



Matthieu J. Miossec

twitter: @RealMattJM

Unidad 9: Análisis genómicos reproducibles en la nube



Programa Unidad 9

- 11 de mayo (lunes)– Introducción a la genómica en la nube con Terra.
- 13 de mayo (miércoles) – GATK ‘Best Practices’ GVCF Workflow [en Terra]
- 18 de mayo (hoy!) – WDL y otras herramientas [en Terra]:
 - Mutect2 (variantes somáticas, cáncer)
 - GermlineCNVCaller (CNVs de la línea germinal)

Agradecimientos

ISCB-LA
SoIBio EMBnet
— 2018 —

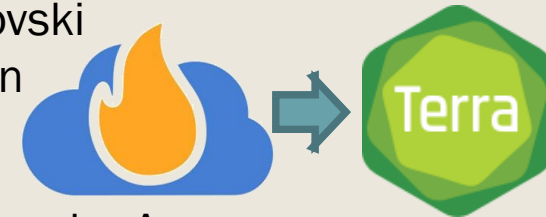
Viña del Mar, Chile
November 5 - 9, 2018



Agradecimiento especiales,
al equipo de Viña del Mar (Nov 2018)




-Tiffany Miller
-Robert Majovski
-Yossi Farjoun



También...

-Geraldine Van der Auwera
-Beri Shifaw
-Allie Hajian
-Kate Noblett

- Data Sciences Platform  BROAD INSTITUTE
Broad Institute of Harvard and MIT
<https://gatk.broadinstitute.org/>

Por los materiales (Terra, workflows,
docs...), gráficos y el apoyo otorgado
durante la preparación de la clase de
2019.

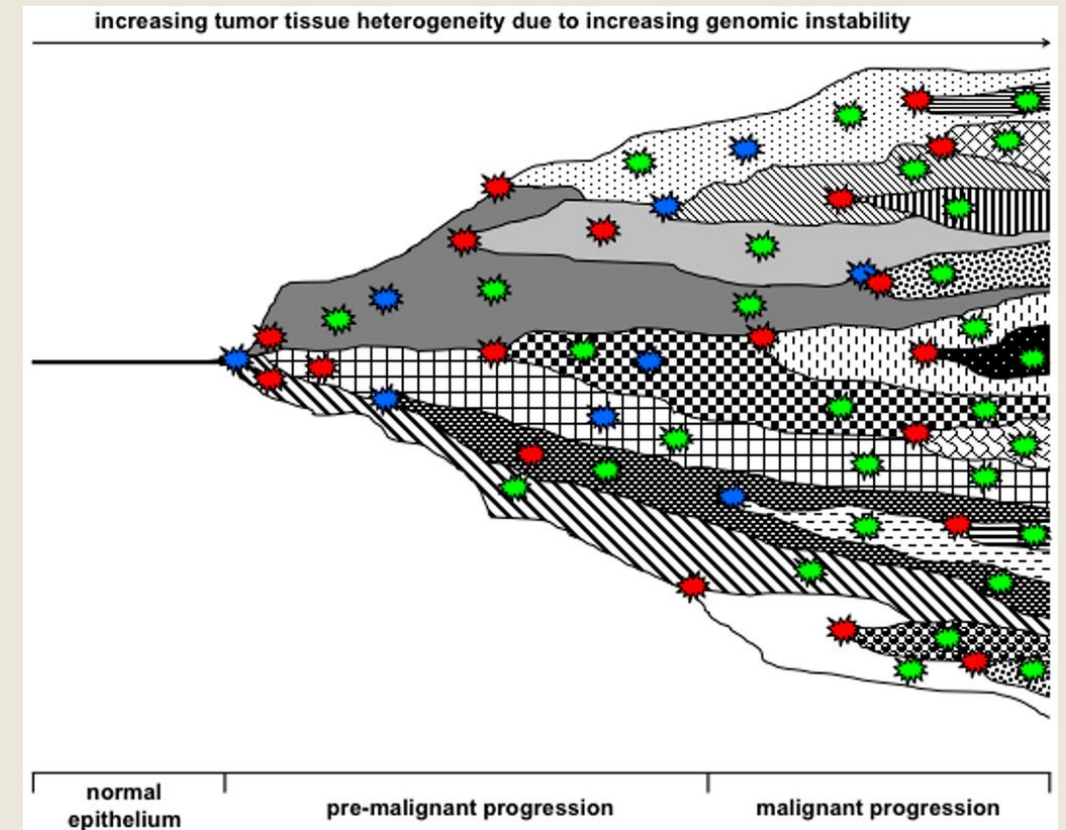
GATK 4 en Terra

- GATK 4 tiene varias herramientas que van más allá del llamado de variantes cortitas en la línea germinal.
 - Llamar variaciones estructurales es el próximo gran desafío, tanto para variaciones en la línea germinal que somática (cáncer).

	GERMLINE	SOMATIC
SNPs & INDELS	HaplotypeCaller GVCF	MuTect2
Copy Number	GATK gCNV	GATK CNV + aCNV
Structural Variation	GATK SVDISCOVERY (beta)	(planned)

Variación en Células Somáticas

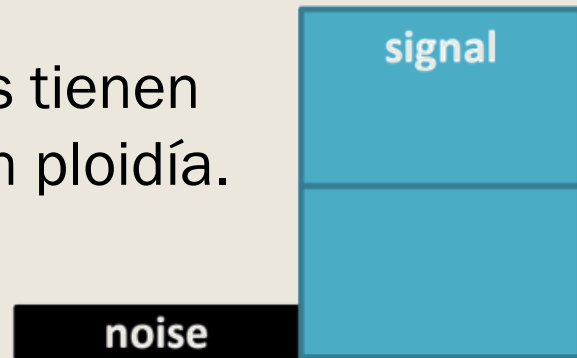
- Células normales: variantes en un locus se limitan por mayor parte a una o dos...
 - si observamos más → ruido
- Células tumorales: Es un otro desafío...
 - En una muestra, ¿Qué corresponde a variación germinal?
¿somática?
(¿'passenger' o 'driver'?)



Tasa de señal vs. ruido

Lo que podemos esperar con variantes en la línea germinal

Frecuencias tienen relación con ploidía.



Los datos tienen artefactos, pero existen diversas maneras de detectarlos una grande proporción de ellos.

... pero con variantes en líneas somáticas



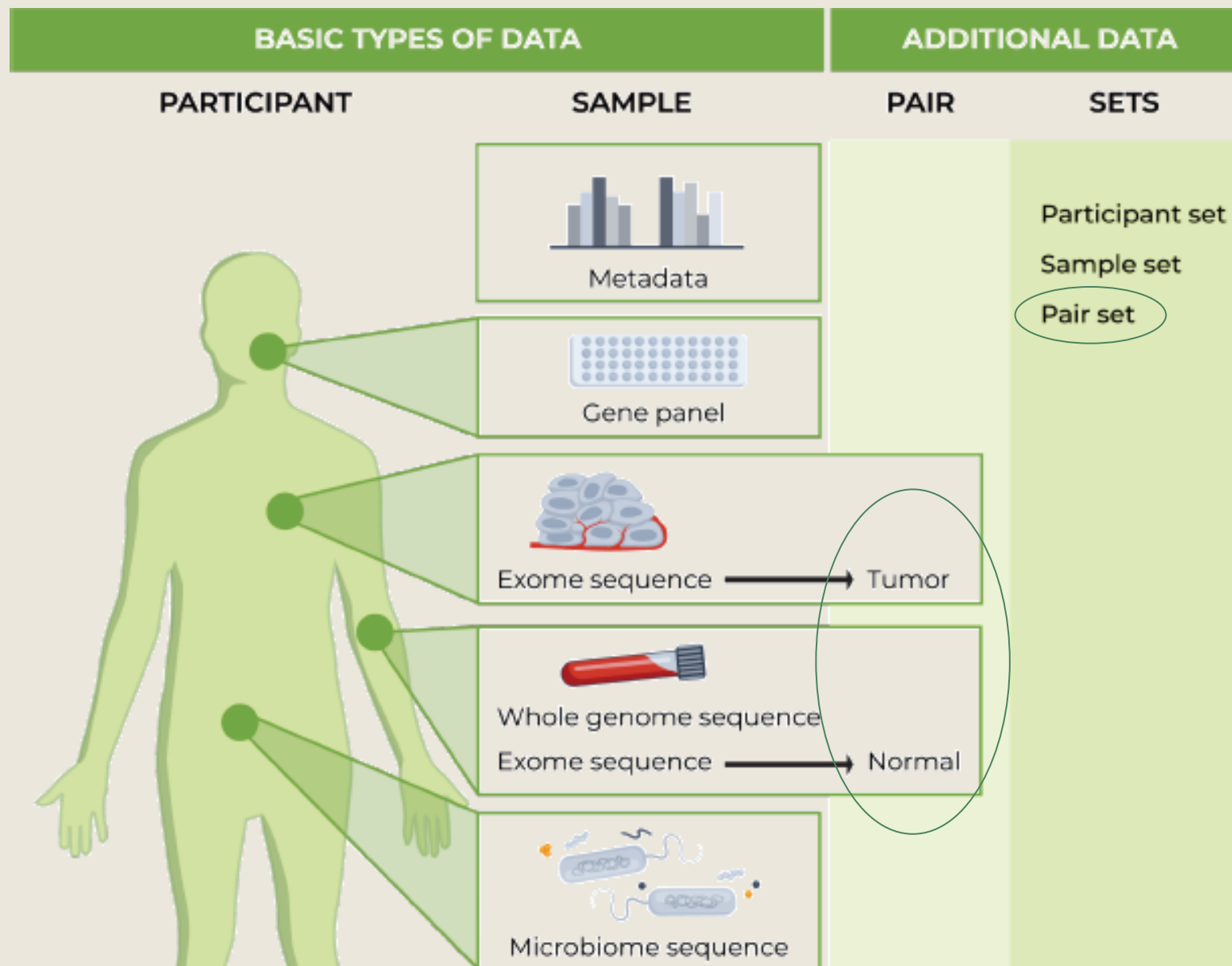
No hay una sola ploidía (mezcla de subclones)

- ✗ Variaciones presentes en la línea germinal
- ✗ Poblaciones subclonales
 - ✗ Células normales en muestra de tumor*
...además de los artefactos presente en toda secuenciación.

* Muestras de células normales también pueden tener contaminación.

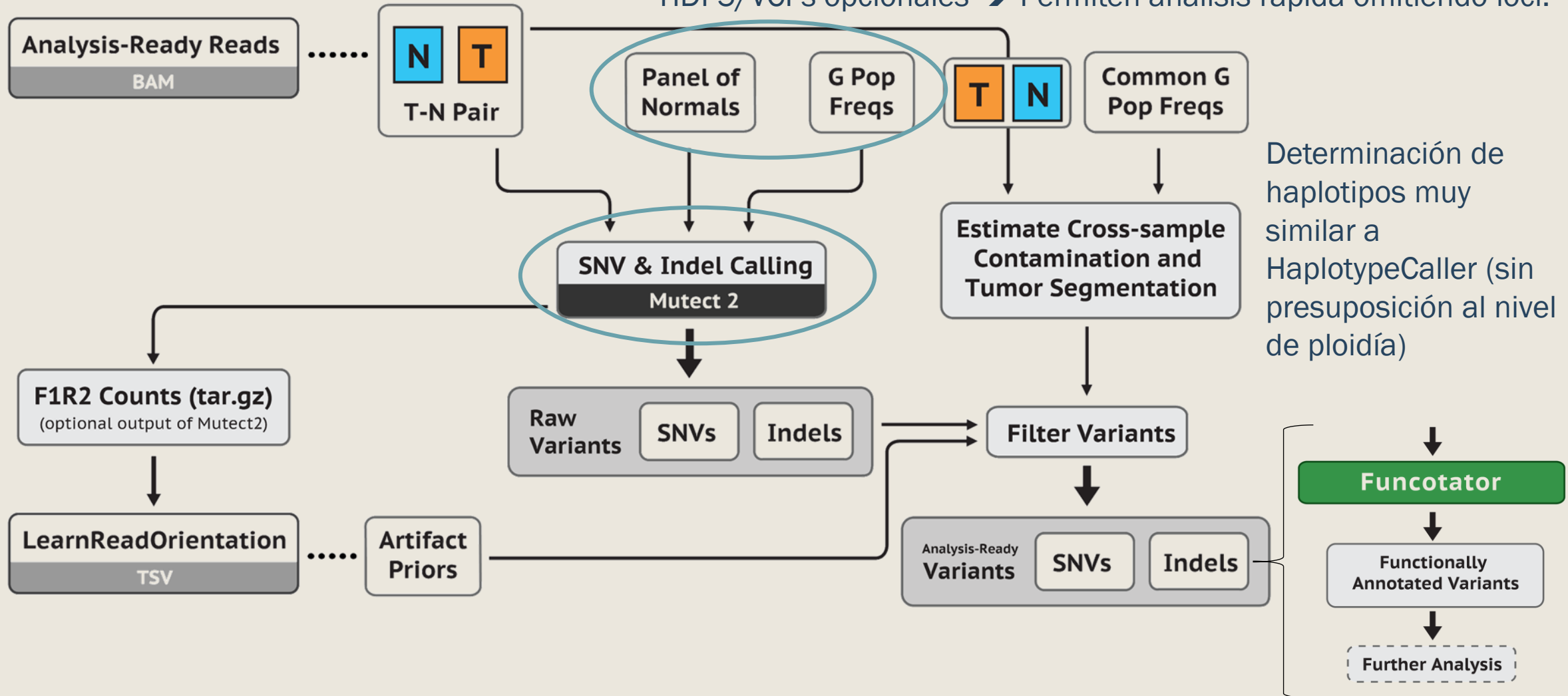
Pare Tumor-Normal

- Pares de muestras **tumor-normal** (del mismo paciente) permite eliminar variación presente en todas las células, es decir, variantes de origen germinal (así que artefactos sistémicos).
 - Si una variante aparece en ambos tipos de secuencia, mucho más probable que sea variante germinal que variante somática en uno y artefacto en otro.
 - Para eliminar dudo sirve también filtrar datos contra bases de datos como gnomAD.



Mutect 2

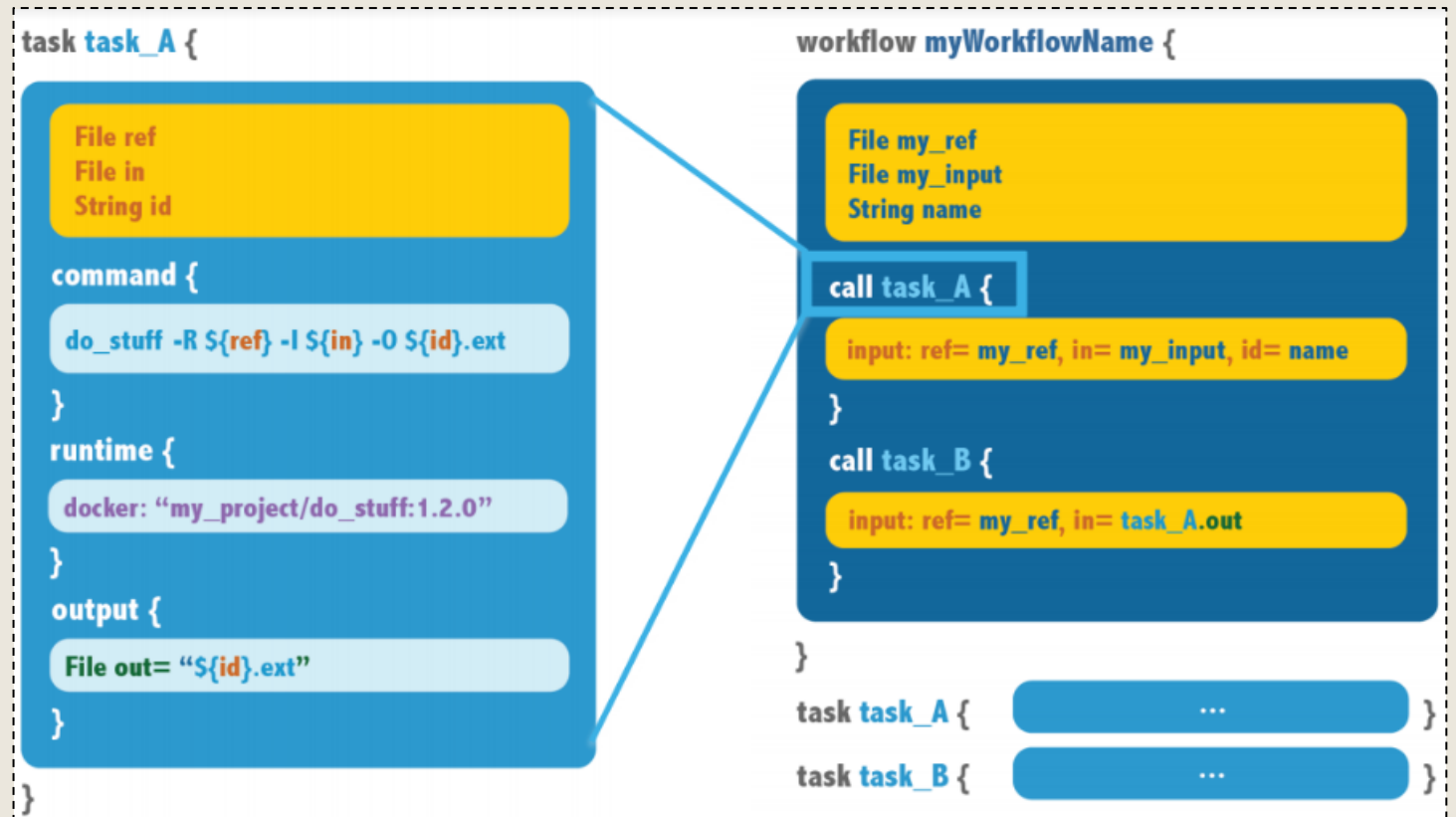
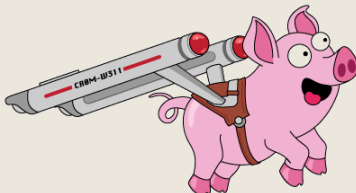
HDF5/VCFs opcionales → Permiten análisis rápida omitiendo loci.



Workflow Description Language



- Un lenguaje simple para describir ‘workflows’.
- Reúne datos de entrada/salida, herramientas y comandos
- Interpretado y ejecutado por **Cromwell**.

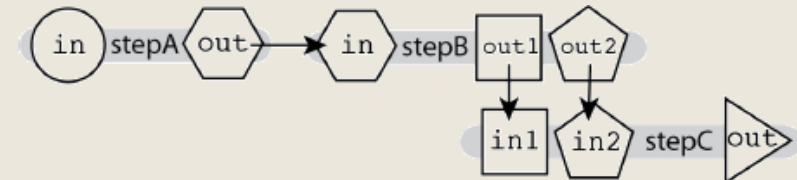


Organización de Tareas en WDL

- Existen tres maneras de organizar nuestras tareas.
 - Cadena lineal o con input/output múltiples.

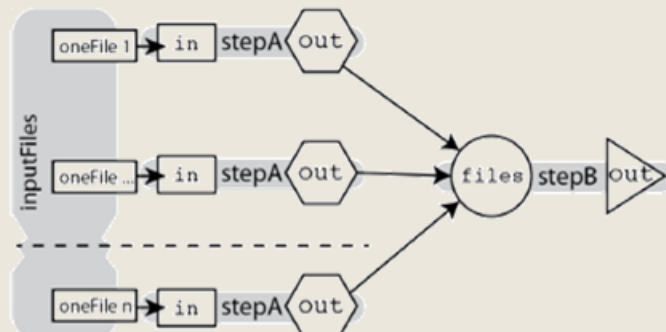


```
call stepA
call stepB { input: in=stepA.out }
call stepC { input: in=stepB.out }
```



```
call stepC { input :
    in1=stepB.out1,
    in2=stepB.out2 }
```

- Scatter-gather (Dispersar-Reunir)



```
Array[File] inputFiles

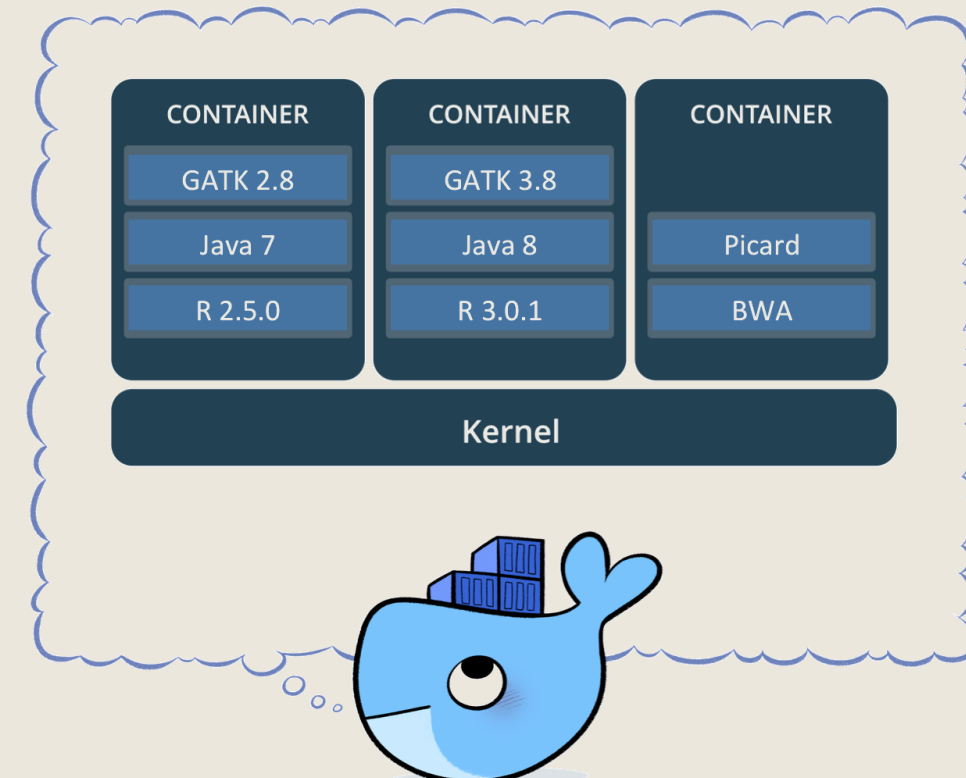
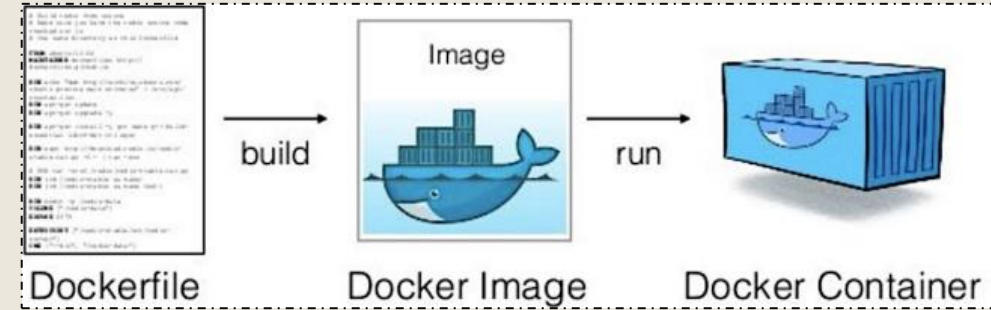
scatter(oneFile in inputFiles) {
    call stepA { input: in=oneFile }
}

call stepB { input: files=stepA.out }
```

No funciona con todo los sistemas operativos de Windows :’(
Con Linux y Mac debería funcionar.

Docker

- Container: Similar a una maquina virtual.
 - Crea una ‘imagen’ de un sistema operativo (versión de Linux) como base.
+ los programas necesarios para ejecutar un conjunto de tareas predefinidas.
- Posible ejecutar varios ‘containers’ en la misma maquina si existen incompatibilidades entre sistemas.



Referirse a un 'container' en WDL

- Simple! Una vez el 'container' Docker listo, lo ponemos en línea a través del **Google Container Repository** (acceso privado/público) o **Docker Hub** (público).
- En el WDL, nos referimos al 'container' Docker en el cuerpo de **runtime** con una línea.
 - docker:"broadinstitute/gatk:4.1.2.0" ➔

```
runtime {  
  docker: "my_project/do_stuff:1.2.0"  
}
```

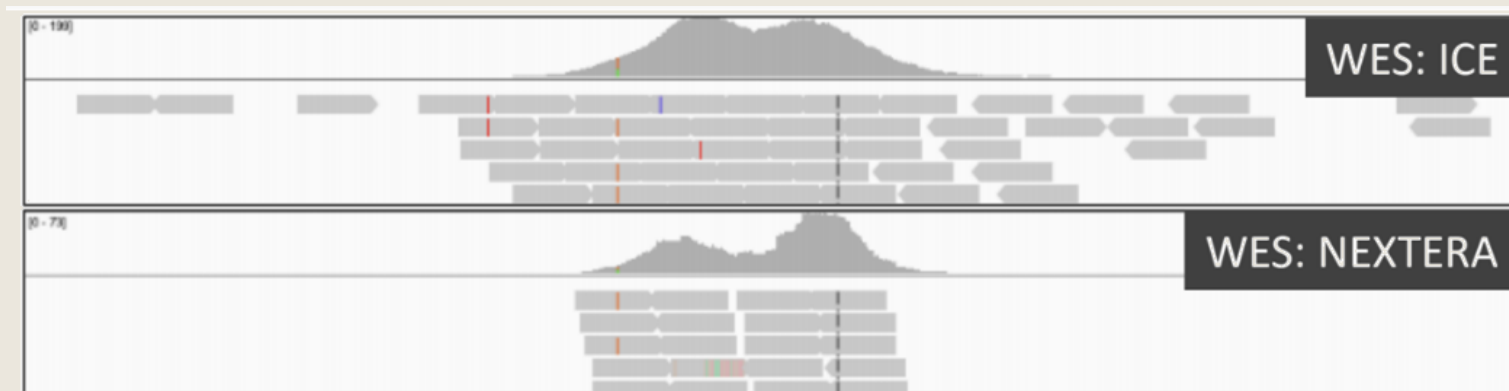
GATK gCNV

- Una nueva herramienta que permite detectar **CNVs, comunes y poco frecuentes**, en la línea germinal de una cohorte de pacientes.
 - No es necesario tener casos/controles, todas las secuencias participan en la creación de un **modelo de cobertura**.
 - Para tener un modelo de cobertura adecuado, necesitamos por lo menos 30 secuencias.
 - Depende de calidad de las muestras (profundidad, grado de similitud entre protocolos de preparación de librería y secuenciación entre secuencias...)

La cobertura

Gran Obstáculo para detección de CNVs

- La cobertura en exomas varia no solo **entre regiones** sino también **entre tecnologías de captura!**



Nada que ver con la cobertura bastante uniforme de genomas enteros!

- Con muchas secuencias se puede separar los dos tipos de variaciones en cobertura: dado captura vs. dado los CNVs

Reusar Modelo de Cobertura

- El **modelo de cobertura** está creado al mismo tiempo que los CNVs están detectados cuando usamos gCNV en **COHORT MODE**.
- Si ya tenemos un **modelo de cobertura** que corresponde bien a nuevos datos de secuenciación (i.e. mismos protocolos) podemos reusarlo para detectar CNVs en los nuevos datos usando GermlineCNVCaller en **CASE MODE**.
 - Más económico en termino de computación.

gCNV 'Workflow'...(uups!)

- Recién salió de estado BETA.

- Al momento 'Best Practices' no existe y 'Featured Workspace' tampoco.
- Sin embargo, existen 'workflows'.
 - Hay simplemente que buscarlo en **Dockstore!**
 - Uno para COHORT y uno para CASE MODE.

✓ Germline copy number variant discovery (CNVs)

Best Practices Workflows | Created 2018-01-07 | Last updated 2018-01-09

[Comm](#)

Purpose

Identify germline copy number variants.

Diagram is not available

Reference implementation is not available

This workflow is in development; detailed documentation will be made available when the workflow is considered fully released.

Extra: Ejecutar WDL en maquina Local

- Podemos ejecutar un script WDL a fuera de la nube usando Cromwell localmente.

```
java -jar cromwell.jar \  
  run hello.wdl \  
  --inputs hello_inputs.json
```

- Con **WOMtool (Workflow Object Model)** verificamos la sintaxis de nuestro scripts (y también creamos input automáticamente)

```
java -jar womtool.jar validate hello.wdl
```

Extra: Instalar Cromwell

- Con Java a la fecha (v1.8+), se necesita poco tiempo para instalar Cromwell localmente:

<https://cromwell.readthedocs.io/en/stable/tutorials/FiveMinuteIntro/>

- Las instrucciones están acompañadas de un primer ejemplo de un WDL, un “Hello World!”.
- Su primer **WDL!**

Extra: "Hello World"

```
workflow myWorkflow {
  call mensajeBienvenida
}

task mensajeBienvenida {
  command {
    echo "hola mundo!"
  }
  output {
    String out = read_string(stdout())
  }
}
```

```
imac-math:WDL_tutorial cbib$ java -jar cromwell-42.jar run miWorkflow.wdl
[2019-06-04 10:04:53,70] [info] Running with database db.url = jdbc:hsqldb:mem:4ec5c15f-619b-4092-85d9-1088a81221fa;shutdown=false;hsqldb.tx=mvcc
```

```
interval = None
[2019-06-04 10:05:09,38] [info] WorkflowExecutionActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f [c2487e5d]: Workflow myWorkflow complete. Final Outputs:
{
  "myWorkflow.mensajeBienvenida.out": "hola mundo!"
}
[2019-06-04 10:05:09,40] [info] WorkflowManagerActor WorkflowActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f is in a terminal state: WorkflowSucceededState
```

```
WDL_tutorial — -bash — 87x93
imac-math:WDL_tutorial cbib$ java -jar cromwell-42.jar run miWorkflow.wdl
[2019-06-04 10:04:53,70] [info] Running with database db.url = jdbc:hsqldb:mem:4ec5c15f-619b-4092-85d9-1088a81221fa;shutdown=false;hsqldb.tx=mvcc
[2019-06-04 10:05:01,44] [info] Running migration RenameWorkflowOptionsInMetadata with a read batch size of 100000 and a write batch size of 100000
[2019-06-04 10:05:01,46] [info] [RenameWorkflowOptionsInMetadata] 100%
[2019-06-04 10:05:01,57] [info] Running with database db.url = jdbc:hsqldb:mem:dbecc2df-d809-4a97-84c0-2a13617db0c2;shutdown=false;hsqldb.tx=mvcc
[2019-06-04 10:05:02,02] [info] SLF4JLogger started
[2019-06-04 10:05:02,39] [info] Workflow heartbeat configuration:
{
  "cromwellId" : "cromid-24440d9",
  "heartbeatInterval" : "2 minutes",
  "ttl" : "10 minutes",
  "failureShutdownDuration" : "5 minutes",
  "writeBatchSize" : 10000,
  "writeThreshold" : 10000
}
[2019-06-04 10:05:02,44] [info] Metadata summary refreshing every 1 second.
[2019-06-04 10:05:02,46] [info] KvWriteActor configured to flush with batch size 200 and process rate 5 seconds.
[2019-06-04 10:05:02,46] [info] WriteMetadataActor configured to flush with batch size 200 and process rate 5 seconds.
[2019-06-04 10:05:02,55] [info] CallCacheWriteActor configured to flush with batch size 100 and process rate 3 seconds.
[2019-06-04 10:05:02,55] [warn] 'docker.hash-lookup.gcr-api-queries-per-100-seconds' is being deprecated, use 'docker.hash-lookup.gcr.throttle' instead (see reference.conf)
[2019-06-04 10:05:02,90] [info] JobExecutionTokenDispenser - Distribution rate: 50 per 1 seconds.
[2019-06-04 10:05:03,00] [info] SingleWorkflowRunnerActor: Version 42
[2019-06-04 10:05:03,01] [info] SingleWorkflowRunnerActor: Submitting workflow
[2019-06-04 10:05:03,08] [info] Unspecified type (Unspecified version) workflow c2487e5d-6a28-4c2a-a4c2-023d12c2373f submitted
[2019-06-04 10:05:03,11] [info] SingleWorkflowRunnerActor: Workflow submitted c2487e5d-6a28-4c2a-a4c2-023d12c2373f
[2019-06-04 10:05:03,11] [info] 1 new workflows fetched by cromid-24440d9: c2487e5d-6a28-4c2a-a4c2-023d12c2373f
[2019-06-04 10:05:03,13] [info] WorkflowManagerActor Starting workflow c2487e5d-6a28-4c2a-a4c2-023d12c2373f
[2019-06-04 10:05:03,14] [info] WorkflowManagerActor Successfully started WorkflowActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f
[2019-06-04 10:05:03,14] [info] Retrieved 1 workflows from the WorkflowStoreActor
[2019-06-04 10:05:03,15] [info] WorkflowStoreHeartbeatWriteActor configured to flush with batch size 10000 and process rate 2 minutes.
[2019-06-04 10:05:03,26] [info] MaterializeWorkflowDescriptorActor [c2487e5d]: Parsing workflow as WDL draft-2
[2019-06-04 10:05:03,92] [info] MaterializeWorkflowDescriptorActor [c2487e5d]: Call-to-Backend assignments: myWorkflow.mensajeBienvenida -> Local
[2019-06-04 10:05:05,21] [info] WorkflowExecutionActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f [c2487e5d]: Starting myWorkflow.mensajeBienvenida
[2019-06-04 10:05:05,93] [info] Assigned new job execution tokens to the following groups: c2487e5d: 1
[2019-06-04 10:05:06,09] [info] BackgroundConfigAsyncJobExecutionActor [c2487e5dmyWorkflow.mensajeBienvenida:NA:1]: echo "hola mundo!"
[2019-06-04 10:05:06,34] [info] BackgroundConfigAsyncJobExecutionActor [c2487e5dmyWorkflow.mensajeBienvenida:NA:1]: executing: /bin/bash /Users/cbib/Desktop/WDL_tutorial/cromwell-executions/myWorkflow/c2487e5d-6a28-4c2a-a4c2-023d12c2373f/call-mensajeBienvenida/execution/script
[2019-06-04 10:05:07,51] [info] BackgroundConfigAsyncJobExecutionActor [c2487e5dmyWorkflow.mensajeBienvenida:NA:1]: job id: 3826
[2019-06-04 10:05:07,52] [info] BackgroundConfigAsyncJobExecutionActor [c2487e5dmyWorkflow.mensajeBienvenida:NA:1]: Status change from - to Done
[2019-06-04 10:05:07,91] [info] Not triggering log of token queue status. Effective log interval = None
[2019-06-04 10:05:09,38] [info] WorkflowExecutionActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f [c2487e5d]: Workflow myWorkflow complete. Final Outputs:
{
  "myWorkflow.mensajeBienvenida.out": "hola mundo!"
}
[2019-06-04 10:05:09,40] [info] WorkflowManagerActor WorkflowActor-c2487e5d-6a28-4c2a-a4c2-023d12c2373f is in a terminal state: WorkflowSucceededState
[2019-06-04 10:05:14,36] [info] SingleWorkflowRunnerActor workflow finished with status 'Succeeded'.
{
  "outputs": {
    "myWorkflow.mensajeBienvenida.out": "hola mundo!"
  },
  "id": "c2487e5d-6a28-4c2a-a4c2-023d12c2373f"
}
[2019-06-04 10:05:17,53] [info] Workflow polling stopped
[2019-06-04 10:05:17,54] [info] 0 workflows released by cromid-24440d9
[2019-06-04 10:05:17,54] [info] Shutting down WorkflowStoreActor - Timeout = 5 seconds
[2019-06-04 10:05:17,55] [info] Shutting down WorkflowLogCopyRouter - Timeout = 5 seconds
[2019-06-04 10:05:17,55] [info] Shutting down JobExecutionTokenDispenser - Timeout = 5 seconds
[2019-06-04 10:05:17,55] [info] Aborting all running workflows.
[2019-06-04 10:05:17,56] [info] JobExecutionTokenDispenser stopped
[2019-06-04 10:05:17,56] [info] WorkflowStoreActor stopped
[2019-06-04 10:05:17,57] [info] Shutting down WorkflowManagerActor - Timeout = 3600 seconds
[2019-06-04 10:05:17,57] [info] WorkflowLogCopyRouter stopped
[2019-06-04 10:05:17,57] [info] WorkflowManagerActor All workflows finished
```

Extra: Variable Configurable con .JSON

```
WDL_tutorial — vi miWorkflowConfigurable.wdl — 123x93

workflow myWorkflow {
  call mensajeBienvenida
}

task mensajeBienvenida {
  String bienvenida
  command {
    echo "${bienvenida}"
  }
  output {
    String out = read_string(stdout())
  }
}
```

```
WDL_tutorial — vi misVariables.json — 123x

{
  "myWorkflow.mensajeBienvenida.bienvenida": "Hola Chile, Hola Mexico."
}
```

```
[2019-06-04 10:49:43,77] [INFO] WorkflowExecutionActor-2ad63220-29b9-41d9-8000-d1
. Final Outputs:
{
  "myWorkflow.mensajeBienvenida.out": "Hola Chile, Hola Mexico."
}
[2019-06-04 10:49:43.80] [info] WorkflowManagerActor WorkflowActor-2ad63220-29b9-
```

Siguiendo el formato llave:valor siguiente:

```
{
  "<nombre_workflow>.<tarea_llamada>.<variable>": "<valor>"
}
```

Extra: WOMtools

- Con WOMtools se puede verificar si la sintaxis del WDL esta correcta.

```
java -jar /gatk/my_data/jars/womtool-38.jar validate  
/gatk/my_data/hello_world/hello_world_2.wdl -i hello_world.inputs.json
```

- ...y también generar un .JSON con input basado en el WDL.

```
java -jar /gatk/my_data/jars/womtool-38.jar inputs  
/gatk/my_data/hello_world/hello_world_2.wdl > hello_world_2.inputs.json
```

Extra: Cadena Simple

```
workflow myWorkflow {  
  call mensajeBienvenida  
  
  call responder{  
    input:  
      mensaje_original = mensajeBienvenida.out  
  }  
}  
  
task mensajeBienvenida {  
  String bienvenida  
  
  command {  
    echo "${bienvenida}"  
  }  
  output {  
    String out = read_string(stdout())  
  }  
}  
  
task responder{  
  String mensaje_original  
  
  command{  
    echo "${mensaje_original} Hola profesor!"  
  }  
  output{  
    File archivo_out = stdout()  
  }  
}
```

```
rom - to Done  
[2019-06-04 10:30:15,69] [info] WorkflowExecutionActor-dcfc61ae-4760-4586-b8d0-bba7  
complete. Final Outputs:  
{  
  "myWorkflow.mensajeBienvenida.out": "Hola Chile, Hola Mexico.",  
  "myWorkflow.responder.archivo_out": "/Users/cbib/Desktop/WDL_tutorial/cromwell-ex  
8d0-bba7eae9e7c1/call-responder/execution/stdout"  
}
```

```
imac-math:execution cbib$ echo $PWD  
/Users/cbib/Desktop/WDL_tutorial/cromwell-executions/myWorkflow/dcfc6  
[imac-math:execution cbib$ more stdout  
Hola Chile, Hola Mexico. Hola profesor!  
imac-math:execution cbib$
```