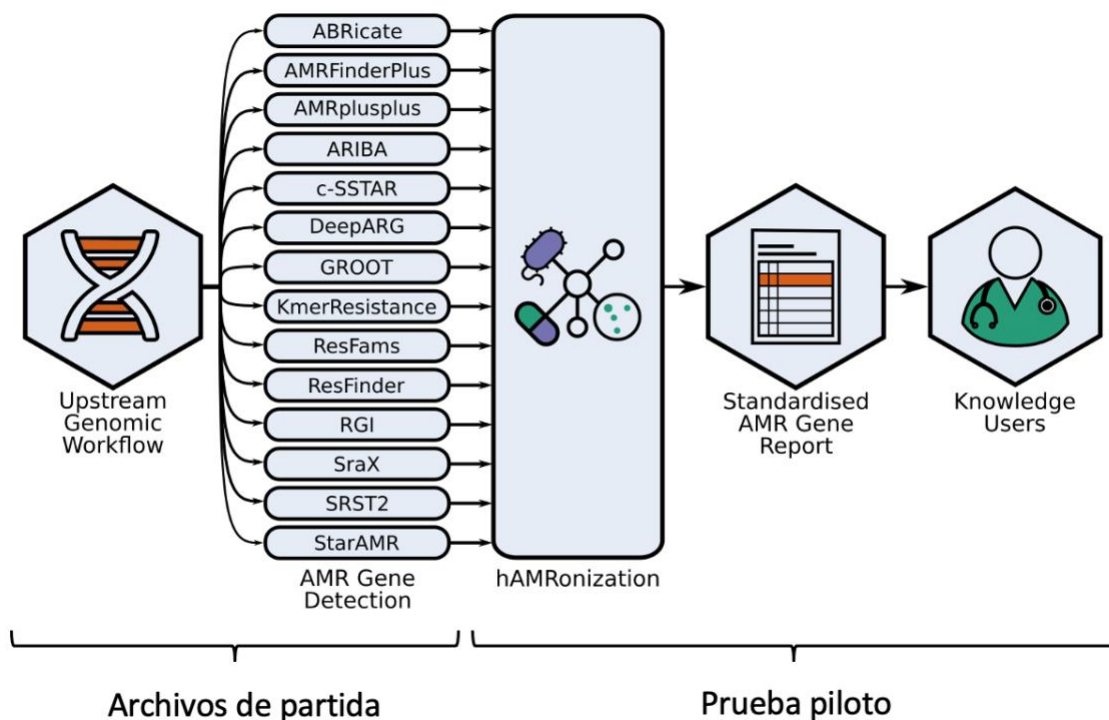


Prueba piloto: ReLAVRA y PulseNet América Latina y el Caribe

Instructivo para el uso de hAMRonization

hAMRonization es un proyecto que tiene como objetivo la estandarización de los archivos de salida de las herramientas de detección de resistencia a los antimicrobianos (RAM) a partir de datos genómicos.

Específicamente, en esta prueba piloto, los archivos resultantes del análisis de las secuencias con distintas herramientas de detección de genes de RAM les serán provistos y deberán utilizar *hAMRonization* para recopilar los resultados en un solo informe unificado.



En la carpeta de drive encontrarán los archivos de partida para esta prueba piloto, separados en distintas carpetas según la herramienta con la que fueron generados:

- 1) ABRicate
- 2) AMRFinderPlus
- 3) ARIBA
- 4) StarAMR
- 5) SRST2

https://drive.google.com/drive/folders/1kyK2cunFhSpuiRaXEsUmChIjsiNrc_HC?usp=sharing

Deberán descargar los archivos presentes en las carpetas mencionadas anteriormente y ordenarlos en su computadora/server de la misma manera. Es decir, generar carpetas para cada herramienta de detección de genes de RAM y guardar los archivos que se descarguen

del drive según corresponda. Luego, deberán correr *hAMRonization* para cada herramienta según se detalla a continuación.

De manera general, *hAMRonization* se corre utilizando el siguiente comando básico dentro de cada carpeta con los archivos de salida de los distintos programas de detección de RAM:

```
hamronize <nombre_herramienta> <archivos_output_de_la_herramienta>  
<metadata_solicitada>
```

Para conocer qué metadata se necesita para armonizar los archivos de salida de cada herramienta, correr el siguiente comando de acuerdo a la herramienta requerida:

```
hamronize <nombre_herramienta> -h
```

ABRicate

Versión utilizada= 1.0.1

Base de datos utilizada= RESFINDER

- *Ejemplo comando para una muestra*

En la misma carpeta donde se encuentran los archivos de salida de ABRicate que bajaron del drive, correr el siguiente comando:

```
hamronize abricate ERR1009154.abricate.tsv --output  
ERR1009154.abricate.hamronized.tsv --analysis_software_version 1.0.1  
--reference_database resfinder_septiembre
```

- *Ejemplo comando para varias muestras*

En la misma carpeta donde se encuentran los archivos de salida de ABRicate que bajaron del drive, correr el siguiente comando:

```
hamronize abricate *.abricate.tsv --output abricate.hamronized.tsv --  
analysis_software_version 1.0.1 --reference_database  
resfinder_septiembre
```

AMRFinderPlus

Versión utilizada= 3.8.4

Base de datos utilizada= ncbi_Sep_2020

- *Ejemplo comando para una muestra*

En la misma carpeta donde se encuentran los archivos de salida de AMRFinderPlus que bajaron del drive, correr el siguiente comando:

```
hamronize amrfinderplus ERR1009154.amrfinder.tsv --output  
ERR1009154.amrfinder.hamronized.tsv --analysis_software_version 3.8.4  
--reference_database_version ncbi_Sep_2020 --input_file_name  
ERR1009154
```

- *Ejemplo comando para varias muestras*

En la misma carpeta donde se encuentran los archivos de salida de AMRFinderPlus que bajaron del drive, correr el siguiente comando:

```
for f in *.amrfinder.tsv; do hamronize amrfinderplus $f --output
${f%.tsv}.amrfinder.hamronized.tsv --analysis_software_version 3.8.4
--reference_database_version ncbi_Sep_2020 --input_file_name
${f%.amrfinder.tsv}; done
```

ARIBA

Versión utilizada= 2.14.6

Base de datos utilizada= RESFINDER

- *Ejemplo comando para una muestra*

En la misma carpeta donde se encuentran los archivos de salida de ARIBA que bajaron del drive, correr el siguiente comando:

```
hamronize ariba ERR1009154.ariba.tsv --output
ERR1009154.ariba.hamronized.tsv --analysis_software_version 2.14.6 --
reference_database_version resfinder_septiembre --
reference_database_id resfinder --input_file_name ERR1009154
```

- *Ejemplo comando para varias muestras*

En la misma carpeta donde se encuentran los archivos de salida de ARIBA que bajaron del drive, correr el siguiente comando:

```
for f in *.ariba.tsv; do hamronize ariba $f --output
${f%.tsv}.hamronized.tsv --analysis_software_version 2.14.6 --
reference_database_version resfinder_septiembre --
reference_database_id resfinder --input_file_name ${f%.ariba.tsv};
done
```

StarAMR

Versión utilizada= 0.5.1

Base de datos utilizada= RESFINDER

StarAMR genera un único archivo para todas las muestras analizadas y sobre este archivo generado (en el drive “staramr.tsv”) se puede correr hamronize:

```
hamronize staramr staramr.tsv --analysis_software_version 0.5.1 --
reference_database_version resfinder_octubre --output
staramr.hamronized.tsv
```

SRST2

srst2 version = unknown

DB=RESFINDER

- *Ejemplo comando para una muestra*

En la misma carpeta donde se encuentran los archivos de salida de SRST2 que bajaron del drive, correr el siguiente comando:

```
hamronize srst2 ERR1009154.srst2.tsv --output
ERR1009154.srst2.hamronized.tsv --analysis_software_version unknown -
```

```
-reference_database_version resfinder_septiembre --input_file_name  
ERR1009154
```

- *Ejemplo comando para varias muestras*

En la misma carpeta donde se encuentran los archivos de salida de SRST2 que bajaron del drive, correr el siguiente comando:

```
for f in *.srst2.tsv; do hamronize srst2 $f --  
analysis_software_version unknown --reference_database_version  
resfinder_septiembre --input_file_name ${f%.srst2.tsv} --output  
${f%.srst2.tsv}.srst2.hamronized.tsv; done
```

Summarize

Finalmente, se puede generar un resumen de todos los informes hAMRonizados obtenidos (independientemente del formato) utilizando la función '*summarize*'. Para conocer más acerca de esta opción utilizar el comando:

```
hamronize summarize -h
```

El comando básico es: `hamronize summarize <opciones> <lista de reportes>`

1) Les sugerimos que, en primera instancia, generen un único archivo por herramienta (a excepción de StarAMR y ABRicate) y luego creen el resumen de todas las herramientas juntas.

Para ello, también utilizaremos '*summarize*'. Dentro de las carpetas con los respectivos resultados armonizados de cada herramienta, correr el siguiente comando en cada carpeta agregando el nombre de la herramienta utilizada para la detección de genes de RAM donde dice <herramienta>:

```
hamronize summarize -o <herramienta>.summarize.tsv -t tsv *.tsv
```

Por ejemplo: `hamronize summarize -o ariba.summarize.tsv -t tsv
*.ariba.hamronized.tsv`

2) Para generar un único archivo que contenga los resultados armonizados de múltiples herramientas de detección de RAM utilizaremos nuevamente '*summarize*' pero esta vez con los archivos generados para cada herramienta en el punto 1) e incluir los archivos armonizados de ABRicate y StarAMR.

Dentro de las opciones para `hamronize summarize` podremos definir el formato del archivo de salida. Para comparar los formatos .tsv y .html (interactivo), tendremos que correr los siguientes comandos en la misma carpeta donde se encuentran los archivos armonizados o especificando la ruta hacia ellos:

- *Ejemplo para obtener el resumen en formato .tsv*

```
hamronize summarize -o combined_report.tsv -t tsv ariba.summarize.tsv  
srst2.summarize.tsv amrfinderplus.summarize.tsv  
abricate.hamronized.tsv staramr.hamronized.tsv
```

- *Ejemplo para obtener el resumen en formato .html*

```
hamronize summarize -o combined_report.html -t interactive  
ariba.summarize.tsv srst2.summarize.tsv amrfinderplus.summarize.tsv  
abricate.hamronized.tsv staramr.hamronized.tsv
```

→ Por último, una vez generados todos los archivos, analizarlos e interpretarlos teniendo en cuenta las preguntas en el archivo “PHA4GE_hAMRonization_español” y enviarnos las respuestas a María Sol Haim (solhaim@gmail.com) con copia a Josefina Campos (jocampos05@gmail.com) y Marcelo Galas (galasmar@paho.org).