TRANSCRIPTOMICS COMPUTATIONAL PROTOCOL

i3L

Indonesia
International
Institute for
Life Sciences

CREATED BY
Dr. rer. Nat. ARLI ADITYA PARIKESIT
DAVID AGUSTRIAWAN, Ph.D.

COURTESSY OF
DEPARTMENT OF BIOINFORMATICS
INDONESIA INTERNATIONAL INSTITUTE FOR LIFE SCIENCES
MARCH 2021

Table of Contents

Indonesia International Institute for Life Science

Transcriptomics

COURSE CODE: BI021

Blast tutorial in Linux Ubuntu

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: David Agustriawan, Ph.D.

Email: david.agustriawan@i3l.ac.id

Supervisor(s)	Email
Dr.rer.nat Arli Aditya Parikesit	arli.parikesit@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

1. Operate ONLY the computer assigned to you.
    a. If you have any troubleshooting, please contact your supervisor or Building Management
    b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change the system preferences unless directed to do so
    c. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
    d. Do not bring food or drinks into the lab unless it is in your backpack

2. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	1
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

This course session is designed to teach how to be familiar with Linux command and its environment. Moreover, this session also provided the step by step on how to perform Blast in Linux Ubuntu.

The main objective of this learning experience are:

- To be familiar with Linux command and environment
- To understand how to perform Blast in Linux Ubuntu

Material

1. Protocol practicum to perform Blast in Linux Ubuntu

Equipment

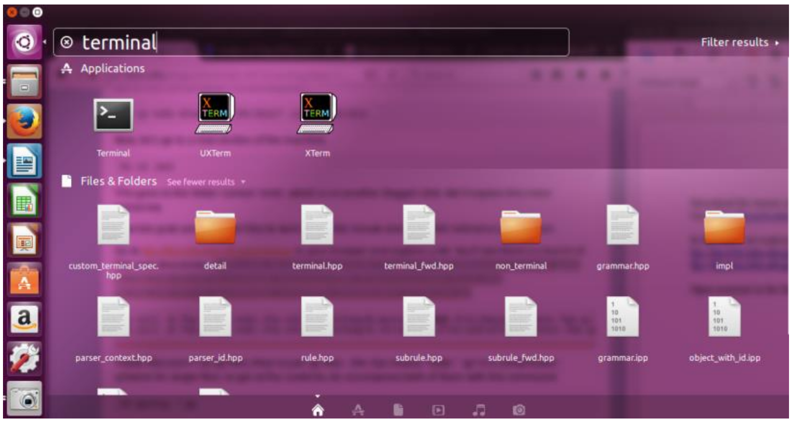
1. Logbook
2. Laptop/PC (available in Bioinformatics laboratory)

Ubuntu Command list

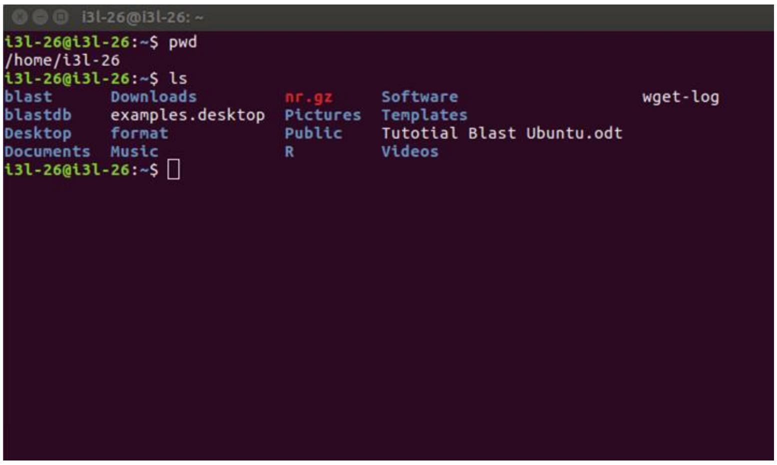
Linux comment	**Function**
mkdir directoryname	Create new directory or folder
touch filename	Create new file
mv oldfilename newfilename	Rename filename
sudo gedit filepath/filename	File edit with gedit
ls	To see the file list from current directory
ls -a	To see the file list with hidden file from current directory
rm –r directoryname	To delete the directory or folder
rm filename	To delete a file
rm *	To delete all the file from current directory

clear	To clear the terminal screen
pwd	To see the current directory full path
cd ~	Go back to home directory
cd	To change the directory
grep	To search for text in a file
cp filepath/filename to filepath/filename	To copy the directory or file

Open terminal in the Linux, click on this icon and in the box search type terminal, then click terminal

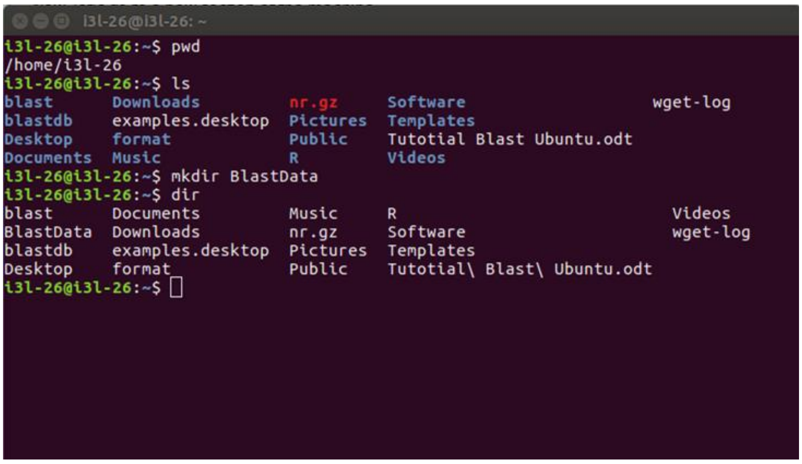


look at your current path (type: *pwd*) and list of directory or folder in your current path (type: *ls*)



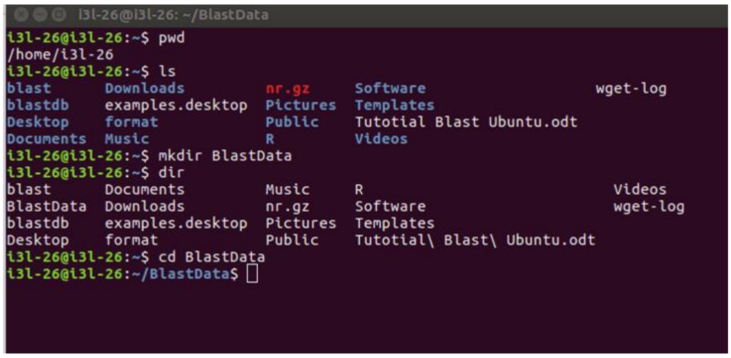
```
i3l-26@i3l-26: ~
i3l-26@i3l-26:~$ pwd
/home/i3l-26
i3l-26@i3l-26:~$ ls
blast      Downloads         nr.gz     Software                       wget-log
blastdb    examples.desktop  Pictures  Templates
Desktop    format            Public    Tutotial Blast Ubuntu.odt
Documents  Music             R         Videos
i3l-26@i3l-26:~$ 
```

then make a directory/folder in your current path (type: *mkdir BlastData*) and check if the folder successfully created (type: *ls* or *dir*)



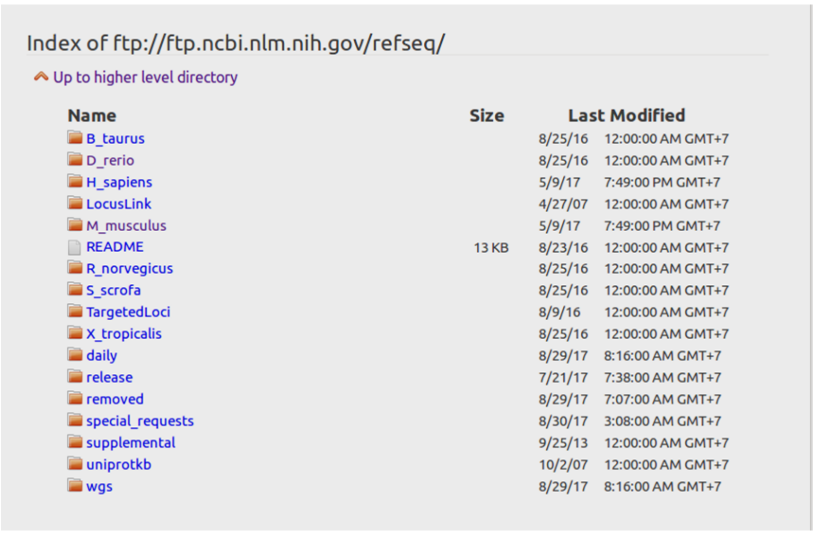
```
i3l-26@i3l-26:~$ pwd
/home/i3l-26
i3l-26@i3l-26:~$ ls
blast      Downloads        nr.gz     Software                      wget-log
blastdb    examples.desktop Pictures  Templates
Desktop    format           Public    Tutotial Blast Ubuntu.odt
Documents  Music            R         Videos
i3l-26@i3l-26:~$ mkdir BlastData
i3l-26@i3l-26:~$ dir
blast      Documents        Music     R                               Videos
BlastData  Downloads        nr.gz     Software                        wget-log
blastdb    examples.desktop Pictures  Templates
Desktop    format           Public    Tutotial\ Blast\ Ubuntu.odt
i3l-26@i3l-26:~$ 
```

change directory to the BlastData (type: *cd BlastData*)



```
i3l-26@i3l-26:~$ pwd
/home/i3l-26
i3l-26@i3l-26:~$ ls
blast      Downloads        nr.gz     Software                      wget-log
blastdb    examples.desktop Pictures  Templates
Desktop    format           Public    Tutotial Blast Ubuntu.odt
Documents  Music            R         Videos
i3l-26@i3l-26:~$ mkdir BlastData
i3l-26@i3l-26:~$ dir
blast      Documents        Music     R                               Videos
BlastData  Downloads        nr.gz     Software                        wget-log
blastdb    examples.desktop Pictures  Templates
Desktop    format           Public    Tutotial\ Blast\ Ubuntu.odt
i3l-26@i3l-26:~$ cd BlastData
i3l-26@i3l-26:~/BlastData$ 
```

Download Mouse and Zebrafish reference proteomes: Go to *ftp://ftp.ncbi.nlm.nih.gov/refseq/* in your browser

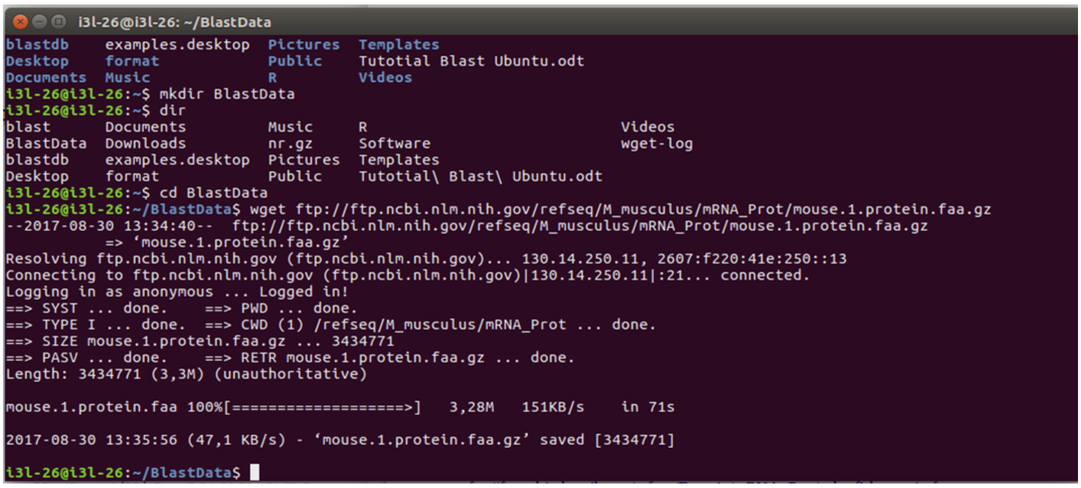

Index of ftp://ftp.ncbi.nlm.nih.gov/refseq/

Name	Size	Last Modified	
B_taurus		8/25/16	12:00:00 AM GMT+7
D_rerio		8/25/16	12:00:00 AM GMT+7
H_sapiens		5/9/17	7:49:00 PM GMT+7
LocusLink		4/27/07	12:00:00 AM GMT+7
M_musculus		5/9/17	7:49:00 PM GMT+7
README	13 KB	8/23/16	12:00:00 AM GMT+7
R_norvegicus		8/25/16	12:00:00 AM GMT+7
S_scrofa		8/25/16	12:00:00 AM GMT+7
TargetedLoci		8/9/16	12:00:00 AM GMT+7
X_tropicalis		8/25/16	12:00:00 AM GMT+7
daily		8/29/17	8:16:00 AM GMT+7
release		7/21/17	7:38:00 AM GMT+7
removed		8/29/17	7:07:00 AM GMT+7
special_requests		8/30/17	3:08:00 AM GMT+7
supplemental		9/25/13	12:00:00 AM GMT+7
uniprotkb		10/2/07	12:00:00 AM GMT+7
wgs		8/29/17	8:16:00 AM GMT+7

In this case, we want to go grab the **mouse and zebrafish protein sets (you can select others).**

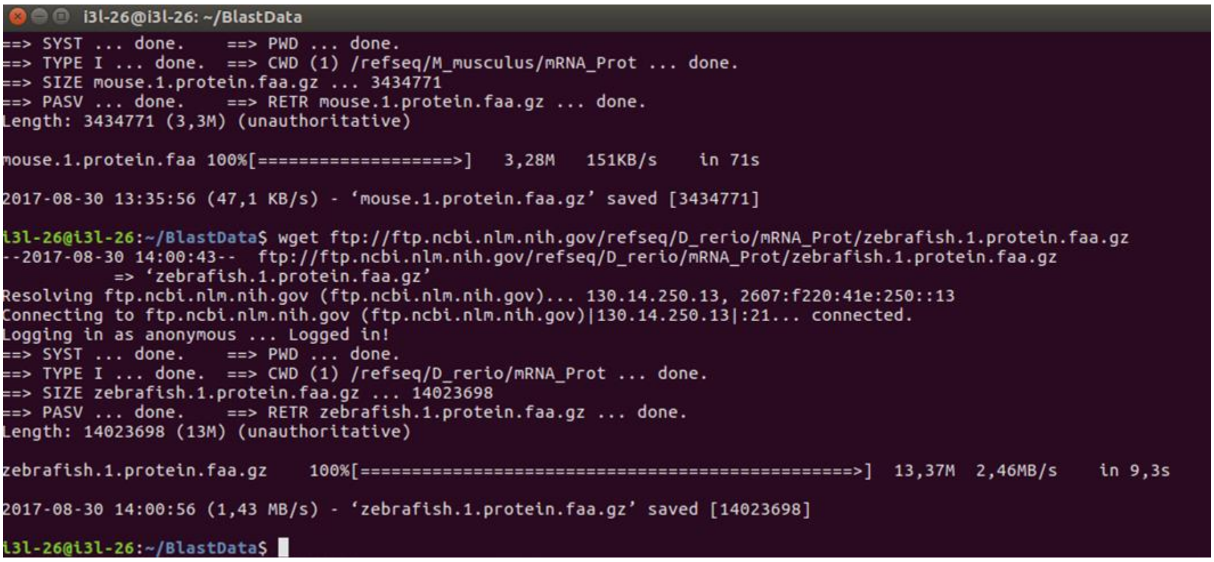
So, grab the mouse protein sets in your current directory “BlastData” in linux terminal
(type: *wget ftp://ftp.ncbi.nlm.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.1.protein.faa.gz*)

As the result we have **mouse.1.protein.faa.gz**.



```
i3l-26@i3l-26: ~/BlastData
blastdb    examples.desktop  Pictures  Templates
Desktop    format            Public    Tutotial Blast Ubuntu.odt
Documents  Music             R         Videos
i3l-26@i3l-26:~$ mkdir BlastData
i3l-26@i3l-26:~$ dir
blast      Documents         Music     R                          Videos
BlastData  Downloads         nr.gz     Software                   wget-log
blastdb    examples.desktop  Pictures  Templates
Desktop    format            Public    Tutotial\ Blast\ Ubuntu.odt
i3l-26@i3l-26:~$ cd BlastData
i3l-26@i3l-26:~/BlastData$ wget ftp://ftp.ncbi.nlm.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.1.protein.faa.gz
--2017-08-30 13:34:40--  ftp://ftp.ncbi.nlm.nih.gov/refseq/M_musculus/mRNA_Prot/mouse.1.protein.faa.gz
           => 'mouse.1.protein.faa.gz'
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.11, 2607:f220:41e:250::13
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.11|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /refseq/M_musculus/mRNA_Prot ... done.
==> SIZE mouse.1.protein.faa.gz ... 3434771
==> PASV ... done.    ==> RETR mouse.1.protein.faa.gz ... done.
Length: 3434771 (3,3M) (unauthoritative)

mouse.1.protein.faa 100%[===================>]   3,28M   151KB/s    in 71s

2017-08-30 13:35:56 (47,1 KB/s) - 'mouse.1.protein.faa.gz' saved [3434771]

i3l-26@i3l-26:~/BlastData$ 
```

and also grab zebrafish protein sets in your current directory "BlastData" in linux terminal (type: *wget ftp://ftp.ncbi.nlm.nih.gov/refseq/D_rerio/mRNA_Prot/zebrafish.1.protein.faa.gz*) as the result we have **zebrafish.1.protein.faa.gz.** Note: you can select others protein sets and download more:

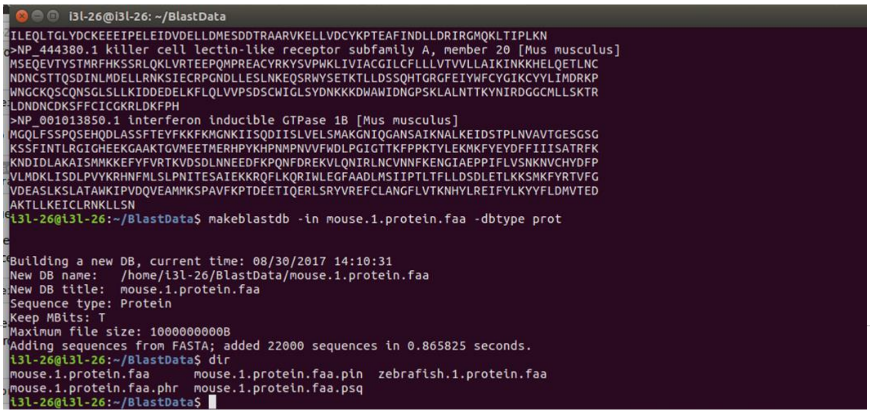


```
i3l-26@i3l-26: ~/BlastData
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /refseq/M_musculus/mRNA_Prot ... done.
==> SIZE mouse.1.protein.faa.gz ... 3434771
==> PASV ... done.    ==> RETR mouse.1.protein.faa.gz ... done.
Length: 3434771 (3,3M) (unauthoritative)

mouse.1.protein.faa 100%[===================>]   3,28M   151KB/s    in 71s

2017-08-30 13:35:56 (47,1 KB/s) - 'mouse.1.protein.faa.gz' saved [3434771]

i3l-26@i3l-26:~/BlastData$ wget ftp://ftp.ncbi.nlm.nih.gov/refseq/D_rerio/mRNA_Prot/zebrafish.1.protein.faa.gz
--2017-08-30 14:00:43--  ftp://ftp.ncbi.nlm.nih.gov/refseq/D_rerio/mRNA_Prot/zebrafish.1.protein.faa.gz
           => 'zebrafish.1.protein.faa.gz'
Resolving ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)... 130.14.250.13, 2607:f220:41e:250::13
Connecting to ftp.ncbi.nlm.nih.gov (ftp.ncbi.nlm.nih.gov)|130.14.250.13|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD (1) /refseq/D_rerio/mRNA_Prot ... done.
==> SIZE zebrafish.1.protein.faa.gz ... 14023698
==> PASV ... done.    ==> RETR zebrafish.1.protein.faa.gz ... done.
Length: 14023698 (13M) (unauthoritative)

zebrafish.1.protein.faa.gz     100%[=============================================>]  13,37M  2,46MB/s    in 9,3s

2017-08-30 14:00:56 (1,43 MB/s) - 'zebrafish.1.protein.faa.gz' saved [14023698]

i3l-26@i3l-26:~/BlastData$ 
```

The .faa means "fasta". 'gz' is a compression scheme for single files; to get at the contents, do uncompress both of them with this command:
(Type: *gunzip *.gz*)

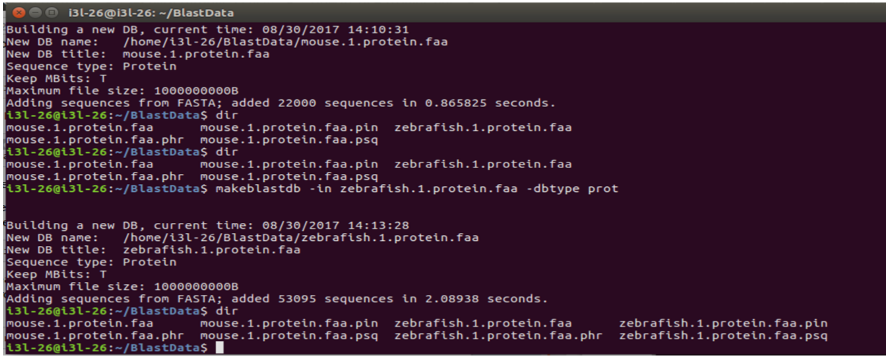
Now, let's convert those protein sets (mouse and zebrafish) into BLAST databases: This lets us use BLAST to query the databases for matches.

(type: *makeblastdb –in mouse.1.protein.faa –dbtype prot*)



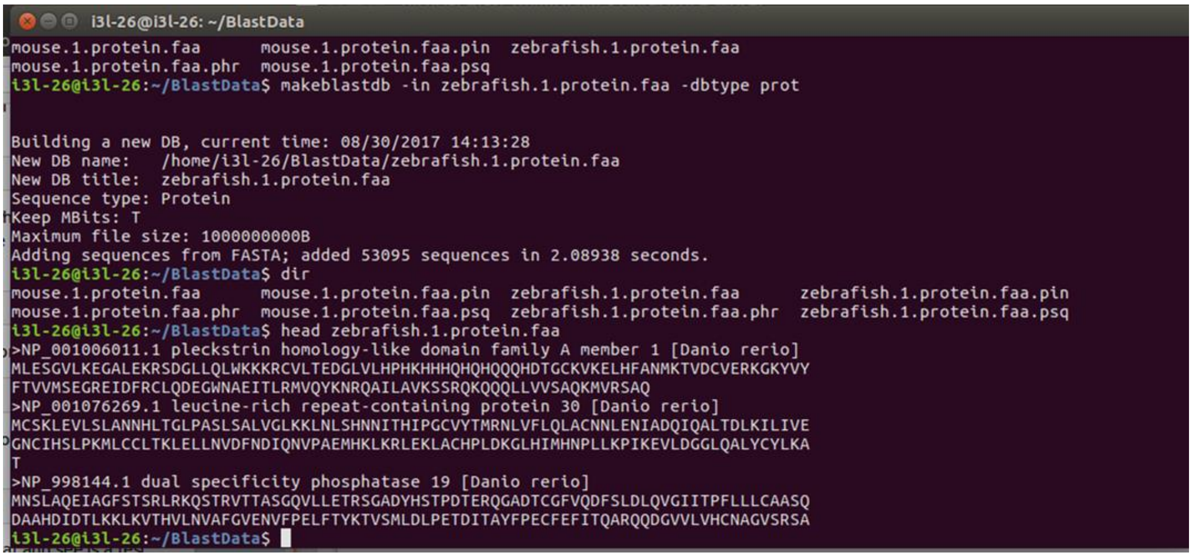
```
i3l-26@i3l-26: ~/BlastData
ILEQLTGLYDCKEEEIPELEIDVDELLDMESDDTRAARVKELLVDCYKPTEAFINDLLDRIRGMQKLTIPLKN
>NP_444380.1 killer cell lectin-like receptor subfamily A, member 20 [Mus musculus]
MSEQEVTYSTMRFHKSSRLQKLVRTEEPQMPREACYRKYSVPWKLIVIACGILCFLLLVTVVLLAIKINKKHELQETLNC
NDNCSTTQSDINLMDELLRNKSIECRPGNDLLESLNKEQSRWYSETKTLLDSSQHTGRGFEIYWFCYGIKCYYLIMDRKP
WNGCKQSCQNSGLSLLKIDDEDELKFLQLVVPSDSCWIGLSYDNKKKDWAWIDNGPSKLALNTTKYNIRDGGCMLLSKTR
LDNDNCDKSFFCICGKRLDKFPH
>NP_001013850.1 interferon inducible GTPase 1B [Mus musculus]
MGQLFSSPQSEHQDLASSFTEYFKKFKMGNKIISQDIISLVELSMAKGNIQGANSAIKNALKEIDSTPLNVAVTGESGSG
KSSFINTLRGIGHEEKGAAKTGVMEETMERHPYKHPNMPNVVFWDLPGIGTTKFPPKTYLEKMKFYEYDFFIIISATRFK
KNDIDLAKAISMMKKEFYFVRTKVDSDLNNEEDFKPQNFDREKVLQNIRLNCVNNFKENGIAEPPIFLVSNKNVCHYDFP
VLMDKLISDLPVYKRHNFMLSLPNITESAIEKKRQFLKQRIWLEGFAADLMSIIPTLTFLLDSDLETLKKSMKFYRTVFG
VDEASLKSLATAWKIPVDQVEAMMKSPAVFKPTDEETIQERLSRYVREFCLANGFLVTKNHYLREIFYLKYYFLDMVTED
AKTLLKEICLRNKLLSN
i3l-26@i3l-26:~/BlastData$ makeblastdb -in mouse.1.protein.faa -dbtype prot


Building a new DB, current time: 08/30/2017 14:10:31
New DB name:   /home/i3l-26/BlastData/mouse.1.protein.faa
New DB title:  mouse.1.protein.faa
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 22000 sequences in 0.865825 seconds.
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ 
```

(type: makeblastdb –in zebrafish.1.protein.faa –dbtype prot)



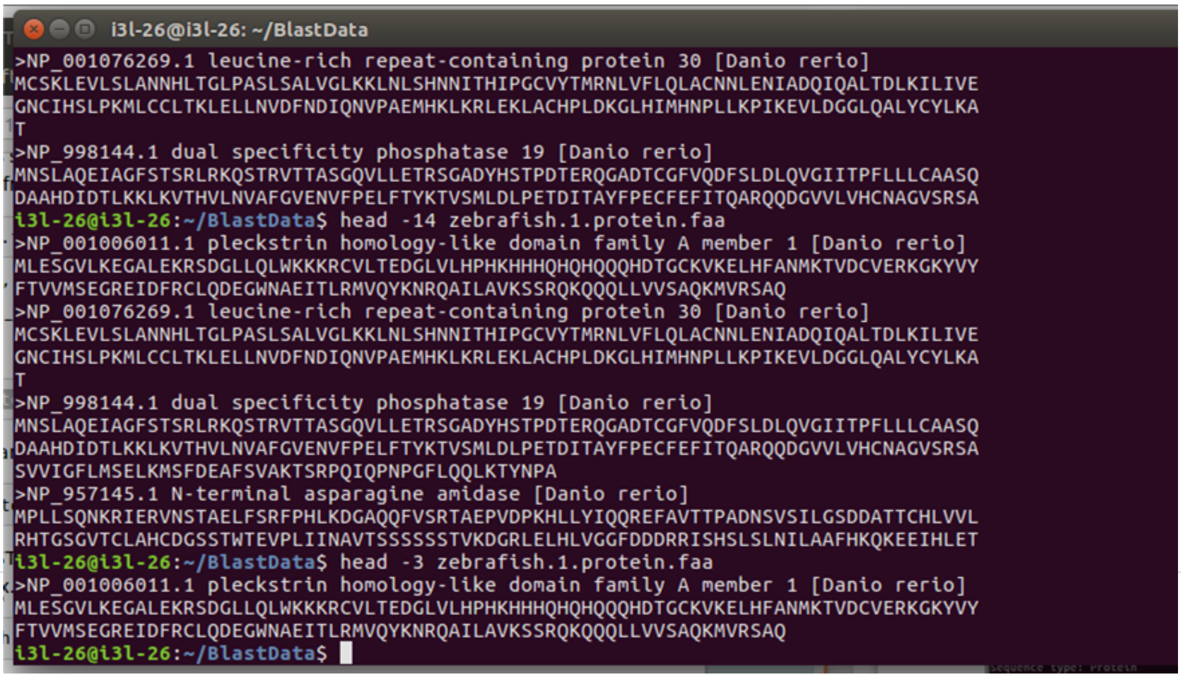
```
Building a new DB, current time: 08/30/2017 14:10:31
New DB name:   /home/i3l-26/BlastData/mouse.1.protein.faa
New DB title:  mouse.1.protein.faa
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 22000 sequences in 0.865825 seconds.
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ makeblastdb -in zebrafish.1.protein.faa -dbtype prot


Building a new DB, current time: 08/30/2017 14:13:28
New DB name:   /home/i3l-26/BlastData/zebrafish.1.protein.faa
New DB title:  zebrafish.1.protein.faa
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 53095 sequences in 2.08938 seconds.
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq  zebrafish.1.protein.faa.phr  zebrafish.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ 
```

You can check the protein sets file: (type: *head zebrafish.1.protein.faa*), There are some fasta files:



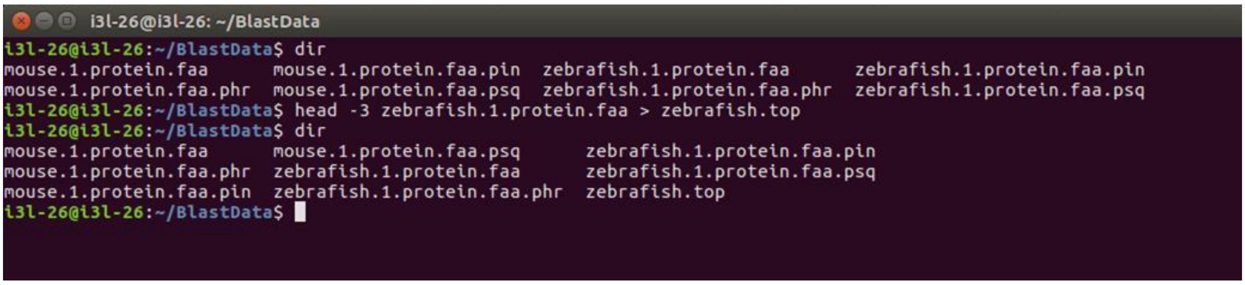
```
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ makeblastdb -in zebrafish.1.protein.faa -dbtype prot


Building a new DB, current time: 08/30/2017 14:13:28
New DB name:   /home/i3l-26/BlastData/zebrafish.1.protein.faa
New DB title:  zebrafish.1.protein.faa
Sequence type: Protein
Keep MBits: T
Maximum file size: 1000000000B
Adding sequences from FASTA; added 53095 sequences in 2.08938 seconds.
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq  zebrafish.1.protein.faa.phr  zebrafish.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ head zebrafish.1.protein.faa
>NP_001006011.1 pleckstrin homology-like domain family A member 1 [Danio rerio]
MLESGVLKEGALEKRSDGLLQLWKKKRCVLTEDGLVLHPHKHHHQHQHQQQHDTGCKVKELHFANMKTVDCVERKGKYVY
FTVVMSEGREIDFRCLQDEGWNAEITLRMVQYKNRQAILAVKSSRQKQQQLLVVSAQKMVRSAQ
>NP_001076269.1 leucine-rich repeat-containing protein 30 [Danio rerio]
MCSKLEVLSLANNHLTGLPASLSALVGLKKLNLSHNNITHIPGCVYTMRNLVFLQLACNNLENIADQIQALTDLKILIVE
GNCIHSLPKMLCCLTKLELLNVDFNDIQNVPAEMHKLKRLEKLACHPLDKGLHIMHNPLLKPIKEVLDGGLQALYCYLKA
T
>NP_998144.1 dual specificity phosphatase 19 [Danio rerio]
MNSLAQEIAGFSTSRLRKQSTRVTTASGQVLLETRSGADYHSTPDTERQGADTCGFVQDFSLDLQVGIITPFLLLCAASQ
DAAHDIDTLKKLKVTHVLNVAFGVENVFPELFTYKTVSMLDLPETDITAYFPECFEFITQARQQDGVVLVHCNAGVSRSA
i3l-26@i3l-26:~/BlastData$ 
```

Then select only a fasta file (type: *head -[number of lines of a fasta file] [name of file]*; so you need to make sure select a complete fasta file. You can try to type: *head -3 zebrafish.1.protein.faa*)



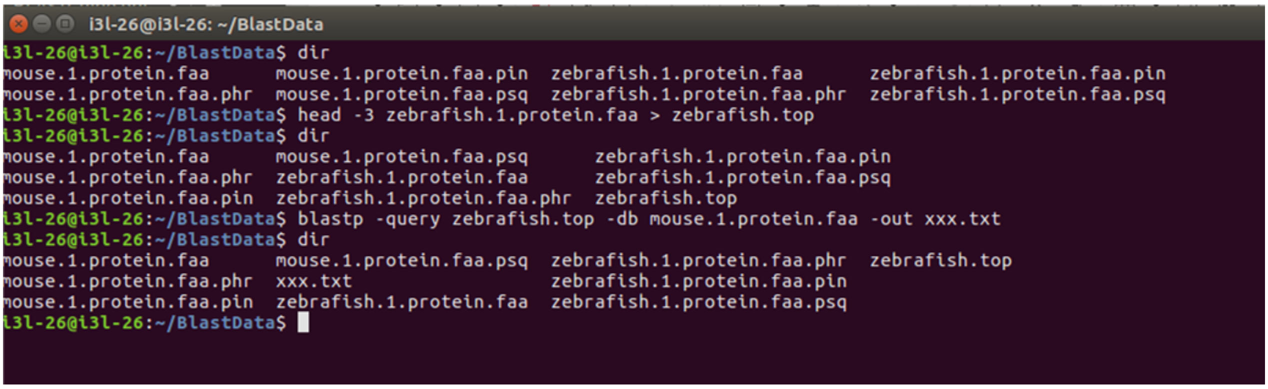
```
>NP_001076269.1 leucine-rich repeat-containing protein 30 [Danio rerio]
MCSKLEVLSLANNHLTGLPASLSALVGLKKLNLSHNNITHIPGCVYTMRNLVFLQLACNNLENIADQIQALTDLKILIVE
GNCIHSLPKMLCCLTKLELLNVDFNDIQNVPAEMHKLKRLEKLACHPLDKGLHIMHNPLLKPIKEVLDGGLQALYCYLKA
T
>NP_998144.1 dual specificity phosphatase 19 [Danio rerio]
MNSLAQEIAGFSTSRLRKQSTRVTTASGQVLLETRSGADYHSTPDTERQGADTCGFVQDFSLDLQVGIITPFLLLCAASQ
DAAHDIDTLKKLKVTHVLNVAFGVENVFPELFTYKTVSMLDLPETDITAYFPECFEFITQARQQDGVVLVHCNAGVSRSA
i3l-26@i3l-26:~/BlastData$ head -14 zebrafish.1.protein.faa
>NP_001006011.1 pleckstrin homology-like domain family A member 1 [Danio rerio]
MLESGVLKEGALEKRSDGLLQLWKKKRCVLTEDGLVLHPHKHHHQHQHQQQHDTGCKVKELHFANMKTVDCVERKGKYVY
FTVVMSEGREIDFRCLQDEGWNAEITLRMVQYKNRQAILAVKSSRQKQQQLLVVSAQKMVRSAQ
>NP_001076269.1 leucine-rich repeat-containing protein 30 [Danio rerio]
MCSKLEVLSLANNHLTGLPASLSALVGLKKLNLSHNNITHIPGCVYTMRNLVFLQLACNNLENIADQIQALTDLKILIVE
GNCIHSLPKMLCCLTKLELLNVDFNDIQNVPAEMHKLKRLEKLACHPLDKGLHIMHNPLLKPIKEVLDGGLQALYCYLKA
T
>NP_998144.1 dual specificity phosphatase 19 [Danio rerio]
MNSLAQEIAGFSTSRLRKQSTRVTTASGQVLLETRSGADYHSTPDTERQGADTCGFVQDFSLDLQVGIITPFLLLCAASQ
DAAHDIDTLKKLKVTHVLNVAFGVENVFPELFTYKTVSMLDLPETDITAYFPECFEFITQARQQDGVVLVHCNAGVSRSA
SVVIGFLMSELKMSFDEAFSVAKTSRPQIQPNPGFLQQLKTYNPA
>NP_957145.1 N-terminal asparagine amidase [Danio rerio]
MPLLSQNKRIERVNSTAELFSRFPHLKDGAQQFVSRTAEPVDPKHLLYIQQREFAVTTPADNSVSILGSDDATTCHLVVL
RHTGSGVTCLAHCDGSSTWTEVPLIINAVTSSSSSSTVKDGRLELHLVGGFDDDRRISHSLSLNILAAFHKQKEEIHLET
i3l-26@i3l-26:~/BlastData$ head -3 zebrafish.1.protein.faa
>NP_001006011.1 pleckstrin homology-like domain family A member 1 [Danio rerio]
MLESGVLKEGALEKRSDGLLQLWKKKRCVLTEDGLVLHPHKHHHQHQHQQQHDTGCKVKELHFANMKTVDCVERKGKYVY
FTVVMSEGREIDFRCLQDEGWNAEITLRMVQYKNRQAILAVKSSRQKQQQLLVVSAQKMVRSAQ
i3l-26@i3l-26:~/BlastData$ 
```

Let’s take the output of ‘head’ and put it in a file, ‘zebrafish.top’, that we can use for other purposes:
(type: head -3 zebrafish.protein.faa > zebrafish.top)



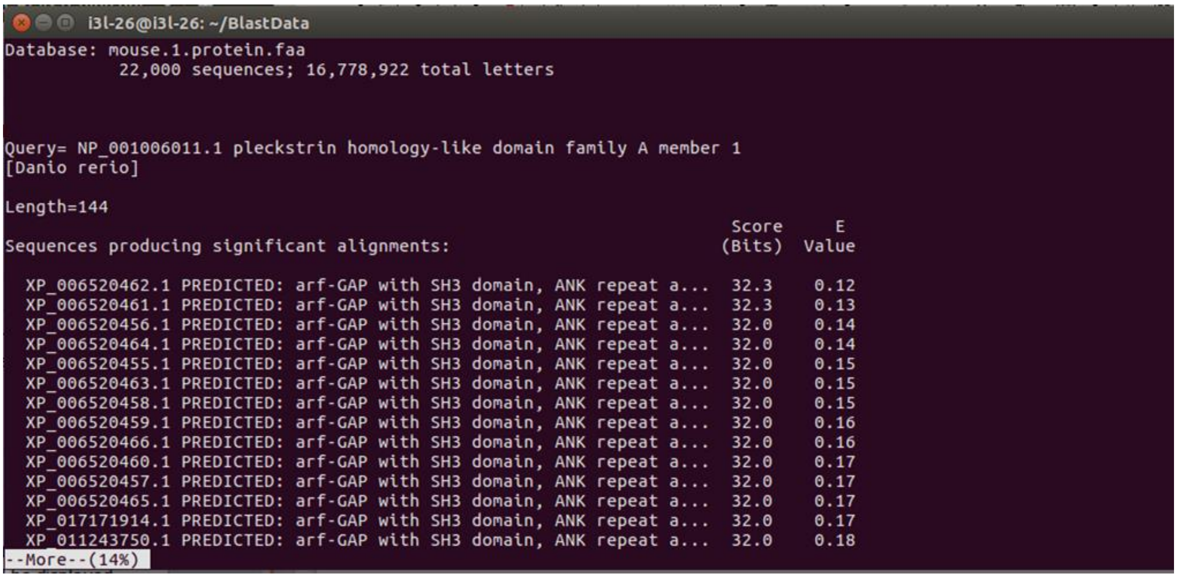
```
i3l-26@i3l-26: ~/BlastData
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq  zebrafish.1.protein.faa.phr  zebrafish.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ head -3 zebrafish.1.protein.faa > zebrafish.top
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.pin  zebrafish.1.protein.faa.phr  zebrafish.top
i3l-26@i3l-26:~/BlastData$ 
```

Now let’s run a BLASTP comparing these zebrafish sequences to the mouse proteins, and we’ll put the results in a file ‘xxx.txt’:
(*type: blastp -query zebrafish.top -db mouse.1.protein.faa -out xxx.txt*)



```
i3l-26@i3l-26: ~/BlastData
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.pin  zebrafish.1.protein.faa      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  mouse.1.protein.faa.psq  zebrafish.1.protein.faa.phr  zebrafish.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ head -3 zebrafish.1.protein.faa > zebrafish.top
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq      zebrafish.1.protein.faa.pin
mouse.1.protein.faa.phr  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.pin  zebrafish.1.protein.faa.phr  zebrafish.top
i3l-26@i3l-26:~/BlastData$ blastp -query zebrafish.top -db mouse.1.protein.faa -out xxx.txt
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq  zebrafish.1.protein.faa.phr  zebrafish.top
mouse.1.protein.faa.phr  xxx.txt                  zebrafish.1.protein.faa.pin
mouse.1.protein.faa.pin  zebrafish.1.protein.faa  zebrafish.1.protein.faa.psq
i3l-26@i3l-26:~/BlastData$ 
```

OK, now take a look at that file with ‘more’ (type: *more xxx.txt*):



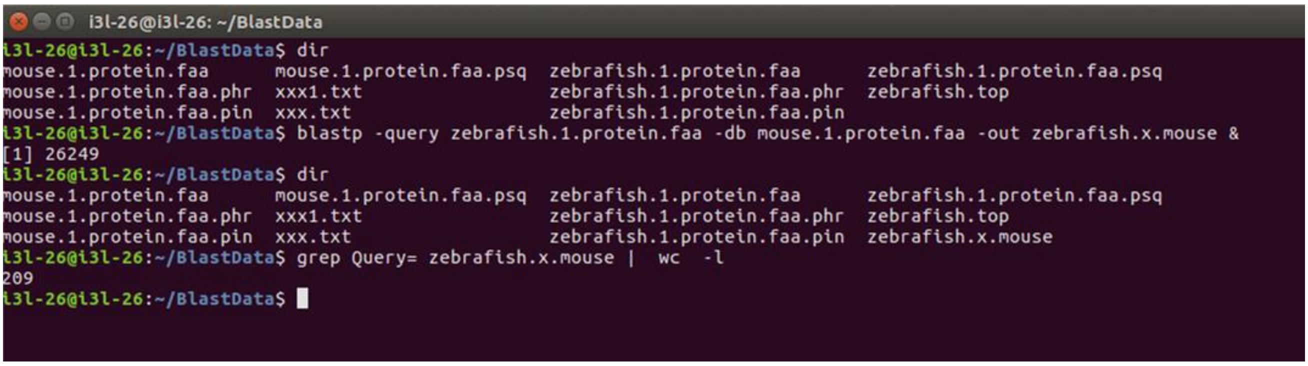
```
i3l-26@i3l-26: ~/BlastData
Database: mouse.1.protein.faa
           22,000 sequences; 16,778,922 total letters



Query= NP_001006011.1 pleckstrin homology-like domain family A member 1
[Danio rerio]

Length=144
                                                                      Score     E
Sequences producing significant alignments:                          (Bits)  Value

  XP_006520462.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.3    0.12
  XP_006520461.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.3    0.13
  XP_006520456.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.14
  XP_006520464.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.14
  XP_006520455.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.15
  XP_006520463.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.15
  XP_006520458.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.15
  XP_006520459.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.16
  XP_006520466.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.16
  XP_006520460.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.17
  XP_006520457.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.17
  XP_006520465.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.17
  XP_017171914.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.17
  XP_011243750.1 PREDICTED: arf-GAP with SH3 domain, ANK repeat a...  32.0    0.18
--More--(14%)
```

You can push enter button to see all the files, and push q to exit. You also can specify the threshold by adding comment for example ‘-evalue 1e-6’
(type: *blastp -query zebrafish.top -db mouse.1.protein.faa -evalue 1e-6 -out xxx.txt*)

Now let's run a bigger BLAST, all zebrafish proteins against all mouse proteins:
(type: *blastp -query zebrafish.1.protein.faa -db mouse.1.protein.faa -out zebrafish.x.mouse &*)

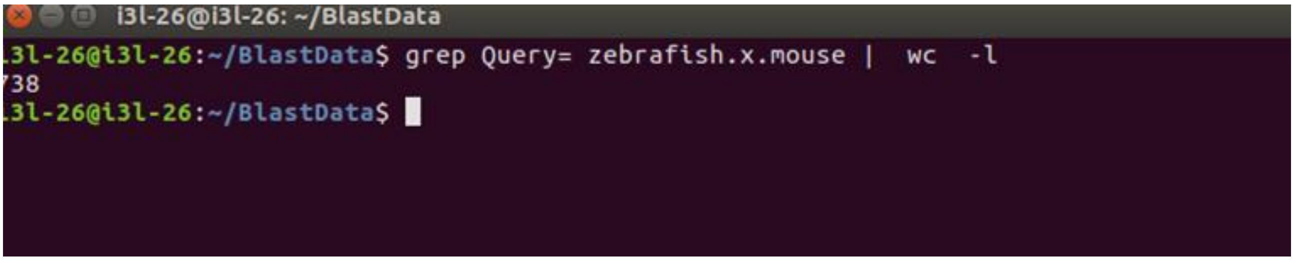
This is going to take a while, which is why we told the computer to give us back a command prompt while blastp runs (that's what the *&* does).

So, how long is it going to take? We can guesstimate by looking at how many sequences have been processed since we started. To do that, run a comment below:
(type: *grep Query= zebrafish.x.mouse | wc -l*)



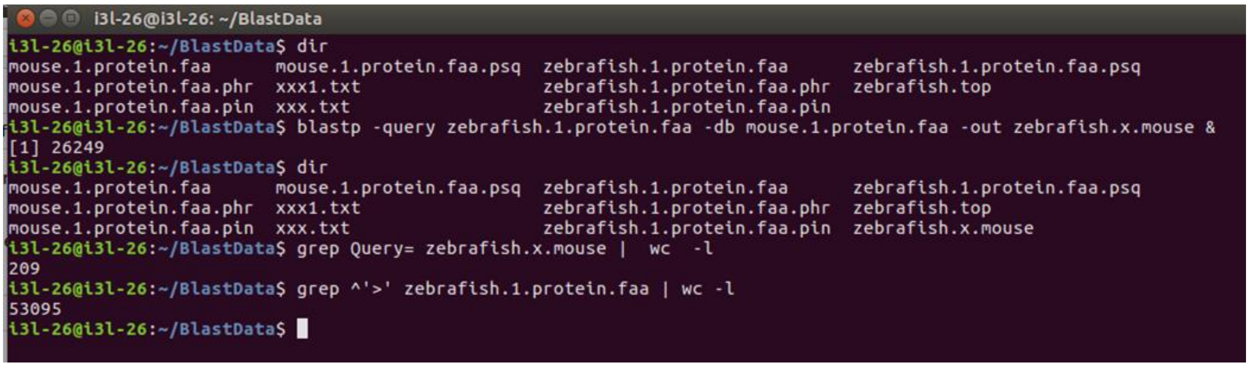
```
i3l-26@i3l-26: ~/BlastData
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.phr  xxx1.txt                 zebrafish.1.protein.faa.phr  zebrafish.top
mouse.1.protein.faa.pin  xxx.txt                  zebrafish.1.protein.faa.pin
i3l-26@i3l-26:~/BlastData$ blastp -query zebrafish.1.protein.faa -db mouse.1.protein.faa -out zebrafish.x.mouse &
[1] 26249
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.phr  xxx1.txt                 zebrafish.1.protein.faa.phr  zebrafish.top
mouse.1.protein.faa.pin  xxx.txt                  zebrafish.1.protein.faa.pin  zebrafish.x.mouse
i3l-26@i3l-26:~/BlastData$ grep Query= zebrafish.x.mouse |  wc  -l
209
i3l-26@i3l-26:~/BlastData$ 
```

here we get 209 sequences have been processed, after some minutes there will be more sequences is processed. After five minutes there are 738 sequences is processed (so with & symbol we don't need to wait the blast process, it will run until the process complete and we can do another comments in the linux terminal). Here, | is what's known as a 'pipe', telling the command line to take the output of 'grep' and send it to the command 'wc', which counts words, lines, and paragraphs. The '-l' tells wc to count the lines only.



```
i3l-26@i3l-26: ~/BlastData
i3l-26@i3l-26:~/BlastData$ grep Query= zebrafish.x.mouse |  wc  -l
738
i3l-26@i3l-26:~/BlastData$ 
```

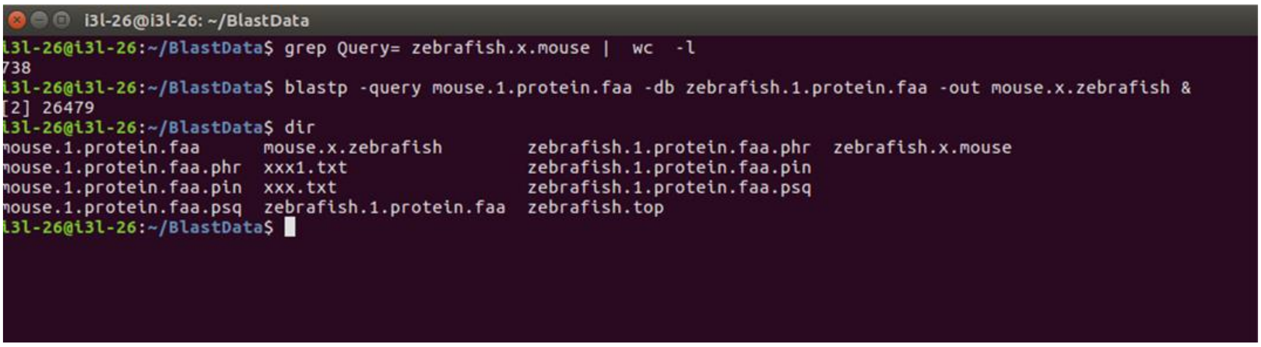
Compare that number to the number of sequences in the zebrafish protein database:
(type: *grep ^'>' zebrafish.1.protein.faa | wc -l*)



```
i3l-26@i3l-26: ~/BlastData
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.phr  xxx1.txt                 zebrafish.1.protein.faa.phr  zebrafish.top
mouse.1.protein.faa.pin  xxx.txt                  zebrafish.1.protein.faa.pin
i3l-26@i3l-26:~/BlastData$ blastp -query zebrafish.1.protein.faa -db mouse.1.protein.faa -out zebrafish.x.mouse &
[1] 26249
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.1.protein.faa.psq  zebrafish.1.protein.faa      zebrafish.1.protein.faa.psq
mouse.1.protein.faa.phr  xxx1.txt                 zebrafish.1.protein.faa.phr  zebrafish.top
mouse.1.protein.faa.pin  xxx.txt                  zebrafish.1.protein.faa.pin  zebrafish.x.mouse
i3l-26@i3l-26:~/BlastData$ grep Query= zebrafish.x.mouse |  wc  -l
209
i3l-26@i3l-26:~/BlastData$ grep ^'>' zebrafish.1.protein.faa | wc -l
53095
i3l-26@i3l-26:~/BlastData$ 
```

Let's start a *second* BLAST, all of mouse against all of zebrafish:

(type: *blastp -query mouse.1.protein.faa -db zebrafish.1.protein.faa -out mouse.x.zebrafish* &)



```
i3l-26@i3l-26:~/BlastData$ grep Query= zebrafish.x.mouse |  wc  -l
738
i3l-26@i3l-26:~/BlastData$ blastp -query mouse.1.protein.faa -db zebrafish.1.protein.faa -out mouse.x.zebrafish &
[2] 26479
i3l-26@i3l-26:~/BlastData$ dir
mouse.1.protein.faa      mouse.x.zebrafish        zebrafish.1.protein.faa.phr  zebrafish.x.mouse
mouse.1.protein.faa.phr  xxx1.txt                 zebrafish.1.protein.faa.pin
mouse.1.protein.faa.pin  xxx.txt                  zebrafish.1.protein.faa.psq
mouse.1.protein.faa.psq  zebrafish.1.protein.faa  zebrafish.top
i3l-26@i3l-26:~/BlastData$ 
```

Indonesia International Institute for Life Science

Transcriptomics

COURSE CODE: BI021

Bowtie

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: David Agustriawan, Ph.D.

Email: david.agustraiwan@i3l.ac.id

Supervisor(s)	Email
Dr.rer.nat Arli Aditya Parikesit	arli.parikesit@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

3. Operate ONLY the computer assigned to you.
    a. If you have any troubleshooting, please contact your supervisor or Building Management
    b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change the system preferences unless directed to do so
    c. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
    d. Do not bring food or drinks into the lab unless it is in your backpack

4. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	2
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

This course session is designed to teach how to use Bowtie in order to map your reads to the reference genome, for example we have a thousand reads files as the output of NGS machine, in order to select the aligned reads with the reference genome we need to map those reads to the reference genome. For example: it aligns short DNA sequences (reads) to the human genome at a rate of over 25 million 35-bp reads per hour. Bowtie indexes the genome with a Burrows-Wheeler index to keep its memory footprint small: typically, about 2.2 GB for the human genome (2.9 GB for paired-end).

The main objective of this learning experience are:

- To understand what is the input files (the format, what kind of files needed) for the bowtie
- To understand how to process the data using bowtie
- To understand what is the output format and how to interpret it

Material

2. Software bowtie
3. FASTQ file
4. Reference genome file

Equipment

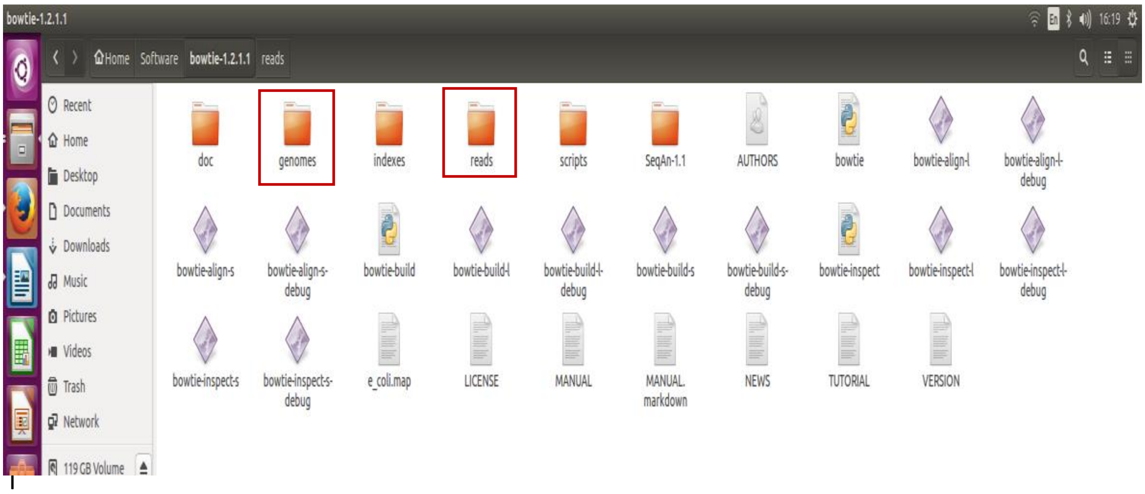
3. Logbook
4. Laptop/PC (available in Bioinformatics laboratory)

Click here to enter text.

Procedure

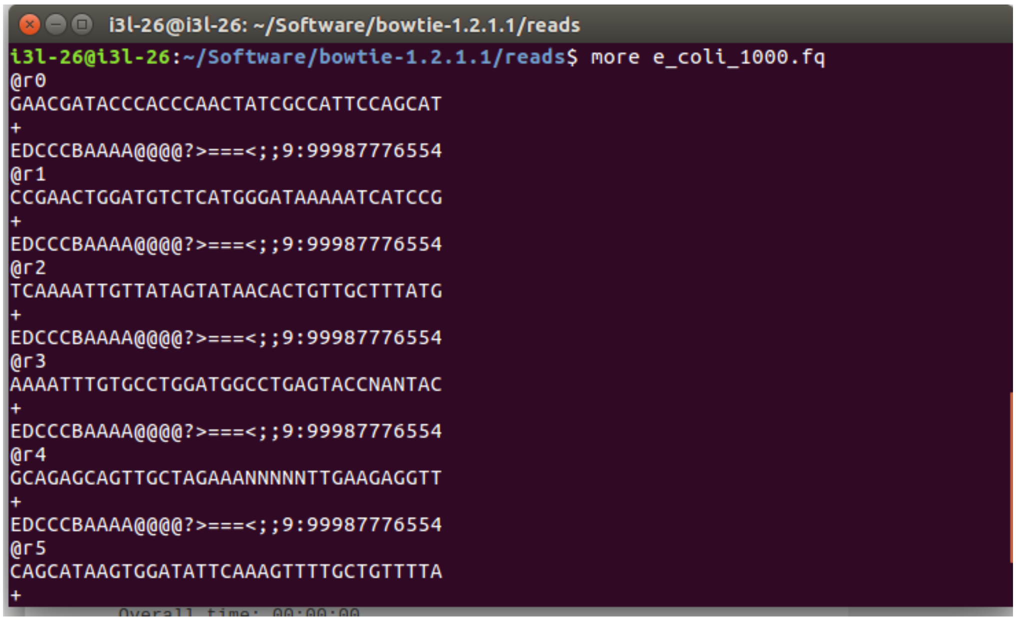
1. Open your linux terminal and go to this path: /home/i3l-26/software/bowtie-1.2.1.1
2. We need to have fastq format file (reads files) and reference genome file. The goal is we want to map the fastq file to the reference genome file. You can prepare your own dataset or find some available data on the internet. This is the link that discuss how to download raw sequence data in fastq format: https://www.biostars.org/p/111040/ and this is the link to download the reference genome: https://www.ensembl.org/info/data/ftp/index.html

   Bowtie already provide the reference genome and fastq format file in the folder or directory **genomes** and **reads**, respectively. Under the path /home/i3l-26/software/bowtie-1.2.1.1:



der reads you will see some reads files in the FASTQ format (.fq). For example, e_coli_1000.fq

If you want to see how the file looks like, open your terminal and move to directory **/home/i3l-26/software/bowtie-1.2.1.1/reads** (type: **cd /home/i3l-26/software/bowtie-1.2.1.1/reads**) And then (type: **more e_coli_1000.fq**), the file is a set of 1,000 35-bp reads.

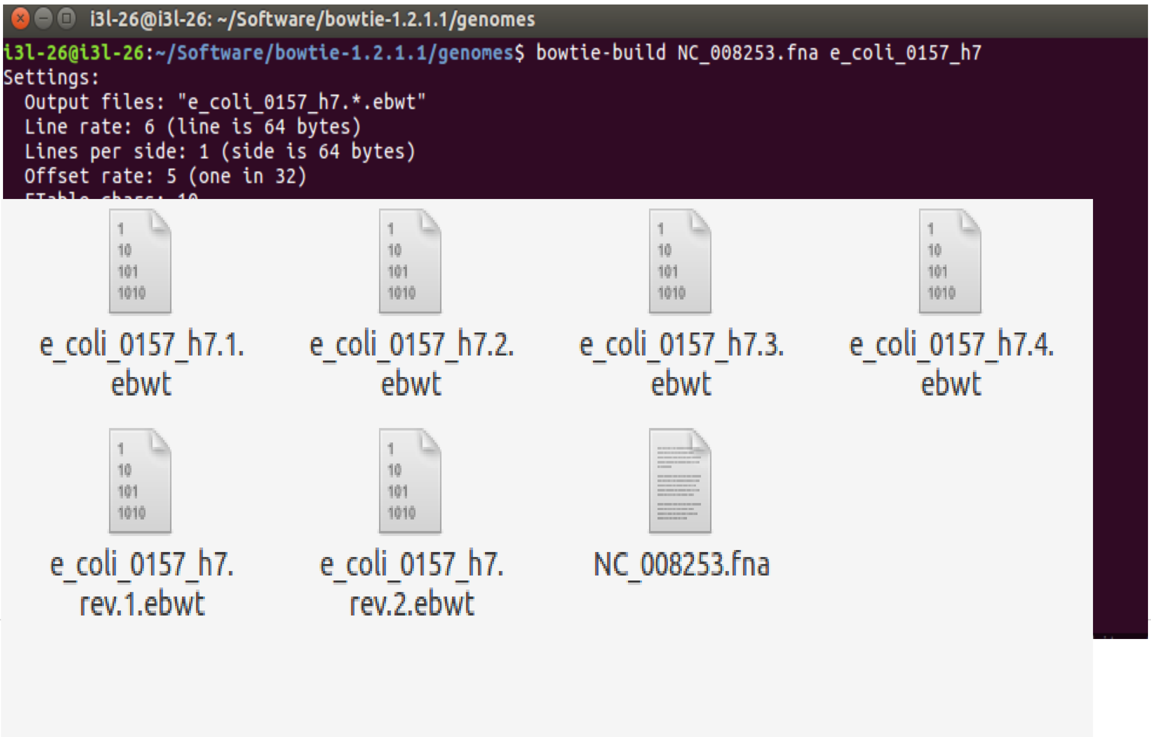


```
i3l-26@i3l-26: ~/Software/bowtie-1.2.1.1/reads
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/reads$ more e_coli_1000.fq
@r0
GAACGATACCCACCCAACTATCGCCATTCCAGCAT
+
EDCCCBAAAA@@@@?>===<;;9:99987776554
@r1
CCGAACTGGATGTCTCATGGGATAAAAATCATCCG
+
EDCCCBAAAA@@@@?>===<;;9:99987776554
@r2
TCAAAATTGTTATAGTATAACACTGTTGCTTTATG
+
EDCCCBAAAA@@@@?>===<;;9:99987776554
@r3
AAAATTTGTGCCTGGATGGCCTGAGTACCNANTAC
+
EDCCCBAAAA@@@@?>===<;;9:99987776554
@r4
GCAGAGCAGTTGCTAGAAANNNNNTTGAAGAGGTT
+
EDCCCBAAAA@@@@?>===<;;9:99987776554
@r5
CAGCATAAGTGGATATTCAAAGTTTTGCTGTTTTA
+
     Overall time: 00:00:00
```

In the folder genome you will see a reference genome file **NC_008253.fna**. It is a fasta format file which consist of a complete set of DNA in a genome.

3. Before we map the reads to the reference genome, we need to create **index file** of the reference genome fasta file. Bowtie indexes the genome with Burrows-Wheeler index to keep its memory footprint small. Go to the folder genome (type: **cd /home/i3l-26/software/bowtie-1.2.1.1/genome**) and then type **bowtie-build NC_008253.fna e_coli_0157_h7**

NC_008253.fna is the genome file name
e_coli_0157_h7 is the basename output of indexed file

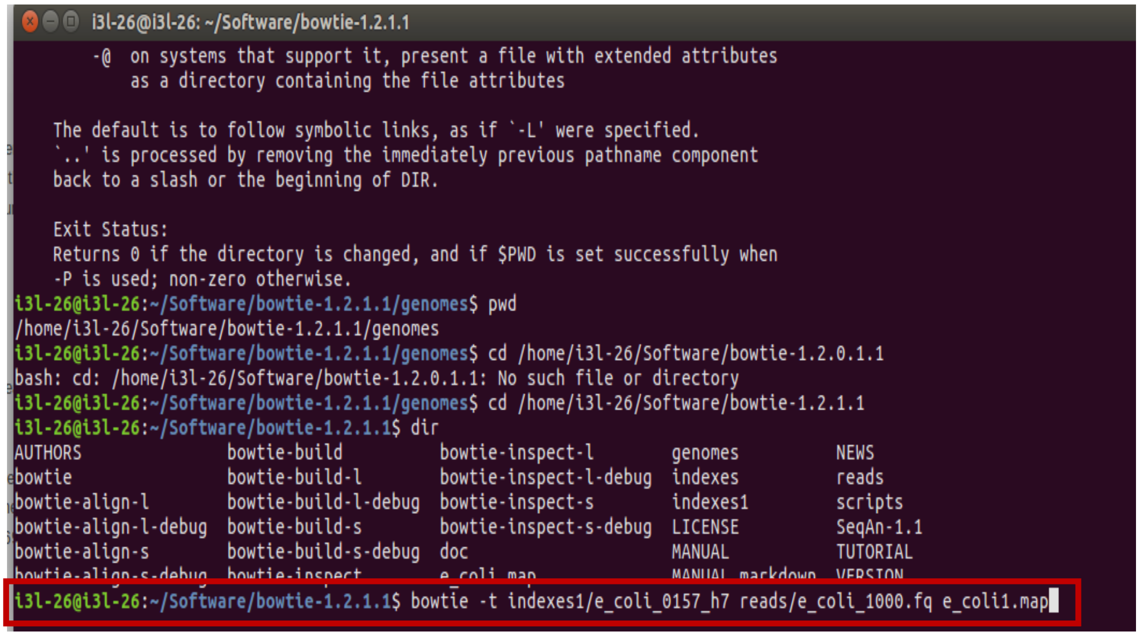


```
i3l-26@i3l-26: ~/Software/bowtie-1.2.1.1/genomes
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ bowtie-build NC_008253.fna e_coli_0157_h7
Settings:
  Output files: "e_coli_0157_h7.*.ebwt"
  Line rate: 6 (line is 64 bytes)
  Lines per side: 1 (side is 64 bytes)
  Offset rate: 5 (one in 32)
  FTable chars: 10
```

in the folder genome, there will be some index files:

Move all the index files into a folder, for example create a folder **index1** under the path: **/home/i3l-26/software/bowtie-1.2.1.1/** and move all the files to that folder.

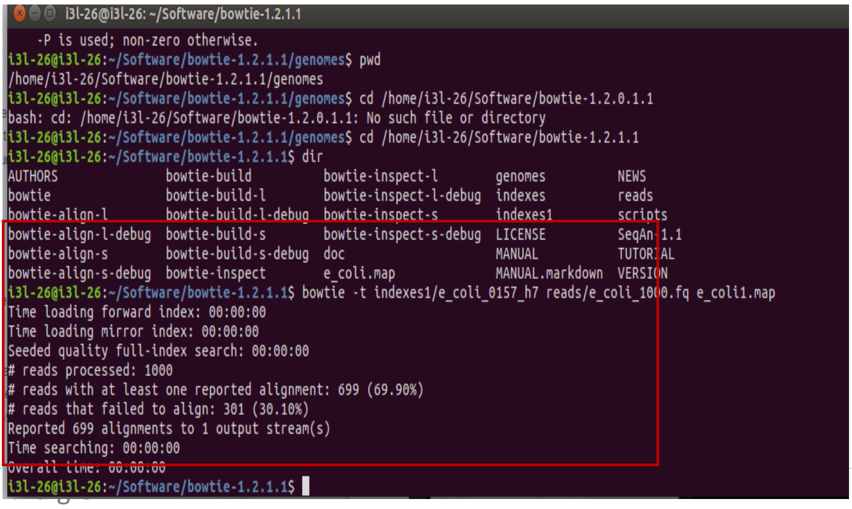
4. Then, we can map the reads file in the folder reads with the index files in the folder index1. Make sure your current directory is in **/home/i3l-26/software/bowtie-1.2.1.1/**
   Then type **bowtie –t indexes1/e_coli_0157_h7 reads/e_coli_1000.fq e_colli1.map**



```
i3l-26@i3l-26: ~/Software/bowtie-1.2.1.1
      -@  on systems that support it, present a file with extended attributes
          as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
    back to a slash or the beginning of DIR.

    Exit Status:
    Returns 0 if the directory is changed, and if $PWD is set successfully when
    -P is used; non-zero otherwise.
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ pwd
/home/i3l-26/Software/bowtie-1.2.1.1/genomes
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ cd /home/i3l-26/Software/bowtie-1.2.0.1.1
bash: cd: /home/i3l-26/Software/bowtie-1.2.0.1.1: No such file or directory
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ cd /home/i3l-26/Software/bowtie-1.2.1.1
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ dir
AUTHORS               bowtie-build          bowtie-inspect-l        genomes          NEWS
bowtie                bowtie-build-l        bowtie-inspect-l-debug  indexes          reads
bowtie-align-l        bowtie-build-l-debug  bowtie-inspect-s        indexes1         scripts
bowtie-align-l-debug  bowtie-build-s        bowtie-inspect-s-debug  LICENSE          SeqAn-1.1
bowtie-align-s        bowtie-build-s-debug  doc                     MANUAL           TUTORIAL
bowtie-align-s-debug  bowtie-inspect        e_coli.map              MANUAL.markdown  VERSION
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ bowtie -t indexes1/e_coli_0157_h7 reads/e_coli_1000.fq e_coli1.map
```

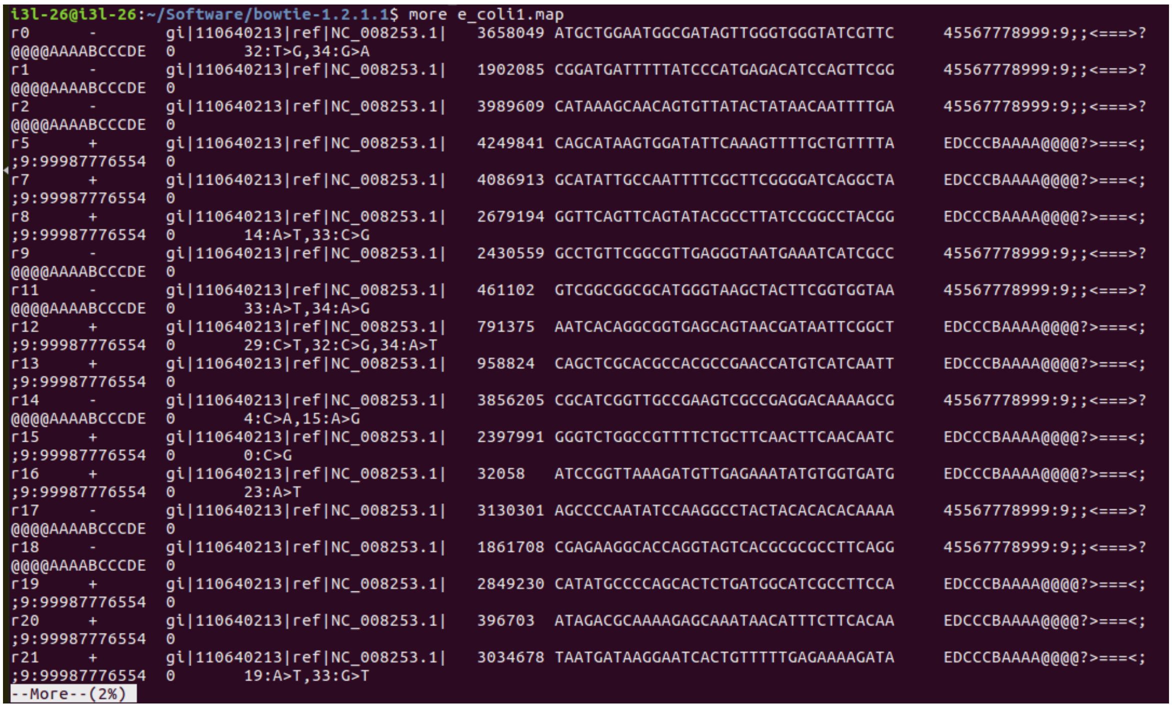
5. Then, there will be report on your terminal:

   It shows that there are 699 from 1000 (69.90%) reads with at least one reported alignment and there are 301 from 1000 (30.10%) reads that failed to align.
   And in the path **/home/i3l-26/software/bowtie-1.2.1.1/** you will also obtain one output called **e_coli1.map** with the default bowtie output; you can see the output format (type: **more e_coli1.map**)

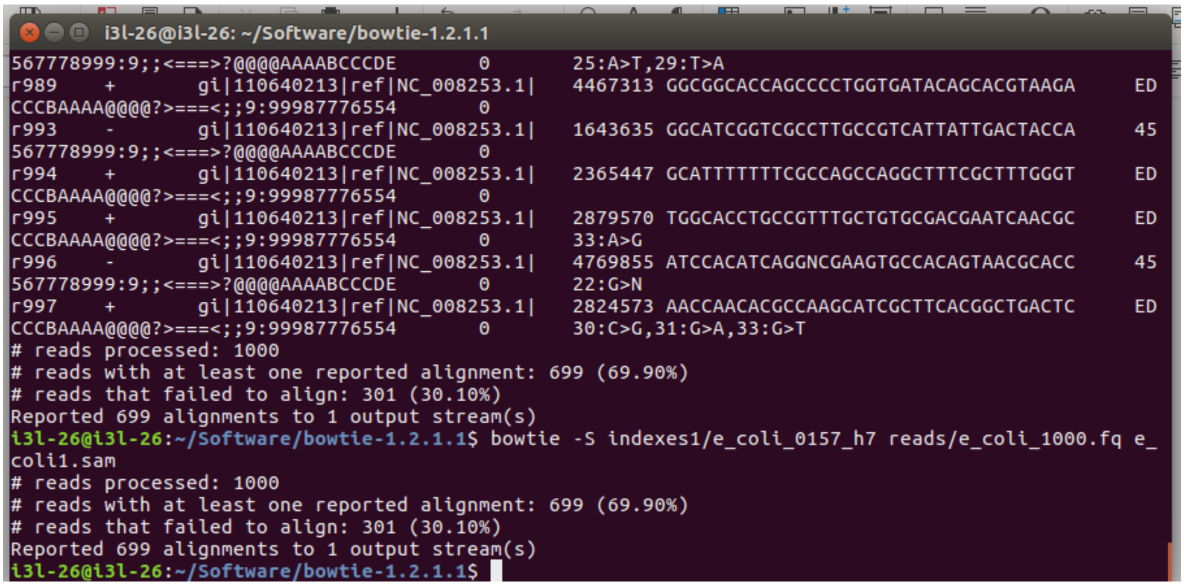


```
i3l-26@i3l-26: ~/Software/bowtie-1.2.1.1
    -P is used; non-zero otherwise.
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ pwd
/home/i3l-26/Software/bowtie-1.2.1.1/genomes
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ cd /home/i3l-26/Software/bowtie-1.2.0.1.1
bash: cd: /home/i3l-26/Software/bowtie-1.2.0.1.1: No such file or directory
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1/genomes$ cd /home/i3l-26/Software/bowtie-1.2.1.1
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ dir
AUTHORS               bowtie-build          bowtie-inspect-l        genomes          NEWS
bowtie                bowtie-build-l        bowtie-inspect-l-debug  indexes          reads
bowtie-align-l        bowtie-build-l-debug  bowtie-inspect-s        indexes1         scripts
bowtie-align-l-debug  bowtie-build-s        bowtie-inspect-s-debug  LICENSE          SeqAn-1.1
bowtie-align-s        bowtie-build-s-debug  doc                     MANUAL           TUTORIAL
bowtie-align-s-debug  bowtie-inspect        e_coli.map              MANUAL.markdown  VERSION
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ bowtie -t indexes1/e_coli_0157_h7 reads/e_coli_1000.fq e_coli1.map
Time loading forward index: 00:00:00
Time loading mirror index: 00:00:00
Seeded quality full-index search: 00:00:00
# reads processed: 1000
# reads with at least one reported alignment: 699 (69.90%)
# reads that failed to align: 301 (30.10%)
Reported 699 alignments to 1 output stream(s)
Time searching: 00:00:00
Overall time: 00:00:00
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$
```

6. We also can create an output file in the **.sam format**
   (type: **bowtie -S indexes1/e_coli_0157_h7 reads/e_coli_1000snp.fq e_coli1.sam**)

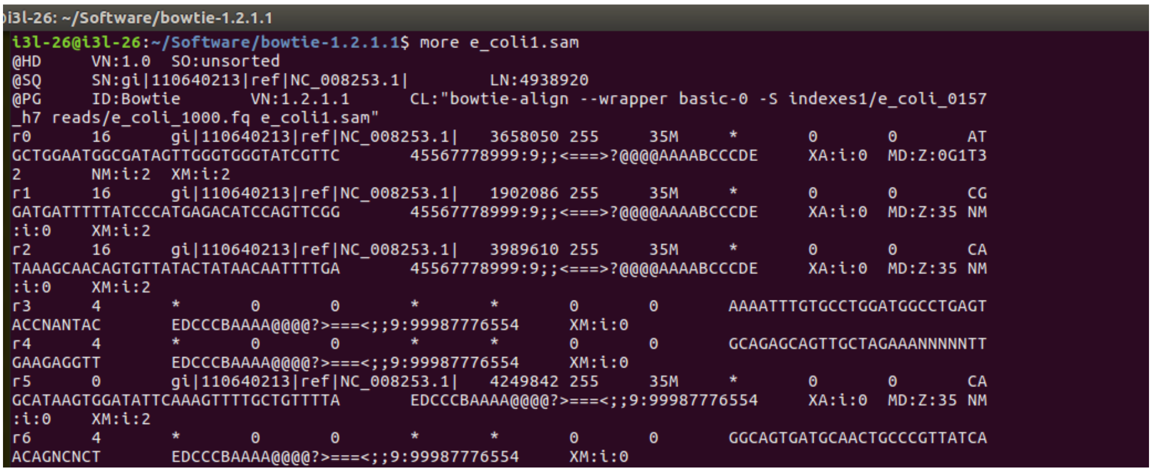


```
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ more e_coli1.map
r0      -       gi|110640213|ref|NC_008253.1|   3658049 ATGCTGGAATGGCGATAGTTGGGTGGGTATCGTTC     45567778999:9;;<===>?
@@@@AAAABCCCDE  0       32:T>G,34:G>A
r1      -       gi|110640213|ref|NC_008253.1|   1902085 CGGATGATTTTTATCCCATGAGACATCCAGTTCGG     45567778999:9;;<===>?
@@@@AAAABCCCDE  0
r2      -       gi|110640213|ref|NC_008253.1|   3989609 CATAAAGCAACAGTGTTATACTATAACAATTTTGA     45567778999:9;;<===>?
@@@@AAAABCCCDE  0
r5      +       gi|110640213|ref|NC_008253.1|   4249841 CAGCATAAGTGGATATTCAAAGTTTTGCTGTTTTA     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0
r7      +       gi|110640213|ref|NC_008253.1|   4086913 GCATATTGCCAATTTTCGCTTCGGGGATCAGGCTA     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0
r8      +       gi|110640213|ref|NC_008253.1|   2679194 GGTTCAGTTCAGTATACGCCTTATCCGGCCTACGG     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0       14:A>T,33:C>G
r9      -       gi|110640213|ref|NC_008253.1|   2430559 GCCTGTTCGGCGTTGAGGGTAATGAAATCATCGCC     45567778999:9;;<===>?
@@@@AAAABCCCDE  0
r11     -       gi|110640213|ref|NC_008253.1|   461102  GTCGGCGGCGCATGGGTAAGCTACTTCGGTGGTAA     45567778999:9;;<===>?
@@@@AAAABCCCDE  0       33:A>T,34:A>G
r12     +       gi|110640213|ref|NC_008253.1|   791375  AATCACAGGCGGTGAGCAGTAACGATAATTCGGCT     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0       29:C>T,32:C>G,34:A>T
r13     +       gi|110640213|ref|NC_008253.1|   958824  CAGCTCGCACGCCACGCCGAACCATGTCATCAATT     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0
r14     -       gi|110640213|ref|NC_008253.1|   3856205 CGCATCGGTTGCCGAAGTCGCCGAGGACAAAAGCG     45567778999:9;;<===>?
@@@@AAAABCCCDE  0       4:C>A,15:A>G
r15     +       gi|110640213|ref|NC_008253.1|   2397991 GGGTCTGGCCGTTTTCTGCTTCAACTTCAACAATC     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0       0:C>G
r16     +       gi|110640213|ref|NC_008253.1|   32058   ATCCGGTTAAAGATGTTGAGAAATATGTGGTGATG     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0       23:A>T
r17     -       gi|110640213|ref|NC_008253.1|   3130301 AGCCCCAATATCCAAGGCCTACTACACACACAAAA     45567778999:9;;<===>?
@@@@AAAABCCCDE  0
r18     -       gi|110640213|ref|NC_008253.1|   1861708 CGAGAAGGCACCAGGTAGTCACGCGCGCCTTCAGG     45567778999:9;;<===>?
@@@@AAAABCCCDE  0
r19     +       gi|110640213|ref|NC_008253.1|   2849230 CATATGCCCCAGCACTCTGATGGCATCGCCTTCCA     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0
r20     +       gi|110640213|ref|NC_008253.1|   396703  ATAGACGCAAAAGAGCAAATAACATTTCTTCACAA     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0
r21     +       gi|110640213|ref|NC_008253.1|   3034678 TAATGATAAGGAATCACTGTTTTTGAGAAAAGATA     EDCCCBAAAA@@@@?>===<;
;9:99987776554  0       19:A>T,33:G>T
--More--(2%)
```



```
567778999:9;;<===>?@@@@AAAABCCCDE       0       25:A>T,29:T>A
r989    +       gi|110640213|ref|NC_008253.1|   4467313 GGCGGCACCAGCCCCTGGTGATACAGCACGTAAGA     ED
CCCBAAAA@@@@?>===<;;9:99987776554       0
r993    -       gi|110640213|ref|NC_008253.1|   1643635 GGCATCGGTCGCCTTGCCGTCATTATTGACTACCA     45
567778999:9;;<===>?@@@@AAAABCCCDE       0
r994    +       gi|110640213|ref|NC_008253.1|   2365447 GCATTTTTTTCGCCAGCCAGGCTTTCGCTTTGGGT     ED
CCCBAAAA@@@@?>===<;;9:99987776554       0
r995    +       gi|110640213|ref|NC_008253.1|   2879570 TGGCACCTGCCGTTTGCTGTGCGACGAATCAACGC     ED
CCCBAAAA@@@@?>===<;;9:99987776554       0       33:A>G
r996    -       gi|110640213|ref|NC_008253.1|   4769855 ATCCACATCAGGNCGAAGTGCCACAGTAACGCACC     45
567778999:9;;<===>?@@@@AAAABCCCDE       0       22:G>N
r997    +       gi|110640213|ref|NC_008253.1|   2824573 AACCAACACGCCAAGCATCGCTTCACGGCTGACTC     ED
CCCBAAAA@@@@?>===<;;9:99987776554       0       30:C>G,31:G>A,33:G>T
# reads processed: 1000
# reads with at least one reported alignment: 699 (69.90%)
# reads that failed to align: 301 (30.10%)
Reported 699 alignments to 1 output stream(s)
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ bowtie -S indexes1/e_coli_0157_h7 reads/e_coli_1000.fq e_
coli1.sam
# reads processed: 1000
# reads with at least one reported alignment: 699 (69.90%)
# reads that failed to align: 301 (30.10%)
Reported 699 alignments to 1 output stream(s)
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$
```

7. We can check the format of sam files (type: **more e_coli1.sam**)



```
i3l-26: ~/Software/bowtie-1.2.1.1
i3l-26@i3l-26:~/Software/bowtie-1.2.1.1$ more e_coli1.sam
@HD     VN:1.0  SO:unsorted
@SQ     SN:gi|110640213|ref|NC_008253.1|        LN:4938920
@PG     ID:Bowtie       VN:1.2.1.1      CL:"bowtie-align --wrapper basic-0 -S indexes1/e_coli_0157
_h7 reads/e_coli_1000.fq e_coli1.sam"
r0      16      gi|110640213|ref|NC_008253.1|   3658050 255     35M     *       0       0       AT
GCTGGAATGGCGATAGTTGGGTGGGTATCGTTC       45567778999:9;;<===>?@@@@AAAABCCCDE     XA:i:0  MD:Z:0G1T3
2       NM:i:2  XM:i:2
r1      16      gi|110640213|ref|NC_008253.1|   1902086 255     35M     *       0       0       CG
GATGATTTTTATCCCATGAGACATCCAGTTCGG       45567778999:9;;<===>?@@@@AAAABCCCDE     XA:i:0  MD:Z:35 NM
:i:0    XM:i:2
r2      16      gi|110640213|ref|NC_008253.1|   3989610 255     35M     *       0       0       CA
TAAAGCAACAGTGTTATACTATAACAATTTTGA       45567778999:9;;<===>?@@@@AAAABCCCDE     XA:i:0  MD:Z:35 NM
:i:0    XM:i:2
r3      4       *       0       0       *       *       0       0       AAAATTTGTGCCTGGATGGCCTGAGT
ACCNANTAC       EDCCCBAAAA@@@@?>===<;;9:99987776554     XM:i:0
r4      4       *       0       0       *       *       0       0       GCAGAGCAGTTGCTAGAAANNNNNTT
GAAGAGGTT       EDCCCBAAAA@@@@?>===<;;9:99987776554     XM:i:0
r5      0       gi|110640213|ref|NC_008253.1|   4249842 255     35M     *       0       0       CA
GCATAAGTGGATATTCAAAGTTTTGCTGTTTTA       EDCCCBAAAA@@@@?>===<;;9:99987776554     XA:i:0  MD:Z:35 NM
:i:0    XM:i:2
r6      4       *       0       0       *       *       0       0       GGCAGTGATGCAACTGCCCGTTATCA
ACAGNCNCT       EDCCCBAAAA@@@@?>===<;;9:99987776554     XM:i:0
```

Useful Links:

- http://bowtie-bio.sourceforge.net/manual.shtml#algn_out

Indonesia International Institute for Life Science

Transcriptomics

COURSE CODE: BI021

Cufflinks

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: David Agustriawan, Ph.D.

Email: david.agustriawan@i3l.ac.id

Supervisor(s)	Email
Dr.rer.nat Arli Aditya Parikesit	arli.parikesit@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

1. Operate ONLY the computer assigned to you.
    a. If you have any troubleshooting, please contact your supervisor or Building Management
    b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change the system preferences unless directed to do so
    c. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
    d. Do not bring food or drinks into the lab unless it is in your backpack

2. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	3
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

Cufflinks assembles transcripts, estimates their abundances, and tests for differential expression and regulation in RNA-Seq samples. It accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols.

The main objective of this learning experience are:

- To understand what is the input files (the format, what kind of files needed) for the cufflinks
- To understand how to process the data using cufflinks
- To understand what is the output format and how to interpret it
- To understand how to visualize the data using UCSC

Material

1. Software cufflinks
2. Software Tophat
3. Software Bowtie2
4. FASTQ file
5. Reference genome file
6. UCSC websites

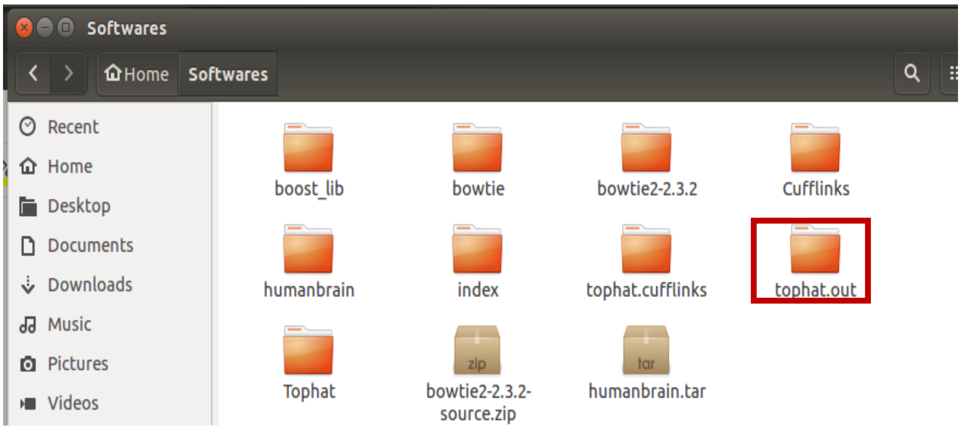
Equipment

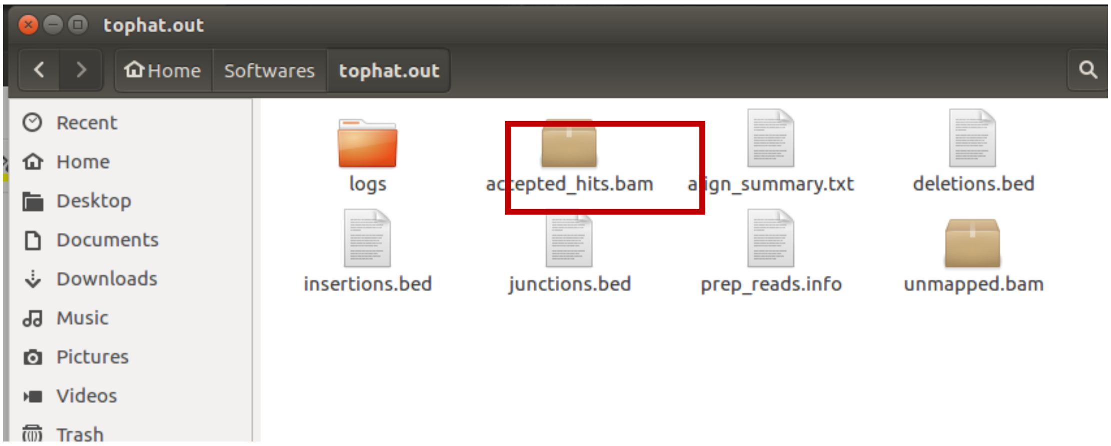
1. Logbook
2. Laptop/PC (available in Bioinformatics laboratory)

Click here to enter text.

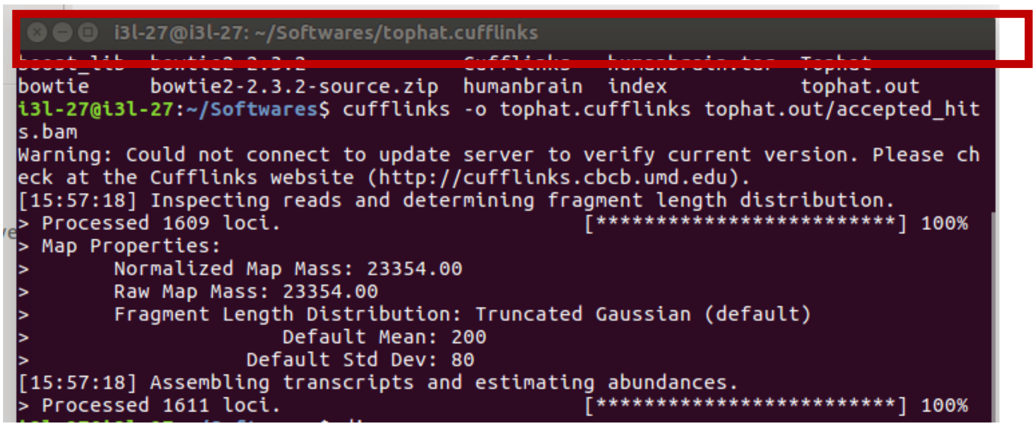
Procedure

1. If you already have the output from Tophat, you can run Cufflinks with it right away. Refer to Tophat output in your computer path at **home/i3l-27/Softwares/tophat.out** where inside the tophat.out folder there is a file with a name: **accepted_hits.bam**



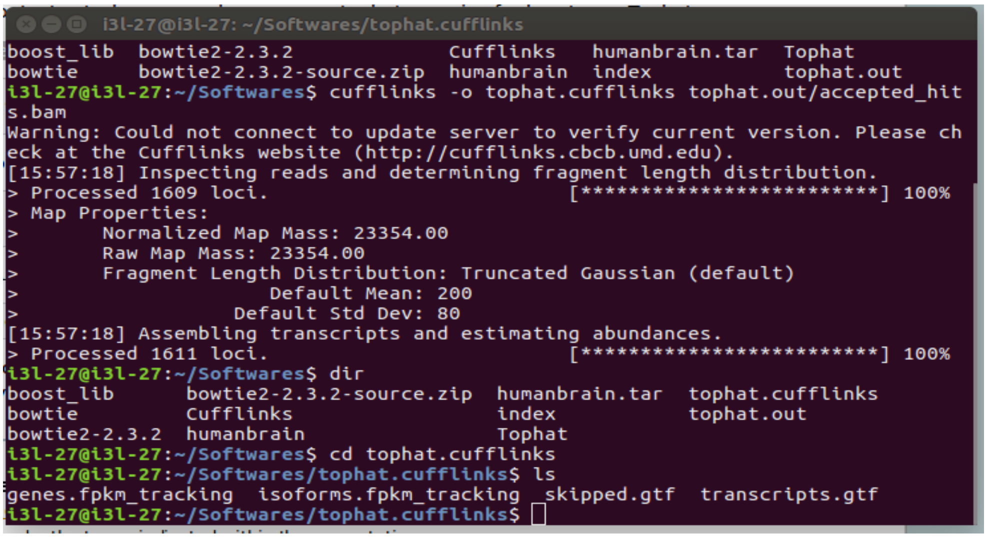


2. Go to this path: **home/i3l-27/Softwares,** then run cufflinks with this command: cufflinks -o tophat.cufflinks tophat.out/accepted_hits.bam



```
boost_lib  bowtie2-2.3.2            Cufflinks   humanbrain.tar  Tophat
bowtie     bowtie2-2.3.2-source.zip humanbrain  index           tophat.out
i3l-27@i3l-27:~/Softwares$ cufflinks -o tophat.cufflinks tophat.out/accepted_hit
s.bam
Warning: Could not connect to update server to verify current version. Please ch
eck at the Cufflinks website (http://cufflinks.cbcb.umd.edu).
[15:57:18] Inspecting reads and determining fragment length distribution.
> Processed 1609 loci.                              [*************************] 100%
> Map Properties:
>       Normalized Map Mass: 23354.00
>       Raw Map Mass: 23354.00
>       Fragment Length Distribution: Truncated Gaussian (default)
>                     Default Mean: 200
>                  Default Std Dev: 80
[15:57:18] Assembling transcripts and estimating abundances.
> Processed 1611 loci.                              [*************************] 100%
```

3. To explore the output, go to the output directory: **home/i3l-27/Softwares/tophat.cufflinks** then **type: ls**

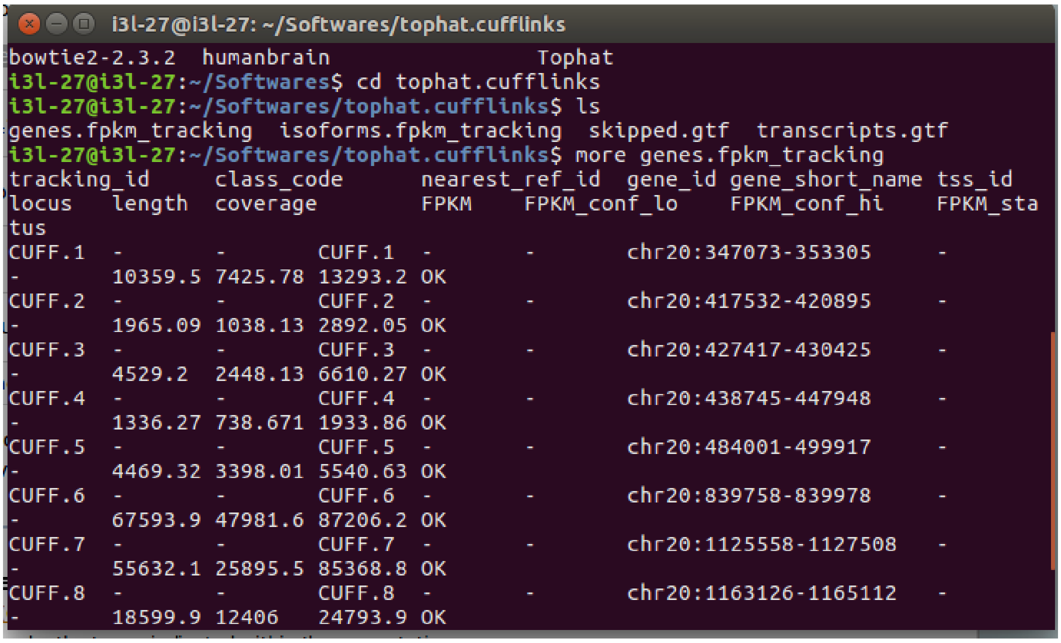


```
boost_lib  bowtie2-2.3.2            Cufflinks   humanbrain.tar  Tophat
bowtie     bowtie2-2.3.2-source.zip humanbrain  index           tophat.out
i3l-27@i3l-27:~/Softwares$ cufflinks -o tophat.cufflinks tophat.out/accepted_hit
s.bam
Warning: Could not connect to update server to verify current version. Please ch
eck at the Cufflinks website (http://cufflinks.cbcb.umd.edu).
[15:57:18] Inspecting reads and determining fragment length distribution.
> Processed 1609 loci.                              [*************************] 100%
> Map Properties:
>       Normalized Map Mass: 23354.00
>       Raw Map Mass: 23354.00
>       Fragment Length Distribution: Truncated Gaussian (default)
>                     Default Mean: 200
>                  Default Std Dev: 80
[15:57:18] Assembling transcripts and estimating abundances.
> Processed 1611 loci.                              [*************************] 100%
i3l-27@i3l-27:~/Softwares$ dir
boost_lib      bowtie2-2.3.2-source.zip  humanbrain.tar  tophat.cufflinks
bowtie         Cufflinks                 index           tophat.out
bowtie2-2.3.2  humanbrain                Tophat
i3l-27@i3l-27:~/Softwares$ cd tophat.cufflinks
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ ls
genes.fpkm_tracking  isoforms.fpkm_tracking  skipped.gtf  transcripts.gtf
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ 
```

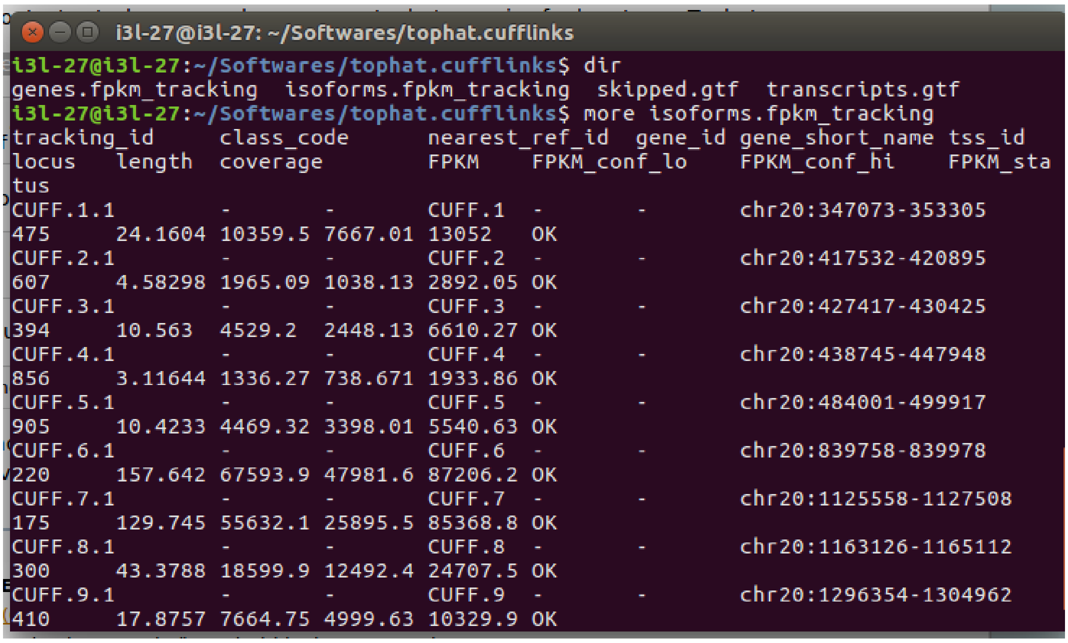
transcripts.gtf: Its a GTF file you can visualise it in a genome browser (gbrowser ucsc etc)
isoforms.fpkm_tracking: Expression values for the transcripts expressed

genes.fpkm_tracking: Expression values for the genes expressed

4. You can check the output results with the command: **more** (remember you can press enter to see more data output, or press q to quit):



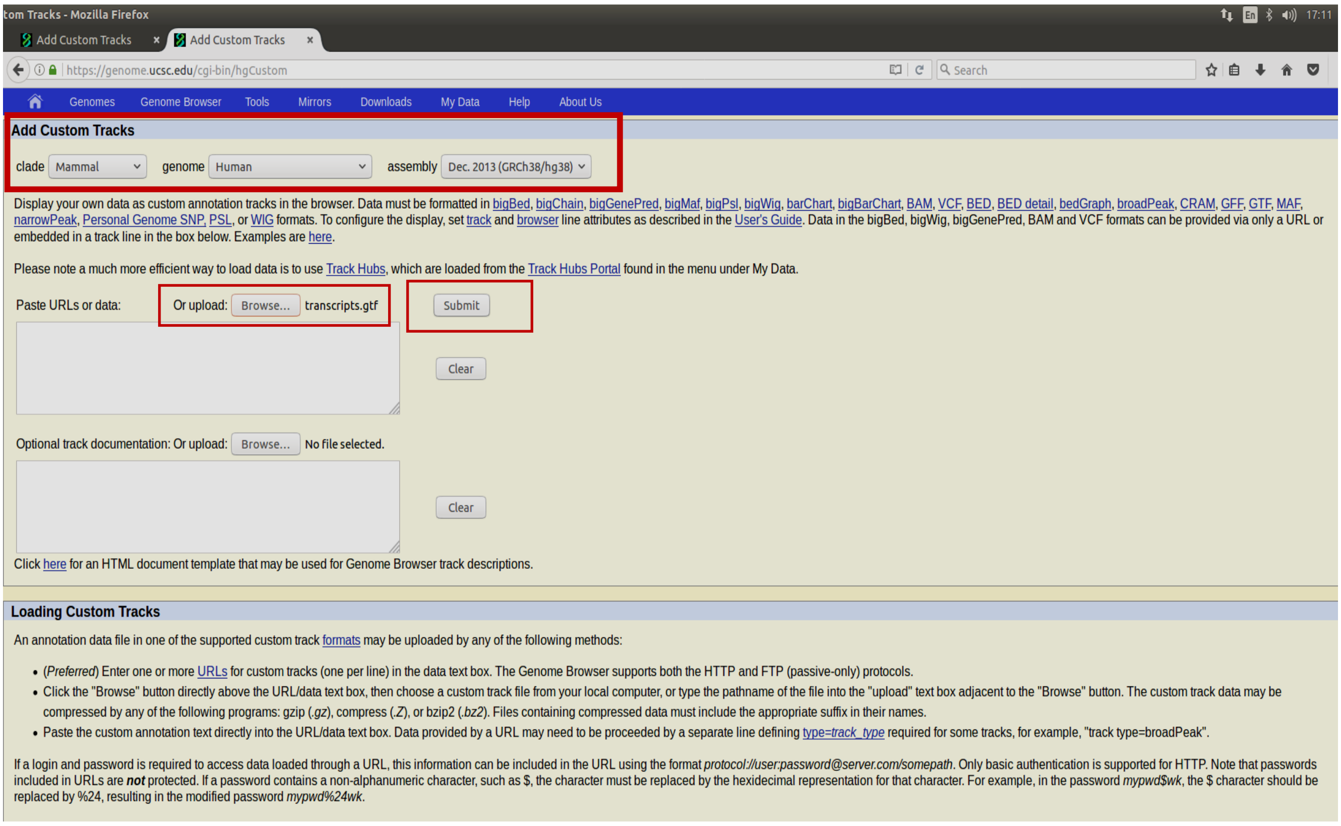
```
bowtie2-2.3.2  humanbrain                Tophat
i3l-27@i3l-27:~/Softwares$ cd tophat.cufflinks
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ ls
genes.fpkm_tracking  isoforms.fpkm_tracking  skipped.gtf  transcripts.gtf
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ more genes.fpkm_tracking
tracking_id     class_code      nearest_ref_id  gene_id gene_short_name tss_id
locus   length  coverage        FPKM    FPKM_conf_lo    FPKM_conf_hi    FPKM_sta
tus
CUFF.1  -       -       CUFF.1  -       -       chr20:347073-353305     -
-       10359.5 7425.78 13293.2 OK
CUFF.2  -       -       CUFF.2  -       -       chr20:417532-420895     -
-       1965.09 1038.13 2892.05 OK
CUFF.3  -       -       CUFF.3  -       -       chr20:427417-430425     -
-       4529.2  2448.13 6610.27 OK
CUFF.4  -       -       CUFF.4  -       -       chr20:438745-447948     -
-       1336.27 738.671 1933.86 OK
CUFF.5  -       -       CUFF.5  -       -       chr20:484001-499917     -
-       4469.32 3398.01 5540.63 OK
CUFF.6  -       -       CUFF.6  -       -       chr20:839758-839978     -
-       67593.9 47981.6 87206.2 OK
CUFF.7  -       -       CUFF.7  -       -       chr20:1125558-1127508   -
-       55632.1 25895.5 85368.8 OK
CUFF.8  -       -       CUFF.8  -       -       chr20:1163126-1165112   -
-       18599.9 12406   24793.9 OK
```



```
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ dir
genes.fpkm_tracking  isoforms.fpkm_tracking  skipped.gtf  transcripts.gtf
i3l-27@i3l-27:~/Softwares/tophat.cufflinks$ more isoforms.fpkm_tracking
tracking_id     class_code      nearest_ref_id  gene_id gene_short_name tss_id
locus   length  coverage        FPKM    FPKM_conf_lo    FPKM_conf_hi    FPKM_sta
tus
CUFF.1.1        -       -       CUFF.1  -       -       chr20:347073-353305
475     24.1604 10359.5 7667.01 13052   OK
CUFF.2.1        -       -       CUFF.2  -       -       chr20:417532-420895
607     4.58298 1965.09 1038.13 2892.05 OK
CUFF.3.1        -       -       CUFF.3  -       -       chr20:427417-430425
394     10.563  4529.2  2448.13 6610.27 OK
CUFF.4.1        -       -       CUFF.4  -       -       chr20:438745-447948
856     3.11644 1336.27 738.671 1933.86 OK
CUFF.5.1        -       -       CUFF.5  -       -       chr20:484001-499917
905     10.4233 4469.32 3398.01 5540.63 OK
CUFF.6.1        -       -       CUFF.6  -       -       chr20:839758-839978
220     157.642 67593.9 47981.6 87206.2 OK
CUFF.7.1        -       -       CUFF.7  -       -       chr20:1125558-1127508
175     129.745 55632.1 25895.5 85368.8 OK
CUFF.8.1        -       -       CUFF.8  -       -       chr20:1163126-1165112
300     43.3788 18599.9 12492.4 24707.5 OK
CUFF.9.1        -       -       CUFF.9  -       -       chr20:1296354-1304962
410     17.8757 7664.75 4999.63 10329.9 OK
```

5. To visualize output file transcripts.gtf, go to UCSC web https://genome.ucsc.edu/cgi-bin/hgCustom
   - Choose the parameter:
     Clade: mammal
     Genome: human
     Browse: browse your data transcripts.gtf from your PC

- Then click submit



Add Custom Tracks

clade Mammal genome Human assembly Dec. 2013 (GRCh38/hg38)

Display your own data as custom annotation tracks in the browser. Data must be formatted in bigBed, bigChain, bigGenePred, bigMaf, bigPsl, bigWig, barChart, bigBarChart, BAM, VCF, BED, BED detail, bedGraph, broadPeak, CRAM, GFF, GTF, MAF, narrowPeak, Personal Genome SNP, PSL, or WIG formats. To configure the display, set track and browser line attributes as described in the User's Guide. Data in the bigBed, bigWig, bigGenePred, BAM and VCF formats can be provided via only a URL or embedded in a track line in the box below. Examples are here.

Please note a much more efficient way to load data is to use Track Hubs, which are loaded from the Track Hubs Portal found in the menu under My Data.

Paste URLs or data: Or upload: Browse... transcripts.gtf

Optional track documentation: Or upload: Browse... No file selected.

Click here for an HTML document template that may be used for Genome Browser track descriptions.

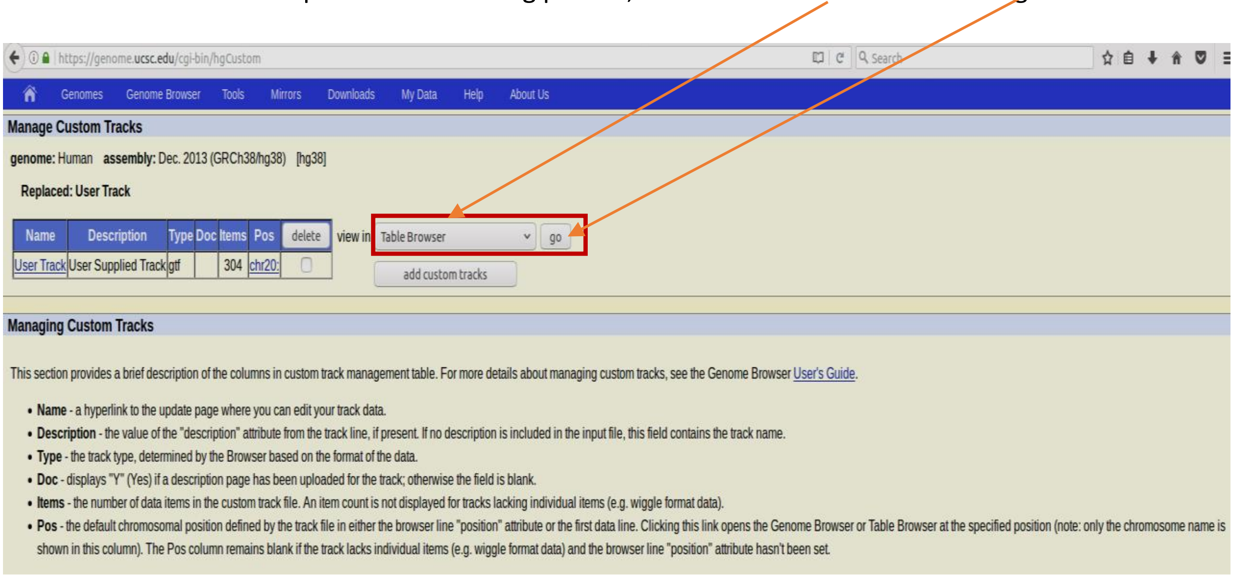
Loading Custom Tracks

An annotation data file in one of the supported custom track formats may be uploaded by any of the following methods:

- (*Preferred*) Enter one or more URLs for custom tracks (one per line) in the data text box. The Genome Browser supports both the HTTP and FTP (passive-only) protocols.
- Click the "Browse" button directly above the URL/data text box, then choose a custom track file from your local computer, or type the pathname of the file into the "upload" text box adjacent to the "Browse" button. The custom track data may be compressed by any of the following programs: gzip (*.gz*), compress (*.Z*), or bzip2 (*.bz2*). Files containing compressed data must include the appropriate suffix in their names.
- Paste the custom annotation text directly into the URL/data text box. Data provided by a URL may need to be proceeded by a separate line defining *type=track_type* required for some tracks, for example, "track type=broadPeak".

If a login and password is required to access data loaded through a URL, this information can be included in the URL using the format *protocol://user:password@server.com/somepath*. Only basic authentication is supported for HTTP. Note that passwords included in URLs are ***not*** protected. If a password contains a non-alphanumeric character, such as $, the character must be replaced by the hexidecimal representation for that character. For example, in the password *mypwd$wk*, the $ character should be replaced by %24, resulting in the modified password *mypwd%24wk*.

Then it will return the output in the following picture, choose table browser and click go:



Manage Custom Tracks

genome: Human **assembly:** Dec. 2013 (GRCh38/hg38) [hg38]

Replaced: User Track

Name	Description	Type	Doc	Items	Pos	delete
User Track	User Supplied Track	gtf		304	chr20:	□

view in Table Browser

Managing Custom Tracks

This section provides a brief description of the columns in custom track management table. For more details about managing custom tracks, see the Genome Browser User's Guide.

- **Name** - a hyperlink to the update page where you can edit your track data.
- **Description** - the value of the "description" attribute from the track line, if present. If no description is included in the input file, this field contains the track name.
- **Type** - the track type, determined by the Browser based on the format of the data.
- **Doc** - displays "Y" (Yes) if a description page has been uploaded for the track; otherwise the field is blank.
- **Items** - the number of data items in the custom track file. An item count is not displayed for tracks lacking individual items (e.g. wiggle format data).
- **Pos** - the default chromosomal position defined by the track file in either the browser line "position" attribute or the first data line. Clicking this link opens the Genome Browser or Table Browser at the specified position (note: only the chromosome name is shown in this column). The Pos column remains blank if the track lacks individual items (e.g. wiggle format data) and the browser line "position" attribute hasn't been set.

Then, picture below is the output, then click get output

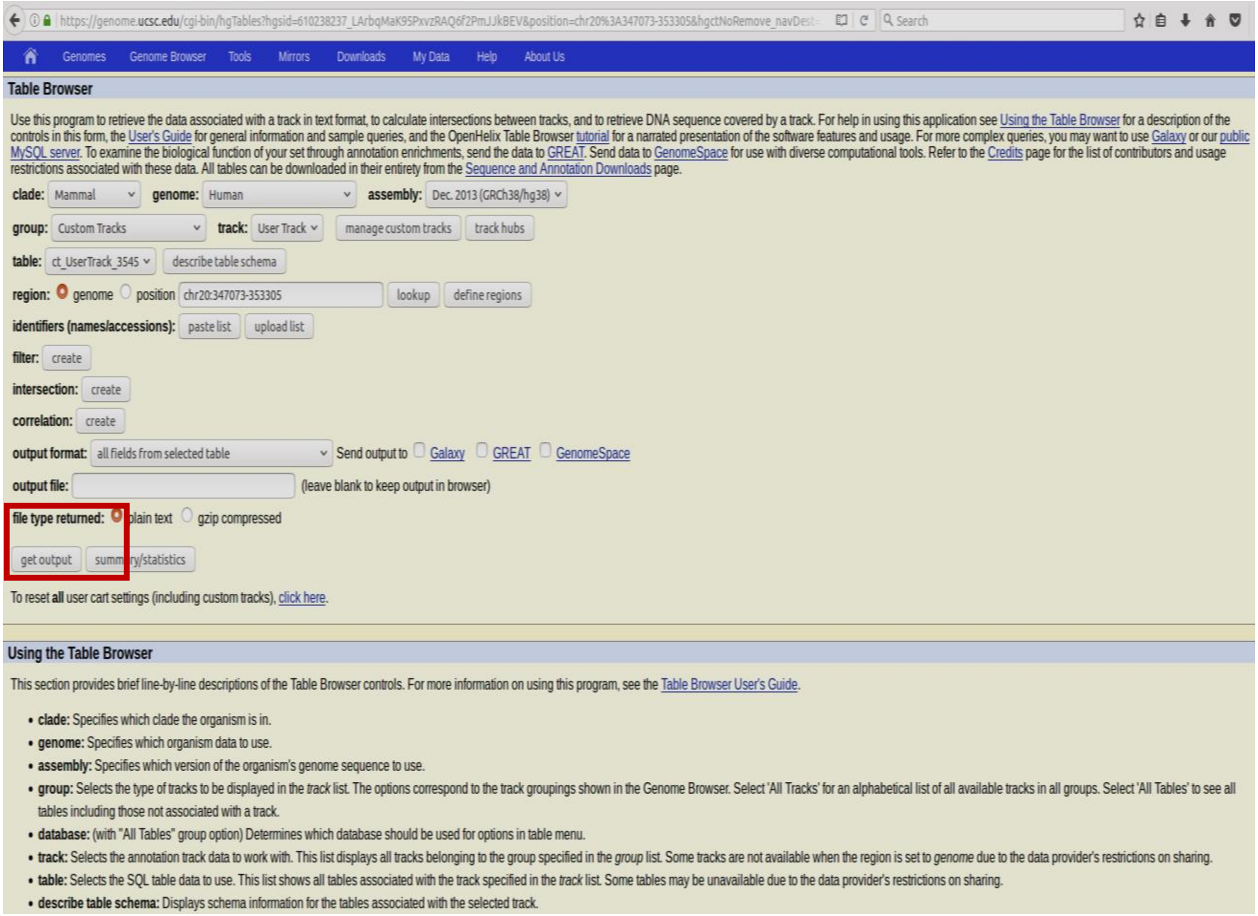

https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=610238237_LArbqMaK95PxvzRAQ6f2PmJJkBEV&position=chr20%3A347073-353305&hgctNoRemove_navDest=
Search
Genomes
Genome Browser
Tools
Mirrors
Downloads
My Data
Help
About Us

Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see Using the Table Browser for a description of the controls in this form, the User's Guide for general information and sample queries, and the OpenHelix Table Browser tutorial for a narrated presentation of the software features and usage. For more complex queries, you may want to use Galaxy or our public MySQL server. To examine the biological function of your set through annotation enrichments, send the data to GREAT. Send data to GenomeSpace for use with diverse computational tools. Refer to the Credits page for the list of contributors and usage restrictions associated with these data. All tables can be downloaded in their entirety from the Sequence and Annotation Downloads page.

clade: Mammal **genome:** Human **assembly:** Dec. 2013 (GRCh38/hg38)

group: Custom Tracks **track:** User Track manage custom tracks track hubs

table: ct_UserTrack_3545 describe table schema

region: ● genome ○ position chr20:347073-353305 lookup define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: all fields from selected table Send output to □ Galaxy □ GREAT □ GenomeSpace

output file: (leave blank to keep output in browser)

file type returned: ● plain text ○ gzip compressed

get output summary/statistics

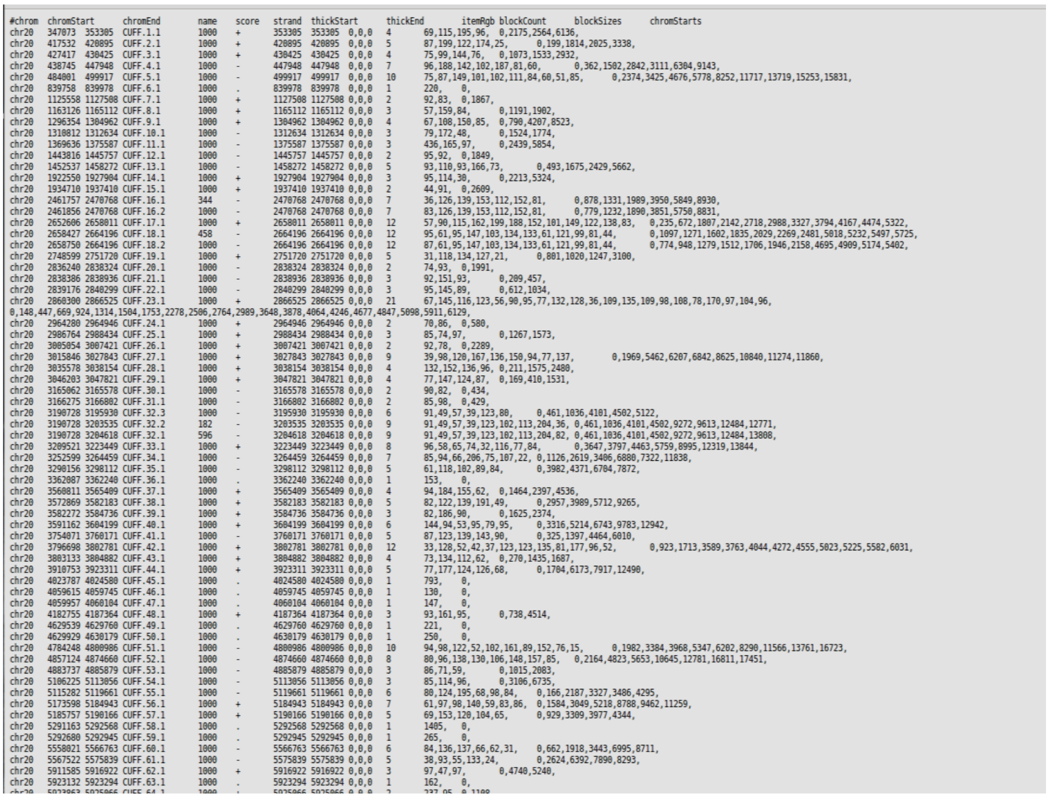
To reset **all** user cart settings (including custom tracks), click here.

Using the Table Browser

This section provides brief line-by-line descriptions of the Table Browser controls. For more information on using this program, see the Table Browser User's Guide.

- **clade:** Specifies which clade the organism is in.
- **genome:** Specifies which organism data to use.
- **assembly:** Specifies which version of the organism's genome sequence to use.
- **group:** Selects the type of tracks to be displayed in the *track* list. The options correspond to the track groupings shown in the Genome Browser. Select 'All Tracks' for an alphabetical list of all available tracks in all groups. Select 'All Tables' to see all tables including those not associated with a track.
- **database:** (with "All Tables" group option) Determines which database should be used for options in table menu.
- **track:** Selects the annotation track data to work with. This list displays all tracks belonging to the group specified in the *group* list. Some tracks are not available when the region is set to *genome* due to the data provider's restrictions on sharing.
- **table:** Selects the SQL table data to use. This list shows all tables associated with the track specified in the *track* list. Some tables may be unavailable due to the data provider's restrictions on sharing.
- **describe table schema:** Displays schema information for the tables associated with the selected track.

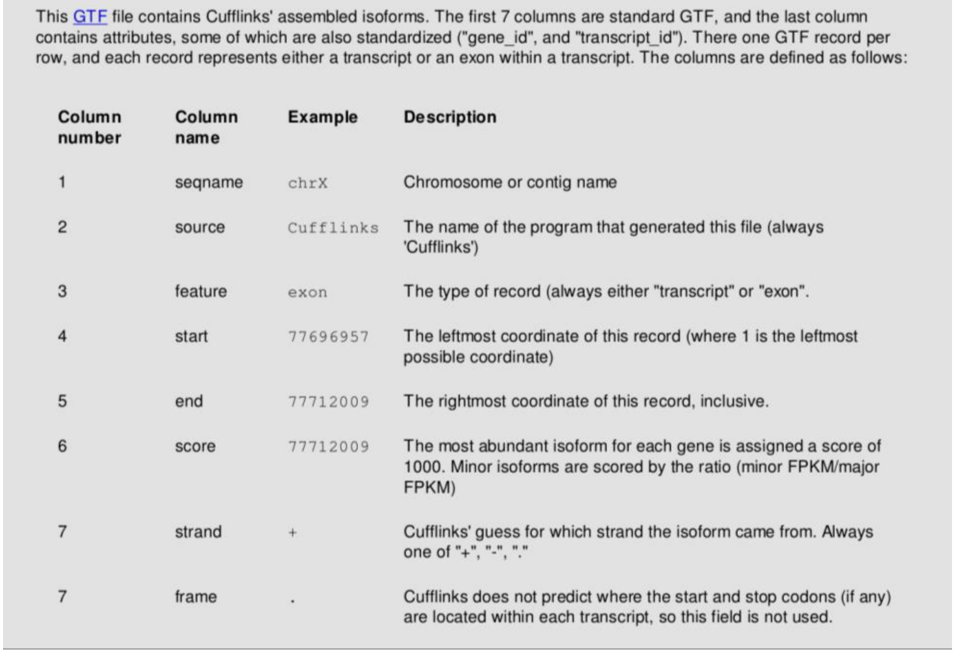
Then, it will return the output below:



```
#chrom  chromStart      chromEnd        name    score   strand  thickStart      thickEnd        itemRgb blockCount      blockSizes      chromStarts
chr20   347073  353305  CUFF.1.1        1000    +       353305  353305  0,0,0   4       69,115,195,96,  0,2175,2564,6136,
chr20   417532  420895  CUFF.2.1        1000    +       420895  420895  0,0,0   5       87,199,122,174,25,      0,199,1814,2025,3338,
chr20   427417  430425  CUFF.3.1        1000    +       430425  430425  0,0,0   4       75,99,144,76,   0,1073,1533,2932,
chr20   438745  447948  CUFF.4.1        1000    -       447948  447948  0,0,0   7       96,188,142,102,187,81,60,       0,362,1502,2842,3111,6304,9143,
chr20   484001  499917  CUFF.5.1        1000    -       499917  499917  0,0,0   10      75,87,149,101,102,111,84,60,51,85,      0,2374,3425,4676,5778,8252,11717,13719,15253,15831,
chr20   839758  839978  CUFF.6.1        1000    .       839978  839978  0,0,0   1       220,    0,
chr20   1125558 1127508 CUFF.7.1        1000    +       1127508 1127508 0,0,0   2       92,83,  0,1867,
chr20   1163126 1165112 CUFF.8.1        1000    +       1165112 1165112 0,0,0   3       57,159,84,      0,1191,1902,
chr20   1296354 1304962 CUFF.9.1        1000    +       1304962 1304962 0,0,0   4       67,108,150,85,  0,790,4207,8523,
chr20   1310812 1312634 CUFF.10.1       1000    -       1312634 1312634 0,0,0   3       79,172,48,      0,1524,1774,
chr20   1369636 1375587 CUFF.11.1       1000    -       1375587 1375587 0,0,0   3       436,165,97,     0,2439,5854,
chr20   1443816 1445757 CUFF.12.1       1000    -       1445757 1445757 0,0,0   2       95,92,  0,1849,
chr20   1452537 1458272 CUFF.13.1       1000    -       1458272 1458272 0,0,0   5       93,110,93,166,73,       0,493,1675,2429,5662,
chr20   1922550 1927904 CUFF.14.1       1000    +       1927904 1927904 0,0,0   3       95,114,30,      0,2213,5324,
chr20   1934710 1937410 CUFF.15.1       1000    +       1937410 1937410 0,0,0   2       44,91,  0,2609,
chr20   2461757 2470768 CUFF.16.1       344     -       2470768 2470768 0,0,0   7       36,126,139,153,112,152,81,      0,878,1331,1989,3950,5849,8930,
chr20   2461856 2470768 CUFF.16.2       1000    -       2470768 2470768 0,0,0   7       83,126,139,153,112,152,81,      0,779,1232,1890,3851,5750,8831,
chr20   2652606 2658011 CUFF.17.1       1000    +       2658011 2658011 0,0,0   12      57,90,115,162,199,188,152,101,149,122,138,83,   0,235,672,1807,2142,2718,2988,3327,3794,4167,4474,5322,
chr20   2658427 2664196 CUFF.18.1       458     -       2664196 2664196 0,0,0   12      95,61,95,147,103,134,133,61,121,99,81,44,       0,1097,1271,1602,1835,2029,2269,2481,5018,5232,5497,5725,
chr20   2658750 2664196 CUFF.18.2       1000    -       2664196 2664196 0,0,0   12      87,61,95,147,103,134,133,61,121,99,81,44,       0,774,948,1279,1512,1706,1946,2158,4695,4909,5174,5402,
chr20   2748599 2751720 CUFF.19.1       1000    +       2751720 2751720 0,0,0   5       31,118,134,127,21,      0,801,1020,1247,3100,
chr20   2836240 2838324 CUFF.20.1       1000    -       2838324 2838324 0,0,0   2       74,93,  0,1991,
chr20   2838386 2838936 CUFF.21.1       1000    -       2838936 2838936 0,0,0   3       92,151,93,      0,209,457,
chr20   2839176 2840299 CUFF.22.1       1000    -       2840299 2840299 0,0,0   3       95,145,89,      0,612,1034,
chr20   2860300 2866525 CUFF.23.1       1000    +       2866525 2866525 0,0,0   21      67,145,116,123,56,90,95,77,132,128,36,109,135,109,98,108,78,170,97,104,96,
0,148,447,669,924,1314,1504,1753,2278,2506,2764,2989,3648,3878,4064,4246,4677,4847,5098,5911,6129,
chr20   2964280 2964946 CUFF.24.1       1000    +       2964946 2964946 0,0,0   2       70,86,  0,580,
chr20   2986764 2988434 CUFF.25.1       1000    +       2988434 2988434 0,0,0   3       85,74,97,       0,1267,1573,
chr20   3005054 3007421 CUFF.26.1       1000    +       3007421 3007421 0,0,0   2       92,78,  0,2289,
chr20   3015846 3027843 CUFF.27.1       1000    +       3027843 3027843 0,0,0   9       39,98,120,167,136,150,94,77,137,        0,1969,5462,6207,6842,8625,10840,11274,11860,
chr20   3035578 3038154 CUFF.28.1       1000    +       3038154 3038154 0,0,0   4       132,152,136,96, 0,211,1575,2480,
chr20   3046203 3047821 CUFF.29.1       1000    +       3047821 3047821 0,0,0   4       77,147,124,87,  0,169,410,1531,
chr20   3165062 3165578 CUFF.30.1       1000    -       3165578 3165578 0,0,0   2       90,82,  0,434,
chr20   3166275 3166802 CUFF.31.1       1000    -       3166802 3166802 0,0,0   2       85,98,  0,429,
chr20   3190728 3195930 CUFF.32.3       1000    -       3195930 3195930 0,0,0   6       91,49,57,39,123,80,     0,461,1036,4101,4502,5122,
chr20   3190728 3203535 CUFF.32.2       182     -       3203535 3203535 0,0,0   9       91,49,57,39,123,102,113,204,36, 0,461,1036,4101,4502,9272,9613,12484,12771,
chr20   3190728 3204618 CUFF.32.1       596     -       3204618 3204618 0,0,0   9       91,49,57,39,123,102,113,204,82, 0,461,1036,4101,4502,9272,9613,12484,13808,
chr20   3209521 3223449 CUFF.33.1       1000    +       3223449 3223449 0,0,0   8       96,58,65,74,32,116,77,84,       0,3647,3797,4463,5759,8995,12319,13844,
chr20   3252599 3264459 CUFF.34.1       1000    -       3264459 3264459 0,0,0   7       85,94,66,206,75,107,22, 0,1126,2619,3406,6880,7322,11838,
chr20   3290156 3298112 CUFF.35.1       1000    -       3298112 3298112 0,0,0   5       61,118,102,89,84,       0,3982,4371,6704,7872,
chr20   3362087 3362240 CUFF.36.1       1000    .       3362240 3362240 0,0,0   1       153,    0,
chr20   3560811 3565409 CUFF.37.1       1000    +       3565409 3565409 0,0,0   4       94,184,155,62,  0,1464,2397,4536,
chr20   3572869 3582183 CUFF.38.1       1000    +       3582183 3582183 0,0,0   5       82,122,139,191,49,      0,2957,3989,5712,9265,
chr20   3582272 3584736 CUFF.39.1       1000    +       3584736 3584736 0,0,0   3       82,186,90,      0,1625,2374,
chr20   3591162 3604199 CUFF.40.1       1000    +       3604199 3604199 0,0,0   6       144,94,53,95,79,95,     0,3316,5214,6743,9783,12942,
chr20   3754071 3760171 CUFF.41.1       1000    -       3760171 3760171 0,0,0   5       87,123,139,143,90,      0,325,1397,4464,6010,
chr20   3796698 3802781 CUFF.42.1       1000    +       3802781 3802781 0,0,0   12      33,128,52,42,37,123,123,135,81,177,96,52,       0,923,1713,3589,3763,4044,4272,4555,5023,5225,5582,6031,
chr20   3803133 3804882 CUFF.43.1       1000    +       3804882 3804882 0,0,0   4       73,134,112,62,  0,270,1435,1687,
chr20   3910753 3923311 CUFF.44.1       1000    +       3923311 3923311 0,0,0   5       77,177,124,126,68,      0,1704,6173,7917,12490,
chr20   4023787 4024580 CUFF.45.1       1000    .       4024580 4024580 0,0,0   1       793,    0,
chr20   4059615 4059745 CUFF.46.1       1000    .       4059745 4059745 0,0,0   1       130,    0,
chr20   4059957 4060104 CUFF.47.1       1000    .       4060104 4060104 0,0,0   1       147,    0,
chr20   4182755 4187364 CUFF.48.1       1000    +       4187364 4187364 0,0,0   3       93,161,95,      0,738,4514,
chr20   4629539 4629760 CUFF.49.1       1000    .       4629760 4629760 0,0,0   1       221,    0,
chr20   4629929 4630179 CUFF.50.1       1000    .       4630179 4630179 0,0,0   1       250,    0,
chr20   4784248 4800986 CUFF.51.1       1000    -       4800986 4800986 0,0,0   10      94,98,122,52,102,161,89,152,76,15,      0,1982,3384,3968,5347,6202,8290,11566,13761,16723,
chr20   4857124 4874660 CUFF.52.1       1000    -       4874660 4874660 0,0,0   8       80,96,138,130,106,148,157,85,   0,2164,4823,5653,10645,12781,16811,17451,
chr20   4883737 4885879 CUFF.53.1       1000    -       4885879 4885879 0,0,0   3       86,71,59,       0,1015,2083,
chr20   5106225 5113056 CUFF.54.1       1000    -       5113056 5113056 0,0,0   3       85,114,96,      0,3106,6735,
chr20   5115282 5119661 CUFF.55.1       1000    -       5119661 5119661 0,0,0   6       80,124,195,68,98,84,    0,166,2187,3327,3486,4295,
chr20   5173598 5184943 CUFF.56.1       1000    +       5184943 5184943 0,0,0   7       61,97,98,140,59,83,86,  0,1584,3049,5218,8788,9462,11259,
chr20   5185757 5190166 CUFF.57.1       1000    +       5190166 5190166 0,0,0   5       69,153,120,104,65,      0,929,3309,3977,4344,
chr20   5291163 5292568 CUFF.58.1       1000    .       5292568 5292568 0,0,0   1       1405,   0,
chr20   5292680 5292945 CUFF.59.1       1000    .       5292945 5292945 0,0,0   1       265,    0,
chr20   5558021 5566763 CUFF.60.1       1000    -       5566763 5566763 0,0,0   6       84,136,137,66,62,31,    0,662,1918,3443,6995,8711,
chr20   5567522 5575839 CUFF.61.1       1000    -       5575839 5575839 0,0,0   5       38,93,55,133,24,        0,2624,6392,7890,8293,
chr20   5911585 5916922 CUFF.62.1       1000    +       5916922 5916922 0,0,0   3       97,47,97,       0,4740,5240,
chr20   5923132 5923294 CUFF.63.1       1000    .       5923294 5923294 0,0,0   1       162,    0,
chr20   5923863 5925066 CUFF.64.1       1000    .       5925066 5925066 0,0,0   2       237,95, 0,1108,
```

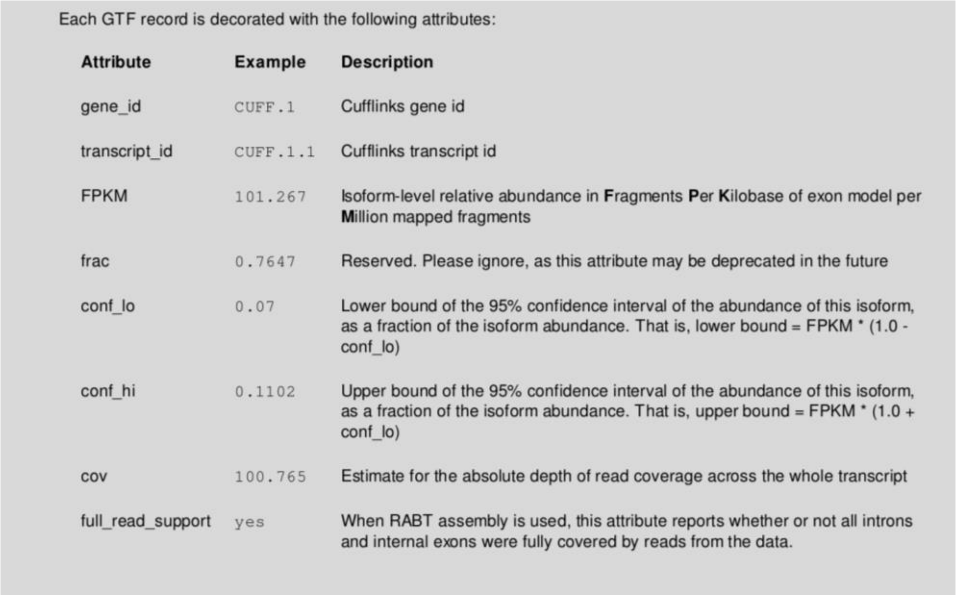
To interpret the output, can be found at cufflink user manual: http://garberlab.umassmed.edu/data/RNASeqCourse/cufflinks.manual.pdf

You can also find the output explanation in the picture below:



This GTF file contains Cufflinks' assembled isoforms. The first 7 columns are standard GTF, and the last column contains attributes, some of which are also standardized ("gene_id", and "transcript_id"). There one GTF record per row, and each record represents either a transcript or an exon within a transcript. The columns are defined as follows:

Column number	Column name	Example	Description
1	seqname	chrX	Chromosome or contig name
2	source	Cufflinks	The name of the program that generated this file (always 'Cufflinks')
3	feature	exon	The type of record (always either "transcript" or "exon".
4	start	77696957	The leftmost coordinate of this record (where 1 is the leftmost possible coordinate)
5	end	77712009	The rightmost coordinate of this record, inclusive.
6	score	77712009	The most abundant isoform for each gene is assigned a score of 1000. Minor isoforms are scored by the ratio (minor FPKM/major FPKM)
7	strand	+	Cufflinks' guess for which strand the isoform came from. Always one of "+", "-", "."
7	frame	.	Cufflinks does not predict where the start and stop codons (if any) are located within each transcript, so this field is not used.



Each GTF record is decorated with the following attributes:

Attribute	Example	Description
gene_id	CUFF.1	Cufflinks gene id
transcript_id	CUFF.1.1	Cufflinks transcript id
FPKM	101.267	Isoform-level relative abundance in **F**ragments **P**er **K**ilobase of exon model per **M**illion mapped fragments
frac	0.7647	Reserved. Please ignore, as this attribute may be deprecated in the future
conf_lo	0.07	Lower bound of the 95% confidence interval of the abundance of this isoform, as a fraction of the isoform abundance. That is, lower bound = FPKM * (1.0 - conf_lo)
conf_hi	0.1102	Upper bound of the 95% confidence interval of the abundance of this isoform, as a fraction of the isoform abundance. That is, upper bound = FPKM * (1.0 + conf_lo)
cov	100.765	Estimate for the absolute depth of read coverage across the whole transcript
full_read_support	yes	When RABT assembly is used, this attribute reports whether or not all introns and internal exons were fully covered by reads from the data.

Useful Links:

- http://garberlab.umassmed.edu/data/RNASeqCourse/cufflinks.manual.pdf
- https://rnaseq.uoregon.edu/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3334321/pdf/nihms-366741.pdf

Tophat

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: Dr.rer.nat Arli Aditya Parikesit

Email: arli.parikesit@i3l.ac.id

Supervisor(s)	Email
David Agustriawan.,PhD	david.agustriawan@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

3. Operate ONLY the computer assigned to you.
   a. If you have any troubleshooting, please contact your supervisor or Building Management
   b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change the system preferences unless directed to do so
   c. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
   d. Do not bring food or drinks into the lab unless it is in your backpack

4. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	4
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

This course session is designed to teach how to use Tophat. Tophat is developed to map reads from RNAseq to a reference sequence and to detect splice junctions. Please go to http://tophat.cbcb.umd.edu/index.html for more information about Tophat. Tophat uses Bowtie2 to map reads to the reference sequence, therefore we have to install Bowtie2 first.

Tophat will focus on exon junctions in the blue signed picture below and bowtie will handle the rest of reads

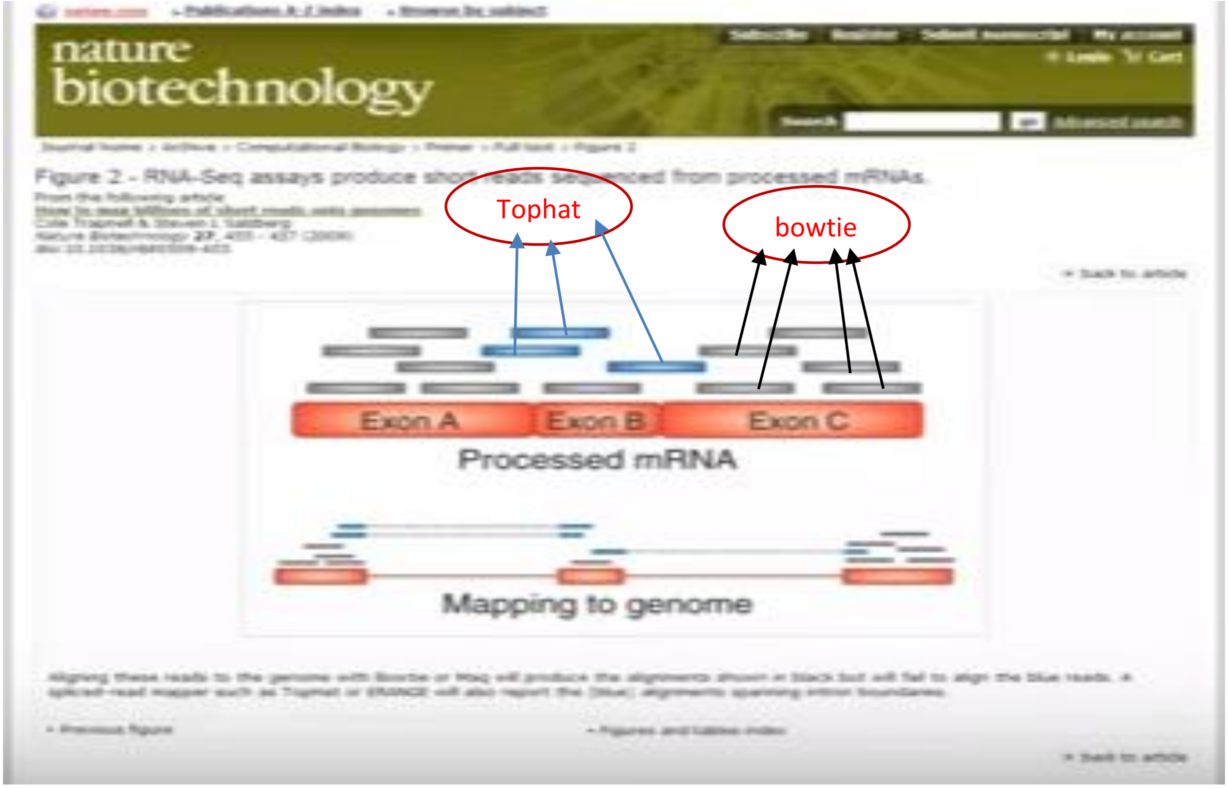


Figure 2 - RNA-Seq assays produce short reads sequenced from processed mRNAs.

From the following article:
How to map billions of short reads onto genomes
Cole Trapnell & Steven L Salzberg
Nature Biotechnology 27, 455 - 457 (2009)
doi:10.1038/nbt0509-455

Aligning these reads to the genome with Bowtie or Maq will produce the alignments shown in black but will fail to align the blue reads. A spliced-read mapper such as TopHat or ERANGE will also report the (blue) alignments spanning intron boundaries.

The main objective of this learning experience are:

- To understand what is the input files (the format, what kind of files needed) for the Tophat
- To understand how to process the data using Tophat
- To understand what is the output format and how to interpret it

Material

7. Software bowtie2
8. Software Tophat
9. FASTQ file
10. Reference genome file

Equipment

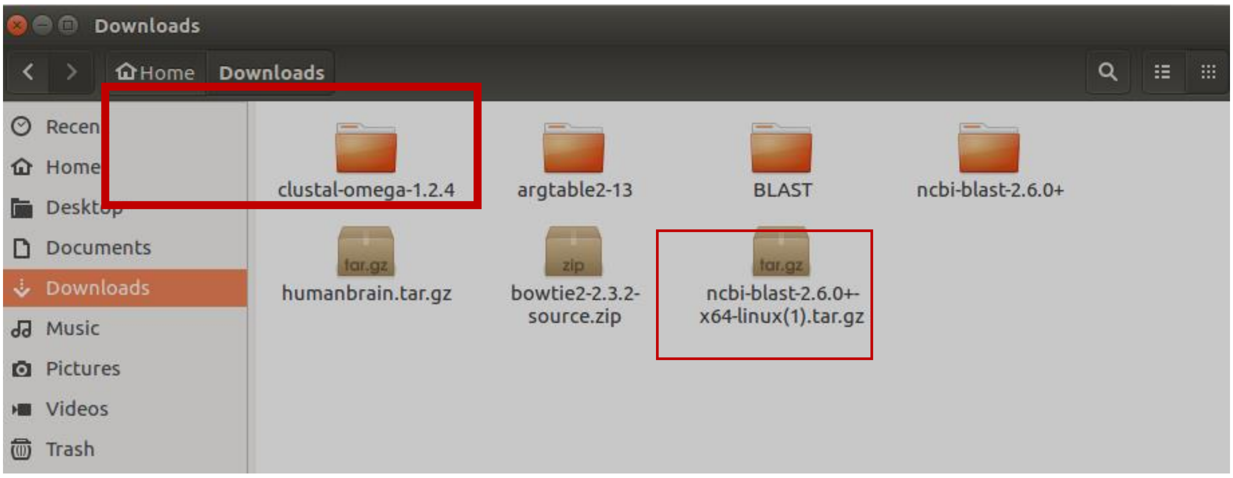
3. Logbook
4. Laptop/PC (available in Bioinformatics laboratory)

Click here to enter text.

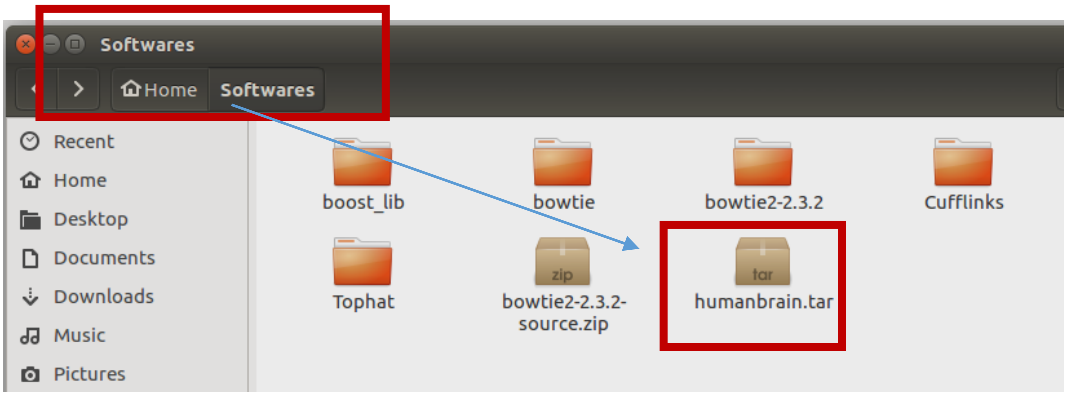
Procedure

1. Download fastq and genome file (chromosome 20) in this link: https://insidedna.me/tool_page_assets/tutorials/tutorial19/humanbrain.tar.gz

   After download it, you can obtain the file in the below path:



2. **Copy or cut** humanbrain.tar.gz folder → into this path: "/home/**i3l-25**/**Software**":

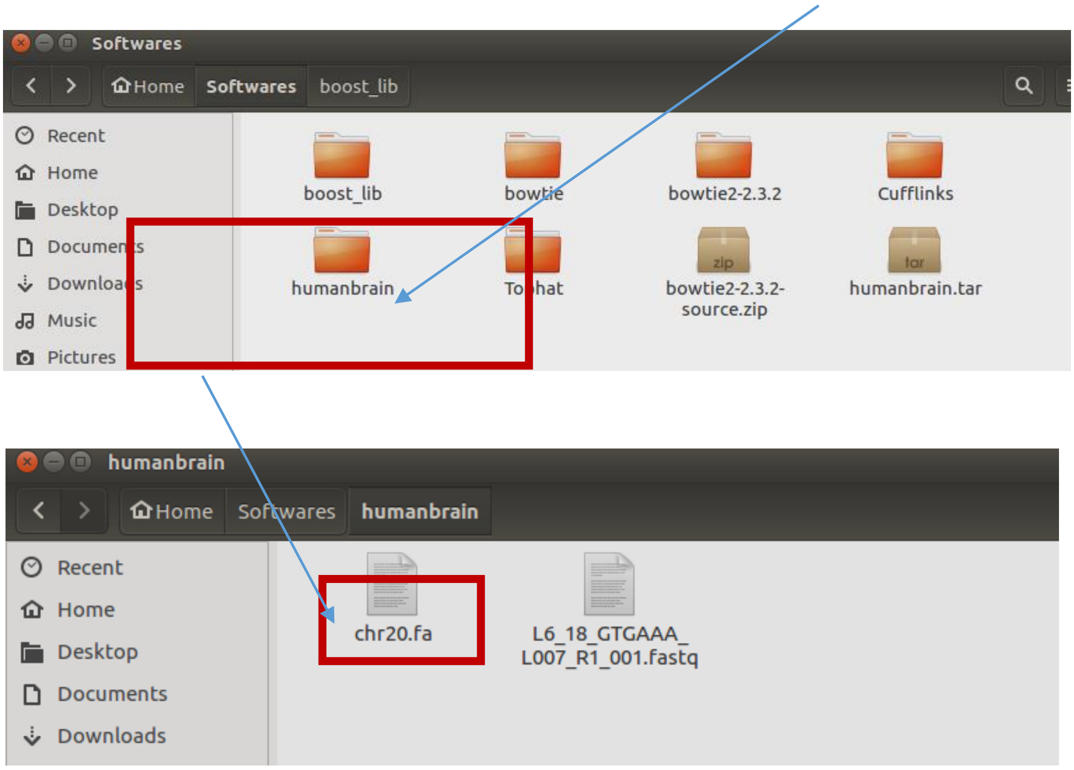


Note:

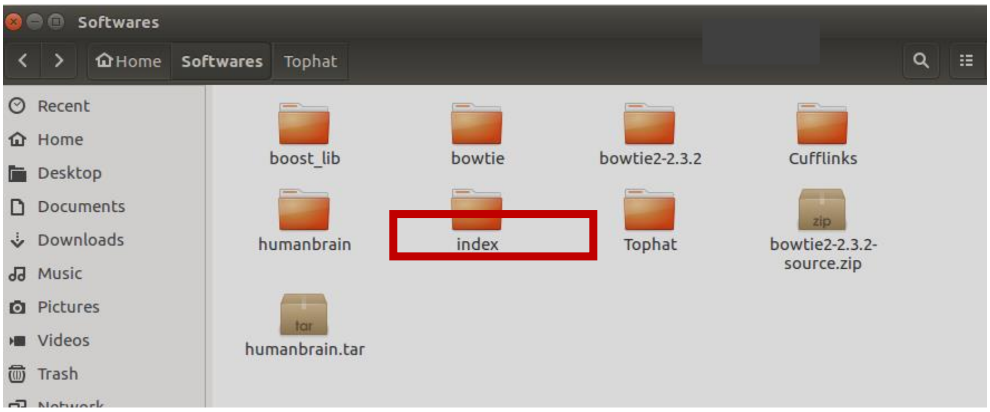
i3l-25: is your computer number. So if your computer number is i3l-26, you need to replace i3l-25 to i3l-26

Software: is a folder under the path "/home/i3l-25"; you need to check in your computer, whether the name is **Software** or **Softwares**

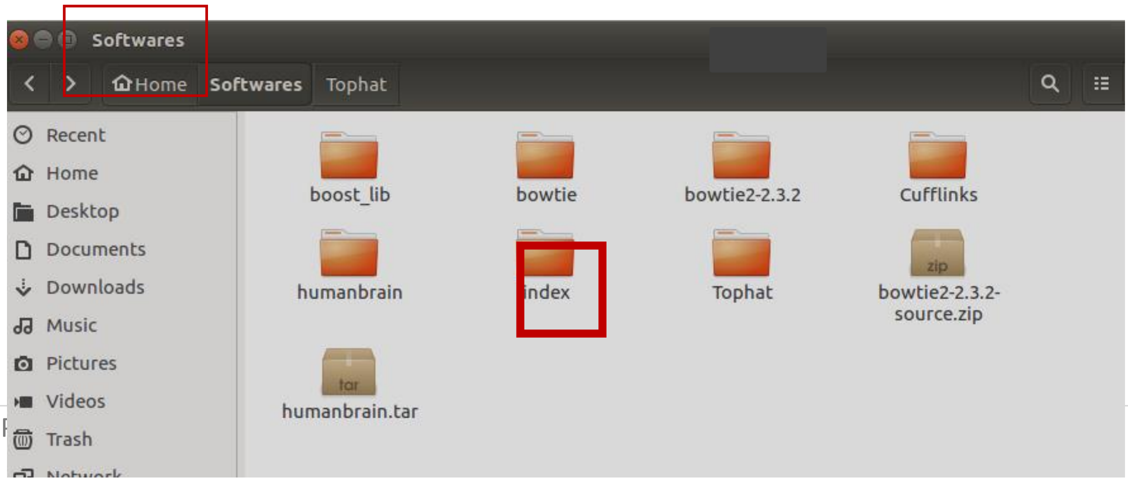
3. **Extract humanbrain.tar folder**, then you will find **humanbrain folder** and inside that folder there are two files: reference genome chromosome 20 (chr20.fa) and fastq file (L6_18_GTGAAA_L007_R1_001.fastq)

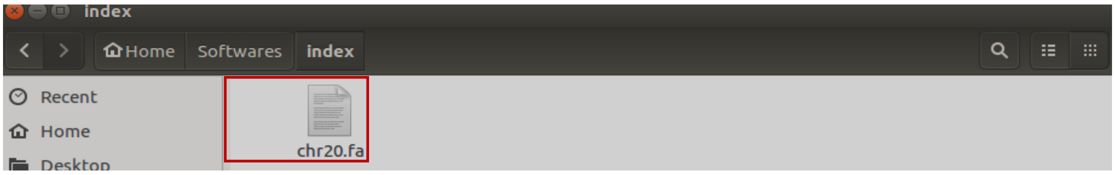


4. Then create a new folder “/home/i31-25/Software/**index**”

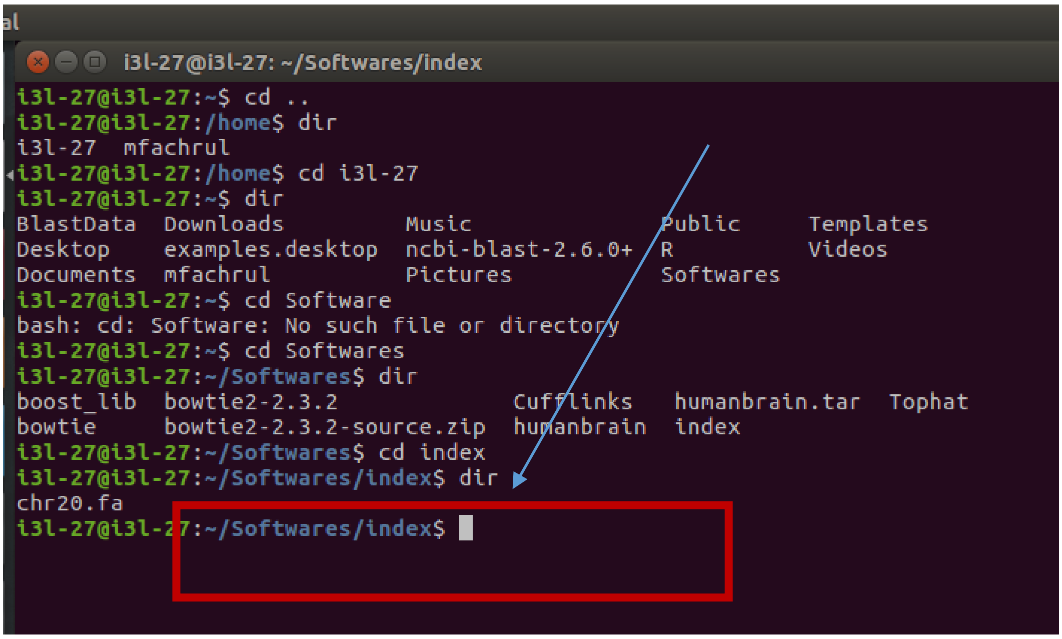


5. **Move (cut)** reference genome chromosome 20 file (chr20.fa) which located inside **humanbrain folder** into the **index folder**



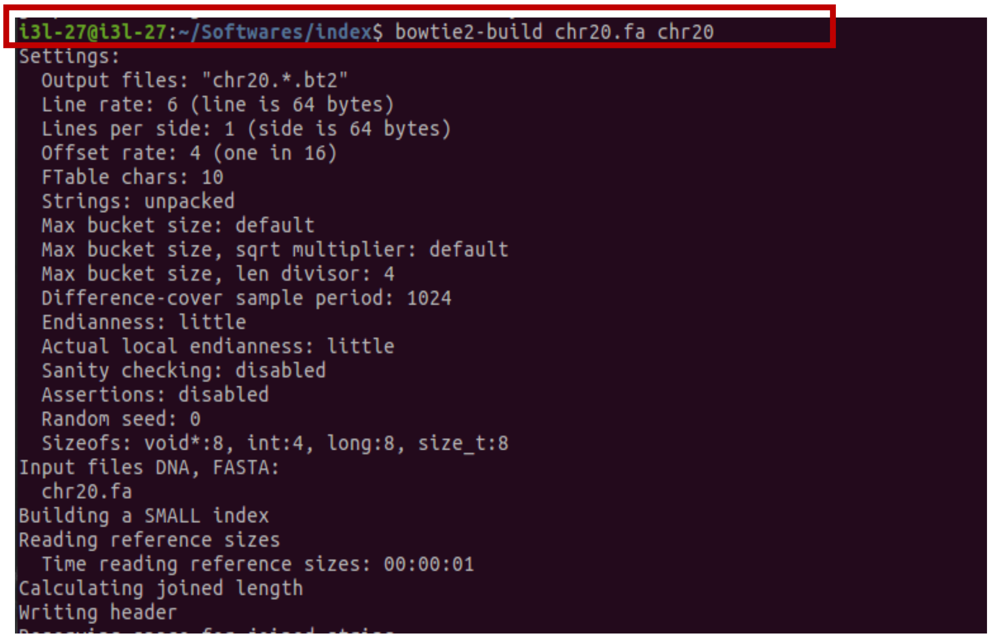


6. Open Linux terminal, go to this path “/home/i3l-25/Software/**index**”.



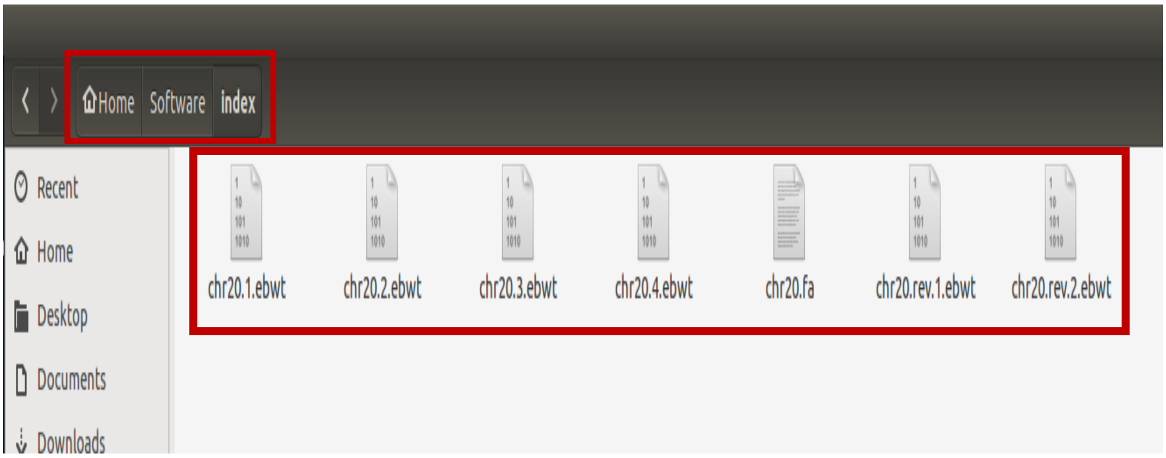
```
i3l-27@i3l-27:~$ cd ..
i3l-27@i3l-27:/home$ dir
i3l-27  mfachrul
i3l-27@i3l-27:/home$ cd i3l-27
i3l-27@i3l-27:~$ dir
BlastData  Downloads         Music             Public     Templates
Desktop    examples.desktop  ncbi-blast-2.6.0+ R          Videos
Documents  mfachrul          Pictures          Softwares
i3l-27@i3l-27:~$ cd Software
bash: cd: Software: No such file or directory
i3l-27@i3l-27:~$ cd Softwares
i3l-27@i3l-27:~/Softwares$ dir
boost_lib  bowtie2-2.3.2             Cufflinks   humanbrain.tar  Tophat
bowtie     bowtie2-2.3.2-source.zip  humanbrain  index
i3l-27@i3l-27:~/Softwares$ cd index
i3l-27@i3l-27:~/Softwares/index$ dir
chr20.fa
i3l-27@i3l-27:~/Softwares/index$
```

7. Then run command for bowtie build indexing. Type: **bowtie2-build chr20.fa chr20**



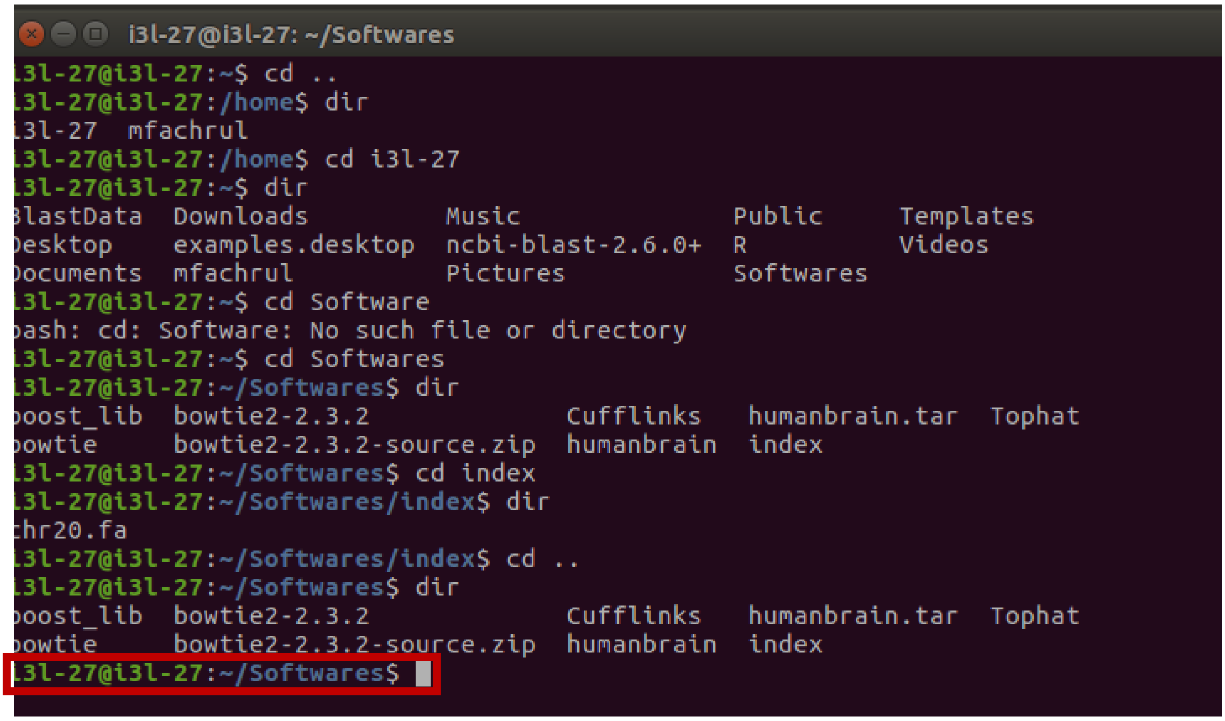
```
i3l-27@i3l-27:~/Softwares/index$ bowtie2-build chr20.fa chr20
Settings:
  Output files: "chr20.*.bt2"
  Line rate: 6 (line is 64 bytes)
  Lines per side: 1 (side is 64 bytes)
  Offset rate: 4 (one in 16)
  FTable chars: 10
  Strings: unpacked
  Max bucket size: default
  Max bucket size, sqrt multiplier: default
  Max bucket size, len divisor: 4
  Difference-cover sample period: 1024
  Endianness: little
  Actual local endianness: little
  Sanity checking: disabled
  Assertions: disabled
  Random seed: 0
  Sizeofs: void*:8, int:4, long:8, size_t:8
Input files DNA, FASTA:
  chr20.fa
Building a SMALL index
Reading reference sizes
  Time reading reference sizes: 00:00:01
Calculating joined length
Writing header
```

Output: as the result, inside the **index folder**, you will find these files below:



In this stage under the path **"/home/i3l-25/Software"**: we already have **bowtie2 index files** in the **index folder**, and fastq file in the **humanbrain folder**.

8. For aligning RNA-seq reads (fastq file) to the reference genome (bowtie2 index files) using Tophat, go to this path "/home/i3l-25/Software"

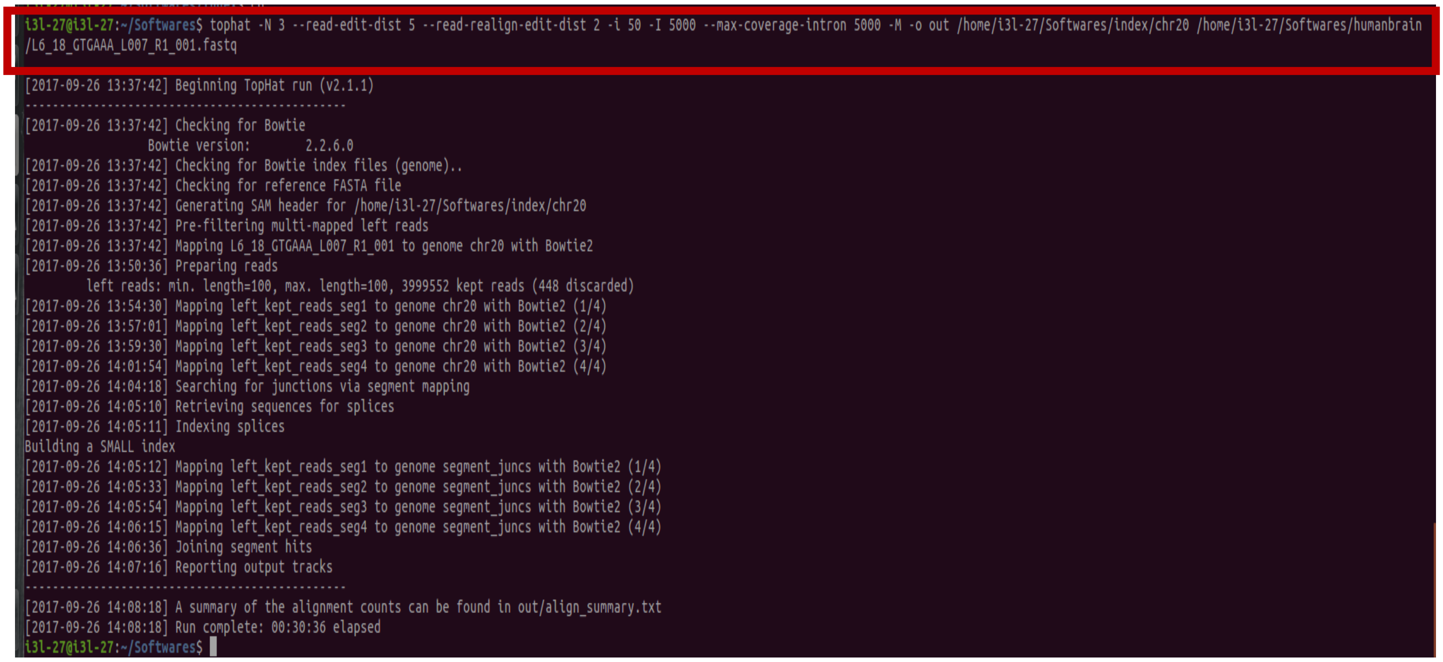


```
i3l-27@i3l-27: ~/Softwares
i3l-27@i3l-27:~$ cd ..
i3l-27@i3l-27:/home$ dir
i3l-27  mfachrul
i3l-27@i3l-27:/home$ cd i3l-27
i3l-27@i3l-27:~$ dir
BlastData  Downloads         Music              Public     Templates
Desktop    examples.desktop  ncbi-blast-2.6.0+  R          Videos
Documents  mfachrul          Pictures           Softwares
i3l-27@i3l-27:~$ cd Software
bash: cd: Software: No such file or directory
i3l-27@i3l-27:~$ cd Softwares
i3l-27@i3l-27:~/Softwares$ dir
boost_lib  bowtie2-2.3.2             Cufflinks   humanbrain.tar  Tophat
bowtie     bowtie2-2.3.2-source.zip  humanbrain  index
i3l-27@i3l-27:~/Softwares$ cd index
i3l-27@i3l-27:~/Softwares/index$ dir
chr20.fa
i3l-27@i3l-27:~/Softwares/index$ cd ..
i3l-27@i3l-27:~/Softwares$ dir
boost_lib  bowtie2-2.3.2             Cufflinks   humanbrain.tar  Tophat
bowtie     bowtie2-2.3.2-source.zip  humanbrain  index
i3l-27@i3l-27:~/Softwares$ 
```

And run this syntax:

tophat -N 3 --read-edit-dist 5 --read-realign-edit-dist 2 -i 50 -I 5000 --max-coverage-intron 5000 -M -o out /home/i3l-25/Software/index/chr20

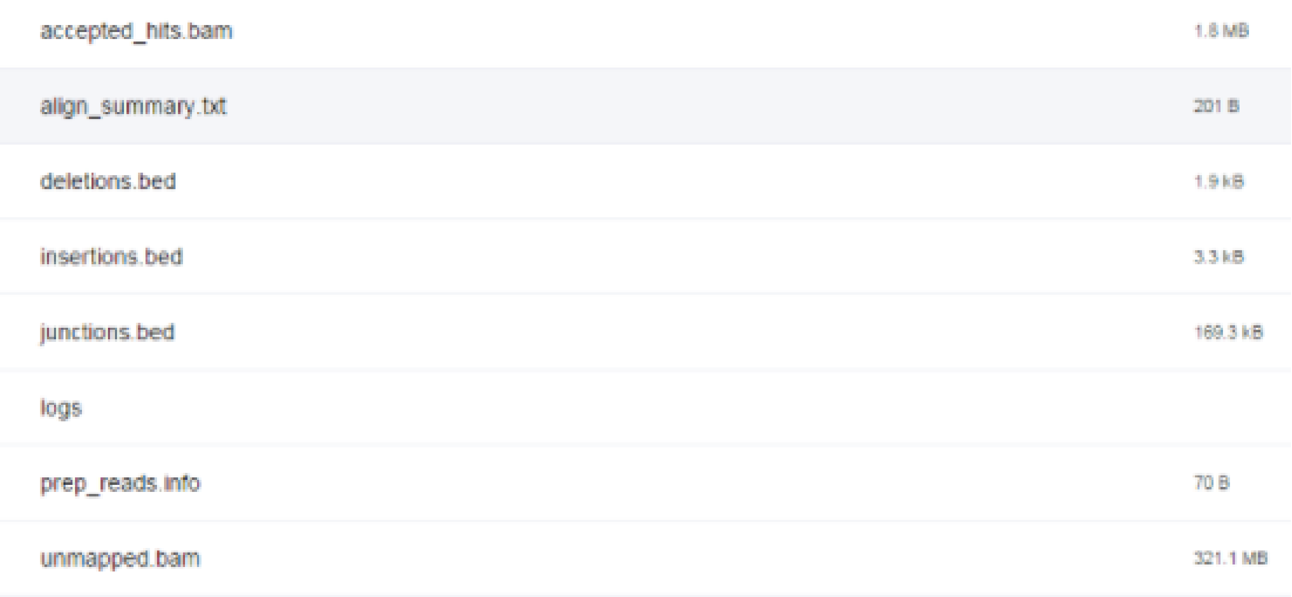
/home/i3l-25/Software/humanbrain/L6_18_GTGAAA_L007_R1_001.fastq



```
i3l-27@i3l-27:~/Softwares$ tophat -N 3 --read-edit-dist 5 --read-realign-edit-dist 2 -i 50 -I 5000 --max-coverage-intron 5000 -M -o out /home/i3l-27/Softwares/index/chr20 /home/i3l-27/Softwares/humanbrain
/L6_18_GTGAAA_L007_R1_001.fastq

[2017-09-26 13:37:42] Beginning TopHat run (v2.1.1)
-----------------------------------------------
[2017-09-26 13:37:42] Checking for Bowtie
                  Bowtie version:        2.2.6.0
[2017-09-26 13:37:42] Checking for Bowtie index files (genome)..
[2017-09-26 13:37:42] Checking for reference FASTA file
[2017-09-26 13:37:42] Generating SAM header for /home/i3l-27/Softwares/index/chr20
[2017-09-26 13:37:42] Pre-filtering multi-mapped left reads
[2017-09-26 13:37:42] Mapping L6_18_GTGAAA_L007_R1_001 to genome chr20 with Bowtie2
[2017-09-26 13:50:36] Preparing reads
         left reads: min. length=100, max. length=100, 3999552 kept reads (448 discarded)
[2017-09-26 13:54:30] Mapping left_kept_reads_seg1 to genome chr20 with Bowtie2 (1/4)
[2017-09-26 13:57:01] Mapping left_kept_reads_seg2 to genome chr20 with Bowtie2 (2/4)
[2017-09-26 13:59:30] Mapping left_kept_reads_seg3 to genome chr20 with Bowtie2 (3/4)
[2017-09-26 14:01:54] Mapping left_kept_reads_seg4 to genome chr20 with Bowtie2 (4/4)
[2017-09-26 14:04:18] Searching for junctions via segment mapping
[2017-09-26 14:05:10] Retrieving sequences for splices
[2017-09-26 14:05:11] Indexing splices
Building a SMALL index
[2017-09-26 14:05:12] Mapping left_kept_reads_seg1 to genome segment_juncs with Bowtie2 (1/4)
[2017-09-26 14:05:33] Mapping left_kept_reads_seg2 to genome segment_juncs with Bowtie2 (2/4)
[2017-09-26 14:05:54] Mapping left_kept_reads_seg3 to genome segment_juncs with Bowtie2 (3/4)
[2017-09-26 14:06:15] Mapping left_kept_reads_seg4 to genome segment_juncs with Bowtie2 (4/4)
[2017-09-26 14:06:36] Joining segment hits
[2017-09-26 14:07:16] Reporting output tracks
-----------------------------------------------
[2017-09-26 14:08:18] A summary of the alignment counts can be found in out/align_summary.txt
[2017-09-26 14:08:18] Run complete: 00:30:36 elapsed
i3l-27@i3l-27:~/Softwares$
```

Then the program will run about 40 minutes or faster. For the output (you can find it at: **“/home/i3l-25/Software/out”**). There will be three useful files:

- **align_summary.txt** with the total number of mapped reads and multi-mapped reads. In our example, we can see that only 0.6% of reads have mapped to the genome. This is not surprising, since the 22nd chromosome contains about 1% of the whole human genome, and the remaining unmapped reads must map to the other chromosomes. Usually, if you use the entire genome as a reference, about 80-90% of all your reads align to the genome, and up to 10-15% of them have multiple alignments.
- ***.bam files** with alignments of reads in special sam format (*.bam is a compressed *.sam file). accepted_hits.bam is the main file that you use for counting expression of the genes. Many tools, such as Cufflinks, can use this file as input to calculate normalized abundances of transcripts for subsequent comparison between samples. To view and manipulate these *.bam files (e.g. sort or merge) you should use samtools tool.
- ***.bed** files with coordinates of introns (junctions.bed) and indels (insertions.bed and deletions.bed).



accepted_hits.bam	1.8 MB
align_summary.txt	201 B
deletions.bed	1.9 kB
insertions.bed	3.3 kB
junctions.bed	169.3 kB
logs	
prep_reads.info	70 B
unmapped.bam	321.1 MB

Interpretation of Tophat syntax:

Since we search introns *de novo*, we specify parameters of intron length:
-i option determines the minimum intron length and
-I option determine the maximum length of introns.

--max-coverage-intron option: sets the maximum intron length that may be found during the coverage search. In our example, we map reads without annotation or specified junctions.

–**N** option: means that the final read alignments that have more than 3 mismatches are discarded.

--read-edit-dist option: shows the minimum edit distance for accepted reads. ‘Edit distance’ is the main metric for alignment quality. It measures the minimum number of operations required to transform one string into another. More specifically, for a sequence alignment, edit distance is defined as the total number of mismatched, inserted or deleted bases in the reference

--read-realign-edit-dist option**:** which directs TopHat to re-align reads for which the edit distance of an alignment obtained in a previous mapping step is above or equal to this option value. If you set this option to **0**, TopHat maps every read in all the mapping steps, reporting the best possible alignment found in any of these mapping steps. It may greatly increase the mapping accuracy, at the expense of an increase in running time. The default value for this option is set such that TopHat does not try to realign reads already mapped in earlier steps.

Finally, –**M** option tells TopHat that we are mapping reads to a whole genome, and thus we wish to exclude multi-mapped reads.

-o out option: means there will be a folder “out” to save all the mapping results output.

Useful Links:

- https://insidedna.me/tutorials/view/tophat2-analysis-of-rna-expression-is
- bowtie build: http://ged.msu.edu/angus/tutorials/bowtie-mapping.html
- run tophat: http://ged.msu.edu/angus/tutorials-2011/mrnaseq-tophat-mapping.html

Erratum for this Transcriptomics Module:

For the reference genome and sample file, kindly use these links:

Reference Genome:

ftp://ftp.ncbi.nlm.nih.gov/genomes/Homo_sapiens/109.20190905/GCF_000001405.39_GRCh38.p13/

ftp://ftp.ncbi.nlm.nih.gov/genomes/Homo_sapiens/109.20190905/GCF_000001405.39_GRCh38.p13/GCF_000001405.39_GRCh38.p13_genomic.fna.gz

Sample file : ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/data/NA19308/sequence_read/

ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/data/NA19308/sequence_read/SRR014948.recal.fastq.gz

Download the file with .gz extension, and uncompress it with the standard linux tools of tar as following:

```
$tar -xzvf file.tar.gz
```

Indonesia International Institute for Life Science

Transcriptomics

COURSE CODE: BI021

RNASeq in R

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: Dr.rer.nat Arli Aditya Parikesit

Email: arli.parikesit@i3l.ac.id

Supervisor(s)	Email
David Agustriawan.,PhD	david.agustriawan@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

1. Operate ONLY the computer assigned to you.
    a. If you have any troubleshooting, please contact your supervisor or Building Management
    b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change the system preferences unless directed to do so
    c. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
    d. Do not bring food or drinks into the lab unless it is in your backpack

2. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	5
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

Measuring gene expression on a genome-wide scale has become common practice over the last two decades or so, with microarrays predominantly used pre-2008. With the advent of next generation sequencing technology in 2008, an increasing number of scientists use this technology to measure and understand changes in gene expression in often complex systems. As sequencing costs have decreased, using RNA-Seq to simultaneously measure the expression of tens of thousands of genes for multiple samples has never been easier. The cost of these experiments has now moved from generating the data to storing and analyzing it.

There are many steps involved in analyzing an RNA-Seq experiment. Analyzing an RNAseq experiment begins with sequencing reads. These are aligned to a reference genome, then the number of reads mapped to each gene can be counted. This results in a table of counts, which is what we perform statistical analyses on in R. While mapping and counting are important and necessary tasks, today we will be starting from the **count data** and getting stuck into analysis.

Mouse mammary gland dataset

The data for this tutorial comes from a Nature Cell Biology paper, *EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival* (Fu et al. 2015). Both the raw data (sequence reads) and processed data (counts) can be downloaded from Gene Expression Omnibus database (GEO) under accession number GSE60450.

This study examines the expression profiles of basal stem-cell enriched cells (B) and committed luminal cells (L) in the mammary gland of virgin, pregnant and lactating mice. Six groups are present, with one for each combination of cell type and mouse status. Each group contains two biological replicates. We will first use the counts file as a starting point for our analysis. This data has already been aligned to the mouse genome. The command line tool featureCounts (Liao, Smyth, and Shi 2014) was used to count reads mapped to mouse genes from Refseq annotation (see the paper for details).

The main objective of this learning experience are:

- Reading in the data
- Format the data
- Filtering to remove lowly expressed genes
- Plot the data

Material

11. Rstudio

Equipment

5. Logbook
6. Laptop/PC (available in Bioinformatics laboratory)

Click here to enter text.

Procedure

6. Data files are available from: https://figshare.com/s/1d788fd384d33e913a2a You should download these files and place them in your `/data` directory.

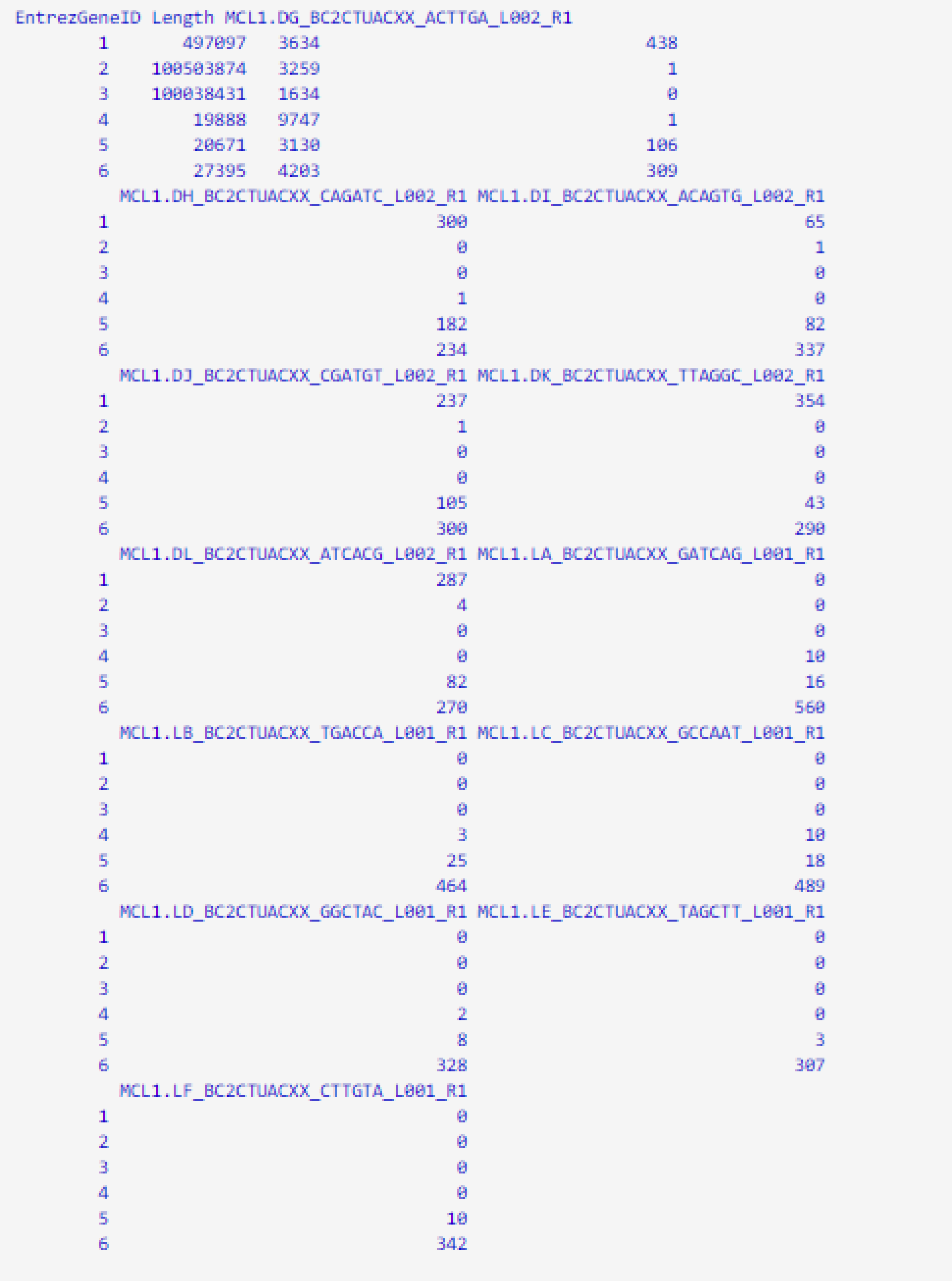
Data files:
sampleinfo.txt SampleInfo_Corrected.txt GSE60450_Lactation-GenewiseCounts.txt
mouse_c2_v5.rdata
mouse_H_v5.rdata ResultsTable_small.txt small_counts.txt

7. *Set up an RStudio project specifying the directory where you have saved the `/data` directory*. Download and read in the data.

```
# Read the data into R
          seqdata <- read.delim("data/GSE60450_Lactation-GenewiseCounts.txt", strings
AsFactors = FALSE)
          # Read the sample information into R
          sampleinfo <- read.delim("data/SampleInfo.txt")
```

Let’s take a look at the data. You can use the `head` command to see the first 6 lines. The `dim` command will tell you how many rows and columns the data frame has

```
head(seqdata)
```



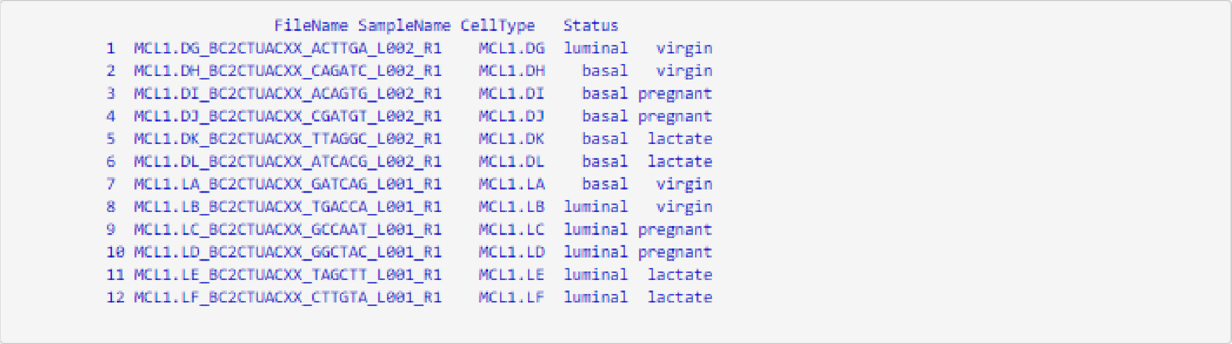
```
  EntrezGeneID Length MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1
1       497097   3634                               438
2    100503874   3259                                 1
3    100038431   1634                                 0
4        19888   9747                                 1
5        20671   3130                               106
6        27395   4203                               309
  MCL1.DH_BC2CTUACXX_CAGATC_L002_R1 MCL1.DI_BC2CTUACXX_ACAGTG_L002_R1
1                               300                                65
2                                 0                                 1
3                                 0                                 0
4                                 1                                 0
5                               182                                82
6                               234                               337
  MCL1.DJ_BC2CTUACXX_CGATGT_L002_R1 MCL1.DK_BC2CTUACXX_TTAGGC_L002_R1
1                               237                               354
2                                 1                                 0
3                                 0                                 0
4                                 0                                 0
5                               105                                43
6                               300                               290
  MCL1.DL_BC2CTUACXX_ATCACG_L002_R1 MCL1.LA_BC2CTUACXX_GATCAG_L001_R1
1                               287                                 0
2                                 4                                 0
3                                 0                                 0
4                                 0                                10
5                                82                                16
6                               270                               560
  MCL1.LB_BC2CTUACXX_TGACCA_L001_R1 MCL1.LC_BC2CTUACXX_GCCAAT_L001_R1
1                                 0                                 0
2                                 0                                 0
3                                 0                                 0
4                                 3                                10
5                                25                                18
6                               464                               489
  MCL1.LD_BC2CTUACXX_GGCTAC_L001_R1 MCL1.LE_BC2CTUACXX_TAGCTT_L001_R1
1                                 0                                 0
2                                 0                                 0
3                                 0                                 0
4                                 2                                 0
5                                 8                                 3
6                               328                               307
  MCL1.LF_BC2CTUACXX_CTTGTA_L001_R1
1                                 0
2                                 0
3                                 0
4                                 0
5                                10
6                               342
```

```
dim(seqdata)
```

```
[1] 27179    14
```

The seqdata object contains information about genes (one gene per row), the first column has the Entrez gene id, the second has the gene length and the remaining columns contain information about the number of reads aligning to the gene in each experimental sample. There are two replicates for each cell type and time point (detailed sample info can be found in file "GSE60450_series_matrix.txt" from the GEO website). The sample info file contains basic information about the samples that we will need for the analysis today.

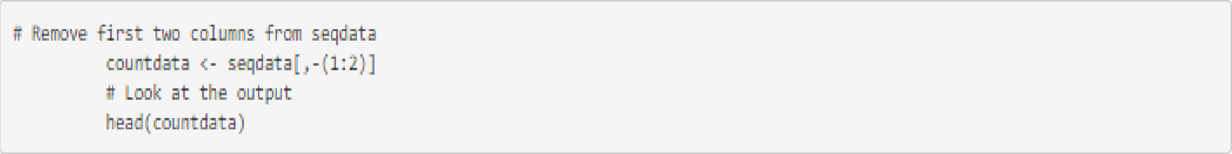
```
sampleinfo
```



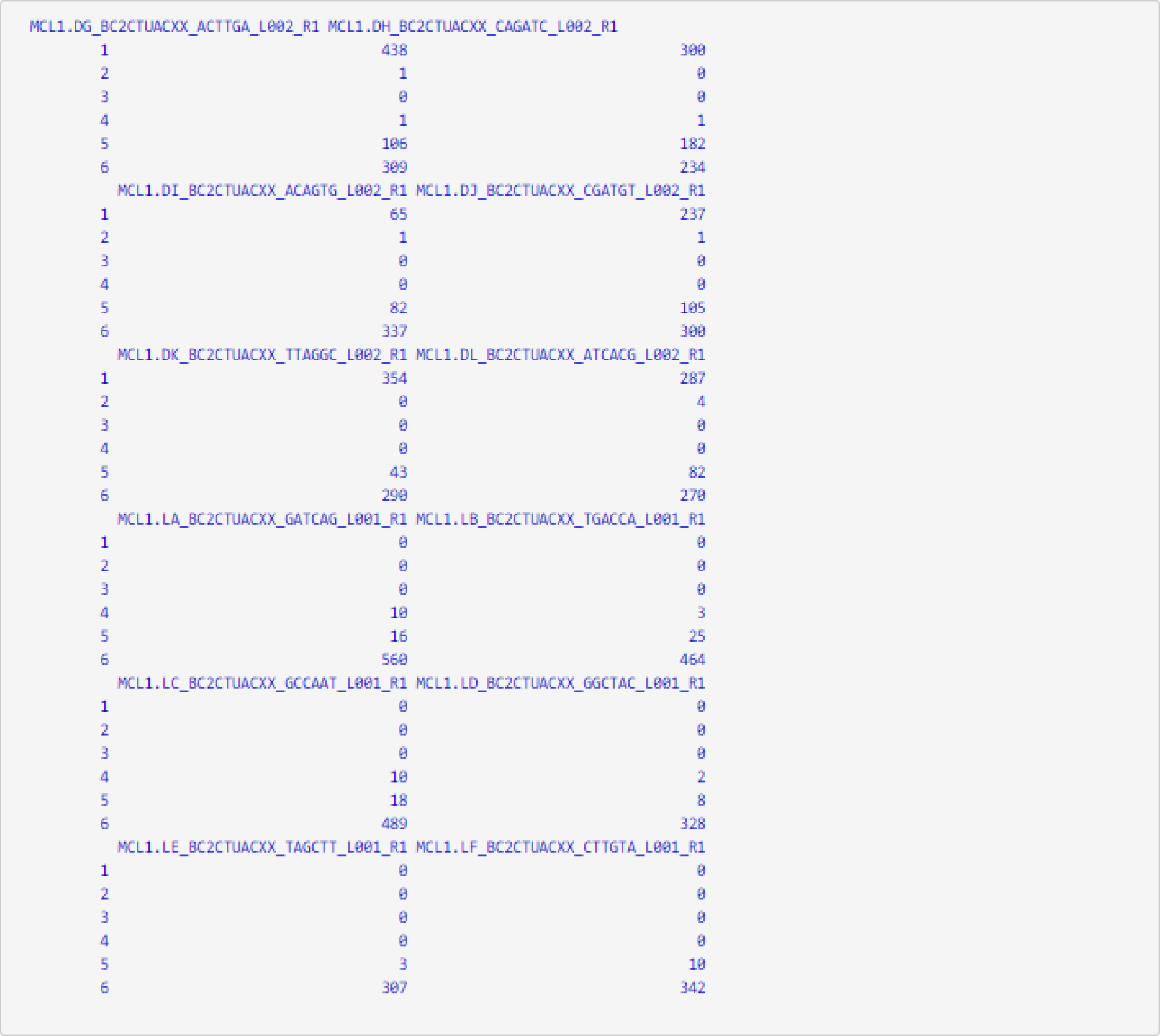
```
                    FileName SampleName CellType   Status
1  MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1    MCL1.DG luminal   virgin
2  MCL1.DH_BC2CTUACXX_CAGATC_L002_R1    MCL1.DH   basal   virgin
3  MCL1.DI_BC2CTUACXX_ACAGTG_L002_R1    MCL1.DI   basal pregnant
4  MCL1.DJ_BC2CTUACXX_CGATGT_L002_R1    MCL1.DJ   basal pregnant
5  MCL1.DK_BC2CTUACXX_TTAGGC_L002_R1    MCL1.DK   basal  lactate
6  MCL1.DL_BC2CTUACXX_ATCACG_L002_R1    MCL1.DL   basal  lactate
7  MCL1.LA_BC2CTUACXX_GATCAG_L001_R1    MCL1.LA   basal   virgin
8  MCL1.LB_BC2CTUACXX_TGACCA_L001_R1    MCL1.LB luminal   virgin
9  MCL1.LC_BC2CTUACXX_GCCAAT_L001_R1    MCL1.LC luminal pregnant
10 MCL1.LD_BC2CTUACXX_GGCTAC_L001_R1    MCL1.LD luminal pregnant
11 MCL1.LE_BC2CTUACXX_TAGCTT_L001_R1    MCL1.LE luminal  lactate
12 MCL1.LF_BC2CTUACXX_CTTGTA_L001_R1    MCL1.LF luminal  lactate
```

We will be manipulating and reformating the counts matrix into a suitable format for downstream analysis. The first two columns in the seqdata dataframe contain annotation information. We need to make a new matrix containing only the counts, but we can store the gene identifiers (the EntrezGeneID column) as rownames.

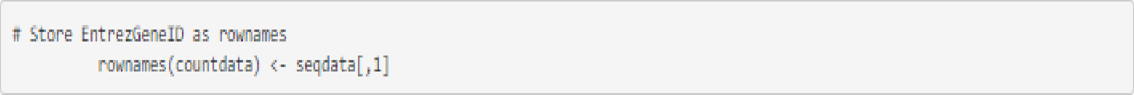
8. Let's create a new data object, countdata, that contains only the counts for the 12 samples.



```
# Remove first two columns from seqdata
        countdata <- seqdata[,-(1:2)]
        # Look at the output
        head(countdata)
```

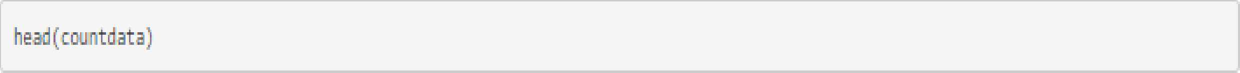


```
MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1 MCL1.DH_BC2CTUACXX_CAGATC_L002_R1
        1                               438                               300
        2                                 1                                 0
        3                                 0                                 0
        4                                 1                                 1
        5                               106                               182
        6                               309                               234
          MCL1.DI_BC2CTUACXX_ACAGTG_L002_R1 MCL1.DJ_BC2CTUACXX_CGATGT_L002_R1
        1                                65                               237
        2                                 1                                 1
        3                                 0                                 0
        4                                 0                                 0
        5                                82                               105
        6                               337                               300
          MCL1.DK_BC2CTUACXX_TTAGGC_L002_R1 MCL1.DL_BC2CTUACXX_ATCACG_L002_R1
        1                               354                               287
        2                                 0                                 4
        3                                 0                                 0
        4                                 0                                 0
        5                                43                                82
        6                               290                               270
          MCL1.LA_BC2CTUACXX_GATCAG_L001_R1 MCL1.LB_BC2CTUACXX_TGACCA_L001_R1
        1                                 0                                 0
        2                                 0                                 0
        3                                 0                                 0
        4                                10                                 3
        5                                16                                25
        6                               560                               464
          MCL1.LC_BC2CTUACXX_GCCAAT_L001_R1 MCL1.LD_BC2CTUACXX_GGCTAC_L001_R1
        1                                 0                                 0
        2                                 0                                 0
        3                                 0                                 0
        4                                10                                 2
        5                                18                                 8
        6                               489                               328
          MCL1.LE_BC2CTUACXX_TAGCTT_L001_R1 MCL1.LF_BC2CTUACXX_CTTGTA_L001_R1
        1                                 0                                 0
        2                                 0                                 0
        3                                 0                                 0
        4                                 0                                 0
        5                                 3                                10
        6                               307                               342
```

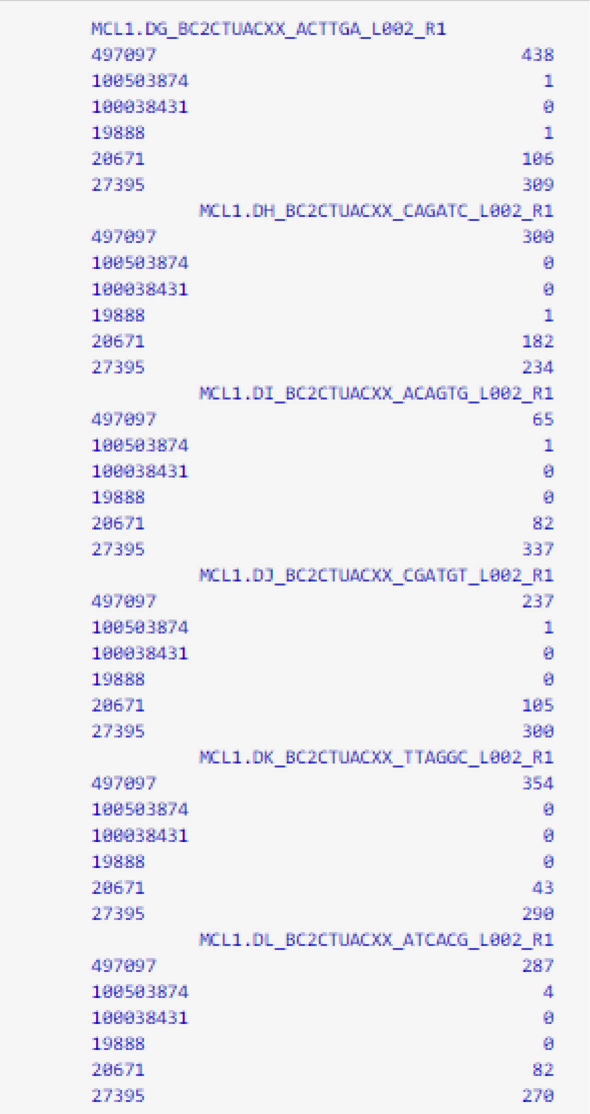


```
# Store EntrezGeneID as rownames
         rownames(countdata) <- seqdata[,1]
```

Take a look at the output

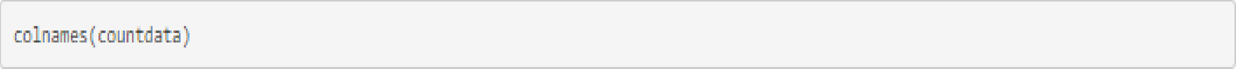


```
head(countdata)
```

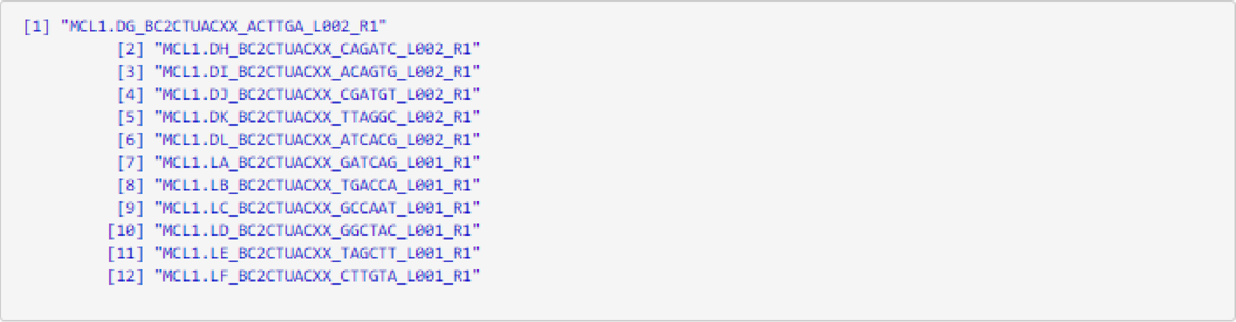


```
MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1
497097                                  438
100503874                                 1
100038431                                 0
19888                                     1
20671                                   106
27395                                   309
          MCL1.DH_BC2CTUACXX_CAGATC_L002_R1
497097                                  300
100503874                                 0
100038431                                 0
19888                                     1
20671                                   182
27395                                   234
          MCL1.DI_BC2CTUACXX_ACAGTG_L002_R1
497097                                   65
100503874                                 1
100038431                                 0
19888                                     0
20671                                    82
27395                                   337
          MCL1.DJ_BC2CTUACXX_CGATGT_L002_R1
497097                                  237
100503874                                 1
100038431                                 0
19888                                     0
20671                                   105
27395                                   300
          MCL1.DK_BC2CTUACXX_TTAGGC_L002_R1
497097                                  354
100503874                                 0
100038431                                 0
19888                                     0
20671                                    43
27395                                   290
          MCL1.DL_BC2CTUACXX_ATCACG_L002_R1
497097                                  287
100503874                                 4
100038431                                 0
19888                                     0
20671                                    82
27395                                   270
```

Now take a look at the column names

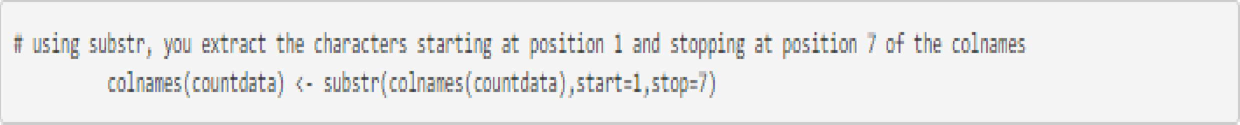


```
colnames(countdata)
```



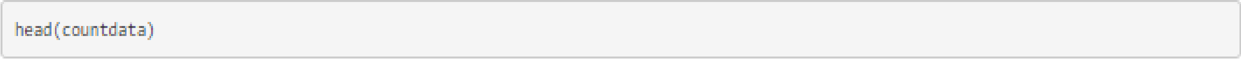
```
[1] "MCL1.DG_BC2CTUACXX_ACTTGA_L002_R1"
          [2] "MCL1.DH_BC2CTUACXX_CAGATC_L002_R1"
          [3] "MCL1.DI_BC2CTUACXX_ACAGTG_L002_R1"
          [4] "MCL1.DJ_BC2CTUACXX_CGATGT_L002_R1"
          [5] "MCL1.DK_BC2CTUACXX_TTAGGC_L002_R1"
          [6] "MCL1.DL_BC2CTUACXX_ATCACG_L002_R1"
          [7] "MCL1.LA_BC2CTUACXX_GATCAG_L001_R1"
          [8] "MCL1.LB_BC2CTUACXX_TGACCA_L001_R1"
          [9] "MCL1.LC_BC2CTUACXX_GCCAAT_L001_R1"
         [10] "MCL1.LD_BC2CTUACXX_GGCTAC_L001_R1"
         [11] "MCL1.LE_BC2CTUACXX_TAGCTT_L001_R1"
         [12] "MCL1.LF_BC2CTUACXX_CTTGTA_L001_R1"
```

These are the sample names which are pretty long so we’ll shorten these to contain only the relevant information about each sample. We will use the `substr` command to extract the first 7 characters and use these as the colnames.

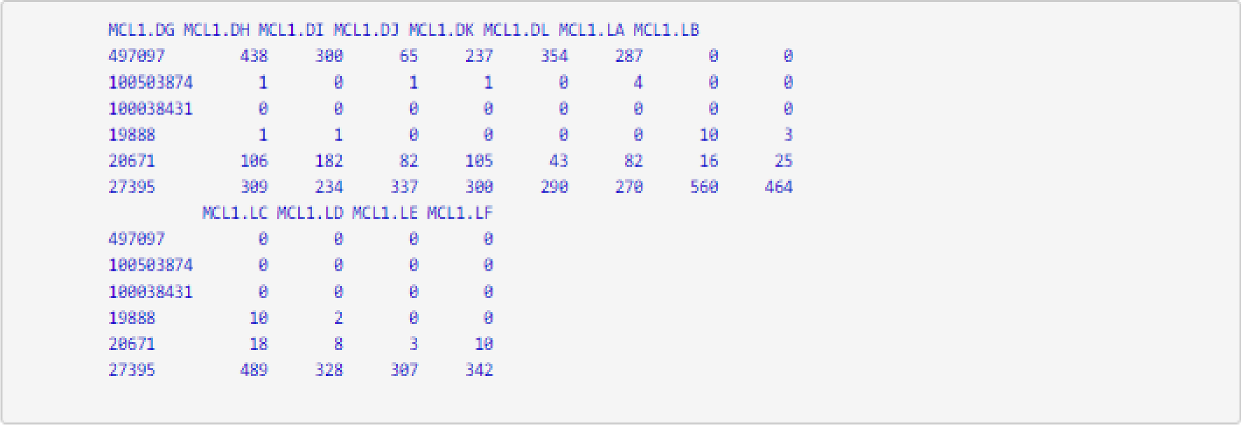


```
# using substr, you extract the characters starting at position 1 and stopping at position 7 of the colnames
        colnames(countdata) <- substr(colnames(countdata),start=1,stop=7)
```

Take a look at the output

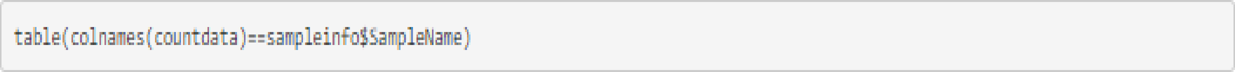


```
head(countdata)
```

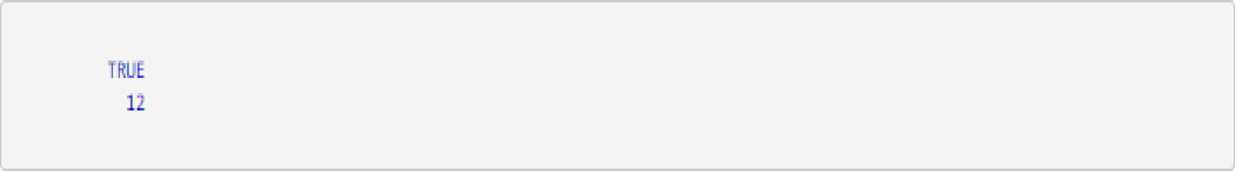


```
MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB
497097         438     300      65     237     354     287       0       0
100503874        1       0       1       1       0       4       0       0
100038431        0       0       0       0       0       0       0       0
19888            1       1       0       0       0       0      10       3
20671          106     182      82     105      43      82      16      25
27395          309     234     337     300     290     270     560     464
          MCL1.LC MCL1.LD MCL1.LE MCL1.LF
497097           0       0       0       0
100503874        0       0       0       0
100038431        0       0       0       0
19888           10       2       0       0
20671           18       8       3      10
27395          489     328     307     342
```

Note that the column names are now the same as SampleName in the sampleinfo file. This is good because it means our sample information in sampleinfo is in the same order as the columns in countdata.



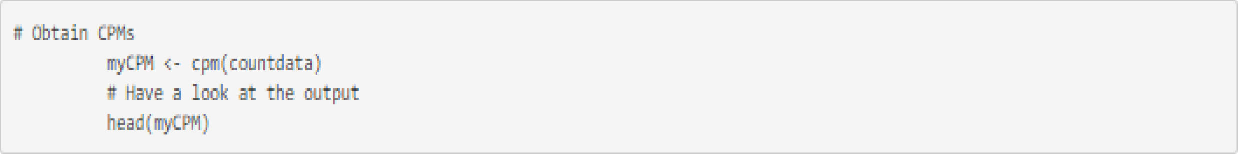
```
table(colnames(countdata)==sampleinfo$SampleName)
```



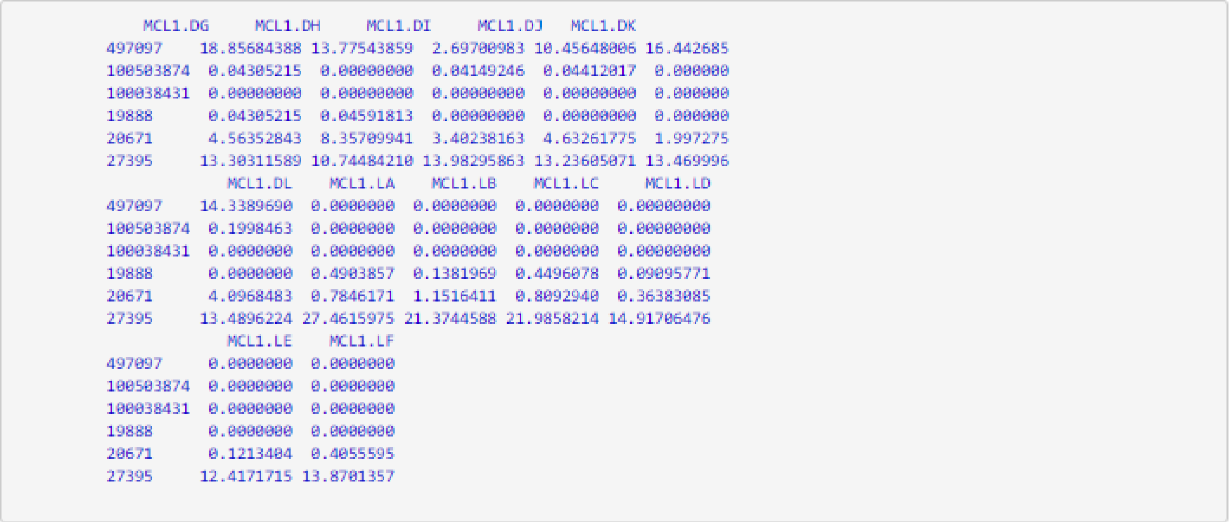
```
TRUE
  12
```

9. Genes with very low counts across all libraries provide little evidence for differential expression and they interfere with some of the statistical approximations that are used later in the pipeline. They also add to the multiple testing burden when estimating false discovery rates, reducing power to detect differentially expressed genes. These genes should be filtered out prior to further analysis.
   There are a few ways to filter out lowly expressed genes. When there are biological replicates in each group, in this case we have a sample size of 2 in each group, we favour filtering on a minimum counts per million threshold present in at least 2 samples. Two represents the smallest sample size for each group in our experiment. In this dataset, we choose to retain genes if they are expressed at a counts-per-million (CPM) above 0.5 in at least two samples.

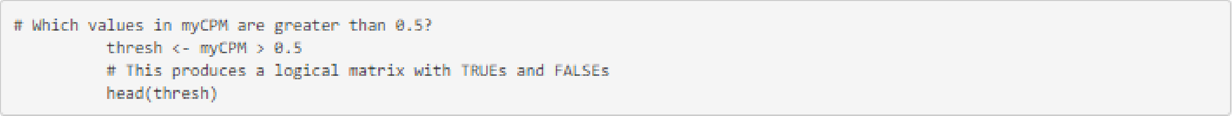
We’ll use the cpm function from the *edgeR* library (M D Robinson, McCarthy, and Smyth 2010) to generate the CPM values and then filter. Note that by converting to CPMs we are normalising for the different sequencing depths for each sample.



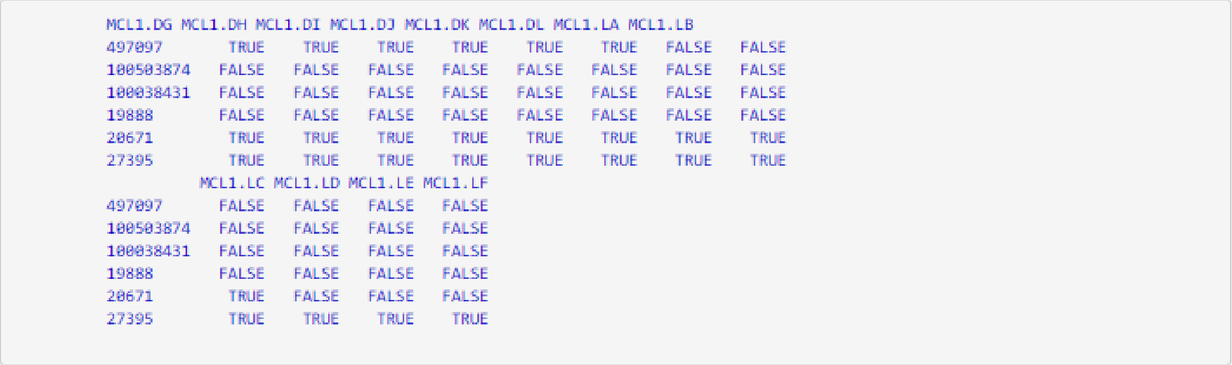
```
# Obtain CPMs
        myCPM <- cpm(countdata)
        # Have a look at the output
        head(myCPM)
```



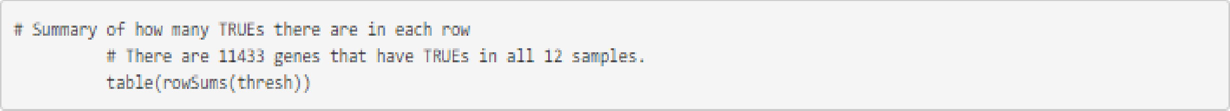
```
             MCL1.DG     MCL1.DH     MCL1.DI     MCL1.DJ   MCL1.DK
497097    18.85684388 13.77543859  2.69700983 10.45648006 16.442685
100503874  0.04305215  0.00000000  0.04149246  0.04412017  0.000000
100038431  0.00000000  0.00000000  0.00000000  0.00000000  0.000000
19888      0.04305215  0.04591813  0.00000000  0.00000000  0.000000
20671      4.56352843  8.35709941  3.40238163  4.63261775  1.997275
27395     13.30311589 10.74484210 13.98295863 13.23605071 13.469996
            MCL1.DL    MCL1.LA    MCL1.LB    MCL1.LC     MCL1.LD
497097    14.3389690  0.0000000  0.0000000  0.0000000  0.00000000
100503874  0.1998463  0.0000000  0.0000000  0.0000000  0.00000000
100038431  0.0000000  0.0000000  0.0000000  0.0000000  0.00000000
19888      0.0000000  0.4903857  0.1381969  0.4496078  0.09095771
20671      4.0968483  0.7846171  1.1516411  0.8092940  0.36383085
27395     13.4896224 27.4615975 21.3744588 21.9858214 14.91706476
            MCL1.LE    MCL1.LF
497097     0.0000000  0.0000000
100503874  0.0000000  0.0000000
100038431  0.0000000  0.0000000
19888      0.0000000  0.0000000
20671      0.1213404  0.4055595
27395     12.4171715 13.8701357
```



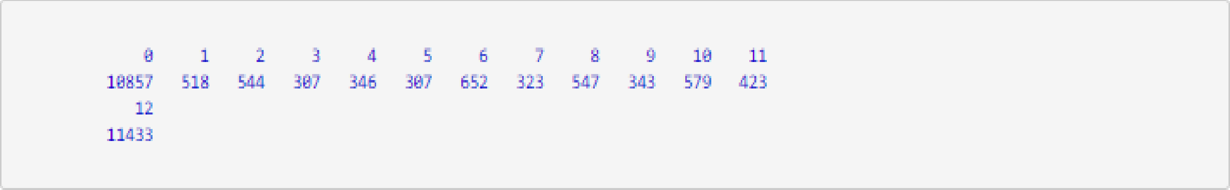
```
# Which values in myCPM are greater than 0.5?
        thresh <- myCPM > 0.5
        # This produces a logical matrix with TRUEs and FALSEs
        head(thresh)
```



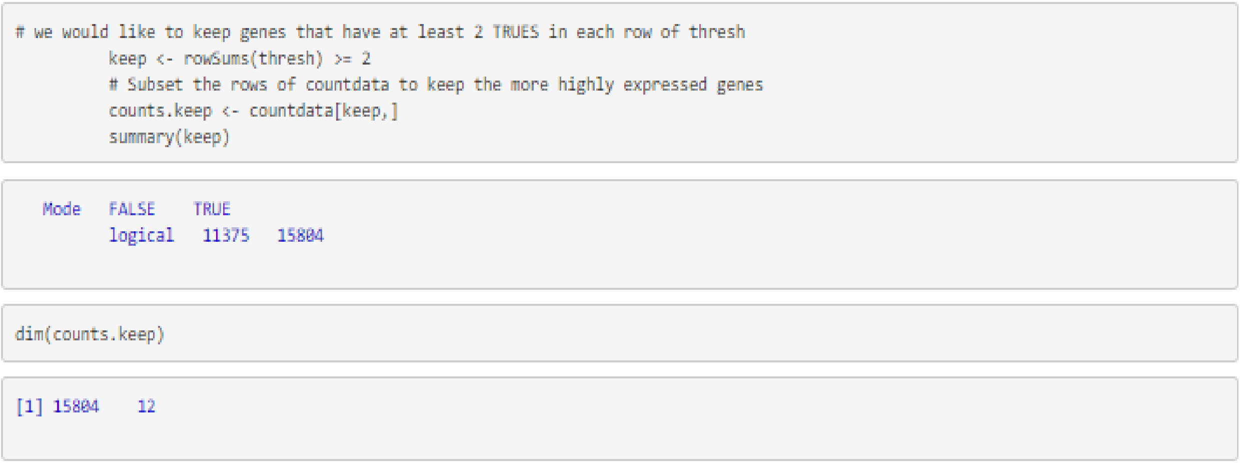
```
MCL1.DG MCL1.DH MCL1.DI MCL1.DJ MCL1.DK MCL1.DL MCL1.LA MCL1.LB
497097       TRUE    TRUE    TRUE    TRUE    TRUE    TRUE   FALSE   FALSE
100503874   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
100038431   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
19888       FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
20671        TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE
27395        TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE    TRUE
            MCL1.LC MCL1.LD MCL1.LE MCL1.LF
497097      FALSE   FALSE   FALSE   FALSE
100503874   FALSE   FALSE   FALSE   FALSE
100038431   FALSE   FALSE   FALSE   FALSE
19888       FALSE   FALSE   FALSE   FALSE
20671        TRUE   FALSE   FALSE   FALSE
27395        TRUE    TRUE    TRUE    TRUE
```



```
# Summary of how many TRUEs there are in each row
        # There are 11433 genes that have TRUEs in all 12 samples.
        table(rowSums(thresh))
```



```
    0     1     2     3     4     5     6     7     8     9    10    11 
10857   518   544   307   346   307   652   323   547   343   579   423 
   12 
11433 
```



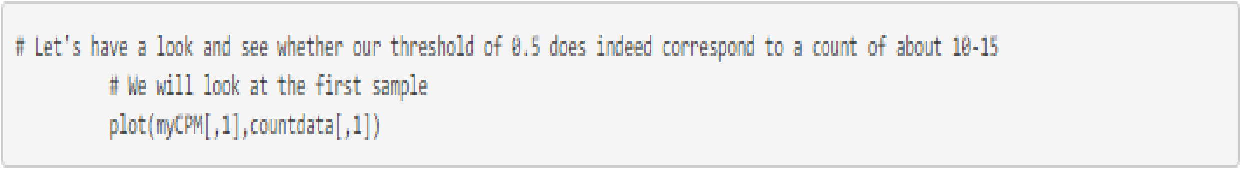
```
# we would like to keep genes that have at least 2 TRUES in each row of thresh
        keep <- rowSums(thresh) >= 2
        # Subset the rows of countdata to keep the more highly expressed genes
        counts.keep <- countdata[keep,]
        summary(keep)
```

```
   Mode    FALSE    TRUE
        logical   11375   15804
```

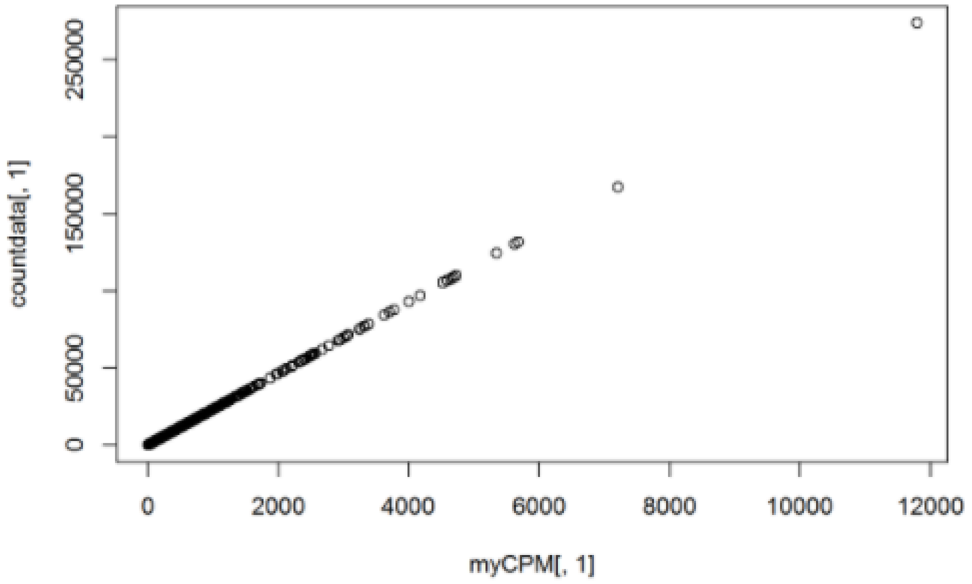
```
dim(counts.keep)
```

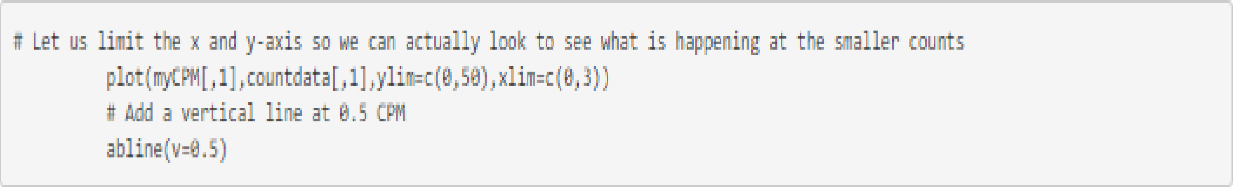
```
[1] 15804    12
```

A CPM of 0.5 is used as it corresponds to a count of 10-15 for the library sizes in this data set. If the count is any smaller, it is considered to be very low, indicating that the associated gene is not expressed in that sample. A requirement for expression in two or more libraries is used as each group contains two replicates. This ensures that a gene will be retained if it is only expressed in one group. Smaller CPM thresholds are usually appropriate for larger libraries. As a general rule, a good threshold can be chosen by identifying the CPM that corresponds to a count of 10, which in this case is about 0.5. You should filter with CPMs rather than filtering on the counts directly, as the latter does not account for differences in library sizes between samples.

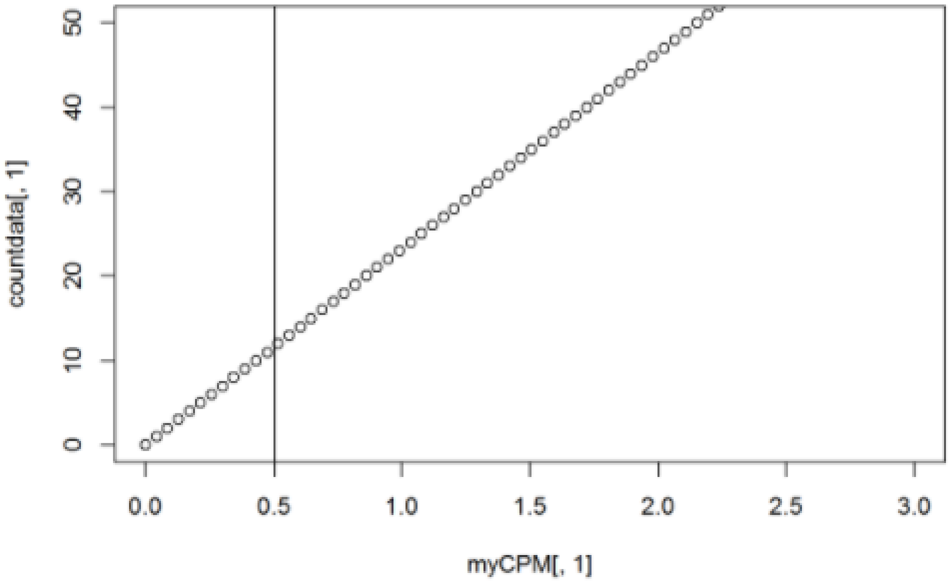


```
# Let's have a look and see whether our threshold of 0.5 does indeed correspond to a count of about 10-15
        # We will look at the first sample
        plot(myCPM[,1],countdata[,1])
```





```
# Let us limit the x and y-axis so we can actually look to see what is happening at the smaller counts
          plot(myCPM[,1],countdata[,1],ylim=c(0,50),xlim=c(0,3))
          # Add a vertical line at 0.5 CPM
          abline(v=0.5)
```



Useful Links:

- http://garberlab.umassmed.edu/data/RNASeqCourse/cufflinks.manual.pdf
- https://rnaseq.uoregon.edu/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3334321/pdf/nihms-366741.pdf

Indonesia International Institute for Life Science

Transcriptomics

COURSE CODE: BI021

GSEA Tutorial

Laboratory Procotol Developer and Supervisor(s) Information

Protocol Developer: Dr.rer.nat Arli Aditya Parikesit

Email: arli.parikesit@i3l.ac.id

Supervisor(s)	Email
David Agustriawan.,PhD	david.agustriawan@i3l.ac.id
Andreas Whisnu.,ST	andreas.whisnu@i3l.ac.id

Notice

1. Operate ONLY the computer assigned to you.
    a. If you have any troubleshooting, please contact your supervisor or Building Management
    b. Do not rename files, adjust the dock size/icons, move items or files to the trash, or change
    c. the system preferences unless directed to do so
    d. Do not exchange keyboards, mice, or other equipment among the computers without notifying your supervisor
    e. Do not bring food or drinks into the lab unless it is in your backpack

2. Remain at your work center, respectful behavior promotes learning. Show integrity and respect for class materials. Responsibility is fundamental. Misused equipment will be replaced by those who damage it.

Session	6
Date	Click here to enter text.
Laboratory	Bioinformatics laboratory

Overview

This course session is designed to teach how to be familiar with GSEA application.

The main objective of this learning experience are:

- To be familiar with GSEA application
- To understand on how to perform GSEA analysis

Material

1. Protocol practicum to perform Blast in Linux Ubuntu

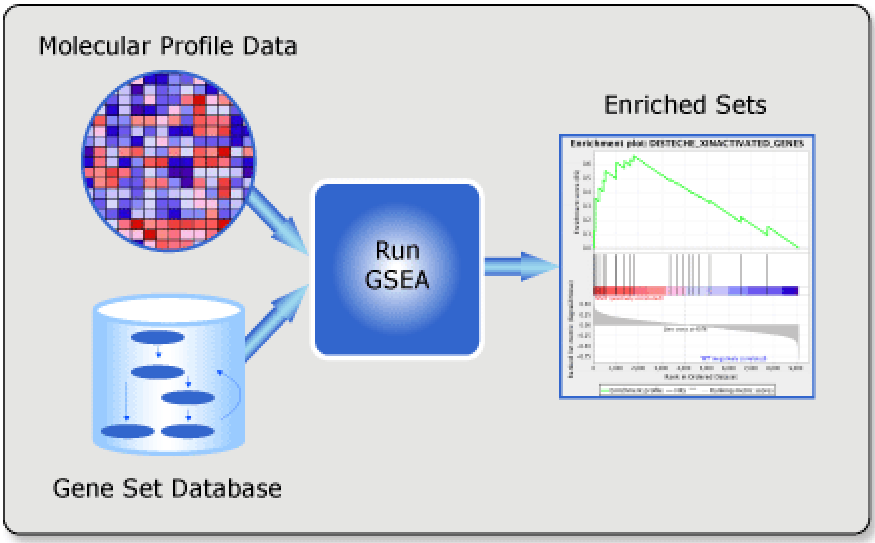
1. GSEA Tutorial - Overview

next >

The GSEA Desktop Application Tutorial provides a brief overview of the main features of the GSEA application. It is organized in a series of slides which may be navigated by pressing "Next", or you may jump to any section of interest using the links to the left. For more detailed information, see the Documentation page.

Equipment

1. Logbook
2. Laptop/PC (available in Bioinformatics laboratory)



2. GSEA Tutorial - Ways to Run GSEA

next >

You can run GSEA in multiple ways:

1. The GSEA desktop application provides an easy-to-use graphical interface. When you launch the application from the download page of the GSEA web site, as you will do in this tutorial, you are using Java Web Start technology (http://java.sun.com/products/javawebstart/) to download, install, and start the application.
2. The GSEA .jar file provides command line access to GSEA and allows you to run the GSEA desktop application without being connected to the internet. You can download the .jar file from the download page of the GSEA web site.

3. R-GSEA makes GSEA available from the R programming environment.
4. A GSEA GenePattern module makes GSEA available from GenePattern.

3. GSEA Tutorial - Launching GSEA

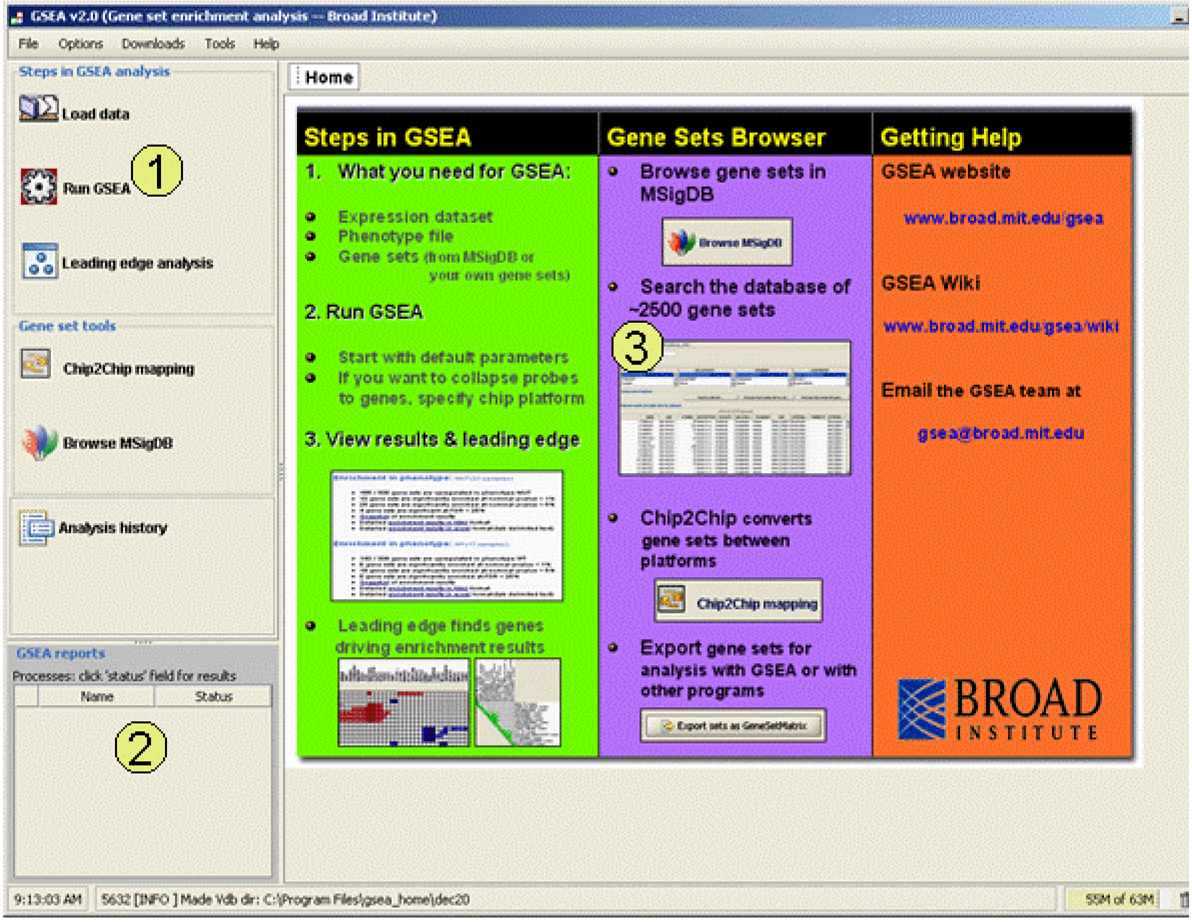
next >

To launch GSEA:

1. Go to the Downloads page.
2. Register as instructed.
3. Click the **Launch** icon to start the GSEA Desktop Application.

When GSEA starts, the main window appears. The main components of the user interface are:

1. The navigation bar on the left, which provides quick access to common GSEA operations.
2. The Processes panel in the bottom left corner, which provides information about the status of your analyses.
3. The main panel on the right, which is used to display diaglogs and results. When you start GSEA, the main panel displays the Home page. As you open new pages, tabs will appear next to the Home tab. To close a page, click the close (X) icon on the tab.



Steps in GSEA

1. What you need for GSEA:
- Expression dataset
- Phenotype file
- Gene sets (from MSigDB or your own gene sets)

2. Run GSEA
- Start with default parameters
- If you want to collapse probes to genes, specify chip platform

3. View results & leading edge
- Leading edge finds genes driving enrichment results

Gene Sets Browser
- Browse gene sets in MSigDB
- Search the database of ~2500 gene sets
- Chip2Chip converts gene sets between platforms
- Export gene sets for analysis with GSEA or with other programs

Getting Help

GSEA website

www.broad.mit.edu/gsea

GSEA Wiki

www.broad.mit.edu/gsea/wiki

Email the GSEA team at

gsea@broad.mit.edu

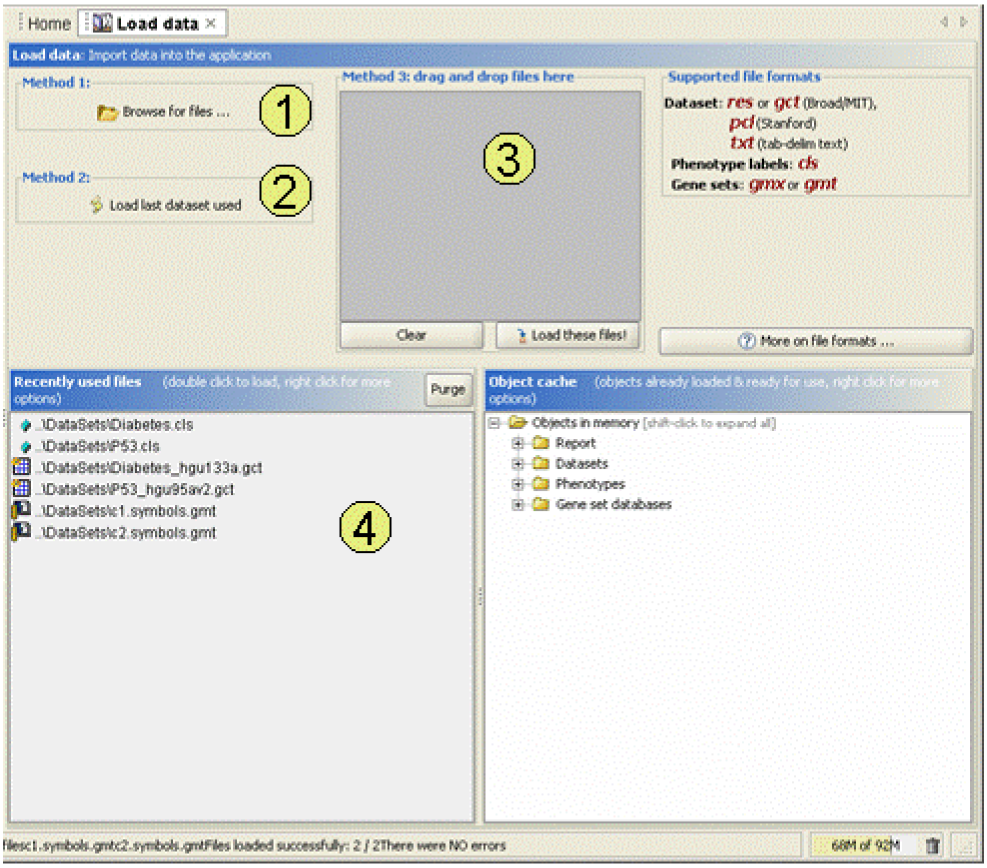
5632 [INFO] Made Vdb dir: C:\Program Files\gsea_home\dec20

4. GSEA Tutorial - Loading Data

next >

Click the **Load Data** icon in the navigation bar. The Load Data page appears. You use this page to load your data files: expression datasets, phenotype labels (e.g tumor vs normal), gene sets, and chip annotations. Once imported these files are stored in memory and are available to the program for analysis.

GSEA supported data files are simply tab delimited ASCII text files, which have special file extensions that identify them. For example, expression data usually has the extension *.gct, phenotypes *.cls, gene sets *.gmt, and chip annotations *.chip. Click the **More on file formats** help button to view detailed descriptions of all the data file formats.


Home
Load data
Load data: Import data into the application
Method 1:
Browse for files ...
1
Method 2:
Load last dataset used
2
Method 3: drag and drop files here
3
Clear
Load these files!
Supported file formats
More on file formats ...
Recently used files (double click to load, right click for more options)
Purge
4
Object cache (objects already loaded & ready for use, right click for more options)
Objects in memory [shift-click to expand all]
Report
Datasets
Phenotypes
Gene set databases
68M of 92M

Dataset: res or gct (Broad/MIT),
pcl (Stanford)
txt (tab-delim text)
Phenotype labels: cls
Gene sets: gmx or gmt

..\DataSets\Diabetes.cls
..\DataSets\P53.cls
..\DataSets\Diabetes_hgu133a.gct
..\DataSets\P53_hgu95av2.gct
..\DataSets\c1.symbols.gmt
..\DataSets\c2.symbols.gmt

files c1.symbols.gmtc2.symbols.gmtFiles loaded successfully: 2 / 2There were NO errors

GSEA provides several ways to load data:

1. Click the **Browse for files** button. When the Open window appears, select the file(s) to load and then click the Open button. To select multiple files, use SHIFT-click or CTRL-click.
2. Click the **Load last dataset used** button. GSEA loads the data used in the most recent gene set enrichment analysis.
3. Drag-and-drop the files from a file browser window into the drag-and-drop pane. When the files that you want to load are listed in that pane, click the **Load these files** button. To remove files from the drag-and-drop pane, click the **Clear** button.
4. The Recently Used Files pane contains files that you have used previously. (The first time you start GSEA, this pane is empty.) Double-click a file to load it.

The Object Cache pane lists the data that you have loaded into memory.

5. GSEA Tutorial - Loading the P53 Sample Data

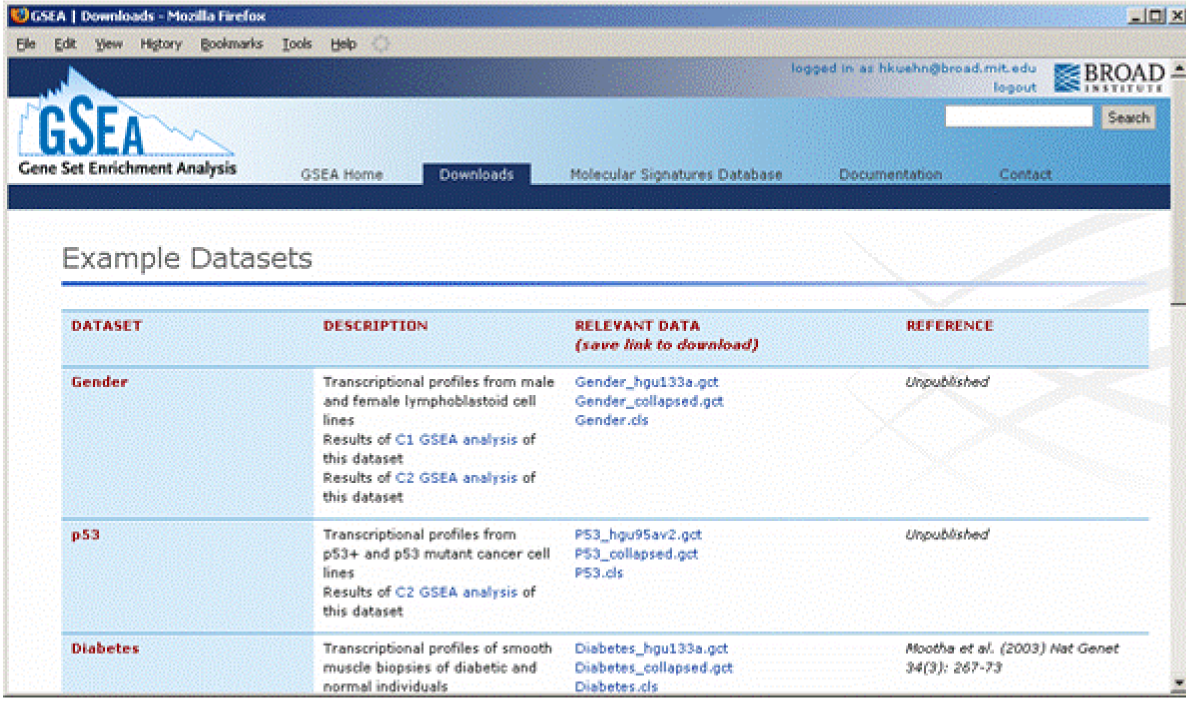
next >

The GSEA web site provides several sample datasets that correspond to results from the GSEA Subramanian & Tamayo PNAS 2005 paper. For the tutorial, you will use the P53 sample data.

To download the P53 sample files:

1. Go to the Datasets page.
2. Download the three p53 data files. For each file: right-click on the file, select **Save link as** and save the file to your local drive.

3. Confirm that the saved files have a .gct or .cls file extension. If a .txt file extension has been appended, remove it.

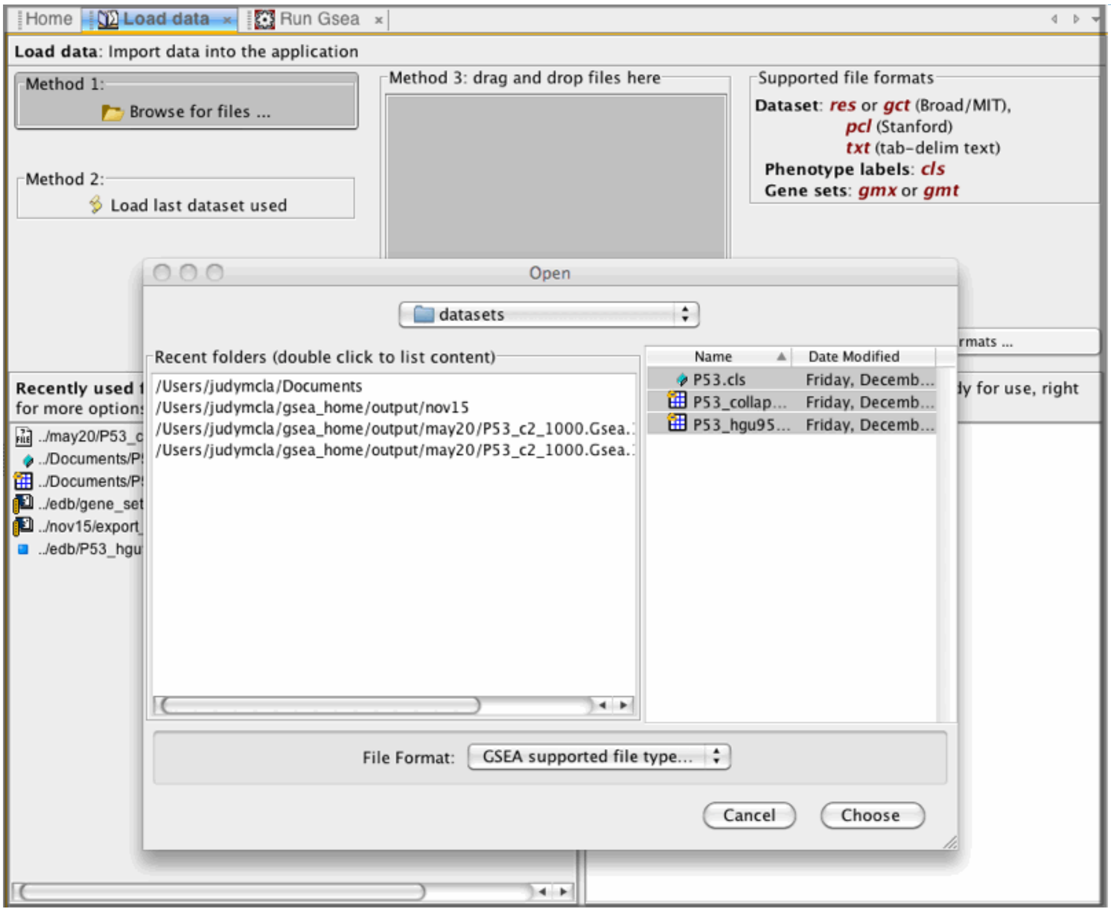

GSEA | Downloads - Mozilla Firefox
File Edit View History Bookmarks Tools Help
logged in as hkuehn@broad.mit.edu
logout
BROAD INSTITUTE
Search
GSEA
Gene Set Enrichment Analysis
GSEA Home
Downloads
Molecular Signatures Database
Documentation
Contact

Example Datasets

DATASET	DESCRIPTION	RELEVANT DATA *(save link to download)*	REFERENCE
Gender	Transcriptional profiles from male and female lymphoblastoid cell lines Results of C1 GSEA analysis of this dataset Results of C2 GSEA analysis of this dataset	Gender_hgu133a.gct Gender_collapsed.gct Gender.cls	*Unpublished*
p53	Transcriptional profiles from p53+ and p53 mutant cancer cell lines Results of C2 GSEA analysis of this dataset	P53_hgu95av2.gct P53_collapsed.gct P53.cls	*Unpublished*
Diabetes	Transcriptional profiles of smooth muscle biopsies of diabetic and normal individuals	Diabetes_hgu133a.gct Diabetes_collapsed.gct Diabetes.cls	*Mootha et al. (2003) Nat Genet 34(3): 267-73*

To load the P53 data into GSEA:

1. Go to the Load Data page of the GSEA application.
2. Click **Browse for files**.
3. Select the three files that you just downloaded.
4. Click Open.



/Users/judymcla/Documents
/Users/judymcla/gsea_home/output/nov15
/Users/judymcla/gsea_home/output/may20/P53_c2_1000.Gsea.
/Users/judymcla/gsea_home/output/may20/P53_c2_1000.Gsea.

Name	Date Modified
P53.cls	Friday, Decemb...
P53_collap...	Friday, Decemb...
P53_hgu95...	Friday, Decemb...

File Format: GSEA supported file type...

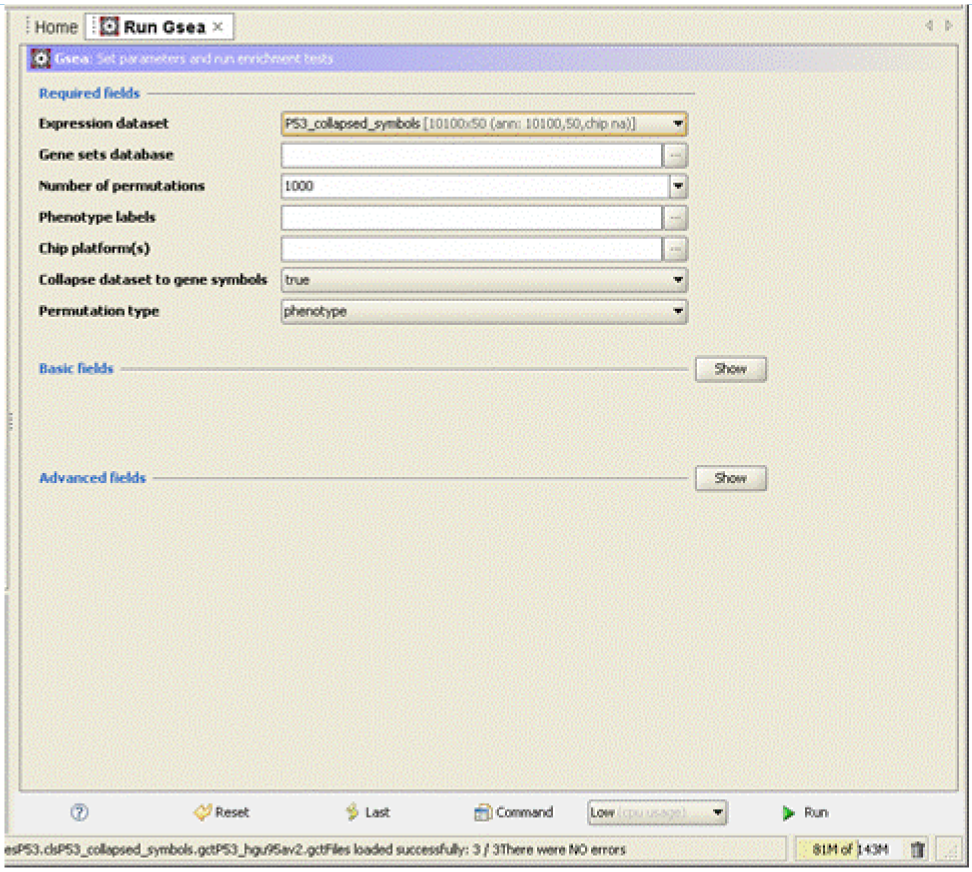
6. GSEA Tutorial - Analysis Parameters

next >

Now that you have loaded your data files, you are ready to run the gene set enrichment analysis. Click the **Run GSEA** icon in the navigation bar. The Run GSEA page displays the parameters for the analysis. There are three categories of parameters:

1. **Required**: Essential parameters which you must specify before the analysis can be run.
2. **Basic**: Additional parameters with standard defaults. Typically, accepting the defaults is ok. Click **Show** to see these parameters.
3. **Advanced**: Parameters that allow control of several more details of the GSEA algorithm and the java implementation. Typically, these do not need to be changed by most users. Click **Show** to see these parameters.

For descriptions of the parameters, click the **? help** button.



Gsea: Set parameters and run enrichment tests

Required fields

Expression dataset: P53_collapsed_symbols [10100x50 (ann: 10100,50,chip na)]

Gene sets database:

Number of permutations: 1000

Phenotype labels:

Chip platform(s):

Collapse dataset to gene symbols: true

Permutation type: phenotype

Basic fields

Advanced fields

Low (cpu usage)

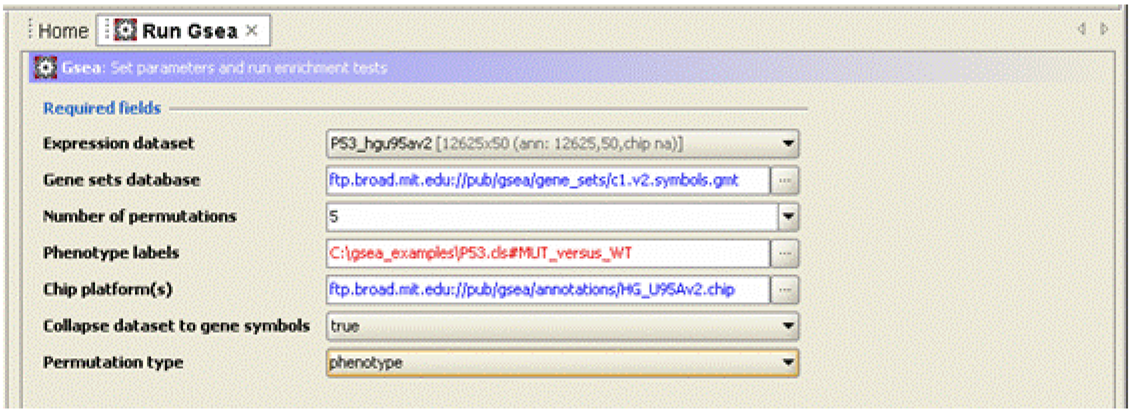
esP53.clsP53_collapsed_symbols.gctP53_hgu95av2.gctFiles loaded successfully: 3 / 3There were NO errors

81M of 143M

7. GSEA Tutorial - Running the Gene Set Enrichment Analysis

next >

To run the analysis, set the parameters and click the **Run** button.



Gsea: Set parameters and run enrichment tests

Required fields

Expression dataset: P53_hgu95av2 [12625x50 (ann: 12625,50,chip na)]

Gene sets database: ftp.broad.mit.edu://pub/gsea/gene_sets/c1.v2.symbols.gmt

Number of permutations: 5

Phenotype labels: C:\gsea_examples\P53.cls#MUT_versus_WT

Chip platform(s): ftp.broad.mit.edu://pub/gsea/annotations/HG_U95Av2.chip

Collapse dataset to gene symbols: true

Permutation type: phenotype

1. Use the drop-down selector to pick the p53_hgu95av2 dataset.
2. Use the **...** button to pick one or more gene sets. GSEA displays a window that lists gene sets in a number of different tabs. For this example, on the **GeneMatrix (from website)** tab select the c1.v2.symbols.gmt.
3. Type in or choose the number of permutations to perform. Typically, you start with a small number (perhaps 5) and, when that successfully completes, try a full set of 1000 permutation. For now, choose 5.
4. Use the **...** button to pick a phenotype. In this sample data, the two phenotypes are the same (MUT_vs_WT or WT_vs_MUT).
5. Use the **...** to select the chip annotation file that matches the probe identifiers in your expression dataset. For this example, on the **Chips (from website)** tab, choose the

HG_U95Av2.chip file.

6. Leave the **Collapse dataset to gene symbols** parameter set to true. This indicates that you want the probe sets in your dataset collapsed to gene symbols.
7. Leave the **Permutation type** parameter set to phenotype.
8. Click **Run** to start the analysis.

8. GSEA Tutorial - Keeping Identifiers Consistent Between Platforms

next >

Typically, the gene or probe identifiers in your expression dataset are the probe identifiers for the DNA chip array used to produce the data. When running the gene set enrichment analysis, it is critical that all of your data files use the same gene or probe identifiers. You can either use the probe identifiers native to your expression dataset, or collapse each probe set into a gene vector and use HUGO gene symbols as your identifiers.

When you run the gene set enrichment analysis, the value you choose for the Collapse dataset to gene symbols parameter tells GSEA which identifiers you want to use:

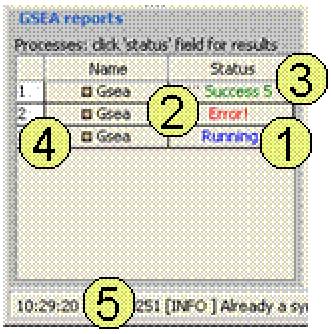
1. Choose true (default) to have GSEA collapse each probe set in your expression dataset into a single gene vector, which is identified by its HUGO gene symbol. In this case, you are using HUGO gene symbols for the analysis. The gene sets that you use for the analysis must use HUGO gene symbols to identify the genes in the gene sets.
2. Choose false to use your expression dataset "as is." In this case, you are using the probe identifiers that are in your expression dataset for the analysis. The gene sets that you use for the analysis must also use these probe identifiers to identify the genes in the gene sets.

Collapsing the probe sets eliminates multiple probes, which can inflate enrichment scores, and facilitates the biological interpretation of the gene set enrichment analysis results. Therefore, the GSEA team recommends leaving the default value for this parameter.

9. GSEA Tutorial - Viewing Program Progress and Results

next >

Use the Processes panel at the lower left corner to view the status of analyses run in this session, including the currently running analysis:



GSEA reports

Processes: click 'status' field for results

	Name	Status
1	Gsea	Success 5
2	Gsea	Error!
	Gsea	Running

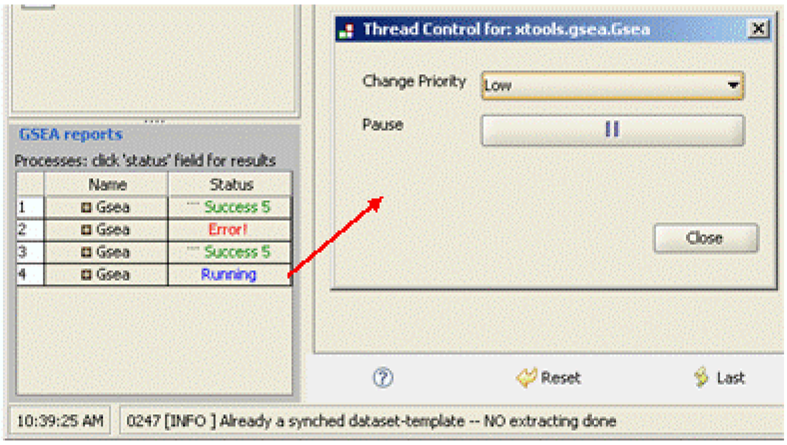
10:29:20 251 [INFO] Already a sy

1. The blue Running label indicates the currently running analysis. You can click on this label to pause or stop an analysis, as shown in the next slide.
2. If a red Error appears, click on it for a description of the error. If you need help resolving an error, include this error text in a posting to groups.google.com/group/gsea-help.
3. When the analysis completes, click the green Success label to display the results in a web browser. For help interpreting the results, see Interpreting GSEA Results in the GSEA User Guide.
4. Click the analysis name to view the parameters used in the analysis (a new Run GSEA page appears, which you can use to re-run the analysis).
5. Click the status bar at the bottom of the window to display the execution log, which shows analysis progress (for example, the number of permutations completed).

10. GSEA Tutorial - Stopping or Pausing a Running Analysis

next >

1. Click the blue Running label to display the thread control panel.
2. You can pause the analysis or change the amount of the computer's processor being used for the analysis.



Processes: click 'status' field for results

	Name	Status
1	Gsea	Success 5
2	Gsea	Error!
3	Gsea	Success 5
4	Gsea	Running

```
10:39:25 AM | 0247 [INFO ] Already a synched dataset-template -- NO extracting done
```

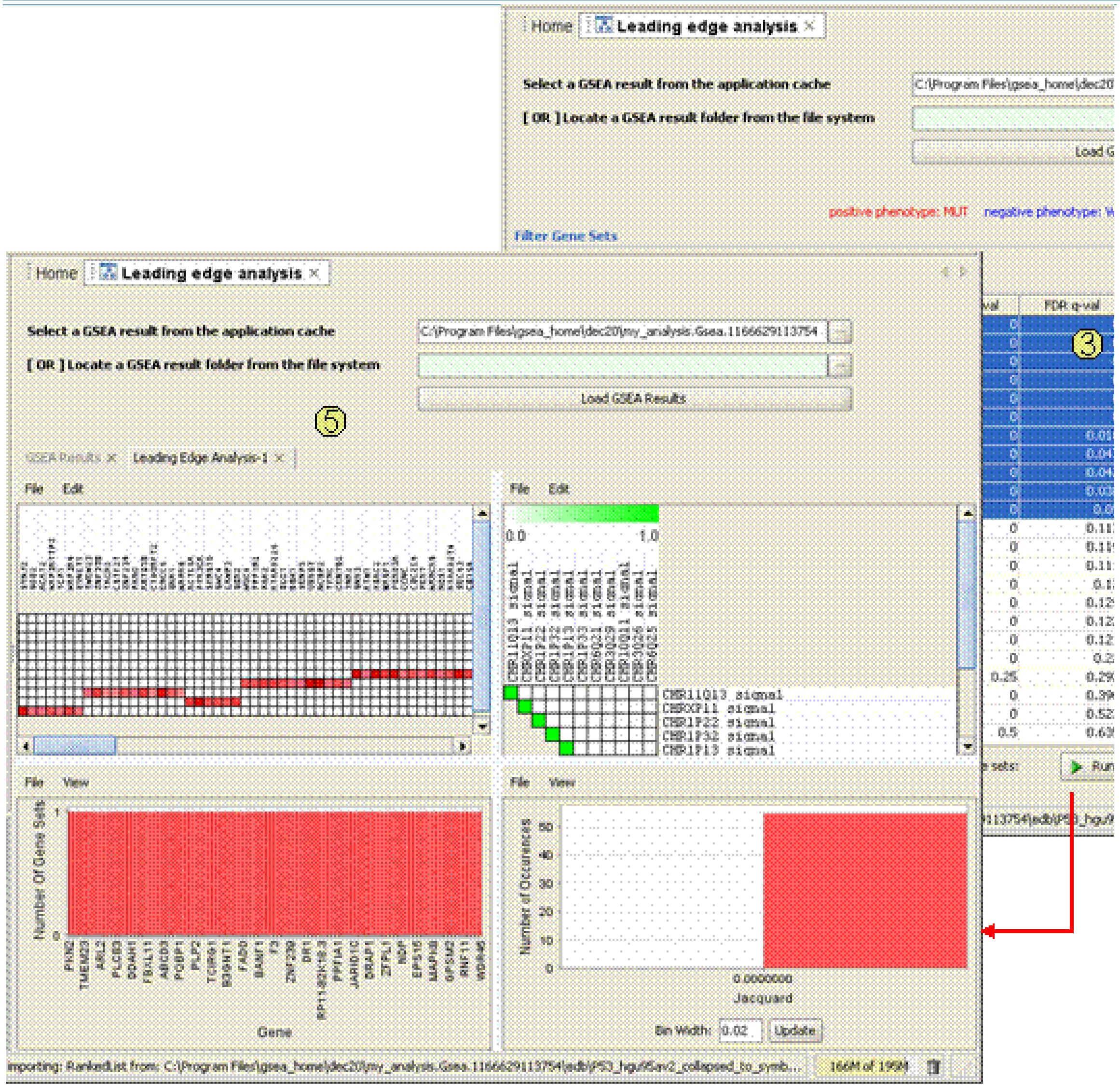
11. GSEA Tutorial - Running the Leading Edge Analysis

next

After running a gene set enrichment analysis, you can use the leading edge analysis to examine the genes in the leading edge subsets of selected enriched gene sets. Genes that appear in multiple subsets are more likely to be of interest than those that appear in only one.

To run a leading edge analysis, click the **Leading Edge Analysis** icon on the GSEA main page. When GSEA displays the Leading Edge Analysis page:

1. Click the **...** button to select a Gene Set Enrichment Report from the application cache (analyses that you have run).
2. Click the **Load GSEA Results** button to display the gene sets that were analyzed in that report.
3. SHIFT-click or CTRL-click to select the gene sets to analyze. For this example, click the FDR column head to order the gene sets by FDR and select the 11 gene sets with an FDR < .01.
4. Click the **Run leading edge analysis** button to start the analysis.
5. The analysis displays four graphs showing the overlap among the leading edge subsets of the selected gene sets. For help interpreting the results, see Interpreting Leading Edge Analysis Results in the *GSEA User Guide*.


Select a GSEA result from the application cache: C:\Program Files\gsea_home\dec20\my_analysis.Gsea.1166629113754

Bin Width: 0.02

Importing: RankedList from: C:\Program Files\gsea_home\dec20\my_analysis.Gsea.1166629113754\edb\P53_hgu95av2_collapsed_to_symb...

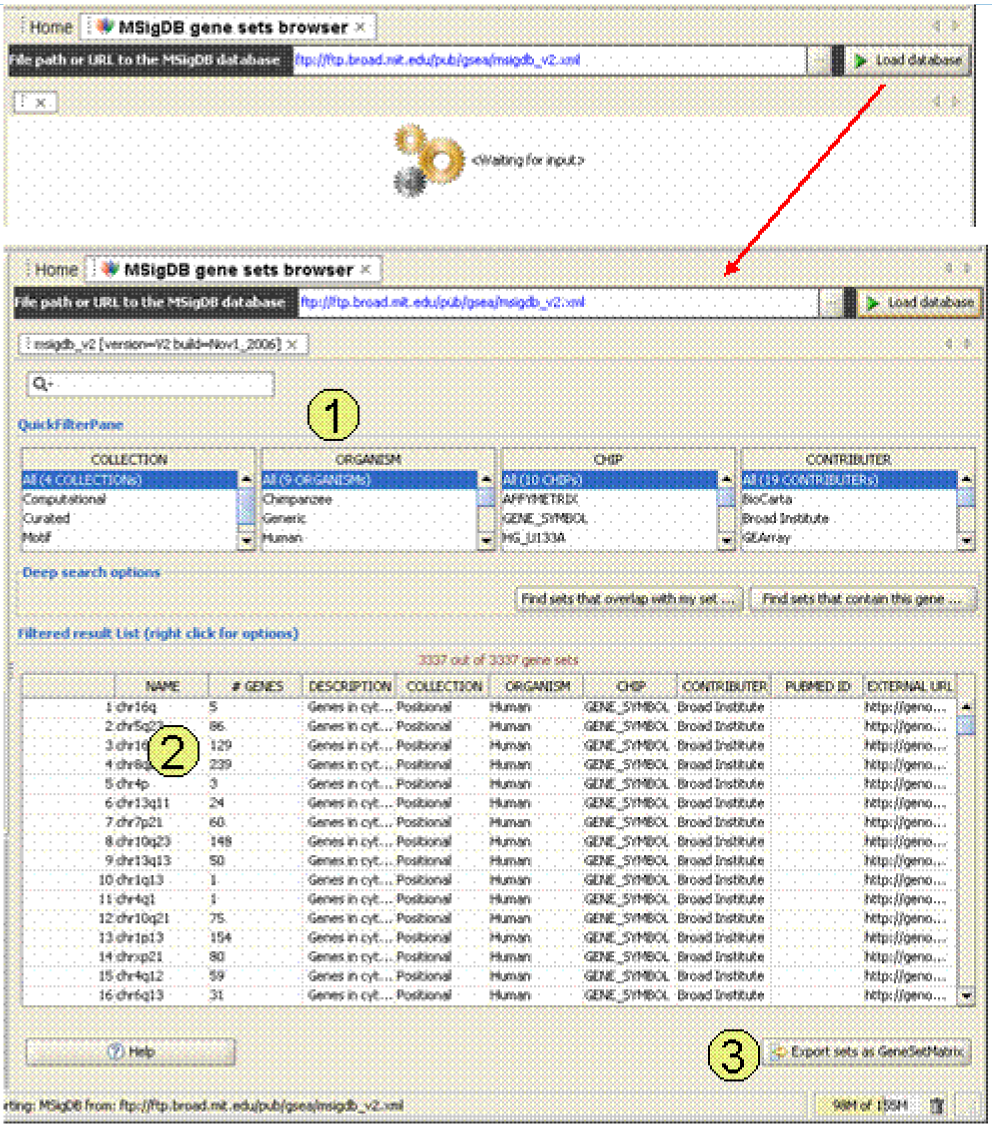
12. GSEA Tutorial - Browsing MSigDB Gene Sets

next >

The power of the gene set enrichment analysis is a function of how well your gene sets represent meaningful coordinated or concordant gene expression behavior that reflects actual biological processes or states. You are welcome to use curated gene sets from the Molecular Signature Database (MSigDB), which is maintained by the GSEA team.

You can browse the MSigDB from the Molecular Signatures Database page of the GSEA web site or the Browse MSigDB page of the MSigDB application that could be downloaded from here http://software.broadinstitute.org/gsea/downloads.jsp#msigdb . To browse the MSigDB from the application:

1. Click the **Browse MSigDB** icon in the navigation bar. An empty Browse MSigDB page appears.
2. Click the **Load database** button to display the latest MSigDB gene sets.


Home
MSigDB gene sets browser
Load database
<Waiting for input>
1
2
3
QuickFilterPane
Deep search options
Find sets that overlap with my set ...
Find sets that contain this gene ...
Filtered result List (right click for options)
Help
Export sets as GeneSetMatrix

File path or URL to the MSigDB database: ftp://ftp.broad.mit.edu/pub/gsea/msigdb_v2.xml

msigdb_v2 [version=V2 build=Nov1_2006]

COLLECTION	ORGANISM	CHIP	CONTRIBUTER
All (4 COLLECTIONs)	All (9 ORGANISMs)	All (10 CHIPs)	All (19 CONTRIBUTERs)
Computational	Chimpanzee	AFFYMETRIX	BioCarta
Curated	Generic	GENE_SYMBOL	Broad Institute
Motif	Human	HG_U133A	GEArray

3337 out of 3337 gene sets

	NAME	# GENES	DESCRIPTION	COLLECTION	ORGANISM	CHIP	CONTRIBUTER	PUBMED ID	EXTERNAL URL
1	chr16q	5	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
2	chr5q23	86	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
3	chr1[illegible]	129	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
4	chr6[illegible]	239	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
5	chr4p	3	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
6	chr13q11	24	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
7	chr7p21	60	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
8	chr10q23	148	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
9	chr13q13	50	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
10	chr1q13	1	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
11	chr4q1	1	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
12	chr10q21	75	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
13	chr1p13	154	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
14	chrxp21	80	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
15	chr4q12	59	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...
16	chr6q13	31	Genes in cyt...	Positional	Human	GENE_SYMBOL	Broad Institute		http://geno...

ting: MSigDB from: ftp://ftp.broad.mit.edu/pub/gsea/msigdb_v2.xml

98M of 155M

From this page you can

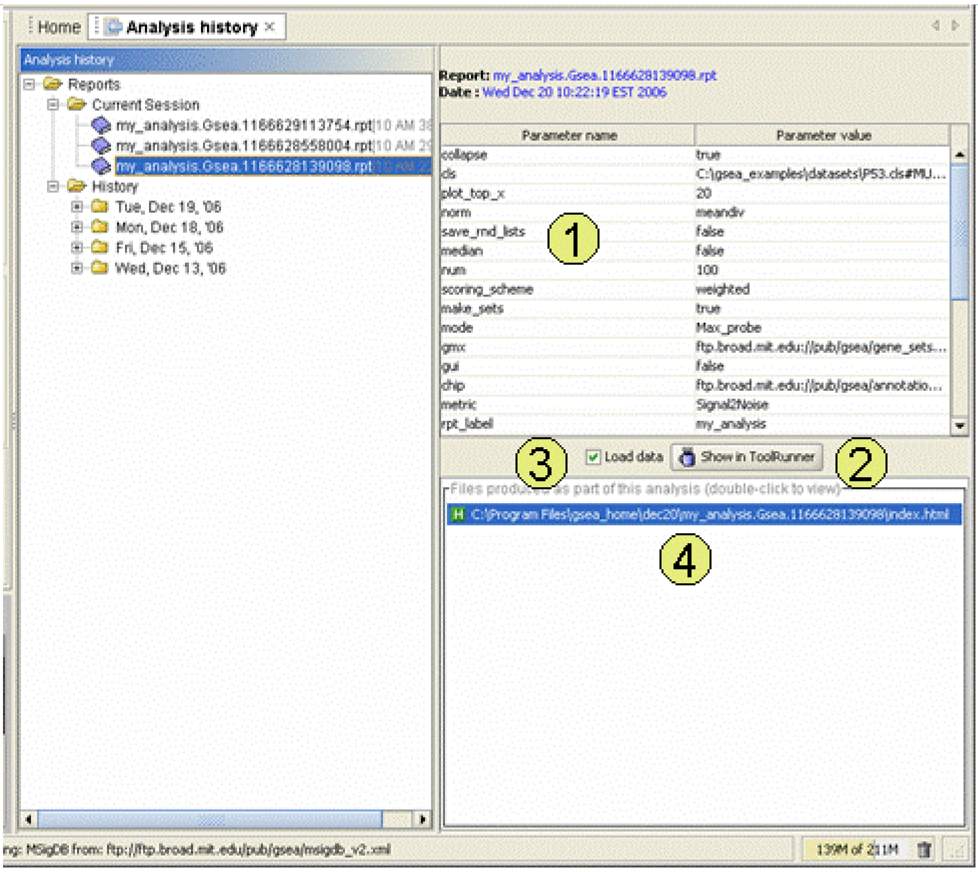
1. Use the fields at the top of the page to filter the gene sets displayed in the table.
2. Select a gene set from the table and right-click to display information about the gene set.
3. When the table displays the gene sets that you are interested in, export the selected gene sets to a gene set file.

GSEA exports the gene set files to your default output folder (**Help>Show GSEA Output Folder**). The gene set files are tab-delimited ASCII text files that can be viewed in Excel or NotePad.

13. GSEA Tutorial - Viewing Analysis History

next >

Click the **Analysis History** icon in the navigation bar to display the Analysis History page, which records and displays analyses that you have run. The left panel lists the reports run in the current session and organizes previously run reports by date. Click on an analysis in the left panel to display information about that analysis in the right panel.



Report: my_analysis.Gsea.1166628139098.rpt
Date : Wed Dec 20 10:22:19 EST 2006

Parameter name	Parameter value
collapse	true
ds	C:\gsea_examples\datasets\P53.cls#MU...
plot_top_x	20
norm	meandiv
save_rnd_lists	false
median	false
num	100
scoring_scheme	weighted
make_sets	true
mode	Max_probe
gmx	ftp.broad.mit.edu://pub/gsea/gene_sets...
gui	false
chip	ftp.broad.mit.edu://pub/gsea/annotatio...
metric	Signal2Noise
rpt_label	my_analysis

C:\Program Files\gsea_home\dec20\my_analysis.Gsea.1166628139098\index.html

ng: MSigDB from: ftp://ftp.broad.mit.edu/pub/gsea/msigdb_v2.xml

In the right panel of the Analysis History page:

1. You can view the parameters used in the analysis.
2. You can choose to re-run an analysis with the exact same set of parameters by clicking the **Show in ToolRunner** button.
3. You can choose to automatically load or not load data from the previous analysis (perhaps you are on a different computer or are only interested in the previous parameters to use with different datasets).
4. You can view files produced by the analysis. Double-click the index.html file to display the analysis results in a web browser.

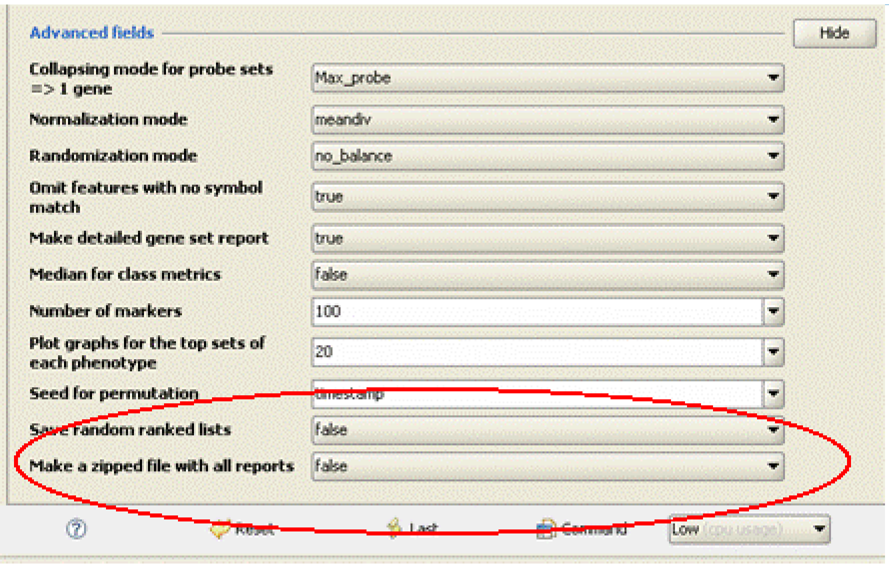
Note: When you run an analysis, by default, GSEA writes the analysis results to the GSEA output folder (**Help>Show GSEA output folder**). The Analysis History page is simply a convenient way to browse the reports in this folder.

14. GSEA Tutorial - Sharing Results with Collaborators

next >

Sharing GSEA analysis results with collaborators is easy. Click Help>Show GSEA output folder to display the folder that holds the GSEA reports, navigate to the subfolder for the report that you want to share, zip it up, and send it to your collaborator. All reports and their hyperlinks are preserved.

Alternatively, when you run an analysis, you can have GSEA create the zip for you by setting the Make a zipped file with all reports parameter to true (by default, the parameter is set to false).



Advanced fields

Collapsing mode for probe sets => 1 gene: Max_probe

Normalization mode: meandiv

Randomization mode: no_balance

Omit features with no symbol match: true

Make detailed gene set report: true

Median for class metrics: false

Number of markers: 100

Plot graphs for the top sets of each phenotype: 20

Seed for permutation: timestamp

Save random ranked lists: false

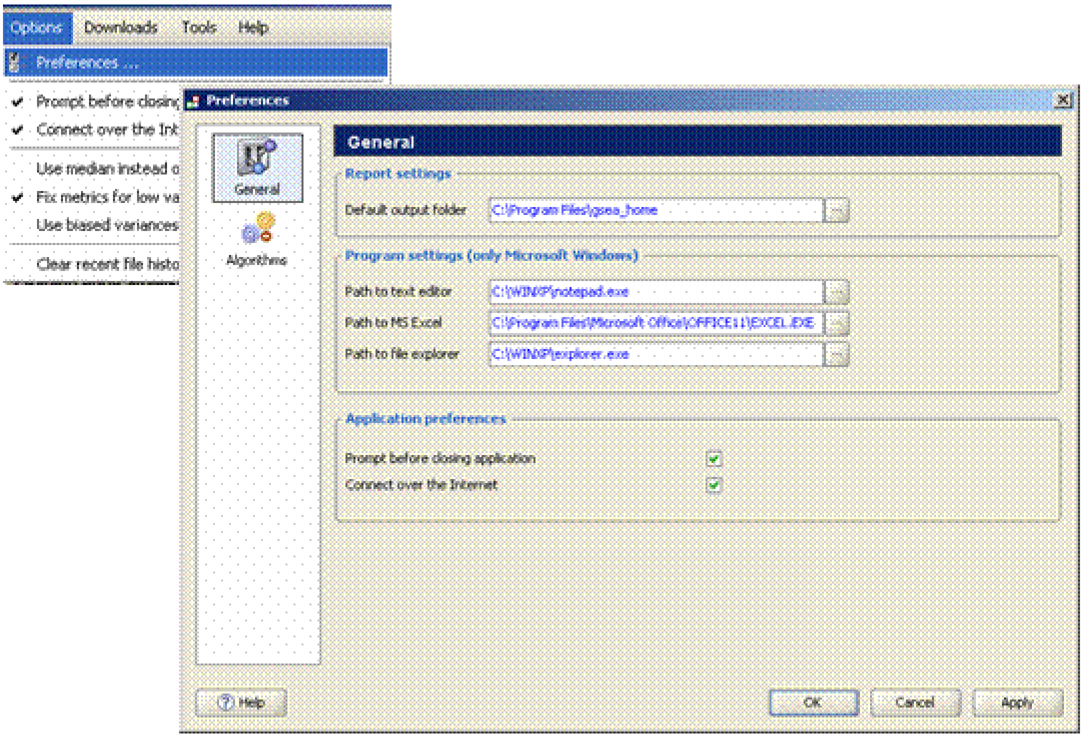
Make a zipped file with all reports: false

15. GSEA Tutorial - Setting Preferences

next >

The Options menu provides several preferences to control the application and algorithm defaults.

One useful preference is the location of your GSEA output folder, which holds all of the analysis results (Help>Show GSEA output folder). By default, the output folder is a subfolder of your GSEA home folder. To change the location of your default output folder, click Options>Preferences. When the Preferences window appears, change the default output folder and click OK.



General

Report settings

Default output folder: C:\Program Files\gsea_home

Program settings (only Microsoft Windows)

Path to text editor: C:\WINXP\notepad.exe

Path to MS Excel: C:\Program Files\Microsoft Office\OFFICE11\EXCEL.EXE

Path to file explorer: C:\WINXP\explorer.exe

Application preferences

Prompt before closing application ☑

Connect over the Internet ☑

16. GSEA Tutorial - Creating Data Files for GSEA

next >

The gene set enrichment analysis requires four files: an expression dataset file, phenotype labels file, gene sets file, and chip annotations file. All four files are tab-delimited ASCII text files that can be created and edited using Excel or any text editor.

1. Expression dataset file: This file contains your expression data: genes/probes, samples, and expression values for each probe in each sample. Your expression data can come from any source (Affymetrix, CDNA 2-color ratio data, and so on). You create an expression data file by converting your expression data into a gct, res, or pcl formatted file. Typically, your

expression data is already in a tab-delimited ASCII text file, which can be turned into a gct, res, or pcl formatted file with relatively minor edits.

2. Phenotype label file: This file lists your phenotype labels and associates each sample in your dataset with a phenotype. You can create this file or have GSEA create it for you (you supply the phenotype information and GSEA creates the appropriate file).
3. Gene sets file: This file defines the gene sets to be analyzed. You can use the gene sets that are available on the Broad ftp site, export gene sets from the MSigDB, or create your own. If you have gene sets that you want to use, GSEA provides a Chip-to-Chip utility, which converts gene/probe identifiers from one DNA chip platform to another (or to HUGO gene symbols).
4. Chip annotations file: This file maps probe identifiers to HUGO gene symbols. GSEA uses it to collapse each probe set in your dataset to a single gene vector (if you choose to collapse your dataset) and to annotate the gene set enrichment report. The chip annotations files for common DNA chip platforms are available on the Broad ftp site. If necessary (for example, if you are using custom chips), you can create your own chip annotations file.

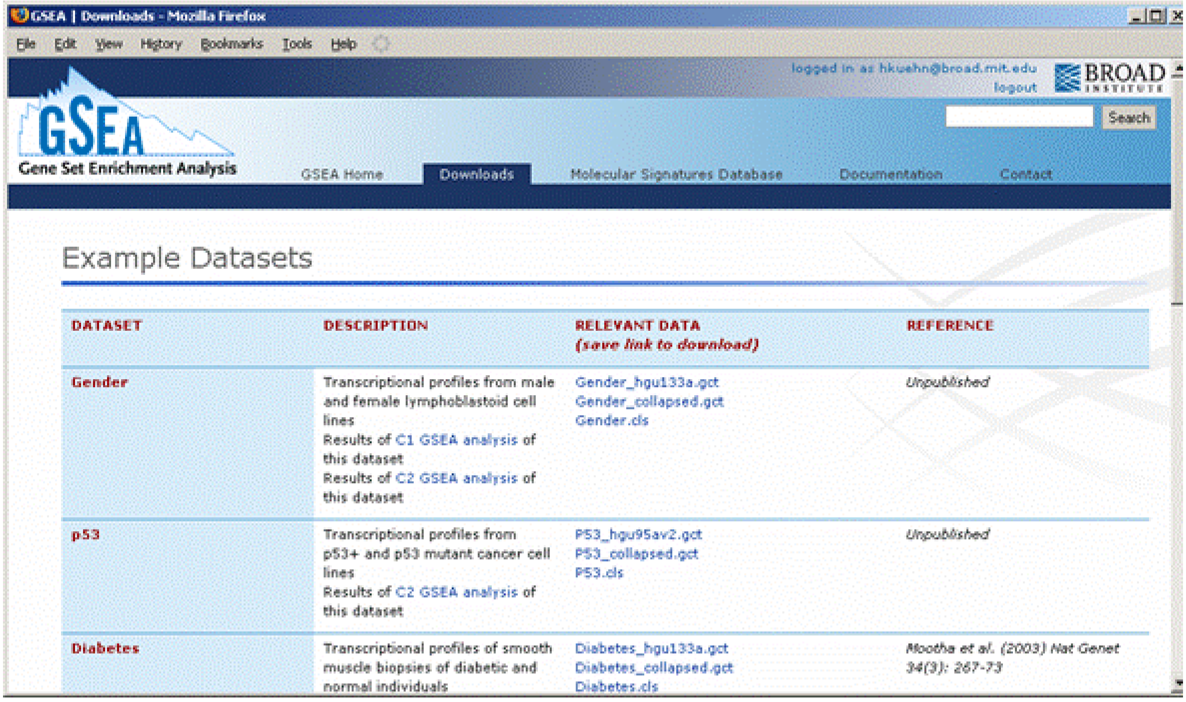
For descriptions of all of the GSEA file formats, see Data Formats. For more information about creating the data files, see Preparing Data Files for GSEA in the GSEA User Guide.

17. GSEA Tutorial - Examples from Published GSEA Results

next >

The GSEA web site provides the datasets that correspond to results from the GSEA Subramanian & Tamayo PNAS 2005 paper:

1. Go to the Downloads page.
2. Near the bottom of the page, click view datasets.


GSEA | Downloads - Mozilla Firefox
File Edit View History Bookmarks Tools Help
logged in as hkuehn@broad.mit.edu
logout
BROAD INSTITUTE
GSEA
Gene Set Enrichment Analysis
Search
GSEA Home
Downloads
Molecular Signatures Database
Documentation
Contact

Example Datasets

DATASET	DESCRIPTION	RELEVANT DATA *(save link to download)*	REFERENCE
Gender	Transcriptional profiles from male and female lymphoblastoid cell lines Results of C1 GSEA analysis of this dataset Results of C2 GSEA analysis of this dataset	Gender_hgu133a.gct Gender_collapsed.gct Gender.cls	*Unpublished*
p53	Transcriptional profiles from p53+ and p53 mutant cancer cell lines Results of C2 GSEA analysis of this dataset	P53_hgu95av2.gct P53_collapsed.gct P53.cls	*Unpublished*
Diabetes	Transcriptional profiles of smooth muscle biopsies of diabetic and normal individuals	Diabetes_hgu133a.gct Diabetes_collapsed.gct Diabetes.cls	*Mootha et al. (2003) Nat Genet 34(3): 267-73*

Note: Because random number generators (for sample permutation) are different and because different seeds are used, numbers in the reports on the website, or reports run with the sample date, will not precisely match those in the paper. However, the significant sets are identical to published results.

18. GSEA Tutorial - Getting Help for GSEA

As you begin to use GSEA, you can get help in several ways:

1. Click **Help>GSEA documentation** to view the Documentation page, which includes the *GSEA User Guide* and a Frequently Asked Questions (FAQ) page.

2. Click the **Help** button, which appears on most GSEA windows, to display context-sensitive help.
3. If you cannot find the information that you are looking for in the documentation, contact us atgroups.google.com/group/gsea-help.

Thanks for taking the time for this Quick Tour of GSEA. If you have questions, comments or suggestions, we'd like to hear them: groups.google.com/group/gsea-help.