



Genomics in the Cloud


Book Club - Week 10

February 1, 2021

Agenda

- Chapter 9: Deciphering Real Genomics Workflows
- Additional resources
- Open discussion





Our guest
speaker

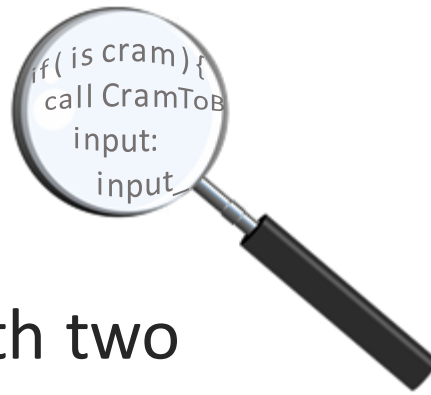
Dr. Matthieu
Miossec



Chapter 9: Deciphering Real Genomics Workflows

Genomics in the Cloud by Geraldine A. Van der Auwera and Brian D. O'Connor (O'Reilly). Copyright 2020 The Broad Institute, Inc. and Brian O'Connor, 978-1-491-97519-0.

Mystery workflows



- Delving further into WDL with two previously unseen workflows:
 - Mystery Workflow #1: Flexibility Through Conditionals

```
$ export CASE1=~ /book/code/workflows/mystery-1
```

- Mystery Workflow #2: Modularity and Code Reuse

```
$ export CASE2=~ /book/code/workflows/mystery-2
```

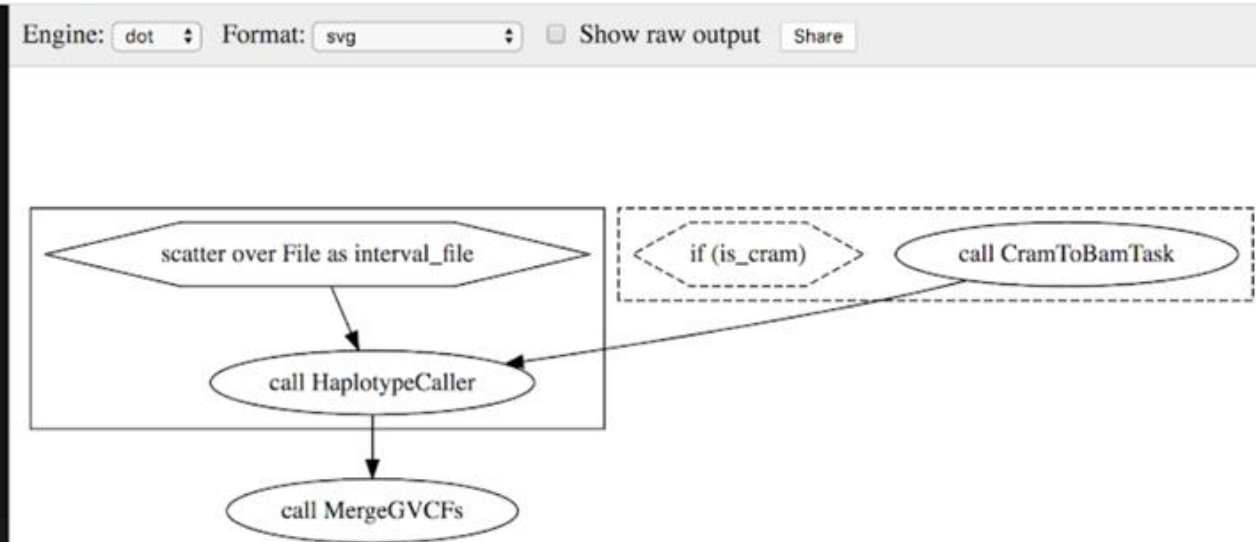
- Sandbox directory for storing output:

```
$ mkdir ~/sandbox-9
```



Mystery Workflow 1 in Graphviz Online

```
1 digraph HaplotypeCallerGvcf_GATK4 {
2   #rankdir=LR;
3   compound=true;
4   # Links
5   CALL_HaplotypeCaller -> CALL_MergeGVCfs
6   SCATTER_1_VARIABLE_interval_file -> CALL_HaplotypeCaller
7   CALL_CramToBamTask -> CALL_HaplotypeCaller
8   # Nodes
9   subgraph cluster_0 {
10    style="filled,dashed";
11    fillcolor=white;
12    CALL_CramToBamTask [label="call CramToBamTask"]
13    CONDITIONAL_0_EXPRESSION [shape="hexagon" label="if (is_cram)" style="dashed" ]
14  }
15  CALL_MergeGVCfs [label="call MergeGVCfs"]
16  subgraph cluster_1 {
17    style="filled,solid";
18    fillcolor=white;
19    CALL_HaplotypeCaller [label="call HaplotypeCaller"]
20    SCATTER_1_VARIABLE_interval_file [shape="hexagon" label="scatter over File as interval_file"]
21  }
22 }
```

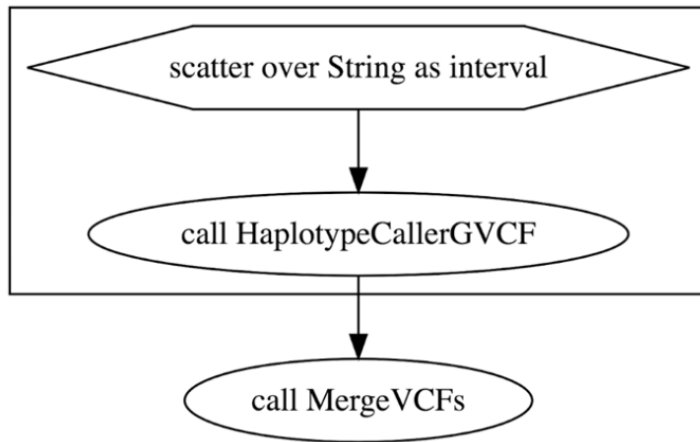


Certainly looks familiar...

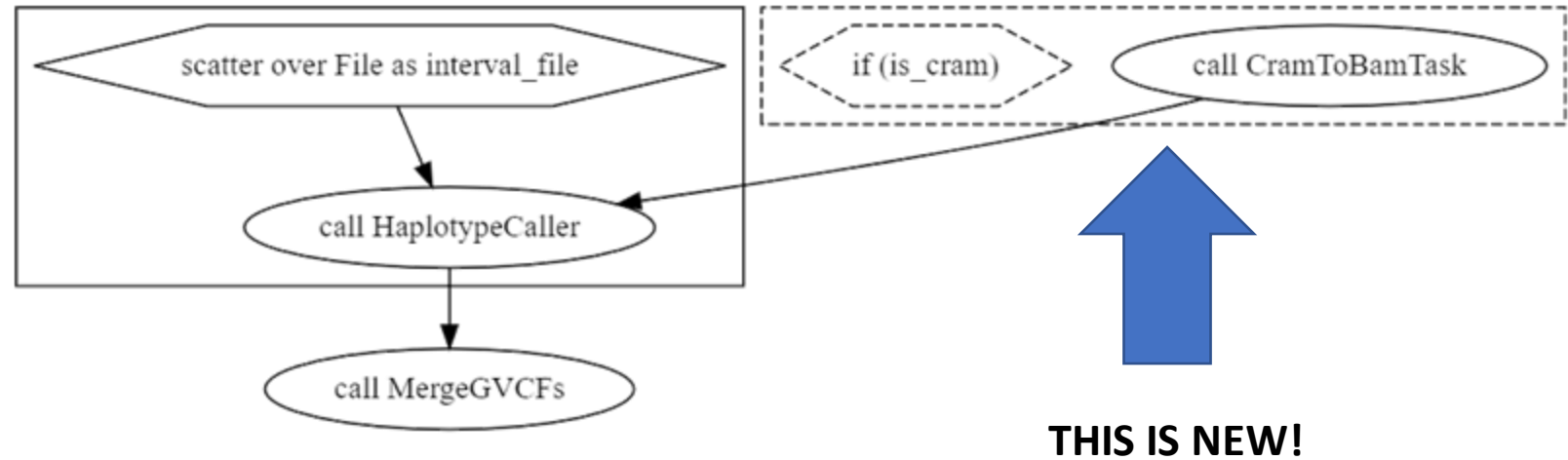


Workflow from previous and current chapter

ScatterHaplotypeCallerGVCF
from Chapter 8

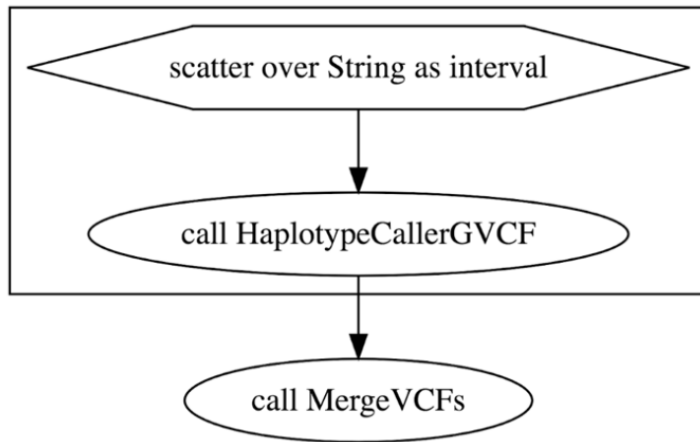


HaplotypeCallerGvcf_GATK4
from Chapter 9 (#1)

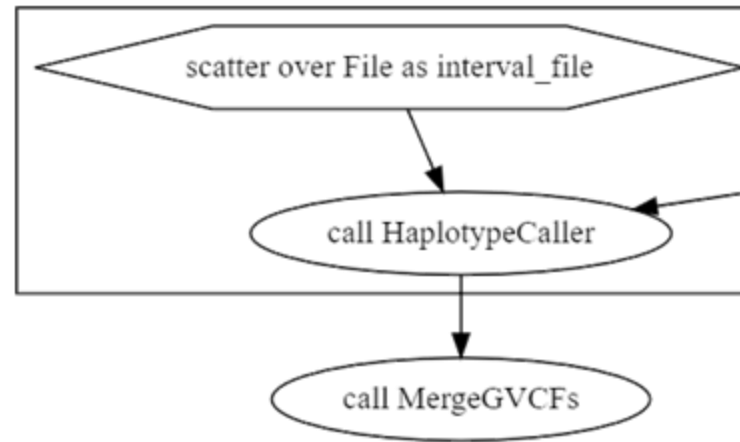


Workflow from previous and current chapter

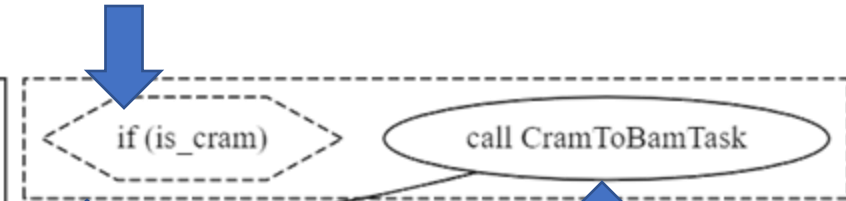
ScatterHaplotypeCallerGVCF
from Chapter 8



HaplotypeCallerGvcf_GATK4
from Chapter 9 (#1)



What does
if do?



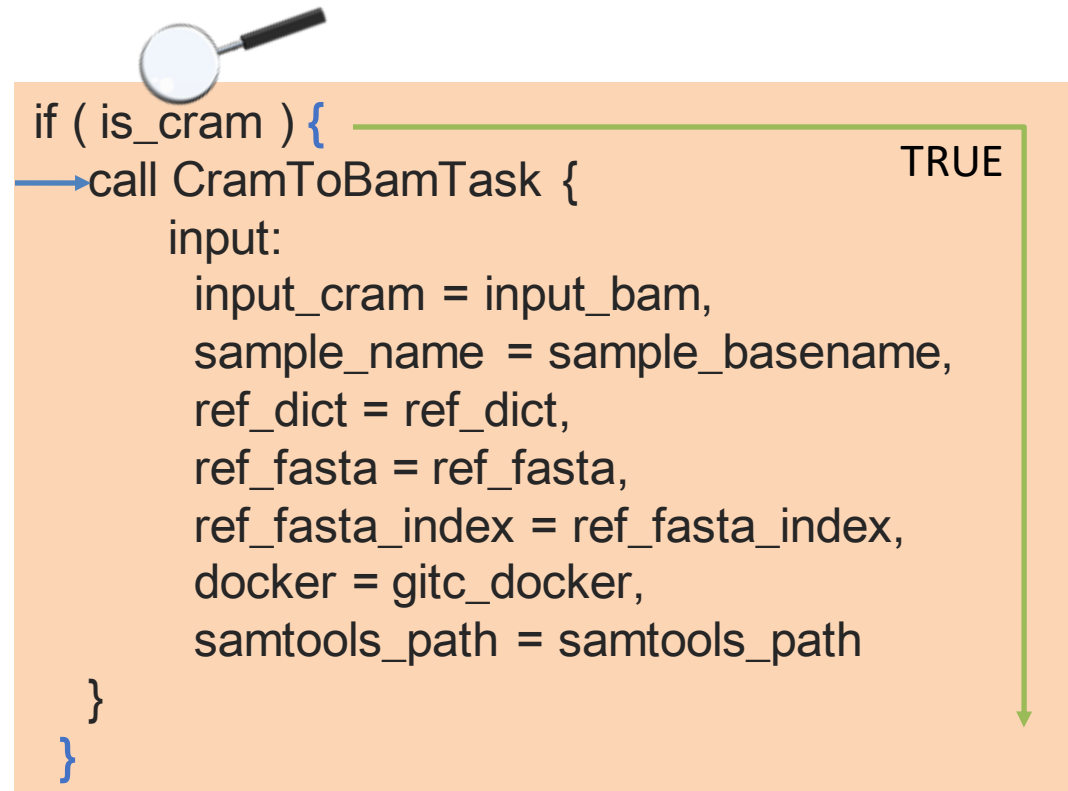
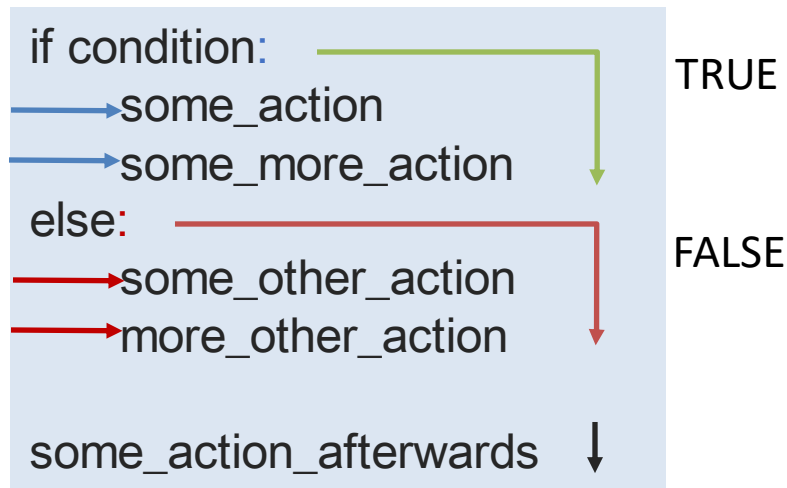
We know this is a
call to a task

Box suggests
control function,
but why dotted
lines.



If statement are ubiquitous in programming

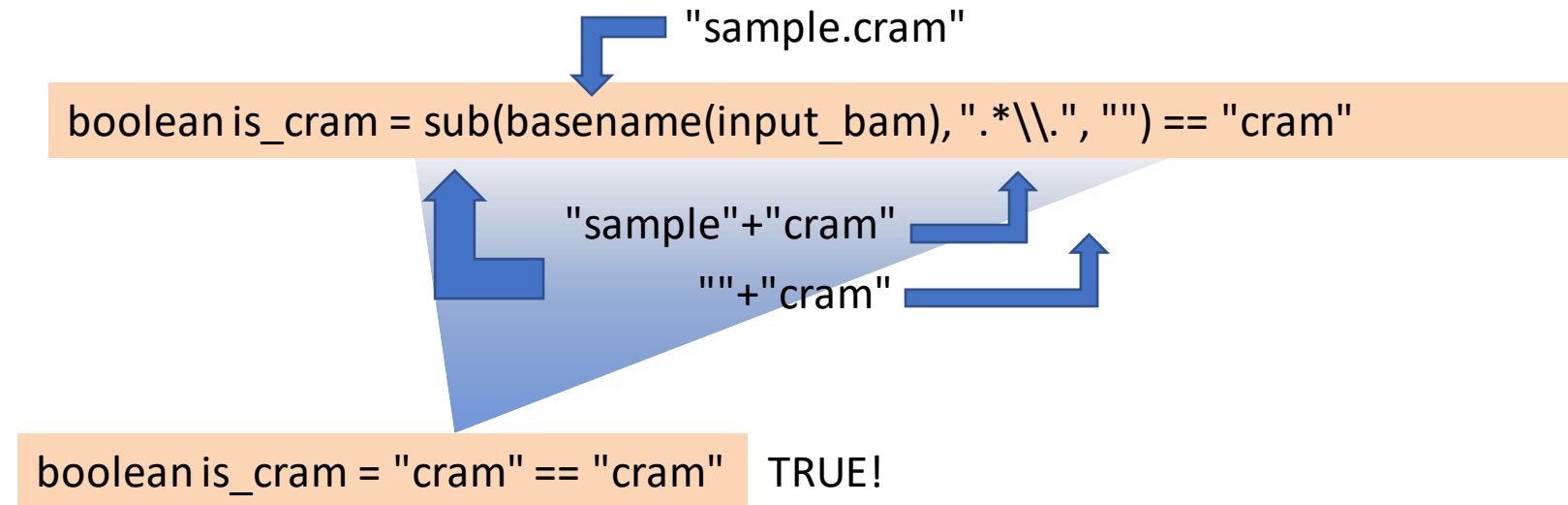
- A block of code is only executed if a given condition/expression is true (anything before or after is executed regardless)



So, is_cram or not?

File input_bam = "gs://my-bucket/sample.cram"

- We've encountered basename() before what does sub() do?



How else may our conditional statement affect our workflow?



select_first([this special case, that default])

- Two lines in HaplotypeCaller's input section caught our attention.

```
HaplotypeCaller{  
  input:  
    input_bam = select_first([CramToBamTask.output_bam, input_bam]),  
    input_bam_index = select_first([CramToBamTask.output_bai, input_bam_index]),
```



Try this first. **If** it doesn't exist... get this

- Another example: ? optional (it can be defined in a .json file/command line input...or not)

```
boolean? make_gvcf  
boolean making_gvcf = select_first([make_gvcf, true])
```

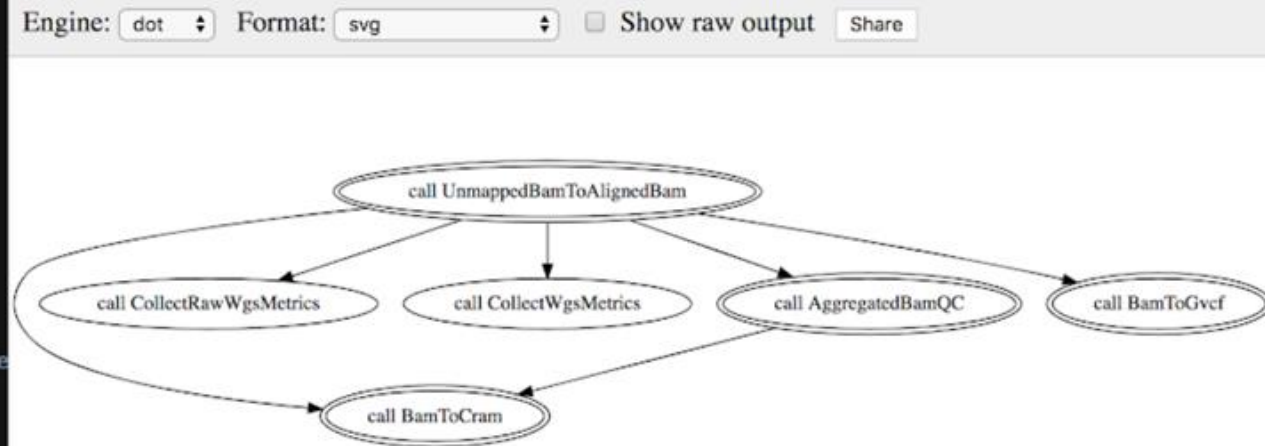


Try this first. Otherwise, use default value: true.



Mystery Workflow 2 in Graphviz Online

```
1 - digraph WholeGenomeGermlineSingleSample {  
2   #rankdir=LR;  
3   compound=true;  
4   # Links  
5   CALL_UnmappedBamToAlignedBam -> CALL_BamToCram  
6   CALL_UnmappedBamToAlignedBam -> CALL_CollectRawWgsMetrics  
7   CALL_UnmappedBamToAlignedBam -> CALL_CollectWgsMetrics  
8   CALL_UnmappedBamToAlignedBam -> CALL_AggregatedBamQC  
9   CALL_UnmappedBamToAlignedBam -> CALL_BamToGvcf  
10  CALL_AggregatedBamQC -> CALL_BamToCram  
11  # Nodes  
12  CALL_AggregatedBamQC [label="call AggregatedBamQC";shape="oval";peripheries=2]  
13  CALL_BamToGvcf [label="call BamToGvcf";shape="oval";peripheries=2]  
14  CALL_UnmappedBamToAlignedBam [label="call UnmappedBamToAlignedBam";shape="oval";peripheries=2]  
15  CALL_BamToCram [label="call BamToCram";shape="oval";peripheries=2]  
16  CALL_CollectRawWgsMetrics [label="call CollectRawWgsMetrics"]  
17  CALL_CollectWgsMetrics [label="call CollectWgsMetrics"]  
18 }
```

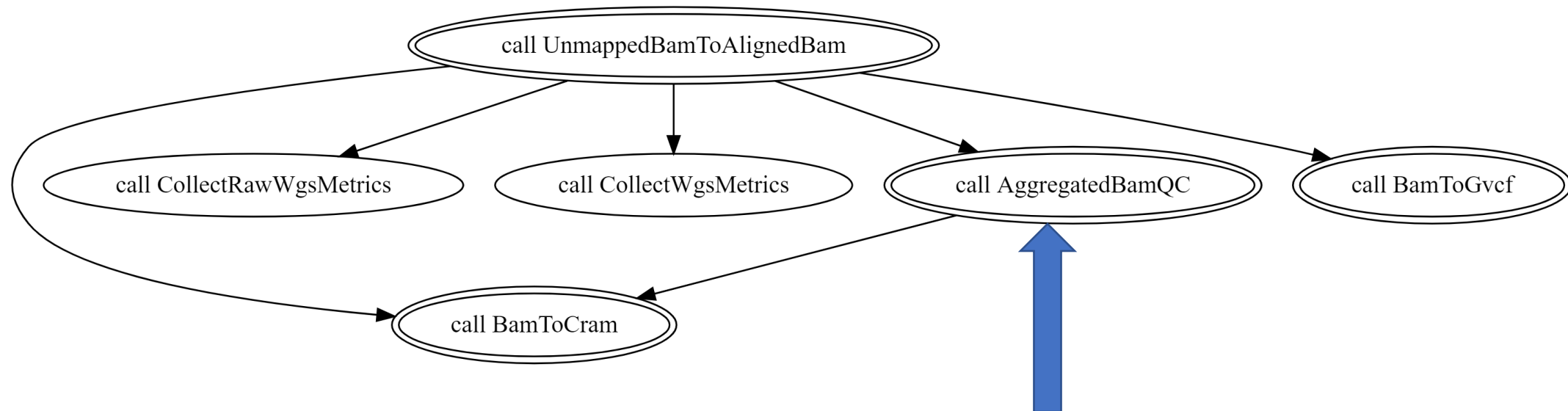


Modularity and Code Reuse?

No boxes

A lot smaller than we expected

WholeGenomeGermlineSingleSample (#2)





Ovals have double outline.



Where are the tasks?

- The calls to tasks look unusual compared to what we have seen so far.



```
call ToGvcf.VariantCalling as BamToGvcf {  
  input:   
    calling_interval_list= references.calling_interval_list,  
    evaluation_interval_list= references.evaluation_interval_list,  
    ...  
    contamination = UnmappedBamToAlignedBam.contamination,  
    input_bam= UnmappedBamToAlignedBam.output_bam,
```

- Then there's these import statements:

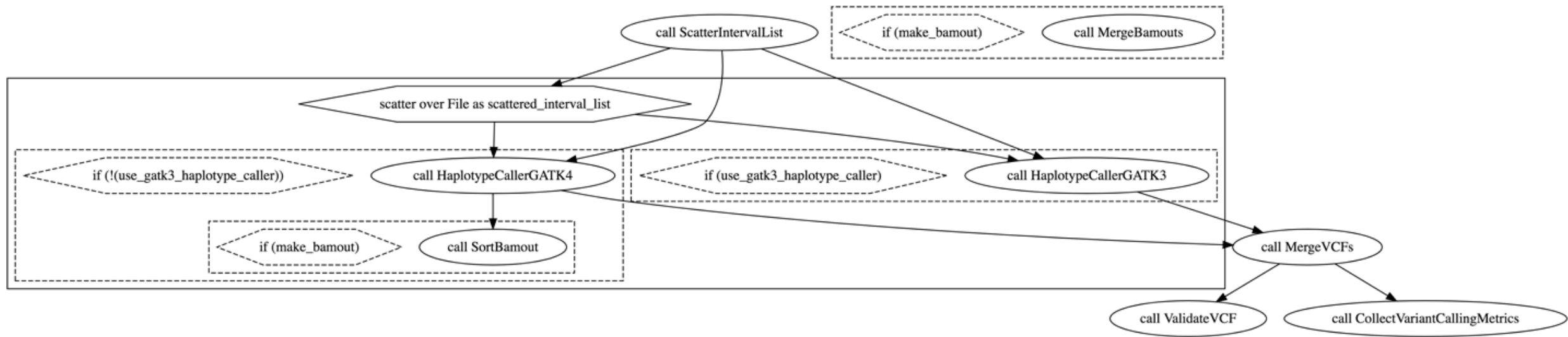
```
import ".././.././../tasks/VariantCalling.wdl" as ToGvcf
```



This is a subworkflow



Graph of VariantCalling.wdl workflow



Additional Resources

- A few more WDL resources:

- WDL 1.1 specifications! (h/t Geraldine):

- <https://github.com/openwdl/wdl/blob/main/versions/1.1/SPEC.md>

“There are no breaking changes in this version (i.e. you can change 'version 1.0' to 'version 1.1' in your WDLs and they will work with any runtime that supports v1.1).” –John Didion on Twitter.

- WDL has a Slack Channel for users!

- https://join.slack.com/t/openwdl/shared_invite/zt-ctmj4mhf-cFBNxliZYs6SY9HgM9UAVw

- Regular expression (if you're still curious about how sub() did its magic!)

- https://en.wikipedia.org/wiki/Regular_expression
 - <https://regex101.com>



A detailed illustration of a pufferfish, likely a species of pufferfish, shown in profile facing right. The fish has a dark, mottled pattern on its upper body and a lighter, more uniform pattern on its lower body. Its mouth is open, revealing small, sharp teeth. The background is a plain, light color.

Thank you for joining us today!

Next week: Chapter 10

Next meeting: February 8, 2021