Bioinformatics Workshop – 2018-09-19

Contents

1 MS Word (latest versions after 2013) Hacks

1.1 Thesis writing guidelines

1. Customizing the interface & functions
2. Creating a template
3. Working with a document
4. Editing and sharing documents
5. Citation management
6. Auto proofing
7. Track changes

1.1.1 Creating a template

- Set your page layout (margins, orientation, page numbering)
- Use section breaks and page breaks
- Write first two pages without any formatting
- Format two pages
- Add headers and other text formatting
- Change the numbering levels according to your thesis

2 Bioinformatics Resources

2.1 For R

https://www.r-bloggers.com

https://stackoverflow.com/ (For all programming languages)

https://www.datacamp.com

2.1.1.1 Statistics

https://stats.stackexchange.com/

2.2 Genomics and Bioinformatics

2.2.1 SEQanswers

http://seqanswers.com/

Discuss scientific topics related to genomics

2.2.2 Biostars

https://www.biostars.org/

Focus on bioinformatics, computational genomics and biological data analysis.

2.2.3 Scientific Research communities

https://www.researchgate.net/

https://www.academia.edu

2.2.4 Other online tools for researchers

http://connectedresearchers.com/online-tools-for-researchers/

3 Linux Basics

3.1 awk

AWK, one of the most prominent text-processing utility on GNU/Linux. It is very powerful and uses simple programming language. It can solve complex text processing tasks with a few lines of code. Starting with an overview of AWK, its environment, and workflow, the tutorial proceeds to explain the syntax, variables, operators, arrays, loops, and functions used in AWK. It also covers topics such as output redirection and pretty printing.

https://www.tutorialspoint.com/awk/index.htm

3.2 pipes

A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing. The Unix/Linux systems allow stdout of a command to be connected to stdin of another command. You can make it do so by using the pipe character '|'.

Pipe is used to combine two or more command and in this the output of one command act as input to another command and this command output may act as input to next command and so on. It can also be visualized as a temporary connection between two or more commands/ programs/ processes. The command line programs that do the further processing are referred to as filters.

This direct connection between commands/ programs/ processes allows them to operate simultaneously and permits data to be transferred between them continuously rather than having to pass it through temporary text files or through the display screen.

Pipes are unidirectional i.e data flow from left to right through the pipeline.

Syntax :

```
command_1 | command_2 | command_3 | .... | command_N
```

https://www.geeksforgeeks.org/piping-in-unix-or-linux/

3.3 Regular expressions (Regex)

Regular expressions are a powerful means for pattern matching and string parsing that can be applied in so many instances. With this incredible tool you can

- Validate text input

- Search (and replace) text within a file
- Batch rename files
- Undertake incredibly powerful searches for files
- Interact with servers like Apache
- Test for patterns within strings
- And many more

3.3.1 What makes up regular expressions

There are two types of characters to be found in regular expressions:

- literal characters
- meta characters

Literal characters are standard characters that make up your strings. Every character in this sentence is a literal character. You could use a regular expression to search for each literal character in that string.

Metacharacters are a different beast altogether; they are what give regular expressions their power. With metacharacters, you can do much more than searching for a single character. Metacharacters allow you to search for combinations of strings and much more.

Continue on…

https://www.linux.com/learn/intro-to-linux-/2017/2/introduction-regular-expressions-new-linux-users

https://www.gnu.org/software/sed/manual/html_node/Regular-Expressions.html

3.4 Grep (Global Regular Expression Print)

Grep, a UNIX command and also a utility available for Windows and other operating systems, is used to search one or more files for a given character string or pattern and, if desired, replace the character string with another one.

https://www.digitalocean.com/community/tutorials/using-grep-regular-expressions-to-search-for-text-patterns-in-linux

3.5 String Editor (sed)

https://www.gnu.org/software/sed/manual/html_node/Overview.html#Overview

3.6 Exercise

1. Create a new directory named “basic_linux”
2. Go to the directory
3. Create a text file named “file1.txt”, write something and save
4. Download the file seq1.txt, seq2.txt, sed.ex.txt from
5. Combine seq1.txt and seq2.txt, remove “**:_**” , add the length of the region to a new column sort file by chr and start position and create an output file.

3.7 Summary

`awk` and `sed` are completly different than `grep`. `awk` and `sed` are text processors. Not only do they have the ability to find what you are looking for in text, they have the ability to remove, add and modify the text as well (and much more).

`awk` is mostly used for data extraction and reporting. `sed` is a stream editor Each one of them has its own functionality and specialties.

4 Linux and Genomics

4.1 Bedtools

4.1.1.1 File formats

Bed

https://genome.ucsc.edu/FAQ/FAQformat.html#format1

bedgraph

https://genome.ucsc.edu/goldenpath/help/bedgraph.html

bigwig

https://genome.ucsc.edu/goldenpath/help/bigWig.html

4.1.2 Bedtools Utilities

Install and syntax

http://quinlanlab.org/tutorials/bedtools/bedtools.html

Read about bedtools intersect, merge, closest, subtract and other bedtools utilities

4.1.3 Exercise

Merge overlapping the regions of bed1.txt and create a new file named bed1_merge.bed

Find overlapping regions of bed1_merge.bed bed2.bed

5 R programming

5.1 R basics

https://github.com/jmonlong/HGSS_Rworkshops/blob/master/Intro-Rbasics-2016/HGSS-Workshop-Rintro-2016.pdf

https://drive.google.com/file/d/0BwSxlHpMjRKtY043UVloVUxzNnM/view?usp=sharing

5.1.1 R for genomics

https://github.com/jmonlong/HGSS_Rworkshops/blob/master/Advanced-LargeGenomicsData-2017/HGSS-Rworkshop2016-17-2.pdf