

Bash Tutorial

Polypliod Summer School 2023 10 - 20 July 2023

Introduction to Bash

Bash is a popular command-line shell and scripting language used in Linux and macOS systems. It is an essential skill for bioinformatics students as it provides powerful tools for data manipulation, file processing, and automation. In this tutorial, we will cover the basics of Bash scripting with a focus on bioinformatics-related tasks.

Table of Contents

1. [Getting Started](#)
 - [Opening the Terminal](#)
 - [Basic Commands](#)
2. [Working with Files and Directories](#)
 - [Navigating the Filesystem](#)
 - [Creating and Removing Files/Directories](#)
 - [Copying, Moving, and Renaming Files/Directories](#)
3. [Text Processing](#)
 - [Viewing File Contents](#)
 - [Searching and Filtering Data](#)
 - [Modifying File Contents](#)
4. [Bioinformatics Tasks](#)
 - [Working with FASTA Files](#)
 - [Counting Sequences](#)
 - [Extracting Sequence Information](#)
5. [Shell Scripting](#)
 - [Creating and Running a Script](#)
 - [Passing Arguments to a Script](#)
6. [Conclusion](#)

Getting Started

Opening the Terminal

To start using Bash, open the terminal application on your Linux or macOS system. You can usually find it in the "Utilities" folder or search for it using the system's search function.

Basic Commands

Here are some essential Bash commands to get you started:

- `ls` - List files and directories in the current directory.
- `cd` - Change directory.
- `pwd` - Print the current working directory.

- `mkdir` - Create a new directory.
- `rm` - Remove a file or directory.
- `cp` - Copy files and directories.
- `mv` - Move or rename files and directories.
- `cat` - Concatenate and display file contents.
- `grep` - Search for patterns in files.
- `sed` - Stream editor for modifying file contents.
- `head` - Display the first few lines of a file.
- `tail` - Display the last few lines of a file.

Working with Files and Directories

Navigating the Filesystem

To navigate the filesystem, you can use the `cd` command followed by the directory path. Here are some examples:

- `cd` - Change to the home directory.
- `cd /path/to/directory` - Change to a specific directory.
- `cd ..` - Move up one directory level.
- `cd ~username` - Change to the home directory of a specific user.

Creating and Removing Files/Directories

To create files and directories, you can use the `mkdir` and `touch` commands:

- `mkdir directory_name` - Create a new directory.
- `touch file_name` - Create a new empty file.

To remove files and directories, use the `rm` command:

- `rm file_name` - Remove a file.
- `rm -r directory_name` - Remove a directory and its contents recursively.

Copying, Moving, and Renaming Files/Directories

To copy files or directories, use the `cp` command:

- `cp source_file destination_file` - Copy a file.
- `cp -r source_directory destination_directory` - Copy a directory and its contents recursively.

To move or rename files and directories, use the `mv` command:

- `mv old_name new_name` - Rename a file or directory.
- `mv file_name directory_name` - Move a file to a different directory.

Text Processing

Viewing File Contents

To view the contents of a file, you can use the `cat`, `head`, or `tail` commands:

- `cat file_name` - Display the entire file.
- `head -n 10 file_name` - Display the first 10 lines of a file.
- `tail -n 5 file_name` - Display the last 5 lines of a file.

Searching and Filtering Data

To search for patterns in files, use the `grep` command:

- `grep pattern file_name` - Search for a pattern in a file.
- `grep -i pattern file_name` - Perform a case-insensitive search.
- `grep -r pattern directory_name` - Search for a pattern recursively in a directory.

Modifying File Contents

To modify file contents, you can use the `sed` command:

- `sed 's/pattern/replacement/g' file_name` - Substitute a pattern with a replacement in a file.

Bioinformatics Tasks

Working with FASTA Files

FASTA files are commonly used to store DNA or protein sequences. Here's an example of working with FASTA files using Bash commands:

- Extract sequences from a FASTA file using `grep`:

```
grep -v ">" sequences.fasta > clean_sequences.fasta
```

Counting Sequences

To count the number of sequences in a FASTA file, you can use Bash commands along with `grep` and `wc`:

```
grep -c ">" sequences.fasta
```

Extracting Sequence Information

To extract specific sequence information from a file, you can use tools like `awk` or `cut`. For example, to extract the sequence lengths from a FASTA file:

```
grep -v ">" sequences.fasta | awk '{ print length($0) }'
```

Shell Scripting

Creating and Running a Script

To create a shell script, you can use a text editor to write a series of Bash commands and save the file with a `.sh` extension. For example:

```
#!/bin/bash

# This is a simple script
echo "Hello, World!"
```

Save the script as `hello.sh`. To run the script, use the following command:

```
bash hello.sh
```

Passing Arguments to a Script <a>

You can pass arguments to a shell script by referencing them using the `$` symbol. For example, modify the `hello.sh` script to accept a name argument:

```
#!/bin/bash

# This is a script that greets a person
echo "Hello, $1!"
```

Run the modified script and provide a name argument:

```
bash hello.sh John
```

The output will be: "Hello, John!"

Conclusion

In this tutorial, we covered the basics of Bash scripting for bioinformatics students. You learned how to navigate the filesystem, work with files and directories, process text, perform common bioinformatics tasks, and create shell scripts. Bash provides a wide range of tools and possibilities for data manipulation and automation, empowering you in your bioinformatics work.

Explore further Bash resources, practice writing scripts, and experiment with different commands to expand your knowledge and efficiency in bioinformatics tasks.

Happy scripting!