OpenCAP Documentation

Release 1.0

Matthew Mosior

Aug 01, 2019

Contents:

OpenCAP

Blueprints for Precision Medicine

OpenCAP

Blueprints for Precision Medicine

CHAPTER 1

About

Welcome to the Open Source CIViC Annotation Pipeline (OpenCAP). This resource is a step-by-step tutorial that allows users to build a custom capture panel that is linked to clinical variant annotations. This capture panel development is accomplished using the publicly available CIViC database. Through the tutorial, we first describe the CIViC database and introduce users to relevant information within CIViC that can be used to build a capture panel. We then go through methods to build a CIViC capture panel using existing variants within the CIViC database. Specifically, variants of interest are identified using custom filters and variant coordinates are queried using CIViC's public API. Coordinates identified are then used to design custom capture probes that target variants of interest. After custom capture panel development, we provide general guidance on how to prospectively employ capture sequencing reagents on tumor samples. This includes sample preparation, nucleic acid isolation, library preparation, high throughput sequencing, and somatic variant calling. Once variants are identified using automated somatic variants calling and somatic variant refinement, we then show users how to link these variants back to the CIViC database to annotate the tumor sample for clinical relevance. Successful execution of this tutorial will provide users with a unique capture panel customized to the individual's need with linkage to clinical relevance summaries for all variants within the panel.

OpenCAP is intended for research use only and clinical applications of any panels designed using the SOP would require further panel validation and development in an appropriate setting (e.g., a CLIA certified and CAP accredited laboratory).

CHAPTER 2

CIViC introduction

2.1 What is CIViC?

The Clinical Interpretations of Variants in Cancer (CIViC) database is an open access, open source, community-driven web resource that captures clinically relevant variants in cancer. CIViC is built on evidence statements, whereby each statement summarizes a variant's potential clinical relevance as described by a publication. These evidence statements are summarized at the variant level and also at the gene level. An example of this hierarchy has been provided:

Clinical releva

2.2 What is an Assertion?

Evidence items can also be used to build Assertions. CIViC assertions aggregate individual evidence items into a single assessment of the clinical relevance of a variant in a specific disease setting. Assertions also allow for incorporation of guidelines (e.g., ACMG, FDA companion tests, drug approvals, AMP variant levels, NCCN guidelines, etc.). An example of a CIViC assertion is shown below:

2.3 Getting Started

Below we have provided a screencast entitled, CIViC - Getting Started. This screencast covers: * Description of CIViC and its goals * Navigating through CIViC's core pages * Browsing, searching, and consuming CIViC knowledgebase content

2.4 CIViC Resources

We have provided a variety of resources to introduce users to the CIVIC database. Please review the following information about CIViC and the CIViC team:

- Nature Genetics Publication
- CIViC Help Documents
- CIViC Glossary
- API Documentation

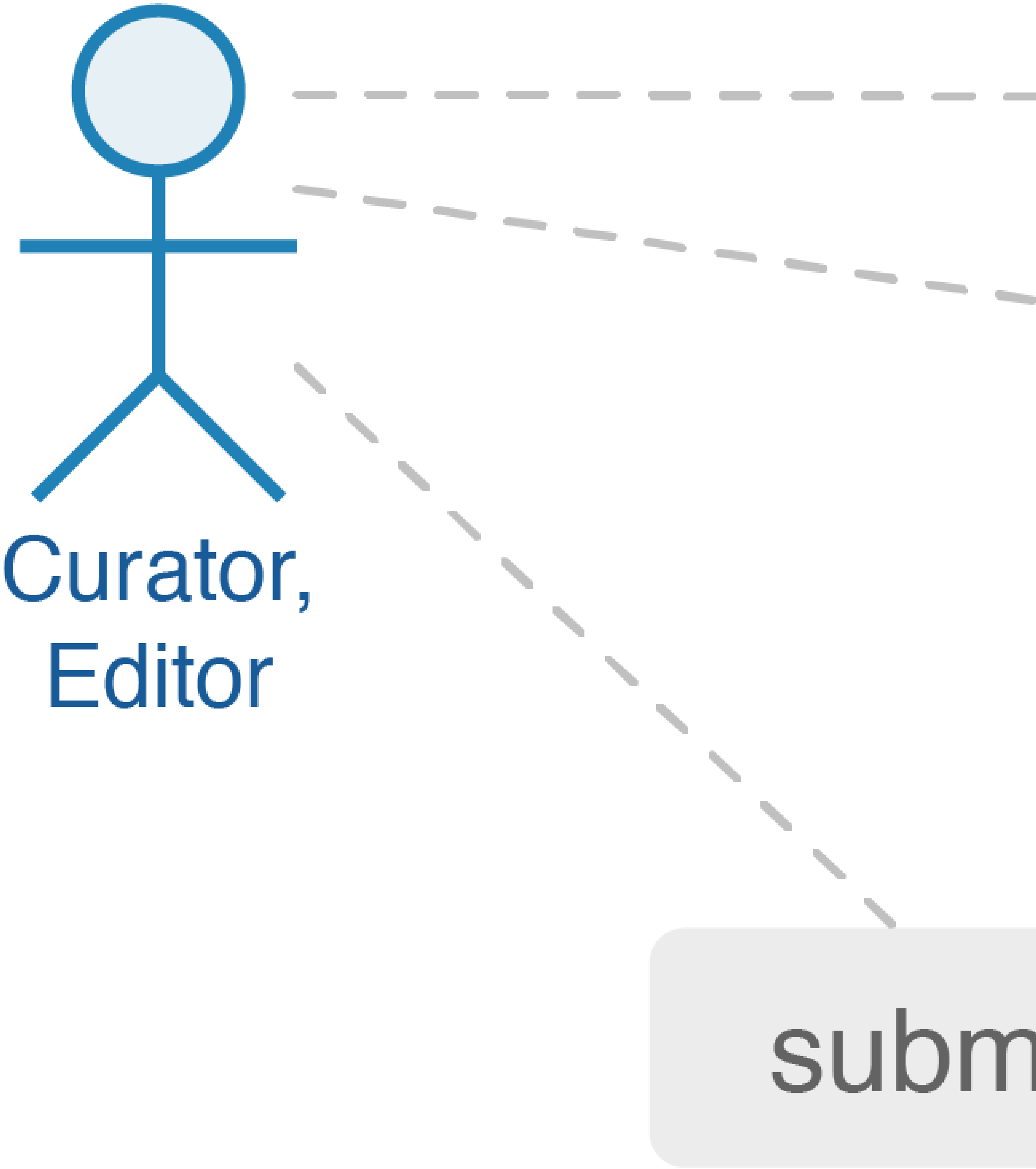
If you have further issues or wish to report a problem, feel free to email the CIViC team at help@civicdb.org

2.5 Contributing to CIViC

Any user can browse or search existing curated knowledge within the database. However, users must create an account and log-in to contribute new content to CIViC. Different types of contributions can be found under Example Activities on the CIViC help pages. These activities include:

- Adding evidence
- Contributing to variant or gene summaries
- Revising existing CIViC content
- Adding assertions
- Other curation tasks such as variant coordinate curation

Once a new evidence item or change to existing evidence is submitted to the CIViC database it will become visible (depending on user display preferences). However, the submission will be listed as a "submitted" or "pending" until it is accepted by an editor. CIViC editors must have attained a sufficient degree of relevant education (typically PhD or MD level), must be extensively familiar with the CIViC interface, have a demonstrated track record of successful curation within the database, and must be approved by two existing editorial members. More information on becoming an editor can be found on the Becoming an Editor Help Docs. An example of how a submitted or revised evidence item becomes accepted in CIViC is shown below:


Curator,
Editor
subm

Regardless of curator status, each activity is recorded in the database. Revision history can be viewed for all items within CIViC and personal contributions can be viewed on an individual's user profile. To promote user activity, CIViC badges can be earned for various curation actions and the Community Leaderboards show the top CIViC contributors, parsed by activity type.

2.5.1 Adding Evidence Items

The main curation activity in CIViC involves adding and editing evidence statements. Below we have provided a screencast entitled, Adding CIViC Evidence to walk users through creating an evidence item in CIViC. This screencast covers:

- Scanning a publication for curatable details
- Signing into CIViC to Add Evidence
- Walking through the Add Evidence form
- Viewing the submitted evidence

More information on evidence Items can be found on the CIViC Help Pages under Evidence. This guide provides detailed information on evidence statement inputs including: variant origin, evidence types, evidence levels, and evidence trust ratings. Additionally, when users add evidence items, we provide hints and helpful prompts in the right-hand column to assist with evidence submission.

2.5.2 Editing entities in CIViC

Any item in the CIViC interface can be edited using the pencil icon. TO DO: insert pencil icon Below we have provided a screencast entitled, Editing entities in CIViC to walk users through editing items (i.e., evidence, variants, genes or assertions) in CIViC. This screencast covers:

- Navigating to an entity's Edit Form
- Importance of edit comments
- Identifying entities with pending changes
- Navigating to an entity's suggested changes
- Reviewing entity revisions

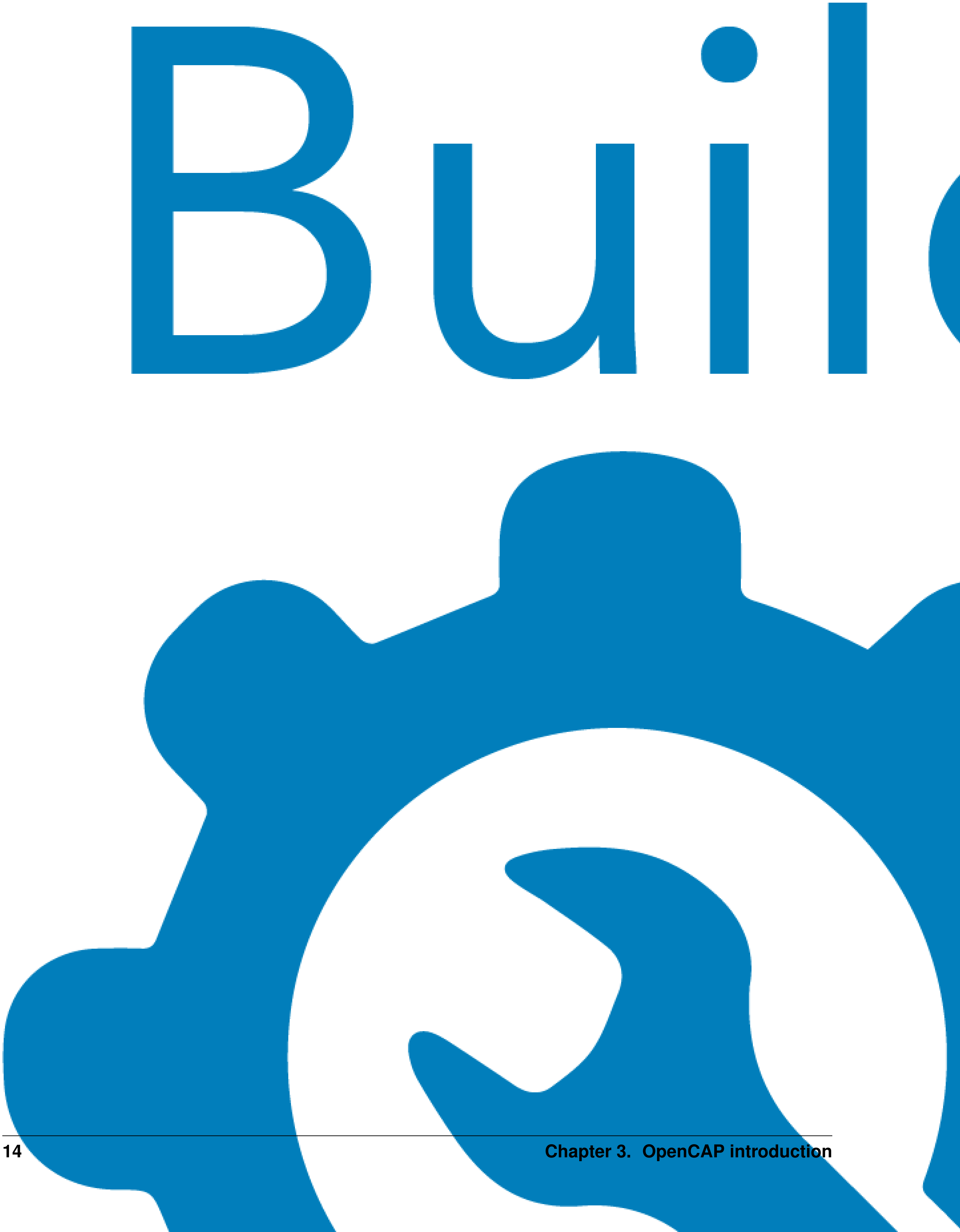
Buil

CHAPTER 3

OpenCAP introduction

The Open-sourced CIViC Annotation Pipeline (OpenCAP) is a tutorial that provides users with a method to develop a customized capture panel for which the variants contained within the panel are linked to clinical relevance summaries in the CIViC knowledgebase. This tutorial contains four chapters:

- **Build Custom Capture Panel**: In this section we will use the CIViC interface to identify variants of interest for custom capture. The interface will then be used to download variants of interest, associated clinical descriptions, and curated coordinates. We will then use an interactive interface (jupyter notebook) to format the variant coordinates for probe development. The output from this exercise will be a file that is compatible with commercial probe development companies for custom panel development.
- **Sequence Samples**: This section describes the massively parallel sequencing pipeline. We first detail methods for sample procurement and nucleic acid extraction. Subsequently we provide an overview of library preparation, target enrichment, and next-generation sequencing (NGS) on the Illumina platform. We also touch on new platforms for NGS, including PacBio and Oxford NanoPore. This high-level overview gives brief insight into how we employ custom capture reagents on tumor samples to enrich for sequence regions of interest.
- **Identify Somatic Variants**: The completion of the sequencing pipeline results in raw sequence files (e.g., FASTQ), which are text-based files that represent nucleotide sequences for individual reads. In this section, we briefly describe how to align sequencing reads to a reference genome, how to call somatic variants using automated software, and how to refine called variants using semi-automated processes.
- **Annotate Variants**: After defining a putative list of somatic variants associated with the patient's tumor, this section describes how to link variants back to the CIViC database to annotate the sample for clinical relevance. We again use an interactive jupyter notebook to import somatic variant calls and output an editable report for the user.


Buil

CHAPTER 4

Build Custom Capture Panel

In this section we will use the CIViC interface to identify variants of interest for custom capture. The interface will then be used to download variants of interest, associated clinical descriptions, and curated coordinates. We will then open an interactive jupyter notebook to reformat the variant coordinates for probe development. The output from this exercise will be a file that is compatible with commercial probe development companies for custom panel development.

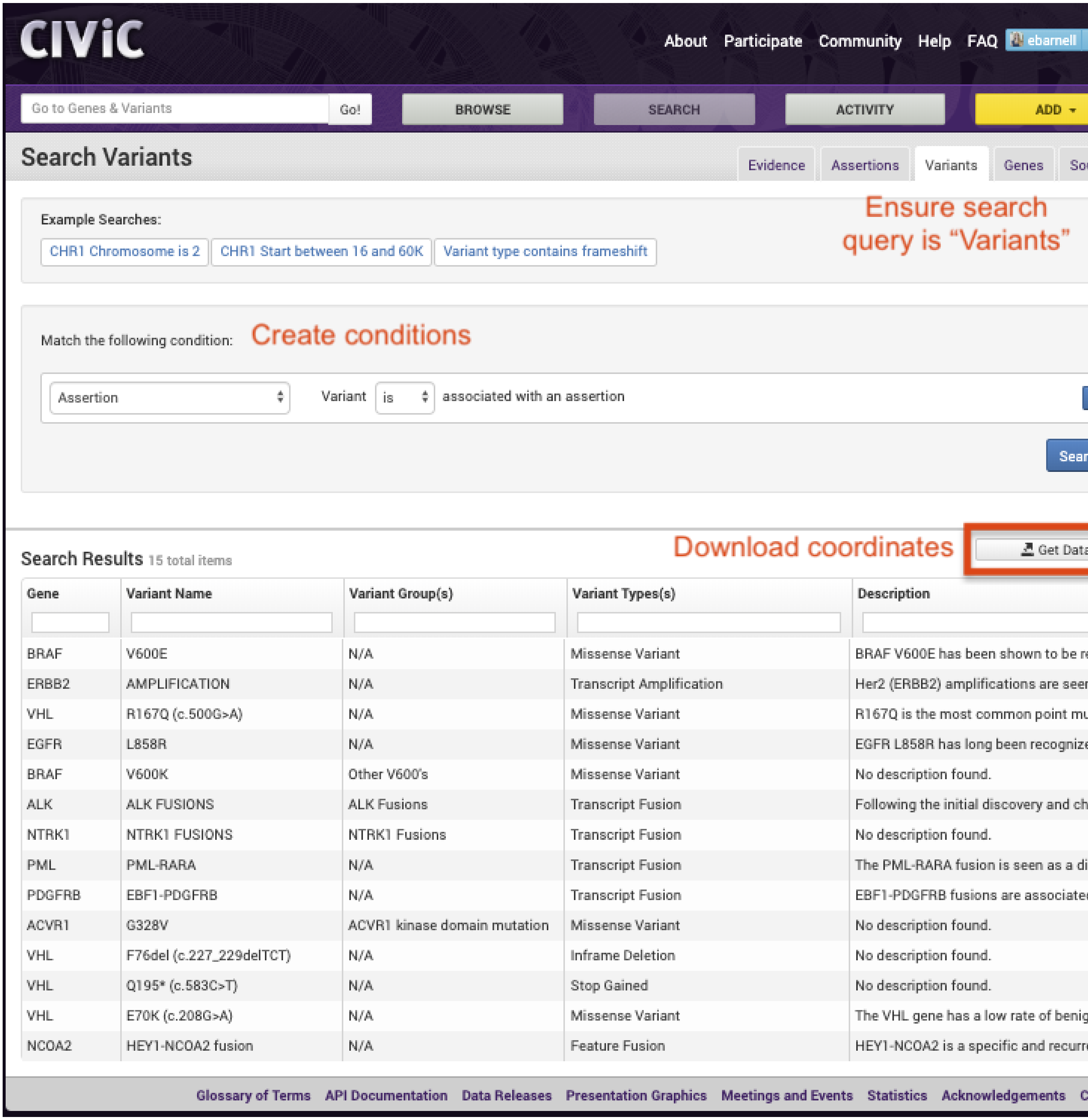
We have built a Binder Jupyter Notebook that contains code to pull in CIViC variants derived from the CIViC Search interface and create a list of genomic coordinates that require capture. While you are reading the tutorial, open the link provided below to start the process of loading a Binder Notebook (Note: loading the Jupyter Notebook can take 5-10 minutes):

Build Jupyter Notebook

4.1 Identify variants for capture

The CIViC database is constantly being updated with new evidence statements and assertions. Therefore, we have provided a real-time query interface that allows users to build a pool of variants required for custom capture. This interface can be accessed by going to the CIViC website, selecting the "SEARCH" button, and navigating to the Variants" tab: 'SEARCH-Variants.

To identify variants for capture, users can add conditions (i.e., search criteria) based on 28 predetermined fields in the drop down menu. If multiple conditions are employed, the user has the option to take the union (i.e., match any of the following conditions) or the intersection (i.e., match all of the following conditions). Additionally, after conditions are employed, the user has the option to further filter the selected variants using column headers in the search grid. The user can filter on Gene Name, Variant Name, Variant Group(s), Variant Types(s), or Description. Once the user is satisfied with the existing variants in the search grid, the user can export the data as a comma-separated values (CSV) file using the "Get Data" button in the search grid. This will provide users with information required to build probes for all variants selected. Below we have provided a screenshot showing how to filter on if the variant contains an assertion:



Search Variants

Ensure search query is "Variants"

Example Searches: CHR1 Chromosome is 2 | CHR1 Start between 16 and 60K | Variant type contains frameshift

Match the following condition: Create conditions

Assertion — Variant is associated with an assertion

Search Results 15 total items

Download coordinates

Gene	Variant Name	Variant Group(s)	Variant Types(s)	Description
BRAF	V600E	N/A	Missense Variant	BRAF V600E has been shown to be r
ERBB2	AMPLIFICATION	N/A	Transcript Amplification	Her2 (ERBB2) amplifications are seer
VHL	R167Q (c.500G>A)	N/A	Missense Variant	R167Q is the most common point mu
EGFR	L858R	N/A	Missense Variant	EGFR L858R has long been recognize
BRAF	V600K	Other V600's	Missense Variant	No description found.
ALK	ALK FUSIONS	ALK Fusions	Transcript Fusion	Following the initial discovery and ch
NTRK1	NTRK1 FUSIONS	NTRK1 Fusions	Transcript Fusion	No description found.
PML	PML-RARA	N/A	Transcript Fusion	The PML-RARA fusion is seen as a di
PDGFRB	EBF1-PDGFRB	N/A	Transcript Fusion	EBF1-PDGFRB fusions are associate
ACVR1	G328V	ACVR1 kinase domain mutation	Missense Variant	No description found.
VHL	F76del (c.227_229delTCT)	N/A	Inframe Deletion	No description found.
VHL	Q195* (c.583C>T)	N/A	Stop Gained	No description found.
VHL	E70K (c.208G>A)	N/A	Missense Variant	The VHL gene has a low rate of benig
NCOA2	HEY1-NCOA2 fusion	N/A	Feature Fusion	HEY1-NCOA2 is a specific and recurr

An example file output created from the CIViC interface can be viewed on the GitHub Repository https://raw.githubusercontent.com/griffithlab/civic-panel/master/binder_interactive/Build_Panel/test_create_variants.tsv

Although this screenshot provides one method to create a variant pool, there are many other examples of criteria that can be useful for identifying variants. Below we have provided a few additional examples of fields that might be helpful for building variant prioritization conditions. Each field has an associated description and links to help documents if applicable.

Field	Description	Example	Associated Help Documents
Assertion	Variant is affiliated with clinical assertions that incorporate multiple evidence statements	"Variant is associated with an assertion"	Assertion - TO DO
CIViC Variant Evidence Score	User can indicate minimum threshold for CIViC Variant Evidence Score	"CIViC Variant Evidence Score is above 20"	Variant Evidence Score - https://civicdb.org/help/variants/variant-evidence-score
Description	Variant description must contain a keyword	"Description contains colorectal cancer"	Variant Summary - https://civicdb.org/help/variants/variants-summary
Disease Implicated (Name)	Variant must contain at least one evidence item that is implicated in the desired disease	"Disease Implicated is Melanoma"	Disease Ontology - http://www.disease-ontology.org/
Evidence Items	User can indicate the required number of evidence items with a certain status	"Evidence Items with status accepted is greater than or equal to 5"	Evidence Monitoring - https://civicdb.org/help/getting-started/monitoring
Gene Name	Entrez gene name associated with variant must meet selected criteria	"Gene Name contains TP53"	Gene Name - https://civicdb.org/help/genes/genes-overview
Name	Variant Name must meet designated criteria	"Name does not contain AMPLIFICATION"	Variant Name - https://civicdb.org/help/variants/variants-naming
Pipeline Type	Variant type is associated with sequence ontology ID(s) that can be evaluated on designated pipeline	"Pipeline Type is DNA-based"	Variant Name - TODO
Variant Type(s)	Variant type, which is the assigned sequence ontology ID, must meet designated criteria	"Variant Type(s) does not contain Transcript Amplification"	Variant Type - https://civicdb.org/help/variants/variants-type

4.2 Categorize variants based on variant length

The CSV file developed using the CIViC Search interface contains the genomic coordinates that encapsulate the variants of interest (i.e., Custom CIViC Variants). Each line in this file represents a single variant that requires probe development. However, before designing probes for these variants, we must further categorize each variant by variant length. This can be accomplished by using CIViC curated coordinates (i.e., variant start position minus variant stop position plus one). If the variant length is less than 250 base pairs, the variant is eligible for hotspot targeting. If the variant is >250 base pairs, the variant requires tiling of the protein coding exons. For variants that require tiling, there are two different types of tiling.

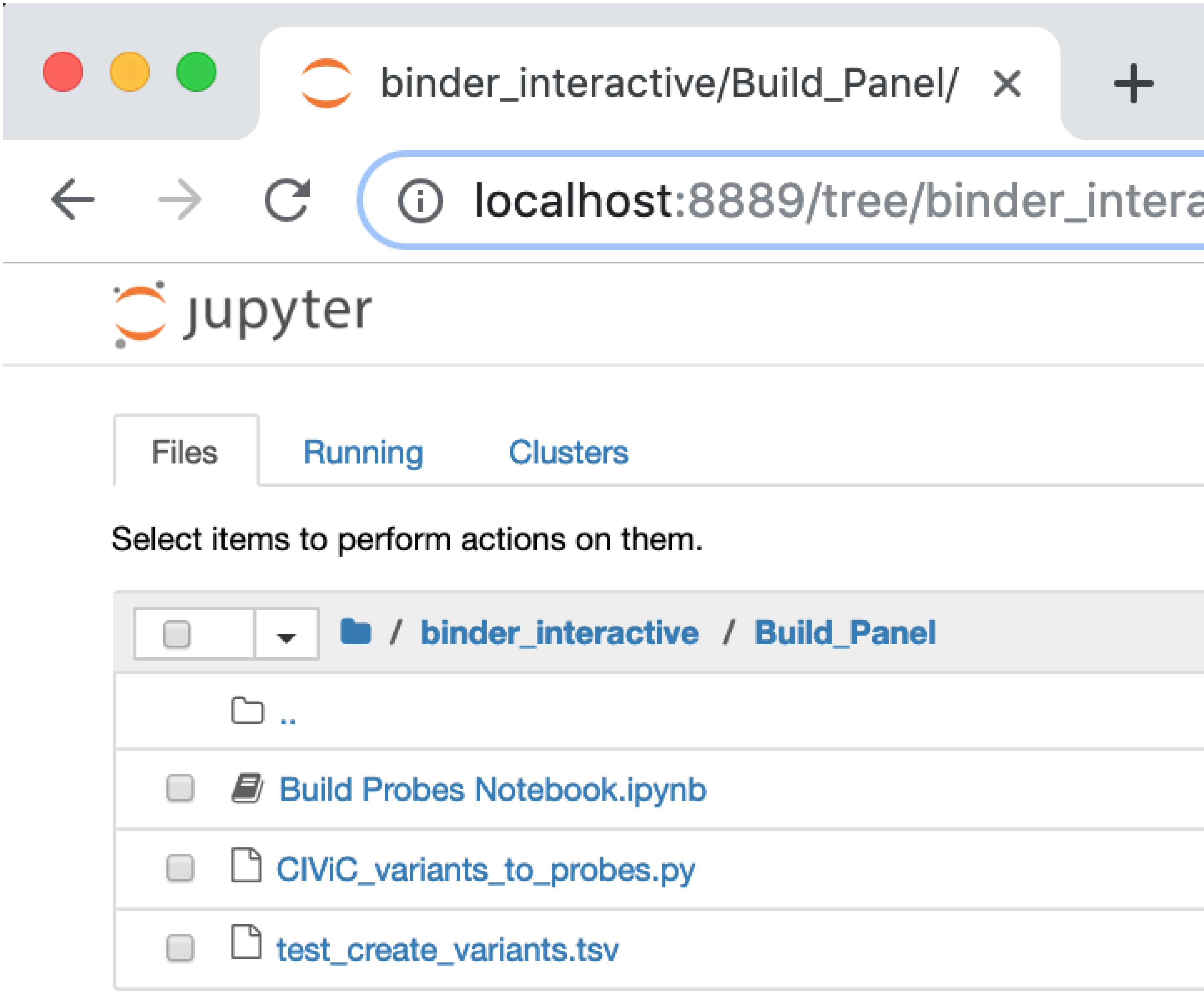
For variants that are large-scale copy number variants (e.g., "AMPLIFICATION", "LOSS", "DELETION", etc.), sparse tiling is appropriate. Sparse tiling requires creating approximately 10 probes spread across all protein coding exons. For variants that are bucket variants, (e.g., "MUTATION", "FRAMESHIFT MUTATION", etc.), full tiling is appropriate. Full tiling requires creating overlapping probes across the entire protein coding exon(s).

Ho

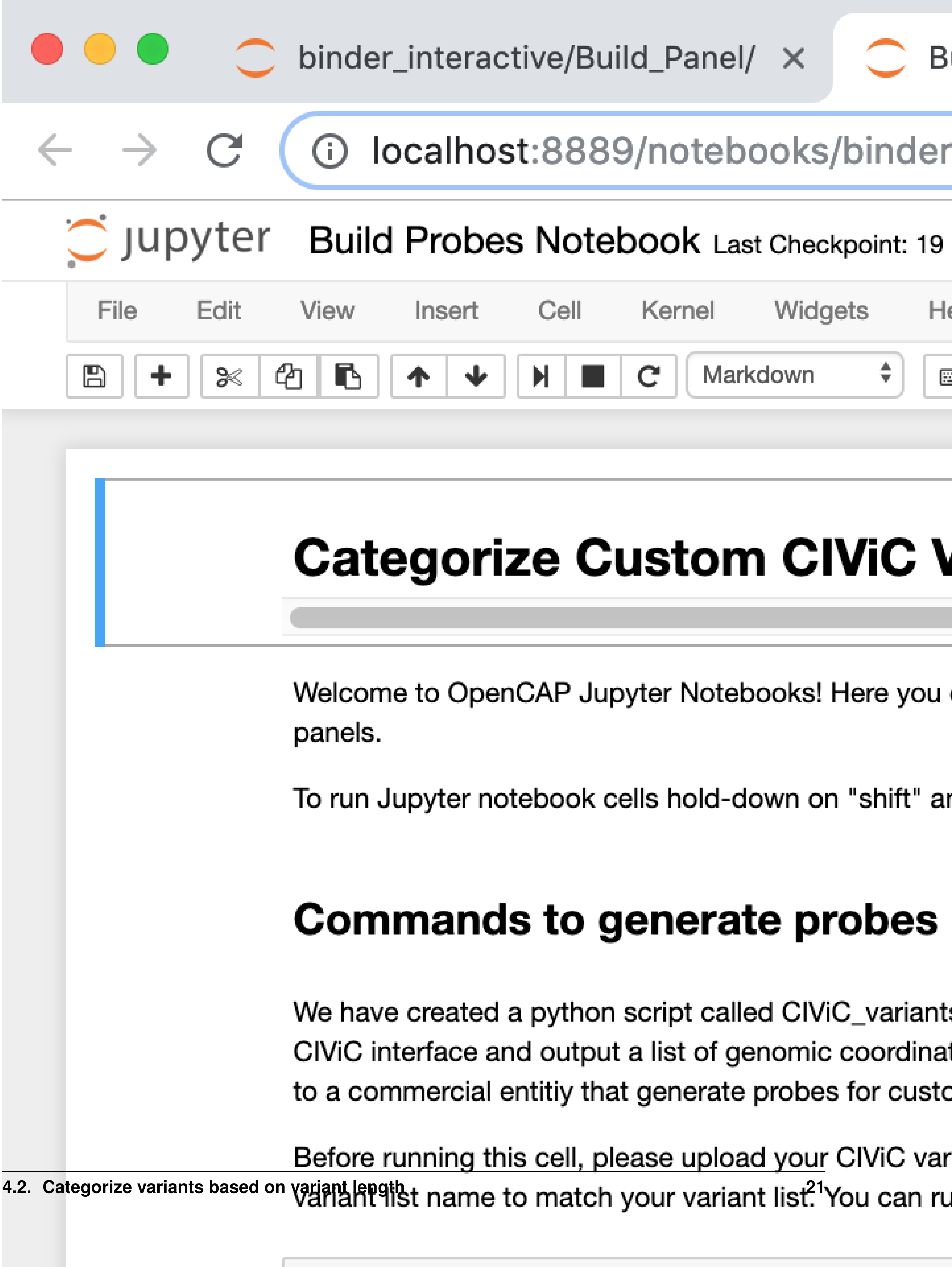
Hopefully you have already started building the Binder Notebook as recommended at the beginning of this page. If not please select the following link:

Build Jupyter Notebook

Once the Jupiter Notebook is loaded, you can add your Custom CIViC Variants file to the environment using the "Upload" button. See below:


binder_interactive/Build_Panel/
localhost:8889/tree/binder_intera
jupyter
Files
Running
Clusters
Select items to perform actions on them.
/ binder_interactive / Build_Panel
..
Build Probes Notebook.ipynb
CIViC_variants_to_probes.py
test_create_variants.tsv

To launch the Jupyter Notebook select the file entitled, “Build Probes Notebook” in the home directory. Selecting this file will direct you to a new tab in the same browser. Each Jupiter Notebook consist of cells which can contain text or code. Running a cell will execute entered functions. A cell can be run by selecting the cell, holding down the “Shift” key and pressing enter. Instructions for how to run the cells are also provided in the notebook. In the Jupyter Notebook cell that contains the python script, ensure that you change the input variant list file name to match the Custom CIViC Variants file that you uploaded to the home directory. The default file name is ‘test_create_variants.tsv’ - See below:


binder_interactive/Build_Panel/
Bu
localhost:8889/notebooks/binder
jupyter
File
Edit
View
Insert
Cell
Kernel
Widgets
He
Markdown

Build Probes Notebook Last Checkpoint: 19

Categorize Custom CIViC V

Welcome to OpenCAP Jupyter Notebooks! Here you
panels.

To run Jupyter notebook cells hold-down on "shift" ar

Commands to generate probes

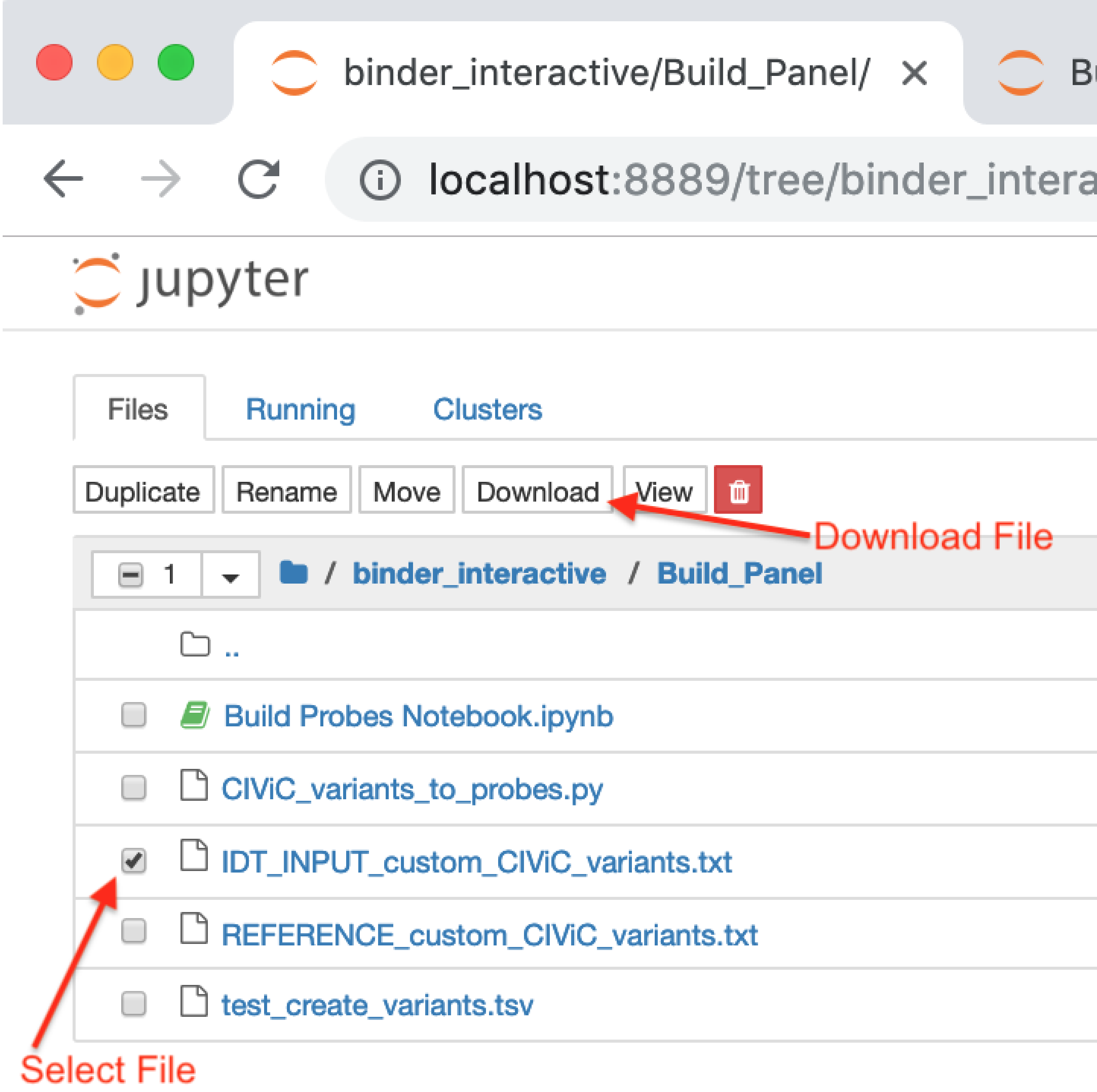
We have created a python script called CIViC_variant
CIViC interface and output a list of genomic coordina
to a commercial entitiy that generate probes for custo

Before running this cell, please upload your CIViC var
variant list name to match your variant list. You can ru

Once you have changed the input file name, hold down the “Shift” key and select “Enter” to process your Custom CIViC varian

- INPUT_custom_CIViC_variants.txt = coordinates for all probes required to capture variants of interest without annotation
- REFERENCE_custom_CIViC_variants.txt = list of all probes required to capture variants of interest with annotation (gene name, probe id, type of tiling)

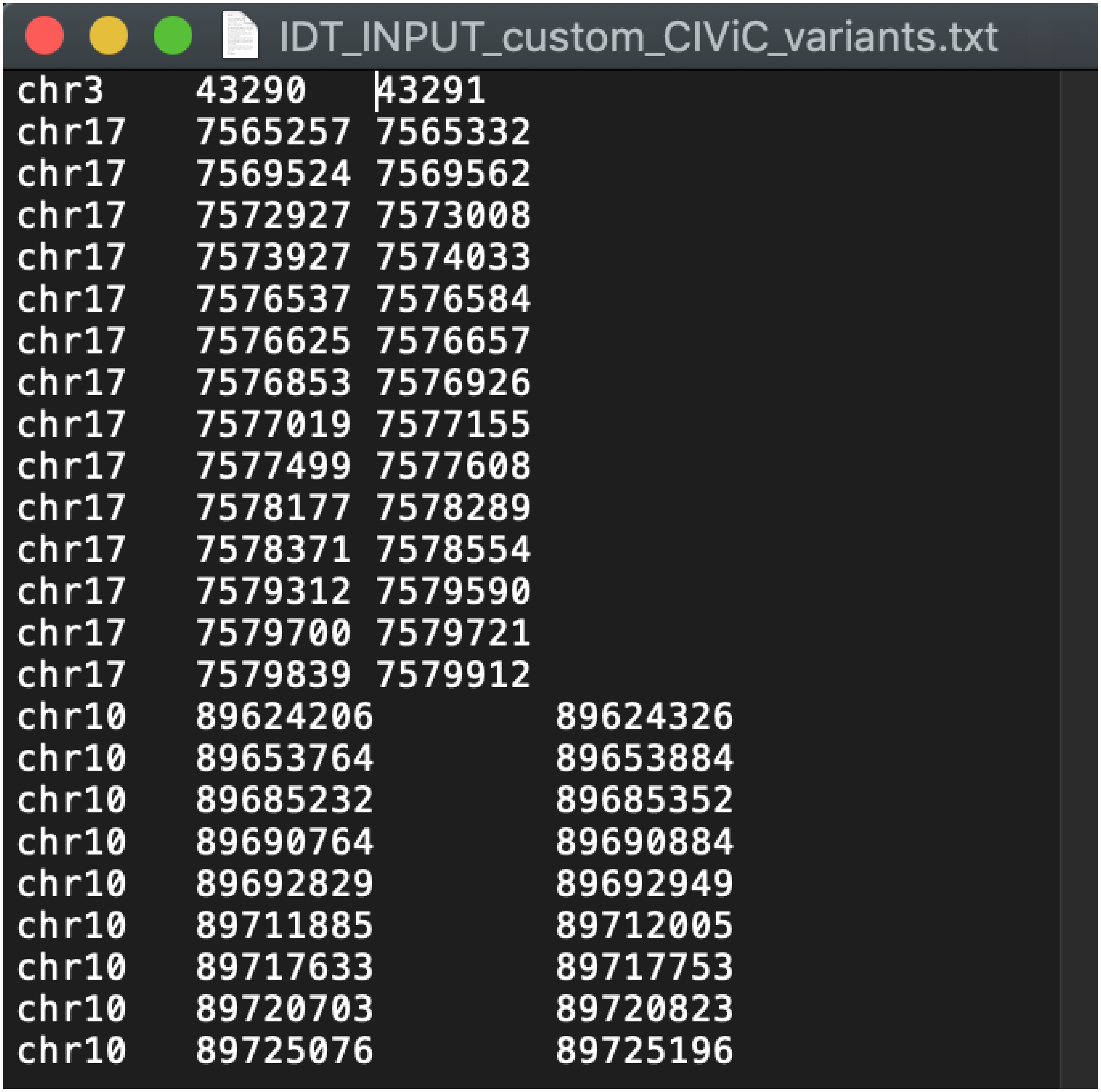
You can download these files to your local computer by checking the box next to the file of interest and selecting the “Download” button:


binder_interactive/Build_Panel/
localhost:8889/tree/binder_intera
jupyter
Files
Running
Clusters
Duplicate
Rename
Move
Download
View
1
/ binder_interactive / Build_Panel
..
Build Probes Notebook.ipynb
CIViC_variants_to_probes.py
IDT_INPUT_custom_CIViC_variants.txt
REFERENCE_custom_CIViC_variants.txt
test_create_variants.tsv

Download File

Select File

The file entitled “INPUT_custom_CIViC_variants.txt” will serve as an example input file, suitable for IDT probe design. This file should be a tab separated text file whereby each row represents a genomic region that requires coverage. An example file is shown below:



```
chr3	43290	43291
chr17	7565257	7565332
chr17	7569524	7569562
chr17	7572927	7573008
chr17	7573927	7574033
chr17	7576537	7576584
chr17	7576625	7576657
chr17	7576853	7576926
chr17	7577019	7577155
chr17	7577499	7577608
chr17	7578177	7578289
chr17	7578371	7578554
chr17	7579312	7579590
chr17	7579700	7579721
chr17	7579839	7579912
chr10	89624206		89624326
chr10	89653764		89653884
chr10	89685232		89685352
chr10	89690764		89690884
chr10	89692829		89692949
chr10	89711885		89712005
chr10	89717633		89717753
chr10	89720703		89720823
chr10	89725076		89725196
```

4.3 Build custom capture panel

After generating the INPUT_custom_CIViC_variants.txt file. You can access custom probe software provided by commercial entities for reagent development. Some of these entities include:

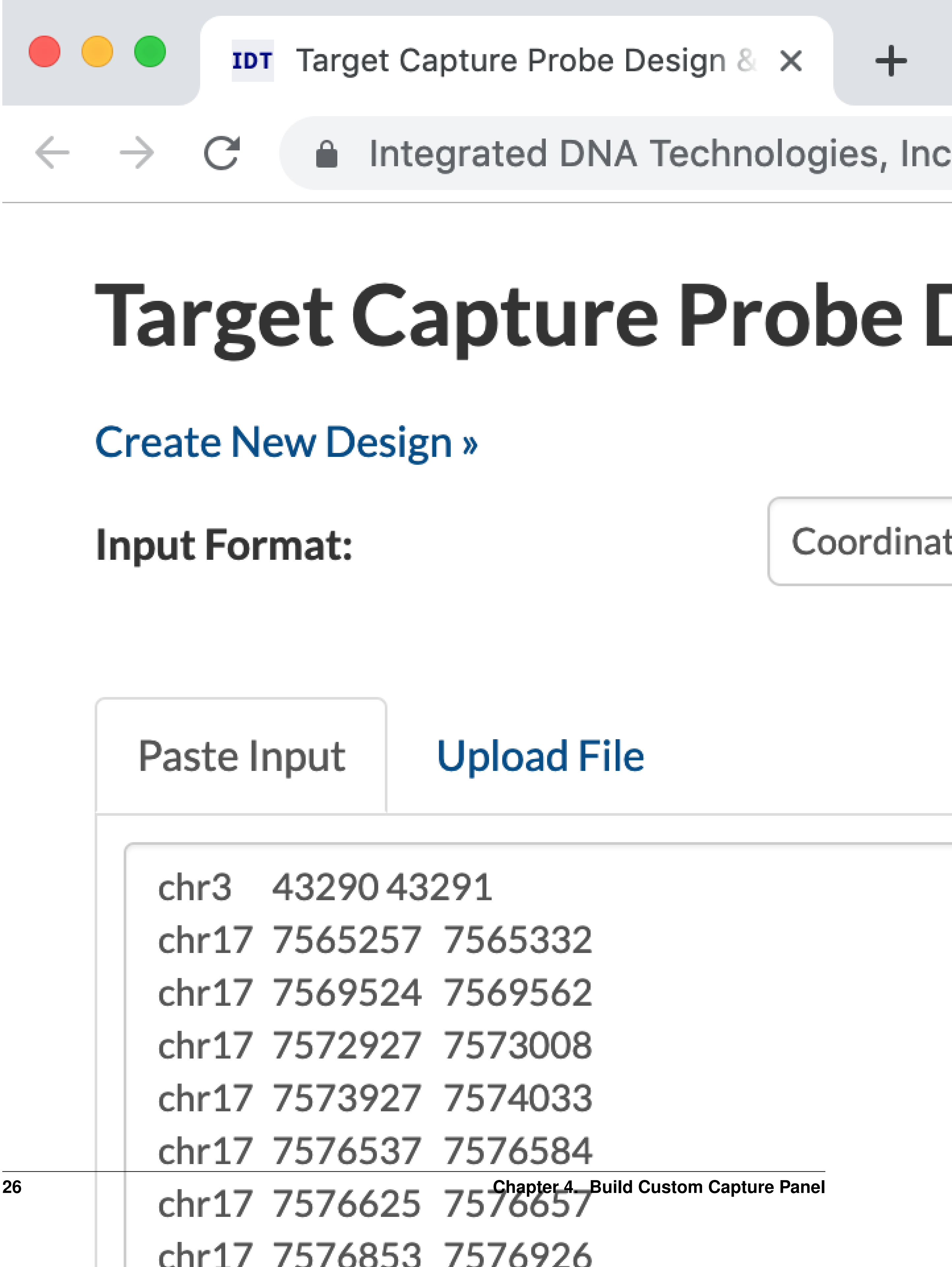
- Integrated DNA Technologies (IDT) Custom Probes
- Twist Biosciences Oligo Pools
- NimbleGen SeqCap

We will demonstrate custom capture panel development using the IDT Target Capture Probe Design & Ordering Tool. First, under "Input Format", select the "Coordinates (BED)" option. Next, select the "Upload File" option and click on the upload human genomic coordinates button. Upload the file that was prepared using the CIViC interface (INPUT_custom_CIViC_variants.txt).

We also recommend looking at the Design Parameters to ensure proper capture design. Ensure that the following parameters ar

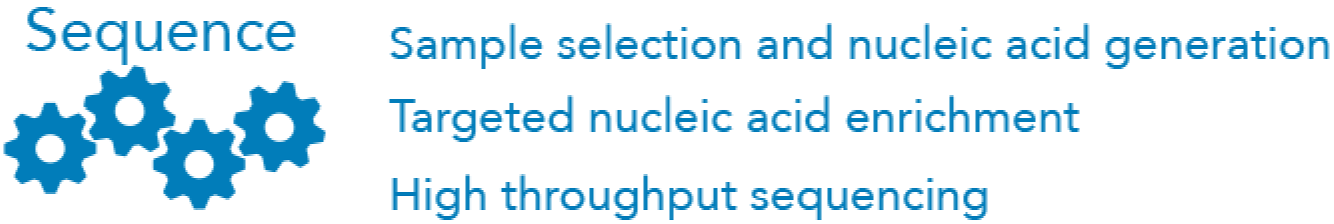
- Target species = "Human (Feb. 2009 GRCh37/hg19)"
- Target Definition = "Full Region"
- Probe Length = 120 basepairs
- Probe Tiling Density = 1X

Successful upload of the text file should look like this:



```
chr3    43290 43291
chr17  7565257  7565332
chr17  7569524  7569562
chr17  7572927  7573008
chr17  7573927  7574033
chr17  7576537  7576584
chr17  7576625  7576657
chr17  7576853  7576926
```

Once the files are successfully uploaded, select the "Continue" button to develop the reagent. Of note, you must be logged into the interface to continue with this process. The next steps include reviewing the design, ordering probes, ordering buffers, and ordering blocking oligos. Once the panel design has been reviewed, you can purchase the reagents through the IDT interface.


Sequence
Sample selection and nucleic acid generation
Targeted nucleic acid enrichment
High throughput sequencing

CHAPTER 5

Sequence Samples

This section describes the massively parallel sequencing pipeline. We first detail methods for sample procurement and nucleic acid extraction. Subsequently we provide an overview of library preparation, target enrichment, and next-generation sequencing (NGS). We also touch on new methods for NGS, which includes describing PacBio and NanoPore Sequencing. This high-level overview gives brief insight into how we employ custom capture reagents on tumor samples to enrich for variants of interest. Note, there are many variations of the following sequencing pipeline that may be appropriate for the individual researcher or clinical use case. This section is meant to provide a general overview of a typical pipeline.

5.1 Sample procurement

For the analysis described here, samples must be derived from a germline tissue (normal sample) and a diseased tissue (tumor sample). Procuring samples from these two sample types requires consideration of the malignancy:

- **Liquid Cancers**: A cancer that begins in blood-forming tissue, such as the bone marrow, or in the cells of the immune system (e.g., leukemia, multiple myeloma, and some lymphomas)
- **Solid Cancers**: An abnormal mass that does not contain cysts or liquid areas (e.g., sarcomas, carcinomas, and some lymphomas)

Sc

It is important to note that blood samples may not be suitable as the normal samples for solid cancers if the tumor is metastatic with high circulating tumor cells or circulating tumor DNA. In these cases, buccal swabs or skin biopsies may be better for tumor-normal comparisons.

5.2 Sample storage

Once samples are procured, they are typically preserved as fresh-frozen (FF) or formalin-fixed paraffin-embedded (FFPE) specimens.

- **Fresh Frozen**: As soon as samples are obtained, fresh frozen preparation requires exposing the sample to liquid nitrogen as quickly as possible. Samples must subsequently be stored in 80°C freezers until extraction.
- **Formalin-fixed paraffin-embedded**: FFPE samples can be prepared using a variety of available kits (e.g., QIAamp DNA FFPE Tissue Kit, MagMAX™ FFPE DNA/RNA Ultra Kit, Quick-DNA/RNA FFPE Miniprep Kit, etc.).

5.3 Nucleic acid generation

If samples are stored as FFPE blocks, they require FFPE DNA extraction. This can be accomplished using commercially available kits. In general, these kits require paraffin removal and tissue rehydration, tissue digestion, mild reversal of cross-linkage, and nucleic acid purification. If samples are stored as fresh-frozen tissue blocks, they only require nucleic acid purification.

Nucleic acid purification requires cell lysis, binding of nucleic acid, washing off non nucleic acid material, drying of nucleic acid, and elution into a buffer. There are many commercially available kits that can perform nucleic acid purification. These steps can also be automated using commercially available equipment (e.g., QIAsymphony® SP, NUCLISENS® easyMAG®, etc.). Below we describe each step in detail:

1) **Lyse**: Tissue samples are typically stored as whole cells. The lysis step is used to disrupt the cellular membrane to expose the nucleic acid. Lysis buffers typically comprises a chaotropic agent, which breaks the hydrogen bond network between water molecules and optionally a surfactant to lower surface tension between membrane components and nucleic acid-containing solution. Some chaotropic agents can include: guanidium thiocyanate or magnesium chloride. Some surfactants can include: Triton-X-100, or sodium dodecyl sulfate.
2) **Bind**: After nucleic acid has been suspended in solution, it can be reversibly bound to a positively charged material for purification. These materials can include magnetic particles, columns, filters, silica beads, or organic solvent-based methods.
3) **Wash**: Once the nucleic acid is bound to a positively charged material, remaining substances in the lysate are washed from solution. A washing solution does not disrupt the covalent bond between the nucleic acid and the positively charged material used for purification.
4) **Dry**: To ensure proper elution, bound nucleic acid typically needs to be completely devoid of all liquid. To avoid degradation, alcohols can be used to expedite the drying step.
5) **Elute**: Elution buffers are solvents that displace the nucleic acid from the positively charged material used for purification. Elution buffers can include: 10 mM Tris at pH 8-9, Warmed MilliQ (60 oC), or 1X TE.
6) **Cleanup**: Elutions can be optionally treated with either RNAse (RNase ONE™ Ribonuclease, RNase A, etc.) or DNAse (e.g., *DNase I* <https://www.neb.com/protocols/0001/01/01/a-typical-dnase-i-reaction-protocol-m0303>, Baseline-ZERO™ DNase, etc.) to eliminate nucleic acid that is not being used in downstream analysis.
7) **Quality check**: After the nucleic acid generation step, it is recommended to assess the quantity and quality of the final elution. This can be accomplished using spectrophotometry and/or electropherograms.

- **Spectrophotometry** measures a substance's ability to absorb a specific wavelength, which in turn is a proxy for concentration and purity. First, the sample is exposed to an ultraviolet light at a wavelength of 260 nanometres (nm) and the DNA and RNA in the sample will absorb a relative amount of the light that is proportional to the concentration. Next a photo-detector measures the light that passes through the sample (i.e., not absorbed), which allows you to calculate the quantification of DNA/RNA in the sample. Nucleic acid quantification using spectrophotometry relies on the Beer–Lambert law.
- **Electropherograms** measure the nucleic acid concentration and size using a fluorescent spectrum. The Agilent 2100 Bioanalyzer evaluates the sample using a spectrum of fluorescents. The Qubit 4 Fluorometer utilizes fluorescent dyes that are specific to the target of interest.

5.4 Library construction

In advance of next-generation sequencing (NGS), construction of sequencing libraries is first required. This typically requires genomic fragmentation, ligation to custom linkers called adapters, and polymerase chain reaction (PCR) amplification.

1) **Genome fragmentation involves breaking the DNA into smaller pieces using physical or chemical means.**

    - Physical fragmentation methods include sonication, nebulization or enzymatic reactions.
    - Chemical fragmentation relies on hydroxyl radicals to break DNA into fragments, which can accommodate more material, but can induce false positives through novel mutations or transversion artifacts.

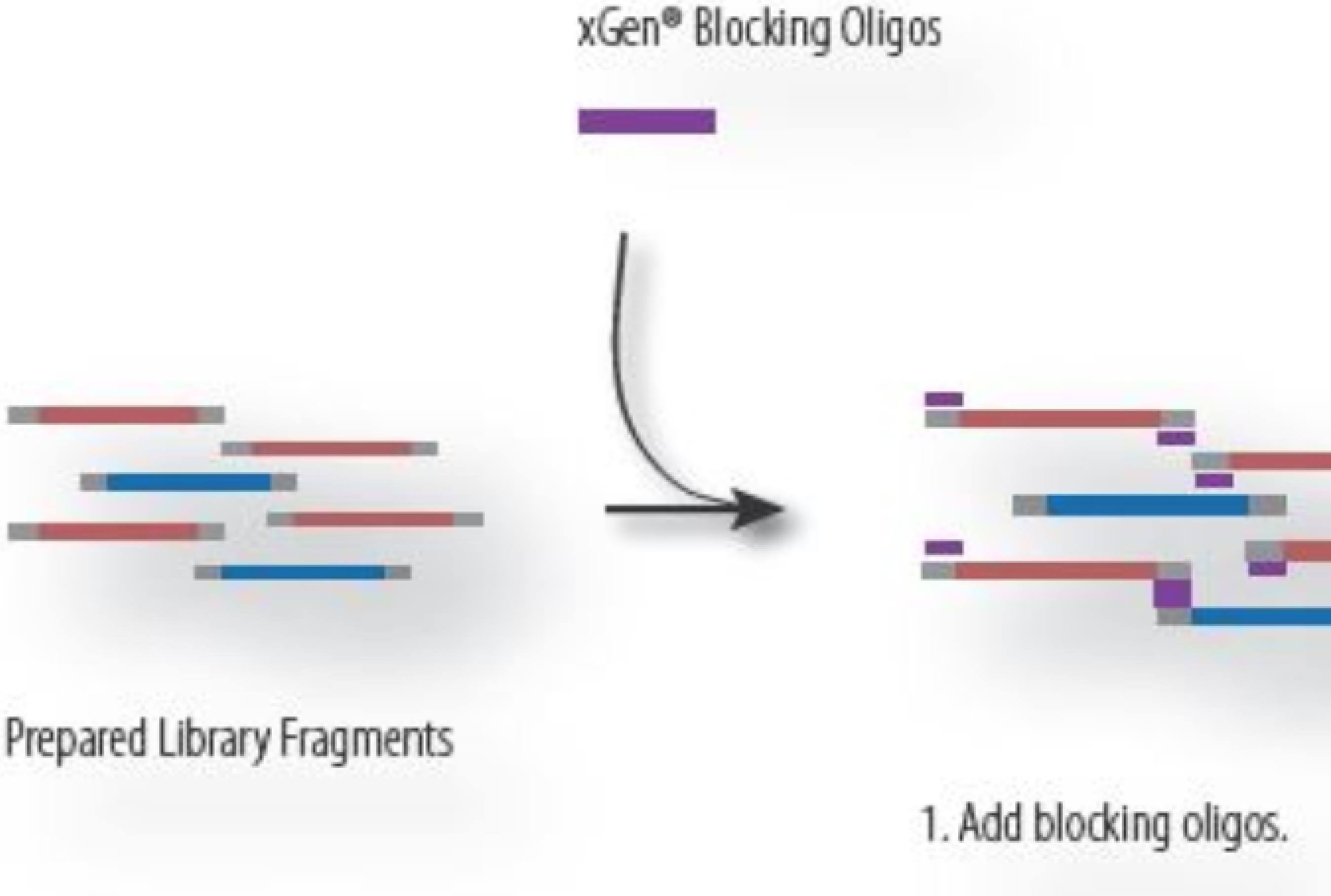
2) **Adaptors** are chemically synthesized double stranded DNA molecules that make sequencing reactions possible. Adaptors are ligated to DNA fragments and may include sequences to allow binding to a flowcell, sequencing primer sites, sample indexes, unique molecular identifier (UMI) sequences, etc.
3) **PCR amplification** is a method to make many copies of a specific DNA segment. PCR requires first denaturing dsDNA to create ssDNA using heat, binding of targeted primers to ssDNA fragments, and elongation of ssDNA to create a copied dsDNA. Amplification is typically performed at multiple steps in the sequencing pipeline.

5.5 Target enrichment strategies

Target enrichment strategies are used to generate a specific collection of DNA fragments for sequencing. These enrichment strategies are often performed on the constructed sequence library or incorporated into a library construction step.

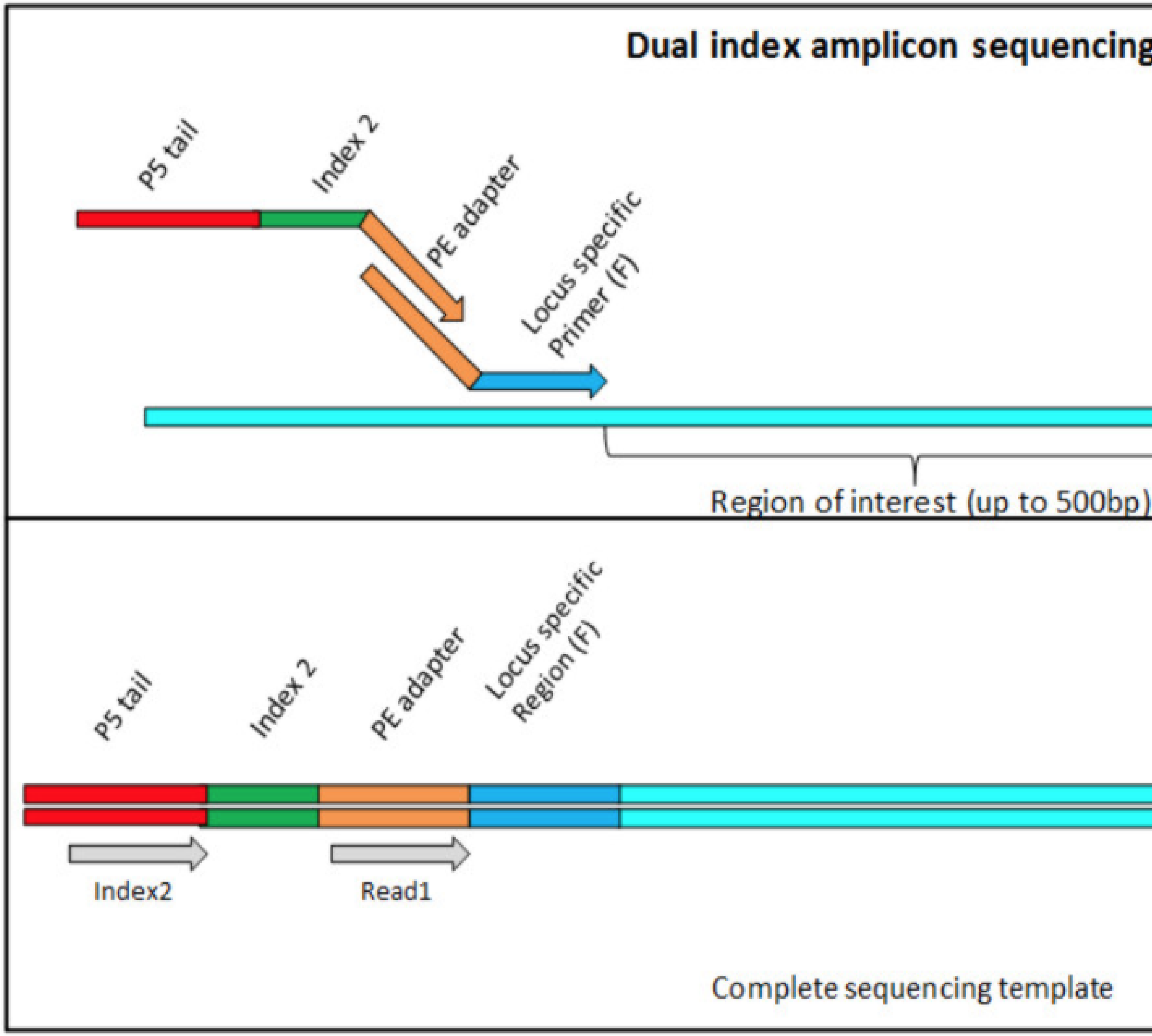
5.5.1 Hybridization Capture

Hybridization capture requires designing specific primers that bind to regions of interest and isolating these bound DNA fragments using chemistry (e.g., use of strepavidin Beads in combination with biotinylated DNA). Genomic DNA that is not bound to the capture probes will be washed away. The remaining DNA, which is enriched for regions of interest, is amplified using PCR and sequenced. Reagents that use hybridization capture include: Swift BioSciences, IDT, Agilent, among others. The process for hybridization capture is described below:



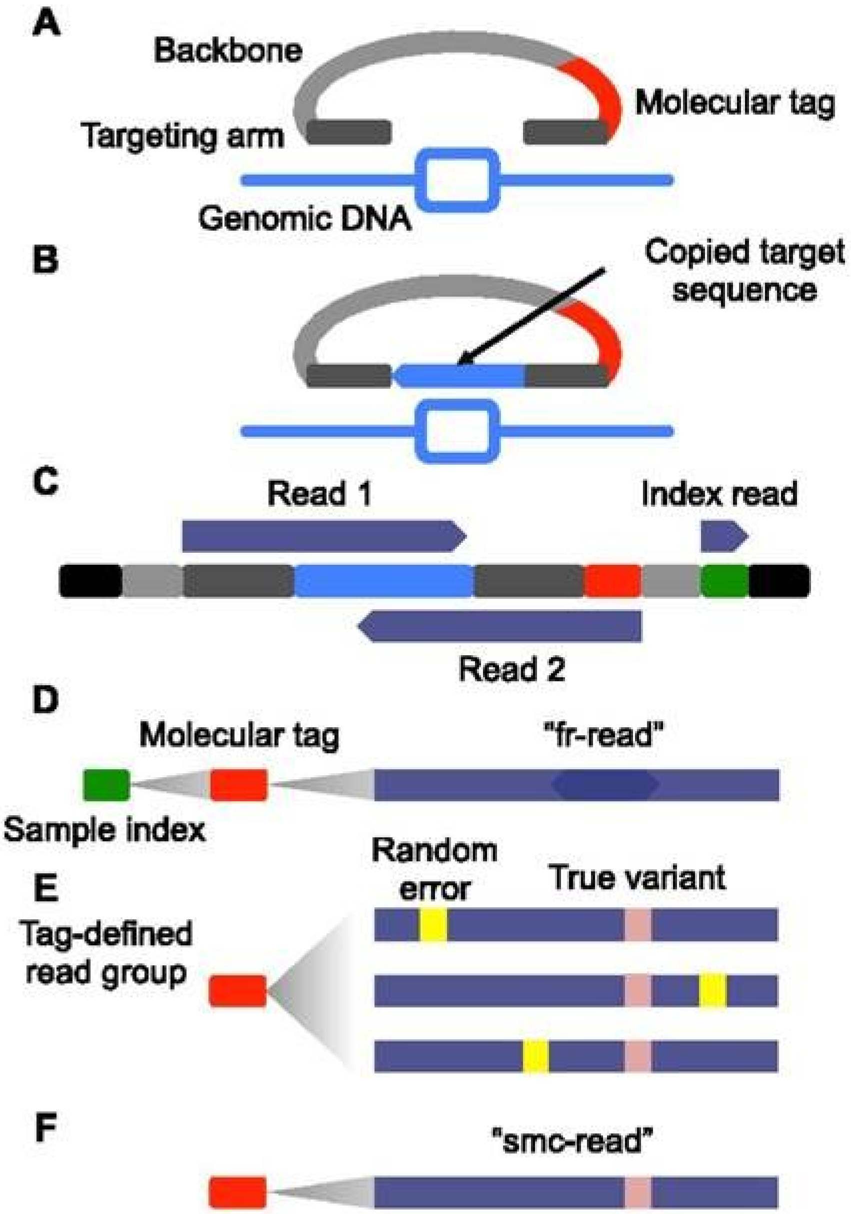
5.5.2 Amplicon Enrichment

Amplicon enrichment uses a slightly different strategy for enrichment of regions of interest. Instead of hybridization based capture, regions of interest are amplified by PCR using sets of primer sequences designed to target regions of interest. Reagents that use amplicon sequencing include: QIAGEN, Illumina, and others. An example of the process of amplicon enrichment is shown below:


Dual index amplicon sequencing
P5 tail
Index 2
PE adapter
Locus specific Primer (F)
Region of interest (up to 500bp)
P5 tail
Index 2
PE adapter
Locus specific Region (F)
Index2
Read1
Complete sequencing template

5.5.3 Unique Molecular Identifiers

Unique molecular identifiers (UMIs) are short sequences or molecular tags that can be added to each read during library preparation. Typically, these molecular identifiers are added prior to amplification so that they tag individual DNA molecules observed in the sample. This allows the individual to assign all amplification products to a single originating DNA molecule after sequencing. Through a process of consensus read formation, individual sequencing-related errors can be discounted, decreasing the effective error-rate of sequencing. UMI-based sequencing can take on many forms, each unique to the individual library preparation. An example of the single molecule molecular inversion probe approach is provided below:



5.5.4 Other considerations

Of note, for evaluation of RNA, total RNA must be subjected to reverse transcriptase treatment (e.g., ProtoScript® II Reverse Transcriptase, SuperScript™ III Reverse Transcriptase) to generate cDNA prior to library preparation.

5.6 High throughput sequencing

5.6.1 Next-generation sequencing

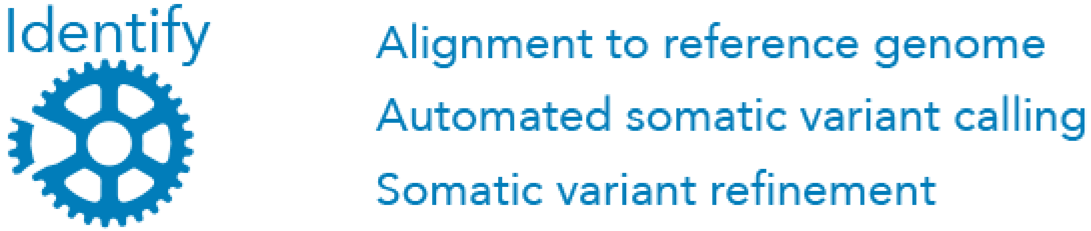
Sequencing is the final step in data production part of a genomic analysis pipeline. The most commonly used sequencing technique is so-called next-generation sequencing (NGS) or high-throughput sequencing, which evaluates millions of sequences in parallel to dramatically reduce time and cost of the analysis. There are at least two popular platforms (in use clinically) that harness the power of next-generation sequencing to efficiently sequence tumor samples:

- **Illumina sequencing** anneals individual reads to a bead or plate using DNA adaptors and the molecule is amplified through PCR. Amplified reads are sequenced by individually adding single fluorescently tagged and blocked-nucleotides to the complementary DNA sequence and exposing the nucleotide to light to produce a characteristic fluorescence. These blocked-nucleotides can then be un-blocked to allow for an additional base to bind and the process repeated until the whole complementary sequence is elucidated. This platform has a high accuracy rate and can evaluate 50-300 base-pairs per read, and very high-throughput runs producing millions to billions of reads. Each run takes approximately 2-3 days to complete for as little as $1,000 per 30x whole genome sample.
- **ThermoFisher ION Torrent** evaluates hydrogen atoms emitted during polymerization of base pairs, which can be measured as a variation in the solution's pH. This method has a low error rate for substitutions and point mutations and it is relatively inexpensive with a fast turn-around for data production (2-7 hours per run), however, the platform has higher error rates for insertions and deletions, it cannot read long chains of mononucleotides, and it cannot currently match the throughput of the Illumina sequencing platform.

5.6.2 Third generation sequencing

Third Generation Sequencing Platforms: PacBio and NanoPore are considered third generation sequencing technologies that can sequence longer reads at a reduced cost to address the existing problems associated with NGS.

- **PacBio** utilizes hairpin adaptors to create a loop of DNA that can be fed through an immobilized polymerase to add complementary base pairs. As each nucleotide is held in the detection volume by the polymerase, a light pulse identifies the base. This platform requires high quality intact DNA with highly controlled fragmentation and can read strands up to 1Mb in length.
- **Oxford NanoPore Sequencing** utilizes biological transmembrane proteins that translocalize DNA. Measurement of changes in electoral conductivity as the DNA passes through the pore elucidates sequence reads. This platform can evaluate variable length reads and is inexpensive relative to other technologies. Specifically, the MinION device is completely portable, commercially available and can evaluate 20-100MB per run. The trade-off is its low fidelity rate of only ~85%.


Identify
Alignment to reference genome
Automated somatic variant calling
Somatic variant refinement

CHAPTER 6

Identify somatic variants

The completion of the sequencing pipeline results in raw FASTA files, which are text-based files that represent nucleotide sequencings. In this section, we describe how to align sequencing reads to reference genome, how to call somatic variants using automated software, and how to refine variants using semi-automated processes.

This section was made possible by the wonderful information provided by the Precision Medicine Bioinformatic Course developed in the Griffith Lab.

6.1 Outputs from Sequencing Pipeline

6.1.1 1) FASTA Format

During library preparation and target enrichment, read strands are generated as input for sequencing platforms. These read strands are digitally read by sequencing machines and printed to a FASTA file. If your sequencing machine parameter included paired-end reads, whereby each read was read twice, then you will have two FASTA files per sample. These raw files have a consistent format that can be easily read by aligners. Each line a FASTA file includes a header and a sequence:

> \> GAPDH_204s.100.1 AATTAGGAGCGATTTGAGATTGCCCCCGATTTATTGACCCGTTTAGCC
>
> \> HAPTB_204s.100.1 AAGGCGTGAGAAAGTGCCCGTGGGTAGTGCGGGAGTGGGATGGTAGCC

Raw FASTA files will include any indexes, linkers, or unique-molecular identifiers (UMIs) that were employed in library preparation or hybridization capture. These sequences will need to be trimmed from the raw sequencing reads prior to alignment. Often this trimming process will be performed by software provided by the commercial entity associated with the instrument being used.

As FASTA files are processed, read strands in FASTA files are annotated with additional information including alignment location, quality, strand, etc.

6.1.2 2) FASTQ Files

In addition to FASTA files, sequencing runs often produce FASTQ files, which provides a high level overview of sequencing quality. FASTQ files also have a consistent format that can be easily read by aligners. Each line a FASTQ file includes a header, a sequence, a separator, and quality scores.

Read example 1:

- > GAPDH_204s.100.1
- AATTAGGAGCGATTTGAGATTGCCCCCGATTTATTGACCCGTTTAGCC
- +
- !‘‘*((_*>*+())))>>>>>***+1.(%%%%%^&****#)CCCCC65

Read example 2:

- > HAPTB_204s.100.1
- AAGGCGTGAGAAAGTGCCCGTGGGTAGTGCGGGAGTGGGATGGTAGCC
- +
- !‘‘*(((*(*+()))>>>>>.%%%%^&**#)C65***+()))>>>>>.%

Quality scores are based on the Phred scale and are enclosed using ASCII Annotation characters (for brevity). Each score is calculated differently depending on the technology/instrument used for sequencing.

6.1.3 3) Pre-Alignment QC

FASTQ files can be used to generate FastQC Reports. These reports show basic statistics about sequencing (total reads, total poor quality reads, sequence length, GC content, etc.) and provide graphs that give the user a feel for sequencing quality. An example of this type of report is shown below:

..image:: images/FastQC_Report.png

Generating pre-alignment QC can be accomplished following the commands on the PreAlignment QC page provided by the Precision Medicine Bioinformatic Course.

6.2 Alignment Strategies

6.2.1 The Reference Genome

The reference genome approximates the complete representation of the human genetic sequence for the 4 billion base pairs in human DNA. Using a representative assembly prevents the need to build an assembly each time a genome is sequenced, however, there are intrinsic flaws to this approach. Specifically, due to single nucleotide polymorphisms (SNPs) intrinsic to an individual, the reference genome does not perfectly match any one individual. Further, repetitive elements (duplications, inverted repeats, tandem repeats), the reference is often incomplete or incorrect. Therefore, new genome assemblies are constantly being built to improve our ability to resolve the true human genome sequence. Most recently, GRCh37 was published in 2009 and GRCh38 was published in 2013. A summary of genome releases has been provided by UCSC.

Currently, the CIViC database supports variants from NCBI36 (hg18), GRCh37 (hg19), and GTCh38 (hg20), however most variants are associated with reference build GRCh37 (hg19). Therefore, we recommend that for the pipeline, the alignment strategy should use GRCh37 (hg19).

6.2.2 Alignment Algorithms

Alignment can be performed using various alignment software. Generally speaking, alignment strategies can either optimize accuracy or processing time.

- Optimal solutions include either Spith-Waterman or Needleman-Wunsch alignment strategies. These algorithms are computationally expensive and process read strands slowly.
- Fast solutions include hash-based solutions such as Burrows-Wheeler transformation. These algorithms create shortcuts to reduce alignment time.

The input for alignment software is the FASTA files and the output from alignment is a Sequence Alignment Map (SAM) or Binary Alignment Map (BAM). Typically, these software also produce alignment QC, which includes information about mapped reads, coverage, etc.

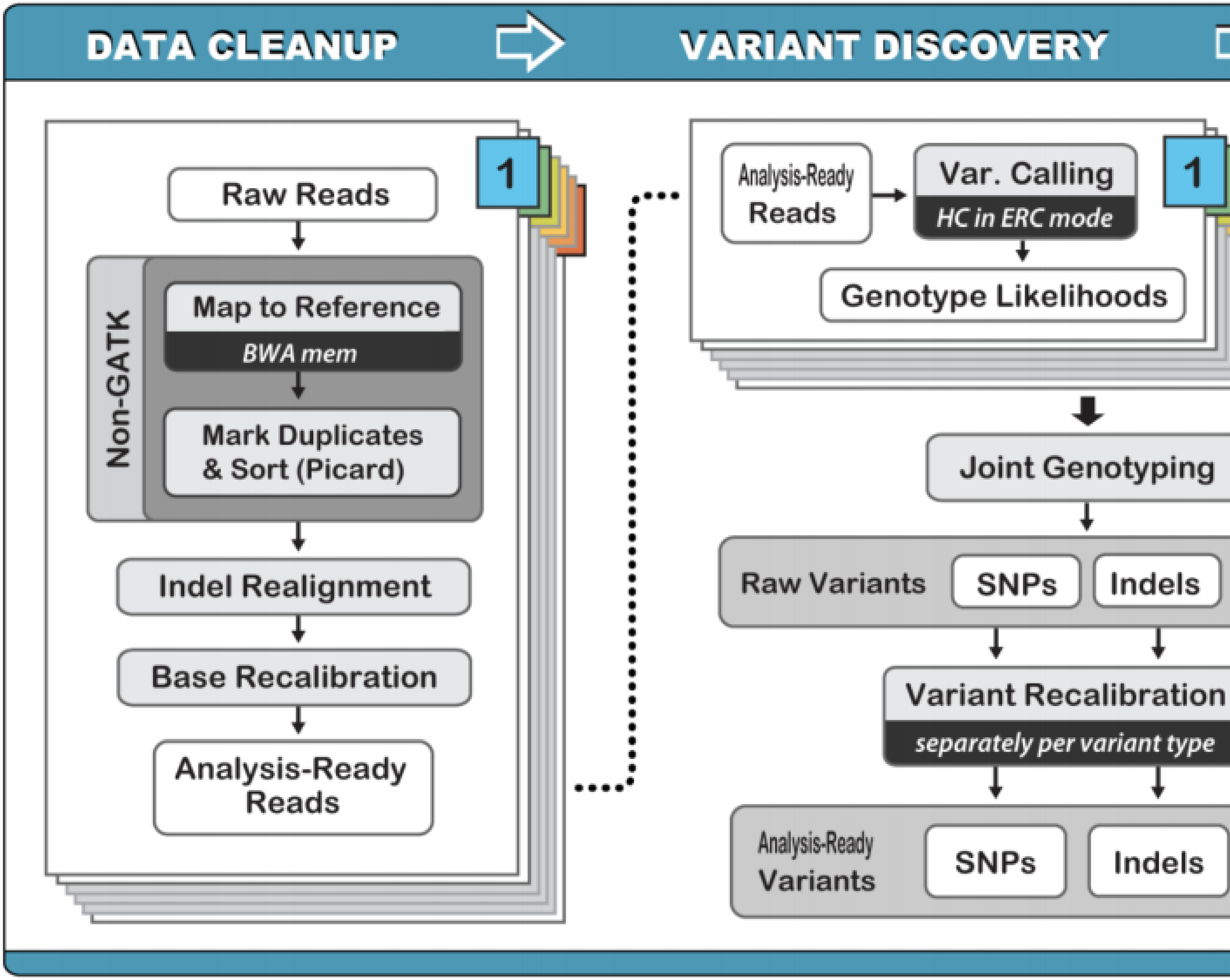
Sequencing alignment can be accomplished following the commands on the Alignment page provided by the Precision Medicine Bioinformatic Course.

6.3 Germline Variant Analysis

The next step in the sequencing pipeline is to use paired tumor and normal alignments for germline and somatic variant calling. Germline variant calling consists of identifying single nucleotide polymorphisms (SNPs), insertions/deletions (indels), and structural variants (SVs) that are intrinsic to the normal tissue. Somatic variant calling is a similar process, but it requires the variant to be exclusively observed in the tumor tissue and not present in the germline (normal) tissue. Below we describe automated methods for bother germline and somatic variant calling.

6.3.1 Germline Variant Analysis

Germline variant calling can be performed using a variety of software. Typically, our lab uses GATK (genome analysis tool kit) for initial germline calling and variant filtering.



The optimal method for germline variant calling using GATK Haplotype Caller, which considers all SNPs, Indels, and SVs together by creating a local de novo assembly. Although this method is computationally intensive, it improves overall variant calling accuracy by eliminating many false positives.

Germline variant calling can be accomplished following the commands on the Germline SNV and Indel Calling page provided by the Precision Medicine Bioinformatic Course.

6.3.2 Germline Variant Refinement

Germline variant refinement can be performed by using heuristic cutoffs for quality metrics or by employing Variant Quality Score Recalibration (VQSR). Hard filtering uses (somewhat arbitrary) cutoffs for quality scores that are provided by the GATK workflow. For example, you can require a minimum QualByDepth (QD) of 2.0. GATK provides strategies for hard filtering in their Hard Filtering Tutorial. VQSR filtering is more sophisticated than hard-filtering. This model estimates the probability that a variant is real and allows filtering at various confidence levels. GATK provides methods for recalibrating variant quality scores in their VQSR Tutorial.

Germline variant filtering can be accomplished following the commands on the Germline Filtering, Annotation, and Review module provided by the Precision Medicine Bioinformatic Course.

6.4 Somatic Variant Analysis

In addition to germline variant calling, somatic variant calling can be performed identifying differences that are intrinsic to the tumor sample and not observed in the matched normal samples. Somatic variant calling requires looking for single nucleotide variants (SNVs), insertions and deletions (indels), copy number variants (CNVs), structural variants (SVs), and loss of heterozygosity (LOH). These different types of variants can be identified by using various software. Here we will go through each of these automated variant callers to describe the types of variants each caller can identify and subsequent strengths and weaknesses.

6.4.1 Somatic SNV/InDel Calling

Algorithms that perform Somatic SNV/Indel Calling include: VarScan, Strelka, and MuTect. It is recommended that aligned bam files are evaluated by multiple different variant callers and subsequently, filtering can be employed by identifying variants that were observed by multiple different callers.

1) Varscan is a platform-independent mutation caller for targeted, exome, and whole-genome resequencing data and employs a robust heuristic/statistic approach to call variants that meet desired thresholds for read depth, base quality, variant allele frequency, and statistical significance.

2) Strelka calls germline and somatic small variants from mapped sequencing reads and is optimized for rapid clinical analysis of germline variation in small cohorts and somatic variation in tumor/normal sample pairs. Both germline and somatic callers include a final empirical variant rescoring step using a random forest model to reflect numerous features indicative of call reliability which may not be represented in the core variant calling probability model.

3) MuTect2 is a somatic SNP and indel caller that combines the DREAM challenge-winning somatic genotyping engine of the original MuTect (Cibulskis et al., 2013) with the assembly-based machinery of HaplotypeCaller.

Somatic Variant Calling with these three automated callers can be accomplished following the commands on the Somatic SNV/InDel Calling module provided by the Precision Medicine Bioinformatic Course.

6.4.2 Somatic SV Calling

Manta is a structural variant caller maintained by Illumina and optimized for calling somatic variation in tumor/normal pairs. Structural variants are rearrangements in DNA involving a breakpoint(s). Generally speaking structural variants can fall into four categories:

1) Insertions: a region is inserted into the DNA

2) Deletions: a region is deleted in the DNA

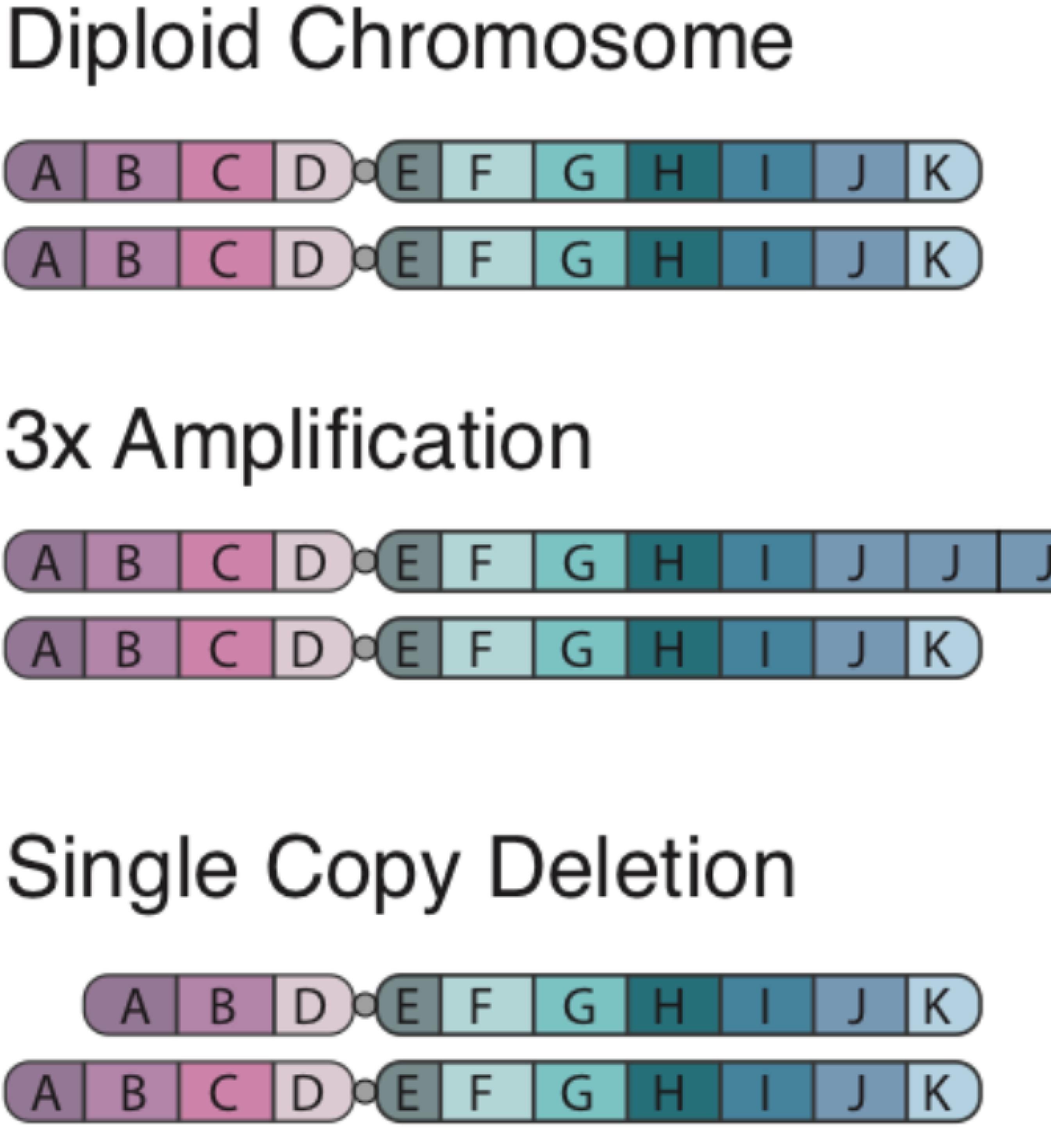
3) Inversions: a section of DNA is reversed

4) Translocations: a section of DNA is remved and re-inserted in a new region

Somatic Structural Variant calling with Manta can be executed by following the commands on the Somatic SV Calling module provided by the Precision Medicine Bioinformatic Course.

6.4.3 Somatic CNV Calling

Copy number alterations occur when a section of the genome is duplicated or deleted. This phenomenon has an important role in evolution for development of homologs/paralogs that allow for development of new function with retention of old function (e.g., alpha and beta hemoglobin). However, these events can also play an intrinsic role in disease development. Examples of Copy Number Alterations are shown below:



There are two algorithms that can be used to identify copy number alterations in tumor samples:

1) copyCat is an R package used for detecting somatic (experiment/control) copy number aberrations. It works by measuring the depth of coverage from a sequencing experiment. For example in a diploid organism such as

human, a single copy number deletion should result in aproximately half the depth (number of reads) compared to the control.

2) CNVkit is a python package for copy number calling specifically designed for hybrid capture and exome sequencing data. During a typical hybrid capture sequencing experiment the probes capture DNA from the sequencing library, however the probes don't always bind perfectly. This results in not only the "on-target" regions being pulled from the library for later sequencing but "off-target" as well where the probes didn't perfectly bind and essentially pulled the wrong region. The effect provides very low read coverage across the entire genome which CNVkit takes advantage of to make CN calls.

Copy Number Variant calling with copyCat and CNVkit can be executed by following the commands on the CNV Calling module provided by the Precision Medicine Bioinformatic Course.

6.4.4 Somatic LOH Calling

Loss of heterozygosity is a common genetic event that occurs in cancer whereby one allele is lost. In this segment, the tumor sample appears to be homozygous whereas the same section is heterozygous in the matched normal sample. Methods for calculating sections of LOH requires first calculating the variant allele frequencies (VAFs) in the normal sample to find heterozygous germline positions. Subsequently, we run bam-readcounts on the tumor sample at these same genomic loci and determine if the tumor sample shows homo- or heterozygosity. If there is an area of homozygosity in the tumor sample that is heterozygous in the normal sample, this represents a section of LOH.

Methods for identifying and visualizing section of LOH can be found in the Somatic LOH Calling module provided by the Precision Medicine Bioinformatic Course.

6.4.5 Somatic Variant Refinement

Following automated somatic variant calling, somatic variant refinement is required to identify a high-quality list of variants associated with an individual's tumor. Especially for SNVs/Indels, we recommend that the final list of variants identified by automated callers is further refined. This can be accomplished by executing one or all of the following:

1) Heuristic filtering based on variant allele frequency (VAF), total coverage, allele read count, or and allele read depth. We recommend that variants require at least 20X coverage in both the tumor and normal sample with a VAF >5%. These numbers can be adjusted based on the experiment and the reagents employed on the samples.

2) Manual review of aligned sequencing reads. The Griffith Lab has defined a Standard Operating Procedure (SOP) for somatic variant refinement of sequencing data with paired tumor and normal sample. The SOP describes a standard way to visualize reads using Integrative Genomic Viewer (IGV) and assign labels to variants to filter in true somatic variants and filter out variants attributable to sequencing and alignment artifacts. Although manual review of aligned sequencing reads is incredible effective in eliminating many false positives observed during automated calling, it is incredibly time-consuming and expensive.

3) Automated refinement using a machine learning approach. DeepSVR is a deep learning model that evaluates variants called by automated callers and provides a value between 0-1 that indicates confidence in that variant being a true positive. The input in the model is 59 features derived from aligned BAM files and the output is a value for three labels: Somatic, Ambiguous, Fail. This model can be employed on samples to further refine a list of somatic variants from automated callers.

Additionally, somatic variants can be further confirmed by comparing identified variants to variant databases such as:

- gnomAD: (123,136 WXS and 15,496 WGS)
- 1000 genome: 1000 genome projects
- Exome Aggregation Consortium: (~60,000 individuals)
- Exome sequencing project: (~6,500 individuals)

Methods for Somatic Variant Refinement can be viewed on the Somatic SNV and Indel Manual Review module provided by the Precision Medicine Bioinformatic Course.

An

CHAPTER 7

Annotate variants

After identifying a putative list of somatic variants associated with the patient's tumor, this section describes how to link variants back to the CIViC database to annotate the sample/variants for clinical relevance. We again use an interactive interface (jupyter notebook) to import somatic variants calls and output a report that can be easily consumed by the user.

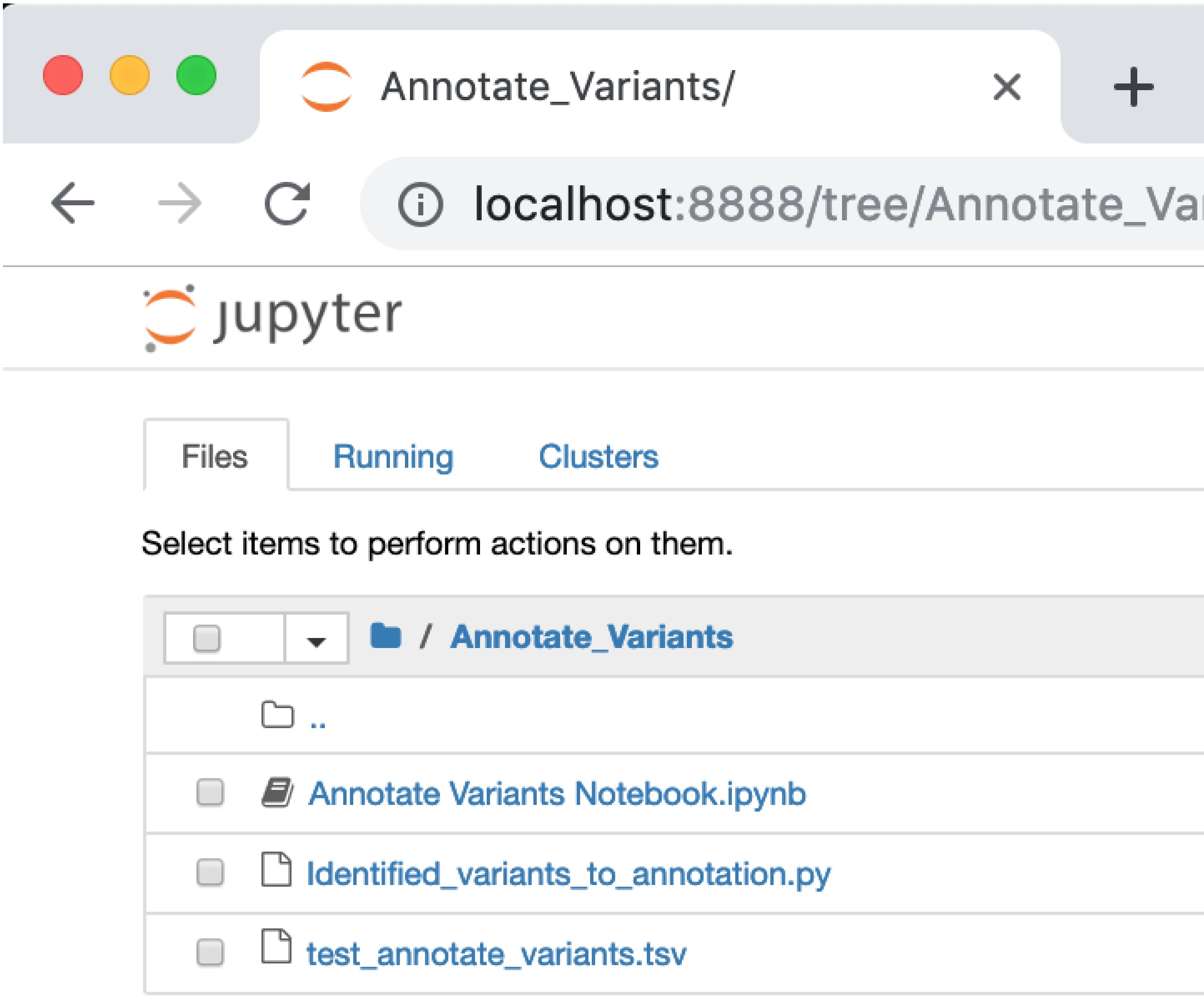
7.1 Build Binder Docker Image

We have built a Binder Jupyter Notebook that contains code to pull in a BED/BED-like file and link variants to clinical relevance annotations for all variants that have CIViC curation. Please open the link provided below to start this process (Note: loading the Jupyter Notebook can take 5-10 minutes):

Build Jupyter Notebook

7.2 Docker Image Set-up

Once the Jupiter Notebook is loaded, the interface should look as follows:



Select items to perform actions on them.

□ / Annotate_Variants

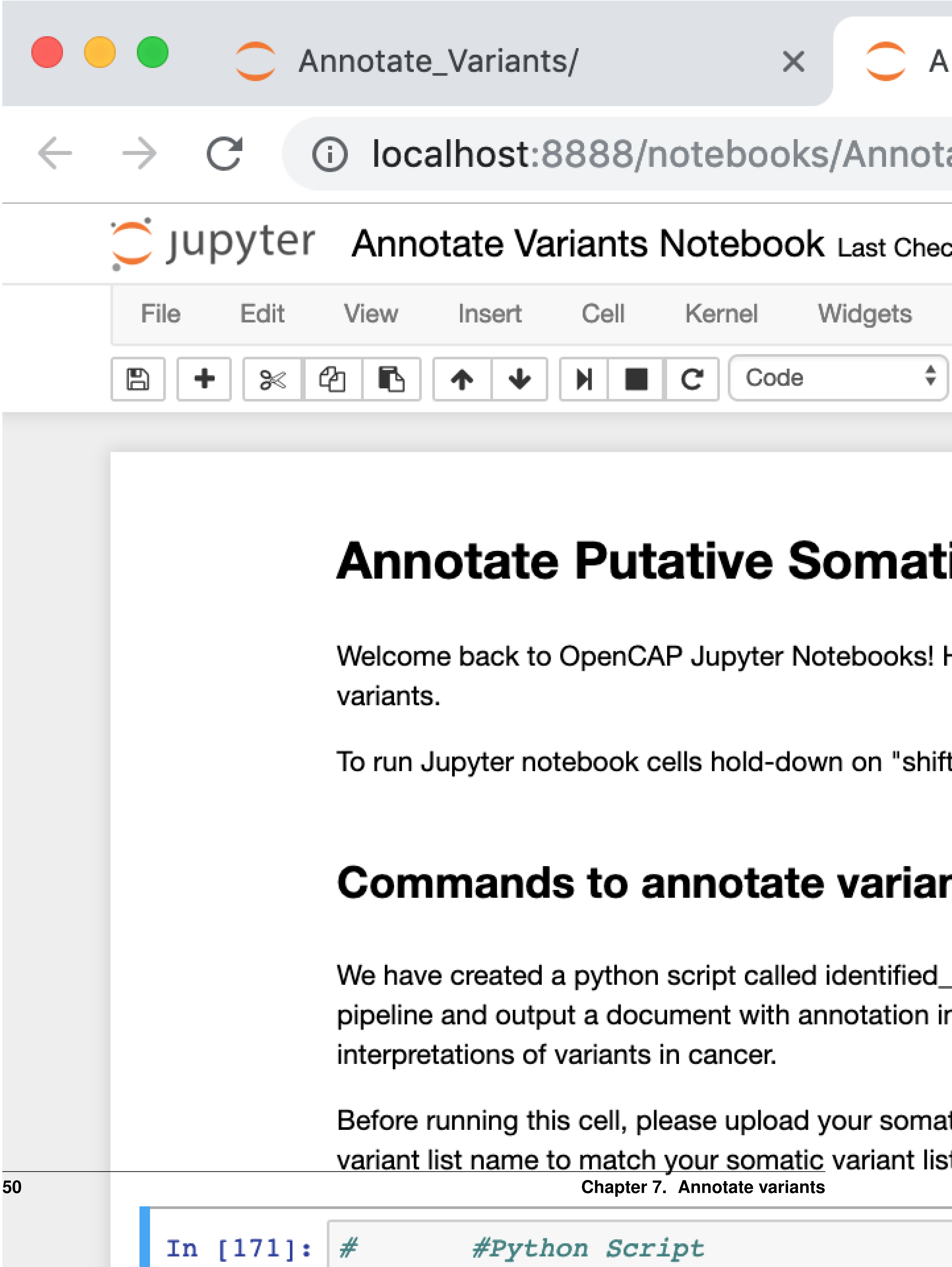
..

□ Annotate Variants Notebook.ipynb

□ Identified_variants_to_annotation.py

□ test_annotate_variants.tsv

Annotate Variants Notebook.ipynb = This notebook is an interactive session that allows users to run python scripts. The specific notebook in this section is set up to run Identified_variants_to_annotation.py. To use this notebook, you must change the input variables (input variant list)



Annotate Putative Somati

Welcome back to OpenCAP Jupyter Notebooks! H
variants.

To run Jupyter notebook cells hold-down on "shift

Commands to annotate varian

We have created a python script called identified_
pipeline and output a document with annotation in
interpretations of variants in cancer.

Before running this cell, please upload your somat
variant list name to match your somatic variant list

```
In [171]: #        #Python Script
```

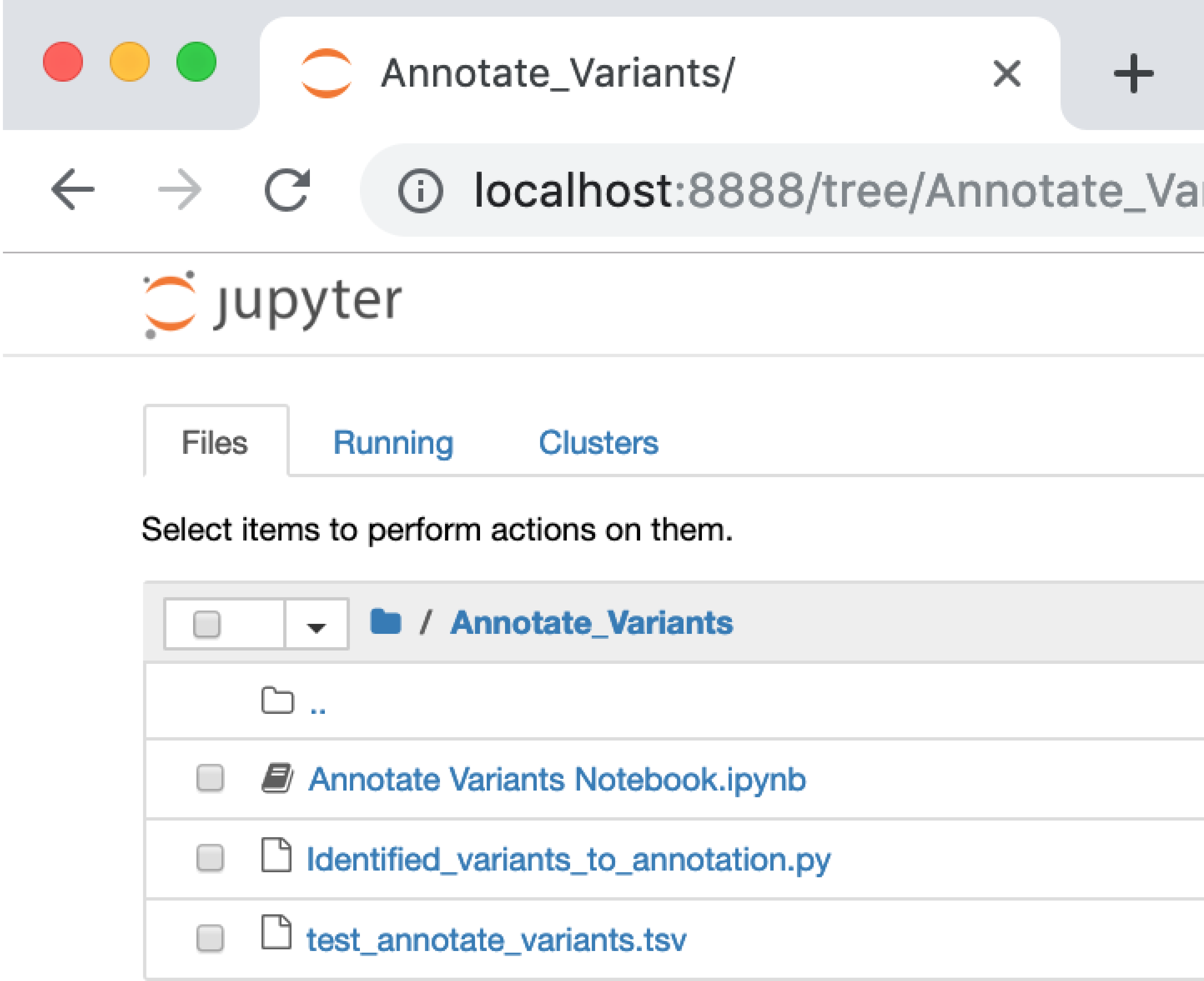
Identified_variants_to_annotation.py = Python script that takes in the somatic variant list (see test_annotate_variants.tsv) and the sample name. The script will iterate through each variant and ascertain if the variant is in CIViC. If a somatic variant is in CIViC, the script will pull all information about the variant (variant descriptions, assertions, and evidence items) and create an OpenCAP output report. After running the script, the output report will be created in the same directory as these files.

test_annotate_variants.tsv = BED-like tab-separated file with variant coordinates that require OpenCAP annotation. The file contains five columns: Chromosome, Start, Stop, Ref, Var. Each row provides genomic coordinates to a single somatic variant that was observed during sequencing. Each variant will be evaluated for presence in the CIViC database using OpenCAP.

7.3 Run OpenCAP in Binder

Using the Binder Docker Image created above, you can run the provided identified_variants_to_annotation.py script. This script will take in variants from the sequencing pipeline and output a document with annotation information.

Before running the Jupyter Notebook code blocks, you must upload your somatic variants to the home directory. This can be accomplished using the "upload" button on the home directory:



Select items to perform actions on them.

/ Annotate_Variants

..

□ Annotate Variants Notebook.ipynb

□ Identified_variants_to_annotation.py

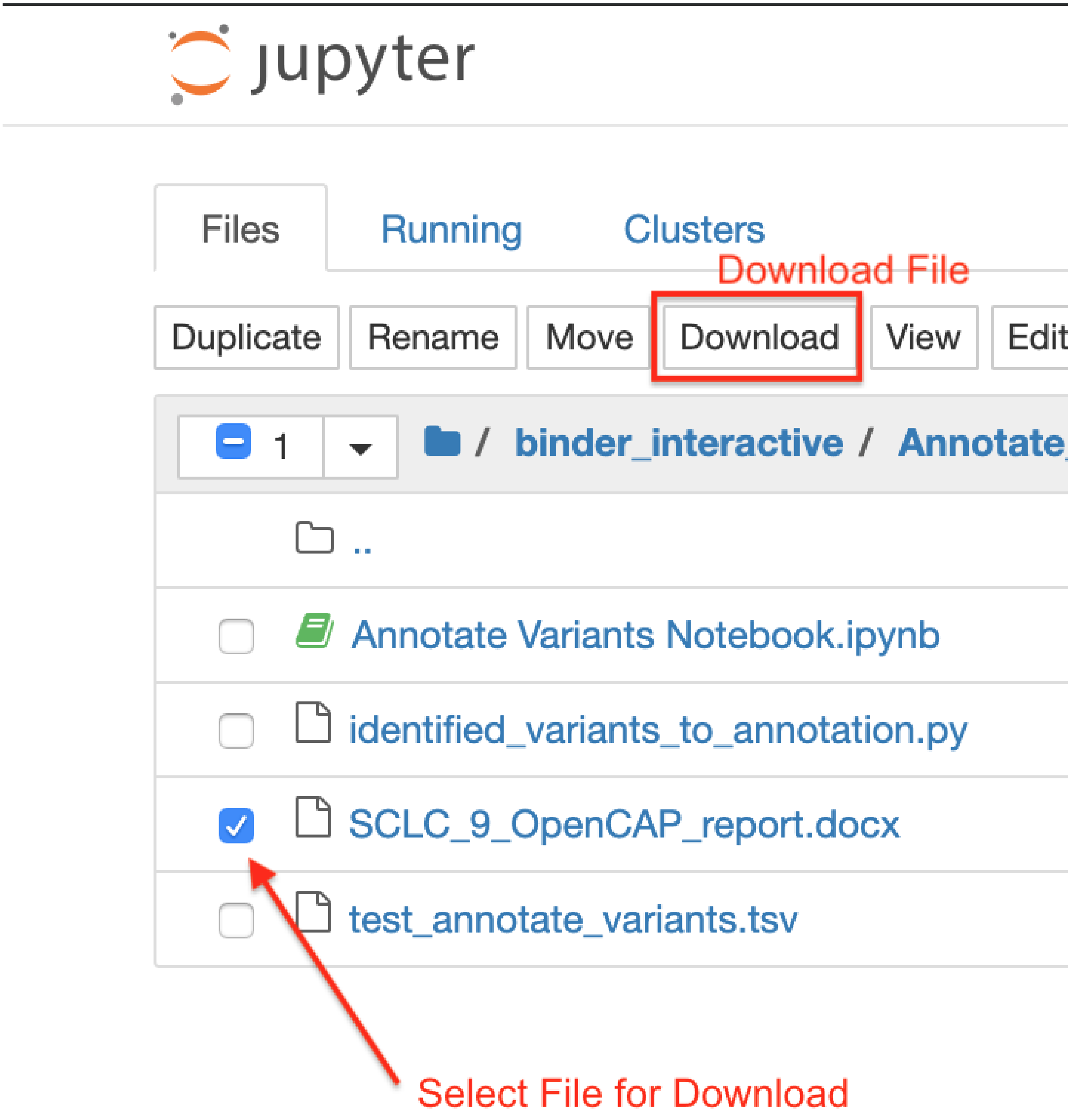
□ test_annotate_variants.tsv

You must also change the sample name to match the sample name of the associated somatic variants. This name will be used to label the output files.

Once you have configured the input variables, you can run the command by holding the "shift" key and selecting "enter".

7.4 OpenCAP Output File

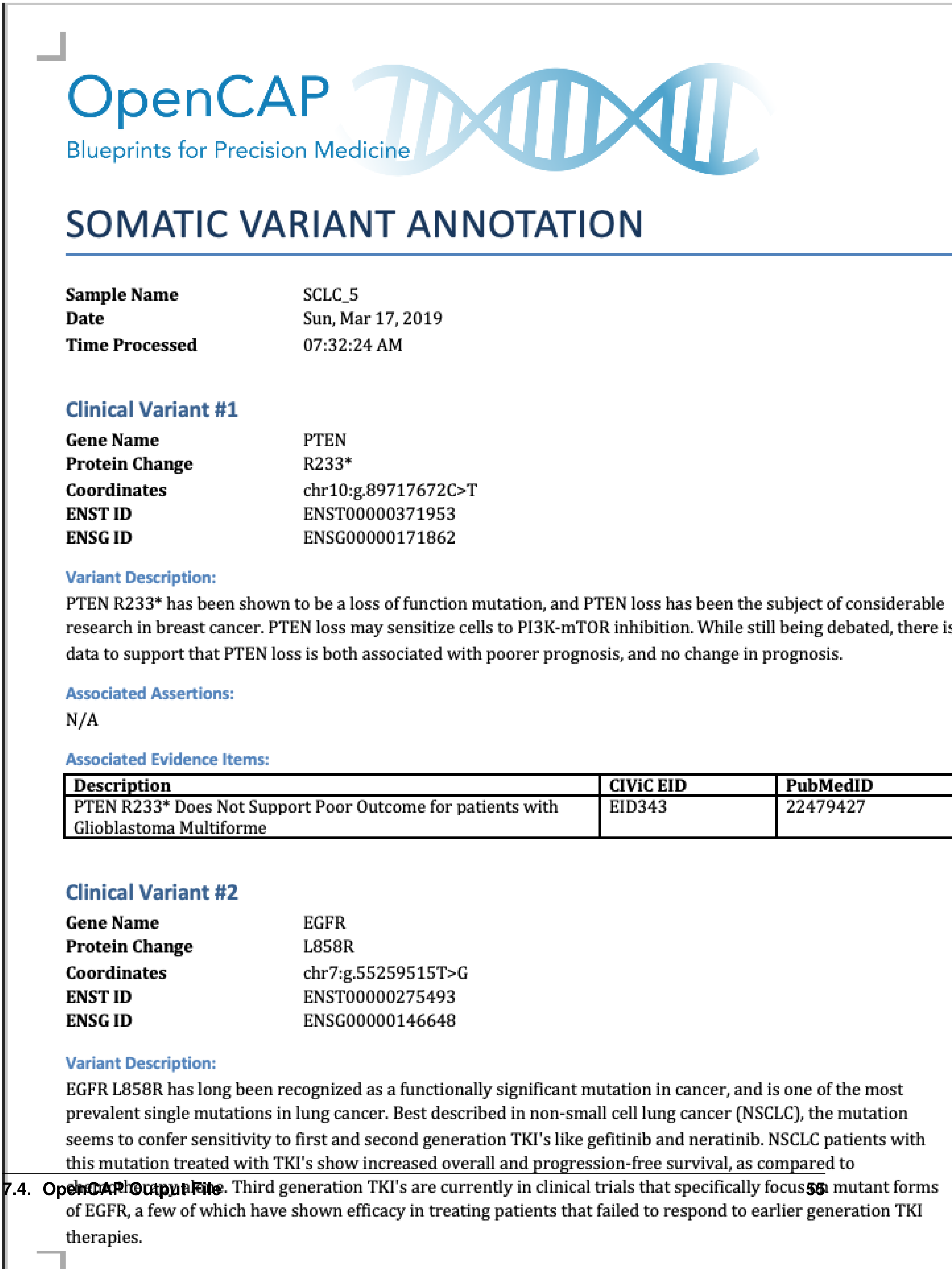
After you run Identified_variants_to_annotation.py using the Jupyter Notebook, a file will be generated in the home directory. This file name will start with the sample name and will end with *"OpenCAP_report.docx"*. Select this file to download the OpenCAP report for your sample. The following screenshot shows you how to download these reports:


jupyter
Files
Running
Clusters
Duplicate
Rename
Move
Download
View
Edit
1
/ binder_interactive / Annotate
..
Annotate Variants Notebook.ipynb
identified_variants_to_annotation.py
SCLC_9_OpenCAP_report.docx
test_annotate_variants.tsv

Download File

Select File for Download

The report will look something like the following:



OpenCAP

Blueprints for Precision Medicine

SOMATIC VARIANT ANNOTATION

Sample Name	SCLC_5
Date	Sun, Mar 17, 2019
Time Processed	07:32:24 AM

Clinical Variant #1

Gene Name	PTEN
Protein Change	R233*
Coordinates	chr10:g.89717672C>T
ENST ID	ENST00000371953
ENSG ID	ENSG00000171862

Variant Description:

PTEN R233* has been shown to be a loss of function mutation, and PTEN loss has been the subject of considerable research in breast cancer. PTEN loss may sensitize cells to PI3K-mTOR inhibition. While still being debated, there is data to support that PTEN loss is both associated with poorer prognosis, and no change in prognosis.

Associated Assertions:

N/A

Associated Evidence Items:

Description	CIViC EID	PubMedID
PTEN R233* Does Not Support Poor Outcome for patients with Glioblastoma Multiforme	EID343	22479427

Clinical Variant #2

Gene Name	EGFR
Protein Change	L858R
Coordinates	chr7:g.55259515T>G
ENST ID	ENST00000275493
ENSG ID	ENSG00000146648

Variant Description:

EGFR L858R has long been recognized as a functionally significant mutation in cancer, and is one of the most prevalent single mutations in lung cancer. Best described in non-small cell lung cancer (NSCLC), the mutation seems to confer sensitivity to first and second generation TKI's like gefitinib and neratinib. NSCLC patients with this mutation treated with TKI's show increased overall and progression-free survival, as compared to chemotherapy alone. Third generation TKI's are currently in clinical trials that specifically focus on mutant forms of EGFR, a few of which have shown efficacy in treating patients that failed to respond to earlier generation TKI therapies.

For a variant to be included in OpenCAP it must be a perfect match (i.e., chromosome, start, stop, reference, variant). Currently, OpenCAP does not support matching bucket variants (e.g., TP53 - MUTATION) or variants without specific genomic changes (e.g., KRAS - G12*). We hope to improve the pipeline over time to allow for annotation of these variants.

For somatic variants that have a perfect match with a CIViC entry, a "Clinical Variant" section has been created for this variant. For each entry, the annotation has four distinct parts:

1) **Variant Information:** For each variant, we list the following:

- Gene name - HUGO Nomenclature
- Protein coding change - HGVS Nomenclature for variant protein change
- Genomic coordinates - HGVS Nomenclature for variant genomic coordinates
- ENST ID - Ensembl transcript identification number for representative transcript
- ENSG ID - Ensembl gene identification number for gene

2) **Variant Description:** If the variant has a description in CIViC, the variant description has been reproduced in this section. The variant description contains a high-level overview of all evidence statements available for this variant.

3) **Associated Assertions:** If the variant has associated assertions, then these assertions have been reproduced in this section. Assertions incorporate multiple evidence items to support a single clinical relevance statement. Typically, assertions include information from nationally recognized organizations such as the NCCN, the FDA, and the AMP.

4) **Associated Evidence Items:** This section provides an overview of evidence items associated with the variant. Evidence items have been condensed into a grid with three columns. The first column is a single description of the evidence item, the second column provides the evidence item identification number(s) that support(s) the description, and finally, we provide the Pubmed identifications numbers associated with these evidence statements. For an evidence statement to be included in the grid, it must have an evidence level greater than "C" (Case Study) and it must be "Accepted".

CHAPTER 8

Indices and tables

- genindex
- modindex
- search