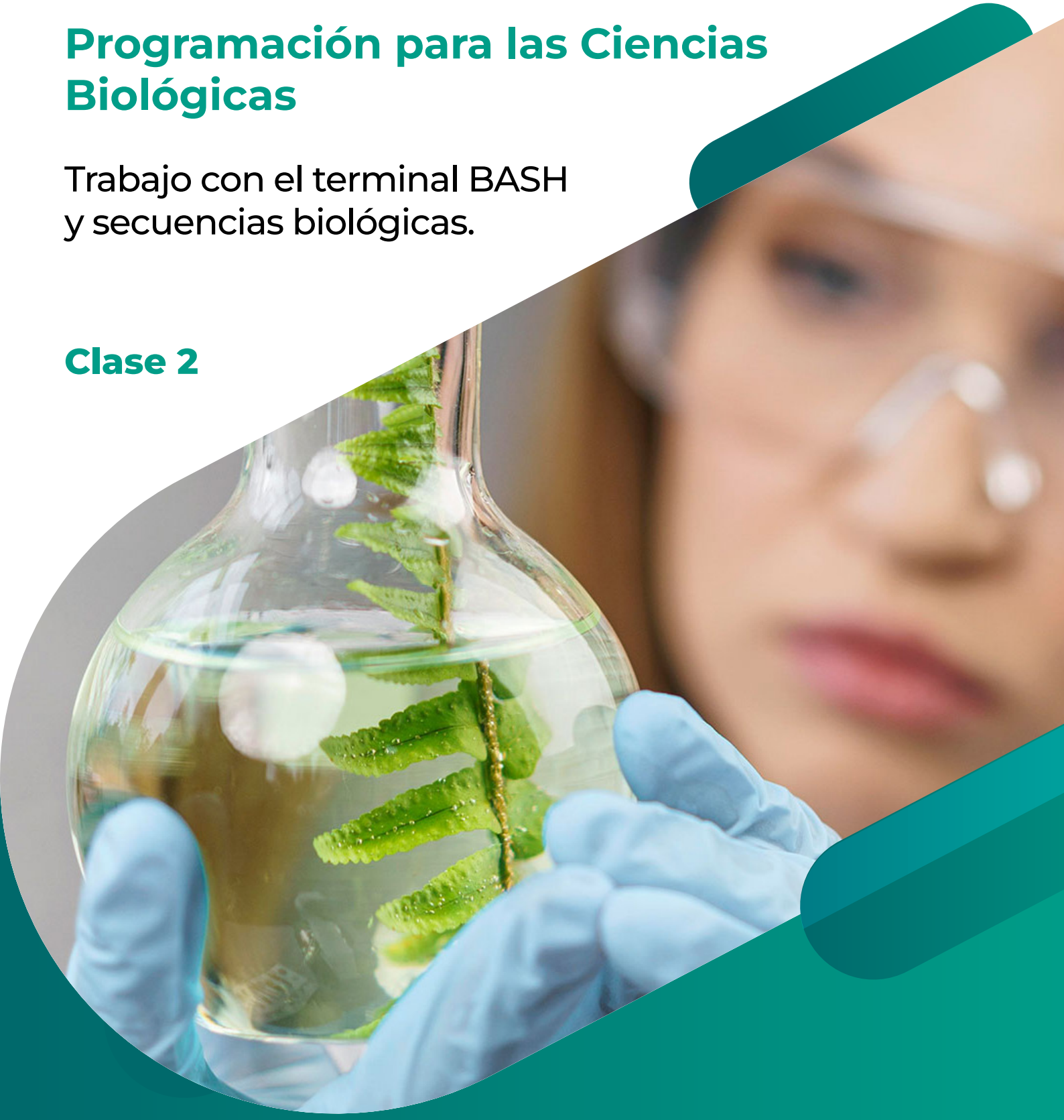
Programación para las Ciencias Biológicas

Trabajo con el terminal BASH y secuencias biológicas.



Clase 2

MAESTRÍA EN MAESTRÍA EN BIOLOGÍA COMPUTACIONAL

La excelencia no se improvisa

FORMULARIO DE PRODUCCIÓN DE CONTENIDOS

NOMBRE DE LA ASIGNATURA: Programación para las ciencias biológicas
NOMBRE COMPLETO DEL DOCENTE CONTENIDISTA: Escobar Terán Charles Edisson

Contenido:

Cada clase deberá mostrar temas y subtemas contenidas en 8 páginas y un **rango de 2800 a 3000 palabras,** de autoría del contenidista contratado estar de acuerdo con el syllabus de la asignatura y la distribución de temas y subtemas determinados en el Formulario de Estructuración de la Asignatura (Formulario 1). En este contenido puede agregar gráficos e imágenes debidamente referenciados de acuerdo a la normativa APA versión 7.	Cumple
En cada clase, deberán presentarse, como mínimo, **2 (dos) enlaces** relacionados con los temas y subtemas complementando al texto redactado. Para cada enlace deberá haber una breve descripción de lo que se está abordando, presentada de forma dialogada **en el contenido**. Estas descripciones y enlaces deben estar destacados en verde. Los enlaces pueden ser a:	
- Videos [Se incluyen anuncios, documentales, fragmentos de películas, entrevistas u otros tipos que puedan ser utilizados como material didáctico complementario en la disciplina].	https://seracademia.com/academia/Bio_Info/M2/Videos/Cuaderno1_BASH.mp4 https://seracademia.com/academia/Bio_Info/M2/Videos/Cuaderno2_BASH.mp4
- Sitios web [Enlaces de sitios que contengan artículos, entrevistas, investigaciones u otro material que pueda ser utilizado como material didáctico complementario en la disciplina].	Cumple
- Otros [Pueden ser archivos adjuntos por el responsable del contenido y enviados junto con el contenido de la disciplina].	Cuaderno jupyter: Cuaderno1_BASH_Apellido_Nombre.ipynb Cuaderno2_BASH_Apellido_Nombre.ipynb

3. Al final de cada clase deberán presentarse, obligatoriamente, todas las ***REFERENCIAS CITADAS*** en el texto del contenido, de acuerdo con la normativa de la American Psychological Association (APA) versión 7 o con la información mínima para ser adecuadas a las normas. Las citas autor-fecha utilizadas a lo largo del contenido deben estar destacadas en azul.

4. Dentro del contenido de la clase subraye **al menos 2 términos** de los temas abordados y, al final coloque las definiciones correspondientes.

5. *RECURSOS DE PROFUNDIZACIÓN:* Son recursos de creación propia del contenidista, aquí podrá explicar conceptos y definiciones que requieran una **explicación más detallada** usando vid-

eos e infografías, tablas, estudios de caso, resolución de actividades, presentación de resolución de situaciones reales u otros recursos que ayuden a la comprensión del contenido. Se requiere la entrega de 1 a 3 recursos por clase.

DESARROLLO

1. INTRODUCCIÓN

Bienvenidos a la segunda clase del curso "Programación para las Ciencias Biológicas". En esta sesión nos adentraremos en el uso práctico del terminal BASH, una herramienta esencial para el análisis de secuencias biológicas. A medida que la bioinformática avanza, se hace indispensable el manejo eficiente de datos biológicos mediante herramientas de línea de comandos. La habilidad para navegar, manipular y analizar secuencias de ADN, ARN y otros tipos de datos biológicos utilizando BASH es una competencia fundamental para cualquier bioinformático o científico de datos.

En esta clase, aprenderemos los comandos esenciales de BASH que facilitarán nuestro trabajo con archivos biológicos. Comenzaremos con una introducción al terminal, donde exploraremos los comandos básicos para verificar el sistema operativo y navegar por directorios, como **lsb_release**, **uname -a**, **pwd** y **cd**. A través de ejercicios prácticos, adquiriremos experiencia en la gestión y manipulación de archivos con comandos como **mkdir**, **rmdir**, **echo**, **cat**, etc.

Una parte clave de la clase será la introducción a herramientas de procesamiento de texto, como **grep**, **wc**, **cut**, **sort** y **awk**, que nos permitirán analizar, filtrar y transformar grandes volúmenes de datos biológicos, como secuencias de ADN y ARN.

En la segunda parte de la clase, nos enfocaremos en el uso avanzado de BASH para el procesamiento de datos biológicos. Aprenderemos a trabajar con tuberías y redirección para concatenar comandos y redirigir flujos de datos, una habilidad fundamental para el análisis eficiente de grandes secuencias. A través de ejemplos prácticos, entenderemos cómo **awk** puede ayudarnos a extraer, filtrar y analizar datos de manera eficiente, especialmente cuando trabajamos con archivos de secuencias en formatos como FASTA o similares. Además, introduciremos **bioawk**, una extensión de **awk** específica para el análisis de secuencias biológicas, que nos permitirá aplicar filtros más avanzados a nuestros datos.

Esta clase está diseñada para ofrecer una comprensión sólida de cómo el uso de línea de comandos puede optimizar los flujos de trabajo en bioinformática, permitiendo a los científicos trabajar con datos biológicos de manera más eficiente y precisa.

¡Veamos como la programación en BASH se convierte en una poderosa herramienta en el análisis de datos biológicos!

Clase 2:

RDA1. Desarrollar algoritmos con lenguaje de programación estructurado de línea de comandos para secuencias biológicas.

Trabajo con el terminal BASH y secuencias biológicas.

1.3.1. Comandos esenciales para bioinformática

1.3.1.1 .Práctica y experiencia con comandos básicos del sistema:

1.3.1.1.1. lsb_release, uname -a: Verificación del sistema operativo y kernel.

1.3.1.1.2. host, pwd, cd: Navegación en directorios.

1.3.1.1.3. mkdir, rmdir, echo, cat, more, less: Manejo básico de archivos.

1.3.1.1.4. grep, wc, cut, sort, awk: Manipulación de texto y análisis de archivos biológicos.

1.3.1.1.5. Prácticas enfocadas en trabajar con archivos de datos biológicos, como secuencias de ADN y ARN.

1.4. Uso avanzado del terminal BASH

1.4.1. Tuberías, filtros y procesamiento de datos biológicos

1.4.1.1. Tuberías y redirección en el análisis de secuencias

1.4.1.1. Uso de tuberías para concatenar comandos y flujos de datos bioinformáticos. *Redirección de salida y entrada en el análisis de archivos biológicos.

1.4.2. Procesamiento de datos biológicos con AWK y herramientas de línea de comandos

1.4.2.1. Manipulación de secuencias biológicas usando AWK para extraer, filtrar y analizar datos de forma eficiente.

1.4.2.2. Ejemplos prácticos de uso de AWK en archivos de secuencias de ADN/ARN.

1.4.2.3. Introducción al uso de bioawk para el análisis de secuencias FASTA

Referencias citadas en la Clase 2.

Videos:

1. Uso del terminal BASH

https://seracademia.com/academia/Bio_Info/M2/Videos/Cuaderno1_BASH.mp4

https://seracademia.com/academia/Bio_Info/M2/Videos/Cuaderno2_BASH.mp4

2. Uso avanzado del terminal BASH

Cuadernos de trabajo Jupyter

1. Cuaderno1_BASH_Apellido_Nombre.ipynb
2. Cuaderno2_BASH_Apellido_Nombre.ipynb

Referencias Bibliográficas

1. General Introduction to AWK® (Funcación GNU, 2024). Disponible en: <https://www.gnu.org/software/gawk/manual/gawk.html>

2. awk(1p) — Linux manual page, Disponible en <https://man7.org/linux/man-pages/man1/awk.1p.html>

3. Proyecto bioawk®, disponible en <https://github.com/lh3/bioawk>

4. TutorialsPoint. (2021). Bash Scripting Tutorial.

Disponible en <https://www.tutorialspoint.com/unix/unix-shell.htm>

Definición de los términos citados en la Clase 2.

1. **lsb_release**: Comando que muestra la información del sistema operativo instalado en una máquina, como la distribución y la versión.
2. **uname -a**: Comando que proporciona información sobre el kernel de Linux y detalles del sistema operativo.
3. **host**: Comando que permite obtener la información sobre un nombre de dominio o una dirección IP, como su dirección IP asociada.
4. **pwd**: Comando que muestra el directorio de trabajo actual en la línea de comandos.
5. **cd**: Comando que se utiliza para cambiar el directorio de trabajo en la terminal.
6. **grep**: Comando utilizado para buscar un patrón de texto específico dentro de archivos o cadenas de texto.
7. **wc**: Comando que cuenta las líneas, palabras o caracteres en un archivo o entrada de texto.
8. **cut**: Comando utilizado para dividir y extraer secciones específicas de un archivo o entrada de texto.
9. **sort**: Comando que ordena las líneas de un archivo o entrada de texto en orden ascendente o descendente.
10. **awk**: Lenguaje de programación y comando que se utiliza para manipular y procesar archivos de texto, especialmente aquellos que tienen una estructura tabular.
11. **bioawk**: Versión especializada de AWK diseñada para trabajar con archivos biológicos, como archivos de secuencias FASTA y GFF.
12. **tubería**: Caracter (|) que permite encadenar varios comandos, de modo que la salida de un comando se convierte en la entrada de otro.
13. **redireccionamiento**: Técnica que permite cambiar el flujo de entrada o salida de un comando, dirigiendo los resultados a un archivo o tomando la entrada de un archivo en lugar de la terminal.

Profundización Clase 2.

Clase 2:

RDA1: Desarrollar algoritmos con lenguaje de programación estructurado de línea de comandos para secuencias biológicas.

Trabajo con el terminal BASH y secuencias biológicas.

1.3. Uso del terminal BASH

El terminal BASH (Bourne Again Shell) es una herramienta poderosa que permite interactuar con el sistema operativo mediante comandos de texto. En bioinformática, el uso eficiente de BASH y sus comandos es esencial para manejar, procesar y analizar grandes volúmenes de datos biológicos. A través del terminal, podemos realizar tareas como la navegación de directorios, manipulación de archivos y análisis de datos de secuencias biológicas sin necesidad de interfaces gráficas. En esta sección, cubriremos los comandos esenciales que todo biólogo computacional debe conocer.

1.3.1. Comandos esenciales para bioinformática

El uso del terminal BASH es fundamental en bioinformática, ya que permite gestionar, procesar y analizar datos biológicos de manera eficiente. A continuación, exploraremos una serie de comandos básicos y herramientas para manejar datos de secuencias biológicas. Estos comandos forman la base de nuestra interacción con los datos y nos permiten trabajar en proyectos de investigación, análisis de genomas, y más.

1.3.1.1 .Práctica y experiencia con comandos básicos del sistema:

1.3.1.1.1. **lsb_release, uname -a:** Verificación del sistema operativo y kernel.

Antes de comenzar a trabajar con herramientas de bioinformática, es importante conocer el sistema operativo en el que estamos trabajando. Los comandos lsb_release y uname -a nos proporcionan información esencial sobre el sistema y el kernel en uso.

- lsb_release: Este comando nos da información sobre la distribución de Linux instalada, incluyendo la versión y el nombre del sistema operativo.

```
jupyter@Debian12-server:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 12 (bookworm)
Release:        12
Codename:       bookworm
```

- uname -a: Este comando nos proporciona información sobre el sistema operativo y el kernel, incluyendo la versión del kernel de Linux y la arquitectura.

```
jupyter@Debian12-server:~$ uname -a
Linux Debian12-server 6.1.0-10-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.38-2 (2023-07-27) x86_64 GNU/Linux
```

Estos comandos son útiles para conocer las características del sistema antes de ejecutar herramientas bioinformáticas, ya que algunas herramientas pueden depender de versiones específicas del sistema operativo o del kernel.

- hostname: sin argumentos mostrará el nombre del host actual, mientras que con un argumento específico, permitirá establecer el nuevo nombre del host localmente.

```
jupyter@Debian12-server:~$ hostname
Debian12-server
```

1.3.1.1.2. **host, pwd, cd**: Navegación en directorios.

La navegación eficiente por los directorios y la manipulación de rutas es esencial cuando se trabaja con grandes volúmenes de datos biológicos. Los comandos pwd, cd y host nos permiten gestionar nuestra ubicación en el sistema de archivos y realizar tareas relacionadas.

- host: Resuelve nombres de dominio a direcciones IP. Este comando puede ser útil para interactuar con servidores de datos o consultar servidores web de bases de datos biológicas.

```
jupyter@Debian12-server:~$ host www.ncbi.nlm.nih.gov
www.ncbi.nlm.nih.gov is an alias for www.wip.ncbi.nlm.nih.gov.
www.wip.ncbi.nlm.nih.gov has address 130.14.29.110
www.wip.ncbi.nlm.nih.gov has IPv6 address 2607:f220:41e:4290::110
```

- pwd: Imprime el directorio de trabajo actual. Este comando es esencial para saber en qué carpeta estamos trabajando en todo momento.

```
jupyter@Debian12-server:~$ pwd
/home/jupyter
```

- cd: Cambia el directorio de trabajo. Nos permite movernos entre diferentes carpetas de nuestro sistema de archivos.

```
jupyter@Debian12-server:~$ cd /home/jupyter/Scripts
jupyter@Debian12-server:~/Scripts$
```

Estos comandos nos permiten orientarnos en el sistema de archivos y ubicar los datos que vamos a analizar, facilitando la organización de proyectos de bioinformática.

1.3.1.1.3. **mkdir, rmdir, echo, cat, more, less**: Manejo básico de archivos.

La manipulación de archivos es una de las tareas más comunes en bioinformática, especialmente al trabajar con datos en formato de texto. Estos comandos nos permiten crear, eliminar, visualizar y manipular archivos.

- mkdir: Crea un nuevo directorio.
    - Ejemplo:

      mkdir datos

- rmdir: Elimina un directorio vacío.
    - Ejemplo:

      rmdir datos

- echo: Imprime un mensaje en la terminal o redirige texto a un archivo.
    - Ejemplo:

      echo "ATGCGTAC" > secuencia.txt

- cat: Muestra el contenido de un archivo en la terminal. Es muy útil para ver el contenido de archivos de texto y secuencias biológicas.
    - Ejemplo:

      cat sampledata/t-coffee/sample_dnaseq1.fasta

Salida:

```
>seq1
GTGAACCTGCGGAAGGATCATTATCGAGTCCAATCCAGCTCATTTCTGGCTGTGCATTCACTTGCACGCCGGGGCGAGTT
GGCTCTTGAACTTCTCACACCTGTGCACCTTCTCGTGCCTTCTCCGTCGTCGTCCTTTGTTGGGTGACTCCACCGCTCGC
>seq2
TCATTTAGAGGAAGTAAAAGTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTATCGAAGAGCAACTCTTC
ATACACCCTGTGAACTGTCGTCTTAGGACGCAATATAAACTCTAGTGTTGTCAATGAAAGTGTTATACCATAACAAGCAT
>seq3
TCATTTAGAGGAAGTAAAAGTCGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTATTGAAACTTCTAAGTTT
TAACACACCTGTGAACCTGTACATTCCCTTCGGGGTCTTACACACAAACTCTAGTGTCGTCTATGTATGTCTAGTTATAT
```

- more y less: Permiten visualizar el contenido de archivos grandes de manera paginada, lo que es útil cuando trabajamos con archivos de secuencias de ADN/ARN extensas.
    - Ejemplo:

      less sampledata/t-coffee/sproteases_large.fasta

      o

      more sampledata/t-coffee/sproteases_large.fasta

Esto le permitirá visualizar el contenido de este archivo y navegador en el con las teclas "flecha arriba" y "flecha abajo" del teclado. Para salir de less o more, presione la letra Q o "Ctrl+c"

Estos comandos nos ayudan a crear la estructura de directorios necesaria para organizar nuestros datos y permiten una fácil visualización de archivos de texto, como las secuencias biológicas.

1.3.1.1.4. **grep, wc, cut, sort, awk:** Manipulación de texto y análisis de archivos biológicos.

Los comandos de procesamiento de texto son fundamentales cuando se trabaja con archivos de secuencias biológicas, ya que nos permiten buscar, contar, extraer y ordenar datos de manera eficiente.

- grep: Permite buscar patrones de texto dentro de un archivo. Es ideal para encontrar secuencias o características específicas dentro de grandes archivos de secuencias biológicas.
    - Ejemplo:

      grep "ATG" sampledata/t-coffee/sproteases_large.fasta

- wc: Cuenta las líneas, palabras y caracteres en un archivo. Es útil para obtener estadísticas rápidas sobre archivos de secuencias.
    - Ejemplo:

      wc -l sampledata/t-coffee/sproteases_large.fasta

muestra el número de líneas de ese archivo

- cut: Extrae columnas específicas de un archivo de texto, como archivos CSV o tabulados. Es útil para extraer información de archivos que contienen múltiples campos.
    - Ejemplo:

      cut -d " " -f1 sampledata/t-coffee/struc1.pdb

muestra el primer campo de ese archivo, usando como delimitador de campos el carácter espacio " "

- sort: Ordena las líneas de un archivo. Puede ser usado para ordenar secuencias de manera alfabética o numérica.
    - Ejemplo:

      sort sampledata/t-coffee/struc1.pdb

- awk: Un lenguaje de programación para procesar texto, que permite realizar filtrado, extracción y análisis avanzados. Es especialmente útil para trabajar con archivos en formatos como GFF o TSV.
    - Ejemplo:

awk '{print $1, $3}' sampledata/t-coffee/struc1.pdb

Estos comandos de manipulación de texto nos permiten analizar, transformar y filtrar grandes conjuntos de datos biológicos de manera eficiente.

1.3.1.1.5. Prácticas enfocadas en trabajar con archivos de datos biológicos, como secuencias de ADN y ARN.

El objetivo principal de esta sección es que los estudiantes adquieran la habilidad para traba-

jar con archivos de secuencias biológicas utilizando los comandos mencionados. En la práctica del cuaderno de trabajo "**Cuaderno1_BASH_Apellido_Nombre.ipynb**", los estudiantes utilizarán, entre otros, **grep** para buscar subsecuencias específicas dentro de archivos FASTA, **wc** para contar las secuencias en un archivo de texto, y **tr** o **fold** para extraer y filtrar información de archivos.

Durante estas prácticas, se trabajará con archivos de secuencias biológicas reales, y los estudiantes aprenderán a aplicar comandos de BASH para realizar tareas comunes de bioinformática, como la búsqueda de patrones, la manipulación de datos de secuencias, y la organización de grandes conjuntos de datos.

1.4. Uso avanzado del terminal BASH

En bioinformática, el manejo y análisis de grandes volúmenes de datos biológicos requiere de herramientas y técnicas avanzadas que nos permitan procesar secuencias de manera eficiente. El terminal BASH es una de las herramientas más poderosas para realizar este tipo de tareas. En esta sección, exploraremos cómo usar tuberías, redirección y herramientas como AWK y bioawk para realizar procesamiento de datos biológicos.

1.4.1. Tuberías, filtros y procesamiento de datos biológicos

Tuberías (|): En BASH, la tubería permite conectar dos comandos de tal forma que la salida del primer comando se convierte en la entrada del siguiente. Esta técnica es útil para encadenar varias operaciones y crear flujos de trabajo complejos y eficientes.

- **Ejemplo**: Supongamos que tenemos un archivo de secuencias FASTA y queremos contar cuántas secuencias contienen un patrón específico, como "ATG". Podemos usar grep para buscar el patrón y luego contar las coincidencias con wc -l:

```
grep "ATG" sampledata/t-coffee/sproteases_large.fasta | wc -l
```

En este caso, grep busca todas las líneas que contienen "ATG" en el archivo sampledata/t-coffee/sproteases_large.fasta , y luego wc -l cuenta cuántas líneas fueron encontradas.

- **Redirección de salida (>)**: Permite redirigir la salida de un comando a un archivo, en lugar de mostrarla en la terminal. Esto es útil cuando necesitamos guardar los resultados de un análisis o filtrado para su posterior análisis.
- **Ejemplo**: Si queremos guardar las secuencias que contienen un patrón específico en un nuevo archivo, podemos usar:

```
grep "ATG" sampledata/t-coffee/sproteases_large.fasta  > secuencias_atg.fasta
```

Este comando buscará todas las secuencias que contienen "ATG" en sampledata/t-coffee/sproteases_large.fasta y las almacenará en el archivo secuencias_atg.fasta.

Las tuberías y los filtros son conceptos fundamentales en BASH. Nos permiten tomar la sali-

da de un comando y pasarla como entrada a otro comando, creando flujos de trabajo eficientes y fáciles de gestionar. Estas técnicas son esenciales en bioinformática para manejar y procesar datos biológicos de manera eficiente.

Redirección de entrada (<): Permite usar un archivo como entrada para un comando. Esto es útil cuando queremos que un comando procese datos de un archivo directamente, en lugar de escribirlos manualmente.

- **Ejemplo**: Supongamos que tenemos un archivo llamado datos.txt que contiene una lista de secuencias, y queremos procesarlo con awk:

```
awk '{print $1}' < datos.txt
```

Este comando tomará el contenido del archivo datos.txt y pasará su contenido a awk para imprimir la primera columna de cada línea.

1.4.1.1. Uso de tuberías para concatenar comandos y flujos de datos bioinformáticos.

El uso de tuberías es muy común en bioinformática, especialmente cuando se trabaja con datos grandes o complejos. Los siguientes ejemplos ilustran cómo combinar comandos de BASH para crear flujos de trabajo complejos en el análisis de datos biológicos:

- **Ejemplo de procesamiento de archivos FASTA**: Si tenemos un archivo FASTA y queremos extraer todas las secuencias que comienzan con "ATG", podemos utilizar un flujo de trabajo que combine varios comandos:

```
grep "^>.*ATG" secuencias.fasta | cut -d"" -f1 > secuencias_con_atg.fasta
```

En este ejemplo, grep busca las secuencias que empiezan con "ATG" en el archivo FASTA, y luego cut extrae el primer campo de cada línea (que puede ser el identificador de la secuencia). Finalmente, la salida se redirige a un nuevo archivo secuencias_con_atg.fasta.

- **Ejemplo de procesamiento de archivos de secuencias**: Para contar cuántas secuencias tienen una longitud superior a 100 nucleótidos en un archivo FASTA:

```
grep -v "^>" secuencias.fasta | awk 'length($0) > 100' | wc -l
```

Este flujo de trabajo usa grep para eliminar las líneas de encabezado (que comienzan con ">"), luego usa awk para filtrar las secuencias cuya longitud es mayor a 100 nucleótidos, y finalmente, wc -l cuenta cuántas secuencias cumplen con esa condición.

1.4.2. Procesamiento de datos biológicos con AWK y herramientas de línea de comandos

AWK es una herramienta extremadamente poderosa para el procesamiento de texto en BASH. Es particularmente útil cuando se trabaja con archivos de texto grandes o datos

tabulados, como archivos GFF, VCF, o TSV. AWK permite realizar operaciones complejas de filtrado, extracción, y análisis de datos de manera eficiente.

1.4.2.1. Manipulación de secuencias biológicas usando AWK para extraer, filtrar y analizar datos de forma eficiente.

Extraer columnas específicas de un archivo de datos tabulados como un archivo GFF, que contiene información sobre genes, exones y otras características genómicas.

- Ejemplo:

```
awk '{print $1, $3, $4, $5}' archivo.gff
```

Este comando imprimirá la primera columna (ID de secuencia), la tercera columna (tipo de característica), y las columnas cuarta y quinta (posiciones de inicio y fin) de cada línea del archivo archivo.gff.

1.4.2.2. Ejemplos prácticos de uso de AWK en archivos de secuencias de ADN/ARN.

Los archivos de secuencias de ADN y ARN, a menudo en formato FASTA, pueden analizarse con AWK de diversas formas:

Contar el número de secuencias en un archivo FASTA:

```
awk '/^>/ {count++} END {print count}' archivo.fasta
```

Este comando cuenta cuántas secuencias están presentes en el archivo, basándose en el hecho de que cada encabezado de secuencia en FASTA comienza con un ">".

Extraer solo las secuencias de ADN (eliminando encabezados) y concatenarlas en un solo archivo:

```
awk '!/^>/ {print $0}' archivo.fasta > secuencias_sin_encabezado.fasta
```

Esto elimina las líneas de encabezado (que comienzan con ">") y guarda solo las secuencias de ADN en el archivo de salida.

1.4.2.3. Introducción al uso de bioawk para el análisis de secuencias FASTA y GFF.

bioawk es una versión especializada de AWK, diseñada específicamente para trabajar con archivos biológicos como archivos FASTA, GFF, y VCF. Este programa extiende AWK con funciones adicionales para facilitar el manejo de datos biológicos.

- **Ejemplo de análisis de secuencias FASTA** con bioawk:

  ```
  bioawk -c fastx '{print $name, $seq}' archivo.fasta
  ```

  Este comando usa el modo fastx de bioawk para extraer y mostrar el nombre y la secuencia de cada entrada en un archivo FASTA.

- **Ejemplo de análisis de archivos GFF** con bioawk:

  ```
  bioawk -c gff '{print $seqname, $feature, $start, $end}' archivo.gff
  ```

  Este comando usa bioawk para extraer el nombre de la secuencia, el tipo de característica, y las posiciones de inicio y fin de cada línea en un archivo GFF.

  Estos comandos de BASH se utilizarán a lo largo del cuaderno de trabajo "Cuaderno2_BASH_Apellido_Nombre.ipynb". Este cuaderno es una herramienta práctica que permitirá aplicar estos conceptos para trabajar con datos biológicos de manera eficiente utilizando el terminal BASH.

Resumen

En esta clase, se abordaron los conceptos fundamentales de **BASH**, el terminal de línea de comandos de Linux, y su aplicación en el análisis de datos biológicos. Se cubrió una variedad de herramientas y comandos esenciales para la bioinformática que permiten a los estudiantes trabajar de manera eficiente con secuencias biológicas, como ADN, ARN y proteínas.

1. **Uso del terminal BASH**: Comenzamos aprendiendo sobre los comandos del sistema que permiten navegar por el sistema operativo y gestionar archivos. Comandos como pwd, cd, lsb_release y uname -a fueron utilizados para explorar el entorno de trabajo, verificar el sistema y cambiar entre directorios. Además, se practicaron comandos como mkdir, rmdir, echo, cat, more, y less para manejar y visualizar archivos.
2. **Manipulación de datos biológicos**: Se introdujeron herramientas como grep, wc, cut, sort y awk que permiten manipular archivos de texto, extraer información relevante de secuencias biológicas y realizar análisis estadísticos simples. El uso de estos comandos facilita la filtración, ordenación y procesamiento de datos genómicos y biológicos.
3. **Uso avanzado del terminal BASH**: La clase también cubrió el uso de técnicas más avanzadas como **tuberías** y **redireccionamiento**. Estas herramientas permiten encadenar comandos y gestionar el flujo de datos entre ellos, lo que es esencial para crear flujos de trabajo eficientes en bioinformática. Se explicó cómo usar tuberías (|) para pasar la salida de un comando como entrada para otro y cómo redirigir la salida o la entrada de datos a archivos.
4. **Procesamiento con AWK y bioawk**: Se profundizó en el uso de **AWK**, una herramienta poderosa para procesar y manipular secuencias biológicas, y **bioawk**, una versión especializada de AWK que está optimizada para trabajar con formatos de datos biológicos como FASTA y similares. Se conoció como extraer, filtrar y analizar datos de secuencias de manera eficiente utilizando estas herramientas.

PUCE Virtual

La excelencia no se improvisa

síguenos

MBC - 12 - 241