

Práctica llamado de bases con Guppy

Para este curso, vamos a usar un set de datos que corresponde a archivos FAST5 generados en una corrida de Oxford Nanopore Technologies (ONT) usando el protocolo ARTIC para secuenciar genomas de coronavirus. El objetivo es, con estos datos, realizar todo el protocolo bioinformático usado en la vigilancia genómica de las variantes de SARS-CoV-2.

El primer paso es realizar el llamado de bases, es decir, pasar de registros de cambios de voltaje a través del tiempo (FAST5), generados por los instrumentos de ONT, a secuencias de aminoácidos con información de la calidad por cada base (FASTQ).

Este proceso puede realizarse en tiempo real, a medida que vamos secuenciando, pero solo en los equipos MinION MK1C, GridION y PromethION, que realizan el llamado de bases localmente. Es necesario tener en cuenta que, dependiendo de la precisión en que queramos determinar las bases y el instrumento empleado, los tiempos de procesamiento pueden ser bastante largos.

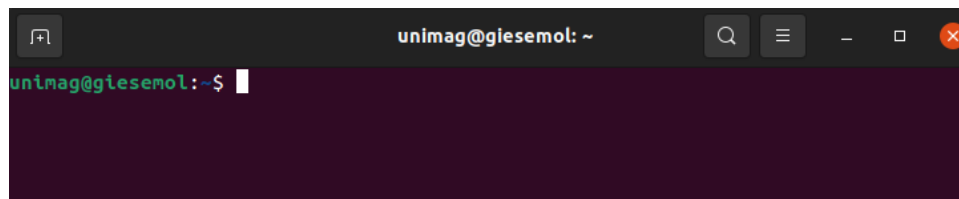
Se recomienda, si es posible, tener computadores o servidores de alto rendimiento con una tarjeta de video NVIDIA de última generación, compatible con el programa CUDA y con una memoria de tarjeta de más de 8 GB. Esto nos permite usar algoritmos exigentes muy precisos con un tiempo de ejecución más corto.

Es por esto por lo que la primera parte del taller corresponde a realizar el llamado de bases local con el programa de ONT, Guppy.

1. Preparación material del curso

El material del curso va a estar disponible en la página de GitHub <https://github.com/dfbautista/Curso-Epidemiologia-Genomica-Practica-Bioinformatica>. Este es el repositorio en donde vamos a tener las guías del curso.

Para descargarlo en nuestro equipo, entramos a la terminal:

A screenshot of a terminal window. The title bar at the top reads 'unimag@glesemol: ~'. Below the title bar, the terminal shows a green prompt 'unimag@glesemol:~\$' followed by a white cursor. The background of the terminal is dark purple.

El comando `cd` nos sirve para cambiar la ubicación en donde nos encontramos. Si no especificamos a dónde nos queremos dirigir, nos lleva a nuestro directorio de inicio. Usamos el comando `git clone` para descargar a nuestra carpeta de inicio el set de datos:

```
git clone https://github.com/dfbautista/Curso-Epidemiologia-Genomica-Practica-Bioinformatica.git
```

Una vez terminado podemos explorar el set de datos usando el comando `cd`:

```
cd Curso-Epidemiologia-Genomica-Practica-Bioinformatica/
```

Con el comando `ls -lh` podemos ver los elementos presentes en el directorio actual, además de su peso y permisos. Los archivos FAST5 que vamos a usar en la práctica y el programa Guppy lo vamos a descargar desde este link:

<https://mega.nz/folder/zkUw2TTT#avTaSv9CrzszIGxXOAgugw>

Si no funciona usar:

<https://drive.google.com/drive/u/0/folders/1Ijt75AIpOP-UBRIpESwAr3HIQL8bOzn>

```
unimag@glesmol:~$ ls -lh
total 3,3G
drwxrwxr-x 3 unimag unimag 4,0K nov 10 14:38 Curso-Epidemiologia-Genomica-Practica-Bioinformatica
drwxrwxr-x 2 unimag unimag 4,0K oct 29 14:01 Data_Edison
drwxr-xr-x 2 unimag unimag 4,0K oct 20 09:43 Desktop
drwxr-xr-x 3 unimag unimag 4,0K oct 29 16:36 Documents
drwxr-xr-x 4 unimag unimag 4,0K oct 29 16:29 Downloads
drwxr-xr-x 2 unimag unimag 4,0K oct 29 14:01 Music
-rw-rw-r-- 1 unimag unimag 634M nov 10 10:20 ont-guppy-cpu_5.0.16_linux64.tar.gz
drwxr-xr-x 2 unimag unimag 4,0K nov 10 13:24 Pictures
drwxr-xr-x 2 unimag unimag 4,0K oct 29 14:01 Public
drwxrwxr-x 4 unimag unimag 4,0K oct 29 14:01 RADProc
-rw-rw-r-- 1 unimag unimag 2,7G nov 10 14:19 Set_fast5-002.zip
drwx----- 4 unimag unimag 4,0K oct 29 14:01 snap
drwxr-xr-x 2 unimag unimag 4,0K oct 29 14:01 Templates
drwxr-xr-x 3 unimag unimag 4,0K oct 29 16:36 Videos
```

Una vez descargado, podemos descomprimir los archivos. En la carpeta `Set_datos` se encuentran el software Guppy (`ont-guppy-cpu_5.0.16_linux64.tar.gz`) y los archivos FAST5 (`Set_fast5.zip`). Para el primero usamos el programa `tar`:

```
tar -zxvf ont-guppy-cpu_5.0.16_linux64.tar.gz
```

Para los archivos es necesario usar:

```
tar -xf Set_fast5.tar.xz
```

Podemos revisar los contenidos de los archivos con `ls` y el nombre de la carpeta. Si no conocemos el uso de un comando podemos preceder el nombre de éste con el comando `man`, p.ej. `man tar`. Esto nos muestra el manual del comando, con todas las opciones disponibles.

2. Uso de Guppy

Antes de empezar a realizar el llamado de bases, es necesario crear una carpeta para guardar los datos. Usando el comando `mkdir` cree una carpeta llamada `Llamado_bases`. En este lugar se van a guardar las secuencias que procese Guppy como secuencias nucleotídicas en un formato FASTQ. Cada una de las secuencias serán guardadas en la carpeta correspondiente a los barcode empleados.

Guppy ofrece diferentes aplicaciones dentro de sí mismo. Para listar todos los programas disponibles utilice el comando `ls` en la carpeta `ont-guppy-cpu/bin/`

Nuestro programa de interés para este curso es `guppy_basecaller`. Para ejecutar el programa y ver todas las opciones que ofrece, escriba la ruta completa y presione enter:

```
ont-guppy-cpu/bin/guppy_basecaller
```

Esto nos sirve para saber si el programa está funcionando y listar todos los argumentos que tiene `guppy_basecaller`. Recuerde en donde se encuentra ubicado cuando corra cualquier comando, si se encuentra en una carpeta diferente al directorio de inicio, la ruta de Guppy no será encontrada.

La gran mayoría de programas que corremos desde la línea de comando tienen la misma forma de estructurar el comando (sintaxis). Lo primero es indicar el nombre del programa a ejecutar. Después, escribir que tipo de opciones o variables queremos modificar. Observe en las primeras líneas, las opciones requeridas y el orden en que se piden, y explore todas las otras opciones que ofrece el programa.

Por ejemplo, si queremos ver la versión de Guppy que estamos usando utilizamos el comando

```
ont-guppy-cpu/bin/guppy_basecaller --version
```

El programa Guppy utiliza inteligencia artificial (modelos neuronales) para realizar el llamado de bases. El algoritmo de redes neuronales utiliza experimentos de anteriores de corridas, comparando el nivel de congruencia entre lo secuenciado y lo obtenido, para aprender y ser más precisos en esta tarea. El resultado de este proceso queda guardado en un archivo de configuración que es necesario indicarlo para correr el programa. Para cada molécula (ADN o ARN), tipo de celda (9.4.1 o 10.3), y algoritmo a utilizar (fast, hac o sup), existe un archivo de configuración. Para ver todas las configuraciones posibles podemos listarlas con el comando:

```
ont-guppy-cpu/bin/guppy_basecaller --print_workflows
```

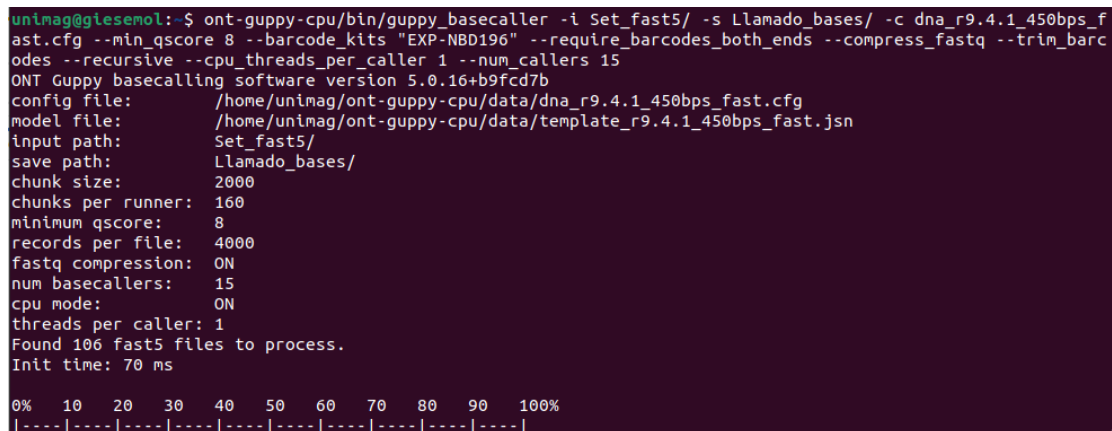
Guppy nos permite, mientras realiza el llamado de bases, separar por barcodes nuestras muestras. Esto es bastante útil para ganar tiempo y determinar si nuestras librerías quedaron bien preparadas. Con la opción `--barcode_kits` podemos indicarle al programa qué kit de librerías usamos y nos guardará los archivos en una carpeta para cada barcode detectado.

Otra opción que podemos usar es separar las lecturas de mala calidad, que tienen un alto porcentaje de error de secuenciación, de las buenas, que nos van a servir a los posteriores pasos. Con la opción `--min_qscore` podemos realizar un corte de calidad, y nuestras secuencias de nucleótidos van a ser clasificadas en una carpeta `fail`, que son las lecturas que no pasaron el filtro, y `pass`, que sí pasaron.

3. Llamado de bases

Una vez conocido las opciones que ofrece Guppy, podemos correr el programa, procurando usar la mejor configuración de argumentos para optimizar los tiempos de corrida. Cabe recordar que Guppy está pensado para correr usando tarjetas gráficas, que mejoran los tiempos de ejecución considerablemente, siendo 10 a 100 veces más rápido. Para nuestra práctica, vamos a correr el programa usando el procesador, con el algoritmo `fast` y calibrado para usar la mayor cantidad de recursos posibles. Siendo un set reducido de datos y con el llamado de bases configurado para ser lo más rápido posible, note cuánto se demora el programa en correr.

```
/ont-guppy/bin/guppy_basecaller -i Set_fast5 -s Llamado_bases/ -c dna_r9.4.1_450bps_sup.cfg --min_qscore 8 --barcode_kits "EXP-NBD196" --require_barcode_kits --compress_fastq --trim_barcode_kits --recursive --cpu_threads_per_caller 1 --num_callers 16
```



```
unimag@glesemol:~$ ont-guppy-cpu/bin/guppy_basecaller -i Set_fast5/ -s Llamado_bases/ -c dna_r9.4.1_450bps_sup.cfg --min_qscore 8 --barcode_kits "EXP-NBD196" --require_barcode_kits --compress_fastq --trim_barcode_kits --recursive --cpu_threads_per_caller 1 --num_callers 15
ONT Guppy basecalling software version 5.0.16+b9fcd7b
config file: /home/unimag/ont-guppy-cpu/data/dna_r9.4.1_450bps_fast.cfg
model file: /home/unimag/ont-guppy-cpu/data/template_r9.4.1_450bps_fast.jsn
input path: Set_fast5/
save path: Llamado_bases/
chunk size: 2000
chunks per runner: 160
minimum qscore: 8
records per file: 4000
fastq compression: ON
num basecallers: 15
cpu mode: ON
threads per caller: 1
Found 106 fast5 files to process.
Init time: 70 ms

0% 10 20 30 40 50 60 70 80 90 100%
|----|----|----|----|----|----|----|----|----|----|
```

Revise con cuidado el comando si sale un error. La mayoría de veces se debe a errores tipográficos. Si presenta alguna duda, siempre es recomendado consultar la documentación oficial del programa en la página de Oxford Nanopore para entender a profundidad

Podemos abrir otra pestaña en la terminal para observar los archivos generados por el programa. Si entramos a la carpeta `Llamado_bases`, vamos a encontrar las carpetas `pass` y `fail`. Dentro de estas

estarán por cada barcode los archivos FASTQ. Observe cuánto pesan en promedio los archivos comparado con FAST5.