Automating the Collection and Processing of Cancer Mutation Data

Master's Project Report
Jeremy Watson
Advisor: Ben Raphael

1. Introduction

The Raphael lab has developed multiple software packages, such as CoMEt[1] (Combinations of Mutually Exclusive Alterations), HotNet2[2], and the visualization and collaboration tool MAGI[3] (Mutation Annotation and Genome Interpretation) that utilize mutation data from The Cancer Genome Atlas (TCGA). While the data is invaluable, the process of collecting it and processing for use is tedious and complex. TCGA is frequently updated, with cancers and samples being added and updated monthly. Additionally, the lab's varying software packages have different requirements for the form and kind of information provided by the data. Before now, this processing was done by scripts written on an ad-hoc basis, which worked, but as with any ad-hoc solution they were not meant to be maintainable or extensible.

My goal for this project was first to rewrite the processing code for the single nucleotide variations (SNV) and copy number aberrations/alterations (CNA) mutation data, and then provide an easy to use tool that would automate the entire process. This covers everything from downloading the raw mutation data to exporting the finalized inputs for the lab software, with an emphasis on simplifying the process of adding new mutation data to MAGI. The tool provides various options for processing, summary statistics, and visualizations. I will use this document to outline the problem, an overview of my implementation, as well as how my solution relates to other lab software packages.

---

[1] Mark D.M Leiserson*, Hsin-Ta Wu*, Fabio Vandin, Benjamin J. Raphael. (2015) Comet: A Statistical Approach to Identify Combinations of Mutually Exclusive Alterations in Cancer. Biology, 16:160.
[2] M.D.M. Leiserson*, F. Vandin*, H-T. Wu, J.R Dobson, J.V. Eldridge, J.L. Thomas, A. Papoutsaki, Y. Kim, B. Niu, M. McLellan, M.S. Lawrence, A. Gonzalez-Perez, D. Tamborero, Y. Cheng, G.A. Ryslik, N. Lopez-Bigas, G. Getz, L. Ding, and B.J. Raphael. Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. Nature Genetics (2014).
[3] Leiserson, et al. Nature Methods, 2015.

2. **Overview and Background**

At a high level there are two types of mutations, which for the purpose of the project is important in that they define two different kinds of input data: Single nucleotide variation (SNV) mutations, and copy number aberration (CNA) mutations. Note that the SNV/CNA names are not universally used, for the purposes of this report that is how I will be referring to them.

SNV Mutations

The specification for my project was to process SNV mutations using the Mutation Annotation Format[4] (MAF), a commonly used format specified by the TCGA. It is a simple columnar tab separated data format, listing out such information as the gene at the location of mutation, type of mutation, and so on.

One of the challenges of this project was that the data comes from many different sources (even though it was pulled from a single repository), and despite a specification existing the data was often incomplete, mislabeled, or otherwise deviated from the specification. Thus, the processing had to be flexible and sensitive to account for a lack or misidentification of data.

CNA Mutations

The standard for CNA mutation data for my project was data produced by GISTIC2.0[5] (Genomic Identification of Significant Targets in Cancer), a software package developed at the Broad Institute. The Broad describes it as “a tool to identify genes targeted by somatic copy-number alterations (SCNAs) that drive cancer growth. By separating SCNA profiles into underlying arm-level and focal alterations, GISTIC estimates the background rates for each category as well as defines the boundaries of SCNA regions.” For my project, I needed to extract the genes contained in amplified or deleted regions, as well as their associated amplification/deletion measures and locations.

---

[4] https://wiki.nci.nih.gov/display/TCGA/Mutation+Annotation+Format+(MAF)+Specification

[5] Mermel, et al. GISTIC2.0 facilitates sensitive and confident localization of the targets of focal somatic copy-number alteration in human cancers. Genome Biol. 2011;12(4):R41. Epub 2011 Apr 28.

Tools and Source Data

The project was written entirely in Python with some light shell scripting, because of my (and other lab members') familiarity, the project's performance requirements, and ease of use for rapid prototyping and development.

The data source is the Broad Genomic Data Analysis Center[6] (commonly GDAC), which provides a variety of raw and processed data, including the MAF and GISTIC2.0 files. I will discuss the Broad GDAC, and how the data is pulled, more in a later section.

3. **Processing SNV Data**

Here I will go into more detail about the input and outputs of the SNV processing code for use by HotNet2, MAGI, and CoMEt, as well as discuss usage.

Inputs

As mentioned previously, the data format for SNV processing is the MAF, a simple tab-separated columnar text file format. The specification lists 34 possible data entries, but only a few are required in order to generate useful output for our purposes. Note that MAGI requires more information than either CoMEt or HotNet2. I will outline each data item and briefly describe it.

MAF data always required:

- HUGO Symbol [7]- The unique symbol for the gene identified as containing the mutation. HUGO (Human Genome Organization) is the body that sets the naming standards for the human genome.
- Sample - The unique sequence identifying the patient and sample. Comes from the Biospecimen Core Resources Center[8] at TCGA.
- Variant Classification - Translational effect of variant allele
- Validation Status - Must be "Valid"
- Mutation Status - Status of mutation (somatic, germline)
- Variant Type - Type of mutation, for example TNP (tri-nucleotide polymorphism) or ONP (oligo-nucleotide polymorphism). Helps determine how to parse amino acid change

[6] https://gdac.broadinstitute.org/
[7] http://www.genenames.org/
[8] https://wiki.nci.nih.gov/display/TCGA/Biospecimen+Core+Resource

MAF data required to output to MAGI:

- Codon - Used with amino acid change to identify original and new amino acid based on mutation
- Amino Acid Change - Used to help identify amino acid change based on mutation
- Transcript ID - Used to check against the transcript database to find the length of transcript

As noted above, more data is required in order to output for MAGI than either HotNet2 or CoMEt. If asked, the code will attempt to output for MAGI, but failing that it will still try and output for CoMEt and HotNet2 where possible. MAGI requires another file, a transcript dictionary that maps transcripts to their lengths

The trickiest operation here, and the most common failure point, is parsing the amino acid changes. As noted before, one of the biggest hurdles is the nonconformity of data to the specification. This is compounded as there is not a single, consistent standard for expressing amino acid or codon changes. The code primarily uses regexes to parse the data, and while they are fairly robust (found through trial and error on large amounts of data) there remains the possibility of missing mutations. The processor's behavior is to be fairly conservative and throw out data it can't parse, as I decided it would be preferable to miss data rather than include bad data.

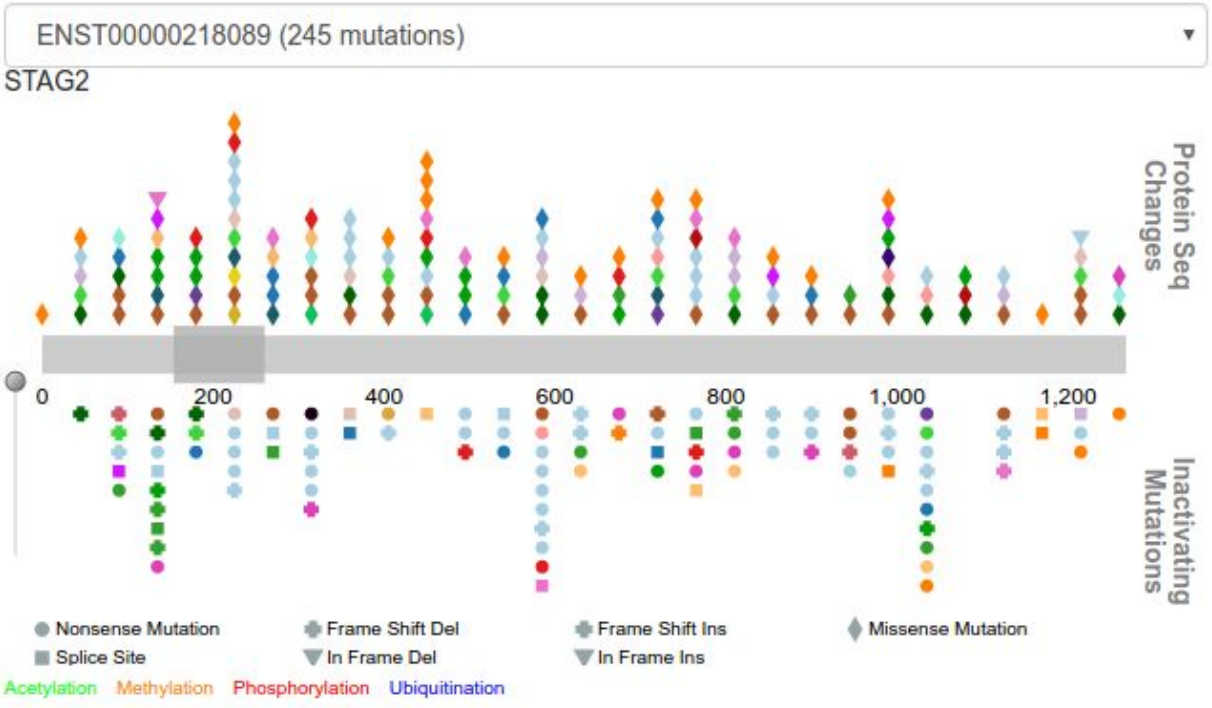
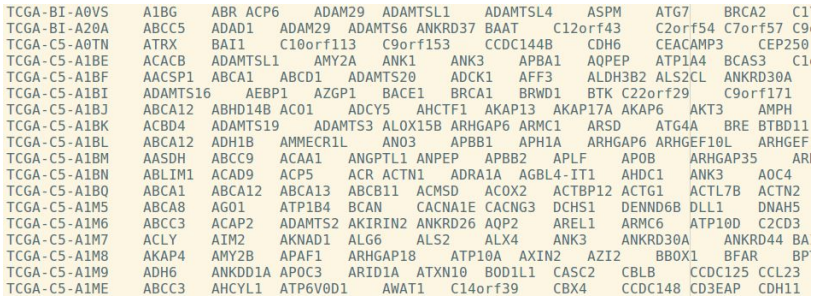


Fig 1. Visualisation for SNP data in MAGI

Outputs

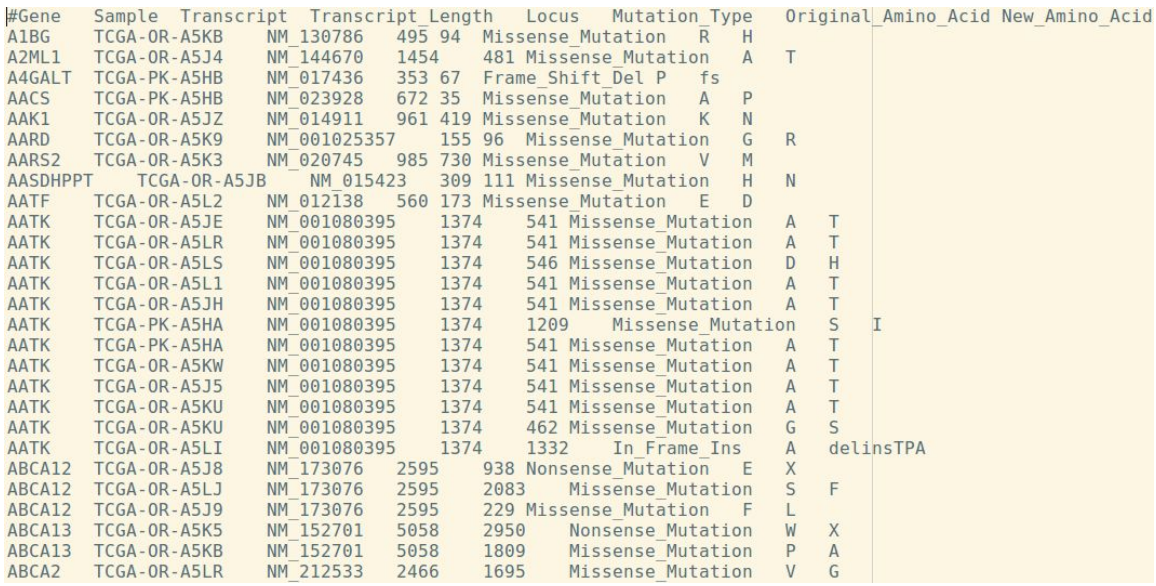
The primary outputs are the up to three files for MAGI, CoMEt, and HotNet2 respectively. For both CoMEt and HotNet2 the outputs are the same, a text file where each line begins with a sample and then a tab separated list of the genes that were found to be in that sample.



```
TCGA-BI-A0VS    A1BG    ABR ACP6    ADAM29  ADAMTSL1    ADAMTSL4    ASPM    ATG7    BRCA2   C1
TCGA-BI-A20A    ABCC5   ADAD1   ADAM29  ADAMTS6 ANKRD37 BAAT    C12orf43    C2orf54 C7orf57 C9
TCGA-C5-A0TN    ATRX    BAI1    C10orf113   C9orf153    CCDC144B    CDH6    CEACAMP3    CEP250
TCGA-C5-A1BE    ACACB   ADAMTSL1    AMY2A   ANK1    ANK3    APBA1   AQPEP   ATP1A4  BCAS3   C1
TCGA-C5-A1BF    AACSP1  ABCA1   ABCD1   ADAMTS20    ADCK1   AFF3    ALDH3B2 ALS2CL  ANKRD30A
TCGA-C5-A1BI    ADAMTS16    AEBP1   AZGP1   BACE1   BRCA1   BRWD1   BTK C22orf29    C9orf171
TCGA-C5-A1BJ    ABCA12  ABHD14B ACO1    ADCY5   AHCTF1  AKAP13  AKAP17A AKAP6   AKT3    AMPH
TCGA-C5-A1BK    ACBD4   ADAMTS19    ADAMTS3 ALOX15B ARHGAP6 ARMC1   ARSD    ATG4A   BRE BTBD11
TCGA-C5-A1BL    ABCA12  ADH1B   AMMECR1L    ANO3    APBB1   APH1A   ARHGAP6 ARHGEF10L   ARHGEF
TCGA-C5-A1BM    AASDH   ABCC9   ACAA1   ANGPTL1 ANPEP   APBB2   APLF    APOB    ARHGAP35    AR
TCGA-C5-A1BN    ABLIM1  ACAD9   ACP5    ACR ACTN1   ADRA1A  AGBL4-IT1   AHDC1   ANK3    AOC4
TCGA-C5-A1BQ    ABCA1   ABCA12  ABCA13  ABCB11  ACMSD   ACOX2   ACTBP12 ACTG1   ACTL7B  ACTN2
TCGA-C5-A1M5    ABCA8   AGO1    ATP1B4  BCAN    CACNA1E CACNG3  DCHS1   DENND6B DLL1    DNAH5
TCGA-C5-A1M6    ABCC3   ACAP2   ADAMTS2 AKIRIN2 ANKRD26 AQP2    AREL1   ARMC6   ATP10D  C2CD3
TCGA-C5-A1M7    ACLY    AIM2    AKNAD1  ALG6    ALS2    ALX4    ANK3    ANKRD30A    ANKRD44 BA
TCGA-C5-A1M8    AKAP4   AMY2B   APAF1   ARHGAP18    ATP10A  AXIN2   AZI2    BBOX1   BFAR    BP
TCGA-C5-A1M9    ADH6    ANKDD1A APOC3   ARID1A  ATXN10  BOD1L1  CASC2   CBLB    CCDC125 CCL23
TCGA-C5-A1ME    ABCC3   AHCYL1  ATP6V0D1    AWAT1   C14orf39    CBX4    CCDC148 CD3EAP  CDH11
```

Fig 2. Example output for CoMEt and HotNet2

For MAGI, the process outputs a slightly more complex file. Each line contains eight pieces of information: Gene, Sample, Transcript, Transcript Length, Locus, Mutation Type, Original Amino Acid, and New Amino Acid. This allows the visualization shown in figure 1 above.



```
#Gene   Sample  Transcript  Transcript_Length   Locus   Mutation_Type   Original_Amino_Acid New_Amino_Acid
A1BG    TCGA-OR-A5KB    NM_130786   495 94  Missense_Mutation   R   H
A2ML1   TCGA-OR-A5J4    NM_144670   1454    481 Missense_Mutation   A   T
A4GALT  TCGA-PK-A5HB    NM_017436   353 67  Frame_Shift_Del P   fs
AACS    TCGA-PK-A5HB    NM_023928   672 35  Missense_Mutation   A   P
AAK1    TCGA-OR-A5JZ    NM_014911   961 419 Missense_Mutation   K   N
AARD    TCGA-OR-A5K9    NM_001025357    155 96  Missense_Mutation   G   R
AARS2   TCGA-OR-A5K3    NM_020745   985 730 Missense_Mutation   V   M
AASDHPPT    TCGA-OR-A5JB    NM_015423   309 111 Missense_Mutation   H   N
AATF    TCGA-OR-A5L2    NM_012138   560 173 Missense_Mutation   E   D
AATK    TCGA-OR-A5JE    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5LR    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5LS    NM_001080395    1374    546 Missense_Mutation   D   H
AATK    TCGA-OR-A5L1    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5JH    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-PK-A5HA    NM_001080395    1374    1209    Missense_Mutation   S   I
AATK    TCGA-PK-A5HA    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5KW    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5J5    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5KU    NM_001080395    1374    541 Missense_Mutation   A   T
AATK    TCGA-OR-A5KU    NM_001080395    1374    462 Missense_Mutation   G   S
AATK    TCGA-OR-A5LI    NM_001080395    1374    1332    In_Frame_Ins    A   delinsTPA
ABCA12  TCGA-OR-A5J8    NM_173076   2595    938 Nonsense_Mutation   E   X
ABCA12  TCGA-OR-A5LJ    NM_173076   2595    2083    Missense_Mutation   S   F
ABCA12  TCGA-OR-A5J9    NM_173076   2595    229 Missense_Mutation   F   L
ABCA13  TCGA-OR-A5K5    NM_152701   5058    2950    Nonsense_Mutation   W   X
ABCA13  TCGA-OR-A5KB    NM_152701   5058    1809    Missense_Mutation   P   A
ABCA2   TCGA-OR-A5LR    NM_212533   2466    1695    Missense_Mutation   V   G
```

Fig 3. Example output for MAGI

Finally, some summary statistics and visualizations are provided for a quick overview and gut checks. The summary breaks down total mutations of each type, missing transcripts, as well as the number of mutations the processor failed to understand. Below (Fig 4) is an example of one of the visualizations, in this case showing the breakdown in mutation types for BRCA.

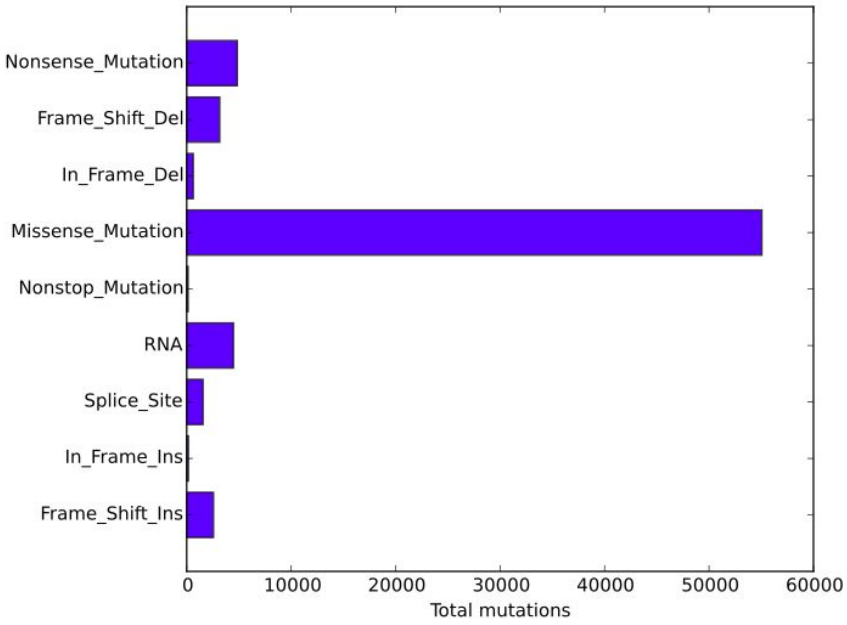


Fig 4. Example visualization for SNV processing

4. **Processing CNA Data**

While SNV data was conveniently captured in a single file based on a specification, the CNA data used for this project comes entirely from the output of the GISTIC2.0 software package. As with the SNV data, I will describe how the code processes the copy number information in terms of the input, parameters, and output.

Inputs

The processor supports two versions of the GISTIC output, although with the older version it must first do some conversion work and is unable to output data for use by MAGI due to missing information. The processor can detect if the older GISTIC (called ‘alternate’) is in use, and if the option is set it will automatically attempt to convert to the newer format.

The two table files, table_amp.conf_99.txt and table_del.conf_99.txt (for amplification and deletions respectively) are always required. They contain the information relating to the location of the CNA, as well as the region of the genome that was amplified or deleted (including the genes present in the region). Each are identified by chromosome and then by two base pair indexes (the start and end of the region containing the CNA).

The other file that is always required is the focal matrix file. This contains a tab separated matrix that is essentially genes by samples, so it is quite large. The data value for each gene/sample pair is the copy number minus two, so a normal score would be zero, a full deletion would be -2.0, and so on.

Finally, the focal segments file is required to output for MAGI, but not HotNet2 or CoMEt. This file contains tab separated columnar data about CNAs indexed by sample, with their start and end locations on chromosomes. This is referenced against the previously described files (locations are compared to find genes).

Outputs

For HotNet2 and CoMEt the data is nearly identical after processing. The only difference is each gene will also have (A) or (D) after the name to indicate an amplification or deletion.

```
TCGA-5T-A9QA    CCND1(A)    FH(A)   INTS4(A)    MCL1(A) PAFAH1B2(D) PCSK7(D)
TCGA-A1-A0SD    CCND1(A)    IGF1R(A)    INTS4(A)    MCL1(A) NRAS(D) PAX7(D) PTE
TCGA-A1-A0SE    CCND1(A)
TCGA-A1-A0SF    CDKN2A(D)   CPSF6(A)    IGF1R(A)    NRAS(D) PAX7(D) RBM15(D)
TCGA-A1-A0SG    IL6ST(D)
TCGA-A1-A0SH    ING5(D) MYC(A)  PAFAH1B2(D) PAX7(D) PCSK7(D)    PIK3CA(A)   POU
TCGA-A1-A0SJ    BRCA1(D)    CCND1(A)    CPSF6(A)    ETV4(D) FH(A)   INTS4(A)
TCGA-A1-A0SK    ING5(D) NRAS(D) RB1(D)  RBM15(D)    SRXN1(D)    TCF3(D) TERT(A)
TCGA-A1-A0SM    BRCA1(D)    CCND1(A)    ERBB2(A)    ETV4(D) FH(A)   INTS4(A)
TCGA-A1-A0SN    CCND1(A)    ERBB2(A)    FOXA1(A)    ING5(D) TUBD1(A)    ZNF217(
TCGA-A1-A0SO    COX18(A)    CSMD1(D)    FAM208B(A)  FGD5(A) FH(A)   FOXA1(A)
TCGA-A1-A0SP    CSMD1(D)    MYC(A)  PAFAH1B2(D) PCSK7(D)    POU2AF1(D)  SDHD(D)
TCGA-A1-A0SQ    CCND1(A)    INTS4(A)    ZNF217(A)
TCGA-A2-A04P    CCND1(A)    COX18(A)    FAM208B(A)  IGF1R(A)    MYC(A)  PAX7(D)
TCGA-A2-A04R    CPSF6(A)    MCL1(A) RB1(D)  TUBD1(A)    ZNF217(A)
TCGA-A2-A04T    CCND1(A)    CCNE1(A)    CDKN2A(D)   COX18(A)    CPSF6(A)    ERC
TCGA-A2-A04U    CCND1(A)    CCNE1(A)    FAM208B(A)  FGD5(A) HRAS(D) IGF1R(A)
TCGA-A2-A04V    CCND1(A)    MYC(A)  ZNF703(A)
TCGA-A2-A04W    BRCA1(D)    COX18(A)    ERBB2(A)    ETV4(D) FOXA1(A)    PAX7(D)
TCGA-A2-A04X    BRCA1(D)    CSMD1(D)    ERBB2(A)    ETV4(D) MCL1(A) PAFAH1B2(D)
TCGA-A2-A04Y    CSMD1(D)    FH(A)   ING5(D) MYC(A)  SMYD3(A)    TUBD1(A)    ZNF
TCGA-A2-A0CK    CCND1(A)
TCGA-A2-A0CL    CCND1(A)    COX18(A)    INTS4(A)    PTEN(D)
TCGA-A2-A0CM    CCNE1(A)    CPSF6(A)    CSMD1(D)    FAM208B(A)  ING1(A) MAP2K4(
TCGA-A2-A0CO    NRAS(D) RBM15(D)    TRIM33(D)
TCGA-A2-A0CT    CCND1(A)    FGD5(A) PAX7(D) RN7SKP204(A)    SDHB(D) THYN1(D)
TCGA-A2-A0CU    ERBB2(A)    ING5(D) MAP2K4(D)   MYC(A)  POU2AF1(D)  SDHD(D) ZNF
TCGA-A2-A0CV    MCL1(A)
TCGA-A2-A0CW    BRCA1(D)    CCNE1(A)    ERCC5(A)    ETV4(D) FGD5(A) FOXA1(A)
```

Fig 5. Example CNA output for HotNet/CoMEt

The output for MAGI resembles the focal segments file, it is columnar and tab separated, with gene-sample-CNA information. It gives the affected gene, the sample, the type of CNA, and the beginning and ending of the CNA.

ABL1	TCGA-AN-A0AL	Del	132859382	140938752
ABL1	TCGA-A7-A6VX	Del	131352373	140938752
ABL1	TCGA-EW-A1P8	Del	133035674	140938752
ABL1	TCGA-AO-A129	Del	98697963	139701666
ABL1	TCGA-A7-A6VW	Del	94791084	138866686
ABL1	TCGA-C8-A1HN	Del	132078657	134039800
ABL1	TCGA-EW-A1P4	Del	113185522	140938752
ABL1	TCGA-D8-A27H	Del	131087575	140938752
AKT1	TCGA-AR-A1AU	Del	60574448	105986740
AKT1	TCGA-A8-A09R	Del	65456525	105986740
AKT1	TCGA-AN-A0FD	Del	104216649	105986740
AKT1	TCGA-AO-A12E	Del	103208802	105986740
AKT1	TCGA-D8-A1XL	Del	90509564	105986740
AKT1	TCGA-A2-A0D4	Del	63374669	105986740
AKT1	TCGA-A8-A09A	Del	63591089	105986740

Fig 6. Example CNA output for MAGI

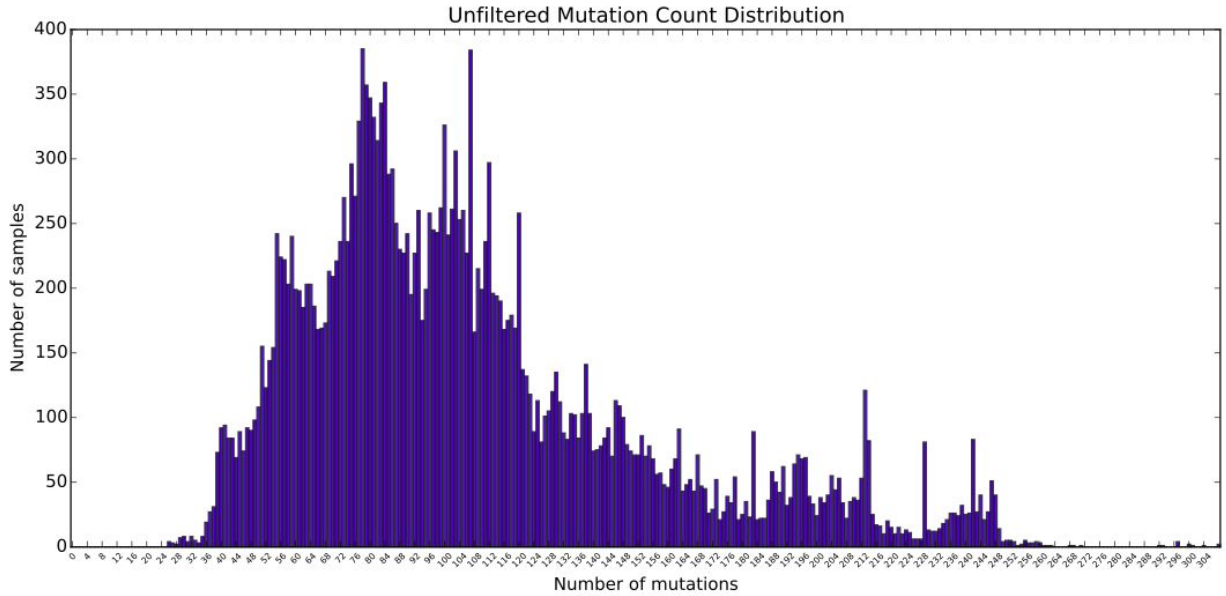


Fig 7. Visualization for the mutation per sample distribution for the BRCA cohort

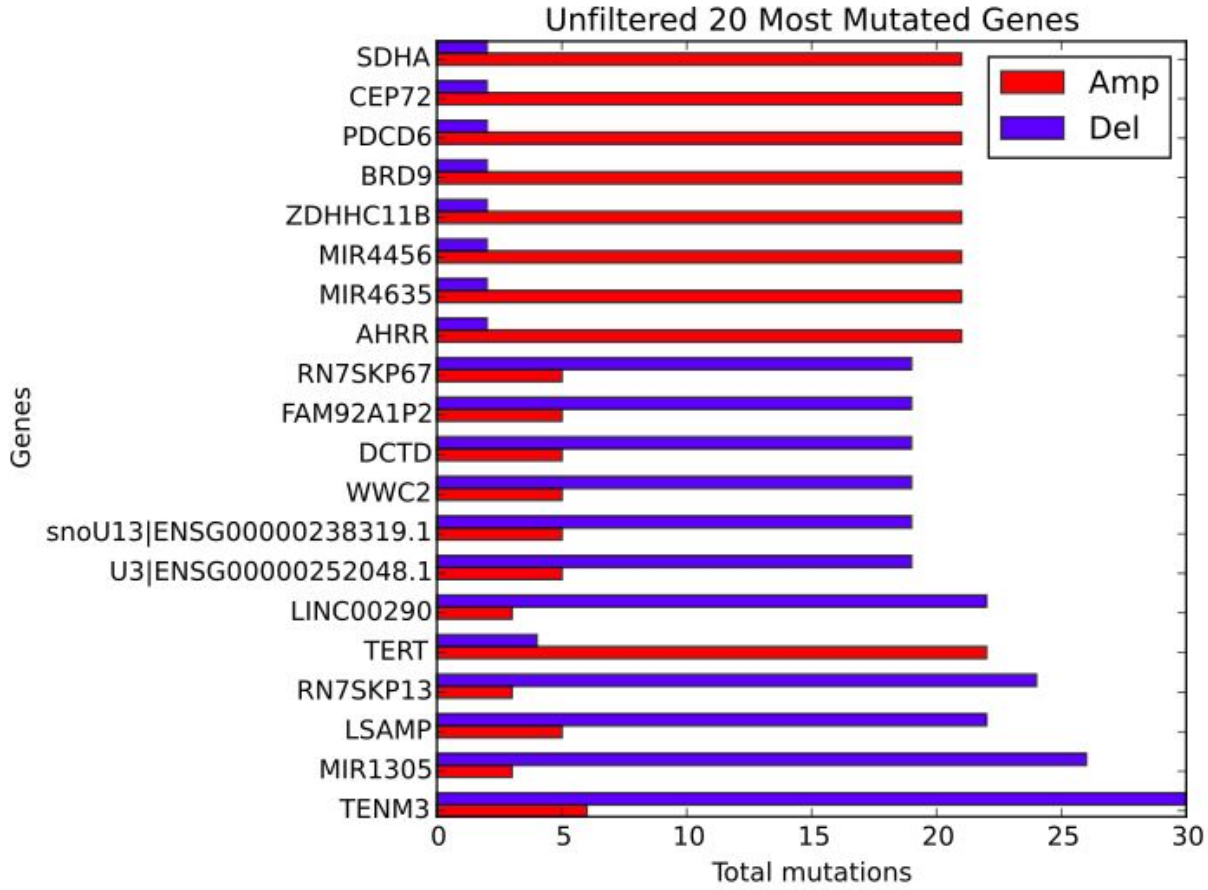


Fig 8. Visualization for the ACC cohort for top CNA mutated genes

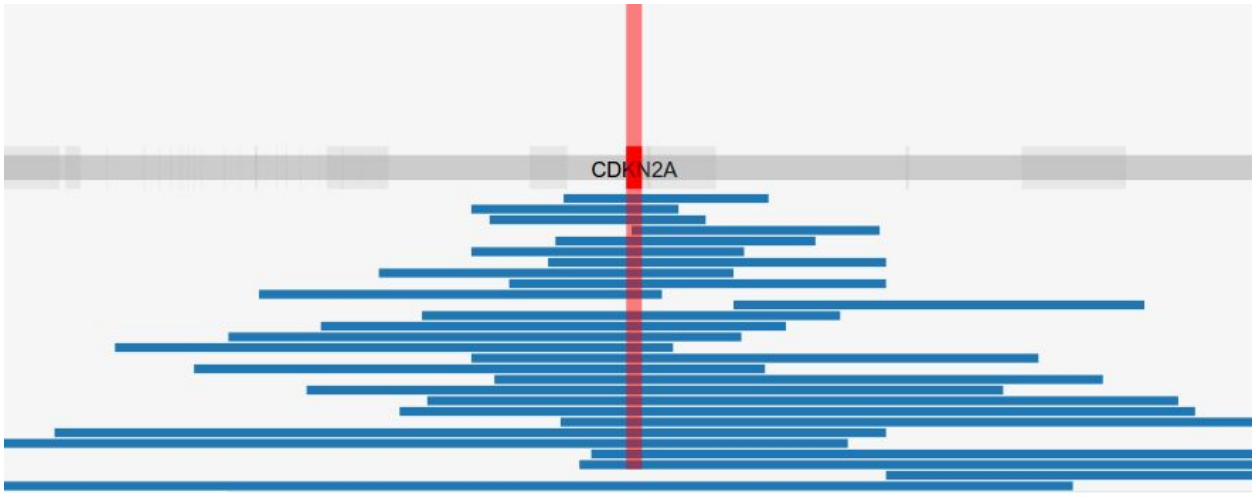


Fig 9. MAGI Visualization of CNA data

5. **Automation with GDAC, Firehose and FireBrowse**

The final part of the project was to automatically pull the data to be processed. As previously mentioned, the Broad GDAC is the sole source of both MAF and GISTIC2.0 (SNV and CNA) data. Fortunately, the Broad provided tools to retrieve the latest datasets and analysis runs. The two tools I looked at were Firehose and FireBrowse, where I ultimately chose to use the older Firehose tool.

Firehose vs. FireBrowse

FireBrowse initially seemed more promising, as it provided an easy API through the use of GET requests that could be easily constructed and run. This meant it would be easy to tune queries, and the data could be retrieved programmatically without the need to save to the disk and then read the data from a file. It could also return data in JSON, tsv, or csv formats. However, the API had two limitations at the time of my project that made it not suitable for our needs.

The first was it could not return raw MAF data. It would give MAF data that had been processed, but this filtered out some of the data that MAGI used (codon changes and more) so using FireBrowse would cut out MAGI completely. Secondly, and similarly, it only provided access to some of the GISTIC2.0 output for any given data run. Again this was close to giving everything needed, but it missed out on data required for MAGI (the focal segments data).

It seemed it might be possible to get all archive data from FireBrowse initially, which would have meant reading the data from the query, then using it to download a file, but would still use the API and so been easier to tune and retrieve data. However, after some experimenting, it was simpler to use the older tool, Firehose, to accomplish the same thing.

Firehose is a CLI tool provided by GDAC that give a few simple options such as cohort and date of the runs to use (GDAC runs analyses on its raw data on approximately a monthly basis). Crucially it allows the user to give the name, or partial name, of the data pipeline they wish to access. This allowed me to pull full archives of both CNA and SNV data. Ultimately this is what led me to choose to stay with the Firehose tool over the FireBrowse API.

Automation

Once I chose the method of pulling the data, the final step was to tie the pulling and processing together. The script acts as as coordinator, it is designed to take in the SNV and CNA processors I wrote as arguments, refinements such as allowing for targeting specific cancer cohorts, as well as a function to retrieve the clinical data for each run.

Each cohort's raw data is downloaded, extracted, and the relevant files as well as the raw archives are saved for auditing. As each analysis is performed, the script cleans up and moves the data into a simple folder structure that makes it easy to quickly browse the data by cohort and targeted process.

Since the script has a higher level view of all of the data, it provides some additional analysis and metadata. It aggregates sample information by cohort in a JSON file, and outputs a manifest JSON file for use by MAGI that allows automated uploads to the tool.

6. **Further Work**

Making use of the FireBrowse API would be very useful, it would allow for simpler code and less file manipulation. As of this writing it still does not have all of the capability, but one improvement could be to start integrating the aspects that fulfill the data needs already and then working with the FireBrowse.

The Broad institute developed a cloud analysis platform named FireCloud, I investigated its use and could not find a use case for this particular project where the cloud platform would be easier or more simple to use than the current format. However, there is something fundamentally attractive about having this automated solution running in the cloud, closer to the data source, and accessible via some simpler API.

7. **Acknowledgements**

I want to thank my advisor, Dr. Ben Raphael, for his guidance on this project. I would also like to thank Dr. Max Leiserson and Dr. Hsin-Ta Wu, for their invaluable help in understanding the problem at hand and providing support for the implementation of this project.