UNIVERSITETET I OSLO
Institutt for informatikk

A comparison of computational tools for prediction of cancer driver genes

Masteroppgave

Kristoffer Strekerud

8. august 2014

Acknowledgements

Most of all I would like to thank my supervisor, Sigve Nakken, for his patience and much needed guidance throughout the work on my thesis. His knowledge on and spirited attitude towards the subject has been an inspiration.

I would also like to thank my co-supervisors Eivind Hovig and Geir Kjetil Sandve for making themselves available to me. In terms of technical assistance, I would like to acknowledge Morten Johansen, Sveinung Gundersen, and Christian Perez-Llamas.

On a personal note I would like to thank my girlfriend, Helén, for her continued support and encouragement.

Kristoffer Strekerud

University of Oslo

August, 2014

Abstract

The significant improvements in throughput and quality of DNA sequencing technology have revolutionized our ability to identify the genetic sequence of human cells. High-throughput genome sequencing of tumor cells has furthermore enabled us to identify the complete spectrum of somatically acquired mutations of individual tumors. Among the thousands of somatic mutations that can be found in a given tumor, only a limited number are likely to be of importance for cancer development. A central challenge in cancer genomics research is thus to identify the mutations that are causally implicated in tumorigenesis, commonly known as cancer driver mutations. Genes that carry driver mutations are known as cancer driver genes. Large-scale bioinformatics analysis of tumor genomes have exploited different strategies in order to distinguish positively selected driver mutations from their neutral counterparts. The different computational approaches are frequently implemented as stand-alone software tools, allowing individual researchers with tumor sequencing data to predict likely cancer driver genes.

The actual installation and application of bioinformatics tools can be cumbersome for cancer researchers with limited computational competence, and the comparative performance of driver gene prediction results with different approaches and algorithms would therefore be difficult to obtain. To this end, we have implemented a single computational workflow for driver gene prediction within the Galaxy framework, a user-friendly web-based platform for data intensive biomedical research. Our workflow accepts a single input file with tumor DNA variation data and will subsequently run three of the most commonly used tools for cancer driver prediction, that is, IntoGen, MutSigCV, and DrGap. A report is generated that indicates the comparative performance of the individual tools (i.e. where the tools are in agreement, and where they are not) as well as simple visualization of their overlapping predictions. The workflow and accompanying tools are available at http://insilico.hpc.uio.no:40065. The workflow described in this thesis is available at: http://insilico.hpc.uio.no:40065/u/strekerud/w/driver-gene-tool-comparison. We have applied our workflow on publicly available datasets from

six major tumor types, and we discuss how usage of combinations or overlaps of driver gene prediction lists can increase the number of true positives found.

Contents

1 Introduction

The majority of human cancers arise from an accumulation of genetic aberrations in somatic cells (Garraway and Lander 2013). That does not mean that all the genetic abnormalities present in a tumor genome have been involved in cancer development. Tumor genomes of different tissues may contain from tens to thousands of somatic mutations. Our current understanding is that only a few, critical aberrations are causally implicated in tumorigenesis, while the rest are relatively benign and make little or no contribution at all (Vogelstein et al. 2013). The conceptual difference between these two types of aberrations in a cancer genome is commonly referred to as 'driver' versus 'passenger' mutations.

A driver mutation is causally implicated in tumorigenesis. It has conferred growth advantage on the cancer cell and has been positively selected in the microenvironment of the tissue in which the cancer arises (Stratton et al. 2009). Genes that carry driver mutations are known as cancer driver genes. In contrast to driver mutations, a passenger mutation has not been selected, has not conferred clonal growth advantage and has therefore not contributed to cancer development. Many passenger mutations are present within cancer genomes because several somatic mutations without any phenotypic consequence tend to occur during cell division, e.g. as DNA replication errors (De and Michor 2011). Others have also suggested that a driver mutation should occur in multiple tumors more often than would be expected by chance (Akavia et al. 2010). This frequency-based view will however to some extent be confounded by the observation that driver mutations appear to target distinct cellular signaling and regulatory pathways (Vandin et al. 2012). Individual cancer patients may very well exhibit a unique combination of somatic mutations that are sufficient to perturb these pathways. Such mutational heterogeneity will thus represent a problem when driver mutation prediction is done solely from frequency of occurrence. Although predictions of cancer driver pathways or merely cancer driver gene sets appear highly relevant, we have limited our study to the identification or prediction of single cancer driver genes.

In order to understand the underlying mechanisms of tumorigenesis, a first step andcentral goal of many large-scale cancer genome analyses is the identification of cancer driver genes that, by definition, carry driver mutations. A key challenge in this respect is to identify properties that distinguish driver from passenger mutations. Various bioinformatics algorithms have been proposed, each exploiting different structural signatures associated with somatic mutations that are under positive selection. Most methods identify genes that are mutated more frequently than expected from the background mutation rate. Such methods are known as recurrence-based or frequency-based approaches. A challenge in this respect is to correctly estimate the background rate in order to minimize the number of false positive predictions (Dees et al. 2012; Lawrence et al. 2013). Driver genes mutated at very low frequency are however difficult to detect using this approach (in addition to the pathway challenge, as mentioned above), and that is why other signals of positive selection across tumor samples have been explored (Tamborero et al. 2013b). Examples of such signals include a high rate of non-silent mutations versus silent mutations, a bias towards the accumulation of functional or deleterious coding mutations, a In order to understand the underlying mechanisms of tumorigenesis, a first step and clustering of mutations in certain regions or functional domains of a protein sequence, or an overrepresentation of mutations in specific functional amino acids, such as phosphorylation sites (Reimand and Bader 2013; Greenman et al. 2007; Gonzalez-Perez and Lopez-Bigas 2012; Tamborero et al. 2013a).

Intuitively, one would expect that the different types of cancer driver genes would exhibit the signals of positive selection exploited by the approaches outlined above to varying degrees. For example, it has been found that cancer mutations are known to cluster in specific residues in oncogenes more strongly than in tumor suppressors (Stehr et al. 2011). In addition, since different cancer subtypes display great mutational heterogeneity, one could also suspect that the performance of different approaches will vary according to tumor type (Vogelstein et al. 2013). Consequently, when applying the methods on real cancer mutation data, one should expect that different subsets of candidate drivers will rank at the top of lists

of driver candidates identified by each method. Furthermore, the implementation details of each method are likely to influence its results. For example, frequency-based methods with a loosely defined background mutation rate will identify a larger number of driver candidates at the cost of higher number of false positives. On the other hand, methods implementing stricter models will identify shorter, more specific lists but might miss some true cancer driver genes.

To our knowledge, there is no bioinformatics tool available within the cancer research community that allows a direct comparison of results produced by different approaches for cancer driver gene prediction. Importantly, successful installation and application of bioinformatics software often represent a significant hurdle for cancer researchers with limited computational or programming competence. In fact, software installation may also represent a significant challenge even for computational biologists, often as a result of poor documentation by many bioinformatics software developers. Making the most common tools for driver gene prediction accessible in an easily accessible web framework would thus be of great value in itself. The primary aim of this thesis is to develop and implement a tool that seamlessly performs gene driver prediction using a set of common approaches, followed by a simple visualization of their comparative performance.

The prediction algorithms we have included are DrGap, IntoGen, and MutSigCV, which exploits most of the dimensions in driver signal detection (Lawrence et al. 2013; Hua et al. 2013; Gonzalez-Perez et al. 2013). Our tool has been implemented as a workflow within the Galaxy Project **F**ramework, which is an open, web-based platform for data intensive biomedical research (Giardine et al. 2005). The benefits of tool development within Galaxy are diverse, both from a bioinformatics developer standpoint as well as the end user that wants to do various genomic analyses. The primary benefit for the end user is ease of use, reproducibility and analysis flexibility. From a developers view, a generic framework for workflow management is especially powerful for development of complex analysis pipelines that combine multiple tools.

A secondary aim of this project has been to explore how the different approaches of driver gene prediction perform for different cancer subtypes, and whether a looking at overlapping (i.e. consensus) lists of candidate drivers from different algorithms could increase performance. In order to address this question, we have applied our cancer driver workflow on publicly available somatic mutation datasets from The Cancer Genome Atlas (TCGA) project

2 Background

2.1 DNA and Mutations

Deoxyribonucleic acid (DNA) holds the biological instructions for life, stored inside us. Coiled tightly around proteins called histones, the DNA is packaged within 23 pairs of chromosomes, found within the nucleus of every cell in our body (Brown 2006). Our DNA consists of long strings of molecules called nucleotides. These nucleotides are linked together consisting of a phosphate group, a sugar group and one of four types of nitrogen bases: adenine (A), thymine (T), guanine (G) and cytosine (C). These nucleotide strings make up, at the most basic level, the coding of our DNA. The most stable form of DNA is organized using hydrogen bonds between base pairs, binding adenine with thymine, and guanine with cytosine. This is how most people see DNA, in its "ladder" form. Although this form is the most common, DNA also appears as single stranded.

A change in DNA, the genetic sequence, is called a mutation. These changes in the genetic sequence can occur when errors are made in the copying of DNA, resulting in a copy that is not exactly like the original. Mutations can also occur due to external influences or environmental mutagens when these induce damage in DNA and the repair machinery in the cells are unable to correctly repair the damaged DNA. An important distinction when it comes to mutations is made between germline and somatic mutation events. Germline mutations are mutations that occur in cells transmitted between generations (i.e. in egg or sperm cells, also known as hereditary mutations), while somatic mutations occur in somatic tissue that is not inherited. Most cancers are caused by somatic mutations, although there are also cases of hereditary cancer types.

With respect to the actual mutation event in the genetic sequence, there are several different types that can take place, the main types being substitutions, insertions, deletions and frameshifts:

Single base substitution - A single nitrogen base is exchanged for another:

CTGGAG --> CTGGGG

Even though only a single nitrogen base has been changed, this can have a significant impact on the protein coded. The substitution of this base will lead to a different amino acid being coded during translation. There are three different outcomes of a substitution when this occurs in the protein-coding sequence. The first, a *missense* mutation, is where an altered codon (set of three nucleotides) leads to an incorrect amino acid being coded into the new protein. The second type, a *nonsense* mutation, is where the new amino acid is treated as a stop codon which terminates a protein from being coded before it's supposed to. The third kind is called a *silent* mutation, a case in which the same amino acid is encoded, only with a different codon. Missense and nonsense mutations are commonly referred to as non-silent mutations.

Base Insertion - new bases have been inserted into the sequence, e.g.:

CTGGAG --> CTGGTGGAG

Here, an extra nucleotide is added to the sequence. This could happen when a strand "wrinkles", allowing room for an extra nucleotide.

Base deletion - a section (one ore more bases) of the sequence is lost:

CTGGAG -> CTAG

As with an insertion, a "wrinkle" in the DNA strand can cause one or more nucleotides to be skipped during DNA replication.

Frameshift (Insertions or deletions cause the way the sequence is read to fundamentally change):

The fat cat sat -> he fat can sat (Deletion),

leading to the sequence being read as “hef atc ats at”

Because of one or more insertions or deletions, the entire “frame” of codons using during translation has been shifted. This will happen when the number of insertions/deletions is not a multiple of three, as codons are coded by groups of three amino acids.

It is important to recognize that mutations come with consequences. Somatic mutations influencing cell division, especially those that allow cells to divide uncontrollably, have been identified as the basis of many forms of cancer. Mutations can also lead to increased susceptibility to illness or disease. In some cases, mutations have been proven beneficial to an organism by making it better able to adapt to environmental factors. No matter how you see it, changes made to molecules can have great impact on the physical characteristics of an organism (Miko and Lejeune 2009)

2.2 Genes

The word “Gene” is a concept that has undergone much change over time. When first coined, it was more of an abstract concept, treated as a unit of inheritance that ferried a characteristic from parent to child. When biochemistry became known, we learned that each gene was connected to an enzyme or a protein. After a while, molecular biology allowed us to picture genes as real, physical things, which are sequences of DNA that can be converted into RNA, which is used as the basis for building an associated protein. At this point we can picture genes sitting as beads on the larger, coiled DNA molecule.

The ladder concept is the definition stilled used by many scientists today, while others argue that this is, at best, only a crude approximation of the actual complexities of genes. Recent studies have suggested that ribbons of RNA can be generated from both strands of DNA, rather than from just one as was conventionally thought. (Pearson H. 2006)

Karen Eilbeck, coordinator of the Sequence Ontology consortium at the University of California in Berkeley states that it took 25 scientiests the better part of two days to reach a definition of a gene that they could all work with. “We had several meetings that went on for hours and everyone screamed at each other”. This still leaves us without a clear definition of what a gene is, which makes life difficult for bioinformaticians, who have become dependant on using computer programs to spot landmark sequences in DNA that signal where one gene ends and another begins. For the purpose of this thesis we will adhere to the definition agreed upon by the consortium: "A locatable region of genomic sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions and/or other functional sequence regions."

(Pearson H. 2006)

2.3 Cancer and its development

In our cells we have hundreds of genes that control the process of cell division. There is a fine balance between activities promoting cell division, and activities suppressing it. As part of this balance, apoptosis (programmed cell death) controls the destruction of damaged cells.

As a cell accumulates mutations, there is a chance that one or more of these mutations will affect the balanced cell division, causing the cells to become cancerous, growing out of control. The Cancer Genome Project has found that most cancer cells possess 60 or more mutations. Many of these mutations will not be involved with cancer growth, and finding the ones that are pose a great challenge for medical researchers. We want to know which of the mutations involved are responsible for different types of cancer. For example, we know that certain growth-promoting genes are commonly mutated in cancer cells, such as the gene coding the protein Ras. In other examples, genes coding the suppression of cell proliferation have been inactivated by mutations.

Because a cell with a growth advantage is able to copy itself at a faster rate than normal cells, eventually it will be able to outperform those normal cells in the battle for resources. At this point, the tumor is benign. When following generations of the cancerous cell intensify this advantage and becomes able to break through tissue, the tumor is defined as malignant. When the cancerous cells start entering the bloodstream or the lymphatic system, this allowing them to travel to other places in the body, it has entered the stage called metastasis.

(Cell Division and Cancer, Scitable, Nature.com, O'Connor and Adams 2010)

2.4 Discovering Driver Genes with bioinformatics

Several approaches towards the identification of driver genes have been implemented through computer software. With mutation data given as input these tools are able to return with a list of genes identified as drivers. Looking through gene sequencing data manually would be a herculean task, close to impossible, but with the help of computer power we are able to go through large amounts of data in a short amount of time.

Most commonly these methods identify genes that are mutated more frequently than expected from the background mutation rate. This approach is applied by tools such as MuSIG-SMG (Dees et al. 2012). However, this approach would not be able to detect driver genes that are mutated at a very low frequency. Another approach, applied by tools such as OncodriveFM (Gonzalez-Perez and Lopez-Bigas 2012), attempts to identify genes that exhibit signals of positive selection, such as a high rate of non-silent mutations compared to silent mutations (the distinction between silent and non-silent mutations are explained in section 2.1).

Based on the observation that gain-of-function mutations tend to occur specifically in particular residues or domains along the genome, tools like OncodriveCLUST exploit the tendency to sustain mutations in certain regions of the protein sequence (Tamborero et al. 2013a). Tools like ACTIVEbias exploit the overrepresentation of mutations in specific functional residues, such as phosphorylation sites (places where a phosphate group is added to a protein or other organic molecule) (Reimand and Bader 2013). The three tools used for the purposes of this thesis and their approaches have been explained more thoroughly in the Methods and implementation part of this thesis.

2.5 Data Representation of DNA variation

When looking into which tools to use in the comparison done in this thesis, it quickly became apparent that there is very little consensus across the board.

While the information used by the tools to classify the driver genes is often very similar, the input formats vary greatly, ranging from customized versions of the MAF (Mutation Annotation Format) and VCF (Variant Call Format) formats to entirely new tabulated formats made specifically for the tool in question.

For the purpose of this thesis we will employ the MAF format. Still, because of its popularity, information on the VCF format is also included.

Mutation Annotation Format (MAF)

MAF is the current format used by The Cancer Genome Atlas (TCGA) to represent somatic and/or germline mutations. It is a tab-separated file consisting of 34 columns containing information on the gene and chromosome affected, as well as the specific position and type of mutation for each sample.

Following categories of somatic mutations are reported in MAF files:

- Missense and nonsense
- Splice site, defined as SNP within 2 bp of the splice junction
- Silent mutations
- Indels that overlap the coding region or splice site of a gene or the targeted region of a genetic element of interest.
- Frameshift mutations
- Mutations in regulatory regions

(TCGA 2013)

The first line of a MAF file will always look like the line below, containing the names of all the headers used in the MAF format.

"Hugo_Symbol Entrez_Gene_Id Center NCBI_Build Chromosome Start_position End_position Strand Variant_Classification Variant_Type Reference_Allele Tumor_Seq_Allele1 Tumor_Seq_Allele2 dbSNP_RS dbSNP_Val_Status T

umor_Sample_Barcode Matched_Norm_Sample_Barcode Match_Norm_Seq_All ele1 Match_Norm_Seq_Allele2 Tumor_Validation_Allele1 Tumor_Validation_A llele2 Match_Norm_Validation_Allele1 Match_Norm_Validation_Allele2 Verific ation_Status Validation_Status Mutation_Status Sequencing_Phase Sequenc e_Source Validation_Method Score BAM_file Sequencer Genome_Change Annotation_Transcript Transcript_Strand Transcript_Exon Transcript_Position cDNA_Change Codon_Change Protein_Change is_coding is_silent categ

The remaining lines will then hold the mutation data, e.g.:

PHTF1 10745 broad.mit.edu 37 1 114242392 114242393 + Frame_Shif t_Ins INS -
T T LUSC-18-3406-Tumor LUSC-18-3406-
Normal Phase_I Unspecified Illumina
GAIIx g.chr1:114242392_114242393insT uc009wgp.1 -
16 2527_2528 c.2075_2076insA c.(2074-2076)AAGfs p.K692fs 1 0 7
A1BG 1 broad.mit.edu 37 19 58861774 58861774 + Frame_Shift_Del
DEL C - - LUSC-18-3406-Tumor LUSC-18-3406-Normal"

At the moment, several MAF files are available for download from the TCGA website hosted by the National Cancer Insitute (https://wiki.nci.nih.gov/display/TCGA/TCGA+MAF+Files). The MAF format is accepted directly by the MutSigCV tool used in this thesis, and will also possibly work as direct input for the Intogen tool in the future. When running the Driver Gene Tool Comparison workflow in Galaxy in this thesis, separate tools will be used that translate the MAF format into custom formats for the purpose of running the Intogen and DrGap tools on the same input file.

Variant Call Format (VCF)

The Variant Call Format is another format, like MAF, that is used to stored gene sequence variants. VCF has risen in popularity because it stores variants only, unlike older formats like the General Feature Format (GFF) which stored all genetic data, which therefore contains large amount of redundant data. The VCF format is stored as a text file, containing lines of meta-information, a header line, and then data lines containing information about a position in the genome. The variant info is

always contained within eight mandatory columns: #CHROM (chromosome), POS (genomic position), ID (unique variant identifier, REF (reference allele), ALT (alternate allele), QUAL (quality score for detected variant), FILTER (custom filters for variant filtration), and INFO (semicolon-separated tags with various kinds of annotations).

Example metadata and headers:

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With
Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS ID       REF ALTQUAL FILTER INFO                              FORMAT
        NA00001        NA00002        NA00003
```

Example genome data:

```
20      14370   rs6054257 G  A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2
GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20      17330   .         T       A      3 q10      NS=3;DP=11;AF=0.017
        GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20      1110696 rs6040355 A          G,T     67   PASS
NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2
2/2:35:4
```

(VCF (Variant Call Format) version 4.0, 1000genomes.org)

2.6 Data Banks

All of the input files used in this thesis have been provided by The Cancer Genome Atlas' MAF file archive. The Cancer Genome Atlas (TCGA) is a coordinated effort to further the understanding of the molecular basis of cancer through the application of genome analysis technologies, which includes whole genome sequencing. TCGA is created as a joint effort of the National Cancer Institute (NCI) and the National Human Genome Research Insitute (NHGRI).

In order to provide some sort of accuracy measuring with the results given by the tools in this thesis, we have implemented a system that compares all results with a list of genes provided by The Cancer Gene Census (http://cancer.sanger.ac.uk/cosmic/census). This is a list of genes implicated via mutation in cancer compiled by experts at the Trust Sanger Institute. This list is updated regularly/as needed and currently more than 1% of all human genes are contained in this list.

For the purpose of allowing the user quick access to more information on the genes presented in the final report generated by the tools in this thesis, all genes in the report will contain a direct link to the corresponding NCBI (National Center for Biotechnology Information) gene information page. This is achieved by comparing the gene HUGO id with the corresponding gene id used by the NCBI. NCBI provides a file with gene names and corresponding id's for this purpose. The file used in this thesis can be found here.

2.7 Statistical Significance

For the purposes of this thesis we have considered the results generated by MutSigCV, DrGap and Intogen to be statistically significant if they have a p-value of less than 0.05. The output files generated by the tools each have a column dedicated to p-values for each gene.

P-values are used in statistics as the estimated probability of rejecting the null hypothesis for the study. In this thesis the null hypothesis would be “Is this gene a driver gene?”. Most authors agree that results with a p-value less than 0.05 can be considered statistically significant. This means that the result is considered to have a less than 1 in 20 chance of being wrong.

When the Driver Gene Tool Comparison tool goes through the different results generated by the tools used in this thesis work, it looks for results with a p-value less than 0.05, extracts them, and adds them to the report.

3 Methods & Implementation

In this chapter, we will outline in more detail the different algorithms for cancer driver prediction that has been implemented in our comparison tool. We will also specify the implementation details of our tool for comparison of algorithms, that is the Driver Gene Tool Comparison (DTGC), which has been implemented in the Galaxy framework.

3.1 Algorithms for cancer driver detection

Each of the three tools used in this comparison utilize different signals and methods in order to classify genes as driver genes. This section will cover each of the tools' method of achieving this classification, as well as the underlying scientific justifications given for using said methods. Understanding the underlying differences between the three tools will give us the necessary clues to understand the similarities, as well as the differences, in the results we end up with when running our comparison tool on a set of somatic mutation data.

3.1.1 MutSigCV

The MutSigCV-developers have found that when current methods used to find driver genes are met with very large sample sizes, the number of genes identified as significant will be in the hundreds. They have found that based on the biology behind them many of these are implausible as driver genes and could in these cases be regarded as false positives. They mention encoding olfactory receptors and the muscle protein Titin as examples of these biological factors. In one case where they tested on data from lung squamous cell carcinoma, a quarter of the genes found to be significant were encoding olfactory receptors, and they also found that the list of significant genes found also contained an inordinate amount of extremely large proteins, such as the before mentioned Titin.

The developers wanted to show that this problem stems mainly from mutational heterogeneity across the genome. By conducting a study based on data sampled at the Broad Institute they were able to analyze three types of heterogeneity: heterogeneity across patients with a given cancer type, heterogeneity in the mutational spectrum of the tumors and most importantly, regional mutational

heterogeneity across the genome. It was when studying this third kind of heterogeneity that they noticed a significant variation in mutation frequencies across the genome. Because of this, they were able to find a strong correlation between somatic mutation frequency in cancers and gene expression level. They found that the mutation rate is almost threefold high in the bottom expression level percentile than in the top one. This observation paired with recent studies reporting that germline mutation rates are correlated with DNA replication time proved to be the two factors that would explain the false positives previously mentioned. Based upon these observations, the team developed MutSigCV, an algorithm that corrects for variation by using patient-specific mutation frequency and spectrum, and gene-specific background mutation rates incorporating expression level and replication time.

When MutSigCV ran the lung cancer data mentioned previously, the number of driver genes found was reduced from 450 to only 11 genes, most of these 11 being genes previously reported to be lung cancer driver genes. The MutSigCV authors could in that way demonstrate that mutational processes have to be taken into account when attempting to identify driver genes. They also note that even though MutSigCV have solved the most serious current problems with driver gene identification, future solutions will probably have to include observed mutation rates from whole-genome sequencing (Lawrence et al. 2013)

3.1.2 DrGap

The DrGap developers define driver genes, from a statistical standpoint, as "[those] genes for which the nonsilent mutation rate is significantly higher than a background (or passenger) mutation rate". They also state that some biological considerations, such as length of protein-coding regions (called CDS), variation in transcript isoforms, variation in mutation types, differences in background mutation rates, redundancy of genetic code, and number of mutations in one gene also have to be made. According to its developers, DrGap has been developed for the purpose of combining common statistical approaches with bioinformatics tools in order to find driver genes. In figure 1, you will see the "DrGap Analysis Pipeline", taken from (Hua et al. 2013). In our use of DrGap, only tumor mutation data will be used (not the optional user-defined gene sets or BAM) in order for the user to run an entire comparison (including MutSigCV and Intogen) on one set of input.

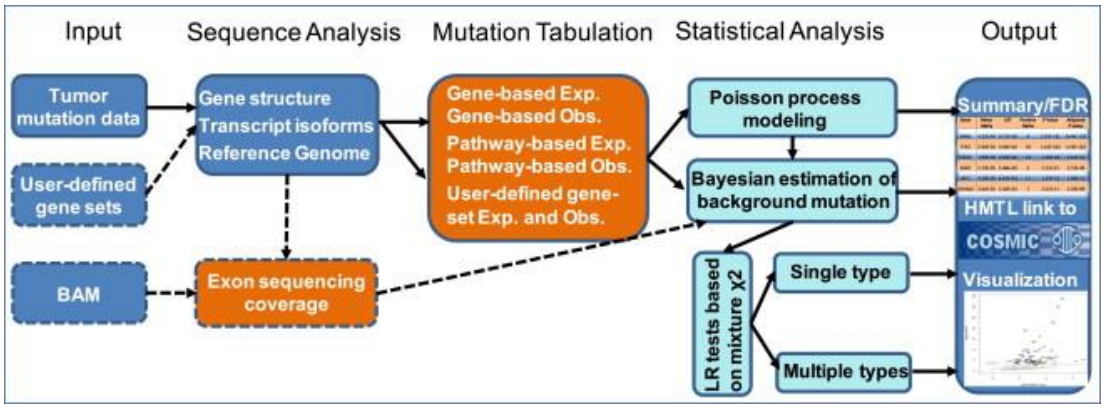


Figure 1: Outline of DrGap pipeline

The developers have found a way to pair their biological knowledge of the properties of driver genes with statistical interpretations. This allows them to run statistical analysis on the data to see whether or not each gene possesses the required properties to be classified as a driver gene. This pairing is done as described in the table below:

Biological Knowledge	Statistical Interpretation
transcript isoforms	sum aggregate of CDS from multiple isoforms of the same gene
variation in mutation types	consider 11 different mutation types
background mutation rates	beta prior of η_{ij} which is background rate of mutation type *j* in individual *i*
differences in background mutation rates	estimate separate mutation rates η_{ij} for each individual tumor
redundancy of the genetic code	define N_{jk} and M_{jk} as the number of base pairs in CDS of gene *k* that can give rise to nonsilent and silent mutations
multiple mutations in one gene	addressed by the Poisson process
sequencing coverage	c_{ik} is the proportion of CDS with a minimum eight sequence coverage in both a tumor and its matched normal DNA from individual *i*
CDS size	$\sum_j (N_{jk} + M_{jk}) = 3L$ where *L* is length of CDS for gene *k*

With this approach, the DrGap team has experienced a greater sensitivity when looking for driver genes when compared to older statistical approaches such as Bernoulli, Binomial, Poisson and Poisson-Gamma. They have noted a particularly high sensitivity compared to the other methods when there are multiple types of driver mutations in the tumor. In tests against data from TCGA studies DrGap consistently outperforms the previously mentioned statistical approaches.

The developers also compared results using DrGap for identifying driver pathways. When compared another piece of software, PathScan, DrGap consistently came out on top, especially in cases where there are fewer driver genes that are mutated (thus requiring better sensitivity).

DrGap has also performed well when compared to other cancer sequencing studies. Using the same dataset Ding et al. identified 22 driver genes and Young and Simon were able to identify 28 driver genes. Collectively they discovered 30 driver genes using their methods. Testing on the same dataset, DrGap identified 59 driver genes.

Within these 59, 29 were the same as those discovered by Ding, Young and Simon with the last one missed by DrGap being the least significant according to their studies.

(Xing Hua 2013)

3.1.3 Intogen

Intogen is run as a combination of two tools, OncodriveFM and OncodriveCLUST. OncodriveFM is created to detect genes that are biased toward the accumulation of mutations with high functional impact, while OncodriveCLUST picks up genes with mutations that tend to cluster in particular regions of the protein sequence. Driver genes as defined by the developers of both tools as “genes whose mutations are selected during tumor development”. Therefore, both tools take this into account when looking for driver genes.

When running Intogen, the tool first measures predicted functional impact. This is done by retrieving impact values using three well known methods, SIFT, Polyphen2 and MutationAssessor. After this is done, Intogen runs OncodriveFM and OncodriveCLUST on the data and assigns each gene p-values. The entire pipeline is executed by a workflow management system called Wok, which is accessible as open source, in the manner shown in Figure 2.

After a runthrough, the user is presented with p-values for both OncodriveFM and OncodriveCLUST as well as a third value called IntogenDriver. In the case where Intogen identifies the gene as a driver (based on p-values from OncodriveFM and OncodriveCLUST) the IntogenDriver value is set to 1.

In an example runthrough on 37 medulloblastoma samples, the Intogen were able to identify three genes not found by another research team, with the gene SF3B1 being noted as especially interesting because of its function of “encoding a splicing factor known to drive hematopoietic malignancies”. (Gonzalez-Perez and Lopez-Bigas 2012)

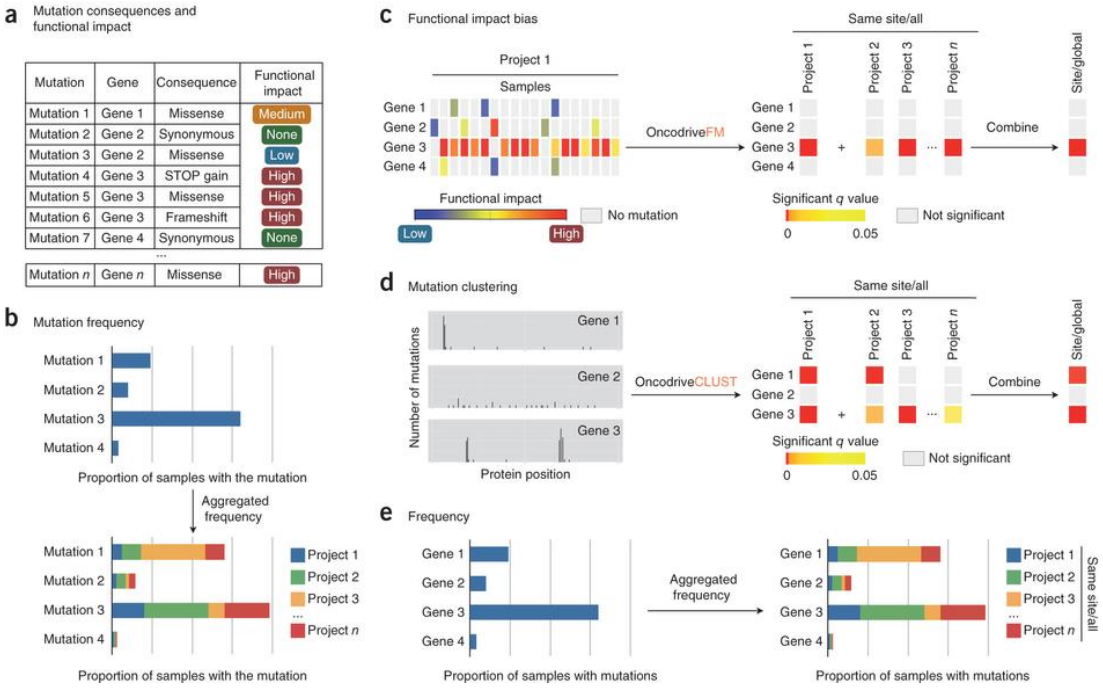

a
Mutation consequences and functional impact
Mutation | Gene | Consequence | Functional impact
Mutation 1 | Gene 1 | Missense | Medium
Mutation 2 | Gene 2 | Synonymous | None
Mutation 3 | Gene 2 | Missense | Low
Mutation 4 | Gene 3 | STOP gain | High
Mutation 5 | Gene 3 | Missense | High
Mutation 6 | Gene 3 | Frameshift | High
Mutation 7 | Gene 4 | Synonymous | None
Mutation n | Gene n | Missense | High
b
Mutation frequency
Mutation 1
Mutation 2
Mutation 3
Mutation 4
Proportion of samples with the mutation
Aggregated frequency
Project 1
Project 2
Project 3
Project n
c
Functional impact bias
Project 1
Samples
Gene 1
Gene 2
Gene 3
Gene 4
OncodriveFM
Same site/all
Project 1
Project 2
Project 3
Project n
Site/global
Combine
Functional impact
Low
High
No mutation
Significant q value
0
0.05
Not significant
d
Mutation clustering
Number of mutations
Protein position
OncodriveCLUST
e
Frequency
Proportion of samples with mutations
Same site/all

Figure 2

3.2 Implementation

When starting out with this project, we faced the challenge of running through three different driver prediction tools and generating a visualization of the comparative performance of those tools. This with only a MAF file given as input. We broke the challenge into the following steps:

1. Handle the conversions between input formats used by the tools (MAF2DRGAP, MAF2INTOGEN).
2. Run each tool using the data given (MutSigCV, DrGap, and Intogen).
3. Collecting and handling the different output formats given by the tools, and then building an HTML report on that data.

Seeing as we accept a single MAF file as input in our comparison, it will have to be converted to the custom DrGap and Intogen formats in order to be run. In order to handle this I have created the MAF2DRGAP and MAF2INTOGEN tools, to be run as an intermediary between the MAF input file given and the DrGap and Intogen tools in the workflow used to run the comparison in Galaxy.

In the case of MAF2INTOGEN, the developers have told me in emails that they will have MAF support in the future, but until then the custom Intogen format will have to be used. These two tools extract the corresponding headers from the MAF file and write them to new input files.

Even though MAF2DRGAP and MAF2INTOGEN are quite simple in its usage, similar tools for converting MAF files do not exist, or more likely, has not been made available to the public. Any person looking to run DrGap or Intogen (locally) today will have to adhere to the very specific input formats dictated by the tools. To alleviate people who find themselves in my position in the future, I will make the tools publicly available through Google Code here: https://code.google.com/p/maf2intogen/ and https://code.google.com/p/maf2drgap/

3.2.1 MAF2DRGAP

The columns extracted from MAF directly are sample_id, gene, chr, pos, ref and var.

The mutation types can't simply be extracted, but have to be translated. MAF2DRGAP extract the columns from MAF(left side) and then translates them to the custom DrGap format (right side). It ignores any variants with mutation types (called variant classifiers in the MAF format) not supported by DrGap. The supported types are silent, missense, nonsense, splicing, Fs_indel and nFs_indel mutations.

Missense_Mutation —> missense
Nonsense_Mutation —> nonsense
Silent —> silent
Splice_Site —> splicing
Frame_Shift_Ins —> Fs_indel
Frame_Shift_Del —> Fs_indel
In_Frame_Ins —> nFs_indel
In_Frame_Del —> nFs_indel

When the translation is complete, the new DrGap format looks like this:

S003 IPO11 5 61733126 G T missense

3.2.2 MAF2INTOGEN

The custom format required by Intogen looks as follows:

chromosome: just the name or number, with or without the 'chr' prefix.
start: start and end are reversed in the case of insertions.
end
strand: defined as + (forward) or - (reverse) (+1, 1, -1 are also allowed).
allele: pair of alleles as REF>ALT, where REF is the reference nucleotide and ALT the alternative allele found. REF and ALT can be '-' to express insertion or deletion.
sample_id: Identifier of the sample.

Example:

X 37901198 37901198 - C>T TCGA-AB-2927-03A-01W-0755-09

MAF2INTOGEN pulls the necessary columns from the MAF input file, and then produces a custom file adhering to the Intogen format. It ignores and discards lines (such as comments in the header) in the MAF file that it deems redundant.

3.2.3 Driver Gene Tool Comparison

Driver Gene Tool Comparison (DGTC) is the tool I created to take care of the final part of the Galaxy workflow used in this thesis, namely the comparison of the results given as output from DrGap, MutSigCV and Intogen. It is written in its entirety in Java, and is deployed as a JAR file.

By extracting the p values from the result files, keeping the ones within the limit (0.05) and discarding the rest, the program builds a database of significant genes. The program also stores where the three tools are in agreement, to be used when creating the report later. This information that the program stores, could feasibly be put to further use, not only for the purpose of creating the report, but for statistical usage. It should be interesting to see which genes are considered significant across several data sets, as well as repeatedly across all three tools.

In order to provide the user with a visualized report of the tool's findings, the program creates an HTML report, which is viewable in Galaxy after a workflow run-through. The report contains information on the original input file used, the number

of significant genes found per tool, as well as a table listing the genes, showing which genes are marked as significant by each tool.

Each gene listed in the report will contain a link to the genes provided by The National Center for Biotechnology Information (NCBI). By clicking one of these links, the user will be able to look up specific information on the gene, such as genomic context or associated conditions.

Also as part of the report, DGTC cross references its results with The Cancer Gene Census (http://cancer.sanger.ac.uk/cosmic/census), which is "an ongoing effort to catalogue those genes for which mutations have been causally implicated in cancer". In doing this cross referencing, we are able to provide the user with information on whether or not the genes found to be statistically significant are represented in the Cancer Gene Census or not.

The report also contains a Venn diagram, created using the Google Charts API (through the charts4j library, requires an internet connection). The diagram illustrates the degree to which the tools agree about which genes are significant, as such:

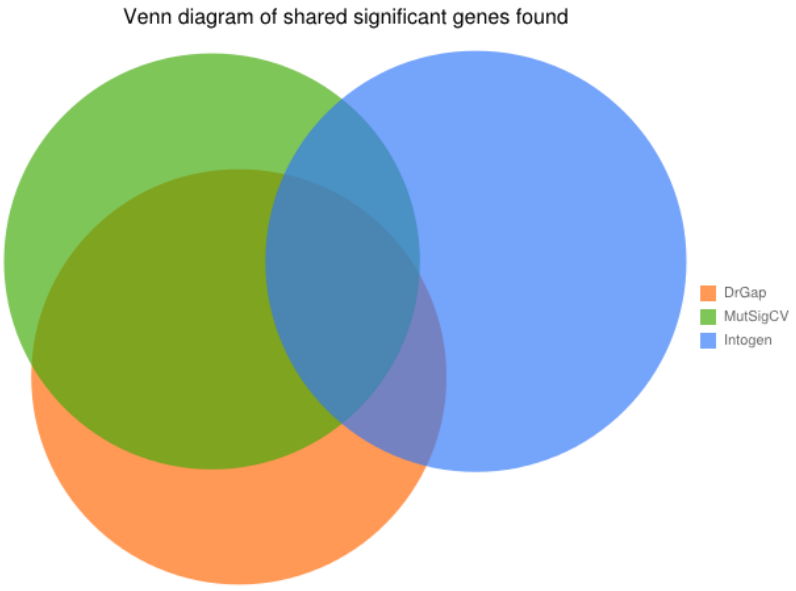


Figure 3

The report itself can be viewed in the results section of this thesis.

Running DGTC

DGTC takes the following arguments to run:

1. The path to where you would like the final report created by DGTC to be stored.
2. The path to the original input file used in the comparison (MAF)
3. Path to MutSigCV result file.
4. Path to DrGap result file.
5. Path to Intogen result file.

Running DGTC from the terminal is done in the following way:

java --jar reportpath inputpath mutsigcvresults drgapresults intogenresults

3.2.4 The Galaxy Project Framework and its implementation

Galaxy is developed as a collaboration by the Center for Comparative Genomics and Bioinformatics at Penn State University and the Mathematics and Computer Science departments at Emory University. It is a web-based platform that is designed to aid researchers doing computational biomedical research. The project adheres to the following three principles as described on their website:

1. Accessible: Users without programming experience can easily specify parameters and run tools and workflows.
2. Reproducible: Galaxy captures information so that any user can repeat and understand a complete computational analysis.
3. Transparent: Users share and publish analyses via the web and create Pages, interactive, web-based documents that describe a complete analysis.

(Galaxy Project Wiki)

For my implementation I have set up a local installation of Galaxy on Insilico, the high performance computer provided by the institute for this purpose. Installing and running Galaxy in itself is fairly simple. It is done by cloning a directory from the Galaxy Project Bitbucket site, and then using Mercurial to update to a stable version.

You run Galaxy by moving to your Galaxy distribution directory and then running the following command:

sh run.sh

Provided that you have either Python 2.6 or Python 2.7 installed on your system, Galaxy will run successfully. When all configuration is done by the run script (this will take a while during its first runthrough), Galaxy supplies you with a web platform at the address specified in the “universe_wsgi.ini” file in your Galaxy directory. If no address is given, the platform will be accessible at the localhost address (usually 127.0.0.1).

At this point the Galaxy platform is ready to run tools on its host. Several default tools are available by default through Galaxy, but for this thesis we want to add tools of our own (MutSigCV, DrGap and Intogen). These will have to be configured manually in order for us to run them through the Galaxy framework. This is done through Galaxy's tool configuration system.

First you will have to add your tool to Galaxy's "toolbox" by adding your own section to "tool_conf.xml" in your Galaxy directory as such:

```
<section name="Driver Gene Tool Comparison" id="DGTC">
    <tool file="DGTC/MutSigCV/MutSigCV.xml" />
    <tool file="DGTC/drgap/drgap.xml" />
    <tool file="DGTC/Intogen/intogen.xml" />
    <tool file="DGTC/MAF2INTOGEN/MAF2INTOGEN.xml" />
    <tool file="DGTC/MAF2DRGAP/MAF2DRGAP.xml" />
    <tool file="DGTC/DTC/dtc.xml" />
 </section>
```

When starting up, Galaxy will then look for the individual tool configuration files given in "tool_conf.xml". If the configuration files are found, each individual tool will be added to Galaxy as a runnable tool. Each tool is configured using the following xml-code format:

```
<tool id="[Tool id]" name="[Name of tool as it is to be presented in Galaxy]">
        <description> A short description of the tool, to be shown in the tool
        chooser menu </description>
        <help> A longer description of how the tool </help>
        <command>The command used to run the tool from the terminal, allowing
        the tool to be run from within Galaxy </command>
        <inputs>
            <param name="input" type="data" label="Description of the
        required input">
        </inputs>
        <outputs>
            Same as inputs, takes params.
        </outputs>
</tool>
```

The DGTC workflow (Galaxy)

The Galaxy Project allows the implementation of tool workflows through its web interface. Given just a single file as input (In our case, a MAF file), the user is able to specify the order in which each tool is to be run. Each tool will not run until all its input requirements have been met. In the case of our DriverToolComparator, it will not run until it has received output files from all three tools. Using the workflow interface provided by Galaxy we were able to create the following workflow:

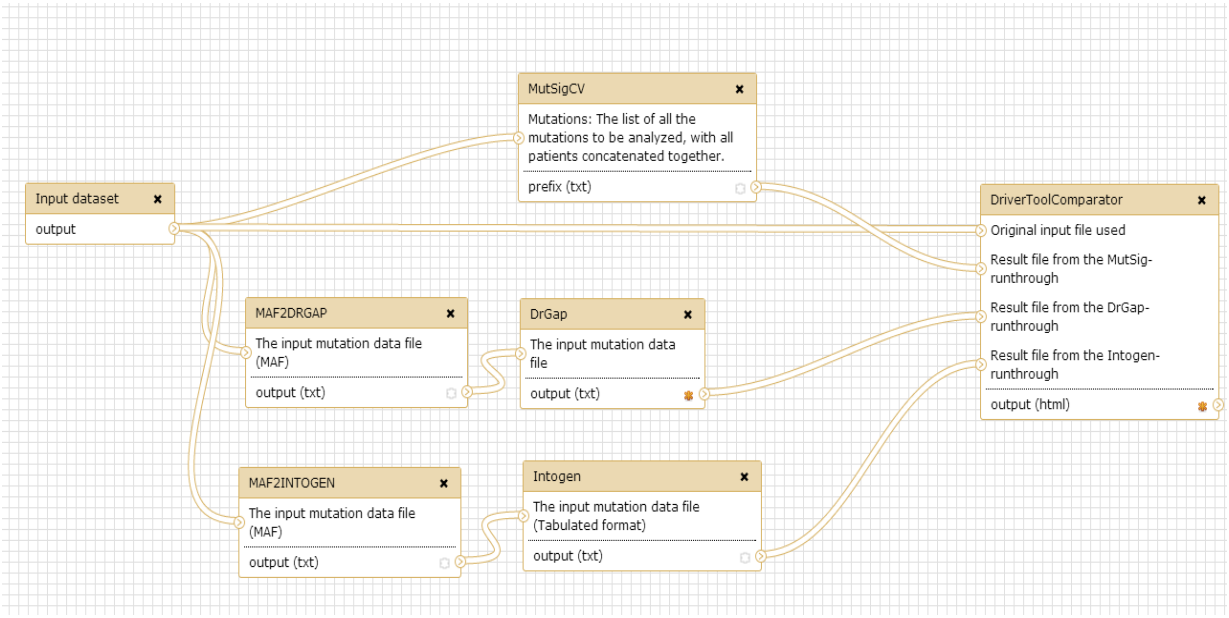


Figure 4

3.3 Installation documentation

3.3.1 MutSigCV

Downloading and installing MutSigCV itself is done by retrieving the zip from the Broad Institute website after registering and accepting their license. After extraction you have to download all the dependencies. Foremost, MutSigCV is dependent on Matlab to run, forcing you to either execute the "run_MutSigCV.sh" script within Matlab itself, or downloading the Matlab Compiler Runtime (MCR) and then referencing its path when running MutSigCV as shown in the figure below. Seeing as for this thesis every tool is implemented through Galaxy, we would have to use the MCR.

Furthermore, the following files are also required, to be downloaded separately:

1. A genome reference sequence, either hg18 or hg19
2. A mutation type dictionary file, detailing the different variant classifications and their effects relevant to MutSigCV
3. As we are using a single input file, thus not computing the coverage ourselves we have to use a pre-prepared file supplied by the MutSigCV developers, which they describe as a "territory" file, "A tabulation of how the reference sequence of the human exome breaks down by gene, categ and effect" (The Broad Institute)
4. A gene covariate file, used to calculate distances between pairs of genes in a "covariate space".

When running MutSigCV, as illustrated in the figure below, all these files are required as arguments when running the "run_MutSigCV.sh" script:

run_MutSigCV.sh <path_to_MCR> my_mutations.maf exome_full192.coverage.txt gene.covariates.txt result_prefix mutation_type_dictionary_file.txt chr_files_hg19

MutSigCV is the only tool in our driver tool comparison that can be successfully run with a MAF file as its only input, given that you have correctly given the paths to its dependencies. This makes MutSigCV fairly easy to implement compared to DrGap and Intogen.

The only issue with MutSigCV presents itself when it is time to handle its output. The result prefix is customizable, but the output files themselves will always go into the current working directory. This is not optional. When running MutSigCV from the terminal, this is manageable enough, but when running MutSigCV through Galaxy the working directory is set to a subdirectory within the Galaxy framework. The problem is solved by including the “from_work_dir” option in the Galaxy tool configuration file for MutSigCV as such:

```
<outputs>
    <data format="txt" name="mutations" label="mutations output"
from_work_dir="result_prefix.mutations.txt"/>
    <data format="txt" name="coverage" label="coverage output"
from_work_dir="result_prefix.coverage.txt"/>
    <data format="txt" name="prefix" label="prefix output"
from_work_dir="result_prefix.sig_genes.txt" />
 </outputs>
```

Galaxy thus retrieves the files with the given prefix from the working directory as output from MutSigCV.

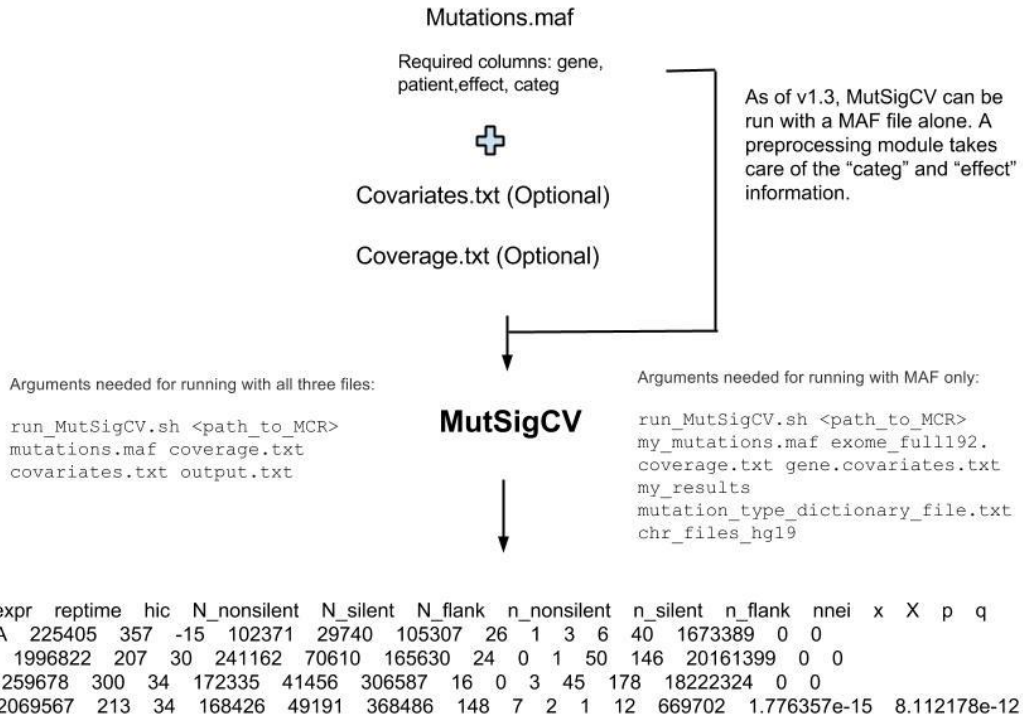


Figure 5

3.3.2 DrGap

To install and run DrGap, you download the tool from the google-code hosted site and then extract the contents of the tar file to a folder. DrGap requires the R library (http://www.r-project.org/) to run, a tool used for statistical computing as well as graphics. This is simple enough to download and install, given that you have the required privileges.

It is not its download and installation, but rather the custom made input file it requires that makes DrGap somewhat more of a task to run. It does not accept more standardized mutation files like MAF or VCF as its input. The creators have instead chosen to create their own format for the purpose of running data sets through DrGap. The format consists of the following headers:

Sample_ID	Gene	Chr	Pos	Ref	Var	Mutation_type
S0001	AHDC1	1	27874554	G	A	missense

The mutation type given has to be one of the following types: silent, missense, nonsense, splicing, Fs_indel, nFs_indel. No other mutation types will be accepted by DrGap.

DrGap takes three required arguments to run:

1. The input mutation data file, in the required format.
2. A predefined gene mutation table supplied by the DrGap creators.
3. The human reference genome file (hg19) in Fasta format.

In my Galaxy configuration file for DrGap I have also decided to specify the location for the output files given by DrGap, to avoid losing the files within the Galaxy working directory. When running DrGap without this option set to a preferred directory, the output files could not be found after a run. I have also added a preferred prefix, to make the output files easy to grab by the Driver Gene Comparison Tool.

DrGap is then run in the following way (illustrated in the figure below):

./drgap hg19_refgene.exp hg19.fasta output_path prefix

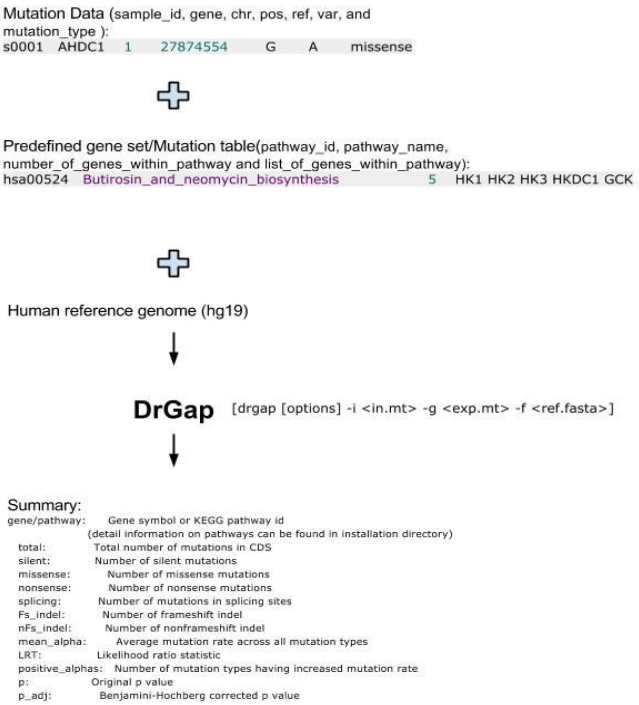


Figure 6

Output is given in the form of four files placed in the chosen output directory. They are distinguished by their suffixes:

1. Summary: This is the one we use in our implementation as it holds the p value required to pin the gene down as statistically significant or not.
2. Detail: A more detailed report on each gene and its different counts
3. Pdf: Graphical representation of mutation values created by R.
4. Log: A log of the output given by the DrGap run

3.3.3 Intogen

While Intogen is available to be run from the developer website, I had to implement a local installation of the tool for the purpose of comparing its results to the results of the other two tools. After downloading a small package, a setup script is required to be run before you can use the tool. Because the tool requires a vast array of other tools to be installed, the developers have opted for running Intogen in a virtual environment. After installing the python virtual environment, the following python libraries are downloaded and installed as part of the setup script (using pip):

- distribute 0.6.35
- requests 1.1.0
- Flask 0.10.1
- Flask-Login 0.2.7
- SQLAlchemy 0.8.2
- blinker 1.3
- Sphinx 1.2b1 or above
- pytz 2013b
- python-dateutil 2.1
- numpy 1.7.1
- scipy 0.12.0
- pandas 0.12.0
- statsmodels 0.4.3
- BgCore
- OncodriveFM
- OncodriveCLUST

While using pip to download and install all these libraries from predetermined locations, you inevitably run into problems with pip not finding all of the libraries, or not finding a distribution of the particular version it requires. In this case I had to install some of the libraries manually.

When installing Intogen on my personal laptop I ran into few problems. The main issues arose when attempting to install Intogen on the high performance computer I used (Insilico). A lot of time was spent trying to get Intogen to run on the HPC.

These are the lessons I have learned, hopefully it could save someone hours of debugging in the future:

- Remove the “--upgrade” arguments from the “lib/common.sh”-file in the Intogen directory. Those are redundant and will lead to some of the libraries (specifically bgcore) to fail its installation.

- “runtime/pyenv/lib/python2.7/site-packages/wok/core/cmd/native.py” contains some commented code on the HPC hindered python from retrieving shared libraries such as libpython which crashes Intogen. Morten Johansen **[Forklare mer om han?]** found that by uncommenting the stated code, Intogen ran successfully. The code that has to be uncommented in this case is (line 60):

```
#for k, v in os.environ.items():
#env[k] = v
```

When installation has completed successfully, you run Intogen with the following command:

```
./run analysis -p [Name of run] [Input file]
```

4 Results

The main purpose of this thesis has been to compare the results of three different tools created in order to identify driver genes in mutation data. This section will contain the results found in the reports created by our comparator (DGTC). Here we will find out how different or similar the tools really are, when boiled down to actual results, tested against different types of cancer.

4.1 Application of DGTC on cancer mutation datasets

The data used as input has been downloaded from The Cancer Genome Atlas's MAF file archive. This site is hosted by The National Cancer Institute (http://www.cancer.gov/) and provides free input data on several different types of cancer. The files themselves can be accessed here: https://wiki.nci.nih.gov/display/TCGA/TCGA+MAF+Files

The results are presented in the format X | Y CGC*, where X is the number of genes predicted to be drivers by the tool and Y is the number of genes in the list X represented in the Cancer Gene Census.

It is important to note that when results are given where more than one tool is involved, intersection is used, not union. An example would be the result of "81 | 7 CGC*" given as the result of running MutSigCV and DrGap on the Colon Adenocarcinoma data set. This means that "81" is the number of genes **both** MutSigCV and DrGap have predicted as driver genes for this data set.

4.1.1 Colon Adenocarcinoma

Dataset: hgsc.bcm.edu_COAD.SOLiD_DNASeq.1.somatic.maf
Number of samples: 53
Deployed: 30/07/2013

Number of genes predicted as drivers by MutSigCV: 121 | 8 CGC*
Number of genes predicted as drivers by DrGap: 537 | 25 CGC*
Number of genes predicted as drivers by Intogen: 129 | 55 CGC*

The gene predictions made by the tools overlaps in the following way:

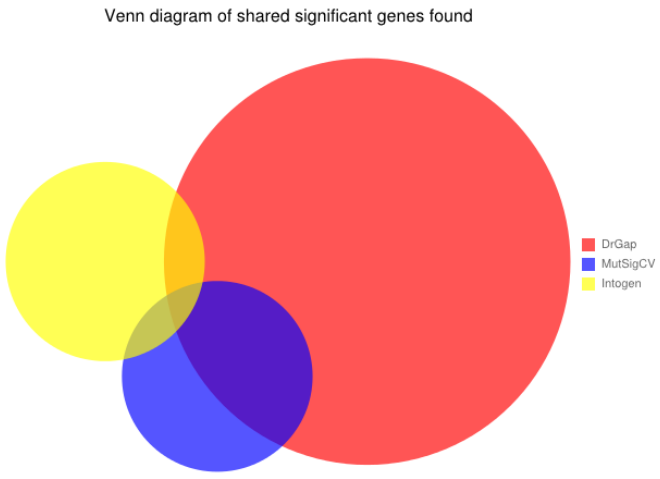
	DrGap	**MutSigCV**	**Intogen**
DrGap		81 | 7 CGC*	25 | 14 CGC*
MutSigCV	81 | 7 CGC*		7 | 6 CGC*
Intogen	25 | 14 CGC*	7 | 6 CGC*	

*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

APC	TP53	FBXW7	KRAS	SMAD4
CASP8	NRAS			

Note: Genes in red not found in CGC file.



For the purpose of checking the validity of the results given by the tool, we used the Genetics Home Reference (http://ghr.nlm.nih.gov/) in order to cross reference the identified genes for this dataset with the database of the National Insitute of Health.

APC

The Genetics Home Reference (GHR) confirms that mutations in the APC gene has led to desmoid tumors (noncancerous growth) and familial adenomatous polyposis (FAP) . In the case of the ladder, GHR states that most people with FAP will develop colorectal cancer.

(http://ghr.nlm.nih.gov/gene/APC)

TP53

This gene is in charge of providing instructions for the creation of a protein called "Tumor protein p53". Acting as a tumor suppressor, TP53 regulates cell division. TP53 is the most commonly changed gene found in human cancer, occurring in about half of all cancer. It is most prevalent however, in breast- and bladder cancer.

(http://ghr.nlm.nih.gov/gene/TP53)

FBXW7

This gene is a member of what is called the F-box protein family. Mutations in this gene have been found in the cell lines of ovarian- and breast cancers, which suggest that the gene might be involved with the pathogenesis of cancer in humans.

(http://www.genecards.org/cgi-bin/carddisp.pl?gene=FBXW7)

KRAS

KRAS provides instruction for the creation of a protein called K-Ras. This protein is involved primarily in regulating cell division. According to GHR, studies suggest that mutations in the KRAS gene are common in pancreatic, lung and colorectal cancers.

(ttp://ghr.nlm.nih.gov/gene/KRAS)

SMAD4

Mutations in the SMAD4 gene, which instructs the creation of a protein involved in transmitting signals from the cell surface to the nucleus, can lead to several different afflictions according to GHR. In the case of cancer, mutations in the SMAD4 are most commonly associated with cancer in the colon or pancreas. It seems that even though SMAD4 is not in the Cancer Gene Census, it has a known connection with cancer development.

(http://ghr.nlm.nih.gov/gene/SMAD4)

CASP8

The GHR suggests that the protein encoded by CASP8 plays a central role in the execution-phase of cell apoptosis (cell death). GHR suggests that mutations of the CASP8 gene might be connected to breast cancer, hepatocellular carcinoma (the most common type of liver cancer) and lung cancer. While the GHR does not suggest that CASP8 could be connected to colorectal cancer, a study done by Kim HS et al. concludes the following:

The presence of caspase-8 mutation in colon carcinomas suggests that caspase-8 gene mutation might lead to the loss of its apoptotic function and contribute to the pathogenesis of colorectal carcinomas, especially at the late stage of colorectal carcinogenesis.

(http://www.ncbi.nlm.nih.gov/pubmed/12949717)

(http://ghr.nlm.nih.gov/gene/CASP8)

NRAS

By providing the instructions for the creation of the protein called N-Ras, which is involed in regulating cell division, mutations in the NRAS gene has been associated with the development of several types of cancer. GHR notes mutations in NRAS are especially common in melanoma, an aggressive form of skin cancer.

(http://ghr.nlm.nih.gov/gene/NRAS)

4.1.2 Breast Invasive Carcinoma

Dataset: genome.wustl.edu_BRCA.IlluminaGA_DNASeq.Level_2.5.3.0.somatic.maf
Number of samples: 776 : 801 (Tumor samples : Normal Samples)
Deployed: 24/06/2013

Number of genes predicted as drivers by MutSigCV: 484 | 28 CGC*
Number of genes predicted as drivers by DrGap: 2805 | 86 CGC*
Number of genes predicted as drivers by Intogen: 228 | 83 CGC*

The gene predictions made by the tools overlaps in the following way:

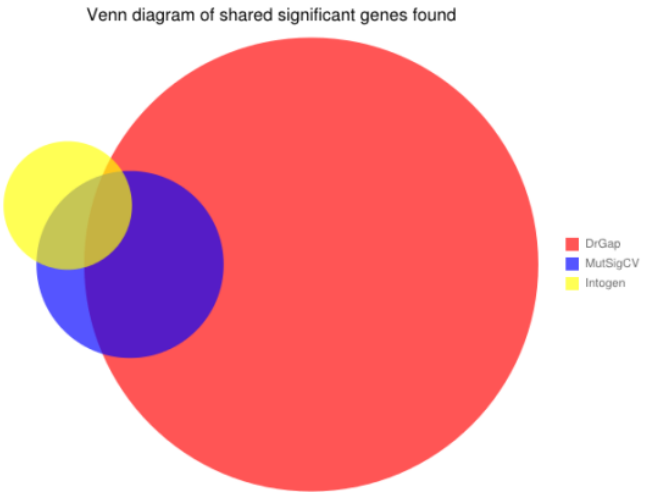
	DrGap	**MutSigCV**	**Intogen**
DrGap		433 | 28 CGC*	84 | 36 CGC*
MutSigCV	433 | 28 CGC*		32 | 19 CGC*
Intogen	84 | 36 CGC*	32 | 19 CGC*	

*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

HLA-B	KRAS	PIK3CA	TP53	ARID1A
SMAD4	CTCF	APC	FBXW7	CASP8
PTEN	CDKN2A	ELF3	ARID2	HLA-A
CTNNB1	ACVR1B	CUL1	NF1	CDH1
PIK3R1	CBFB	SOX9	ABCB1	MLL2
RPL22	TGFBR2	MAP2K4	GATA3	TTK
ACVR2A	NRAS			

Note: Genes in red not found in CGC file.


Venn diagram of shared significant genes found
DrGap
MutSigCV
Intogen

4.1.3 Prostate Adenocarcinoma

Dataset: hgsc.bcm.edu_PRAD.IlluminaGA_DNASeq.1.somatic.maf
Number of samples: 263:259 (Tumor Samples:Normal Samples)
Deployed: 08/01/2014

Number of genes predicted as drivers by MutSigCV: 131 | 10 CGC*
Number of genes predicted as drivers by DrGap: 644 | 33 CGC*
Number of genes predicted as drivers by Intogen: 145 | 54 CGC*

	DrGap	MutSigCV	Intogen
DrGap		98 | 10 CGC*	31 | 17 CGC*
MutSigCV	98 | 10 CGC*		15 | 8 CGC*
Intogen	31 | 17 CGC*	15 | 8 CGC*	

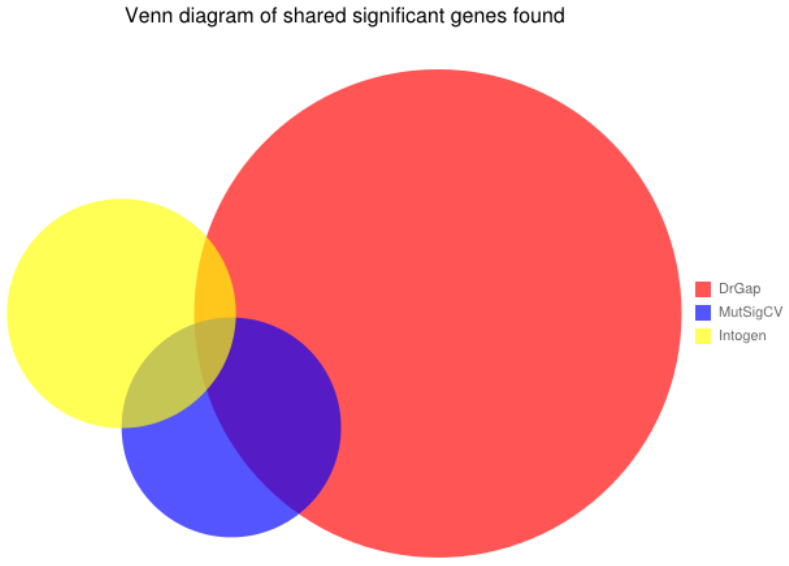
*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

TP53	SPOP	PTEN	CDK12	FOXA1
CDKN1B	CTNNB1	KDM6A	PIK3CA	QKI
IDH1	TP53BP1	SMAD4		

Note:

Genes in red not found in CGC file.



4.1.4 Lung Squamous Cell Carcinoma

Dataset: step4_LUSC_Paper_v8.aggregated.tcga.maf2.4.migrated.somatic.maf
Number of samples: 178:178 (Tumor Samples:Normal Samples)
Deployed: 27/06/2013

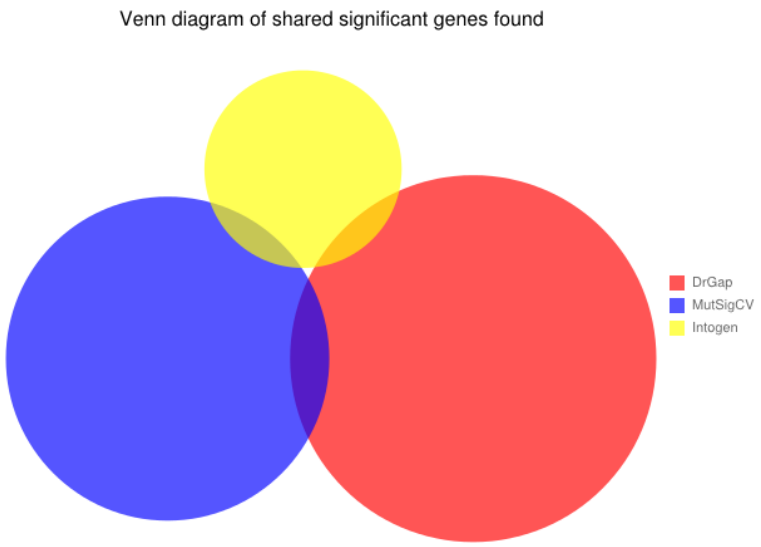
Number of genes predicted as drivers by MutSigCV: 266 | 13 CGC*
Number of genes predicted as drivers by DrGap: 340 | 15 CGC*
Number of genes predicted as drivers by Intogen: 100 | 30 CGC*

	DrGap	**MutSigCV**	**Intogen**
DrGap		25 | 4 CGC*	9 | 5 CGC*
MutSigCV	25 | 4 CGC*		10 | 6 CGC*
Intogen	9 | 5 CGC*	10 | 6 CGC*	

*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

TP53	CDKN2A	KEAP1	NFE2L2	MLL2
KDM6A				



4.1.5 Brain Lower Grade Glioma

Dataset: hgsc.bcm.edu_LGG.IlluminaGA_DNASeq.1.somatic.maf
Number of samples: 289:289 (Tumor samples: Normal samples)
Deployed: 10/12/2013

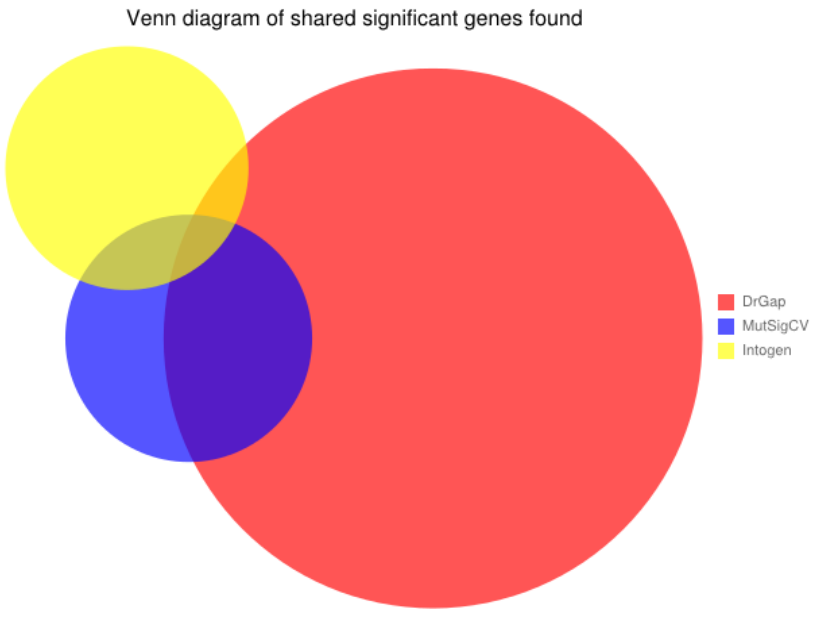
Number of genes predicted as drivers by MutSigCV: 138 | 22 CGC*
Number of genes predicted as drivers by DrGap: 644 | 38 CGC*
Number of genes predicted as drivers by Intogen: 136 | 49 CGC*

	DrGap	MutSigCV	Intogen
DrGap		123 | 21 CGC*	30 | 23 CGC*
MutSigCV	123 | 21 CGC*		16 | 16 CGC*
Intogen	30 | 23 CGC*	16 | 16 CGC*	

*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

ATRX	IDH1	TP53	PIK3CA	PTEN
PIK3R1	IDH2	NOTCH1	SMARCA4	ARID1A
EGFR	PTPN11	MAX	NF1	RB1
RPL22				



4.1.6 Skin Cutaneous Melanoma

Dataset: hgsc.bcm.edu_SKCM.IlluminaGA_DNASeq.1.somatic.maf
Number of samples: 344:344 (Tumor samples: Normal Samples)
Deployed: 17/04/2014

Number of genes predicted as drivers by MutSigCV: 82 | 14 CGC*
Number of genes predicted as drivers by DrGap: 2126 | 67 CGC*
Number of genes predicted as drivers by Intogen: 232 | 84 CGC*

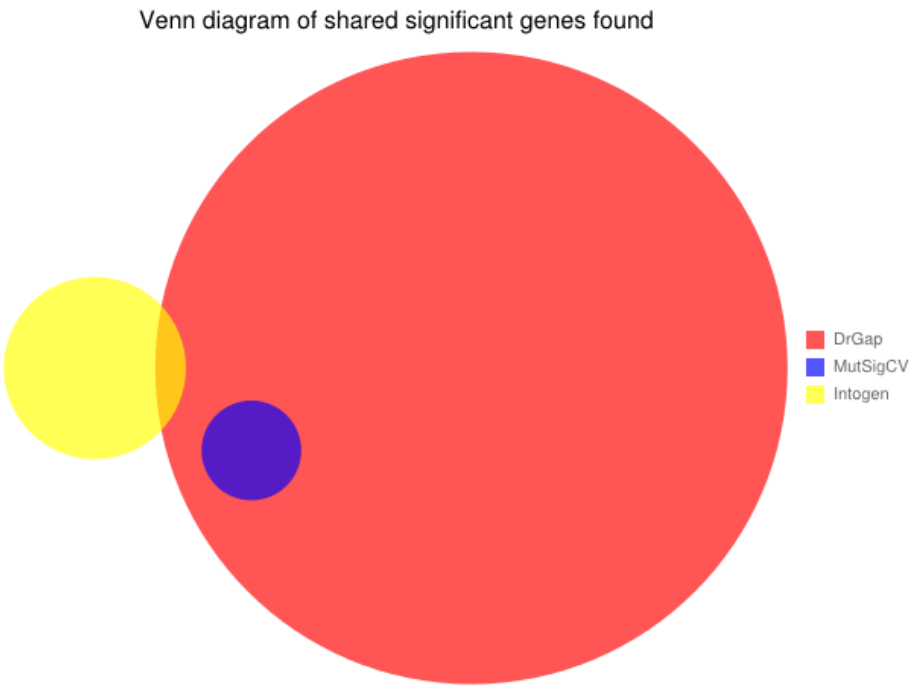
	DrGap	**MutSigCV**	**Intogen**
DrGap		67 | 14 CGC*	44 | 29 CGC*
MutSigCV	67 | 14 CGC*		14 | 13 CGC*
Intogen	44 | 29 CGC*	14 | 13 CGC*	

*Number of genes from this list found in the Cancer Gene Census

Genes found to be significant by all three tools:

CDKN2A	NRAS	TP53	BRAF	PTEN
RAC1	HRAS	RPL5	NF1	DDX3X
RPL22	RB1	ARID2	CTNNB1	

*Genes in red not found in CGC file, genes with no link not found in NCBI database



4.1.7 Result summary

Single tool results:

Cancer Type	MutSigCV	DrGap	Intogen
Colon	121 (0.07)	537 (0.05)	129 (0.43)
Breast	484 (0.06)	2805 (0.03)	228 (0.36)
Prostate	131 (0.08)	644 (0.05)	145 (0.37)
Lung	266 (0.05)	340 (0.04)	100 (0.3)
Brain	138 (0.16)	644 (0.06)	136 (0.36)
Skin	82 (0.17)	2126 (0.03)	232 (0.36)
Pan-cancer	1222 (0.08)	6452 (0.04)	970 (0.43)

Note: Numbers given in parenthesis is the fraction of the genes in the given list identified by the Cancer Gene Census

More than one tool used:

Cancer Type	MutSigCV + DrGap	MutSigCV + Intogen	DrGap + Intogen	All three tools
Colon	81 (0.09)	7 (0.85)	25 (0.56)	7 (0.85)
Breast	433 (0.06)	32 (0.59)	84 (0.43)	32 (0.59)
Prostate	98 (0.1)	15 (0.53)	31 (0.55)	13 (0.6)
Lung	25 (0.16)	10 (0.6)	9 (0.55)	6 (0.67)
Brain	123 (0.17)	16 (1)	30 (0.77)	16 (1)
Skin	67 (0.21)	14 (0.93)	44 (0.66)	14 (0.93)
Pan-cancer	827 (0.1)	94 (0.72)	223 (0.56)	88 (0.75)

Note: Numbers given in parenthesis is the fraction of the genes in the given list identified by the Cancer Gene Census

Cancer Gene Census accuracy (Single tool)

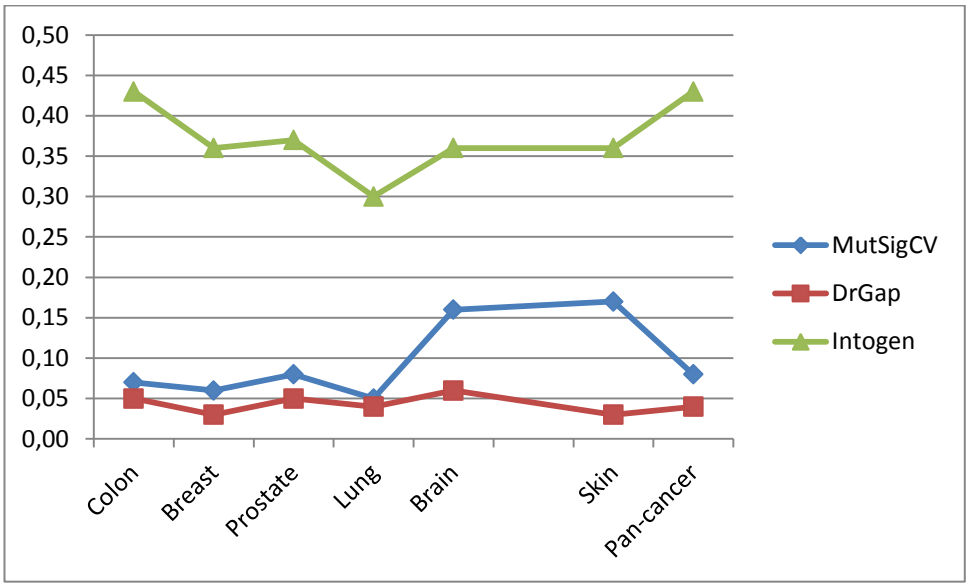


Figure 7: This chart illustrates the accuracy of the different tools when compared to the list given by the Cancer Gene Census. A score of 1 for a given cancer type means that all of the identified genes by the tool are identified by the Cancer Gene census.

Cancer Gene Census accuracy (more than one tool)

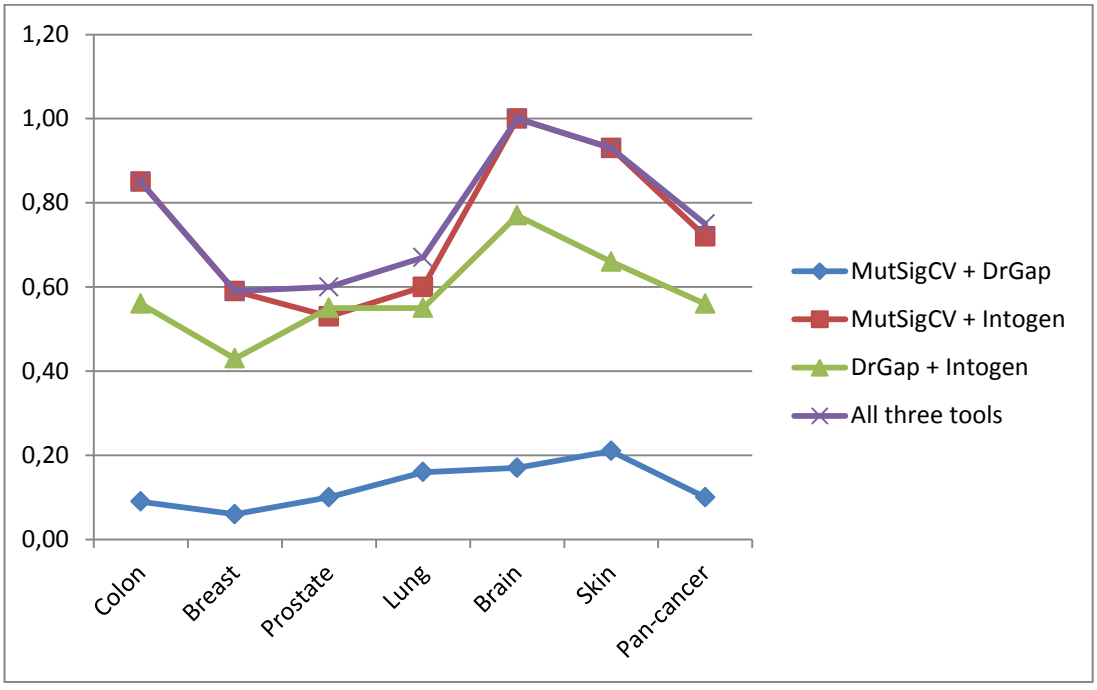


Figure 8: This chart illustrates the accuracy of the different tools when compared to the list given by the Cancer Gene Census. A score of 1 for a given cancer type means that all of the identified genes by the tool are identified by the Cancer Gene census.

4.2 Discussion

4.2.1 On Implementation

When attempting to implement the three tools used in this comparison there were a few areas of concern: The downloading and installation of the tools themselves, handling input, output, and dependencies as well as taking care of the implementation/configuration of the tools into Galaxy.

Even though the documentation of the tools might be good (at least in the cases of MutSigCV and Intogen) the user bases are quite small (DrGap has been downloaded 107 times as of 25/04/2014). When faced with problems, finding people who have had similar issues in the past proved difficult, as there is no large community of people to reach out to. Because of this, we had to communicate with the tool developers directly when faced with issues. The Intogen developers were the only team that had implemented a way to contact them with questions regarding the implementation of their tool through a form on their website. Implementing such a form makes it easier for the user to contact the developers with questions, instead of having to send out emails to specific people.

As there are no standards for input used for the three tools, we searched for tools used to handle input for the different tools, but no such tools seems to exist, at least not readily. Because of this, we had to write them ourselves (MAF2INTOGEN and MAF2DRGAP). Anyone who wishes to run their own installations of DrGap or Intogen has to adhere to their custom formats. Now, after downloading the two tools we have created from google code, all you need to run all three tools used in this thesis will be a single MAF file.

4.2.2 Results

In june 2013, a study called “Comprehensive identification of mutational cancer driver genes across 12 tumor types” was conducted by David Tamborero, Abel Gonzales-Perez and their colleagues (Tamborero, Gonzalez-Perez et al. 2013). They hoped to show that applying a combination of complementary methods allows identifying a comprehensive and reliable list of cancer driver genes. The tools the team chose to use were MuSIC-SMG, OncodriveFM, OncodriveCLUST and ActiveDriver. Individually, these tools are looking for four different things in order to classify a gene as a driver:

1. More frequent mutations than expected from the background mutation rate, or rather, genes that are significantly mutated (MuSIC-SMG)
2. A bias towards the accumulation of functional mutations (OncodriveFM)
3. Exploiting the tendency to sustain mutations in certain regions of the protein sequence (OncodriveCLUST)
4. The overrepresentation of mutations in specific functional residues, such as phosphorylation sites (ActiveDriver)

They applied these tools to data from 12 different cancer types and created a list of 291 high-confidence cancer driver genes:

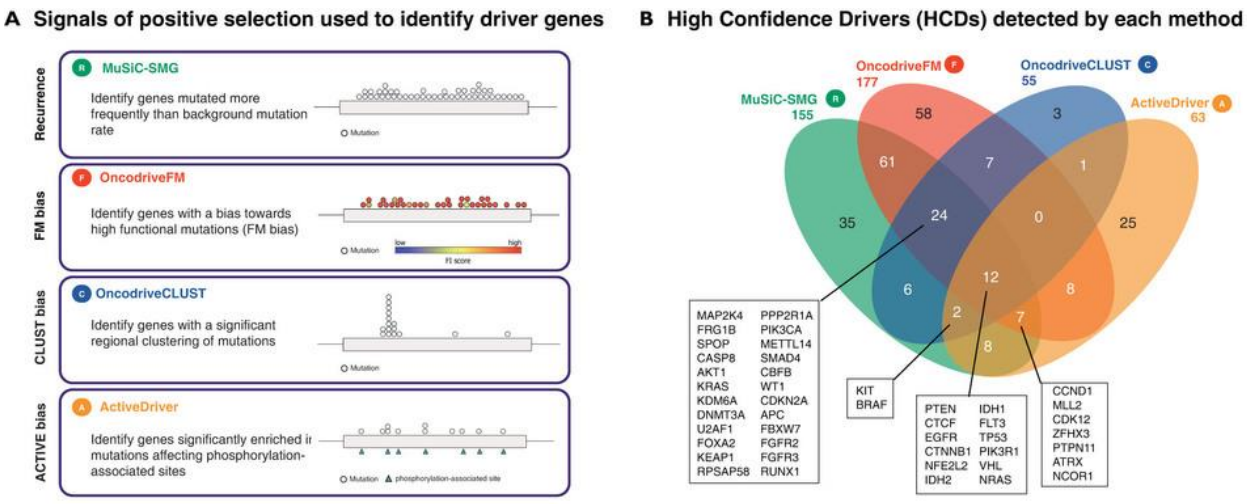


Figure 9

In order to provide some validation to their results, like we have opted to do in this project, the researchers compared their results with the list of known cancer drivers provided by the Cancer Gene Census (CGC). Here it's important to note that while CGC is the most reliable catalog of known cancer genes to date, the fact that, arguably, many cancer genes are yet to be uncovered means that CGC can only serve as a surrogate estimator of the accuracy of each method applied. Genes not found in the CGC, but identified by the tools might still be cancer driver genes, but are not yet recognized by the CGC. Because of this, it is important to recognize that false positives or negatives might occur.

Like in our thesis, they have supplied the fraction of genes in the lists given by the tools and tool-combinations in parenthesis in the figure below:

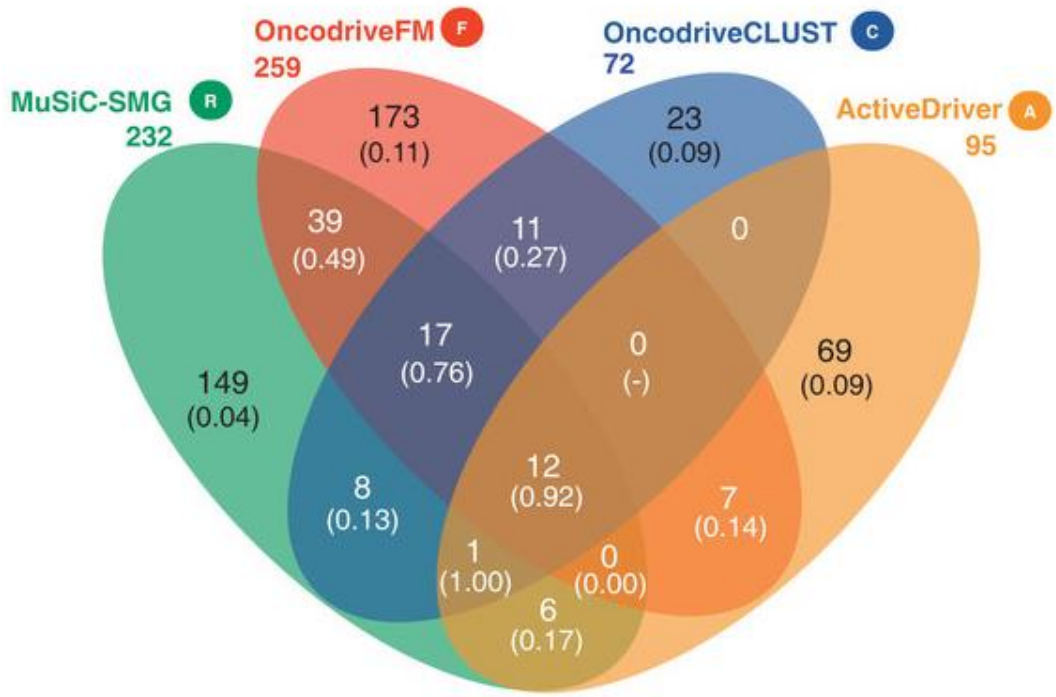


Figure 10

It is easy to spot that the combination of tools greatly increases the accuracy of the gene lists when compared with the CGC.

In many ways we have automated the process implemented in the study done by Tamborero, Abel Gonzales-Perez and their colleagues. While Intogen applies OncodriveFM and OncodriveCLUST, MuSIC-SMG and ActiveDriver have been swapped for the more modern approaches of MutSigCV and DrGap.

So how do these results compare to the results we have come up with in this thesis? Across all cancers (pan-cancer) the results corresponds with the following chart:

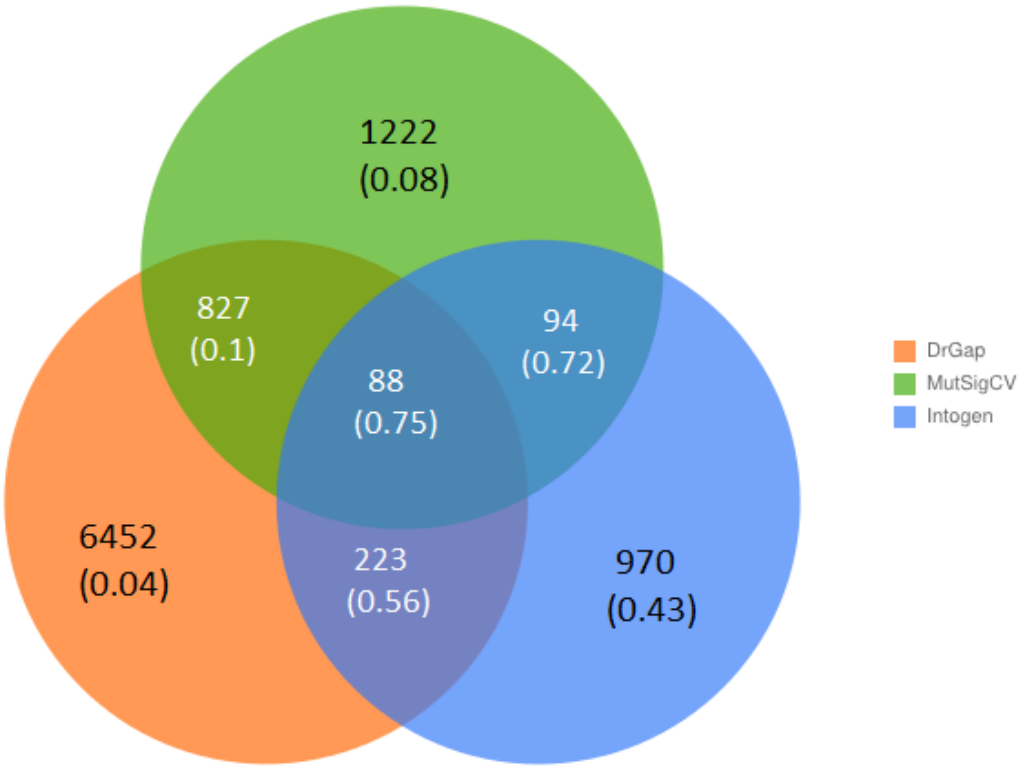


Figure 11

In the case of DrGap, individually it never exceeds an accuracy of 6% compared to 17% for MutSigCV and the quite substantial 43% of Intogen.

Intogen is fairly accurate in itself (while it's actually a combination of OncodriveFM and OncodriveCLUST, as mentioned earlier), at 43% accuracy. When combined with MutSigCV (at 8% accuracy individually), however, it is able to reach an accuracy of 72%. When Intogen is combined with DrGap, with an individual performance at 4% accuracy, the combination reaches an accuracy of 56%, which is an increase of 13%. These two findings show that a tool's individual performance might not be a good indicator of how well it works in combination with other tools. This matches well with the findings of Tamborero, Abel Gonzales-Perez and their colleagues as can be seen in the graphs in this section.

The combination of MutSigCV and Intogen often provides a large increase in accuracy when compared to the individual runs of the tools. For example, the individual results on skin cancer are 17% and 36% accuracy individually, for MutSigCV and Intogen respectively. Together, however, they reach an accuracy of 93%, with 13 of the 14 genes identified in by the Cancer Gene Census.

It quickly becomes apparent that using more than one approach to identifying driver genes increases the accuracy when compared to the CGC. Individually, the tools are returning large amounts of genes identified as driver genes, with often as little as 3%-5% accuracy. Individually the tools range in accuracy from 3% to 43%, while in combination the accuracy ranges from 10% to 93%.

5 Conclusion

By running our Driver Gene Comparsion Tool on six different datasets across six different types of cancer, we were able to show that the intersecting results of more than one tool created for the purpose of identifying driver genes greatly exceeded the results given by the tools individually.

The combination of several approaches towards the identification of driver genes has proven to narrow down the often wide net of results cast by the individual approaches, greatly improving the accuracy of the results.

References

*Adzhubei IA, Schmidt S, Peshkin L, Ramensky VE, Gerasimova A, Bork P, Kondrashov AS, Sunyaev SR. 2010. A method and server for predicting damaging missense mutations. Nat Methods **7**: 248–249.*

*Akavia UD, Litvin O, Kim J, Sanchez-Garcia F, Kotliar D, Causton HC, Pochanard P, Mozes E, Garraway LA, Pe'er D. 2010. An integrated approach to uncover drivers of cancer. Cell **143**: 1005–1017.*

Brown TA. 2006. Genomes. Garland Science.

*De S, Michor F. 2011. DNA replication timing and long-range DNA interactions predict mutational landscapes of cancer genomes. Nat Biotechnol **29**: 1103–1108.*

Dees ND, Zhang Q, Kandoth C, Wendl MC, Schierding W, Koboldt DC, Mooney TB, Callaway MB, Dooling D, Mardis ER, et al. 2012. MuSiC: Identifying mutational significance in cancer genomes. Genome Res.

*Garraway LA, Lander ES. 2013. Lessons from the cancer genome. Cell **153**: 17–37.*

*Giardine B, Riemer C, Hardison RC, Burhans R, Elnitski L, Shah P, Zhang Y, Blankenberg D, Albert I, Taylor J, et al. 2005. Galaxy: a platform for interactive large-scale genome analysis. Genome Res **15**: 1451–1455.*

Gonzalez-Perez A, Lopez-Bigas N. 2012. Functional impact bias reveals cancer drivers. Nucleic Acids Res.

Gonzalez-Perez A, Perez-Llamas C, Deu-Pons J, Tamborero D, Schroeder MP, Jene-Sanz A, Santos A, Lopez-Bigas N. 2013. IntOGen-mutations identifies cancer drivers across tumor types. Nat Methods.

*Greenman C, Stephens PJ, Smith R, d' Aubenton-Carafa GL, Hunter C, Bignell GR, Davies HR, Teague JW, Butler AP, Stevens C, et al. 2007. Patterns of somatic mutation in human cancer genomes. Nature **446**: 153–158.*

*Henikoff S. 2001. Predicting deleterious amino acid substitutions. Genome Res **11**: 863–874.*

*Hua X, Xu H, Yang Y, Zhu J, Liu P, Lu Y. 2013. DrGaP: a powerful tool for identifying driver genes and pathways in cancer sequencing studies. Am J Hum Genet **93**: 439–451.*

*Lawrence MS, Stojanov P, Polak P, Kryukov GV, Cibulskis K, Sivachenko A, Carter SL, Stewart C, Mermel CH, Roberts SA, et al. 2013. Mutational heterogeneity in cancer and the search for new cancer-associated genes. Nature **499**: 214–218.*

Miko I, Lejeune L. 2009. Essentials of Genetics. Cambridge, MA: NPG Education.

O'Connor CM, Adams JU. 2010. Essentials of Cell Biology. Cambridge, MA: NPG Education.

*Pearson H. 2006. Genetics: what is a gene? Nature **441**: 398–401.*

*Reimand J, Bader GD. 2013. Systematic analysis of somatic mutations in phosphorylation signaling predicts novel cancer drivers. Mol Syst Biol **9**: 637.*

*Reva B, Antipin Y, Sander C. 2011. Predicting the functional impact of protein mutations: application to cancer genomics. Nucleic Acids Res **39**: e118.*

*Stehr H, Jang S-HJ, Duarte JM, Wierling C, Lehrach H, Lappe M, Lange BMH. 2011. The structural impact of cancer-associated missense mutations in oncogenes and tumor suppressors. Mol Cancer **10**: 54.*

*Stratton MR, Campbell PJ, Futreal PA. 2009. The cancer genome. Nature **458**: 719–724.*

*Tamborero D, Gonzalez-Perez A, Lopez-Bigas N. 2013a. OncodriveCLUST: exploiting the positional clustering of somatic mutations to identify cancer genes. Bioinformatics **29**: 2238–2244.*

*Tamborero D, Gonzalez-Perez A, Perez-Llamas C, Deu-Pons J, Kandoth C, Reimand J, Lawrence MS, Getz G, Bader GD, Ding L, et al. 2013b. Comprehensive identification of mutational cancer driver genes across 12 tumor types. Sci Rep **3**: 2650.*

*Vandin F, Upfal E, Raphael BJ. 2012. De novo discovery of mutated driver pathways in cancer. Genome Res **22**: 375–385.*

*Vogelstein B, Papadopoulos N, Velculescu VE, Zhou S, Diaz LA, Kinzler KW. 2013. Cancer genome landscapes. Science **339**: 1546–1558.*