\_intro.html](http://iona.wi.mit.edu/bio/education/unix_intro.html)
    - tak account, transfer files, X Windows, unix commands, video.
- Whitehead Linux cluster - LSF help
  - [http://iona.wi.mit.edu/bio/bioinfo/docs/LSF\\_help.php](http://iona.wi.mit.edu/bio/bioinfo/docs/LSF_help.php)
    - Popular commands, video, manuals
- UNIX Tutorial for Beginners
  - <http://www.ee.surrey.ac.uk/Teaching/Unix/>