Introduction to Unix for Bioinformatics

Handling genomic coordinates

Contents

Opening a session.

Starting a terminal

1. Connect to the computer using your login and password.
2. Open a terminal. You should find a terminal (terminal, konsole, terminator,...) under the *Application Menu*: Applications > Accessories > Terminal

Using this terminal you should now be able to communicate with the system through a set of shell commands (your default shell command interpreter should be BASH).

Creating a working directory

1. Using the *mkdir command* (**ma**k**e** **dir**ectory) create a working directory named *td1_genomic_coordinates* in your HOME folder.
2. Use the *cd* (**c**hange **d**irectory) command to enter the *td1_genomic_coordinates* directory

View solution| Hide solution

Solution

```
cd ~  # or cd
pwd # check the current working directory
mkdir td1_genomic_coordinates
cd td1_genomic_coordinates # or cd ~/td1_genomic_coordinates
```

[back to contents]

Downloading RefSeq transcript coordinates

RefSeq transcript

The Reference Sequence (RefSeq) collection aims to provide a comprehensive, integrated, non-redundant, well-annotated set of sequences. Here we will download informations about RefSeq transcripts (that is known full-length transcript) related to Mus musculus genome (mm9 version). Refseq transcript can be obtained from UCSC genome browser.

1. Go to the UCSC web site
2. In the middle of the left menu select "Downloads".
3. Select the Mouse genome.
4. In the "Mar. 2006 (mm9)" section select "Annotation database" to enter the UCSC ftp web site.
5. Find the file `refGene.txt.gz` file copy the link location (URL) by right-clicking on the link.
6. In a terminal use the *wget* command followed by the address to retrieve the *refGene.txt.gz* file.

7. Uncompress the file using the *gunzip* command.

View solution| Hide solution

Solution

```
wget
   http://hgdownload.cse.ucsc.edu/goldenPath/mm9/database/refGe
gunzip refGene.txt.gz
```

What kind of informations are available for each transcript?

1. Go to the UCSC web site
2. Select *Tables* in the top menu
3. Select *Clade: Mammal, Genome: Mouse, assembly: mm9, group: Genes and Gene Prediction tracks, table: refGene* and click on *describe table schema*

Let's have a look at the refGene file?

1. Use the *wc* command with *-l* arguments to count the number of lines in the refGene.txt file
2. Use the *head* command with *-n* arguments to see the first 5 lines of the refGene.txt file.
3. Use the *tail* command with *-n* arguments to see the last 5 lines of the refGene.txt file.
4. Have a look at the whole file using *less* (use up/down arrow keys and type "**q**" to **q**uit))
5. Using *head* extract the first line and use a *pipe* to redirect the output to the *od* command with *-c* argument.

What can you guess from the results?

View solution| Hide solution

Solution

```

wc -l refGene.txt
head -n 5 refGene.txt
tail -n 5 refGene.txt
less refGene.txt
head -n 1 refGene.txt  | od -c
```

Some descriptive statistics on refSeq transcript using basic unix command

1. Using the *cut* command retrieve the 13th column
2. Using the *cut* command retrieve the 13th column and redirect the output to the *grep* command to find genes whose symbol start with *Bcl*. Count them by redirecting the output to the wc -l command
3. What is the gene that display the highest number of transcripts (use *sort* and *uniq* with *-c* argument). Check the results by typing the gene symbol in the UCSC genome browser (top menu > genomes)
4. What is the gene that display the highest number of exons.
5. Some transcrits may be coding (NM_...) or non-coding (NR_...). How many transcripts of each class are available in the refGene.txt file?
6. How many genes does each chromosome holds?

View solution| Hide solution

Solution

```
cut -f13 refGene.txt
cut -f13 refGene.txt | grep "^Bcl"
cut -f13 refGene.txt | grep "^Bcl" | sort | uniq | wc -l
cut -f13 refGene.txt | sort | uniq -c | sort -n | tail -1
cut -f13 refGene.txt | sort | uniq -c | sort -n | tail -1
```

```
cut -f9,13 refGene.txt | sort -n | tail -1
cut -f2 refGene.txt  | sed 's/_.*//' | sort | uniq -c
cut -f3 refGene.txt  | sort | uniq -c | sort -n
```

The Bedtools suite

What is bedtools

The bedtools manual says:

> *The BEDTools utilities allow one to address common genomics tasks such as finding feature overlaps and computing coverage. The utilities are largely based on four widely-used file formats: BED, GFF/GTF, VCF, and SAM/BAM. Using BEDTools, one can develop sophisticated pipelines that answer complicated research questions by "streaming" several BEDTools together. The following are examples of common questions that one can address with BEDTools.*
>
> 1. *Intersecting two BED files in search of overlapping features.*
> 2. *Computing coverage for BAM alignments based on genome features.*
> 3. *Merging overlapping features.*
> 4. *Screening for paired-end (PE) overlaps between PE sequences and existing genomic features.*
> 5. *Calculating the depth and breadth of sequence coverage across defined "windows" in a genome.*
> 6. *Screening for overlaps between "split" alignments and genomic features.*

Installing the bedtools suite

Check if the bedtools suite is already installed on your system (type *intersectBed* in a terminal). If bedtools is not available, follow the procedure below to perform installation.

```
wget https://bedtools.googlecode.com/files/BEDTools.v2.16.2.ta
mkdir -p  ~/bin
mv BEDTools.v2.16.2.tar.gz ~/bin
cd ~/bin
tar xvfz BEDTools.v2.16.2.tar.gz
cd BEDTools-Version-2.16.2
make
echo PATH=$PATH:~/bin/BEDTools-Version-2.16.2/bin >> ~/.bashrc
source ~/.bashrc
```

Transforming refGene file into bed format

First we need to transform the refGene.txt file into a bed to use it as input for the bedtools suite. We will do it with *awk*.

```
awk 'BEGIN{FS=OFS="\t"}{print $3,$5,$6,$2"|"$13,0,$4}' refGene
  refGene.bed
```

Upload the resulting bed file into UCSC (top menu, genome > add custom track > choose file > submit). Go to the genome browser to visualize the new track.

Getting promoter regions of each transcript

To obtain promoter regions of each transcript we will first extract the TSS coordinates of each transcript. The TSS will correspond to the txStart or txEnd coordinates depending on wether genes is located on the positive or negative strand respectively (see refSeq table definition).

```
awk 'BEGIN{FS=OFS="\t"}($6=="+"){print $1,$2,$2,$4,$5,$6}' ref
  refGene.TSS.bed
awk 'BEGIN{FS=OFS="\t"}($6=="-"){print $1,$3,$3,$4,$5,$6}' refGene.bed
  >> refGene.TSS.bed
```

Now we will use slopBed to increase the size of each feature (TSS) to define a promoter region around the TSS ([-2000,+500]). Also this can be done with a simple *awk* command, sploBed will restrict the resizing to the size of the chromosome (i.e. no start < 0 and no end > chromosome size). We thus need to download the length of each chromosome (here from UCSC ftp site).

```
wget
   http://hgdownload.cse.ucsc.edu/goldenPath/mm9/database/chrom
gunzip chromInfo.txt.gz
cat chromInfo.txt
```

Now we can run slopBed. We will also reorder features (TSS) according to chromosomes coordinates

```
slopBed -i refGene.TSS.bed -l 2000 -r 500 -s  -g chromInfo.txt
   sortBed > refGene_Prom_2000_500.bed
```

Check the results by loading the refGene_Prom_2000_500.bed as a new user track in the UCSC genome browser.

NB: The RSAT web server also proposes advance tools for retrieving promoter regions (excluding coding regions, repeat elements,...). This tools can also be run using UNIX command lines.

Intersecting promoter regions with predicted transcription factor binding sites

The dataset that will be used thereafter was obtained from the following publication:

- Lepoivre C, Bergon A, Lopez F, Perumal NB, Nguyen C, Imbert J, Puthier D. TranscriptomeBrowser 3.0: introducing a new compendium of molecular interactions and a new visualization tool for the study of gene regulatory networks. BMC Bioinformatics. 2012 Jan 31;13:19.

In this article, the authors scanned vertevrates alignments in search for highly conserved putative transcription factor binding sites (TFBS). The resulting file contains the coordinates of the predicted TFBS (mm9 genome version)

1. Download the conserved_predicted_TFBS_mm9.bed.gz file
2. Use intersectBed to find for each promoter region its associated sets of predicted TFBS.
3. Using *grep* and *sed* select the non-redondant list of targets for transcription factor E2F (represented by M00920:V$E2F_Q6_01 identifier), or ETS (represented by M00771:V$ETS_Q4 identifier), or MEF2A (represented by M00941:V$MEF2_Q6_01 identifier), or NFKB (represented by M00054:V$NFKAPPAB_01 identifier) or NRSF (represented by M01028:V$NRSF_Q4 identifier).

View solution| Hide solution

Solution

```
wget
   http://www.bigre.ulb.ac.be/courses/statistics_bioinformatics
   reg/conserved_predicted_TFBS_mm9.bed
head conserved_predicted_TFBS_mm9.bed
intersectBed -wa -wb -a refGene_Prom_2000_500.bed -b
   conserved_predicted_TFBS_mm9.bed > conservedsites_prom_intersect.bed
grep  -w M00920 conservedsites_prom_intersect.bed   | sed 's/.*|//' |
   cut -f1 | sort | uniq > M00920_targets.txt
grep  -w M00771  conservedsites_prom_intersect.bed  | sed 's/.*|//' |
   cut -f1 | sort | uniq > M00771_targets.txt
grep  -w M00941  conservedsites_prom_intersect.bed  | sed 's/.*|//' |
   cut -f1 | sort | uniq >  M00941_targets.txt
grep  -w M00054  conservedsites_prom_intersect.bed  | sed 's/.*|//' |
   cut -f1 | sort | uniq >  M00054_targets.txt
grep  -w M01028  conservedsites_prom_intersect.bed  | sed 's/.*|//' |
   cut -f1 | sort | uniq >  M01028_targets.txt
```

Functional annotation using Database for Annotation, Visualization and Integrated Discovery

We have seen that each transcription factor is associated with a set of putative targets. What can we learn from these targets? Is the transcription factor associated to genes involved in some particular biological functions?

To answer this question we will use the *Database for Annotation, Visualization and Integrated Discovery* (DAVID). DAVID will try to find whether the selected genes share some particular features, by analyzing their annotations from several sources including Gene Ontology, KEGG pathways, Reactome Pathways, genomic locations, ...

1. Go to the DAVID web site.
2. In the left menu select *Functional annotation.*
3. In the left menu select the upload tab. Paste the list of genes associated to one of the transcription factors.
4. Select *step 2* > *OFFICIAL_GENE_SYMBOL.*
5. Select *step 3* > *Gene List.*
6. Press *Submit List.*
7. The program now displays the "Gene list manager", with a list of species sorted by decreasing number of matching gene names. Select "Mus musculus" and click "Select species".
8. Optionally, you can click on the list under "List manager" and rename if (for example, name it "M00920 E2F mouse").
9. In the "Annotation Summary Results" page, click the button *Functional Annotation Chart*.
10. In the new window, select *option* and ask for *Fold Enrichment*, *Bonferroni*, *Benjamini*, *FDR*, *Fisher Exact*, *LT,PH,PT*

Note: the Fisher test will be explained in the next practical "GO statistics".

What can you guess from the results?

1. Which are the most significant functional classes?
2. How do you interpret their P-value?
3. Is the significance proportional to the fold enrichment?
4. Compare the P-values obtained after correction for multiple testing (Boneroni, Benjamini).
5. Evaluate the results at the bottom of the table: can these associations be considered as significant? Why?

View solution| Hide solution

Gene list

The list of E2F target genes predicted with the matrix M00920 is provided in the following file.

M00920_targets.txt

Interpretation

The figure below shows a snapshot of the top rows of functional annotation chart returned by David.

Sublist	Category	Term	RT	Genes	Count	LT	PH	PT	%	P-Value
☐	KEGG_PATHWAY	Cell cycle	RT		14	26	128	5738	18.7	7.5E-16
☐	GOTERM_BP_FAT	cell cycle	RT		19	59	611	13588	25.3	3.7E-11
☐	SP_PIR_KEYWORDS	cell cycle	RT		16	66	447	17854	21.3	5.0E-11
☐	SP_PIR_KEYWORDS	nucleus	RT		39	66	3808	17854	52.0	7.9E-11
☐	GOTERM_BP_FAT	DNA metabolic process	RT		16	59	421	13588	21.3	1.6E-10
☐	GOTERM_BP_FAT	chromosome organization	RT		14	59	404	13588	18.7	1.1E-8
☐	SP_PIR_KEYWORDS	dna replication	RT		8	66	85	17854	10.7	2.4E-8
☐	SP_PIR_KEYWORDS	acetylation	RT		27	66	2325	17854	36.0	4.9E-8
☐	GOTERM_BP_FAT	DNA unwinding during replication	RT		5	59	11	13588	6.7	9.6E-8

The most significant association for our list of genes is the KEGG pathway "Cell cycle". The second row indicates a significant intersection with another source of annotations: the Gene Ontology term "cell cycle". Although these two functional classes have the same name, they come from different databases, with different annotation strategies. To

avoid redundancy, the interface of DAVID allows you to selectively display some subsets of the Gene Ontology annotations.

Let us analyze in detail the second result row (GOTERM_BP_FAT cell cycle).

- The column "count" gives us the most directly interpretable data: 19 of the 80 genes submitted to David are associated to the term "cell cycle" in the GOTERM_BP_FAT (Gene Ontology, Biological Processes, "FAT" annotations) annotations. Let us denote this number by the symbol $x=19$.
- The column **PT** indicates the total number of genes having some annotation in GOTERM_BP_FAT ($N=13588$). This does not cover the total set of human genes (~25,000). However, since the other genes have no annotation in GOTERM_BP_FAT, we would not be able to test if they do or not belong to the cell cycle annotations. We will thus consider that the "universe" of possibilities is restricted to the N genes having at least some annotation in GOTERM_BP_FAT.
- The column **LT** indicates that, among the 80 genes submitted to DAVID, only $k=59$ have some annotation in GOTERM_BP_FAT. This is the maximal number of genes which could match some biological process in GO annotations.
- The colum **PH** indicates the number of genes associated to the cell cycle in GOTERM_BP_FAT: $m=611$.
