2013

Discovering driver somatic mutations, copy number alterations and methylation changes using Markov Chain Monte Carlo

Bokhari Yahya
Virginia Commonwealth University

Discovering driver somatic mutations, copy number alterations and methylation changes using Markov Chain Monte Carlo

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Bioinformatics at Virginia Commonwealth University

By

Yahya Abdulfattah Bokhari
B.Sc. (Applied Medical Sciences), King Abdulaziz University, 2004
Saudi Arabia

Director: Tomasz Arodz, Ph.D.
Assistant Professor
Department of Computer science

Virginia Commonwealth University
Richmond, VA
December, 2013

Acknowledgement

I am truly thankful to God Almighty for all the blessings and successes in my life.

My deepest gratitude goes to my advisor, Dr. Tomasz Arodz, for his patience, unlimited support and his great help throughout this research project and thesis writing. It would not have been possible to finish the project without having him as an adviser. I also want to extend my thanks to the committee members, Dr. Vojislav Kecman and Dr. Danail Bonchev, for their interest in my project.

I want to express my immense appreciation to my wife Areej Alhareeri who supported me and encourage me from the first day I joined VCU to the last moment of finishing my master thesis. I would like to thank my mother who supported me every single moment with her prayers. I would like to thank Dr. Herschell Emery for believing in me and for the support he gave me as a Master student in Bioinformatics.

Table of Contents

List of tables

Abstract

DISCOVERING DRIVER SOMATIC MUTATIONS, COPY NUMBER ALTERATIONS AND METHYLATION CHANGES USING MARKOV CHAIN MONTE CARLO

Yahya Abdulfattah Bokhari, B.S.

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Bioinformatics at Virginia Commonwealth University

Virginia Commonwealth University, 2013

Director: Tomasz Arodz, Ph.D.
Assistant Professor
Department of Computer Science

Nowadays we have tremendous amount of genetic data needing to be interpreted. Somatic mutations, copy number variations and methylation are example of the genetics data we are dealing with. Discovering driver mutations from these combined data types is challenging. Mutations are unpredictable and have broad heterogeneity, which makes our goal hard to accomplish. Many methods have been proposed to solve the mystery of genetics of cancer. In this project we manipulate those above mentioned genetics data types and choose to

use and modified an existing method utilizing Markov Chain Monte Carlo (MCMC). The method introduced two properties, coverage and exclusivity. We obtained the data from The Cancer Genome Atlas (TCGA). We used MCMC method with three cancer types: Glioblastoma Multiform (GBM) with 214 patients, Breast Invasive Carcinoma (BRCA) with 474 patients and Colon Adenocarcinoma (COAD) with 233 patients.

Chapter 1

Introduction

In this work we aim at computational discovery of driver mutations that play a major rule in cancer progression and distinguish them from passenger mutations that are present in cancer cells but do not contribute to the disease. We start with introducing relevant biological background, and then move to computational methods for processing cancer mutation data and algorithms for discovering driver mutations.

The Genetic Roots of Cancer

Cancer is a term given to any disease in which abnormal cells divide indefinitely and have the ability to invade other tissues. The American Cancer Society estimates the diagnosis of 1,660,290 new cases of cancer and that the later will account for 580,350 deaths in 2013 (American Cancer Society, 2013). Cancer is a complex and heterogeneous disease to which all body organs and tissues are susceptible. It is believed to be a genetic disease of cells. Carcinogenesis typically involves the following chain of mutations that deregulates cellular proliferation; m1: inactivation of a tumor suppressor gene results in cell proliferation, m2: mutation(s) that inactivates a DNA repair pathway, m3: a mutation in a proto-oncogene that leads to the generation of an oncogene and m4: mutation(s) that inactivates additional tumor suppressor genes resulting in cancerous proliferation (reviewed in Cornelisse & Devilee). Normal genes in the cell that control cellular proliferation are called proto-

oncogenes, which can be mutated to form oncogenes that promote uncontrolled cellular proliferation. Such mutations are considered dominant or gain of function mutations. Therefore, one mutated copy of the gene is enough to promote cancer. In contrast, tumor suppressor genes are those encoding proteins that normally inhibit tumor formation inhibiting cellular proliferation. These mutations are recessive or loss of function mutations where loss of both copies of a gene is required to inactivate a tumor suppressor gene (Hanahan & Weinberg, 2000). Ongoing research over the past years have revealed that cancer occurs as a consequence of a hereditary mutation or an environmentally induced mutation with or without genetic-environmental interaction (Knudson, 2002). A change in the genome of a particular cell in the form of a mutation could be of various types. These include: point mutations which causes amino acid substitutions that either truncate a protein product or scramble it's sequence; chromosomal imbalances or instability resulting in amplification, overexpression or loss of a gene; epigenetic modifications of the DNA of which is the most important is DNA methylation (Bertram, 2000).

The Role of Somatic Mutations, Copy Number Alterations and DNA Methylation in Cancer

The cancer cell is like any other cell that constitutes the human body in being a direct descendent through mitotic cell division of the fertilized egg from which the human being has developed. However, a cancer cell as well as most normal cells acquire a number of alterations in the DNA sequence from it's progenitor fertilized egg. These alterations are collectively named somatic

mutations to distinguish them from germline mutations that are passed from parents to offspring (Stratton et al., 2009). It is thought that somatic mutations occur in the genomes of normal cells through successive round of cell division during development in utero and during regeneration of body tissues in postnatal life. Moreover, the DNA in normal cells is susceptible to continuous damage by endogenous and exogenous mutagens. Most of this damage is repaired except for a small fraction that may be converted into fixed mutations, which may confer a selective growth advantage leading to clonal proliferation of these cancerous cells (Stratton, 2011). There are several distinct classes of somatic mutations in cancer cell genomes including substitutions of one base by another; insertion or deletion of small or large DNA segments; rearrangements in which a segment of DNA is broken and relocated to elsewhere in the genome; changes in the copy number of DNA segments; epigenetic alterations that are stably inherited through mitotic DNA replication. Somatic copy number alterations (SCNAs, distinguished from germline copy number variations, CNVs) are structurally variant regions with either gains or losses of genomic DNA. They play a major role in cancer development through the amplification of oncogenes or the deletion of tumor suppressor genes.

Epigenetic marks are defined as modifications of the DNA and associated proteins that alter gene expression independent of alterations in the DNA sequence. The four main epigenetic modifications are DNA methylation, histone modification, chromatin remodeling and RNA-mediated targeting. In addition, epigenetic regulation is an essential phenomenon for proper development and

cellular differentiation in normal human tissues. The well known and best studied epigenetic modification is DNA methylation at the carbon 5 of cytosine residues that precede guanines, refereed to as CpG dinucleotides, by DNA methyltransferases (DNMTs). 70-80% of cytosines in the genome of normal cells are methylated. Furthermore, areas of the genome with high concentration of CpGs are called CpG islands and are located in the promoter region of 50% of human genes, thus these areas are mostly unmethylated (Huidobro et al., 2013). Earlier studies on gene expression and DNA methylation have shown an interaction between cancer and epigenetics. Promoter hypermethylation leading to transcriptional silencing and global hypomethylation are of the most characterized epigenetic changes in human cancers (Dawson & Kouzarides, 2012). Therefore, we now know that growth-promoting genes are activated through hypomethylation in tumors. Moreover, tumor suppressor genes silencing have been found to be linked to promoter hypermethylation (reviewed in Iacobuzio-Donahue, 2009). Given the plasticity and heritability nature of epigenetic modifications, DNA methylation is ideally suited to the processes of clonal variation and clonal inheritance, which are required for the transformation of a normal cell into a malignant cell.

“Driver” vs “Passenger” Genes

Cancer genomes carry two biological classes of somatic mutations, which are driver and passenger mutations depending on the corresponding nature of this mutation to cancer development (Stratton et al., 2009). Driver mutations provide the neoplastic clone with growth advantage. Therefore, they allow this

neoplastic clone to proliferate more than normal cells from the same tissue, invade the surrounding tissues and in most cases permit metastasis. These mutations reside by definition in a subset of genes known as “cancer genes”. Given that, the number of mutated cancer genes is reflected by the number of driver mutations in a cancer cell, thus the required dysregulated cellular biological processes to convert a normal cell into a cancer clone. On the other hand, passenger mutations, which in most cases constitute the majority of mutations, are those that do not confer a growth advantage. Instead, it is thought that these mutations were present in the ancestor of the cancer cell when it acquired any of it’s driver mutations (Stratton, 2011). As a result, these passenger mutations arise from mutational exposures, genome instability or from the increased cell division and doubling that give rise to a clinically detectable cancer from a single transformed cell (Haber & Settleman, 2007). Having both driver and passenger mutations in the cancer genome makes it challenging to distinguish them from each other, hence discovering the genes that play an essential role in tumorigenesis.

Analysis of cancer samples using high-throughput methods

During the past decades, there have been major advances in characterizing cancer genomes through first generation sequencing (also known as Sanger sequencing). Alternatively, next generation sequencing (NGS) has been developed over the past 7 years with higher throughput and increased sensitivity derived from it’s deep coverage. Moreover, the application of NGS has tremendously decreased the time and cost required for data generation (

reviewed in Dong & Wang, 2012). However, the availability of large data sets from these techniques implicates a number of challenges. One challenge is the ability to distinguish “driver mutations” that are important for cancer development from “passenger mutation” that have accumulated in somatic cells but are not of an importance in cancer development. One standard approach for identifying “driver mutations” is to test for genes that are recurrently mutated in a large number of cancer genomes. This approach has been useful in identifying only a subset of “driver mutations” due to the extensive mutational heterogeneity not only among different cancer types but also among individuals with the same tumor type (Vandin, Upfal & Raphael, 2012). This brings is to another challenge that arises from the generation of tremendous amount of data, which is the need of computational and algorithmic tools for the analysis of the growing data sets from NGS of cancer genomes.

The data generated from these cancer genome characterization efforts have put the need for data accessibility and in depth comparative analyses of different cancer types. The Cancer Genome Atlas (TCGA) is a perfect resource for this purpose, providing complete catalogs of the genomic alterations in a collection of patient samples that have been characterized in a cancer-type specific manner (Chin et al., 2011).

Existing Computational Methods for the Identification of Driver Genes

There are several methods to discover driver mutations. Below, we outline two existing methods in order to give a background on some existing approaches for discovering driver mutations. The first method was proposed in (Ciriello, Cerami, & Sander, 2012) and the second method was initially introduced by (Vandin, Upfal, & Raphael, 2011). The last method is the method we used in this work and it is introduced by (Vandin, Upfal, & Raphael, 2012).

I. Mutual exclusivity analysis to identify oncogenic network modules

The algorithm consists of five steps:

1. Building a binary matrix of genes that has been evaluated as significant or not significant. The gene is considered significant if it is recurrently mutated. Moreover, genes that recurrently experience a high level of amplification or homozygote deletion proportional to their expression are considered significant as well.
2. Categorizing filtered genes and pairing them while using previous knowledge of biological networks and pathways.
3. Building a graph and connecting similar gene pairs.
4. Extracting local fully connected sub graph clusters, which are most likely to have similar biological characteristics.
5. Putting the extracted clusters through further filtration to confirm mutual exclusivity as well as excluding the possibility that these gene clusters have been constructed by chance. The Permutation Test is a test that permutes the genes, appeared in the original binary matrix, across the samples several times.

During permutation, edges are added and deleted, resulting in random modules. After calculating the p-value of about Q |whole set of edges| random generated modules, finding a low P-value makes the original cluster module unlikely to have been generated by chance.

The filtered clusters are considered potential driver networks that initiate cancer (Ciriello, Cerami, & Sander, 2012).

II. Heat diffusion

The aim of heat or fluid diffusion (Vandin, Upfal, & Raphael, 2011) is to extract sub-networks that contain highly mutated genes. Observing two highly mutated genes connected by a single low-degree node is of great interest. On the other hand, it is less interesting to have a single high-degree node connecting several highly mutated genes. The steps constituting the algorithm used to extract sub-networks are as follow:

1. Diffuse heat to each mutated gene proportional to the frequency of mutation of a given gene. The heat then conducts through the edges for a certain period of time.
2. Assign an influence measure between graphed gene pairs according to the heat distribution. A low-degree node will have small neighbors to diffuse the heat to; accordingly the nodes will remain hot. Alternately, high-degree nodes will have neighbors of any size to diffuse the heat to, and thus will not be able to keep their heat.
3. Break the network into sub-networks according to the heat distribution and

the score of the node (gene). A highly mutated gene that is present in a high number of patients has a higher score than a gene that is not frequently mutated.

4. Evaluate these sub-networks statistically and assess the possibility of having similar sub-network by chance.

III. De novo discovery of mutated driver pathways in cancer

The method proposed by (Vandin, Upfal, & Raphael, 2012) is based on two assumptions. The first assumption is that a statistically significant cancer pathway is likely to be perturbed to cause cancer, i.e., utilizing a genome-wide approach for screening a group of patients with the same cancer type will reveal the perturbation of a certain pathway in these patients. The second assumption is that one driver gene mutation in an important cancer pathway is enough to perturb the pathway (McCormick, 1999) (Vogelstein & Kinzler, 2004). Furthermore, given the rare and single pattern appearance of a driver mutation, we expect a mutually exclusive pattern of driver mutations (Chen-Hsiang, 2008). Markov chain Monte Carlo (MCMC) is the algorithm used to identify driver mutations using the above-mentioned assumptions in (Vandin, Upfal, & Raphael, 2012). The purpose of using (MCMC) is to find a set of mutated genes that both cover most of the patients (high coverage) and most of the covered patients have one gene mutated from that set (high exclusivity). The data used in MCMC was somatic mutation and copy number variations (CNVs) from TCGA. Details about using MCMC will be further mentioned in the materials and methods section.

Goal of the thesis

Our goal in this thesis is to discover driver mutations in cancer. We relied on the existing method described in (Vandin, Upfal, & Raphael, 2012). The main task was to apply the method to a broader set of data containing additional types of mutations. We included copy number alteration and methylation as additional types of mutations.

We used MCMC with small modifications in the parameters and data. The first modification is based on our insufficient knowledge of the number of cancer-specific driver genes. We extended the method to include variable number of genes. Thus, we used MCMC to look for a set of driver genes between three and seven. The second modification is based on our desire to include copy number alterations (CNAs) and methylation in the data. Since somatic mutations are not the only cause of cancer, its use as a single type of mutations could mask other underlying causes of cancer. Given that CNAs and methylation contribute greatly to the onset of cancer, including them in the data will help in completing the underlying picture of cancer.

It is challenging to use somatic mutations, CNA and methylation as an input data. One of the reasons is that somatic mutations have variability in the mutated position of each gene mutated. Therefore, a much bigger data size is needed because of the possibility of two patients with the same mutated gene would have different mutation position. Another potential difficulty is the fact that

methylation is in the nucleotide level, which could complicate its comparison to somatic mutations even when disregarding the mutation position. Moreover, more than one gene could be included in the case of CNAs.

To use somatic mutations, CNAs and methylation at once, we focused on the genetic level by following certain rules:

- We disregard the nucleotide position variations and only consider the gene name in somatic mutations.
- In methylation, if a given position is abnormally methylated we consider that gene to be mutated.
- In CNAs, any gene included in significant CNAs is considered mutated.

Materials and Methods

1. Sources

1.1 Somatic mutation data

We downloaded all available data in TCGA level 2 for Glioblastoma Multiforme (GBM), Breast Invasive Carcinoma (BRCA) and Colon Adenocarcinoma (COAD). TCGA does not have level 3 data available for somatic mutations. Therefore, although that would be the higher TCGA level having the most possible verified data, we had to use level 2. We gathered this TCGA data from the following sources: for GBM, the Broad Institute (http://broad.mit.edu/); for Breast Invasive Carcinoma, the Genome Institute (http://genome.wustl.edu/); and for Colon Adenocarcinoma, the Human Genome Sequencing Center or HGSC (https://www.hgsc.bcm.edu/).

1.2 Copy number variation (CNV) data

We downloaded all available data in level 3 for Glioblastoma Multiform (GBM), Breast Invasive Carcinoma (BRCA) and Colon Adenocarcinoma (COAD). TCGA data on GBM was provided by the Memorial Sloan–Kettering Cancer Center (MSKCC), Harvard-Partners Center for Genetics and Genomics (Harvard Medical School), Broad Institute and Stanford University HudsonAlpha Institute for Biotechnology. TCGA copy number variation (CNV) data on BRCA and COAD was provided by the Broad Institute.

1.3 Methylation data

We downloaded all available data in level 3 for Glioblastoma Multiform (GBM), Breast Invasive Carcinoma (BRCA) and Colon Adenocarcinoma (COAD). TCGA data on GBM, BRCA and COAD was provided by USC Epigenome Center, University of Southern California.

2. Filtering the data

Our target was to extract the genes' "TCGA-barcode," "Hugo-Symbol" and "Entrez-Gene_Id" (if available), and to collect these data into a file organized by columns in that order (Where the data is not available, we had to infer or deduce the required information). To reach that objective we went through several filtering processes, which are different for each data type.

2.1 Filtering somatic mutation data

We downloaded mutation data as a Mutation Annotation Format (MAF), which has a lot of columns describing the samples. The only columns we utilized from this format were "Hugo_Symbol," "Entrez_Gene_Id" and "Tumor_Sample_Barcode." Using Unix "cut" command, I extracted these needed columns. Then we used AWK, a Unix based interpreter language, to reorder the columns to the required format for our data file. The TCGA barcode consists of the flowing information: "TCGA" as the initial tag followed by the tissue source type code, the participant code, sample code, portion code, plate code and finally the center code. We only used the portion consisting of the "TCGA" tag, the tissue source code type and the participant code; instead of writing a program I used regular expression to cut unwanted parts of the full TCGA barcode. In the

extracted table we found that some Entrez Gene Ids had a zero value, which does not match the Hugo Symbol and means there is lack of information for that Id. To deal with this situation, we developed "GeneSymbol_to_Entrez_converter.py," a converter that uses a reference map to go through each line of our data file and check that each Hugo Symbol has a matching Entrez Gene Id. We downloaded the reference table that maps "Hugo-Symbol" to "Entrez-gene-Id" from the HUGO Gene Nomenclature Committee (HGNC) to use in the "GeneSymbol_to_Entrez_converter.py". Once each column in the file had the required information of "TCGA-barcode," "Hugo-Symbol" and "Entrez-Gene_Id", the somatic mutation data was ready to be merged with the other data with the same format.

2.2 Filtering CNV data

The downloaded package for CNV includes the following files and folders:

- CNV_SNP_Array folder.
- FILE_SAMPLE_MAP.txt file.
- METADATA folder.

The "CNV_SNP_Array" folder has the data in multiple files with many lines to each file. Data files have the following header:

- *Sample:* sample ID.
- *Chromosome:* chromosome number.
- *Start:* where the segment begins.
- *End:* where the segment ends.

- *Num_Probes:* Probes used in an each segment.
- *Segment Mean:* if the segment mean is around zero, there is no loss or gain. If the segment is above zero, we can recognize a gain, and if the segment is less than zero, that indicates a loss.

We needed to modify the data in order to have it ready to use we needed to do the following:

2.2.1 Transforming the data to be binary

We wanted to use segment mean to decide if a given gene was included in the CNV. The obstacle is that the segment mean is not a 0 or 1 value. To get 0 or 1 value, we calculated the mean and the standard deviation of each probe segment-mean occurring in all samples and extracted the abnormal segment-means. We did not use the typical measurement of the segment mean mentioned above. Instead, we wanted to have a new cutoff value to decide if the segment had been repeated or lost. We decided to use the mean and the standard deviation as a cutoff value, so we first calculated the mean and the standard deviation of each probe segment-mean which occurred in all samples using,"CNV_FILES_Mean_SD_Calculator.py", a program we wrote. We wanted to know how many patients were above 1, 2, 3 and 4 standard deviations in order to decide what the cutoff should be. "Calculate_How_many_Patients_above_1-2-3-4_SD.py" calculates the number of samples above and under 'X' standard deviation. We choose 3SDs to be the cutoff as there were too many samples below 2SDs and few above 4SDs. We considered a specific probe segment-mean a gain in copy number if it was above 3 standard deviations of the mean of

all segment-means of that specific probe in all samples. Likewise, we considered a specific probe segment-mean a loss in copy number if it was below 3 standard deviation of the mean of all segment-means of that specific probe in all samples. To make these calculations and extract the desired data, we used "Abnormal_CNV_Extractor.py,", which only extracts from each file the abnormal lines and copies them in a different directory.

2.2.2 Converting gene beginnings and gene ends into gene names

The following rules were applied to decide whether the gene was inside or outside the segment:

a) If the whole gene was inside the segment then we considered the copy number of that gene as a variant.
b) If the whole segment was inside a gene we also considered the copy number of that gene as a variant.
c) If 50% of the gene was inside the segment we considered the copy number of that gene as a variant.

To achieve the conversion goal, we wrote "GenePostion_to_GeneName_Converter.py". This program goes through each line on each file and, depending on the beginning and the end of each line, converts it to Ensembl gene ID. This program needed a reference file containing the beginning and the end of each gene, which can be gotten from the Ensembl Genome Browser.

2.2.3 Converting the Ensembl to Entrez and HUGO gene symbol

Using the data table downloaded from the HUGO Gene Nomenclature Committee (HGNC) that contains Ensembl, Entrez and HUGO gene symbols, we wrote the "Ensembl_to_Entrez_GeneSymbol_converter.py" program to convert the Ensembl gene ID into desirable names.

2.2.4 Mapping the sample ID to TCGA sample code

The downloaded data has an ID sample different from global TCGA barcode ID, which we want to unify. The downloaded folder has “FILE_SAMPLE_MAP.txt”, which maps the sample ID to the TCGA barcode ID. TCGA_Mapper.py is a program that maps the IDs and prints all files into a new folder with the required TCGA barcode.

2.2.5 Organizing the desired file structure

After we got the required sample code, we wanted each column in the file to be formatted in the following order: TCGA barcode, Entrez ID and gene ID symbol. Instead of writing a new program code to organize the data this way, I wrote this code as a part of one of the previous programs. The final output is one file that has the above-mentioned columns in order, ready to be merged with other filtered data.

2.3 Filtering methylation data

The downloaded package for methylation data includes the following files and folders:

- DNA_Methylation folder.

- FILE_SAMPLE_MAP.txt file.
- METADATA folder.

The “DNA_Methylation” folder has the data in multiple files and each file has many lines. Data files have the following header:

TCGA Barcode probe name beta value gene symbol chromosome position.

- *TCGA barcode:* sample ID.
- Probe name*:* Probes used at each position.
- Beta value: the intensity ratio.
- Gene symbol: HUGO symbol.
- Chromosome: chromosome number.
- Position: at what position in the chromosome.

We needed to modify the data in order to have it ready to use we needed to do the following:

2.3.1 Compare the methylation data to a reference

We needed to have a normal reference of beta value for each probe. We do not know if a position is methylated or unmethylated, but we know that when the beta value is close to zero it’s more likely to be unmethylated; also, the closer beta value is to one, the more likely it is to be methylated. The Gene Expression Omnibus (GEO) has data for a Genome-Wide Methylation Analysis containing data of normal patients for brain, breast and colon cancers among the data for abnormal patients. We used many programs to filter the data downloaded from

GEO. Using Unix commands, we cut the column that has data known to be normal patients' samples for brain, breast or colon cancer. The methylated data in GEO per cancer type is scattered in different project files and we need them to be in one file to simplify the analysis and infer what normal status (methylated or unmethylated) of each probe position should be. "Methylation_files_joiner.py" is a program which joins the normal methylation columns from different GEO files into one file containing probe names and columns with beta values of normal patients with same tissue type. We have 5 normal methylated brain tissue data, 37 normal methylated breast tissue data and 33 normal methylation colon tissue data. Each line in those data starts with the probe name and is followed by the beta value of each patient for that probe. We saw earlier the somatic mutation data has the following columns information; "TCGA-barcode," "Hugo-Symbol" and "Entrez-Gene_Id" and we need the same information with methylation data. GEO has data table that maps each probe to the gene Entrez ID and the gene symbol ID and its synonyms. "Gene_Id_Synonyms_appender.py" is a program which uses GEO mapping table to map each probe name to gene Entrez ID and gene symbol ID, and adds the new two columns of the gene Entrez ID and gene symbol ID into the new file that has the data for normal beta values.

2.3.2 Extracting the common probes between methylation data folders

Unify the probes in the raw data files. "DNA_Methylation" downloaded from TCGA has the raw methylated data in two files. The first file starts with "hu-usc.edu__HumanMethylation450" and the second starts with "hu-usc.edu__HumanMethylation27". The HumanMethylation27 file has fewer probes

than HumanMethylation450 file. We wanted each probe in the sample file to have the normal reference beta value we created. All probes in HumanMethylation27 file have a normal reference beta value, while HumanMethylation450 does not because the GEO that we extracted normal beta value from used the same probes and illumine platform as HumanMethylation27. We solved this in 4 steps

a) Finding the common probes between HumanMethylation27 and the normal reference beta value file.
   "Common_Uncommon_Probes_Finder.py" is a program, which reads two files and outputs the uncommon and the common probes. We ran the two files and found out that all HumanMethylation27 file probes had a normal beta value.
b) Finding the common probes between HumanMethylation450 file and the normal reference beta value file. We ran the two files and found out that there were 457,999 probes out of 485,578 probes within HumanMethylation450, which were not in the normal reference beta value file, so we need to get rid of the lines of extra probes.
c) Finding the common probes between HumanMethylation27 and HumanMethylation450. "Common_Uncommon_Probes_Finder.py" read one file from HumanMethylation27 folder, one file from the normal reference beta value file and one from HumanMethylation450 folder, and then output the extra probes in HumanMethylation27 and the common ones in both files. We found all 27,579 probes in HumanMethylation27 in HumanMethylation450.

d) Deleting uncommon probes from all files existing in the HumanMethylation450 folder. We did not use any probe that existed in the HumanMethylation27 files but not in HumanMethylation450. So we wrote a program called “Unmatched_Probes_Deleter.py” to go through each HumanMethylation450 file and delete all 457,999 uncommon probes. By doing these steps, we have the same number of probes in the HumanMethylation27 and HumanMethylation450 folder files and a reference of normal beta values for each probe.

2.3.3 Extracting abnormally methylated/unmethylated data

Having our probes in normal beta value matched to the probes in the TCGA data, we wanted to go through each probe of each patient and extract the abnormal methylated data. The purpose of this step was to have a reference data to decide whether a given position in a specific gene should be methylated or not. “MethStat.py” is a program for calculating the statistical summary for each probe. Calculated data includes the number of patients, the mean and standard deviation, the minimum, maximum and the skew. We need to have this information to establish the cutoff for methylated/unmethylated for each probe. Our first approach to analyzing the data was to use the mean and 2 SD; if the probe reading was 2 SD away from a normal reference probe, we considered the gene contained in that position methylated and it would be included in the abnormal methylated gene file. This approach did not work because it showed that most probes in all patients were methylated, which is unlikely to be true. We made another attempt using the minimum/maximum of the normal reference

instead. If the minimum normal beta value is above 0.75 and the patient beta value is below 0.25, then this patient's gene is abnormally unmethylated. If the maximum normal beta value is below 0.25 and the patient beta value is above 0.75, then this patient's gene is abnormally methylated.

"Meth_Analyzer_MinMax.py" uses the above-mentioned algorithm to extract the abnormal lines. The final output is one file that has the following columns "TCGA-barcode," "Hugo-Symbol" and "Entrez-Gene_Id" that are ready to be merged with other filtered data.

3. Merging the filtered data (Somatic Mutation, CNV, Methylation)

We wanted to have one file of unified (filtered) genetic information about Somatic Mutation, CNV and Methylation containing the data from patients that fulfill all three categories. We didn't want patients with missing information; in other words, in order for a patient to exist in our final filtered data, he/she must have a TCGA barcode in each of the Somatic Mutation, CNV and the Methylation filtered files. "Common_Patients_extractor.py" is a program that goes to each data file using the format "same column headers" and finds the common barcodes between the three filtered data. After that it goes through each file again and prints the lines with the common barcodes into one file. We have done this to the three cancer types: Glioblastoma Multiform (GBM) with 214 barcodes (patients), Breast Invasive Carcinoma (BRCA) with 474 barcodes (patients) and Colon Adenocarcinoma (COAD) with 233 barcodes (patients). Having done that, we were ready to use MCMC algorithm to extract driver genes for each cancer type.

4. Algorithm for discovering driver mutations

4.1 Problem Description

To have a better understanding of the problem described in (Vandin et al., 2012), let $P = \{p | p \in cancer\ patients\}$ and $G = \{g | \exists\ p \in P : gene\ g\ is\ mutated\ in\ p\}$. Relation R from G to P is defined as follow: For all $(g, p) \in G \times P$, $(g, p) \in R$ means g is either mutated, methylated or having a copy number alteration, and we use $g\ R\ p$ as a notation for the relation. Let $\Gamma(g) = \{p | p \in P \cap g\ R\ p\}$ denote the set of patients having gene g mutated. Let M= {M ⊆ G} and Γ(M) denotes the group of patients so that each has at least one gene in set M mutated, methylated or having a copy number alteration. Set M is considered mutually exclusive if each patient within this set has only one gene g mutated. For example let g_x and g_y be any two genes ∈ M, in this case set M is considered mutually exclusive if $\Gamma(g_x) \cap \Gamma(g_y) = \emptyset$. We wanted to have a mutually exclusive set M that also covers most of the patients. In other words they wanted to maximize Γ(M) such that M is mutually exclusive. However, this is computationally difficult. Furthermore, the rule of mutually exclusive is impractical to follow due to the presence of noise (passenger mutations) and measurement errors.

4.2 Coverage and exclusivity maximization

In regard to solving the above-mentioned problem, (Vandin et al., 2012) proposed to modify the mutual exclusivity restriction for set M and consider $|\Gamma(M)|$ as a criterion. With the application of this modification, mutually exclusive

is now considered approximately exclusive in which most of the patients have no more than one gene mutated in M whereas the coverage is not changed in which the majority of patients have one or more genes mutated in M. Implementing this relaxation rule results in a trade off between exclusivity and coverage. Moreover it leads to a certain number of patients with coverage overlap, in which patients can have more than one gene in the set M mutated. The mathematical expression of the coverage overlap is:

$$\omega(M) = \sum_{g \in M} |\Gamma(\mathrm{g})| - |\Gamma(\mathrm{M})|.$$

Coverage overlap equation in words mean the summation of all set of patients that have gene g from set M mutated subtracted from the all patients with at least one gene in set M mutated. For Example if our driver genes set is x y and z and we have a patient with those 3 genes mutated then $\omega(M) = 3 - 1$, which is the excess mutation above the required number of mutation. In other words $\omega(M)$ is number of mutations above one per patient.

Coverage overlap is addressed by subtracting it from the coverage $|\Gamma(\mathrm{M})|$ and have the weight $W(M)$:

$$W(M) = |\Gamma(\mathrm{M})| - \omega(M).$$

$$W(M) = 2\,|\Gamma(\mathrm{M})| - \sum_{g \in M} |\Gamma(\mathrm{g})|.$$

The penalty ω(M) = 0 when M is mutually exclusive. Now that we have a defined weight, the problem is to maximize the weight $W(M)$ by finding the appropriate gene set M.

4.3 The Metropolis-Hastings algorithm

Metropolis-Hastings algorithm (Metropolis, 1953) (Hastings, 1970) is a well-known established algorithm. It has been used in different applications as well as to solve the above-described problem. We used Metropolis-Hastings algorithm with k = number of the genes in the set and n = number of iteration.

Pseudo code	Description
Initialization:	
M_1 = RANDOMSET(k,G)	From the whole gene set G choose k genes randomly to be the starting gene set M_1.
Iteration:	
For i = 1 to n	
r = RANDOMGENE(G).	Randomly choose gene r from the whole gene set G.
s = RANDOMGENE(M_i).	Randomly choose gene s from the current gene set M_i.
$M_{-s+r} = M_i - s + r$.	Replace gene s in set M_i with gene r as a new candidate set.
$W_i = cW(M_{-s+r}) - cW(M_i)$.	Calculate W_i by replacing gene s in set M_i with gene r.
p = min$[1, e^{w_i}]$.	p is Min between 1 and the weight difference W_i.
$M_{i+1} = M_{-s+r}\ with\ probability(p)$.	The more p the more likely $M_{i+1}$to be equal M_{-s+r}.
Otherwise $M_{i+1} = M_i$.	Go to next iteration with the same set M_i if M_{-s+r} fail not replace M_i.

4.4 Size changing algorithm

We improved the algorithm by incorporating variability in set sizes. In each iteration there is a chance that instead of going through the rest of the algorithm, we change the number of the genes in the set M_i by removing or adding genes from gene set G. The variability of the set size accomplished is as specified below.

Suppose k_{max} is the maximum number of gene k inside set M and k_{min} is the minimum number of gene k at any given set M_i.

Pseudo code	Description
Initialization:	
M_1 = RANDOMSET(k,G)	From the whole gene set G choose k genes randomly to be the starting gene set M_1.
Iteration:	
For i = 1 to n	
Boolean=FlipTheCoin()	Return binary value.
if (Boolean==0)	
Continue with Metropolis-Hastings algorithm.	
else:	
k = RANDOMNUMBER($k_{max,}$ k_{min})	Random a number between k_{max}, k_{min} inclusively.
$if\ (k > \|M_i\|)$	If k is greater than the current set elements.
$while\ (k > \|M_i\|)$	
$r = \mathrm{RANDOMGENE(G)}$	Randomly choose gene r from the whole gene set G.
$M_k = M_i + r$	$M_k = M_i$ plus gene r
else	
$while\ (k < \|M_i\|)$	Else if k is less than the current set elements.
$r = \mathrm{RANDOMGENE}(M_i)$	Randomly choose gene r from the whole gene set G.
$M_k = M_i - r$	$M_k = M_i$ minus gene r
$W_i = cW(M_k) - cW(M_i).$	Calculate W_i by replacing gene s in set M_i with gene r.
p = min$[1, e^{w_i}]$.	p is Min between 1 and the weight difference W_i.
$M_{i+1} = M_k\ with\ probability(p).$	The more p the more likely M_{i+1} to be equal M_k.
Otherwise $M_{i+1} = M_i$.	Go to next iteration with the same set M_i if M_k fail not replace M_i.

We run MCMC three times for each disease. Each time the run ends we delete all lines that included discovered genes. We wanted to know after that if the genes we discovered are drivers. We matched each gene we discovered to the following dataset.

5. Validation of results

A significant problem in discovering driver mutations is how to verify which are the cancer driver genes, since if we already had a sound method of positively discovering this; our current research would be unnecessary. There is no solution to this widely known problem, although we do have some methods for narrowing the field of candidate drivers down to dataset that most likely to be drivers, polymorphism or otherwise unknown.

5.1 Catalog of Somatic Mutations in Cancer (COSMIC) and whole genome (WG)

Gonzalez-Perez introduced a method of transformed Functional Impact Score, which assesses non-synonymous single nucleotide tolerance of variations. The mutations causing the highest functional impact have a high probability of being actual cancer drivers. They downloaded all somatic nSNVs from COSMIC. Also, they gathered the whole genome (WG) dataset by pooling somatic mutations from several sources including the International Cancer Genome Consortium (ICGC) Data Coordination Center. The algorithm they used results in several subset dataset (Gonzalez-Perez , 2012). We only considered three of them. The first dataset is cosmic5.dataset, which is list of somatic

mutations that appear in more than 4 Cosmic samples. The second dataset is wgCGC.dataset, which is a list of somatic mutations detected by "whole cancer genome" projects that occur in known cancer genes. We pooled cosmic5.dataset with wgCGC.dataset to get one dataset of known driver somatic mutation genes. We tested each list of genes we got from MCMC algorithm against this dataset to check if they are drivers. The third file we used is Pol.dataset, list of known polymorphisms extracted from the HumVar (a dataset of disease-related SNVs and neutral polymorphisms) dataset. We used this dataset to test if the list of genes we got are known to be polymorphism.

5.2 Kyoto Encyclopedia of Genes and Genomes (KEGG)

KEGG is a database resource for understanding high-level functions and utilities of the biological system, from molecular-level information technologies generated by genome sequencing and other high-throughput experimental (KEGG, 2013). We extract a list of known cancer genes from KEGG web database to be another source of comparing the list of genes we got.

5.3 Foundation Medicine (FM)

We also tested the gene list against a list from Foundation Medicine that was gathered by testing 304 patients for solid tumor analyzed by NGS assay (Palmer, 2013).

5.4 Literature Review

We used Google Scholar and PubMed to check on genes that does not exist on either of the previous datasets. We wanted to confirm that a given gene is not a polymorphism.

Chapter 3

Results and Discussion

In order to find maximum weight of mutually exclusive set of genes we ran our program using MCMC algorithm against three cancer types; Glioblastoma Multiforme (GBM), Breast Invasive Carcinoma (BRCA) and Colon Adenocarcinoma (COAD). The program was ran three times, following each run we had candidate driver genes, therefore we went back to the original dataset and delete the genes and rerun the program. The results are described in tables where each disease consists of two tables. The first one is the results table from the program and the second one is the validation of the results table. In the result tables, each row represents a run. The parameter we pass into the program was to iterate 100,000 times and to explore set cardinality between 3 and 7 elements to improve the weight. In validation of the results tables we gave a value of 1 if a given gene exists in that dataset or zero otherwise. The Literature review column is a quick review to check if the gene does not exist in any datasets. We wanted to confirm that a given gene mutation is not a known polymorphism and that it has the potential to be a driver mutation. The last column, which we called “unknown” contains a value of 1 if the gene is ambiguous or 0 otherwise.

1. Glioblastoma Multiforme (GBM)

1.1 GBM Results Table (Table 1)

Weight	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6	Gene7
137	FSD1	TP53	PIGB	RNF19A	CDADC1	COPB2	
157	HOXD4	PCLO	HNRNPA1P7	ENPEP	MTMR3	CPNE8	UXS1
152	ONECUT2	VEPH1	TCHH	PLEKHH1	STAG3L3	PPP1R16B	

1.2 GBM Validation Table

Gene	COSMIC Driver	COSMIC Polymorphism	KEGG	Foundation Medicine	Literature Reviews	Unknown
FSD1	0	0	0	0	(Amandine, 2010)	1
TP53	1	1	1	1	NA	0
PIGB	0	0	0	0	0	1
RNF19A	0	0	0	0	0	1
CDADC1	0	0	0	0	0	1
COPB2	0	0	0	0	0	1
HOXD4	0	0	0	0	0	1
PCLO	0	0	0	0	0	1
ENPEP	0	0	0	0	0	1
MTMR3	0	0	0	0	(Song, 2010)	1
PNE8	0	0	0	0	0	1
UXS1	0	0	0	0	0	1
ONECUT 2	0	0	0	0	0	1
VEPH1	0	0	0	0	0	1
TCHH	0	0	0	0	0	1
PLEKHH1	0	0	0	0	0	1
PPP1R16 B	0	0	0	0	0	1

2. Breast Invasive Carcinoma (BRCA)

2.1 BRCA Results Table

Weight	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6	Gene7
263	CDH12	TP53	CDH1	NR2F1	ARHGEF3-AS1		
231	NCAM2	SOLH	PHF3	EPC1	TRIM58	TNFRSF11B	
243	PIK3CA	ROBO2	DSCR4	CD19	HMCN1	FCN2	

2.2 BRCA Validation Table

Gene	COSMIC Driver	COSMIC Polymorphism	KEGG	Foundation Medicine	Literature Reviews	Unknown
CDH12	0	1	0	0	(Wang J. , 2011)	1
TP53	1	1	1	1	NA	0
CDH1	1	1	0	1	NA	0
NR2F1	0	0	0	0	(Smits , 2013)	1
NCAM2	0	0	0	0	(Takahashi, 2011)	1
SOLH	0	0	0	0	0	1
PHF3	0	1	0	0	0	0
EPC1	0	0	0	0	(Matthias , 2010)	1
TRIM58	0	0	0	0	0	1
TNFRSF11B	0	1	0	0	NA	0
PIK3CA	1	0	0	1	NA	0
ROBO2	0	0	0	0	0	1
DSCR4	0	0	0	0	0	1
CD19	0	0	0	0	0	1
HMCN1	0	1	0	0	0	0
FCN2	0	0	0	0	0	1

3. Colon Adenocarcinoma (COAD)

3.1 COAD Results Table

Weight	Gene1	Gene2	Gene3	Gene4	Gene5	Gene6	Gene7
159	TP53	JAKMIP1	ARMCX6	DPP9	DCLK3	ZNF532	C6orf225
179	APC	SCNN1A	GLP2R	SDF2L1			
135	SHOX	NCAM2	BHMT2	STAM2	CYSTM1		

3.2 COAD Validation Table

Gene	COSMIC Driver	COSMIC Polymorphism	KEGG	Foundation Medicine	Literature Reviews	Unknown
TP53	1	1	1	1	NA	0
JAKMIP1	0	0	0	0	(Okai, 2013)	1
ARMCX6	0	0	0	0	0	1
DPP9	0	0	0	0	(Wilson , 2012)	1
DCLK3	0	0	0	0	0	1
ZNF532	0	0	0	0	0	1
C6orf225	0	0	0	0	0	1
APC	1	1	0	1	NA	0
SCNN1A	0	0	0	0	(Endoh , 2004)	1
GLP2R	0	0	0	0	0	1
SDF2L1	0	0	0	0	0	1
SHOX	0	0	0	0	(Kneip, 2011)	1
NCAM2	0	0	0	0	(Wang S. , 1999)	1
BHMT2	0	0	0	0	0	1
STAM2	0	0	0	0	0	1
CYSTM1	0	0	0	0	0	1

By analyzing our results that were obtained by running the program we observed 19 candidate genes from GBM, 17 candidate Genes from BRCA and 16 candidate Genes from COAD. We have divided the Genes into 3 categories:

4. Discussion

4.1 Genes identified in the driver datasets

For GBM and COAD we did not find any genes from the driver dataset exclusively while for BRCA we found PIK3CA as a known driver gene. In contrast, we found many genes that appear in both driver and polymorphism datasets. For GBM we identified TP53, which is a known tumor suppressor gene. For BRCA we were able to identify TP53 and CDH1, the latter of which is also known to be a tumor suppressor gene. In COAD we have again TP53 and APC, which is also known as a tumor suppressor gene. Having genes in driver and polymorphism datasets is not a common finding. One explanation for the presence of some genes in both sets is that the COSMIC polymorphism dataset has some single nucleotide mutation in the driver genes that does not affect the tumor suppression function of the gene. In our TCGA data filtration we only considered the gene name instead of taking each single nucleotide mutation for two reasons. The first one is that a much bigger data would be needed for the analysis of single nucleotide in each gene as each gene could have multiple single nucleotide substitutions. The other reason is that we had copy number and methylation data, which are different, level data sources and we wanted to unify the input data to be gene symbol only. Therefore, we narrowed the search

for driver genes into a limited number of genes that needed further investigation to exclude them from the polymorphism list.

4.2 Genes that were found in polymorphism datasets

CDH12, PHF3, TNFRSF11B and HMCN1 were genes appearing in the polymorphism dataset and they all belong to BRCA. Although, it is reasonable to exclude the genes appearing in this dataset as driver, having copy number variation as part of our data from TCGA may be due to a deletion in a tumor suppressor gene or an amplification of an oncogene that can cause cancer. It is also possible that the genes we found in the polymorphism dataset may not be excluded as driver genes because driver gene datasets are not exhaustive. Knowing these two facts, it is of great importance to investigate further the genes in the polymorphism dataset to confirm that they are not driver genes. For example CDH12 that found under this category is known of promoting the invasion of salivary adenoid cystic carcinoma (Wang J. , 2011).

4.3 Genes that were not found in either driver or polymorphism datasets "unknown"

The majority and the rest of the genes occur under this criterion. We approach these unknown genes by doing literature review on those genes. The purpose of the literature review was to exclude those genes from being a polymorphism. We did not find strong evidence that those genes are considered a polymorphism. Moreover, we find that some of these "unknown" genes have the potential to be driver genes. For example FSD1 methylation has the potential

to be a driver gene (Amandine, 2010) in GBM. Another gene MTMR3 in GBM plays a role in colon cancer, which makes it a candidate driver gene. On the other hand, CDH12 that was found by COSMIC polymorphism dataset promotes the invasion of salivary adenoid cystic carcinoma (Wang J. , 2011). Furthermore, in COAD JAKMIP1 overexpression has shown an association with cell cancer proliferation in vitro (Okai, 2013). This suggests that copy number alterations could be the driver gene of cancer and the driver gene lists are not exhaustive.

5. Future directions

It is notable that some of the candidate genes that we obtained from our run are considered candidate drivers because we had patients not covered by any genes and the set size is flexible. Increasing the set by genes that cover even one patient increases the weight of the set, even if those genes are mutated only in that one patient. To alleviate this, we may consider limiting the set of genes to those above some frequency threshold in the dataset.

Large number of genes previously not recognized as drivers may indicate the existence of limitations of the utilized method. We expected to have the majority of the genes in the driver dataset but found otherwise. The method can be improved if we take into account the effect of copy number variation and methylation. For example we could only consider copy number variation or methylation that has an effect on gene expression

References

Amandine, E. (2010). DNA methylation in glioblastoma: impact on gene expression and clinical outcome. *BMC Genomics* , 11:701.

American Cancer Society. (2013). *Cancer Facts & Figures 2013.* Atlanta: American Cancer Society.

Bertram, J. S. (2000). The molecular biology of cancer. *Molecular aspects of medicine* , 167-223.

Chen-Hsiang, Y. (2008). Combinatorial patterns of somatic gene mutations in cancer . *Combinatorial patterns of somatic gene mutations in cancer* , 2605–2622.

Ciriello, G., Cerami, E., & Sander, C. (2012). Mutual exclusivity analysis identifies oncogenic network modules . *Genome Res.* , 398-406 .

Dawson, M. A., & Kouzarides, T. (2012). Cancer Epigenetics: From Mechanism to Therapy. *Cell* , 12-27.

Endoh , H. (2004). Prognostic model of pulmonary adenocarcinoma by expression profiling of eight genes as determined by quantitative real-time reverse transcriptase polymerase chain reaction. *J Clin Oncol* , 22(5):811-9.

Gonzalez-Perez , A. (2012). Improving the prediction of the functional impact of cancer mutations by baseline tolerance transformation. *Genome Medicine* , 4:89 .

Haber, D. A., & Settleman, J. (2007). Cancer: Drivers and passengers. *Nature* , 145-146.

Hanahan, D., & Weinberg, R. A. (2000). The hallmarks of cancer. *Cell* , 57-70.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57* , 97–109.

KEGG. (2013, NOV 28). Retrieved from Kyoto Encyclopedia of Genes and Genomes: http://www.genome.jp/kegg-bin/get_htext?ko00001+K15607

Kneip, C. (2011). SHOX2 DNA methylation is a biomarker for the diagnosis of lung cancer in plasma. *J Thorac Oncol* , 6(10):1632-8.

Knudson, A. G. (2002). Cncer Genetics . *American Journal of Medical Genetics* , 96-102.

Matthias , C. (2010). Down-regulation of the Fetal Stem Cell Factor SOX17 by H33342. *J Biol Chem* , 6412–6418.

McCormick, F. (1999). Signalling networks that cause cancer. *Trends Cell Biol 9* , M53–M56.

Metropolis, N. (1953). Equation of state calculations by fast computing machines. *J Chem Phys 21* , 1087–1092.

Okai, I. (2013). Overexpression of JAKMIP1 associates with Wnt/beta-catenin pathway activation and promotes cancer cell proliferation in vitro. *Biomed Pharmacother* , 67(3):228-34.

Palmer, G. (2013, 11 28). *http://www.foundationmedicine.com/pdf/posters-abstracts/FoundationMedicine_2012-06_ASCO_Palmer.pdf*. Retrieved from Foundation Medicine: http://www.foundationmedicine.com/pdf/posters-abstracts/FoundationMedicine_2012-06_ASCO_Palmer.pdf

Smits , B. (2013). The gene desert mammary carcinoma susceptibility locus Mcs1a regulates Nr2f1 modifying mammary epithelial cell differentiation and proliferation. *PLoS Genet* .

Song, S. (2010). Mutational analysis of mononucleotide repeats in dual specificity tyrosine phosphatase genes in gastric and colon carcinomas with microsatellite instability. *APMIS* , 118(5):389-93.

Takahashi, S. (2011). Neural cell adhesion molecule 2 as a target molecule for prostate and breast cancer gene therapy. *Cancer Sc* , 102(4):808-14.

Vogelstein, B., & Kinzler, K. W. (2004). Cancer genes and the pathways they control. *NATURE MEDICINE* , 789–799 .

Wang , J. (2011). CDH12 promotes the invasion of salivary adenoid cystic carcinoma. *Oncol Rep* , 101-8.

Wang , S. (1999). Refined mapping of two regions of loss of heterozygosity on chromosome band 11q23 in lung cancer. *Genes Chromosomes Cancer.* , 25(2):154-9.

Wilson , C. (2012). Expression profiling of dipeptidyl peptidase 8 and 9 in breast and ovarian carcinoma cell lines. *Int J Oncol* , 41(3):919-32.

Vita

Yahya Abdulfattah Bokhari was born on November 29, 1981, in Makkah, Saudi Arabia. He graduated from Alshohada high school, Riyadh, Saudi Arabia in 1999. He received his Bachelor of Science in Medical Technology Sciences, from King Abdulaziz University, Riyadh, Saudi Arabia in 2004 and subsequently worked as a medical technologist in a clinical cytogenetics laboratory in King Abdulaziz Medical City, Riyadh, Saudi Arabia for four years. In 2008, he obtained the American Society Of Clinical Pathology (ASCP) Board of Certification in Cytogenetics. On 2008, Yahya received a scholarship from King Abdullah Research center to pursue his graduate study in the field of Bioinformatics. He joined the Bioinformatics program in Virginia Commonwealth University in 2010.