May 2015

Streamlined Mutation Analysis for Clinical Next Generation Sequencing Data

Rebecca Ann Stolarczyk
Worcester Polytechnic Institute

STREAMLINED MUTATION ANALYSIS FOR CLINICAL NEXT GENERATION SEQUENCING DATA

An Interactive Qualifying Project Report
Submitted to the Faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Engineering

In Cooperation With
The Laboratory of Diagnostic Molecular Oncology
Department of Pathology
UMass Memorial Medical Center

By:

_____

Rebecca Stolarczyk

May 9, 2015

APPROVED:

_____

Professor Elizabeth Ryder, Ph.D.
Project Advisor

Abstract

The use of Next Generation Sequencing (NGS) for the diagnosis and treatment of cancer has become increasingly prevalent in the field of molecular oncology. The Laboratory of Diagnostic Molecular Oncology at UMass Medical Center recently implemented a clinical NGS assay to test cancer specimens received from the hospital, but lacked an adequate bioinformatics solution for analyzing the data. To streamline the analysis process, which was previously completed manually by a laboratory technician, a program was developed in Excel Visual Basic that filters raw data for review and compiles a concise report from the mutational findings and patient demographics. Testing was conducted using data from a variety of different tumor profiles to ensure technical accuracy. Feedback was also collected from the laboratory staff in regards to the program's usability, and adequate adjustments were made to the program in response. It was found that using the program effectively reduced analysis time from 3 hours to 45 minutes per case, allowing technicians to complete significantly more cases in a single day. Ultimately, this solution shortens the turnaround time for clinical specimens, reduces the likelihood of errors, and improves patient care for the hospital.

Acknowledgements

I would first like to thank Professor Liz Ryder for advising this IQP and providing ongoing support over the course of the project. She has helped greatly by proofreading several drafts of the report and offering feedback on both the writing and technical aspects of the project.

I would also like to thank Keith Tomaszewicz at the Diagnostic Molecular Oncology Laboratory for his overall guidance and assistance with this project. Keith provided many ideas which were incorporated into the final program, as well as continual feedback and suggestions on my work. Keith also provided data that was used for testing the program and the technical insight needed to make the program clinically accurate. Dr. Lloyd Hutchinson, the laboratory director, also helped to make this project successful by allowing me to utilize the laboratory and bioinformatics software.

Finally, I wish to thank the entire staff of the laboratory for their corporation and support, as well as for providing feedback in response to the program. With these suggestions, I was able to improve the program and successfully complete this IQP.

Table of Contents

List of Tables

List of Figures

Executive Summary

Next Generation Sequencing assays have been established in laboratories around the world as the new “gold standard” for cancer diagnosis and treatment. The results, although highly accurate and reproducible, yield large amounts of data that require significant processing in order to determine clinical relevance. As a result, laboratories with limited staff and resources may lack access to adequate bioinformatics solutions for NGS data. Several bioinformatics programs currently exist on the market, however most are extremely expensive and often not within the budget for a small laboratory.

The Laboratory of Diagnostic Molecular Oncology (DMO) at UMass Memorial Medical Center recently implemented an NGS assay for clinical cancer specimens, but lacked a reasonable alternative to expensive bioinformatics software. The assay currently includes two panels, an Ampliseq Cancer Hotspot panel and a Qiagen Myeloid Neoplasm panel for the detection of mutations in solid and blood-related tumors, respectively. Combined, the panels cover over 90 genes that are commonly mutated in cancer patients. To analyze the data generated by sequencing, the laboratory compares findings from two different pieces of software: Variant Caller by Ion Torrent and Nextgene by Softgenetics. Initially, laboratory technicians were reviewing this data manually, a process that proved to be unnecessarily tedious and time consuming. This project aimed to develop a streamlined bioinformatics solution for analyzing clinical NGS data and to provide a service to the community that would otherwise be unavailable with current resources.

To begin, the primary objectives of the project were discussed with the DMO Lab and a concise list of features was developed. A design for the program was then created using the concept of dividing analysis into four phases: pre-review, post-review, Meditech reporting, and

Supercontrol analysis. Each of these steps was automated using a series of Visual Basic modules that are sequentially triggered by the click of a button.

The first module, Pre-Review, involves filtering and sorting the raw data based on specified attributes such as mutation frequency and coverage. After running this script, the technician reviews the remaining unfiltered variants and determines whether or not they are true positive mutations. This information is manually entered into the program by the technician. The next module, Post-Review, is then run to create a printable report containing relevant patient and mutation information. Once the report is prepared, the technician manually checks for mutations in tumor-specific regions-of-interest to ensure that there are no false-negatives being reported. Finally, the technician runs the Meditech module to compile an electronic report with clinically relevant diagnostic information and raw data. Supercontrol modules were also developed for both the Ampliseq and Myeloid panels to ensure sensitivity DNA controls passed quality control metrics for each run and detected a pre-determined mutational profile.

To ensure technical accuracy, several sets of data from a variety of tumor profiles were run through the program and compared against previous mutation findings. During testing, it was found that analysis time was reduced from an average of 3 hours to 45 minutes per case. Feedback was also collected from the laboratory staff to identify possible areas of improvement for the program. Some primary areas of concern included detecting low-frequency mutations, minimizing the possibility of user error, and customizing the program for different types of tumors. Changes were made in response to these suggestions, and testing was repeated to ensure that the accuracy of the program was not compromised.

The bioinformatics solution developed for this project effectively improved the accuracy of NGS analysis and significantly reduced the time needed to review cases. As a result,

physicians will be provided with more detailed diagnosis reports, and patients who are being treated at the hospital will receive results more quickly. Ultimately, this project will streamline the NGS workflow in the DMO Laboratory, reduce the likelihood of incorrect reporting, and improve cancer patient care as a whole for UMass Memorial Medical Center.

1. Introduction

Recent advancements in the field of biotechnology have led to new approaches for the diagnosis and treatment of cancer in both research and clinical settings. These developments have significantly improved patient care and survival rates of this disease across the globe. Over the past forty years, the five-year relative survival rate has increased from 49% to 68% for all cancers combined. The steady decline of the cancer death rate can be attributed to improved prevention techniques, increased use of early detection methods, and the progression of new therapeutic approaches (Cancer Facts and Figures, 2014).

The development of Next Generation Sequencing at the turn of the twentieth century marked the dawn of a new age for cancer research and patient care. The high-throughput technology of this assay has proven to be significantly more efficient than "first generation" sequencing methods which previously dominated the industry. This novel approach to genetic variant detection has revolutionized the field of oncology and improved patient care in hospitals across the world.

Along with the solutions that NGS provides, however, come a number of new challenges that must be addressed. In particular, the data files produced by the instrument are incredibly large and thus require extensive filtering and processing in order to obtain comprehensible results. The downstream analysis of NGS data calls for more in-depth investigation than previous sequencing assays and often requires an entire bioinformatics team to interpret the results. In laboratories that lack these resources, the processing of NGS data poses a critical challenge. This is especially pertinent in clinical settings, where the turnaround time of test results must be minimized without compromising the accuracy of the assay. The lack of NGS data analysis solutions in laboratories with limited resources is an issue that calls for the development of new

bioinformatics approaches that streamline this process and are readily available to researchers and clinicians.

1.1 The Nature of Cancer

Cancer, at the most fundamental level, is the inability of cells to die. This phenomenon occurs in cells with acquired genetic mutations that trigger abnormal cell growth and provide a survival advantage over normal surrounding tissue. In cancer, the integrity of cells is compromised and many of the properties that are responsible for healthy organism development become the mechanisms which drive tumor growth. Although cell autonomy and versatility are necessary for the immense diversity of tissue in the body, they have the potential to disrupt normal cell maintenance and function. Metazoan cells possess an entire organismic genome, and many retain the ability to proliferate after the full development of an organism. These key features, necessary for organism complexity and species evolution, can thereby also provide a framework for irregular and detrimental cell behavior (Weinberg, 2014).

1.1.1 Origination

The origination of cancer in an individual can be attributed to the corruption of a cell's genomic sequence and the acquisition of a novel and often irregular phenotype. Upon the accruing of these mutations, cells may gain access to genetic information that they are normally denied and begin behaving in ways that are incompatible with the expected function of the tissue. In particular, genetic mutations that alter the regulation of cellular proliferation pose a significant threat to surrounding cells and ultimately the organism as a whole. Unlike normal cells that function in response to their environment, cancer cells, driven by their atypical and highly

unstable genome, fail to collaborate with surrounding tissue and divide irrepressibly (Weinberg, 2014).

1.1.2 Classification

The classification of cancer is predominantly based on the anatomical location in which the disease originates. The vast majority of cancers can be classified in one of the three main categories: carcinoma, sarcoma, and hematopoietic malignancies.

Carcinomas describe cancers which develop in epithelial cells (ie. skin, glandular tissues, mucosal membranes) and metastasize via the lymphatic channels. Carcinomas, which are the most commonly diagnosed cancer, are further divided into adenocarcinomas (lung, breast, colon, pancreas, etc.), squamous cell carcinomas (skin, esophagus, cervix, etc.), and numerous other organ-specific diseases such as renal cell carcinoma and small-cell lung carcinoma.

Sarcomas, which are relatively rare, arise from cells of mesenchymal origin (ie. muscle, bone, cartilage, fat). These cancers are spread through the blood stream and include osteosarcoma (bone-forming cells), liposarcoma (fat cells), and leiomyosarcoma (smooth muscle cells).

Lymphoma, leukemia, and myeloma arise from blood-forming tissues and hence are considered hematopoietic malignancies. These cancers differ primarily in the mechanisms by which they spread throughout the body. Lymphomas infiltrate solid organs via the lymphatic system and, in some rare cases, the blood stream. These malignancies are further classified as Non-Hodgkin’s and Hodgkin’s lymphoma. Conversely, leukemia describes the presence of malignant hematopoietic cells in peripheral blood. These malignancies include acute lymphocytic leukemia, acute myelogenous leukemia, chronic myelogenous leukemia, and chronic lymphocytic leukemia. Finally, multiple myeloma arises from malignant plasma cells

residing in the bone marrow which disturb the production of normal hematopoietic cells (Weinberg, 2014).

1.2 Molecular Oncology

The study of molecular oncology focuses predominantly on the investigation of tumors at the molecular level and the development of targeted therapies against these malignancies. Molecular oncology provides a novel approach to cancer diagnosis and treatment, and ultimately aims to reduce mortality of the disease as a whole. Emphasis is placed on three principles which address malignancies across the different stages of the tumor progression pathway: identification of germline and somatic mutations which may pre-dispose an individual to the disease, early detection and diagnosis of cancer, and the development of new therapies that target cancer cells at the molecular level (Wagener, 2001).

The clonal nature of cancer cells provides a pathway for tumor development and a framework for the multistage model of carcinogenesis. This model describes the progression of cancer in terms of a series of pathway events that occur as a result of acquired mutations in tumors (Barcellos-Hoff et al., 2013). The highly unstable nature of cancer cells can be attributed to these genetic and epigenetic[1] abnormalities that interfere with critical cellular processes such as DNA repair and signal transmission. As a result, processes such as proliferation, differentiation[2], and apoptosis[3] can be disrupted and ultimately lead to the development of a manifestly malignant tumor (Camacho, 2014).

[1] *Epigenetic*- Of or relating to factors that influence genetics other than an individual's DNA, such as histone modifications and RNA-associated silencing.
[2] *Differentiation*- The process by which less specialized cells become a more specialized cell type. For example, mesenchymal cells differentiate into osteoblasts (bone), myoblasts (muscle), fibroblasts (tendons and ligaments), and a number of other cell types.
[3] *Apoptosis*- The highly regulated process occurring in multicellular organisms by which cells that are no longer needed commit "cell suicide", or programmed cell death.

1.2.1 The Significance of Genetic Mutations

Mutations which influence these pathway events can be of germinal or somatic origin. The former occurs in tissue responsible for the development of sex cells and, unlike somatic mutations, can be passed on to progeny if the mutated sex cell participates in fertilization. Because these mutations are often recessive, it is possible for an individual with a normal phenotype to possess an undetected heterozygous germinal mutation (Griffiths, 2000). These mutations may predispose an individual to a variety of diseases, including cancer, despite their deceivingly normal phenotype (Figure 1). Germinal mutations can be detected using a variety of approaches, including whole-genome sequencing and targeted sequencing, which allow researchers to identify particular genes of interest that may ultimately play a part in the tumor progression pathway (Detecting Germline Mutations, 2014).

Conversely, somatic mutations originate in cells of somatic tissue and are often marked by a mutant sector: a clone of genetically identical mutant cells that have originated from a single mutated progenitor in the tissue. In diploids, only a dominant mutation would appear in the phenotype of the clone of cells, whereas a recessive mutation would not be expressed (Griffiths, 2000). These mutations can be detected in cancer using a tumor versus normal sequencing approach, in which both tumor cells and surrounding normal tissue are sequenced and mutations occurring only in the cancer tissue (ie. somatic) are identified in comparison to germline mutations that exist in the surrounding normal tissue (Somatic Mutations, 2014). The progression of a normal cell to a state of malignancy via acquired somatic mutations is illustrated in Figure 1.

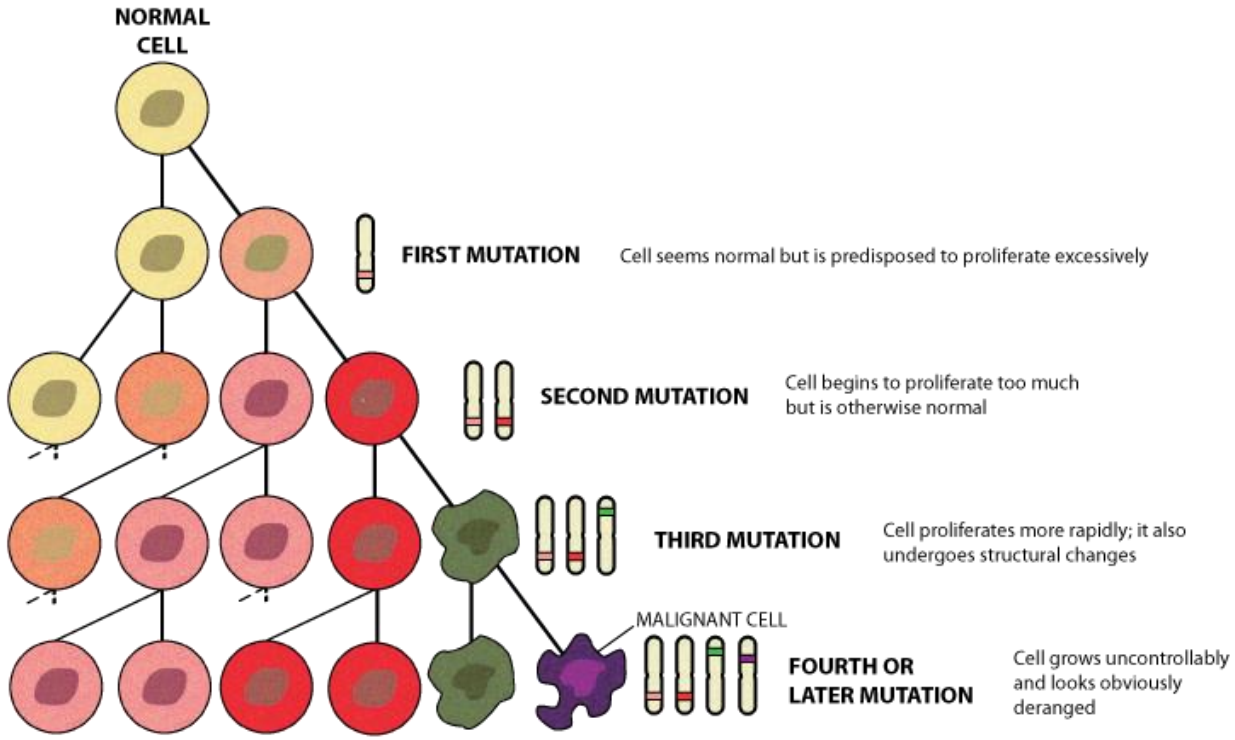


Figure 1. The Tumor Progression Pathway. A normal cell first obtains a spontaneous somatic mutation which has no significant effect on cell behavior but leads to a predisposition of excessive proliferation. When the mutated cell divides, a second mutation may be accumulated which causes the cell to proliferate excessively but remains normal otherwise. Another round of cell division can lead to a third mutation, in which proliferation increases and structural changes occur that alter the appearance of the cell. After several more mutations, the cell line becomes malignant, grows uncontrollably, and is visibly misshapen (Rai, 2011).

1.2.2 The Applications of Targeted Therapies

The molecular approach of cancer treatment involves the use of targeted therapies which inhibit tumor growth by interfering with specific molecules responsible for progression and spread of the disease, termed "molecular targets". These therapies can generally be classified in one of two categories: monoclonal antibodies or small molecules. The former targets antigens present on the surface of the cell, such as extracellular growth factors or transmembrane receptors. Conversely, small molecules aim to disrupt enzymatic activity of proteins within the cell and thus can penetrate the cell membrane in order to reach their target. Treatments are chosen with respect to the cancer type and other factors which may affect response to therapy

such as patient demographics. Ultimately, targeted therapy aims to improve the precision of cancer treatment while simultaneously lessening the side effects experienced by the patient (Abramson, 2014).

1.3 Next Generation Sequencing

The deciphering and analysis of DNA sequences is critical for virtually all aspects of biological research as well as a number of clinical assays. A new approach to this technique, termed Next Generation Sequencing (NGS), first began surfacing at the turn of the twenty-first century and provided a more efficient alternative for DNA sequencing. Within a few years, a number of NGS platforms were developed (including the Ion Torrent, Illumina, Roche 454, and SOLiD) and began replacing conventional sequencing protocols, such as the Sanger method, that had previously dominated the industry. The results obtained from this novel technology are highly accurate, reproducible, and cost efficient, providing an excellent alternative to previous sequencing assays. NGS has since become an integral part of laboratory methodologies across the globe with applications in both clinical and research settings (Introduction to Next-Generation Sequencing, 2013).

1.3.1 Workflow

The NGS workflow begins with the preparation of a library for each sample by performing multiplex PCR[4] in order to amplify multiple target sequences of the template DNA. This reaction requires strategic preparation in order to choose which primers to use and which portions of the genome to amplify. Each sample is ligated with an oligonucleotide adapter

[4] *Multiplex PCR*- A modification of conventional polymerase chain reaction that uses multiple primers to amplify several regions of DNA in order to rapidly screen for variations in a large gene.

(Figure 2) and barcoded using a specific sequence in order to identify the original source of the DNA using either emulsion or polony PCR (Figure 3A). This reaction indexes and prepares the fragments with the biochemistry necessary for sequencing (Vierstraete, 2012).

Figure 2. Ligation of Oligonucleotide Adapter. During library preparation, adapters are ligated to the partially digested sample DNA in order to index the fragments and prepare them for sequencing (Introduction to Next-Generation Sequencing, 2013).

The barcoded libraries, each containing small segments of the sample's DNA, are then pooled and sequenced in parallel. There are a number of techniques that exist for this step, including semiconductor sequencing (Ion Torrent), pyrosequencing (454), sequencing by ligation (SOLiD), and reversible terminator sequencing (Illumina). Although the sequencing methods vary from platform to platform, the ultimate principal remains the same (Vierstraete, 2012). The strands of DNA generated from the sequencing reaction, called reads, are composed of both the fragment sequence and the barcode for that particular sample (Figure 3B). The reads are de-multiplexed and separated by sample using the barcode sequence (Figure 3C). Finally, each set

of reads is aligned to the reference genome in order to identify variants in the sample sequence (Figure 3D) (Introduction to Next-Generation Sequencing, 2013).

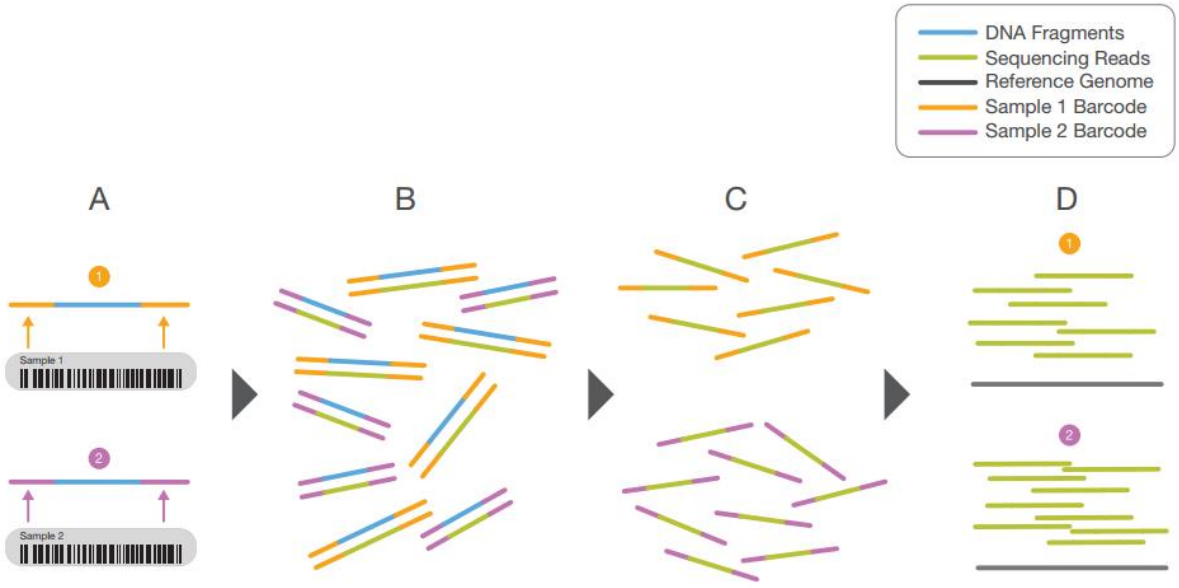


Figure 3. Conceptual Workflow of Next Generation Sequencing. The figure illustrates the primary principles of amplified single molecule sequencing using two barcoded samples (A) which are sequenced (B), differentiated (C), and matched to a reference genome (D) (Introduction to Next-Generation Sequencing, 2013).

1.3.2 Data Analysis and Interpretation

Raw NGS data obtained directly from the instrument is generally in the FASTQ format: a text file in which the millions of reads from sequencing are stored. These reads are short sequences of DNA composed of the four bases (A, C, G, T) that directly correspond to the nucleotides incorporated during the sequencing reaction. A read-mapping software is used for the process of reference mapping, in which the reads in the data output file are attempted to be aligned to a reference genome sequence by the program. Once the reads are matched, the genome of the sample can be compiled by assembling the reads in specific locations on their respective chromosomes (Gullapalli et al., 2012).

After the assembly process has been completed, the data is prepared for downstream analysis. Just as the sequencing techniques of this assay vary from platform to platform, the data output and analysis methods also differ for each NGS system. A number of applications are available from NGS companies, software vendors, and academic institutions that provide a variety of data processing algorithms. These functions may include specialized assembly of sequencing reads for the detailed genetic analysis of an organism or quantification of reads for investigation of gene expression levels. Unique methodologies exist for a variety of research applications and enable the universal techniques of Next Generation Sequencing to be applied to more specific data-processing tasks (Introduction to Next-Generation Sequencing, 2013).

1.3.3 Applications in Molecular Oncology

Next Generation Sequencing provides a comprehensive and efficient solution for cancer research and clinical diagnosis. The high accuracy and reproducibility of this assay allow for the detection of somatic mutations that have been acquired in tumor cells and germline mutations that may predispose individuals to cancer. Additionally, NGS can be used to identify other genetic factors that may influence tumor behavior such as gene expression[5], epigenetic changes, and chromosomal abnormalities[6]. A number of NGS approaches exist for tumor profiling[7], including whole-genome sequencing, whole-exome sequencing, targeted sequencing, and deep sequencing. Whole-genome sequencing provides the best picture of an individual's genetic profile, including SNPs, insertion, deletions, and structural rearrangements. Whole-exome

[5] *Gene expression*- The synthesis of a functional gene product from the information stored in a gene. These products include proteins and functional RNA.
[6] *Chromosomal abnormalities*- Also termed chromosome anomaly, these are portions of chromosomal DNA that are irregular. The three primary single chromosome mutations are inversion, deletion, and duplication. Examples of these abnormalities include structural aberrations and an atypical number of chromosomes. These abnormalities are common in cancer cells.
[7] *Tumor Profiling*- The classification of tumors into subtypes in order to individualize cancer treatment.

sequencing is slightly more selective, and involves sequencing only the subset of DNA known as exons, which encode proteins. Targeted sequencing, which focuses only on a small subset of the genome, is common for cancer diagnosis as it is cost-effective and allows clinicians to sequence only particular regions of interest. Finally, deep sequencing is used for highly polyclonal tumors and low purity tumors because it increases the sensitivity and complexity of the results. These methods can be used both individually and in combination to predict, diagnose, and treat cancer. (Cancer Genomics, 2014)

The processing of clinical NGS data is critical because molecular profiling of cancer depends greatly on the unique alterations in the patient's genome. Often, the concept of variant detection is used to identify clinically relevant deviations in the sample DNA. These mutations can appear in numerous forms, from single-base point mutations to extensive insertions and deletions of chromosomal material resulting from the highly unstable nature of cancer genomes (Gullapalli et al., 2012).

In order to determine the significance of mutations in a patient's DNA, several factors must be considered. On average, the genome of an individual contains approximately 10 million single-base variations, termed single nucleotide polymorphisms (SNPs). These are the most commonly occurring genetic variations in humans, and generally have no effect on growth or development. When identifying clinically relevant mutations in a cancer specimen, it is important to account for SNPs that are also present in surrounding normal tissue and have no effect on tumor progression (Single Nucleotide Polymorphisms, 2014).

Additionally, different cancers may call for the investigation of particular genes and regions of interest in the patient's DNA. Specific genes and locations of the genome have been identified in numerous tumors as mutation "hotspots". This information may aid in the diagnosis

of a specimen or the molecular profiling[8] of a tumor. For example, mutations of the NPM1 gene are the most common genetic abnormalities in patients with acute myeloid leukemia (AML). In the diagnosis and treatment of AML, NPM1 mutations are therefore considered clinically relevant and can provide critical information about the prognosis and long-term implications of the disease (Verhaak, 2005).

1.3.4 Ethical Considerations

The recent development of Next Generation Sequencing has also prompted a number of new challenges in regards to the ethical implications of this assay. The sequencing of larger portions of the genome with NGS inherently leads to an increased frequency of unexpected variants in the data. These so-called incidental findings (IF) refer to detected mutations that are not directly related to the disease under investigation but may be clinically relevant for diagnosis or treatment of the patient. For example, a test run with NGS that aims to identify the presence of a particular mutation in a cancer patient may also detect an additional mutation with potential significance. In this case, it becomes unclear whether this information should be disclosed to the patient and, if so, what implications it will have for the individual. For clinical IF, methods have been proposed which classify the variants in one of three ways: "clinically actionable", "clinically valid but not directly actionable", and "unknown or of no clinical significance." This would provide a framework for the clinician in regards to disclosing NGS test results to patients (Davey 2014).

Another primary ethical concern associated with NGS is the education of doctors about this new technology and its implications in the field of healthcare. Because much of the territory

[8] . Molecular profiling- Examination of the genetic characteristics of a patient's tumor in order to provide specialized therapy.

of NGS is uncharted, new discoveries are constantly being made which require interpretation by clinicians. As new genetic variants are discovered, hypotheses must be made in regards to the pertinence to a patient's condition. Much of this hypothesizing involves a degree of speculation which makes it difficult to determine which results are clinically relevant and will affect patient treatment. This is especially critical in terms of pediatric care and what information to disclose to the parents of an ill child (Lantos et al., 2011).

Finally, classification of NGS results as either clinical or research work is incredibly important for regulatory purposes. In the case of novel variants that are detected in an individual and not found in the general population, results may be used to make a diagnosis. However, these results may also be useful for investigating the relationship between the genetic abnormalities and condition of the patient. In this case, NGS work is generally classified based on the ultimate purpose of the sequencing. If used with the goal of making a diagnosis, the sequencing is considered to be for clinical purposes. Conversely, if used as a means of studying newly discovered genetic variants, the results can be classified as research. In many cases, the line between clinical and research sequencing is not as distinct and thus raises concerns in the regulation of NGS (Lantos et al., 2011).

1.4 The Software Development Lifecycle

An adaptation of the system development life cycle (SDLC) was used as the framework for designing and creating the program. The model, seen in Figure 4, involves seven consecutive phases, beginning with problem definition and concluding with software maintenance (Davis et al., 1999).

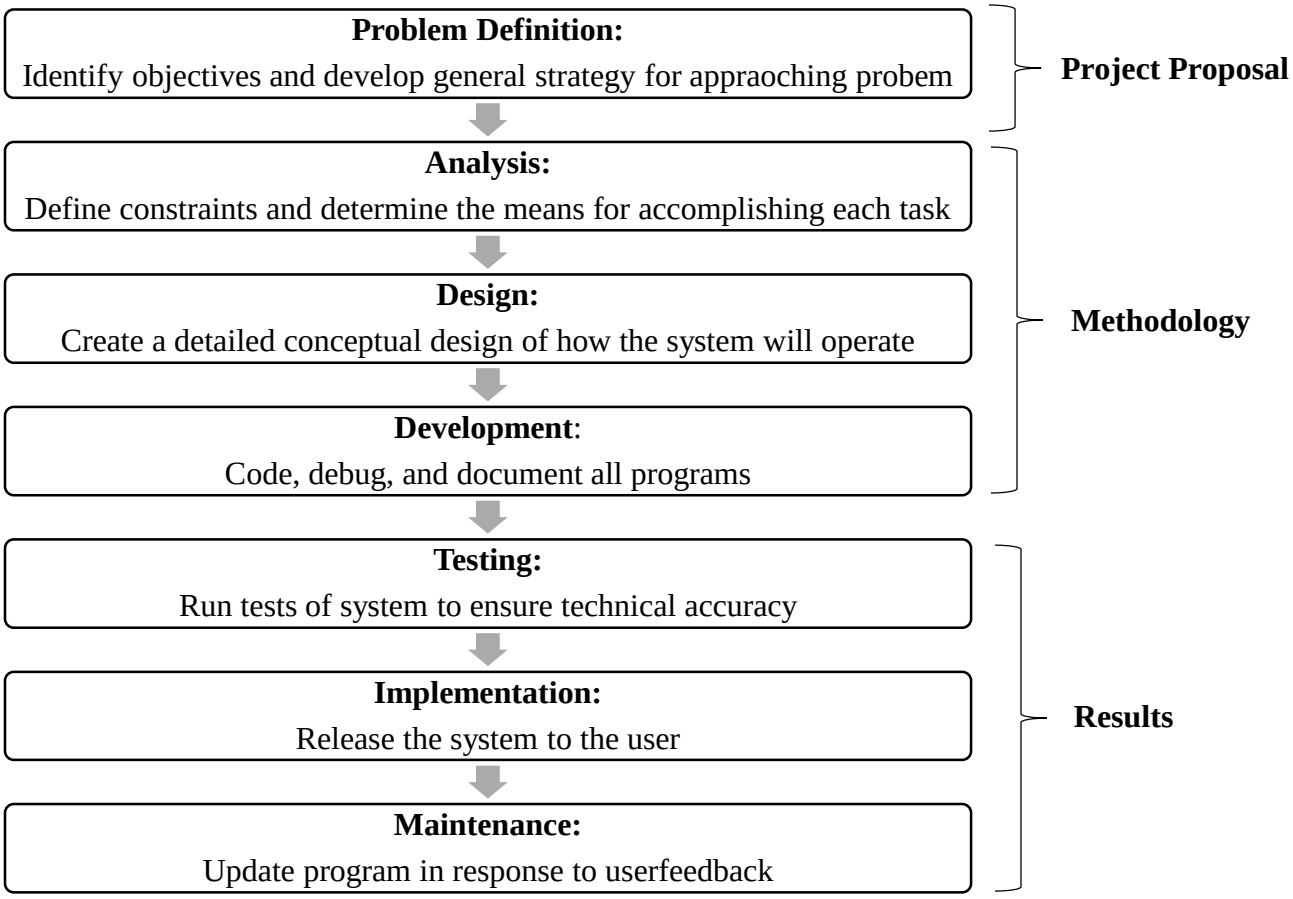


Figure 4. Waterfall Method of the System Development Life Cycle. This adaptation of the SDLC involves seven consecutive phases for creating software.

The first step, problem definition, was completed during the project proposal phase of the project (see Chapter 1.4). The next three steps (analysis, design, and development) will be discussed in detail in the Methodology (see Chapter 2). Finally, the last three steps (testing, implementation, and maintenance) will be discussed in the Results (see Chapter 3).

1.5 Project Proposal

The goal of this Interactive Qualifying Project (IQP) was to design and develop a program for the Diagnostic Molecular Oncology Lab at UMass Memorial Medical Center to aid with the analysis of clinical Next Generation Sequencing results. The laboratory recently implemented an NGS assay for the diagnosis of clinical specimens from the hospital for a

number of tumor types, including colon, pancreatic, lung, melanoma, and AML. Prior to this project, the analysis was completed manually by a medical technician and involved the comparison of results from two different software programs in order to determine which detected mutations in patient samples are real and should be reported. Additionally, medical reports for each sample were written manually by the technician. Both of these processes proved to be unnecessarily tedious and required an excessive amount of time from laboratory personnel.

An analysis template was developed in Excel Visual Basic using an extensive system of macros. This program filters the detected variants from the sequencing software based on specified technical attributes, provides a comprehensive comparison of the results from the two different programs, and compiles an automated medical report summarizing the detected mutations and other pertinent specimen information. The program automates a number of processes that were previously completed manually by lab personnel and significantly minimizes the time needed to analyze NGS data. Ultimately, the program improves turnaround-time for clinical specimens and improves patient care for the hospital as a whole.

2. Methodology

2.1 Analysis

In order to develop a conceptual design for the program, the primary needs of the project were first assessed via interview with the DMO Laboratory. A concise list of functions to be included in the final program was created, as seen below. These functions were further divided into four consecutive steps: preliminary review, post review, Meditech report generation, and Supercontrol analysis.

Pre-Review

- Prepare and format raw Nextgene and Variant caller data in a concise, easy-to-analyze report
- Filter and sort raw Nextgene and Variant caller by the following:
    - SNPs
    - Artifacts
    - Low frequency mutations
    - Low coverage mutations
    - Low frequency and coverage mutations

Post Review

- Compile a summary final report that includes the following:
    - Detected mutation overview
    - Concise Regions-of-Interest (ROI) table
    - Patient identity-check metrics
- Compile a detailed final report that includes detailed metrics of the following:

- Common variants
- Variant Caller-only mutations
- Nextgene-only mutations
- SNPs

Meditech Report Generation

- Generate a comprehensive Meditech diagnostic report from the patient demographics and mutation analysis

Supercontrol Analysis

- Generate Quality Control (QC) Metrics database entry from Supercontrol data for a given NGS run

2.2 Design

2.2.1 User Workflow

To design the program, the analysis was first approached from the user's perspective as illustrated in Figure 5. When performing NGS on a patient case, the technician first completes all sample preparation and runs the DNA on the Ion Torrent Personal Genome Machine (PGM). A targeted sequencing panel of 50 genes is used which includes a number of cancer hotspot regions. This yields raw data in the form of a binary sequencing alignment file (.bam extension). This data is run through the Variant Caller software provided by Ion Torrent, as well as the third-party Nextgene software. Both programs output a complete mutation report in the form of a text file, which is conventionally converted to an excel file by the technician for ease-of-use.

To begin the analysis process, the technician will first enter data into a patient demographics sheet with pertinent sample and run information. After ensuring that all fields are completed, the technician can add the raw data from both programs to the analysis template. In order to simplify the review process, the data will first be formatted and filtered. This will both sort the detected variants in a way that is more concise and reduce the number of mutations that need to be manually reviewed by eliminating those that meet specified parameters. After the automated "Pre-Review" process, the technician can manually review the remaining mutations and decide which are true and should be reported for the specimen. The "Post-Review" program will then transfer mutation details to both the Final Report Summary and Final Report Details pages, as well as other pertinent information such as an ROI table and patient identify-check report. A Meditech report will automatically be compiled from the patient demographics and mutation analysis, which can be copied and pasted into the Meditech software by the technician. Finally, the Supercontrol data for each PGM run can be analyzed using a separate section of the program which will generate an entry for the lab's Quality Control Metrics database.

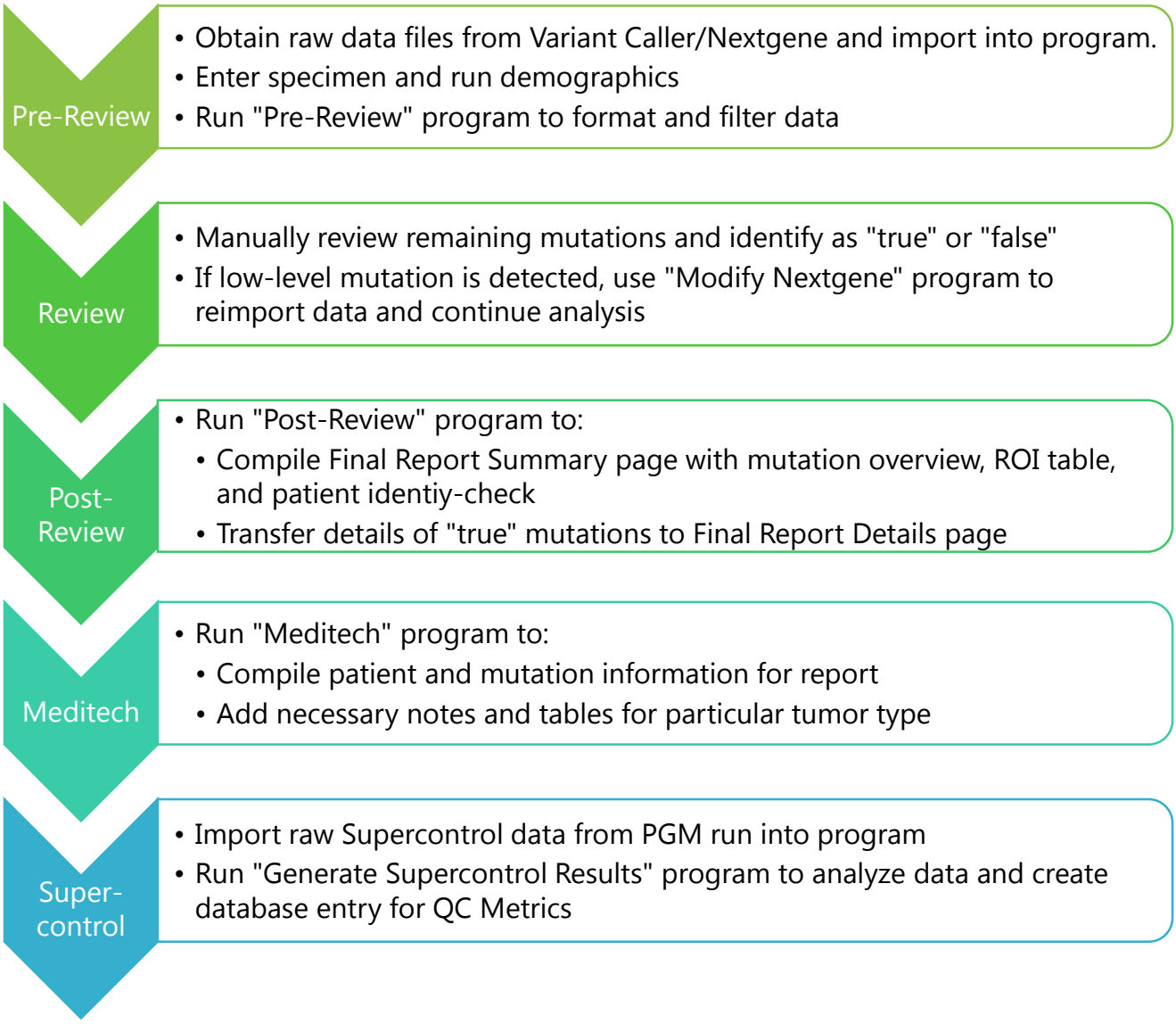


Figure 5. User Workflow. The figure illustrates the conceptual workflow for a lab technician using the program from importing data to reviewing and generating results.

2.3 Development

2.3.1 User Interface

The user interface, designed in Microsoft Excel, consists of a single workbook with a series of worksheets. Each sheet is labeled and grouped by overall function to the program. Following is a description of each group and the worksheets that are included:

Demographics: This group consists of a single worksheet designated for patient information (name, DMO #, sex, etc.), run information (PGM Run #, Barcode #, etc.), tumor information (type and percent), and technician information (name and date). The forms consist of both text and drop down lists in order to prevent discrepancies and shorten analysis time (see Figure 6).

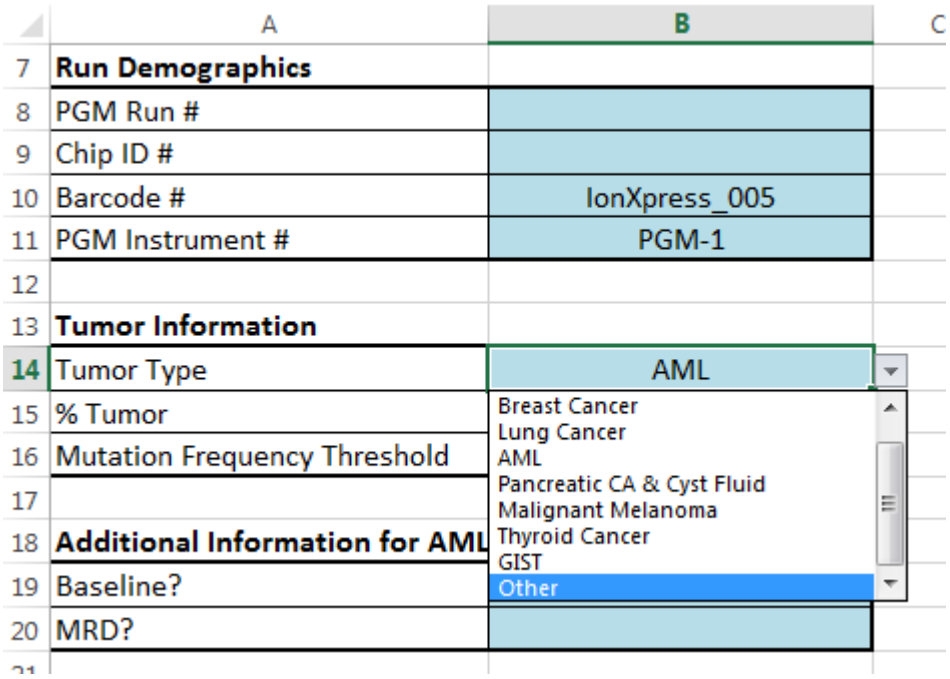


Figure 6. Demographics Dropdown Form. An excerpt of the Demographics worksheet is pictured above. The blue cells designate fields that require user input, either with text or choosing from a dropdown list (seen above for the "Tumor Type" field.

References: This worksheet contains the names of the references used in the Variant Caller and Nextgene software, as well as the current version information for the Excel program.

Imported Data: These worksheets are designated for raw, imported data and include: Allele Coverage, Nextgene, Nextgene Modified, Variant Caller, Coverage Analysis, and Sample ID. All worksheets are initially empty and require data to be pasted into cell A1 (see Figure 7).

	A	B	C	D	E	F	G	H	I	J	K
1	Chrom	Position	Ref	Variant	Allele Cal	Filter	Frequenc	Quality	Filter	Type	Allele So
2	chr1	43814979	G	A	Absent	-	0	86.87	-	SNP	Hotspot
3	chr1	43814981	G	A	Absent	-	0	82.24	-	SNP	Hotspot
4	chr1	43815008	TGG	AAA	Absent	-	0	80.86	-	MNP	Hotspot
5	chr1	43815008	TG	AA	Absent	-	0	80.86	-	MNP	Hotspot
6	chr1	43815008	T	A	Absent	-	0	80.86	-	SNP	Hotspot
7	chr1	43815008	T	C	Absent	-	0	80.86	-	SNP	Hotspot
8	chr1	43815008	TG	GC	Absent	-	0	80.86	-	MNP	Hotspot
9	chr1	43815009	G	C	Absent	-	0	83.13	-	SNP	Hotspot
10	chr1	43815009	G	T	Absent	-	0	83.13	-	SNP	Hotspot
11	chr1	43815020	G	A	Absent	-	0	77.82	-	SNP	Hotspot
12	chr1	1.15E+08	C	T	Absent	-	0	82.31	-	SNP	Hotspot
13	chr1	1.15E+08	T	A	Absent	-	0	82.07	-	SNP	Hotspot
14	chr1	1.15E+08	T	A	Absent	-	0	85.09	-	SNP	Hotspot

Figure 7. Allele Coverage Worksheet with Data. This is an excerpt of the Allele Coverage worksheet with raw, unformatted data pasted into cell A1. The data extends in both the column and row directions for several hundred lines.

Final Reports: These include both the Final Report Summary (Figure 8) and Final Report Details (Figure 9) pages. Final Report Summary requires manual user input only for the ROI table. The three primary program buttons are also located in the top right corner of this worksheet in sequential order. Final Report Details is populated entirely from the mutation reports and requires no manual input.

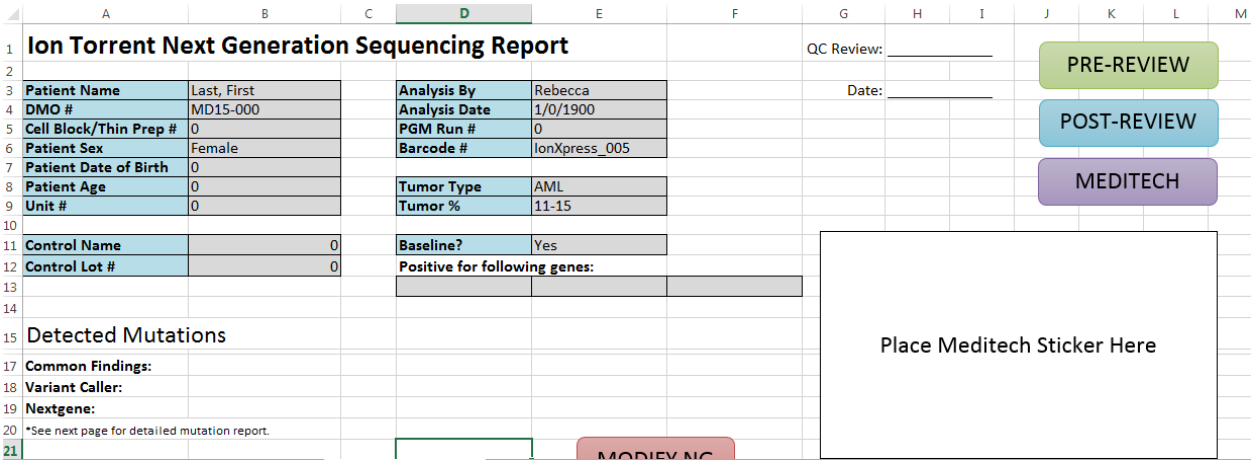


Figure 8. Excerpt of Final Report Summary Worksheet. The figure depicts a portion of the Final Report Summary page, including the autopopulated fields at the top, the section for the mutation summary ("Detected Mutations"), and the three primary program buttons in the top right corner (Pre-Review, Post-Review, and Meditech).

Ion Torrent Next Generation Sequencing Report												
Review	Chromosome	Position	Gene ID	Total Coverage VC	Total Coverage NG	Quality/ Score	Stand Bias	Frequency VC	Frequency NG	Mutation Call	Amino Acid Change	Db_snp/ Cosmic
Common Findings												
TRUE	chr12	25398284	KRAS	5171	5075	5602.8	0.5208	29.9	29.81	c.35C>CG	12G>GA	---
TRUE	chr20	57484421	GNAS	1626	1546	1130.76	0.5258	12.4	12.16	c.605G>AG	202R>HR	COSM27895
Variant Caller Findings												
NextGENE Findings												

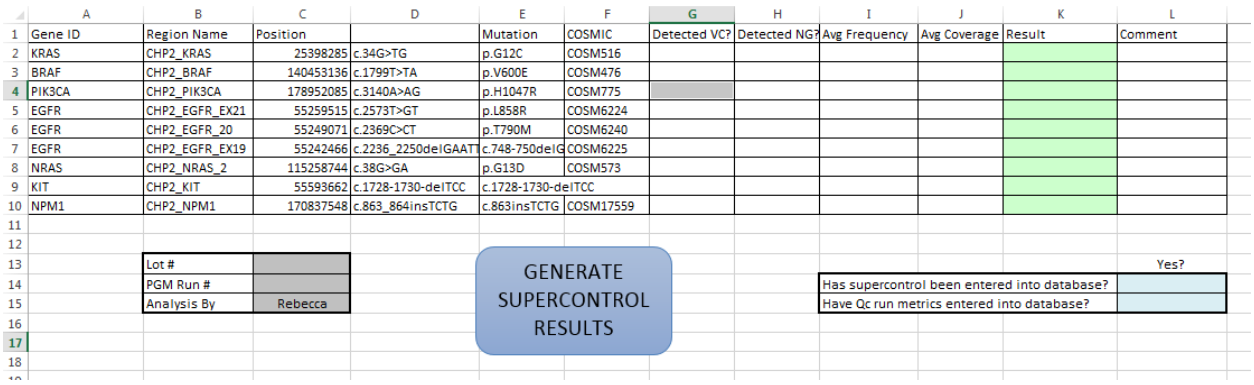
Figure 9. Excerpt of Final Report Details Worksheet. This figure depicts a portion of the Final Report Details page, including all of the fields that are included in regards to the detected mutations (such as position, gene ID, coverage, frequency, etc.) as well as the three different categories of mutations (Common Findings, Variant Caller Findings, and Nextgene Findings). The table also extends below to include SNPs detected by Nextgene.

Meditech Report: This consists of two worksheets on which the Meditech Diagnosis report (Figure 10) and Meditech Raw Data report are automatically compiled from the patient demographics and mutation analysis. Both reports include buttons for copying the entirety of the data to the clipboard in order to be pasted into the Meditech program.

MD15-000	Last, First
	COPY REPORT
NEGATIVE -	FLT3-TKI, IDH1, IDH2, JAK2, KIT mutations were NOT detected (see NOTE and Interpretation).
OTHER FINDINGS -	GNAS, KRAS MUTATIONS WERE DETECTED (see NOTE and Interpretation).
INTERPRETATION	----------
GENE #1:	KRAS (Kirsten rat sarcoma viral oncogene homolog)
MUTATION:	p.G12A (c.35C>G)
TYPE:	Missense
COSMIC #:	---
MUTATION%:	29.81
FUNCTION:	Oncogene, Activation (turns off GTPase function) - constitutive activation of MAP kinase pathwa
DIAGNOSIS:	Not a previously reported somatic mutation
PROGNOSIS:	
THERAPY:	No "FDA approved" therapy targeting this mutation for this tumor type at this time.
GENE #2:	GNAS (GNAS complex locus)
MUTATION:	p.R202H (c.605G>A)
TYPE:	Missense
COSMIC #:	COSM27895

Figure 10. Excerpt of Meditech Report Worksheet. This figure depicts a portion of the Meditech Report page with a compiled report, including the result section and part of the interpretation. Additionally, the user button “Copy Report” can be seen in the upper right-hand corner.

Supercontrol Data: These worksheets include 1% Supercontrol Allele Coverage, 1% Supercontrol NG, 1% Supercontrol VC, Supercontrol Results (see Figure 11), and Supercontrol Myeloid Results. The interface is similar to that of the specimen mutation reports and includes tabs to import data as well as a summary report for the laboratory database.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Gene ID	Region Name	Position		Mutation	COSMIC	Detected VC?	Detected NG?	Avg Frequency	Avg Coverage	Result	Comment
2	KRAS	CHP2_KRAS	25398285	c.34G>TG	p.G12C	COSM516						
3	BRAF	CHP2_BRAF	140453136	c.1799T>TA	p.V600E	COSM476						
4	PIK3CA	CHP2_PIK3CA	178952085	c.3140A>AG	p.H1047R	COSM775						
5	EGFR	CHP2_EGFR_EX21	55259515	c.2573T>GT	p.L858R	COSM6224						
6	EGFR	CHP2_EGFR_20	55249071	c.2369C>CT	p.T790M	COSM6240						
7	EGFR	CHP2_EGFR_EX19	55242466	c.2236_2250delGAATT	c.748-750delG	COSM6225						
8	NRAS	CHP2_NRAS_2	115258744	c.38G>GA	p.G13D	COSM573						
9	KIT	CHP2_KIT	55593662	c.1728-1730-delTCC	c.1728-1730-delTCC							
10	NPM1	CHP2_NPM1	170837548	c.863_864insTCTG	c.863insTCTG	COSM17559						
11												
12												
13		Lot #			GENERATE SUPERCONTROL RESULTS							Yes?
14		PGM Run #							Has supercontrol been entered into database?			
15		Analysis By	Rebecca						Have Qc run metrics entered into database?			
16												
17												
18												

Figure 11. Excerpt of Supercontrol Results Worksheet. This figure depicts a portion of the blank Supercontrol Results worksheet, including the resulting table with all of the mutations to check for in the raw data, as well as the user button "Generate Supercontrol Results". After this script is run, the table will be accordingly populated and a database entry line will be generated below (off-screen).

QM Data: This worksheet includes rows of data to be pasted into the Quality Metrics database with mutation and run information from the current specimen.

Meditech Databases: These include all databases necessary for compiling the Meditech report, including gene names, general function, specific function, diagnosis (see Figure 12), prognosis, therapy, and resources. All databases are fully editable and expandable by the user. Spacing of text in the database is specific to the character limitations of the Meditech reports.

	A	B	C	D	G	H	I	J	K	L	M	N
1	Gene	COSMIC_id	CDS_mut_syntax	AA_mut_synt	Known Somatic?	Meditech Text						
2	ABL1	COSM12560	c.944C>T	p.T315I	NO	Not a previously reported somatic mutation						
3	ABL1	COSM12573	c.763G>A	p.E255K	NO	Not a previously reported somatic mutation						
4	ABL1	COSM12574	c.764A>T	p.E255V	NO	Not a previously reported somatic mutation						
5	ABL1	COSM12575	c.951C>G	p.F317L	YES	Confirmed "hotspot" somatic mutation found in					a variety of tumors	
6	ABL1	COSM12576	c.757T>C	p.Y253H	NO	Not a previously reported somatic mutation						
7	ABL1	COSM12577	c.749G>A	p.G250E	NO	Not a previously reported somatic mutation						
8	ABL1	COSM12578	c.1052T>C	p.M351T	NO	Not a previously reported somatic mutation						
9	ABL1	COSM12602	c.827A>G	p.D276G	NO	Not a previously reported somatic mutation						
10	ABL1	COSM12604	c.1187A>G	p.H396R	NO	Not a previously reported somatic mutation						
11	ABL1	COSM12605	c.1075T>G	p.F359V	NO	Not a previously reported somatic mutation						
12	ABL1	COSM12608	c.730A>G	p.M244V	NO	Not a previously reported somatic mutation						
13	ABL1	COSM12609	c.756G>C	p.Q252H	NO	Not a previously reported somatic mutation						
14	ABL1	COSM12610	c.758A>T	p.Y253F	NO	Not a previously reported somatic mutation						
15	ABL1	COSM12611	c.1064A>G	p.E355G	NO	Not a previously reported somatic mutation						
16	ABL1	COSM12631	c.742C>G	p.L248V	NO	Not a previously reported somatic mutation						
17	ABL1	COSM12632	c.756G>T	p.Q252H	NO	Not a previously reported somatic mutation						
18	ABL1	COSM131574	c.1159T>A	p.L387M	NO	Not a previously reported somatic mutation						
19	ABL1	COSM49071	c.1150C>A	p.L384M	NO	Not a previously reported somatic mutation						
20	ABL1	COSM49074	c.949T>C	p.F317L	NO	Not a previously reported somatic mutation						
21	AKT1	COSM33765	c.49G>A	p.E17K	YES	Confirmed "hotspot" somatic mutation found in					a variety of tumors	

Figure 12. Excerpt of Gene Diagnosis Database. As seen in the figure, the diagnosis database is organized by gene and COSMIC ID and also includes pertinent information such as coding sequence and amino acid mutations. The text that will be incorporated into a Meditech report is seen in the far right column, with proper spacing for the report. This text can be edited by the user simply by clicking the cell. Additionally, entries can be added.

User Options: This includes both Demographic Options, which contains all of the text used in the drop-downs on the Demographics tab (see Figure 13), and Tumor ROI Options, which contains the list of genes by tumor type that are used for several filters in the macro (see Figure 14). Both worksheets are intended to be fully customizable by the user, so that specific text options can be changed as well as more complex technical aspects such as ROI genes.

	A	B	C	D	E	F	G	H
1	**Patient Sex**	**PGM Instrument**	**Baseline?**	**MRD?**	**Transplant Status**	**Analysis By**	**Tumor %**	**Barcode**
2	Male	PGM-1	Yes	Yes	Pre	Keith	≤ 5	IonXpress_001
3	Female	PGM-2	No	No	Post	Rebecca	6-10	IonXpress_002
4						XM	11-15	IonXpress_003
5							16-20	IonXpress_004
6							21-25	IonXpress_005
7							26-30	IonXpress_006

Figure 13. Excerpt of Demographic Options Worksheet. All of the drop down lists in the Demographics Worksheet reference the text on this page. As seen in the figure, the field is listed across the top in bold, and the drop-down text options are listed below. This allows text to be edited by the user, and also for options to be added (such as the "Analysis By" field, which lists all of the lab technicians that may run the analysis.)

	A	B	C	D	E	F	G	H	I	J
1	**Tumor Type**	**ROI Genes**								
2	Colon Cancer (CRC)	BRAF	KRAS	NRAS	PIK3CA					
3	Breast Cancer									
4	Lung Cancer	BRAF	EGFR	ERBB2	KRAS	TP53				
5	AML	FLT3-TKI	IDH1	IDH2	JAK2	KIT	KRAS	NPM1	NRAS	TP53
6	Pancreatic CA & Cyst Fluid	BRAF	CTNNB1	GNAS	KRAS	PIK3CA	VHL			
7	Malignant Melanoma	BRAF	HRAS	KIT	NRAS	TP53				
8	Thyroid Cancer	BRAF	GNAS	HRAS	KRAS	NRAS	PIK3CA	RET		
9	GIST	KIT	PDGFRA							
10	Other									

Figure 14. Excerpt of Tumor ROI Options Worksheet. This database serves two functions: 1.) The "Tumor Type" column is referenced on the Demographics drop-down field. Thus, text can be edited and tumor types can be added which will also appear on the Demographics page. 2.) The ROI genes for a given tumor type can be added and deleted, which affects a number of processes for compiling the Meditech report (namely, the Results section).

Analysis Databases: This includes the ROI database (Figure 15), SNP database (Figure 16), and Artifact database. Each was compiled from databases currently used for analysis by the lab. The ROI database specifies which positions to include for a given tumor type when developing the Pertinent Negatives table (see Chapter 2.3.4). The SNP and Artifact databases specify which variants to filter in the raw data. All databases are editable and expandable.

	A	B	C	D	E	F	H	I	J	K	L	M	N	O
1	Gene	Chromosome location Coverage of ROI	Exon	Codons covered	ROI manully reviewed (depth) x	Wildtype or Mutant?	Colon Cancer (CRC)	Breast Cancer	Lung Cancer	AML	Pancreatic CA & Cyst Fluid	Malignant Melanoma	Thyroid Cancer	GIST
41	JAK2	9,5073770	14	603-622						x				
42	KIT	4,55224546	8	ommited	ommited	ommited						x		
43	KIT	4,55561763	2	23-60						x		x		x
44	KIT	4,55589750	8	ommited	ommited	ommited				x				x
45	KIT	4,55592186	9	494-509						x		x		x
46	KIT	4,55593418	10	523-549								x		
47	KIT	4,55593431	10	523-549						x				x
48	KIT	4,55593576	11	553-592								x		
49	KIT	4,55593590	11	553-592						x				x
50	KIT	4,55594221	13	627-664						x		x		x
51	KIT	4,55595497	14	664-694								x		
52	KIT	4,55595519	14	664-694						x				x
53	KIT	4,55597495	15	715-732						x		x		x
54	KIT	4,55599281	17	795-828								x		
55	KIT	4,55599320	17	795-828						x				x
56	KIT	4,55602674	18	829-865								x		
57	KIT	4,55602694	18	829-865						x				x
58	KRAS	12,25378647	4	114-150			x		x		x	x	x	x
59	KRAS	12,25380277	3	39-66			x		x	x	x	x	x	x
60	KRAS	12,25398225	2	05-37			x			x	x	x	x	x
61	KRAS	12,25398255	2	05-37					x					
62	MPL	1,43815040	10	502-522						x				

Figure 15. Excerpt of ROI Database. Each row includes the details of a different ROI position, and the columns list the different tumor types. Pertinent genes are marked off with an "x" below the given tumor type.

	A	B	C	D	E	F	G	H	I	J
1	Chromosme Position	Chromosome	Gene	Coding Change	Amino Acid Change	COSMIC	Type	Average Variant Frequency	dbSNP	Confirmed in Literature
2	534242	11	HRAS	c.81A>AG	27H>HH	COSM249860	SNP-SILENT	47.4	rs12628	YES
3	1220321	19	STK11	na	na		INTRON	48.2	na	na
4	1220573	19	STK11				INTRON	51.8		
5	1221293	19	STK11	c.816C>CT	272Y>YY	COSM29005	SNP-SILENT	~10%	rs9282859	YES
6	1807894	4	FGFR3	c.1953G>A	651T>T		SNP-SILENT	50	rs7688609	YES
7	1807922	4	FGFR3	na	na		INTRON	52.9	na	na
8	7578210	17	TP53				SNP-SILENT	51.2		
9	7579472	17	TP53	c.215G>CG	72P>RP	COSM45985	SNP-MISSENSE		rs1042522	YES
10	10191469	3	VHL				INTRON	25.9		
11	17945640	19	JAK3				INTRON	48.6		

Figure 16. Excerpt of SNP Database. The figure depicts the various fields that are included in the SNP database. A similar concept is also used for the Artifacts Database. The tables are completely customizable and expandable. The only required field is Chromosome Position, which is used when filtering the raw data.

2.3.2 Data Importation and Setup

To begin, all fields of the Demographics tab are completed by the user and the Allele Coverage, Nextgene, Coverage Analysis, and Sample ID data are imported into the program. This step is entirely manual and requires no macros to complete.

2.3.3 Preliminary Review

The preliminary review step consists of all data processing that occurs prior to manual mutation review by the laboratory technician. Namely, this includes preparing, formatting, and filtering the raw data imported from Variant Caller and Nextgene. The categories and individual macro functions are outlined in Figure 17. These are all of the processes that are carried out upon clicking the “Pre-Review” button.

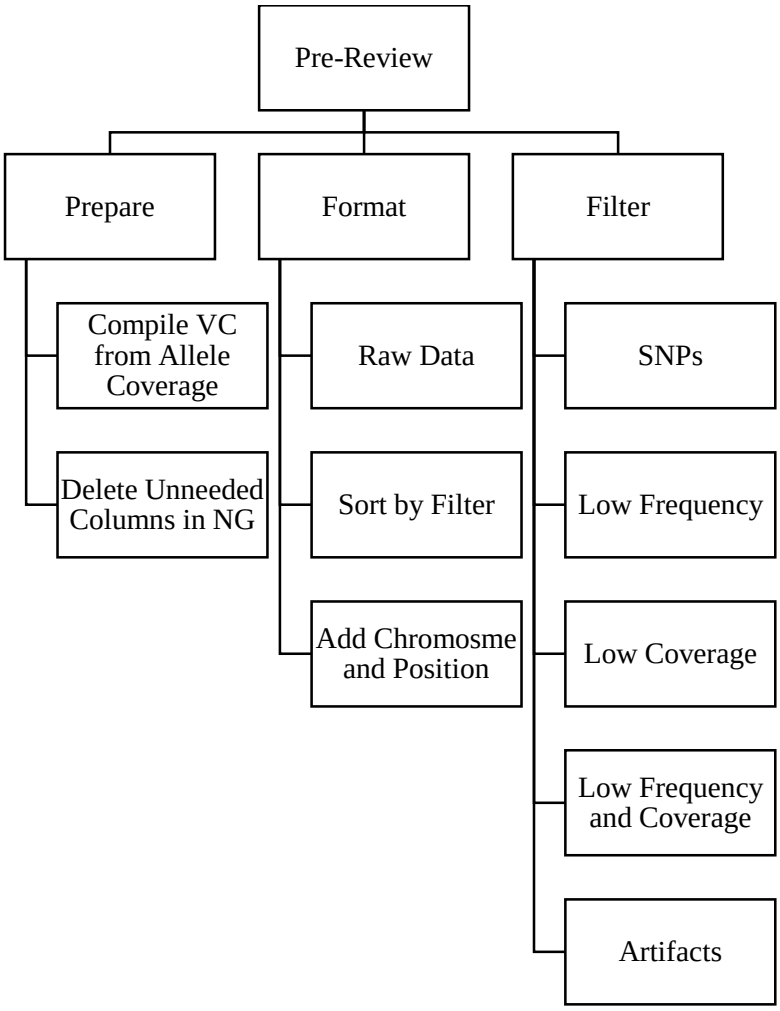


Figure 17. Summary of Pre-Review Script. The figure provides a visual breakdown of the Pre-Review script into three primary categories (Prepare, Format, and Filter) and the individual macros that fall into each category.

The preparation step includes compiling the Variant Caller data from the Allele Coverage and deleting unneeded columns in the Nextgene data. The Variant Caller report is compiled by searching all lines of the Allele Coverage data for mutations that are specified as “Heterozygous” or “Homozygous” under the “Allele Call” column. All other variants will either be listed as “Absent” or “No Call” and thus are not included in the report. In regards to the Nextgene data, the raw data file includes several columns that are not needed for data analysis. For this reason, they are immediately deleted in order to consolidate the report.

The format group includes all processes involved with the visual formatting of the report. The raw data is formatted by removing excess text at the top of the mutation report and adding a "Review" column for comments that will later be added by the technician. Additionally, a column is added that concatenates the chromosome and position numbers of each mutation in the form "Chromosome, Position" which aids with the manual review process. Finally, the "sort by filter" step occurs after filter the raw data and ultimately organizes the variants by their given filter and brings unfiltered data to the top of the report.

Lastly, the filter category of macros includes all filters that are applied to the raw data (see Table 1 for parameter details). The SNP and artifact filters compare the position of each variant in the report to the positions in the respective databases. Conversely, the coverage and frequency filters compare values in the data to specified values (frequency is determined by tumor percentage; coverage less than 500 is filtered). The code in Figure 18 is an excerpt from the "Pre-Review" macro and depicts the order in which each individual function is referenced for the Variant Caller analysis (repeated for Nextgene).

Table 1. Pre-Review Filtering Parameters

SNP-MISSENSE	Missense SNP (change in amino acid)
SNP-SILENT	Silent SNP (no change in amino acid)
SNP-INTRON	SNP in Intron
ARTIFACT	Artifact (Homopolymer, Deletion, Strand Bias)
<MUT FREQ THRESH	Frequency of variant is less than the given threshold
<500X COV	Coverage of variant is less than 500x
LOW FREQ AND COV	Frequency and coverage of variant are less than the given thresholds

```
'The following allows the template to be saved as any file name without interrupting function of macros
Dim strFileName As String
    'Define a string
strFileName = ActiveWorkbook.Name
    'Set string to name of the active workbook (the open analysis template file)



'MACROS

'Allele Coverage
Sheets("Allele Coverage").Select
Application.Run "'" & strFileName & "'" & "!pre_prepare_vc"
Range("A1").Select

'Variant Caller
Sheets("Variant Caller").Select
Application.Run "'" & strFileName & "'" & "!pre_format_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_snps_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_mutfreq_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_lowcov_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_mutandcov_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_artifacts_vc"
Application.Run "'" & strFileName & "'" & "!pre_format_sortbyfilter"
Application.Run "'" & strFileName & "'" & "!pre_format_chrpos_vc"
Range("A1").Select
```

Figure 18. Excerpt of Pre-Review Code. The figure above depicts a portion of the code that is run upon clicking the “Pre-Review” button. Each application is run sequentially for both Variant Caller and Nextgene (not shown).

2.3.4 Post Review

The post review step consists of all processing that occurs after the technician manually review the mutations and determines which are true. This step is divided into two main categories: Final Report Details and Final Report Summary. These categories as well as the specific functions in each are depicted below in Figure 19.

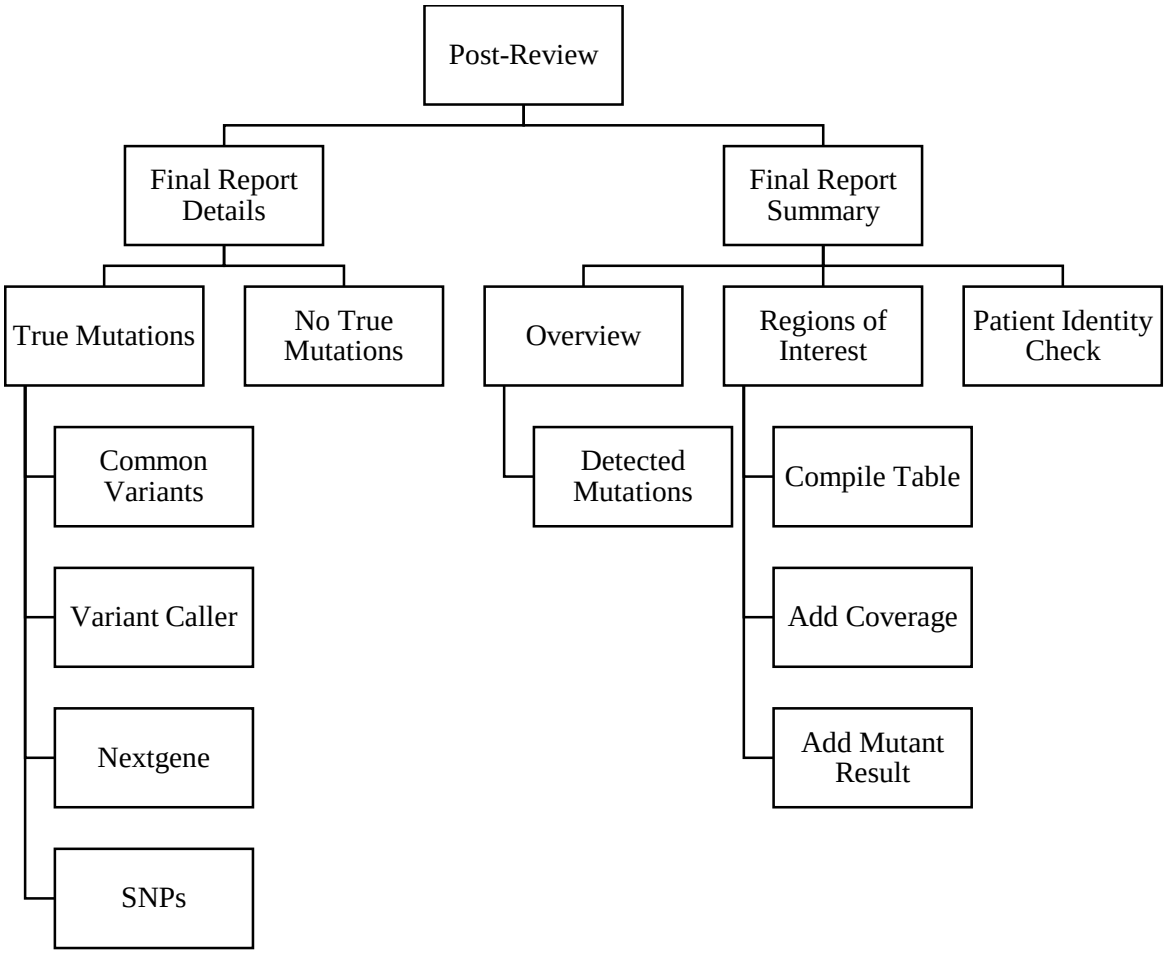


Figure 19. Summary of Post-Review Script. The figure provides a visual breakdown of the Post-Review script into two primary categories (Final Report Details and Final Report Summary) and the individual macros that fall into each category.

The Final Report Details category consists of determining whether a mutation in the raw data was marked as "TRUE" or "FALSE" by the technician, and adding it to the detailed report accordingly. As seen in the diagram above, only those variants marked as "TRUE" are included in the Final Report. The program first determines whether the variant is found in both Variant Caller and Nextgene ("Common Finding") by comparing the positions in each report and checking for a match. If the positions match, the program then checks if the variant is marked as a true mutation in both programs. If so, relevant information from the line of data in Nextgene and Variant caller is transferred to a single row in the Final Report Details worksheet. For

example, the “Frequency VC” field would be populated with the mutation frequency reported by Variant Caller, whereas the “Frequency NG” field would be populated with the mutation frequency reported by Nextgene. This allows the technician to have two frames of reference when reviewing the sample and ensures that both programs are producing consistent results. After compiling the Common Findings section of the report, the program goes on to check for “TRUE” mutations that are present in only the Variant Caller or Nextgene data. If found, the information is transferred to a new row in the detailed report. Finally, SNPs are added from the Nextgene report.

The Final Report Summary serves as a more concise report for the technician. The regions at the top of the report are autofilled from the Demographics tab using a formula (and thus is not included in the macro). The three main functions of the program for this worksheet are generating the mutation overview, regions of interest table, and patient identity check. The mutation overview is generated by searching the Final Report Details page for mutations in the Common Findings, Variant Caller, and Nextgene sections. If mutations are present, the gene name is added to a string along with the amino acid mutation in parenthesis. This string is reported in the respective section of the summary page.

The ROI table, labeled “Pertinent Negatives” on the summary page, is a table of genes that are relevant for the given tumor type and must be manually reviewed by the technician to ensure that no mutations exist. The table is compiled from the ROI database by finding the column of the database that corresponds to the tumor on the Demographics worksheet and adding all positions that are checked off in the table for that column. These positions include details such as gene name, chromosome number, position, exon, and other information that is pertinent for review. Additionally, the program adds the coverage in the “ROI manually

reviewed (depth) x" column from the Allele Coverage data. Figure 20 shows an example table compiled for colon cancer.

23	Gene	Chromosome location Coverage of ROI	Exon	Codons covered	ROI manully reviewed (depth) x	Wildtype or Mutant?
24	BRAF	7,140453136	15	581-611	2810	
25	BRAF	7,140481450	11	439-473	3339	
26	CTNNB1	3,41266130	3	40-46	2406	
27	KRAS	12,25378647	4	114-150	2789	
28	KRAS	12,25380277	3	39-66	5407	
29	KRAS	12,25398225	2	05-37	4458	
30	NRAS	1,115252220	4	124-150	4775	
31	NRAS	1,115256529	3	43-70	3738	
32	NRAS	1,115258746	2	03-31	3919	
33	PIK3CA	3,178936080	7	522-549	2223	
34	PIK3CA	3,178938850	8	677-720	2846	
35	PIK3CA	3,178947850	9	898-924	6153	
36	PIK3CA	3,178952050	10	1018-1051	2738	

Figure 20. Sample Colon ROI Table. The figure depicts a sample table that would be complied for the Final Report Summary page from the ROI database.

Finally, the patient identity check ensures that all data files and information in the program belong to the correct sample. This test checks the barcode, MD number, and patient sex identified in the Demographics worksheet against all imported data files to ensure that there is a match. If there is a discrepancy between any of the values, an error message is generated at the bottom of the summary report to alert the user. If all identity metrics pass, a note is also generated. Figure 21 summarizes all of the scripts that have been discussed with an excerpt from the Post-Review code.

```
'Populate Final Report Details worksheet
Sheets("Final Report Details").Select
Application.Run "'" & strFileName & "'" & "!post_det_find_common"
Application.Run "'" & strFileName & "'" & "!post_det_find_vc"
Application.Run "'" & strFileName & "'" & "!post_det_find_ng"
Application.Run "'" & strFileName & "'" & "!post_det_find_snps"
Application.Run "'" & strFileName & "'" & "!post_det_no_mutations"



'Populate Final Report Summary worksheet
Sheets("Final Report Summary").Select
'Summary of detected mutations
Application.Run "'" & strFileName & "'" & "!post_sum_mutations"

'ROI Table
Application.Run "'" & strFileName & "'" & "!post_sum_roi_table"
Application.Run "'" & strFileName & "'" & "!post_sum_roi_coverage"

'Patient Identity Check
Application.Run "'" & strFileName & "'" & "!post_sum_pat_ide_check"
```

Figure 21. Summary of Post-Review Code. The figure above is an excerpt from the Post-Review script to show the macros that are triggered upon clicking the button. Each application is run sequentially, beginning with the Details page and ending with the Summary report.

2.3.5 Meditech Report

The Meditech step of the program includes all processes that occur after the post review step in order to generate a complete Meditech Diagnosis report and Raw Data report. As seen in Figure 22, this script is divided into three general categories: diagnosis report programs, raw data report programs, and additional user features.

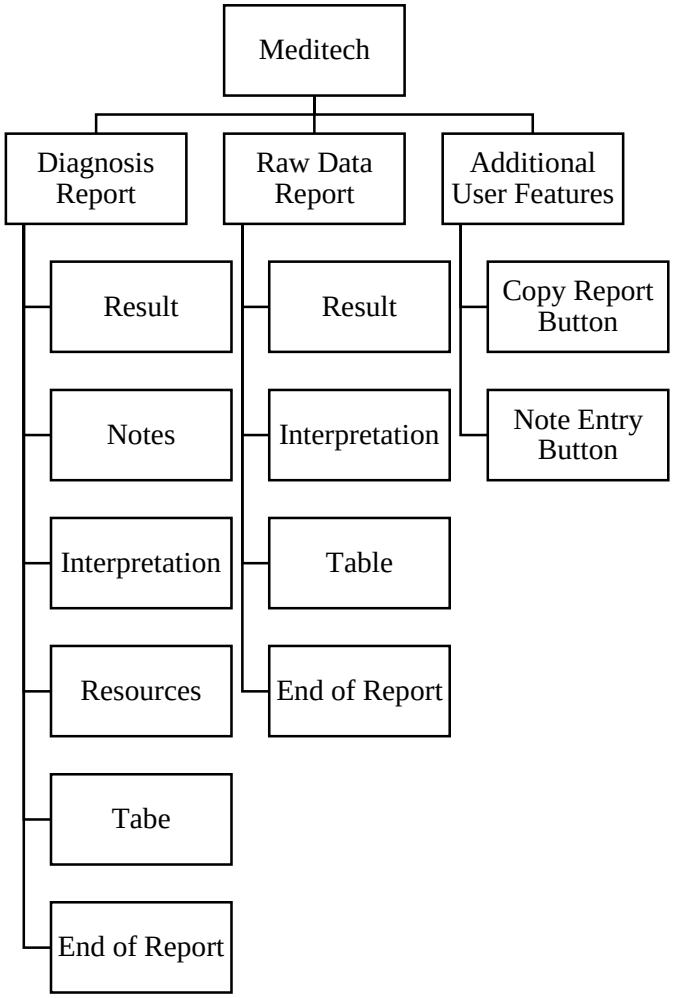


Figure 22. Summary of Meditech Script. The figure provides a visual breakdown of the Meditech report script into two primary categories (Report Generation and User Functions) and the individual macros that fall into each category.

The diagnosis and raw data scripts include all of those that are run sequentially to compile the final Meditech reports. The Result script compares the mutations in the Final Report Details worksheet to the genes in the Tumor ROI database in order to determine whether the sample should be classified as positive or negative (see Table 2). A string is developed with a list of genes for each possible result. These results are then presented concisely at the top of the report.

Table 2. Possible Diagnosis Report Results

RESULT	**DESCRIPTION**
POSITIVE	Mutation was found that matches *at least one* of the ROI genes for the given tumor type.
NEGATIVE	No mutations were found that match any of the ROI genes for the given tumor type.
OTHER FINDINGS	Mutation was found that does not match ROI genes but will be reported for the specimen.

The Notes script searches a database developed for the program (see Figure 23) that lists notes by tumor, gene, and mutation. The specificity of the notes vary, and thus the script must if-then statements in order to include the most accurate note for the specimen. For example, if a note is present in the database for the given tumor type and gene, but not the specific mutation, the script will include this note in the Meditech report. In order to accomplish this, the script searches the database first by tumor, then by mutations present in the sample, and finally by specific mutation as well as other circumstances (low coverage note, genetic counseling note, etc.) These notes are included directly below the results section of the report.

	B	C	D	E	
1	**Tumor**	**Description**	**Gene**	**Mutation**	**Note Text**
2	AML	Baseline Testing			NOTE: Mutation studies were performed on this specimen to serve as a ba
3	AML	Baseline Testing Cancellation			NOTE: Mutation studies were performed on this specimen to serve as a ba
4	All	Low Coverage Note 1			NOTE: 158 of 205 (77%) of the PCR amplicons had <500X sequencing covera
5	All	Low Coverage Note 2			NOTE: 40 of 205 (19.5%) of the PCR amplicons had <500X sequencing cover
6	All	Low Coverage Note 3			NOTE: In addition, 24% of the gene sequences in the AmpliSeq Cancer Hot
7	All	Genetic Counseling			NOTE: CONSULTATION WITH A GENETICS COUNSELOR IS RECOMMENDED
8	All	Single Mutation	APC	p.E1317Q	The APC p.E1317Q (c.3949G>C) variant has been reported as a low frequen
9	All	Single Mutation	ATM	p.F858L	The ATM p.F858L (c.2572T>C) variant has been reported as a low frequency
10	All	Single Mutation	MET	p.T992I	The MET p.T992I/p.T1010I (c.2975C>T/c.3029C>T) variant has been reportec
11	All	Single Mutation	MET	p.T1010	The MET p.T992I/p.T1010I (c.2975C>T/c.3029C>T) variant has been reportec
12	Colon	Single Mutation	KRAS	p.G12V	NOTE: The KRAS mutation p.G12V (GGT>GTT) resulting in a substitution of
13	Colon	Single Mutation	KRAS	p.G12D	NOTE: The KRAS mutation p.G12D (GGT>GAT) resulting in a substitution of
14	Colon	Single Mutation	KRAS	p.G12A	NOTE: The KRAS mutation p.G12A (GGT>GCT) resulting in a substitution of
15	Colon	Single Mutation	KRAS	p.G12C	NOTE: The KRAS mutation p.G12C (GGT>TGT) resulting in a substitution of
16	Colon	Single Mutation	KRAS	p.G12R	NOTE: The KRAS mutation p.G12R (GGT>CGT) resulting in a substitution of
17	Colon	Single Mutation	KRAS	p.G12S	NOTE: The KRAS mutation p.G12S (GGT>AGT) resulting in a substitution of
18	Colon	Single Mutation	KRAS	p.G12F	NOTE: The KRAS mutation p.G12F (GGT>TTT) resulting in a substitution of G
19	Colon	Single Mutation	KRAS	p.G13D	NOTE: The KRAS mutation p.G13D (GGC>GAC) resulting in a substitution of

Figure 23. Excerpt of Meditech Notes Database. All notes are organized by tumor type, gene, and mutation. Most are also given a brief description (if applicable) to make the database more easily navigable for the user. The text used in the Meditech report is found in the far right column under "Note Text." The database is fully customizable and expandable.

The interpretation section of the report is designated to the specific mutations detected in the sample and details of the diagnosis and treatment associated with each. This script searches the Final Report Details worksheet for mutations (in all sections, "Common Findings", "Nextgene", and "Variant Caller") and the details of mutation such as gene, mutation call, amino acid change, and COSMIC ID number. The script compiles the report one mutation block at a time by assigning all of these attributes to variables for the given mutation and using the variables to print the report text. Additionally, these attributes (namely, amino acid change and COSMIC ID) are used to search the Meditech databases (see section 2.2.1 User Interface) for information such as gene function, diagnosis, prognosis, and therapy. These scripts operate in a manner similar to the Notes script, by searching the database for the most specific information that can be included for the given sample. For example, if treatment information is available for only the gene and not the specific mutation, it will be included in the Meditech report.

The Resources script then searches a single database for a block of information related to the tumor type. Currently, there is text present for all of the tumor types that are available in the program. However, if tumor types were to be added in the future, the resources database would potentially have to be updated to include new information for the report.

The Table script compiles a table summarizing the findings (which is abbreviated in the diagnosis report and detailed in the raw data report). This table is specific to the cancer type, and only includes genes that are of interest for the particular tumor, as well as any other mutations for which the sample is positive.

Finally, the End-of-Report script adds all of the information found at the bottom of the report. This includes Table 2, references, and the disclaimer. The program references a database which contains the end-of-report data based on tumor type.

2.3.6 Supercontrol Analysis

Supercontrol analysis is only completed once per run (not for each sample) and ensures that mutations are being detected correctly by the Variant Caller and Nextgene software. In order to streamline this test, a separate portion of the program was designed to process Supercontrol data and generate a quality control database entry. This program operates on the same principles as the “Pre-Review” macro, but is specialized to the needs of this analysis as seen in the function breakdown in Figure 24.

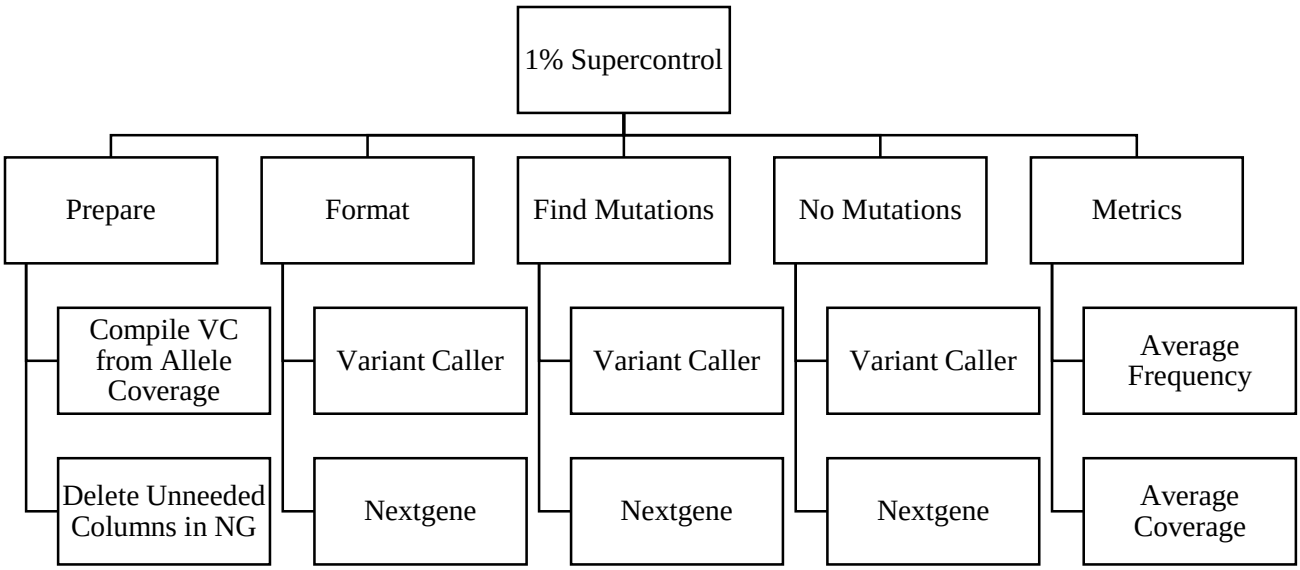


Figure 24. Summary of Supercontrol Script. The figure provides a visual breakdown of the Supercontrol script into five primary categories (Prepare, Format, Find Mutations, No Mutations, and Metrics) and the individual macros that fall into each category. These scripts mimic those used for mutation analysis in the Pre-Review script.

After the user imports the Allele Coverage and Variant Caller data into the respective Supercontrol worksheets, the data will be prepared and formatted in the same way as the Pre-Review step for mutation analysis. Once this step is complete, the positions in the Nextgene and Variant Caller reports are compared against at a predefined list of mutations on the supercontrol results worksheet. Using the sample principle as filtering the SNPs and artifacts based on database entries, the mutations are found in the data and highlighted by comparing position number. Once finding a mutation in the data, the frequency and coverage are also added to the Supercontrol report (averaged between Variant Caller and Nextgene if found in both). Once the table is complete (see Figure 25), the database entry line is populated using a series of formulas that pull information from both the Demographics worksheet and the Supercontrol table. This row can then be copy and pasted in the QC database in order to track PGM run metrics over time.

	A	B	C	D	E	F	G	H	I	J	K
1	Gene ID	Region Name	Position		Mutation	COSMIC	Detected VC?	Detected NG?	Avg Frequency	Avg Coverage	Result
2	KRAS	CHP2_KRAS	25398285	c.34G>TG	p.G12C	COSM516	YES	YES	3	2176	Pass
3	BRAF	CHP2_BRAF	140453136	c.1799T>TA	p.V600E	COSM476	YES	YES	7.64	1305	Pass
4	PIK3CA	CHP2_PIK3CA	178952085	c.3140A>AG	p.H1047R	COSM775	YES	YES	5.66	1394	Pass
5	EGFR	CHP2_EGFR_EX21	55259515	c.2573T>GT	p.L858R	COSM6224	YES	YES	7.85	2511	Pass
6	EGFR	CHP2_EGFR_20	55249071	c.2369C>CT	p.T790M	COSM6240	YES	YES	8.86	1022	Pass
7	EGFR	CHP2_EGFR_EX19	55242466	c.2236_2250delGAATT	c.748-750delG	COSM6225	No	YES	6.49	2532	Review
8	NRAS	CHP2_NRAS_2	115258744	c.38G>GA	p.G13D	COSM573	YES	YES	3.16	1690	Pass
9	KIT	CHP2_KIT	55593662	c.1728-1730-delTCC	c.1728-1730-delTCC		No	No	7.95	3220	Flag
10	NPM1	CHP2_NPM1	170837548	c.863_864insTCTG	c.863insTCTG	COSM17559	YES	YES	5.90	1567	Pass

Figure 25. Sample Supercontrol Result Table. The figure shows a completed results table from a Supercontrol data run. The table specifies whether the mutation was found in Variant Caller, Nextgene, or both as well as the average frequencies and coverage from the run. A result is produced using conditional formatting ("Pass" if found in both programs, "Review" if found in one program, "Flag" if not found in either).

3. Results

3.1 Testing

In order to test the accuracy of the program, several blinded case studies were performed using data from patients with a variety of tumor profiles. The raw data was imported into the program and the case was run to completion. The results obtained from the program were compared to those obtained by the manual method of analysis to ensure that the outputs were consistent.

To demonstrate the testing methods used, a sample case study will be discussed in detail in this chapter. The data was obtained from several anonymous melanoma patients with a single common mutation to ensure patient privacy.

3.1.1 Demographics

To begin, the details of the patient and PGM run were inputted into the demographics form. Again, data from different patients with similar tumor profiles were used in order to simulate the analysis that would be performed on a real patient. The tumor percentage from each patient was averaged, and a tumor percentage range from 31-40% was chosen, yielding a mutation frequency threshold of 5% (see Figure 26).

Tumor Information	
Tumor Type	Malignant Melanoma
% Tumor	31-40
Mutation Frequency Threshold	5

Figure 26. Excerpt of Completed Demographics Form. For the case study, malignant melanoma data with an average tumor percentage of 31-40% was chosen, yielding a mutation frequency threshold of 5%.

3.1.2 Review

The raw data files were pasted into the respective tabs of the program. After running the Pre-Review program, the results from Variant Caller and Nextgene were examined separately. All mutations except one from the Variant Caller data were filtered as SNPs. The remaining variant was a mutation of the BRAF gene on chromosome 7 at the position 140453136, highlighted below in Figure 27. This mutation was also detected by the Nextgene software, as seen in Figure 28.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1					Color Key:		Red: Less than mutation frequency threshold								
2	DMO Number	Analysis By			Green:SNP		Orange: Less than 500X coverage								
3	MD15-xxx	Rebecca			Blue:Artifact		Pink: Less than mutation frequency threshold and less than 500X coverage								
4															
5	Review	Chrom	Position	Chr,Position	Ref	Varian	Allele Call	Filter	Frequency	Quality	Filter	Type	Allele Sour	Allele Nam	Gene ID
6		chr7	140453136	7,140453136	A	T	Heterozygous	-	42.1	8867.89	-	SNP	Hotspot	COSM476	BRAF
7	SNP-SILENT	chr4	1807894	4,1807894	G	A	Homozygous	-	100	16388.3	-	SNP	Novel	---	FGFR3
8	SNP-SILENT	chr4	55141055	4,55141055	A	G	Homozygous	-	100	35657.1	-	SNP	Novel	---	PDGFRA
9	SNP-SILENT	chr5	112175770	5,112175770	G	A	Heterozygous	-	97.4	34230.3	-	SNP	Novel	---	APC
10	SNP-SILENT	chr7	55249063	7,55249063	G	A	Homozygous	-	100	25067.6	-	SNP	Novel	---	EGFR
11	SNP-SILENT	chr10	43613843	10,43613843	G	T	Homozygous	-	100	15613.5	-	SNP	Novel	---	RET
12	SNP-SILENT	chr10	43615633	10,43615633	C	G	Heterozygous	-	68.8	10084.4	-	SNP	Novel	---	RET
13	SNP-SILENT	chr11	534242	11,534242	A	G	Heterozygous	-	38.3	4674.15	-	SNP	Hotspot	COSM2498	HRAS
14	SNP-MISSENSE	chr4	55593464	4,55593464	A	C	Heterozygous	-	60.7	16411.5	-	SNP	Hotspot	COSM2802	KIT
15	SNP-MISSENSE	chr17	7579472	17,7579472	G	C	Heterozygous	-	52.4	8764.08	-	SNP	Novel	---	TP53
16	INTRON	chr4	55980239	4,55980239	C	T	Heterozygous	-	21.7	2006.82	-	SNP	Novel	---	KDR
17	INTRON	chr5	149433596	5,149433596	TG	GA	Heterozygous	-	54.8	4381.98	-	MNP	Novel	---	CSF1R
18	INTRON	chr13	28610183	13,28610183	A	G	Homozygous	-	100	31947.7	-	SNP	Novel	---	FLT3
19															
20															

Figure 27. Filtered Variant Caller Data. The highlighted row shows the single unfiltered BRAF mutation.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	IonXpress_006_R_2015_02_17_07_38_12_user_PG2-113-02-13-15-1_Output.pjt																				
2	DMO Number	Analysis By		Color Key	Red:Less than mutation frequency threshold																
3	MD15-xxx	Rebecca		Green:SNP	Orange:Less than 500X coverage																
4				Blue:Artifact	Pink:Less than mutation frequency threshold AND less than 500X coverage																
5	Review	Index	Reference	Chromosome	Gene	Chr	Chr,Position	Reference	Gene Nuc	Coverage	Score	A(#F,#R)	C(#F,#R)	G(#F,#R)	T(#F,#R)	Ins(#F,#R	Del(#F,#R	SNP db_x	Mutation	Mutant A	Amino Acid Change
6		1	4.32E+08	209113152	IDH1	2	2,209113152	G	C	3737	8.1	0;0	0;0	1994;1601	0;0	28;56	73;69		c.355_356	2.25;3.80	FS
7		5	6.64E+08	1807894	FGFR3	4	4,1807894	G	G	925	23.7	495;427	0;0	1;1	0;0	0;1	1;0	rs7688609	c.1959G>A	99.68	653T>T
8		12	7.15E+08	55595543	KIT	4	4,55595543	T	T	2239	12.8	0;0	0;1	0;0	1112;1099	88;92	26;1		c.2036_20	8.04;1.21	FS
9		13	7.15E+08	55595550	KIT	4	4,55595550	T	T	2235	9.6	0;1	0;2	0;0	1092;1070	131;154	42;28		c.2045_20	12.75;3.13	FS
10		15	9.61E+08	112175770	APC	5	5,112175770	G	G	2036	23.6	943;1037	1;2	35;18	0;0	4;3	0;0	rs41115	c.4479G>A	97.25	1493T>TT
11		17	1.02E+09	170837529	NPM1	5	5,170837529	A	A	851	4	305;463	0;0	0;1	34;0	0;0	48;0		IVS847-2_	5.64;4.00	Splice
12		18	1.02E+09	170837530	NPM1	5	5,170837530	G	G	852	4	0;1	0;0	187;464	78;0	18;0	122;0		IVS847-2_	2.11;14.32	Splice
13		23	1.37E+09	140453136	BRAF	7	7,140453136	A	T	1745	25.6	506;514	1;0	0;0	364;360	0;1	0;0	rs1219132	c.1799T>T	41.49	600V>VE
14		29	1.65E+09	133748408	ABL1	9	9,133748408	A	A	2146	9.7	1022;1084	0;0	2;0	0;0	51;45	17;21		c.1130_11	4.47;1.77	FS
15		37	1.74E+09	89624263	PTEN	10	10,89624263	A	A	1646	2.6	602;1015	0;0	1;0	0;0	18;15	4;24		c.40_41in	2.00;1.70	FS
16		40	1.79E+09	534242	HRAS	11	11,534242	A	T	1005	22	321;311	0;0	173;200	0;0	0;1	0;0	rs12628	c.81T>TC	37.11	27H>HH
17		41	1.9E+09	108172409	ATM	11	11,108172409	A	A	2696	5.8	1517;1131	0;0	2;0	0;1	32;39	34;11		c.5216_52	2.63;1.67	FS
18		44	2.07E+09	28602330	FLT3	13	13,28602330	C	G	2773	11	0;0	1239;1391	0;0	0;0	20;25	69;74		c.2038_20	1.62;5.16	FS
19		51	2.41E+09	7577115	TP53	17	17,7577115	A	T	1116	13.9	570;541	0;0	0;1	0;0	32;29	1;3		c.823_824	5.47	FS
20		54	2.44E+09	37880235	ERBB2	17	17,37880235	C	C	3157	9.1	1;0	1437;1648	0;0	0;1	17;17	36;34		c.2283_22	1.08;2.22	FS
21		55	2.56E+09	1221314	STK11	19	19,1221314	C	C	1483	7.3	0;0	403;707	0;1	0;0	5;5	203;169	rs6787300	c.842delC	25.08	FS
22	SNP-SILENT	8	7.14E+08	55141055	PDGFRA	4	4,55141055	A	A	1978	26.1	1;1	0;0	1087;886	0;1	2;0	1;1	rs1873778	c.1701A>G	99.75	567P>P
23	SNP-SILENT	19	1.29E+09	55249063	EGFR	7	7,55249063	G	G	1481	23.9	682;796	0;0	0;2	0;0	0;0	0;1	rs1050171	c.2361G>A	99.8	787Q>Q
24	SNP-SILENT	34	1.7E+09	43613843	RET	10	10,43613843	G	G	905	21.9	0;0	0;0	0;0	462;440	0;0	3;0	rs1800861	c.2307G>T	99.67	769L>L
25	SNP-SILENT	36	1.7E+09	43615633	RET	10	10,43615633	C	C	1143	23.4	1;0	165;192	381;404	0;0	0;3	0;0	rs1800863	c.2712C>G	68.68	904S>SS

Figure 28. Filtered Nextgene Data. Rows 6 through 21 show the unfiltered variants which have been sorted to the top. The BRAF mutation that was also detected by Variant Caller is highlighted.

Upon examining this position in the Nextgene Browser, it was clear that the mutation was, in fact, true (see Figure 29) and thus was marked as “TRUE” in the review column of the program. The other unfiltered positions detected by Nextgene were manually reviewed in the browser, and were all determined to be false.

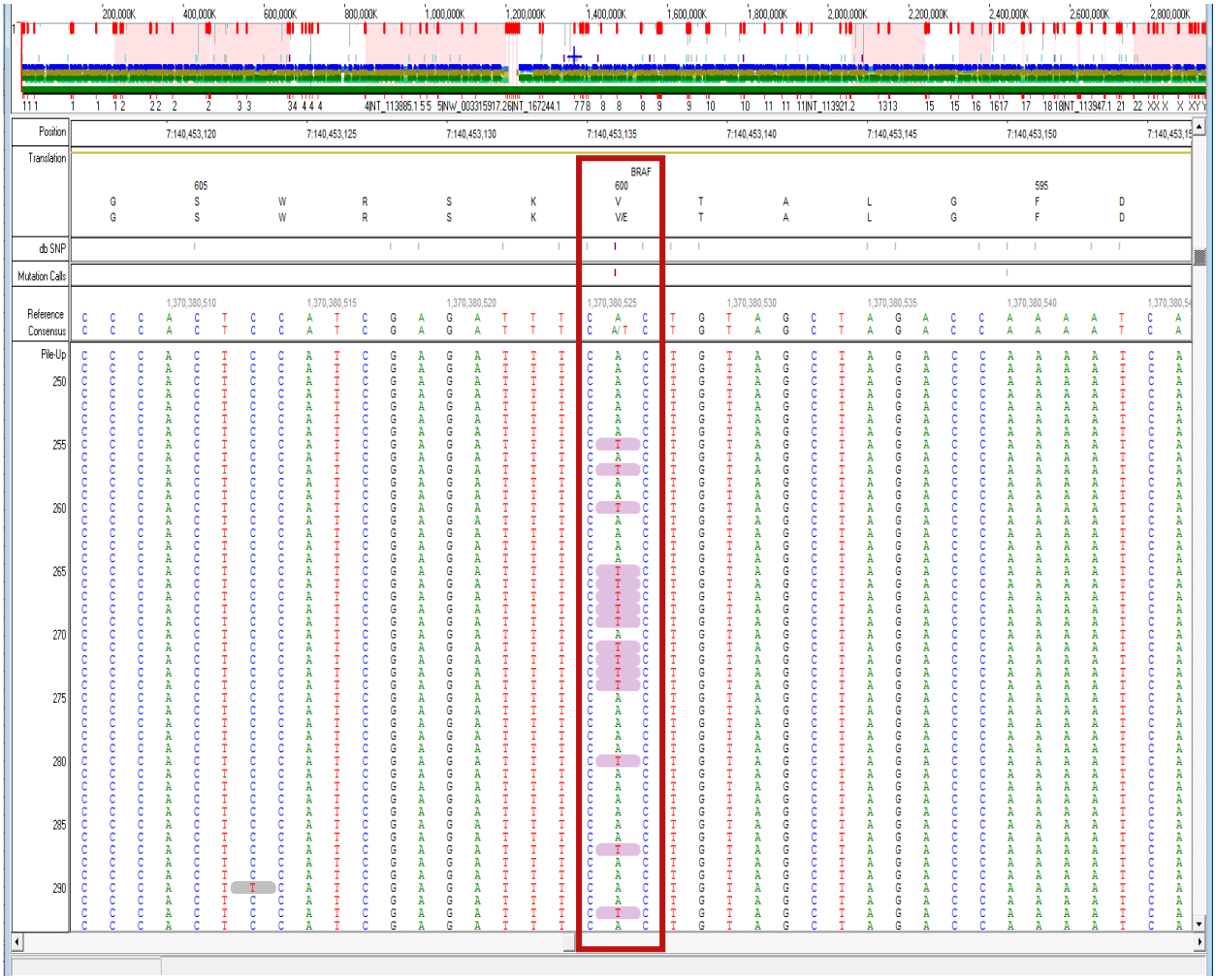


Figure 29. Screenshot of BRAF V600E Mutation in Nextgene Browser. The mutation from A to T (seen highlighted in purple) yields a missense mutation of V to E at codon 600 of the BRAF gene (outlined in red). The sequence around the mutation is relatively clean, as opposed to the single clearly-mutated base.

Upon running the Post-Review program, the Final Report Summary page contained the detected mutation summary (see Figure 30) and the melanoma-specific ROI table.

Detected Mutations	
Common Findings:	BRAF (600V>VE) mutation was detected by Variant Caller and Nextgene.
Variant Caller:	See common findings.
Nextgene:	See common findings.
*See next page for detailed mutation report.	

Figure 30. Detected Mutation Summary. The BRAF mutation (marked as "TRUE" in both Variant Caller and Nextgene) is categorized as a common finding. The amino acid mutation and codon number (600V>VE) is listed parentheses after the mutation.

Each position in the ROI table was manually reviewed in Nextgene Browser. Figure 31 depicts an example of a wildtype region (KRAS, codons 5-37), and Figure 32 depicts an example of a mutant region (KIT, codons 523-549). For the case study data, all positions except the BRAF mutation detected by the software were determined to be wildtype (see Figure 34).

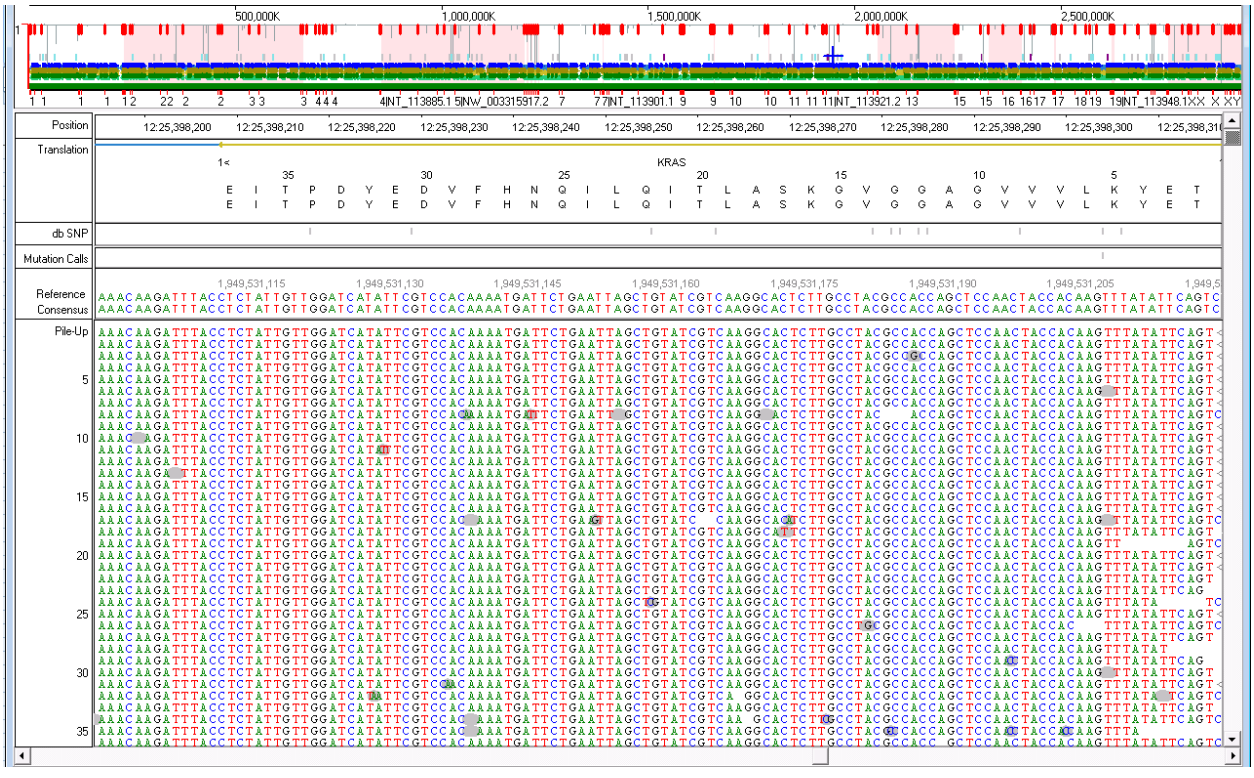


Figure 31. Sample Wildtype Region in Nextgene Browser. The region depicted is codons 5-37 of the KRAS gene, a common hotspot in many cancers including melanoma. By viewing this region in the Nextgene Browser, it can be determined that the sequence is relatively clean and void of mutations. This, this region would be identified as "wildtype" in the ROI table.

Figure 32. Sample Mutant Region in Nextgene Browser. The region depicted is codons 523-549 of the KIT gene, a common hotspot in many cancers including melanoma. A distinct mutation pattern can clearly be seen in the center of the image where there is repetition of a C base (highlighted in purple) at a position that is normally an A.

A	B	C	D	E	F
Pertinent Negatives					
Gene	Chromosome location Coverage of ROI	Exon	Codons covered	ROI manully reviewed (depth) x	Wildtype or Mutant?
BRAF	7,140453136	15	581-611	1844	MUTANT
BRAF	7,140481450	11	439-473	1414	Wildtype
CDKN2A	9,21971020	2	99-139	1204	Wildtype
CDKN2A	9,21971150	2	51-89	2522	Wildtype
CTNNB1	3,41266130	3	40-46	938	Wildtype
GNA11	19,3118925	5	202-219	1211	Wildtype
GNAQ	9,80409487	5	206-246	2563	Wildtype
HRAS	11,533880	3	42-82	1320	Wildtype
HRAS	11,534270	2	5-35	1128	Wildtype
KIT	4,55561763	2	23-60	5610	Wildtype
KIT	4,55592186	9	494-509	3206	Wildtype
KIT	4,55593418	10	523-549	4114	Wildtype
KIT	4,55593576	11	553-592	5894	Wildtype
KIT	4,55594221	13	627-664	1792	Wildtype
KIT	4,55595497	14	664-694	2490	Wildtype
KIT	4,55597495	15	715-732	4163	Wildtype
KIT	4,55599281	17	795-828	2828	Wildtype
KIT	4,55602674	18	829-865	2015	Wildtype
KRAS	12,25378647	4	114-150	1424	Wildtype
KRAS	12,25380277	3	39-66	3499	Wildtype
KRAS	12,25398225	2	05-37	2737	Wildtype
NRAS	1,115252220	4	124-150	2207	Wildtype
NRAS	1,115256529	3	43-70	2253	Wildtype
NRAS	1,115258746	2	03-31	2345	Wildtype

Figure 33. Completed ROI Table. The BRAF mutation is identified as “MUTANT” and all other positions as “Wildtype”.

Figure 34 depicts the completed Final Report Details page, containing the BRAF mutation and accompanying mutation details. The amino acid change “600V>VE” indicates a heterozygous mutation at codon 600 of the BRAF gene with a change from Valine (V) to Glutamic acid (E).

Ion Torrent Next Generation Sequencing Report

Review	Chromosome	Position	Gene ID	Total Coverage VC	Total Coverage NG	Quality/ Score	Stand Bias	Frequency VC	Frequency NG	Mutation Call	Amino Acid Change	Db_snp/ Cosmic
Common Findings												
TRUE	chr7	140453136	BRAF	1852	1745	8867.89	0.5085	42.1	41.49	c.1799T>TA	600V>VE	COSM476
Variant Caller Findings												
No Variant Caller findings to report.												
NextGENE Findings												
No NextGENE findings to report.												

Figure 34. Completed Final Report Details Page. The BRAF V600E mutation, detected by both Variant Caller and Nextgene, is listed in the "Common Findings" section.

3.1.3 Meditech Reports

After reviewing the case, the Meditech diagnosis report and raw data report were compiled. Figure 35 depicts an excerpt of the Meditech diagnosis report that was created by the program, and Figure 36 depicts the Meditech raw data report.

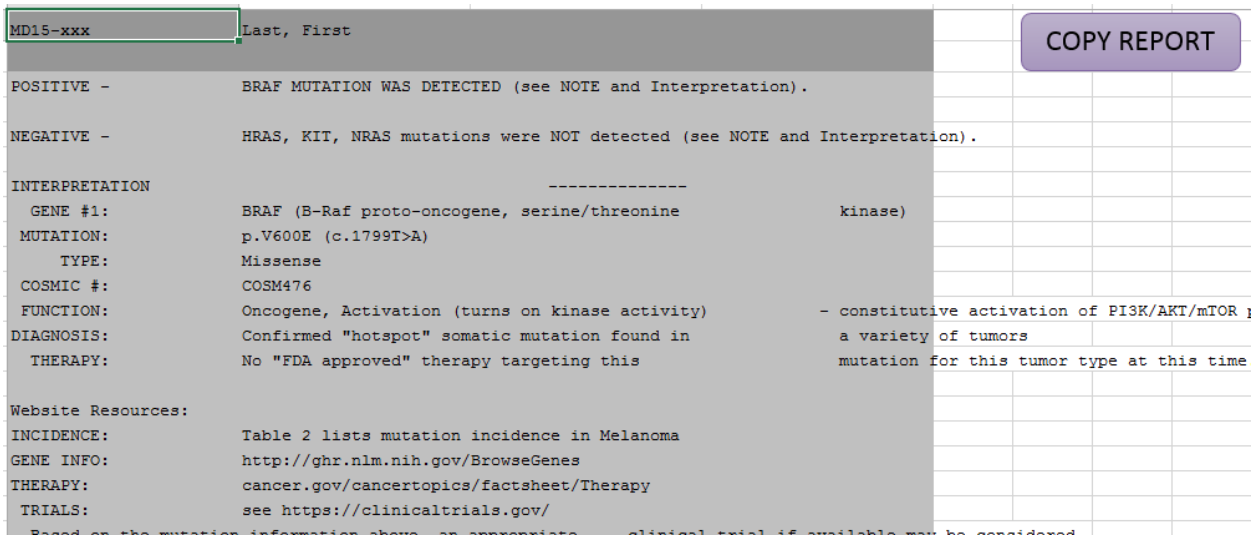


Figure 35. Completed Meditech Diagnosis Report. The "Copy Report" button copies the entire report to the clipboard in order to be pasted into the Meditech program.

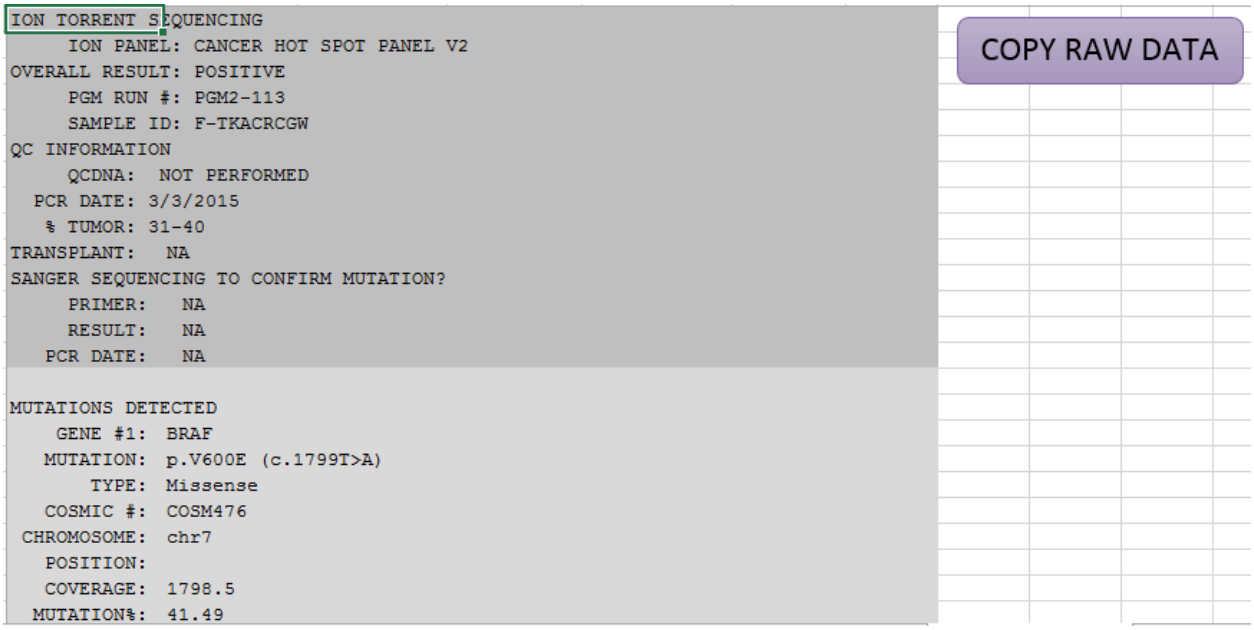


Figure 36. Completed Meditech Raw Data Report. The "Copy Raw Data" button copies the entire raw data report to the clipboard in order to be pasted into the Meditech program.

Figures 37 and 38 show the Diagnosis and Raw Data reports, respectively, after pasting them directly into the Meditech software. Upon saving the report, the data is automatically added to the Meditech database which can be accessed by technicians and physicians at a later time.

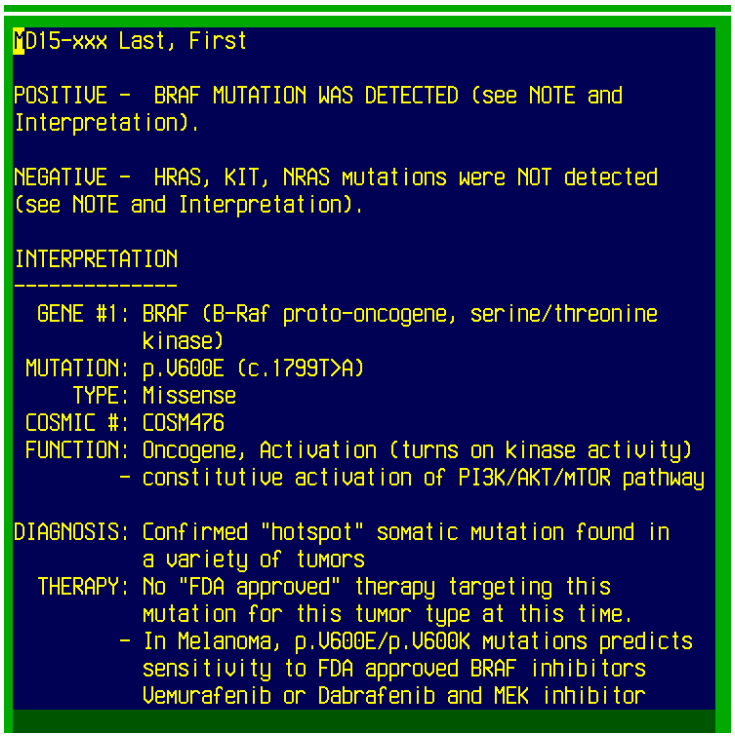


Figure 37. Completed Meditech Diagnosis Field. The figure above shows the diagnosis data pasted directly into the Meditech program from excel without editing.

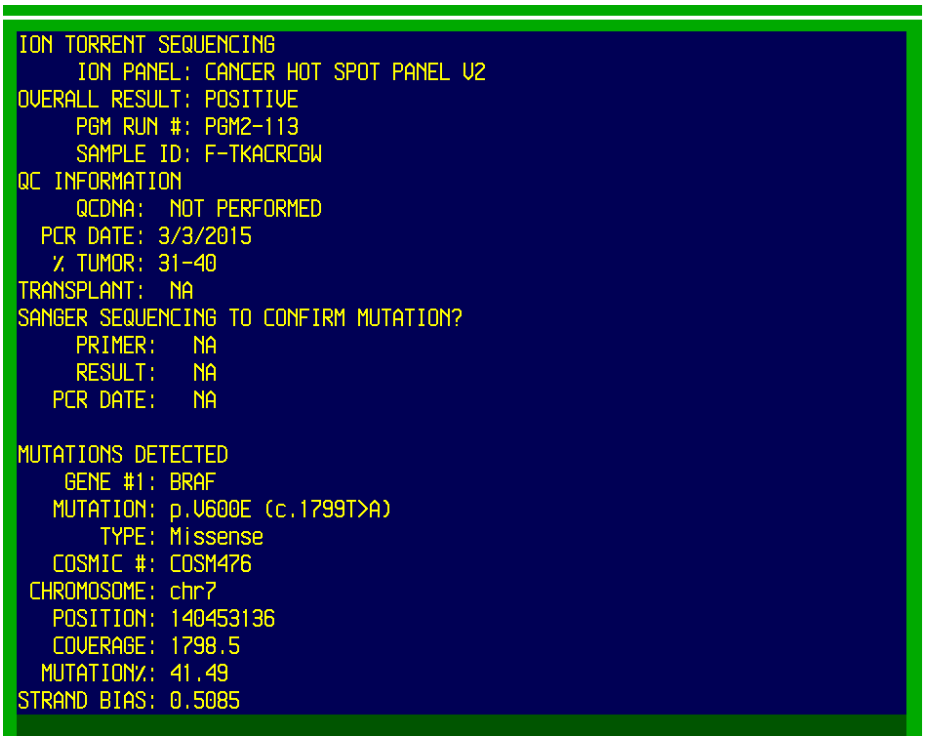


Figure 38. Completed Meditech Raw Data Field. The figure above shows the raw data pasted directly into the Meditech program from excel without editing.

3.1.4 Supercontrol Analysis

To test the supercontrol analysis, data from a successful PGM run was obtained and imported into the program. The supercontrol program was run and it was confirmed that all nine Ampliseq mutations were detected in the control sample. This process was repeated with supercontrol data from the myeloid panel, and it was once again confirmed that the program was able to successfully identify mutations in the control sample.

3.1.5 Implications for the Patient

In melanoma patients, the mutation from Valine (V) to glutamic acid (E) at codon 600 of the BRAF gene confers increased sensitivity to BRAF inhibitors. It is by far the most common BRAF V600 mutation and occurs in the activation segment of the kinase domain. Generally, mutations of this gene are found in patients who lack other driver mutations such as KIT and NRAS. Response to MEK and KIT inhibitors is currently unknown. There are no "FDA approved" therapies targeting V600E in melanoma at this time (Lovly et al., 2014).

3.2 Implementation

3.2.1 Staff Training

The first step of implementing the program involved familiarizing the staff of the laboratory with the user interface and analysis workflow. A presentation and successive training sessions were held with the technologists to show them how to use the different functions of the program. Additionally, a user manual was written with detailed descriptions of each analysis step (see Appendix A).

3.2.2 Obtaining User Feedback

In order to obtain feedback from the user, the staff was asked to run several test cases through the program and identify possible improvements that could made. Following is a list of some primary areas of concern:

1. If a low-level mutation is detected through ROI analysis *after* the program has already been run, how can the mutation be added to the raw data and final reports without having to start the entire analysis from the beginning.
2. All missense and silent SNPs should be included in the printed report for tracking reasons, but in some cases there may be far too many to include on the Final Report Details page with mutations.
3. Is there a way to eliminate typing errors for parts of the analysis which require manual review and input by the user, such as the review column in the data ("True" or "False") and the ROI result column ("Wildtype" or "Mutant").
4. If new panels are to be added to the NGS assay in the future (such as the myeloid panel, which is currently in progress), how can supercontrol analysis be performed if the mutations in the control are different than those for the Ampliseq panel (see Appendix C).
5. For AML patients, is there a way to include minimal residual disease (MRD) information in the final reports, and for the ROI table to be customized based on mutations that the patient was previously positive for.
6. For quality control purposes, each update of the program must include a version number and the genome references used in Variant Caller and Nextgene, which should be displayed on the printed report.

3.2.3 Changes Made in Response to Feedback

Taking the staff's feedback into account, several changes were made to the program over the course of testing:

1. *"Modify Nextgene" Button*

   In order to account for low-level mutations that are detected after the program has already been run, a "Modify Nextgene" button was created on the Final Report Summary worksheet. If a mutation is detected during ROI analysis that was not initially present in the Nextgene report and the user wishes to add it to the data, they can re-export the report in the Nextgene program (including the new mutation), import the new data into the "Nextgene Modified" worksheet, and run the Modify Nextgene program. This will locate the new mutation included in the data, add it to the bottom of the existing raw data report, and add the necessary details to the final reports without altering the analysis which has already been completed. This alleviates the need to start the entire analysis from the beginning, thus making it faster and easier for the user.

2. *SNP Report*

   In order to make room for large numbers of SNPs to be included in the printed report, the code was modified to transfer SNP data to an entirely new worksheet. This report contains only SNPs, while the mutation data remains on the Final Report Details worksheet.

3. *Dropdowns for Review Column and ROI Table*

   To limit user error in fields which require manual text input, drop-downs were incorporated which allow the user to choose from a list rather than adding free-text. This was accomplished using the "Data Validation" feature of excel, and using the "Name

Manager" to create ranges with the text options for the drop-down (similar to the dropdowns used on the Demographics worksheet). This allows the drop-downs to be completely customizable and to add text options if needed. For the Review column of the data, the user can choose from the following: "True", "False", "Strand Bias", "Homopolymer", "VUS", "Intron", "Splice Site", "SNP-Missense", "SNP-Silent". For the ROI result column, the user can choose from either "Wildtype" or "Mutant".

4. *Myeloid Supercontrol Analysis*

   In order to account for the differences between the Ampliseq and Myeloid panel controls, a new supercontrol analysis worksheet was created. Using similar principles to the Ampliseq analysis, this worksheet contains a list of 13 mutations present in the supercontrol (see Appendix C) which are checked against the control data for a given Myeloid panel run. The analysis also yields a database entry for the laboratory's Myeloid Quality Control Database

5. *AML MRD Modifications*

   AML cases identified as "MRD" are those which are being tested for the presence of a mutation that the patient was previously positive for. To include this information in the report, a more detailed MRD section was created for the Demographics page. This includes a table to be filled out by the technician with information regarding the mutations which were previously detected in the patient (see Figure 39). This information is then transferred to the Final Report Summary page, indicating whether the patient is still positive for the mutation based on findings from the current specimen (see Figure 40). Additionally, the MRD mutation is added to the AML ROI table if not already present.

13	**Tumor Information**					
14	Tumor Type	AML		If Other Tumor Type, Please Specify		
15	% Tumor	NA				
16	Mutation Frequency Threshold					
17						
18	**Additional Information for AML**					
19	Baseline?	No		Transplant Status	Post	
20	MRD?	Yes				
21		**Gene**	**Codon**	**Chromosome**	**Chromsome Position**	
22	If yes, please list the genes it is positive for, codon, chromsome, and position.	NPM1	290	5	171410548	
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						

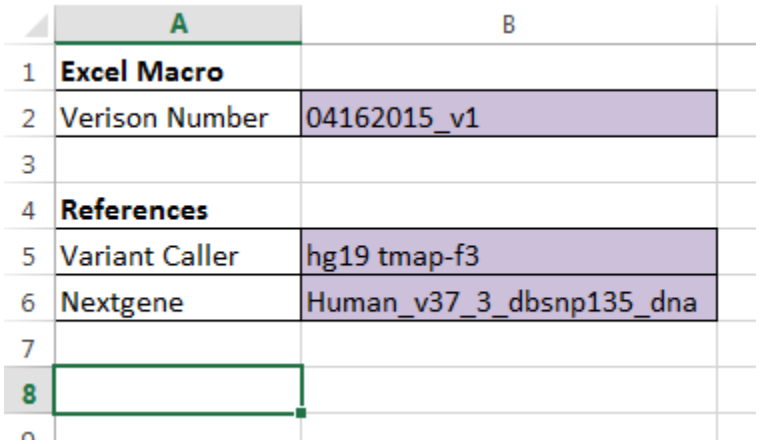
Figure 39. Sample MRD Section of Demographics Page. For MRD testing, information regarding the mutation(s) that the patient was previously positive for can be entered into the table and will be transferred to the reports. An NPM1 mutation at codon 290 has been used as an example.

	A	B	C	D	E	F	G	H
1	**Ion Torrent Next Generation Sequencing Report**						QC Review:	
2							Date:	
3	**Patient Name**	Last, First		**Analysis By**	0			
4	**DMO #**	MD15-xxx		**Analysis Date**	1/0/1900		**Excel Macro Version**	04212015_v2
5	**Cell Block/Thin Prep #**	0		**PGM Run #**	0		**Variant Caller Reference**	hg19 tmap-f3
6	**Patient Sex**	0		**Barcode #**	0		**Nextgene Reference**	Human_v37_3_dbsnp135_dna
7	**Patient Date of Birth**	1/0/1900		**PGM Instrument #**	0			
8	**Patient Age**	0		**Tumor Type**	AML			
9	**Unit #**	0		**Tumor %**	NA			
10	**Chip ID #**	0		**Analysis Threshold**	0			
11	**Control Name**	Supercontrol		**AML Baseline?**	No			
12	**Control Lot #**	011415		**Transplant Status**	Post			
13				**This MRD specimen was negative for the following variants detected at baseline: NPM1 (codon 290).**				

Figure 40. Sample Final Report for MRD Test. The MRD note, seen in row 13, indicates the status of the specimen in regards to previous findings. For example, if a patient was positive for an NPM1 mutation at baseline (see Figure 39) and is now NPM1 negative, the negative MRD note will be displayed, along with the gene and codon of the mutation.

6. *Reference and Version Information*

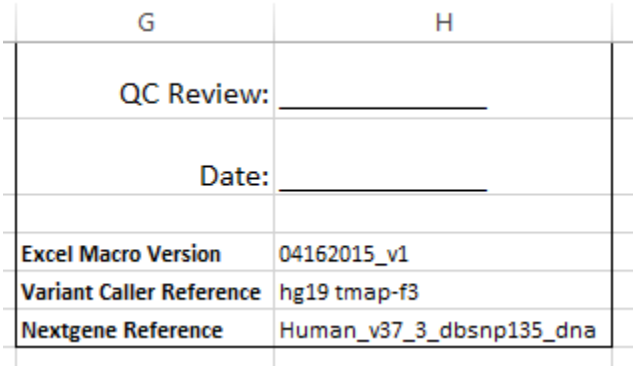
A "Reference" worksheet was added which includes both the program version (which is manually updated each time a change is made) as well as the Variant Caller and Nextgene reference numbers (see Figure 41).



	A	B
1	**Excel Macro**	
2	Verison Number	04162015_v1
3		
4	**References**	
5	Variant Caller	hg19 tmap-f3
6	Nextgene	Human_v37_3_dbsnp135_dna
7		
8		

Figure 41. Fields from the Reference Worksheet. The version number is manually updated each time a change is made in the code, using the format "date_version number". For example, the version picture above was last edited on 4/16/2015, and was the first version on that day.

This information is automatically transferred to the Final Report Summary worksheet upon running the program (see Figure 42).



G	H
QC Review:	_____
Date:	_____
Excel Macro Version	04162015_v1
Variant Caller Reference	hg19 tmap-f3
Nextgene Reference	Human_v37_3_dbsnp135_dna

Figure 42. Quality Control information on the Final Report Summary Worksheet.

3.3 Maintenance

Although the program was designed to be self-sustaining, continual upkeep of the program will ensure accuracy and efficiency. This includes troubleshooting new issues that are discovered by technicians and continuing to improve the program for the user. New versions of the Nextgene and Variant Caller software may require changes to the sorting and filtering of raw data. Additionally, the expansion of the NGS assay to include new gene panels will require updating the program (see Chapter 3.2). Virtually all of the databases in the software were designed to be expandable, namely those containing mutation data and drop-down options. The methods for adding and changing database entries can be found in Appendix A.

4. Conclusion

4.1 Applications for IQP

The goal of the IQP program at WPI is to apply principles of science and technology to address a need that exists in society. This project incorporated concepts of computer science, bioinformatics, and molecular oncology to develop a solution for NGS analysis in the Pathology Department of UMass Medical Center and to provide a service to the community which would otherwise be unavailable with the current resources. Ultimately, this project will benefit the cancer patients living in the Worcester community who are receiving care at the hospital.

One of the primary impacts of this project is decreasing the time that it takes to complete analysis of clinical NGS cases. Prior to the development of this program, manual review of cases took technicians an average of three hours to complete. Using the program, the technicians were repeatedly able to complete analysis in 45 minutes for a single case. By reducing the time needed to review cases, the turnaround time for clinical specimens will be improved and technicians will be able to analyze more cases each day.

Another significant impact of the program is increasing the accuracy of NGS analysis. The software was designed with several checkpoints to prevent issues such as specimen mix-up and incorrect reporting. For example, the patient-identity-check feature ensures that all data imported into the program belongs to the same sample, thus preventing data from different patients being mixed accidentally. Additionally, features such as the ROI table and highlighted genes of importance in the raw data reduce the likelihood of false negatives.

More significantly, the impact of this project extends beyond the laboratory to the hospital and those who are receiving care through UMass. As a non-profit healthcare system, UMass Memorial Health Care provides all levels of care to individuals in Central and Western

Massachusetts (US News Best Hospitals, 2015). With results from laboratory testing, the UMass Memorial Medical Center diagnoses and treats approximately 2800 new cancer patients each year (Clinical Care, 2015). By shortening the turnaround time for clinical specimens, this project has aimed to reduce the time which patients must wait for results. Additionally, the customizable nature of the program will allow for more detailed clinical reports. Laboratory technicians can now provide results that are highly specific to patients' tumor profiles without needing to manually analyze the data and compile the report.

4.2 Limitations and Recommendations

One of the most significant limitations over the course of this project was the time constraint. Because the IQP is completed over three academic terms (21 weeks), the objectives of the project had to not only be compliant with the requests of the laboratory but also reasonable to complete in a limited amount of time.

Another major constraint was designing the reports to be compatible with the Meditech software. Because they are used directly by physicians for diagnosis and treatment, the technical and aesthetic aspects of the reports had to be considered. To account for this, text in the databases was added to meet the character and spacing limitations of the Meditech program.

Future extensions of the work introduced in this project could involve expanding the analysis capabilities of the program and improving the user-interface for lab technicians. To make the program more efficient, the reference databases can be expanded to include notes for more genes and mutations. Additionally, more solid tumor types (such as desmoid and appendiceal carcinoma) can be added for the Ampliseq panel, and more blood-related cancers (such as chronic myelogenous leukemia and myelodysplastic syndrome) for the Myeloid panel.

Finally, future work could be done to expand the program features for new panels that are added to the NGS assay (such as the addition of MRD fields and a new supercontrol analysis worksheet for the Myeloid panel). These panels may be targeted at cancer types that are not currently included in the Ampliseq and Myeloid panels, such as breast cancer or lymphoma. The addition of new panels may call for modified analysis methods or the expansion of databases to include genes not currently detected.

References

Abramson RG (2014). Overview of targeted therapies for cancer. My Cancer Genome http://www.mycancergenome.org/content/other/molecular-medicine/overview-of-targeted-therapies-for-cancer

Barcellos-Hoff MH, Lyden D, Wang TC. (2013). The evolution of the cancer niche during multistage carcinogenesis. Nat Rev Cancer, 13(7), 511-518. doi: 10.1038/nrc3536

Camacho, J (2012). Molecular Oncology Principles and Recent Advances. p3-8. Mexico City, Mexico: Bentham Science Publishers.

Cancer Facts and Figures (2014). American Cancer Society: http://www.cancer.org

Cancer Genomics (2014). Illumina: http://applications.illumina.com/applications/cancer.html

Clinical Care (2015). Cancer Center of Excellence. University of Massachusetts Medical School. http://www.umassmed.edu/ccoe/clinical-care/

Davey S (2014). Next Generation Sequencing: Considering the Ethics. International Journal of Immunogenetics 41: 457-462.

Davis W, Yen D (1999). The Information System Consultant's Handbook. p7-8. Boca Raton, Florida: CRC Press LLC

Detecting Germline Mutations in Cancer (2014). Illumina: http://applications.illumina.com/applications/cancer/germline-mutations.html

Griffiths AJF, Miller JH, Suzuki DT, et al (2000). An Introduction to Genetic Analysis. 7th edition. New York: W. H. Freeman; 2000. Somatic versus germinal mutation: http://www.ncbi.nlm.nih.gov/books/NBK21894/

Gullapalli RR, Desai KV, Santana-Santos L, et al (2012). Next generation sequencing in clinical medicine: Challenges and lessons for pathology and biomedical informatics. Journal of Pathology Informatics: http://www.jpathinformatics.org/article.asp?issn=2153-3539;year=2012;volume=3;issue=1;spage=40;epage=40;aulast=Gullapalli

An Introduction to Next-Generation Sequencing Technology (2013). Illumina: http://res.illumina.com/documents/products/illumina_sequencing_introduction.pdf

Lantos, J. D., Artman, M., & Kingsmore, S. F. (2011). Ethical considerations associated with clinical use of next-generation sequencing in children. The Journal of Pediatrics, 159(6), 879–80.e1. doi:10.1016/j.jpeds.2011.07.035

Lovly, C., W. Pao, J. Sosman. (2014). BRAF c.1799T>A (V600E) Mutation in Melanoma. My Cancer Genome http://www.mycancergenome.org/content/disease/melanoma/braf/54/

Single Nucleotide Polymorphisms (2014). Genetics Home Reference: http://ghr.nlm.nih.gov/handbook/genomicresearch/snp

Rai, Arun (2011). Digital image. Synapse: http://www.bu.edu/synapse/files/2011/01/cancerprogression.png

US News Best Hospitals (2015). US News: http://health.usnews.com/best-hospitals/area/ma/umass-memorial-medical-center-6141095

Verhaak RGW, Goudswaard CS, van Putten W, et al. (2005). Mutations in nucleophosmin (NPM1) in acute myeloid leukemia (AML): association with other gene abnormalities and previously established gene expression signatures and their favorable prognostic significance. Blood, 106(12), 3747-3754. http://www.bloodjournal.org/content/106/12/3747

Vierstraete A (2012). Next generation sequencing. Ghent University: http://users.ugent.be/~avierstr/nextgen/Next_generation_sequencing_web.pdf

Wagener, C (2001). Molecular oncology: prospects for cancer diagnosis and therapy. Oncology: www.roche.com/pages/downloads/company/pdf/rtpenzberg01e.pdf

Weinberg, R (2014). The Biology of Cancer. 2nd Edition. New York, NY: Garland Science.

Appendix A: User Manual

Using the Program

Pre-Review

1. Save the workbook in the patient's folder with the DMO number and analysis date in the title.
2. Complete all fields of the *Demographics* worksheet with the patient and run information.
3. On the *Allele Coverage* worksheet, paste the downloaded allele coverage data into cell A1.
4. On the *Nextgene* worksheet, paste the downloaded Nextgene data into cell A1. When exporting the Nextgene mutation report, choose the settings seen in Figure A.1. To change the settings, go to Report>Mutation Report Settings in Nextgene Viewer.

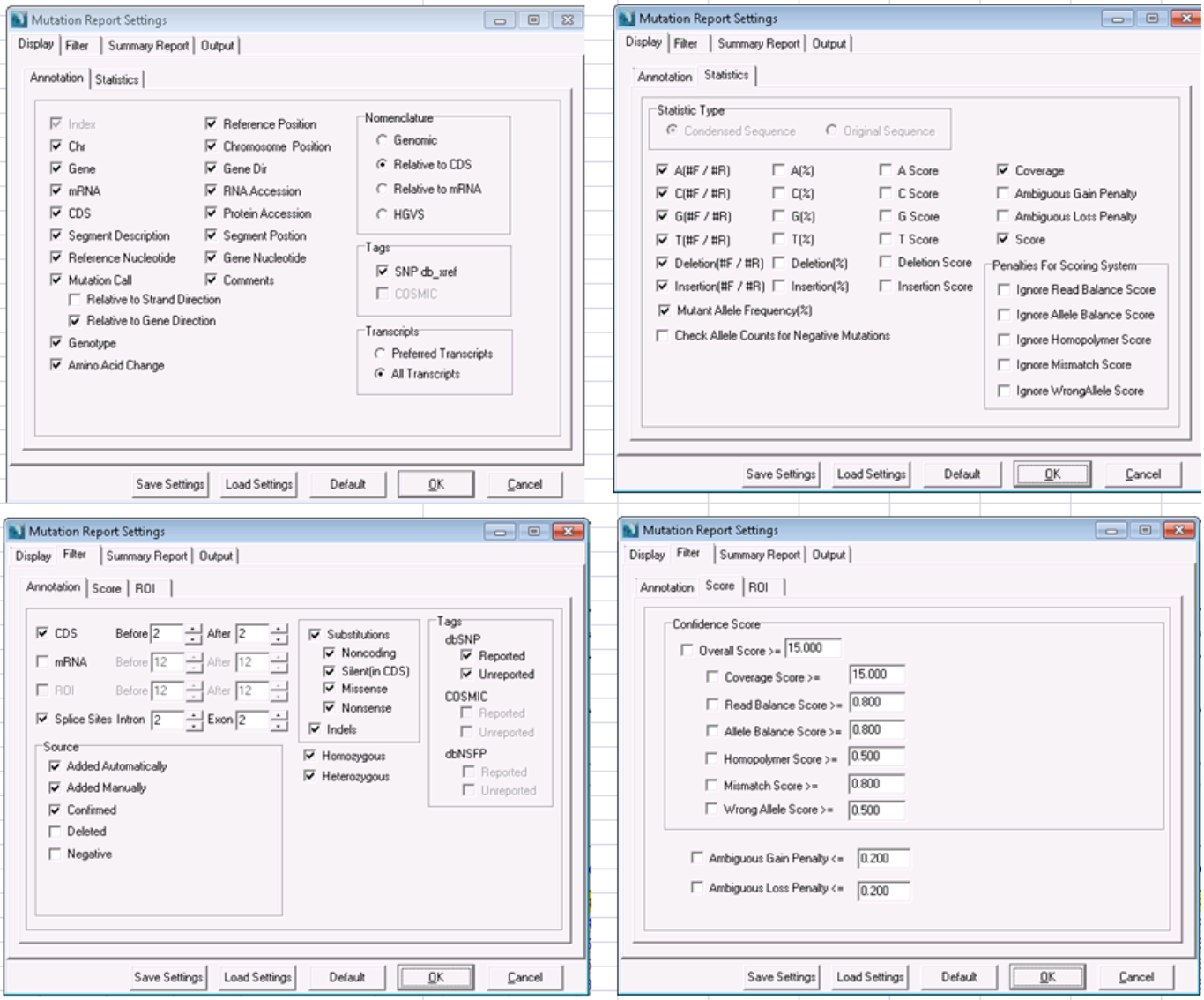


Figure A.1. Nextgene Mutation Report Settings.

5. On the *Coverage Analysis* worksheet, paste the coverage analysis data from the Ion Torrent browser into cell A1. This can be obtained from the PGM Run Summary by selecting the barcode from the **coverageAnalysis** menu, copying the entire Coverage Analysis Report (Ctrl+A), and pasting the data into the worksheet (Ctrl+V) as seen in Figure A.2.

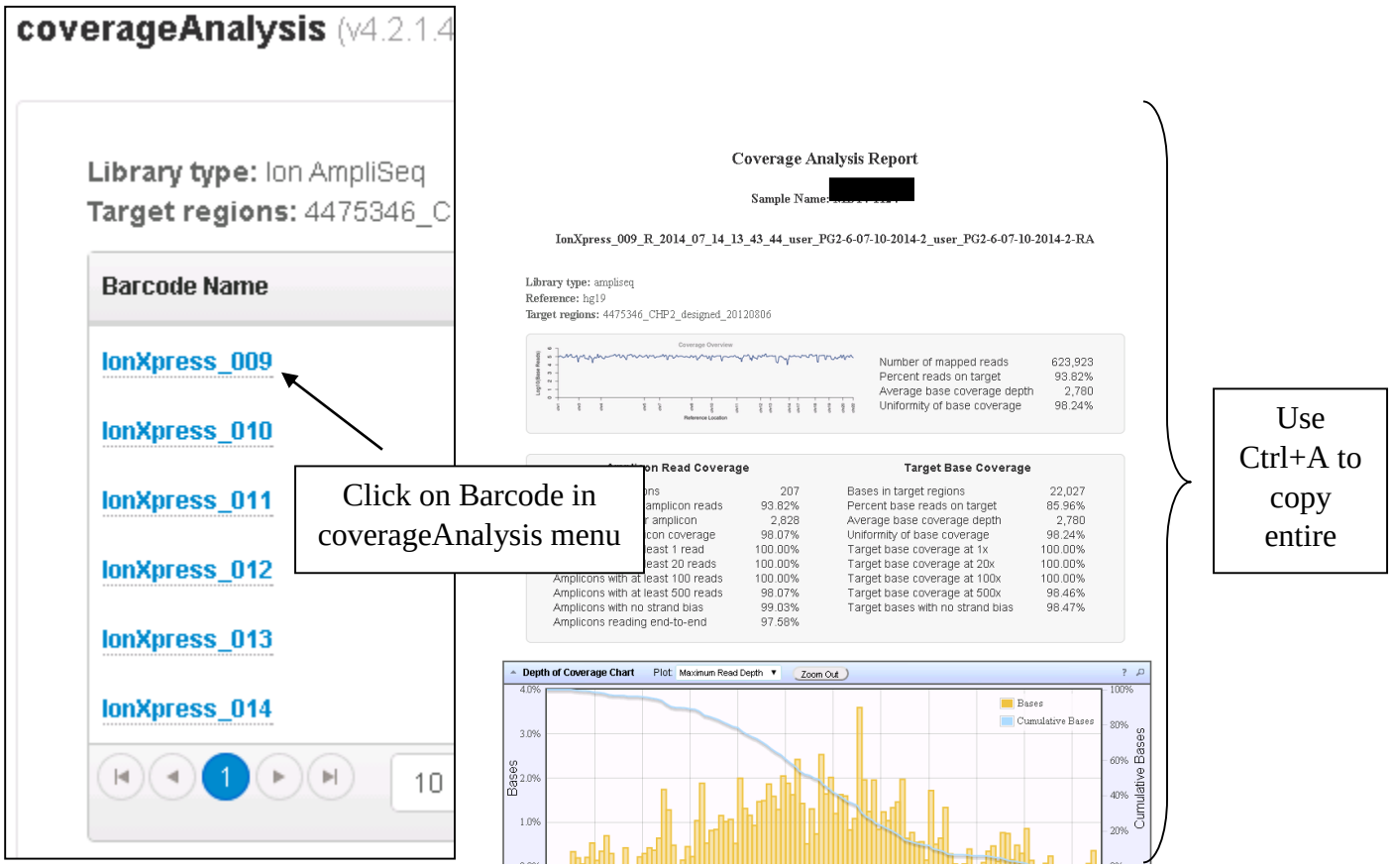


Figure A.2. Exporting Coverage Analysis Report

6. On the *Sample ID* worksheet, paste the sample ID data from the Ion Torrent browser. This can be obtained from the PGM Run Summary by selecting the barcode from the **sampleID** menu, copying the entire Sample ID Report (Ctrl+A), and pasting the data into the worksheet (Ctrl+V) as seen in Figure A.3.

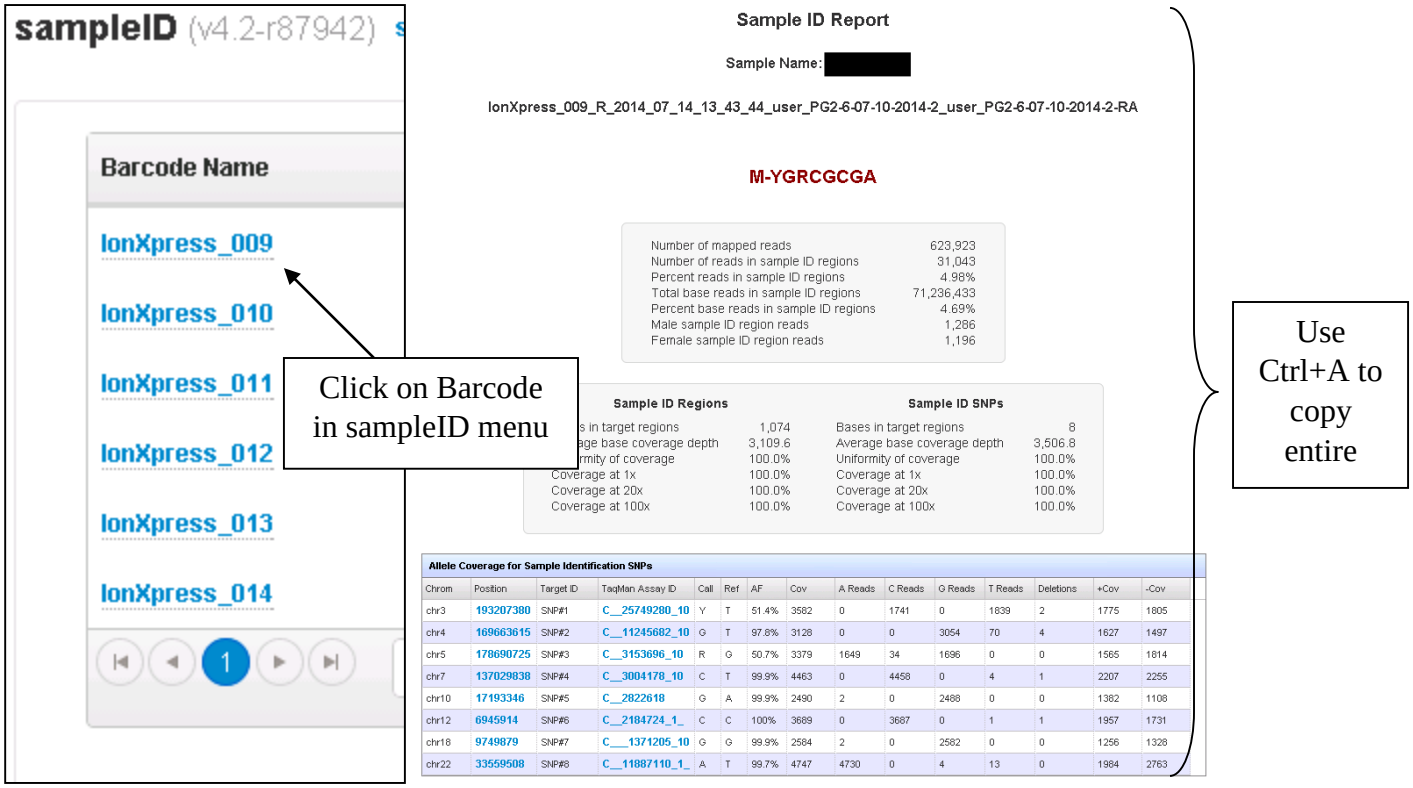


Figure A.3. Exporting Sample ID Report

7. On the *Final Repot Summary* worksheet, click the green **Pre-Review** button (see Figure A.4).

Figure A.4. Pre-Review Button

Once the Pre-Review program has been run, the data will be filtered and sorted by the parameters listed below in Table A.1. All unfiltered mutations will be moved to the top of the data and ready for the manual review process.

Table A.1. Pre-Review Filtering Parameters

SNP-MISSENSE	Missense SNP (change in amino acid)
SNP-SILENT	Silent SNP (no change in amino acid)
SNP-INTRON	SNP in Intron
ARTIFACT	Artifact (Homopolymer, Deletion, Strand Bias)
<MUT FREQ THRESH	Frequency of variant is less than the given threshold
<500X COV	Coverage of variant is less than 500x
LOW FREQ AND COV	Frequency and coverage of variant are less than the given thresholds

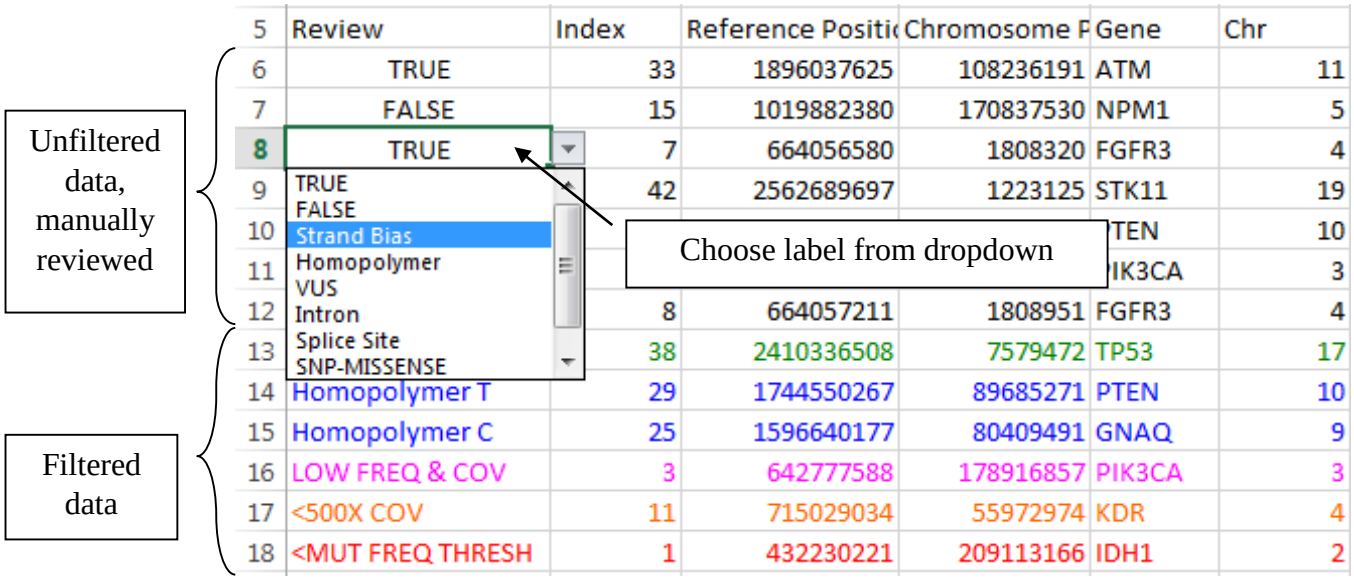
Post-Review

1. On the *Nextgene* worksheet, review each mutation and type the appropriate label in the "Review" column (A) for each position* (see Figure A.5).
2. On the *Variant Caller* worksheet, review each mutation and type the appropriate text in the "Review" column (A) for each position.*

Table A.2. Manual Review Label Options

Text	Description	Transferred to Final Report?
TRUE	True mutation	Yes
FALSE	False mutation	No
Strand Bias	One DNA strand favored over other in sequencing	No
Homopolymer	Sequence of identical bases	No
VUS	Variant of unknown significance	No
Intron	Variant in Intron	No
Splice Site	Mutation in splice site	No
SNP-MISSENSE	Missense SNP (change in amino acid)	Yes
SNP-SILENT	Silent SNP (no change in amino acid)	Yes

*Note: Only mutations labeled as TRUE, SNP-MISSENSE, and SNP-SILENT will be transferred to the Final Report.



	Review	Index	Reference Positi	Chromosome P	Gene	Chr
6	TRUE	33	1896037625	108236191	ATM	11
7	FALSE	15	1019882380	170837530	NPM1	5
8	TRUE	7	664056580	1808320	FGFR3	4
9		42	2562689697	1223125	STK11	19
10					TEN	10
11					IK3CA	3
12		8	664057211	1808951	FGFR3	4
13		38	2410336508	7579472	TP53	17
14	Homopolymer T	29	1744550267	89685271	PTEN	10
15	Homopolymer C	25	1596640177	80409491	GNAQ	9
16	LOW FREQ & COV	3	642777588	178916857	PIK3CA	3
17	<500X COV	11	715029034	55972974	KDR	4
18	<MUT FREQ THRESH	1	432230221	209113166	IDH1	2

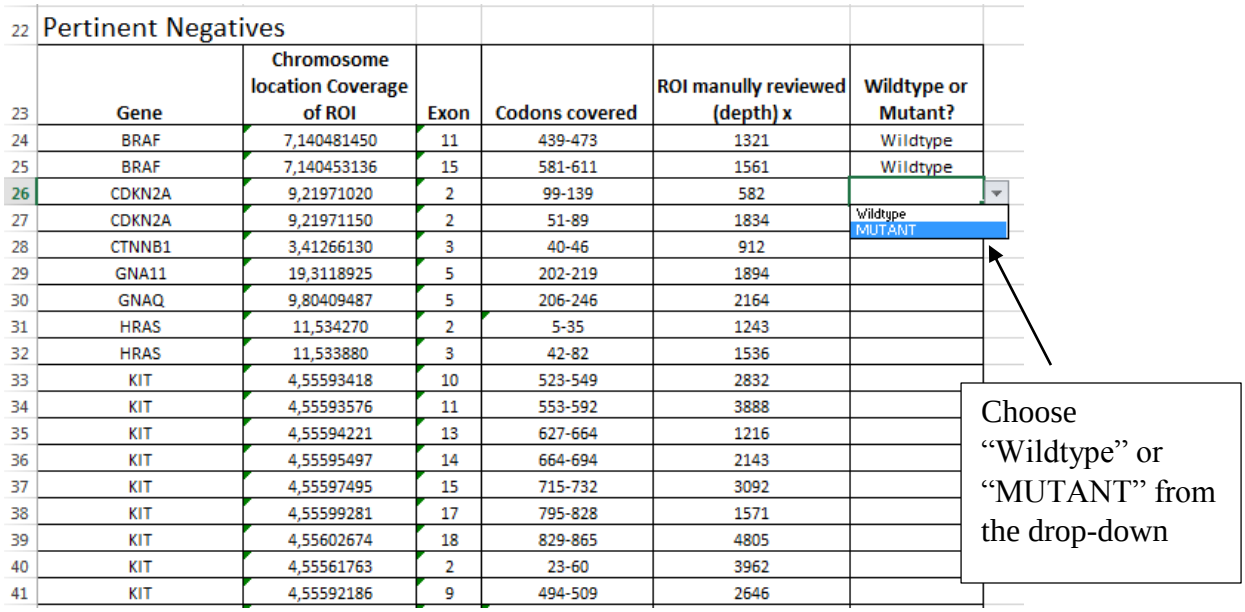
Figure A.5. Reviewing Variants

3. On the *Final Report Summary* worksheet, click the blue **Post-Review** button (see Figure A.6).

Figure A.6. Post-Review Button

Meditech Report

1. After running Post-Review, complete "Wildtype or Mutant" portion of Pertinent Negatives table on *Final Report Summary* worksheet by manually reviewing each position in Nextgene and ensuring that no mutation is present.



22 Pertinent Negatives

23 Gene	Chromosome location Coverage of ROI	Exon	Codons covered	ROI manully reviewed (depth) x	Wildtype or Mutant?
24 BRAF	7,140481450	11	439-473	1321	Wildtype
25 BRAF	7,140453136	15	581-611	1561	Wildtype
26 CDKN2A	9,21971020	2	99-139	582	
27 CDKN2A	9,21971150	2	51-89	1834	
28 CTNNB1	3,41266130	3	40-46	912	
29 GNA11	19,3118925	5	202-219	1894	
30 GNAQ	9,80409487	5	206-246	2164	
31 HRAS	11,534270	2	5-35	1243	
32 HRAS	11,533880	3	42-82	1536	
33 KIT	4,55593418	10	523-549	2832	
34 KIT	4,55593576	11	553-592	3888	
35 KIT	4,55594221	13	627-664	1216	
36 KIT	4,55595497	14	664-694	2143	
37 KIT	4,55597495	15	715-732	3092	
38 KIT	4,55599281	17	795-828	1571	
39 KIT	4,55602674	18	829-865	4805	
40 KIT	4,55561763	2	23-60	3962	
41 KIT	4,55592186	9	494-509	2646	

Wildtype
MUTANT

Choose "Wildtype" or "MUTANT" from the drop-down

Figure A.7.ROI Table

2. Once completed, ensure that all mutations and demographics present in *Final Report Details* and *Final Report Summary* worksheets are correct.
3. On the *Final Report Summary* worksheet, click the purple **Meditech** button (see Figure A.8).

Figure A.8. Meditech Diagnosis Report Button

4. Go to the *Meditech Report* worksheet and ensure that all data has correctly been transferred to the report. Click the purple **Copy Report** button to copy the entire Meditech report to the clipboard (see Figure A.9).

Figure A.9. Copy Meditech-Diagnosis-Report Button

5. Open the Meditech software and paste the report into the diagnosis field.
6. On the *Meditech Raw Data* worksheet, click the purple **Copy Raw Data** button to copy the entire Meditech raw data report to the clipboard (see Figure A.10).

Figure A.10. Copy Meditech-Raw-Data Button

Quality Metrics Database

1. On the *QM Data* worksheet, copy all visible rows and paste the data into the Quality Metrics Database.

Modify Nextgene

1. On the *Nextgene Modified* worksheet, paste the downloaded Nextgene data (including the new mutation) into cell A1. When exporting the Nextgene mutation report, use the mutation report settings outlined in the "Pre-Review" section.
2. On the *Final Report Summary* worksheet, click the red **Modify Nextgene** button (see Figure A.11).

Figure A.11. Modify-Nextgene Button

Supercontrol and Quality Control Database

1. On the *1% Supercontrol Allele Coverage* worksheet, paste the downloaded allele coverage data into cell A1.
2. On the *1% Supercontrol NG* worksheet, paste the downloaded Nextgene data into cell A1. When exporting the Nextgene mutation report, use the mutation report settings outlined in the "Pre-Review" section.
3. If running supercontrol analysis for the Ampliseq panel, choose the "Supercontrol Results- Ampliseq" worksheet and click the blue **Generate Ampliseq Supercontrol Results** button (see Figure A.12).

Figure A.12. Generate Ampliseq Supercontrol Results Button

4. If running supercontrol analysis for the Myeloid panel, choose the "Supercontrol Results- Myeloid" worksheet and click the green **Generate Myeloid Supercontrol Results** button (see Figure A.13).

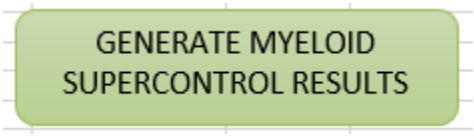


Figure A.13. Generate Myeloid Supercontrol Results Button

5. After running the analysis, copy the entire line of result data and paste it into the NGS Quality Control (QC) Database.

Important Notes:

- Do not type over or delete any of the text in the grey areas of the workbook.
- Do not save over the original template document.
- Do not run a program more than once or use them in a different order than described above. Doing so may disrupt the macros and produce incorrect results.

Changing Options

Adding Tumor Types

1. On the *Tumor ROI Options* worksheet, select the next empty cell in the "Tumor Type" column and enter the name of the tumor.
2. Beginning in column B of the same row, enter each ROI gene for the new tumor type. Ensure that each ROI gene is entered in a new, adjacent cell, and that the cell contains only the name of the gene with no spaces or other text. Figure A.14 shows an example tumor type, "Liver", being added to the database with the ROI genes "BRAF", "KRAS", and "NRAS".

	A	B	C	D	E	F	G	H	I
1	**Tumor Type**	**ROI Genes**							
2	Colon Cancer (CRC)	BRAF	KRAS	NRAS	PIK3CA				
3	Breast Cancer								
4	Lung Cancer	BRAF	EGFR	ERBB2	KRAS				
5	AML	FLT3-TKI	IDH1	IDH2	JAK2	KIT	KRAS	NPM1	NRAS
6	Pancreatic CA & Cyst Fluic	BRAF	CTNNB1	GNAS	KRAS	PIK3CA	VHL		
7	Malignant Melanoma	BRAF	HRAS	KIT	NRAS				
8	Thyroid Cancer	BRAF	GNAS	HRAS	KRAS	NRAS	PIK3CA	RET	
9	GIST	KIT	PDGFRA						
10	Other								
11	Liver	BRAF	KRAS	NRAS					
12									

Figure A.14. Expanding the Tumor ROI Database

Removing Tumor Types

1. On the *Tumor ROI Options* worksheet, select the entire row of the tumor type which you wish to remove from the database. Right click on the row and select "Delete". Figure A.15 shows the example tumor type, "Liver", being removed from the database.

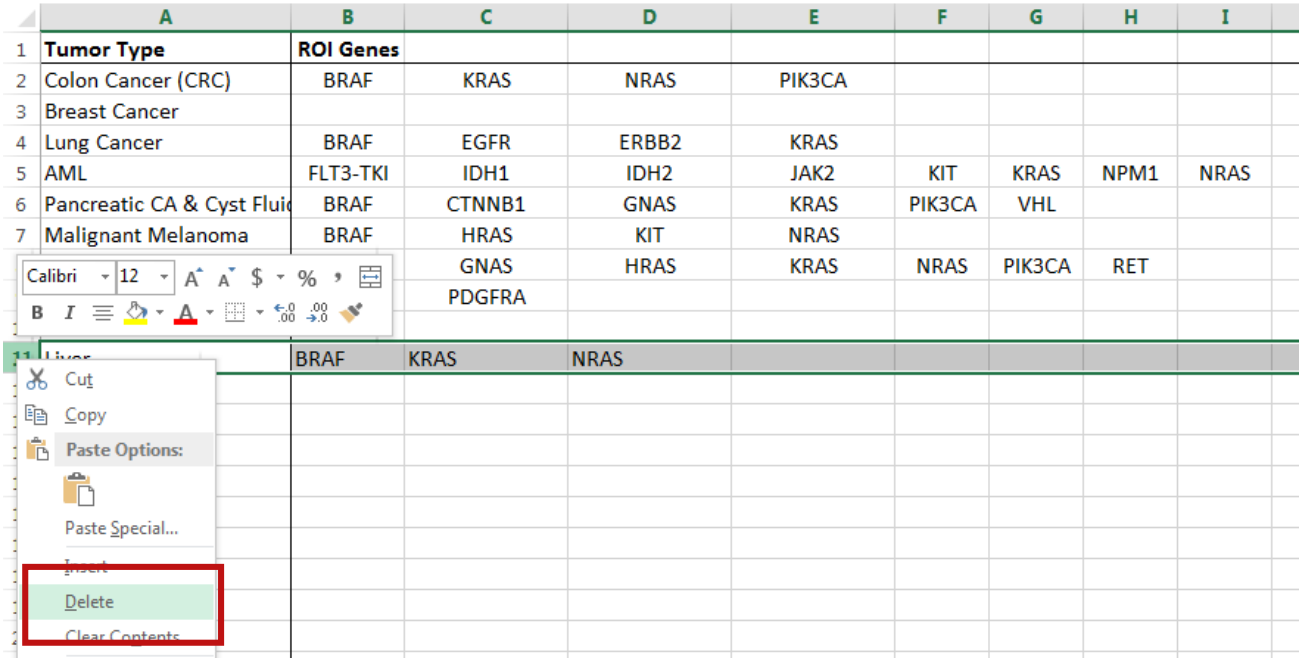


Figure A.15. Deleting Tumor Type from ROI Database

Adding/Removing ROI Genes for Existent Tumor Types

1. To **add** an ROI gene for an existing tumor type, go to the *Tumor ROI Options* worksheet and select the next empty cell in the row corresponding to the desired tumor type. Enter the name of the gene, with no spaces or other characters. Figure A.16 shows an example gene, "TP53", being added to the ROI list for the tumor type "Liver".

	A	B	C	D	E	F	G	H	I
1	**Tumor Type**	**ROI Genes**							
2	Colon Cancer (CRC)	BRAF	KRAS	NRAS	PIK3CA				
3	Breast Cancer								
4	Lung Cancer	BRAF	EGFR	ERBB2	KRAS				
5	AML	FLT3-TKI	IDH1	IDH2	JAK2	KIT	KRAS	NPM1	NRAS
6	Pancreatic CA & Cyst Fluid	BRAF	CTNNB1	GNAS	KRAS	PIK3CA	VHL		
7	Malignant Melanoma	BRAF	HRAS	KIT	NRAS				
8	Thyroid Cancer	BRAF	GNAS	HRAS	KRAS	NRAS	PIK3CA	RET	
9	GIST	KIT	PDGFRA						
10	Other								
11	Liver	BRAF	KRAS	NRAS	TP53				
12									

Figure A.16. Deleting Tumor Type from ROI Database

2. To **delete** an ROI gene, select the cell containing the gene and delete all of the text.

Changing Demographic Options

All worksheet dropdowns (except for tumor type) are controlled by the *Demographic Options* worksheet. Table A.3 provides a list of options the user may want to edit, and the

corresponding column on the *Demographic Options* worksheet. To add to any of the following columns, select the next empty cell and enter the new text.

Table A.3. Demographics Options

Column Title (on *Demographic Options* worksheet)	**Function**
Analysis By	List of technicians for "Analysis By" field on *Demographics* worksheet
Gene	List of genes for MRD table on *Demographics* worksheet
Review Options	List of labels for "Review" column of Variant Caller and Nextgene data
PGM Run	List for "PGM Run #" field on *Demographics* worksheet
MD Number	List of MD numbers for "DMO Number" field on *Demographics* worksheet

Adding to Databases

Note Entry Program

1. On the *Mediech Note Entry Program* worksheet, complete all of the blue fields with information for the new note.
2. Click the blue **Generate Note** button (see Figure A.17).

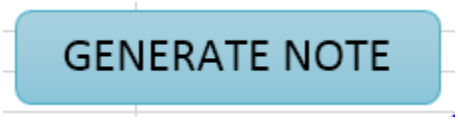


Figure A.17. Generate Note Button

3. Copy the entire row of data and paste it into the next empty row of the *Meditech Notes* worksheet. Figure A.18 shows an example note created for a BRAF V600E mutation in Melanoma. The data was entered in the fields at the top and automatically formatted in line 11 to be pasted into the database.

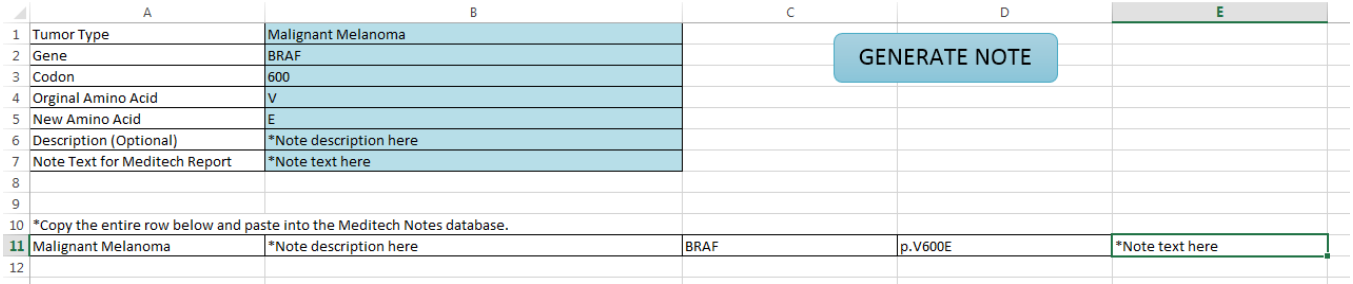


Figure A.18. Using the Meditech Note-Entry Program

Expanding Meditech Gene Databases

Note: These directions pertain to the following Meditech databases: Gene Function General, Gene Function, Gene Diagnosis, Gene Prognosis, and Gene Therapy

1. To edit a current database entry, select the cell containing the text that you wish to change and enter the new text. Manual spacing of words may be required for the data to be transferred to Meditech with certain formatting (ie. indentations).
2. To add an entry, select the next empty row at the bottom of the database and enter the mutation information in the correct columns based on the column titles in Row 1 (ie. gene, COSMIC ID number, amino acid mutation, etc.)

Expanding Meditech Resources Database

1. To edit the *Resources by Tumor* worksheet, select the cell in the appropriate tumor type that you wish to edit and enter the new text.

*Note: Resources for new tumor types will require additional programming.

Expanding Meditech End of Report Database

1. To edit a current database entry in the *End of Report by Tumor* worksheet, select the cell containing the text that you wish to change and enter the new text. Manual spacing of words may be required for the data to be transferred to Meditech with certain formatting (ie. indentations).
2. To add an entry for a new tumor type, select the next empty column and type the name of the tumor in row 1 **as it appears in the Tumor ROI Options database.** Begin entering end-of-report text in the same column in row 2.
3. To add a row of text to an existing database entry, select the next empty cell under the column of the desired tumor type and enter the new text.

Appendix B: NGS Panels

Table B.1. Genes Included in the Ion Ampliseq Cancer Hotspot Panel v2

ABL1	EGFR	GNAS	KRAS	PTPN11
AKT1	ERBB2	GNAQ	MET	RB1
ALK	ERBB4	HNF1A	MLH1	RET
APC	EZH2	HRAS	MPL	SMAD4
ATM	FBXW7	IDH1	NOTCH1	SMARCB1
BRAF	FGFR1	JAK2	NPM1	SMO
CDH1	FGFR2	JAK3	NRAS	SRC
CDKN2A	FGFR3	IDH2	PDGFRA	STK11
CSF1R	FLT3	KDR	PIK3CA	TP53
CTNNB1	GNA11	KIT	PTEN	VHL

Table B.2. Genes Included in the Qiagen Myeloid Neoplasm Panel

ABL1	DDX41	JAK2	PRPF40B	SRSF2
ASXL1	DNMT3A	JAK3	PTPN11	STAG2
ATRX	EED	KAT6A	RAD21	SUZ12
BCOR	ETV6	KIT	RB1	TET2
BCORL1	EZH2	KRAS	RUNX1	TLR9
BRAF	FLT3	MLL-only PTD	SETBP1	TP53
Calreticulin	GATA1	MPL	SF1	U2AF1
CBL	GATA2	MyD88	SF3A1	U2AF2
CBLB	IDH1	NF1	SF3B1	WT1
CEBPA	IDH2	NPM1	SH2B3	ZRSR2
CSF3R	IKZF1	NRAS	SMC1A	
DAXX	JAK1	PHF6	SMC3	

Appendix C: Supercontrol Mutations

Table C.1. Ampliseq Supercontrol Mutations

Gene ID	Region Name	Position	Mutation Call	Amino Acid Change	COSMIC ID
KRAS	CHP2_KRAS	25398285	c.34G>TG	p.G12C	COSM516
BRAF	CHP2_BRAF	140453136	c.1799T>TA	p.V600E	COSM476
PIK3CA	CHP2_PIK3CA	178952085	c.3140A>AG	p.H1047R	COSM775
EGFR	CHP2_EGFR_EX21	55259515	c.2573T>GT	p.L858R	COSM6224
EGFR	CHP2_EGFR_20	55249071	c.2369C>CT	p.T790M	COSM6240
EGFR	CHP2_EGFR_EX19	55242466	c.2236_2250delGAATTAAGAGAAGCA	c.748-750delGAATT	COSM6225
NRAS	CHP2_NRAS_2	115258744	c.38G>GA	p.G13D	COSM573
KIT	CHP2_KIT	55593662	c.1728-1730-delTCC	c.1728-1730-delTCC	
NPM1	CHP2_NPM1	170837548	c.863_864insTCTG	c.863insTCTG	COSM17559

Table C.2. Myeloid Supercontrol Mutations

Gene ID	Position	Mutation Call	Amino Acid Change	COSMIC
IDH1	209113113	c.394C>T	p.R132C	COSM28747
NPM1	170837544	c.863insTCTG	p.W288fs*12	COSM17559
DNMT3A	25457242	c.2645G>A	p.R882H	COSM52944
IDH2	90631934	c.419G>A	p.R140Q	COSM41590
RUNX1	36252939	c.423insAGGG	FS	none
CBL	119148982	c.1202G>A	p.C401Y	COSM87284
ETV6	12006459	c.427C>CT	p.Q143X	none
DNMT3A	25457265	c.2622T>TA	p.Y874X	none
FLT3	28608269	c.1787_1788ins30bp	In-Frame	none
KRAS	25398285	c.34G>C	p.G12R	COSM518
EZH2	148543659	c.149T>TC	p.L50S	none
NRAS	115258744	c.182A>T	p.G13D	COSM583
BRAF	140453131	c.1799T>A	p.V600E	COSM476

Appendix D: VBA Module Database

Table D.1. Database of VBA Modules

	Macro Name	Function
Pre-Review	**PRE_**	Runs all Pre-Review macros
	pre_prepare	Prepares data for formatting. Compiles VC report from allele coverage; deletes unneeded columns of NG data
	pre_format	Formats raw data
	pre_filter_snps	Filters SNPs; changes green
	pre_filter_mutfreq	Filters low mutation frequency; changes red
	pre_filter_lowcov	Filters low coverage (<500x); changes orange
	pre_filter_freqandcov	Filters low mutation frequency and coverage; changes pink
	pre_filter_artifacts	Filters artifacts; changes blue
	pre_format_sortbyfilter	Sorts unfiltered data to top and filtered data by filter type
	pre_format_chrpos	Adds a column with chromosome and position separated by a comma
	pre_format_color	Adds color coded legend to top of raw data report
	pre_format_style	Adds borders and styles to raw data report
	pre_format_cov_an	Formats raw coverage analysis data by removing unneeded percent signs.
Post-Review	**POST_**	Runs all Post-Review Macros
	post_det_find_common	Populates common variants section of Final Report Details with mutations found by both programs
	post_det_find	Populates respective section of Final Report Details with mutations found in that program
	post_det_find_snps	Populates Final Report SNPs worksheet with missense and silent SNPs found by NG
	post_det_no_mutations	Adds message to Final Report Details table if no mutations were found for respective section
	post_sum_mutations	Adds a mutation summary to Final Report Summary page for all detected mutations (common, VC, and NG)
	pot_sum_roi_table	Compiles ROI columns from ROI databased based on identified tumor type
	post_sum_roi_coverage	Adds coverage to ROI table based on Allele Coverage data

	post_sum_mrd	Adds MRD note for AML cases
	post_sum_pat_ide_check	Runs all patient identity checks and displays message at bottom of Final Report Summary page
	post_qm_sort	Sorts rows with data to top of QM data report
	post_qm_hide	Hides unneeded rows of data in the QM report
	post_highlight_roi	Highlights ROI positions in raw data
Supercontrol-Ampliseq	**SUPER_**	Runs all 1% Supercontrol macros
	super_prepare_vc	Compiles Variant Caller report from allele coverage data for supercontrol
	super_format	Formats raw data
	super_find_muts	Identifies supercontrol mutations in data; highlights green
	super_nomut	Adds "NO" to "Detected?" column of supercontrol results if mutation was not found in data
	super_avg_freq	Averages mutation frequency between programs and adds to "Avg Frequency" column of supercontrol results
	super_coverage	Averages mutation coverage between programs and adds value to "Avg Coverage" column of supercontrol results
	super_addtitle	Adds title to Supercontrol report
Supercontrol-Myeloid	super_prepare_vc	*See supercontrol macros
	super_format	*See supercontrol macros
	super_my_find_muts	Identifies supercontrol mutations in data; highlights green
	super_my_nomut	Adds "NO" to "Detected?" column of supercontrol results if mutation was not found in data
	super_my_avg_freq	Averages mutation frequency between programs and adds to "Avg Frequency" column of supercontrol results
	super_my_coverage	Averages mutation coverage between programs and adds value to "Avg Coverage" column of supercontrol results
	super_addtitle	*See supercontrol macros
Meditech	**MEDI_**	Runs all Meditech macros
	medi_clear	Clears entire Meditech report on click of button.
	medi_clear_raw	Clears entire Meditech raw data report on click of button.

	medi_copy	Copies entire Meditech report on click of button.
Meditech Raw Data	medi_copy_raw	Copies entire Meditech raw data report on click of button.
	medi_raw_1_result	Adds result to Meditech raw data report.
	medi_raw_2_interp	Adds interpretation to Meditech raw data report
	medi_raw_3_table	Adds ROI table to Meditech raw data report.
	medi_raw_4_end_of_report	Adds end of Meditech raw data report.
Meditech Report	medi_report_1_result	Adds result to Meditech report from mutation findings.
	medi_report_2_notes	Adds notes to Meditech report from database.
	medi_report_2b_notes_other	Adds other notes selected by user to Meditech report.
	medi_report_3_interp	Adds interpretation to Meditech report from databases.
	medi_report_4_resour	Adds resources to Meditech report based on tumor type.
	medi_report_5_table1	Compiles Table 1 in Meditech report based on mutation findings and ROI table.
	medi_report_6_end_of_report	Adds end of Meditech report based on tumor type
Modify Nextgene	MODING_	Runs all Modify Nextgene macros
	pre_prepare_ng	*see macro in Pre-Review
	moding_find_new	Transfers row to Nextgene report and adds it to Final Report Details
	post_sum_mutations	*see macro in Post-Review

Appendix E: Code

```
Attribute VB_Name = "Module1"
Sub pre_format_vc()
Attribute pre_format_vc.VB_ProcData.VB_Invoke_Func = " \n14"
'Add Review Column
    Columns("A:A").Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
'Move data down to row 5
    Rows("1:1").Select
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
'Add "Review" text
    Range("A5").Select
    ActiveCell.FormulaR1C1 = "Review"
    Range("A5").Select
'Add DMO and Analysis Info at top
    'Add text
        'DMO number
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "DMO Number"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "=Demographics!RC[1]"
        'Analysis by
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "Analysis By"
    Range("B3").Select
    ActiveCell.FormulaR1C1 = Sheets("Demographics").Range("E8").Text
End Sub
Attribute VB_Name = "Module10"
Attribute VB_Name = "Module11"
Attribute VB_Name = "Module12"
Sub pre_prepare_vc()
'Compiles Variant Caller report from allele coverage data
'All data with Allele Call of "Heterozygous" or "Homozygous" is added to report
'Go to allele coverage data
Sheets("Allele Coverage").Select
Dim LSearchRow As Integer
    'Line of Allele Coverage Data
Dim LCopyToRow As Integer
    'Line of Variant Caller Data
'Copy header to variant caller
Rows("1:1").Select
Selection.Copy
Sheets("Variant Caller").Select
Rows("1:1").Select
ActiveSheet.Paste
'Go back to allele coverage data
Sheets("Allele Coverage").Select
'Start searching Allele Coverage Data at Row 2
LSearchRow = 2
'Start copy data to Variant Caller at Row 2
LCopyToRow = 2
'For all of the allele coverage data
While Len(Sheets("Allele Coverage").Range("A" & CStr(LSearchRow)).Text) > 0
'If text in column E is "Heterozygous" or "Homozygous"
If Sheets("Allele Coverage").Range("E" & CStr(LSearchRow)).Text = "Heterozygous" Or
Sheets("Allele Coverage").Range("E" & CStr(LSearchRow)).Text = "Homozygous" Then
'Select row to copy
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).SpecialCells(xlCellTypeVisible).Select
Selection.Copy
'Paste row into Variant Caller in next empty row
Sheets("Variant Caller").Select
Rows(CStr(LCopyToRow) & ":" & CStr(LCopyToRow)).Select
ActiveSheet.Paste
'Move counter to next row
LCopyToRow = LCopyToRow + 1
'Go back to allele coverage data and keep searching
```

```
Sheets("Allele Coverage").Select
End If
'Check next row of allele coverage data
LSearchRow = LSearchRow + 1
Wend
Sheets("Allele Coverage").Range("A1").Select
End Sub
Attribute VB_Name = "Module13"
Attribute VB_Name = "Module14"
Sub pre_format_sortbyfilter()
Dim NextColumn As Integer
    'Next empty column of data (last column + 1)
Dim LSearchRow As Integer
    'search row
Dim SortRange As Range
    'full range of data to be sorted
Dim LastRow As Integer
    'last row of data
Dim NextColumnRange As Range
    'Beginning of Unfiltered? column data
'Set NextColumn
Range("A5").Select
NextColumn = Range(Range("A5"), Selection.End(xlToRight)).End(xlToRight).Column + 1
'Set LastRow
Range("B5").Select
LastRow = Range(Range("B5"), Selection.End(xlToRight)).End(xlDown).Row
'Start search in row 6
LSearchRow = 6
'Select row of 5 of column and add title
Cells(5, NextColumn).Select
ActiveCell.FormulaR1C1 = "Unfiltered?"
'For all values in the sheet
While Len(Range("C" & CStr(LSearchRow)).Value) > 0
'If data is not filtered (ie. column A is empty)
If IsEmpty(Range("A" & CStr(LSearchRow))) Then
'Type "TRUE" in Unfiltered? column
Cells(LSearchRow, NextColumn).Select
ActiveCell.FormulaR1C1 = "TRUE"
'Else, type "FALSE" in Unfiltered? column
Else
Cells(LSearchRow, NextColumn).Select
ActiveCell.FormulaR1C1 = "FALSE"
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
'Hide column
Cells(1, NextColumn).Select
Selection.EntireColumn.Hidden = True
'Sort unfiltered data to top and filtered data to bottom by filter type
Set SortRange = Range(Cells(5, 1), Cells(LastRow, NextColumn))
Set NextColumnRange = Range(Cells(6, NextColumn), Cells(6, NextColumn))
SortRange.Sort Key1:=NextColumnRange, Order1:=xlDescending, Key2:= _
        Range("A6"), Order2:=xlDescending, Header:=xlGuess, OrderCustom:=1, _
        MatchCase:=False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, _
        DataOption2:=xlSortNormal
End Sub
Attribute VB_Name = "Module15"
Sub post_det_find_common()
'Finds variants that are common between Nextgene and Variant Caller data
'and marked as TRUE
'based on chromosome position
Sheets("Variant Caller").Select
'VARIABLES
Dim VCRow As Integer
    'line of data in variant caller
Dim LastNGRow As Integer
    'last line of data in nextgene
Dim CRow As Integer
    'row of Final Report to paste
'Next empty row of Final Report; initially 4
```

```
CRow = 4
'Start search at row 6 in VC
VCRow = 6
'Last row of data in NG
LastNGRow = Sheets("Nextgene").Range("D6").End(xlDown).Row
'For each row of data in variant caller
While Len(Sheets("Variant Caller").Range("C" & VCRow).Text) > 0
'For each row of data in NG
For i = 6 To LastNGRow
'If position in VC matches position in NG
If Sheets("Variant Caller").Range("C" & CStr(VCRow)).Value = Sheets("Nextgene").Range("D" &
CStr(i)).Value Then
'And both are marked as TRUE mutation
If Sheets("Variant Caller").Range("A" & CStr(VCRow)).Value = True And
Sheets("Nextgene").Range("A" & CStr(i)).Value = True Then
'Transfer data to Final Report under Common Variants
'Review
Sheets("Final Report Details").Select
Sheets("Final Report Details").Range("A" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("A" & CStr(VCRow)).Text
'Chromosome
Sheets("Final Report Details").Range("B" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("F" & CStr(i)).Text
'Position
Sheets("Final Report Details").Range("C" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("C" & CStr(VCRow)).Text
'GeneID
Sheets("Final Report Details").Range("D" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("E" & CStr(i)).Text
'Total Coverage VC
Sheets("Final Report Details").Range("E" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("T" & CStr(VCRow)).Text
'Total Coverage NG
Sheets("Final Report Details").Range("F" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("J" & CStr(i)).Text
'Quality/Score
Sheets("Final Report Details").Range("G" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("J" & CStr(VCRow)).Text
'Strand Bias
Sheets("Final Report Details").Range("H" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("AD" & CStr(VCRow)).Text
'Frequency VC
Sheets("Final Report Details").Range("I" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("I" & CStr(VCRow)).Text
'Frequency NG
Sheets("Final Report Details").Range("J" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("T" & CStr(i)).Text
'Mutation Call
Sheets("Final Report Details").Range("K" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("S" & CStr(i)).Text
'Amino Acid Change
Sheets("Final Report Details").Range("L" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("U" & CStr(i)).Text
'Cosmic
Sheets("Final Report Details").Range("M" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("N" & CStr(VCRow)).Text
'Db SNP
Sheets("Final Report Details").Range("N" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("R" & CStr(i)).Text
CRow = CRow + 1
'Move to next mutation in NG
Exit For
End If
End If
Next
VCRow = VCRow + 1
Wend
End Sub
Attribute VB_Name = "Module16"
Attribute VB_Name = "Module17"
Sub pre_format_chrpos_ng()
```

```
'Adds a column to NG data and populates it with the chromosome and position separated by a comma
'Create Column for Chr,Position
    Range("G1").Select
    Selection.EntireColumn.Select
    Selection.Insert Shift:=xlToRight
'Add column title
Range("G5").Select
Selection.FormulaR1C1 = "Chr,Position"
'For all Nextgene data
For i = 6 To Range("B6").End(xlDown).Row
'Populate with chromosome from column F and position from column D
Range("G" & CStr(i)).Select
ActiveCell.Formula = "=CONCATENATE(F" & CStr(i) & "," & """,""" & ",D" & CStr(i) & ")"
Next i
End Sub
Sub pre_format_chrpos_vc()
'Adds a column to VC data and populates it with the chromosome and position separated by a comma
'Create Column for Chr,Position
Range("D1").Select
Selection.EntireColumn.Select
Selection.Insert Shift:=xlToRight
'Add column title
Range("D5").Select
Selection.FormulaR1C1 = "Chr,Position"
'For all Variant caller data
For i = 6 To Range("B6").End(xlDown).Row
'Populate with chromosome from column F and position from column D
Range("D" & CStr(i)).Select
ActiveCell.Formula = "=CONCATENATE(LOOKUP(99^99,--(""0""&MID(B" & CStr(i) &
",MIN(SEARCH({0,1,2,3,4,5,6,7,8,9},B6&""0123456789"")),ROW($1:$10000)))),"","",C" & CStr(i) & ")"
Next i
End Sub
Attribute VB_Name = "Module18"
Sub post_det_find_vc()
'Finds variants that are marked as TRUE in only variant caller
Sheets("Variant Caller").Select
'VARIABLES
Dim VCRow As Integer
    'line of data in variant caller
Dim CRow As Integer
    'row of Final Report to paste
Dim Matches As Integer
    'used to determine whether mutation is already in common findings list
    'matches = 1 if already in list
    'matches is empty if not found in list
'Next empty row of Final Report; initially 20
CRow = 20
'Start search at row 6 in VC
VCRow = 6
'For each row of data in variant caller
While Len(Sheets("Variant Caller").Range("C" & VCRow).Text) > 0
'If marked as a TRUE mutation in variant caller
If Sheets("Variant Caller").Range("A" & CStr(VCRow)).Value = True Then
'For each common variant listed in Final report
'Check position against this list to make sure that it is only found in Variant Caller
For i = 4 To 18
'If position matches mutation already in list (ie. Common Findings)
If Sheets("Variant Caller").Range("C" & CStr(VCRow)).Value = Sheets("Final Report
Details").Range("C" & CStr(i)).Value Then
Matches = 1
'Exit search, move to next Variant Caller mutation
Exit For
Else
'do nothing, check next position
End If
Next i
'If mutation was found in common findings list
If Matches = 1 Then
'do nothing
'If not found in list and TRUE mutation
Else
```

```
'Transfer data to Final Report under Variant Caller Findings
'Review
Sheets("Final Report Details").Select
Sheets("Final Report Details").Range("A" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("A" & CStr(VCRow)).Text
'Chromosome
Sheets("Final Report Details").Range("B" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("B" & CStr(VCRow)).Text
'Position
Sheets("Final Report Details").Range("C" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("C" & CStr(VCRow)).Text
'GeneID
Sheets("Final Report Details").Range("D" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("O" & CStr(VCRow)).Text
'Total Coverage VC
Sheets("Final Report Details").Range("E" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("T" & CStr(VCRow)).Text
'Total Coverage NG
Sheets("Final Report Details").Range("F" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Quality/Score
Sheets("Final Report Details").Range("G" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("J" & CStr(VCRow)).Text
'Strand Bias
Sheets("Final Report Details").Range("H" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("AD" & CStr(VCRow)).Text
'Frequency VC
Sheets("Final Report Details").Range("I" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("I" & CStr(VCRow)).Text
'Frequency NG
Sheets("Final Report Details").Range("J" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Mutation Call
Sheets("Final Report Details").Range("K" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Amino Acid Change
Sheets("Final Report Details").Range("L" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Cosmic
Sheets("Final Report Details").Range("M" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Variant Caller").Range("N" & CStr(VCRow)).Text
'Db SNP
Sheets("Final Report Details").Range("N" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
CRow = CRow + 1
End If
End If
VCRow = VCRow + 1
Matches = 0
Wend
End Sub
Attribute VB_Name = "Module19"
Sub post_det_find_ng()
'Finds variants that are marked as TRUE in only Nextgene
Sheets("Nextgene").Select
'VARIABLES
Dim NGRow As Integer
    'line of data in Nextgene
Dim CRow As Integer
    'row of Final Report to paste
Dim Matches As Integer
    'used to determine whether mutation is already in common findings list
    'matches = 1 if already in list
    'matches is empty if not found in list
'Next empty row of Final Report; initially 28
CRow = 24
'Start search at row 6 in NG
NGRow = 6
'For each row of data in Nextgene
While Len(Sheets("Nextgene").Range("C" & NGRow).Text) > 0
'If marked as a TRUE mutation in Nextgene
```

```
If Sheets("Nextgene").Range("A" & CStr(NGRow)).Value = True Then
'For each common variant listed in Final report
'Check position against this list to make sure that it is only found in Nextgene
For i = 4 To 18
'If position matches mutation already in list (ie. Common Findings)
If Sheets("Nextgene").Range("D" & CStr(NGRow)).Value = Sheets("Final Report Details").Range("C" &
CStr(i)).Value Then
Matches = 1
'Exit search, move to next Nextgene mutation
Exit For
Else
'do nothing, check next position
End If
Next i
'If mutation was found in common findings list
If Matches = 1 Then
'do nothing
'If not found in list and TRUE mutation
Else
'Transfer data to Final Report under Nextgene Findings
'Review
Sheets("Final Report Details").Select
Sheets("Final Report Details").Range("A" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("A" & CStr(NGRow)).Text
'Chromosome
Sheets("Final Report Details").Range("B" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("F" & CStr(NGRow)).Text
'Position
Sheets("Final Report Details").Range("C" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("D" & CStr(NGRow)).Text
'GeneID
Sheets("Final Report Details").Range("D" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("E" & CStr(NGRow)).Text
'Total Coverage VC
Sheets("Final Report Details").Range("E" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Total Coverage NG
Sheets("Final Report Details").Range("F" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("J" & CStr(NGRow)).Text
'Quality/Score
Sheets("Final Report Details").Range("G" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("K" & CStr(NGRow)).Text
'Strand Bias
Sheets("Final Report Details").Range("H" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency VC
Sheets("Final Report Details").Range("I" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency NG
Sheets("Final Report Details").Range("J" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("T" & CStr(NGRow)).Text
'Mutation Call
Sheets("Final Report Details").Range("K" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("S" & CStr(NGRow)).Text
'Amino Acid Change
Sheets("Final Report Details").Range("L" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("U" & CStr(NGRow)).Text
'Cosmic
Sheets("Final Report Details").Range("M" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Db SNP
Sheets("Final Report Details").Range("N" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("R" & CStr(NGRow)).Text
CRow = CRow + 1
End If
End If
NGRow = NGRow + 1
Matches = 0
Wend
End Sub
Attribute VB_Name = "Module2"
```

```
Attribute VB_Name = "Module20"
Sub post_det_find_snps()
'Finds variants that are marked as SNP-Missense or SNP-Silent in Nextgene
Sheets("Nextgene").Select
'VARIABLES
Dim NGRow As Integer
    'line of data in Nextgene
Dim CRow As Integer
    'row of Final Report to paste
'Next empty row of Final Report; initially 28
CRow = 4
'Start search at row 6 in NG
NGRow = 6
'For each row of data in Nextgene
While Len(Sheets("Nextgene").Range("C" & NGRow).Text) > 0
'If marked as a TRUE mutation in Nextgene
If Sheets("Nextgene").Range("A" & CStr(NGRow)).Value = "SNP-MISSENSE" Or
Sheets("Nextgene").Range("A" & CStr(NGRow)).Value = "SNP-SILENT" Or Sheets("Nextgene").Range("A"
& CStr(NGRow)).Value = "SNP-INTRON" Then
'Transfer data to Final Report under Nextgene Findings
'Review
Sheets("Final Report SNPs").Select
Sheets("Final Report SNPs").Range("A" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("A" & CStr(NGRow)).Text
'Chromosome
Sheets("Final Report SNPs").Range("B" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("F" & CStr(NGRow)).Text
'Position
Sheets("Final Report SNPs").Range("C" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("D" & CStr(NGRow)).Text
'GeneID
Sheets("Final Report SNPs").Range("D" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("E" & CStr(NGRow)).Text
'Total Coverage VC
Sheets("Final Report SNPs").Range("E" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Total Coverage NG
Sheets("Final Report SNPs").Range("F" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("J" & CStr(NGRow)).Text
'Quality/Score
Sheets("Final Report SNPs").Range("G" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("K" & CStr(NGRow)).Text
'Strand Bias
Sheets("Final Report SNPs").Range("H" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency VC
Sheets("Final Report SNPs").Range("I" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency NG
Sheets("Final Report SNPs").Range("J" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("T" & CStr(NGRow)).Text
'Mutation Call
Sheets("Final Report SNPs").Range("K" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("S" & CStr(NGRow)).Text
'Amino Acid Change
Sheets("Final Report SNPs").Range("L" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("U" & CStr(NGRow)).Text
'Cosmic
Sheets("Final Report SNPs").Range("M" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Db SNP
Sheets("Final Report SNPs").Range("N" & CStr(CRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("R" & CStr(NGRow)).Text
CRow = CRow + 1
End If
NGRow = NGRow + 1
Wend
End Sub
Attribute VB_Name = "Module21"
Sub POST_()
'On click, macro runs all functions included in "Post-Review" category
```

```
'This includes _find,
'The following allows the template to be saved as any file name without interrupting function of
macros
Dim strFileName As String
    'Define a string
strFileName = ActiveWorkbook.Name
    'Set string to name of the active workbook (the open analysis template file)
'MACROS
'Populate Final Report Details worksheet
Sheets("Final Report Details").Select
Application.Run "'" & strFileName & "'" & "!post_det_find_common"
Application.Run "'" & strFileName & "'" & "!post_det_find_vc"
Application.Run "'" & strFileName & "'" & "!post_det_find_ng"
Application.Run "'" & strFileName & "'" & "!post_det_find_snps"
Application.Run "'" & strFileName & "'" & "!post_det_no_mutations"
'Populate Final Report Summary worksheet
Sheets("Final Report Summary").Select
'Summary of detected mutations
Application.Run "'" & strFileName & "'" & "!post_sum_mutations"
'ROI Table
Application.Run "'" & strFileName & "'" & "!post_sum_roi_table"
Application.Run "'" & strFileName & "'" & "!post_sum_roi_coverage"
Application.Run "'" & strFileName & "'" & "!post_sum_roi_mutant"
'MRD Note
Application.Run "'" & strFileName & "'" & "!post_sum_mrd"
'Patient Identity Check
Application.Run "'" & strFileName & "'" & "!post_sum_pat_ide_check"
Range("A1").Select
'QM database
'Application.Run "'" & strFileName & "'" & "!post_qm_sort"
Application.Run "'" & strFileName & "'" & "!post_qm_hide"
Range("A1").Select
'Highlight ROI positions in Nextgene and Variant Caller data
Sheets("Variant Caller").Select
Application.Run "'" & strFileName & "'" & "!post_highlight_roi_vc"
Sheets("Nextgene").Select
Application.Run "'" & strFileName & "'" & "!post_highlight_roi_ng"
'Add footers to all pages that will be printed
Sheets("Nextgene").Select
Application.Run "'" & strFileName & "'" & "!post_print_footer"
Sheets("Variant Caller").Select
Application.Run "'" & strFileName & "'" & "!post_print_footer"
Sheets("Final Report Summary").Select
Application.Run "'" & strFileName & "'" & "!post_print_footer"
Sheets("Final Report Details").Select
Application.Run "'" & strFileName & "'" & "!post_print_footer"
Sheets("Final Report SNPs").Select
Application.Run "'" & strFileName & "'" & "!post_print_footer"
Sheets("Variant Caller").Select
Application.Run "'" & strFileName & "'" & "!post_format_columns_vc"
Sheets("Nextgene").Select
Application.Run "'" & strFileName & "'" & "!post_format_columns_ng"
Sheets("Final Report Summary").Select
Range("A1").Select
End Sub
Attribute VB_Name = "Module22"
Sub post_sum_roi_table()
'Compiles ROI table on Final Report Summary page
'based on Tumor Type chosen on Demographics page
Sheets("Final Report Summary").Select
'Dim Variables
Dim TableStart As Integer
Dim TumorType As String
Dim TumorColumn As Integer
Dim LastColumn As Integer
'Identify tumor type
'TumorType is chosen by Demographics tab
TumorType = Sheets("Demographics").Range("B14").Text
'Identify the row for the given tumor type in the ROI database
IdentifyRow:
'For each tumor type in ROI database
```

```
Sheets("ROI Database").Select
For i = 8 To Range(Range("H8"), Selection.End(xlToRight)).End(xlToRight).Column
If Sheets("ROI Database").Cells(1, i).Text = TumorType Then
TumorColumn = i
Exit For
Else
End If
Next i
'If TumorRow variable is still equal to 0 (ie. tumor type not in ROI database)
If TumorColumn = 0 Then
'Automatically consider as "Other" tumor type
TumorType = "Other"
GoTo IdentifyRow
Else
End If
'For each position in ROI database
For i = 2 To Sheets("ROI Database").Range("A1").End(xlDown).Row
'If the ROI is to be included for particular tumor type
'Identified by an "x" in the table
If Len(Sheets("ROI Database").Cells(i, TumorColumn).Text) > 0 Then
'Copy database row- from column A to column F
Sheets("ROI Database").Select
Range("A1").Select
LastColumn = Range("A1", Selection.End(xlToRight)).End(xlToRight).Column
Sheets("ROI Database").Range(Cells(i, 1), Cells(i, LastColumn)).Select
'Range("A" & CStr(i) & ":" & "F" & CStr(i)).Select
Selection.Copy
'Paste on Final Report Sumamry Page
Sheets("Final Report Summary").Select
'Row to start table, two rows down from last row of data
TableStart = Range("A" & Rows.Count).End(xlUp).Row + 1
'Start compiling table at TableStart Row below "Pertinent Negatives" title
Cells(TableStart, 1).Select
ActiveSheet.Paste
TableStart = TableStart + 1
End If
Next i
Dim MRDGene As String
Dim CodReg As String
Dim Chr As String
Dim Pos As String
Dim Exon As String
'MRD
For i = 22 To 31
Sheets("Demographics").Select
If IsEmpty(Range("B" & CStr(i))) Then
    Exit For
Else
MRDGene = Range("B" & CStr(i)).Text
CodReg = Range("C" & CStr(i)).Text
Chr = Range("D" & CStr(i)).Text
Pos = Range("E" & CStr(i)).Text
End If
Sheets("Final Report Summary").Select
'Check if already in ROI table
For j = 24 To Range("A24").End(xlDown).Row
'If genes match
If Range("A" & CStr(j)).Text = MRDGene Then
    'Check codons
'Numeric
    If IsNumeric(Range("D" & CStr(j)).Text) Then
        'If already in table, exit for
        If CodReg = "FS" Then
        GoTo AddToTable
        ElseIf Range("D" & CStr(j)).Text = CodReg Then
        Exit For
        Else
        GoTo AddToTable
        End If
    'Not numeric
    Else
```

```
        StartCodon = Left(Range("D" & CStr(j)).Text, InStr(1, Range("D" & CStr(j)).Text, "-") -
1)
        EndCodon = Right(Range("D" & CStr(j)).Text, Len(Range("D" & CStr(j)).Text) - InStr(1,
Range("D" & CStr(j)).Text, "-"))
        'If already in table, exit for
        If CodReg = "FS" Then
        GoTo AddToTable
        ElseIf CodReg >= CInt(StartCodon) And CodReg <= CInt(EndCodon) Then
        Exit For
        Else
AddToTable:
AddRow = Range("A24").End(xlDown).Row + 1
'Gene
Range("A" & CStr(AddRow)).Select
ActiveCell.FormulaR1C1 = MRDGene
'Chr, Position
Range("B" & CStr(AddRow)).Select
ActiveCell.FormulaR1C1 = Chr & "," & Pos
'Exon
Range("C" & CStr(AddRow)).Select
ActiveCell.FormulaR1C1 = "na"
'CodReg
Range("D" & CStr(AddRow)).Select
ActiveCell.FormulaR1C1 = CodReg
'Add Border
Range("A" & CStr(AddRow) & ":F" & CStr(AddRow)).Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
Exit For
End If
    End If
'If last j; not found in table
ElseIf j = Range("A24").End(xlDown).Row Then
GoTo AddToTable
End If
```

```
Next j
Next i
Sheets("Final Report Summary").Select
'Add drop down for result
'For each position in ROI table
For i = 24 To Sheets("Final Report Summary").Range("A24").End(xlDown).Row
Sheets("Final Report Summary").Range("F" & CStr(i)).Select
'Data validation; use dropdown list
'references list on Demographic Options tab
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=ROI_result"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
Next i
End Sub
Attribute VB_Name = "Module23"
Sub pre_prepare_ng()
'Deletes unneeded data columns from NG report
    Range("E1:I1").Select
    Selection.EntireColumn.Select
    Selection.Delete Shift:=xlToLeft
    Range("F1:G1").Select
    Selection.EntireColumn.Select
    Selection.Delete Shift:=xlToLeft
    Range("Q1").Select
    Selection.EntireColumn.Select
    Selection.Delete Shift:=xlToLeft
    Range("T1").Select
    Selection.EntireColumn.Select
    Selection.Delete Shift:=xlToLeft
End Sub
Attribute VB_Name = "Module24"
Sub medi_report_1_result()
'Adds summary of result to report
'Positive, Negative, and Other Findings
'Define Variables
Dim MyRow As Integer
'last row of data plus 2
'target row for compiling data
Dim TumText As String
'string of tumor name
Dim TumRow As Integer
'row of ROI genes in database for tumor type
Dim Genes As String
'true genes
Dim TrueCounter As Integer
'number of true mutations
Dim GeneArray() As String
'positive genes
Dim NegGeneArray() As String
'negative genes
Dim OthGeneArray() As String
'other genes
'TEXT FOR REPORT
Dim OtherText As String
Dim SingularMutText As String
Dim PluralMutText As String
SingularMutText = " MUTATION WAS "
PluralMutText = " MUTATIONS WERE "
OtherText = "DETECTED (see NOTE and Interpretation)."
NegSingularMutText = " mutation was "
NegPluralMutText = " mutations were "
```

```
NegOtherText = "NOT detected (see NOTE and Interpretation)."
'Identify tumor from demographics
TumText = Sheets("Demographics").Range("B14").Text
'Find list of ROI genes in database
For i = 2 To Sheets("Tumor ROI Options").Range("A" & Rows.Count).End(xlUp).Row
If Sheets("Tumor ROI Options").Range("A" & CStr(i)).Text = TumText Then
TumRow = Sheets("Tumor ROI Options").Range("A" & CStr(i)).Row
Exit For
End If
Next i
'POSITIVE
'Set counter to 0
TrueCounter = 0
Sheets("Tumor ROI Options").Select
'Create string of gene names for TRUE mutations based on tumor type
For i = 4 To 26
    'For all ROI genes for the given tumor type
    For j = 2 To Sheets("Tumor ROI Options").Range(Range("A" & CStr(TumRow)),
Selection.End(xlToRight)).End(xlToRight).Column
    'If true mutation
    If Sheets("Final Report Details").Range("A" & CStr(i)).Value = True Then
    'And gene is important for tumor type
        If Sheets("Final Report Details").Range("D" & CStr(i)).Text = Sheets("Tumor ROI
Options").Cells(TumRow, j).Text Then
    'Add gene to the GeneArray string
    ReDim Preserve GeneArray(0 To TrueCounter) As String
    GeneArray(TrueCounter) = Sheets("Final Report Details").Range("D" & CStr(i)).Text
    'Add to the counter for a total of how many TRUE mutations
    TrueCounter = TrueCounter + 1
    Exit For
        End If
    'If not true mutation OR empty space, move to new row of Final Report
    Else
    Exit For
    End If
    Next j
Next i
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
'last row of data in report plus two
'this is the next row that data will be added to report
'leaves a space between rows
'Strings of genes for meditech text
'Based off of elements of GeneArray
Dim GeneStr As String
Dim GeneStrFinal As String
'If No true mutations; GeneStrFinal = ""
If TrueCounter = 0 Then
GeneStrFinal = ""
'Move to next part of program, Negative
GoTo Negative
'If >1 TRUE mutation then alphabetize
Else
'Sort GeneArray alphabetically with "BubbleSort1" function
medi_report_alphabetize GeneArray
'Join all elements from GeneArray to form GeneStr, separate with comma and a space
For i = LBound(GeneArray) To UBound(GeneArray)
    GeneStr = Join(GeneArray, ", ")
    If Len(GeneStr) > 0 Then
    Exit For
    End If
Next
'Set GeneStrFinal to GeneStr after all genes have been added
GeneStrFinal = GeneStr
End If
'Add text to report
'If one TRUE gene (TrueCounter = 1)
If TrueCounter = 1 Then
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = GeneStrFinal & SingularMutText & OtherText
'If more than one TRUE gene (TrueCounter > 1)
```

```
ElseIf TrueCounter > 1 Then
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = GeneStrFinal & PluralMutText & OtherText
Else
End If
'Add positive text
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "POSITIVE - "
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'NEGATIVE
Negative:
Dim NegCounter As Integer
'counter for important genes that are not present in sample
NegCounter = 0
Dim DoesItMatch As Integer
'either 1 (if gene matches) or 0 (if gene does not match)
'For all billable genes for the given tumor type
Sheets("Tumor ROI Options").Select
Range("A" & CStr(TumRow)).Select
For i = 2 To Sheets("Tumor ROI Options").Range(Range("A" & CStr(TumRow)),
Selection.End(xlToRight)).End(xlToRight).Column
    'For each gene in GeneArray --> billable genes that are positive
    'Starts at 0 because of array indexing
    For j = 0 To (TrueCounter - 1)
'If billable gene matches positive gene--> IGNORE
    If Sheets("Tumor ROI Options").Cells(TumRow, i).Text = GeneArray(j) Then
        'Exit for; begin checking next billable gene
        DoesItMatch = 1
        Exit For
    'If billable gene does not match positive gene, continue checking other positive genes
    Else
    DoesItMatch = 0
    End If
    Next j
'After checking all positive genes against single billable gene
'If gene matches: do NOT include in negative text
If DoesItMatch = 1 Then
    'do nothing
'If gene does NOT match: include in negative text
ElseIf DoesItMatch = 0 Then
    'Add gene to NegGeneArray
    ReDim Preserve NegGeneArray(0 To NegCounter) As String
    NegGeneArray(NegCounter) = Sheets("Tumor ROI Options").Cells(TumRow, i).Text
    NegCounter = NegCounter + 1
End If
'Run next billable gene
Next i
'Strings of genes for meditech text
'Based off of elements of NegGeneArray
Dim NegGeneStr As String
Dim NegGeneStrFinal As String
'If No negative mutations; GeneStrFinal = ""
If NegCounter = 0 Then
NegGeneStrFinal = ""
'If >1 true mutation then alphabetize
Else
'Sort GeneArray alphabetically with "BubbleSort1" function
medi_report_alphabetize NegGeneArray
'Join all elements from NegGeneArray to form NegGeneStr, separate with comma and a space
For i = LBound(NegGeneArray) To UBound(NegGeneArray)
    NegGeneStr = Join(NegGeneArray, ", ")
    If Len(NegGeneStr) > 0 Then
    Exit For
    End If
Next
'Set NegGeneStrFinal to NegGeneStr after all genes have been added
NegGeneStrFinal = NegGeneStr
End If
'Add text to report
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
```

```
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "NEGATIVE - "
'If one negative mutation gene (TrueCounter = 1)
If NegCounter = 1 Then
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = NegGeneStrFinal & NegSingularMutText & NegOtherText
'If more than one negative mutation (TrueCounter > 1)
ElseIf NegCounter > 1 Then
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = NegGeneStrFinal & NegPluralMutText & NegOtherText
Else
End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'OTHER FINDINGS
'Set counter to 0
TrueCounter = 0
DoesItMatch = 0
Sheets("Tumor ROI Options").Select
'Create string of gene names for TRUE mutations based on tumor type
For i = 4 To 26
    'For all ROI genes for the given tumor type
    For j = 2 To Sheets("Tumor ROI Options").Range(Range("A" & CStr(TumRow)),
Selection.End(xlToRight)).End(xlToRight).Column
    'If true mutation
    If Sheets("Final Report Details").Range("A" & CStr(i)).Value = True Then
        'And gene is important for tumor type
        If Sheets("Final Report Details").Range("D" & CStr(i)).Text = Sheets("Tumor ROI
Options").Cells(TumRow, j).Text Then
        DoesItMatch = 1
        End If
    'If not true mutation OR empty space, move to new row of Final Report
    Else
    Exit For
    End If
    Next j
'If true mutation and not one of billable genes
    If Sheets("Final Report Details").Range("A" & CStr(i)).Value = True And DoesItMatch = 0 Then
    'Add gene as an "other finding" to OthGeneArray
    ReDim Preserve OthGeneArray(0 To TrueCounter) As String
    OthGeneArray(TrueCounter) = Sheets("Final Report Details").Range("D" & CStr(i)).Text
    'Add to the counter for a total of how many TRUE mutations
    TrueCounter = TrueCounter + 1
    End If
'Reset DoesItMatch
DoesItMatch = 0
Next i
'If no true mutations for other findings (OthGeneArray is empty)
If TrueCounter = 0 Then
    'End the Sub
    GoTo Finish
Else
Dim OthGeneStr As String
Dim OthGeneStrFinal As String
'Alphabetize gene array
medi_report_alphabetize OthGeneArray
'Create string of genes for report from OthGeneArray
For i = LBound(OthGeneArray) To UBound(OthGeneArray)
    OthGeneStr = Join(OthGeneArray, ", ")
    If Len(OthGeneStr) > 0 Then
    Exit For
    End If
Next
'Set OthGeneStrFinal equal to GeneStr
OthGeneStrFinal = OthGeneStr
'Add text to report
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "OTHER FINDINGS - "
'If one TRUE gene (TrueCounter = 1)
If TrueCounter = 1 Then
```

```
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = OthGeneStrFinal & SingularMutText & OtherText
'If more than one TRUE gene (TrueCounter > 1)
ElseIf TrueCounter > 1 Then
Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = OthGeneStrFinal & PluralMutText & OtherText
Else
End If
End If
Finish:
Sheets("Meditech Report").Select
End Sub
Sub medi_report_alphabetize(ByRef pvarArray As Variant)
'Takes array pvarArray and alphabetizes elements
    Dim i As Long
    Dim iMin As Long
    Dim iMax As Long
    Dim varSwap As Variant
    Dim blnSwapped As Boolean
    iMin = LBound(pvarArray)
    'lower bound of array, first element
    iMax = UBound(pvarArray) - 1
    'upper bound of array, last element
    Do
        blnSwapped = False
        For i = iMin To iMax
            'If element (i) comes after (i+1) in the alphabet
            If pvarArray(i) > pvarArray(i + 1) Then
            'Swap order in array
                varSwap = pvarArray(i)
                pvarArray(i) = pvarArray(i + 1)
                pvarArray(i + 1) = varSwap
                blnSwapped = True
            End If
        Next
        iMax = iMax - 1
    Loop Until Not blnSwapped
End Sub
Attribute VB_Name = "Module25"
Sub post_det_no_mutations()
'Makes a note on Final Report Details page to identify that the sample has been reviewed and no
mutations found
Sheets("Final Report Details").Select
'No common findings
If Len(Range("A4").Text) = 0 Then
Range("A4").Select
Selection.FormulaR1C1 = "No common findings to report."
End If
'No variant caller findings
If Len(Range("A20").Text) = 0 Then
Range("A20").Select
Selection.FormulaR1C1 = "No Variant Caller findings to report."
End If
'No nextgene findings
If Len(Range("A24").Text) = 0 Then
Range("A24").Select
Selection.FormulaR1C1 = "No NextGENE findings to report."
End If
End Sub
Attribute VB_Name = "Module26"
Attribute VB_Name = "Module27"
Sub post_sum_mutations()
'Creates a summary of detected mutations of Final Report Summary page
'Common Findings String
Dim CFStr As String
'Nextgene String
Dim NGStr As String
'Variant Caller String
Dim VCStr As String
Dim Gene As String
Dim Mutation As String
```

```
'True counters
Dim CFTrue As Integer
Dim NGTrue As Integer
Dim VCTrue As Integer
CFTrue = 0
NGTrue = 0
VCTrue = 0
'Common Findings
For i = 4 To 18
If Len(Sheets("Final Report Details").Range("D" & CStr(i)).Text) > 0 Then
Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
Mutation = Sheets("Final Report Details").Range("L" & CStr(i)).Text
CFTrue = CFTrue + 1
If CFTrue > 1 Then
CFStr = CFStr & ", " & Gene & " (" & Mutation & ")"
Else
CFStr = Gene & " (" & Mutation & ")"
End If
End If
Next i
'Variant Caller
For i = 20 To 22
If Len(Sheets("Final Report Details").Range("D" & CStr(i)).Text) > 0 Then
Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
VCTrue = VCTrue + 1
If VCTrue > 1 Then
VCStr = VCStr & ", " & Gene
Else
VCStr = Gene
End If
End If
Next i
'Nextgene
For i = 24 To 26
If Len(Sheets("Final Report Details").Range("D" & CStr(i)).Text) > 0 Then
Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
Mutation = Sheets("Final Report Details").Range("L" & CStr(i)).Text
NGTrue = NGTrue + 1
If NGTrue > 1 Then
NGStr = NGStr & ", " & Gene & " (" & Mutation & ")"
Else
NGStr = Gene & " (" & Mutation & ")"
End If
End If
Next i
' Add mutation summary to Final Report Summary page
'COMMON FINDINGS
Sheets("Final Report Summary").Select
Range("B17").Select
'If more than one mutation
If CFTrue > 1 Then
ActiveCell.FormulaR1C1 = CFStr & " mutations were detected by Variant Caller and Nextgene."
'If one mutation
ElseIf CFTrue = 1 Then
ActiveCell.FormulaR1C1 = CFStr & " mutation was detected by Variant Caller and Nextgene."
'If no mutations
ElseIf CFTrue = 0 Then
ActiveCell.FormulaR1C1 = "No mutations were detected by both Variant Caller and Nextgene."
End If
'VARIANT CALLER
Range("B18").Select
'If more than one mutation
If VCTrue > 1 Then
ActiveCell.FormulaR1C1 = VCStr & " mutations were detected by Variant Caller."
'If one mutation
ElseIf VCTrue = 1 Then
ActiveCell.FormulaR1C1 = VCStr & " mutation was detected by Variant Caller."
'If common findings
ElseIf VCTrue = 0 And CFTrue >= 1 Then
ActiveCell.FormulaR1C1 = "See common findings."
'If no mutations
```

```
ElseIf VCTrue = 0 And CFTrue < 1 Then
ActiveCell.FormulaR1C1 = "No mutations were detected by Variant Caller."
End If
'Nextgene
Range("B19").Select
'If more than one mutation
If NGTrue > 1 Then
ActiveCell.FormulaR1C1 = NGStr & " mutations were detected by Nextgene."
'If one mutation
ElseIf NGTrue = 1 Then
ActiveCell.FormulaR1C1 = NGStr & " mutation was detected by Nextgene."
'If common findings
ElseIf NGTrue = 0 And CFTrue >= 1 Then
ActiveCell.FormulaR1C1 = "See common findings."
'If no mutations
ElseIf NGTrue = 0 And CFTrue < 1 Then
ActiveCell.FormulaR1C1 = "No mutations were detected by Nextgene."
End If
End Sub
Attribute VB_Name = "Module28"
Sub setup_roi_database()
Sheets("ROI Database (2)").Select
For i = 2 To 100
For j = 2 To 123
If Sheets("ROI").Range("C" & CStr(i)).Text = Sheets("ROI Database (2)").Range("B" & CStr(j)).Text
Then
Range("P" & CStr(j)).Select
ActiveCell.FormulaR1C1 = "x"
End If
Next j
Next i
Sheets("ROI").Select
End Sub
Attribute VB_Name = "Module29"
Sub post_sum_roi_coverage()
'Completes "ROI manually reviewed (depth) x" column
'of ROI table on Final Report Summary page
'Uses allele coverage to find depth
Dim j As Long
'If data is present in allele coverage report
If Len(Sheets("Allele Coverage").Range("B1").Value) > 0 Then
'Identifies chromosome position from ROI table
Dim lngRight As Long, lngCommaPos As Long
Dim intI As Integer
    For intI = 24 To Sheets("Final Report Summary").Range("A24").End(xlDown).Row
        lngCommaPos = InStr(1, Range("B" & intI).Value, ",")
        lngRight = Right(Range("B" & intI).Value, Len(Range("B" & intI).Value) - lngCommaPos)
'For all data in Allele Coverage report
For j = 2 To Sheets("Allele Coverage").Range("B2").End(xlDown).Row
'If position matches position in table
If Sheets("Allele Coverage").Range("B" & CStr(j)).Value = lngRight Then
'Add coverage at that spot
Sheets("Final Report Summary").Range("E" & CStr(intI)).Select
ActiveCell.FormulaR1C1 = Sheets("Allele Coverage").Range("R" & CStr(j)).Text
j = Sheets("Allele Coverage").Range("B2").End(xlDown).Row
'Else if the coverage position is within 5 of the position in table
ElseIf Abs(Sheets("Allele Coverage").Range("B" & CStr(j)).Value - lngRight) <= 5 Then
'Add coverage at that spot
Sheets("Final Report Summary").Range("E" & CStr(intI)).Select
ActiveCell.FormulaR1C1 = Sheets("Allele Coverage").Range("R" & CStr(j)).Text
j = Sheets("Allele Coverage").Range("B2").End(xlDown).Row
'Else if the coverage position is within 10 of the position in table
ElseIf Abs(Sheets("Allele Coverage").Range("B" & CStr(j)).Value - lngRight) <= 10 Then
Sheets("Final Report Summary").Range("E" & CStr(intI)).Select
ActiveCell.FormulaR1C1 = Sheets("Allele Coverage").Range("R" & CStr(j)).Text
j = Sheets("Allele Coverage").Range("B2").End(xlDown).Row
'Else if the coverage position is within 20 of the position in table
ElseIf Abs(Sheets("Allele Coverage").Range("B" & CStr(j)).Value - lngRight) <= 20 Then
Sheets("Final Report Summary").Range("E" & CStr(intI)).Select
ActiveCell.FormulaR1C1 = Sheets("Allele Coverage").Range("R" & CStr(j)).Text
j = Sheets("Allele Coverage").Range("B2").End(xlDown).Row
```

```
'Else if the coverage position is within 30 of the position in table
ElseIf Abs(Sheets("Allele Coverage").Range("B" & CStr(j)).Value - lngRight) <= 30 Then
Sheets("Final Report Summary").Range("E" & CStr(intI)).Select
ActiveCell.FormulaR1C1 = Sheets("Allele Coverage").Range("R" & CStr(j)).Text
j = Sheets("Allele Coverage").Range("B2").End(xlDown).Row
End If
Next j
    Next intI
'If no data present, exit Sub
Else
Exit Sub
End If
End Sub
Sub post_sum_roi_wild_or_mut()
'Completes "wildtype or mutant" column of ROI table on Final Report Summary page
'Based on the detected mutations in Final Report Details page
'Define variables
Dim i As Integer
Dim MediResult As Integer
Dim SearchNum As Integer
Dim StartRange As Integer
Dim EndRange As Integer
Dim CodonNum As Integer
Dim SearchforCodon As Integer
Dim CHalfway As Integer
Dim NoIVS As String
Dim JustNum As String
Dim CodonNumDouble As Double
Sheets("Final Report Summary").Select
'For all positions in ROI table
For i = 24 To Range("A24").End(xlDown).Row
'For all mutations on Final Report Details page
For j = 4 To 26
'If already mutant then skip
Sheets("Final Report Summary").Range("F" & CStr(i)).Select
If ActiveCell.FormulaR1C1 = "MUTANT" Then
'If genes match
ElseIf Trim(Sheets("Final Report Summary").Range("A" & CStr(i)).Text) = Trim(Sheets("Final Report
Details").Range("D" & CStr(j)).Text) Then
    'If codon is numeric
    If IsNumeric(Sheets("Final Report Summary").Range("D" & CStr(i))) Then
        'If codon matches codon in final report
        SearchNum = Sheets("Final Report Summary").Range("D" & CStr(i)).Value
        If InStr(1, Sheets("Final Report Details").Range("L" & CStr(j)), SearchNum) > 0 Then
            'Change text to MUTANT
            Sheets("Final Report Summary").Select
            Sheets("Final Report Summary").Range("F" & CStr(i)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
        'If IVS
        ElseIf Left(Sheets("Final Report Details").Range("K" & CStr(j)).Text, 3) = "IVS" Then
        'Text without IVS
        NoIVS = Right(Sheets("Final Report Details").Range("K" & CStr(j)).Text, Len(Sheets("Final
Report Details").Range("K" & CStr(j)).Text) - 3)
            'If -
            If InStr(NoIVS, "-") > 0 Then
                JustNum = Left(NoIVS, InStr(NoIVS, "-") - 1)
            'ElseIf +
            ElseIf InStr(NoIVS, "+") > 0 Then
                JustNum = Left(NoIVS, InStr(NoIVS, "+") - 1)
            End If
       CodonNumDouble = JustNum / 3
       CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
       If SearchNum = CodonNum Then
            'Change text to MUTANT
            Sheets("Final Report Summary").Select
            Sheets("Final Report Summary").Range("F" & CStr(i)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
        End If
        'If codon does not match codon in final report
        Else
        Sheets("Final Report Details").Select
```

```
        'Define what to search for
            'CHalfway is the halfway point of the codons (where the beginning and ending number
split)
            CHalfway = Len(GetNums(Range("K" & CStr(j)))) / 2
            'If CHalfway is an integer (ie starting and ending point have same number of digits)
            If Int(Len(CHalfway)) / Len(CHalfway) = 1 Then
            '
                'SearchforCodon is the start number of the mutation
                SearchforCodon = Left(GetNums(Range("K" & CStr(j))), CHalfway)
            'If CHalfway is a decimal
            Else
                'SearchforCodon is the start number of the mutation (number with fewer digits)
                SearchforCodon = Left(GetNums(Range("K" & CStr(j))), Int(CHalfway))
            End If
        'Match SearchNum to SearchforCodon
        SearchNum = Sheets("Final Report Summary").Range("D" & CStr(i)).Value
        CodonNumDouble = SearchforCodon / 3
        CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
        If SearchNum = CodonNum / 3 Then
        'Change text to MUTANT
        Sheets("Final Report Summary").Select
        Sheets("Final Report Summary").Range("F" & CStr(i)).Select
        ActiveCell.FormulaR1C1 = "MUTANT"
        End If
        End If
    'If codon isn't numeric
    Else
        'If codon range includes codon in Final Report Summary
        If IsNumeric(Sheets("Meditech Data").Range("H" & CStr(i + 5)).Value) Then
        StartRange = Sheets("Meditech Data").Range("H" & CStr(i + 5)).Value
        Else
        StartRange = 0
        End If
        If IsNumeric(Sheets("Meditech Data").Range("K" & CStr(i + 5)).Value) Then
        EndRange = Sheets("Meditech Data").Range("K" & CStr(i + 5)).Value
        Else
        EndRange = 0
        End If
            'If codon in final report is numeric
            If IsNumeric(GetNums(Sheets("Final Report Summary").Range("L" & CStr(j)))) Then
                CodonNum = GetNums(Sheets("Final Report Summary").Range("L" & CStr(j)))
            'If FS
            ElseIf Sheets("Final Report Summary").Range("L" & CStr(j)).Text = "FS" Then
            CodonNum = -100
            'If IVS
            ElseIf Left(Sheets("Final Report Summary").Range("K" & CStr(j)).Text, 3) = "IVS" Then
                'Text without IVS
                NoIVS = Right(Sheets("Final Report Summary").Range("K" & CStr(j)).Text,
Len(Sheets("Final Report Summary").Range("K" & CStr(j)).Text) - 3)
                'InStr(Sheets("Final Report Summary").Range("K" & CStr(j)).Text,"-")
                    'If -
                    If InStr(NoIVS, "-") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "-") - 1)
                    'ElseIf +
                    ElseIf InStr(NoIVS, "+") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "+") - 1)
                    End If
                CodonNumDouble = JustNum / 3
                CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
            Else
                'Define what to search for
                'CHalfway is the halfway point of the codons (where the beginning and ending
number split)
                CHalfway = Len(GetNums(Range("K" & CStr(j)))) / 2
                'If CHalfway is an integer (ie starting and ending point have same number of
digits)
                If Int(Len(CHalfway)) / Len(CHalfway) = 1 Then
                 '
                   'SearchforCodon is the start number of the mutation
                   CodonNum = Left(GetNums(Range("K" & CStr(j))), CHalfway) / 3
                'If CHalfway is a decimal
```

```
                Else
                    'SearchforCodon is the start number of the mutation (number with fewer
digits)
                    CodonNum = Application.WorksheetFunction.RoundUp(Left(GetNums(Range("K" &
CStr(j))), Int(CHalfway)) / 3, 0)
                End If
            End If
        If CodonNum >= StartRange And CodonNum <= EndRange Then
            'Change text to MUTANT
            Sheets("Final Report Summary").Select
            Sheets("Final Report Summary").Range("F" & CStr(i)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
        End If
    End If
Else
'Change text to Wildtype
'Sheets("Final Report Summary").Select
'Sheets("Final Report Summary").Range("F" & CStr(i)).Select
'ActiveCell.FormulaR1C1 = "Wildtype"
End If
Next j
Next i
End Sub
Attribute VB_Name = "Module3"
Sub pre_filter_snps_vc()
'Filters SNPs in variant caller data based on SNPs in template database
'SNP variants changed to green
'VARIABLES
Dim LSearchRow As Integer
    'row of data in variant caller data
Dim SNPCounter As Integer
    'last row of data in SNP database
'Start search in row 6
LSearchRow = 6
'For all of the data in variant caller
While Len(Sheets("Variant Caller").Range("C" & CStr(LSearchRow)).Value) > 0
'Define last row of data in SNP sheet
SNPCounter = Sheets("SNP Database").Range("A" & Rows.Count).End(xlUp).Row
'For each SNP in database
For i = 2 To SNPCounter
'If position in VC matches SNP position
If Sheets("Variant Caller").Range("C" & CStr(LSearchRow)).Value = Sheets("SNP
Database").Range("A" & CStr(i)).Value Then
'Change text in column A on VC to SNP Type
Sheets("Variant Caller").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("SNP Database").Range("G" & CStr(i)).Value
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
'If SNP, move to next row of data in variant caller
i = SNPCounter
End If
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub pre_filter_mutfreq_vc()
'Filters variants with mutation frequency below the given threshold.
'Threshold is determined by tumor percentage on Demographics page
'Low mutation frequency variants are changed to red
'VARIABLES
Dim LSearchRow As Integer
    'row of data in VC
Dim FreqThreshold As Integer
    'frequency threshold for specimen
'Start search in row 6
LSearchRow = 6
'Define frequency thresholds for given percent tumor
Sheets("Demographics").Select
FreqThreshold = Range("B16").Value
```

```
'For all of the data in variant caller
Sheets("Variant Caller").Select
While Len(Range("H" & CStr(LSearchRow)).Value) > 0
'If number in column H ("Frequency") is < FreqThreshold
If Range("H" & CStr(LSearchRow)).Value < FreqThreshold Then
'Then change text in column A ("Review") to <MUT FREQ THRESH
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "<MUT FREQ THRESH"
'And make entire row red
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 3
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub pre_filter_lowcov_vc()
'Filters variants with coverage below 500
'Low coverage variants are changed to orange
'VARIABLES
Dim LSearchRow As Integer
    'row of variant caller data
Dim CovThreshold As Integer
    'threshold for low coverage
Sheets("Variant Caller").Select
'Start search in row 6
LSearchRow = 6
'Define coverage threshold.
CovThreshold = 500
'For all of the variant caller data
While Len(Range("H" & CStr(LSearchRow)).Value) > 0
'If coverage is < CovThreshold
If Range("S" & CStr(LSearchRow)).Value < CovThreshold Then
'Then change text in column A to "< 500X COV"
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "<500X COV"
'And make entire row red
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 46
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub pre_filter_mutandcov_vc()
'Filters variants with coverage below 500 and low mutation frequency
'Variants are changed to pink
'VARIABLES
Dim LSearchRow As Integer
    'row of variant caller data
Dim FreqThreshold As Integer
    'frequency threshold determined by tumor percentage
Dim CovThreshold As Integer
    'coverage threshold
Sheets("Variant Caller").Select
'Start search in row 6
LSearchRow = 6
'Define coverage threshold.
CovThreshold = 500
'Define frequency thresholds for given percent tumor
Sheets("Demographics").Select
FreqThreshold = Range("B16").Value
'For all of the variant caller data
Sheets("Variant Caller").Select
While Len(Range("H" & CStr(LSearchRow)).Value) > 0
'If coverage is <500 and frequency is below threshold
If Range("S" & CStr(LSearchRow)).Value < CovThreshold And Range("H" & CStr(LSearchRow)).Value <
FreqThreshold Then
'Then change text in column A to "LOW FREQ & COV"
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "LOW FREQ & COV"
```

```
'And make entire row pink
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 26
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub pre_filter_artifacts_vc()
'Filters artifacts in variant caller data based on artifacts in template database
'Artifact variants changed to blue
Dim LSearchRow As Integer
    'row in variant caller data
Dim ArtCounter As Integer
    'final row in artifact database
'Start search in row 6
LSearchRow = 6
'For all variant caller data
While Len(Sheets("Variant Caller").Range("C" & CStr(LSearchRow)).Value) > 0
'Last row of data in Artifact database
ArtCounter = Sheets("Artifact Database").Range("C" & Rows.Count).End(xlUp).Row
'For each artifact in the database
For i = 2 To ArtCounter
'If variant position matches position of an artifact in the database
If Sheets("Variant Caller").Range("C" & CStr(LSearchRow)).Text = Sheets("Artifact
Database").Range("C" & CStr(i)).Text Then
'Change text in column A to type of Artifact (specified by "Details" column in database)
Sheets("Variant Caller").Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Artifact Database").Range("D" & CStr(i)).Text
'And make entire row blue
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 32
'If artifact, exit for and move to next row of data
Exit For
End If
'If no match, check variant against next artifact in database
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub PRE_()
'On click, macro runs all functions included in "Pre-Review" category
'This includes _filter, _format
'The following allows the template to be saved as any file name without interrupting function of
macros
Dim strFileName As String
    'Define a string
strFileName = ActiveWorkbook.Name
    'Set string to name of the active workbook (the open analysis template file)
'Error Messages
'Do not check for errors if Analyzed by "Rebecca"
    'Used for testing scripts
If Sheets("Demographics").Range("E8").Text = "Rebecca" Then
    'do nothing
Else
'DEMOGRAPHICS
Sheets("Demographics").Select
'No Patient Name
If IsEmpty(Sheets("Demographics").Range("B2")) Then
Range("B2").Select
MsgBox "Please enter the Patient Name."
SubError = 1
Exit Sub
End If
'No DMO Number
If IsEmpty(Sheets("Demographics").Range("B3")) Then
Range("B3").Select
MsgBox "Please enter the DMO Number."
SubError = 1
Exit Sub
End If
```

```
'No Analysis By
If IsEmpty(Sheets("Demographics").Range("E8")) Then
Range("E8").Select
MsgBox "Please complete the 'Analysis By' Field."
SubError = 1
Exit Sub
End If
'No Date
If IsEmpty(Sheets("Demographics").Range("E9")) Then
Range("E9").Select
MsgBox "Please enter the date."
SubError = 1
Exit Sub
End If
'No Tumor Percentage
If IsEmpty(Sheets("Demographics").Range("B15")) Then
Range("B15").Select
MsgBox "Please choose the tumor percentage."
SubError = 1
Exit Sub
End If
'No Tumor type
If IsEmpty(Sheets("Demographics").Range("B14")) Then
Range("B14").Select
MsgBox "Please select the tumor type."
SubError = 1
Exit Sub
End If
'No Run Number
If IsEmpty(Sheets("Demographics").Range("B8")) Then
Range("B8").Select
MsgBox "Please enter the PGM Run number."
SubError = 1
Exit Sub
End If
'AML
'No baseline info for AML
If Range("B14").Text = "AML" And IsEmpty(Sheets("Demographics").Range("B19")) Then
Range("B19").Select
MsgBox "Please complete the Baseline field."
SubError = 1
Exit Sub
End If
'No MRD info for AML (baseline = NO)
If Range("B14").Text = "AML" And IsEmpty(Sheets("Demographics").Range("B20")) Then
Range("B20").Select
MsgBox "Please complete the MRD field."
SubError = 1
Exit Sub
End If
'No positive genes listed/listed starting in wrong box
If Range("B14").Text = "AML" And Range("B20").Text = "Yes" And IsEmpty(Range("B22")) Then
Sheets("Demographics").Range("B22").Select
MsgBox "Please list positive MRD genes starting in cell B22."
SubError = 1
Exit Sub
End If
'No transplant status
If Range("B14").Text = "AML" And IsEmpty(Sheets("Demographics").Range("E19")) Then
Sheets("Demographics").Range("E19").Select
MsgBox "Please choose Transplant Status."
SubError = 1
Exit Sub
End If
'Check all MRD data
If Range("B14").Text = "AML" Then
For i = 22 To 31
If Len(Sheets("Demographics").Range("B" & CStr(i)).Text) > 0 Then
    If IsEmpty(Sheets("Demographics").Range("C" & CStr(i))) Or
IsEmpty(Sheets("Demographics").Range("D" & CStr(i))) Or IsEmpty(Sheets("Demographics").Range("E"
& CStr(i))) Then
```

```
    Sheets("Demographics").Range("C" & CStr(i)).Select
    MsgBox "Please complete all MRD fields."
    SubError = 1
    Exit Sub
    End If
End If
Next i
End If
'RAW DATA
'Allele Coverage
Sheets("Allele Coverage").Select
If IsEmpty(Sheets("Allele Coverage").Range("A1")) Then
Range("A1").Select
MsgBox "Please paste Allele Coverage data."
SubError = 1
Exit Sub
End If
'Nextgene
Sheets("Nextgene").Select
If IsEmpty(Sheets("Nextgene").Range("A1")) Then
Range("A1").Select
MsgBox "Please paste Nextgene data."
SubError = 1
Exit Sub
End If
'Make sure that variant caller is empty
Sheets("Variant Caller").Select
If IsEmpty(Sheets("Variant Caller").Range("A1")) Then
    'do nothing
Else
Range("A1").Select
MsgBox "Please delete all Variant Caller data and re-run Pre-Review."
SubError = 1
Exit Sub
End If
'Make sure that Nextgene Modified is empty
Sheets("Nextgene Modified").Select
If IsEmpty(Sheets("Nextgene Modified").Range("A1")) Then
    'do nothing
Else
Range("A1").Select
MsgBox "Please delete all Nextgene Modified data and re-run Pre-Review."
SubError = 1
Exit Sub
End If
'Coverage Analysis
Sheets("Coverage Analysis").Select
If IsEmpty(Sheets("Coverage Analysis").Range("A1")) Then
Range("A1").Select
MsgBox "Please paste Coverage Analysis data."
SubError = 1
Exit Sub
End If
'Sample ID Analysis
Sheets("Sample ID").Select
If IsEmpty(Sheets("Sample ID").Range("A1")) Then
Range("A1").Select
MsgBox "Please paste Sample ID data."
SubError = 1
Exit Sub
End If
End If
'MACROS
'Allele Coverage
Sheets("Allele Coverage").Select
Application.Run "'" & strFileName & "'" & "!pre_prepare_vc"
Range("A1").Select
'Variant Caller
Sheets("Variant Caller").Select
Application.Run "'" & strFileName & "'" & "!pre_format_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_snps_vc"
```

```
Application.Run "'" & strFileName & "'" & "!pre_filter_mutfreq_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_lowcov_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_mutandcov_vc"
Application.Run "'" & strFileName & "'" & "!pre_filter_artifacts_vc"
Application.Run "'" & strFileName & "'" & "!pre_format_sortbyfilter"
Application.Run "'" & strFileName & "'" & "!pre_format_chrpos_vc"
Application.Run "'" & strFileName & "'" & "!pre_format_dropdown_vc"
Application.Run "'" & strFileName & "'" & "!pre_format_color_vc"
Application.Run "'" & strFileName & "'" & "!pre_format_style"
Range("A1").Select
'Nextgene
Sheets("Nextgene").Select
Application.Run "'" & strFileName & "'" & "!pre_prepare_ng"
Application.Run "'" & strFileName & "'" & "!pre_format_ng"
Application.Run "'" & strFileName & "'" & "!pre_filter_snps_ng"
Application.Run "'" & strFileName & "'" & "!pre_filter_mutfreq_ng"
Application.Run "'" & strFileName & "'" & "!pre_filter_lowcov_ng"
Application.Run "'" & strFileName & "'" & "!pre_filter_mutandcov_ng"
Application.Run "'" & strFileName & "'" & "!pre_filter_artifacts_ng"
Application.Run "'" & strFileName & "'" & "!pre_format_sortbyfilter"
Application.Run "'" & strFileName & "'" & "!pre_format_chrpos_ng"
Application.Run "'" & strFileName & "'" & "!pre_format_dropdown_ng"
Application.Run "'" & strFileName & "'" & "!pre_format_color_ng"
Application.Run "'" & strFileName & "'" & "!pre_format_style"
Range("A1").Select
'Coverage Analysis
Sheets("Coverage Analysis").Select
Application.Run "'" & strFileName & "'" & "!pre_format_cov_an"
Range("A1").Select
'Go back to Final Report Summary page
Sheets("Final Report Summary").Select
Range("A1").Select
End Sub
Attribute VB_Name = "Module30"
Sub post_sum_pat_ide_check()
'Checks all barcodes, MD-numbers, and patient identifiers in data against the barcode from PGM
run
'Reports that identity check passed at bottom of Summary page
'BARCODES
Dim DemBar As String
    'demographics barcode
Dim VCBar As String
    'variant caller barcode
Dim NGBar As String
    'nextgene barcode
Dim SampBar As String
    'sample ID barcode
Dim CovBar As String
    'coverage analysis barcode
Dim CheckMatch As Boolean
Dim FailedTest As String
Dim Fails As Integer
'Identify all barcodes in data
DemBar = Sheets("Demographics").Range("B10").Text
VCBar = Sheets("Variant Caller").Range("AW6").Text
NGBar = Left(Sheets("Nextgene").Range("A1").Text, 13)
SampBar = Left(Sheets("Sample ID").Range("A3").Text, 13)
CovBar = Left(Sheets("Coverage Analysis").Range("A3").Text, 13)
'Check all data against barcode identified in Demographics
If DemBar = VCBar Then
Else
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Variant Caller Barcode"
    Fails = Fails + 1
    Else
    FailedTest = "Variant Caller Barcode"
    Fails = Fails + 1
    End If
End If
If DemBar = NGBar Then
Else
```

```
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Nextgene Barcode"
    Fails = Fails + 1
    Else
    FailedTest = "Nextgene Barcode"
    Fails = Fails + 1
    End If
End If
If DemBar = SampBar Then
Else
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Sample ID Barcode"
    Fails = Fails + 1
    Else
    FailedTest = "Sample ID Barcode"
    Fails = Fails + 1
    End If
End If
If DemBar = CovBar Then
Else
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Coverage Analysis Barcode"
    Fails = Fails + 1
    Else
    FailedTest = "Coverage Analysis Barcode"
    Fails = Fails + 1
    End If
End If
'MD NUMBERS
'Identify all MD numbers in data
Dim DemNum As String
Dim SampNum As String
Dim CovNum As String
DemNum = Sheets("Demographics").Range("B3").Text
SampNum = Right(Sheets("Sample ID").Range("A2").Text, Len(Sheets("Sample ID").Range("A2").Text) -
InStr(1, Sheets("Sample ID").Range("A2").Text, " M"))
CovNum = Right(Sheets("Coverage Analysis").Range("A2").Text, Len(Sheets("Coverage
Analysis").Range("A2").Text) - InStr(1, Sheets("Coverage Analysis").Range("A2").Text, " M"))
If DemNum = SampNum Then
Else
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Sample ID MD Number"
    Fails = Fails + 1
    Else
    FailedTest = "Sample ID MD Number"
    Fails = Fails + 1
    End If
End If
If DemNum = CovNum Then
Else
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Coverage Analysis MD Number"
    Fails = Fails + 1
    Else
    FailedTest = "Coverage Analysis MD Number"
    Fails = Fails + 1
    End If
End If
'PATIENT SEX
Dim DemSex As String
Dim DemSexAbbr As String
Dim SampSex As String
DemSex = Sheets("Demographics").Range("B5").Text
If DemSex = "Male" Then
DemSexAbbr = "M"
ElseIf DemSex = "Female" Then
DemSexAbbr = "F"
End If
SampSex = Left(Sheets("Sample ID").Range("A5").Text, 1)
If DemSexAbbr = SampSex Then
Else
```

```
    If Fails >= 1 Then
    FailedTest = FailedTest & ", Patient Sex"
    Fails = Fails + 1
    Else
    FailedTest = "Patient Sex"
    Fails = Fails + 1
    End If
End If
'Line for Final Report Summary
Sheets("Final Report Summary").Select
Dim MessageRow As Integer
MessageRow = Range("A" & Rows.Count).End(xlUp).Row + 2
Cells(MessageRow, 1).Select
If Fails = 0 Then
ActiveCell.FormulaR1C1 = "*All patient identity tests have passed."
Selection.Font.ColorIndex = 0
Selection.Font.Size = 15
Else
ActiveCell.FormulaR1C1 = "*The following patient identify tests have failed: " & FailedTest
Selection.Font.ColorIndex = 3
Selection.Font.Size = 15
End If
'Align text left in cell
With Selection
        .HorizontalAlignment = xlLeft
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
End Sub
Attribute VB_Name = "Module31"
Sub medi_sort(ByRef pvarArray As Variant)
    Dim i As Long
    Dim iMin As Long
    Dim iMax As Long
    Dim varSwap As Variant
    Dim blnSwapped As Boolean
    iMin = LBound(pvarArray)
    iMax = UBound(pvarArray) - 1
    Do
        blnSwapped = False
        For i = iMin To iMax
            If pvarArray(i) > pvarArray(i + 1) Then
                varSwap = pvarArray(i)
                pvarArray(i) = pvarArray(i + 1)
                pvarArray(i + 1) = varSwap
                blnSwapped = True
            End If
        Next
        iMax = iMax - 1
    Loop Until Not blnSwapped
End Sub
Attribute VB_Name = "Module32"
Sub SUPER_()
'Runs all macros associated with 1% supercontrol analysis
Dim strFileName As String
strFileName = ActiveWorkbook.Name
'MACROS
    'Allele coverage
    Sheets("1% Supercontrol VC").Select
    Application.Run "'" & strFileName & "'" & "!super_prepare_vc"
    'Variant caller
    Sheets("1% Supercontrol VC").Select
    Application.Run "'" & strFileName & "'" & "!super_format_vc"
    Application.Run "'" & strFileName & "'" & "!super_find_muts_vc"
    'Nextgene
```

```
    Sheets("1% Supercontrol NG").Select
    Application.Run "'" & strFileName & "'" & "!pre_prepare_ng"
    Application.Run "'" & strFileName & "'" & "!super_format_ng"
    Application.Run "'" & strFileName & "'" & "!super_find_muts_ng"
    'Compile Results
    Sheets("Supercontrol Results- Ampliseq").Select
    Application.Run "'" & strFileName & "'" & "!super_nomut_vc"
    Application.Run "'" & strFileName & "'" & "!super_nomut_ng"
    Application.Run "'" & strFileName & "'" & "!super_avg_freq"
    Application.Run "'" & strFileName & "'" & "!super_coverage"
    'Add title to supercontrol page
    Application.Run "'" & strFileName & "'" & "!super_addtitle"
End Sub
Sub super_prepare_vc()
'Compiles Variant Caller report from 1% Supercontrol Allele Coverage data
'All data with Allele Call of "Heterozygous" or "Homozygous" is added to report
'Go to 1% Supercontrol Allele Coverage data
Sheets("1% Supercontrol Allele Coverage").Select
Dim LSearchRow As Integer
    'Line of 1% Supercontrol Allele Coverage Data
Dim LCopyToRow As Integer
    'Line of Variant Caller Data
'Copy header to variant caller
Rows("1:1").Select
Selection.Copy
Sheets("1% Supercontrol VC").Select
Rows("1:1").Select
ActiveSheet.Paste
'Go back to 1% Supercontrol Allele Coverage data
Sheets("1% Supercontrol Allele Coverage").Select
'Start searching 1% Supercontrol Allele Coverage Data at Row 2
LSearchRow = 2
'Start copy data to Variant Caller at Row 2
LCopyToRow = 2
'For all of the 1% Supercontrol Allele Coverage data
While Len(Sheets("1% Supercontrol Allele Coverage").Range("A" & CStr(LSearchRow)).Text) > 0
'If text in column E is "Heterozygous" or "Homozygous"
If Sheets("1% Supercontrol Allele Coverage").Range("E" & CStr(LSearchRow)).Text = "Heterozygous"
Or Sheets("1% Supercontrol Allele Coverage").Range("E" & CStr(LSearchRow)).Text = "Homozygous"
Then
'Select row to copy
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).SpecialCells(xlCellTypeVisible).Select
Selection.Copy
'Paste row into Variant Caller in next empty row
Sheets("1% Supercontrol VC").Select
Rows(CStr(LCopyToRow) & ":" & CStr(LCopyToRow)).Select
ActiveSheet.Paste
'Move counter to next row
LCopyToRow = LCopyToRow + 1
'Go back to 1% Supercontrol Allele Coverage data and keep searching
Sheets("1% Supercontrol Allele Coverage").Select
End If
'Check next row of 1% Supercontrol Allele Coverage data
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub super_format_vc()
'Formats 1% Supercontrol VC report
  Rows("1:1").Select
    Selection.Insert Shift:=xlDown
    Rows("2:2").Select
    Selection.Insert Shift:=xlDown
    Selection.Insert Shift:=xlDown
    Selection.Insert Shift:=xlDown
    Range("A1").Select
    Selection.EntireColumn.Select
    Selection.Insert Shift:=xlToRight
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "Lot #"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "PGM Run #"
```

```
Range("A4").Select
ActiveCell.FormulaR1C1 = "Analysis By"
Range("B1").Select
Selection.EntireColumn.Select
Range("A1").Select
Selection.EntireColumn.AutoFit
Range("A2:B4").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .Weight = xlThin
    .ColorIndex = xlAutomatic
End With
```

```
    Range("B3:B4").Select
    With Selection.Interior
        .ColorIndex = 15
        .Pattern = xlSolid
    End With
    Range("B2").Select
    With Selection.Interior
        .ColorIndex = 15
        .Pattern = xlSolid
    End With
'Add cell references
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[7]C[6]"
    Range("B3").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[5]C"
    Range("B4").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[4]C[3]"
    Range("B5").Select
End Sub
Sub super_find_muts_vc()
'Finds mutations in VC 1% supercontrol data
Dim LSearchRow As Integer
Dim i As Integer
Dim GeneCounter As Integer
Sheets("1% Supercontrol VC").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("1% Supercontrol VC").Range("C" & CStr(LSearchRow)).Value) > 0
'Last row of data in Supercontrol mutation sheet
GeneCounter = Sheets("Supercontrol Results- Ampliseq").Range("A1").End(xlDown).Row
For i = 2 To GeneCounter
'If the positions match
If Sheets("1% Supercontrol VC").Range("C" & CStr(LSearchRow)).Value = Sheets("Supercontrol
Results- Ampliseq").Range("C" & CStr(i)).Value Then
Sheets("1% Supercontrol VC").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Ampliseq").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Ampliseq").Select
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
    'ELSE IF COSMIC ID Matches
    ElseIf Sheets("1% Supercontrol VC").Range("M" & CStr(LSearchRow)).Value =
Sheets("Supercontrol Results- Ampliseq").Range("F" & CStr(i)).Value Then
Sheets("1% Supercontrol VC").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Ampliseq").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Ampliseq").Select
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
Next i
LSearchRow = LSearchRow + 1
Wend
```

```
Range("A1").Select
Exit Sub
End Sub
Sub super_format_ng()
'Formats 1% Supercontrol NG report
Rows("1:4").Select
    Selection.ClearContents
    Rows("1:1").Select
    Selection.Insert Shift:=xlDown
    'Rows("2:2").Select
    'Selection.Insert Shift:=xlDown
    'Selection.Insert Shift:=xlDown
    'Selection.Insert Shift:=xlDown
    Range("A1").Select
    Selection.EntireColumn.Select
    Selection.Insert Shift:=xlToRight
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "Lot #"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "PGM Run #"
    Range("A4").Select
    ActiveCell.FormulaR1C1 = "Analysis By"
    Range("B1").Select
    Selection.EntireColumn.Select
    Range("A1").Select
    Selection.EntireColumn.AutoFit
    Range("A2:B4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
```

```
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .Weight = xlThin
        .ColorIndex = xlAutomatic
    End With
    Range("B3:B4").Select
    With Selection.Interior
        .ColorIndex = 15
        .Pattern = xlSolid
    End With
    Range("B2").Select
    With Selection.Interior
        .ColorIndex = 15
        .Pattern = xlSolid
    End With
'Add cell references
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[7]C[6]"
    Range("B3").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[5]C"
    Range("B4").Select
    ActiveCell.FormulaR1C1 = "=Demographics!R[4]C[3]"
    Range("B5").Select
End Sub
Sub super_find_muts_ng()
'Finds mutations in NG 1% supercontrol data
Dim LSearchRow As Integer
Dim i As Integer
Dim GeneCounter As Integer
Sheets("1% Supercontrol NG").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("1% Supercontrol NG").Range("E" & CStr(LSearchRow)).Value) > 0
'Last row of data in Supercontrol mutation sheet
GeneCounter = Sheets("Supercontrol Results- Ampliseq").Range("A1").End(xlDown).Row
For i = 2 To GeneCounter
'If the positions match
If Sheets("1% Supercontrol NG").Range("D" & CStr(LSearchRow)).Value = Sheets("Supercontrol
Results- Ampliseq").Range("C" & CStr(i)).Value Then
Sheets("1% Supercontrol NG").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Ampliseq").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Ampliseq").Select
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
        'ELSE if c. numbers match
```

```
        ElseIf Sheets("1% Supercontrol NG").Range("R" & CStr(LSearchRow)).Value =
Sheets("Supercontrol Results- Ampliseq").Range("D" & CStr(i)).Value Then
        Sheets("1% Supercontrol NG").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Ampliseq").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Ampliseq").Select
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
Next i
LSearchRow = LSearchRow + 1
Wend
Range("A1").Select
Exit Sub
End Sub
Attribute VB_Name = "Module33"
Sub super_message()
'ERROR MESSAGES
'TAB VC
Sheets("1% Supercontrol Allele Coverage").Select
Sheets("1% Supercontrol Allele Coverage").Range("A1").Select
If IsEmpty(Selection) Then
MsgBox "Please paste Supercontrol Allele Coverage data."
Exit Sub
End If
'TAB Nextgene
Sheets("1% Supercontrol NG").Select
If IsEmpty(Sheets("1% Supercontrol NG").Range("A1")) Then
MsgBox "Please paste Supercontrol Nextgene data."
Exit Sub
End If
'No Lot #
Sheets("Supercontrol Results").Select
If IsEmpty(Sheets("Supercontrol Results").Range("C13")) Then
MsgBox "Please enter Lot#."
Exit Sub
End If
End Sub
Attribute VB_Name = "Module34"
Sub pre_format_cov_an()
'Formats coverage analysis data
'Removes percent sign from all percentages without affect number value
Dim NewString As String
'For C7:C10
For i = 7 To 10
'If it contains a percent sign, replace with number only
If InStr(1, Range("C" & CStr(i)).Text, "%") > 0 Then
NewString = Left(Range("C" & CStr(i)).Text, (Len(Range("C" & CStr(i)).Text) - 1))
Range("C" & CStr(i)).Select
Selection.NumberFormat = "@"
ActiveCell.FormulaR1C1 = NewString
End If
Next i
'For B13:B22
For i = 13 To 22
'If it contains a percent sign, replace with number only
If InStr(1, Range("B" & CStr(i)).Text, "%") > 0 Then
NewString = Left(Range("B" & CStr(i)).Text, (Len(Range("B" & CStr(i)).Text) - 1))
Range("B" & CStr(i)).Select
Selection.NumberFormat = "@"
ActiveCell.FormulaR1C1 = NewString
End If
```

```
Next i
'For D13:D21
For i = 13 To 21
'If it contains a percent sign, replace with number only
If InStr(1, Range("D" & CStr(i)).Text, "%") > 0 Then
NewString = Left(Range("D" & CStr(i)).Text, (Len(Range("D" & CStr(i)).Text) - 1))
Range("D" & CStr(i)).Select
Selection.NumberFormat = "@"
ActiveCell.FormulaR1C1 = NewString
End If
Next i
End Sub
Attribute VB_Name = "Module35"
Sub super_nomut_vc()
Dim GeneCounter As Integer
GeneCounter = Sheets("Supercontrol Results- Ampliseq").Range("A2").End(xlDown).Row
For i = 2 To GeneCounter
If Range("G" & CStr(i)).Text = "YES" Then
    'do nothing
Else
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "NO"
End If
Next i
End Sub
Sub super_nomut_ng()
Dim GeneCounter As Integer
GeneCounter = Sheets("Supercontrol Results- Ampliseq").Range("A2").End(xlDown).Row
For i = 2 To GeneCounter
If Range("H" & CStr(i)).Text = "YES" Then
    'do nothing
Else
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "NO"
End If
Next i
End Sub
Sub super_avg_freq()
Dim LSearchRow As Integer
Dim VCValue As Single
Dim NGValue As Single
'Set variables
LSearchRow = 2
VCValue = Empty
NGValue = Empty
'VCValue & NGValue will remain Empty if no mutation is found
'For each mutation on Results Page
While Len(Range("A" & CStr(LSearchRow)).Text) > 0
'VARIANT CALLER
For i = 6 To Sheets("1% Supercontrol VC").Range("C" & Rows.Count).End(xlUp).Row
'Found in VC
If Sheets("1% Supercontrol VC").Range("C" & CStr(i)).Value = Sheets("Supercontrol Results-
Ampliseq").Range("C" & CStr(LSearchRow)).Value And IsNumeric(Sheets("1% Supercontrol
VC").Range("H" & CStr(i)).Value) Then
    VCValue = Sheets("1% Supercontrol VC").Range("H" & CStr(i)).Value
    'If value is found, exit loop and proceed to next step
    Exit For
End If
Next i
'NEXTGENE
For j = 6 To Sheets("1% Supercontrol NG").Range("B" & Rows.Count).End(xlUp).Row
'Found in NG
If Sheets("1% Supercontrol NG").Range("D" & CStr(j)).Value = Sheets("Supercontrol Results-
Ampliseq").Range("C" & CStr(LSearchRow)).Value And IsNumeric(Sheets("1% Supercontrol
NG").Range("S" & CStr(j)).Value) Then
NGValue = Sheets("1% Supercontrol NG").Range("S" & CStr(j)).Value
    Exit For
    'If value is found, exit loop and proceed to next step
End If
Next j
'If mutation is NOT found in either
```

```
If IsEmpty(VCValue) And IsEmpty(NGValue) Then
    'do nothing
'If mutation is not found in VC but found in NG
ElseIf VCValue = 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = NGValue
'If mutation is found in VC but not found in NG
ElseIf VCValue > 0 And NGValue = 0 Then
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = VCValue
'If mutation is found in BOTH
ElseIf VCValue > 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = (VCValue + NGValue) / 2
End If
'Clear Variables
VCValue = Empty
NGValue = Empty
'Search for next mutation
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub super_coverage()
Dim LSearchRow As Integer
Dim VCValue As Single
Dim NGValue As Single
'Set variables
LSearchRow = 2
VCValue = Empty
NGValue = Empty
'VCValue & NGValue will remain Empty if no mutation is found
'For each mutation on Results Page
While Len(Range("A" & CStr(LSearchRow)).Text) > 0
'VARIANT CALLER
For i = 6 To Sheets("1% Supercontrol VC").Range("C" & Rows.Count).End(xlUp).Row
'Found in VC
If Sheets("1% Supercontrol VC").Range("C" & CStr(i)).Value = Sheets("Supercontrol Results-
Ampliseq").Range("C" & CStr(LSearchRow)).Value Then
    VCValue = Sheets("1% Supercontrol VC").Range("T" & CStr(i)).Value
    'If value is found, exit loop and proceed to next step
    Exit For
End If
Next i
'NEXTGENE
For j = 6 To Sheets("1% Supercontrol NG").Range("B" & Rows.Count).End(xlUp).Row
'Found in NG
If Sheets("1% Supercontrol NG").Range("D" & CStr(j)).Value = Sheets("Supercontrol Results-
Ampliseq").Range("C" & CStr(LSearchRow)).Value Then
NGValue = Sheets("1% Supercontrol NG").Range("I" & CStr(j)).Value
    Exit For
    'If value is found, exit loop and proceed to next step
End If
Next j
'If mutation is NOT found in either
If IsEmpty(VCValue) And IsEmpty(NGValue) Then
    'do nothing
'If mutation is not found in VC but found in NG
ElseIf VCValue = 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = NGValue
'If mutation is found in VC but not found in NG
ElseIf VCValue > 0 And NGValue = 0 Then
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = VCValue
'If mutation is found in BOTH
ElseIf VCValue > 0 And NGValue > 0 Then
```

```
Sheets("Supercontrol Results- Ampliseq").Select
Sheets("Supercontrol Results- Ampliseq").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = (VCValue + NGValue) / 2
End If
'Clear Variables
VCValue = Empty
NGValue = Empty
'Search for next mutation
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module36"
Sub modng_find_new()
'Appends added mutation to Nextgene tab from modified nextgene report
'Go to Nextgene Modified data
Sheets("Nextgene Modified").Select
'VARIABLES
Dim LSearchRow As Integer
    'Line of Nextgene Modified
Dim LastRowNG As Integer
    'Line of Nextgene data
'Start search in row 6
LSearchRow = 6
'For all of the data in Nextgene Modified
While Len(Sheets("Nextgene Modified").Range("B" & CStr(LSearchRow)).Value) > 0
'Define last row of data in Nextgene Data
LastRowNG = Sheets("Nextgene").Range("B6").End(xlDown).Row
'For each SNP in database
For i = 6 To LastRowNG
'If position in Nextgene Modified matches Nextgene
If Sheets("Nextgene Modified").Range("C" & CStr(LSearchRow)).Value = Sheets("Nextgene").Range("D"
& CStr(i)).Value Then
Exit For
Else
    'If match has not been found and last row
    If i = LastRowNG Then
    Sheets("Nextgene Modified").Select
    Range("A" & CStr(LSearchRow) & ":S" & CStr(LSearchRow)).Select
    Selection.Copy
    Sheets("Nextgene").Select
    Range("B" & CStr(LastRowNG + 1)).Select
    ActiveSheet.Paste
    GoTo Finish
    End If
End If
Next i
LSearchRow = LSearchRow + 1
Wend
'Add "TRUE", add extra needed column to align data
Finish:
Range("A" & CStr(LastRowNG + 1)).Select
ActiveCell.FormulaR1C1 = True
Range("G" & CStr(LastRowNG + 1)).Select
Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
'Transfer data to final report
Dim RepRow As Integer
'Row of final report to add new mutation
'Find next available row in Nextgene results
Sheets("Final Report Details").Select
'If spot one is filled
If Len(Range("A24").Text) > 0 Then
    'If spot two is filled
    If Len(Range("A25").Text) > 0 Then
    RepRow = 26
    Else
    RepRow = 25
    End If
Else
RepRow = 24
End If
'Review
```

```
Sheets("Final Report Details").Range("A" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("A" & CStr(LastRowNG + 1)).Text
'Chromosome
Sheets("Final Report Details").Range("B" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = "chr" & Sheets("Nextgene").Range("F" & CStr(LastRowNG + 1)).Text
'Position
Sheets("Final Report Details").Range("C" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("D" & CStr(LastRowNG + 1)).Text
'GeneID
Sheets("Final Report Details").Range("D" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("E" & CStr(LastRowNG + 1)).Text
'Total Coverage VC
Sheets("Final Report Details").Range("E" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Total Coverage NG
Sheets("Final Report Details").Range("F" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("J" & CStr(LastRowNG + 1)).Text
'Quality/Score
Sheets("Final Report Details").Range("G" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("K" & CStr(LastRowNG + 1)).Text
'Strand Bias
Sheets("Final Report Details").Range("H" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency VC
Sheets("Final Report Details").Range("I" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = "-"
'Frequency NG
Sheets("Final Report Details").Range("J" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("T" & CStr(LastRowNG + 1)).Text
'Mutation Call
Sheets("Final Report Details").Range("K" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("S" & CStr(LastRowNG + 1)).Text
'Amino Acid Change
Sheets("Final Report Details").Range("L" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("U" & CStr(LastRowNG + 1)).Text
'Db_snp / Cosmic
Sheets("Final Report Details").Range("M" & CStr(RepRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Nextgene").Range("R" & CStr(LastRowNG + 1)).Text
End Sub
Attribute VB_Name = "Module37"
Sub MODNG_()
Dim strFileName As String
strFileName = ActiveWorkbook.Name
Sheets("Nextgene Modified").Select
'Delete unneeded columns
Application.Run "'" & strFileName & "'" & "!pre_prepare_ng"
'Transfer row to Nextgene report; add to Final Report Details
Application.Run "'" & strFileName & "'" & "!modng_find_new"
'Add to mutation summary on final report
Application.Run "'" & strFileName & "'" & "!post_sum_mutations"
Sheets("Final Report Summary").Select
End Sub
Attribute VB_Name = "Module38"
Attribute VB_Name = "Module39"
Sub MEDI_()
Dim strFileName As String
strFileName = ActiveWorkbook.Name
'Error Messages
Sheets("Final Report Summary").Select
'For entire ROI table
For i = 24 To Range("A24").End(xlDown).Row
If Len(Range("F" & CStr(i)).Text) = 0 Then
Range("F" & CStr(i)).Select
MsgBox ("Please choose result for each row of Pertinent Negatives table.")
Exit Sub
End If
Next i
Sheets("Meditech Report").Select
'''''''Meditech Diag Report
'Result. Positive, Negative, Other Findings
Application.Run "'" & strFileName & "'" & "!medi_report_1_result"
```

```
'Notes
Application.Run "'" & strFileName & "'" & "!medi_report_2_notes"
Application.Run "'" & strFileName & "'" & "!medi_report_2b_notes_other"
'Interpretation. Gene, Mut, Type, Cosmic, Diag, Prog, Therapy
Application.Run "'" & strFileName & "'" & "!medi_report_3_interp"
'Add resources based on tumor type
Application.Run "'" & strFileName & "'" & "!medi_report_4_resour"
'Add Table 1 with gene, chr, result, and coverage
Application.Run "'" & strFileName & "'" & "!medi_report_5_table1"
'Add end of report with Table 2 and References
Application.Run "'" & strFileName & "'" & "!medi_report_6_end_of_report"
'''''''Meditech Raw Data Report
'Result: mutations detected or no mutations detected
Application.Run "'" & strFileName & "'" & "!medi_raw_1_result"
'Interpretation: individua data for mutations
Application.Run "'" & strFileName & "'" & "!medi_raw_2_interp"
'Table with mutation info (pertinent negatives table)
Application.Run "'" & strFileName & "'" & "!medi_raw_3_table"
'End of report
Application.Run "'" & strFileName & "'" & "!medi_raw_4_end_of_report"
Sheets("Final Report Summary").Select
Range("A1").Select
End Sub
Attribute VB_Name = "Module4"
Sub pre_format_ng()
'Formats raw data and deletes uneccessary text
'Add "Review" Column
    Columns("A:A").Select
    Selection.Insert Shift:=xlToRight, CopyOrigin:=xlFormatFromLeftOrAbove
'Add "Review" text
    Range("A5").Select
    ActiveCell.FormulaR1C1 = "Review"
'Clear unncessary text
    Range("B1:C1").Select
    Selection.ClearContents
    Range("B3:C3").Select
    Selection.ClearContents
    Range("B2").Select
    Selection.ClearContents
'Move file name to cell A1
    Range("C2").Select
    Selection.Cut
    Range("A1").Select
    ActiveSheet.Paste
'Add DMO and Analysis Info at top
    'Add text
        'DMO number
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "DMO Number"
    Range("A3").Select
    ActiveCell.FormulaR1C1 = "=Demographics!RC[1]"
        'Analysis by
    Range("B2").Select
    ActiveCell.FormulaR1C1 = "Analysis By"
    Range("B3").Select
    ActiveCell.FormulaR1C1 = Sheets("Demographics").Range("E8").Text
End Sub
Attribute VB_Name = "Module40"
Sub medi_report_2_notes()
'Adds notes to Meditech Report
Dim AAChange As String
Dim Gene As String
'Define Tumor Type
TumorType = Sheets("Demographics").Range("B14").Text
For i = 4 To 26
'If mutation in row on Final Report
If Len(Sheets("Final Report Details").Range("B" & CStr(i)).Text) > 0 Then
    Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
    AAChange = Sheets("Final Report Details").Range("L" & CStr(i)).Text
'Change format of mutation text to "p.A123B"
'Define variables
```

```
'amino acid
Dim NewAA As String 'modified Amino Acid change
Dim PartOne As String 'section of AA change before ">"
Dim PartTwo As String 'section of AA change after ">"
Dim LettersOnlyOne As String 'single letter extracted from PartOne
Dim LettersOnlyTwo As String 'two letters extracted from PartTwo
Dim MutNum As Integer 'number extracted from AA change
'If FS, in-frame, or splice don't change anything, New is same as Original
If AAChange = "FS" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "In-Frame" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "Splice" Then
    NewAA = AAChange
    NewMutCall = MutCall
'If "-" (ie. Variant caller, skip to next part of function)
ElseIf AAChange = "-" Then
    Exit Sub
'Otherwise
Else
'Determine Amino Acid
'
'
    'before ">"
PartOne = Left(AAChange, InStrRev(AAChange, ">") - 1)
    'after ">"
PartTwo = Right(AAChange, Len(AAChange) - InStrRev(AAChange, ">"))
'Extract letter (amino acid) from part 1
Dim X As Long
  For X = 1 To Len(PartOne)
    If Mid(PartOne, X, 1) Like "[!A-Za-z]" Then Mid(PartOne, X, 1) = " "
  Next
  LettersOnlyOne = Replace(PartOne, " ", "")
'Extract two letters
'This should have two letters, one that repeats with LettersOnlyOne
Dim Y As Long
  For Y = 1 To Len(PartTwo)
    If Mid(PartTwo, Y, 1) Like "[!A-Za-z]" Then Mid(PartTwo, Y, 1) = " "
  Next
  LettersOnlyTwo = Replace(PartTwo, " ", "")
'Determine which letter is a repeat and remove
    'IF FIRST LETTER MATCHES, REMOVE
If LettersOnlyOne = Left(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Right(LettersOnlyTwo, 1)
    'IF SECOND LETTER MATCHES, REMOVE
ElseIf LettersOnlyOne = Right(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Left(LettersOnlyTwo, 1)
End If
'Get number from original AACall
MutNum = GetNums(AAChange)
'Create NewAA
NewAA = "p." & LettersOnlyOne & MutNum & LettersOnlyTwo
End If
'CHECK NOTE DATABASE
Sheets("Meditech Notes").Select
'Check notes for all tumor types
For j = 2 To Range("A" & Rows.Count).End(xlUp).Row
'Tumor = All
If Range("A" & CStr(j)).Text = "All" Then
    'Genes match
    If Range("C" & CStr(j)).Text = Gene Then
        'Mutations match
        If Range("D" & CStr(j)).Text = NewAA Then
            NoteText = Range("E" & CStr(j)).Text
            Sheets("Meditech Report").Select
            MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
            Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = NoteText
            Sheets("Meditech Notes").Select
```

```
        End If
    End If
End If
Next j
'For all notes in database
For j = 2 To Range("A" & Rows.Count).End(xlUp).Row
'Tumors match
If Range("A" & CStr(j)).Text = TumorType Then
    'Genes match
    If Range("C" & CStr(j)).Text = Gene Then
        'Mutations match
        If Range("D" & CStr(j)).Text = NewAA Then
            NoteText = Range("E" & CStr(j)).Text
            Sheets("Meditech Report").Select
            MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
            Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = NoteText
        'General for gene
        ElseIf IsEmpty(Range("D" & CStr(j))) Then
            'AML Baseline
            If Range("A" & CStr(j)).Text = "AML" And Range("B" & CStr(j)).Text = "Baseline
Testing" And Range("B19" & CStr(j)).Text = "Yes" Then
                NoteText = Range("E" & CStr(j)).Text
                Sheets("Meditech Report").Select
                MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
                Range("A" & CStr(MyRow)).Select
                ActiveCell.FormulaR1C1 = NoteText
            End If
            NoteText = Range("E" & CStr(j)).Text
            Sheets("Meditech Report").Select
            MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
            Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = NoteText
        End If
    'General for tumor type
    ElseIf IsEmpty(Range("C" & CStr(j))) Then
    NoteText = Range("E" & CStr(j)).Text
    Sheets("Meditech Report").Select
    MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
    Range("A" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = NoteText
    End If
End If
Next j
End If
Next i
End Sub
Attribute VB_Name = "Module41"
Sub medi_report_3_interp()
'Define variables
Dim LSearchRow As Integer
Dim Pos As String
Dim Gene As String
Dim MutCall As String
Dim AAChange As String
Dim Cosm As String
Dim MutFreq As String
Dim MutCounter As Integer
MutCounter = 0
'total of all mutations in final report
Dim GeneNum As Integer
GeneNum = 1
'gene number for interp text
'ADD TEXT
'Add interpretation text with underline
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Sheets("Meditech Report").Range("A" & CStr(MyRow + 1)).Select
ActiveCell.FormulaR1C1 = "INTERPRETATION                                                    --------------"
'Check if mutations are present for Common Findings, Variant Caller, and nextgene
If Len(Sheets("Final Report Details").Range("B4").Text) = 0 Then
```

```
    If Len(Sheets("Final Report Details").Range("B20").Text) = 0 Then
        If Len(Sheets("Final Report Details").Range("B24").Text) = 0 Then
            'If MRD is negative (no mutations) and baseline = "Yes"
            If Sheets("Demographics").Range("B19").Text = "No" And
Sheets("Demographics").Range("B20").Text = "Yes" Then
            'Skip to negative MRD interpretation and then exit sub
            MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
            ActiveCell.FormulaR1C1 = Sheets("Meditech Interpretation").Range("D3").Text
            Exit Sub
            Else
            'Skip to no mutation text and then exit sub
            MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
            ActiveCell.FormulaR1C1 = Sheets("Meditech Interpretation").Range("D2").Text
            Exit Sub
            End If
        End If
    End If
End If
'Add interp note
Sheets("Meditech Interpretation").Select
'For all interp notes
For i = 2 To Range("D2").End(xlDown).Row
'Tumor type matches and not for negative
If Range("A" & CStr(i)).Text = Sheets("Demographics").Range("B14").Text And Range("C" &
CStr(i)).Text <> "Negative" Then
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Meditech Interpretation").Range("D" & CStr(i)).Text
Exit For
End If
Next i
'If at least one mutation is present
'For all lines in Final Report
For i = 4 To 26
'If mutation (if length of text in column B is greater than 0)
If Len(Sheets("Final Report Details").Range("B" & CStr(i)).Text) > 0 Then
'Position
Pos = Sheets("Final Report Details").Range("C" & CStr(i)).Value
'Gene
Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
'Mutation Call
MutCall = Sheets("Final Report Details").Range("K" & CStr(i)).Text
'Amino Acid Change
AAChange = Sheets("Final Report Details").Range("L" & CStr(i)).Text
'COSMIC ID
Cosm = Sheets("Final Report Details").Range("M" & CStr(i)).Text
'Mutation Frequency
    'use Nextgene frequency
    If i <= 18 Or i >= 24 Then
    MutFreq = Sheets("Final Report Details").Range("J" & CStr(i)).Text
    'otherwise use variant caller frequency
    Else
    MutFreq = Sheets("Final Report Details").Range("I" & CStr(i)).Text
    End If
'Add interpretation fields
'GENE
Sheets("Meditech Report").Select
If GeneNum = 1 Then
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
Else
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
Range("A" & CStr(MyRow)).Select
End If
ActiveCell.FormulaR1C1 = "  GENE #" & CStr(GeneNum) & ":"
GeneNum = GeneNum + 1
'Find full name of gene from database
Dim FullName As String
    For j = 2 To Sheets("Gene Names").Range("A1").End(xlDown).Row
        If Gene = Sheets("Gene Names").Range("A" & CStr(j)).Text Then
```

```
        FullName = Sheets("Gene Names").Range("B" & CStr(j)).Text
        Exit For
        End If
    Next j
'Add text to report
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Gene & " (" & FullName & ")"
'MUTATION
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = " MUTATION:"
'Define variables
'amino acid
Dim NewAA As String
    'modified Amino Acid change
Dim PartOne As String
    'section of AA change before ">"
Dim PartTwo As String
    'section of AA change after ">"
Dim LettersOnlyOne As String
    'single letter extracted from PartOne
Dim LettersOnlyTwo As String
    'two letters extracted from PartTwo
Dim MutNum As Integer
    'number extracted from AA change
'mutation call
Dim NewMutCall As String
    'modified Mutation Call
Dim PartOneCallOrig As String
    'section of MutCall before ">", including "c."
Dim PartOneCall As String
    'section of MutCall before ">", without "c."
Dim PartTwoCall As String
    'section of MutCall after ">"
Dim LettersOnlyOneCall As String
    'single letter extracted from PartOneCall
Dim LettersOnlyTwoCall As String
    'single letter extracted from PartTwoCall
Dim MutNumCall As Integer
    'number extracted from MutCall
'If FS, in-frame, or splice don't change anything, New is same as Original
If AAChange = "FS" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "In-Frame" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "Splice" Then
    NewAA = AAChange
    NewMutCall = MutCall
'If "-" (ie. Variant caller, skip to next part of function)
ElseIf AAChange = "-" Then
'Add "type" text and move to cosmic ID
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "     TYPE:"
    GoTo CosmicSection
'Otherwise
Else
'Determine Amino Acid
'
'
    'before ">"
PartOne = Left(AAChange, InStrRev(AAChange, ">") - 1)
    'after ">"
PartTwo = Right(AAChange, Len(AAChange) - InStrRev(AAChange, ">"))
'Extract letter (amino acid) from part 1
Dim X As Long
  For X = 1 To Len(PartOne)
    If Mid(PartOne, X, 1) Like "[!A-Za-z]" Then Mid(PartOne, X, 1) = " "
  Next
```

```
  LettersOnlyOne = Replace(PartOne, " ", "")
'Extract two letters
'This should have two letters, one that repeats with LettersOnlyOne
Dim Y As Long
  For Y = 1 To Len(PartTwo)
    If Mid(PartTwo, Y, 1) Like "[!A-Za-z]" Then Mid(PartTwo, Y, 1) = " "
  Next
  LettersOnlyTwo = Replace(PartTwo, " ", "")
'Determine which letter is a repeat and remove
    'IF FIRST LETTER MATCHES, REMOVE
If LettersOnlyOne = Left(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Right(LettersOnlyTwo, 1)
    'IF SECOND LETTER MATCHES, REMOVE
ElseIf LettersOnlyOne = Right(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Left(LettersOnlyTwo, 1)
End If
'Get number from original AACall
MutNum = GetNums(AAChange)
'Create NewAA
NewAA = "p." & LettersOnlyOne & MutNum & LettersOnlyTwo
'Determine Mutation Call
'
'
    'before ">"
PartOneCallOrig = Left(MutCall, InStrRev(MutCall, ">") - 1)
    'Take away c. at beginning
PartOneCall = Right(PartOneCallOrig, Len(PartOneCallOrig) - 2)
    'after ">"
PartTwoCall = Right(MutCall, Len(MutCall) - InStrRev(MutCall, ">"))
'Extract letter from PartOneCall
Dim Z As Long
  For Z = 1 To Len(PartOneCall)
    If Mid(PartOneCall, Z, 1) Like "[!A-Za-z]" Then Mid(PartOneCall, Z, 1) = " "
  Next
  LettersOnlyOneCall = Replace(PartOneCall, " ", "")
'Extract two letters from PartTwoCall
'This should have two letters, one that repeats with LettersOnlyOneCall
Dim a As Long
  For a = 1 To Len(PartTwoCall)
    If Mid(PartTwoCall, a, 1) Like "[!A-Za-z]" Then Mid(PartTwoCall, a, 1) = " "
  Next
  LettersOnlyTwoCall = Replace(PartTwoCall, " ", "")
'Determine which base is a repeat and remove
    'IF FIRST LETTER MATCHES, REMOVE
If LettersOnlyOneCall = Left(LettersOnlyTwoCall, 1) Then
LettersOnlyTwoCall = Right(LettersOnlyTwoCall, 1)
    'IF SECOND LETTER MATCHES, REMOVE
ElseIf LettersOnlyOneCall = Right(LettersOnlyTwoCall, 1) Then
LettersOnlyTwoCall = Left(LettersOnlyTwoCall, 1)
End If
'Get number from original MutCall
MutNumCall = GetNums(MutCall)
'New call
NewMutCall = "c." & MutNumCall & LettersOnlyOneCall & ">" & LettersOnlyTwoCall
End If
'Add text to report
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = NewAA & " (" & NewMutCall & ")"
'TYPE
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "     TYPE:"
Range("B" & CStr(MyRow)).Select
'Missense
If InStr(1, MutCall, ">") > 0 Then
ActiveCell.FormulaR1C1 = "Missense"
'Nonsense
ElseIf InStr(1, MutCall, "X") > 0 Then
ActiveCell.FormulaR1C1 = "Nonsense"
'Insertion
ElseIf InStr(1, MutCall, "ins") > 0 Then
```

```
ActiveCell.FormulaR1C1 = "Insertion"
'Deletion
ElseIf InStr(1, MutCall, "del") > 0 Then
ActiveCell.FormulaR1C1 = "Deletion"
End If
'COSMIC ID
CosmicSection:
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = " COSMIC #:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Cosm
'MUTATION %
'Only if AML
If Sheets("Demographics").Range("B14").Text = "AML" Then
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "MUTATION%:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = MutFreq
End If
'FUNCTION
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = " FUNCTION:"
'Get function from database
Dim Funct As String
For j = 2 To Sheets("Gene Function").Range("A1").End(xlDown).Row
    'First, try to match with specific function for the mutation
        'If COSMIC IDs match
        If Cosm = Sheets("Gene Function").Range("C" & CStr(j)).Text Then
        Funct = Sheets("Gene Function").Range("D" & CStr(j)).Text
        Exit For
        'Else If GENE and Mutation match
        ElseIf Gene = Sheets("Gene Function").Range("A" & CStr(j)).Text And NewAA = Sheets("Gene
Function").Range("B" & CStr(j)).Text Then
        Funct = Sheets("Gene Function").Range("D" & CStr(j)).Text
        Exit For
    'If can't match by COSMIC ID or Gene And Mutation, use general gene function
        Else
            For k = 2 To Sheets("Gene Function General").Range("A1").End(xlDown).Row
                If Gene = Sheets("Gene Function General").Range("A" & CStr(k)).Text Then
                Funct = Sheets("Gene Function General").Range("B" & CStr(k)).Text
                Exit For
                End If
            Next k
        End If
        Exit For
Next j
'Add text to report
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Funct
'DIAGNOSIS
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "DIAGNOSIS:"
Dim Diagnosis As String
'If no COSMIC ID or just dbSNP (starts with "rs")
If Cosm = "---" Or InStr(1, Cosm, "rs") > 0 Then
Diagnosis = "Not a previously reported somatic mutation"
'Otherwise get diagnosis from database
Else
    For j = 2 To Sheets("Gene Diagnosis").Range("A1").End(xlDown).Row
        'Check gene
        If Gene = Sheets("Gene Diagnosis").Range("A" & CStr(j)).Text Then
                'Check cosmic ID
                If Cosm = Sheets("Gene Diagnosis").Range("B" & CStr(j)).Text Then
                Diagnosis = Sheets("Gene Diagnosis").Range("H" & CStr(j)).Text
                Exit For
                End If
        End If
```

```
    Next j
End If
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Diagnosis
'PROGNOSIS
'Only if AML
If Sheets("Demographics").Range("B14").Text = "AML" Then
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "PROGNOSIS:"
Dim Prognosis As String
'If no COSMIC ID or just dbSNP (starts with "rs")
If Cosm = "---" Or InStr(1, Cosm, "rs") > 0 Then
Progosis = ""
'Otherwise get prognosis from database
Else
    For j = 2 To Sheets("Gene Prognosis").Range("A1").End(xlDown).Row
        'Check gene
        If Gene = Sheets("Gene Prognosis").Range("A" & CStr(j)).Text Then
                'Check cosmic ID
                If Cosm = Sheets("Gene Prognosis").Range("B" & CStr(j)).Text Then
                Prognosis = Sheets("Gene Prognosis").Range("H" & CStr(j)).Text
                Exit For
                End If
        End If
    Next j
End If
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Prognosis
End If
'THERAPY
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "  THERAPY:"
Dim Therapy As String
For j = 2 To Sheets("Gene Therapy").Range("A1").End(xlDown).Row
        'Check gene
        If Gene = Sheets("Gene Therapy").Range("A" & CStr(j)).Text Then
                Therapy = Sheets("Gene Therapy").Range("B" & CStr(j)).Text
                Exit For
        End If
    Next j
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Therapy
End If
Next i
End Sub
Function GetNums(Target As String)
'Extracts numbers from alphanumeric string
    Dim MyStr As String, i As Integer
    MyStr = ""
    If Len(Target) = 0 Then GoTo GoExit
    For i = 1 To Len(Target)
        If IsNumeric(Mid(Target, i, 1)) Then MyStr = MyStr & Mid(Target, i, 1)
    Next i
GoExit:
    GetNums = MyStr
End Function
Attribute VB_Name = "Module42"
Sub medi_report_4_resour()
'Copy's respective resource block for the given tumor type
'Appends to Meditech report beneath interpretation
MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
'last row of data in report plus one
'COLON CANCER
    If Sheets("Demographics").Range("B14").Text = "Colon Cancer (CRC)" Then
    Sheets("Resources by Tumor").Select
    Range("B2:M10").Select
    Selection.Copy
'BREAST CANCER
    ElseIf Sheets("Demographics").Range("B14").Text = "Breast Cancer" Then
```

```
    Sheets("Resources by Tumor").Select
    Range("B13:M21").Select
    Selection.Copy
'LUNG CANCER
    ElseIf Sheets("Demographics").Range("B14").Text = "Lung Cancer" Then
    Sheets("Resources by Tumor").Select
    Range("B24:M32").Select
    Selection.Copy
'AML
    ElseIf Sheets("Demographics").Range("B14").Text = "AML" Then
    Sheets("Resources by Tumor").Select
    Range("B35:M43").Select
    Selection.Copy
'PANCREATIC CA & CYST FLUID
    ElseIf Sheets("Demographics").Range("B14").Text = "Pancreatic CA & Cyst Fluid" Then
    Sheets("Resources by Tumor").Select
    Range("B46:M54").Select
    Selection.Copy
'MELANOMA
    ElseIf Sheets("Demographics").Range("B14").Text = "Malignant Melanoma" Then
    Sheets("Resources by Tumor").Select
    Range("B57:M65").Select
    Selection.Copy
'THYROID CANCER
    ElseIf Sheets("Demographics").Range("B14").Text = "Thyroid Cancer" Then
    Sheets("Resources by Tumor").Select
    Range("B68:M76").Select
    Selection.Copy
'GIST
    ElseIf Sheets("Demographics").Range("B14").Text = "GIST" Then
    Sheets("Resources by Tumor").Select
    Range("B79:M87").Select
    Selection.Copy
'OTHER
    ElseIf Sheets("Demographics").Range("B14").Text = "Other" Then
    'do nothing
End If
Sheets("Meditech Report").Select
Range("A" & CStr(MyRow)).PasteSpecial xlPasteValues
End Sub
Attribute VB_Name = "Module43"
Sub medi_copy()
'Copies entire meditech report to clipboard on click of the button
Range("A1:D" & Range("A" & Rows.Count).End(xlUp).Row).SpecialCells(xlCellTypeVisible).Select
Selection.Copy
Range("A1").Select
End Sub
Sub medi_clear()
'Clears entire meditech report on click of the button
Range("A3:D1000").ClearContents
'Reset Font, Size, and cell type
    Columns("A:D").Select
    With Selection.Font
        .Name = "Courier New"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    With Selection.Font
        .Name = "Courier New"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
```

```
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    Selection.NumberFormat = "General"
Range("A1").Select
End Sub
Attribute VB_Name = "Module44"
Sub post_sum_roi_mutant()
'Adds "Mutant" to ROI table of Final Report Summary
'Determine Start and End Range of Codons
Sheets("Final Report Summary").Select
Dim LastROI As Integer
    'last row of ROI table on Final Report Summary Page
Dim LengthROI As Integer
    'length of ROI table
Dim StartNum As Integer
Dim EndNum As Integer
Dim CodonRange As String
LastROI = Sheets("Final Report Summary").Range("A24").End(xlDown).Row
LengthROI = LastROI - 23
'Make an array to hold values
'Start codon array
Dim StartArray() As String
Dim EndArray() As String
'For each position in ROI table
For i = 24 To LastROI
CodonRange = Range("D" & CStr(i)).Text
'If range and not single number
If InStr(1, CodonRange, "-") > 0 Then
'Start Codon
StartNum = Left(CodonRange, InStr(1, CodonRange, "-") - 1)
'End Codon
EndNum = Right(CodonRange, Len(CodonRange) - InStr(1, CodonRange, "-"))
Else
StartNum = 0
EndNum = 0
End If
ReDim Preserve StartArray(0 To i - 23) As String
StartArray(i - 24) = StartNum
ReDim Preserve EndArray(0 To i - 23) As String
EndArray(i - 24) = EndNum
Next i
'Check for mutation on Final Report Details
Dim Gene As String
Dim MutCall As String
Dim AAChange As String
'For entire ROI table on Final Report Summary
For i = 24 To LastROI
    'For entire Final Report Details
    For j = 4 To 26
'If no mutation on line
If IsEmpty(Sheets("Final Report Details").Range("B" & CStr(j))) Then
    'do nothing; check next i
'If mutation
Else
    'If gene matches gene in ROI table
    If Trim(Sheets("Final Report Details").Range("D" & CStr(j)).Text) = Trim(Sheets("Final Report
Summary").Range("A" & CStr(i)).Text) Then
    'Set strings equal to mutation info
    Gene = Sheets("Final Report Details").Range("D" & CStr(j)).Text
    MutCall = Sheets("Final Report Details").Range("K" & CStr(j)).Text
    AAChange = Sheets("Final Report Details").Range("L" & CStr(j)).Text
    CodonRange = Sheets("Final Report Summary").Range("D" & CStr(i)).Text
        'If codon range is numeric, not a range. ex: 100
        If IsNumeric(CodonRange) Then
            'If codon matches codon in ROI table
            If InStr(1, AAChange, CodonRange) > 0 Then
                GoTo MutantText
            'If IVS
```

```
            ElseIf Left(MutCall, 3) = "IVS" Then
                'Get text without IVS
                NoIVS = Right(MutCall, Len(MutCall) - 3)
                    'If -
                    If InStr(NoIVS, "-") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "-") - 1)
                    'ElseIf +
                    ElseIf InStr(NoIVS, "+") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "+") - 1)
                    End If
                'Divide just the number from mutation call by 3 to get codon
                CodonNumDouble = JustNum / 3
                CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
                If SearchNum = CodonNum Then
                GoTo MutantText
                End If
        'If codon does not match codon in final report use mutation call
        Else
            'CHalfway is the halfway point of the codons (where the beginning and ending number
split)
            CHalfway = Len(GetNums(MutCall)) / 2
            'If CHalfway is an integer (ie starting and ending point have same number of digits)
            If Int(Len(CHalfway)) / Len(CHalfway) = 1 Then
                'SearchforCodon is the start number of the mutation
                Searchfor = Left(GetNums(MutCall), CHalfway)
            'If CHalfway is a decimal
            Else
                'SearchforCodon is the start number of the mutation (number with fewer digits)
                Searchfor = Left(GetNums(MutCall), Int(CHalfway))
            End If
        CodonNumDouble = Searchfor / 3
        CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
            If CodonRange = Searchfor / 3 Then
            GoTo MutantText
            End If
        End If
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
    'If codon isn't numeric, but is a range. ex: 100-200
    Else
        StartRange = StartArray(i - 24)
        EndRange = EndArray(i - 24)
            'If AAChange in final report is numeric
            If IsNumeric(GetNums(AAChange)) Then
                CodonNum = GetNums(AAChange)
                GoTo CheckMatch
            'ElseIf IVS
            ElseIf Left(MutCall, 3) = "IVS" Then
                'Text without IVS
                NoIVS = Right(MutCall, Len(MutCall) - 3)
                    'If -
                    If InStr(NoIVS, "-") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "-") - 1)
                    'ElseIf +
                    ElseIf InStr(NoIVS, "+") > 0 Then
                    JustNum = Left(NoIVS, InStr(NoIVS, "+") - 1)
                    End If
                'Divide mutcall by 3
                CodonNumDouble = JustNum / 3
                CodonNum = Application.WorksheetFunction.RoundUp(CodonNumDouble, 0)
                GoTo CheckMatch
            'Else use mutation call
            Else
                'Define what to search for
                'CHalfway is the halfway point of the codons (where the beginning and ending
number split)
                CHalfway = Len(GetNums(MutCall)) / 2
                'If CHalfway is an integer (ie starting and ending point have same number of
digits)
                If Int(Len(CHalfway)) / Len(CHalfway) = 1 Then
                   'SearchforCodon is the start number of the mutation
                   CodonNum = Trim(Left(GetNums(MutCall), CHalfway) / 3)
```

```
                'If CHalfway is a decimal
                Else
                    'SearchforCodon is the start number of the mutation (number with fewer
digits)
                    CodonNum = Application.WorksheetFunction.RoundUp(Left(GetNums(MutCall),
Int(CHalfway)) / 3, 0)
                    GoTo CheckMatch
                End If
            End If
CheckMatch:
        If CodonNum >= CInt(StartRange) And CodonNum <= CInt(EndRange) Then
'Change text to MUTANT
MutantText:
            Sheets("Final Report Summary").Range("F" & CStr(i)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
            Exit For
        End If
    End If
End If
End If
Next j
Next i
End Sub
Attribute VB_Name = "Module45"
Attribute VB_Name = "Module46"
Sub medi_report_5_table1()
'Compiles table 1 in the Meditech Report
'Table includes pertinent genes, chr, result, and coverage
Sheets("Meditech Report").Select
MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
'last row of data in report plus one
Dim TumorRow As String
Dim StartRowGene As Integer
Dim StartRow As Integer
Dim MutantGene As String
Dim IsNewGene As Integer
Dim MutantChr As String
Dim MutantCov As String
IsNewGene = 0
Sheets("Meditech Report").Select
'SET TITLE
    Range("A" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "Table 1: Mutation status of Genes with clinical relevance ---------
-------------------------------------------"
    Range("A" & CStr(MyRow + 2)).Select
    ActiveCell.FormulaR1C1 = "            "
    Range("B" & CStr(MyRow + 2)).Select
    ActiveCell.FormulaR1C1 = "            "
    Range("C" & CStr(MyRow + 2)).Select
    ActiveCell.FormulaR1C1 = "            "
    Range("D" & CStr(MyRow + 2)).Select
    ActiveCell.FormulaR1C1 = ">500X"
MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
    Range("A" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "GENE        "
    Range("B" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "CHR         "
    Range("C" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "RESULT      "
    Range("D" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "COVERAGE ------------------------------------------------------"
'DETERMINE TUMOR TYPE
Dim TumText As String
Dim TumRow As Integer
MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
StartRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
'Identify tumor from demographics
TumText = Sheets("Demographics").Range("B14").Text
'Find list of ROI genes in database
For i = 2 To Sheets("Tumor ROI Options").Range("A" & Rows.Count).End(xlUp).Row
If Sheets("Tumor ROI Options").Range("A" & CStr(i)).Text = TumText Then
```

```
TumRow = Sheets("Tumor ROI Options").Range("A" & CStr(i)).Row
Exit For
End If
Next i
'COMPILE TABLE
'ADD GENES based on Tumor ROI Options page
'For all ROI genes for the given tumor type
Sheets("Tumor ROI Options").Select
For i = 2 To Sheets("Tumor ROI Options").Range(Range("A" & CStr(TumRow)),
Selection.End(xlToRight)).End(xlToRight).Column
    Sheets("Meditech Report").Select
    Sheets("Meditech Report").Range("A" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = Sheets("Tumor ROI Options").Cells(TumRow, i)
    MyRow = MyRow + 1
Next i
Sheets("Meditech Report").Select
'IF ANY GENES WITH MUTANT RESULT NOT INCLUDED ABOVE
For i = 4 To 26
    'If mutation in row
    If Len(Sheets("Final Report Details").Range("D" & CStr(i)).Text) > 0 Then
        If Sheets("Final Report Details").Range("A" & CStr(i)).Text = True Then
        MutantGene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
        MutantChr = Sheets("Final Report Details").Range("B" & CStr(i)).Text
        MutantCov = Sheets("Final Report Details").Range("F" & CStr(i)).Value
        If MutantCov = "-" Then
            MutantCov = Sheets("Final Report Details").Range("E" & CStr(i)).Value
            End If
        'For each gene in Table 1
        For j = StartRow To Sheets("Meditech Report").Range("A" & CStr(StartRow)).End(xlDown).Row
        'IF gene is already included in table
        If Sheets("Meditech Report").Range("A" & CStr(j)).Text = MutantGene Then
            'Exit For; IsNewGene remains 0
            IsNewGene = 0
            Exit For
        'IF gene is new; not included in table
        Else
            'Add 1 to IsNewGene to indicate that the mutated gene is not yet included in table
            IsNewGene = 1
        End If
        'Next gene in Table 1
        Next j
        'Add gene to table if not yet included AND mutant
        If IsNewGene = 1 Then
MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
            Sheets("Meditech Report").Select
            Sheets("Meditech Report").Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MutantGene
            Sheets("Meditech Report").Range("B" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MutantChr
            Sheets("Meditech Report").Range("C" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
            Sheets("Meditech Report").Range("D" & CStr(MyRow)).Select
            If MutantCov >= 500 Then
                ActiveCell.FormulaR1C1 = "Yes"
            Else
                ActiveCell.FormulaR1C1 = "<500X"
            End If
        End If
        End If
    End If
'Next ROI gene to check if mutant
Next i
'If any MRD genes not included
For i = 22 To 31
Sheets("Demographics").Select
If IsEmpty(Range("B" & CStr(i))) Then
    Exit For
Else
MRDGene = Range("B" & CStr(i)).Text
MRDCodon = Range("C" & CStr(i)).Text
MRDChr = Range("D" & CStr(i)).Text
```

```
MRDPos = Range("E" & CStr(i)).Text
End If
Sheets("Meditech Report").Select
'For each gene in Table 1
        For j = StartRow To Sheets("Meditech Report").Range("A" & CStr(StartRow)).End(xlDown).Row
        'IF gene is already included in table
        If Sheets("Meditech Report").Range("A" & CStr(j)).Text = MRDGene Then
            'Exit For; IsNewGene remains 0
            IsNewGene = 0
            Exit For
        'IF gene is new; not included in table
        Else
            'Add 1 to IsNewGene to indicate that the mutated gene is not yet included in table
            IsNewGene = 1
        End If
        'Next gene in Table 1
        Next j
        'Add gene to table if not yet included
        If IsNewGene = 1 Then
        MyRow = Sheets("Meditech Report").Range("A" & Rows.Count).End(xlUp).Row + 1
            Sheets("Meditech Report").Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MRDGene
        End If
Next i
Sheets("Meditech Report").Select
'RESULT
'ADD MUTANT
'Start checking at the beginning of Table 1
CheckRow = StartRow
While Len(Sheets("Meditech Report").Range("A" & CStr(CheckRow)).Value) > 0
'For all rows in the Perinent Negatives Table
For i = 24 To Sheets("Final Report Summary").Range("A24").End(xlDown).Row
'If already mutant result; exit for
If Sheets("Meditech Report").Range("C" & CStr(CheckRow)).Value = "MUTANT" Then
    Exit For
'IF MUTANT AND MATCHES GENE
ElseIf Sheets("Final Report Summary").Range("F" & CStr(i)).Text = "MUTANT" And Sheets("Final
Report Summary").Range("A" & CStr(i)).Text = Sheets("Meditech Report").Range("A" &
CStr(CheckRow)).Text Then
Sheets("Meditech Report").Range("C" & CStr(CheckRow)).Select
ActiveCell.FormulaR1C1 = "MUTANT"
Exit For
End If
Next i
CheckRow = CheckRow + 1
Wend
'ADD WILDTYPE RESULT
For i = StartRow To Sheets("Meditech Report").Range("A" & CStr(StartRow)).End(xlDown).Row
If Sheets("Meditech Report").Range("C" & CStr(i)).Text = "MUTANT" Then
    'do nothing
Else
    Sheets("Meditech Report").Range("C" & CStr(i)).Select
    ActiveCell.FormulaR1C1 = "Wildtype"
End If
Next i
'ADD CHROMOSOME
CheckRow = StartRow
'For all of Table 1
While Len(Sheets("Meditech Report").Range("A" & CStr(CheckRow)).Text) > 0
For i = 24 To Sheets("Final Report Summary").Range("B24").End(xlDown).Row
'If chromosome already present, exit for
If Len(Sheets("Meditech Report").Range("B" & CStr(CheckRow)).Text) > 0 Then
    Exit For
'If Genes Match
ElseIf Sheets("Meditech Report").Range("A" & CStr(CheckRow)).Text = "FLT3-TKI" Then
    If "FLT3" = Sheets("Final Report Summary").Range("A" & CStr(i)).Text Then GoTo AddInfo
ElseIf Sheets("Meditech Report").Range("A" & CStr(CheckRow)).Text = Sheets("Final Report
Summary").Range("A" & CStr(i)).Text Then
AddInfo:
'ADD Chromosome
Sheets("Meditech Report").Range("B" & CStr(CheckRow)).Select
```

```
ActiveCell.Formula = "=IF(LEN(LEFT('Final Report Summary'!B" & CStr(i) & ",(FIND(" & """,""" &
",'Final Report Summary'!B" & CStr(i) & ",1)-1)))=1,CONCATENATE((LEFT('Final Report Summary'!B" &
CStr(i) & ",(FIND(" & """,""" & ",'Final Report Summary'!B" & CStr(i) & ",1)-1))),"" & """
""),CONCATENATE((LEFT('Final Report Summary'!B" & CStr(i) & ",(FIND(" & """,""" & ",'Final Report
Summary'!B" & CStr(i) & ",1)-1))),"" & """  """ & "))"
'ADD <500X to Table
If Sheets("Final Report Summary").Range("E" & CStr(i)).Value < 500 Then
Sheets("Meditech Report").Range("D" & CStr(CheckRow)).Select
ActiveCell.FormulaR1C1 = "<500X"
Else
Sheets("Meditech Report").Range("D" & CStr(CheckRow)).Select
ActiveCell.FormulaR1C1 = "Yes"
End If
Exit For
End If
Next i
CheckRow = CheckRow + 1
Wend
'CORRECT SPACING
Dim l As Byte, w As Byte
Dim CWidth(6) As Integer, X As Integer
CWidth(1) = 12
CWidth(2) = 12
CWidth(3) = 12
CWidth(4) = 12
'CWidth(5) = 1
'CWidth(6) = 1
X = 1
For Each C In Sheets("Meditech Report").Range("A" & CStr(StartRow) & ":D" & Range("A" &
Rows.Count).End(xlUp).Row)
                'Repeat as long as there is data
    With C
        l = Len(.Value)
        'l = number of chars in the cell
        .NumberFormat = "@"
        'change the cell to Number Format
        .Value = .Value & Space(CWidth(C.Column) - l)
        'add trailing spaces
    End With
Next
'Sort table alphabetically
LastRow = Cells(Rows.Count, 4).End(xlUp).Row
    ActiveWorkbook.Worksheets("Meditech Report").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Meditech Report").Sort.SortFields.Add Key:=Range("A1:A1"),
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Meditech Report").Sort
        .SetRange Range("A" & CStr(StartRow) & ":D" & CStr(LastRow))
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
End Sub
Attribute VB_Name = "Module47"
Sub SUPER_MY()
Dim strFileName As String
strFileName = ActiveWorkbook.Name
'ERROR MESSAGES
'TAB VC
Sheets("1% Supercontrol Allele Coverage").Select
Sheets("1% Supercontrol Allele Coverage").Range("A1").Select
If IsEmpty(Selection) Then
MsgBox "Please paste Supercontrol Allele Coverage data."
Exit Sub
End If
'TAB Nextgene
Sheets("1% Supercontrol NG").Select
If IsEmpty(Sheets("1% Supercontrol NG").Range("A1")) Then
MsgBox "Please paste Supercontrol Nextgene data."
Exit Sub
```

```
End If
'No Lot #
Sheets("Supercontrol Results- Myeloid").Select
If IsEmpty(Sheets("Supercontrol Results- Myeloid").Range("C13")) Then
MsgBox "Please enter Lot#."
Exit Sub
End If
'MACROS
    'Allele coverage
    Sheets("1% Supercontrol Allele Coverage").Select
    Application.Run "'" & strFileName & "'" & "!super_prepare_vc"
    'Variant caller
    Sheets("1% Supercontrol VC").Select
    Application.Run "'" & strFileName & "'" & "!super_format_vc"
    Application.Run "'" & strFileName & "'" & "!super_my_find_muts_vc"
    'Nextgene
    Sheets("1% Supercontrol NG").Select
    Application.Run "'" & strFileName & "'" & "!pre_prepare_ng"
    Application.Run "'" & strFileName & "'" & "!super_format_ng"
    Application.Run "'" & strFileName & "'" & "!super_my_find_muts_ng"
    'Compile results
    Sheets("Supercontrol Results- Myeloid").Select
    Application.Run "'" & strFileName & "'" & "!super_my_nomut_vc"
    Application.Run "'" & strFileName & "'" & "!super_my_nomut_ng"
    Application.Run "'" & strFileName & "'" & "!super_my_avg_freq"
    Application.Run "'" & strFileName & "'" & "!super_my_coverage"
    Application.Run "'" & strFileName & "'" & "!super_addtitle"
End Sub
Sub super_my_find_muts_vc()
Dim LSearchRow As Integer
Dim i As Integer
Dim GeneCounter As Integer
Sheets("1% Supercontrol VC").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("1% Supercontrol VC").Range("C" & CStr(LSearchRow)).Value) > 0
'Last row of data in Supercontrol mutation sheet
GeneCounter = Sheets("Supercontrol Results- Myeloid").Range("A1").End(xlDown).Row
For i = 2 To GeneCounter
'If FLT3
If Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text = "FLT3" Then
If Sheets("1% Supercontrol VC").Range("C" & CStr(LSearchRow)).Value = 28608269 Or Sheets("1%
Supercontrol VC").Range("C" & CStr(LSearchRow)).Value = 28608263 Then
Sheets("1% Supercontrol VC").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
'If any other gene
Else
'If the positions match
If Sheets("1% Supercontrol VC").Range("C" & CStr(LSearchRow)).Value = Sheets("Supercontrol
Results- Myeloid").Range("C" & CStr(i)).Value Then
Sheets("1% Supercontrol VC").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
```

```
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
    'ELSE IF COSMIC ID Matches
    ElseIf Sheets("1% Supercontrol VC").Range("M" & CStr(LSearchRow)).Value =
Sheets("Supercontrol Results- Myeloid").Range("F" & CStr(i)).Value Then
Sheets("1% Supercontrol VC").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
End If
Next i
LSearchRow = LSearchRow + 1
Wend
Range("A1").Select
Exit Sub
End Sub
Sub super_my_find_muts_ng()
Dim LSearchRow As Integer
Dim i As Integer
Dim GeneCounter As Integer
Sheets("1% Supercontrol NG").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("1% Supercontrol NG").Range("E" & CStr(LSearchRow)).Value) > 0
'Last row of data in Supercontrol mutation sheet
GeneCounter = Sheets("Supercontrol Results- Myeloid").Range("A1").End(xlDown).Row
For i = 2 To GeneCounter
'If FLT3
If Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text = "FLT3" Then
If Sheets("1% Supercontrol NG").Range("D" & CStr(LSearchRow)).Value = 28608269 Or Sheets("1%
Supercontrol NG").Range("D" & CStr(LSearchRow)).Value = 28608263 Then
Sheets("1% Supercontrol NG").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
Else
'If the positions match
If Sheets("1% Supercontrol NG").Range("D" & CStr(LSearchRow)).Value = Sheets("Supercontrol
Results- Myeloid").Range("C" & CStr(i)).Value Then
Sheets("1% Supercontrol NG").Select
Range("A" & CStr(LSearchRow)).Select
```

```
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
        'ELSE if c. numbers match
        ElseIf Sheets("1% Supercontrol NG").Range("R" & CStr(LSearchRow)).Value =
Sheets("Supercontrol Results- Myeloid").Range("D" & CStr(i)).Value Then
        Sheets("1% Supercontrol NG").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Supercontrol Results- Myeloid").Range("A" & CStr(i)).Text & "-
Mutation"
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
    With Selection.Interior
        .ColorIndex = 35
        .Pattern = xlSolid
    End With
Sheets("Supercontrol Results- Myeloid").Select
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "YES"
i = GeneCounter
End If
End If
Next i
LSearchRow = LSearchRow + 1
Wend
Range("A1").Select
Exit Sub
End Sub
Sub super_my_nomut_vc()
Dim GeneCounter As Integer
GeneCounter = Sheets("Supercontrol Results- Myeloid").Range("A2").End(xlDown).Row
For i = 2 To GeneCounter
If Range("G" & CStr(i)).Text = "YES" Then
    'do nothing
Else
Range("G" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "NO"
End If
Next i
End Sub
Sub super_my_avg_freq()
Dim LSearchRow As Integer
Dim VCValue As Single
Dim NGValue As Single
'Set variables
LSearchRow = 2
VCValue = Empty
NGValue = Empty
'VCValue & NGValue will remain Empty if no mutation is found
'For each mutation on Results Page
While Len(Range("A" & CStr(LSearchRow)).Text) > 0
'VARIANT CALLER
For i = 6 To Sheets("1% Supercontrol VC").Range("C" & Rows.Count).End(xlUp).Row
'Found in VC
If Sheets("1% Supercontrol VC").Range("C" & CStr(i)).Value = Sheets("Supercontrol Results-
Myeloid").Range("C" & CStr(LSearchRow)).Value And IsNumeric(Sheets("1% Supercontrol
VC").Range("H" & CStr(i)).Value) Then
    VCValue = Sheets("1% Supercontrol VC").Range("H" & CStr(i)).Value
    'If value is found, exit loop and proceed to next step
    Exit For
```

```
End If
Next i
'NEXTGENE
For j = 6 To Sheets("1% Supercontrol NG").Range("B" & Rows.Count).End(xlUp).Row
'Found in NG
If Sheets("1% Supercontrol NG").Range("D" & CStr(j)).Value = Sheets("Supercontrol Results-
Myeloid").Range("C" & CStr(LSearchRow)).Value And IsNumeric(Sheets("1% Supercontrol
NG").Range("S" & CStr(j)).Value) Then
NGValue = Sheets("1% Supercontrol NG").Range("S" & CStr(j)).Value
    Exit For
    'If value is found, exit loop and proceed to next step
End If
Next j
'If mutation is NOT found in either
If IsEmpty(VCValue) And IsEmpty(NGValue) Then
    'do nothing
'If mutation is not found in VC but found in NG
ElseIf VCValue = 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = NGValue
'If mutation is found in VC but not found in NG
ElseIf VCValue > 0 And NGValue = 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = VCValue
'If mutation is found in BOTH
ElseIf VCValue > 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("I" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = (VCValue + NGValue) / 2
End If
'Clear Variables
VCValue = Empty
NGValue = Empty
'Search for next mutation
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub super_my_coverage()
Dim LSearchRow As Integer
Dim VCValue As Single
Dim NGValue As Single
'Set variables
LSearchRow = 2
VCValue = Empty
NGValue = Empty
'VCValue & NGValue will remain Empty if no mutation is found
'For each mutation on Results Page
While Len(Range("A" & CStr(LSearchRow)).Text) > 0
'VARIANT CALLER
For i = 6 To Sheets("1% Supercontrol VC").Range("C" & Rows.Count).End(xlUp).Row
'Found in VC
If Sheets("1% Supercontrol VC").Range("C" & CStr(i)).Value = Sheets("Supercontrol Results-
Myeloid").Range("C" & CStr(LSearchRow)).Value Then
    VCValue = Sheets("1% Supercontrol VC").Range("T" & CStr(i)).Value
    'If value is found, exit loop and proceed to next step
    Exit For
End If
Next i
'NEXTGENE
For j = 6 To Sheets("1% Supercontrol NG").Range("B" & Rows.Count).End(xlUp).Row
'Found in NG
If Sheets("1% Supercontrol NG").Range("D" & CStr(j)).Value = Sheets("Supercontrol Results-
Myeloid").Range("C" & CStr(LSearchRow)).Value Then
NGValue = Sheets("1% Supercontrol NG").Range("I" & CStr(j)).Value
    Exit For
    'If value is found, exit loop and proceed to next step
End If
Next j
'If mutation is NOT found in either
```

```
If IsEmpty(VCValue) And IsEmpty(NGValue) Then
    'do nothing
'If mutation is not found in VC but found in NG
ElseIf VCValue = 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = NGValue
'If mutation is found in VC but not found in NG
ElseIf VCValue > 0 And NGValue = 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = VCValue
'If mutation is found in BOTH
ElseIf VCValue > 0 And NGValue > 0 Then
Sheets("Supercontrol Results- Myeloid").Select
Sheets("Supercontrol Results- Myeloid").Range("J" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = (VCValue + NGValue) / 2
End If
'Clear Variables
VCValue = Empty
NGValue = Empty
'Search for next mutation
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub super_my_nomut_ng()
Dim GeneCounter As Integer
GeneCounter = Sheets("Supercontrol Results- Myeloid").Range("A2").End(xlDown).Row
For i = 2 To GeneCounter
If Range("H" & CStr(i)).Text = "YES" Then
    'do nothing
Else
Range("H" & CStr(i)).Select
ActiveCell.FormulaR1C1 = "NO"
End If
Next i
End Sub
Sub super_addtitle()
    Rows("1:1").Select
    Selection.Insert Shift:=xlDown
    Range("A1").Select
    Rows("1:1").RowHeight = 47.25
    Range("E1").Select
    ActiveCell.FormulaR1C1 = "Ion PGM Supercontrol Metrics"
    With ActiveCell.Characters(Start:=1, Length:=28).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 16
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("H17").Select
    Rows("1:1").RowHeight = 32.25
End Sub
Attribute VB_Name = "Module48"
Attribute VB_Name = "Module49"
Sub medi_report_6_end_of_report()
'Adds end of report, including Table 2 and references
'Specific to tumor type
Dim TumorType As String
TumorType = Sheets("Demographics").Range("B14").Text
'Find end of report that coordinates with tumor type
Sheets("End of Report by Tumor").Select
For i = 1 To Sheets("End of Report by Tumor").Range("A1").End(xlToRight).Column
If Sheets("End of Report by Tumor").Cells(1, i).Text = TumorType Then
Range(Cells(2, i), Cells(Cells(Rows.Count, i).End(xlUp).Row, i)).Select
```

```
Selection.Copy
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
Range("A" & CStr(MyRow)).PasteSpecial xlPasteValues
End If
Next i
End Sub
Sub pre_format_color_vc()
'Adds color key to Variant Caller data
Sheets("Variant Caller").Select
   Range("E1").Select
    ActiveCell.FormulaR1C1 = "Color Key:"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Green:Artifact"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Blue:Artifact"
    Range("E4").Select
    ActiveCell.FormulaR1C1 = ""
    Range("F1").Select
    ActiveCell.FormulaR1C1 = "Red: Less than mutation frequency threshold"
    Range("F2").Select
    ActiveCell.FormulaR1C1 = "Orange: Less than 500X coverage"
    Range("F3").Select
    ActiveCell.FormulaR1C1 = _
        "Pink: Less than mutation frequency threshold and less than 500X coverage"
    Range("E1:F3").Select
    Range("F3").Activate
    Selection.Font.Bold = True
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Green:SNP"
    Range("E2").Select
    Selection.Font.ColorIndex = 10
    Range("E3").Select
    Selection.Font.ColorIndex = 41
    Range("F1").Select
    Selection.Font.ColorIndex = 3
    Range("F3").Select
    Selection.Font.ColorIndex = 5
    Range("F2").Select
    Selection.Font.ColorIndex = 45
    Range("F3").Select
    Selection.Font.ColorIndex = 7
    Range("E1:N3").Select
    Range("E1:N3").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("L4").Select
End Sub
Attribute VB_Name = "Module5"
```

```
Sub pre_filter_snps_ng()
'Filters SNPs in Nextgene data based on SNPs in template database
'SNP variants changed to green
'Variables
Dim LSearchRow As Integer
    'Row of nextgene data
Dim SNPCounter As Integer
    'Last row of data in SNP database
Sheets("Nextgene").Select
'Start search in row 6
LSearchRow = 6
'For all of the nextgene data
While Len(Sheets("Nextgene").Range("D" & CStr(LSearchRow)).Value) > 0
'Last row of data in SNP sheet
SNPCounter = Sheets("SNP Database").Range("A" & Rows.Count).End(xlUp).Row
'For each SNP in the database
For i = 7 To SNPCounter
'If value in Column D (chr position) matches position of a SNP in database
If Sheets("Nextgene").Range("D" & CStr(LSearchRow)).Text = Sheets("SNP Database").Range("A" &
CStr(i)).Text Then
'Change text in review column of Nextgene data to SNP type (entered in database)
Sheets("Nextgene").Select
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("SNP Database").Range("G" & CStr(i)).Value
'And make entire row green
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 10
'If SNP found, move to next variant in nextgene data
Exit For
End If
'If SNP not found, check variant position againt next SNP in database
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module50"
Sub post_qm_sort()
'Sorts QM database
'Moves all mutations to top so that they can be copied together
Sheets("QM Data").Select
    Rows("3:23").Select
    ActiveWorkbook.Worksheets("QM Data").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("QM Data").Sort.SortFields.Add Key:=Range( _
        "AN3:AN23"), SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:= _
        xlSortNormal
    With ActiveWorkbook.Worksheets("QM Data").Sort
        .SetRange Range("A3:AP23")
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
End Sub
Sub post_sum_mrd()
'Adds MRD genes to final report if AML
Dim GeneStr As String 'string of MRD genes from demographics
Sheets("Final Report Summary").Select
'If AML
If Sheets("Demographics").Range("B14").Text = "AML" Then
    'If MRD positive
    If Sheets("Demographics").Range("B20").Text = "Yes" Then
    'For each MRD gene
    For i = 22 To 31
    If Len(Sheets("Demographics").Range("B" & CStr(i)).Text) > 0 Then
    MRDGene = Sheets("Demographics").Range("B" & CStr(i)).Text
    MRDPos = Sheets("Demographics").Range("E" & CStr(i)).Text
    MRDCodon = Sheets("Demographics").Range("C" & CStr(i)).Text
        Sheets("Final Report Details").Select
        For j = 4 To 26
        'If gene and position match and TRUE
```

```
        If Range("D" & CStr(j)).Text = MRDGene And Range("C" & CStr(j)).Text = MRDPos And
Range("A" & CStr(j)).Text = True Then
            If Len(PosGenes) = 0 Then
                If MRDCodon = "FS" Then
                PosGenes = MRDGene & " " & MRDCodon & "."
                Else
                PosGenes = MRDGene & " codon " & MRDCodon & "."
                End If
            Else
                If MRDCodon = "FS" Then
                PosGenes = PosGenes & ", " & MRDGene & " " & MRDCodon & "."
                Else
                PosGenes = PosGenes & ", " & MRDGene & " codon " & MRDCodon & "."
                End If
            End If
        Exit For
        'If does not match
        Else
            If j = 26 Then
                If Len(NegGenes) = 0 Then
                    If MRDCodon = "FS" Then
                    NegGenes = MRDGene & " " & MRDCodon & "."
                    Else
                    NegGenes = MRDGene & " codon " & MRDCodon & "."
                    End If
                Else
                    If MRDCodon = "FS" Then
                    NegGenes = NegGenes & MRDGene & " " & MRDCodon & "."
                    Else
                    NegGenes = NegGenes & ", " & MRDGene & " codon " & MRDCodon & "."
                    End If
                End If
            End If
        End If
        Next j
    End If
    Next i
    'Add note on Final Report Summary page
    Sheets("Final Report Summary").Select
    Range("D13").Select
    If Len(PosGenes) = 0 Then
        ActiveCell.FormulaR1C1 = "This MRD specimen was negative for the following variants
detected at baseline: " & NegGenes & "."
    ElseIf Len(NegGenes) = 0 Then
        ActiveCell.FormulaR1C1 = "This MRD specimen was positive for the following variants
detected at baseline: " & PosGenes & ". "
    Else
    ActiveCell.FormulaR1C1 = "This MRD specimen was positive for the following variants detected
at baseline: " & PosGenes & " This MRD specimen was negative for the following variants detected
at baseline: " & NegGenes
    End If
    'Format Cell
    Range("D13").Select
    Selection.Font.Bold = True
    With Selection.Font
        .Color = -4165632
        .TintAndShade = 0
    End With
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
```

```
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    End If
End If
End Sub
Sub pre_format_color_ng()
'Adds color key to Nextgene data
Sheets("Nextgene").Select
    Range("G2").Select
    ActiveCell.FormulaR1C1 = "Color Ke"
    With ActiveCell.Characters(Start:=1, Length:=8).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("G2").Select
    Selection.ClearContents
    Range("G2").Select
    ActiveCell.FormulaR1C1 = "Color Key"
    With ActiveCell.Characters(Start:=1, Length:=9).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("G3").Select
    ActiveCell.FormulaR1C1 = "Green:SNP"
    With ActiveCell.Characters(Start:=1, Length:=9).Font
        .Name = "Arial"
        .FontStyle = "Bold"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .ColorIndex = xlAutomatic
    End With
    Range("G4").Select
    ActiveCell.FormulaR1C1 = "Blue:Artifact"
    Range("G4").Select
    Selection.Font.Bold = True
    Range("H2").Select
    ActiveCell.FormulaR1C1 = "Red: Less than mutation frequency threshold"
```

```
    Range("H2").Select
    ActiveCell.FormulaR1C1 = "Red:Less than mutation frequency threshold"
    Range("H2").Select
    Selection.Font.Bold = True
    Range("H3").Select
    ActiveCell.FormulaR1C1 = "Orange:Less than 500X coverage"
    Range("H3").Select
    Selection.Font.Bold = True
    Range("H4").Select
    ActiveCell.FormulaR1C1 = _
        "Pink:Less than mutation frequency threshold AND less than 500X coverage"
    Range("H4").Select
    Selection.Font.Bold = True
    Range("G3").Select
    Selection.Font.ColorIndex = 10
    Range("G4").Select
    Selection.Font.ColorIndex = 41
    Selection.Font.ColorIndex = 5
    Range("H2").Select
    Selection.Font.ColorIndex = 3
    Range("H3").Select
    Selection.Font.ColorIndex = 45
    Range("H4").Select
    Selection.Font.ColorIndex = 7
    Range("H3").Select
    Selection.Font.ColorIndex = 46
    ActiveWindow.SmallScroll Down:=-12
    Selection.Font.ColorIndex = 45
    Range("G2:N4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .Weight = xlMedium
        .ColorIndex = xlAutomatic
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("C3").Select
End Sub
Sub medi_raw_1_result()
'Adds result of case to raw data
'Mutations detected or no mutations detected
Sheets("Meditech Raw Data").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2 'last row of data in report plus two
'If positive (ie. mutations detected)
If Range("A3").Text = "OVERALL RESULT: POSITIVE" Then
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "MUTATIONS DETECTED"
Else
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "NO MUTATIONS DETECTED"
End If
End Sub
Attribute VB_Name = "Module51"
Sub medi_raw_2_interp()
```

```
'Define variables
Dim LSearchRow As Integer
Dim Pos As String
Dim Gene As String
Dim MutCall As String
Dim AAChange As String
Dim Cosm As String
Dim MutFreq As String
Dim Chromo As String
Dim StrBias As String
Dim Coverage As Double
Dim MyRow As Integer
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
'last row of data in report plus one
Dim MutCounter As Integer
MutCounter = 0
'total of all mutations in final report
Dim GeneNum As Integer
GeneNum = 1
'gene number for interp text
'Check if mutations are present for Common Findings, Variant Caller, and nextgene
If Len(Sheets("Final Report Details").Range("B4").Text) = 0 Then
    If Len(Sheets("Final Report Details").Range("B20").Text) = 0 Then
        If Len(Sheets("Final Report Details").Range("B24").Text) = 0 Then
            Exit Sub
        End If
    End If
End If
'If at least oen mutation is present
'For all lines in Final Report
For i = 4 To 26
'If mutation (if length of text in column B is greater than 0)
If Len(Sheets("Final Report Details").Range("B" & CStr(i)).Text) > 0 Then
'Chromosome
Chromo = Sheets("Final Report Details").Range("B" & CStr(i)).Text
'Position
Pos = Sheets("Final Report Details").Range("C" & CStr(i)).Value
'Gene
Gene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
'Mutation Call
MutCall = Sheets("Final Report Details").Range("K" & CStr(i)).Text
'Amino Acid Change
AAChange = Sheets("Final Report Details").Range("L" & CStr(i)).Text
'COSMIC ID
Cosm = Sheets("Final Report Details").Range("M" & CStr(i)).Text
'Coverage; average of NG and VC coverage
If IsNumeric(Sheets("Final Report Details").Range("E" & CStr(i)).Value) And
IsNumeric(Sheets("Final Report Details").Range("F" & CStr(i)).Value) Then
    Coverage = Application.WorksheetFunction.Average(Sheets("Final Report Details").Range("E" &
CStr(i)).Value, Sheets("Final Report Details").Range("F" & CStr(i)).Value)
'Variant coverage
ElseIf IsNumeric(Sheets("Final Report Details").Range("E" & CStr(i)).Value) And
IsNumeric(Sheets("Final Report Details").Range("F" & CStr(i)).Value) = False Then
    Coverage = Sheets("Final Report Details").Range("E" & CStr(i)).Value
'NG coverage
ElseIf IsNumeric(Sheets("Final Report Details").Range("E" & CStr(i)).Value) = False And
IsNumeric(Sheets("Final Report Details").Range("F" & CStr(i)).Value) Then
    Coverage = Sheets("Final Report Details").Range("F" & CStr(i)).Value
End If
'Strand Bias
StrBias = Sheets("Final Report Details").Range("H" & CStr(i)).Value
'Mutation Frequency
    'use Nextgene frequency
    If i <= 18 Or i >= 24 Then
    MutFreq = Sheets("Final Report Details").Range("J" & CStr(i)).Text
    'otherwise use variant caller frequency
    Else
    MutFreq = Sheets("Final Report Details").Range("I" & CStr(i)).Text
    End If
'Add interpretation fields
'GENE
```

```
Sheets("Meditech Raw Data").Select
If GeneNum = 1 Then
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
Else
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
Range("A" & CStr(MyRow)).Select
End If
ActiveCell.FormulaR1C1 = "    GENE #" & CStr(GeneNum) & ":"
GeneNum = GeneNum + 1
'Add text to report
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Gene
'MUTATION
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "   MUTATION:"
'Define variables
'amino acid
Dim NewAA As String
    'modified Amino Acid change
Dim PartOne As String
    'section of AA change before ">"
Dim PartTwo As String
    'section of AA change after ">"
Dim LettersOnlyOne As String
    'single letter extracted from PartOne
Dim LettersOnlyTwo As String
    'two letters extracted from PartTwo
Dim MutNum As Integer
    'number extracted from AA change
'mutation call
Dim NewMutCall As String
    'modified Mutation Call
Dim PartOneCallOrig As String
    'section of MutCall before ">", including "c."
Dim PartOneCall As String
    'section of MutCall before ">", without "c."
Dim PartTwoCall As String
    'section of MutCall after ">"
Dim LettersOnlyOneCall As String
    'single letter extracted from PartOneCall
Dim LettersOnlyTwoCall As String
    'single letter extracted from PartTwoCall
Dim MutNumCall As Integer
    'number extracted from MutCall
'If FS, in-frame, or splice don't change anything, New is same as Original
If AAChange = "FS" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "In-Frame" Then
    NewAA = AAChange
    NewMutCall = MutCall
ElseIf AAChange = "Splice" Then
    NewAA = AAChange
    NewMutCall = MutCall
'If "-" (ie. Variant caller, skip to next part of function)
ElseIf AAChange = "-" Then
'Add "type" text and move to cosmic ID
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "       TYPE:"
    GoTo CosmicSection
'Otherwise
Else
'Determine Amino Acid
'
'
    'before ">"
PartOne = Left(AAChange, InStrRev(AAChange, ">") - 1)
    'after ">"
```

```
PartTwo = Right(AAChange, Len(AAChange) - InStrRev(AAChange, ">"))
'Extract letter (amino acid) from part 1
Dim X As Long
  For X = 1 To Len(PartOne)
    If Mid(PartOne, X, 1) Like "[!A-Za-z]" Then Mid(PartOne, X, 1) = " "
  Next
  LettersOnlyOne = Replace(PartOne, " ", "")
'Extract two letters
'This should have two letters, one that repeats with LettersOnlyOne
Dim Y As Long
  For Y = 1 To Len(PartTwo)
    If Mid(PartTwo, Y, 1) Like "[!A-Za-z]" Then Mid(PartTwo, Y, 1) = " "
  Next
  LettersOnlyTwo = Replace(PartTwo, " ", "")
'Determine which letter is a repeat and remove
    'IF FIRST LETTER MATCHES, REMOVE
If LettersOnlyOne = Left(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Right(LettersOnlyTwo, 1)
    'IF SECOND LETTER MATCHES, REMOVE
ElseIf LettersOnlyOne = Right(LettersOnlyTwo, 1) Then
LettersOnlyTwo = Left(LettersOnlyTwo, 1)
End If
'Get number from original AACall
MutNum = GetNums(AAChange)
'Create NewAA
NewAA = "p." & LettersOnlyOne & MutNum & LettersOnlyTwo
'Determine Mutation Call
'
'
    'before ">"
PartOneCallOrig = Left(MutCall, InStrRev(MutCall, ">") - 1)
    'Take away c. at beginning
PartOneCall = Right(PartOneCallOrig, Len(PartOneCallOrig) - 2)
    'after ">"
PartTwoCall = Right(MutCall, Len(MutCall) - InStrRev(MutCall, ">"))
'Extract letter from PartOneCall
Dim Z As Long
  For Z = 1 To Len(PartOneCall)
    If Mid(PartOneCall, Z, 1) Like "[!A-Za-z]" Then Mid(PartOneCall, Z, 1) = " "
  Next
  LettersOnlyOneCall = Replace(PartOneCall, " ", "")
'Extract two letters from PartTwoCall
'This should have two letters, one that repeats with LettersOnlyOneCall
Dim a As Long
  For a = 1 To Len(PartTwoCall)
    If Mid(PartTwoCall, a, 1) Like "[!A-Za-z]" Then Mid(PartTwoCall, a, 1) = " "
  Next
  LettersOnlyTwoCall = Replace(PartTwoCall, " ", "")
'Determine which base is a repeat and remove
    'IF FIRST LETTER MATCHES, REMOVE
If LettersOnlyOneCall = Left(LettersOnlyTwoCall, 1) Then
LettersOnlyTwoCall = Right(LettersOnlyTwoCall, 1)
    'IF SECOND LETTER MATCHES, REMOVE
ElseIf LettersOnlyOneCall = Right(LettersOnlyTwoCall, 1) Then
LettersOnlyTwoCall = Left(LettersOnlyTwoCall, 1)
End If
'Get number from original MutCall
MutNumCall = GetNums(MutCall)
'New call
NewMutCall = "c." & MutNumCall & LettersOnlyOneCall & ">" & LettersOnlyTwoCall
End If
'Add text to report
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = NewAA & " (" & NewMutCall & ")"
'TYPE
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "       TYPE:"
Range("B" & CStr(MyRow)).Select
'Missense
If InStr(1, MutCall, ">") > 0 Then
```

```
ActiveCell.FormulaR1C1 = "Missense"
'Nonsense
ElseIf InStr(1, MutCall, "X") > 0 Then
ActiveCell.FormulaR1C1 = "Nonsense"
'Insertion
ElseIf InStr(1, MutCall, "ins") > 0 Then
ActiveCell.FormulaR1C1 = "Insertion"
'Deletion
ElseIf InStr(1, MutCall, "del") > 0 Then
ActiveCell.FormulaR1C1 = "Deletion"
End If
'COSMIC ID
CosmicSection:
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "   COSMIC #:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Cosm
'CHROMOSOME
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = " CHROMOSOME:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Chromo
'POSITION
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "   POSITION:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Pos
'COVERAGE
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "   COVERAGE:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Coverage
'MUTATION %
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "  MUTATION%:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = MutFreq
'STRAND BIAS
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "STRAND BIAS:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = StrBias
'CONFIRMED, SANGER DATE, AND PCR PRIMER
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "  CONFIRMED:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "na"
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "SANGER DATE:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "na"
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 1
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "PCR  PRIMER:"
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "na"
End If
Next i
End Sub
Sub medi_clear_raw()
Range("A15:G1000").ClearContents
'Reset Font, Size, and cell type
    Columns("A:D").Select
```

```
    With Selection.Font
        .Name = "Courier New"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    With Selection.Font
        .Name = "Courier New"
        .Size = 10
        .Strikethrough = False
        .Superscript = False
        .Subscript = False
        .OutlineFont = False
        .Shadow = False
        .Underline = xlUnderlineStyleNone
        .TintAndShade = 0
        .ThemeFont = xlThemeFontNone
    End With
    Selection.NumberFormat = "General"
Range("A1").Select
End Sub
Sub medi_raw_3_table()
'Compiles table in the Meditech Raw data
'Table includes pertinent gene, chr, pos, exon, codons, result, and coverage
Sheets("Meditech Raw Data").Select
MyRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 1
'last row of data in report plus one
Dim TumorRow As String
Dim StartRowGene As Integer
Dim StartRow As Integer
Dim MutantGene As String
Dim IsNewGene As Integer
Dim MutantChr As String
Dim MutantCov As String
IsNewGene = 0
Sheets("Meditech Raw Data").Select
'SET TITLE
    Range("A" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "        "
    Range("B" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "    "
    Range("C" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "CHR       "
    Range("D" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "     "
    Range("E" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "HOTSPOT "
    Range("F" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = "          "
    Range("G" & CStr(MyRow + 1)).Select
    ActiveCell.FormulaR1C1 = ">500X(*)"
MyRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 1
    Range("A" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "GENE    "
    Range("B" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "CHR "
    Range("C" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "POSITION  "
    Range("D" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "EXON "
    Range("E" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "CODONS   "
    Range("F" & CStr(MyRow)).Select
    ActiveCell.FormulaR1C1 = "RESULT   "
    Range("G" & CStr(MyRow)).Select
```

```
    ActiveCell.FormulaR1C1 = "COVERAGE ----------------------------------------------------"
'DETERMINE TUMOR TYPE
Dim TumText As String
Dim TumRow As Integer
MyRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 1
StartRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 1
'Identify tumor from demographics
TumText = Sheets("Demographics").Range("B14").Text
'COMPILE TABLE
Sheets("Meditech Raw Data").Select
'For all rows in the Perinent Negatives Table
For i = 24 To Sheets("Final Report Summary").Range("A24").End(xlDown).Row
'GENE
Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Final Report Summary").Range("A" & CStr(i)).Text
ChrPos = Sheets("Final Report Summary").Range("B" & CStr(i)).Text
'CHR
Range("B" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Left(ChrPos, (InStr(1, ChrPos, ",") - 1))
'POSITION
Range("C" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Right(ChrPos, (Len(ChrPos) - InStr(1, ChrPos, ",")))
'EXON
Range("D" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Final Report Summary").Range("C" & CStr(i)).Text
'HOTSPOT CODONS
Range("E" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Final Report Summary").Range("D" & CStr(i)).Text
'RESULT
Range("F" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Final Report Summary").Range("F" & CStr(i)).Text
'>500 coverage
If Sheets("Final Report Summary").Range("E" & CStr(i)).Value < 500 Then
Sheets("Meditech Raw Data").Range("G" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "<500X"
Else
Sheets("Meditech Raw Data").Range("G" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "Yes"
End If
MyRow = MyRow + 1
Next i
'IF ANY GENES WITH MUTANT RESULT NOT INCLUDED ABOVE
For i = 4 To 26
    'If mutation in row
    If Len(Sheets("Final Report Details").Range("D" & CStr(i)).Text) > 0 Then
        If Sheets("Final Report Details").Range("A" & CStr(i)).Text = True Then
        MutantGene = Sheets("Final Report Details").Range("D" & CStr(i)).Text
        MutantChr = Sheets("Final Report Details").Range("B" & CStr(i)).Text
        MutantCov = Sheets("Final Report Details").Range("F" & CStr(i)).Value
        If MutantCov = "-" Then
            MutantCov = Sheets("Final Report Details").Range("E" & CStr(i)).Value
            End If
        MutantPos = Sheets("Final Report Details").Range("C" & CStr(i)).Value
        'For each gene in Table 1
        For j = StartRow To Sheets("Meditech Raw Data").Range("A" &
CStr(StartRow)).End(xlDown).Row
        'IF gene is already included in table
        If Sheets("Meditech Raw Data").Range("A" & CStr(j)).Text = MutantGene Then
            'Exit For; IsNewGene remains 0
            IsNewGene = 0
            Exit For
        'IF gene is new; not included in table
        Else
            'Add 1 to IsNewGene to indicate that the mutated gene is not yet included in table
            IsNewGene = 1
        End If
        'Next gene in Table 1
        Next j
        'Add gene to table if not yet included AND mutant
        If IsNewGene = 1 Then
MyRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 1
```

```
            'Gene
            Sheets("Meditech Raw Data").Select
            Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MutantGene
            'Chromsome
            Sheets("Meditech Raw Data").Range("B" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MutantChr
            'Position
            Sheets("Meditech Raw Data").Range("C" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = MutantPos
            'Exon
            Sheets("Meditech Raw Data").Range("D" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = "na"
            'Hotspot Codons
            Sheets("Meditech Raw Data").Range("E" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = "na"
            'Result
            Sheets("Meditech Raw Data").Range("F" & CStr(MyRow)).Select
            ActiveCell.FormulaR1C1 = "MUTANT"
            'Coverage
            Sheets("Meditech Raw Data").Range("G" & CStr(MyRow)).Select
            If MutantCov >= 500 Then
                ActiveCell.FormulaR1C1 = "Yes"
            Else
                ActiveCell.FormulaR1C1 = "<500X"
            End If
        End If
        End If
    End If
'Next ROI gene to check if mutant
Next i
'CORRECT SPACING
Dim l As Byte, w As Byte
Dim CWidth(7) As Integer, X As Integer
CWidth(1) = 8
CWidth(2) = 4
CWidth(3) = 10
CWidth(4) = 5
CWidth(5) = 9
CWidth(6) = 9
CWidth(7) = 8
X = 1
For Each C In Sheets("Meditech Raw Data").Range("A" & CStr(StartRow) & ":G" & Range("A" &
Rows.Count).End(xlUp).Row)
                'Repeat as long as there is data
    With C
        l = Len(.Value)
        'l = number of chars in the cell
        .NumberFormat = "@"
        'change the cell to Number Format
        .Value = .Value & Space(CWidth(C.Column) - l)
        'add trailing spaces
    End With
Next
'Sort table alphabetically
LastRow = Cells(Rows.Count, 4).End(xlUp).Row
    ActiveWorkbook.Worksheets("Meditech Raw Data").Sort.SortFields.Clear
    ActiveWorkbook.Worksheets("Meditech Raw Data").Sort.SortFields.Add Key:=Range("A1:A1"),
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal
    With ActiveWorkbook.Worksheets("Meditech Raw Data").Sort
        .SetRange Range("A" & CStr(StartRow) & ":G" & CStr(LastRow))
        .Header = xlGuess
        .MatchCase = False
        .Orientation = xlTopToBottom
        .SortMethod = xlPinYin
        .Apply
    End With
End Sub
Sub medi_copy_raw()
'Copies entire meditech raw data report to clipboard on click of the button
Range("A1:G" & Range("A" & Rows.Count).End(xlUp).Row).SpecialCells(xlCellTypeVisible).Select
```

```
Selection.Copy
Range("A1").Select
End Sub
Attribute VB_Name = "Module52"
Attribute VB_Name = "Module53"
Sub pre_format_style()
Attribute pre_format_style.VB_ProcData.VB_Invoke_Func = " \n14"
'Adds borders and colors to formatted NG data
    Range("A5").Select
    Range(Selection, Selection.End(xlToRight)).Select
    Range(Selection, "D" & Selection.Range("D5").End(xlDown).Row).Select
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Columns("G:G").EntireColumn.AutoFit
'Format info box in top left
    Range("A1:B3").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
```

```
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Columns("B:B").EntireColumn.AutoFit
Range("A1:B1").Select
Range("B1").Activate
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Range("A1:B3").Select
Range("B3").Activate
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Range("A2:B3").Select
Range("B3").Activate
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
```

```
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Range("A3:B3").Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = -0.149998474074526
    .PatternTintAndShade = 0
End With
Range("A2:B2").Select
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent1
    .TintAndShade = 0.599993896298105
    .PatternTintAndShade = 0
End With
Range("A2:A3").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
```

```
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Range("B2:B3").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Range("A2:A3").Select
Range("A3").Activate
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlMedium
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
```

```
        .Weight = xlThin
    End With
    Range("A6").Select
End Sub
Sub post_highlight_roi_ng()
'Highlights ROI genes in nextgene data
Dim LSearchRow As Integer
Dim LastTableRow As Integer
Sheets("Nextgene").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("Nextgene").Range("B" & CStr(LSearchRow)).Value) > 0
'Last row of data in ROI table on Final Report Summary page
LastTableRow = Sheets("Final Report Summary").Range("A23").End(xlDown).Row
For i = 24 To LastTableRow
'If genes match
If Sheets("Final Report Summary").Range("A" & CStr(i)).Text = Sheets("Nextgene").Range("E" &
CStr(LSearchRow)).Text Then
'Highlight entire row
Sheets("Nextgene").Range("A" & CStr(LSearchRow) & ":U" & CStr(LSearchRow)).Select
With Selection.Interior
        .ColorIndex = 6
        .Pattern = xlSolid
    End With
End If
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Sub post_highlight_roi_vc()
'Highlights ROI genes in Variant Caler data
Dim LSearchRow As Integer
Dim LastTableRow As Integer
Sheets("Variant Caller").Select
'Start search in row 6
LSearchRow = 6
While Len(Sheets("Variant Caller").Range("B" & CStr(LSearchRow)).Value) > 0
'Last row of data in ROI table on Final Report Summary page
LastTableRow = Sheets("Final Report Summary").Range("A23").End(xlDown).Row
For i = 24 To LastTableRow
'If genes match
If Sheets("Final Report Summary").Range("A" & CStr(i)).Text = Sheets("Variant Caller").Range("P"
& CStr(LSearchRow)).Text Then
'Highlight entire row
Sheets("Variant Caller").Range("A" & CStr(LSearchRow) & ":AX" & CStr(LSearchRow)).Select
With Selection.Interior
        .ColorIndex = 6
        .Pattern = xlSolid
    End With
End If
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module54"
Sub MEDI_NOTE()
'Generates row for Meditech Note database
Sheets("Meditech Note Entry Program").Select
'Tumor Type
Range("A11").Select
ActiveCell.FormulaR1C1 = Range("B1").Text
'Description
Range("B11").Select
ActiveCell.FormulaR1C1 = Range("B6").Text
'Gene
Range("C11").Select
ActiveCell.FormulaR1C1 = Range("B2").Text
'Mutation
Range("D11").Select
ActiveCell.FormulaR1C1 = "p." & Range("B4").Text & Range("B3").Text & Range("B5").Text
'Note Text
```

```
Range("E11").Select
ActiveCell.FormulaR1C1 = Range("B7").Text
End Sub
Attribute VB_Name = "Module55"
Sub medi_raw_4_end_of_report()
'Adds end of Meditech Raw Data Report
Dim MyRow As Integer
MyRow = Sheets("Meditech Raw Data").Range("A" & Rows.Count).End(xlUp).Row + 2
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "GENE SEQUENCES WITH LESS THAN 500X COVERAGE:"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "None relevant to this tumor type"
MyRow = MyRow + 2
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "PERTINENT NEGATIVES"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "The following genes did not show any mutation that could be confirmed,
with the exception of those genes listed in the diagnosis or raw data above."
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "ABL1   EZH2   JAK3   PTEN"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "AKT1   FBXW7  IDH2   PTPN11"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "ALK    FGFR1  KDR    RB1"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "APC    FGFR2  KIT    RET"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "ATM    FGFR3  KRAS   SMAD4"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "BRAF   FLT3   MET    SMARCB1"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "CDH1   GNA11  MLH1   SMO"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "CDKN2A GNAS   MPL    SRC"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "CSF1R  GNAQ   NOTCH1 STK11"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "CTNNB1 HNF1A  NPM1   TP53"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "EGFR   HRAS   NRAS   VHL"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "ERBB2  IDH1   PDGFRA"
MyRow = MyRow + 1
Sheets("Meditech Raw Data").Range("A" & CStr(MyRow)).Select
ActiveCell.FormulaR1C1 = "ERBB4  JAK2   PIK3CA"
End Sub
Attribute VB_Name = "Module56"
Sub medi_report_2b_notes_other()
'Adds other Meditech notes to report based on whether user chooses to include them
Sheets("Meditech Notes- Other").Select
'For all notes in database
For i = 2 To Range("C2").End(xlDown).Row
If Range("A" & CStr(i)).Text = "Yes" Then
Range("C" & CStr(i)).Select
Selection.Copy
Sheets("Meditech Report").Select
MyRow = Range("A" & Rows.Count).End(xlUp).Row + 2
```

```
Range("A" & CStr(MyRow)).Select
Range("A" & CStr(MyRow)).PasteSpecial xlPasteValues
Sheets("Meditech Notes- Other").Select
End If
Next i
End Sub
Attribute VB_Name = "Module57"
Sub post_qm_hide()
Attribute post_qm_hide.VB_ProcData.VB_Invoke_Func = " \n14"
'Hides unneeded rows in QM data
Sheets("QM Data").Select
'If no mutations detected
If Range("AA3").Value = 0 Then
For j = 27 To 42
Cells(3, j).Select
ActiveCell.FormulaR1C1 = "na"
Next j
End If
For i = 4 To 23
If Range("AA" & CStr(i)).Value = 0 Then
Rows(CStr(i) & ":" & CStr(i)).Select
Selection.EntireRow.Hidden = True
End If
Next i
End Sub
Attribute VB_Name = "Module58"
Sub pre_format_dropdown_vc()
'Adds dropdown for Review column in VC
Sheets("Variant Caller").Select
'Find Last Row of Data
LastRow = Range("A" & Rows.Count).End(xlUp).Row
'For each row
For i = 6 To LastRow
'Data validation; use dropdown list
'references list on Demographic Options tab
Range("A" & CStr(i)).Select
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Review_Options"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
Next i
End Sub
Attribute VB_Name = "Module59"
Sub pre_format_dropdown_ng()
'Adds dropdown for Review column in NG
Sheets("Nextgene").Select
'Find last row of data
LastRow = Range("A" & Rows.Count).End(xlUp).Row
'For each row
For i = 6 To LastRow
'Data validation; use dropdown list
'references list on Demographic Options tab
Range("A" & CStr(i)).Select
    With Selection.Validation
        .Delete
        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
        xlBetween, Formula1:="=Review_Options"
        .IgnoreBlank = True
        .InCellDropdown = True
        .InputTitle = ""
        .ErrorTitle = ""
        .InputMessage = ""
```

```
        .ErrorMessage = ""
        .ShowInput = True
        .ShowError = True
    End With
Next i
End Sub
Attribute VB_Name = "Module6"
Sub pre_filter_artifacts_ng()
'Filters artifacts in Nextgene data based on artifacts in template database
'Artifact variants changed to blue
Dim LSearchRow As Integer
    'row in Nextgene data
Dim ArtCounter As Integer
    'final row in artifact database
'Start search in row 6
LSearchRow = 6
'For all Nextgene data
While Len(Sheets("Nextgene").Range("C" & CStr(LSearchRow)).Value) > 0
'Last row of data in Artifact database
ArtCounter = Sheets("Artifact Database").Range("C" & Rows.Count).End(xlUp).Row
'For each artifact in the database
For i = 2 To ArtCounter
'If variant position matches position of an artifact in the database
If Sheets("Nextgene").Range("D" & CStr(LSearchRow)).Text = Sheets("Artifact Database").Range("C"
& CStr(i)).Text Then
'Change text in column A to type of Artifact (specified by "Details" column in database)
Sheets("Nextgene").Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = Sheets("Artifact Database").Range("D" & CStr(i)).Text
'And make entire row blue
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 32
'If artifact, exit for and move to next row of data
Exit For
End If
'If no match, check variant against next artifact in database
Next i
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module60"
Sub post_print_footer()
'Adds footer to bottom of sheet with MDNumber, Patient Sex, and Patient Name
Dim MDNUm As Range
Dim PatientName As Range
Dim PatientSex As Range
On Error Resume Next
'Set Patient MD Number and Name to variables
Set MDNUm = Application.Sheets("Demographics").Range("B3")
Set PatientName = Application.Sheets("Demographics").Range("B2")
Set PatientSex = Application.Sheets("Demographics").Range("B5")
'Set right footer to include MD number, patient name, and patient sex
Application.ActiveSheet.PageSetup.RightFooter = "&11" & MDNUm.Value & "; " & PatientName.Value &
"; " & PatientSex.Value
End Sub
Attribute VB_Name = "Module61"
Sub post_format_columns_vc()
Attribute post_format_columns_vc.VB_ProcData.VB_Invoke_Func = " \n14"
'Adjusts size of columns for printing
Sheets("Variant Caller").Select
'Hide Columns
    Columns("G:G").Select
    Selection.EntireColumn.Hidden = True
    Columns("H:H").Select
    Selection.EntireColumn.Hidden = True
    Columns("J:J").Select
    Selection.EntireColumn.Hidden = True
    Columns("K:K").Select
    Selection.EntireColumn.Hidden = True
    Columns("O:O").Select
    Selection.EntireColumn.Hidden = True
    Columns("U:AV").Select
```

```
    Selection.EntireColumn.Hidden = True
'Autofit Columns
   Columns("A:A").EntireColumn.AutoFit
   Columns("N:N").EntireColumn.AutoFit
   Columns("P:S").EntireColumn.AutoFit
   Columns("T:T").EntireColumn.AutoFit
   Columns("AW:AX").EntireColumn.AutoFit
End Sub
Sub post_format_columns_ng()
'Adjusts size of columns for printing
Sheets("Nextgene").Select
'Hide Columns
    Columns("C:C").Select
    Selection.EntireColumn.Hidden = True
    Columns("D:D").Select
    Selection.EntireColumn.Hidden = True
    Columns("F:F").Select
    Selection.EntireColumn.Hidden = True
'Autofit/Resize Columns
   Columns("A").ColumnWidth = 15.86
   Columns("K:R").EntireColumn.AutoFit
   Columns("S:S").EntireColumn.AutoFit
   Columns("U:U").EntireColumn.AutoFit
End Sub
Attribute VB_Name = "Module62"
Attribute VB_Name = "Module63"
Attribute VB_Name = "Module64"
Sub pre_format_common()
'Finds variants that are common between Nextgene and Variant Caller data
'based on chromosome position
'Moves to top of data
Sheets("Variant Caller").Select
'VARIABLES
Dim VCRow As Integer 'line of data in variant caller
Dim LastNGRow As Integer 'last line of data in nextgene
'Set NextColumnVC
Range("A5").Select
NextColumnVC = Range(Range("A5"), Selection.End(xlToRight)).End(xlToRight).Column + 2
'Start search at row 6 in VC
VCRow = 6
LastVCRow = Sheets("Variant Caller").Range("D6").End(xlDown).Row
Sheets("Nextgene").Select
'Last row of data in NG
LastNGRow = Sheets("Nextgene").Range("D6").End(xlDown).Row
'Set NextColumnVC
Range("A5").Select
NextColumnNG = Range(Range("A5"), Selection.End(xlToRight)).End(xlToRight).Column + 2
Sheets("Variant Caller").Select
'For each row of data in variant caller
While Len(Sheets("Variant Caller").Range("C" & VCRow).Text) > 0
'For each row of data in NG
For i = 6 To LastNGRow
'If position in VC matches position in NG
If Sheets("Variant Caller").Range("C" & CStr(VCRow)).Value = Sheets("Nextgene").Range("D" &
CStr(i)).Value Then
'Increase common counter
Common = Common + 1
'Type "TRUE" in Common? column
'VARIANT CALLER
Cells(VCRow, NextColumnVC).Select
ActiveCell.FormulaR1C1 = "TRUE"
'NEXTGENE
Sheets("Nextgene").Select
'Type "TRUE" in Common? column
Cells(i, NextColumnNG).Select
ActiveCell.FormulaR1C1 = "TRUE"
Sheets("Variant Caller").Select
Exit For
'Else, if last i and no match
'VARIANT CALLER
ElseIf i = LastRowNG Then
```

```
If Len(Sheets("Variant Caller").Cells(VCRow, NextColumnVC).Text) > 0 Then
    'do nothing
Else
Cells(VCRow, NextColumnVC).Select
ActiveCell.FormulaR1C1 = "FALSE"
End If
'NEXTGENE
If Len(Sheets("Nextgene").Cells(i, NextColumnNG).Text) > 0 Then
    'do nothing
Else
Sheets("Nextgene").Select
Cells(i, NextColumnNG).Select
ActiveCell.FormulaR1C1 = "FALSE"
Sheets("Variant Caller").Select
End If
Else
'NEXTGENE
If Len(Sheets("Nextgene").Cells(i, NextColumnNG).Text) > 0 Then
    'do nothing
Else
Sheets("Nextgene").Select
Cells(i, NextColumnNG).Select
ActiveCell.FormulaR1C1 = "FALSE"
Sheets("Variant Caller").Select
End If
End If
'Check against next row in Nextgene
Next i
'Check next row of data in NG
VCRow = VCRow + 1
Wend
'Hide column
Sheets("Variant Caller").Select
Cells(1, NextColumnVC).Select
Selection.EntireColumn.Hidden = True
Sheets("Nextgene").Select
Cells(1, NextColumnNG).Select
Selection.EntireColumn.Hidden = True
'Sort unfiltered data to top and filtered data to bottom by filter type
Sheets("Variant Caller").Select
Set SortRange = Range(Cells(5, 1), Cells(LastVCRow, NextColumnVC))
Set NextColumnRange = Range(Cells(6, NextColumnVC), Cells(6, NextColumnVC))
SortRange.Sort Key1:=NextColumnRange, Order1:=xlDescending, Key2:= _
        Range("A6"), Order2:=xlDescending, Header:=xlGuess, OrderCustom:=1, _
        MatchCase:=False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, _
        DataOption2:=xlSortNormal
Sheets("Nextgene").Select
Set SortRange = Range(Cells(5, 1), Cells(LastNGRow, NextColumnNG))
Set NextColumnRange = Range(Cells(6, NextColumnNG), Cells(6, NextColumnNG))
SortRange.Sort Key1:=NextColumnRange, Order1:=xlDescending, Key2:= _
        Range("A6"), Order2:=xlDescending, Header:=xlGuess, OrderCustom:=1, _
        MatchCase:=False, Orientation:=xlTopToBottom, DataOption1:=xlSortNormal, _
        DataOption2:=xlSortNormal
End Sub
Attribute VB_Name = "Module7"
Sub pre_filter_lowcov_ng()
'Filters variants with coverage below 500
'Low coverage variants are changed to orange
'VARIABLES
Dim LSearchRow As Integer
    'row of Nextgene data
Dim CovThreshold As Integer
    'threshold for low coverage
Sheets("Nextgene").Select
'Start search in row 6
LSearchRow = 6
'Define coverage threshold.
CovThreshold = 500
'For all of the Nextgene data
While Len(Range("C" & CStr(LSearchRow)).Value) > 0
'If coverage is < CovThreshold
```

```
If Range("I" & CStr(LSearchRow)).Value < CovThreshold Then
'Then change text in column A to "< 500X COV"
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "<500X COV"
'And make entire row red
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 46
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module8"
Sub pre_filter_mutfreq_ng()
'Filters variants with mutation frequency below the given threshold.
'Threshold is determined by tumor percentage on Demographics page
'Low mutation frequency variants are changed to red
'VARIABLES
Dim LSearchRow As Integer
    'row of data in NG
Dim FreqThreshold As Integer
    'frequency threshold for specimen
'Start search in row 6
LSearchRow = 6
'Define frequency thresholds for given percent tumor
Sheets("Demographics").Select
FreqThreshold = Range("B16").Value
'For all of the data in Nextgene
Sheets("Nextgene").Select
While Len(Range("D" & CStr(LSearchRow)).Value) > 0
'If number in column S ("Mutant Allele Frequency") is < FreqThreshold
If Range("S" & CStr(LSearchRow)).Value < FreqThreshold Then
'Then change text in column A ("Review") to <MUT FREQ THRESH
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "<MUT FREQ THRESH"
'And make entire row red
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 3
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
Attribute VB_Name = "Module9"
Sub pre_filter_mutandcov_ng()
'Filters variants with coverage below 500 and low mutation frequency
'Variants are changed to pink
'VARIABLES
Dim LSearchRow As Integer
    'row of Nextgene data
Dim FreqThreshold As Integer
    'frequency threshold determined by tumor percentage
Dim CovThreshold As Integer
    'coverage threshold
Sheets("Nextgene").Select
'Start search in row 6
LSearchRow = 6
'Define coverage threshold.
CovThreshold = 500
'Define frequency thresholds for given percent tumor
Sheets("Demographics").Select
FreqThreshold = Range("B16").Value
'For all of the Nextgene data
Sheets("Nextgene").Select
While Len(Range("C" & CStr(LSearchRow)).Value) > 0
'If coverage is <500 and frequency is below threshold
If Range("I" & CStr(LSearchRow)).Value < CovThreshold And Range("S" & CStr(LSearchRow)).Value <
FreqThreshold Then
'Then change text in column A to "LOW FREQ & COV"
Range("A" & CStr(LSearchRow)).Select
ActiveCell.FormulaR1C1 = "LOW FREQ & COV"
```

```
'And make entire row pink
Rows(CStr(LSearchRow) & ":" & CStr(LSearchRow)).Select
Selection.Font.ColorIndex = 26
End If
'Check next row of data
LSearchRow = LSearchRow + 1
Wend
End Sub
```